

Twitter Stock Sentiment Analysis

(1454 words, 6 mins 7 secs reading time)

Author: Jake Waffle

What is Sentiment Analysis

Sentiment analysis is a strategy that is usually associated with social media. This strategy is useful for determining the general public's sentiment towards something (e.g. a company, stocks, crypto currencies, etc) and is ultimately a classification problem. The possible classifications are generally positive, negative and neutral, but can include specific feelings/emotions like anger, happy, sad, etc. When analyzing the things people say, this sort of strategy allows us to determine how people are feeling towards the company/stock/etc in question. That then allows us to make decisions in response to how people are feeling. A company might use this strategy to determine how well a past decision has been viewed by the general public and then use that information to learn from their victories or mistakes. A stock broker might also use a tool that utilizes sentiment analysis to help them decide if they should sell or buy certain stocks based on what other people might do with those stocks.

Stocks on Social Media

The general public talks about stocks on a daily basis and the value of each stock is greatly affected by people selling or buying their shares. Each stock has a particular identifier (e.g. \$NVDA) that is usually referenced in these discussions, which makes it possible to accurately determine which specific stock people talking about. These discussions also convey a certain sentiment of how these people feel towards the stock at the time of the posts on social media.

A stock broker probably watches the value charts for stocks and the announcements made by the companies, but the general sentiment towards the companies is another important aspect for allowing a broker to accurately predict the movement of a stock's value. Stock brokers at the very least have some sort of news feed for helping them predict future stock values. It's unclear whether social media is the best source of information since the experts might be withholding their views while the uninformed masses might be more vocal about their opinions. However, it's probably a good idea to listen to normal people because they can definitely affect the stock market if they all sell or buy at the same time. In the end, a wide array of sources on stocks is probably necessary to make accurate decisions, but stock sentiment analysis on social media could be one of the valuable sources of information.

Data Collection

There are various sources of social media discussions out there: Twitter, Reddit, Facebook, etc. These social media sites usually have an API that allows us to write scripts for retrieving relevant discussions. These relevant discussions should at the very least contain a stock identifier (e.g. \$NVDA) so that it's possible to associate a sentiment with a stock. In addition to the social media discussions, it is also useful to have access to the stock time-series values to check the accuracy of the price change predictions.

For our demo we will use a [dataset containing Twitter tweets that contain stock discussions](#). This dataset has a version for training, which includes sentiment labels: negative, neutral and positive. It also has an unlabeled

test version, which does have creation time associated with each tweet.

Method to the Model

We will be using a simple [BERT](#) model that is fine-tuned for classifying tweet sentiments. We will use the previously mentioned training dataset for training BERT to do sentiment classification. Then we will be using the unlabeled test dataset for predicting sentiment. From there we will use [Polygon](#) to get stock price data to determine if the stock price has increased/decreased or remains unchanged following the time of each tweet. This will allow us to determine the accuracy that we can predict stock price movements based on the sentiment of the individual tweets.

Fine tuning is a technique where we adapt an existing pre-trained model to learn a specific task. For our situation that task is classifying tweet sentiments. The BERT model throughout our training process is frozen (unchanged during training) and the actual learning of the sentiment classification is handled by a neural network that takes in the hidden state of the BERT model and outputs the probabilities for the three possible sentiment classifications: negative, neutral, positive.



After training our model we ended up with a decent accuracy after 7 epochs, but the training probably could have used more epochs to improve the model. A small portion of our training set was dedicated to testing, so was ultimately left out of the training process. Below you can see the accuracy of our trained model as well as some interesting prediction results.



The Final Test

Our test is going to specifically target tweets that are focused on the SPY stock. This was done just to simplify the analysis of our results. Below we can see the price movement of the SPY stock within the time period for the tweets that we have in our test set.



This price movement information is used to determine the expected sentiment for all of our tweets. We then use that expected sentiment to check the accuracy of our sentiment analysis. This strategy is not perfect however because the authors of these tweets don't know the future of the stock market. Just because a tweet might have a positive sentiment towards a stock doesn't mean that the stock actually will have an increase in its price movement the following day.

Below we can see the expected sentiment predictions in comparison to the predicted sentiment results of our model across the same timeline as the previous stock price movement graph. The orange line represents the sentiment based on the stock price movement, while the blue line represents the sentiment predictions from our model. Due to our model serializing our classifications as numbers, 0 is a negative sentiment, 1 is a neutral sentiment and 2 is a positive sentiment. We should be able to see a negative price movement in our previous graph for the places where the expected sentiment is negative. One caveat is that our expected sentiment predictions are only based off of a single day's overall price movement between open and close and there are cases where a tweet is made multiple days before we have actual stock price information due to the market being closed over the weekends.



Our results aren't groundbreaking when using a single tweet to determine the actual price movements, but this could be a start to a much better solution. Due to the nature of our unlabeled tweets not knowing the future of the actual stock price movements it's hard to say if our model predicted their sentiments incorrectly or if the authors were just incorrect about how the stock is going to change in the following days.

Conclusion

Sentiment analysis is an interesting technique for predicting stock price changes in the future, but is far from perfect. There's no telling if the authors of the tweets being analyzed are going to be correct with their predictions, but this gives us a way to programmatically determine the overall sentiment of the public to get more information about stocks. A stock broker needs multiple sources of information for their analysis and this could be a useful source for them.

One downside of the strategy discussed in this post is that we only examine a single tweet at a time to predict stock price movements. In an ideal implementation you would get a lot of tweets for a single stock for the current day and determine the overall sentiment for the stock. This would allow us to hopefully get a better prediction because we would have input from potentially many people that all have different sources of information.

A future improvement towards making this automated would include determining overall sentiment for a stock, but the next step after that could be automatic buy/sell logic that would react to the sentiment by buying or selling at the next market open based on the predicted sentiment. From there it could either wait for the expected price movement to sell or buy. Before actually going through with making a bot for automatically trading stocks it's probably best to use historical data to determine how well the bot would have done if it was being run during some time period in the past. Using a test setup like that we could theoretically determine how much money the bot could have made if it was actually doing real trades, but that also requires additional configurations based on the amount of money it starts with and how much stock it purchases when it acts on the sentiment predictions.