

Reducing Test Time Using Regression Algorithms in Post Silicon Validation

Shivani Mamodia

Student, IIITD

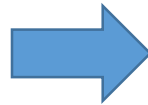
Intern, STMicroelectronics

Overview

- I have Prepared 2 scripts.
- Language used : Python.
- Script 1: Used to generate dataset in the format on which ML algorithms can be implemented.
 - It takes the dataset provided by STM(eg, excel file for Eiger10) as input. On completion of script, it creates another excel sheet that transforms the input data in desired format.
- Script 2: Takes excel sheet from script 1 and implements three different regression (ML) algorithms on it.

Script 1

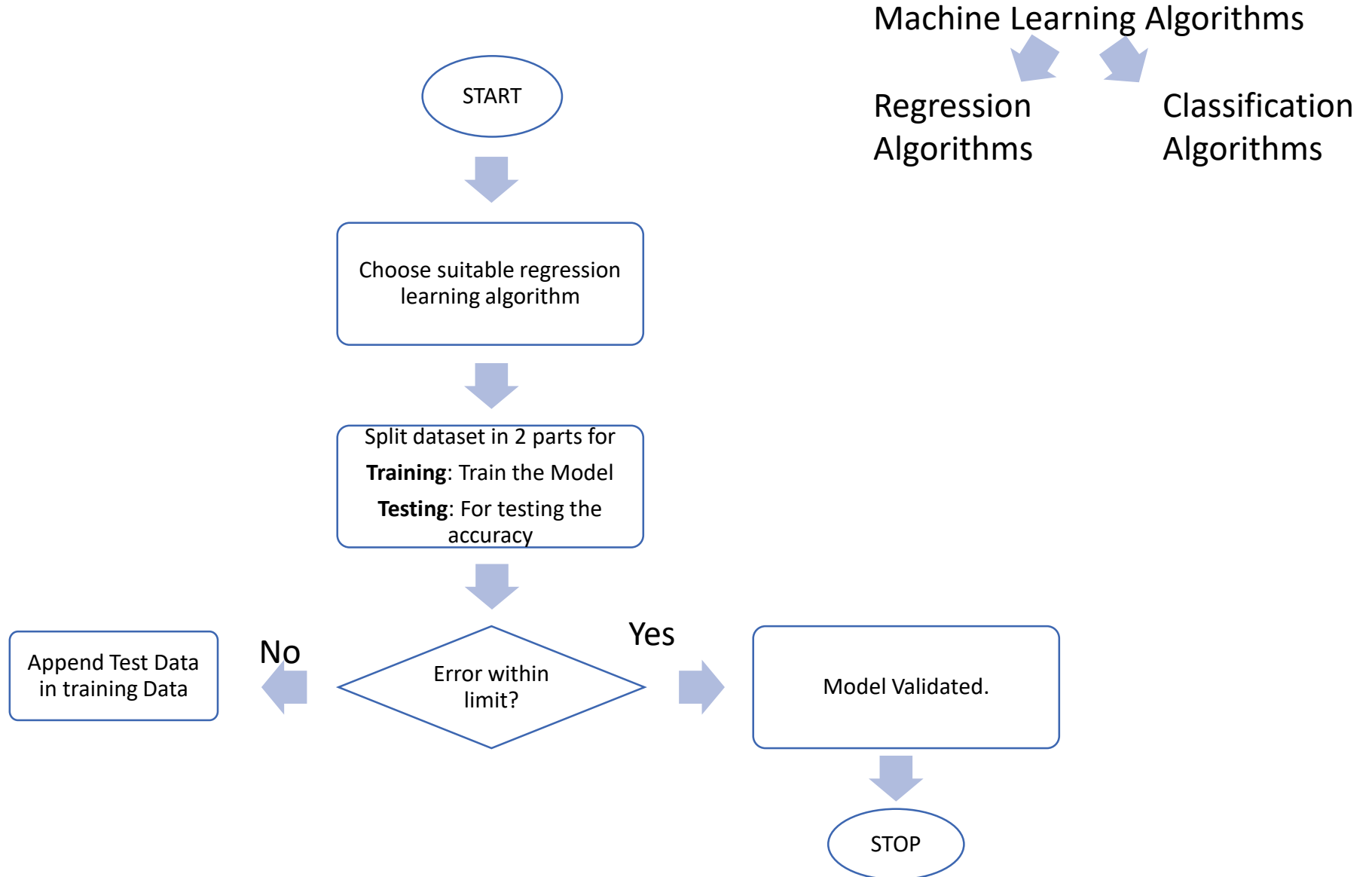
Temp	INL	DNL
165	0.9	0.89
25	0.85	0.98
-45	0.78	0.99



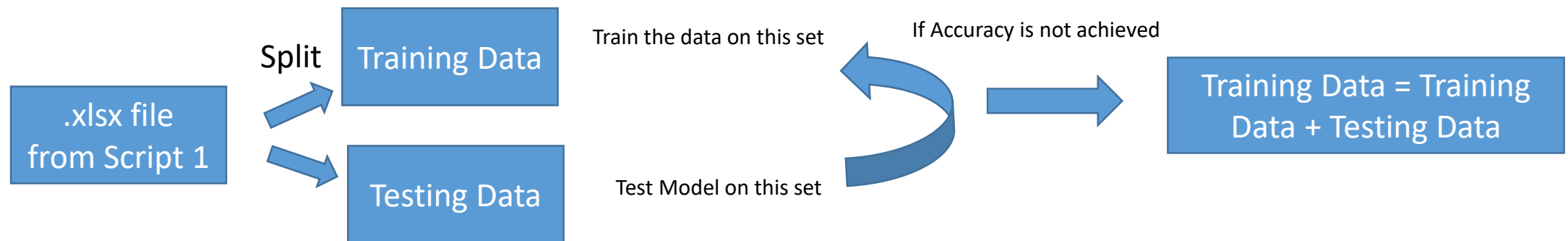
INL_165	INL_25	INL_45	DNL_165	DNL_25	DNL_45
0.9	0.85	0.78	0.89	0.98	0.99

- This step is done since our target is to provide data for only two temperatures (say 25 and -45 C) and predict values for 165 C based on the dataset we have.

Script 2:



- For any ML algorithm to implement, it is important to have a strong training dataset. So our plan is to start with dataset of one product(for SARADC 12 bits).
- We will split this dataset into two in 80:20 ratio. The larger among the two will be used for training our model. And the other will be used for testing the accuracy of our model.
- Initially accuracy will be very less. Hence we will keep on appending our training data set with testing dataset until we get reasonable accuracy.



Goal:

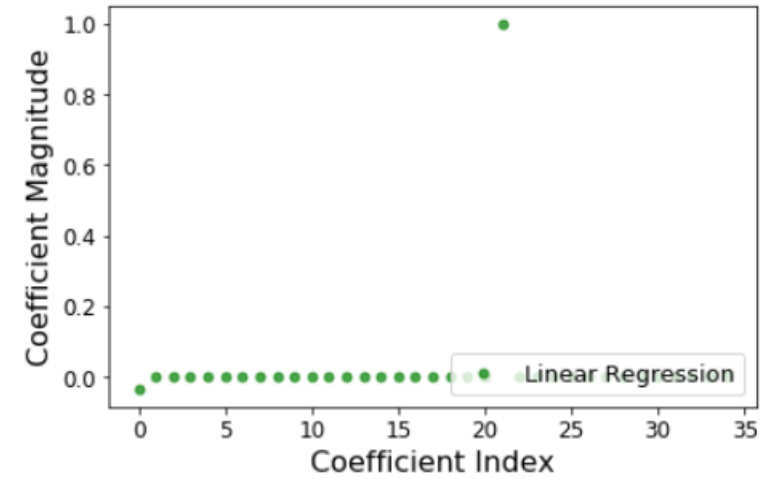
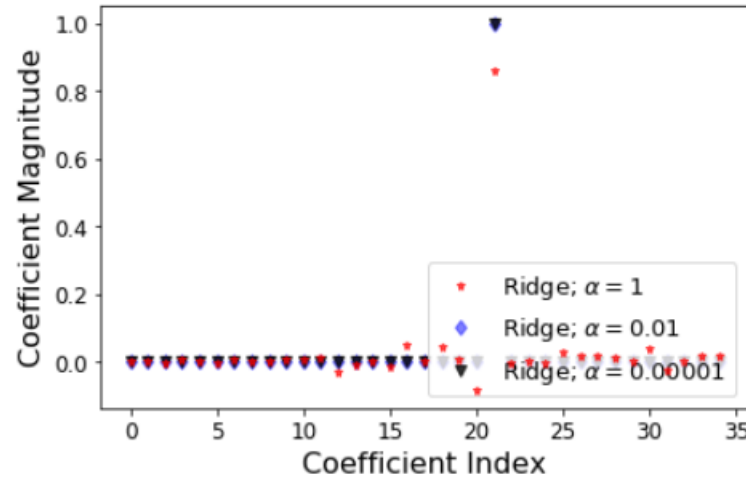
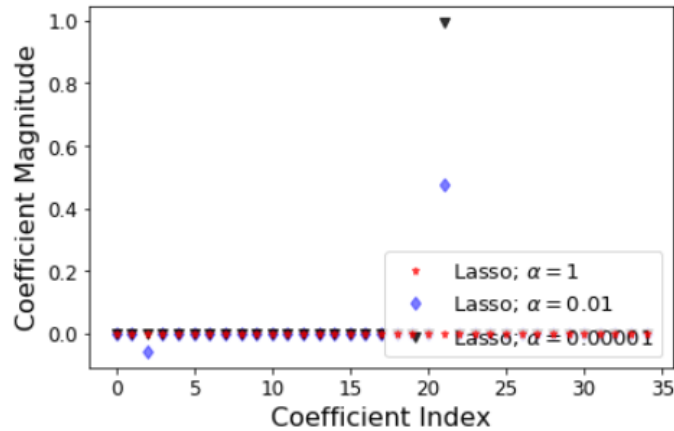
- We are trying to reduce test time by reducing test points.
 - While measuring static/Dynamic parameters for SARADC 12 bits, we measure it across different temperatures and voltage ranges.
 - In this project, we trained a model based on the dataset provided by STM.
 - **Expected behavior:** After training we will provide data only for 2 temperatures. The model will predict the values at third temperature.
 - This project is done for SARADC 12 bits but can be expanded for other IPs as well using the same concept.

Regression Algorithm

These algorithms are used to predict the continuous values. We have implemented 3 different regression algorithms on the data provided.

- Linear Regression
 - Lasso Regression
 - Ridge Regression
-
- ✓ Ridge and Lasso regression are powerful techniques generally used for creating parsimonious models in presence of a '**large**' number of features.
 - ✓ Our Dataset has large feature vector. For example, under static measurements for one sample, we are measuring values of OFFSET Error, GAIN Error, FSE, DNLP, DNLN, Max DNL_CODE, INLP, INLN etc.. Across different voltage and temperature ranges. All these columns are feature vector for our algorithm.
 - ✓ Our Regression Algorithm tries to correlate these values to predict the missing values. We can also see the correlation among different feature vector and then can drop those feature vectors which are irrelevant for our predictions.
 - ✓ So, in a nutshell, we have to decide the weight given to our feature vectors for accurate predictions. That's it!
 - ✓ The way Linear, Ridge and lasso gives weights for different feature vectors is different.

- Ridge Regression:**
 Performs L2 regularization, i.e. adds penalty equivalent to **square of the magnitude** of coefficients
 Minimization objective = $LS\ Obj + \alpha * (\text{sum of square of coefficients})$
- Lasso Regression:**
 Performs L1 regularization, i.e. adds penalty equivalent to **absolute value of the magnitude** of coefficients
 Minimization objective = $LS\ Obj + \alpha * (\text{sum of absolute value of coefficients})$
- Linear Regression**
 Minimization objective = $LS\ obj$
 The coefficients are decided by appropriate algorithm used.



The train and test score along with error for all three algorithms are reported below:

	$\alpha = 1$			$\alpha = 0.01$			$\alpha = 0.0001$		
Algorithm	Train score	Test Score	Error	Train score	Test Score	Error	Train score	Test Score	Error
Linear	1	1	2.1513e-28	--	--		--	--	
Lasso	0.1694	0.116	0.032	0.844	0.843	0.00766	0.99	0.99	7.668e-07
Ridge	0.902	0.928	0.00033	0.99	0.99	6.535e-08	0.999	0.999	6.67e-12

Thank you 😊