RO4

FYP Final Report

Optimal Investment Strategy Using Scalable Machine Learning and Data Analytics for Small-Cap Stocks

by

Ashish Kumar Aggarwal, Anwesha Behera and Anish Hiranandani

RO4

Advised by

Prof. David ROSSITER

Submitted in partial fulfillment

of the requirements for COMP 4981

in the

Department of Computer Science

The Hong Kong University of Science and Technology

2017-2018

Date of submission: April 19, 2018

Abstract

Small capitalization stocks, characterised by higher volatility and higher potential returns, are mainly traded by individual investors. The purpose of this project is to provide a platform to these retail investors to invest in a portfolio of small cap stocks listed on NASDAQ by using machine learning and data analytics.

The project was divided into three distinct parts: price prediction, portfolio allocation and a web application. Prediction of stock prices a month ahead was done using a Long Short-Term Memory recurrent neural network and multiple regression methods. Of the two, multiple regression gave better results. Using the results from the multiple regression prediction method, stocks were allocated based on user inputs using a convex optimization library on the basis of Markowitz's Mean Variance Theorem. An interactive web application was made so that the user may interact with the web application easily and compute and visualise the suggested portfolio. Our project outperformed popular market benchmarks for small capitalization stocks in 35 of the 36 simulated months.

Our team worked according to the software engineering and design principles learnt over the course of study to build a modular system with an interactive, user-friendly web application for potential investors.

Table of Contents

List of Figures

List of Equations

1. Introduction

1.1 Overview

Artificial intelligence (AI) and Data Analytics are playing an increasingly important role in investment banking with automated, intelligent systems projected to have as much as US$2.2 trillion of assets under management by 2022, growing at an astonishing compound annual growth rate (CAGR) of 68% [1]. Our project involved developing an optimal investment strategy using AI and Machine Learning technologies.

For practical purposes, our team focussed on a subset of securities: Small-cap stocks in United States listed on NASDAQ with over 15 years of listing data to standardise our results. A similar approach can be taken to invest over multiple asset classes as well as cross-asset portfolios.

Small-cap stocks, with a market capitalization of less than US$ 2 billion, are characterised by low share prices, high volatility, huge growth potential and little to no analyst coverage of individual stocks [2]. Institutional investors, which have large amounts of money to invest, often eschew small-cap stocks due to onerous market regulations and the potential impact of trading them on their actual stock prices [3]. Hence most of the trading in small cap stocks is done by retail (individual) investors.

Retail investors lack the sophistication, access to real-time analytics, market data, resources, expert knowledge and scale of institutional investors and hence make their investment decisions based on publicly available (lower-quality) research, brokerage firms and their own understanding of the economy, industries and market dynamics [4]. In fact, the difference in sophistication and resources is so great that retail investors do not even trade in more complex securities like swaps and forwards. This indicates the need for easy-to-use, smart solutions that can be used by retail investors.

Small cap stocks are more attractive to retail investors due to lower prices, ability to invest in more diversified portfolios with less investment, absence of sophisticated investors and potentially increased opportunities for profit due to higher volatility [5], [6].

One of the effects of institutional investors being absent from small caps market is that research firms do not devote too many resources to analyse small-cap stocks and prepare research reports. For retail investors, because the small-cap universe is so under-reported, there is a high probability that small-cap stocks are improperly priced, offering an opportunity to profit from the inefficiencies caused by the lack of coverage of small-cap stocks. In addition to that, small-cap companies do not publish as much financial data as larger companies. Finding good investment opportunities in small cap stocks is a very time consuming and tedious task, albeit with potentially higher rewards [2].

Our project aimed to study the small cap stocks, understand the market patterns and inherent correlations not so obvious to the current stock valuation models used by investors [7] and then used artificial intelligence and machine learning technologies to suggest optimal investment portfolios to retail investors based on their risk appetite. The focus of the project was to develop an optimal investment strategy for investing in small cap stocks listed on NASDAQ in United States by leveraging emerging technologies like cloud computing, deep learning, and front-end frameworks, following the best software engineering and user interaction practices that we have learnt over the course of our study. The project aimed to provide better returns to investors than the current models and investment strategies used in the industry as well as make it easier for retail investors who lack the resources and expertise to make informed, data-driven investment decisions. The work on the project was divided into three distinct phases: Price Prediction, Portfolio Allocation and web application to interact with the system.

Small-cap stocks are a subset of stock market that is under-analysed yet offers potentially higher returns, has lower prices and is primarily traded by retail investors who lack the resources, sophistication and expertise to make informed decisions in a dynamic market. Leveraging artificial intelligence, data analytics and cloud technologies, the project aimed to provide actionable insights to retail investors, allowing them to make informed, data-driven decisions and obtain better returns on their investments according to their risk profile.

1.2 Objectives

The project was conducted in three distinct parts:

a) Predict price movement of small cap stocks using historical stock data and technical analysis (independent of external factors).
b) Allocate stocks to maximise the return of the portfolio within the risk threshold of the user.
c) Develop a web application that allows user to specify their risk threshold, interact with the system and see the performance of the portfolio over time.

Our ultimate objectives were to:

1. Experiment and evaluate which machine learning algorithms provide us with the best results in terms of stock price projection by conducting time series stock price prediction using techniques like Long Short-term Memory (LSTM) and regression analysis.
2. Present a solution that is comparable in terms of performance to the market standards when measured using industry-specific parameters.
3. Provide the end user with an interactive user interface in the form of a web application to interact with, allowing them to adjust their risk threshold and view their current portfolio and allow changes to suggested portfolio after the most recent computations.

One of the key assumptions made in our project is that no transaction costs are taken into account in the calculation of the suggested portfolio's worth. In real-world scenarios, these would include brokerage fees, market costs like the difference in the buying and selling price of stocks (the bid-ask spread) dependent on the time of transaction, taxes etc. These costs depend on how, where and when the user executes his portfolio after using our platform and hence are considered beyond the scope of this project.

1.3 Literature Survey

1.3.1 Modern Portfolio Theory - Harry Markowitz (1952)

Modern portfolio theory (MPT) was the first attempt at an algorithmic asset allocation. Formulated by Harry Markowitz in 1952, it is an investment theory based on the idea that risk-averse investors can construct portfolios to optimise or maximise expected return based on a given level of market risk, emphasizing that risk is an inherent part of higher reward [8]. It is also known as the mean-variance theory where the risk inherent is modelled as the variance in the expected return of the portfolio while minimizing the variance of the expected return.

This paper is one of the most influential papers in finance and investment theory, as it lays down a mathematical formulation for asset allocation, but it is not without its criticisms. It fails to accurately model real-world conditions, assumes that returns follow a Gaussian distribution, which has shown not to be the case, and relies on the efficient market hypothesis, which again has been proven to be too idealistic.

Our project subscribes to the idea of a diversified portfolio having better returns and lower risk but shares the same reservations about the model’s ability to accurately predict real-world prices [8].

1.3.2 Stochastic Portfolio Theory

Stochastic portfolio theory (SPT) proposed by E. Robert Fernholz in 2002 takes a stochastic approach to stock price prediction as opposed to the deterministic model of

Markowitz [9]. SPT models the prices of individual securities as continuous-time random processes, providing an effective framework to analyse different investment strategies based on real-world market dynamics.

SPT provides a better estimation of price movements when compared to MPT, yet it has its limitations. Mainly, it is a framework to analyse investment strategies not devise new investment strategies which is the purpose of our project [9].

1.3.3 I Know First

I Know First is a fin-tech company that gives daily prediction for stocks, interest rates, commodities and world indices using machine learning and artificial intelligence [10]. Their algorithm identifies relations between different financial assets and predicts future movement. The following diagram illustrates the process of prediction:

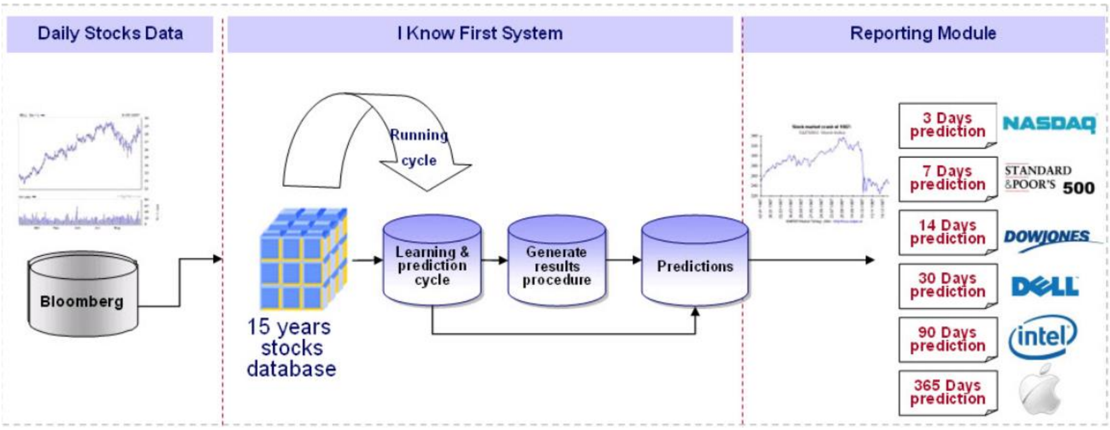

Figure 1: Prediction engine used by I Know First

This technology provides the user prediction of stock prices but does not give any information about the risk involved and fails to provide a holistic overview of the market [10]. Figure 2 shows one of the sample forecasts by "I Know First":

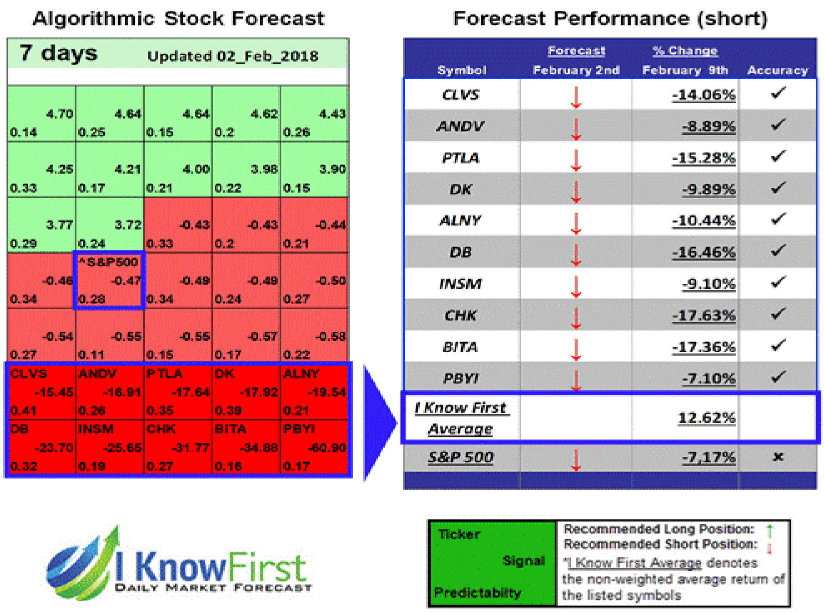

Figure 2: I Know First forecast example

Our group aimed to use a similar technique to predict movement of small-cap stocks using machine learning and help users make better investment decisions, taking into account their risk profiles.

1.3.4 Stock Market Prediction using Neural Networks based on High-Low Point (HLP)

The research paper entitled "Stock Market Prediction using Neural Networks based on HLP" by L.Wang and Q.Wang for the International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC) 2011 introduces a novel way to represent stock price data for prediction using a Neural Network [11]. HLP stands for High-Low Point: It is based on the assumption that though stock movement data is very granular, there is a considerable amount of "noise" in such high frequency data. This paper generates a better prediction engine by taking the highest and lowest price for each stock in a span of given days (a parameter set in the algorithm). The high and low point are then fed into the neural network as features [11].

1.3.5 Applying Machine Learning Techniques to Stochastic Portfolio Theory

Stochastic Portfolio Theory (SPT) is a stochastic analysis framework that analyses the performance of investment strategies and compares them to benchmark indices. The research paper entitled “Stochastic Portfolio Theory: A Machine Learning perspective” [12] addresses the problem of learning an optimal investment strategy using historical data based on a combination of standard and user specified requirements.

The SPT strategy analyses given strategies and picks the one with maximum expected returns. The inverse problem of finding an optimal investment strategy using solely historical data is solved by using machine learning techniques. The new technique results in an in-sample return of 37.54% as compared to the benchmark in-sample return of 13.46% [12].

Our project provides retail investors with a platform that can help them make data-driven, informed financial decisions to possibly increase their expected returns. Our research covered existing investment theories like ‘Modern Portfolio theory’ formulated by Harry Markowitz in 1952 [8], ‘Stochastic Portfolio theory’ proposed by Robert Fernholz in 2002 [9] as well as more recent topics in portfolio management spurred by the advancement of artificial intelligence, machine learning and data analytics technologies. The literature survey covered Stochastic portfolio theory using machine learning [12], algorithmic portfolio strategy as used by ‘I Know First’ [10] while L. Wang and Q. Wang discuss a novel method of preprocessing stock data using High-Low point (HLP) [11]. The idea was to look at the investment strategies used in the industry as well as to get acquainted with the emerging techniques and models for optimal portfolio allocation.

2. Methodology

The project had three main areas of focus as described in the next section. Our team followed the ***phased software development process.*** This ensured that through ***iterative development*** our team worked in these areas simultaneously to incorporate adaptability and improvise as needed throughout the course of the project.

2.1 Design

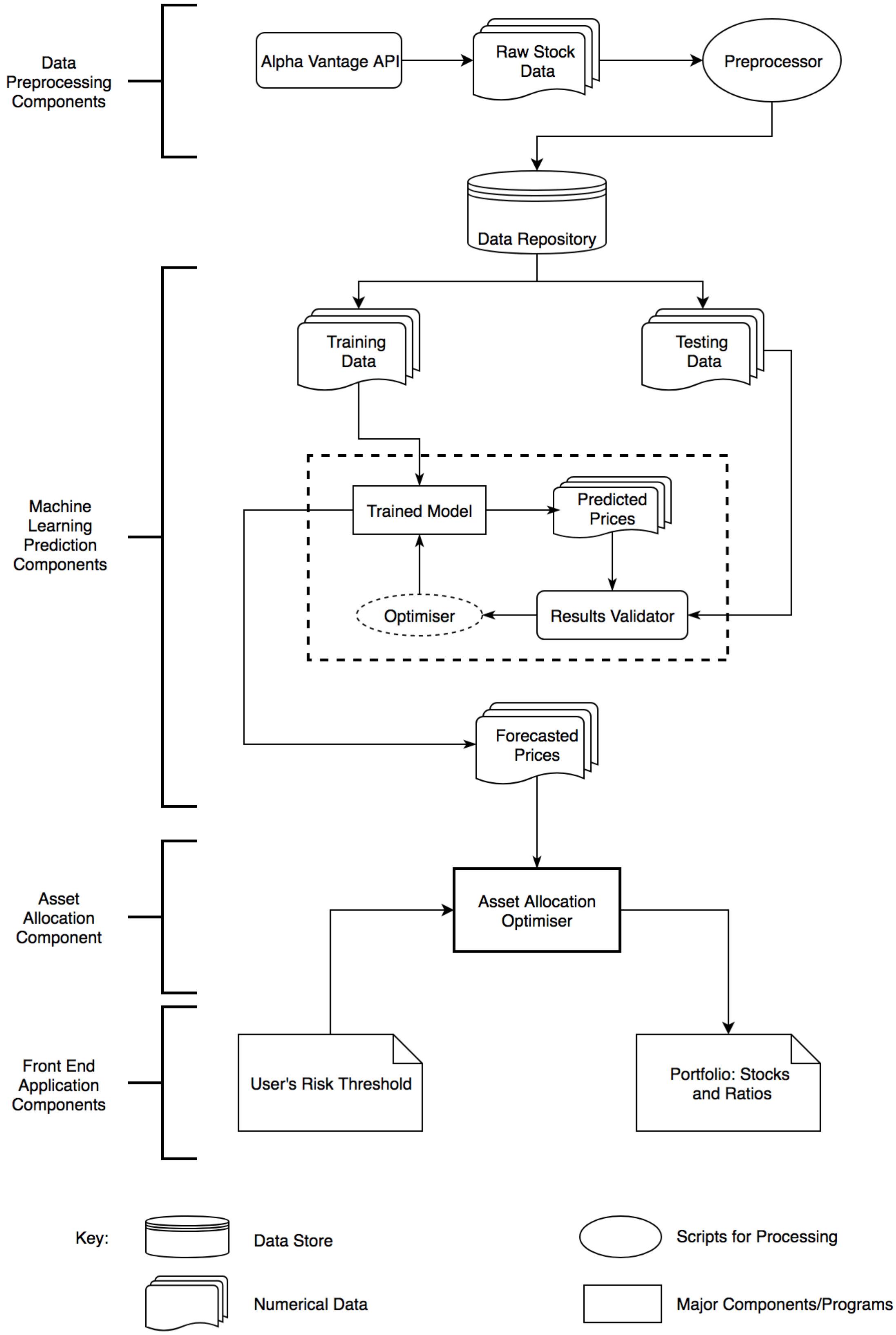

Figure 3: System Architecture and Flow

Our project was divided into 3 main aspects:

2.1.1 Prediction of prices using Machine Learning

2.1.1.1 Price Data Feed

Every NASDAQ small-cap stock's price would be updated using a data feed from Alpha Vantage [13], a reliable API for equities data. These updates would then be appended to the data file storing historical prices in the desired format using a Python script.

2.1.1.2 Machine Learning Model

The machine learning model is implemented in Python. It obtains historical prices from the aforementioned data file and segregates it into training data and testing data in the ratio 70:30. The training data is used to build the machine learning model, and the testing data is used to validate the model's results.

2.1.1.3 Predicted Prices

The model's results - the predicted price data points are saved and later used by the asset allocator to aid the computation of an optimal portfolio for the end user.

2.1.2 Asset Allocation

2.1.2.1 User Input

Portfolio allocation takes the maximum volatility threshold and suggested number of stocks in the portfolio as user input. This allows the program to adapt to the user's needs, portfolio breadth and risk profile. The aim of taking user input is to provide flexibility to the user and to make sure that the system is accessible to different types of users. For this project, the allocator rebalances the portfolio monthly, however, that time frame can be customised based on users' needs.

2.1.2.2 Asset Allocation

The allocation script takes the predicted values of the next month as generated by the prediction model and computes the month-end predicted value and risk statistics for each stock. Allocation is done for the last trading day of the next month since we assume that portfolio will be rebalanced on the last trading day of each month.

2.1.2.3 Sorting Parameter

The allocator sorts stocks on these three parameters and picks up the number of stocks specified by the user in descending order:

a) Risk Efficiency: ratio of return/risk for predicted values
b) Absolute Return: absolute predicted return of each stock for the month
c) Lowest Risk: stocks with lowest levels of risk

2.1.2.4 Mean Variance Optimization

Asset Allocation is done using Mean Variance optimization by building a covariance matrix of the selected stocks and 5 years of their historical data using predicted month-end prices as the expected return of each stock.

2.1.2.5 Portfolio Output to Web Application

A long-short portfolio of stocks is generated having a risk factor less than user's risk threshold. The generated portfolio is returned this output to the web application as a suggestion to the user.

2.1.3 Web Application Component

The web application serves as a platform for the user to interact with the whole project's system. To help express and evaluate our design ideas, low-level prototypes of the web application pages were built that acted as the skeletons for individual pages of the web application. Process flow diagrams for each feature were created to emulate the user interaction flow.

Our application offers three main services to the user:

a) Stocks Explorer

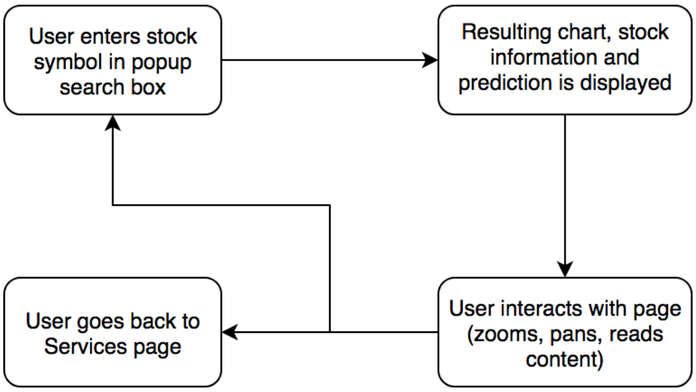

Figure 4: User interaction flow for Stocks Explorer

This page allows the user to search for stocks dynamically and view their past prices along with our model's prediction for the upcoming month. With an interactive set of tools, the user would be able to zoom and pan across the graphs for their own reference and increase their understanding of the NASDAQ small-cap stock market in general.

b) Portfolio Manager

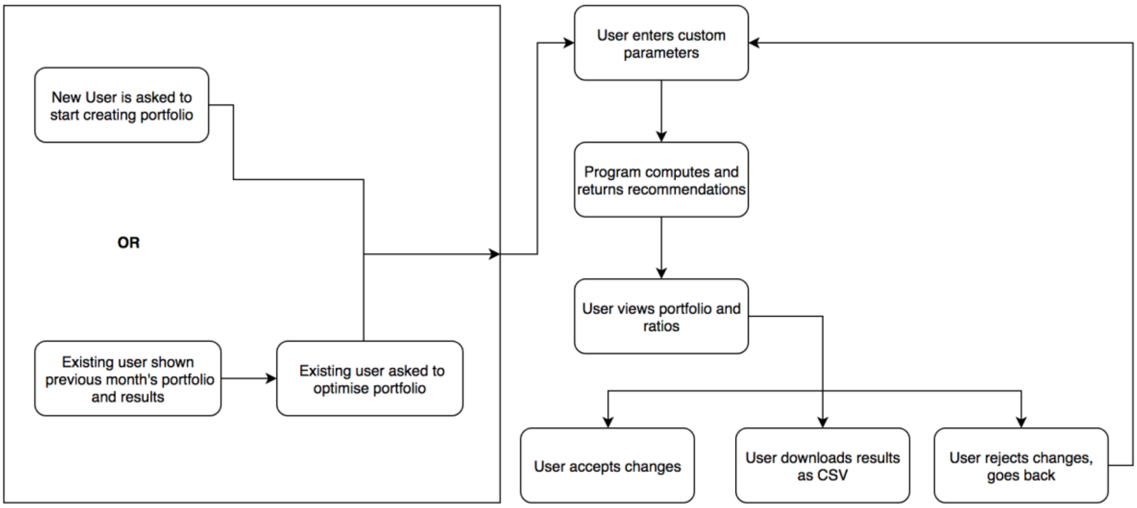

Figure 5: User interaction flow for Portfolio Manager

As the main feature of the web application, the portfolio manager provides a single page intuitive view of the user's current portfolio, giving the user crucial information at a glance. If the user chooses to update and optimise his portfolio for the month, the application would provide the user a form to enter his preferences like the maximum volatility the portfolio can take for the next month (**r)** and maximum number of stocks (**n)**. The application would compute and fetch up to **n** best stocks to invest in and in what ratios, giving the user the freedom to either accept the proposed changes to the portfolio or dismiss them.

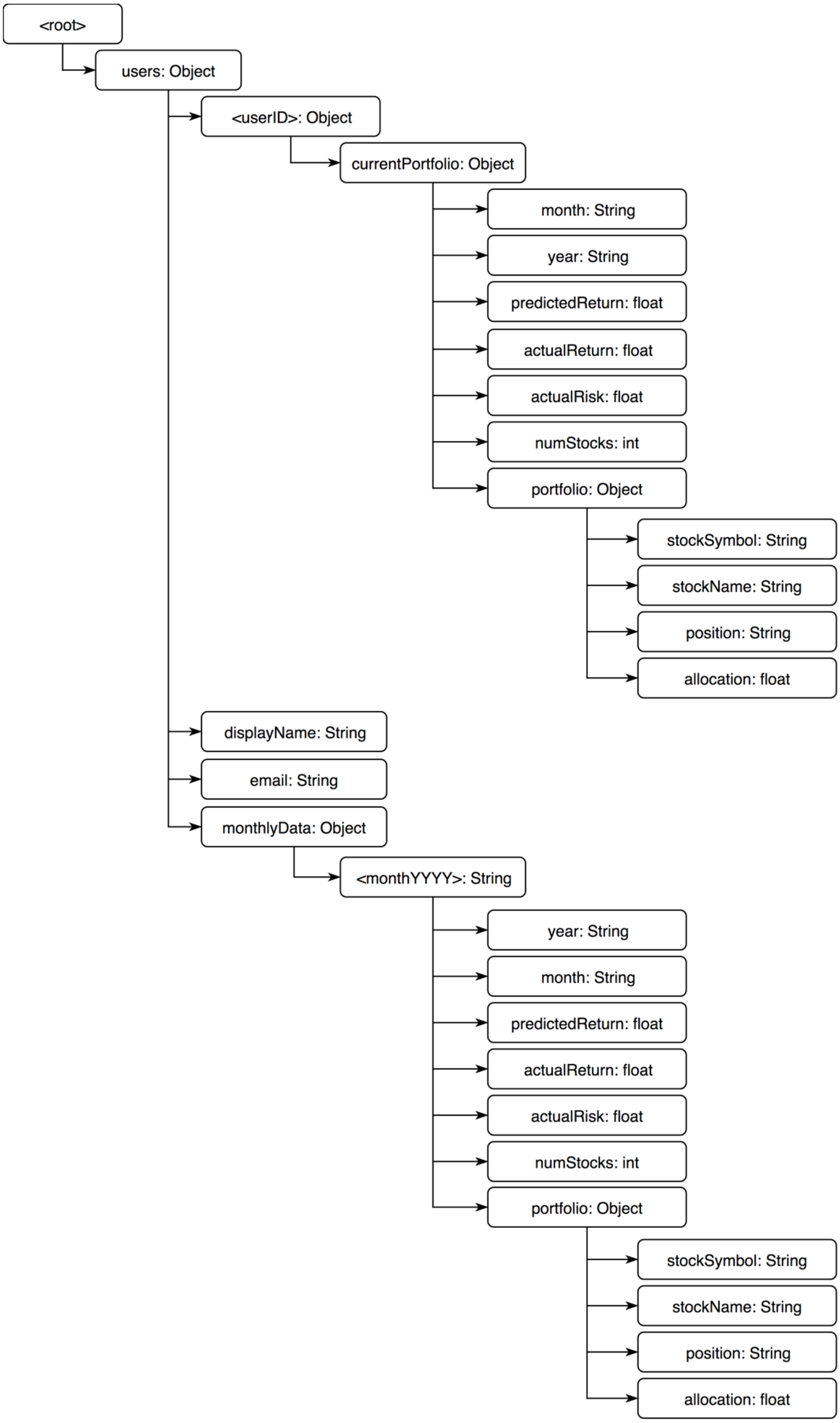

Figure 6: Model for User's Portfolio Data

c) Portfolio Growth Analyser

This feature would act as a testament to the project's performance. The page provides a graphical view of how the user's portfolio has fared over time. An array of benchmarks i.e. exchange traded funds (ETFs) that the user could have alternatively invested in and the Russell 2000 market index are compared vis-a-vis the user's portfolio and its growth. The user can also compare the machine learning model's predictions for portfolio growth with the actual growth.

2.2 Implementation

2.2.1 Price Prediction Model

2.2.1.1 Study of algorithms

Our team studied the concepts and techniques related to data analysis in particular the machine learning techniques and implementations, and read research papers on similar market data analyses.

2.2.1.2 Collection, preprocessing and standardisation of stock data

As our project deals with the small-cap stock market, our team used Zacks' Stock Screener Tool [13] to obtain the lists of stocks, their symbols and metadata listed on NASDAQ. The stocks symbols were processed to remove inconsistencies, especially for preferred stocks whose symbols' format was different from what was used to query the Alpha Vantage API, for instance SEN.A changed to SENA. Other types of inconsistent data included stocks that no longer operated in the market and had a value of US$ 0.

A Python script was written that used Alpha Vantage's API to retrieve stock price quotes for all the stocks in the above list. This program fetched quotes from 1 January 2000 till 29 September 2017, later updated to retrieve data till 28 February 2018. Our team noticed that a few stocks, having had their initial public offerings (IPOs) much after 2000, had very few data points in comparison to other more seasoned stocks. To standardise the number of data points across all stocks, our team eliminated stocks that were not on the market on or before 1 October 2000. We reformatted the data to compile 250 stocks' prices into one document for easier handling by the machine learning algorithm. After the data was prepared, our team used Python and TensorFlow to read in the quotes and implement the LSTM network described in the next section. This program was run on an AWS EC2 server (refer to Appendix D) to train the model and deal with heavy computations.

2.2.1.3 Implementation of Long-Short Term Memory (LSTM) Neural Network

An LSTM block is a simple recurrent neural network (RNN) that is used to build larger neural networks to perform complicated tasks. The need to use LSTM (and recurrent neural networks in general) arises when dealing with predictions that involve dependence on historical data or data that depends on a sequence of events rather than one single event. Some useful applications of this architecture are time-series prediction, speech recognition and music composition. A cell typically contains three gates, namely: input, forget and output. These gates are responsible for controlling the flow of information in the neural network and act as artificial neurons themselves. A typical LSTM cell [15] is illustrated in Figure 7.

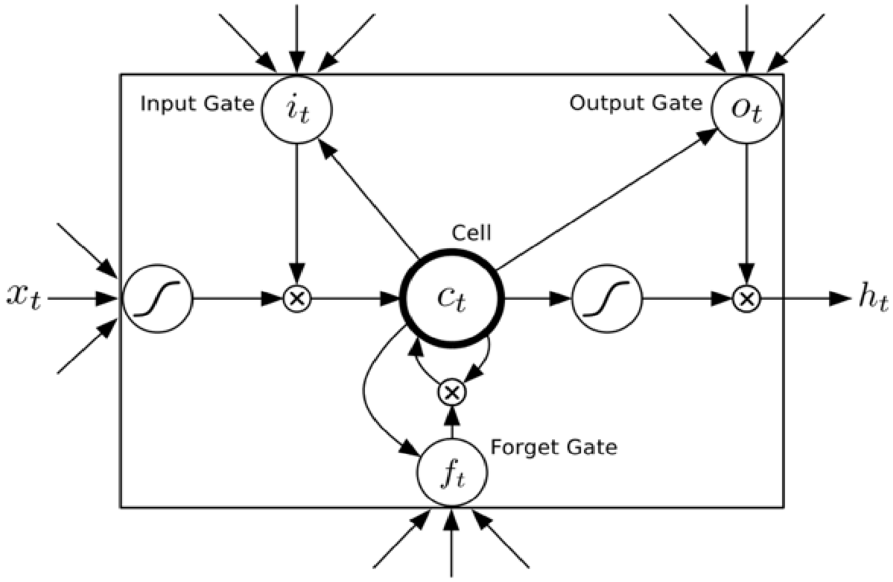

Figure 7: A typical LSTM cell

$$f_t = \sigma_t(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c (W_c x_t + U_c h_{t-1} + b_c)$$
$$h_t = \sigma_t \circ \sigma_h(c_t)$$

Equation 1: Update equations of LSTM cell

In Equation 1 above, at any time-step t:

- All W's, U's and b's are weight parameters that need to be learnt during training
- x: input to the LSTM cell
- f: forget gate activation vector
- i: input gate activation vector
- o: output gate activation vector
- h: output vector of cell
- c: cell state vector
- σ_g: sigmoid function

- σ_c: hyperbolic tangent function

General RNNs are unable to learn long dependencies because of the vanishing gradient problem which occurs due to the error becoming very small in magnitude as the time interval between events increases. In an LSTM cell, when errors are back-propagated from the output, they remain in the block's memory and the error is continuously fed back in each of the gates until training is complete. This makes backpropagation effective in training LSTM to remember information for long durations. In our case, we believed that this would help us handle historical time-series stock data better. This is what prompted us to use LSTM for our Stock Prediction part.

Using LSTM, we tried to model stock prices as a function of historical data. In our experiments, we varied the network architecture (number of hidden units) - the following results are obtained using 50 hidden units. One of the parameters of the network which can be changed is "sequence-length" - sequence length determines how long back in history do we go to search for patterns to predict future stock prices. As a first attempt, we trained the network with sequence length = 10, i.e. using 10 consecutive days stock prices to predict the following day's (11th day) price. For the testing phase, we used real stock data to predict prices and test our model. Following is a chart showing results for one of the stocks chosen randomly, Ameris Bankcorp (symbol: ABCB). In Figure 8, areas where the red and blue lines superimpose the actual stock price line indicate that the predicted values closely follow the real values.

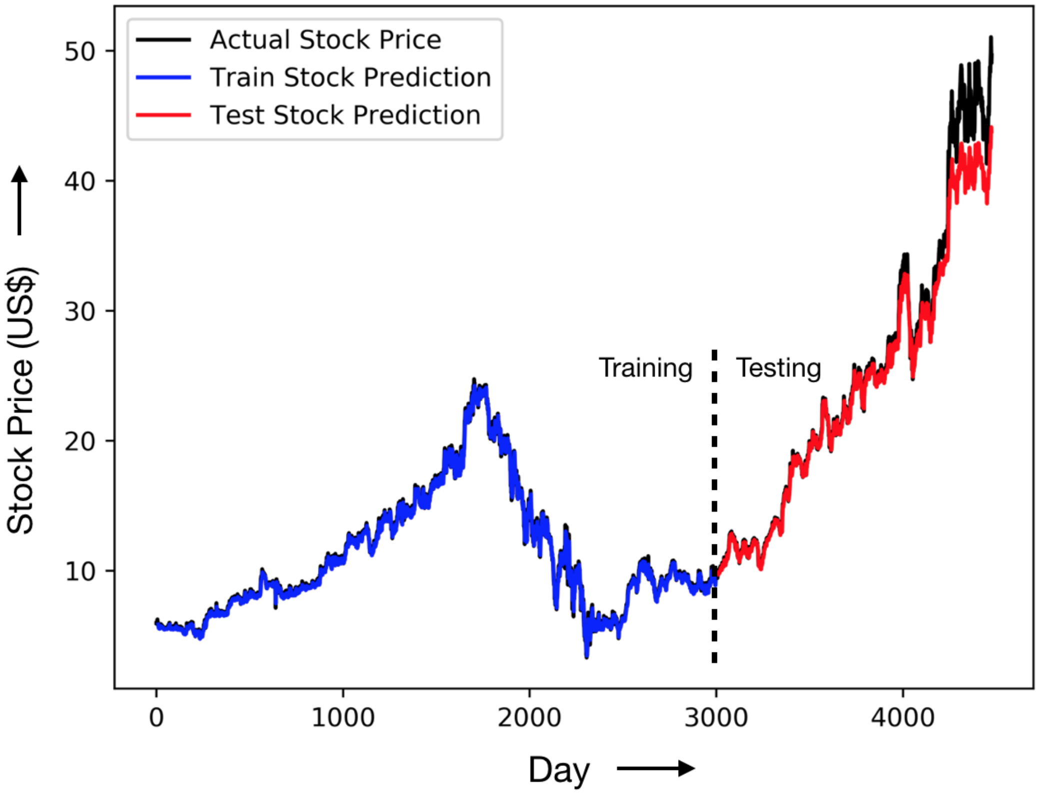

Figure 8: LSTM model results for ABCB using real data in Testing phase

Our team further implemented a more dynamic model through which we tried to find the optimal sequence length for each stock which would give us the most accurate forecasts. To do this, we compared the test RMSE, or root mean square error (loss function) for sequence lengths from 5 to 300 and stored the value for which the error was minimum. To compare the results with a statistical model, we plotted the x-day simple moving average where x = optimal sequence length found for LSTM.

Since we will never have future prices for the testing phase (assuming all historical data is used to train the model) forecast prices for the future, we used predicted values to make our predictions in the testing phase. Due to this, small errors in predictions increased in magnitude and caused a flat line. The main issue that we faced with this approach was we were only able to forecast stock prices for one day in advance. Since we wanted a 30-day forecast as the output from our machine learning model, we modified the input to be such that it takes 7 consecutive days data to predict predict prices one month ahead of the last day in the sequence. Our training model consisted of 2 layers having 15 units each. We stored predicted results for a period of 1 year (March 2017 - February 2018) and compared results with other models described later. This machine learning technique was implemented using Sequential and LSTM functions in the Keras library using tensorflow backend. Our team created tensors of appropriate size from the time series data for training, testing and forecasting.

Loss function:
The loss function used here is Mean Squared Error. The neural network uses a Root Mean Square Propagation (rmsprop) optimiser to minimise the mean square error between the actual values and the real values.

2.2.1.4 Implementation of Linear Regression model

2.2.1.4.1 Simple Linear Regression

Simple Linear Regression aims to examine the relationship between 1 dependent variable and 1 independent variable. As the name suggests, this model looks to find the best linear representation of this dependence. The hypothesis here is that the dependence between the 2 variables can be expressed as a linear function which is believed to be a good estimator of the true function (usually of unknown form). For example, consider an independent variable x and a dependent variable y, if these are related by a function f such that y = f(x), where f is a linear function of the form:

$$f(x) = \beta_1 x + \beta_0$$

Equation 2: Simple Linear Regression equation

with β_1 and β_0 as parameters of the model.
The goal of this model is to find the parameters β_1 and β_0 for function f which best estimates the true relationship between the variables. This machine learning model was implemented using sklearn’s Linear Regression module by creating matrices of appropriate size from the time series data for training, testing and forecasting.

Loss function (Least Squares):

During the training phase, the algorithm optimises the values for β_1 and β_0 by minimizing the sum of squared residuals, where residuals are differences between the expected value and observed value of a quantity.

Measure of Accuracy (R^2 loss) [16]:

The coefficient R^2 is defined as

$$R^2 = 1 - \frac{\sum(y_{true} - y_{predicted})^2}{\sum(y_{true} - \frac{\sum_{i=0}^{n} y_{true}}{n})^2}$$

Equation 3: R^2 loss definition

The best possible score is 1.0 and a constant model that always predicts the expected value of y, disregarding the input features, would get a R^2 score of 0.0.

In our experiments, we modeled the stock price on day *t* as a linear function of the stock price on day *t-30.* The hypothesis of this experiment was that the best predictor of a stock price on a given day is the stock price 30 days prior. In addition, this helped us to fix the forecasting issue we faced in LSTM predictions (not being able to predict prices more than 1 day in advance accurately).

$$StockPrice_t = \beta_1 \times StockPrice_{t-30} + \beta_1$$

Equation 4: Simple Regression Line of Fit

In Figure 9, areas where the red and blue lines superimpose the actual stock price line indicate that the predicted values closely follow the real values.

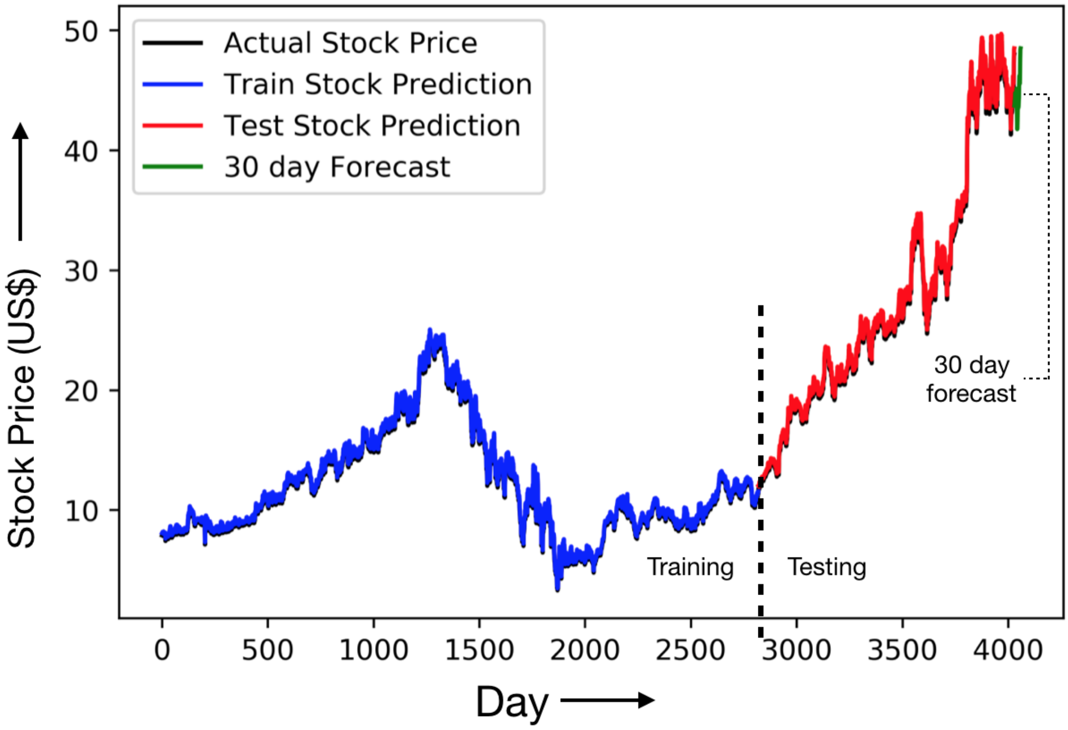

Figure 9: Results of Linear Regression model for ABCB with 30 days forecast

2.2.1.4.2 Multiple Linear Regression

The above concept of simple linear regression can be extended to incorporate multiple independent variables. In this case, the model takes the form, $y = f(x1, x2, \ldots, xm)$ where $x1, x2, \ldots, xm$ are m independent variables and f is of the form $\beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_{1m} x_m + \beta_0$

In our experiments, we modelled the stock price on day *t* as a linear function of stock prices on days $t - 30$ and $t - 60$.

$$StockPrice_t = \beta_1 \times StockPrice_{t-30} + \beta_2 \times StockPrice_{t-60} + \beta_0$$

Equation 5: Multiple Regression Line of Fit

We evaluated similar models for stock prices of different combinations of historical data and observed that predicted prices were closest to real prices for the multiple linear regression model described above.

2.2.2 Asset Allocation Model

The asset allocation model takes the predicted prices from the prediction model and seeks to devise an optimal portfolio based on the user's risk threshold. It starts with preprocessing data so that it is easier to work with it. Optimization is done by optimizing the convex curve of the Markowitz portfolio generated using a covariance matrix. The aim is to keep the asset allocator modular in design so that it can easily adapt to the user's dynamic needs.

2.2.2.1 User Input

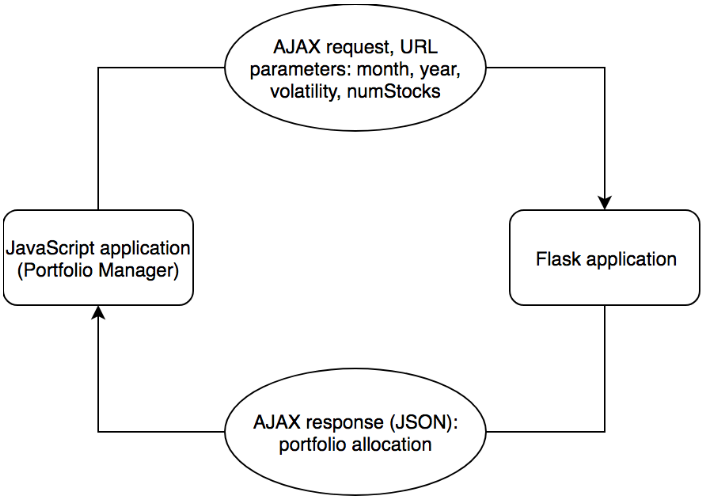

Figure 10: Interaction between JavaScript and Flask applications

The asset allocator is called by the web application as an AJAX call to the Flask app and returns the stock symbols and their respective ratios and trade type as JSON to the web application. Web application passes the user inputs to the Flask application via the AJAX call.

2.2.2.2 Preprocessing and Consolidating data

The predicted price data is collected and consolidated into a single data file for the month under 9 columns:

a) Stock: represented by stock symbol
b) Previous Price: price at the end of current month
c) Actual Price: price at the end of the next month (Only for testing, not used in

allocation)

d) Predicted Price: predicted price at the end of next month
e) Normalised Risk: volatility, a proxy for the risk measure, is an indicator of how confident we are in our price prediction. It is calculated as a standard deviation of the predicted prices for the upcoming month normalised based on the previous price
f) Return: monthly return based on the predicted price change as a percentage of last month's actual price
g) Absolute Return: absolute value of the return
h) Actual Return: actual monthly return calculated as the percentage difference of actual price from previous price, only used for testing purposes
i) Risk Efficiency: ratio of absolute return to normalised risk

2.2.2.3 Optimization Parameters

Based on the consolidated script, various optimization parameters can be calculated:

a) Risk Ratio: the ratio of expected return to volatility of the stock
b) Absolute Return: the absolute expected return of a stock based on the Predicted and previous price
c) Risk: the program may rank stocks by risk in ascending order

These optimization parameters are calculated using the data in the data file and are used for portfolio allocation. These parameters help pick the number of stocks specified by the user so that the portfolio can be allocated for those stocks.

2.2.2.4 Covariance Matrix

Based on the selected stocks, we build a covariance matrix of the selected stocks. The covariance matrix is built by taking into account the real end of day closing data for all trading days in the last 5 years.

The covariance matrix is built for portfolio balancing as our aim is to make a profit no matter the direction of market movement and isolate the effect of single stocks on the total portfolio value. Using a covariance matrix, we can evaluate how stocks have moved in relation to each other to minimise the variance of expected return of the portfolio and hence reduce the total risk taken by the investor.

2.2.2.5 Mean Variance Optimization

Mean Variance Optimization is based on Markowitz's theory of Portfolio Allocation first published in a paper "Portfolio Selection" in 1952 [8].

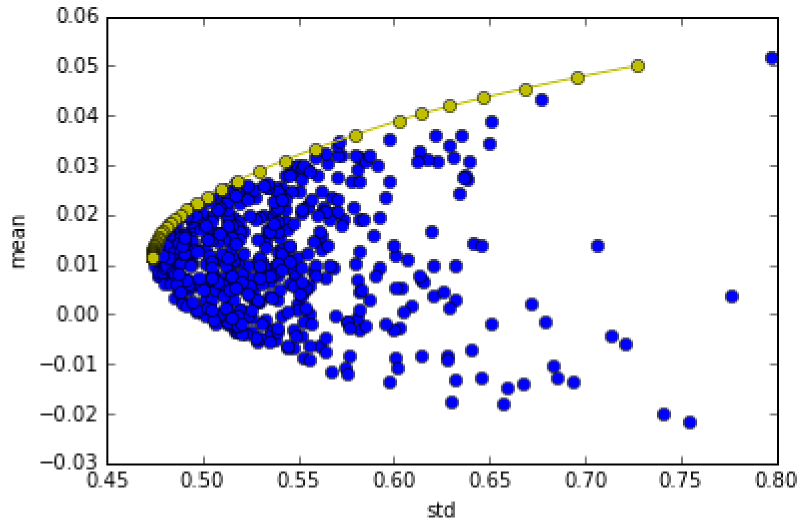

Figure 11: Markowitz Bullet [18]

The paper states that if random weighted portfolios are plotted over a return v/s risk graph, they will form a convex curve called the Markowitz' bullet. The yellow line is called the efficient frontier which denotes the best return that a portfolio can get for that level of risk or the minimum risk a portfolio can have for a particular value of expected return. Our script uses the 'cvxopt' library to optimise for the convex curve.

Our script uses this theory to suggest optimal portfolio for the user. Given the expected return, risk measure and the covariance matrix, we set the following restraints:

a) No value of absolute ratio can be less than zero.

b) The total of all the weights in the portfolio is equal to 1.
c) Expected return of the portfolio is greater than or equal to the minimum return. This minimum return is set dynamically by the script to maximise the return within the user specified risk threshold.

If the expected risk of the portfolio is less than the user specified risk, we increase the minimum expected return of the portfolio till the risk, which is recalculated after each allocation, is near the user specified risk. The script outputs the portfolio that has the maximum expected return within the user specified risk threshold.

2.2.2.6 Output to the Web Application

Based on the user input of risk threshold and desired number of stocks in the portfolio, the results are generated by the asset allocator. These results are then returned to the AJAX caller in the web application in the form of a JSON response (Figure 10) to make it compatible with the NoSQL database used by the web application to store user data.

2.2.3 Web Application

The website has been implemented using a variety of front-end programming languages and frameworks like HTML, CSS, AngularJS, jQuery. Using Bootstrap as our library for basic webpage components and styles, our team employed D3.js to build interactive data visualisation tools. The application logic is written in AngularJS, as it provides useful features like two-way data binding and dependency injection over native JavaScript.

Figure 12: Screenshot of Home Page for project's web application

2.2.3.1 Logging in or Registering

After the user reads through the descriptions of the services the application provides them, they may click on the Login/Signup button. The project redirects the user to a Login/Register web page that offers multiple authentication methods to create an account in the application: Google, Facebook or the user's email (for a standalone account).

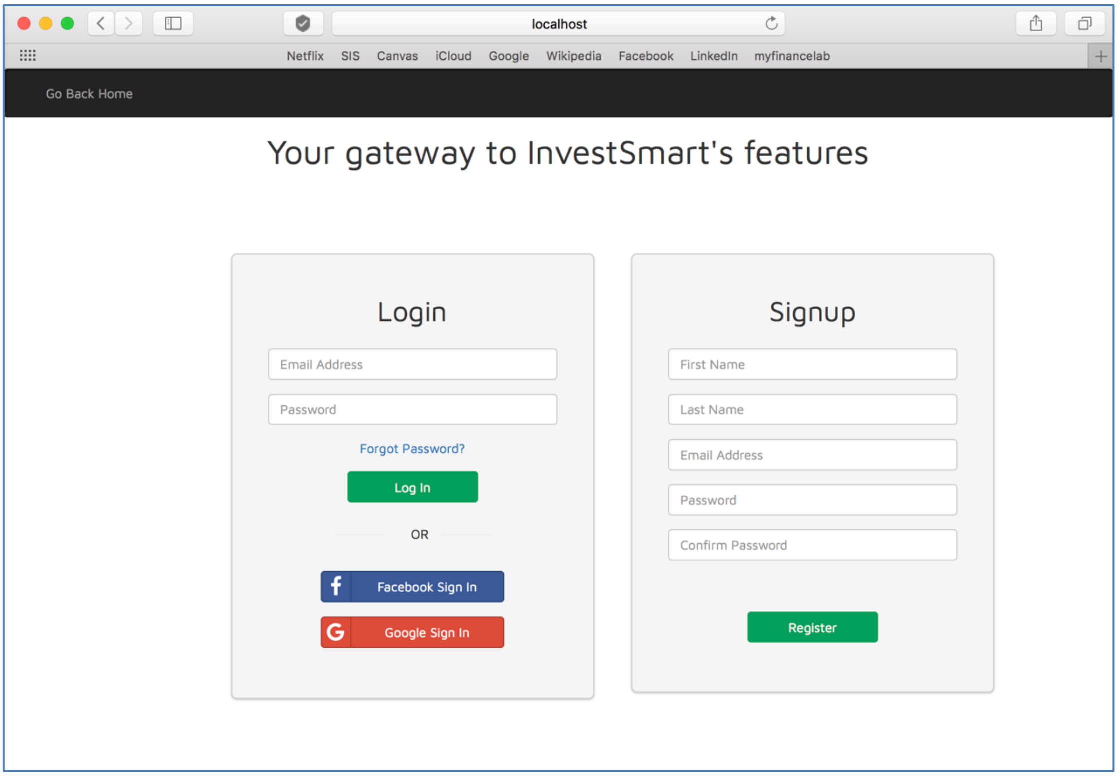

Figure 13: Screenshot of Login/Signup Page

The user, on deciding to use Google or Facebook to sign in, is asked to authorise the usage of their name and email information for the application. The security and ease of access that popular accounts like Google and Facebook provide to their users motivated our team to integrate these services' APIs into our application. Firebase's own Authentication service is used to connect and securely transmit access tokens and Firebase's NoSQL database is used to keep track of profile information should the user choose to register using his email instead.

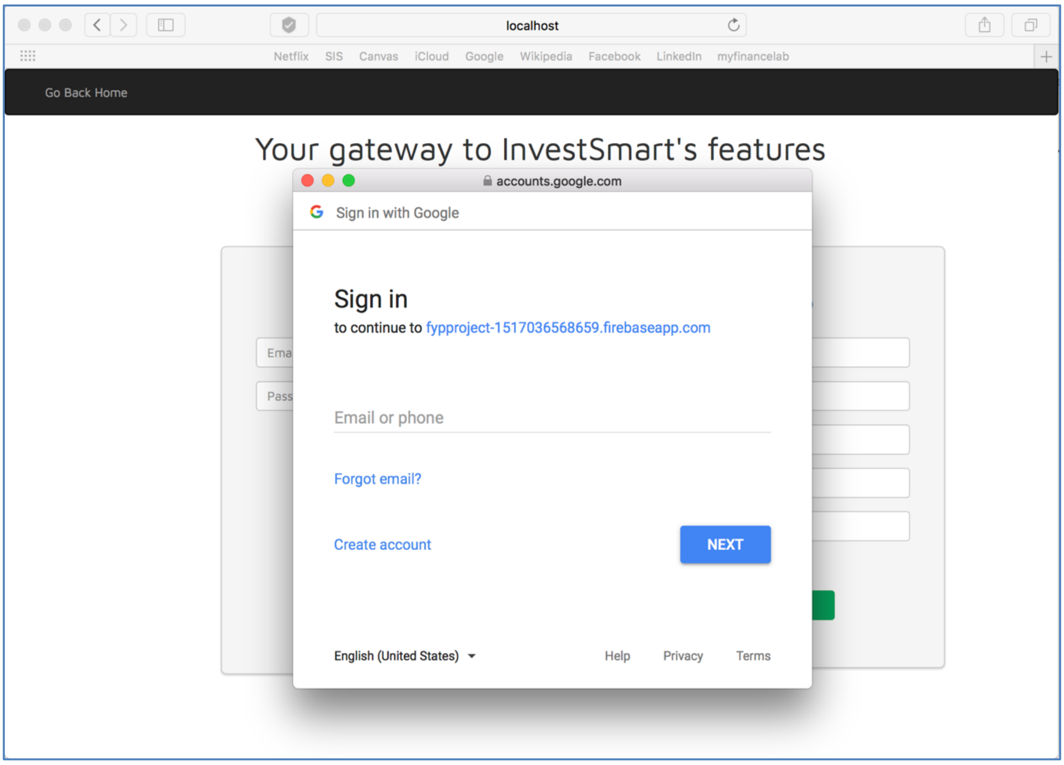

Figure 14: Screenshot of sign-in popup via Google

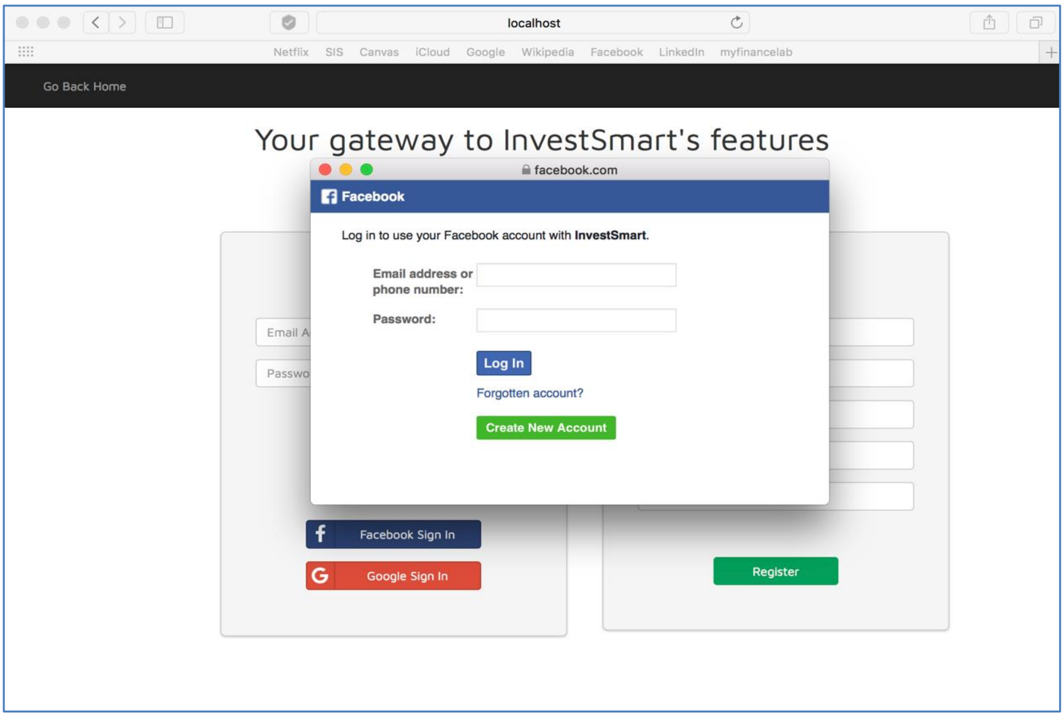

Figure 15: Screenshot of sign-in popup via Facebook

2.2.3.2 Services Page and Main Features

After the sign-in process is completed successfully, the website redirects the user to a Services page that provides the user links to the three features as described in Section 2.1.3. (shown as the three boxes in Figure 16). This page displays the users' profile image from either Google or Facebook if signed in through those platforms, or shows a placeholder profile image if the user has created a standalone account on the web application. The user may sign out at any time either via the button at the centre of the Services page or the button on the top right corner in any other Feature page.

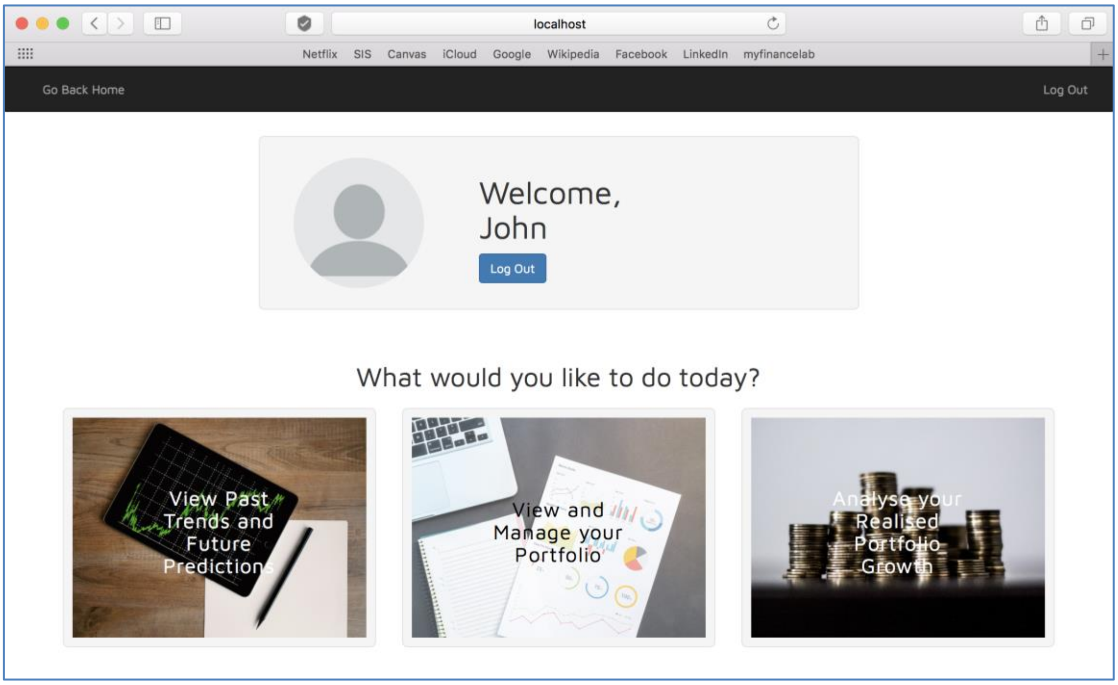

Figure 16: Screenshot of main Services page of web application

2.2.3.3 Stocks Explorer

The **Stocks Explorer** renders the historical prices stored in the data repository as charts for each stock using D3.js. The page also provides brief information about the stock itself and displays the predicted price (according to the machine learning model) along with a ticker indicating its increase or decrease in the upcoming month.

Interactive gestures such as zooming in, dragging to pan across the graph, along with a mini-graph for time period context have also been implemented. These gestures have been implemented using D3.js to model each individual behaviour and component.

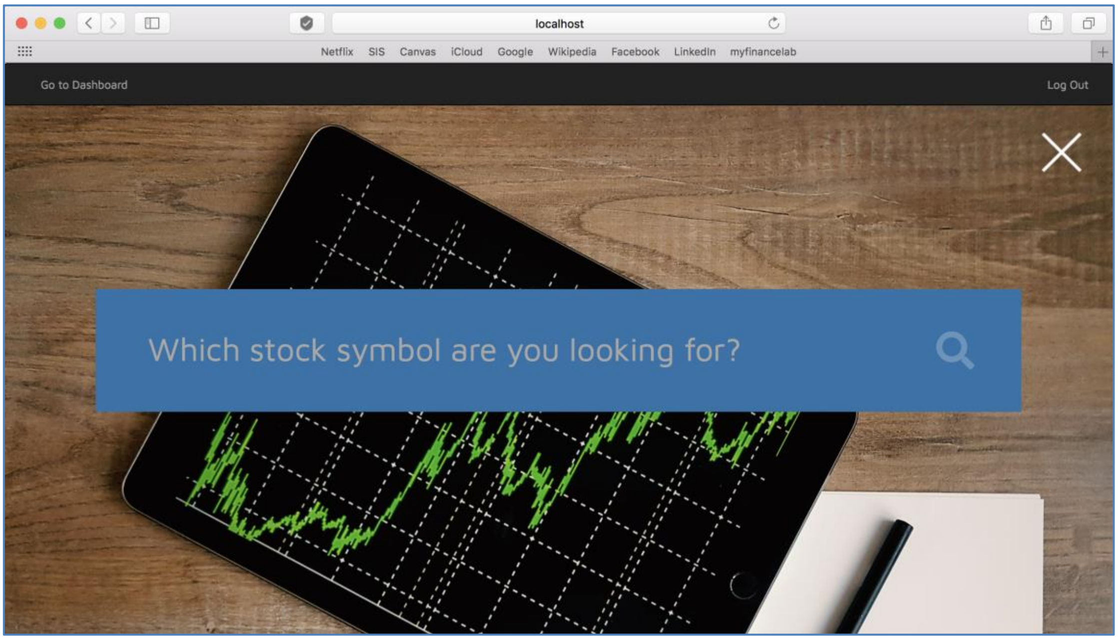

Figure 17: Popup screen for querying stocks in Stocks Explorer

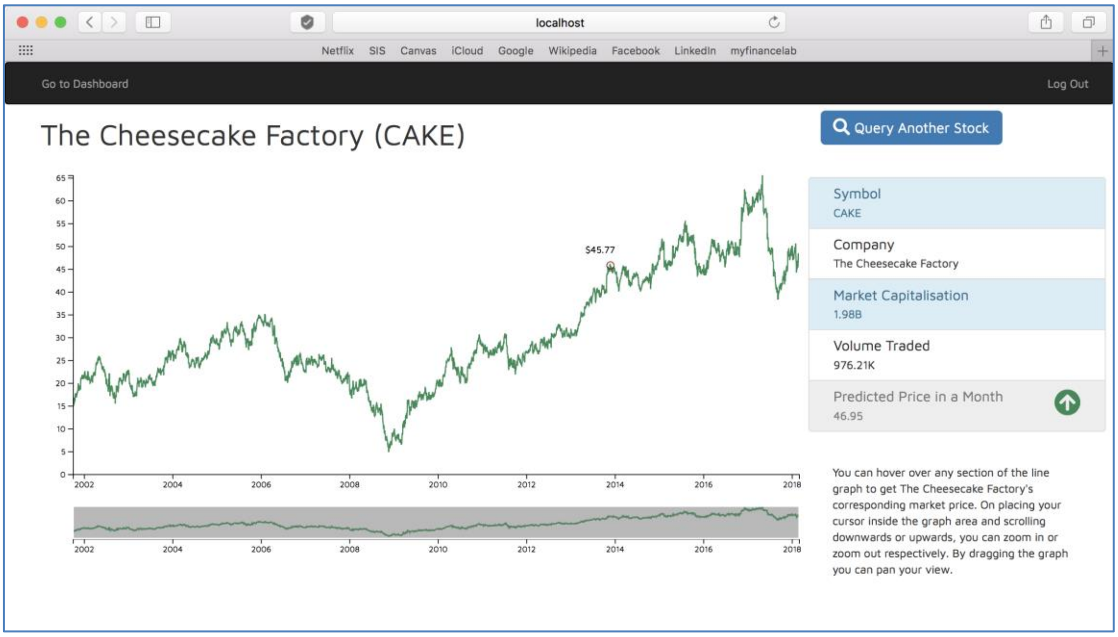

Figure 18: Screenshot of result in Stocks Explorer

2.2.3.4 Portfolio Manager

The **Portfolio Manager** first detects if the user is new or has used the portfolio manager. It accordingly either prompts the user to start making use of this service, or renders their portfolio from the previous month.

- The previous month's portfolio is displayed using a donut chart built using D3.js. Other important data such as the month's growth percent, number of stocks and volatility are displayed as well, as seen in Figure 19.

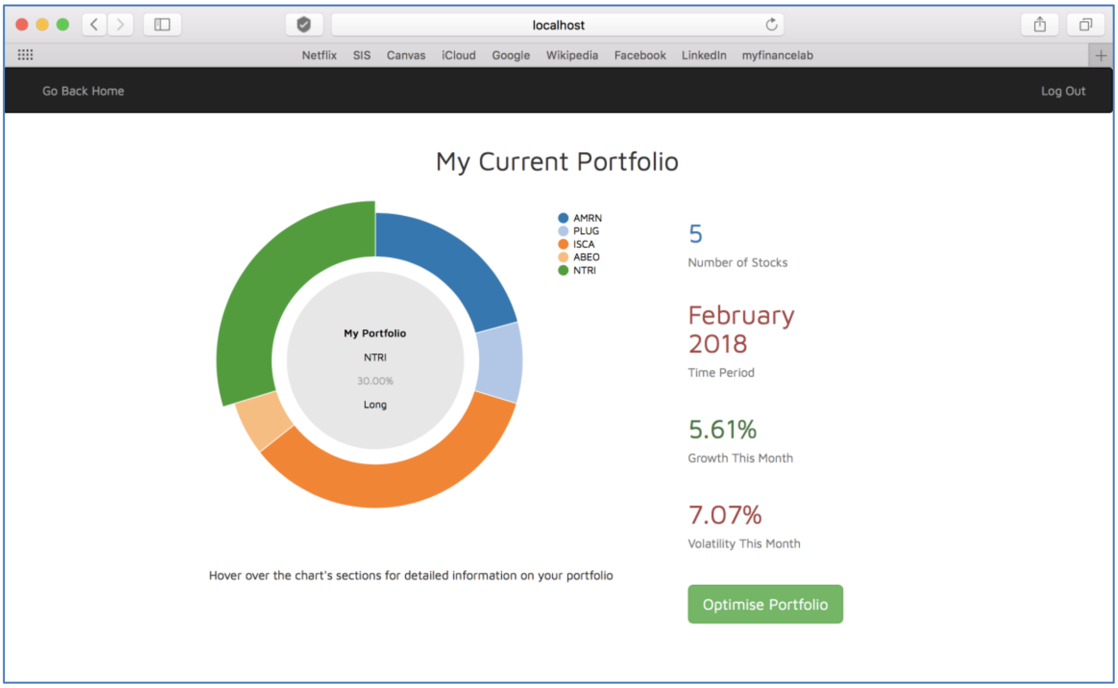

Figure 19: Screenshot of Portfolio Viewer

- If the user chooses to optimise their portfolio for the current month, the page is redirected to a form where they are asked to enter the maximum number of stocks they would like the allocator to pick. If the user is new, the form also asks to enter up to how much volatility the user is willing to take (Figure 20).

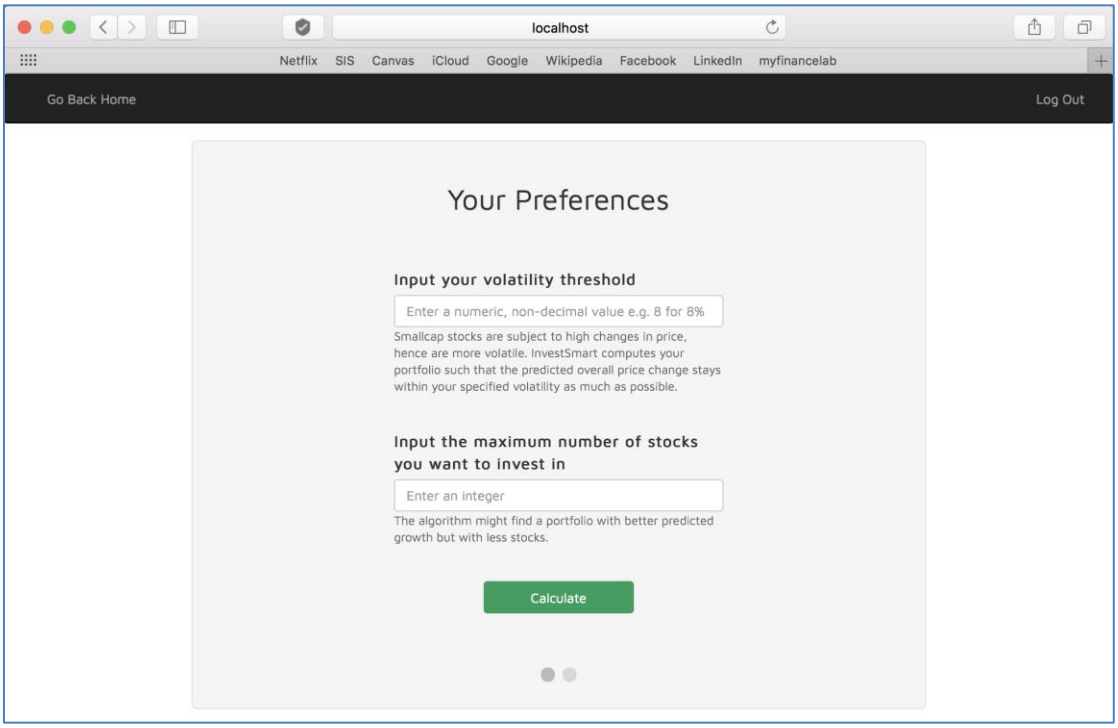

Figure 20: Screenshot of Portfolio Optimiser Input window

- The portfolio allocator uses the month's predicted prices to accordingly compute a combination of stocks customised according to the user's input parameters. The user may choose to accept the changes, reject and edit the input parameters in the previous page, or save the results as a CSV on their local computer system (Figure 21).

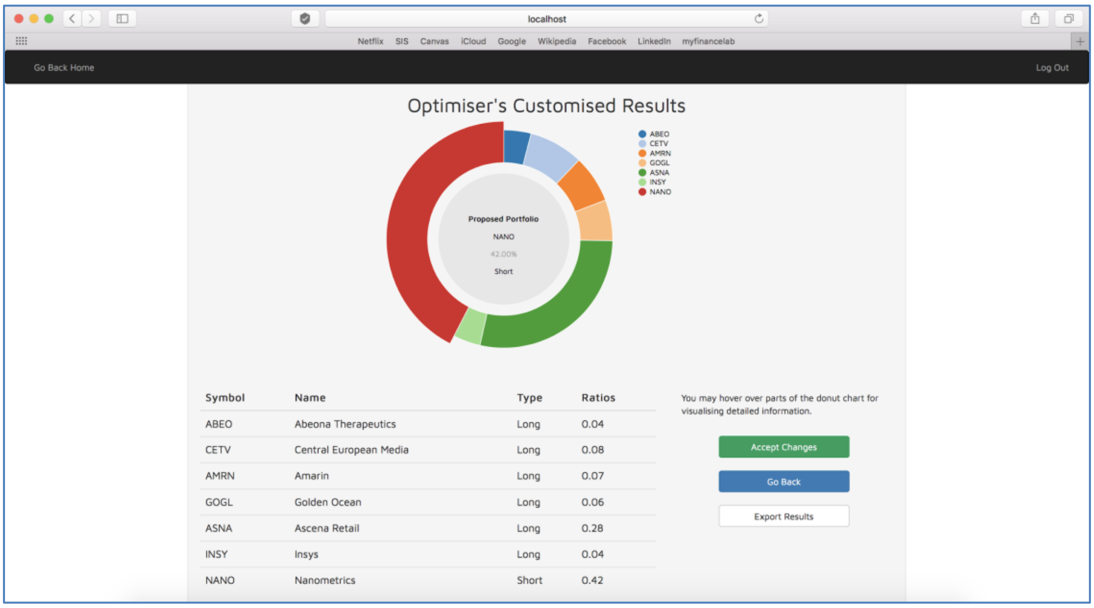

Figure 21: Screenshots of Portfolio Manager's results

After receiving the user parameters, the application sends them via an AJAX call to a Flask application on a different server. Given that JavaScript is a client-side framework and Python is a server-side framework, Flask enabled our team to create another application using Python and include the asset allocation script in it. This AJAX call runs the asset allocation script (described in Section 2.2.2) for the user's specific parameters. After the script is executed the results are returned to our main JavaScript application, processed and rendered as a chart and table on the web page and saved on Firebase if the user accepts the changes to their portfolio.

Information about which stocks the user invested in previous months as well as their ratios is stored individually for each user on Firebase in the format shown in Figure 6. This is accessed each time the portfolio manager is loaded and accordingly the overall percent change in the relevant stocks' prices is used to compute the user's portfolio growth per month.

2.2.3.5 Portfolio Growth Analyser

The **Portfolio Growth Analyser** refers to the user's past monthly portfolio compositions to render two graphs:

- A multi-line chart comparing the portfolio's actual monthly growth with benchmarks' monthly growth. The Russell 2000 index (^RUT), Vanguard Small Cap Index Investors Fund (NAESX) and Fidelity Small Cap Growth Fund (FCPGX) are used as benchmarks. ^RUT helps compare the movement of the user's portfolio with the top 2000 stocks in the small-cap stock market. Funds like NAESX and FCPGX are popular alternatives that the user may invest his money in, instead of our portfolio recommended stocks. When the user's mouse hovers over each data point, the dynamic coloured labels give the user a quick glance at which element has the highest growth (in green) for the month, the lowest growth (in red) and the elements in the middle (in orange and yellow). A brief summary of the portfolio for that given month is also displayed (Figure 22).

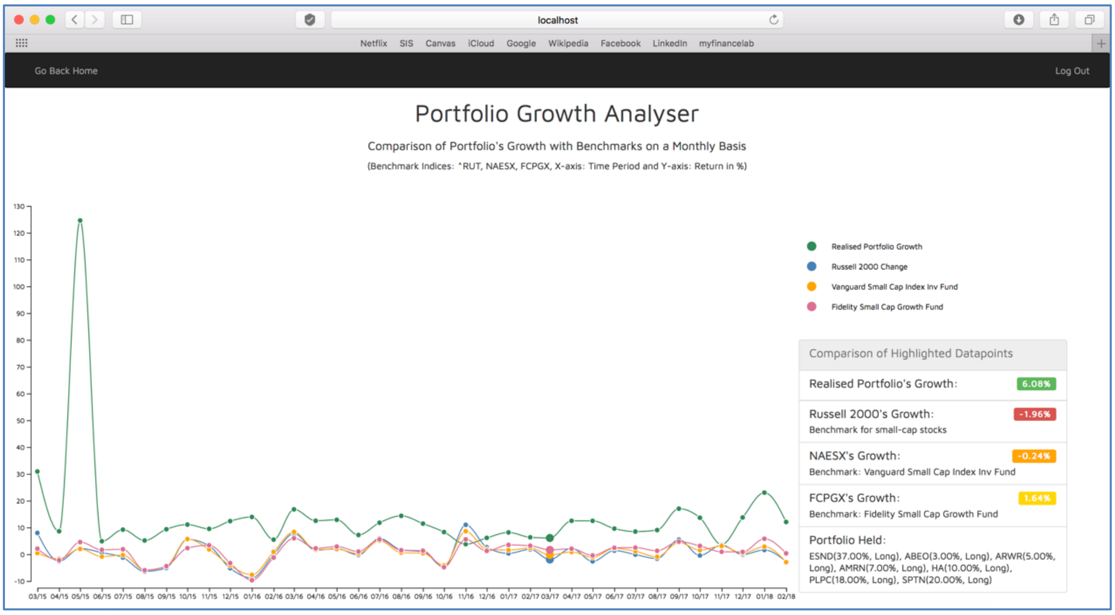

Figure 22: Comparison of Actual Growth with Benchmarks

- A multi-line chart comparing the portfolio's actual monthly growth with the machine learning model-predicted monthly growth (Figure 23).

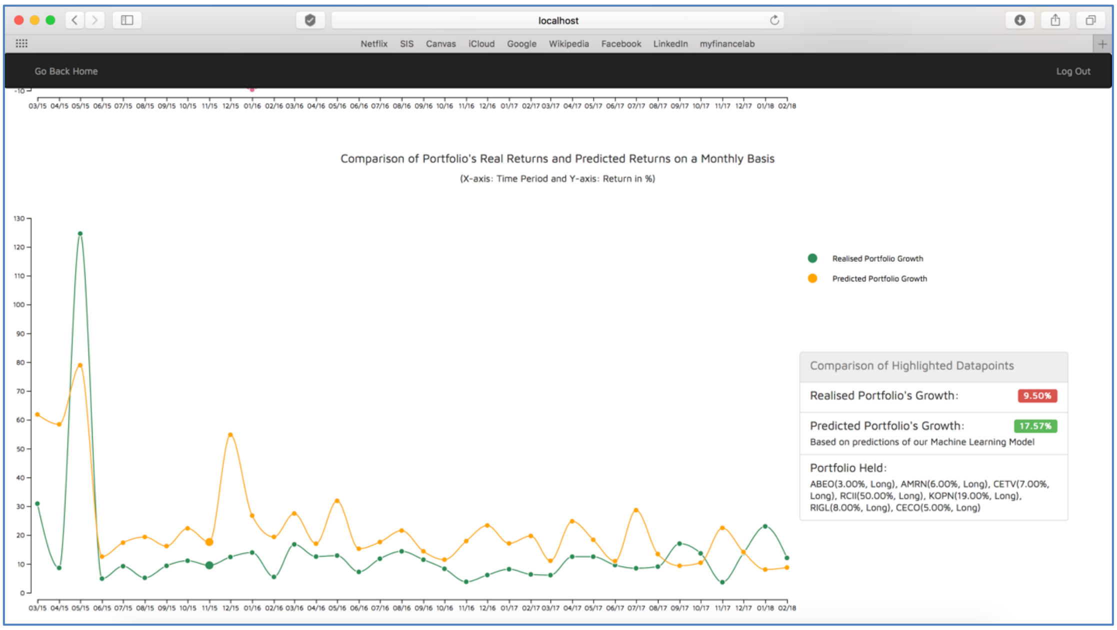

Figure 23: Comparison of Actual Growth with Predicted Growth

Our team hopes that by juxtaposing the portfolio with the benchmarks, the user will be convinced of our system's credibility and be assured that the portfolio allocator is an overall better choice as compared to the alternatives like the above-mentioned exchange traded funds (ETFs). The portfolio manager does not take into account real life constraints like transaction fees in the calculations and hence the portfolio's growth cannot be linearly correlated with actual earnings for the user.

2.3 Testing

2.3.1 Price Prediction Model Testing

The programs written to pre-process and manipulate the data have been debugged and tested to make sure they work as intended. As mentioned in sections 2.2.1.3 and 2.2.1.4.1, measures of accuracy like root mean square error (RMSE) and R^2 loss have been used to train and assess the performance of our machine learning models. Since there are multiple stocks and predictions are made over a large interval of time, it is difficult to quantitatively evaluate the price prediction using a single number. To solve this issue, we compared predicted stock prices with actual stock prices and highlighted values which deviate from the actual values by more than 10%.

Assessing the prediction model's performance in the context of the system as a whole is important as the portfolio stock recommendations, not the individual stocks' prices, determines the success of this project. The Portfolio Growth Analyser page in the web application has a multi-line graph dedicated to compare the predicted and real percent change every month.

2.3.2 Asset Allocation Model Testing

Testing for asset allocation was done according to the established practices of software testing including white box testing, black box testing and unit testing. No integration, regression or fuzzy testing was performed since the script is intended to function on a standalone basis taking inputs with pre-defined formats. If inputs are not according to the format defined, then there will be a syntax error when the data is processed by the script. This is a result of the design approach that our team took where different parts of the system interact with each other over well-defined interfaces and consequently, changes in implementation of one part doesn't necessitate testing all other parts of the application again.

White box testing was done using manual testing as well as static code analysis tools like Pylint which check code for different syntax and coding errors. Black box testing was done by monitoring the memory, CPU usage, disk I/O and context switching statistics on a Linux terminal to make sure that there are no memory leaks, especially in the convex optimization part of the system.

Unit tests were performed on sub-parts of the script to make sure they execute as intended. Specifically, the proper import of data was tested and gaps, if any, were filled by zero. The convex optimization function’s input format and output format (made compatible with the rest of the script) was validated. Moreover, some checks were placed on the input parameters after it was found that dynamically adjusted minimum return can sometimes violate the optimization parameters passed to the convex optimization library. It was corrected by placing an upper bound on the minimum expected return based on the available stock data.

One of the errors found was that the total weights may sum up to less than 1.0 when weights are rounded down to two significant figures. The error was corrected by rounding up the weight of the most weighted stock in the portfolio.

2.3.3 Web Application Testing

The third component - Portfolio Growth Analyser was built for the user’s validation as well as for the team’s assessment of the project. Our team created multiple user accounts, each having different volatility thresholds and simulated the user recommendation process for 36 months, starting from March 2015 to February 2018. The machine learning model doesn’t use data from this time period in the training or testing stage. This process gave our team feedback on how well our system would have performed over the past 3 years if different users had used our application during this time period.

The Stocks Analyser ensures that the stock symbol queried is valid, and data on the stock exists in the repository. Another check put in place is the user parameters entered in the Portfolio Manager that makes sure that incorrect values like negative volatility, extremely high volatility, or negative number of stocks are not send to the backend of the application. For each of these checks, if the user does provide a wrong input the web application issues a warning label or alert and asks the user to try again. The application has also been made to allow users to access the features of the web application only if the user is logged in. If the user visits the web URL without logging in, the page will be redirected to an error page prompting the user to log in first.

Our team gathered feedback on the web application from 10 volunteer testers, in particular about the user experience and design aspects by employing usability testing and using a Google form. One of the suggestions that stood out was offering the user the option to save the results of the portfolio allocation, so that the user can refer to the portfolio's details at a later point of time conveniently. Consequently, the mini-feature has been added to the page as a button that henceforth allows the user to download the portfolio details as a CSV file to their local file system.

2.4 Evaluation

2.4.1 Price Prediction Model Evaluation

Using the method described in 2.3.1, we found that the Multiple Linear Regression model gave the most consistent results in terms of prediction and hence it is used as the final prediction model in our web application.

2.4.2 Asset Allocation Model Evaluation

As stated in the objectives, the goal of the project is to suggest the user an optimal portfolio based on the user's needs. To evaluate how well we have achieved the stated goals over the time period of three years between March 2015 to February 2018, two measures are used: optimality and prediction accuracy.

$$Optimality = \frac{(actual\ portfolio\ growth)}{(optimal\ portfolio\ growth)} \times 100$$

Equation 6: Formula for Calculation of Optimality

Optimality is the actual portfolio value growth as a percentage of the optimal portfolio growth value within the risk threshold specified by the user. Optimal portfolio growth value is the portfolio allocation using real-data with the benefit of hindsight. For instance, if we were allocating a portfolio for March 2018, the optimal portfolio allocation on March 31, 2018 using the actual closing prices of stock in March instead of the predicted values.

$$Prediction\ Accuracy = \frac{(actual\ portfolio\ growth)}{(predicted\ portfolio\ growth)} \times 100$$

Equation 7: Formula for Calculation of Prediction Accuracy

Prediction accuracy is the actual portfolio percentage growth as a percentage of predicted portfolio value growth.

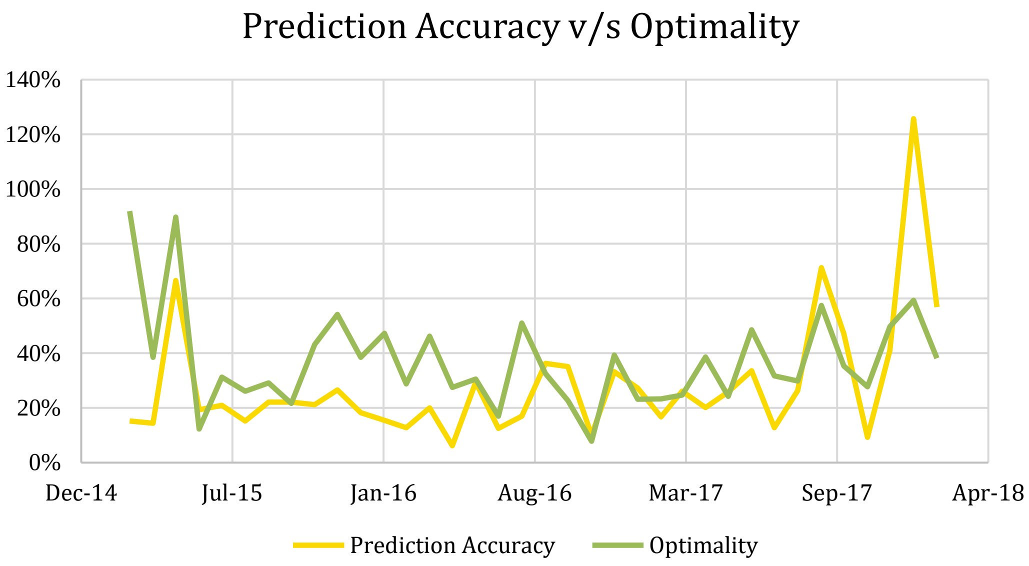

Figure 24: Prediction Accuracy vs. Optimality graph

For the 36 months of data, prediction accuracy was on average 28.6% while the optimality of the suggested portfolio was 37.2%. Trends observed in the data indicate that with higher prediction accuracy, the suggested portfolio could potentially be allocated with higher optimality.

2.4.3 Web Application Evaluation

During the usability testing phase, our team asked the testers to rate our application on the following components on a scale of 1 to 5, with 1 being the most unfavourable and 5 being the most favourable. The feedback we received was as follows:

Usability Testing	*Average Rating*
Usability of Login page	4.2 / 5.0
Usability of Services page	4.7 / 5.0
Usability of Stocks Explorer page	4.1 / 5.0
Usability of Portfolio Manager page	4.4 / 5.0
Usability of Portfolio Growth Analyser page	4.4 / 5.0

3. Discussion

The challenges that we have faced in the project are:

- *Accurate prediction of stock prices over a period of time.*
  There are limitations to how far in the future we can predict prices. Our initial target was to predict prices for 3 months at a time. However, since this did not give accurate results, our final model predicts prices for a 1 month period.
- *Adaptation of portfolio allocation theories for price prediction models generated using machine learning techniques.*
  Not all financial measures used by theories and models like Capital Asset Pricing model can be predicted accurately using machine learning techniques like LSTM.
- *Data collection and pre-processing.*
  Getting standardised stock price data from third party sources needs a lot of pre-processing to be able to be effectively utilised by machine learning models and for the purpose of portfolio allocation.
- *Integration of the Flask application into the web application.*
  Given that the Flask component and JavaScript component run on different servers, it took us effort to tackle the Cross-Origin Domain Access Policies implemented recently for stricter web safety standards.

For the price prediction component, our team initially expected that using an advanced technique such as a recurrent neural network would give us better results, which was not the case. Our team simulated this approach for a few months and suspected that the model was overfitting on the training data and unable to generalise. Given the massive amount of data we were trying to process and model, we fell short of resources in terms of time and computation power to accordingly fix the overfitting problem. Comparing the results of our limited experiments (of a few months) with the multiple linear regression model, our team always got better results and hence we chose to move forward with the regression model.

Most commercial investment managers like Vanguard and Fidelity give an estimate of the fees they charge for carrying out the trade in the market, for instance the transaction fees, brokerage fees, and real-time market costs like bid-ask spread. Our project aims to find the optimal subset of stocks, give an estimate of projected growth and acts merely as a suggestion tool for its users. Given that these costs vary from time to time and depend on how the user conducts the transactions, our project does not account for the transaction fees and this can be considered as a limitation of our project.

The Portfolio Growth Analyser serves to encourage the user to continue to use the application as they would be positively reinforced to visit the website every month to check on their portfolio's growth. Another mini-feature that can be added to the project is sending a monthly reminder via email to the user to check their portfolio and its results.

4. Conclusion

In our project, we developed a web application that would help individual investors make informed decisions about investing in small-cap stocks in the right proportions and at the right time. Our team achieved this by using a multiple linear regression model for the stock price forecasting part and a convex optimization algorithm based on Markowitz's theory of portfolio allocation. A user-friendly web application was coded using AngularJS with the help of D3.js for visualisation. Users have the opportunity to optimise their recommended portfolio every month and track their portfolio growth using the Portfolio Growth Analyser. We have performed better than the selected benchmarks (Russell 2000, Fidelity Small-cap Growth Fund, Vanguard Small-cap Index Investors Fund) in 35 out of 36 simulated months.

A possible future area of development is the exploration of more machine learning algorithms and comparison of results to our existing method. For example, more computation power would allow us to experiment with Recurrent Neural Networks and see if better results can be obtained. Another area of exploration is the incorporation of non-financial data such as Twitter data, weather data and Google Trends to make more accurate predictions for better portfolio optimisation.

5. References

[1] Robo Advising - Catching Up and Getting Ahead. [Online]. Available: https://home.kpmg.com/content/dam/kpmg/pdf/2016/07/Robo-Advising-Catching-Up-And-Getting-Ahead.pdf. [Accessed Sep 19, 2017].

[2] I. Staff. (2017). "An Introduction To Small Cap Stocks". *Investopedia*. [Online]. Available: http://www.investopedia.com/articles/01/080101.asp. [Accessed Sep 19, 2017].

[3] I. Staff. (2017). "Small Cap". *Investopedia.* [Online]. Available: http://www.investopedia.com/terms/s/small-cap.asp. [Accessed Sep 19, 2017].

[4] B. Carlson. (2017). "The Difference Between Institutional & Individual Investors", *A Wealth of Common Sense*. [Online]. Available: http://awealthofcommonsense.com/2016/02/the-difference-between-institutional-individual-investors/. [Accessed Sep 19, 2017].

[5] I. Staff. (2017). "Institutional Investor". *Investopedia.* [Online]. Available: http://www.investopedia.com/terms/i/institutionalinvestor.asp. [Accessed Sep 19, 2017].

[6] "Retail Investor Definition & Example | InvestingAnswers". [Online]. Available: http://www.investinganswers.com/financial-dictionary/investing/retail-investor-911. [Accessed Sep 19, 2017].

[7] G. Brackenridge. (2017)."Artificial intelligence is transforming investment strategies". *CNBC*. [Online]. Available: https://www.cnbc.com/2017/06/06/machine-learning-transforms-investment-strategies-for-asset-managers.html. [Accessed Sep 19, 2017].

[8] H. Markowitz (1952). "Portfolio Selection". *The J. of Finance*, vol. 7, no. 1, pp. 77-9. [Online]. Available: http://www.jstor.org/stable/2975974. [Accessed Sep 19, 2017].

[9] E. Fernholz, *Stochastic portfolio theory*. New York: Springer, 2011.

[10] I Know First – Daily Market Forecast. Stock Forecast Based On a Predictive Algorithm. [Online]. Available at: https://iknowfirst.com/stock-forecast-algorithm. [Accessed Feb 14, 2018].

[11] L. Wang and Q. Wang, "Stock Market Prediction Using Artificial Neural Networks based on HLP", in 3rd Intl. Conf. on Intelligent Human-Machine Systems and Cybernetics (IHMSC 2011), 2011, pp. 116–119

[12] Y. Kom Samo and A. Vervuurt, "Stochastic Portfolio Theory: A Machine Learning Perspective", *Arxiv*, 2016. [Accessed Sep 19, 2017].

[13] "API Documentation | Alpha Vantage". [Online]. Available at: https://www.alphavantage.co/documentation/ [Accessed Feb 11, 2018].

[14] "Stock Screener Tool - Zacks Investment Research". [Online]. Available at: https://www.zacks.com/screening/stock-screener. [Accessed Feb 11, 2018].

[15] Anon, (2015), "long short term memory", [Blog] http://blog.otoro.net/2015/05/14/long-short-term-memory/ [Accessed Feb 11, 2018].

[16] "sklearn.linear_model.LinearRegression —scikit-learn 0.19.1 documentation". [Online]. Available at: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html [Accessed Feb 15, 2018].

[17] T. Balch, "Machine Learning for Trading". Udacity, Georgia Institute of Technology, 2017.

[18] T. Starke, D. Edwards, T. Wiecki. (2015). "The Efficient Frontier: Markowitz portfolio optimization in Python". [Blog]. Quantopian. Available at: https://blog.quantopian.com/markowitz-portfolio-optimization-2/ [Accessed 18 Apr. 2018].

6. Appendix A: Glossary

[6.1] **CAGR**:
Refers to the growth compounded annually. Can be used for industries, economy or even a single stock.

[6.2] **NASDAQ**:
National Association of Securities Dealers Automated Quotations is the second largest stock exchange in the world, located in New York City, United States.

[6.3] **NYSE**:
The largest stock exchange in the world, located in New York City, United States.

[6.4] **Asset Classes**:
A group of securities that exhibits similar characteristics, behaves similarly in the marketplace and is subject to the same laws and regulations.

[6.5] **Cross-Asset Portfolio**:
Portfolio which contains a combination of multiple assets classes.

[6.6] **Market Capitalization**:
Total Market value of a company's outstanding shares. Obtained by multiplying the number of outstanding stocks with the price of each stock.

[6.7] **Efficient Market Hypothesis**:
Stocks (other assets) will be fairly priced by the market taking into account all the relevant information about the company as well as the macroeconomic conditions.

[6.8] **Investment Strategy**:
An investor's plan of attack to guide their investment decisions based on individual goals, risk tolerance and future needs for capital.

[6.9] **Risk Threshold**:
The degree of variability in investment returns that an investor is willing to withstand.

[6.10] **Expected Returns**:
Expected return is the amount of profit or loss an investor anticipates on an investment

[6.11] **HLP**:
High-Low Point refers to the high and low price of an asset over a given time period

[6.12] **In-sample**:
Observations and predictions based on the data that was used to model training.

[6.13] **CAPM**:
The model used to determine a theoretically appropriate required rate of return of an asset, to make decisions about adding assets to a well-diversified portfolio.

[6.14] **Exchange Traded Funds (ETFs)**:
Funds comprising of assets like stocks, bonds, commodities that are made to track specific indices.

7. Appendix B: Meeting Minutes

7.1 Minutes of the 1st Project Meeting

Date: July 15, 2017
Time: 5:00 PM
Place: Skype
Present: Ashish, Anwesha, Anish
Absent: None
Recorder: Ashish

1. Approval of minutes

This was the first formal group meeting hence there were no minutes to approve.

2. Report on progress

2.1 All team members read through different articles and sources to read up on potential topics

2.2 Each team member prepared ideas to pitch during the meeting

2.3 All team members read up on current trends in Machine Learning and AI

2.4 Getting familiar with the objectives and goals of the Final Year Project

3. Discussion items

3.1. Various ideas pitched by each team member were discussed and evaluated

- Healthcare (Ashish)
- Source Code/Unit Test Generation (Anwesha)
- Making Hiring Decisions based on Data (Anish)
- Using Trade Volume to predict currency movements (Ashish)
- Fake news Detection (Anwesha)
- Language Detection on Audio files using machine learning (Anish)

3.2. We discussed the timeline of the project and the broad goals and overview of the values and ideas that we would like to work on during the project, for e.g. we would like to solve some real-world problems rather than some theoretical problems.

4. Objectives for the next meeting

4.1 To further do more research about each of the topics discussed

4.2 Explore more potential topics in the areas that we are interested in

- Finance, Healthcare, Social Media, Software/Technology

4.3 To agree on a common timeline and schedule for the summer

4.4 Narrow down the list of topics (hopefully to one)

4.5 Discuss the Project with Dr. David Rossiter

5. Meeting adjournment and next meeting

The meeting was adjourned at 6:15 PM.

The next meeting will be on August 3, 2017 via Skype.

7.2 Minutes of the 2nd Project Meeting

Date: August 3, 2017
Time: 6:15PM
Place: Coffee Shop, HKUST
Present: Prof. Rossiter, Anwesha, Anish (first half via voice call)
Absent: Ashish, Anish (second half)
Recorder: Anwesha

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Ashish summarised our findings with respect to our discoveries for each potential idea and sent an email to Prof. Rossiter with the same.

2.2 Anwesha and Anish prepared some points to raise for the meeting with Prof. Rossiter.

2.3 All team members read past projects' report to familiarise themselves with distinguished FYPs.

3. Discussion items

3.1. Prof. Rossiter and Anwesha went through each idea discussed in the email and discussed potential problems/benefits of pursuing each idea as an FYP:

3.1.1 Cross-industry analysis: Prof. Rossiter indicated that the idea has much potential if the correct items are used together, otherwise it may be too easy for an FYP.

3.1.2 Source Code/Unit Test Generation: Source code might be too difficult and may not show results as a project within 9 months for the FYP, done by big tech research divisions. Unit tests might make the project very restricted and narrow it down further in terms of use case than we should hope to.

3.1.3 Fake News Detection: Trending and hot topic, would have been perfect as President Cup material, but data would be extremely hard to find.

3.1.4 Language detection: detecting languages from audio would be complexity-wise good as an FYP but knowledge on multimedia processing would have been useful, otherwise would take time to learn.

3.1.5 Hiring/Attrition: real data would be hard to find to use for the project.

3.2 Prof. Rossiter gave a few tips for expanding on our current ideas, including using social media and news as data sources, advising to start small then add layers to our project so that obstacles may not render our project failure in the case of time shortage.

4. Objectives for the next meeting

4.1 To discuss and reflect on points raised by Prof. Rossiter

4.2 To narrow down our project ideas to one topic.

4.3 To research deeper about the topic we choose.

5. Meeting adjournment and next meeting

The meeting was adjourned at 7:15 PM.

The next meeting will be on August 13, 2017 via Skype.

7.3 Minutes of the 3rd Project Meeting

Date: August 13, 2017
Time: 6:30PM
Place: Skype
Present: Anwesha, Anish, Ashish
Absent: None
Recorder: Anish

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Anwesha briefly mentioned via group chat which topics to read into before our meeting, and our team looked into cross-industry applications of machine learning and language audio detection.

3. Discussion items

3.1 All team members gave their input on the pros and cons of each topic. After this discussion, we narrowed down to 2 topics of interest: Language Audio Detection and Applications of machine learning to financial data.

3.2 Based on these discussions, our team concluded that language audio detection, although interesting would be challenging since our group doesn't have relevant experience in the field of multimedia computing.

3.3 Our team planned to research further on applications of machine learning to financial data such as stock prices, derivatives and futures and look for research articles on where non-intuitive factors may have been used to predict any of the former.

4. Objectives for the next meeting

4.1 To validate our findings and opinions and confirm our project topic with Prof. Rossiter.

5. Meeting adjournment and next meeting

The meeting was adjourned at 7:15 PM.

The next meeting will be on September 1, 2017 with Prof. Rossiter at HKUST.

7.4 Minutes of the 4th Project Meeting

Date: September 1, 2017
Time: 3:00PM
Place: Room 3554, Prof. Rossiter's office
Present: Prof. Rossiter, Anwesha, Anish, Ashish
Absent: None
Recorder: Anwesha

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Ashish summarised what our team did over the summer and how our team came to the topic chosen as it was the first time our team met in person since Fall 2017.

3. Discussion items

3.1 All team members pitched in points for how our team came to the idea of using non-intuitive factors to predict a financial indicator, using non-intuitive factors like weather patterns or news that our team could use to build on our main project to predict how the market will perform.

3.2 Prof. Rossiter asked about our FYP's title and encouraged us to formulate a brief phrase to describe our project then work from there to collaborate on what our team wanted to include in the project.

3.3 Prof. Rossiter mentioned not to make the title too specific or narrow to the factors used for machine learning predictions, rather use words that encompassed an umbrella of what possibly could be worked with and achieved in our project.

3.4 Anwesha introduced Google Trends to Prof. Rossiter as a possible source for understanding what news might affect investor sentiment among other things. The team and Prof. Rossiter noticed that Google Trends could not be used for HK data.

3.5 Prof. Rossiter suggested our team have a look at the Bloomberg terminal and see what information our team can use from there if needed and if allowed.

3.6 Prof. Rossiter introduced and demoed his mini project to predict best averages for a particular stock and explained his approach.

3.7 Anish and Ashish enquired about how the program gets around to giving its results and the team got a better understanding of what might be needed for the initial layer of the FYP.

3.8 Prof. Rossiter advised to devise the stages of our team's plan for the next coming months and work towards completing these.

4. Objectives for the next meeting

4.1 To come up with the FYP title and move forward with our efforts from there.

4.2 To briefly discuss what the stages would be in our action plan.

4.3 To prioritise factors (financial, non-financial) to be used for our initial model.

5. Meeting adjournment and next meeting

The meeting was adjourned at 3:45 PM.

The next meeting will be on September 5, 2017 in a group study room in the library to be decided later.

7.5 Minutes of the 5th Project Meeting

Date: September 5, 2017
Time: 1:30PM
Place: LC-05 Learning Commons
Present: Ashish, Anwesha, Anish
Absent: None
Recorder: Ashish

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Confirm the topic and inform Professor of the same.

2.2 Research more on the topic and understand the current status quo

2.3 Explore different potential methods of applying Machine learning to the project

2.4 Getting familiar with the objectives and goals of the FYP Proposal Report

3. Discussion items

3.1. Various techniques and ideas for the implementation of the project were discussed. Team broadly discussed different tools and technologies that can be used over the course of the project. Challenges in implementing certain technologies and ideas were also discussed.

3.2. We discussed the timeline of the project and the broad goals and overview of the values and ideas that we would like to work on during the project, eg. we would like to solve some real-world problems rather than some theoretical problems.

4. Objectives for the next meeting

4.1 To further do more research about each of the topics discussed

4.2 Explore more potential topics in the areas that we are interested in

- Finance, Healthcare, Social Media, Software/Technology

4.3 To agree on a common timeline and schedule for the summer

4.4 Narrow down the list of topics (hopefully to one)

4.5 Discuss the Project with Dr. David Rossiter

5. Meeting adjournment and next meeting

The meeting was adjourned at 3:00 PM.

The next meeting will be on September 10th, 2017 (TBC)

7.6 Minutes of the 6th Project Meeting

Date: September 10, 2017
Time: 3:30PM
Place: HKUST Jockey Club Hall
Present: Anwesha, Anish, Ashish
Absent: None
Recorder: Anish

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Anwesha briefly mentioned via group chat which topics to read into before our meeting, and our team looked into cross-industry applications of machine learning and language audio detection.

3. Discussion items

3.1 After finalising our FYP topic, our group discussed possible data sources, data representation and project methodology. Our group decided that we would be using financial data for small cap stocks to calculate potential profit.

3.2 Our group worked out a potential project plan including timelines using GANTT chart, division of work using Leader/Assistant method and planned to begin the FYP proposal.

4. Objectives for the next meeting

4.1 To validate the project timeline, discuss a partial proposal draft with Prof. Rossiter.

5. Meeting adjournment and next meeting

The meeting was adjourned at 6:00 PM.

The next meeting will be on September 12, 2017 with Prof. Rossiter at HKUST.

7.7 Minutes of the 7th Project Meeting

Date: September 12, 2017
Time: 2:30PM
Place: Room 3554, Prof. Rossiter's office
Present: Prof. Rossiter, Anwesha, Anish, Ashish
Absent: None
Recorder: Anwesha

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 All team members began work on drafting the proposal for the FYP.

3. Discussion items

3.1 Ashish brought Prof. Rossiter up to speed with the collaboration tools that our team has been using so far.

3.2 Prof. Rossiter advised us on catering the grammar our team will use on the FYP proposal to the current experimental stage our team is at.

3.3 Anwesha asked and collected feedback on the draft of the objective, which also provided a clearer idea of our team's goals.

3.4 Anish and Ashish helped explain the proposed stages of our project to Prof. Rossiter.

4. Objectives for the next meeting

4.1 To keep up with our assigned responsibilities for the FYP proposal.

4.2 To review the proposal draft and send a copy to Prof. Rossiter for his feedback.

5. Meeting adjournment and next meeting

The meeting was adjourned at 3:00 PM.

The next meeting will be on September 14, 2017 in a venue to be decided later.

7.8 Minutes of the 8th Project Meeting

Date: September 19, 2017
Time: 2:30PM
Place: Room 3554, Prof. Rossiter's office
Present: Prof. Rossiter, Anwesha, Anish, Ashish
Absent: None
Recorder: Ashish

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 All team members worked on the proposal for the FYP.

2.2 Team members explored different methods of doing the project, researching potential techniques for prediction of stock prices.

3. Discussion items

3.1 Team prepared the first draft of the FYP Proposal

3.2 Prof. Rossiter gave feedback on the draft as well suggested few areas of improvements for the draft.

3.3 Team discussed and got feedback on Literature Review from Prof. Rossiter

3.4 Team also discussed the possible tasks as a part of the FYP and divided the tasks according to each member's specialization and interests.

3.5 The team discussed steps to take besides completing the proposal, namely the tasks to start working on. These include understanding the market standard, current research and investigating possible directions for our project on a more granular level.

3.6 Title of the project was discussed with Professor Rossiter was modified accordingly to better reflect the focus of the project.

4. Objectives for the next meeting

4.1 To complete the FYP proposal and inculcate changes as suggested by Prof. Rossiter

4.2 Submit the FYP proposal after formatting the report in accordance with CSE Department's standards.

4.3 Discuss and plan for future actions according to the timeline mentioned in the proposal.

4.3.1 Experiment with online tools available to check and validate our strategies for stock prediction.

5. Meeting adjournment and next meeting

The meeting was adjourned at 3:20 PM.

The next meeting will be on September 26, 2017.

Venue: TBC

7.9 Minutes of the 9th Project Meeting

Date: September 26, 2017
Time: 6:30PM
Place: Skype
Present: Anwesha, Anish, Ashish
Absent: None
Recorder: Anwesha

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 All team members reflected on tips given by the CT for the other reports.

2.2 All team members reflected on suggestions given by Prof. Rossiter regarding services and platforms to run algorithms such as the Python Quants.

3. Discussion items

3.1 The team discussed sources of data like Yahoo! Finance, and services to host the data like ElasticSearch and most of the team members decided to read more about what can be done with ElasticSearch, including visualisation services like Kibana along with it.

3.2 Anish, having been experimenting with LSTM in general, decided to start working on writing a program to deal with stock data (time series).

3.3 Anwesha decided to look at data sources and feeds for the same.

3.4 Ashish plans to ask professors about research methods and look at what services like Python Quants can offer the team.

4. Objectives for the next meeting

4.1 To complete the tasks as mentioned in the discussion items.

5. Meeting adjournment and next meeting

The meeting was adjourned at 7:00 PM.

The next meeting will be held on 3 October 2017, venue TBC.

7.10 Minutes of the 10th Project Meeting

Date: October 3, 2017
Time: 1:30PM
Place: Skype
Present: Anwesha, Anish, Ashish
Absent: None
Recorder: Ashish

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Writing script for data collection and preprocessing (Anwesha)

2.2 Reading relevant research material on the topic and financial data (Ashish)

2.3 Researching on different possible platforms to test algorithms (Anish)

3. Discussion items

3.1 The team discussed issues with data handling and collection as well as the possible alternatives to downloading data locally.

3.2 Team decided to explore cloud computing services like AWS to handle data and other tasks

3.3 Anish worked on LSTM in relation with small-cap stocks.

3.4 Anwesha decided to look at data sources and feeds for the same and improve on her script to download data from Alpha Vantage.

3.5 Ashish plans to research more on the financial aspect and explore AWS

4. Objectives for the next meeting

4.1 Continue processing, downloading more small-cap stocks data. (Anwesha)

4.2 Look at the AWS computing stack and see how the resources can be used for the project. (Ashish)

4.3 Anish will use the LSTM technique on the data collected as well as on online platforms like Python Quant.

4.4 Discuss the monthly report and project details with Professor Rossiter.

4.5 Discuss the team's participation in President's Cup using the FYP Project.

5. Meeting adjournment and next meeting

The meeting was adjourned at 2:30 PM.

The next meeting will be held on 17 October 2017, venue Room 4221.

7.11 Minutes of the 11th Project Meeting

Date: October 17, 2017
Time: 1:15PM
Place: Lab 4221, HKUST
Present: Anwesha, Anish, Ashish
Absent: None
Recorder: Anish

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Historical data of around 350 stocks has been collected and stored using AWS

2.2 Read up on more details on financial data as well as machine learning algorithms

2.3 Machine Learning algorithm (LSTM) has been used to learn and predict stock prices.

2.4 Visualization using plots and prediction data is saved for each stock

3. Discussion items

3.1 The team discussed issues with data collection for remaining stocks

3.2 The team discussed possible ways to quantify the prediction results and compare performance across stocks. Eg: Classifying movement as +ve or -ve, using root mean square error and percentage change ratio

3.3 The team discussed ways to make the prediction engine optimal for each stock by finding the most optimal value for sequence length for LSTM

4. Objectives for the next meeting

4.1 Discuss and implement methods to compare prediction performance across stocks

5. Meeting adjournment and next meeting

The meeting was adjourned at 2:30 PM.

The next meeting will be held on 24 October 2017, venue Room 4221.

7.12 Minutes of the 12th Project Meeting

Date: October 31, 2017
Time: 1:00PM
Place: Learning Commons, Library, HKUST
Present: Anwesha, Anish, Ashish
Absent: None
Recorder: Anwesha

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Moving averages of a few stocks have been calculated

2.2 Moving averages' proximity to train and test data were compared to LSTM's proximity using the plots themselves

2.3 Better ways to quantitatively compare the two were experimented with and tried

3. Discussion items

3.1 The team went over some concepts of training and testing data for clarity and discussed possibly reassigning a small portion of the data for a second round of testing (right now train:test = 7:3 but proposed 6:3:1)

3.2 The team went over the algorithmic procedure of optimising the sequence length

3.3 The team discussed the possibility of using ensemble methods after satisfactory experimentation with LSTM

4. Objectives for the next meeting

4.1 Evaluate with moving averages in a more formal way

4.2 Optimise sequence length for a representative subset of stocks

4.3 Change parameters within LSTM to aim for better results.

5. Meeting adjournment and next meeting

The meeting was adjourned at 1:30 PM.

The next meeting will be held on 7 November 2017, venue TBC.

7.13 Minutes of the 13th Project Meeting

Date: November 8, 2017
Time: 11:00 A.M.
Place: Room 3554, HKUST
Present: Anwesha, Anish, Ashish
Absent: None
Recorder: Ashish

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Optimal Sequence length for 14 stocks was implemented and run on the server

2.2 Moving average has been included in the analysis data. Ways to improve it were discussed

2.3 Research on different methods of comparing and analysing time series were explored

3. Discussion items

3.1 Ways to improve the performance and efficiency to calculate the optimal sequence length for LSTM were discussed

3.2 Professor Rossiter suggested that we implement a simple trading algorithm to more effectively compare our strategies

3.3 It was decided to implement the the optimal sequence length for moving average as well for fairer comparison between moving average and LSTM

4. Objectives for the next meeting

4.1 Discuss ways to implement trading strategy to evaluate the algorithm

4.2 Run simulation for large number of stocks and see the results

4.3 Change parameters within LSTM to aim for better results.

5. Meeting adjournment and next meeting

The meeting was adjourned at 12:00 PM.

The next meeting will be held on 22 November 2017, venue Room 3554.

7.14 Minutes of the 14th Project Meeting

Date: November 22, 2017
Time: 1:00 P.M.
Place: Room 3554, HKUST
Present: Anwesha, Anish, Ashish
Absent: None
Recorder: Ashish

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Exploring use of various machine learning and heuristic approaches for portfolio allocation

2.2 Discussed ways to improve the accuracy of predicted data

2.3 Basic layout and features for a web application were discussed

3. Discussion items

3.1 Contents of the presentation for the next meeting were discussed

3.2 Discuss the latest LSTM results after tweaking of parameters and optimizing loss

3.3 Discuss the contents of the Udacity course for trading using machine learning

4. Objectives for the next meeting

4.1 Presentation for Professor Rossiter. Prepare and discuss the presentation contents

4.2 Explore the Udacity course more and see how the learnings can be implemented in the project

4.3 Try to improve the results of time series prediction. Explore more ways to improve accuracy

5. Meeting adjournment and next meeting

The meeting was adjourned at 1:30 PM.

The next meeting will be held on 19 December 2017, venue Room 3520.

7.15 Minutes of the 15th Project Meeting

Date: December 19, 2017
Time: 1:00 P.M.
Place: Room 3520, HKUST
Present: Anwesha, Anish, Ashish, Prof. Rossiter, Team RO5
Absent: None
Recorder: Anish

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 The standard deviations of all stocks were calculated to quantify risk associated with investing in these

2.2 A first attempt at the portfolio allocation had been made, using expected returns from the predicted price data and risk obtained as mentioned above

3. Discussion items

3.1 Progress so far, including the team's motivation, introduction to the project and system architecture were presented

3.2 Discussions on the team's approach to the LSTM algorithm and comparison to moving average were held

3.3 The team's recent start with the portfolio allocation system and the concepts used for it were discussed

4. Objectives for the next meeting

4.1 Incorporation of Prof. Rossiter's suggestions for our project

4.2 Expand current approach to the asset allocation to return more than 2 stocks at a time.

4.3 Start design and implementation of web application

5. Meeting adjournment and next meeting

The meeting was adjourned at 2:00 PM.

The next meeting will be held in Spring 2018

7.16 Minutes of the 16th Project Meeting

Date: February 7, 2018
Time: 1:00 P.M.
Place: Room 3554, HKUST
Present: Anwesha, Anish, Ashish
Absent: None
Recorder: Ashish

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Exploring use of regression to predict time series data

2.2 Started to implement portfolio allocation based on data and CAPM model

2.3 Worked on the web application for user to interact with the system

3. Discussion items

3.1 Discuss the progress over winter and the comments by Professor Rossiter

3.2 Discuss the UI/UX of the web application made by Anwesha

3.3 Discuss the strategies used for portfolio allocation

4. Objectives for the next meeting

4.1 Complete and review monthly reports and progress report

4.2 Discuss progress of the web app and portfolio allocation

4.3 Discuss the first draft of Progress report with communication tutor and Professor Rossiter

5. Meeting adjournment and next meeting

The meeting was adjourned at 1:30 PM.

The next meeting will be held on 12 February 2018, venue Room 3554.

7.17 Minutes of the 17th Project Meeting

Date: February 27, 2018
Time: 1:00 P.M.
Place: Room 3554, HKUST
Present: Anwesha, Anish, Ashish
Absent: None
Recorder: Ashish

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Completed the progress report after feedback from Prof. Rossiter

2.2 Data collection for Mean variance optimization problem to allocate stocks

2.3 Worked on the web application for user to interact with the system

3. Discussion items

3.1 Discuss the results of new method for price prediction to improve accuracy

3.2 Discuss the UI/UX of the web application made by Anwesha

3.3 Discuss the strategies used for portfolio allocation

4. Objectives for the next meeting

4.1 Complete portfolio allocation and discuss results

4.2 Discuss progress of the web app

4.3 Start working on the final project deliverables like Report and Presentaion

5. Meeting adjournment and next meeting

The meeting was adjourned at 1:30 PM.

The next meeting will be held on 13 March 2018, venue Room 3554.

7.18 Minutes of the 18th Project Meeting

Date: April 9, 2018
Time: 2:00 P.M.
Place: Room 3554, HKUST
Present: Anwesha, Anish, Ashish
Absent: None
Recorder: Anish

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Completed price prediction module and decided that multiple linear regression works the best for our application.

2.2 Ran scripts for optimal portfolio allocation for period of 1 year from March 2017 to February 2018.

2.3 Completed the first 2 stages of the web application: Stocks Explorer and Portfolio Manager

3. Discussion items

3.1 Analyse results of portfolio allocation and compare with benchmark indices

3.2 Discuss the possibility of incorporating trading costs in our application

4. Objectives for the next meeting

4.1 Complete Portfolio Growth Analyser in the web application

4.2 Complete a draft of the final report and discuss presentation planning

5. Meeting adjournment and next meeting

The meeting was adjourned at 3:00 PM.

The next meeting will be held on 16 April 2018, venue Room 3554.

7.19 Minutes of the 19th Project Meeting

Date: April 16, 2018
Time: 2:00 P.M.
Place: Room 3554, HKUST
Present: Anwesha, Anish, Ashish
Absent: None
Recorder: Anish

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Completed all parts of the project: Stocks Explorer, Portfolio Manager and Portfolio Growth Analyser.

2.2 Worked on minor changes suggested by Professor Rossiter on the web application.

2.3 Completed a draft of the final report.

3. Discussion items

3.1 Discuss the draft of the final report.

3.2 Discuss results obtained from the portfolio manager for 3 years (March 2015-February 2018).

4. Objectives for the next meeting

4.1 Complete the final report and self-assessment form.

4.2 Discuss the flow of presentation among team members and start working on slides.

5. Meeting adjournment and next meeting

The meeting was adjourned at 3:00 PM.

The next meeting will be held on 21 April 2018, venue Room 2127C (Final Presentation).

8. Appendix C: Project Planning

8.1 Distribution of Work

Task	Ashish	Anwesha	Anish
Researching possible topics	○	●	○
Reading and Research on chosen topic	●	○	○
Data collection and preprocessing	○	●	○
Implementation of LSTM algorithm	○	○	●
Optimisation of LSTM algorithm	○	○	●
Comparison of LSTM results with moving average	○	○	●
2-stocks portfolio allocation implementation	●	○	○
Web application UI prototyping	○	●	○
Multiple asset portfolio allocation	●	○	○
Implementation of Regression algorithm	○	○	●
Implementation of Web application	○	●	○
Unit testing of Python scripts and web application	●	○	○
Evaluation of system performance	●	○	○
Proposal	●	○	○
Monthly reports	○	○	●
Final report	○	●	○
Poster design	●	○	○
Presentation preparation	○	○	●

● Leader ○ Assistant/Team effort

8.2 GANTT Chart

Task	Jul		Aug		Sep		Oct		Nov		Dec		Jan		Feb		Mar		Apr	
Researching possible topics	■	■	■																	
Reading and Research on chosen topic			■	■	■	■														
Data collection and preprocessing						■	■	■												
Implementation of LSTM algorithm							■	■												
Optimisation of LSTM algorithm								■	■											
Comparison of LSTM results with moving average										■										
2-stocks portfolio allocation implementation										■	■	■								
Web application UI prototyping													■							
Multiple asset portfolio allocation													■	■						
Implementation of Regression analysis algorithms														■	■					
Implementation of Web application														■	■	■	■			
Unit testing of Python scripts and web application																	■			
Evaluation of system performance																		■		
Proposal					■															
Monthly reports						■		■		■		■			■					
Progress report															■					
Final report																		■	■	
Poster design																			■	■
Presentation preparation																			■	■

9. Appendix D: Hardware & Software

Our team used the following resources and software in our project:

9.1 Resources

Purpose	Resource(s)
Software Development	3 Development PCs
Training and testing of ML algorithms	Amazon Web Services (AWS) Elastic Cloud Compute (EC2) server
Low-level prototypes, wireframes	Moqups.com

9.2 Software

Purpose	Resource(s)
Machine Learning	Python 2.7, TensorFlow, Scikit-Learn, Numpy, Pandas, Matplotlib, Keras
Data Storage	Firebase
Source Code Management	GitHub
Front-end languages and frameworks	AngularJS 1.6, jQuery, HTML5, CSS, D3.js, Bootstrap, Flask