EARTHQUAKE PREDICTION MODEL USING PYTHON

Batch member:

- DINESH.G
- 510521104006

PHASE-II SUBMISSION DOCUMENT

INTRODUCTION:

In this article,we are going to discuss about the basic concepts of featured engineering and hyperparameter tuning and it's concepts which is used to enhance the earthquake prediction model.

HYPERPARAMETER TUNING:

HYPERPARAMETER:

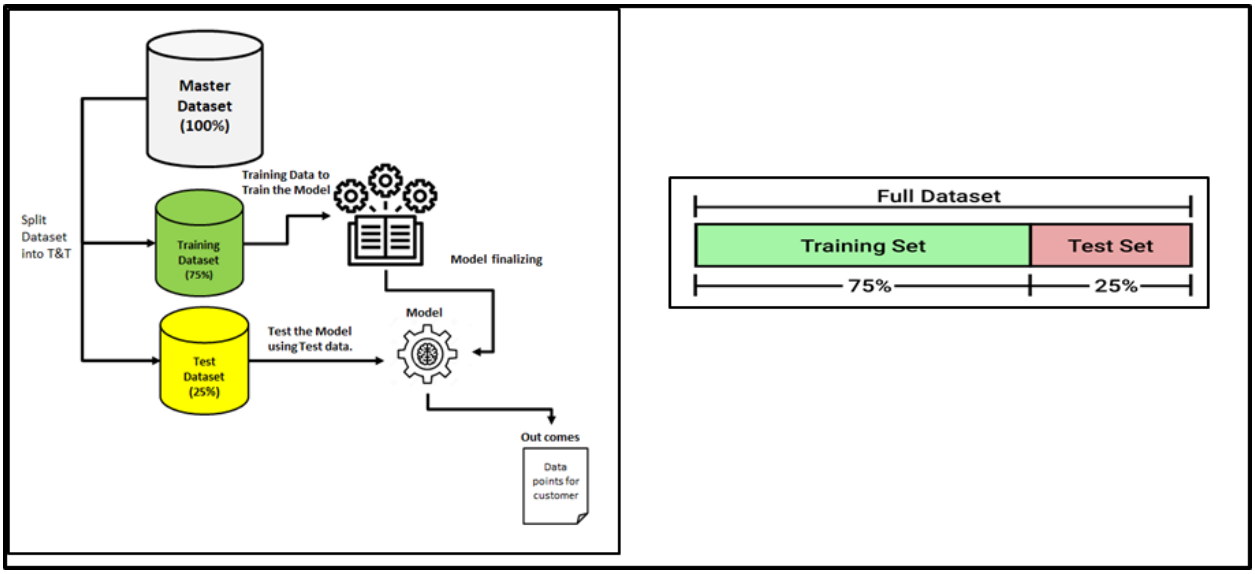
Make it simple, for every single machine learning model selection is a major exercise and it is purely dependent on selecting the equivalent set of hyperparameters, and all these are indispensable to train a model. It is always referring to the parameters of the selected model and be remember it cannot be learnt from the data, and it needs to be provided before the model gets into the training stage, ultimately the performance of the machine learning model improves with a more acceptable choice of hyperparameter tuning and selection techniques. The main intention of this article is to make you all aware of hyperparameter tuning.

Hyperparameter tuning allows data scientists to tweak model performance for optimal results. This process is an essential part of machine learning, and choosing appropriate hyperparameter values is crucial for success. For example, assume you're using the learning rate of the model as a hyperparameter.

ML LIFE CYCLE:

If you ask me what is Hyperparameters in simple words, the one-word answer is Configuration.

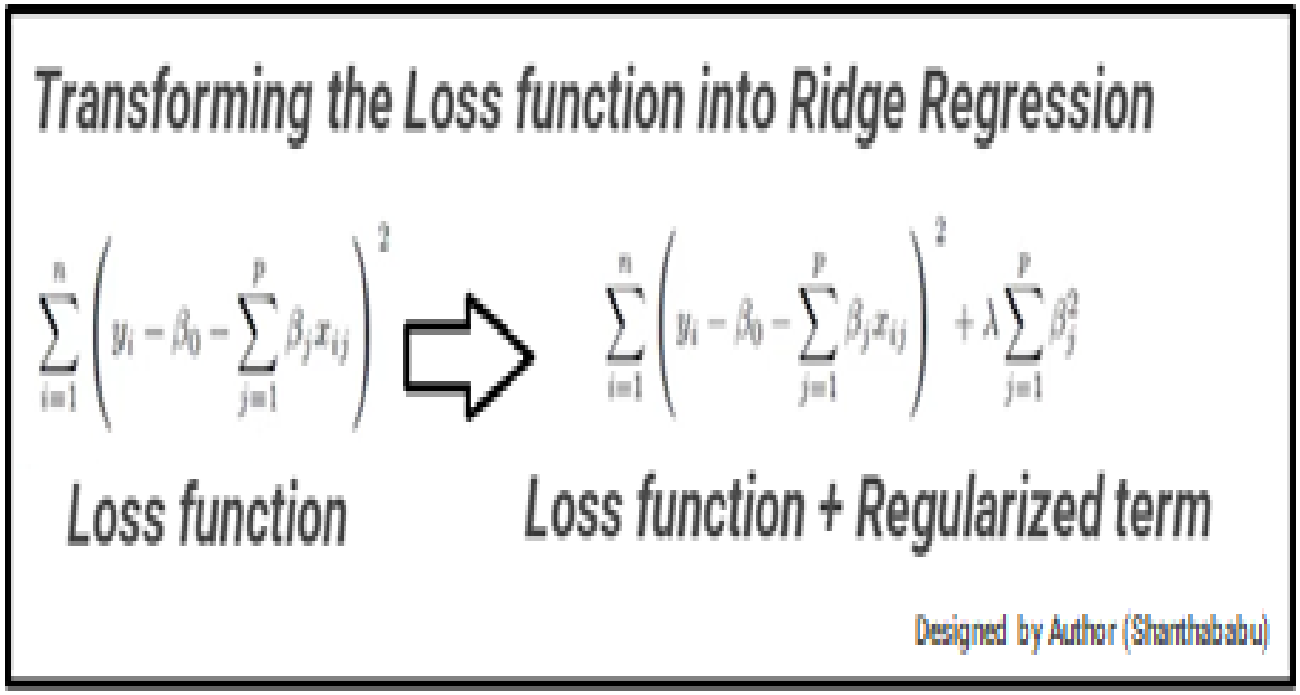
Without thinking too much, I can say quick Hyperparameter is “Train-Test Split Ratio (80-20)” in our simple linear regression model.

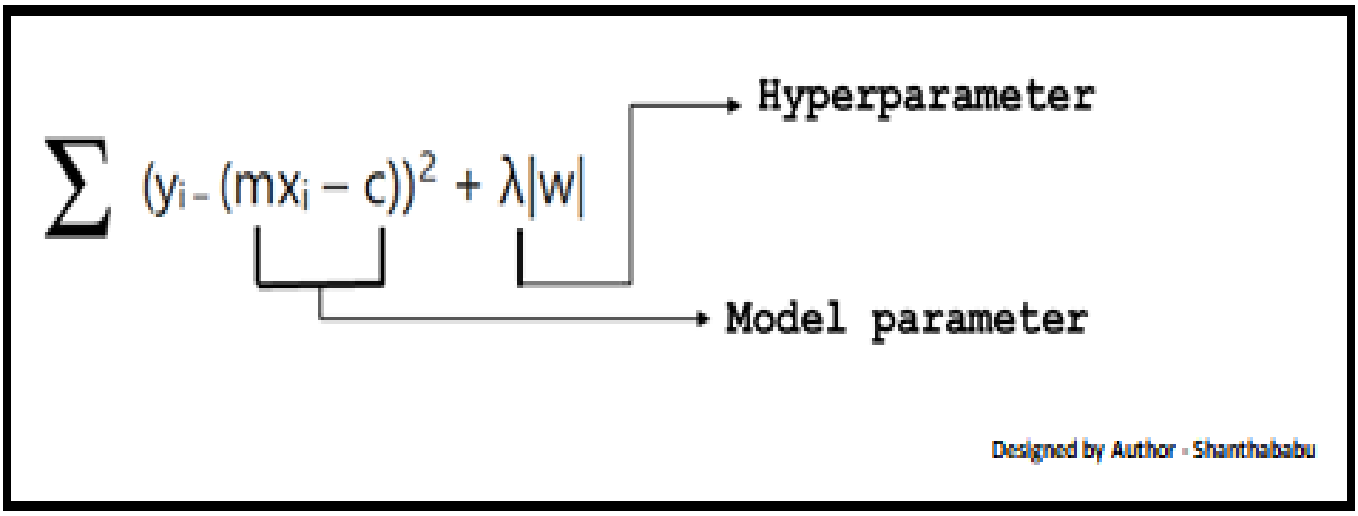


Let me give one more example – You can compare this with selecting setting the **font** and its **size** for better readability and clarity while you document your content to be perfect and precise.

Coming back to machine learning and recalling Ridge Regression (L2 Regularization) and Lasso Regression (L1 Regularization), In regularized terms we use to have lambda (λ) I mean the Penalty Factor helps us to get a smooth surface instead of an irregular graph.

This is nothing but hypermeters.

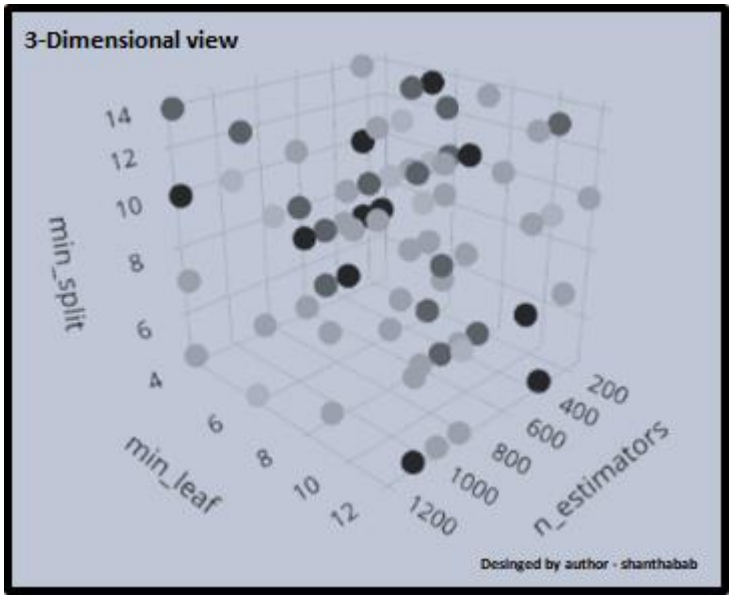




Hyperparameter Space:

As we know that there is a list of HPs for any selected algorithm(s) and our job is to figure out the best combination of HPs and to get the optimal results by tweaking them strategically, this process will be providing us with the platform for **Hyperparameter Space** and this combination leads to provide the best optimal results, no doubt in that but finding this combo is not so easy, we have to search throughout the space. Here every combination of selected HP value is said to be the "**MODEL**" and have to evaluate the same on the spot. For this reason, there are

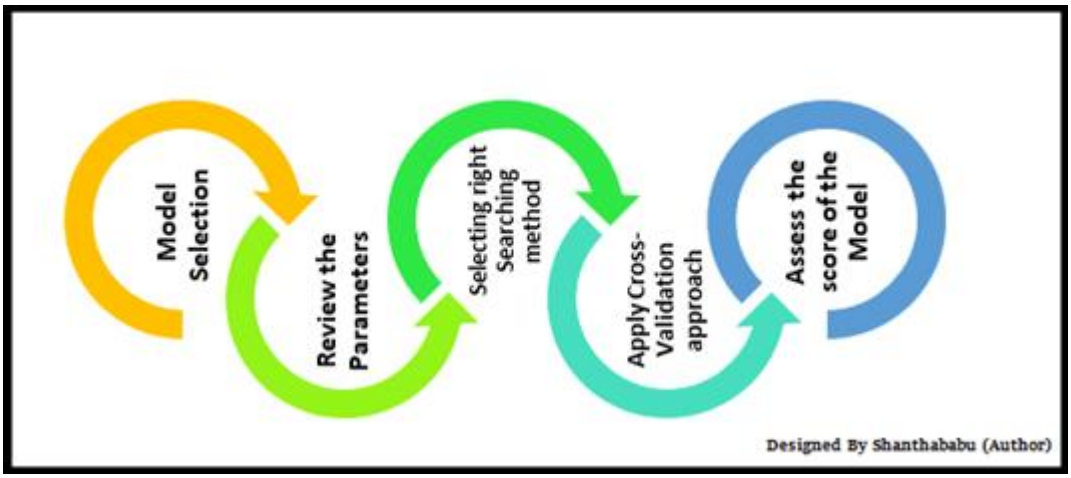
two generic approaches to search effectively in the HP space are **GridSearch CV** and **RandomSearch CV.** Here CV denotes **Cross-Validation.**



STEPS TO PERFORM HYPERPARAMETER TUNING:

- Select the right type of model.
- Review the list of parameters of the model and build the HP space
- Finding the methods for searching the hyperparameter space
- Applying the cross-validation scheme approach
- Assess the model score to evaluate the model.

Train, Test Split Estimator:



With the help of this, we use to set the test and train size for the given dataset and along with random state, this is permutations to generate the same set of splits., otherwise you will get a different set of test and train sets, tracing your model during evaluation is bit complex or if we omitted this system will generate this number and leads to unpredictable behaviour of the model. The random state provides the seed, for the random number generator, in order to stabilize the model.

```
train_test_split( X,y,test_size=0.4,random_state=0)
```

Logistic Regression Classifier:

The parameter C in Logistic Regression Classifier is directly related to the regularization parameter λ but is inversely proportional to C=1/λ.

```
LogisticRegression(C=1000.0,
random_state=0)LogisticRegression(C=1000.0,ran
dom_state=0)
```

KNN (k-Nearest Neighbors) Classifier:

As we know the k-nearest neighbour's algorithm (KNN) is a non-parametric method used for regression and classification problems. Predominantly this is used for

classification problems, in which the number of neighbours and power parameter

```
KNeighborsClassifier(n_neighbors=5,p=2,metric='
minkowski')
– n_neighbors is the number of neighbors
– p is Minkowski (the power parameter)
If p = 1 Equivalent to manhattan_distance,
p = 2. For Euclidean_distance
```

Support Vector Machine Classifier:

```
SVC(kernel='linear', C=1.0, random_state=0)
– kernel specifies the kernel type to be used in
the chosen algorithm,
kernel = 'linear', for Linear Classification
kernel = 'rbf' for Non-Linear Classification.
C is the penalty parameter (error)
random_state is a pseudo-random number
generator
```

Decision Tree Classifier:

Here, the criterion is the function to measure the quality of a split, max_depth is the maximum depth of the tree, and

random_state is the seed used by the random number generator.

DecisionTreeClassifier(criterion=’entropy’, max_depth=3, random_state=0)

Lasso Regression:

Lasso(alpha = 0.1) the regularization parameter is alpha.

Principal Component Analysis:

PCA(n_components = 4)

Perceptron Classifier:

Perceptron (n_iter=40, eta0=0.1, random_state=0)

n-iter-is the number of iterations,

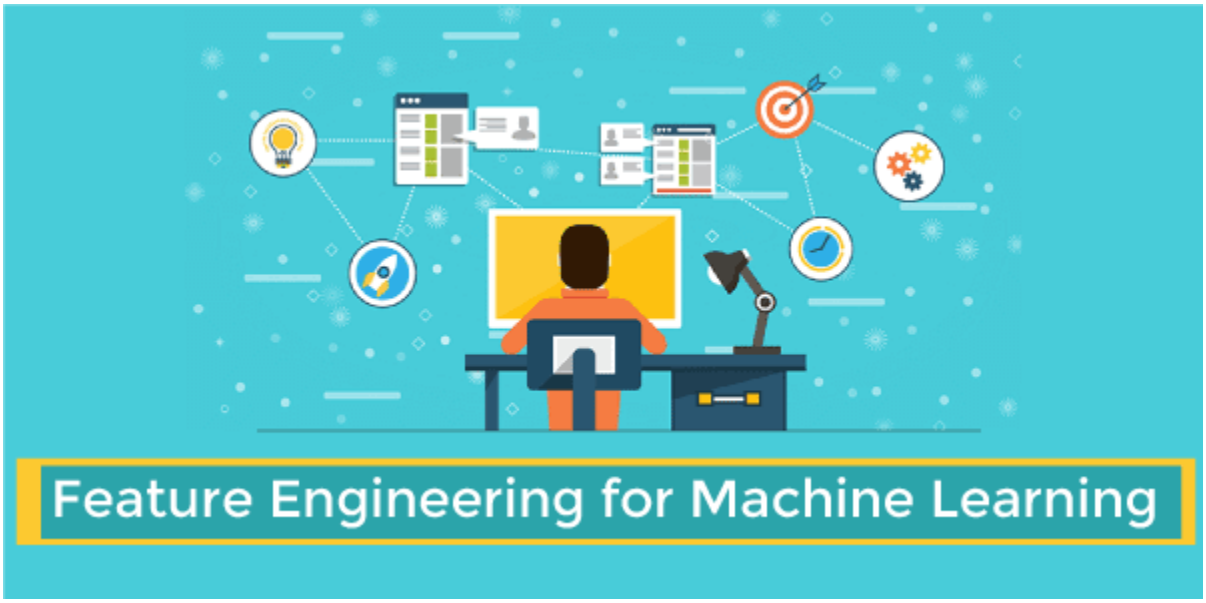
-eta0 is the learning rate,

-random_state is random number generator.

FEATURED ENGINEERING:

Feature Engineering for Machine Learning:

Feature engineering is the pre-processing step of machine learning, which is used to transform raw data into features that can be used for creating a predictive model using Machine learning or statistical Modelling. Feature engineering in machine learning aims to improve the performance of models. In this topic, we will understand the details about feature engineering in Machine learning.



What is a feature?

Generally, all machine learning algorithms take input data to generate the output. The input data remains in a tabular form consisting of rows (instances or observations) and columns (variable or attributes), and these attributes are often known as features. For example, an image is an instance in computer vision, but a line in the image could be the feature.

Feature engineering is the pre-processing step of machine learning, which extracts features from raw data. It helps to represent an underlying problem to predictive models in a better way, which as a result, improve the accuracy of the model for unseen data. The predictive model contains predictor variables and an outcome variable, and while the feature engineering process selects the most useful predictor variables for the model.

PROCESSES:

1.Feature Creation: Feature creation is finding the most useful variables to be used in a predictive model. The process is subjective, and it requires human creativity and intervention. The new features are created by mixing existing features using addition, subtraction, and ration, and these new features have great flexibility.

2.Transformations:

The transformation step of feature engineering involves adjusting the predictor variable to improve the accuracy and performance of the

model. For example, it ensures that the model is flexible to take input of the variety of data; it ensures that all the variables are on the same scale, making the model easier to understand. It improves the model's accuracy and ensures that all the features are within the acceptable range to avoid any computational error.

3.Feature Extraction:

Feature extraction is an automated feature engineering process that generates new variables by extracting them from the raw data. The main aim of this step is to reduce the volume of data so that it can be easily used and managed for data modelling. Feature extraction methods include **cluster analysis, text analytics, edge detection algorithms, and principal components analysis (PCA**).

4.Feature Selection:

While developing the machine learning model, only a few variables in the dataset are useful for building the model, and the rest features are either redundant or irrelevant.

BENEFITS:

- It helps in avoiding the curse of dimensionality.
- It helps in the simplification of the model so that the researchers can easily interpret it.
- It reduces the training time.
- It reduces overfitting hence enhancing the generalization.

STEPS IN FEATURE ENGINEERING:

1.Imputation:

Feature engineering deals with inappropriate data, missing values, human interruption, general errors, insufficient data sources, etc. Missing values within the dataset highly affect the performance of the algorithm, and to deal with them "Imputation" technique is used. Imputation is responsible for handling irregularities within the dataset.

For example, removing the missing values from the complete row or complete column by a huge percentage of missing values.

For numerical data imputation, a default value can be imputed in a column, and missing values can be filled with means or medians of the columns.

For categorical data imputation, missing values can be interchanged with the maximum occurred value in a column

2.Handling Outliers

Outliers are the deviated values or data points that are observed too away from other data points in such a way that they badly affect the performance of the model. Outliers can be handled with this feature engineering technique. This technique first identifies the outliers and then remove them out.

Standard deviation can be used to identify the outliers. For example, each value within a space has a definite to an average distance, but if a value is greater distant than a certain value, it can be considered as an outlier. Z-score can also be used to detect outliers.

3. Log transform:

Logarithm transformation or log transform is one of the commonly used mathematical techniques in machine learning. Log transform helps in handling the skewed data, and it makes the distribution more approximate to normal after transformation. It also reduces the effects of outliers on the data, as

because of the normalization of magnitude differences, a model becomes much robust.

4.binning:

In machine learning, overfitting is one of the main issues that degrade the performance of the model and which occurs due to a greater number of parameters and noisy data. However, one of the popular techniques of feature engineering, "binning", can be used to normalize the noisy data. This process involves segmenting different features into bins.

5.head splitting:

As the name suggests, feature split is the process of splitting features intimately into two or more parts and performing to make new features. This technique helps the algorithms to better understand and learn the patterns in the dataset.

The feature splitting process enables the new features to be clustered and binned, which results in extracting useful information and improving the performance of the data models.

6.0ne hot heading:

One hot encoding is the popular encoding technique in machine learning. It is a technique that converts the categorical data in a form so that they can be easily understood by machine learning algorithms and hence can make a good prediction. It enables group the of categorical data without losing any information.

CONCLUSION:

The basic concepts such as

- Featured engineering
- Hyperparameter tuning

and it’s concepts are explained in above mentioned documents.