Natural Language Processing For Stock Price Prediction

A PREPRINT

Abhishek Sharma[*]
University of California, Berkeley
abhisheks@berkeley.edu

August 17, 2019

ABSTRACT

Given a stock's previous data and some news information, can we predict if it's price will go up or down? In a nutshell, this is the question I attempted to answer with machine learning. The reason this problem intrigued me was because traditional regression techniques often fail in this use case. This is because stock patterns don't follow a mathematical function, which is when regression is useful. So, I theorized that a program which thinks like a human would do a better job of stock prediction. The main driver of stock price is demand. If lots of people want to buy a stock, it's price goes up. Since a company's public perception can have a strong effect on its demand, I hypothesized that predicting this value might provide more insight into forecasting the stock price movement. I decided that news is the best data source for mining public perception, which I would compute in the form of a sentiment score, taken across all news of a single day.

***K*eywords** Stock Price Prediction · Machine Learning · Natural Language Processing

1 Problem Formation

Since we just want to maximize stock return, I reduced this problem to binary classification, 0 representing a loss (net change < 0) and 1 representing a win (net change 0). Our goal is to successfully predict a stock's increase or decrease on a given day. All we have is the opening price, daily news, previous day's return, and previous day's net change. Now, the second part of this problem is also classification. In order to effectively predict the effect of news on stock prices, the news must have a sentiment score, which can be applied to a separate model that predicts the effect of this sentiment on the stock's price.

2 Subproblems

I had to split my method into 2 tasks, NLP for public perception analysis, and classification for increase/decrease prediction, using public perception and other factors. I created 2 models. The first took a list of strings (each representing a news article) and returned a net sentiment score, using sentiment analysis. The second model took a day's opening price and net sentiment score (calculated from the first model), and the net change from the day before, as well as a 0/1 signifying an increase/decrease the day before. It returned a 1 if the price would increase, and a 0 if it would decrease by the end of the given day. An alternative form of the model used regression to return an actual estimate for net change in stock price. The second model was straightforward, and I used NLTK to expedite development. I implemented multiple approaches for the second model. These were a neural network, random forests, and logistic regression for classification/ linear regression for regression

[*] Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

3 Process Issues

After creating sentiment scores, I noticed that most days had a negative net sentiment from news. This contradicts the movement of most stock prices, which is generally positive. Also, I could not tune regression to have accuracy below 100absolute percent error). Some explanations for general error include lack of account for other variables as well as quality of data.

4 Conclusion

If given more time, I would make 3 main changes: better data, more attention to regression, and more work on sentiment analysis. As of now, the data is quite general, so it might make more sense to create my own dataset with news focusing on a single company. This would most likely yield more accurate sentiment scores. Also, the built-in sentiment library from NLTK might not be the best approach to mining public approach from news. A more accurate model might combine my two models, and train a network to go directly from news text + previous stock data + stock price changes (training data) to a prediction of f(news text + previous stock data) = predicted change. Also, regression would be useful when trading on a budget. If I analyze multiple stocks, it helps to know which one will have the highest relative return, so I can buy that stock on that day, and repeat over time