1. Introduction

This is inspired by the Kaggle competition by Two Sigma. The challenge is to use news analytics to predict stock price performance. There is a large amount of market and news data available but the difficulty is finding relevant trends within news data that predict how the stock market will respond. As one would expect, this also means that there may be a large signal-to-noise ratio when it comes to using news analytic data.

A large number of financial institution's primary function is to understand and predict stock market trends. This project would be directly applicable to such institutions. However, the client base for this project goes beyond financial institutions. Any entity that makes use of the stock market can strategically use results from the project to their advantage. By correlating news analytics with stock prices one can make better decisions about one's stock portfolio; one can be better informed as to what stocks to buy and sell and at what times. Ultimately, this project has significance on the global economy as a whole.

2. Data Wraggling/Cleaning

The data used for this project comes directly from the Kaggle API. The news analytics data was provided by Thomson Reuters. Copyright ©, Thomson Reuters, 2017. While market data is provided by Intrinio. Because data sets are not available real-time from these sources on Kaggle the data date ranges from 2007 to late 2018. This should provide significant enough data for training, validation and test sets. The market data has typical measurements such the company names, opening and closing price, etc. The news data contains information such as when the news was sourced, the subject, the relevant companies and how the source is related to the company (e.g. was the article positive for the company). Two data sets (one for the stock market data and one for the news data) were downloaded from the Kaggle kernel using urlretrieve. A summary of each data set is as follows.

2.1. Market Data

- Dataset contains 4,072,956 rows and 17 columns.
- Data contents include date/time stamps for the market data, a company name and identification code, and data pertaining to stock market measurements (e.g. opening and closing prices, volume, returns over 10 day period, etc.)
- There are null values in several columns. In the returnsClosePrevMktres1 and returnsOpenPrevMktres1 columns there are 15,980 null values in each column. In the returnsClosePrevMktres10 and returnsOpenPrevMktres10 there are 93,054 null values in each column. These values are not necessarily needed to model the system and therefore there is little need to change the null values.

2.2. News Data

- Dataset contains 9,328,750 rows and 36 columns.
- Data contents include date/time stamps for news segment, the company name and identification code that the news segment is referring to, and details of the news segment (e.g. word count, headline, type of segment, measurement if the segment was positive, negative or neutral toward the company, etc.)
- There are null values in two columns. The headline column has 73,960 null values. The headlineTag column has 6,341,993 null values. Both of these columns will not

play a role in the mathematical modeling and there is no logical way of filling them in with alternative values. Hence, the null values are kept in the dataset.

2.3. Additional cleaning and Merging Data

- Each dataset was loaded into python using pd.read_csv(file_name, parse_dates =[1]) so that the date/time was read as a datetime object.
- The null values (see description of each data set above) were all in columns that will not be used in the mathematical analysis. Also, some of the null values (such as headlines) cannot logically be replaced with anything therefore these values were kept null.
- The news data had outliers that are deemed to be statistically realistic and therefore there is no need to adjust for these outliers. The market data had several hundred outliers in several columns that seemed to be a result of miscalculations or incomplete information in calculations. The market data had columns such as opening and closing stock data that had no outliers. It is noted that the outliers are in columns such as returnsOpenNextMktres10 or returnsClosePrevMktres10 which are columns that are calculated from opening price data from the date of the row under observation and data from a date before or after the observation row date. Sometimes a company will appear or disappear from the data set over time; reasons could include bankruptcy, the make or drop from the ranking of companies on the list or the company is newly formed. This means that if a company appears or disappears from the list of companies in the data set there will be an error in calculated some of these values. Hence, the outliers in such cases are not realistic values but rather are calculation errors. It is found that there are 248 outliers in the returnOpenNextMktres10 column; this column is a key value that will be used when modeling. There are two ways one could deal with these outliers. One, the outlier can be replaced with a mean value for that specific company. Two, the entire row can be eliminated completely from the data set. Since there are over 4 million rows of data and only a few hundred outliers it seems reasonable that eliminating the outlier rows would not alter analysis results significantly. Also, there is no basis for changing the value of the outlier therefore it is more logical to remove the data. Hence, the outlier rows were eliminated from the data set.
- The last step of data wraggling was to join the two data sets. The logical way to join the data sets was to join them along the company name and the date/time. The challenge is that the news data and the stock data have timestamps at different times during the day. Hence, they will never exactly line up and therefore the data cannot be directly joined using the date/times provided. To join the data effectively we first assume that the data we want to join is only day-to-day and the time of day is irrelevant. Therefore, the datetime variables in the market data and the news data are converted to datetimes with the date only and no timestamp. Then the market and news data were merged along the company names and the date. This resulting dataframe was then saved the a csv file to be used further.

3. Exploratory Data Analysis

3.1. Stock Market and Individual Company Trends

We can look at the opening stock prices (almost identical to closing prices) on average for the entire market in the dataset over time (Figure 1). Notice, there are short-term fluctuations that look like noise but also long-term trends that probably are related to large scale events. For instance, the dip in the stock tickers prices in 2008-early 2009 is a result of the 2008 recession. Also, there are smaller declines in stock prices like in early 2016 that was a result in crashing of crude oil prices. These are a result of large-scale events that globally effect the overall market. The ticker price for an individual stock will feel the effects of these large-scale events but also have fluctuations due to the company health.

Next to the stock market graph is an example of a company chosen to be used for display of individual company stock data. In this case Agilent Technology Inc. is used. Above the opening stock price for the company is shown versus time. As seen in the plot the stock prices tend to look like they are noisy from day-to-day with long term trends being the predominate trends. In comparison to the overall stock market data it is seen that the major events seen in the market data are also seen in the individual company. However, there are additional fluctuations seen in the company data that are due primarily to the individual company's health. The challenge from seen in this data is that the news data will provide information to predict these long-term trends and possibly short term trends. From this graph it is clear that short term trends may be more difficult to predict because of the general noisy nature of day-to-day trading. As will be seen in the statistical analysis there is a time lag to determine what is long term versus short term for individual companies.

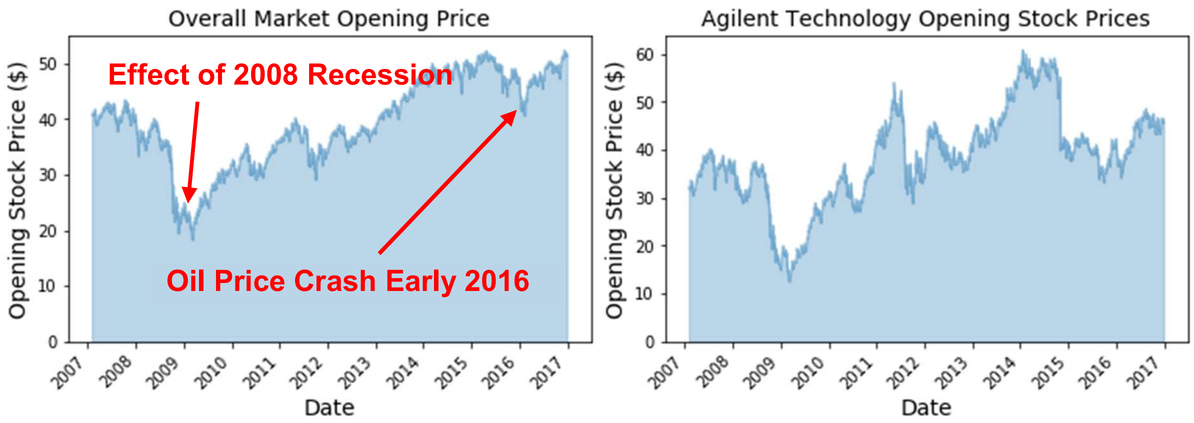


Figure 1 Left: Stock market average opening ticker price over time. Right: opening ticker price for Agilent Technologies Inc over time.

3.2. 10-day Market Adjusted Leading Return (Modeling Target)

Here the 10-day market adjusted leading return is graphed versus the date (Figure 2). This value is indicative of the amount the stock price changes over the course of 10 days. As seen the value fluctuates wildly from day-to-day. If instead the 10-day market adjusted leading return is graphed as a histogram we see that the value looks nearly normal about a value of 0. The distribution actually looks slightly skewed to larger values which makes sense. Because the stock prices ultimately increased over the timeframe in which the data was taken one would expect that there would be more values of the 10-day market adjusted leading return that are positive. For this particular stock the overall stock price increased over the timeframe hence the distribution is skewed to larger values of the residual.

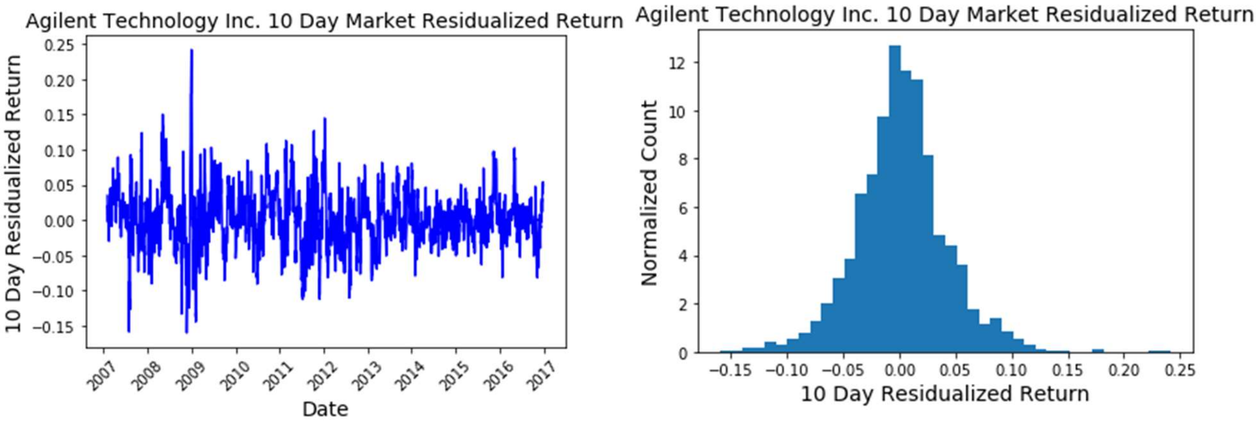


Figure 2 The 10-day market adjusted leading return graphed over time (left) and as a histogram (right) for Agilent Technologies Inc.

Here is an example of a stock with a very different market trend (Figure 3). Ultimately Twitter Inc. stock dropped over time. When comparing the 10-day market adjusted leading return of Twitter to Agilent Technology we see that Twitter has a similar skewed distribution except it is skewed toward negative values. Again, this makes sense because the stock price of Twitter Inc. dropped over time and therefore there should be a tendency for the distribution of the residual to be negative.

Even better than comparing just two company histograms, we can make violin plots of the 10-day market adjusted leading return for several different companies. Notice, many companies are nearly symmetric about zero but are slightly skewed depending on if the company stock had increased or decreased over time. Also, the range of the residual values vastly varies depending on company. This poses an additional challenge because it suggests that some companies are more prone to dramatic changes than others which signifies volatility. Also, if one wanted to use such information to make money off of short trading then the long-term trends would be less important. For instance, Twitter ultimately lost stock value over time however there is larger variance in the 10-day market adjusted leading return which means if one could predict the 10-day leading return perfectly then more money could be made from Twitter stock than Agilent Technology stock.

3.3. News Sentiment

Here the sentiment class is shown over time for Agilent Technologies Inc. (Figure 4). The values of 1, 0 and -1 correspond to a news segment with a sentiment of positive, neutral and negative toward Agilent Technology Inc. Notice, that over time there seems to be more positive news segments than negative news segments. Otherwise, the graphic is unremarkable. The sentiment values are explicitly be compared as well. The sentiment positive and sentiment negative values (both of which range from 0 to 1.0) are shown for all news segments related to Agilent Technology Inc. Notice, generally a high sentiment positive values corresponds to a lower sentiment negative value and vice-versa. This makes

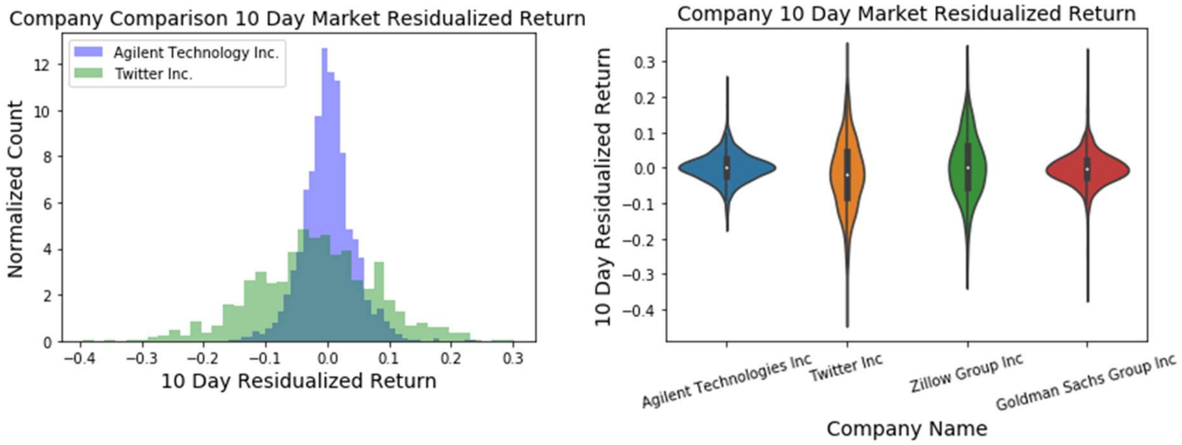


Figure 3 Left: Histogram of 10-day market adjusted leading return for Agilent Technologies Inc and Twitter Inc. Right : Violin chart of 10-day market adjusted leading return for different companies.

sense because if an article has a high positive sentiment then one would expect that there is less of a negative sentiment associated with it. However, some news articles are not clearly positive or negative based on these values. Hence, when the sentiment class is chosen there are a fraction of articles that are either neutral or can are not clearly positive or negative toward Agilent Technology Inc. This is a compounding factor that will come into

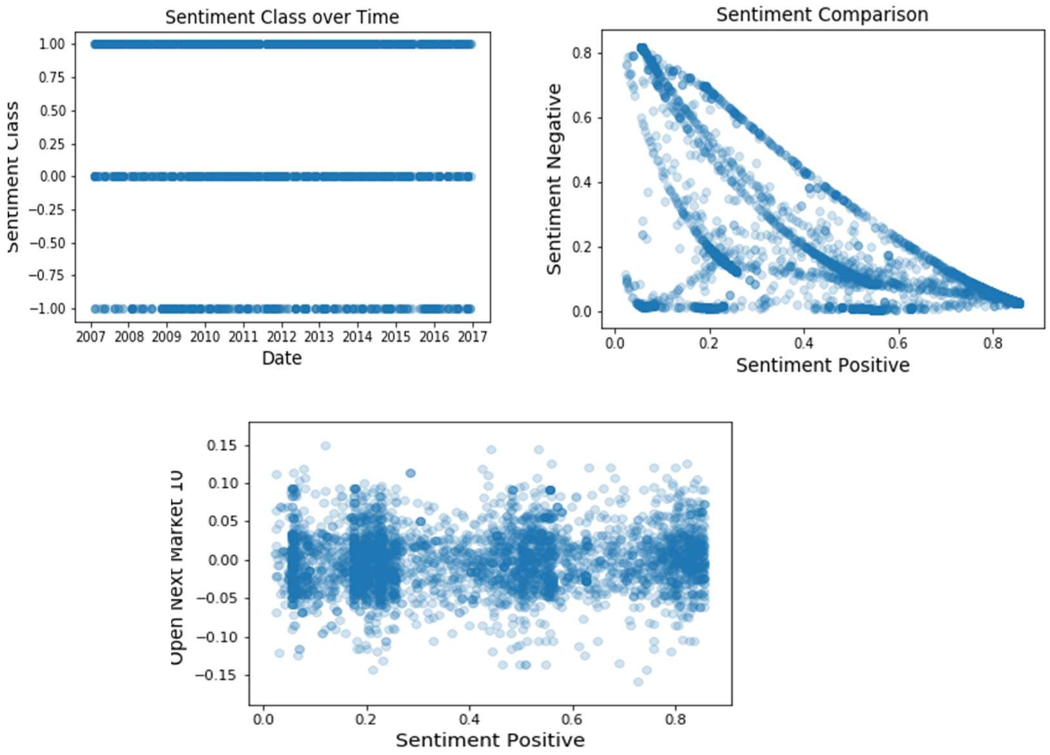


Figure 4 Sentiment class over time (upper left), sentiment negative versus sentiment positive (upper right) and 10-day market adjusted leading return versus sentiment positive for Agilent Technologies Inc.

explanations for errors in any predictive model that would be developed later one. If one were simply trying to find a relationship between news segment sentiment and the 10-day market adjusted leading return then the first step would be to visualize the data. Notice, when graphically observing sentiment positive and 10-day market adjusted leading return there is no apparent correlation. Though it makes sense that these two variables would be correlated this graphic suggests that the correlation is much more complex than a simple linear trend.

Here the distribution of 10-day market adjusted leading return is graphed for each sentiment class (positive, neutral and negative)(Figure 5). Notice, there are slight difference between these distributions however to the naked eye the differences are subtle. This again, speaks to the necessity of more complex modeling to uncover the differences between different types of news reports. The mean of the residual is also calculated for each sentiment class and displayed as a bar chart above. As one logically expects, a positive news sentiment has a positive mean residual and a negative news segment has a mean negative residual. The neutral news segments have a slightly negative residual but not as significantly negative as the negative news segment.

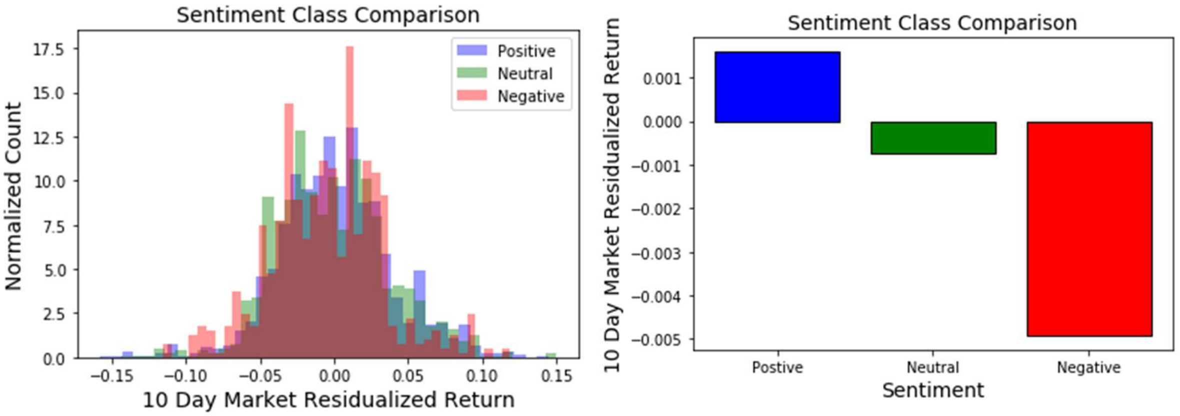


Figure 5 Left : HIstograms of the postive, neutral and negative sentiment articles with the respective 10-day market adjust leading return. Right : Bar graph of the average 10-day market adjust leading return for each sentiment class.

3.4. News Provider Effect for Example Company (Agilent Technologies Inc.)

Taking a step further and look at the effect of the news organization on the 10-day market adjusted leading return(Figure 6). Notice, the distribution is different for different news organization. When taking the mean 10-day market adjusted leading return for each news organization for Agilent Technology Inc. there is a relatively tight range of residual values. CNW, which is the largest mean value is probably not a good idicator because there were very few data points going into that calculation. If we exclude that news provider then the remaining news providers show residual absolute values within 0.01. This is small compared to the range of residual values shown in the violin plot in section "10 Day Market Return for Different Companies." This suggests that no given news organization was involved with large changes in stock prices compared to other providers.

The news sentiment class is shown for different news providers for Agilent Technology Inc. Remember, 1, 0 and -1 correspond to positive, neutral and negative news segment respectively. Once again, the violin plots clearly show that there is a wide distribution of news sentiment classes between the news organizations. For instance, GNW only has positive and neutral news segments on Agilent Technology Inc. while RTRS looks like there is almost an even split between positive, neutral and negative news segments. When taking the mean of the sentiment classes it is seen that all of the news agencies had either neutral or positive news segments on average about Agilent Technology Inc. This is probably a testament to the company itself rather than a bias in the news. As can be seen in the bar chart, even though all news organization were on average positive about the company there was a varying degree to how positive they were. Once again, comparing GNW to RTRS it is seen that on average the sentiment score of 0.857 was associated with GNW while RTRS had an average sentiment score of 0.121. Considering different news organization bias and their effect of the market will be something to consider moving forward in the modeling.

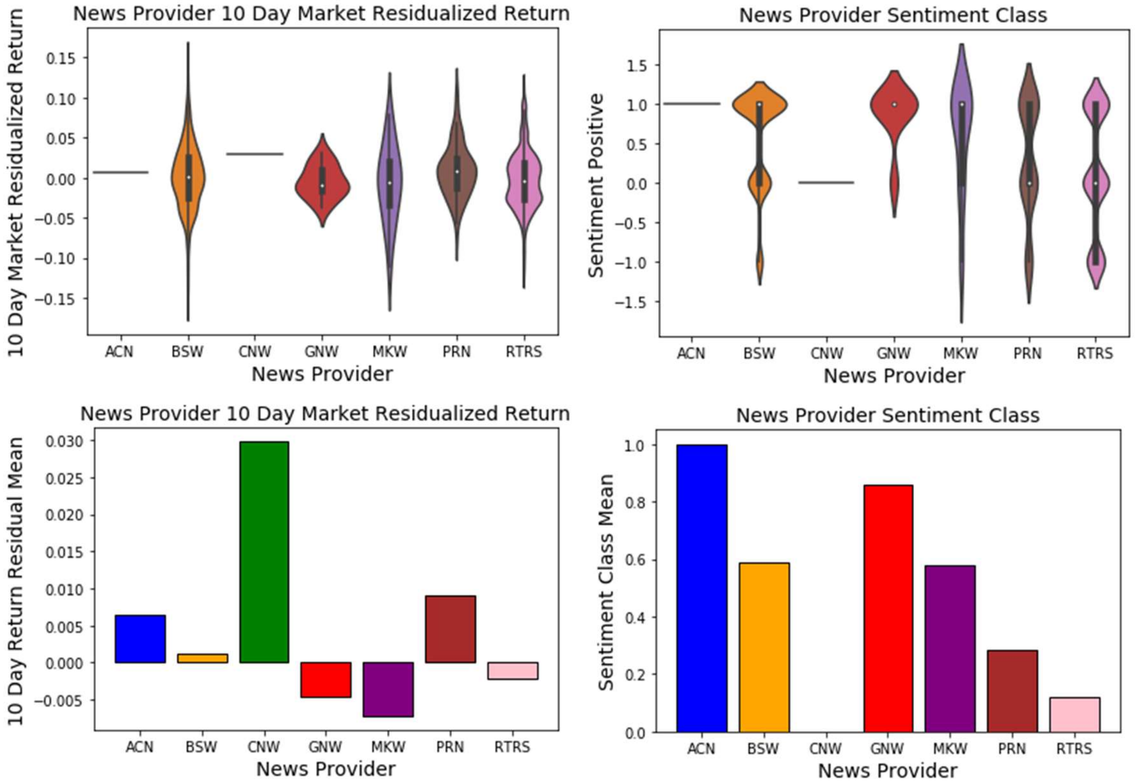


Figure 6 Violin plots of the 10-day leading return (upper left) and the sentiment class mean (upper right) for different news providers. Bar plots of 10-day leading return mean (lower left) and mean sentiment class for news providers. Agilent Technologies Inc.

3.5. Statistical Analysis (Short-term Return Autocorrelation)

There are a large number of exploratory data analysis techniques to be drawn from the stock market data and the news data. Here a statistical analysis is focused on drawing conclusions between correlations between different features in the data set and reporting on statistical measurements for individual companies. Before exploring these data it would be of interest to understand how short term returns vary day-to-day. Graphing stock market opening or closing prices over time makes data look as if the short term returns or random and independent. This can be checked by performing a ljung-box test and comparing the p-value to a threshold value. If the p-value is lower than the threshold then the null hypothesis fails and therefore the data is not considered random noise. This is done for Agilent Technologies and Apple Inc.

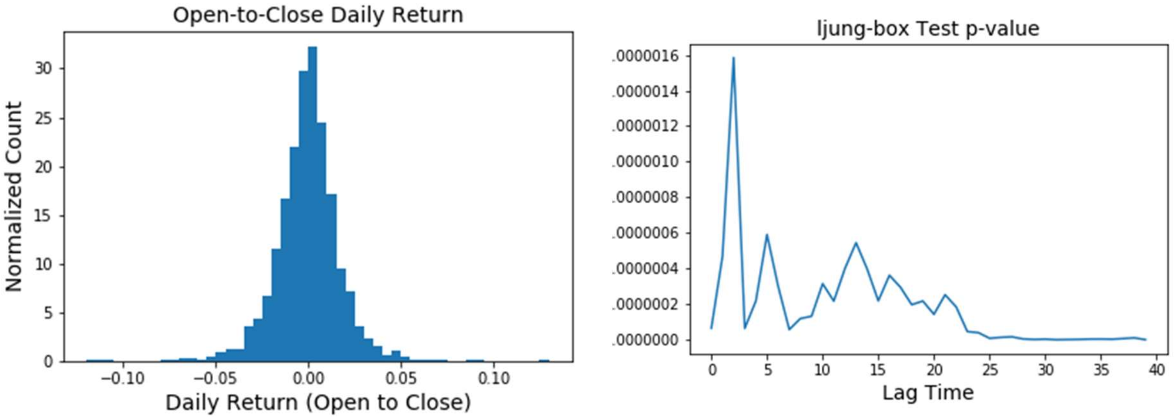


Figure 8 Left : Histogram of daily returns. Right : Ljung-box test p-values versus lag time. Apple. Inc.

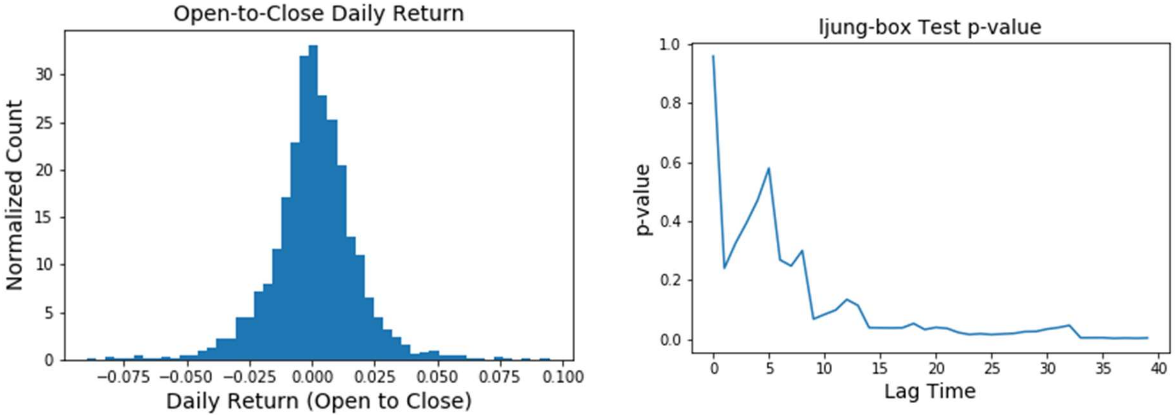


Figure 7 Left : Histogram of daily returns. Right : Ljung-box test p-values versus lag time. Agilent Technologies Inc.

In the figures is a summary of the daily returns and the p-values for the ljung-box test performed on daily returns for Agilent Technologies Inc.(Figure 7) and Apple Inc.(Figure

8). Notice, at short lag times <10 the p-value is significant for Agilent Technologies and therefore the null hypothesis fails to be rejected meaning the daily returns at these lag times can be considered random noise. However, at larger time lags the p-value drops which indicates that the null hypothesis will eventually be rejected for large enough lag times for Agilent Technologies Inc. Hence, at long lag times the daily returns do not show random behavior. This trend is statistically interesting and shows a threshold for a timeframe in which daily returns are no longer random and independent for some companies. Other companies, such as Apple, do not have significant p-values for any lag time and therefore the daily returns are not considered noise and show an autocorrelation. Hence, the trading strategy will value depending on the company. This is an interesting result that could be used in modeling and is included in the future improvements section.

Based on the ljung-box test there is a suggestion that there is a non-random change in the returns over longer lag times. We can plot the returns for different time periods on the same histogram to get a sense as to what the difference is with different return time periods. The histograms below show the differences in returns using daily pricing, the return over the previous one day and the return over the previous 10 days. Notice, as the return time increases the spread of returns increases. Also, in this case, the average return steadily increases from 1.98e-4 daily, 4.93e-4 previous 1 day and 4.40e-3 for the previous 10 days.

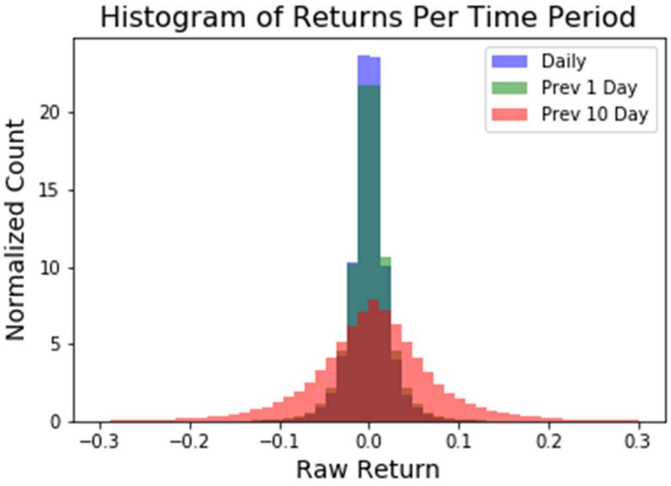


Figure 9 Histogram of returns for diffeent time periods.

3.6. Cross Correlation Between Features

When modeling a complex system of features it is important to understand what, if any, correlations exist between features. Here a series of features was chosen and then pair-wise correlations were calculated between each set of features (Figure 10). Several features from the original market and news data sets were chosen along with calculated features day_diff and log_r which are the difference between the closing and opening daily price and the daily log return defined as log10(Close Price/ Open Price).

A heat map with these pair-wise correlations is shown for the entire data set and also for a single company (Agilent Technologies Inc.).

It is apparent from the heat map that there are not strong (either negatively or positively) correlations between different feature pairs. The features that are closely correlated are logically related in the first place. For instance, returnsOpenPrevRaw1 and returnsOpenPrevMktres1 are highly correlated which makes sense. The returnsOpenPrevMktres1 is simply the raw value except it has be corrected to take into account a standardization term for the market. Hence, the exact values in these heat maps do not in themselves provide deep insight. However, it is interesting that there seems to be stronger correlations for the individual company (Agilent Technologies Inc.) compared to the correlations with the entire data set. This may suggest that how company features are related is dependent on the type of company and therefore to properly model the system these difference may have to be accounted for.

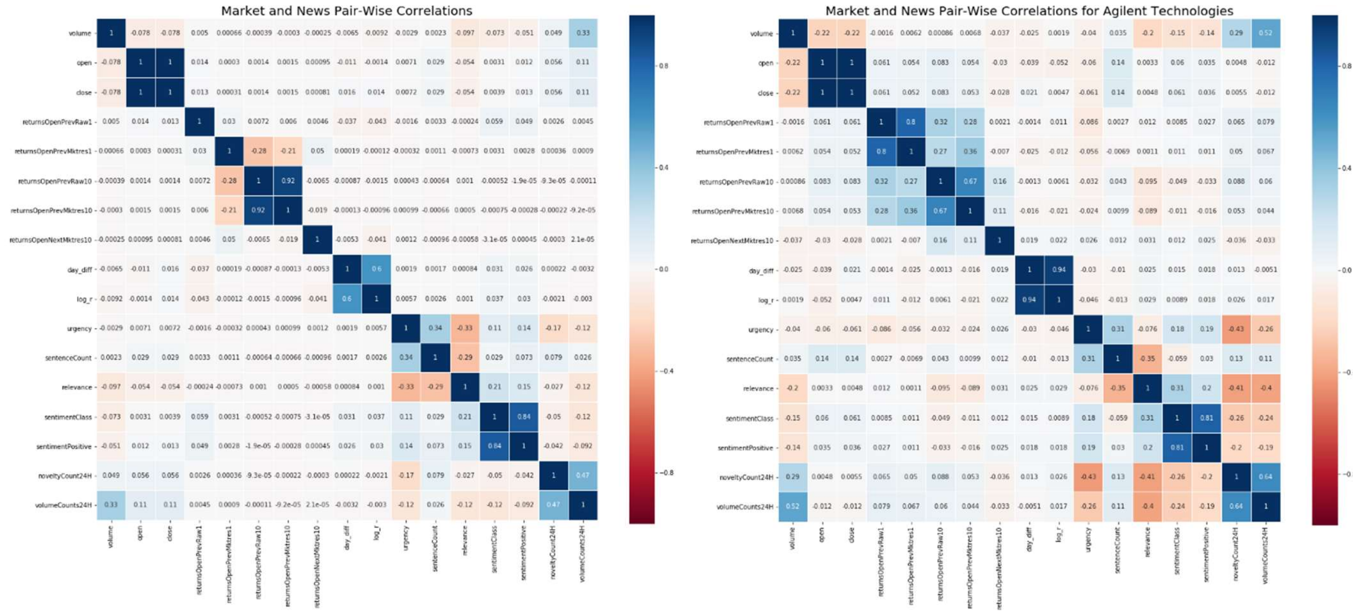


Figure 10 Left : Heat map of pair-wise correlations for entire data set. Right : Heat map of pair-wise correlations for Agilent Technologies Inc.

4. Modeling

The goal of the project is to predict stock market movements from market and news data. There are some features that are useful in the model and others that are disregarded because they do not tend to add additional information. It is worth acknowledging that there were several iterations to arrive at the features chosen and the hyper-parameters used in modeling however all instances are not discussed. Rather, the final choice of these features and parameters are discussed with details of how these final choices were made.

4.1. Modeling Goal and Target

The ultimate goal of this project is to use features from market and news data to predict stock market movement. To do this the desired model outcome is to predict whether the feature r_{ti} =returnOpenNextMktres10 is positive or negative. The Two Sigma challenge on Kaggle asks for the result be presented as a value between -1 and 1. Here a value of -1 means 100% certainty that the target is negative and a value of 1 means 100% certainty the target is positive. Thus, the model outputs a distribution with values $\hat{y}_{ti} \in [-1, 1]$. The scoring metric for this competition is composed of this output distribution and several other values in the market data:

$$x_t = \sum_i \hat{y}_{ti} r_{ti} u_{ti}$$

$$score = \frac{\bar{x}_t}{\sigma(x_t)}$$

where r_{ti} is the 10-day market-adjusted leading return for day t for instrument i and u_{ti} is 0/1 universe feature that controls whether the asset is used in scoring that particular day. Notice, the definition of this scoring metric is essentially a 10-day Sharpe ratio. The larger this value the better the trading strategy.

4.2. Feature Engineering and Word Embedding

There were a few simple features that were create directly from features in the market and news data. See appendix to see all original features and features chosen for modeling. First, the company names were assigned a numerical value to be used to distinguish companies while modeling. Another feature and a metric commonly used in stock analysis is the log return which is defined as log10(Close/Open) per day. Finally, we take rolling averages over 5 and 10 trading days. The features in which averages are taken are returnOpenPrevRaw10 and sentimentClass. The sentimentClass is considered one of the best measures for judging the sentiment of articles and therefore is considered a good choice for creating more features on. The returnOpenPrevRaw10 was seen to be the most significant feature in previous iterations of modeling and therefore creating more features from it seemed logical.

There are multiple features in the news data that are text-based and therefore cannot be directly used in modeling. Namely, the features subjects and headlines have text information. The subjects have a series of subject codes to indicate different topics within the article. The headlines are the explicit headlines for the articles. The word clouds (Figure 12) show the most common subjects and headline words in the last million news articles. To make use of the information content in these features the text is converted to numerical values using a series of analysis steps. First, the information in each feature was vectorized using a CountVectorizer(). This gives the words as word counts. The resulting matrix for the subjects features was X_subject.shape = (4155955, 1552) and for the headlines X_headlines.shape = (4155955, 167317). This means that there are 1552 unique subject codes and 167317 unique headline words. This is too high of dimensionality to model practically. Hence, the second step is to reduce the dimensionality of the vectorized text data. To do this the word matrices were broken down to principle components using TruncatedSVD(). The word matrices were then converted to a compact number of features. A total of 10 components for the subjects and 10 components for the headlines was chosen. The individual components total variance ratio is shown in the plot below and, in the legend, the total variance explained by the 10 components used is shown. Thus, a total of 20 new features is added to the modeling data. To understand what information is captured in news subjects and headlines using just 10 features the variance ratio of the 10 components can be graphed. This ratio is the ratio of the total variance of the word matrices produced for the subject and headlines. The graph shows the individual component variance ratio and the total variance ratio captured by all 10 components. Notice, using 10 components about 33.1% of the variance for the subjects is captured and about 14.4% of the variance for the headlines is captured.

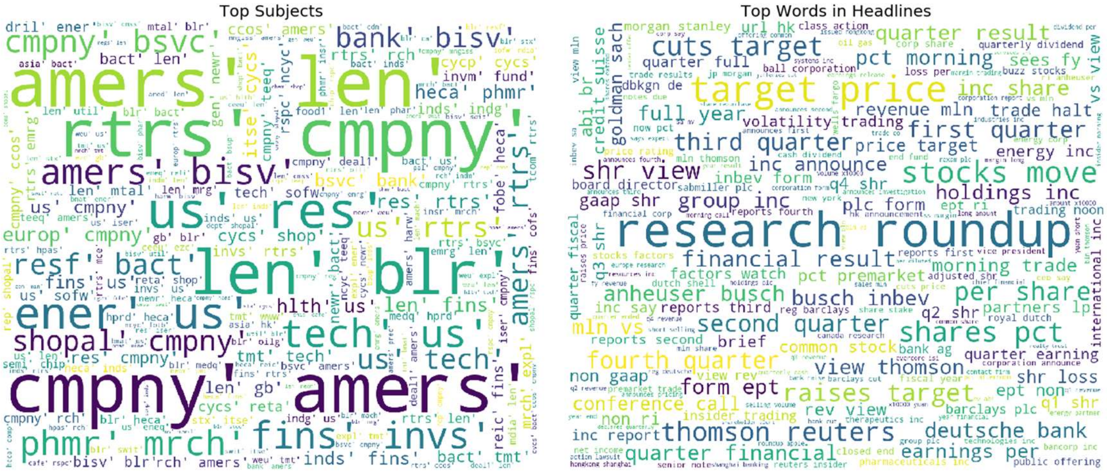


Figure 12 Word clouds for article subjects (left) and article headlines (right).

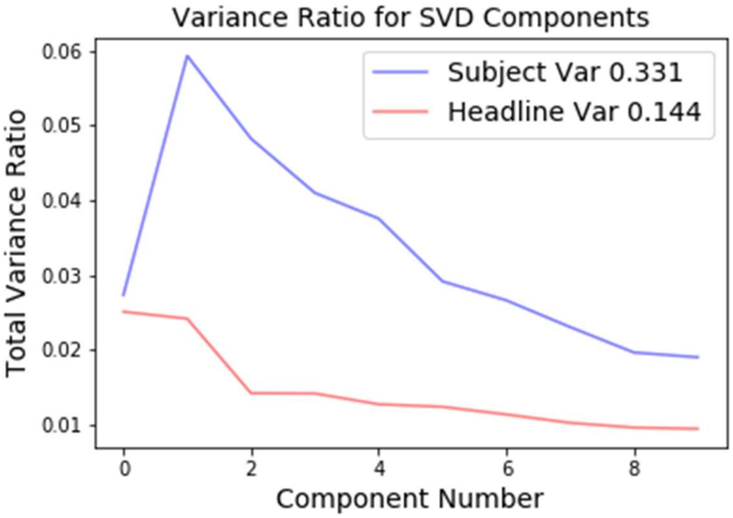


Figure 11 Ratio of total variance for the subject and headline principles components. The legend shows the total variance ratio explained by the 10 components utilized.

Though not part of feature engineering explicitly there was a realization while developing the models. A time segment in the stock market data was highly difficult to model and lowered the accuracy of the models. This was the result of the 2008 recession which causes highly unpredictable behavior. Thus, predicting future stock market movements is better accomplished by eliminating this data from the training set. Data from January 2010 onward was used in the modeling and saw significant improvement in model accuracy compared to including data from 2008 and 2009.

4.3. Machine Learning Algorithms

Several machine learning algorithms were tested and then they were used and ultimately an ensemble approach was chosen. Hyper-parameters for each model were tuned using GridSearchCV() with 5-fold cross validation. There tended to be little (if any) different in the performance when choosing different hyper parameters. The exception was with KNN in which the accuracy score varied with the number of neighbors parameter. Ultimately, the default parameters were used for most other algorithms for the final model because parameter tuning provided little improvement. Here the algorithms that were tested are.

XGBoost : Gradient boosting algorithm that uses decision trees. Made to be efficient and flexible and provides parallel tree boosting.

LightGBM : Gradient boosting framework that uses tree based learning algorithms. Created to support lower memory usage and higher training speed. Sacrifices on some accuracy for speed.

Naïve Bayes : Supervised training algorithm that applies Bayes' theorem with the "naïve" assumption of conditional independence between every pair of features.

KNN : Non-parametric learning algorithm that classifies data into different classes based on the feature space nearest neighbors.

Random Forest : An ensemble learning method for classification that operates by constructing many decision tress and taking the mode of the classes in a classification problem.

The models output a probability of the 10-day market adjust leading return to be positive or negative for each instance in the test set. A probability below 0.5 indicates negative and a probability above 0.5 indicates positive. The probabilities are then converted to the confidence values, $\hat{y}_{ti} \in [-1, 1]$, used in the competition with $\hat{y}_{ti} = 2p_{ti} - 1$. The confidence distribution and ROC curve for each model is shown in the appendix. Two ensemble models are also used in the final analysis. The first ensemble model uses a weight average of the XGBoost and the Random Forest classifiers such that the final probabilities were $p_{ti} = (54\, p_{ti\_XGBoost} + 46 p_{ti\_RF})/100$. The final model selected used XGBoost, KNN and Random Forest algorithms and then performed stacking with Logistic Regression as the meta classifier algorithm. Using LighGBM in the stacking method produced errors in running the code therefore this algorithm was avoided during the stacking ensemble method. The table below shows the final results for the accuracy and scoring metric for each model. Notice, the ensemble stacking method produces the best results. The best result for the scoring metric is 0.9547 which is ranked as 34 out of 693 users on the Kaggle leaderboard which is a Kaggle silver medal. For completeness the area under the ROC curve is included in the table.

Table 1 Accuracy, AUC of the ROC and Two Sigma scoring metric for each model tested.

Model	Accuracy	AUC	Score
XGBoost	0.5867	0.624	0.9241
LightGBM	0.5738	0.606	0.8275
Naïve Bayes	0.5052	0.509	0.0810
KNN	0.5251	0.536	0.4241
Random Forest	0.5635	0.590	0.8597
Weight Average	0.5822	0.619	0.9541
Stacking	0.5892	0.588	0.9547

4.4. Ensemble Stacking Classifier and Feature Importance

The model that produces the best accuracy and scoring metric takes XGBoost, KNN and Random Forest then uses stacking with a Logistic Regression algorithm as the meta classifier. The resulting confidences from the test data are plotted in the histogram below along with a ROC curve for the stacking model. Notice, the auc for the stacking algorithm of 0.589 is actually worse than that of XGBoost alone, 0.624. This is just due to the nature of how the algorithm operates. Since XGBoost also gives a good accuracy and score it is beneficial to look at additional outputs from this algorithm. In particular we can look at the feature importance from the model to see which features played the most important role in training. Notice, that the top two features in terms of importance, r5 and returnsOpenPrevMktres10, are both features produced by or from the market data. The feature r5 is just the 5 trading day rolling average of returnsOpenPrevMktres10. This means that despite the goal of using news data to predict market movements, the market data plays a more significant role in predicting future stock movements. The third most important feature is a 10 trading day rolling average of the sentiment class. This indicates that news data also plays a role in predicting stock movements.

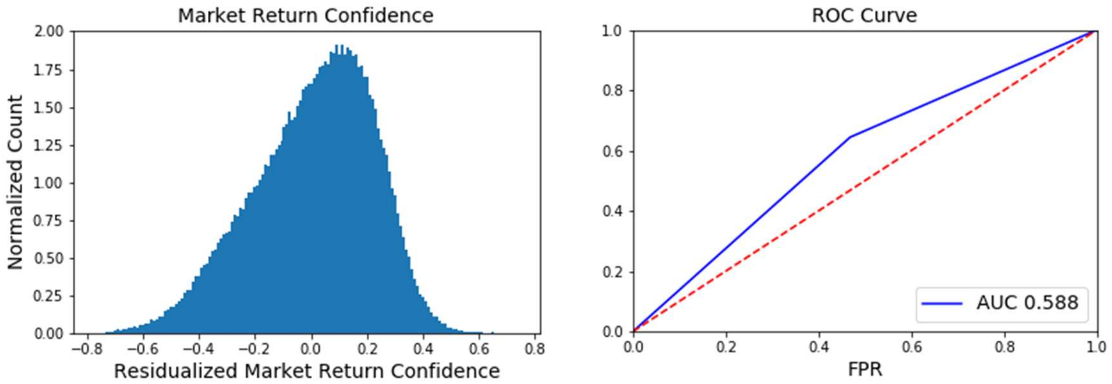


Figure 14 Left : Confidence distribution on the test data using the stacking model. Right : ROC curve for the stacking model.

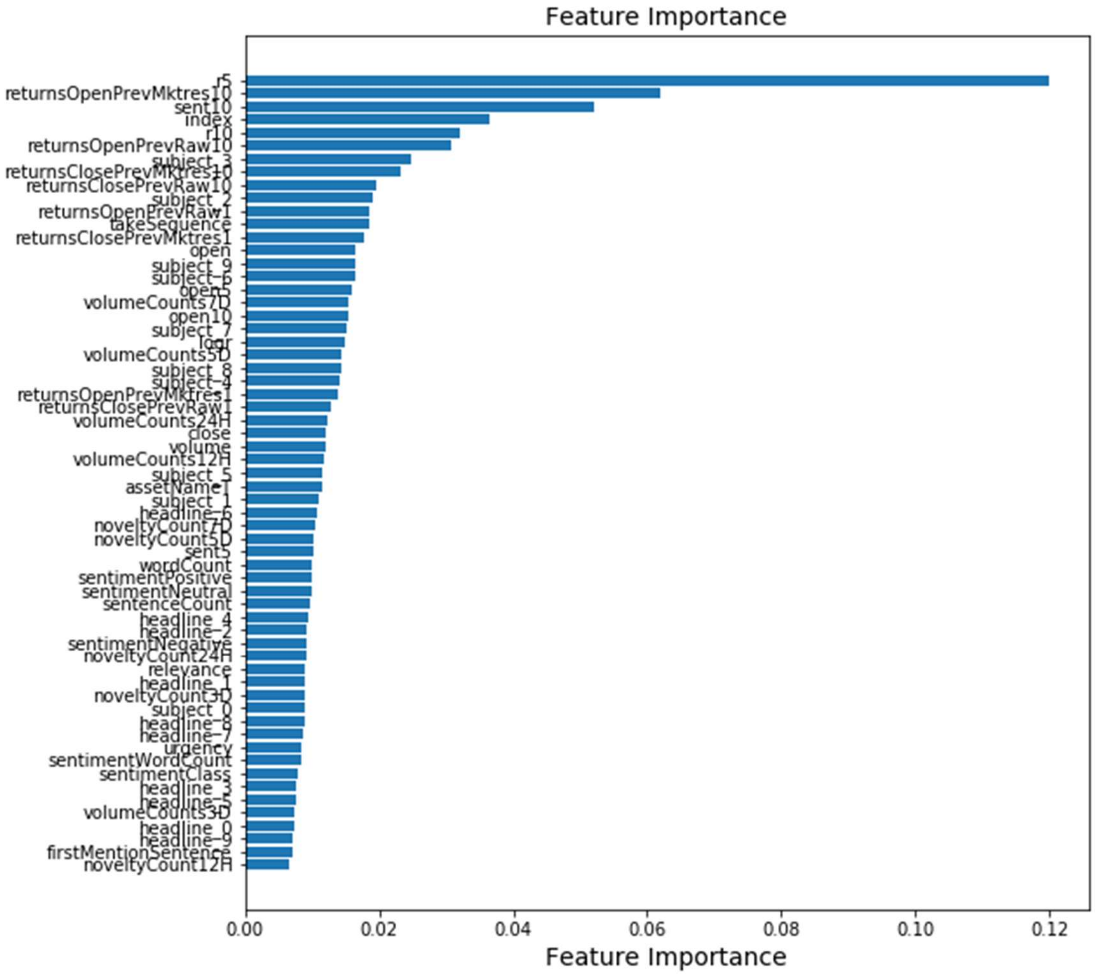


Figure 13 Feature importance after training of the XGBoost classifier model.

4.5. Assessment of Model Strengths and Weaknesses

The best accuracy and scoring metric is produced by the ensemble stacking model with XGBoost, KNN and Random Forest as the classifiers with a logistic regressor for the meta classifier. The next step is to look at the data that this model predicts well and which data it poorly predicts. To do this we can choose data that has a confidence values, $\hat{y}_{ti} \in [-1, 1]$, whose absolute value is large (good predictive ability) and the absolute value that is small (poor predictive ability). The exact value for being large or small is abstract however using trial and error limits were set such that there were at least 500 data points for the good and poor predictive values. Good predictive data points are those in which $|\hat{y}_{ti}| > 0.63$ while poor predictive data points are those in which $|\hat{y}_{ti}| < 0.001$. shows the stock opening price with a red line indicating the best and worst prediction data points. Looking at the graphs there is no clear indicator as to why the data is better predicted for one instance compared to another.

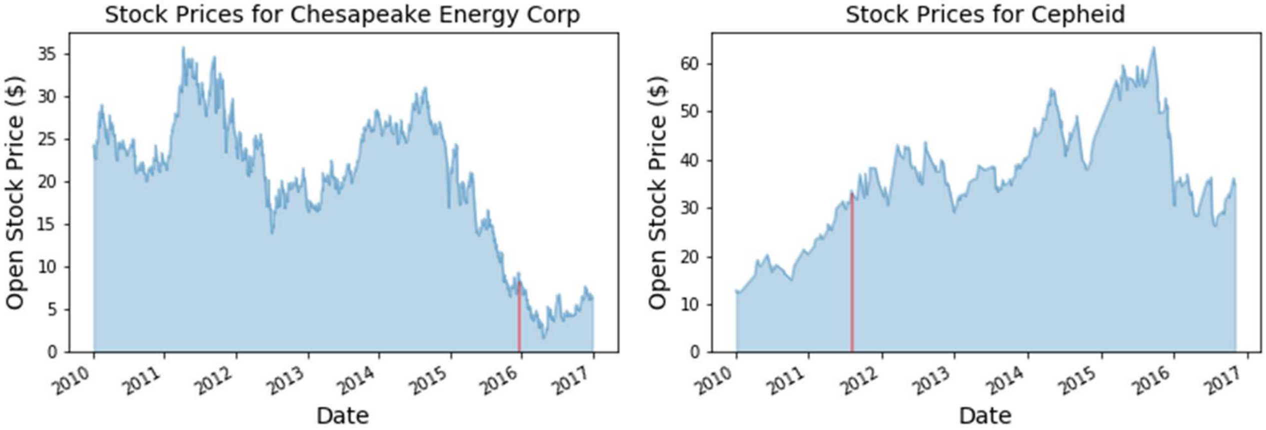


Figure 15 Left : Opening stock price with red line indicating the best prediction data. Right : Opening stock price with red line indicating the worse prediction data.

The next thing we can look at are mean values of the features and target for the good and poor prediction data. Looking at returns and sentiment class in particular are interesting because if the data has no bias then the mean value should be close to zero. However, if the returns or sentiment class are strongly positive or negative then it indicates a bias in the data set. When looking at the poor predictive data points there the 10-day market adjusted return, other return features and the sentiment class are all small absolute values. This indicates that there the data that is predicted the worst is has evenly distributed data for positive and negative returns. This makes sense because in Figure 14 the data about zero is somewhat symmetric. When looking at the data that is predicted well the returns are all significantly negative and the sentiment class is -0.524 which means news articles were largely negative. This is interesting in that it means that the best prediction values are those that have negative returns with negative news sentiment. This is not too surprising because news articles are largely bias towards positive sentiment as seen in the exploratory data analysis. Hence, when an uncommon negative news segment comes out then this probably plays a larger role in predicting the market returns. Once again, Figure 14 the skewed tail on the distribution shows that the 10-day market adjusted leading return is better predicted for small values.

When looking at companies in the good and poor prediction data sets there are a few re-occurrences however there is no single company that has a significant number of appearances. The companies are not organized by industry therefore it is difficult to put them into categories to better understand if there are any trends with industry. However, we can find common words in the good and poor prediction data sets and look for trends. In Table 2 the common words within company names for the good and poor prediction data sets are shown. Notice, common words such as “Inc”, “Corp”, “Co”, etc. are seen in both tables. The interesting outlier in the good prediction data set is the word “Energy.” This is a clue to the types of companies that were easily predicted. Not too much further down the good prediction table the words “Oil” and “Petroleum” appear. This information will come in handy when explaining the best predicted data.

Table 2 Left: Good prediction common words in company names. Right: Poor prediction common words in company names.

word	count	
2	Inc	180
22	Corp	152
59	Energy	83
16	Ltd	76
7	SA	47
37	Co	46
19	Resources	41
26	PLC	40

word	count	
8	Inc	352
1	Corp	187
54	Group	51
20	Co	49
5	Ltd	48
226	Financial	32
98	Holdings	26
64	&	24

Lastly, the time of year for good and poor predictions can be looked at. The raw numbers for the number of data points in each month is shown in the appendix. There is little of note in the poor prediction data points; the data is approximately evenly spread across the months. There is one particular month in the good prediction data that stands out. There is a large amount of data that is predicted well in September, 2015. To understand what is occurring at this point in time the 10-day market adjusted leading return is plotted over time for the months leading to and proceeding September, 2015. This did not give good insight into what is occurring. However, remembering that the company keyword “Energy” and knowing that this was around the same time they oil price crashed, the data can be filtered by looking for company keywords “Energy”, “Oil” and “Petroleum” (Figure 16). Looking at the data around September, 2015 the 10-day leading return shows a significant drop that seems to be a behavior that is easily predicted by the model.

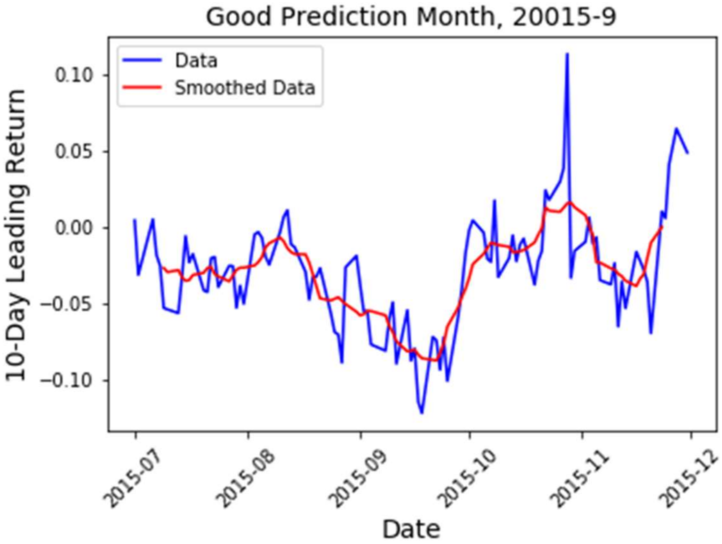


Figure 16 Good prediction month for data using "Energy", “Oil” and “Petroleum” as keywords in company name.

5. Future Improvements

5.1. Company Classification/Grouping

Individual company stock tickers respond differently to the overall stock market trends and news articles. For instance, there are many companies in which the stock ticker price is related to the company health and therefore news related to the company can have a direct impact on the company stock. However, not all companies hold growth stocks and the ticker price is not as directly related to the company. Also, as seen in section 3.5 (Page 8) and section 3.6 (Page 9) cross correlation between features and lag time to see return autocorrelation are company dependent. A future step to improve this project would be to classify companies into groups and then model the groups independently. The classification method could be to develop a series of metrics that can be used to perform an unsupervised classification or more data could be collected about each company that allows direct grouping. For instance, stock companies are classified using Global Industry Classification Standard (GICS) or Industrial Classification Benchmark (ICB). These are industrial-based classifications however there are other lists to classify companies.

5.2. Stock Market Data Anomalies

Generally stock ticker data for individual companies is messy and there are anomalies that regularly occur that are highly unpredictable. One such issue is that of stock splits. This occurs when the number of shares in a company increases and the stock price is adjusted such that the before and after market capitalization of the company remains the same. This was not accounted for in this project but could have an impact algorithm training. The January effect which refers to stocks outperforming the market if they had a poor fourth quarter. There is also a tendency for the market to move more and positively on Friday than Monday. There are a large number of issues when dealing with stock market data that should be more carefully treated in future models.

5.3. Word Embedding

The word embedding methodology done in this project is one of the simplest one could perform. A word matrix is created with counts for the number of words in each subject and headline instance in the data. More advanced algorithms exist to score each word for frequency or word relationships to better interpret the headline meaning. Thus, a more advance algorithm could be used for interpreting the text data. Additionally, the word matrices were reduced in dimensionality by breaking them down into principle components and then 10 components for both the subjects and the headlines were kept as additional modeling features. The variance accounted for by only 10 components may not have been sufficient to fully capture the text information. Hence, simply increases the number of components for modeling features is an additional recommendation.

6. Conclusions

The goal of this project was to predict stock movements using news data. The model developed predicts the confidence for the 10-day market adjusted leading return to be positive or negative. The project results can be summarized:

- No single machine learning method was sufficient to give a good enough final accuracy. An ensemble stacking method using XGBoost, KNN and Random Forest with a Logistic Regressor as the meta classifier gave the best accuracy and scoring metric.
- Using the methods in this project a score of 0.9547 was produced which is sufficient to rank 34 out of 693 in the Two Sigma Kaggle competition.
- In the process of model development it was noted that the most important features for predicting the target were all derived from the market data. The most important features were the 5 day rolling average and the non-averaged 10-day market adjusted previous return. The third most important feature was from the news data and was the 10 day rolling average sentiment class.
- The best algorithm tended to best predict data that had negative returns and sentiment class. Thus, predictions are best for declining stocks with negative news articles.
- There are multiple improvements that could be implemented for better model accuracy. Some of these improvements would require gather of more data (company classification) and others simply require more computation effort and time (word embedding).

7. Appendix

7.1. Features

Market Features

```
['Unnamed: 0', 'time', 'assetCode', 'assetName', 'volume', 'close', 'open',
'returnsClosePrevRaw1', 'returnsOpenPrevRaw1', 'returnsClosePrevMktres1',
'returnsOpenPrevMktres1', 'returnsClosePrevRaw10', 'returnsOpenPrevRaw10',
'returnsClosePrevMktres10', 'returnsOpenPrevMktres10',
'returnsOpenNextMktres10', 'universe']
```

News Features

```
['Unnamed: 0', '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']
```

Modeling Features

```
['close', 'firstMentionSentence', 'headline_0', 'headline_1', 'headline_2',
'headline_3', 'headline_4', 'headline_5', 'headline_6', 'headline_7',
'headline_8', 'headline_9', 'index', 'noveltyCount12H', 'noveltyCount24H',
'noveltyCount3D', 'noveltyCount5D', 'noveltyCount7D', 'open', 'relevance',
'returnsClosePrevMktres1', 'returnsClosePrevMktres10',
'returnsClosePrevRaw1', 'returnsClosePrevRaw10', 'returnsOpenPrevMktres1',
'returnsOpenPrevMktres10', 'returnsOpenPrevRaw1', 'returnsOpenPrevRaw10',
'sentenceCount', 'sentimentClass', 'sentimentNegative', 'sentimentNeutral',
'sentimentPositive', 'sentimentWordCount', 'subject_0', 'subject_1',
'subject_2', 'subject_3', 'subject_4', 'subject_5', 'subject_6', 'subject_7',
'subject_8', 'subject_9', 'takeSequence', 'urgency', 'volume',
'volumeCounts12H', 'volumeCounts24H', 'volumeCounts3D', 'volumeCounts5D',
'volumeCounts7D', 'wordCount', 'assetNameT', 'open5', 'open10', 'r5', 'r10',
'logr', 'sent5', 'sent10']
```

7.2. Machine Learning Models Graphs

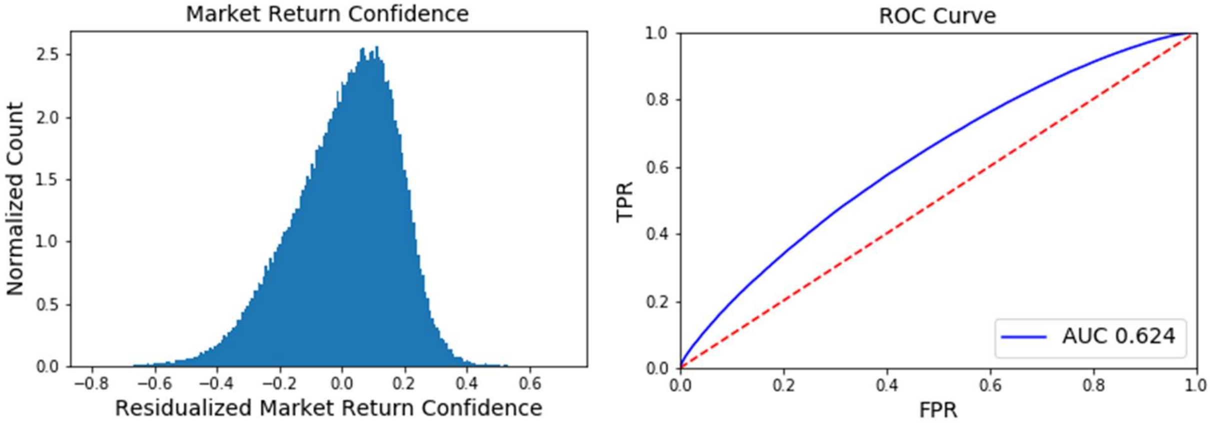


Figure 17 XGBoost Classifier Left: Confidence distribution for 10-day leading return. RIght: ROC curve for algorithm.

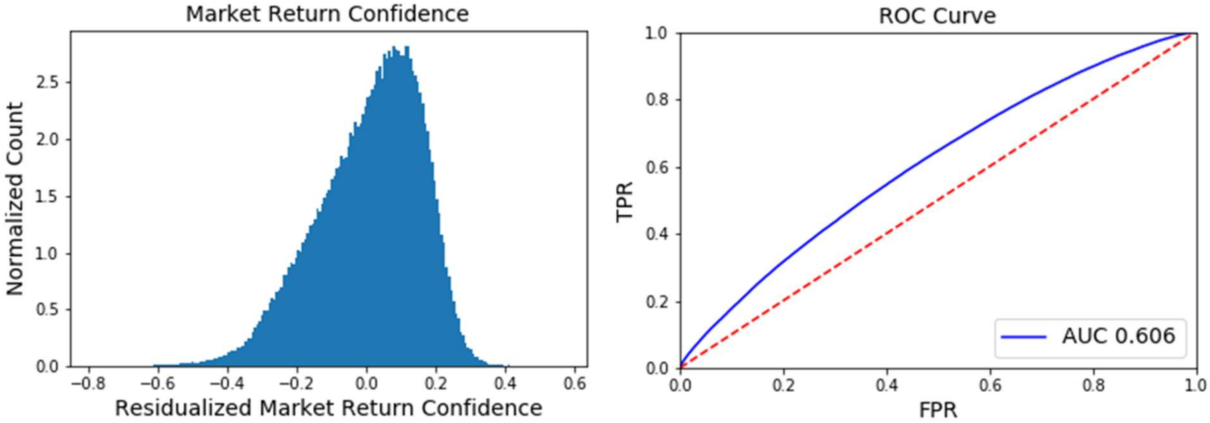


Figure 18 Light GBM Left: Confidence distribution for 10-day leading return. RIght: ROC curve for algorithm.

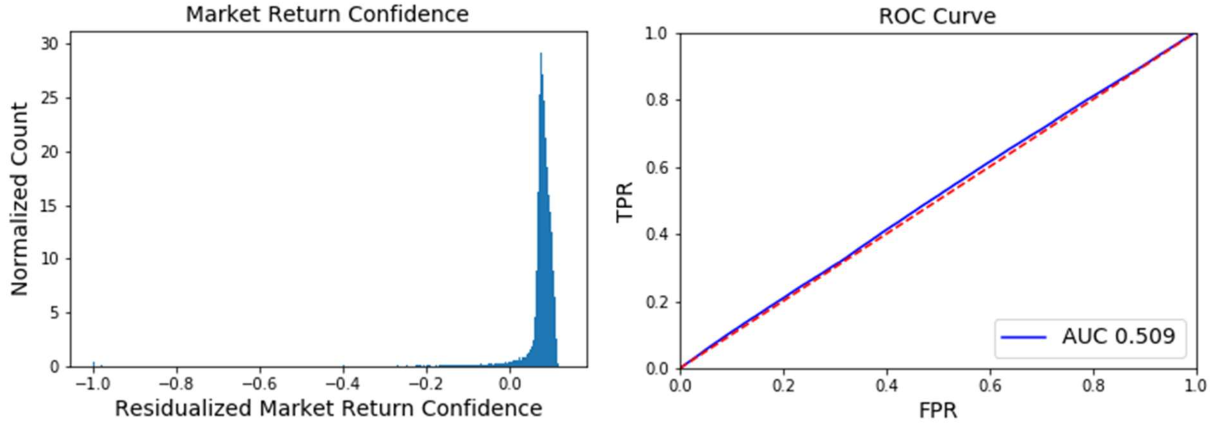


Figure 19 Naïve Bayes: Confidence distribution for 10-day leading return. RIght: ROC curve for algorithm.

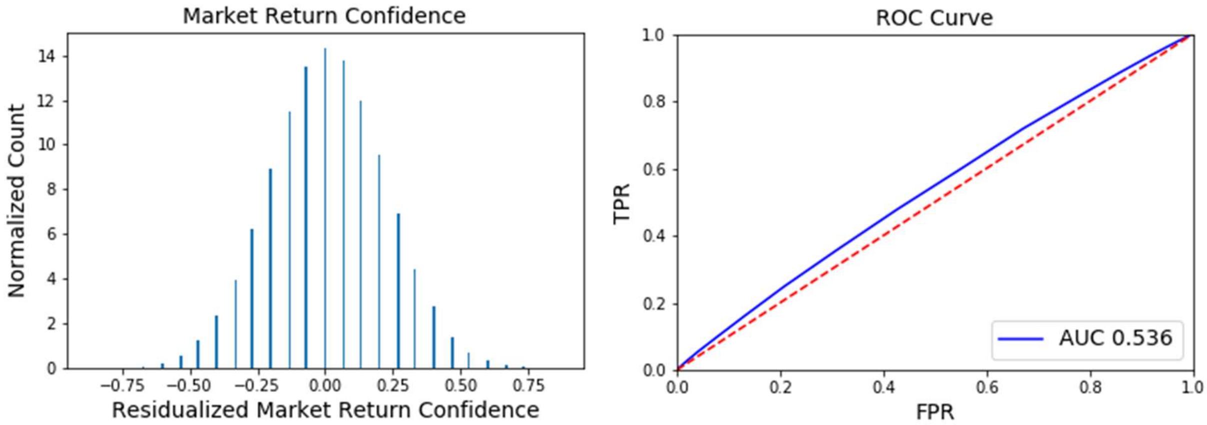


Figure 20 KNN Left: Confidence distribution for 10-day leading return. RIght: ROC curve for algorithm.

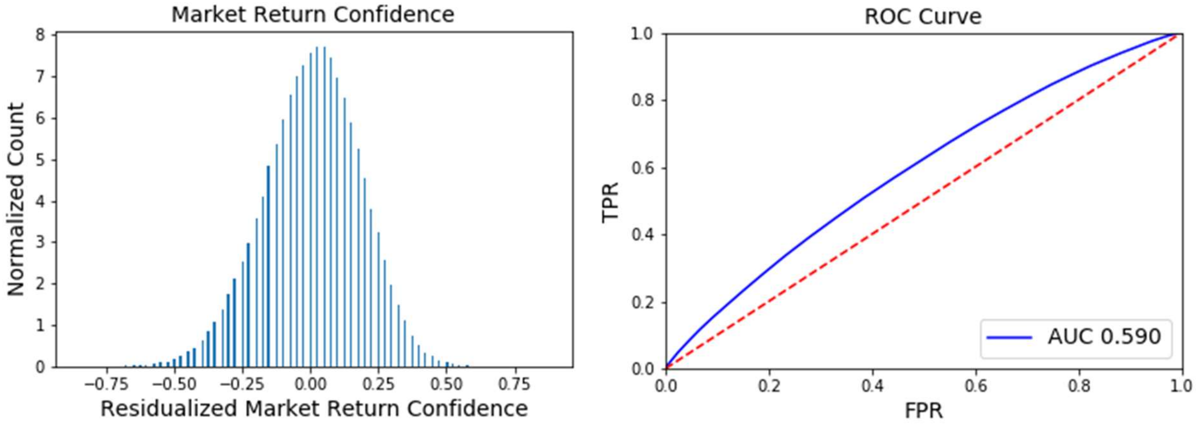


Figure 21 Random Forest Left: Confidence distribution for 10-day leading return. RIght: ROC curve for algorithm.

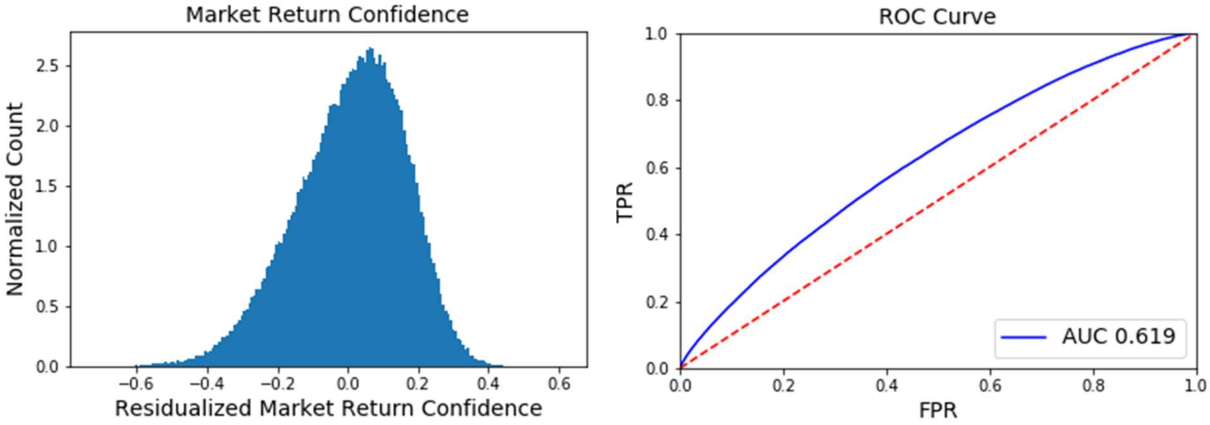
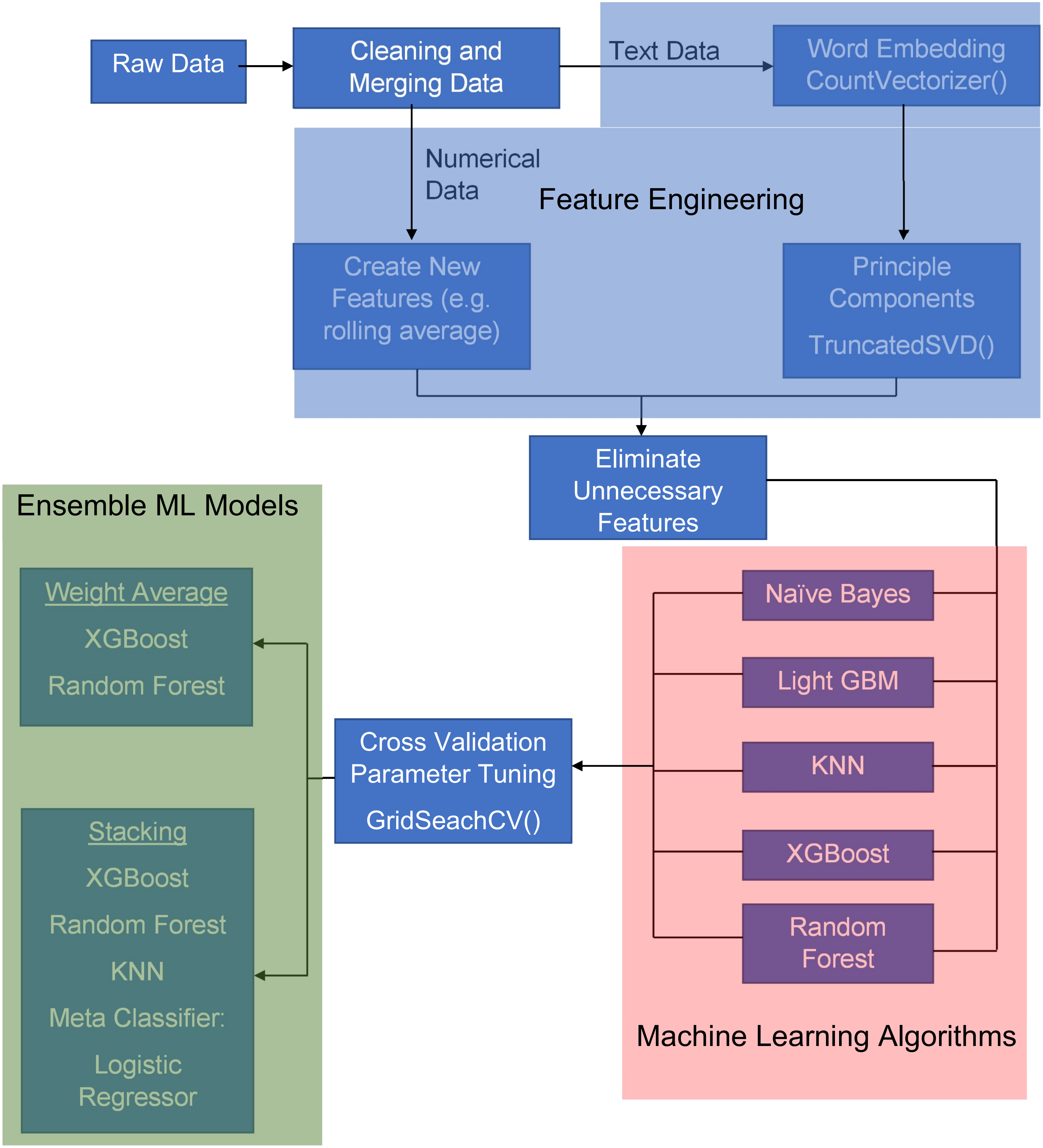


Figure 21 Weighted Average XGBoost and Random Forest Left: Confidence distribution for 10-day leading return. RIght: ROC curve for algorithm.

7.3. Project Flowchart



7.4. Raw Data from Strengths and Weaknesses Section

Good Prediction Data Count by Month

```
            num
year month
2010
   1       6.0
   2      14.0
   3       4.0
   4       1.0
   6      45.0
   9       1.0
2011
   8       1.0
  10       1.0
2012
   4      18.0
   5       6.0
   6       2.0
2013
   2       1.0
   4       3.0
   6       1.0
  10       1.0
2014
  10       8.0
  11      18.0
  12      53.0
2015
   1      20.0
   2       9.0
   7       3.0
   8      38.0
   9     165.0
  10      48.0
  11       2.0
  12      25.0
2016
   1      37.0
   2      12.0
   3       3.0
   4       7.0
   5      11.0
   6       2.0
  11       1.0
```

Poor Prediction Data Count by Month

```
            Num
year month
2010                2014
   1       1.0         7      14.0
   2       1.0         8      17.0
   3       9.0         9       8.0
   4      11.0        10      12.0
   5       8.0        11       9.0
   6       4.0        12       7.0
   7      10.0      2015
   8       7.0         1       6.0
  10       3.0         2       6.0
  11       7.0         3      11.0
  12       2.0         4      11.0
2011                   5       6.0
   1      13.0         6      14.0
   2      12.0         7      13.0
   3       8.0         8      10.0
   4       5.0         9      12.0
   5      10.0        10       9.0
   6       9.0        11       4.0
   7      10.0        12       7.0
   8      12.0      2016
   9       5.0         1      11.0
  10      10.0         2       9.0
  11      10.0         3       5.0
  12       7.0         4       7.0
2012                   5      13.0
   1       6.0         6       5.0
   2      12.0         7      18.0
   3       4.0         8       6.0
   4       6.0         9      17.0
   5       5.0        10       7.0
   6      11.0        11      16.0
   7       8.0        12      13.0
...        ...
```