Technocolabs Data Science Internship

STOCK PRICE TREND PREDICTION BASED ON NEWS

Major Project Report

*Image by **George Fullerton** (2018) via The Cash Academy*

Abstract

Stock prices vary every day and predicting it has always been the hottest topic in the industry. There have been a lot of predictions made using machine learning to predict the stock price using price. But along with that, detecting how news headlines affect the stock prices is something our project has aimed for. Since, everyday news has affected the stock market in one way or the other, like some tweets on Twitter were responsible for the rise and fall of some cryptocurrency. Similarly, this project aimed to find the relation between news headlines and stock price. In this project, we have made use of sentimental analysis and the LSTM model to predict stock prices based on news headlines. The stock price data and news headlines were used from 2008 to 2016 to feed and test the model. Later, it was deployed into a website for ease of use. Using it in real-time has given significant accuracy. And with that, we were able to conclude that news articles do play an important role in the stock price variance.

Introduction

Prediction of stock prices and understanding their change has been an area of interest for analysts and researchers for a long time. However, stock prices have a highly volatile nature that stems from various socio-economic factors, making them very difficult to predict. Studies in sentiment analysis have found that there is a strong correlation between changes in stock prices and the publication of news.

Our method of predicting the change in stock price utilizes sentiment analysis on news articles to determine the future stock prices. We have used Deep Learning models- LSTM and ARIMA for the training and testing of the dataset. Out of the two models, LSTM presents better results in comparison, which is why it has been used as the model for training, testing, and deployment.

Post-training of the model, Flask API of python was used to develop the web application which was then deployed using Heroku.

Data

We were provided with three datasets.

1. Stock Data (Numerical)
2. Reddit Data (Textual)
3. News Data (Textual)

Out of these three, we found that Reddit data was already included in News data. So it was no longer needed. We worked on Stock price data and News data. Both the datasets had 1989 rows of data. Later, we split this data into an 80:20 ratio for training and testing.

Preprocessing

Stock Price Data

This data included Open, Close, Volume, High, Low, Adj Close columns. Out of Open and Close prices were being provided as inputs along with the news. So we removed the remaining columns and didn't include them in training as well.

The figure below shows the initial state of the dataset.

	Date	Open	High	Low	Close	Volume	Adj Close
0	2016-07-01	17924.24023	18002.38086	17916.91016	17949.36914	82160000	17949.36914
1	2016-06-30	17712.75977	17930.60938	17711.80078	17929.99023	133030000	17929.99023
2	2016-06-29	17456.01953	17704.50977	17456.01953	17694.67969	106380000	17694.67969
3	2016-06-28	17190.50977	17409.72070	17190.50977	17409.72070	112190000	17409.72070
4	2016-06-27	17355.21094	17355.21094	17063.08008	17140.24023	138740000	17140.24023
...	...	...	...	...	...	...	...
1984	2008-08-14	11532.07031	11718.28027	11450.88965	11615.92969	159790000	11615.92969
1985	2008-08-13	11632.80957	11633.78027	11453.33984	11532.95996	182550000	11532.95996
1986	2008-08-12	11781.70020	11782.34961	11601.51953	11642.46973	173590000	11642.46973
1987	2008-08-11	11729.66992	11867.11035	11675.53027	11782.34961	183190000	11782.34961
1988	2008-08-08	11432.08984	11759.95996	11388.04004	11734.32031	212830000	11734.32031

News Data

In this step, the Combined_News_DJIA dataset containing the news headlines was preprocessed and cleaned. The textual data was cleaned by removing the white spaces, removing punctuation and the letter 'b' from each sentence, imputing the missing values with mode, changing the data, renaming the columns for easy access, converting all the sentences into lower cases, and combining the news into 1 column.

	Date	Label	headlines
0	2008-08-08	0	Georgia downs two Russian warplanes as countri...
1	2008-08-11	1	Why wont America and Nato help us If they wont...
2	2008-08-12	0	Remember that adorable 9yearold who sang at th...
3	2008-08-13	0	US refuses Israel weapons to attack Iran repo...
4	2008-08-14	1	All the experts admit that we should legalise ...

The above figure shows the cleaned news headlines

Sentiment Analysis

Sentiment Analysis can help us decipher the mood and emotions of the general public and gather insightful information regarding the context. Sentiment Analysis is a process of analyzing data and classifying it based on the need of the research. These sentiments can be used for a better understanding of various events and the impact caused by them.

In this project, we have calculated the subjectivity and polarity of the cleaned news headlines using textblob python library and also calculated the sentiment scores (i.e., Compound, positive, negative, and neutral values) using Sentiment Intensity Analyzer from the vaderSentiment python library.

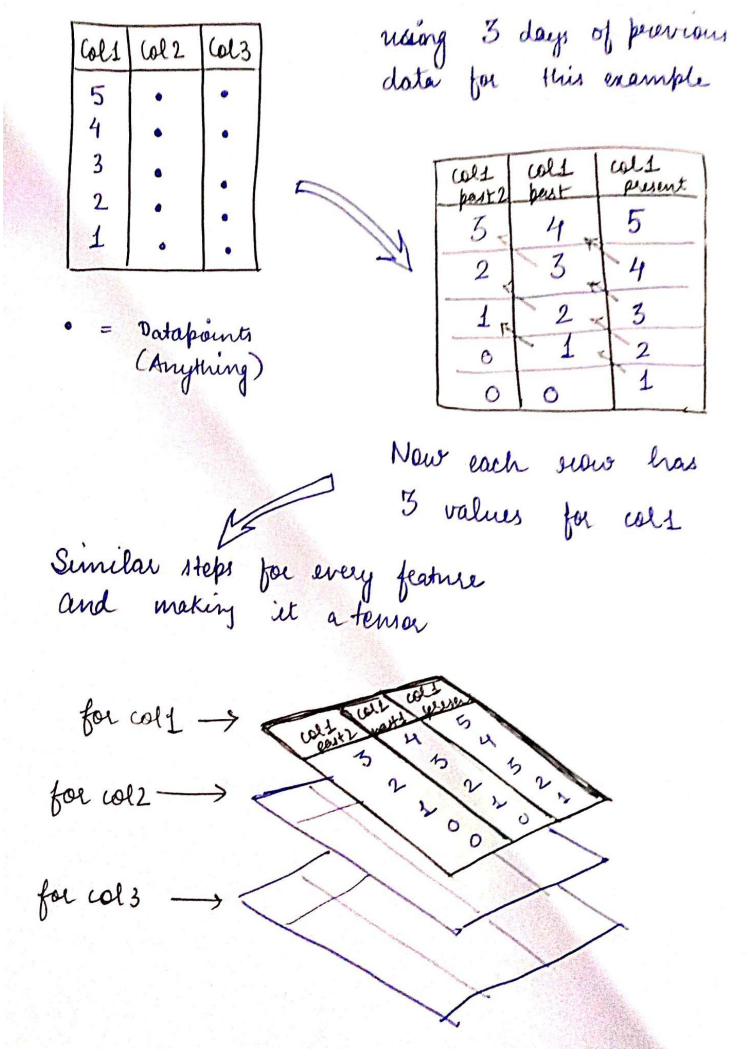
Model Building

We trained two models, one using LSTM and the other using ARIMA.

LSTM

For LSTM, we created a tensor that included 60 days of previous data in the same row using the data. And this was done for every feature. The diagram on right depicts the process.

We fed this dataset into the LSTM model made of 4 Sequential Layers. The RMSE given by the model for close prices is 249.8.



ARIMA

The best model predicted by autoARIMA is SARIMAX(1,0,3). It has an AIC of 528.6 and the P Values are significant. For training the ARIMA model, only close prices are used.

Performance of SARIMAX:

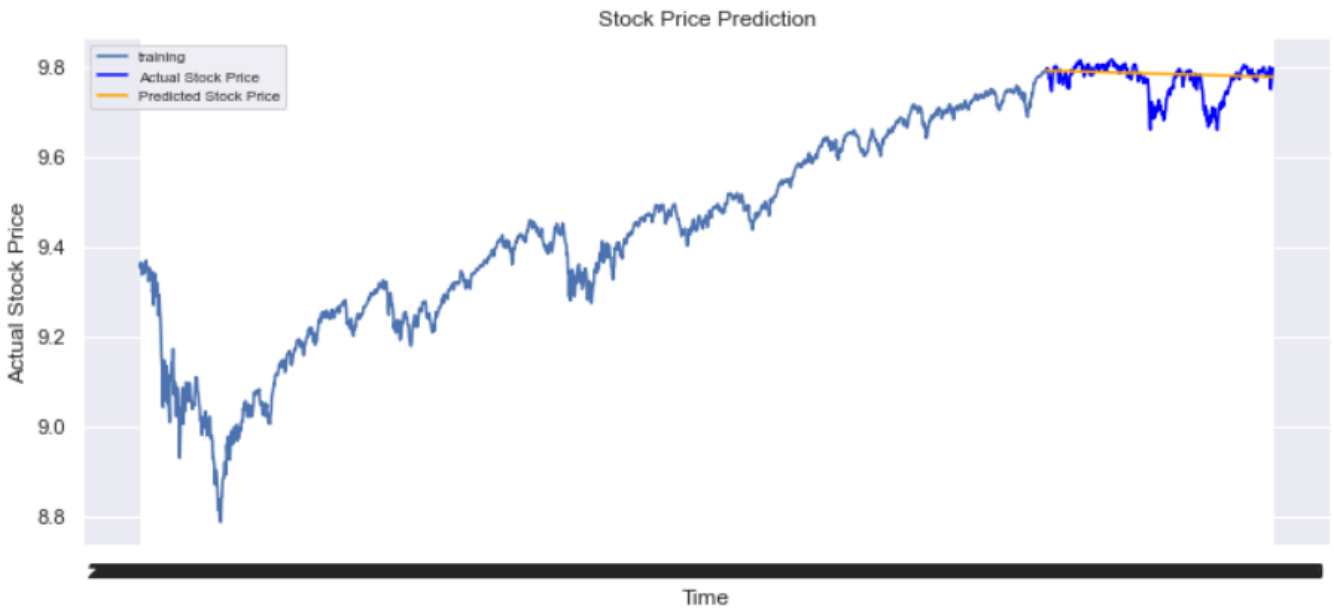
- MSE: 0.001474261472687624
- MAE: 0.025327653159186156
- RMSE: 0.038396112728863896

Evaluation

The numbers in the ARIMA model look impressive than LSTM but it only uses close prices and ignores the news data.

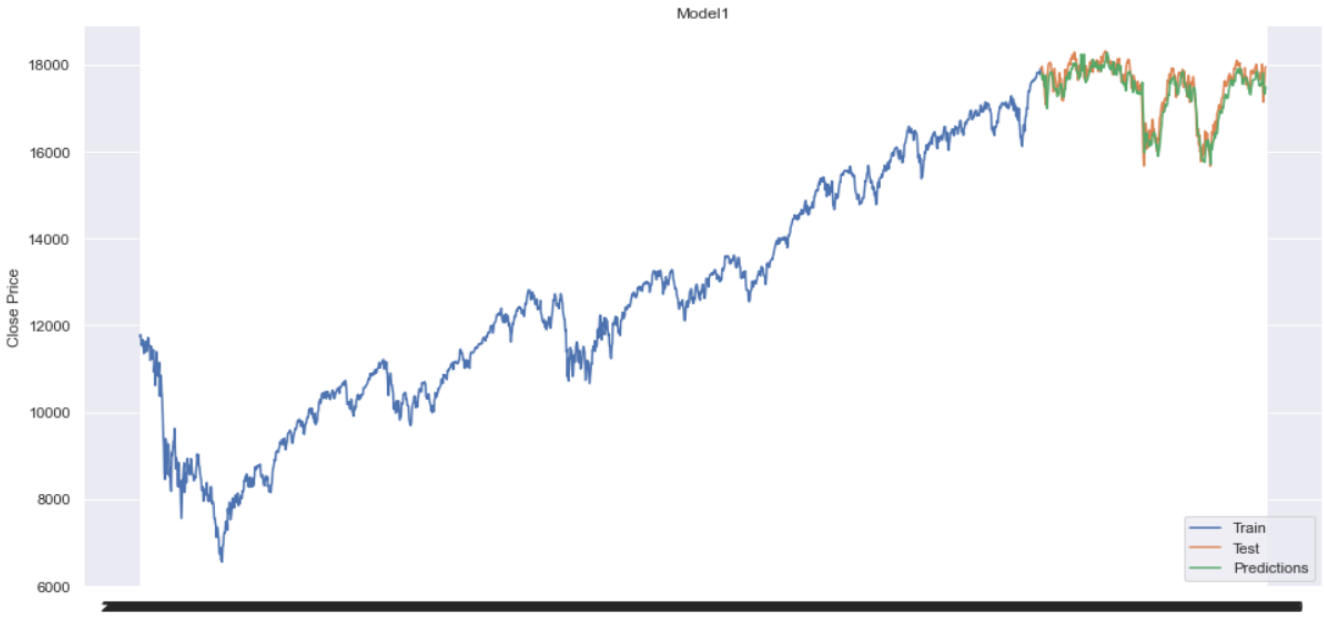
Comparing the graphs:

ARIMA results

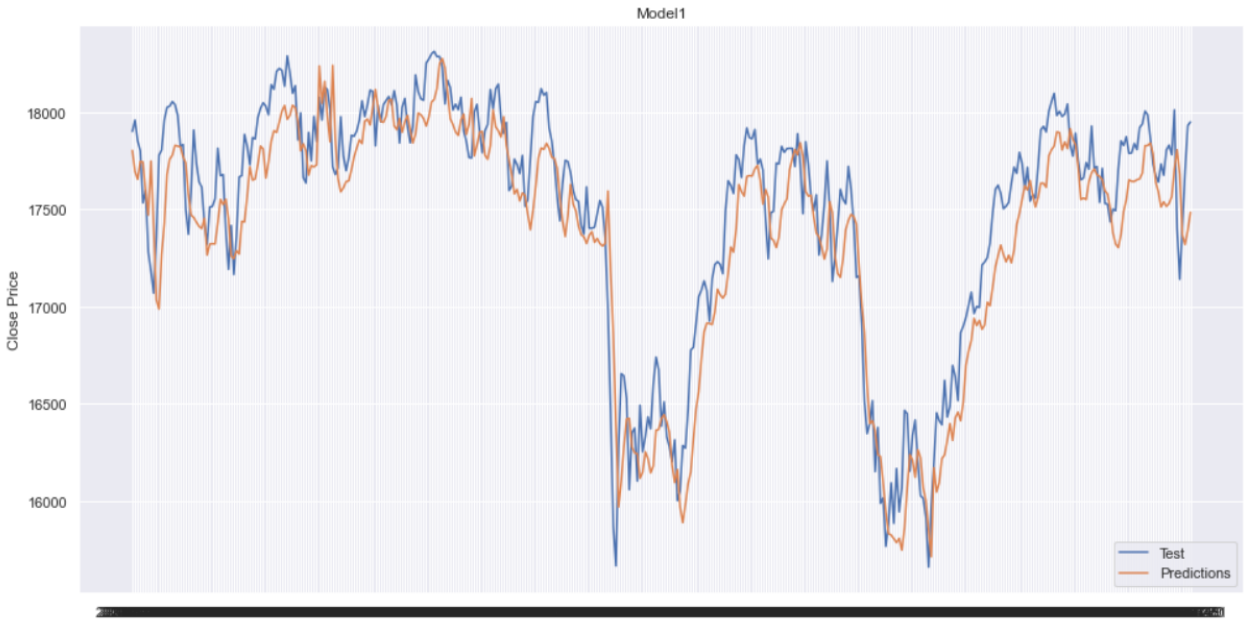


LSTM results:

Graph for the whole dataset



Graph for the test dataset



From the Graph, it was clear that the LSTM model was predicting close to the original data. So, the final model used for predictions is the **LSTM model.**

Deployment

Flask

We used the **FLASK** framework to build our web application. We have two main URLs used to route to perform the specific tasks.

One is general **home (' / ')** where our webpage is displayed and another one is the **('/predict')** which displays the output for the given content using the deep learning model

Steps performed to predict the output:

- First, we get the input from the users and store them in respective variables namely open_price, close_price, and news
- We then calculate the sentiment scores for the News variable using respective python libraries
- We store the open-price, close_price, and calculated sentiment scores of the news into a list
- We download the dataset from .csv file
- Add the list created for the input data to the last row of the dataset and then scale the entire dataset with MinMax scaler
- We then take the last 60 rows of data excluding the newly added row and make a NumPy array and pass it as input to the model and predict the outcome of the next row which is nothing but the input data given by the user
- We scale back the values using inverse transform and display the predicted output and given output

Conclusion

With our model, we were able to predict future stock prices with the help of open price, close price, and the sentiment of news. So we were able to conclude that news articles do play an important role in the stock price variance as they reflect how the big players are making decisions in buying and selling stocks.

Future Work

More features can be added to the app like analyzing live tweets from Twitter and news from a news API. An option can be added to get live stock details of a particular stock and predict its future price.

Acknowledgments

The authors would like to thank our mentors, leaders, family, and friends who supported the completion of this research project. Appreciating everyone who helped us knowingly or unknowingly with this project.

Team

1. Dhruv Bhatia (Team Mentor)
2. Anjali Bathla (Student at IIT Roorkee)
3. Shraddha Joshi (MSc Data Science Student at Christ University, Lavasa)
4. Thota Sai Keerthana (IIITDM Kancheepuram)
5. Rajan Jangir(Engineering College Bikaner)
6. Himanshu Bhatia (Institute of Informatics and Communication, University of Delhi)
7. Parul Sharma (Institute of Informatics and Communication, University of Delhi)
8. Chakradhar Reddy Yerragudi (Information Technology Student, Manipal Institute Of Technology)
9. J.Suchita (MBA - Business Analytics at Christ University, Bangalore)
10. Aakruti Ghatole(Cummins College of Engineering for Women, Pune)