Predicting Buy Sell Pattern of Stock Market Experts

Nishant Jain, Nitesh Singh Jaswal

Indiana University, Bloomington

Abstract

Generally available models try to model stock market index by capturing different factors and historical data, and topping it with news scraping. Our aim here is a bit different, we want to consider the already available high quality recommendations/predictions from investors/stock brokers which are based on expert based models for modelling stock market conditions. And we want to classify the investors based on their recommendations. That way one can know which investor to follow to invest in stocks. We can add one more dimension by considering the segment of the stock, for example it could be the case that some investors recommendations work good for the automobile industry, for some, the recommendations are more accurate for steel industry. Finally if feasible we might be able to provide stock recommendations based on the stock recommendation pattern of the investors.

Keywords: Stock predictions, Investor Predictions, Random Forests, Neural Network

1. Introduction

We usually see models trying to predict the stock market effectively by predicting the stock prices, bonds, securities or other financial features to base their buy, sell or hold decisions. Our objective was slightly different. Instead of basing a model around these core features themselves, we aimed to build a model that derives some learning from the history of buy/sell decisions of various market experts and investors themselves, and then give its own decision based on the learning it has done. We have two sub-goals to achieve this. We first want to limit our model to data related to only one company in particular (Apple Inc.) and see how it performs. Our second goal was to train our model on a dataset with many companies in it. We have used two different learning models to perform each of these goals. A multilayered Neural Network and a Random Forest Classifier.

2. Gathering Data

2.1. Who is a Market Expert?

According to the Oxford Dictionary, the word *expert* when used as noun, is defined as A person who is very knowledgeable about or skillful in a particular area. But then the question arises on how can we quantify the knowledge and skillfulness of a person? Hence we defined a set of criteria to select who qualifies as an expert and who does not. The criteria were Experience in trading, Net worth of the expert and their Average Monthly investment in the US market. Using this criteria we were able to shortlist 6 prominent investors whose trading patterns we tried to model.

2.2. Web Scraping and Data Filtering

We tried to find data in a cleaned format but data specific for our task was not readily available. So we turned to web scraping techniques to scrape the data off of online tables on websites [1]. We also had to make sure to avoid to gather data only from 2010 2018 to avoid any discrepancies arising due to the financial crisis of 2008. Now, even after cleaning and formatting our scraped data, it was not ready for use. This was because there were company specific discontinuities that affected the data we gathered. For example, on 9 June 2014, Apple Inc.(AAPL) stocks were split in a 1:7 ratio. To compensate for this we had to multiply the all AAPL stocks before 9 June 2014 by 7. A more recent case is when Insperity(NSP) split their stock in a 1:2 ratio on 19 Dec 2017. This task, although time consuming, was necessary to ensure there were no discontinuities in our data. We then split our data in a 90:10 ratio and used one for training our models and the other to test our accuracy. We also created two different datsets from our model. One with investment data for multiple stocks from multiple investors and one datset with investment data for Apple Inc.(AAPL) stock from multiple investors. We chose Apple, because it is one of the most financially stable companies in the world and hence it's stock will show lesser variance due to external factors we may not have accounted for. We will be referring to them as *Dataset 1* and *Dataset 2* respectively.

[1] http://www.gurufocus.com/

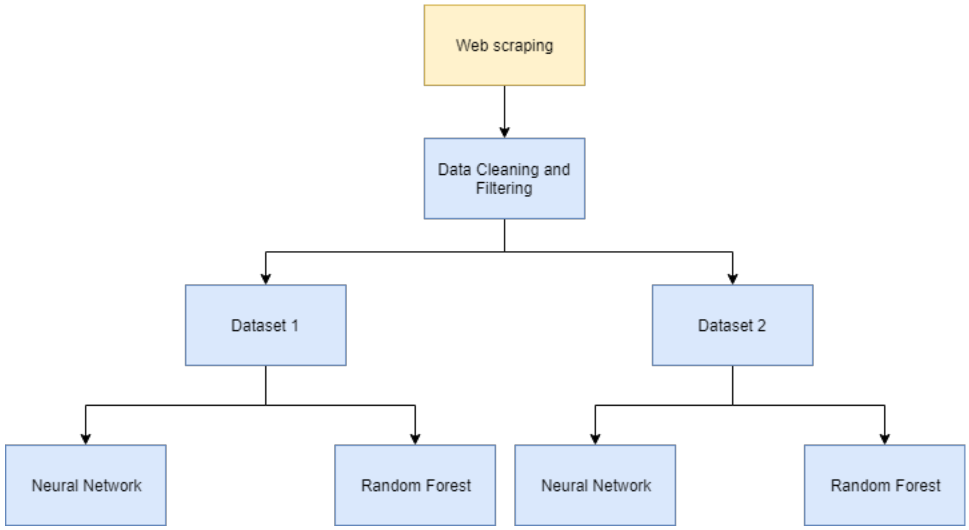

Figure 1: Data and model process

3. Models

As mentioned previously, we trained each type of model on two different datasets.

3.1. Random Forest Classifier

Stock markets are inherently non-linear in nature [1] and using linear learning techniques like Support Vector Machines and Perceptron Training often result in results. *Luckyson Khaidem, Snehanshu Saha, and Sudeepa Roy Dey* have shown in their paper [2] how Random Forest Classifiers have largely been unexplored in the domain of stock forecasting and their ability to map non-linear state spaces make them an ideal candidate in domain of stock market trading. Since the classifier constructs trees by minimizing entropy, the decision features it selects tend to have the highest variance among each other.

We created a forest of with 200 estimators and maximum tree depth of 10. We obtained the following train and test accuracy:- For Dataset 1, the RF model correctly predicted the Buy/Sell pattern of investors with an accuracy of 72.6% without overfitting as was the case in with Dataset 2. The reason for overfitting in the latter was most probably due to the fact that we were

	Training Accuracy	Testing Accuracy
Dataset 1	77.69%	72.6%
Dataset 2	90.8%	90%

Table 1: Training and Test Accuracy for Random Forests

only dealing with the investment pattern related to a single company, a few features may have dominated the others leading to the overfitting of the decision trees themselves.

3.2. Multilayered Neural Network

We created a 23 layer Neural Network with 21 hidden layers. All the layers used the Rectified Linear Unit(RELU) activation function with the final layer having a Softmax layer to classify our final decision in to one of the two classes(Buy, Sell) We also used the Adam Optimization Algorithm [3] during backpropagation.

We did try to model *Dataset 1* differently by using subsets of data containing buy-sell decisions of only single investor, but as we were getting similar results, we decided to proceed with the full data.

	Training Accuracy	Testing Accuracy
Dataset 1	78.29%	72.8%
Dataset 2	55.11%	68%

Table 2: Training and Test Accuracy for Multilayer Neural Network

The Neural Network performed relatively similarly to the Random Forest approach for *Dataset 1* but performed poorly for *Dataset 2*. This could probably be explained by lack of training data. By providing the data only related to Apple Inc. we severely limited the amounted of data we equipped the model with. Due to time constraints we unfortunately could not scrape more data to train the model.

4. Conclusion

Given the data constraints, we were able to correctly predict/classify the buy-sell decisions of the stock market experts.

Using neural networks, the model with *Dataset 1* performed quite robustly as compared to the one having only Apple stocks, *Dataset 2*. The reason being that the second model contained only one type of stock and less amount of data points. Hence it was not able to incorporate the intricacies which are present in the first model due to multiple stocks, limited number of investors and huge data points. This shows that buy-sell predictions are more dependent upon the investors complete portfolio, and not only on the market index and stock price.

Using Random Forest, the results are same for *Dataset 1*, but they are quite different and better for *Dataset 2*. The accuracy of 90% on *Dataset 2* is a bit unusual, but the reasons for this are the same as mentioned above. Hence we think the models for *Dataset 1* are quite robust and can be further improved to readily use in their day to day stock analysis and market trading decisions by individuals.

Next steps would be to incorporate more variables and more data. Do time series cross-validations [4] to tune the parameters for more accuracy. Further we want to develop a framework using this model which does virtual trading and learns on it's mistakes by itself. One potential approach we identified for this is the by using Q-Learning and Recurrent Reinforcement Learning.

5. References

[1] Andreas Humpe, Peter Macmillan, *Non-linear predictability of stock market returns: Comparative evidence from Japan and the US*. Investment Management and Financial Innovations, Volume 11, Issue 4, 2014.

[2] Luckyson Khaidem, Snehanshu Saha, Sudeepa Roy Dey, *Predicting the direction of stock market prices using random forest*. Applied Mathematical Finance 2016.

[3] Diederik P. Kingma, Jimmy Ba, *Adam: A Method for Stochastic Optimization*. 3rd International Conference for Learning Representations, San Diego 2015.

[4] Courtney Cochrane, *Time Series Nested Cross-Validation* Medium, 2018.

[5] Web Scraping in R, *https://ropensci.org/tutorials/rselenium_tutorial*

[6] Keras, *https://keras.io/*