David CHAN Chi Fung

8th August, 2016

P5 Capstone Project

Investment and Trading

Definition

1. Project Overview

Algorithmic trading is one of the most popular trading strategies nowadays. Algorithmic trading is a method of executing an order using automated pre-programmed trading instructions accounting They were developed so that traders do not need to constantly watch the stock and repeatedly send slices out manually.

Building up a reliable predictor is the main goal in this project, moreover a reliable predictor is a cornerstone of a successful trading-program.

In this project, a predictor is created to forecast Hang Seng Index[1] with machine learning skill. Hang Seng Index is the most widely quoted indicator of the performance of the Hong Kong stock market and is freefloat-adjusted MV weighted index with a 10% cap on individual securities.

The index level of Hang Seng Index will be predicted by related asset, which will be the input of this predictor, User can therefore choose to invest in those corresponding finance product, such as index future, index option and index ETF

All the code will be show in the ipython notebook **only** and the written project here **will not involve any code.**

[1] Hang Seng Index: https://www.hsi.com.hk/HSI-Net/HSI-Net

2. Problem Statement

The goal in this project is to predict the daily index level of Hang Seng Index which is representing the level of Hong Kong stock market. It means that there will probably be some high correlated asset with Hang Seng Index in Hong Kong

Therefore, we will firstly find out the relationship between Hang Seng Index with few popular and high traded turnover asset in Hong Kong before predicting the index level in order to find out the suitable assets to be the input of the predictor. The following assets will be studied in this process:

I. Hang Seng China enterprises Index (^HSCE)

II. HSBC HOLDINGS (0005.hk)

III. AIA (1299.hk)

IV. China Mobile (0941.hk)

V. Tencent (0700.hk)

In order to run this program successfully, there are few python libraries and API which are necessary.

A. yahoo_finance[2] (API)

B. pandas

C. datetime

D. matplotlib

E. sklearn

All the data will be gathered from yahoo finance[3] (API). In order to explore all the selected data easily, Pandas, datetime and matplotlib will help us to develop a fancy data frame and plot all useful graph, like showing the daily price with some common used financial indicators and displaying the predicted index level in graph.

In the machine learning part, two predictors will be created, decision tree regression and linear regression. Before building up the predictors, there will be a necessary pre-process, this is to explore the data and drop off the unrelated data.

Finally, the model will be chosen to maximize R^2 score, which will be computed based on the true and predicted Index levels. A suitable predictor having a higher R^2 will be chosen as our final predictor.

[2] Yahoo Finance API: https://developer.yahoo.com/python/ , https://pypi.python.org/pypi/yahoo-finance

[3] Yahoo Finance: http://finance.yahoo.com/

Analysis

1.1 Dataset

Before starting the project, a clean dataset is important. This section will go to have a brief discussion of how the whole data will be, such as how many data point include, what kind of data will be included and how the data will be displayed.

Apart from Hang Seng index itself, five different assets' daily price will be chosen, Hang Seng China enterprises Index (^HSCE), HSBC HOLDINGS (0005.hk), AIA (1299.hk), China Mobile (0941.hk) and Tencent (0700.hk) respectively.

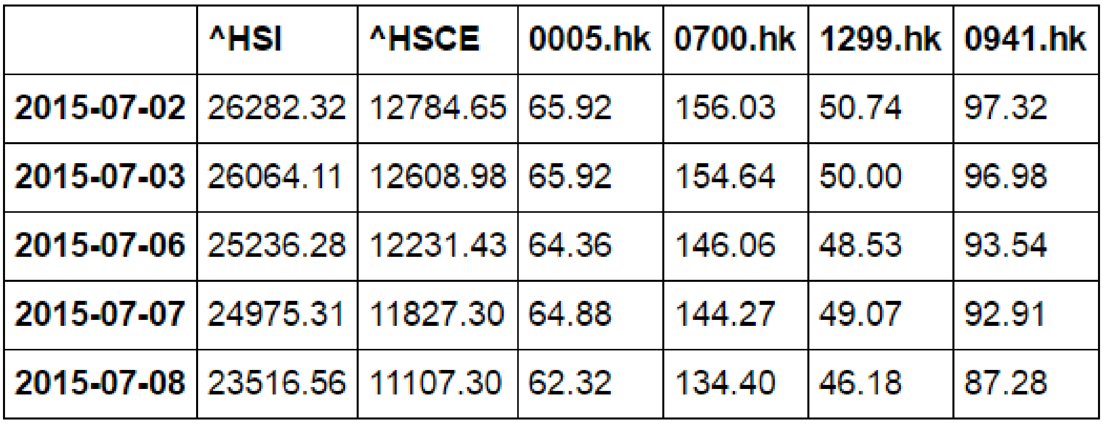
In stock market, there are few different type of price will be recorded, even though we are just focusing on daily price in this project. There will be opening price, closing price, highest price, lowest price and adjusted closing price in each day. Adjusted closing price, a stock's closing price on any given day of trading that has been amended to include any distributions and corporate actions that occurred at any time prior to the next day's open, is going to be utilized for analyzing.

Stock Market is affected by the whole world economic environment very much, such as exchange rate, interest rate, commodity-market and government policy. All these external factors are changing day by day, so **1 year** data is being suggested to be utilized in this project in order to have a similar external economic environment when the program is learning from the data.

In this project, all the adjusted closing price from 2015-07-01 to 2016-07-01 will be collected. Yet, there will be some missing data in this period, such as holiday or stock market activates. To deal will this problem, there will be some principles of how we fill in those missing data.

To begin with, when the date that index level of Hang Seng index is missing which means that Hong Kong stock market is closed in that date, the date in the dataframe will therefore be deleted. After that, stock might be stopped to be traded because of suspension, Resumption and Delisting. It will provide a missing data to the dataframe. Missing data will firstly be filled based on the last price (copying the last adjusted closing price as the price now). It means moving the data forward from the last daily price. Probably, there is no previous daily price to move forward, for example the stock is initial listed on the day inside the period rather than before the period. In order to make our data completed, a moving backward filling in will be used. It means that the missing price will be equal to the price that firstly appears in the dataset (copying today price as yesterday price, which is a missing data).

Finally, all the data will be gathered into a single dataframe as shown:



	^HSI	^HSCE	0005.hk	0700.hk	1299.hk	0941.hk
2015-07-02	26282.32	12784.65	65.92	156.03	50.74	97.32
2015-07-03	26064.11	12608.98	65.92	154.64	50.00	96.98
2015-07-06	25236.28	12231.43	64.36	146.06	48.53	93.54
2015-07-07	24975.31	11827.30	64.88	144.27	49.07	92.91
2015-07-08	23516.56	11107.30	62.32	134.40	46.18	87.28

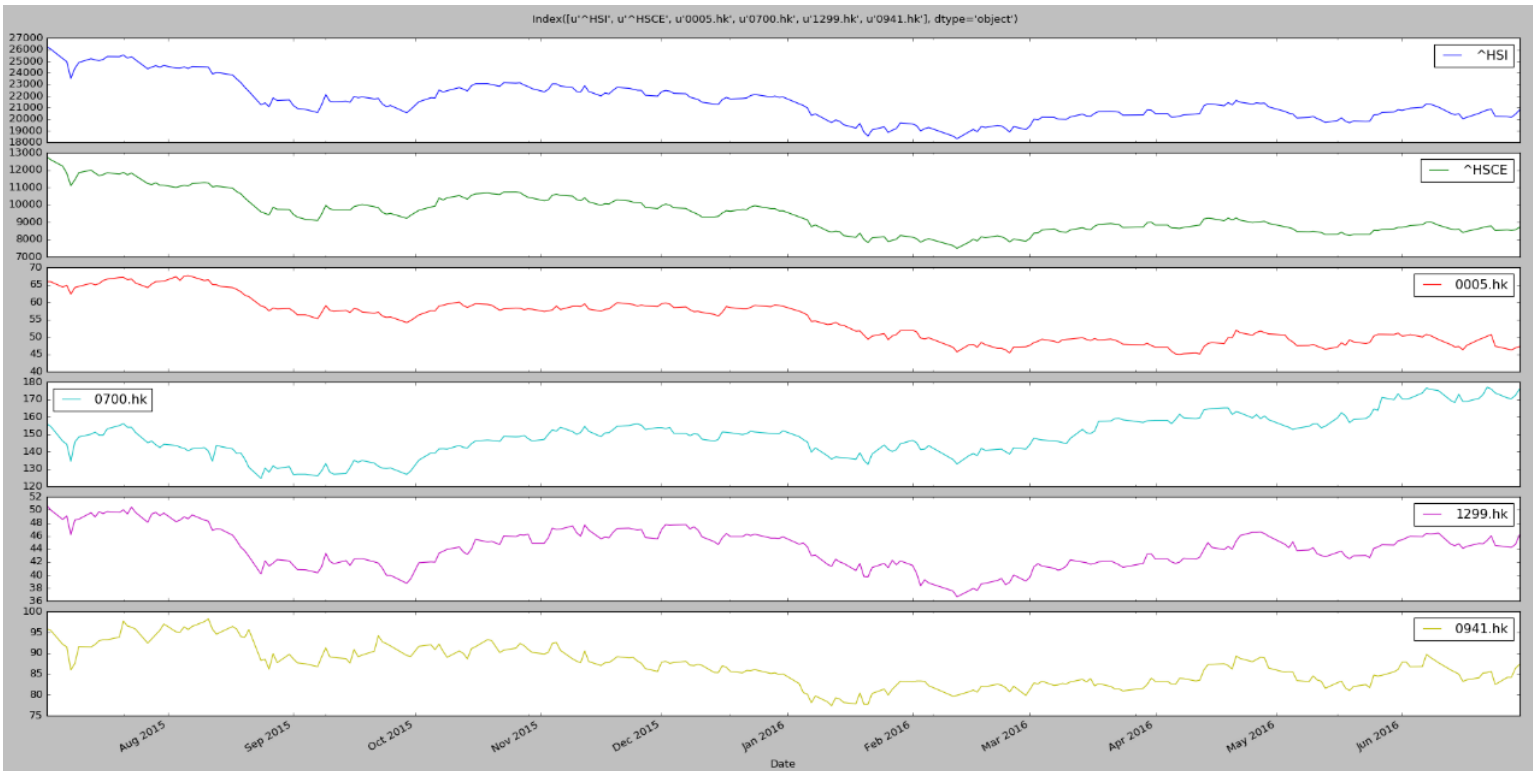
(Data.Head())[4]

1.2 Data Exploration

There are totally 248 trading days in this period of time. Here is a table to present the highest and lowest price or index level of these assets:

	^HSI	^HSCE	0005.hk	0700.hk	1299.hk	0941.hk
Max level or price	26282.32	12784.65	67.68	177.1	50.74	98.25
Min level or price	18319.58	7505.37	44.98	124.63	36.66	77.41

The follow figure is plotting the movement of each asset,



showing that the chosen assets have a very similar movement to Hang Seng index.

[4] .head() is a python function to show first few row of the selected dataframe.

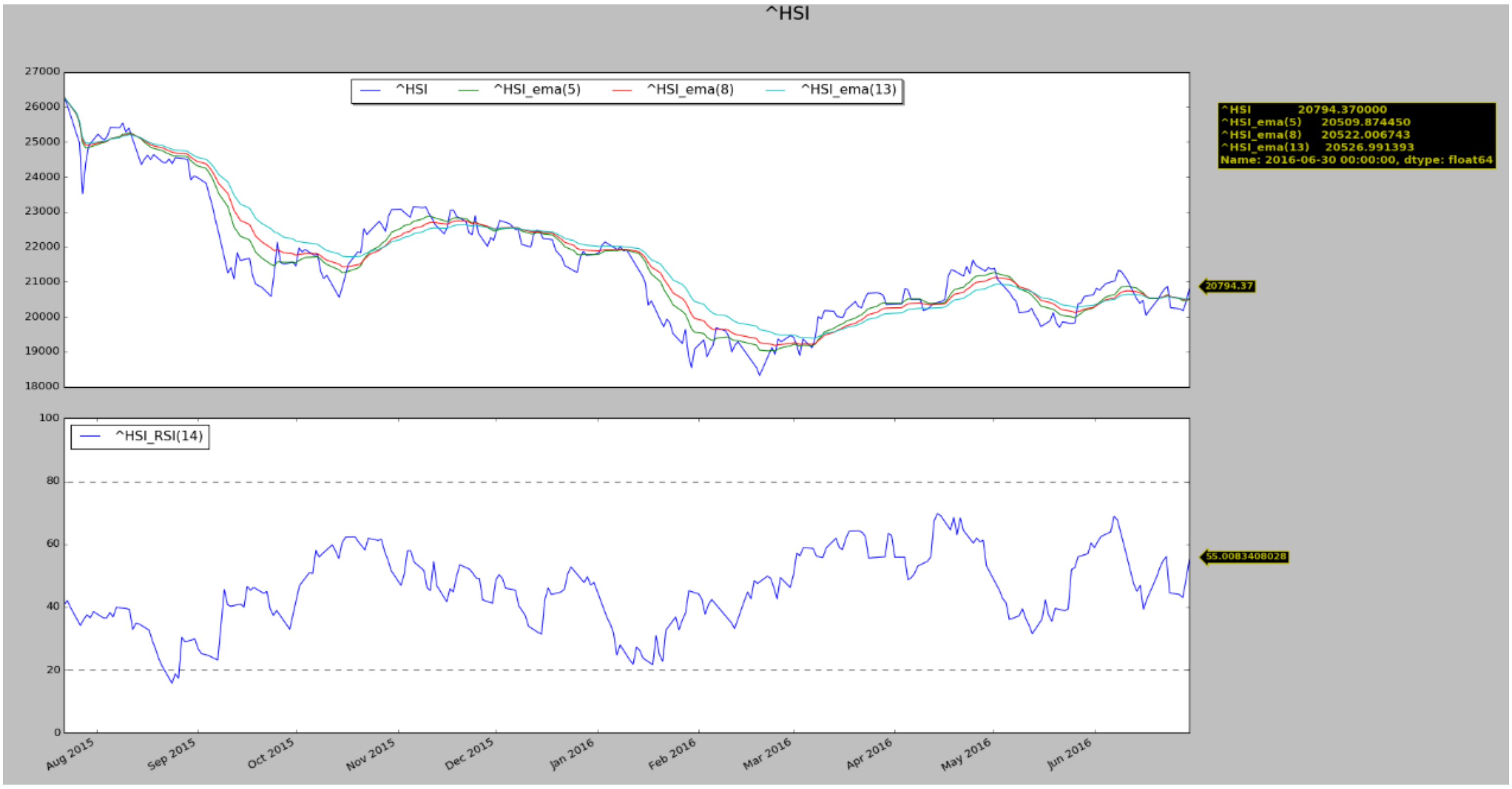
Finally, in order to make this project to be more financial and friendlier to investors, here is a defined function graph_with_indicator to explore a single asset. This function might not be related to the machine learning part very much, however, it will be useful for investors to go via the asset as it can produce two common used financial indicators, exponential Moving Average (EMA)[5] and relative Strength Index (RSI)[6].

Both exponential Moving Average (EMA) and relative Strength Index (RSI) are very common used technical analysis indicators in financial trading that will provide some kind of buying and selling signal.

As it is not the main part of the machine learning part, rather than explaining too much here, there will be two reference website providing related information of these. But it is important to note that this function is an extended function for investors to use this program rather than just focusing on machine learning and predicting.

Here is an example to display Hang Seng Index in graph, and more explanation will be in ipython notebook.

Example:



```
^HSI            20794.370000
^HSI_ema(5)     20509.874450
^HSI_ema(8)     20522.006743
^HSI_ema(13)    20526.991393
Name: 2016-06-30 00:00:00, dtype: float64
```

As shown above, this function will provide the EMA in 3 different windows, 5, 8 and 13, also the RSI in window =14. Moreover, there will be an annotation for the last price of the asset.

[5] Exponential Moving Average(EMA): http://www.investopedia.com/terms/e/ema.asp

[6] Relative Strength Index (RSI): http://www.investopedia.com/terms/r/rsi.asp

1.3 Machine Learning Algorithms

The goal of this project is to predict the index level of Hang Seng index. This project is going to use all historical data, such as the price of different stocks as mentioned, to predict the future index level of Hang Seng index.

A supervised machine learning algorithms is going to be used. Supervised learning is the machine learning task of inferring a function from labeled training data. In this project, the labeled training data are the price level of different assets and the output will be the index level of Hang Seng index.

In surprised learning, there are totally two main different type, which are classification and regression. Here, we are going to regression model as it is a continuous response variables. There are two different regression model, which are linear regression and decision tree regression.

Linear regression is trying to find the best fit line in the dataset. The advantage of using linear regression is that linear regression is one of the simplest and most popular modeling methods. However, linear regression is appropriate only if the data can be modeled by a straight line function, which is often not the case. Also, linear regression cannot easily handle categorical variables nor is it easy to look for interactions between variables.

Decision tree regression is using a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. The advantage of using decision tree regression is that it can solve the problem without a straight linear relationship. However, it probably consume much time the model.

Finally, there will be more and deeper discussion of each model in order to explain how both models run in this project.

1.4 Metrics and Benchmark

In this project, R^2 will be used as performance to evaluate the learning problem as it is a continuous numerical data. Here is the formal of it:

$$R^2 = 1 - \frac{SSR}{SST}$$

$$\mathrm{SSR} = \sum_{i=1}^{N} e_i^2$$

Where e is the error term.

$$\mathrm{SST} = \sum_{i=1}^{N} (y_i \ - \bar{y})$$

Normally, it is as high as preferable for R^2 representing that indicates the proportion of the variance in the dependent variable that is predictable from the independent variable. In this project, the less error the more preferable, R^2 will be great here to explain how reliable our predictor is as it is focusing on error term mostly, the higher error from prediction, the lower R^2 it will be.

Therefore, a high R^2 will be preferable in this project.

There are two different model will be trained. In order to choosing the most suitable model for this topic. R^2 will take part to determine which will be chosen as the final one. After choosing the model by higher R^2, another benchmark will be used.

Basically, R^2 is being used to compare different model. A model with higher R^2 means that this model can explain the data better than another one. However, R^2 itself is meaningless if we are not comparing different model. It is hard to say that R^2 is high enough in a single model.

Thus, +/-5% error boundary interval will be used, Here is the calculation of error:

$$\text{Error} = \text{Predicted value} - \text{Actual value}$$

It shows that if the predicted value of the index level of Hang Seng Index is higher than the actual value of Hang Seng Index, there will be a positive error, vice versa.

There are few advantage of using +/-5% error boundary interval. First, it is easier to readers who do not have much knowledge about machine learning or statistics. It can directly show readers how reliable how this predictor. Secondary, it is important to estimate the potential gain or loss by using this predictor to trade in finance market because earning money by using a reliable predictor is the main goal of this project. If the model’s error is finally below +/-5%, then we can expect the loss from trading by using this predictor that will less than 5%

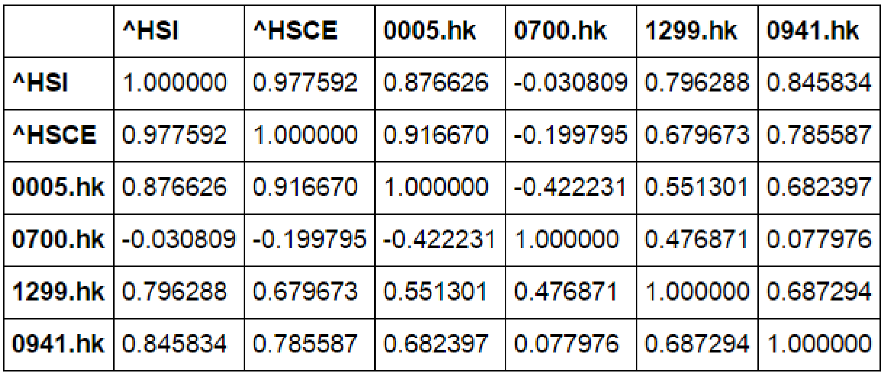
To conclude, there are two benchmarks will be used totally, first is R^2. We initially use R^2 to decide which model will be used. After that, +/-5% error boundary interval will take involved. We expect error from our final model’s predicted value that can be lower than +/-5% in order to show readers that the model built here can provide trading strategy that has less than 5% loss.

Predictor

In this part, some algorithms and Techniques will be involved here to explain how the predictor works. After defining all the metrics that will be utilized in this machine learning problem, an example will be shown to estimate how reliable this predictor will be.

1.1 Data removal

The correlation table below shows that the chosen assets have a high level of correlation to Hang Seng Index.



	^HSI	^HSCE	0005.hk	0700.hk	1299.hk	0941.hk
^HSI	1.000000	0.977592	0.876626	-0.030809	0.796288	0.845834
^HSCE	0.977592	1.000000	0.916670	-0.199795	0.679673	0.785587
0005.hk	0.876626	0.916670	1.000000	-0.422231	0.551301	0.682397
0700.hk	-0.030809	-0.199795	-0.422231	1.000000	0.476871	0.077976
1299.hk	0.796288	0.679673	0.551301	0.476871	1.000000	0.687294
0941.hk	0.845834	0.785587	0.682397	0.077976	0.687294	1.000000

(Correlation table)

For correlation,

$$\mathrm{corr(X,Y)} = \frac{cov(X,Y)}{\sigma_x\sigma_y} = \frac{E\left[(X-\mu_x)(Y-\mu_y)\right]}{\sigma_x\sigma_y}$$

, where E is the expected value operator, cov means covariance, X and Y expected values µX and µY and standard deviations σX and σY

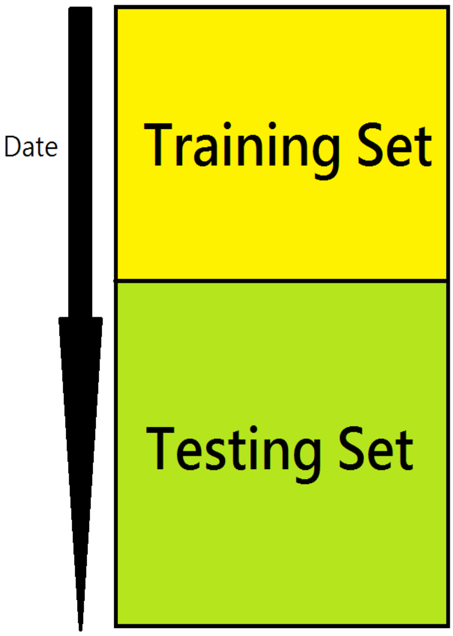
The correlation table shows us the relationship of each asset, the near to 1/-1, the higher relationship between asset. We can clearly find that Hang Seng index has a significant relationship with Hang Seng China enterprises Index (^HSCE), HSBC HOLDINGS (0005.hk), AIA (1299.hk) and China Mobile (0941.hk).

Tencent (0700.hk) will be dropped from the dataset as it has a very low correlation with Hang Seng Index. Although it is one of the most popular and biggest listed company in Hong Kong Exchange, it seems that Tecent cannot provide an obvious information to predict the level of Hang Seng Index.

1.2 Data splitting

Before inputting data into machine learning function, we should first splitting data into two different set, training set and testing set. Training set will be used to train our predictor to be the best estimator and testing set is being used to testing how reliable the predictor is.

As it is a time series data, the predictor will probably suffer from look-ahead bias if the time period time of testing set is before training set. Therefore, the training set will be prior the testing set.



In the training part, the splitting ratio will be 0.7. It means that 70% of the data will be treated as training set and 30% will be testing set. As there are totally 248 trading days in the dataset. Thus, there are 174 daily data in training set and 74 daily data in testing respectively.

1.3 Estimator

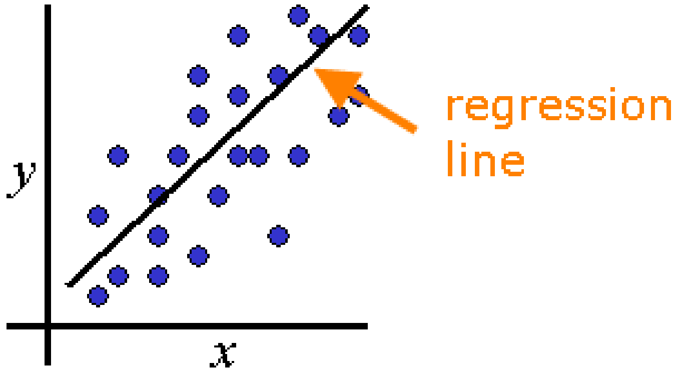
Two estimators will be created here in order to solve the continuous number data. They are linear regression and decision tree regression respectively. In this section, an introduction of these two models will be made before implementing the program.

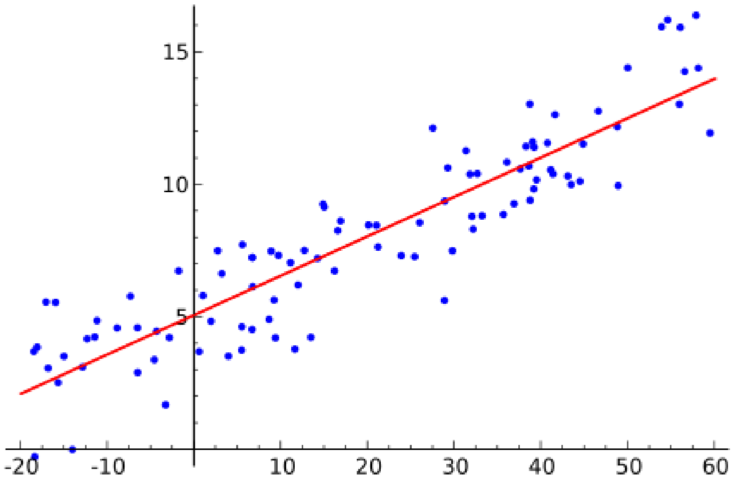
1.3.1 Linear Regression

The calculation of regression is like this:

$$y_i = \beta_0 + \sum \beta_i x_i \quad , i = 1 \cdots n$$

It means that we can just simply input(x), then we can calculate the output(y) where the output will be the best fit of the correspond data. Here are the graph simple shows this logic:





In this project, the input will be the index level of HSCE and the price of AIA, HSBC holding and China Mobile and the output will be the index level of Hang Seng Index.

To begin with, we will first train the model by using the index level of HSCE and the price of AIA, HSBC holding and China Mobile from training set. There are some default parameters setting of the algorithm, which are fit_intercept, normalize, copy_X and n_jobs.

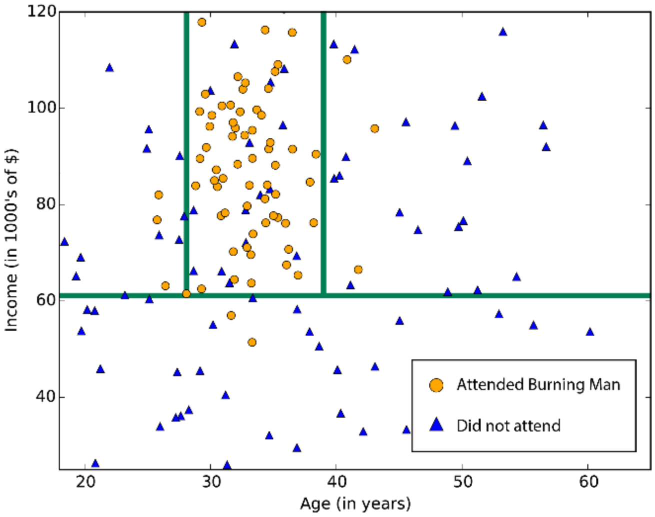
fit_intercept will set to be false because the intercept is not necessary to be calculated as it is ongoing time series problem. Furthermore, normalize will set to be true as default because the value of input in the model are very different, such as the index level of HSCE and the price of HSBC are very different. Thus, it will be better to normalize the X value (input). Finally, both copy_X and n_jobs will simply set as default because it will not affect the performance of the model directly.

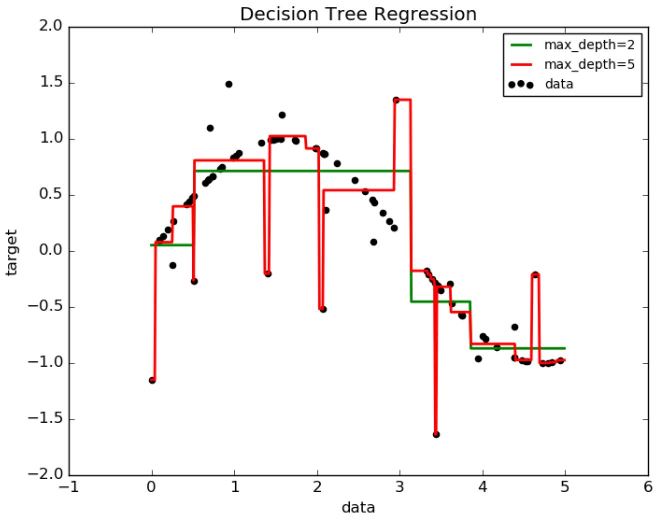
It is important to note that the output value is the necessary equal to the true value, but it will be the best fit one. For a good regression estimate, the predicted value will very near to the true value.

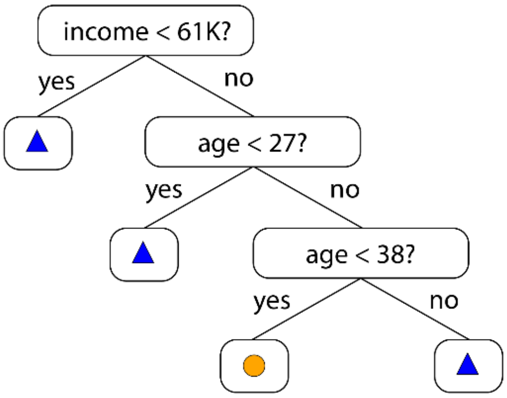
1.3.2 Decision Tree Regression algorithm

Decision Trees (DTs) are a non-parametric supervised learning method used for classifica3tion and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

For instance, in the example below, decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.







There are lots of different parameters inside this algorithm. One of the most important one is max_depth. A higher max_depth will make the model more complex and fit the data much more as shown in the graph above. Yet, it might cause overfitting problem, so it is not meaning that a higher max_depth must be better.

In the implement part, we will leave it as default first because the first thing we need to do is to determine which model will be used between decision tree and linear regression. If decision tree model is being chosen finally, we will fine turn the max_depth in order to avoid overfitting and underfitting, where overfitting means that the model is too complex to generalize the training data but not help for the data it doesn’t face before and underfitting means that the model is too simple that unable to generalize the data well enough.

1.4 Implementation

After all the preparation, such as deleting the unrelated data and splitting data, two regression model, linear regression and decision trees regression, will be implemented.

Initially, this section will start with training the two models before predicting the index level of Hang Seng index.

After training the model, we will predict the index level of Hang Seng Index from both training and testing data in order to calculate the R^2 for both models.

Finally, comparing the R^2 will help us to determine which model will be our final predictor. The principle of choosing the model is very simple as mention in the previous section. It is the higher R^2, the more preferable.

The following table shows the R^2 of training and test set in both models:

	Linear Regression	Decision Trees Regression
R^2 for training set	0.9926	1.0000
R^2 for test set	0.8906	0.0523.
Training time	0.0000000000 seconds	0.0039999485 seconds
Testing time	0.0000000000 seconds	0.0000000000 seconds

In training set, both models performance very well regarding to R^2 that both models are higher than 0.9. It means that more than 0.9 of data point can be explain in these models.

However, there is a bid different in test set. Linear regression perform far better than decision trees regression that linear regression got 0.8906 while decision trees regression got 0.0523.It means that when the model is facing to the data that have not faced before, linear regression obviously perform better in this project.

Moreover, there is one more difference between both models that is the training time. Although both models spend very less time in training and testing section, it shows that decision trees regression spends more time on training.

Therefore, linear regression will be chosen as our predictor because both performance and time spending are better than decision tree regression.

1.4 Turning model

After choosing linear regression model as our predictor, we now move on to turning our model to be better. In this section, there are two main part, which are GridSearchCV and data splitting. In the first part, we will try to find out the best setting of parameters of the model. In the second part, we are going to train and test our data in different data splitting. Finally, we will try to compare both R^2 and time spending in order to choose the best setting environment.

1.4.1 GridSearchCV

After determining the final model, GridSearchCV will involve in order to fine tune the linear regression model.

In LinearRegression, there are few different parameters, which are fit_intercept, normalize, copy_X and n_jobs

With different setting of the parameters, the regression will provide a slightly different result. R^2, the performance of the regression, will be therefore a bit different.

GridSearchCV is a process that helps us to find out which setting of parameters will provide the best result in terms of R^2.

Here is the setting of parameters:

	Before turning	After Turning
fit_intercept	True	True
normalize	False	True
copy_X	1	1
n_jobs	True	True

And here are the R^2 of before turning and after turning”

	Before turning	After Turning
R^2 for training set	0.9926	0.9936
R^2 for test set	0.8906	0.8709

It finally shows that the fine turn result provide a better performance in training set but worse in test set. However, the difference is not very significant.

1.4.1 Data Splitting

Originally, we used 0.7 as our splitting ratio. It means that 70% of the data will be treated as training set and 30% will be testing set. In this section, we are going to try to use 0.6, 0.8 and 0.9 as our splitting ratio in order to have a better performance.

The result are shown in the following table.

splitting ratio Performance	0.6	0.7 (original)	0.8	0.9
R^2 for training set	**0.9974**	0.9936	**0.9930**	0.9933
Training time	0.0040001869 seconds	0.0000000000 seconds	0.0000000000 seconds	0.0000000000 seconds
R^2 for test set	**0.2603**	0.8709	**0.9631**	0.9259
Testing time	0.0009999275 seconds	0.0010001659 seconds	0.0009999275 seconds	0.0000000000 seconds

The result shown above, showing that 0.8 data splitting probably the best suit in our model.

First, the time spending in both training and testing section do not have a significant difference in different data splitting ration. All are very near to zero. It shows that our predictor run really fast. We therefore move on the compare the R^2.

A lower data splitting ratio, 0/6, has provided the best R^2 for training set. Unfortunately, it gave the worsen R^2 for test set, which is obviously worse than other very much. Therefore, we should reject to use it as it cannot performance well to the data that it did not face to before.

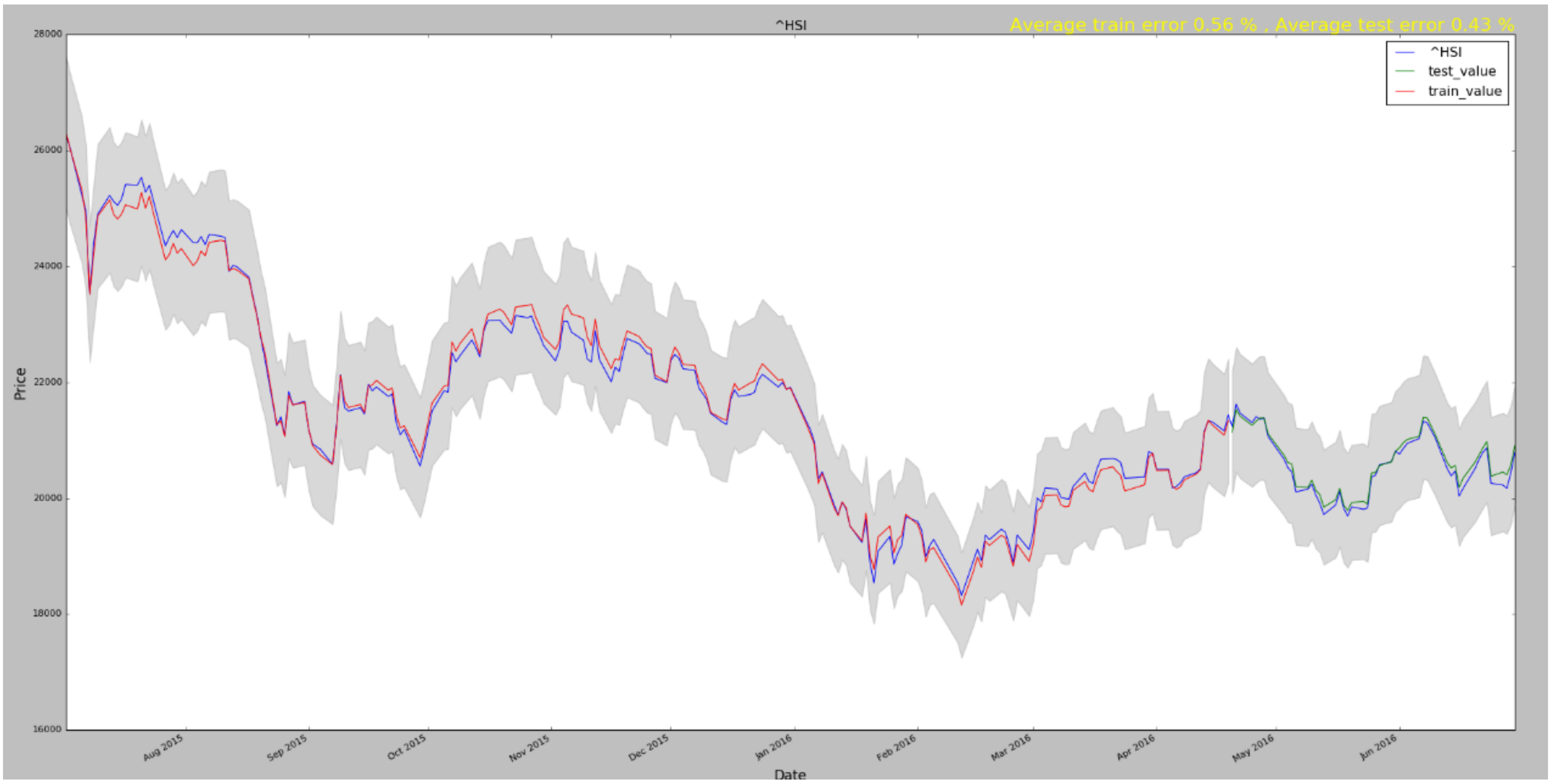
Both 0.8 and 0.9 provide a worse R2 for training set, but it is not significant, we can therefore ignore the difference here. However, R^2 for test set in 0.8 gave a significant better performance, which is far better than other ratio.

Thus, it will be better to use 0.8 as our splitting ratio in our final model.

Conclusion

After picking the suitable predictor, one more thing will be included. It is turning our model. Normally, there are lots of parameters in a model. Then, GridSearchCV can help us to pick the best parameters when we are training the model. After using GridSearchCV, we chose the best splitting ratio, which is 0.8. Both turning will help us have a better result of the prediction.

And here is the final result in graph of the prediction.



As mentioned in the previous section, linear regression model is chosen finally as it provide a higher R^2 comparing to decision tree regression model. It provides R^2 equal to 0.9930 in training set and 0.9631 in testing set. It is quite high as it can show that more than 95% of data can be well explained from the model.

However, R^2 might be a bit unfamiliar to people who don't know statistics or machine learning and R^2 is a bit meaningless if we are going to do ant comparing. Thus, Rather than looking to R^2 other, it will be friendlier to look at the actual percentage error between the prediction and actual value as investors or traders will look at the return much more than anything. To make it simple and general, we accept a +/-5% error as mentioned in the benchmark part. And the graph below will show us how reliable of this program.

The blue line in the graph is showing the actual value of Hang Seng index, the red line is the training value from the predictor, the green is the testing value from the predictor and the grey shadow is the +/- 5% Interval from the true value.

The graph here shows us that the predictor is reliable all the time as the error is very low, being closed to the actually value of the Hang Seng index very much.

In finance market, the market price will be overvalued or undervalued to the true value of an asset as we know that there are some over-bought or over-sold that lead to a mispricing problem. By using this program, it is possible to find out the true value of the Hang Seng Index, then, investors is possible to earn profit when the market price Is far away to the predicted value.

Summary

By now, this machine learning project successfully shows that it is possible to predict the index level of Hang Seng Index if we have the adjust price of different high correlated assets by using Linear regression model to predict it.

Initially, we try to collect all the data from Yahoo Finance API, including the index level of Hang Seng Index, Hang Seng China enterprises Index , the adjust closing price of HSBC HOLDINGS (0005.hk), AIA (1299.hk), China Mobile (0941.hk) and Tencent (0700.hk).

Then, we delete Tencent (0700.hk) from the dataset as it has a low correlation with Hang Seng Index. After that, both decision tree regression and linear regression are used to compare which model provide a better R^2.Next, linear regression are chosen to be our final predictor in our model because of a higher R^2 In testing set.

Around 0.9 R^2 is preferable as mentioned in pervious part. The result shows that the training set in this project can give a far higher performance and the testing set gives a very near 0.9 R^2. Thus, the program can work as expected in term of performance.

However, there are still few problem. 1) How this program helping trader earning money? 2) If the market is closed, for example the market is closed, is it still possible to predict the market?

The following part, Improvement, might give some information about this issue.

Improvement

It will be arguable that the trading chance might not be so many in this program and the index cannot be trade directly.

There are few ways to extend this program in a more powerful way. 1. The program is better to be implemented intraday, using 1 minutes or 5 minutes. In a intraday trading it is more chance to have a mispricing problem, for example there will be a time lagging that the price of AIA, HSCE or China Moble is changed, but the index do not response to it immediately yet, then it will be a high chance to earn profit.

Secondly, we can try to use this program to predict the future market rather than index. It means that investor can directly trade to the predicted result. Or investor can try to use it to trade index ETF, both these asset will have a time lagging problem as mentioned in an intraday fast market.

However, these improvement ways need to have a huge intraday dataset that source is limited. Bloomberg professional will be a great platform to provide this data set, however, the cost is not cheap to register it

After that, it is possible to predict the index level after market, by using ADR (American depositary receipt), which is a negotiable security that represents securities of a non-U.S. company that trades in the U.S. financial markets.[7] As the US market is open when HK market is close. It means that it is possible to predict the opening index level of Hang Seng by using the ADR overnight (HK time).

[7] ADR: https://en.wikipedia.org/wiki/American_depositary_receipt