Using Machine Learning and Deep Learning Techniques to Estimate Stock Expenditure

California Polytechnic University, Pomona – Department of Computer Science

Rene B. Dena

Abstract

Predicting how the stock market will perform is one of the most difficult things to do. There are so many factors involved in the prediction – physical factors vs. physiological, rational and irrational behavior, etc. All these aspects combine to make share prices volatile and very difficult to predict with a high degree of accuracy.

In this project, I examine whether or not machine learning can be a game changer in this domain. Using features like the latest announcements about an organization, their quarterly revenue results, etc., machine learning techniques have the potential to unearth patterns and insights we didn't see before, and these can be used to make unerringly accurate predictions.

In this project, I worked with historical data about the stock prices of a publicly listed company. I implemented a mix of machine learning algorithms to attempt and predict the future stock price of the given company, first starting with a simple algorithm like averaging, and then moving on to advanced techniques like Auto Regressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM).

Introduction

The core idea behind this project is to showcase how such algorithms can be implemented and perform differently. In this project, the main goal is to describe and implement these techniques.

The first step is understanding the problem. It is important to establish what we're aiming to solve. Broadly, stock market analysis is divided into two parts – Fundamental Analysis and Technical Analysis. Fundamental Analysis involves analyzing the company's future profitability on the basis of its current business environment and financial performance whereas Technical Analysis, on the other hand, includes reading the charts and using statistical figures to identify the trends in the stock market. As you might have guessed, my main focus of this project will involve the technical analysis part.

Data

All data used in this project was obtained from a dataset provided by Quandl where one can find historical data for various stocks. Quandl offers essential financial and economic data alongside a suite of unique, alpha-generating alternative datasets. With their unrivaled consumption experience, they have cemented a reputation for understanding and delivering what professional quantitative analysts need and want. For this particular project, I have used the data for 'Tata Global Beverages'. First, the data was loaded and then the target variable was defined from the data.

```
data.head()
```

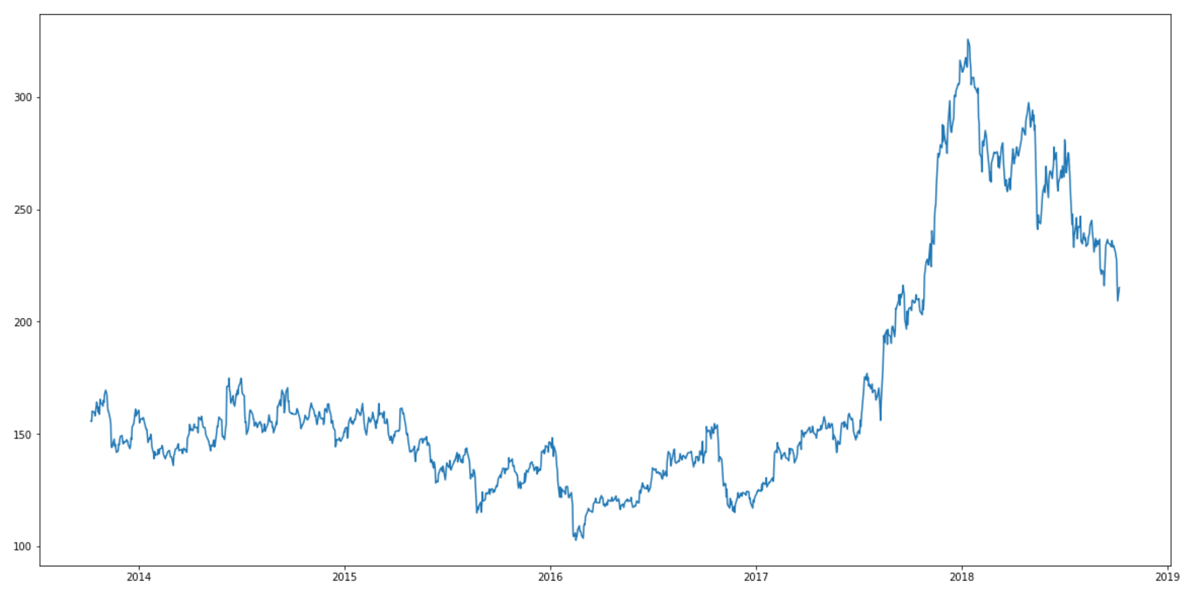
	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-10-08	208.00	222.25	206.85	216.00	215.15	4642146.0	10062.83
1	2018-10-05	217.00	218.60	205.90	210.25	209.20	3519515.0	7407.06
2	2018-10-04	223.50	227.80	216.15	217.25	218.20	1728786.0	3815.79
3	2018-10-03	230.00	237.50	225.75	226.45	227.60	1708590.0	3960.27
4	2018-10-01	234.55	234.60	221.05	230.30	230.90	1534749.0	3486.05

There were multiple variables in the dataset – date, open, high, low, last, close, total trade quantity, and turnover.

- The columns *Open* and *Close* represent the starting and final price at which the stock is traded on a particular day.
- *High*, *Low* and *Last* represent the maximum, minimum, and last price of the share for the day.
- *Total Trade Quantity* is the number of shares bought or sold in the day and *Turnover (Lacs)* is the turnover of the particular company on a given date.

One important thing to note about the data is that it accounted for when the market was closed on weekends and public holidays. As can be seen by the table previously refrenced, some date values are missing – 2/10/2018, 6/10/2018, 7/10/2018. Of these dates, the 2nd is a national holiday while 6th and 7th fall on a weekend.

Another note to keep in mind is the fact that the profit or loss calculation is usually determined by the closing price of a stock for the day, hence I chose to consider the closing price as my target variable. Plotting the target variable gives us an understanding for what shape the data will take.

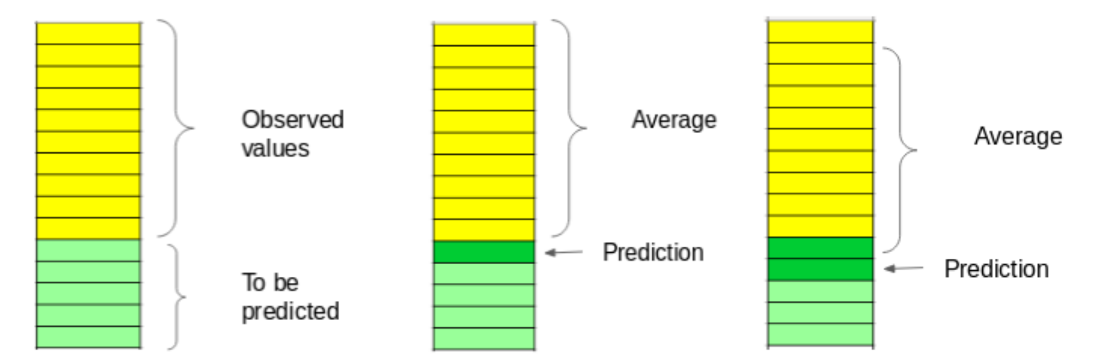


So this is the data for in which I will explore these variables and use different techniques to attempt and predict the daily closing price of the stock.

Method

Moving Average is easily one of the most common things we use in our day-to-day lives. For instance, calculating the average marks to determine overall performance, or finding the average temperature of the past few days to get an idea about today's temperature – these all are routine tasks we do on a regular basis. So for this project, this was my starting point to use on the dataset for making predictions.

The predicted closing price for each day will be the average of a set of previously observed values. Instead of using the simple average, I used the moving average technique which uses the latest set of values for each prediction. In other words, for each subsequent step, the predicted values are taken into consideration while removing the oldest observed value from the set. Here is a simple figure that can help one visualize with more clarity:

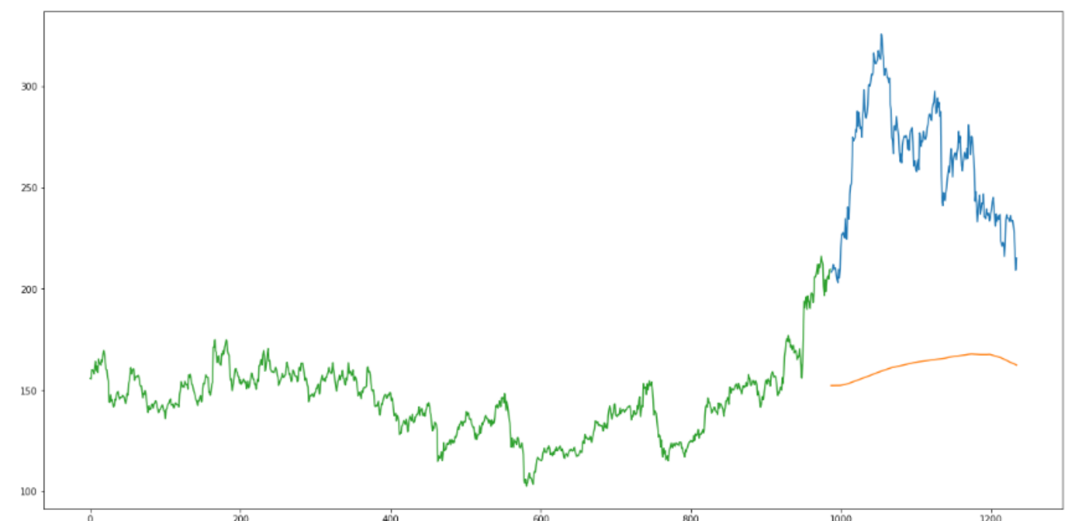


I first implemented this technique on the dataset. The first step is to create a dataframe that contains only the *Date* and *Close* price columns, then split it into train and validation sets to verify the predictions.

Note; before starting, I had split the data into its train and validation components. In doing so, this prevented us from using random splitting that will could potentially destroy the time component. To do this, I have set the last year's data into validation and the 4 years' data before that into train.

Final step was to create predictions for the validation set and check the RMSE using the actual values.

Doing so gave me a *RMSE result of:* 104.5142. However, just checking the RMSE does not help us in understanding how the model performed. To get a more intuitive understanding, I graphed a chart to help better visualize. So here is a plot of the predicted values along with the actual values.



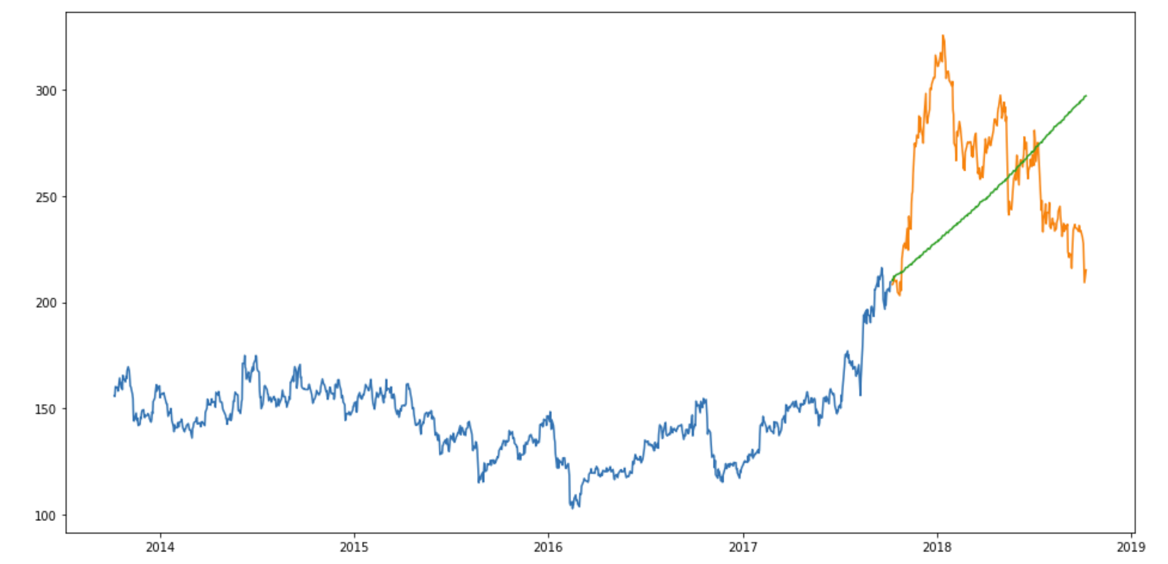
As showcased by the graph, the results are were not very promising. However, the predicted values are of the same range as the observed values in the train set as there is an increasing trend initially and then a slow decrease.

Next, I looked at some time series forecasting techniques to find out how they perform on the stock market data and whether or not they can be used to predict stock prices.

First one being Auto Regressive Integrated Moving Average (ARIMA). ARIMA is a very popular statistical method for time series forecasting. ARIMA models take into account the past values to predict the future values. There are three important parameters in ARIMA:

i. p (past values used for forecasting the next value)
ii. q (past forecast errors used to predict the future values)
iii. d (order of differencing)

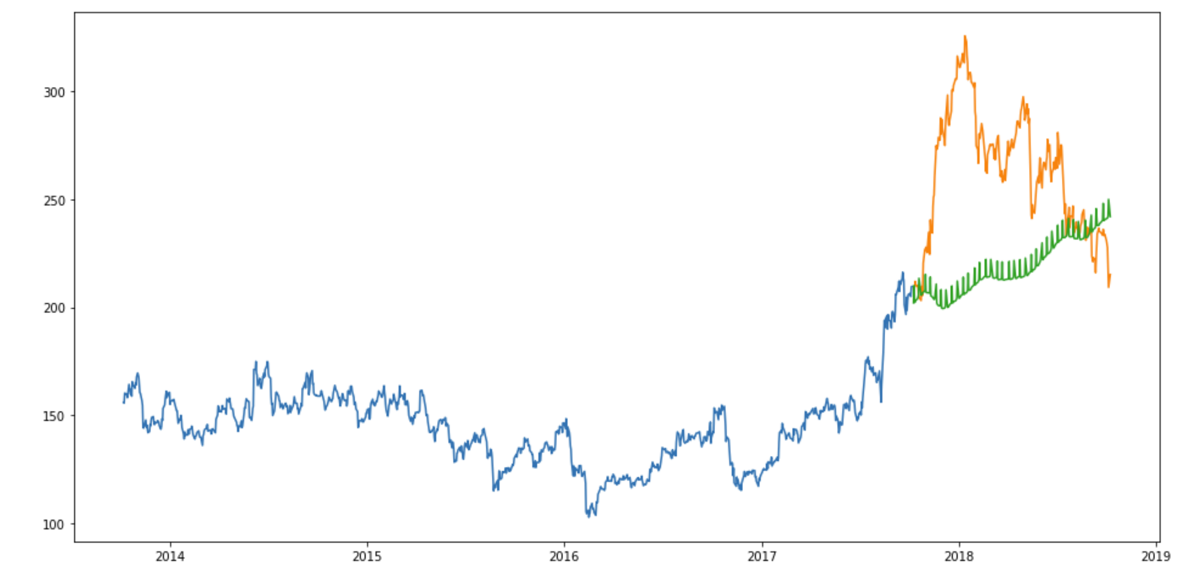
Because parameter tuning for ARIMA consumes a lot of time, I used Auto ARIMA which automatically selects the best combination of (p,q,d) that provides the least error. Fitting the ARIMA model gave me a *RMSE result of: 44.9552.*



As mentioned above, an auto ARIMA model uses past data to understand the pattern in the time series. Using these values, the model captured an increasing trend in the series. Although the predictions using this technique are far better than that of the previously implemented *Moving Average* model, these predictions are still not close to the real values. As evident from the plot, the model has captured a trend in the series, but does not focus on the seasonal part. For the next model, I implemented a time series model that takes both trend and seasonality of a series into account.

That model is commonly know as *Prophet.* There are a number of time series techniques that can be implemented on the stock prediction dataset, but most of these techniques require a lot of data preprocessing before fitting the model. Prophet, designed and pioneered by Facebook, is a time series forecasting library that requires no data preprocessing and is extremely simple to implement. The input for Prophet is a dataframe with two columns: date and target (ds and y). Prophet tries to capture the seasonality in the past data and works best when a dataset is large.

Implementation of the Prophet model gave me a *RMSE result* of: 56.9349.

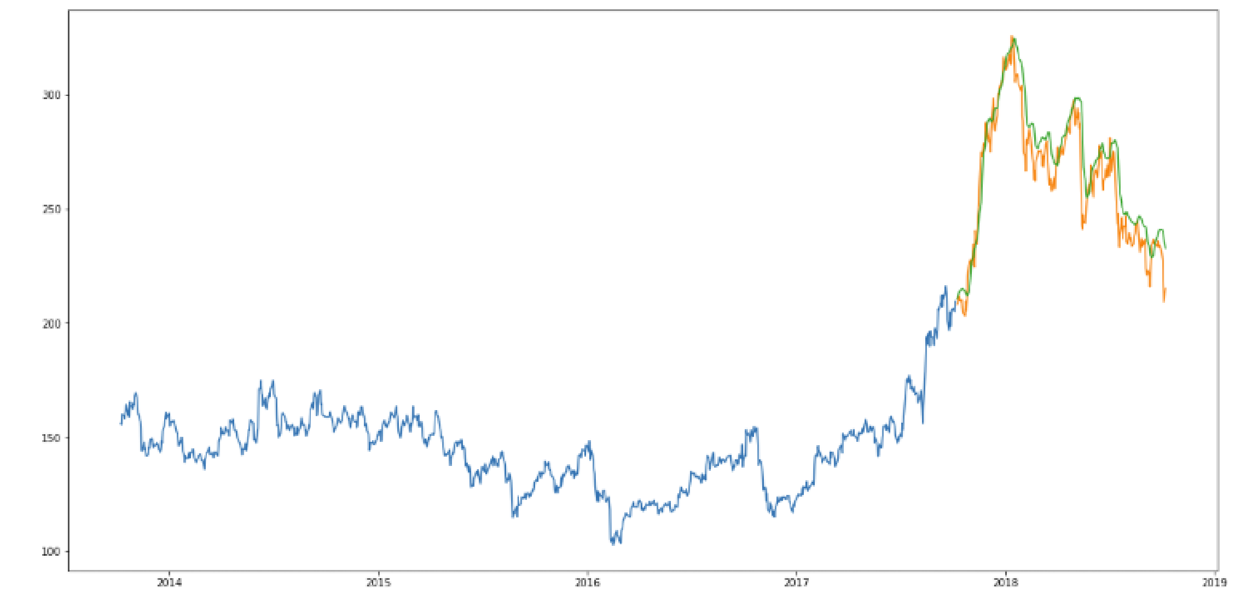


Prophet (like most time series forecasting techniques) tries to capture the trend and seasonality from past data. This model usually performs well on time series datasets, but seems to preform poorly in this case. As it turns out, stock prices do not have a particular trend or seasonality. It highly depends on what is currently going on in the market and thus the prices rise and fall. Hence forecasting techniques like ARIMA and Prophet would not show good results for this particular problem.

For this reason alone, I decide implement another advanced technique. This next model technique is known as Long Short Term Memory (LSTM). LSTMs are widely used for sequence prediction problems and have proven to be extremely effective. The reason they work so well is because LSTM is able to store past information that is important, and forget the information that is not. LSTM has three gates:

i. The input gate: The input gate adds information to the cell state
ii. The forget gate: It removes the information that is no longer required by the model
iii. The output gate: Output Gate at LSTM selects the information to be shown as output

Implementation of LSTM gave me a *RMSE result of*: 11.7723.



As shown by the RMSE and plotted graph, LSTM has easily outshone all of the previous implemented algorithms. The LSTM model can be tuned for various parameters such as changing the number of LSTM layers, adding dropout value or increasing the number of epochs.

Results

Moving Average RMSE:	104.5142
Auto Regressive Integrated Moving Average (ARIMA) RMSE:	44.9552
Prophet RMSE:	56.9349
Long Short Term Memory (LSTM):	11.7723

Conclusion

Being able to predict how the stock market will perform has been proven to be difficult time and time again. Because so many factors play a role in how much a stocks value will increase or decrease at any given period, there is always some uncertainty in predicting a stocks value with a high degree of accuracy. By being able to gather data from a publicly listed company, we were able implement machine and deep learning techniques on the given data to see whether or not they would be useful in predicting the given company's stock value. As shown by the results, some techniques out preformed others in being able to predict the stock prices for the given company using the same train and validation data. For this data in paticular, Long Short Term Memory (LSTM) seemed to perform the best as it was the only time series forecasting technique that stored important information, and disregarded irrelevant information in order to predict the stock price for the given company.

When all is said and done, one must keep in mind that the predictions from LSTM is definitely not enough to identify whether a stock price will increase or decrease. As mentioned earlier, stock price is affected by the news of a given company and other factors like demonetization or merger/demerger of the companies. There are certain intangible factors as well which can often be impossible to predict beforehand.

Acknowledgements

D. Peterson, Ralph Roskies, J. Ray Scott, Nancy Wilkins-Diehr, "XSEDE: Accelerating Scientific Discovery", Computing in Science & Engineering, vol.16, no. 5, pp. 62-74, Sept.-Oct. 2014, doi:10.1109/MCSE.2014.80

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562.

Special thanks to Quandl who provided me the dataset and inspired me to make new discoveries and incorporate them into every day strategies. Helped me realize the new and better ways to understand the complex information that creates markets and market movement.

I would also like to express my special thanks of gratitude to Dr. Hao Ji who provided me with the tools needed to complete such project.

References

Locke, Benji L. "How To Identify Patterns in Time Series Data: Time Series Analysis." *Stat Soft*, 6 May 2017, www.statsoft.com/textbook/time-series-analysis. Accessed 3 Dec. 2018.

Williams, Michelle J. "A Beginner's Guide to LSTMs and Recurrent Neural Networks." *Sky Mind AI*, 12 Apr. 2015, skymind.ai/wiki/lstm. Accessed 9 Dec. 2018.

Hayashi, Hideaki J. "Is Prophet Really Better than ARIMA for Forecasting Time Series Data?." *Learn Data Science*, 17 Oct. 2017, blog.exploratory.io/is-prophet-better-than-arima-for-forecasting-time-series-fa9ae08a5851. Accessed 3 Dec. 2018.

Singh, Aishwarya W. "A Complete Tutorial on Time Series Modeling." *Analytics Vidhya*, 25 Oct. 2018, www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/. Accessed 5 Dec. 2018.

Brownlee, Jason B. "LSTM Model Architecture for Rare Event Time Series Forecasting." *Machine Learning Mastery*, 2 Nov. 2018, machinelearningmastery.com/lstm-model-architecture-for-rare-event-time-series-forecasting/. Accessed 4 Dec. 2018.