Predicting Direction of Stock Movement with NLP and Sentiment Analysis

Liangliang (Will) Chen, Wenyu Chen, Yukun (Marco) Xu, and Lu Yang

1. BACKGROUND

According to the strong form of Efficient Market Hypothesis, all market information both public and private is fully reflected in prices. Given this hypothesis, no investors can generate abnormal return, which is known as zero "alpha" in the stock market. However, due to asymmetric information and liquidity issue, the stock market is not consistently efficient which leads us to exploit information in such a "big data" environment to "predict" the future stock movement. With the accelerating development of technology, there are more data sources and more efficient models to be utilized in the financial market. After years' of efforts on finding alphas using market fundamentals and traditional models such CAPM and Fama-French, alpha has gradually becoming less significant. Therefore, people attempt to use Machine Learning techniques with novel information such as climatic news, twitter tweets to obtain more accurate information of future financial market. Among all the accessible data, news are the most publicly available and can be extracted efficiently. By means of Natural Language Processing (NLP), we can extract the sentiments embedded in the news, analyze their impacts on the financial market and thus predict the future price movements.

2. OBJECTIVE

In this project we would like to employ the method of NLP with news sentiment analysis to automate a stock trading strategy based on the prediction of the stock price movements. The project will be implemented in three steps. The first step involves collecting news data from Google News website using web scraping, and the historical prices of stocks in S&P 500 from Yahoo! Finance. The second step is to clean the data we collected and encode the text information into some numerical data (sentiment scores) using NLP algorithms. In this step, we also attempt to implement feature engineering to construct significant features and prepare dataset for our prediction models. The third step is to define our label we are trying to classify, split the features we constructed into training, validation and testing datasets, and finally feed them into machine learning models. With the whole process, we use a feedback mechanism to update our feature engineering methods and model selection continually.

3. DATA SOURCE

We have mainly collected our data from two sources. The following will explain their sources, types, meanings in details.

3.1. Market Data: the historical stock prices from Yahoo! Finance

- Date type: float
- Date range: trading days between 2015/01/01 - 2017/09/30
- Adj Close: the adjusted closing price of a stock which is a constituent of S&P 500

3.2. News Data: the top 10 news headlines about a particular stock from Google News website

- Date type: text
- Date range: consistent with the trading days in the market data
- Headlines: the top 10 news headlines extracted using web scraping

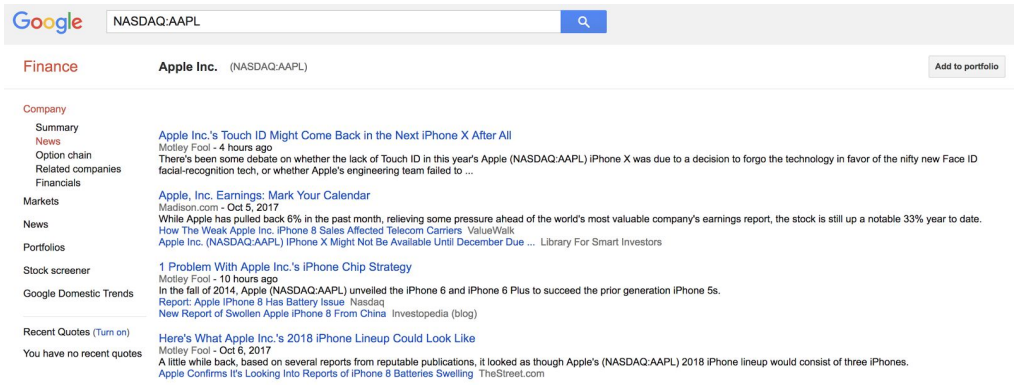
4. MODELING PROCESS

The whole modeling process contains several parts, and we will introduce them separately.

4.1. Data Collection

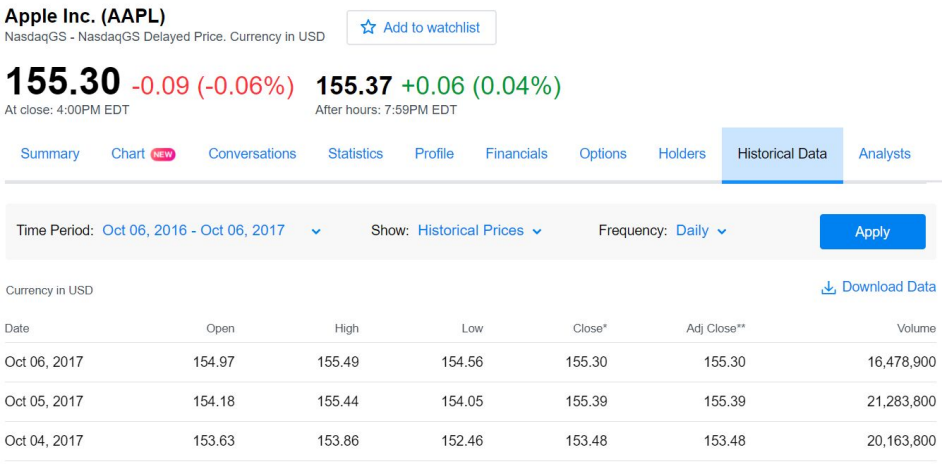
A. We use web scraping with 'urllib' and 'beautifulsoup' to collect historical news headlines from Google News. For each stock, we scrape the top 10 news headlines and store them in a dataframe ascending by the news date.
B. We collect the market data of stock prices from Yahoo! Finance. Considering the effect of stock dividend split, we choose adjusted closing prices to calculate the stock daily returns, and use those to build up labels indicating the stock price's movement . We set the label +1 if the stock's daily return is positive, and 0 if it is negative.
C. We merge the news headlines and the labels into a dataframe and export it to a csv file. To store all the dataframes, we create a folder containing all the exported csv files. The below screenshots illustrate what the news data and market data look at.

News Data:



Source: Google News

Market Data:



Apple Inc. (AAPL)
NasdaqGS - NasdaqGS Delayed Price. Currency in USD

Add to watchlist

155.30 -0.09 (-0.06%) At close: 4:00PM EDT

155.37 +0.06 (0.04%) After hours: 7:59PM EDT

Summary | Chart NEW | Conversations | Statistics | Profile | Financials | Options | Holders | Historical Data | Analysts

Time Period: Oct 06, 2016 - Oct 06, 2017 | Show: Historical Prices | Frequency: Daily | Apply

Currency in USD | Download Data

Date	Open	High	Low	Close*	Adj Close**	Volume
Oct 06, 2017	154.97	155.49	154.56	155.30	155.30	16,478,900
Oct 05, 2017	154.18	155.44	154.05	155.39	155.39	21,283,800
Oct 04, 2017	153.63	153.86	152.46	153.48	153.48	20,163,800

Source: Yahoo! Finance

4.2. Data Cleaning

A. We use the regular expression to find the garbage characters in the news headlines and replace them with space character to remove their effect.
B. We use the 'nltk' package to download the stop words set in English and exclude these words, numbers and punctuations from the news headlines to improve our data's purity.
C. We also use the word tokenizer to mark the part of speech for each word in the news headlines and only kept all the nouns, verbs and adjectives, that is, words with sentimental information that could have predictive ability on the stock market.

4.3. Feature Engineering

A. Word Count Vectorization
    a. We use CountVectorizer from 'Scikit-Learn' to realize the vectorization of word frequency in a headline.
    b. We also test TfidfTransformer from 'Scikit-Learn' to take into account of the product of word frequency and inverse document-frequency.
B. Positive & Negative Word Count Analysis
    We use positive and negative text files to retrieve the positive and negative sentiments from the news headlines. By doing this, we added two features 'positive' and 'negative', which count the number of words of corresponding sentiments in each news headline. The table below demonstrates an example of the cleaned dataset.

Date	Headlines	positive	negative	Label
2017-08-11	alphabet inc googl stock premium foia fired go...	1	2	1
2015-11-17	youtube introduces new music app video catalog...	2	5	-1
2016-10-21	googles parent alphabet loses top drone delive...	5	5	1
2017-07-28	analysts predictions alphabet inc googl jm smu...	5	4	1
2017-09-27	special dividends alphabet edition alphabet sa...	4	1	1

C. Encoding features
We encode the stock tickers and represent them in the form of vectors because we expect the stock ticker provides a constraint on the consistence of the news headlines corresponding to a particular stock.

D. Word2Vec method
We have also employed the Word2Vec to transform each news headline into a numerical vector, and calculate the corresponding word vectors' similarities (between 0 and 1). Therefore, we could use a matrix containing the similarity information among those news headlines as our features. The example shown below prints out the similarity between two news headlines.

```
similarity: 0.913756546798
alphabet inc googl stock premium foia fired google employees labor complaint alphabet alleges
week key earning takeaways alphabet inc nasdaq googl tesla inc nasdaq tsla snap inc snap stoc
k cant hide earnings sign free facebook inc fb stock profits buy microsoft corporation msft s
tock hybrid strength deepminds ai struggling beat starcraft ii exclusive tesla developing sel
fdriving tech semitruck wants test pentagons silicon valley unit helps target terrorists jaso
n calacanis shares thoughts disneys streaming plans

sell google stock buy facebook stock intel stock todays technical bullish stock following mee
ting industry alphabet stocks flying high go contrarian alphabet inc alphabet inc goog averag
e ebitda share growth rate percent alphabet presents citi global technology conference transc
ript diehard bargain hunter american homes rent amh inc googl stock trend analysis alphabet i
nc goog amazoncom inc amzn partner microsoft corporation msft needed google says evidence rus
sian ads election china startup races tesla driverless trucks arizona
```

For better training and prediction, we have also conducted two information engineering on the label definition.

A. To enable the model to find the words that have significant predictive power on stock price movement, we have modified the label definition such that only if the daily return is higher than threshold or lower than the negative threshold, we would label them as 1 and -1. We consider the other daily return as neutralized data points and label them 0. The threshold is a hyper-parameter which we can tune to balance the bias and robustness of our prediction. In this way, we extend the label set from {0, 1} to {-1, 0, 1} to account for neutralized words. Below is a comparison between before and after re-defining labels.

Before the label category engineering:

Date	Label	0
9/20/2016	0	Apple Inc (AAPL) Shares Cross 2% Yield Mark
9/21/2016	0	The Apple Inc. A10 Chip Miracle
9/22/2016	1	Apple Inc. (NASDAQ:AAPL) and McLaren Most Powerful Brand Ever
9/23/2016	0	Your First Look at the Apple Inc. A10X Processor
9/26/2016	1	If Apple built a car, here are the companies it would probably work with
9/27/2016	1	Apple Inc.'s Jet Black iPhone Is Ridiculously Hard to Make
9/28/2016	1	What Is This Apple Inc. Mystery Device (AAPL)
9/29/2016	0	Apple, Inc. Sold Over $2.5 Billion Worth of Apple Watches in 2015
9/30/2016	1	Here's What Apple Is Doing at its First Research Center in China

After the label category engineering:

Date	Label	0
2016/9/20	-1	Apple Inc (AAPL) Shares Cross 2% Yield Mark
2016/9/21	0	The Apple Inc. A10 Chip Miracle
2016/9/22	-1	Apple Inc. (NASDAQ:AAPL) and McLaren Most Powerful Brand Ever
2016/9/23	1	Your First Look at the Apple Inc. A10X Processor
2016/9/26	0	If Apple built a car, here are the companies it would probably work with
2016/9/27	0	Apple Inc.'s Jet Black iPhone Is Ridiculously Hard to Make
2016/9/28	-1	What Is This Apple Inc. Mystery Device? (AAPL)
2016/9/29	0	Apple, Inc. Sold Over $2.5 Billion Worth of Apple Watches in 2015
2016/9/30	0	Here's What Apple Is Doing at its First Research Center in China

B. In order to fully exploit the information embedded in the headlines. We consider that there are many other factors which would affect the stock movement, and we hope to predict the idiosyncratic return which is particularly driven by the news. So we subtract the market return from stock returns to obtain the excess returns, thus focusing on the alpha of each stock and ignoring the effect of market beta.

According to the unique characteristic of stocks, we also consider two different modeling method

A. The model is trained with whole stock universe and predict any stock with strong robustness
B. The model is specifically trained with particular stock, and predict this stock with strong precision

4.4. Model Selection

- Logistic Regression
- Random Forest Classifier
- Gradient Boosting Tree Classifier

4.5. Parameter Tuning

- Number of estimators
- Percentage of test set
- Max tree depth
- Minimum number of samples to be at a leaf node
- Max number of features to consider for node splitting

5. RESULTS

5.1. Evaluation Metrics

Since our project focuses on equity price prediction, we use F1 score, accuracy score, and ROC curve to evaluate the performance of our machine learning model.

- Precision: The fraction of the documents retrieved that are relevant to the user's information need.
- Recall: The fraction of the documents that are relevant to the query that are successfully

retrieved.

- F1 Score: The harmonic mean of precision and recall.

$$\text{F1 Score} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$$

- Accuracy Score: The size of the intersection divided by the size of the union of two label sets, is used to compare set of predicted labels for a sample to the corresponding set of labels in true target.

$$\text{Accuracy Score} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}}$$

- ROC curve: plotting the true positive rate (TPR) against the false positive rate (FPR) at various settings. It is the sensitivity as a function of fall-out.

5.2. Accuracy Table for Benchmark Model

The benchmark model we use is the logistic regression from "sklearn.linear_model" module. The prediction accuracy table is as follows:

	F1 score	Accuracy score
Benchmark model	0.384	0.379

5.3. Accuracy Table for Naive Model

Since our model is for binary classification, we used random coin flip as the naive model. The percentage of 1's and 0's in our training dataset is as in the following table:

	1	0
Naive model	52.45%	47.55%

5.4. Accuracy Table for Logistic Regression

	F1 score	Accuracy score
Single Stock	52.45%	47.55%
Multi-stock Portfolio	53.72%	53.77%

5.5. Accuracy Table for Random Forest

	F1 score	Accuracy score
Single Stock	48.88%	54.84%
Multi-stock Portfolio	51.30%	54.70%

5. 6. ROC Curve for parameter and model settings

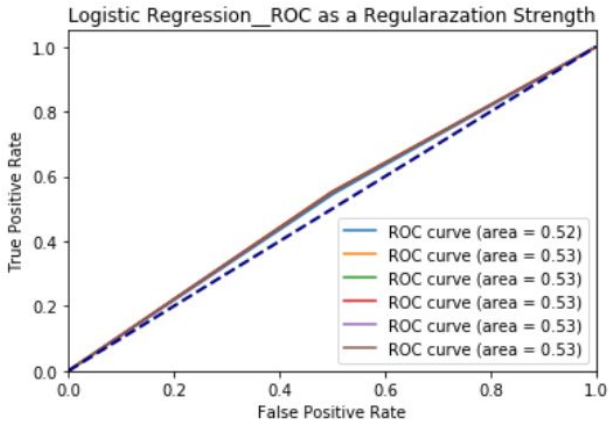


Figure 1: Parameter tuning w.r.t regularization strength: logistic regression on multi-stock portfolio

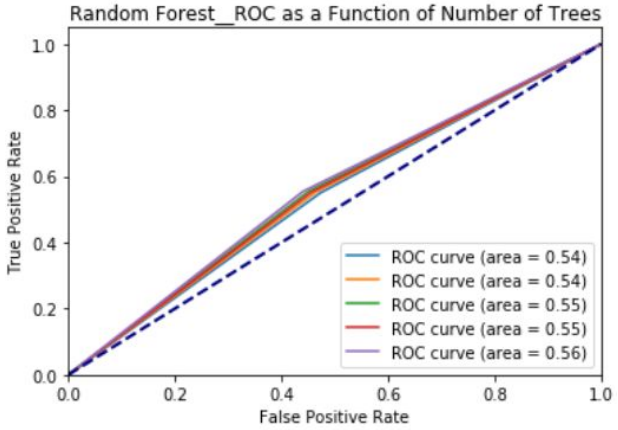


Figure 2: Parameter tuning w.r.t number of trees: random forest on multi-stock portfolio

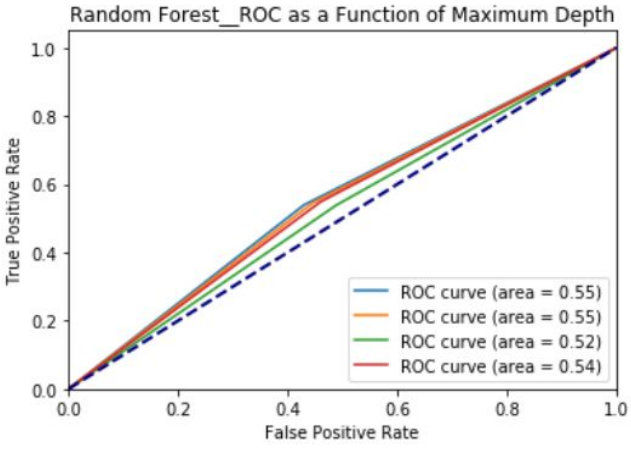


Figure 3: Parameter tuning w.r.t maximum tree depth: random forest on multi-stock portfolio

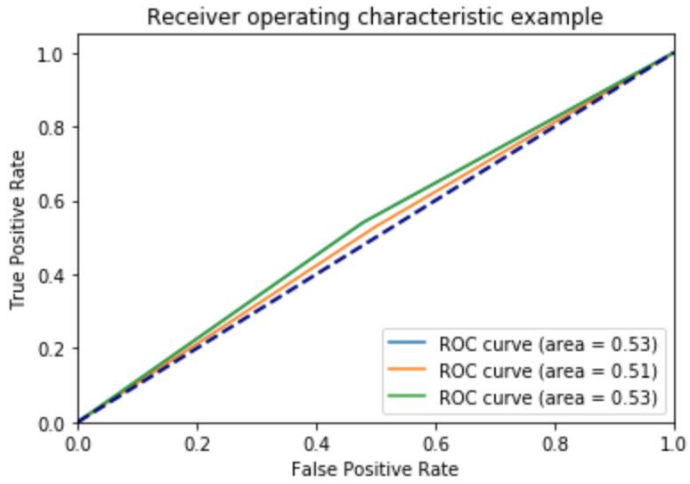


Figure 4: Parameter tuning w.r.t max_features: 'auto', 'log2', 'sqrt' in random forest on multi-stock portfolio

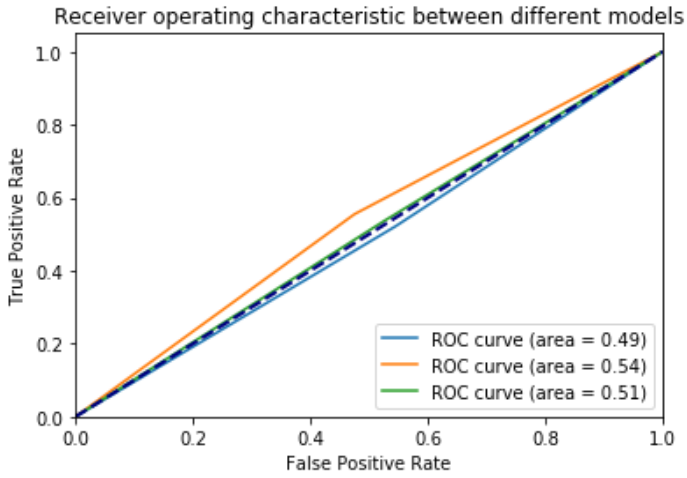


Figure 5: Model Selection on multi-stock portfolio: Random Forest, Gradient Boosting Tree, Logistic Regression

6. CONCLUSION

For this project, we have tested different machine learning models for using news sentiments to predict stock price movement.
With our model, we build a whole pipeline from news scraping to sentiment analysis, and finally provide a trading signal, which people could use to trade or analyse the instantaneous market sentiment.

A. Label engineering: Instead of using the raw stock return to label the training dataset, we used excess stock return with respect to the market to label the dataset for machine learning. This is believed to be closer to the reality as the news headlines related to each individual company tend to affect the alpha (excess return) of that stock. In additional to that, we also realized that the stock return might end up in three states: up, down and

neutral. It's tricky to figure out what range of stock movement should be categorized as neutral movement. We performed a thorough search to figure out the optimum value that would yield the highest accuracy.

B. Feature engineering: As the news contains some information about the future stock returns, we removed the noise, like garbage characters, stop words, numbers and non-indicative words and only kept useful and key information in the dataset in the hope that signals wouldn't be covered by noise. In additional to that, we also structured some additional parameters to capture the negative and positive sentiment as well as characteristics of individual stocks. With all the efforts, we have seen significant improvement in the structure and organization of the features and data labels, which helps pave the way for the next step, model building.

C. Model selection: From the baseline model Logistic Regression, to the advanced model such as Random Forest and Gradient Boosting Tree, we test to see if there is a particular model fit for the sentiment analysis. Considering the bias-variance tradeoff, we check the precision and recall of different models, and try to find out a better algorithm with a more accurate prediction ability on the testing set.

D. Parameter tuning:
In addition to select different models to improve our prediction, we can also choose different parameters in a single model to optimize the prediction results under such a particular model. This idea is referred as tuning parameters, which is usually implemented by cross validating the training dataset.

In the Logistic Regression model, we focused on two possible parameters. The first one is the penalty term known as the norm. In the sklearn.LogisticRegression module, we have an option between 'l-2' and 'l-1' norm. 'l-2' norm is regarded as Euclidean norm and 'l-1' introduces the sparsity in the regularization. We observed that there's no noticeable difference between the 'l-1' and 'l-2' case. Our understanding is in text analysis we have to engage many features in the model and the weights between different features are small. Secondly, the C-value in the module is the inverse of regularization, which means smaller values specify stronger regularization. As seen in Figure 1, there's slight difference between different c-values.

In the Random Forest model, we focused on several parameters in the sklearn.RandomForestClassifier module. The first parameter we attempted to tune is the max_feature ('auto': sqrt(n_features), 'log2': log2(n_features)). As it is shown in Figure 4, we can tell that when max_feature is used by 'log2', the model outperforms the others. The second parameter we studied is n_estimators, which controls the number of trees used in the model. As shown in Figure 2, the performance is improving over the number of trees used. The highest value we used in this simulation is 1500. Take into account that longer time is needed for more trees, we think 500 is the best compromise between performance and time. The third parameter we studied is max_depth, which determines the maximum depth of the trees. We found that with small tree-depth, the prediction of

negative signal is much worse than that of the positive signal and as tree depth increases, the prediction of negative signal improves.

E. Model we recommend for stock price prediction

Random Forest model with the following parameters: n-estimator=500, max_features = 'auto', min_samples_leaf = 10, max_depth=20). Here are the results:

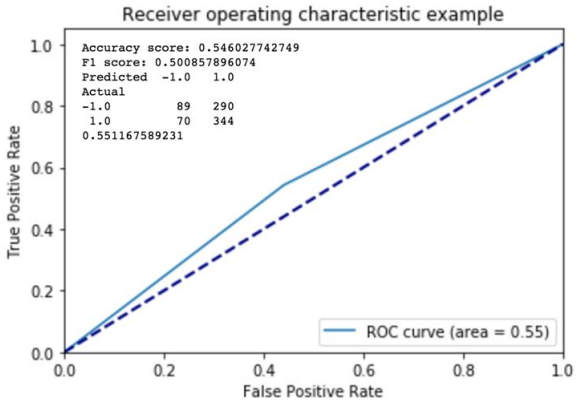


Figure 6: ROC and accuracy scores of our optimized model

7. FUTURE IMPROVEMENT

Given that the best accuracy scores that our model achieved were consistently staying between 52% - 55%, we have considered several improvements for our future analysis.

A. With the news headlines that we extracted from Google News using web scraping, we have been working on several improvements on data processing and feature engineering. However, both of two directions: word frequency and word2vec gave us very similar accuracy results, and after attempts of defining our labels in three ways we did not see a significant improvement in our prediction. Therefore, we divert our attention onto the raw data we extracted in the first place. The news data we collected are mainly from Google News website. As you know, they are continuously updating when you searching the daily news on a particular company, for example, Apple.Inc. The top 10 news of Apple.Inc may not be relevant or sufficient enough for retrieving the sentiments on its stock price's movement. Moreover, as we observed from our data, lots of companies we included in our dataset actually do not have enough news released out every trading day. Even though their stocks' prices are moving, we might not have enough features that can tell us that information. In order to handle such a problem about data collection, we would try to collected news from a variety of sources, announcements from celebrities' blogs, and tweets under a company's twitter official website.

B. From the perspective of Machine Learning techniques, we want to try LSTM, known as Long Short Term Memory Network to create a more complex feature representation of the inputs in the sense that it stores information from arbitrarily long time ago without losing the memory of the sentiments we retrieved from previous news.

C. On the basis of finance theory, the stock return is affected by multiple factors, and we

know the news information is definitely an important factor. Therefore, we hope to build up a multi-factor model, and include the news sentiment score as a new factor. The multi-factor model would separate the stock return into different factors, including size of the company, market beta, revenue conditions on the company, and also the news sentiment. Since the news sentiment factor has low correlation with other traditional financial factors, it would provide people a deeper insight into the influence factors of stock return.

D. Considering the effectiveness and half-life of the news information, we hope to test the model on different time frames. Since there is some news which has a short term effect on the stock movement, for example, a breaking news from Federal Reserve might affect the U.S. dollar in milliseconds, and the market would absorb the news in a few minutes. However, there is also some news which has a longer term effect on the stock movement, such as the news about the change of a company's core business. In this way, we hope to test our model on different time basis, and this might lead to an improvement on our prediction.