RMSC 4002

Data Analysis for Finance and Risk Management Science

Course Project

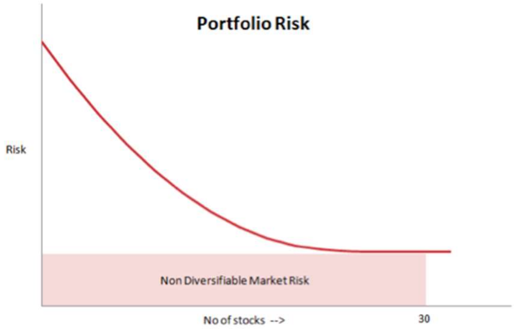
An analysis on the performance of Decision Tree and Artificial Neural Network on predicting stock market

Name	Student ID
CHENG Wing Ryan	1155102964
Ip Man Fai	1155115622

Table of Contents

Introduction

Predicting the stock market is always an interesting topic to explore. In the 20th century, there are so many classical methods that implement statistical knowledge implicitly such as technical indicators. And nowadays, the machine learning algorithm seems to be one of the most popular methods to predict the stock market. In this report, we are going to classify the rise and fall of the Dow Jones index by classification tree and ANN, and compare the performance of both algorithms and figure out which is better. Afterward, we will use ANN to predict the index instead of classifying its position. The first question that people may ask is, why index instead of stocks? Where risk and diversification can explain this question well. The index is like a basket of different stocks, increasing the number of stocks to the basket can reduce the risk of some particular stock. In this project, the Dow Jones Index, which is one of the most popular indexes in the world, is chosen for our analysis.



1. Dataset Description

1.1 Stock and index Selection

Throughout this project, some stocks or indexes are selected for obtaining the related variables. In terms of stock, Dow 30 are the stocks that are the component of the Dow Jones Index so we believe we can get some information about the Dow Jones Index. In terms of index, we consider the GSPC, IXIC, RUT, and VIX which are the Index from America, thus we also believe there should be a relationship with Dow Jones Index.

Therefore, we favorably select those stocks and indexes for our investigation. We have imported the daily adjusted close/open/high/low prices and volume by the built-in *get.hist.quote* function from R package *tseries* from 1/2/2010 to 31/12/2018.

1.2 Response

The response categorical variable represents the position compared to the adjusted close price of the present day and the end day of each month i.e. Increasing Position or Decreasing Position. Because it is hard for us to deal with the daily position so we use the past 30 days to predict the end of each month.

$$\mathrm{Y} = \mathbb{I}\left(\mathrm{P}_{\mathrm{iDay,jMonth,wYear}} > \mathrm{P}_{\mathrm{iEnd,jMonth,wYear}}\right),$$

$$\mathrm{iDay} \in \{1,\cdots,31\}, \mathrm{iEnd} \in \{28,\cdots,31\},$$

$$\mathrm{jMonth} \in \{1,\cdots,12\}, \mathrm{wYear} \in \{2010,\cdots,2018\},$$

1.3 Explanatory Variables

a) Daily Return

Return, or the daily return, is the relative change in stock price today as compared to the stock price yesterday.

.

$$\text{Return } \mathrm{R_t} = \frac{P_t - P_{t-1}}{P_{t-1}}$$

The positive return represents there is an increase in stock price today compared to the stock price yesterday.

b) Inverse of coefficient of variation

Average Price is a measure of the average stock price over a while. It is defined as the previous n days.

$$\text{Average Pirce} = \frac{1}{n}\sum_{t=0}^{n-1} \mu_t$$

Standard Deviation of Price, or volatility, is a measure for the variation of the stock price over a while. It is defined as the standard deviation of the stock price in previous n days.

$$\text{Standard Deviation of Pice} = \sqrt{\frac{1}{n-1}\sum_{t=0}^{n-1}\left(P_t - \overline{P}\right)^2}$$

The Higher standard deviation represents the average price is not that representative because the daily Price is far away from the average.

The inverse of the coefficient of variation is defined as a ratio of average price to the variance of price. It is the standardized measures of inverse dispersion of the past 30 days Price.

$$\text{Inverse of coefficient of variation } = \frac{\mu_t}{\sigma_t^2}$$

c) Others

We try to get more information from the close/open/high/low prices and volume, then we treat that information yesterday as the explanatory variables.

2. Model Analysis

2.1 Classification Tree

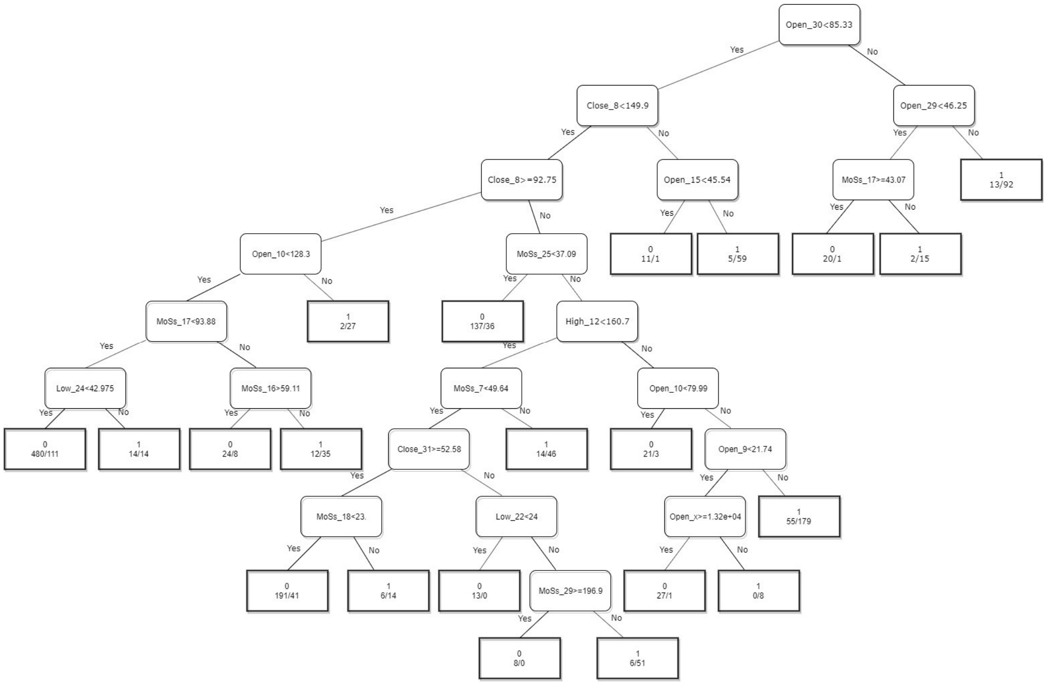
Methodology

To begin with, we apply the decision tree model to explain the position and the other variables from the Index and Dows 30 stock markets. Loosely speaking, the classification tree is a method to generate a set of simple rules that can be applied to classify the observations. This method firstly builds up a classification tree based on the binary splitting of variables following the algorithm which is required the terminal node is as pure as possible by the impurity measure. The goal of the algorithm is choosing the best variable for splitting. In terms of R programming, there is a built-in library rpart that can be used to implement the decision tree model.

Result

We select a random subset from the dataset as the in-sample data and develop a decision tree model. The rest of the data will be treated as out-sample data and used to test the model.

Model 1



From the result, the classification tree model splits into 41 nodes. Thus, we obtain 21 rules from the classification tree model.

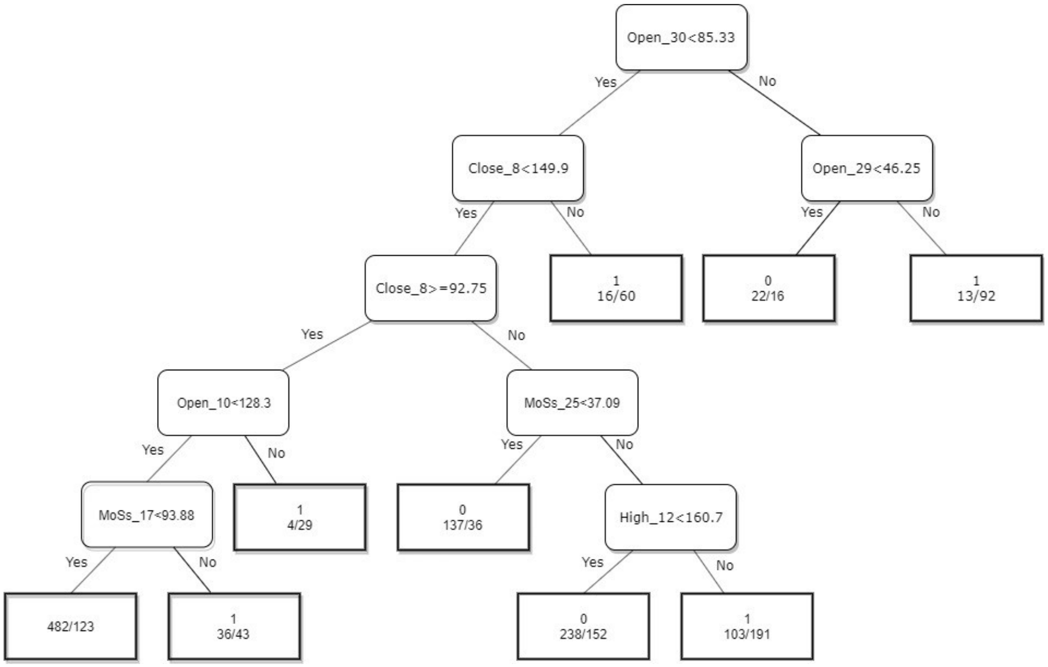
The corresponding misclassification rate are:

In-sample data (misclassification rate = 17.90%)			Out-sample data (misclassification rate = 24.22%)		
	0	1		0	1
0	932	202	0	204	63
1	119	540	1	45	136

The misclassification rate of the training data is small while the misclassification rate of the testing data is quite large because the classification tree model contains some dummy nodes which causes overfitting. Some child nodes even contain 100% confidence with a small group of data. Therefore, we try to prune the decision tree.

Model 2 (Pre-pruning)

Pre-pruning is a step that sets some specific early stopping criteria before developing the classification tree. The Criteria are set as parameter values while building the model. The tree stops growing when it meets any of these pre-pruning criteria: (1) maximum depth of a tree which is the length of the longest path from a root node to a Leaf node; (2) minimum number of records that must exist in a node for a split to happen or be attempted; (3) minimum number of records that can be present in a Terminal node.



From the result, the classification tree model splits into 17 nodes. Thus, we obtain 9 rules from the classification tree model.

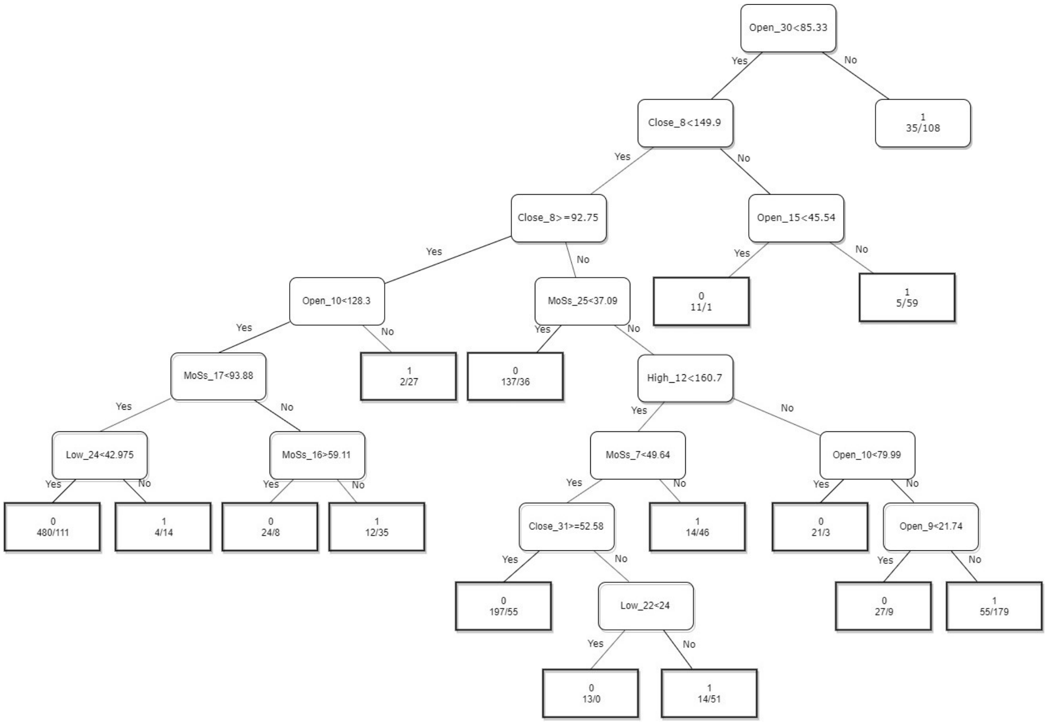
The corresponding misclassification rate are:

In-sample data (misclassification rate = 27.83%)			Out-sample data (misclassification rate = 31.92%)		
	0	1		0	1
0	879	327	0	200	94
1	172	415	1	49	105

In terms of the above decision tree model, we decide to limit the maximum length number from the Root node to the leaf nodes as 5 and the minimum record number results in each node as 100. After pruning the tree, the misclassification rate of training and testing data rises rapidly while the difference between both rates is decreasing.

Model 3 (Post-pruning)

Post-pruning is a useful manner to allow the decision tree to grow fully and observe the optimal CP(complexity parameter) value which is used to control tree growth. Once the cost of adding a variable is higher than the value of CP, the tree will keep the remaining part and stop growing.



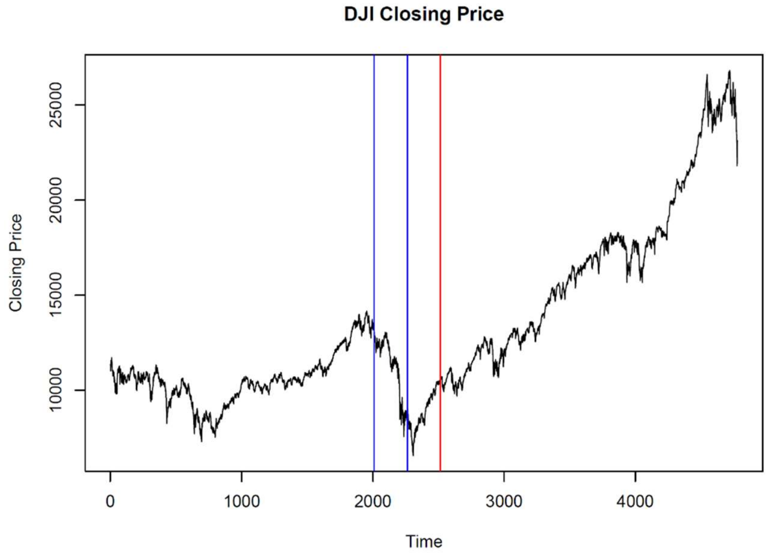
For the above classification tree model, we decide to choose the optimal CP value as 0.013 from the CP plot. From the result, the classification tree model splits into 31 nodes. Thus, we obtain 16 rules from the classification tree model.

The corresponding misclassification rate are:

In-sample data (misclassification rate = 20.30%)			Out-sample data (misclassification rate = 25.67%)		
	0	1		0	1
0	910	223	0	202	68
1	141	519	1	47	131

Although we cut the tree from 41 to 31 nodes, the massification rate of training and test data do not increase rapidly. Besides, the difference between both rates decreases.

Time difference



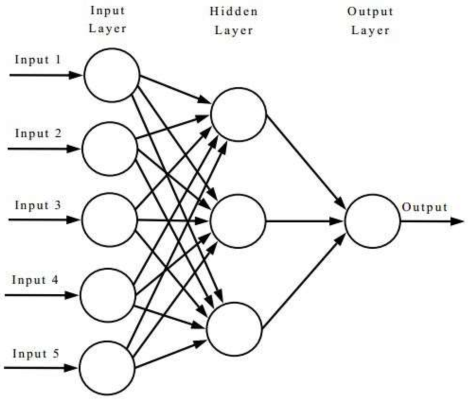
We try to use the past 18 years from 2000 to 2018 as out-sample data to test the classification tree model after post-pruning (Model 3). We obtain the misclassification rate which is around 34.5% which is much larger than the original test data. Apart from the other reasons, the main reason is the time series model is different from the existence of change point. From the time series plot, we can observe the time series pattern is different before and after the start of 2010, the red line. Before the start of 2010, the trend of the price is moving as a cycle surrounding the average, however, after the start of 2010, the trend rapidly increases. One of big issues in history, financial crisis, occurs in 2018, between the blue lines. After the end of the financial crisis, then the price index initially recovers. Therefore, the classification tree should carefully consider the time impact while testing or training the model.

Conclusion in CTREE

Since the number of variables is large, the decision tree model is overfitting in the beginning. After we prune the tree with two different algorithms, the decision tree model then turns towards fair. Unfortunately, however, we cannot drop the misclassification rate as small as possible. The reason behind is that the variables are not pure and significant to sufficiently explain the response which causes the model using more variables. Thus, the misclassification rate of the model increases in spite of that we prune the tree properly. It is a possible way to search the frequency of some specified keywords from celebrities on twitter which is one of the factors that influence people buying or selling the stocks in further study.

2.2 Artificial Neural Networks

Methodology



Artificial neural networks (ANN) is a statistical methodology to recognize the correlations between the parameters of a given independent variable and its response variable. This method is achieved by R's package nnet. ANN simulates the human brain and it is divided into three parts, i.e. Input layer, Hidden layers, and Output layer. Input Layer is the layer that accepts inputs, each neuron receives only one input; the output layer is the layer that produces output and the output of each neuron directly goes outside; the Hidden layer is the layer that connects input and output layers. This figure is a typical 5-3-1 ANN.

Since ANN can output both categorical results and continuous results, thus our initial approach is to predict whether the index of the last day of the month raise or fall compares to the index of the last day of last month. An example is illustrated as follow:

date	index	Up/Down
2010-03-01	10403.79	Up
2010-03-02	10405.98	Up
…..	…..	…..
2010-03-23	10888.83	Down
…..	…..	…..

2010-03-29	10895.86	Down
…..	…..	…..
2010-03-31	10856.63	NA

All daily data except the last day of the month are considered to predict whether the index of the last day of the month rises or falls on those particular days. The further approach is to predict the actual of the index of the last day of the month.

Problems and solutions

There are two major problems occurred when we use ANN. The first problem is the size of the hidden layer, and the second problem is the optimization of ANN.

The naïve solution of the first problem is to test the performance of ANN when we change the size of the hidden layer range from (2, 2(number of inputs)). However, due to the limitation of computational ability, this report will only test the size of the hidden layer up to 12.

The error function of ANN is $\mathrm{E}(\mathrm{w}_{\mathrm{ij}}) = [Y - V(w_{ij})]'[Y - V(w_{ij})]$, where w_{ij} is the weight of ANN, and our goal is to minimize the error function. However, one major problem we faced is that we may only found the local minima of the error function instead of global minima. Thus, stochastic gradient descent (SGD) is introduced. We train numbers of ANNs and found the model with the highest accuracy when predicting whether the index rise or fall, and when predicting the actual price, the model with the least $\sum(|\text{predict price} - \text{actual price}|)$ will be chosen.

Results

Initial approach – classification problem

Data cleaning

In this report, there is no machine technique is applied to reduce the noise or reduce the dimensions. Nonetheless, $\frac{\mu_{20days}}{\sigma_{20days}}$ of the closing price of different indexes (i.e. DJI, GSPC, IXIC, RUT, VIX) are included. A standardization method $\frac{\mathrm{x-min}}{max-min}$ has also been applied to all input variables.

Performance of ANN fitting

Take an arbitrary ANN model and see how it fits with the whole set of data.

	Down	Up
Down	914	16
Up	9	1265
Accuracy = 0.988657		

As we can see, ANN fits itself pretty well. It is interesting to see whether ANN can maintain its performance when fitting the testing data.

	Down	Up
Down	134	31
Up	51	226
Accuracy = 0.8144796		

We can see the performance drops quite a lot, but it is still in an acceptable range, it means although ANN fits itself well, overfit problem still doesn't exist.

Choosing the size of the hidden layer

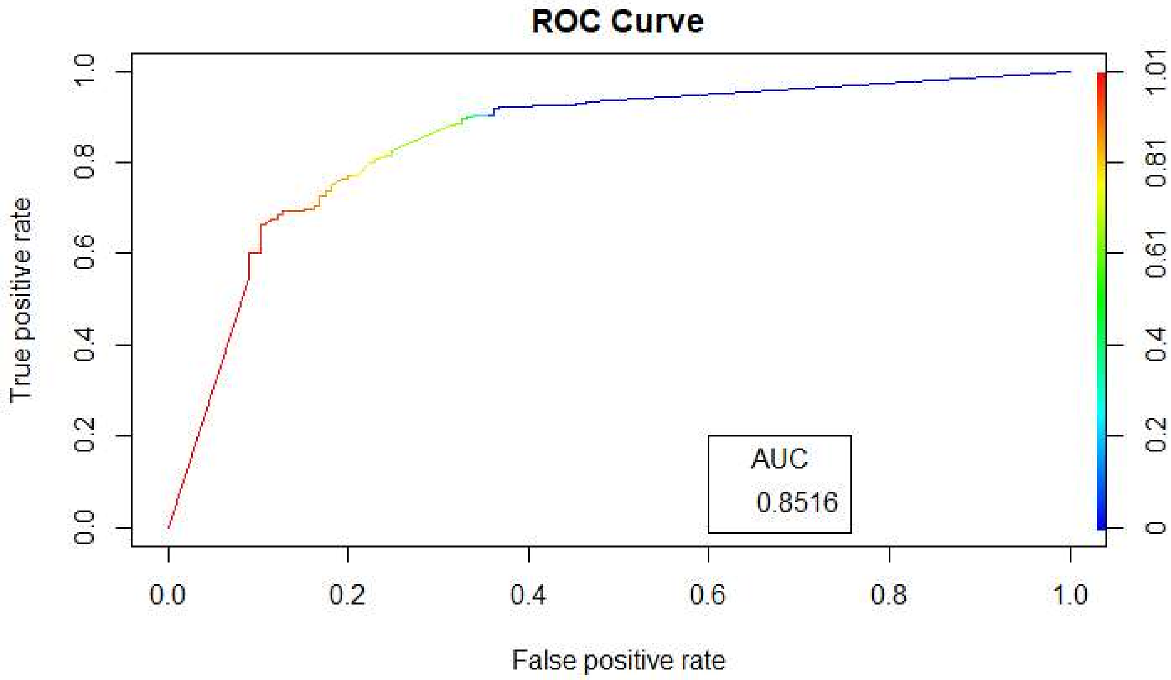
The following tables are the confusion matrix of the predicted accuracy of size 2 and size 12. ANN with a size 2 hidden layer seems no big difference compared to a size 12 hidden layer ANN. As a result, instead of abusing computation ability, i.e. choosing an ANN with a hidden layer with size 12, a size 2 ANN may be better.

Size 2	Down	Up
Down	119	44
Up	47	229
Accuracy = 0.7927107		

Size 12	Down	Up
Down	128	34

Up	38	239
Accuracy = 0.8359909		

ROC curve has shown the performance of ANN in classifying whether the index will go up or down. AUC with 0.8516 seems to perform quite well.



However, it is not the case. The following table shows the distribution of the accuracy in terms of different sizes of the hidden layers. There is a positive trend between the size of the hidden layer increase and the accuracy. But when we condition on the var component, we can see that a size 11 hidden layer ANN has the smallest variance, and then we soon figured out that a size 11 hidden layer ANN produced the best result in terms of the classification problem.

n = 50	size_2	size_3	size_4	size_5	size_6	size_7
mean	0.692528	0.723417	0.751435	0.760592	0.75959	0.774442
var	0.004612	0.003188	0.001987	0.001606	0.002071	0.001021
max	0.792711	0.8041	0.824601	0.822323	0.817768	0.822323
min	0.378132	0.544419	0.574032	0.589977	0.558087	0.685649
q1	0.668565	0.700456	0.740888	0.73861	0.75	0.76082
q2	0.703872	0.71754	0.76082	0.766515	0.768793	0.783599
q3	0.72836	0.762528	0.776765	0.790433	0.785877	0.794989

n = 50	size_8	size_9	size_10	size_11	size_12
mean	0.781458	0.771754	0.792073	0.799954	0.801093
var	0.000377	0.00426	0.000448	0.000285	0.000359
max	0.826879	0.826879	0.838269	0.840547	0.835991
min	0.733485	0.378132	0.728929	0.767654	0.76082
q1	0.769932	0.767654	0.779613	0.788724	0.789294
q2	0.781321	0.783599	0.79385	0.797267	0.8041
q3	0.797267	0.801253	0.805809	0.810364	0.81492

Interpretation of coefficient

ANN is known as a ‘black box’ model, as it connects brunch of layers and we cannot interpret all coefficient. It is possible to identify which coefficient contributes the most and which contributes the least in a single layer, but it is difficult to interpret. Thus, this report will not interpret the coefficient of ANN.

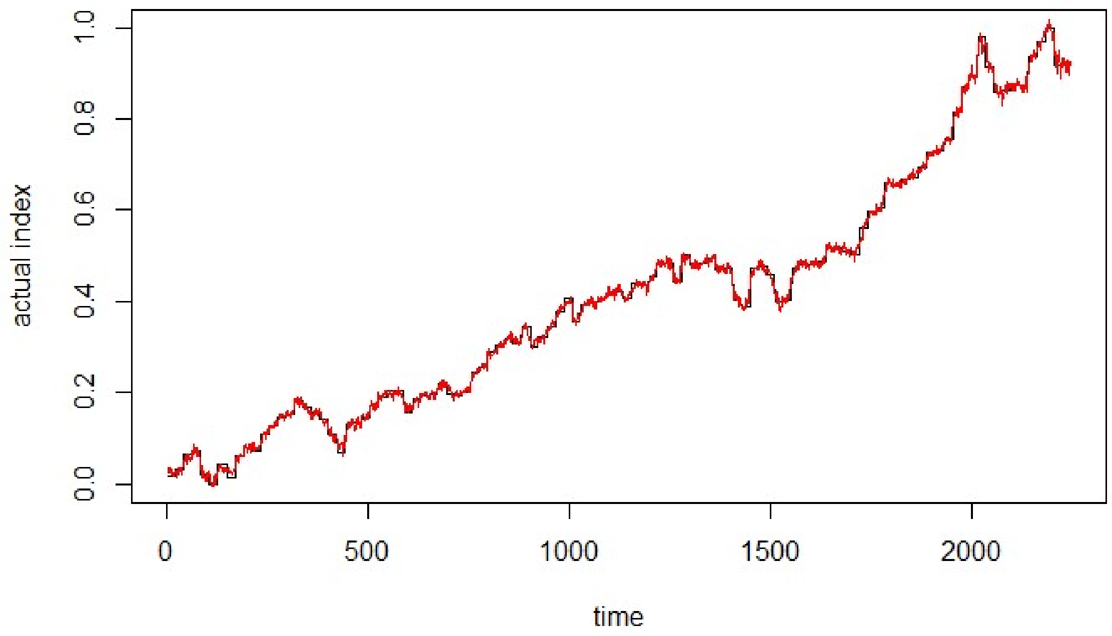
Further approach – predict the continuous result

Data cleaning

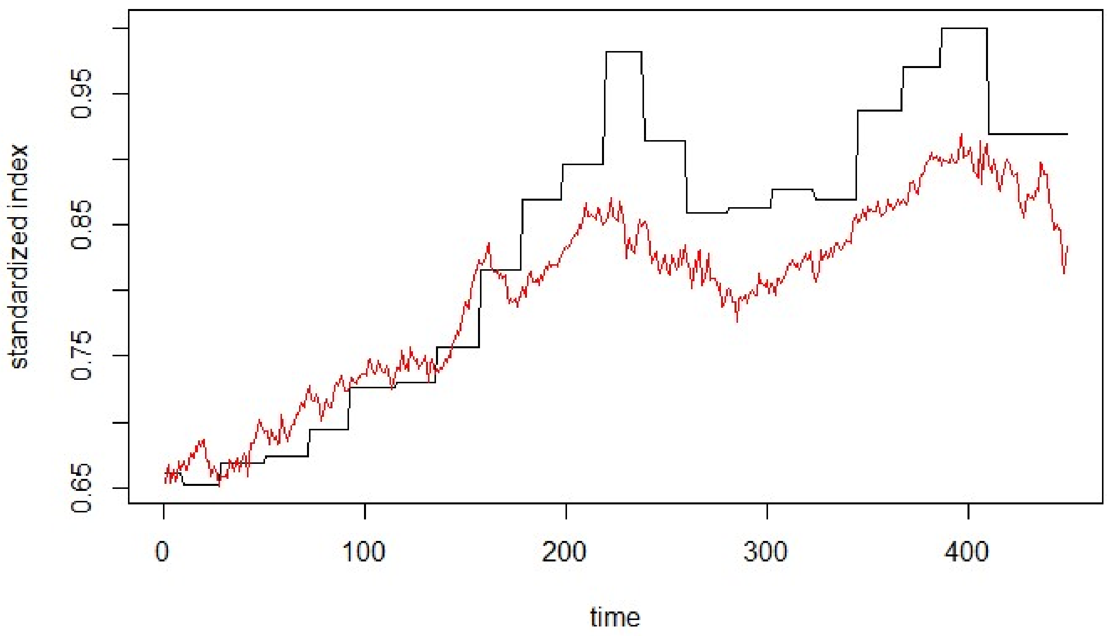
The data used in the second approach is the same as the initial approach. The only difference is the response variable changed from the position of the index of the last day of the month to the price of the last day of the month.

Performance of ANN fitting itself

Take an arbitrary ANN model and the error with the test data and its fitted value is 17.24332, after an un-standardization the error in terms of the index is 297466.6. This is quite a favorable result, ANN seems doing well in terms of predicting the real index.



It is interesting to see the performance of ANN perform on test data. Again, an arbitrary model has been trained and the performance is as follows.



It predicts quite well at the beginning, but then the error increased to break the tolerance level, ANN predicts badly afterward. However, one good news is that ANN can still predict the trend, i.e. up and down, and this effect is already been discussed in the previous part.

Choosing the size of the hidden layer

The following table shows the distribution of the error of the predicted price and the same price by using the formula $\text{error} = \sum(|\text{predict price} - \text{actual price}|)$. It is quite hard to see the trend between the size of hidden layer and the error. A size 9 hidden layer ANN seems can produce the best model consistently by looking at its variance.

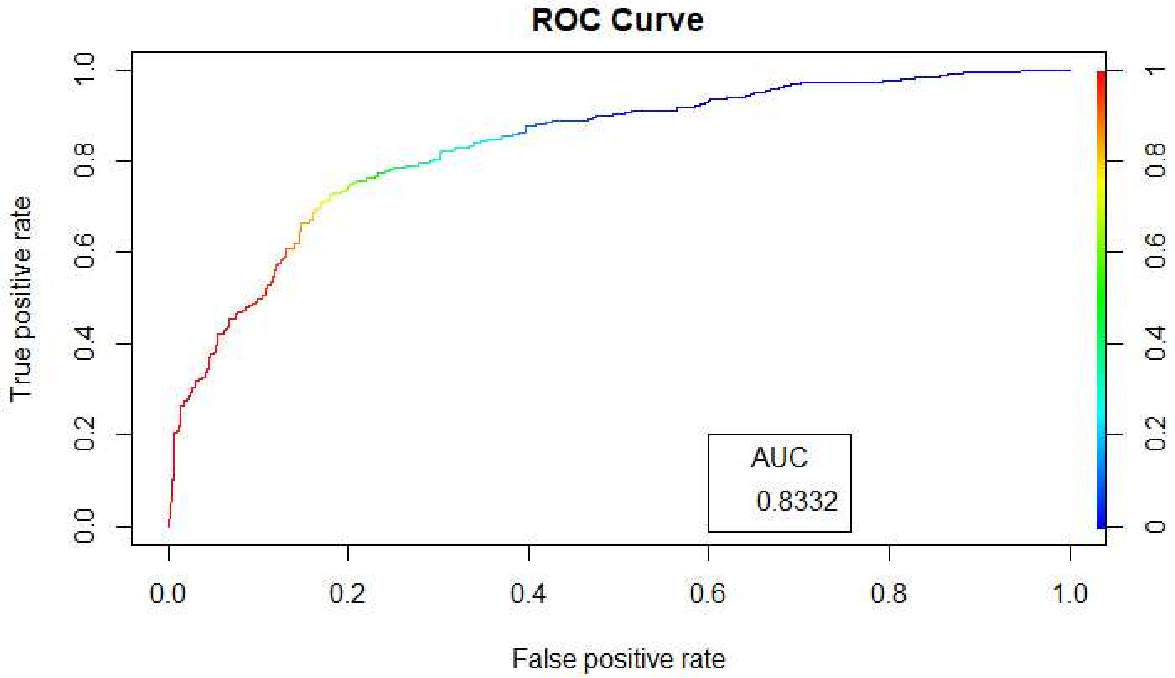
n = 50	size_2	size_3	size_4	size_5	size_6	size_7	size_8	size_9	size_10	size_11	size_12
mean	56.34857	75.09436	81.06334	92.94715	107.2341	105.0833	114.8998	101.9969	111.3623	111.6129	118.023
var	2090.666	1322.168	1692.458	1427.877	6021.337	1390.447	10759.36	742.776	650.8212	1591.447	2075.064
max	237.9367	237.9367	237.9367	246.9954	587.7316	264.9806	807.6571	165.3676	184.4907	292.0702	276.4095
min	21.45409	31.01093	36.27022	30.18084	25.03738	55.79065	56.32853	24.34114	71.09989	38.74771	21.94387
q1	33.95836	53.9165	48.54087	71.02959	74.63919	80.46796	86.38519	90.62381	93.91576	89.32516	89.29693
q2	40.85743	62.37233	64.1294	84.53138	93.47658	99.62379	94.54869	101.0689	112.3021	101.1793	110.6908
q3	48.66548	93.75935	107.6675	107.5493	118.8768	114.5563	113.3368	112.0698	121.525	123.5805	131.8327

Time effect on ANN

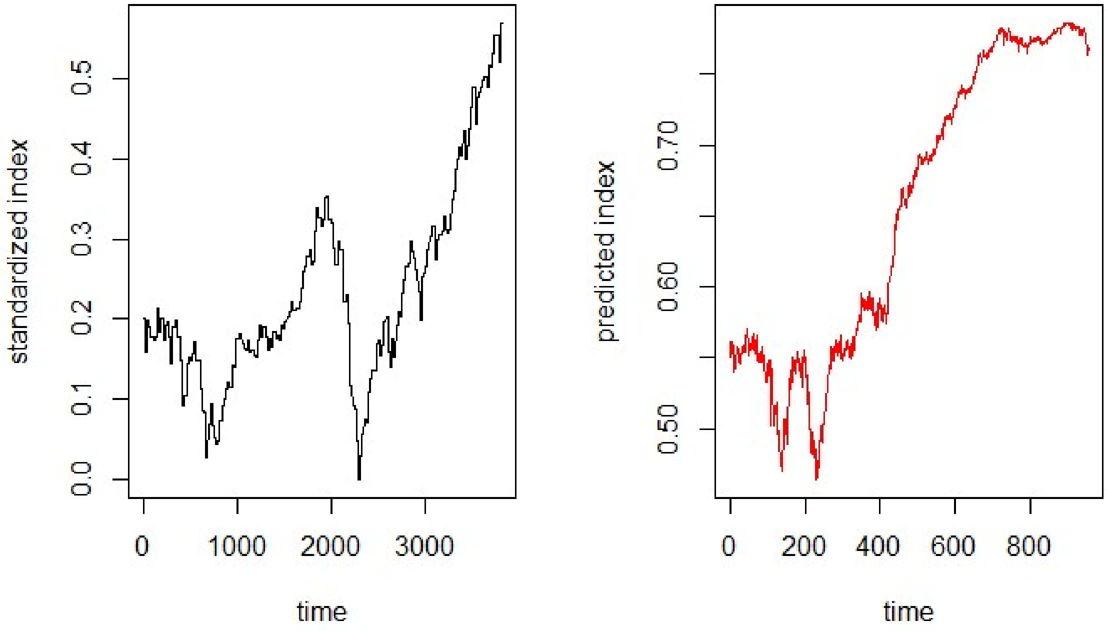
First, we see the difference in time effect on a classification problems. Originally the data we are using range from 2010 to 2018. Now we are testing from 2000 to 2018. The confusion matrix is as follow:

Size 11	Down	Up
Down	339	116
Up	103	386
Accuracy = 0.7680085		

and the ROC curve is as follow:



Surprisingly, when more data is included, the performance of the classification problem decreases. Let see the performance of predicting the actual price when the amount of data increase.



The trend of the predicted index can somehow follow the actual index, but it barely close to the actual index. Thus, we can conclude that an increase in data will reduce the performance of ANN.

Conclusion on ANN

ANN performs quite well on classification problems, but when it comes to continuous data, ANN starts to perform badly. Which is as expected since it is easier to obtain an adequate result with a binary classification problem. However, if some machine learning techniques are applied during the data cleaning process, then a better trained ANN will be expected.

3. Assumption and limitation

We assume that the time point that we can do the prediction is after we receive the open price. This is a kind of a 'greedy method' and hard to be implemented in the real world. Since training an adequate ANN cost quite a long time and sometimes it may consume over an hour. Thus, in the practical case, the open price of that trading date should not be considered. But obviously, it will lose some prediction power since some information has not been considered.

Besides, we treat all data as an independent variable. But they are time-series data. We did consider the time effect during our data cleaning process, but that is not enough.

In addition, pruning tree is required skill and it demands the expert experience of considering the optimal parameters for controlling the tree growing because the full tree does not mean it is the best model. Sometimes, it should do further study if the result of the leaf node is half-half because the classification tree cannot further explain the information with the present variables.

Further, both decision tree and ANN cannot predict the next date data well, not even classifying the index going up or down. Thus, we have to use 30 days data to predict the end date result, instead of the next date result. Which we expect that RNN and LSTM can do better.

4. Comparison of ANN and CTREE

ANN not only classify the observation as same as the classification tree but also able to predict the exact Index. ANN may perform well in both classification and prediction problems with high accuracy if the machine learning techniques are equipped.

However, it is a 'black box' model which is developed by the layers acting the role as a neural network and so the model is hard to be interrupted by human language. It always only results in the prediction or classification without any information, thus it is not useful when we try to explain the relationship between the response and the variables.

The result of the classification tree is easy understanding because it can discover the simple rule for classifying the observation. The higher the position of the tree the leaf nodes, the higher the significance level the variables are. Moreover, it is time-saving and easy to implement in computing. However, it spends more time to do the training model and data during discovering the best decision tree model with large accuracy.

5. Conclusion

Both ANN and classification tree contains its pro and cons, ANN, however, performs better than decision tree in predicting the value. The main reason behind is the algorithm of ANN can learn from the dataset although the quality of data is not good enough, classification is strongly dependent on the quality of dataset which should be highly correlated to the response.

It is recommended to do further study of the factors influencing the Dow Jones Index, especially the market message which will affect the demand of people buying or selling the stocks. One possible way is searching the frequency of the keywords on the Internet. For instance, we can count the time of daily searching of the Dow 30 stocks in the google searcher engine. Sometimes, it is more direct to analyze the relationship between stock price and human-behavior factors by using NLP. However, it is already a new topic and a completely different story which far beyond our level.

Contribution

CHENG Wing Ryan	0.5
Ip Man Fai	0.5