Singapore University of Technology and Design

# Stock Market Prediction

50.038 Computational Data Science

Denny Handoko Bahar (1001579)
Victor Toh Wei Jie (1002090)
Bernard Cheng Zheng Zhuan (1002053)

# Contents

# 1. Introduction

The topic on financial market forecasting is a popular time series problem and there is a vast amount of information which can be incorporated to make predictions (E.g. macroeconomics information, financial statements, etc.) to varying degrees of accuracy. The scope of this project is to investigate the possibility of predicting the financial markets to some degree of accuracy using only limited information; in this case the closing price. The beauty of using a security's closing price to make predictions is that the information is easily accessible and sourced by the public. Therefore, it shows that predictions can be done by using only simple data sets that are easily available.

The team's strategy is to use the time series behaviour of a security to make predictions and utilise another related security to derive a relationship between the pair to improve the prediction. The main criteria for choosing the pair of securities is by measuring the level of cointegration and using the concept of mean reverting time series. Using the criteria, the focus is shifted to Exxon-Chevron stock pair for illustration and discussion purposes. The context is to predict just one stock, Exxon, while taking advantage of its time series relationship with Chevron.

For this project, two types of predictions are tackled: the direction and the change of stock price for some period in the future. To predict the direction of the stock, in this case for Exxon, this can be interpreted as a classification problem where the label contains a binary value to represent "Up" or "Down" direction. To predict the percentage price change of Exxon stock, this can be interpreted as a regression problem where the response variable is the percentage change of Exxon stock price.

The construction of the data frame to be used for the predictive model involves 2 important considerations; the predictor variables and the look-ahead period for prediction. From examination, the period for stock price prediction is set to be 16 days ahead, equal to the half-life period of the mean reverting time series from the residual of the cointegrated pair Exxon-Chevron. The predictor variables considered are momentum, volatility, correlation and cointegration of Exxon and Chevron.

Finally, machine learning and deep learning algorithms are utilised to make predictions and evaluated using nested cross validation method. For the classification prediction, the performance used to evaluate the model is the accuracy of the prediction. For regression, the performance used to evaluate the model is the Mean of Squared Error (MSE).

This paper is organized as follows: Section 2 describes the details of the dataset collected, the pre-processing steps and the selection of cointegrated pair. Section 3 provides the methodology of acquiring the response and predictor variables which can be used to make prediction. Section 4 presents the predictive models and their evaluations. Lastly, Section 5 discuss the obtained results and conclusion.

## 2. Data Description

This section describes the collections of the dataset, the pre-processing steps and the methodology for selecting the cointegrated pair.

### 2.1 Dataset & Collection

The data sets are obtained from the Bloomberg Terminal in comma separated value (CSV) format. The data consist of the daily open, high, low and close prices and volume traded for 12 pairs of securities which totals 24 data sets. The 12 pairs are made up of assets from across industries: 1 household, 1 index, 2 F&B, 1 energy, 1 retail, 1 financial service, 1 metal, 1 aerospace/defence, 1 airline, 1 telecommunication and 1 automobile. The securities are presented in Table I and it is grouped by pairs and industries.

The data time frame obtained ranged from a start period of 1 January 1991 to the end period of 31 December 2017, a total of 27 years. The reason for a long time period is to have a sufficiently large sample needed to train and test the models. Naturally, the securities are chosen based on the industry type and it must be publicly traded between the selected period.

**Table 1: Dataset**

| Pair | Security 1 | Security 2 | Industry |
|---|---|---|---|
| 1 | Johnson & Johnson | P&G | Household |
| 2 | S&P500 | Down Jones Industrial 30 | Index |
| 3 | Coca Cola | Pepsi | F&B |
| 4 | McDonald | Wendy's | F&B |
| 5 | Exxon | Chevron | Energy |
| 6 | Walmart | Target | Retail |
| 7 | Bank of America | Wells Fargo | Financial Service |
| 8 | Gold | Silver | Metal |
| 9 | United Technologies | Raytheon | Aerospace & Defence |
| 10 | Southwest | Skywest | Airline |
| 11 | Comcast | AT&T | Telecommunication |
| 12 | Honda | Ford | Automobile |

### 2.2 Data Pre-processing

In the analysis the consideration is based solely on the closing price on the security. The first part of the data pre-processing step is to correct the missing data for the column under closing price. For the prices with missing closing price, an estimate was done by taking the average of the high and low price of the day. Finally, inner join was performed on the dates using Pandas library to standardize the dataset.

### 2.3 Selecting Cointegrated Pair

The difference between the pair could be either a mean reverting time series or non-mean reverting time series. For predictions, it is an advantage to use a cointegrated pair with a mean reverting time series. This is because a mean reverting time series suggests that the stock will return to its mean and fluctuate around the mean. Therefore, for a mean reverting time series the current price level gives more information about the next movement of the price: If the price level is greater than the mean, it will have a bias to move downward and if the price level is lower than the mean, it will have a bias to move upward.

In most cases, financial prices do not behave like a mean reverting time series. But fortunately, an artificial time series which is mean reverting can be fabricated by taking the difference between a pair of cointegrated stocks. For a pair of securities, $Y(t)$ and $X(t)$, $Y(t)$ is said to be cointegrated with $X(t)$ if there exist a β which can linearly transform $X(t)$, such that $Y(t) - \beta X(t) = E(t)$, where $E(t)$ is a mean reverting time series.

A mean reverting test called Augmented Dickey Fuller (ADF) Test is used to test if $E(t)$ is a mean reverting time series. The price change of time series $E(t)$ can be described as follows:

$$\Delta E(t) = \lambda E(t-1) + \mu + \beta t + \alpha_1 \Delta \mathrm{E}(\mathrm{t}-1) + \cdots + \alpha_k \Delta \mathrm{E}(\mathrm{t}-\mathrm{k}) + \varepsilon_t$$

Where $\Delta E(t) = E(t) - E(t-1)$, $\Delta E(t-1) = E(t-1) - E(t-2)$ and so on. The ADF hypothesis test is as follows:

$$H_0: \lambda = 0 \text{ (Random walk)}, \; H_a: \lambda < 0 \text{ (Mean reverting)}$$

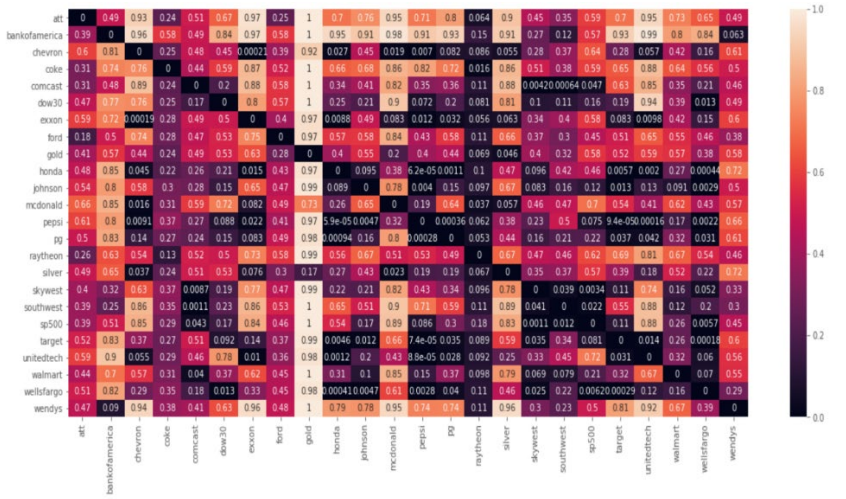
Hence, if the null hypothesis that λ = 0 can be rejected and accept the alternative that $E(t)$ is mean reverting. The steps of finding β and performing ADF test is referred to as cointegration test.

To find β, an ordinary linear regression was performed on the pair by setting one security from the pair as the dependent variable and the other as independent variable. The next step is to carry out ADF test on the residual. These steps are executed on the permutation of the 24 datasets obtained, excluding self-pairing. The reason to perform cointegration test for securities from different industries is twofold. Firstly, because of the complexity and interlinkage of different financial markets, the securities may still be related in some way. One example is the case of Down Jones and Exxon. Since Exxon is a component in Down Jones, both securities may have certain degree of cointegration. Secondly, cointegration is not symmetrical in nature. That is, if $Y(t) - \beta X(t)$ is mean reverting, $X(t) - \beta Y(t)$ may not necessarily be mean reverting.

The p-value of ADF test results are then presented in Figure I as a matrix heatmap. Cointegrated pairs output lower p-value which are illustrated by a darker colour. Due to the non-symmetrical property of cointegration, the matrix is not symmetric. The matrix is interpreted as follows: reading the matrix

horizontally dictates the selected row security is the $Y(t)$ and the columns as $\beta X(t)$ and vice versa is true for reading the matrix vertically.

**Figure 1: Heatmap of cointegrated pair**




To select the pair, the top pair is filtered with the lowest p-value and the most fundamentally sound pair is selected to ensure that this relationship is not based on statistical coincidence. The top 10 pairs can be seen in Table 2 and the pairs are formatted as $Y(t) \sim X(t)$. The Pepsi ~ Honda pair is not chosen despite having the lowest p-value because they do not seem to have a strong fundamental to support their relationship. The low p-value may have been arisen due to chance. The choice was made to select the Exxon ~ Chevron pair at the 5th place because the pair have a low p-value and are fundamentally sound.

**Table 2: Top 10 Cointegrating Pair**

| No. | Pairs | P-value |
|---|---|---|
| 1 | Pepsi ~ Honda | 0.000059 |
| 2 | Target ~ Pepsi | 0.000074 |
| 3 | United Tech ~ Pepsi | 0.000088 |
| 4 | Target ~ Wells Fargo | 0.000175 |
| 5 | Exxon ~ Chevron | 0.000187 |
| 6 | P&G ~ Pepsi | 0.000276 |
| 7 | Wells Fargo ~ Honda | 0.000408 |
| 8 | Comcast ~ Southwest | 0.000636 |
| 9 | P&G ~ Honda | 0.000940 |
| 10 | S&P500 ~ Skywest | 0.001051 |

The rest of the analysis will be focused on Exxon ~ Chevron data pair to make predictions. The prediction is done on the stock price of Exxon while using its relationship with Chevron to improve on the predictions made.

## 3. Problem and Model

From previous section, the decision was to focus on the Exxon-Chevron pair and, mainly to predict the stock price of Exxon. This section will discuss the methodology used to derive the set of response variables and predictor variables to predict the direction of Exxon stock price some period in the future (classification) and to predict the percentage price change of Exxon stock some period in the future (regression).

The reason to predict the percentage price change rather than the absolute stock price for the regression problem is because many of the machine learning algorithms, especially the neural networks, work better with normalized data. In this respect, predicting percentage price change is more advantageous.

Let the Exxon stock price be $Y(t)$, the first and second problems can be written as follows:

Problem 1:

$$Sign(Y(t) - Y(t-k)) = f(X_1(t-k), \dots, X_n(t-k))$$

Problem 2:

$$\frac{Y(t) - Y(t-k)}{Y(t-k)} = f(X_1(t-k), \dots, X_n(t-k))$$

Where $Y(t-k)$ is Exxon stock price from the past $k$ days and $X_n(t-k)$ is the predictor variable $n$ at time $t-k$.

### 3.1 Determine look-ahead period

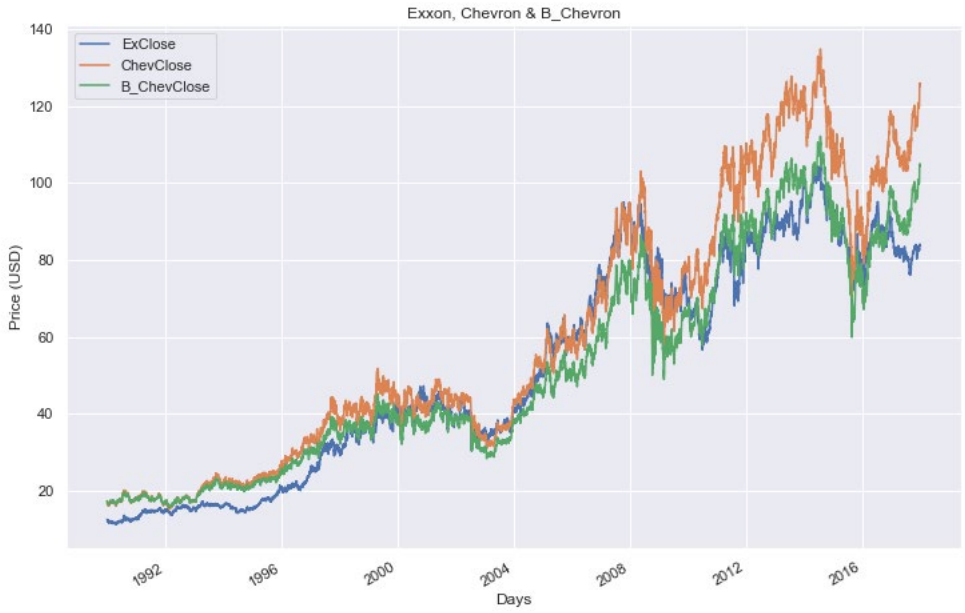
The goal here is to determine how many days ahead to predict Exxon stock price. To do that, the half-life period of the mean reverting time series is calculated. The mean reverting series is determined by taking $Y(t) - \beta X(t)$ where $Y(t)$ is Exxon price and $X(t)$ is Chevron price.

**Figure 2: Exxon & Chevron Stocks Price (1991 – 2017)**

From Figure 2, just by observing the direction of the trend it is obvious that Exxon and Chevron stocks price are very correlated, and they move in a similar path.

To linearly transform Chevron, ordinary linear regression was performed where the dependent and independent variables are Exxon and Chevron respectively. Since this is a time series problem, the linear regression is moving with window of 100 days. Figure 3 shows plots of Exxon, Chevron and linearly transformed Chevron (β_Chevron).

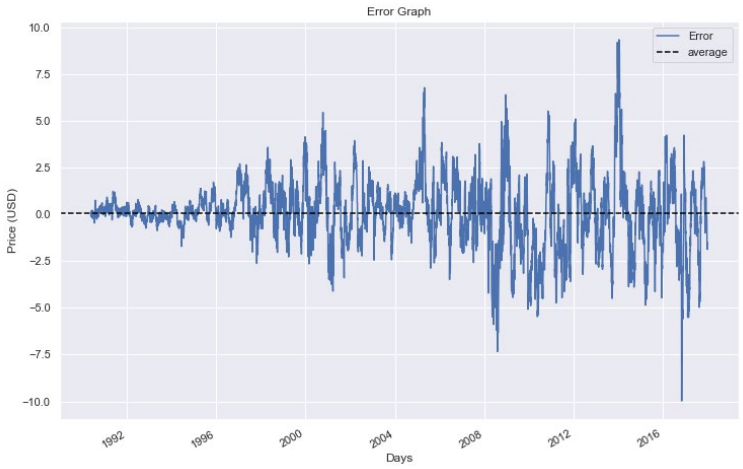
**Figure 3: Exxon, Chevron & β_Chevron**




The next step is then to find the error between Exxon and β_Chevron, $Y(t) - \beta X(t)$. Figure 4 below shows an error graph where the series seemingly fluctuates along the average line. Even though it is not a stationary time series because the variance is not constant, it is obvious that the error is a mean reverting time series with mean equal to zero.

The series is a result of $Y(t) - \beta X(t)$, where $Y(t)$ is Exxon and $X(t)$ is Chevron. Therefore, as the series go above the mean, Exxon will have a bias to move downward and as the series go below the mean, Exxon will have a bias to move upward.

**Figure 4: Error between Exxon & β_Chevron**




Finally, the half-life period of the error time series is calculated. Let the error time series to be $E(t)$ and since $E(t)$ is a mean reverting time series, the series can be described as a stochastic process with Ornstein-Uhlenbeck formula for mean-reverting process:

$$dE(t) = (\lambda E(t-1) + \mu)dt + d_\varepsilon$$

Where $d_\varepsilon$ is taken as the noise with which is normally distributed. The expected value of the equation is calculated as follows:

$$E[E(t)] = E_0 e^{\lambda t} - \frac{\mu}{\lambda}(1 - e^{\lambda t})$$

This expression shows that $E(t)$ will decay to value of $-\frac{\mu}{\lambda}$ (in this case it is 0 since $\mu = 0$) with a half-life period of $-\frac{log(2)}{\lambda}$. For the series to be mean reverting, $\lambda < 0$. Since Exxon and Chevron is determined to be a strong cointegrated pair from the previous section, it is expected to be a negative value.

The exact value of $\lambda$ was obtained by taking the linear regression of $\Delta E$ against $\Delta E(t-1)$. The obtained $\lambda$ has a value of -0.053173 and the half-life period is equal to approximately 16 days.

Armed with the new information, the stock price of Exxon is set to be predicted for a window period of 16 days into the future. Using this strategy takes advantage on $E(t)$ bias to revert to its mean. Therefore, it makes sense to predict on the time period for which $E(t)$ is reverting.

With this the prediction problems are reformulated as the following:

Problem 1:

$$Sign(Y(t) - Y(t-16)) = f(X_1(t-16), \dots, X_n(t-16))$$

Problem 2:

$$\frac{Y(t) - Y(t-16)}{Y(t-16)} = f(X_1(t-16), \dots, X_n(t-16))$$

Where $Y(t-16)$ is Exxon stock price from the past 16 days and $X_n(t-16)$ is the predictor variable $n$ at time $t-16$.

### 3.2 Variables

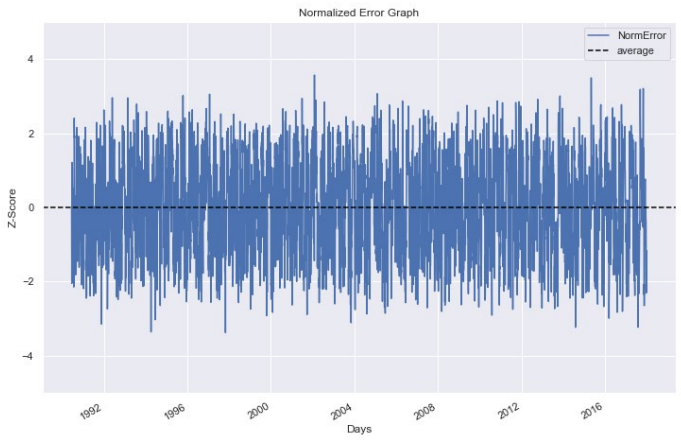
The goal of the project is to examine if the time series properties of Exxon can be a good predictor and to explore whether incorporating a related stock like Chevron can be used to further improve the prediction. Therefore, the set of predictors selected must contain predictors which describes Exxon and the relationship between Exxon and Chevron.

The predictors which describe Exxon are its past 50-day moving momentum and 50-day moving volatility. Momentum is acquired by calculating the percentage gain and volatility is calculated by taking the standard deviation of the returns.

The predictors which describe relationship between Exxon and Chevron are the past 50-day moving momentum and volatility of Chevron, 50-day moving correlation, 50-day moving ADF p-value and the normalized error. The error series value must be normalized so that it can be used as a predictor. The moving window used to normalize the error value will be equal to the half-life period, 16 days.

$$Normalized\ E(t) = (E(t) - MA_{16}(t))/MSD_{16}(t)$$

**Figure 5: Normalized Error**



The variables are summarized below.

Response variables:

- Problem 1 (Classification): Direction, {“UP”, “DOWN”}
- Problem 2 (Regression): Percentage change in price

Predictor variables:

- 50-day Exxon momentum
- 50-day Exxon volatility
- 50-day Chevron momentum
- 50-day Chevron volatility
- 50-day correlation
- 50-day ADF p-value
- Normalized error

## 3.3 Machine Learning Models

The generated data frame can be evaluated under two main methods: Regression and Classification. Each method addresses and tackles its own set of issues. Applying regression on the data frame will allow for the price prediction of the stock, and classification will allow for the prediction of the direction of the stock. Different models listed below are applied to solve these two approaches.

### 3.3.1 Recurrent Neural Network

The model architecture employed for this problem uses an input Long Short-term Memory (LSTM) layer. Since the input training data was arranged in chronological order, the LSTM layer was used as it helps to account for any long-term time-dependencies within the data. In order to input the collected data into the LSTM layer, the 2-dimensional array of data is converted into a 3-dimensional array by appending each subsequent *n*-day look back slices upon each other.

The team has experimented around with the neural network model by adding extra hidden layers with varying neurons, along with different levels of dropout to avoid the case of overfitting. Lastly, there is an output layer appended to yield the predicted result.

The optimiser used was the “Adaptive Moment” estimation (Adam). Adam optimiser is used as it combines the advantages of two other extensions of stochastic gradient descent (SGD), specifically Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). The parameters used for the Adam optimiser are as follows:

$$learning\ rate\ (\alpha) = 0.001, beta_1(\beta_1) = 0.9, beta_2\ (\beta_2) = 0.999, decay = 0.01$$

### 3.3.2 Random Forest Algorithm

Random forest algorithm was used as both a classification and regression algorithm in the predictions. Basically, this algorithm consists of an ensemble of decision trees and its works well being a supervised machine learning algorithm. Without the need for hyper-parameter tuning to achieve decent results, it saves on computational time and complexity.

Random forest regression was done with the dependent variable as the label 'price_change', the closing price of the stock and the features to be constrained initially to only Exxon-related variables, i.e. Exxon price 13 days ago and 100-Day Exxon momentum. The random forest was executed in sci-kit learn library with the hyper-parameters tuned as follows:

$$n_{estimators} = 1000, \qquad random_{state} = 42,$$

where $n_{estimators}$ is the number of decision trees used in the forest and $random_{state}$is the random seed generated to ensure replicability of results. The algorithm was then run to fit on the test set to get the regression or classification results.

Subsequently, the constraint for features are relaxed to gently introduce other features that are associated with the pair variable. This is done to analyse and evaluate the effect of including information from the stock's pair on the prediction. This will be explained further under the 'Results and Discussion' section.

### 3.3.3 Linear Regression

For regression to predict the numerical value of stock price, another machine learning model used is the ordinary least squares method. This creates a model which minimises the sum of squared errors between the observed data and predicted data. Linear Regression model was trained using the sci-kit learn library on the training set of data. The trained model is then used to fit against the test data.

### 3.3.4 Logistic Regression

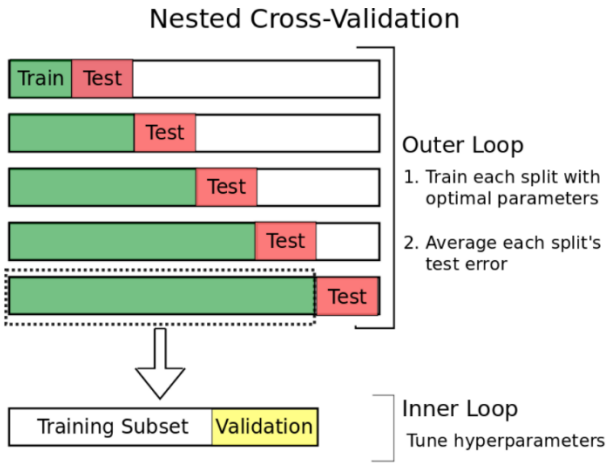
Since the output of a classification method is binary, logistic regression is another model that is conducted and appropriate for classification. This model utilises a sigmoid function to fit in the features and map it to a value between 0 and 1. It is easily implemented using sci-kit learn library with the parameter of C, the inverse of regularisation strength, set to a value of 5.

# 4 Evaluation of Methodology

## 4.1 Nested Cross-Validation

Datasets involving stock prices and its associated features are naturally in a time-series format. When dealing with time series data, the traditional methods for splitting into training and testing sets such as k-fold cross validation is discouraged. To account for temporal dependencies within the chronologically structured data-frame, performing nested cross-validation is necessary. The team's strategy is to incorporate a nested cross-validation model which is illustrated graphically in Figure 6.

**Figure 6: Nested Cross-Validation**




The method utilizes hold-out cross-validation where a subset of the data (split temporally) is reserved for validating the model performance. In this case $k = 5$ and there is a total of $k + 1$ subsets or splits in the dataset. This removes bias from any arbitrary train-test split in conventional k-folds validation methods, resulting in an almost unbiased estimate of the true training and test error.

## 4.2 Regression

The label variable 'price_change' consists of continuous price values.

Mean of squared errors (MSE) is used to evaluate the models' predictions of 'price_change'. MSE measures the average of the squares of the residuals, which is the difference between the predicted value and its corresponding actual value. It is calculated as shown below:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{X}_i - X_i)^2$$

The lower the MSE value, the more effective of the proposed model is at predicting 'price_change'.

### 4.3 Classification

The label variable 'Direction' consists of discrete direction values.

For RNN model, Categorical Cross Entropy is used to evaluate the model's loss function value. It is calculated as the double summation of the indicator function of the *i*-th observation belonging to the *c*-th category for N observations and C categories, as shown below.

$$Categorical\ Cross\ Entropy = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} 1_{y_i \in C_c} \log p_{model}[y_i \in C_c]$$

The model evaluation method is based on accuracy, which is defined as:

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total}$$

Thus, the higher the accuracy, the greater the number of true positives and negatives the model can predict over the entire testing set, and therefore the better the model.

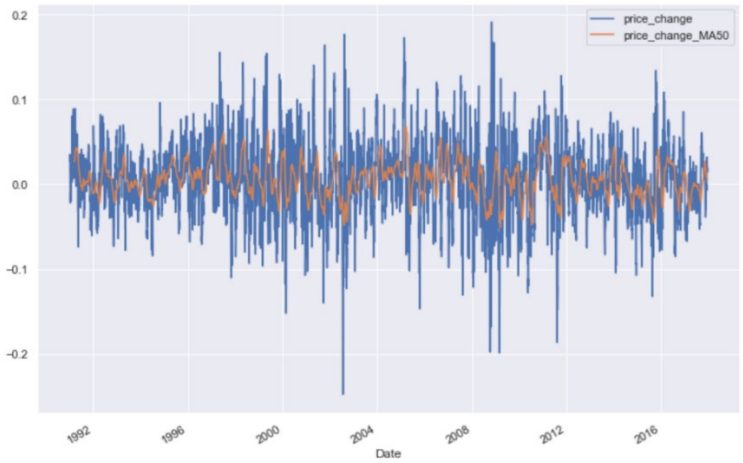
# 5 Results and Discussion

## 5.1 Benchmark

While evaluating the results from classification and regression, it is important to establish a benchmark that serves as a basis of comparison. Stock market prices reflect the aggregated sentiment of many buyers and sellers in the market, and the numerous driving forces that affect the market accounts for the unpredictability and fluctuations in the market. Given the randomness and uncertainty, the benchmark for classification of the stock direction is thus taken as 50%.

The benchmark for regression is calculated by finding the squared difference between the price change index and the 50-day moving average. The benchmark comes out as 0.00197 for the Exxon stock, to be compared against the mean of square errors by the algorithm models.

**Figure 7: Price change & 50-day moving average of price change**




## 5.2 Classification

### 5.2.1 Recurrent Neural Network

Table 3 shows the different configurations used when tuning of the Recurrent Neural Network (RNN) for predicting the direction variable. The aim is to find the model configuration that produces the lowest mean test accuracy, calculated by taking an average of the test accuracy evaluated for each iteration of the nested cross-validation method.

No dropout is applied to the Recurrent Neural Network models as the Long short-term memory layer consist of the forget gate, that replicates the effect of having dropout regularisation. Mini-batch sizes used for training all classification models is set at 32. Early stopping callback is set at having a patience value of 1 for the 'val_loss' output (i.e. while training the model for a pre-defined number of epochs, if the

validation loss of a particular epoch is higher than the validation loss of the previous epoch, terminate the model training process).

Initially, when the default learning rate of 0.01 was applied, the resulting validation loss values tend to fluctuate and even increases over each epoch. Therefore, the learning rate associated with the Adam optimizer is kept fixed at 0.001 as to keep the value of the validation loss value stable or decreasing.

**Table 3: Tuning of hyperparameters of RNN for classification**

| S/N | Model Parameters/ Hyperparameters | | | Train using Exxon variables only | | Train using all variables | |
|---|---|---|---|---|---|---|---|
| | Model Layer(s) | Optimizer | Epoch Number | Mean Training Accuracy | Mean Test Accuracy | Mean Training Accuracy | Mean Test Accuracy |
| 1 | LSTM (64), Dense (2) | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, decay=0.01) | 10 | 0.5828449688 | 0.5609733731 | 0.5839101933 | 0.5639822227 |
| 2 | LSTM (64), Dense (2) | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, decay=0.01) | 30 | 0.5846148803 | 0.5581415147 | 0.5918944608 | 0.5505290174 |
| 3 | LSTM (64), Dense (2) | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, decay=0.01) | Early Stopping | 0.5824024909 | 0.5609733731 | 0.5830317928 | 0.5615043466 |
| 4 | LSTM (128), Dense (2) | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, decay=0.01) | 10 | 0.5830317928 | 0.5615043466 | 0.5824287118 | 0.5569021072 |
| 5 | LSTM (128), Dense (2) | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, decay=0.01) | Early Stopping | 0.5824024909 | 0.5609733731 | 0.5824254342 | 0.5602654085 |
| 6 | LSTM (128), Dense (2) | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, decay=0.01) | 10 | 0.583828253 | 0.5634512493 | 0.5826384791 | 0.548405906 |
| 7 | LSTM (64), LSTM (64), | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, | 10 | 0.583139954 | 0.5613273554 | 0.5827302523 | 0.5597344351 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Dense (2) | decay=0.01) | | | | | |
| 8 | LSTM (64), LSTM (64), Dense (2) | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, decay=0.01) | Early Stopping | 0.5824024909 | 0.5609733731 | 0.5844706653 | 0.5574335501 |

Comparing across the different models in Table 3, the results show that training the model using all the variables to predict the change of direction does not always result in better mean test accuracy than just training using Exxon variables (50-day Exxon momentum, 50-day Exxon volatility).

It is observed that a few epochs are needed for this model as increasing the number of training epochs tends to decrease the mean test accuracy. Early stopping is also shown to be unreliable in the prevention of overfitting/underfitting in certain models.

Besides tuning number of training epochs, the number of hidden units and model architecture was also tweaked to see if it leads to an increase in mean test accuracy. Results from Table 3 show that increasing number of hidden units in each LSTM layer does help improve mean test accuracy for training using Exxon variables but worsens the mean test accuracy for training using all variables. This phenomenon also occurs when the additional hidden LSTM layers are added to the model architecture. The results seem to suggest that with more input features, a simpler model is better than a complex model.

The models that resulted in the highest test accuracy is highlighted above. Both models achieved better results than the specified benchmark for classification of 50%, implying that these models do a better job when compared to the randomness of the financial markets.

### 5.2.2 Logistic Regression/Random Forest Classification

The accuracy on the test set was obtained by averaging all the folds during the nested cross validation. The results are summarised in Table 4.

**Table 4: Accuracy Comparison for Classification Algorithms**

| Features | Accuracy | |
|---|---|---|
| | Logistic Regression | Random Forest |
| Exxon variables only | 0.568 | 0.520 |
| All | 0.570 | 0.529 |
| Exxon + ADF, Corr, Momentum | 0.570 | 0.531 |
| Exxon + Corr, Momentum, Normalised | 0.563 | 0.526 |
| Exxon + Volatility, Normalised, Momentum | 0.565 | 0.510 |

Comparing accuracies across the classification models, it is observed that the addition of pairwise related features slightly improves the accuracy of the model. This implies that using additional variables that describe the pair relationship helps in the prediction of the direction of Exxon to a certain degree. Overall, Logistic Regression appears to yield the highest accuracy, followed by RNN with a marginal difference.

## 5.3 Regression

### 5.3.1 Recurrent Neural Network

Table 5 shows the different configurations used while tuning the Recurrent Neural Network (RNN) for the problem of predicting the 'price_change' label value. The aim is to find the model configuration that produces the lowest mean test MSE (mean squared error); calculated by taking an average of the test MSE evaluated for each iteration of the nested cross-validation method.

Similar to training for the problem of classification, no dropout is applied to the Recurrent Neural Network models as the Long Short-Term memory layer consist of the forget gate which replicates the effect of having dropout regularisation. Mini-batch sizes used for training all classification models is set at 32. Early stopping call-back is set at having a patience value of 1 for the 'val_loss' output.

The learning rate associated with the Adam optimizer is kept fixed at 0.001 as to keep the value of the validation loss value stable or decreasing.

**Table 5: Tuning of hyperparameters of RNN for regression**

| S/N | | Model Parameters/ Hyperparameters | | | Train using Exxon variables only | | Train using all variables |
|---|---|---|---|---|---|---|---|
| | Model Layer(s) | Optimizer | Epoch Number | Mean Training MSE | Mean Test MSE | Mean Training MSE | Mean Test MSE |
| 1 | LSTM (64), Dense (1) | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, decay=0.01) | 50 | 0.001593219395 | 0.002217678547 | 0.001553815157 | 0.002361817287 |
| 2 | LSTM (64), Dense (1) | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, decay=0.01) | Early Stopping | 0.001608783213 | 0.002240133612 | 0.001662966753 | 0.002732830443 |
| 3 | LSTM (64), Dense (1) | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, decay=0.01) | 100 | 0.001591669534 | 0.00221712216 | 0.001526552217 | 0.002334856291 |
| 4 | LSTM (64), | Adam (α=0.001, $\beta_1$=0.9, | 200 | 0.001589702674 | 0.002217934673 | 0.001514534965 | 0.002348642368 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Dense (1) | $\beta_2$=0.999, decay=0.01) | | | | | |
| 5 | LSTM (128), Dense (1) | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, decay=0.01) | 100 | 0.001590733505 | 0.002215046662 | 0.001528470717 | 0.00235556351 |
| 6 | LSTM (64), LSTM (64) Dense (1) | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, decay=0.01) | 50 | 0.001595790748 | 0.002218803218 | 0.001545996942 | 0.002351842176 |
| 7 | LSTM (64), LSTM (64) Dense (1) | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, decay=0.01) | 100 | 0.001591762871 | 0.002218085737 | 0.001523591896 | 0.00232685599 |
| 8 | LSTM (128), LSTM (128) Dense (1) | Adam (α=0.001, $\beta_1$=0.9, $\beta_2$=0.999, decay=0.01) | 100 | 0.001591905667 | 0.002216051246 | 0.001528278493 | 0.002351083193 |

Comparing across the different models in Table 5, it seems to show that training the model using all the variables to predict the change of direction always result in lower higher MSE than just training using Exxon variables (50-day Exxon momentum, 50-day Exxon volatility).

It is also observed that more epochs are needed for the all the models to improve mean test MSE. Early stopping is also shown to be unreliable in prevent overfitting/underfitting in certain models as the resultant mean test accuracy is lower than without applying early stopping.

Besides tuning number of training epochs, the number of hidden units and model architecture were also tweaked to see if it leads to an increase in mean test MSE. Results from Table 5 show that adding more hidden LSTM layers results in lower mean test MSE when training using all variables, while training using only Exxon variables with additional LSTM hidden layers does not improve test accuracy much. Increasing the number of units within the LSTM layer is shown to improve mean test MSE when training on just Exxon variables, while it has a less positive effect when trained on all variables. The models that resulted in the highest mean test MSE is highlighted above.

### 5.3.2 Random Forest/Linear Regression

For each of the cross-validation sets, both random forest and linear regression algorithms are used to fit the model, and the mean of squared errors tallied for the predictions on the test set is summarised in Table 6.

**Table 6: MSE Comparison for Regression Algorithms**

| Features | Mean of Squared Errors (MSE) | |
|---|---|---|
| | Random Forest | Linear Regression |
| Exxon variables only | 0.00259 | 0.00223 |
| All | 0.00251 | 0.00229 |
| Exxon + ADF, Corr, Momentum | 0.00253 | 0.00227 |
| Exxon + Corr, Momentum, Normalised | 0.00250 | 0.00224 |
| Exxon + Volatility, Normalised, Momentum | 0.00247 | 0.00226 |

Comparing the mean of squared errors (MSE) across the different models, it is observed that all three regression algorithms gives similar results for the squared errors. Although the models did not surpass the specified benchmark MSE for regression of 0.00197, it is still reasonably close to the benchmark value.

Visualisations of the regression algorithms that gives a comparison between the actual price and predicted prices are shown in Figures 9, 10 and 11. The plots depicts actual pre-value of Exxon in an 80-20 split test set against the regressed prices values. In the plot, the behaviour of the predicted prices models after the actual price to a satisfactory degree.

**Figure 9: Model S/N 5 trained using only Exxon variables with mean test MSE of 0.002215046662**

**Figure 10. Model S/N 7 trained using all variables with mean test MSE of 0.00232685599**

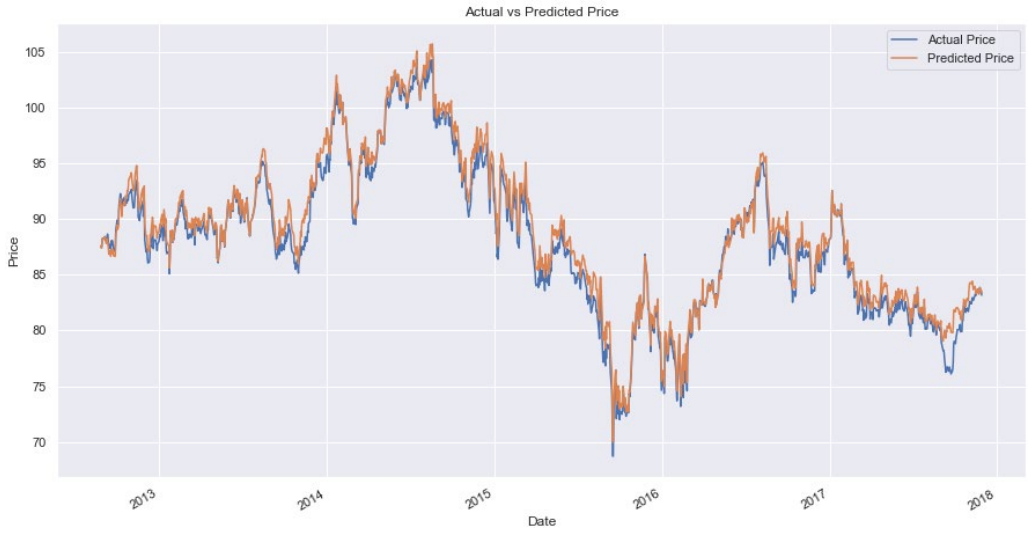
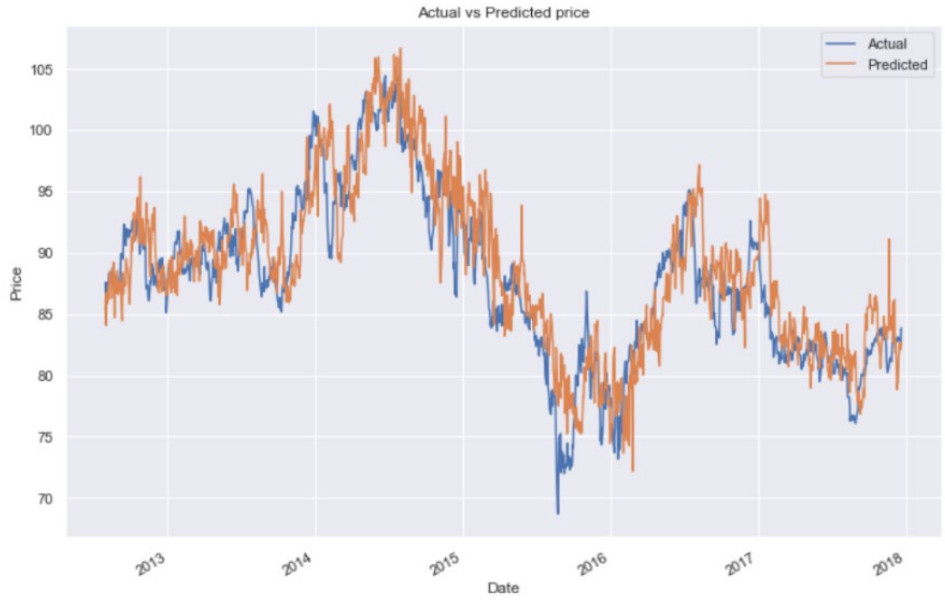



**Figure 11: Exxon stock prices, Random Forest Algorithm, Predicted Vs Actual**



### 5.4 Limitations

In the project scope, the predictors taken into consideration are based solely on the closing price. Therefore, this limits the type of data which can be used for prediction. To improve on the predictive analytics of the models, other data types commonly associated with financial instruments such as the econometrics, stocks financials fundamentals and news sentiment can be explored. Additionally, instead of relying on one model for prediction, researching on ensemble learning method which takes into consideration multiple algorithms can be done.

## 6. Conclusion

Predictive analytics in time series market forecasting remains as one of the most important methodology in finance. Predictive analytics helps businesses and individuals make better informed decisions, and the context of stock prediction remains popular in the realm of data science. In the study, it is found that using concepts such as mean reverting series and correlated stock features while evaluating the results based on appropriate machine learning algorithms aids in the analysis of prediction of stock prices. The dynamic nature of predictive analytics implies that there are no hard and fast rules to approach a problem, and in this case, using related security as additional feature predictors yielded satisfactory predictions. In the future, addressing limitations and incorporating new methods of models could potentially improve predictive results.

# References

Cochrane, C. (2018). *Time Series Nested Cross Validation*. Retrieved from Towards Data Science: https://towardsdatascience.com/time-series-nested-cross-validation-76adba623eb9

Corrius, J. (2018). *Simple stationarity tests on time series*. Retrieved from Medium: https://medium.com/bluekiri/simple-stationarity-tests-on-time-series-ad227e2e6d48

Honchar, A. (2017). *Neural networks for algorithmic trading. Correct time series forecasting + backtesting*. Retrieved from Medium: https://medium.com/@alexrachnog/neural-networks-for-algorithmic-trading-1-2-correct-time-series-forecasting-backtesting-9776bfd9e589

lcampos. (2017). *Forecasting S&P 500 using Machine Learning*. Retrieved from https://quantdare.com/forecasting-sp-500-using-machine-learning/