Sentiment Analysis on Financial News Headlines and Stock Returns

---

Milestone 1 Thesis Project - SIADS 591 & 592 - May 1, 2021

Project Team:
Yizhuo Wang (yzw@umich.edu), Thomson Choi (tschoi@umich.edu)

Instructional Team:
Chris Teplovs (cteplovs@umich.edu), and Anthony Whyte (arwhyte@umich.edu)

GitHub Repository: *MADS-Thesis-Project*

1. Motivation

The regular up and down of stock fluctuation caused by supply and demand. More people buy a stock that price rises, and conversely, more people sell a stock that price falls. Early before, the investors sentiment and cross-section of stock returns is proven correlated (Baker, Wurgler, 2006), and more recently, a study declaring the relationship between supply and demand is highly sensitive to the news of the moment and the role of media coverage can explain stock market fluctuations (Strycharz et al. 2018). We want to determine whether there is a correlation between the news sentiment and stock fluctuation in which mediately affects investors actions. For this project, we will be looking into the sentiment results from financial news headlines and some cross-section stock fluctuation overtime relying on the visualization techniques to illustrate the correlations.

We clearly know that there are many factors affecting the stock fluctuation. In our analysis, we assume that the people transforming the news sentiments into trading actions are non-professional investors, who sell with negative news and buy with positive news. We will not go into the sentimental causation of actions happening in people. We infer that the strong news sentiments affect these investors' judgement and action significantly as result, and then causes the stock fluctuation after the rapid butterfly effect happening on trading.

2. Brief Introduction to Sentiment Analysis in Text

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language (Collobert et al. 2011), specifically how to program computers to process and analyze large amounts of natural language data.

Sentiment analysis refers to the use of NLP, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. There are mainly three levels of sentiment analysis, which are document level, sentence level and entity and aspect level. In our case, sentiment analysis is sentence level which is the process of understanding if a given text from news headlines is telling positively, negatively, neutrally or compound. The reason we choose headlines is because they have similar length, easier to parse and group than full articles, which vary in length.

3. Data Sources

Data Source	**Kaggle.com**	**Yahoo**
Name	Daily Financial News for 6000+ Stocks (2009-2020)	Yahoo Finance via yfinance package
Size	Company research analyst ratings titles: 320MB / 1.4 million records Financial news headlines: 391MB / 1.8 million records	Estimated 5 MB for our selection
Location	https://www.kaggle.com/miguelaenlle/massive-stock-news-analysis-db-for-nlpbacktests	https://github.com/ranaroussi/yfinance https://aroussi.com/post/python-yahoo-finance
Format	CSV files inside a zip archive	A Pandas dataframe is returned by the yfinance Python package which enables download of stock market data from Yahoo Finance website.
Access Method	Download from Kaggle.com	Install yfinance package in python
Important Variables	Headlines, ticker symbol and date	Adjusted closing price, ticker symbol and date
Records Used	Dow Jones 30 stocks	AXP, DOW, HD,TRV,WBA
Time Period	2009-2020	2016-2020

4. Data Manipulation Methods

A. NLTK & VADER Brief Introduction

We try to classify sentiments from financial news headlines using Rule-Based Natural Language Toolkit (NLTK) and Valence Aware Dictionary and sEntiment Reasoner (VADER).

In cases where many researchers preferred using statistical NLP to tackle the problems, but in our case, the regular rule-based approach might be fruitful. The financial news is a domain specific subject, therefore using the rule-based NLP can bring us high accuracy levels.

The affective content of text is carried often by sentiment-bearing lexical items used metaphorically in a new domain (Devitt, Ahmad, 2007). A lexicon is the vocabulary of a person, language, or branch of knowledge. In our rule-based NLP study, we need a lexicon that serves as a reference manual to measure the sentiment of headlines' text. We will be building a Lexicon-based Sentiment Classifier. The tool we will be utilizing is VADER, which is a model from Python's most advanced NLP tool, NLTK. VADER is like the GPT-3 of rule-based NLP models. We can easily import a SentimentIntensity Analyzer, and it will generate acceptable F1 scores on the sentiments' polarity.

B. Data Cleaning and Preparing

News and Analyst Ratings Headlines Data

The data was downloaded as csv and read into two Pandas Dataframes. There are 6000+ stocks covered from 2009 to 2020, with 1.8 million news headlines and 1.4 million analyst ratings headlines. After preliminary exploration, we observe that the headlines varies greatly in quantity and time coverage for each stock, for example some stocks have news or analyst ratings coverage for a short period of time only, or some stocks have very few headlines overall.

Number of Headlines

In order to design an analysis pipeline, we started by **filtering down the stocks universe** (*Pandas:* `isin`) in which we would explore the data by looking at stocks which belong in the Dow Jones Industrial Average Index ("DJIA") which contains 30 stocks of large companies in the U.S. We choose this index because the large companies are very well-known and we expect to see significant coverage in the news and analyst ratings, so that we have enough data points for a correlation analysis with stock prices. However with the DJIA universe, we still saw companies with either very few news headlines or few analyst ratings headlines (*Pandas:* `groupby`*,* `agg`) in the dataset, so we further trimmed the universe to **include stocks with >10 headlines & >10 ratings**, which gave us 9 stocks shown on the right chart with the number of headlines in each Dataframe.

	headline	ratings
AAPL	32.0	441.0
AXP	1926.0	1857.0
CSCO	508.0	1010.0
DOW	2418.0	1227.0
HD	1841.0	2612.0
KO	77.0	2797.0
TRV	2203.0	656.0
UNH	170.0	230.0
WBA	1375.0	1194.0

Time Period Coverage

The date column of the dataset was in string format and had NaN values, we wrote a helper function `time_conv` to `map` to a datetime data type in Pandas. By observing the min and max dates of headlines of the above 9 stocks, we saw most stocks have data ending May or June 2020. Some stocks have data starting late 2019 or 2020 only, which are **too short so we exclude** these and **only include stocks with data starting prior to 2019 only** *(Pandas filter < datetime of 2019-01-01)***.** Since we have two Dataframes, we did `groupby` and `agg` on both and used `merge` to combine them. This gives roughly 1.5 years of data minimum, and we are left with **5 stocks**.

	min_headlines	max_headlines	min_ratings	max_ratings
stock				
AXP	2016-04-20	2020-06-03	2009-08-07	2020-06-10 10:22:11
DOW	2010-03-01	2020-06-03	2009-08-25	2020-06-09 10:52:15
HD	2018-04-23	2020-06-04	2009-08-10	2020-06-10 08:14:08
TRV	2010-03-05	2020-05-29	2009-10-22	2020-06-11 10:05:13
WBA	2018-03-23	2020-05-27	2014-12-31	2020-06-08 11:34:19

Sentiment Score Processing on News Headlines and Analyst Ratings Headlines

We aim to examine the overall affective polarity by using sentiment scores derived from headlines text that can be compared with stock price and returns. Additionally, we will count out the highest frequency words appeared for these sentiments.

We use VADER by importing `nltk.sentiment.vader` and `SentimentIntensityAnalyzer`. Since VADER is lexicon based and each word has a sentiment value assigned, we wanted to prepare it by updating the lexicon with more relevant words for our use case which is financial text.

Since financial news have specific terminology, in order to make VADER more comprehensive, we involve two extra sources to enrich the lexicon. We patched the lexicon with roughly 3,000 positive/negative words from the Loughran McDonald Sentiment Word Lists. The textual analysis and words collected in financial applications are published by Tim Loughran and Bill McDonald (Loughran, McDonald, 2016). However, these positive/negative words do not have sentimental values: we set the values of words to `mean` values of positive words and negative words respectively from the existing lexicon contained from the nltk package. (Pandas: read_excel and set the words that do not exist in lexicon to values of the mean from existing lexicon), Moreover, we added a few financial specialized vocabulary manually, which is collected and valued by other specialties. Finally, we `update` the lexicon.

To add sentimental scores onto headlines columns, we apply `vader.polarity_scores` on the headlines column, and we obtain scores of negative, neutral, positive and compound.

headline	url	publisher	date	stock	neg	neu	pos	compound
US Indexes End Higher Wednesday	http://www.gurufocus.com/news/1154923/us-indexes-end-higher-wednesday	GuruFocus	2020-06-03 00:00:00	AXP	0.000	1.000	0.000	0.00
Thinking about trading options or stock in Micron Technology, Boeing, Penn National Gaming, ...	http://www.gurufocus.com/news/1154684/thinking-about-trading-options-or-stock-in-micron-technology-boeing-penn-national-gaming-american-express-or-carnival-corp	GuruFocus	2020-06-03 00:00:00	AXP	0.000	1.000	0.000	0.00

Since we have two Dataframes *headlines* (news) and *ratings* (analyst ratings), we wrote a function `add_scores` to manipulate the Dataframe by taking the Dataframe as a variable. We also wrote a helper function `get_score_df` to filter the scores by stock indexed on date from the Dataframes, which will be useful in subsequent analysis.

Stock Price Data

We acquire stock price data from the `yfinance` library built to access Yahoo Finance data. First we observe the date range of the 5 stocks from headlines and ratings by `min` and `max` on date column and create a Dataframe with this information. By doing `max` of the start date of *headline* and *ratings* and `min` of the end date of the *headline* and *ratings* per stock, we have the date period where both are available. This is the date range we try to download the stock price data with (`yfinance.download`).

We found that stock DOW was not available. Further research on Yahoo Finance showed that the stock DOW had data started only 20 Mar 2019 because it was listed as stock ticker DWDP prior to that, and went through a merger. It is not possible to retrieve delisted stock data from this package, and therefore we `drop` stock DOW from analysis, leaving us with 4 stocks. The stock price variable we're interested in is column 'Adj Close' from the download from yfinance.

5. Analysis and Visualization

We use `wordcloud` tool to generate the high-frequency words of each sentiment from the four stocks (AXP, HD, TRV, WBA). To make it easy to distinguish, we remove some companies' names and nonsignificant financial terminologies. The bigger size of the words means heavier weight appearing on the headlines. It is clear to see that some polarized verbs frequently appeared for positive and negative sentiments.

Negative

Neutral

Positive

Compound

We aim to produce an **analysis pipeline that can be repeated and reused across stocks and source of headlines (news or analyst ratings)**. This allows us to run the comprehensive analysis on one stock, and repeat for other stocks, so we can compare the results of them; as well as running using news headlines or analyst ratings headlines. It is also possible to run the same analysis on other stocks outside of the 4 stocks resulting from the Data Cleaning process above, as long as the headlines and price data are available and put in the same format as the expected input to the analysis. To this end, we **parameterized all the variable inputs** starting from *stock_analysis* = 'TRV', and writing **functions** that allow users to define the input Dataframe (*headlines* or *ratings*), which sentiment score type ('pos', 'neg', 'neu', 'compound') and the stock return period in number of days. This is contained in the **notebook** in entirety in **Section 3: Analysis of One Stock.**

Visualization (Section 3.2)

We are using Matplotlib and Altair libraries to summarize and present the visualizations. Matplotlib is an exhaustive library that comprises numerous functionalities. Altair is a declarative statistical visualization library. The combination of using these two will give us more flexible, intuitive, structured and interactive visualization.

A. Visualization - Stock Price, Return and Sentiment Scores

First visualization attempts to visualize the relationship between price, returns and sentiment.

`plot_chart` **function:**

Using Altair, we plot 1) stock price as line or stock price return (**Note on log-return** below) as dot, both in blue, and 2) sentiment scores as dots, visualized in a color map from negative in red to positive in green, overlaying each other.

Note on log-return*: We use log of the stock price on t divided by previous period t-1 to give a natural log percentage return (log-return) in our analysis. Log-return has nice properties for* *financial analysis as it indicates a continuous return, and the returns are additive over multiple periods.*

$$return = ln\frac{P_t}{P_{t-1}}$$

In Pandas, the log return computation is achieved by
`np.log(px_data.shift(-n_days_return) / px_data)`
Inside the log function, In the numerator we use shift to bring the n-days later price to the date of the denominator price, giving us the log-return of the stock price that is forward looking from the date of the headline, as an indication of the 'predictive power' of the headline.

From the visualization of stock price directly vs sentiment, as expected there is no pattern to be observed as the stock price is path dependent. However, even on the log-returns vs sentiment chart, the relationship was also unable to be visually picked up. We will do numeric analysis and use a SPLOM to help us next.

Example: The plot of the **compound** sentiment score from the **analyst ratings headlines** of stock and the stock price.

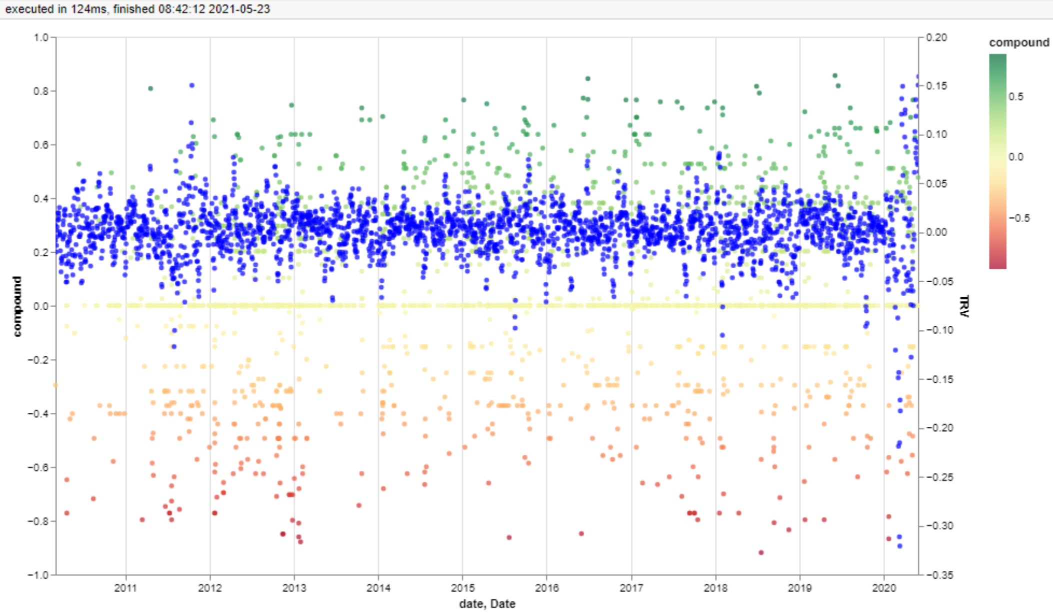
```
plot_chart(ratings, stock_analysis, 'compound', "redyellowgreen", 'price')
```

executed in 109ms, finished 08:42:12 2021-05-23

Example: The plot of the **compound** sentiment score from the **news headlines** of stock and the **5-day returns** of the stock price.

```
plot_chart(headlines, stock_analysis, 'compound', "redyellowgreen", 'return', n_days_return=5)
```

executed in 124ms, finished 08:42:12 2021-05-23

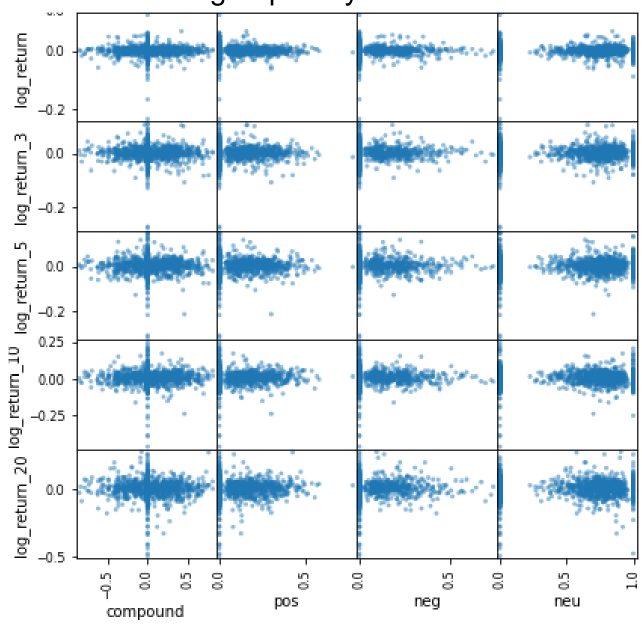


B. Correlation of Sentiment Scores and Stock Price Return Analysis

We then perform a regression analysis on sentiment scores and stock returns with SPLOM and numerically.

`get_analysis` function:

The function takes inputs of Dataframe of headlines and the stock to get to output a Dataframe of time series data with the sentiment scores and forward looking log-returns of 1, 3, 5, 10, 20-day periods. Inside the function, the sentiment scores are grouped by date on the mean of headlines published on the same day. The sentiment scores are also shifted by 1 day later as suggested from the documentation of the data source to allow for time of publication to be reflected during stock trading hours next day.



Visualization: SPLOM

With the output of `get_analysis` we use a SPLOM (*pandas.plotting.scatter_matrix*) to observe patterns between the sentiment scores attributes vs the log_returns.

Example SPLOM on stock 'TRV' with news headlines (Right hand):

We expected to see compound sentiment to have positive relationship with returns, similarly for positive, and a negative relationship on negative sentiment. However, from the SPLOM the relationship was not apparent. We will observe the numerical correlation next.

Numeric Correlation

With a helper function `slim_corr`, we look at the correlation numerically:

Again, the correlation metrics are very weak numerically. We look for alternative analysis to explore further segmentation which might have more conclusive results.

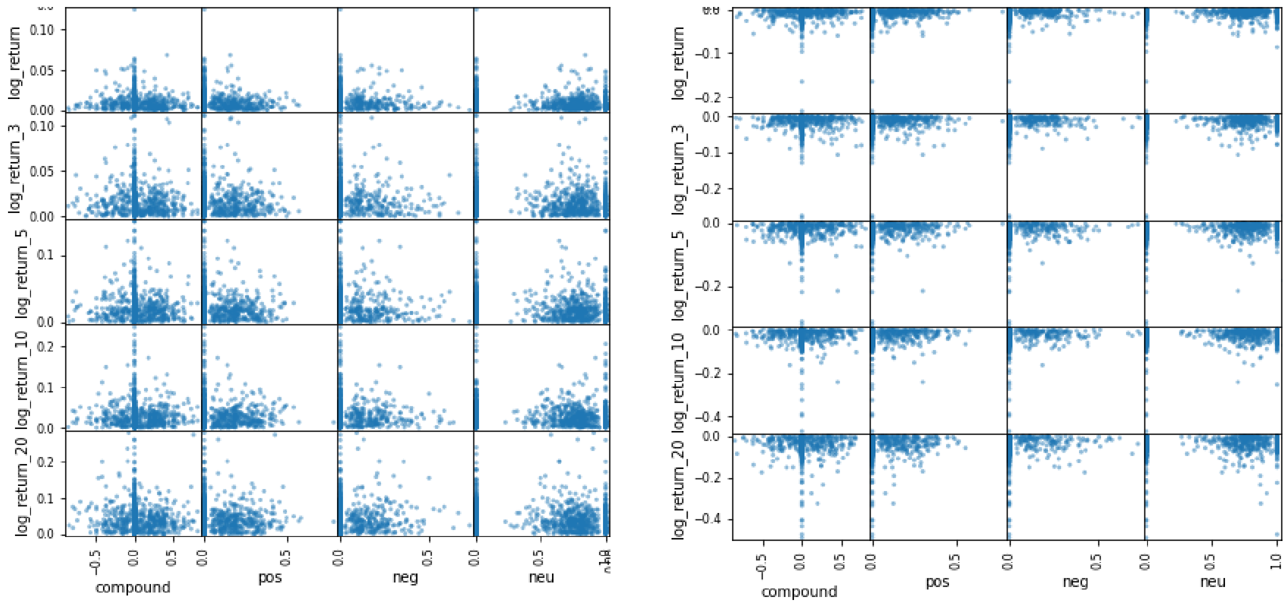
Correlation of Positive/Negative return days vs Sentiment Scores Separately

We want to separate the stock returns dataset on positive days vs negative days and try again the correlation with sentiment scores.

With a helper function `filter_return`, we produced similar Dataframes to repeat the analysis in the previous section.

	compound	pos	neg	neu
log_return	0.000788	-0.004762	0.003746	0.007719
log_return_3	-0.006797	0.005521	0.020218	0.018318
log_return_5	0.002064	0.011517	0.014736	0.022675
log_return_10	0.018400	0.024130	0.010267	0.015283
log_return_20	-0.002329	0.022738	0.031726	0.010929

SPLOM on positive (left) and negative (right) returns on stock 'TRV' with new headlines.



Again, there was still not a very apparent relationship between returns and scores.

Comparison of correlations on full, positive, negative returns versus sentiment scores (Section 3.5)

With a helper function **`comp_corr`**, we produced a comparison of the numeric correlation of above analysis on full, positive, negative returns versus sentiment scores.

We can see the correlations are weak and some values are counterintuitive.

```
comp_corr(score_type='compound')
```

executed in 29ms, finished 08:42:23 2021-05-23

	Full Return	Pos Return	Neg Return
log_return	0.000788	-0.031399	0.036324
log_return_3	-0.006797	-0.001067	-0.026451
log_return_5	0.002064	0.000929	0.000416
log_return_10	0.018400	0.008352	0.003619
log_return_20	-0.002329	-0.028100	-0.009328

Next we attempt a confusion matrix with the frequency of positive and negative returns and sentiment scores. We treat sentiment as predictor and returns as actual observations.

Confusion Matrix

With a helper function **`confusion_matrix`**, we produced confusion matrices of a type of sentiment score by count.

Example: Confusion matrix of stock 'TRV' with 5-day returns and 'compound' scores

	Positive Return	Negative Return
Positive Score	279.0	190.0
Negative Score	171.0	137.0

We will observe the following metrics, assuming a notation of PN to denote positive score / negative returns:

Recall: $PP / (PP + PN)$
Precision: $PP / (PP + NP)$
Specificity: $NN / (NN + NP)$

From the confusion matrix of compound sentiment scores as predictor and with 5-day returns as actual observations we can see:

Stock AXP with new headlines: Recall is 60%, Precision is 68%, Specificity is 37%.
Stock TRV with news headlines: Recall is 59%, Precision is 62%, Specificity is 44%.
Stock HD with news headlines: Recall is 61%, Precision is 87%, Specificity is 57%.
Stock WBA with new headlines: Recall is 48%, Precision is 80%, Specificity is 52%.

Stock AXP with ratings headlines: Recall is 57%, Precision is 65%, Specificity is 47%.
Stock TRV with ratings headlines: Recall is 56%, Precision is 38%, Specificity is 44%.
Stock HD with ratings headlines: Recall is 62%, Precision is 63%, Specificity is 39%.
Stock WBA with ratings headlines: Recall is 50%, Precision is 66%, Specificity is 43%.

From the above we can see news headlines have better prediction power than analyst ratings headlines from their compound sentiment scores. However, the power seems to vary among stocks. And it seems better predicting positive returns than predicting negative returns. This does not give us too much confidence in using the scores as future predictions of stock returns movements.

We will try one more analysis which uses each day's max values of scores (max absolute for compound score) instead of aggregating by mean as the daily sentiment scores.

Further Analysis with daily max sentiment value data points

We modified `get_analysis` to make `get_analysis_maxabs` function that changes the aggregation of multiple headlines on the same day from taking mean to taking the max or max absolute value with sign.

Repeating the same analysis, the `comp_corr` of stock 'TRV' did not improve significantly.

	Full Return	Pos Return	Neg Return
log_return	-0.008118	-0.053797	0.025612
log_return_3	-0.014570	-0.025560	-0.030487
log_return_5	-0.005804	-0.017429	-0.005975
log_return_10	0.004287	0.000883	0.010084
log_return_20	-0.020418	-0.043448	-0.013819

From the confusion matrix of TRV, the result doesn't change much. Below is the difference:

	Positive Return	Negative Return
Positive Score	-1.0	1.0
Negative Score	2.0	0.0

6. Conclusion

With this project we have built a pipeline and functions that allows us to consume data of text headlines on stocks, retrieve stock prices and the log returns and produce visualizations, correlation analysis and confusion matrices.

From our analysis of a small universe of 4 stocks, the results from analysis did not give us a high confidence in significant correlation between the sentiment scores and stock price returns. We tried the analysis with all 4 stocks and both their news headlines and analyst ratings headlines sources, and the result was similarly insignificant as a reliable prediction. This may be due to the sentiment scoring method or stock price returns impacting factors which are external to the news headlines (e.g. general stock market trend).

For future work, it is possible to explore:

- a different lexicon and scoring
- different types of sentiment scoring method
- looking for the number of days returns that would have the most significant results from sentiment score prediction
- running on a large universe of stocks, segmented by industry or market-capitalization and tune the correlation model for more significant results.

7. Statement of Work

Thomson Choi:

- Production of Notebook: All Sections except Word Cloud
- Report: 4. Data Manipulation Methods (except NLTK & VADER brief introduction), 5. Analysis and Visualization, and 6. Conclusion.

Yizhuo Wang:

- Production of Notebook: Exploration of Data Sources, Word Cloud
- Report: 1. Motivation, 2. Brief Introduction to Sentiment Analysis in Text, 3. Data Sources, 8. References and overall formatting.

8. References

Baker M., Wurgler. (2006) *Investor Sentiment and The Cross-Section of Stock Returns*. doi: 10.1111/J.1540-6261.2006.00885.X

Strycharz J., Strauss N., & Trilling D. (2018) *The Role of Media Coverage in Explaining Stock Market Fluctuations: Insights for Strategic Financial Communication, International Journal of Strategic Communication*. doi: 10.1080/1553118X.2017.1378220

Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K., & Kuksa P. (2011) *Natural Language Processing (Almost) from Scratch.* https://leon.bottou.org/publications/pdf/jmlr-2011.pdf

Devitt A., Ahmad K. (2007) *The Language of Emotion and Financial News*. https://www.researchgate.net/publication/235918282_The_Language_of_Emotion_and_Financial_News

Loughran T., McDonald B. (2016) *Textual Analysis in Accounting and Finance: A Survey, Journal of Accounting Research*. http://ssrn.com/abstract=2504147