Money Maker – An AI enabled investment portfolio manager  
CS7IS2 Project (2019-2020)

Abhishek Verma, Adhishwar Singh Mittal, Nishant Mohan

[vermaab@tcd.ie](mailto:vermaab@tcd.ie), [amittal@tcd.ie](mailto:amittal@tcd.ie), [mohanni@tcd.ie](mailto:mohanni@tcd.ie)

**Abstract.** Investing in stocks is a great way to set aside money while you are busy with life and make that money work for you so that you can reap benefits out of that investment. The question is how to choose which stocks to invest in? With the advent of machine learning we are able to make informed prediction from historical data about the future state of a stock. This is what motivated us to create an intelligent system that strategizes our investments over a group of stocks to get the maximum benefit. We compare 4 scenarios in this paper with different investment strategies that we implemented and tested in a virtual trading environment.

This document is a guideline for writing the final report for the CS7IS2 module *Artificial Intelligence*. You should follow its general structure as shown below.

You should not change its format (font, size, margin, space, etc.).

Report that not comply to the format or exceed the maximum length will be penalised (-5 marks).

Brevity is desirable in communication, however you should provide all those details necessary for the good understanding of the described methods and algorithms.

The report will be graded on the basis of:

* Originality;
* Technical soundness;
* Organisation;
* Clarity of presentation
* Adequacy of bibliography/Results (this last point strongly depends on the type of report)

**Your report should provide a survey and an experimental comparison of multiple solution approaches to a particular problem. This is a critical review of at least three papers that significantly contributed to advance the state-of-the-art for the problem you are analysing. It should not be a mere summary of the papers. You are expected to conduct an analytical review of the methods under analysis to try to find common aspect and differences, connections between methods, drawbacks and open problems. Unless the faced problem has emerged recently, students should choose their papers by diversifying the range of approaches used to solve the problem. A good guideline could be to choose a paper from a decade or two ago, and a couple of more recent papers. You need to experimentally evaluate approaches in a simulation of a problem, in a range of scenarios, and analyse the pros and cons of each approach.**

1 Introduction

Stock markets are highly volatile. While the GDP growth of many countries remain positive, they have substantial fluctuations in their stock markets. There is no single method to accurately predict a stock’s value because the value of a stock is a function of numerous variables. While there have been many attempts [3,4,5,6,7] at prediction using sentiment analysis based on news reports, annual reports and time series data based on previous trends etc., there is no single go-to solution. Stock prediction is a hard problem because one needs to know the relationship between all financial assets as well as the link between assets and the economy.

The Automated Regression Integrated Moving Average (ARIMA) model has been the leading approach to make short term predictions in financial domain. ARIMA is an amalgam of Auto-Regression and Moving Average (ARMA) which makes it a statistical approach for forecasting. The functioning of ARIMA is built on the design that consecutive values of a time series are greatly dependent, successive value can be expressed as linear combination of previous values and errors:

(1)

Where is the actual value and is the random error at t, and are the coefficients, p and q are integers that are often referred to as autoregressive and moving average, respectively.

ARIMA predicts the value based on its three principal parameters (p,d,q), where p is the lag observations included in the model, d is differencing degree and q is the moving average window. In order to get optimum values of p & q, two different correlation plots are drawn- auto-correlation function (ACF) and Partial Auto Correlation function (PACF). The ACF provides moving average order, it shows how effectively current value from the series relates with its previous values. The PACF discards irrelevant features since it fetches correlation between residuals and next value of the series.

In practical applications, historical data of stocks forms a non-stationary time series because stock prices are prone to fluctuation. Forecasting such non-stationary time series without transforming into stationary time series can be relatively cumbersome as stationary time series has constant statistical properties such as average and variance. ARIMA works well with stationary time series. However, this pre-processing part can take a lot of time to determine values of p,q and transform the data in adequate format. Fortunately, Auto-ARIMA obviates laborious part of the method by taking care of these three steps.

Even when the stock value can be predicted with moderate success, we still need to build up a portfolio of investments so that maximum possible financial gains are achieved. Building a robust model to accurately predict a stock’s price is only half the work done. The next stage is building a portfolio of stocks to invest in, so that the financial gains can be maximized. Portfolio theory was laid initially by Markowitz [1], wherein the author proposed mean-variance model.

The model is given as:

Minimize (2)

Subject to (3)

0≤≤1 (4)

(5)

Where Xi are portfolio weights or investment proportions, are the expected return of stock i. Equation 2 is an objective function to minimize risk of the portfolio represented by variance of a portfolio. Equation 2 guarantees desired return b of a portfolio. Equation 3 conveys that the model is for purchase trades only. Equation 4 guarantees total resource allocation. There has been substantive research on improving the search for best stocks for the portfolio [10,2,9], we discuss some of these in the next section.

The main motivation behind this article is to provide a starting point for the beginners in stock market investment. As such, we explain the concepts and algorithms used in detail. In the present work, we demonstrate a stock portfolio management system wherein we first identify a set of stocks to invest in, and then to optimize the amount to be invested in each of them for maximum financial gains. We first employ an Auto-ARIMA model on a set of 400 stocks to predict the best performing ones. Next, we build up a portfolio of stocks to invest in. We demonstrate and compare four algorithms that select which stocks make up to the portfolio each day as well as their proportion in the total amount invested:

1. Reward Maximization
2. Risk Minimization
3. Reward Maximization and Risk Minimization
4. Hold-out with Risk Minimization

The organization of the paper is as follows. Section 2 discusses several previous research applying ARIMA models, and search algorithms employed for portfolio selection. Section 3 defines the problem and discusses the search algorithms employed. We discuss the results in section 4 and present our conclusions in section 5.

2 Related Work

The problem of stock prediction has been studied by many researchers in the past. As mentioned earlier, ARIMA and its combination with other methods has been previously applied in order to improve the predictions. Mondal et. al. [4] realised that only predicting the value may not be sufficient, rather the accuracy of prediction which depends on the model as well as parameters used in the model are also important in establishing the utility of a model. They made separate categories of stocks based on the sector of the company to measure the model accuracy on. Akaike Information Criteria (AIC) is a generally acceptable measure for the goodness of fit of a statistical measure; lower the AIC, better the model. The authors use AIC with correction for finite sample sizes in their research. The authors prove empirically that the model’s accuracy differs in different sectors. As such, different models are needed for different sectors.

Adebiyi et. al. [3] compared the performance of ARIMA and ANN architectures in a stock value prediction task. They concluded that the performance of both models was good and forecasting error for both the models was quite low. ANNs outperformed ARIMA by a small degree. Also, the forecast of ARIMA was directional while that of ANNs was towards value forecasting. Despite better performance of ANNs, the authors did spend more time towards fine-tuning the model and parameters of ANNs as compared to ARIMA. Therefore, in a real-life application, the choice of model would be a function of cost and resources since the difference in reward, which is stock value prediction accuracy in this application, is small.

The second task, which is searching and selecting best stocks for a portfolio has also been extensively studied. While this work was pioneered by Markowitz [1], several limitations were identified. One is that proportion of stocks to be bought is concentrated towards few stocks only. Another limitation is due to the tendency of model to avoid extreme variances, which leads to restrain from low and high value returns at the same time.

Owing to these limitations, there have been several researches on other methods of portfolio selection. These are broadly categorized into single-objective and multi-objective models. Single-objective models can either minimize the risk or maximize the profit, while multi-objective models can take care of both, risk and profit, at the same time [8].

A wide range of technical analysis have asserted that historical market data draw some recurring patterns. The Patterns are useful in achieving the risk-free profits. These methods are already accepted as a body of active portfolio management. Modern state Portfolio theory confirms that without risk profits cannot be made, so going for maximum returns for specified risk level seems to be convincing. It doesn’t leave any place for human emotion for making investment decision. For instance, Bitvai et.al. [10] used this kind of greedy approach for the prediction of stocks followed by investments for the historical data of several UK companies and results were strong enough to surpass the well-known baselines.

Armañanzas et. al. [2] applied three different algorithms to the multiobjective problem- greedy approach, simulated annealing and ant colony optimization. In greedy approach, the neighbors of an initial random solution are visited and selected if one of them is better than the previous one. However, the disadvantage of this approach is that the solution may fetch a locally optimal solution. Simulated annealing mimics the mechanical process of annealing a solid in order to reduce the tensions or energy within the solid. Therefore, just as a particle within a solid being annealed tries to find a position of minimum energy when we vary the temperature of solid, this algorithm tries to find an optimal neighboring position that minimizes the objective function of this approach and reaches an optimum point by varying a search control parameter. The ant colony optimization algorithm is wherein many agents act as ants and build up a solution or a part of a solution, while taking into consideration the current global state of the task at the same time. The authors use this method and employ three agents, one to maximize the profit, second to minimize the risk and the third balancing the trade-off between them, to identify the best strategy for portfolio selection.

Another class of algorithms known as Genetic Algorithms take evolutionary process into account to throw a fittest solution to solve the problem. At first step an encoding structure is constructed to show solution as a chromosome. Genetic Algorithms have fitness function to have a count on goodness of chromosome. To set up new generation selection, crossover and mutations are three genetic operators which are implemented over the population.GA finds a solution region considering a population of individuals which helps them not to stick in local optimums .But that comes at cost of computational time, making GAs slower than many other methods. General GAs are not capable of resolving the combination optimization problem like portfolio problem which has multiple variants made of security (risk) and trading (investment) rule. For tackling combination optimization problem, combination GA with newly designed encoding and genetic operators are required. Combination GA algorithms outdo the uniform allocation methods and show their effectiveness [9].

Therefore, we see that many different approaches have been employed to solve the same problem. Based on broad level characteristics of these algorithms, we select four algorithms to compare the performance after stock value prediction using Auto-ARIMA.

3 Problem Definition and Algorithm

The stocks data sourced from the yahoo finance API provides us with periodic information about each stock’s closing price. The problem is how to use this information to choose a group of stocks to invest in just like we do in real life after analysing the behaviour of the stocks over time. The decision we take here are based broadly on two criteria – Risk and Reward. We need to make trade-offs amongst these two metrics to make a call as to where we must invest or withdraw to maintain a portfolio of investments that over longer durations of time reward us with growth in the dividends.

For the sake of problem definition, we make a few assumptions:

* The data from Yahoo Finance is reliable
* We can invest in fractional stock units
* The investment is minute such that it itself does not impact the stock price
* We can invest among different exchanges interchangeably
* We can trade free of charge any number of times

**3.1 State Space Design**

For the purpose of this problem we define a space with 400 randomly chosen stocks. We define the portfolio to have at max 5 stocks chosen to be invested in at any particular point in time. The ledger will therefore have 5 stocks and a reserve balance that we named – “pocketmoney”. The reason of the reserve is to withdraw from the market in case none of the stocks show promising results.

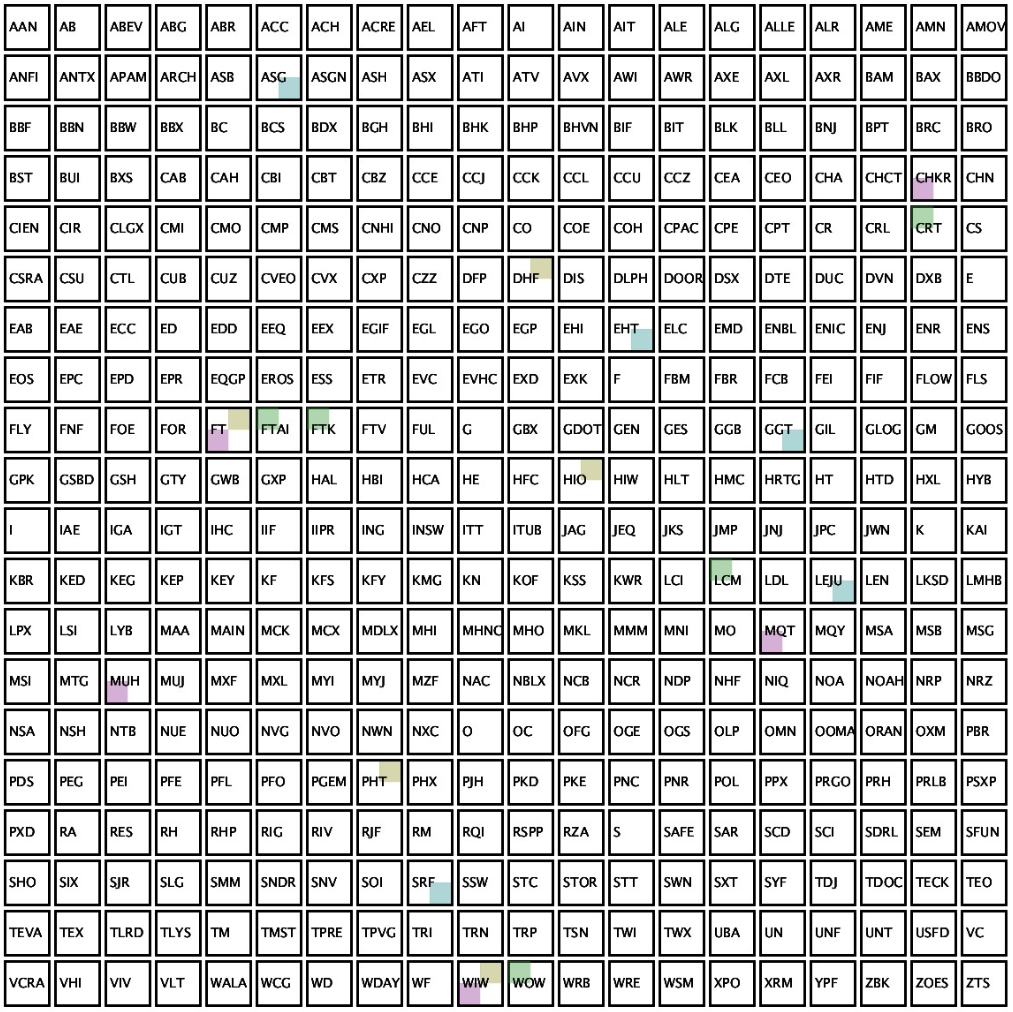


Figure 1: State Space

Figure 1 shows all the 400 stocks present in the state space. Since, we have 4 different investing techniques, we have 4 different ledgers and therefore, each stock banner is broken into 4 equal parts. The 4 parts show which of the 4 agents have chosen that stock in the current state where each agent is denoted by a different color.



Figure 2: Ledger states for different scenarios

Figure 2 denotes the results of 4 scenarios or 4 different algorithms used to choose the best stocks and make the appropriate investment. The 5 stocks are the stocks chosen in the order of highest likeability to result in reward as per the algorithm. The unit price is the price of the stock at the current state and the investment is the amount chosen to be invested by the system.

Before selecting the stocks the system uses the historical stock price data and fits an ARIMA [ ] forecast model to predict the future price. The ARIMA parameters (p,q,d) are chosen using an automated randomised search algorithm implemented to come up with the best model using the Akaike Information Criteria (AIC) which is also used as a measure about the possible error in the model which is translated into the Risk metric.

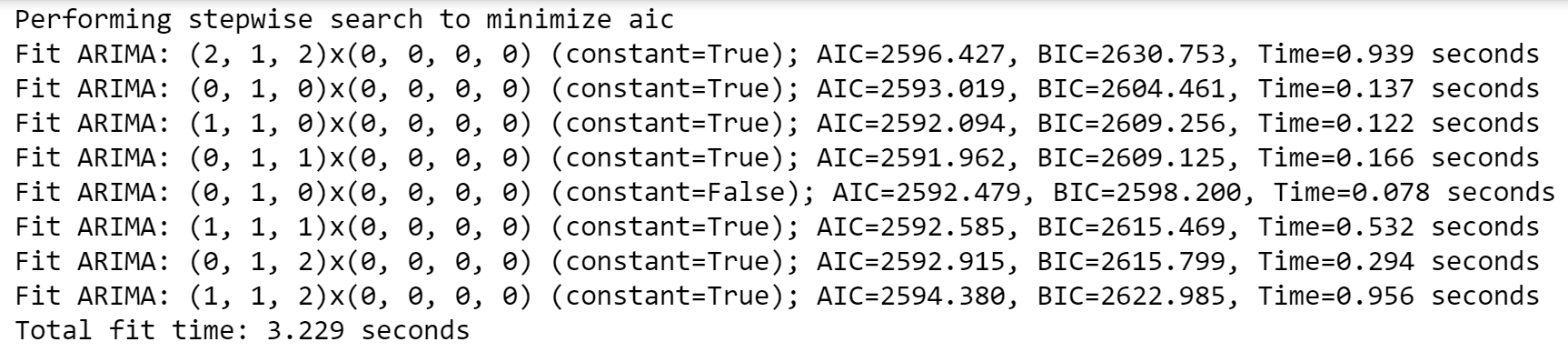


Figure 3: ARIMA randomized search for optimal parameters

Figure 3 shows an example result space for the randomized search for model’s optimal parameters for one of the stocks at a certain point in time. The model is selected on the basis of the minimum AIC value instead of BIC as BIC assumes that the true model is in the candidate set which is asymptotically less optimal while AIC makes no such assumption.

**3.2 Evaluation metric design -**

With the help of the selected ARIMA model we have an estimate of the next day’s prediction and it’s confidence level or the goodness of fit – AIC. We used them to define the two key metrics used to make the choice of which stocks to invest in and the amount to be invested in each chosen stock.

The decision metrics can be defined as –

1. **Risk**: Although the risk as a broader term is a function of the amount invested in the stock and the possibility of the reward, for the sake of the algorithm we simplify the metric to be independent of the amount by choosing a binary change metric in the amount of investment. We choose whether to make the investment or not instead of choosing the amount to be invested. Therefore, the risk is either maximum or minimum based on whether any amount was invested, thereby making the risk a function of only the confidence interval or the Akaike Information Criteria. As the AIC follows a distribution equally skewed on both sides of the mean we can simply use the normalized value for stock comparison on the basis of the forecast AIC to choose between stocks with highest probability of correct predictions.

Risk = AIC

1. **Reward**: The reward is defined as the possible increase in the invested amounts as a result of increase in the stock price. So, for the problem at hand we define it as the percentage increase in the unit price of the stock.

Reward = [(StockPricei**/**StockPricei-1) – 1] \* 100

Since the percentage increase directly determines the dividends earned or losses incurred, we can use it to compare stocks to pick the stocks with the maximum earnings.

These metrics are used by the agent to make the decisions in each algorithm.

**3.3 AI Agent**

Any AI algorithm needs an autonomous entity that looks at the states to decide on the next steps on the basis of pre-defined metrics. In the problem at hand we define the agent that looks at the decision data produced by the forecast model and interprets the risk and reward metrics as defined above to make a call on the investment strategy. The specific agent in our system decides which are the best stocks and how much to invest in the selected stocks. Since there are 4 scenarios, there are 4 different type of progression methodologies used by the agent as defined in the next section.

**3.4 Progression Methodologies**

The 4 broad categories of algorithm implemented in this work are –

**3.4.1. Reward Maximization**

This algorithm can be thought of as the greedy approach keeping only the reward metric as the decision data. The algorithm can be summarized as

*for(i in date\_range){*

*for(i in stocks\_list){*

*predict next day stock price;*

*calculate AIC;*

*calculate price percentage difference;*

*}*

*select top 5 stocks by maximum percentage difference;*

*if(percentage difference > 0){*

*invest amount = balance/portfolio\_size in each stock;*

*}*

**Note:** Any extra balance if not invested is stored in the reserve i.e. PocketMoney which is independent of any changes in the stock price values.

**3.4.2 Risk Minimization**

This algorithm picks the stocks with the minimum risk or AIC values for the forecast and invests in the stock if the percentage difference in unit price values are more than 0. The algorithm can be summarized as –

*for(i in date\_range){*

*for(i in stocks\_list){*

*predict next day stock price;*

*calculate AIC;*

*calculate price percentage difference;*

*}*

*select top 5 stocks by minimum AIC if percentage difference > 0;*

*invest amount = balance/portfolio\_size in each stock;*

**Note:** The algorithm takes any stocks out of the 400 which meet the above criteria and since there are no other limitations no amount will be held outside that iteration of trade in the PocketMoney.

**3.4.3 Multi-objective optimization**

The problem of finding the best stock to maximise the earnings are a function of both risk and reward. So here we scalarise the problem into a single objective function such that we get multiple pareto optimal points which negotiate the possibility of rewards. A scalarising function for the 2-dimensional features vector in the decision data is calculated as the combined ranks in risk and reward priority tables. The algorithm can be summarized as follows –

*for(i in date\_range){*

*for(i in stocks\_list){*

*predict next day stock price;*

*calculate AIC;*

*calculate price percentage difference;*

*}*

*calculate combined ranks;*

*if(percentage difference > 0){*

*select top 5 stocks by combined ranks;*

*invest amount = balance/portfolio\_size in each stock;*

*}*

**Note:** The program doesn’t hold out any balance in the PocketMoney as long as the criteria are fulfilled by any 5 stock.

**3.4.4. Hold-out with Risk Minimization**

The simple yet effective extension of the “Buy low – sell high” methodology is the hold out method where we risk more conservatively and expand the stock balance more steadily by holding out any earnings in the game and investing only the initial amount the agent gets right at the beginning of the progression. For example, if the agent is provided a 1000 dollars at the beginning of the 1st iteration or day and it earns 5 dollars by the end of that day, it will hold the extra 5 dollars in the PocketMoney and invest only the 1000 dollars it had at the beginning. In case the agent incurs a loss of 5 dollars and has only 995 dollars to invest the next day it will invest all the 995 dollars. This implements the risk minimization paradigm considering both the decision of investing as well as the amount of investing at any point of time. By investing less the risk of loosing reduces along with the obvious reduction in probability of higher reward.

*for(i in date\_range){*

*for(i in stocks\_list){*

*predict next day stock price;*

*calculate AIC;*

*calculate price percentage difference;*

*}*

*if(percentage difference > 0){*

*select top 5 stocks by AIC;*

*calculate investment as min(balance,initial\_balance);*

*invest amount = investment/portfolio\_size in each stock;*

*}*

*calculate extra balance as balance,initial\_balance;*

*if(extra balance > 0){*

*PocketMoney = PocketMoney + extra balance;*

*}*

**3.5. Ledger Update**

The calculation of the earnings is decided by the actual price of the stocks gathered from the next day’s data. The investments are maintained in the ledger file with stock names and the amount of investment along with the price the stock was purchased at. With the new price of the stock, we can calculate the true increase in the stock value to calculate the updated balance. The algorithm for ledger update can be summarized as follows –

*for(i in date\_range){*

*read new data;*

*read previous ledger state;*

*update balance values for true difference;*

*if(scenario = hold-out){*

*store extra balance;*

*}*

*update new balance;*

*if(i == last date){*

*exit chart*

*}*

The program runs after the stock selection methodologies and stores the current state in the ledger file. This file maintains the state space of ledgers as demonstrated in Figure 2.

4 Experimental Results

This section should provide the details of the evaluation. Specifically:

* Methodology: describe the evaluation criteria, the data used during the evaluation, and the methodology followed to perform the evaluation.
* Results: present the results of the experimental evaluation. Graphical data and tables are two common ways to present the results. Also, a comparison with a baseline should be provided.
* Discussion: discuss the implication of the results of the proposed algorithms/models. What are the weakness/strengths of the method(s) compared with the other methods/baseline?

5 Conclusions

Provide a final discussion of the main results and conclusions of the report. Comment on the lesson learnt and possible improvements.

A standard and well formatted bibliography of papers cited in the report. For example:

References

1. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1 (1997) 108–121

2. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Abadi, M., Ito, T. (eds.): Theoretical Aspects of Computer Software. Lecture Notes in Computer Science, Vol. 1281. Springer-Verlag, Berlin Heidelberg New York (1997) 415–438

3. van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. Lecture Notes in Computer Science, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York (1995)

4. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)