Springboard Data Science Intensive Program

Capstone Project 2:

Predicting Stock Movements Using News Dataset

By: Ivy Huong Nguyen, Ph. D.

February 2019

Table of Contents

I. INTRODUCTION AND DATA DESCRIPTION

Data analytics is an emerging field which has made its marks in almost every field on this planet. We see its utility and applications spanning from predicting voting result as in the presidential election of 2008 to classifying images for cervical cancer treatment. The plethora of data nowadays enables our ability to discover new connections between one field to another. In this project, we challenge ourselves to uncover a new connection between news data and stock movement. In other words, can we predict the stock performance using news? More importantly, we have to explore which information in the news are most useful to predict stock price performance. By accomplishing this task, we could unveil the predictive power of news, which potentially generates significant economic impact all over the world. This project is inspired by a Kaggle competition (https://www.kaggle.com/c/two-sigma-financial-news) hosted by Two Sigma, a company that is passionate about applying technology and data science to financial forecasts for over 17 years. The market data is provided by Intrinio whereas the news data is given by Thomson Reuters. All data is collected since 2007 to present and contains specific information as follows:

- The market dataset contains financial market information such as opening price, closing price, trading volume, calculated returns, etc. This dataset is a subset of US-listed instruments, which changes daily and is determined based on the trading amount and the ability of information. Returns are calculated based on either open-to-open or close-to-close from one trading day to the next one. Returns that are not adjusted against any benchmark are considered “raw” while those that are adjusted based on the movement of the whole market are labeled as “Mktres” or market-residualized. The number associated to each return variable represents the number of days the return can be calculated over. Finally, returns are tagged with “Prev’ if they are backward looking in time and vice versa. The following variables are always found within the market dataset:
    - `time`:   represents the current time which are taken at 22:00 UTC
    - `assetCode`: a unique id of an asset
    - `assetName`: the name of a company that corresponds to a group of assetCodes
    - `universe`: a boolean indicates whether or not the instrument on that day will be included in scoring and the trading universe changes daily.
    - `volume`: trading volume in shares for the day
    - `close`: the close price for the day and this price is not adjusted for splits or dividends.
    - `open`: the open price for the day and this price is not adjusted for splits or dividends

  - returnsClosePrevRaw1
  - returnsOpenPrevRaw1
  - returnsClosePrevMktres1
  - returnsOpenPrevMktres1
  - returnsClosePrevRaw10
  - returnsOpenPrevRaw10
  - returnsClosePrevMktres10
  - returnsOpenPrevMktres10
  - returnsOpenNextMktres10 next 10-day market-residualized return and is also the target variable in this project

- The news dataset provides information about news articles published about assets. Each asset is identified by an assetCode whereas each company would have multiple assetCodes. The following variables are found within this dataset:
  - time
  - sourceTimestamp
  - firstCreated
  - sourceId
  - headline
  - urgency
  - takeSequence
  - provider
  - subjects
  - audiences
  - bodySize
  - companyCount
  - headlineTag
  - marketCommentary
  - sentenceCount
  - wordCount
  - assetCodes
  - assetName
  - firstMentionSentence
  - relevance
  - sentimentClass
  - sentimentNegative
  - sentimentNeutral
  - sentimentPositive
  - sentimentWordCount
  - noveltyCount12H
  - noveltyCount24H
  - noveltyCount3D
  - noveltyCount5D
  - noveltyCount7D
  - volumeCounts12H
  - volumeCounts24H
  - volumeCounts3D
  - volumeCounts5D
  - volumeCounts7D

The detailed description of these variables will not be provided here and can be accessed via: https://www.kaggle.com/c/two-sigma-financial-news/data. All data is stored and retrieved as Pandas dataframes in the Kernels environment to save memory space.

II. EXPLORATORY DATA ANALYSIS (EDA)

1) EDA on the market data

The market dataset has a total of 4,072, 956 data points with 16 different unique features. However, the 16 unique features are always present. Within the 16 variables, there are four variables that have null values present (returnsOpenPrevMktres1, returnsOpenPrevRaw10, and returnsOpenPrevMktres10). Please note that the listed EDA numbers are not fixed and would changed daily due to the fluctuation of trading volume and the ability of information obtained by the instruments as explained in the introduction above. All variables are float64 type except the assetName and the assetCode, which are category and object, respectively.

The top 10 assets that have the highest trading volume were found to be:

Asset Code	Total Trading Volume
BAC.N	3.601165e+11
GE.N	1.438696e+11
F.N	1.336455e+11
MSFT.O	1.321638e+11
INTC.O	1.233342e+11
CSCO.O	1.144949e+11
PFE.N	1.011882e+11
WFC.N	8.645566e+10
JPM.N	7.891265e+10
AAPL.O	7.288006e+10

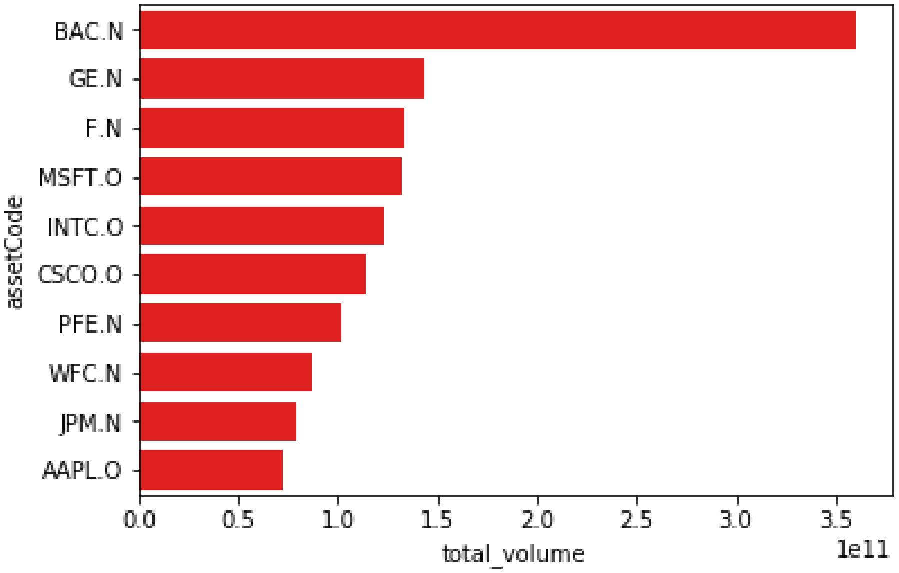
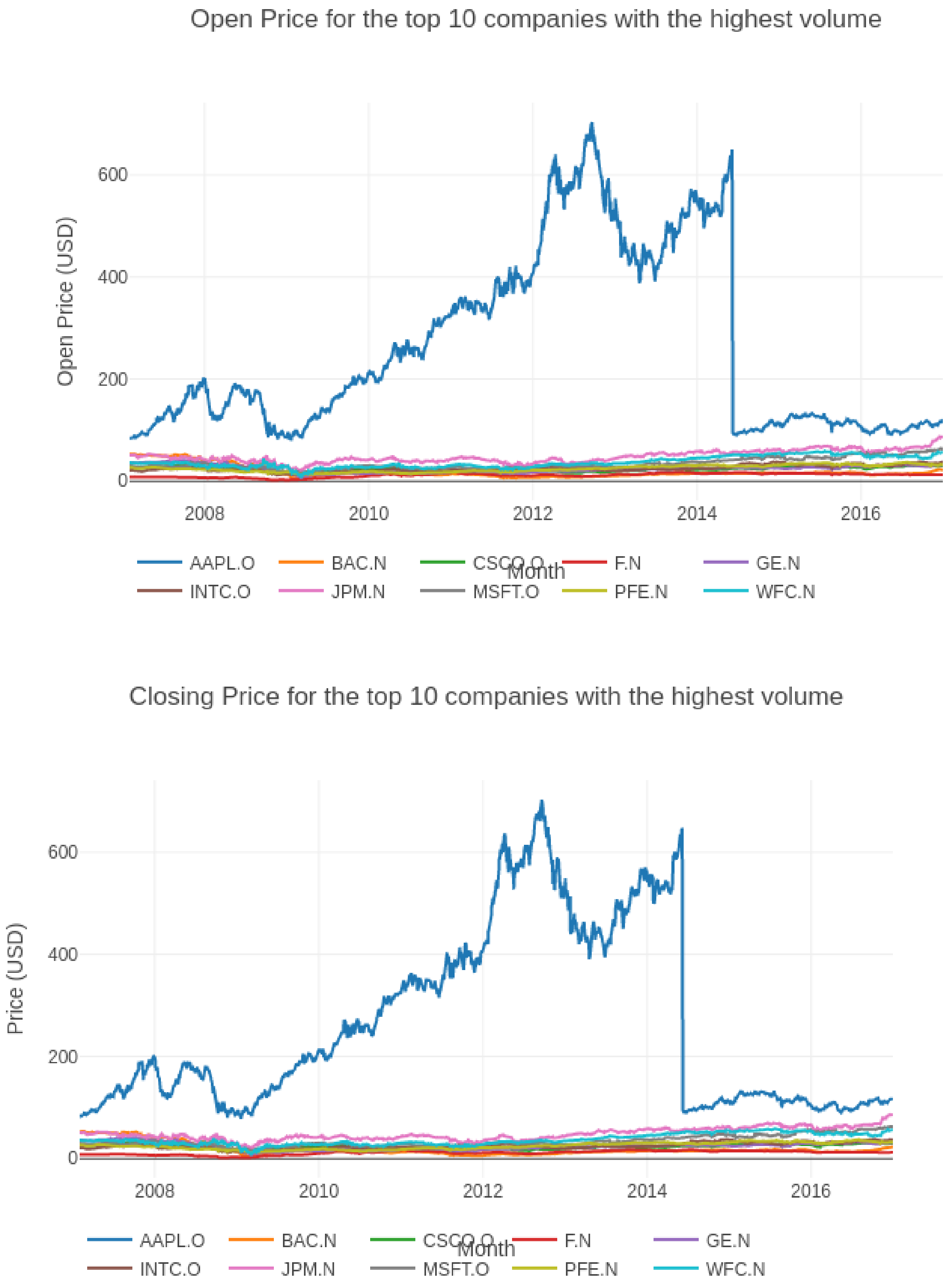


Figure 1. The top 10 assets with the highest total trading volume of stocks. Bank of America corporation has the largest amount of trading in comparison to the others whereas Apple has the lowest amount in the list.

Based on Figure 1, we can clearly see that Bank of America has the largest trading volume in comparison to the other assets. We further expanded our investigation for this dataset by looking at the close price and the open price of these top 10 assets that have the highest trading volume overtime. It appears that there is a large gap in the close price and the open price of the APPL.O asset between May of 2014 and June of 2014. This gap may seem to be unusual at first; however, after searching through some of the Internet data regarding the stock price of APPL.O during this time period, it appears that this gap matches with the 7-for-1 split that Apple initiated on June 9 of 2014. The 7-for-1 split allows Apple stockholders to get 6 more stocks for every stock they own under the circumstance that the APPL.O stock price would drop from $650/stock to $92/stock. However, in order to obtain a better data trends for the top 10 assets with the highest trading volume, the EDA of this dataset was carried on without including APPL.O due to its extreme large stock price before May 2014 (Figure 3).

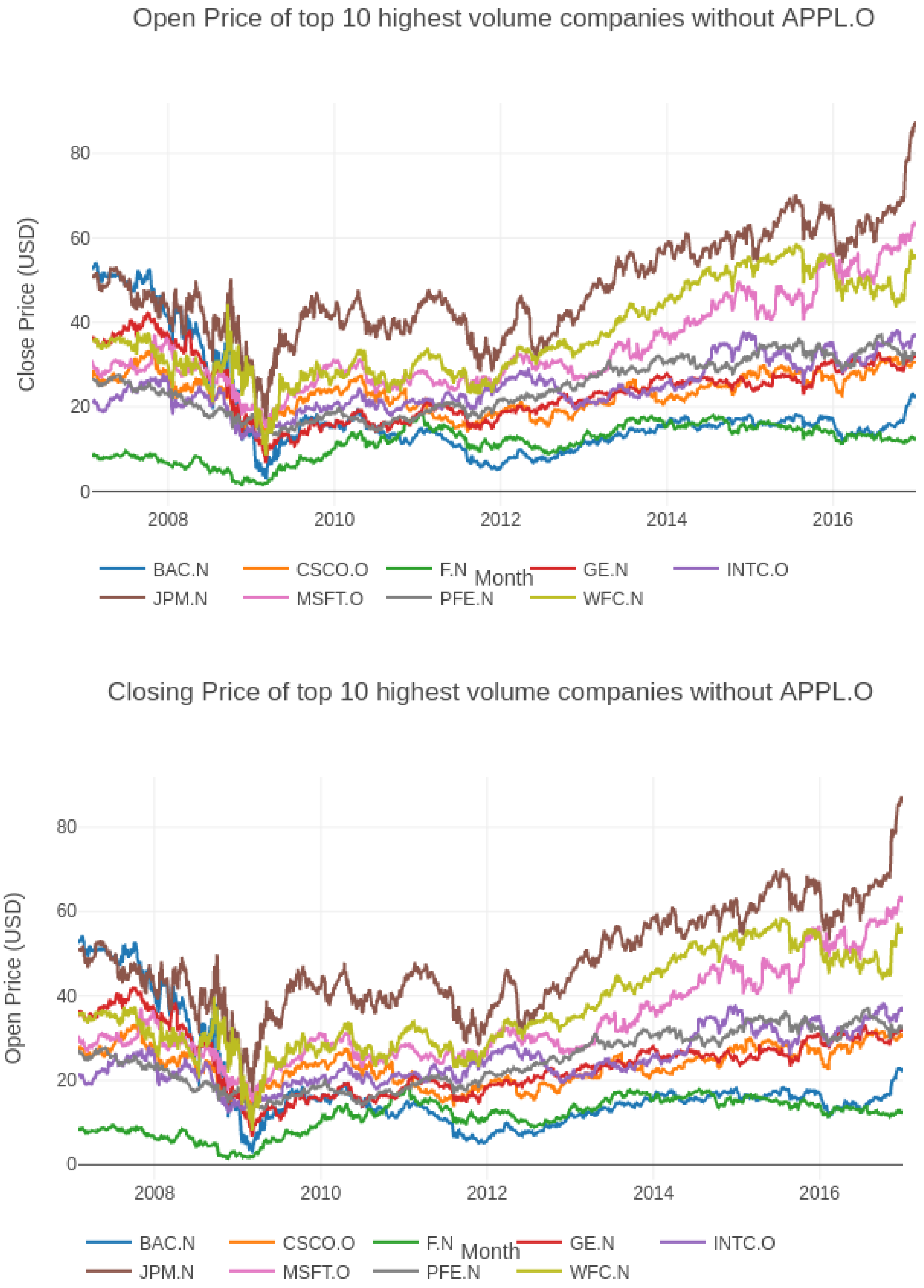
After excluding APPL.O from the list of the top 10 assets that have the highest trading volume, we can clearly see the dropping and rising patterns of the stock overtime. More significantly, we can clearly see the collapse of Lehman Brothers during the year of 2008 in both close price and open graphs. This event is represented by a deep shallow shown in both graphs of Figure 3 starting from 2008 to about 2010. Another significant event that we can also point out in these graphs is towards the end of the year 2011, which marks the Black Monday of 2011 in the finance and investing industry. The Black Monday of 2011 refers to August 8 of 2011, when the U. S. and global stock market crashed and it was also the first in the history the U. S. stock was downgraded.

We further explored the patterns of the target variable of this study, which is the returnsOpenNextMktrest10 variable. In this segment of the EDA process, we shifted our analysis to the other assets instead of only focusing on the top highest trading volume assets. We started the EDA process for the target variable by randomly selecting 20 assets of the dataset and looking at their returnsOpenNextMktres10 fluctuation overtime (Figure 4). The fluctuation of the market-residualized returns based on open-to-open price and within a 10-day interval of the 20 randomly selected assets is represented in Figure 4. Once again we can somewhat see the collapse of the Lehman Brothers back in 2008 and the Black Monday of 2011, which are shown by the variance in the returnsOpenNextMktrest10 variable in these periods.


Open Price for the top 10 companies with the highest volume

Closing Price for the top 10 companies with the highest volume

Figure 2. Close and open price of the top 10 assets with the highest trading volume over time.



Open Price of top 10 highest volume companies without APPL.O

Closing Price of top 10 highest volume companies without APPL.O

Figure 3. Close and open price of the top 9 companies with the highest trading volume (APPL.O was excluded) over time.

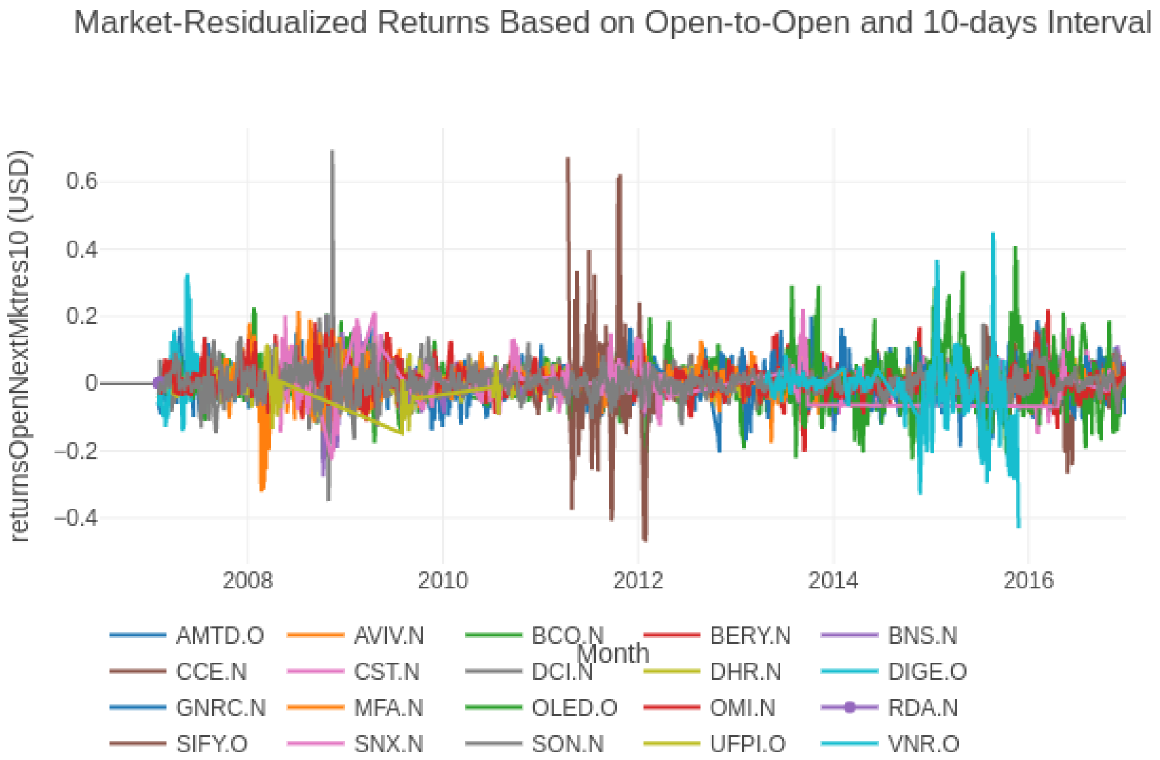


Figure 4. Market-residualized returns based on open-to-open price and 10-day interval of 20 randomly selected assets.

__Feature engineering, data error elimination process__

So far, we have only used features that are already available in the given market dataset. We took one step further by generating new feature in order to gain a better visualization of the patterns within this dataset. The new feature was generated by taking the average difference between the open and close stock price by date. This means we would group the open and close price of each date for all assets, take the difference and then calculated the difference average (Figure 5). According to Figure 5, the fluctuation is greatest from 2008-2010, which is when the collapse of the Lehman Brothers happened. One thing to be pointed out is that the second dip in the year of 2010 could be a data errors or outliers since the Lehman Brothers collapse lasted till the end of 2009 and thus such a large dip should not be caused by this event.

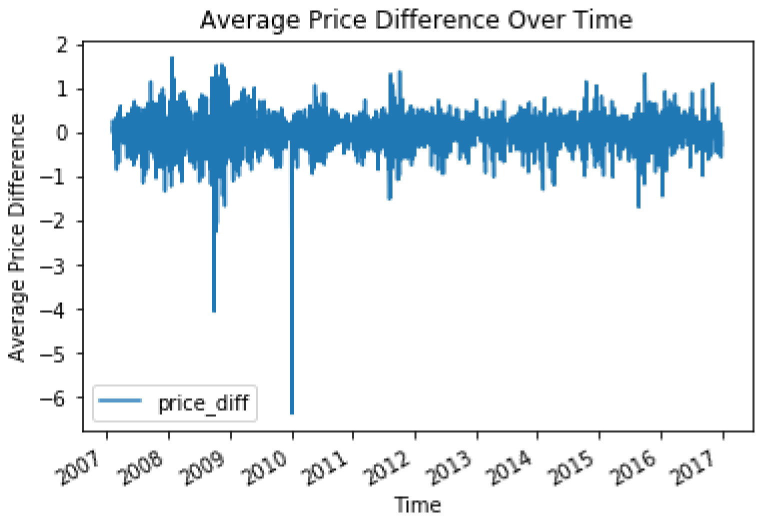


Figure 5. The average price difference overtime between the open price and the close price of all assets.

To further investigate this abnormally pattern, we sorted the price difference column that was generated earlier by taking the difference between the open and the close price. We noticed that the Towers Watson and Co has a huge price difference (almost $10,000 in difference) on the 4th of Jan in 2010. In order to validate whether this huge difference, the given dataset was compared to Yahoo finance's data for the Towers Watson and Co assset. According to Yahoo's finance data, the Towers Watson and Co should have a close price of 43.46 with the adjusted close price of 38.38 on the 4th of Jan in 2010. Based on this information, we were able to conclude that the given market dataset has an error and thus the market data point for the Towers Watson and Co on Jan 4th of 2010 should be elimated fixed before conducting any further analysis. In addition to the Towers Watson and Co, we also noticed there are 4 other data points of 4 different assets with large difference between their open and close stock price. These are: Bank of New York Mellon Corp on the 29th of September in 2008, Apria Healthcare Group Inc. on 5th of June in 2008, Cephalon Inc. on the 5th of June in 2008, and Archrock Inc. on the 27th of September in 2007. These data points, however, are not errors since the time frame matches with when the Lehman brothers collapse happened.

A better way to find all errors in this dataset would be to take the ratio between the close stock price and the open stock price of the same day for each asset. If the close price increases or decreases more than twice the open price then we classify that data point as an error and replace it with the median of the open price or the median of the close price. Median was chosen over mean as the imputed value because of the skewness of the distribution in the given dataset in term of the close/open price. The average price difference between the close and open stock price was replotted after all suspicious error data points were imputed with the median value of the corresponding column, as shown in Figure 6.

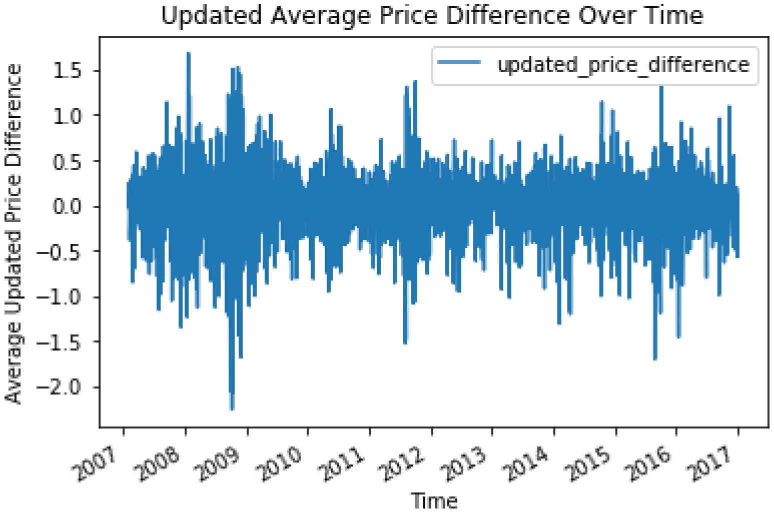


Figure 6. The updated average price difference overtime between the open price and the close price of all assets after imputation process.

After fixing errors identified in the market dataset, we could see the clear drop of the Lehman brother collapse without a second deep in 2010. Additionally, we could also see the dip happened in August of 2011, which represents the Black Monday of 2011. There are other significant financial events that could be pointed and compared to public data via this new plot.

<u>2) EDA on the news dataset</u>

The news dataset contains information regarding any news articles or alerts that are published about the assets listed in the market dataset in Part 1) of this section. The news dataset has a total of 9,328,750 data points with 35 different unique features. Within these features, there are 5 categorical features, 1 boolean feature, 3 datetime features, 3 object features, and 23 float/int features. The EDA process for this dataset was focused on 6 different segments including:

- Sentimental Rating
- Urgency
- Word Counts
- Word Counts over Sentence
- Number of News by provider
- Headline tag

a) Sentimental Rating

There are 3 different categories under the sentimental rating segment: positive, neural, and negative. Each category is given a probability. We conducted a overall sentimental rating comparison among these three categories by taking the mean probability of each category (Figure 7).

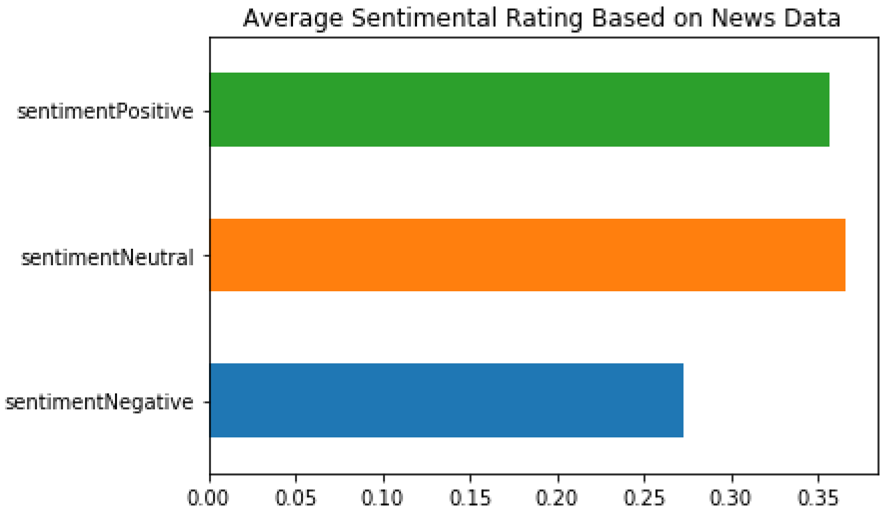


Figure 7. Average Sentimental Rating Based on News Dataset

It appears that the sentimental rating favors towards the positive and the neutral side rather than the negative rating side. We extended the analysis for this feature to see which asset has the highest positive sentimental rating among the others. Trimax Corp. has the highest positive sentimental rating among all assets whereas Stewart & Stevenson LLC ha the highest neutral sentimental rating. Atlas Acquisition Holdings Corp has the highest negative sentimental rating among all.

b) Urgency and word counts

Urgency is used to differentiate the story type of the news where 1 is alert and 3 is article. Due to the absolute classification of this feature, we only counted the number of 1, 2, versus 3 to see how diverse our data is. We counted the number the news that is classified under 1, 2, or 3 for the urgency feature, as shown in Figure 8. It appears that more than 6 millions of these news are articles and more than 3 millions of them are classified as alert .Interestingly, the urgency ‘2’ is almost never used in comparison to the other categories. However, please keep in mind this dataset is updated as we speak so that could be the case at the time this EDA was conducted.

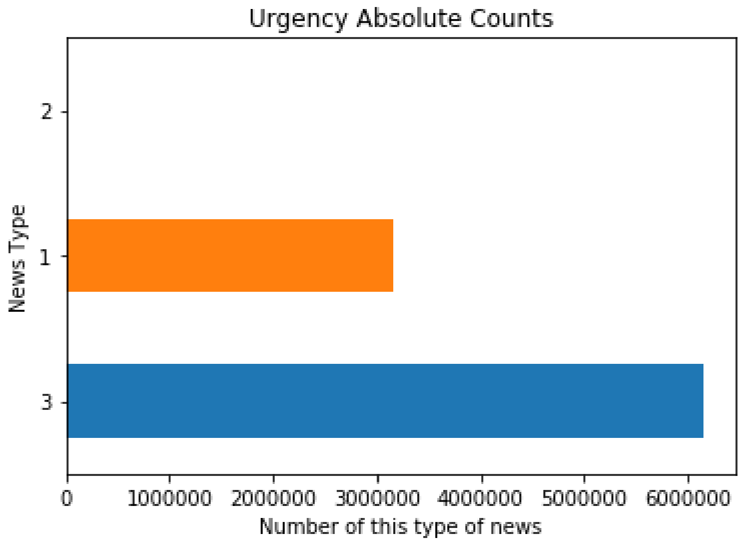


Figure 8: Urgency Counts for the entire news dataset

We further our investigation for this dataset by looking at the number of word counts reported for each news. Below are the two graphs generated for the number of word counts before and after we applied a filter, which is to limit the number of word counts to 2000 (Figure 9). A descriptive statistical summary was also performed for the number of word counts. The average ratio number of word counts for the news dataset is 580 words per news with the median value of 259 words. A large portion of this feature is outliers, which can be seen via a box plot (Figure 10). It should be noted that the news data has a right-skewed (positive skewness, median < mean) distribution in term of word counts.

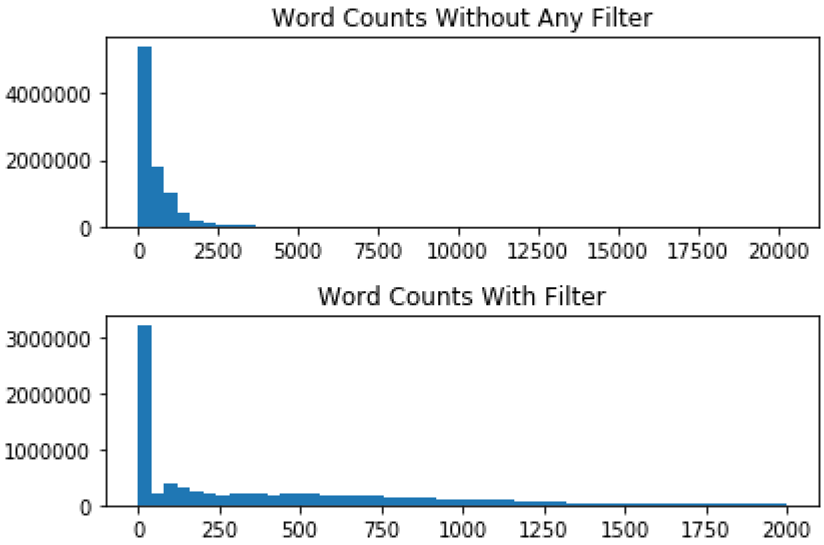


Figure 9: Histograms show the number of word counts. The top histogram was generated without filter whereas filter (number of word counts < 2000) was applied for the bottom histogram.

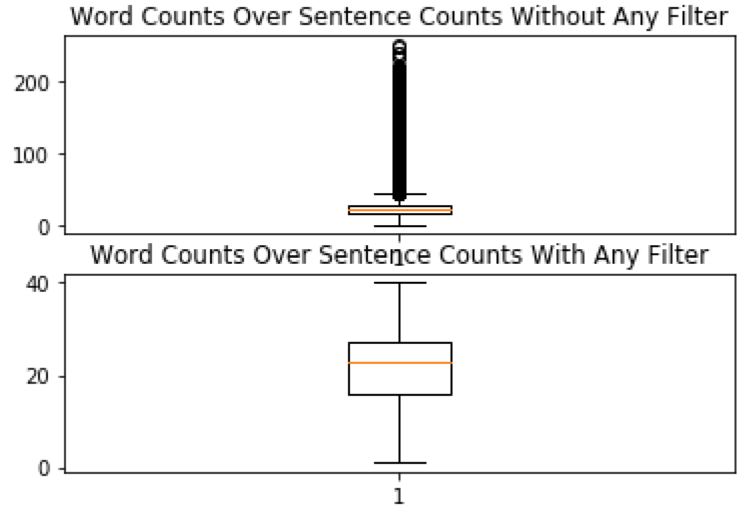


Figure 10. Box plots show the distribution of the number of word counts over Sentence Counts with and without filter.

Based on the ratio between word counts and sentence counts, we were able to obtain the average number of words that a sentence would have, which is 23 words per sentence. The median number of words per sentence is 23. The outliers can be seen clearly on the box plot before any filter being applied.

c) Providers and headline tags

Beyond the above features, we also explored the number of news provided by each provider and investigated the details of headline tags feature. It appears that RTRS, Reuters, is the most common provider in these news that commonly have untagged headline. These trends are shown in Figures 11 and 12.

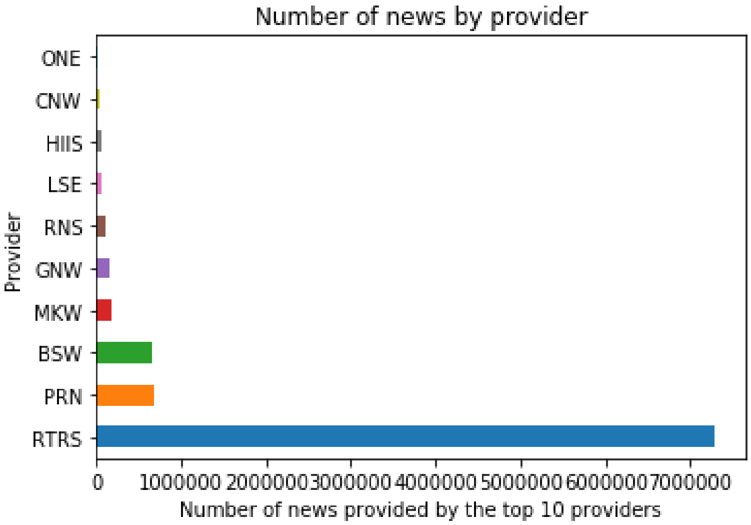


Figure 11. Top 10 providers that have the highest number of news reported.

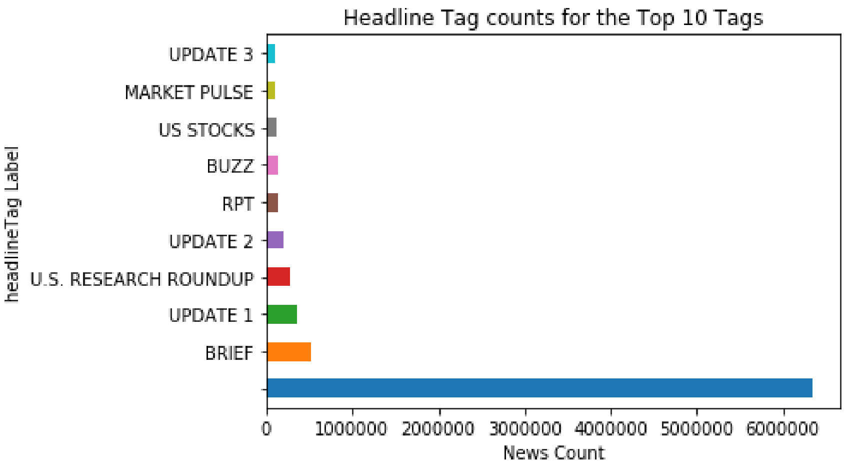


Figure 12. Top 10 headline tags that have the highest number of counts.

We also explored the impact of each word in the headline tags of the top 10 companies with the highest trading volume via visualization generated through Scattertext or wordcloud. In this project, we chose to use WordCloud due to its familiarity. The WordCloud generated image shows “Reuters” and “Insider” to show up a lot in all headlines. This is not very surprising since we found out earlier that Reuters is also the most common provider of these news. Two banking insinuations that stand out from this analysis are Wells Fargo and Bank America, which could imply their relation to the volume of trading came from these asset codes.

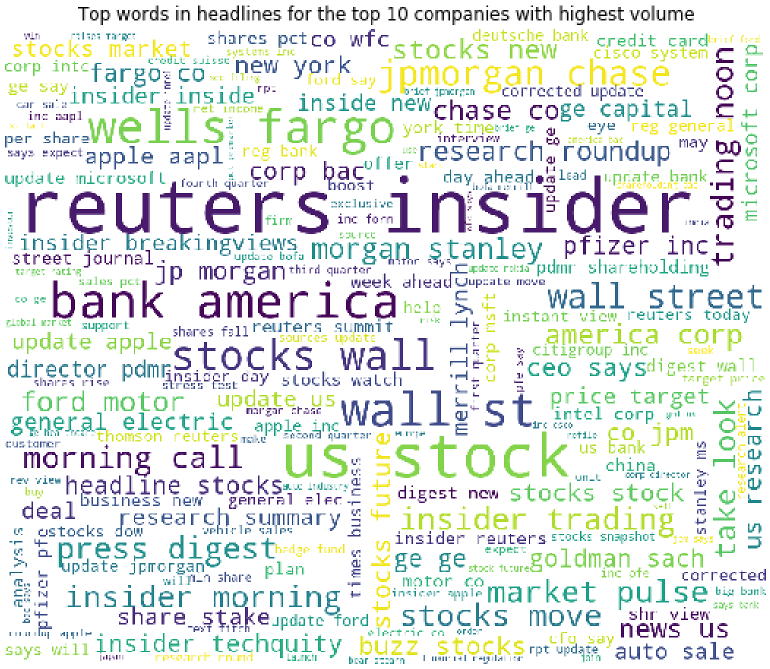


Figure 13. WordCloud generated image shows the impact of each word presented in the headlines of the top 10 companies that have highest trading volume.

III. DATA PROCESSING

This section will explain how we would merge the two given datasets (market and news) before proceeding to our model selection portion. A more detailed workflow is shown in the model_selection_Capstone2.ipynb file. There are three different features that we could use to join these two datasets together: assetCode, assetName, or time. The data of this project can only be accessed via a Kaggle Kernel, which has compute constraints. Due to these constraints, we cannot simply merge the two datasets as given without any transformation or data processing. The first step was to eliminate potentially irrelevant features in the news dataset. The following columns were eliminated: `sourceTimestamp`, `firstCreated`, `sourceId`, `headline`, `takeSequence`, `subjects`, and `audiences`. Since each assetName could have multiple assetCodes in the news dataset, we decided to unstack the news dataset by giving each assetCode its own data point. The unstacked news dataset is then regrouped by date and assetCode since there could be multiple news available to each assetCode per day. We used 'mean' as an aggregation method during our re-grouping process for the news dataset. After unstacking and regrouping the news dataset, we merged the new news dataset and the market dataset by using the two features: assetCode and time.

IV. MODEL SELECTION

The model selection process consists of 2 different stages: a) comparing the performance of the Logistic Regression model versus the performance of the LightGBM model b) tuning the hyperparameters for the best model.

a) Comparing Model Performance

Logistic regression model is the first choice when it comes to a binary classification problem. The logistic regression is named after the function that was used at the core of the method, the logistic function. This particular function is developed to describe the properties of population growth in ecology. The most seen graph that has been used to describe the logistic function is an S-shaped curve which can take any real-valued number and map it into a probability between 0 and 1. This specific characteristic of logistic function makes it a perfect candidate for the stated problem in this project. However, logisitc regression model may not be the best model to handle a large size of dataset as in this case. LightGBM model is a relatively new but efficient model that is based on tree-based learning algorithm. LightGBM model is a gradient boosting framework which grows trees vertically or leaf-wise. LightGBM is preferred as 'light' because of its high speed and its ability to handle large size dataset. These facts make LightGBM model also a good candidate to solve the problem since we are limited by the amount of run time (9 hours) on Kaggle kernel and memory space.

Before constructing any model, we would need to split our joined news_market dataset into two portions: 80% of the joined dataset is for training and 20% of the joined dataset is for testing. This type of split was accomplished by the train_test_split of the ‘model_selection’ package from sklearn. The Area Under the ROC Curve (AUC) score of the logistic regression is 51.41% whereas that of the lightGBM model is 56.62% (Figure 16). The distribution of feature importances of the logistic regression model and the lightGBM model are shown in Figures 14 and 15, respectively. The top three features that have the highest contribution to the logistic regression model are ‘open’, ‘close’, and ‘ sentimentWordCount-mean’. The lightGBM model has a totally different distribution of the feature importance, in which the top three most important features are: returnsOpenPrevMktres10, returnsOpenPrevRaw10, and volume. It should be noted that all features contribute to the contruction of the lightGBM model whereas some features are not useful (do not contribute) in the construction of the logistic regression model. We decided to further tune the LightGBM model due to its higher AUC score after the training process.

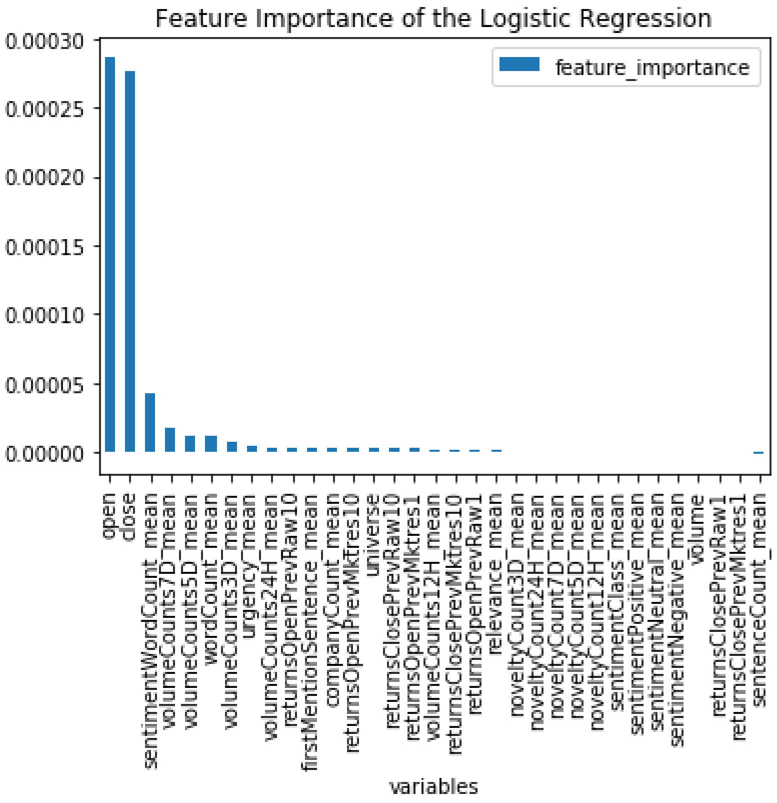


Figure 14: Bar plot of feature importance of the logistic regression model

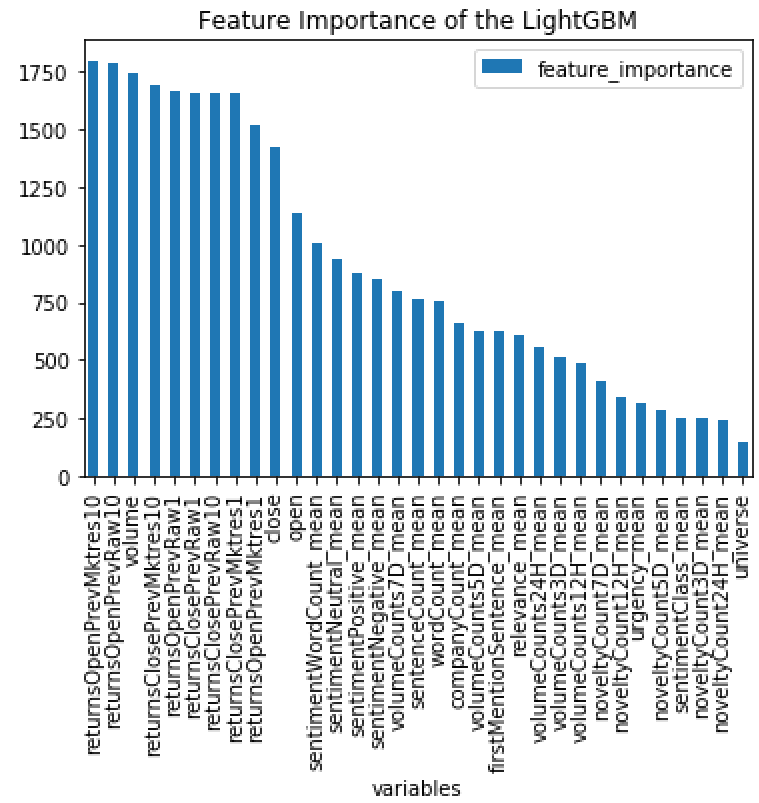


Figure 15: Bar plot of feature importance of the lightGBM model

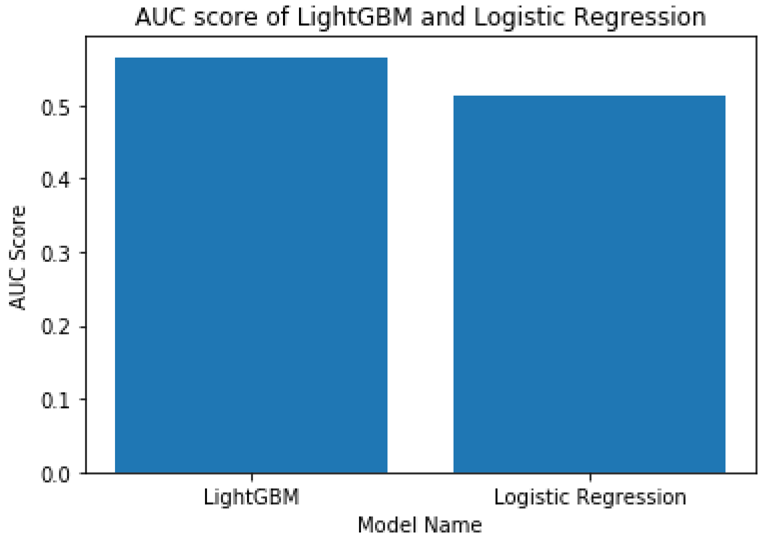


Figure 15: Bar plot shows the comparison of the AUC score between the logistic regression model and the lightGBM model.

b) Tuning LightGBM Model

To further optimize the accuracy of the LightGBM model, we can vary the following features

- use large max_bin

- use small learning_rate in addition to increasing the num_boost_round
- use large num_leaves
- use bigger training dataset (not applicable in our case)
- try using 'dart' method
- or try using categorical features directly

In this case, using large max_bin can significantly slow down our working process due to the amount of data points that were given in this project while using a large num_leaves may cause over-fitting. The last option, which is to use categorical features directly are not really appropriate in this scenario since the data-preprocessing already eliminated the categories due to its irrelevance to the ability to predict the market returns. Thus, we can either combine using small learning_rate with increasing the num_boost_round or use 'dart' method to see if one of those methods would help us improve the accuracy of the LightGBM model. We used the AUC score to compare the three lightGBM models as shown in Figure 17. According to the result in Figure 17, the two different approaches that we did for tuning the LightGBM model did not give us much of an improvement. Therefore, we can just use the original model (lgb_model) to make predictions for this study.

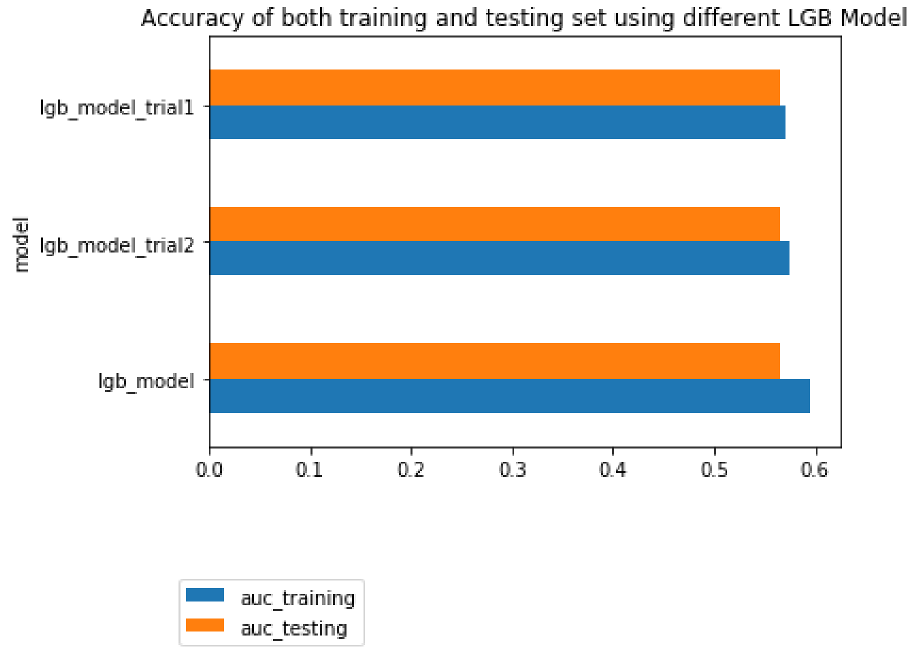


Figure 17. AUC scores of the three lighGBM models.

V. PREDICTION AND CONCLUSIONS

The prediction process was accomplished via Kaggle kernal as shown in the ‘Part2: Model_Selection_and_Prediction.ipynb’ notebook attached in the repository of this project. Overall, the lightGBM model gives us a better AUC score in comparison to the logistic regression model. The accuracy score for the testing dataset is not available at this point since this Kaggle competition is still going at the time the project was completed.