Stock Market Analysis – Final Report

Stock Price Movements Based on Company Performance

Kristian Montoya
CSCI 4502
University of Colorado
Boulder, CO

Zaki Kidane
CSCI 4502
University of Colorado
Boulder, CO

INTRODUCTION

Understanding and predicting stock market movements is useful for individual stock market traders as well as investment firms and banks. So far, the task of successfully predicting stock prices has been done by people. We ask the question, "Can Machine Learning algorithms effectively outperform the average trader?" and seek to answer that question. By introducing the performance history of different stocks, we can visualize the data and look for correlations between stock-price history to help determine the future performance of any given stock. We intended to store the collected data in a DataFrame for efficient retrieval and manipulation to be used by a machine learning model. Having compared different machine learning algorithms, we found that Recurrent Neural Networks provided the most accurate predictions for temporal sequence data such as daily stock prices for given companies.

RELATED WORK

There have been two main approaches with regards to stock market analysis and trading: fundamental analysis (also known as the 'Warren Buffet Style') and technical analysis (*Picasso et al., 2019)*. Fundamental analysis focuses on the company performance, financial conditions, operations and macroeconomic indicators to help decide whether to buy stocks in a company or not. This method of investing is generally not used for short trading spans as day-trading (where one buys and sells stocks within a day) or swing-trading (holding on to stocks for up to 20 days). It is used for long-term investment in a company stock for a year or more. On the other hand, technical analysis

depends on historical stock price trends and makes the assumption that the trends generally repeat. This method is used for short-term, high frequency trading.

Some researchers have used technical analysis to predict price movements (*Chervello-Royo et al, 2015., Patel, 2015*). Some use fundamental analysis methods by mostly finding correlations between the company's yearly reports' features and the price movements, such as Price-to-Earning ratio(*Chen et al, 2017)*. There are also other methods that have been attempted, such as the use of sentiment analysis from social media (*Nguyen,* 2015). The approach on this paper will be fundamental analysis, analyzing correlations between company-performance fundamentals and stock prices.

PROPOSED WORK

Datasets

The dataset used throughout the project is the "New York Stock Exchange" that can be found on Kaggle [1]. The dataset is made up of four different csv files that relay an abundance of information. The first and main set of the data includes 79 different features that gather yearly data of companies over the span of four years. These features are basic information of the well-being of a given company and include several key components, including: cash ratio, gross profit, earnings per share, etc. Many of these features may prove to be a driving force behind the company's stock price at any given time, given the right frame of time within which stocks are bought and sold is found (for example, consistent growth in profits over a decade may mean the stock price considerably increases over that decade). However, it turned out that many of these features recorded for four years proved to have little predictive power and only hurt our model's performance by adding unnecessary noise. To side-step this problem, we found that simply using stock price history to perform technical analysis through RNN's provided better results. Doing so allowed our model to increase in performance and give more accurate results when testing our model.

The second and third dataset used are composed of opening and closing prices of a given company over the course of a year that was recorded on a daily basis (over weekdays). Conveniently, the company's ticker symbol (TSLA corresponds to the company, Tesla) is consistent throughout all the datasets. This allowed to build and collect all data in a single DataFrame (or however one decides to collect/organize the data for the appropriate algorithm that is being tried out) to better understand the overall performance of a company. By combining the data of a given company from all the datasets, a better picture

was formed and connections were easier to find. Combining the datasets together also did not bring any significant improvement in boosting our model's overall performance; the company's annual performance and state of wellbeing, paired with daily opening/closing prices collected over several years, gave some indication on where the company stands performance wise. However, simply using the daily opening and closing prices as well as the daily highs and low and the stock price of that company provided the most accurate results.

The final dataset introduces some basic information about the company's sector, sub-industry, and location of the company's main address. Even though this could have allowed us to section up companies based on their industry or sector to find trends over certain periods of times, we prioritized gaining as accurate future trends as possible to stick to the goal of simulating a stock trading decision. In reality, stock traders utilize sectors to hone in on potential profitable single stocks. If one can predict future stock prices for a single stock and even do so for several stocks using the same algorithm, it is a more efficient form of getting the desired result, which is finding winning stocks. However, for investment firms, it may be a useful analysis to, for instance, find out whether companies within a certain industry have similar stock performance over a specified period; or more generally, find trends over sectors/industries instead of individual stocks. This is a possible direction that could be taken with the given dataset, if that is a priority or an interest.

Methods

In addition to creating an online repository to share the code update and writing code to spit out the tables' main features (column headers, number of rows etc.), the preliminary step was checking that the dataset was factual and consistent. To accomplish this, visualizations of individual stock prices and their volumes were made. After looking for anomalies and comparing the prices for random dates and stocks with online resources, we determined that the dataset was indeed accurate and consistent. An example of anomalies that popped up when checking for accuracy of the daily price history dataset was Apple's significant drop in stock price in 2014. During that year, there was no global economic collapse, but the graph resembled that of one. After fact-checking the data, it was found that the apparent drop in price was due to the 7-1 stock split in that year, which dropped the stock price to approximately 14% of the original price.

Once our visualizations have been created and saved to the GitHub, created another notebook to test and develop an ML model to fit

our prepared data. We split the original adjusted-price data set into two sets, a training and test set, that the model would use to train. We used a 75-25 split between the training and test sets, normalized the data with a min-max scaler through the sklearn library, and set up a model with basic LSTM architecture (Long-Short Term Memory). It is worth noting that we also tested a linear regression model to see how it would perform, but it fit the data so poorly that we omitted the attempt completely and only used the LSTM architecture. This was simply a design choice since the LSTM architecture is solely intended for sequential data, whereas a linear regression model is not. The architecture of the sequential model used four LSTM layers with a single density as the final layer. Once the model's architecture was established, we played around with different hyperparameter settings to tune the model and find what setting exploited the model best in relation to the data. Our final design parameters consisted of having the model use 500 epochs with a batch size of 50.

Once we established the design of our model, we converted all the appropriate code into an OOP layout so we could create any number of models to fit any number of companies. Specifically, we created objects for random, individual companies that would have their respective price data, model, and data frames inside the created object. This allowed us to create a "factory" of models for random companies so we could train, fit, and predict future prices for a variety of companies without being redundant within our code. This process was handled in the fourth notebook and replicates the design and intention of the third notebook; the only key difference being the number of companies the code could handle at runtime.

EVALUATION AND MILESTONES

As mentioned earlier, by combining the datasets in a clear manner allowed us some flexibility when cleaning and presenting the data through visualizations. Once we appropriately combined, reduced, and normalized our data we focused on selecting an appropriate Machine Learning model that fits our data to get the best results after training/testing. Such models that were tried out include linear regression and RNN's. Linear regression did not produce as accurate results as we had hoped so we discarded it and went with the more appropriate model, RNN. Of course, this included hyper tuning different parameters within each model to extract better results, but this was not the main focus of the project; our main goal was finding a concise way of combing, cleaning, and readying our data to find interesting trends. Once we trained and gathered the test results for each model, we

directly compared the accuracy between the models to find which one best suits our goals and grants the highest performance.

The performance we set out aiming for was to predict whether prices would rise or fall (that is, whether a stock is a good buy or sell) in a given time frame accurately **85%** of the time. We came to that accuracy rate as a middle ground between generating enough profits (maximized at 100% accuracy) and realistic expectations for the predictability of the stock market, which can be based on other features we will not include such as the overall economic health of the Unites States economy. Our results were encouraging, with the lowest training accuracy of 95.7%. Therefore, as far as predictive capability, the results were successful.

POSSIBLE FUTURE IMPROVEMENTS

The method used in predict future stock price movements for this project was the technical analysis, that is, looking at past prices and attempting to find trends to predict future price movements. As mentioned in the related works, this method can be combined with the fundamental analysis to reinforce predictions and base predictions on more holistic information. In addition, social sentiment may play a significant role in predicting future stock prices. Therefore, using Natural Language Processing, data from social media websites such as Twitter may be used to make predictions. Another possible avenue to explore is using NLP on news article sites, which give out not only the social sentiment in relation to a certain stock but an analysis of current events with the company. Any combination of these methods could enhance the accuracy and overall performance of the model and create a practical, innovate tool for analyzing and predicting the performance of any company's stock.

REFERENCES

- Chen, Y., Chen, Y., & Lu, C. (2017). Enhancement of stock market forecasting using an improved fundamental analysis-based approach. Soft Computing, 21(13), 3735-3757. doi:10.1007/s00500-016-2028-y

- Nguyen, T. H., Shirai, K., & Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. Expert Systems with Applications, 42(24), 9603-9611. doi:10.1016/j.eswa.2015.07.052

- Picasso, A., Merello, S., Ma, Y., Oneto, L., & Cambria, E. (2019). Technical analysis and sentiment embeddings for market trend prediction. Expert Systems with Applications, 135, 60-70. doi:10.1016/j.eswa.2019.06.014

- Cervelló-Royo, R., Guijarro, F., & Michniuk, K. (2015). Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data. Expert systems with Applications, 42(14), 5963-5975.

- Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert systems with applications, 42(1), 259-268.