Money Maker – An AI enabled Investment Portfolio Manager

CS7IS2 Project (2019-2020)*

Abhishek Verma, Adhishwar Singh Mittal, Nishant Mohan

vermaab@tcd.ie, amittal@tcd.ie, mohanni@tcd.ie

Abstract. Investing in stocks is a great way to set aside money while one is busy with life and make that money work so that one can reap benefits out of that investment. The question is how to choose which stocks to invest in? With the advancements in machine learning we can now make informed prediction from historical data about the future state of a stock. We present an intelligent system that strategizes investments over a group of stocks to get the maximum benefit. We compare four algorithms in this paper with different investment strategies that we implement and test in a virtual trading environment.

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 [1,2,3,4,5] at prediction using sentiment analysis based on news reports, annual reports and time series data based on previous trends, 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 Auto-Regressive 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. ARIMA assumes that consecutive values of a time series dependent on historical values. So, successive value can be expressed as linear combination of previous values and errors:

$$\mathrm{Y_t} = \emptyset_0 + \emptyset_1 \mathrm{Y_{t-1}} + \emptyset_2 \mathrm{Y_{t-2}} + \ldots + \emptyset_\mathrm{p} \mathrm{Y_{t-p}} + \varepsilon_\mathrm{t} - \theta_1 \varepsilon_{\mathrm{t-1}} - \theta_2 \varepsilon_{\mathrm{t-2}} - \ldots - \theta_\mathrm{q} \varepsilon_{\mathrm{t-q}} \quad (1)$$

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

---

* https://git.io/JvhX4

ARIMA predicts the value based on its three principal parameters (p,d,q), where p is the lag observations, d is differencing degree and q is the moving average window.

Predicting non-stationary time-series such as stocks is tough as this data has varying statistical properties such as average and variance. ARIMA requires the data to be stationary, as such implementing Auto-ARIMA obviates the need to determine the parameters p and q. It is preferred that model which fits well should be obtained by compromising on number of parameters; information criterion is something which fulfils this condition. Two most common ones are Akaike information criterion (AIC) and Bayesian information criterion (BIC), differing only in a penalty term. Burnham and Anderson have given empirical arguments in favour of using AIC over BIC. [6]

Even when the stock value can be predicted with moderate success, we still need to build a portfolio of investments for maximum possible financial gains while balancing risk. 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 [7], wherein the author proposed mean-variance model. The model is given as

Minimize

$$\sum_{\mathrm{i}=1}^{\mathrm{N}} \sum_{\mathrm{j}=1}^{\mathrm{N}} X_i X_{\mathrm{j}} \sigma_{i\mathrm{j}} \tag{2}$$

Subject to

$$\sum_{\mathrm{i}=1}^{\mathrm{N}} X_{\mathrm{i}} \mu_i > \beta \tag{3}$$

$$0 \leq X_{\mathrm{j}} \leq 1 \tag{4}$$

$$\sum_{\mathrm{i}=1}^{\mathrm{N}} X_{\mathrm{i}} = 1, \mathrm{i} = 1,2,3 \ldots, \mathrm{N} \tag{5}$$

Here X_{i} are investment proportions, and the expected return from stock i is μ_i. Equation 2 is the objective aiming at minimizing risk of the portfolio, variance of each stock being σ_i. We expect a return β from the portfolio, Equation 3 guarantees the same. Equation 4 makes sure that the model only buys. Total resource allocation is guaranteed by Equation 5. There has been substantive research on improving the search for best stocks for the portfolio [8,9,10], 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 intelligently optimize the amount to be invested in each of them for maximum gains. We first employ an Auto-ARIMA model on a set of 400 stocks to predict the best performing ones. Next, we build a portfolio of stocks to invest in. We demonstrate and compare four algorithms that select which stocks make it to the portfolio each day as well as their proportion in the total amount invested:

1. Profit Maximization
2. Risk Minimization

3. Multi-Objective Optimization
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. As mentioned earlier, ARIMA and its combination with other methods has been previously applied in order to improve the predictions. Mondal et. al. [2] realized that only predicting the value may not be enough, 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 accuracy on. As such, different models are needed for each sector.

Adebiyi et. al. [1] compared the performance of ARIMA and ANN architectures. They concluded that the performance of both models was good and forecasting error for both the models was quite low. ANNs beat ARIMA by a small degree. Also, the forecast of ARIMA was directional while that of ANNs was towards value forecasting. In applications, the choice of model would be a function of cost and resources since the difference in reward may be small as compared to the model complexity.

The second task, which is searching and selecting best stocks for a portfolio was pioneered by Markowitz [7]. However, 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 [8,9,10]. These are broadly categorized into single-objective and multi-objective models. Single-objective models can either minimize the risk or maximize the profit, while latter can take care of both at the same time [11].

Modern portfolio management knowledge confirms that without risk profits cannot be made, therefore going for maximum returns for specified risk level seems to be convincing. It does not leave any place for human emotion for making investment decision. For instance, Bitvai et.al. [8] used this kind of greedy approach for the prediction of stocks followed by investments for the historical data and results were strong enough to surpass the well-known baselines.

Armañanzas et. al. [9] applied three algorithms to the multi-objective problem-greedy approach, simulated annealing and ant colony optimization. In greedy approach, the neighbours of an initial random solution are visited and selected if one of them is better than the previous one. Simulated annealing mimics the mechanical process of annealing a solid in order to reduce the tensions or energy within the solid. Therefore, this algorithm tries to find an optimal neighbouring position that minimizes the objective function and reaches an optimum point by varying a search control parameter. The ant colony optimization 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 (GA) take evolutionary process into account to pitch a fittest solution.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. 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 [10].

Hence, we see that many approaches have been employed to solve the same problem. Based on broad level characteristics of these algorithms, we select four algorithms, as explained in detail in the following sections.

3 Problem Definition and Algorithm

The stocks data sourced from the yahoo finance python API [12] provides us with information about each stock's daily closing price. The problem is how to use this information to choose a group of stocks to invest in just as we do in real life after analysing the behaviour of 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 constraints to make a call for where we must invest or withdraw to maintain a portfolio of investments for maximum dividends over an extended period.

3.1 State Space Design

We define our state space as 400 randomly chosen stocks as shown in Fig.1, and the portfolio to have a maximum of 5 stocks to be invested in at any point in time. The ledger will therefore have 5 stocks and a reserve balance that we name "wallet". The reason of the reserve is to withdraw from the market in case none of the stocks show promising results.

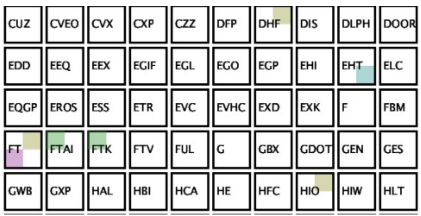

Fig. 1. A snippet of State Space for 50 out of 400 stocks. Letters denote the stock symbol. Color represents the state selected by an algorithm. Each stock banner is broken into four equal parts representing each algorithm. The four parts show which of the four agents/algorithm have chosen that stock in the current state.

3.2 Constraints

With the help of the selected ARIMA model we have an estimate of the next day's prediction and its 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.

2. *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.

$$\text{Reward} = [(\text{StockPrice}^{i}/\text{StockPrice}^{i-1}) - 1] * 100 \tag{6}$$

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 based on 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. We have four different algorithms as discussed below.

3.4 Selection Algorithms

Profit Maximization (PM)

This algorithm is essentially a greedy approach, keeping only the reward metric as the decision data. The algorithm can be summarized as shown below.

```
for(i in date_range){
  for(i in stocks_list){
      predict next day stock price;
      calculate price percentage difference;
    }
  select top 5 stocks by maximum percentage difference;
  if(percentage difference > 0){
    invest amount = balance/portfolio_size in each stock;
  }
}
```

Any extra balance if not invested is stored in the reserve i.e. Wallet, which is independent of any changes in the stock price values.

Risk Minimization (RM)

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 price values are more than 0.

```
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;
}
```

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 Wallet.

Multi-Objective Optimization (PMRM)

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

```
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;
   }
}
```

The program does not hold out any balance in the Wallet, if the criteria are fulfilled by any 5 stocks.

Hold-out with Risk Minimization (HRM)

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 at the beginning of the 1st day and it earns $5 by the end of day, it will hold the extra $5 in the Wallet and invest only the $1000 it had at the beginning. In case the agent incurs a loss of $5 and has only $995 to invest the next day it will invest all the $995. 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 losing 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;
          }
}
```

4 Experimental Results

We make a few assumptions for this study:

- We can invest in fractional stock units
- We can invest among different exchanges interchangeably
- We can trade free of charge any number of times

4.1 Methodology

The system uses the historical stock price data and fits an ARIMA [2] 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 AIC which we use as our risk metric.

```
Performing stepwise search to minimize aic
Fit ARIMA: (2, 1, 2)x(0, 0, 0, 0) (constant=True); AIC=2596.427, BIC=2630.753, Time=0.939 seconds
Fit ARIMA: (0, 1, 0)x(0, 0, 0, 0) (constant=True); AIC=2593.019, BIC=2604.461, Time=0.137 seconds
Fit ARIMA: (1, 1, 0)x(0, 0, 0, 0) (constant=True); AIC=2592.094, BIC=2609.256, Time=0.122 seconds
Fit ARIMA: (0, 1, 1)x(0, 0, 0, 0) (constant=True); AIC=2591.962, BIC=2609.125, Time=0.166 seconds
Fit ARIMA: (0, 1, 0)x(0, 0, 0, 0) (constant=False); AIC=2592.479, BIC=2598.200, Time=0.078 seconds
Fit ARIMA: (1, 1, 1)x(0, 0, 0, 0) (constant=True); AIC=2592.585, BIC=2615.469, Time=0.532 seconds
Fit ARIMA: (0, 1, 2)x(0, 0, 0, 0) (constant=True); AIC=2592.915, BIC=2615.799, Time=0.294 seconds
Fit ARIMA: (1, 1, 2)x(0, 0, 0, 0) (constant=True); AIC=2594.380, BIC=2622.985, Time=0.956 seconds
Total fit time: 3.229 seconds
```

Fig. 2. ARIMA randomized search for optimal parameters

Figure 1 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 based on 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. The model uses rolling data of 2 years and predicts stock values starting January 1st, 2020 for subsequent 2 months, which is our study period.

Next, the four methods of stock selection are used individually to run through the study period. The agent is given an initial seed money of $1000 to invest. 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 shown below.

```
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 Fig.3.

	Stock	Unit Price	Investment	Stock	Unit Price	Investment	
Scenario−1	FTAI	19.600	0.000	DHF	3.110	200.645	Scenario−3
	LCM	650.000	0.000	HIO	5.070	200.000	
	FTK	2.010	0.000	FT	7.800	200.514	
	WOW	7.190	0.000	WIW	11.580	200.693	
	CRT	8.680	0.000	PHT	9.460	198.323	
Scenario−2	FT	7.800	200.514	LEJU	2.019	200.896	Scenario−4
	WIW	11.580	200.693	ASG	6.440	198.154	
	CHKR	0.530	200.000	SRF	6.830	203.274	
	MQT	13.010	200.154	EHT	10.050	199.603	
	MUH	15.180	199.343	GGT	8.040	200.499	

Fig. 3. Ledger states for different algorithms

Figure 3 displays the results of four scenarios or algorithms used to choose the best stocks and make the appropriate investment. The five 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.

4.2 Evaluation

The performance of the algorithms is studied by comparing against each other for the study period. A wallet balance of above $1000 increases our confidence in the algorithm, while below that level decreases the same. A good algorithm is one which keeps improving the wallet balance and is consistent in its performance. Also, it would be aware of the general trend of stock market and adjust its investments accordingly.

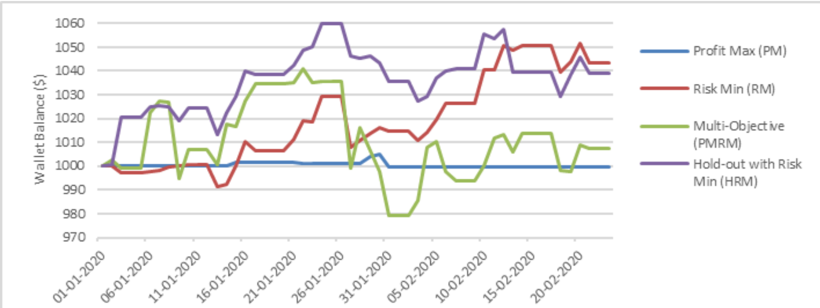

Fig. 4. Progression of Wallet Balance each day during the study period

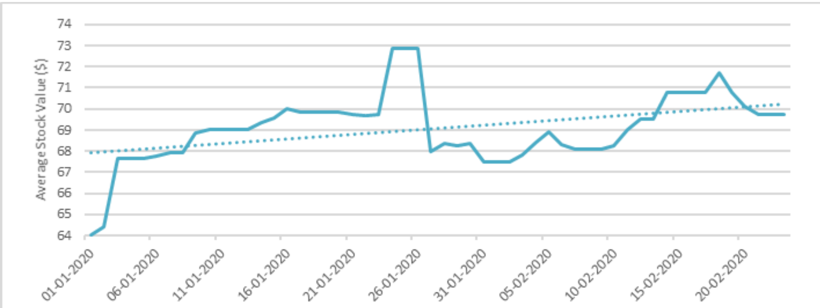

Fig. 5. Average of the 400 randomly selected Stock Value during the study period

4.3 Results and Discussion

Figure 4 shows the progression of wallet balance as the models advance and make prediction for each day. PM evidently performs the worst as it works defensively in order to get a threshold profit initially set by the user and seems to keep the initial investment intact with no losses and no profit. PMRM initially performs good up until the half of the study period. However, the wallet balance falls for the remainder period. The algorithm barely manages to give a positive balance. RM starts slow but eventually picks up and provides good returns. Lastly, HRM emerges as the best algorithm right from the start and manages to keep its dominance as the best performing model. However, it should be noted the RM and HRM come very close towards the end of the study period. As such, it is possible that RM has a gestation period, after which it performs as well as HRM does.

Average Stock Price of all the stocks studied during the study period is plotted in Figure 5 to show the trends in stock market. It can be seen from the trendline, that the market has a very small increment in the average stock value of the 400 randomly selected stocks, with a minimum of $64 to a maximum of $72. Since this increase is very small, it is improbable that the in general positive returns from the four models are simply a result of a positive sentiment in the market. Therefore, we can eliminate the possibility that success of our models is due to the stock market performing good.

5 Conclusions

We present an AI-enabled investment portfolio manager that identifies the best performing stocks from a pool of stocks and invests intelligently. This study presents empirical results from four artificially intelligent algorithms, which aim to maximise returns- Profit Maximization, Risk Minimization, Multi-Objective and Holdout with Risk Minimization. We found that Holdout with Risk Minimization performs the best.

We propose that further work is needed in order to establish the robustness of this method for stock portfolio selection. We intend to improve the prediction of stocks using Long Short-Term Memory Networks. Also, we hope to improve the selection of portfolio by using a greater number of stocks as opposed to the current 400. Features from sentiment analysis can be incorporated to improve the prediction accuracies.

6 References

1. Adebiyi, A. A., Adewumi, A. O., Ayo, C. K.: Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction. J. Applied Math. (2014) 1–7
2. Mondal, P., Shit, L., Goswami, S.: Study of Effectiveness of Time Series Modeling (Arima) in Forecasting Stock Prices. Int. J. Comp. Sci., Eng. and Applications (2014) 4(2), 13–29
3. Nguyen, T. H., Shirai, K., Velcin, J.: Sentiment analysis on social media for stock movement prediction. Expert Systems with Applications (2015) 42(24), 9603–9611
4. Seng, J. L., Yang, H. F.: The association between stock price volatility and financial news – a sentiment analysis approach. Kybernetes (2017) 46(8), 1341–1365
5. Makrehchi, M., Shah, S., Liao, W.: Stock Prediction Using Event-Based Sentiment Analysis. IEEE/WIC/ACM Int. Joint Conf. on WI and IAT (2013)
6. Burnham K., Anderson D.: Multimodel inference: understanding AIC and BIC in model selection. Sociol Methods & Res (2004) 33:261–304
7. Markowitz, H.: Portfolio Selection. The Journal of Finance (1952) 7(1), 77–91
8. Bitvai, Z., Cohn, T.: Day trading profit maximization with multi-task learning and technical analysis. Machine Learning (2014) 101(1-3), 187–209
9. Armanianzas, R., Lozano, J.: A Multiobjective Approach to the Portfolio Optimization Problem. IEEE Congress on Evolutionary Computation (2005)
10. Chen, J. S., Hou, J. L., Wu, S. M., Chang Chien, Y. W.: Constructing investment strategy portfolios by combination genetic algorithms. Expert Sys App (2009) 36(2), 3824–3828
11. Rather, A. M., Sastry, V. N., Agarwal, A.: Stock market prediction and Portfolio selection models: a survey. OPSEARCH (2017) 54(3), 558–579
12. Yahoo Finance Python API, https://pypi.org/project/yahoo-finance/., Last accessed: 07-04-2020