

RO1

FYP RO1 - Personalized Algo-trading Using Deep Learning & Machine Learning models
(COMP/DSCT)

FYP Proposal

Personalized Algo-trading Using Deep Learning & Machine Learning models

by

MUN Sun Bin, KWOK Ue Nam, KONG Kin Cheung, and HO Yat Man Peter

RO1

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

2021-2022

Date of submission: 20th April 2022

Abstract

In the last couple of years, algorithmic trading has become one of the most discussed topics in the financial market. However, applying algorithmic trading strategies requires knowledge of investing and also in computer programming. The purpose of this project was to create a personalized algo-trading platform with the help of machine learning and deep learning models to allow users to invest in the financial market with minimal knowledge and effort.

Our system is divided into two parts: the users' portfolio assessment and the price prediction. First, it assesses the users' risk-bearing level and areas of interest by a preference survey, so the platform would select a list of stocks for users to invest in. After that, stock price predictions are done by multiple machine learning and deep learning models. With the list of stocks and the price prediction, the platform then can make the final investment decisions for the users. To display the users' portfolios in an intuitive manner, a one-page web dashboard application was built to visualize portfolio performance and asset distribution.

Our team experimented with a variety of machine learning and deep learning models and utilized the results for making rational trading decisions. These models and an intuitive web dashboard application offer a reliable and user-friendly investment experience for beginning investors.

Table of Contents

1. Introduction	6
1.1 Overview	6
1.1.1 Stock Market status quo	6
1.1.2 Machine Learning and Deep Learning in Trading	6
1.1.3 Algorithmic Trading	7
1.2 Objectives	9
1.2.1 Database Implementation	9
1.2.2 Machine Learning Model Training	9
1.2.3 Risk Quantification	10
1.2.4 Portfolio Presentation	10
1.3 Literature Survey	12
1.3.1 Existing Algo-Trading Companies	12
1.3.2 Stock Price Prediction Models	13
2. Methodology	15
2.1 Design	15
2.1.1 User Preference Survey	17
2.1.2 User Classification & Asset Deposit	17
2.1.3 Selecting the Combination of Algorithms/Strategies	17
2.1.4 Web Dashboard Application	18
2.2 Implementation	20
2.2.1 User Survey - Input Form	20
2.2.2 Model Training/Testing/Evaluation	20
2.2.3 Model Assignment using Volatility of the Model	20
2.2.4 Data Storage & Database Structure	21
2.2.5 Web Dashboard Application Components	23
2.3 Testing	28
2.3.1 Backtesting on the Prediction Models	28
2.3.2 Backtesting on User Portfolio	29
2.3.3 Web Dashboard Application Testing	32
2.4 Evaluation	33
2.4.1 Evaluation of Customization	33
2.4.2 Evaluation of Our Platform	34
2.4.2.1 Overall performance	34
2.4.2.2 Performance by year	34
2.4.2.3 Performance by risk level	36
2.4.3 Evaluation of Web Dashboard Application	38
3. Discussion	39
3.1 Challenges	39
3.2 Relationship with the research problem or hypothesis	40
4. Conclusion	41
4.1 Summary of Achievements	41

4.2 Ideas for further development	42
4.2.1 Zooming in on certain aspects	42
4.2.2 Evaluation Metric on Risk	42
4.2.3 Methods to allocate stocks with the given list of Stock Prediction	42
4.2.4 Trying More Deep Learning Models	43
4.2.5 Applying More Ensemble Models to Our Platform	44
4.2.6 Sector-wise Model Training	44
4.2.7 Backtest with different lengths of period or different date ranges	44
5. References	45
6. Appendix A: Meeting Minutes	49
6.1 Minutes of the 1st Meeting	49
6.2 Minutes of the 2nd Meeting	50
6.3 Minutes of the 3rd Meeting	51
6.4 Minutes of the 4th Meeting	52
6.5 Minutes of the 5th Meeting	54
6.6 Minutes of the 6th Meeting	55
6.7 Minutes of the 7th Meeting	56
6.8 Minutes of the 8th Meeting	57
7. Appendix B Required Hardware and Software	58
7.1 Hardware	58
7.2 Software	58
8. Appendix C: Project Planning	59
8.1 Distribution of Work	59
8.2 Gantt Chart	60
9. Appendix D : Example User Survey	62

1. Introduction

1.1 Overview

1.1.1 Stock Market status quo

Due to recent advances and the convenience of mobile technologies, investors nowadays can easily enter the stock market through mobile applications developed by security brokers. Due to the economic slowdown caused by governments' COVID-19 policies, many central banks are rapidly debasing their currencies to hold off imminent economic collapse and maintain the status quo. As a result, a large amount of capital has flowed into stock markets, as more and more people fear that the value of their hard-earned savings will continue to be eroded by inflation [1]. However, many of them are not experienced in the stock or securities markets, so they have inadequate knowledge and time to manage their investments. The most common and universal way to participate in the market is via a mobile application which helps to create an account and make trading activities according to the user's choice.

1.1.2 Machine Learning and Deep Learning in Trading

Machine Learning (ML) is a study of constructing a model operating under complex algorithms which will improve over the iterative learning process without programmed instructions.

Further stretching, Deep Learning (DL) is an extension of Machine Learning that makes use of multiple artificial neural networks to return higher-level features of the data. The field studies constructing algorithms to perform like the neural network in the human brain.

In recent years, both Machine Learning and Deep Learning models have been applied to a vast range of industries, including the financial industry. Some popular models used by quantitative analysts and data scientists for stock prediction include

1. Convolutional Neural Network (CNN)
2. Long Short-term Memory (LSTM)
3. Recurrent Neural Network (RNN)
4. Reinforcement Learning (RL)
5. Q-Learning (reinforcement learning without an agent)

Some models may be a combination of two (e.g R-CNN, a combination of RNN and CNN) as well. Moreover, the concept of the transformer [2] and BERT model [3], allowed the adaptation of NLP application to the stock prediction using text corpus inputs (market news, stock discussion community, and more).

1.1.3 Algorithmic Trading

Among all the practical applications of ML and DL in the financial industry, such as fraud detection, market insights, and financial advisory, algorithmic trading (or electronic trading) plays a great role. According to Coherent Market Insights, the equity market is expected to contribute \$8.61 billion of the algo-trading market share by 2027 [4]. Moreover, Coalition Greenwich reported that the top 12 investment banks profited around USD 2 billion from the portfolio and algorithmic trading in 2020 [5].

Because the financial markets are vibrant, fast-paced, and hard to predict, algorithmic trading is of great use with its higher entry speed, concrete execution protocol, and lower degree of bias and emotion compared to human-initiated trading. Based on the input data, models are trained to accurately predict prices and can be applied to real markets to form profitable portfolios.

With the diverse pool of strategies, we developed a comprehensive stock trading system with an ensemble of algorithms/strategies that suggests what strategies are appropriate for each user. Our team applies ensemble learning by first filtering out the outlier predictions

and rearranging the contribution of different models in the final price decision. The predicted value of each stock then determines how to distribute stocks in a portfolio and whether the portfolio has an appropriate risk level and offers the best combination of making a maximum profit for each user.

1.2 Objectives

The main goal of our project was to develop a user-friendly algorithmic trading application with a diverse choice of models for stock price prediction and portfolio distribution and experiment with whether such a method can generate greater revenue than a simple buy-and-hold strategy. Moreover, when comparing our project to other existing Robo-advisors or algorithmic trading platforms, we hope that our final product can be better in terms of two aspects.

1. Using ensemble learning, (i.e. using multiple machine learning models) and different trading strategies, we would like to achieve higher accuracy in stock price prediction compared to existing Robo-advisors in the market.
2. Develop a truly automated platform that does not require users to make any judgments or choices during investing.

To achieve these two goals, we have been working on the following objectives:

1.2.1 Database Implementation

The database stores the training data, test data, and machine learning models.

- (a) Historical data is stored for training the models and testing the models
- (b) Real-time data is pushed into the database constantly through APIs, keeping the database updated.
- (c) After a user is classified according to the desired level of risk, a dedicated model is retrieved for making choices according to his risk level.

1.2.2 Machine Learning Model Training

To have good performance in trading, having accurate stock price predictions is very important to making trading decisions. One of the ways to achieve this is to use multiple models to predict different results and then combine the results to produce the final

predictions of stock prices. The method we use here is to filter out some of the outliers/top and bottom-n results and then average out the remaining predictions. Our method of averaging out can be done with weighting. The weighting can be calculated by using two approaches:

1. Based on every prediction, higher weighting is assigned to the data in the main cluster.
2. Based on previous training history, the models with better performances are assigned with the higher weighting.

1.2.3 Risk Quantification

To adequately categorize users' risk-bearing levels, we designed a survey to understand users' financial situations and investment preferences. This includes making the folder after knowing the risk level and the areas of interest of users, we automatically select a few corresponding potential stocks for price and trend prediction. With the price prediction results and the users' profiles, the platform will automatically trade the selected stocks.

1.2.4 Portfolio Presentation

We built a user-friendly web dashboard for visualizing the user's portfolio and the performance of our models compared with the US stock market. It contains the following parts:

- (a) a pie chart is used to visualize the distribution of stocks currently held in the user's portfolio.
- (b) Multi-line charts are used to compare the performance of our strategies and the "buy and hold" strategy.

To achieve the first objective, we scraped approximately 10 years of historical data from the

US stock market and then split it into two parts: the training set and the testing set. After training the models, we filtered out some of the models that returned top and bottom n-results and used the results of the remaining models to calculate the average for stock price prediction.

To achieve the second objective, we use a virtual machine for the database, and it constantly pushes new data into the database.

To achieve the last three objectives, we built a web application with an intuitive UI to handle most of the interaction with the users.

As mentioned above, the financial market is so uncertain and difficult to predict, that one of the major challenges was figuring out how to train an accurate and reliable stock price prediction model. As mentioned above, we use different machine learning models to predict stock prices independently, and then we apply filtered weighting. The major challenge was deciding which models to be dropped out of and how to allocate the weighting of the fund results among the remaining models.

1.3 Literature Survey

There exist many methods of algorithmic trading and also quant firms with electronic trading execution systems. We build on two main areas of related work: Existing algo-trading companies and the technologies currently used.

1.3.1 Existing Algo-Trading Companies

Wealthfront

Some companies use algorithmic trading to perform portfolio management. Machine Learning models deployed in each firm tend to make decisions based on low-risk/high-return rules. One example of a company using a Robo-advisor (a digital platform providing algorithmic financial planning without any human supervision) is Wealthfront (<https://www.wealthfront.com>). Wealthfront users set investing goals and duration before placing the seed money, and then the platform takes care of the rest. It automatically manages portfolios while also allowing customized portfolios of single stocks, ETFs, and other funds. However, Wealthfront provides a rigid list of stocks in creating a portfolio, and users do not get to have a 'personalized' portfolio upon their will [6].

Aqumon

One of Asia's leading trading platforms - Aqumon (<https://www.aqumon.com/en/>), has rapidly developed recently. Aqumon looks similar to the outcome of our project, with four major functions: Long Term Asset Allocation Portfolios, SmartStock Thematic Stock Portfolios, Stock Trading, and Cash Management Portfolios. One of the selling points for Aqumon is that they claim that their portfolios are built using the Nobel-winning Markowitz Efficient Frontier Model and the Black-Litterman Model.

One of the advantages of Aqumon is that they provide various combinations of ETFs for the user to choose from. The users are given advice based on their risk-bearing level, but the

users can make their own choices. However, financial-market newcomers might not know how to make rational decisions. Therefore, our platform provides a fully automated trading platform that does all the trading decisions in the back end without asking for users' decisions.

1.3.2 Stock Price Prediction Models

Research on predicting stock prices included two major techniques: 1) machine learning methods and 2) deep learning techniques. Both techniques can help us to predict the price movement in the future. The accuracy may not be high, but it's better than following a crowd, which tends to buy high and sell low.

Machine Learning Methods

There are several common techniques in the machine learning field that are used for stock price prediction, such as Support Vector Machine (SVM) [7, 8,9], Random Forest (RF) [10,11], and Multiple Linear Regression (MLR) [12,9]. SVM and RF are practical and useful methods for classification in making trading actions (Buy/Sell/Hold), but they are not able to give a numerical prediction of the stock price. MLR is effective for finding the relationship between the historical data and the future price, but it is limited to linear relationships. Deep Learning can overcome these limitations. Hybrid models [10,11] combined with data mining methods and machine learning methods are also used to enhance the improvement.

Deep Learning Methods

Other than machine learning methods, deep learning has drawn our attention as well. Artificial Neural Networks (ANNs)[11], Convolutional Neural Networks (CNNs) [13,14,15,16], and Recurrent Neural Networks (RNNs) [14,15,17,18] are the generally used deep learning methods to tackle price prediction problems. The major advantage of a neural network is that it can solve more complicated problems, as the hidden layers in the neural network can

extract more features from the data. [13] and [16] used a CNN to predict stock price movement by using 1-D price data as input data. A CNN is capable of capturing features from the data in the form of both 1-D and 2-D, which can be used to capture patterns in the stock price data. Other than a CNN, as stock price data is time-series data, an RNN may be useful as well for time series data. Long Short-term Memory (LSTM) [14,15,17,18], an improved version of an RNN, can also be a good choice to handle time-series data. However, the design of LSTM is more focused on historical data, which may not be very effective for predicting future events. Moreover, deep learning methods are quite sensitive to noise, which exists a lot in stock data. Therefore, hybrid models [14,15] have been applied in deep learning methods as well. These models are generally better than standalone models.

The existing research and products usually apply one single model to the prediction of price trends. They focus on optimizing the models or making comparisons between models. However, they usually do not use ensemble learning, i.e. they do not consist of any fusion or integration of the results from different models. Thus, in our project, we spread out the prediction error from different models to lower the risk.

As mentioned above, algorithmic trading now dominates stock market trading. On top of the current systems and algorithms implemented, we developed an application that generates an optimal combination of multiple algorithms and then decides which stocks are most likely to yield a higher return in the trading market.

2. Methodology

2.1 Design

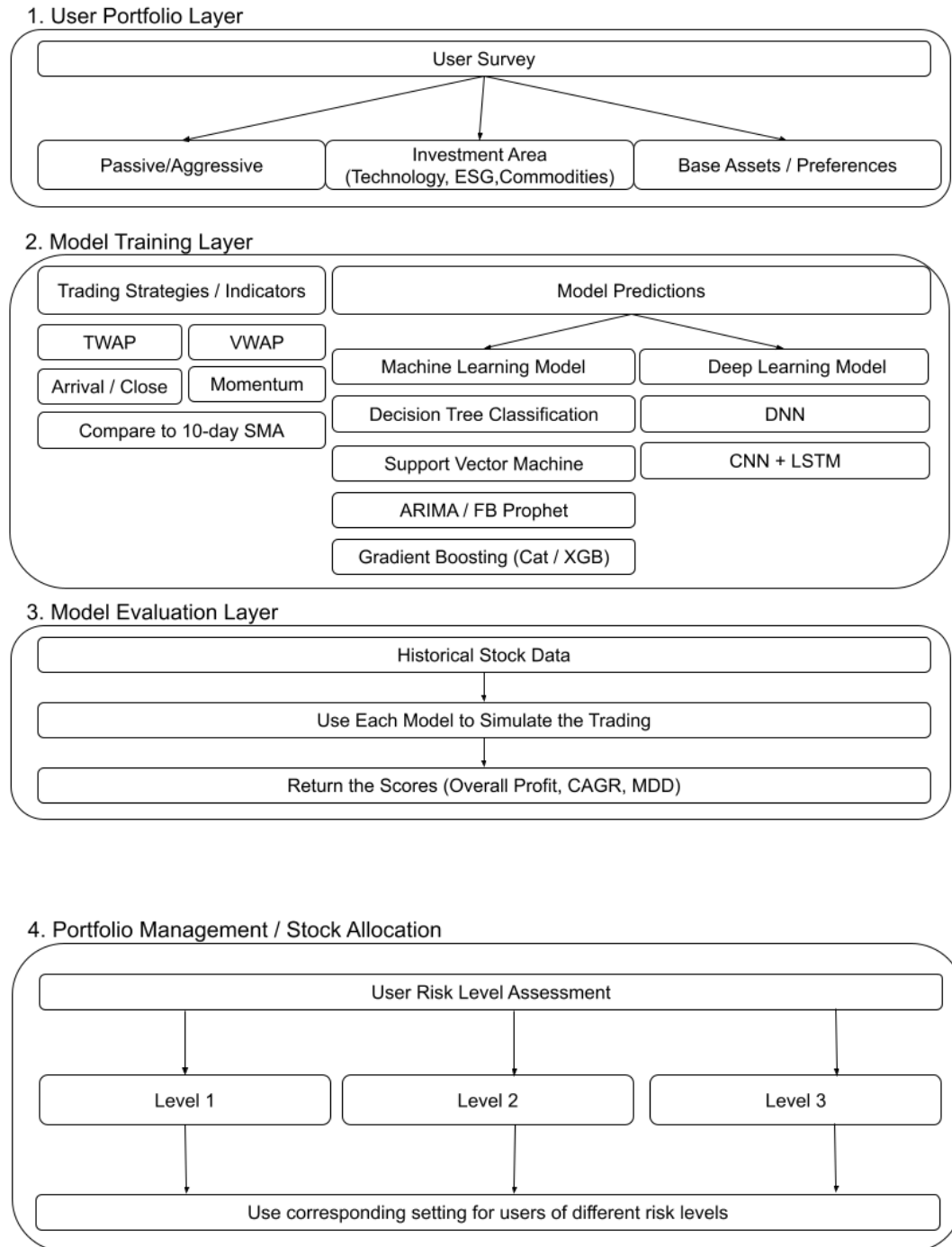


Figure 1. The final design of the customized portfolio pipeline

(COMP/DSCT)

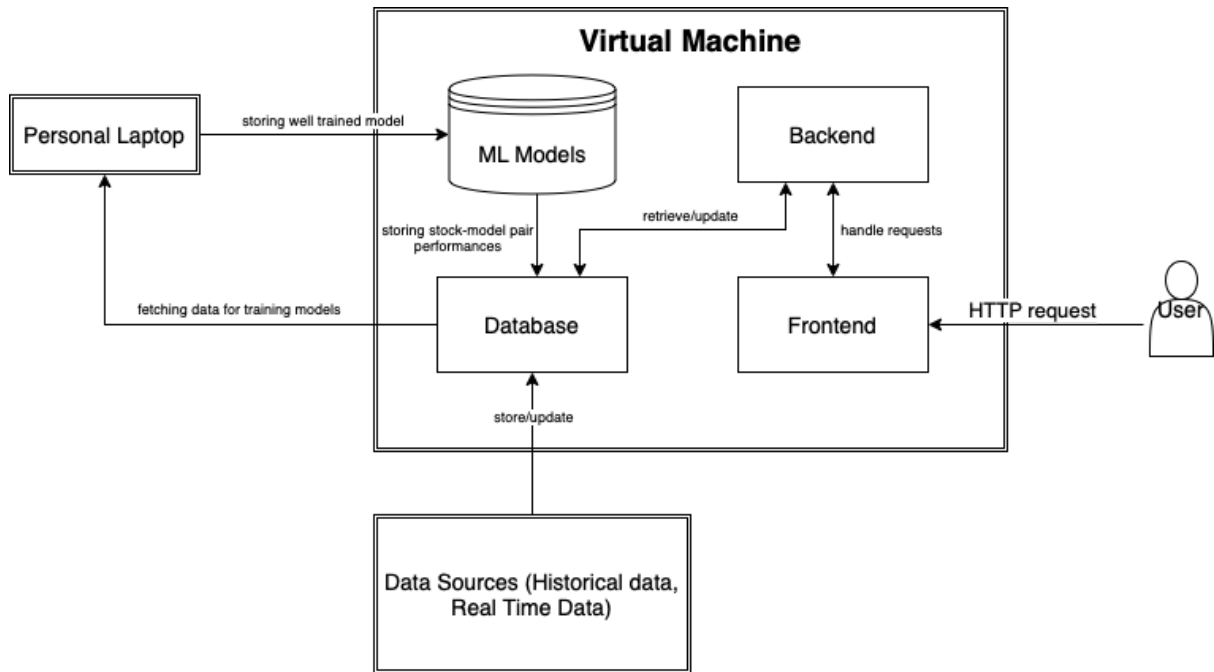


Figure 2. The final design of system architecture for our platform

The pipeline our team designed goes through the following steps:

2.1.1 User Preference Survey

Users are first given a list of questions related to their stock types preference and style of investment. All the responses are recorded in the database and users are classified on a scale of 1 to 3 (1 being extremely passive and 3 being extremely active/risky).

In the user preference survey, we designed and also generated 1,000 dummy data that randomly selected responses from the questionnaire. The detailed list of questionnaires is in Appendix D.

2.1.2 User Classification & Asset Deposit

After the user has completed all the questionnaires, the pipeline finalizes the following regarding the user:

- Which type of portfolio the user expects from the pipeline
- Which category (Blockchain, traveling, Semiconductors, Cloud Computing, Social Media, Entertainment, Retail, Franchise, Real Estate, Telecommunication, Energy & Resources, Commodity, Luxury goods, etc) user prefer to invest in
- How much risk the user is willing to take (user risk level assessment divided into three classes)
- Whether the user is looking for long term investment or relatively shorter-term investment (HFT)

If the users wish to adjust the level on their own, they can re-take the survey (from section 2.1.1) to re-adjust their category.

2.1.3 Selecting the Combination of Algorithms/Strategies

This is the list of algorithms that we used for stock prediction:

1. Machine Learning Algorithms:
 - a. Adapting Machine Learning algorithms (e.g. SVM(Support Vector Machine)
 - b. XGBoost (For higher computational speed and model performance)
 - c. Catboost (Another gradient boosting method)
 - d. Simple DNN (Deep Neural Network) in association with TWAP
2. Deep Learning Algorithms:

(COMP/DSCT)

- a. Deep Learning algorithms like DNN (Deep Neural Network)
- b. Combined Deep Learning algorithms like CNN-LSTM (Convolution Neural Network with Long Short Term Memory)
 - i. CNN-LSTM is used for both regression and classification

Moreover, here is a list of models that we attempted to use, but decided not to add to our model due to the poor prediction score and RMSE scores:

1. ARIMA model: Due to the seasonality in the stock data, it was inefficient to use the time series data with the period of 10 years despite the data preprocessing to convert the data type to stationary
2. The alternative method that we tried was to use the Facebook Prophet library [19]. However, after using the model the performance was not as high as we expected and was eliminated from our decision-making process.

Further analysis of these two models is indicated in section 3. Discussion.

The trading strategy in our project is comparing the 5-day SMA. The system will compare the predicted result with the 5-day SMA to decide to buy or sell.

Each algorithm will be labeled based on its level of risk. Using those labels, each user will be evaluated with a corresponding list of models used for their investment.

2.1.4 Web Dashboard Application

Based on the classification result of the risk level of a user and the stock sector preference gathered from the survey, a set of suitable stock and models will be used to construct a customized user portfolio. The web application serves as a platform to visualize the users' portfolio in an intuitive one-page dashboard. Our design on the dashboard page consists of four major components to show different kinds of essential information to our user:

- Basic Portfolio Information (Total Assets, Total Cash, Total Profit, Risk level)
- Performance Graph (with baseline comparison with the Buy-and-Hold Strategy)
- Stock Distribution Chart
- Transaction History Table

Apart from the dashboard page, some other supplementary pages are needed to show additional information for the users to better understand the performance of their portfolio and trading activities. The web pages in our application include

- Dashboard (main page): which allows users to understand their portfolio at the first glance
- Transaction page: which allows users to access the full list of trading activities made by the customized strategy from our platform

2.2 Implementation

2.2.1 User Survey - Input Form

The input form to be included in the web application has been designed. With the result, we categorize the users into three different levels of risk-bearing. The strategy/algorithm used in the backend is based on the corresponding risk level of the users.

2.2.2 Model Training/Testing/Evaluation

We used Google Colab and Jupyter notebooks for our major model training and testing tools. We tried out various models stated above along with the hyperparameters tuning.

For most of the models, the performance was acceptable and was able to give us useful predictions for making investment choices. Therefore, we decided to keep most of the models for further implementation.

Evaluation of the models will be compared using the 5-day SMA. The system will compare the predicted result with the 5-day SMA to make the decision to buy or sell. Then, with the given seed money, we will compare the P/L of each stock with the model, or without the model (Buy & Hold strategy)

2.2.3 Model Assignment using Volatility of the Model

Based on the dummy user database, we can deduce the risk level of each user by the following steps:

1. All the stocks and model results are paired up to deduce the volatility of the prediction overall.
 - a. In examining the trade record for each model given a stock inside the sector, we deduced the standard deviation of each pair by using "Sector", "Stock",

(COMP/DSCT)

"CAGR_Performance(in %)", and "MDD_Performance(in %)" to calculate the standard deviation

- b. Then, among the standard deviations calculated, we use the standard deviation of CAGR performance as the volatility for each (stock, model) pair
- c. Then, we manually set two cutoff values to classify the volatility into three models (cutoff was selected based on the maximum and minimum volatility of the entire (stock, model) pair, and make the approximate distribution of low, medium, and high class as 1:2:1).

After deducing the risk level of each (stock, model) pair, we can match the level with the risk level responded by each user.

2. Based on the risk level and the sector preference, the user is given the top three stocks with the lowest volatility
3. Then, after setting capital as \$10000 and the handling fee as 0.05%, we use the acquired backtesting result to find out the portfolio CAGR and portfolio performance for each user.

2.2.4 Data Storage & Database Structure

A database is needed for storing the historical data of stocks. It is essential to the models' training and evaluation. MySQL database is the first option since our team is familiar with it and it is suitable for time series data like stocks' price data. For historical data, we have gathered the free historical market data from Stooq which consists of about 5000 stocks' market data. The historical data will be stored in our database as a backup in case of any modification of the data source. Before the training, necessary data preprocessing such as normalization and removing outliers will be conducted for removing the effect on the scale of different features.

We set up a free tier virtual machine instance on the AWS platform and set up a MySQL database on it. We put the historical data we got into the database for backup. Since we are doing our development on our laptops most of the time, we keep the virtual machine stopped to save money.

In our database implemented, there are four kinds of tables:

1. `user_survey`: consist of the results from the user surveys and the classified risk level of a user
2. `stock_hist`: contain the historical price (Open, High, Low, Close) of a stock and the transaction volume
3. `stock_model_pair`: contain the trading decision made by the trained model on a stock
4. `user_portfolio`: contain the full portfolio of a user by combining the performance of the three chosen stock-model pairs

FYP RO1 - Personalized Algo-trading Using Deep Learning & Machine Learning models
(COMP/DSCT)

user_survey		stock_hist		stock_model_pair		user_portfolio	
id	int	id	int	id	int	id	int
age	VARCHAR	date	date	date	date	Date	date
portion	VARCHAR	open	decimal(10,3)	action	varchar	Action_1	varchar
item	VARCHAR	high	decimal(10,3)	price	decimal(10,3)	Price_1	decimal(10,3)
risk	VARCHAR	low	decimal(10,3)	position	int	Pos_bal_1	int
why	VARCHAR	close	decimal(10,3)	cash	decimal(10,3)	Cash_bal_1	decimal(10,3)
term	VARCHAR	volume	int	pos_bal	int	Cum_profit_1	decimal(10,3)
bearing	VARCHAR			cash_bal	decimal(10,3)	Value_1	decimal(10,3)
types	VARCHAR			cum_profit	decimal(10,3)	Sector_1	varchar
profit_exp	VARCHAR			total_bal	decimal(10,3)	Model_1	varchar
conf	VARCHAR					Stock_1	varchar
res1	VARCHAR					Action_2	varchar
res2	VARCHAR					Price_2	decimal(10,3)
tf1	VARCHAR					Pos_bal_2	int
tf2	VARCHAR					Cash_bal_2	decimal(10,3)
tf3	VARCHAR					Cum_profit_2	decimal(10,3)
tf4	VARCHAR					Value_2	decimal(10,3)
stock_type	VARCHAR					Model_2	varchar
risk_level	VARCHAR					Sector_2	varchar
						Stock_2	varchar
						Action_3	varchar
						Price_3	decimal(10,3)
						Pos_bal_3	int
						Cash_bal_3	decimal(10,3)
						Cum_profit_3	decimal(10,3)
						Value_3	decimal(10,3)
						Model_3	varchar
						Sector_3	varchar
						Stock_3	varchar
						Total_cash_bal	decimal(10,3)
						Total_value	decimal(10,3)
						Total_cum_profit	decimal(10,3)
						Buy_and_hold_total	decimal(10,3)
						Capital_bal	decimal(10,3)
						Portfolio_cagr	decimal(10,3)
						Buy_and_hold_cagr	decimal(10,3)
						Portfolio_performance	decimal(10,3)

Figure 3. Structure of database for our platform

2.2.5 Web Dashboard Application Components

A web dashboard has been built for presenting the performance and stock distribution in the user's portfolio. The front-end client was built by ReactJS which allows us to build encapsulated components for developing complex UIs with flexibility and easy management

FYP RO1 - Personalized Algo-trading Using Deep Learning & Machine Learning models
(COMP/DSCT)

of components. For styling, instead of designing every component on CSS, we use the Material UI as the library for basic web page components and styles.

We have also built a Node.js server as the middleware of our front-end client and the database. We use Express to handle the HTTP request from the client-side as well as retrieve the correct data from the MySQL database by queries.

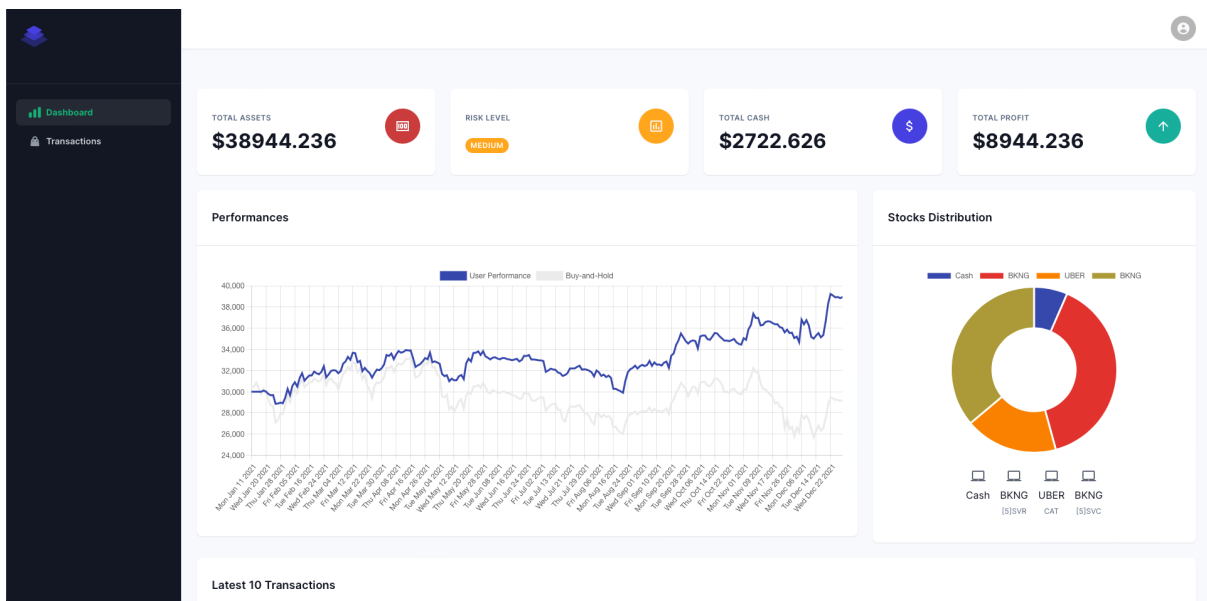


Figure 4. Screenshot of the four components for the basic information of a user

2.2.4.1 Account Information Components

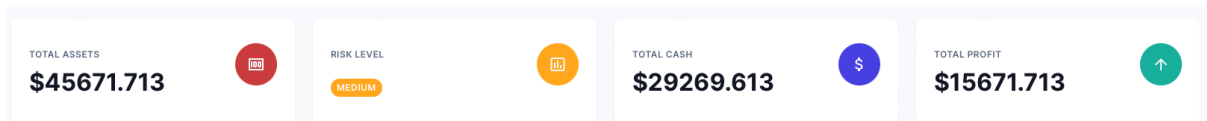


Figure 5. Screenshot of the dashboard page of the web application

The first section of the dashboard contains four components that show the basic information of a user portfolio: the Total Asset Value, the Risk Level, the Total Cash Balance, and the Total Profit.

FYP RO1 - Personalized Algo-trading Using Deep Learning & Machine Learning models
(COMP/DSCT)

2.2.4.2 Performances Graph

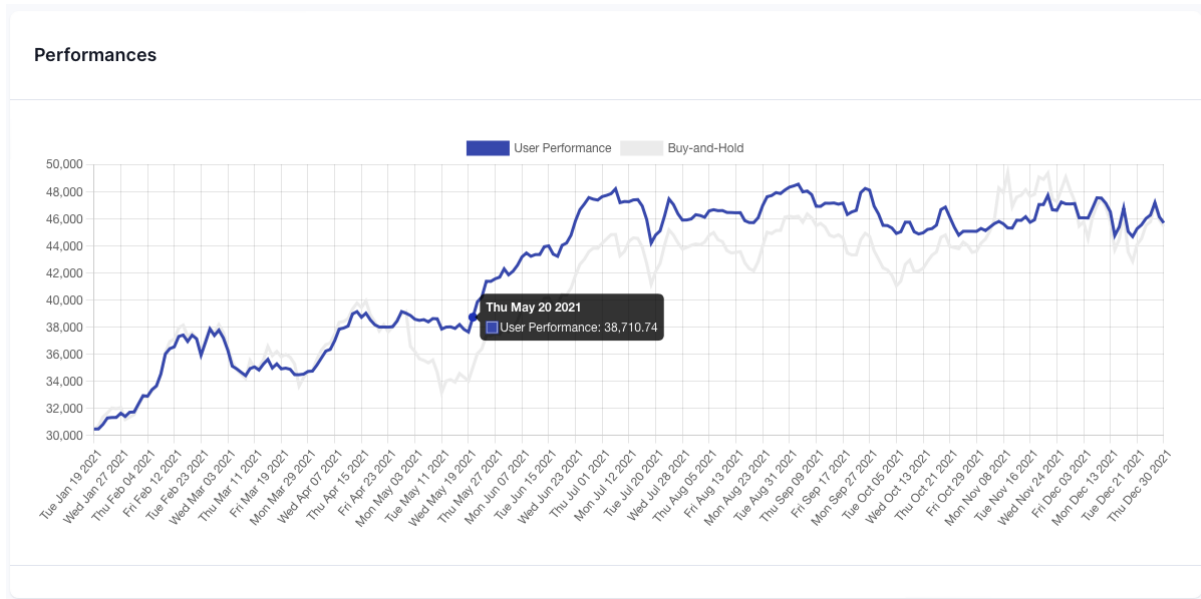


Figure 6. Screenshot of Performances Chart

The multi-lines chart shows the performance of the user by adapting our models and strategies throughout the year, with a baseline performance of the buy-and-hold strategy for reference. The user can toggle over the line to see the exact value of their performance on a specific day, as shown in the figure above.

2.2.4.3 Stock Distribution Chart

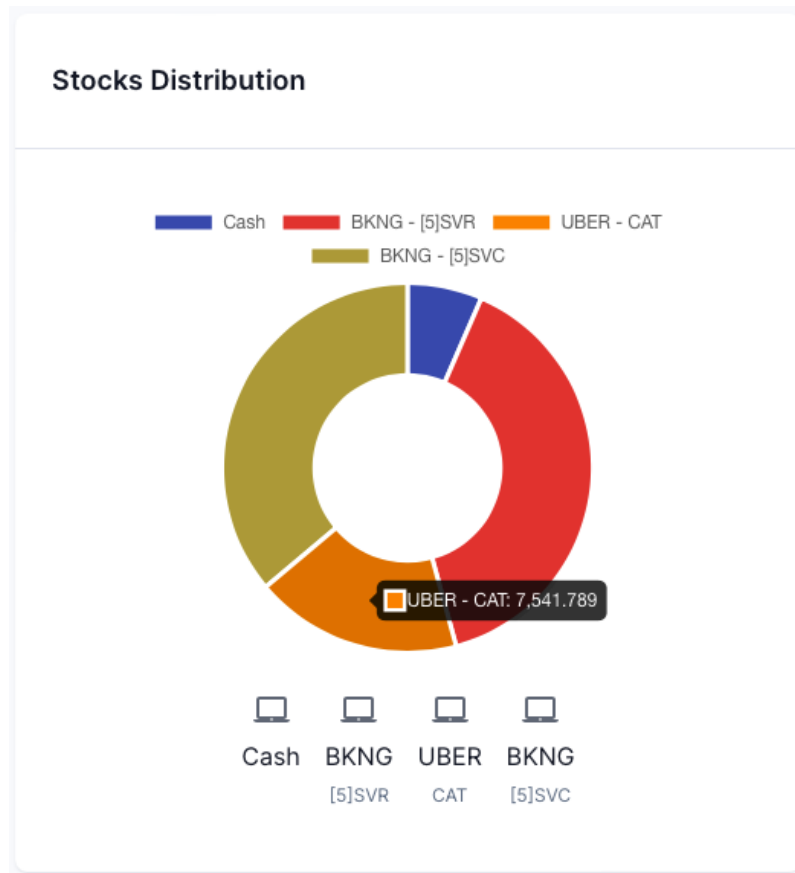


Figure 7. Screenshot of Portfolio Distribution Chart

The doughnut chart can efficiently show a user's portfolio distribution on cash and on all the holding stocks. By toggling on the sectors, a detailed amount of assets allocated to a specific stock will be shown. It gives a clear idea of how many assets were being allocated to a specific stock under our models. Since we are selecting the top 3 stock-model pairs for a portfolio, the chart is also showing the model we used to make trading decisions on a stock.

FYP RO1 - Personalized Algo-trading Using Deep Learning & Machine Learning models
(COMP/DSCT)

2.2.4.4 Transaction History Table

DATE	STOCK	AMOUNT	PRICE	ACTION
Fri Dec 31 2021	T	405	\$24.228	SELL
Wed Dec 29 2021	VZ	193	\$51.133	SELL
Tue Dec 28 2021	VZ	194	\$51.515	BUY
Thu Dec 23 2021	VZ	194	\$51.437	SELL
Fri Dec 17 2021	T	405	\$23.326	BUY
Mon Dec 13 2021	VZ	194	\$49.171	BUY
Fri Dec 10 2021	T	420	\$22.522	SELL
Wed Dec 08 2021	VZ	194	\$49.201	SELL
Tue Dec 07 2021	T	420	\$22.831	BUY
Fri Dec 03 2021	VZ	194	\$50.411	BUY

[View all](#)

Figure 8. Screenshot of the latest transaction history table

The table shows the latest 10 transaction records with the columns of “Date”, “Stock”, “Amount”, “Price” and “Action”. By clicking on the “View all” button, the page will be redirected to the web page, showing the full transaction history over the year.

DATE	STOCK	AMOUNT	PRICE	ACTION
Fri Jan 15 2021	UBER	177	\$56.39	BUY
Thu Jan 28 2021	BKNG	4	\$2036	BUY
Thu Jan 28 2021	BKNG	4	\$2036	BUY
Fri Jan 29 2021	BKNG	4	\$1996.19	SELL
Fri Jan 29 2021	BKNG	4	\$1996.19	SELL
Mon Feb 01 2021	BKNG	4	\$2035	BUY
Mon Feb 01 2021	BKNG	4	\$2035	BUY
Thu Feb 25 2021	BKNG	4	\$2343.9	SELL
Thu Feb 25 2021	BKNG	4	\$2343.9	SELL
Fri Mar 05 2021	BKNG	4	\$2303.43	BUY
Fri Mar 05 2021	BKNG	4	\$2303.43	BUY

Figure 9. Screenshot of the Transaction page

2.3 Testing

We performed backtesting on both the prediction models and user portfolios strategy with some historical data.

2.3.1 Backtesting on the Prediction Models

For the historical data, we first set a cut-off date to separate the dataset. The data before the cut-off date are used to train the model while the data later than the date are used to test the model. The performance of the model was assessed by the various Indicators such as Compound Annual Growth Rate (CAGR) and Maximum DrawDown (MDD).

In daily trading, technical indicators like On-Balance-Volume, Accumulation/Distribution Line, MACD, Aroon Indicator, Average Directional Index, and Stochastic Oscillator will have experimented with model evaluation [20].

Instead of using the r_square score or mean squared error to measure model performance, we developed a python class to simulate the trading process and calculated the CAGR and MDD to evaluate the model performance.

In the model backtesting part, we have the following assumption:

1. Initial Capital of each backtesting = USD 10,000
2. Period of each backtesting: 1 year
3. Backtest the model with one stock at a time

The model backtesting pipeline was implemented as follows (See Figure 5):

1. Use the trained model to make predictions for different stocks in different industries
2. For each model and each stock in each testing period (1-year each), we input the historical data and prediction to simulate the trading process
3. Calculate the CAGR for each result

4. If the mean CAGR for all stocks for one model is better than the mean of the Buy & Hold Strategy for all stocks, then we say it is acceptable and will be included in our model list.

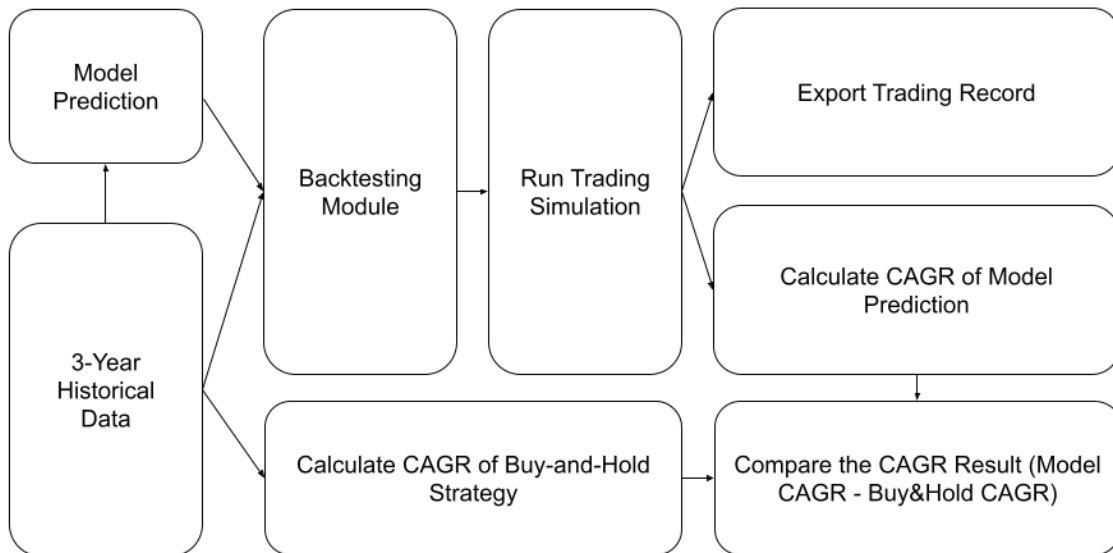


Figure 10. Backtesting Pipeline of ML/DL Models

For model prediction, it returned a list of “Buy”, ”Sell” and “Hold”, along with the timestamp to take these actions. The decisions were made based on comparing the predicted price to the simple moving average of the actual price for the previous days. After testing, we found out that the model's performance of taking the simple moving average of the past 5 days is the most suitable one. After the backtesting of the models, the results are compared to the buy&hold strategy in terms of CAGR and MMD.

2.3.2 Backtesting on User Portfolio

Our approach to backtesting our platform is different from that of the prediction model. This backtesting is based on the user portfolio, which included risk assessment, portfolio management, price prediction, and auto-trading. As we had a risk assessment and portfolio

management for the users, there are quite a lot of combinations to test. For simplicity, we set up 1000 “fake” users for the portfolio backtesting.

The 1000 dummy users were generated with python and they were classified into three different risk levels. From 2.2.3, each (stock, model) pair consists of a volatility score, a risk level, and the sector to which the stock belongs. If a user is with a “high” risk level, we keep those (stock, model) pairs that are classified as “high” risk level AND match the sector preference given by the user. As we are trying to obtain a better result in that risk level, we sorted (stock, model) pairs by the volatility score descendingly and selected the top 3 stocks for their portfolio.

In users’ portfolio backtesting, we have the following assumption:

1. The number of (stock, model) pairs to invest = 3
2. Initial Capital for each (stock, model) pair = USD 10,000
3. Period of backtesting: 1 year (started on Jan 1 and ended on Dec 31)

We used 7 models in total and there are 48 stocks among 13 sectors. Below is how we perform classification on (stock, model) pairs. We used data from 2012-2018 (7 periods, with 1 year long each) to perform backtesting for each (stock, model) pair. Then we aggregated the standard deviation of the CAGR of each (stock, model) pair as the volatility.

The thresholds of the risk level are 13 and 30, which approximately gives us the ratio of the (stock, model) pairs in each level to 1:2:1. (stock, model) pairs with volatility scores smaller than 13 will be classified as a low level while pairs with volatility scores between 13 and 30 will be classified as medium level and those with volatility scores larger than 30 will belong to a high level.

Below is actual data of (stock, model) pairs volatility score example from our backtesting:

	Stock					
Model	FB	AMZN	NFLX	IBM	LMT	PVH
CNN LSTM (Regression)	Medium	Medium	High	Low	Low	Medium
CNN LSTM (Classification)	Medium	High	High	Medium	Medium	High
TWAP	High	High	High	Medium	Medium	High
SVR	Medium	Medium	High	Medium	Medium	High
SVC	High	Medium	High	Low	Medium	High
XGBoost	Medium	Low	Low	Low	Low	Low
CatBoost	Medium	Low	Low	Low	Low	Low

Table 1. (stock, model) pairs volatility score example

Risk Level	Example (Stock, Model) Pairs				
Low	(AMZN, XGB)	(AMZN, CAT)	(LMT, CNN_LSTM_REG)	(IBM, SVC)	etc.
Medium	(IBM, TWAP)	(FB, CAT)	(FB, CNN_LSTM_CLASS)	(LMT, SVR)	
High	(FB, SVC)	(PVH, SVR)	(PVH, CNN_LSTM_REG)	(NFLX, TWAP)	

Table 2. (stock, model) pairs in each risk level

For each user, he will be assigned with a risk level. If the user is being classified as low-risk level, then he can only invest the stock that is inside the low-risk level (stock, model) pairs using the specific trained model.

Then we will filter away the stocks that are not on his stock preference list. With the remaining options, we will sort the result by volatility score descendingly and assign the **top 3** (stock, model) pairs for him.

For both backtesting, we keep 2019 - 2021 data as unseen data. Therefore no training processes utilized the information stored in the 2019 - 2021 data.

For each user portfolio backtesting, we assigned the same amount of initial capital (i.e. USD 10,000) and used historical data to simulate the trading process. After that, we summarized the CAGR and made comparisons with simple “buy and hold” strategies.

2.3.3 Web Dashboard Application Testing

To test the web dashboard application, we focused on three aspects, which are data accuracy, compatibility on different devices, and user feedback.

To test the data accuracy, we pushed different user portfolios from the database into the front-end application to ensure the correct information is shown on different components. We have performed the process across all the dummy user portfolios in the database to finalize the UI design for different portfolios.

To test the compatibility on different devices, we tried out different window sizes and launched the web application on different devices to make sure that every component is not overlapping and complete graphs and charts are shown accordingly. We have designed four different layouts for different window sizes to ensure the responsiveness of our web dashboard application.

To gather user feedback on our web application, we asked 8 volunteer testers to try out our web dashboard application and they are asked to give feedback by completing a Google form afterward.

2.4 Evaluation

In the evaluation section, we will go over the objectives and whether those are achieved, and to what extent the research was done.

2.4.1 Evaluation of Customization

After a series of backtesting and assigning the (stock, model) prediction to each user, our team succeeded in calculating the P/L of 3 years (2019 - 2021) using the top three (stock, model) pairs for each user. The following are the first 10 users and the result is 2021. The performance includes the Risk Level, performance by the portfolio, and the comparison between the CAGR of the portfolio and simple buy&hold strategy for selected three stocks.

UserID	Risk_Level	Year	Portfolio_Performance	Stock_Count	Buy&Hold_CAGR	Portfolio_CAGR
1	low	2019	-2.68%	3	17.69%	15.01%
2	medium	2019	8.32%	3	37.22%	45.55%
3	low	2019	0.32%	3	27.07%	27.40%
4	medium	2019	13.22%	3	15.21%	28.43%
5	medium	2019	13.22%	3	15.21%	28.43%
6	medium	2019	13.22%	3	15.21%	28.43%
7	medium	2019	-4.33%	3	43.98%	39.66%
8	medium	2019	13.22%	3	15.21%	28.43%
9	low	2019	0.32%	3	27.07%	27.40%
10	medium	2019	31.35%	3	47.73%	79.08%

Table 3. Performance and CAGR of first 10 users in 2019

UserID	Risk_Level	Year	Portfolio_Performance	Stock_Count	Buy&Hold_CAGR	Portfolio_CAGR
1	low	2020	12.24%	3	-28.02%	-15.78%
2	medium	2020	17.36%	3	40.38%	57.74%
3	low	2020	11.64%	3	-11.56%	0.08%
4	medium	2020	53.00%	3	-8.83%	44.17%
5	medium	2020	53.00%	3	-8.83%	44.17%
6	medium	2020	53.00%	3	-8.83%	44.17%
7	medium	2020	-3.27%	3	42.38%	39.11%
8	medium	2020	53.00%	3	-8.83%	44.17%
9	low	2020	11.64%	3	-11.56%	0.08%
10	medium	2020	108.33%	3	22.07%	130.40%

Table 4. Performance and CAGR of first 10 users in 2020

FYP RO1 - Personalized Algo-trading Using Deep Learning & Machine Learning models

(COMP/DSCT)

UserID	Risk_Level	Year	Portfolio_Performance	Stock_Count	Buy&Hold_CAGR	Portfolio_CAGR
1	low	2021	-14.53%	3	25.40%	10.87%
2	medium	2021	47.16%	3	42.22%	89.38%
3	low	2021	-17.83%	3	42.74%	24.91%
4	medium	2021	28.15%	3	3.87%	32.02%
5	medium	2021	28.15%	3	3.87%	32.02%
6	medium	2021	28.15%	3	3.87%	32.02%
7	medium	2021	9.03%	3	-2.01%	7.02%
8	medium	2021	28.15%	3	3.87%	32.02%
9	low	2021	-17.83%	3	42.74%	24.91%
10	medium	2021	45.86%	3	14.22%	60.08%

Table 5. Performance and CAGR of first 10 users in 2021

2.4.2 Evaluation of Our Platform

2.4.2.1 Overall performance

As discussed in 2.4.1 as customization, we calculated the P/L of 3 periods with 1 year each (2019 - 2021) using the top three (stock, model) pairs for each 1,000 dummy users. We averaged out all the results. Here is the summary of the comparison between the average CAGR of the “Buy & Hold” strategy and making transactions with our platform.

	Platform Performance	Buy and Hold CAGR	Our Platform CAGR
Average	26.19%	21.94%	47.79%

Table 6. Average of Platform CAGR, “Buy and Hold” CAGR, and Platform Performance

As observed, the CAGR performance of our platform performs better than the simple “Buy & Hold” strategy and therefore makes more profit based on the price prediction value for a combination of (stock, model).

2.4.2.2 Performance by year

We performed some analysis on the CAGR performance by year. We averaged out the data in backtesting and got the following result:

(COMP/DSCT)

Year	Platform Performance	Buy and Hold CAGR	With Platform CAGR
2019	9.24%	30.50%	39.74%
2020	43.81%	17.15%	60.95%
2021	25.53%	18.16%	42.68%

Table 7. Average of Platform CAGR, “Buy and Hold” CAGR, and Platform Performance

From the result above, we can see that in all 3 different periods, our platform performance outperformed the “Buy and Hold” Strategy. By further analysis of the maximum and minimum of the “Buy and Hold” strategy and our model performance below, we can see that our platform is doing better than the “Buy and Hold” strategy most of the time.

Year	Buy and Hold CAGR	With Platform CAGR
2019	-20.04%	-20.84%
2020	-31.80%	-16.05%
2021	-12.59%	0.2%

Table 8. Minimum of Platform CAGR, “Buy and Hold” CAGR, and Platform Performance

Year	Buy and Hold CAGR	With Platform CAGR
2019	119.50%	124.75%
2020	98.76%	137.33%
2021	90.55%	116.08%

Table 9. Maximum of Platform CAGR, “Buy and Hold” CAGR, and Platform Performance

Moreover, we would like to know how many users were getting more profit or getting less loss by using our platform than that of the “Buy and Hold” strategy. Below are the percentages of users obtaining positive platform performance, which means the platform CAGR is larger than “Buy and Hold” CAGR:

Year	Outperformance
2019	80%
2020	91%
2021	89%

Table 10. Percentage of users obtaining positive platform performance

FYP RO1 - Personalized Algo-trading Using Deep Learning & Machine Learning models
(COMP/DSCT)

What our potential customers will be more concerned about is the chance of losing their investment. Therefore, we counted the number of users obtaining positive CAGR in two strategies in Table 11. In a usual stock market environment, our platform can help more people to obtain positive CAGR while our platform can maintain an adequate performance when the market is being hit severely, such as COVID-19.

Year	# of Users that have Positive CAGR in “Buy and Hold” Strategy	# of Users that have Positive CAGR in our Platform	Difference
2019	981	991	+10
2020	553	981	+428
2021	956	993	+37

Table 11. The number of users obtaining positive CAGR in two strategies

The figures below are the percentage of users that obtained a positive CAGR with our platform **given that** the users obtained a negative CAGR in the “Buy and Hold” strategy. Our platform can help them to reduce the chance to lose money and even make some profit for them.

Year	# of Users that have negative CAGR in “Buy and Hold” Strategy	# of Users that have (Negative CAGR in “Buy and Hold” Strategy AND Positive CAGR in our Platform)	%
2019	12	11	91.7%
2020	440	430	97.7%
2021	37	37	100%

Table 12. Percentage of users with positive CAGR with our platform **given that** they have negative CAGR in the “Buy and Hold” strategy

2.4.2.3 Performance by risk level

Another approach to interpreting the result is to perform analysis on different risk levels. For each user, we will have results of 3 testing periods (year 2019, 2020, and 2021). Because we assume all users will join us for 1-year only and we do not know when they will choose to join us. So we simply take an average of all these 3 testing periods to aggregate the result for each user.

FYP RO1 - Personalized Algo-trading Using Deep Learning & Machine Learning models
(COMP/DSCT)

After aggregation, we obtained the following result, we can see that our platform's CAGR is aligned with the risk levels being assigned to the user. A higher risk level gives a higher return. The reason why the platform performance is lower in the “High” risk level than “Medium” risk level is because the “High” risk level consists of more volatility (huge rise or huge drop) The model may not be able to predict this volatility and avoid losing money from the market situation.

Risk Level	Platform Performance	Buy and Hold CAGR	With Platform CAGR
High	8.74%	61.20%	69.94%
Medium	33.91%	14.23%	48.14%
Low	5.57%	19.24%	24.80%

Table 13. Average of Platform CAGR, “Buy and Hold” CAGR and Platform Performance

Risk Level	Buy and Hold CAGR	With Platform CAGR
High	86.68%	97.89%
Medium	81.43%	91.66%
Low	39.51%	48.28%

Table 14. Maximum of Platform CAGR, “Buy and Hold” CAGR, and Platform Performance

For the minimum of the performance below, we see that our platform is going better in both “High” and “Medium” levels but not in the “Low” level. But the small difference in the “Low” level is acceptable compared to the huge outperformance in the other two levels.

Risk Level	Buy and Hold CAGR	With Platform CAGR
High	10.58%	15.85%
Medium	-9.08%	-8.49%
Low	-0.649%	-2.37%

Table 15. Minimum of Platform CAGR, “Buy and Hold” CAGR, and Platform Performance

In order to have a clear picture of the outperformance, we break it down into the different risk levels. We see that our platform is doing good in the “Low” level and doing much better in the other two levels. This result in Table 12 is consistent with the performance in Table 11 that our platform may not be very good in trading with low-risk level users.

Risk Level	# of Users	# of Users with Positive Platform Performance	%
High	148	125	84.5%
Medium	706	676	95.8%
Low	139	87	62.5%

Table 16. Percentage of User Portfolio with Positive Platform Performance

2.4.3 Evaluation of Web Dashboard Application

To evaluate the quality of our web dashboard application, our team asked the testers to rate our application on different evaluation metrics on a scale of 1 to 5, with the higher value being the more favorable. The feedback we received was as follows:

Evaluation Metrics	Average Rating
User-friendliness	3.9 / 5.0
Informative	4.1 / 5.0
UI design	4.6 / 5.0
Interactiveness	3.8 / 5.0

3. Discussion

3.1 Challenges

The challenges we faced in the project were the following:

- Lack of precise risk metric

We faced difficulty in accurately portraying the risk level based on the volatility of the (stock, model) pair because there are limitations in the data we acquired from the Stooq data API. As a work-around, we used CAGR volatility to assess the level of 'risk'. Further work that can be achieved will be discussed in section 4.2

- Model selection

We chose models to be used in our platform with RMSE, Mean Absolute Error, and $r_squared$ score to determine whether the model is sufficient. Through such metrics, we decided to drop the time series and experiment in a more delicate way to make the dataset stations models (ARIMA, ARCH/GARCH, and Facebook Prophet). However, we are wary of improving the performance. Again, this will be discussed in section 4.2.

- Real-time Market Data Collection

As we mentioned in the project objectives, we planned to collect some real-time market data through API and automate the data pre-processing procedure for storing the market data in our database. Since we failed to find a free and reliable data source for the real-time market data, we did not include the real-time data during the model training and testing.

- Risk quantification

Our team has quantified the risk of a user based on the choices made in the user preference survey. However, the method we applied for quantifying risk is simple and naive when compared to professional quantitative risk analysis.

(COMP/DSCT)

Our team lacked the knowledge and experience to perform a proper risk analysis on an actual user, so we considered the object of risk quantification on users as partially achieved.

3.2 Relationship with the research problem or hypothesis

From the data stated above, we can conclude that with the metric we proposed (CAGR summation of 1,000 users), the model we proposed has a higher revenue compared to the simple Buy & Hold strategy. To rephrase what the report suggested in the Objectives section, our team plans “ to develop a **user-friendly algorithmic trading application** with a diverse choice of models for stock price prediction and portfolio distribution and experiment whether such a **method can generate greater revenue than simple buy-and-hold strategy**”.

Before we conducted research, we assumed that instead of holding the stock, using the models to predict the stock price and make decisions accordingly will perform better. Based on our metric as returned average CAGR, we conclude that using integrated models indeed performs better than the simple Buy & Hold strategy.

4. Conclusion

4.1 Summary of Achievements

During the FYP project, our team accomplished the following:

1. Constructed an example user survey to collect information about the type of portfolio, category/sector that the user is interested in, the level of risk the user is willing to take, and the term of investment
2. Selected the following algorithms for the price prediction
 - a. Final model list:
 - i. SVM(Support Vector Machine) - classifier and regressor
 - ii. Gradient Boost models - XGBoost / CatBoost
 - iii. Simple DNN (Deep Neural Network) in association with TWAP
 - iv. Combined Deep Learning algorithms like CNN-LSTM (Convolution Neural Network with Long Short Term Memory) - classifier and regressor
 - b. Experimented model:
 - i. Random Forest / Decision Tree
 - ii. Time series models (ARIMA, ARCH, GARCH model, Facebook Prophet library)
 - iii. Simple MLP model
3. Model training, testing, and evaluation using the customized backtesting class module to simulate the trading process and calculate the CAGR and MDD
4. Matched three (stock, model) pairs for 1,000 dummy users and calculated their portfolios' profit and loss
5. Built a user-friendly web dashboard application to interact with the user and visualize the portfolio by charts and graphs

4.2 Ideas for further development

Our original plan was to combine all the models together in an ensemble and create portfolios with multiple models combined. To further enhance the project, we could consider the following method.

4.2.1 Zooming in on certain aspects

1. Train algorithms like SVM (Support Vector Machine), CNN-LSTM, RNN, DNN, and Random Forest only for stocks within each sector
2. Determine each model's volatility and assign models based on the user's desired risk level
3. Assign a (stock, model) pair to each user, and calculate the profit and loss

Here is the list of additional works that can further be developed for the project.

4.2.2 Evaluation Metric on Risk

In assessing the risk, we could use different metrics to evaluate the risk level of each (stock, model) pair. Currently, the level of risk is assessed based on the CAGR volatility of each (stock, model) pair. Therefore, we could use alternative risk metrics (such as beta-systematic risk, Sharpe ratio, VaR - Value at Risk, R-squared value, and more) to improve the risk portrayed in the portfolio.

4.2.3 Methods to allocate stocks with the given list of Stock Prediction

After the stocks have been predicted with a given list of models, prices could be filtered and then averaged out to deduce the final list of predicted prices for stocks. In this step, we could also think of adding the traditional trading strategy implemented.

e.g. Customer A is categorized as a level 3 user (fairly passive investor)

1. Use strategies and algorithms in level 3 (for example, it can be VWAP, KNN, and TWAP)
2. Evaluate stocks in the suggested portfolio with the list of strategies given above. For specific stock A, there will be three different evaluations available.
 - a. VWAP: \$20.0
 - b. KNN : \$30.0
 - c. LSTM: \$40.0

Then the final returned stock value for stock A is \$30 in this example. Apply the algorithm and average out (step 1 and step 2) to all the values added in the 'basket' of the portfolio.

Our current model is simply dividing the asset into three ways, and simulating the investment. In our backtesting model, the initial investment of \$10,000 was distributed three-way to three different stocks that are under the selected category and risk level, and those three are the stocks with the lowest CAGR volatility within that criteria. However, we could improve the model by making the distribution take the risk into account, and distribute the assets accordingly.

4.2.4 Trying More Deep Learning Models

With more research and studies on Reinforcement Learning and Q-learning models, our team could add these models as candidates for stock prediction. Reinforcement Learning is one of the machine learning models where the agent (decision maker) makes an action (sequential decision) in order to maximize the reward given in the environment. Although the stock market is unpredictable and normally the Markov Decision Process (the environment in which all states are Markovian and observable), Reinforcement Learning can be used with model-free predictions like Monte-Carlo learning (episodic MDPs) or Temporal-Difference (TD) learning (updating the value based on estimated returns).

4.2.5 Applying More Ensemble Models to Our Platform

As the stock market data is full of noise and the information stored in a dataset can be complex and in high-dimension, we could try more ensemble models empowered with the ability to capture features from complex datasets.

One of the original plans was to combine different models together to perform ensemble learning methods. For example, with different price prediction results, we could treat these results as a vector input to predict the actual price data and further perform modeling, such as linear regression. In this case, we could treat the coefficient of each model result as the significance of the model in the prediction process. The coefficient of the model will be larger if the model prediction is closer to the actual price.

4.2.6 Sector-wise Model Training

Different stocks in the market consist of different volatilities and complexities. This may be caused by policy changes or supply chain management. In general, industrial influence such as the policy of “widely adopting solar power in US government facilities”, which may boost the performance of stocks in the ESG sector. It would be better if we could perform model training sector by sector and evaluate if there is a significant difference between the models’ performance.

4.2.7 Backtest with different lengths of period or different date ranges

We used 3 years for each backtesting period, and each of them started on Jan 1 and ended on Dec 31. But in reality, our users may join us at different times, like joining after the first quarter of the year or in the middle of the year. So for the backtesting part, we could use a 4-quarter time frame in all 3 years. For example, the first period could be [2019Q1 ~ 2019Q4] and the second period could be [2019Q2 ~ 2020Q1], and so on. Then we would

FYP RO1 - Personalized Algo-trading Using Deep Learning & Machine Learning models

(COMP/DSCT)

have 9 periods instead of 3 periods only. This could be useful to further enhance our platform performance.

5. References

1. J. Cox. (2021, April 9). "Investors have put more money into stocks in the last 5 months than the previous 12 years combined". *CNBC*.
<https://www.cnbc.com/2021/04/09/investors-have-put-more-money-into-stocks-in-the-last-5-months-than-the-previous-12-years-combined.html> (accessed 15 Oct, 2021).
2. C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi and J. Zhong, "Attention Is All You Need In Speech Separation," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 21-25
<https://arxiv.org/abs/1706.03762> (Accessed 2 April, 2022)
3. Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805,2018 <https://arxiv.org/abs/1810.04805> (Accessed 2 April, 2022)
4. Insights, Coherent Market. "Algorithmic Trading Market to Surpass US\$ 25,257.0 Million by 2027", CMI, 3 Oct. 2019.
<https://www.coherentmarketinsights.com/market-insight/algorithmic-trading-market-2476> (accessed 12 Oct, 2021).
5. C. Whittal. (2021, February 19). "US banks reap boom in the Departmentbond portfolio and algo trading".
<https://www.ifre.com/story/2742168/us-banks-reap-boom-in-bond-portfolio-and-algo-trading-dfxq32cqc8> (accessed 9 Oct, 2021).
6. Microsoft, "Cryptocurrency System Using Body Activity Data," Int'l Patent WO 2020/060606 A1, Mar. 26, 2020.
7. B. M. Henrique, V. A. Sobreiro & H. Kimura. (September 2018). "Stock price prediction using support vector regression on daily and up to the minute prices". *The Journal of Finance and Data Science*,4(3).

- <https://www.sciencedirect.com/science/article/pii/S2405918818300060> (accessed 14 Aug, 2021)
8. S. Madge. (2015). "Predicting Stock Price Direction using Support Vector Machines". https://www.cs.princeton.edu/sites/default/files/uploads/saahil_madge.pdf (accessed 15 Aug, 2021).
 9. G. Vaishnavi, V. R. Shriya & K. Ashwini. (2019). "Stock Market Prediction using Linear Regression and Support Vector Machines". *International Journal of Applied Engineering Research ISSN 0973-4562*, 14(8). https://www.ripublication.com/ijaer19/ijaerv14n8_24.pdf (accessed 12 Sep, 2021).
 10. S. R. Polamuri, K. Srinivas & A. K. Mohan. (September 2019). "Stock Market Prices Prediction using Random Forest and Extra Tree Regression". *International Journal of Recent Technology and Engineering*, 8(3). https://www.researchgate.net/publication/347994783_Stock_Market_Prices_Prediction_using_Random_Forest_and_Extra_Tree_Regression (accessed 23 Aug, 2021).
 11. M. Vijh, D. Chandola & V. A. Tikkiwal. (16 April 2020). "Stock Closing Price Prediction using Machine Learning Techniques". *Procedia Computer Science*, 167. <https://www.sciencedirect.com/science/article/pii/S1877050920307924> (accessed 10 Sep, 2021).
 12. S. Shakhla1, B. Shah1 & N. Shah1. (Oct 2018). "Stock Price Trend Prediction Using Multiple Linear Regression". *International Journal of Engineering Science Invention (IJESI)*, 7(10). https://www.researchgate.net/publication/341322930_Stock_Price_Trend_Prediction_Using_Multiple_Linear_Regression (accessed 25 Aug, 2021).
 13. S. Chen & H. He. (2018). "Stock Prediction Using Convolutional Neural Network". Presented at AIAAT. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1757-899X/435/1/012026/pdf> (accessed 31 Jul, 2021)

14. W. Lu, J. Li & Y. Li. (24 November 2020). "A CNN-LSTM-Based Model to Forecast Stock Prices". *Artificial Intelligence for Smart System Simulation*, 2020.
<https://www.hindawi.com/journals/complexity/2020/6622927/> (accessed 12 Sep, 2021)
15. S. Mehtab & J. Sen. (8 November 2020). "*Stock Price Prediction Using CNN and LSTM Based Deep Learning Models*". Praxis Business School, Kolkata, INDIA, Department of Data Science. <https://arxiv.org/pdf/2010.13891.pdf> (accessed 30 Sep, 2021).
16. A. Siripurapu. 2015. "*Convolutional Networks for Stock Trading*". Stanford University, Department of Computer Science.
http://vision.stanford.edu/teaching/cs231n/reports/2015/pdfs/ashwin_final_paper.pdf (accessed 28 Sep, 2021).
17. A. Moghar & M. Hamiche. (2020). "Stock Market Prediction Using LSTM Recurrent Neural Network". *Procedia Computer Science*, 170.
<https://www.sciencedirect.com/science/article/pii/S1877050920304865> (accessed 23 Sep, 2021).
18. J. Qiu, B. Wang & C. Zhou. (3 January 2020). "*Forecasting stock prices with long-short term memory neural network based on attention mechanism*".
<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0227222> (accessed 21 Sep, 2021).
19. Facebook Team, "Forecasting at scale," Prophet. [Online]. Available:
<https://facebook.github.io/prophet/> (Accessed: 19-Apr-2022)
20. C. Potters. (30 April 2021). "*7 Technical Indicators to Build a Trading Toolkit*".
<https://www.investopedia.com/top-7-technical-analysis-tools-4773275> (accessed 13 Oct, 2021).
21. "Free personality test". (n.d.). <https://www.16personalities.com/free-personality-test> (accessed 12 Oct, 2021).

22. Smith, Christopher. "Wealthfront Review: How It Works, Pros & Cons." Clark Howard, 27 Apr. 2021.

<https://clark.com/personal-finance-credit/investing-retirement/wealthfront-review/>

(accessed 1 Oct, 2021).

23. H. Sutcliffe. "COVID-19: The 4 building blocks of the Great Reset." World Economic Forum. <https://www.weforum.org/agenda/2020/08/building-blocks-of-the-great-reset/>

(accessed 6 Feb, 2022).

6. Appendix A: Meeting Minutes

6.1 Minutes of the 1st Meeting

Date: 7 Aug 2021

Time: 11:00 - 11:57

Via: Discord

Participants:

KWOK, Ue Nam (20597580 unkwok)

HO, Yat Man Peter (20606082 ympho)

KONG, Kin Cheung (20595166 kckongaa)

MUN, Sun Bin (20455378 sbmun)

Meeting Discussion:

1. Contact the professor
2. Starting to prepare the proposal (Due on 17 Sept)
 - Due on 17 Sept
 - Suggested page number: 6
 - The task for this week: Overview - Sun and Thomas
 - The task for this week: Objective - Peter and Anthony
3. Discussing the direction
 - More focusing on Machine Learning
4. Schedule meeting 3-4 weeks later for discussing different methodologies
5. Regular meeting (Sunday 10 am)
6. Focusing on the pipeline (X on the GUI or application platform/server)
 - Data Extraction (Where to get stock data?)
 - Data Preprocessing
 - Model Design (Different algorithms and strategies)
 - Model Training
 - Evaluation (Evaluation Method? Simulate the trading once)

Main focus: devising the unique algorithm for the stock prediction

Task Distribution:

1.1 Overview - Sun, Thomas

1.2 Objective - Peter, Anthony

1.3 - Altogether next week

Research Distribution:

- Sun: Algorithm for trading (VWAP, TWAP...) + some previous Python/ML implementation of these algorithms
- Thomas: Sentiment analysis/NLP
- Peter: previous FYP possible competitor
- Anthony: CNN or DNN (LSTM) for stock price data

Next Meeting: 14 Aug 2021

6.2 Minutes of the 2nd Meeting

Date: 14 Aug 2021

Time: 10:30 - 11:30

Via: Discord

Participants:

KWOK, Use Nam (20597580 unkwok)

HO, Yat Man Peter (20606082 ympho)

KONG, Kin Cheung (20595166 kckongaa)

MUN, Sun Bin (20455378 sbmun)

Meeting Discussion:

1. Confirmation on using US Stock Market
2. Brainstorm on the final product
 - More focus on a trading model
 - Multiple Models for different combinations
 - Base on Users' risk level
 - Should we add Sentiment Analysis by NLP
 - Collaborative Filtering for User Classification (SVM, CF, etc...)
3. Visualization of fund results the Product and the Logic Flow
4. Email to Professor (Seek Advice and Share our Ideas)

Task Distribution:

1. FYP Report - Overview (Sun and Anthony)
2. FYP Report - Objective (Peter and Thomas)

Next Meeting: 1 Sept 2021 15:00-16:00

6.3 Minutes of the 3rd Meeting

Date: 1 Sept 2021

Time: 15:00 - 16:00

Venue: LG3-016

Participants:

KWOK, Ue Nam (20597580 unkwok)

MUN, Sun Bin (20455378 sbmun)

HO, Yat Man Peter (20606082 ympho)

KONG, Kin Cheung (20595166 kckongaa)

Meeting Discussion:

1. Confirmation on Meeting with Professor

Task Distribution:

1. Contact Professor for meeting
2. FYP Report - Literature Review (all of us)
3. FYP Report - Design
4. FYP Report - Implementation
5. FYP Report - Testing
6. FYP Report - Evaluation

Next Purposed Meeting: 6 Sept 2021 14:00-15:00 (with Prof. Rossiter)

6.4 Minutes of the 4th Meeting

Date: 6 Sept 2021

Time: 14:00 - 14:33

Venue: Room 3554

Participants:

Prof. Rossiter

KWOK, Ue Nam (20597580 unkwok)

MUN, Sun Bin (20455378 sbmun)

HO, Yat Man Peter (20606082 ympho)

KONG, Kin Cheung (20595166 kckongaa)

Meeting Discussion:

1. Introduction on our plan to Prof. Rossiter
2. How to evaluate the model performance
 - a. Using historical data to test it
 - b. Sharpe Ratio
 - c. Annual Income
 - d. CAGR
 - e. Buy then Sell now
 - f. Need to compare Inflation too (House Price)
3. VWAP - Not a strategy but an indicator
4. Model
 - a. Show the model performance for the user
 - b. Diversify on stock to reduce the risk
5. Dashboard
 - a. showing distribution on investment
6. Data
 - a. Finding reliable data source
 - b. Yahoo Finance (not recommended) / Google Finance
 - c. Major Index (HKEX)
7. Data Structure
 - a. Database to store the historical data (Save a copy)
 - b. How to store it
8. API - Real-Time data is usually expensive, but we may be able to get it through brokers
 - a. Look for the new system for free stock API
- 9. Budget (Keep Receipt !!!!!!!)**
 - a. Safer HKD1500
 - b. More dangerous HKD3000
10. Research Paper / Algorithm Recommendation
 - a. Trading Bot
 - b. Quora
 - c. Reddit
11. [Optional] Semantic Analysis (NLP)
12. Software Development

(COMP/DSCT)

- a. (PWA?) Up to us
- b. (VM?)

Task Distribution:

- 1. Literature Review - Aqumon (Thomas)
- 2. Literature Review - Research Paper on SVM, Random Forest, CNN, LSTM (Anthony)
- 3. FYP Report - Implementation - (Sun) Look for more algorithms and strategies (quant wise)
- 4. FYP Report - Testing (Thomas)
- 5. FYP Report - Evaluation (Anthony)

Next Purposed Meeting: 15 Sept 2021 3:00-3:30 (with Prof. Rossiter)

6.5 Minutes of the 5th Meeting

Date: 15 Sept 2021

Time: 15:00 - 15:42

Venue: Room 3554

Participants:

Prof. Rossiter

KWOK, Ue Nam (20597580 unkwok)

MUN, Sun Bin (20455378 sbmun)

HO, Yat Man Peter (20606082 ympho)

KONG, Kin Cheung (20595166 kckongaa)

Meeting Discussion:

1. Update on our progress
 - a. Progress on tasks from each member
2. Review on the draft of proposal report
 - a. Comment from professor
3. Suggestion on proposal report
 - a. Methodology graphic - Give more details and break down
 - b. Diagram - Should not be handwritten
 - c. Overview - Do not too specific and too long
 - d. Overview - Consistent format
4. Discussion on trading strategies
 - a. Averaging out (?)
 - b. Removing Outliers or not (?)

Task to completed:

1. Trading Strategy Research
2. More research on modeling

Next Purposed Meeting: 29 Sept 2021 12:00-12:30 (with Prof. Rossiter)

6.6 Minutes of the 6th Meeting

Date: 29 Oct 2021

Time: 11:50 - 12:32

Venue: Room 3554

Participants:

Prof. Rossiter

KWOK, Ue Nam (20597580 unkwok)

MUN, Sun Bin (20455378 sbmun)

HO, Yat Man Peter (20606082 ympho)

KONG, Kin Cheung (20595166 kckongaa)

Meeting Discussion:

1. Feedback of drafted monthly report from professor
2. Update on our progress
3. Feedback on real-time Data API
 - a. What common API from market
 - b. Which of them are not recommended
4. Trading frequency (?)
 - a. Based on “**personalized**” trading platform Backtesting process
 - b. What is the best way and best time to buy in
 - c. How to simulate the process
5. Web dashboard interface design
 - a. What our targeted users want to see
 - b. How to visualise
 - c. What data is needed
6. Risk assessment questionnaire
 - a. Samples risk assessment question from banks
7. Future timeline and plans

Next Purposed Meeting: To Be Determined

6.7 Minutes of the 7th Meeting

Date: 24 Feb 2022

Time: 17:30 - 18:00

Venue: Zoom

Participants:

Prof. Rossiter

KWOK, Ue Nam (20597580 unkwok)

MUN, Sun Bin (20455378 sbmun)

HO, Yat Man Peter (20606082 ympho)

KONG, Kin Cheung (20595166 kckongaa)

Meeting Discussion:

1. Feedback of drafted monthly report from professor
2. Update on our progress in winter semester
 - a. Model training and testing result
 - b. How to evaluate the performance (r^2 or MSE or MAE)
3. Implementation on backtesting
 - a. How to do that (?)
 - b. What metric to use (?)
4. Confirm the front end design and implement database design
 - a. Start design the front end layout
 - b. Implement simple database design on AWS
5. Risk level adjustment

Task Distribution:

1. Finalise all code for model training in 2 weeks
2. Brainstorm on risk factors that may affect the performance

Next Purposed Meeting: To Be Determined

6.8 Minutes of the 8th Meeting

Date: 6 Apr 2022

Time: 12:00 - 12:42

Venue: Zoom

Participants:

Prof. Rossiter

KWOK, Ue Nam (20597580 unkwok)

MUN, Sun Bin (20455378 sbmun)

HO, Yat Man Peter (20606082 ympho)

KONG, Kin Cheung (20595166 kckongaa)

Meeting Discussion:

1. Feedback of progress report from professor
2. Update on our progress
 - a. Merge the risk assessment questionnaire to the system
3. FYP Presentation guidelines
 - a. How to do it in a good way
 - b. What are expected to be shown in presentation

Next Purposed Meeting: To Be Determined

7. Appendix B Required Hardware and Software

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

7.1 Hardware

- 4 Development Laptops

7.2 Software

- Jupyter / Google Colab → For model training and testing
- Google Drive → For train data and test data storage
- Python → Programming language
- Virtual Machine (AWS) → For hosting the database and servers
- MySQL → For data and model storage
- ReactJS → For building the front end
- Chart.js → Library for visualizing the data
- Node.js + Express → Backend server
- Stooq → Historical Market Data Source

8. Appendix C: Project Planning

8.1 Distribution of Work

● Leader ○ Assistant/Team effort

Task	Peter	Anthony	Sun	Thomas
Project Designing				
3. Database Design	○	●	○	○
4. Model Selection	●	●	●	●
5. GUI Design	●	○	○	○
6. Risk Questionnaire design	○	○	●	○
Model Training & Testing				
7. Multipler Linear Regression	○	●	○	○
8. Simple DNN + TWAP/WAP	○	○	●	○
9. CNN-LSTM Regressor	○	●	○	○
10. CNN-LSTM Claassifier	○	●	○	○
11. Decision Tree	○	○	○	●
12. ARCH/GARCH Model - Time Series	○	○	●	○
13. ARIMA Model - Time Series	○	○	●	○
14. Support Vector Regressor	○	●	●	●
15. Support Vector Classifier	○	●	●	○
16. Gradient Boost (XGBoost)	○	●	●	○
17. Gradient Boost (CatBoost)	○	○	●	●
18. Facebook Prophet	○	○	●	○
Database Implementation				
19. Database set up on VM	○	○	○	●
20. Database Structure	●	○	○	○
API Data Extraction				
21. Apt Dataset Selection	○	●	●	●
22. Data preprocessing	○	●	●	●
GUI Implementation				
23. User portfolio	●	○	○	○
24. Pie chart distribution	●	○	○	○
25. Trend line analysis	●	○	○	○
26. Returned portfolio dashboard	●	○	○	○

FYP RO1 - Personalized Algo-trading Using Deep Learning & Machine Learning models

(COMP/DSCT)

Model Evaluation				
27. Backtesting - Model	○	●	●	●
28. Backtesting - User Portfolio	○	●	●	●
29. Further Evaluation	●	●	●	●
Other				
30. Video Trailer	●	●	●	●
31. FYP Presentation	●	●	●	●

Table 5. Task Distribution of our Project

8.2 Gantt Chart

Tasks	2021				2022				
	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May
Project Designing									
1. Database Design		■	■						
2. GUI Design	■								
Model Training & Testing									
3. Multipler Linear Regression				■	■	■			
4. Simple DNN + TWAP/VWAP				■	■	■			
5. CNN-LSTM Regressor				■	■	■			
6. CNN-LSTM Claassifier				■	■	■			
7. Decision Tree				■	■	■			
8. ARCH/GARCH Model - Time Series				■	■	■			
9. ARIMA Model - Time Series				■	■	■			
10. Support Vector Regressor				■	■	■			
11. Support Vector Classifier				■	■	■			
12. Gradient Boost (XGBoost)				■	■	■			
13. Gradient Boost (CatBoost))				■	■	■			
Database Implementation									
14. Database set up on VM					■	■	■		
15. Database Structure					■	■			
API Data Extraction									
16. Apt Dataset Selection		■	■	■					
17. Data API & Pipeline Connectivity							■	■	
GUI Implementation									
18. User portfolio							■	■	

FYP RO1 - Personalized Algo-trading Using Deep Learning & Machine Learning models

(COMP/DSCT)

19. Pie chart distribution										
20. Trend line analysis										
21. Returned portfolio dashboard										
Model Evaluation										
22. Backtesting - Model										
23. Backtesting - User										
24. Further Evaluation										
Other										
25. Video Trailer										
26. FYP Presentation										

Table 6. Gantt Chart for our Project

9. Appendix D : Example User Survey

▼ General Enquiry

▼ 1. What is your age range?

Show code

1. What is your age range?

age 55

```
##@title Default title text
```

Default title text

```
age = input_result.children[0].value
```

▼ 2. What is the portion of investment among your entire asset/securities/mortgage/loan?

Show code

What is the portion of investment among your entire asset/securities/mortgage

portion 50

```
portion = input_result.children[0].value
```

▼ 3. How much is your source of income?

Show code

3. How much is your source of income?

Choose one:

```
income_total = dropdown.value
```

▼ Risk Bearing Level

1. For the certain amount of investment you make, how much gain/loss are you expecting/comfortable with? (bar scroll) - max drawdown

Show code

```
1. For the certain amount of investment you make, how much gain/loss are you  
0-5%
```

Show code

2. What is the reason you started investing?

Show code

```
2. What is the reason you started investing?  
Reason: 1. Stock as a part of portfolio
```

```
why = x.value
```


3. What is the minimum timeframe that you are willing to invest?

Show code

```
3. What is the minimum timeframe that you are willing to invest?  
Term: Very short (1 month) Short Term (within ... Regular (6month~1... Quite Long (1~3
```

```
term = x.value
```

4. In a scale of 1~10, what is your acceptance in taking the risk for your investment?

In a scale of 1-10, what is your acceptance in taking the risk for your investment bearing  5

```
bearing = input_result.children[0].value
```

5. MCQ : Leverage, Individual Stocks, ETF, Fund

[Show code](#)

5. MCQ : Leverage, Individual Stocks, ETF, Fund

Types 

```
types = list(x.value)
```

6. How much profit are you expecting from your investment?

[Show code](#)

6. How much profit are you expecting from your investment?

profit_exp: 

```
profit_exp = x.value
```

Applicative Questions

1. How would you rate yourself as an investor? 1-10 scale

[Show code](#)

1. How would you rate yourself as an investor? 1-10 scale

conf  5

- ▼ 2. If the market crashes and your investment portfolio drops by more than 20%. What should you do?

[Show code](#)

```
2. If the market crashes and your investment portfolio drops by more than 20%.
```

```
Reason: 1. Sell everything and give up
```

```
res_1 = x.value
```

- ▼ 3. On the other hand, if the market strikes up to 15%, what would you do?

[Show code](#)

```
3. On the other hand, if the market strikes up to 15%, what would you do?
```

```
Reason: 1. Sell everything and gain profit
```

```
res_2 = x.value
```

- ▼ Agree/Disagree

[Show code](#)

(COMP/DSCT)

True or False : 1. I am looking for the long term / short term investment

- Long term
- Short term
- Don't know

tf1 = tf_1.value

tf2 = tf_2.value

tf3 = tf_3.value

tf4 = tf_4.value

True

Don't know

▼ Selection of stocks

Don't know

[Show code](#)

Select all the stocks you are interested in

Types	<ul style="list-style-type: none">FranchiseReal estateTelecommunicationEnergy & ResourcesCommodity
-------	--