Market Prediction Based on Daily News from The New York Times

Capstone Project 2

August 8, 2019

David Gray

Goal of Project

The goal of this project is to train a machine learning model to predict the movement of the stock market based on the news for that day or week. To motivate this project, a parallel might be drawn between the news that is presented to the public (what the newspaper anticipates the public wants to learn about) and the volume of specific search terms (what the public wants to learn about).

Justification for Approach

It has been stated that the search frequency of terms is correlated with stock movement. This is from the website, Seeking Alpha:

- “Past research suggests that the relative change in the volume of Google searches for financial terms such as “debt” or “stocks” can be used to anticipate stock market trends.
- “In this analysis, the search term “debt” was used to obtain monthly search volume data from Google Trends.
- “The analysis shows that a decrease in search volume typically preceded price increases in the S&P 500 index, and vice versa.

- “Switching between ETF (SPY) and ETF (IEF) based on monthly search volume data from 2005 to 2018 would have made a profit of 634% versus 220% for buy-and-hold SPY.”

This is some motivation to try to predict the stock market with information deemed important. Reasons to study this question of connecting the news with stock market movement is both to learn the tools of NLP and machine learning model generation, and also to see whether financial gain could be made with a model generated such as in this work.

Data Sources and Model Prediction

The news collected for training and testing consisted of ten abstracts from national news articles and ten abstracts of international news for each day. These abstracts were joined for analysis first from each day and subsequently for five-day periods. The beginning to end dates were July 20, 2009 to July 19, 2019. Only those news articles were collected on days where the market was open, and the stock market news was accessed from Yahoo Finance.

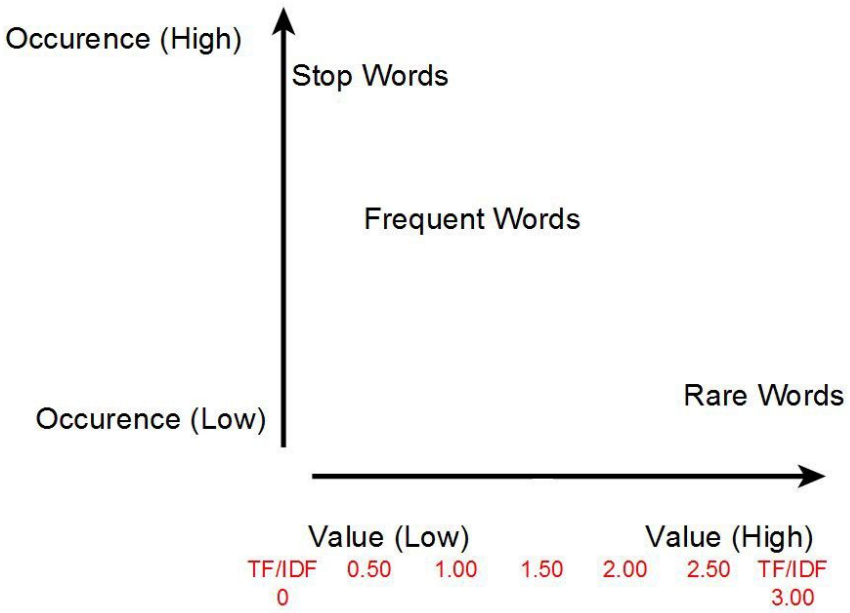
In summary, the approach of this project is to gather daily news from ten years of New York Times’ abstracts and connect that with the movement of the S&P 500 (binary up or down movement) in order to predict the stock market on a daily basis using natural

language processing (NLP). Importantly, the model predicted movement for the day - whether it increased or decreased - and not the magnitude of change.

This data set was collected directly from The New York Times using an API that the news organization provides. There are access limits for its use, so a delay was set between every API call of twenty-three seconds to abide by the daily limits.

NLP Techniques

Standard NLP techniques are employed, which include tokenizing the collection of twenty total articles, removing the stop words, and lemmatizing the remaining words. The resulting data set consists of approximately 40,000 columns (unique words) and 2,500 rows (days of news). When bigrams were constructed in addition, there were over 500,000 columns. A pandas dataframe is used to store the information of how many counts per word exist for each day of news. So far, both count vectorization and TF-TDF approaches were used. The count vectorization approach has resulted in a slightly higher accuracy than TF-TDF. Here is a graph summary of the basis for TF-IDF:



In this plot, we see that high value words are rare, while low value words occur frequently. Words that occur so frequently that they are not considered valuable are removed. These are the “stop words.” From: Towards Data Science

Results from Daily News with Random Forest

In sum, three machine models were tested and selected based on their utility in NLP. The results are at best comparable to always guessing that the stock market increased. This result is perhaps not surprising given the number of factors moving the market. Other reports online of models created for this purpose report similar results. With the relatively simple approach of counting words (rather than TD-IDF), a random forest model had an accuracy of 51.35%. The percentage of days that the stock market increased was 54.53%. By implementing TD-IDF, the accuracy of the model decreased to 49.09%.

Additional Steps

After the initial testing of a random forest model, the following was attempted:

- Use bigrams in tokenizing words
- Train a machine learning model using approach other than Random Forest, including Naive Bayes and Support Vector Machine (SVM)
- Collect the daily news into groups of five (five workdays per “week”) and associate that with the related market movement to train based on a longer time scale.

Results from Additional Steps

Weekly Change:

Machine Learning Model	Naive Bayes	Random Forest	SVM
Accuracy	60.84%	60.43%	60.84%

The percent of five-day periods where the market increased was 60.83%, so Naive Bayes and SVM performed just a fraction better than guessing. Here are the results of grid search with SVM:

Mean	**STD**	**Params**
0.51690	0.05330	{'kernel': 'linear', 'C': 1.0}
0.60835	0.00235	{'kernel': 'rbf', 'C': 1.0}

0.51690	0.05330	{'kernel': 'linear', 'C': 10.0}
0.59642	0.01011	{'kernel': 'rbf', 'C': 10.0}

With bigrams, the random forest model fell slightly to 59.65%. The run-time for processing was very long, preventing analysis with Naive Bayes and SVM.

Earnings

One thing to note is that although the accuracy could be low for predicting whether the stock market increases or decreases on a given day, money could still be made if it predicted increases or decreases when there are higher magnitude changes to make up for any losses. That is, buying stocks at the start of the day would yield higher gains since it rises higher.

Conclusion

While the model did not predict with relatively high accuracy the movement of the stock market, there were lessons learned in the process of preparing the data and developing the model. There are at least several aspects to consider in potential next steps.

- Remove counts of numeric values, such as “2,000” besides years. This could potentially be creating noise in the model.
- Consider processing a different section of the news for analysis, such as business and the editorial page if possible.

- To repeat with national or international news, gather headlines instead of random stories for processing.
- Prepare all of the code necessary for analysis of the data after it has been downloaded and processed. This is to prevent any delays of trying to load the data at a later time.
- Perform grid search on all the models to find the best parameters.
- Perform grid search with a great range and number of parameters