Final Year Project Plan

1. Abstract

- Objective: research ML/AI approaches to predict price of a stock, develop several models, compare and analysis result
- Method summary
- Findings
- Summary

Contents

2. Introduction

3. Literature Review

3.1. Introduction

Due to the potential for financial reward, predicting the changes in the price of a stock is a popular area of research. Models for these predictions have increased in number and complexity over time; from simple models such as moving averages, to early applications of machine learning methods, to complex artificial neural networks. This literature review focuses on the technical side of prediction, using historical data to predict future results. It attempts to highlight the performance of these models and techniques, creating a benchmark for comparison for the models I create.

3.2. Efficient Market Hypothesis

Before looking at the models, it is essential to address whether stock markets can be predicted. Fama (1965) states one hypothesis is the efficient market hypothesis (EHM), which theorises that the price of the stock reflects all available information about it and therefore, forecasting future prices is impossible This hypothesis supports the random walk hypothesis, which suggests that movement in a stock's price is random, dictated by emerging and unpredictable information.

However, there is significant critique to these hypotheses, supporting a semi-strong form of the efficient market hypothesis, that only publicly available data is reflected in the price of a stock (Jensen, M. C.,1978). As such you cannot consistently predict the market with publicly available data, you would need insider knowledge. However, Jensen, M. C's article gives evidence that there are anomalies suggesting that there is a flaw to the semi-strong EHM.

If the markets are not fully efficient, then by analysing publicly available data, which may not be fully reflected in the price of a stock, you can achieve a prediction outperforming random chance, leading to financial gain.

3.3. Statistical models

The most naïve model for predicting the stock, supported by the efficient market hypothesis is to simply use the latest price as the prediction.

A slightly more complex approach involves simple moving averages (SMA). A SMA is the average price of a stock over a given number of days. These are used as indicators of the trend of a stock. If a stock's current price is above the moving average, it is said to be on an upwards trend. Similarly, if the current price is below the moving average, it is said to be on a downwards trend. While moving averages are a good indicator for the trend of a

price, due to their nature, they lag behind the actual price movement, so may not be sufficient for short-term predictions.

A more advanced statistical model is the autoregressive integrated moving average (ARIMA). ARIMA models have been used in a range of industries for time series analysis. Such examples include Irish Inflation (Meyler, A., Kenny, G., and Quinn, T, 1998).

Dell Stock Prediction
Adebiyi, Adewumi, and Ayo (2014) applied ARIMA to predict Dell's stock prices from August 1988 to February 2011. Their model achieved:

- Bayesian Information Criterion (BIC): 4.5588.
- Adjusted R2R^2R2: 0.9897.
- Standard Error of Regression (SER): 2.361.

NYSE and Nigerian Stock Exchange Predictions
Adebiyi, A. A., Adewumi, A. O. and Ayo, C. K (2014) used ARIMA to predict stocks on the New York Stock Exchange and Nigerian Stock Exchange. Their model achieved:

- BIC: 23,736
- R^2: 0.9972
- SER: 0.7872

In 2018, Almasarweh, M. and Al Wadi, S used ARIMA to predict banking stock market data. They used the metric of root mean squared error to compare their models instead and achieved the lowest value of 1.4.

These studies demonstrate the effectiveness ARIMA can have on time series data and by extension stock market predictions. However, it is limited by its linear reliance, failing to map more complex relationships. This is an area machine learning and artificial intelligence models aim to fill.

3.4. Machine Learning Models

Random Forrest

Sadorsky demonstrates the use of random forest to predict the direct of clean energy stocks with an 80% accuracy. They identified that the most important technical indicator was the 200-day moving average and the on-balance volume (OBV)

Support Vector Machine

Meesad, P. and Rasel, R. I. (2013) used support vector regression models to predict the ACI group of company Limited on the Dhaka stock exchange. The used a mean average

percentage error (MAPE) to determine their best models. They achieved a MAPE of 0.04 for '1 day ahead' predictions, 0.15 for '5 days ahead' and 0.22 for '22 days ahead'.

Gradient boosting: TODO

3.5. Artificial Neural Networks (ANN)

Adebiyi, A. A., Adewumi, A. O. and Ayo, C. K (2014) compared their ARIMA model to artificial neural network model, they found that the best model for predicting the price of Dell's stock was three layers in size and consisted of a 10-17-1 architecture. They used the mean squared error and achieved and error of 0.071589 after 5000 epochs

Zhang, G. P. (2003) created a hybrid ARIMA and ANN model, they trained it on sunspots, Canadian lynx data as well as the GPB/USD exchange rate. They used mean squared error and mean absolute deviation to analysis their model over a period of 1, 6 and 12 months, their findings can be seen below.

Table 4
Exchange rate forecasting results[a]

	1 month		6 months		12 months	
	MSE	MAD	MSE	MAD	MSE	MAD
ARIMA	3.68493	0.005016	5.65747	0.0060447	4.52977	0.0053597
ANN	2.76375	0.004218	5.71096	0.0059458	4.52657	0.0052513
Hybrid	2.67259	0.004146	5.65507	0.0058823	4.35907	0.0051212

[a]*Note*: All MSE values should be multiplied by 10^{-5}.

4. Methodology

4.1. Explore Pre-ML/AI methods.

1.1. Introduce early methods.
  1.1.1. Simple Moving average (SMA)
  1.1.2. Exponential smoothing
    1.1.2.1. Like a moving average but with weights applied to the averages following an exponential pattern – further away, less weight.

1.1.3. Relative strength index (RSI)
1.1.4. Autoregressive Integrated Moving average (ARIMA)
1.1.5. (optional) expansions ARIMA models

4.2. Early machine learning techniques

1.1. Random forest (counter to overfitting)
1.2. Support vector machines (finding nonlinear decision boundaries)
1.3. Gradient boosting

4.3 Neural networks

1.2. Hidden layer model
1.3. Larger models
1.4. Deep learning
1.5. Long-short term memory

4.4. Further predictions

1.1. Predicting multiple days
1.1.1. Targeting single X Day in the future
1.1.2. Targeting all days up to X days in the future
1.1.2.1. Single prediction
1.1.2.2. Recursive

4.5. Percentage return using models.

1.1. Single days
1.2. Multiple days

5. Implementation

With python and juypter notebooks using libraries such as pandas, numpy matplotlib, yahoo finance, sklearn.

- Retrieve data with yahoo finance.
- Preprocess with pandas and numpy.
- Create models using sklearn.
- Use sklearn grid search to rain multiple models with varying hyper parameters.
- use the best model with data from mor recent time periods.
- graph predictions vs target with matplotlib.

6. Results and Data Visualization

1. Single Next day prediction
1.1. Compare models using root mean squared error, mean absolute error, and other metrics.

1.2. Compare correct direction prediction.
1.3. Determine percentage return if invested using model.
1.4. Graph actual vs target, error and return on investment.
2. Multiple prediction days
2.1. Compare models' prediction score using average root mean squared error of days.
2.2. Compare average correctly predicted days for each prediction.
2.3. Determine percentage return on investment.
2.4. Graph actual vs target, error and return on investment.

7. Discussion

Comparison with metrics found in literature review.

8. Conclusion

9. References

Adebiyi, A. A., Adewumi, A. O. and Ayo, C. K. (2014) 'Comparison of ARIMA and Artificial Neural Networks models for stock price prediction', *Journal of Applied Mathematics*, 2014, Article 614342.

A. Ariyo, A. O. Adewumi and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, Cambridge, UK, 2014, pp. 106-112, doi: 10.1109/UKSim.2014.67.

Almasarweh, M. and Al Wadi, S. (2018) 'ARIMA model in predicting banking stock market data', *Published by Canadian Center of Science and Education*, pp. 309-312

Ball, R. (1978) 'Anomalies in relationships between securities' yields and yield-surrogates', *Journal of Financial Economics*, 6, pp. 1033-1126. North-Holland Publishing Company.

Fama, E. F. (1965) 'The behaviour of stock market prices', *Journal of Business*, 38(1), pp. 34-105.

Jensen, M. C. (1978) 'Some anomalous evidence regarding market efficiency', *Journal of Financial Economics*, 6, pp. 95-101.

Meesad, P. and Rasel, R. I. (2013) 'Predicting stock market price using support vector regression', *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, Dhaka, Bangladesh, pp. 1-6. doi: 10.1109/ICIEV.2013.6572570.

Meyler, A., Kenny, G., and Quinn, T. (1998) *Forecasting Irish inflation using ARIMA models*, MPRA Paper No. 11359

Sadorsky, P. (2021) 'A random forests approach to predicting clean energy stock prices', *Journal of Risk and Financial Management*, 14(2), p. 48.

What Follows is not apart of the final report draft but a summary of progress, problems, changes in scope, possible improvement, future plans and

Description of Progress

So far, I have created several juypter notebooks that do the following:

1. Import historical stock data from yahoo finance, so far, I have limited myself to apple stock from the past 10 years.
2. Preprocess the data, cleaning missing values and scale the data to remove the effect of larger numbers.
3. Create indicators like moving averages for each day.
4. Select Target for each prediction.
5. Sperate data in training and test data
6. Create models according to specific techniques with different hyperparameters.
7. Train models on training data
8. Test models on test data.
9. Compare error metrics of models.
10. Display graph of best model. Compare predicted vs actual value.
11. Select the best performing model and use it to predict data outside of training and test data scope.
12. Display graph and metrics of results.
13. Display percentage of correctly predicted directions

So far, I have done this for the following types of models:

- Random forest
- Support vector machines.
- Gradient boosting
- Single layer artificial neural networks
- Multiple layer artificial neural networks

I have also created a single layer artificial neural network to predict x number of days in the future and am looking to expand on this more.

The Code can be found in my git hub repo:
https://github.com/KERPalmer/university_final_year_project

Problems and changes in direction

Early in the project I created some notebooks to graph some ARIMA models and simple moving averages, while not entirely useless I will need to figure out how to include them, perhaps pivoting to using them as comparison as well as the literature view material.

At the beginning of this project, I mentioned the idea of creating an app for this project but that was quickly realized as being out of scope and unnecessary.

I had also expressed an interest in using semantic analysis with scraped social media data. If I have time to explore this option, I would like to research it, but I believe I will struggle to get a working model. I must also consider using that time to further improve existing models or expand in other directions.

One major problem I encountered was using different operating systems. Until this year I have always developed using a Mac and Mac OS. Recently I built a windows computer and am using it for the bulk of building and training prediction models. Installing and running these two different operating systems, one of which I was unfamiliar with, proved to be a challenge.

I encountered several issues with installing a version of python compatible with the python version I was using, this significantly ate up into the time I had assigned to the project and did cause a lot of frustration.

One of the major changes in direction that was inspired by my supervisor was to expand the prediction window. I was going to focus on building single day prediction models but have now decided to pivot to include the following ideas.

- Predicting a day further in the future, for example two days after
- Predicting multiple days at once, for example the next 3 days
- Using previous predictions to predict the next day, recursively.

I am hoping that this will make up the bulk of the project and can use any excess time to improve the models.

Revised Work plan

The current work plan is as follows.

Starting from the beginning of February and semester 2:

- Week 1: research gradient boosting and find research materials.
- Week 2-3: expand prediction of multiple days to other model types.
- Week 3-4: use model prediction to create return on investment metric.
- Week 5: compares model performances.
- Week 6: considers and implement improvements.

- Week 7- 9: finalise write up