Capstone Project
Nifty50 Closing Price Forecasting

-Presenting by:
B Manoj Kumar Reddy

Content

- Problem statement
- Inferences from visualization of features
- Checking stationarity of data
- Feature engineering
- Forecasting using time series techniques
- Forecasting using classification techniques
- Conclusion
- Improvement
- Challenges

Problem Statement

- Perform time series analysis on the NIFTY stock price and forecasting using univariate ARIMA and ARIMAX modeling techniques.
- The problem is further simplified to just predict the direction of nifty index movements in the next N days (throughout our experiments N can take values 1, 5, and 30). Initially, we will take N=1, that means we want to predict the NIFTY 50 index movement in the next day. This is represented as a classification task where there are two possible outcomes (either the index went up in the next day or it went down).

Data Summary:

Data set name-- Nifty-50_Data

Date--

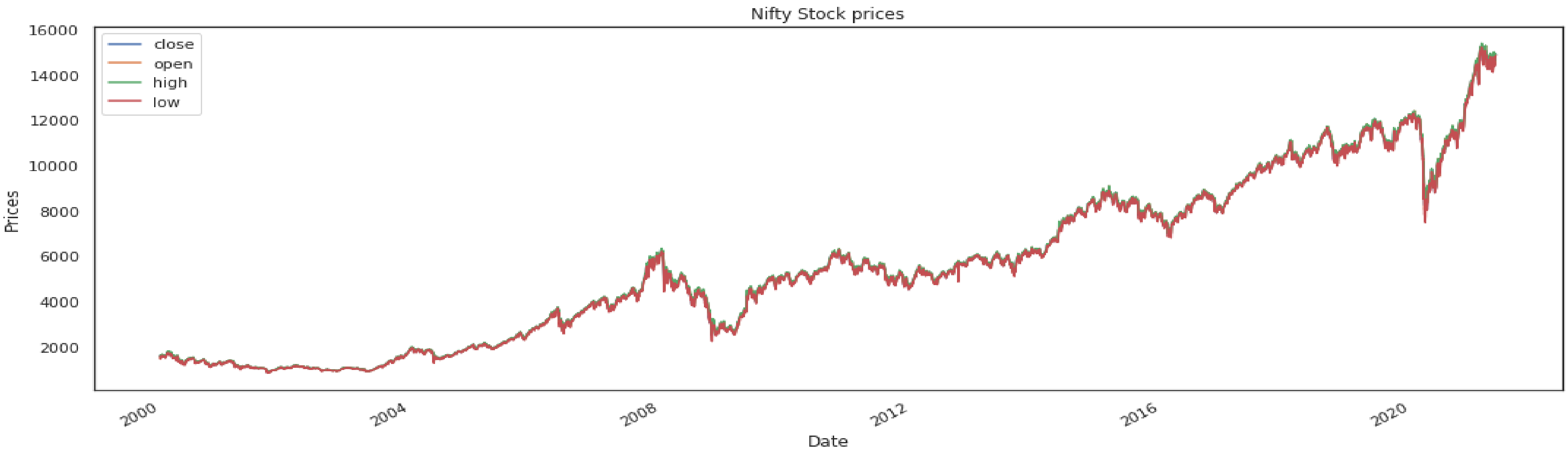
- Start -- 2000-01-03
- End-- 2021-05-09

Shape--

- Rows -- 5301
- Columns-- 5

Columns-- [‘Open’, 'High', ‘Low’,’Close’]

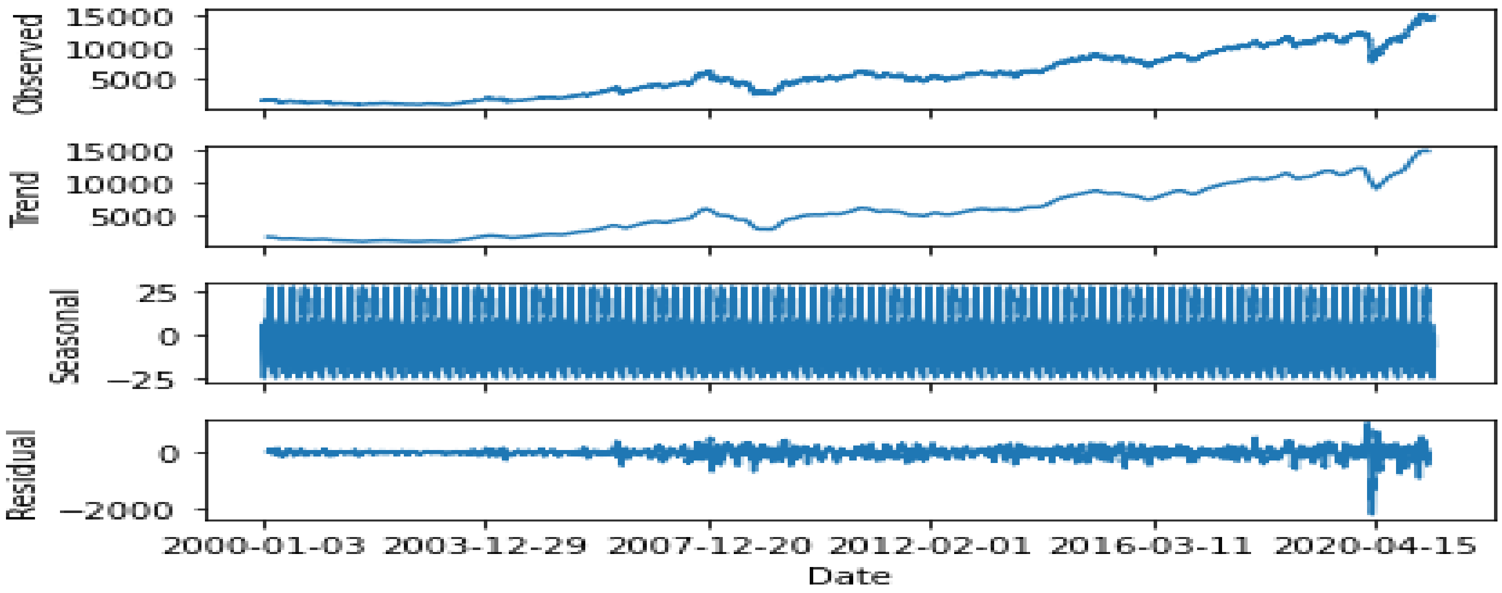
Trend of features across years



From the above graphs, we have observed that-

- Drop 2008-2009:- This can be attributed to the Great Recession that happened during this period.
- Drop 2016:- This can be attributed to Demonetisation drive by the central government.
- Drop 2020:- This is due to the global breakdown amid coronavirus pandemic induced lockdown in India.
- Rise 2020-2021:- The stock price started rising.This can be attributed to the lifting of lockdown in the country and across the world.

Decomposition Plot on Close Price(Additive)



Result:

Trend: Their is positive trend in the in the dataset.
Seasonality: Seasonality in the data.
Residual: Residual in the data.

Univariate Forecasting using different time series techniques

- ARIMA
- ARIMAX
- Facebook Prophet

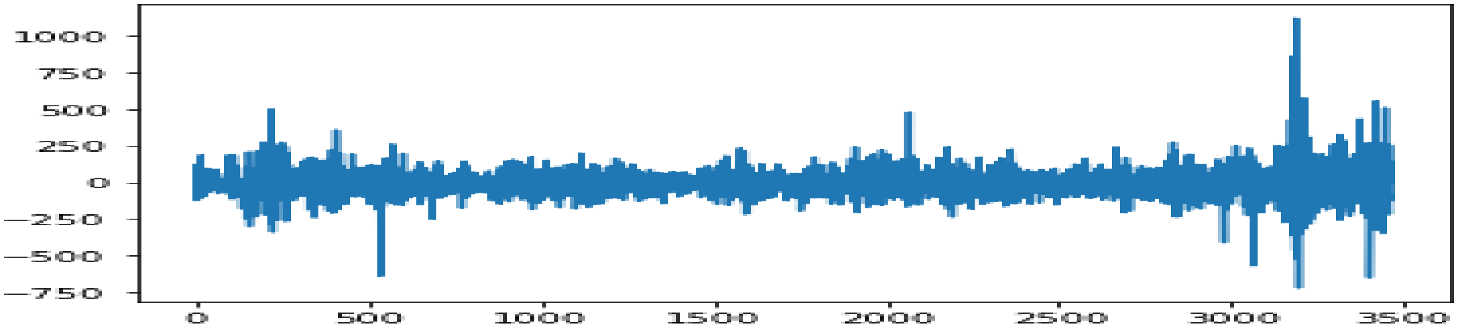
Check for stationarity:

```
Results of Dickey-Fuller Test on close price:
Test Statistic                    0.793839
p-value                           0.991536
#Lags Used                       29.000000
Number of Observations Used    5236.000000
Critical Value (1%)              -3.431600
Critical Value (5%)              -2.862092
Critical Value (10%)             -2.567064
dtype: float64
```

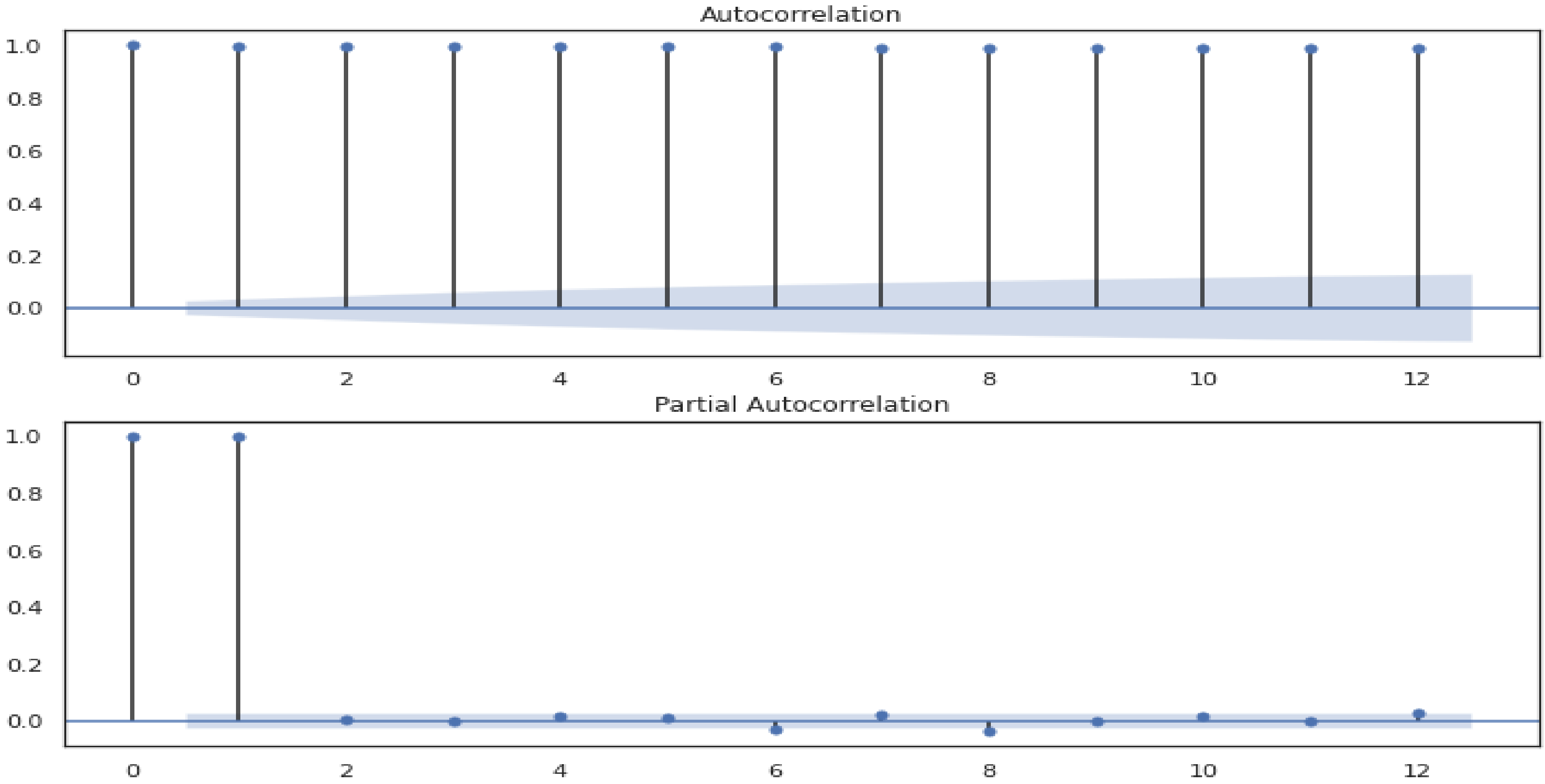
```
Results of Dickey-Fuller Test on 1st difference close price:
Test Statistic                -1.320886e+01
p-value                        1.057066e-24
#Lags Used                     2.800000e+01
Number of Observations Used    5.236000e+03
Critical Value (1%)            3.431600e+00
Critical Value (5%)            2.862092e+00
Critical Value (10%)           2.567064e+00
dtype: float64
```

DF close price : Here, the test statistic is greater than the critical value, we fail to reject the null hypothesis (which means the series is not stationary).

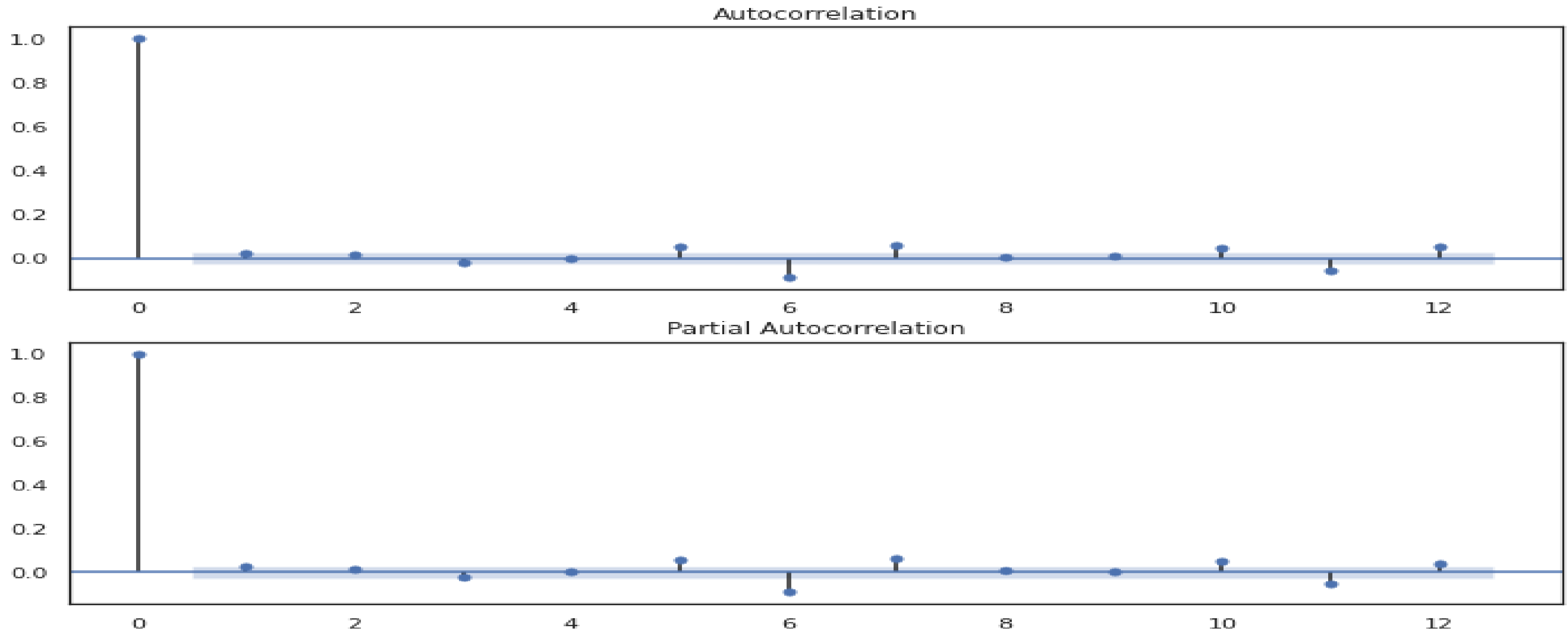
DF 1st diff close price: Here, test statistic is less than the critical value, we can reject the null hypothesis (aka the series is stationary)



ACF and PACF for Closing Price



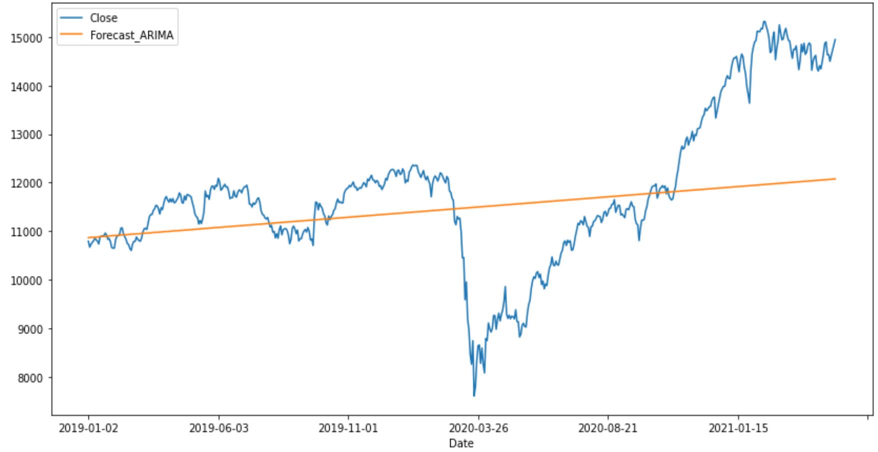
ACF and PACF First Difference Closing Price

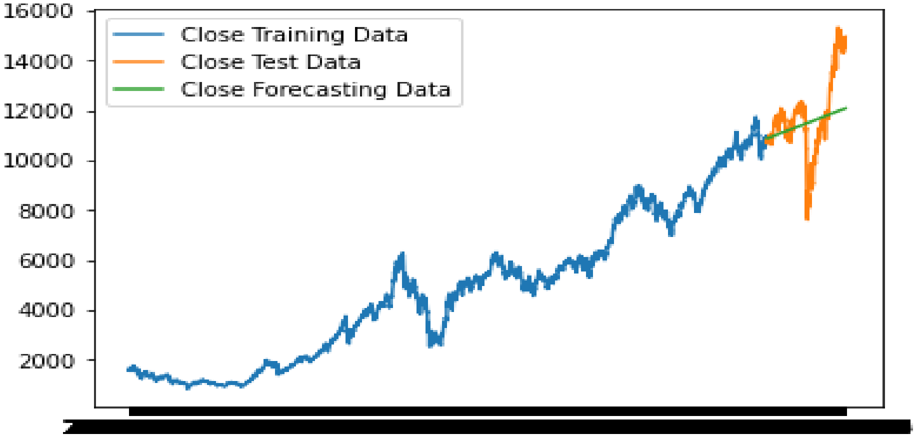


- Here, both ACF and PACF are diminishing thus the ARMA(p,q) best fits our analysis

ARIMA:

An „AutoRegressive Integrated Moving-Average“ (ARIMA) model belongs to the one of the most used methodology approaches for analyzing time series. This is mostly because of it offers great flexibility in analyzing various time series and because of achieving accurate forecasts, too. Its other advantage is that for analyzing single time series it uses its own historical data.





Best model: ARIMA(3,1,3)
Mean absolute percentage error: 16.35 %

ARIMAX:

$$Y_t = C + \nu(B)X_{t-1} + N$$

When Xt and Nt are assumed to follow ARMA model is known as the ARMAX model. This ARMAX model is quite different from ARMA model, because we work with two different series Xt and Yt - output series Yt is related to input series Xt

Exogenous Features(X):

Lags: **Time lags are one of the key features of the time series analysis. In our analysis we have considered 3,5,10,15,30 lags of all the existing features.**

Momentum: **How about calculating some statistical values based on past values? This method is called the rolling window method because the window would be different for every data point. Here, we use MA and EWMA to find new features from the analysis.**

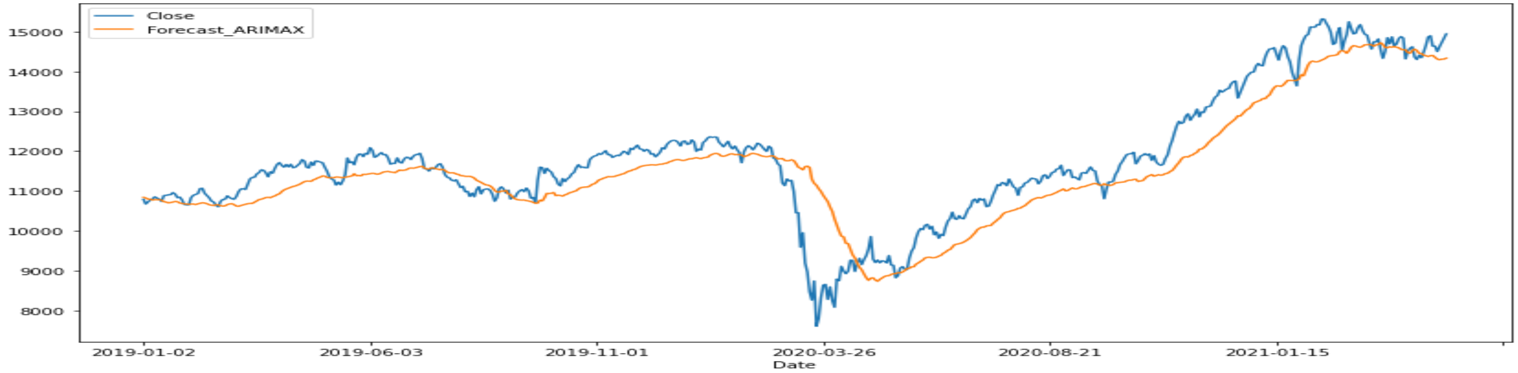
Final Features:

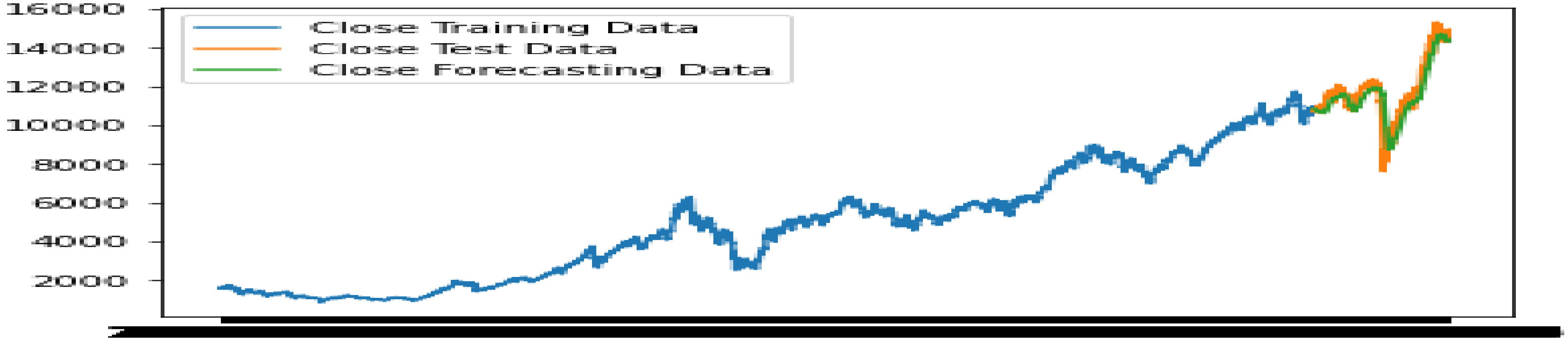
Date	Open	High	Low	Close	Open_mean_lag3	Open_mean_lag7	Open_mean_lag30	Open_std_lag3	Open_std_lag7	Open_std_lag30
2000-01-03	1482.15	1592.90	1482.15	1592.2	5547.483398	5542.514160	5514.119141	45.522327	68.892075	141.049759
2000-01-04	1594.40	1641.95	1594.40	1638.7	1482.150024	1482.150024	1482.150024	45.522327	68.892075	141.049759
2000-01-05	1634.55	1635.50	1555.05	1595.8	1538.275024	1538.275024	1538.275024	79.372734	79.372734	79.372734
2000-01-06	1595.80	1639.00	1595.80	1617.6	1570.366699	1570.366699	1570.366699	78.991394	78.991394	78.991394
2000-01-07	1616.60	1628.25	1597.20	1613.3	1608.250000	1576.724976	1576.724976	22.787222	65.737923	65.737923

```
exogenous_features= ['Open_mean_lag3', 'Open_mean_lag7',
    'Open_mean_lag30', 'Open_std_lag3', 'Open_std_lag7', 'Open_std_lag30',
    'High_mean_lag3', 'High_mean_lag7', 'High_mean_lag30', 'High_std_lag3',
    'High_std_lag7', 'High_std_lag30', 'Low_mean_lag3', 'Low_mean_lag7',
    'Low_mean_lag30', 'Low_std_lag3', 'Low_std_lag7', 'Low_std_lag30',
    'Close_mean_lag3', 'Close_mean_lag7', 'Close_mean_lag30',
    'Close_std_lag3', 'Close_std_lag7', 'Close_std_lag30']
```

ARIMAX Result:

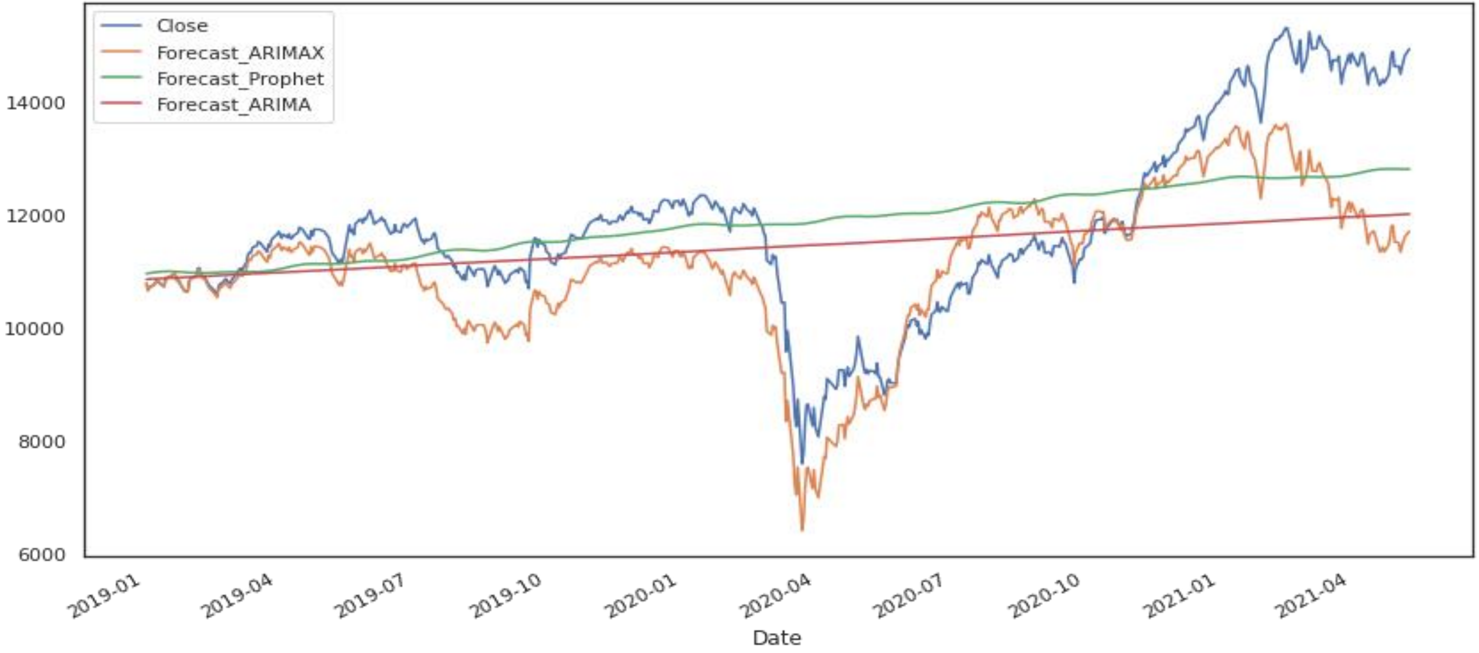
Best model: ARIMA(1,0,0)
Mean absolute percentage error: 8 %





Facebook Prophet

Forecast Comparison



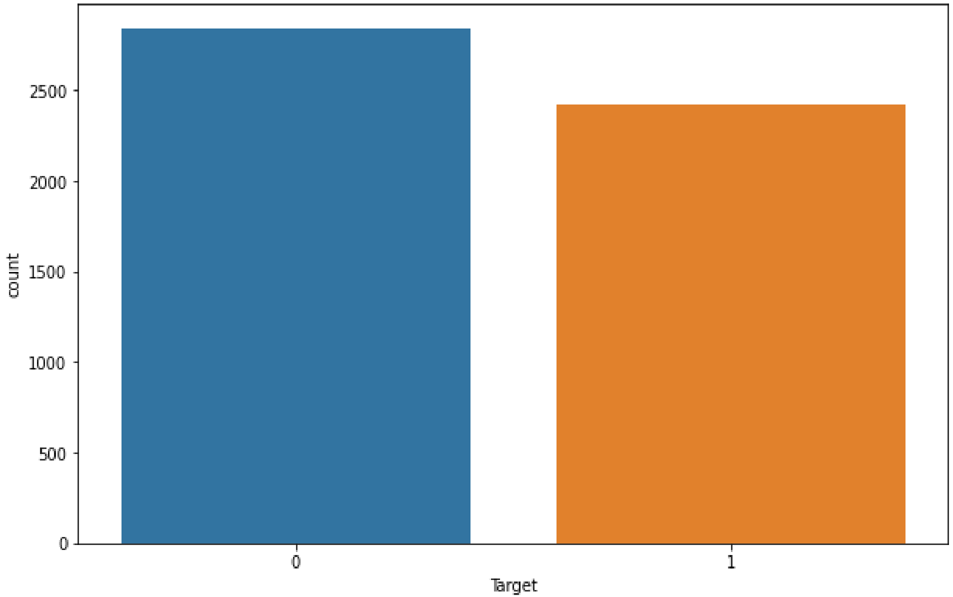
Mean absolute percentage error of Prophet: 8.86 %
Mean absolute percentage error of ARIMA: 11.03 %
Mean absolute percentage error of ARIMAX: 8.14 %

Forecasting Using Classification Techniques

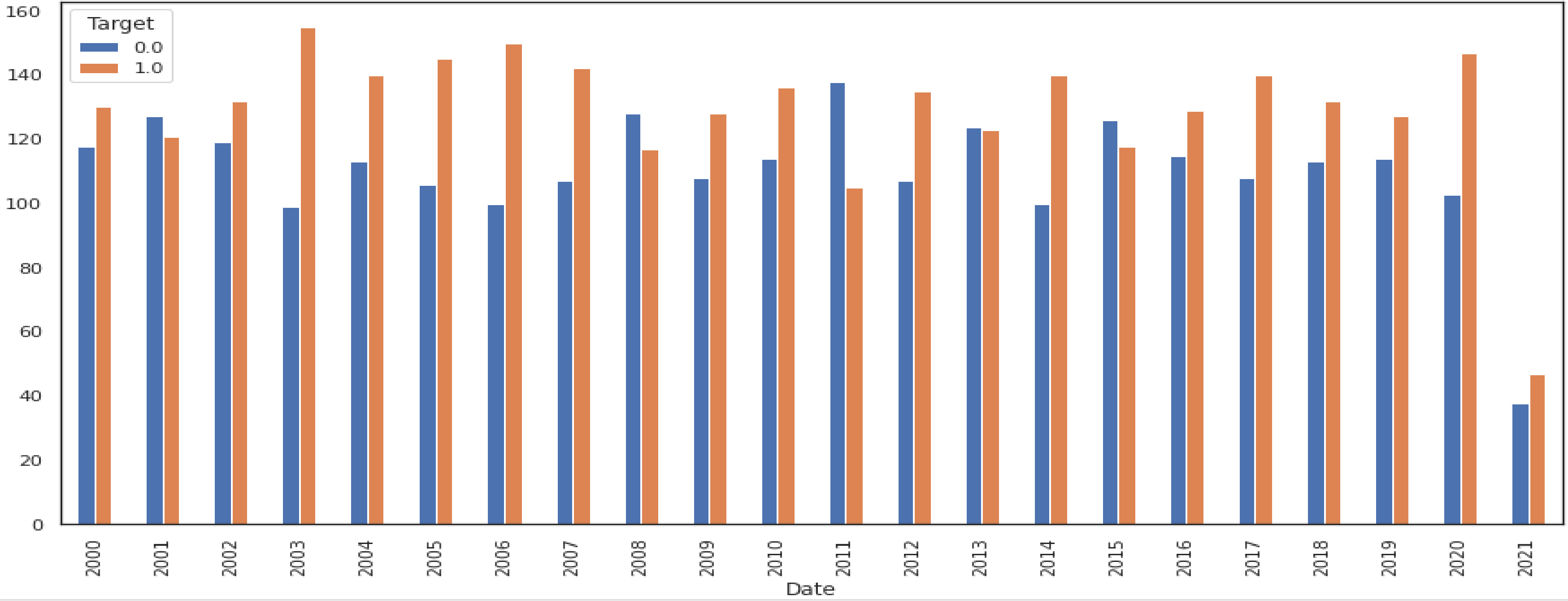
- Time series model can be converted into a classification models.
- Data can be fed into a classification algorithms .

Creating Target Variable

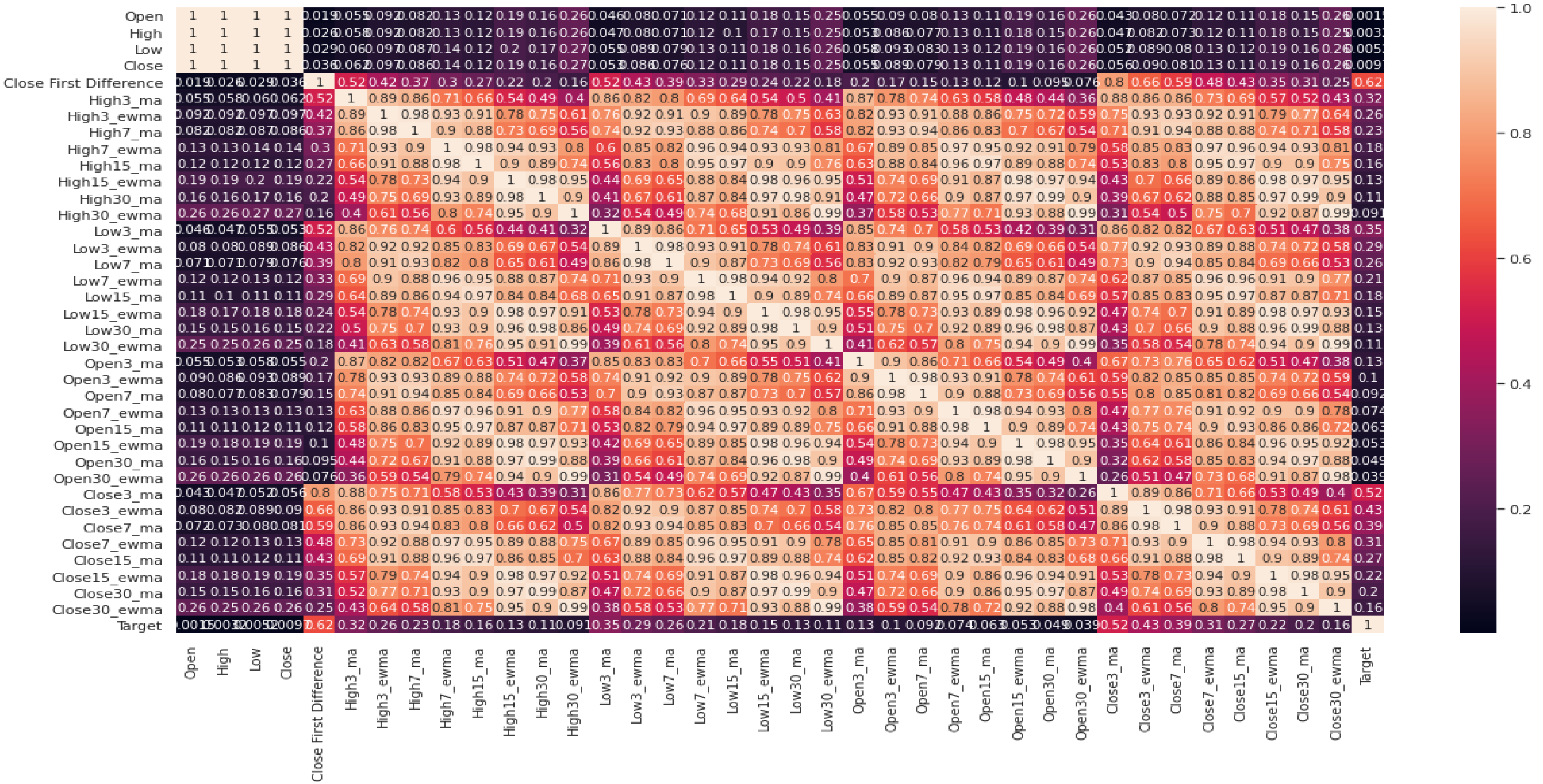
- Target = Current Day **Close price**-Previous Day **Close price**
- Class 1 = Target >= 0
- Class 0 = Target < 0



Distribution of Target Variable Classes in each year



Correlation among features



Classification Models Used

- XGBoost Classifier
- LGB Classifier

Why XGBoost and LGB?

- Independent features are highly correlated
- Faster
- Higher efficiency
- Less memory usage

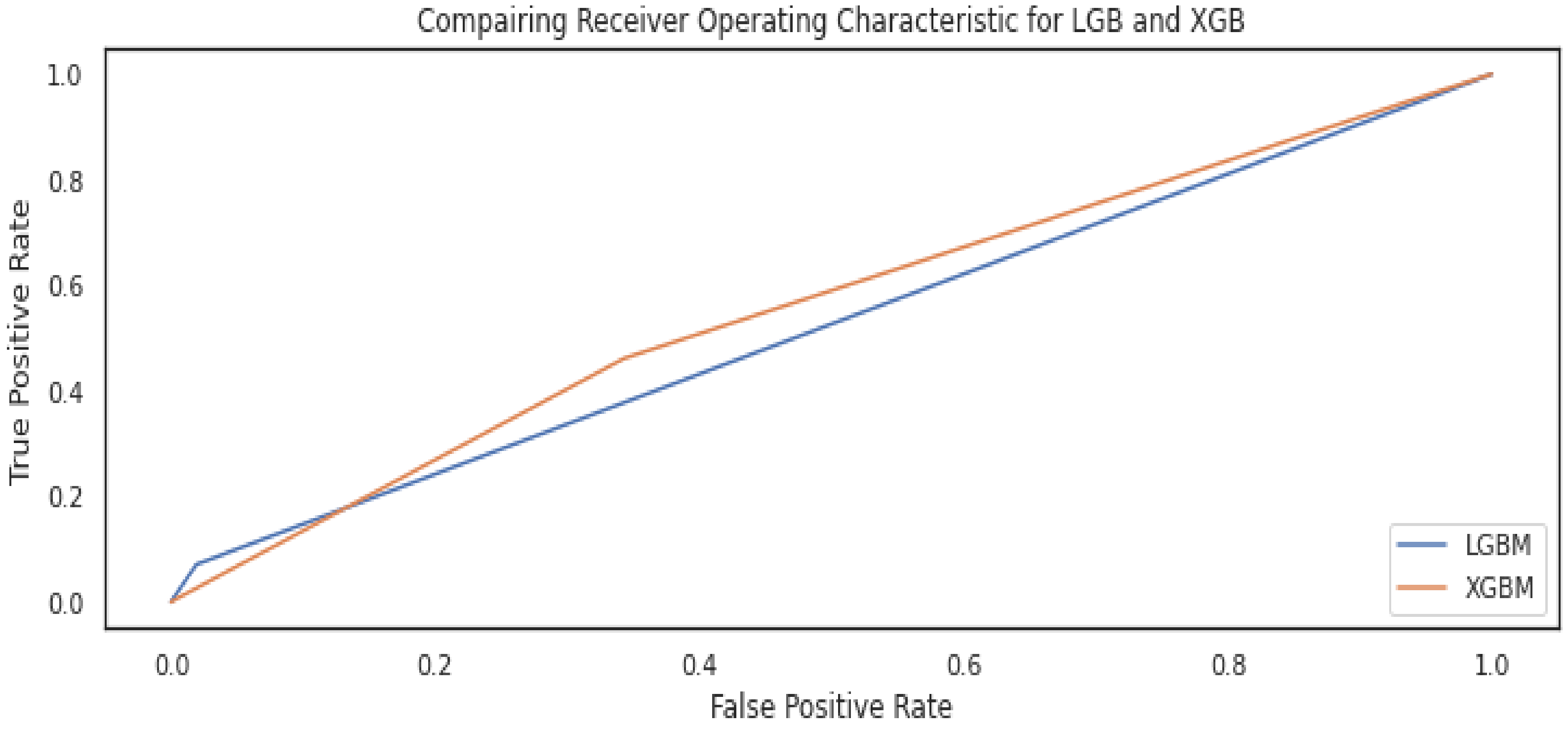
Train-Validation-Test Split

- Data split based on date.
- Train set: Between 2000-01-01 and 2016-01-01
- Validation set: Between 2016-01-01 and 2019-06-01
- Test set:Between 2019-06-01 and 2021-05-09

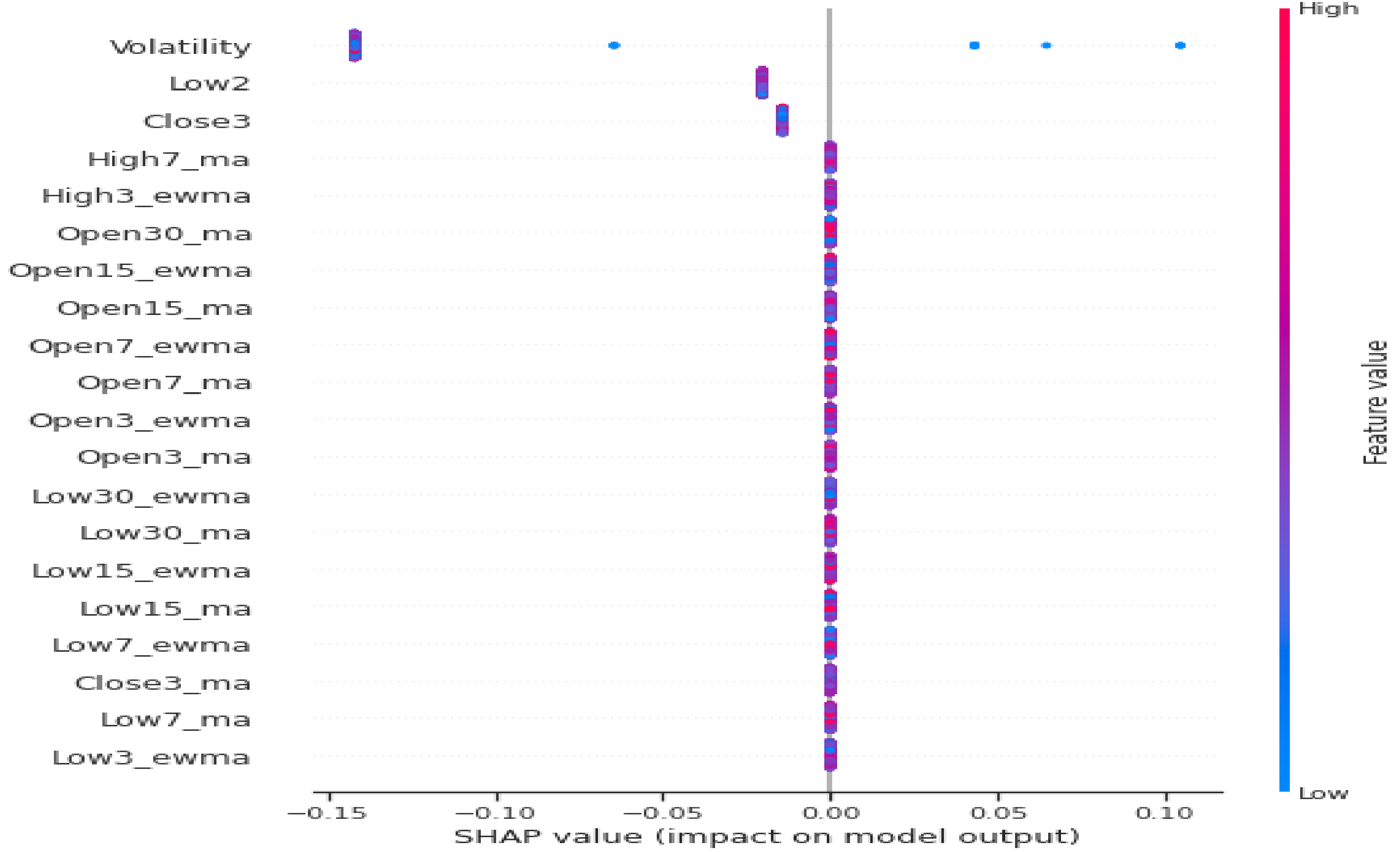
Model Performance

Accuracy of Models			
Model	Test Accuracy	Validation Accuracy	Test Accuracy
XGB	0.56	0.58	0.54
LGB	0.556	0.57	0.456

Comparing ROC curves for classifiers



Feature Importance Analysis



Conclusion

- While using regression models,residuals are quite high.
- Difficult to predict numerical target value.
- While using classification models, we are have higher chance of predicting the outcome.

Improvement

- Forecast for more than 1 day for unseen data.
- More features can be included to increase model efficiency.

Challenges

- Time Series is often considered a difficult topic to master.
- Dataset was small as we have only 5 columns and 5300 rows.
- Coming up with features was though as none of us was having any domain knowledge.
- Model performance keep on changing when we changed any new feature.

Q & A