CS-484

Project Report

Stock Movement Prediction and Trading

Author:
Mathias Wiesbauer

Professor:
Dr. Huzefa Rangwala

Contents

December 11, 2018

Abstract

The goal of this project was to investigate if machine learning can be used to make trading decisions for retirement investing. Many people with retirement accounts utilize a simple buy and hold strategy where a stock is not being actively traded. This strategy can lead to lost opportunities because returns are not maximized by selling high and buying low. It is important to maximize returns to increase the effect of compounding interests. Compounding interests can lead to significant gains in retirement accounts that ultimately determine if somone can retire comfortably or not retire at all. In our example we determined that an increase in returns from 5% to 10% annually will lead to a six fold increase in compound interest. Then we developed a stock movement prediction algorithm that outperformed the buy and hold stragetgy by an average of 6.5% for five stocks accross several industries over the year 2015 during backtesting. In no instance of the tested stocks did the prediction algorithm perform worse than the buy and hold strategy. This demonstrates how many people lose a big opportunity to increase their available funds for retirement.

1 Introduction

This work is not to intended to compete with large investment firms or to support day traders. The methods developed below are not adequate for high frequency trading, if anything this is an extremely low frequency trading system. The intention is rather to demonstrate how a releatively simple machine learning model could help people retire more comfortably.

The motivation for this project stems from how difficult it is to trade stocks and achieve good returns consistently. Making smart trading decisions requires a lot of time in the form of research, knowledge about technical indicators, market behaviour, and fundamental data about the companies and the economy as a whole.

We wanted to investigate if we could outperform the rather simplistic approach of purchasing stock, and never trading the stock again until retirement. This is important because retirement accounts are a long term investement, and need to be properly managed to maximize returns so the to investor can enjoy a life without financial hardships for the duration of his retirement.

According to a stydy from Northwestern Mutual "one in three americans has less than $5,000 in retirement savings." [1]. Saving for retirement is difficult for various reasons, one being the opportunity cost of not having the money at a time when most people are starting families require the money for childcare, and mortgage payments. This further highlights the importance of maximizing returns to make the most of the already limited investment holdings.

Maximizing returns has a tremendous impact on the long term savings in the form of compund interest. If we assume an initial investment at age 30 of $10,000 and continue investing $500 every month for the next 30 years of our working life, everything else being equal, the most important factor are the returns achieved. Table 1 below compares the results after 30 years with varying rates of return. Doubling the annual rate of return from 5% to 10% will approximately triple the available funds for retirement after 30 years, and the factor contributing the most to this development is the compound interest.

Rate of return	Initial Investment	Monthly Investment	Inflation	Compound Interest	Simple Interest	Total after 30 years
5%	$10,000	$500	3%	$112,230	$150,339	$452,569
7%	$10,000	$500	3%	$263.701	$210,454	$664,155
10%	$10,000	$500	3%	$723,532	$300,608	$1,214,140

Table 1: Comparing Retirement Returns

Saving for retirement is important to reduce old-age poverty, and old-age work. We would like to demonstrate a feasible method for investors to increase their returns and have more confidence to have enough funds for their retirement.

2 Problem Statement

The underlying problem is the prediction of the future behaviour of a stock. Predicting the future behaviour of a stock is hard because of the intrinsic volatility of stock [2], the unforcastable news cylce influencing stock performance [3], and most of all the efficient-market hypothesis which states that above average returns are impossible to obtain [4].

According to Manojlovic [4] there are two main methods for stock prediction, one is the prediction of value the other the prediction of movement which is only concerned with predicting whether the price of a stock will fall or rise.

For our problem we would like to reliably predict when the value of a stock will rise, or when the value of a stock will fall in the time period t_n.

Our goal is to predict the direction of movement of the stock price for a time period t_n meaning if we anticipate a decline in stock value then our trading recommendation will be to sell the stock, conversely if we anticipate an increase in stock value we want to recommend a purchase of the stock.

To do this we use patterns within the data and make predictions base on the patterns. This is a classification problem and we use two labels, a buy label (1), and a sell label (-1).

2.1 Notations

t_n	look ahead time period for n days
TA	Technical Analysis, investing based on information of previous prices and trading volumes [5].
TI	Technical indicators

3 Literature Review

Our literature review was focused on the prediction of stock movement using tree based classifiers and papers by Manojlovic [4], O'Connor [6] Basak [2] were considered in this review.

While O'Connor [6] was focused on comparing the performance of different machine learning methods for stock forecasting, namely SVM, Random Forest and ANN. Manojlovic [4] and Basak [2] on the other hand focused on the performance of random forest classifiers to predict stock movement.

All papers mention the Efficient Market Hypothesis, the theory suggests that it is impossible to achieve above average returns because the markets are already pricing all available information into the stock and thereby behaving purely rational, this theory canot be entirely true as O'Connor [6] notes because the theory would not allow for bubbles to occur. The field of behavioral economics studies the impact psychology has on stock prices and according to O'Connor [6] it is the foundation why TA can be used to predict trading patterns.

Using random forest classifiers and adding technical indicators to predict stock movements on the Croation stock exchange seems to be an idea first investigated by Manojlovic [4]. And although regression trees are very common in stock prediction scenarios the random forest classification is less used in the literature to predict stock movements although it performs better than SVM and in line with ANN [6]. O'Connor also found that classification is the superior method to predict stock movement.

When preparing the data for analysis all three papers augment the data with technical indicators althought the amount of technical indicators used varies widely, from six [2] [6] to 12 [4]. The number of estimators used for the random forest also varied from 100 [4] [2] to 500 [6]. Given the difference in the amount of parameters between Manojlovic and O'Connor it was interesting to see that their accuracy was very similar. On the other hand the Results between Manojlovic and Basak were very different even though the parameters were similar with the only exception being the TI, but since Basak used only six TI his increase in accuracy is even more surprising to us. We would like to note at this point that we were only able to reproduce the results from Basak $> 90\% Accuracy$ with a training error. When we computed the trading signals did not shift the trading signals by $-t_n$ so the patterns encountered would match the future returns, once we corrected the error in our model the results obtained were in line with Manoklovic and O'Connor $65\% - 70\% Accuracy$.

4 Methods and Techniques

We decided to use a random forest classification algorithm classifier similar to [4] and [2] for this problem because a tree based classifier provides more information about the important features used for prediction. This is especially helpful if we decided to increase the dimensions by adding more technical indicators. Further we don't have to normalize our data and the low dimensionality of our dataset with a large amount of training data seems to suggest a tree based classifier.

To predict the movement of a stock in the future period t_n we had to compute the training data in the form of buy and sell signals first. We did this by augmenting the data with six different TI. In addition to the TI we also had to compute the return for the period t_n and to set the class labels for each record of the dataset for training and crossvalidation.

Once we had the augmented dataset only the six TI's were used for training with the class labels. To derive the best parameters we used hyper parameter selection.

After implementing following the methodology from Basak et al.[2] we were able to replicate both accuracy and F-score but the prediction accuracy did not directly translate into a trading strategy that was superior to the buy and hold strategy. Upon further investigation we noticed that the results obtained were in error because the recommendations were trained on future values. We corrected for this by shifting the labels by the period after doing that we obtained results similar to O'Connor [6] and Manjolovic [4].

The training parameters were then used in the development of a trading strategy to test the performance. We defined a timewindow of 10 years to train the algorithm and then the algorithm would predict the performance of from the current date to $t_n = 7$ into the future. Thus prediction generates a trading signal of either buy or sell. The time window is then being shifted one day ahead and the procedure repeated for the entire year of 2015. Each time the window is being shifted the all parameters are recalculated and the model retrained to adjust to the most recent data.

5 Discussion and Results

5.1 Datasets

We accuired the dataset by subscribing to a commercial vendor named Tiingo, they offer an API through which the daily cost prices can be acquired programatically for any symbol traded on the US stock exchanges. The data included the following fields as shown in Figure 1.

Metric Name	JSON Code	Type	Description
Date	`date`	date	The date this data pertains to
Open	`open`	float	The actual (not adjusted) open price of the asset on the specific date
High	`high`	float	The actual (not adjusted) high price of the asset on the specific date
Low	`low`	float	The actual (not adjusted) low price of the asset on the specific date
Close	`close`	float	The actual (not adjusted) closing price of the asset on the specific date
Volume	`volume`	int	The actual (not adjusted) number of shares traded during the day
Adjusted Open	`adjOpen`	float	The adjusted opening price of the asset on the specific date. Returns null if not available.
Adjusted High	`adjHigh`	float	The adjusted high price of the asset on the specific date. Returns null if not available.
Adjusted Low	`adjLow`	float	The adjusted low price of the asset on the specific date. Returns null if not available.
Adjusted Close	`adjClose`	float	The adjusted close price of the asset on the specific date. Returns null if not available.
Adjusted Volume	`adjVolume`	int	The adjusted number of shares traded during the day - adjusted for splits. Returns null if not available
Dividend	`divCash`	float	The dividend paid out on "date" (note that "date" will be the "exDate" for the dividend)
Split Factor	`splitFactor`	float	A factor used when a company splits or reverse splits. On days where there is ONLY a split (no dividend payment), you can calculate the adjusted close as follows: `adjClose = "Previous Close"/splitFactor`

Figure 1: Data Fields

The resolution of the time series dataset was one record for each trading day. All fields contained continues values with the exception of the date, volume and adjusted volume fields. The nature of the data is such

that it contains a lot of noise, however we did not apply any cleaning or normalization to the data. For each training instance we downloaded the past 10 years of daily closing prices, and calculated the exponential weighted moving average (EWMA) to smooth the data and reduce outliers as shown in Figure 2.

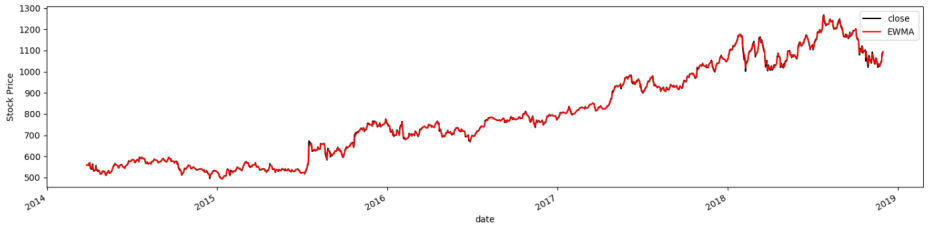

Figure 2: Exponential Weighted Moving Average

The EWMA was then used as the basis to calculate the following technical indicators all of which are momentum indicators as suggested by [2]. Some of them are shown in Figure 3.

- Relative Strength Index (RSI)
- Stochastic oscillator (SO)
- Williams percentage range (WR)
- Moving Average Convergence Divergence (MACD)
- Price Rate of Change (PROC)
- On Balance Volume (OBV)

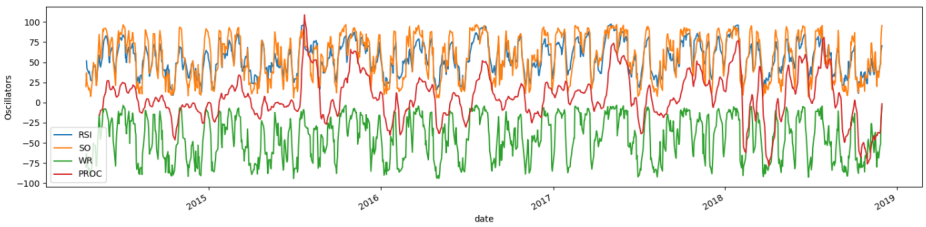

Figure 3: Momentum Indicators

After the TI were callculated we computed the return for the period t_n and added a signal column indicating the stock direction.

5.2 Evaluation Metrics

To evaluate the performance of the machine learning model we use a two phased approach, the first phase was to look at the results from crossvalidation to verify the theoretical prediction performance of the trained model namely the F-Score, and the feature importance. The obtained F-Score was at around 64% for $t_n = 7$ and the feature importance for our TI was almost equally distributed.

The second phase was to compare the obtained prediction results with a baseline. In our case the baseline is a buy and hold trading strategy. We compared the returns achieved with the buy and hold strategy with the returns achieved with our prediction strategy in Figure 1.

5.3 Experimental Results

Model Training When training the model we looked at the F-score and the distribution of the feature importance of our technical indicators used for training. We ran a hyper parameter search and determined the number of estimators to be 100, the criterion used is gini, and the minimum number of samples per leave is 5 to prevent overfitting.

For crossvalidation we used a 1/6 split and stratification because the data is always a little bit skewed with an approximate 60:40 split between rising to falling stock.

Backtesting We tested the model on historical stock data by training the model on the period from the last ten years to the current day and predicting the movement for the next seven days.

We then moved our training and prediction window by one day and repeated the process.

Company	Symbol	Strategy	Buy Date 2015-01-02	Sell Date 2015-12-31	% Return	Improvement
Apple	AAPL	Buy/Hold	$4,996	$4,892	-2,1%	13.1%
		Prediction	$4,996	$5,528	11.0%	
Abbot Labs	ABT	Buy/Hold	$4,981	$5,089	2.2%	5.8%
		Prediction	$4,981	$5,398	8.0%	
American Express	AXP	Buy/Hold	$4,978	$3,773	-24.2%	0.2%
		Prediction	$4,978	$3,787	-24.0%	
Caterpillar	CAT	Buy/Hold	$4,925	$3,778	-23.3	3.3%
		Prediction	$4,925	$4,000	-20.0	
CBS Corp.	CBS	Buy/Hold	$4,975	$4,329	-13.0%	10.0%
		Prediction	$4,975	$4,871	-3.0%	
J.C. Penney	JCP	Buy/Hold	$4,994	$5,288	5.9%	7.1%
		Prediction	$4,994	$5,656	13.0%	

6 Conclusion

Machine learning is a powerfull tool that can help a non professsional investor make trading decisions. The backtesting results have shown that the achieved return with machine learning is far from predictable although in all instances better than the buy and hold strategy. We therefore think that the application of machine learning to aid in long term trading decisions especially in the application of retirment portfolios is advantageous. It was a more difficult problem to tackle than initially anticipated therfore prompting us to reduce the scope from the initially envisioned portfolio recommendation system. The biggest difficulty was not the achievment of good prediction accuracy buy the transfer and application of the learned model as a trading strategy with good performance. This shows that there is a gap from a theoretical model to its practical application that has to be bridged.

6.1 Directions for Future Work

Although the results are encuraging there are ideas we found in the literature that could be investigated further. We would like to see the addition of a third class neutral and how it would impact the trading stratgey. We have seen some trades executed with the current model that did not lead to significant gains. A neutral class might be able to optimize trades that do not yield satisfying returns.

The addition of more technical indicators and their impact on the model and the backtesting results would be another interesting avenue to explore. We have seen models with up to 12 technical inidcators in the literature. However these models did not show signifacnt improvement in terms of prediction accuracy.

We have seen in our experiements that some trades result in a loss, it would be interesting to explore if adding a regressive model to predict the anticipated change in value would including the expected probability would allow to fine tune the trades and increase returns by fusing both models.

References

[1] *1 In 3 Americans Have Less Than $5,000 In Retirement Savings*. Newsroom | Northwestern Mutual. URL: https://news.northwesternmutual.com/2018-05-08-1-In-3-Americans-Have-Less-Than-5-000-In-Retirement-Savings (visited on 12/08/2018).

[2] Suryoday Basak et al. "Predicting the direction of stock market prices using tree-based classifiers". In: *North American Journal of Economics and Finance* (2018), <xocs:firstpage xmlns:xocs=""/>. ISSN: 1062-9408. DOI: 10.1016/j.najef.2018.06.013.

[3] Shihao Gu, Bryan T. Kelly, and Dacheng Xiu. *Empirical Asset Pricing via Machine Learning*. SSRN Scholarly Paper ID 3159577. Rochester, NY: Social Science Research Network, June 11, 2018. URL: https://papers.ssrn.com/abstract=3159577 (visited on 12/08/2018).

[4] T. Manojlovi and I. tajduhar. "Predicting stock market trends using random forests: A sample of the Zagreb stock exchange". In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). May 2015, pp. 1189–1193. DOI: 10.1109/MIPRO.2015.7160456.

[5] Osi Momoh. *Stock Analysis*. Investopedia. May 27, 2010. URL: https://www.investopedia.com/terms/s/stock-analysis.asp (visited on 10/28/2018).

[6] William O'Connor. "Classification and time series forecasting: Applications in the stock market". PhD thesis. ProQuest Dissertations Publishing, 2016. URL: http://search.proquest.com/docview/1798009220/?pq-origsite=primo (visited on 12/11/2018).