Predicting Stock Value with Natural Language Processing

By Zakaria Zerhouni

Problem Statement

- Can we use Natural Language Processing to predict if a stock will increase or decrease in value based on online news articles?

Collecting the Data

In order to collect the data to make a prediction, the Selenium WebDriver was used to access articles on Seeking Alpha. A ticker symbol was input and then articles for that ticker were drawn.

- https://www.selenium.dev/
- https://seekingalpha.com/

Then yfinance was used to get historical stock data from Yahoo Finance.

- https://pypi.org/project/yfinance/

Issues with Data Collection

- Only 100 articles would load
- Inconsistent tagging in articles
- Advertisements were present in the list of articles

Data Cleaning and EDA

- Advertisements were able to be dropped as they did not have any text data associated with them.
- Timestamps had to be adjusted to match the last day of trading to have a value associated with that article.

Setting the Target

In order to decide if the price of a stock would increase or decrease, the percent difference was averaged for the week after the article was published.

If there was a positive increase in value the target was coded as 1 for a success.

If there was no change or a negative percent difference it was coded as a 0.

Model Selection

- In order to select a model a set of 6 different classifiers were attempted to benchmark performance.
- The text was processed using scikit-learn's CountVectorizer() to preprocess the text and transform it into a form that could be used with the models.
- The default hyperparameters were used for the benchmarking.
- The baseline accuracy was established to be 0.59

Benchmarking Results

Model	Training Score	Testing Score
MLP Classifier	1.0	0.68
K Neighbors Classifier	0.67	0.59
Decision Tree Classifier	1.0	0.45
Random Forest Classifier	1.0	0.59
AdaBoost Classifier	1.0	0.54
Multinomial Naive Bayes	1.0	0.72

Attempts to Improve Performance

- Multinomial Naive Bayes provided the best performance with an accuracy of 0.72.
- The Multi-layer Perceptron Neural Network was second best with an accuracy of 0.68.
- Hyper parameter tuning took place, as well as attempts to substitute the Natural Language Processing. These resulted in long grid searches and new implementations, but no marked increase in performance.

Model Selection and Summary

Multinomial Naive Bayes provided the best results consistently.

Often attempts to change the model or the Natural Language Processing led to worse results.

Multinomial Bayes clearly provided us with the ability to estimate an accuracy score of 0.72 compared to the 0.59 baseline accuracy.

Conclusion

The results of predicting even with limited data show that a prediction can be made for the next week.

Going forward, in order to make a decision, all the articles in a week can be collected. Then choose a threshold of confidence to buy a stock if the value is anticipated to increase in the next week, or sell a stock if the value is anticipated to decrease.

For n articles, if there are k successes, our ratio will be k/n.

Next Steps

- The biggest weakness of the model most likely comes from a lack of data. For any given stock only 100 articles would be collected. A new method that could aggregate thousands of articles would help to build a better vocabulary and provide better counts for predictions.
- The best method for this would be to collect data over a longer period and build a more robust data set for training our model.
- Currently, historical stock data did not improve the predictions. Looking for other metrics that could correlate to growth could assist in better predictions.

Options for Experimentation

- Use the articles of multiple stocks and finding the common words between these sets.
- Updating the target to look for a certain level of growth. For example, if the percent difference of growth isn't over 1.5%, then we do not count that article as a success.
- Investigate articles that do not provide the correct prediction and look for trends to update the vocabulary of our model.

New Problems to Investigate

- Attempting new time intervals. Are predictions better for shorter intervals? Worse for longer intervals?
- What is the maximum time into the future a collection of articles can predict?