CS 6210: Natural Language Processing
Final Project

Predicting Stock Market Movements with News Headlines

By
Thinh Lam
Thai Khanh Huynh
Siddhartha Pant

Introduction:

Predicting movements in the stock market is one of the most challenging and widely studied tasks in computational finance. Investors, researchers, and investment management professionals are always looking for possible 'signals' that could give them additional insight into market performance, which can in turn translate into investment returns. In this project, we use various techniques from Natural Language Processing to assess if daily breaking news headlines can be used to reasonably predict movements in the stock market for the specific trading day.

For the purposes of this project, the breaking news headlines are aggregated from a channel on Reddit, a prominent social news and discussion website. In addition, the market is represented by the Dow Jones Industrial Average Index (DJI). DJI tracks 30 large publicly owned blue-chip companies trading on the New York Stock Exchange (NYSE) and the NASDAQ. DJI is widely accepted as an appropriate proxy to represent the performance of the broader market.

In this project, predicting movements in the stock market is considered a binary classification task. More specifically, we aim to predict if the market moved up/stayed the same or moved down for a given trading day.

Data Collection and Pre-processing:

The dataset for the project is obtained from Kaggle. The data set can be accessed through the following link:
Daily News for Stock Market Prediction (https://www.kaggle.com/aaron7sun/stocknews)

The dataset consists of historical news headlines from Reddit WorldNews Channel (/r/worldnews) from August 8, 2008 - July 1, 2017. The headlines are ranked by Reddit users' votes, and top 25 headlines are considered for a single date. For each date, there is a corresponding label column with binary data that indicates whether DJI moved up/stayed (encoded with 1) the same or went down that day (encoded with 0). The data set has 1,989 observations spanning 27 columns (1 column for the date, 25 columns for the headlines, and 1 column for the label depicting market movements). The data set is reasonably balanced between the two labels: 1,065 observations pertain to trading days when the market moved up/stayed the same, and 924 observations refer to trading days with negative returns.

The data set was already reasonably well processed and did not require any major interventions for data wrangling. We applied standard data pre-processing techniques such as removing regular expressions that are not alpha-numeric and stripping extra spaces. All corresponding headline records started with 'b ', followed by the actual headline. As 'b ' was not useful to our analysis, we removed it from every record. We then applied lemmatization on the data set, as it would allow us to group words with the same inflected forms and

analyze them as a single item. We also removed stop words from the data set using the NLTK stop word corpus.

We created two distinct forms of the processed data set for our analysis: a data frame consisting of the top 25 headlines in separate columns, and a data frame that combines the 25 headlines into a single corpus column. In our analysis, we chose the form that was best suited to be used in implementing specific models.

Project Scope

In this project, we have identified three broad problem statements:

1. Predictive Modeling: Given a set of 25 top daily headlines for a specific trading day, can we predict the direction of the movement of the market for that day?

   We used a wide range of traditional Supervised Machine Learning Techniques and Deep Learning Techniques using Word2Vec Embeddings to build various predictive models and analyze their respective performances.

   In addition, we approached predictive modeling from the lens of sentiment intensity analysis. Each of the headlines were assigned sentiment scores and an overall sentiment score for the day was calculated. We then built a model to predict the direction of the movement of the market based on the overall sentiment score for a specific date.

   Moreover, we also built a predictive model based on the conclusions from topic modeling. We clustered each of the headlines into 100 distinct topics. We built a predictive model to predict the direction of the movement of the market based on the assigned topics of each of the headlines.

2. Topic Modelling: In order to get a better understanding of the news captured by the data set, we built a topic model using Latent Dirichlet Allocation (LDA). We then attempted to study the patterns in the top 8 topics and classify them into specific themes.

3. Model Explainability: We attempted to get additional visibility and insights into our supervised machine learning based predictive models to clearly understand how the models were approaching the classification task. We used the Local Interpretable Model-Agnostic Explanations (LIME) framework to perform this task. As LIME is a fairly recent open-source package, its implementation at the moment is limited to a select list of traditional machine learning models.

Predictive Modeling using Supervised Machine Learning Techniques

For this part of the project, we considered the data set with the combined corpus and converted them into n-gram representations. We considered unigram, bigram, trigram, four gram, and fivegram representations. The average length of a headline in our data set after data processing was 5. As a result, fivegram seemed to be a fitting upper limit to cap the n-gram representations.

We then created a model pipeline with four supervised machine learning models: Logistic Regression, Support Vector Machine (SVM) Classification, Multinomial Naïve Bayes Classification and Bernoulli Naïve Bayes Classification model. These models are widely used supervised machine learning techniques to create text classification models [8].

As text data, in general and in particular this case, are highly unstructured, it is difficult to guesstimate the structure of the data. We attempted to use a model that achieves classification based on a sigmoid function (Logistic Regression) and a linear kernel (SVM classification with linear kernel). We also considered the use of Naïve Bayes Classification as in a considerable number of instances the features in the data set can be considered to be independent. However, it was difficult to ascertain whether frequency or probability of occurrences of word was a better factor to consider in the naive bayes classification implementation. As a result, we considered to use both the Multinomial and Bernouli Naïve Bayes Classification models.

We then ran a five-fold grid search cross validation in each of the five representations of the n-gram model to identify the best model for unigram, bigram, trigram, four gram, and fivegram representations. As the data set was relatively balanced between positive and negative labels, we considered accuracy as the scoring metric in the grid search. In the grid search for Logistic Regression and SVM classifier, we were particularly interested in optimizing the regularization parameters associated with the model. We varied C (the regularization parameter) between the following values: 1, 10, 100, 1000, and 10,000. For the Multinomial and Naive Bayes Classifier, we were interested in optimizing the Laplace smoothing parameter. We ran the grid search on a range of alpha values between 0.05 and 1.

We then tested the model evaluation metrics of each of the selected models with the test set.

The results of the Grid Search Cross Validation were as follows:

n-gram	Optimal Model	Optimal Parameter	Confusion Matrix	Accuracy	Precision	Recall	f1
Unigram	Multinomial Naive Bayes	Alpha = 0.72	[[0 180] [1 217]]	0.545	0.547	0.995	0.706
Bigram	Logistic Regression	C = 1	[[3 177] [1 217]]	0.553	0.551	0.995	0.709
Trigram	Multinomial Naive Bayes	Alpha = 0.74	[[21 159] [16 202]]	0.560	0.559	0.927	0.697
Fourgram	SVM Classifier	C = 10	[[42 138] [39 179]]	0.555	0.565	0.821	0.669
Five-gram	Multinomial Naive Bayes	Alpha = 0.42	[[21 159] [24 194]]	0.540	0.549	0.890	0.680

Overall, predicting the direction of the movement in the market using top 25 breaking news headlines by applying Supervised Machine Learning techniques did not fare us much better than a random guess, as our accuracy metric in all cases was just very slightly over 0.5.

In general, across the board, the models seem to have a natural tendency to classify observations towards a positive label. To that end, the models seem to have unreasonably high recall scores, given their predictive capabilities. We noticed that the unigram and bigram based models have the tendency to pull the True Negative and False Negative predictions to values that are very close to zero.

In comparative terms, the best overall model was given by the four-gram SVM Classifier with Linear Kernel. The precision and recall parameters for this model were comparatively more balanced, and the accuracy of the model (0.555) was very close to the highest accuracy value obtained by the analysis (0.566).

Model explainability using LIME:

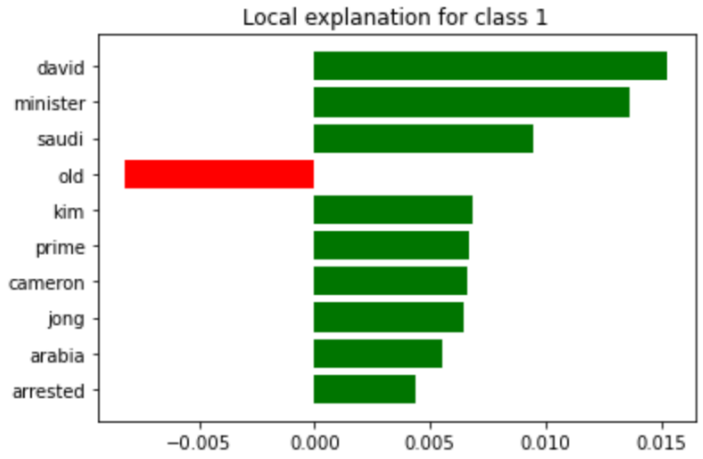
In this section we used the LIME framework to explore deeper insights into our supervised learning methods. The results of the grid search validation indicated that the four-gram model with SVM classification with the linear kernel was the best predictive model. However, grid search is a highly computationally intensive algorithm, and it may be difficult for the users of the model to see exactly what factors in the data are driving the predictions.

Prediction visualizations in ML are usually provided using a decision tree which can be very useful when some of the features are numerical. However, since our features in this case are categorical, the decision tree would be unreasonably big. Using the LIME framework, we

were able to get more context on the underpinnings of the implementation model. LIME is an open-source package that is geared at enhancing the explainability of machine learning models. LIME takes in a model's probability predictions and provides prediction reasoning[5].

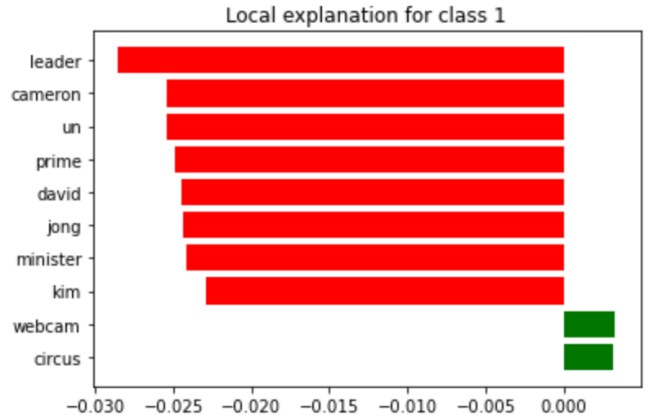
The figure below shows the prediction of the four-gram model on an arbitrary observation on the dataset. From the figure below, it is clear that the model predicted a positive market movement for the specific day. However, LIME also provides insights on the words that contribute to that prediction, along with a scaled magnitude and direction of the contributions.

```
%matplotlib inline
fig3 = exp3.as_pyplot_figure()
```



The following figure depicts a case when the model predicts a negative market movement for an observation. Again, the key words impacting this prediction along with their scaled magnitude and directions can be observed.

```
fig4 = exp4.as_pyplot_figure()
```



We can also see that some of the words driving the predictions are similar word tokens. However, in different contexts/sections of the sentence, a word can have a negative or a positive weight.

We analyzed the results from the LIME framework on several similar arbitrary observations and noticed that the predictive models constructed using the news headlines were not consistent in the way in which they made classification decisions using similar tokens of words. This is indicative of the fact that the predictive models were not very stable.

Predictive Modeling using Word2Vec Embedding and CNN

In this section, we implemented Word2Vec embedding to represent the news headlines and conducted predictive modeling using Convolutional Neural Networks (CNN) . The main motivation behind using Word2Vec embedding was to study the impact of correlations between words that have otherwise been ignored in one-hot encoding methods, or other typical techniques involving n-gram representations.

CNN has widespread applications in detecting special patterns in the data [4]. By changing the size of kernels and concatenating the outputs, we were hoping to detect patterns in the different groups of adjacent words in the data set. As the headlines have different lengths, it was difficult to pass them as arguments in the input layer of the Convolutional Neural Network (CNN). Therefore, we created a matrix with a vocabulary size of 300 dimensions. We also only considered words that appear at least two times in the data set. We then used the Tokenizer function in Keras to convert each top news as a sequence of indices of word vectors. After that, for each word, we added its corresponding vector in the Embedding layer before we started training the CNN model.

In our model, we used 128 filters in the hidden layers with the size of 3 words. We used the Rectified Linear Unit (ReLU) activation function in the model, with the exception of the output layer, that used a sigmoid activation function. Finally, we compiled the model using the binary cross entropy loss function, “Adam” optimizer and accuracy as the scoring metric. [L3]

Our accuracy with the implementation of Word2Vec and CNN was 0.45. Although the complexity of the model was increased, the resulting output was significantly lower than the evaluation metrics achieved through simpler Supervised Machine Learning models.

Predictive Modeling using Sentiment Intensity Analysis

In this section, we attempted to study if the overall sentiment of the news can be used to predict the direction of movements in the market. We performed sentiment analysis on each of the headlines, and assigned a corresponding sentiment score. These sentiment scores were used to calculate an overall sentiment score for the day, which was in turn used to predict the direction of movement in the market.

We used VADER (Valence Aware Dictionary and sEntiment Reasoner) and Sentiment Intensity Analyzer libraries in Python to perform sentiment analysis. The library was used to determine the overall positivity or negativity of each news, which seeks to find the cosine similarity between each word with “positive”, “negative”, “and ‘neutral” and assigns a score between 0 and 1. A low score indicates a negative sentiment, and a high score indicates a positive sentiment. Notably, we found out that Sentiment Intensity Analyzer is more likely to assign a negative sentiment to a particular text. In order to account for this, we adjusted for the bias by a factor of 0.13. We arrived at this number using several trial and error methods. We then calculated the overall sentiment score across all the headlines for the day and compared the overall sentiment score of the news to that of the market.

Upon comparing the original Labels and Predicted outputs, we computed all the model evaluation metrics to determine the efficiency of our analysis. The model evaluation metrics indicated an accuracy of 0.51, precision of 0.54, recall of 0.54 and f1 of 0.54.

In this analysis, we also found that it is challenging to determine whether a given top news is positive or negative, thus the overall positivity or negativity of each day based on those news. This analysis also pointed out that without the correlations between adjacent words, it will be challenging to determine whether the given news is positive or negative.

Topic Modelling

One of our problem statements was to understand the type of information contained in the headlines of the news data. We wanted to get a better understanding of some of themes that are captured in the news headlines that are serving as the predictors in our model. To that end, in this section of the project, we performed Topic Modelling using Latent Dirichlet Allocation (LDA)[3].

LDA is a powerful generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar[7]. LDA assumes that the distribution of topics in a document and the distribution of words in topics are both Dirichlet distributions. LDA is an Unsupervised Machine Learning modeling technique and is widely used in Topic Modeling[7].

For this task, we considered the data set in the form of a data frame consisting of the top 25 headlines in separate columns. In order to effectively model the topics of each of the

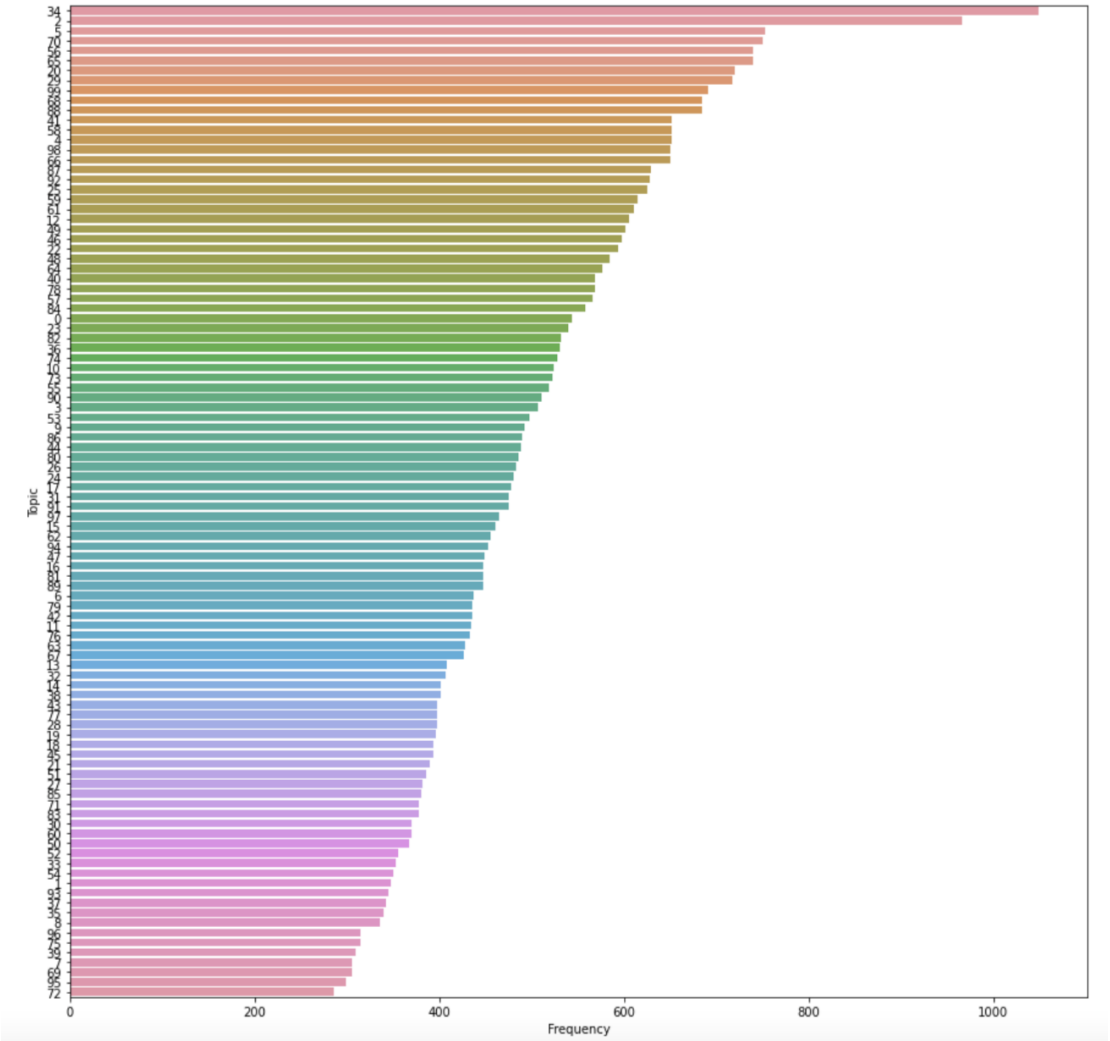
headlines, it was important for each of these headlines to be a distinct observation in the data set. To that end, we pivoted the data in all of these 25 columns into distinct rows and stored it in a separate data frame. The resulting data frame had a single column with each row representing a distinct headline in the data set. The data frame had 49,725 rows.

One of the major challenges of topic modeling is to eventually end up with outputs that can be attributed to specific themes and topics. In other words, topic modeling tasks are prone to providing output with topics that have significant overlaps, which affects their interpretability. As most of the news headlines tend to focus on broadly related yet niche themes, avoid overlaps between topics was a very important consideration for us while pursuing the analysis.

To implement LDA, we used a TF-IDF vectorizer to transform the data frame. To eliminate words that have overlaps across all the documents, we only considered words that appear in a maximum of 75% of the documents. Furthermore, to eliminate words that are very niche and do not provide insights into the topic model, we only chose words that appear at least 5 times. We also capped the maximum features of the TF-IDF matrix to 30,000 to consider only the most important features of the data.

We then fit the tf-idf matrix to the LDA model. We chose to model the data set over 100 distinct topics. LDA has two major hyperparameters: alpha and eta. A higher alpha result in the text being represented by more topics, and a higher eta results in a topic being represented by more words. In both of these scenarios, higher alpha and eta values lead to greater overlaps in the topics outputted by the LDA. As a result, we assigned the model alpha and eta values of 0.025 (a fourth lower than the default value used by Python for a total of 100 topic components).

After fitting the LDA model, we were able to assign a corresponding topic to each of the headlines. We then analyzed the frequency of the occurrences of each of the topics in the dataset. The plot below captures this information:



It can be seen that Top 8 topics roughly capture 25% of the data. The topics that are most widely represented in the data are as follows:

Topic Number	Number of Headlines
34	1,049
2	966
5	753
70	750
56	740
65	739
20	720
29	717

We then tried to investigate the Top 10 words in each of these ten topics to understand the themes represented by these topics. The results and the inferences from the analysis are captured in the table below:

Topic Number	Top 10 Words	Theme
34	['copyright', 'access', 'web', 'data', 'google', 'facebook', 'block', 'twitter', 'user', 'internet']	Social Media and Technology
2	['secret', 'document', 'reveal', 'ecuador', 'cable', 'founder', 'cia', 'julian', 'assange', 'wikileaks']	Wikileaks
5	['surprise', 'nuclear', 'airline', 'il', 'jong', 'kim', 'south', 'korean', 'north', 'korea']	Nuclear Politics and North Korea
70	['strike', 'killed', 'mosque', 'drone', 'pakistan', 'yemen', 'kill', 'blast', 'bomber', 'suicide']	War and Terrorism
56	['indonesia', 'japan', 'hit', 'alert', 'new', 'flu', 'tsunami', 'magnitude', 'quake', 'earthquake']	Natural Disaster
65	['asylum', 'deep', 'colombia', 'plastic', 'trapped', 'miner', 'nsa', 'whistleblower', 'edward', 'snowden']	NSA Leaks and Snowden
20	['party', 'picture', 'riot', 'year', 'bay', 'house', 'sentenced', 'white', 'pirate', 'prison']	Crime
29	['corrupt', 'scotland', 'independence', 'crimea', 'russia', 'russian', 'violent', 'fighter', 'jet', 'georgia']	Global Politics

The key takeaway from this analysis is that most of the topics covered by general news headlines are not very relevant to a standalone investor making decisions about their investment holdings in the market. Much to our surprise, economy was not a major theme in the headlines present in the dataset. To ensure that we did not incorrectly miss out this theme by selecting only the Top 8 topics, we ran the analysis again for Top 12 topics, and yet economy was not a relevant theme in the analysis.

As a result, the topic modeling analysis validates our findings from the previous models that breaking news headlines, particularly if randomly sourced, may not be a good predictor for the direction of the movement in the stock market.

Predictive Modeling using Topic models

In this section, we attempted to build a predictive model to predict the direction of movement in the market, based on the topic of the associated headlines. After all headlines were grouped into 100 topics using the LDA analysis, we assigned a topic score to each of the headlines. We then treated each topic value as a category and dummy encoded it, for further predictive modeling using a sequential model.

As the resulting dummy encoded data consisted of 2500 (25 headlines X 100 topics) columns, we used search space between the width as min_value and 5000 as max with step as 2500. We considered the appropriate amount of hidden layers in the Neural Network to be between 2 and 4.

With the defined search space, and fed data, we found the best hyperparameter to be 2 layers, 5000 nodes on the first and 2500 on the second upon hyper parameter tuning.
During model fitting, we found that our training accuracy quickly increased from around 50% to near 100% while validation accuracy remained around 50%. This indicated overfitting in the model. Upon evaluating the model with the test set, we were able to further discover overfitting as the model's accuracy barely hit 0.50 with the test set.

This conclusion prompted us to see if certain features had a stronger correlation with labels in the training set. We took a step back and examined dependencies between each top headline column with the label using chi-square test via chi2_contingency from scipy.stats.
Using the standard significance value of 0.05 and 0.1, we found, in this specific case, only column Top25 and label are not independent and Top23 has some relationship with the label. The rest is completely independent of the label. We then extracted the features with dependencies, and fit these features to another model. We fit the abridged data set to the same model as before, just with less hidden layer search space.

The model yielded an accuracy of 0.54. With this modification to the model, we were able to alleviate overfitting. However, the predictive performance of this model isn't better than the evaluation metrics achieved through simpler Supervised Machine Learning models.

Key Takeaways:

Based on our analysis using various techniques, we observed that breaking news headlines that are sourced from general sources is not a very good signal to be used as a predictor of movements in the broader stock market. One important reason for that is the fact that most breaking news headlines have negative sentiments attached to it, whereas market movements are not always affected by negative news. In fact, in most years, markets have yielded positive returns and have therefore been able to attract investments.

In addition, the themes and topics captured by most breaking news headlines are not necessarily important to a standalone investor. Typically, a stand lone investor has a focused interest in the performance of their individual holdings. As a result, in most cases, their decision to buy or sell in the market are unaffected by themes like political controversies, natural disasters, crime, and international relations.

In a considerable number of instances, the same news can have differing impacts on different parties. For example, news that may be good for some companies in the market may be unfavorable for others. As a result, the same news can create conflicting impacts to various players in the same market. This could be another reason why headlines were not good predictors of the direction of the movement. Maybe the model evaluation metrics would have been better if we were looking to evaluate the impact of headlines on the price movement of a specific company or a sector, as opposed to the broader market.

Limitations:

One of the most challenging aspects of this project was to isolate the impact of individual headlines to our predictive models. When we combined the corpus, we had to take fragments of text that overlapped between multiple headlines and didn't always make coherent sense. However, treating each headline as its own feature was not possible, because it would be unfair to assign the label pertaining to market movement to a specific headline.

Furthermore, our models all assume that the impact of all of the top 25 headlines are roughly equivalent, whereas in reality some news headlines dominate market sentiment significantly more than others. We were unable to identify a prudent computational mechanism where we could assign weights to the specific headlines.

Finally, the data set used in this analysis selects top 25 headlines based on user ratings on a social news and discussion website. As a result, the selection of headlines in the dataset in and of itself comes with various sets of biases including the demographics and preferences of the members of the discussion channel. The headlines in the data set may not always be in alignment with news that are important in determining the sentiments of the financial market and the economy.

Statement of Contributions:

Team Member	Contributions
Sidd Pant	Sidd worked on data pre-processing, building predictive models using Supervised Machine Learning and developing a Topic Modelling framework using Latent Dirichlet Allocation.
Thai Hyunh	Thai worked on data pre-processing, created Word2vec embedding for CNN model, model explainability using LIME, and build the predictive models using the output from the topic model
Thinh Lam	Thinh worked on data pre-processing, and building predictive CNN models using Word2Vec Embeddings and Sentiment Intensity Analysis and model interpretability using LIME.

References

1. Omernick M., Chollet F.. Text sentiment classification starting from raw text files. Pulished May17 2020. www.keras.io.

2. Tian L., Lai ., Moore D. J.. Polarity and Intensity: The Two Aspects of Sentiment Analysis. Proceedings of the First Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML), pages 40–47, Melbourne, Australia July 20, 2019

3. Amin Z. M., Nadeem N. Convolutional Neural Network: Text Classification Model for Open Domain Question Answering System. Department of Computer Science and Engineering, 1,2University of Engineering and Technology, Lahore, Pakistan.

4. Jang B. , Kim M., Harerimana G., Kang S. and Kim W. J..Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism. Appl. Sci. 2020, 10, 5841

5. Ribeiro T. M., Singh S., Guestrin C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

6. Zhang W., Yoshida T. , Tang X.. Text classification based on multi-word with support vector machines. Published Elsevier. Knowledge-Based Systems 21 (2008) 879–886

7. Song Y..Pan S.Liu S., Zhou X. M., Qian W. Topic and Keyword Re-ranking for LDA-based Topic Modeling. Published CIKM’09, November 2–6, 2009, Hong Kong, China

8. Singh G., Kumar B., Gaur L., Tyagi A.Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification. 2019 International Conference on Automation, Computational and Technology Management (ICACTM)