Alan:

Hello everyone, thank you for coming to our presentation. I am Alan, and my group members are Gordon, Simon and Howard. Our advisor is Professor Dik-Lun Lee and our project is A system for predicting stock price and offering financial advice. Now, let's start our presentation.

[Click]Since the birth of the stock market, stock investment has become very popular for making profits. As considerable profits can be made through investments, people have tried to investigate and predict the stock market using different methods. [Click] Nevertheless, accurate prediction is unlikely to be achieved because many unknown factors are affecting the stock market. [Click]Hence, strategies such as risk management and portfolio management have been used to mitigate the risk of having inaccurate predictions of stock prices.

[Click]However, utilizing these strategies requires users to have advanced financial knowledge. [Click]For investors with no financial background, they usually have no idea on how to start their investment.[Click]

Therefore, this project is aimed to develop an automated system to [Click] collect stock and news data, [Click]predict the stock price using machine learning algorithms and [Click]provide a one-stop solution including risk evaluation, portfolio recommendation and data visualization.

Of course, our system cannot handle every stock market in the world, it requires so much computing power. We had to define our scope first.[Click]

[Click]

For the selection of the stock market, to find the most stable stock market for our current project goal, we have 3 main criteria which are the total market capital, number of stocks and diversity of stocks.

Therefore, we chose to start with US stock market. [Click]In 2020, total market capital in US stock market occupies 55% of total world equity market value. [Click]Also, there are 3,671 stocks in the US market, [Click]they can be separated into 11 sectors and further classified into 156 sub-industries. Therefore, working with the US market should be more comprehensive compared to using other stock markets.

[Click]

For the selection of the stocks, as mentioned above, there are over 3,500 stocks in the US stock market, it might not be necessary to analyze all of them, it requires way more computing power. [Click]

As a result, stocks in S&P500 were chosen for analysis and portfolio recommendation. S&P500 is one of the important indexes for estimating the overall performance of the US stock market. The stocks in S&P500 occupy around 80% of the total market capitalization. They are more suitable for our prediction. [Click] Because stocks with less market capital are

more easily manipulated by some holding companies and [Click] some less well-known companies may not have enough news for analysis. Therefore, it should be more efficient if our system focuses on S&P500 alone.

[Click]

Our advising system is not just visualizing the predictions of stocks but also recommend a user-tailored portfolio. It is difficult to handle different investment goals, not to mention it requires our users to know how to set their own investment goal. [Click] Therefore, we adopted a popular concept of setting investment goal – FIRE. FIRE means Financial Independence and Retired Early. Once you have earned 25 times of your annual expense, you make investment with your assets and take 4% of it to cover your expense each year. So we can now have an expected annual growth of our portfolio. [Click] For users not yet retired, their expected growth is the average market growth which is 13.6% according to the past performance of S&P500. [Click] For retired users, their expected growth is 4% + inflation.

[Click]

There are several components in our system. [Click]

Data crawlers for collecting stock and news data [Click] stock price preprocessing and new analysis using FinBert to output sentimental score of news [Click] Data models using Bidirectional LSTM to predict stock price [Click] Risk evaluation for risk evaluation of a stock [Click] Portfolio recommendation for generating user-tailored portfolios [Click] Web application for data visualization and user interactions. [Click] Mobile application for user to receive alerts. [Click]

Next, we will focus on 4 key components in our system. [Click]

Now let's start with news analysis.[Click]

Two different methods are used to collect historical and real-time stock news data. Google News is used to collect historical financial news using different computers and different IP addresses. For real-time stock news, we have used Finviz's stock page to crawl latest financial news daily. After crawling required news data, it is first stored as json files, and then imported to our MYSQL database. And now, we have collected 8 years of stock news starting from 2013.[Click]

After collecting stock news, we have performed data pre-processing. First, news without an exact posting date is filtered out because we need the exact date to combine news with our stock price data. Then, because Google searches the entire webpage for keywords including ads and hyperlinks, we have filtered out unrelated news collected from Google News that has few or even no relevant stock keywords. After that, we have generated sentiment scores from each piece of news using sentiment analysis.[Click]

In our project, FinBERT model is used to generate sentiment scores. FinBERT is the state-of-the-art analysis model that can be directly applied to our system. It is generated by training BERT model using financial domain specific texts, and it has been validated to

outperform dictionary-based analysis. Therefore, we have used it to generate sentiment scores range from -1 to 1 from the content of stock news.[Click]

Next, we will talk about our stock data and prediction model.[Click]

We have collected stock price data using yfinance API every trading day and store them as a Dataframe in csv format. As our goal focuses on long term investors, we have used daily data for this project instead of hourly or minute data. Currently, we have collected 10 years of daily historical stock price starting from 2011.[Click]

Similar to stock news, we have also performed data pre-processing for stock price. We have filled in missing values inside our dataset. Then, technical indicators are calculated and concatenated with the stock price data. These technical indicators provide domain specific knowledge to gain better understanding and they are commonly used by investors and financial analysts to determine the state of a stock. These indicators, such as SMA, EMA, MACD, RSI, can reflect the trend, momentum, volatility and volume of a stock.[Click]

After pre-processing is done, technical indicators, stock price data and our sentiment scores will be concatenated and input to our model. In total, we have 5 columns of stock price, 21 columns of technical indicators and 1 column of news sentiment as our model input.[Click]

Gordon:

Onto the stock price prediction models, first, we have defined our prediction goal as following:

Predict the upcoming 90 days of stock price based on a fixed window size.

We carried out experiments to find out the optimal machine learning architectures and configurations.[Click]

We have chosen MAPE, mean absolute percentage error for the loss function. MAPE is used because it can evaluate the percentage difference between the stock prices instead of the value difference. Compare with another common regression loss function RMSE, root mean square error, the RMSE value will be too small and difficult to evaluate whether the loss indicates good performance or not since we have applied normalization on the inputs. The respective accuracy is defined as 1- MAPE * 100%.

[click]

We also applied Minmaxscaler to normalize the input features.

One special note is that since this task is a regression task, we used linear activation for the output layer so as to not limit the output range of values.[Click]

Experiment 1

Onto the experiments.

For the first experiment, we find out the best machine learning architecture as well as the corresponding configuration of hyperparameters that gives the lowest validation loss. The testing architecture includes RNN, Uni-LSTM, Bi-LSTM and GRU. Grid search is used to systematically find out the optimal hyperparameters. We have tuned the number of hidden units, number of layers, dropout rate, learning rate and the window size.

[Click]

From the experiment, we found out that Bidirectional LSTM produces the best results with the lowest validation loss of 0.111. [Click]The optimal hyperparameters are shown in the table on the right.

[Click]

Experiment 2

Next, our second experiment is to test out the inputs to the machine learning models. We would like to verify whether the combination of stock price, technical indicators and sentiment score do really produce the best results. We compared different combinations of inputs using the best model that we get from the previous experiment. [Click] The results show that indeed inputting all of the data features produces the best results.

[Click]

The best inputs and models are then used to generate the final models. Around 470 models are trained. Using the test set to evaluate the performance of the models trained, we got an average accuracy of 86%, which shows that the model indeed generates acceptable performance that can be used in risk management.

Simon:

CE [click]

After the prediction, we incorporate some financial knowledge to make portfolio recommendations. [click] We first introduced the certainty equivalent concept to quantify people's attitudes or preferences over risk and return. There are three main variables in the equation which are ER expected return, A user acceptable risk, and RI stock risk index. [click] According to this equation, if the result shows higher CE, the stock is relatively profitable at the same level of risk. Later, I will discuss how expected return and stock risk index are defined and Howard will mention the user acceptable risk in the portfolio recommendation part.

Risk Evaluation [click]

Before doing portfolio recommendation, we would like to do risk evaluation first because it can help us to reduce the prediction error and users can refer to the risk index to make their own decision. [click] The risk is defined by using the stop points to control the risk in acceptable range and then quantify the risk by using some risk management technique.

Stop Points [click]

Two stop points are generated, including the upper stop and down stop. [click] The two stop points are commonly set up by using ATR average True Range because it can reflect the market fluctuation. [click] The upper stop is used to set an upper limit for selling points based on our forecast trend. Since we will predict the outcoming 90 days stock price, then we designed to get the minimum value between maximum price and minimum price within a certain fluctuation range.

[click][click] Besides the upper stop, we have the down stop to set a lower limit to exit the market based on historical price data. This point also set up a protection mechanism to limit the potential losses due to forecast errors. [click] It is defined as the max value between Chandelier stop and YoYo stop. Chandelier stop only reflect market fluctuation within a period, but YoYo stop reflect the immediate market change so these two are complementary. By combining them, we get both advantages of the two stops.

Stock Risk Index [click] [click]

After calculating the two stops, we use the model tested accuracy to estimate our expected return. [click] And then based on current price to calculate the expected return of the two stops. [click] Finally we will calculate the stock risk index by using those returns and model accuracy, as the result, it indicates the potential risk along with the expected return.

Screening [click]

After risk evaluation, we will perform screening first to filter unrecommended stock from prediction. It includes 3 cases: the first case is that the prediction trend is rising but the historical trend drops beyond our down stop which means it is out of the acceptable fluctuation range of historical data, so the price is volatile and the stock is not recommended.

[click] The second case is a bearish trend. Even if the price is still higher than our down stop, we will not recommend buying it because it's supposed to lose money.

[click] The final case is the trend appearing like a V shape through the Down stop. Even though the trend is rising later, we will not recommend buying it because it has the chance to drop below the down stop which will cause losses.

[click] On the other hand, there are two cases that will be recommended. Both cases have a rising trend and still within the expected fluctuation range.

Transition [click]

After performing our risk evaluation part, we will go into the portfolio recommendation part to make a more personalized recommendation.

Howard:

First, we wanna show you the flow of portfolio recommendations..

[click]

Portfolio recommendation requires 3 data input. User Financial info, stock risk index and stock expected return.

User Acceptable Risk index is calculated from the User financial info. The Certainty Equivalent is then calculated based on these data. After that, stocks are identified as either aggressive or defensive using Certainty Equivalent. An aggressive portfolio for maximizing profit or a defensive portfolio for minimizing risk is eventually generated.

Now, we will look into each component in detail.

[click]

The following data are collected in the web application

[click]

Based on the information collected, we can calculate the user acceptable risk index with this formula. The user acceptable risk index is depending on the progress of the user on achieving his or her investment goal. You can see the later part of the formula. [click]

The current asset is the net liquidation value of equity minus the debt amount or zero. For the investment goal, by the definition of FIRE, it should be 25 times of the user's annual expense.

[click]

Now, we have all the input needed for calculating Certainty Equivalent.

[click]

This is the formula of calculating Certainty Equivalent.

It is similar to the risk premium of a stock to a specified user .

[click]

With a higher CE, the stock is relatively profitable at the same level of risk for the user.

[click]

Now, we can classify the stocks into aggressive stocks and defensive stocks

Aggressive stock is mainly for the user not yet retired and defensive stock is for the user retired.

Aggressive stock is aimed to gain maximum profit for users to achieve FIRE quickly. Its expected annual CE should align with the general performance of the stock market. As our system is using stocks in S&P 500, the expected CE should be the average growth S&P 500 in the past 10 years, which is 13.6%.

On the other hand, defensive stock is aimed to cover the annual expense of the retired users, it doesn't need to earn a remarkable profit but it must be very stable. By the 4% rule in FIRE, the portfolio growth should be 4% plus the inflation rate for covering user's expenses. The average inflation rate in US in the last 10 years is 1.4% so the expected annual CE is 5.4%

[click]

But there are still too many stocks we can recommend to users, we don't need all of them. Therefore, we only take 10 aggressive stocks with the highest expected return into the Aggressive stock pool and 10 defensive stocks with the lowest risk index to Defensive stock pool.

[click]

Here is the distribution of the 2 types of stock.

As you can see, there are some defensive stocks having a relatively high risk index.

[click]

To avoid higher-risk stocks being recommended in the defensive stock pool, the stocks with risk index ≥ 0.043 are excluded, where this value is the intersection of the distributions of aggressive stocks and defensive stocks.

[click]

To summarize the definition of aggressive stock and defensive stock, here is a matrix showing the conditions of whether the stock is in the aggressive stock pool or in the defensive stock pool or not recommended.

Now, we can start generating portfolios for users.

[click]

For the users not yet retired, they should use an aggressive portfolio. This user just starts his or her investment, he or she can accept more risk in order to have a remarkable profit. So this portfolio has only 5 aggressive stocks and each stock occupies 20% of the portfolio.

When users have more asset in their account, their portfolio should be more risk-averse to prevent total loss

[click]

For this user, he or she almost achieves FIRE. Therefore, even the portfolio is still aggressive, but it has 5 defensive stocks for balancing the risk. The ratio of aggressive stocks and defensive stocks is depending on the progress of the FIRE. With a higher total asset amount like 800,000 dollar, the proportion of defensive stocks will be higher and vice versa.

[click]

After the users achieve FIRE, they should use a defensive portfolio which consists of only 10 defensive stocks.

These two types of portfolios are experimented to estimate their respective performance.

[click]

We only tested the portfolio generated from June 2020 to November 2020. Since our maximum prediction period is 90 days ahead, the stocks purchased in November 2020 are sold in February 2021, this indicates that our experiment has elapsed from June 2020 to February 2021, a total of 9 months.

The cumulative profit of aggressive Portfolio in 9 months is 9.18%, which is smaller than the expected growth 10.2%. Also, as you can see, the profit is mainly depending on the outstanding profit in November. If we exclude this month, its growth is -4.39%. Also, it was running deficits in 2 months so the performance is not really stable.

[click]

On the other hand, a defensive portfolio has a way better performance. Its cumulative profit is 9.02% which is similar to that of an aggressive portfolio and way beyond the expected growth. Even though we exclude the remarkable profit in November, it still had a 4.52% gain. It is much more stable.

[click]

Both portfolios returned profit in our test set. The annualized return for the portfolios is therefore 12.24% and 12.03% respectively. [click] The defensive portfolio had a great performance [click] but the aggressive portfolio could not meet our expectation, its return is just similar to the defensive portfolio. [click] Also, the maximum loss is only 0.8% in defensive portfolio comparing to the 5.2% in aggressive portfolio, it shows the defensive portfolio successfully avoided great loss.

[click]

Now, we will start the demonstration of our web application and mobile application.

Gordon:

[click]

Now, let's take a look at the mobile application. The main functionality is to let users receive latest news information. Users can login to their account with the same credentials from the web application. They will see a list of news that is related to the stocks that they have purchased. The sentiment scores of the news are labeled in either green, red or grey, indicating whether a piece of news is positive, negative or neutral. Users can tap on the list tile to open up the news in the website.

[click]

Users can also receive news notifications when our system crawls a piece of news articles related to a stock that the user owns. This allows the users to catch up with the latest development of the stocks.

[click]

To conclude our system as a whole. First, we have automated systems to regularly collect stock price and news data. Next, we have trained machine learning models to predict the upcoming stock price with reasonable accuracy. After that, the results are used in risk management to identify the profitable stocks. Then portfolio management is carried out to recommend users a suitable portfolio to help them reaching FIRE. Our portfolio back testing shows a good amount of return following our portfolio recommendations. Last but not least, we provided a user friendly web application to let users manage their portfolios as well as a mobile application to receive latest news.

[click]

Finally, we would like to talk about some future work for this project. Although accurate predictions for stock price can never be achieved, we still believe that by inputting more types of data into the model, we can get better results. For example, we can input political news or sector news to find deeper relationships. Besides, our current system only generates sentiment score based on a piece of news article. This limits the amount of data that we get from the news. We believe event extraction can be used to extract the main content and then directly apply the embedding to the machine learning model to produce a better result. One practical improvement for our system is to add more functionalities to the web and mobile application such as implementing the portfolio management into the mobile application.

[click]

That’s the end of our presentation and we are open for questions.