Prediction in Market Volatility

A case study in predicting market volatility and building short-term trading strategies using data from Reddit's WallStreetBets.

Bidhya Pokharel

September 2021

CONTENT

- **Project** *Approach*
- *What does the* ***DATA*** *tell us?*
- *Our* ***PREDICTION*** *models*
- ***PERFORMANCE*** *evaluate*
- ***CONCLUSION*** *&* ***NEXT STEP***

Helps to make a prediction on stock prices and market volatility.

SCENARIO

The aim of this project is to use data from posts made made on the sub-reddit “Wallstreet-Bets” to make a prediction of given scenario.

Help to predict if specific stocks rose or fell in the given time frame.

SCENARIO

Covers two datasets:

JSON file:

- *Contains comment of Reddit’s post.*
- *Performed Sentiment Analysis.*

Excel file:

- *Trimmed this huge org. provided data as per the other similar file hosted on Kaggle.*

Predict Market Volatility, why?

How can predicting market volatility add values to business world. Current scenarios' relation between stock market and social media.

Sudden market volatility increment affects the investment so predicting ***Market volatility in advance can increase /lead us to profits in*** *Stock market.*

80% of investors today use it as their regular Workflow &

Approx. 30% *obtain information about the investment market through different*

Social Media (it).

Economic Times Research

Target Variable

- Created comparing today’s close price and yesterday’s closing price.
- Check how the ***sentiment analysis*** of comment made in the day affects the closing price.

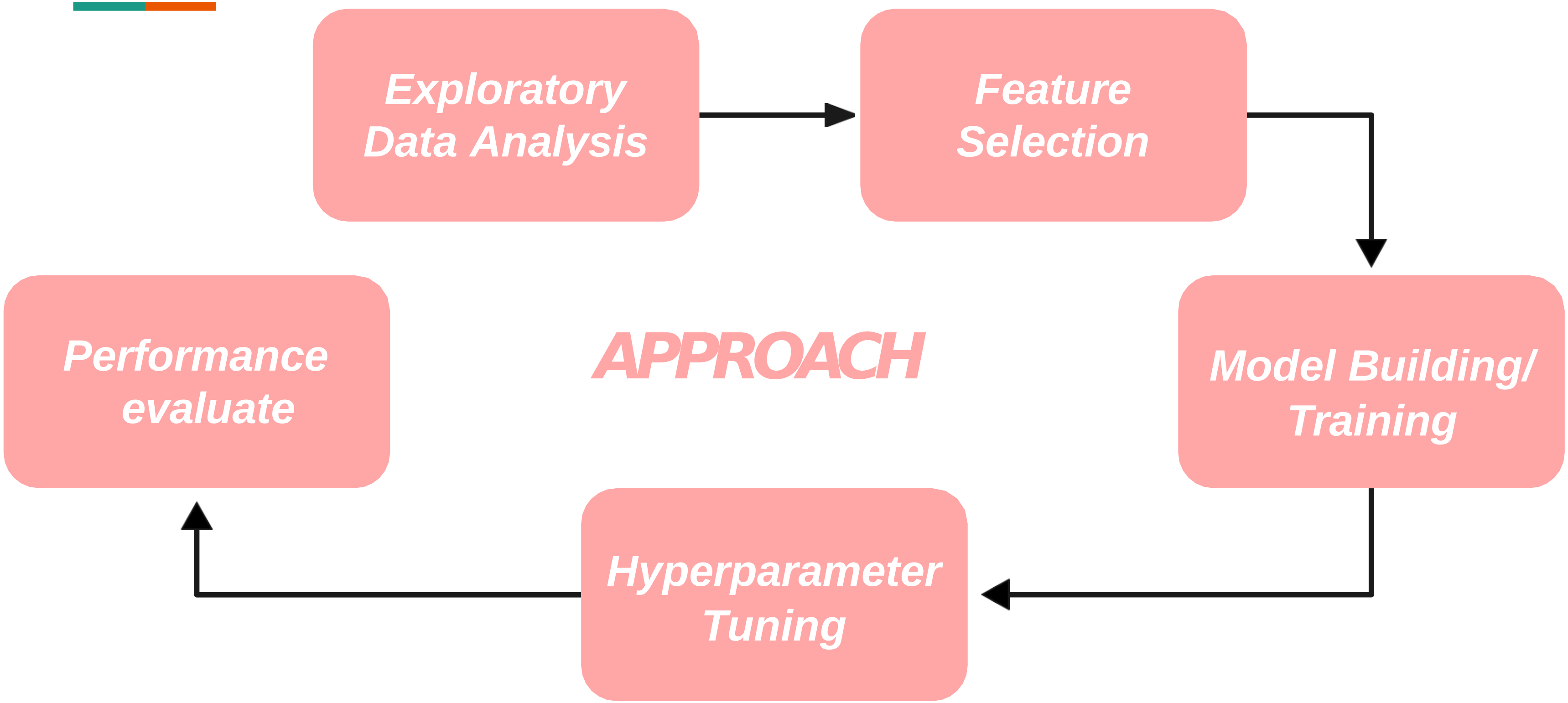
Why Data Science?

If we can Predict the future profit/ loss, we can AVOID the market volatility and get maximum profit from current stocks.

APPROACH

How data science helps to predict the market volatility, and how we are going to do with it.

APPROACH

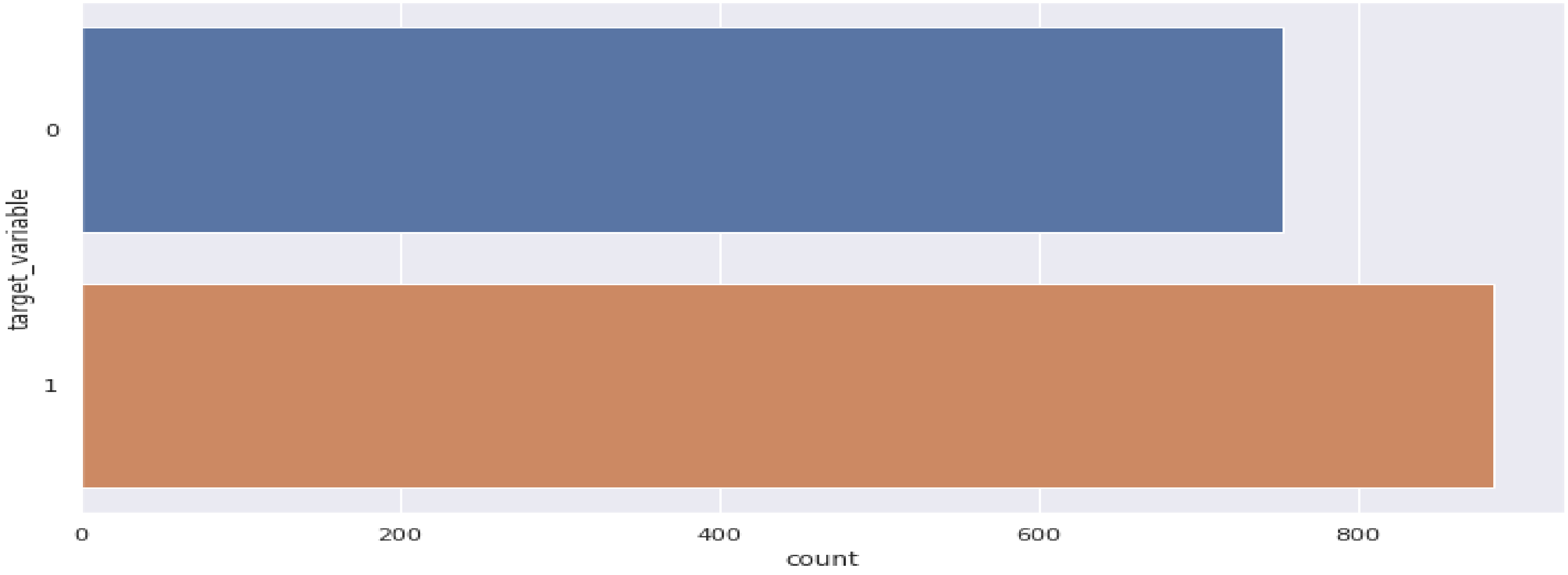

Exploratory Data Analysis
Feature Selection
Model Building/ Training
Hyperparameter Tuning
Performance evaluate

Data Analysis

What the data told us? Let go for an EDA on the data set.

Our Target Variable (P/L)

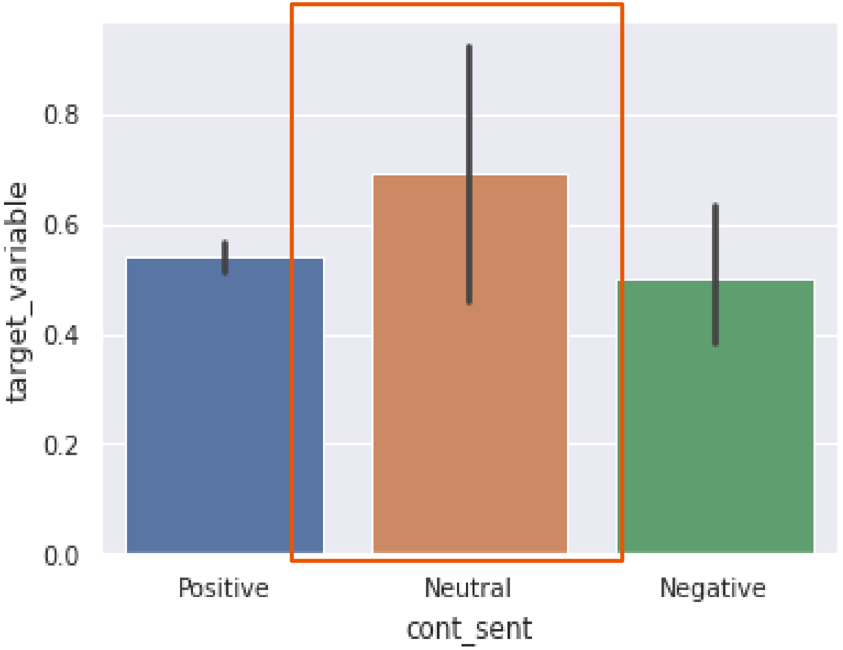
More positive response/profit in datasets.



Neutral > Positive> Negative Responses seems to affect target variable.

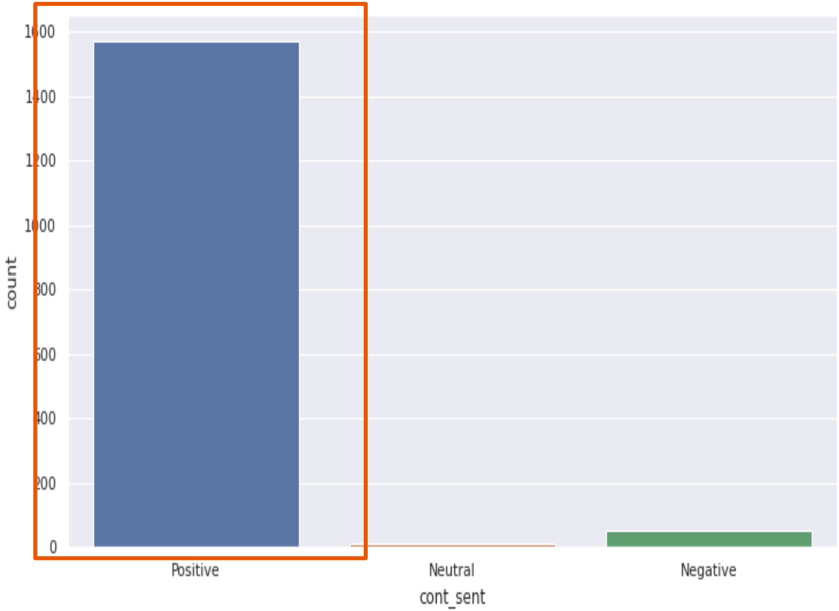
SUMMARY

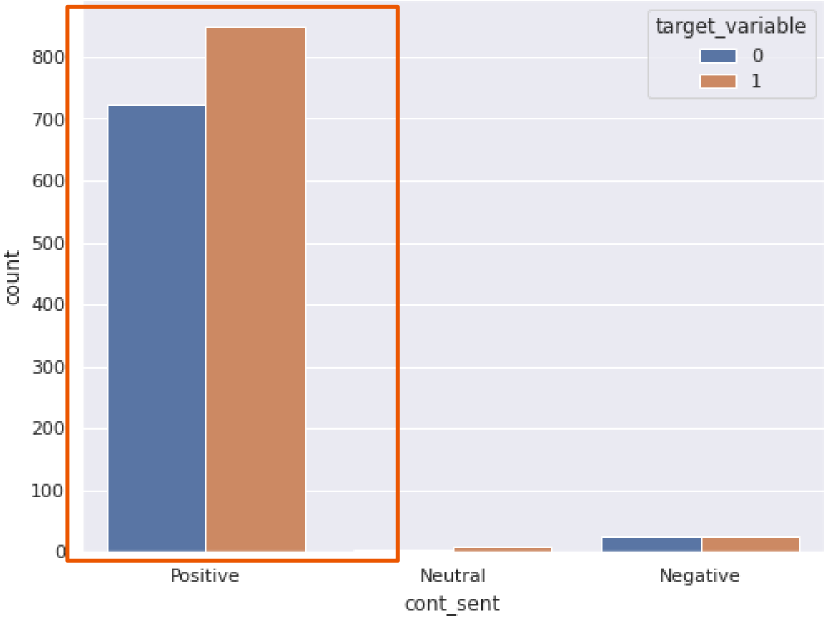
Bi-variate Plot (Polarity-Sentiment Analysis)



Count Plot (Univariate and Bivariate)

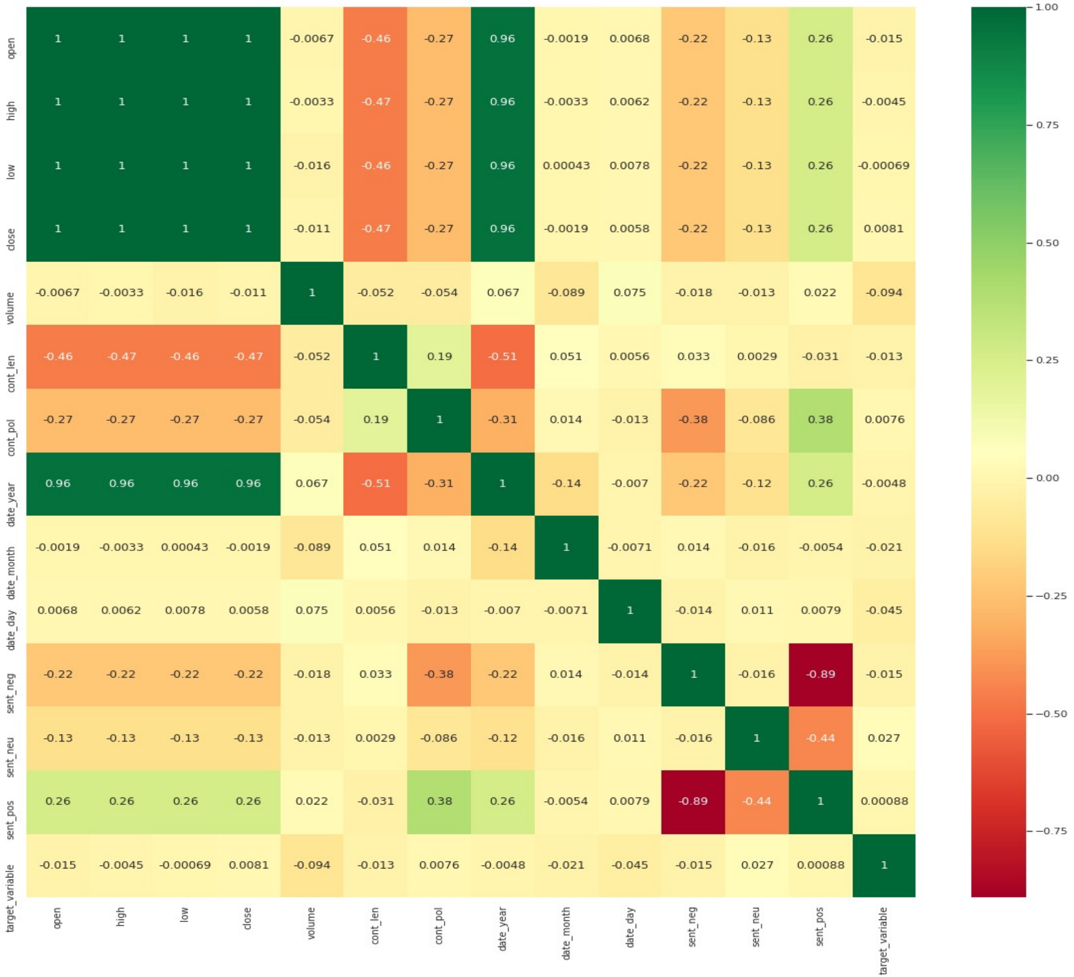
- Univariate Plot (Number of positive responses > Negative > Neutral)
- Positive response influences profit(1) in target variable/Closing Price.





Relation Among all the Dependent And Independent Variables

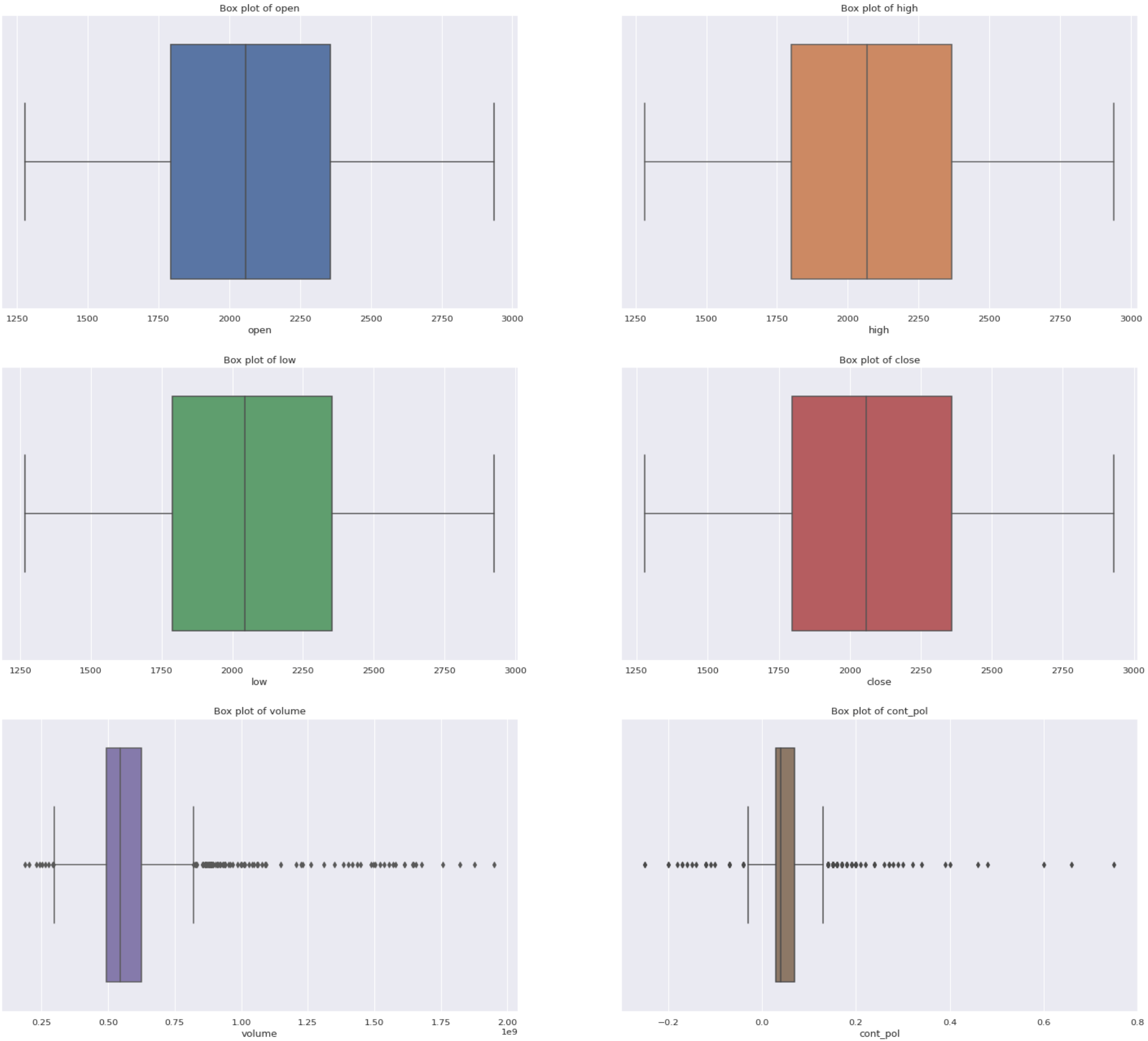
Heatmap



Check the Outliers:

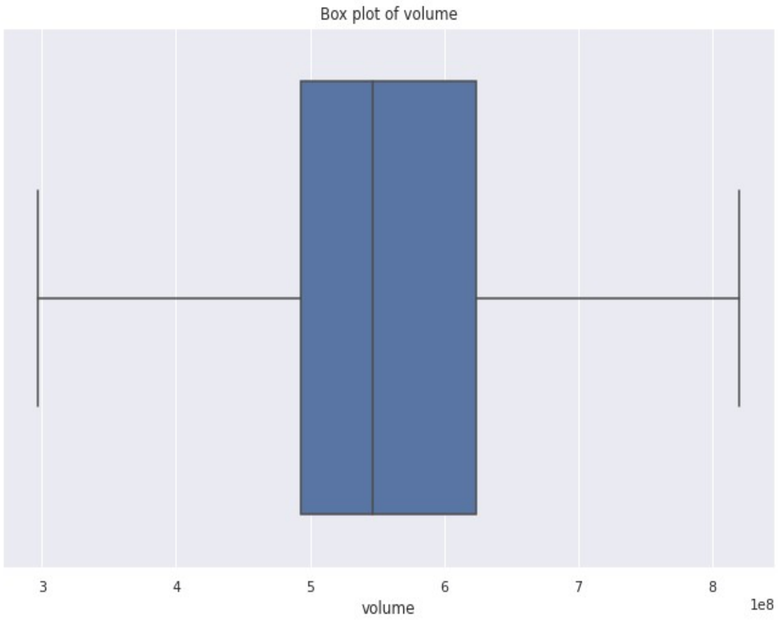
All the variables are outliers free other than:
- Volume
- Content Polarity

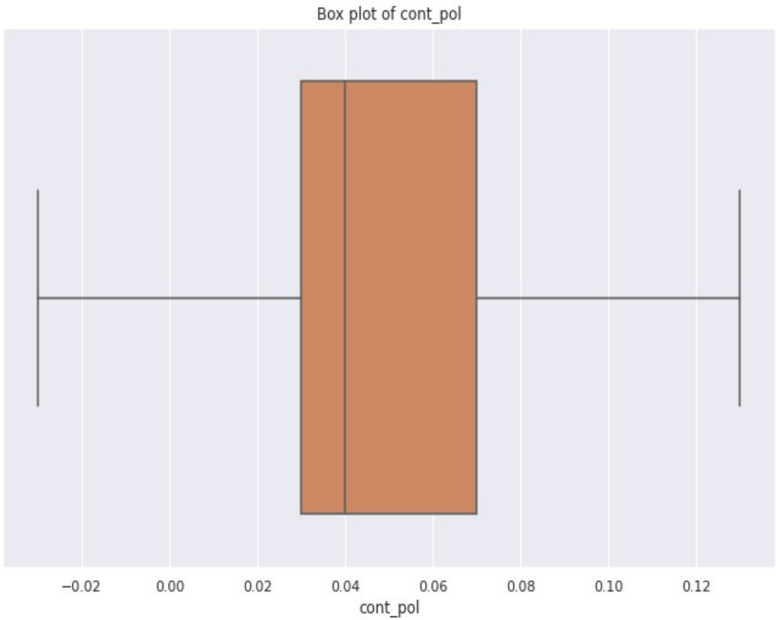
BoxPlot



Box Plot (After):

- IQR was performed where the outliers were treated via flooring and capping.
- Volume and Content Polarity columns are now outliers free.





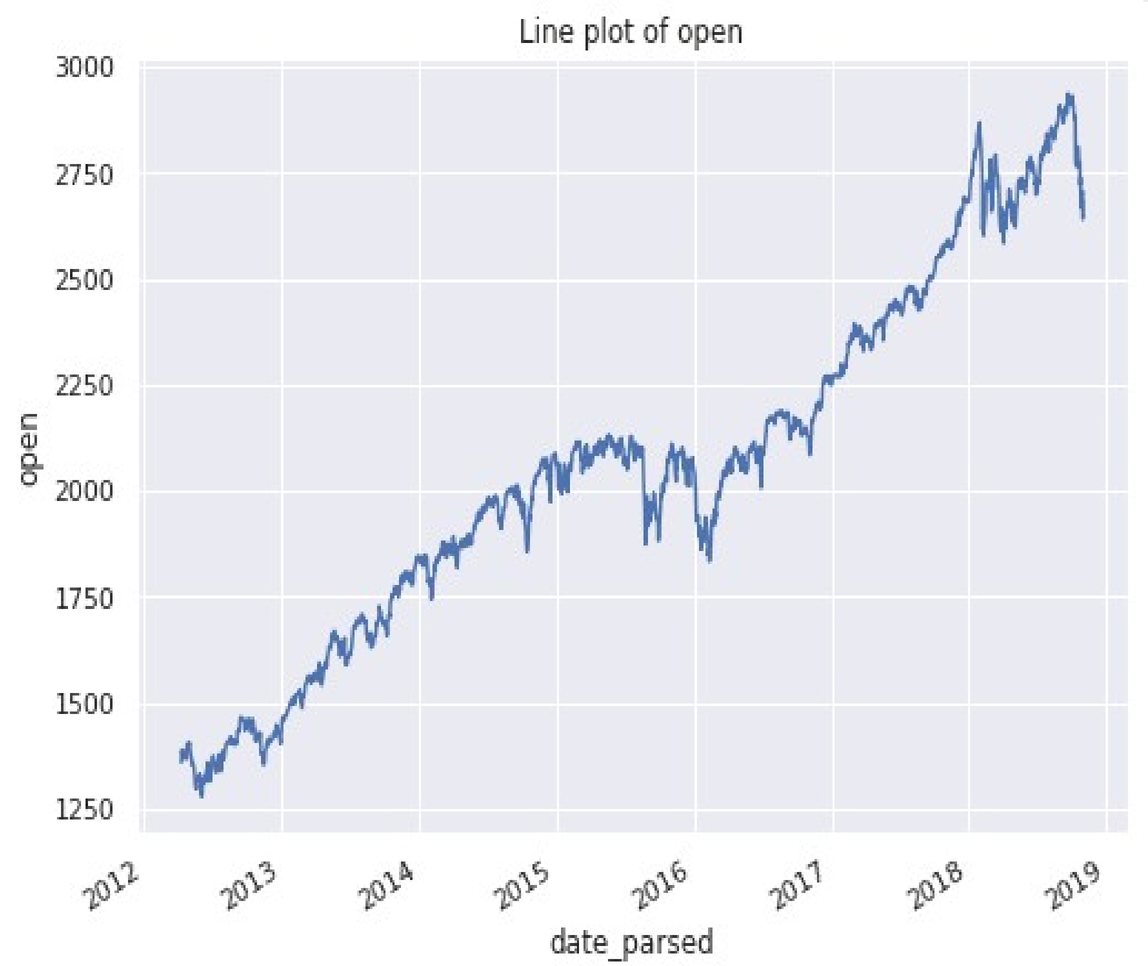
Data of all the variables are Normally Distributed.

Disribution Plot

- It seems that all the other variables like : Close, High, Low has similar line plot other than Volume.

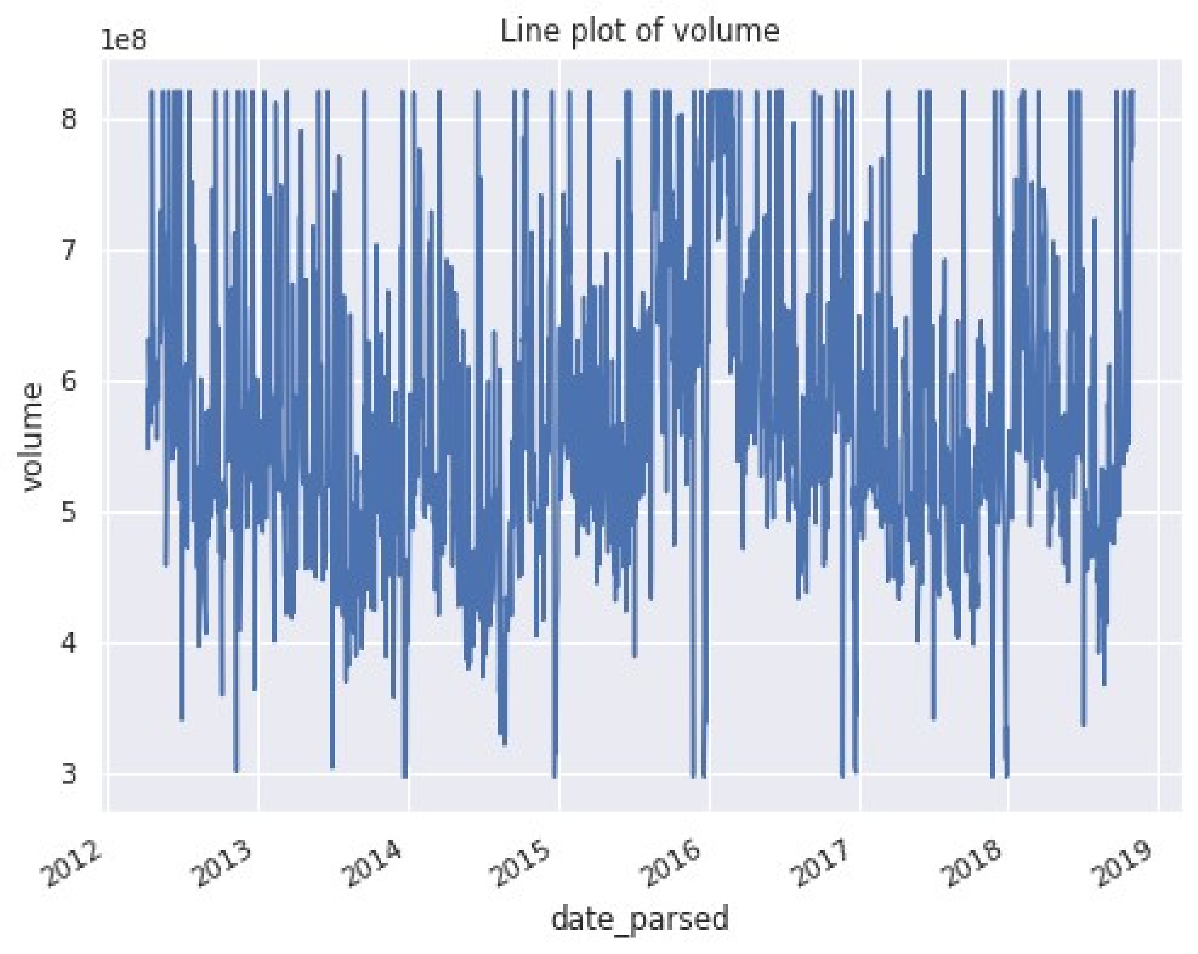
- All the variables value seems to increase as per the time.

Line Plot



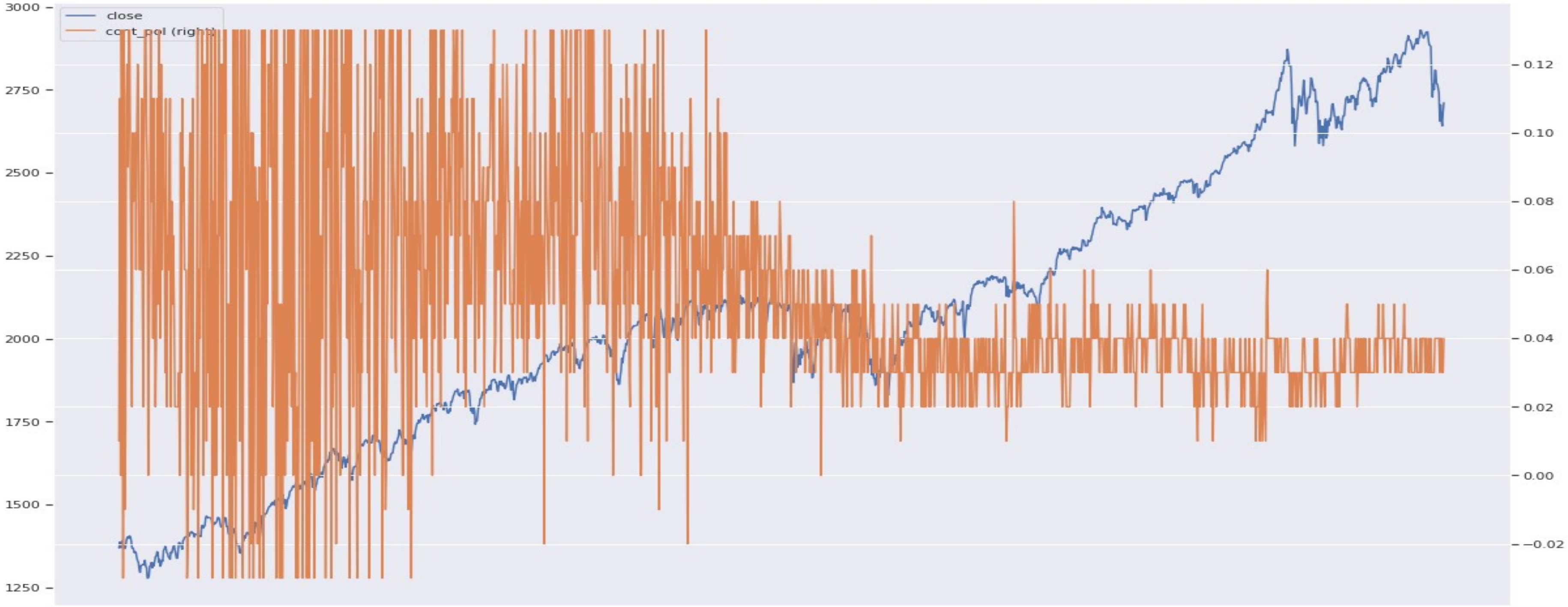
Volume seems to increase and decrease along with time.

Line Plot



Box Plot (After):

- Seems like closing price and Content Polarity are correlated with one another.
- Closing price increases with year but content polarity seems not to.

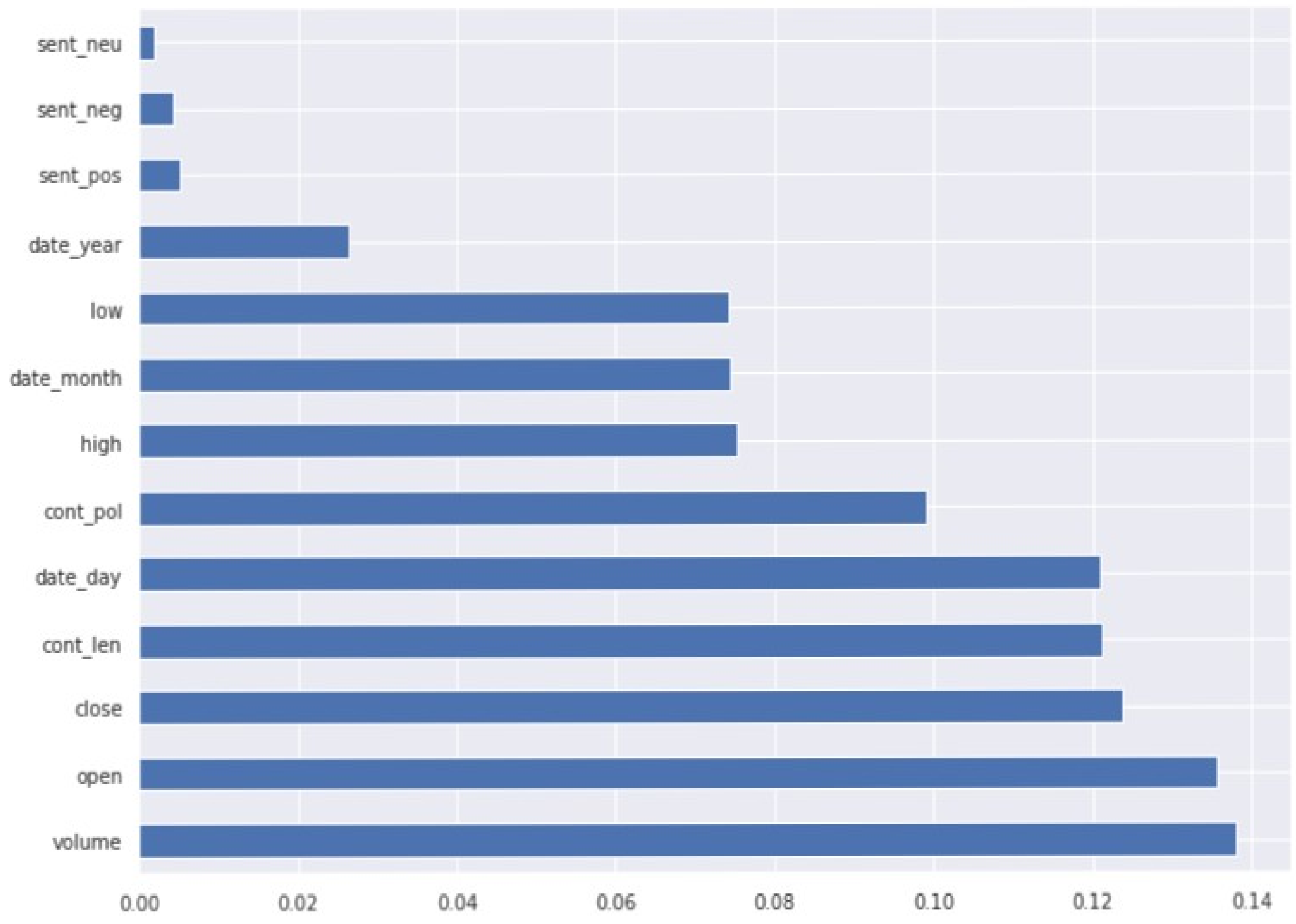


PREDICTION MODEL

Build a classification model to predict the market volatility.

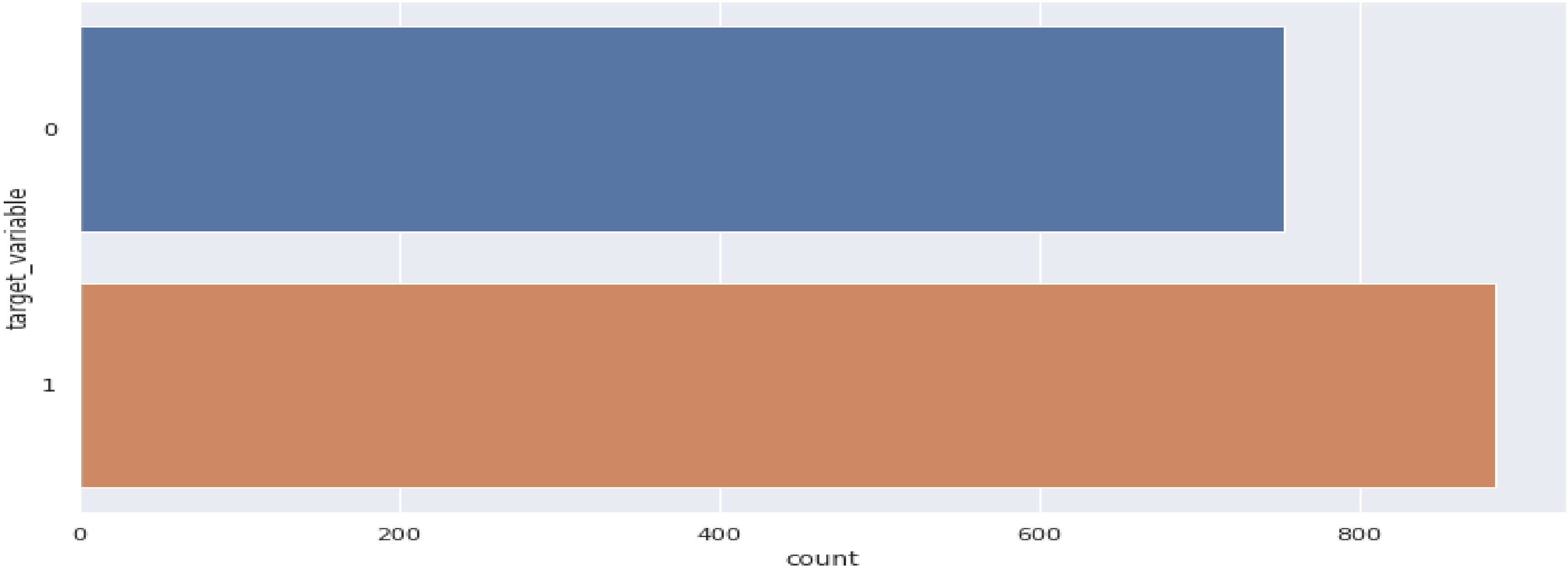
Most Imp. Features :

1. Volume
2. Open
3. Close
4. Cont_len
5. Date



TARGET VARIABLE

- Though positive target variable is quite more in comparison to negative target variable
  we cannot say dataset is imbalanced because data the difference is not so huge.



MODEL Building/Training

Logistic Regression was selected for a model.

```
!pip install logisticregression
```

```
from sklearn.learn_model import LogisticRegression
```

```
from sklearn.metrics import classification_report, accuracy_score
```

```
log_reg = LogisticRegression
```

```
log_reg.fit(x_train, y_train)
```

```
y_pred = log_reg.predict(X_test)
print(classification_report(y_test, y_pred))

acc_score = accuracy_score(y_test,y_pred)
acc_score_per = acc_score * 100
print('The accuracy score is', acc_score, '/', acc_score_per, '%'.)
```

MODEL BUILDING – Logistic Regression

- Classification Report and Accuracy score of our model (Before Hyperparameter Tuning)

	precision	recall	f1-score	support
0	0.83	0.43	0.57	161
1	0.62	0.92	0.74	167
accuracy			0.68	328
macro avg	0.73	0.67	0.65	328
weighted avg	0.73	0.68	0.66	328

The accuracy score is 0.676829268292683 / 67.6829268292683 %.

PERFORMANCE EVALUATION

Hyperparameter Tuning/ Evaluation metrics to increase the accuracy of the model.

True Negative: 69
(Predicted Loss as Loss)

False Positive: 92
(Predicted Loss as Profit)

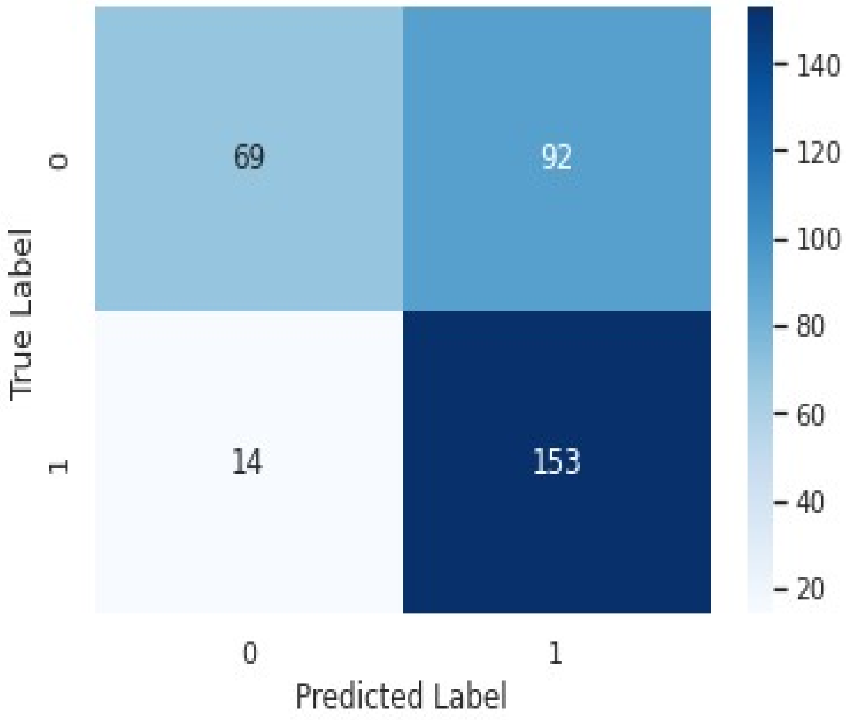
False Negative: 14
(Predicted Profit as Loss)

True Positive: 153
(Predicted Profit as Profit)

Summary

Confusion Matrix

Before Hyperparameter Tuning



AOC-ROC Curve

ROC Score

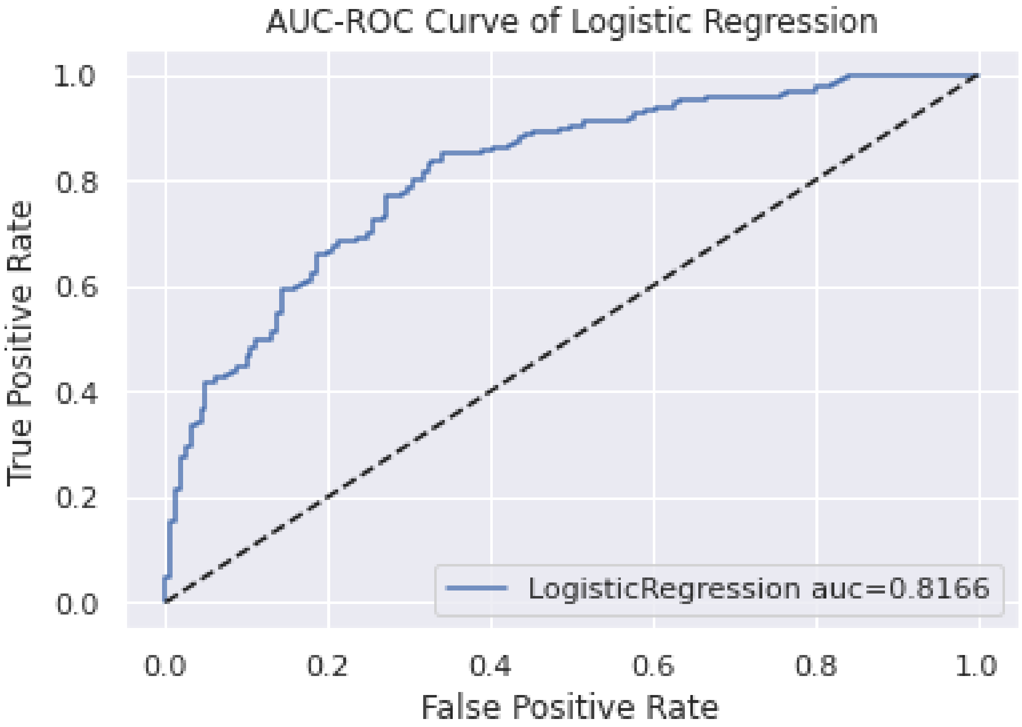
60.8166400119016626 /
81.66400119016626 %

Graph

The left corner of model is quite near to top-left corner but not exactly so the roc curve of is average.

Summary

Before Hyperparameter Tuning



Hyperparameter Tuning (GridSearchCV)

```
from sklearn.model_selection import GridSearchCV

penalty=['l1', 'l2', 'elasticnet']
solver=['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']
max_iter=[100,200,300,350]

random_grid={'penalty':penalty,
             'solver':solver,
             'max_iter':max_iter,
             }

log_reg_grid_search= GridSearchCV(estimator=log_reg,
param_grid=random_grid, cv=20, n_jobs=-1, verbose=2)
```

True Negative: 148
(Predicted Loss as Loss)

False Positive: 13
(Predicted Loss as Profit)

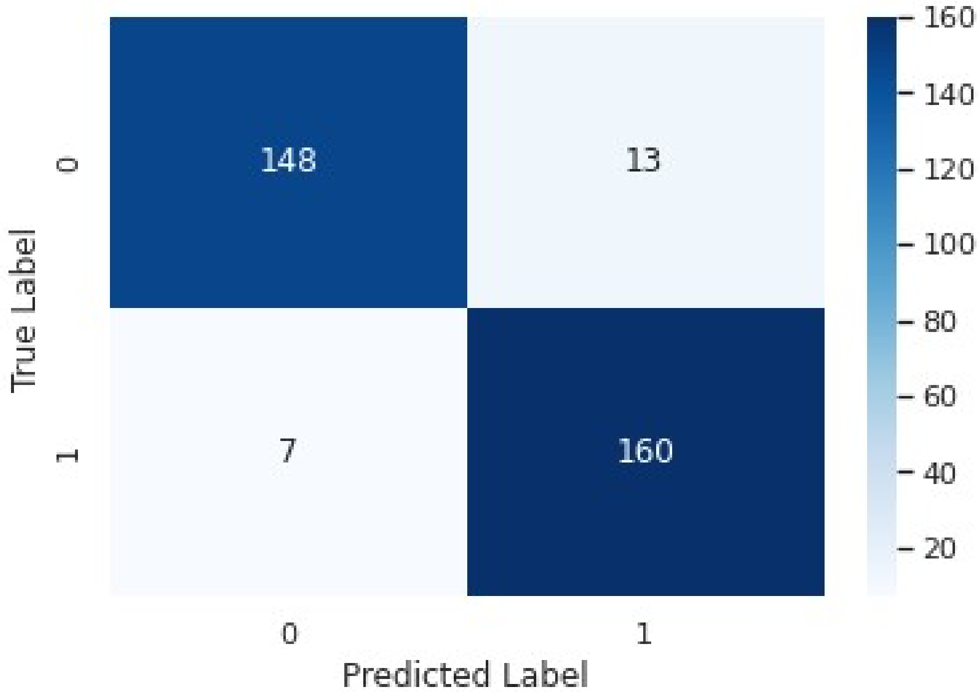
False Negative: 7
(Predicted Profit as Loss)

True Positive: 160
(Predicted Profit as Profit)

Summary

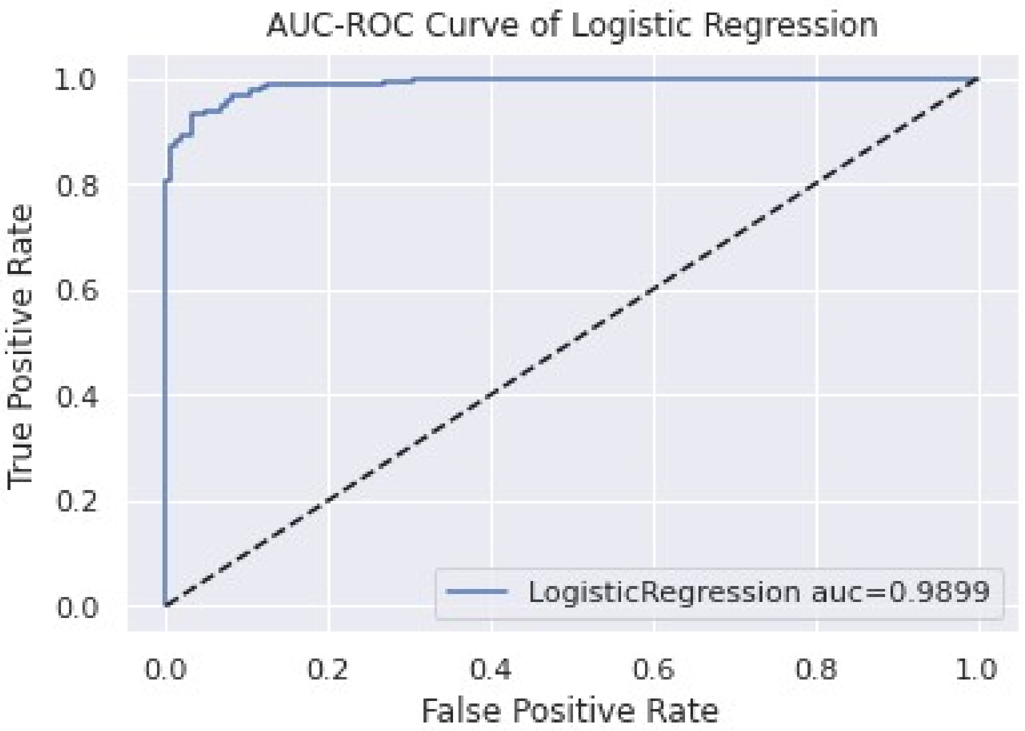
Confusion Matrix

After Hyperparameter Tuning



AOC-ROC Curve

After Hyperparameter Tuning



ROC Score

0.98988358686354 /
98.988358686354 %

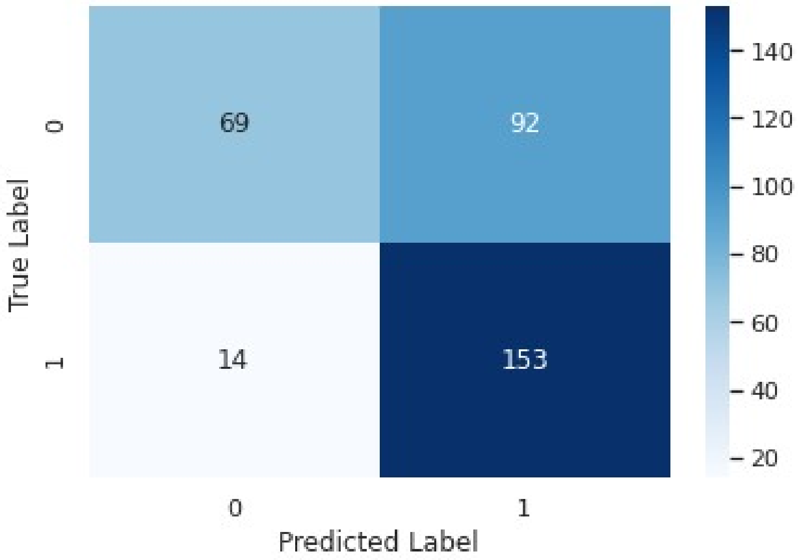
Graph

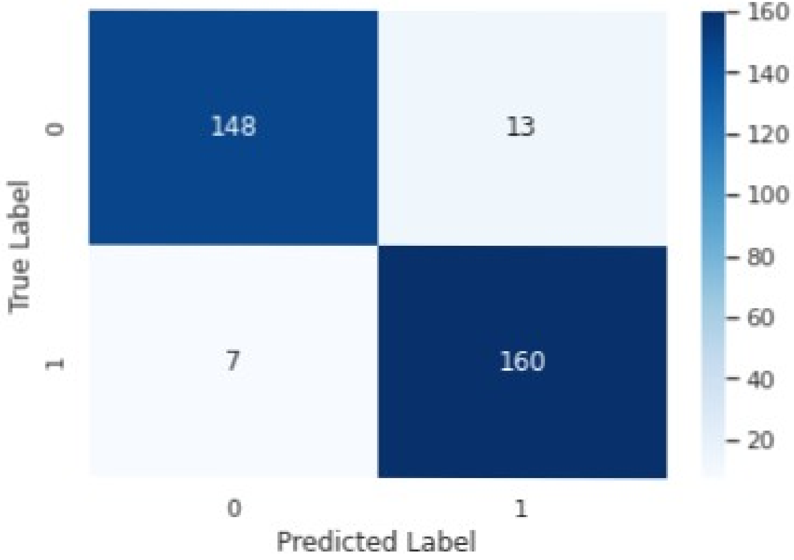
The left corner of model is so close to top-left corner hence model is good.

Summary

MODELS PERFORMANCES

- *Classification Report and Accuracy score of our model (After Hyperparameter Tuning)*
- *Increment in True Positive/ False Positive as expected..*





MODEL BUILDING – Logistic Regression

- Classification Report and Accuracy score of our model (After Hyperparameter Tuning)

	precision	recall	f1-score	support
0	0.95	0.92	0.94	161
1	0.92	0.96	0.94	167
accuracy			0.94	328
macro avg	0.94	0.94	0.94	328
weighted avg	0.94	0.94	0.94	328

The accuracy score is 0.9390243902439024 / 93.90243902439023 %.

Model Deployment

Flask along with HTML/CSS was used to deploy in local server.
Later deployed using Heroku.

MODEL DEPLOYMENT

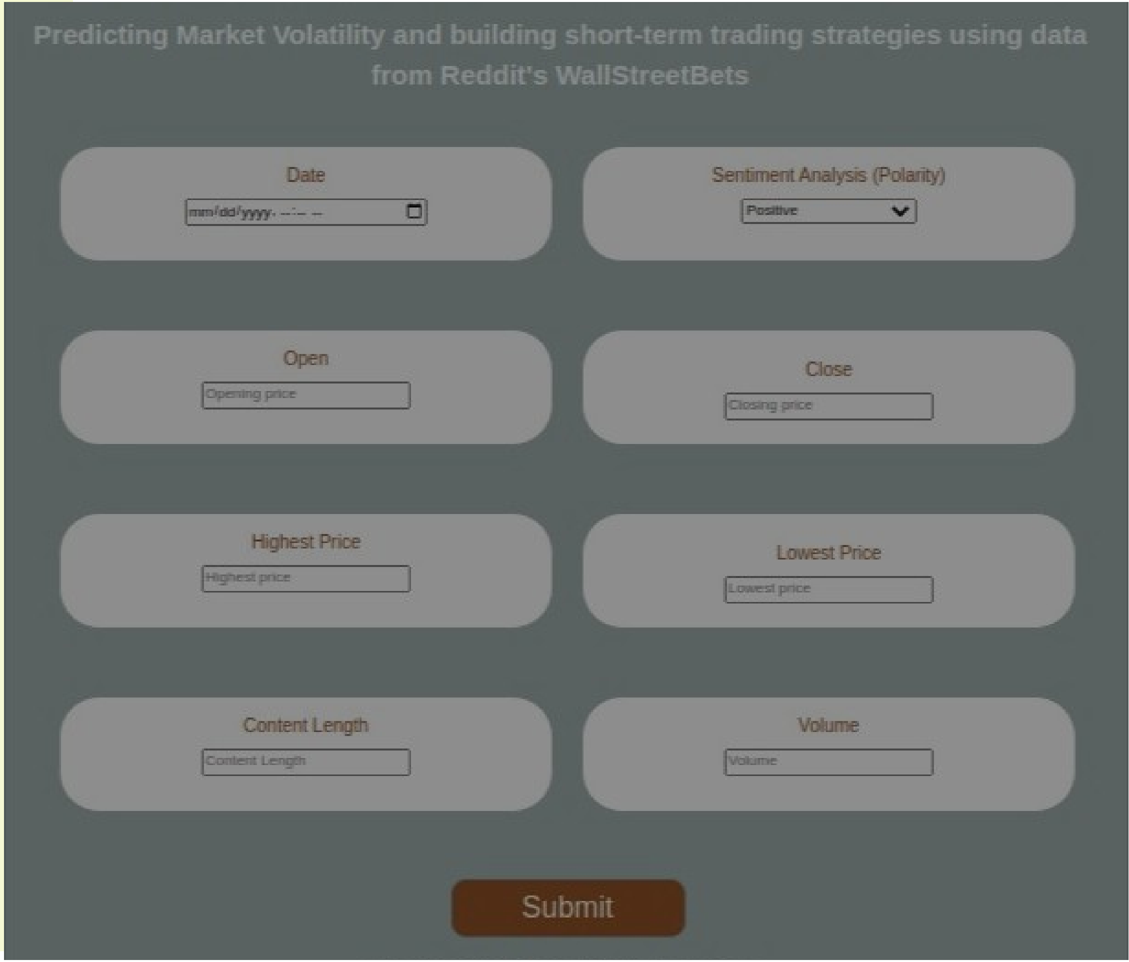
Tools:

Flask, HTML,
CSS, Heroku

Input

Date, Sentiment Analysis, Open, Close, Higher Price, Lower Price, Content Length, Close

Summary



MODEL DEPLOYMENT

Tools:

Flask, HTML,
CSS, Heroku

Output

Gives the predicted output from the trained model in the form of Profit/Loss.

Summary

Result:

CONCLUSION

MODEL CONCLUSION:

CONDITIONS which have the following characteristics,

- ***Having HIGH opening price itself;***
- ***High Volume;***
- ***Positive Sentiment Analysis;***
- ***Lengthy/Informative Detailed comments;***

are likely to lead us to Profit.

WHAT CAN WE DO

General

1. Publish more ***Positive Contents*** ;

2. Promote more **detailed and informative** contents

3. Reduce/Remove the **negative contents** from Social Media asap if found.

LIMITATION & NEXT STEP

HOW TO IMPROVE

1. Only applied Logistic Regression:

Apply and compare other tuned performance.

2. Used only Reddit's API:

Collect API from as much as resources possible.

END