

```
In [1]: #LIBRARIES
import pandas as pd
import numpy as np
```

```
In [2]: # TO PLOT A GRAPHS
import matplotlib.pyplot as plt
```

```
In [3]: # ADD&REDUCE
from operator import add
from functools import reduce
```

```
In [4]: #MISSING NO
import missingno as msno
```

```
In [5]: # NORMALIZATION&STANDARDIZATION
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
```

```
In [6]: # LABEL_ENCODER
from sklearn.preprocessing import LabelEncoder
```

```
In [7]: #SIMPLE LINEAR REGRESSION&MULTI LINEAR REGRESSION
from sklearn.linear_model import LinearRegression
from sklearn import linear_model
```

```
In [8]: # SPLITTING THE DATASET
from sklearn.model_selection import train_test_split
```

```
In [9]: # TO PRINT RMSE[ROOT MEAN SQUARED ERROR]
from sklearn.metrics import mean_squared_error
```

```
In [10]: # LOGISTIC REGRESSION
from sklearn.linear_model import LogisticRegression
```

```
In [11]: # CONFUSION MATRIX
from sklearn.metrics import confusion_matrix
```

```
In [15]: # DECISIONTREE
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import export_graphviz
import graphviz
from sklearn.metrics import classification_report, f1_score
```

```
In [16]: # K-NEAREST NEIGHBOR
from sklearn.neighbors import KNeighborsClassifier
```

```
In [17]: # SVM[SUPPORT VECTOR MACHINE]
import seaborn as sns
```

```
import seaborn as sns
#SVC[SUPPORT VECTOR CLASSIFIER]
from sklearn.svm import SVC
```

In [18]:

```
#RANDOM FOREST
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, f1_score
```

In []:

```
#CNN[CONVOLUTION NEUTRAL NETWORK]
import tensorflow as tf
from keras.model import sequential
from keras.layers import Dense, Activation, Dropout
```