https://github.com/siyu-liu315/Tema8-888

Trading on Sentiment: Using Stock News and Twitter Sentiment For Better Financial Predictions

Cohort A Team 8
Senbo Zhang, Zinan Chen, Siyu Liu,
Qiuhao Chengyong, & Tyler McMurray

Our Statement

The Stock Market is an area that is full of information and has a great quantity of data. The Stock Market is also an important aspect of not only the United States economy but also important to the global economy as well due to the high intertwinedness of most countries' economies. Despite the great importance and great potential to benefit from the stock market there has been no sure fire way to accurately predict information from the stock market. Whether this is an impossible task to dive into or its due to the limitation of current ideas it still seems important to conduct a proper analysis. Going through the analysis and utilizing tools like machine learning techniques to predict the stock market might not be fruitful in its power to profit from the stock market but it might be useful for better understanding relevant concepts, relationships, and limitations with working analytics and financial data.

An increasingly important method in the machine learning world is Natural Language Processing, NLP, which has many subsets within itself. We would like to utilize typical financial information in conjunction with derived data from various NLP sources for our analysis. We believe that by using NLP it might help to highlight and discover certain important aspects that impact the stock market, that might not be seen in the common quantitative financial data used in other analyses. A great example of how it may be important is the relevant situation of March 2020 with the outbreak of the virus. The stock market was in trouble due to the disease, many indexes that track the stock market were hitting lows and are continuing to decrease. NLP has the ability to discover insights and patterns within sources like news, twitter, and other unconventional data sources that may be impacting the stock market that might not be so easily or at all represented within the financial quantitative data.

Due to both the growing desire to utilize NLP techniques with more conventional quantitative data and the extreme relevance of the stock market currently with the outbreak of the COVID-19 and its implication on the stock market as news and tweets most certainly have some impact on how people are operating within the stock market. We want to utilize sentimental analysis on various sources like news and twitter information to add features that we will use with more traditional quantitative financial data to use in predictive modeling in order to get a better understanding of the importance of external sources on the stock market. We will also measure our predictive models accuracy and compare itself against differing window sizes to see if there are any changes.

Our Datasets

Throughout the project we have utilized various sources of data. The most important datasets that we collected and utilized in our final project and script code are two textual datasets one including tweets and the other including news articles relevant to the stock market. The other important dataset was stock closing price which utilized the yfinance package which is an API to download financial information off of Yahoo Finance.

Our Twitter Dataset

We utilized tweets pulled from twitter. We at first attempted to utilize the twitter API but found a dataset that included a large quantity of sorted tweets that were specific to multiple companies. These distinctions between stocks were made easily available by the "Cashtag" function of Twitter. This function allows users to discuss stocks with a format “$ + stock tickers” such as “$AAL” and “$AMZN”. Our dataset is from the website followthehashtag.com which is a commercial platform providing twitter data. This dataset covered around one million tweets mentioning cashtags of Nasdaq 100 component stocks in 89 days, from March 10, 2016 to June 15, 2016. Inside the dataset each observation was a tweet which includes the text of the tweet, who they are, the number of followers, number they are following, and other similar information.

Our News Dataset

We also used a kaggle dataset that included news articles headlines that were crawled from Reddit from a subreddit that included Daily News. There are two columns in the Reddit News dataset. The first column is the “date”, and the second column is the “news headlines”. The time range of the dataset is from June 08, 2008 to July 01, 2016. There were almost 74,000 unique headlines during that period.

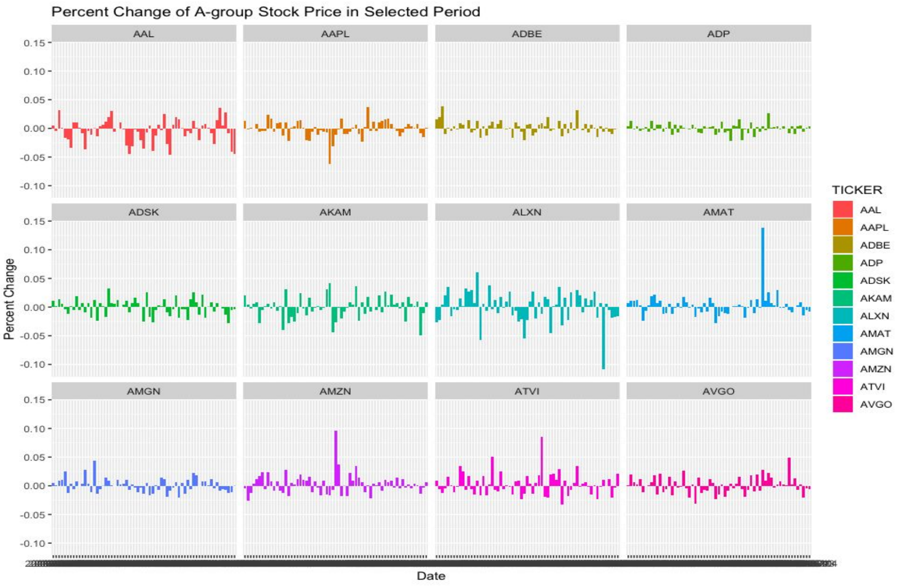
Our Yahoo Finance Dataset

Using the yfinance API, we pulled information from Yahoo Finance. The pulled information included the daily closing price of 100 companies of NASDAQ. We decided to pull a one year period starting june 2015 to june 2016. We also pulled six years of information on the NASDAQ index, from 2010 to 2016.

Our Exploration Of The Data & Analysis

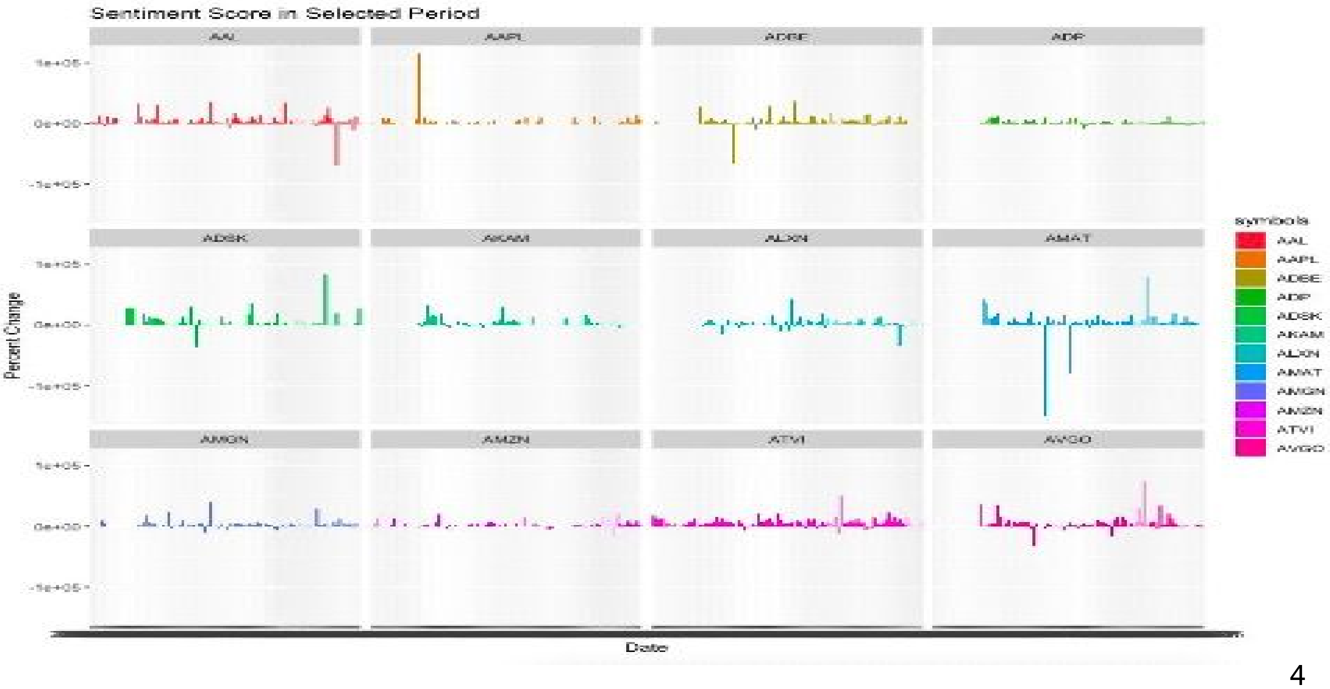
Stock price change analysis

The bar graph below, the next page, shows the price fluctuations of 12 companies over 79 days. As we have numerous companies in consideration for this project and for the simplicity of this visualization we decided to just show companies with tickers starting with “A” .



Twitter information Analysis

We split the content and followers by the ticker symbols, then we created 12 individual data frames for the sentiment analysis part. We choose the tokenize sentiment method since most tweets are not complete sentences. Compared with the sentiment chart, we found that the correlation between social emotion and stock trend are remarkable at price over-fluctuation. Then, we want to combine the sentiment score and the stock price change to a new dataset to train the model.



An Exploration of Our News Data

We downloaded the News dataset from Kaggle. The news dataset includes the 2008-08-08 to 2016-06-30 top 20 news' topic from Reddit. We combined the Top 20 topics to be one variable called Top 1. We used sentence sentiment to analyze news topics for each date and get a new variable called polarity. We used the polarity dataset to combine the NASDAQ 100 stock index by Date. Before we use the R sentiment method, we tried BERT to predict stock price change (buy or sell). The accuracy for the BERT prediction was 54% which was not ideal for us. Finally, we decided to make the polarity to be one variable in the LSTM method to predict the NASDAQ 100 stock index price.

Rolling Windows

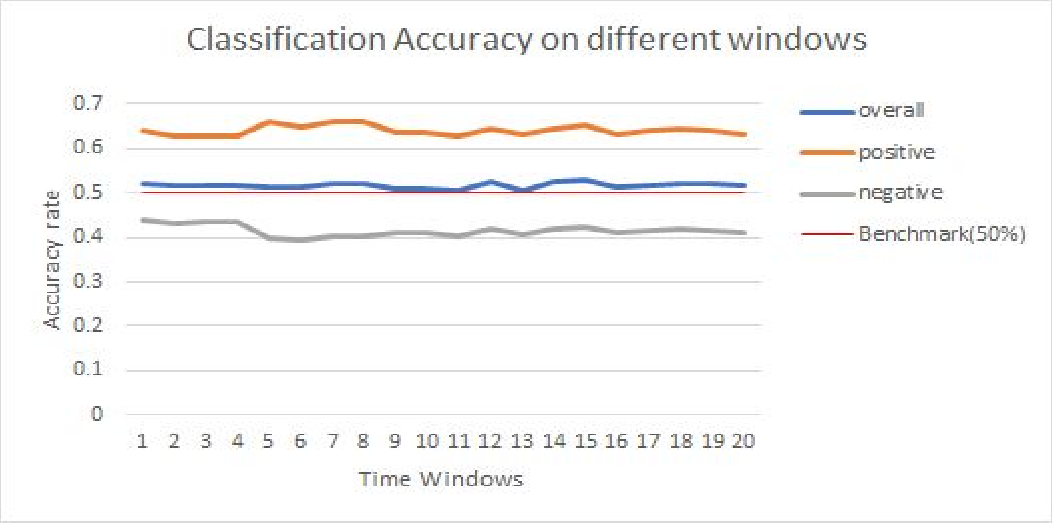
We have a large range of data over many days. We could have just utilized a simple approach and used all the data available for our predictions but using all the data might not actually be the best decision. So a method we attempted was creating a rolling window over many algorithms to see if different algorithms and including different amounts of days would yield better results for our one day prediction. We found that for our prediction algorithms the best window size was including eight days of data for our regression model and seven days for our classification model. This was true for both our regression prediction for future stock price but also for our classification problem using sentiment analysis to predict if price will go up the next day.

Our Methodology With Our Results

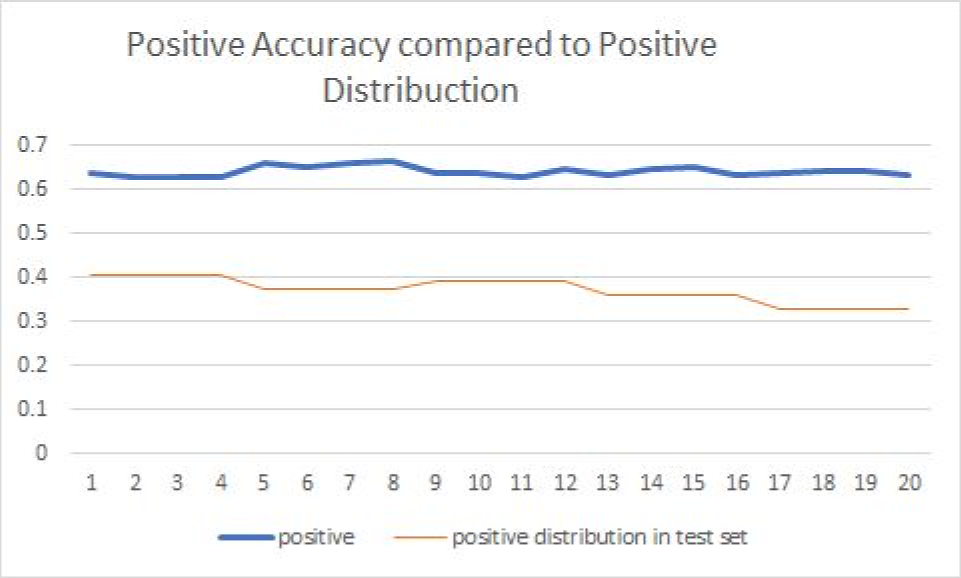
Combining Sentiment & Quantitative Data for A Better Portfolio

The goal here was to create a small portfolio containing the top five companies predicted to increase the next day. The important aspect of this portfolio is that including other external sources, like in this case textual data related to these companies a better product would be created.

We first took our tweets we had procured and ran a sentimental analysis on these tweets. All of the tweets' sentiment are analyzed and combined with compounding factors (users' following number) which adjust its weights. Then we aggregated all users' sentiment score of each stock on a daily basis. Market change percent of the prior day is another feature we use as stock would have some momentum. We tested a few different classification algorithms like a random forest and boosting. We decided to train the model with 80% being in the train set and rest test set to validate the best model. We incorporated a rolling window into this algorithm which increased its accuracy, using 7 days of past data to make tomorrow's prediction. For this classification model we found that the random forest model performed best for this problem. We got a little bit higher than 50% overall accuracy for all windows. However, unfortunately the results are not as significant as we expected. The highest accuracy is around 52% at 7 days windows.



We also found an interesting pattern in our results. It seems that our model performs better when it predicts the stock will increase. The accuracy is above 60%. We compare this accuracy with a 50% benchmark(flip a coin) as well as the distribution of stock increase in the test set.



Our next step was creating a regression model to try to predict the next day's closing stock price. We decided to scale the data and roll the window throughout the entire year to predict stock price starting at just one day. We also tried different algorithms to see which may be the best for our problem. After running through many algorithms and all of the window sizes we found that the best algorithm for this specific problem was a simple linear regression with a window size of 8.

The final step to this was combining both models into a better product. Originally just using the regression model 60% of random buckets beat the regression algorithm outright and it did not beat the average of all the companies within the dataset. These random buckets were just randomly sampled companies from within the dataset. There were ten random buckets in total and it was one way in which we tried validating our results. However, joining the model predictions together and filtering for classifications, that said to buy the stock, lead to much better results. By joining the classifications with the regression it got much better at identifying the false positives of the regression. The Portfolio, when combined, beat out all of the random baskets and the average of the market. As we will discuss later in our paper, we believe there might be more overarching reasons for such stellar performance. The important takeaway that should really be derived from this project is the importance and usefulness of combining ulterior information in predictions that can have important value that has not always been considered or might not be. These tweets hold, quite literally in this example, valuable insight that other data points are not picking up right away.

Sentiment analysis from News & Time Series: LSTM model

Components of Time Series

In general, a time series consists of three systematic components: level, trend, and seasonality.

Level means the average stock price (Index price) in the series

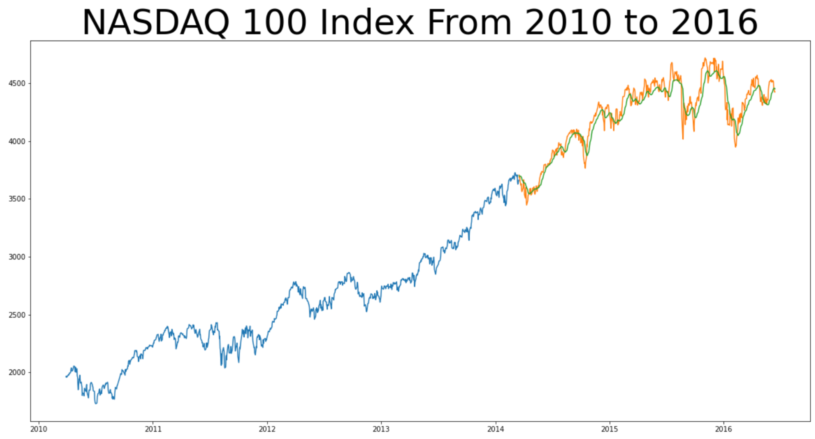
Trend means the increasing or decreasing price in the series

Seasonality means the repeating cycle in the series

It also includes one non-systematic component called noise, the random variation in the series.

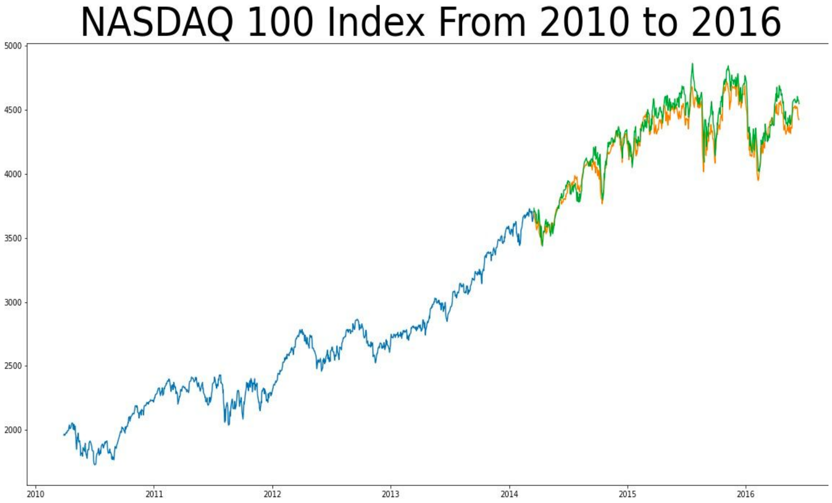
Model Selection

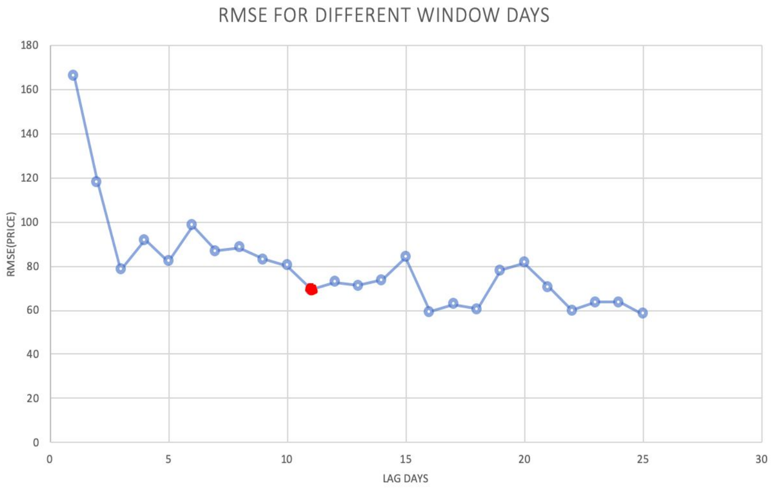
We decided to use the LSTM model to predict stock price based on time series because the algorithm can store important past information and forget unimportant information which makes the prediction more accurate. The graph below shows the result when we used LSTM to predict the NASDAQ 100 index based on historical data. Below and for further reference the blue line indicates past movement. The green line represents our predictions and the orange the real movements of the index.



Advanced model: LSTM + Sentiment Analysis

Even though the LSTM performs very well on stock price prediction, it is not enough to determine whether the stock price will go up or down since the stock price is also influenced by company news, public sentiment, and other factors which are often unpredictable. We create a new stock market forecasting model based on LSTM and sentiment. The first step is to create a stock price prediction model with LSTM, and the second step is to add an extra feature to the news sentiment to improve the accuracy. We also consider the lag between the market information and its impact on buying/selling behavior by running the model with different lag days to find the best window with the minimum MSE. The new forecasting model significantly reduces the MSE compared with the basic LSTM model, and the variation between predict price and actual price also decreases.





Our Own Comments & Criticism

There are many areas in which we could see how to better our project in the future. We also realize that our project results seem alarmingly well. We noticed these areas and wanted to address them.

The first area that we saw we could improve upon was including more algorithms into consideration for this project. We utilized many like random forest, time series, regressions, and others but there could definitely be more that may be better suited for trying to make a very strong portfolio from predictions. However, due to the limitations of time and the lack of confidence in effectively implementing other algorithms led us to use the ones we were comfortable with for this project.

Another area that could have been improved upon was working with the rolling windows. The goal of working with those windows led us to select one day to base all of our results on for one model. Instead we could have tried the rolling window for each day to see if that window was truly the best for our type of problem.

Which brings us on to the extremely good results that we have achieved. The results are specifically designed to give us the best predictions for that one days but implementing this model and concept to any other day could prove that it does not work well at all in the long run or for any other day. Meaning if we tried this same form of validation using the same procedures for the next day after we could see extremely different results. Most simplistically, this algorithm worked one specific day we were completely targeting not for an actual trading algorithm. Also the actual amount of companies we used were extremely small, in the future we should incorporate more companies and more diversified companies. This data is extremely old as well and most likely would not be at all relevant to today.

Our conclusion

Despite the seemingly optimistic results that we have achieved, we remain sceptical of its true value outside of this very niche product and project we did for many of the reasons stated above. Further evaluation over a longer period of time is needed to even think of considering it fruitful.

What it does show however is the importance of using alternative data sources rather than just using traditional ones. These alternative forms of data which in our project were tweets and news headlines when combined with more traditional data like stock closing prices can really improve results. It also highlights the importance of procuring data that might hold more insight into bettering predictions whether that being using API to pull off twitter or creating web crawls to pull information off websites like news articles. It also highlights the importance of using alternative types of data like text, which is becoming more important. A future consideration to this is using speech recognition algorithms to gauge a large amount of public and legal data.

These algorithms might pick up tones of voice or niche things that not everyone can or wants to read into as an algorithm could, to use to better predictions.

The point of this project was never to create the best product. We knew from the start that many have tried and very few have successfully created products from machine learning algorithms in the stock market. What it taught us was how important it was to use different techniques, algorithms, tools, and alternative data sources and types that might hold value and insights that other data might not fully represent.

Reference:

Aaron7sun. "Daily News for Stock Market Prediction." *Kaggle*, 13 Nov. 2019, www.kaggle.com/aaron7sun/stocknews.

Brownlee, Jason. "How to Decompose Time Series Data into Trend and Seasonality." *Machine Learning Mastery*, 28 Aug. 2019, machinelearningmastery.com/decompose-time-series-data-trend-seasonality/.

"One Hundred NASDAQ 100 Companies - Free Twitter Datasets." *Followthehashtag // Free Twitter Search Analytics and Business Intelligence Tool*, followthehashtag.com/datasets/nasdaq-100-companies-free-twitter-dataset/.

"Yahoo Finance - Stock Market Live, Quotes, Business & Finance News." *Yahoo! Finance*, Yahoo!, finance.yahoo.com/.

"Yfinance." *PyPI*, pypi.org/project/yfinance/.