CAPSTONE PROJECT

Machine Learning Nanodegree

Edgar Velázquez

October 7th, 2017

Definition

Project Overview

In the 1600s, one of the first publicly recognized traded company was the East India Company or East India Trading Company. Investors financed or bought shares of ships for the exploration of new corners of the world searching for gold and treasures in exchange for a percentage of the values obtained. This represented high risk investments for investors because there was a 33% chance of the ships being seized by pirates and sometimes some of them never even returned or got lost at sea. This then resulted in the loss of shares and thus a loss of money for the investors.

Nowadays, although technologies and interests have changed from buying shares of ships to buying stocks and company values or debts, the risks are still similar. The difference from then to now is that back then there was not the same access to information nor the same amount of data about a certain topic or endeavor that we have today. We live in the age of information. Data is everywhere and valuable information can be gathered about companies that can give some insights about the future of such companies and thus the values of their shares.

Hedge funds, investment firms, and data analysts have been using the access to data to their advantage for years by reading the news, studying company history and trends in order to predict the value of stock shares of a company. The first attempt to use a financial model to predict stock prices was made by two statisticians named Box and Jenkins in 1976 using quantitative and statistical analysis.

Problem Statement

The seemingly random nature of the stock market constitutes a problem for inexperienced and experienced investors alike at the time of predicting the future price of a stock.

The challenge is to accurately forecast the random future price of a stock in a period of days using current time series, and the sentiment analysis of current news headlines. For instance, the latest time series displaying the volatility percent, volume, adjusted close, and percent change will be used to forecast the next 1 to 10 days of future prices of the stock. As part of the features of the data frame, the result of the sentiment analysis such as compound, negative percentage, neutral and positive percentage of all the headlines of each date in the data time series will be included. The source of the news will be New York Times and the most relevant headlines in related categories such as politics, business, economy, tech, etc.

Metrics

To test the results of the model predictions the difference between the actual future price and the model's predicted price are going to be used to get the percentage of difference. The percentage of difference need to be lower than 20% in order to assume that the stock price can be safely predicted with this technique.

Actual Price: $100.00

Model's Predicted Price: $90.00

Difference: 100/90 = $10.00

Difference %: Actual Price / Difference = 10%

The percentage of difference will be applied to every prediction of every single day in the future and then get the average of all the percentage of differences. To further confirm the accuracy of the results, the Mean Absolute error regression loss will be included as well to measure the quality of the prediction.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|prediciton_i - real_i|$$

The Mean Absolute error is the outcome of the sum of all predicted prices minus the real prices for each day divided by the total number of stock prices. This method is commonly used in regression models to tell the margin of error between the real price and the predicted price. It is a good metric for this purpose because it compares the distance of the prediction from the mean of the actual price stock. The result tells you the overall margin of error that we are trying to reduce from our prediction. It is good for handling large amounts of error and give an accurate result.

The last Evaluation metric is going to be R^2 (R squared) coefficient of determination to give a percentage of accuracy to our results. This metric returns a result value between 0 and 1 giving an overall percentage of how well the model is doing and it is pretty easy interpret which is why this metric is chosen for this purpose. It is calculated by getting the result of the sum of squares of the prediction divided by the sum of squares of the actual values.

$$R^2 = \frac{SSR}{SST} = \frac{\sum \hat{y}_i^{\,2}}{\sum y_i^{\,2}}$$

The use of all of these evaluation methods will result in very solid and accurate measurable outcome of the predicted values.

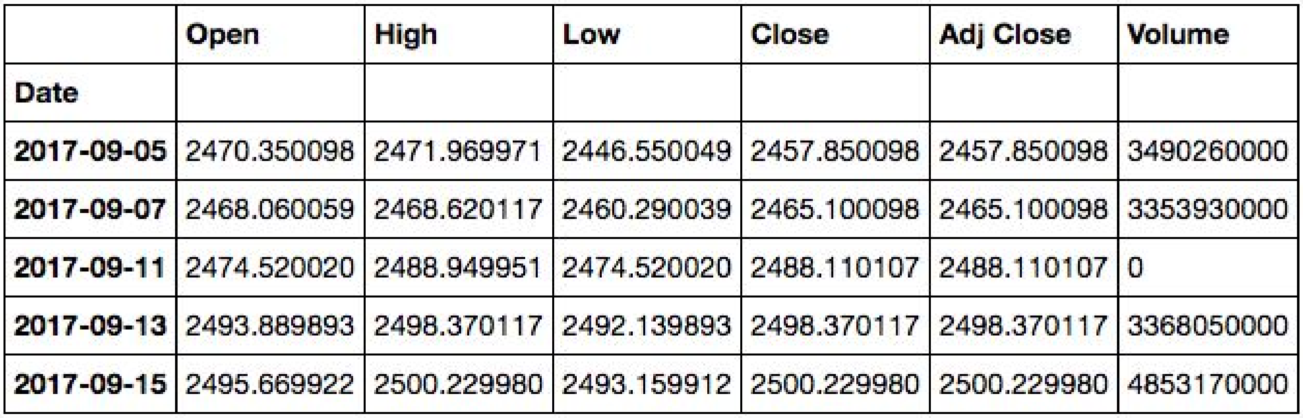
Analysis

Data Exploration

In this example the chosen stock will be S&P 500 index but the models are going to be tested against other stocks such as Google and Apple stocks as well. The S&P 500 is composed of 500 large-cap U.S based companies stocks which market cap is $5.3 billion or more. More than 10 years of data is collected for this process starting from January, 01 2007 to September 15 2017 (2673 records).

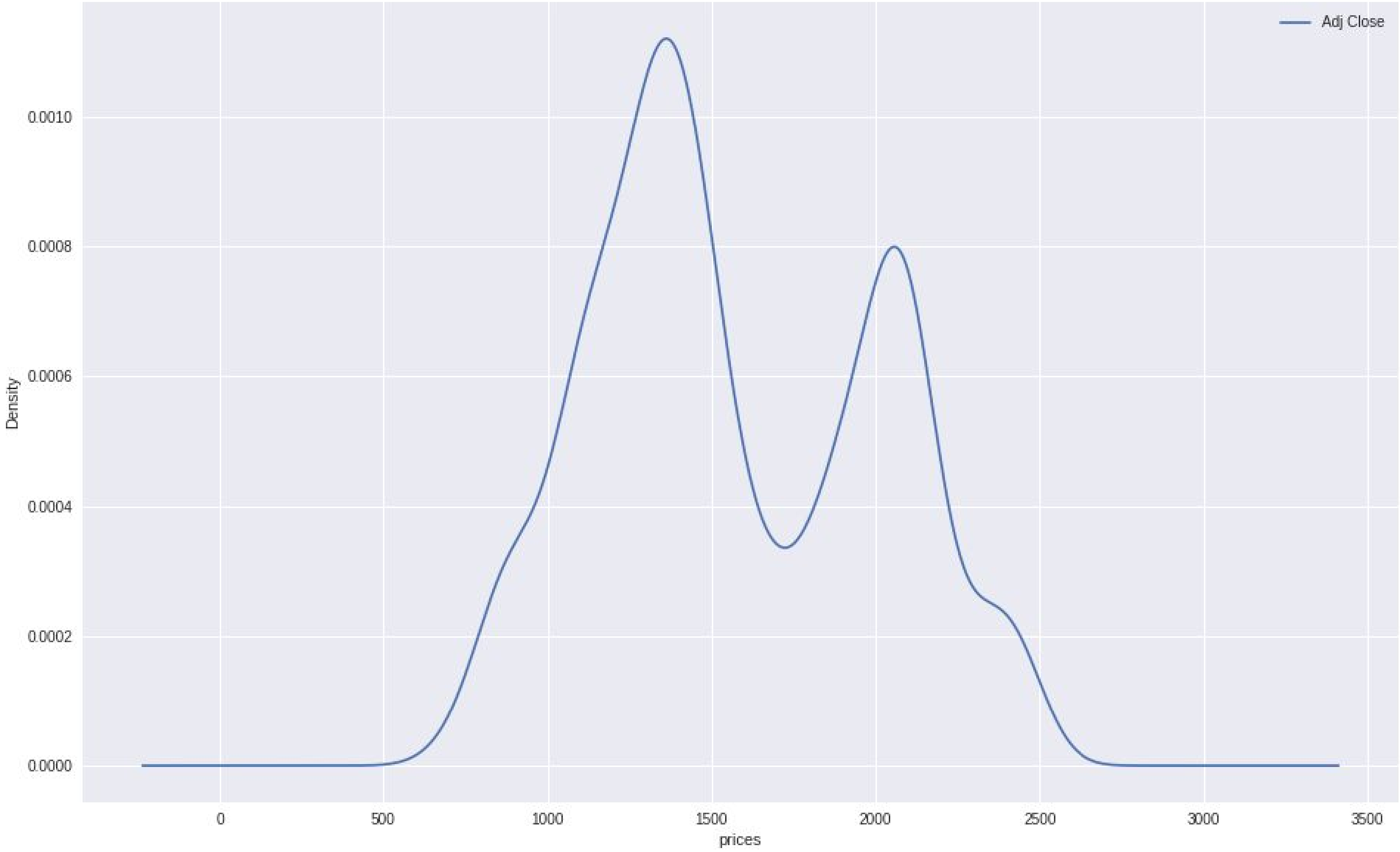
The following dataset is the one that is going to be used during this process.

- Open: The initial price of the stock at the beginning of the day.
- High: The peak or highest value reached by the stock in the day.
- Low: The lowest price reached by the stock during the day.
- Close: The final price of the stock at the end of the trading day.
- Adjusted Close: The final price with the adjusted stock splits.
- Volume: The amount of stock traded in the day.

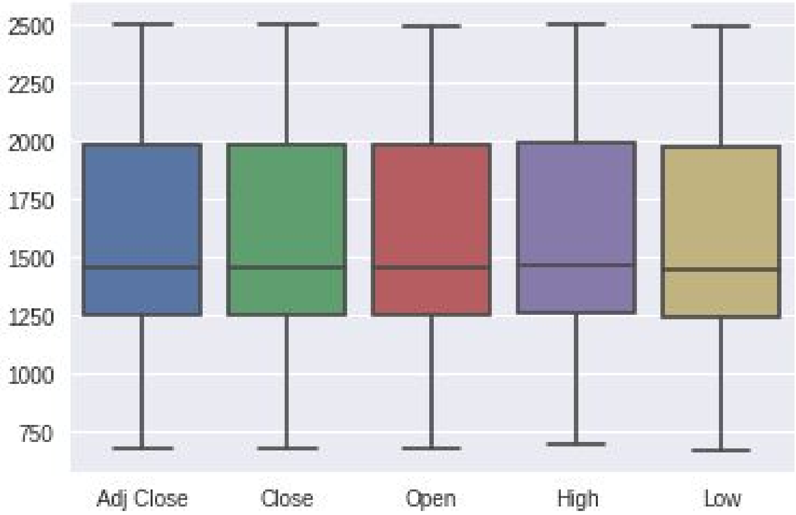


	Open	High	Low	Close	Adj Close	Volume
Date						
2017-09-05	2470.350098	2471.969971	2446.550049	2457.850098	2457.850098	3490260000
2017-09-07	2468.060059	2468.620117	2460.290039	2465.100098	2465.100098	3353930000
2017-09-11	2474.520020	2488.949951	2474.520020	2488.110107	2488.110107	0
2017-09-13	2493.889893	2498.370117	2492.139893	2498.370117	2498.370117	3368050000
2017-09-15	2495.669922	2500.229980	2493.159912	2500.229980	2500.229980	4853170000

This time series dataset was obtained through the Yahoo Finance API (https://finance.yahoo.com/). In order to explore the data further, some statistical analysis has been conducted. In the graph we can see that in the density distribution, there is a bimodal distribution with 2 local maximum of ~$1400 and ~$2100.

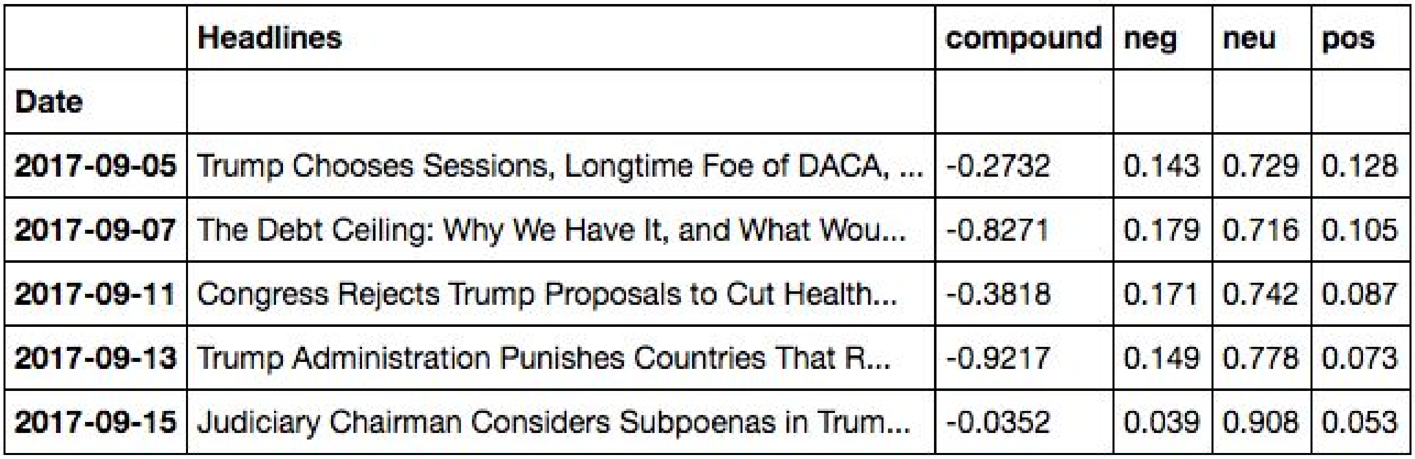


The time series dataset stock prices range between 600 and 2500. The price range and density seem to be evenly distributed and do not present outliers.



The following data frame was my own personal collection and the current result was acquired using the New York Times Developers API (https://developer.nytimes.com/). The compound, neg, neu, and pos was obtained using the SentimentIntensityAnalyzer tool from the Natural Language Toolkit (http://www.nltk.org).

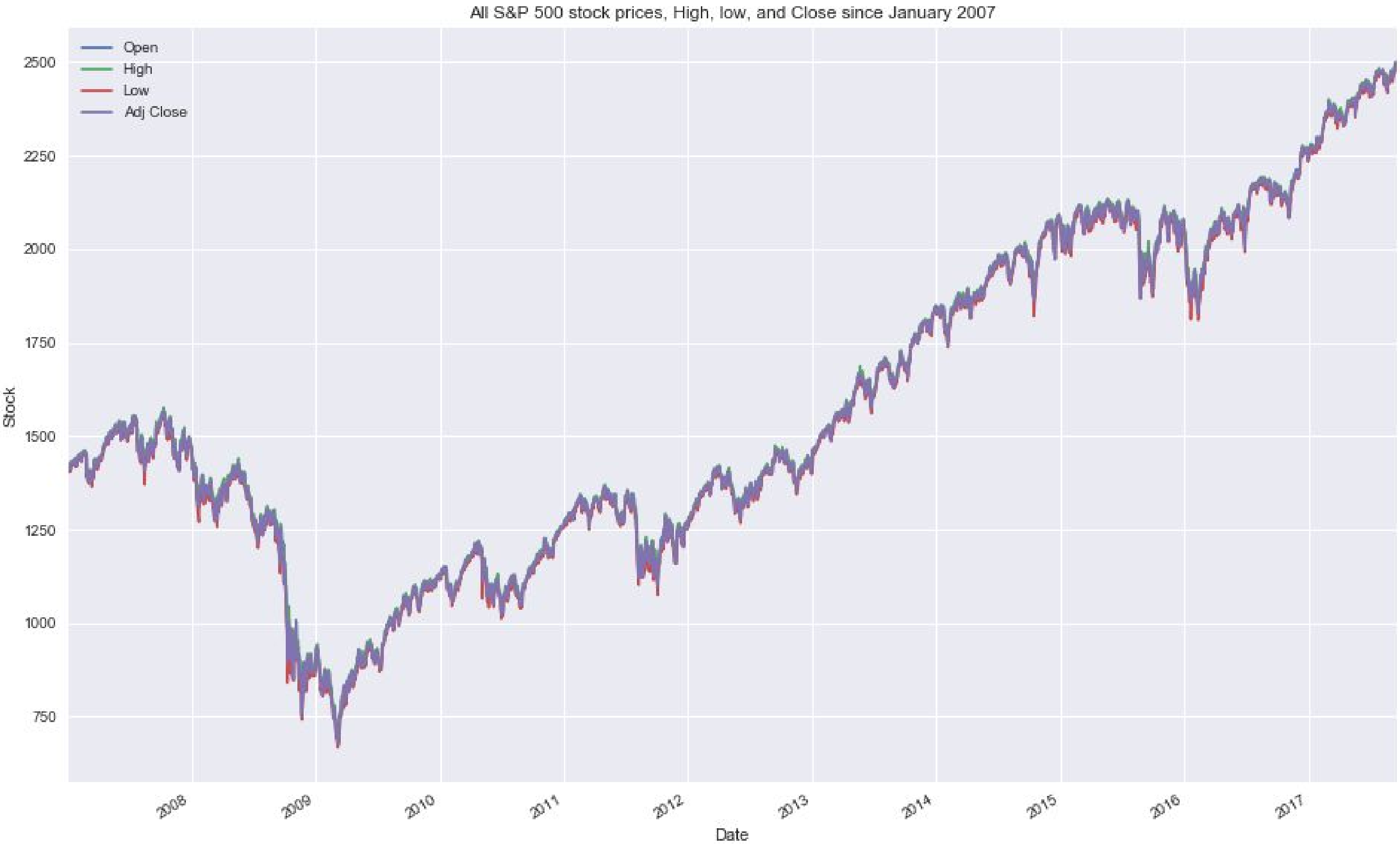
According to the Natural Language Toolkit documentation, this tool returns the number of positive, negative and neutral words and a percentage of the overall, neutrality, positivity, and negativity of the whole paragraph. The record is also more than 10 years of data starting from January 1 2007 and ending on September 15 2017 (2673 records) and around 30 different Headlines per date. All headlines belong to the categories of business, national, world, u.s. , politics, opinion, tech, science, health, and foreign.



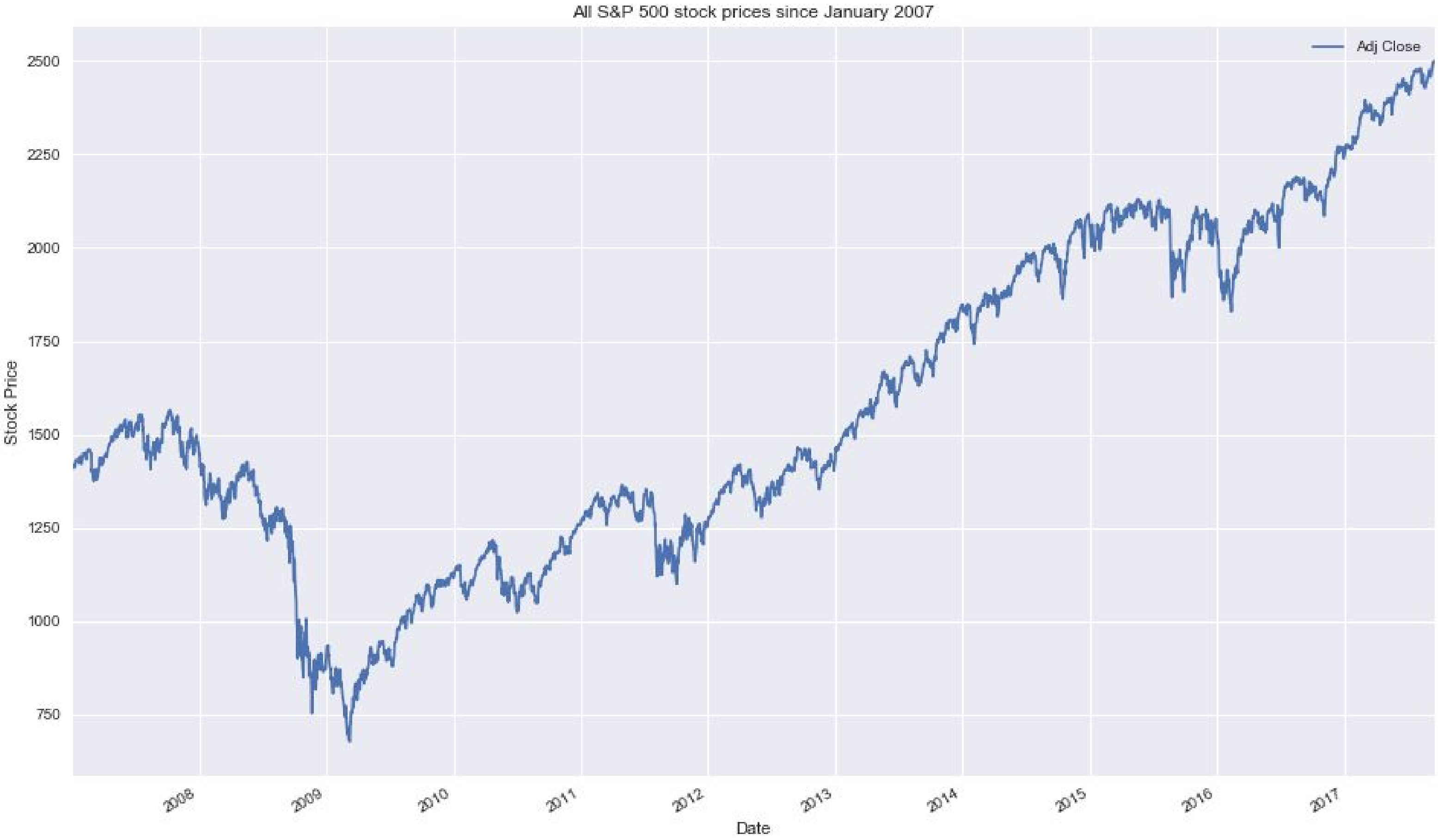
	Headlines	compound	neg	neu	pos
Date					
2017-09-05	Trump Chooses Sessions, Longtime Foe of DACA, ...	-0.2732	0.143	0.729	0.128
2017-09-07	The Debt Ceiling: Why We Have It, and What Wou...	-0.8271	0.179	0.716	0.105
2017-09-11	Congress Rejects Trump Proposals to Cut Health...	-0.3818	0.171	0.742	0.087
2017-09-13	Trump Administration Punishes Countries That R...	-0.9217	0.149	0.778	0.073
2017-09-15	Judiciary Chairman Considers Subpoenas in Trum...	-0.0352	0.039	0.908	0.053

Explanatory Visualization

To visualize the time series dataset, the selected column used for this purpose are the Open, High, Low, and Adj Close. The first example shows the trend of the stock price and volatility including high, low and open prices.

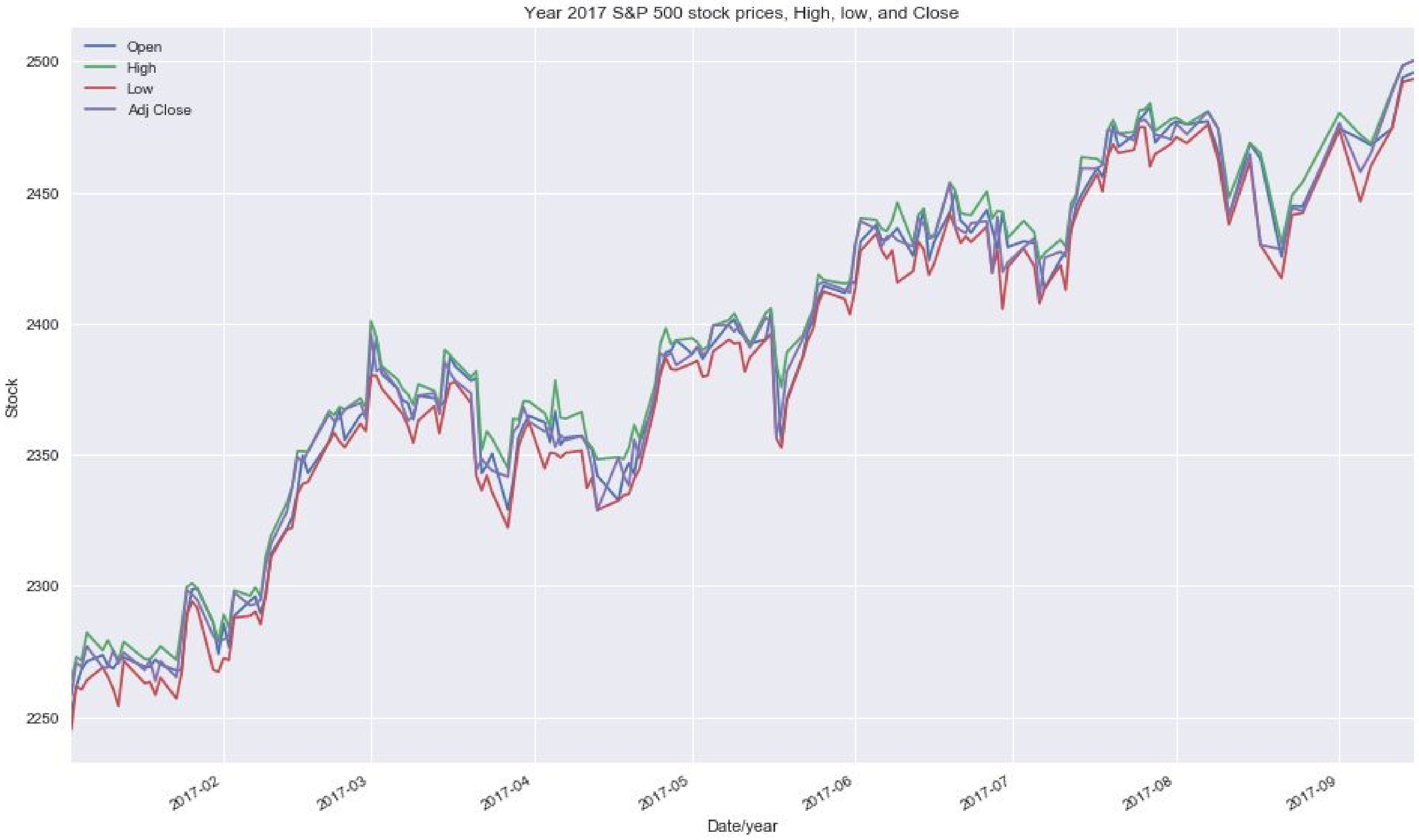


Isolating the Adjusted Close stock price will show a more detailed trend and overall direction of the stock price.



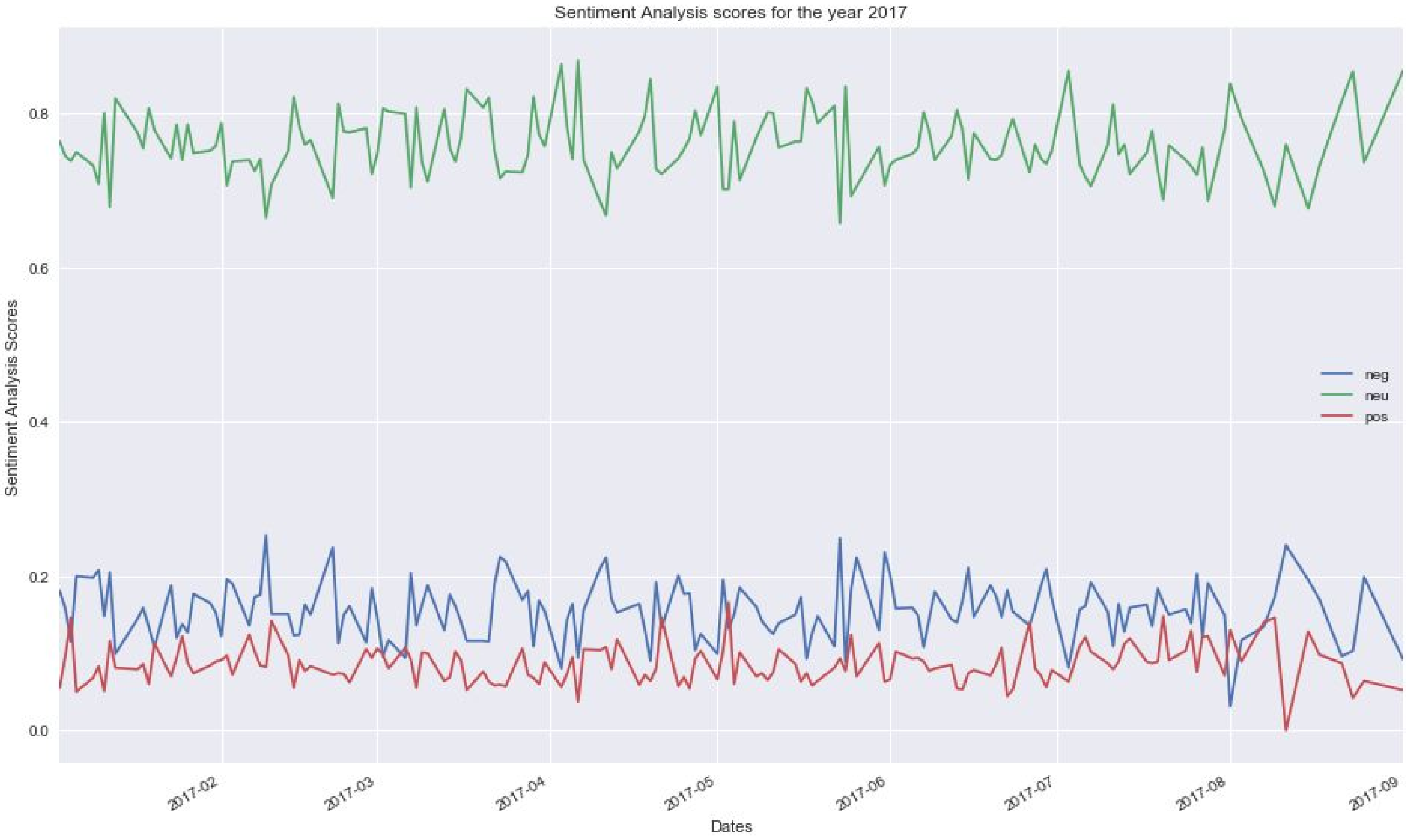
It can be seen clearly that the S&P 500 stock price has been growing since January 2007 going from around $1400 to over $2500 by September 2017. It can also be noted a clear fall of the stock price starting in the year 2008. This dip falls in within the timeline of the financial crisis that happened in the year 2008. The graph then shows the stock came back up to its normal price at the beginning of the year 2013 where it grows steadily since then.

Isolating just the year 2017 since January 01 will display a closer look of the High, Low, and Adjusted Close.



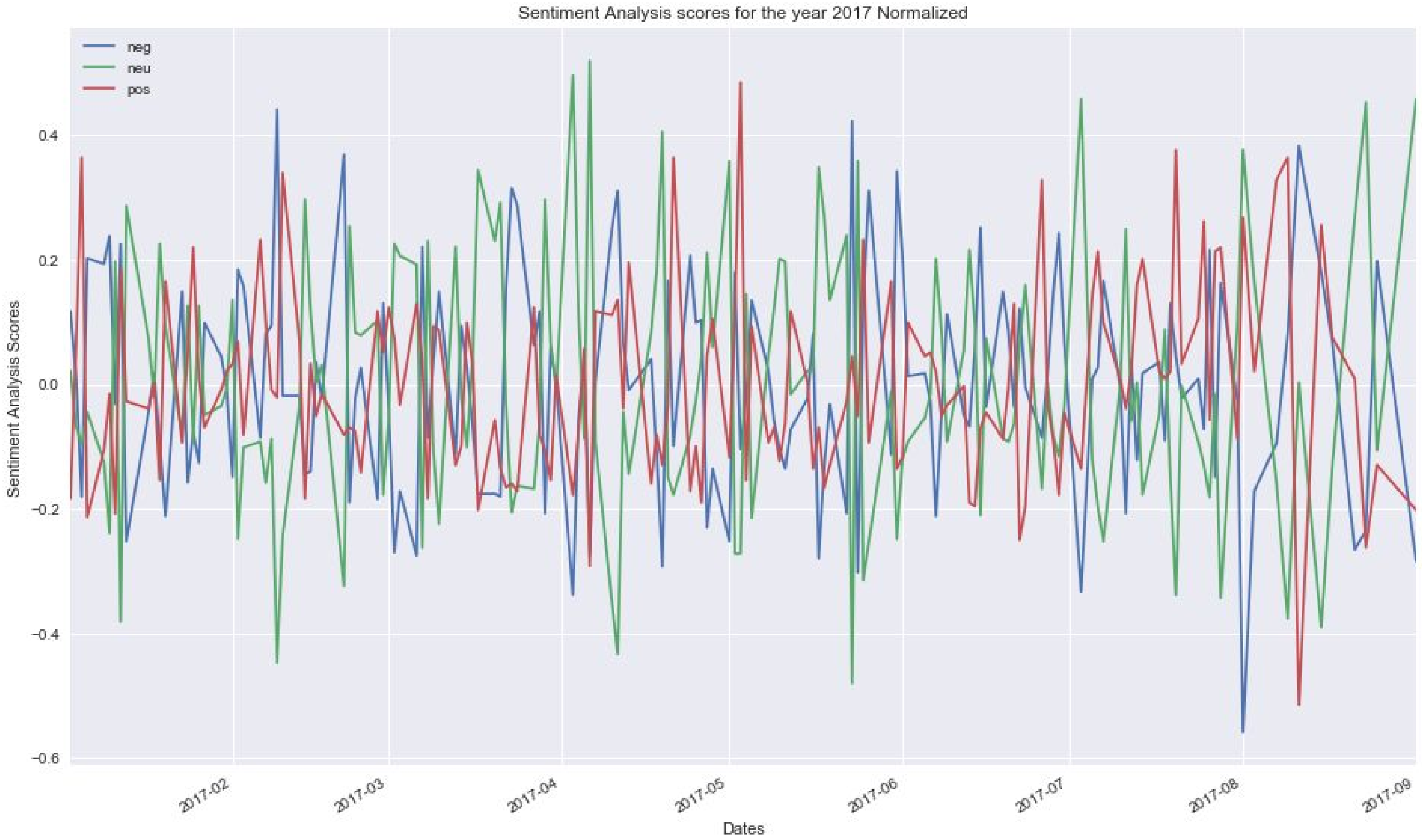
A closer look to the year 2017 shows a more clear difference between the High and Low prices demonstrating the level of volatility of this particular stock.

The following data frame displays the sentiment analysis scores for the year 2017. The scores are taken from the collected New York Times Headlines. The columns that are displayed are Neutral, Negative, and Positive which correspond to the count of words and overall polarity of the headlines for each date.



In the Natural Language Toolkit (nltk) documentation, the Sentiment Analysis tool returns a sentiment intensity score to sentences. The average of the negative score is generally greater than the positive and the neutrality of the score is a lot higher than both.

Normalizing the dataset will display a better representation of the sentiment scores.



The normalized sentiment analysis scores represents a more accurate fluctuation between negative, positive, and neutral sentiment scores. The difference can be noted right at the end of the excerpt graph, in the 2017-09-01 date**,** where it can be seen the negative score at the lowest point, positive just some points above it, and the neutral score as the highest meaning that the headlines for that period were highly neutral in nature.

Algorithms and Techniques

The algorithms considered for the purpose of this projects are the Linear Regression algorithm, the K Nearest Neighbor algorithm, the Random Forest algorithm, the Support Vector Machine algorithm, and a Deep Neural Networks algorithm.

The Linear Regression algorithm, besides being a simple algorithm, is perfect for predicting the direction of a certain trend, or stock price in this case. It does this by calculating the best fit line and then predicting the next point in the line. It is proven to be one of the most popular algorithms to start with. Its simplicity makes it easy to apply and test.

$$y = mx + b$$

Among the ensemble methods the Random Forest algorithm is also a popular one among financial data scientists. The Random Forest uses a set number of decision trees and then it returns the mean of all the results of the decision trees applied as the final result. It is one of the most accurate and efficient algorithms for both classification and regression. The number of estimators parameter is the number of decision trees that will be included in the Forest. To start the process we will use the default of 10 trees. Every decision tree in the Random Forest uses the entropy formula to create the tree branch divisions.

$$Entropy = \sum_{i=1}^{C} -p_i * \log_2(p_i)$$

The Deep Neural Network algorithm is one of the new most powerful models used for many different purposes. It can be used for binary classification, categorical, and regression with very low level of error. The perceptron algorithm takes as parameter some inputs, weights and bias and it goes all of these parameter go through an activation function resulting on an updated weight thus resulting on a lower margin of error and more accurate results. This process is called gradient descent. A Deep Neural Network or Multi Layer perceptron uses the perceptron algorithm as its core. As our default parameters, a configuration of 100, 200, and 100 hidden layers is going to be used to start. The activation function chosen is the relu activation function and the batch size is of 100 records per loop.

$$y = \varphi(\sum_{i=1}^{n} w_i x_i + b) = \varphi(\mathbf{w}^T\mathbf{x} + b)$$

Some other considered algorithms are The Support Vector Machine Regression that is great for it's clear margin of separation in its results and the K nearest Neighbors regressor that takes the features as parameters and tries to predict the closest number of a sequence which will prove useful to predict the next stock price from a sequence of stock prices.

As parameters chosen to be used in the Support Vector Machine Regression the Linear kernel is selected and along with this a maximum of 2000 iterations.

For the K Nearest Neighbor algorithm, a number of 3 neighbors will be used for this purpose.

Benchmark

As for the chosen benchmark model, the Dr. Eric Brown (http://ericbrown.com/) created a very compelling analysis using the S&P 500 index trying to predict the future price of the stock as well. In his analysis, he used both R^2 and the MAE as evaluation metrics getting 0.91 with the R^2 metric and 36.16 % in the MAE metric. The analysis can be found in http://pythondata.com/stock-market-forecasting-with-prophet. The tool used by him is created by Facebook and it is called Prophet which is a procedure used for forecasting time series data (https://facebook.github.io/prophet/).

Methodology

Data Preprocessing

To start the process, the S&P 500 time series dataset of over 10 years of length from January 2007 to September 2017 will be used. Using feature engineering, from the Adj Close and the High columns the daily percentage change or daily return will be calculated to create a new feature for the dataset.

Daily Percentage Change = price[t] / price[t-1] - 1
where t = current day

To further complete the dataset and generate a label, the Adj Close will be shifted N numbers of days to the past so that this column becomes a new feature called Future Price. To this, we are going to add the Bollinger Bands and the Rolling mean as new features calculated from the current Adj Close in a 20 days window. The new columns after the mentioned calculations are Rolling mean, Upper Band, and Lower Band that are added to the existing columns (Daily Percentage Change, Rolling mean, Upper Band, Lower Band). The resulting dataset is what we are going to feed to our models first.

Rolling Mean = 20 days moving average
Upper Band = N days moving average + (20 Days standard deviation of price x 2)
Lower Band = N days moving average - (20 Days standard deviation of price x 2)

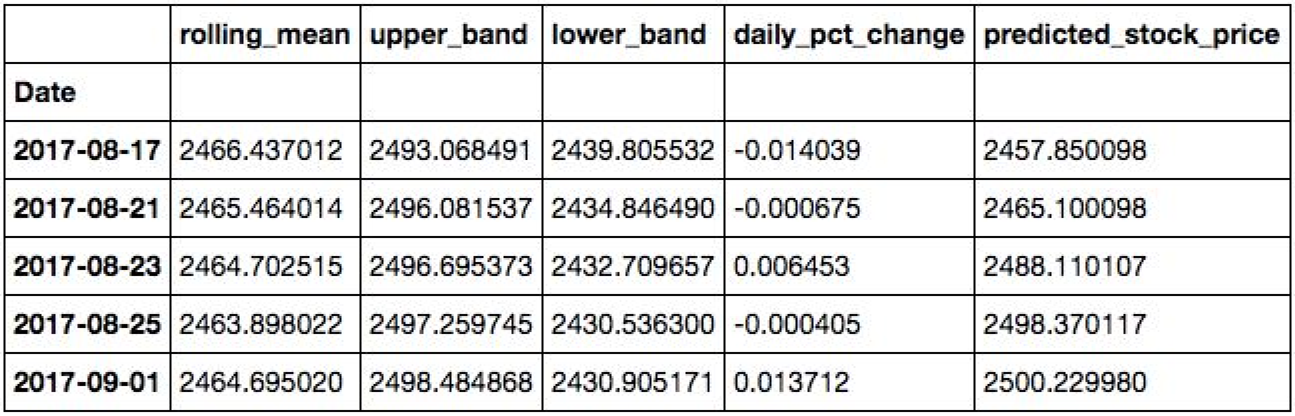
The Headlines dataset, as we saw in the exploratory visualization section, need to be normalized in order to accurately represent the sentiment polarity score of each headlines. The normalization will be done using the mean normalisation formula.

$$x' = \frac{x - \mathrm{mean}(x)}{\max(x) - \min(x)}$$

Implementation

The process of generating trained machine learning models is divided in three steps. Before any steps are initiated, all the algorithms are imported from the sklearn library and the dataset is preprocessed. To set the label, the Adj Close column is shift N days into the past and it is renamed as predicted_stock_price. The N value for the purpose of this project is going to be 5 so the prediction that is needed is 5 days into the future. A helper function called train_test_data_model is defined and its function is to split the dataset into training and testing sets, it trains the data and then it tests it, it also prints out the R2 and the Mean Absolute error scores and it plots the predicted value along with the expected label to compare with other results. This function makes the testing of many different algorithms and combinations a lot easier. It is important to note that during the split, the data is not shuffled in order to maintain accuracy and to not mix future prices with past prices in the training dataset.

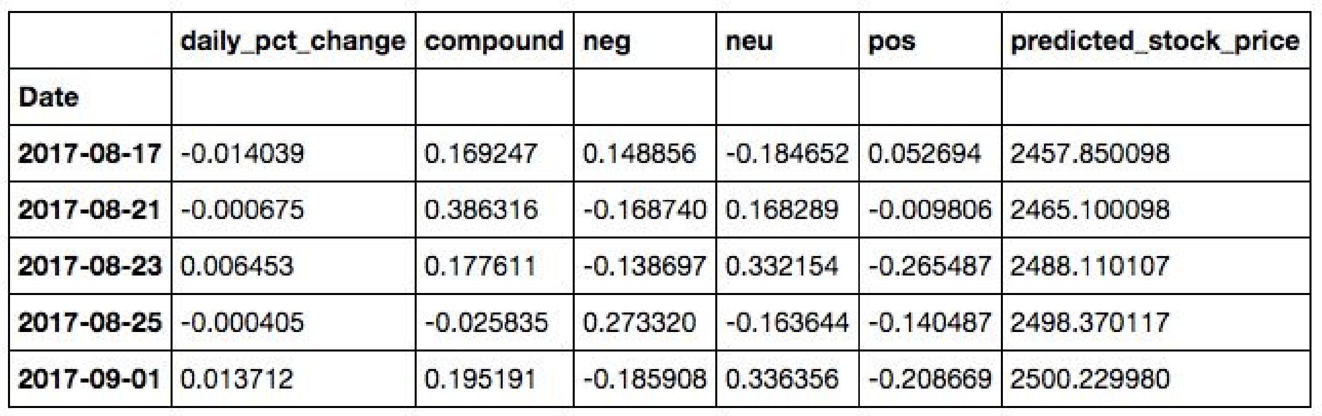
- First, before the sentiment analysis of the headlines is used, the dataset without the headlines analysis will to serve to train and test the chosen machine learning algorithms. The results of this step will be compared with the next 2 steps to gather the best performance and thus the best option data set features. The preprocessed time series has a new set of columns (rolling_mean, upper_band, lower_band, daily_pct_change, predicted_stock_price) set in the Data Preprocessing section.



	rolling_mean	upper_band	lower_band	daily_pct_change	predicted_stock_price
Date					
2017-08-17	2466.437012	2493.068491	2439.805532	-0.014039	2457.850098
2017-08-21	2465.464014	2496.081537	2434.846490	-0.000675	2465.100098
2017-08-23	2464.702515	2496.695373	2432.709657	0.006453	2488.110107
2017-08-25	2463.898022	2497.259745	2430.536300	-0.000405	2498.370117
2017-09-01	2464.695020	2498.484868	2430.905171	0.013712	2500.229980

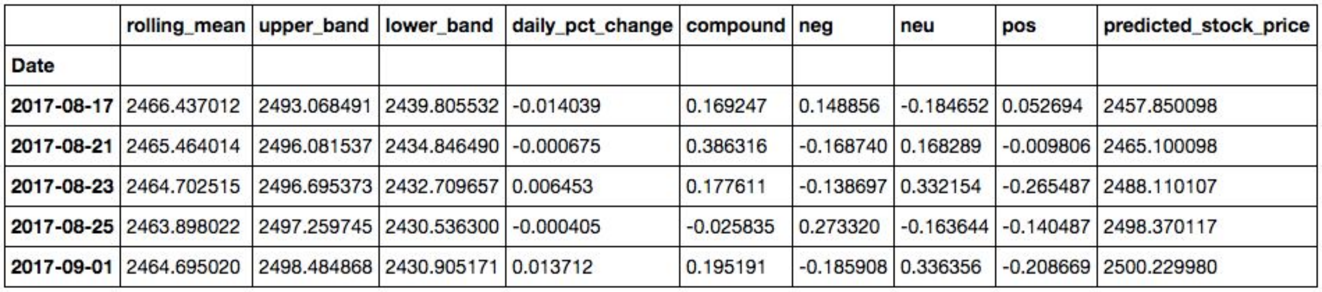
With the preprocessed dataset in order, the chosen algorithms can be tested. The train_test_data_model function will loop through all the algorithms and output the R^2 Square and the Mean Absolute Error of every single algorithms as each is trained and tested. This process is repeated for the next 2 steps as well.

- The result of this first try will then work as a guideline to then attempt the second try which consists of using just the Daily Percentage Change and the sentiment analysis of the headlines as features and the Future Price as label. The columns of the dataset then will be (Daily Percentage Change, compound, neg, neu, pos, Future Price). The train_test_data_model function is ran to gather all the performances scores with this dataset.



	daily_pct_change	compound	neg	neu	pos	predicted_stock_price
Date						
2017-08-17	-0.014039	0.169247	0.148856	-0.184652	0.052694	2457.850098
2017-08-21	-0.000675	0.386316	-0.168740	0.168289	-0.009806	2465.100098
2017-08-23	0.006453	0.177611	-0.138697	0.332154	-0.265487	2488.110107
2017-08-25	-0.000405	-0.025835	0.273320	-0.163644	-0.140487	2498.370117
2017-09-01	0.013712	0.195191	-0.185908	0.336356	-0.208669	2500.229980

- The third step will be the combination of the first two steps and all mentioned columns will be implemented together as such (compound, neg, neu, pos, Rolling mean, Upper Band, Lower Band, Daily Percentage, Future Price). The result of the third step will then be compared with the results of the previous steps to observe which of the steps is the most accurate and thus selecting the best of the three steps to predict future stock prices.



	rolling_mean	upper_band	lower_band	daily_pct_change	compound	neg	neu	pos	predicted_stock_price
Date									
2017-08-17	2466.437012	2493.068491	2439.805532	-0.014039	0.169247	0.148856	-0.184652	0.052694	2457.850098
2017-08-21	2465.464014	2496.081537	2434.846490	-0.000675	0.386316	-0.168740	0.168289	-0.009806	2465.100098
2017-08-23	2464.702515	2496.695373	2432.709657	0.006453	0.177611	-0.138697	0.332154	-0.265487	2488.110107
2017-08-25	2463.898022	2497.259745	2430.536300	-0.000405	-0.025835	0.273320	-0.163644	-0.140487	2498.370117
2017-09-01	2464.695020	2498.484868	2430.905171	0.013712	0.195191	-0.185908	0.336356	-0.208669	2500.229980

During this implementation, no real complications have been observed other than the collection of the headlines data itself. This problem has been resolved with the creation of the helper function *getMainDataset* found in the finance.py file. The parameter *current_headlines* when set to true will collect and format all current headlines and return it as part of the main data frame. This helps with the process of collection of new headlines in case the dataset is not present.

Refinement

During the process of the three steps mentioned above it was noted that the best combination of dataset and features is the one used in the first step: the time series with preprocessed features. The best machine learning model was the Deep Neural Networks with results of 0.91337266755 of R^2 score and 35.8791117466 of Mean Absolute Error score. The random parameter search (RandomizedSearchCV) is then used to improve the performance of this algorithm and find the best hyper parameters. The parameters implemented for the search were the hidden_layer_sizes, batch_size, and activation function. The best parameters result were (100,100) of hidden_layer_sizes, batch_size as 121, and identity as the activation function returning a R^2 score of 0.909547764987 and a Mean Absolute Error score of 37.039209359 showing a reduced overfitted score but a more accurate line of trend.

Results

Model Evaluation and Validation

The chosen model is the Deep Neural Networks for it has displayed the best performance throughout all the steps taken. In the first step where the dataset was the time series and the preprocessed columns, the performance was the best among all the other algorithms scoring R^2 0.91337266755 and Mean Absolute Error score of 35.8791117466.

In the second step, where the data Frame used was the Daily Percentage Change along with the Sentiment Analysis scores, the Random Forest algorithm performed better than the Deep Neural Networks algorithm with a R^2 score of -10.290848623 and a Mean Absolute error of 507.994360173 and the Deep Neural Networks with a R2 of -18.4114715998 and a Mean Absolute Error of 704.954292832. In this instance, the algorithms seem to perform worse as no real correlation seems to exist between the sentiment polarity scores, the time series rate of return and predicted prices.

In the third steps, with the combined time series and sentiment scores, the Deep Neural Networks algorithm performed better than the other algorithms again making it the best in 2 out of 3 tests.

The chosen model and data set configuration was then refined using the RandomizedSearchCV method to find the best parameters and with this the result were reflected a lower overfitted score with a R2 score of 0.909547764987and a Mean Absolute Error of 37.039209359 getting the best performance so far. A last test was also performed to check the last date in the test set, 2017-08-09. The result of this was used to compare how far off the result of the prediction was for this date and the actual price. The result of the prediction was of $2483.22 and the actual price is $2444.04 having a margin of difference of $39.18 and a percentage of difference of 1.60%.

To test if the model is robust, the model design was tested with the Google time series from the same time frame (January 01 2007). The results returned by the model was a R2 score of 0.871006237103 and a Mean Absolute Error of 25.2141038124. The date chosen to test the price was 2017-08-09 and the predicted stock price for this date was $974.35 and the future value was $942.58. The margin of difference is of $31.77 which is a 3.37% difference. The results of this last test further proves that the procedure and algorithm chosen display strong results.

Justification

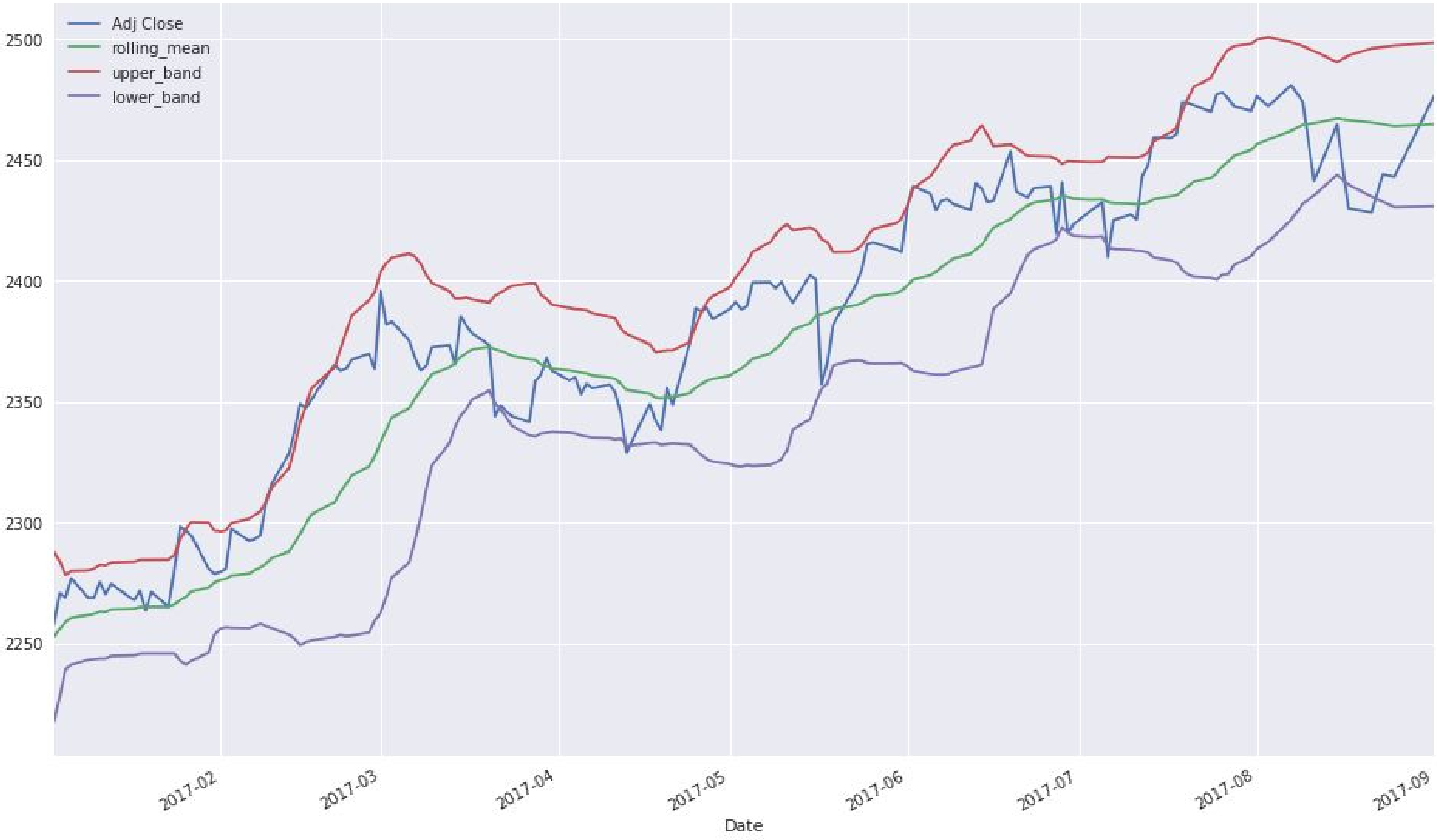
Even though it is hard to successfully predict the price of a stock 100% accurately, the methods applied to become somewhat accurate in forecasting are valid. Facebook has released a time series data forecasting tool that is pretty powerful and it is aimed to data scientists and analysts. In his Model, as mentioned in the Benchmark section, Dr. Eric Brown used this tool to forecast the S&P 500 future price. He used the same metrics applied here which are the R2 score and the Mean Square Error. The data set, however, is different. Even though it is also the S&P 500 index time series, the span of time is different. The starting date for the time series he used is 2008-12-08 and it ends on 2017-08-30 whereas the one used in this project starts on 2007-01-01 and end on 2017-09-15. The forecasting time is also different as Dr Brown tests the model in a ~2 years time span. The R2 score he got in this test was 0.91 and a Mean Absolute Error of 36.18 performing slightly better than the model chosen here. The model chosen here performed 0.8 % worse in the Mean Absolute Error Metric and 0.000452235 % worse in the R2 score.

Conclusion

Free-Form Visualization

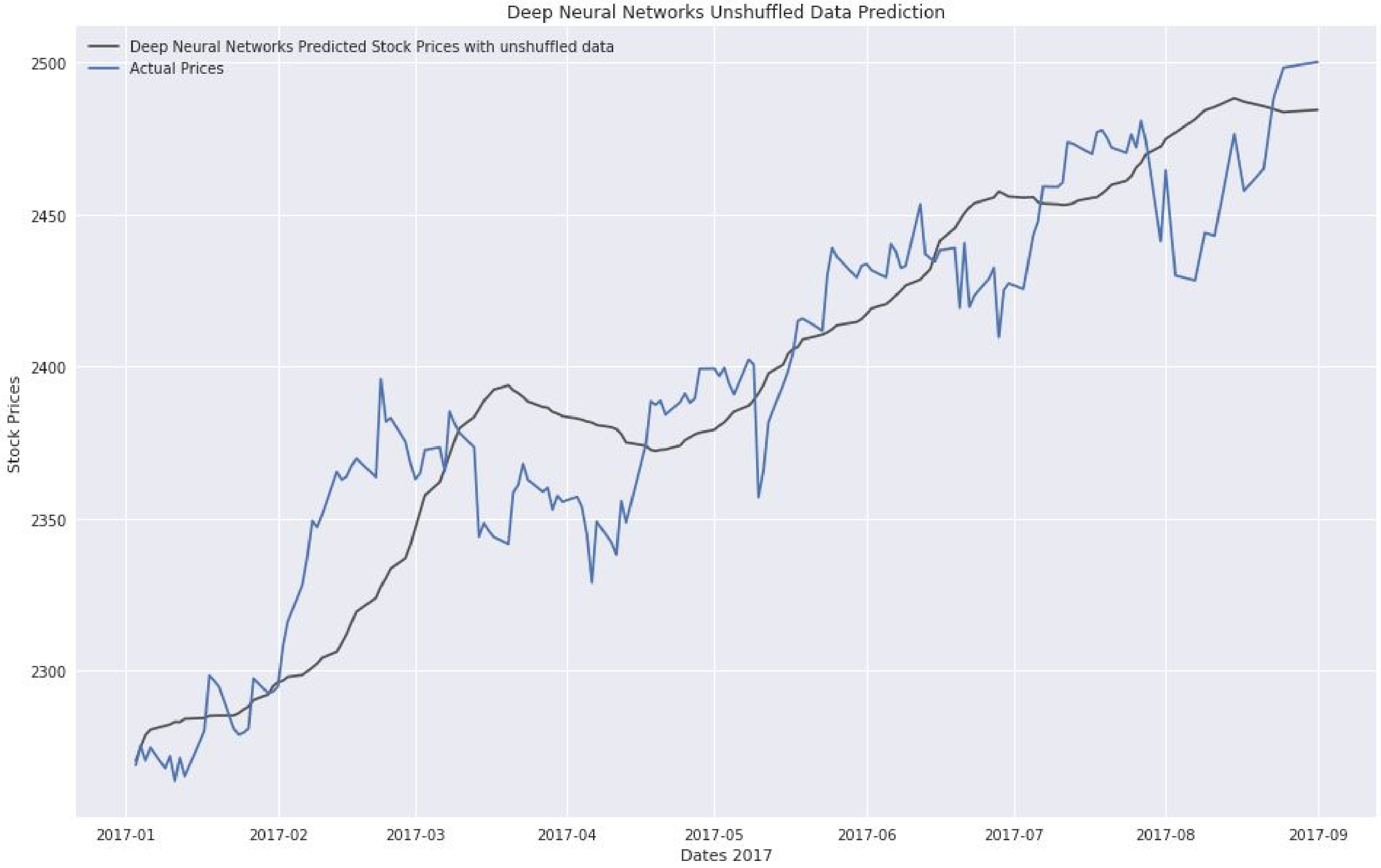
The following graphs are the Scores and Graph during the model selection and testing steps. All the Graphs provided are based on the year 2017 starting on January , 1 and end on 2017-08-09.

During the Feature engineering process, this is the graph for the Bollinger Bands.

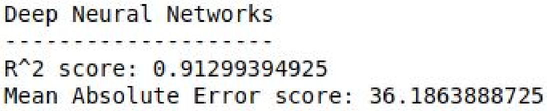


First Step DNN Performance.

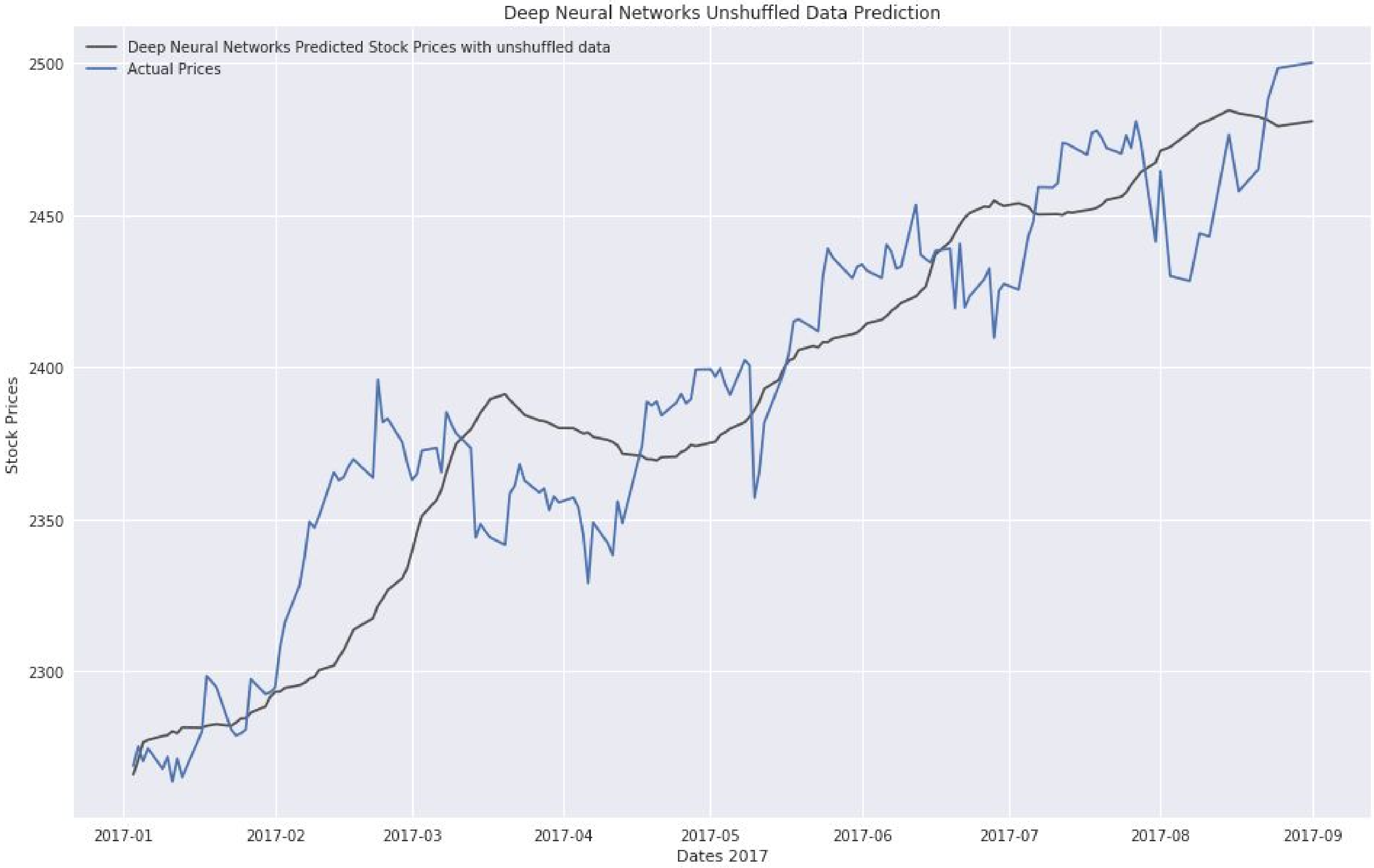
```
Deep Neural Networks
--------------------
R^2 score: 0.91337266755
Mean Absolute Error score: 35.8791117466
```



Third Step DNN Performance (Time Series with Sentiment scores).



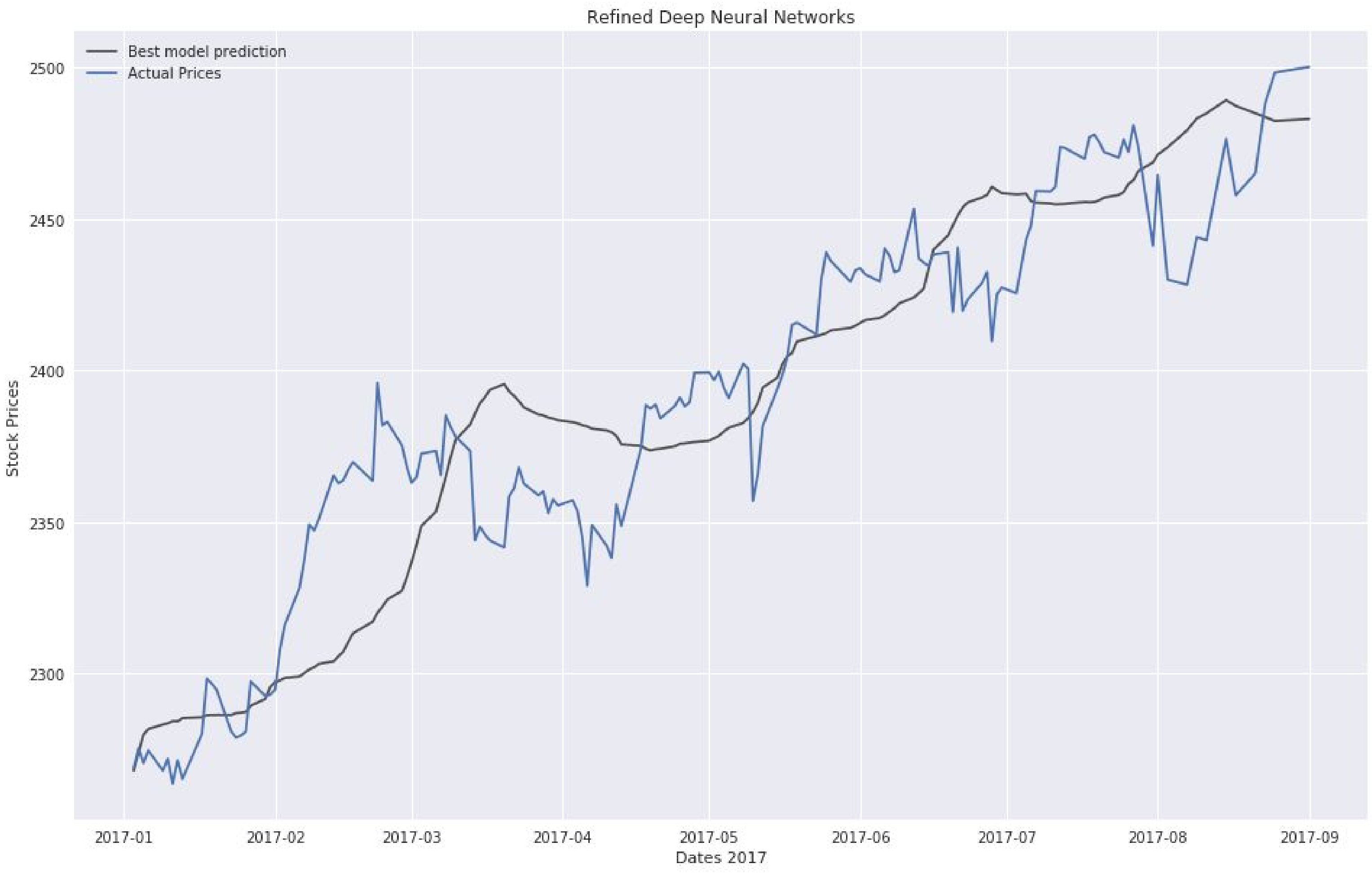
```
Deep Neural Networks
--------------------
R^2 score: 0.91299394925
Mean Absolute Error score: 36.1863888725
```



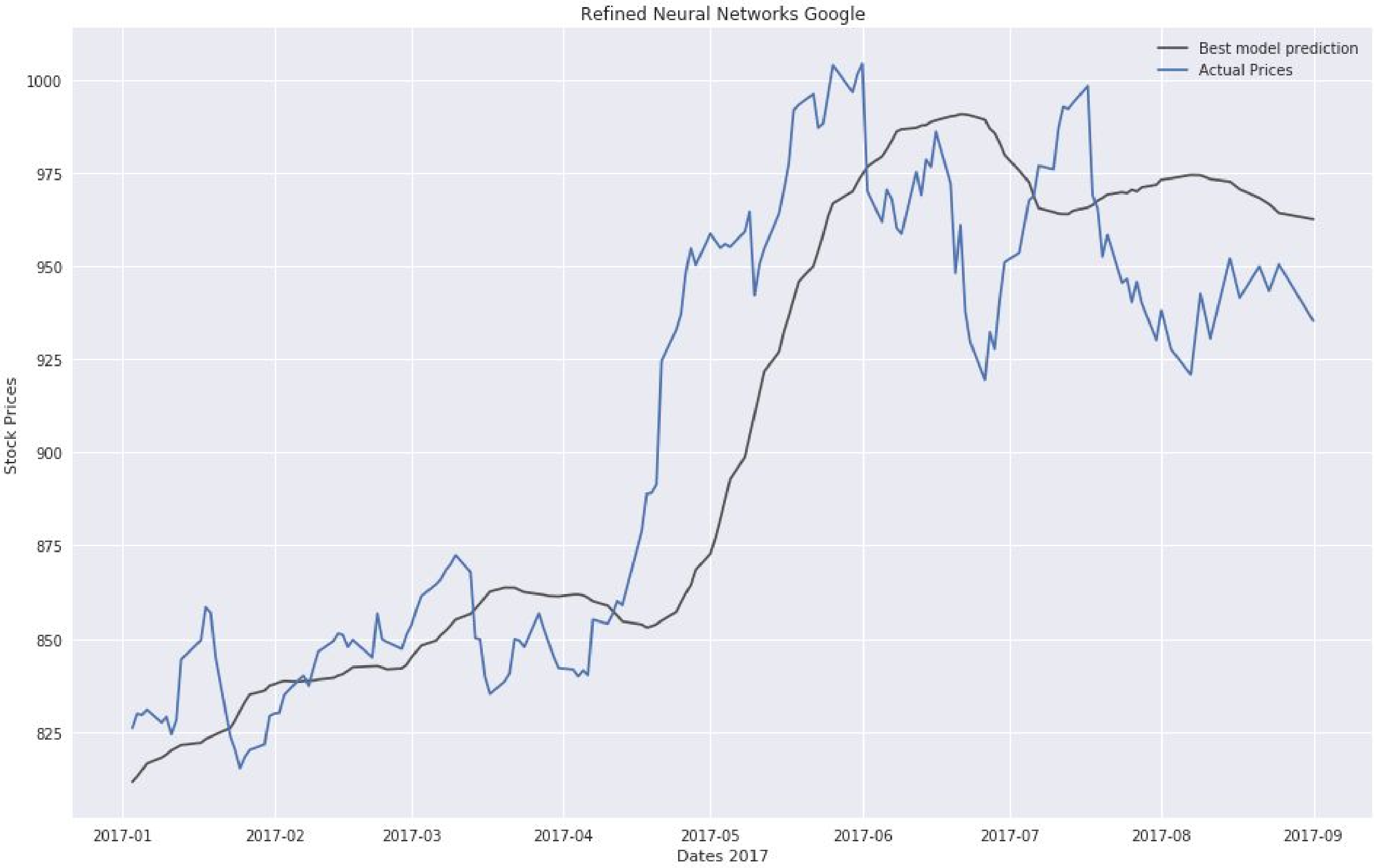
The Deep Neural Networks was the best in 2 out of 3 tests. The results after the Hyper Parameter:

R^2 score: 0.909547764987

Mean Absolute Error score: 37.039209359



Test with Google time series



Reflection

The process followed the following steps:

1. Import both S&P 500 time series and headlines sentiment scores matching both by date.
2. Preprocess the datasets. Calculate Bollinger bands, and daily percentage change for the time series. For the sentiment scores, normalize all scores to get a better representation of the data.
3. Import all 5 machine learning algorithms (Random Forest, Linear Regression, K Nearest Neighbors, Support Vector Machine, Deep Neural Networks).
4. Divide the testing steps into 3. First the train, test and plot all algorithms on the data time series with no headlines scores.
5. Second, train, test and plot performance of the headlines scores with the Daily percentage change and predicted prices to see the quality and performance of the algorithms on this dataset.
6. Third, train, test and plot performance on the combined times series with sentiment scores.
7. Get the best performance on both dataset and algorithm comparing which dataset or combination performed the best and compare all algorithms to get the best performing algorithms.
8. Refine the chosen algorithm on the best dataset configuration with hyper parameter tuning using RandomizedSearchCV.
9. Test the newly tuned algorithm on a different dataset, for this purpose we used the Google time starting from January 1, 2007.
10. Compare the results against the Benchmark and analyse the differences and the strength of the chosen algorithm.

The difficult aspect of the projects were to create the best outline for the whole process to get accurate outcomes. Data analysts already use the Bollinger bands and moving average as part of their features whenever they try to predict future stock prices. To add the headlines to this process was quite the challenge as the headlines represent just the polarity of the news. To find actual correlation between time series and Headlines is a challenge. It was expected that the combination of both datasets was going to perform better than the data sets individually.

Improvement

I believe there are further improvements to do to the process applied in this project. It was a challenge to gather all the headlines and try to find the best correlation between time series data and headline scores. There has to be a better method to implement this combination.

One of the algorithms I researched to predict future prices was the Recurrent Neural Networks with Long Short Term Memory (LSTM) that I did not have enough knowledge to implement. I believe that I would have gotten better results in this project if I knew how to study the sequences themselves and knew how to apply LSTM.

If I used this solution as a new benchmark, I do think there are better solutions. Research on how to better predict the stock market prices are done continuously. Powerful algorithms and new methods are trained and tested every day. There is a high possibility that better models already exist.

References

- https://bebusinessed.com/history/history-of-the-stock-market/
- https://link.springer.com/article/10.1007/BF00167127
- https://www.fool.com/knowledge-center/what-is-the-sp-500.aspx
- http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/
- https://www.kaggle.com/wiki/MeanAbsoluteError
- https://developer.nytimes.com/
- https://finance.yahoo.com/
- http://www.nltk.org/
- https://www.youtube.com/watch?v=JuLCL3wCEAk&t=66s
- https://www.udacity.com/course/machine-learning-for-trading--ud501
- https://pythonprogramming.net/regression-introduction-machine-learning-tutorial/
- https://www.linkedin.com/pulse/python-tutorial-bollinger-bands-andrew-hamlet/
- http://pythondata.com/stock-market-forecasting-with-prophet/
- https://facebook.github.io/prophet/