**McMaster University**

**DeGroote School of Business**

**Big Data in Finance**

**F743**

**Adeel Mahmood**

**Big Data Project**

Saad Ahmed

Farhan Ahtisham

Richard Jalonen

Maaz Khan

Deeban Rex

Sangeet Saurabh

**December 10, 2018**

**Problem Description**:

The Efficient Market Hypothesis is an investment theory that states the stock market is unpredictable and cannot be anticipated. However, investment banks and financial firms have become multibillion-dollar behemoths by making such market predictions. The act of predicting stock prices by using historical data is called Technical Analysis. Financial firms employ Quantitative Traders who conduct rigorous Technical Analyses to make educated bets in the stock market. These traders utilize sophisticated machine learning models and combine it with their understanding of the market to generate steady profit streams for their businesses.

In this project, we use a simple algorithm to perform our own Technical Analyses and apply it as an investment strategy. The goal of this project is to utilize different machine learning models and try to predict the daily movement of stock using the most accurate model.

This problem is being solved already by financial firms and if someone is betting without using such tools, they are undertaking a huge risk. Stock market prediction only rewards those who are the quickest to react and investors that jump in later reap much fewer profits.

**Solution Summary**

In essence, the objective of this project is to predict the next day’s stock movement (up or down) given past data. For instance, if we can predict with great accuracy that tomorrow’s price of a particular stock will go up. Then, we can buy that stock today and sell it tomorrow to make a profit on that upwards movement. However, there is a cost per transaction and thus the gain must be enough to cover the cost of trading and have profit left over.

The tool was developed with the Python programming language and heavily relies on the machine learning Library known as Sci-Kit Learn. Anyone can run this code if they meet necessary library requirements.

The software needs these inputs specified to run:

|  |  |
| --- | --- |
| **Input** | **Description** |
| Symbols | All the companies that need to be analyzed can be specified here. The last symbol is always the NASDAQ. For example:  symbols = ['AAPL','AMZN','MSFT','GOOG','IBM','ORCL','INTC', '^IXIC'] |
| Start Date | Specified as “YYYY-MM-DD”. This is the start date for stock data collection for the companies chosen in Symbols variable |
| End Date | Specified as “YYYY-MM-DD”. This is the end date for stock data collection for the companies chosen in Symbols variable |
| Past N Days | How many days in the past should be analysed per Symbol |

Here are the accuracy results of each model for the stock-market movement prediction:

|  |  |
| --- | --- |
| **Model** | **Accuracy Score** |
| *Past N Days* | 5 |
| Logistic regression | 51.01% |
| K-Nearest Neighbour | 56.33 % |
| Naive Bayes | 50.71 % |
| Decision Tree | 62.94 % |
| Gini Random Forest | 63.15 % |
| Entropy Random Forest | 63.40 % |
| ADABOOST | 63.02 % |
| Neural Network | 50.93 % |

All machine learning models give more than 50% chance of predicting the stock market correctly. However, the most accurate model appears to be Random Forest Classifier with entropy criterion that yields an accuracy of 63.40%. In a simple simulation provided at the end of the code, this accuracy of score of 63.40% can yield an ROI of 21% just after 1000 transactions.

**Solution Detail**

**Assumptions**

Here is a list of assumptions that were made for this program.

· Assuming Past data of stocks can predict future prices

· All specified symbols have data available in the chosen timeframe

· Missing average sector data from the day before

· Date is not a factor in prediction

· Inflation of currency is not considered

· Other economic variables are not considered.

**Implementation**

Almost all machine learning models are used except for Support Vector Machines (SVM) as they were too slow to be practical. Since the highest scoring algorithm was random forest (a bagging algorithm), boosting the accuracy with ADABOOST did not yield any fruitful improvement.

First, we gather the daily ‘closing stock price’ of top 43 companies in the tech sector from the beginning of 1995 till September 1st 2018. We also collect the daily ‘closing stock price’ of the NASDAQ for the same time period. There is no missing data as stock prices are guaranteed to exist everyday, thus minimal cleanup was required. Moreover, this is an entirely supervised learning of the stock data. In the final dataset used for machine learning, there are close to 250,000 rows of data that training and testing is performed on. The additional library that is needed aside from Sci Kit Learn is called Data Reader and it is used to fetch stock data.

Next, for each company’s stock that has a higher price than the day before, we assign a ‘Movement’ value of +1. Likewise, for every stock that has a lower price than the day before, we assign a ‘Movement’ value of -1. The ‘Movement’ of the stock’s next day price is what we will try to predict with our testing dataset. To help predict this ‘Movement’, we supply the following independent variables (ones denoted as *computed*).

**Independent variables used for predicting the next stock movement**:

|  |  |  |
| --- | --- | --- |
| ***Variable*** | ***Type*** | ***Description*** |
| *Past N Days* | Input | Number of previous days used to calculate the following variables |
| *Daily Change* | *Computed* | Percent change in company’s stock price from the day before. Computed using:  (new price – last price) / last price |
| *Stock Price Volatility* | *Computed* | The average of Past N Days worth of ‘Daily Changes’.  If Past N days = 5, then  Stock Price Volatility = (yesterday’s ‘Daily Change’+ the day before ‘Daily Change’ + … + Five Day before ‘Daily Change’) / 5 |
| *Stock Momentum* | *Computed* | The average stock movement over Past N Days for a particular company.  If in Past N days = 5, the stock had three ‘up’ movements and two ‘down’ movements, then  Stock Momentum = (1 + 1 + 1 – 1 – 1) / 5 |
| *Index Volatility* | *Computed* | The average of Past N Days worth of ‘Daily Changes’ of  the NASDAQ |
| *Index Momentum* | *Computed* | The average movement over Past N Days for the NASDAQ |
| *Sector Momentum* | *Computed* | The average movement of all stocks in a sector for *the day* |

Here is a sample of the data that shows independent variables (colored in yellow) and the dependent variable (colored as green):

**Results and Recommendation**

The following chart shows the return on investment with varying amount of capital invested using this algorithm.

The algorithm’s ROI improves from loss to a maximum of around 21% for just 1000 transactions. The ROI will become higher as more transactions are undertaken and thus can be scaled. We realize that the accuracy predicted by our models is too high and may not apply to real world due to the underlying assumptions. Thus, area for improvement would be to make this model more realistic by incorporating inflation into the mix amongst other real-world variables. Financial firms are already conduction such technical analyses and here is a glimpse of how to get started.