```
# Importing Libaries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import os
import math
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import RandomizedSearchCV
from sklearn.metrics import r2_score,mean_squared_error as mse , mean_absolute_erro
from sklearn.ensemble import RandomForestRegressor
from math import sqrt
from sklearn.metrics import mean_absolute_error

import warnings
warnings.filterwarnings("ignore")
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
path = "/content/Twitter_stock_final_dataset (1).csv"
df = pd.read_csv(path)
df
```

	Year	Month	Day	StockName	Positive	Negative	Neutral	Total Tweets	Close
0	2020	1	1	apple	10	2	8	20	75.0875
1	2020	1	1	microsoft	9	0	11	20	160.6200
2	2020	1	1	tesla	17	3	3	23	86.0520
3	2020	1	1	nvidia	1	0	0	1	59.9775
4	2020	1	1	paypal	1	0	1	2	110.7500
...	...	...	...	...	...	...	...	...	...
2978	2021	9	20	tesla	61	21	39	121	730.1700
2979	2021	9	20	nvidia	3	4	3	10	211.1300
2980	2021	9	20	paypal	1	1	2	4	269.9100
2981	2021	9	21	nvidia	4	4	1	9	212.4600
2982	2021	9	21	paypal	3	3	2	8	269.4900

! 0s completed at 11:56 AM ● ×

0	2020	1	1	apple	10	2	8	20	75.0875
1	2020	1	1	microsoft	9	0	11	20	160.6200
2	2020	1	1	tesla	17	3	3	23	86.0520
3	2020	1	1	nvidia	1	0	0	1	59.9775
4	2020	1	1	paypal	1	0	1	2	110.7500
...	...	...	...	...	...	...	...	...	...
2978	2021	9	20	tesla	61	21	39	121	730.1700
2979	2021	9	20	nvidia	3	4	3	10	211.1300
2980	2021	9	20	paypal	1	1	2	4	269.9100
2981	2021	9	21	nvidia	4	4	1	9	212.4600
2982	2021	9	21	paypal	3	3	2	8	269.4900

2983 rows × 15 columns

```
df.index = df['Date']
df
```

	Year	Month	Day	StockName	Positive	Negative	Neutral	Total Tweets	C:
Date									
2020-01-01	2020	1	1	apple	10	2	8	20	75.
2020-01-01	2020	1	1	microsoft	9	0	11	20	160.
2020-01-01	2020	1	1	tesla	17	3	3	23	86.
2020-01-01	2020	1	1	nvidia	1	0	0	1	59.
2020-01-01	2020	1	1	paypal	1	0	1	2	110.
...	...	...	...	...	...	...	...	...	
2021-09-20	2021	9	20	tesla	61	21	39	121	730.
2021-09-20	2021	9	20	nvidia	3	4	3	10	211.
2021-09-20	2021	9	20	paypal	1	1	2	4	269.

```
df['StockName'] = le.fit_transform(df["StockName"])
df['Day_of_week']= le1.fit_transform(df["Day_of_week"])
df['Year'] = le2.fit_transform(df["Year"])
df.head(10)
```

	Year	Month	Day	StockName	Positive	Negative	Neutral	Total Tweets	C
Date									
2020-01-01	0	1	1	0	10	2	8	20	75.
2020-01-01	0	1	1	1	9	0	11	20	160.
2020-01-01	0	1	1	4	17	3	3	23	86.
2020-01-01	0	1	1	2	1	0	0	1	59.
2020-01-01	0	1	1	3	1	0	1	2	110.
2020-01-02	0	1	2	0	42	11	31	84	75.
2020-01-02	0	1	2	1	8	1	7	16	160.
2020-01-02	0	1	2	4	30	3	21	54	86.
2020-01-02	0	1	2	2	2	0	2	4	59.
2020-01-02	0	1	2	3	0	0	2	2	110.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2983 entries, 2020-01-01 to 2021-09-21
Data columns (total 15 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Year          2983 non-null   int64
 1   Month         2983 non-null   int64
 2   Day           2983 non-null   int64
 3   StockName     2983 non-null   int64
 4   Positive      2983 non-null   int64
```

Date									
2020-01-01	0	1	1	0	10	2	8	20	75.
2020-01-01	0	1	1	1	9	0	11	20	160.
2020-01-01	0	1	1	4	17	3	3	23	86.
2020-01-01	0	1	1	2	1	0	0	1	59.
2020-01-01	0	1	1	3	1	0	1	2	110.
...	...	...	...	...	...	...	...	...	
2021-09-20	1	9	20	4	61	21	39	121	730.
2021-09-20	1	9	20	2	3	4	3	10	211.
2021-09-20	1	9	20	3	1	1	2	4	269.
2021-09-21	1	9	21	2	4	4	1	9	212.
2021-09-21	1	9	21	3	3	3	2	8	269.

2983 rows × 14 columns

Dividing the dependent and independent columns

```
X = np.array(df.drop(['Close'], axis = 1))
y = np.array(df['Close'])
```

Linear Regression

```
from sklearn.linear_model import LinearRegression
```

```
print("MAE:",mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)
```

```
Results of sklearn.metrics:
MAE: 1.9247603539326705
MSE: 9.758769327640998
RMSE: 3.1239028998419585
R-Squared: 0.999736957318314
```

Random forest Regression

```
from sklearn.ensemble import RandomForestRegressor
from sklearn import metrics
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import TimeSeriesSplit
import math
tscv = TimeSeriesSplit()


for train_index, test_index in tscv.split(X):
    X_train, X_test = X[train_index], X[test_index]
```

```
from sklearn.model_selection import TimeSeriesSplit
import math
from math import sqrt


tscv = TimeSeriesSplit()
for train_index, test_index in tscv.split(X):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    rf = RandomForestRegressor()
    params={'max_depth': [10,20,30,50,70,100,150,200],
    'min_samples_split':[5, 10,15,20,50,100],
    'criterion':['mae','mse'],
    'n_estimators':[20,50,100,150,200,500,1000]}
    cross_val = RandomizedSearchCV(estimator=rf, param_distributions=params, n_iter
    cross_val.fit(X_train,y_train)
print('='*100)
print('The Best Parameters are : ',cross_val.best_params_)
```

```
[Parallel(n_jobs=-1)]: Done  33 tasks      | elapsed:  4.8min
[Parallel(n_jobs=-1)]: Done  42 tasks      | elapsed:  5.7min
[Parallel(n_jobs=-1)]: Done  50 out of  50 | elapsed:  6.1min finished

Fitting 5 folds for each of 10 candidates, totalling 50 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done   5 tasks      | elapsed:    1.3s
[Parallel(n_jobs=-1)]: Done  10 tasks      | elapsed:    3.5s
[Parallel(n_jobs=-1)]: Done  17 tasks      | elapsed:   28.1s
[Parallel(n_jobs=-1)]: Done  24 tasks      | elapsed:  1.2min
[Parallel(n_jobs=-1)]: Done  33 tasks      | elapsed:  1.3min
[Parallel(n_jobs=-1)]: Done  42 tasks      | elapsed:  1.8min
```

```
The R2_score is =0.9989206024249903
 The RMSE is 2.5155767188767832
```