NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

BC3409 AI In Accounting & Finance

Group Report

Seminar 2 Group 7

Instructor: Professor Teoh Teik Toe

Name	**Matriculation Number**
Marcus Foo Jun Rong	U1922071K
Richard Yang Chen Xiao	U1921319D
Ian Pang Yi En	U1921704F
Zoey Ong	U1921748F
Shao Yakun	U1920578C

Table of Contents

Executive summary

The team aims to construct a portfolio using AI hat outperforms the S&P 500 benchmark by trying to improve the Sharpe ratio while minimising the tracking error, using the S&P universe.

We made use of financial data from Sharadar and Yahoo Finance to train and test our Light GBM model, using data from earlier years to train and validate the model before testing the finalised model on data from more recent years. The main objective of the Light GBM model was to help us rank the relative performance of stocks, and group the top 25% of stocks into one category. We were able to achieve a final accuracy of 54.47% with the model.

Next, we optimised the portfolio using the Pyomo IPOPT optimiser, and introduced constraints to make sure that the optimised results lie within a realistic and acceptable range of values. We tightened our constraints during period of higher volatility and compared our results with the S&P 500 benchmark. After optimisation, our portfolio has a 5-year annualised Sharpe Ratio of 1.24, Tracking error of 21.89%, and positive Information Ratio of 0.16 which implies that our portfolio was superior to the S&P benchmark. As a form of stress-test for our portfolio's long-term performance, we also conducted a Monte Carlo Simulation of 1000 trials and found that only 17.3% ended the 10-year simulation with negative returns, which means that our portfolio will be able to sustain its performance in the long-term as well.

We proceeded to perform a return attribution analysis which allows us to quantify the impact of specific investment decisions within the portfolio, and show how they add or remove value relative to the S&P 500 benchmark. From this analysis, we see that our model has a higher allocation of value stocks compared to the benchmark on an average basis, and is able to adjust the portfolio composition in a reasonable way to generate a higher return. We also find that our solution is able to structure the portfolio in an unconventional way and bring positive active return.

In order to make it more accessible for users, we create an Oracle dashboard that users will be able to interact with. The dashboard consists of four tabs, which are Portfolio Performance, Portfolio Attribution, Stress Test, and Macro Trends.

Zoom Link:
https://us02web.zoom.us/rec/share/zSzId917eip7Cdc2MWF-6B2l3q0Phf5WdJjinDlAmXfLyaQ9V1D7EZDubyRKqddg.lSLM1YbhNYL_g_8c?startTime=1636616905000
(Presentation starts at 30:50)

1. Introduction

Due to the systematic nature of trading as well as the large amounts of financial data that goes into making every decision, AI has several use cases in Quantitative Trading. For the most part, AI is a powerful and fast-paced solution which helps with the processing of data and identifying patterns and trends in historical data which can be used to improve current investment strategies.

As of recent years, several applications of AI involve the use of making ‘Black Box’ models, which are machine learning models that provide little to no access to the internal workings and logic of the functions used in the model. However, in the financial world, without having fundamental understanding of how the AI model makes internal decisions, it is difficult to convince investors to place their trust in the product or prove that the model is making sensible and relevant decisions.

Hence, we believe that it is important to not only implement an AI solution that not only produces results, but one that is able to provide interpretable insights for the consumers of our product.

2. Business Problem Definition

Using the S&P universe, our team aims to construct a portfolio that outperforms the S&P 500 benchmark by trying to improve the Sharpe ratio while minimising the tracking error.

The Sharpe ratio is one of the most widely used methods for calculating risk-adjusted returns[1] . A portfolio with a higher Sharpe ratio is considered to have achieved better returns relative to the amount of risk it took on[2]. When considering investment choices and the performance of portfolios, the risk and returns should be evaluated together.

Tracking error measures how consistently our portfolio diverges from the benchmark. Since we will be comparing our portfolio to the S&P 500 as the benchmark, tracking error shows our portfolio’s consistency compared to the benchmark over a given period of time[3]. We believe that investors are looking for a reliable investment that provides consistent returns, hence we should aim to keep the tracking error as low as possible.

In the following report, we will discuss the steps taken by the team to produce a solution that not only outperforms the benchmark, but also allow users to gain access to some of the internal workings of our model.

[1] Fernando, J., 2021. Sharpe Ratio Definition. [online] Investopedia. Available at: <https://www.investopedia.com/terms/s/sharperatio.asp> [Accessed 7 November 2021].

[2] News.morningstar.com. 2021. The Sharpe Ratio Defined. [online] Available at: <http://news.morningstar.com/classroom2/course.asp?docId=2932&page=4> [Accessed 9 November 2021].

[3] Investopedia. 2021. Tracking Error Definition. [online] Available at: <https://www.investopedia.com/terms/t/trackingerror.asp> [Accessed 9 November 2021].

3. Literature Review

There have been multiple prior published research on using different types of AI forecasting models to predict stock prices in the future. Accurate stock price forecast would help investors pinpoint specific stocks that are outperforming benchmarks or other stocks, which is why there has been countless research over the past few years in an attempt to develop better models[4].

Despite the numerous research on predicting stock prices, there are still many out there who doubt that AI would be able to consistently predict stock prices due to existing theories[5]. Examples of such are the Random Walk Theory which and the Efficient Market Hypothesis, both of which agree that it is impossible to continuously outperform the market.

To explore other options, we look into existing literature on predicting the relative returns, and find that relative returns exhibit substantial predictability, more so than aggregate returns[6] . This means that we are likely to find more success in identifying better performing stocks if we measure their relative performance instead.

4. Data Preparation

4.1 Data Sources

A total of 9 separate CSV files were imported and used in the course of this project. The majority of the files were taken from the Sharadar, which is an independent research and analytics firm specializing in extraction, standardization and organization of financial data[7]. One notable data set that we took from the Sharadar data is the *financials.csv* file, which contains the daily key financial statement metrics such as Market Capitalization, Price to Earnings (PE) ratio, and Price to Books (PB) ratio.

Another source of data we referred to was Yahoo Finance, which was used to attain the macro data needed, including the *vix.csv* file containing the Cboe Volatility Index data from 23 August 1990 up to 21 September 2021. The Cboe VIX index is designed to be a proxy of the expected volatility of the S&P 500 index, and measures how much the market thinks the S&P 500 Index will fluctuate in the 30 days from the time of each tick of the VIX Index[8].

[4] Mondal P, Shit L, Goswami S. Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices[J]. International Journal of Computer Science, Engineering and Applications, 2014, 4(2): 13.

[5] Beck, C., 2021. Predicting the Stock Market is Hard: Creating a Machine-Learning Model (Probably) Won't Help. [online] Medium. Available at: <https://towardsdatascience.com/predicting-the-stock-market-is-hard-creating-a-machine-learning-model-probably-wont-help-e449039c9fe3> [Accessed 10 November 2021].

[6] Haddad, V., Kozak, S. and Santosh, S., 2017. Predicting Relative Returns.

[7] Data.nasdaq.com. 2021. Nasdaq Data Link. [online] Available at: <https://data.nasdaq.com/publishers/sharadar> [Accessed 10 November 2021].

[8] Cboe.com. 2021. Cboe VIX FAQ. [online] Available at: <https://www.cboe.com/tradable_products/vix/faqs/> [Accessed 10 November 2021].

4.2 Data Manipulation

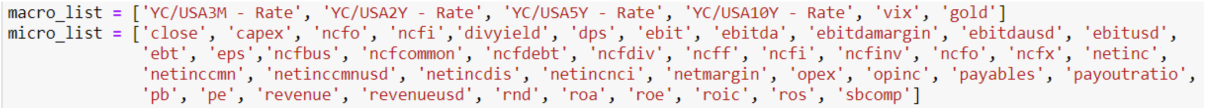
Since we sourced the data from reliable organisations and websites, we found that there was not a lot of data cleaning that had to be completed. Hence, we were able to quickly move into manipulating the data tables in preparation for the modelling.

4.2.1 Converting to a Multiindex DataFrame

We also converted the relevant tables into multiindex DataFrames, whereby we set the index to be the following tuple: (Date, Ticker), instead of the default index. This way, we would be able to view a list of all the stocks, and their individual variable values at every date stamp.

4.2.2 Backwards Rolling Features

We also rolled a list of variables backwards for a period of 1, 2, 3, 6, 9, and 12 months. The new variables will give us the percentage change in the value of the original variable from x number of months before and the current date for each time stamp.
Using backwards rolling provides us with more information about the behaviour of each of the variables compared to a single snapshot of one period.[9]



```
macro_list = ['YC/USA3M - Rate', 'YC/USA2Y - Rate', 'YC/USA5Y - Rate', 'YC/USA10Y - Rate', 'vix', 'gold']
micro_list = ['close', 'capex', 'ncfo', 'ncfi','divyield', 'dps', 'ebit', 'ebitda', 'ebitdamargin', 'ebitdausd', 'ebitusd',
              'ebt', 'eps','ncfbus', 'ncfcommon', 'ncfdebt', 'ncfdiv', 'ncff', 'ncfi', 'ncfinv', 'ncfo', 'ncfx', 'netinc',
              'netinccmn', 'netinccmnusd', 'netincdis', 'netincnci', 'netmargin', 'opex', 'opinc', 'payables', 'payoutratio',
              'pb', 'pe', 'revenue', 'revenueusd', 'rnd', 'roa', 'roe', 'roic', 'ros', 'sbcomp']
```

Figure 1: List of variables to be included in backwards rolling

4.2.3 Merging Columns

With the new multi-index tables, we proceed to merge all of the different tables into one, such that we would be able to make use of all the variables at once when training the model. This was done using the join() function, joining the tables based on the multiindex mentioned before to create the combined DataFrame *df_combined.*

4.3 Data Exploration

Before we started on the modelling process, we first conducted some simple exploratory data analysis, such that we have a clearer idea of the characteristics of the data that we would be working with. The following are a few of the graphs that we visualised from the data, other visualisations done can be found in Appendix A.

[9] Chen, J., 2021. Rolling Returns. [online] Investopedia. Available at: <https://www.investopedia.com/terms/r/rollingreturns.asp> [Accessed 10 November 2021].

4.3.1 Time Series plot of Closing Price of S&P 500 Index and Gold

We plotted the time series of both the VIX Index as well as the price of gold from the first recorded date given in the data set. From the line plot below, we can see that the price of gold increased during the beginning of 2020 at the same time there was a spike in VIX Index. This is likely due to investors opting to hold gold over other assets when they feel that the S&P 500 Index is too volatile.

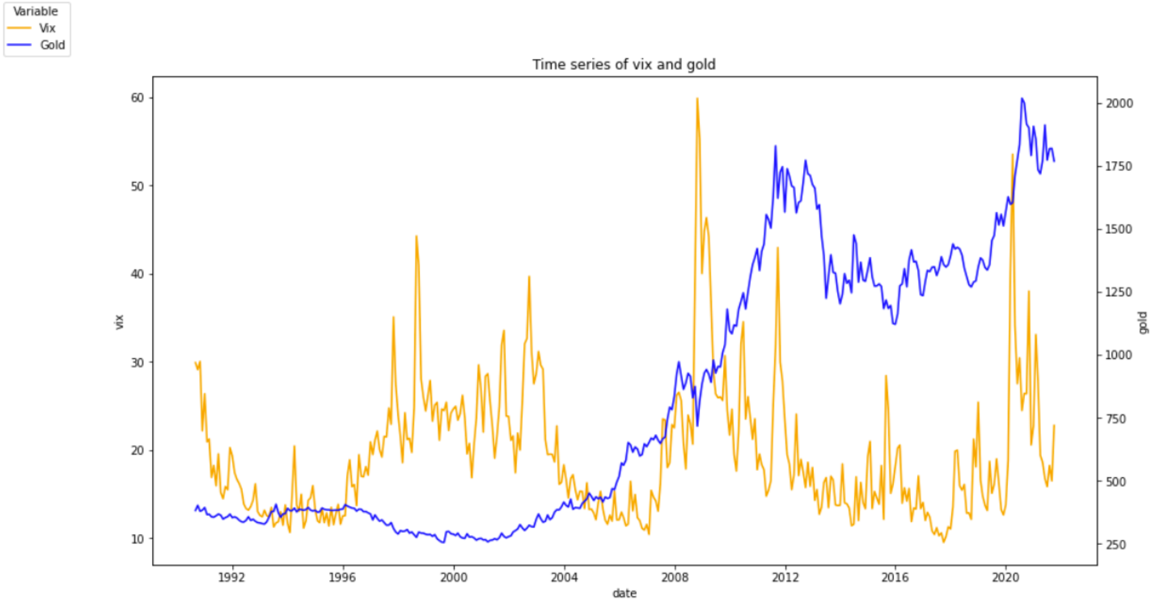
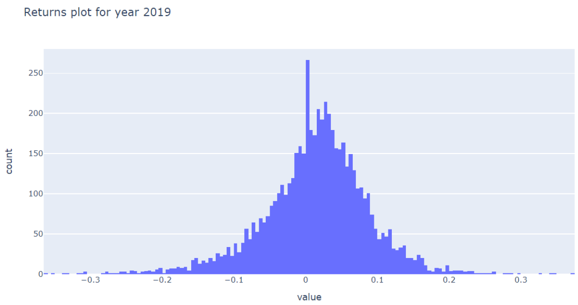


Figure 2: Time Series of VIX Index and Price of Gold

4.3.2 Distribution plot of Returns per Year

The team also made use of distribution plots to display the spread of returns for each year, from 2016 to 2021. From these plots, we are able to see how widely distributed the returns of each year are. The two plots attached below of Years 2019 and 2020 show us how the pandemic affected the distribution of the returns in 2020. This can be seen from the wider distribution in Figure 3, which signifies that the variance of the returns are much higher compared to that of 2019 (Figure 3).



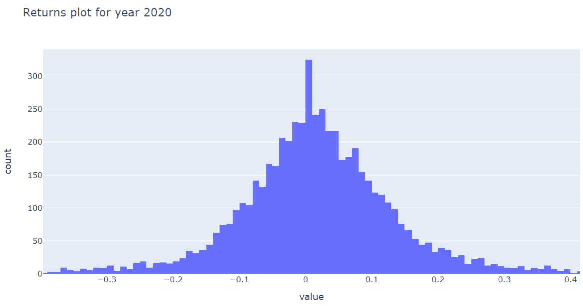


Figure 3: Distribution of Returns in 2019

Figure 4: Distribution of Returns in 2020

5. Modelling

“Black Box” models used regularly for AI do not provide much insight into how the models attain their final output. Without these insights, it will be difficult to provide a clearer understanding to investors as to how we obtain our results and prove why our model works. After experimenting with various models, we decided to use the Light Gradient Boosting Machine (Light GBM), a tree-based model where we are able to extract the individual trees to get an understanding of how the model makes its decision to determine the classes. This will allow us the opportunity to prove that the model is making sensible and relevant decisions.

5.1 Description of Model

Light GBM is a fast, distributed, high-performance gradient boosting framework based on the decision tree algorithm. Weak learners in the previous trees are modified to handle more difficult predictions. In contrast to XGboost, which uses a pre-sorted algorithm and Histogram-based algorithm for computing the best split, Light GBM uses a new technique called the Gradient-based One-Side Sampling instead.In summary, this provides a large number of benefits such as faster training speeds, higher efficiency and lower memory usage [10].

5.2 Dependent variable

Our dependent variable will be the percentage change for the variable ‘close’ 1 year in advance, encoded into a binary category. This is calculated from forward rolling ‘close’ by 1 year which gives us the stock’s return if we were to purchase the stock at that point in time. Thereafter, we encoded the top 25% of the values into category ‘1’ and the remaining of the values to ‘0’. This is done for all the stocks each time period.

5.3 Objective

The objective of the model is to classify the dependent variable into category ‘1’ or category ‘0’. This was done as we would like to rank the relative performance of the stocks rather than their absolute performance . It is very difficult to predict the absolute returns of stocks and investors would be very skeptical if we are able to do so.

Hence, for this model, we are more interested in looking into the assigned probability by the model that the asset belongs to class ‘1’ to be used in the model, rather than the actual assigned classes.

[10] Exsilio Blog. 2021. Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog. [online] Available at: <https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/> [Accessed 12 November 2021].

5.4 Splitting Data into Train, Test, and Validation Sets

Before building the Light GBM model, the data was first split into a train, validation and test set split by years in order to perform hyperparameter tuning in the next step. The ratio of train:validation:test sets are set to be 0.6 : 0.2 : 0.2. Having a validation set ensures that we are able to tune the hyperparameters to the validation set.

Without a validation set, we would have to do the hyperparameter optimization on the train set, and this would lead to overfitting of the model and hence result in less generalization. Finally, we want to test if the model with the optimized hyperparameters would predict well on the out of sample test set as well. If the accuracy remains similar, it would mean that the model is more robust when predicting on cases we have not seen before.

5.5 Hyperparameter tuning

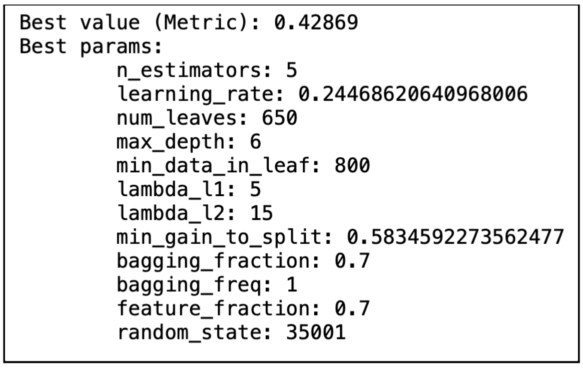
To tune the Light GBM hyperparameters, we made use of the package ‘optuna’ to tune the following 12 hyperparameters: ‘n_estimators’, ‘learning_rate’, ‘num_leaves’, ‘max_depth’, ‘min_data_in_leaf’, ‘lambda_l1’, ‘lambda_l2’, ‘min_gain_to_split’, ‘bagging_fraction’, ‘bagging_freq’, ‘feature_fraction’ and ‘random_state’.

We decided to use this instead of grid search as grid search is too computationally expensive for our large dataset. One point to note is that we set the search to range from only 1-5 for ‘n_estimators’ as we would like to see the trees being plotted to be able to identify and explain any interesting findings. ‘n_estimators’ refers to the number of trees to be inside the ensemble.

We chose to use macro-F1 score as the metric to determine the best hyperparameters for the model. The F1 score takes into account both false positives and false negatives, which makes it more insightful compared to accuracy, especially since we are dealing with an uneven class distribution[11]. The macro-F1 score is then computed by taking the average of the F1 scores of the clases. The macro-F1 score is penalised more when the model does not perform well with the minority classes, and is useful in our case as our main goal is to predict the minority category ‘1’s . The model parameters with the best macro-F1 score would be chosen as the best parameters after 100 trials.

For the optimization, the model was trained on the train set and we used the validation set to choose the best parameters.
These are the results of the optimization.

[11] Medium. 2021. LightGBM vs XGBOOST: Which algorithm win the race !!!. [online] Available at: <https://towardsdatascience.com/lightgbm-vs-xgboost-which-algorithm-win-the-race-1ff7dd4917d> [Accessed 12 November 2021].



```
Best value (Metric): 0.42869
Best params:
        n_estimators: 5
        learning_rate: 0.24468620640968006
        num_leaves: 650
        max_depth: 6
        min_data_in_leaf: 800
        lambda_l1: 5
        lambda_l2: 15
        min_gain_to_split: 0.5834592273562477
        bagging_fraction: 0.7
        bagging_freq: 1
        feature_fraction: 0.7
        random_state: 35001
```

Figure 5: Best Parameters from Optimization

5.6 Model findings

Finally, we used the best parameters determined from the optimization process and tested our model on the test set. Our final accuracy was 54.47%. It is important to note that although the accuracy is not very high, we are more interested in the probability assigned to the tickers by the model as we will be using this value to determine the weights of the stocks in our portfolio.

5.7 Interesting findings from LGBM

As the number of estimators was set to 5, we could print out the individual trees to see how they are split by and examine them. This would be useful in explaining to investors an idea of how our model makes predictions instead of just showing them the results.

One sample we can look at is from the third tree whereby we can conclude that it is a tree that focuses on the macro environment. As we can see from the small snippet below, it first looks at the US 3 Month Yield Curve, then looks at the rolling close of the stock price over the past 9 months. Therefore, we can conclude that this tree is trying to make use of macro features to improve on the previous models. If we zoom in further, we can see the individual variable conditions that lead to the model giving the asset a certain score.

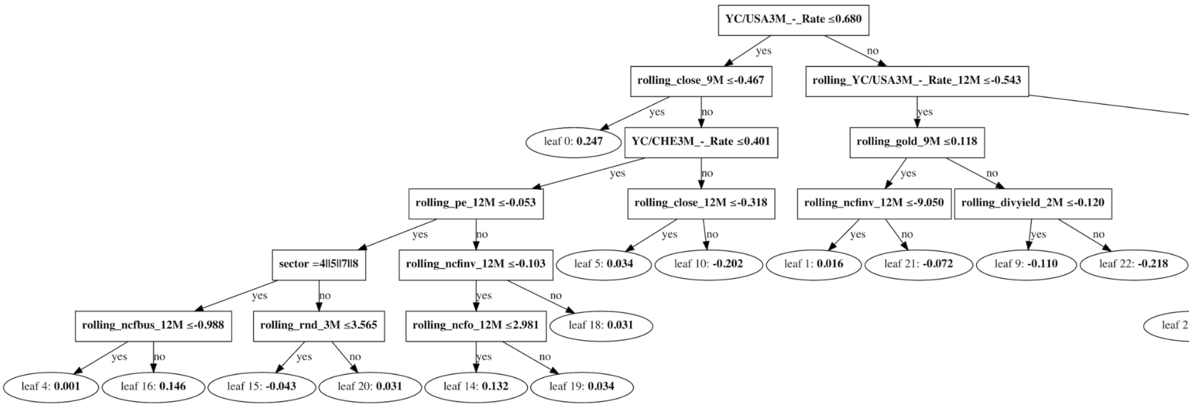


Figure 6: Extracted Tree 3 from Light GBM model

In contrast, when we look at tree 4, we can see that the tree is looking more at micro features, where the tickers themselves play a large role in determining if a stock is placed in category '1' or category '0'. In this tree, we can see from the left branch where as long as a ticker belongs to one of the tickers listed, the model will already lean towards predicting the stock to be in category 1. As an example, the ticker labeled as 4 is 'AAPL' and the model leans towards positively categorising it and it could be because Apple has historically achieved high growth in the past.

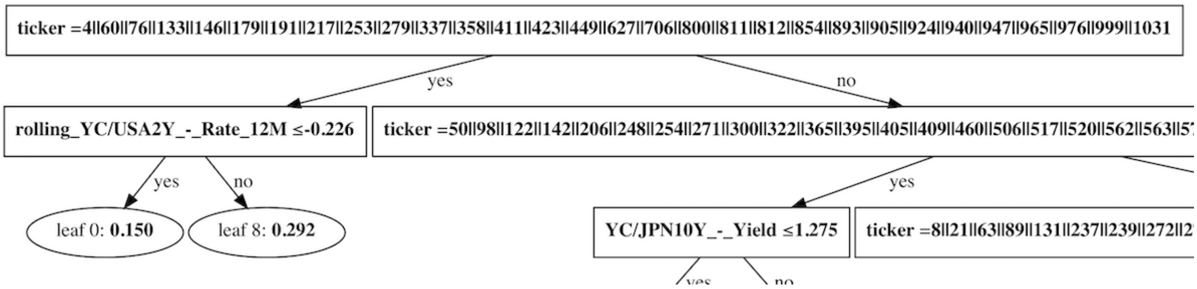

ticker =4||60||76||133||146||179||191||217||253||279||337||358||411||423||449||627||706||800||811||812||854||893||905||924||940||947||965||976||999||1031
yes
no
rolling_YC/USA2Y_-_Rate_12M ≤-0.226
ticker =50||98||122||142||206||248||254||271||300||322||365||395||405||409||460||506||517||520||562||563||5
yes
no
yes
leaf 0: 0.150
leaf 8: 0.292
YC/JPN10Y_-_Yield ≤1.275
ticker =8||21||63||89||131||237||239||272||2
yes
no

Figure 7: Node from Tree 4 of Light GBM model

5.8 SHAP

The Shapley value is a concept which implements a game theoretic approach to determine the expected marginal contribution of each component to an outcome. Feature Importance is most commonly done using Mean Decrease in Accuracy (MDA). SHAP values are similar to a variable importance plot, and enables us to identify some of the most impactful factors. Following (Chan and Man, 2020), we opted to study SHAP values to determine feature importance, which is reportedly more stable than conventional MDA[12].

[12] Man, X. and Chan, E., 2021. The best way to select features?. [online] arXiv.org. Available at: <https://arxiv.org/abs/2005.12483> [Accessed 12 November 2021].

We plotted the features with the highest SHAP values in the plot below, and found that ‘ticker’ has the highest SHAP values by a significant amount. This is followed by multiple macro oriented features such as the treasury yield rates and gold.

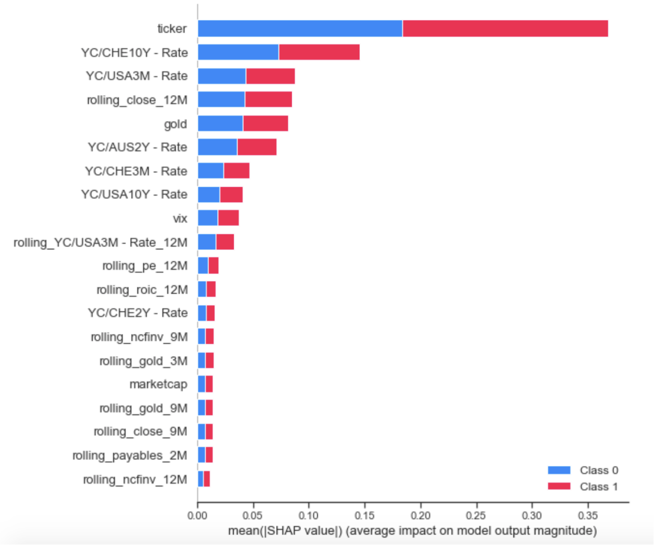


Figure 8: Plot of SHAP values against variables

6. Portfolio

6.1 Optimisation

Portfolio Benchmark

In order to ensure that our model is able to produce satisfactory results, we introduced a benchmark to compare our final results after portfolio optimisation.

The SP500 index is broadly considered the best measure of large US stocks. By recording the index includes about 80% of the 500 leading companies in the available market capitalization, the SP500 index is considered a measure of the US stock market's average record. This makes it a suitable benchmark for our experiment. We proceed to optimise a long-only portfolio.

6.1.1 Description of Pyomo Optimiser

Pyomo is a Python-based, open-source optimization modeling language with a diverse set of optimization capabilities. Given the non-linear nature of the objective function after imposing the active weight constraints, we have opted to use the Interior Point Optimizer (IPOPT) optimiser[13] (IPOPT, 2005).

[13] Coin-or.github.io. 2021. Ipopt: Documentation. [online] Available at: <https://coin-or.github.io/Ipopt/index.html> [Accessed 12 November 2021].

Decision Variables

$$S:\ List\ of\ S\&P\ Securities$$

$$K:\ List\ of\ Sectors$$

$$L_{security,\,t}:\ Active\ Security\ Constraint\ at\ time\ t$$

$$L_{sector}:\ Active\ Sector\ Constraint$$

$$w_{p,s,t}:\ Portfolio\ Weight\ for\ security\ s\ at\ time\ t$$

$$w_{b,s,t}:\ Benchmark\ Weight\ for\ security\ s\ at\ time\ t$$

$$p_{s,t}:\ Alpha\ Score\ for\ security\ s\ at\ time\ t$$

Objective Function

$$max\ (\sum_{i}^{S} p_{i,t} w_{p,i,t})$$

Constraints

$$(\sum_{i}^{S} w_{p,i,t})\ =\ 1$$

$$0\ <=\ w_{p,s,t}\ <=\ 1\ for\ s\ \epsilon\ S$$

$$-\ L_{security,\,t}\ <=\ w_{p,s,t}\ -\ w_{b,s,t}\ <=\ L_{security,t},\ for\ s\ \epsilon\ S$$

$$-\ L_{sector}\ <=\ \sum_{i}^{S}(w_{p,i,t}\ -\ w_{b,i,t})\ <=\ L_{sector}$$

6.1.2 Constraints Implementation

Here we explain the constraints that were implemented in our optimisation problem. The detailed codes used will be included in Appendix B.

proper_weights and ***portfolio_sum_to_one***
These two constraints ensure that the weights of each asset remain within the 0 to 1 range and that the sum of all weights is equal to 1 at all times respectively.

active_security_bet

This constraint ensures that the weight of each asset does not deviate too far from the weights assigned by the benchmark.

We have set the value of L_security to 0.02, which means that the weight of any asset stock in our portfolio must be within +- 2% relative to the S&P benchmark.

active_sector_constraint

Similar to active_security_bet, this constraint ensures that the combined weight of the assets in each sector does not deviate too far from the combined weights of the stocks assigned by the benchmark in the same sector.

For this project, L_sector is set to 0.2. This means that the combined weight of the stocks in any sector will be constrained to have an aggregate sector weight of +-20% relative to the benchmark.

6.2 VIX Volatility Regime

Gaussian Mixture Model (GMM) is a probabilistic model that assumes that the underlying data is derived from a mixture of k Gaussian Distributions with unknown parameters. The unknown distribution parameters are suggested by implementing the Expectation Maximization algorithm, and evaluated by minimising the Kullback-Leibler Divergence. For readers who wish to have a deeper understanding of the EM algorithm, we will refer them to (Mohamed and Saïdane, 2009)[14]. Following (Botte and Bao, 2020) at Two Sigma[15], we utilise GMM to identify 3 market regimes based on the movements of the VIX Index, which is commonly used as a proxy for forward-volatility.

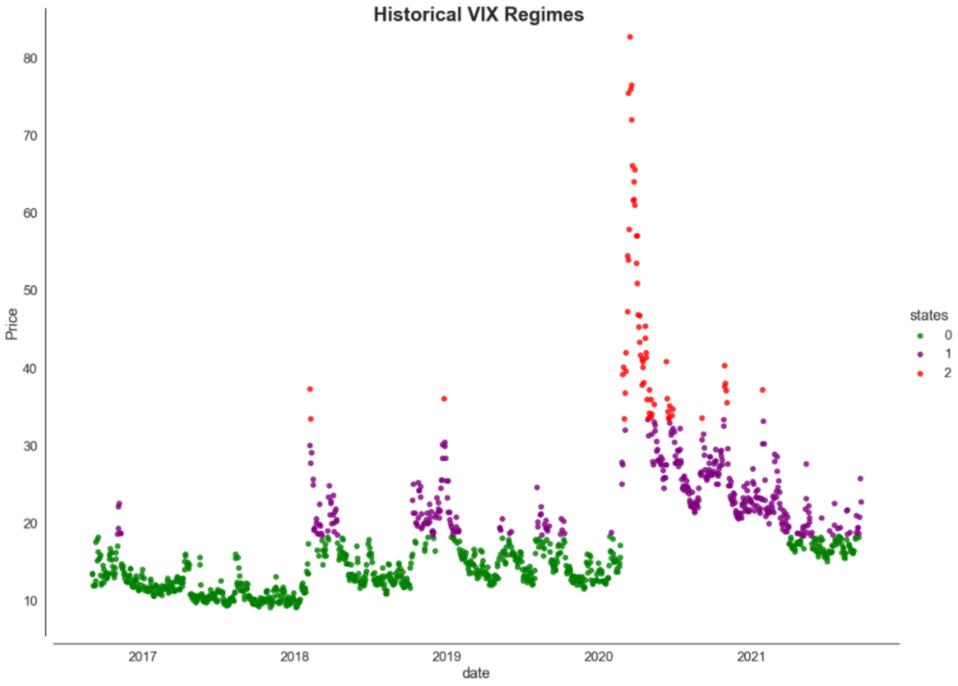


Figure 9: VIX Regimes

The model is trained similar to the before-mentioned train-test splits. We clearly see 3 distributions of unique VIX levels. Intuitively, we viewed the green, purple, and red color points as monthly periods of low, medium, and high volatility regimes. For periods of high

[14] Jaïdane-Saïdane, M. and Mohamed, O., 2021. Generalized Gaussian mixture model. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/abstract/document/7077399/authors#authors> [Accessed 12 November 2021].

[15] Botte, A. and Bao, D., 2021. A Machine Learning Approach to Regime Modeling - Two Sigma. [online] Two Sigma. Available at: <https://www.twosigma.com/articles/a-machine-learning-approach-to-regime-modeling/> [Accessed 12 November 2021].

volatility, we tighten the constraints for active security bets by a factor of 2. The rationale for this is force the optimiser to hold a larger basket of stocks for diversification benefits, helping to reduce volatility within the portfolio (Statman, 1987)[16]. In periods of higher economic uncertainty, we logically impose the condition that our portfolio must select more stocks and diversify holdings on securities with higher likelihood of outperforming the universe within the next year.

6.3 Backtest

The backtest is conducted purely out-of-sample after the validation set, and is a realistic representation of the portfolio's performance in the real world. Following the previous section, we also see that the portfolio increases the number of stocks in the portfolio during periods of higher volatility (refer to Appendix B, Figure 21).

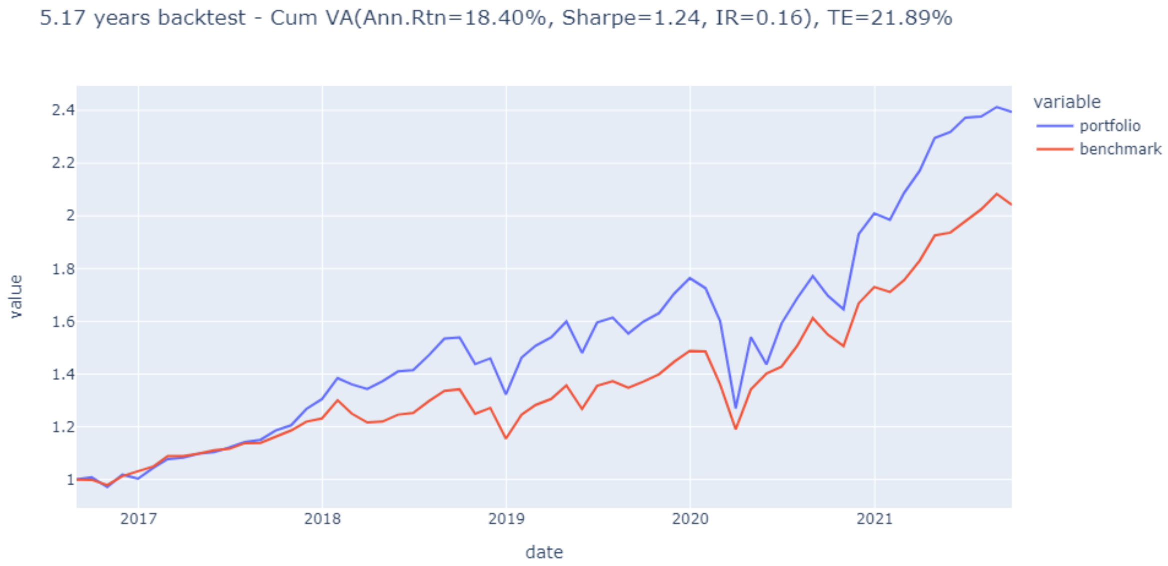


Figure 10: Backtest Performance

6.3.1 Performance Metrics

We measure the performance of our portfolios with three measures; Information Ratio, Sharpe Ratio, and Tracking Error.

6.3.2 Sharpe Ratio

$$SR_a = \frac{E[R_p - R_f]}{\sigma_p},$$

$$\text{where } R_p \text{ and } R_f \text{ represent the portfolio and risk free returns, and } \sigma_p \text{ represents the volatility of the portfolio}$$

[16] Statman, M., 2021. [online] JSTOR. Available at: <https://www.jstor.org/stable/2330969> [Accessed 12 November 2021].

The Sharpe Ratio (Sharpe, 1966)[17] is a common financial measure of risk-adjusted returns, where a value of more than one indicates good performance. Our portfolio has a 5-year annualised Sharpe Ratio of 1.24.

6.3.3 Information Ratio

$$TE = \sqrt{\frac{\sum_{i}^{N}(R_p - R_b)^2}{N-1}},$$

$$IR = \frac{R_p - R_b}{TE},$$

$where\ R_p\ and\ R_b\ represent\ the\ portfolio\ and\ benchmark\ returns\ for\ the\ universe\ of\ N\ stocks$

The Tracking Error is a measure of the risk in an investment portfolio that is due to active management decisions made by our stock-selection model. The Information Ratio answers two broad questions. Firstly, whether the portfolio was able to outperform its benchmark. Secondly, whether the portfolio was consistent in its outperformance. Our portfolio had a positive Information Ratio of 0.16 which implies that our portfolio was superior to the S&P benchmark.

6.4 Transfer Coefficient

The Transfer Coefficient (TC) is a measure that measures the degree of transfer from research insights into active weights. Statistically, we have proxied the Transfer Coefficients by looking at the cross-sectional correlations of active returns and active weights for every year. Given the unique nature of 2020's economic environment due to COVID-19, we have displayed the 2020 plot below. Our portfolio obtained a pearson correlation of 0.00, and rank correlation of 0.01, which is expected given that we have given our portfolio a much tighter constraint during this period.

This result can be explained by two points. First, we have not incorporated a turnover constraint which will help improve the stability of the portfolio holdings, and would lead to more points concentrating in the top right and bottom left quadrants. Second, the S&P Index is a cap-weighted index with significant holdings in the top 10 market-cap stocks. As such, there will be a larger concentration in these stocks due to the active security constraints.

This can also be seen in our backtest performance, where our portfolio was able to rebound much faster than the benchmark in March 2020. Visually, the ideal scenario is when the data points are scattered in the bottom left and top right quadrant, indicating an under-weighting (over-weighting) of stocks with lower (higher) forward returns.

[17] Sharpe, W., 2021. The Sharpe Ratio. [online] Web.stanford.edu. Available at: <https://web.stanford.edu/~wfsharpe/art/sr/SR.htm> [Accessed 12 November 2021].

As a sanity check for our optimiser, we also see from the TC plot (refer to Appendix B, Figure 22) that our portfolio correctly follows the tighter 1% active security constraint when in a high volatility regime.

6.5 Simulation

As a form of stress-test for our portfolio's long-term performance, we have conducted a Monte Carlo Simulation of 1000 trials with an initial capital of $1000 for 10 years. We modelled this stochastic process using the Geometric Brownian Motion Model, which is represented as:

$$dS_t = S_t \mu dt + S_t \sigma \epsilon \sqrt{\Delta t},$$

$\mathit{where}\ \mu\ \mathit{and}\ \sigma\ \mathit{are\ the\ expected\ annual\ returns\ and\ volatility\ derived\ from\ our}\ 5\ \mathit{year\ backtest}$

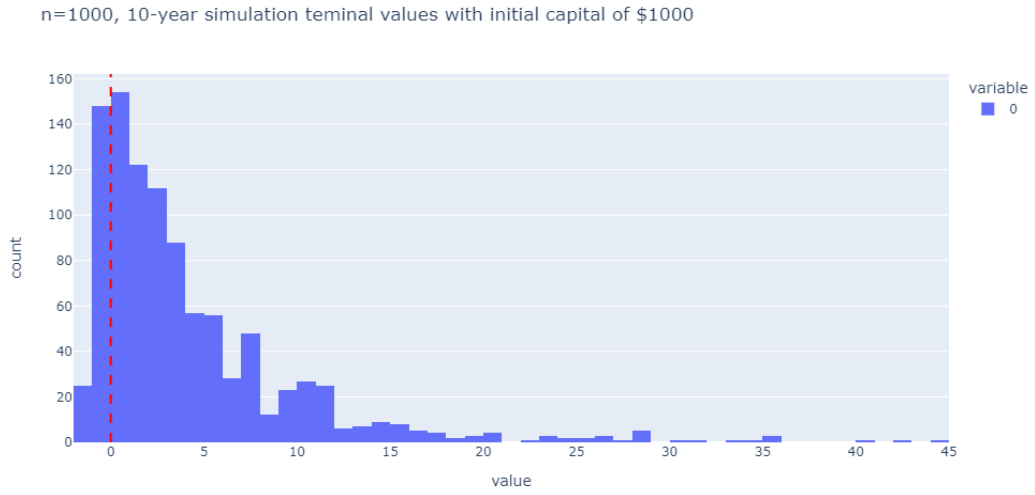


Figure 11: Distribution of Portfolio Return

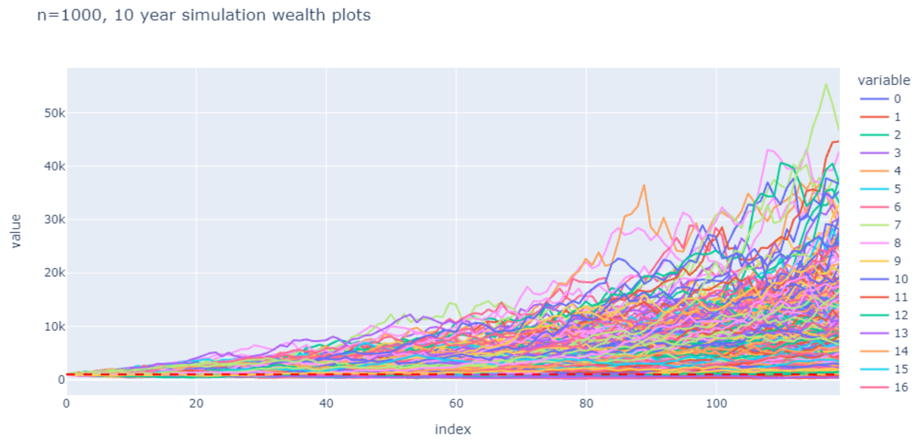


Figure 12: Monte Carlo Simulation

Out of the 1000 trials, only 17.3% ended the 10-year simulation with negative returns. We also observe a right-skewed distribution of returns, a highly desirable portfolio trait. We conclude that our portfolio can outperform the benchmark in the short-term, while sustaining its performance in the long-term as seen from the stress test.

7. Model Evaluation - Return Attribution

As seen from the above, the portfolio constructed by our model is able to make a considerably higher return compared to the benchmark. This is so because our model adjusts the portfolio composition periodically based on a set of algorithms developed through a pure statistical approach.

In this section, we hope to evaluate and understand our model's mechanism from a fundamental perspective and to explore what is contributing to the active returns over time, and to what extent the investment decisions made by our model could be justified using fundamental investment principles. This is done through a return attribution analysis:
Return attribution analysis uses fundamental factor models to decompose the relative contributors to the excess return (or active return) of a number of different factors. (Note that active return = portfolio return - benchmark return), this allows us to quantify the impact of specific investment decisions (based on specific fundamental factors) within the portfolio, and show how they add or remove value relative to the benchmark[18]. We will conduct our evaluation by examining how the impacts of each fundamental factor change overtime.

We will first explain briefly about our code implementation process and how the fundamental factors are selected, and then provide our interesting findings and conclusion.

7.1 Implementation

The implementation of return attribution consists of 3 steps: 1) Normalization of features, 2) factor-grouping, and 3) Regression analysis.

7.1.1 Normalization of features

As the features in the dataset have varying ranges, normalization is therefore needed for a fair analysis, where for each unique date, the value for each useful feature is normalized vertically across all stocks.

7.1.2 Factor-Grouping

As the number of features in our dataset is numerous, our analysis is therefore conducted based on "groups" features instead of individual features. We selected a set of generic fundamental factors that are closely related to stock price (more details see section 7.2 Factor-Grouping), and choose the relevant features from the dataset to be grouped within each generic factor to conduct the regression analysis.

[18] Carl R. Bacon, Marc A.Wright (2019). Return Attribution, CIPM References.

7.1.3 Regression analysis

To determine the contribution of each factor to the returns, a linear regression analysis is conducted on the factors against the returns. Below is the equation for our linear regression, where the feature importance score is indicated by the value of coefficients

$$Return\% = b_1{*}macro\_market\_risk + b_2{*}macro\_volatilty + b_3{*}micro\_financial\_risk + b_4{*}micro\_earnings + b_5{*}micro\_valuation + b_6{*}micro\_size + \varepsilon$$

, where b_n is the coefficient of each factor, and ε is a constant

7.2 Factor-Grouping

Overall, 6 generic factors are selected for our analysis. They can be classified in 2 categories: Micro (company-related) & Macro aspect. Below shows the list of our factors:

Micro factors	**Macro factors**
Company’s financial risk factors	Economic conditions (“macro_market_risk”)
Company’s earnings/operations factors	Market Volatility
Company’s size	-
Current valuation	-

Next, we will explain our rationale for selecting those factors:

7.2.1 Micro factors (company-specific)

One typical theory which could support our selection of the micro factors is the Fama-French model. The Fama-French model is one of the popularly used factor models that practitioners used to decompose excess returns, where it states that the size and value of the company are important factors that will affect the asset pricing[19]. In addition to that, other studies (Akbari et al., 2012) has also proven that there is significant relationship between the valuation multiples (e.g. P/E ratio) and the company’s systematic risk[20], as well as between the size metrics (e.g. Market value, book value) and the systematic risk[21]. Therefore, we choose **valuation** & **company size** as two groups of company-related factors in our return attribution

[19] Fama, E. F., & French, K. R. (1992). The cross-section of expected stock returns. Journal of Finance, 47(2), 427–465

Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. Journal of Financial Economics, 33(1), 3–56

[20] Jnani, M., & and Hadi Zadeh, H. (2001). The study of relation between prices to income over achieved return income. Economical Stock Journal, 50, 311-318

[21] Mosavi Kashi, M. (1999). The effect of company size on return of investment rate in companies accepted in Tehran Stock Market. Unpublished MA thesis. Shahid Beheshti University. Iran.

analysis, the two groups are named as “micro_valuation” and “micro_size” in our codes respectively.

In addition, a company’s idiosyncratic risk is another factor that will affect its pricing relative to the market. Idiosyncratic risk refers to the risk that is inherent to a company due to the way it operates its business. To accomodate this risk factor with the features available in our dataset, we breakdown a company’s idiosyncratic risk further into two aspects: operational aspect and financial aspect. For the **operational aspect**, we included features that could reflect the operating efficiency (e.g. return-on-asset ratio) as well as business profitability (e.g. profit margins). We named this factor group as “micro_earnings” in our code implementation. As for the **financial aspect**, we included factors that reflects the risks related to a company’s capital structure (e.g. debt-to-equity ratios) , and it is referred to as “micro_financial_risk” variable in our codes.

7.2.2 Macro factors

Macro factors are classified in two categories: **economic conditions** & **volatility**. Firstly, studies have shown that macroeconomic factors such as interest rates and inflation are significant factors that affect the change in stock prices (Rapach et al , 2005). For example, when interest rates move up, typically the stock price will be negatively affected as the company's earnings would decline due to the higher cost of funding to do business. Therefore, we group factors such as interest rates, FX, and commodity prices into one factor known as economic conditions (referred to as “macro_market_risk” in our codes).

In addition, we added volatility (which includes the Cboe Volatility Index (VIX)) as an additional factor in our return attribution model. VIX is a real-time market index representing the market’s expectations for volatility over the next 1 month. As it is frequently used by investors to measure the level of risk, fear, or stress in the market when making their investment decisions, it is therefore also an important factor that affects the market prices.

7.3 Interesting findings

7.3.1 Allocation bias compared to benchmark

Firstly, we found that the portfolio constructed by our model has a higher allocation of value stocks compared to the benchmark on an average basis, as shown below where a_5 (coefficient for micro_valuation) is above that of the benchmark index (S&P 500), as shown in the diagram in Appendix C. We view this as a reasonable adjustment, as it matches with the belief that value stocks tend to generate higher returns over the long term

7.3.2 “Smart” adjustments

Secondly, our model is able to adjust the portfolio composition in a reasonable way to generate a higher return. For example, in 2020 where most companies are hurt badly by the covid outbreak, our model adjusts the portfolio with a preference towards larger companies that generates a higher revenue (i.e. the micro_size factors, which consists of features such as market capitalization and absolute revenue), as seen from the below diagram (Jan-Jul 2020). Such adjustment is reasonable and could generate a higher return as larger companies tend to have more resources to support them through the difficult time and to revive faster compared to smaller companies.

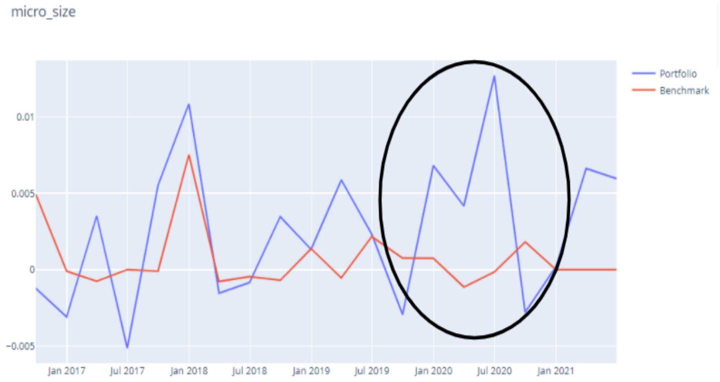


Figure 13: Feature importance for micro_size

7.3.3 Value-adds

While some of the investment decisions could be justified in a fundamental way, there are cases where the active return could not be attributed to any of the factors, one example is during the period from Sep-Dec 2020, as shown in the picture below.

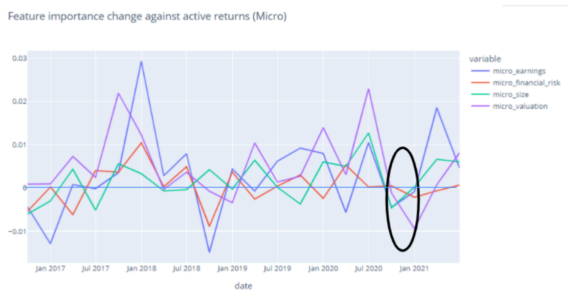
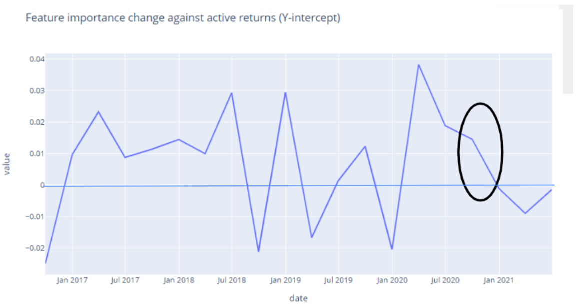
Feature Importance	Alpha-intercept
	

Figure 14. Feature importance & alpha-intercept

This part of active return contributed by the positive alpha-intercept is due to our model's special algorithm - as it first structures the portfolio with an algorithm developed from the LGBM learning process prior to optimization. This structuring process is unconventional to fundamental principles but can produce positive active returns, which is the **competitive edge** of our model.

7.4 Conclusion of Evaluation

To conclude, we could see that some investment decisions made by our models could be explained from a fundamental approach, but there are also others, which do produce positive active returns, but could not reasonably be explained from a fundamental aspect. Those mechanisms are the competitive edge of our product, as they are made from a statistical approach to structure the portfolio in an unconventional way and bring positive active return, and they could not be replicated easily by any of the traditional portfolio management methods or strategies used by portfolio managers.

8. Applications of Findings

8.1 Oracle Dashboard

In order to allow users or portfolio managers to fully utilize the portfolio, it is not enough to simply describe the results. An intuitive dashboard can go a long way in allowing users to utilise the model without learning the actual machine learning algorithm. We do so by giving users the ability to not just see, but also track and interact with the portfolio.

We recognise that users will have different levels of involvement with the investment process. Some users may want minimal interaction with the portfolio, e.g. they will only need basic information like the performance and tickers currently invested. Other users may want a more hands-on approach. E.g., on top of basic information of the portfolio, they'll also analyse macro trends to manually rebalance the portfolio.

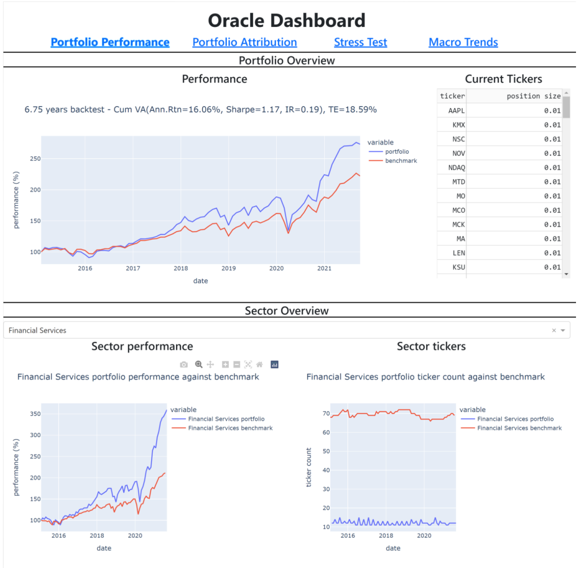
The solution we came up with is a comprehensive dashboard that breaks down the portfolio into relevant pieces of information. The most basic information will be on the front page. More technical features are also provided so that experienced users can make full use of the dashboard to dig into the inner workings of the portfolio algorithm. The dashboard was created with *dash* and is deployed on the internet at https://bc3409-portfolio-dashboard.herokuapp.com/apps/portfolio-performance.

The dashboard consists of four tabs: Portfolio Performance, Portfolio Attribution, Stress Test, Macro Trends. The tabs (from left to right) address the needs of users in increasing levels of technicality.

The Portfolio Performance tab tracks basic information e.g., portfolio's annual return, Sharpe ratio, Tracking Error, etc., portfolio sector performance. The Portfolio Attribution tab tracks how macro and micro factors affect the benchmark and portfolio. The Stress Test tab shows the Historical VIX Regimes and stress test results. The Macro Trends tab shows the interactions between different macro indicators.

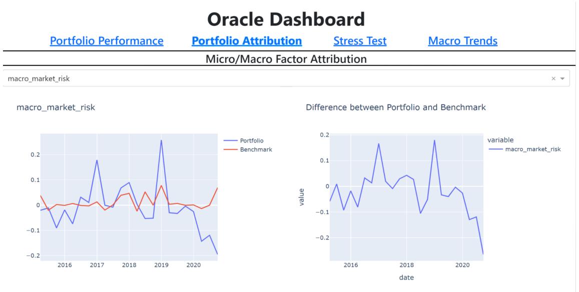
Below are screenshots of the dashboard.

Portfolio Performance

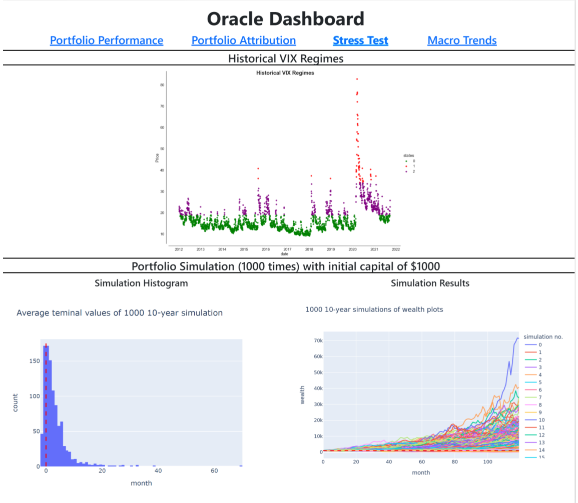

Oracle Dashboard
Portfolio Performance
Portfolio Attribution
Stress Test
Macro Trends
Portfolio Overview
Performance
Current Tickers
6.75 years backtest - Cum VA(Ann.Rtn=16.06%, Sharpe=1.17, IR=0.19), TE=18.59%
performance (%)
date
variable
portfolio
benchmark
Sector Overview
Financial Services
Sector performance
Sector tickers
Financial Services portfolio performance against benchmark
Financial Services portfolio ticker count against benchmark
Financial Services portfolio
Financial Services benchmark
ticker count

ticker	position size
AAPL	0.01
KMX	0.01
NSC	0.01
NOV	0.01
NDAQ	0.01
MTD	0.01
MO	0.01
MCO	0.01
MCK	0.01
MA	0.01
LEN	0.01
KSU	0.01

Portfolio Attribution



Stress Test



Macro Trends

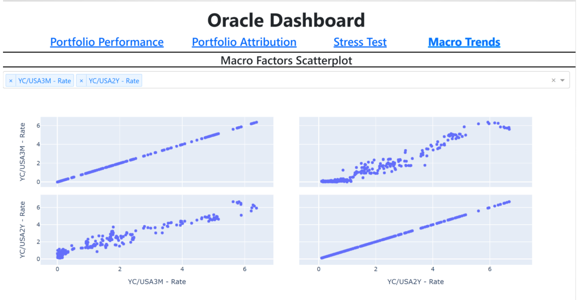


Figure 15: Screenshots from Dashboard

8.2 Limitations of the portfolio

One limitation of our portfolio is that it can only suggest the positions for the upcoming month. This is simply due to the nature of the market. Any further forecast will result in highly inaccurate results, which is not the intent. By limiting the forecast to a month, we are able to achieve decent performance, while still giving users adequate time to rebalance if they need to.

Another limitation of the portfolio is that the underlying machine learning model needs to be constantly updated. The accuracy of our model comes from the historical data it was trained with. Financial trends change very quickly. The model is only as good as the data it is fed. In order to keep the model accurate, new data needs to be fed to the model regularly. Luckily for us, this can be done by automating the deployment process and establishing a pipeline. With services like AWS, this process can be achieved easily.

8.3 Project Feasibility

Portfolio construction is a costly, time consuming and laborious process. But by leveraging machine learning, the process becomes a lot simpler. Computer algorithms are able to make decisions based on hundreds of variables in practically an instant, compared to human managers who will require time to analyze the variables individually.

The portfolio algorithm can be run on any computer on the network, and does not require the use of specialized hardware or equipment. After the construction of the portfolio, it can be easily tracked with the dashboard. The dashboard is comprehensive enough to be used by beginners to seasoned professionals alike.

8.4 Future Plans

We recommend users to test out the performance of the portfolio with live data. This will be the best benchmark of the portfolio's real performance.

Of course, this project is far from perfect. There are improvements that we hope to make in the future. We split them into four categories: data, model, portfolio construction, and general improvements.

The first and easiest improvement we can make is to use higher quality **data**. In order to improve the portfolio, we need to first improve the underlying model that predicts for the best performing stocks. The team has thought of two ways. 1. More feature engineering to form new streams of information and 2. creating another model/ purchasing data to predict stock sentiments on the internet using NLP. This will increase the amount of features for the model, which could lead to better predictive power.

With better quality data, we can then improve the **model**. Because we are limited by the amount of data we have, we need to make the most use of them. In our case, we performed a train-validate-test split. Upon further research, we found another method for training the model: Time Based Cross Validation. The general idea is to implement a "sliding window" approach to training and testing[22]. By fixing the size of the window and train-test proportion, we then slide the window to create multiple train-test splits using the same amount of data. This produces a more robust way of train-testing on time-series data.

The **portfolio construction** can also be improved. Right now, we are basing it off a naive assumption of the world. In the real world, we also have to take into account turnover rate

[22] Herman-Saffar, O., 2021. Time Based Cross Validation. [online] Medium. Available at: <https://towardsdatascience.com/time-based-cross-validation-d259b13d42b8> [Accessed 12 November 2021].

and transaction costs because greater amounts of these will reduce the performance of the portfolio. Hence, these additional constraints will be added in future iterations.

Other than making our portfolio more robust, we are also looking to improve our tool as well. For **general improvements**, we are thinking of creating a scheduling task to automatically update the portfolio periodically. This cut down the manual work needed to maintain the portfolio. It can be done by using services like AWS to automate the pipeline: from ingesting data, to transforming, to training the model, to deployment. Appendix E provides a sample architecture we can use.

8.5 Conclusion

In this report, we identified the business problem relating to portfolio construction and generated solutions to drive greater profitability and speed.

To solve this, we first used AI to predict for best performing stocks in the S&P 500 universe. Then, we synthesised a portfolio based on the predicted top performing stocks using several constraints such as limiting the number of tickers per industry. Based on our backtesting, our portfolio outperforms the S&P 500 benchmark over a 5-year horizon. To make the portfolio easy to analyse and monitor, we also created a dashboard that allows users to fully take advantage of the insights generated by the portfolio.

Given the ease of use and superior performance of the portfolio, it would be worthwhile for users to test our portfolio on real-time data.

References

1. Fernando, J., 2021. *Sharpe Ratio Definition*. [online] Investopedia. Available at: <https://www.investopedia.com/terms/s/sharperatio.asp> [Accessed 7 November 2021].

2. News.morningstar.com. 2021. *The Sharpe Ratio Defined*. [online] Available at: <http://news.morningstar.com/classroom2/course.asp?docId=2932&page=4> [Accessed 9 November 2021].

3. Investopedia. 2021. *Tracking Error Definition*. [online] Available at: <https://www.investopedia.com/terms/t/trackingerror.asp> [Accessed 9 November 2021].

4. Mondal P, Shit L, Goswami S. Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices[J]. International Journal of Computer Science, Engineering and Applications, 2014, 4(2): 13.

5. Beck, C., 2021. *Predicting the Stock Market is Hard: Creating a Machine-Learning Model (Probably) Won't Help*. [online] Medium. Available at: <https://towardsdatascience.com/predicting-the-stock-market-is-hard-creating-a-machine-learning-model-probably-wont-help-e449039c9fe3> [Accessed 10 November 2021].

6. Haddad, V., Kozak, S. and Santosh, S., 2017. Predicting Relative Returns.

7. Data.nasdaq.com. 2021. *Nasdaq Data Link*. [online] Available at: <https://data.nasdaq.com/publishers/sharadar> [Accessed 10 November 2021].

8. Cboe.com. 2021. *Cboe VIX FAQ*. [online] Available at: <https://www.cboe.com/tradable_products/vix/faqs/> [Accessed 10 November 2021].

9. Chen, J., 2021. *Rolling Returns*. [online] Investopedia. Available at: <https://www.investopedia.com/terms/r/rollingreturns.asp> [Accessed 10 November 2021].

10. Exsilio Blog. 2021. *Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog*. [online] Available at: <https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/> [Accessed 12 November 2021].

11. Medium. 2021. *LightGBM vs XGBOOST: Which algorithm win the race !!!*. [online] Available at: <https://towardsdatascience.com/lightgbm-vs-xgboost-which-algorithm-win-the-race-1ff7dd4917d> [Accessed 12 November 2021].

12. Man, X. and Chan, E., 2021. *The best way to select features?*. [online] arXiv.org. Available at: <https://arxiv.org/abs/2005.12483> [Accessed 12 November 2021].

13. Coin-or.github.io. 2021. *Ipopt: Documentation*. [online] Available at: <https://coin-or.github.io/Ipopt/index.html> [Accessed 12 November 2021].

14. Jaïdane-Saïdane, M. and Mohamed, O., 2021. *Generalized Gaussian mixture model*. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/abstract/document/7077399/authors#authors> [Accessed 12 November 2021].

15. Botte, A. and Bao, D., 2021. *A Machine Learning Approach to Regime Modeling - Two Sigma*. [online] Two Sigma. Available at: <https://www.twosigma.com/articles/a-machine-learning-approach-to-regime-modeling/> [Accessed 12 November 2021].

16. Statman, M., 2021. [online] JSTOR. Available at: <https://www.jstor.org/stable/2330969> [Accessed 12 November 2021].

17. Sharpe, W., 2021. *The Sharpe Ratio*. [online] Web.stanford.edu. Available at: <https://web.stanford.edu/~wfsharpe/art/sr/SR.htm> [Accessed 12 November 2021].

18. Carl R. Bacon, Marc A.Wright (2019). Return Attribution, CIPM References.

19. Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. Journal of Financial Economics, 33(1), 3–56

20. Jnani, M., & and Hadi Zadeh, H. (2001). The study of relation between prices to income over achieved return income. Economical Stock Journal, 50, 311-318

21. Mosavi Kashi, M. (1999). The effect of company size on return of investment rate in companies accepted in Tehran Stock Market. Unpublished MA thesis. Shahid Beheshti University. Iran.

22. Herman-Saffar, O., 2021. *Time Based Cross Validation*. [online] Medium. Available at: <https://towardsdatascience.com/time-based-cross-validation-d259b13d42b8> [Accessed 12 November 2021].

Appendix

Appendix A - Other Visualisations

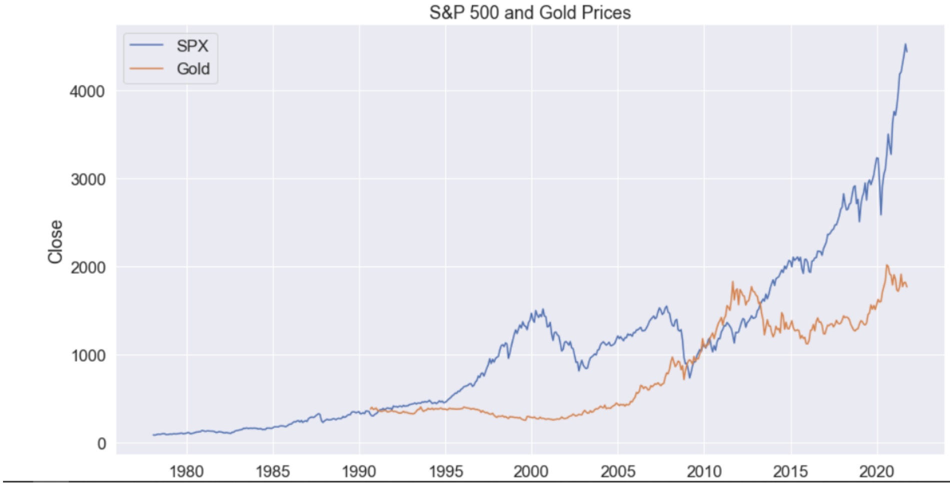


Figure 16: S&P 500 Index and Gold Prices

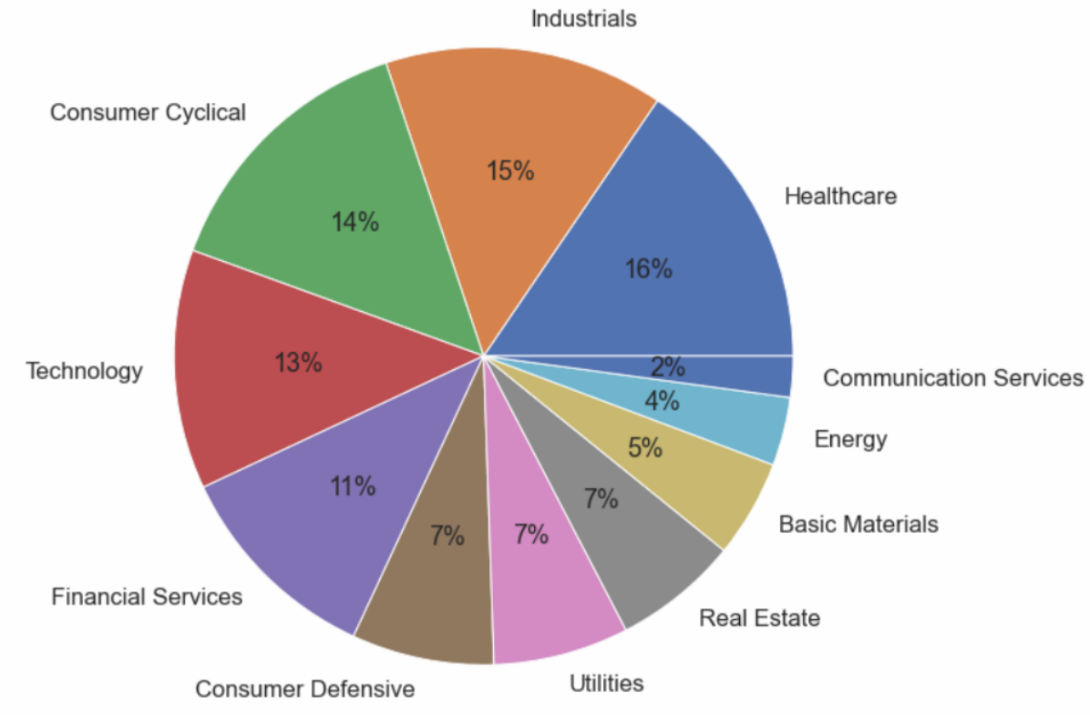
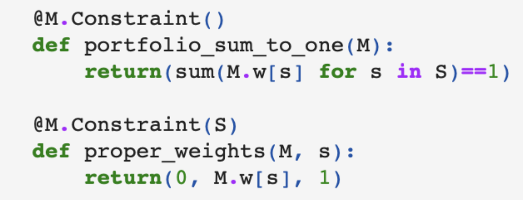


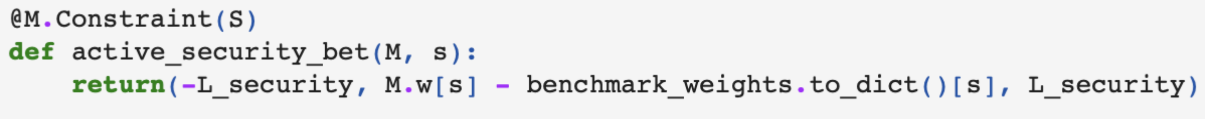
Figure 17: Pie Chart of the distribution of Sectors

Appendix B



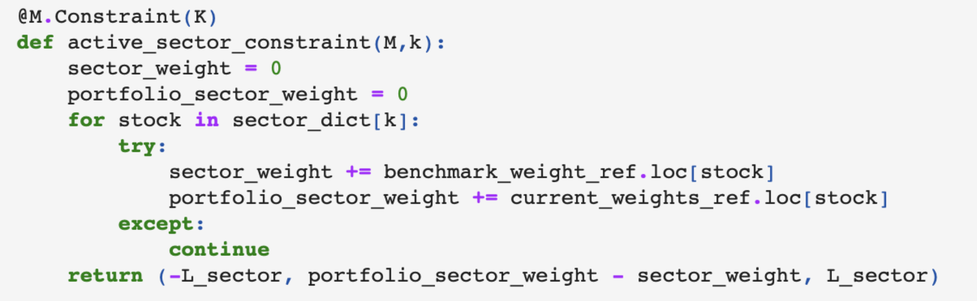
```
@M.Constraint()
def portfolio_sum_to_one(M):
    return(sum(M.w[s] for s in S)==1)

@M.Constraint(S)
def proper_weights(M, s):
    return(0, M.w[s], 1)
```

Figure 18: proper_weights and portfolio_sum_to_one



```
@M.Constraint(S)
def active_security_bet(M, s):
    return(-L_security, M.w[s] - benchmark_weights.to_dict()[s], L_security)
```

Figure 19: active_security_bet



```
@M.Constraint(K)
def active_sector_constraint(M,k):
    sector_weight = 0
    portfolio_sector_weight = 0
    for stock in sector_dict[k]:
        try:
            sector_weight += benchmark_weight_ref.loc[stock]
            portfolio_sector_weight += current_weights_ref.loc[stock]
        except:
            continue
    return (-L_sector, portfolio_sector_weight - sector_weight, L_sector)
```

Figure 20: active_sector_constraint

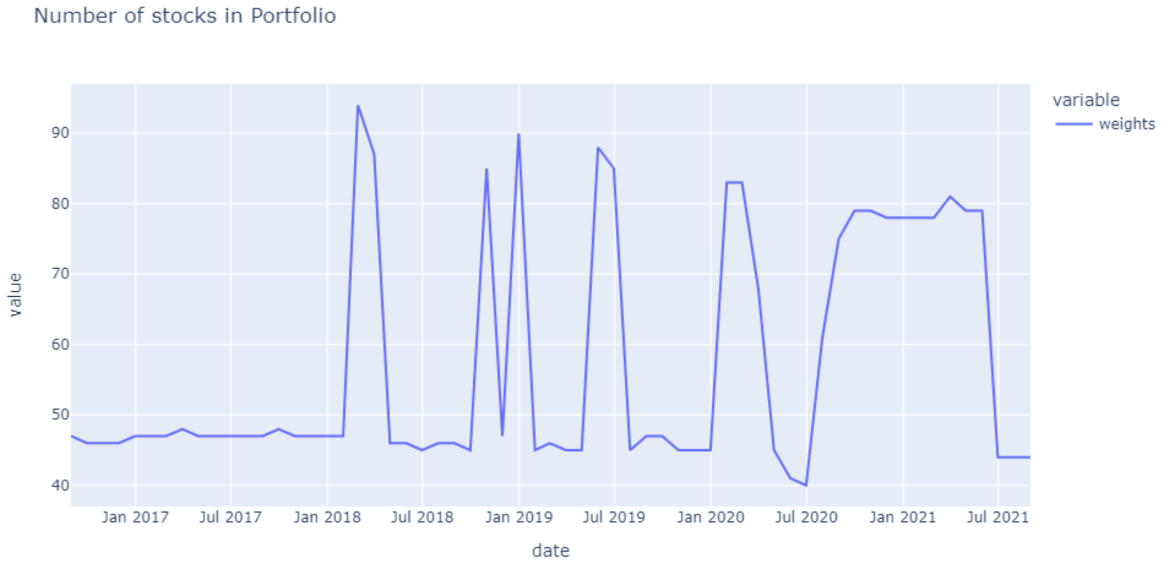


Figure 21: Number of Stocks in Portfolio

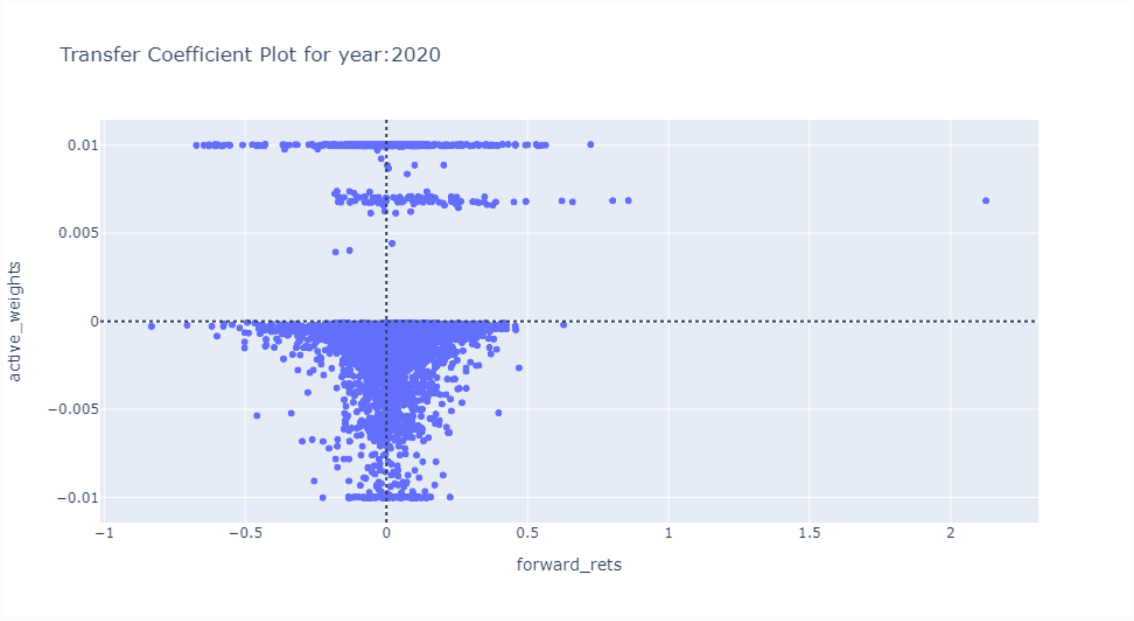


Figure 22: 2020 Transfer Coefficient Plot

Appendix C

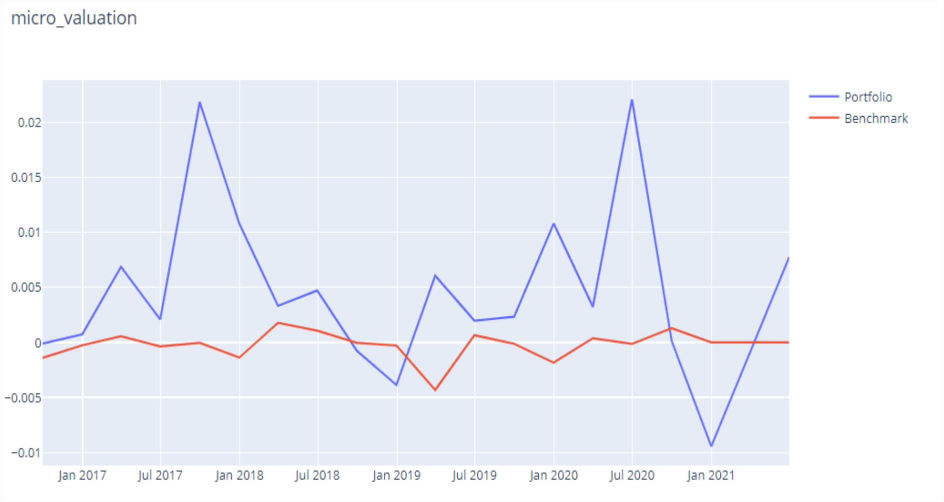


Figure 23: Plot of Micro_valuation factor importance

Appendix D

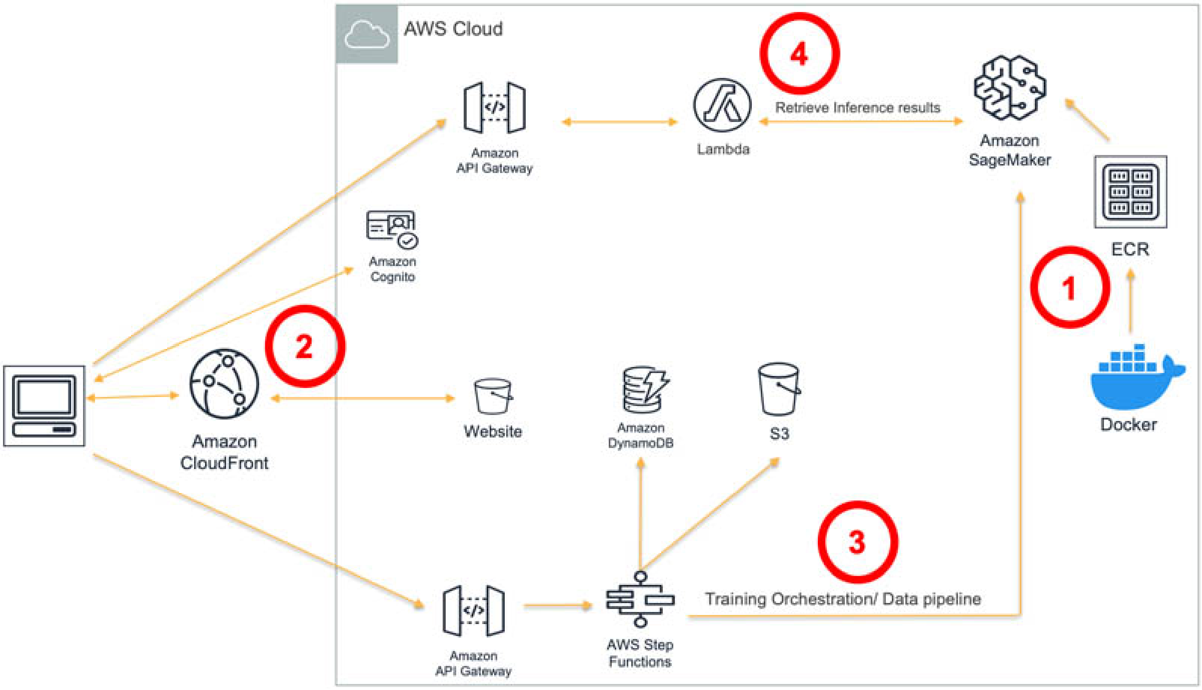


Figure 24: Sample Architecture used for Pipelining

Taken from:
https://aws.amazon.com/blogs/machine-learning/orchestrate-custom-deep-learning-hpo-training-and-inference-using-aws-step-functions/