The Hong Kong Polytechnic University

Department of Applied Mathematics

AMA4951 Capstone Project

Artificial Intelligence applications with time series data in stock market

By **[WONG CHUN WAH]**

[20017388D]

Bachelor of Science (Honors) in Data Science and Analytics

([Supervisor: Dr. James Huang])

Abstract

"A prediction about the direction of the stock market tells you nothing about where stocks are headed, but a whole lot about the person doing the predicting.", quoted from Warren Buffett, a successful American investor. Stock market prediction has always been the importance of concern with the huge potential revenue. Identifying the stock buying and selling strategy is the major tasks using a series of data analytics techniques. Thanks for the mature pace of technologies, the AI algorithms brought the new insightful uses in stock market. This research paper reveals the processes of AI algorithms and its applications in stock markets. Under the algorithm frameworks, there could be numerous methods to build up the machine learning and deep learning algorithms. To evaluate the performance, the model analysis would also have been a key area for the investing purposes. The machine learning algorithms deals with the stock price up and down classification problem whilst the deep learning algorithms handles the more complex part which is the close price movement regression problem. The aim of this project is to provide the valuable investing information after a series of the data analysis in AI aspect. The limitations of algorithms and future works would be included and the key findings in this project would illustrate in the last part. This expected to assist the investors to formulate their investment strategy .

Content

1.Introduction

As the rapid development of artificial intelligence, AI applications in many industries is everywhere nowadays and it can produce huge amounts of values to the company but also in financial market. In this capstone project, the aim is to apply the updated machine learning and deep learning techniques to create some discoveries in the financial market and the research target would be the US stock market. This research project focuses on the US stock performance in the year of 2021 and the python programming would be also included to conduct the stock research. Also, some related theories in different areas such as mathematics, statistics, information theory, would also be considered to explain the trends of stock market. In the first part, there would have some background information of AI including the history, developments and applications. Followed by the introduction of some machine learning and deep learning techniques and those applications demonstrations in the stock market. The AI applications consists of two main parts which are classification and regression problems. For the sake of complexity and fluctuation in US stock market, the machine learning algorithms would be applied into classification part while the deep neural networks algorithms would be applied into regression part. Specifically, this project involves the time series stock forecasting using the previous time series stock value. Lastly, for the conclusion part, the focus point would be the elaboration of some insightful result discovered from all the AI algorithms performance.

2.Definition of Artificial Intelligence

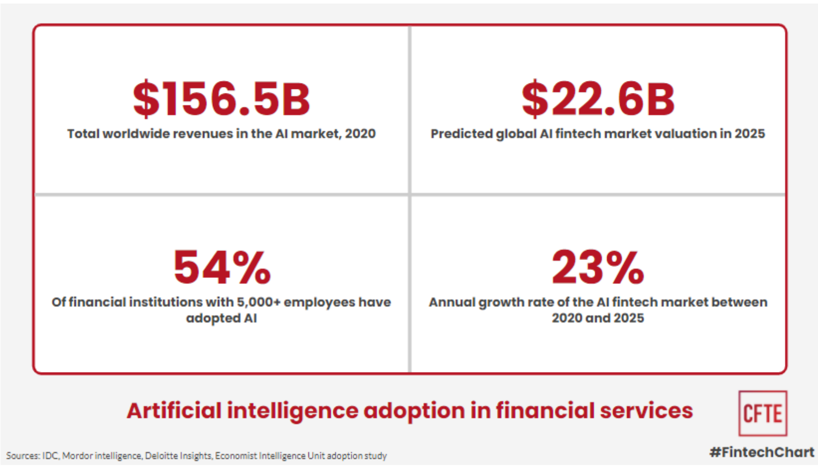
Artificial Intelligence (AI) is created by humanity and its structure try to mimic the human or animals thinking logical system to demonstrate the problem-solving skills to assist human to solve the different kinds of problem in real life. In general, the computer programs are every part of artificial intelligence that generates the command for AI to make the decision in every steps. In order to get the optimal act in different situation, the AI need to learn from the past according to some techniques. The techniques or knowledge can come with many interdisciplinary but mainly are computer science, statistics and mathematics. The AI applications could have four mainstream parts and those are perception, cognition, creativity and wisdom. The perception of AI means try to experience the surrounding environment under imitating the human perception in order to do the same action with human. The common techniques of AI perception are computer vision, face recognition, sound recognition. For the cognition, the task of AI is to learn, judge and analyze the information by using the past information and then draw the corresponding conclusion. Some examples are email spam identification, medical image analysis. The third part is the creativity which is focusing on the art creation. The creations include some songs, novels, paintings. The last aspect of AI is wisdom. It is the most controversial part that is how likely AI can have their mind acting like a human. It involves some self-values and the consciousness of AI.

3.History of AI

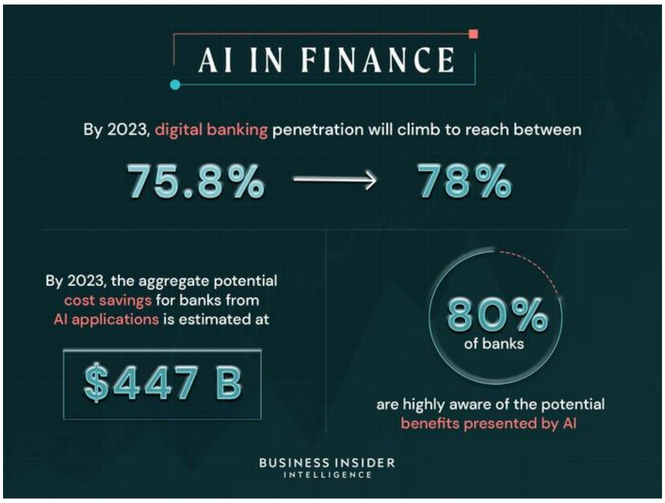
The early AI development of history can date track to the 20 century of 1930 to 1950 year. During that time period, the huge discovery of neuroscience had shown that the human brain system is the electronic network. It consists of numerous of neuron. The level of neuron only exists two states that is all or nothing. This similar to the computer operating signal , 1 or 0. In 1951, the very first AI program that is a checkers programs were written by the Christopher Strachey, British computer scientist. Also, the chess program written by the Dietrich Prinz, the computer scientist as well. The power of the programs could be challenging the local chess players in the middle of 1950 to 1960 and the influence of AI was starting to be concerned. Hence, the game AI is always a standard for evaluating the development of AI. Another famous event of AI development is the Turing test. The Turing test is an experiment thought that developed by the famous British computer scientist, Alan Turing. The main purpose of Turing test is to prove whether the computer which is an AI can demonstrate the human acting and it is not distinguishable by normal human. The testing experiment is simple that is requiring one person to have some conversations to the machines with using some electronic devices. The person would not be informed that the communication objects at the beginning. If the person finally cannot figure out the identity of target, then the AI machine may probably own its intelligence. This experiment thought can inspire people to think more about the existence of AI mind. Beyond the Turing test, the development of AI had stuck in some periods. These are called “AI winter”. The main possible reasons are the lack of capital support and some experts in development.

4.The importance of AI applications in financial market

As the big data era has come, the amount of data generates rapidly in every industry. In financial industry, the AI applications can be various due to many kinds of scalable algorithms. It brings the high level of data analysis onto the financial data and facilitate the business to have an optimal decision-making and doing the risk management. To the AI application of stock market, there are numerous powerful AI techniques and one of them could be the AI automation algorithmic trading. Normally This uses the complex AI program to handle the transactions automatically after analyzed by a bunch of AI algorithms and also depends on the person of investment risk attitude. The decision of investment is hence favorable to the investors. More importantly, according to some lately research, International Data Cooperation (2020) revealed that the total worldwide revenues in the AI market in 2020 can attain to 156.5 billion of dollars. Furthermore, the annual growth rate of AI fintech market between 2020 and 2025 is 23%. It is obvious to see the AI applications in financial market is becoming more general. As a result, the data disclose how AI applications important to the financial market.

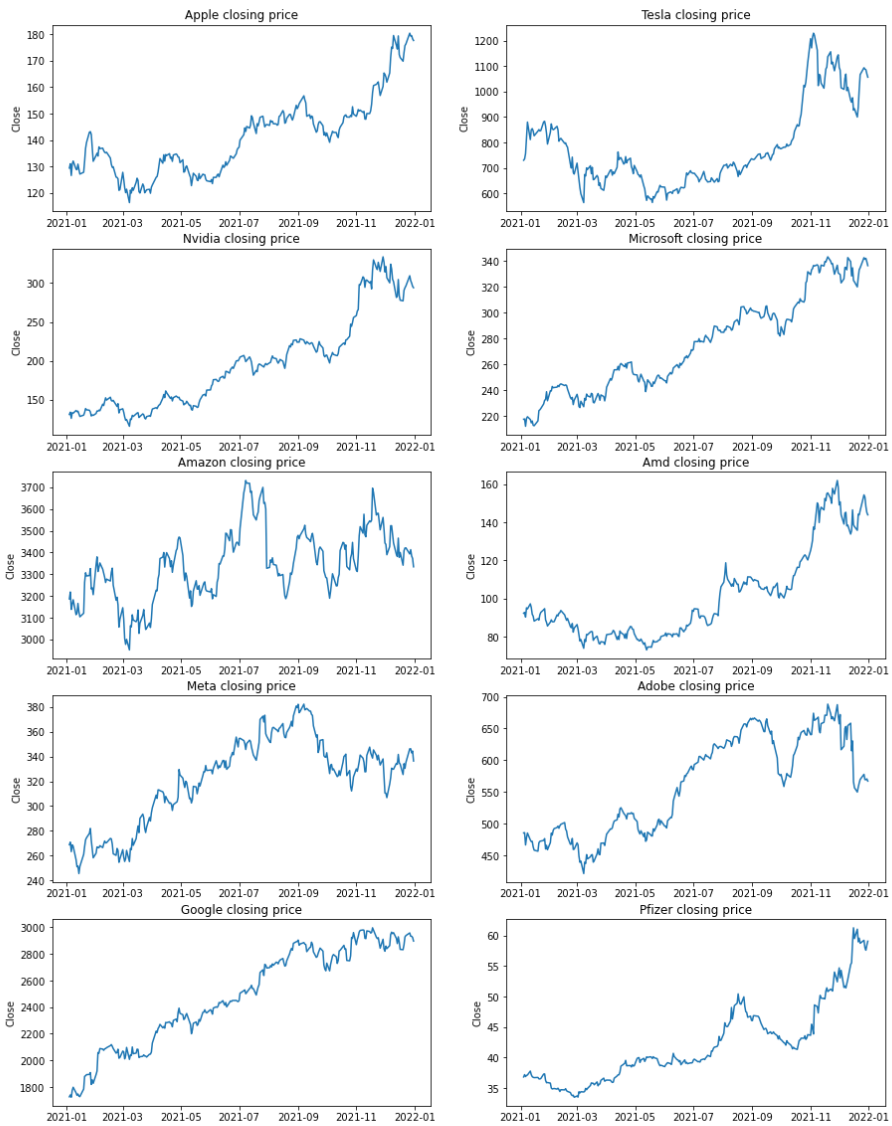


Beyond the AI application in stock market, another one is the AI application in banking industry. As more and more different financial managements shows up, the bank cooperation needs to deal with the large client groups and consider providing the high quality of financial services to the bank customers. One of the benefits in AI banking is to acknowledge the customer financial decision-making in deep. With the different financial consuming pattern from the client's group, using the AI banking can help client to effectively formulate the personal financial banking strategy according to the economical circumstance. The client may consider the optimal financial products provided by the AI. This is a win-win case that is not only helping the client manage the wealth but also saving the bank cooperation cost in human resources. Moreover, Phaneuf (2022) indicates that the digital banking penetration will reach to at most 78% before 2023 from 2022. The total potential amounts of cost savings in bank industry can be attained at $447 Billion. More importantly, nowadays 80 out of 100 banks in the world are looking for the benefits brought from AI transformation. The results show that the AI applications would dominate the traditional banking structure.



5.The dependency measurement of stocks

In this research project, the 10 active US stock in 2021 would be chosen to the stock research targets and they are "Apple", "Tesla", "NVIDIA", "Microsoft", "Amazon.com", "AMD", "Meta Platforms", "Adobe", "Google" and "Pfizer". Before implementing the AI algorithms in the stock research, it is important to know that the movement relationships between all the stocks. The historical stock price information is collected from yahoo finance API in python.



The stock data is collected from Yahoo finance using the API from python. To investigate the daily price movement, The following formula would be defined as “Open” price minus “Close” price on that day. That is:

$$Price\ movement\ of\ the\ day\ =\ Opening\ price\ of\ the\ day - \ Closing\ price\ of\ the\ day$$

taking the Apple stock as an example, the following column shown below:

```
apple
```

Date	High	Low	Open	Close	Volume	Adj Close	Movement
2021-01-04	133.610001	126.760002	133.520004	129.410004	143301900.0	128.617111	4.110001
2021-01-05	131.740005	128.429993	128.889999	131.009995	97664900.0	130.207291	-2.119995
2021-01-06	131.050003	126.379997	127.720001	126.599998	155088000.0	125.824326	1.120003
2021-01-07	131.630005	127.860001	128.360001	130.919998	109578200.0	130.117844	-2.559998
2021-01-08	132.630005	130.229996	132.429993	132.050003	105158200.0	131.240906	0.379990
...	...	...	...	...	...	...	...
2021-11-24	162.139999	159.639999	160.750000	161.940002	69463600.0	161.940002	-1.190002
2021-11-26	160.449997	156.360001	159.570007	156.809998	76959800.0	156.809998	2.760010
2021-11-29	161.190002	158.789993	159.369995	160.240005	88748200.0	160.240005	-0.870010
2021-11-30	165.520004	159.919998	159.990005	165.300003	174048100.0	165.300003	-5.309998
2021-12-01	170.300003	164.529999	167.479996	164.770004	152052500.0	164.770004	2.709991

If the movement value is positive which is larger than 0, then it means the stock price generally increase and the performance is good on that day, otherwise, the movement would become negative and that means the stock is generally decrease on that day.

5.1 Covariance measurement on stock analysis:

The covariance measurement can be using to describe the joint variability of random variables. It helps to reveal the linear relationship in each stock's movement as the stock movement values can be treated as the random variables. If one random variable holds a large value corresponding to another random variable that also holds a large value, the covariance hence becomes large and positive. However, if one is increasing to a larger value while another one tends to become smaller, then the covariance would become small and negative. It may show the inverse linear relationship. Otherwise, the covariance becomes tend to zero which indicates there is very small degree of linear relationship and the relations of random variables of joint distribution could be concluded as independent. The covariance formula with using the expectation theory describes as below:

$$\operatorname{cov}(X, Y) = \operatorname{E}\left[(X - \operatorname{E}[X])(Y - \operatorname{E}[Y])\right]$$

The covariance matrix can be applied into each close price of stocks. Let's see the covariance matrix of each stock:

```
movement.cov()
```

	apple	tesla	nvidia	microsoft	amazon	amd	meta	adobe	google	pfizer
apple	1.331814	4.136085	1.449581	1.517602	21.479877	0.729056	1.766328	3.383897	11.569184	0.090517
tesla	4.136085	127.755961	12.485306	6.035588	92.271361	6.842475	9.656622	17.929791	39.826794	-0.226100
nvidia	1.449581	12.485306	6.514265	2.666366	39.227678	2.512475	3.466611	7.321003	21.719163	-0.002107
microsoft	1.517602	6.035588	2.666366	4.004635	40.199911	1.305883	3.550013	7.866229	25.427983	0.203110
amazon	21.479877	92.271361	39.227678	40.199911	1008.453749	19.273725	59.213991	107.442265	369.728406	1.479876
amd	0.729056	6.842475	2.512475	1.305883	19.273725	2.017083	1.739149	3.532864	9.628461	0.008249
meta	1.766328	9.656622	3.466611	3.550013	59.213991	1.739149	10.021159	9.336731	36.245576	0.134963
adobe	3.383897	17.929791	7.321003	7.866229	107.442265	3.532864	9.336731	28.166336	57.176299	0.312263
google	11.569184	39.826794	21.719163	25.427983	369.728406	9.628461	36.245576	57.176299	315.999439	1.351994
pfizer	0.090517	-0.226100	-0.002107	0.203110	1.479876	0.008249	0.134963	0.312263	1.351994	0.205955

The highest covariance is the pairs of Google stock and Amazon stock while the lowest covariance is the pairs of Tesla stock and Pfizer stock. As each stocks price may have different price scales, it is reasonable to see that some of the covariance value is relatively small or large. Nevertheless, the covariance concepts may reveal some certain referential value about the movement relationship.

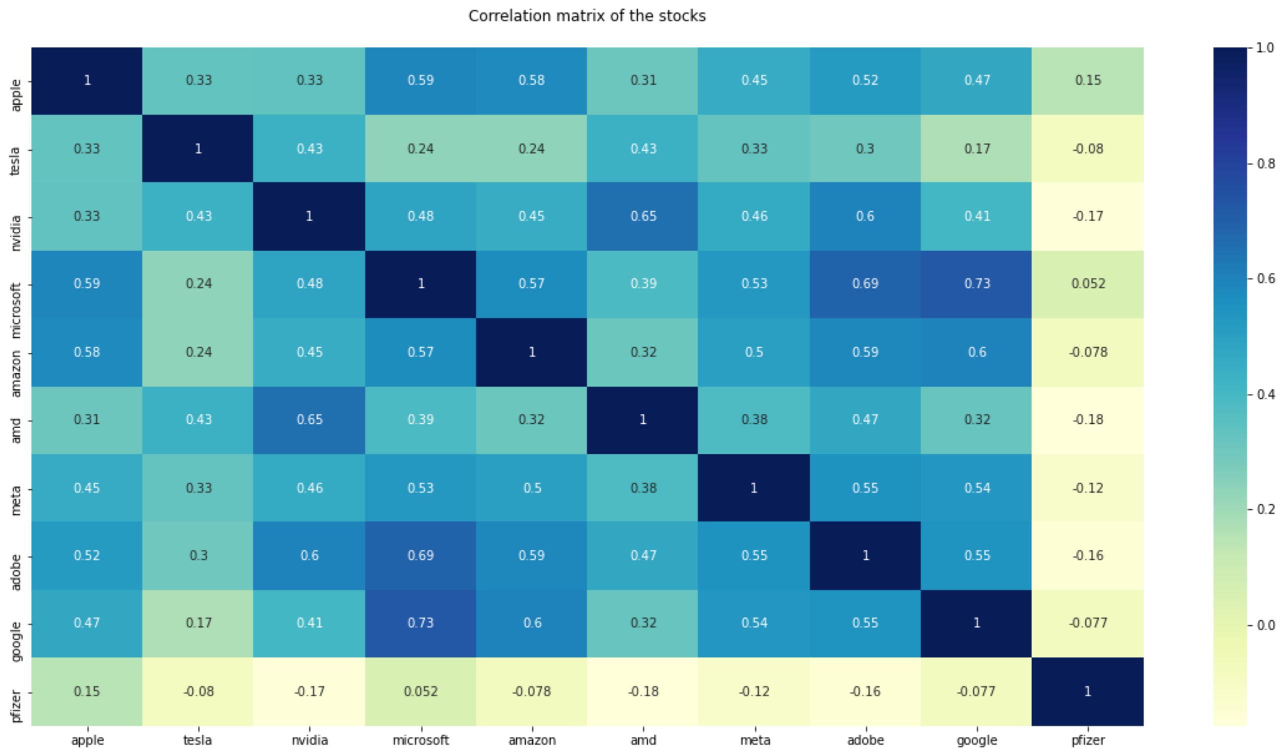
5.2 Correlation measurements on stock analysis:

As the concept of covariance is introduced and to further measure the relationship between the stocks, the correlation is one of the important measurements to evaluate the linear relationship for all the pairs of stock. This is the statistical measurement to describe the degree of linearity for those two variables coordination and strictly bounded between -1 and 1. It can also be the scaled version of covariance measurement. In order to see the linearity in each of the stocks , the correlations matrix would be used for assessing the linearity. The formula shown below:

$$\operatorname{corr}(\mathbf{X}) = \begin{bmatrix} 1 & \frac{\mathrm{E}[(X_1-\mu_1)(X_2-\mu_2)]}{\sigma(X_1)\sigma(X_2)} & \cdots & \frac{\mathrm{E}[(X_1-\mu_1)(X_n-\mu_n)]}{\sigma(X_1)\sigma(X_n)} \\ \frac{\mathrm{E}[(X_2-\mu_2)(X_1-\mu_1)]}{\sigma(X_2)\sigma(X_1)} & 1 & \cdots & \frac{\mathrm{E}[(X_2-\mu_2)(X_n-\mu_n)]}{\sigma(X_2)\sigma(X_n)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\mathrm{E}[(X_n-\mu_n)(X_1-\mu_1)]}{\sigma(X_n)\sigma(X_1)} & \frac{\mathrm{E}[(X_n-\mu_n)(X_2-\mu_2)]}{\sigma(X_n)\sigma(X_2)} & \cdots & 1 \end{bmatrix}.$$

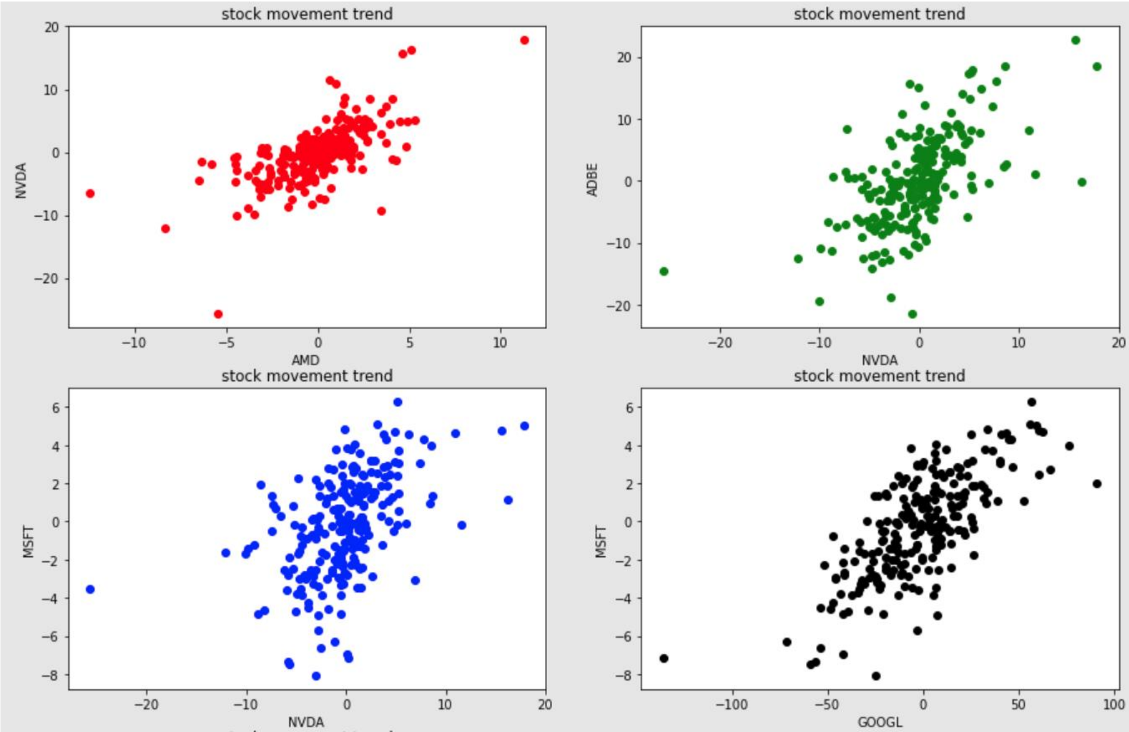
This is an n-by-n matrix and the X random variable according to the column and row position to calculate the mean and sigma. The diagonal element must be 1 as this is the situation for comparing the same random variable.

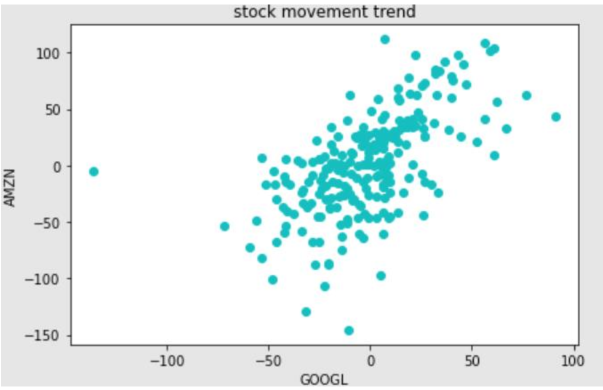
After introducing the concept of correlation, let's see the correlation matrix below:



This correlation matrix shows every pair of stocks in movement of correlation corresponding to the row and column. The correlation stay between -1 to 1 and it describes the degrees of linear relationships. There are three main referential cases. The first case is when the correlation is close to 1, that means they have huge strong positive linear relationship. For the second case, if the correlation is close to -1, this shows the negative linear relationship that is when one stock goes up rapidly, another one of stock will dive at the same time. The last case is the correlation of closing to 0, this can be the bad case as there would display almost no linear relationship for any linearity reference.

As the matrix shown, there are some slight strong correlation relationships in some pairs. To select those highest correlation stocks, the threshold of correlation value is going to set to be 0.6 and there would have 5 pairs of stock meet the threshold requirements. Those following scatter graphs would become:

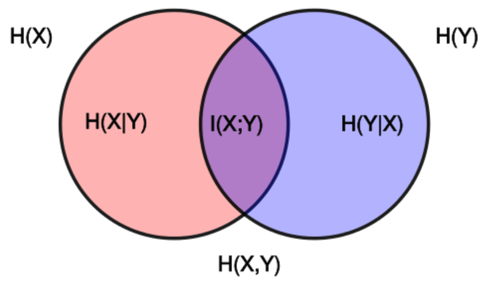




As a result, these 5 scatter graphs shows the linear relationship according to the following stock movements.

5.3 Mutual Information measurement on stock analysis:

The Mutual Information measurement is another significant relationship measurement to discover the mutual dependency of random variables. The mutual information technique was developed in information theory. It is a type of information gain and quantifies the information amounts under the entropy-based between the variables. This metric helps to show how degree of influence of one variable affects to another variable. Some complex relations of the variables are adequate to be measured by mutual information, especially in nonlinear relation. The idea of mutual information can be depicted with a Venn diagram:



The $H(X)$ and $H(Y)$ are individual entropy measurement of the random variable X and Y respectively. Furthermore, the conditional entropy and joint entropy can be clearly seen and that is joint entropy equals to the amount of information. So, the formula becomes:.

$$\mathrm{I}(X;Y) = \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X,Y)$$

Using the mutual information concepts, the mutual information of joint discrete random variables of X and Y can be calculated under the double summations:

$$\mathrm{I}(X;Y) = \sum_{y\in\mathcal{Y}} \sum_{x\in\mathcal{X}} p_{(X,Y)}(x,y) \log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)\,p_Y(y)}\right)$$

For the joint continuous distribution with random variables of X and Y, the formula would be written with calculating the double integrals:

$$\mathrm{I}(X;Y) = \int_{\mathcal{Y}} \int_{\mathcal{X}} p_{(X,Y)}(x,y) \log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)\,p_Y(y)}\right) dx\,dy$$

Where $P_{(X,Y)}(x,y)$ is the joint distribution and $P_X(x)$ and $P_Y(y)$ are the marginal distributions.

To apply the mutual information techniques in stock market analysis, the time features would usually be matters to the stock. In view of this hypothesis, the time value can be specified according to each trading record of calendar datetime value such as Weekdays, Months and Days. For example, taking the apple stock again, the dataset would be:

```
apple[apple.index > '2020-12-31']
```

Date	High	Low	Open	Close	Volume	Adj Close	Price Movement	Range	Price Trend	Weekday	Month	Day
2021-01-04	133.610001	126.760002	133.520004	129.410004	143301900.0	128.453445	4.110001	6.849998	Up	Monday	1	4
2021-01-05	131.740005	128.429993	128.889999	131.009995	97664900.0	130.041626	-2.119995	3.310013	Down	Tuesday	1	5
2021-01-06	131.050003	126.379997	127.720001	126.599998	155088000.0	125.664207	1.120003	4.670006	Up	Wednesday	1	6
2021-01-07	131.630005	127.860001	128.360001	130.919998	109578200.0	129.952271	-2.559998	3.770004	Down	Thursday	1	7
2021-01-08	132.630005	130.229996	132.429993	132.050003	105158200.0	131.073929	0.379990	2.400009	Up	Friday	1	8
...	...	...	...	...	...	...	...	...	...	...	...	...
2021-12-27	180.419998	177.070007	177.089996	180.330002	74919600.0	180.100540	-3.240005	3.349991	Down	Monday	12	27
2021-12-28	181.330002	178.529999	180.160004	179.289993	79144300.0	179.061859	0.870010	2.800003	Up	Tuesday	12	28
2021-12-29	180.630005	178.139999	179.330002	179.380005	62348900.0	179.151749	-0.050003	2.490005	Down	Wednesday	12	29
2021-12-30	180.570007	178.089996	179.470001	178.199997	59773000.0	177.973251	1.270004	2.480011	Up	Thursday	12	30
2021-12-31	179.229996	177.259995	178.089996	177.570007	64062300.0	177.344055	0.519989	1.970001	Up	Friday	12	31

As the last couples of columns shown, the time value includes Weekday, Month and Day would be our time features to understand the time relations of stock. Since this research focuses on the 2021 stock market trend analysis, the Year feature is not necessary.

Suppose now the mutual information calculation would be used in close price, the Random variable X includes the time value which is Weekday, Month and Day. The random variable Y value would be the following stock of close price. After the calculation, the mutual information table is shown below:

	Weekday	Month	Day
apple	0.000095	0.965595	0.079471
tesla	0.000000	0.639358	0.052835
nvidia	0.000000	1.117689	0.141599
microsoft	0.000000	1.261409	0.167011
amazon	0.046398	0.384706	0.044541
amd	0.000000	0.965752	0.073294
meta	0.000000	0.829581	0.035101
adobe	0.000000	0.861916	0.037816
google	0.000000	1.189124	0.151785
pfizer	0.000000	1.026187	0.217924

Based on the result, the Month feature generally holds a high mutual information value to other features. The Weekday feature is always 0 except amazon and apple stock. It can conclude that the Month feature affect the close price much while the Weekday feature has less influence on the close price.

Beyond the close price, the trading volume is also important as it indicates the popularity of stock trading. Similarly , the Random variable X continues to include the time value which is Weekday, Month and Day. The random variable Y value would be the following stock of volume amounts. After the calculation, the mutual information table is shown below:

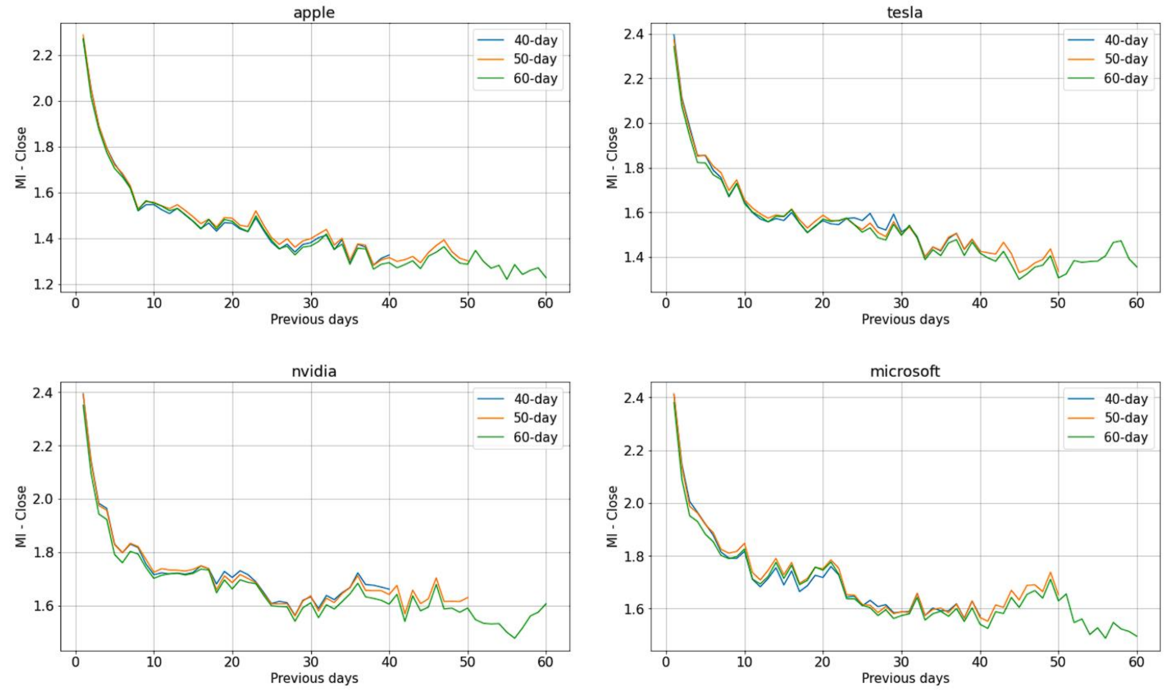
	Weekday	Month	Day
apple	0.000000	0.164855	0.000000
tesla	0.000000	0.159320	0.000000
nvidia	0.012704	0.139091	0.000000
microsoft	0.008176	0.070548	0.000000
amazon	0.000000	0.016255	0.000000
amd	0.017336	0.103831	0.026888
meta	0.000000	0.160712	0.012465
adobe	0.025250	0.120711	0.000000
google	0.036192	0.101374	0.004946
pfizer	0.022210	0.206882	0.000000

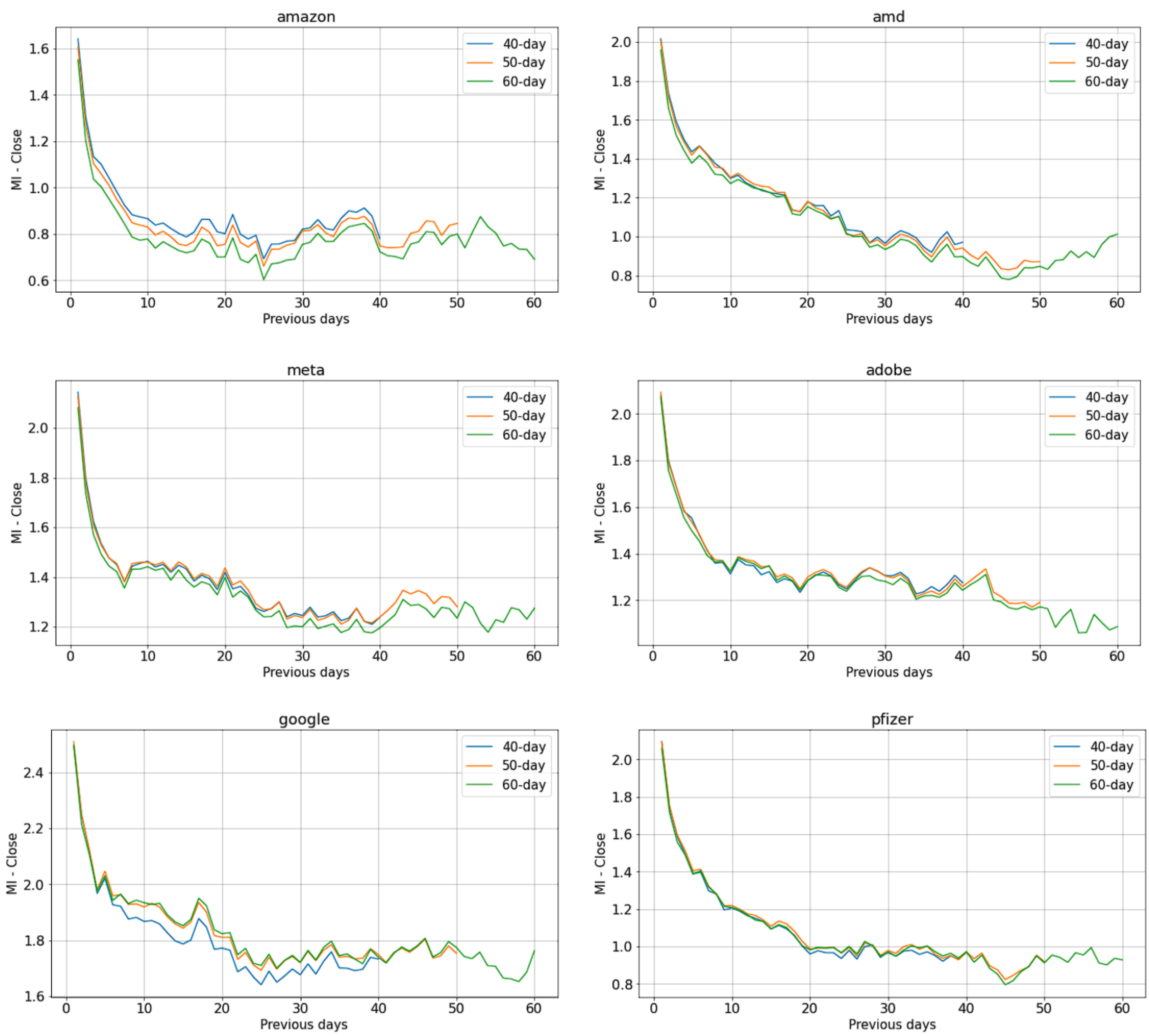
Based on the result, the overall mutual information performance is not satisfactory in 2021 because the value is relatively low. However, the Month feature is again holding the highest weight between all the features. It can conclude that the trading volume amount may not highly affected by the time of 2021.

5.4 Mutual Information in time series data:

Through applying the mutual information techniques into the calendar time features, it is more important to apply it into the time series data of each stock. The historical data in different time may somehow affect the tomorrow stock trading. With using the mutual information techniques to measure the time relations of the stock data, the time series historical data would become the random variable X and the closest stock day of data would become the random variable Y. Moreover, the scale of days need to be chosen in order to see all the time relations of previous stock trading days. Suppose now the scale of days is set to be 40, then let the previous 40-day of stock value become the random variable X and that would be $\{ X_t : t = 1, 2, \ldots, 40\}$. The random variable Y of $\{Y_t\}$ is merely the next day of stock value before the previous 40-day stock value. As the previous table shown, the mutual information technique on stock volume doesn't provide insightful differences. The close price on mutual information graph only being considered and it would again become our stock target value for assessing the time series relationship.

The mutual information trend of close price can be described as a line chart in order to clearly see the time series dependency in different daytime scales. Here, the testing daytime includes 40-day, 50-day and 60-day and the line chart would be:





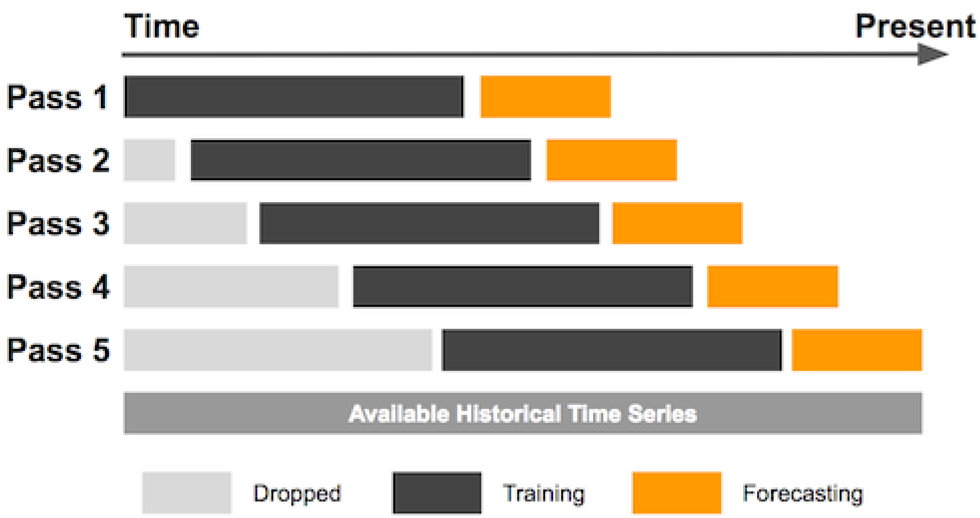
According to the line chart mutual information trend, generally all daytime scales appear the similar movement pattern. It is common to see the first previous 10-day carry a large mutual information value as those days are the closest day to the next day. In addition, most of the mutual information value of 60-day time series data is the lowest after the previous 40-day including Tesla, Nvidia, Amazon, Amd, Meta, Adobe. Furthermore, as for the highest 40-day value comparing other day value, amazon stock is the sole.

6. Fundamental machine learning background

In artificial intelligence, machine learning is one of the powerful techniques that enables the AI automates the models by learning from the historical pattern. The learning algorithms built with the past data and mainly consists of the statistical and mathematical theories. With using such theories, it can help machine learning algorithms finding out the optimal outcome under understanding the data environment. Nowadays, machine learning techniques are very popular that always applied into variety of fields such as medical diagnosis, insurance, financial market. Before obtaining the best solution by machine learning algorithms, one of the important tasks is to have a problem-defining procedure according to the scenario. There are mainly two problems which is regression and classification. The first problem means predicting the quantitative target value such as stock price, house rents, while the classification problem is to forecast the qualitative target value such as heart disease diagnosis, insurance fraudulent. In view of the broad application in machine learning, meanwhile it also divides to two types of algorithms and those are supervised learning and unsupervised learning. The algorithms in supervised learning can learn and test from the dataset automatically. The result can be evaluated after the training, and it is easy to do performance scaling. For the unsupervised learning algorithms, the similarity need to learn by itself and there is no assessment for them, but the manpower can be largely saved. So generally, the performance of supervised learning algorithms are favorable.

6.1 Machine learning algorithms on stock classification problem

The machine learning algorithms can be applied into our stock market project. The classification problem would be the stock trend prediction with a class label which is up or down. Similar to the mutual information sections, the objective of classification problem is to obtain the next day of performance using the previous days of time series data. This is called time series rolling window forecasting. To illustrate the concept, the relevant picture depicted as follow:



The black bar area become the training of past stock up and down data records. The small orange bar area would be the prediction of the model. The grey bar area would not be considered as training as it is far away to the predictive target.

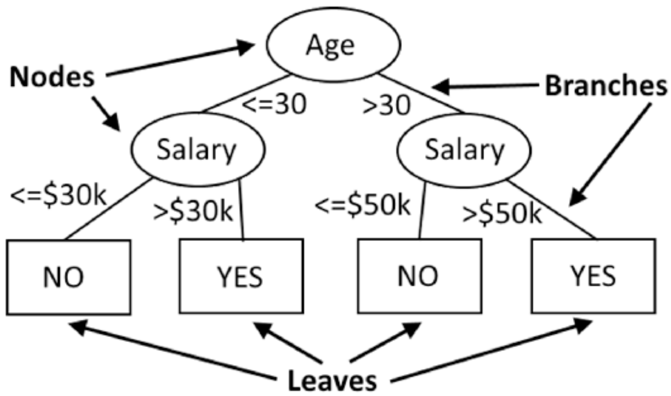
Regarding our research focusing period, it is the stock performance of year 2021, the training dataset and validation dataset would be divided according to the first 9 months(from January to September,188 days) and last 3 months (from October to December,64 days) respectively. Moreover, the total 8 mainstream classifiers would be considered and called from the scikit-learn packages in python which are:

1. Decision Tree Classifier
2. Random Forest Classifier
3. Linear Discriminant Analysis
4. K-Neighbors Classifier
5. Gaussian Naive Bayes
6. Support Vector Machine
7. Logistic Regression
8. Quadratic Discriminant Analysis

6.2 Reviewing the classification algorithm structures

1. Decision Tree classifiers:

Decision tree model is common to be used in different science areas such as operation research, decision science. In machine learning, decision tree learning is to use the tree structure concept to make the prediction. In the dataset, there would have many features and those become the attributes of the nodes. The nodes is to do the comparison on the attribute and generate the outcome for developing the tree in-depth so that the tree model can find out the final predictive result. The branches supports the expansion on the tree model after the attribute's comparison. Using the research paper from Yates (2018), the following flow chart would be:



Suppose this scenario is to decide whether a company should approve the loan to a person, the decision tree model can make a prediction based on the person of information such as salary, age etc.

In general, decision tree model is very popular in machine learning because of the numerous advantages. It is simple to use and understand. The model does not take many preprocessing and the concept inside the model which is merely counting the result and find out the most reasonable results based on all the attributes. Most importantly, there are many metrics to evaluate the decision tree model so that the model performance can be easily seen. This benefits to do the model scaling and increase the model transparency. The metrics is mainly to measure the impurity of the tree model and there are three main metrics such as:

For each node named as t and let X and n be the dataset and the numbers of class:

$$Entropy(X,t) = -\sum_{i=1}^{n} p(i|t)log_2 p(i|t)$$

$$Gini(t) = 1 - \sum_{i=1}^{n} p(i|t)^2$$

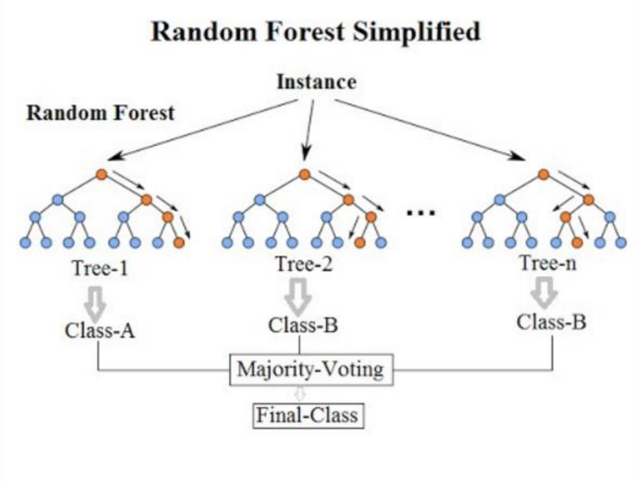
$$ClassifiationError = 1 - \max\{p(i|t)\}$$

$$Information\, gain = Entropy(X) - Entropy(X,t)$$

The ideas of measurement in impurity is to calculate the entropy in the models. The concept of entropy implies the degree of uncertainty of the decision tree model. It is the same with the previous concept which is mutual information techniques. When the entropy is large, the conclusion is hard to make as the result involves much random noise. Therefore, the lower entropy is favorable to decision tree model.

2. Random forest classifiers:

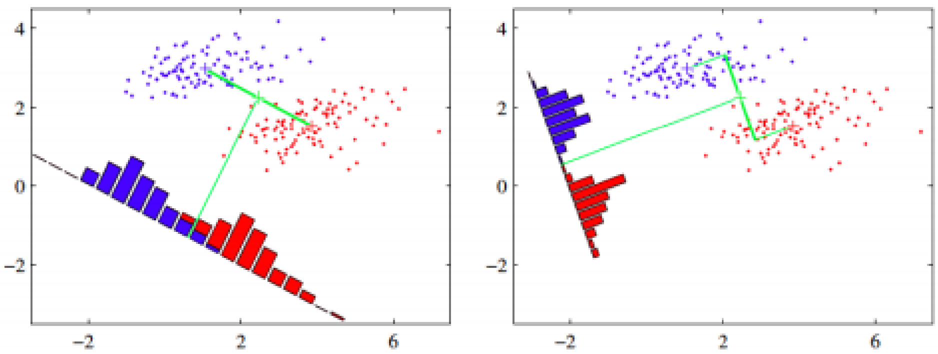
The random forest algorithm is one of the ensembles learning methods to deal with the classification problem. With using the decision tree concepts, the algorithm of idea is to pick the multiple decision trees randomly with certain attributes to generate the following results. After that, it would make the majority of voting among the multiple tree models to decide the final class in the classification problem. That is, depicts it by graphs,



The random forest is one of the powerful algorithms in machine learning. With using the techniques of decision tree, it can say that there could have many sub-models to do the predictions under the random forest models and each of the sub-models can randomly select the different important attributes to generate the result. Those could be the small decision tree. This helps to avoid the overfitting problem and be robust to the outliers. Therefore, generally the random forest algorithms has the high accuracy on doing the prediction and handle the prediction in large dataset effectively.

3. Linear Discriminant Analysis (LDA):

The LDA tools was originally developed in statistics field. Comparing with other classification algorithms, LDA aims to find out the linear combinations among the features and maximize the distance of separation between the classes that is covariance but minimize the variance within every classes. Therefore, that would have a subspace to represent the combination. Similar to principal component analysis (PCA), both they are finding the best linear combination expressed by one dependent variable. According to Bishop (2006), the following concept described as below:



The graphs shows how the 2D data projected onto 1D space while maximizing the covariance. This is called dimensional reduction and the separation can be clearly seen. Therefore, the classifications can be made using such concepts. To illustrate the concepts in mathematical forms, let $S_b \in R_{n\times n}$ and $S_w \in R_{n\times n}$ where S_b is the covariance of between-class and S_w is the covariance of within-class.

$$S_b = \sum_{k=1}^{K} (m_k - m) N_k (m_k - m)^T$$

$$S_w = \sum_{k=1}^{K} \sum_{n=1}^{N_k} (X_{nk} - m_k)(X_{nk} - m_k)^T$$

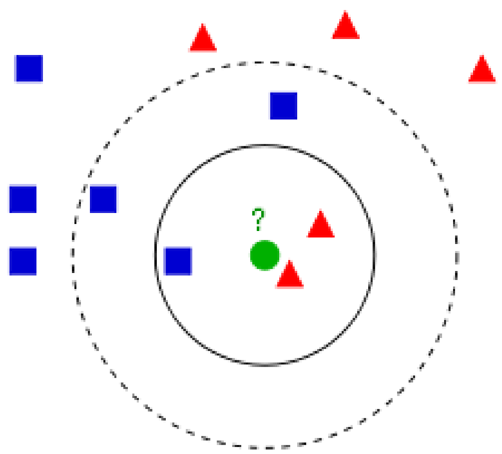
The X_{nk} is the n^{th} data inside the k^{th} class while N_k is the numbers of data inside the class k. The m would be the total mean of all data and m_k would be the mean of the k^{th} class. After calculating the two covariances, the next target is to minimize the both covariances in the ratio-form so that the orthogonal projection could be finished. That is,

$$minimize \quad \frac{w^T S_b w}{w^T S_w w}$$

The minimization steps would be skipped as it involves other further concepts and could be very complex. After finishing the minimization, the separation between the classes could appear and classify the data into the labels.

4. K-nearest Neighbors Classifier (k-NN):

In machine learning, k-NN algorithm could be the simplest structure comparing other algorithms. The k of the k-NN algorithm means the numbers of k of nearest data point whenever the classifier has to determine the classes for new data. Firstly, the k-NN model needs to store features input to form the vectors inside the multi-dimensional space with one following class label each. This can assist to set the decision boundaries. Next, as the decision region is known, the important part is to determine the k for the k-NN model. Since there the k value is the key to determine the performance for k-NN, it will extend to the optimization problem or hyperparameter tuning cases. As these topics are out of the research scope, there would not have further explanations on it. Taking the example on below:



As the figure shown, let's say the green dot needs to be assigned with one label. If setting the k = 2 or 3, within the dot line circle, obviously the green dot would be assigned to red rectangle groups as those red rectangle data points has the shortest distance. However, if the k = 5, the green dots would be assigned into the class group of blue squares by seeing the dashed line circle. As a result, k-NN algorithms are the simplest classification algorithm because the structure is easy to understand, and there are some distance calculations options for k-NN model such as Euclidean distance, Hamming distance.

5. Gaussian Naïve Bayes:

Gaussian Naïve Bayes, also called Naïve Bayes classifier, is the probabilistic classifier in statistics and so it follows a probability distribution and its assumption set each of the features to be independent. To apply the model in machine learning, some statistical properties needs to be followed including bayes theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

The bayes theorem is a posterior probability consists of prior probability and the evidence. The Gaussian Naïve Bayes model leverage a part of this theorem which is the numerator of formula to do the classification. The denominator could be ignored as it is not related to the concern event that is probability of event B. Therefore, the classification formula can be written as:

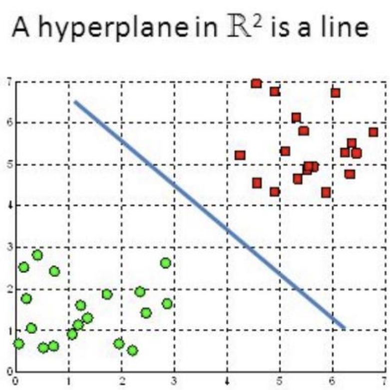
$$\hat{y} = \underset{m \in \{1,\ldots,M\}}{\operatorname{argmax}}\, p(C_m) \prod_{i=1}^{n} p(x_i|C_m)$$

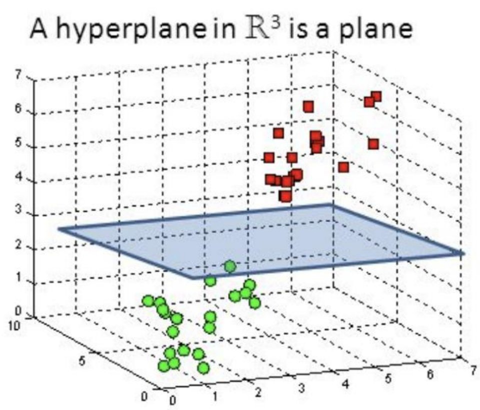
$$where\ n\ and\ m\ is\ the\ numbers\ of\ data\ and\ numbers\ of\ classes\ respectively$$

The objective of the Naïve Bayes classifier is to maximize the posterior probability among the classes. This method is also named as maximum a posteriori probability (MAP) in statistics.

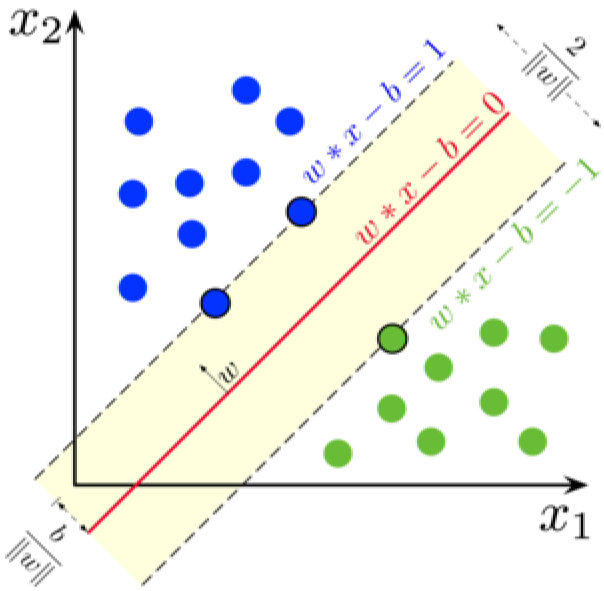
6. Support Vector Machine (SVM):

In machine learning, SVM algorithms can be one of the most powerful and robust algorithms. The ideas of SVM is to construct a support vector using training data to find out the optimal maximum margin for separating the data. The following margin leads the SVM algorithm to determine the decision boundaries which is called the separation line in 2-dimenstional space. When the data is in high dimensional space, it would be defined as hyperplane. As a result, the below graph describe the following concepts:



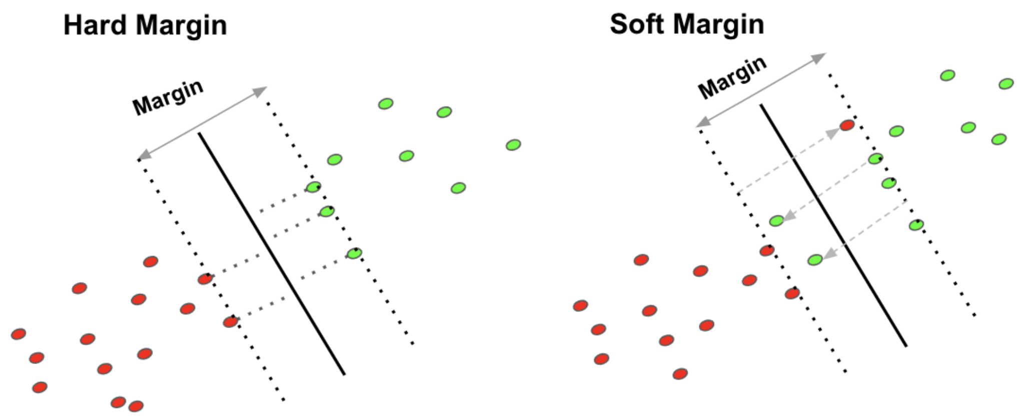


The separation plane plays an important role in SVM algorithm as it decides the class labels to the new data. Taking the 2-dimenstional space as example, the scenario could be:



As shown in the graph, the separation boundary can be regards as a line as this is in 2-dimenstional space. The data example represents the form: $(x_1, y_1), \dots, (x_n, y_n)$. The w and x are the vectors while hyperplane would be written as $w^T x - b = 0$ where b is a constant. The $\frac{b}{\|w\|}$ and $\frac{2}{\|w\|}$ are to describe the distance between hyperplane and the origin and the width distance of the hyperplane respectively.

To do the classification, there are two types of margins to assign the class labels. When one new data appears, the data would be assigned with blue circle class if $w^T x - b \geq 1$. If $w^T x - b \leq -1$, the class label would be green circle class. This is called hard margin and it often being adopted if the data is expected to have a strict and more accurate result. If there is not a few data stays between the margin, another margin would be favorable to used that is soft margin because the data are often really close to the hyperplane and may not be separated linearly. However, adopting soft margin allows there would have some mistakes in classification. That is the case:



In some complex scenario, some real-life cases often contains some non-linear relation between the data. Hence, the best hyperplane would not be easy to find out. Generally, regularization is the method to deal with such cases. The goal is to set up the penalty for the SVM algorithm and minimize the cost function, the cost function would be defined as:

$$J(w) = \sum_{i=1}^{n} \max\left(0, 1 - y_i[w^T x_i + b]\right) + \lambda \|w\|^2$$

Where λ is the penalty parameter. It decides the trade-off between the distance of maximal margin and the classification accuracy. For more advance adjustment, there can have another option that is kernel trick. The use of it is to solve the more complex circumstance by applying the mathematical complex function. The kernel topic would be skipped as this is excluded from this research project and it involves not a few explanations.

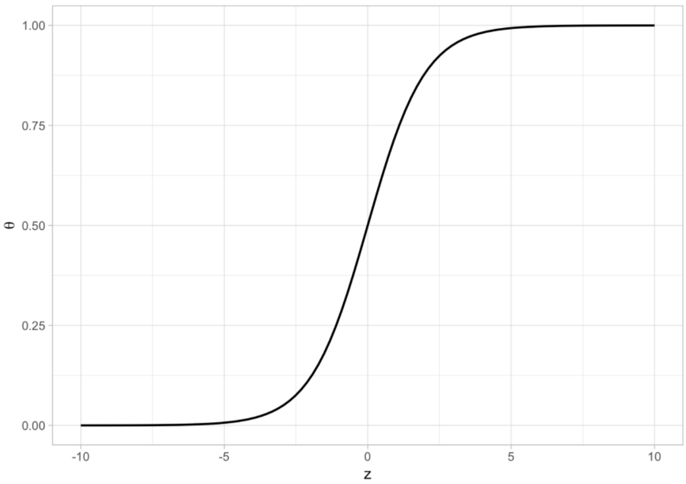
7. Logistic Regression:

In statistics, logistic regression model is a type of statistical model to calculate the probability and generate the binary dependent variable for the event. With using the linear regression concepts, it substitutes the ordinary least squares into the logistic function so it would also be the generalized linear model (GLM). In machine learning, it is called sigmoid function.

Let $z \in (-\infty, \infty)$ and the Logistic function be $logit(\theta)$:

$$logit(\theta) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$$

Therefore, mathematically, the 2d graph would become:



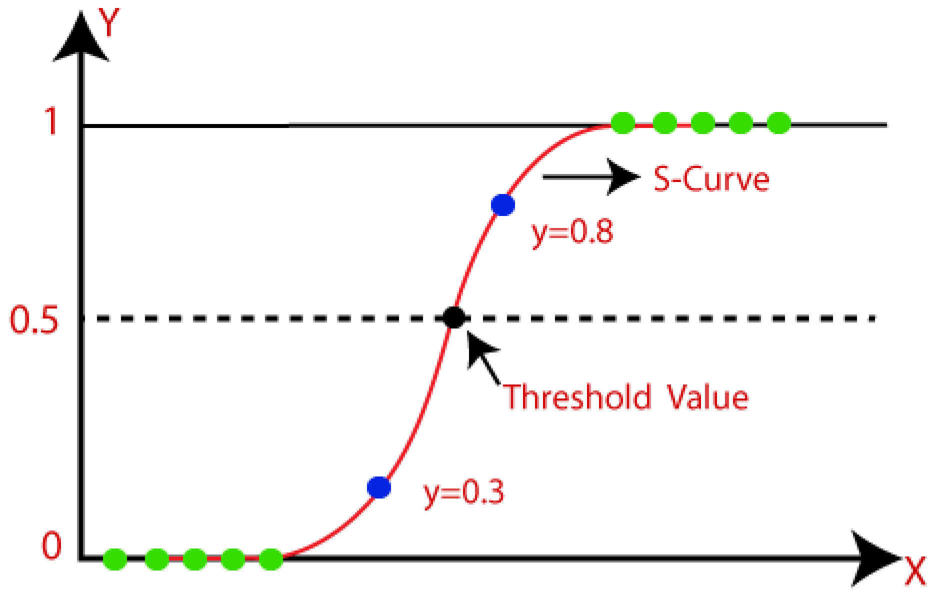
Before developing the logistic regression model, the linear regression need to be obtained first. That is:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

Put the linear regression model into logistic function, it would become:

$$logit(\hat{y}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}}$$

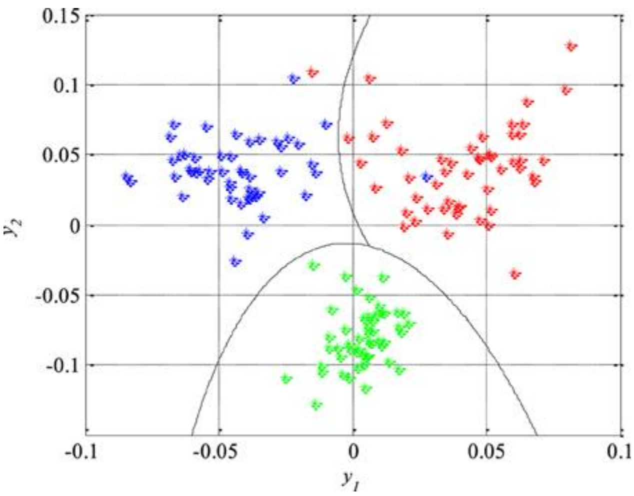
the logistic model makes the classification based on the threshold value. The threshold value commonly set to 0.5 and it helps to classify the new data. When the value output smaller than 0.5 from the model, it would be labelled to first class, otherwise, it would be another class label. Following example shown as below:



To achieve the high accuracy result, the logistic regression model needs to follow the statistical assumptions including linearity, no multicollinearity.

8. Quadratic Discriminant Analysis (QDA):

QDA algorithm has the close relations to the previous mentioned LDA algorithm. The assumption for LDA is that each of the classes need to follow Gaussian distribution. However, the QDA algorithm merely requires the co-variance from each classes exists. Comparing LDA and QDA, LDA would be suitable using in constructing the linear separation boundary to classify the data. For QDA, it normally used to deal with non-linear relation separation boundary that is decision boundary would be quadratic. In view of the concepts, suppose there are three features variable, LDA will contain the $x$vector:$[x_1,x_2,x_3]$while QDA will contain the x vector: $[x_1,x_2,x_3,x_1^2,x_1x_2,x_1x_3,x_2^2,x_2x_3,x_3^2]$. As a result, the quadratic classifier can be written as $x^T Ax + b^T x + c$. For instance, the quadratic classifier can be depicted as:



6.3 Machine learning classification models evaluation:

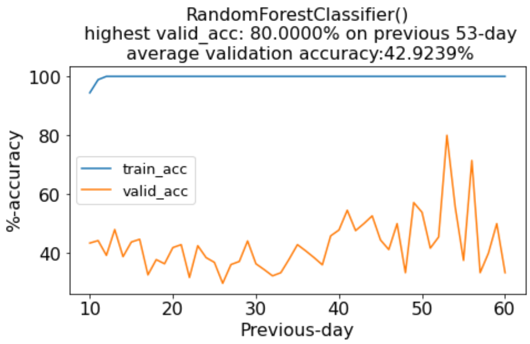
Next, the input time series data is set to be at least previous 10-day to at most previous 60-day. To evaluate the classification algorithm accuracy, the following formula simply defined as:

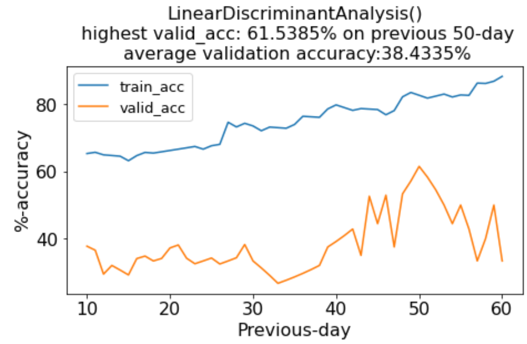
$$Accuracy = \frac{Numbers\ of\ correct\ prediction\ from\ the\ algorithm}{Total\ numbers\ of\ prediction\ from\ the\ algorithm}$$

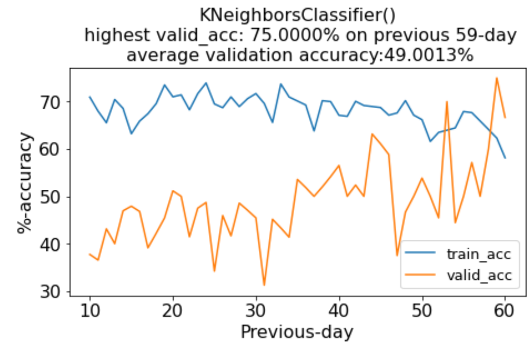
$$= \frac{true\ positive + true\ negative}{true\ positive + true\ negative + false\ positive + false\ negative}$$

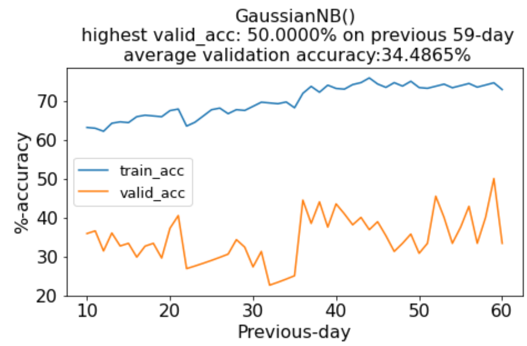
Let’s see the performance of each stock prediction with each classifier:

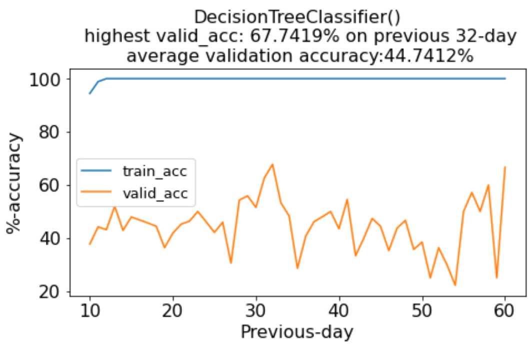
For Apple stock:

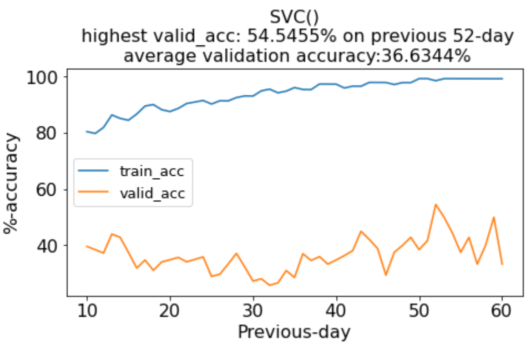


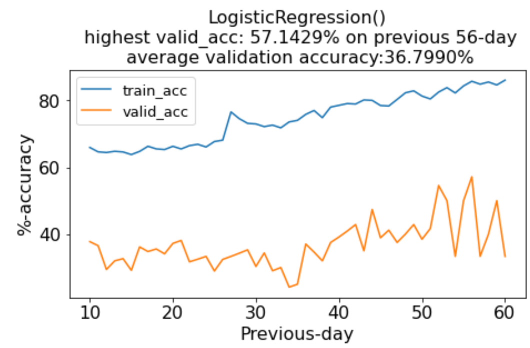


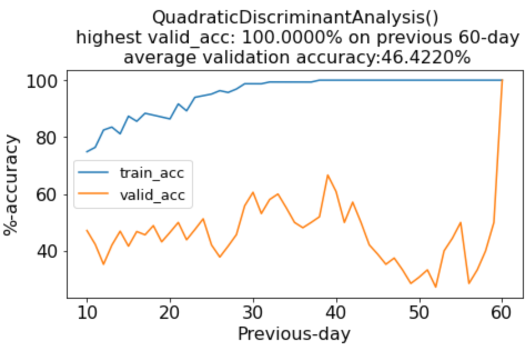




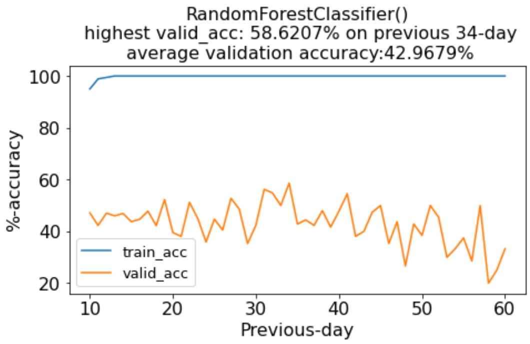


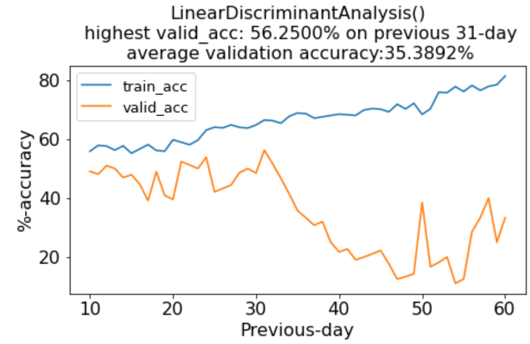


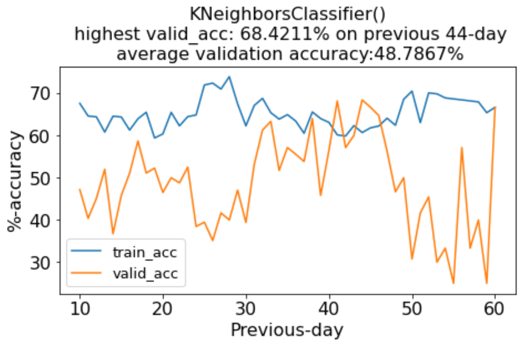


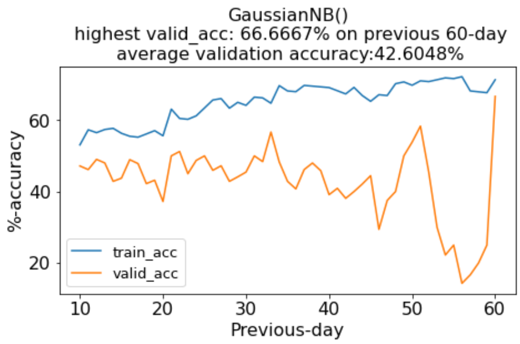


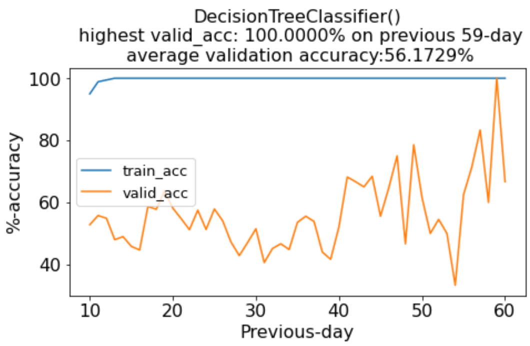
For Tesla stock:

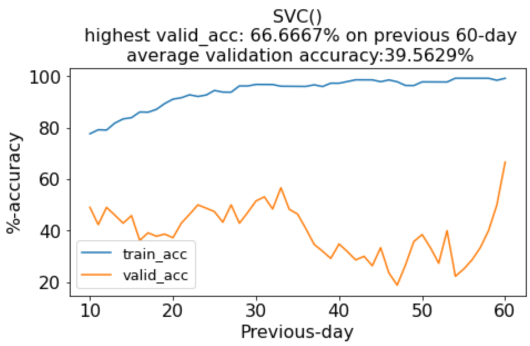


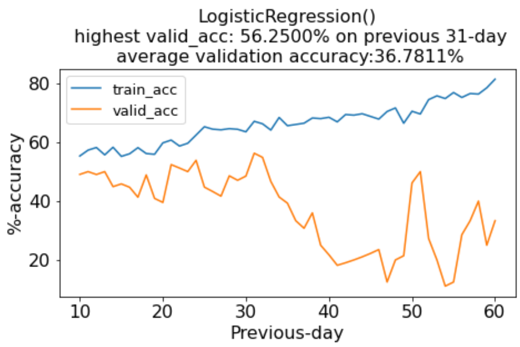


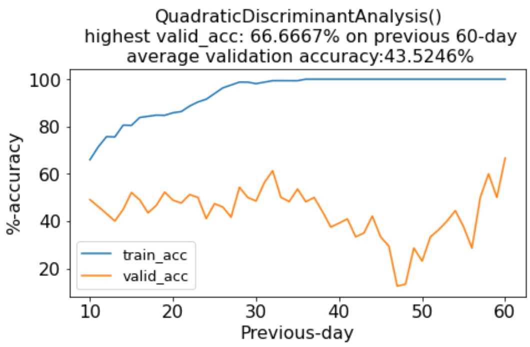




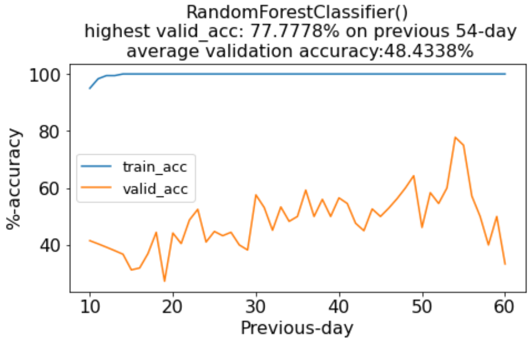


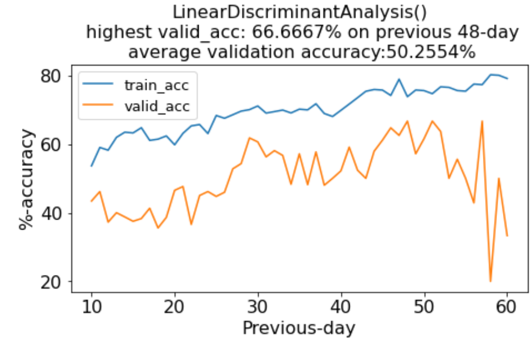


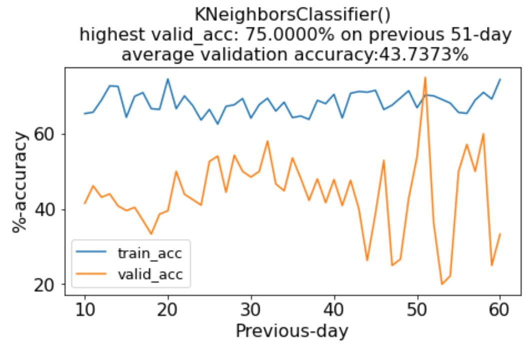


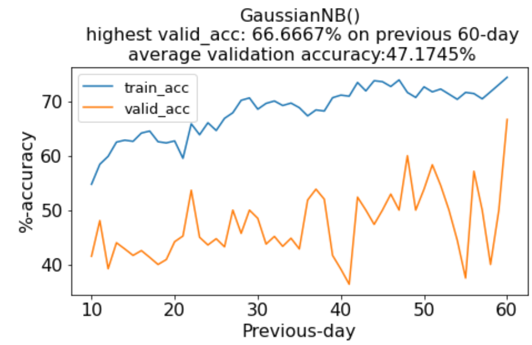


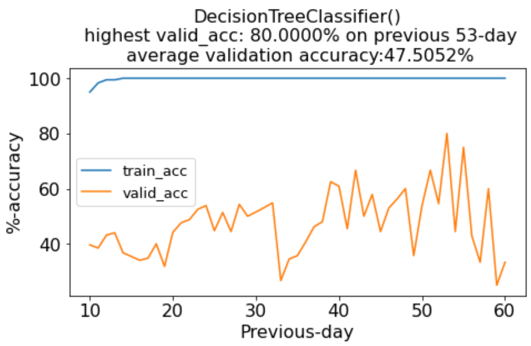
For Nvidia stock:

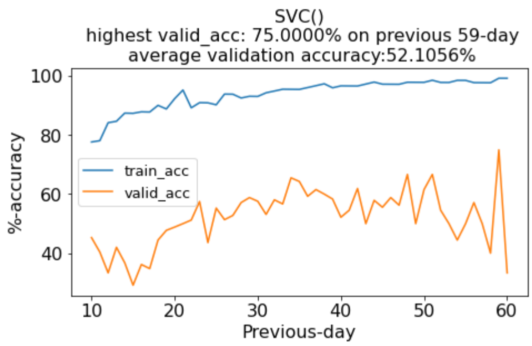


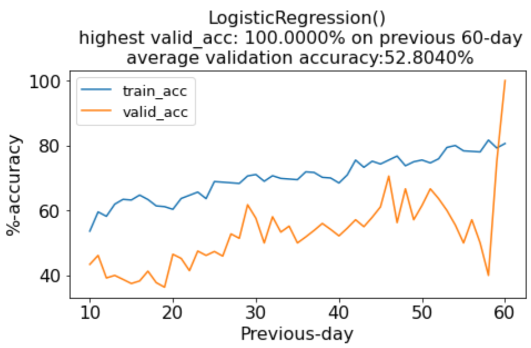


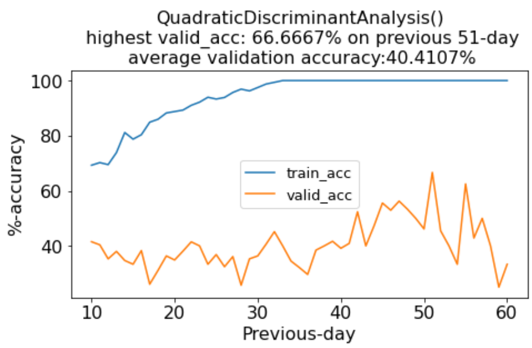




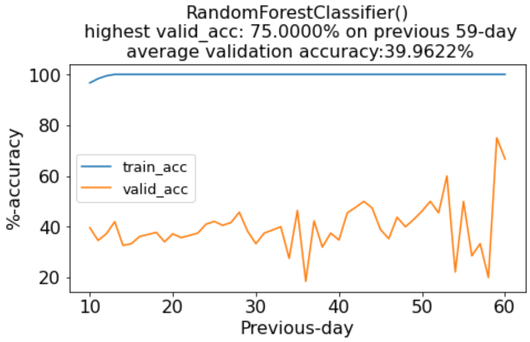


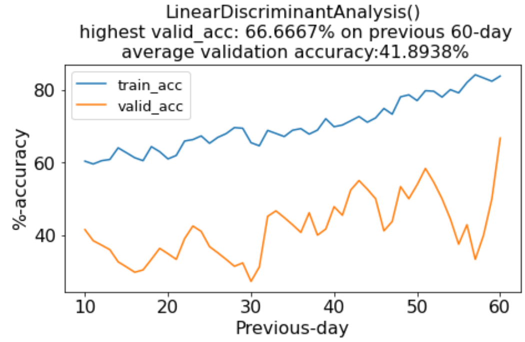


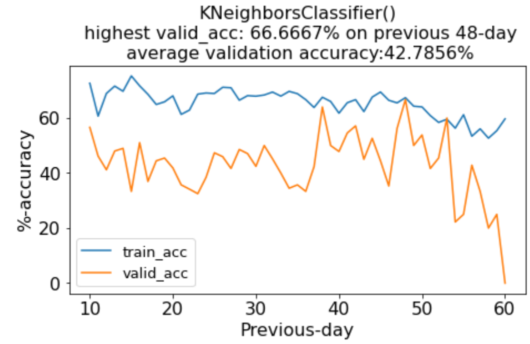


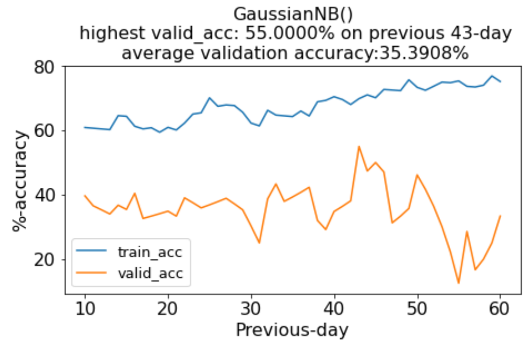


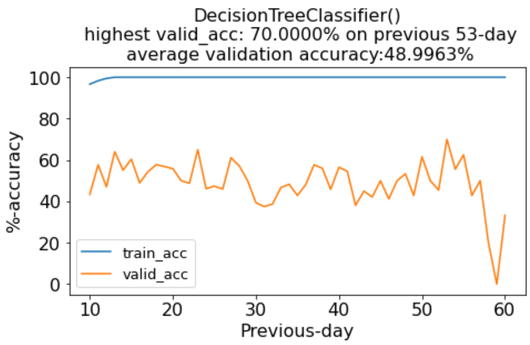
For Microsoft stock:

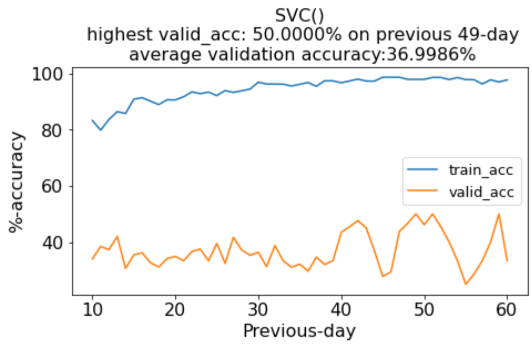


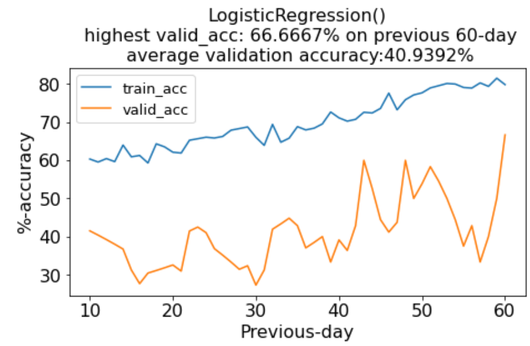


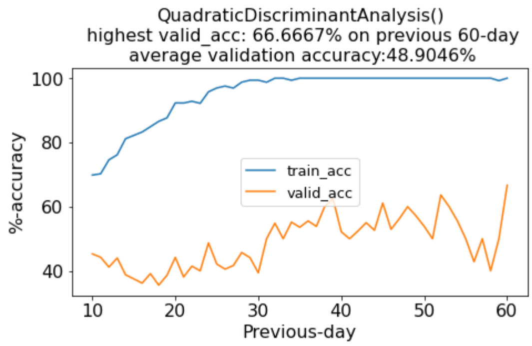




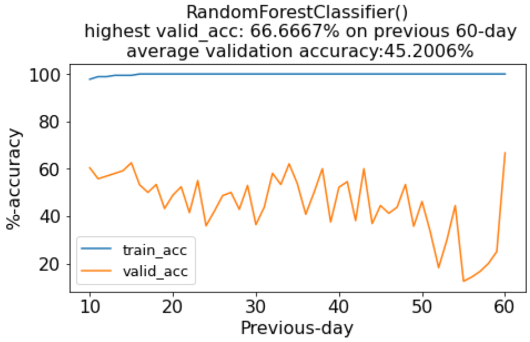


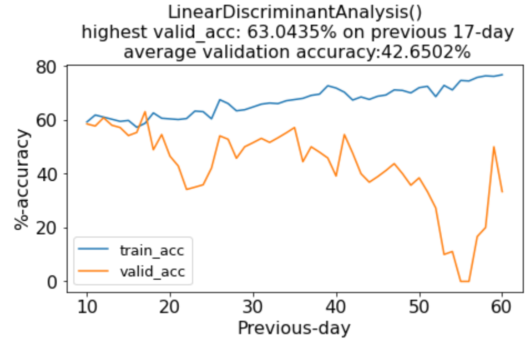


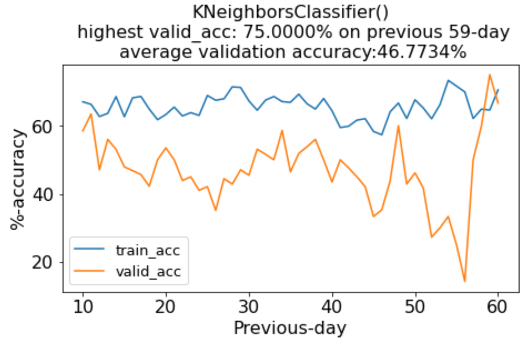


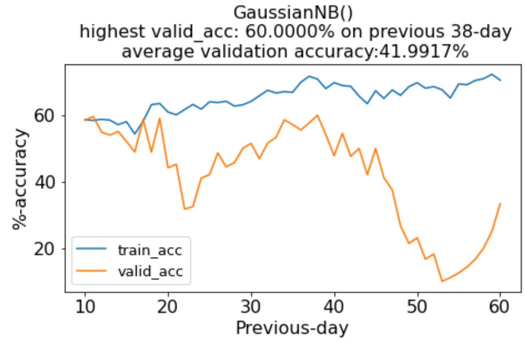


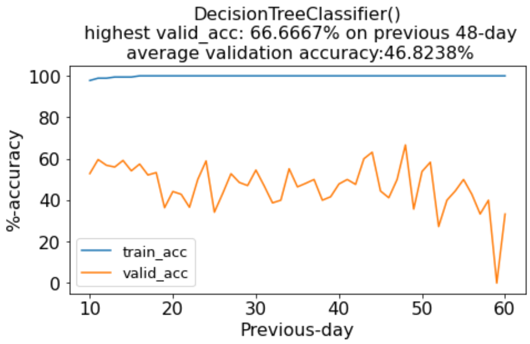
For Amazon stock:

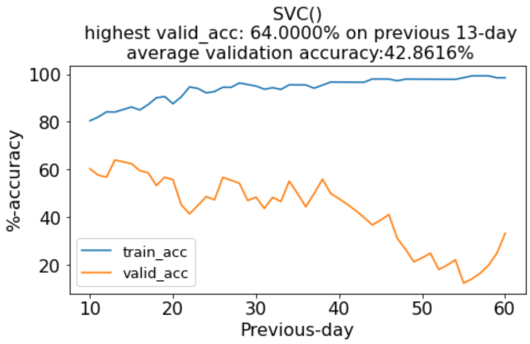


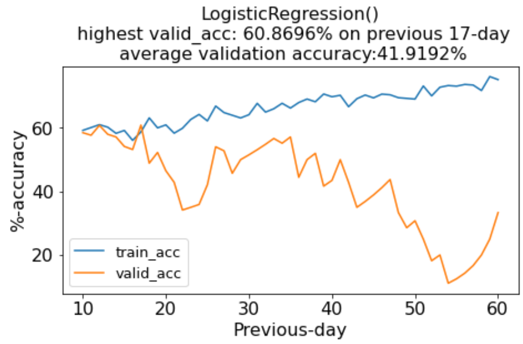


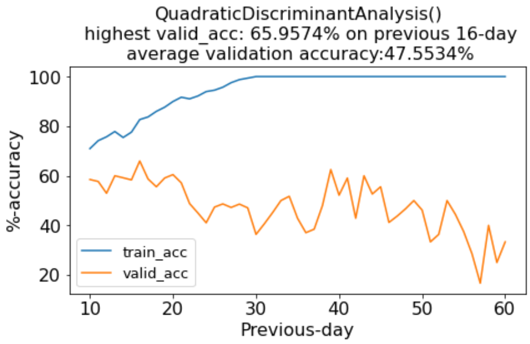




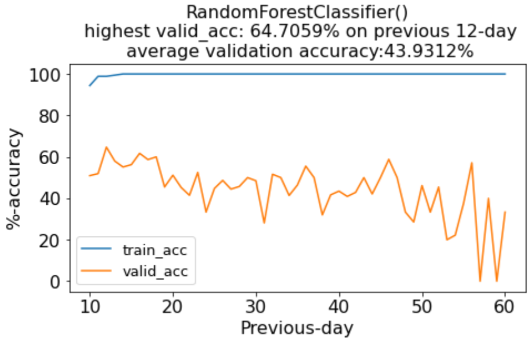


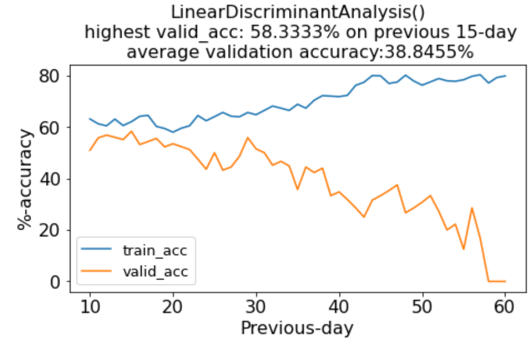


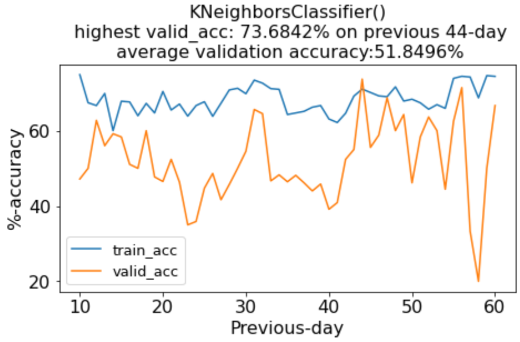


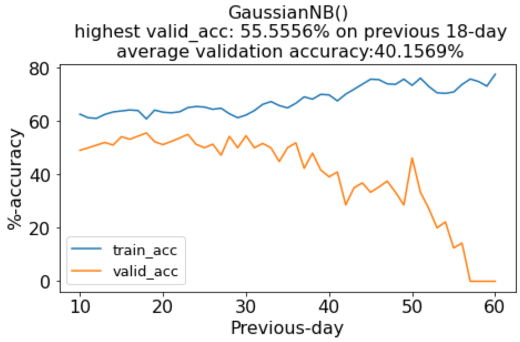


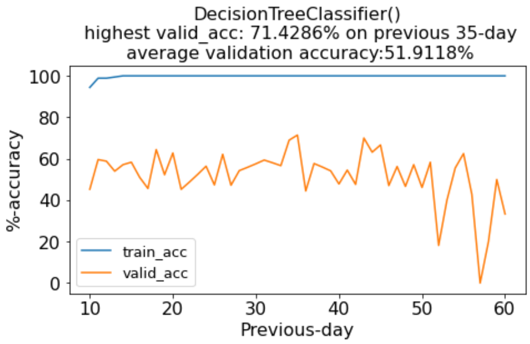
For AMD stock:

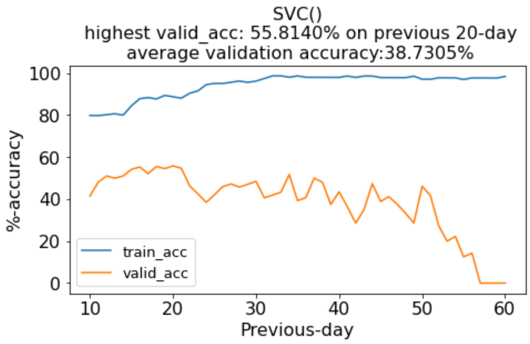


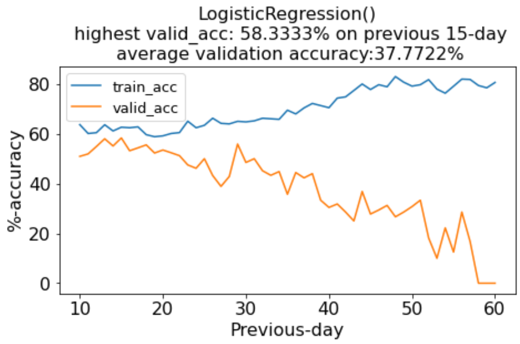


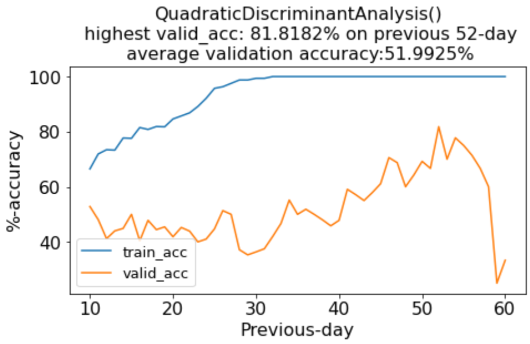




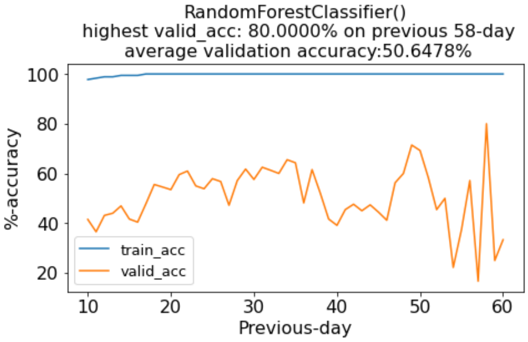


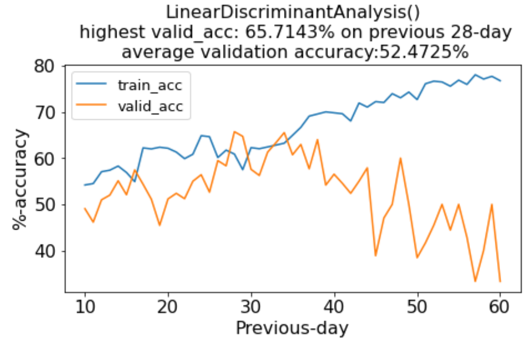


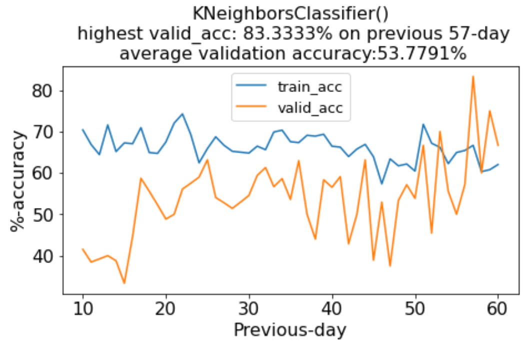


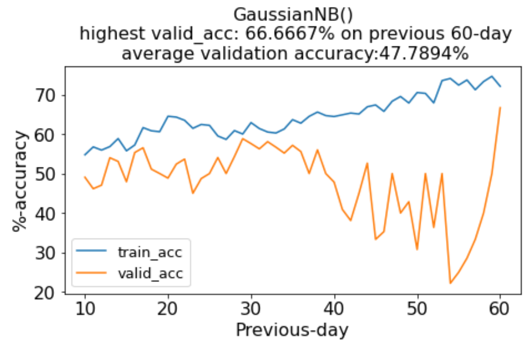


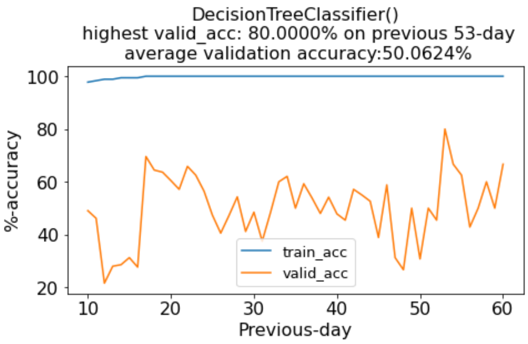
For Meta stock:

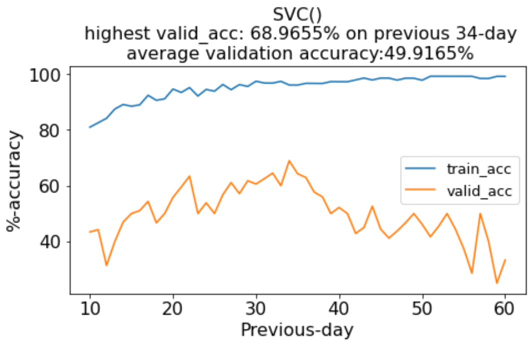


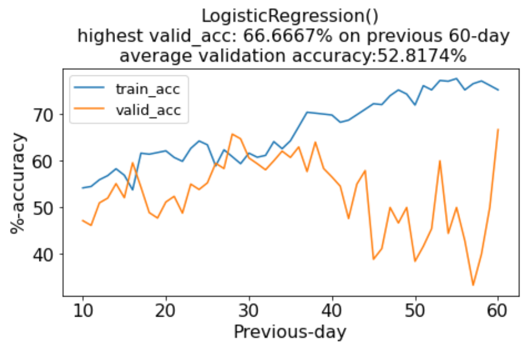


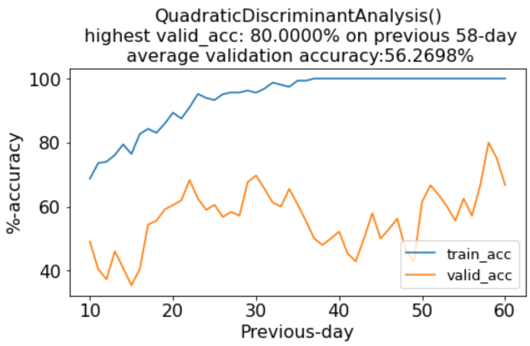




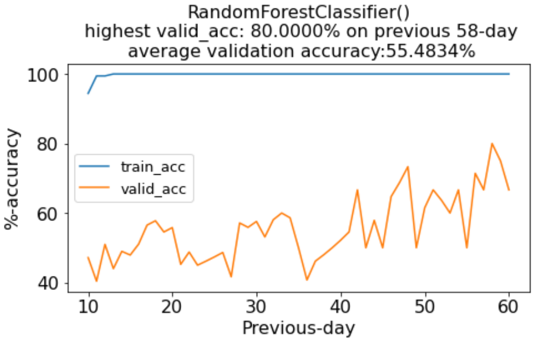


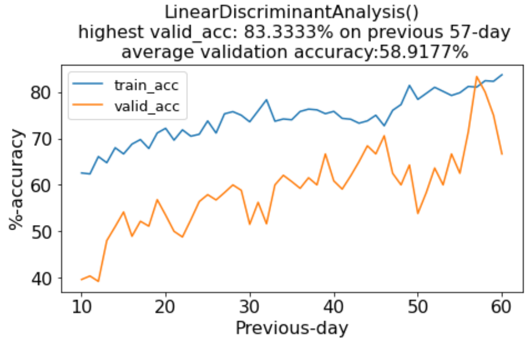


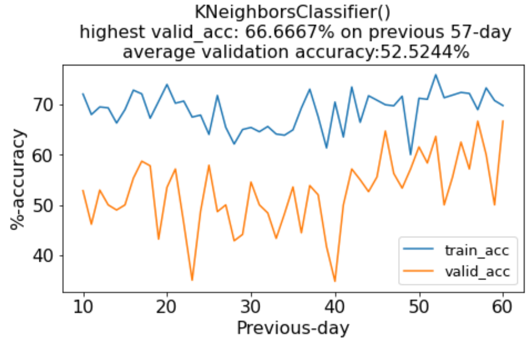


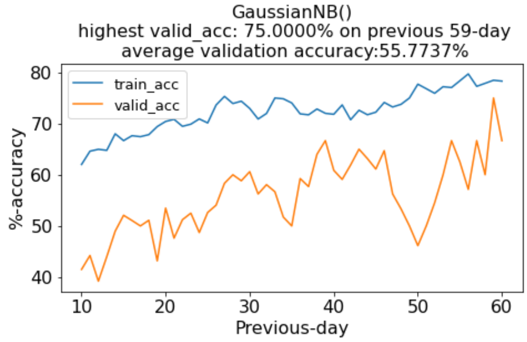


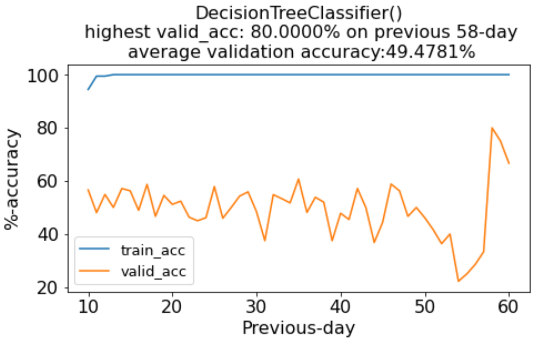
For Adobe stock:

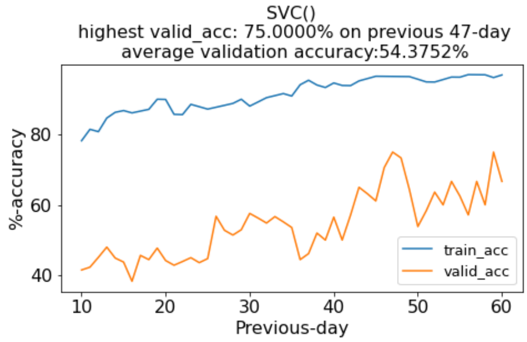


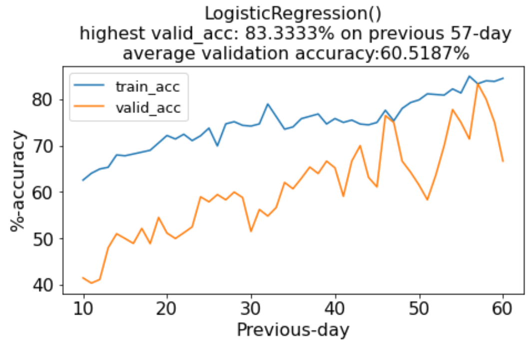


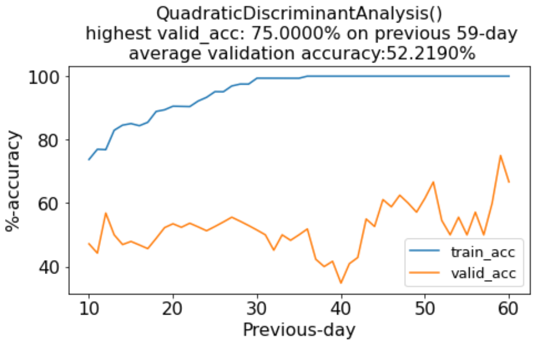




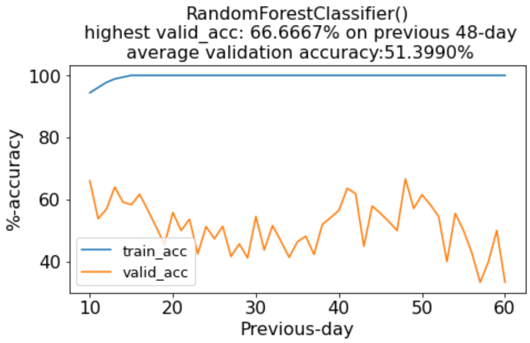


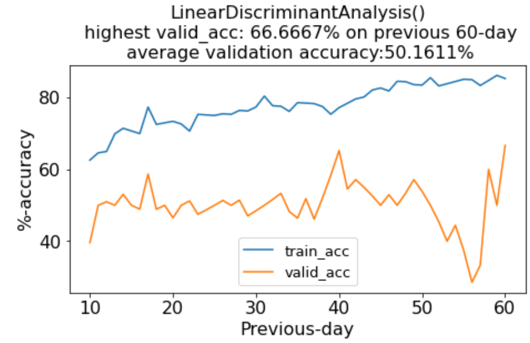


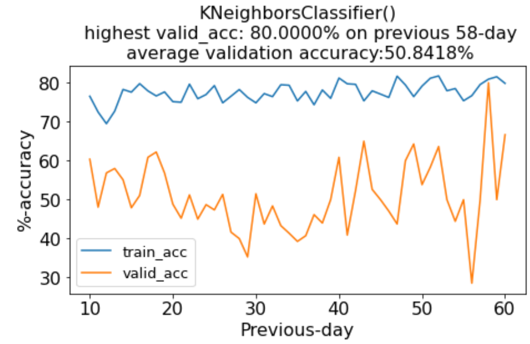


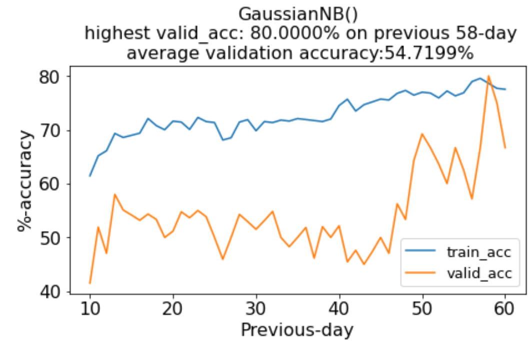


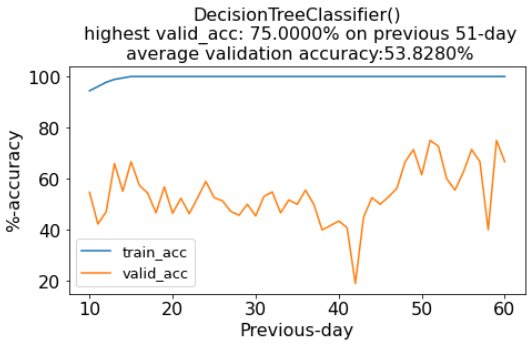
For Google stock:

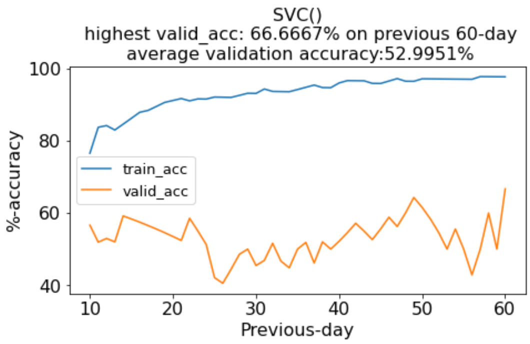


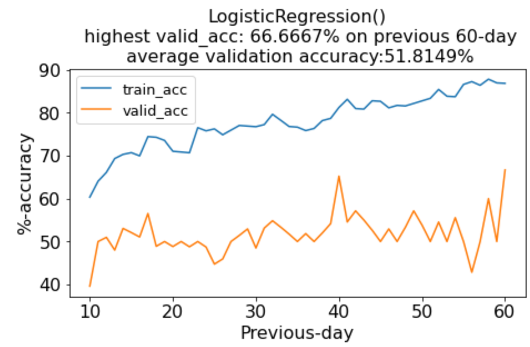


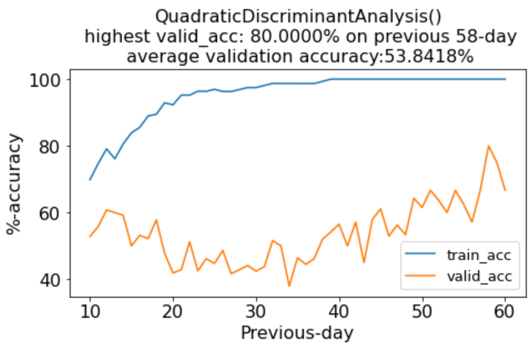




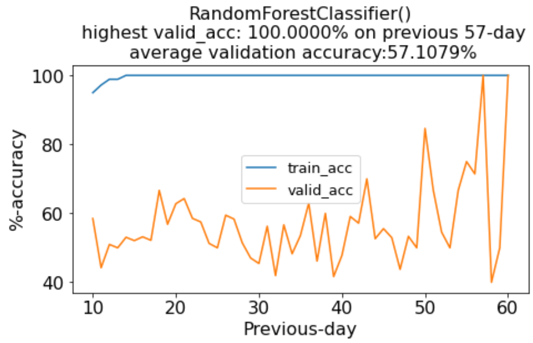


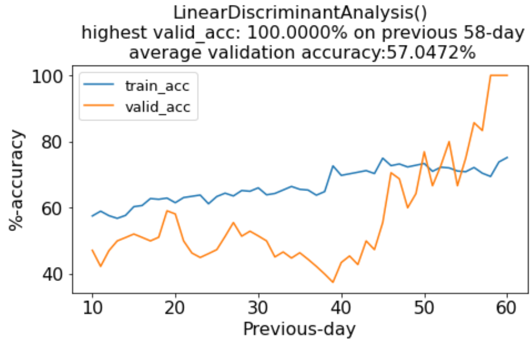


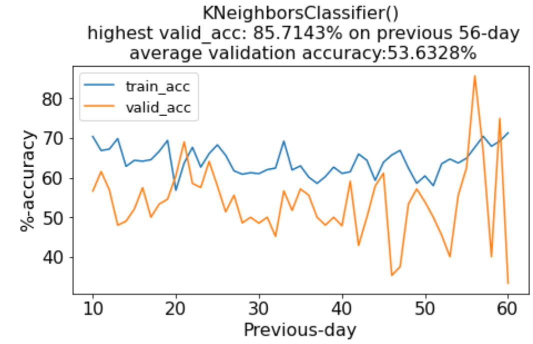


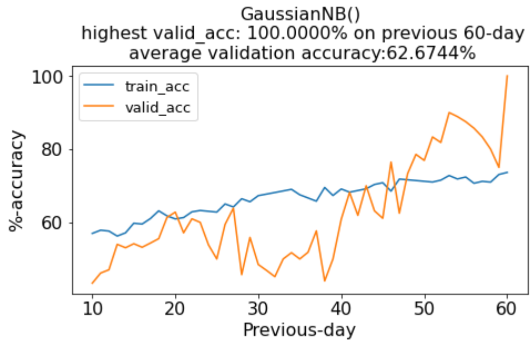


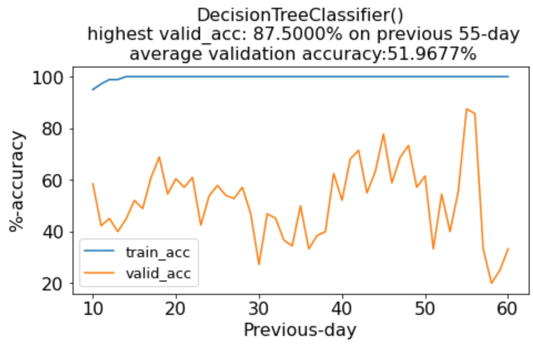
For Pfizer stock:

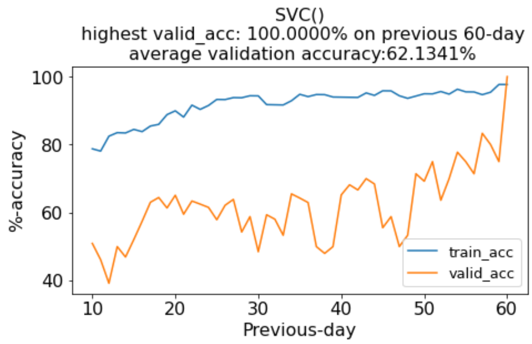


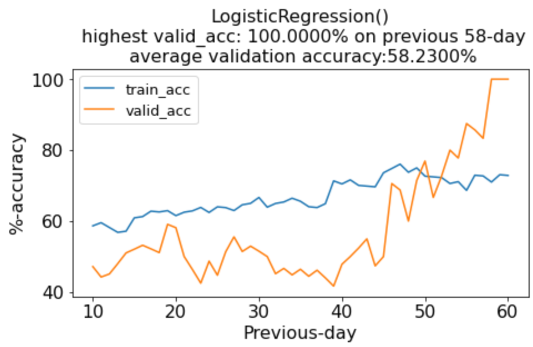


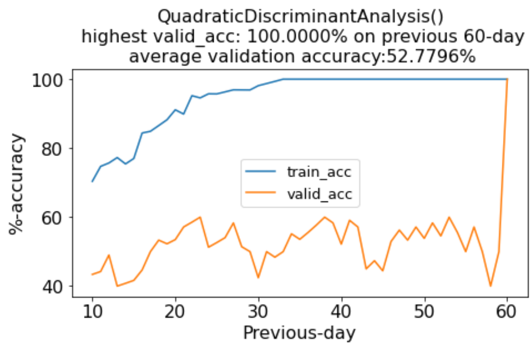












6.4 Machine learning classification models accuracy analysis:

After a bunch of performance line charts, it is clearly to see all of the classifiers performs differently in every stocks. This is not surprising as the stock market has fluctuations always and the close price moves with many noises. In view of this situation, to understand the performance more in deep with a statistical form, there are some summary statistics information for evaluating the performance to every models. Summarizing the accuracy in statistical table from python, the relevant table shown below:

	RandomForestClassifier()	LinearDiscriminantAnalysis()	KNeighborsClassifier()	GaussianNB()	DecisionTreeClassifier()	SVC()	LogisticRegression()	QuadraticDiscriminantAnalysis()
count	510.000000	510.000000	510.000000	510.000000	510.000000	510.000000	510.000000	510.000000
mean	47.805774	46.606616	49.371195	46.276277	50.148738	46.631443	47.039567	49.391824
std	12.006282	13.890292	10.494048	13.734225	12.133248	13.163140	14.294220	10.829339
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	12.500000
25%	40.782828	38.461538	43.333333	38.095238	44.186047	37.500000	38.803855	41.951613
50%	47.826087	48.644689	50.000000	47.871377	50.000000	47.058824	48.888889	50.000000
75%	55.442177	54.545455	56.250000	53.961538	57.142857	55.555556	54.838710	56.250000
max	100.000000	100.000000	85.714286	100.000000	100.000000	100.000000	100.000000	100.000000

To summarize the average of model accuracy regarding the time series period from previous 10-day to previous 60-day, generally all models of the accuracy are close to 50%. The interquartile range from 25% to 75% are mostly started from around 40% to 56%. However, by checking with the max value, most of the model accuracy achieved with 100%

Through reviewing the highest accuracy of all the models in each stock classification prediction, the possible highest accuracy range can sum up between the highest accuracy from all the model in each stock. This helps to understand which time point benefits to the best model for getting the highest accuracy. Based on the result, the range would be:

Highest accuracy model analysis for each stock					
Stocks:	Smallest day used:	Largest day used:	Average of day used:	Highest accuracy model (% - accuracy)	Previous day used from the best model
Apple	32	60	53	Quadratic Discriminant Analysis (100%)	60
Tesla	31	60	47	Decision Tree (100%)	59
Nvidia	48	60	55	Logistic Regression(100%)	60
Microsoft	43	60	54	Random forest (100%)	59
Amazon	13	60	34	K-neighbors (75%)	59
AMD	12	52	26	Quadratic Discriminant Analysis (81.82%)	52
Meta	28	60	51	K-neighbors (83.33%)	57
Adobe	47	59	57	Linear Discriminant Analysis, logistic regression (83.33%)	(57,57)
Google	48	60	57	K-neighbors, Quadratic Discriminant Analysis, Gaussian Naïve Bayes (80%)	(58,58,58)
Pfizer	55	60	58	Random forest, Linear Discriminant Analysis, K-neighbors, Gaussian Naïve Bayes, Decision Tree, Support Vector Machine, logistic regression, Quadratic Discriminant Analysis (100%)	(57,58,56,60,60, 58,60)

6.5 summary for machine learning classification model:

Although the average of model accuracy are not satisfactory overall, some interesting discoveries can be found using the pervious time series period. The predictive accuracy largely affected by different time point training. With analyzing the plotted line chart of all the model accuracy, the time point tells how many previous of day to be used as the input data for the model would achieve the high accuracy result. Based on the mean and the numbers of previous day used from the highest accuracy model, all the predictive results showed that the best model achieved the highest accuracy with at most 100% generally distributed from the remote time series data between previous 56-day to 60-day. Furthermore, the k-nearest neighbor's classifiers and quadratic discriminant analysis performs the best in US stock classification problem as those models often achieved the highest accuracy which appeared amongst the four different stocks out of ten.

7. Fundamental background of deep learning:

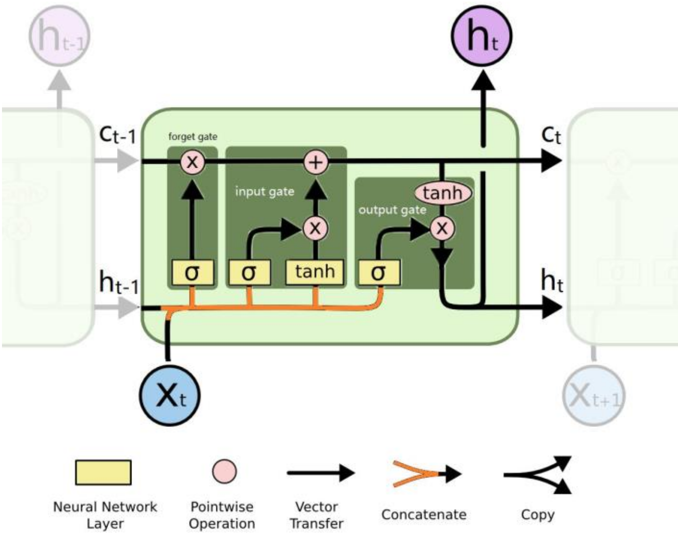
In artificial intelligence, deep learning is the most powerful learning standards constructed under the artificial neural networks and it can be the more advanced level of a set of machine learning algorithms in another words. The idea of neural networks were inspired from the human brain system structure, and it mimics the human thinking structure. This means the structure follows the network topology theory. The first architecture contains different kinds of layer and each of the layer has its own functionality. Typically, the artificial neural network consists of three main layers which is input layers, hidden layers and output layers. The input layers is to collect the input information and process it properly for passing those input data to the next layers. The hidden layers is the main working layer to understand the data in deep. This layers often involves many mathematical function analyses and try to investigate the complex relation of the data. For the last layers, output layers merely shows the following results after the bunch of data processing. More importantly, every layer could contain many neurons. The neuron of neural network is the basic elements for handling the distributed processing task in the whole project. Commonly to see, there could have millions or even billions of neurons in the neural network system. Therefore, in practice, the use of deep neural network usually is to deal with some extremely complex problem in real-life such as image recognition, natural language processing etc.

7.1 Deep learning algorithms in stock regression problems

Recently, artificial neural network are widely used in finance field, especially in stock market. Comparing with machine learning algorithms, deep learning algorithms are much more powerful because of the details and complex processing structure. Meanwhile, the artificial neural network involves not a few techniques and concepts that need to understand before the implementation. In view of the complexity of deep learning algorithms, this capstone project aims to adopt one of the most popular neural networks for investigating the stock regression problem. That is, the long short-term memory (LSTM) networks, a type of recurrent neural network, would be chosen to investigate the stock price prediction performance. The main reason is the characteristics of LSTM structure is benefit for doing the numerical prediction under the full of noises environment. Regarding its processing network structure, the relevant concepts would be introduced in subsequent pages. Moreover, the LSTM network often performs better in time series forecasting and it sourced with many times in other scholar papers. Before constructing the LSTM network, similar to the previous machine learning classification model, the input data again become the previous time-series stock value. The trained network metrics would be some common regression model metrics including the mean squared error, root mean squared error, mean absolute error and mean absolute percentage error.

7.2 Reviewing the deep learning algorithm structure

The LSTM consists of three different gates which is input gate, forget gate and output gate. The structure diagram depicted as below:



The diagram showed one cell processing structure in the recurrent neural network and all the cell operates with the same procedure. The functionality of three gates described as follow:

1. Input gate: This gate is to process the new data with two activated functions. Those two activated functions are sigmoid and hyperbolic tangent functions. The data normally represented in vector form. After transforming the vector value through these two functions, the transformed value would be elementwise multiplicated and its value passes into another node that is the node of value processed from the forget gate.

2. Forget gate: The forget gate is to discard the no meaningful value from the previous data. From the diagram, it can be seen that the sigmoid function is used. If some values of the vector comes up with 0, it means

those value would be completely discarded and forgotten. That is the main step to select the important information of the data.

3. Output gate: The main gate to output the updated and refreshed value and pass it to another cell repeat the same procedure.

In mathematical forms, let $x_t \in \mathrm{R}$, $f_t \in (0,1)^h$, $i_t \in (0,1)^h$, $o_t \in (0,1)^h$, $h_t \in (-1,1)^h$, $\tilde{c}_t \in (-1,1)^h$, $c_t \in (-1,1)^h$, $f_t \in (0,1)^h$, $W \in \mathrm{R}^{h\times d}$, $U \in \mathrm{R}^{h\times h}$ and $b \in \mathrm{R}^h$.

it would be:

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

To summarize, every implementing it would have some new data x_t where t is a time point. The new data represents as vector to all the gates. The f_t is the activated vector in forget gate where W and U are the weighted matrix and b is some bias vectors. Similarly, $i_t \ and \ o_t$ represents the activated vector from input gate and output gate respectively whereas the $\tilde{c}_t$ and c_t means the activated vector from hyperbolic tangent function and updated cell state respectively. The h_t is the processed predicted value with two copies, one is for output and another one is to support the next processing in next cell.

7.3 Deep learning algorithms performance evaluation:

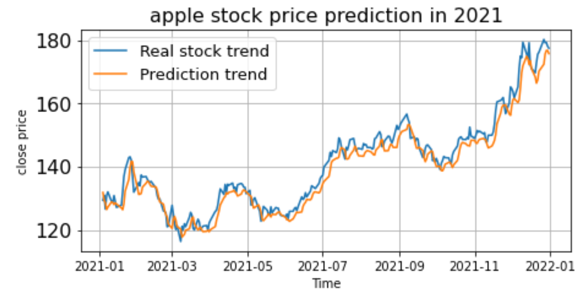
As the deep learning algorithms constructed under the complex structure, the cost of training time could be expensive. Taking it into consideration, again the time series window rolling forecasting would be adopted and the input data firstly would be chosen as previous 60-day time series close price. As the advantages of LSTM network, it can store the useful value and discard the meaningless value. To leverage this powerful technique, the training data set would be extended to the past 5-years (2016-2020) and the testing dataset becomes the year of 2021 close price. Regarding to this stock regression problem, python would be used to help developing the algorithm again. The loss function and optimizer are the gradient descent of mean squared error and Adam optimizer respectively for the deep learning algorithm. On the other hand, the Close price value needs to be pre-processed before the training. The main reason is the close price in every stock has different level of scales. Here, two pre-processing data methods applied into the data which is Z -score standardization and Min-Max normalization, the formulas as follows:

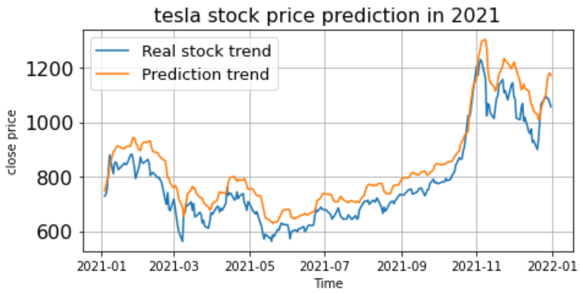
$$standardization(x) = x_{norm} = \frac{x - mean(x)}{standard\_deviation(x)}$$

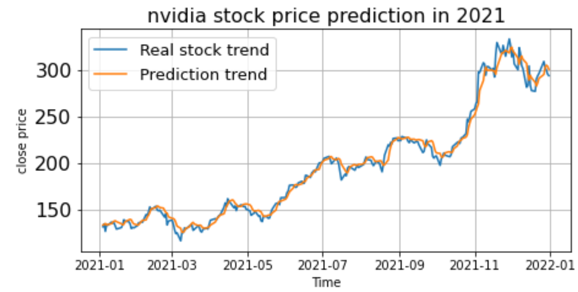
$$MinMax\_normalization(x) = x_{norm} = \frac{x - min(x)}{max(x) - \min\ (x)}$$

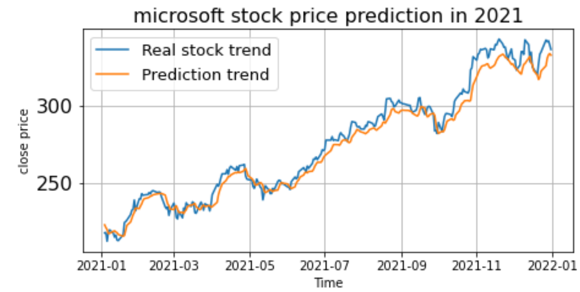
After pre-processing the data, the LSTM network algorithm would be ready to build.

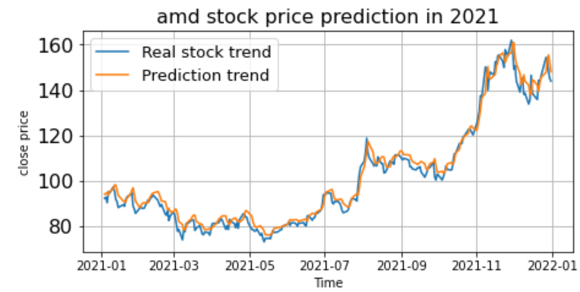
For Min-Max normalization version:

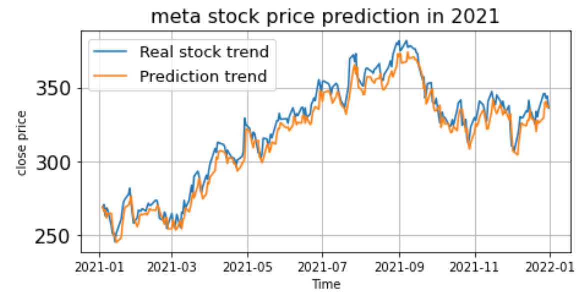


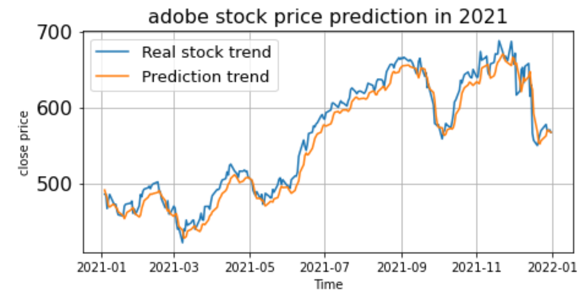


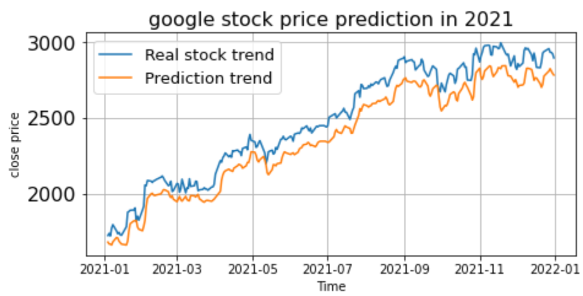


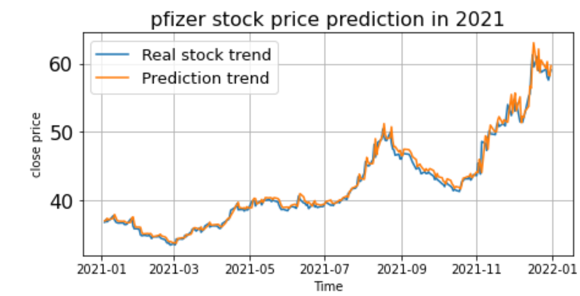












Training information:

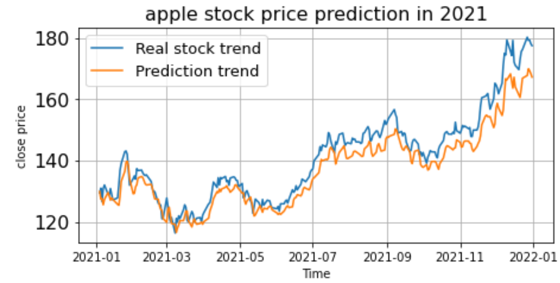
```
stock: apple
Epoch 1/5
600/600 - 21s - loss: 7.6127e-04 - mse: 7.6127e-04 - mae: 0.0189 - mape: 2238.2917 - 21s/epoch - 35ms/step
Epoch 2/5
600/600 - 19s - loss: 4.2768e-04 - mse: 4.2768e-04 - mae: 0.0143 - mape: 6205.6357 - 19s/epoch - 32ms/step
Epoch 3/5
600/600 - 19s - loss: 4.6390e-04 - mse: 4.6390e-04 - mae: 0.0127 - mape: 6639.2246 - 19s/epoch - 32ms/step
Epoch 4/5
600/600 - 19s - loss: 2.3322e-04 - mse: 2.3322e-04 - mae: 0.0102 - mape: 5506.6216 - 19s/epoch - 32ms/step
Epoch 5/5
600/600 - 19s - loss: 1.9506e-04 - mse: 1.9506e-04 - mae: 0.0096 - mape: 2289.1978 - 19s/epoch - 32ms/step

stock: tesla
Epoch 1/5
600/600 - 19s - loss: 5.1296e-04 - mse: 5.1296e-04 - mae: 0.0131 - mape: 40.3331 - 19s/epoch - 32ms/step
Epoch 2/5
600/600 - 17s - loss: 2.7859e-04 - mse: 2.7859e-04 - mae: 0.0101 - mape: 29.2342 - 17s/epoch - 28ms/step
Epoch 3/5
600/600 - 17s - loss: 2.1956e-04 - mse: 2.1956e-04 - mae: 0.0091 - mape: 27.3393 - 17s/epoch - 28ms/step
Epoch 4/5
600/600 - 17s - loss: 1.5258e-04 - mse: 1.5258e-04 - mae: 0.0075 - mape: 22.8954 - 17s/epoch - 28ms/step
Epoch 5/5
600/600 - 17s - loss: 1.2443e-04 - mse: 1.2443e-04 - mae: 0.0066 - mape: 20.0090 - 17s/epoch - 28ms/step

stock: nvidia
Epoch 1/5
600/600 - 20s - loss: 3.7637e-04 - mse: 3.7637e-04 - mae: 0.0131 - mape: 16.7528 - 20s/epoch - 33ms/step
Epoch 2/5
600/600 - 18s - loss: 1.6649e-04 - mse: 1.6649e-04 - mae: 0.0095 - mape: 11.2839 - 18s/epoch - 30ms/step
Epoch 3/5
600/600 - 18s - loss: 1.4078e-04 - mse: 1.4078e-04 - mae: 0.0088 - mape: 10.4044 - 18s/epoch - 29ms/step
Epoch 4/5
600/600 - 18s - loss: 1.1603e-04 - mse: 1.1603e-04 - mae: 0.0078 - mape: 8.7556 - 18s/epoch - 29ms/step
Epoch 5/5
600/600 - 18s - loss: 7.0279e-05 - mse: 7.0279e-05 - mae: 0.0059 - mape: 7.0997 - 18s/epoch - 29ms/step

stock: microsoft
Epoch 1/5
600/600 - 18s - loss: 0.0010 - mse: 0.0010 - mae: 0.0172 - mape: 982.1262 - 18s/epoch - 30ms/step
Epoch 2/5
600/600 - 16s - loss: 3.6037e-04 - mse: 3.6037e-04 - mae: 0.0132 - mape: 7807.4761 - 16s/epoch - 27ms/step
Epoch 3/5
600/600 - 16s - loss: 2.7456e-04 - mse: 2.7456e-04 - mae: 0.0120 - mape: 8677.3174 - 16s/epoch - 27ms/step
Epoch 4/5
600/600 - 16s - loss: 2.9133e-04 - mse: 2.9133e-04 - mae: 0.0122 - mape: 9035.0703 - 16s/epoch - 27ms/step
Epoch 5/5
600/600 - 17s - loss: 2.2217e-04 - mse: 2.2217e-04 - mae: 0.0097 - mape: 12716.9062 - 17s/epoch - 28ms/step

stock: amazon
Epoch 1/5
600/600 - 21s - loss: 0.0021 - mse: 0.0021 - mae: 0.0261 - mape: 9.4448 - 21s/epoch - 34ms/step
Epoch 2/5
600/600 - 18s - loss: 7.8839e-04 - mse: 7.8839e-04 - mae: 0.0204 - mape: 7.8306 - 18s/epoch - 31ms/step
Epoch 3/5
600/600 - 19s - loss: 6.4420e-04 - mse: 6.4420e-04 - mae: 0.0179 - mape: 6.5050 - 19s/epoch - 31ms/step
Epoch 4/5
600/600 - 18s - loss: 6.1013e-04 - mse: 6.1013e-04 - mae: 0.0179 - mape: 6.5101 - 18s/epoch - 31ms/step
Epoch 5/5
600/600 - 19s - loss: 3.1494e-04 - mse: 3.1494e-04 - mae: 0.0124 - mape: 4.5630 - 19s/epoch - 31ms/step

stock: amd
Epoch 1/5
600/600 - 22s - loss: 7.3601e-04 - mse: 7.3601e-04 - mae: 0.0166 - mape: 18.4526 - 22s/epoch - 36ms/step
Epoch 2/5
600/600 - 19s - loss: 3.9304e-04 - mse: 3.9304e-04 - mae: 0.0135 - mape: 14.6299 - 19s/epoch - 32ms/step
Epoch 3/5
600/600 - 19s - loss: 2.6095e-04 - mse: 2.6095e-04 - mae: 0.0109 - mape: 11.9050 - 19s/epoch - 32ms/step
Epoch 4/5
600/600 - 19s - loss: 1.8311e-04 - mse: 1.8311e-04 - mae: 0.0095 - mape: 11.2958 - 19s/epoch - 32ms/step
Epoch 5/5
600/600 - 19s - loss: 2.1652e-04 - mse: 2.1652e-04 - mae: 0.0103 - mape: 12.2141 - 19s/epoch - 32ms/step

stock: meta
Epoch 1/5
600/600 - 21s - loss: 0.0016 - mse: 0.0016 - mae: 0.0264 - mape: 10.8218 - 21s/epoch - 35ms/step
Epoch 2/5
600/600 - 19s - loss: 6.8228e-04 - mse: 6.8228e-04 - mae: 0.0193 - mape: 7.9754 - 19s/epoch - 31ms/step
Epoch 3/5
600/600 - 19s - loss: 5.4249e-04 - mse: 5.4249e-04 - mae: 0.0170 - mape: 7.1026 - 19s/epoch - 32ms/step
Epoch 4/5
600/600 - 19s - loss: 4.8820e-04 - mse: 4.8820e-04 - mae: 0.0155 - mape: 6.6058 - 19s/epoch - 32ms/step
Epoch 5/5
600/600 - 19s - loss: 4.0588e-04 - mse: 4.0588e-04 - mae: 0.0141 - mape: 5.8234 - 19s/epoch - 32ms/step
```

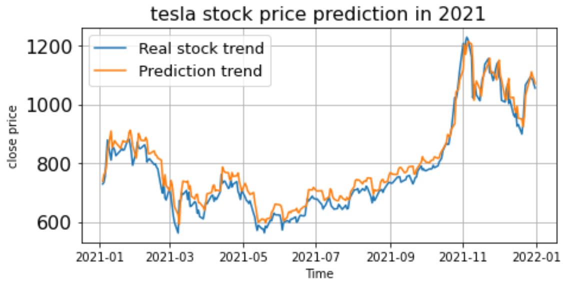
```
stock: adobe
Epoch 1/5
600/600 - 22s - loss: 0.0012 - mse: 0.0012 - mae: 0.0213 - mape: 11.4139 - 22s/epoch - 36ms/step
Epoch 2/5
600/600 - 18s - loss: 4.2385e-04 - mse: 4.2385e-04 - mae: 0.0138 - mape: 6.9611 - 18s/epoch - 31ms/step
Epoch 3/5
600/600 - 18s - loss: 4.0787e-04 - mse: 4.0787e-04 - mae: 0.0135 - mape: 7.6205 - 18s/epoch - 30ms/step
Epoch 4/5
600/600 - 19s - loss: 3.2594e-04 - mse: 3.2594e-04 - mae: 0.0127 - mape: 6.7868 - 19s/epoch - 31ms/step
Epoch 5/5
600/600 - 18s - loss: 2.9741e-04 - mse: 2.9741e-04 - mae: 0.0122 - mape: 6.5877 - 18s/epoch - 31ms/step
```

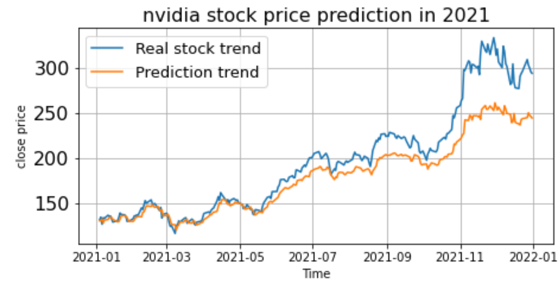
```
stock: google
Epoch 1/5
600/600 - 18s - loss: 7.3582e-04 - mse: 7.3582e-04 - mae: 0.0174 - mape: 11526.4346 - 18s/epoch - 30ms/step
Epoch 2/5
600/600 - 16s - loss: 3.0321e-04 - mse: 3.0321e-04 - mae: 0.0129 - mape: 1578.2689 - 16s/epoch - 27ms/step
Epoch 3/5
600/600 - 16s - loss: 2.4576e-04 - mse: 2.4576e-04 - mae: 0.0115 - mape: 2626.7542 - 16s/epoch - 27ms/step
Epoch 4/5
600/600 - 16s - loss: 1.6988e-04 - mse: 1.6988e-04 - mae: 0.0096 - mape: 3299.5220 - 16s/epoch - 27ms/step
Epoch 5/5
600/600 - 16s - loss: 1.4344e-04 - mse: 1.4344e-04 - mae: 0.0088 - mape: 6916.5415 - 16s/epoch - 27ms/step
```

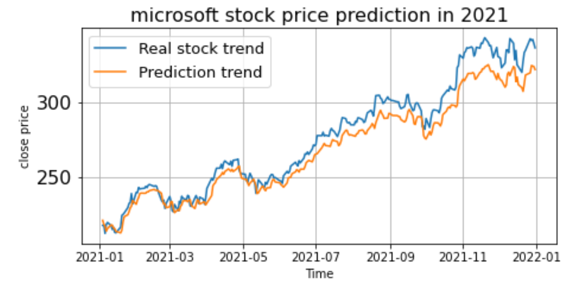
```
stock: pfizer
Epoch 1/5
600/600 - 19s - loss: 0.0013 - mse: 0.0013 - mae: 0.0259 - mape: 70845.0156 - 19s/epoch - 32ms/step
Epoch 2/5
600/600 - 18s - loss: 5.6845e-04 - mse: 5.6845e-04 - mae: 0.0175 - mape: 69657.6172 - 18s/epoch - 30ms/step
Epoch 3/5
600/600 - 18s - loss: 4.2903e-04 - mse: 4.2903e-04 - mae: 0.0153 - mape: 54870.9922 - 18s/epoch - 30ms/step
Epoch 4/5
600/600 - 18s - loss: 3.4077e-04 - mse: 3.4077e-04 - mae: 0.0136 - mape: 41080.0391 - 18s/epoch - 30ms/step
Epoch 5/5
600/600 - 19s - loss: 2.8643e-04 - mse: 2.8643e-04 - mae: 0.0124 - mape: 30320.0391 - 19s/epoch - 31ms/step
```

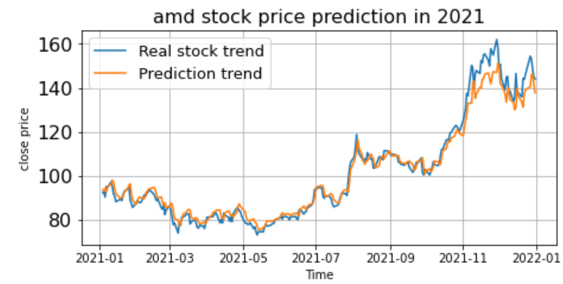
For Z-score standardization version:

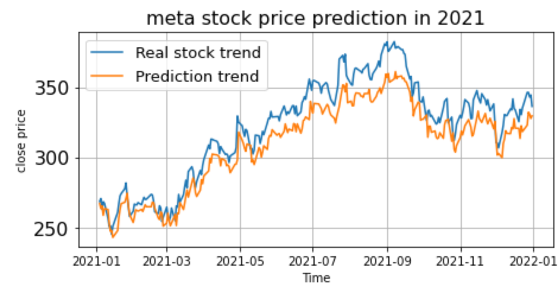


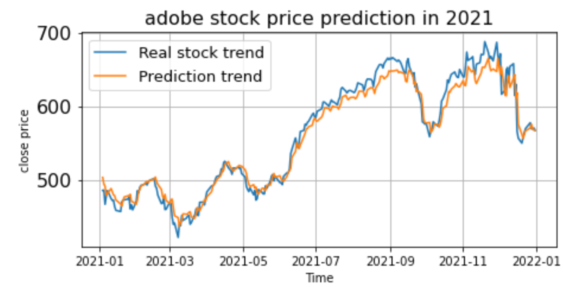


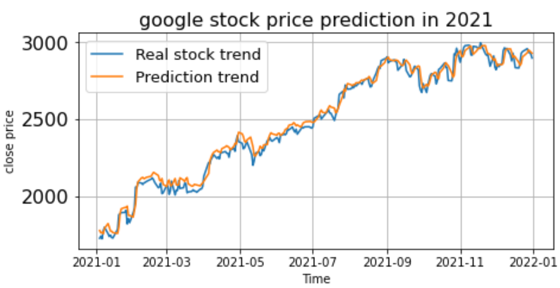


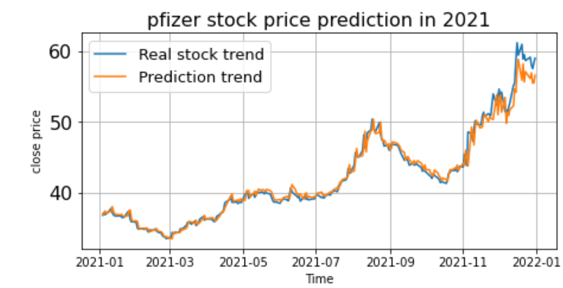












Training information:

```
stock: apple
Epoch 1/5
600/600 - 18s - loss: 0.0102 - mse: 0.0102 - mae: 0.0597 - mape: 36.3141 - 18s/epoch - 30ms/step
Epoch 2/5
600/600 - 16s - loss: 0.0040 - mse: 0.0040 - mae: 0.0424 - mape: 26.6605 - 16s/epoch - 27ms/step
Epoch 3/5
600/600 - 16s - loss: 0.0033 - mse: 0.0033 - mae: 0.0379 - mape: 30.4256 - 16s/epoch - 26ms/step
Epoch 4/5
600/600 - 16s - loss: 0.0023 - mse: 0.0023 - mae: 0.0333 - mape: 25.2528 - 16s/epoch - 26ms/step
Epoch 5/5
600/600 - 16s - loss: 0.0023 - mse: 0.0023 - mae: 0.0322 - mape: 22.3905 - 16s/epoch - 26ms/step

stock: tesla
Epoch 1/5
600/600 - 19s - loss: 0.0097 - mse: 0.0097 - mae: 0.0511 - mape: 16.6127 - 19s/epoch - 32ms/step
Epoch 2/5
600/600 - 17s - loss: 0.0040 - mse: 0.0040 - mae: 0.0360 - mape: 13.7260 - 17s/epoch - 28ms/step
Epoch 3/5
600/600 - 17s - loss: 0.0025 - mse: 0.0025 - mae: 0.0281 - mape: 9.5576 - 17s/epoch - 28ms/step
Epoch 4/5
600/600 - 17s - loss: 0.0020 - mse: 0.0020 - mae: 0.0240 - mape: 11.1459 - 17s/epoch - 28ms/step
Epoch 5/5
600/600 - 17s - loss: 0.0018 - mse: 0.0018 - mae: 0.0250 - mape: 9.9554 - 17s/epoch - 28ms/step

stock: nvidia
Epoch 1/5
600/600 - 18s - loss: 0.0053 - mse: 0.0053 - mae: 0.0483 - mape: 24.8781 - 18s/epoch - 30ms/step
Epoch 2/5
600/600 - 16s - loss: 0.0025 - mse: 0.0025 - mae: 0.0367 - mape: 16.0091 - 16s/epoch - 26ms/step
Epoch 3/5
600/600 - 16s - loss: 0.0018 - mse: 0.0018 - mae: 0.0306 - mape: 15.5696 - 16s/epoch - 27ms/step
Epoch 4/5
600/600 - 16s - loss: 0.0019 - mse: 0.0019 - mae: 0.0301 - mape: 14.1877 - 16s/epoch - 27ms/step
Epoch 5/5
600/600 - 16s - loss: 0.0013 - mse: 0.0013 - mae: 0.0259 - mape: 12.1182 - 16s/epoch - 26ms/step

stock: microsoft
Epoch 1/5
600/600 - 22s - loss: 0.0071 - mse: 0.0071 - mae: 0.0511 - mape: 64.7753 - 22s/epoch - 37ms/step
Epoch 2/5
600/600 - 19s - loss: 0.0027 - mse: 0.0027 - mae: 0.0356 - mape: 48.5035 - 19s/epoch - 32ms/step
Epoch 3/5
600/600 - 19s - loss: 0.0024 - mse: 0.0024 - mae: 0.0349 - mape: 35.4592 - 19s/epoch - 32ms/step
Epoch 4/5
600/600 - 19s - loss: 0.0018 - mse: 0.0018 - mae: 0.0296 - mape: 36.8576 - 19s/epoch - 32ms/step
Epoch 5/5
600/600 - 19s - loss: 0.0022 - mse: 0.0022 - mae: 0.0323 - mape: 38.9752 - 19s/epoch - 32ms/step

stock: amazon
Epoch 1/5
600/600 - 20s - loss: 0.0113 - mse: 0.0113 - mae: 0.0704 - mape: 160.0211 - 20s/epoch - 33ms/step
Epoch 2/5
600/600 - 19s - loss: 0.0056 - mse: 0.0056 - mae: 0.0547 - mape: 96.6085 - 19s/epoch - 32ms/step
Epoch 3/5
600/600 - 17s - loss: 0.0037 - mse: 0.0037 - mae: 0.0443 - mape: 101.3523 - 17s/epoch - 29ms/step
Epoch 4/5
600/600 - 17s - loss: 0.0034 - mse: 0.0034 - mae: 0.0420 - mape: 69.6953 - 17s/epoch - 29ms/step
Epoch 5/5
600/600 - 17s - loss: 0.0026 - mse: 0.0026 - mae: 0.0377 - mape: 58.8374 - 17s/epoch - 28ms/step

stock: amd
Epoch 1/5
600/600 - 21s - loss: 0.0091 - mse: 0.0091 - mae: 0.0616 - mape: 33.2783 - 21s/epoch - 35ms/step
Epoch 2/5
600/600 - 19s - loss: 0.0039 - mse: 0.0039 - mae: 0.0428 - mape: 26.1481 - 19s/epoch - 32ms/step
Epoch 3/5
600/600 - 19s - loss: 0.0034 - mse: 0.0034 - mae: 0.0416 - mape: 21.6292 - 19s/epoch - 32ms/step
Epoch 4/5
600/600 - 19s - loss: 0.0064 - mse: 0.0064 - mae: 0.0473 - mape: 20.9756 - 19s/epoch - 32ms/step
Epoch 5/5
600/600 - 19s - loss: 0.0021 - mse: 0.0021 - mae: 0.0309 - mape: 16.8560 - 19s/epoch - 32ms/step

stock: meta
Epoch 1/5
600/600 - 21s - loss: 0.0151 - mse: 0.0151 - mae: 0.0831 - mape: 83.9920 - 21s/epoch - 36ms/step
Epoch 2/5
600/600 - 19s - loss: 0.0072 - mse: 0.0072 - mae: 0.0601 - mape: 53.0308 - 19s/epoch - 31ms/step
Epoch 3/5
600/600 - 19s - loss: 0.0054 - mse: 0.0054 - mae: 0.0505 - mape: 45.6644 - 19s/epoch - 31ms/step
Epoch 4/5
600/600 - 19s - loss: 0.0047 - mse: 0.0047 - mae: 0.0473 - mape: 33.9434 - 19s/epoch - 31ms/step
Epoch 5/5
600/600 - 19s - loss: 0.0042 - mse: 0.0042 - mae: 0.0438 - mape: 29.0464 - 19s/epoch - 31ms/step
```

```
stock: adobe
Epoch 1/5
600/600 - 21s - loss: 0.0111 - mse: 0.0111 - mae: 0.0628 - mape: 56.0009 - 21s/epoch - 36ms/step
Epoch 2/5
600/600 - 19s - loss: 0.0038 - mse: 0.0038 - mae: 0.0443 - mape: 41.5954 - 19s/epoch - 32ms/step
Epoch 3/5
600/600 - 19s - loss: 0.0032 - mse: 0.0032 - mae: 0.0402 - mape: 32.3600 - 19s/epoch - 32ms/step
Epoch 4/5
600/600 - 19s - loss: 0.0031 - mse: 0.0031 - mae: 0.0395 - mape: 32.1483 - 19s/epoch - 32ms/step
Epoch 5/5
600/600 - 20s - loss: 0.0025 - mse: 0.0025 - mae: 0.0357 - mape: 30.2193 - 20s/epoch - 33ms/step

stock: google
Epoch 1/5
600/600 - 23s - loss: 0.0063 - mse: 0.0063 - mae: 0.0541 - mape: 23.8374 - 23s/epoch - 39ms/step
Epoch 2/5
600/600 - 20s - loss: 0.0029 - mse: 0.0029 - mae: 0.0404 - mape: 19.6571 - 20s/epoch - 34ms/step
Epoch 3/5
600/600 - 20s - loss: 0.0020 - mse: 0.0020 - mae: 0.0331 - mape: 15.3031 - 20s/epoch - 34ms/step
Epoch 4/5
600/600 - 20s - loss: 0.0018 - mse: 0.0018 - mae: 0.0309 - mape: 14.9236 - 20s/epoch - 34ms/step
Epoch 5/5
600/600 - 21s - loss: 0.0016 - mse: 0.0016 - mae: 0.0295 - mape: 14.2126 - 21s/epoch - 34ms/step

stock: pfizer
Epoch 1/5
600/600 - 22s - loss: 0.0296 - mse: 0.0296 - mae: 0.1237 - mape: 51.6115 - 22s/epoch - 37ms/step
Epoch 2/5
600/600 - 20s - loss: 0.0159 - mse: 0.0159 - mae: 0.0896 - mape: 35.7305 - 20s/epoch - 33ms/step
Epoch 3/5
600/600 - 20s - loss: 0.0130 - mse: 0.0130 - mae: 0.0808 - mape: 33.0170 - 20s/epoch - 34ms/step
Epoch 4/5
600/600 - 20s - loss: 0.0113 - mse: 0.0113 - mae: 0.0747 - mape: 32.2146 - 20s/epoch - 34ms/step
Epoch 5/5
600/600 - 20s - loss: 0.0111 - mse: 0.0111 - mae: 0.0744 - mape: 32.6849 - 20s/epoch - 33ms/step
```

7.4 Deep learning algorithms performance analysis:

According to the two data preprocessing methods of performance, the predictive results are impressive as it is clear to see all the stock trends are perfectly modeled from the line charts. For the predictive and real stock trend, the local lowest and highest points often can be simulated in different time points. Next, let's see the different kinds of error estimators and investigate the LSTM neural network performance. Summing up the estimators from python, the following table shown below:

For Min-Max version:

	Stock	MSE	RMSE	MAE	MAPE	R_square
0	apple	11.144678	3.338365	2.685989	1.876769	0.947934
1	tesla	5378.092145	73.335477	65.715163	8.612097	0.792256
2	nvidia	50.763754	7.124869	5.129211	2.595985	0.985212
3	microsoft	39.153159	6.257249	5.066859	1.780356	0.971193
4	amazon	4897.469498	69.981923	54.763189	1.621786	0.808031
5	amd	12.799280	3.577608	2.769854	2.744024	0.976553
6	meta	61.054279	7.813724	6.484155	2.011690	0.949541
7	adobe	220.753661	14.857781	12.195509	2.175975	0.961810
8	google	14666.937289	121.107131	113.383595	4.533885	0.890456
9	pfizer	0.812234	0.901240	0.616939	1.382040	0.980374

For Z-score standardization:

	Stock	MSE	RMSE	MAE	MAPE	R_square
0	apple	25.676372	5.067186	4.177845	2.843754	0.880045
1	tesla	1435.679990	37.890368	31.898525	4.272950	0.944543
2	nvidia	689.075341	26.250245	17.857082	7.466634	0.799268
3	microsoft	96.492917	9.823081	8.113122	2.772501	0.929006
4	amazon	29868.222131	172.824252	163.178834	4.836443	-0.170762
5	amd	17.693398	4.206352	2.917686	2.715596	0.967588
6	meta	186.034809	13.639458	12.025297	3.633141	0.846250
7	adobe	185.848095	13.632611	10.744936	1.878197	0.967849
8	google	1861.774626	43.148286	34.349972	1.443400	0.986095
9	pfizer	1.075378	1.037004	0.647624	1.405196	0.974016

Generally, the min-max normalization version of deep learning algorithm performs better. As the summary table shown, both two version of all error estimates and the R^2 has huge differences in value. Therefore, min-max normalization is more suitable for preprocessing the data in advance.

8. Limitations on AI algorithms and its future development

All the AI algorithms in this project are demonstrated and its applications of performance could be very impressive in the stock market. Especially in deep learning field, the LSTM neural network has perfectly simulated the stock trend on its unseen data section and the R^2 normally high which can largely explain the variation. For the adjustments part, both algorithms in classification and regression exists some limitations in the performance. As it is clearly to observe that the validation accuracy of classifiers in machine learning hugely variates. Although the model accuracy can achieve 100%, the best model is chosen given the observed validation accuracy trend every stock and its prediction period may not be sufficient to generalize. Assume the validation period extends to be wider, it involves more investigations on model analysis in order to discover the characteristics of best performance. What's more, the LSTM may bring the expensive costs for training. From reviewing the training information, it takes around one and a half of minutes to finish the training in one stock. That's for ten stocks in our project, it could be spending 15 minutes to finish the training. More importantly, with the non-deterministic setting of algorithms in python, the LSTM would have different output results based on the randomness of initialization. To obtain the optimal result, the major task is to tune the hyper parameters given the situation to find out the better performance. Identifying the ideal model for generalization is the huge process and also the important future development of the AI algorithms.

9. Summary of the capstone project

To sum up, the AI algorithms has been demonstrated in the US stock market and its applications could be assisting to produce the valuable investment information. Recall the first main section, the dependency measurement in every stock, leverages the correlation and mutual information concepts to reveal the strong stock price movements of pairs and the degree of close price association in time series format. The correlation result helps the investor understand the how strong of the linear relationship in particular stocks. This may benefit the investors could have the same strategy in those related stocks. The mutual information analysis not only illustrated the stock price relations of calendar date but also the past days relations. The investor may refer to the result and have an attention on the price movement in the key time period. Dependency analysis gave the concise understanding of every stock movement in 2021. For the AI algorithm part, the machine learning classifiers showed its performance in stock price up and down classification. It would help investors for the stock buy/sell decision. The deep learning LSTM neural network successfully simulated the stock price trend. To conclude that, the AI applications in stock market has always been having a huge impact and the great potential development area would largely attract more and more scholars to do the AI research in stock. It could be foreseen that the AI technologies in stock would be the mainstream powerful stock analysis tools. Last but not least, this project has shown some AI applications in stock market and brought the main ideas of AI algorithm in stock market.

Reference

Chollet, F. (2017). *Deep learning with Python*. Manning Publications.

Columbus, L. (2020, October 31). *The State Of AI Adoption In Financial Services*. Forbes. https://www.forbes.com/sites/louiscolumbus/2020/10/31/the-state-of-ai-adoption-in-financial-services/?sh=7ad924b12aac

Gandhi, R. (2018, July 5). *Support vector machine — Introduction to machine learning algorithms*. TowardsDataScience. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Géron, A. (2017). *Hands-on machine learning with scikit-learn and TensorFlow: Techniques and tools to build learning machines*. O'Reilly Media.

Irina, R. (2001). An Empirical Study of the Naïve Bayes Classifier. https://www.researchgate.net/publication/228845263_An_Empirical_Study_of_the_Naive_Bayes_Classifier

Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: A guide for data scientists*. O'Reilly Media.

Nello, C., & John, S. T. (2013). Support vector machines. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. https://doi.org/10.1017/cbo9780511801389.008

Phaneuf, A. (2022, February 8). *Artificial intelligence in financial services: Applications and benefits of AI in finance*. Business Insider. https://www.businessinsider.com/ai-in-finance

Reddy, D. M., Reddy, K. A., Babu, H. V., & Saileela, Y. (n.d.). *Stock Market Analysis using LSTM in Deep Learning*. http://dx.doi.org/10.17577/IJERTV9IS040649

Savvas, V., Koen, B., & Carmen, A. G. (2018). Neural Networks for Fast Surface Code Decoding. *Designing neural network based decoders for surface codes*.

Tharwat, A., Gaber, T., Ibrahim, A., & Hassanien, A. E. (2017). Linear discriminant analysis: A detailed tutorial. *AI Communications*, *30*(2). https://doi.org/10.3233/aic-170729

Yates, D., Islam, M. Z., & Gao, J. (2019). SPAARC: A fast decision tree algorithm. *Communications in Computer and Information Science*. https://doi.org/10.1007/978-981-13-6661-1_4