NUS INVESTMENT SOCIETY
Quantitative Finance Department

Textual Analysis of News Headlines and the Impact on AAPL Stock Price Direction

Arun Kumarr Ravi, Hon Jia Jing, Ng Tze Yuan Edmund

June 2020

Abstract

This paper examines the role of financial news articles in different textual representations and their ability to predict stock price direction one day after an article release. The Bag of Words model has been the dominant approach in textual analysis. This paper seeks to examine the effectiveness of other approaches such as Sentiment Analysis and Word Embeddings, in addition to the Bag of Words model for comparison. These different approaches will then be used as part of a machine learning procedure to evaluate AAPL stock price direction. The results demonstrate that on average, Sentiment Analysis and Word Embeddings outperform the Bag of Words model, especially when an ensemble modeling of different classification models is used.

Introduction & Motivation

Stock market prediction has attracted attention from researchers and investors. Numerous scientific studies have attempted to devise models that predict stock price movement with a high degree of accuracy. The analysis of financial news articles on stock price movement has gained traction in these studies as research has shown that these two factors have strong correlation with one another (Alanyali et al., 2013).

The amount of financial news articles available has increased dramatically in recent years, providing machine learning models with larger training sets and hence, allowing it to chew out increasingly accurate financial models. A significant amount of current literature on financial text mining relies on identifying a predefined set of keywords and assigning weights in proportion to the movement of a share price. These types of analysis have a weak ability to forecast the direction of share prices (Schumaker & Chen, 2006).

Hence, in this paper we are going to use several linguistic textual expressions, including Bag of Words, Word Embedding, and Sentiment Analysis to analyze financial news information. Bag of Words has been the de facto standard for textual representation, however we believe experimenting with other representations that have qualities lacking from Bag of Words such as sentiment score will yield improved results.

The news information and stock prices of Apple Inc (AAPL) is used in this paper due to the amount of relevant news information available and Apple's volatility per price index. Compared to other technology giants such as Amazon and Alphabet, Apple has a relatively low volatility ("Apple Inc (AAPL) Stock Volatility", 2020). This provides a more predictable dataset for the model.

This paper is arranged as follows, the section on Related Work provides an overview of literature concerning Stock Market prediction, textual representations, and machine learning techniques. The section on Methodology describes our research methodology and provides an overview of our experimental design. The sections on Processing of News Headline and Model Selection showcase our different approaches and discuss their implications. Finally, we present our experiment results and deliver our experimental conclusions with a brief discussion on future directions for this stream of research.

Related Work

Stock Market prediction using Machine Learning is a popular project that has been around for a long time, where we try to predict the direction of movement of stock prices in the market. It is extremely challenging to predict stock prices movement with the many factors involved, with the Efficient Market Hypothesis suggesting that it is pointless to try to predict the stock market using technical analysis (Downey, 2020). Furthermore, there are existing studies proving that prediction using historical financial data or news data alone is inefficient (Mohan et al., 2019). Nonetheless, there has been a large improvement in the results of model built, and news information has been shown to have a strong correlation with behaviour of stock prices (Gidófalvi , 2001).

Textual Analysis of news information has been used for a while now to predict stock market trends. The Bag of Words model has been used extensively to analyze financial news information, as seen in the paper written by Schumaker and Chen (Schumaker & Chen, 2006). However there are a number of disadvantages to the Bag of Words model as acknowledged by many researchers. Due to a large number of vocabulary, it leads to high dimensional features, assumes all words are independent of each other, and leads to a highly sparse vector with a lot of zero values. As a result, more textual analysis is being conducted using other approaches such as Sentiment Analysis and Word Embeddings.

Word Embeddings using Word2vec is one of the preferred techniques for Natural Language Processing as it better indicates the similarity and analogy relationship between different words. Goldberg also goes on to say that Word Embeddings benefit computationally since the majority of neural network toolkits do not play well with very high-dimensional, sparse vectors and thus the generalization power of Word Embeddings' dense representation is better (Goldberg, 2017).

Li and the fellow authors of "Knowledge-based Systems" argue that the Bag of Words model's weakness is a result of its lack of sentiment analysis (Li, Xie, Chen, Wang, & Deng, 2014). News sentiments have been shown to be an effective predictor of stock price movement as there is strong correlation between news sentiments and stock price movement, with researchers achieving high accuracy in prediction both with only news sentiments as feature and news sentiments together with financial data as feature (Shah, Isah, & Zulkernine). This can be seen in Li's research which shows that Sentiment Analysis outperforms the Bag of Words model in both validation and independent testing data sets.

Methodology

From the research into related works, it can be seen that both Sentiment Analysis and Word Embeddings perform much better than the Bag of Words approach. Hence, this paper seeks to explore a model that uses a combination of both Sentiment Analysis and Word Embeddings. At the same time, we will also explore a model that uses the de facto standard for textual representation, that is Bag of Words, and compare both models to determine which is a better approach.

An overview of the methodology is displayed in Figure 1.

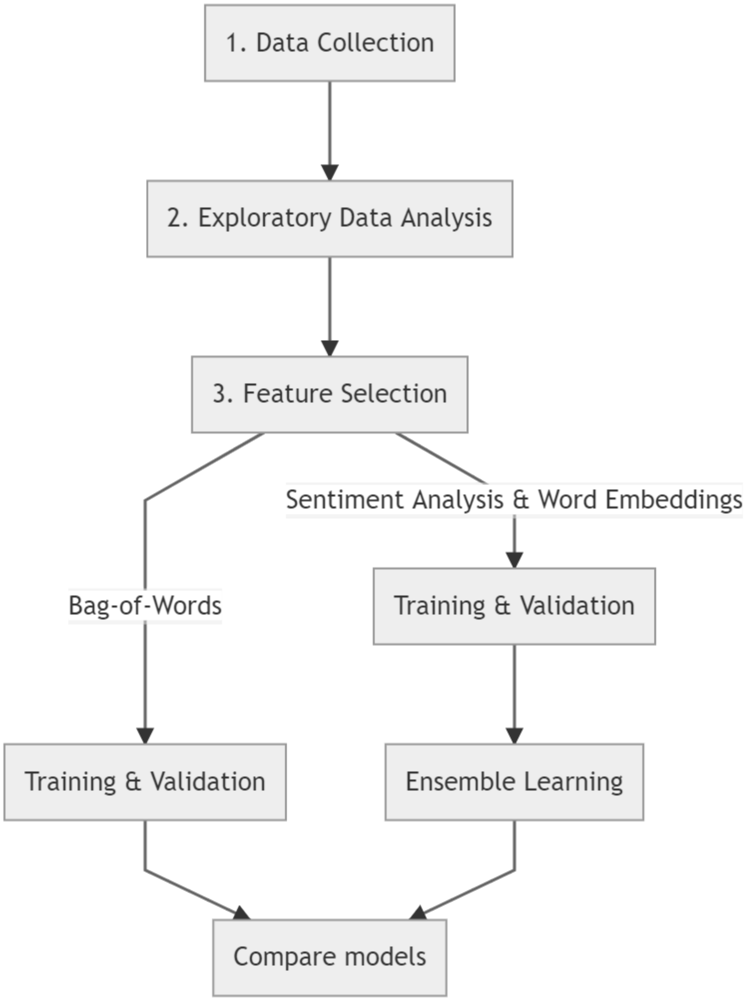


Figure 1: Flowchart diagram of methodology

1) Data Collection

News headlines pertaining to AAPL were collected from Bloomberg terminal, selecting only the top 50 news in English for each month. A total of 1595 headlines were obtained, with dates ranging from 01/12/2016 to 30/12/2019. Financial data pertaining to price data (Open, Close, High, Low, Volume and Adjusted Close) of AAPL were collected from Yahoo! Finance. A total of 774 days of data were obtained, with dates ranging from 01/12/2016 to 30/12/2019.

Financial data pertaining to the quarterly reports (Revenue, Changes in working capital, Dividends paid and Net changes in cash) for AAPL were collected from SimFin, and the quarterly values were interpolated into a daily time series using a spline interpolation function. The interpolation allows the quarterly data to be spread out across the entire quarter and it will be in the same frequency as the price data and news headlines.

The target output is a binary classification where 1 represents an increase in stock prices the next day, and -1 otherwise.

The dataset is relatively balanced in nature, with 56% of the data having target output 1 and 44% having target output -1.

2) Exploratory Data Analysis

a) Financial Data

We explored the dataset to check for correlations both between the independent variables and between the independent variables and the dependent variable.

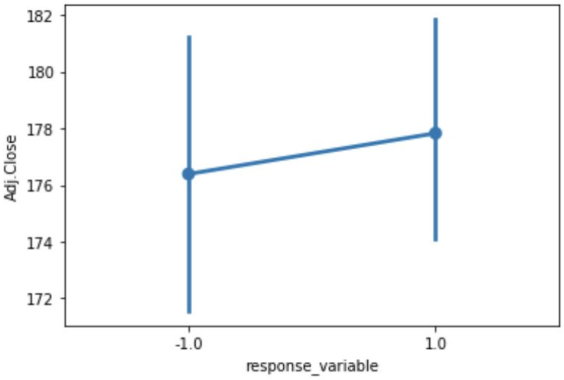
I. Correlations between independent variables

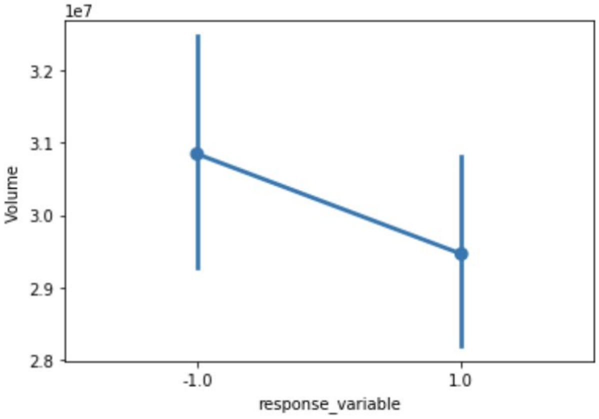
	Open	High	Low	Close	Adj.Close	Volume
Open	1.000000	0.999235	0.998882	0.998100	0.997640	0.016574
High	0.999235	1.000000	0.998667	0.998931	0.998579	0.028357
Low	0.998882	0.998667	1.000000	0.999151	0.998526	-0.007105
Close	0.998100	0.998931	0.999151	1.000000	0.999392	0.004847
Adj.Close	0.997640	0.998579	0.998526	0.999392	1.000000	0.007145

Volume	0.016574	0.028357	-0.007105	0.004847	0.007145	1.000000

There is strong positive correlation between Open, High, Low, Close and Adjusted Close, whereas there is weak correlation between Volume and the other variables.
Therefore, it is not necessary to include all of Open, High, Low, Close and Adjusted Close.

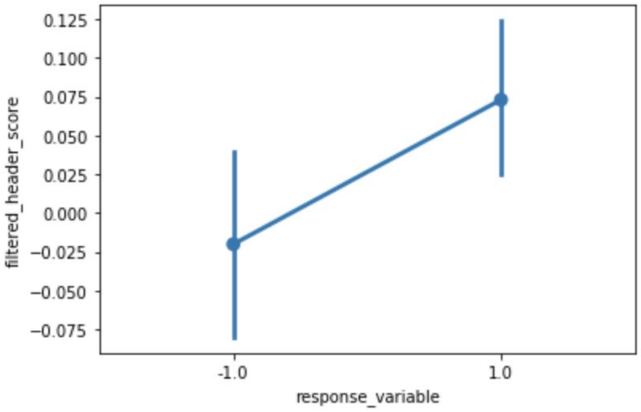
II. Correlations between independent variables and dependent variable





Majority of the independent variables showed a positive correlation with the dependent variable (the plots of the other 4 independent variables are shown in the Annex, all 4 with a positive correlation as well). Whereas, only the volume variable showed a negative correlation with the dependent variable.

b) News data



There is a strong correlation between the sentiment score obtained from the processed news headlines (this is discussed under the section Processing of News Headline) and the target output, where a more positive sentiment score is correlated with a positive direction of movement in stock prices.

3) Feature Selection

We constructed a Random Forest model using only the Financial Data, both price data and quarterly report data, then ran the feature importance function available within the Random Forest model to obtain the feature importance score. The feature importance indicates how useful each feature was in the construction of the Random Forest model. It can be seen that data from the Quarterly Financial Report is significant in predicting AAPL stock price movement, as shown in the table below. We decided to keep only the top 5 features, namely Volume, Changes in Working Capital, Revenue, Dividend paid and Open Price in our model, as these 5 features contribute significantly to the decision making in the Random Forest model. The table shows the feature importances of the financial data.

No.	**Feature**	**Feature Importance**
1.	Volume	0.124921
2.	Changes in Working Capital	0.115194
3.	Revenue	0.105840
4.	Dividend Paid	0.100465
5.	Open	0.095085
6.	Close	0.093599
7.	Adjusted Close	0.092197
8.	Low	0.092090
9.	Net Changes in Cash	0.091881
10.	High	0.088728

Processing of News Headline

As mentioned in the Methodology, we will be exploring both the Bag of Words model and a separate model that uses a combination of Sentiment Analysis and Word Embeddings.

In this section, the three different approaches will be discussed in detail.

1. Bag of Words

Using the news headlines collected, a dictionary of words used was created. In the preprocessing stage for the dictionary, stop words were removed and words that were used only once across all headlines were also removed so as to focus only on commonly used words. The dictionary is then used to create an array of arrays for all headlines. Each array shows which dictionary words have been used and how many times they have been used in each headline. These arrays for each headline will be used in our model.

2. Sentimental Analysis

Sentiments from the news headlines were extracted using the Vader Lexicon Package under the NLTK library. Firstly, the headlines were preprocessed, where stopwords were removed and then lemmatised. Next, the sentiments were extracted, and the raw compounded scores are then obtained. With the raw compounded score, we further classified them into 1 for positive sentiments if the raw score is more than 0.1, -1 for negative sentiments if the raw score is less than -0.1, and 0 for neutral sentiments if otherwise. The classified scores will be used as features for analysis.

3. Word Embeddings

Word Embeddings were constructed using Word2Vec from the Gensim library. In the preprocessing stage, the news headlines were tokenized and made into lower capital letters. Symbols and stop words were removed as well. Using Word2Vec from the Gensim library, the processed headlines were then used to construct a word embedding model. The same processed headlines are used as inputs into the word embedding model to output vectors, which will subsequently be used in the classification models.

Model Selection

In this paper, we will explore the following models: Support-Vector Machine (SVM), Random Forest, K-Nearest Neighbour (KNN), and AdaBoost.

A. Support-Vector Machine (SVM)

SVM is a supervised machine learning algorithm which can be used for classification problems. In the SVM algorithm, each data point is plotted as a point in an N-dimensional space. Then, classification is done by finding a hyperplane in the N-dimensional space that distinctly classifies the data points. To separate the classes of data points in the most optimal way, multiple hyperplanes are taken into account and the hyperplane with the maximum margin, i.e the maximum distance between data points of classes, is identified and chosen by the SVM algorithm. The SVM algorithm was chosen as one of the models to explore as the Bag-of-Words preprocessing step produces dataframes with a large number of features.

B. Random Forest

Random forest consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest outputs a possible class prediction based on the inputs and the class with the highest count becomes the eventual prediction of the random forest. Random Forest was chosen as it works well with a smaller dataset, and is able to handle high dimensional data. It can also output the importance of each feature, which was made use of during feature selection. However, one weakness of decision trees is its inaccuracy on the test set despite good accuracy on the training set as there is a tendency for overfitting. We decided on using Random Forest as one of our models to explore as it allows for bootstrapping which compensates for our lack of data and prevents overfitting which is a huge problem in this project due to our large number of features and small amount of data.

C. K-Nearest Neighbour (KNN)

KNN is a non-parametric and lazy learning algorithm that assumes similar objects exist in close proximity to each other. Non-parametric means there is no assumption for underlying data distribution and a lazy algorithm refers to the idea that it does not need any training data points for model generation. Hence, KNN is a simple model, and it has only one hyperparameter, which makes it easy to optimise if it works well. However, it suffers from the curse of dimensionality, which means the required data grows exponentially as the number of dimensions increases. High dimension also leads to overfitting as well. Hence, KNN might not work well with the Bag of

Words approach, given the high dimensionality of the processed data, and it is more suitable with Word Embeddings and Sentiments Analysis.

D. AdaBoost

AdaBoost is another decision tree but it works differently from Random Forest, using boosting rather than bagging ensemble method. This boosting algorithm combines multiple weak classifiers to form one strong classifier, and each classifier takes into account the error made previously. The weak classifiers are trained on different training data and weights are given to classifiers based on accuracy of the predictions. Hence, it is worth exploring the Adaboost classifier, as it might work better for our data and output better accuracy.

Results & Discussion

In the training of the model, the train_test_split function is used to obtain the training and testing set, with 80% of the data used as the training set and 20% of the data used as the test set. A specific random state is also used to ensure reproducibility of the experiment. Due to the lack of data, we were unable to obtain a validation set, and the training set will be used to tune the hyperparameters instead.

The results of the 4 models (SVM, Random Forest, KNN, AdaBoost) are shown below:

Support Vector Machine:

Inputs	**Hyperparameter**	**Train accuracy**	**Test accuracy**	**AUC**

Quarterly report, filtered sentiments class	random_state=14
	C=100
	Kernel=rbf
	Gamma=0.1
		0.614286	0.67925	0.65137

Quarterly report, filtered sentiments class, word embeddings	random_state=14
	C=100
	Kernel=rbf
	Gamma=0.1
		0.738095	0.58490566	0.5897435

Quarterly report, word embeddings	random_state=14
	C=100
	Kernel=rbf
	Gamma=1
		0.67441	0.54966	0.52210

Filtered sentiments class	random_state=14
	C=0.01
	Kernel=linear
	Gamma=0.001
		0.55952	0.57547	0.5

Word embeddings	random_state=14
	C=100
	Kernel=sigmoid
	Gamma=0.1
		0.52380	0.59433	0.60382

Quarterly report, bag of words	random_state=14
	C=100
	Kernel=sigmoid
	Gamma=0.1
		0.610208	0.541667	0.544

Random Forest:

Inputs	**Hyperparameter**	**Train accuracy**	**Test accuracy**	**AUC**

Quarterly report, filtered sentiments class	random_state=14
	max_features=auto
	n_estimators=200
	max_depth=10
	min_samples_split=10
	min_samples_leaf=2
	bootstrap=False
		0.92142	0.57547	0.55245

Quarterly report, filtered sentiments class, word embeddings	random_state=14
	max_features=log2
	n_estimators=200
	max_depth=10
	min_samples_split=10
	min_samples_leaf=2
	bootstrap=False
		0.92142	0.57547	0.55245

Quarterly report, word embeddings	random_state=14
	max_features=auto
	n_estimators=100
	max_depth=10
	min_samples_split=2
	min_samples_leaf=2
	bootstrap=True
		0.86710	0.56953	0.53805

Filtered sentiments class	random_state=14
	max_features=auto
	n_estimators=100
	max_depth=10
	min_samples_split=2
	min_samples_leaf=1
	bootstrap=True
		0.55952	0.57547	0.5

Word embeddings	random_state=14
	max_features=None
	n_estimators=8
	max_depth=8
	min_samples_split=2
	min_samples_leaf=1
	bootstrap=False
		0.86666	0.556603	0.512550

Quarterly report, bag of words	random_state=14
	max_features=None
	n_estimators=200
	max_depth=100
	min_samples_split=10
	min_samples_leaf=2
	bootstrap=True
		0.99047	0.56603	0.55009

K-Nearest Neighbour:

Inputs	**Hyperparameter**	**Train accuracy**	**Test accuracy**	**AUC**

Quarterly report, filtered sentiments class	random_state=14
	leaf_size=48
	n_neighbor=12
	power=1
	algorithm=kd_tree
	weights=uniform
		0.61190	0.64151	0.63607

Economic news, filtered sentiments class, word embeddings	random_state=14
	leaf_size=49
	n_neighbor=15
	power=2
	algorithm=brute
	weights=uniform
		0.59523	0.65094	0.62677

Quarterly report, embeddings	random_state=14
	leaf_size=1
	n_neighbor=5
	power=1
	algorithm=ball_tree
	weights=uniform
		0.71262	0.56953	0.54144

Filtered sentiments class	random_state=14
	leaf_size=49
	n_neighbor=29
	power=1
	algorithm=auto
	weights=uniform
		0.54286	0.66981	0.64317

Word embeddings	random_state=14
	leaf_size=1
	n_neighbor=27
	power=2
		0.60476	0.57547	0.53205

	algorithm=kd_tree
	weights=uniform			

Quarterly report, bag of words	random_state=11
	leaf_size=4
	n_neighbor=21
	power=1
	algorithm=ball_tree
	weights=uniform
		0.67619	0.50943	0.50943

AdaBoost:

Inputs	**Hyperparameter**	**Train accuracy**	**Test accuracy**	**AUC**

Quarterly report, filtered sentiments class	random_state=10
	n_estimators=1000
	learning_rate=0.001
		0.58809	0.67924	0.59396

Quarterly report, filtered sentiments class, word embeddings	random_state=10
	n_estimators=1000
	learning_rate=0.001
		0.58809	0.67924	0.59396

Quarterly report, word embeddings	random_state=1
	n_estimators=1000
	learning_rate=0.1
		0.69933	0.59602	0.56497

Filtered sentiments class	random_state=10
	n_estimators=500
	learning_rate=0.001
		0.56428	0.58490	0.57424

Word embeddings	random_state=2
	n_estimators=500
	learning_rate=0.1
		0.74761	0.62264	0.59433

Quarterly report, bag of words	random_state=10
	n_estimators=1000
	learning_rate=0.001
		0.63333	0.67924	0.58672

The results affirm the research that both Sentiment Analysis and Word Embeddings can perform much better than the Bag of Words approach. On average, Sentiment Analysis and Word Embeddings produce a higher AUC score, which refers to the area under the ROC Curve. The AUC provides an aggregate measure of performance across classification models and can be interpreted as how well the classification model is capable of distinguishing between classes. Hence, the higher the AUC, the better the model.

Noting the results above, we will be ensembling the top 3 classification models, based on the AUC, using a simple Uniform Blending method in an attempt to obtain an even more accurate result. The classification models chosen are as follows:

1) Support Vector Machine with Quarterly report and Filtered Sentiments class as inputs
2) K-Nearest Neighbour with Filtered Sentiments class as inputs
3) K-Nearest Neighbour with Quarterly report and Filtered Sentiments class as inputs

The ensemble modeling using the top 3 classification models has indeed produced a much better result, which can be seen in the table below.

Test Accuracy	**AUC**
0.70755	0.68470

Conclusion & Further Research

This paper proposes an alternative approach to textual analysis, away from the de facto standard of using the Bag of Words model. By using an ensemble modeling of 3 different classification models, an accuracy score of over 70% was achieved. This is a 40% increase in accuracy score from the Bag of Words model.

At the same time, there is still room to further improve the predictive power of the final model. In particular, we have identified two main areas of improvement. Firstly, the amount of data collected is lacking. In building machine learning models, especially for K-Nearest Neighbor, the curse of dimensionality increases the need for data by an exponential rate when the dimensions increases. Hence, having more data could have allowed the model to better learn and understand the pattern of the dataset, producing a higher accuracy score. Secondly, the scope of the independent variables could be wider. As the focus of this paper was more on textual analysis of news headlines, we only collected news headlines pertaining to AAPL. However, we understand that stock price movements can also be affected by economic news such as unemployment rates and inflation rates. Hence, increasing the scope of analysis, to include economic news, could have produced a higher accuracy score as well.

It is also important to note that one reason why the Bag of words model might have failed in this case is due to the lack of data, which does not perform well with the large number of features that arises from the Bag of Words model.

Nonetheless, in this paper, we have successfully demonstrated the effectiveness of using Sentiment Analysis and Word Embeddings over the Bag of Words model when there is a lack of training data. In the future, further research can be conducted into this space, taking into consideration the areas of improvement listed above.

References

Apple Inc (AAPL) Stock Volatility. (2020, June 1). Retrieved from https://www.netcials.com/stock-volatility-nasdaq/AAPL-Apple-Inc/#fifthlist

Downey, L. (2020, February 5). Efficient Market Hypothesis (EMH). Retrieved from https://www.investopedia.com/terms/e/efficientmarkethypothesis.asp

Gidófalvi , G. (2001, June 15). Retrieved from https://people.kth.se/~gyozo/docs/financial-prediction.pdf

Mohan, S., Mullapudi, S., Sammeta, S., Vijayvergia, P., & Anastasiu, D. C. (2019). Stock Price Prediction Using News Sentiment Analysis. *IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, 205–208. Retrieved from https://ieeexplore.ieee.org/document/8848203

Schumaker, R., & Chen, H. (2006, December). Retrieved from https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1733&context=amcis2006

Goldberg, Y. (2017). *Neural network methods for natural language processing*. San Rafael, CA, United States: Morgan & Claypool Publishers. doi: https://doi.org/10.2200/S00762ED1V01Y201703HLT037

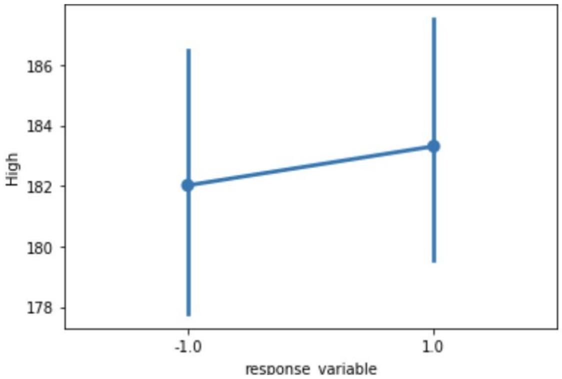
Li, X., Xie, H., Chen, L., Wang, J., & Deng, X. (2014). News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, *69*, 14–23. doi: https://doi-org.libproxy1.nus.edu.sg/10.1016/j.knosys.2014.04.022

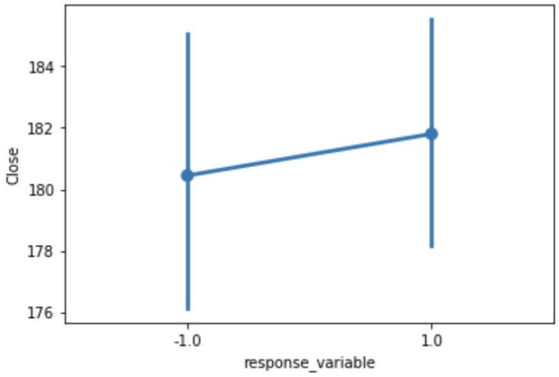
Shah, D., Isah, H., & Zulkernine, F. (n.d.). Retrieved from https://arxiv.org/ftp/arxiv/papers/1812/1812.04199.pdf

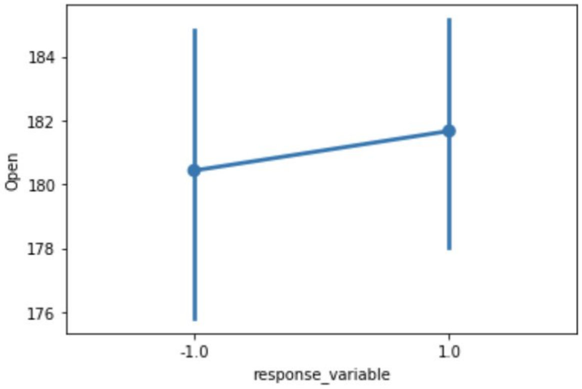
Alanyali, M., Moat, H. & Preis, T. Quantifying the Relationship Between Financial News and the Stock Market. *Sci Rep* 3, 3578 (2013). https://doi.org/10.1038/srep03578

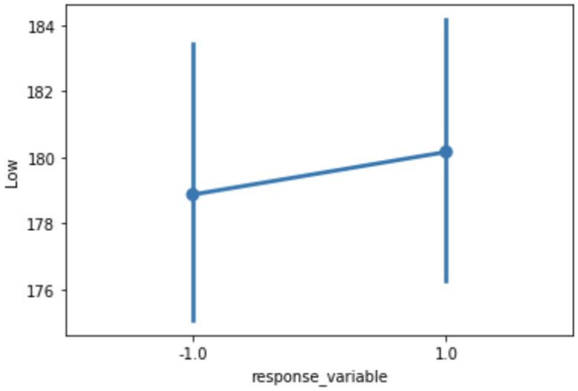
Annex

Plots showing correlation between independent variables (High, Close, Open and Low) and dependent variable:









Inputs	Hyperparameters	Train Accuracy	Test Accuracy	AUC
Filtered sentiments class + word embeddings	Penalty=’l1’ loss=’squared_hinge’ dual=’False’	0.5700935	0.56074766	0.520743
Filtered sentiments score + word embeddings	Penalty=’l1’ loss=’squared_hinge’ dual=’False’	0.5934579	0.5607476	0.5212045
Filtered sentiments class	Penalty=’l1’ loss=’squared_hinge’ dual=’False’	0.563084	0.598130	0.5702127
Filtered sentiments score	Penalty=’l1’ loss=’squared_hinge’ dual=’False’	0.5700934	0.560747	0.525354
Word embeddings	Penalty=’l1’ loss=’squared_hinge’ dual=’False’	0.5864485	0.3925233	0.5
Bag of Words	Penalty=’l1’ loss=’squared_hinge’ dual=’False’	0.5452436	0.5	0.520

Inputs	Hyperparameters	Train Accuracy	Test Accuracy	AUC
Filtered sentiments class + word embeddings	n_estimators=6, Bootstrap=’True’,max_depth=3	0.6728971	0.5607476	0.538596
Filtered sentiments score + word embeddings	n_estimators=9, Bootstrap=’True’,max_depth=4	0.661214953	0.6448598	0.616161
Filtered sentiments class	n_estimators=10, Bootstrap=’True’,max_depth=4	0.67056074	0.6074766	0.5707070

Filtered sentiments score	n_estimators=12, Bootstrap='True',max_depth=4	0.656542	0.644859	0.598168
Word embeddings	n_estimators=2, Bootstrap='True',max_depth=8	0.6939252	0.59813084	0.5891941
Bag of Words	n_estimators=12, Bootstrap='True',max_depth=6	0.7780373	0.5981308	0.5512820

Inputs	Hyperparameters	Train Accuracy	Test Accuracy	AUC
Filtered sentiments class + word embeddings	k= 49	0.5887850	0.5607476	0.562893
Filtered sentiments score + word embeddings	k=46	0.588785	0.5607476	0.553333
Filtered sentiments class	k=24	0.5630841	0.5046728	0.497113
Filtered sentiments score	k=35	0.59813084	0.588785	0.5651154
Word embeddings	k=42	0.5654205	0.579439	0.5695970
Bag of Words	k=25	0.5730858	0.520834	0.528

Inputs	Hyperparameters	Train Accuracy	Test Accuracy	AUC

Filtered sentiments class + word embeddings	n_estimators=116
	learning_rate=0.1
		0.707943	0.5981308	0.561046

Filtered sentiments	n_estimators=33	0.633177	0.588785	0.5309523

score + word embeddings	learning_rate=0.1			

Filtered sentiments class	n_estimators=30
	learning_rate=0.1
		0.6004672	0.5794392	0.546897

Filtered sentiments score	n_estimators=112
	learning_rate=0.1
		0.654205	0.504672	0.528852

Word embeddings	n_estimators=55
	learning_rate=0.1
		0.65654205	0.5887850	0.52252

Bag of Words	n_estimators=40
	learning_rate=0.1
		0.70533642	0.5416667	0.541