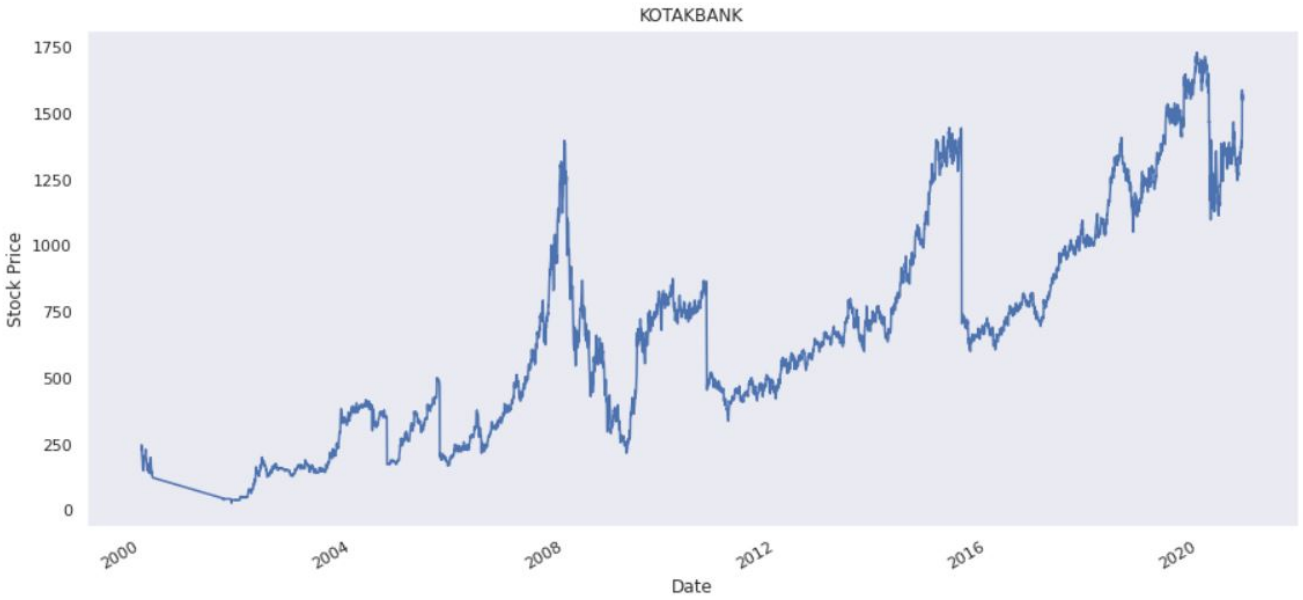
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df=pd.read_csv("/content/KOTAKBANK.csv",index_col=[0])
df.index=pd.to_datetime(df.index)
df.head()
```

Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverable Volume	%Deliverble
2000-01-03	KOTAKMAH	EQ	212.35	220.0	229.35	220.00	229.35	229.35	229.13	7086	1.623640e+11	NaN	NaN	NaN
2000-01-04	KOTAKMAH	EQ	229.35	247.7	247.70	225.25	247.70	246.95	244.12	73681	1.798729e+12	NaN	NaN	NaN
2000-01-05	KOTAKMAH	EQ	246.95	229.0	240.00	227.20	228.00	228.40	233.75	105799	2.473093e+12	NaN	NaN	NaN
2000-01-06	KOTAKMAH	EQ	228.40	235.1	239.00	217.00	224.95	225.90	226.84	40202	9.119546e+11	NaN	NaN	NaN
2000-01-07	KOTAKMAH	EQ	225.90	213.0	219.00	207.85	207.85	208.85	209.94	24463	5.135747e+11	NaN	NaN	NaN

```
[274] df=df.drop(['Symbol','Series','Prev Close','VWAP','Turnover','Trades','Deliverable Volume','%Deliverble'],axis=1)
      df.head()
```

Date	Open	High	Low	Last	Close	Volume
2000-01-03	220.0	229.35	220.00	229.35	229.35	7086
2000-01-04	247.7	247.70	225.25	247.70	246.95	73681
2000-01-05	229.0	240.00	227.20	228.00	228.40	105799

```
[276] df.Close.plot(figsize=(15,7))
      plt.xlabel('Date')
      plt.ylabel('Stock Price')
      plt.title('KOTAKBANK')
      plt.grid()
      plt.show()
```



```
[278] df.Close.plot(figsize=(15,7))
      plt.xlabel('Date')
      plt.ylabel('Stock Price')
      plt.title('KOTAKBANK')
      plt.grid()
      plt.show()
```

```
import talib as ta
df['RSI']=ta.RSI(df['Close'].values,timeperiod=14)
df['DIFF1']=df['Close'].diff().values
df['DIFF2']=df['Close'].diff(2).values
df['DIFF3']=df['Close'].diff(3).values
df['DIFF4']=df['Close'].diff(4).values
print(df.head())
df.tail()
```

```
             Open    High    Low   Last  ...  DIFF1  DIFF2  DIFF3  DIFF4
Date                                     ...
2004-01-01  399.0  399.00  383.0  391.9  ...    NaN    NaN    NaN    NaN
2004-01-02  397.5  397.50  379.0  383.0  ...  -6.80    NaN    NaN    NaN
2004-01-05  388.0  390.90  370.0  373.0  ... -12.35 -19.15    NaN    NaN
2004-01-06  375.8  375.80  363.0  363.0  ...  -4.10 -16.45 -23.25    NaN
2004-01-07  350.0  376.95  350.0  370.1  ...   2.65  -1.45 -13.80  -20.6

[5 rows x 11 columns]
```

Date	Open	High	Low	Last	Close	Volume	RSI	DIFF1	DIFF2	DIFF3	DIFF4
2020-10-26	1399.0	1429.00	1347.35	1410.55	1416.90	8022562	64.565282	33.85	23.85	23.60	48.10
2020-10-27	1452.5	1602.00	1452.50	1582.70	1587.65	33167143	79.086428	170.75	204.60	194.60	194.35
2020-10-28	1575.0	1575.00	1541.00	1556.00	1551.35	7245009	72.302863	-36.30	134.45	168.30	158.30
2020-10-29	1532.0	1588.80	1525.00	1569.00	1570.90	7286144	73.615457	19.55	-16.75	154.00	187.85
2020-10-30	1574.0	1574.65	1518.60	1545.00	1547.40	4201984	69.360330	-23.50	-3.95	-40.25	130.50

```
[245] df['Trends']=np.where(df.Close.shift(-1)>df.Close,1,0)
      print(df.head(15))
      df.tail(15)
```

```
             Open    High     Low    Last  ...  DIFF2  DIFF3  DIFF4  Trends
Date                                      ...
2004-01-01  399.0  399.00  383.00  391.9  ...    NaN    NaN    NaN       0
2004-01-02  397.5  397.50  379.00  383.0  ...    NaN    NaN    NaN       0
2004-01-05  388.0  390.90  370.00  373.0  ... -19.15    NaN    NaN       0
2004-01-06  375.8  375.80  363.00  363.0  ... -16.45 -23.25    NaN       1
2004-01-07  350.0  376.95  350.00  370.1  ...  -1.45 -13.80 -20.60       1
2004-01-08  377.9  388.00  374.00  388.0  ...  15.90  11.80  -0.55       1
2004-01-09  386.0  402.00  386.00  393.5  ...  20.75  23.40  19.30       0
2004-01-12  388.3  389.00  378.00  382.5  ...  -1.65  11.60  14.25       1
2004-01-13  390.0  390.00  379.80  385.0  ...  -6.20   1.30  14.55       0
2004-01-14  384.5  389.95  384.50  386.0  ...   2.60  -6.55   0.95       1
2004-01-15  395.0  397.50  385.00  391.5  ...   4.95   7.90  -1.25       0
2004-01-16  390.0  395.00  375.05  382.0  ...  -3.20  -3.55  -0.60       0
2004-01-19  378.0  384.80  375.10  378.0  ... -12.85  -7.55  -7.90       1
2004-01-20  385.0  392.00  383.50  391.0  ...   8.25  -0.25   5.05       0
2004-01-21  383.2  390.00  370.00  380.5  ...  -0.80  -5.15 -13.65       0

[15 rows x 12 columns]
```

Date	Open	High	Low	Last	Close	Volume	RSI	DIFF1	DIFF2	DIFF3	DIFF4	Trends
2020-10-12	1328.00	1358.20	1307.50	1312.85	1312.90	5175329	49.522672	-6.95	-7.20	-11.80	-25.35	1
2020-10-13	1319.65	1349.70	1307.20	1341.10	1344.95	4872154	56.427840	32.05	25.10	24.85	20.25	1
2020-10-14	1335.25	1358.00	1324.15	1355.00	1353.35	3358472	58.047671	8.40	40.45	33.50	33.25	0
2020-10-15	1353.35	1365.95	1301.25	1307.00	1309.55	3496590	48.022629	-43.80	-35.40	-3.35	-10.30	1

```
[247] df.dropna(inplace=True)
      predictor_list=['RSI','DIFF1','DIFF2','DIFF3','DIFF4']
      X=df[predictor_list]
      y=df['Trends']
      y.tail()
```

```
Date
2020-10-26    1
2020-10-27    0
2020-10-28    1
2020-10-29    0
2020-10-30    0
Name: Trends, dtype: int64
```

```
[248] print(df['Trends'].count())
      df['Trends'].sum()
```

```
4167
2168
```

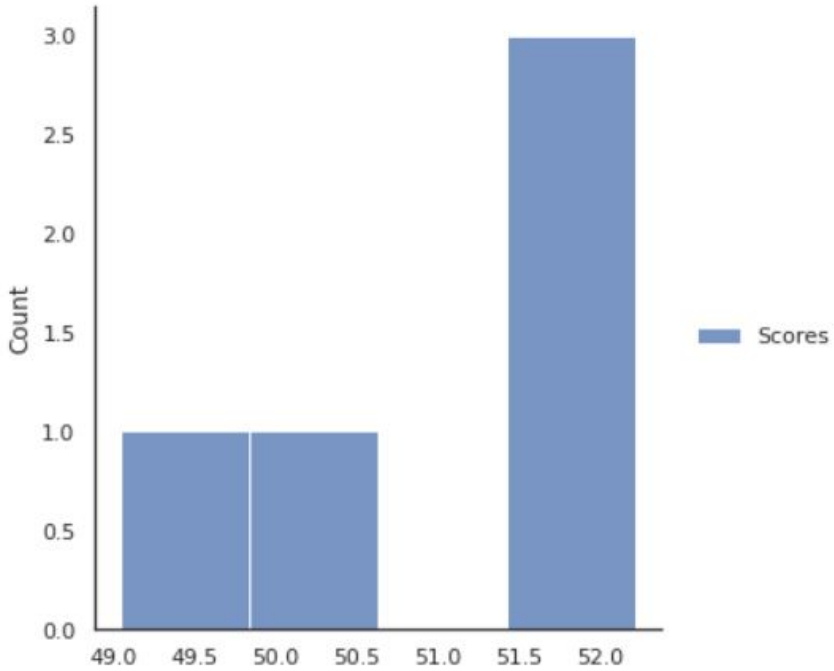
```
[249] from sklearn.model_selection import train_test_split
      X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42,stratify=y)
      print(X_train.shape,X_test.shape)
      print(y_train.shape,y_test.shape)
```

```
(3333, 5) (834, 5)
(3333,) (834,)
```

```
[254] print(accuracy_model)
```

```
[51.91846522781775, 49.040767386091126, 51.50060024009604, 52.22088835534213, 50.540216086434576]
```

```
[264] scores = pd.DataFrame(accuracy_model,columns=['Scores'])
      sns.set(style="white", rc={"lines.linewidth": 3})
      sns.displot(scores)
      plt.show()
      sns.set()
```




```
[256] from sklearn.metrics import confusion_matrix
```

```
[256] from sklearn.metrics import confusion_matrix
      from sklearn.metrics import accuracy_score
      cm = confusion_matrix(y_test, y_pred)
      print("CONFUSION MATRIX \n",cm)
```

```
CONFUSION MATRIX
 [[155 245]
 [167 266]]
```

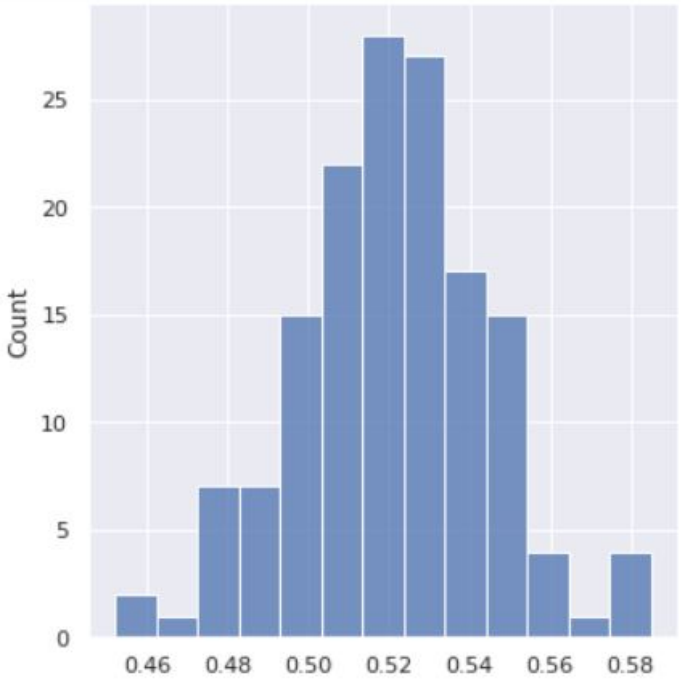
```
[257] from sklearn.metrics import classification_report
      y_pred=model.predict(X_test)
      report=classification_report(y_test,y_pred)
      print(report)
```

```
              precision    recall  f1-score   support

           0       0.48      0.39      0.43       400
           1       0.52      0.61      0.56       433

    accuracy                           0.51       833
   macro avg       0.50      0.50      0.50       833
weighted avg       0.50      0.51      0.50       833
```

[269]



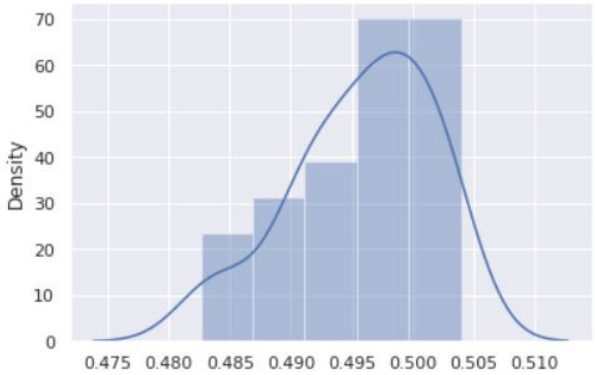
[262]
```
y_pred_logistic= model.predict(X_test)
cm = confusion_matrix(y_test, y_pred_logistic)
print(cm)
report=classification_report(y_test,y_pred_logistic)
print(report)
```

```
[[155 245]
 [167 266]]
              precision    recall  f1-score   support

           0       0.48      0.39      0.43       400
           1       0.52      0.61      0.56       433
```

[270]
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
%matplotlib inline
k_range = range(1, 31)
k_scores = []
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score(knn, X, y, cv=5, scoring='accuracy')
    k_scores.append(scores.mean())
sns.distplot(k_scores)
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b81fd57f0>
```



```
[271] y_pred_knn= classifier_knearest.predict(X_test)
      cm = confusion_matrix(y_test, y_pred_knn)
      print(cm)
      report=classification_report(y_test,y_pred_knn)
      print(report)
```

```
[[206 194]
 [217 216]]
              precision    recall  f1-score   support

           0       0.49      0.52      0.50       400
           1       0.53      0.50      0.51       433

    accuracy                           0.51       833
   macro avg       0.51      0.51      0.51       833
weighted avg       0.51      0.51      0.51       833
```