A Framework for Financial News Analysis and Stock Movements Prediction

Course project for CSE 6240: Web Search and Text Mining, Spring 2020

Maiqi Ding
Georgia Institute of Technology
mding41@gatech.edu

Xiaoyu Du
Georgia Institute of Technology
xdu93@gatech.edu

Nazanin Tabatabaei
Georgia Institute of Technology
nata3@gatech.edu

ABSTRACT

The ability of predicting trends in the capital markets is extremely important for investment. Before making a trading decision, investors estimate influence that news have on the market and use them as a powerful source of data in financial forecasting. We went through all phases including data collection, data preparation and text mining, then developed a stock index prediction model based on information extracted from news articles, mainly the sentiment analysis of the news headlines, to provide investors with reference of decision-making. The dataset used in this work contains multiple stock market indices from the internet and news articles collected via New York Times archive API. We focused on developing and training different sentiment analysis methods of the news headlines to evaluate the sentiment. News articles are filtered to ensure they are relevant to stock market and analyzed using natural language processing methods. Different Machine Learning models are used to understand the relationship between news sentiment and market change and furthermore predict the price of DJIA and Russell 2000 stock indices. The experimental result show that utilizing PCA for dimensionality reduction improves the prediction accuracy in comparison with other methods, based on RMSE metrics. In addition, combining the Linear method(Elastic net) with the non-linear method(Decision tree regressor), showed reduction in the MAE and RMSE.

KEYWORDS

news, sentiment analysis, machine learning, word vector, stock prediction

1 INTRODUCTION

News can reflect the change on the market and have an impact on financial investor's decisions. Investors analyze recently published news as an important indicator for judging market prices. The reason is that news articles often include information about a company's fundamentals, the associated activities of the company and the expectations of other market participants. A strong relationship between the fluctuation of a stock price and the publication of a related news article has been shown in previous research works [4]. A number of efforts have been made to create a framework that analyses a large amount of textual data relevant to stock market and forecast the stock price.

This research is focused on using non-quantifiable data such as New York Times' news articles headlines and predicting future stock changes depending on the news sentiment, assuming that news articles have impact on stock market. We are using NY Times Archive API to gather the news website articles data over the span of 20 years. Sentiment analysis of the news headlines is then used for training various Machine Learning models to understand their effect and predict the price of Dow Jones Industrial Average (DJIA) and Russel stock indices, collected from Yahoo finance website.

We have used Russell and DJIA stock indices to predict the overall change in US top companies' stock market, instead of predicting individual company's stock prices. For integrity throughout the project, we considered Adjusted Close price as everyday stock price. To analyze the sentiment of the news headlines, we used 2 techniques. In the first technique, we used VADER sentiment Analyzer from NLTK package, which is trained using social media and news data. In the second approach, we used the bag of words model.

We used multiple number of modifications to the two baselines. In general using PCA improved the performance of our models.In addition, combining the Linear method(Elastic net) with the non-linear method(Decision tree regressor), showed reduction in the MAE and RMSE. We also used clustering techniques for classification to show the effect of news sentiment to rise and fall of the stock return prices.

2 LITERATURE SURVEY

In the finance field, Stock market forecasting is very important in the planning of business activities. Early research on stock market prediction was based on random walk theory and the Efficient Market Hypothesis[5] and the only thing we need to do is to follow the market[8], however such method failed to make use of data which has its own value. With the failure of Efficient Market Hypothesis, a popular theory about factors that can explain the stock movements[6] has attracted researchers to capture the volatility and predicting its next moves[9]. There two distinct trading philosophies for stock market prediction: fundamental analysis and technical analysis[10]. Technical analysis considers past price and volume to predict the future trend whereas fundamental analysis, on the other hand, focus on analysing activities one business involved and its financial data to get some insights.

However, in addition to historical prices, The overall social mood with respect to a given company is now being considered as an important variable which affects the stock price of that company[11]. Online information such as public opinions, social media discussions[10], and Wikipedia activities[13] are being used to study their impact on pricing and investors' behaviors.

Research shows that combining the data from news and social media along with the historical data can be a good approach to predicting stock prices of a company. It's theoretically right that news should also have an impact according to behaviour finance theory, however, it cannot be predicted quantitatively without the application of text mining technologies.

A lot of great works are done in the popular topic of forecasting stock price with techniques of Machine Learning. Text information including stock-specific and sub-industry-specific news articles are used in stock price prediction[12]. However, the limitation is that in this work two categories of news were examined: news articles related to a target stock and news articles relevant to its sub industry. In our project we want to evaluate the stock market as a whole part.

3 DATA DESCRIPTION

3.1 Data Preparation

We used the Russell 2000 stock indices, collected from Yahoo finance website[2] and collected news articles[4] from NY Times Archive API[1] over the span of 20 years, from January 1st of 2000 to December 31st of 2019. Afterwards, we removed categories of articles, which were irrelevant to stock market. Article sections that are kept at the end for sentiment analysis are as follows: 'Business', 'National', 'World', 'U.S.' , 'Politics', 'Opinion', 'Tech', 'Science', 'Health' and 'Foreign'. Out of 66M articles, approximately 719k articles are filtered out after applying the above filters.

We concatenated all the articles headlines for a single day and merged them with appropriate date and Adjusted Close price of Russell 2000 stock index value. Composite index prices such as Russell 2000 reflect the overall change in the stock market. In general, the machine will get the output for one individual stock wrong most of the time, but when combined with other stocks, the variance in each stock insight will balance out. Therefore, the machine has a higher probability of getting the output right on average when we draw insights for a combination of stocks. Hence, most researchers prefer to predict stock prices of composite index instead of predicting individual company's stock prices.

As the stock market is closed on weekends and US holidays, there are no open/close prices for any of the stocks on those days. We have used the interpolation method from pandas package to interpolate the prices and fill in the missing values.

3.2 Raw Data Statistics

As a result, we retrieved 20 years news data and stock market index. In total, we have 7302 data points and each data point represents one single market day, starting from January 3, 2000 to December 30, 2019. Originally, our dataset comes with several labels. The first is date, we have two prices using Russell 2000 index's close price and adjusted price on the US market, ranging from 327 to 1740. In the following work, we select the adjusted price as the primary indicator of the market price.

As a text dataset, the part of new titles contain 527661 vocabulary in total. Everyday, there are 300 articles published on the New York Times and contains 1984 words in average. The median value of these two values are 264 and 1860. Besides, we use the titles of the news article as the articles label, the average length of news articles consists 6.61 words. More statistics of the news data is provided in Table 1.

Table 1: Statistics of Daily News

	Number of Articles	Words
minimum	7	37
maximum	764	7214
mean	300	1984
median	264	1860
sum	2191124	14486696

Because of the lack of features in the original dataset we use, we added several features to the dataset. First we used VADER[7] to analyze the sentiment of the articles and therefore added four distinct features. They are labelled as 'compound', 'neg', 'neu' and 'pos', which stand for compound, negative, neutral and positive sentiment. The compound label was assigned a value from -1 to 1, more closer to -1 means the compound sentiment is more negative and vice versa, all other three values were assigned a value ranging from 0 to 1, represent the level of the corresponding sentiment.

Also, in the term of text length, we added the number of daily articles and the number of the words uses in that as two different labels.Because the stock market is extremely tortuous, we just simply used the change of close price compared to the previous day's close price as another label.

3.3 Data Analysis

As we derived some features and added them to our dataset, we did more explorations about our dataset. The very important part of this project is the analysis of the articles.

When constructing the features, we follow some guides from some online competitions and articls[3]. Based on the VADER method we used in this work, different types of sentiments can be computed. We first look at all these sentiments results and do the statistics in Table 2. The most significant finding is that the compound sentiment is quite negative, reaching the average value of -0.4839 and median value of -0.9903, which is an interesting finding. The market index is growing gradually in the time period we selected, which is contrary to the mean value of the compound sentiment, making it difficulty to explain the relationship between the media opinion and market change.

To find the linear relationship between the sentiment we computed and the stock market index, we did the correlation analysis. The finding is shown in Figure 1.

From the figure and the correlation analysis we did, we can find the index negatively correlate with the sentiment, it is hard for us to explain now. We use all four sentiment to do the analysis and the results are similar. Then we analyzed the cross correlation between price and compound sentiment. We set the lag from 1 to 30, trying to find the impact of news on the market with the lag. The results show that the correlation raise to the value of 0.026 with the 4 days lag.

4 EXPERIMENTAL SETTINGS

4.1 Split and Cross Validation Choice and Models Hyperparameter

For baseline1,we used the **k-fold sequential cross validation (k-SCV).** Since stock market data is of the form of time series, other

Table 2: Sentiment Analysis

	Mean	Median
Compound	-0.4835	-0.9903
Negative	0.1157	0.1150
Neutral	0.7879	0.7880
Positive	0.0958	0.0950

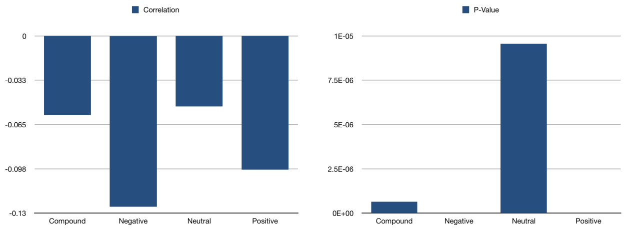


Figure 1: Correlation between prices and sentiments

methods such as ordinary k-fold cross validation are not applicable. In this method, we train on all days up to a specific day and test for the next days. For the purpose of our analysis, we use k = 3, 6, and 12. For example for a 6-SCV, we trained our model on data from January first to October 31st of every year and tested it for the remaining of the same year.

For baseline 2, **we used 70 percent as training data and the other 30 percent as test data**, because for baseline2, we have 2000 words to form a bag-of-words as 2000 dummy variables, we need much more data to train. When using the Elastic net regression model, we choose alpha = 0.00028 we also calculate the same evaluation matrices for baseline 1(alpha=0) as baseline 2 for comparison.

For both baselines, The evaluation metrics used for comparing the performance of our models include **RMSE, MAE,R2, bias.** Because we are predicting the stock return.

For classification models we used accuracy and f1 scores.

4.2 System Settings

We use laptops running Jupyter Notebooks.
Laptop 1: Comodity computer with RAM: 16GB, GPU: Intel UHD Graphics 630, CPU: Intel Core i7-8750H CPU @ 2.20GHz
Laptop 2: Comodity computer with RAM: 64GB, GPU: Intel UHD Graphics 630, CPU: Intel Core i5-9300H

5 BASELINE RESULTS AND DISCUSSIONS

5.1 Baseline Description

In the first baseline, we modified VADER sentiment Analyzer from NLTK package(Add extra lexicon data), which is trained using social media and news data. In the second approach, we create the bag-of-words with 2000 most frequent used words then train the sentiment of our own.

5.1.1 Baseline 1 Description. The first baseline is taken from an article published by Dinesh D[4].In this baseline, we have used VADER Sentiment analysis of our merged news headlines. Output of sentiment analysis is then fed to machine learning models from scikit-learn library to predict the stock prices of DJIA indices. The machine learning models used in this baseline are Random Forest, Logistic Regression and Multi-Layer Perceptron(MLP) Classifiers.

As the prices of the stocks fluctuate a lot, we have used a technique called smoothing which is used in financial markets to take a moving average of the values, which results in comparatively smooth curves. For moving average implementation, we have used the EWMA method from pandas package.

5.1.2 Baseline 2 Description. For baseline 2, instead of using the article and use the VADER sentiment analysis model, we trained the words sentiment to the Russell index and built Elastic net model to predict. As we mentioned before, because VADER is for common news but not for financial news and financial indexes specially, it's better to train new sentiments on the financial news, variables and use it.

The steps are as following, first we count all the vocabularies and then calculate the frequency of the words. After that, we sort the words and get the top 2000 frequent-used words. Then we construct the bags-of-word for each date and we treat each word as a variable, if it appears 3 times in today's news, its value is set to 3. Since now we have 2000 variables, we would like to use Elastic net method to select variables automatically: $\beta = argmin(||y-X||^2+\lambda_1|\beta|_1+\lambda_2\beta^2)$ The prediction result with baseline 2 for test data is showed in Figure 2, with the similar range and characteristics to the real data.

5.2 Result Discussion

The result is very interesting, because we use 3 days log return, 10 days log return and 22 days log return, each represents short term, midterm and long term, we observe that we predict relative better in long term rather than the short term.The R-Square is getting bigger and bigger.

Also, we compared between baseline 1 and baseline 2, we find that baseline 1 is slightly better in almost all error evaluation matrices, though we only use 4 variables in baseline1, that means our self-designed model and self-trained sentiment for words is still not better than VADER.

6 PROPOSED METHOD

In one experiment, we updated the VADER lexicon with words and sentiments from other sources/lexicons such as the Loughran-McDonald Financial Sentiment Word Lists, and ran the various models mentioned above on the new lexicon and compared the results. In this experiment we also used Russel stock index instead of DJIA.

Another important modification was using stock return instead of stock price, which is defined as

$$\text{stock return} = \frac{\text{today's stock price} - \text{yesterday's stock price}}{\text{yesterday's stock price}}$$

We also decided to make use of the n-day lagged sentiment score, which allows us to compare today's article headlines to stock returns of n days later. This is an important point as we need our score index to predict the future, not to tell us what is happening in the present. The model we used for this purpose was elastic-net.

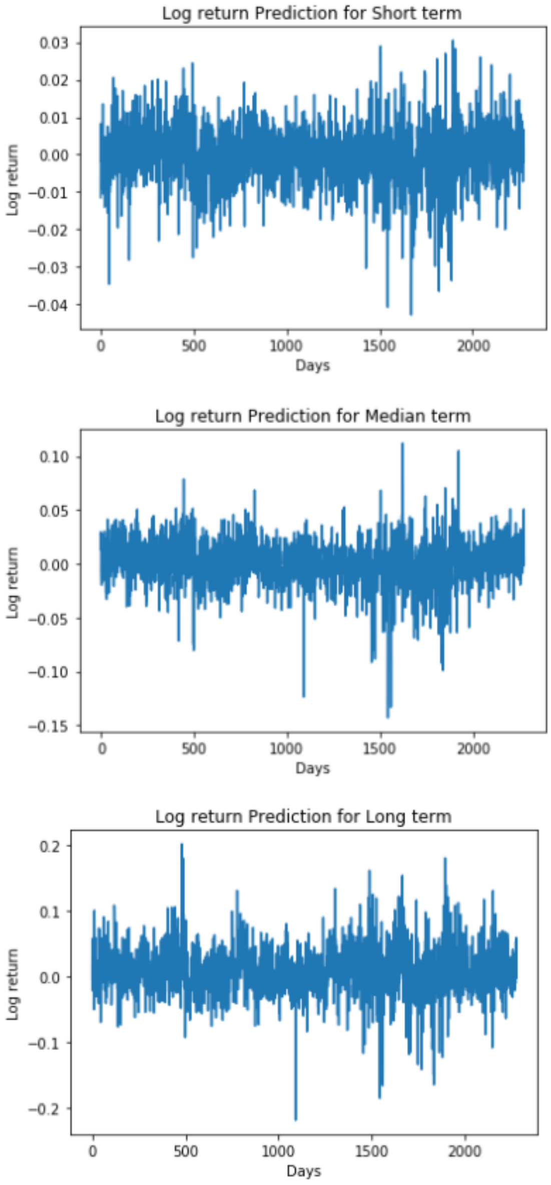


Figure 2: Prediction Result for Different Time Scale

As a second exploration to baseline 1 we used clustering techniques for classification to show the effect of news sentiment to rise and fall of the stock return prices. To do so, we labeled returns more than 0 as class 1, and returns less than 0 as class 0 and used sentiment analysis to predict this rise and fall. For this prediction we used random forest classifier and LDA.

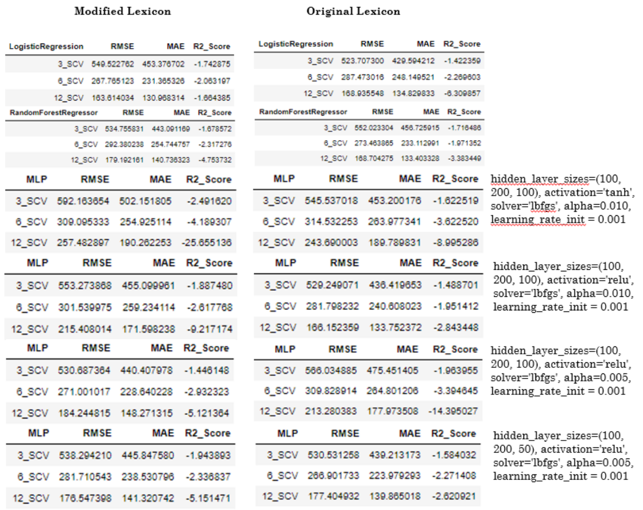
Moreover, we decided to allow a margin of error. We labeled returns more than 1% as class 1, and returns less than -1% as class −1, and everything in between as class 0. Then, We used LDA and random forest to predict this classification based on VADER sentiment analysis of the articles. For the random forest models, after hypertuning, we used the best max_depth of 2 and n_estimators of 16.

We also tried explorations on the second baseline. From the input variables part, we had 2000 word vectors for each day previously. However,now, using Principal component analysis, we are able to find the main variance among all these word variables by reducing the variables' size to 100. Therefore, this method increased the model's stability a lot. It also revealed that in the long term prediction,the average error decreased from 5% to 4%.

We also incorporated a non-linear prediction model Decision tree regressor, and combined its prediction results with the previous Elastic-net method as a weighted average. We observed that in the long term prediction,the average error decreased from 4% to 3%.

7 EXPERIMENTS

Below table shows the comparison between the results of the first baseline and the modified VADER lexicon. In general only in a few cases the modified lexicon worked better, the original VADER lexicon worked better in the rest of the cases as presented in Figure 3.



Modified Lexicon				Original Lexicon				
LogisticRegression	RMSE	MAE	R2_Score	LogisticRegression	RMSE	MAE	R2_Score	
3_SCV	549.522762	453.376702	-1.742875	3_SCV	523.707300	429.594212	-1.422359	
6_SCV	267.765123	231.365326	-2.063197	6_SCV	287.473016	248.149521	-2.269603	
12_SCV	163.614034	130.968314	-1.664385	12_SCV	168.935548	134.829833	-6.309857	
RandomForestRegressor	RMSE	MAE	R2_Score	RandomForestRegressor	RMSE	MAE	R2_Score	
3_SCV	534.755831	443.091169	-1.678572	3_SCV	552.023304	456.725915	-1.716486	
6_SCV	292.380238	254.744757	-2.317276	6_SCV	273.463865	233.112991	-1.971352	
12_SCV	179.192161	140.736323	-4.753732	12_SCV	168.704275	133.403328	-3.383449	
MLP	RMSE	MAE	R2_Score	MLP	RMSE	MAE	R2_Score	hidden_layer_sizes=(100, 200, 100), activation='tanh', solver='lbfgs', alpha=0.010, learning_rate_init = 0.001
3_SCV	592.163654	502.151805	-2.491620	3_SCV	545.537018	453.200176	-1.622519	
6_SCV	309.095333	254.925114	-4.189307	6_SCV	314.532253	263.977341	-3.622520	
12_SCV	257.482897	190.262253	-25.655136	12_SCV	243.690003	189.789831	-8.995286	
MLP	RMSE	MAE	R2_Score	MLP	RMSE	MAE	R2_Score	hidden_layer_sizes=(100, 200, 100), activation='relu', solver='lbfgs', alpha=0.010, learning_rate_init = 0.001
3_SCV	553.273868	455.099961	-1.887480	3_SCV	529.249071	436.419653	-1.488701	
6_SCV	301.539975	259.234114	-2.617768	6_SCV	281.798232	240.608023	-1.951412	
12_SCV	215.408014	171.598238	-9.217174	12_SCV	166.152359	133.752372	-2.843448	
MLP	RMSE	MAE	R2_Score	MLP	RMSE	MAE	R2_Score	hidden_layer_sizes=(100, 200, 100), activation='relu', solver='lbfgs', alpha=0.005, learning_rate_init = 0.001
3_SCV	530.687364	440.407978	-1.446148	3_SCV	566.034885	475.451405	-1.963955	
6_SCV	271.001017	228.640228	-2.932323	6_SCV	309.828914	264.801206	-3.394645	
12_SCV	184.244815	148.271315	-5.121364	12_SCV	213.280383	177.973508	-14.395027	
MLP	RMSE	MAE	R2_Score	MLP	RMSE	MAE	R2_Score	hidden_layer_sizes=(100, 200, 50), activation='relu', solver='lbfgs', alpha=0.005, learning_rate_init = 0.001
3_SCV	538.294210	445.847580	-1.943893	3_SCV	530.531258	439.213173	-1.584032	
6_SCV	281.710543	238.530796	-2.336837	6_SCV	266.901733	223.979293	-2.271408	
12_SCV	176.547398	141.320742	-5.151471	12_SCV	177.404932	139.865018	-2.620921	

Figure 3: Comparison between Baseline 1 and Modified Lexicons

We also compared this modified baseline with the results of our second baseline, which you can see in Table 3. This poor overall performance was also predictable, when looking at the correlation between the sentiment scores and stock prices in Figure 4.

We used LDA and Random forest to predict classification of rise and fall of the Russell index based on VADER sentiment analysis of the articles. In one Experiment we only used compound column, in another we used all 4 sentiments to train our models as shown in Figure 5.

Moreover, we decided to allow a margin of error.We labeled returns more than 1percent as class 1, and returns less than -1 percent as class −1, and everything in between as class 0. Then we used LDA and Random Forest again to make predictions as shown in Figure 6.

Table 3: Baseline Results

Results/Baselines	Baseline1	Baseline2
Bias(3-day Lag return)	0.000305	0.0006586
Bias(10-day Lag return)	0.0004725	0.0008652
Bias(22-day Lag return)	0.0006527	-0.0040737
MAE(3-day Lag return)	0.01058	0.0124
MAE(10-day Lag return)	0.02008	0.0255
MAE(22-day Lag return)	0.029187	0.0407
RMSE(3-day Lag return)	0.01419	0.0161
RMSE(10-day Lag return)	0.026577	0.0333
RMSE(22-day Lag return)	0.0388	0.0533
R-Square(3-day Lag return)	0.003	0.2980
R-Square(10-day Lag return)	0.0065	0.5927
R-Square(22-day Lag return)	0.00635	0.9085

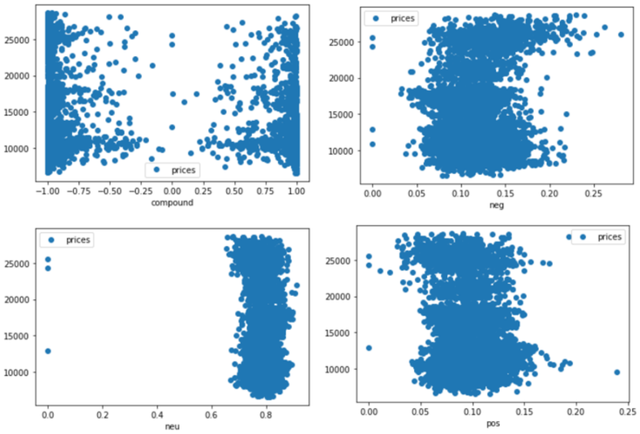
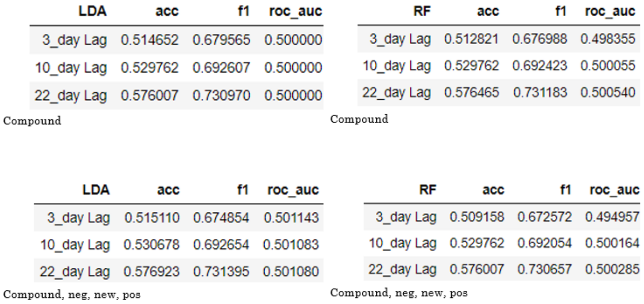


Figure 4: Correlation between the Sentiment Scores and Stock Prices



LDA	acc	f1	roc_auc
3_day Lag	0.514652	0.679565	0.500000
10_day Lag	0.529762	0.692607	0.500000
22_day Lag	0.576007	0.730970	0.500000

Compound

RF	acc	f1	roc_auc
3_day Lag	0.512821	0.676988	0.498355
10_day Lag	0.529762	0.692423	0.500055
22_day Lag	0.576465	0.731183	0.500540

Compound

LDA	acc	f1	roc_auc
3_day Lag	0.515110	0.674854	0.501143
10_day Lag	0.530678	0.692654	0.501083
22_day Lag	0.576923	0.731395	0.501080

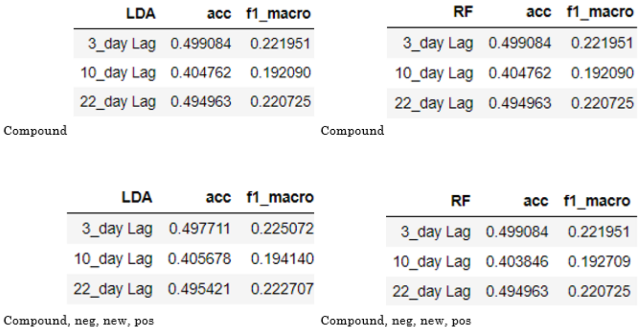
Compound, neg, new, pos

RF	acc	f1	roc_auc
3_day Lag	0.509158	0.672572	0.494957
10_day Lag	0.529762	0.692054	0.500164
22_day Lag	0.576007	0.730657	0.500285

Compound, neg, new, pos

Figure 5: Results using LDA and Random Forest

For the modified baseline 2 experiments, after we apply the Principal component analysis, we reduced the long term prediction absolute error by 1%, after we incorporate the non-linear model Decision tree regression,we obtained another 1% decrease in the long term, which showed in Figure 7.



LDA	acc	f1_macro
3_day Lag	0.499084	0.221951
10_day Lag	0.404762	0.192090
22_day Lag	0.494963	0.220725

Compound

RF	acc	f1_macro
3_day Lag	0.499084	0.221951
10_day Lag	0.404762	0.192090
22_day Lag	0.494963	0.220725

Compound

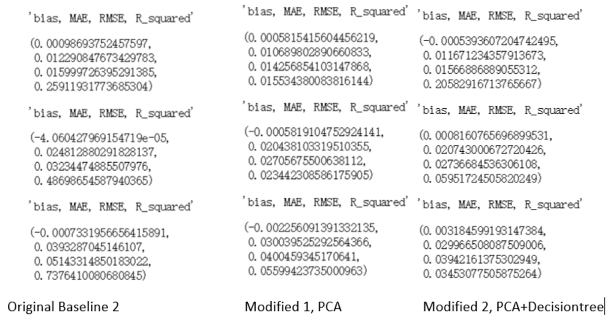
LDA	acc	f1_macro
3_day Lag	0.497711	0.225072
10_day Lag	0.405678	0.194140
22_day Lag	0.495421	0.222707

Compound, neg, new, pos

RF	acc	f1_macro
3_day Lag	0.499084	0.221951
10_day Lag	0.403846	0.192709
22_day Lag	0.494963	0.220725

Compound, neg, new, pos

Figure 6: Results using LDA and Random Forest with Margin Inclusion



```
'bias, MAE, RMSE, R_squared'

(0.00098693752457597,
 0.01229084767342978З,
 0.015999726395291385,
 0.25911931773685304)

'bias, MAE, RMSE, R_squared'

(-4.060427969154719e-05,
 0.024812880291828137,
 0.03234474885507976,
 0.48698654587940365)

'bias, MAE, RMSE, R_squared'

(-0.0007331956656415891,
 0.0393287045146107,
 0.05143314850183022,
 0.7376410080680845)
```

Original Baseline 2

```
'bias, MAE, RMSE, R_squared'

(0.0005815415604456219,
 0.010689802890660833,
 0.014256854103147868,
 0.015534380083816144)

'bias, MAE, RMSE, R_squared'

(-0.0005819104752924141,
 0.020438103319510355,
 0.02705675500638112,
 0.023442308586175905)

'bias, MAE, RMSE, R_squared'

(-0.002256091391332135,
 0.030039525292564366,
 0.0400459345170641,
 0.05599423735000963)
```

Modified 1, PCA

```
'bias, MAE, RMSE, R_squared'

(-0.0005393607204742495,
 0.011671234357913673,
 0.01566886889055312,
 0.20582916713765667)

'bias, MAE, RMSE, R_squared'

(0.0008160765696899531,
 0.020743000672720426,
 0.02736684536306108,
 0.05951724505820249)

'bias, MAE, RMSE, R_squared'

(0.003184599193147384,
 0.029966508087509006,
 0.03942161375302949,
 0.03453077505875264)
```

Modified 2, PCA+Decisiontree

Figure 7: Results of Baseline 2 Compared with its Modified Versions

In general the results of our experiments show that news, people, and financial market are closely related, and our method explores that relation. By combining linear and non-linear models, we achieved an improvement according to our error metrics. Also, classification method showed a bettwe accuracy over the regression method. Even though the small modification to the lexicon didn't show good results, it's a starting point.

Comparing the various number of lag-days has benefit us in finding the optimum. It shows most of the improvement from modification of baseline 2 is on long term prediction result.

8 CONCLUSION

This project provides a financial news sentiment analysis model, for predicting future stock price index in the investment market. As planned, this framework seemed to be mildly predictive of the relationship between new sentiment and stock market movement. However, there are a lot of factors can affect the market movement. Additionally, the data we collected is limited to general economy-related news and we didn't consider the more detailed and more specific news such as news related to one targeted company, so the result is not so accuracy.

Our framework is still in an early stage but its initial exploration meets our expectation. Future work may involve developing a sentiment analyzer which could work better in specified industry situations. In our work, we only use news headlines from one newspaper to train a lexicon-based analyser, it is very likely that our analyser will face overfit problem and don't have the generic ability to analysis other news articles. Updating current lexicon-based sentiment analyzers with more lexicons from news articles can also be a good approach. Also, in our project we only considered news article sentiment analysis for prediction but in the real scenarios, stock fluctuations show trends which get repeated over a period of time. So there's a lot of scope in merging the stock trends with the sentiment analysis to predict the stocks which could probably give better results.

9 CONTRIBUTION

Maiqi was responsible for data analysis. Nazanin was in charge of data collection, data preparation, making the ppt presentation, baseline 1 and its modifications. Xiaoyu was responsible for baseline 2 and its modifications. The entire group completed writing the report together. All team members have contributed a similar amount of effort.

REFERENCES

[1] 2020. New York Times archive api. https://developer.nytimes.com/archive_api.json

[2] 2020. Yahoo Finance Website. https://finance.yahoo.com/quote/%5ERUT

[3] Yanzhou Wang Barthold Albrecht and Xiaofang Zhu. 2019. *Kaggle Competition 2sigma Using News to Predict Stock Movements.* http://cs229.stanford.edu/proj2018/report/82.pdf

[4] Dinesh D. 2017. *Stock Predictions through News Sentiment Analysis.* https://software.intel.com/en-us/blogs/2017/07/14/stock-predictions-through-news-sentiment-analysis

[5] Eugene F. Fama. 1965. The Behavior of Stock-Market Prices. *The Journal of Business* 38, 1 (1965), 34–105. http://www.jstor.org/stable/2350752

[6] Eugene F. Fama and Kenneth R. French. 1993. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics* 33, 1 (1993), 3 – 56. https://doi.org/10.1016/0304-405X(93)90023-5

[7] C.J. Hutto and Eric Gilbert. 2015. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014.*

[8] Michael C. Jensen, Fischer Black, and Myron S. Scholes. 2006. The Capital Asset Pricing Model: Some Empirical Tests.

[9] Kalyani Joshi, Bharathi N, and Jyothi Rao. 2016. Stock Trend Prediction Using News Sentiment Analysis. *International Journal of Computer Science and Information Technology* 8 (06 2016), 67–76. https://doi.org/10.5121/ijcsit.2016.8306

[10] Thien Hai Nguyen, Kiyoaki Shirai, and Julien Velcin. 2015. Sentiment analysis on social media for stock movement prediction. *Expert Syst. Appl.* 42 (2015), 9603–9611.

[11] Dev Shah, Haruna Isah, and Farhana Zulkernine. 2018. *Predicting the Effects of News Sentiments on the Stock Market.*

[12] Y. Shynkevich, T. M. McGinnity, S. Coleman, and A. Belatreche. 2015. Stock price prediction based on stock-specific and sub-industry-specific news articles. In *2015 International Joint Conference on Neural Networks (IJCNN).* 1–8.

[13] Pengyu Wei and Ning Wang. 2016. Wikipedia and Stock Return: Wikipedia Usage Pattern Helps to Predict the Individual Stock Movement. 591–594. https://doi.org/10.1145/2872518.2890089