# Machine Learning Engineer Nanodegree

## Financial Time-Series Prediction via Machine Learning

*Bryan Chik*

*15th August 2017*

##### I. Definition

#### Project Overview

Throughout the past 50 years, there has been numerous attempts and a rich discussion on financial markets mechanics and looking drivers of asset returns. However, one drawback of classical methods in predicting asset returns are most models are linear in nature. Therefore, most models have bad prediction power, as they cannot cater any non-linearity with the markets. Machine learning techniques could overcome this problem and qualifies as a good candidate to try solving the asset price prediction problem.

In this project, I would like to explore the possibility of applying machine learning techniques to predict Hong Kong Hang Seng index’s next day price movement, given closing prices of other 42 global exchange indices. A trading strategy will be designed based on the results of this research.

Initially I have obtained daily closing price of Hang Seng index and 42 other selected exchange indices globally from data provided by FactSet, starting from 1st January 2010 to 30th June 2017. A full list of indices can be found in the appendix.

The data is then trained to various model candidates and use for model performance evaluation and prediction.

#### Problem Statement

In this project, I am predicting Hang Seng Index’s next day movement, which has 3 states: Up, Neutral and Down. This problem is a classification problem.

“Up” label is defined as the daily price movement compared to yesterday’s close is greater than 0.5%. “Down” label is defined as daily price movement compared to yesterday’s close is less than -0.5%. While “Neutral” label is defined as not “Up” or “Down”.

4 classification algorithms will be utilized and see which one has the best predictability. Namely

* Logistic Regression
* AdaBoost
* Feed-Forward Neural Network
* LSTM

The performance of the above models are compared against a benchmark model, which in this case we are using Gaussian Naïve Bayes model, using 2 evaluation metrics for classification problems:

* F-beta Score (beta = 0.6)
* Precision

I anticipate there is at least 1 model would be better than the benchmark model, with higher precision and F-beta score. This in turn meaning the model does have higher prediction power, and can be used as an input to make trading decisions based on the prediction.

#### Metrics

2 specific metrics are chosen for evaluating the model performance:

* F-Beta Score (beta = 0.6)
* Precision

F-beta score is defined as

where = precision, = recall

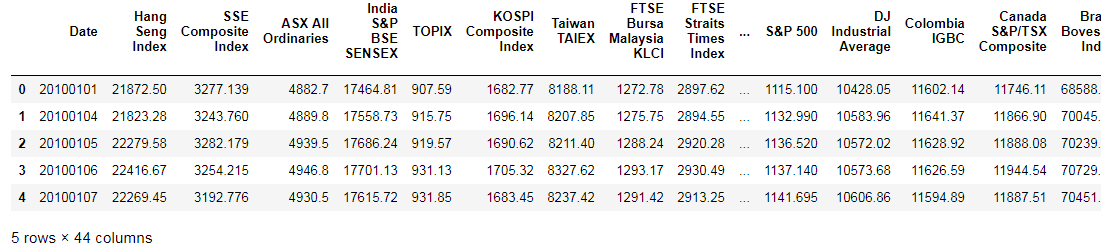
F-beta score is an overall measure that describes model accuracy, taking into account precision and recall. Note that I have placed special emphasize on precision with a beta value larger than 0.5. The reason behind is that in practical trading, it is easier to go long than short, so therefore, we want to predict better the “Up” labels.

For the same reason, precision is included in the evaluation metrics.

##### II. Analysis

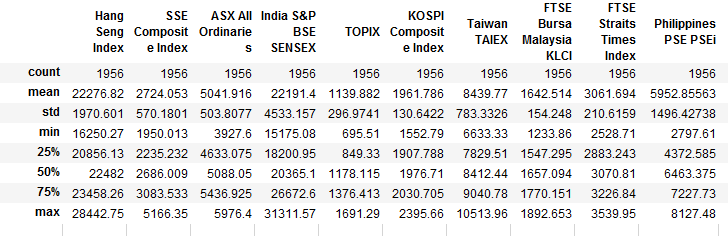
#### Data Exploration & Visualization

The data that I used to study is sourced from [FactSet Research System’s](http://www.factset.com/) investment management workstation terminal. I have downloaded daily closing index levels of 43 global equity exchange indices, including Hang Seng index, in the form of a csv file. The first column is the date index, starting from earliest to closest. A snippet of the data is shown below.



*Figure 1: First 5 rows of the raw index close data*

Next, I have generated some basic descriptive statistics of the data set. This is an attempt to get some knowledge around the distributional properties and shape of the data, and with the aim to determine any data pre-processing is required or not. An excerpt of the output is shown below.



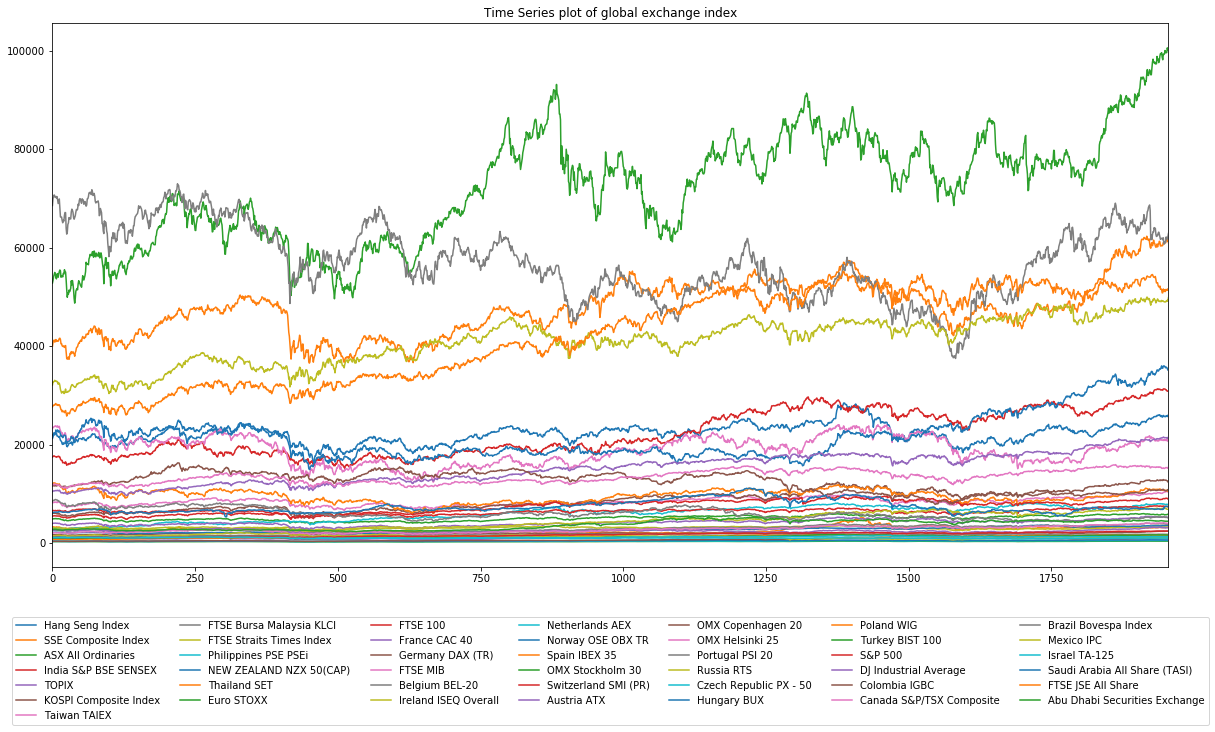
*Figure 2: Descriptive Statistics of Asia Pacific indices*

There are some important observations within the table:

1. We have around 1,956 sample data points
2. Different indices have different levels. For example, mean Hang Seng level is 22,276.82, while for Japan TOPIX index the mean level is around 1,139.88
3. Different indices have different volatility profiles and range

From the above observation, particularly 2 and 3, imply that the data is not directly usable as they are at different levels or scale. In order to preserve the structure of the data, the raw data has to go through a normalization process.

To further visualize the structure of the data, below is a plot of the data

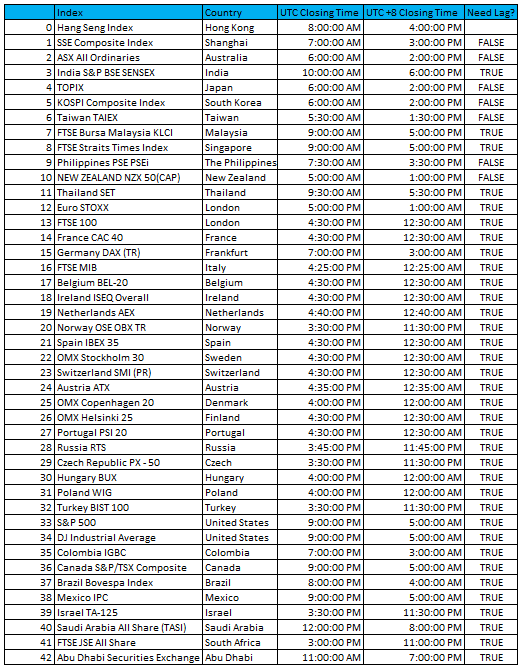


*Figure 3: Time series plot of index data*

From the above figure, the indices have various levels. In order to preserve their structure, normalization is required.

There is an extra point to note from the above plot. It looks like all the indices exhibit a trend. This trend component will cause issues during our fitting process, and therefore we need to “de-trend” the data. In statistics and time-series analysis context, the data has to be transformed into a stationary series before they can serve as an input in fitting our models.

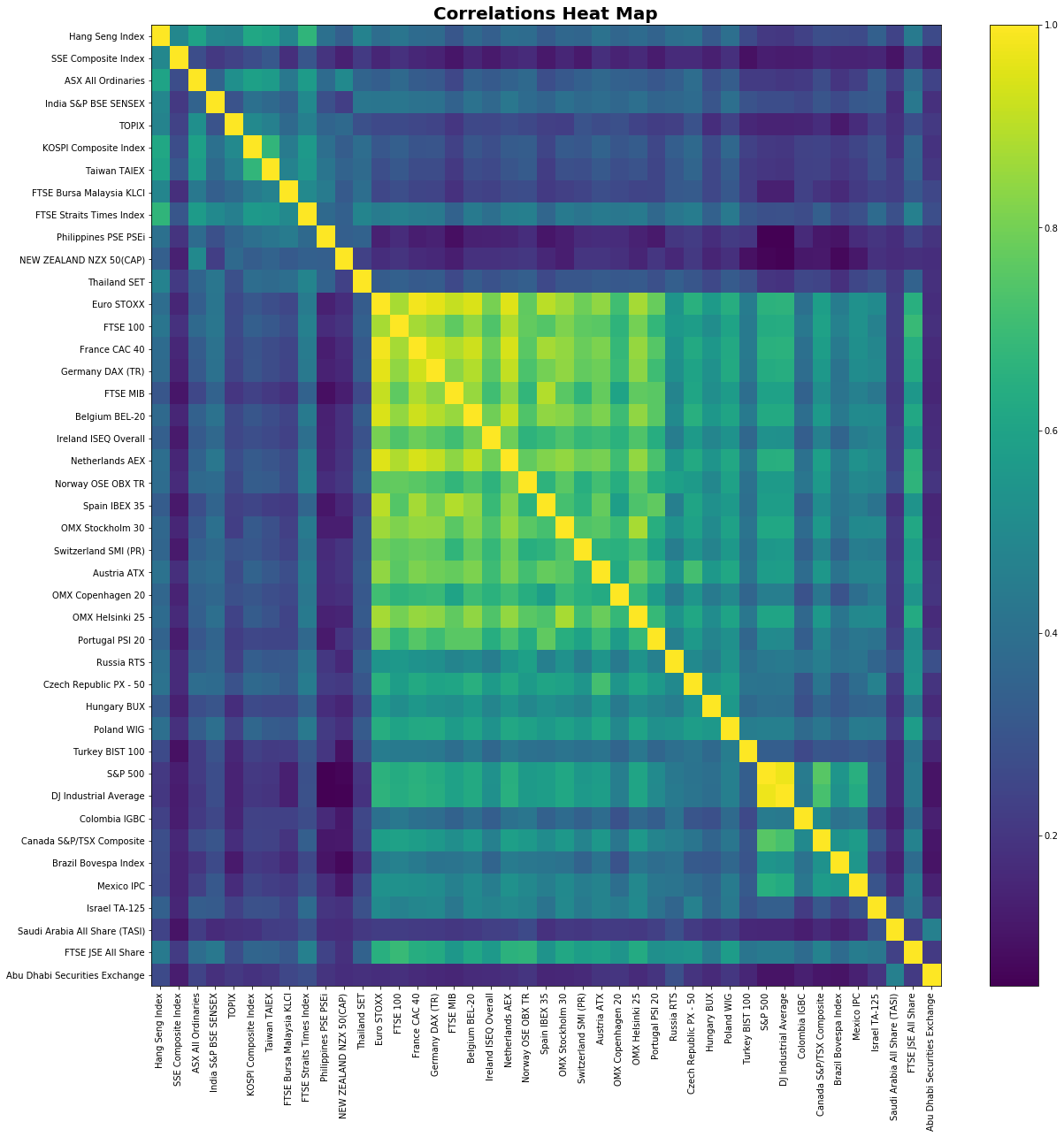
A practical issue with the current data is that since the exchanges are in different time zones, we will need to apply a proper lag to indices that are in slower time zones. Based on each exchange’s closing time, the below table lists out which indices need to be lagged. This step is essential and critical to avoid “look-ahead bias” which you predicting the past using the future.



*Figure 4: List of exchange closing time*

*Source:* [*https://en.wikipedia.org/wiki/List\_of\_stock\_exchange\_opening\_times*](https://en.wikipedia.org/wiki/List_of_stock_exchange_opening_times)

Finally, the selected indices would need to possess some kind of correlation to Hang Seng in order for them to qualify as a predictor. If the index is independent of Hang Seng, then it has no contribution to Hang Seng’s movement information and can be dropped. To verify this, I have generated a sample correlation heat map below to investigate the correlation structure, using the lagged transformed data.



*Figure 5: Sample correlation of the lagged index data*

From the above chart, we can observe the following

1. Indices within the same geographical region tends to have higher correlation
2. Developed market indices are more correlated to each other compared to emerging market indices
3. Hang Seng index itself are more correlated to Asian indices, but indices from Eurozone and Americas also have positive correlation

From the observations above, we can initially confirm the indices should possess various degrees of prediction power towards Hang Seng index. Intuitively this makes sense as stocks listed in Hang Seng index are likely to have business running in different countries globally. Therefore, each stock should possess exposures to different countries, and its stock price will be affected by those markets.

#### Algorithms and Techniques

In this project, I have explored 4 different algorithms and compared their performance:

* Logistic Regression
* AdaBoost
* Feed-forward Neural Network
* LSTM

All 4 of the algorithm can be used as a classifier and able to take in continuous numerical feature inputs for prediction.

Logistic Regression

Logistic Regression served as a start as a lot of classical financial economics literature has been written based on a linear model form. We used the pre-processed data and fit the logistic regression model. Logistic regression aims at finding the minimizing the cross-entropy between the predicted labels and the true labels (which in our case, Up/Down/Neutral). The model finds the regression coefficient that, for -penalized cost

\underset{w, c}{min\,} \frac{1}{2}w^T w + C \sum_{i=1}^n \log(\exp(- y_i (X_i^T w + c)) + 1) .

Or -penalized cost

\underset{w, c}{min\,} \|w\|_1 + C \sum_{i=1}^n \log(\exp(- y_i (X_i^T w + c)) + 1) .

Source: [http://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression](http://scikit-learn.org/stable/modules/linear_model.html%23logistic-regression)

During the fit, time-series cross-validation and grid-search is performed to find the optimal regularization hyper-parameter and whether or penalization provides the best fit.

Special attention should be placed during the cross-validation. In this project, I have used a time-series cross-validation method which is an extension to -fold cross-validation technique. Traditional -fold cross-validation is not applicable in our context due to the fact that the data possess ordering. If we randomly shuffle the data, there is a possibility where the training set contain future time points and used to predict the past within the test set. Instead, the time series cross-validation will divide the data into folds. Label as the -th fold, where . At the -th iteration during cross-validation, we will be using as the training set, and as the test set. The chart below depicts the cross validation process. This can ensure that the ordering of the time series can be preserved and no future time points will be used to predict the past.

https://i.stack.imgur.com/fXZ6k.png

*Figure 6: Diagram depicts the time-series cross validation method used in this project. Source:* [*https://stats.stackexchange.com/questions/14099/using-k-fold-cross-validation-for-time-series-model-selection*](https://stats.stackexchange.com/questions/14099/using-k-fold-cross-validation-for-time-series-model-selection)

Finally, is a hyper-parameter that controls the proportion between the cross-entropy error and the regularization. A larger puts more emphasize on minimizing the cross-entropy, and diminishing the regularization effect on the regression coefficients.

AdaBoost

The next model I have tried is the AdaBoost model, which is a model falls in a bigger class of ensemble learning models. Ensemble methods are well-known for its strong performance in real-world machine learning problems.

Ensemble methods rest on a principle that iteratively fitting a weak learner could converge to a strong learner and therefore gives better predictive power.

In this project, I have used the default weak learner from scikit-learn library, which is by using Decision Trees and also the default boosting algorithm, SAMME.R . Similar to the Logistic Regression case, a grid-search is performed on the below hyper-parameters

* Number of estimators: Maximum number of estimators that the boosting algorithm will stop. The grid search is performed in the discrete set
* Learning Rate: Controls the shrinkage of the contribution to each classifier. The grid search is performed in the discrete set }

Same cross-validation methodology is used from logistic regression.

Feature importance metrics can also be generated in AdaBoost algorithms and cross check with intuition to see whether the model results make sense.

Feed-Forward Neural Network

Next I have turned to deep learning techniques, starting with a simple Feed-forward Neural Network. With a single hidden layer and 21 perceptrons. Neutral Network is able to capture non-linearity exhibit with financial markets, and therefore I would expect the model should do better.

The model architect is depicted below:

***Insert Chart below***

Other parameters of the Neural Network are as follows:

* Drop-out Probability = 0.6
* Activation: ReLU
* Epoch: 100
* Optimizer: Nesterrov Adam
* Learning Rate = 0.0003
* Batch Size = 1

Instead of cross-validation, the sample is split into 80% training and 20% test set. 100 epoch runs are done during the training step, and predicts are run on the test set.

ReLU and drop-out are nowadays a standard feature to enhance the model performance. ReLU are known to provide better and more stable results, while drop-out can prevent overfitting.

Nesterov Adam optimizer is essentially Adam optimizer incorporated with momentum features. Nesterov momentum tends to have superior performance in locating the optimal point. (*Incorporating Nesterov Momentum in Adam -* <http://cs229.stanford.edu/proj2015/054_report.pdf>).

Learning rate is finely tuned such that when running 100 epochs the model loss is stabilized. More will be discussed in the next section.

LSTM Network

Finally, a type of recurrent network, long short-term memory network is used to fit the data. The LSTM network is capable of capturing long-term dependencies of the network states which traditional neural networks are not able to handle. There are well documented literatures describing financial time series possess memories, or “momentum” effects. As a result, LSTM is perhaps a very good candidate model such that this effect can be captured within the network architecture.

Similar to the Feed-forward Neural Network, I have only used 1 hidden layer and 21 percetrons. The model architecture is depicted below:

***Insert Chart below***

Other parameters of the Neural Network are as follows:

* Drop-out Probability = 0.6
* Activation: ReLU
* Epoch: 100
* Optimizer: Nesterrov Adam
* Learning Rate = 0.0003
* Batch Size = 1

Similar to the Feed-Forward Neural Network case, I have used 80% of the data as training set and 20% as test set to validate the model.

#### Benchmark

I have chosen the Naïve Gaussian Bayes as the benchmark model. Empirically, the returns distribution is approximately Gaussian. Therefore we can compare the models whether they are better off than computing the best likelihoods, under a Gaussian assumption.

The same evaluation metrics are computed (i.e. F-beta and precision) to compare across the models.

##### III. Methodology

#### Data Preprocessing

From the discussion in *Data Exploration* section, the following preprocessing steps are applied to the data