Social Media and Investment Data in Predicting Financial Themes

Course project for CSE 6240: Web Search and Text Mining, Spring 2021

Sami Belhareth[**]
sbelhareth3@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Reagan Kan[*†]
rkan3@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Atticus Ignelzi[*]
aignelzi3@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Abstract

The objective of our study is to predict the daily price movement of publicly traded company shares using posts and tweets from Reddit and Twitter. We used 51,792 tweets from 45,372 users and 1,031 Reddit "posts" within the period of 7 January 2021 and 7 February 2021. We focus on 8 companies, 4 that had significant volatility in the time frame — GameStop, BlackBerry, Nokia, AMC — and 4 that are among the largest companies on the US stock market — Apple, Microsoft, Google, Amazon. The price movement is discretized into a binary variable, 1 for positive price movement, and 0 otherwise. This label distribution is skewed to be roughly 67% 0's and 33% 1's. In this study, we train and evaluate a novel classifier, which uses pretrained GloVe embeddings with an LSTM network, and incorporates graph centrality metrics. [Add one sentence describing the performance of the model.]

ACM Reference Format:
Sami Belhareth, Reagan Kan, and Atticus Ignelzi. 2021. Social Media and Investment Data in Predicting Financial Themes: Course project for CSE 6240: Web Search and Text Mining, Spring 2021. In *Proceedings of CS 6240 (Spring'21).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

1.1 Aim

The objective of this study is to successfully predict the price movement of publicly traded company shares through Reddit and Twitter data. In order to do this, first, we must perform comprehensive web scraping for Reddit and Twitter posts that mention certain stock tickers, and obtain data on the users who wrote the posts. Then we must successfully make use of natural language processing to create word embeddings for use in a deep neural network. Lastly, in order to improve on the baseline algorithm, graph data of social media posts (centrality measures, number of likes/upvotes) and shares in publicly traded companies in order to determine the aforementioned variables.

1.2 Challenges

There may be accuracy issues with natural language processing, given the complexity in trying to not only determine user sentiment [list challenges here]

1.3 Impact

On January 28th, 2021, Robinhood and a number of other stock brokers restricted trading of GameStop, stock that increased drastically in the preceding two weeks. This drew media attention towards Reddit retail traders, and media coverage declared these retail traders as responsible for the dramatic increase in prices of the shares of other companies, including BlackBerry, AMC, and Nokia. This work can help elucidate whether, in fact, retail traders and individuals active on social media and online forums indeed have an effect on market trends, and whether Reddit, in particular, is of interest to those wishing to predict market trends.

2 Literature Survey

The paper by Wang et al. relates to our work because it involves the use of both financial indicators and social media data to predict short-term stock price changes [7]. The authors first extracted words from comments describing stocks. The number of positive and negative words were used to create a sentiment discrepancy index (SDI) that was used in conjunction with financial indicators. PCA analysis was used to build a linear predictive model. A shortcoming is that there was limited use of sentiment data. Financial indicators were weighted heavily in the linear model. There was a Pearson correlation coefficient of 0.79 between predicted and actual stock price when incorporating SDI, versus 0.56 when not incorporating SDI. Our critique is that this paper did not describe validation splits, thus their results may be a product of overfitting.

[*]All authors contributed equally to this research.

Spring'21, Georgia Tech, Atlanta, GA, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

The paper by Ranawat et al. relates to our work since its model uses tweets to predict the direction of change in Apple's stock price [5]. The authors trained a LSTM ANN (SGD optimizer and Binary Cross-entropy loss function) with a fine tuned GloVe embedding layer. A shortcoming of the model is that it can only predict the direction of change, up or down, with 76% accuracy. The hyperparameter search was restricted to varying 1 hyperparameter at a time. Our critique is that the accuracy metric can give an overconfident view of the model performance.

Work by Mittal et al. relates to our work because it presents a method for predicting the stock market trends [4]. The authors train various ML models using sentiment features extracted from Tweets and DJIA values. Their models were evaluated using a 5-fold sequential validation scheme: train on the first 3 days, evaluate on the next 5 days. They find "Calm" and "Happy" sentiments to be most indicative. A shortcoming of the overall model is that it is not an end-to-end pipeline, as it uses hand-engineered static correlation rules for sentiment analysis. Our critique is that the size of the testing dataset is not disclosed. This ambiguity makes it hard to assess the generalization error of the model.

3 Data Set Description

3.1 Data Preparation

The Yahoo! Finance website was used to download CSVs containing market data for the securities of interest. We used this tool to retrieve the date, open and close prices for each of our eight target stocks. (High, low, adjusted close, volume were also pulled but were not used.) This data is used to construct our ground truth classification labels. Specifically, we computed the percent change of the price ((close - open) / open) for each date in our month long period: January 7, 2021, to February 7, 2021. There is a slight caveat in the date range. We include an extra day at the end, February 8, 2021, Because our prediction is set up as 1-ahead classification, i.e. given data on a particular date, predict the stock price movement of the next trading day. We frame our problem as a binary classification task, in other words, dates with a positive percent change are labeled 1 and dates with negative or zero percent change are labeled 0.

Web scraping was used to obtain data from Twitter over the specified range, using code from an open-source repository [6]. A web scraping application was also used to scrape the top Reddit posts in specific subreddits over the same time period. The code for the application was found on an open-source repository [3].

After web scraping, we were left with a lot of irrelevant tweets and posts, so we filtered for English texts that mention our eight target stocks, either by the company name or stock ticker symbol. We used the Regular Expression functionality built in Pandas DataFrame to match for these keywords and to ensure that the keywords were indeed words by themselves. For example, the Blackberry Corporation is among our list of companies. Its ticker symbol is "BB". Blindly keeping tweets containing the substring "BB" can yield false matches such as "BUBBLEGUM".

The next step is to remove noisy words from the texts. Words like "the", "this", "at" convey little semantic meaning and, if kept in the tweets, would only provide noisy signals to the classifier. Thus, we remove these so-called stopwords from our tweets. We utilize Python Natural Language Toolkit (NLTK) for this step. Specifically, we tokenize the strings by calling NLTK's tokenizers and filter out words that are included in NLTK's set of 179 stopwords. The remaining words are then sent downstream to our baseline classifiers, which take care of vectorization as their first step.

3.2 Raw Data Statistics

Within the month-long period, we scraped a total of 60,145,444 tweets. Of those, there were 6,713,066 tweets in English. Then, we filtered for tweets containing keywords (company name and ticker name, e.g. Microsoft → microsoft, msft) associated with the eight stocks. Specifically, we converted all English tweets to lowercase and kept those that contained any of the keywords. Our filtering process yielded 51,792 tweets. Before removing stopwords, on average each tweet had 23.719 words, 1.922 sentences, and 13.670 words per sentence. After removing stopwords, on average each tweet has 14.371 words and 8.374 words per sentence. The average sentence count did not change. Those tweets were generated from 45,372 unique twitter accounts and were a part of 39,071 unique conversations.

For our month long period, we also scraped the top 1000 posts from the most popular investing subreddits, r/stocks, r/investing, r/wallstreetbets, and r/securityanalysis. We concatenate the post content to the post title to form the final "post". These 4000 posts were filtered down to 1031 posts using the same filtering procedure that was applied to the Twitter data. Before removing stopwords, on average each tweet had 242.185 words, 12.238 sentences, and 17.774 words per sentence. After removing stopwords, on average each tweet has 135.372 words and 9.53 words per sentence. Once again, the average sentence count did not change.

Combining the Reddit and Twitter data, we have a total of 52,823 data points. However, even though our date range spans an entire month, it contains only 21 trading days. So before we split our data into 70/10/20 train/validation/testing splits, we dropped the data points that fell on weekends and holidays. This resulted in 26,635 training, 3,805, and 7610 testing data points. Each data point is coupled with a binary label to form the final classification data pair. The labels are matched to data points according to the date the post/tweet was posted and the stock that it mentions.

3.3 Data Analysis

This section will discuss ground truth label distribution and analysis of sentiment and graph features. First, we note

that there is a class-imbalance, around 67% 0's and 33% 1's. This is where the first insight is drawn. The natural dataset bias towards label 0 means that metrics like accuracy and Area-Under-the-Curve (AUC) of the Receiver-Operating-Characteristic curve will provide an overly optimistic view of the classifier's performance (simply guessing label 0 results in 67% accuracy). Instead we will evaluate our baseline models using F1-score, Precision, and Recall.

Next, we examine the proposed sentiment and graph features. For the sentiment features we look to VADER (Valence Aware Dictionary for sEntiment Reasoning), a rule-based lexicon for sentiment analysis. It is useful for our dataset since it considers abbreviations, emoticons, and slang commonly found in social media [2]. We leverage SentimentIntensityAnalyzer, a VADER implementation in NTLK, to obtain 4 scores, Positive, Neutral, Negative, and Compound, for sentiment features. Believing that graphs and networks contain valuable information, we examine the use of graph features as predictive features in our classification pipeline. We define the graph as a set of nodes and edges. There are 2 node types, texts (tweets or posts) and users, and 2 edge types, text-to-text and text-to-user. Text-to-text edges are formed between texts in the same conversation or reply chain. Text-to-user edges connect user nodes with the nodes representing the tweets or posts made by that user. Then, the graph features for any text is the centrality and importance scores for the corresponding node in the graph. We specifically look at node degree, PageRank & Hyperlink-Induced Hyper Search scores, and Betweenness, Eigenvector, and Node degree centrality. There are 14 features since each tweet can inherit from two nodes in the graph, the text node or the user node. Since this report focuses on baselines, which do not use graph features, we only analyze the graph features for the bigger Twitter dataset. We believe that our general findings can be applied to the Reddit data due to the similar behavior of social media users. From Twitter data, we built a graph with 96,936 nodes and 359,556 edges. We use Python NetworkX for graph construction and feature computation.

With features defined, we discuss the analysis findings from visualizing 2D/3D Feature Subspaces. We created 2D scatter plots with points being tweets and the x,y axes being sentiment and graph features, respectively. Each point is colored based on their truth label. For each of the 8 stocks, we created 56 plots, since there are 4 sentiment and 14 graph features. None of the 2D plots show linearly separable clusters. However, when we add a third axis using another graph feature, separable clusters form in the 3D space. We include plots from the AAPL and GME stocks as examples. See Figure 1 for more details. Other feature combinations have similar structures and patterns.

Our examination of these plots leads to our second and third insights. First of all, looking at only 3 raw features, we can see separable clusters. This is the second insight; we have evidence that sentiment and graph features have predictive ability and should be included in our experimental classifiers. Secondly, the separation between the observed clusters is not always clean and can have non-linear overlap. This is the third insight; the non-linear behavior suggests that our models should be non-linear. This motivates our use of artificial neural networks with non-linear activation functions for Baseline 2 and our experimental classifier. To compare linear and non-linear models, Baseline 1 is an SVM with a linear kernel.

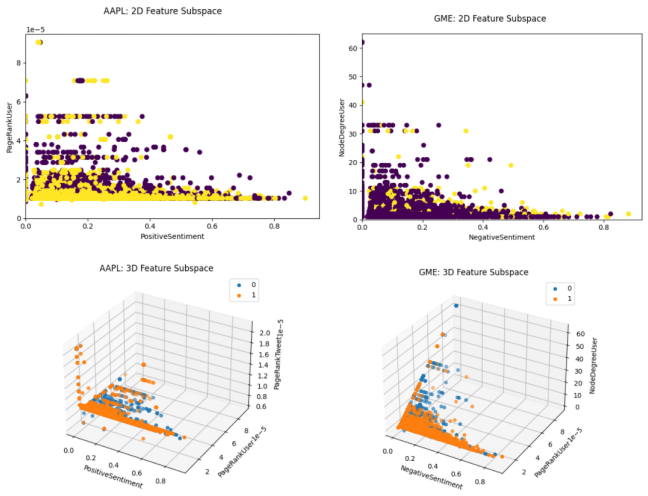

Figure 1. The 2D plots (top row) are not linearly separable. But clusters form in 3D. In the bottom left plot, the orange points form a plane parallel to the PageRankTweet axis and the blue points form a plane that is off at an angle. The planes in the bottom right plot are much closer and parallel, but are still largely separate clusters. In both cases, a separating plane exists that divides most points correctly.

4 Experimental Settings and Baselines

The objective of this project is a binary classification task. Given data on a particular date, predict the stock price motion (0 or 1) for the next date. Label 0 corresponds to negative or zero change and label 1 represents positive change. The available data for any given date are the tweets and Reddit posts that mention the target stock name or ticker. For any tweet or post, we extract features and feed them as input to the classifiers. In addition to the sentiment and graph features mentioned in the Data Analysis section, our classifiers also vectorize and embed the unstructured text data into vectors. More details are included under the Baseline section.

From insight 1 under the Data Analysis section, we observed that our binary dataset is unbalanced. Thus, we selected evaluation metrics that can handle unbalanced data distributions: F1-score, precision, and recall.

For our experiments, the social media data was divided with a 70/10/20 training/validation/test split. The models discussed are small enough to be trained on a MacBook Pro CPU in under two hours.

4.1 Baseline Description

The first baseline model uses a TFIDF vectorizer and a support vector machine to predict the stock movement, which is a commonly used and simple approach in natural language processing. The TFIDF vectorizer calculates the frequency of each term in every tweet/post and divides it by the frequency of that term in the entire text corpus. This vector representation of each post is then classified with a support vector machine to predict if the stock price will rise or fall the next day. For the SVM, a linear kernel was used and the validation set was utilized to select a C hyperparameter that optimizes the model's performance on the validation set, which turned out to be C=0.1.

The second baseline model was taken from a recent paper entitled 'Artificial Intelligence Prediction of Stock Prices Using Social Media' by Ranawat and Giani [5]. The model from this paper uses GloVe word embeddings that were pretrained on 2 billion of tweets, which are then fed into a long short-term memory (LSTM) network to predict stock movements. The authors of this study claim that this approach has two main advantages over the more traditional approach used in the first baseline. First, the GloVe word embeddings capture semantic similarities between words by projecting them to a similar space. Secondly, the LSTM model takes into account word order and context, which the TFIDF approach ignores. Initial values for the three relevant hyperparameters, LSTM hidden state size, dropout, and batch size, for this model were taken from the paper by Ranawat. The validation set was used to fine-tune these hyperparameter values and the initial values from the paper were found to be optimal (or statistically indistinguishable from optimal) on our dataset as well. Therefore, the chosen hyperparameter values were 100, 0.4, and 8 for LSTM hidden state size, dropout, and batch size respectively.

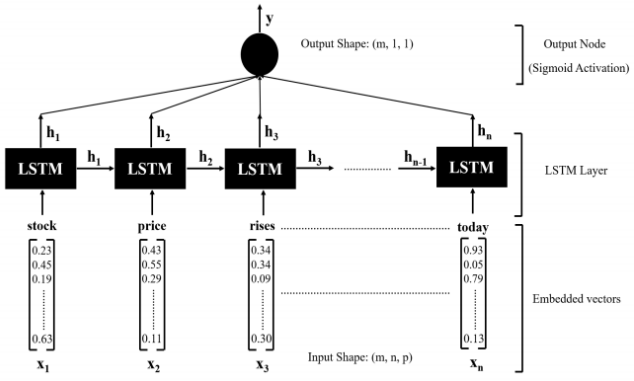

Figure 2. GloVe Embeddings Baseline Network Architecture

We will explore ways to add features to the baseline model architectures in an attempt to get better performance on our specific dataset. We will incorporate these improvements in our proposed methods discussed in the next section.

Since our dataset includes a limited number of stocks during a time period of increased market volatility, it is difficult to compare our results to the state-of-the-art from literature.

5 Proposed Methods

The initial design for the proposed methodology consisted of the following pipeline: input node embedding layer, LSTM layer, concatenation layer (LSTM output, graph node centrality and importance, sentiment scores), a fully connected layer and, finally, a singular output node with a softmax activation function. However, as stated under Figure 4, the LSTM baseline even with oversampling of the minority class, could not outperform the simpler linear SVM baseline. For this reason and the principle of Occam's Razor, we opted for the following changes in the final proposed architecture: replace the LSTM layer with TF-IDF vectorization and replace the fully connected classifier with a linear SVM. Because the TF-IDF vector length is on the order of the vocabulary size, the SVM would become biased against the graph and sentiment features which are much less in number at 17 total features. A comprehensive detailing of the 17 features used is discussed in the Data Analysis section above (Section 3.1). Thus, we reduce the dimensions of the TF-IDF vectors using principal component analysis (PCA). After evaluating various values for the number of PCA components on a validation set, we arrived at an optimal value of 500 components.

We expect the proposed model to be an improvement over the baselines for a couple reasons. The baselines look purely at either the vectorized form or LSTM features of the input text. Although these features capture some notion of the meaning behind the words, they do not explicitly measure the sentiment despite having been shown to be correlated to stock price in prior literature [1, 4]. Secondly, the graph features capture the structure of the Reddit and Twitter networks and the importance of the posts, tweets, and authors in those networks. Adding this additional information should produce a model with stronger predictive power.

6 Experiments and Results

The first baseline model used GloVe embeddings and an LSTM, and was evaluated on the held-out testing set. The precision, recall and F1 score for each class can be seen in Figure 4.

The second baseline model used TFIDF vectorization and a support vector machine, and was evaluated on the held-out testing set. The precision, recall and F1 score for each class can be seen in Figure 5.

Our proposed method was conducted using the exact same data as the baseline. The evaluation metrics, as shown in Figure 6, were also the same as the baseline's.

As evidenced by the f1 accuracy showing in Figure 6, our new model did not offer the improvement over the TF-IDF baseline model that we were hoping for. As a matter of fact,

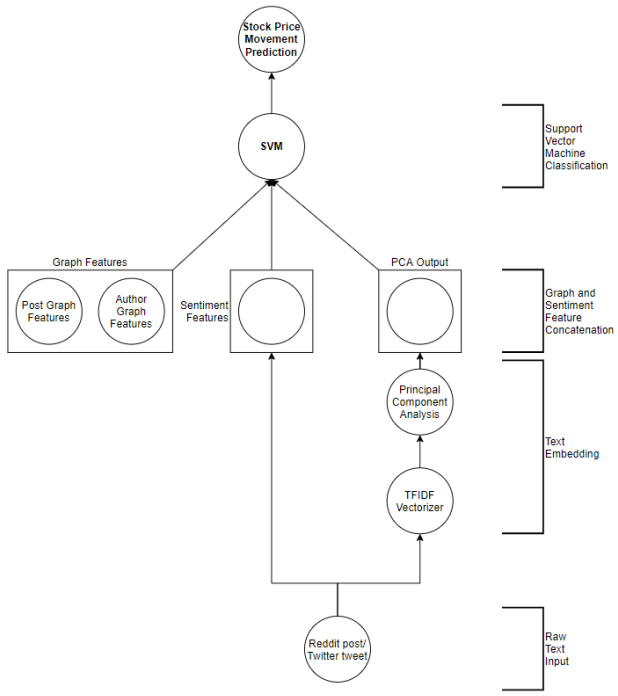

Figure 3. Final Architecture.

GloVe Embeddings	precision	recall	f1-score
Accuracy			0.66
Macro Average	0.57	0.52	0.49
Weighted Average	0.60	0.66	0.59
Classification			
0	0.68	0.93	0.78
1	0.45	0.12	0.19

Figure 4. GloVe Embeddings Baseline Model Results. With oversampling of the minority class, the macro f1-score did improve to 0.60 from 0.49. However, this is still worse than the SVM baseline.

it performed worse than before, with a macro f1-score of 0.62 compared to a macro f1-score of 0.64 for the TF-IDF baseline. This is likely due to several reasons. First, it is likely that the added graph and sentiment features did not contain significant information, at least by the means we calculated them, and thus were not of use for the prediction task. Second, our use of PCA for the TF-IDF vectors caused a slight loss in the information present in these vectors.

TFIDF	precision	recall	f1-score
Accuracy			0.70
Macro Average	0.66	0.63	0.64
Weighted Average	0.69	0.70	0.69
Classification			
0	0.74	0.85	0.79
1	0.57	0.41	0.48

Figure 5. TFIDF Baseline Model Results

SVM Architecture	precision	recall	f1-score
Accuracy			0.62
Macro Average	0.69	0.69	0.62
Weighted Average	0.76	0.62	0.62
Classification			
0	0.90	0.47	0.61
1	0.47	0.90	0.62

Figure 6. Proposed Model Results

7 Conclusion

Our model has several shortcomings and limitations. Among these include the short time period and small number of companies examined. Furthermore, several of the stocks chosen were highly volatile stocks. We also ignored the hierarchy of comments and following/follower relationship between users, which could allow us to weight comments and posts more appropriately. Furthermore, having the comment hierarchy could allow us to infer when a certain stock is being referenced to by a comment, even when the stock is not mentioned explicitly. Since our approach discarded texts that did not contain keywords, our model has access to less information.

Addressing the shortcomings in future work would contribute to a more successful model. Measures that could be taken include working with a greater number of companies, analyzing stock prices over a longer time period, choosing less volatile stocks, and implementing more robust text and graph features. An example of one implementation of graph features is the use of a graph convolutional neural network (GCNN) to create vector representation for each post.

We could also revisit the use of a neural network for our classification. Instead of using GloVe embeddings, we could perhaps use an alternative form of word embeddings, such as pretrained BERT embeddings.

8 Contribution

All team members contributed a similar amount of effort.

References

[1] Scott Coyne, Praveen Madiraju, and Joseph Coelho. Forecasting stock prices using social media analysis. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pages 1031–1038, 2017. doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.169.

[2] C. Hutto and Eric Gilbert. Vader: a parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1), May 2014. ISSN 2334-0770. URL https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

[3] Joseph Lai. Josephlai241/urs, March 2021. URL https://github.com/JosephLai241/URS. original-date: 2019-03-20T20:04:13Z.

[4] Anshul Mittal. Stock Prediction Using Twitter Sentiment Analysis, 2011. URL https://www.semanticscholar.org/paper/Stock-Prediction-Using-Twitter-Sentiment-Analysis-Mittal/4ecc55e1c3ff1cee41f21e5b0a3b22c58d04c9d6.

[5] Kavyashree Ranawat and Stefano Giani. Artificial intelligence prediction of stock prices using social media. *arXiv:2101.08986 [cs]*, January 2021. URL http://arxiv.org/abs/2101.08986. arXiv: 2101.08986.

[6] TWINT Project. Twintproject/twint, March 2021. URL https://github.com/twintproject/twint. original-date: 2017-06-10T15:16:49Z.

[7] Y. Wang and Y. Wang. Using social media mining technology to assist in price prediction of stock market. In *2016 IEEE International Conference on Big Data Analysis (ICBDA)*, pages 1–4, March 2016. doi: 10.1109/ICBDA.2016.7509794.