COMPUTATIONAL GASTRONOMY: AN ARTIFICIAL INTELLIGENCE DRIVEN APPROACH TO RECIPES, FLAVORS, NUTRITION, HEALTH, AND SUSTAINABILITY

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

BY

MANSI GOEL (PHD20205)

UNDER SUPERVISION OF

PROF. GANESH BAGLER

DEPARTMENT OF COMPUTATIONAL BIOLOGY
INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

August 2025

Certificate

This is to certify that the thesis titled **Computational Gastronomy: An Artificial Intelligence Driven Approach to Recipes, Flavors, Nutrition, Health, and Sustainability** submitted by **Mansi Goel** (PhD20205), to the Indraprastha Institute of Information Technology Delhi (IIIT-Delhi), for the award of the degree of **Doctor of Philosophy**, is a bonafide record of the research work done by her under my supervision. I consider that this thesis fulfills the requirement of rules and regulations.

The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Ganesh Bagler
Thesis Supervisor
Professor
Department of Computational Biology
IIIT-Delhi, India, 110020

Acknowledgements

First and foremost, I am deeply grateful to my PhD supervisor Prof. Ganesh Bagler for his invaluable guidance, encouragement, and unwavering support throughout my PhD journey. His expertise, insightful feedback, and vision have been instrumental in shaping both my academic and personal growth. I am sincerely thankful for the opportunities and knowledge he has shared, which have enriched this dissertation.

I extend my sincere thanks to my PhD monitoring committee members, Prof. GPS Raghava and Prof. Debarka Sengupta, for their constructive feedback and and thoughtful suggestions during my annual progress reviews. Their guidance helped refine my research direction and strengthened the quality of my work. I also thank Prof. Ranjan Bose (Director, IIIT-Delhi) for giving me the opportunity to pursue my PhD at IIIT-Delhi and for providing the necessary resources. I sincerely appreciate the support of the administrative and IT staff at IIIT-Delhi, Mrs. Shipra Jain, Mr. Raju Biswas, Mrs. Anshu Dureja, Mr. Imran Khan, Mrs. Sarika, Mr. Mohit, Mr. Kapil Dev Garg, Mr. Adarsh, for their prompt and consistent support.

I am grateful for the opportunity to intern at the National Institute of Informatics (NII) Japan under the mentorship of Prof. Frederic Andres. This experience broadened my academic horizons and enriched my research perspective. I deeply appreciate his guidance, support, and the collaborative environment he fostered during my time there. I extend my sincere gratitude to all my collaborators and co-authors who have contributed to my research. Their valuable insights, discussions, and contributions have significantly enriched my work. Special thanks to my yoga guru, Mr. Ajay Saxena, for contributing to my mental well-being throughout this journey.

I am immensely grateful to my colleagues, lab mates, and seniors for fostering a collaborative, supportive, and enjoyable work environment. Their willingness to exchange ideas and provide guidance has enriched my research experience. I extend my heartfelt gratitude to my seniors, Dr. Sumeet Patiyal, Dr. Anjali Dhall, and Dr. Neetesh Pandey for their guidance and mentorship throughout my journey. A special mention to Mr. Gaurav Rai, for his intellectual and academic support, especially during the most challenging phases of my journey. His encouragement and thoughtful presence have been a source of strength and motivation. My sincere thanks go to my CoSyLab juniors Vibhuti Khanduri, Harshika Sharma, and Madhvi Sinha for their enthusiasm and support. My heartfelt thanks extend to my friends and colleagues Vishakha Gautam, Mohit Sharma, Sakshi Arora, Shubham Choudhury, Nishant Kumar, Megha Sundriyal, Himanshu Singh, Asra Sakeen, Dr. Madhu Sharma, Gunjan Singh, Nisha Bajiya, Ritu Tomar,

Sukriti Sacher, Saveena Solanki, Naman Mehta, Gayatri Panda, Pradeep Singh, Aayushi Mittal, and Sanjay Mohanty who have been a part of my journey, offering both companionship and motivation. Their presence has made this experience more fulfilling. I would like to thank my dear friends, Nikitasha Sharma and Sakshi Tandon, who have remained a constant source of emotional support, laughter, and encouragement throughout my PhD. Their unwavering belief in me and their presence during both high and low points have meant the world to me, and I am truly grateful for their enduring friendship.

On a personal note, I am forever indebted to my parents, Mr. Kuldeep Goel and Mrs. Alka Goel, for their unconditional love, support, and encouragement. Their belief in me has been my greatest strength. A special thanks to my sister, Dr. Kritika Goel, my brother-in-law, Dr. Shashank Verma, and my niece, Kiansh, whose constant support and motivation have meant the world to me. I am also deeply grateful to my extended family, friends, and well-wishers for their love, encouragement, and unwavering support.

This milestone would not have been possible without the contributions and encouragement of all those who have been part of this journey, directly or indirectly. This acknowledgment is a small token of my immense gratitude towards everyone who has supported me along the way.

Mansi Goel

Abstract

Cooking is a quintessential creative pursuit with profound significance for humanity. Food and cooking transcend mere sensory pleasure and have serious nutrition and public health outcomes. Beyond being linked to the culinary and cultural heritage, food systems play a pivotal role in sustainability and are therefore critical for the very survival of life on the Earth. In an era where data-driven and computational insights are transforming every domain, culinary endeavors have primarily been seen through an artistic outlook. This thesis delves into various facets of computational gastronomy, presenting a comprehensive framework for the organization and analysis of cooking recipes, novel recipe generation, algorithmic creativity, molecular informatics, and sustainability. The work presented here bridges gaps in food science, flavor science, and food system studies by addressing the challenges in the structure of recipes, prediction of molecular flavor, and generating novel recipes.

The journey of this thesis starts with the development of RecipeDB2, a scalable and structured framework for representing recipes, associated ingredients, and nutritional profiles of diverse cuisines from across the globe. Building upon RecipeDB2, a recipe ontology was introduced, which integrates recipes, ingredients, and flavor molecules. Such an ontology, which incorporates diverse datasets, including RecipeDB2 and FlavorDB2, serves as the backbone for subsequent analyses, enabling semantic reasoning and supporting various machine-learning tasks. To build on this structured representation, we implemented deep-learning-based named entity recognition algorithms on recipes, extracting meaningful entities from unstructured ingredient phrases. After creating a structured database of recipes, we delved into novel recipe generation by fine-tuning large language models in a generic as well as cuisine-specific manner. These machine-generated recipes were further evaluated through the Turing Test for Chefs to assess the efficacy of novel recipe generation strategies.

Sustainability is another pivotal theme in this thesis. By evaluating the carbon footprint of recipes, one can assess the environmental impact of tradition-driven culinary choices, thus enabling informed decision-making for a sustainable food future. Apart from culinary science, this thesis delves into molecular informatics by introducing FlavorDB2, a dataset that documents the molecular composition and flavor profiles of natural ingredients. By understanding the chemical properties of flavors, this research deepens the understanding of taste and its potential applications in culinary arts. Aligned with this thread, the thesis explores the application of machine learning and deep learning models to predict the taste (sweet, umami) and toxicity of small molecules, towards addressing challenges in food safety, lifestyle disorders, and culinary innovation. The thesis culminates with a real-world application of computer vision for dish

detection in food platters. We showcase the use of object detection techniques for correctly identifying dishes from the Indian platters, as a proof of concept, with potential applications for meal logging, diet management, and personalized nutrition.

This thesis offers a multidimensional perspective on computational gastronomy by integrating structured data, natural language processing, large language models, deep learning, molecular informatics, and computer vision. It presents the story of how Artificial Intelligence can mimic human creativity and enhance our understanding of food, enabling innovations that foster better health, and environmental sustainability. By bridging the culinary arts with data science and computation, this work contributes to the growing interdisciplinary field of computational gastronomy, shaping the future of food through computational approaches. Computational gastronomy offers a data-driven approach to food, paving the way for ground-breaking advancements in the culinary landscape. This emerging field defines the traditional, artistic outlook toward food and cooking, demonstrating how the fusion of food, data, and computation can lead to innovative and sustainable culinary solutions.

Keywords - Structured Datasets, Natural Language Processing, Novel Recipe Generation, Carbon Footprint, Sustainability, Object Detection, Food Platter, Webserver, Sweetness Prediction, Toxicity Prediction, Machine Learning, Deep Learning, Artificial Intelligence, Regression Models, Database

Table of Contents

List of Tables

List of Figures

Abbreviations

AI : Artificial Intelligence

AR : AdaBoost Regressor

BERT : Bidirectional Encoder Representations from Transformers

CNN : Convolutional Neural Network

COMFA : Comparative Molecular Field Analysis

COMSIA : Comparative Molecular Similarity Indices Analysis

DNN : Deep Neural Network

DT : Decision Tree

ET : ExtraTrees

GAN : Generative Adversarial Network

GBM : Gradient Boosting Machine

GBR : Gradient Boosting Regressor

GCN : Graph Convolutional Network

GNN : Graph Neural Network

GPT-2 : Generative Pre-trained Transformer 2

KNN : K-Nearest Neighbors

LDA : Linear Discriminant Analysis

LightGBM : Light Gradient Boosting Machine

LIME : Local Interpretable Model-Agnostic Explanations

LLM : Large Language Model

LLaMA : Large Language Model Meta AI

LR : Logistic Regression

LSTM : Long Short-Term Memory

MAE : Mean Absolute Error

MLP : Multi-Layer Perceptron

NB : Naive Bayes

NER : Named Entity Recognition

NLP : Natural Language Processing

PCA : Principal Component Analysis

PEFT : Parameter Efficient Fine-Tuning

QLoRA : Quantized Low-Rank Adaptation

QDA : Quadratic Discriminant Analysis

QSAR : Quantitative Structure-Activity Relationship

RF : Random Forest

R : Correlation Coefficient

RNN : Recurrent Neural Network

RMSE : Root Mean Square Error

RoBERTa : Robustly Optimized BERT Pretraining Approach

RR : Ridge Regressor

SHAP : SHapley Additive exPlanations

SMILES : Simplified Molecular Input Line Entry System

SMOTE : Synthetic Minority Over-sampling Technique

SVM : Support Vector Machine

t-SNE : t-Distributed Stochastic Neighbor Embedding

TF-IDF : Term Frequency-Inverse Document Frequency

XGBoost : eXtreme Gradient Boosting

List of Publications

0.1 Peer-Reviewed Journals

1. **Mansi Goel**, A. Amawate, A. Singh and G. Bagler, "ToxinPredictor: Computational Models To Predict the Toxicity of Molecules," Chemosphere, 2024.

2. **Mansi Goel**, N. Grover, D. Batra, N. Garg, R. Tuwani, A. Sethupathy, and G. Bagler, "FlavorDB2: An Updated Database of Flavor Molecules," Journal of Food Science, 2024.

3. **Mansi Goel**, V. Nathavani, S. Dharaiya, V. Kothadia, S. Srivastava, and G. Bagler, "Cultural Context Shapes The Carbon Footprints Of Recipes," International Journal of Gastronomy and Food Science, 2024.

4. G. Bagler and **Mansi Goel**, "Computational gastronomy: capturing culinary creativity by making food computable," npj Systems Biology and Applications, 2024.

5. **Mansi Goel**, A. Sharma, A. Chilwal, S. Kumari, A. Kumar, and G. Bagler, "Machine learning models to predict sweetness of molecules," Computers in Biology and Medicine (CBM), 2022.

6. **Mansi Goel** and G. Bagler, "Computational gastronomy: A data science approach to food," Journal of Biosciences, 2022.

0.2 Conferences

1. **Mansi Goel**, A. Agrawal, S. Agrawal, J. Kapuriya, A. Vamshi, R. Gupta, S. Rastogi, Niharika, and G. Bagler, "Deep Learning Based Named Entity Recognition Models for Recipes," Proceedings of Language Resources and Evaluation Conference (LREC-COLING), 2024.

2. **Mansi Goel**, S. Dargar, S. Ghatak, N. Verma, P. Chauhan, A. Gupta, N. Vishnumolakala, H. Amuru, E. Gambhir, R. Chhajed, M. Jain, A. Jain, S. Garg and G. Bagler, "Dish Detection in Indian Food Platters: A Computational Framework for Diet Management," 8th International Conference on Computer Vision and Image Processing (CVIP), 2023.

3. **Mansi Goel**, P. Chakraborty, V. Ponnaganti, M. Khan, S. Tatipamala, A. Saini, and G. Bagler, "Ratatouille: A tool for Novel Recipe Generation," IEEE 38th International Conference on Data Engineering Workshops (ICDEW), 2022.

4. D. Pandey, P. Parmar, G. Toshniwal, **Mansi Goel**, V. Agrawal, S. Dhiman, L. Gupta, and G. Bagler, "Object Detection in Indian Food Platters using Transfer Learning with

YOLOv4," 2022 IEEE 38th International Conference on Data Engineering Workshops (ICDEW), 2022. **(co-first author)**

0.3 Under Review

1. **Mansi Goel**, S. Bhagat, S. Srivastava, M. Patel, H. Parikh, S. Mehroliya and G. Bagler, "A framework for recipe data structure with applications for culinary and nutritional insights," 2025.

2. **Mansi Goel**, D. Sammi, D. Chaudhary and G. Bagler, "RecipeOnt: A Comprehensive Recipe Ontology for Culinary Knowledge Integration," 2025.

3. **Mansi Goel**, S. Mehta, A. Gupta, G. Dey and G. Bagler, "RatatouilleGen: Cuisine-Constrained Novel Recipe Generation," 2025.

4. **Mansi Goel**, R. Ramachandran, A. Tibrewal, S. Gupta, G. Panda, S. Agrawal, R. Sinha and G. Bagler, "Mining Culinary Patterns to Differentiate Global Cuisines using Deep Learning Models," 2025.

5. **Mansi Goel**, P. Singh, D. Garg, A. Bhargav and G. Bagler, "UmamiPredict: A unified Machine Learning Model," 2025.

6. S. Lakra, R. Oberoi, **Mansi Goel**, R. Singh, A. Roy, S. Jha, R. Singh and G. Bagler, "BIO-NER: A Deep Learning Approach for Named Entity Recognition in Recipes using Begin-Inside-Outside Encoding," 2025. **(co-first author)**

7. H. Sharma, **Mansi Goel**, D. Sahu, M. Sayed, P. Mangla, P. Shekhawat, S. Yadav, and G. Bagler, "AllerStack- Computational Stack Model To Predict the Allergenicity of Proteins," 2025. **(contributing author)**

Chapter 1

Introduction to Computational Gastronomy

1.1 Thesis Overview

Gastronomy has primarily been considered an artistic endeavor despite many efforts made to understand its scientific basis [1]. Over millennia, culinary traditions have led to rich legacies that define cultural identities and reflect the evolution of food processing, cooking techniques, and ingredient combinations [2]. Similar to languages that embody linguistic histories, cuisines capture the culinary legacies of cultures. Understanding the nuances of cuisines, food, and cooking enables asking many interesting questions. Why do we eat what we eat? Can we leverage the rich culinary knowledge to create a rule-based understanding beyond the artistic outlook towards cooking?

The advent of artificial intelligence and data-driven approaches has enabled machines to excel in board games, literature, art, and music. Could algorithmic approaches be extended to gastronomy? Imagine an AI-generated recipe winning the MasterChef show! While it may seem preposterous, algorithmic protocols that mimic cognitive and sensory processes may soon embrace cooking as reality.

Traditionally, aspects such as cultural influences and health associations have been studied qualitatively. However, the increasing availability of structured data and the advent of computational methods are dramatically changing the outlook toward gastronomy. The application of data-driven strategies [3, 4] for investigating gastronomic questions has opened up an all-new paradigm for the study of food and cooking. Computational Gastronomy is a data science that blends food, data, and computation towards achieving data-driven food innovations [5]. A data and computing-centric approach will enable the transformation of the food landscape toward achieving better public health, nutrition, and sustainability. Just as digitization has led to breakthroughs in imaging, text processing, and automated content creation, formalizing food as structured data can unlock new possibilities in culinary science. The development of computational models for recipes, ingredients, and molecular flavor composition can drive advancements in multiple areas. When combined with robotics and automation, such frameworks could enable intelligent cooking assistants, personalized diet recommendations, and even fully automated meal preparation systems. Understanding the molecular basis of taste and odor could lead to flavor printers capable of generating custom molecular concoctions that mimic specific sensory profiles. AI-driven nutritional analysis could power digital diet coaches, recommending personalized meal plans based on health metrics and dietary goals. On a broader scale, computational

gastronomy can facilitate the development of net-zero recipes—recipes optimized for minimal environmental impact by leveraging ingredients' carbon footprints.

Beyond its scientific and technological potential, computational gastronomy carries profound implications for public health and sustainability (see Figure 1.1). Food is not just a source of nourishment; it is a critical determinant of health outcomes and environmental sustainability. The increasing demand for sustainable food solutions necessitates innovative computational approaches to optimize ingredient sourcing, minimize food waste, and promote environmentally friendly dietary habits.

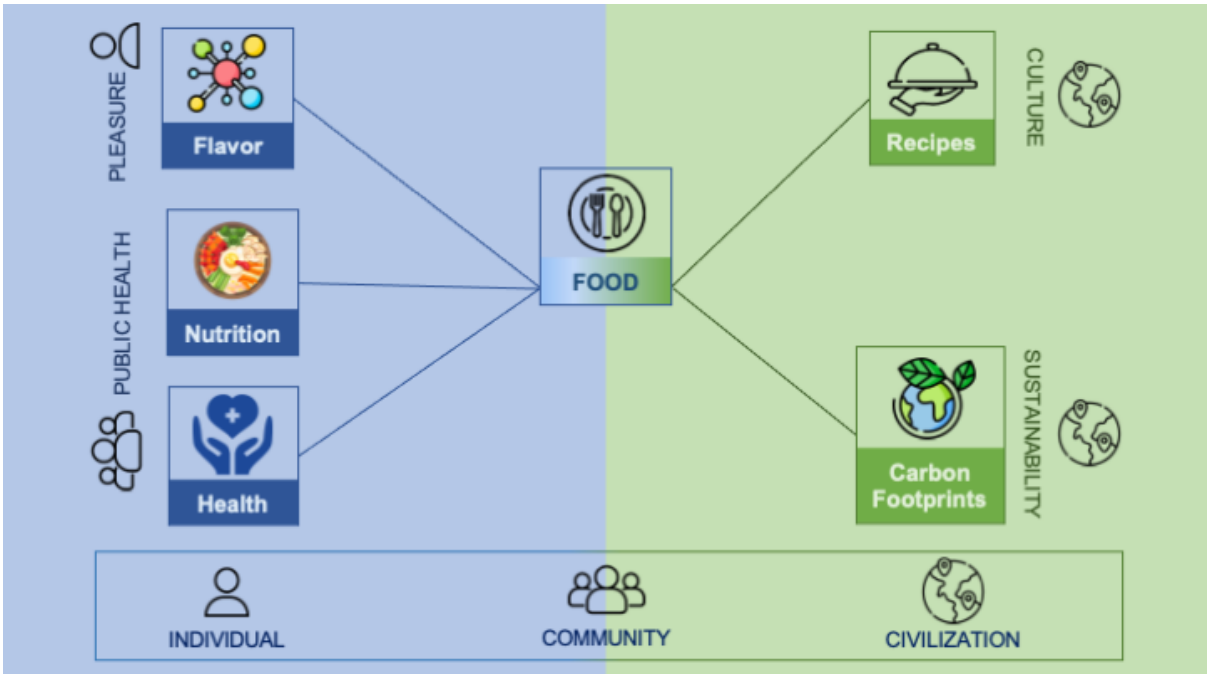

Figure 1.1: Capturing culinary creativity with Computational Gastronomy blending recipes, flavor, nutrition, health, and sustainability.

1.2 Challenges and Motivation

There has been increasing interest in this niche from academia and industry to ask deep gastronomic questions, harvest culinary data, and build algorithms. Despite its vast potential, computational gastronomy faces significant challenges, primarily in the structured compilations of rich data with cultural, taste, nutritional, health, and sustainability correlates of food. The diversity of food ingredients across geography, variation in their nutritional and flavor profiles, and regional carbon footprints add complexity to the problem. The availability of high-quality data and techniques for their analysis are among other challenges hindering the growth of this niche.

While databases such as RecipeDB [6] and FlavorDB [7] provide structured information, they

lack granularity in ingredient properties, molecular concentrations, and regional variations. Extracting meaningful features from unstructured recipe texts [8] remains challenging due to variations in ingredient nomenclature, cooking instructions, and cultural diversity. Quantifying the carbon footprint of recipes and promoting environmentally friendly dietary choices require integrating diverse datasets and predictive models. Food perception involves multiple modalities, including taste, smell, and vision. Computational models must integrate these diverse data sources to predict taste and classify dishes accurately.

Despite these challenges, the potential of computational gastronomy to shape the future of food systems is immense. With advancements in data collection, machine learning, natural language processing, and computer vision, the integration of computational methods can revolutionize gastronomy, from AI-assisted recipe generation to personalized nutrition and sustainable food systems.

1.3 Thesis Objectives

This thesis explores the landscape of computational gastronomy, combining culinary arts with advanced data science methodologies to uncover new opportunities for food research and application (see Figure 1.2). The journey commences with the development of RecipeDB2, a comprehensive and structured framework of recipes, ingredients, and their nutritional profile. A key aspect of this work is the creation of a recipe ontology, which organizes culinary knowledge and makes it more accessible for computational tasks. Using natural language processing techniques such as named entity recognition, the thesis extracts relevant entities from unstructured data to create a structured dataset. This technological advancement significantly enhances the database's functionality and lays the groundwork for innovative applications.

After creating a structured dataset, this thesis further delves into generating novel recipes using large language models, showcasing the creative potential of artificial intelligence in culinary innovation. This thesis investigates machine learning applications such as cuisine classification and estimating the carbon footprint of recipes. The environmental considerations promote sustainable dietary choices and connect gastronomy with ecological awareness, reflecting the growing importance of sustainability in food systems.

Apart from culinary science, this thesis delves into molecular informatics by introducing FlavorDB2, a dataset that documents the molecular composition and flavor profiles of various ingredients. By understanding the chemical properties of flavors, this research deepens the understanding of taste and its applications in culinary arts. In this context, the thesis explores the use of machine learning and deep learning models to predict the taste and toxicity of molecules. Lastly, the thesis investigates the practical application of computer vision techniques for dish detection in Indian food platters by implementing deep learning models to recognize ingredients

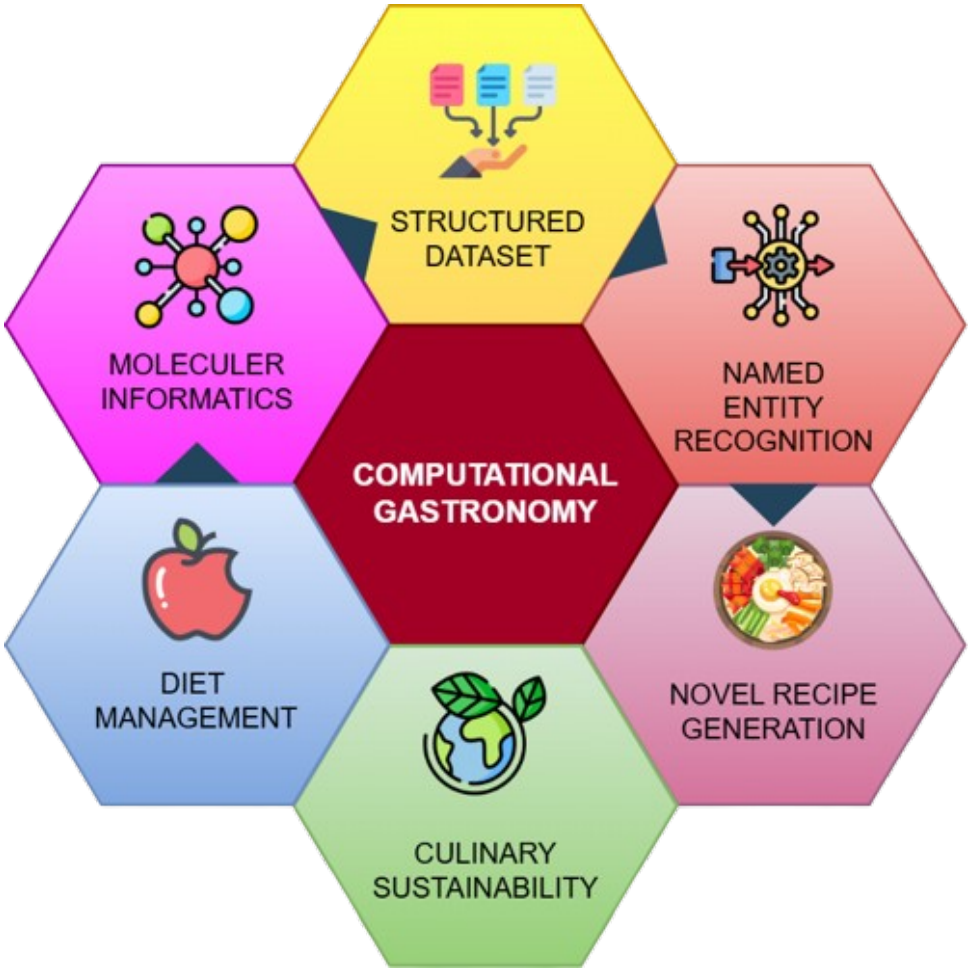

Figure 1.2: Computational Gastronomy is a data science that blends food with data and the power of computation for achieving data-driven food innovations.

in food images. This work demonstrates how computational techniques can be employed in real-world culinary scenarios, such as automated dietary tracking, personalized nutrition, and food recognition for culinary services.

1.4 Thesis Organization

The thesis is divided into the following twelve chapters.

- **Chapter 1:** The first chapter introduces computational gastronomy and highlights the importance of studying recipes as structured data. It outlines the challenges, motivation, thesis objectives, and organization.

- **Chapter 2:** Provides a comprehensive review of existing computational gastronomy literature, covering significant theories, methodologies, and findings. Research gaps and the potential impact of computational approaches are discussed.

- **Chapter 3:** Introduces RecipeDB2, a framework for structuring recipe data. RecipeDB2

transforms unstructured recipes into a machine-readable format, making them scalable and searchable for computational tasks.

- **Chapter 4:** Presents a recipe ontology integrating recipes and flavors.

- **Chapter 5:** This chapter outlines the methodologies for extracting structured information from unstructured recipe texts. It discusses model implementation for identifying ingredient names, quantities, units, and cooking methods, converting raw text into structured data.

- **Chapter 6:** Explores algorithms for generating novel recipes, including generic and cuisine-specific approaches. The generated recipes are evaluated using the Turing Test for Chefs, an assessment framework for machine-generated recipes.

- **Chapter 7:** Examines the sustainability aspect of food systems by evaluating the carbon footprint of recipes. Methodologies for estimating the environmental impact of recipes based on their ingredients are discussed.

- **Chapter 8:** Identifies culinary patterns in classifying cuisines across the world.

- **Chapter 9:** Introduces FlavorDB2, a dataset mapping ingredients to their flavor molecules.

- **Chapter 10:** Focuses on molecular informatics in predicting the taste and toxicity of molecules. Models are presented to predict the taste (sweet and umami) and toxicity of flavor molecules.

- **Chapter 11:** Showcases the application of computer vision techniques to detect and recognize ingredients in Indian food platters. The challenges of identifying dishes across varying class complexities are addressed using deep learning models.

- **Chapter 12:** The final chapter summarizes the contributions of the thesis, discusses the impact of computational gastronomy, and outlines future research directions.

Chapter 2

Literature Survey

2.1 Gastronomy Datasets

Probing food from a data-driven perspective requires the availability of well-curated, structured data resources. Traditional recipes encode the cultural wisdom that has gone into making a delicious dish. While there is a plethora of websites (such as allrecipes.com, geniuskitchen.com, epicurious.com, foodnetwork.com, and tarladalal.com) that provide a compilation of human-readable recipes, there has been a dearth of a structured compilation of these recipes which enables complex queries. Addressing this gap necessitates natural language processing (NLP) algorithms that capture relevant information (quantity, unit, temperature, processing state, etc.) from the recipe text. Diwan et al. [8] designed named entity recognition (NER) models for extracting such details using knowledge mining techniques. These models have applications when translating recipes between languages, determining similarities between recipes, generating novel recipes, and estimating the nutritional profile of recipes.

RecipeDB (https://cosylab.iiitd.edu.in/recipedb) has been created to investigate the culinary correlates of dietary elements for probing their association with sensory responses as well as consequences for nutrition and health [6]. It is a structured compilation of recipes, ingredients, and nutrition profiles interlinked with flavor profiles and health associations. This repertoire comprises of more than 118,000 recipes from cuisines across the globe (6 continents, 26 geo-cultural regions, and 74 countries), cooked using various processes (heat, cook, boil, simmer, bake, etc.), by blending thousands of ingredients. Ingredients are further linked to their flavor molecules, nutritional profiles, and empirical records of disease associations obtained from Medline.

Ingredients are chosen to be used in recipes, primarily by virtue of their taste and odor (together referred to as flavor). Experimental assays such as Gas Chromatography and Mass Spectroscopy probe the constituent flavor molecules present in ingredients. A structured compilation of flavor compounds from natural ingredients is a prerequisite for quantification of the taste of ingredient combinations and any subsequent investigations. FooDB (http://foodb.ca), one of the resources that addressed this need, compiled molecules from food ingredients, albeit its focus was not on the chemical basis of flavor. Another resource, Flavornet, provides a list of flavor molecules and their odor profiles but does not provide information on their natural sources [9]. Some other attempts in this direction have focused on the flavor-specific compilation of data, such as bitter (BitterDB) and sweet (SuperSweet), and volatile compounds of scents (SuperScent) [10, 11, 12].

Some other efforts have targeted nutritional factors (NutriChem), polyphenols (Phenol-Explorer), and the medicinal value of food [13, 14, 15, 16].

FlavorDB (https://cosylab.iiitd.edu.in/flavordb) is a comprehensive repository of flavor compounds, physicochemical structures, their natural sources, flavor percepts, and functional groups [17]. It helps in finding molecules of desired flavor percepts, exploring ingredient molecules, discovering relevant food blendings, and knowing the chemical properties of flavor compounds. FlavorDB contains 25,595 flavor molecules, and 34 categories cover 936 ingredients linked to 527 unique natural resources. The resource is conceptually divided into entity space, representing natural ingredients utilized in food, and flavor space, representing flavor molecules and their chemical properties. Each molecule is described with descriptors such as PubChem ID, CAS number, IUPAC, SMILES, and 2D/3D properties. In connection with the flavor percepts, it lists 33 taste and 1,068 odor receptors with unique UniProt IDs and names. Users can search FlavorDB via the entity (ingredient) name, category, flavor molecules, and their precepts.

2.2 Food Pairing

In a reductionist manner, a recipe can be perceived as a combination of ingredients. With this notion, the theoretical number of recipes that can be generated is astronomically large. Even with a conservative estimate, one gets 10^{30} recipes with an average recipe size of 10 and 1,000 available ingredients. Traditional recipes don't represent all possible combinations of ingredients due to bias in preferences, climate, geography, cultural lock-ins, religious taboos, and, to an extent, even genetics. A fundamental question in gastronomy is whether ingredient combinations follow systematic principles. Chef Heston Blumenthal's food pairing hypothesis suggests that ingredients with shared flavor molecules blend well together [18]. The availability of data on ingredients used in the recipes and flavor profiles of natural ingredients facilitates the investigation of the food pairing hypothesis in a quantifiable manner.

In one of the earliest computational gastronomy investigations, Ahn et al. probed the data of 56,498 recipes as a bipartite network of 381 ingredients used in the recipes from a range of world cuisines and their 1,021 flavor molecules [19, 20]. Here, ingredients and flavor molecules are nodes, and an edge represents the association between the two. They compared the average food pairing of recipes in cuisine with that of a random cuisine (a random control created by generating recipes by an arbitrary combination of ingredients while maintaining the recipe size distribution). In the absence of a statistically significant difference between the food pairing index of a real cuisine vis-à-vis its random counterpart, one may infer that the prior does not differ from the latter regarding preferential use of ingredient combinations. A positive deviation in food pairing would indicate a uniform blend of ingredients, confirming the food pairing

hypothesis. On the contrary, a negative deviation would be a signature of a contrasting blend of ingredient combinations. Ahn et al. [19] observed that cuisines from North America, Western Europe, Southern Europe, and Latin America show uniform (positive) food pairing, confirming Chef Blumenthal's food pairing hypothesis. East Asian cuisine (recipes from South Korea), on the other hand, showed the signature of a contrasting blend of flavor profiles. In another study that focused on recipes from the Indian subcontinent, authors investigated the food pairing phenomenon. They demonstrated a strong role of spices in defining ingredient interactions, contributing to a negative food pairing signature. [21, 22]. Further studies extended the food pairing framework with the data of Arabian [23] and Medieval European [24] cuisine and observed a positive food pairing.

The food bridging hypothesis [25] introduced the idea of the indirect molecular connections between ingredients, particularly relevant to Eastern cuisines. Large-scale analyses probing 45,772 recipes from 22 regions worldwide, [26] showed the ubiquitous nature of positive food pairing, with exceptions in regions like Japan and Korea. One of the relevant topics in the context of food pairing is food-beverage pairing. Charles Spence proposed perceptual (sense-based) and cognitive/intellectual (mind-based) pairing principles [27]. While the former involves food-beverage pairing based on similar aroma, taste, color, and flavor, the latter involves pairing based on cultural and geographical identity. Perceptual pairing has become quite popular lately vis-à-vis cognitive pairing. Another question around food pairing is whether one can predict the new pairings. Park et al. [28] proposed a KitcheNette model leveraging the Siamese neural network to determine the food-ingredient pairings and predict the unknown pairings using MIT's Recipe1M database. Beyond food pairing, researchers have explored recipe transformation across cuisines [29]. In this study, the authors utilized the Yummly dataset comprising 39,774 recipes from 20 countries and applied neural networks to recommend ingredient substitutions, thereby enabling cross-cultural adaptations. The model learned distinctive ingredient usage patterns associated with various regional cuisines and represented ingredients as vectors in an embedding space. This representation captured contextual similarities between ingredients, allowing the system to identify culturally specific components and propose appropriate substitutes typical of the target cuisine. As a result, the approach facilitated culturally coherent recipe transformations while maintaining the original recipe structure.

2.3 Health and Nutrition

Beyond the flavor, nutrition, and health impacts of food ingredients are among the major aspects. Food interaction with the human body is complex, influenced by biochemical, genetic, and environmental factors, making it challenging to predict precise health outcomes. Computational approaches, including data-driven analysis, provide a pathway to investigate food-health relationships systematically. Several studies have explored the association between food in-

gredients and health outcomes through data mining and machine learning techniques. Rakhi et al. [30, 31] developed SpiceRx (https://cosylab.iiitd.edu.in/spicerx), a repository that maps spice-disease associations by extracting information from the MEDLINE database.. This study utilized named entity recognition and machine learning techniques to identify relationships between 188 culinary herbs and spices and 8,957 spice-disease associations. Their findings suggested a strong positive association of spices with conditions such as diabetes and inflammation, while some spices exhibited adverse associations. The Convolution Neural Network (CNN) classifier presented with an accuracy of 86.7% and precision of 90.7%. The repository integrates spice-phytochemicals association using Phenol Explorer [13] and KNapSAcK [32] and association with diseases using CTD (Comparative Toxicogenomic Database) [33]. SpiceRx provides a platform to investigate tripartite associations between spices, their phytochemicals, and associated health effects. Extending the scope of SpiceRx, Tuwani et al. have built a repository DietRx (https://cosylab.iiitd.edu.in/dietrx) that integrates the associations among food, diseases, chemicals, and genes. DietRx represents 1,781 food entities obtained by text mining around 38,000 MEDLINE articles using named entity recognition techniques. The data is further enriched with 6,992 food chemicals and 20,550 food-genes associations, thereby providing a platform for investigating dietary ingredients and their health consequences.

Data from social networks is another dimension that can be explored to associate food with diseases. Abbar et al. [34] analyzed food consumption through Twitter, correlating them with obesity and diabetes prevalence across U.S. citizens. By mining 502 million tweets and comparing them with CDC obesity and diabetes data, their study demonstrated how digital footprints can provide population-scale dietary insights. In another study that probed the food-nutrition axis, Sajadmanesh et al. [35] analyzed 157,000 recipes to explore correlations between average nutrient values in country-specific cuisines and national health indicators. By linking cuisine-based nutritional profiles with obesity, diabetes, and health expenditure statistics, they found that sugar and carbohydrates are positively associated with obesity and diabetes, while protein intake is negatively associated. These associations suggest potential impacts of dietary patterns on population-level health outcomes, particularly highlighting the inverse relationship between high-protein diets and obesity prevalence. Beyond individual health, food sustainability is gaining prominence in nutritional research. Van Erp et al. [36] proposed a framework linking nutrition and sustainability using recipe and nutrition databases. Their work introduced a NutriScore scale to evaluate the nutritional quality of recipes and a recommendation system to optimize food choices for health, environmental impact, and supply chain efficiency. Further, knowledge graphs have been leveraged to enhance ingredient substitutions for healthier diets [37]. An ingredient substitution heuristic rooted in the ingredients' semantics was developed to identify substitutions. This approach was evaluated using web-scraped databases, and it outperformed state-of-the-art methods by suggesting healthy substitutions.

2.4 Taste Prediction

Flavor molecules are characterized by taste and odor percepts. While there are five broad taste attributes, namely sweet, bitter, salty, sour, and umami, the number of odor percepts is much larger [17]. Prediction of taste and odor based on molecular attributes is a key challenge in computational gastronomy [38]. Among these, predicting taste has been largely focused on the bitter and sweet attributes, as they hold significant relevance in food science and health. Several advancements have been made in the field of taste prediction, particularly in understanding the molecular basis of bitter and sweet tastes [39, 12, 40, 41, 42]. Significant work was conducted by Tuwani et al. [43], who built state-of-the-art machine learning algorithms to predict bitter-sweet taste profiles. Their study curated a diverse dataset, including 918 bitter, 1510 non-bitter, 1205 sweet, and 1171 non-sweet molecules. Molecular descriptors were generated using cheminformatics tools, including ChemoPy, Dragon 2D, Dragon 2D/3D, Canvas, and ECFPs, which served as features for training models. Feature selection was done using the Boruta algorithm, while dimensionality reduction was achieved using Principal Component Analysis (PCA). The dataset was subjected to 5-fold stratified cross-validation to evaluate the performance of models trained using Random Forest, Ridge Logistic Regression, and Adaboost algorithms. The models were evaluated using precision, recall, AuPR, AuROC, F1, and sensitivity score metrics. The results revealed that Adaboost and Random Forest, in combination with Dragon 2D/3D molecular descriptors and the Boruta algorithm, yielded the most accurate predictions for sweet/non-sweet taste classification, while Random Forest with PCA performed well for bitter/non-bitter predictions. Notably, the study concluded that average precision was a more reliable performance metric, achieving a precision of 90%. A user-friendly web server, BitterSweet (https://cosylab.iiitd.edu.in/bittersweet) has been developed, which allows users to predict the taste profiles of small molecules based on their IUPAC, SMILES, or common names. The database comprises over 394,000 molecules with predicted BitterSweet taste profiles and 3,086 with verified taste profiles, making it a valuable resource for researchers and practitioners.

2.5 Cuisine Classification

Efforts have been made to probe the interrelatedness of cuisines by virtue of their similarity. In one of the studies, Sajadmanesh et al. 2017 [35], using the data of 157,000 recipes from the World Bank, BBC, and Yummly websites, the authors obtained Jensen-Shannon divergence to find the ingredient-based similarity between the cuisine. A flavor-based correlation matrix was formed between recipes to find a similar cuisine based on flavor molecules. The results showed that Welsh cuisine was dominated by Asian culture, whereas Indian cuisine was found to have similarities with Ethiopian and African cuisines with the dominance of spices.

In another study, Sharma et al. [44] curated 118,071 recipes and identified the most significant patterns in 26 geo-cultural cuisines worldwide using frequent itemset mining and ingredient authenticity. To find the similarity between cuisines, the authors implemented hierarchical agglomerative clustering using three distance metrics (Jaccard, Cosine, and Euclidean). Euclidean distance-based clustering gave good results when validated using geographical distance among cuisines. Some clusters (India and Africa, France and Canada) showed similarities in their cuisines despite being geographically separated.

2.6 Novel Recipe Generation

Recent advances in the field of computation and dataset availability have significantly contributed to the development of novel recipe generation techniques. Researchers have explored several techniques, including rule-based systems, deep learning architectures, and multimodal methodologies, to enhance automated recipe creation. A rule-based generation algorithm has been implemented, EPICURE [45], generated recipes based on predefined templates. More recent efforts have leveraged large-scale datasets and machine-learning models to improve the generation process.

An early study in novel recipe generation introduced the 'neural checklist model' [46], a text generation framework to generate recipes from a given list of ingredients. This model employed an encoder-decoder architecture based on a recurrent neural network with an attention mechanism to track the utilization of ingredients. The performance was assessed by comparing with four baseline models followed by human evaluations. While it achieved an average human rating of 4.2 on a scale of 1 to 5, its effectiveness was highly dependent on fine-tuning hyperparameters. Transformer-based architectures have also been widely adopted for recipe generation. RecipeGPT [47], a model fine-tuned on MIT's Recipe1M dataset, generates recipe instructions based on a given title and list of ingredients or predicts missing ingredients based on a title and instructions. RecipeGPT's evaluation involved comparing generated recipes with reference recipes, focusing on ingredient overlap and structural coherence.

Another dimension of this domain is to generate recipes from food images. Salvador et al. [48] proposed an 'inverse cooking system' that generates recipes by predicting ingredient lists from food images, which are then used to generate step-by-step cooking instructions. The study demonstrated that image-to-ingredients prediction was more effective as compared to image-to-recipe retrieval. By comparing their model against two retrieval-based approaches, the authors showed that their model outperforms both human and retrieval baselines. However, the study also highlighted key limitations, including difficulties in accurately recognizing ingredients, weak correlations between predicted ingredients and generated instructions, and redundancy in ingredient enumeration. Building on this, Wang et al. [49] proposed Decomposed Generation

Network model to improve the generation of cooking instructions from food images. The instructions were split into phases based on the cooking process and a sub-generator was assigned, which was concatenated to generate the recipe.

Personalized recipe generation [50] has also been explored as an extension of text-based and multimodal models. Yu et al. [51] proposed a Routing Enforced Generative Model, which categorized ingredients into dish types (e.g., low sugar, high fiber, grilling) and generated recipes accordingly. The model's effectiveness was evaluated through human assessments of readability, accuracy, feasibility, creativity, and overall quality. Further research focuses on adaptive recipe generation [52] using unsupervised methods based on similarity, preference, and dietary restrictions. Their approach leveraged word embedding models such as word2vec to identify ingredient substitutions. However, the study did not implement advanced neural language models like BERT or GPT, highlighting a potential avenue for future improvements.

Multimodal methodologies have been developed to address challenges in recipe generation by integrating various data sources. Cook-Gen [53] generated cooking actions from recipes to overcome issues with irregular data patterns. FIRE [54] methodology for recipe generation to create the food title, ingredients list, and cooking instructions by taking food images as input. Another study [55] focused on enhancing user culinary experiences by leveraging contextual and relational information to rank plausible substitutions to showcase the potential for personalized cooking. In contrast with traditional monolithic approaches, a more agile method [56] employed image models and multiple data points to address the limitations of existing multimodal models. Along with these data-driven approaches, the generative grammar of cooking ponders on the underlying culinary rules and grammar that dictate the art of cooking [57].

Chapter 3

RecipeDB2: A Unified Framework for Recipe Data Structure

3.1 Introduction

Cooking is the art of transforming raw ingredients into flavorful and nutritious dishes—a skill passed down through generations via recipes [58, 2]. These recipes encode essential knowledge about processing natural ingredients, making them palatable and nutritionally beneficial. Over time, distinct cooking styles, referred to as cuisines, have emerged, conforming to geo-cultural constraints. Despite technological advancements and evolving eating habits, daily dietary intake remains deeply rooted in cultural traditions [59]. Cooking has been pivotal in the evolution of large brain sizes in *Homo sapiens* [60] and plays a crucial role in shaping the gut microbiome [61]. Thus, recipes serve as a bridge between our taste preferences and health outcomes.

As part of the Computational Gastronomy paradigm, building a structured and comprehensive repository of annotated recipes is essential to decode the complex interplay between taste, nutrition, and health in recipes [62, 5]. Traditionally, recipes have been passed down orally and, more recently, as written records [63]. However, these formats remain unstructured and unannotated, posing challenges for computational analysis. In recent years, various datasets have been developed in the context of recipes such as Recipe1M [64], Epicurious(https://www.epicurious.com/), Yummly (https://www.yummly.com/), and RecipeNLG [65]. Recipe1M is a large dataset containing over one million recipes sourced from various online platforms. However, its unstructured format comprising recipes, ingredients, and images often lacks critical metadata, including cooking techniques, preparation methods, and nutritional information. Epicurious dataset offers a more structured format with well-defined ingredient lists and cooking instructions. It includes user ratings, preparation times, and seasonal categories but lacks nutritional data. Similarly, Yummly serves as a recipe discovery platform with a reasonably structured dataset that features ingredients, cooking steps, and dietary tags, yet it does not encompass comprehensive nutritional information. RecipeNLG, designed primarily for natural language generation tasks, provides structured recipes but lacks a broader nutritional or cultural context.

Other widely-used datasets have gained prominence in recent studies. The Food.com dataset comprises over 400,000 user-contributed recipes along with reviews, ratings, and user interactions, making it a valuable resource for studying user preferences and recommendation systems. Similarly, MealRec is a large-scale dataset designed to support food recommendation and meal planning, comprising over one million meal sessions annotated with contextual features such as time, user behavior, and co-consumed recipes. These datasets highlight the growing diversity

and reinforce the need for structured, annotated repositories to advance research in this interdisciplinary field. RecipeDB [6] is a structured dataset that addresses most of these shortcomings, but it has a limited size and low mapping efficiency between ingredients and their nutritional profiles. The Named Entity Recognition (NER) performance in RecipeDB was low, which affected the prediction of named entities and the further mapping to their nutritional correlates.

We developed a framework for recipe data structure and built RecipeDB2 to address these challenges by implementing an enhanced NER model and building a larger, scalable repository of recipes from across the globe. It builds upon the foundation of RecipeDB [6], to offer more detailed annotations of culinary attributes such as geo-cultural cuisine, dietary style, cooking techniques, utensils, ingredient details, and nutritional profiles. This repository makes recipes computable, opening up new avenues for data-driven exploration of global cuisines by accounting for their culinary nuances.

Among other data resources related to the objectives of RecipeDB, FooDB focuses on the compilation of food chemicals (http://foodb.ca). FoodBase [66] provides an annotated food entity resource and has been extended to include corpora such as CafeteriaSA and CafeteriaFCD, supporting sentiment analysis and food composition analysis. Furthermore, recent developments in the domain leverage NLP workflows to extract relations between food and biomedical entities—examples include FoodChem [67] and FooDis [68], which facilitate the construction of domain-specific food knowledge graphs. Notably, Cenikj et al. [69] demonstrate the integration of large language models to construct large-scale food and biomedical knowledge graphs, illustrating the potential of advanced NLP methods in food informatics. Other knowledge graph efforts, such as FoodKG[70], further exemplify semantics-driven approaches for food recommendation and reasoning. Some other databases, such as FlavorDB [7], FlavorDB2 [71], BitterDB [72], and SuperSweet [73], focused on taste and olfaction, attempting to address the interaction of natural entities with human sensory machinery. Databases such as NutriChem [74](nutritional factors), and DietRx emphasized the food-nutrition-health axis.

RecipeDB2 is an expanded and annotated resource designed to probe the relationships between culinary practices and nutrition (Figure 3.1). Recipes are broken down into their culinary elements using advanced algorithms, creating a searchable database that reflects geo-cultural contexts, dietary preferences (Vegan, Pescetarian, Lacto-vegetarian, Ovo-vegetarian, Ovo-lacto-vegetarian), cooking methods, and ingredient attributes. It provides a detailed breakdown of each ingredient, including name, quantity, unit, state, and additional characteristics. RecipeDB2 also integrates data from the USDA database, offering deeper nutritional insights with consequences for health. The recipe framework for recipe data structure presents RecipeDB2 as an illustration. This updated resource empowers researchers, nutritionists, chefs, and food enthusiasts to explore the complex relationships between food, culture, nutrition, and sensory experiences.

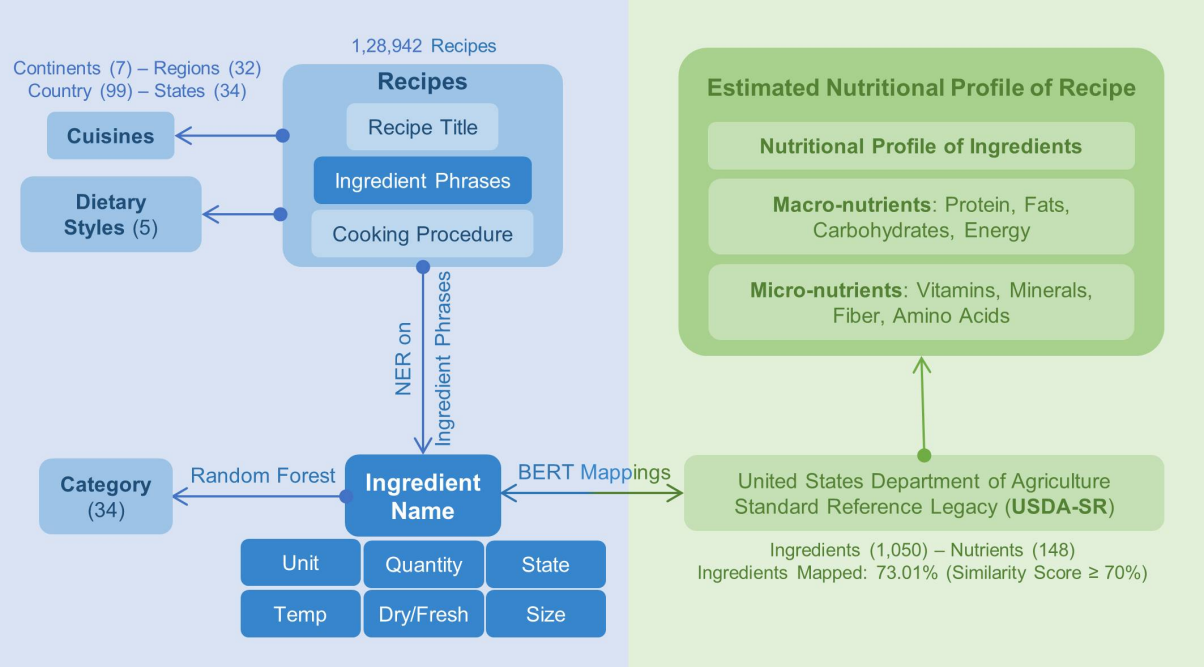

Figure 3.1: The flow diagram depicting the framework for recipe data structure involving data compilation, curation, annotations, mapping, and estimation of the nutritional profiles. Building this framework involved the application of multiple machine-learning models. A deep-learning-based named entity recognition model was used for extracting relevant culinary elements from the ingredient phrases. BERT was implemented to map ingredients from the recipes to those in the USDA nutrition table. A Random Forest model was used to predict the ingredient category. This framework for recipe data structure is generic and scalable.

3.2 Framework Overview

3.2.1 Data compilation

We created an extensive, structured database of recipes by appending the RecipeDB [6] (118,171 recipes) with those obtained from Archana's Kitchen (https://www.archanaskitchen.com/) (9,730 recipes) and Awesome Cuisine (https://www.awesomecuisine.com/) (1,132 recipes). These datasets were selected based on several critical criteria, including their structural uniformity, the availability of geo-cultural mapping for ingredients and dishes, and the substantial number of recipes. The dataset encompasses a wide variety of information about each recipe, including the recipe title, list of ingredients, servings, detailed cooking instructions, preparation time, cooking time, total time, and associated recipe images.

To facilitate detailed data extraction, we divided each recipe into two main parts: the ingredients section and the cooking instructions section. This division helps to isolate and analyze relevant components more effectively. Our goal was to extract structured elements and relevant culinary information from these sections, such as ingredient names, quantities, units, and preparation

steps, for further analysis. This structured data has applications in tasks ranging from nutritional estimation and recipe generation to the geo-cultural classification of dishes.

3.2.2 Geo-cultural mapping of recipes

Every recipe in the dataset was mapped to its geo-cultural correlate at different levels of hierarchy: Continent, Region, Sub-region (Country), Sub-sub-region (State, where applicable), and Sub-sub-sub-region (City, where applicable). This multi-level mapping provides a deeper understanding of each recipe's geographical and cultural context. The continent and country-level mapping of recipes was obtained from their original data sources. For the region-level mapping, we employed a more flexible approach, focusing on culinary and cultural similarities rather than strictly adhering to geo-political boundaries. With such an approach, one can capture shared culinary traditions and influence across political borders.

Countries were grouped into regions by virtue of their shared geo-cultural correlates, and regions were further attached with coarse-grained labels of continents. The sub-region, sub-sub-region, and sub-sub-sub-region levels provide a more granular view down to specific countries, states, or cities, where available, further refining the geo-cultural context of the recipes. The data comprised 7 continents, 32 regions, 99 sub-regions, 34 sub-sub-regions, and 12 sub-sub-sub-regions. Appendix A Table A.1 presents the schema of the continent and region mapping, providing clarity to geo-cultural mappings of recipes.

3.2.3 NER on ingredient section

We extracted 'named entities' from the ingredients section of each recipe to capture structured data essential for nutritional analysis. We used the state-of-the-art spaCy-transformer model [75] to extract the named entities [8]. Through this process, we identified seven critical attributes for providing relevant nutritional information about a recipe. These attributes allow us to standardize and quantify ingredients for further analysis. The attributes identified are as follows:

- Name: The ingredient's name, such as salt or pepper, provides the recipe's foundational component.
- Unit: The unit of measurement associated with the ingredient, such as gram, cup, tablespoon, or teaspoon, specifying how the ingredient's quantity is measured.
- Quantity: The specific amount of the ingredient, paired with the unit, for example, 1 teaspoon, 10 grams, or 2 cups, to provide a precise measurement for recipe scaling and nutritional estimation.
- State: The processing state or form of the ingredient, such as ground, chopped, thawed, or blanched, reflecting its preparation before use.

Ingredients	Frequency	Units	Unit Count
salt	53234	[‘cup’, ‘dash’, ‘g’, ‘ml’, ‘ounce’, ‘pinch’, ‘tablespoon’, ‘teaspoon’, ‘tsp’]	9
garlic	42789	[‘bulb’, ‘clove’, ‘cup’, ‘g’, ‘halved’, ‘ounce’, ‘pressed’, ‘tablespoon’, ‘teaspoon’, ’tsp’]	10
onion	37195	[‘cup’, ‘g’, ‘halved’, ‘kg’, ‘lb’, ‘ounce’, ‘pound’, ‘slice’, ‘tablespoon’, ‘teaspoon’]	10
water	27799	[‘can’, ‘cup’, ‘g’, ‘gallon’, ‘liter’, ‘ml’, ‘ounce’, ‘pint’, ‘quart’, ‘tablespoon’, ‘teaspoon’]	11
butter	26624	[‘cup’, ‘g’, ‘gram’, ‘lb’, ‘ml’, ‘ounce’, ‘pound’, ‘stick’, ‘tablespoon’, ‘tbsp’, ‘teaspoon’, ‘tsp’]	12
egg	22901	[‘cup’, ‘dozen’, ‘extra large’, ‘large’, ‘whole’]	5
sugar	22144	[‘cup’, ‘dash’, ‘g’, ‘gram’, ‘kg’, ‘lb’, ‘ml’, ‘ounce’, ‘pinch’, ‘tablespoon’, ‘tbsp’, ‘teaspoon’, ‘tsp’]	13
olive oil	21257	[‘cup’, ‘dash’, ‘ml’, ‘ounce’, ‘tablespoon’, ‘tbsp’, ‘teaspoon’]	7
tomato	18317	[‘can’, ‘cup’, ‘g’, ‘gram’, ‘halved’, ‘kg’, ‘lb’, ‘ounce’, ‘pound’, ‘quart’, ‘slice’, ‘tablespoon’]	12
black pepper	15743	[‘cup’, ‘dash’, ‘pinch’, ‘tablespoon’, ‘teaspoon’]	5
milk	15287	[‘can’, ‘cup’, ‘g’, ‘gallon’, ‘gram’, ‘liter’, ‘ml’, ‘ounce’, ‘pint’, ‘quart’, ‘tablespoon’, ‘teaspoon’]	12
lemon juice	12409	[‘cup’, ‘dash’, ‘ml’, ‘ounce’, ‘tablespoon’, ‘tbsp’, ‘teaspoon’, ‘tsp’]	8
pepper	12072	[‘corn’, ‘cup’, ‘dash’, ‘pinch’, ‘tablespoon’, ‘teaspoon’]	6
salt pepper	11019	[‘dash’, ‘pinch’, ‘teaspoon’]	3
flour	10779	[‘cup’, ‘g’, ‘kg’, ‘lb’, ‘ml’, ‘ounce’, ‘tablespoon’, ‘teaspoon’]	8
oil	10651	[‘cup’, ‘ml’, ‘ounce’, ‘quart’, ‘tablespoon’, ‘tbsp’, ‘teaspoon’, ‘tsp’]	8
ginger	10519	[‘cm’, ‘cup’, ‘dash’, ‘g’, ‘inch’, ‘ounce’, ‘piece’, ‘pinch’, ‘slice’, ‘tablespoon’, ‘teaspoon’, ‘tsp’]	12
carrot	9981	[‘cup’, ‘g’, ‘lb’, ‘ounce’, ‘pound’, ‘slice’, ‘tablespoon’]	7
parsley	9798	[‘bunch’, ‘cup’, ‘g’, ‘ounce’, ‘sprig’, ‘tablespoon’, ‘teaspoon’]	7
cinnamon	9257	[‘cup’, ‘dash’, ‘inch’, ‘piece’, ‘pinch’, ‘stick’, ‘tablespoon’, ‘teaspoon’]	8

Table 3.1: 20 most popular ingredients, their frequency, popularly used units, and unit counts.

- Size: Any portion size descriptors for the ingredient, such as small, medium, or large, which further refine the quantity and help standardize serving sizes in recipes.

- Temperature: The temperature condition of the ingredient before or during preparation, such as hot, cold, or room temperature, can affect the cooking process and the ingredient’s properties.

- Dry/Fresh: A classification that identifies whether the ingredient is dry (e.g., dried herbs, spices) or fresh (e.g., fresh basil, vegetables), which influences the flavor profile and shelf life of the ingredient.

3.2.4 Standard units conversion

Ingredients in recipes most often come with a variety of measurement units. For example, salt can be measured in tablespoons, teaspoons, pinches, dashes, packets, envelopes, etc. Similarly, other ingredients are paired with diverse units of measurement. RecipeDB2 data features an extensive collection of 35,474 unique ingredients and 1,163 distinct measurement units. Table 3.1 shows the 20 most popular ingredients and their unique units, where salt (53234; 41.28%), garlic (42789; 33.18%), onion (37195; 28.84%), water (27799; 21.55%), and butter (26624; 20.64%) were among the most popular ingredients. On the measurement side, Figure 3.2 shows the frequency rank distribution of units, commonly used units in recipes are cup (291,756), teaspoon (224,404), tablespoon (187,824), lb (417,70), and ounce (39,571). This wide variation in ingredients and measurement units highlights the complexity of recipe creation and cooking.

We identified 81,240 unique ingredient-unit pairs. Among these, 4,961 (6.1%) were already in standard units (grams for solids (3,932) and milliliters for liquids (1,029)), while 93.9% were in non-standard units. We aimed to convert all non-standard units into standard ones to ensure

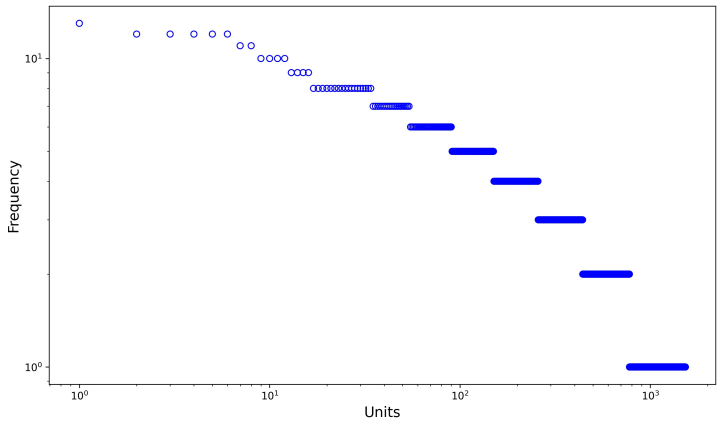

Figure 3.2: Frequency rank distribution of units. The data indicate the presence of a thick-tailed (scale-free) distribution, suggesting that few units are present in disproportionately high frequency.

consistency across recipes. We manually found the standard unit conversion of each unit for each ingredient. The ingredient-unit pairs were far too many for manual conversion, hence we sorted the pairs based on the frequency and converted the most frequent pairs into their standard units. We successfully converted 43.90% of the most frequently used ingredient-unit pairs into standard units. Figure 3.3 illustrates 50 commonly used ingredient-unit pairs in our dataset. Ingredients such as salt, black pepper, cumin, cinnamon, oregano are widely used in teaspoons, garlic with clove, beef in lb, butter, olive oil, and vegetable oil in tablespoons, water, milk, sugar, and flour in cups.

3.2.5 Integration of RecipeDB2 to USDA ingredient

To find the nutritional profile of ingredients, we need to accurately map ingredients from the RecipeDB2 dataset to their corresponding entries in the Standard Reference Legacy Release database (https://fdc.nal.usda.gov/) (United States Department of Agriculture; USDA). RecipeDB2 presents 128,942 recipes comprising 35,474 ingredients from 32 regions, while USDA has 1,050 unique ingredients. The disparity in the number of entities between these databases is due to the many-to-one mappings of RecipeDB ingredients to those in the USDA. A typical USDA ingredient occurs in recipes in various avatars.

We implemented three mapping strategies: Jaccard similarity [76], BERT-based [77, 78], and RoBERTa-based [79]. Each approach leverages different techniques for identifying the best matches between ingredients based on textual similarity and contextual understanding. The Jaccard similarity is a well-known metric that compares two sets by measuring the intersection over the union. We compared the ingredient names from RecipeDB2 and USDA by breaking them down into sets of tokens (e.g., “ground cumin” becomes ground, cumin). The Jaccard score was calculated by dividing the number of shared words (intersection) by the total number of unique

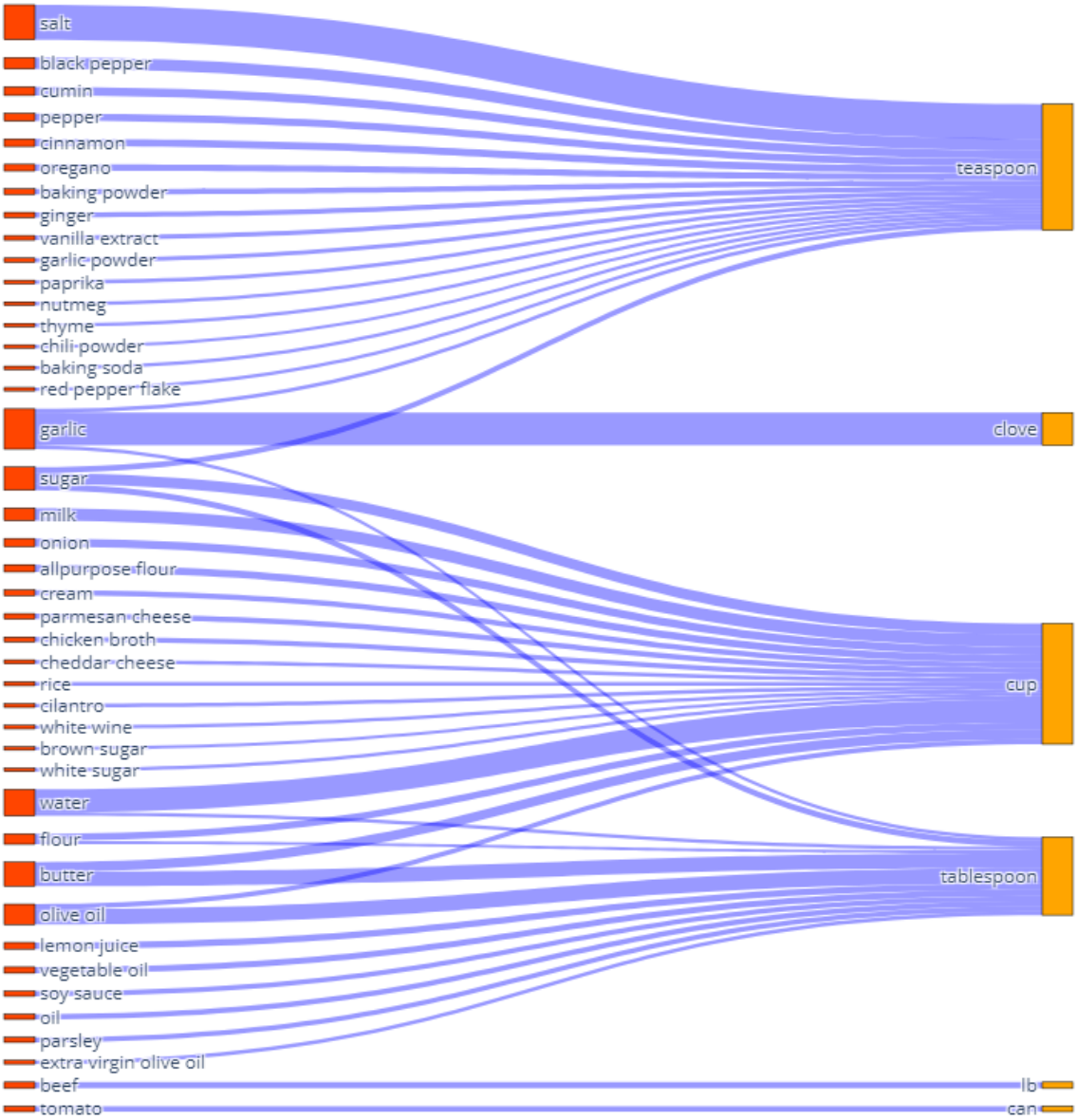

Figure 3.3: A list of 50 most popular ingredient-unit pairs. An ingredient may have one or more units with which it occurs in the recipe's text. For a longer list, please see Appendix A Figure A.1.

words (union) between two ingredients. BERT (Bidirectional Encoder Representations from Transformers) is a language model that captures the contextual meaning of words and phrases. We used a pre-trained BERT model for ingredient mapping to encode the RecipeDB2 and USDA ingredient names into dense vector representations. These vectors were then compared using cosine similarity to find the closest match. RoBERTa (Robustly Optimized BERT Pretraining Approach) is an improved variant of BERT designed with more robust training strategies and adjustments. Similar to BERT, RoBERTa encodes text into contextualized vector representations, which we used to compute the similarity between RecipeDB2 and USDA ingredient names.

We manually evaluated the mapping results for the 200 most popular ingredients. Table 3.2

shows the detailed performance comparison of RecipeDB2 ingredient mapping to that of the USDA. The BERT model outperformed the RoBERTa and Jaccard models, achieving the best overall performance with F1 scores and accuracy of 87.90% and 79.50%, respectively, making it the preferred method for ingredient-to-USDA mapping. Out of 35,474 ingredients in RecipeDB2, 25,903 were successfully mapped to USDA using BERT with a similarity score of $\geq 70\%$. Figure 3.4 (inset) shows the most frequently used ingredients mapping using the BERT strategy. This process allows us to retrieve nutritional information for approximately 73.01% of the ingredients, which we then use to calculate the nutritional profiles of the recipes.

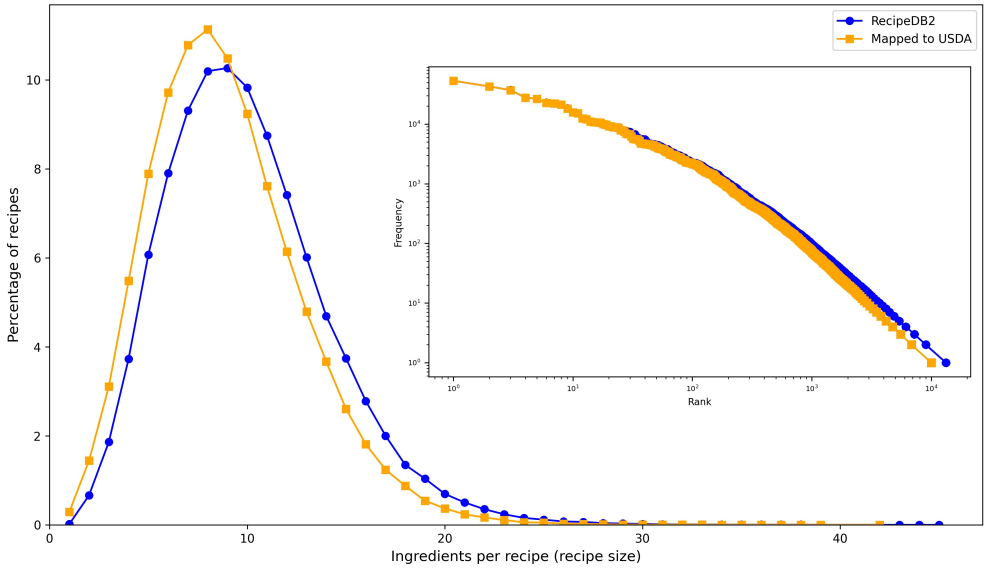

Figure 3.4: Recipe size distribution before and after mapping to USDA, suggesting no significant difference between the two. (inset) Frequency rank distribution before and after mapping to USDA.

3.2.6 Nutritional profile of recipes

The nutritional profile of a recipe was calculated by aggregating the nutritional values of each ingredient, factoring in their respective quantities, and applying standard unit conversions. By mapping ingredient measurements to standardized values, this approach reasonably estimates various nutritional components, including calorific content, macro-nutrients (such as proteins, fats, and carbohydrates), and essential micro-nutrients (like vitamins and minerals). This comprehensive analysis offers valuable insights into the nutritional composition of recipes, enabling a deeper understanding of how different ingredients contribute to the overall health value of a dish. It also facilitates comparisons across recipes, dietary trends, and regional culinary practices, enabling one to make informed choices about nutrition and meal planning.

Top X Ingredients	Model	Accuracy	Precision	Recall	F1 Score
10	**BERT**	**80.00**	**80.00**	**100**	**88.88**
	RoBERTa	60.00	60.00	100	75.00
	Jaccard	20.00	40.00	28.57	33.33
20	**BERT**	**70.00**	**70.00**	**100**	**82.35**
	RoBERTa	50.00	50.00	100	66.67
	Jaccard	25.00	57.14	25.00	34.78
50	**BERT**	**82.00**	**80.43**	**100**	**89.15**
	RoBERTa	52.00	51.02	100	67.57
	Jaccard	32.00	76.47	30.23	43.33
100	**BERT**	**84.00**	**82.79**	**100**	**90.58**
	RoBERTa	60.00	58.76	100	74.02
	Jaccard	30.00	86.20	27.47	41.66
200	**BERT**	**79.50**	**81.42**	**95.51**	**87.90**
	RoBERTa	56.50	56.02	97.27	71.09
	Jaccard	25.50	97.72	23.11	37.38

Table 3.2: Comparison of performance of models in accurately mapping RecipeDB2 to USDA ingredient using metrics accuracy, precision, recall, and F1 score. The word embeddings were generated for the BERT, RoBERTa, and Jaccard models.

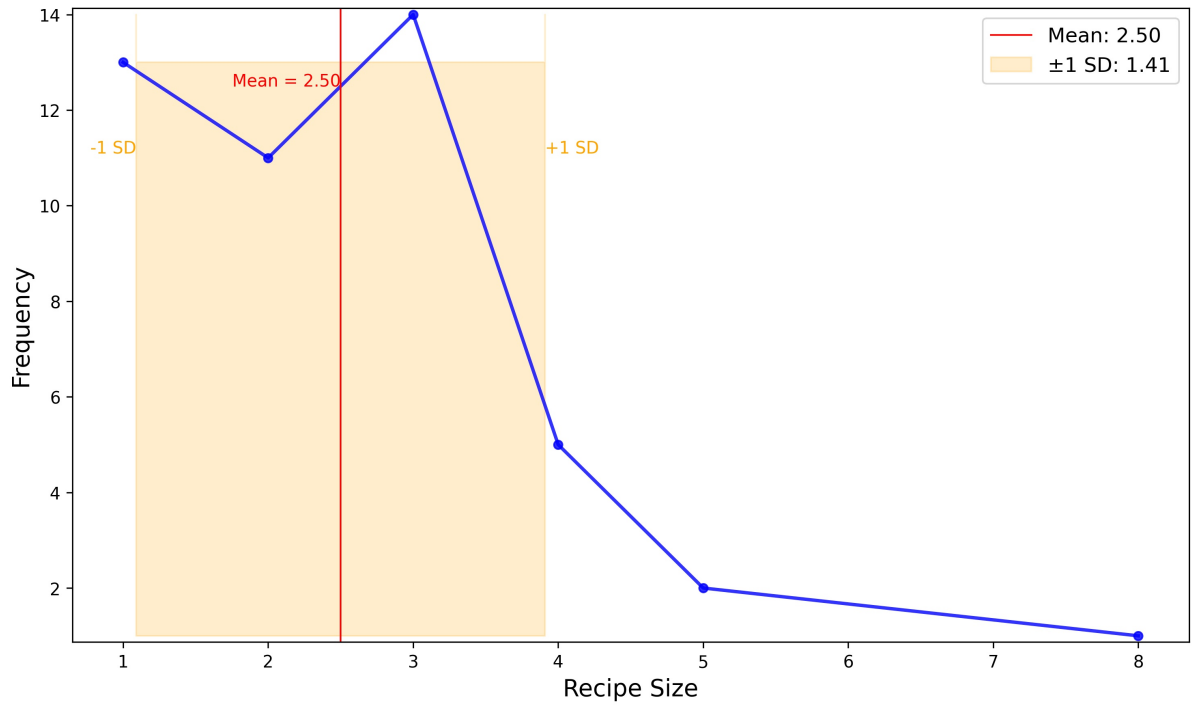

Figure 3.5: Statistics of 43 recipes that were left out of the BERT mapping protocol. The distribution of the recipe sizes has an average of 2.5, indicating that most of the recipes for which no ingredient was mapped to USDA consisted of a small number of ingredients. Christmas Truffle was an outlier, exhibiting a significantly larger recipe size of 8, suggesting it may involve ingredients that are difficult to find a corresponding USDA match.

RecipeDB2 presents a repository of 128,942 recipes. After applying the ingredient mapping strategy, we successfully mapped ingredients to 128,899 recipes (fully or partially), demonstrating that our approach covers the majority of the dataset. However, 43 recipes remained completely unmapped due to the missing ingredient data. These entirely unmapped recipes have fewer ingredients, with a minimum of 1 to a maximum of 8 (Christmas Truffle) (see Figure 3.5). On average, each unmapped recipe contained 2.5 ingredients, for a total of 115 unmapped ingredients across all 43 recipes. This indicates that our mapping strategy efficiently maps the most popular ingredients and recipes with a larger number of ingredients. The recipe size distribution (in Figure 3.4 indicates that before mapping, the average recipe size was 9.978, which marginally reduced to 8.943 post-mapping, suggesting that our approach successfully maps most ingredients to their USDA nutritional profile.

The macro-nutrient analysis of the RecipeDB2 dataset provides insightful statistics that reflect the diversity and composition of recipes within the dataset (see Figure 3.6). The protein content across recipes is from 0 to 285.84 grams, indicating that some recipes may not contain protein sources, while some recipes contain protein-rich ingredients such as legumes, meat, and dairy. The average protein content across all recipes is 49.48 grams, indicating a moderate protein level in most recipes. Carbohydrate content significantly varies from 0 to 1203.84 grams, pointing to recipes such as baked goods or pasta dishes containing carbohydrate-dense ingredients. The average carbohydrate content is 245.20 grams, reflecting a balanced proportion of carbohydrates in the recipes [80]. Total lipid content in the dataset ranges from 0 to 660.41 grams, indicating that the influence of fat content in recipes is likely influenced by the use of oils, nuts, and fatty cuts of meat. The average lipid content is 102.02 grams. The energy values contain considerable diversity, ranging from 0 to 8745 calories, with an average energy of 1989.01 calories across the recipes.

3.2.7 Predicting the category of ingredients

We manually categorized 10,659 ingredients out of a total of 35,474 from RecipeDB [6], which is approximately 30% of the dataset. Left with a large number of ingredients with no category label, we implemented various machine learning models to predict the category for the remaining 24,815 ingredients (70%). Analysis of the uncategorized ingredients revealed a wide distribution in frequency, with some ingredients appearing only once and some appearing in 8,844 recipes. On average, each uncategorized ingredient appeared approximately 5.5 times. This suggests that one should prioritize manually labeling the categories of the most frequently used ingredients.

The models aim to automate the classification process and improve efficiency in assigning a category to each ingredient. Table 3.3 provides the performance comparison of several machine learning models, including Random Forest (RF), Logistic Regression (LR), Gradient Boosting (GB), Support Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT), and K-Nearest

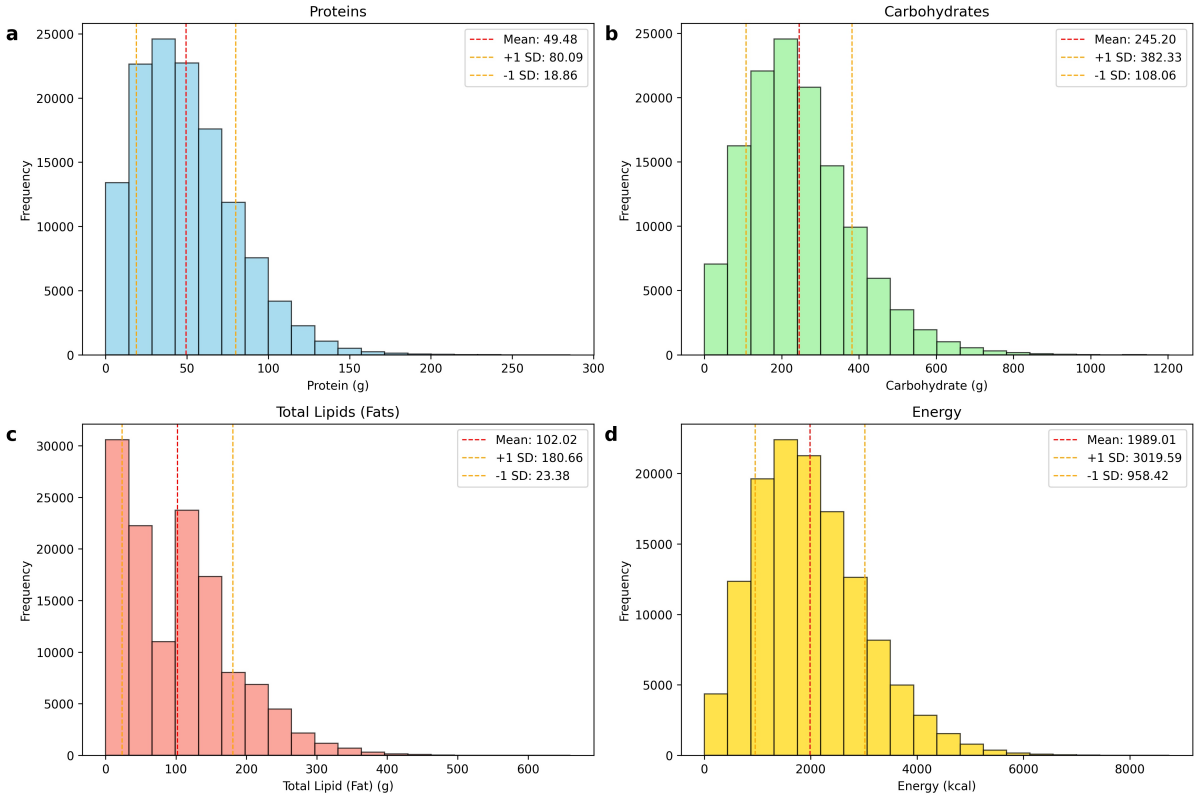

Figure 3.6: Macro-nutrient analysis of recipes illustrating the distribution of key nutritional components. (a) Protein distribution (grams), (b) Carbohydrates (grams), (c) Total lipids (fats) (grams), and (d) Energy (calories). The data highlights variations in macro-nutrient composition across the recipes along with their mean and standard deviation, thus providing insights into the scatter of these macro-nutrients.

Neighbors (KNN). The evaluation metrics used for comparison include accuracy, precision, recall, and F1 score. Random Forest classifier achieved the best performance with an accuracy of 89.12% and an F1 score of 89.16%, indicating the model captures complex relationships between ingredients and their corresponding categories. RecipeDB2 has 34 categories such as Dish (3723), Spice (3314), Vegetable (3303), Meat (3261), Dairy (2437), Additive (2313), etc. (see Figure 3.7). Figure 3.8 shows the composition of the ingredient category based on region.

3.2.8 Dietary style annotations

To categorize recipes by dietary style, we applied the following rules based on the presence or absence of specific ingredient categories. A recipe was labeled as Vegan if it excluded all forms of meat, eggs, dairy, fish, seafood, and certain composite dishes. Recipes were classified as Pescetarian if they included fish or seafood but excluded meat, dairy, and composite dishes. For Lacto-Vegetarian recipes, ingredients include dairy but exclude meat, eggs, fish, seafood, and composite dishes. Ovo-vegetarian recipes include eggs while excluding meat, fish, seafood, dairy, and composite dishes. Finally, Ovo-Lacto Vegetarian recipes included eggs and dairy but

Model	Accuracy	Precision	Recall	F1 Score
RF	**89.12**	**89.38**	**89.11**	**89.16**
DT	87.19	87.44	87.18	87.25
SVM	85.31	86.06	85.31	85.48
LR	82.04	83.17	82.04	82.16
GB	80.74	83.05	80.74	81.09
NB	70.20	71.86	70.19	67.72
KNN	67.78	71.83	67.79	68.92

Table 3.3: Performance of machine learning models in predicting the category of each ingredient in terms of accuracy, precision, recall, and F1 score.

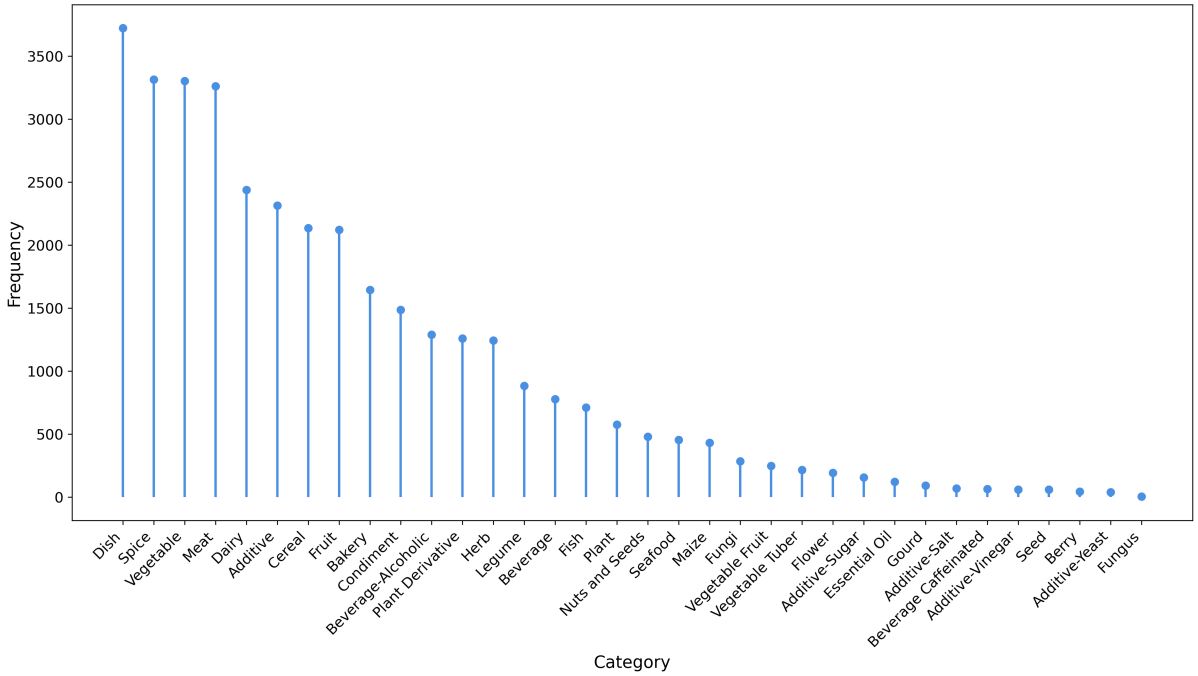

Figure 3.7: Frequency of ingredients in each of the 34 categories after implementing the Random Forest-based strategy. A total of 35,474 ingredients were mapped, of which 10,659 (30%) most frequently occurring were mapped manually, and the rest 24,815 (70%), were done using the automated, machine learning-based protocol.

excluded meat, fish, seafood, and composite dishes.

3.3 Webserver Implementation

RecipeDB2 facilitates exploring and understanding the intricate relationships between cuisines, recipes, and ingredients. With interactive data visualizations and a range of intuitive search options, users can quickly access relevant information, uncover culinary patterns, and gain deeper insights. The platform supports in-depth analysis of recipes and ingredients, helping users explore the rich diversity of global cuisine.

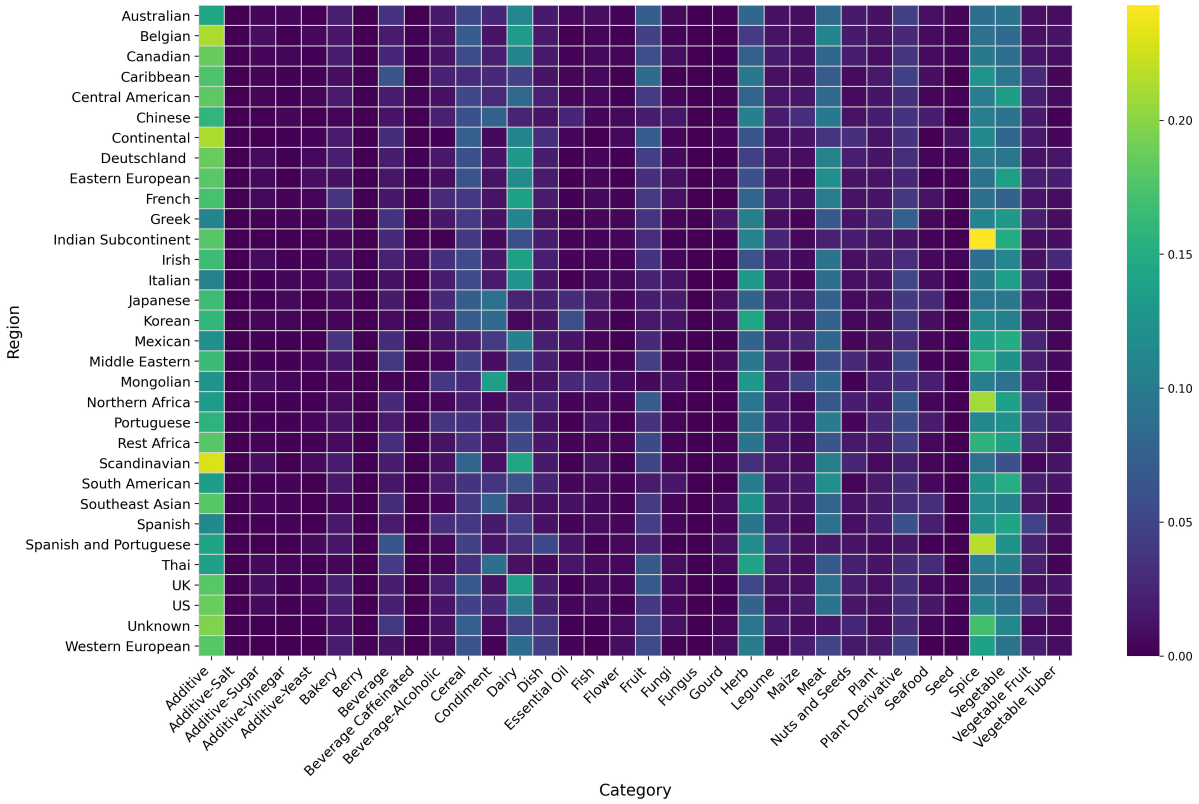

Figure 3.8: Ingredient category composition of recipes for different cuisines (Region). For each cuisine, the heatmap represents the fraction of ingredients belonging to each ingredient category, highlighting the dominant and less prevalent ingredient categories in each cuisine. Additives, Spices, and Vegetables are among the most dominant ingredient categories, with a very heavy representation in the recipes, in general.

Built as a relational database using MongoDB, RecipeDB2 is optimized for efficient data management and querying. The web server is powered by Express, a NodeJS-based web development framework, renowned for its minimalistic and unopinionated design. Express facilitates seamless integration with MongoDB using libraries like Mongoose for schema modeling and efficient querying. The frontend of RecipeDB2 is developed using ReactJS a framework for building frontend, ensuring a responsive and user-friendly interface. The platform employs an NGINX HTTP server to enhance performance, routes requests to the Express application, and enables data compression, leading to faster page load times. ElasticSearch engine would significantly improve the platform's search capabilities by allowing for faster, full-text searches and more advanced query features, particularly in handling large datasets.

RecipeDB2 (https://cosylab.iiitd.edu.in/recipedb2/) is optimized for modern web browsers, providing the best experience on the latest versions of Google Chrome, Firefox, Opera, Internet Explorer, and Microsoft Edge. This cross-browser compatibility ensures smooth navigation and interaction across different platforms and devices.

3.4 Use Cases

3.4.1 Searching recipes by cuisine

RecipeDB2 enables users to search for recipes by cuisine at the 'region' or 'country' level. For example, users can explore the Mediterranean region and Italian as a country, or search for Italian cuisine directly. Each search field provides an autosuggest function. The results page displays a comprehensive list of Italian recipes, including recipe names, estimated macro-nutrient information, and links to individual recipe pages. Clicking on a recipe name directs users to a detailed recipe page, while the 'More Info' tab provides a pop-up with an extensive nutritional profile, covering both macro- and micro-nutrients as provided by the USDA.

3.4.2 Searching recipe using macro-nutrients

RecipeDB2 allows users to search for recipes based on their nutritional profiles, focusing on macro-nutrients such as fats, protein, carbohydrates, and energy. This feature allows users to identify recipes that meet specific dietary requirements or preferences. Additionally, a detailed breakdown of the nutritional profile is available, offering insights into the ingredients that contribute to the overall nutritional value of each recipe. Such nutritional transparency helps users make informed choices about their meals, catering to their health and dietary goals.

3.4.3 Multi-search and multi-attribute search for ingredients and categories

The implementation of multi-search and multi-attribute search capabilities for ingredients and categories significantly enhances the user experience by offering flexibility in refining recipe results. Users can conduct simultaneous searches using both ingredients included and excluded, allowing for more nuanced filtering that caters to individual dietary preferences and restrictions. The advanced search feature further empowers users by enabling them to apply multiple criteria within the 'Ingredients Used' and 'Ingredients Not Used' tabs, ensuring a better search experience. Additionally, users can filter recipes based on both used and unused categories simultaneously, with the ability to select multiple values within each category.

3.5 Discussion

RecipeDB2 is a comprehensive database that enhances culinary knowledge. RecipeDB2 offers a detailed recipe dataset, capturing information on ingredients, preparation methods, cooking

techniques, cultural origins, and nutritional profiles. The user-friendly RecipeDB2 interface enriches the user experience, facilitating a better exploration and understanding of recipes across diverse culinary contexts. By integrating the culinary context of the recipes and ingredients along with their nutritional correlates, RecipeDB2 empowers users to explore culinary relationships.

One of the primary directions in which such a structured dataset can be driven is that of generating novel recipes by applying large language models [81]. In principle, such a novel recipe generation engine can be tailored to individual user preferences, dietary restrictions, ingredient availability, cost, and calorie content using the RecipeDB2 dataset. The integration of culinary and nutritional details of recipes enables users to make informed dietary choices and promotes healthier eating habits. The database serves as a valuable resource for exploring the cultural and historical significance of various cuisines and ingredients.

When compared to other existing recipe datasets, such as Recipe1M [64], Epicurious, Yummly, RecipeNLG [65], and RecipeDB [6], RecipeDB2 stands out for its structured and comprehensive approach to culinary data. RecipeDB2 marks a huge step forward in culinary data by identifying named entities, mapping ingredients to find the nutritional profile, and predicting the category of ingredients. By incorporating metadata such as preparation methods, cultural origins, dietary style, and nutritional information, RecipeDB2 improves data richness and accessibility. This systematic approach enables more nuanced analysis and applications in culinary research while effectively resolving the limits of unorganized datasets. RecipeDB2 empowers researchers, chefs, and food enthusiasts to delve deeper into the complex world of food. This comprehensive approach broadens culinary expertise and emphasizes the significant influence of food on our lives and cultures. Future work will focus on expanding the database's capabilities by integrating domain ontologies such as FoodOn and LanguaL to improve ingredient normalization and enable semantic reasoning across recipes, enhancing user interactivity, and promoting research in culinary sciences. We also aim to incorporate food-chemical relationships using external resources such as FooDB and FlavorDB2, facilitating detailed flavor and health-centric exploration. On the analytical front, we plan to introduce machine-learning pipelines for nutrient prediction [82], ingredient substitution, and recipe rating models.

Chapter 4

RecipeOnt: A Comprehensive Recipe Ontology for Culinary Knowledge Integration

4.1 Introduction

Cooking has evolved over centuries, combining elements of science, culture, and culinary art. It transforms raw ingredients into edible dishes through the application of various cooking techniques and reflects a deep connection between *Homo sapiens* and food. Beyond fulfilling basic nutritional needs, the nuances of cooking convey cultural identity, social customs, and historical traditions. Recipes, as structured representations of cooking practices, capture the complex relationships between ingredients, cooking methods, nutritional values, and cultural influences. Despite significant technological advancements, fully utilizing the vast amount of information embedded in culinary practices remains a challenge. With the advent of artificial intelligence (AI) and its increasing use for tasks such as personalized diet advisory, recipe generation, and dietary planning, the need to organize and structure culinary knowledge becomes more pressing. This is essential for human understanding and will enable machines to process this knowledge effectively. Ontologies and knowledge graphs address this need by representing complex information in a machine-readable format using classes, properties, and relationships [83, 84, 85].

Several efforts have been made to develop structured ontologies for food-related knowledge, but these systems have notable limitations in scope and functionality. AGROVOC [86] primarily focuses on agricultural terms and concepts, providing useful information for agricultural research and policy, but lacks a detailed culinary context. The FOODS ontology [87] attempts to address medical and health-focused diets, such as diabetes, but does not capture broader culinary contexts, including specifics of cultural aspects and flavor of the dish. Among other efforts, FoodWiki [88] dwells on the Turkish cuisine and food safety matters, offering a localized and culturally specific resource but lacking the generalizability and complex culinary relationships. RecipeKG [89], while based on global recipes, does not fully capture the detailed connections between ingredients, cooking methods, nutritional profiles, and flavor science. These limitations highlight the ability of existing systems to support advanced data-driven reasoning and broader applications. Efforts like FoodOn and Taaable have advanced food ontologies further. FoodOn provides a structured and interoperable vocabulary for food products, agricultural sources, and food safety, yet it focuses primarily on the food supply chain and traceability, leaving out

molecular-level flavor profiles, culinary actions, and cross-lingual recipe representations. Taaable [90] supports case-based reasoning for recipe adaptation, emphasizing ingredient substitution and dietary constraints, but does not incorporate nutritional science or flavor chemistry.

To address these gaps, we propose RecipeOnt, a comprehensive ontology designed to represent culinary knowledge in a structured and integrative manner. RecipeOnt combines data from RecipeDB [6], FlavorDB [7], and FlavorDB2 [71], covering over 118,000 recipes, molecular flavor profiles of ingredients, their nutritional details, and cooking processes. It goes beyond the existing efforts by mapping relationships between ingredients, detailing step-by-step cooking methods, and incorporating cultural contexts. Additionally, RecipeOnt adheres to global standards such as schema.org [91] and RDF Schema (RDFS) [92], facilitating integration with other systems and applications.

The potential applications of RecipeOnt are wide-ranging. It can enhance AI models' understanding of recipes, enable the creation of novel dishes through flavor analysis, assist in meal planning for specific dietary needs, and support culinary education through structured, visual representations. By addressing the limitations of existing food ontologies and providing a more comprehensive and adaptable framework, RecipeOnt opens new possibilities for innovation in computational gastronomy, food science, and artificial intelligence. Through the integration of culinary traditions, food science, and advanced data processing techniques, RecipeOnt serves as a powerful tool for advancing research and practical applications in these fields.

4.2 Materials and Methods

4.2.1 Dataset

A rich and diverse collection of data sources, tools, and methodologies supported the development of RecipeOnt, ensuring the ontology's broad coverage. Two key datasets, RecipeDB [6] and FlavorDB2 [71], formed the foundation for this ontology. RecipeDB has 118,172 recipes featuring 20,280 unique ingredients, along with detailed cooking instructions from various cuisines. Additionally, it provides the nutritional profiles of macro and micronutrients for individual ingredients and recipes, as well as information about the culinary tools, processes used, servings, and cooking time of the recipes. FlavorDB2 complements this with molecular and chemical profiles of ingredients, offering a deeper understanding of their composition. Furthermore, it provides the origins of natural sources for the ingredients, either from plants or synthetic sources, adding additional richness to the representation.

4.2.2 Data preprocessing

Data preprocessing involved several key steps to ensure the dataset's integrity, consistency, and usability for creating a knowledge graph aligned with our ontology, capable of supporting advanced applications in the culinary domain. Firstly, RecipeDB and FlavorDB datasets were merged based on the common attributes. This integration step laid the foundation for connecting recipes, ingredients, flavor molecules, associated nutritional profiles, cooking time, servings, region, country, continent, and category. Next, we standardized the column names to match the classes in our ontology. Uniform naming conventions ensured consistency across datasets, simplifying the process of mapping entities and their attributes. Unique identifiers such as RecipeIDs, IngredientIDs, FlavorMoleculeIDs, and NutritionalProfileIDs were utilized to represent entities. Finally, validation was conducted to ensure the accuracy and completeness of the data. Nutritional and molecular information underwent meticulous cross-checking to identify and rectify any inconsistencies or gaps.

4.2.3 Analytical approach

Our approach focused on systematically identifying relationships, integrating data across domains, and ensuring varying levels of granularity to enrich the ontology and enable nuanced knowledge representation. Recipes were systematically mapped to the ingredients and the cooking techniques. This mapping provided a robust framework to represent culinary processes and ingredient combinations. Furthermore, FlavorDB extended these mappings by associating each ingredient with its flavor molecule profile, linking chemical properties to sensory outcomes. Simultaneously, the nutritional profile data enriched this structure by providing detailed information about the nutritional content of each recipe and ingredient. Then, we identified the overlapping entities. For instance, the ingredient 'tomato' was not only mapped to its culinary usage but also linked to its chemical profile, such as lycopene, one of the flavor molecules, as documented in FlavorDB. This integration highlighted the interplay between ingredients' chemical properties and their roles in global cuisines, offering valuable insights into how flavor profiles and nutritional content contribute to regional and cultural culinary practices. Finally, we focused on data granularity to ensure flexibility and depth in representation. This involved creating hierarchical classes within the ontology, ranging from broad categories like 'vegetables' to specific entities like 'spinach'. This hierarchical structuring allowed for representation at various levels of abstraction, catering to different analytical and application needs.

RecipeOnt builds on these insights by integrating data from RecipeDB and FlavorDB to create a modular, scalable, and interoperable ontology. It addresses the gaps identified in the pre-existing work by incorporating molecular flavor profiles alongside traditional culinary data, supporting semantic alignment with global standards like `schema.org` and `rdfs` and automating

ontology construction processes to enhance scalability and reduce manual effort.

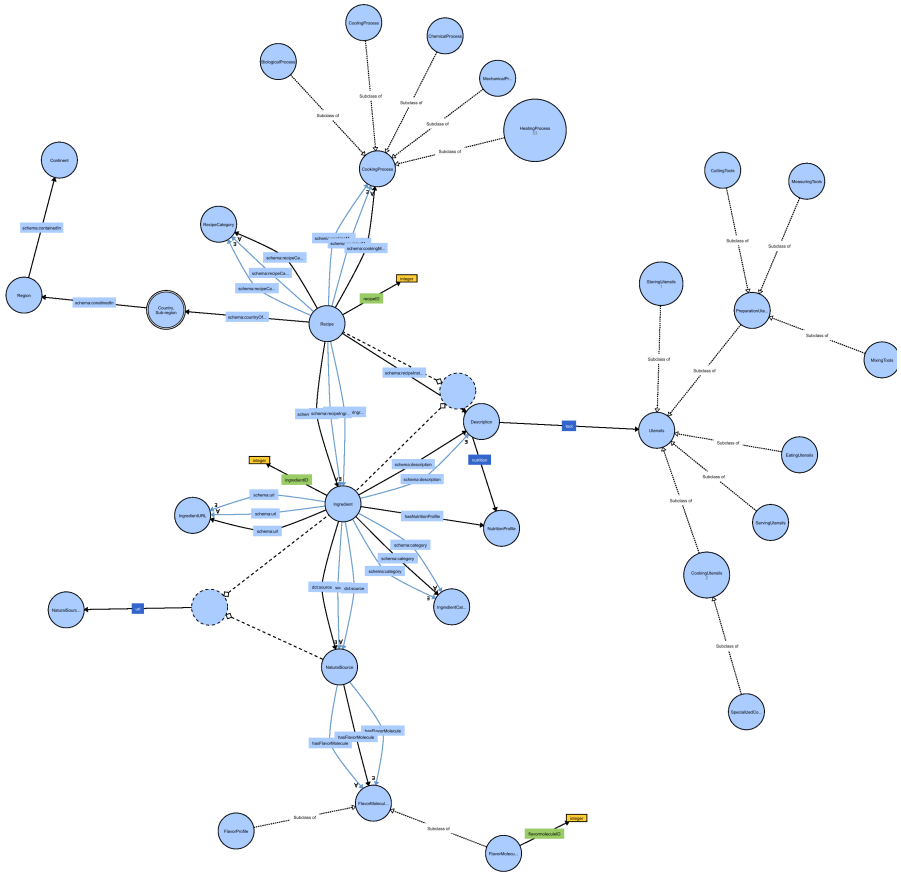

Figure 4.1: Visualization of Recipe Ontology consisting of 34 classes, 16 object properties, 3 data properties, and 56 individuals. This ontology consists of 36 nodes and 54 edges.

4.2.4 Ontology construction: Tools and technologies

The construction of RecipeOnt followed a modular ontology engineering methodology, explicitly adopting Ontology Design Patterns (ODPs) to guide the conceptual modeling and ensure scalability, reusability, and semantic clarity. Modular development allowed us to separate domain subcomponents such as Recipe Structure, Ingredients, Flavor Molecules, Nutrition, and Cooking Actions, each encapsulated as distinct yet interoperable ontology modules. This modularization was instrumental in enabling consistent development and iterative refinement.

We used Protégé [93], a widely used platform for building and managing ontologies. To enhance the conceptual design and refinement process, we utilized the ComodIDE plugin [94] in Protégé. CoModIDE facilitated the application of well-known design patterns (e.g., part-of, participation, and process description patterns), enabling the structured creation of class hierarchies, role restrictions, and inter-entity relationships. This pattern-based approach helped

maintain conceptual consistency and aided in aligning RecipeOnt with best practices in ontology design.

Additionally, RecipeOnt leveraged standard vocabularies such as `schema.org` [91] for general web concepts such as recipes, tools, and cooking methods, Dublin Core terms (DC terms) [95] for metadata annotation (e.g., source, contributor), and OM (Ontology of units Measure) [96] to ensure interoperability and standardization within the domain. The selection of OM over QUDT was a conscious design decision based on comparative evaluation. While QUDT offers detailed coverage of physical quantities, it often introduces modeling complexity with its extensive reliance on named graphs and SPARQL rules. OM, in contrast, provides sufficient expressivity for our culinary use cases (e.g., grams, milliliters, kilocalories) while maintaining better usability within OWL-centric environments like Protégé and RDFLib in Python. RecipeOnt aligns with widely accepted conventions by adopting these vocabularies, enabling seamless integration with other systems and promoting reusability across various applications.

The findings of the data study directly influenced the ontology's design: Core entities, such as `Recipe`, `Ingredient`, and `Flavor Molecule`, were defined, with properties reflecting observed relationships (e.g., `hasNutritionProfile`, `schema:containedIn`, `schema:cookingMethod`, `dct:source`, `sioc:link`, `schema:tool`, etc.). Naming conventions and data structures were aligned with semantic web standards that ensure better interoperability with existing ontologies. Figure 4.1 represents the ontology structure using WebVOWL, emphasizing the modular and semantically linked components.

Custom Python libraries facilitate the transformation of tabular data into RDF (Resource Description Framework) format [92], enabling seamless data integration into the ontology. This approach streamlined the conversion process and ensured compatibility with semantic web standards. The final ontology, enriched with instantiated data, exceeded 250 MB in size. To the best of our knowledge, no other ontology in the culinary domain has previously been instantiated with such an extensive volume of data, making RecipeOnt a unique and comprehensive Computational Gastronomy resource.

4.3 Results

4.3.1 Key characteristics of RecipeOnt

RecipeOnt is a comprehensive ontology comprising over 2.2 million triple statements, making it one of the most extensive resources in the culinary domain. The ontology is structured with more than 50 primary classes, which serve as the foundation for organizing the data. Additionally, it includes over 300 axioms that define the rules and constraints governing the relationships between entities, ensuring consistency within the ontology. One of the key strengths

of RecipeOnt is its highly interconnected nature. It establishes intricate relationships between recipes, ingredients, processes, and tools, creating a rich network of associations that captures the complexity of culinary knowledge. Furthermore, the ontology encapsulates data for each entity, including nutritional information, chemical properties, and procedural details. This multi-faceted representation allows for a holistic understanding of the entities and their interactions, enabling advanced applications such as intelligent recipe recommendations, flavor pairings, and nutritional analysis.

A significant methodological advancement in the construction of RecipeOnt was the transformation of relational databases (RDBs) into OWL-based ontologies [97]. This approach directly maps the database constructs, such as tables, columns, and foreign keys, to OWL classes, object properties, and datatype properties. Doing so automatically generated a domain ontology, allowing for a seamless transition from traditional relational data models to a semantically rich framework. This transformation enriched the data semantics, enabling intelligent information retrieval and more meaningful query results. However, it also introduced certain challenges. Handling complex relationships within the data required careful modeling to ensure accuracy and consistency. Optimizing ontology performance and ensuring interoperability with existing knowledge graphs posed technical difficulties, necessitating advanced strategies for effective integration. Despite these challenges, this approach significantly enhanced the representational power of RecipeOnt, making it a valuable tool for advanced applications in the culinary domain.

4.3.2 Ontology evaluation

We evaluated RecipeOnt using the Ontology Quality Metrics Representation (OQuaRE) [98], providing a comprehensive assessment of its structural complexity, semantic richness, and overall quality. Table 4.1 presents the computed metrics for RecipeOnt with detailed explanations of the metrics.

Annotation Richness (ANOnto) measures the mean number of annotations per class, with RecipeOnt achieving a high value of 4.403. This enhances clarity, interoperability, and reusability for external systems and semantic applications. Attribute Richness (AROnto) of 0.538 reflects the average number of properties per class, suggesting a balanced representation of attributes without introducing excessive complexity. The Class Coupling (CBOnto) value of 1.00 indicates minimal coupling between classes, promoting independence and enhancing scalability. Object Coupling (CBOnto2) exhibits a score of 0.538, reflecting a structured yet flexible relationship network that supports modularity while maintaining essential interconnections between key concepts.

RecipeOnt demonstrates high Class Richness (CROnto) with a value of 1.0769, indicating diverse instances per class. The Depth of Inheritance (DITOnto) of 3.00 suggests a moderately

deep hierarchical structure, providing detailed classification while maintaining manageability. Relationship Richness (INROnto) of 1.173 reflects a high number of relationships per class, supporting complex semantic queries and enhancing reasoning capabilities. The Ancestor Class Ratio (NACOnto) of 0.591 indicates a moderate inheritance ratio, suggesting a balanced hierarchy without excessive reliance on multiple ancestor classes. The Number of Children (NOCOnto) of 2.178 reflects a moderate branching factor, with most classes having approximately two direct subclasses, ensuring balanced category expansion.

Property Richness (NOMOnto) scores 0.307, indicating a lower number of properties per class. While this suggests simplicity, it may also limit the representation of complex relationships. The Ancestor Paths (POnto) value of 0.538 reflects moderate inheritance depth, balancing deep hierarchies with direct class representations to facilitate efficient navigation. Subclass Ratio (PROnto) of 0.792 indicates high subclass diversity, reflecting broad coverage of different recipe elements and their variations. The Response for Class (RFCOnto) value of 34.857 is significantly high, indicating extensive properties linked to each class. This enhances the ontology's expressiveness and supports comprehensive, detailed exploration.

The Relationship Ratio (RROnto) of 0.207 is low, suggesting fewer relationships relative to the total elements, which may limit in-depth semantic reasoning. Mean Parent (TMOnto) of 0.538 indicates that most classes have a single or limited number of parent classes, simplifying ontology maintenance while preserving structural clarity. The Mean Path Length (WMCOnto) of 1.481 reflects a moderately deep structure, balancing granularity with usability. The Complex Inheritance (WMCOnto2) value of 1.00 indicates minimal multiple inheritance, ensuring a clear and simple class hierarchy that reduces conceptual overlap.

RecipeOnt exhibits high-quality characteristics in annotation richness, class diversity, and relationship richness, making it suitable for capturing complex culinary knowledge. Its moderate performance in areas like inheritance depth, object coupling, and property richness ensures a balanced and scalable structure. Although property richness and relationship ratio suggest areas for improvement, RecipeOnt provides a comprehensive and well-structured framework for semantic applications in Computational Gastronomy.

4.4 Competency Questions

The list of Competency Questions (CQs) that RecipeOnt should be able to answer based on the given text:

- CQ1: What ingredients are required for a recipe?
- CQ2: Which recipes share common ingredients?

- CQ3: Which ingredient with a similar flavor profile can be substituted for a given ingredient?
- CQ4: What is the nutritional profile (calories, proteins, vitamins, etc.) of a recipe?
- CQ5: What are the chemical constituents of an ingredient?
- CQ6: What are the necessary utensils for preparing a recipe?
- CQ7: What cooking methods are used when preparing a recipe?
- CQ8: What recipes use a specific cooking technique (e.g., roasting, baking, steaming)?
- CQ9: What recipes can be prepared with a given set of available ingredients?
- CQ10: What recipes suit a dietary preference (e.g., vegan, pescatarian, high-protein)?
- CQ11: What pairs of ingredients are recommended based on flavor compatibility?

These competency questions serve as a foundation to validate the ontology's scope and effectiveness in addressing challenges involving real-world culinary knowledge representation.

4.5 Important Classes, Properties, and Axioms in RecipeOnt

RecipeOnt is a comprehensive ontology designed to organize culinary knowledge, integrating recipes, ingredients, cooking techniques, tools, nutritional values, and molecular properties. This ontology enables a semantic representation of cooking knowledge, which can be used for various applications, from recipe recommendations to nutritional analysis and personalized meal planning.

4.5.1 Key classes

RecipeOnt defines key classes that capture the essence of culinary practices. *food:Recipe* class is central to the ontology. It links ingredients, cooking methods, and utensils for retrieving recipe information. Ingredients represented by *food:Ingredient*, are fundamental building blocks of recipes. This class not only identifies the items used in cooking but also connects them to their nutritional profiles, cooking processes, and even flavor molecules. The relationship between ingredients and cooking methods is further explored through *food:CookingProcess*, which captures different culinary techniques, such as baking, boiling, and frying. The class *food:Utensil* defines the tools and appliances needed for cooking (e.g., oven, blender). Nutritional information, captured by *food:NutritionProfile*, adds depth to the ontology. This class links ingredients and

recipes to their caloric values and macronutrient contents, providing a comprehensive view of a dish's nutritional value. The *food:recipeCategory* organizes the recipes into categories based on their ingredients, such as vegetables, dairy, meat, etc. This categorization allows us to find recipes that meet specific dietary needs or personal preferences. Similarly, the *food:Cuisine* class associates recipes with their cultural or regional origins, such as Italian, Chinese, or Indian cuisine. This classification highlights the cultural diversity of culinary traditions, enabling users to explore food from different parts of the world. The *food:FlavorMolecule* class represents the molecular characteristics of ingredients.

4.5.2 Important axioms

Axioms in RecipeOnt define logical constraints, classifications, and relationships between entities. These axioms serve to formalize the knowledge represented in the ontology and ensure the consistency and correctness of the data. They define how different classes relate to each other, impose restrictions on properties, and provide rules for classification and hierarchy.

Class hierarchies (Subclasses)

The class hierarchy in RecipeOnt organizes related classes in a structured manner. The axiom $food : FlavorProfile \subseteq food : FlavorMolecule$ indicates that the class of FlavorProfile is a subclass of FlavorMolecule. This means that every flavor profile is composed of individual flavor molecules, which are specific chemical compounds contributing to the taste and aroma of food. Similarly, the axiom $food : CookingUtensil \subseteq food : Utensil$ establishes that the class CookingUtensil is a subclass of Utensil. This hierarchy ensures that the ontology distinguishes between utensils used for cooking (e.g., pots, pans, knives) and those that might be used for other purposes, like serving utensils or decorative items. By defining this subclass relationship, RecipeOnt allows for a more specific understanding of the role of utensils in cooking processes. Another subclass axiom is $food : BiologicalProcess \subseteq food : CookingProcess$, which specifies that biological processes (such as fermentation) are a cooking process. This axiom recognizes that some food preparation techniques involve biological mechanisms.

Property restrictions

RecipeOnt defines property restrictions that govern the relationships between classes. The restrictions $\forall food : hasIngredient.food : Ingredient$ states that every recipe must include ingredients as its components. It prevents recipes from being defined without their necessary components, such as the list of food items and quantities used in the preparation. Another key property restriction is $\forall food : hasFlavorProfile.food : FlavorProfile$, which asserts

that flavor profiles should only be linked to ingredients. This rule ensures that the connection between flavor profiles and recipes is made through ingredients rather than directly associating flavor profiles with the recipe itself.

4.6 Use cases

RecipeOnt offers a broad range of hands-on applications combining culinary heritage with modern technological advancements. Its systematic recipe structure for specifying recipes, ingredients, and culinary methods makes it a valuable resource for various innovative applications in the culinary and technology domains.

4.6.1 Improving AI and Language Models

RecipeOnt plays a pivotal role in enhancing the capabilities of artificial intelligence systems, particularly large language models (LLMs). By embedding rich semantic descriptions and detailed relationships between food items and cooking methods, RecipeOnt allows AI systems to operate with greater contextual awareness and precision. For example, including semantic data improves the accuracy of retrieval-augmented generation (RAG) techniques [99], enabling AI to retrieve and synthesize more relevant and precise information for culinary queries. Furthermore, RecipeOnt supports context-aware ingredient substitutions, ensuring that recipes align with specific dietary restrictions or individual preferences, such as nutritional or culturally significant variations. AI systems enhanced by RecipeOnt can also provide enriched question-answering capabilities, enabling them to justify culinary principles, such as explaining the rationale behind complementary flavor pairings or the science behind specific cooking techniques. These improvements significantly enhance the usability and relevance of AI in the culinary world, empowering users with more intuitive, context-aware, and nutritionally informed culinary solutions.

4.6.2 Novel Recipe Generation

The rich ingredient data and intricate flavor profiles encoded within RecipeOnt open new avenues for AI-driven creativity in recipe development. By leveraging molecular flavor pairing, AI systems can generate novel recipes balancing the taste and texture. This capability encourages culinary innovation and pushes the boundaries of what is possible in modern gastronomy. Moreover, RecipeOnt enables LLMs to adapt recipes for various dietary preferences, such as creating gluten-free, vegan, or keto-friendly alternatives without compromising taste or authenticity. These adaptations ensure that recipes cater to diverse lifestyles and needs. Another

significant use case is the ability of LLMs to recommend sustainable ingredient substitutions based on the carbon footprint of recipes, promoting eco-friendly and sustainable living.

4.6.3 Integrating Semantic Frameworks and IoT

Integrating semantic frameworks with IoT systems opened new avenues for intelligent food-related applications, such as personalized dietary recommendations through devices connected to an IoT system [100, 101]. These devices can identify the health profile of users, detect ingredients, and make meal recommendations according to user preferences. However, gaps in semantic alignment and coverage across existing ontologies limit their effectiveness.

4.7 Discussion

RecipeOnt represents a groundbreaking innovation in culinary science, poised to transform the way we understand and interact with knowledge derived from cooking. By serving as a point of connection for flavor, nutrition, culture, and technique, RecipeOnt unlocks the potential to maximize AI applications in the culinary domain. This integration bridges the traditional with the technological, offering intuitive insights that revolutionize recipe management and culinary exploration. One of the most interesting features is the seamless integration of molecular flavor science with traditional recipe datasets. This enables RecipeOnt to recognize the components of a dish and understand the rationale behind ingredient pairings, fostering opportunities for unprecedented innovation in recipe generation and flavor experimentation. Furthermore, RecipeOnt's adoption of global semantic standards, such as schema.org and RDFs, ensures compatibility with other knowledge graphs and systems, facilitating interoperability and making integration into existing digital ecosystems easier.

RecipeOnt has limitations, such as the potential overrepresentation of popular cuisines within large datasets, which marginalize rare or regional recipes. This imbalance limits the ontology's cultural inclusiveness and may hinder its application in globally diverse culinary contexts. Furthermore, the scalability of the ontology is an issue, particularly when incorporating emerging culinary trends or enabling real-time updates. These updates require continuous fine-tuning and expansion, posing a challenge to maintaining the system's relevance in a dynamic culinary landscape. From a more technical perspective, the current version of RecipeOnt lacks temporal modeling, a crucial aspect for accurately capturing the sequence and duration of cooking steps. The absence of support for formal reasoning, such as consistency checking, inferencing, or rule-based deductions, limits the ontology's potential for advanced knowledge discovery and decision support. Additionally, multilingual support is still limited, which restricts the ontology's usability in non-English contexts and hampers cross-cultural applications and global dissemination.

Despite these challenges, RecipeOnt stands as a transformative tool that redefines the boundaries of culinary science and innovation. It empowers AI to grasp the intricate art of cooking while supporting advancements in sustainable food practices. By bridging the wisdom of culinary traditions with cutting-edge technology, RecipeOnt strikes a delicate balance that few systems achieve. Future work will aim to enhance the ontology's temporal expressiveness, integrate multilingual labels, and incorporate reasoning mechanisms, all while maintaining the balance between precision, usability, and cultural inclusivity.

4.8 Conclusions

RecipeOnt creates new standards for structuring culinary knowledge, combining ingredient and recipe data with molecular science, cultural insights, and nutritional information. The potential of this ontology resides in fields as diverse as AI and food science, education, and environmental sustainability. Its value is not merely in the data it holds but in how it enables deeper reasoning and creativity. RecipeOnt lets one answer complex questions, such as generating novel recipes and deciding what we serve. Whether it is assisting an AI in recommending healthy dishes or explaining why certain combinations of flavors work together, RecipeOnt provides the tools needed to improve our thoughts on food. This means that, although there is still work to be done with the scope of the coverage and eliminating the current limitations of RecipeOnt, it lays a solid foundation for the future of culinary research and technology. More than an ontology, it bridges the art of cooking and food science for endless possibilities for innovation and discovery.

Table 4.1: OQuaRE Metrics with Descriptions, Interpretations, and RecipeOnt Scores

Metric Name	Description	5 (Best)	4	3	2	1 (Worst)	RecipeOnt
ANOnto	Mean number of annotations per class, reflecting additional metadata.	> 0.8	0.6–0.8	0.4–0.6	0.2–0.4	< 0.2	4.403
AROnto	Mean number of attributes (properties) per class, reflecting attribute complexity.	> 0.8	0.6–0.8	0.4–0.6	0.2–0.4	< 0.2	0.538
CBOnto	Number of related classes, assessing interdependence.	1–3	3–6	6–8	8–12	> 12	1.000
CBOnto2	Coupling between objects excluding the root class.	-	-	-	-	-	0.538
CROnto	Mean number of instances (individuals) per class, reflecting instance diversity.	> 0.8	0.6–0.8	0.4–0.6	0.2–0.4	< 0.2	1.0769
DITOnto	Longest path length from the root class to a leaf class, indicating hierarchical depth.	1–2	2–4	4–6	6–8	> 8	3.000
INROnto	Mean number of object properties per class, indicating relational richness.	> 0.8	0.6–0.8	0.4–0.6	0.2–0.4	< 0.2	1.173
NACOnto	Mean number of ancestor classes per leaf class, reflecting inheritance depth.	1–2	2–4	4–6	6–8	> 8	0.591
NOCOnto	Mean number of direct subclasses per class, indicating hierarchical breadth.	1–2	2–4	4–6	6–8	> 8	2.178
NOMOnto	Average number of properties per class, indicating property diversity.	≤ 2	2–4	4–6	6–8	> 8	0.307
PONTO	Number of ancestors per class.	> 0.8	0.6–0.8	0.4–0.6	0.2–0.4	< 0.2	0.538
PRONTO	Ratio of subclass relationships to total relationships and properties.	> 0.8	0.6–0.8	0.4–0.6	0.2–0.4	< 0.2	0.792
RFCOnto	Number of properties directly accessible from each class.	1–3	3–6	6–8	8–12	> 12	34.857
RROnto	Ratio of properties to total number of relationships and properties.	> 0.8	0.6–0.8	0.4–0.6	0.2–0.4	< 0.2	0.207
TMOnto	Mean number of parents per class, reflecting multiple inheritance complexity.	1–2	2–4	4–6	6–8	> 8	0.538
TMOnto2	Mean number of classes with more than one direct ancestor.	-	-	-	-	-	null
WMCOnto	Mean number of properties and relationships per class, reflecting class richness.	1–2	2–4	4–6	6–8	> 8	1.481
WMCOnto2	Mean path length from 'Thing' to leaf classes.	-	-	-	-	-	1.000

Chapter 5

Deep Learning Based Named Entity Recognition of Recipes

5.1 Introduction

Food plays a central role in our lives. Beyond its primary purpose of nourishment and taste, it encompasses a broad spectrum of endeavors touching on health and sustainability. Information extraction in food texts has become increasingly crucial in the modern culinary landscape, where food is not just sustenance but reflects our diverse tastes and interests. As we explore culinary experiences and adapt to dietary preferences, extracting valuable information from food-related texts empowers us to make informed choices. Information extraction [102] enables efficient utilization of food-related data. This includes identifying ingredients and nutritional details in recipes [103], ensuring dietary safety by detecting allergens [104], optimizing restaurant operations through menu analysis [105], enhancing food safety by tracking recalls, cost, and sustainability. These technological enhancements provide deeper perspectives on what we eat and facilitate personalized meal planning, culinary research, and innovation in the food industry.

Recipes are unstructured text, and named entities are their building blocks. Named entity recognition (NER) is a technique for extracting information from unstructured or semi-structured data with known labels [106]. It requires the many-to-one mapping of various named entities in text to their domain-specific categories. NER can extract information from various domains, including reviews, news articles, scientific literature, and food texts. NER not only acts as a standalone tool for information extraction but also plays an essential role in a variety of natural language processing (NLP) applications such as text understanding [107, 108], information retrieval [109, 110], automatic text summarization [111], question answering [112], machine translation [113], and knowledge base construction [114], etc. Recent studies have implemented various deep-learning models, such as BERT [115, 116, 117], DistilBERT [118, 119, 120], DistilRoBERTa [121, 122, 123], spaCy [124], and flair [125, 126, 127].

Traditional NER models, such as Hidden Markov models [128] and conditional random fields [129], rely heavily on rule-based features [130, 131]. DrNER [132] in rule-based NER that can extract food entities from evidence-based dietary recommendations. This work was extended to develop another rule-based NER FoodIE [133], where the rules incorporate computational linguistics information. FoodIE achieved promising results on independent benchmark datasets and has been used to create the FoodBase corpus, the first NER corpus in the food domain. The limitation of the FoodIE method is its dependency on external resources, which have become inaccessible after its publication, rendering the method unusable. With a similar

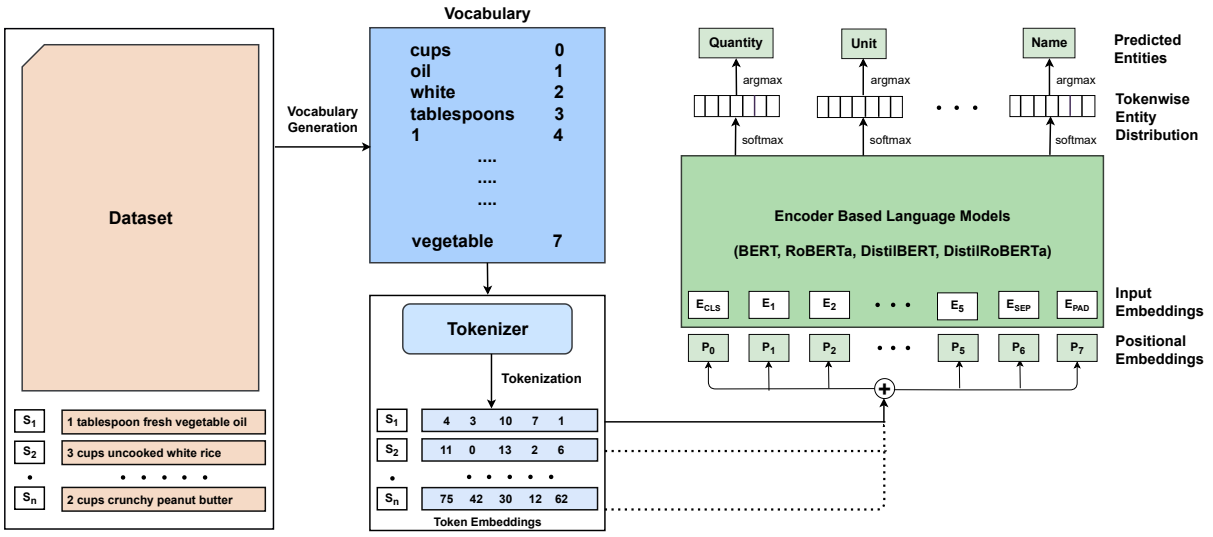

Figure 5.1: The pipeline implemented for fine-tuning supervised deep learning-based named entity recognition comprises three phases. To begin with, we built vocabularies for each of the three datasets. Further, we utilized these vocabularies to convert every word of an input sentence into corresponding token indexes that were subsequently converted to token embeddings via tokenization. Finally, Encoder-Only language models were employed to predict entity tags for the input token embeddings. The spaCy-transformer emerged as the best model with F1 scores of 95.9%, 96.04%, and 95.71% for Manually_Annotated, Augmented, and Machine_Annotated datasets, respectively.

spirit, a data-driven method to find named entities, BuTTER [134], was trained on the FoodBase corpus based on Bidirectional Long Short-Term Memory and conditional random field methods. [135] implemented NER on cooking instructions from multilingual recipes (French, German, and English). They implemented a Conditional Random Field layer on top of Bidirectional Long-Short Term Memory models, achieving F1 scores over 96% in mono and multi-lingual contexts for all classes. Another research [136] implemented a NER approach to identify the food quality descriptors from chats between customers and customer support staff. Previous research [8] used the RecipeDB dataset [6] to identify the named entities in ingredient phrases and cooking instructions. They reported an F1 score of 0.95 (ingredient), 0.88 (processes), and 0.90 (utensils). SciFoodNER [137] is a BERT-based method for recognizing named entities in scientific texts and achieved an F1 score of 0.90. NER can accurately identify ingredient names, quantities, units, state, size, dry/fresh, and temperature within recipes and food-related content.

Computational Gastronomy represents the study of food, flavors, nutrition, health, and sustainability from the computing perspectives [5]. This new data science niche dramatically changes the outlook on food and cooking, traditionally considered artistic endeavors. In this context, building NER models for recipe texts is an exciting proposition, given its applications spanning multiple domains, including disease prediction, cost estimation, flavor profiling, and comprehensive nutritional analysis of recipes. Herein, we present a computational pipeline using

encoder-based language models to extract NERs from recipe text (Figure 5.1).

The salient contributions of research studies presented here are (a) the introduction of augmented and machine-annotated ingredient phrase datasets, (b) analysis of the distribution of RecipeDB ingredient phrases, and (c) a thorough investigation of NER approaches on recipe texts involving statistical, deep-learning-based fine-tuning of language models and few-shot prompting on LLMs.

5.2 Dataset

We have used the manually annotated data consisting of 6,611 ingredient phrases [8] that were sourced from RecipeDB [6], where all named entities were manually labeled (Manually_Annotated_Dataset). An augmented dataset comprising 26,445 ingredient phrases was created by label-wise token replacement, synonym replacement, and segment shuffling (Augmented_Dataset).

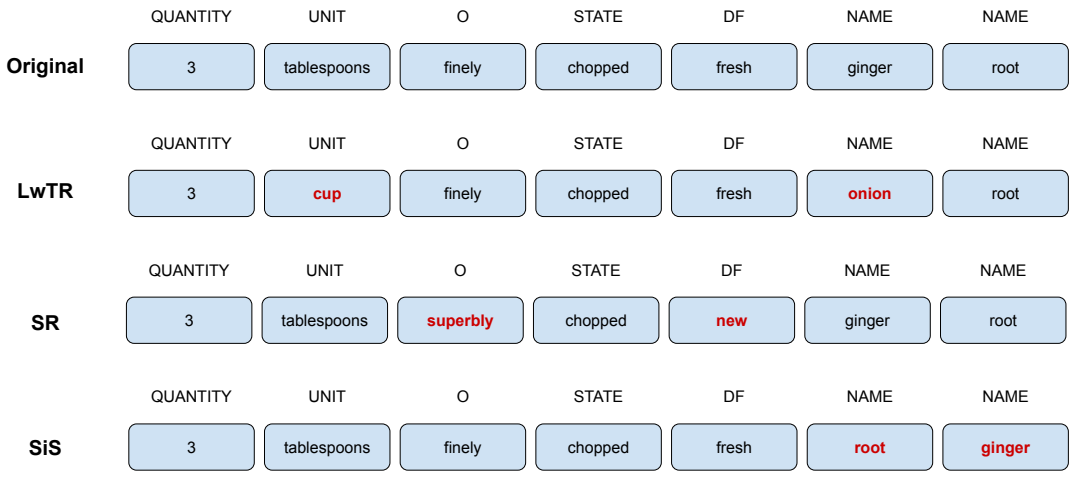

Figure 5.2: Illustration of data augmentation strategies to generate new samples. (a) LwTR: Labelwise Token Replacement: replace a token with a random token of the same label. (b) SR: Synonym Replacement: replace a token with its synonym from Wordnet. (c) SiS: Shuffle within segments: shuffle the tokens under their corresponding label within an ingredient phrase.

We created an extensive repository of 349,762 unique ingredient phrases from the RecipeDB dataset [6] involving semi-automated processing protocol and human curation (Machine_Annotated_Dataset). These ingredient phrases were divided into 2,067 clusters (Stratified Entity Frequency Sampling) based on seven named entity tags (name, quantity, unit, df (dry/fresh), state, size, and temp) and 25% of data (88,526 phrases) were sampled for training. We used 2,187 Manually_Annotated_Dataset for testing.

5.2.1 Data preprocessing

Starting with the 1,150,000 ingredient phrases obtained from the RecipeDB dataset [6], we implemented a preprocessing protocol of lemmatization and manual annotations. A team of culinary experts manually identified the most frequent error patterns present in the machine_annotated dataset. These mistakes were collectively rectified using Python scripts.

- The model could not fully capture the unique culinary language dynamics different from our usual natural language. Color is an adjective, but it might be part of the ingredient. For example, 'Yellow lentils' where 'yellow' in natural language is a color and a usual natural language model would classify it as a 'STATE' of an ingredient. Other examples are red Romano pepper, red chilies, etc.

- Fixing the incorrect placement of named entities. These included the quantity incorrectly labeled as a unit or vice-versa and an ingredient incorrectly classified as a unit or vice-versa. A null value was used to indicate the absence of the unit in the unique list of training datasets.

- Append the fraction and integer together in the quantity as a string to avoid misclassification.

- Removal of special characters. For example, in '+1 chikoo', '+' is dropped.

5.2.2 Data augmentation

Language models need a larger dataset for training. Hence, to extend the Manually_Annotated_Dataset, we implemented three augmentation techniques (Figure 5.2).

Labelwise Token Replacement (LwTR): LwTR replaces the token with a random token from the training set with the same label after taking a call on whether a token should be replaced based on the binomial distribution. This procedure ensures that the original label sequence is preserved.

Synonym Replacement (SR): In a procedure analogous to LwTR, the SR method replaces the token randomly with its synonyms from the Wordnet lexical database.

Shuffle within Segments (SiS): In SiS, the token sequence is first split into segments with the same label, so each segment has some probability of shuffling (as per binomial distribution). The token within the same segment is then shuffled while keeping the order of tokens unchanged.

5.2.3 Machine-annotated dataset

We had a training dataset with 6,611 and 2,187 labeled ingredient phrases for training and testing. Given ten ingredients per recipe on average in a recipe, this yields around 661 recipes for training

and 218 recipes for testing. These data are of limited utility when training transformer-based language models on which our experiments are based and which are known to excel in NLP tasks such as named entity recognition.

Dataset creation

Given the size of the ingredient phrase corpus (1,150,000 ingredient phrases), it was deemed impractical to annotate the entire RecipeDB. After removing duplicates (an ingredient phrase may be a part of several recipes), we were left with 349,762 unique phrases. We adopted a hybrid approach to address this challenge. First, we trained the Stanford NER on the labeled corpus (6,611 + 2,187 = 8,798 Ingredients) to annotate the unique ingredient phrases from RecipeDB. Then, we manually cleaned the machine-generated annotations to identify the error patterns and correct them programmatically. We implemented Stratified Entity Frequency Sampling, a clustering and sampling approach, to sample 25% (88,526 phrases) of the unique ingredient phrases.

Stratified Entity Frequency Sampling

The unique challenges posed by our dataset led to the development of a clustering and sampling technique that we term 'Stratified Entity Frequency Sampling (SEFS).' SEFS ensures a diverse and representative selection of annotated data from a vast corpus, maximizing the capture of varied ingredient phrase patterns. SEFS operates on the premise that ingredient phrases vary based on the combination and frequency of entities they contain. Some phrases may contain the ingredient's name only, while others could be more descriptive, indicating quantity, unit, state, size, and temperature. Ensuring a wide-ranging representation of these combinations in our sample was imperative to train a robust model.

Clustering: The first step in SEFS is to cluster the unique ingredient phrases based on their entity composition. An entity frequency vector is created for each phrase, where each vector component represents the count of a specific entity (name, quantity, unit, state, size, or temperature) in the phrase. These vectors serve as the basis for clustering, where each unique vector corresponds to a cluster. This ensures that ingredient phrases with the same entity composition and frequency are grouped.

Sampling: Once clustered, we sample from these groups to create our dataset. A uniform sampling might not capture the richness and variability of the corpus. Therefore, we adopt a stratified sampling approach. In this method, we sample a fixed proportion (25%, in our case) from each cluster. This guarantees that the resultant dataset contains diverse ingredient phrase patterns.

SEFS ensures that our sample is not biased towards any particular type of ingredient phrase. It

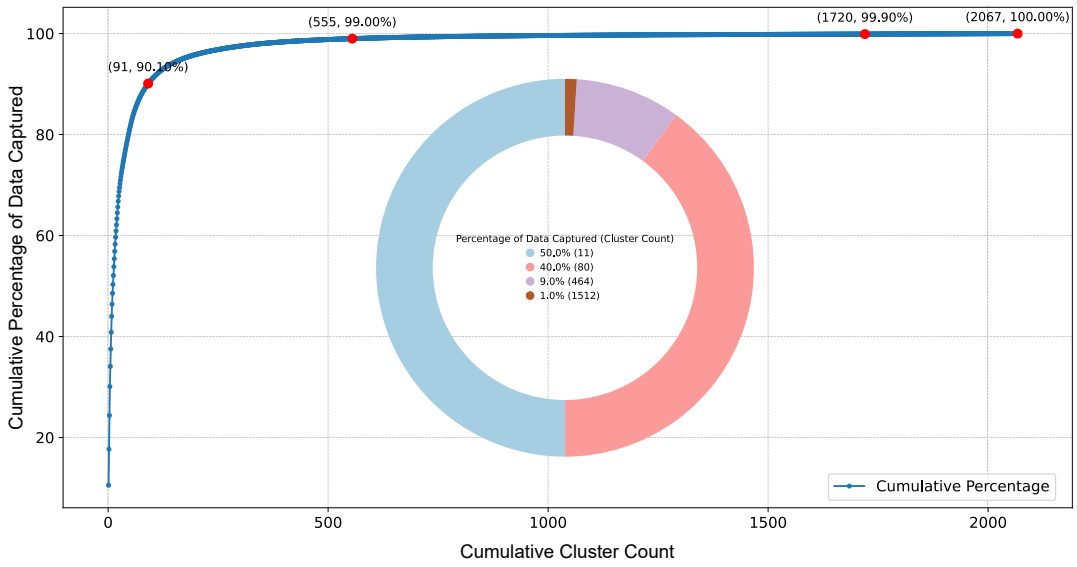

Figure 5.3: Analysis of the percentage of ingredient phrases captured by various clusters. The distribution is extremely skewed, with a few clusters hoarding most ingredient phrases. Half of the ingredient phrases, for example, are captured by merely the eleven largest clusters.

captures the breadth and diversity of the RecipeDB, making it particularly suited for training transformer-based models that thrive on varied data. Moreover, the stratified sampling ensures that even rarer patterns, which could be missed in a random sampling approach, are included in the dataset.

Figure 5.3 depicts the skewed distribution of ingredient phrases across clusters. Around 90% of the total ingredient phrases (1.15 million) can be represented by only 91 unique entity frequency vectors, and the remaining 10% of the phrases require 1,976 different frequency vectors for their representation. This shows that random sampling of ingredient phrases may lead to a bias towards the majority frequency vectors and justifies the SEFS sampling strategy.

5.3 Named Entity Recognition Models

5.3.1 Model configurations

Building upon the previous work [8], we re-implemented the Stanford-NER [138]. The Stanford NER was trained using CRFClassifier with default parameters on an 8 GB CPU RAM system. We implemented diverse deep-learning NER models (BERT, DistilBERT, RoBERTa, and DistilRoBERTa) and NLP frameworks (spaCy, and flair) to find the named entities in the ingredients section. We fine-tuned our datasets on base-case variants of BERT, DistilBERT, RoBERTa, and DistilRoBERTa models with their pre-trained weights using an SGD optimizer

with a learning rate 1e-2. All these models were run on an NVIDIA A100 80GB PCIe GPU card with a batch size of 44 and up to 12 epochs. We have used two different pipelines of spaCy 3.6.1 (en_core_web_lg - a classical rule-based NLP pipeline optimized for CPU, and en_core_web_trf - a RoBERTa-based transformer pipeline). Flair used a pre-trained xlm-roberta-large model to perform the NER.

5.3.2 Modelling techniques

BERT [139] captures the contextual nuances in language by considering the surrounding context of a word in a sentence. Apart from BERT, we employed its other three variants - DistilBERT [118], RoBERTa [115] and DistilRoBERTa [118]. NLP frameworks such as spaCy [140], flair [141] have been implemented to find the named entities of ingredient phrases. A tool Stanford NER [138] employs Conditional Random Fields to analyze and tag entities in a given text with their respective categories. One of its notable features is its ability to recognize and classify entities in multiple languages, making it valuable for multilingual applications.

5.4 Model Evaluation

We employ macro-F1 score, precision, and recall to evaluate our models' predictive performance. These metrics address the inherent class imbalance in our datasets, where accuracy can be misleading. The F1 score provides a robust measure in such cases. Precision and recall are equally critical for our task, as we prioritize correctly identifying all valid ingredient tags (particularly names and quantities) without omissions. While the macro-F1 score is an average of tag-wise F1 scores, it's important to note that it doesn't directly follow the typical harmonic mean relationship with precision and recall. This is because macro-averaging calculates these metrics separately for each label and then averages them, giving equal weight to all labels – a crucial distinction for interpreting results in multi-label classification tasks.

5.5 Results

Pattern recognition aimed at NER across manual, augmented, and machine-annotated datasets is difficult due to degenerate tags corresponding to the same named entity. These ambiguous associations have origins in the linguistic subtleties referring to food's taste, value, and utility. For example, the word 'sour' in 'sour cream' signifies STATE, whereas in 'ice cream,' it collectively represents an ingredient; hence, both entities should belong to the NAME tag.

Herein, we present state-of-the-art models based on deep learning and statistical approaches

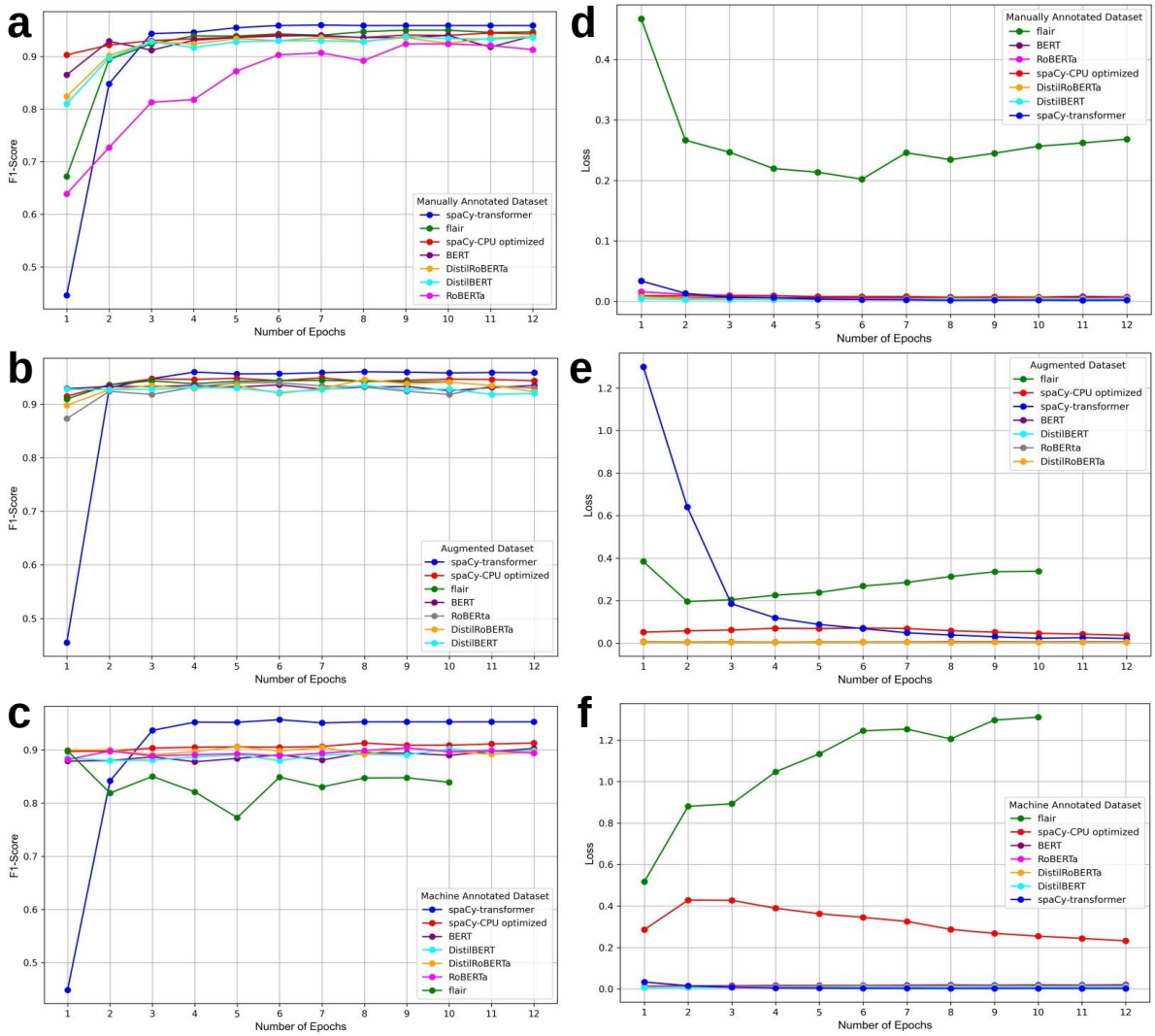

Figure 5.4: Model comparison based on F1-scores and loss. (a), (b) and (c) represent epoch-wise F1-score for Manually Annotated, Augmented, and Machine Annotated Datasets, respectively. Similarly, (d), (e), and (f) represent the epoch-wise Loss score for three datasets.

for named entity recognition in recipe texts. This section is arranged as follows: Section 5.5.1 discusses the implementation of Stanford NER [138], which uses statistical-based techniques for NER. In Section 5.5.2, we evaluate relevant deep-learning-based models fine-tuned on our datasets for performance. Section 5.5.3 describes the tag-wise analysis of named entities using the best-performing model. Finally, Section 5.5.4 delves into the few-shot prompting experiments using state-of-the-art LLMS.

Test Set	Train Set					
	[8]			Present Study		
	AR	GK	Both	AR	GK	Both
AR	96.82	93.17	97.09	96.82	93.31	97.04
GK	86.72	95.19	94.98	86.71	95.16	95.02
Both	89.72	94.72	96.11	**89.16**	94.72	**95.52**

Table 5.1: Performance comparison of [8] and our implementation of Stanford NER. AllRecipes.com (AR) and geniuskitchen.com (GK) refer to the source of recipes from where the raw data was compiled to create the Manually_Annotated dataset.

5.5.1 Stanford NER implementation

We used the Stanford NER [138], to reproduce the earlier work of Nirav et.al [8] and have found consistent results (Table 5.1). We obtained the same results for seven out of nine experiments, and for the rest of the two, the deviation was $< 1\%$. These results signify the importance of CRF-based methods, which have been the go-to methods for recipe NER in most previous works [8, 142, 143, 144, 145]. We implement deep-learning-based, state-of-the-art fine-tuned models by building on what we learned from these articles and rooted in extensive datasets introduced in this study.

5.5.2 Supervised fine-tuning of Encoder-based language models

To enhance the performance of Named Entity Recognition on recipes, we began with a baseline model, Stanford NER [138]. We further implemented seven deep-learning models, including BERT variants (BERT, DistilBERT, RoBERTa, and DistilRoBERTa) and NLP toolkits (SpaCy with CPU optimization, SpaCy equipped with transformer, and flair). To ensure a comprehensive assessment, each model was fine-tuned across three distinct datasets before being consistently evaluated on the Manually Annotated test dataset of 2187 ingredient phrases [8].

Modelling Technique	Manually_Annotated			Augmented			Machine_Annotated		
	F1 (%)	P (%)	R (%)	F1 (%)	P(%)	R (%)	F1 (%)	P(%)	R (%)
spaCy-transformer	**95.90**	**95.89**	**95.91**	**96.04**	**96.05**	**96.04**	**95.71**	**95.73**	**95.69**
spaCy-CPU optimized	94.46	94.52	94.41	94.91	94.92	94.90	91.30	91.36	91.24
Stanford NER	95.52	95.64	95.39	95.16	94.37	95.96	89.9	91.31	88.53
DistilBERT	93.80	95.20	93.60	93.50	93.50	94.60	90.20	92.20	89.70
BERT	94.00	94.70	94.10	93.60	93.70	94.10	90.30	91.50	90.20
DistilRoBERTa	93.80	94.80	93.90	94.60	94.10	95.90	90.60	91.60	90.60
RoBERTa	92.40	92.90	92.60	94.00	94.50	94.10	90.40	91.60	90.20
flair	95.01	96.11	96.05	94.45	95.87	96.14	89.85	88.71	89.22

Table 5.2: Performance evaluation on Manually Annotated, Augmented and Machine_Annotated datasets

Figure 5.4 depicts epoch-wise F1 and validation loss scores for all three datasets across all

models. Table 5.2 encapsulates the results from the best epoch for every dataset-model pair. Despite starting with a lower F1 score, the spaCy-transformer exhibits a rapid learning curve, eventually surpassing the performances of its counterparts. Such discrepancies, especially during the initial epochs across various models, can be attributed to the inherent variability arising from the seed values of model weights and consistent hyperparameters employed. The Augmented dataset, as expected, shows slight performance gains, which is explained by the fact that DL models are data-hungry and given more examples, they would learn the entity-tag mapping better. However, the Machine_Augmented dataset with silver labels created using DL models previously trained on Manually_Annotated datasets appears to echo the inherent variability and noise, coupled with potential mislabeling. This explains a slight decrease in its performance compared to the manually annotated dataset.

A particularly captivating observation emerged from our analysis of the Distil-versions compared to their original BERT-based counterparts. Contrary to conventional assumptions, the Distil-variants held their ground and frequently outperformed the base models. This phenomenon merits a closer examination. Several plausible factors could be driving this unexpected outcome. Firstly, the base BERT variants might be predisposed to overfitting the peculiarities of the training set. Such a tendency would culminate in an escalated validation loss, suggestive of an overly tailored model struggling to generalize to new, unseen data.

Additionally, the presence of fine-grained, spurious correlations within the dataset could be more readily captured by these base models. While seemingly advantageous, this heightened sensitivity might be counterproductive by leading the model to internalize these inconsequential patterns as meaningful, skewing its predictions. Moreover, the potential presence of label noise within the datasets might cause Base BERT models to be overly adept at learning these noise-influenced labels. Consequently, while they might produce tags mirroring the original distribution, these tags might deviate from the expected results in the validation set, thereby being marked erroneous. To summarize, Distil versions, being smaller with fewer parameters, are weaker in capturing the 'bad patterns'–spurious correlations and label noise, which surprisingly acts in favor of their performance metrics.

As we see from our results on the augmented dataset, some models perform better on the augmented datasets, such as spaCyNER and DistillRoBERTa. Because deep-learning-based language models are data-hungry, we enhanced the volume of our dataset by using data augmentation techniques. Consequently, the model performances get a boost as they get more examples to learn about the inherent nature of ingredient phrases.

Analysis of the results obtained in the previous section reveals that spaCy-transformer stands out as the best deep-learning-powered package for recognizing entity tags in recipe texts. It outperformed all other models and baselines on all three datasets. It has also shown stable, consistent learning for our models with the least variance compared to others, as shown in

Figure 5.4. This study reveals a significant breakthrough: spaCy-transformer outperforms the established Stanford NER tagger in recipe entity classification tasks.

	1	lb	frozen	cut	green	beans
Original	QUANTITY	UNIT	TEMP	STATE	NAME	NAME
Stanford NER	QUANTITY	UNIT	TEMP	NAME	NAME	NAME
spaCy	QUANTITY	UNIT	TEMP	STATE	NAME	NAME

	4	slices	bread	,	thick	slice
Original	QUANTITY	UNIT	NAME	O	O	UNIT
Stanford NER	QUANTITY	UNIT	NAME	O	O	O
spaCy	QUANTITY	UNIT	NAME	O	O	UNIT

Figure 5.5: Error analysis of Stanford NER tagger. Stanford NER tagger incorrectly classifies "cut" as NAME instead of STATE, which was correctly identified by spaCy-transformer. Similarly, "slice" classifies as OTHER instead of UNIT, which was correctly identified by spaCy-transformer.

Error analysis comparing spaCy-transformer with Stanford NER: The two most frequent error patterns of Stanford NER that emerged in our analysis were the misclassification of the entity STATE as NAME and the entity UNIT as OTHER (see Figure 5.5). We exemplify these error patterns, showcasing instances where Stanford NER and spaCy-transformer differ in their predictions.

Erroneous predictions using spaCy-transformer: The spaCy-transformer model exhibits erroneous predictions, which include misclassification of ingredient names and brands (see Figure 5.6).

5.5.3 Tag-wise analysis of named entities

In our investigation of epoch-wise learning trends for various entity tags using our top-performing model, a notable correlation emerges between the frequency of a tag in the dataset and its learning trajectory within the model. Consistently, across all three datasets, the 'Quantity' tag exhibits the earliest and most robust learning, while the 'Temperature' tag lags, both in initiation and overall learning, as shown in Figure 5.7 by their F1 scores. This disparity underscores the model's limitations in grasping rare tags as effectively as with prevalent ones. A plausible interpretation of this observation is that while attempting semantic understanding, the models also rely on memorizing specific entity-tag pairings. Consequently, less frequent tags that offer fewer memorization opportunities tend to be under-learned.

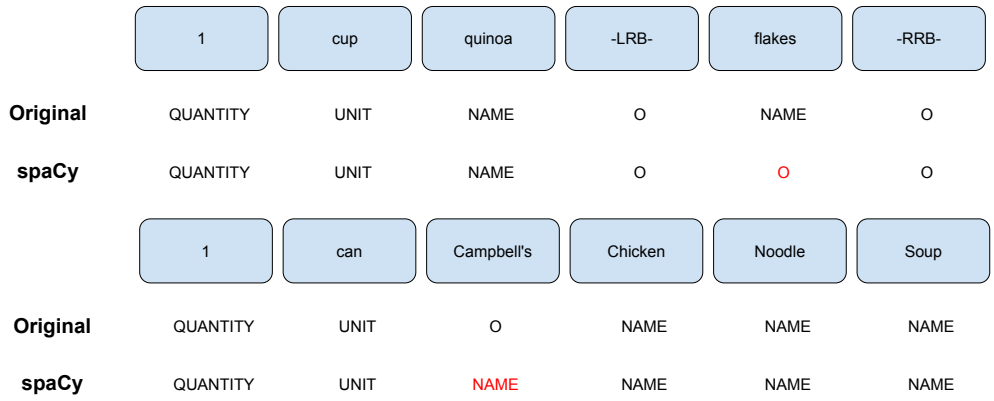

Figure 5.6: Error analysis of spaCy-transformer. spaCy misclassifies "flakes" as OTHER (O) instead of a specific form of quinoa (NAME). Similarly, "Campbell's" is the chicken noodle soup brand name (O) instead of an ingredient name (NAME). -LRB- and -RRB- stand for left and right round brackets, respectively.

5.5.4 Analysis of few-shot prompting on LLMs

Few-shot NER leverages the power of LLMs, such as Chat-GPT and GPT-4 [146, 147], to tackle the challenging task of entity recognition with minimal annotated data. A prompt is given as input to the LLM that outlines the NER task and specifies the context and available examples (Figure 5.8). This prompt acts as a few-shot learning signal, enabling the model to understand the task and context. The pre-trained LLM predicts named entities in a given text. Few-shot NER is useful with limited labeled data, as it can quickly adapt to new entity types and domains. While fine-tuning specific data can further enhance performance, the strength of LLMs lies in their ability to perform remarkably well in a wide range of NLP tasks.

Table 5.3 indicates that pre-trained LLMs have limited exposure to food and culinary datasets during their initial pretraining. Consequently, their performance in in-context learning, especially in food-related named entity recognition, is suboptimal. This deficiency in domain-specific knowledge acquired during pretraining significantly affects their in-context learning capabilities and overall task performance. It underscores the need to fine-tune these models with domain-specific datasets to enhance their effectiveness in specialized tasks.

Model	Macro-F1 (%)	Micro-F1 (%)
LLaMA2-7b	5.88	44.29
LLaMA2-13b	17.06	54.20
Mistral-7b	32.78	47.51
Vicuna-7b	32.90	51.41

Table 5.3: Results of NER using Few-Shot prompting on the state-of-the-art LLMs.

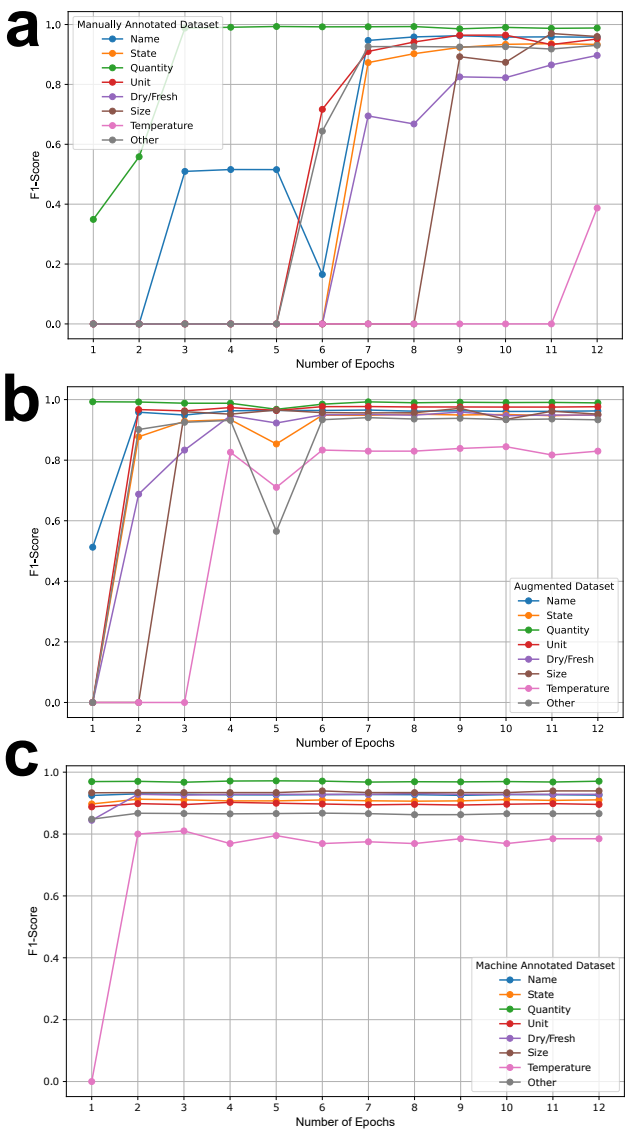

Figure 5.7: Tag-wise learnability of named entities and their final results using the best-performing model–the spaCy-transformer. Figures (a), (b) and (c) depict these results for the Manually_Annotated, Augmented, and Machine_Augmented datasets, respectively.

5.6 Discussion and Conclusions

Our study presented one of the most extensive labeled data resources of named entities from recipe ingredient phrases. Further, we present deep-learning and statistical models built to achieve state-of-the-art results. Nonetheless, our studies are limited in certain aspects of culinary context, nuances of data, and modeling paradigm.

Our present study focuses on only ingredient phrases while not accounting for the recipe instructions, which often carry semantic information about cooking that encodes cultural nuances. Further, static pre-trained models, such as BERT, RoBERTa, and XLM-RoBERTa, come with

```
Below is an instruction that describes a task, paired with an input that provides further context. Write a response that
appropriately completes the request.

### Instruction:
Your task is to do Named Entity Recognition of input sentence. You must assign entity tags to each word in given input
sentence from
[QUANTITY, UNIT,  NAME, TEMP, STATE, SIZE, DF, O].

Number of tokens in input and output sentences must be equal.

Where,
NAME is the name of the ingredient added into the recipe, like onion, garlic etc.
UNIT is the unitary amount of the ingredient added into each step of recipe, like cup, tablespoon, etc.
QUANTITY is a multiple of the UNIT tag which gives the total quantity of the ingredient used in every step of the recipe.
TEMP is the temperature based state of the ingredient, like frozen, hot etc.
STATE is the condition of the ingredient used, like chopped, ground etc.
SIZE is the qualitative amount of the ingredient in each step of the recipe.
DF is the Dry or Fresh condition of the ingredient.
O is Others which is used for entities which are none of these : [QUANTITY, UNIT, NAME, TEMP, STATE, SIZE, DF]

Some Examples:

Input: '2 tablespoons vegetable oil , divided'
Output: [QUANTITY, UNIT, NAME, NAME, O, STATE]

Input: '2 tablespoons dried marjoram'
Output: [QUANTITY, UNIT, DF, NAME]

Input: '1 -LRB- 12 ounce -RRB- box Barilla Gluten Free Penne'
Output: [QUANTITY, O, QUANTITY, UNIT, O, UNIT, NAME, NAME, NAME, NAME]

Input: '2 jalapeno peppers , seeded and minced'
Output: [QUANTITY, NAME, NAME, O, STATE, O, STATE]

### Input:
{input_sentence}
### Output:
```

Figure 5.8: The hand-crafted prompt given to LLMs during few-shot prompting.

inherent biases and might not be fine-tuned to capture the nuances of the food lexicon. Complex culinary instructions may not be amenable to extracting meaningful information. For example, the phrase ‘ground roasted peanuts’ holds multiple layers of information, posing a severe challenge for NER. Names of ingredients unique to specific cuisines might be tokenized sub-optimally, leading to NER errors.

In the future, this research may be extended to include LLM fine-tuning, implementing NERs on cooking instruction, prompt engineering for LLMs for NER on recipes, soft prompt tuning, chain of thought, and implementation of multilingual NER.

Chapter 6

Artificial Intelligence Driven Novel Recipe Generation

6.1 Introduction

The automatic generation of recipes is a challenging yet fascinating problem that blends computational creativity with culinary science. Traditionally, chefs create new recipes by altering ingredients, techniques, and proportions in existing recipes. However, this manual process is time-consuming and highly dependent on individual expertise. To overcome this problem, a question arises in the mind of researchers: 'What if a computer generates healthy and tasty recipes?'. With advancements in artificial intelligence and natural language processing (NLP), researchers have explored the possibility of automating recipe generation using deep learning models. This chapter presents an evolution of our work in AI-driven recipe generation, progressing from generic recipe generation to a more structured, cuisine-constrained approach.

Recipe generation is a specialized text-generation task that involves creating structured outputs consisting of recipe titles, ingredient phrases (with quantities and units), and cooking instructions. Unlike generic text generation, recipe generation requires models to maintain logical consistency, ensure that ingredient usage aligns with the cooking steps, and incorporate appropriate units and measurements. Additionally, personalization plays a crucial role in recipe generation, as dietary restrictions, cultural preferences, and regional cuisines significantly influence the acceptability and usability of a recipe.

Several methodologies have been explored for recipe generation, including statistical models, rule-based systems, and deep learning approaches. Researchers have proposed Transformer-based model [148], routing enforced generation model [51], hierarchical language model [149, 46, 150], knowledge-based models [151, 152] and GRU (Gated Recurrent Unit) based encoder-decoder architectures [50, 153]. GPT-2 and LSTM models have been used to generate recipes from titles and ingredients [148], evaluated using BLEU and ROUGE scores. While these models successfully generated coherent instructions and ingredient lists, most struggled with accurately predicting ingredient quantities and units. Additionally, some lacked contextual awareness, failing to incorporate user inputs effectively.

Recent advancements in transformer-based architectures have significantly improved recipe generation. Earlier methods, such as LSTMs and GRUs, attempted to capture sequence patterns in recipe datasets but often struggled with long-term dependencies. Transformer models such as GPT [154], BERT [155], and RoBERTa [154, 155, 79] have demonstrated superior text

generation capabilities by improving context retention and fluency. The conditional transformer model CTRL [156] consisting of 1.63 billion parameters, was trained with control tokens to generate domain-specific text, enhancing its applicability to recipe generation. Chaudhary et al. [157] introduced ChefAI.IN, an AI system trained on 6,000 traditional Indian recipes from Archana's Kitchen. ChefAI.IN aims to generate Indian recipes using an evolutionary algorithm.

The advent of transfer learning further advances the AI-driven recipe generation. Transfer learning enables models pre-trained on large datasets to be fine-tuned for domain-specific applications. In text processing, recipe generation can be categorized as a text-to-text task [158] where text is fed as input to the model, generating the target text. Several efforts have been made to utilize language modeling over recipe datasets. Parvez et al. [149] used the 'Now You're Cooking' dataset containing 150,000 recipes to construct an LSTM-based language model. Based on the research of [149], Agarwal et al. [150] created a hierarchically disentangled language model by curating 158,473 recipes using named-entity recognition and unsupervised methods. Kiddon et al. [46] proposed an encoder-decoder-based neural checklist model using 84,000 recipes to generate recipes by maintaining a checklist of accessible and already used ingredients. With a similar spirit, Lee et al. [148] presented a transformer-based GPT-2 model (RecipeGPT) for recipe generation using the Recipe1M dataset.

Majumder et al. [50] generated the personalized recipes based on user preference using the dataset of 180k recipes and 700k user interactions. The authors used a bidirectional encoder-decoder to generate recipes and evaluate the generated recipes. In a similar spirit, a routing-enforced generative model [51] was proposed to generate recipes considering the ingredients and user preference. A recent study [159] showed that OpenAI's GPT2 model outperforms the state-of-the-art models to generate recipes. Lam et al. [160] presented a ViT5-based model for ingredient-driven recipe generation using the CookyVN dataset of 26,752 Vietnamese recipes. Their system suggests traditional recipes or generates new ones based on input ingredients, achieving high ROUGE scores (64.45 for ROUGE-1, 35.92 for ROUGE-2, and 38.21 for ROUGE-L). Their study also emphasized the broader role of AI in enhancing everyday life through recipe recommendations, calorie tracking, and personalized meal planning.

Despite these advancements, existing models face several limitations. Several recipes were not able to generate precise ingredient quantities and units, which are critical for usability. Furthermore, some models fail to incorporate user inputs effectively, leading to generic or contextually irrelevant recipes. Addressing these challenges, this chapter presents a structured, generic and cuisine-constrained approach to AI-driven recipe generation. By leveraging fine-tuned transformer models trained on recipe dataset, our approach aims to enhance ingredient relevance, logical coherence, and personalization in generated recipes.

6.2 Materials and Methods

6.2.1 Dataset

We have used the RecipeDB [6] dataset, a structured compilation of 118,171 recipes, 20,262 ingredients, and nutritional information from various cuisines across six continents, representing 26 geo-cultural regions and 74 countries. It encompasses a diverse range of cooking processes (268), such as heating, cooking, boiling, simmering, and baking. Ingredients in RecipeDB are linked to their respective flavor molecules through FlavorDB [7, 71], as well as detailed nutritional profiles from the USDA. This rich resource facilitates researchers into culinary practices—including recipes, ingredients, cooking techniques, and dietary preferences. By interconnecting culinary arts with nutritional profiles, RecipeDB opens up new avenues for applications in food technology, personalized nutrition, and culinary innovation.

6.2.2 Data preprocessing

We preprocessed the dataset to enhance the quality of the generated recipes, including removing incomplete and redundant recipes to enhance clarity and usability. We standardized the length of each recipe to a maximum of 2,000 characters; based on recipe-size distribution, most recipes fall within this range. This standardization streamlines the data and ensures that all recipes are comparable in size. Further, we removed unnecessary white spaces and duplicate ingredients, contributing to the dataset's consistency and cleanliness. The recipes were then reformatted into a structured format, including the recipe title, geographical region, list of ingredients, ingredient phrases, and cooking instructions. This structured format simplifies the data and enables more effective model training, generating innovative and diverse recipes.

6.2.3 Generic recipe generation

We fine-tuned the RecipeDB dataset on the LSTM model at the character and word levels. The model retrieved the embeddings for each character or word and applied a dense layer to generate logits predicting the log-likelihood of the following character or word. The recipes generated by the LSTM-based models lack an in-depth understanding of the relationship between the ingredients and preparatory instructions. This limitation arises because LSTM-based models are often prone to over-fitting, and applying a dropout algorithm to resolve this issue is a significant challenge.

To address these limitations, we implemented the Generative Pretrained Transformer-based model (GPT2) [154], an open-source transformer model developed by OpenAI. GPT2 utilizes an attention mechanism, currently the principal component in a state-of-the-art transformer

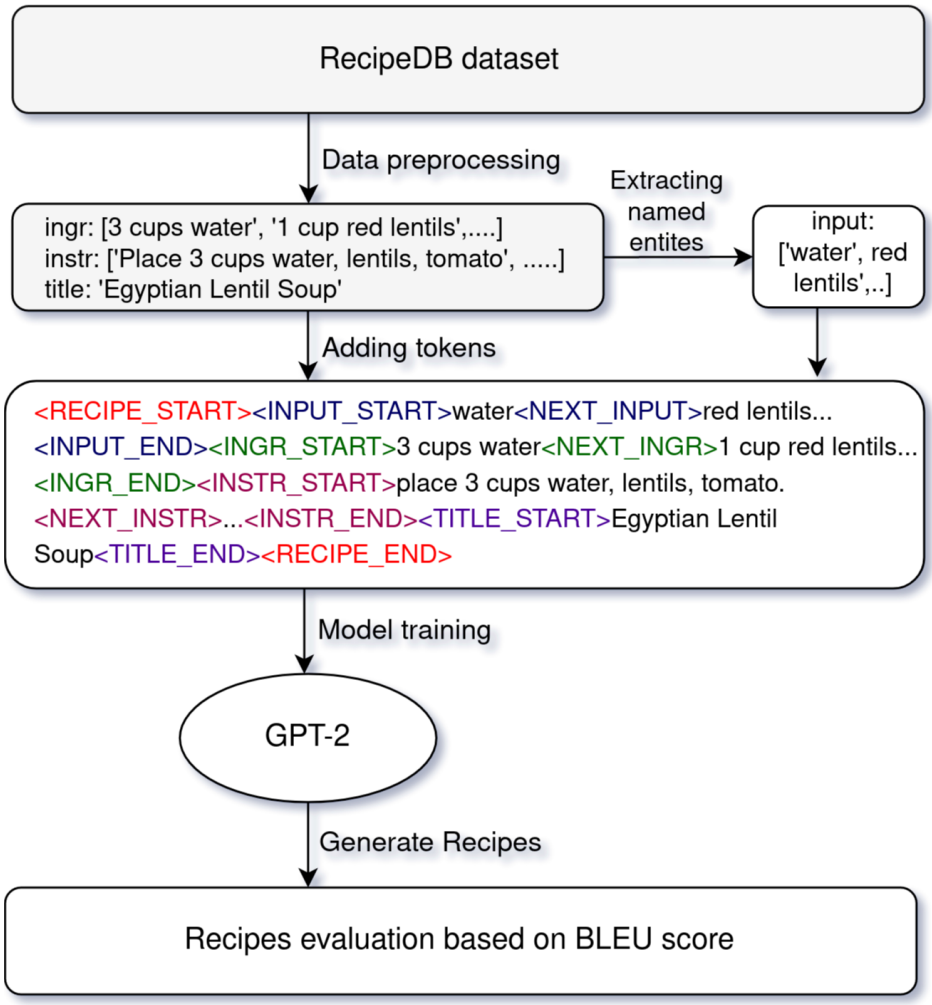

Figure 6.1: Schema to generate novel recipes based on user-specified list of ingredients.

model. The Hugging Face (https://huggingface.co/) was used as the base model for training our language model.

To ensure the model understands the structure of recipes, we preprocessed the data into a long string containing all recipes with different tags employed to distinguish between various sections of a recipe, as shown in Figure 6.1. This preprocessing step was crucial for fine-tuning the GPT2 model to generate coherent and structured recipes.

6.2.4 Cuisine-constrained recipe generation

We generate innovative recipes tailored to specific regional cuisines by constraining ingredient selection (see Figure 6.2). For instance, Indian recipes incorporate ingredients like turmeric, cumin, and lentils alongside traditional cooking techniques, whereas Italian recipes emphasize basil, olive oil, and pasta while preserving authenticity.

We divided the dataset into training and testing with an 80:20 ratio. To facilitate the implementa-

tion of the PyTorch model, we added special tokens to the data and saved the respective sets in text files, ensuring compatibility with our chosen models. In our efforts to harness the capabilities of open-source large language models (LLMs), we explored several options, including GPT-2, LLaMA-3.1-8B [161], and Mistral-7B-Instruct-v0.3 [162]. However, due to hardware constraints, we focused on models with parameter counts under 8 billion. These models balance size, capability, and compatibility with resource-limited environments. GPT-2 is a transformer-based model developed by OpenAI, widely recognized for its generative capabilities in natural language tasks. While GPT-2 is smaller in scale than the newer models, it remains a strong baseline for text generation due to its robust training on diverse datasets. LLaMA-3.1-8B is a part of Meta's LLaMA family; this model is designed to be efficient and highly capable of performing NLP tasks. Its small parameter size makes it a practical choice for applications constrained by computational resources. To optimize its performance, we implemented quantization to reduce memory usage and training time without compromising functionality. Mistral-7B-Instruct-v0.3 is a state-of-the-art model optimized for instruction-following tasks. Its 7 billion parameters provide computational efficiency and accuracy. Each model was fine-tuned over 50 epochs with a batch size of 4, gradient accumulation steps set to 8, and mixed precision (fp16) enabled. The training utilized a learning rate of 5e-4, an Adam epsilon value of 1e-8, and a weight decay of 0.01. The models were fine-tuned using cross-entropy loss as the primary objective, aiming to minimize the divergence between the generated recipes and the reference data.

We applied QLoRA (Quantized Low-Rank Adaptation), which significantly reduces memory requirements during fine-tuning to optimize these models for our computational constraints [163]. Using the bitsandbytes library, we quantized the models with the nf4 (NormalFloat4) configuration, effectively balancing precision and computational efficiency. This configuration minimizes memory usage while maintaining the accuracy required for high-quality text generation. Additionally, we incorporated the PEFT (Parameter-Efficient Fine-Tuning) library to fine-tune a small subset of model parameters. PEFT is particularly useful for large-scale models, focusing on adapting the most critical parameters for task-specific applications. By employing this technique, we could efficiently fine-tune LLaMA-3.1-8B and Mistral-7B on our recipe dataset, avoiding retraining the entire model. The combination of QLoRA and PEFT allowed us to adapt the models to our task of recipe generation, enabling efficient fine-tuning while remaining within hardware limitations. This approach ensured that the models would generate recipes that were high in quality and contextually relevant.

In addition to transformer-based models, we also implemented word-level and character-level LSTM models. The LSTM models were incorporated to capture sequential dependencies over varying granularities, allowing us to analyze the performance of recurrent neural networks in comparison to large-scale transformer models. This approach enabled us to assess how well LSTMs preserve contextual information, especially for tasks requiring fine-grained understanding at both the word and character levels.

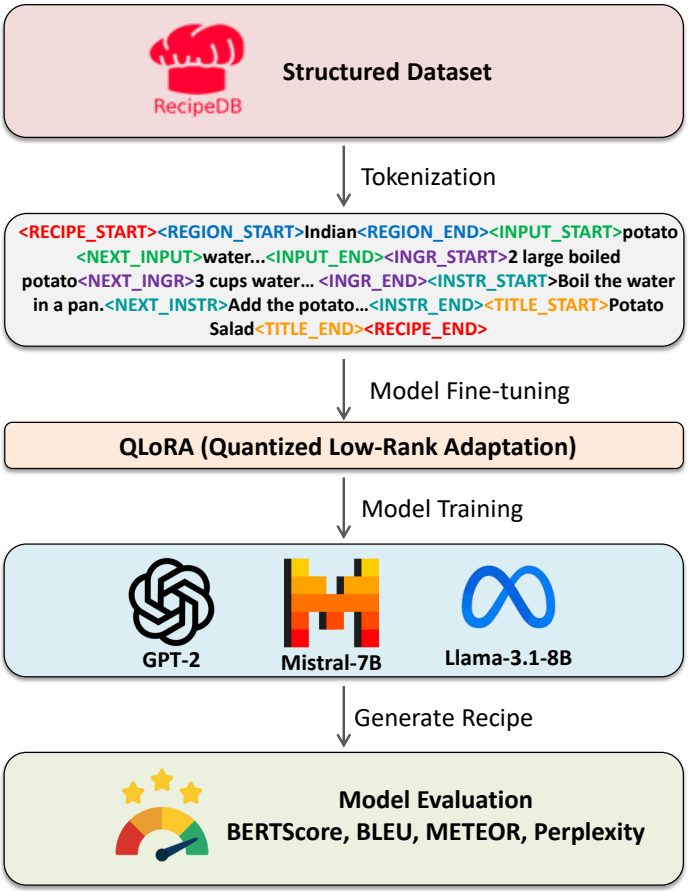

Figure 6.2: Schematic pipeline for cuisine-specific novel recipe generation.

6.2.5 Evaluation metrics

Evaluating the quality of generated recipes is challenging due to the complexity and diversity of culinary language. In this study, we utilized intrinsic evaluation methods [164] to measure the performance of recipe generation models. We employ BLEU, METEOR, BERTScore [165], and Perplexity, as these metrics offer insights into linguistic accuracy, semantic fidelity, and fluency of the generated recipes. Below is an explanation of each metric and its relevance to recipe generation, followed by a discussion on the most suitable metric for this task.

Given our focus on novel recipe generation, these metrics help us analyze different aspects of creativity, linguistic variation, and adherence to established culinary norms:

Given our focus on novel recipe generation, these metrics help us analyze different aspects of creativity, linguistic variation, and adherence to established culinary norms:

1. BLEU (Bilingual Evaluation Understudy): An n-gram-based evaluation metric measuring the similarity between generated text and reference text. BLEU is widely used for its simplicity and effectiveness in tasks where exact word or phrase matches are essential. In recipe generation, high BLEU scores may suggest that the model reproduces existing recipes with limited novelty, whereas low scores could indicate creativity. BLEU does not account for semantic similarity.

2. METEOR (Metric for Evaluation of Translation with Explicit ORdering): Another n-gram-based metric that considers synonyms, stemming, and word order, providing a more nuanced evaluation than BLEU. It is more robust than BLEU in handling linguistic variations, such as different word choices. Like BLEU, METEOR focuses on surface-level linguistic overlap and struggles to evaluate deeper semantic connections or creativity in the recipes.

3. BERTScore: This measures the semantic similarity of the recipe by comparing the contextual embeddings of generated and reference texts using pre-trained transformer models (e.g., BERT). A high BERTScore indicates that the generated recipes capture the semantic structure and intent of the reference recipes, even if the exact wording or phrasing differs. This is especially relevant for novel recipe generation, showing the model's ability to preserve recipes' logical flow.

4. Perplexity: This probability-based metric estimates the fluency and syntactic correctness of the generated recipes. A low perplexity suggests that the model adheres to established culinary grammar and language conventions, ensuring that novel recipes remain comprehensible and coherent. A higher perplexity score may indicate a lack of fluency or plausibility, reducing the practical applicability of the generated recipes. Perplexity does not assess similarity to reference data or the semantic quality of the generated text. It only measures how well the text aligns with the statistical patterns learned during training, which may not guarantee coherence or culinary validity.

For recipe generation, BERTScore emerges as the most suitable metric due to its ability to evaluate semantic fidelity and contextual relevance. Unlike BLEU and METEOR, which rely on surface-level overlap, BERTScore can identify whether the generated recipe captures a recipe's intent, logic, and structure, even if it introduces novel variations or uses different phrasing. However, Perplexity is also essential, as it ensures that the generated text is fluent and adheres to established culinary language norms. High semantic accuracy (BERTScore) combined with low perplexity ensures that recipes are novel. By leveraging these metrics together, we comprehensively evaluate the generated recipes' novelty and quality. BLEU and METEOR offer insights into surface-level similarity, highlighting linguistic differences, while BERTScore assesses semantic alignment. Perplexity complements these metrics by ensuring the generated recipes maintain fluency and culinary plausibility, even when showcasing creative variations. This holistic approach provides a balanced assessment of creativity, coherence, and usability in recipe generation.

6.3 Results

6.3.1 Generic recipe generation

To evaluate the quality of generated recipes, we calculated the BLEU score—a higher BLEU score indicates the closer resemblance of generated text to the actual reference text. Table 6.1 shows that GPT-2 model outperformed all other models, including LSTM (Char-level, Word-level) and DistilGPT2. The superior performance of GPT-2 can be attributed to its self-attention mechanism, which effectively captures long-range dependencies within recipes. This allows it to generate coherent and contextually appropriate ingredient lists and instructions.

DistilGPT2, a smaller and efficient version of GPT-2, achieved a moderate BLEU score of 0.442, demonstrating its capability to generate meaningful recipes while being computationally less demanding. However, GPT-2 achieved a BLEU score of 0.806, showcasing its superior ability to generate high-quality recipes. The generated recipes are accessible via our web application, Ratatouille.

Among the recurrent models, word-level LSTM performed better than character-level LSTM due to its ability to capture higher-level semantic structures rather than individual characters. However, LSTMs inherently struggle with long-range dependencies, which explains their lower BLEU scores than transformer-based models.

Training the GPT2 model was computationally intensive and required approximately 16 hours on an Nvidia A100 Tensorcode GPU server. This substantial training time reflects the complexity and scale of the task, highlighting the trade-off between model performance and computational cost.

Table 6.1: Performance statistics of generic recipe generation models

Model	**BLEU Score**
Char-level LSTM	0.347
Word-level LSTM	0.412
DistilGPT2	0.442
GPT-2 medium	**0.806**

6.3.2 Cuisine-constrained recipe generation

Table 6.2 highlights the performance of recipe generation models across the evaluation metrics, offering insights into the strengths and limitations of each model in generating high-quality recipes. All models demonstrated relatively high BERT-F1 scores, stating their ability to maintain strong semantic alignment with reference recipes. Among the models, GPT-2 achieved the highest BERT-F1 score (85.03%), METEOR score (26.95%), and BLEU score (62.01%),

reflecting its superior capability to generate semantically rich and syntactically accurate recipes. Users can explore cuisine-specific recipe generation through RatatouilleGen.

LLaMA-3.1-8B achieved competitive results with a BERT-F1 score of 84.62% and a METEOR score of 26.14%. However, its BLEU score (26.14%) was considerably lower than GPT-2, suggesting that while LLaMA maintains a strong semantic alignment, it struggles to reproduce exact n-grams. Notably, its perplexity score (4.68) was the lowest among the models, indicating a better fit to the data and more fluent outputs. Mistral-7B, with a BERT-F1 score of 82.15%, showed reasonable semantic alignment but has low BLEU (43.95%) and METEOR (25.23%) scores. It's higher perplexity (6.43) indicates that it produces more diverse outputs, albeit with less certainty.

The Word-LSTM model performed the worst across all metrics, with a BERT-F1 score of 56.52%, a METEOR score of 3%, and a BLEU score of 0.25%. Its high perplexity value (482.97) suggests it struggled to learn the complex dependencies within the recipes, leading to incoherent outputs. The Char-LSTM model performed better than the Word-LSTM, achieving a BERT-F1 score of 77.67%. However, its METEOR (1.9%) and BLEU (1.39%) scores were still significantly lower than transformer-based models.

Model	BERT-F1	METEOR	BLEU	Perplexity
GPT-2	85.03	26.95	62.01	5.12
LLaMA-3.1-8B	84.62	26.14	26.14	4.68
Mistral-7B	82.15	25.23	43.95	6.43
Word-LSTM	56.52	3	0.25	482.97
Char-LSTM	77.67	1.9	1.39	243.87

Table 6.2: Performance evaluation of recipe generation models based on BERT-F1, METEOR, and BLEU scores.

6.4 Webserver Implementation

Web-based applications serve as an effective medium for demonstrating the proposed recipe generation models. To make our work accessible to users, we developed two interactive web applications:

- Ratatouille (https://cosylab.iiitd.edu.in/ratatouille/) – An application to generate novel recipes based on a user-specified list of ingredients.
- RatatouilleGen (https://cosylab.iiitd.edu.in/ratatouillegen/) – A cuisine-constrained novel recipe generation application.

The applications provide an interactive interface where users can input ingredients and, in the case of RatatouilleGen, select a cuisine preference. The backend processes these inputs using the

proposed generative model to generate recipes, including the title, ingredients, and instructions. The architecture of both applications is built on a Python framework. For the frontend, ReactJS is utilized due to its lightweight and efficient design, while Flask is employed for the backend. Both frontend and backend components are containerized using Docker and deployed on a web server.

Figure 6.3: 'Turing Test for Chef' framework for judging the efficacy of computer-generated recipes in fooling a chef.

6.5 Turing Test for Chefs: Evaluating Recipe Authenticity

The 'Turing Test for Chef' framework is intended for chefs who will evaluate the efficacy of recipe generation algorithms (Figure 6.3). After a chef enrolls on the platform, they are presented with a randomly selected recipe from the stack of authentic recipes or those created by a text generation model. While the interface has a high granularity to record the chef's response on a scale of 0 to 5 (0 represents a fake recipe, and 5 stands for an authentic one), the responses were binarized to create a confusion matrix.

A total of 24 chefs participated in the experiment and evaluated 1,472 recipes, leaving out 30 that chefs skipped. The confusion matrix reflects the true positives (real recipes assessed to be real), true negatives (fake recipes assessed to be fake), false negatives (real recipes assessed to be fake), and false positives (fake recipes assessed to be real). The last class of recipes (false positives) is of special interest to us as they reflect the extent to which the computer-generated recipes resembled the real ones in the chef's eyes. With an F1 score of 69.88%, our study demonstrates the capabilities of fine-tuned models trained with well-annotated, structured data to generate meaningful recipe texts that can fool chefs, thereby (barely) passing the 'Turing Test for Chef.' The models have much to improve upon in capturing subtle culinary nuances and will improve

with reinforcement learning, enhanced training data, and superior model architectures. There is ample scope for training models to include relevant culinary details, including multimodal sensory data such as color and texture, and cooking instruction details unaccounted thus far.

6.6 Conclusions

In this chapter, we explored the development and evaluation of AI-driven recipe generation models, focusing on both generic and cuisine-specific approaches. We implemented neural network-based LSTM and transformer-based GPT-2 models for generating novel recipes from a given list of ingredients. A notable modification in our approach was the inclusion of ingredient quantities in each recipe, a feature overlooked in earlier studies [166]. Our evaluation revealed that the transformer-based GPT-2 model outperformed the LSTM model, achieving a BLEU score of 0.806, highlighting its ability to generate semantically correct recipes. We introduced Ratatouille, a web-based application to generate recipes based on an input list of ingredients.

Beyond generic recipe generation, we extended our work to cuisine-specific recipe generation (RatatouilleGen) by integrating culinary constraints into the model. GPT-2 model outperformed other models with a BERT score of 85.03%, and BLEU of 62.01%. This approach ensures that generated recipes align with the authentic characteristics of diverse regional cuisines.

One of the major challenges in this research was the evaluation of generated recipes, a task inherently complex due to the subjective nature of food preferences and the multifaceted structure of recipes. To address this, we developed the Turing Test for Chefs, a web-based framework where professional chefs evaluated the authenticity of AI-generated recipes. A total of 24 chefs participated, assessing 1,472 recipes (excluding 30 skipped instances). The analysis of their results using confusion matrix showed that our model achieved an F1 score of 69.88%, with a notable percentage of false positives, indicating that chefs mistook AI-generated recipes for real ones. These findings demonstrate that while AI-driven models can generate syntactically and semantically coherent recipes, they still require refinements to fully capture culinary nuances, cooking techniques, and ingredient interactions.

This chapter presents promising results in AI-driven recipe generation, but several avenues for future exploration remain. One key direction is personalized and constraint-based recipe generation, where incorporating constraints such as dietary preferences, calorie content, nutritional profile, and budget can make AI-generated recipes more user-specific. Additionally, refining generative models through reinforcement learning by integrating expert feedback can enhance the model's ability to capture culinary subtleties, ultimately leading to more authentic and contextually accurate recipes.

Chapter 7

Cultural Context Shapes the Carbon Footprint of Recipes

7.1 Introduction

Food systems contribute significantly to greenhouse gases (GHGs) [167], responsible for global warming and climate change [168, 169, 170]. Expressed in CO2-equivalent, carbon footprint [171] (CF) captures the total amount of GHGs generated over the life cycle of an entity, primarily accounting for carbon dioxide, methane, and nitrous oxide emissions. It is directly linked to climate change with severe environmental and existential consequences for the planet. Understanding and mitigating the adverse environmental impact of global warming and climate change begins with quantification of carbon footprint. Estimating the carbon footprints of dietary consumption will provide better food choices toward a sustainable future.

Dominated by methane and nitrous oxide, food-centric GHG emissions [172, 173] have origins in various factors from the food system lifecycle encompassing stages from farm to fork [174]. Besides deforestation, storage, transportation [175, 176], and livestock activities [177], the energy needed to convert raw ingredients into edible food significantly contributes to GHGs. Global dietary consumption is dominated by food cooked using traditional recipe protocols. Vetter et al. [178] analyzed the data from 20 food items (13 single crops, 4 animal-based products, and 3 crop groups) and determined that rice has the highest GHG emission among crops (0.73 kg CO2 eq kg-1) followed by other pulses. Food products such as ruminant meat (21.7%), rice (9.4%), and cereals (5.3%) are among the highest contributors to GHG emissions, indicating the environmental sensitivity of meat consumption. In a similar observation, Pathak et al. [179] found that the GHG emission from non-vegetarian food is significantly higher than that of vegetarian ingredients. Mutton was observed to emit 11.9 times as much GHGs as milk, 12.1 times as fish, 12.9 times as rice, and 36.5 times as chapati, a wheat-based flat bread.

Today, human civilization faces the challenge of making food sustainably accessible for an estimated ten billion world population anticipated by 2050 [180]. Towards addressing this challenge, computational approaches [81] rooted in data of recipes [6] and the grammar of cooking [181] present an excellent opportunity for designing environmentally sustainable recipes while minimizing their carbon load. This combinatorial optimization problem requires blending ingredients to create tasty and nutritional dishes. A paramount question, therefore, is, 'How to combine ingredients to create recipes that are simultaneously palatable and environmentally sustainable?' Computational gastronomy [5] enables answering such questions of culinary

origin through a data-driven approach by mining the underlying patterns in food, nutrition, cooking, and carbon footprint.

Garcia et al. [182] reported low carbon footprints of Mediterranean and Atlantic diets having high nutritional scores. On the contrary, Northern and Western Europe and the United States have been reported to have the highest carbon footprints, highlighting the contribution of dairy products, which are also a primary source of high-quality nutrients and proteins. However, extensive analyses of the carbon footprints of recipes from diverse cuisines from across the world have hitherto not been done.

In this study, we aimed to compile the carbon footprint data of food ingredients and estimate the footprints of recipes from across global cuisines (Figure 7.1a). Creating a data repository of the carbon footprints of food ingredients and globally consumed recipes will enable making climate-friendly culinary choices. Such a resource will also enable the application of data-driven approaches for designing environment-friendly recipes and help facilitate sustainable food practices to minimize climate impact. Toward achieving this objective, we integrated carbon footprint data of food products from SU-EATABLE LIFE [183] with RecipeDB [6], a gold-standard repository of the ingredient composition of recipes. Our analysis provides insights into the environmental impact of recipes by examining the role of cultural practices and dietary preferences.

7.2 Methods

7.2.1 Carbon footprint data of food products

SU-EATABLE LIFE [183] (henceforth SuEatable) is the most extensive dataset of the carbon footprints of food products. It provides CF of food items grouped under broad typologies across four food groups (agricultural processed, animal husbandry, crops, and fishing) from 18 world regions and 112 countries (See Appendix B Table B.1). These data points compiled from over 889 research articles provide the ecological impact of food ingredients in CO2 equivalent per kilogram. 'Agricultural processed' and 'crops' commodity groups account for 64.56% of the food products, with the rest equally represented by 'fishing' and 'animal husbandry' (Figure 7.1b). While there is much to be desired in the consistency, geospatial resolution, and coverage of food products across diverse cuisines, this dataset presents the best open resource for ingredient-level CF data. SuEatable [183] dataset was cleaned to address typos, bring consistency in country and food item names, achieve data structure consistency, implement stemming, and purge duplicate entries. The ingredient names in RecipeDB [6] were obtained using state-of-the-art named entity recognition algorithms [8, 150]. The following changes were made in the food items names: 'chocolate or cream filled cookies' to 'chocolate cream cooky',

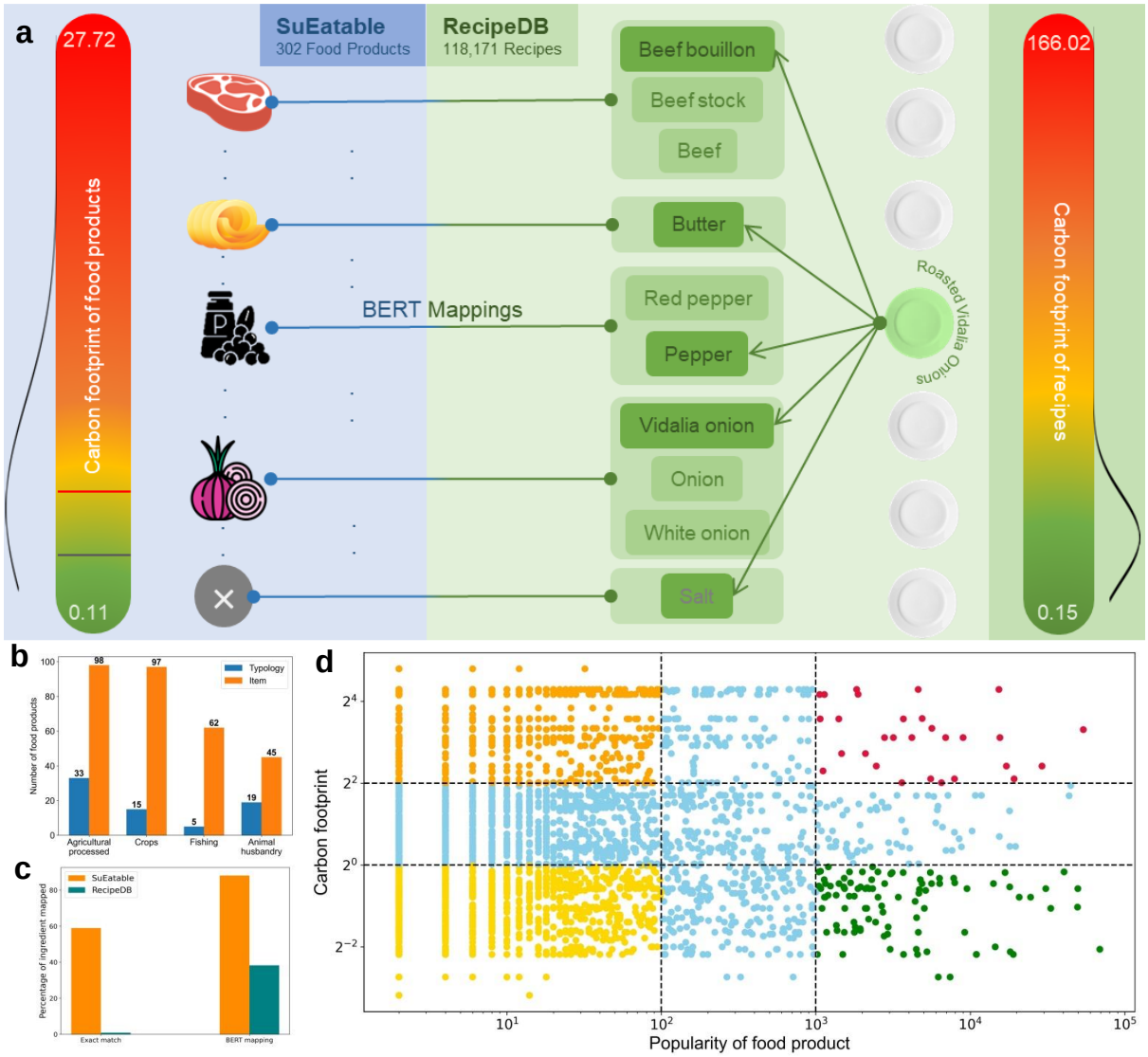

Figure 7.1: Integrating carbon footprint data of individual food products (SuEatable) with recipe composition (RecipeDB) facilitates insights into the environmental impact of ingredients from their culture-driven popular use in culinary preparations. (a) Schematic depicting the process involved in the integration of SuEatbale and RecipeDB data for the estimation of the carbon footprint of recipes. The lines adjacent to the color scale depict the nature of the corresponding data distribution. For the 'food products', the green and red lines depict CF<1 and CF>4 cut-offs, respectively. (b) SuEatable data statistics across food groups. (c) Ingredient mapping before and after BERT mapping, a computational protocol for meaningfully linking ingredients between SuEatable and RecipeDB. (d) Distribution of ingredients, highlighting culturally popular ingredients with low (green) and high (red) footprints and rarely used ingredients that have low (yellow) and high (orange) ecological impact.

'simple cookies' to 'cooky', 'mineral water' to 'water', 'beef bone free meat' to 'beef bone free', 'beef meat with bone' to 'beef with bone', 'chicken bone free meat' to 'chicken bone free', 'chicken meat with bone' to 'chicken with bone.' Similar changes were done for lamb and pork ingredient names. Post-cleansing, we were left with 302 SuEatable food products.

7.2.2 Integration of SuEatable and RecipeDB

To estimate the carbon footprint of a recipe, one needs to break it down into its constituent ingredients (food products) and map them to their respective footprints (Figure 7.1a). RecipeDB provides the ingredient constitution of 118,171 recipes from 26 regions [6]. For any recipe, mapping its RecipeDB ingredients with a relevant SuEatable 'food product' will help us determine its estimated carbon footprint. Therefore, we set out to map 20,280 ingredients from this structured recipe repository to the 302 food products (Figure 7.1a). The disparity in the number of entities between these databases is due to the many-to-one mapping of RecipeDB ingredients to that in SuEatable. Every SuEatable food product appears in different avatars in a recipe. For example, 'sugar' gets mapped to RecipeDB ingredients: white sugar, caster sugar, icing sugar, and confectioner sugar.

An exact match of the ingredient name with the food product label led to 178 entities getting mapped between RecipeDB and SuEatable (Figure 7.1c). Keeping cognizance of many-to-one mappings, we further implemented Word2Vec [184], RoBERTa [79] (roberta-base-nli-mean-tokens), and BERT [185] (bert-base-nli-mean-tokens) language processing models to obtain the word embeddings for food product names in RecipeDB and SuEatable. We mapped the RecipeDB ingredient names to the SuEatable food item having maximum similarity between their embeddings.

7.2.3 Carbon load of a recipe

Carbon Load reflects the percentage of high CF ingredients (CF>4) in a *Recipe i*.

$$\text{Carbon Load of } \textit{Recipe } i = \frac{n_i^{CF>4}}{n_i^{mapped}} \tag{7.1}$$

$n_i{}^{CF>4}$ denotes the number of high CF ingredients in the recipe.

n_i^{mapped} denotes the number of ingredients in the recipe successfully mapped to SuEatable.

Accordingly, averaging across all the recipes in the cuisine will yield the carbon load of the cuisine.

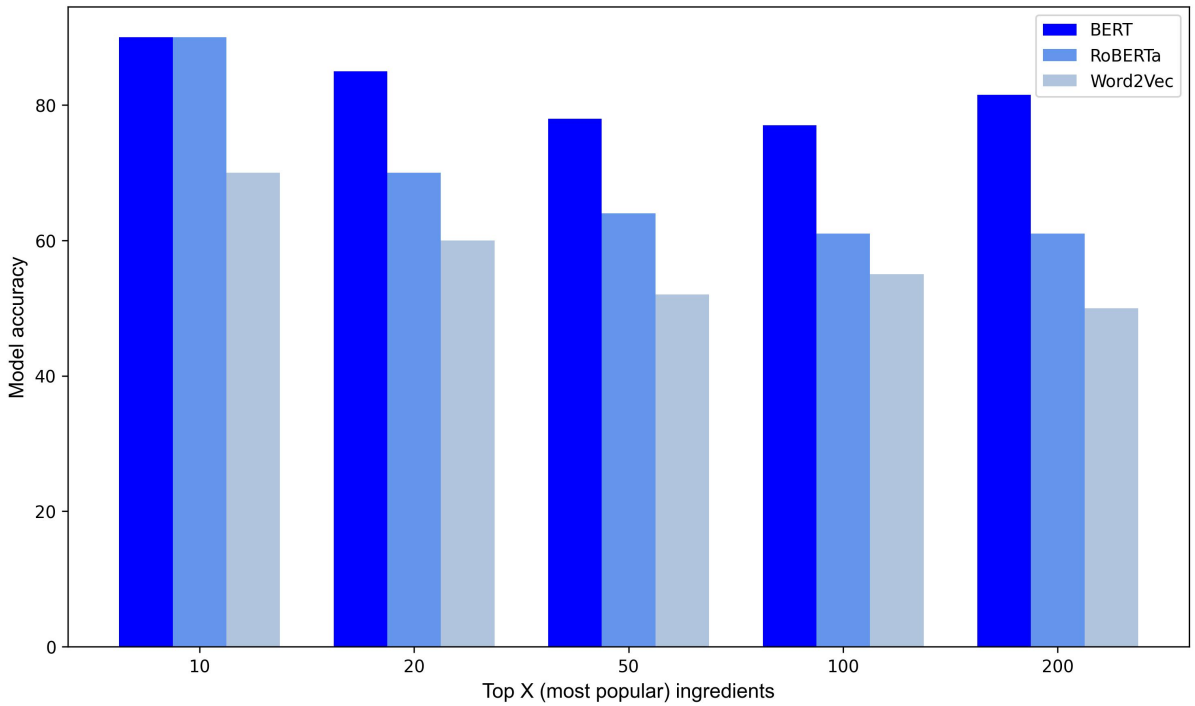

Figure 7.2: Comparison of performance of models in accurately mapping RecipeDB ingredients to SuEatable food product. We used word embeddings generated with BERT, RoBERTa, and Word2Vec for mapping ingredients. For each model, the performance was evaluated for mapping the 200 most popular RecipeDB ingredients to their SuEatable counterparts by manual assessment. The figure presents model accuracy for the Top 10, 20, 50, 100, and 200 most popular ingredients for all three models. While, expectedly the performance drops with an increasing number of ingredients considered, BERT emerged as the best model. The ingredient pairs were pruned by using >81% as the cosine similarity cutoff across all three models.

7.3 Results

7.3.1 Model performance for SuEatable and RecipeDB integration

We compared the performance of Word2Vec [184], RoBERTa [79], and BERT [185] models on the 200 most popular ingredients (with a similarity cutoff of $> 81\%$). BERT was identified as the best model with accuracy and F1 scores of 81.5% and 86.32%, respectively. Figure 7.2 presents a detailed comparison of performance across the models. Figure 7.3 presents the success with which the most popular ingredients from RecipeDB were mapped to their correct analogs in SuEatable. Mapping with BERT-based word embeddings significantly improved the data integration with 7753 (38.2%) ingredient names from RecipeDB mapped to 266 (88.07%) SuEatable food items labels (Figure 7.1b). A significant proportion of popular ingredients, most frequently used in recipes, get mapped with their carbon footprints. See Table 7.1 and Figure 7.4 for a detailed comparison.

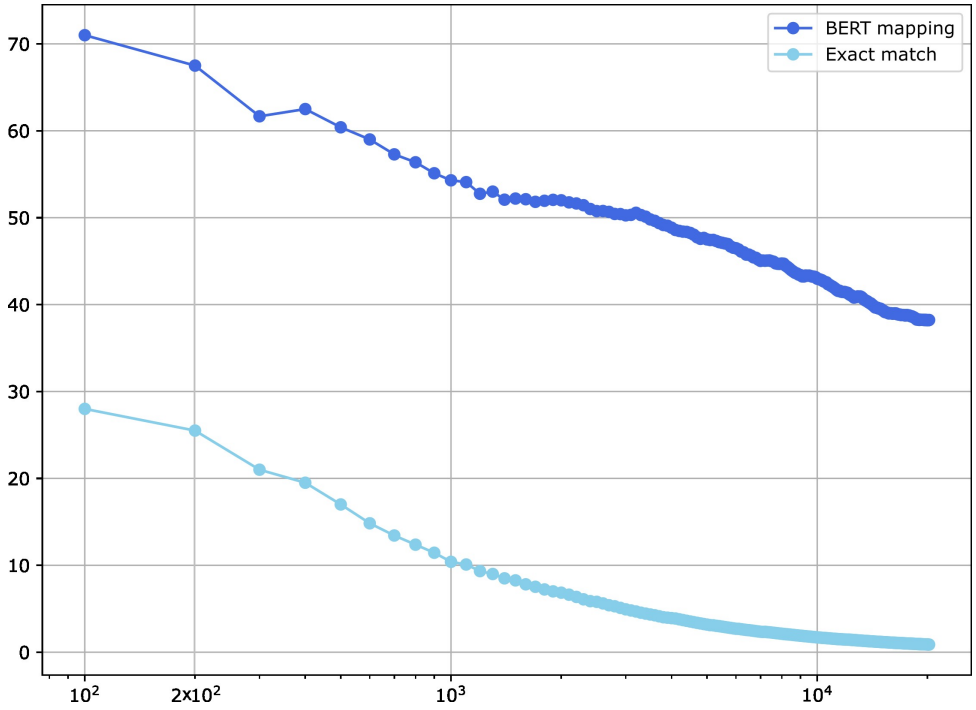

Figure 7.3: Percentage of RecipeDB ingredients successfully mapped to SuEatable for different cutoffs of popularity (Top X most popular). These statistics evaluate mapping protocols in matching the most popular ingredients with their carbon footprint. Further to its superior performance (Figure 7.2), the ability of the BERT model in mapping the most popularly used ingredients in recipes makes it the best candidate and, hence was used as the basis for further calculations. Please see Figure 7.4 for a comparison of the 'Exact Match' strategy with all the models implemented (BERT, RoBERTa, and Word2Vec).

Among the most popularly used recipe ingredients successfully mapped after implementing the BERT-based mapping strategy were garlic clove, black pepper, lemon juice, vegetable oil, and parmesan cheese. This strategy also mapped many rarely used recipe ingredients such as walnut vinegar, welch grape jelly, ratafia biscuit, and gefilte fish. Despite the enhanced mappings post-implementation of the BERT-based mapping strategy, 36 SuEatable food items (coffee ground, haddock, krill, asiago, pomfret, mealworm, emu bone free, and a few others) were left out. See Appendix B Table B.2 and Appendix B Table B.3 for a longer list of correctly mapped (most popular and rarely used) ingredients and those left unmapped (Appendix B Table B.4) with the BERT-based strategy.

7.3.2 Carbon footprint of food ingredients

The integration of SuEatable with RecipeDB presents an interesting perspective on the ecological impact of food ingredients and their popularity in recipes (Figure 7.1d). Logarithmic scales were used to depict ingredient popularity (log 10) and carbon footprint (log 2) due to the wide range of data and uneven scatter. Some ingredients are highly prevalent and are used in thousands of

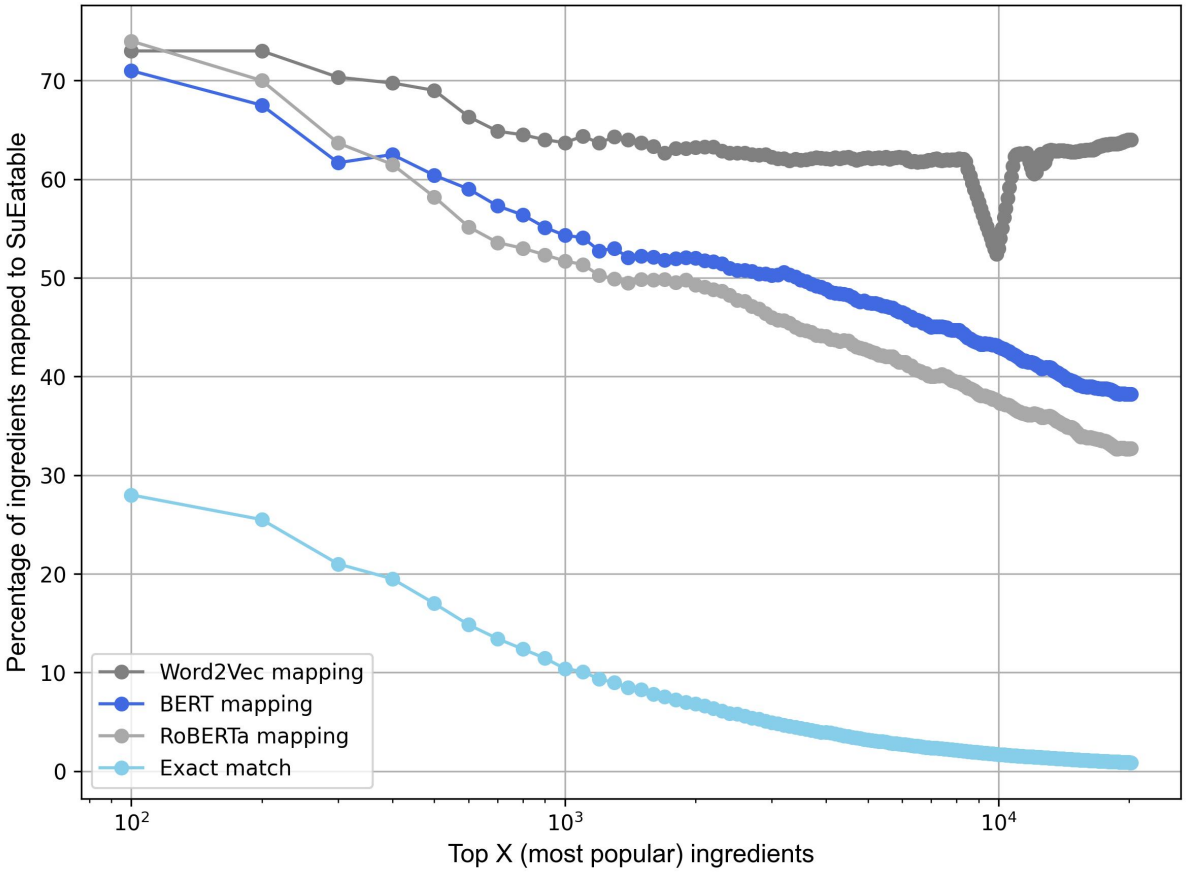

Figure 7.4: Comparison of the model performance for successfully mapping Top X RecipeDB ingredients to SuEatable. Despite a superior mapping of ingredients of the Word2Vec model, by comparing with the manually curated ground truth, we show that BERT is a better model (See Table 7.1).

recipes across cuisines (onion, 69096; butter, 54026; garlic clove, 49786), whereas many are rarely used. Carbon footprints (measured in CO2 equivalent per kilogram) values range between 0.11 (bean and its varieties) and 27.72 (beef and its varieties). Due to the low spatial resolution of data, CF values represent global averages.

Ingredients popularly used in culinary preparations (>1% of recipes) and having a low carbon footprint (CF<1) are shown in 'green color'. These 98 ingredients with low environmental consequences and high culinary prevalence include onion, garlic, water, sugar, tomato, lemon, carrot, ginger, and potato, among others. On the other hand, among the 29 ingredients with serious negative environmental consequences due to heavy culinary use and high carbon footprint (Red: CF>4 and >1% of recipes) include butter, milk, cream, cheeses, varieties of beef, chocolates, vanilla, bacon, lamb, pork, fish, and shrimp. The analysis also revealed 1793 ingredients with severe environmental consequences and low cultural adoption (Orange: CF>4 and <0.1% of recipes). These infrequently used culinary ingredients best avoided for positive environmental outcomes include lobsters and variations of beef, pork, bacon, and lamb.

Top X Ingredients	Model	Accuracy	Precision	Recall	F1 Score
10	BERT	90	100	88	93.61
	RoBERTa	90	100	88	93.61
	Word2Vec	70	100	66	79.51
20	BERT	85	88.2	93.7	90.86
	RoBERTa	70	76.47	86.66	81.24
	Word2Vec	60	100	52.94	70.53
50	BERT	78	88.23	81.08	84.5
	RoBERTa	64	82.97	77.14	74.99
	Word2Vec	52	100	38.46	55.55
100	BERT	77	84.5	83.33	83.91
	RoBERTa	61	67.56	76.92	71.93
	Word2Vec	55	100	40	57.14
200	BERT	85.5	93.43	83.91	86.32
	RoBERTa	61	67.85	74.21	70.88
	Word2Vec	50	100	37.41	54.45

Table 7.1: Performance comparison for models implemented to map the embeddings of ingredient names (RecipeDB) to those of food products (SuEatable).

7.3.3 Carbon footprint of ingredient categories

Ingredients are classified to represent their source, culinary use, and flavor profiles. Accordingly, ingredients were grouped into 27 categories (vegetable, legume, meat, fish, seafood, dairy, spice, etc.), borrowing the RecipeDB's classification system (Appendix B Table B.5). The ingredients' category-wise CF statistics (median, lower, and upper quartiles) present their typical environmental impact (Figure 7.5a). Ingredients of vegetable, dairy, meat, miscellaneous, herb, and spice categories are among the most frequently used for culinary preparations. The categories with the highest overall CF are meat (10.347), seafood (10.089), dairy (6.827), dish (5.62), and fish (4.154).

Superimposing ingredient popularity in culinary preparations on top of CF data presents a more transparent picture accounting for cultural influences (Figure 7.5b). With their heavy use in culinary preparations (76.77% of all recipes), dairy products emerge as environmentally the most damaging. Following closely, meat products come next on the list with undesirable environmental consequences, with 71.41% of all recipes using ingredients such as variations of beef, pork, bacon, eggs, and chicken. Despite high CF, seafood, dish, and fish ingredients are infrequently used in recipes (6.64%, 4.98%, and 3.19%, respectively).

Vegetable, herb, and fruit categories of ingredients are of high culinary utility (87.21%, 59.82%, and 33.99% of recipes, respectively) with comparatively low emissions (1.178, 1.021, and 1.340, respectively). Incidentally, gourd (cucumber), vegetable flower (broccoli and cauliflower), and caffeinated beverage (cocoa, coffee powder) classes have rare culinary usage (2.45%, 2.73%, and 2.02% of recipes) and the lowest footprint (0.295, 1.05, and 1.518, respectively).

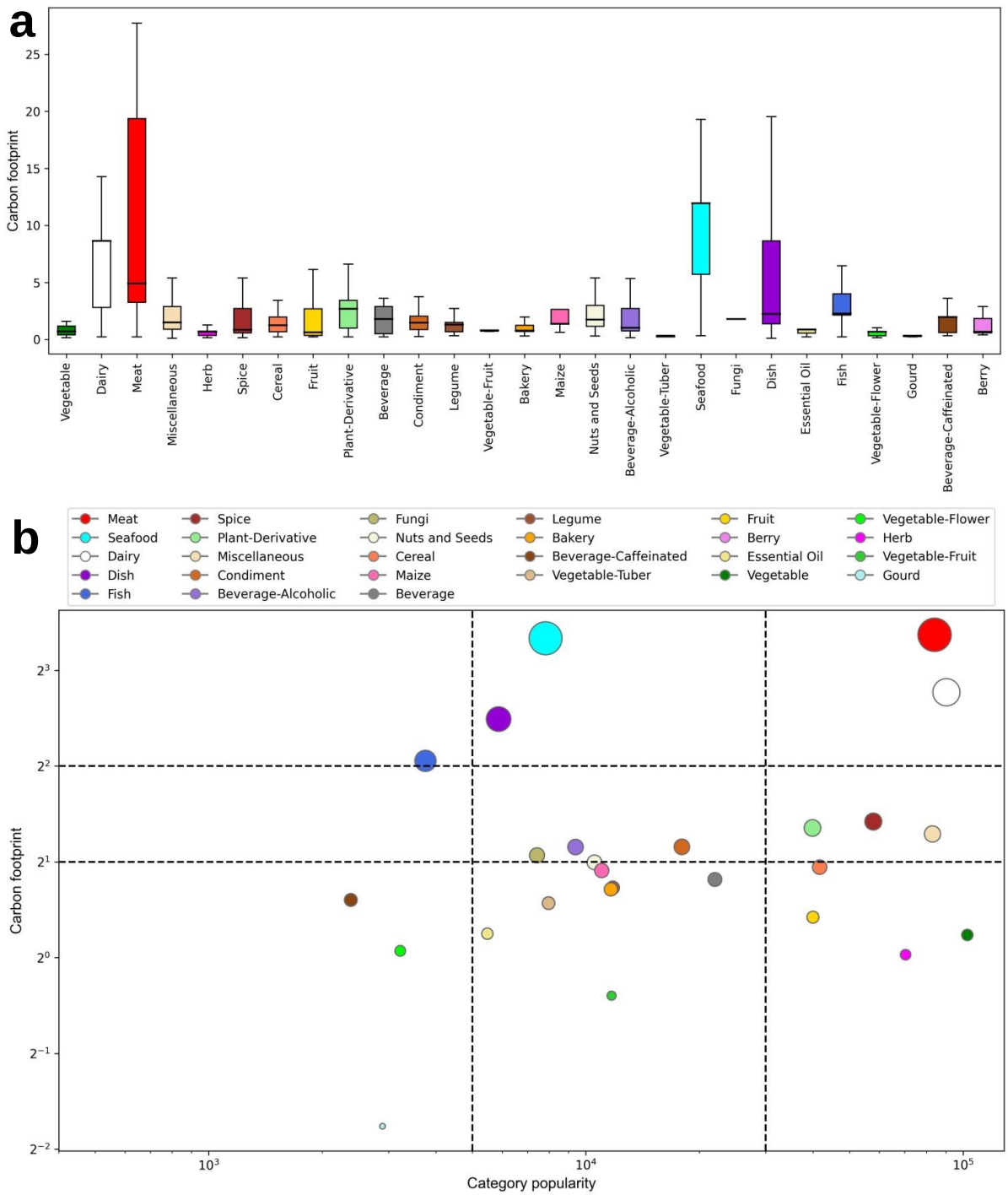

Figure 7.5: Carbon footprint analysis for ingredient categories. (a) CF statistics of ingredient categories (ordered according to decreasing number of ingredients). The box plot shows the median, first and third quartiles, and min-max of data. (b) Environmental impact of various ingredient categories and their popularity in culinary preparations. The bubble size is proportional to the average CF of ingredients in the category.

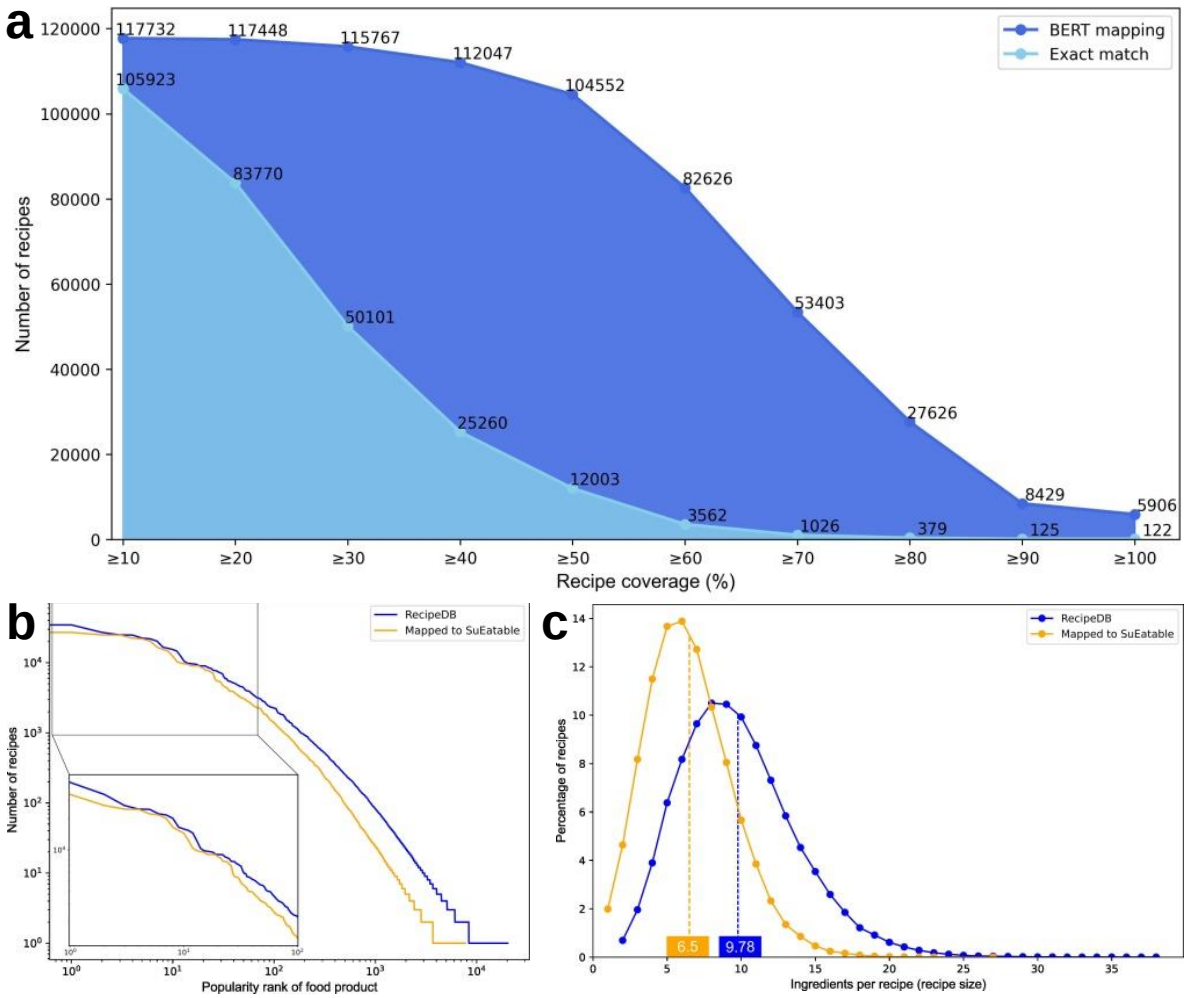

Figure 7.6: Effectiveness of the BERT-based strategy for mapping ingredients from recipes to their carbon footprints. (a) Recipe coverage statistics present the number of recipes with a coverage above a threshold. (b) Comparative statistics show that very few of the most popular ingredients were left out of the mapping protocol. The inset zooms in on the top 200 ingredients to highlight the success of the BERT-based mapping. (c) Recipe size distribution, before and after mapping. The average values are highlighted for each of the distributions.

7.3.4 Estimating the carbon footprint of recipes

Recipes are the cultural capsules that dictate dietary consumption. Beyond the level of ingredients and their categories, the actual environmental impact of food is better assessed by estimating the carbon footprint of recipes, the basic units of dietary consumption, the metaphorical culinary currency. By mapping ingredients in recipes to their global average carbon footprint values, we estimated a ballpark figure of their carbon footprint. 'Recipe Coverage' captures the proportion of all the recipe ingredients mapped with a SuEatable food item. A 100% recipe coverage suggests none of its ingredients were left out in the process of mapping and, hence, is the ideal situation when estimating the recipe's carbon footprint. Good recipe coverage leads to

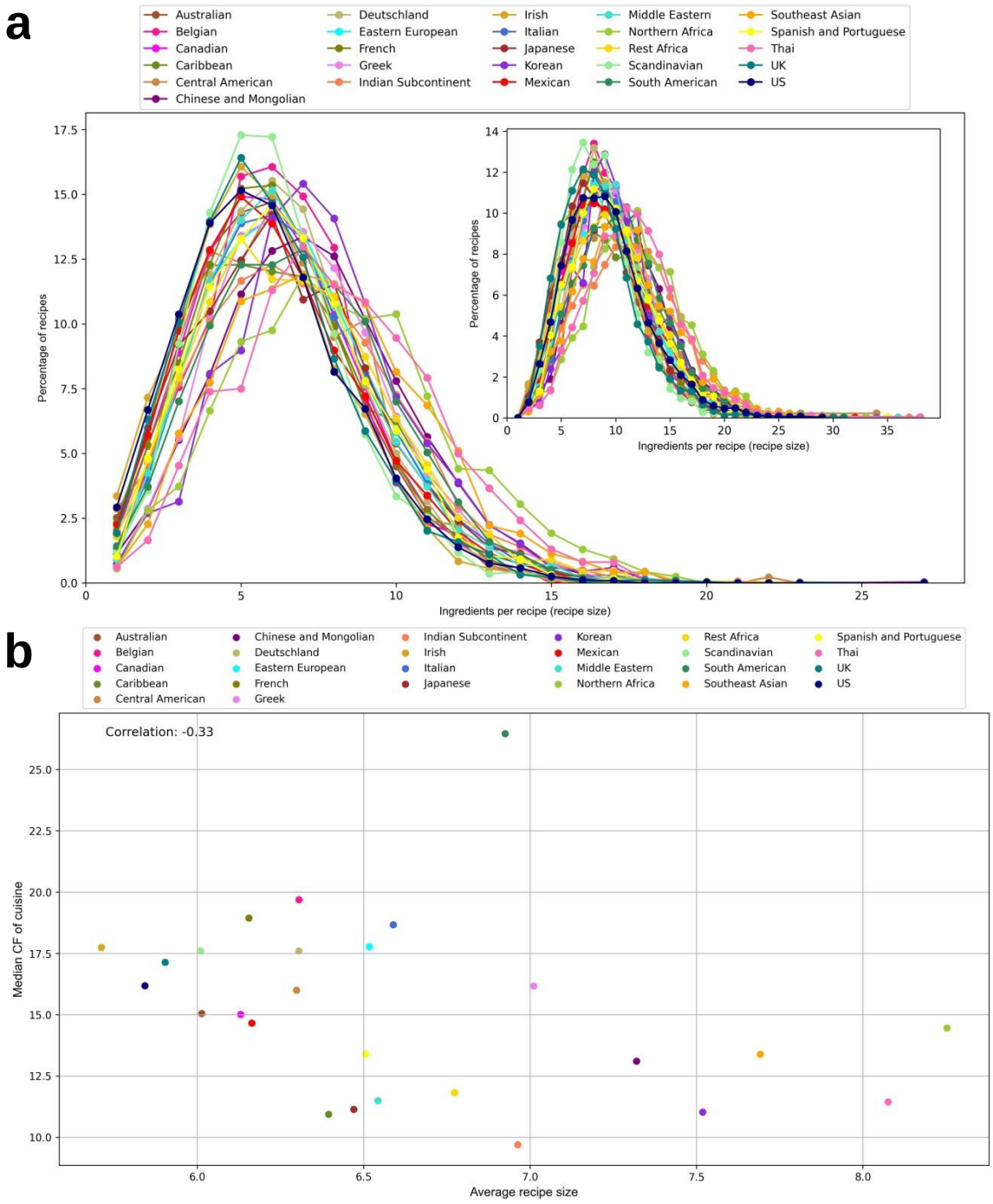

Figure 7.7: Recipe size statistics and their association with median carbon footprint of cuisines. (a) These statistics present the recipe size distribution of cuisines before mapping ingredients (inset) and after BERT-based mapping. The latter shows reduced average recipe sizes, as SuEatable does not have carbon footprint value for every RecipeDB ingredient. (b) Importantly, the average recipe sizes are comparable across the world cuisines and do not show a correlation with their median carbon footprint (CF) values.

reasonable CF estimates.

Using exact match and BERT-based mapping protocol (as discussed earlier in the section on

'Integration of SuEatable & RecipeDB,' Figure 7.1c), we achieve good coverage across all the recipes (Figure 7.6a). Most recipes (99.62%) could be included in the analysis with a lenient recipe coverage expectation of $\geq 10\%$. Even with a stringent recipe coverage cutoff of $\geq 50\%$, a significant number of recipes were accounted for (88.47%), indicating the effectiveness of the strategy implemented for mapping the ingredients. Given the heterogeneous distribution of the ingredient popularities, it is pertinent that the mapping protocol does not miss the most frequently used ingredients (Figure 3b). The BERT-based embedding maps 71% and 67.5% among the top 100 and 200 most popular ingredients, respectively (Figure 7.3). On average, half of all ingredients in each recipe were matched with their carbon footprint while primarily missing infrequently used ingredients (Figure 7.6c).

7.3.5 Comparing carbon footprints of cuisines

The cuisine-level CF statistics present an interesting picture of the environmental impact of culinary practices from various geo-cultural regions (Figure 7.10a). The carbon footprint of all the recipes across the 26 cuisines (World) varies between 0.15 and 166.019, with a median CF of 15.64. For example, among the recipes with low carbon footprints are the Mexican dish 'Cilantro-Lime Rice' and the French dish 'Sobronade' with estimated CF of 0.44 and 1.88, respectively. Towards the higher end of the spectrum are the Italian dish 'North End Sunday Gravy' (116.84) and the US dish 'Ring of Fire Chili' (166.02).

Dishes such as the Chinese 'Rosemary Rice' (17.21) are in the middle of the spectrum. Cultural practices that influence the idiosyncratic ingredient usage patterns render recipes in some cuisines more environmentally harmful than the global average. Among the cuisines with higher median CF than the global median are South American, Belgian, French, Italian, Eastern European, Irish, Scandinavian, Deutschland, UK, US, Greek, and Central American. On the other hand, ingredient combinations in the recipes of the Indian Subcontinent, Caribbean, Korean, Japanese, Thai, Middle Eastern, Rest Africa, Chinese and Mongolian, Southeast Asian, Spanish and Portuguese, Northern Africa, Mexican, Canadian, and Australian cuisines render them environmentally less harmful with a lower median CF lower than the global value.

Probing deeper into the CF distribution of recipes in each cuisine presents better insights (inset of Figure 7.10b). Carbon footprint distribution indicates a bounded, Gaussian-like distribution for all 26 cuisines. Most cuisines had high CF recipes above the global average. The cumulative CF distribution helps discern the cuisine-to-cuisine differences (Figure 7.10b). Recipe size cannot explain these differences, as cuisines have comparable recipe sizes on average (Figure 7.7). Interestingly, the observed differences in the carbon footprints of the cuisines are best explained by the heavy use of high-CF food products (Figure 7.8, Appendix B Figure B.1, Appendix B Figure B.2, and Figure 7.9).

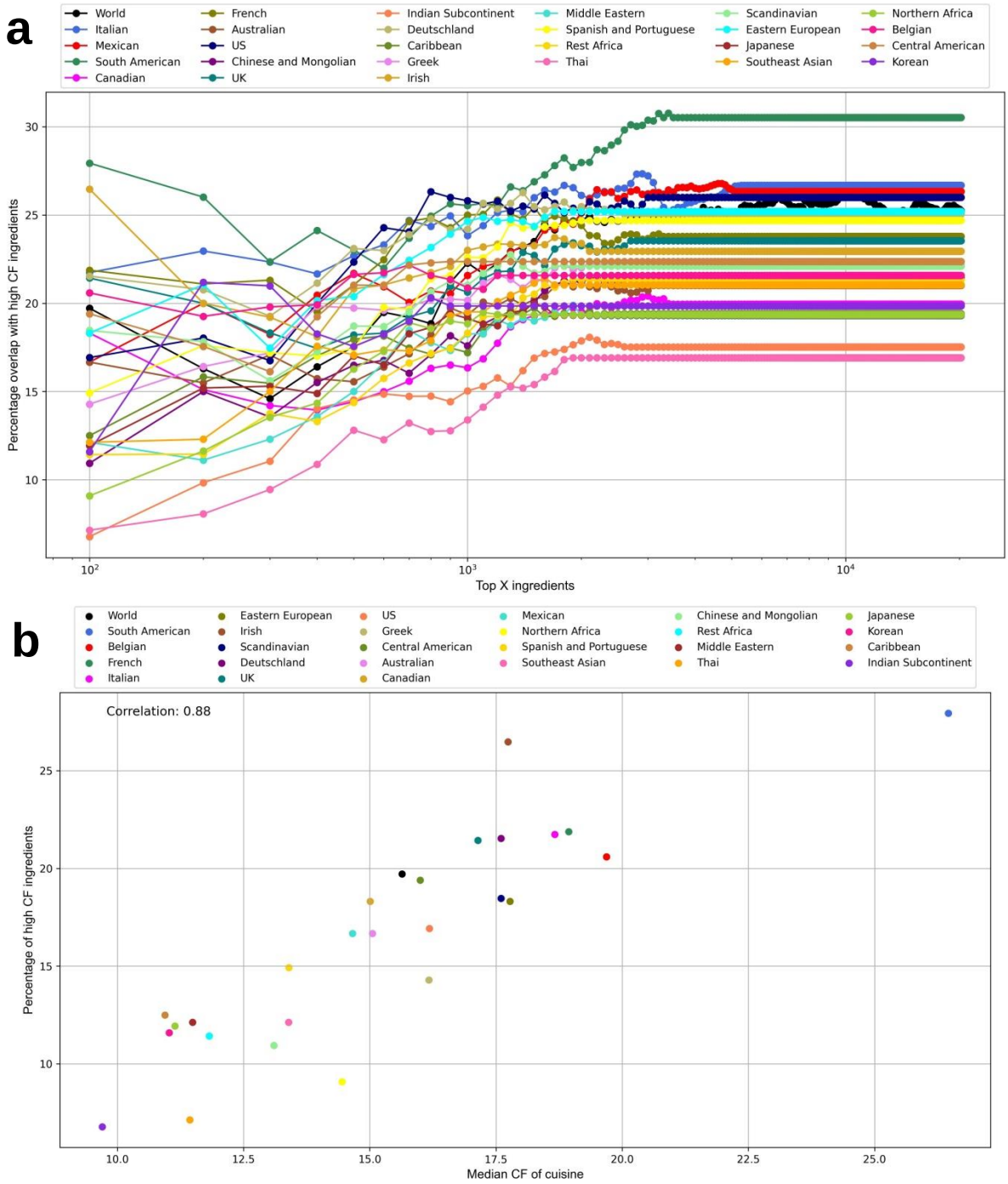

Figure 7.8: Overlap of Top X RecipeDB ingredients with High CF (CF>4) ingredients. (a) The number of shared ingredients between the top most popular food products and those having a severe environmental impact (CF>4). Clearly, the cuisines get segregated based on how many of the most frequently used ingredients in their recipes are environmentally unsustainable. (b) The median CF of cuisines shows a high correlation (0.88) for the Top 100 most frequently used ingredients. Please see Figure B.1 and Figure B.2 in 'Appendix B' for the correlation of median CFs with the Top 1000 and Top 10,000 ingredients, respectively.

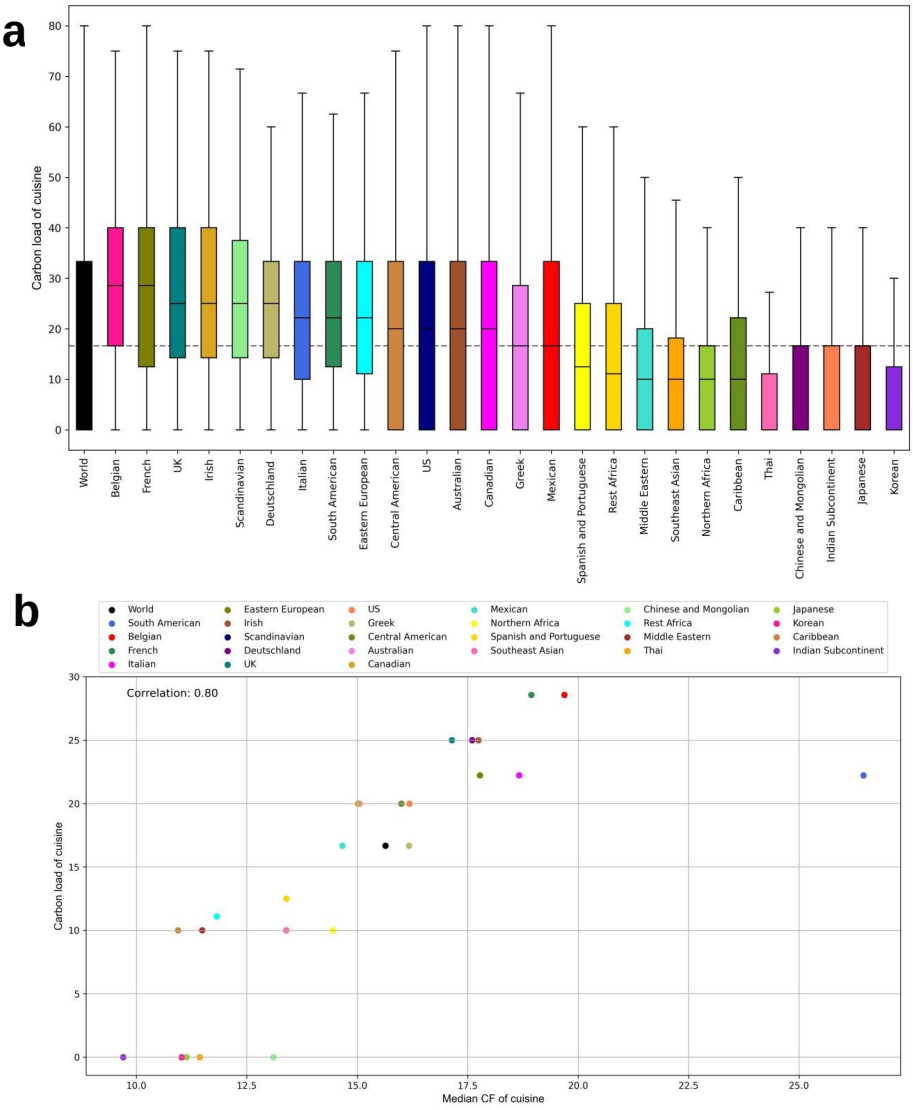

Figure 7.9: Carbon load of cuisines. The percentage of high CF ingredients in a recipe is defined as its 'carbon load.' Accordingly, averaging across all the recipes in the cuisine will yield the carbon load of the cuisine. (a) Carbon load of cuisines. The dotted grey line represents the global median. The global average of recipes with high CF ingredients is 19.95% (Median: 16.66%, Standard deviation: 19.18%). The percentage of recipes in Belgian, French, UK, Irish, Scandinavian, Deutschland, Italian, South American, Eastern European, Central American, US, Australian, and Canadian regions are higher than the global median value. (b) The median CF of cuisines shows a high correlation (0.80) with the percentage of high CF ingredients in recipes.

7.3.6 Vegetarian and non-vegetarian recipes

Going beyond investigating the carbon footprints of food products at the level of ingredients, their categories, recipes, and cuisines, we further probed the sustainability of vegetarian and non-vegetarian recipes. The prevailing literature pins the onus of the observed carbon footprint of the food system primarily on animal-sourced products [186]. Heavy consumption of animal-derived

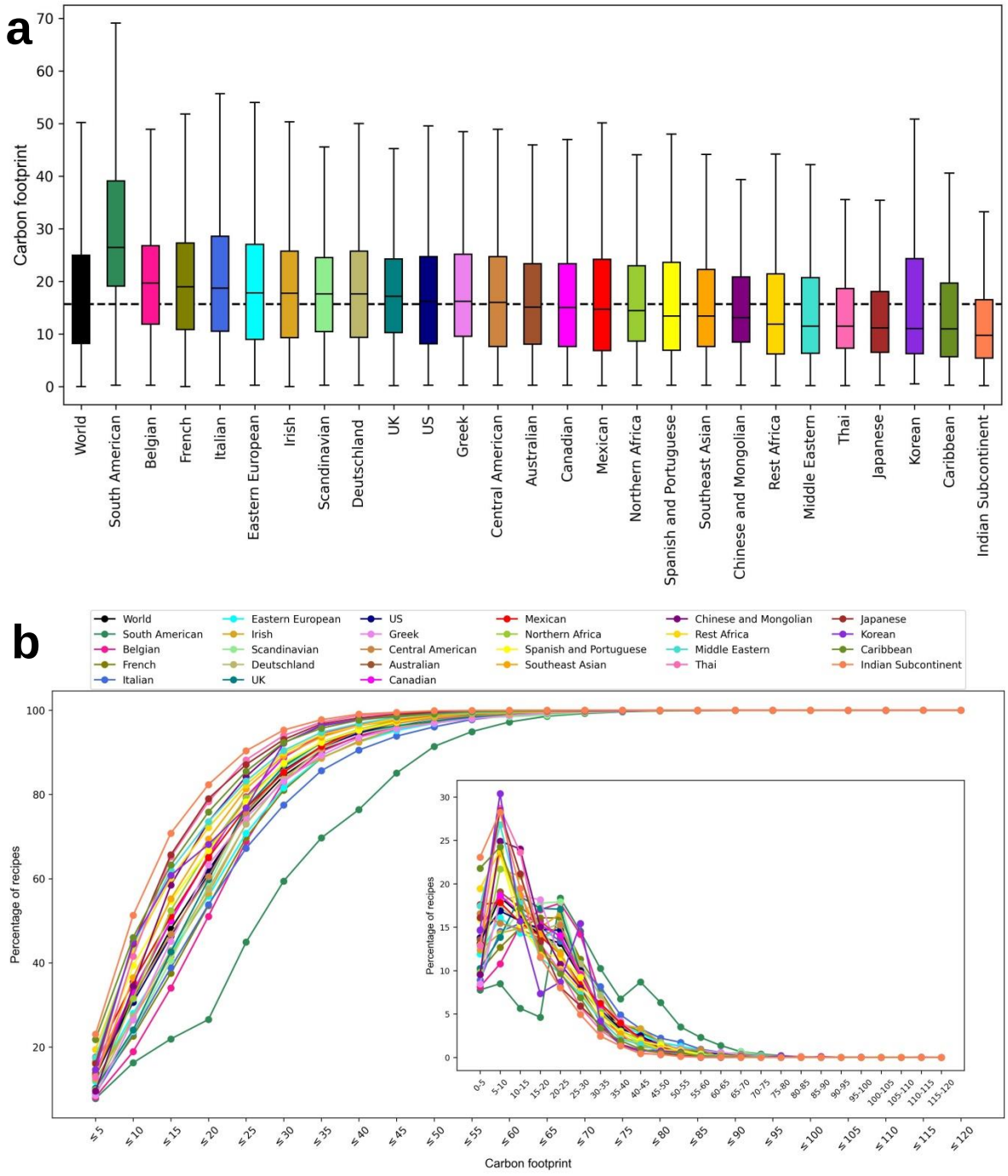

Figure 7.10: Comparison of recipe carbon footprints across world cuisines. (a) Recipe CF statistics across cuisines (ordered according to decreasing value of median CF). The box plot shows the median, first and third quartiles, and min-max of the data. (b) The cumulative distribution of recipe CF of cuisines helps better segregation of cuisines. The inset shows recipe CF distributions.

products could be one basis of such inference, other than inherent emissions linked to the product. Culturally influenced culinary differences shape the composition of recipes, dictating the use of animal products and other ingredients that have a bearing on the recipe CF. To probe the role of cultural influences in specifying the carbon load of recipes from various cuisines, we segregated

the recipes into those using animal-derived food products and those that do not. A recipe was classified as 'non-vegetarian' if it had one or more animal products of either the 'meat,' 'seafood,' or 'fish' category. Among the remaining recipes, those incorporating at least one ingredient from the following categories were labeled as 'vegetarian': vegetable, vegetable-tuber, fruit, plant derivative, legume, nuts, seeds, maize, and vegetable-flower. Miscellaneous recipes exclusively use ingredients that are neither from 'vegetarian' nor 'non-vegetarian' categories.

Overall, the collection of global recipes is dominated by non-vegetarian recipes, with 64.33% of culinary preparations incorporating at least one animal product (Appendix B Figure B.3). Some cuisines are significantly enriched with non-vegetarian recipes compared to the world average - South American, Italian, Greek, Central American, French, Irish, Belgian, Eastern European, UK, Scandinavian, and Deutschland. On the other hand, some others have a lesser proportion of recipes that include animal-sourced ingredients - Thai, Japanese, Indian Subcontinent, Chinese and Mongolian, Caribbean, Southeast Asian, Korean, Spanish and Portuguese, Rest Africa, Canadian, US, Australian, Middle Eastern, North Africa, and Mexican.

7.4 Carbon Footprint of Vegetarian and Non-vegetarian Recipes

The average carbon footprint of recipes that use meat, egg, fish, or seafood is 22.15 (Standard Deviation: 12.84) as compared to significantly lower (Kolmogorov-Smirnov Test) environmental impact of vegetarian recipes (10.638, Standard Deviation: 7.94) devoid of such animal-derived ingredients (Figure 7.11a). Kolmogorov-Smirnov test (with an alpha value of 0.05) was implemented to find the statistical significance of the difference between the carbon footprints of vegetarian and non-vegetarian recipes. The null hypothesis was, 'carbon footprint distributions of vegetarian and non-vegetarian recipes are indistinguishable.' A consistent pattern across all 26 cuisines suggests that cultural influences render non-vegetarian recipes environmentally unsustainable (Figure 7.11b). Among the cuisines with a carbon load of non-vegetarian recipes above the global average are South American, Italian, Greek, Central American, and French. Mexican, Northern Africa, and Middle Eastern recipes are at par. Thai recipes have an exceptionally low carbon footprint. Among the other cuisines with recipes employing animal-sourced ingredients that are most environment friendly are Korean, Chinese, Mongolian, South American, Indian Subcontinent, Middle Eastern, Southeast Asian, and Northern Africa.

Idiosyncratic ingredient combinations in vegetarian recipes can make them environmentally untenable despite the exclusive use of plant-derived products. While not using animal products, counter-intuitively, the biased combinations of high CF plant ingredients leave some cuisines (Italian, Belgian, and UK) with a heavy carbon load of vegetarian recipes. Korean, Southeast Asian, Caribbean, Spanish and Portuguese, Thai, South American, Japanese, and Indian Sub-

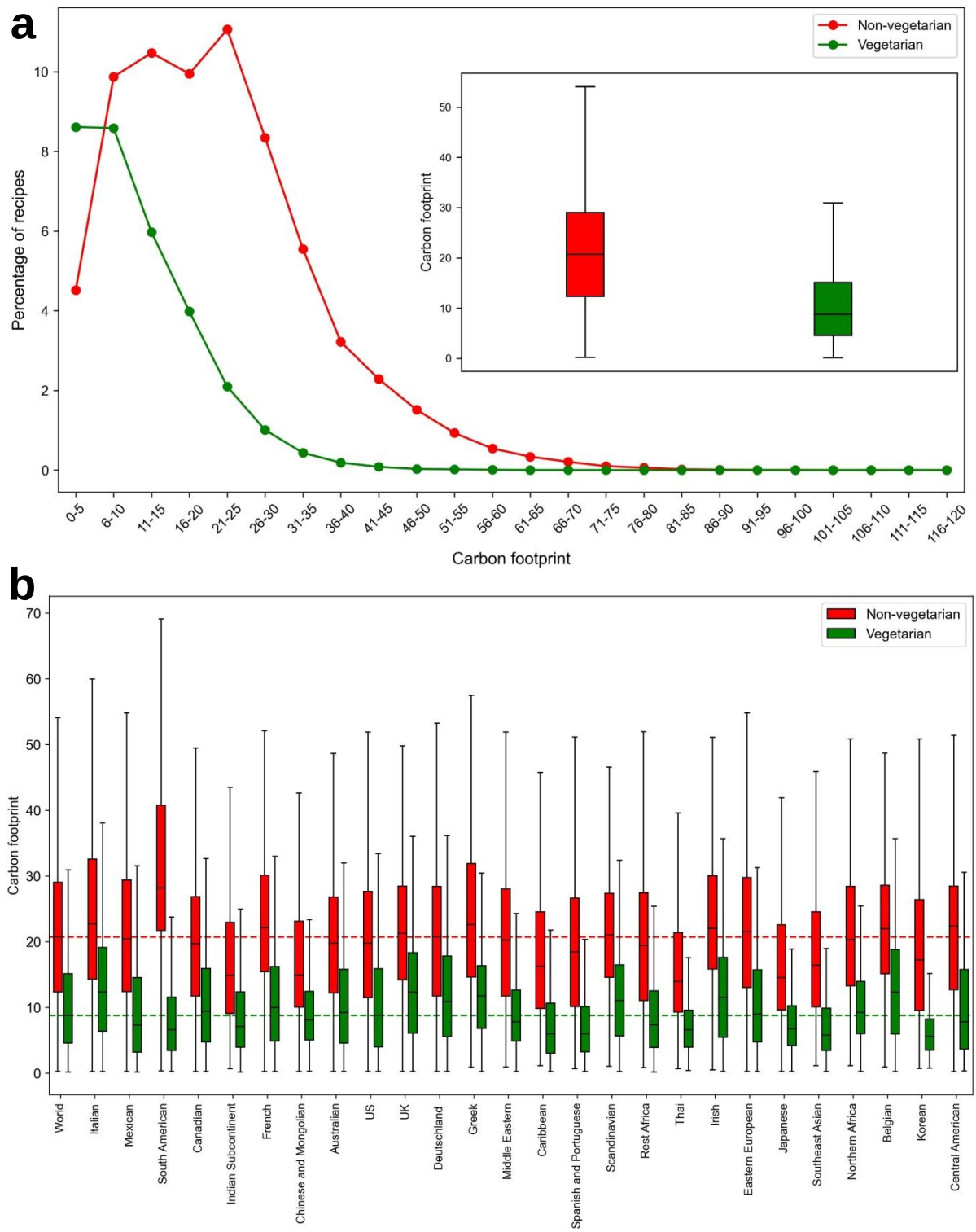

Figure 7.11: Comparison of the environmental impact of vegetarian and non-vegetarian recipes. (a) Carbon footprint distribution of vegetarian and non-vegetarian recipes (World). The comparative statistics of average carbon footprints. (b) Comparison of the median CF of the vegetarian and non-vegetarian recipes across the cuisines. The box plot shows the median, first and third quartiles, and min-max of the data.

continent cuisines are salient among those with the lowest footprint of vegetarian recipes, as expected.

7.5 Webserver Implementation

We have created a web server (SustainableFoodDB) to enable the exploration of the carbon footprints of recipes using their titles or ingredients used. The search page provides a vegetarian and non-vegetarian label of recipes other than their estimated carbon footprints. Recipe-specific pages provide a comprehensive picture along with the cuisine of the recipe, its constituent ingredients, and the availability status of their CF values. Ingredient-specific pages provide the source(s) of data [183] for their carbon footprints. The server also features a 'Carbon Footprint Calculator', which facilitates computation of estimated carbon footprints of recipes in a user-friendly manner when fed with the list of ingredient and their quantities. The web server implementation was done using a tech stack of ReactJS, CSS, Bootstrap, and MaterialUI for the front end and MongoDB, NodeJS, and ExpressJS for the back end. SustainableFoodDB: (https://cosylab.iiitd.edu.in/SustainableFoodDB/).

7.6 Discussion

Dietary choices are dictated by various intertwined factors, including taste specified by ingredient combinations [19, 21, 22, 26], nutrition [187, 169], cost, allergies [188], religion [189], and psychology [190]. These choices may have serious deleterious consequences for the environment due to the GHG emissions arising from the food system [167]. Culture, intertwined with religion and regional diversity, has a deep influence on the carbon footprint, especially in the context of food practices. Religious beliefs impact the dietary choices between plant-based and animal-based food. Regional diversity influences the availability and source of ingredients with unique climates and agricultural practices. Traditional cooking styles, deeply rooted in cultural practices, also contribute to the carbon equation. For a sustainable food ecosystem, we need concerted efforts to collect data on all aspects of the food system.

Quantification of the carbon footprint of dietary choices will enable mitigation of the adverse environmental impacts due to global warming and climate change [172, 168]. By blending the state-of-the-art carbon footprint data of food products [183] and recipe composition [6], we systematically investigate the carbon load of traditionally consumed recipes and compare cuisines for their environmental impact. We also present an extensive repository [191] of estimated carbon footprints of recipes from across the world. Among the key results, recipe composition has a strong bearing on its carbon load, and recipes using animal-based products tend to have a higher footprint than their plant-based counterparts [186]. More than 80% of

future warming from food consumption will be from meat, rice and dairy products (high-methane food groups) [168]. Therefore, focusing on reducing emissions from production, consumption and waste of these food groups can play a major role in avoiding carbon emissions associated with food consumption.

While this study captures the estimated carbon footprint of popularly consumed recipes by plugging in CF data from an extensive list of ingredients, going forward, it is desirable to account for the quantity of the ingredients used. Further, beyond the global estimates, the carbon footprint data needs to be at higher geospatial resolution. Various factors contribute to the footprint of food products, such as transportation mechanisms [175, 176] and consumption patterns [192]. The carbon footprint focuses primarily on greenhouse gas emissions [171], and many factors, such as water usage, land degradation, and resource depletion, may not be fully accounted for.

With increasingly nuanced food labels introduced by regulatory bodies, carbon footprint values could potentially be used to highlight the environmental impact, along with the nutrition and allergy indications. Among the most exciting directions in artificial intelligence is emulating culinary creativity using large language models to generate novel recipes [81]. Data-driven strategies can, therefore, be leveraged to generate hitherto unseen recipes that are not only tasty but also environmentally sustainable. The adverse impacts of animal-sourced ingredients indicate that efforts toward producing industrial-scale plant-based meat and lab-grown meat are timely and relevant.

Livestock management is one of the dimensions with much scope for implementing strategies to reduce emissions such as carbon sequestration via improved pasture management and better livestock integration in the circular bioeconomy [193]. Reducing the CF of the food system is a much broader challenge and would require systemic change in dietary behaviors, adoption of energy-efficient technologies, transition to renewable energy sources, mechanisms for food waste decarbonization [194], and innovations in agricultural practices, food processing techniques, and transportation technologies. Aligned with the idea of computational gastronomy [5], quantification of carbon footprints of recipes through the lens of cultural context will go a long way in steering the food system towards sustainability.

Chapter 8

Mining Culinary Patterns to Differentiate Global Cuisines

8.1 Introduction

Every cuisine is imprinted with a culinary fingerprint characterized by the unique combination of ingredients, preparation techniques, and flavor profiles that collectively define its culinary heritage [2]. The globalization of food cultures has led to increased exchanges of culinary styles and techniques [195]. As ingredients and culinary protocols travel across geographical and cultural borders, the boundaries between cuisines become increasingly fluid, offering challenges and opportunities for classifying culinary styles. In this context, finding idiosyncratic patterns that define cuisines presents as an interesting challenge. Machine learning and natural language processing can be used to analyze recipe data systematically, enabling researchers [196, 197] to explore how ingredient usage varies across cuisines and predict regional classifications based on recipe composition. Cuisine classification is an important problem within the realm of computational gastronomy [62, 5], where the goal is to identify the culinary origin of a dish based on its ingredients, preparation techniques, and the tools used. Correctly identifying the cuisine of a recipe is not only important for preserving culinary heritage but also for personalizing food recommendations, managing dietary restrictions, and enhancing user experiences in the culinary industry.

The study of culinary data, including recipes, ingredients, and associated nutritional information, is a growing field in the context of culinary and data science. Researchers have explored the relationships between ingredients and cultural patterns, nutritional profiling, and ingredient substitution [62, 5]. The use of computational techniques to analyze recipes has paved the way for applications in ingredient prediction, recipe recommendation, and understanding culinary diversity across cuisines.

Analyzing ingredient patterns to uncover culinary relationships has been a focal point in recent research. Teng et al. [198] explored flavor networks and food pairings to understand ingredient compatibility across cultural recipes. They employed network analysis to model the connections between ingredients and their shared flavors, highlighting that Western and Asian cuisines differ significantly in ingredient pairing patterns. Such studies show that ingredient relationships are culturally influenced, and they lay the groundwork for computational analysis of recipe data, such as ingredient prediction and culinary recommendation systems [199, 200]. Another research [201, 37, 104] developed machine-learning methods to suggest ingredient substitutes

based on the ingredient similarity. These studies contribute to improving recipe recommendation systems, allowing personalized dietary recommendations and ingredient alternatives in various cuisines. The dataset in this study, which incorporates ingredient masking, aligns with this research by enabling experiments that examine how ingredient substitutions affect recipe classification and culinary prediction.

Nutritional profiling in recipes has become important for understanding dietary impacts and making health-based recommendations [103, 202]. Studies have shown that by analyzing the macro- and micro-nutrient composition of ingredients, it is possible to create healthier recipe recommendations. Fallaize et al. [203] used food databases and machine learning models to predict nutritional quality and dietary compliance in recipes. The integration of nutritional and culinary data in recipe databases enables more comprehensive models that predict not only flavor and ingredient compatibility but also potential health benefits and dietary adherence.

In recent years, various machine learning approaches have been applied to classify recipes based on their ingredients [197, 198, 204, 196]. Such approaches involve transforming text-based ingredient lists into numerical vectors, which can then be used for predictive modeling. The use of vectorization techniques, such as Term Frequency-Inverse Document Frequency (TF-IDF) and CountVectorizer in text classification tasks [205]. These vectorization techniques, alongside label encoding for categorical variables, help transform recipes into structured data that machine learning models can interpret. Advanced algorithms like XGBoost [206] and LightGBM [207] are widely recognized for their efficiency in multi-class classification tasks, including recipe categorization by cuisine or region. Traditional classifiers, such as Support Vector Machines (SVM), indicate that ensemble models often outperform in tasks involving high-dimensional, textual dataset. Artificial Neural Networks (ANNs) have also been applied in recipe classification [208, 209], particularly for tasks requiring deep pattern recognition [210, 211]. Research by LeCun et al. [205] demonstrated that neural networks could capture complex relationships in unstructured data, including textual descriptions of ingredients and cooking methods. Studies using ANN architectures with dropout layers and activation functions like ReLU show that these models can generalize well across large, diverse datasets, making them valuable for culinary data analysis.

In this study, we aim to develop a machine learning-based classification model capable of accurately identifying the cuisine of a recipe based on its constituent elements: ingredients, utensils, and cooking processes. We explore the relationships between these factors and examine their impact on the model's performance. The task of classifying cuisines based on ingredients is complex, as many ingredients are shared across different cuisines, and subtle variations in preparation or ingredient combinations can be significant. To address this, we employ advanced NLP techniques to transform raw ingredient lists into structured data. By analyzing ingredient usage patterns, the types of utensils employed, and common cooking methods, our model is designed to uncover distinctive features that characterize specific cuisines.

We conducted various experiments, incorporating different machine learning algorithms and feature engineering techniques, to optimize the performance of our classification model. Our results show how well cuisine classification can be achieved through a deeper understanding of the interactions between ingredients, processes, and utensils, contributing valuable insights to the growing field of computational gastronomy. Figure 8.1 shows the detailed schema of our study.

Beyond the immediate goal of classification, this study contributes to broader applications in food science and health informatics. Understanding ingredient patterns can aid in the development of dietary recommendation systems, improve ingredient substitution models, and support nutritional profiling for various cuisines. Our findings have potential applications in personalized meal planning, cross-cultural recipe adaptations, and even predictive models for ingredient availability or substitution in regions with limited resources. Through this research, we aim to enrich the growing field of computational gastronomy, providing a foundation for future studies that explore the intersection of food, culture, and data science. By systematically analyzing ingredient usage and culinary structures, our study sheds light on how machine learning can capture the nuanced differences in global cuisines, creating possibilities for innovation in culinary arts and food technology.

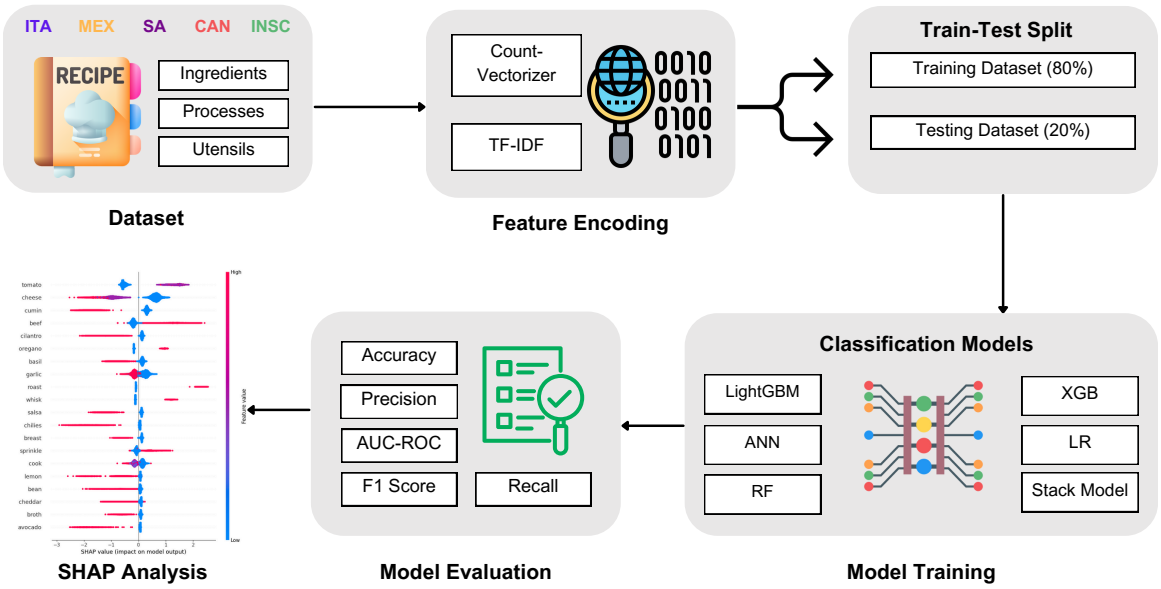

Figure 8.1: The schematic pipeline for cuisine classification using a dataset of the top five cuisines from RecipeDB: Italian, Mexican, South American, Canadian, and Indian Subcontinent. Features were extracted using CountVectorizer and TF-IDF, followed by splitting the dataset into training and testing sets. Classification models are implemented and evaluated based on metrics such as accuracy, precision, AUC-ROC, recall, and F1 score. SHAP analysis helps identify the important features that contribute to the model's performance.

8.2 Materials and Methods

8.2.1 Dataset

We have used the RecipeDB dataset [6], an extensive repository of 118,083 recipes and 20,280 ingredients representing a wide range of 26 cuisines from around the globe. The dataset contains culinary and nutritional information, which is crucial to our analysis. It has been curated explicitly for research purposes to examine ingredient relationships, culinary patterns, and potential nutritional insights across different cuisines. This dataset comprises several key columns, each representing essential recipe composition and preparation attributes. The dataset was initially collected from a large-scale online repository of global recipes. During the collection process, the focus was on ensuring diversity across cuisine types and ingredient lists. The dataset's diversity and clean structure make it well-suited for machine-learning tasks such as ingredient prediction, nutritional profiling, and cross-cultural recipe analysis.

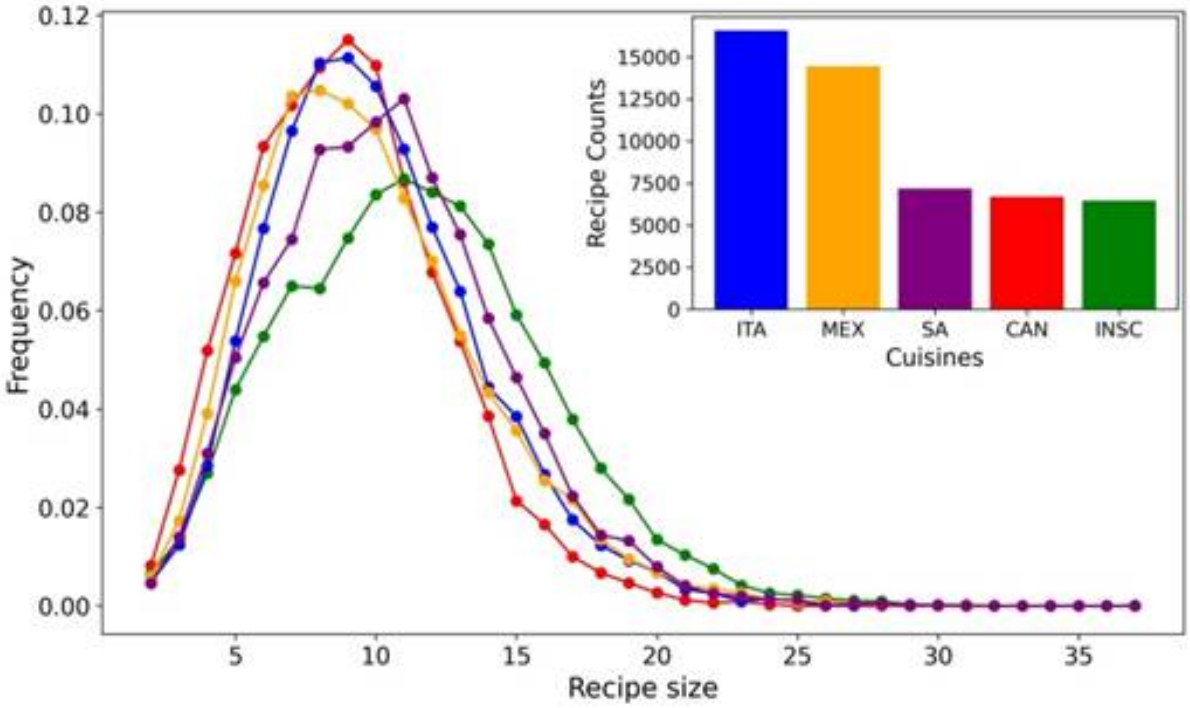

Figure 8.2: Cuisine-wise distribution of recipes with a Gaussian overlay to visualize the density of recipe counts. The main plot shows the distribution of recipes across different cuisines. The inset provides a detailed breakdown of the total number of recipes available for each cuisine.

We sampled the dataset of 26 cuisines in the Top 2, 3, and 5 cuisines. Due to class imbalance in the dataset of 26 cuisines (see Figure 8.2), we sampled the dataset into the top 5 cuisines (Italian (ITA), Mexican (MEX), South American (SA), Canadian (CAN), and Indian Subcontinent (INSC)). Figure 8.3 shows a significant drop in accuracy after the top 5 cuisines, limiting our focus on the top 5 cuisines. We were left with 51,349 recipes and 11,744 ingredients.

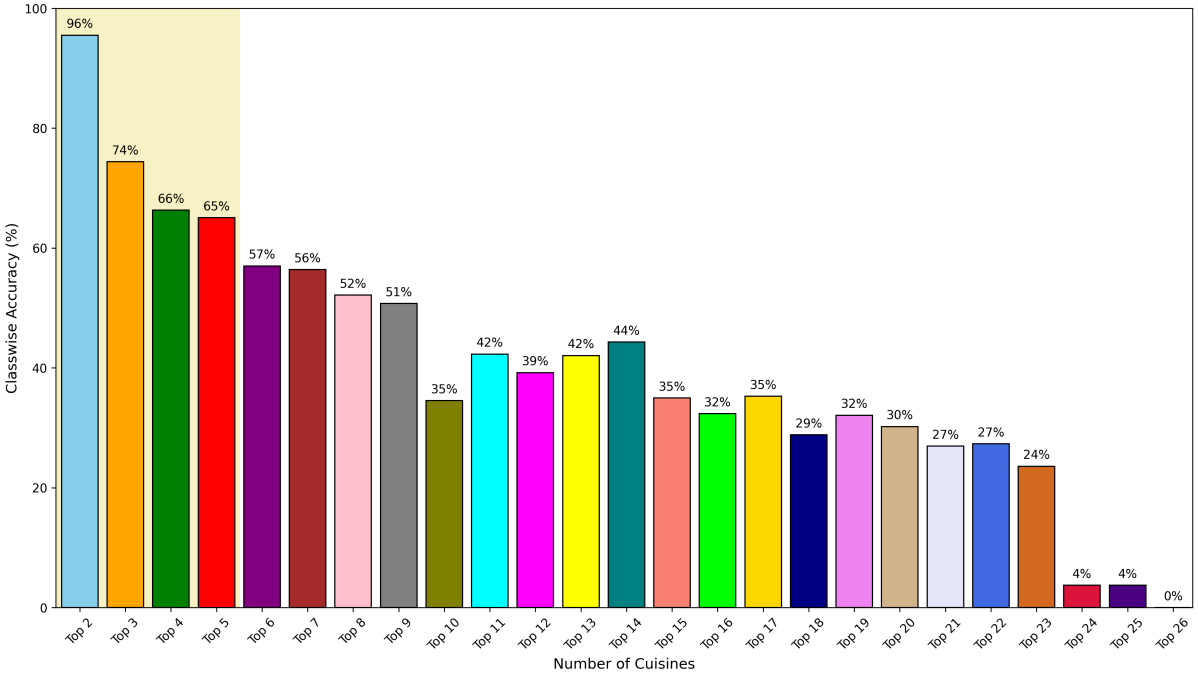

Figure 8.3: Classification accuracy for Top X cuisines. Starting from the Top 2, models were built to classify all the cuisines (Top 26). Top 5 cuisines (highlighted in yellow) were short-listed for the rest of the experiments based on the model performance.

8.2.2 Models impelementation

We split the dataset into features (X) and target labels (y), where X contained the ingredients, utensils, and processes, and y represented the 'Region' labels. Using CountVectorizer, we transformed the text features into a numerical format, limiting the feature set to the top 10,000 words to maintain computational efficiency and mitigate overfitting. Subsequently, we encoded the 'Region' labels into a numerical format using LabelEncoder. We split the dataset into training and validation sets, an 80:20 ratio, to evaluate the model's performance. This split ensured that our model could be tested on unseen data, providing a realistic estimate of its performance. We implemented several classification models for a multi-class classification task, including XGBoost (XGB), LightGBM (LGBM), Logistic Regression (LR), Random Forest (RF), Artificial Neural Network (ANN), and a stack model consisting base models (XGB, LGBM) and a meta model (LR).

For XGBoost, we used the 'multi' objective to handle multiple classes and trained the model on the training dataset, evaluating its performance on the validation set. We monitored the multi-class log-loss (mlogloss) during training and applied early stopping with patience of 10 rounds to prevent overfitting. The LightGBM model was configured with the 'gbdt' boosting type and 'multiclass' objective, aligned with the unique 'Region' labels. Similar to XGBoost, early stopping based on multi-class log-loss (multi_logloss) was implemented to prevent overfitting. A multinomial Logistic Regression classifier was configured with the 'saga' solver for

efficient convergence on large datasets. The training was performed with random_state fixed at 42 for reproducibility. The Random Forest classifier was used as a tree-based ensemble model, configured with 100 trees (n_estimators=100) and a fixed random_state (42). The model demonstrated robustness and interpretability for feature importance analysis. After training all models, predictions on the validation set were decoded back to the original 'Region' labels using LabelEncoder for evaluation.

We developed an Artificial Neural Network (ANN) with an input layer of 1024 neurons using ReLU activation, a dropout layer with a 50% dropout rate to prevent overfitting, a hidden layer of 512 neurons with ReLU activation, a second 50% dropout layer, another hidden layer with 256 neurons and ReLU activation, a final dropout layer with a 50% dropout rate, and an output layer with neurons corresponding to the unique 'Region' labels, using softmax activation to predict class probabilities. The model was compiled with sparse categorical cross-entropy as the loss function, the Adam optimizer, and accuracy as the evaluation metric. Early stopping was implemented to monitor validation loss and restore the best weights if no improvement was observed for ten epochs. We trained the ANN for a maximum of 100 epochs with a batch size of 128. The training was stopped early when validation loss failed to improve for ten consecutive epochs. After training, the model was evaluated on the validation set. To gain further insights, predictions were generated for the validation set and decoded back to the original 'Region' labels.

8.2.3 Evaluation metrics

We evaluated the performance of our recipe classification models using a variety of metrics to gain comprehensive insights. Precision measured the proportion of correctly classified recipes among all predictions made for a specific class. A high precision score indicated the model's ability to minimize false positives effectively, ensuring that recipes classified as belonging to a particular region genuinely belonged to that region. Recall, on the other hand, evaluated the model's ability to identify all recipes belonging to a specific class, reflecting the proportion of actual positives correctly identified by the model. A high recall score was particularly valuable when it was crucial not to miss recipes from certain regions, ensuring inclusivity in the classification process. The F1-score, harmonizing precision and recall into a single metric, provided a balanced measure of the model's performance, especially for imbalanced datasets where certain regions had significantly more recipes than others. By combining these metrics, we obtained a comprehensive understanding of the model's ability to balance accurate predictions and inclusivity while maintaining overall performance.

In addition to these metrics, we calculated the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) to evaluate the model's ability to distinguish between classes. A high AUC-ROC score indicates that the model performs well across different thresholds, ef-

Model	Method	Accuracy	Recall	F1 Score	Precision
LightGBM	Count-Vectorizer	85.18	85.18	85.04	85.01
ANN	Count-Vectorizer	84.86	84.86	84.58	84.58
LightGBM	TF-IDF	84.83	84.82	84.79	84.69
ANN	TF-IDF	84.75	84.73	84.61	84.55
XGB	Count-Vectorizer	84.56	84.23	84.13	84.17
XGB	TF-IDF	84.28	83.90	83.77	83.78
Stack Model (XGB, LGBM, LR)	Count-Vectorizer	83.97	83.95	83.93	83.95
LR	TF-IDF	83.90	83.90	83.73	83.71
LR	Count-Vectorizer	83.87	83.87	83.73	83.71
Stack Model (XGB, LGBM, LR)	TF-IDF	83.79	83.80	83.75	83.72
RF	Count-Vectorizer	83.59	83.59	83.03	83.33

Table 8.1: Performance of classification models using the ingredients as a feature vector. LightGBM achieves the highest accuracy of 85.18% with CountVectorizer, followed by ANN with 84.86%. The results underline the effectiveness of advanced models like LightGBM and ANN in handling text-based features for multi-class classification tasks.

fectively discriminating between recipe categories. This metric was particularly insightful for understanding the trade-offs between sensitivity and specificity in multi-class classification.

8.3 Results

8.3.1 Ingredient as a feature vector

The evaluation of recipe classification models based on the ingredient feature vector indicates that LightGBM achieved the best performance when used with the CountVectorizer, reporting the highest accuracy of 85.18%, along with a recall of 85.18%, precision of 85.01%, and an F1-score of 85.04%. This highlights its robustness and capability to handle the text-based ingredient features effectively. Following LightGBM, Artificial Neural Network (ANN) with CountVectorizer achieved an accuracy of 84.86% with similar recall and slightly lower F1-score and precision. Models generally perform slightly better with CountVectorizer compared to TF-IDF, suggesting that the raw frequency-based representation captures sufficient information for classification tasks. Table 8.1 shows the detailed performance of classification models using the feature vector of ingredients.

8.3.2 Pairwise feature vectors

We evaluated the performance of recipe classification models using ingredients and utensils as a feature vector. Table 8.2 highlights each model's accuracy, recall, F1-score, and precision. LightGBM with Count-Vectorizer achieved the highest accuracy of 85.13%, recall (85.12%),

Model	Method	Accuracy	Recall	F1 Score	Precision
LightGBM	Count-Vectorizer	85.13	85.12	85.07	85.01
ANN	Count-Vectorizer	84.21	84.21	84.01	84.17
LightGBM	TF-IDF	84.12	84.12	84.01	84.20
ANN	TF-IDF	84.00	84.00	84.01	83.98
Stack Model (XGB, LGBM, LR)	Count-Vectorizer	83.94	83.94	84.58	84.60
XGBoost	Count-Vectorizer	83.92	83.92	83.86	83.88
Stack Model (XGB, LGBM, LR)	TF-IDF	83.89	83.89	84.26	84.28
XGBoost	TF-IDF	83.85	83.85	83.81	83.83
RF	Count-Vectorizer	82.70	82.70	82.63	82.66
LR	Count-Vectorizer	82.46	82.46	82.31	82.38
LR	TF-IDF	82.23	82.23	82.20	82.28

Table 8.2: Performance of classification models with the ingredients and utensils as feature vectors.

F1-score (85.07%), and precision (85.01%), suggesting balanced effectiveness in identifying recipe classes. Including utensils alongside ingredients improved the overall performance of most models. Utensils provide additional context, such as preparation methods, which contribute to better recipe classification.

Table C.1 in Appendix C highlights the performance of recipe classification models using ingredients and processes as a feature vector. LightGBM with Count-Vectorizer achieved the highest accuracy of 84.13% with recall, F1 score, and precision of 83.13%, 84.00%, and 84.44%, respectively. These results emphasize that combining ingredients with cooking processes provides additional context that enhances the model's ability to distinguish between recipes. We further evaluated the performance of recipe classification models using a combination of utensils and cooking processes as feature vectors (see Table C.2 in Appendix C). The ANN model paired with Count-Vectorizer delivered the highest accuracy of 47.56%, although the overall performance was relatively lower compared to feature vectors involving ingredients. This result suggests that while utensils and processes provide useful contextual information, their discriminatory power might be less robust when used without ingredient data.

8.3.3 Combined feature vector: ingredient-utensil-process

We evaluated the performance of recipe classification models using ingredients, utensils, and processes as a feature vector. By combining these elements, the feature representation encapsulates a more comprehensive understanding of recipes, including their components, required tools, and procedural steps. Table 8.3 highlights each model's accuracy, recall, F1-score, and precision. The LightGBM model with Count-Vectorizer yielded the highest accuracy of 85.60%, showcasing its ability to leverage the combined feature vector effectively. While the LightGBM model consistently outperformed other methods, the ANN model remains competitive, particularly for tasks involving more complex feature spaces. Models like XGBoost and Stack Models

Models	Method	Accuracy	Recall	F1 Score	Precision
LightGBM	**Count-Vectorizer**	85.60	85.60	85.46	85.46
ANN	Count-Vectorizer	85.25	85.25	84.78	84.80
LightGBM	TF-IDF	85.13	85.11	85.04	85.03
ANN	TF-IDF	84.89	84.89	84.77	84.78
XGBoost	Count-Vectorizer	84.75	84.73	84.70	84.68
XGBoost	TF-IDF	84.68	84.68	84.66	84.65
Stack Model (XGB, LGBM, LR)	Count-Vectorizer	84.67	84.67	84.58	84.60
Stack Model (XGB, LGBM, LR)	TF-IDF	84.36	84.36	84.26	84.28
RF	Count-Vectorizer	82.70	82.70	82.13	82.16
LR	Count-Vectorizer	82.52	82.52	82.50	82.51
LR	TF-IDF	82.45	82.45	82.38	82.41

Table 8.3: Performance of classification model with the ingredients, processes and utensils as feature vectors.

performed moderately well but lagged behind LightGBM and ANN in terms of accuracy and F1-score. Simpler models like LR and RF showed lower performance, suggesting that these approaches may not fully capture the richness of the combined feature vector.

Figure 8.4 presents the AUC-ROC (Area Under the Receiver Operating Characteristic) curves for models trained on different feature combinations derived from the dataset. The feature subsets include ingredients only, ingredient pairs (ingredient and process, ingredient and utensils), and combined features (ingredient, utensils, and processes). The results indicate that the combined feature set achieves the highest AUC-ROC score across all cuisines. However, only ingredients contribute the most to accurate recipe classification. This highlights the dominant role of ingredients in defining the unique characteristics of cuisines, as they directly reflect the flavors, textures, and cultural elements inherent to each recipe.

Figure 8.5 highlights the results of SHAP (SHapley Additive exPlanations) analysis, which provides insights into the contribution of individual features to the classification model's results. This approach ensures transparency in model decisions and helps identify the most influential features in the dataset. SHAP analysis reveals that ingredients are the most critical features driving the model's predictions. Ingredients such as tomato, cheese, cumin, beef, and cilantro are among the top contributors. These ingredients are strongly associated with specific cuisines—for instance, cumin and cilantro are often linked to Mexican and Indian recipes, while cheese to Italian and beef to South American cuisines (see Figure 8.6). Processes such as roasting, sprinkling, and cooking also emerge as important features, with a lesser influence than ingredients. These processes provide additional context about the preparation and cooking methods that are characteristic of certain cuisines. For example, roasting is commonly associated with South American recipes, while sprinkling is often indicative of finishing techniques used in various global cuisines (see Figure 8.7). The inclusion of utensils in the model, while not as impactful as ingredients or processes, still contributes to distinguishing cuisines (see Figure C.1 in Appendix C).

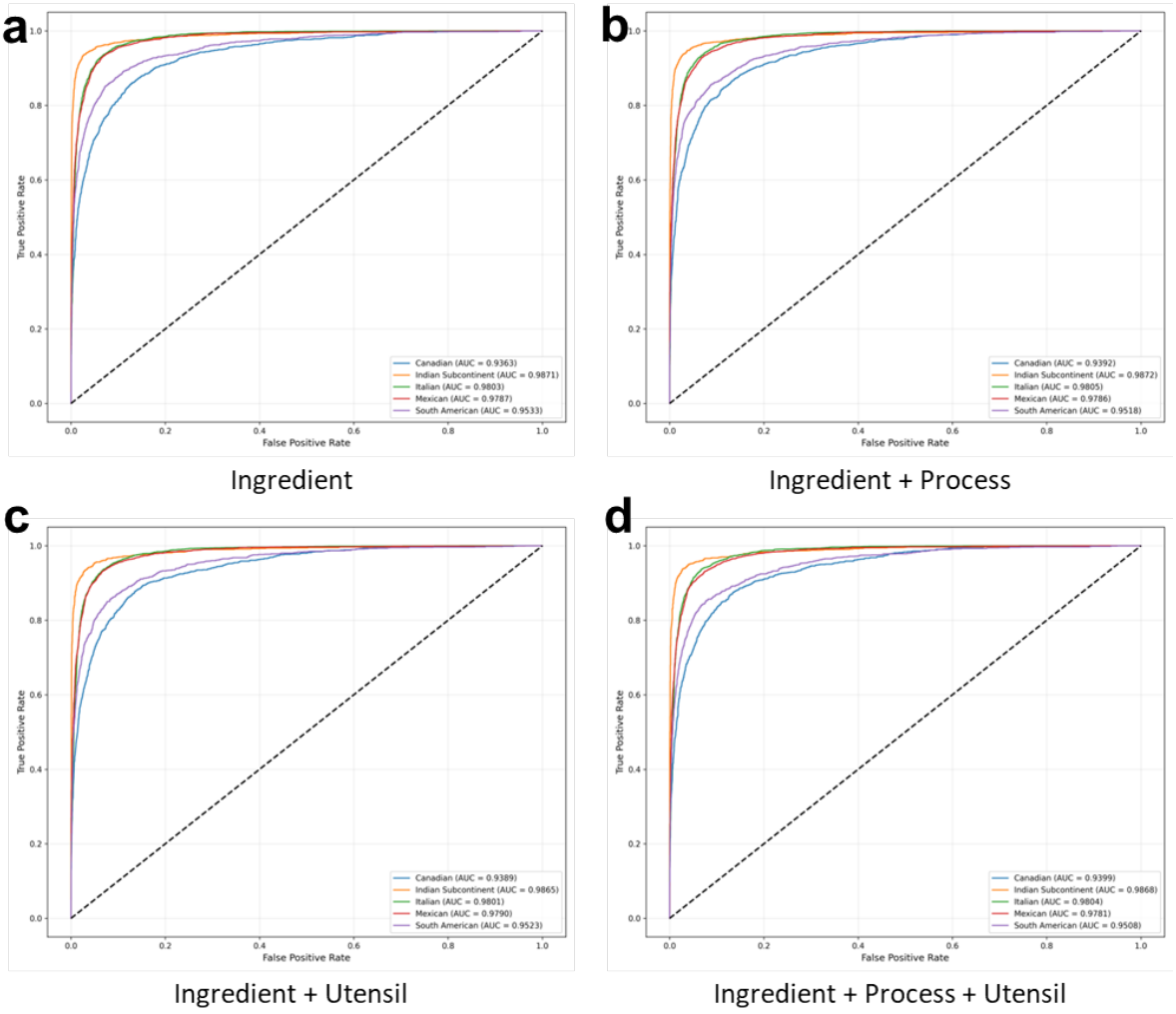

Figure 8.4: AUC-ROC plots for different subsets of datasets used in model training. (a) AUC-ROC for models using only ingredient features. (b) AUC-ROC for models using both ingredients and processes. (c) AUC-ROC for models trained with ingredients and utensils. (d) AUC-ROC for models utilizing all three feature types—ingredients, processes, and utensils.

8.4 Conclusions

In this study, we evaluated the performance of various machine learning models for recipe classification, focusing on the impact of ingredients, utensils, and cooking processes. We applied both CountVectorizer and TF-IDF to assess the performance of models including ANN, LightGBM, LR, XGBoost, RF, and a stack ensemble model. Our results demonstrate that the LightGBM model, particularly with CountVectorizer outperforms other models with high accuracy and F1 score. The performance of stack models and XGBoost varies based on vectorization, showing that choosing the right vectorization greatly impacts the results.

We observed that the ingredients showed a greater influence on classification accuracy than utensils and cooking methods. Even though incorporating characteristics from all three groups

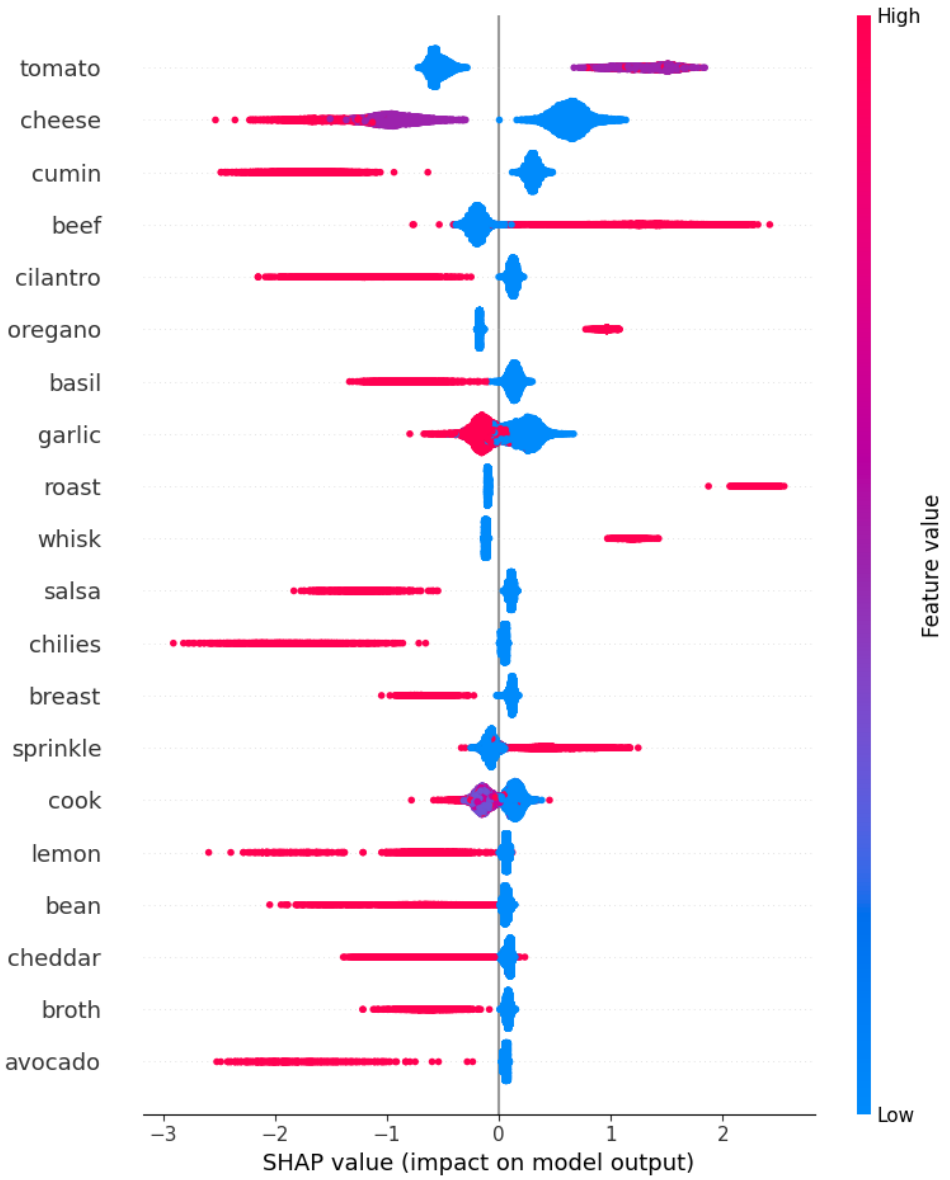

Figure 8.5: SHAP (SHapley Additive exPlanations) analysis results for the classification model trained on combined features (ingredients, utensils, and processes). The plot highlights the most important features contributing to cuisine classification. Ingredients such as tomato, cheese, cumin, beef, and cilantro are identified as top contributors, while processes like roasting, sprinkling, cooking, and certain utensils provide additional contextual information. These insights validate the model's features for accurate classification.

led to strong model performance, the ingredient information strongly impacts the predictions' accuracy, highlighting its crucial role in categorizing recipes. The study provides valuable insights for future research in culinary data analysis, with potential applications in recipe recommendation systems and automated cuisine categorization.

The limitations of this study involve limited feature scope as it includes ingredients, utensils, and cooking processes as features, neglecting factors such as cooking time, temperature, and

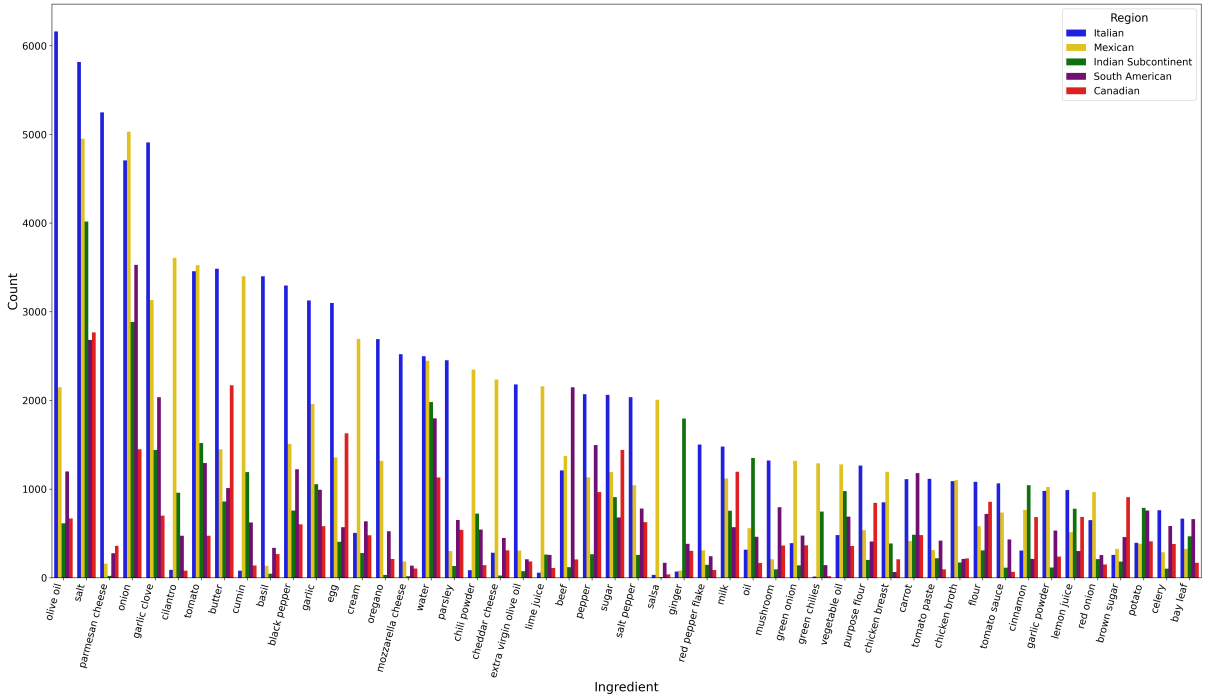

Figure 8.6: Distribution of popularly used ingredients based on the regions.

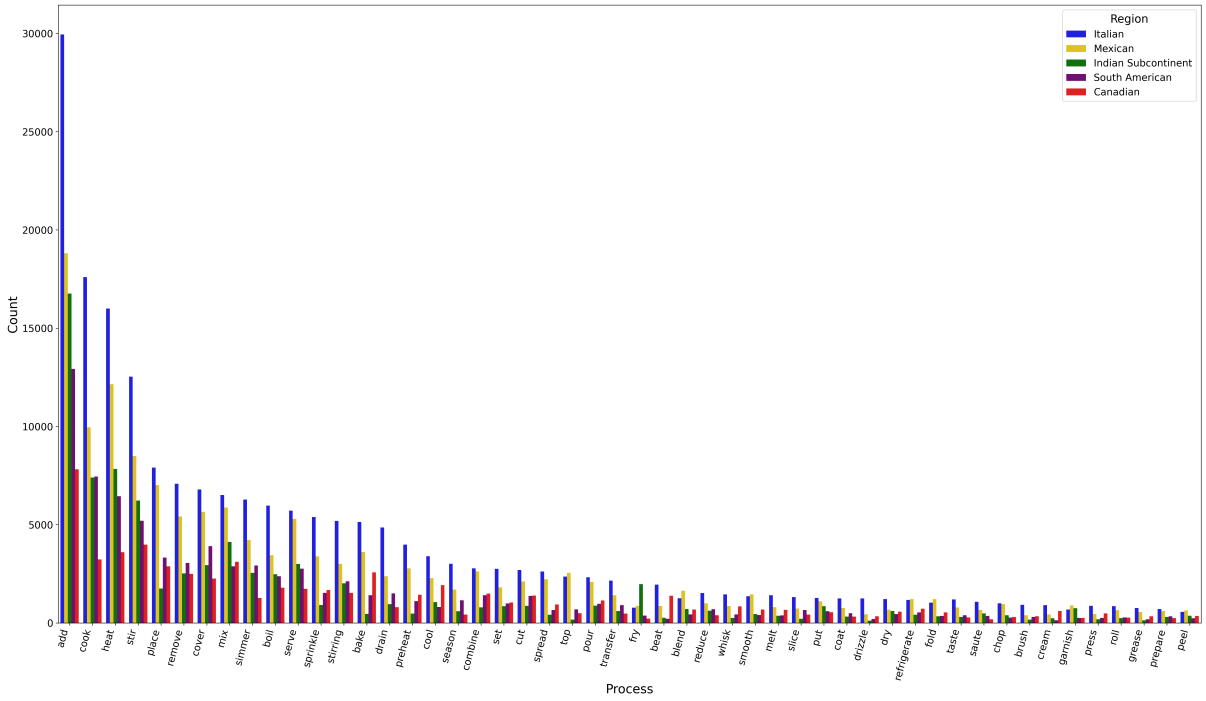

Figure 8.7: Distribution of popularly used processes based on the regions.

regional ingredient availability. These insights could enhance model accuracy further. The results indicate that ingredients had a more significant effect on accuracy than utensils and procedures regarding feature importance. This indicates that the role of tools and methods in cooking may not have been fully acknowledged, resulting in a possible bias towards recipes with a lot of ingredients. The dataset's size and diversity could limit the models' performance.

Chapter 9

FlavorDB2: A Database of Flavor Molecules

9.1 Introduction

Flavor is expressed through the interaction of molecules via gustatory and olfactory mechanisms. Knowing the pivotal role of flavor molecules in food and fragrances, the existence of a comprehensive repository becomes valuable. Such a repository encompasses the information on flavor molecules characterizing their flavor profile, chemical properties, regulatory status, consumption statistics, taste/aroma threshold values, reported uses in food categories, and synthesis. Flavor compounds form the physical basis for a wide range of gustatory and olfactory sensations experienced by humans [212]. These compounds evoke a complex taste and odor perception by interacting with the biological machinery. Flavor perception is an emergent property of the complex biological system, the complete understanding of which still eludes us [213, 214, 215, 216]. While the associations between molecular properties and sensory perception indicate the chemical basis of flavor [212], such knowledge is mainly heuristics-based and remains unstructured. This highlights the need for further exploration and refinement in flavor perception, pushing the boundaries of scientific inquiry in this fascinating field.

Creating a detailed catalogue of various aspects of flavor compounds offers a data-centric approach leading to a systemic perspective of flavor sensation. On one side, such a repository would have molecular descriptors enumerating the physicochemical properties of the flavor compounds. On the other side, it would have natural sources (in which these compounds are found), sensory features (such as flavor percepts), mechanisms of synthesis, regulatory status, and applications (such as food categories in which the compound is utilized), among others. Such a compilation enables the exploration of data-driven inquiries, highlighting the intricacies of flavor perception. FlavorDB was created to integrate multidimensional aspects of flavor molecules and represent their molecular features, flavor profiles, and details of natural sources [17].

Despite being the most comprehensive repository of flavor compounds, FlavorDB missed out on some of the critical details of flavor compounds available in the literature. FlavorDB2 expands the flavor space, thereby providing a significant advancement from FlavorDB. FlavorDB2 is a combination of ‘entity space’ and ‘molecular space’ (Figure 1), where the former provides the entities utilized in food from natural sources, and the latter shows the molecular and flavor profiles. One of the databases with closely aligned objectives, FooDB, collates molecules from food ingredients, albeit its focus is not on the chemical basis of flavor or flavor pairing

(http://foodb.ca). Some of the resources that have attempted to create a data compilation on specific aspects of flavor include Flavornet [9], BitterDB [12], SuperSweet [11], SuperScent [10]. Among the other efforts of data compilation are those targeted at nutritional factors (NutriChem), polyphenols (Phenol-Explorer), and the medicinal value of food [16, 13, 15, 14]. The uniqueness of FlavorDB lies in its extensive coverage of flavor compounds and nutritional information to more accurately convey the ingredient's flavor profile. This initiative not only facilitates a deeper understanding of flavor compounds but also provides a framework for analyzing their diverse applications and implications within the realms of gastronomy, chemistry, and sensory science. Through the lens of FlavorDB2 (https://cosylab.iiitd.edu.in/flavordb2/), researchers and food enthusiasts can delve into the rich tapestry of flavor chemistry, unlocking new avenues for exploration and innovation in the culinary and fragrance industries.

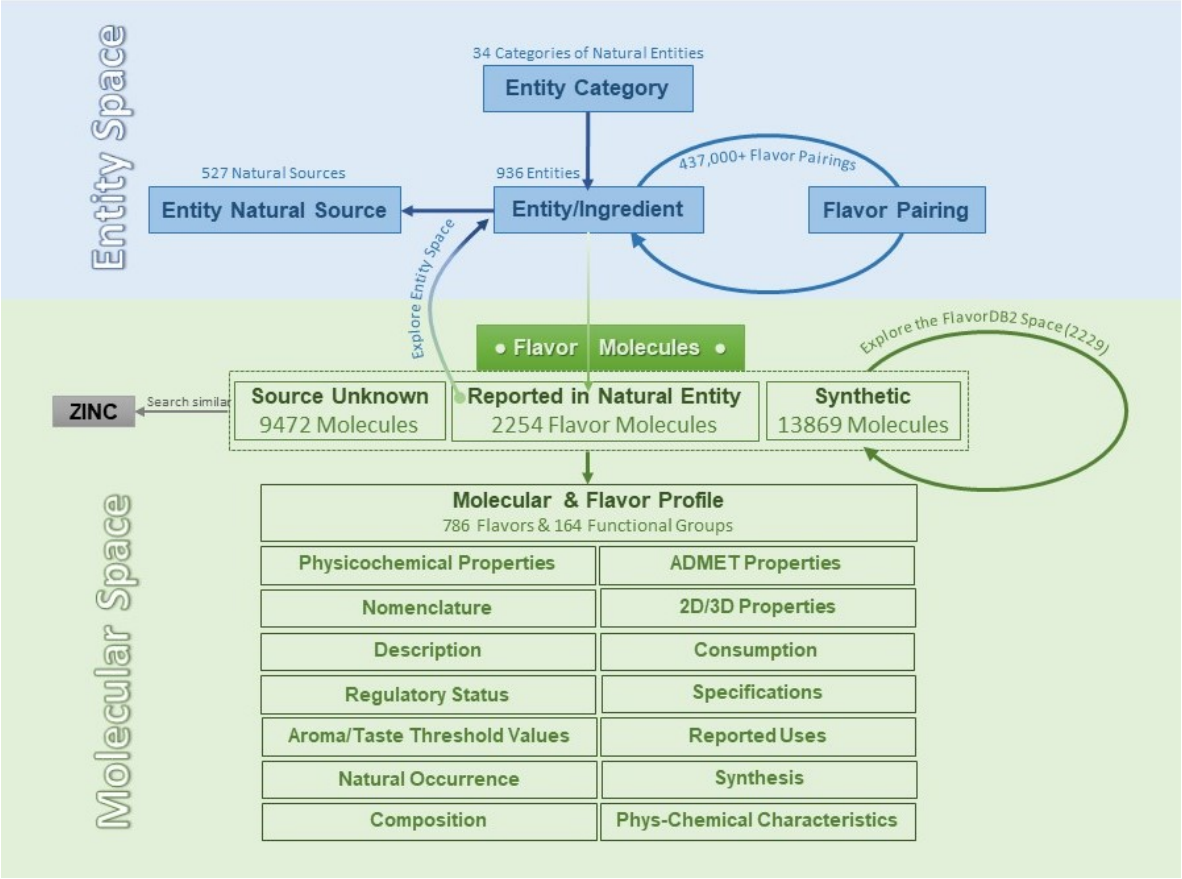

Figure 9.1: FlavorDB2 provides significantly advanced attributes of flavor compounds and search mechanisms. It presents a comprehensive repository of flavor compounds through a user-friendly interface and interlinked search engines for exploring the flavor universe.

9.2 Overview of FlavorDB2

FlavorDB2 collates information from various sources, including FooDB, Flavornet, SuperSweet, BitterDB, and Fenaroli's Handbook of Flavor Compounds [217], to build a comprehensive, structured repository of flavor molecules. While the number of compounds (25,595), ingredients (936), and ingredient categories (34) remain the same as that in FlavorDB, FlavorDB2 adds depth by adding a wide range of molecular features. Molecular features include: (i) variety of nomenclatures/IDs, (ii) flavor description, (iii) regulatory status, (iv) taste/aroma threshold values, (v) natural occurrence, (vi) composition, (vii) consumption statistics, (viii) specifications, (ix) reported uses in food categories, and (x) synthesis. These features represent vital attributes of flavor compounds that bring value to the database users.

The detailed breakdown of expanded contents to enrich the characterization of flavor molecules is provided. We have exhaustively accounted for an array of distinct features of flavor compounds, making this expansion as comprehensive as possible. Figure 9.8 depicts the schematic view of FlavorDB2, highlighting the search and molecular properties of the data.

9.2.1 Dataset collection and compilation

We sought to compile a detailed profile of each of the naturally occurring synthetic compounds comprising a wide variety of molecular attributes: (i) variety of nomenclatures/IDs, (ii) flavor description, (iii) regulatory status, (iv) taste/aroma threshold values, (v) natural occurrence, (vi) composition, (vii) consumption statistics, (viii) specifications, (ix) reported uses in food categories, and (x) synthesis. These features represent vital attributes of flavor compounds that bring value to the database users. For detailed statistics and additional information, refer to Figures 9.2, 9.3, 9.4, 9.5, 9.6, and 9.7, which provides comprehensive data on flavor consumption, food categories, regulatory status, and chemical properties.

9.2.2 Molecular nomenclatures

One of the issues in dealing with flavor compounds is variation in their nomenclature. A flavor compound could be identified with any of the following nomenclatures/IDs: common name, IUPAC (International Union of Pure and Applied Chemistry), SMILES (Simplified Molecular-Input Line-Entry System), Empirical Formula, CAS (Chemical Abstracts Service) Number, FEMA (Flavor Extract Manufacturers Association) Number, FL (FLAVIS) Number, NAS (National Academy of Sciences) Number, COE (Council of Europe) Number, EINECS (European Inventory of Existing Commercial Substances) Number, or JFCFA (Joint FAO/WHO Expert Committee on Food Additives) Number. Given this redundancy in naming, one of the challenges in the flavor molecular space is to arrive at a suitable compound given one of its

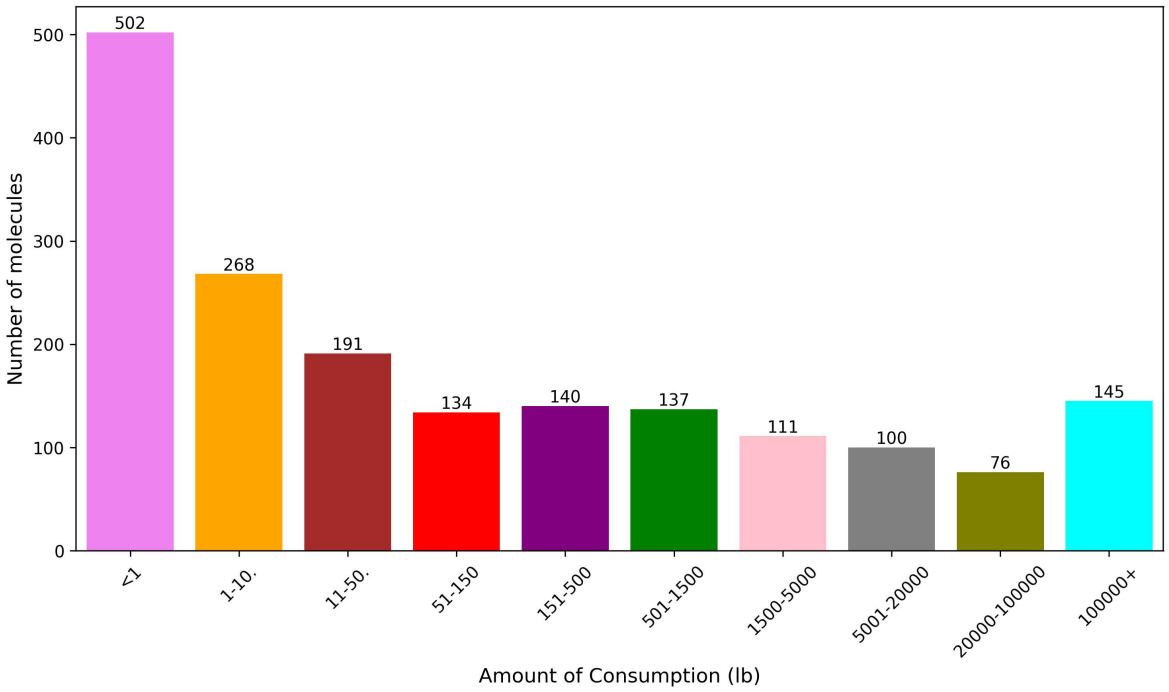

Figure 9.2: Bar plot depicting the consumption rate (in lb) per molecule, revealing that the maximum number of molecules has a consumption rate of less than 1 lb.

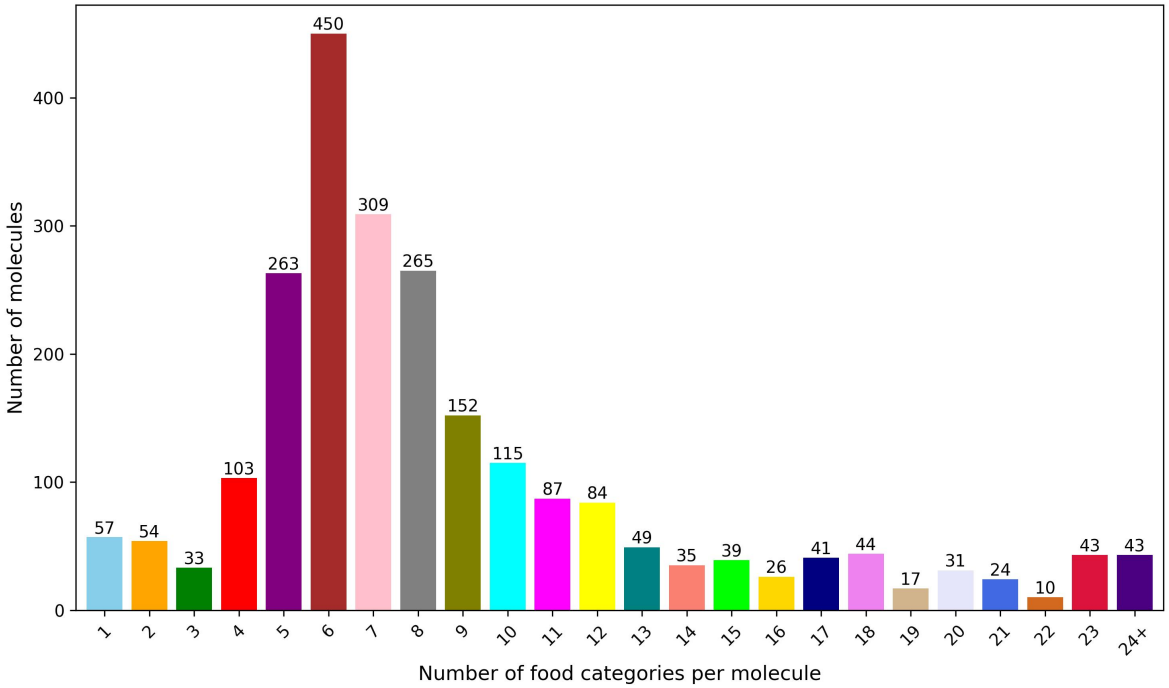

Figure 9.3: Statistics of number of food categories per molecule. The maximum number of molecules has six food category terms.

diverse IDs. Hence, we have implemented a search for all significant molecular IDs used for flavor compounds (in basic and/or advanced search). The nomenclature field also provides a list of synonyms, helping further reduce the ambiguity in identifying the desired compound. Using

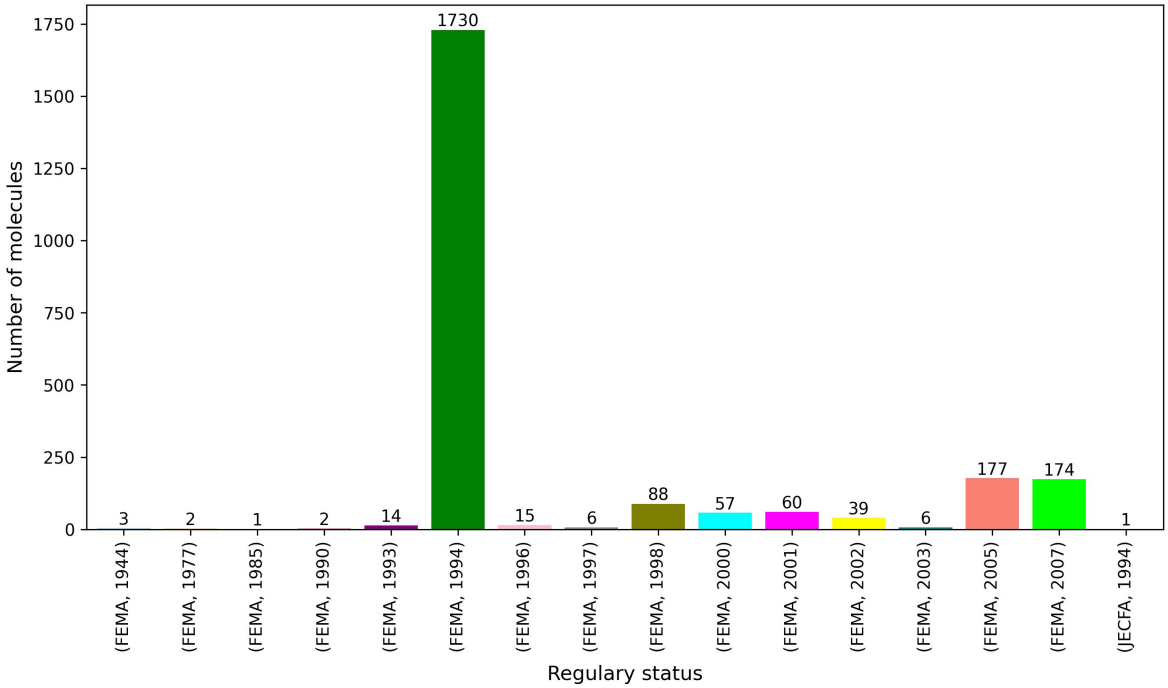

Figure 9.4: Bar plot showing the regulatory status of molecules, which depicts that a maximum number of molecules are associated with (FEMA, 1994).

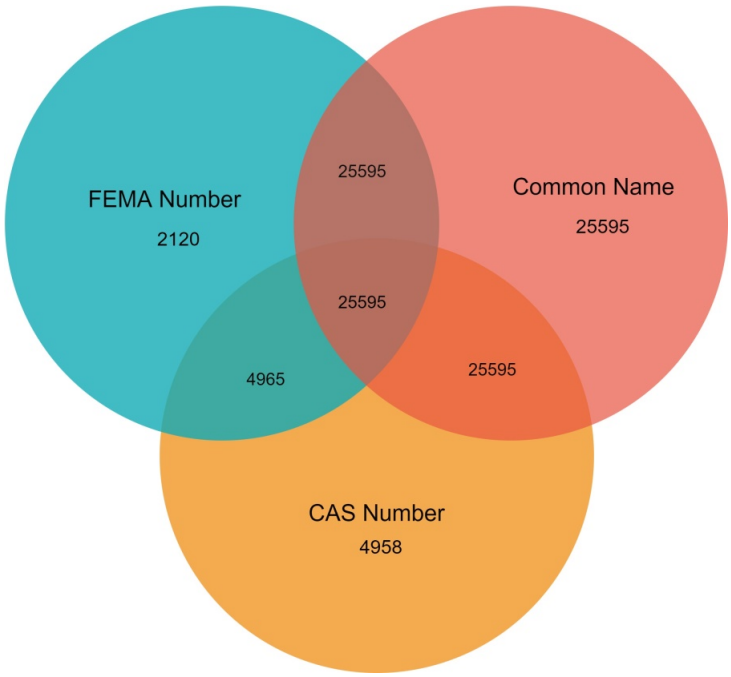

Figure 9.5: Statistics of availability of chemical properties of flavor molecules. A Venn diagram depicts that all molecules have their common name, while 2120 and 4958 molecules have their FEMA number and CAS number, respectively.

diverse molecular numbering systems, FlavorDB2 enables users to locate their desired flavor compounds within the repository, enhancing accessibility and usability for researchers, chefs, and industry experts.

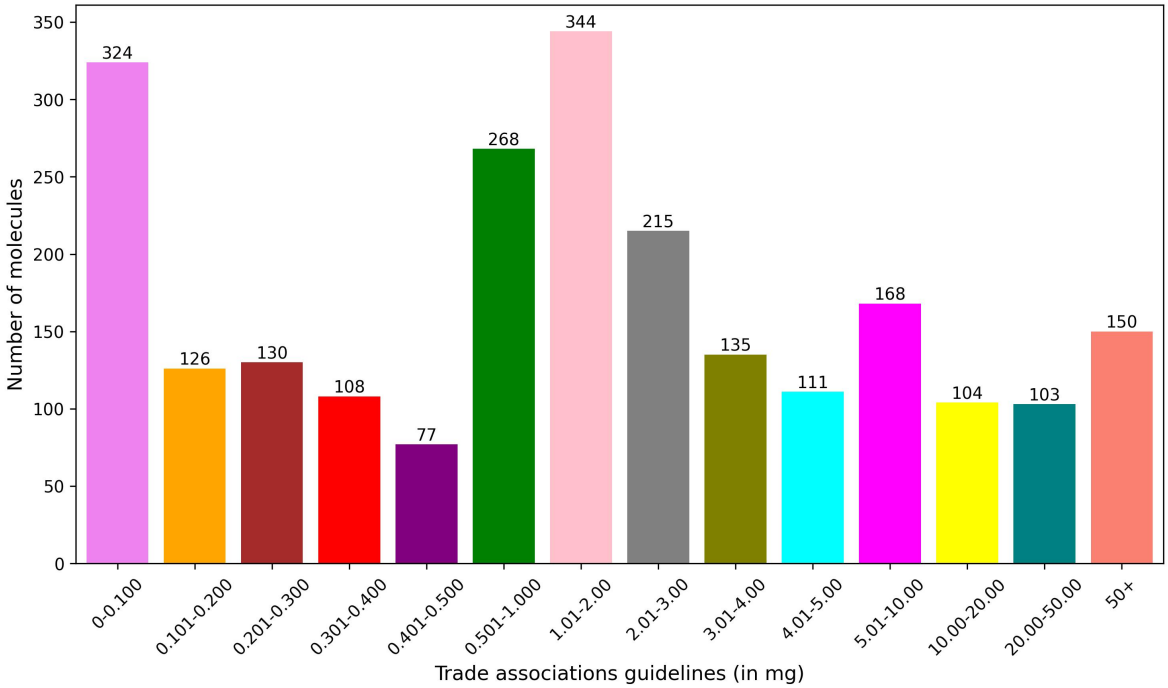

Figure 9.6: Bar plot depicting each molecule's Trade association guidelines (in mg). The maximum number of molecules has trade association guidelines between 1.01 to 2.00, followed by 0 to 0.100.

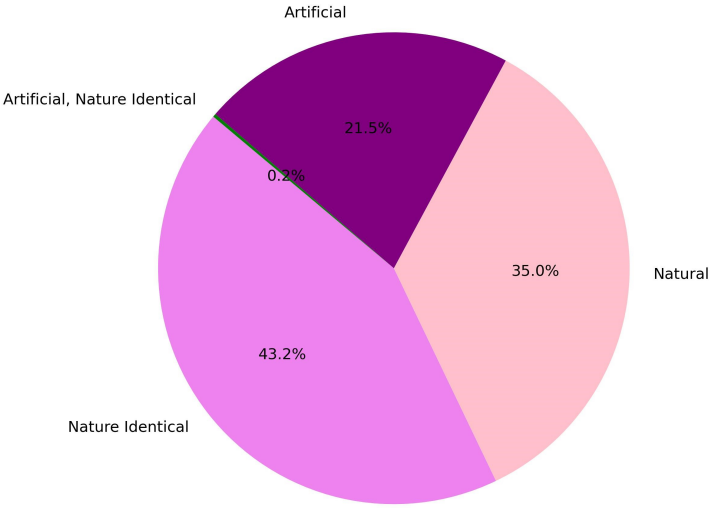

Figure 9.7: Statistics of International Organization of the Flavor Industry (IOFI) for each flavor molecule. The pie chart depicts that the maximum number of molecules are naturally identical rather than artificial.

9.2.3 Description

Beyond a list of flavor descriptors previously made available in FlavorDB, the 'Description' field provides a brief, nuanced flavor description that delves deeper into the sensory experience of each compound. For instance, the compound allyl hexanoate is reported to be found in pineapple,

among other ingredients, but the description field depicts a vivid picture – 'A colorless liquid with sweet, pineapple-like taste, and fruit-like aroma (pineapple)'. This field captures the essence of the compound, offering a clearer understanding of its flavor profile and potential applications to the users.

9.2.4 Regulatory status

This field provides regulatory and safety details for each compound, drawing from regulatory bodies such as COE (Council of Europe), JECFA (Joint FAO/WHO Expert Committee on Food Additives), FDA (Food and Drug Administration), IOFI (International Organization of the Flavor Industry), and trade association guidelines. The information includes permissible quantity or concentration levels of the compound, measured in parts per million (ppm), parts per billion (ppb), or milligrams (mg), as well as an indication of its nature-identical status. For allyl hexanoate, apart from providing the acceptable level of usage, including COE (5ppm), FDA (21 ppm), CFR (172.515), JECFA (0-0.13 mg/kg), and Trade Association Guidelines (10.749 mg). Additionally, the field mentions that the compound originates from natural sources.

9.2.5 Aroma/Taste threshold values

This field provides information on aroma and taste threshold values at specific concentrations with nuanced descriptions of sensory features. For example, this field specifies 'taste attributes at 10 ppm as sweet, juicy, fresh, pineapple, and fruity' for compound allyl hexanoate. Similarly, another compound, 2,3-Butanedione, has taste characteristics as sweet, buttery, creamy, and milky at 50 ppm. Phenol has medicinal, creosote, smoky, spicy, phenolic, and leather-like with notes of fried meat and coffee aroma characteristics at 5.5 ppm, while its taste attributes at 3ppm are spicy, phenolic, tobacco, musty, woody, medicinal, smoky, tarlike and slightly spicy clove-like.

9.2.6 Natural occurrence

The natural occurrences field details the natural ingredients and sources in which the compound is reported to be found. For allyl hexanoate, this field reads as 'Developed in a baked potato, mushroom, and pineapple', showcasing its diverse range of origins within various food sources. Similarly, compound Ethyl Octanoate is naturally found in apple, apricot, orange juice, grapefruit juice, guava, pineapple, cheddar cheese, other cheeses, butter, beer, cognac, rum, whiskey, cider, grape wines, cocoa, coconut meat, passion fruit, mango, pawpaw, and mastic gum leaf oil.

9.2.7 Consumption

This field provides the consumption value of a compound in terms of annual (in pounds) and individual consumption (mg/kg/day). The reported annual consumption is 16033.3 lb for allyl hexanoate, whereas the individual consumption is 0.01358 mg/kg/day. A compound Lactic Acid has 5,300,000 lb annual consumption with individual consumption of 4.4915 mg/kg/day.

9.2.8 Specifications

This field provides details of the physical and chemical characteristics of the compound, such as appearance, solubility, acid value, specific gravity, refractive index, assay, boiling, and melting points. For allyl hexanoate, this field suggests the appearance of the compound as 'colorless to light-yellow', soluble in ethanol with a specific gravity of 0.884-0.890 (25°C), and boiling point of 185°C. Benzaldehyde is reported as a colorless liquid with a refractive index of 1.544–1.547 (20°C), slightly soluble in water but soluble in ethanol, ether, and fixed or volatile oils, with a boiling point of 178°C, and with a specific gravity of 1.041–1.046 (25°C).

9.2.9 Reported uses

A compound is used in various food categories (such as Alcoholic Beverages, Hard Candy, Chewing gum, etc.) as a flavoring agent. This field provides the compound's various usage as per FEMA. Allyl hexanoate is reported to be used in alcoholic beverages, gravies, baked goods, hard candy, chewing gum, meat products, frozen dairy, non-alcoholic beverages, gelatin puddings, and soft candy. Another flavour compound 3-Heptanone is used in baked goods, meat products, frozen dairy, non-alcoholic beverages, gelatin puddings, and soft candy.

9.2.10 Synthesis

A compound is synthesized through a chemical process, either from a natural source or via synthetic mechanisms. This field provides details of the source compound and the process used for its synthesis. Allyl hexanoate is reportedly synthesized by esterification of n-caproic acid with allyl alcohol in the presence of concentrated H_2SO_4 or naphthalene-β-sulfonic acid in benzene under a nitrogen blanket. Similarly, Ethyl Propionate is synthesized from propionic acid, ethyl alcohol, and concentrated H_2SO_4 in chloroform at the boil.

9.3 Use Cases

9.3.1 Flavor compounds used in food category

Users can search FlavorDB2 for ‘food category’ to find all the compounds associated with a specific category. The database provides the popularly used typical and maximum concentrations for each compound within a category. Additionally, users can create compounded queries by combining multiple food categories using @ (AND) or ! (OR) operator. This feature adds the flexibility to identify a refined and precise flavor compound based on the culinary category.

9.3.2 Flavor compounds based on regulatory status

FlavorDB2 allows users to explore the flavor compounds based on their IOFI values: natural, artificial, nature-identical, not nature-identical, and identical. One can also search based on COE values, namely approved, approved in some quantity, used provisionally, and unknown. The database will generate the list of flavor molecules based on the input query. This feature is crucial for the food and fragrance industries.

9.3.3 Flavor compound synthesis

To identify a flavor compound that involves a given chemical in its synthesis process, one could search by the name(s) of the chemical(s). For example, searching by ‘acetic acid’ opens a list of flavor compounds involving acetic acid in their synthesis. One can create a compounded query to add more compounds with the help of @ (AND) or ! (OR) operator. The query can be performed on compounds and chemical processes (such as ‘esterification’, and ‘distillation’) or similar terms used in the synthesis field.

9.3.4 Exploring the flavor properties of ingredients

Cinnamon is widely recognized for its sweet and spicy aroma. Using the ‘Entity Search’ option in FlavorDB, it can be identified that cinnamon contains multiple flavor molecules. One such molecule, Cinnamaldehyde, is known for its sweet, spicy, and woody characteristics. It can be inferred that the distinct aroma and taste of cinnamon are primarily due to the presence of Cinnamaldehyde. These examples demonstrate how FlavorDB and its features can be interactively utilized to make data-driven decisions addressing various aspects of flavor.

9.3.5 Applications

Investigating molecules and components associated with flavor sensation is critical for various applications, including food pairing [18, 218] and molecular gastronomy. Chefs and culinary enthusiasts can identify novel and harmonious pairings that elevate culinary creations by analyzing the chemical profiles of different ingredients [38, 219, 220].

9.4 Web-server Implementation

Consistent with FlavorDB, the FlavorDB2 database has been designed using the MySQL database management system (https://www.mysql.com) to ensure data integrity and scalability. Django (https://www.djangoproject.com), a Python web development framework, has been used for web server development [221]. To enhance the functionality of FlavorDB2, jQuery and Bootstrap are used in the responsive framework. D3.js, Google Charts libraries, HTML, CSS, and Javascript are utilized in the frontend. Apache HTTP Server routes requests to Django and helps to enable data compression for faster page load times. FlavorDB2 (https://cosylab.iiitd.edu.in/flavordb2/) is accessible in the latest versions of Google Chrome, Firefox, Opera, Internet Explorer, and Microsoft Edge. By prioritizing user experience and accessibility, FlavorDB2 establishes itself as a valuable research tool readily available to the scientific community and beyond.

9.5 Conclusions

FlavorDB2 (https://cosylab.iiitd.edu.in/flavordb2/) serves as an updated and advanced repository of flavor molecules, integrating new attributes for analyzing flavor compounds from various perspectives. With its user-friendly interface, FlavorDB2 stands as a valuable tool for researchers, chefs, and fragrance industry experts, facilitating discovery and innovation in flavor and aroma domains.

Looking ahead, the resource can be enhanced in several meaningful directions. Incorporating the latest version of domain-specific databases such as BitterDB [222] will ensure improved accuracy and data completeness. Additionally, integrating comprehensive olfactory datasets like Pyrfume [223] could expand the scent-related dimensions and enable more nuanced modeling of perceptual aroma profiles. Linking compounds to vendor or sourcing information may further increase the practical applicability of the database, especially in culinary and commercial contexts. These enhancements will strengthen FlavorDB2's role as a robust, extensible platform for research and innovation in computational gastronomy, sensory science, and beyond.

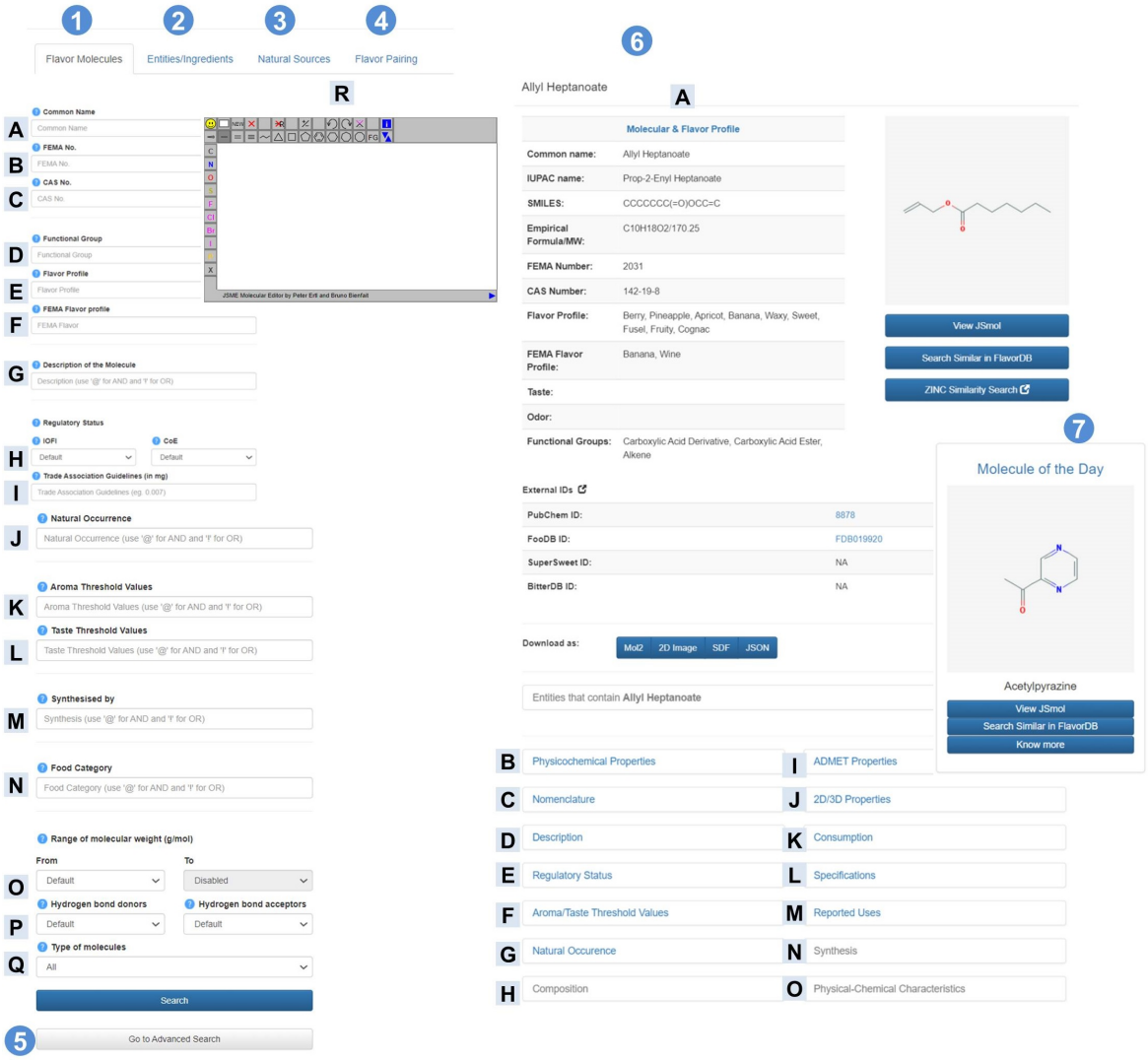

Figure 9.8: Schematic view of FlavorDB2 highlighting the search and molecular properties of the data. (1) Flavor Molecules Search (A. Common name, B. FEMA Number, C. CAS Number, D. Functional Group, E. Flavor Profile, F. FEMA Flavor Profile, G. Description of the Molecule, H. Regulatory Status, I. Trade Association Guidelines, J. Natural Occurrence, K. Aroma Threshold Values, L. Taste Threshold Values, M. Synthesized by, N. Food Category, O. Range of Molecular Weight, P. Hydrogen bond donors and acceptors, Q. Type of molecules, R. JSME Molecular Editor), (2) Entities/Ingredients Search, (3) Natural Sources Search, (4) Flavor Pairing Search, (5) Advanced Search, (6) Molecular Profile and Chemical Properties (A. Molecular and Flavor Profile, B. Physicochemical Properties, C. Nomenclature, D. Description, E. Regulatory Status, F. Aroma/Taste Threshold values, G. Natural Occurrence, H. Composition, I. ADMET Properties, J. 2D/3D Properties, K. Consumption, L. Specifications, M. Reported Uses, N. Synthesis, O. Physical-Chemical Characteristics), (7) Molecule of the Day.

Chapter 10

Molecular Taste and Toxicity Prediction

10.1 Predicting the Sweetness of Molecules

Sweetness is the most crucial sensory attribute of compounds that add calories and nutritional value to food [224]. Sweet amalgams, such as sugars and sweeteners, are highly employed throughout the food industry and significantly impact human health [225]. Over-consumption of sweeteners can lead to lifestyle disorders such as type-2 diabetes, heart disease, and other obesity-related diseases [226]. High sugar intake, in particular, is associated with weight gain, elevated blood glucose levels, and decreased insulin sensitivity, which can lead to type 2 diabetes [227].

To address these health concerns, low-calorie artificial sweeteners such as saccharin, aspartame, and sucralose have been developed. These sweeteners can satisfy the craving for sweetness without the associated caloric intake, making them valuable alternatives to sugar. Consequently, identifying sweet molecules with low calorific value is of paramount importance. Computational models that predict the sweetness of compounds are of great value in improving public health outcomes.

Several studies have attempted to determine the relationship between molecular structure and sweetness using the quantitative structure-activity relationship (QSAR) approach. In 1981, Iwamura [228] used QSAR models to identify the relationships between the structure of L-aspartyl dipeptide analogs and their sweetness potency. The compounds were extracted from literature and were divided into four categories (L-aspartic acid amide derivatives (I), L-Aspartylaminoethyl esters (II), L-Aspartyl aminopropionates (III), L-Aspartylaminoacetates (IV)) based on their structures. Spillane [229] concluded that hydrophobicity is an important parameter to determine the sweetness of the compound. Kinghorn [230] made an effort to discover sweeteners (terpenoids and phenolic) originating from plants using 31 compounds (17 sweet, 14 non-sweet). Aspartic acid amide derivatives were examined for structure-activity relationships [231] using 53 molecules and reported a correlation coefficient of 0.59. With a similar spirit, Yang et al. [232] investigated the sweetness of 320 sweet compounds extracted from the literature. They proposed that support vector machine and artificial neural network models are more desirable in demonstrating sweetness with a correlation coefficient of 0.87 and 0.88, respectively.

A QSAR-based study of 50 sulfamate compounds (21 sweet, 20 sweet-bitter, and 9 bitter) was done by Drew et al. [233]. Principal component and discriminant analyses were used to classify

the dataset to obtain a correlation coefficient of 0.902. Another QSAR-based study [234, 235] was implemented to predict the sweetness of 149 molecules (50 sulfamates, 41 isovanillyl, 40 sucrose derivatives, and 49 guanidine). In a similar spirit, Rojas et al. [40] proposed a quantitative structure-property-based approach to predict the sweetness of 233 molecules. They calculated molecular descriptors using Dragon and used the replacement method to select optimum descriptors. In another study, Rojas et al. [236] predicted the sweetness of 649 molecules (435 sweet and 214 non-sweet), considering both extended connectivity fingerprints and molecular descriptors using partial least squares discriminant analysis and N-nearest neighbors classification models. Further in this field, the QSAR study was implemented [226] with an extensive dataset of 487 sweet molecules. The sweetness was obtained by applying genetic function approximation and an artificial neural network, and a correlation coefficient of 0.832 and 0.831 was achieved, respectively. Ojha et al. [237] accounted for 239 compounds by considering only 2D molecular descriptors. They applied a multi-linear regression model validated using the k-fold cross-validation approach and predicted the sweetness using the QSAR model. Zheng et al. [238] collected a dataset of 530 sweeteners and 850 non-sweeteners to predict the sweet or non-sweet molecules and achieved an accuracy of 91%. They collected the dataset of 352 sweeteners with the relative sweetness value from the literature and obtained a correlation coefficient of 0.78. In a similar spirit, Weichen Bo et al. [239] collected the dataset from BitterDB [12] [18], FlavorDB [17] [19], and Supersweet [11] [20] to predict the bitter and sweet taste of molecules. They applied structural-taste relationship models using CNN, MLP-Descriptor, and MLP-Fingerprint and obtained a higher AUC value of 0.94 on MLP-Fingerprint. Due to the small dataset, this study yielded an inefficient artificial neural network model with low prediction accuracy.

In the present study, we have addressed one of the major problems faced by researchers, which is the unavailability of a large-scale dataset of small molecules and their sweetness values. We have compiled extensive data of 671 sweet molecules from the literature with known experimental sweetness values ranging from 0.2 to 22,500,000. We implemented deep learning-based regression models (Random Forest regressor (RF), Gradient Boost regressor (GBR), Adaboost Regressor (AR), Lasso Regressor (LR), Ridge Regressor (RR), XGBoost Regressor (XGB), and Multilayer Perceptron Regressor (MLP)) to predict the sweetness of small molecules. Our model is evaluated with the correlation coefficient, root mean square error, and mean absolute error. To enable the scientific community and industry professionals, we have built a user-friendly web server to provide prediction of the sweetness value of a molecule starting with its SMILES, Sweetpred (https://cosylab.iiitd.edu.in/sweetpred/).

10.1.1 Materials and Methods

Dataset collection

Obtaining sufficient experimentally validated data is one of the most difficult challenges in bioinformatics. We have compiled a comprehensive dataset of 671 sweet molecules from various credible sources [228, 230, 231, 232, 233, 234, 235, 40, 240] with sweetness values ranging from 0.20 to 22,500,000. To ensure the dataset's integrity and avoid redundancy, we utilized the Pybel library [241] to remove 164 redundant molecules, resulting in a refined collection of unique sweet molecules. This extensive dataset, named SweetpredDB, includes molecules with molecular weights ranging from 122.14 to 1,287.62 Daltons and is publicly accessible on Github.

Given the wide range of sweetness values, we considered the logarithmic transformation of sweetness (log sweetness or logSw), which varies from -0.69 to 7.35. It is evident from Figure 10.1 that most molecules have a log sweetness value between the range of 3.0 and 4.0. Among these, several molecules stand out for their exceptionally high sweetness values, including lugduname, bernardame, sucrononic acid, superaspartame, asp 215, isovanillic 11, neotame, 4,1',4',6'-4Br-sucrose, and benzenepropanoic acid.

Structure building

The structures of compounds were generated using Marvin Sketch 18.16 and stored in a string format using SMILES (Simplified Molecular Input Line-Entry System). Sweet molecules with Markush bonds and D/L configurations were excluded due to the challenges in extracting their SMILES representations. The schema describing the data collection process, feature selection, and model building is depicted in Figure 10.2.

Molecular descriptors

A molecular descriptor defines the compound's physical, chemical, and structural properties, which are crucial in cheminformatics and predictive modeling. In our study, we utilized two software tools to calculate a comprehensive set of molecular descriptors: PaDel-Descriptor and Mordred. PaDel-Descriptor [242] is an open-source software using the Java library that offers both command-line and graphical user interfaces for flexibility. Mordred, another freely available software, calculates molecular descriptors using the RDKit library and is accessible via the command-line interface [243].

We calculated 19,151 molecular descriptors for each compound from PaDel (663 2D, 134 3D, and 14,532 fingerprints) and Mordred (1,614). These descriptors encapsulate various proper-

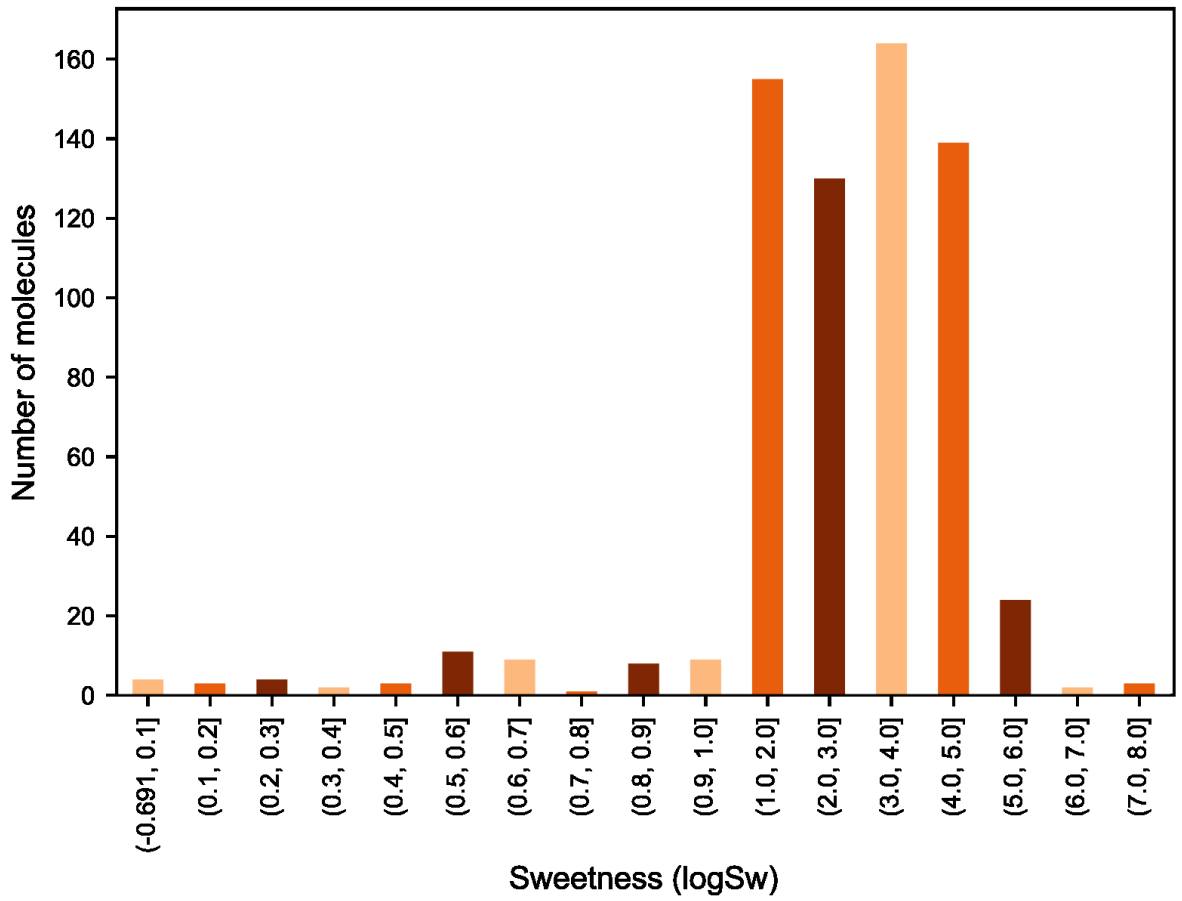

Figure 10.1: A histogram of molecules and their sweetness.

ties of molecules, including structural, thermodynamical, atomic, topological, and chemical attributes. These extensive molecular descriptors were integral for predicting the sweetness of molecules. By providing a detailed representation of each compound's properties, they enabled the development of robust computational models. The calculated descriptors enhance the predictive accuracy and provide insights into the molecular characteristics that influence sweetness, thus advancing our understanding of the relationship between molecular structure and sensory attributes.

Data preprocessing and feature selection

Due to the wide range of 19,151 molecular descriptors, it was essential to preprocess the data to ensure the quality and relevance of the descriptors used in our models. Initially, each descriptor was normalized using StandardScaler to standardize the range of the features. Additionally, we removed outliers by applying a Quantile Transformer, which helps to make the data distribution more uniform and reduces the impact of extreme values. Previous research [244] suggested that not all the descriptors obtained by PaDel were relevant for predictive modeling. To address this, we implemented the variance threshold method to eliminate features with low variance,

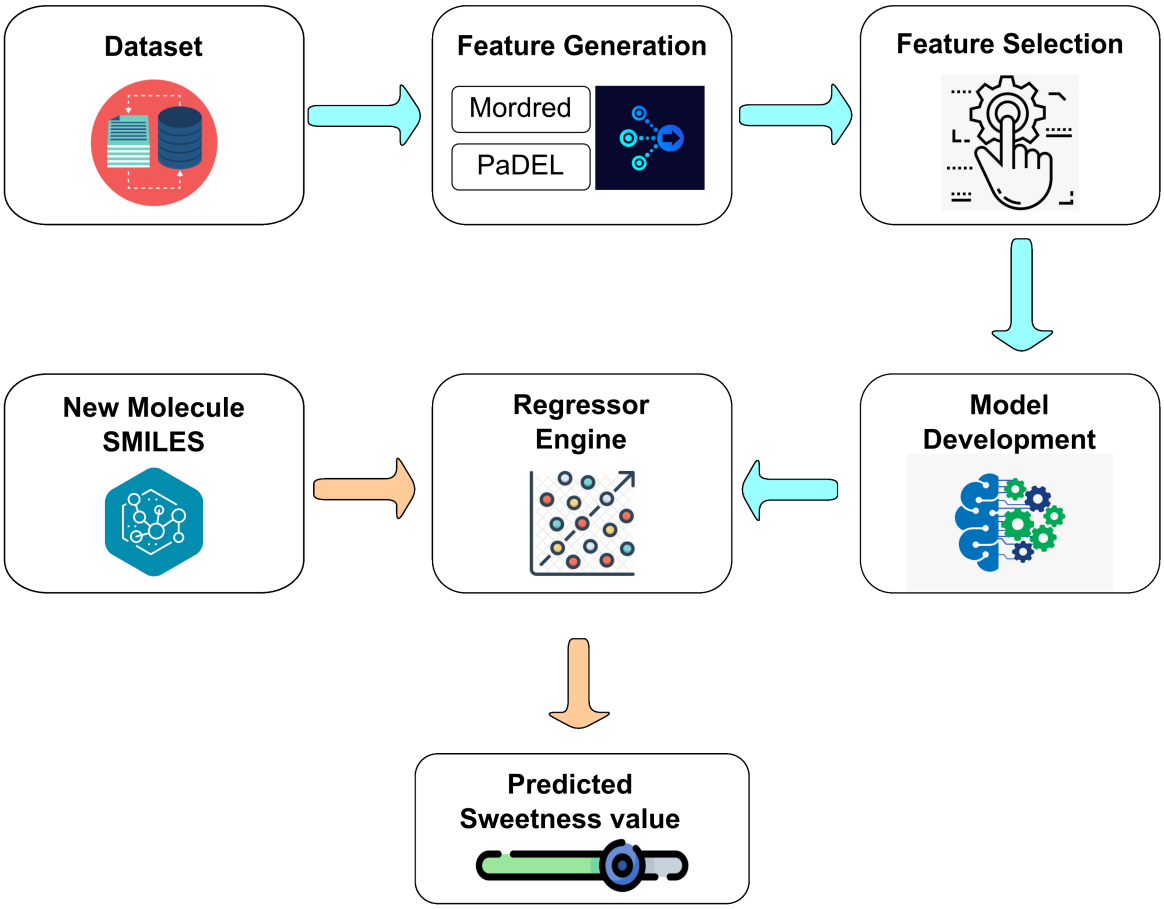

Figure 10.2: Schema for building a regressor model for sweetness prediction.

as such features provide little to no information about the target variable. Furthermore, we applied the Correlation-Based method to remove features with high correlation (≥ 0.7), reducing redundancy and multicollinearity in the dataset.

We narrowed the initial set to 1,300 relevant molecular descriptors through these preprocessing steps. To refine this selection further and enhance model performance, we utilized the SelectKBest feature selection algorithm. This algorithm ranks features based on their statistical significance in predicting the target variable and selects the top features for model training. By implementing feature selection techniques, we ensured that the final set of molecular descriptors used for model training was non-redundant. This process improved the models' efficiency and enhanced their predictive accuracy, enabling more reliable identification of sweet molecules. This refined dataset thus provides a robust foundation for developing advanced computational models to predict sweetness, ultimately contributing to the creation of healthier and more effective sweetening agents.

Model implementation

The dataset comprising 1,300 molecular descriptors was divided into training and testing sets with an 80:20 ratio. To enhance the accuracy of our predictive models, we applied the SelectKBest feature selection method with the score function of 'f_regression' to extract the best 15 molecular descriptors.

We implemented several regression models, namely Gradient Boost Regressor (GBR), Random Forest Regressor (RF), Multilayer Perceptron Regressor (MLP), Adaboost Regressor (AR), Lasso Regressor (LR), Ridge Regressor (RR), and XGBoost Regressor (XGB), to predict the sweetness of compounds. The efficiency of the models was assessed using the correlation coefficient (R), root mean square error (RMSE), and mean absolute error (MAE). R (10.1) determines the strength of association between actual and predicted values. RMSE (10.2) denotes the error rate between actual and predicted values. MAE (10.3) measures the absolute difference between actual and predicted sweetness values. Using these evaluation metrics, we were able to comprehensively assess the performance of each regression model, determining their effectiveness in accurately predicting the sweetness of molecules.

$$R = \frac{\sum_{i=1}^{N}(X_i - \bar{X}_i)(Y_i - \bar{Y}_i)}{\sqrt{\sum_{i=1}^{N}(X_i - \bar{X}_i)^2(Y_i - \bar{Y}_i)^2}} \tag{10.1}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y - \hat{Y})^2} \tag{10.2}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|Y - \hat{Y}| \tag{10.3}$$

Model interpretability

Evaluating the model's performance is a critical and complex task in machine learning. SHAP (SHapley Additive exPlanations) is an open-source library that answers questions about the reliability and interoperability of machine learning and deep learning models. Originating from Shapley values introduced in game theory in 1951, SHAP [245] provides a unified measure to understand feature importance.

We used a tree-based SHAP explainer to compute the SHAP values of the best model. Highly positive and highly negative SHAP values depict a high contribution of features in predicting the sweetness of molecules. SHAP values closer to zero suggest that the feature has less impact on the prediction. One of the most effective tools in SHAP is the summary plot. This visualization plots SHAP values on the x-axis. It ranks features from high to low based on their contribution

on the y-axis, providing a comprehensive dataset overview. By using SHAP, we can identify and interpret the essential features that drive the predictions of sweet molecules, enhancing our understanding of the model's outcomes.

10.1.2 Results

To determine the sweetness of 671 small molecules, we applied the SelectKBest feature extraction algorithm and obtained the top 15 molecular descriptors. We implemented several regression models and calculated the correlation coefficient to obtain the predictability of the regression models. Table 10.1 shows the performance statistics of regression models based on the correlation coefficient, root mean square error, and mean absolute error for the training and test datasets. Gradient Boost Regressor and Random Forest Regressor outperformed other models with correlation coefficients, and root mean square errors of 0.94, 0.23, and, 0.92, 0.28, respectively. These results indicate that both models have high predictive accuracy for determining the sweetness of molecules.

Figures 10.3 and 10.4 illustrate the relationship between experimental sweetness and predicted sweetness (logSw) for the Gradient Boosting Regressor and Random Forest Regressor, respectively. These plots demonstrate the predictive capabilities of these models in accurately estimating the sweetness of the compounds.

Table 10.1: Performance statistics of regression models to predict the sweetness of molecules.

Regressor	Training Dataset			Testing Dataset		
	R	RMSE	MAE	R	RMSE	MAE
Gradient Boost Regressor (GBR)	0.99	0.11	0.09	0.94	0.33	0.34
Random Forest Regressor (RF)	0.99	0.11	0.09	0.94	0.33	0.34
Adaboost Regressor (AR)	0.94	0.40	0.34	0.90	0.60	0.45
Lasso Regressor (LR)	0.96	0.30	0.20	0.87	0.54	0.45
XGBoost Regressor (XGB)	0.90	0.68	0.64	0.83	0.80	0.75
Ridge Regressor (RR)	0.90	0.50	0.54	0.80	0.70	0.65
Multilayer Perceptron (MLP)	0.80	0.53	0.45	0.76	0.68	0.50

We conducted a SHAP analysis to interpret the relevant features and validate the best-performing model (GBR). Figure 10.5 illustrates the highly contributing features to predict the sweetness of molecules, highlighting each feature's importance in the Gradient Boosting Regressor's performance. This analysis helps in understanding which molecular descriptors are most influential in determining sweetness, thereby enhancing the interpretability and reliability of the model.

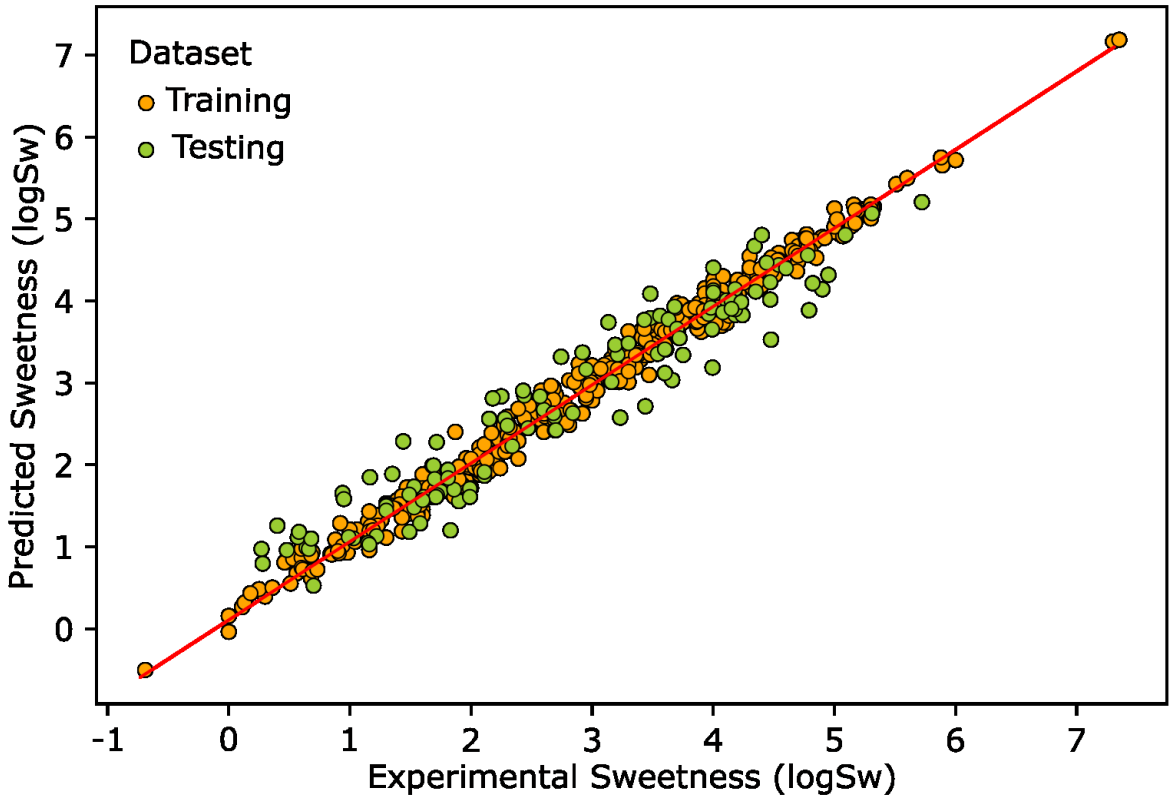

Figure 10.3: Correlation between sweetness predicted with Gradient Boost Regressor and experimental values.

Comparison of model performance with literature

We further evaluated our models on datasets used in previous studies (Table 10.2). To begin with, we used the dataset of 320 sweet molecules from Zhong et al. [232], in which the authors had applied multilinear regression and support vector machine models, achieving correlation coefficients of 0.87 and 0.88, respectively. In contrast, we achieved correlation coefficients of 0.91 and 0.89 on gradient boost and random forest regressors, respectively. Further, using a dataset of 103 molecules, Yang et al. [240] implemented support vector machine and artificial neural network models with correlation coefficients of 0.94 and 0.93, respectively. On the contrary, we obtained a correlation coefficient of 0.93 with gradient boost and 0.92 with random forest regressor. In another study, Goel et al. [226] used 487 sweet molecules and obtained a correlation coefficient of 0.83 using genetic function approximation and artificial neural network models. Our models returned comparable results on this dataset with correlation coefficients of 0.83 and 0.82 using gradient boost and random forest regressors. Vepuri et al. [231] applied genetic function approximation models (Comparative Molecular Field Analysis (CoMFA) and Comparative Molecular Similarity Indices Analysis (CoMSIA)) on 53 molecules, achieving correlation coefficients of 0.92 and 0.76, respectively. Our models obtained correlation coefficients of 0.90 and 0.89 on this dataset.

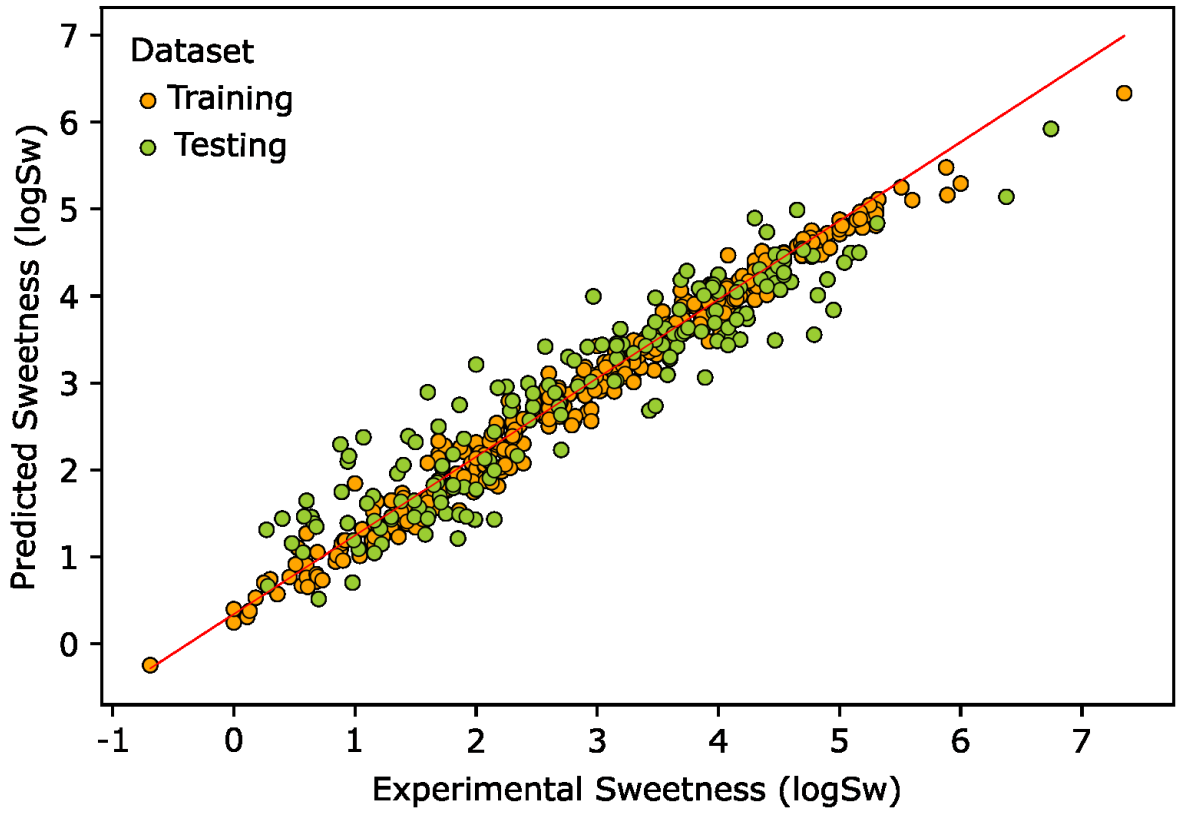

Figure 10.4: Correlation between sweetness predicted with Random forest Regressor and experimental values.

In summary, our models (Gradient Boost Regressor and Random Forest Regressor) yielded at par or better prediction performance compared to previous studies. Furthermore, our study utilized the largest dataset of sweet molecules hitherto investigated to predict the sweetness of small molecules.

Table 10.2: Comparison of model performance of SweetPred with previous literature.

Datasets	Previous Models		SweetPred	
	Model	R	Model	R
Zhong et al. [232] 320 molecules	MLR	0.87	GBR	0.91
	SVM	0.88	RF	0.89
Yang et al. [240] 120 molecules	SVM	0.94	GBR	0.93
	ANN	0.93	RF	0.92
Goel et al. [226] 487 molecules	GFA	0.83	GBR	0.83
	ANN	0.83	RF	0.82
Vepuri et al. [231] 53 molecules	COMSIA	0.92	GBR	0.90
	COMFA	0.76	RF	0.89

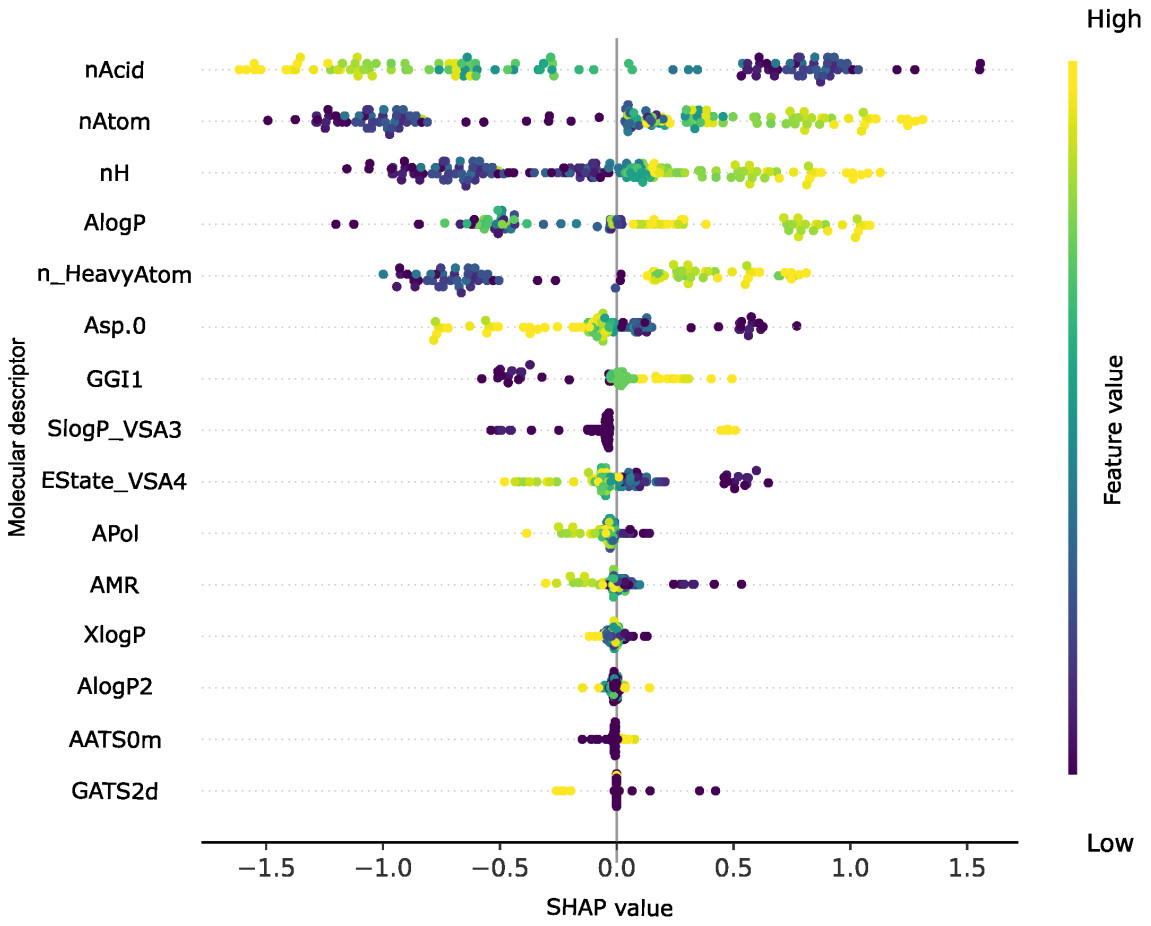

Figure 10.5: SHAP analysis plot depicting the top 15 relevant features for the Gradient Boosting Regressor.

Web-server implementation

To facilitate the prediction of sweetness for small molecules, we developed a web server, Sweetpred (https://cosylab.iiitd.edu.in/sweetpred/). This web server deploys the best-performing model from our study, providing a valuable tool for researchers and industry professionals to predict the sweetness of molecules. The tech stack includes HTML, CSS, and JavaScript for the front end, ensuring a user-friendly and visually appealing interface. The backend uses the Flask framework, which handles server-side operations efficiently. Sweetpred is a responsive website and can be viewed in the latest versions of Google Chrome, Firefox, Opera, Internet Explorer, and Microsoft Edge.

Users can easily input molecular data and receive predictions on sweetness, leveraging the advanced computational model developed in our research. Sweetpred is a significant contribution to the scientific community, aiding in developing healthier and more effective sweetening agents. The platform's versatility and accessibility ensure that it can serve as a valuable tool for various applications, enhancing research capabilities and supporting informed decision-making in the industry.

10.1.3 Conclusions

In this study, we curated the most extensive dataset of small molecules, SweetpredDB. This dataset is a valuable resource for researchers aiming to develop computational models for predicting the sweetness of molecules. The extensive range of sweetness values and molecular weights provides a robust foundation for training and validating models. By leveraging SweetpredDB, scientists can accelerate the discovery and design of new sweeteners that meet consumer demands for low-calorie and health-conscious options. Furthermore, the public availability of this dataset promotes transparency. It encourages collaboration within the scientific community, enhancing our understanding of sweetness and its implications for human health and the food industry.

We calculated the molecular descriptors using PaDel and Mordred open-source software and implemented machine learning and deep learning-based regression models to predict the sweetness of small molecules. Our results showed that the Gradient Boost Regressor and Random Forest Regressor outperformed other models (Multilayer Perceptron, Adaboost, Lasso, Ridge, and XGBoost) with a correlation coefficient of 0.94 and 0.92, respectively. These models can accurately predict sweetness levels, facilitating the discovery of practical and health-conscious sweetening agents. Some molecules presented as outliers violating the predictions of our models, such as sodium propylsulfamate, sodium (2-methylpropyl) sulfamate, sodium butylsulfamate, sodium (3-methylbutyl) sulfamate, with exceptionally low sweetness values, and the molecules with exceedingly high sweetness are lugduname, bernardame, sucrononic acid, superaspartame, asp 215, isovanillic 11, and neotame. Our study identified that atom count, AlogP, topological charge, electrostatic energy, MOE-type descriptors, and chipath were among the highly correlated features with sweetness.

Given the prevalence of diet-linked disorders, our study provides an effective computational screening strategy to identify compounds with desirable sweetness and low calorific load. A data-driven approach for computational prediction of sweetness is an essential step towards addressing major diseases such as type-2 diabetes. Thus, the state-of-the-art machine learning models presented here hold significant potential for achieving better public health. The ability to predict the sweetness of molecules offers numerous benefits for both individuals and the food industry. From empowering consumers to make healthier dietary choices to facilitating product development and innovation, predictive models are vital in promoting human health and advancing the food sector.

10.2 UmamiPred: Predicting the Umami Taste of Molecules and Peptides

Umami, the fifth basic taste after sweet, sour, bitter, and salty, is induced by specific amino acids and nucleotides like L-glutamate and inosinate, which activate specialized taste receptors [246, 247, 248]. Traditional foods such as soy sauce, cheese, and fermented Asian goods have long been known for their umami taste [249]. Its identification as a fundamental sensory modality apart from the four classic tastes happened in 2002 [250]. The molecular basis of umami perception, involving the interaction of umami chemicals with taste receptors such as T1R1/T1R3 [251], has garnered significant scientific interest due to its significance in food science, flavor enhancement, and nutritional research. Understanding the structural properties of umami peptides, particularly those with low molecular weight, that contribute to umami taste is critical for generating innovative flavor enhancers and increasing the palatability of foods [252, 248].

Umami peptides and molecules exhibit considerable sequence and structural diversity, complicating their systematic characterization and predictive modeling. Umami peptides typically consist of short sequences ranging from dipeptides to decapeptides, with varied amino acid compositions and positional preferences that influence their taste-modulating properties [252, 253]. Due to their negatively charged side chains, specific residues, such as glutamate (E) and aspartate (D), are frequently implicated in umami activity. Still, their contributions can vary based on sequence context and peptide length [248]. Furthermore, terminal modifications, hydrophobicity, and secondary structure propensities contribute to the functional diversity of umami peptides. In parallel, small-molecule umami compounds, including amino acids, nucleotides, and flavor enhancers like monosodium glutamate (MSG) and inosinate, present a wide range of chemical scaffolds and functional groups that interact with umami receptors through distinct binding modes [249, 251]. This chemical heterogeneity results in complex structure–activity relationships that are challenging to capture through traditional empirical methods alone. Consequently, the high degree of sequence and structural diversity within umami peptides and molecules underscores the necessity for robust computational frameworks to integrate sequence-based, structural, and chemical descriptor information for accurate prediction and classification.

While the biological mechanisms of umami have been thoroughly studied, the ability to predict umami taste from molecular structures using computational methods is still an emerging topic. QSAR modeling, molecular docking, homology modeling, and machine learning are examples of computational techniques for finding and characterizing umami peptides [254, 255, 256, 257, 258]. These strategies can help identify new umami enhancers, understand umami taste perception, and develop personalized food products for individual taste preferences. Several studies have attempted to classify umami small molecules and peptides using computational approaches. Charoenkwan et al. [253] introduced iUmami-SCM, a sequence-based model

to predict umami peptides by analyzing primary amino acid sequences, achieving a balanced accuracy (BACC) of 82.4%, sensitivity of 71.4%, and Matthews correlation coefficient (MCC) of 67.9%. Despite its effectiveness, the model's reliance on sequence information restricts its ability to capture deeper structural features, limiting its accuracy and generalizability when applied to diverse datasets. To address the limitations of sequence-only models, Charoenkwan et al. proposed UMPred-FRL [259]. This meta-predictor integrates six machine-learning algorithms with seven feature encodings, including amino acid composition, dipeptide composition, and composition-transition-distribution encodings. A broader set of features helped improve its predictive accuracy, achieving a BACC of 86% and MCC of 73.5%. However, the model struggles with complex structural variations in peptides due to the manual nature of feature extraction, which may overlook critical descriptors.

Virtuous Umami [260] utilizes the UMP442 peptides dataset, where molecular descriptors using Mordred and ensemble models combining Support Vector Machines and Random Forest classifiers were implemented. With a similar spirit, IUP-BERT [261] leverages the Bidirectional Encoder Representations from Transformers (BERT) to extract the feature embeddings followed by the SVM model, achieving an accuracy of 94% in predicting the umami peptides. UmamiPreDL [262] utilizes a pre-trained BERT transformer (ProtBERT) trained on over 217 million protein sequences coupled with Convolutional Neural Networks for the classification task. They achieved an accuracy of 94% on 5-fold cross-validation. A dataset of 867 (439 umami and 428 non-umami) compounds by combining the UMP442 peptide dataset with newly extracted small molecules from a patent database by Senomyx Inc. (now Firmenich Inc.) was curated [263]. Molecular descriptors were computed using RDKit and implemented a pre-trained transformer model, TabPFN [264], for the classification on small tabular datasets, achieving an accuracy of 93% on the combined peptide and molecular datasets and 90% on the peptide dataset.

To support these computational models, several benchmark datasets have been developed for umami peptide prediction. One widely used resource is the UMP789 database, introduced in the Umami-BERT study [265], which curated 789 experimentally validated umami and non-umami peptides from the literature. Its primary advantage lies in manual curation and precise binary classification, making it a reliable training and benchmarking dataset. However, its relatively small size and limited source diversity present challenges for broader generalizability. Besides UMP789, earlier studies have used datasets such as UMP442, BIOPEP-UWM, and manually compiled peptide collections, differing in curation rigor, peptide length distribution, and annotation consistency. While larger datasets offer increased chemical diversity, they often suffer from label noise and inconsistent experimental validation. A comprehensive assessment of these datasets, their limitations, and strengths remains vital for advancing robust and generalizable umami peptide prediction frameworks.

Several studies focused on peptide-based prediction [261, 250, 266, 265], which lacks the

generalizability for broader use across diverse chemical classes. This limitation has hindered the practical deployment of models in identifying umami taste compounds beyond peptides. Although NLP-based encoders have the potential to generate contextual-based embeddings, their ability to combine small molecule and peptide prediction has yet to be utilized. Furthermore, traditional cheminformatics approaches, combined with statistical machine learning techniques, still need to be utilized despite their potential to enhance umami classification through comprehensive feature extraction and robust predictive models.

Computational models present a novel method for accurately predicting umami features, given the complexity of taste perception and the structural diversity of chemicals. This study offers a thorough framework for the umami classification of peptides and small molecules using machine learning models via Simplified Molecular Input Line Entry System (SMILES) representations (Figure 10.6). Unlike existing studies focusing on either peptides or small molecules in isolation, our approach integrates both data types into a single predictive model, thereby expanding the chemical space covered and enhancing model generalizability. Furthermore, we implement a comprehensive feature extraction strategy that combines Morgan Fingerprints, RDKit descriptors, and Mol2Vec embeddings. This multi-layered feature representation captures diverse aspects of molecular structure and physicochemical properties, addressing the limitations of single-source feature extraction methods used in prior research. To support practical applications, we have deployed our model as a publicly accessible web server, providing researchers and industry professionals with an easy-to-use tool for rapid umami taste prediction of novel compounds.

10.2.1 Material and Methods

Dataset collection

The dataset used in this study comprises of peptides (442) and small molecules (426). We have used the UMP442 database [253], which includes 442 peptides, of which 140 are classified as umami and 302 as non-umami peptides. The minimum and maximum lengths of peptides are 2 and 39, with an average molecular weight of 582.54. A modified version of the UMP442 dataset, available online, includes SMILES (Simplified Molecular Input Line Entry System) representations of these peptides. As discussed in Pallante et al. [260], the SMILES format enables the treatment of peptides as small molecules, allowing advanced molecular feature encodings and the computation of molecular descriptors, facilitating an in-depth analysis.

The present study incorporates small molecule data from a publicly available patent (US Patent No. 8,124,121 B2) filed by the American biotechnology company Senomyx Inc. (now Firmenich Inc.). This patent describes synthesizing a wide range of small molecules designed for use as flavoring agents and enhancers in food, beverages, and oral pharmaceuticals. The patent details the synthesis of 426 small molecules, categorized into 127 sweet and 299 umami molecules.

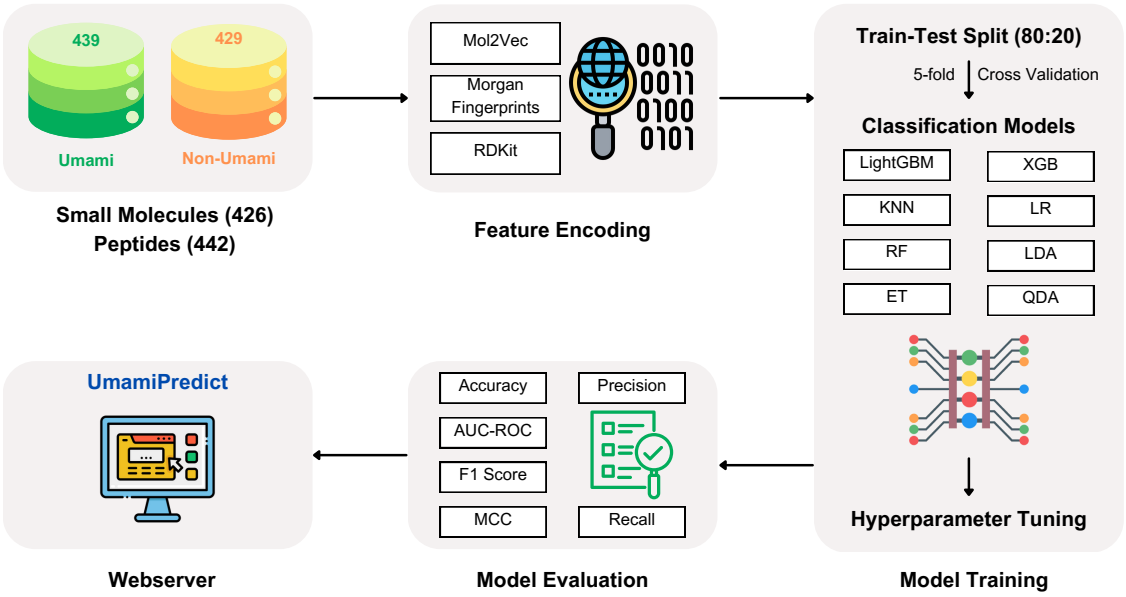

Figure 10.6: A systematic pipeline to predict the umami taste of small molecules and peptides using machine learning models, including curation of umami and non-umami datasets, feature encoding, model training, model evaluation, and web server creation. Based on the performance, Mol2vec feature encoding is used for peptides, Morgan for small molecules, and a combined dataset.

Umami molecules are defined within the patent based on their ability to activate the human T1R1/T1R3 receptor, a property that was confirmed through in vitro dose-response analysis. The IUPAC names of these molecules were manually extracted from the patent and converted into their corresponding SMILES representations using the OPSIN (Open Parser for Systematic IUPAC Nomenclature) API [267, 268] to build a comprehensive dataset of small molecules. To ensure accuracy, the generated structures from OPSIN were cross-referenced with the original structures detailed in the patent. The final dataset combines 868 samples, including 439 umami and 429 non-umami compounds. The non-umami class comprises both bitter peptides and sweet molecules. The dataset contains no duplicates and is balanced; hence, no data augmentation was required to address class imbalance issues.

However, combining peptides and small molecules introduces inherent heterogeneity in terms of their physicochemical and structural properties. Peptides, composed of amino acid sequences, exhibit characteristics such as sequence order, backbone flexibility, and potential secondary structures, while small molecules possess distinct 3D structures, functional groups, and stereochemical arrangements. Although both data types were converted into SMILES representations to enable unified descriptor computation, many conventional molecular descriptors primarily capture atom-bond level properties. They may not fully represent the sequence-dependent or conformational features of peptides. As a result, there may be limitations in the ability of these

descriptors to equally characterize and align the structural nuances of both peptides and small molecules. This disparity presents a potential challenge for developing predictive models that generalize across these chemically distinct compound classes.

To facilitate a unified representation of both peptides and small molecules, we converted peptide sequences into SMILES format. This conversion enables standardized cheminformatics tools and molecular descriptors, which are traditionally designed for SMILES-based inputs. Unlike raw peptide sequences, SMILES encodes detailed structural information, including stereochemistry, ring structures, and functional groups, allowing for a richer and more chemically relevant representation. This structural uniformity is crucial for applying the same feature extraction techniques, such as Morgan fingerprints, RDKit descriptors, and Mol2Vec embeddings, across both peptides and small molecules, ensuring consistency in the model input. Therefore, SMILES representation offers a practical and chemically informative format that supports a more generalizable and scalable predictive framework.

Feature extraction

We employed three methods to extract the features of small molecules: Morgan Fingerprints, RDKit descriptors [243], and the Mol2Vec [269] molecular embedding technique. Each method provided a unique perspective on the molecular structure, enabling comprehensive feature representation for the classification of umami and non-umami compounds.

Morgan Fingerprints were utilized to identify critical molecular substructures to discriminate between umami and non-umami compounds. Morgan uses circular fingerprints to encode a molecule's local atomic environment. The process begins with each atom receiving an initial identifier based on attributes such as atomic number and bond type. The approach then iteratively examines the neighboring atoms around each atom up to a predetermined radius, encapsulating these atomic environments into distinct substructures. In our investigation, we used the Extended Connectivity Fingerprints (ECFP) variant with a radius of two, which detects molecular substructures within two bonds of each atom. This radius encodes local molecular characteristics, such as functional groups, ring systems, and other essential substructures. The resulting 2048-bit binary vectors efficiently describe these chemical neighborhoods and are highly scalable for big datasets, yielding feature vectors that are both compact and informative. This circular fingerprinting technique excels in detecting cyclic structures and subtle differences in molecular substructure, making it ideal for distinguishing between umami and non-umami compounds. Morgan Fingerprints' robustness and scalability make them an excellent tool for identifying flavor-related compounds [255].

RDKit descriptors are frequently used in cheminformatics because of their computational efficiency and ability to offer interpretable features based on chemical attributes such as bioactivity, reactivity, and solubility. RDKit descriptors provide the structural and physicochemical features

of the molecules. We used RDKit to obtain 208 molecular descriptors for each molecule. These descriptors captured essential features such as molecular weight, LogP (partition coefficient, which indicates hydrophobicity), the number of hydrogen bond donors and acceptors, the number of rotatable bonds, aromaticity, and ring count [257].

Finally, we used Mol2Vec [269], an NLP-inspired approach for creating continuous vector embeddings of chemical substructures, to encode the molecules in a context-aware manner. Mol2Vec, like the Word2Vec paradigm, tokenizes molecules into their constituent substructures and generates vector representations depending on their chemical context within the molecule. Morgan Fingerprints (radius 0) were used to represent each molecule's substructures, and the embeddings were pre-trained on a vast quantity of molecular data to capture interactions between distinct chemical substructures. In our investigation, each molecule was recorded as a 300-dimensional continuous vector, with each dimension including relevant information about the substructure's identity and chemical environment. Mol2Vec provided a generalizable and efficient approach to molecular feature extraction by leveraging pre-trained embeddings [253].

The combination of these three feature extraction methods—Morgan Fingerprints for structural substructures, RDKit descriptors for physicochemical attributes, and Mol2Vec for context-aware embeddings—enabled us to extract a wide range of features from our dataset. Each solution had a distinct advantage: Morgan Fingerprints supplied scalable and compact bit vectors, RDKit descriptors provided interpretable molecular attributes, and Mol2Vec presented chemically contextualized molecular embedding. Together, they laid the groundwork for developing robust machine-learning models capable of accurately categorizing flavor-related chemicals.

Model implementation

We have implemented various machine learning models to predict the umami taste of molecules. The models include Logistic Regression (LR), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), K-Nearest Neighbors (kNN), Random Forest (RF), ExtraTrees (ET), Light Gradient Boosting Machine (LightGBM), and Extreme Gradient Boosting (XGB). LR served as a baseline classifier, modeling the probability of the binary classification outcome. LDA and QDA were used for linear and quadratic decision boundary modeling, respectively, while kNN classified molecules based on the majority class among the nearest neighbors in the feature space. We have used an Intel Xeon R Gold CPU server, utilizing 10-12 GB of RAM to train the models.

Ensemble methods such as RF and ET constructed multiple decision trees, capturing more complex relationships within the molecular data. Advanced gradient boosting frameworks, including LightGBM and XGB, further enhance model performance by iteratively building trees and minimizing classification errors. Optuna, a state-of-the-art hyperparameter optimization framework, was employed to tune model parameters and enhance their predictive performance.

To rigorously assess the predictive performance and generalizability of the machine learning models, we adopted a K-fold cross-validation strategy with K = 5. In this approach, the dataset was randomly partitioned into five equal-sized folds. Each model was trained on four folds and tested on the remaining one, repeating this process five times so that each fold served as a test set once. The final performance metrics were computed as the average across all five folds, reducing the risk of overfitting and ensuring model robustness on unseen data.

Evaluation metrics

A wide variety of evaluation metrics were used to examine the efficacy of our models, which include Matthews Correlation Coefficient (MCC), Receiver Operating Characteristic Area Under the Curve (ROC-AUC), Area Under the Precision-Recall Curve (AUC-PR), F1-Score, Precision (P), Recall (R), and Specificity. Each of these metrics provided unique insights into the models' accuracy, class distribution handling, and capacity to accurately detect both umami and non-umami compounds, ensuring a comprehensive perspective. AUC-PR was used as the primary metric for hyperparameter tuning via 5-fold cross-validation on the training set.

10.2.2 Results

Data characterization

Our dataset of small molecules and peptides is balanced, with equal representation of the Umami (Taste 1) and Non-umami (Taste 0) classes; however, individual datasets exhibit imbalance. The UMP442 dataset (peptides) contains 140 umami and 302 non-umami molecules, while small molecules comprise 299 umami and 127 non-umami compounds. T-SNE (t-distributed Stochastic Neighbor Embedding) analysis of the Morgan Fingerprint embedding reveals patterns of high-dimensional data in a lower-dimensional space. We have used the t-SNE parameters as n_components =2, random_state=0, and perplexity=15. The t-SNE plots (Figure 10.7) show distinct clusters for umami and non-umami classes, though significant overlap exists. In the protein dataset, the separation between classes is less prominent, suggesting that the peptide data presents challenges for classification due to its poor separability. On the other hand, the molecules dataset shows more pronounced clustering, with clearer groupings of umami and non-umami molecules, hinting at the potential efficacy of models for specific datasets. In the combined dataset, regions of overlap between the two classes persist, indicating that molecular fingerprints alone may not fully capture the features necessary to distinguish taste-related properties.

We have implemented SMOTE (Synthetic Minority Oversampling Technique) for handling class imbalance in the dataset during model training. SMOTE synthetically generates new instances

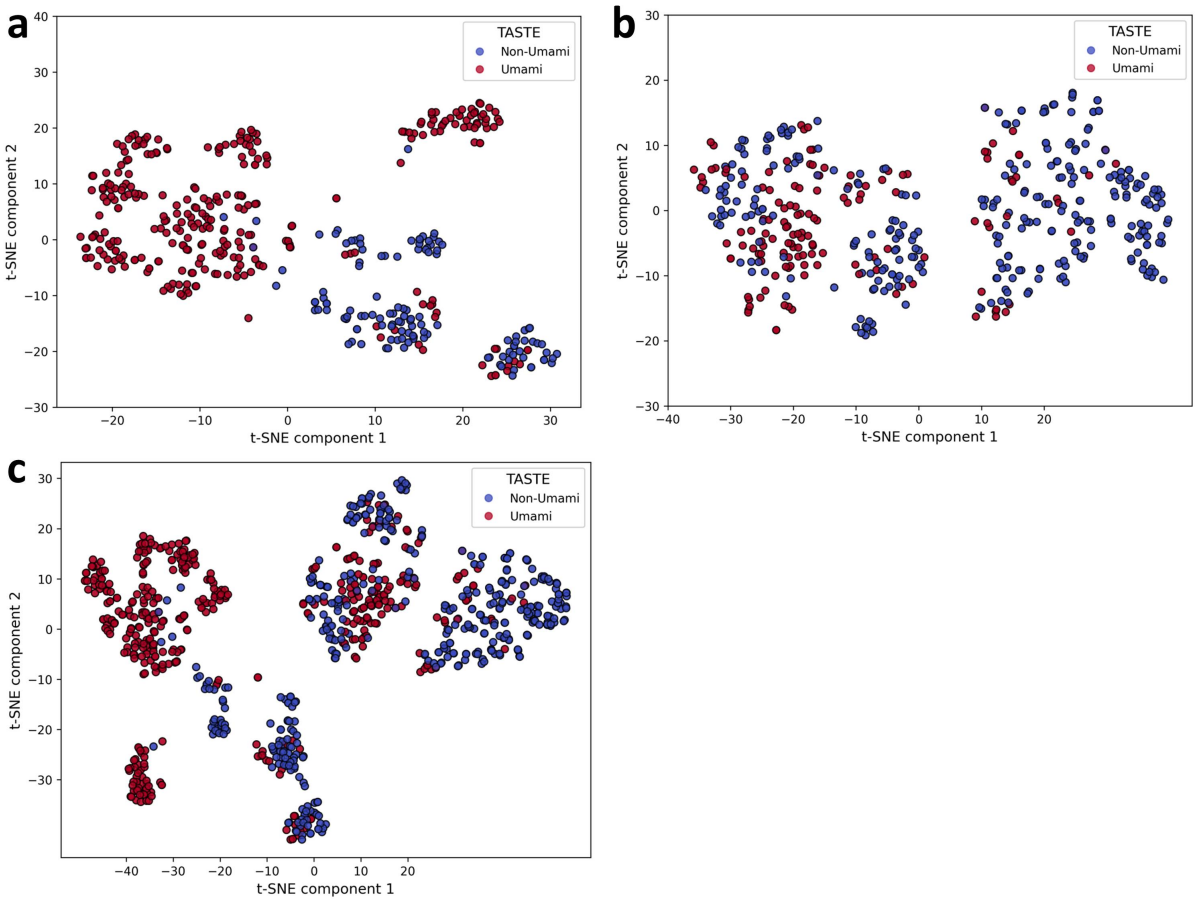

Figure 10.7: t-SNE plots characterizing the umami and non-umami datasets. (a) Molecules Dataset, (b) Peptides Dataset, and (c) Combined Dataset.

of the minority class by interpolating between existing minority samples.

Performance of classification models on peptide dataset

Table 10.3 shows the results of Mol2Vec encoding on the peptide-only dataset for classifying umami peptides, as Mol2Vec captures contextual and structural relationships between molecular substructures in a data-driven manner, making it more suitable for representing the sequential nature and diverse structural motifs of peptides. The five-fold cross-validation was conducted with no overfitting observed. We implemented various classification models: Random Forest (RF), XGBoost (XGB), Extra Tree (ET), K-Nearest Neighbors (kNN), Light Gradient Boost (LightGBM), Logistic Regression (LR), Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA). LightGBM classifier achieved the best performance with an accuracy of 92.13%, AUROC of 94.05%, and F1 score of 83.72%, followed by a comparable performance with the XGB and RF classifiers.

Model	Training							Testing						
	Acc	AUC-ROC	F1	MCC	P	R	Sp	Acc	AUC-ROC	F1	MCC	P	R	Sp
LightGBM	100	100	100	100	100	100	100	92.13	94.05	83.72	78.58	81.82	85.71	94.12
XGB	100	100	100	100	100	100	100	92.13	92.65	82.93	77.86	85.00	80.95	95.59
RF	98.87	99.87	98.32	97.46	98.32	98.32	99.15	92.13	90.58	82.93	77.86	85.00	80.95	95.59
LR	85.27	93.24	78.33	67.18	77.69	78.99	88.46	92.13	89.64	76.19	68.84	76.19	76.19	92.65
LDA	86.40	93.19	79.83	69.58	79.83	79.83	89.74	89.89	89.64	75.56	67.60	70.83	80.95	89.71
QDA	66.86	92.67	3.31	10.59	100	1.68	100	88.76	89.50	38.00	0.00	23.60	1.00	100
kNN	100	100	100	100	100	100	100	88.76	93.35	77.27	69.96	73.91	80.95	91.18
ET	98.58	99.82	97.89	96.83	98.31	97.48	99.15	87.64	91.35	82.93	77.86	85.00	80.95	95.59
NB	57.22	67.74	58.40	30.80	43.44	89.08	41.03	76.40	70.13	46.34	26.25	31.15	90.48	38.24
SVM	90.93	96.66	86.67	79.81	85.95	87.39	92.74	50.56	92.30	79.07	72.44	77.27	80.95	92.65

Table 10.3: Performance comparison of various models on peptide dataset.

Performance of classification models on small molecules dataset

Table 10.4 shows the results of Morgan fingerprints encoding on the molecule classification task, yielding the best results among the chosen encodings. LDA and ET classifiers achieved the best accuracy of 98.84%. LightGBM and XGB classifiers displayed comparable performance with an accuracy of 97.67% alongside perfect ROC-AUC and AUC-PR scores of 1. Their F1 Scores were recorded at 0.9839, and MCCs were noted at 0.9438, indicating strong predictive performance as well. These models also achieved perfect precision and maintained a recall of 96.83%, while specificity remained at 100%, further emphasizing their effectiveness. RF and LR models demonstrated accuracies of 95.35% with ROC-AUC scores of 0.9979 and 1, respectively. Their F1 Scores were 0.9672, and their MCCs were 0.8932, highlighting decent predictive capabilities. These models also had perfect precision, with recalls of 93.65% and specificity at 100%, showing they could identify both classes effectively. In contrast, kNN classifier achieved a lower accuracy of 94.19%, accompanied by a ROC-AUC score of 0.9959 and an AUC-PR of 0.9985. Its F1 Score was 0.9593, with precision at 98.33% and recall at 93.65%, while the MCC was recorded at 0.8607, suggesting a reasonably practical performance, albeit not as substantial as the best models. QDA significantly underperformed across all metrics, attaining an accuracy of 73.26% and an MCC of 0, suggesting that it could have yielded meaningful predictions for this classification task. Its ROC-AUC score was 0.5, and the AUC-PR was 0.8663, indicating that the model did not perform better than random guessing.

Model	Training							Testing						
	Acc	AUC-ROC	F1	MCC	P	R	Sp	Acc	AUC-ROC	F1	MCC	P	R	Sp
LightGBM	97.35	99.72	98.10	93.75	97.89	98.31	95.19	98.84	99.93	99.20	97.11	100	98.41	100
ET	100	100	100	100	100	100	100	98.84	99.59	99.20	97.11	100	98.41	100
SVM	99.71	100	99.79	99.31	99.58	100	99.04	98.84	100	99.20	97.11	100	98.41	100
XGB	98.24	99.88	98.73	95.84	98.73	98.73	97.12	97.67	100	98.39	94.38	100	96.83	100
LDA	100	100	100	100	100	100	100	97.67	100	98.39	94.38	100	96.83	100
LR	100	100	100	100	100	100	100	95.35	100	96.72	89.32	100	93.65	100
NB	97.94	98.52	98.49	95.35	100	97.03	100.00	95.35	94.06	96.83	88.13	96.83	96.83	91.30
RF	98.82	99.89	99.16	97.22	98.74	99.58	97.12	94.19	99.72	95.93	86.07	98.33	93.65	95.65
kNN	93.53	98.43	95.22	85.52	97.77	92.80	95.19	94.19	99.59	95.93	86.07	98.33	93.65	95.65
QDA	89.41	87.98	92.91	75.32	86.76	100	65.38	88.37	81.50	92.54	69.15	87.32	98.41	60.87

Table 10.4: Performance of classification models on small molecules dataset.

The results indicate that both LDA and the ensemble methods—such as ExtraTrees, Light-

GBM, XGBoost, and Random Forest—are particularly effective for molecule classification using Morgan fingerprints. The outstanding performance of LDA, with metrics approaching perfection, implies that the data is likely linearly separable within the feature space provided by Morgan Fingerprints. The strong outcomes of the ensemble methods further corroborate their appropriateness for this task. High ROC-AUC and AUC-PR scores demonstrate exceptional discriminative capabilities, while elevated precision and recall values highlight the models' strengths in accurately identifying positive and negative classes.

Although the kNN classifier did not surpass the leading models, its performance remains reasonable, suggesting that alternative algorithms can capture relevant patterns when adequately tuned. Conversely, the poor results of the QDA model indicate that its underlying assumptions may not align with the characteristics of this dataset and its feature representation. The enhanced scores compared to those from the peptide dataset suggest inherent separability within the molecular data, which has positively influenced model performance. Overall, this analysis underscores the efficacy of using Morgan fingerprints and advanced classification algorithms to achieve high accuracy in molecule classification tasks.

Combined performance

Table 10.5 shows the performance of the classification model on a combined dataset of peptides and molecules. LightGBM achieves the highest accuracy of 96.55%, F1 score of 99.38%, AUC-ROC of 93%, and MCC of 99.3%, suggesting a strong correlation between the observed and predicted classifications. XGB and LDA showed comparable performance with an accuracy of 95.4% and MCC of 99.12%. The five-fold cross-validation was conducted with no overfitting observed for each model.

Model	Training							Testing						
	Acc	AUC-ROC	F1	MCC	P	R	Sp	Acc	AUC-ROC	F1	MCC	P	R	Sp
LightGBM	99.14	99.98	99.18	98.27	98.64	99.72	98.49	96.55	99.61	99.38	93.04	94.87	97.37	95.92
ET	99.86	100	99.86	99.71	100	99.72	100	95.98	99.29	95.30	91.84	97.26	93.42	93.42
LDA	93.80	99.00	94.05	87.59	94.44	93.66	93.96	95.90	99.33	94.67	90.65	95.95	93.42	93.42
XGB	98.99	99.95	99.04	97.98	98.90	99.17	98.79	94.25	99.30	93.51	88.38	92.31	94.74	93.88
kNN	99.86	100	99.86	99.71	100	99.72	100	93.68	98.13	92.41	87.32	97.10	88.16	88.16
SVM	93.95	98.22	94.20	87.88	94.46	93.94	93.96	93.68	99.06	92.72	87.14	93.33	92.11	94.90
RF	94.67	99.19	94.80	89.42	96.84	92.84	96.68	92.53	98.78	91.16	84.85	94.37	88.16	88.16
LR	95.39	99.29	95.58	90.76	95.84	95.32	95.47	92.53	98.52	91.39	84.80	92.00	90.79	90.79
QDA	98.27	99.77	98.36	96.55	97.56	99.17	97.28	89.66	95.40	88.75	79.56	84.52	93.42	93.42
NB	83.14	84.25	80.79	70.76	100	67.77	100	86.21	85.15	81.82	73.27	96.43	71.05	71.05

Table 10.5: Performance of classification models on combined dataset (small molecules and peptides).

Model overfitting

To ensure the robustness of our models and guard against overfitting, we performed Stratified K-fold cross-validation on the peptides, molecules, and combined datasets due to the inherent

class imbalance. We chose a five-fold split to maintain consistency with the train-test split strategy, which used 20% of the data for testing. The performance across folds was consistent, with no substantial deviation in accuracy, F1-score, or AUC-PR metrics. This indicates that the models generalized well across different subsets of data, providing confidence in their predictive power.

To thoroughly evaluate the contribution of different feature representations, we conducted an ablation study analyzing the individual, pairwise, and combined effects of Morgan fingerprints, RDKit descriptors, and Mol2Vec embeddings. This analysis was performed separately for small molecules, peptides, and the combined dataset. The findings indicate that Morgan fingerprints consistently yield the best performance for small molecules and the combined dataset. In contrast, Mol2Vec embeddings are more effective for the peptide dataset, likely due to their ability to better capture the sequential and contextual nature of peptide structures.

To provide context for our model's performance, we included a comparison with previously published algorithms evaluated on the same or overlapping datasets. These comparisons are summarized in Table D.1 of Appendix D, which includes models trained on datasets such as UMP442 and a patent dataset of 439 umami and 428 non-umami, and the UMP-IND dataset of 444 compounds (140 umami, 304 non-umami). Additionally, we have provided details of the hyperparameter optimization process for each task-specific best-performing feature representation. The tuned hyperparameters for the peptide dataset (using Mol2Vec), the small molecule dataset (using Morgan fingerprints), and the combined dataset (also using Morgan fingerprints) are documented in Table D.2 of Appendix D.

Webserver implementation

We have designed a user-friendly web server, UmamiPredict (https://cosylab.iiitd.edu.in/umami/) to facilitate the prediction of the umami taste of molecules and peptides dataset based on SMILES strings. The application integrates advanced front-end and back-end technologies to deliver an intuitive and responsive user experience. We have used React.js with Bootstrap for the frontend for responsive and interactive UI components. Python's Flask framework was used in the backend; the REST API processes the SMILES strings and returns predictions. RDKit was employed for molecular operations, and the Random Forest model (trained using Mol2Vec features) was integrated for prediction. MongoDB was integrated to store user queries, results, and molecular data. Papaparse was utilized for CSV parsing in batch processing, and JSME Molecular Editor was integrated for molecular drawing functionality.

Users can input individual, mini-batch, or batch files of SMILES strings to predict the umami taste of molecules and peptides. Batch prediction supports CSV upload for predicting multiple molecules simultaneously, with results sent to the user or displayed in the interface. Molecular drawing includes a molecular editor for graphical input of molecular structures. UmamiPredict's

architecture ensures seamless communication between the frontend and backend, while providing real-time feedback and error handling for robust performance. It stands as a comprehensive tool for researchers and developers in molecular science, facilitating rapid insights and data-driven decision-making.

10.2.3 Conclusions and Discussion

This study presents a computational framework for predicting the umami taste of compounds using peptide and small-molecule datasets. We effectively captured various structural and physicochemical properties of compounds by employing advanced molecular feature extraction methods—Morgan Fingerprints, RDKit descriptors, and Mol2Vec embeddings. Ensemble machine learning models, particularly LightGBM, XGBoost, and ExtraTrees, accurately classify the umami and non-umami compounds. On the peptide-only dataset, the LightGBM and XGB classifiers attained an accuracy of 92.13%, while on the small molecules dataset, LightGBM and ExtraTrees classifiers achieved an accuracy of 98.84%. When evaluating the combined dataset, LightGBM achieved the highest accuracy of 96.55%, indicating the effectiveness of integrating peptide and small molecule data for umami prediction. Higher performance across different datasets suggests that the feature extraction methods are robust and generalizable. Cross-validation and analysis of evaluation metrics confirmed that the models do not overfit and maintain consistent predictive capabilities across different data splits.

It is noteworthy that the QDA classifier exhibits lower performance. Upon further investigation, we observed that the feature distributions for peptides are highly non-Gaussian and exhibit multicollinearity, which likely violates the underlying assumptions of QDA (i.e., normality and distinct class-specific covariances). Additionally, the peptide feature space appears to be complex and potentially non-linearly separable, rendering it unsuitable for QDA's quadratic decision boundaries. These factors likely contributed to the model's inability to distinguish between classes effectively. Future work could address these issues by applying dimensionality reduction, feature transformation, or alternative modeling approaches better suited to the distributional characteristics of the data.

While the study advances existing methodologies by integrating peptide and small molecule datasets within a unified computational framework, it also highlights inherent challenges in modeling chemically diverse compound classes. Converting peptides into SMILES format facilitated standardized descriptor computation and model training alongside small molecules. However, this representation has limitations. Unlike small molecules, which are effectively characterized by atom-bond connectivity and functional group descriptors, peptides possess sequence-dependent properties such as residue order, backbone flexibility, and the potential to form secondary structures. These features play a crucial role in biological activity and taste perception but are not fully captured by conventional 2D cheminformatics descriptors derived

from SMILES strings. This disparity likely contributed to the comparatively lower predictive performance observed on peptide datasets.

Moreover, while this study primarily utilized two-dimensional molecular features, umami perception is influenced by additional factors, including three-dimensional conformation, stereochemistry, and dynamic molecular interactions with taste receptors. Incorporating 3D molecular descriptors—capturing spatial arrangement, surface properties, and conformational flexibility—could offer a more comprehensive representation of the physicochemical determinants of umami taste. Future work should consider integrating 3D structure-aware descriptors, dynamic simulations, or sequence-based peptide embeddings to better capture these nuances, particularly for peptides.

The increasing success of deep learning-based predictors, such as Umami-BERT and GNN-based models, suggests another direction for future improvements. While our approach demonstrates strong performance using classical models, it lacks the hierarchical representation learning capacity of transformers and graph-based architectures. Sequence-aware deep learning models can capture subtle context-specific patterns in peptide sequences or molecular graphs inaccessible to handcrafted or fixed-dimension descriptors. As such, benchmarking our approach against these modern architectures would provide deeper insight into its comparative strengths and limitations regarding generalizability, speed, and interpretability.

Interpretability itself remains a key challenge. While we performed feature importance analysis for models built on interpretable descriptors (e.g., RDKit), this was not feasible for abstract, high-dimensional embeddings such as Mol2Vec or Morgan fingerprints. These embeddings lack explicit feature labels, making it challenging to relate model outputs to underlying chemical properties. This limitation reduces the mechanistic interpretability of predictions, which is especially important for drug and food molecule design applications. Future extensions could incorporate SHAP-compatible embeddings or hybrid models that balance performance and explainability.

In conclusion, the proposed computational approach offers a robust and scalable tool for identifying umami taste compounds, with potential applications in the food and pharmaceutical industries for developing flavor enhancers and taste-modulating agents. By combining advanced feature extraction techniques with ensemble learning models and integrating chemically diverse datasets, this study provides a foundation for more accurate and generalizable taste prediction models. Future directions include expanding the dataset to encompass various taste compounds and exploring deep learning architectures capable of learning hierarchical molecular representations for improved mechanistic insight.

10.3 ToxinPredictor: Computational Models To Predict the Toxicity of Molecules

Human exposure to a wide range of potentially dangerous chemicals is now an inescapable part of modern life. Every day, our bodies are exposed to and must digest various chemicals [270], including medications, food additives, pesticide residues in food, toxins in drinking water, and dangerous airborne pollutants. These naturally occurring or synthetically created substances can enter the body in various ways, including oral consumption, cutaneous absorption, and inhalation. These exposures can cause adverse effects, including allergic reactions, non-acute and sub-acute poisoning, and, in more severe cases, long-term disability or death due to their mutagenic, carcinogenic, or toxic properties [271]. Furthermore, the rapid pace of industrial and technological advancements has resulted in hundreds of new synthetic chemicals, which can accumulate in ecosystems and eventually find their way into human populations, exacerbating health risks [272]. In addition to chemicals in pharmaceuticals and everyday products, human bodies are regularly exposed to toxic gases [273] and aerosols, such as carbon monoxide, lead, cigarette smoke, wood smoke, and workplace chemicals. Exposure to dietary contaminants, pesticides, and herbicides has also been associated with an increased risk of chronic disease and mortality [274].

Furthermore, the human body is vulnerable to exogenous toxins and internally produced toxic substances. For example, toxins and oncometabolites created during microbial metabolic activities in the gut have been linked to developing diseases such as diabetes, renal disease, and cancer [275]. This highlights the complexities of toxicity since dangerous compounds can originate externally from environmental and industrial sources and inwardly via natural biological processes. Chemical ingredients commonly used in human applications undergo rigorous clinical trials to ascertain their safety within specific limits. Traditionally, one of the simplest experimental measures of toxicity has been through bioassays involving animals injected with the toxin [276]. However, these experimental approaches are often tedious, time-consuming, and costly and have significant limitations, such as ethical concerns and the challenge of extrapolating animal data to humans [277, 278, 279, 280]. As a result, there is an increasing need for alternative methods to assess the toxicity of molecules based on their inherent chemical properties without extensive laboratory testing [281, 282, 283, 284, 285, 286, 287, 288, 289].

In this context, computational methods have emerged as a promising solution for predicting the toxicity of compounds using their structural and molecular characteristics [290, 291, 292]. These methods leverage molecular descriptors [293] and chemical fingerprints [294], which extract essential chemical and structural information about a molecule. DeepTox [283] utilized the Tox21 dataset [295], which contains in vitro toxicity screening results for thousands of compounds and authorized medications, to predict the toxicity of molecules using a deep neural network. Another method, eToxPred [296], predicts the synthetic accessibility and toxicity of

compounds. It is trained through publicly available FDA-approved datasets such as KEGG Drug [297], TOXNET [298], T3DB [299], and TCM [300], and achieved the AUC of 0.82 using a Tree classifier. ToxiM [285] is another tool that uses machine learning and chemoinformatics approaches, leveraging molecular descriptors and fingerprints to provide accurate toxicity predictions. It is based on a Random Forest model that outperforms other baseline models with an AUROC of 90% and is freely accessible through a web server. Furthermore, a multi-task deep neural network (MTDNN) model has been proposed, which simultaneously predicts in vitro, in vivo, and clinical toxicity by leveraging data from Tox21 [295], ClinTox, and RTECS datasets. MTDNN model achieves a high balanced accuracy of 96% for clinical toxicity prediction and employs the Contrastive Explanations Method and Genetic Algorithms to identify pertinent molecular substructures responsible for toxicity, providing interpretable explanations for the model predictions.

ToxCSM [284] is another machine learning-based tool that predicts toxicity across 36 endpoints by representing molecules using graph signatures, descriptors, and similarity scores. It performs well in classification tasks using the random forest model, achieving an AUC of 0.99 and a Pearson's correlation coefficient of 0.97 in the regression task. In a similar spirit, MolToxPred [286], an ensemble learning model, aimed to predict the toxicity of molecules by utilizing molecular descriptors and fingerprints. The stacked model, which integrates random forest, multi-layer perceptron, and LightGBM, achieved an AUROC of 87.76% on the test set and 88.84% on the external validation set, outperforming the base models. Additionally, it identifies structural alerts and potential biological pathways associated with toxicity.

The computational approaches for predicting toxicity are limited due to insufficient datasets. With the continuous introduction of new chemicals and the growing concern over long-term exposure risks, there is a need to enhance toxicity prediction methods. A machine-learning system can emerge as a valuable alternative capable of handling large amounts of data, offering faster and more cost-effective ways to predict the toxicological profiles of chemical compounds. In this study, we implemented machine learning and deep learning models to predict the toxicity of small molecules. Our models leverage advanced algorithms to learn patterns from the dataset, overcoming the limitations of existing models. This study not only demonstrates state-of-the-art performance in toxicity prediction but also paves the way for future advancements in the field by offering a scalable and robust solution for handling the growing chemical space. Further, we also developed a user-friendly web server, ToxinPredictor, to make our model accessible to researchers, industry professionals, and regulatory bodies.

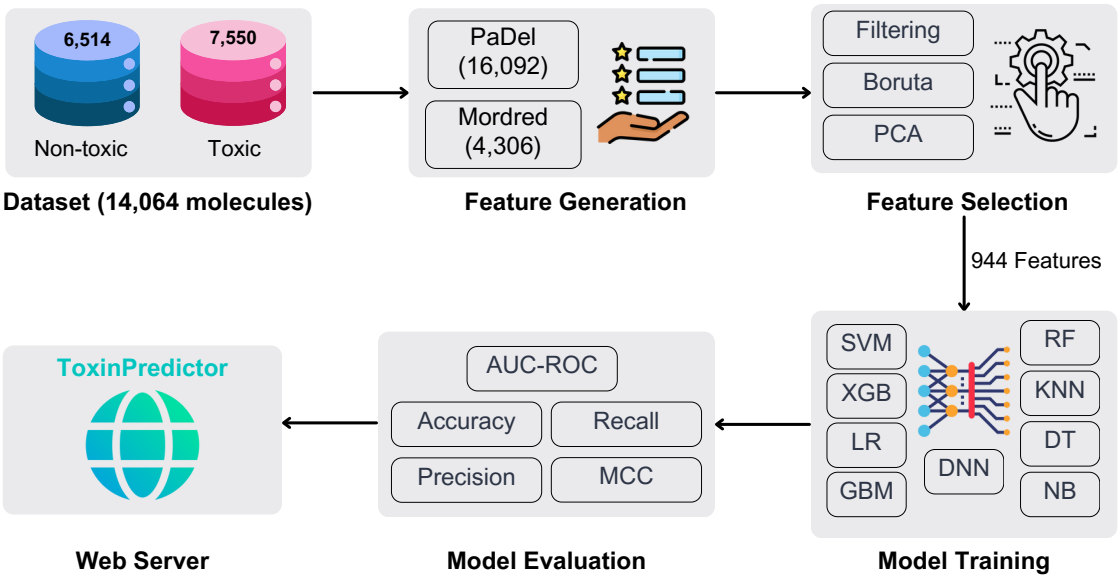

Figure 10.8: Schematic diagram of the computational protocol implemented for predicting the toxicity of small molecules. The workflow includes a dataset, feature generation, feature selection using Boruta and PCA, followed by model training using various machine learning algorithms and model evaluation. The best-performing SVM model is implemented in the web server, ToxinPredictor.

10.3.1 Material and Methods

Dataset collection

High-quality and accurately labeled datasets are foundational for developing and performing effective supervised learning models. In this study, we created a comprehensive dataset of 14,064 unique molecules, with 7,550 toxic and 6,514 non-toxic compounds for molecular toxicity prediction, with toxicity labels of 1 being toxic and 0 being a safe molecule. Following the approach outlined in the ToxiM [285], we manually collected 1,389 safe molecules from the RECON1 model in UCSD's BIGG database [301] and 3,678 toxic molecules from the T3DB database [299]. To enhance the diversity and balance of our dataset, we incorporated data from the MolToxPred [286], contributing 5,933 safe and 4,696 toxic molecules. We have used a Simplified Molecular-Input Line-Entry System (SMILES) [302] format to represent these molecules.

Data preprocessing

We preprocessed the dataset to ensure quality and consistency, including removing redundant entries, eliminating confounding entries with contradictory toxicity labels, and excluding molecules that caused computational errors during feature generation using PaDEL-Descriptor and RDKit.

Feature extraction

The chemical structures of molecules, represented as SMILES strings, were converted into numerical representations of molecules (molecular descriptors and fingerprints). We extracted the molecular features from SMILES using RDKit [243] and PaDELPy [242] libraries in Python.

Using RDKit, we generated 4,306 molecular descriptors. These descriptors encompass a wide array of physicochemical properties, including but not limited to molecular weight, LogP, number of rotatable bonds, and topological indices. We also computed 2D and 3D Morgan fingerprints (Extended-Connectivity Fingerprints, ECFPs) with a bit vector size of 2048. These fingerprints effectively capture local molecular substructures and their connectivity. We employed PaDELPy to generate 16,092 additional fingerprints to enrich our feature set further. These include CDK Fingerprints, MACCS Fingerprints, AtomPairs2D Fingerprints, Estate Fingerprints, PubChem Fingerprints, Klekota-Roth Fingerprints, and Substructure Fingerprints.

Our feature extraction process yielded a total of 20,398 features (4,306 from RDKit and 16,092 from PaDELPy), along with three categorical columns: SMILES (name of the SMILES string), Toxicity (toxicity label), and Source (the dataset it came from). This diverse feature set provides a comprehensive molecular representation, incorporating various aspects of molecular structure and properties, including topological, geometrical, and chemical characteristics.

Feature selection

The high dimensionality of molecular descriptors presents significant challenges in building toxicity prediction models, potentially leading to overfitting and inferior interpretability. To address this, we implemented the feature selection process to identify the most relevant features, enhancing the model's accuracy. The dataset consisted of a total of 20,401 features for 14,064 molecules.

Firstly, we removed the duplicate columns, resulting in the reduction of features to 13,721. Removing redundant data that could have skewed the model's performance is critical. We applied a low variance threshold of 0.09, removing features that exhibited minimal variability across the dataset. This step further reduced the number of features to 13,719, ensuring that essential features were retained. To address the issue of multicollinearity, we eliminated the redundant and highly correlated features using a correlation threshold of 0.84, significantly reducing the feature set to 10,140. This step was critical since significant correlations between features might lead to unstable model estimates and overfitting. Relevance filtering was implemented to ensure that only features highly correlated (correlation coefficient threshold ≥ 0.005) with the target variable (toxicity) were retained. The filtering method yielded a final set of 9,260 molecular characteristics.

Boruta algorithm for feature importance To capture linear and non-linear relationships

between features and toxicity, we employed the sophisticated Boruta algorithm [303]. This feature selection method, based on random forests [304], offers a robust approach to identifying essential features. The algorithm creates shadow features by randomly shuffling values within each column, then trains a random forest classifier on the combined dataset of real and shadow features. Through multiple iterations, features that consistently outperform the shadow features are identified as necessary. This process revealed the most relevant 1,185 features for toxicity prediction, providing a feature set optimized for predictive power and interpretability.

Dimensionality reduction through Principal Component Analysis We applied Principal Component Analysis (PCA) [305] to the Boruta-selected features as a final step in our feature engineering process. This technique further reduced dimensionality while retaining the maximum amount of information. We significantly compressed our feature space by setting a threshold to retain components that explained 99% of the variance. This dimensionality reduction step yielded a set of principal components, striking a balance between data compression and information preservation, crucial for accurate toxicity prediction. Finally, we were left with 944 principal components.

The rigorous feature selection procedure yielded a relevant feature set for toxicity prediction. This strategy reduced the danger of overfitting, improved model interpretability, and, ultimately, improved the prediction performance of our models.

Model implementation

To predict the toxicity of molecules, we split the dataset into 80:20 ratios for training and testing. We implemented the following nine models: Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Gradient Boosting Machine (GBM), XGBoost (XGB), Support Vector Machine (SVM), K-Nearest Neighbors (kNN), Naive Bayes (NB), and Deep Neural Network (DNN). DNN is a deep-learning model, while the rest are machine-learning models. We used grid search to optimize the hyperparameters of machine learning models (see Table D.6 in Appendix D). Grid search examines a specific subset of the hyperparameter space [306], evaluating all possibilities to ensure each model's best set of parameters.

For the DNN model, we employed Bayesian Optimization for hyperparameter tuning via the Keras Tuner library [307]. This approach efficiently explored the hyperparameter space, optimizing the number of layers (ranging from 2-6), number of units per layer (32-512), dropout rates (0.3-0.5), and learning rate (10-4 to 10-2). Based on past assessments, Bayesian Optimisation iteratively updates the search space, resulting in a more concentrated and efficient search than the classical grid search.

The final DNN architecture incorporated multiple dense layers with L2 regularization ($\lambda = 0.01$) to prevent overfitting. LeakyReLU activation functions ($\alpha = 0.01$) were applied to handle the

vanishing gradient problem. Layer Normalization was applied to each dense layer to improve training stability and speed up the convergence. The training process for the DNN was optimized using callbacks. We used the ReduceLROnPlateau callback, which reduced the learning rate by 0.2 when validation loss plateaued for ten epochs. EarlyStopping callback was used to monitor validation loss, prevent overfitting, and ensure efficient convergence.

To address the class imbalance in our dataset, we applied the Synthetic Minority Over-sampling Technique (SMOTE) [308] before model training. SMOTE creates samples for the minority class, balancing the class distribution and enhancing model performance in under-represented classes.

Evaluation metrics

The performance of the models was evaluated using threshold-dependent parameters (Accuracy, Precision, Recall, Matthews Correlation Coefficient, and F1 score) and threshold-independent parameters (Area Under the Receiver Operating Characteristic Curve). Accuracy is a key parameter used to assess the correctness of the model by computing the proportion of accurate optimistic and true pessimistic predictions. P is the ratio of true positive predictions to all positive predictions made by the model. Recall (R), on the other hand, calculates the percentage of accurate positive predictions among all real positive cases. The F1 score, the harmonic mean of accuracy and recall, is a balanced measure that considers false positives and false negatives, providing a single metric for evaluating the precision-to-recall trade-off. Matthews Correlation Coefficient (MCC) measures the quality of binary classifications using true and false positives and negatives. The MCC produces a result ranging from -1 to +1, with +1 indicating a perfect prediction, 0 indicating no better than a random prediction, and -1 indicating complete disagreement between prediction and observation.

Area Under the Receiver Operating Characteristic Curve (AUROC) indicates the likelihood that the model will rank a randomly selected positive instance higher than a randomly picked negative instance. This threshold-independent metric gives a complete picture of the model's discriminative capabilities across all conceivable classification thresholds, making it especially useful for evaluating different models and their robustness in diverse circumstances.

10.3.2 Results

Performance of classification models

We implemented eight machine-learning models, such as Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Gradient Boosting Machine (GBM), XGBoost (XGB), Support Vector Machine (SVM), K-Nearest Neighbors (kNN), and Naive Bayes (NB), and one deep-

learning model, Deep Neural Network (DNN).

Model	Training						Testing					
	AUC	Acc	P	R	MCC	F1	**AUC**	Acc	P	R	MCC	F1
SVM	**96.9**	90.4	92.6	87.8	80.9	90.1	**91.7**	85.4	87.5	82.5	70.8	84.9
DNN	**97.9**	93.6	96.1	90.9	87.4	93.4	**90.4**	83.8	86.2	80.6	67.8	83.3
LR	**93.4**	85.9	87.8	83.3	71.8	85.5	**90.0**	82.8	83.9	81.2	65.7	82.5
XGB	**99.6**	96.7	97.9	95.5	93.4	96.7	**90.0**	82.6	84.8	79.5	65.3	82.0
GBM	**95.8**	89.0	91.7	85.9	78.2	88.7	**89.3**	82.1	84.6	78.5	64.4	81.4
RF	**99.9**	97.7	98.1	97.2	95.4	97.7	**87.5**	80.2	83.0	75.9	60.6	79.3
kNN	**93.6**	85.1	87.5	81.8	70.3	84.6	**87.5**	80.2	83.0	75.9	60.6	79.3
DT	**93.7**	87.6	92.6	81.8	75.8	86.9	**79.3**	75.5	78.4	70.5	51.3	74.2
NB	**64.0**	61.4	60.4	66.1	22.9	63.1	**64.7**	61.7	60.9	65.4	23.5	63.1

Table 10.6: Performance comparison of different classification models on training and testing datasets. Metrics: AUC = Area Under the Receiver Operating Characteristic Curve, Acc = Accuracy, P = Precision, R = Recall, MCC = Matthews Correlation Coefficient, and F1 = F1-Score.

Table 10.6 shows the detailed performance metrics of classification models, including Accuracy, Precision, Recall, F1-Score, and the Area Under the Receiver Operating Characteristic Curve (AUROC). The SVM model outperformed other models, achieving the highest AUROC score of 91.65% and an accuracy of 85.36%. This shows that the SVM model effectively differentiates between the classes in our dataset, with superior performance in terms of both true positive rate and false positive rate. Following the SVM, DNN performed well, with an AUROC score of 90.37%. This demonstrates DNNs' efficacy in handling high-dimensional data and generating clear classification boundaries.

Performance comparison on previous literature

We implemented the model pipeline (Figure 10.8) on two datasets separately, i.e., ToxiM and MolToxPred. ToxiM contains 1,389 safe molecules and 3,678 toxic molecules, while MolToxPred has 5,933 safe and 4,696 toxic molecules. We preprocessed the datasets, extracted the features, and then applied the feature selection strategy individually. Initially, we had 11,978 and 13,606 features in ToxiM and MolToxPred, respectively, which were reduced to 382 and 531 after the selection strategy (see Table D.3 in Appendix D).

We trained our models on these datasets and compared the performance of our models on previously applied pipelines. Table 10.7 shows the performance of our DNN model, and Table D.4 and Table D.5 in Appendix D show the detailed performance of all models on ToxiM and MolToxPred datasets. Compared to previous literature, our DNN model achieved better performance with AUROC scores of 98.47% and 89.13% on ToxiM and MolToxPred, respectively, while the SVM model performed similarly with AUROC scores of 98.7% and 89.7% on the same datasets.

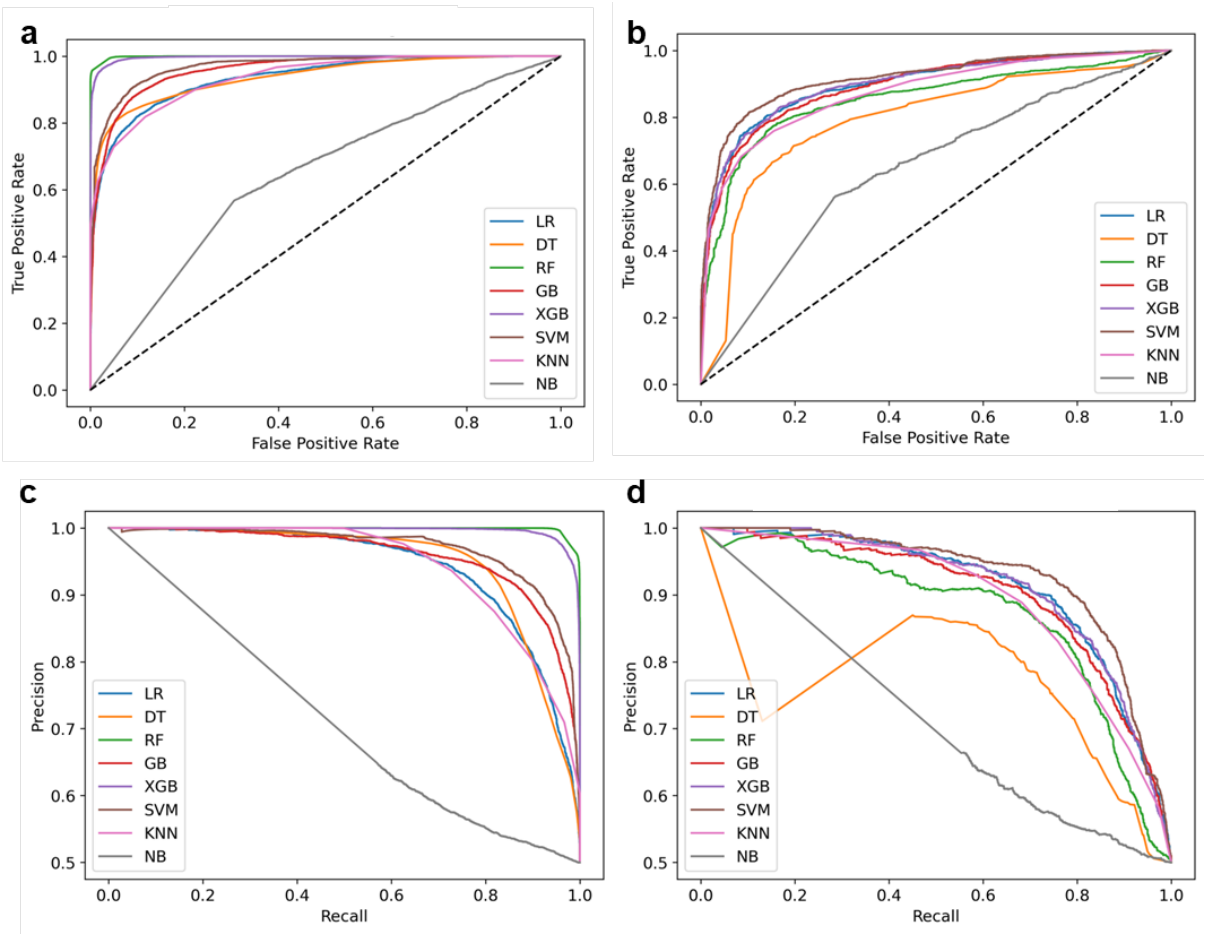

Figure 10.9: Area Under the Receiver Operating Characteristic Curve (AUROC) and Precision-Recall (PR) curves. (a) AUROC of the training dataset, (b) AUROC of the testing dataset, (c) PR curve of the training dataset, and (d) PR curve of the testing dataset.

Dataset	Previous Performance						Toxin Predictor					
	AUC	Acc	MCC	P	R	F1	AUC	Acc	MCC	P	R	F1
ToxiM	**90**	93	84	95	NA	NA	**98.47** (DNN)	96.22 (DNN)	92.60 (DNN)	98.7 (DNN)	93.7 (DNN)	96 (DNN)
							98.7 (SVM)	95.4 (SVM)	96.6 (SVM)	94.0 (SVM)	90.8 (SVM)	95.3 (SVM)
MolToxPred	**87.76**	80.91	61	80.99	71.18	76.4	**89.13** (DNN)	81.28 (DNN)	62.61 (DNN)	82.2 (DNN)	81 (DNN)	80.6 (DNN)
							89.7 (SVM)	82.9 (SVM)	83.9 (SVM)	81.5 (SVM)	65.9 (SVM)	82.7 (SVM)

Table 10.7: Comparison of previous performance and our SVM and DNN models on ToxiM and MolToxPred datasets.

10.3.3 SHAP Feature Analysis

We used SHAP analysis to analyze the most relevant features influencing the predictions of our best-performing model, SVM. SHAP provides a unifying framework for interpreting the results

of complex machine learning models by giving an important value to each feature for a specific prediction. This interpretability is critical in toxicity prediction, as knowing the significance of individual molecular characteristics can shed light on the underlying mechanisms that cause toxicity.

By applying SHAP analysis [309] to our SVM model (Figure 10.10), we identified the most relevant molecular descriptors and structural attributes contributing to toxicity predictions. These descriptors capture structural and physicochemical features of molecules, providing insights into the underlying factors that influence toxicity. Among the most important aspects are PubChem fingerprints (e.g., PubchemFP255, PubchemFP179, PubchemFP186), which encode the existence of certain substructures or functional groups known to have toxicological consequences. These can contain reactive moieties, aromatic systems, or specific functional groups such as halogens or hydroxyl groups, which can influence how a molecule interacts with biological systems. The substructure fingerprints (e.g., SubFP84) are based on the presence of preset chemical pieces, allowing the model to identify key toxicophores—substructures that are typically associated with negative consequences, such as electrophilic groups that may react with protein or DNA. Similarly, graph-based fingerprints (e.g., GraphFP695, GraphFP311) account for the entire molecular connectivity and topology, including cyclic structures, atom adjacency, and bond patterns critical in defining a molecule's chemical behavior and reactivity.

Furthermore, extended fingerprints (e.g., ExtFP228, ExtFP76) probe deeper into the molecular environment by considering circular atom neighborhoods, providing a more localized view of how atoms interact with their surroundings. This can help identify a molecule's sections where toxicity is more likely to occur, such as certain atom groups involved in binding or catalytic events in a biological environment. BCUT descriptors (e.g., BCUT2D_LOGPLOW) capture broader physicochemical features such as lipophilicity, which is important for a molecule's capacity to permeate cell membranes and accumulate in biological systems, hence determining toxicity. The combination of these descriptors enables the model to comprehensively examine both structural and physicochemical properties, providing a detailed understanding of the molecular mechanisms influencing toxicity predictions.

The insights gained from SHAP analysis enhance the transparency and trustworthiness of our model, making it more than just a black-box predictor. This enables one to delve into the specific features that lead to higher toxicity predictions, providing a scientific basis for further investigation and hypothesis generation. Additionally, these insights can guide the optimization of chemical compounds by highlighting which features to modify to reduce toxicity, thereby supporting safer and more effective drug design. SHAP analysis is further presented visually within the ToxinPredictor platform, offering users interactive plots that clearly show the impact of each feature on the prediction outcomes. This user-friendly visualization enables both expert and non-expert users to intuitively understand the model's reasoning, fostering greater confidence in the predictions and facilitating more informed decision-making in research and development.

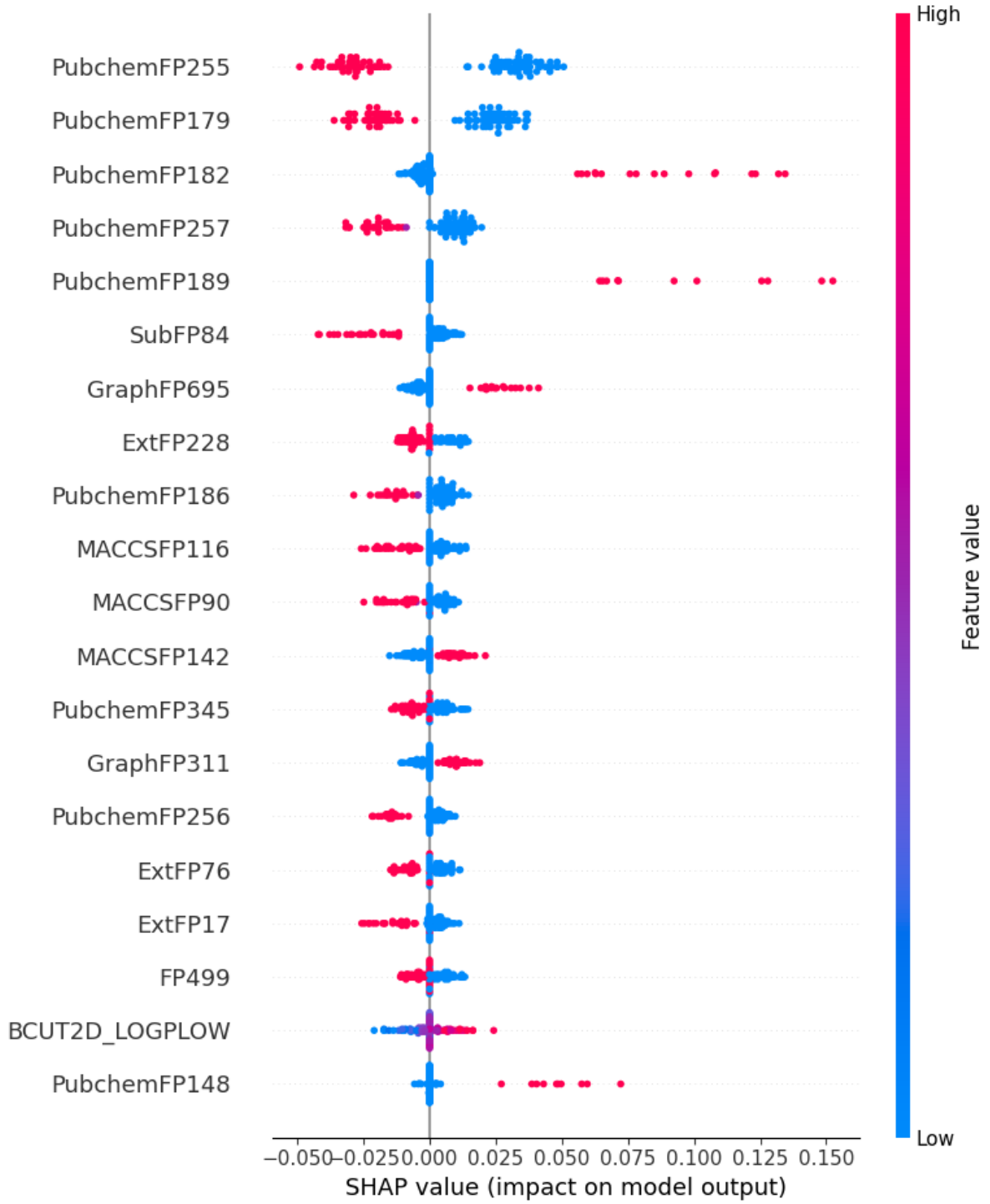

Figure 10.10: SHAP analysis of SVM model highlighting the contribution of key features in predicting molecule toxicity. The Boruta feature selection method was applied, and the SHAP values indicate the impact of each feature on model predictions, where positive values increase the likelihood of toxicity and negative values decrease it.

10.3.4 Webserver Implementation

We developed a robust, scalable, and user-friendly web server to predict the toxicity of molecules, ToxiPredictor (https://cosylab.iiitd.edu.in/toxinpredictor/). Bootstrap and React have been used for the frontend, while Flask is used for the backend. The web server comprises several key components, including a landing page that illustrates the model's workflow, a contact section, and a prediction interface. One of the most critical features is the prediction interface, which provides flexibility to users. Users can input a single SMILES string to assess the toxicity of molecules. ToxiPredictor allows users to upload a CSV file containing multiple SMILES for toxicity assessment for larger datasets. This design ensures flexibility in handling individual compounds and larger datasets, enhancing the platform's utility for various applications.

One of the standout features of the webserver is the integration of JSME [310], which enables molecular drawing directly within the interface, providing an intuitive tool for researchers to visualize and assess molecular structures. The web server allows for real-time data processing, enabling users to assess the toxicological risks of small molecules quickly and efficiently. This platform enhances the practical utility of our model, offering a valuable tool for accelerating research, ensuring chemical safety, and supporting decision-making processes across industries and regulatory bodies.

10.3.5 Conclusions and Discussion

This study presents ToxinPredictor, a machine learning-based method for predicting the toxicity of small molecules, showcasing its potential in aiding drug development and environmental safety assessments. Utilizing a well-curated dataset of 7,550 toxic and 6,514 non-toxic molecules, the model effectively learned from their structural and chemical properties, offering a reliable solution for toxicity prediction. Boruta and PCA for feature selection enhanced the model's predictive capabilities, outperforming existing tools. SVM model achieved a state-of-the-art performance with an AUROC of 91.7%, F1-score of 84.9%, and accuracy of 85.4%. We applied the SHAP analysis to the best model (SVM) and identified the most relevant molecular descriptors and structural attributes contributing to toxicity predictions. Despite challenges in data quality, ToxinPredictor paves the way for more efficacious toxicity predictions, contributing to safer drug development and environmental health.

The future directions of this study include integrating biological pathways data, using transfer learning for rare molecules, and implementing multi-task learning for predicting multiple toxicity endpoints. Expanding datasets to encompass a broader range of environmental pollutants and toxins and incorporating experimental validation will further strengthen the model's applicability to drug development and environmental safety assessments. The code and dataset are available on GitHub.

Chapter 11

Dish Detection in Indian Food Platters

11.1 Introduction

Diet is central to the epidemic of lifestyle disorders such as obesity, type 2 diabetes, and cardiovascular conditions. Scientific evidence suggests that calorie restriction is among the most impactful preventive and remedial strategies. One of the significant challenges for effective dietary management is keeping track of diet and, therefore, calorie consumption. Implementing high-performance object detection models embedded in internet-enabled smartphones presents an excellent opportunity to create computational solutions for automated diet logging and nutrition management. Among other culinary traditions, Indian cuisine is known for its visually complex platters, making dish detection a challenging problem. Collecting platter images with manual dish annotations is a necessary precursor for building a computational pipeline for accurate dish classification and detection.

Building on the previous research that converged on a state-of-the-art model for the ten most popular Indian dishes [311], we aimed to scale up labeled data compilation and identify the best model by comparing various architectures. The model thus identified is a potent candidate for implementation in mobile devices and can help achieve public health goals through dietary interventions. Figure 11.1 depicts a schematic of the computational framework for dish detection in food platters, culminating in a proof-of-concept mobile implementation. Such a computational framework has diverse applications for food recommendation systems, diet logging, nutritional interventions, and mitigation of lifestyle disorders.

The creation of the proposed framework entails compiling platter images, pre-processing, dish annotations, and implementing models for dish classification and detection. Image classification is a core task of detection models and has become a tractable problem with the development of deep convolutional neural networks [312] and the availability of large-scale, hand-labeled datasets such as ImageNet [313]. Implementing deep convolutional neural networks for multi-label classification has yielded promising results [314].

Pictures of food platters are complex owing to variations in perspective and patterns in dish arrangements. The application of food image classification has attracted much attention recently. Early applications of CNN architectures (AlexNet, GoogLeNet, and ResNet) for food image classification [315] yielded poor results. Amato et al. [316] used a pre-trained GoogLeNet [317] convolutional neural network model to analyze trends in food images from social media. ETHZ

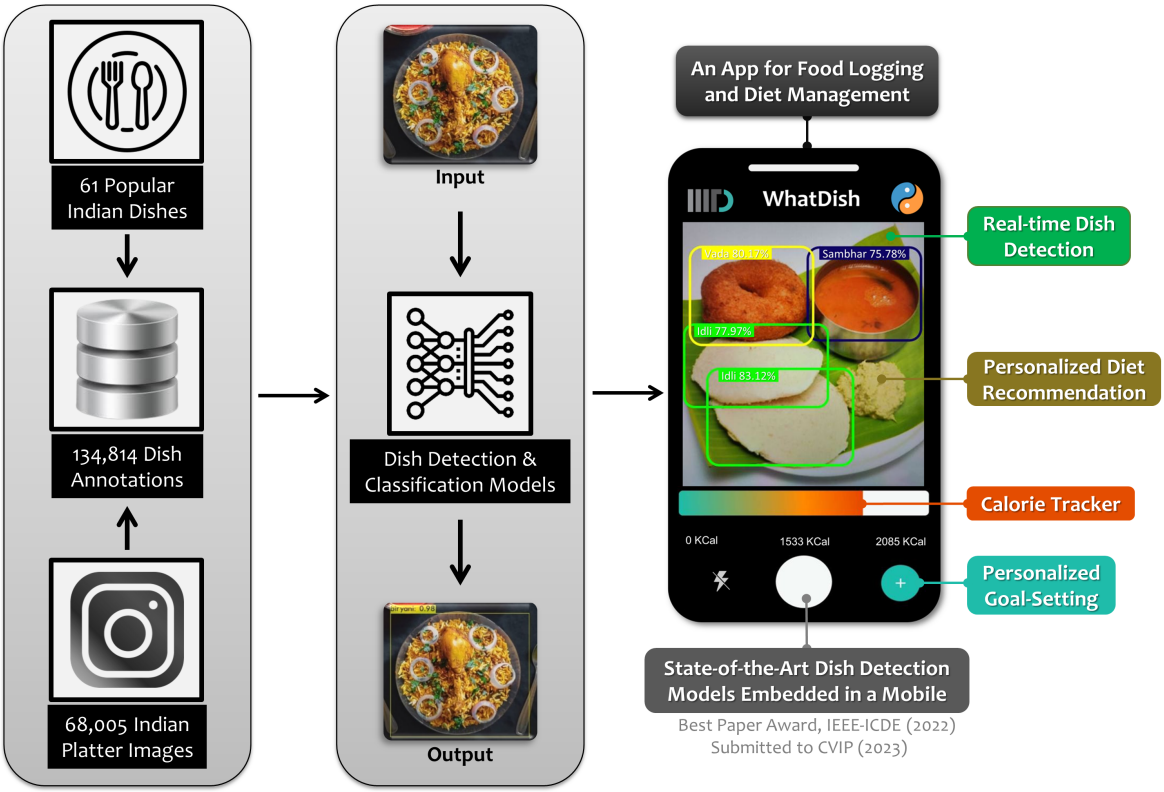

Figure 11.1: A computational framework implemented for automated dish detection. As a case study, the framework was implemented for Indian cuisine, starting with data compilation, pre-processing, dish annotations, and implementation of classification and dish detection models. Further, by embedding the state-of-the-art dish detection model, a mobile application was designed aimed at automated food logging and diet management. The app is capable of real-time dish detection, calorie tracking, and personalized goal-setting.

food-101 [318] dataset comprising 101,000 food images across 101 classes was used for fine-tuning the model before classifying images using KNN. With a similar spirit, Minija and Emmanuel [319] implemented a support vector machine model to classify food images from the FoodLog dataset (6,512 images) with an accuracy of 95%.

Kagaya et al. [320] trained a CNN model and outperformed other baseline models with an average accuracy of 73.7% for ten classes. In another experiment, a pre-trained AlexNet model was implemented on the UEC-FOOD-100 dataset [321] containing 100 Japanese food classes to achieve an accuracy of 72.26% [322]. In another AlexNet implementation, Yanai et al. [323] applied image classification for Japanese food datasets UEC-FOOD-100 and UEC-FOOD-256 [324] (each containing 100 images per class) to achieve the top-1 accuracy of 78.8% and 67.6%, respectively. Myers et al. [325] presented a protocol for food classification using a pre-trained GoogLeNet [317] model fine-tuned on Food101 to achieve the top-1 accuracy of 79%. In another oriental food classification implementation [326], a pre-trained Inception model fine-tuned on Thai food images (THFOOD-50) was used to achieve an accuracy of 80.34%. Based on

these prior studies, we implemented diverse model architectures for multi-label classification on the IndianFood61 dataset comprising an extensive compilation of Indian platter images labeled for the 61 most popular Indian dishes.

Beyond image classification, object detection is vital in computer vision tasks such as segmentation and object tracking, among other applications. Among the earliest studies, object detection has been used for real-time human face detection [327, 328]. Other applications have implemented SIFT [329], HOG [330], and SURF [331] techniques on ImageNet and COCO datasets. With the advances in deep learning, two-stage (SPPNet [332], Fast R-CNN [333], FPN [334]) and one-stage detectors (YOLO [335], SSD [336], RetinaNet [337], CornerNet [338], DETR [339]) were introduced for enhanced object detection.

Object detection has immense utility for industry and public health when applied in the context of food platters. Among the earliest applications of food detection, Matsuda et al. [321] used traditional computer vision techniques (SIFT, HOG) to achieve an accuracy of 55.8%. Another study proposed a food localization and recognition method with an activation map to detect food using bounding boxes [340]. With a dataset of 60 traditional Chinese food items (BTBUFood-60), Cai et al. [341] implemented a Faster R-CNN to achieve an accuracy of 67%. In the context of complex Indian food platters, the earliest study used Single Shot Detector and Inceptionv2 on a dataset with 60 classes (70 images per class) to achieve the mAP score of 73.8% [342]. This study had two significant flaws concerning the relevance and quantity of the data. Among the 60 classes, eight were not traditional Indian dishes (pizza, pasta, noodles, cake slices, ice cream, brownie, mayo, and ketchup). They included nine trivial or irrelevant food classes such as tomato, cucumber, water, lemon slice, onion sliced, boiled egg, milk, chilly, and juice.

Despite its rich and diverse culinary heritage, the shortage of labeled collections of Indian food images is a significant challenge in developing accurate and effective models for food classification, dish detection, and recipe recommendations. We have created the IndianFood10 dataset of around 10,000 manually annotated food images of ten traditional Indian dishes and implemented a YOLOv4 model. We further extended the dataset to 61 traditional Indian dishes. As proof of the concept, we present a framework implementing state-of-the-art deep-learning models for dish detection in Indian food platters by comparing diverse network architectures on a rich dataset of manually annotated images.

11.2 Materials and Methods

11.2.1 Data compilation and annotation

We identified the most popular traditional Indian dishes (10 and 61) across its regional cuisines. While the documented number of Indian recipes is in the range of 2500-5800 [6, 21, 22], the

staple dishes that are most frequently consumed tend to be much fewer. With over 1 billion users sharing more than 100 million posts daily, Instagram is one of the most prolific online sources of user-uploaded pictures along with their hashtag descriptors [343]. We scraped images from Instagram using relevant hashtags that potentially contained traditional Indian food platters using Python selenium and the requests library.

Thus compiled, our 'IndianFood10' and 'IndianFood61' datasets consist of 11,547 and 68,005 multi-class images with 10 and 61 food classes ($\sim$1000 images for each class) respectively. The average number of platter images for each dish class was 1115 $\pm$ 500 (For details, see Table E.1 in Appendix E). Among the outlier classes were *idli* (494) and *gulabjamun* (533), *plain rice* (4067), and *Indian bread* (3074). We manually annotated and labeled each image with a bounding box for every dish class using makesense.ai, an open-source software [344]. The number of annotations for each dish class reflects its occurrence across the platters. On average, a dish occurred in platters 2210 $\pm$ 1584 times (For details, see Table E.1 in Appendix E). While dishes such as *dal* (853) and *rasam* (798) were found with low occurrence, *momos* (6496) and *barfi* (6375) were over-represented. The annotated images and corresponding text files were saved in YOLO format. The text file includes the food class ID and coordinates of the bounding boxes for each food item in the image.

11.2.2 Multi-label classification

Image classification involves their categorization based on relevant features. Primitive classification strategies relied on traditional methods such as Bag-of-Words, PASCAL VOC, and SIFT. With advances in deep learning, CNN-based image classification techniques with end-to-end learning pipelines are increasingly used. These protocols overcome the bottleneck of manual extraction of image-specific features associated with traditional methods. We implemented the below-mentioned CNN-based models that list the dishes appearing in an image without locating their coordinates. This task is computationally inexpensive than object detection.

AlexNet: AlexNet is a 60 million parameter model consisting of 8 layers with five convolutional layers, two fully connected hidden layers, and one fully connected output layer. Krizhevsky et al. [312] demonstrated that learned features could outperform manually designed features contrary to the known notion in computer vision.

SqueezeNet: SqueezeNet has fewer parameters than AlexNet but implements fire modules, compression, and downsampling techniques to achieve enhanced accuracy. SqueezeNet1_1 combines 1x1 and 3x3 convolutional filters in the fire modules and has residual connections between fire modules that help reduce the network's computational complexity.

VGGNet: Visual Geometry Group or VGG [345] has a similar architecture to that of AlexNet with a large number of parameters and weight layers that enable improved performance. It

consists of identical convolutional layers and a maximum pooling layer; the convolutional layer maintains the input height and width while the pooling layer halves it. VGG-16 contains 16 weight levels, five convolutional blocks in series, and two fully connected layers with 4096 dimensions. VGG-19 has a similar architecture to VGG-16 with 19 weight layers.

ResNet: Residual network [346] is a neural network with additional residual layers that enhance object classification performance. ResNet34 (with 34 weighted layers) involves shortcut connections to convert a plain network into its residual network counterpart. The residual block contains 3*3 convolutional layers and a batch normalization layer, and the ReLU activation function follows each convolutional layer. ResNet50 architecture is similar to ResNet34 with one significant difference–the building block is modified into a bottleneck design to handle training duration. With a 3-layer block, ResNet50 is more accurate than ResNet34. ResNet152 has an improved architecture with the addition of a 1x1 convolutional layer over and above the 3x3 layers.

DenseNet: DenseNet architecture is similar to ResNet, with one significant difference. DenseNet concatenates the previous layer's output with the future layer, whereas ResNet uses an additive method to merge layers.

11.2.3 Object detection

Object detection is a critical problem in computer vision used to locate and identify all objects in an image by combining object localization and classification. Previously, handcrafted traditional methods such as Viola-Jones detectors, HOG detectors, and Deformable Part-based Models were used due to a lack of effective image representation techniques. We have implemented state-of-the-art object detection models to detect dishes from images of traditional Indian platters.

DETR: DETR (DEtection TRansformer) is a transformer-based object detection algorithm that was introduced in 2020 by the Facebook AI Research team [347]. DETR predicts the set of objects present in an image, along with their class labels and precise bounding boxes, as opposed to traditional object detection algorithms that use Region Proposal Networks or anchor boxes to predict object locations. It consisted of two major components–an encoder and a decoder. The encoder is a convolutional neural network that extracts image features, and the decoder is a transformer-based network that processes the image features and returns the final object detection. DETR aligns the predicted object with the ground truth object using a bipartite matching loss function, allowing it to accurately detect objects. It has achieved remarkable performance on the object detection benchmark (COCO). It is an end-to-end trainable architecture that eliminates the need for post-processing steps such as Non-Maximum Supervision.

Faster R-CNN: Faster R-CNN is a two-stage object detection model, where the first stage (Region Proposal Network) generates object proposals and the second stage (region-based CNN)

classifies the proposed regions and generates bounding boxes [333]. Such a two-stage approach helps improve the accuracy of the object detection model [333, 348].

RetinaNet: RetinaNet is a one-stage model that uses a Feature Pyramid Network to generate feature maps at different scales to make predictions. Each level produces a set of anchors (bounding boxes) of different sizes and aspect ratios to localize objects in the image. These anchors are then passed through a series of convolutional layers to predict each anchor's class probabilities and offsets. RetinaNet uses a standard CNN (ResNet-50/EfficientNet) as its backbone model [337].

YOLO: YOLO (You Only Look Once) is a one-stage object detection model with an architecture similar to a Fully Convolutional Neural Network. It consists of an input layer, a backbone network (CSPDarkNet53), a neck (Spatial Pyramid Pooling, Path Aggregation Network), and a head (locate bounding boxes and make the predictions). YOLOv4 [349] contributes mosaic data augmentation, which leads the model to find smaller objects and pay less attention to the environment. It uses the Spatial Attention module to improve the accuracy and speed compared to previous YOLO models. YOLOv5, developed by Ultralytics, is a faster model than YOLOv4. It introduces a new backbone architecture, CSPNet, based on the PyTorch framework, which reduces the computations needed for detecting an object. YOLOv5 further improves object detection accuracy with three types of data enhancements: scaling, color space adjustment, and mosaic enhancement. YOLOv7 [350] has a faster and more robust network architecture with an optimized feature integration method, more accurate object detection performance, a better loss function, and increased model training efficiency. YOLOv8 is the latest version of YOLO, which introduces new features and improvements to enhance performance, flexibility, and efficiency. YOLOv8x is a variant of YOLOv8 consisting of 68.2M parameters. YOLO models are state-of-the-art object detection models with impressive speed and performance. The main difference between them lies in the details of architecture and the techniques used for performance optimization.

11.2.4 A computational framework for dish detection

We implemented pre-trained state-of-the-art models on the ImageNet dataset and fine-tuned them on the IndianFood61 dataset for the multi-label classification. All images were resized to 512*512 pixels. The class labels and targets were represented as one-hot vectors. For fine-tuning the model, we modified the last layer by converting it into a linear layer with an output dimension equal to the number of classes, adding a sigmoid layer to obtain probabilities between 0 and 1 for each class, and using the pre-trained weights for the training of models. The output is a vector of dimension 61 (number of classes), with each entry representing the probability of each class.

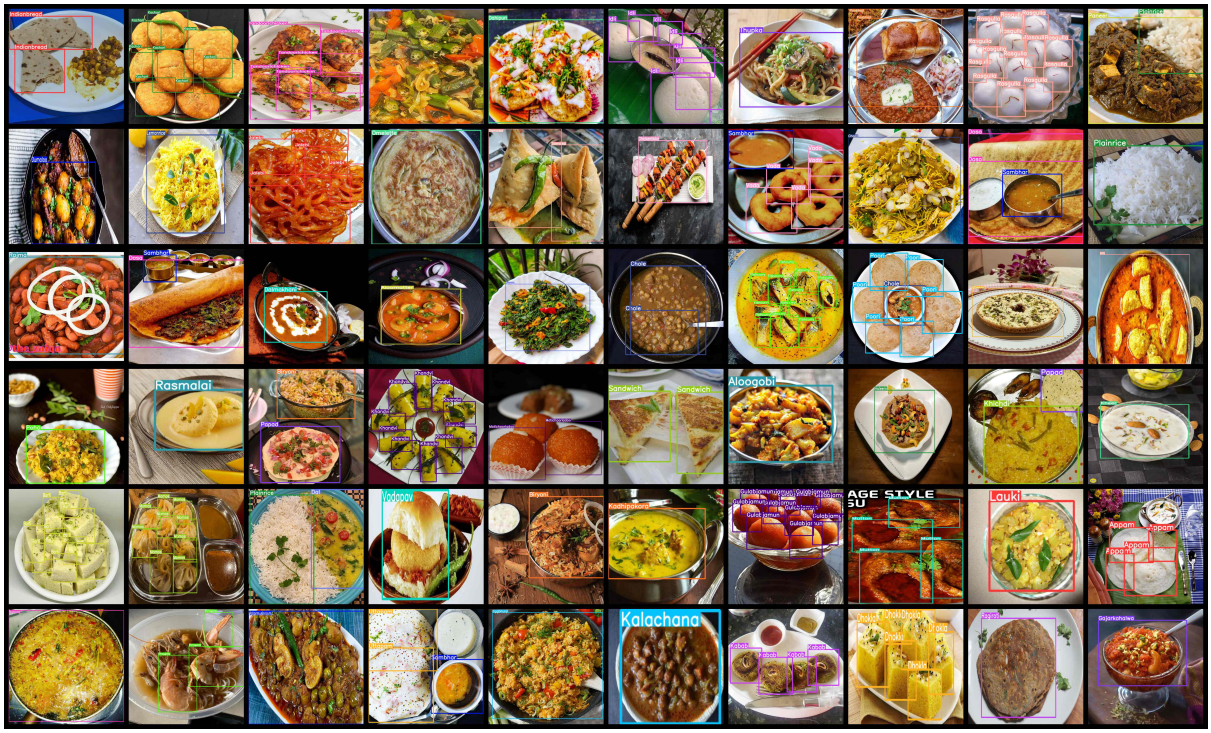

Figure 11.2: An illustration of the dish detection model with bounding boxes predicted for Indian dishes. The picture shows 60 of the 61 dishes from the IndianFood61 dataset.

We used a two-stage process for fine-tuning the models. First, we froze the weights for the backward layers and trained the weights for the last layer with a higher learning rate. Next, we unfroze the model and train all the layers with a lower learning rate. For finding the learning rate, we used the learning rate finder proposed by Leslie Smith in 2015 [351]. The basic idea here is to start with a small learning rate and gradually increase it in small batches. We used BCELoss (Binary Cross Entropy Loss) with Adam optimizer for the loss function.

To detect each item from an Indian food platter, we split the IndianFood10 and IndianFood61 datasets with a 90:10 ratio for training and testing. We implemented Faster R-CNN, DETR, RetinaNet, YOLOv4, YOLOv5, YOLOv7, and YOLOv8 object detection algorithms. The YOLO models were trained for 100 epochs using ReLU and sigmoid activation functions. The deep-learning models were implemented using a V100 server with 32 GB Tesla V100 GPU and Intel(R) Xeon(R) Silver 4210 CPU.

11.2.5 Evalation metrics

Precision (eq 11.1), Recall (eq 11.2), F1 score (eq 11.3), and mAP score (eq 11.4) were used for evaluating the performance of the models. mAP score represents the mean of average precision across the classes. The mAP score of multi-label classification differs from the evaluation of object detection models because we use Intersection over Union (IoU) as the threshold for

detection. IoU is a metric measuring the overlap between the predicted and ground truth boxes. We set the IoU threshold to 0.5 for calculating the precision and mAP score for object detection.

$$Precision = \frac{TP}{TP + FP} \tag{11.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{11.2}$$

$$F1 = \frac{2 * P * R}{P + R} \tag{11.3}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{11.4}$$

11.3 Results

11.3.1 Multi-label classification on IndianFood61 dataset

We compared the performance of ten deep-learning models for multi-label classification. Table 11.1 presents the performance of models on the IndianFood61 dataset. Resnet152 presented a state-of-the-art performance with mAP score, F1 score, and precision of 84.51%, 88.01%, and 90.56%, respectively. Dishes such as *dal*, *papad*, *mutton*, *kabab*, and *chicken tikka* were towards the lower end of the performance spectrum with lesser than 70% F1 score which could be attributed to a relatively low number of annotations corresponding to these dishes. A detailed performance comparison of the best classification model (ResNet152) for each of the 61 dish classes of the IndianFood61 dataset is provided in Table E.1 of Appendix E.

Table 11.1: Comparison of multi-label classification models on IndianFood61 dataset.

Classification Model	**mAP (%)**	**F1 (%)**	**P (%)**	**R (%)**
AlexNet	47.20	59.55	81.49	46.91
SqueezeNet1_0	56.04	67.39	80.97	57.71
SqueezeNet1_1	57.19	68.01	81.47	58.36
DenseNet 121	72.93	79.22	83.99	74.96
VGG16	75.69	81.68	86.04	77.74
DenseNet 201	77.16	82.38	86.34	78.76
VGG19	77.98	83.36	88.01	79.17
ResNet50	82.30	86.72	89.56	84.05
DenseNet169	83.75	87.21	89.78	84.78
ResNet152	**84.51**	**88.01**	**90.56**	**85.59**

11.3.2 Object detection on IndianFood61 dataset

Beyond the classification ability of the models for the correct identification of dishes in the picture of a platter, marking the pixels of each dish is of practical importance for diet logging. Table 11.2 presents the comparison of eight object detection models on the IndianFood61 dataset. YOLOv8x outperformed all other models with state-of-the-art performance on mAP (87.70%), F1 (83.94%), and precision (83.90%) metric. The model detects most dishes with high accuracy (greater than 95%) except for *mutton*, *kabab*, *chicken tikka*, and *aloo gobi*, which had mAP scores lesser than 60%. Figure 11.2 displays 60 dish classes of the IndianFood61 dataset with bounding boxes predicted using the YOLOv8x model.

Table 11.2: Comparison of object detection models on IndianFood61 dataset.

Object Detection Model	**mAP (%)**	**F1 (%)**	**P (%)**	**R (%)**
DETR	52.02	60.97	52.00	73.70
RetinaNet	71.80	72.30	71.80	72.80
Faster R-CNN	75.45	70.14	75.50	65.49
YOLOv5	78.80	74.00	60.60	95.00
YOLOv7	83.50	78.92	76.60	81.40
YOLOv8	83.40	78.94	78.70	79.20
YOLOv8x	**87.70**	**83.94**	**83.90**	**84.00**

Figure 11.3 shows the confusion matrix of 61 dish classes using the YOLOv8x model. As evident from the matrix, the model detects most dishes with high accuracy (greater than 95%) except for *mutton*, *kabab*, *chicken tikka*, and *aloo gobi*, which had mAP scores lesser than 60%. This could be attributed to the poor resolution of the dish segment, diversity in the visual appearance of the dish, or the meta label of dish class that represents a variety of preparations.

Precision, Recall, and F1 curves visualize the model's response with varying confidence. The optimum model performance is assessed for the confidence threshold of 0.5. In addition to the Precision, Recall, and F1 curves, Figure 11.4 also shows the PR curve for the IndianFood61 dataset using the YOLOv8x model.

Figure 11.5 depicts each epoch's mAP score and box loss (training and validation). The saturation of mAP score indicates the convergence of the model performance, suggesting that further training is unnecessary. Box Loss is the loss function implemented during training, with its value representing the difference between the predicted and the actual bounding boxes. The decreasing train loss indicates that the model fits well with the training data and is, therefore, improving its ability to accurately predict bounding boxes. The low Box Loss for testing suggests that the model is generalizing well to new examples without traces of overfitting with training data. Table E.1 in Appendix E presents the detailed performance comparison of the best dish detection (YOLOV8x) model for each of the 61 dish classes of the IndianFood61 dataset.

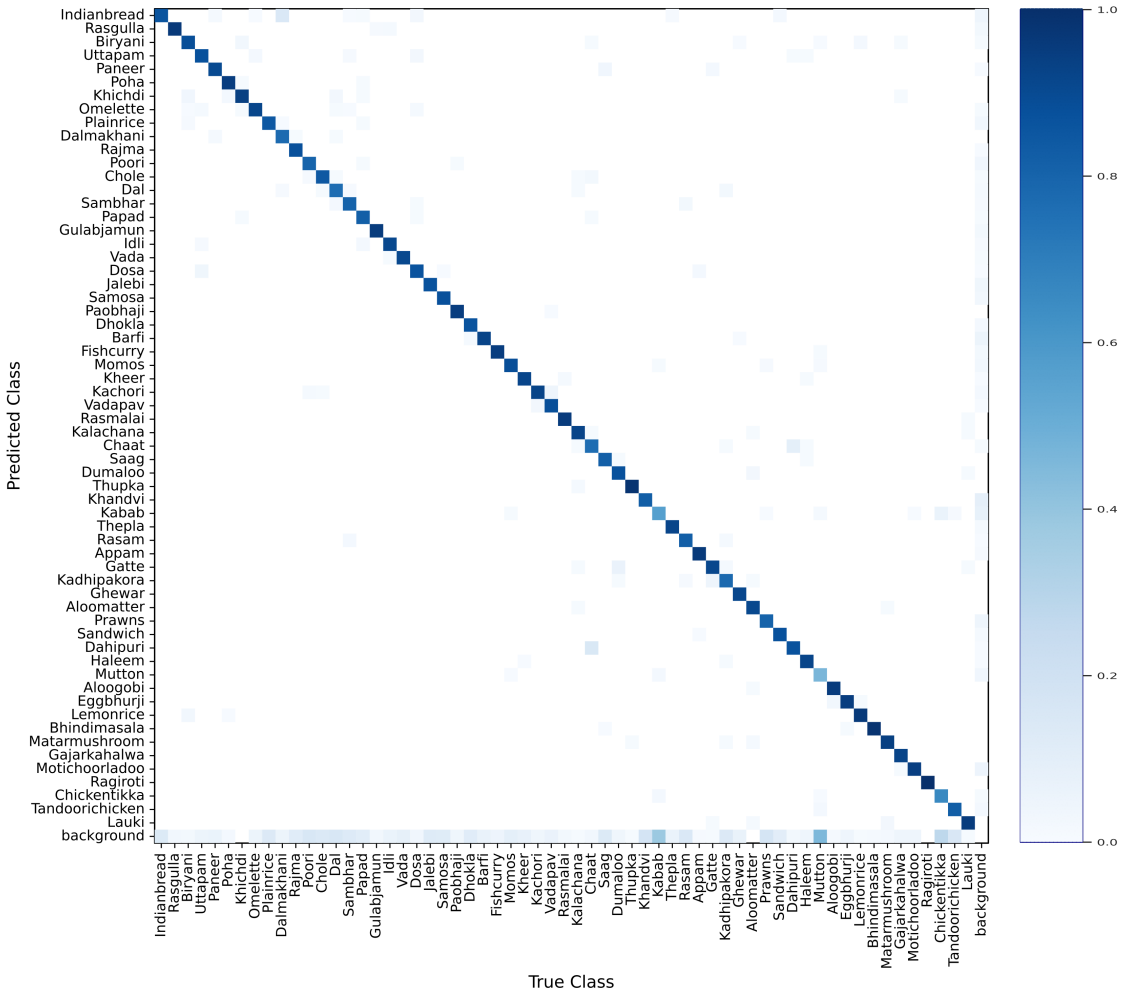

Figure 11.3: The confusion matrix for the IndianFood61 dataset using the YOLOv8x model shows the extent of concurrence between predicted and true classes.

11.3.3 Comparison on IndianFood10 dataset

We evaluated the performance of our models for image classification and object detection on IndianFood10, a dataset of ten Indian dishes that are a subset of the IndianFood61 (see Table 11.3, and Table 11.4). The state-of-the-art YOLOv8x model returns an impressive performance with a mAP score of 95.55% in contrast to the 91.80% returned by YOLOv4; the best model reported earlier [311].

11.4 Discussion

Effortless diet logging is crucial to mitigating lifestyle disorders through nutrition management and calorie restriction. Object detection applied to identify dishes in food platters can be of value in this objective. As a precursor, we created the IndianFood10 dataset for ten popular traditional Indian dishes and implemented the YOLOv4 model with mAP score of 91.8%.

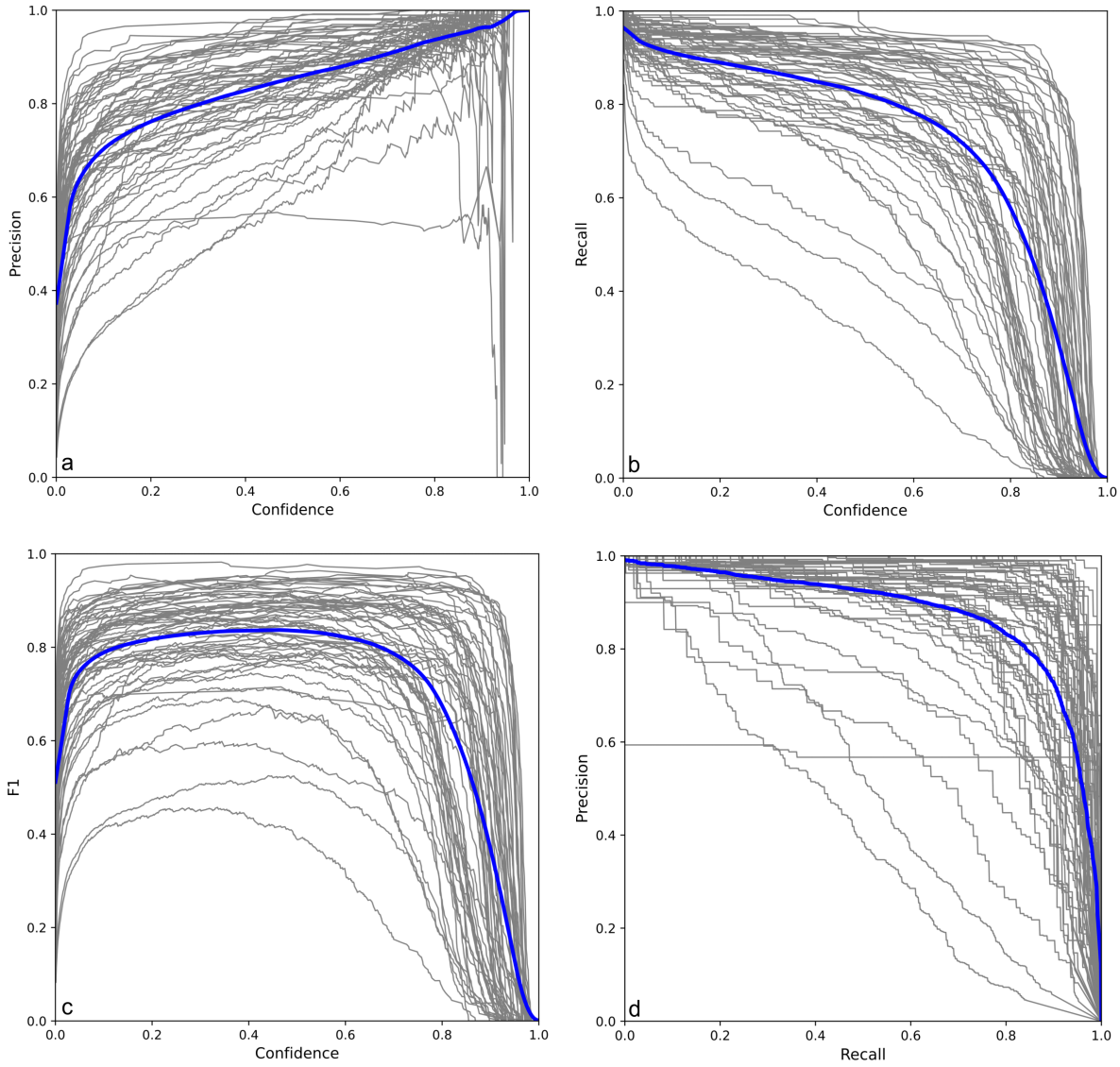

Figure 11.4: Performance of YOLOv8x model on IndianFood61 dataset. (a) Precision curve, (b) Recall curve, (c) F1 curve, and (d) Precision Recall curve. Gray lines represent the data for each of the 61 dishes; the blue line presents the average statistics.

Table 11.3: Comparison of multi-label classification models on IndianFood10 dataset.

Classification Model	**mAP (%)**	**F1 (%)**	**P (%)**	**R (%)**
SqueezeNet1_0	84.22	95.39	95.92	94.86
SqueezeNet1_1	83.85	95.79	96.02	95.56
DenseNet 121	95.02	95.71	95.90	95.52
VGG16	93.68	95.43	95.96	94.90
DenseNet 201	95.19	81.62	82.68	80.58
VGG19	94.27	86.04	87.03	85.07
ResNet50	95.12	94.68	95.03	94.33
DenseNet169	95.13	94.22	94.39	94.05
ResNet152	**95.14**	**95.14**	**95.56**	**94.33**

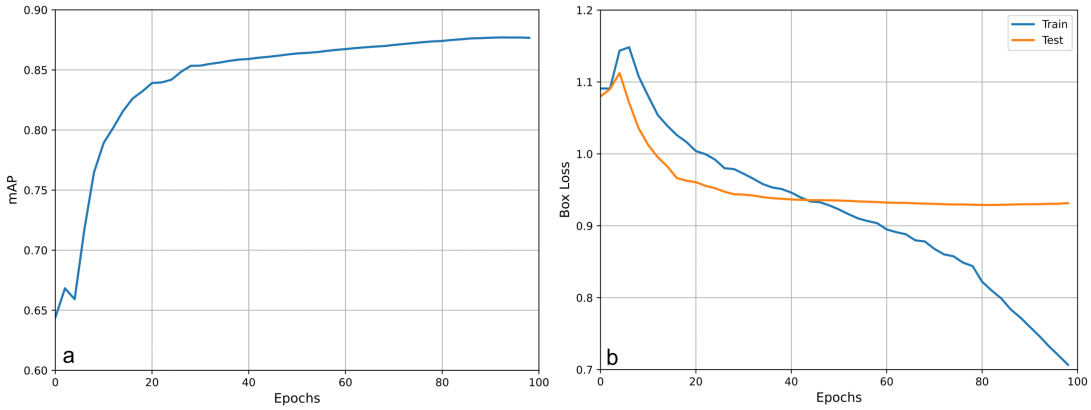

Figure 11.5: YOLOv8x model performance over 100 epochs for IndianFood61 dataset. (a) mAP scores, and (b) box loss.

Table 11.4: Comparison of object detection models on IndianFood10 dataset.

Object Detection Model	**mAP (%)**	**F1 (%)**	**P (%)**	**R (%)**
RetinaNet	83.91	76.74	83.90	70.70
DETR	86.00	83.70	86.00	81.50
Faster R-CNN	86.32	78.50	86.30	71.99
YOLOv4 [311]	**91.80**	**90.00**	**88.91**	**91.11**
YOLOv5	93.52	89.00	88.90	89.10
YOLOv7	94.41	90.00	88.80	91.20
YOLOv8	94.60	91.40	92.20	90.70
YOLOv8x	**95.55**	**92.00**	**94.00**	**90.00**

Building on this, we scaled up the labeled dataset of dish classes and created the IndianFood61, an extensive compilation of 68,005 images of food platters with 134,814 manual dish annotations for 61 popular Indian traditional dishes. We implemented deep-learning models for multi-label classification and object detection to achieve state-of-the-art performance. ResNet152 and YOLOv8x were identified as the best architecture for image classification and dish detection, respectively. Beyond nutrition management, food detection systems can also be leveraged to locate foreign objects in meals and to empower visually impaired individuals.

The proposed framework needs to be extended to include more staple dishes from worldwide cuisines to reduce false negatives. The dataset currently lacks traditional beverages and packaged food items among the commonly consumed food products. Future iterations should replace bounding boxes with segmentation masks to improve detection accuracy and granularity.

Chapter 12

Conclusions and Future Directions

This thesis advance the field of computational gastronomy and culinary data analysis by developing comprehensive, structured resources and innovative AI-driven approaches for recipe generation, classification, and addressing sustainability concerns.

RecipeDB2 stands out as a significant contribution, providing a rich repository of culinary data that integrates ingredient details, preparation methods, cultural origins, and nutritional information. Unlike other existing recipe datasets, RecipeDB2 systematically organizes culinary data, facilitating novel applications such as ingredient-based searches, personalized recipe generation, and dietary recommendations. Furthermore, the ontology-based approach in **RecipeOnt** establishes a structured framework for linking ingredient properties with molecular, nutritional, and cultural insights, bridging the gap between food science and AI-driven applications. The development of **named entity recognition** system for extracting named entities from ingredient phrases using deep learning models has achieved state-of-the-art performance. This system plays a crucial role in improving recipe generation, classification, and nutritional analysis. However, processing complex ingredient phrases remains challenging, particularly when handling multi-word expressions, regional ingredient variations, and ambiguous terms.

One of the central themes of this thesis has been the exploration of AI models for **recipe generation**. Comparing traditional LSTM networks with transformer-based architectures such as GPT-2 and LLaMA demonstrates that transformers can generate syntactically and semantically coherent recipes with high accuracy. The development of the Ratatouille system—complemented by its cuisine-specific extension, RatatouilleGen—highlights how the integration of culinary constraints can yield recipes that closely mirror the authentic characteristics of diverse regional cuisines. Evaluation via a **Turing Test for Chefs** provided robust insights into the practical viability of these models, with chefs often unable to distinguish between AI-generated and real recipes. These findings underscore the transformative potential of AI in enhancing culinary creativity and supporting personalized nutrition.

In addressing broader challenges in the culinary domain, this research incorporates **sustainability** by linking nutritional information with estimated carbon footprints. Current models use global estimates, future enhancements should incorporate more granular, geospatial data to accurately access the environmental costs of food products. Furthermore, the application of machine learning in food analysis **classifies the recipes** based on ingredients, utensils, and cooking processes revealing that ingredient composition plays a crucial role in categorizing dishes. Additionally, the development of **FlavorDB2** enriches flavor molecule research, providing

valuable insights into taste prediction and food pairings. The proposed machine learning models for **predicting taste (sweet and umami) and toxicity of molecules** further expand the applications of AI in computational gastronomy, offering potential benefits for food and pharmaceutical industries. The research also extends into **dish detection** with the creation of IndianFood61, a large dataset of traditional Indian dishes. The implementation of object detection models such as YOLOv8 for dish recognition and nutrition estimation lays the groundwork for future advancements in automated food logging and dietary assessment.

12.1 Future Directions

While this thesis makes significant strides in computational gastronomy, several limitations and avenues for future research remain. One key area is the enhancement of named entity recognition systems for culinary applications. Currently, our models focus on extracting ingredient phrases, but extending their capabilities to full recipe instructions will allow for a richer understanding of cooking techniques and cultural nuances. Fine-tuning large language models for NER tasks in the culinary domain can improve recognition accuracy, reduce biases, and enable better handling of multi-word ingredient expressions and regional ingredient variations. Additionally, multilingual NER would enhance the applicability of these models to diverse global cuisines, ensuring that ingredient recognition is not limited to English-based datasets.

Another important avenue for future work lies in AI-driven recipe generation. Although transformer-based models like GPT-2 and LLaMA have demonstrated strong performance, there is potential to enhance their output through refined prompt engineering techniques. By integrating methods such as soft prompt tuning and reinforcement learning with expert culinary feedback, future models could better capture subtle culinary details, ingredient interactions, and authentic cooking methods. Further research should focus on constraint-based recipe generation, where factors like dietary restrictions, calorie content, and specific cuisine preferences are incorporated. Expanding cuisine classification models to include factors such as cooking time and temperature would contribute to a more holistic understanding of culinary practices.

On the sustainability front, current models use global estimates to assess the environmental costs of food production. Future enhancements should incorporate more granular, geospatially resolved data along with additional parameters like water usage, land degradation, and resource depletion to more comprehensively capture environmental impacts. Finally, continued development of AI applications in culinary science, such as further advancements in FlavorDB2 and machine learning models for taste and toxicity prediction, holds great promise for both the food and pharmaceutical industries. These efforts will help drive innovations in personalized nutrition, sustainable food production, and culinary creativity, ultimately fostering interdisciplinary collaborations that blend the art of cooking with data science.

Appendix A

RecipeDB2: A Unified Framework for Recipe Data Structure

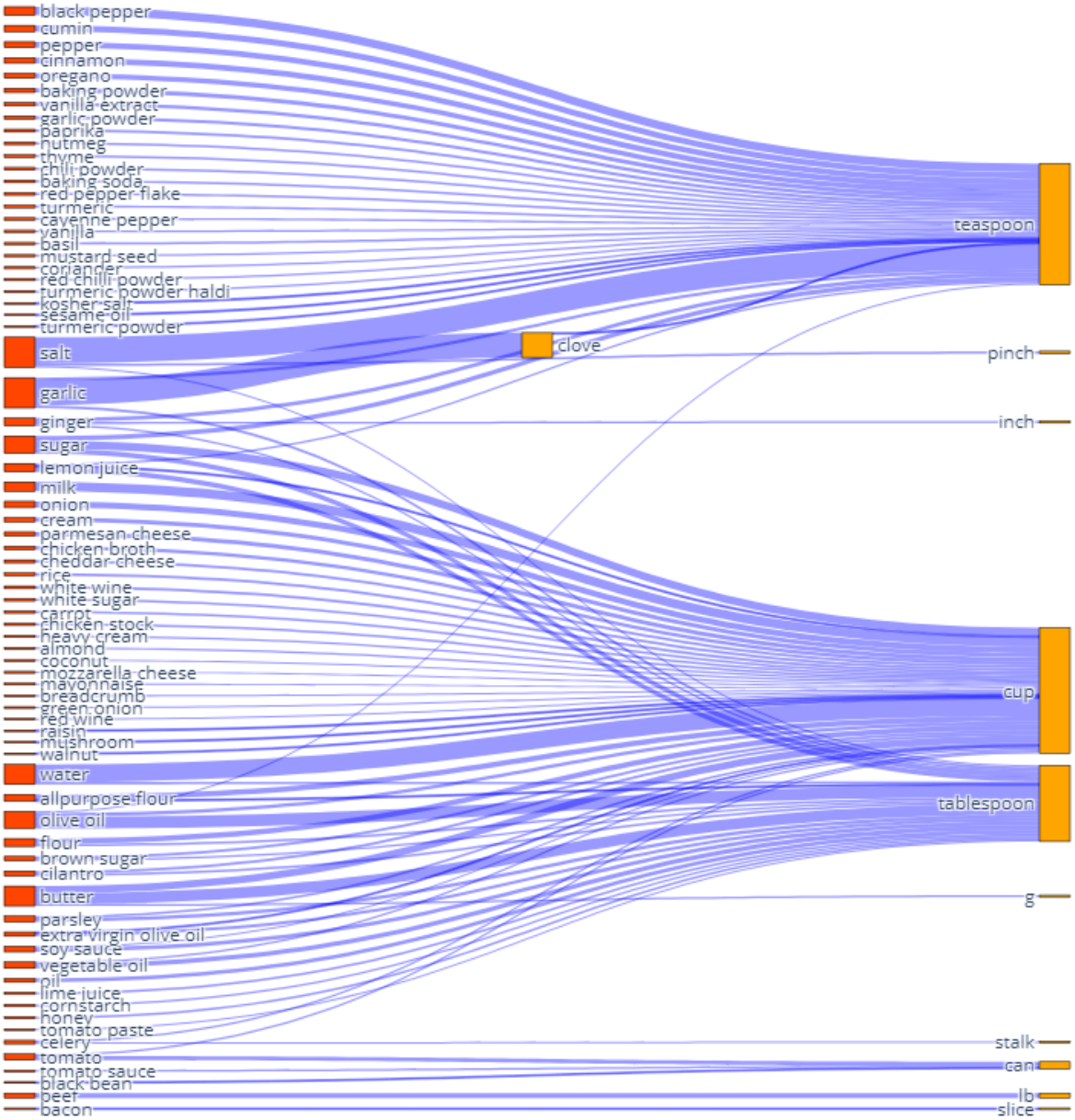

Figure A.1: Visualization of the 100 most popular ingredient-unit pairs derived from the dataset. This figure illustrates the frequency of each ingredient along with its corresponding measurement unit, providing insights into common culinary practices. For example, ingredients such as 'sugar' are frequently measured in cups, while 'spices' are often measured in teaspoons or grams.

Supplementary Table A.1 shows the mapping of RecipeDB2 continent and region, along with the recipe count.

Continent	Region	Recipe Count
African	Middle Eastern	325
African	Northern Africa	1610
African	Rest Africa	2774
Asian	Chinese	5954
Asian	Indian Subcontinent	13780
Asian	Japanese	2063
Asian	Korean	673
Asian	Middle Eastern	3674
Asian	Mongolian	75
Asian	Southeast Asian	2105
Asian	Thai	2724
Australasian	Australian	5819
European	Belgian	1060
European	Continental	1784
European	Deutschland	4321
European	Eastern European	2504
European	French	6485
European	Greek	4208
European	Irish	2529
European	Italian	16983
European	Middle Eastern	1
European	Portuguese	687
European	Scandinavian	2811
European	Spanish	2163
European	Spanish and Portuguese	70
European	UK	4430
European	Western European	42
Latin American	Caribbean	3030
Latin American	Central American	455
Latin American	Mexican	14656
Latin American	South American	7171
North American	Canadian	6694
North American	US	5126
Unknown	Unknown	72

Table A.1: Mapping of cuisines from their (32) Regions to (7) Continents and (region-wise) number of recipes.

Appendix B

Cultural Context Shapes the Carbon Footprint of Recipes

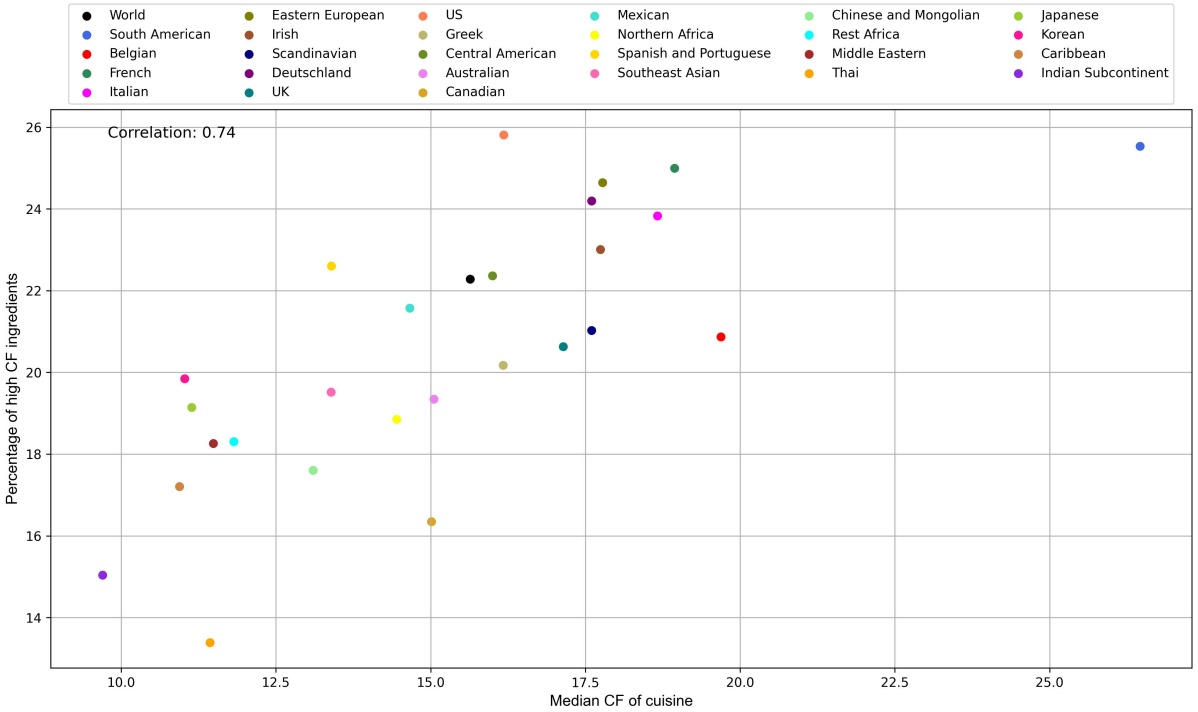

Figure B.1: Correlation of Top 1000 RecipeDB ingredients with High CF (CF>4) ingredients. The median CF of cuisines shows a correlation of 0.74 for the Top 1000 most frequently used ingredients.

Table B.1: List of regions and countries in the SuEatable dataset.

Region	**Country**
Africa	Africa (Near East And North)
Africa	Africa (Sub-Saharan)
Africa	Ghana
Africa	Madagascar
Africa	Mauritius
Africa	Morocco
Africa	North Africa
Africa	Senegal
Africa	Somalia
Africa	South Africa

Africa	Tanzania
Africa	Uganda
Asia	Asia (East And South-East)
Asia	Asia (South)
Asia	Bangladesh
Asia	China
Asia	E Asia
Asia	India
Asia	Indian Ocean
Asia	Indonesia
Asia	Iran
Asia	Japan
Asia	Kazakhstan
Asia	Korea
Asia	Malaysia
Asia	Maldives
Asia	Pakistan
Asia	Philippines
Asia	Thailand
Asia	Vietnam
Atlantic Ocean	Atlantic Ocean
C America	Costa Rica
C America	El Salvador
C America	Guatemala
C America	Mexico
C America	Nicaragua
C Europe	C Europe
CS America	America (Central And South)
E Europe	Bulgaria
E Europe	Czech Republic
E Europe	E Europe
E Europe	Estonia
E Europe	Hungary
E Europe	Latvia
E Europe	Lithuania
E Europe	Romania
E Europe	Russia

E Europe	Serbia
E Europe	Slovakia
E Europe	Slovenia
Europe	EU
Europe	Europe
Mediterranean Area	Croatia
Mediterranean Area	Cyprus
Mediterranean Area	France (Corsica)
Mediterranean Area	Greece
Mediterranean Area	Israel
Mediterranean Area	Italy
Mediterranean Area	Malta
Mediterranean Area	Mediterranean Area
Mediterranean Area	Portugal
Mediterranean Area	Spain
Mediterranean Area	Spain (Valencia)
Mediterranean Area	Tunisia
Mediterranean Area	Turkey
N America	Alaska
N America	Canada
N America	Canada (East)
N America	Canada (West)
N America	N America
N America	USA
N America	USA Mid-West
N Europe	Austria
N Europe	Belarus
N Europe	Belgium
N Europe	Denmark
N Europe	Finland
N Europe	France
N Europe	France (North)
N Europe	France (South)
N Europe	Germany
N Europe	Iceland
N Europe	Ireland
N Europe	Luxembourg

N Europe	Moldavia
N Europe	N Europe
N Europe	Netherlands
N Europe	Norway
N Europe	Poland
N Europe	Sweden
N Europe	Switzerland
N Europe	UK
N Europe	UK (Scotland)
N Europe	Ukraine
Non EU Countries	Non EU Countries
Oceania	Australia
Oceania	New Zealand
Oceania	Oceania
Pacific Ocean	Pacific Ocean
S Africa	Reunion Island
S America	Argentina
S America	Brazil
S America	Chile
S America	Colombia
S America	Ecuador
S America	Peru
S America	S America
S America	S America (Latin America And Caribbean)
S America	Uruguay
S America	Venezuela
W Europe	W Europe
World	World Estimate

Table B.2: 50 most popular RecipeDB ingredients and their SuEatable mappings (Accuracy of BERT model for Top 50 ingredients = 78%).

RecipeDB Ingredient	**SuEatable Food Product**	**Similarity**	**CF**	**Frequency**
Onion	Onion	1	0.24	69096
Butter	Butter	1	9.9	54026
Garlic Clove	Garlic	0.93	0.67	49786
Water	Water	1	0.49	49546

Olive Oil	Olive Oil	1	3.84	44782
Egg	Egg	1	3.23	43722
Sugar	Beet Sugar	0.85	0.89	40542
Tomato	Tomato	1	0.48	33250
Garlic	Garlic	1	0.67	30872
Milk	Cream	0.81	5.34	29170
Pepper	Pepper	1	0.58	24608
Salt Pepper	Pepper	0.82	0.58	20774
Flour	Millet Flour	0.88	1.37	19674
Cinnamon	Vanilla	0.81	4.3	19248
Lemon Juice	Lemon	0.87	0.22	19018
Carrot	Carrot	1	0.23	18174
Purpose Flour	Millet Flour	0.84	1.37	18046
Vegetable Oil	Vegetable	0.89	0.69	18036
Cumin	Quorn	0.91	2.5	17984
Cream	Cream	1	5.34	17160
Ginger	Ginger	1	0.88	16816
Parmesan Cheese	Cheese Semihard	0.89	8.65	15578
Soy Sauce	Soy Cream	0.92	1.62	15454
Beef	Beef With Bone	0.87	19.54	15348
Potato	Potato	1	0.25	14528
Green Onion	Green Bean	0.86	0.73	14148
Chicken Broth	Chicken With Bone	0.87	3.25	11826
Lemon	Lemon	1	0.22	10986
Lime Juice	Lime	0.81	0.22	10956
Chicken Breast	Chicken With Bone	0.90	3.25	10792
Mushroom	Mushroom	1	1.78	10284
Garlic Powder	Garlic	0.89	0.67	9532
Celery	Celery	1	0.32	9216
Cheddar Cheese	Cheese Semihard	0.89	8.65	8996
Cornstarch	Corn Can	0.91	1.36	8776
Nutmeg	Cashew Nut	0.84	1.56	8486
White Wine	Wine White	0.93	0.74	7944
Vanilla Extract	Vanilla	0.91	4.3	7912
Honey	Honey	1	1.74	7906
Red Bell Pepper	Red Chilli	0.84	0.8	7820
Tomato Paste	Tomato Peel	0.96	1.28	7636

Coriander	Radish	0.82	0.15	7412
Chicken	Chicken With Bone	0.82	3.25	7178
Tomato Sauce	Tomato Peel	0.92	1.28	6962
Mozzarella Cheese	Cheese Semihard	0.85	8.65	6936
Almond	Almond	1	1.9	6616
Bacon	Bacon	1	4.03	6516
Green Bell Pepper	Green Bean	0.85	0.73	6344
Red Pepper	Red Chilli	0.87	0.8	6330
Turmeric	Radish	0.82	0.15	6216
Cream Cheese	Hazelnut Cream	0.84	2.71	6188

Table B.3: 50 least frequently used RecipeDB ingredients and their SuEatable mappings.

RecipeDB Ingredient	**SuEatable Food Product**	**Similarity**	**CF**	**Frequency**
Rainbow Chocolate Chip	Hazelnut Chocolate	0.91	3.43	2
Chocolate Candy Melt	Hazelnut Chocolate	0.90	3.43	2
Spinach Ravioli	Spinach	0.83	0.48	2
Cherry Chocolate Bar	Almond Chocolate	0.89	4.8	2
Serrano Pepper Heat	Pepper	0.84	0.58	2
Brioche Breadcrumb	Bread Whole	0.81	0.78	2
Bread Improver Knead	Bread Multicereal	0.81	0.7	2
Abiu	Hake	0.83	10.12	2
Chicken Mince	Chicken With Bone	0.90	3.25	2
Raspberry Cordial	Raspberry	0.94	0.63	2
Fruit Sultana Apricot	Apricot	0.89	0.36	2
Cheese Feta	Cheese Semihard	0.90	8.65	2
Clix Biscuit	Cucumber	0.82	0.32	2
Mango Peach Tea	Peach	0.85	0.43	2
Pepper Beef Sausage	Pork Sausage	0.86	17.94	2
Cream Preferably	Cream	0.90	5.34	2
Chicken Carcass Meat Remaining	Chicken With Bone	0.81	3.25	2
Plain Biscuit Cracker Crumb	Plain Cracker	0.85	1.24	2
No Oil French Dressing	Pesto Without Garlic	0.81	2.72	2

Violet Crumble Chocolate Candy	Almond Chocolate	0.85	4.8	2
Bush Tomato	Tomato Peel	0.86	1.28	2
Cadbury Chocolate Candy Bar	Hazelnut Chocolate	0.89	3.43	2
Hazelnut Milk Chocolate	Hazelnut Chocolate	0.94	3.43	2
Lsa	Tangerin	0.83	0.38	2
Coffee Creamer Blueberry	Blueberry Juice	0.86	3	2
Pumpkin Pepitas	Pumpkin	0.92	0.38	2
Bran Cereal Protein Powder	Cornflakes Cereal	0.81	2.64	2
Nut Corn Flake	Cornflakes Cereal	0.88	2.64	2
Grumichama	Hake	0.84	10.12	2
Pear Juice Fruit	Pear Juice	0.96	0.49	2
Gluten Flour Mix	Sorghum Flour	0.88	1.33	2
Milk Chocolate Dark	Milk Chocolate	0.86	3.6	2
Double Espresso	Espresso	0.87	0.55	2
Cinnamon Apple Tea	Cranberry Juice	0.83	2.88	2
Cheese Tart	Cheese Semihard	0.90	8.65	2
Arnott Milk Coffee Biscuit	Coffee Powder	0.82	0.33	2
Ciabatta Choice Bread	Bread Whole	0.81	0.78	2
Oval Pita Pocket Bread	Bread Multicereal	0.81	0.7	2
Cocoa More	Cocoa Cake	0.82	1.97	2
Almond One	Almond	0.95	1.9	2
Pistachio Paste	Pistachio	0.93	1.6	2
Ice Cream Topping Peppermint Crisp	Cranberry Juice	0.82	2.88	2
Chicken Schnitzel	Chicken With Bone	0.85	3.25	2
Berry Fruit	Pear Juice	0.81	0.49	2
Frisee Lettuce	Lettuce	0.94	0.4	2
Dutch Cocoa Hershey	Hazelnut Chocolate	0.82	3.43	2
Milk Chocolate Hershey	Milk Chocolate	0.94	3.6	2

Chocolate Flavor Crisp Rice Cereal	Chocolate Cream Cooky	0.81	1.69	2
Chocolate Ripple Ice Cream	Chocolate Cream Cooky	0.92	1.69	2
Caramel Topping Syrup	Cranberry Juice	0.85	2.88	2

Table B.4: 60 most popular RecipeDB ingredients that were unmapped to SuEatable.

Salt	**Cayenne Pepper**	**Dijon Mustard**	**Flat Leaf Parsley**
Black Pepper	Red Pepper Flake	Black Olive	Celery Rib
Parsley	Salt Black Pepper	Mustard	Garam Masala
Cilantro	Clove	Mint	Pine Nut
Oregano	Worcestershire Sauce	Basil Leaf	Mint Leaf
Brown Sugar	Kosher Salt	Cumin Seed	Steak
Oil	Heavy Cream	Vinegar	Coriander
Basil	Baking Soda	Allspice	Sage
Extra Virgin Olive Oil	Salsa	White Pepper	Cayenne
Baking Powder	Canola Oil	Balsamic Vinegar	Turkey
Paprika	Sea Salt	White Vinegar	Ice
Bay Leaf	Jalapeno Pepper	White Onion	Rum
Chili Powder	Black Bean	Italian Seasoning	Seasoning
Thyme	Curry Powder	Jalapeno	Tortilla Chip
White Sugar	Rosemary	Ham	Chilli Sauce

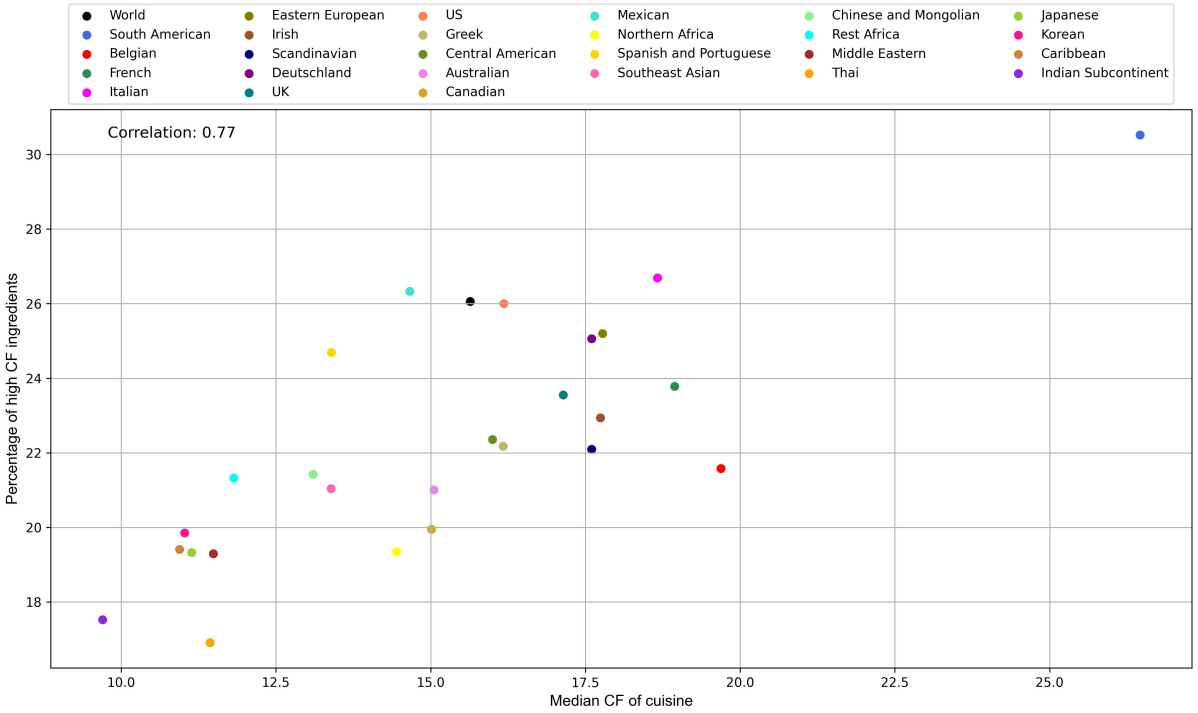

Figure B.2: Correlation of Top 10000 RecipeDB ingredients with High CF (CF>4) ingredients. The median CF of cuisines shows a correlation of 0.77 for the Top 10000 most frequently used ingredients.

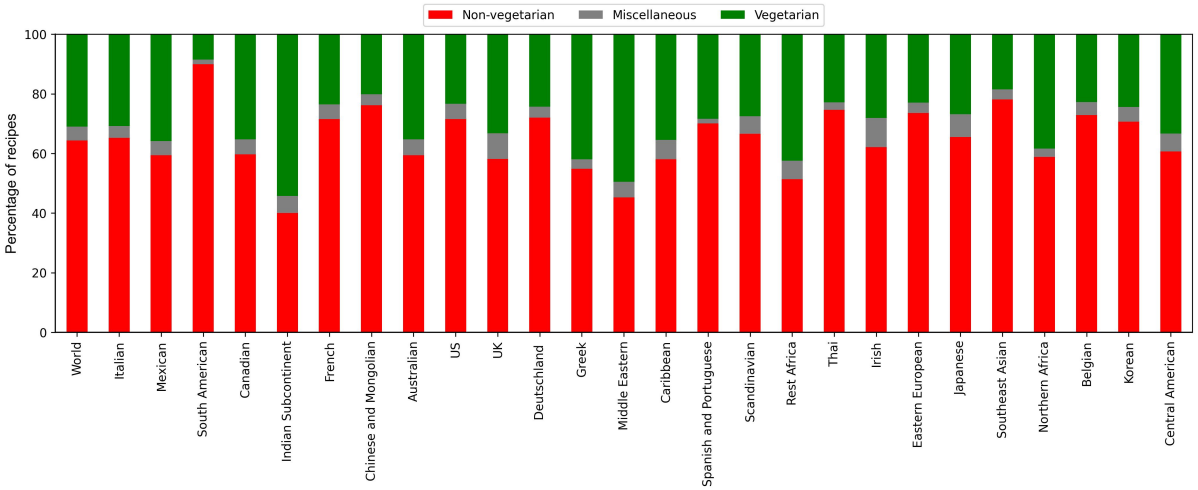

Figure B.3: Extent of non-vegetarian recipes. Statistics of vegetarian, non-vegetarian, and miscellaneous recipes across cuisines.

Table B.5: List of ingredient categories and associated ingredients.

Ingredient Category	**Ingredient Frequency**	**Illustrative List of Ingredients**
Meat	1212	Egg, Beef, Chicken, Bacon, Lamb, Pork, Turkey, etc.
Fruit	724	Lemon, Raisin, Orange, Coconut, Cherry, Tomato, Apple, Banana, Cranberry, Raspberry, Mango, etc.
Dairy	653	Butter, Milk, Cream, Cheese, Yogurt, Curd, etc.
Cereal	498	Flour, Rice, Vermicelli, Green bean, Sesame seed, Pasta, Wheat, Maize, etc.
Plant-Derivative	491	Olive oil, Honey, Red wine, Vinegar, Chocolate, etc.
Dish	465	Vegetable broth, Sausage, Tortilla, Tomato soup, Ice-cream, Cake, etc.
Vegetable	464	Onion, Tomato, Carrot, Spinach, Chilly, Lettuce, Egg-plant, Cabbage, Plum, etc.
Bakery	450	Bread, Biscuit, Tortilla, etc.
Miscellaneous	439	Water, Sugar, Cinnamon, Vegetable oil, Vanilla extract, Margarine, Ketchup, etc.
Beverage	239	Lemon juice, Pineapple juice, Green tea, Coffee, Shakes, etc.
Fish	223	Salmon, Tuna, Anchovy, Cod, Haddock, etc.
Spice	198	Pepper, Salt pepper, Cumin, Ginger, Nutmeg, Turmeric, etc.
Legume	190	Vanilla, Pea, Chickpea, Bean, lentil
Nuts and Seeds	161	Almond, Walnut, Peanut, Cashew, Pistachio, Hazelnut
Beverage-Alcoholic	158	White wine, Red wine, Mirin, Apple cider, Rum, Beer, etc.
Condiment	157	Soy sauce, Tomato sauce, Mayonnaise, etc.
Maize	146	Corn and its variations
Seafood	108	Shallot, Shrimp, Mussel, Prawn, Clam, Scallop, Squid
Herb	105	Garlic, Coriander, Hummus, Fennel, Thyme, Celery, etc.
Fungi	57	Mushroom and its variations
Vegetable-Flower	52	Broccoli, Artichoke, Cauliflower
Essential Oil	49	Sesame oil, Walnut oil
Vegetable-Fruit	39	Bell pepper and its variations
Beverage-Caffeinated	39	Cocoa powder, and Coffee variations
Vegetable-Tuber	31	Potato and its variations
Gourd	18	Cucumber and its variations
Berry	7	Wheat berry, Rye berry, Berry syrup, Berry fruit

Appendix C

Mining Culinary Patterns to Differentiate Global Cuisines

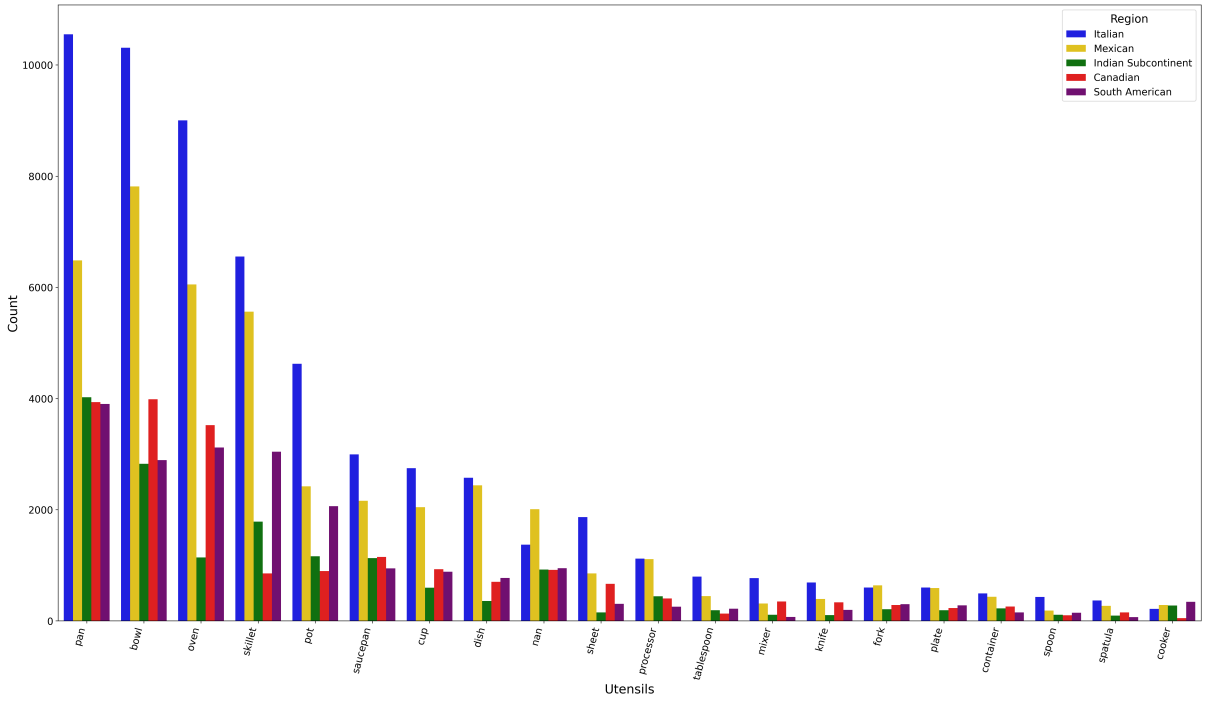

Figure C.1: Distribution of popularly used utensils based on the regions.

Model	Method	Accuracy	Recall	F1 Score	Precision
LightGBM	Count-Vectorizer	84.13	84.13	84.00	84.44
ANN	Count-Vectorizer	84.00	84.00	83.96	83.98
LightGBM	TF-IDF	83.98	83.92	83.90	84.00
ANN	TF-IDF	83.96	83.90	83.87	83.90
Stack Model (XGB, LGBM, LR)	Count-Vectorizer	83.64	83.61	83.59	83.62
XGBoost	Count-Vectorizer	83.48	83.48	83.45	83.46
Stack Model (XGB, LGBM, LR)	TF-IDF	83.42	83.40	83.40	83.43
XGBoost	TF-IDF	83.34	83.35	83.32	83.33
RF	Count-Vectorizer	81.98	81.98	81.95	81.96
LR	Count-Vectorizer	81.87	81.90	81.79	81.85
LR	TF-IDF	81.70	81.70	81.56	81.63

Table C.1: Performance of classification model with the ingredients and processes as feature vectors.

Models	Method	Accuracy	Recall	F1 Score	Precision
ANN	Count-Vectorizer	47.56	47.54	47.50	47.52
ANN	TF-IDF	46.95	46.94	46.92	46.93
LightGBM	Count-Vectorizer	45.73	45.73	45.69	45.70
XGBoost	Count-Vectorizer	45.69	45.69	45.67	45.67
LightGBM	TF-IDF	45.50	45.47	45.45	45.48
Stack Model (XGB, LGBM, LR)	Count-Vectorizer	45.37	45.37	45.34	45.36
XGBoost	TF-IDF	45.31	45.33	43.29	43.00
Stack Model (XGB, LGBM, LR)	TF-IDF	45.24	45.25	45.21	45.22
RF	Count-Vectorizer	44.34	44.34	44.30	44.31
LR	Count-Vectorizer	43.87	43.85	43.82	45.86
LR	TF-IDF	43.61	43.64	43.61	43.63

Table C.2: Performance of classification model with the utensils and processes as feature vectors.

Appendix D

Molecular Taste and Toxicity Prediction

D.1 UmamiPred

Table D.1 presents the comparison of previous models on the same dataset for umami and non-umami classification.

Model	Year	Dataset	Acc	Recall	Sp	AUC-ROC	F1-Score	MCC
TabPFN [263]	2024	Umami vs. Non-Umami (Combined Patent + UMP442 dataset of 439 umami and 428 non-umami)	0.93	0.93	0.93	0.93	0.93	NA
UmamiPreDL [262]	2024	Umami vs. Non-Umami (UMP-IND dataset of 444 compounds (140 umami, 304 non-umami))	0.94	0.98	0.89	0.954	0.929	0.891
iUmami-SCM [253]	2020	Umami vs. Non-Umami (UMP-IND dataset of 444 compounds (140 umami, 304 non-umami))	0.865	0.714	0.934	0.898	NA	0.679
UMPred-FRL [259]	2021	Umami vs. Non-Umami (UMP-IND dataset of 444 compounds (140 umami, 304 non-umami))	0.888	0.786	0.934	0.919	NA	0.735
VirtuousUmami [260]	2022	Umami vs. Non-Umami (UMP-IND dataset of 444 compounds (140 umami, 304 non-umami))	0.876	0.786	0.918	0.85	NA	NA
IUP-BERT [261]	2022	Umami vs. Non-Umami (UMP-IND dataset of 444 compounds (140 umami, 304 non-umami))	0.896	0.893	0.902	0.933	NA	0.774
Umami-MRNN [266]	2023	Umami vs. Non-Umami (UMP-IND dataset of 444 compounds (140 umami, 304 non-umami))	0.905	NA	NA	NA	NA	0.811
Umami_YYDS [352]	2024	Umami vs. Non-Umami (UMP-IND dataset of 444 compounds (140 umami, 304 non-umami))	0.896	NA	NA	NA	NA	NA

Table D.1: Performance comparison of existing models for umami vs. non-umami compound classification.

Table D.2 presents the hyperparameter tuning settings for peptide, small molecule, and combined datasets using Mol2vec and Morgan fingerprints feature encodings for umami and non-umami classification.

Model	Peptide	Small Molecule	Combined
RF	n_estimators: 115, max_depth: 27, min_samples_split: 8, min_samples_leaf: 1, max_features: sqrt	n_estimators: 344, max_depth: 11, min_samples_split: 5, min_samples_leaf: 1, max_features: sqrt	n_estimators: 541, max_depth: 41, min_samples_split: 12, min_samples_leaf: 1, max_features: sqrt
LightGBM	num_leaves: 67, max_depth: 31, learning_rate: 0.040, n_estimators: 179, min_child_samples: 5	num_leaves: 57, max_depth: 33, learning_rate: 0.040, n_estimators: 280, min_child_samples: 44	num_leaves: 83, max_depth: 14, learning_rate: 0.035, n_estimators: 117, min_child_samples: 8
XGBoost	max_depth: 30, learning_rate: 0.079, n_estimators: 618, subsample: 0.511, colsample_bytree: 0.980	max_depth: 29, learning_rate: 0.051, n_estimators: 71, subsample: 0.699, colsample_bytree: 0.722	max_depth: 8, learning_rate: 0.008, n_estimators: 958, subsample: 0.920, colsample_bytree: 0.868
LR	penalty: l1, C: 0.289, solver: liblinear	penalty: l1, C: 3.94, solver: liblinear	penalty: l1, C: 2.677, solver: saga
LDA	solver: lsqr, shrinkage: 0.438	solver: lsqr, shrinkage: 0.7902	solver: eigen, shrinkage: 0.495
QDA	reg_param: 0.644	reg_param: 0.000667	reg_param: 0.0034
kNN	n_neighbors: 13, weights: distance, p: 5	n_neighbors: 9, weights: uniform, p: 4	n_neighbors: 6, weights: distance, p: 4
ET	n_estimators: 54, max_depth: 25, min_samples_split: 5, min_samples_leaf: 3, max_features: log2	n_estimators: 70, max_depth: 50, min_samples_split: 6, min_samples_leaf: 1, max_features: sqrt	n_estimators: 932, max_depth: 43, min_samples_split: 3, min_samples_leaf: 1, max_features: sqrt
NB	-	-	-
SVM	C: 9.569, kernel: rbf, gamma: scale	C: 1.088, kernel: poly, gamma: scale	C: 0.090, kernel: linear, gamma: scale

Table D.2: Comparison of hyperparameter settings across peptide dataset (Mol2vec), small molecule dataset (Morgan fingerprints), and combined dataset.

D.2 ToxinPred

As evident from Table D.3, the feature selection pipeline of ToxinPred significantly reduced the dimensionality of all three datasets.

Stage	Features Remaining	ToxIM	MolTox	Combined
Initial	Features	20398	20398	20398
Dropping duplicate columns	Features	11978	13606	13721
Variance threshold	Features	11975	13605	13719
Correlation-based Filtering	Features	8772	10117	10140
Relevance Filtering	Features	8248	7407	9260
Boruta Algorithm	Features	486	636	1185
PCA	Principal Components	382	531	944

Table D.3: Feature selection process of ToxinPred across datasets.

Table D.4 shows the performance of classification models on the MolToxPred dataset. SVM model performs best with AUC of 89.7%, F1 score of 82.7% and accuracy of 82.9%.

Model	Training						Testing					
	AUC	Acc	P	R	MCC	F1	**AUC**	Acc	P	R	MCC	F1
SVM	**96.0**	88.9	90.3	87.1	77.8	88.7	**89.7**	82.9	83.9	81.5	65.9	82.7
DNN	**96.6**	90.8	92.9	88.3	81.7	90.5	**88.8**	81.28	82.2	79.1	62.1	80.7
XGB	**99.8**	97.3	97.3	97.3	94.6	97.3	**88.2**	80.6	82.6	77.4	61.2	79.9
GBM	**95.5**	88.6	89.9	87.0	77.2	88.4	**86.3**	78.6	79.9	76.3	57.2	78.1
KNN	**92.6**	83.1	79.4	89.3	66.6	84.1	**86.0**	76.7	73.0	84.7	54.1	78.4
LR	**89.1**	81.1	81.8	80.0	62.1	80.9	**85.1**	77.4	77.6	77.1	54.8	77.3
RF	**99.9**	97.9	98.3	97.6	95.9	97.9	**84.9**	78.8	84.1	70.9	58.3	77.0
DT	**95.8**	89.9	90.0	89.7	79.8	89.9	**73.9**	71.7	70.9	73.6	43.5	72.2
NB	**69.7**	64.9	64.3	67.2	29.8	65.7	**70.7**	65.8	65.0	68.3	31.6	66.6

Table D.4: Performance of classification models on MolToxPred's dataset.

Table D.5 shows the performance of classification models on the ToxiM dataset. The SVM model performs best with an AUC of 89.7%, an F1 score of 82.7%, and an accuracy of 82.9%.

Model	Training						Testing					
	AUC	Acc	P	R	MCC	F1	**AUC**	Acc	P	R	MCC	F1
SVM	**99.3**	96.4	97.7	95.0	92.8	96.3	**98.7**	95.4	96.6	94.0	90.8	95.3
XGB	**99.9**	99.0	99.9	98.1	98.1	99.0	**98.8**	96.5	97.4	95.6	93.0	96.5
GBM	**99.9**	98.6	99.7	97.6	97.3	98.6	**98.7**	96.2	97.0	95.3	92.3	96.1
DNN	**99.7**	98.2	99.6	96.8	96.4	98.2	**98.4**	96.3	98.8	93.7	92.7	96.2
RF	**99.9**	99.1	99.6	98.5	98.1	99.1	**98.3**	96.6	96.6	96.6	93.2	96.6
LR	**98.7**	95.1	96.5	93.6	90.2	95.0	**98.2**	94.7	95.6	93.6	89.3	94.6
KNN	**99.4**	93.8	99.5	88.1	88.3	93.5	**97.1**	92.5	98.2	86.6	85.6	92.0
DT	**98.3**	97.3	99.6	95.1	94.8	97.3	**93.0**	92.6	94.8	90.2	85.3	92.4
NB	**67.5**	63.6	69.3	49.0	28.5	57.4	**64.9**	62.5	66.6	50.1	25.8	57.2

Table D.5: Performance of classification models on ToxiM's dataset.

Table D.6 shows the hyperparameters of models.

Model	**Hyperparameters**
LR	Random State: 42, Max Iterations: 200, Regularization Strength (C): 0.01
DT	Random State: 42, Max Depth: 10
RF	Random State: 42, Number of Estimators: 100
GB	Random State: 42, Learning Rate: 0.2
XGB	Random State: 42, Label Encoder: False, Evaluation Metric: Log Loss, Learning Rate: 0.1
SVM	Probability: True, Regularization Parameter (C): 1
KNN	Number of Neighbors: 7
NB	uses default settings

Table D.6: Hyperparameters of classification models for ToxinPred.

Appendix E

Dish Detection in Indian Food Platters

Table E.1: Performance of each dish class of IndianFood61 dataset for the best multi-label classification (Resnet152) and object detection (YOLOV8x) models. For each of the 61 dish classes, the number of images and annotations are presented, other than the details of model performance for classification (precision, recall, and F1 score) and object detection (precision, recall, and mAP score).

Dish	**Images**	**Annotations**	**Classification**			**Object Detection**		
			P (%)	**R (%)**	**F1 (%)**	**P (%)**	**R (%)**	**mAP (%)**
Aloo Gobi	1070	1102	95.19	96.12	95.65	56.50	96.70	57.30
Aloo Matter	1039	1071	95.96	89.62	92.68	98.00	90.20	98.40
Appam	998	1559	98.97	96.00	97.46	85.40	95.10	97.30
Barfi	1056	6375	92.55	84.47	88.32	81.50	87.40	90.00
Bhindi Masala	1088	1155	96.15	92.59	94.34	93.30	97.30	98.40
Biryani	1427	1632	91.11	87.86	89.45	84.90	90.30	90.70
Chaat	1044	1159	82.80	74.04	78.17	76.90	80.70	86.40
Chicken Tikka	946	1162	73.75	62.77	67.82	63.80	53.70	56.70
Chole	1096	1173	90.18	84.87	87.45	76.40	78.10	80.90
Dahi Puri	1050	1102	84.26	86.67	85.45	80.20	88.50	89.30
Dal	853	909	70.97	45.83	55.70	65.00	70.30	66.10
Dal Makhani	1522	1683	93.08	84.03	88.32	92.20	75.50	85.30
Dhokla	1026	5129	98.99	96.08	97.51	85.00	80.90	83.40
Dosa	818	1171	86.84	84.62	85.71	81.10	85.60	88.10
Dum Aloo	1006	1022	88.64	78.00	82.98	95.50	84.60	94.50
Egg Bhurji	1037	1119	94.79	88.35	91.46	95.10	92.80	96.80
Fish Curry	1176	3390	94.35	87.43	90.76	91.50	90.30	94.70
Gajar Ka Halwa	961	1070	96.47	84.54	90.11	91.50	91.90	96.30
Gatte	1007	1050	87.23	82.00	84.54	87.80	91.10	96.00
Ghewar	1055	1330	98.99	93.33	96.08	93.10	89.10	95.50
Gulabjamun	533	4284	98.28	85.07	91.20	92.30	92.50	97.50
Haleem	1027	1178	96.91	92.16	94.47	86.50	90.40	90.50
Idli	494	1999	90.70	70.91	79.59	89.30	88.50	92.10
Indian Bread	3074	4109	80.34	88.40	84.18	74.00	77.60	80.90
Jalebi	1290	4058	95.90	90.70	93.23	81.00	75.80	85.70
Kabab	1000	5348	70.33	64.00	67.02	59.10	47.10	52.70
Kachori	1062	3377	92.71	84.76	88.56	86.30	90.20	94.40
Kadhi Pakora	1002	1096	79.21	80.81	80.00	85.90	74.70	84.60
Kala Chana	1031	1083	95.60	86.14	90.62	95.00	90.40	95.60
Khandvi	1015	5084	99.00	98.02	98.51	67.10	67.20	72.40
Kheer	1034	1347	94.74	87.38	90.91	85.90	90.90	94.70

Khichdi	1198	1240	91.07	90.27	90.67	82.60	92.20	93.60
Lauki	994	1007	96.47	84.54	90.11	89.30	93.90	96.30
Lemon Rice	989	1014	91.00	91.00	91.00	92.80	94.10	97.50
Matar Mushroom	1004	1027	93.81	91.00	92.39	93.60	94.10	98.40
Momos	1083	6496	95.45	77.78	85.71	85.60	83.50	90.50
Motichoor Ladoo	1027	6212	94.00	92.16	93.07	82.50	89.30	91.20
Mutton	1161	3130	78.05	55.17	64.65	55.60	33.50	41.00
Omelette	1278	1427	89.68	88.98	89.33	84.50	88.30	92.30
Paneer	1478	1531	96.18	84.00	89.68	84.70	86.80	93.20
Pav Bhaji	1084	1137	100	94.50	97.17	96.50	93.00	96.20
Papad	761	1039	79.17	52.05	62.81	75.00	79.00	77.90
Plain Rice	4067	4435	84.86	91.81	88.20	82.60	79.10	85.00
Poha	1219	1337	93.28	94.07	93.67	97.20	94.60	98.10
Poori	1071	1970	85.95	79.39	82.54	73.00	69.20	77.10
Prawns	1047	4204	88.17	78.85	83.25	76.70	71.00	80.40
Ragi Roti	1041	1513	99.05	100	99.52	81.80	100	96.40
Rajma	1139	1271	96.12	94.29	95.19	89.20	86.70	89.80
Rasam	798	846	85.92	77.22	81.33	80.50	80.60	88.40
Rasgulla	591	4623	92.31	82.76	87.27	85.50	93.50	95.00
Rasmalai	1005	1168	98.98	97.00	97.98	93.40	94.50	98.30
Saag	1228	1364	87.39	79.51	83.26	89.10	75.80	85.80
Sambhar	1045	1122	81.55	67.74	74.01	76.90	77.10	82.60
Samosa	1127	3579	89.81	85.09	87.39	77.40	83.00	86.30
Sandwich	1016	2070	96.08	96.08	96.08	79.50	79.50	84.90
Tandoori Chicken	1206	2671	93.02	100	96.39	80.60	74.50	86.60
Thepla	1006	2053	97.87	92.00	94.85	86.60	84.20	90.70
Thupka	1047	1139	94.23	94.23	94.23	97.30	96.50	99.40
Uttapam	805	1669	89.55	82.19	85.71	89.40	84.00	91.10
Vada	714	2722	95.31	83.56	89.05	93.60	88.00	93.10
Vada Pav	939	1472	88.42	90.32	89.36	88.20	90.70	88.30

References

[1] J. A. Brillat-Savarin, *The Physiology of Taste: or Meditations on Transcendental Gastronomy*, 2009.

[2] M. Pollan, *Cooked: A natural history of transformation*. Penguin, 2014.

[3] D. Zeevi, T. Korem, N. Zmora, D. Israeli, D. Rothschild, A. Weinberger, O. Ben-Yacov, D. Lador, T. Avnit-Sagi, M. Lotan-Pompan, J. Suez, J. A. Mahdi, E. Matot, G. Malka, N. Kosower, M. Rein, G. Zilberman-Schapira, L. Dohnalová, M. Pevsner-Fischer, R. Bikovsky, Z. Halpern, E. Elinav, and E. Segal, "Personalized Nutrition by Prediction of Glycemic Responses," *Cell*, vol. 163, no. 5, pp. 1079–1094, 2015.

[4] E. D. Sonnenburg and J. L. Sonnenburg, "Nutrition: A personal forecast," *Nature*, vol. 528, no. 7583, pp. 484–486, dec 2015.

[5] M. Goel and G. Bagler, "Computational gastronomy: A data science approach to food," *Journal of Biosciences*, vol. 47, no. 1, p. 12, 2022.

[6] D. Batra, N. Diwan, U. Upadhyay, J. S. Kalra, T. Sharma, A. K. Sharma, D. Khanna, J. S. Marwah, S. Kalathil, N. Singh, R. Tuwani, and G. Bagler, "RecipeDB: A resource for exploring recipes," *Database*, vol. 2020, 2020.

[7] N. Garg, A. Sethupathy, R. Tuwani, R. Nk, S. Dokania, A. Iyer, A. Gupta, S. Agrawal, N. Singh, S. Shukla *et al.*, "Flavordb: a database of flavor molecules," *Nucleic Acids Research*, vol. 46, no. D1, pp. D1210–D1216, 2018.

[8] N. Diwan, D. Batra, and G. Bagler, "A named entity based approach to model recipes," *Proceedings - 2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW)*, pp. 88–93, 2020.

[9] H. Arn and T. E. Acree, "Flavornet: A database of aroma compounds based on odor potency in natural products," *Developments in Food Science*, vol. 40, no. C, p. 27, 1998.

[10] M. Dunkel, U. Schmidt, S. Struck, L. Berger, B. Gruening, J. Hossbach, I. S. Jaeger, U. Effmert, B. Piechulla, R. Eriksson, J. Knudsen, and R. Preissner, "SuperScent - A database of flavors and scents," *Nucleic Acids Research*, vol. 37, no. SUPPL. 1, pp. D291–D294, 2009.

[11] J. Ahmed, S. Preissner, M. Dunkel, C. L. Worth, A. Eckert, and R. Preissner, "SuperSweet-A resource on natural and artificial sweetening agents," *Nucleic Acids Research*, vol. 39, no. SUPPL. 1, 2011.

[12] A. Wiener, M. Shudler, A. Levit, and M. Y. Niv, "BitterDB: A database of bitter compounds," *Nucleic Acids Research*, vol. 40, no. D1, pp. D413–D419, 2012.

[13] V. Neveu, J. Perez-Jiménez, F. Vos, V. Crespy, L. du Chaffaut, L. Mennen, C. Knox, R. Eisner, J. Cruz, D. Wishart, and A. Scalbert, "Phenol-Explorer: an online comprehensive database on polyphenol contents in foods." *Database : the journal of biological databases and curation*, vol. 2010, 2010.

[14] A. Scalbert, C. Andres-Lacueva, M. Arita, P. Kroon, C. Manach, M. Urpi-Sarda, and D. Wishart, "Databases on food phytochemicals and their health-promoting effects," pp. 4331–4348, may 2011.

[15] J. A. Rothwell, J. Perez-Jimenez, V. Neveu, A. Medina-Remón, N. M'Hiri, P. García-Lobato, C. Manach, C. Knox, R. Eisner, D. S. Wishart, and A. Scalbert, "Phenol-Explorer 3.0: A major update of the Phenol-Explorer database to incorporate data on the effects of food processing on polyphenol content," *Database*, vol. 2013, 2013.

[16] K. Jensen, G. Panagiotou, and I. Kouskoumvekaki, "NutriChem: A systems chemical biology resource to explore the medicinal value of plant-based foods," *Nucleic Acids Research*, vol. 43, no. D1, pp. D940–D945, 2015.

[17] N. Garg, A. Sethupathy, R. Tuwani, R. Nk, S. Dokania, A. Iyer, A. Gupta, S. Agrawal, N. Singh, S. Shukla, K. Kathuria, R. Badhwar, R. Kanji, A. Jain, A. Kaur, R. Nagpal, and G. Bagler, "FlavorDB: A database of flavor molecules," *Nucleic Acids Research*, vol. 46, no. D1, pp. D1210–D1216, 2018.

[18] H. Blumenthal, *The Big Fat Duck Cookbook*. Bloomsbury, 2008.

[19] Y. Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A. L. Barabási, "Flavor network and the principles of food pairing," *Scientific Reports*, vol. 1, pp. 1–7, 2011.

[20] S. E. Ahnert, "Network analysis and data mining in food science: the emergence of computational gastronomy," *Flavour*, vol. 2, no. 1, pp. 2–4, 2013.

[21] A. Jain, R. N. K, and G. Bagler, "Spices form the basis of food pairing in Indian cuisine," *arxiv:1502.03815*, no. 7, pp. 1–30, 2015.

[22] A. Jain, N. K. Rakhi, and G. Bagler, "Analysis of food pairing in regional cuisines of India," *PLoS ONE*, vol. 10, no. 10, pp. 1–17, 2015.

[23] S. T. Tallab and M. S. Alrazgan, "Exploring the Food Pairing Hypothesis in Arab Cuisine: A Study in Computational Gastronomy," in *Procedia Computer Science*, vol. 82, no. 3. Elsevier Masson SAS, 2016, pp. 135–137.

[24] K. R. Varshney, L. R. Varshney, J. Wang, and D. Myers, "Flavor Pairing in Medieval European Cuisine: A Study in Cooking with Dirty Data," *arXiv*, 2013.

[25] T. Simas, M. Ficek, A. Diaz-Guilera, P. Obrador, and P. R. Rodriguez, "Food-bridging: A new network construction to Unveil the principles of cooking," *Frontiers in ICT*, vol. 4, no. 6, 2017.

[26] N. Singh and G. Bagler, "Data-driven investigations of culinary patterns in traditional recipes across the world," in *Proceedings - IEEE 34th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 2018, pp. 157–162.

[27] C. Spence, "Food and beverage flavour pairing: A critical review of the literature," p. 109124, 2020.

[28] D. Park, K. Kim, Y. Park, J. Shin, and J. Kang, "Kitchenette: Predicting and ranking food ingredient pairings using siamese neural networks," in *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2019-Augus, 2019, pp. 5930–5936.

[29] M. Kazama, M. Sugimoto, C. Hosokawa, K. Matsushima, L. R. Varshney, and Y. Ishikawa, "A neural network system for transformation of regional cuisine style," *Frontiers in ICT*, vol. 5, no. 7, pp. 1–8, 2018.

[30] N. K. Rakhi, R. Tuwani, J. Mukherjee, and G. Bagler, "Data-driven analysis of biomedical literature suggests broad-spectrum benefits of culinary herbs and spices," *PLoS ONE*, vol. 13, no. 5, p. 276105, 2018.

[31] N. K. Rakhi, R. Tuwani, N. Garg, and G. Bagler, "SpiceRx: an integrated resource for the health impacts of culinary spices and herbs," *bioRxiv:1101.273599*, pp. 1–24, feb 2018.

[32] F. M. Afendi, T. Okada, M. Yamazaki, A. Hirai-Morita, Y. Nakamura, K. Nakamura, S. Ikeda, H. Takahashi, M. Altaf-Ul-Amin, L. K. Darusman, K. Saito, and S. Kanaya, "KNApSAcK family databases: Integrated metabolite-plant species databases for multifaceted plant research," *Plant and Cell Physiology*, vol. 53, no. 2, 2012.

[33] A. P. Davis, C. J. Grondin, R. J. Johnson, D. Sciaky, B. L. King, R. McMorran, J. Wiegers, T. C. Wiegers, and C. J. Mattingly, "The Comparative Toxicogenomics Database," *Nucleic Acids Research*, vol. 45, no. D1, pp. D972–D978, jan 2017.

[34] S. Abbar, Y. Mejova, and I. Weber, "You tweet what you eat: Studying food consumption through twitter," in *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2015-April. Association for Computing Machinery, 2015, pp. 3197–3206.

[35] S. Sajadmanesh, S. Jafarzadeh, S. A. Ossia, H. R. Rabiee, H. Haddadi, Y. Mejova, M. Musolesi, E. De Cristofaro, and G. Stringhini, "Kissing cuisines: Exploring worldwide culinary habits on the web," in *26th International World Wide Web Conference 2017, WWW 2017 Companion*, 2017, pp. 1013–1021.

[36] M. van Erp, C. Reynolds, D. Maynard, A. Starke, R. Ibáñez Martín, F. Andres, M. C. Leite, D. Alvarez de Toledo, X. Schmidt Rivera, C. Trattner, S. Brewer, C. Adriano Martins, A. Kluczkovski, A. Frankowska, S. Bridle, R. B. Levy, F. Rauber, J. Tereza da Silva, and U. Bosma, "Using Natural Language Processing and Artificial Intelligence to Explore the Nutrition and Sustainability of Recipes and Food," *Frontiers in Artificial Intelligence*, vol. 3, 2021.

[37] S. S. Shirai, O. Seneviratne, M. E. Gordon, C. H. Chen, and D. L. McGuinness, "Identifying Ingredient Substitutions Using a Knowledge Graph of Food," *Frontiers in Artificial Intelligence*, vol. 3, jan 2021.

[38] "Predicting human olfactory perception from chemical features of odor molecules," *Science*, vol. 355, no. 6327, pp. 820–826, 2017.

[39] A. Dagan-Wiener, I. Nissim, N. Ben Abu, G. Borgonovo, A. Bassoli, and M. Y. Niv, "Bitter or not? BitterPredict, a tool for predicting taste from chemical structure," *Scientific Reports*, vol. 7, no. 1, pp. 1–13, 2017.

[40] C. Rojas, P. Tripaldi, and P. R. Duchowicz, "A New QSPR Study on Relative Sweetness," *International Journal of Quantitative Structure-Property Relationships*, vol. 1, no. 1, pp. 78–93, 2016.

[41] S. Zheng, M. Jiang, C. Zhao, R. Zhu, Z. Hu, Y. Xu, and F. Lin, "e-Bitter: Bitterant prediction by the consensus voting from the machine-learning methods," *Frontiers in Chemistry*, vol. 6, no. MAR, mar 2018.

[42] P. Banerjee and R. Preissner, "Bitter sweet forest: A Random Forest based binary classifier to predict bitterness and sweetness of chemical compounds," *Frontiers in Chemistry*, vol. 6, no. 4, 2018.

[43] R. Tuwani, S. Wadhwa, and G. Bagler, "BitterSweet: Building machine learning models for predicting the bitter and sweet taste of small molecules," *Scientific Reports*, vol. 9, no. 1, pp. 1–13, 2019.

[44] T. Sharma, U. Upadhyay, J. Kalra, S. Arora, S. Ahmad, B. Aggarwal, and G. Bagler, "Hierarchical clustering of world cuisines," in *Proceedings - 2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW)*, 2020, pp. 98–104.

[45] R. Dale, "Generating recipes: an overview of Epicure," *Current Research in Natural Language Generation*, pp. 229–255, 1990.

[46] C. Kiddon, L. Zettlemoyer, and Y. Choi, "Globally coherent text generation with neural checklist models," in *Proceedings - Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 329–339.

[47] H. Lee, K. Shu, P. Achananuparp, P. K. Prasetyo, Y. Liu, E. P. Lim, and L. R. Varshney, "RecipeGPT: Generative Pre-training Based Cooking Recipe Generation and Evaluation System," in *The Web Conference 2020 - Companion of the World Wide Web Conference (WWW)*. Association for Computing Machinery, apr 2020, pp. 181–184.

[48] A. Salvador, M. Drozdzal, X. Giro-I-Nieto, and A. Romero, "Inverse cooking: Recipe generation from food images," in *Proceedings - IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 445–10 454.

[49] H. Wang, G. Lin, S. C. H. Hoi, and C. Miao, "Decomposed Generation Networks with Structure Prediction for Recipe Generation from Food Images," *arXiv*, pp. 1–10, 2020.

[50] B. P. Majumder, S. Li, J. Ni, and J. McAuley, "Generating personalized recipes from historical user preferences," in *Proceedings - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, nov 2019, pp. 5976–5982.

[51] Z. Yu, H. Zang, and X. Wan, "Routing enforced generative model for recipe generation," in *Proceedings - 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 3797–3806.

[52] A. Morales-Garzón, J. Gomez-Romero, and M. J. Martin-Bautista, "A word embedding-based method for unsupervised adaptation of cooking recipes," *IEEE Access*, vol. 9, pp. 27 389–27 404, 2021.

[53] R. Venkataramanan, K. Roy, K. Raj, R. Prasad, Y. Zi, V. Narayanan, and A. Sheth, "Cook-gen: Robust generative modeling of cooking actions from recipes," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2023, pp. 981–986.

[54] P. Chhikara, D. Chaurasia, Y. Jiang, O. Masur, and F. Ilievski, "Fire: Food image to recipe generation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 8184–8194.

[55] B. Fatemi, Q. Duval, R. Girdhar, M. Drozdzal, and A. Romero-Soriano, "Learning to substitute ingredients in recipes," *arXiv preprint arXiv:2302.07960*, 2023.

[56] D. Noever and S. E. M. Noever, "The multimodal and modular ai chef: Complex recipe generation from imagery," *arXiv preprint arXiv:2304.02016*, 2023.

[57] G. Bagler, "A generative grammar of cooking," *arXiv preprint arXiv:2211.09059*, 2022.

[58] G. Crosby, *Cook, taste, learn: How the evolution of science transformed the art of cooking*. Columbia University Press, 2020.

[59] A. Gatley, "The significance of culinary cultures to diet," *British Food Journal*, vol. 118, no. 1, pp. 40–59, 2016.

[60] R. Wrangham, *Catching Fire: How Cooking Made Us Human*. Basic Books, 2009.

[61] R. N. Carmody, J. E. Bisanz, B. P. Bowen, C. F. Maurice, S. Lyalina, K. B. Louie, D. Treen, K. S. Chadaideh, V. Maini Rekdal, E. N. Bess *et al.*, "Cooking shapes the structure and function of the gut microbiome," *Nature Microbiology*, vol. 4, no. 12, pp. 2052–2063, 2019.

[62] G. Bagler and M. Goel, "Computational gastronomy: capturing culinary creativity by making food computable," *NPJ Systems Biology and Applications*, vol. 10, no. 1, p. 72, 2024.

[63] T. Wharton, "Recipes: Beyond the words," *Gastronomica*, vol. 10, no. 4, pp. 67–73, 2010.

[64] J. Marin, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba, "Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 187–203, 2021.

[65] M. Bień, M. Gilski, M. Maciejewska, W. Taisner, D. Wisniewski, and A. Lawrynowicz, "Recipenlg: A cooking recipes dataset for semi-structured text generation," in *Proceedings of the 13th International Conference on Natural Language Generation*, 2020, pp. 22–28.

[66] G. Popovski, B. K. Seljak, and T. Eftimov, "Foodbase corpus: a new resource of annotated food entities," *Database*, p. baz121, 2019.

[67] G. Cenikj, B. K. Seljak, and T. Eftimov, "Foodchem: A food-chemical relation extraction model," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2021, pp. 1–8.

[68] G. Cenikj, T. Eftimov, and B. K. Seljak, "Foodis: A food-disease relation mining pipeline," *Artificial intelligence in medicine*, vol. 142, p. 102586, 2023.

[69] G. Cenikj, L. Strojnik, R. Angelski, N. Ogrinc, B. Koroušić Seljak, and T. Eftimov, "From language models to large-scale food and biomedical knowledge graphs," *Scientific reports*, vol. 13, no. 1, p. 7815, 2023.

[70] S. Haussmann, O. Seneviratne, Y. Chen, Y. Ne'eman, J. Codella, C.-H. Chen, D. L. McGuinness, and M. J. Zaki, "Foodkg: a semantics-driven knowledge graph for food recommendation," in *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18*. Springer, 2019, pp. 146–162.

[71] M. Goel, N. Grover, D. Batra, N. Garg, R. Tuwani, A. Sethupathy, and G. Bagler, "Flavordb2: an updated database of flavor molecules," *Journal of Food Science*, vol. 89, no. 11, pp. 7076–7082, 2024.

[72] A. Wiener, M. Shudler, A. Levit, and M. Y. Niv, "Bitterdb: a database of bitter compounds," *Nucleic Acids Research*, vol. 40, no. D1, pp. D413–D419, 2012.

[73] J. Ahmed, S. Preissner, M. Dunkel, C. L. Worth, A. Eckert, and R. Preissner, "Supersweet—a resource on natural and artificial sweetening agents," *Nucleic Acids Research*, vol. 39, pp. D377–D382, 2010.

[74] K. Jensen, G. Panagiotou, and I. Kouskoumvekaki, "Nutrichem: a systems chemical biology resource to explore the medicinal value of plant-based foods," *Nucleic Acids Research*, vol. 43, no. D1, pp. D940–D945, 2015.

[75] M. Goel, A. Agarwal, S. Agrawal, J. Kapuriya, A. V. Konam, R. Gupta, S. Rastogi, Niharika, and G. Bagler, "Deep Learning Based Named Entity Recognition Models for Recipes," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, 2024, pp. 1–13.

[76] S. Bag, S. K. Kumar, and M. K. Tiwari, "An efficient recommendation generation using relevant jaccard similarity," *Information Sciences*, vol. 483, pp. 53–64, 2019.

[77] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, vol. 1. Minneapolis, Minnesota, 2019, p. 2.

[78] N. Reimers, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint:1908.10084*, 2019.

[79] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv:1907.11692*, 2019.

[80] C. Trattner, D. Elsweiler, and S. Howard, "Estimating the healthiness of internet recipes: a cross-sectional study," *Frontiers in Public Health*, vol. 5, p. 16, 2017.

[81] M. Goel, P. Chakraborty, V. Ponnaganti, M. Khan, S. Tatipamala, A. Saini, and G. Bagler, "Ratatouille: A tool for novel recipe generation," in *38th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 2022, pp. 107–110.

[82] G. Ispirova, T. Eftimov, S. Džeroski, and B. K. Seljak, "Msgen: Measuring generalization of nutrient value prediction across different recipe datasets," *Expert Systems with Applications*, vol. 237, p. 121507, 2024.

[83] F. Batista, J. P. Pardal, P. V. N. Mamede, and R. Ribeiro, "Ontology construction: cooking domain," *Artificial Intelligence: Methodology, Systems, and Applications*, vol. 41, no. 1, p. 30, 2006.

[84] R. Ribeiro, F. Batista, J. P. Pardal, N. J. Mamede, and H. S. Pinto, "Cooking an ontology," in *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*. Springer, 2006, pp. 213–221.

[85] N. Mihindukulasooriya, S. Tiwari, C. F. Enguix, and K. Lata, "Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text," in *International Semantic Web Conference*. Springer, 2023, pp. 247–265.

[86] C. Caracciolo, A. Stellato, A. Morshed, G. Johannsen, S. Rajbhandari, Y. Jaques, and J. Keizer, "The agrovoc linked dataset," *Semantic Web*, vol. 4, no. 3, pp. 341–348, 2013.

[87] C. Snae and M. Bruckner, "Foods: a food-oriented ontology-driven system," in *2008 2nd ieee international conference on digital ecosystems and technologies*. IEEE, 2008, pp. 168–176.

[88] T. A. Holton, V. Vijayakumar, and N. Khaldi, "Bioinformatics: Current perspectives and future directions for food and nutritional research facilitated by a food-wiki database," *Trends in food science technology*, vol. 34, no. 1, pp. 5–17, 2013.

[89] C. Chelmis and B. Gergin, "Recipe networks and the principles of healthy food on the web," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 17, 2023, pp. 95–102.

[90] F. Badra, R. Bendaoud, R. Bentebibel, P.-A. Champin, J. Cojan, A. Cordier, S. Després, S. Jean-Daubias, J. Lieber, T. Meilender *et al.*, "Taaable: Text mining, ontology engineering, and hierarchical classification for textual case-based cooking," in *9th European Conference on Case-Based Reasoning-ECCBR 2008, Workshop Proceedings*, 2008, pp. 219–228.

[91] R. V. Guha, D. Brickley, and S. Macbeth, "Schema. org: evolution of structured data on the web," *Communications of the ACM*, vol. 59, no. 2, pp. 44–51, 2016.

[92] D. Allemang and J. Hendler, *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier, 2011.

[93] N. F. Noy, M. Crubézy, R. W. Fergerson, H. Knublauch, S. W. Tu, J. Vendetti, and M. A. Musen, "Protégé-2000: an open-source ontology-development and knowledge-acquisition environment: Amia 2003 open source expo," in *Amia annual symposium proceedings*, vol. 2003, 2003, p. 953.

[94] C. Shimizu and K. Hammar, "Comodide–the comprehensive modular ontology engineering ide," in *ISWC 2019 Satellite Tracks (Posters Demonstrations, Industry, and Outrageous Ideas) co-located with 18th International Semantic Web Conference (ISWC 2019) Auckland, New Zealand, October 26-30, 2019.*, vol. 2456. CEUR-WS, 2019, pp. 249–252.

[95] D. Core, "Dublin core metadata initiative," *http://www. dublincore. org*, 2004.

[96] J. M. Keil and S. Schindler, "Comparison and evaluation of ontologies for units of measurement," *Semantic Web*, vol. 10, no. 1, pp. 33–51, 2019.

[97] D. Vats and A. Sharma, "Transforming ontology from csv file and relational databases: A methodology," in *2021 IEEE International Conference on Technology, Research, and Innovation for Betterment of Society (TRIBES)*. IEEE, 2021, pp. 1–6.

[98] A. Duque-Ramos, J. T. Fernández-Breis, R. Stevens, and N. Aussenac-Gilles, "Oquare: A square-based approach for evaluating the quality of ontologies," *Journal of research and practice in information technology*, vol. 43, no. 2, pp. 159–176, 2011.

[99] K. Sawarkar, A. Mangal, and S. R. Solanki, "Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers," *arXiv preprint arXiv:2404.07220*, 2024.

[100] S. Acharyya, A. Ghosh, S. Nag, S. B. Majumder, and P. K. Guha, "Smart and selective gas sensor system empowered with machine learning over iot platform," *IEEE Internet of Things Journal*, 2023.

[101] M. N. Kamel Boulos, A. Yassine, S. Shirmohammadi, C. S. Namahoot, and M. Brückner, "Towards an "internet of food": food ontologies for the internet of things," *Future Internet*, vol. 7, no. 4, pp. 372–392, 2015.

[102] X. Wei, X. Cui, N. Cheng, X. Wang, X. Zhang, S. Huang, P. Xie, J. Xu, Y. Chen, M. Zhang, Y. Jiang, and W. Han, "Zero-Shot Information Extraction via Chatting with ChatGPT," *arXiv:2302.10205*, 2023.

[103] J. Kalra, D. Batra, N. Diwan, and G. Bagler, "Nutritional profile estimation in cooking recipes," in *Proceedings - 2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW)*, 2020, pp. 82–87.

[104] C. Pellegrini, E. Özsoy, M. Wintergerst, and G. Groh, "Exploiting food embeddings for ingredient substitution," in *Proceedings - 14th International Conference on Health Informatics (HEALTHINF)*, 2021, pp. 67–77.

[105] M. H. Syed and S. T. Chung, "Menuner: Domain-adapted bert based ner approach for a domain with limited dataset and its application to food menu domain," *Applied Sciences*, vol. 11, no. 13, p. 6007, 2021.

[106] H. L. Chieu and H. T. Ng, "Named entity recognition," *Stanford Lecture CS229*, pp. 1–7, 2002.

[107] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ErniE: Enhanced language representation with informative entities," *Proceedings - 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1441–1451, 2020.

[108] P. Cheng and K. Erk, "Attending to Entities for Better Text Understanding," in *Proceedings - 34th Conference on Artificial Intelligence (AAAI)*, vol. 34, no. 05, 2020, pp. 7554–7561.

[109] J. Guo, G. Xu, X. Cheng, and H. Li, "Named entity recognition in query," in *Proceedings - 32nd Annual International Conference on Research and Development in Information Retrieval (SIGIR)*, 2009, pp. 267–274.

[110] D. Petkova and W. Bruce Croft, "Proximity-based document representation for named entity retrieval," in *Proceedings - International Conference on Information and Knowledge Management*, 2007, pp. 731–740.

[111] C. Aone, "A trainable summarizer with knowledge acquired from robust nlp techniques," *Advances in automatic text summarization*, vol. MIT Press, pp. 71—-80, 1999.

[112] D. Mollá, M. van Zaanen, and D. Smith, "Named Entity Recognition for Question Answering," in *Proceedings - ALTW*, 2006, pp. 51–58.

[113] B. Babych and A. Hartley, "Improving machine translation quality with automatic named entity recognition," in *Proceedings - 7th International EAMT workshop (EACL)*, 2003, pp. 1–8.

[114] O. Etzioni, M. Cafarella, D. Downey, A. M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Unsupervised named-entity extraction from the Web: An experimental study," *Artificial Intelligence*, vol. 165, no. 1, pp. 91–134, 2005.

[115] W. Liu and X. Cui, "Improving Named Entity Recognition for Social Media with Data Augmentation," *Applied Sciences*, vol. 13, no. 9, p. 5360, apr 2023.

[116] Q. Fang, Y. Li, H. Feng, and Y. Ruan, "Chinese Named Entity Recognition Model Based on Multi-Task Learning," *Applied Sciences*, vol. 13, no. 8, p. 4770, apr 2023.

[117] M. Suleman, M. Asif, T. Zamir, A. Mehmood, J. Khan, N. Ahmad, and K. Ahmad, "Floods Relevancy and Identification of Location from Twitter Posts using NLP Techniques," 2023.

[118] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT," pp. 2–6, 2019.

[119] M. J. Hossain, M. I. H. Bhuiyan, and Z. R. Abdullah, "CpG Island Detection Using Transformer Model with Conditional Random Field," in *Proceedings - IEEE Bombay Section Signature Conference (IBSSC)*, 2022, pp. 1–5.

[120] S. Silalahi, T. Ahmad, and H. Studiawan, "Named Entity Recognition for Drone Forensic Using BERT and DistilBERT," in *Proceedings - 2022 International Conference on Data Science and Its Applications (ICoDSA)*, 2022, pp. 53–58.

[121] S. Davidson, J. Hosier, Y. Zhou, and V. K. Gurbani, "Improved Named Entity Recognition for Noisy Call Center Transcripts," in *Proceedings - 7th Workshop on Noisy User-Generated Text*, 2021, pp. 361–370.

[122] X. Qu, J. Zeng, D. Liu, Z. Wang, B. Huai, and P. Zhou, "Distantly-Supervised Named Entity Recognition with Adaptive Teacher Learning and Fine-Grained Student Ensemble," *Proceedings - 37th AAAI Conference on Artificial Intelligence*, vol. 37, pp. 13 501–13 509, 2023.

[123] F. B. Rodrigues, W. F. Giozza, R. de Oliveira Albuquerque, and L. J. Garcia Villalba, "Natural Language Processing Applied to Forensics Information Extraction With Transformers and Graph Visualization," *IEEE Transactions on Computational Social Systems*, 2022.

[124] M. Kumar, "An Algorithm for Automatic Text Annotation for Named Entity Recognition using spaCy Framework," *Research Square*, pp. 1–18, 2023.

[125] B. Mathis, "Extracting Proceedings Data from Court Cases with Machine Learning," *Stats*, vol. 5, no. 4, pp. 1305–1320, 2022.

[126] D. Pathak, S. Nandi, and P. Sarmah, "AsNER - Annotated Dataset and Baseline for Assamese Named Entity recognition," pp. 6571–6577, 2022.

[127] A. Kumar, B. Starly, and C. Lynch, "ManuBERT: A pretrained Manufacturing science language representation model," *SSRN*, 2023.

[128] L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[129] J. Lafferty, A. Mccallum, and F. Pereira, "Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data Abstract," *Proceedings - Eighteenth International Conference on Machine Learning*, vol. 2001, no. June, pp. 282–289, 1999.

[130] G. Luo, X. Huang, C. Y. Lin, and Z. Nie, "Joint named entity recognition and disambiguation," in *Proceedings - Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015, pp. 879–888.

[131] A. Passos, V. Kumar, and A. McCallum, "Lexicon infused phrase embeddings for named entity resolution," *Proceedings - 18th Conference on Computational Natural Language Learning (CoNLL)*, pp. 78–86, 2014.

[132] T. Eftimov, B. K. Seljak, and P. Korošec, "A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations," *PLoS ONE*, vol. 12, no. 6, p. e0179488, 2017.

[133] G. Popovski, S. Kochev, B. K. Seljak, and T. Eftimov, "FoodIE: A Rule-based Named-entity Recognition Method for Food Information Extraction," *Proceedings - International Conference on Pattern Recognition Applications and Methods*, vol. 1, pp. 915–922, 2019.

[134] G. Cenikj, G. Popovski, R. Stojanov, B. K. Seljak, and T. Eftimov, "BuTTER: BidirecTional LSTM for Food Named-Entity Recognition," in *Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020*. IEEE, 2020, pp. 3550–3556.

[135] C. Radu, C. E. Staicu, L. M. Mitrică, M. Dînsoreanu, R. Potolea, and C. Lemnaru, "Extracting Settings from Multilingual Recipes with Various Sequence Tagging Models: an Experimental Study," in *Proceedings - 18th International Conference on Intelligent Computer Communication and Processing Conference (ICCP)*, 2022, pp. 65–72.

[136] A. K. Brahma, P. Potluri, M. Kanapaneni, S. Prabhu, and S. Teki, "Identification of Food Quality Descriptors in Customer Chat Conversations using Named Entity Recognition," in *ACM International Conference Proceeding Series*, ser. CODS-COMAD '21. New York, NY, USA: Association for Computing Machinery, 2020, pp. 257–261.

[137] G. Cenikj, G. Petelin, B. Korousic Seljak, and T. Eftimov, "SciFoodNER: Food Named Entity Recognition for Scientific Text," in *Proceedings - 2022 IEEE International Conference on Big Data*. IEEE, 2022, pp. 4065–4073.

[138] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," in *Proceedings - 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*. Ann Arbor, Michigan: Association for Computational Linguistics, jun 2005, pp. 363–370.

[139] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings - 2019 Conference of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, vol. 1, 2019, pp. 4171–4186.

[140] H. Matthew, I. Montani, S. Van Landeghem, and B. Adriane, "spaCy Industrial-strength Natural Language Processing in Python," *spaCy*, 2020.

[141] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf, "FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP," in *Proceedings - Association for Computational Linguistics*. Minneapolis, Minnesota: Association for Computational Linguistics, jun 2019, pp. 54–59.

[142] N. Patil, A. Patil, and B. V. Pawar, "Named Entity Recognition using Conditional Random Fields," *Procedia Computer Science*, vol. 167, pp. 1181–1188, 2020.

[143] Q. Wei, T. Chen, R. Xu, Y. He, and L. Gui, "Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks," *Database : the journal of biological databases and curation*, vol. 2016, p. baw140, 2016.

[144] X. Yang and W. Huang, "A conditional random fields approach to clinical name entity recognition," in *CEUR Workshop Proceedings*, vol. 2242, 2018, pp. 1–6.

[145] M. Sato, H. Shindo, I. Yamada, and Y. Matsumoto, "Segment-level neural conditional random fields for named entity recognition," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, 2017, pp. 97–102.

[146] S. Wang, X. Sun, X. Li, R. Ouyang, F. Wu, T. Zhang, J. Li, and G. Wang, "GPT-NER: Named Entity Recognition via Large Language Models," *arXiv:2304.10428*, 2023.

[147] B. Ji, "VicunaNER: Zero/Few-shot Named Entity Recognition using Vicuna," *arXiv:2305.03253*, 2023.

[148] H. Lee, K. Shu, P. Achananuparp, P. K. Prasetyo, Y. Liu, E. P. Lim, and L. R. Varshney, "RecipeGPT: Generative Pre-training Based Cooking Recipe Generation and Evaluation System," in *The Web Conference 2020 - Companion of the World Wide Web Conference (WWW)*, 2020, pp. 181–184.

[149] M. R. Parvez, B. Ray, S. Chakraborty, and K. W. Chang, "Building language models for text with named entities," *Proceedings - 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, vol. 1, pp. 2373–2383, 2018.

[150] Y. Agarwal, D. Batra, and G. Bagler, "Building Hierarchically Disentangled Language Models for Text Generation with Named Entities," in *Proceedings - 28th International Conference on Computational Linguistics (COLING)*, 2020, pp. 26–38.

[151] R. Koncel-Kedziorski, D. Bekal, Y. Luan, M. Lapata, and H. Hajishirzi, "Text generation from knowledge graphs with graph transformers," *Proceedings - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, vol. 1, pp. 2284–2293, 2019.

[152] Y. C. Chen, Z. Gan, Y. Cheng, J. Liu, and J. Liu, "Distilling knowledge learned in BERT for text generation," *Proceedings - Annual Meeting of the Association for Computational Linguistics*, pp. 7893–7905, 2020.

[153] A. Bosselut, A. Celikyilmaz, X. He, J. Gao, P. S. Huang, and Y. Choi, "Discourse-Aware neural rewards for coherent text generation," *Proceedings - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, vol. 1, pp. 173–184, 2018.

[154] R. Alec, W. Jeffrey, C. Rewon, L. David, A. Dario, and S. Ilya, "Language Models are Unsupervised Multitask Learners — Enhanced Reader," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[155] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *Proceedings - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, vol. 1, pp. 4171–4186, 2019.

[156] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "CTRL: A Conditional Transformer Language Model for Controllable Generation," *arXiv:1909.05858*, 2019.

[157] S. Chaudhary, B. Soni, A. Sindhavad, A. Mamaniya, A. Dalvi, and I. Siddavatam, "ChefAI.IN: Generating Indian Recipes with AI Algorithm," in *2022 International Conference on Trends in Quantum Computing and Emerging Business Technologies (TQCEBT)*. IEEE, 2022, pp. 1–6.

[158] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, 2020.

[159] Y. Zhang, S. Sun, M. Galley, Y. C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan, "DIALOGPT: Large-scale generative pre-training for conversational response generation," *Proceedings - Annual Meeting of the Association for Computational Linguistics*, pp. 270–278, 2020.

[160] K. N. Lam, Y. N. T. Pham, and J. Kalita, "Cooking Recipe Generation Based on Ingredients Using ViT5," in *Lecture Notes in Networks and Systems*, vol. 752 LNNS. Springer, 2023, pp. 34–39.

[161] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv:2407.21783*, 2024.

[162] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7B," *arXiv:2310.06825*, 2023.

[163] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLORA: Efficient Finetuning of Quantized LLMs," *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[164] H. Saadany and C. Orăsan, "BLEU, METEOR, BERTScore: Evaluation of Metrics Performance in Assessing Critical Translation Errors in Sentiment-oriented Text," *arXiv:2109.14250*, pp. 48–56, 2022.

[165] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[166] M. Bień, M. Gilski, M. Maciejewska, W. Taisner, D. Wisńiewski, and A. Ławrynowicz, "RecipeNLG: A Cooking Recipes Dataset for Semi-Structured Text Generation," in *Proceedings - 13th International Conference on Natural Language Generation (INLG)*, vol. 2, 2020, pp. 22–28.

[167] M. Crippa, E. Solazzo, D. Guizzardi, F. Monforti-Ferrario, F. N. Tubiello, and A. Leip, "Food systems are responsible for a third of global anthropogenic GHG emissions," *Nature Food*, vol. 2, no. 3, pp. 198–209, 2021.

[168] C. C. Ivanovich, T. Sun, D. R. Gordon, and I. B. Ocko, "Future warming from global food consumption," *Nature Climate Change*, vol. 13, no. 3, pp. 297–302, 2023.

[169] Y. Li, H. Zhong, Y. Shan, Y. Hang, D. Wang, Y. Zhou, and K. Hubacek, "Changes in global food consumption increase GHG emissions despite efficiency gains along global supply chains," *Nature Food*, vol. 4, no. 6, pp. 483–495, 2023.

[170] T. U. Nations, "Food and Climate Change: Healthy diets for a healthier planet — United Nations," 2022.

[171] T. Wiedmann and J. Minx, "A Definition of Carbon Footprint," *Science*, vol. 1, no. 01, pp. 1–11, 2007.

[172] M. A. Clark, N. G. Domingo, K. Colgan, S. K. Thakrar, D. Tilman, J. Lynch, I. L. Azevedo, and J. D. Hill, "Global food system emissions could preclude achieving the 1.5° and 2°C climate change targets," *Science*, vol. 370, no. 6517, pp. 705–708, 2020.

[173] Intergovernmental Panel on Climate Change (IPCC), "Synthesis Report. Geneva: Intergovernmental Panel on Climate Change," 2007.

[174] D. Pandey, M. Agrawal, and J. S. Pandey, "Carbon footprint: Current methods of estimation," *Environmental Monitoring and Assessment*, vol. 178, no. 1-4, pp. 135–160, 2011.

[175] D. Coley, M. Howard, and M. Winter, "Food miles: Time for a re-think?" *British Food Journal*, vol. 113, no. 7, pp. 919–934, jan 2011.

[176] M. Li, N. Jia, M. Lenzen, A. Malik, L. Wei, Y. Jin, and D. Raubenheimer, "Global food-miles account for nearly 20

[177] G. Zervas and E. Tsiplakou, "An assessment of GHG emissions from small ruminants in comparison with GHG emissions from large ruminants and monogastric livestock," pp. 13–23, 2012.

[178] S. H. Vetter, T. B. Sapkota, J. Hillier, C. M. Stirling, J. I. Macdiarmid, L. Aleksandrowicz, R. Green, E. J. Joy, A. D. Dangour, and P. Smith, "Greenhouse gas emissions from agricultural food production to supply Indian diets: Implications for climate change mitigation," *Agriculture, Ecosystems and Environment*, vol. 237, pp. 234–241, 2017.

[179] H. Pathak, N. Jain, A. Bhatia, J. Patel, and P. K. Aggarwal, "Carbon footprints of Indian food items," *Agriculture, Ecosystems and Environment*, vol. 139, no. 1-2, pp. 66–73, 2010.

[180] United Nations, "Growing at a slower pace, world population is expected to reach 9.7 billion in 2050 and could peak at nearly 11 billion around 2100: UN Report - United Nations Sustainable Development," 2019.

[181] G. Bagler, "A generative grammar of cooking," *arxiv:2211.09059*, 2022.

[182] S. González-García, X. Esteve-Llorens, M. T. Moreira, and G. Feijoo, "Carbon footprint and nutritional quality of different human dietary choices," pp. 77–94, 2018.

[183] T. Petersson, L. Secondi, A. Magnani, M. Antonelli, K. Dembska, R. Valentini, A. Varotto, and S. Castaldi, "A multilevel carbon and water footprint dataset of food commodities," *Scientific Data*, vol. 8, no. 1, 2021.

[184] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings - 1st International Conference on Learning Representations, ICLR 2013, Workshop Track*, 2013.

[185] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proceedings - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019, pp. 3982–3992.

[186] X. Xu, P. Sharma, S. Shu, T. S. Lin, P. Ciais, F. N. Tubiello, P. Smith, N. Campbell, and A. K. Jain, "Global greenhouse gas emissions from animal-based foods are twice those of plant-based foods," *Nature Food*, vol. 2, no. 9, pp. 724–732, sep 2021.

[187] M. Z. Jeddi, P. E. Boon, F. Cubadda, R. Hoogenboom, H. Mol, H. Verhagen, and D. T. Sijm, "A vision on the 'foodture' role of dietary exposure sciences in the interplay between food safety and nutrition," *Trends in Food Science and Technology*, vol. 120, pp. 288–300, 2022.

[188] H. Renz, K. J. Allen, S. H. Sicherer, H. A. Sampson, G. Lack, K. Beyer, and H. C. Oettgen, "Food allergy," *Nature Reviews Disease Primers*, vol. 4, no. 1, p. 17098, 2018.

[189] S. Jameel, "Climate change, food systems and the Islamic perspective on alternative proteins," *Trends in Food Science and Technology*, vol. 138, pp. 480–490, 2023.

[190] J. De Boer, H. Schösler, and J. J. Boersema, "Climate change and meat eating: An inconvenient couple?" *Journal of Environmental Psychology*, vol. 33, pp. 1–8, 2013.

[191] M. Goel, S. Khalate, S. Kotaiah, P. Kasoundhan, P. Patel, and G. Bagler, "SustainableFoodDB," 2023. [Online]. Available: https://cosylab.iiitd.edu.in/SustainableFoodDB/

[192] L. Aleksandrowicz, R. Green, E. J. Joy, P. Smith, and A. Haines, "The impacts of dietary change on greenhouse gas emissions, land use, water use, and health: A systematic review," 2016.

[193] Food and Agriculture Organization of the United Nations, "Livestock solutions for climate change," pp. 108–116, 2017.

[194] I. K. Cheng and K. K. Leong, "Data-driven decarbonisation pathways for reducing life cycle GHG emissions from food waste in the hospitality and food service sectors," *Scientific Reports*, vol. 13, no. 1, p. 418, 2023.

[195] D. Inglis and D. Gimlin, "The globalization of food," *Nova York*, 2015.

[196] T. Sharma, U. Upadhyay, and G. Bagler, "Classification of cuisines from sequentially structured recipes," in *Proceedings - 2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW)*, 2020, pp. 105–108.

[197] H. Su, T. W. Lin, C. T. Li, M. K. Shan, and J. Chang, "Automatic recipe cuisine classification by ingredients," in *Proceedings - 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014, pp. 565–570.

[198] C. Y. Teng, Y. R. Lin, and L. A. Adamic, "Recipe recommendation using ingredient networks," in *Proceedings of the 4th Annual ACM Web Science Conference, WebSci'12*, 2012, pp. 298–307.

[199] Y. Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A. L. Barabási, "Flavor network and the principles of food pairing," *Scientific Reports*, vol. 1, p. 196, 2011.

[200] Y. Y. Ahn and S. Ahnert, "The flavor network," *Leonardo*, vol. 46, no. 3, pp. 272–273, 2013.

[201] T. Kusmierczyk, C. Trattner, and K. Nørvag, "Temporal patterns in online food innovation," in *Proceedings - 24th International Conference on World Wide Web*, 2015, pp. 1345–1350.

[202] R. Azzi, S. Despres, and G. Diallo, "NutriSem: A Semantics-Driven Approach to Calculating Nutritional Value of Recipes," in *Advances in Intelligent Systems and Computing*, vol. 1159 AISC. Springer, 2020, pp. 191–201.

[203] R. Fallaize, R. Z. Franco, F. Hwang, and J. A. Lovegrove, "Evaluation of the eNutri automated personalised nutrition advice by users and nutrition professionals in the UK," *PLoS ONE*, vol. 14, no. 4, p. e0214931, 2019.

[204] J. Chen and C. W. Ngo, "Deep-based ingredient recognition for cooking recipe retrieval," in *Proceedings - 2016 ACM Multimedia Conference*, 2016, pp. 32–41.

[205] G. LeCun, Yann and Bengio, Yoshua and Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[206] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings - ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-Augu, 2016, pp. 785–794.

[207] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, pp. 3147–3155, 2017.

[208] R. S. Abdul Kareem, T. Tilford, and S. Stoyanov, "Fine-grained food image classification and recipe extraction using a customized deep neural network and NLP," *Computers in Biology and Medicine*, vol. 175, p. 108528, 2024.

[209] E. Mohammadi, N. Naji, L. Marceau, M. Queudot, E. Charton, L. Kosseim, and M. J. Meurs, "Cooking up a neural-based model for recipe classification," in *12th International Conference on Language Resources and Evaluation, Conference*, 2020, pp. 5000–5009.

[210] M. Goel, S. Dargar, S. Ghatak, N. Verma, P. Chauhan, A. Gupta, N. Vishnumolakala, H. Amuru, E. Gambhir, R. Chhajed, M. Jain, A. Jain, S. Garg, and G. Bagler, "Dish Detection in Indian Food Platters: A Computational Framework for Diet Management," in *Communications in Computer and Information Science*, vol. 2009 CCIS. Springer, 2024, pp. 231–243.

[211] P. McAllister, H. Zheng, R. Bond, and A. Moorhead, "Combining deep residual neural network features with supervised machine learning algorithms to classify diverse food image datasets," *Computers in Biology and Medicine*, vol. 95, pp. 217–233, 2018.

[212] J. F. Kennedy and I. R. Cosnett, "Food flavours biology and chemistry," *Carbohydrate Polymers*, vol. 46, no. 3, p. 296, nov 2001.

[213] B. Malnic, J. Hirono, T. Sato, and L. B. Buck, "Combinatorial receptor codes for odors," *Cell*, vol. 96, no. 5, pp. 713–723, 1999.

[214] O. G. Mouritsen, "The science of taste," *Flavour*, vol. 4, no. 1, 2015.

[215] R. D. Newcomb and K. Ohla, "The genetics and neuroscience of flavour," *Flavour*, vol. 2, no. 1, dec 2013.

[216] G. M. Shepherd, "Neuroenology: how the brain creates the taste of wine," *Flavour*, vol. 4, no. 1, 2015.

[217] G. A. Burdock, "Fenaroli's Handbook of Flavor Ingredients," *Fenaroli's Handbook of Flavor Ingredients*, 2004.

[218] E. Chambers IV and K. Koppel, "Associations of volatile compounds with sensory aroma and flavor: The complex nature of flavor," pp. 4887–4905, apr 2013.

[219] C. Spence, C. Hobkinson, A. Gallace, and B. P. Fiszman, "A touch of gastronomy," *Flavour*, vol. 2, no. 1, pp. 1–15, feb 2013.

[220] H. This, "Molecular gastronomy," pp. 5–7, 2005.

[221] A. Holovaty and J. Kaplan-Moss, *The Definitive Guide to Django: Web Development Done Right*. Apress Berkeley, CA, 2009.

[222] E. Ziaikin, M. David, S. Uspenskaya, and M. Y. Niv, "Bitterdb: 2024 update on bitter ligands and taste receptors," *Nucleic acids research*, vol. 53, no. D1, pp. D1645–D1650, 2025.

[223] E. A. Hamel, J. B. Castro, T. J. Gould, R. Pellegrino, Z. Liang, L. A. Coleman, F. Patel, D. S. Wallace, T. Bhatnagar, J. D. Mainland *et al.*, "Pyrfume: A window to the world's olfactory data," *Scientific Data*, vol. 11, no. 1, p. 1220, 2024.

[224] R. H. Lustig, L. A. Schmidt, and C. D. Brindis, "Public health: The toxic truth about sugar," *Nature*, vol. 482, no. 7383, pp. 27–29, 2012.

[225] C. Kuhn, B. Bufe, M. Winnig, T. Hofmann, O. Frank, M. Behrens, T. Lewtschenko, J. P. Slack, C. D. Ward, and W. Meyerhof, "Bitter taste receptors for saccharin and acesulfame K," *Journal of Neuroscience*, vol. 24, no. 45, pp. 10 260–10 265, 2004.

[226] A. Goel, K. Gajula, R. Gupta, and B. Rai, "In-silico prediction of sweetness using structure-activity relationship models," *Food Chemistry*, vol. 253, pp. 127–131, 2018.

[227] M. E. Lean and L. Te Morenga, "Sugar and type 2 diabetes," pp. 43–53, 2016.

[228] H. Iwamura, "Structure-Sweetness Relationship of L-Aspartyl Dipeptide Analogues. A Receptor Site Topology," *Journal of Medicinal Chemistry*, vol. 24, no. 5, pp. 572–583, 1981.

[229] W. J. Spillane and M. B. Sheahan, "Semi-quantitative and quantitative structure-taste relationships for carbo-and hetero-sulphamate (RNHSO3-) sweeteners," *Journal of the Chemical Society, Perkin Transactions 2*, no. 7, pp. 741–746, 1989.

[230] A. D. Kinghorn and D. D. Soejarto, "Discovery of terpenoid and phenolic sweeteners from plants," in *Pure and Applied Chemistry*, vol. 74, no. 7. Walter de Gruyter GmbH, jul 2002, pp. 1169–1179.

[231] S. B. Vepuri, N. R. Tawari, and M. S. Degani, "Quantitative structure-activity relationship study of some aspartic acid analogues to correlate and predict their sweetness potency," *QSAR and Combinatorial Science*, vol. 26, no. 2, pp. 204–214, 2007.

[232] M. Zhong, Y. Chong, X. Nie, A. Yan, and Q. Yuan, "Prediction of sweetness by multilinear regression analysis and support vector machine," *Journal of Food Science*, vol. 78, no. 9, sep 2013.

[233] M. G. Drew, G. R. Wilden, W. J. Spillane, R. M. Walsh, C. A. Ryder, and J. M. Simmie, "Quantitative Structure-Activity Relationship Studies of Sulfamates RNHSO3Na: Distinction between Sweet, Sweet-Bitter, and Bitter Molecules," *Journal of Agricultural and Food Chemistry*, vol. 46, no. 8, pp. 3016–3026, 1998.

[234] J. S. Barker, C. K. Hattotuwagama, and M. G. Drew, "Computational studies of sweet-tasting molecules," *Pure and Applied Chemistry*, vol. 74, no. 7, pp. 1207–1217, jul 2002.

[235] A. Bassoli, M. G. Drew, C. K. Hattotuwagama, L. Merlini, G. Morini, and G. R. Wilden, "Quantitative structure-activity relationships of sweet isovanillyl derivatives," *Quantitative Structure-Activity Relationships*, vol. 20, no. 1, pp. 3–16, 2001.

[236] C. Rojas, R. Todeschini, D. Ballabio, A. Mauri, V. Consonni, P. Tripaldi, and F. Grisoni, "A QSTR-based expert system to predict sweetness of molecules," *Frontiers in Chemistry*, vol. 5, no. JUL, p. 53, 2017.

[237] P. K. Ojha and K. Roy, "Development of a robust and validated 2D-QSPR model for sweetness potency of diverse functional organic molecules," *Food and Chemical Toxicology*, vol. 112, pp. 551–562, 2018.

[238] S. Zheng, W. Chang, W. Xu, Y. Xu, and F. Lin, "e-Sweet: A machine-learning based platform for the prediction of sweetener and its relative sweetness," *Frontiers in Chemistry*, vol. 7, no. JAN, p. 35, 2019.

[239] W. Bo, D. Qin, X. Zheng, Y. Wang, B. Ding, Y. Li, and G. Liang, "Prediction of bitterant and sweetener using structure-taste relationship models based on an artificial neural network," *Food Research International*, vol. 153, p. 110974, 2022.

[240] X. Yang, Y. Chong, A. Yan, and J. Chen, "In-silico prediction of sweetness of sugars and sweeteners," *Food Chemistry*, vol. 128, no. 3, pp. 653–658, 2011.

[241] N. M. O'Boyle, C. Morley, and G. R. Hutchison, "Pybel: A Python wrapper for the OpenBabel cheminformatics toolkit," *Chemistry Central Journal*, vol. 2, no. 1, 2008.

[242] C. W. Yap, "PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints," *Journal of Computational Chemistry*, vol. 32, no. 7, pp. 1466–1474, may 2011.

[243] H. Moriwaki, Y. S. Tian, N. Kawashita, and T. Takagi, "Mordred: A molecular descriptor calculator," *Journal of Cheminformatics*, vol. 10, no. 1, pp. 1–14, 2018.

[244] H. Singh, S. Singh, D. Singla, S. M. Agarwal, and G. P. Raghava, "QSAR based model for discriminating EGFR inhibitors and non-inhibitors using Random forest," *Biology Direct*, vol. 10, no. 1, pp. 1–12, 2015.

[245] D. Bowen and L. Ungar, "Generalized SHAP: Generating multiple types of explanations in machine learning," *arXiv:2006.07155*, 2020.

[246] I. E. Hartley, D. G. Liem, and R. Keast, "Umami as an 'Alimentary' taste. A new perspective on taste classification," *Nutrients*, vol. 11, no. 1, p. 182, 2019.

[247] B. Lindemann, Y. Ogiwara, and Y. Ninomiya, "The discovery of umami," *Chemical Senses*, vol. 27, no. 9, pp. 843–844, 2002.

[248] S. Yamaguchi and K. Ninomiya, "Umami and food palatability," *Journal of Nutrition*, vol. 130, no. 4 SUPPL., pp. 921S—-926S, 2000.

[249] P. Hajeb and S. Jinap, "Umami Taste Components and Their Sources in Asian Foods," *Critical Reviews in Food Science and Nutrition*, vol. 55, no. 6, pp. 778–791, 2015.

[250] X. Li, L. Staszewski, H. Xu, K. Durick, M. Zoller, and E. Adler, "Human receptors for sweet and umami taste," pp. 4692–4696, 2002.

[251] Y. Maruyama, E. Pereira, R. F. Margolskee, N. Chaudhari, and S. D. Roper, "Umami responses in mouse taste cells indicate more than one receptor," *Journal of Neuroscience*, vol. 26, no. 8, pp. 2227–2234, 2006.

[252] R. Xia, Y. Qiao, H. Xu, Z. Hou, G. Qian, Y. Wang, Y. Li, M. Yan, S. Pan, and G. Xin, "Unlocking the Potential of the Umami Taste-Presenting Compounds: A Review of the Health Benefits, Metabolic Mechanisms and Intelligent Detection Strategies," *Food Reviews International*, pp. 1–21, 2024.

[253] P. Charoenkwan, J. Yana, C. Nantasenamat, M. M. Hasan, and W. Shoombuatong, "IUmami-SCM: A Novel Sequence-Based Predictor for Prediction and Analysis of Umami Peptides Using a Scoring Card Method with Propensity Scores of Dipeptides," *Journal of Chemical Information and Modeling*, vol. 60, no. 12, pp. 6666–6678, 2020.

[254] L. E. Carpio, Y. Sanz, R. Gozalbes, and S. J. Barigye, "Computational strategies for the discovery of biological functions of health foods, nutraceuticals and cosmeceuticals: a review," *Molecular Diversity*, vol. 25, no. 3, pp. 1425–1438, 2021.

[255] N. Pulikkal, N. P. Sukumaran, and D. Haridas, "Virtual Screening: An In Silico Approach to Flavor Compounds," *Natural Flavors, Fragrances, and Perfumes: Chemistry, Production, and Sensory Approach*, pp. 207–224, 2021.

[256] H. Xiu, Y. Liu, H. Yang, H. Ren, B. Luo, Z. Wang, H. Shao, F. Wang, J. Zhang, and Y. Wang, "Identification of novel umami molecules via QSAR models and molecular docking," *Food and Function*, vol. 13, no. 14, pp. 7529–7539, 2022.

[257] J. Zhang, D. Sun-Waterhouse, G. Su, and M. Zhao, "New insight into umami receptor, umami/umami-enhancing peptides and their derivatives: A review," *Trends in Food Science and Technology*, vol. 88, pp. 429–438, 2019.

[258] Y. Zhang, X. Bao, Y. Zhu, Z. Dai, Q. Shen, and Y. Xue, "Advances in machine learning screening of food bioactive compounds," *Trends in Food Science and Technology*, vol. 150, p. 104578, 2024.

[259] P. Charoenkwan, C. Nantasenamat, M. M. Hasan, M. A. Moni, B. Manavalan, and W. Shoombuatong, "UMPred-FRL: A new approach for accurate prediction of umami peptides using feature representation learning," *International Journal of Molecular Sciences*, vol. 22, no. 23, 2021.

[260] L. Pallante, A. Korfiati, L. Androutsos, F. Stojceski, A. Bompotas, I. Giannikos, C. Raftopoulos, M. Malavolta, G. Grasso, S. Mavroudi, A. Kalogeras, V. Martos, D. Amoroso, D. Piga, K. Theofilatos, and M. A. Deriu, "Toward a general and interpretable umami taste predictor using a multi-objective machine learning approach," *Scientific Reports*, vol. 12, no. 1, 2022.

[261] L. Jiang, J. Jiang, X. Wang, Y. Zhang, B. Zheng, S. Liu, Y. Zhang, C. Liu, Y. Wan, D. Xiang, and Z. Lv, "IUP-BERT: Identification of Umami Peptides Based on BERT Features," *Foods*, vol. 11, no. 22, 2022.

[262] A. P. Indiran, H. Fatima, S. Chattopadhyay, S. Ramadoss, and Y. Radhakrishnan, "UmamiPreDL: Deep learning model for umami taste prediction of peptides using BERT and CNN," 2024.

[263] P. Dutta, K. Gajula, N. Verma, D. Jain, R. Gupta, and B. Rai, "Computational screening of umami tastants using deep learning," 2024.

[264] N. Hollmann, S. Müller, K. Eggensperger, and F. Hutter, "TabPFN: A transformer that solves small tabular classification problems in a second," *11th International Conference on Learning Representations, ICLR 2023*, 2023.

[265] J. Zhang, W. Yan, Q. Zhang, Z. Li, L. Liang, M. Zuo, and Y. Zhang, "Umami-BERT: An interpretable BERT-based model for umami peptides prediction," *Food Research International*, vol. 172, p. 113142, 2023.

[266] L. Qi, J. Du, Y. Sun, Y. Xiong, X. Zhao, D. Pan, Y. Zhi, Y. Dang, and X. Gao, "Umami-MRNN: Deep learning-based prediction of umami peptide using RNN and MLP," *Food Chemistry*, vol. 405, p. 134935, 2023.

[267] J. B. Findlay and D. J. Pappin, "The opsin family of proteins." *The Biochemical Journal*, vol. 238, no. 3, pp. 625–642, 1986.

[268] D. M. Lowe, P. T. Corbett, P. Murray-Rust, and R. C. Glen, "Chemical name to structure: OPSIN, an open source solution," *Journal of Chemical Information and Modeling*, vol. 51, no. 3, pp. 739–753, 2011.

[269] S. Jaeger, S. Fulle, and S. Turk, "Mol2vec: Unsupervised Machine Learning Approach with Chemical Intuition," *Journal of Chemical Information and Modeling*, vol. 58, no. 1, pp. 27–35, 2018.

[270] S. Giri and A. Bader, "A low-cost, high-quality new drug discovery process using patient-derived induced pluripotent stem cells," *Drug Discovery Today*, vol. 20, no. 1, pp. 37–49, 2015.

[271] J. P. Shonkoff, A. S. Garner, B. S. Siegel, M. I. Dobbins, M. F. Earls, L. McGuinn, J. Pascoe, and D. L. Wood, "The lifelong effects of early childhood adversity and toxic stress," *Pediatrics*, vol. 129, no. 1, pp. e232–e246, 2012.

[272] J. V. Rodricks, *Calculated risks: The toxicity and human health risks of chemicals in our environment*. Cambridge University Press, 2006.

[273] H. Ketha and U. Garg, *Toxicology cases for the clinical and forensic laboratory*. Academic Press, 2020.

[274] A. Seaton, D. Godden, W. MacNee, and K. Donaldson, "Particulate air pollution and acute health effects," *The Lancet*, vol. 345, no. 8943, pp. 176–178, 1995.

[275] S. Nowicki and E. Gottlieb, "Oncometabolites: tailoring our genes," *The FEBS Journal*, vol. 282, no. 15, pp. 2796–2805, 2015.

[276] E. Borenfreund and J. A. Puerner, "Toxicity determined in vitro by morphological alterations and neutral red absorption," *Toxicology Letters*, vol. 24, no. 2-3, pp. 119–124, 1985.

[277] G. J. Harry, M. Billingsley, A. Bruinink, I. L. Campbell, W. Classen, D. C. Dorman, C. Galli, D. Ray, R. A. Smith, and H. A. Tilson, "In vitro techniques for the assessment of neurotoxicity." *Environmental Health Perspectives*, vol. 106, no. suppl 1, pp. 131–158, 1998.

[278] H. Van de Waterbeemd, "From in vivo to in vitro/in silico adme: progress and challenges," *Expert Opinion on Drug Metabolism and Toxicology*, vol. 1, no. 1, pp. 1–4, 2005.

[279] G. A. Van Norman, "Phase ii trials in drug development and adaptive trial design," *JACC: Basic to Translational Science*, vol. 4, no. 3, pp. 428–437, 2019.

[280] D. E. Pires, W. N. Veloso, Y. Myung, C. H. Rodrigues, M. Silk, P. M. Rezende, F. Silva, J. S. Xavier, J. P. Velloso, C. H. da Silveira *et al.*, "Easyvs: a user-friendly web-based tool for molecule library selection and structure-based virtual screening," *Bioinformatics*, vol. 36, no. 14, pp. 4200–4202, 2020.

[281] P. M. Hinderliter, K. R. Minard, G. Orr, W. B. Chrisler, B. D. Thrall, J. G. Pounds, and J. G. Teeguarden, "Isdd: A computational model of particle sedimentation, diffusion and target cell dosimetry for in vitro toxicity studies," *Particle and Fibre Toxicology*, vol. 7, pp. 1–20, 2010.

[282] P. Banerjee, A. O. Eckert, A. K. Schrey, and R. Preissner, "Protox-ii: a webserver for the prediction of toxicity of chemicals," *Nucleic Acids Research*, vol. 46, no. W1, pp. W257–W263, 2018.

[283] A. Mayr, G. Klambauer, T. Unterthiner, and S. Hochreiter, "Deeptox: toxicity prediction using deep learning," *Frontiers in Environmental Science*, vol. 3, p. 80, 2016.

[284] A. G. de Sá, Y. Long, S. Portelli, D. E. Pires, and D. B. Ascher, "toxcsm: comprehensive prediction of small molecule toxicity profiles," *Briefings in Bioinformatics*, vol. 23, no. 5, p. bbac337, 2022.

[285] A. K. Sharma, G. N. Srivastava, A. Roy, and V. K. Sharma, "Toxim: a toxicity prediction tool for small molecules developed using machine learning and chemoinformatics approaches," *Frontiers in Pharmacology*, vol. 8, p. 880, 2017.

[286] A. Setiya, V. Jani, U. Sonavane, and R. Joshi, "Moltoxpred: small molecule toxicity prediction using machine learning approach," *RSC Advances*, vol. 14, no. 6, pp. 4201–4220, 2024.

[287] J. Wenzel, H. Matter, and F. Schmidt, "Predictive multitask deep neural network models for adme-tox properties: learning from large data sets," *Journal of Chemical Information and Modeling*, vol. 59, no. 3, pp. 1253–1268, 2019.

[288] J. Jiang, R. Wang, and G.-W. Wei, "Ggl-tox: geometric graph learning for toxicity prediction," *Journal of Chemical Information and Modeling*, vol. 61, no. 4, pp. 1691–1700, 2021.

[289] A. Karim, A. Mishra, M. H. Newton, and A. Sattar, "Efficient toxicity prediction via simple features using shallow neural networks and decision trees," *Acs Omega*, vol. 4, no. 1, pp. 1874–1888, 2019.

[290] C. N. Cavasotto and V. Scardino, "Machine learning toxicity prediction: latest advances by toxicity end point," *ACS Omega*, vol. 7, no. 51, pp. 47 536–47 546, 2022.

[291] L. Zhang, H. Zhang, H. Ai, H. Hu, S. Li, J. Zhao, and H. Liu, "Applications of machine learning methods in drug toxicity prediction," *Current Topics in Medicinal Chemistry*, vol. 18, no. 12, pp. 987–997, 2018.

[292] Y. Wu and G. Wang, "Machine learning based toxicity prediction: from chemical structural description to transcriptome analysis," *International Journal of Molecular Sciences*, vol. 19, no. 8, p. 2358, 2018.

[293] J. Dong, D.-S. Cao, H.-Y. Miao, S. Liu, B.-C. Deng, Y.-H. Yun, N.-N. Wang, A.-P. Lu, W.-B. Zeng, and A. F. Chen, "Chemdes: an integrated web-based platform for molecular descriptor and fingerprint computation," *Journal of Cheminformatics*, vol. 7, pp. 1–10, 2015.

[294] L. Xue and J. Bajorath, "Molecular descriptors in chemoinformatics, computational combinatorial chemistry, and virtual screening," *Combinatorial Chemistry and High Throughput Screening*, vol. 3, no. 5, pp. 363–372, 2000.

[295] R. Huang, M. Xia, D.-T. Nguyen, T. Zhao, S. Sakamuru, J. Zhao, S. A. Shahane, A. Rossoshek, and A. Simeonov, "Tox21challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs," *Frontiers in Environmental Science*, vol. 3, p. 85, 2016.

[296] L. Pu, M. Naderi, T. Liu, H.-C. Wu, S. Mukhopadhyay, and M. Brylinski, "e toxpred: A machine learning-based approach to estimate the toxicity of drug candidates," *BMC Pharmacology and Toxicology*, vol. 20, pp. 1–15, 2019.

[297] M. Kanehisa, S. Goto, M. Furumichi, M. Tanabe, and M. Hirakawa, "Kegg for representation and analysis of molecular networks involving diseases and drugs," *Nucleic Acids Research*, vol. 38, no. suppl_1, pp. D355–D360, 2010.

[298] G. C. Fonger, D. Stroup, P. L. Thomas, and P. Wexler, "Toxnet: A computerized collection of toxicological and environmental health information," *Toxicology and Industrial Health*, vol. 16, no. 1, pp. 4–6, 2000.

[299] D. Wishart, D. Arndt, A. Pon, T. Sajed, A. C. Guo, Y. Djoumbou, C. Knox, M. Wilson, Y. Liang, J. Grant *et al.*, "T3db: the toxic exposome database," *Nucleic Acids Research*, vol. 43, no. D1, pp. D928–D934, 2015.

[300] C. Y.-C. Chen, "Tcm database taiwan: the world's largest traditional chinese medicine database for drug screening in silico," *PLOS One*, vol. 6, no. 1, p. e15939, 2011.

[301] Z. A. King, J. Lu, A. Dräger, P. Miller, S. Federowicz, J. A. Lerman, A. Ebrahim, B. O. Palsson, and N. E. Lewis, "Bigg models: A platform for integrating, standardizing and sharing genome-scale models," *Nucleic Acids Research*, vol. 44, no. D1, pp. D515–D522, 2016.

[302] D. Weininger, "Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules," *Journal of Chemical Information and Computer Sciences*, vol. 28, no. 1, pp. 31–36, 1988.

[303] M. B. Kursa, A. Jankowski, and W. R. Rudnicki, "Boruta–a system for feature selection," *Fundamenta Informaticae*, vol. 101, no. 4, pp. 271–285, 2010.

[304] M. Pal, "Random forest classifier for remote sensing classification," *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217–222, 2005.

[305] A. Makiewicz and W. Ratajczak, "Principal components analysis (pca)," *Computers and Geosciences*, vol. 19, no. 3, pp. 303–342, 1993.

[306] R. Bardenet, M. Brendel, B. Kegl, and M. Sebag, "Collaborative hyperparameter tuning," in *International Conference on Machine Learning*. PMLR, 2013, pp. 199–207.

[307] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[308] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of Artificial Intelligence Research*, vol. 61, pp. 863–905, 2018.

[309] G. Van den Broeck, A. Lykov, M. Schleich, and D. Suciu, "On the tractability of shap explanations," *Journal of Artificial Intelligence Research*, vol. 74, pp. 851–886, 2022.

[310] B. Bienfait and P. Ertl, "Jsme: a free molecule editor in javascript," *Journal of Cheminformatics*, vol. 5, pp. 1–6, 2013.

[311] D. Pandey, P. Parmar, G. Toshniwal, M. Goel, V. Agrawal, S. Dhiman, L. Gupta, and G. Bagler, "Object Detection in Indian Food Platters using Transfer Learning with YOLOv4," in *Proceedings - 2022 IEEE 38th International Conference on Data Engineering Workshops, ICDEW 2022*, 2022, pp. 101–106.

[312] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[313] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[314] S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, and D. Traore, "Deep Convolution Neural Network sharing for the multi-label images classification," *Machine Learning with Applications*, vol. 10, p. 100422, 2022.

[315] P. Pandey, A. Deepthi, B. Mandal, and N. B. Puhan, "FoodNet: Recognizing Foods Using Ensemble of Deep Networks," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1758–1762, 2017.

[316] G. Amato, P. Bolettieri, V. M. De Lira, C. I. Muntean, R. Perego, and C. Renso, "Social media image recognition for food trend analysis," in *SIGIR 2017 - Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 1333–1336.

[317] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, 2015, pp. 1–9.

[318] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101–mining discriminative components with random forests," in *European Conference on Computer Vision*. Springer, 2014, pp. 446–461.

[319] S. J. Minija and W. R. Emmanuel, "Food image classification using sphere shaped - Support vector machine," in *Proceedings of the International Conference on Inventive Computing and Informatics, ICICI 2017*. IEEE, 2018, pp. 109–113.

[320] H. Kagaya, K. Aizawa, and M. Ogawa, "Food detection and recognition using convolutional neural network," in *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia*, 2014, pp. 1085–1088.

[321] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," in *Proceedings - IEEE International Conference on Multimedia and Expo*. IEEE, 2012, pp. 25–30.

[322] Y. Kawano and K. Yanai, "Food image recognition with deep convolutional features," in *Proceedings - 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2014, pp. 589–593.

[323] K. Yanai and Y. Kawano, "Food image recognition using deep convolutional network with pre-training and fine-tuning," in *2015 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2015*. IEEE, 2015, pp. 1–6.

[324] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *European Conference on Computer Vision*. Springer, 2015, pp. 3–17.

[325] A. Myers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. Murphy, "Im2Calories: Towards an automated mobile vision food diary," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1233–1241.

[326] C. Termritthikun and S. Kanprachar, "Accuracy improvement of Thai food image recognition using deep convolutional neural networks," in *2017 International Electrical Engineering Congress, iEECON*. IEEE, 2017, pp. 1–4.

[327] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2001, pp. 511–518.

[328] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, pp. 137–154, 2004.

[329] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings - IEEE International Conference on Computer Vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[330] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proceedings - British Machine Vision Conference (BMVC)*. BMVA Press, 2009, pp. 121–124.

[331] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proceedings - European Conference on Computer Vision*, vol. 3951 LNCS. Springer, 2006, pp. 404–417.

[332] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[333] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[334] X. T. Vo, T. D. Tran, D. L. Nguyen, and K. H. Jo, "Stair-Step Feature Pyramid Networks for Object Detection," in *Communications in Computer and Information Science*, vol. 1405, 2021, pp. 168–175.

[335] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, 2016, pp. 779–788.

[336] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European Conference on Computer Vision*, vol. 9905 LNCS. Springer, 2016, pp. 21–37.

[337] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, 2020, pp. 318–327.

[338] H. Law and J. Deng, "CornerNet: Detecting Objects as Paired Keypoints," in *International Journal of Computer Vision*, vol. 128, no. 3, 2020, pp. 642–656.

[339] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "DEFORMABLE DETR: DEFORMABLE TRANSFORMERS FOR END-TO-END OBJECT DETECTION," *ICLR 2021 - 9th International Conference on Learning Representations*, 2021.

[340] M. Bolanos and P. Radeva, "Simultaneous food localization and recognition," in *Proceedings - International Conference on Pattern Recognition*, vol. 0, 2016, pp. 3140–3145.

[341] Q. Cai, J. Li, H. Li, and Y. Weng, "BTBUFood-60: Dataset for Object Detection in Food Field," in *Proceedings - 2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 2019, pp. 1–4.

[342] A. Ramesh, A. Sivakumar, and S. Sherly Angel, "Real-time Food-Object Detection and Localization for Indian Cuisines using Deep Neural Networks," in *Proceedings - 2020 IEEE International Conference on Machine Learning and Applied Network Technologies, ICMLANT*. IEEE, 2020, pp. 1–6.

[343] A. L. Nobles, E. C. Leas, S. Noar, M. Dredze, C. A. Latkin, S. A. Strathdee, and J. W. Ayers, "Automated image analysis of instagram posts: Implications for risk perception and communication in public health using a case study of HIV," *PLoS ONE*, vol. 15, no. 5, p. e0231155, 2020.

[344] M. Hewitt, "Make Sense," https://github.com/SkalskiP/make-sense/, pp. 395–438, 2006. [Online]. Available: https://www.makesense.ai/

[345] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Proceedings of 3rd International Conference on Learning Representations (ICLR)*, 2015.

[346] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, 2016, pp. 770–778.

[347] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[348] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[349] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020.

[350] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," *arXiv:2207.02696*, pp. 7464–7475, 2023.

[351] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*. IEEE, 2017, pp. 464–472.

[352] Y. Gu, Y. Niu, J. Zhang, B. Sun, X. Mao, Z. Liu, and Y. Zhang, "Identification of novel umami peptides from yeast protein through enzymatic, sensory, and in silico approaches," *Journal of Agricultural and Food Chemistry*, vol. 72, no. 36, pp. 20 014–20 027, 2024.