Predicting Stock Market Using Sentiment Analysis and Machine Learning

Ambrose Karella, Janam Patel
College of Computing & Informatics, Drexel University, Philadelphia, PA 19104, USA

Abstract—The one and only thing an investor can ask for is to have the best tool that can predict the outcome of the stock market. But the stock market doesn't rely solely on numbers, companies' financial status, or previous patterns. It can also be affected by external factors like social media and financial news. This discourse has a large impact on the stock's movement since many people make their decision based on these discussions. Machine learning algorithms and models can help extract this information from news and social media posts and process it into actionable insight.

In an attempt to create this actionable insight, our team created a pipeline to extract thousands of full articles, process them using sentiment analysis, and use these sentiment scores along with other metrics to predict price direction. As a result, our team was able to create a classification with 60% accuracy. In a test with real-world data, this model was able to make between 0.5% and 4.8% profit.

Keywords—Market, Stock, Stocks, Prediction, Classification

---

1 - Introduction

The purpose of this project was to explore the possibility of predicting stock price movement through the use of written communication about said stocks. Written communication can take the form of news articles, social media, video transcripts, or press releases. Our primary focus was the evaluation of both news and social media posts.

Using information pulled from various news and social media sites, we looked into the possibility that these sources could provide valuable insight into the movement of a stock. To test the utility of this data, we used various methods including TF*IDF, sentiment analysis, stock relationships via scraping, and various machine learning techniques.

2 - Dataset and Collection

Major components of our dataset were sourced via Finnhub's Stock API. This allowed us to complete an initial collection of a ticker's open price, close price, volume, social sentiment score, article headline, headline sentiment, and links to said article. Using the API provided by Finnhub sped up initial collection dramatically and allowed us

to explore more advanced collection methods. After collecting URLs of each news article, we built a method to scrape the full article off sites that had the most articles in the dataset. These sites include Yahoo, Marketwatch, CNBC, and 7 others. Due to some preventative measures of select sites, like captchas and JavaScript loaded pages, we were only able to gather articles for 65% of our dataset. However, the collection of full articles allowed our team to collect some richer data.

With the full articles, we were able to collect words important to the outlook of the article. Positive words like “rose”, “holding”, and “growth” and negative words like “loss”, “fell”, and “underperformed” were collected. This allowed us not only to evaluate full articles based on off-the-shelf sentiment analysis libraries like StanfordCoreNLP and NLTK but to also base the article’s outlook on the appearance of negative and positive words related to stock trading. For select sites, we were also able to collect tickers mentioned in articles. This allowed us to build a list of related stocks.

Attribute	**Datatype**
ticker	string
date	date
headline	list
source	list
headline	list
source	list
url	list
amount_of_articles	integer
open	float
close	float
volume	integer
social_sentiments	float
mentions	integer
news_sentiment	float
close_better	boolean
tomorrow_better	boolean

Table 1: Dataset attribute and their datatypes.

3 - Exploratory Data Analysis

Our team’s exploratory data analysis uncovered some issues with the collection and offered a few insights into which attributes may be important to training machine learning models. One of the first issues that came to our attention was that Finnhub’s social sentiment score seems to be in beta and is relatively new. We assume that it was rolled out on March 16, 2021, because no data before that date exists. Another issue we faced is that news article amounts per stock can vary widely on a day-to-day basis, with the minimum mentions in our set being 0 and the maximum being 1814.

During our exploratory data analysis, we also discovered that this amount of mentions has the most correlation to our “tomorrow_better” label, a binary indication if the stock performs better on the next day.

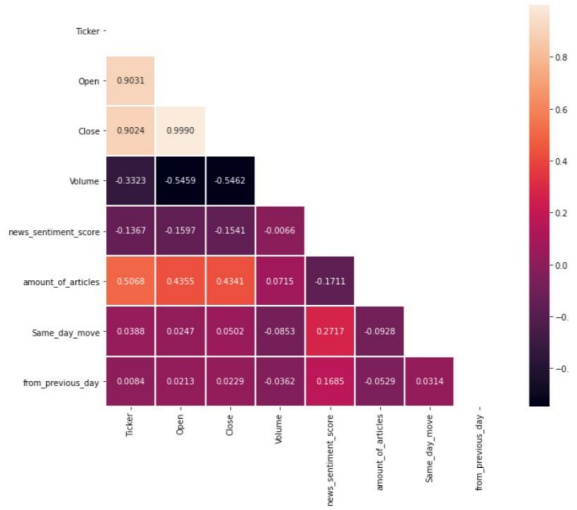

Figure 1: Dataset correlation heatmap.

Looking further into this, it appears that since the "tomorrow_better" label trails upticks in mentions, it could be even more correlated to a prediction two days into the future rather than one.

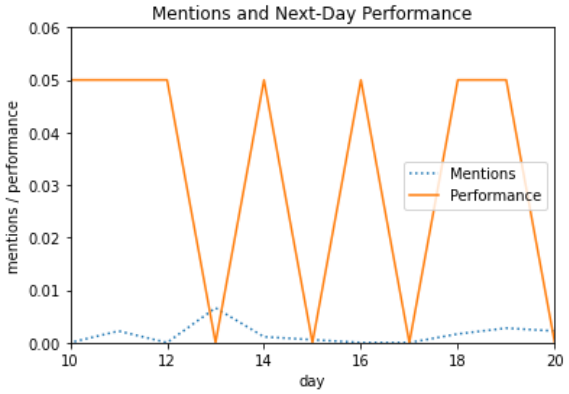

Figure 2: Mentions and next-day performance. Note that next-day performance is binary with only high and low states.

4 - Machine Learning Methodology

4.1 - Expectations

Due to the limitations in information, our team was skeptical that our model could score much higher than about 50% accuracy. Our low expectation was due to the number of mentions being the most correlated attribute to stocks rising. Both news and social sentiment, which our team was most hopeful to contribute to predicting stock direction, proved to only have weak negative correlations. No evaluation of any text seemed to have an effect on the labels.

4.2 - Process

In order to find the best model to predict the data, we first tested a variety of classifiers. This includes logistic regression, K-nearest neighbors, decision trees, etc. Our two best scoring classifiers were K-nearest neighbors and random forest. Prior to tuning, they scored 0.65 and 0.599 respectively.

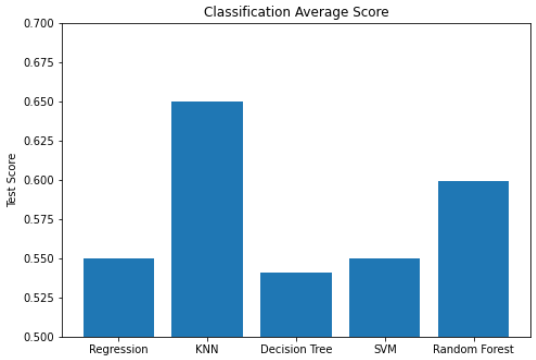

Figure 3: Average score of various classification algorithms after 50 iterations.

After tuning and cross-validation, K-nearest neighbors scored 0.575 and random forest scored 0.603. This slightly outperformed our expectations, and given the seemingly random nature of the stock market, 60% accuracy may be better odds than the average person's speculation in the market.

5 - Conclusion

While our pipeline performed better than a 50% average, it would be interesting to see the possibility of implementing this technology in a real-world application. It may not perform well on full stocks, due to the variability of stock prices leading to larger gains or losses depending on the initial price of the stock. However, applying this method to fractional purchases of stocks has the potential to be profitable.

In order to test the feasibility of our model, our team calculated all the purchases our model would make over a 20 day period of UBER. With an average purchase price of $47.91 on the 6 days purchased, our model made a profit of $2.31 by buying near open and selling at market close. This would be a 4.8% profit over the span of 20 days. A more realistic scenario may be buying closer to close because the model would have access to the volume and price that is nearing close. In this scenario, the profit was $0.25, which is a 0.5% profit.