Predicting the directionality of S&P500

Capstone Project

Yemitan Isaiah Olurotimi

November 3rd, 2019

I. Definition

Domain Background

S&P 500 is a stock market index that measures the stock performance of 500 large companies listed on stock exchanges in the united states. The index is widely considered as the best indicator of the day to day performance of the U.S stocks.

Over 50 years for data was analyzed, It was deduced that there is approximately 52% chance that the stock price will increase and 48% that it will decrease for any given day.

This project aims to use some machine learning techniques to predict the directionality of stock prices.

Project Statement

Stock prices are usually described as a statistical process called “random walk” which means each days closing value is unpredictable and random.

The problem to be solved with this project is to apply machine learning to predict the directionality of a stock price for any given day using over 50 years of historical data.

For solving the problem, Yahoo finance data was used. The data contains the following columns.

- Date
- Open
- High
- Low

- Close
- Adj Close
- Volume

We used LSTM and Arima to predict the closing price for each day or month.

Metrics

In this project, the model was evaluated using Mean squared error and mean absolute percentage error.

Mean squared error measured the average of the squared of the errors (Average of the difference between the estimated values and actual value)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

Mean absolute percentage error is a measure of prediction accuracy of a forecasting method. It is used for regression problems.

$$\mathrm{M} = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|,$$

II. Analysis

Data Exploration

The dataset for S&P 500 is publicly available and will be obtained from yahoo finance.
The time period for the data set is between October 1969 to October 2019 which is 50 years.
This is data contains 12,614 rows and 7 columns which are;

- Date: Date for the given data.
- Open: The stock opening price for the given date.
- High: Highest stock price for the given date.
- Low: Highest stock price for the given date.
- Close: The stock closing price for the given date.

- Adj Close: Adjusted closing price for the given date.
- Volume: Volume of stock traded for the given date.

Link: https://finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC

Exploratory Visualization

The important columns that were used are; Date, Open and Adj Close. Adj close was preferred over close as it is the final closing price for the given date.

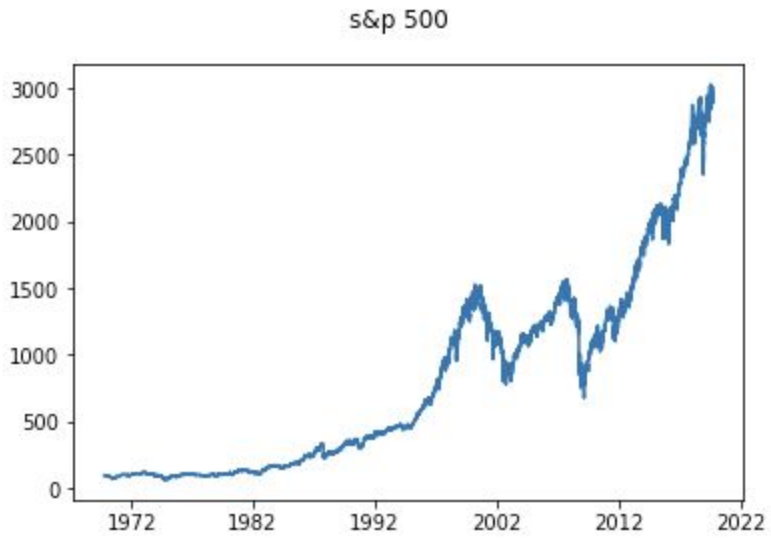
Below is a sample of the data

	Date	Open	High	Low	Close	Adj Close	Volume
0	1969-10-13	93.559998	94.860001	93.199997	94.550003	94.550003	13620000
1	1969-10-14	94.550003	96.529999	94.320000	95.699997	95.699997	19950000
2	1969-10-15	95.699997	96.559998	94.650002	95.720001	95.720001	15740000
3	1969-10-16	95.720001	97.540001	95.050003	96.370003	96.370003	19500000
4	1969-10-17	96.370003	97.239998	95.379997	96.260002	96.260002	13740000

Description of the data

	Open	High	Low	Close	Adj Close	Volume
count	12614.000000	12614.000000	12614.000000	12614.000000	12614.000000	1.261400e+04
mean	804.955881	809.684315	799.922962	805.128490	805.128490	1.321124e+09
std	743.595826	747.094008	739.739642	743.668664	743.668664	1.709035e+09
min	62.279999	63.230000	60.959999	62.279999	62.279999	6.650000e+06
25%	131.232502	132.649994	129.915001	131.262497	131.262497	5.667250e+07
50%	467.059998	468.914993	465.544999	467.089997	467.089997	3.046950e+08
75%	1277.127533	1283.952484	1267.552551	1277.345001	1277.345001	2.606752e+09
max	3024.469971	3027.979980	3014.300049	3025.860107	3025.860107	1.145623e+10

Data Chart



Algorithms and Techniques

The task for this project is to predict the directionality of stock prices of the next N day(s)/ month in the future using historical data as its input.

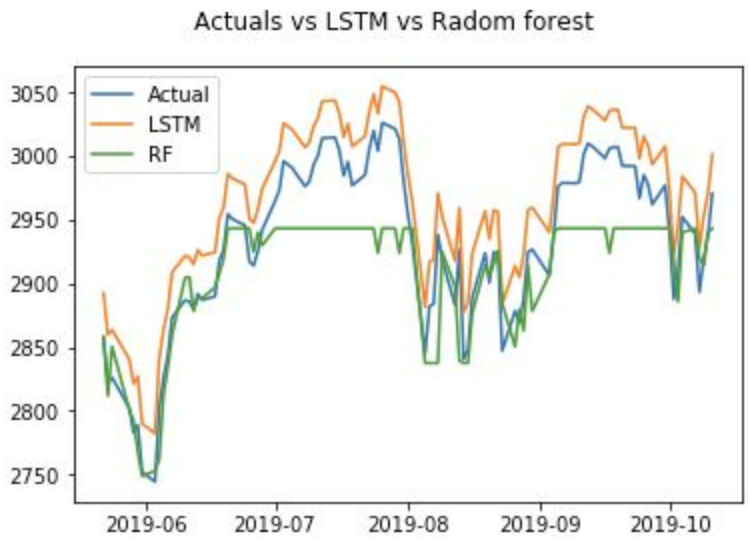
Multiple ML algorithms will be used to predict the closing stock price. The algorithms that will be used are Long short term memory (LSTM), Autoregressive integrated moving averages (Arima) and random forest for the benchmark. These models will be used individually and compared to see which performs best.
I was not able to use fbProphet due to some technical reasons with my computer.

Benchmark

The benchmark model includes multiple sources such as trading economics, forecast chart, and the use of random forest since stock prices are perceived as a random walk.

- Trading Economics https://tradingeconomics.com/spx:ind/forecast
- Forecast Chart https://www.forecast-chart.com/index-sp-500.html
- Random forest
- Financial forecast centre https://www.forecasts.org/data/data/SP500.htm

From the diagram below it is seen that random forest was used to compare with the actual data and predicted data from LSTM



III. Methodology

Data Preprocessing

We preprocessed the data by splitting the data into the training and testing data. For LSTM, MinMaxScaler was used to scale and normalize the data.
For Arima, we took the logarithm of the input and applied time-shifting.

Implementation

For the implementation of the project, LSTM and Arima were used. The LSTM model includes 3 layers with 128 unit, sigmoid activation function, dropout of 0.2 and mean squared error. Before the data was applied to the model for Arima, we needed to check if it was stationary. We applied log, time-shifting to be able to check if it was stationary. Mean squared error and mean absolute error was used for both LSTM and Arima.

Refinement

The refinement process was mostly trial and error. To improve the model, I tried different layers, activation functions, number of epochs, changed the hyperparameters to see which is the most appropriate for the model. I played around with the dropouts,

numbers of layers, normalization, and some parameters. For Arima, I made sure the data was stationary thereby applying log and time-shifting before applying the model.

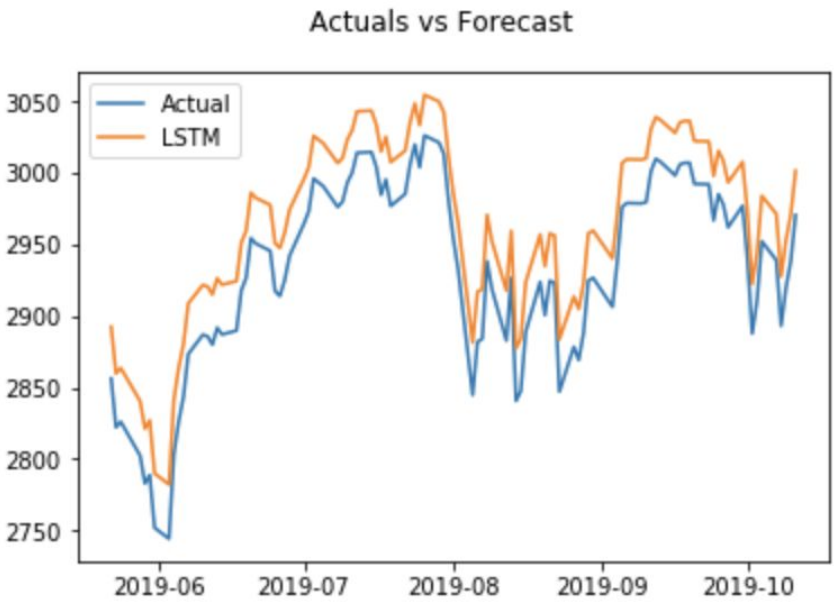
IV. Results

Model Evaluation and Validation

The best model which is LSTM with the following hyperparameters

- Dropout: 0.2
- Activation: Sigmoid

From the metrics using Mean squared error and mean absolute percentage error, It is seen that LSTM performed well.
LSTM aligns well with the solution expected and performs better with the benchmark.
See the chart below.



Justification

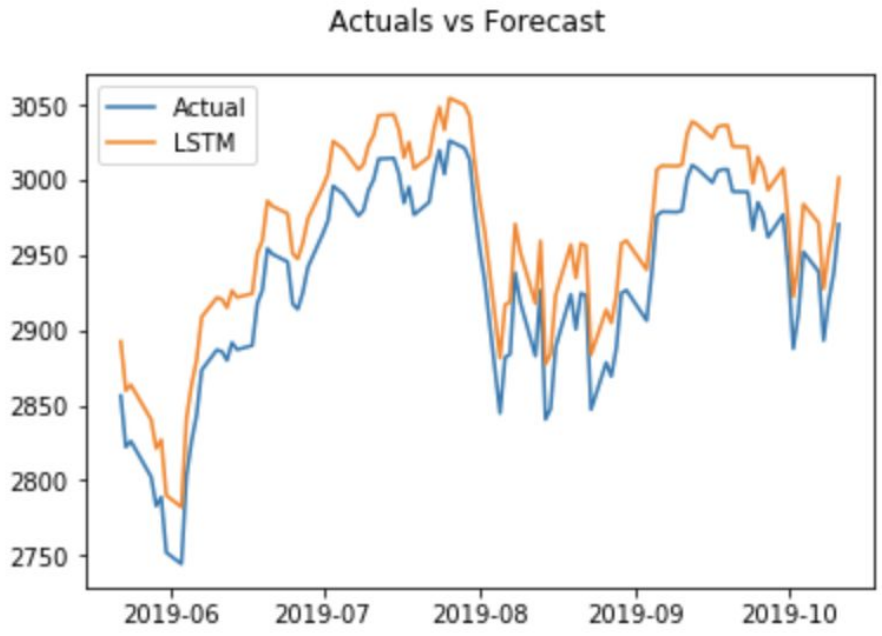
The model outperformed the benchmark and the other model used. Although, the solution outperformed the benchmark, but I do not believe that this solution is significant enough to have solved the problem this is because there are various factors that determine the dimensionality of

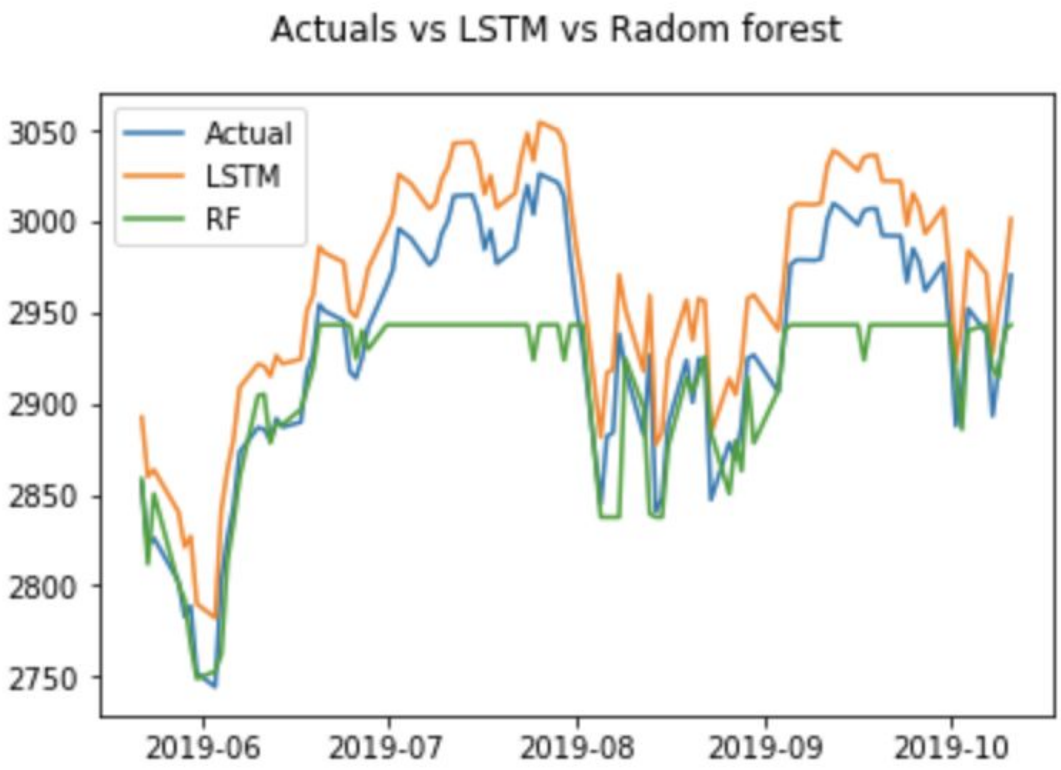
the stock price apart from historical data. Hence to make the solution more robust, I need to take into consideration the other factors.

V. Conclusion

Free-Form Visualization

The chart below shows the forecast in comparison with the actual chart and the benchmark. The model predicts the upward and downward trends and it is not accurate, but the dimensionality is often predicted.





Reflection

The project was an interesting one for me as I was able to try out some ML techniques like LSTM and Arima.
In the project, I checked if there are null/NAN values in the project, Adj close was used in place of close. Then, important features for the model was harvested and used.
The data were also normalized to be used by the model. I experimented with different layers, hyperparameters and activation functions.
I found all aspect of the project in general interesting as I was learning new techniques and having hands-on experience with the techniques.
The difficult aspect of the project was using Arima. I had to leverage some resources to be able to understand Arima and implement it op the project.
It is important to note that the stock price cannot only be predicted using historical data as a lot more factors affect the price. The solution fits the problem to some extent, but not entirely as the stock price is affected by other factors.

Improvement

As I said before, the stock price is affected by other factors aside historical data hence, to improve this project I will use sentiment analysis on news site and twitter to get real-time information that can affect the price and use it in the project accordingly.
I can also use reinforcement learning to train an agent to predict the price.

Reference

https://en.wikipedia.org/wiki/S%26P_500_Index
https://en.wikipedia.org/wiki/Stock_market_prediction
https://www.fool.com/knowledge-center/what-is-the-sp-500.aspx
https://finance.yahoo.com/quote/%5EGSPC/
https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/
https://towardsdatascience.com/machine-learning-techniques-applied-to-stock-price-prediction-6c1994da8001
https://machinelearningmastery.com/time-series-forecasting-performance-measures-with-python/
https://otexts.com/fpp2/accuracy.html https://en.wikipedia.org/wiki/Mean_squared_error

https://www.kaggle.com/myonin/bitcoin-price-prediction-by-arima

https://towardsdatascience.com/machine-learning-part-19-time-series-and-autoregressive-integrated-moving-average-model-arima-c1005347b0d7

https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/

https://ademos.people.uic.edu/Chapter23.html

https://www.kaggle.com/someadityamandal/bitcoin-time-series-forecasting

https://www.kaggle.com/myonin/bitcoin-price-prediction-by-arima

https://machinelearningmastery.com/time-series-forecasting-python-mini-course/

https://www.youtube.com/watch?v=7vunJlqLZok

https://www.kaggle.com/kp4920/s-p-500-stock-data-time-series-analysis

https://stackabuse.com/time-series-analysis-with-lstm-using-pythons-keras-library/

https://www.datacamp.com/community/tutorials/lstm-python-stock-market