Utilising Machine Learning for portfolio selection with the aim of out-performing benchmark indices

Github address:

<https://github.com/MCoffey1129/Utilising-Machine-Learning-for-portfolio-selection-with-the-aim-of-out-performing-benchmark-indices>

**Introduction**

A capital market is said to be efficient if it fully and correctly reflects all relevant information in determining security prices……implies that it is impossible to make economic profits by trading based on an information set which is available to all participants (Malkiel B. G. 1989).[1]

The aim of this project is to assess if a 6 month buy and hold trading strategy based on Machine Learning portfolio selection can out-perform the S&P 500 for the same period (the purchase and sale is assumed to occur on the last trading day of January and July each year). The list of stocks used in the project are either traded on the NASDAQ or the NYSE.

Classification models rather than regression models are used to generate stock portfolios. Each of the models will be trained to predict if a stock will increase by 10% or more 6 months from that date. The probabilities generated from the models will be used to generate the stock portfolios i.e. the higher the probability the model thinks a stock price will increase by 10% or more 6 months from that date the more likely the stock will be included in the portfolio.

The following models will be used in this project:

* Random Forest model,
* Stacked model – RF, KNN, Naïve Bayes, Support Vector Machines and XGBoost as the Base models and XGBoost as the meta model.
* Genetic algorithms (using the TPOTClassifier)
* Multiple Neural Network model (average of 3 diverse Neural Networks)
* Combined Model (average of the Stacked model and the Multiple Neural Network model)

Fu et al (2018) found that when looking at Random Forests, Neural Networks, logistic regression found that a stacked version of the models outperformed any individual model [2]. The focus of the project will be to find a combined model which leverages the diverse insights from each of the individual models.

**Modelling approach**



***Figure 1: Modelling approach***

1. **Data Import**

The data used in the model was imported from the Alpha Vantage website via an API key and includes:

* Company\_overview:

Name, Sector and Industry are used while the other elements are deleted as they are forward looking which would positively skew the model

* Earnings, Income\_Statement, Balance Sheet and Cash\_Flow\_statement:

The quarterly reports published by each company is imported into Python

* Time\_Series\_Monthly\_Adjusted:

The monthly adjusted stock price for each stock is imported into Python.

The target for our classification model is “gt\_10pc\_gth” which is 1 if the account’s stock price increased by 10% after 6 months and 0 otherwise.

The data is shaped to match the below such that each stocks information for that date is contained in one row.

Graphical user interface, application

Description automatically generated

***Table 1: High level overview of the data structure used in the assignment***

1. **Feature Engineering**

Large institutional investors tend to use a combination of favourable valuation, profitability, solvency and liquidity ratios combined with forecasted discounted cash-flow models to assess if a stock is undervalued.

Feature engineering was applied to the data to obtain these ratios.

Application

Description automatically generated

***Table 2: High level overview of the data structure used in the assignment (including engineered fields)***

1. **Data Preparation**

**3.1 Train/Test/Deploy split:**



***Figure 2: Train, test and deploy datasets***

* The model is trained on the training data and is evaluated on both the Test and Deploy dataset.
  + The training data consists of 6 trading windows from Jul ’17 to Jan ’19 (c. 75% of the data)
  + The test data for which the training model is tested on consists of one trading window at Jul ’20 (c. 12.5% of the data)
  + The training model is subsequently tested on the deploy dataset to see if the model still performs well 6 months later or should it be re-trained (c. 12.5% of the data). Please note this trading window is only 5 months as we do not have data as of the last trading day of July.

**3.2 Missing data & scaling features**

**Missing data:**

* Any character fields not required for modelling were dropped.
* A cross validation test on the training data proved that using KNNImputer (n\_neighbors=5) to impute missing numeric fields yielded a higher cross validation score over setting the remaining missing numeric fields to 0. Going forward different values for “n\_neighbors” should be tested.

**Scaling data:**

* Scale the values such that each are in the range [0,1]

Scaling was necessary for feature selection and modelling (completed using MinMaxScaler())

**Feature reduction:**

* The training dataset contained 1,012 features, if the model performs well on the training set there is a greater risk of overfitting)
* SelectKbest was considered but it is difficult to know at what chi squared level features should be dropped
* Recursive feature elimination performs Feature ranking with recursive feature elimination and cross-validated selection of the best number of features. The f1 score drops from 27.43% to 13.65% after removing 100 of the features. Given this decrease and the high degree of correlation between features reducing the feature dimensionality via PCA was used. The number of features was reduced to 100 which led to a minor c.10% decrease in variance. The cross validated training set precision increased as a result of applying PCA and there was reduced overfitting in the Artificial Neural Networks

Chart, line chart

Description automatically generated

***Figure 3: F1 score decrease generated by performing Recursive Feature Elimination***

1. **Modelling**

**4.1 Hyperparameter tuning:**

* The hyperparameters were tuned via GridSearchCV the scoring classifier used to evaluate the best hyperparameters was Precision. We use Precision as we want the model to be very selective in what stocks it predicts will have a stock growth of greater than 10% in the period essentially we want to try and eliminate Type I errors (cases which the model predicts will have growth greater than 10% but have a growth rate of less than 10%) while we are not as concerned with Type II errors (cases which have a stock price growth of more than 10% which the model has predicted will have a stock price growth of less than 10%).
  1. **Random Forest**
* A tuned Random Forest model with the following hyperparameters was used in the project: criterion= 'gini', max\_features='log2', min\_samples\_leaf=4, min\_samples\_split=10, n\_estimators=500.
* This model was evaluated on its own and as a base estimator of a “Stacked” model.
  1. **K Nearest Neighbours**
* The KNN model used in this project has n\_neighbours = 100
  1. **Support Vector Machines**
* A support vector machine with the following hyperparameters were used in the project C=0.25, gamma=0.3, kernel='rbf'
  1. **Naïve Bayes**
* A Gaussian Naïve Bayes model was used a Base estimator for the Stacked model
  1. **XGBoost**
* A tuned XGBoost model with the following hyperparameters was used as a base estimator in the Stacked model, gamma=5, learning\_rate=0.05, max\_depth=3, min\_child\_weight=1. XGBoost was also used as the meta model.

Diagram

Description automatically generated

***Figure 4: Stacked Model architecture***

* 1. **Genetic Algorithm:**
* The TPOTClassifier algorithm with a population size of 20 an offspring size of 3 and 5 generations of the algorithm was used in the project. The best pipeline was MLPClassifier(input\_matrix, alpha=0.1, learning\_rate\_init=0.001)
  1. **Neural Networks**

Graphical user interface, text, application

Description automatically generated

***Table 4: ANN table***

* Three diverse neural networks were run on the data to reduce the risk of overfitting the data (evaluated using a validation dataset which makes up 20% of the training dataset):
  + The multiple Neural Network used in the project is the average probability from each of these three diverse Neural Network models

Chart, scatter chart

Description automatically generated Chart, line chart

Description automatically generated

Chart, line chart, scatter chart

Description automatically generated

***Figure 5: Precision comparison of each of the Neural Networks***

1. **Results**

The S&P 500 return from Jul ’20 to Jan ’21 was 13.5% and the return from Jan ’21 to May ’21 was 13.2%.

Definitions in the below table:

* “Precision” refers to the test and deployment model precision.
* “Top 30 return” refers to the 6-month stock price return of the portfolio generated by taking the 30 stocks with the highest modelled probability of having a 6-month stock price growth of greater than 10%
* “Top 100 return,” is the same as “Top 30 return” only it includes 100 stocks
* The “Combined model” takes an average of the probabilities of the Stacked model and the Multiple Neural Network model.
* The “Min drawdown” portfolio takes the 200 stocks with the highest modelled probability of having a share price greater than 10% and takes only the top 30 stocks with the smallest drawdown over the last 4 years.
* The “Diversified” portfolio simply takes the stock with the highest probability from each industry and combines each of these into an overall well diversified portfolio (this was completed using the Combined model probabilities).

A picture containing text

Description automatically generated

***Table 4: Results split by Model.***

* 1. **Findings:**
* The “Combined” “Top 30” return model which is an average of the Stacked Model, and the Multiple Neural Networks generates 37.2% return in the Test data and 6.3% return on the Deployment data (assuming equal weights in the portfolio). An investment of €10k at Jul ’20 would have returned €14.58k at the end of May ’21 (46% return) versus 28% return from the S&P 500 for the same period.
* The Stacked model and the Random Forest model perform very well on the Test dataset (July ’20 – Jan ’21) with returns of 66.3% and 63.5% respectively.
* The “min drawdown” and “diversified” portfolio also outperform the S&P 500 in the period.
* Overall, the models look like they are already outdated by Jan ’21, I have retested the models including the test dataset in the training data and the result for the deployment period is much improved.
  1. **Seaborn Charts:**

**Test results:**

Below are two charts; the first is a comparison of the stock price in July ’20 and the future price at Jan ’21. This graph is overlaid with the “Top 30” stock portfolio for each model (the graph is at a log scale). Blue dots refer to stocks outside of the model portfolios.

The second graph contains the boxplots of the stock price return in the period overlaid by the swarmplots of the “Top 30” stock portfolio for each model. The boxplots represent overall sector performance for the 6 months.

Chart, scatter chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

***Figure 6/7: Test Results***

**Insights:**

* The Neural Network model, the Stacked model and the Random Forest model each selected Jumia Technologies as a good investment at July ’20 (6 month return on the stock was 270%)
* The Random Forest model also selected Neuronetics Inc. (560% return) as well as EHang holdings (698% return)
* The Neural Network model is the only model to select any stocks from the “Basic Materials” sector and most of the stocks it has selected seems to have underperformed. The Neural Network model selected only 4 stocks from the “Consumer Cyclical” sector of which two have outperformed Daqo New Energy Corp (296% return) and Revolve Group Inc. (127% return)
* There are four Sectors in which none of the models had any stocks “Consumer Defensive,” “Utilities,” “None” and “Other.” The lack of selection will need to be reviewed but it is most likely a result of poor performance of these industries over the last number of years.
* As the models are mainly biased toward technology stocks a better idea is to select the stock with the highest probability (from the Combined model) for each industry to create a well-diversified portfolio, this will allow the model to show what stocks in each industry are undervalued. This portfolio generates a 78.2% return in the period with investment in stocks such as Sunworks Inc. (1,243% return) and UP Fintech holding (200% return)

**Deploy results:**

Below are the same two charts; except the window is from Jan ’21 to May ’21

.

Chart, scatter chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

***Figure 8/: Deploy Results***

**References**

[1] Malkiel B. G. (1989) Efficient Market hypothesis, Finance pp127-134

[2]Fu, Xing Yu et al (2018) “A machine Learning Framework for stock selection”