Stock Market Price Prediction

Group 15

CS4191

Problem

- Stock prediction uses **Time Series Forecasting**.
- We have used the following univariate Time Series Forecasting methods:
  1) ARIMA
  2) LSTM
- Variable Predicted : Closing Price
- The data used is for the IT sector of India.

ARIMA - Introduction

- Combines two models, AR(Auto Regressive) and MA(Moving Average).
- **AR** =>

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} + \epsilon_1$$

Here, the current term, depends only on its lagged values.

- **MA**=>

$$Y_t = \alpha + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$$

Here, the current term depends on its lagged forecast errors.

ARIMA - Introduction

- Combined model =>

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$$

- The AR and MA combined model can only be used for stationary time series.
- Thus, the first step involves finding the variable d, i.e., the order of differencing required to make the time series stationary.

ARIMA - Step 1 (Finding d)

- We check the stationarity of the data at 0th order of differencing

```
result = adfuller(x_train['Close'].dropna())
print('p-value: %f' % result[1])
```

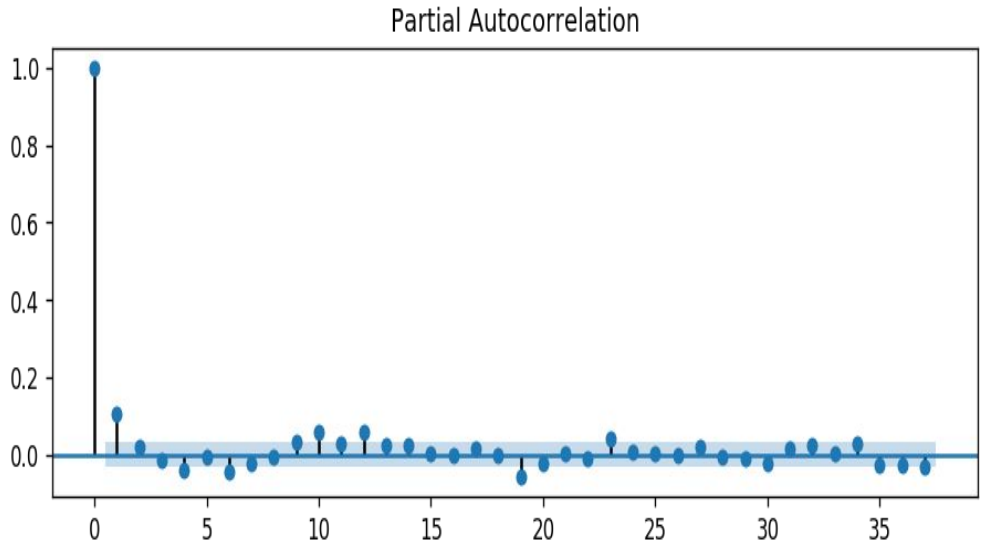
- We consider =>1) H_0(Null Hypothesis) : Time series is non-stationary
  2) Alpha (Significance level) : 0.05
- If p-value < Alpha, we reject H_0, thus proving that the time series is stationary.
- **In our case, the 0th order showed stationarity. But during testing, the predictions were not good.**
- **On using 1st order, the predictions proved to be good.**
- **Hence, we considered d=1.**

ARIMA- Step 2 (Finding AR(p))

- ‘p’ defines the number of time lags to be used.

```
plot_pacf(x_train['Close'].diff().dropna())#, ax=axes[1])
```

- First two time lags have an autocorrelation value > threshold.
- Hence, p=2.

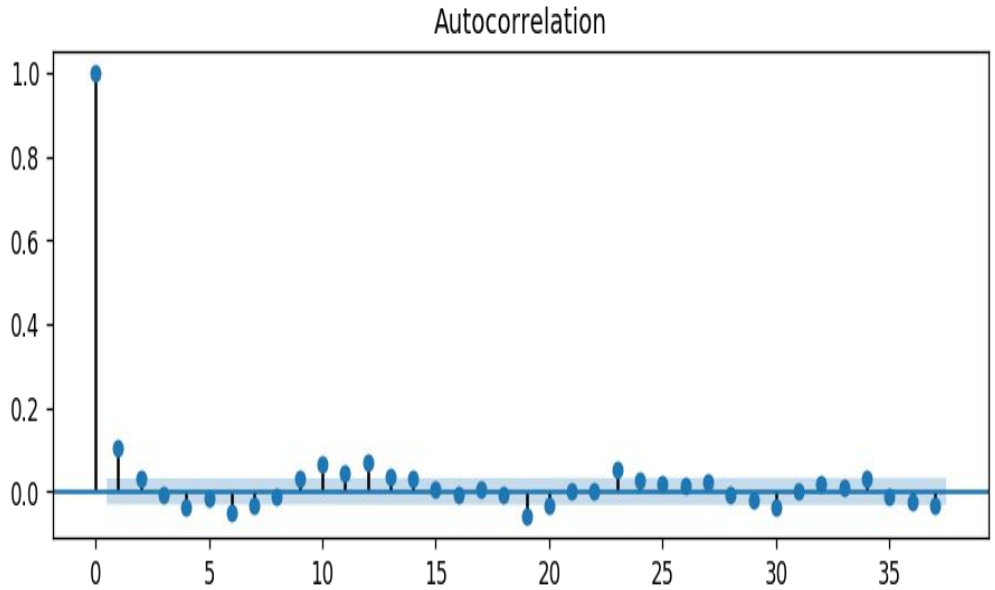


ARIMA- Step 3(Finding MA(q)

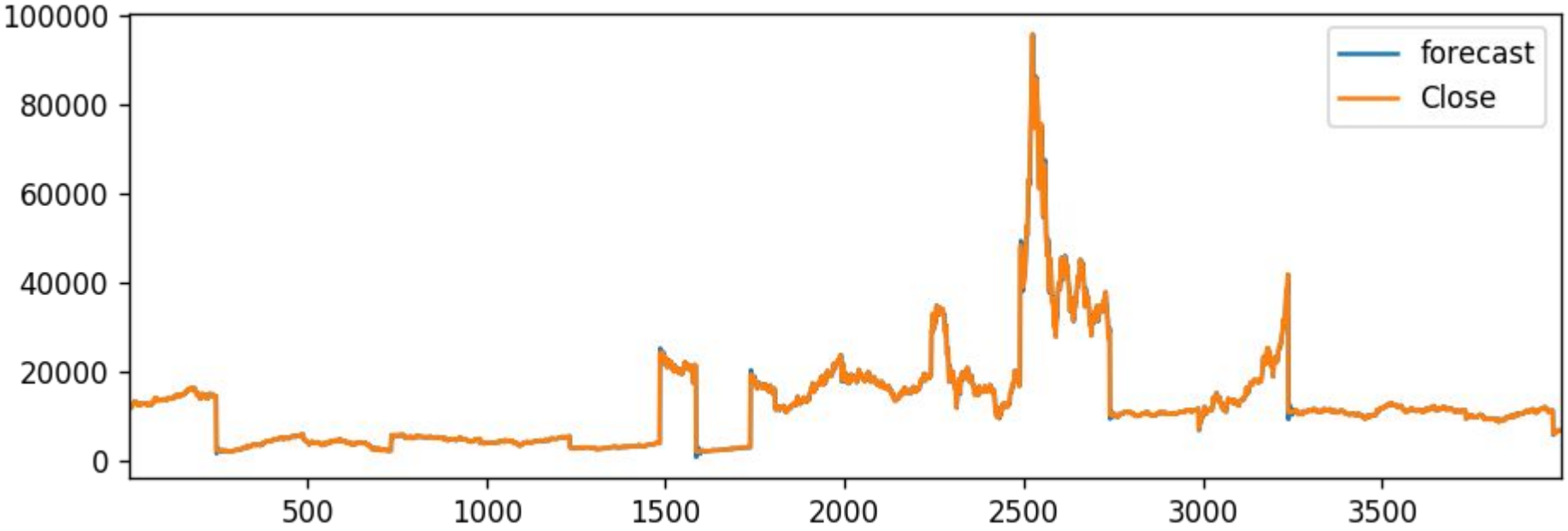
- ‘q’ is the number of lagged forecasted errors.

```
plot_acf(x_train['Close'].diff().dropna())
```

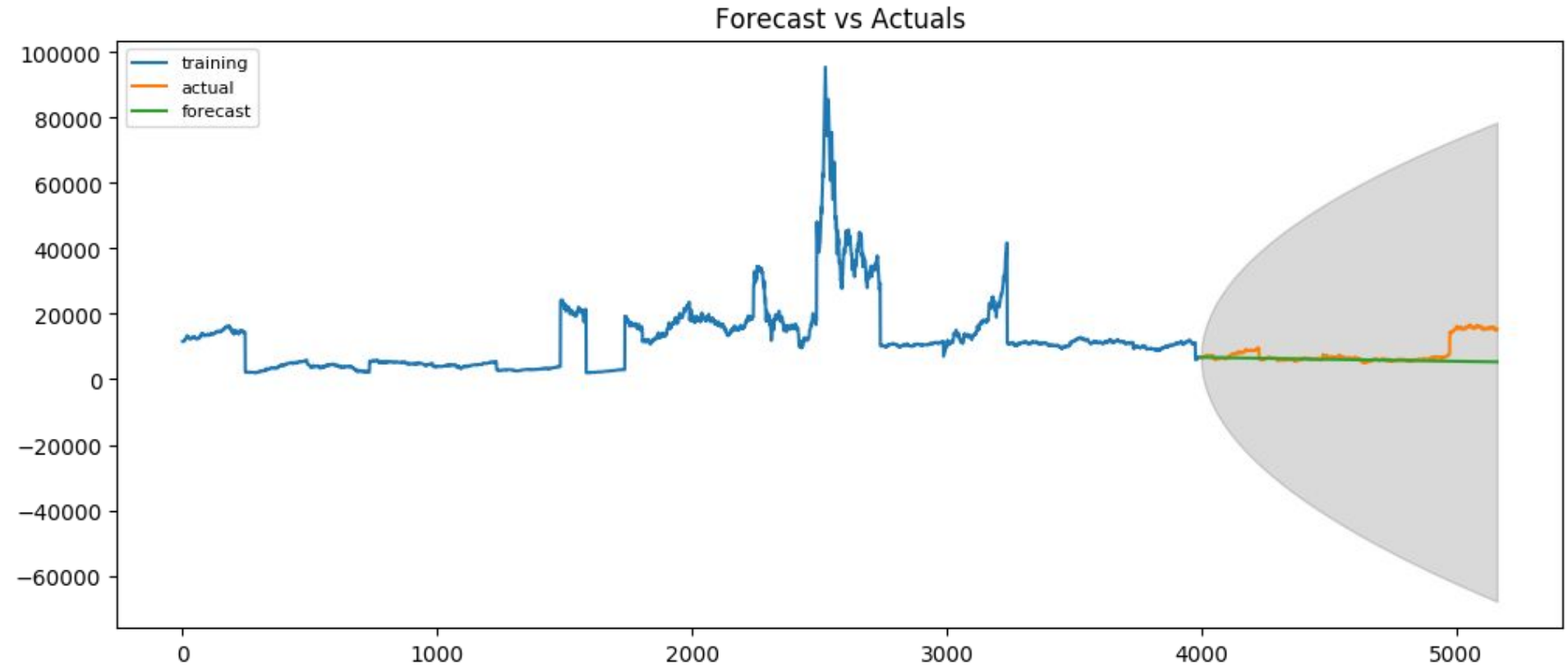
- 2 lagged forecasted errors have Autocorrelation value > threshold.
- Hence, q=2.



ARIMA- Training ARIMA (2,1,2)



ARIMA - Testing ARIMA (2,1,2)



LSTM

- A deep learning model used for Time Series Forecasting.
- We tried this model instead of RNN because it takes care of both short term and long term memory.
- This model despite its merits did not work well for our data. The predictions made were very different from the actual values.
- Hence, we are going to go forward with ARIMA.

Inclusion of NLP

Can we use NLP to further increase the accuracy of our product?

The answer is yes and no. The reasons are as follows:

- How to weigh sentiment? And what variables will define a positive and negative sentiment?
- Even if we use it, textual information needs to converted to numeric values in order to be used.
- Which sentiment to trust? Can we make predictions using NLP?
- How much data do we need to create a trustworthy sentiment analyzer for tweets?

Solution?

One can identify the trend in the sentiment and match it with the spike in the closing price of stock, rather than explicitly feeding the sentiment as an input for the prediction. The so called Sentiment Index can then be used to identify whether the closing price of the stock will decrease or increase, thereby aiding in the decision to be taken by the prospective buyer.

Thank You