# PHASE 4

**DEVELOPMENT PART 2:**

# As we know Development part 1 deals with Data preprocessing like importing the libraries, cleaning the data set, filling the missing values and encoding the categorical data. This part(development part 2) deals with

* **Model Training**

# Feature Engineering

* **Evaluation MODEL TRAINING:**

Machine Learning is one of the booming technologies across the world that enables computers/machines to turn a huge amount of data into predictions. However, these predictions highly depend on the quality of the data, and if we are not using the right data for our model, then it will not generate the expected result. In machine learning projects, we generally divide the original dataset into training data and test data. We train our model over a subset of the original dataset, i.e., the training dataset, and then evaluate

**whether it can generalize well to the new or unseen dataset or test set. Therefore, train and test datasets are the two key concepts of machine learning, where the training dataset is used to fit the model, and the test dataset is used to evaluate the model.**

# FEATURE ENGINEERING:

Feature Engineering is also known as Feature Scaling. Feature engineering is the process of transforming raw data into features that are suitable for machine learning models. In other words, it is the process of selecting, extracting, and transforming the most relevant features from the available data to build more accurate and efficient machine learning models.

# EVALUATION:

Model evaluation is the process that uses some metrics which help us to analyze the performance of the model. As we all know that model development is a multi-step process and a check should be kept on how well the model generalizes future predictions. Therefore evaluating a model plays a vital role so that we can judge the performance of our model. The evaluation also helps to analyze a model’s key weaknesses. There are many

metrics like Accuracy, Precision, Recall, Area under curve,F1 score, Confusion Matrix, and Mean Square Error. Cross Validation is one technique that is followed during the training phase and it is a model evaluation technique as well.

# DATASET LINK:

# https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dataset

**CODE FOR DEVELOPMENT PART 2:**

# MODEL TRAINING:

**Splitting the dataset into the Training set and Test set:**

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 1)

print(X\_train) print(X\_test) print(y\_train) print(y\_test)

Feature Engineering

from sklearn.preprocessing import StandardScaler sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train) X\_test = sc.transform(X\_test) print(X\_train)

print(X\_test)

# Evaluation:

**Accuracy:**

from sklearn.metrics import precision\_score, recall\_score, f1\_score, accuracy\_score

from sklearn.tree import DecisionTreeClassifier tree = DecisionTreeClassifier()

tree.fit(X\_train, y\_train) y\_pred = tree.predict(X\_test)

print(“Accuracy:”,accuracy\_score(y\_test,y\_pred))

# Precision and Recall:

print( "Precision:" , precision\_score(y\_test, y\_pred,average="weighted"))

print('Recall:', recall\_score(y\_test,y\_pred,average="weighted"))

# F1 Score:

print('F1 score:', f1\_score(y\_test, y\_pred,average= "weighted" ))

# Confusion Matrix:

confusion\_matrix = metrics.confusion\_matrix(y\_test,y\_pred)

cm\_display=metrics.ConfusionMatrixDisplay(confusion\_matrix=confu sion\_matrix,display\_labels=[0, 1, 2])

cm\_display.plot() plt.show()

# Mean Square Error:

fromsklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error, mean \_squared error

mse = mean\_squared\_error(y\_true=Y\_test,y\_pred=Y\_pred) print( "Mean Square Error" , mse)

# Area under curve:

**import numpy as np**

**from sklearn .metrics import roc\_auc\_score**

**y\_true = [1, 0, 0, 1]**

**y\_pred = [1, 0, 0.9, 0.2]**

**auc = np.round(roc\_auc\_score(y\_true, y\_pred), 3)**

**print("Auc", (auc))**