# Price Pattern Recognition in Index Futures Market

MATH3599 - Professional Practice for Mathematical Sciences
Session 2 2022

Lennox Hemingway, Rajiv Mehta and William Reynolds ∗
45107793; 45433062; 45378711 ∗

*In financial markets internationally, greater than 60 percent of trading activities across all asset classes rely on automated and algorithmic trading in favour of human traders. Specialized programs reliant of algorithms and learned patterns autonomously buy and sell assets with a cardinal aim of generated a low risk, positive return over a specified time horizon. This aim of this report apply various machine learning algorithms implemented in Python on one minute Index Equity Futures Market data across three major international indexes. The greater the accuracy or hit rate of the machine learning model, on average, indicates the model's ability to currently predict closing price movements reliably.*

# Contents

# 1 Introduction

## 1.1 Project Overview

Kensho data is a newly established business that specialises in providing quantitative and strategic solutions to clients within the financial industry. The quantitative solutions Kensho provides are largely in the form of algorithmic trading implementation and methodology. Analysing financial data and developing algorithms to locate patterns and generate predictions is a standard focal point for quantitative trading firms. Kensho Data is currently expanding within this domain. This report seeks to provide an overview of the performance of various rudimentary and advanced price prediction techniques. Accordingly, this report will allow Kensho to expand upon the methodologies they see potential in to help provide solutions to their clients.

This analysis forms a critical step for Kensho's development and expansion. By providing an overview of the performance of various techniques, it saves Kensho time and cost that can be directed to the development of the algorithms that exhibit the greatest potential. As Kensho is looking to expand in the data and quantitative solutions market, having readily available programs that give results will give Kensho Data the ability to acquire more clients and grow in the increasingly competitive financial sector. Quantitative methods for finance in recent years have developed into a burgeoning field of research. However, due to the potential profitability associated with such developments, it is shrouded in secrecy, where the best performing methods are private and are intellectual property, owned by the individual trading firms. Furthermore, the data itself is often not publicly available either.

Nevertheless, there is a vast amount of research and code available in the public domain which utilises open-source data and elementary methods. GitHub has a range of finance pattern projects including basic machine learning, regressions and forest models using openly sourced data. The accuracy of projects is lower; however, it serves as a great reference point that can be further developed and improved upon. For high-frequency data, there is less freely available information. High-frequency trading is currently at its apex of popularity, where secrecy is a core component to develop and maintain profitability.

### 1.1.1 Project Background

The goal of the analysis is to develop models that can take financial data as an input and make predictions on the future values. The goal of these models is to have an accuracy rating greater than 50%, meaning on average it is more likely to make more correct predictions, than not. Kensho has provided the data for this analysis, which is private high-frequency foreign exchange and index futures data. Although this data was stipulated to be used as input for the model, there was no set methodology or any particular model/algorithm required, so long as the method and model are able to obtain a net positive accuracy. The recommendation from Kensho was to develop a simple minimum viable product model and develop upon this. Thus, ensuring that the analysis conducted included modelling which Kensho could further develop, if it is deemed suitable.

### 1.1.2 Index Future Markets

Index futures contracts are standardised contracts that can be bought, sold and traded online, where the underlying asset is the index fund. A futures contract is a derivative instrument that is an agreement to buy or sell depending on the position taken, of an underlying asset for a particular price at a particular price. These contracts are usually for financial instruments or assets. In our case, this instrument is the index fund.

There are two appeals to future contracts. The first, and perhaps most important, is risk mitigation. A futures contract allows the investor/business to "lock in" a particular price to buy or sell a good. This reduces the volatility of their profits and losses and smoothens cash flows.

The second and most relatable to this report, is speculative investing. A futures contract can allow the investor to amplify earning potentials, by simply trading these contracts without ever holding onto them until they need to be exercised.
To illustrate this. Consider the scenario where an investor owns a futures contract to purchase 20 shares tomorrow, at a price of \$100 each, for a total of \$2000. However, the current price of these shares on the market is \$150 each. The investor could wait until tomorrow and buy the shares, or they could sell the contract which would be worth the difference.

$$\begin{aligned}\text{Contract} &= \text{Quantity} \times (\text{Market Price} - \text{Contract Price}) \\ &= 10 \times (\$150 - \$100) \\ &= \$500\end{aligned}$$

Without needing to trade any shares, the investor can sell this contract and make a \$500 profit. Index Funds are a weighted basket of shares or assets. These are critical, as it can put a measure on the performance of certain groups of shares, in addition to allowing investors to obtain a financial asset that will track the movement of these groups of shares without needing to purchase each one individually. These weighted baskets are often in the form of top companies in countries. For our data the index funds Kensho provided were:

- FTSE 100: The top 100 companies listed on the London Stock Exchange
- ASX 200: The top 200 companies listed on the Australian Stock Exchange
- S&P 500: The top 500 companies listed on American stock exchanges.

An important consideration is that index equity futures markets are large in volume and actively traded markets daily, thus providing ample data to be analysed on an ongoing basis.

# 2 Methodology

## 2.1 Data

As Kensho is a financial data aggregator and distributor, therefore the data required no cleaning and was standardized across time stamps. No data processing was required and the data did not include missing or erroneous values and was in the correct format. Kensho provided data from equity index futures contracts. The specification was three different indexes as previously mentioned, FTSE 100, ASX 200 and SP 500. Each of these index future contracts spanned over the previous six years with one-minute intervals. The result was three datafiles each with over one-million lines and seven variables.

## 2.2 Statistical Methods

To determine the appropriate statistical model, it was necessary to perform preliminary data exploration. Given the large quantity of data, the periodicity and volatility thereof, it was clear to produce a meaningful solution required a dynamic approach.This problem can be approached from 2 perspectives, fundamentally, using financial analysis or quantitatively, using the tools of computer science and statistics. Several rudimentary statistical method techniques were investigated, however, these techniques were not viable to support the required outcome, price prediction. Given the periodicity and volume of the data, many statistical method techniques were not valid to be applied in order to solve this problem. It was clear that creating a robust and dynamic solution required the use of supervised or unsupervised machine learning. Supervised machine learning is an approach applied exclusively to labelled datasets, where a mapping function is found between the input variable/s and output variable. The dataset is designed to train and supervise the algorithms into predicting labeled outcomes or classifying data accurately and measure the accuracy using regression and classification. Conversely, Unsupervised Machine Learning uses algorithms to analyze unlabeled data sets, inferring or predicting patterns hidden within the data through clustering using distinguishable characteristics. Unsupervised machine learning algorithms are regarded as computationally complex and require large datasets compared to unsupervised algorithms.

The aim was to generate an algorithm which could be applied to any data set with similar values, a close price and a time stamp. The core principle being to predict a value or direction using only information leading up to that point in time. This problem can be solved in 2 methods. Firstly, to predict the direction of the price for the next period. Thus, the dependent variable is the categorical, forming a binary classification problem for the supervised machine learning algorithms. The next approach that can be taken is to predict the exact value of the t-th period's price using information from $t-1$. Hence, the dependent value is instead a continuous variable and therefore the problem is no longer of classification. To achieve this, the use of unsupervised machine learning using deep learning algorithms, a subset of machine learning (IBM, 2022). Deep learning differs from traditional machine learning as it eliminates aspects of the pre-processing of data and has the ability to ingest and process unstructured data, removing most of the manual intervention.

It is important to note that the chosen statistical package employed to perform this analysis was python. This was selected as it offers advanced data analysis capabilities and industry standard machine learning capabilities ready for scalable production environments, whereas R is purpose built for statistical modeling (IBM, 2022a).

# 3 Model

With two divergent approaches to solve this problem, unsupervised and supervised machine learning, it was clear that the model would be split into 2 distinct phases with achieving a similar outcome using different methodology and complexity. Phase 1 focused on the application and comparison of various supervised machine learning models/algorithms and the effectiveness in predicting price direction. These models were: Logistic Regression, Support Vector Classifier, Random Forest Classifier and Linear Discriminant Analysis. Phase 2 of the modeling, utilises unsupervised deep learning by constructing a Long Short-Term Memory Recurrent Neural Network to predict the exact closing price value. Ultimately, this enables a comparison between the 2 approaches, in terms of error, accuracy and complexity.

For both models rely on fitting a training set, which is then validated using a testing set. An advantage of this approach, retaining a set of data distinct from the training set, allows for the evaluation and comparison of the predictive performance of each model, reducing the risk of overfitting (Joseph, 2022). The next logical question is what is the optimal split for this? From the literature, it is clear there is no universal guidance on the optimal ratio for any given data set. However, the Pareto principle, 80:20 is the most common ratio employed by practitioners across a wide variety of data sets (Joseph, 2022). The following ratios were employed across each phase:

**Phase 1 - Supervised learning:** 80/20 - train test

**Phase 2 - Unsupervised learning:** 80/10/10 - train validation test

## 3.1 Phase One

### 3.1.1 Model Set-up

With the desired dependent variable known, that is a prediction price direction at time $t$, it was required to determine the independent variable that could be used to predict the outcome. Whilst there were numerous approaches that could be explored from our data set and various, through research and testing, it was determined that the following data points were most influential in predicting price direction: the 1 and 2 period lagged returns computed using the close price and volume percentage change. Thus, the Phase 1 model was constructed as:

**Dependent** - Price Direction
**Independent** - 1 period lag return, 2 period lag return and 1 period volume percent change.

It is important to note the following assumption, where computed independent and dependent variables resulted in an error, these rows were omitted from the input data. Next, the model was set up to flexibly ingest data from various indices, at different start dates and input sizes. The subsequent phase 1 model analysis has been performed with following standardized inputs across each of the indices.

**Start Date:** 17:39:00 on 29 - 12 - 2016
**Minutes (Rows) Used:** 250,000
**Test/Train Split:** 80% - 20%

### 3.1.2 Supervised Machine Learning Models

**Linear Discriminant Analysis:** Linear Discriminant Analysis is a supervised machine learning algorithm used for classification problems, whereby it separates 2 or more classes, identifying a linear combination of the classes or features and modelling the group differences. The resulting linear combination of classes can be utilised as linear classifier and for classification (MachineLearningMastery, 2016). It provides a viable alternative to the limitations of logistic regression. For the Phase 1 model, similarly, there were two binary classes.

**Logistic Regression:** Logistic regression is a supervised machine learning algorithm, which is used for the prediction of categorical dependent variables, $P^d$, from a set of independent variables, $P_1$, $P_2$, $V_1$. It estimates a continuous quantity, or the probability of an event occurring, where probability between zero and one,is compared to a threshold allowing the new data to be classified. For the Phase 1 model, it was applied in the binary case, positive and negative pride prediction.

**Support Vector Machine Algorithm:** Support Vector Machine is a supervised machine learning algorithm, used to solve classification problems. The aim of the algorithm is to create a best line/decision boundary or hyperplane that can separate an n-dimensional space into classes such that new data can be categorized, where the dimension of the hyperplane is dependent on features of data. The algorithm determines extreme points or support vectors to create the hyperplane. The distance between support vectors and the hyperplane, known as the margin, is maximized to find the optimal hyperplane, Appendix 7.2.1 For our dataset there are 3 input features, thus the hyperplane is a two dimensional plane.

**Random Forest Classifier** The Random Forest Classifier is a supervised machine learning algorithm applied to solve classification problems. It constructs multiple decision trees using the "$m$" features randomly from the data. It constructs a decision tree, where $m < k$, where "$k$" is the total number of features). This is repeated $n$ times to create $n$ decision trees with random combinations of $k$ features. Using each $n$ decision tree, the random variable is passed in the model in order to predict and store outcome from the $n$ decision trees. As the dependent variable, $P^d$ is categorical, each of the $n$ trees in the random forest, predicts the category, or price direction, *up or down*, for which the new information belongs by majority vote.

## 3.2 Phase Two

### 3.2.1 Long Short Term recurrent Neural Network

A 'Recurrent Neural Network' (RNN) is an unsupervised machine learning algorithm, where information from previous outputs is used in the computational step for the prediction of the next output and is commonly used for predicting sequence data. This is done by the creation of artificial neurons. In text prediction, an RNN would store all information that has previously been used, so that the next time a user writes a sentence that is similar the model can predict the next words and suggest them to the user.

As the model predicts future values using all information from previous outputs, this type of model is an almost perfect choice to predict the trading of the Equity Index Futures Markets. Though, this causes an underlying problem with the model when predicting using time series data called the 'Vanishing Gradient Problem'. The vanishing gradient problem occurs when too much data has been added into the model which in turn causes the prediction models gradient to become very small.

When this occurs, the model finds it harder to learn new patterns and trends and in turn no longer be able to make accurate future predictions.

To solve the vanishing gradient problem, the chosen model 'Long Short-Term Model' (LSTM) was designed to be capable of learning long-term dependencies. This type of machine learning algorithm is commonly used in text prediction where key words in sentences are stored in memory which are key indicators for potential future words in the sentence. For instance, in the sentence "John's favourite food is pasta", key words such as 'John', 'food' and 'pasta' would be stored and used in future predictions so that when a new sentence is formed such as "John feels like getting food today, he suggests..." The following suggestion would appear to be 'pasta' due to previous inputs. The LSTM introduces a state variable into the model.
This state variable is known as the LSTM cell, which allows the model to choose which information to either update, delete and output for future predictions. When it comes to the data provided, this is particularly useful as there are large periods of minutes where the price does not change, making the LSTM the optimal choice for market value predictions.

### 3.2.2 Model Setup

Due to the nature of this model and the restricted time frame for completion of this project, a univariate LSTM was created which would make predictions on the 'Close' value over a specified period. This model, has five main parameters that need to be changed depending on the testing that is taking place:

- Data – The data sheet and the column the user wants to test on can be changed depending on the testing requirements;
- Input parameters – The model has been set up so the user can input specified data range to be tested;
- Number of Lags – The model can be customised to have more lags used to predict a certain time frame. The default is set to 3.
- Number of EPOCHS – The number of times that the model will run through the training data and learn patterns. Standard starting value is 100, and too high of a number will cause the model to overfit.

This model is more robust as it does not test whether the predicted value increases or decreases at a certain time point, rather it outputs predicted values. This is advantageous as it accommodates the case where the 'Close' Price remains constant across multiple time periods. For testing purposes, the model is set up to tabulate the predicted and observed values showcasing the head and tail of the testing data.

# 4 Results

## 4.1 Phase One

To measure the success of each of the four supervised machine learning models applied, three core measures can be utilized. Firstly, the Hit Rate, which is the percentage of the total number of correct predictions. Secondly, the Misclassification Rate, the percentage of incorrect predictions. Lastly, the Confusion matrix, a classification table that categorizes the predictions made by each algorithm and compares it to the actual value in the data set for a given point in time. The classifications are classed into four categories: True Positive, False Positive, False Negative and True Negative, Appendix 7.2.3.

The performance of each supervised machine learning model remained consistent within each index, achieving an approximately equivalent hit rate. For the ASX, FTSE and S&P, the results were approximately 73%, 60% and 70% correct predictions respectively, Appendix 7.2.3. It is important to note the context of the problem: to predict a price at time, $t$, using the available information at time, $t-1$. If an algorithm predicts a true positive price direction then, the algorithm user could theoretically place a trade to capture an increase in value and profit. Conversely, where an algorithm predicts a false negative misclassification, it does not necessarily adversely affect the user, as positions are frequently traded and held for short durations. Hence, representing a missed opportunity or the value of an owned future contract held increases. Accordingly, the high misclassification rates across each model and index, approximately $30\% - 40\%$, are primarily composed of false negatives; it may not pose a significant limitation, Appendix 7.2.3. When the time complexity, or the computational complexity which determines the quantity of time required to run each algorithm, is considered, the Logistic Regression and Linear Discriminant Analysis model are more favourable. The Support Vector Machine and Random Forest Classifier require approximately 30 minutes of run time to produce a result, which is insufficient in the context of this problem. It is clear that increasing the Hit Rate and reducing misclassifications requires a more advanced approach and further testing of the input variables, each of which may vary from index to index, Appendix 7.2.3.

## 4.2 Phase Two

Measuring the success of the LSTM is quite a difficult task and is very subjective due to the nature of the algorithm. As the algorithm is outputting predicted values which are compared to the observed values in the static dataset, it is important to look at the predictors to determine the best choices to get the closest predicted values to what was observed.

Appendix 7.3.2, depicts the observation data split that is used for the training, validation and testing of the input data. It further indicates the range of values within the data, which is useful to understand what occurred during the specified time frame. In this model, a time frame of 6 months has been employed.

Appendix 7.3.2, highlights the Mean Absolute Errors (MAEs) at each individual EPOCH. In order to check to see if the model is overfitting or not, when a higher number of EPOCHs are implemented, it is expected that a negative exponential is formed. By taking the number of EPOCHs where the graph flattens out, it indicates that the training data is no longer giving any useful information and that the output should be redone with that output. If the output showcases a noisy chart like seen in 7.3.2, indicates a higher number of EPOCHs are required.

Appendix 7.3.2 also plots the predicted vs. observed values for the training, validation and testing data splits. This visually indicates the difference between the predicted and observed values. It is key to note that in testing, when smaller datasets are used, the predicted values will be further away from the observed. This is a product of the characteristics of the model, where it requires large volumes of data to become effectively trained. Next, the code terminal output creates a table showcasing the start and the end of the testing data. It is observed that the predicted values at the start of the test set are closer to the observed but by the end of the test, the difference between predicted and observed values widened. This could indicate that the data split percentages should be changed to make the testing set smaller. As this represents a singular test scenario, it is clearly observed that further testing is required. Additionally, that no singular perfect combination and implementation of this model due to its nature and varying inputs and variable combinations.

## 4.3 Modelling Limitations and Future Scope

The phase one models utilised elementary machine learning methods, obtaining reasonable accuracy. It is clear that this can certainly be optimized further. Due to time constraints, default parameters were used for the models. Accuracy could certainly be improved by refining the parameters, for example, tree size, number of trees and criterion. The other limitation of phase one comes from the limited testing on various data sizes, testing various independent input variables and modifying the test-train split. Despite reaching significant accuracy, the results may present a mischaracterisation of the true accuracy and it would likely be lower when tested on smaller sample sizes and implemented into a live environment. Given the analysis was performed on a static dataset, the feasibility of modifying these algorithms to a live environment has not been explicitly investigated. Ideally, Kensho would like to implement our algorithms into a live environment where data is continually updated and can be fed into the models directly.

Accordingly, care is required to understand that the accuracy of the results is not indicative of how the algorithms would perform in a live environment. Another significant limitation of the Phase one model is that the algorithm's categorical dependent variable, $P^D$, does not accommodate for the occurence no movement in the 'close' price as price direction in binary. Accordingly the Phase one model, omits the unchanged price periods from the dataset, which indicates the need for a more dynamic solution. Similarly, where the independent variables are computed to be zero, the program makes an alteration to the data to make this value non-zero, (0.0001), ensuring the algorithm will run. Accordingly, it was decided not to investigate these refinements further, as the Phase 2 model provided a more robust, fit to problem solution.

Despite the significant improvement from phase one, the phase two model was not without its limitations. The LTSM poses shortcomings which must be considered. Firstly, the model requires considerable levels of retesting to optimise the input parameters and therefore maximise accurate predictions. This results in a high time complexity for the model and limits the applicability to apply the model in a live data environment. Next, there are substantial barriers to maintain the model given the advanced scope of the model. Furthermore, the LSTM model requires substantial input data to generate accurate predictions; this would be an issue if Kensho was looking to test the model on a new futures market where limited input data is available. Accordingly, this informs the Phase two future scope.

The modifications and optimisations can be segmented into two categories. Firstly, due to the limited testing, it is likely that with more testing the parameters and ratio of training to testing size can be improved to produce better results. The second implementation would be to optimise the code such that it could be used in a live data environment and reduce the time complexity of the program. This would suit Kensho as they could test the program with active market data and verify its performance. Additionally, in future applications this model can be converted to either be multivariate or can be used to test on other predictors such as ‘Open’ or the differences between two predictors. The other direction that could be worth exploring would be to look at other forms of machine learning and artificial intelligence models which is currently an active area of research. (Dixon & London, 2021).

# 5 Conclusion

This report sought to develop a price prediction model utilising all preceding information at time, $t-1$, to predict price at time, $t$. The analysis employed a sequential approach to solve this problem, increasing in complexity and flexibility. The Phase 1 Models addressed price prediction as a binary classification problem predicting the direction of the price at time $t$ using supervised machine learning classifier algorithms. Despite apparent predictive success of the models, the construction of the price direction categorical dependent variable ultimately limited the effectiveness of the algorithms where price remained constant between periods. Accordingly, this masks the true accuracy of each algorithm, and hence to use supervised machine learning mandates further testing to refine the construction, input parameters and optimal data input to be built into a live environment.

Next, the Phase 2 Model utilised unsupervised deep learning and constructed a LSTM RNN. Accordingly, this provided the ability to predict the value of the index future contract at time $t$, which is a significant advantage when compared to supervised machine learning binary classification. With a combination of additional input optimisation and additional testing to reduce the time complexity makes it a suitable candidate to be progressed into a live environment.

# 6 References

MachineLearningMastery (2016) Linear discriminant analysis for machine learning. Available at: https://machinelearningmastery.com/linear-discriminant-analysis-for-machine-learning/ (Accessed: October 19, 2022).

CME Group (2022) Equity index products - CME group. Equity index products - CME group. Available at: https://www.cmegroup.com/markets/equities.html (Accessed: October 28, 2022).

Dixon, M. and London, J. (2021) Financial forecasting with -RNNS: A time series modeling approach, Frontiers. Frontiers. Available at: https://www.frontiersin.org/articles/10.3389/fams.2020.551138-/full (Accessed: October 28, 2022).

FinanceTrain (2022) Classifier model in machine learning using Python. Classifier model in machine learning using Python. Available at: https://financetrain.com/classifier-model-in-machine-learning-using-python (Accessed: October 22, 2022).

IBM (2022) Python vs. R: What's the difference? Python vs. R: What's the difference? Available at: https://www.ibm.com/cloud/blog/python-vs-r (Accessed: October 17, 2022).

IBM (2022a) What is deep learning? What is deep learning? Available at: https://www.ibm.com/cloud/learn/deep-learning (Accessed: October 20, 2022).

JavaPoint (2022) Support Vector Machine (SVM) algorithm . Support Vector Machine (SVM) algorithm . Available at: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm (Accessed: October 14, 2022).

Joseph, V.R. (2022) Optimal ratio for data splitting, Wiley Online Library. Wiley Online Library. Available at: https://onlinelibrary.wiley.com/doi/10.1002/anie.201905241 (Accessed: October 28, 2022).

KenshoData (2022) Home: Kensho data. Home: Kensho data. Available at: https://www.kensho-data.com.au/ (Accessed: October 28, 2022).

PluralSight (2022) Introduction to LSTM Units in RNN. Introduction to LSTM Units in RNN. Available at: https://www.pluralsight.com/guides/introduction-to-lstm-units-in-rnn (Accessed: October 10, 2022).

TowardsDataScience (2022) Back to basics: Assumptions of common machine learning models . Available at: https://towardsdatascience.com/back-to-basics-assumptions-of-common-machine-learning-models-e43c02325535 (Accessed: October 28, 2022).

# 7 Appendix

## 7.1 Data

Below is an example of the index futures data received from Kensho. Each Index was its own datafile and each datafile had the same variables can be observed below.

| Close | High | Interest | Low | Open | Time | Volume |
|---|---|---|---|---|---|---|
| 6668 | 6668 | 358578 | 6665 | 6666 | 2020-01-03T18:59:00 | 35 |
| 6668 | 6669 | 358578 | 6668 | 6669 | 2020-01-03T19:00:00 | 11 |
| 6667 | 6667 | 358578 | 6667 | 6667 | 2020-01-03T19:01:00 | 14 |
| 6665 | 6666 | 358578 | 6664 | 6666 | 2020-01-03T19:02:00 | 14 |
| 6666 | 6666 | 358578 | 6666 | 6666 | 2020-01-03T19:03:00 | 3 |

## 7.2 Phase One

### 7.2.1 Model Reference

**Phase One:**

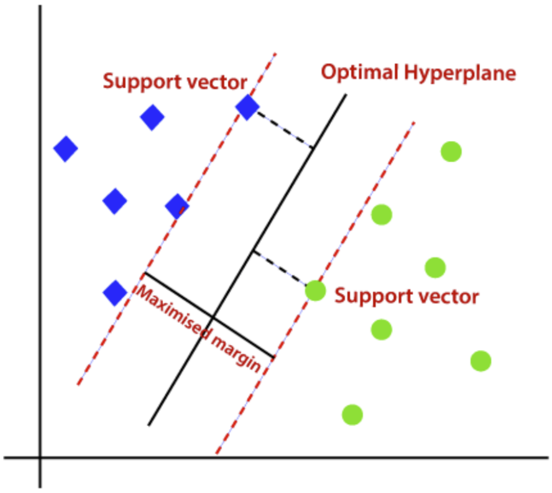


Figure 1: Support Vector Machine Algorithm Diagram
Source: *JavaPoint, 2022*

**Phase Two:**

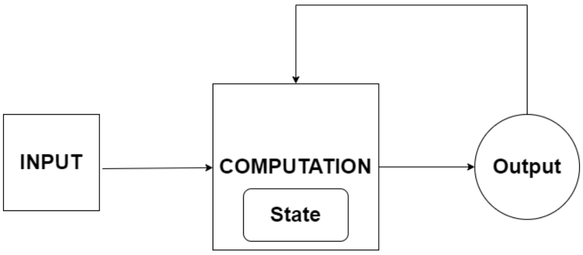


Figure 2: Illustration of the state variable in the model

### 7.2.2 Model Input Specifications

**Input File:**

ASX SPI 200 Index, FTSE 100 Index and E-mini S&P 500 (Dollar)

**Analysis Start Date:**

*2016-12-29T09:35:00.000000Z*

**Analysis End Date:**

*2017-12-19T07:45:00.000000Z*

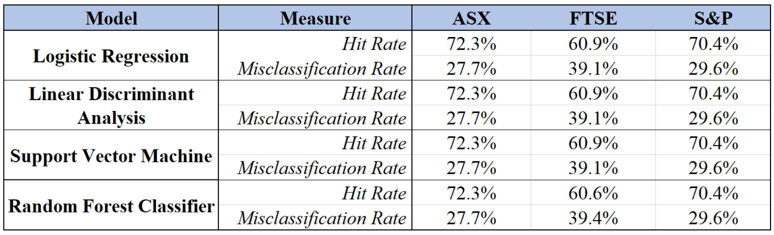
**Percentage of Data Used:**

*24%*

**Training vs. Testing Split Selected:**

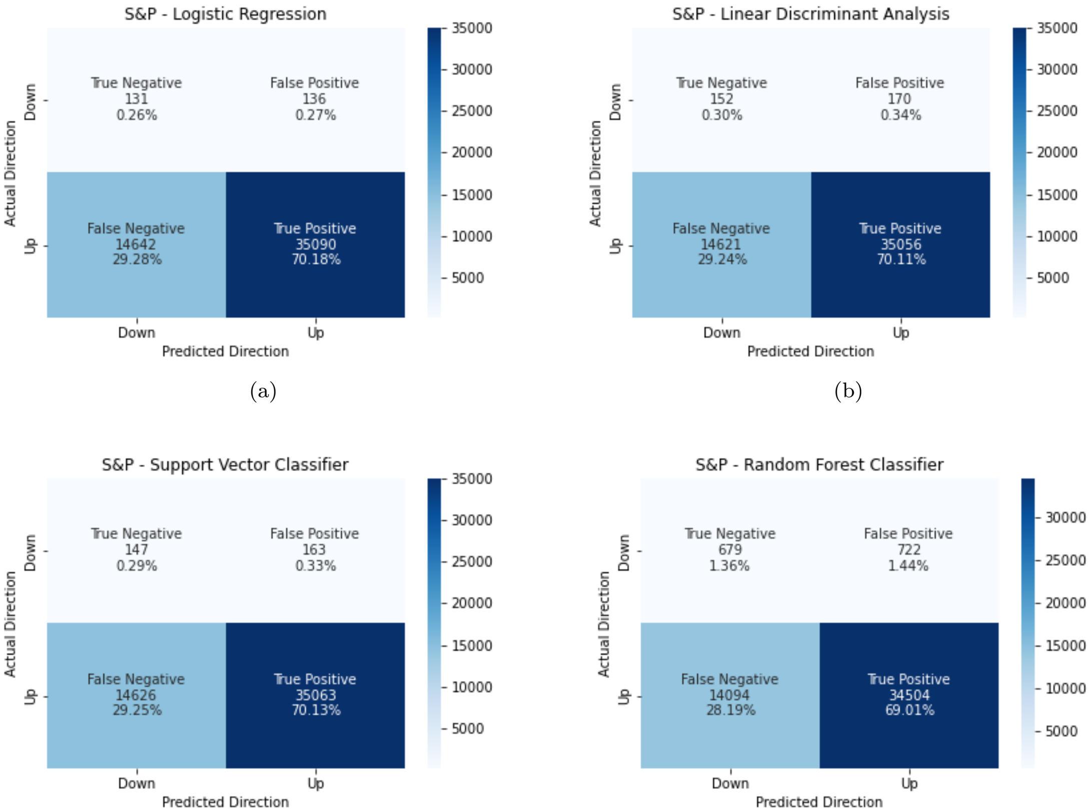
*80% vs. 20%*

### 7.2.3 Model Output



| Model | Measure | ASX | FTSE | S&P |
|---|---|---|---|---|
| **Logistic Regression** | *Hit Rate* | 72.3% | 60.9% | 70.4% |
| | *Misclassification Rate* | 27.7% | 39.1% | 29.6% |
| **Linear Discriminant Analysis** | *Hit Rate* | 72.3% | 60.9% | 70.4% |
| | *Misclassification Rate* | 27.7% | 39.1% | 29.6% |
| **Support Vector Machine** | *Hit Rate* | 72.3% | 60.9% | 70.4% |
| | *Misclassification Rate* | 27.7% | 39.1% | 29.6% |
| **Random Forest Classifier** | *Hit Rate* | 72.3% | 60.6% | 70.4% |
| | *Misclassification Rate* | 27.7% | 39.4% | 29.6% |

Figure 3: Phase 1 Model Tabulated Results



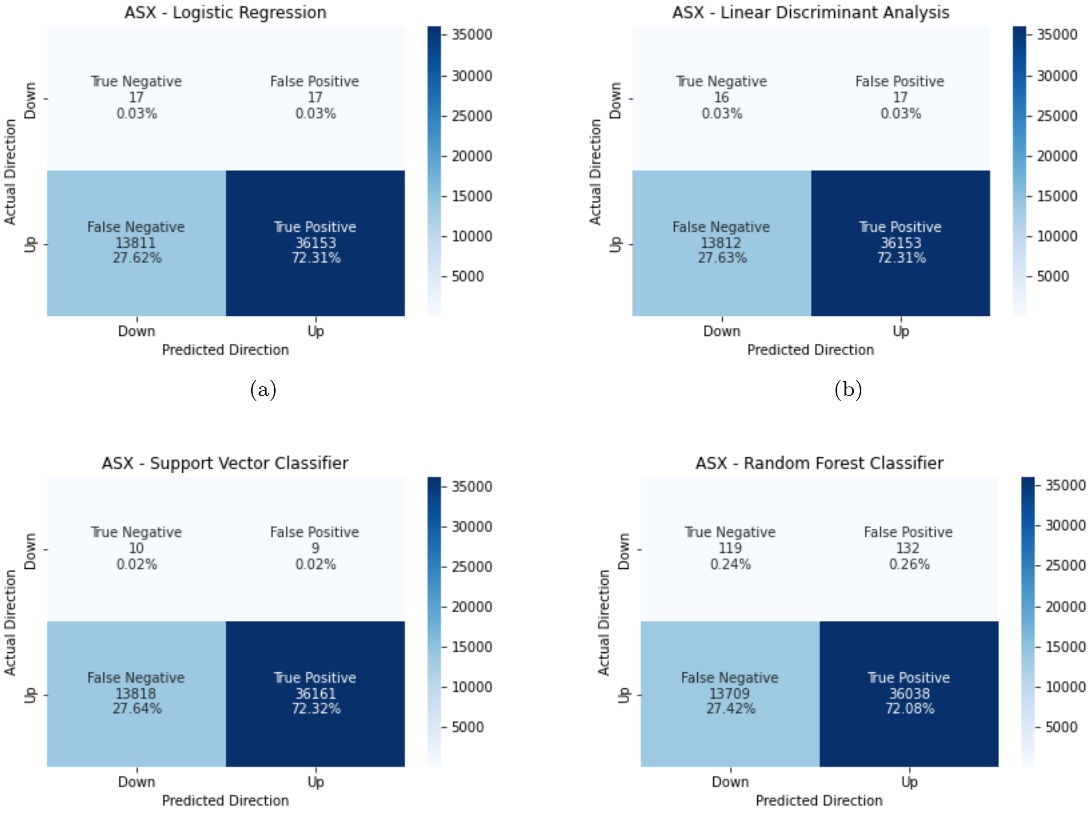
(a)

(b)

(c)

(d)

Figure 4: Phase 1 Models - S&P 500 Index Futures Market



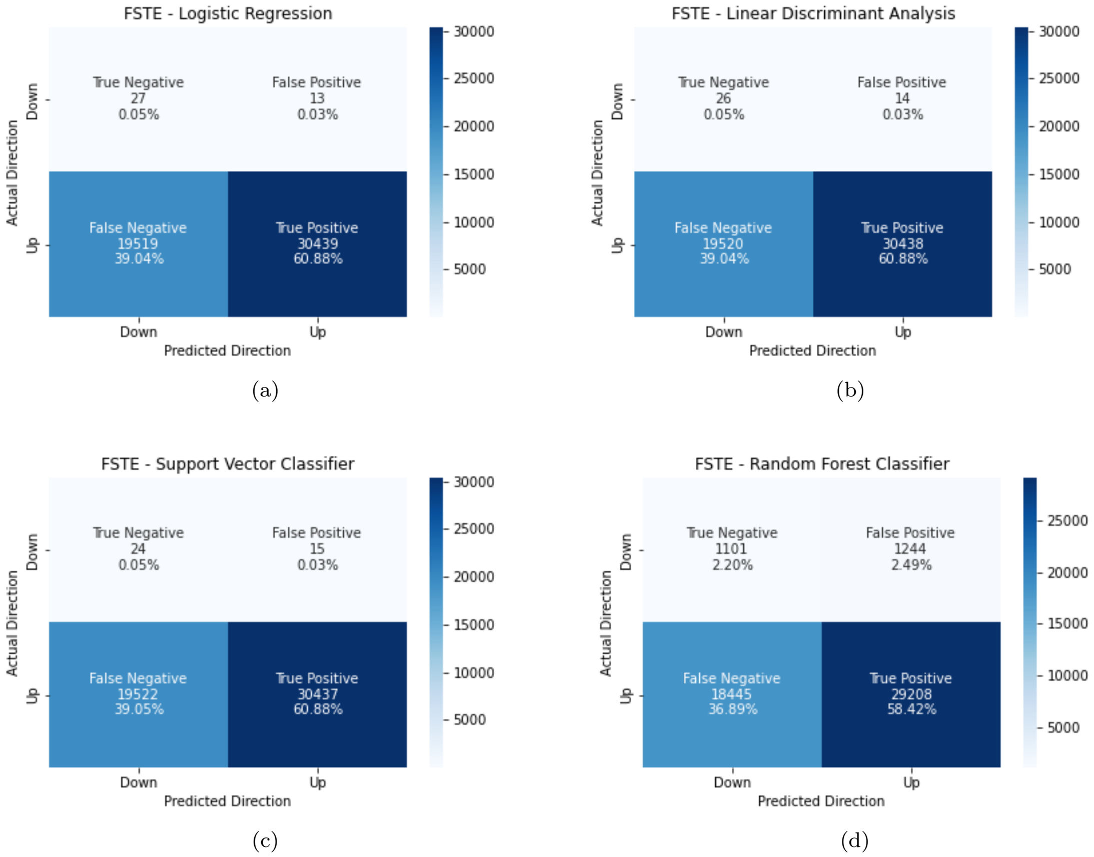
(a)

(b)

(c)

(d)

Figure 5: Phase 1 Models - ASX 200 Index Futures Market



(a)

(b)

(c)

(d)

Figure 6: Phase 1 Models - FTSE 100 Index Futures Market

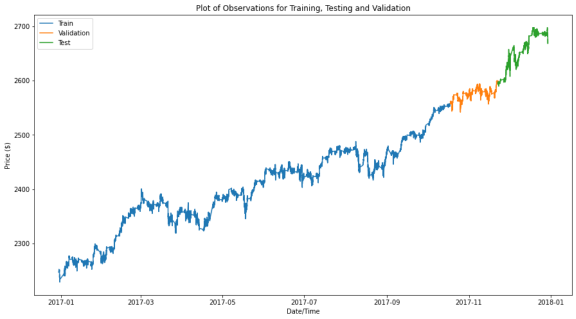
## 7.3 Phase 2

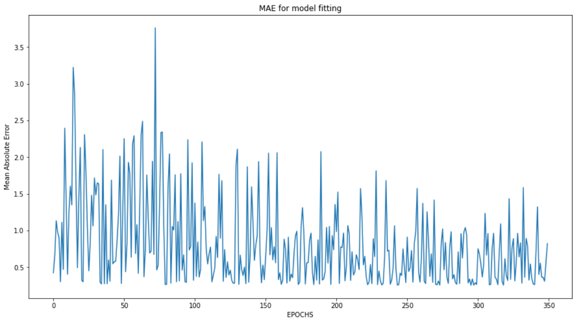
### 7.3.1 Model Input Specification

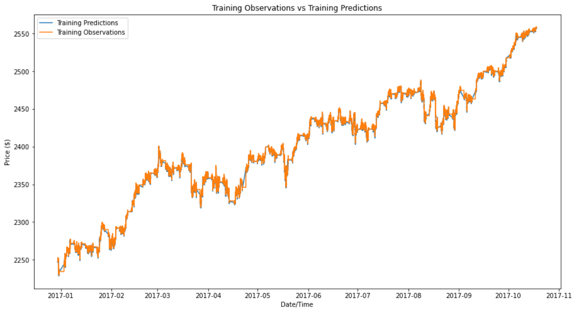
For the sample output provided in this section, the following parameters were used for testing:
**Data:** *E-mini SP 500 (Dollar) - cont (USD)*
**Date Range:** *#0 (2016-12-30T0:2:34) to 343589 (2018-01-01T23:00)*
**EPOCHS:** *350*
**Lags:** *3*

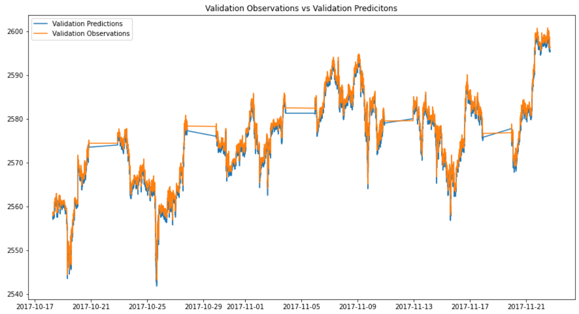
### 7.3.2 Model Output

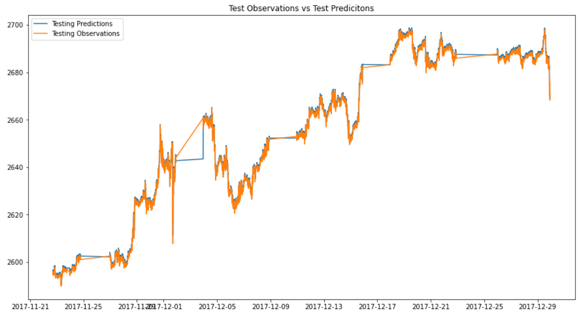
Below are the outputs for the Phase 2 model using the parameters as above.

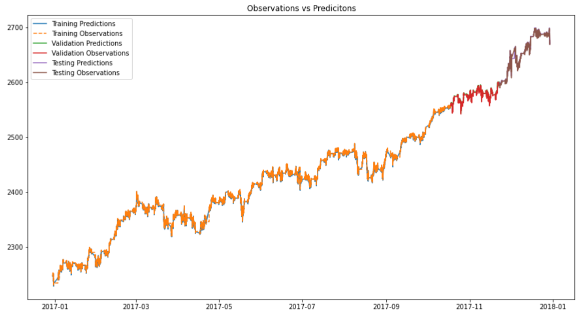


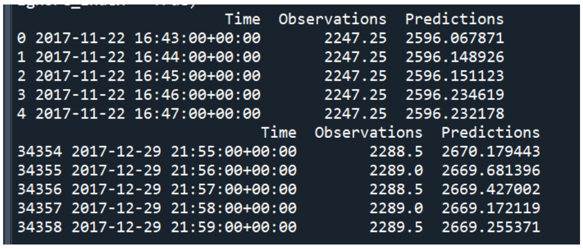












```
                       Time  Observations  Predictions
0 2017-11-22 16:43:00+00:00       2247.25  2596.067871
1 2017-11-22 16:44:00+00:00       2247.25  2596.148926
2 2017-11-22 16:45:00+00:00       2247.25  2596.151123
3 2017-11-22 16:46:00+00:00       2247.25  2596.234619
4 2017-11-22 16:47:00+00:00       2247.25  2596.232178
                           Time  Observations  Predictions
34354 2017-12-29 21:55:00+00:00        2288.5  2670.179443
34355 2017-12-29 21:56:00+00:00        2289.0  2669.681396
34356 2017-12-29 21:57:00+00:00        2288.5  2669.427002
34357 2017-12-29 21:58:00+00:00        2289.0  2669.172119
34358 2017-12-29 21:59:00+00:00        2289.5  2669.255371
```

## 7.4 Individual Appendicies

### 7.4.1 Rajiv Metha: LSTM RNN Model

**Methodology** As discussed in phase 2 found in section 3.2 of this report, after trialling the previous models discussed in phase 1 found in section 3.1 plus other models that were trialled such as a random forest regressor model which was not included in the report due to them not being plausible for the use of time series data, it was decided that the complexity of the model should be increased to achieve accurate results.

This led to the decision to make a neural network for price prediction. After extensive research on different forms of neural networks, such as the forward feed neural networks and a standard recurrent neural network, it was found that the Long Short Term Neural recurrent network model (LSTM) was the optimal choice. This is because the LSTM solves the 'Vanishing Gradient Problem' which discusses how in a RNN too much data collection in the model can cause the gradient to disappear which causes the AI to no longer learn or detect any useful change and therefore, will make either poor predictions or overfit the model. The LSTM model solves this problem by introducing a state variable. This state variable is known as the LSTM cell as seen in section 7.2.1, which perform sigmoid calculations to perform the following three gate actions:

- Input gate - decides if the cell should be updated based on the new data input.
- Forget gate – decides if the memory cell should be set to 0 and therefore, does not retain any of the new output information.
- Output gate – decides whether the information from the input and forget gate should be shown in the next hidden state where the hidden states make predictions on the next set of inputs.

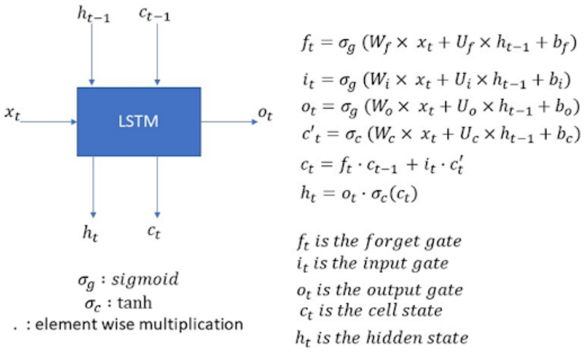

$h_{t-1}$
$c_{t-1}$
$x_t$
LSTM
$o_t$
$h_t$
$c_t$
$\sigma_g$ : *sigmoid*
$\sigma_c$ : tanh
. : element wise multiplication
$f_t = \sigma_g\ (W_f \times\ x_t + U_f \times h_{t-1} + b_f)$
$i_t = \sigma_g\ (W_i \times\ x_t + U_i \times h_{t-1} + b_i)$
$o_t = \sigma_g\ (W_o \times\ x_t + U_o \times h_{t-1} + b_o)$
$c'_t = \sigma_c\ (W_c \times\ x_t + U_c \times h_{t-1} + b_c)$
$c_t = f_t \cdot c_{t-1} + i_t \cdot c'_t$
$h_t = o_t \cdot \sigma_c(c_t)$
$f_t$ *is the forget gate*
$i_t$ *is the input gate*
$o_t$ *is the output gate*
$c_t$ *is the cell state*
$h_t$ *is the hidden state*

Figure 7: LTSM Cell Diagram
Source: *PluralSight, 2022*

**Results** Due to the nature of this problem and algorithm, there is no best results as factors such as input size, density layers, number of lags for prediction, EPOCHS (number of times the training data is fitted to the model) and training/validation/testing percentages all need to be changed and tested as they are all dependent on each other and can vary the outputs. Therefore, for this showcase, we will be using the data from the year 2018. Figure 8 showcases how the data is split between its training, validation and testing percentages 80% - 10% 10%.

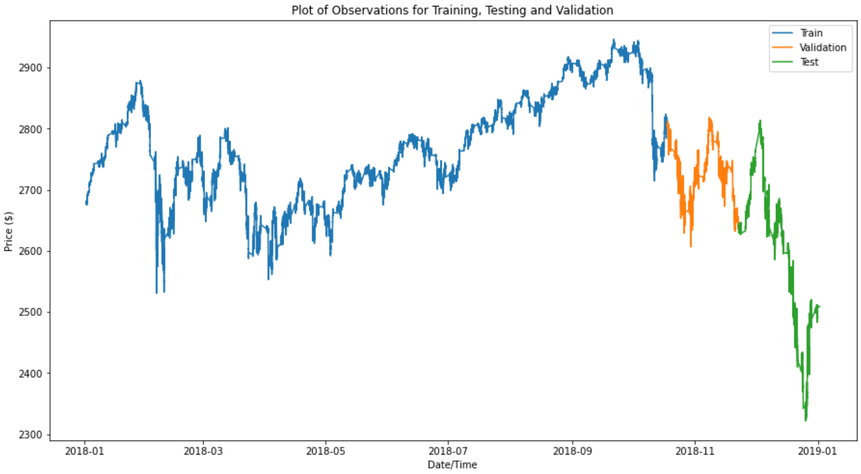


Figure 8: Data Split

The next Figure 9 showcases how the model is fitting after the repeated training of the training data determined by the number of EPOCHS. By having a value of EPOCHS being too large it can cause overfitting of the model or by choosing a small value can cause underfitting. For this test 350 EPOCHS were chosen.

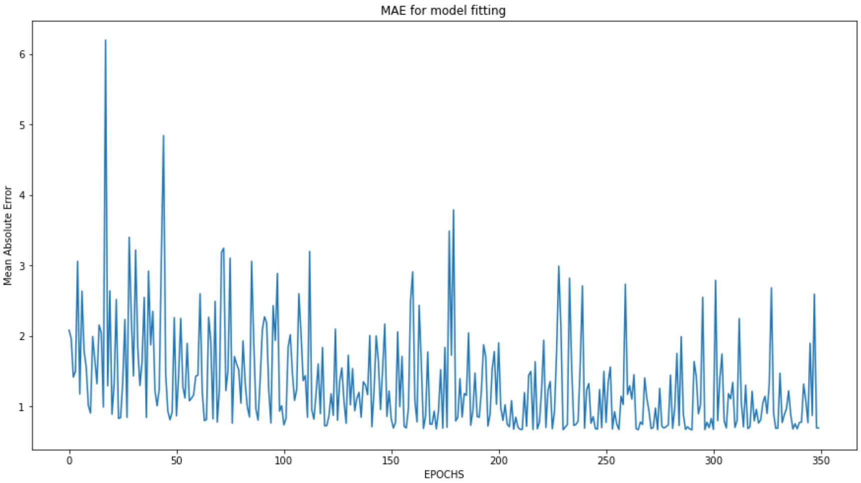
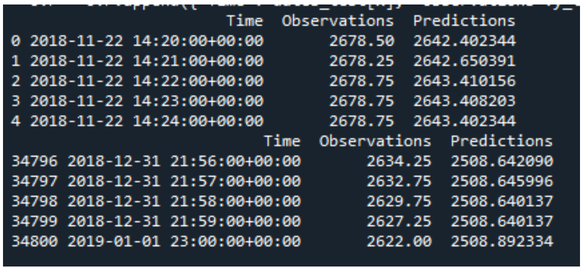


Figure 9: MAE for Training Fit

As seen in Figure 9, a bell curve is starting to form, though it appears that the model needs more EPOCHS in order to make more accurate predictions as it spikes between a Mean Absolute Error of approximately 0.7 and 3 towards the 350th EPOCH.



```
                          Time  Observations  Predictions
0 2018-11-22 14:20:00+00:00       2678.50  2642.402344
1 2018-11-22 14:21:00+00:00       2678.25  2642.650391
2 2018-11-22 14:22:00+00:00       2678.75  2643.410156
3 2018-11-22 14:23:00+00:00       2678.75  2643.408203
4 2018-11-22 14:24:00+00:00       2678.75  2643.402344
                              Time  Observations  Predictions
34796 2018-12-31 21:56:00+00:00       2634.25  2508.642090
34797 2018-12-31 21:57:00+00:00       2632.75  2508.645996
34798 2018-12-31 21:58:00+00:00       2629.75  2508.640137
34799 2018-12-31 21:59:00+00:00       2627.25  2508.640137
34800 2019-01-01 23:00:00+00:00       2622.00  2508.892334
```

Figure 10: Numerical Output

The algorithm then outputs both graphs and a table displaying the observations vs prediction. It can be seen that predictions that the model makes are not too far off from the observations made at the time frames at the beginning of testing but towards the end of the test set, it starts to be roughly 00 off. This could indicate that the number of test predictions should be lower plus further testing with the other variables is required.

**Conclusion** As this mode is unsupervised and also is testing predictions vs observations rather than the amount of times the value goes up or down in price, further testing is needed to get more precise results. On a theoretical level, this model perfectly fits the type of data it is trying to predict outside of a few limitations that it has such as the large amounts of testing required and the variability of the parameters.

### 7.4.2 Lennox Hemingway: Phase One - Supervised Machine Learning

**Methodology** To apply a supervised machine learning algorithm to a given set of data, it is firstly important that the desired outcome is clearly understood. In the context of this project the aim was to create a method of price prediction for a financial market. Explicitly, this was defined as, to utilise the information up to time,$t-1$, to predict a price at time, $t$. This can be solved by predicting the direction of the price at time, $t$, as the magnitude of the 1 period close price change is limited. Accordingly, this is computed as:

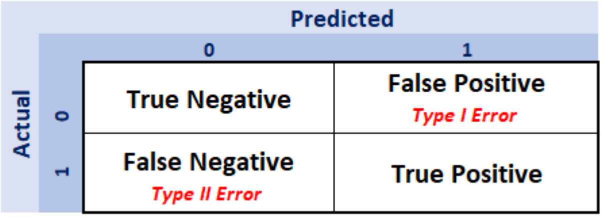
$$P_t^d = \begin{cases} +1 & \text{for} \quad P_t > P_{t-1} \\ -1 & \text{for} \quad P_t < P_{t-1} \end{cases}$$

That is, price direction, $P_t^d$, is up where the close price, $P$, at time $t$ is greater than the close price at time, $t-1$, and down when closer price at time, $t$, is less than, $t-1$. Hence, the dependent variable is categorical. Thus, the problem becomes a binary classification problem, well suited for supervised machine learning classification algorithms, whereby the data is classified into two specific categories, a positive or negative in price direction, given a set of independent variables.

Through utilising the shifted lag series of prior trading period close price, 1 and 2 period lagged percentage returns are computed, forming the first two categorical independent input variables. These two variables help capture the immediate momentum of the previous periods, and are a likely predictor of future price direction. Similarly, the percentage change in traded volume provides an indication of the demand size and change within the market and may provide an indication of the upward or downward pressure on future price direction. Through utilising percentage changes, the input variables are hence normalised in scale. Hence, the model formed as three continuous random independent variable inputs to predict a categorical dependent variable. Whilst there are numerous, multi-variate and uni-variate independent variables which may have influence on Price Direction, research and preliminary testing demonstrated that these to have considerable predictive potential.

**Model** As the general operation of each of the four supervised machine learning models tested is set out in the report above, herein the assumptions and input setting will be explored. For the Linear Discriminant Analysis, all default settings provided in the Python package: $'sklearn.discriminant-\_analysis'$ were used. The assumptions for ML Linear Discriminant Analysis require the following: a categorical dependent variable, no outliers and the same approximately variance for each independent variable (MachineLearningMastery, 2016). This has been met by design and through the normalization of the independent variables. Similarly, for Logistic Regression, all default parameter settings from $'sklearn.linearmodel'$ were used. For the Support Vector Machine model, the input settings suggested by (FinanceTrain,2022). Noting that there are no assumptions required to be satisfied (TowardsDataScience, 2022). Similarly, the Random Forest Classifier has no assumptions which are required to be met and input settings were default or informed through research (TowardsDataScience, 2022; FinanceTrain,2022). It is important to note in the context of the project's aim that for both the Support Vector Machines and Random Forest Classifier the randomisation must be considered as only information from time, $t-1$, to predict time $t$, therefore the relevant input parameter was modified to adhere to this using '$random\_state'$ parameter.

**Results** To measure the success of each of the four models applied, a binary classification contingency table, a confusion matrix, can be used. The confusion matrix categorizes the predictions produced from each algorithm. It provides a measure for the quantity of correct predictions that the classifier produced, both upward and downward price predictions. $FP$, false positive, represents the incorrectly classified up period, where the algorithm predicts as downward price direction, alternatively known as a Type I Error. $TP$, true positive, where the algorithm predicts up and the actual price direction goes up. Similarly, $FN$, false negative, incorrectly classified up periods, , alternatively known as a Type II error. Lastly, $TN$, true negative where the algorithm classifies it as a down movement and price direction goes down.



| | | Predicted | |
|---|---|---|---|
| | | 0 | 1 |
| Actual | 0 | True Negative | False Positive<br>*Type I Error* |
| | 1 | False Negative<br>*Type II Error* | True Positive |

Figure 11: Confusion Matrix

From the confusion matrix, we are able to assess the model's performance generally by computing the classification accuracy or Hit Rate, this is computed as a percentage of the total number of correct predictions. This is expressed as:

$$H^r = \frac{1}{n}\sum_{t=1}^{n} I_{P_t^d=\widehat{P_t^d}} \qquad \text{where: } P_t^d \text{ is the prediction for minute, } t\text{, and:}$$

$$I_{P_t^d=\widehat{P_t^d}} = \begin{cases} 1 & \text{for} \quad P_t^d = \widehat{P_t^d}, t \in 1, ..., n \\ 0 & \text{for} \quad P_t^d \neq \widehat{P_t^d}, t \in 1, ..., n \end{cases}$$

**Similarly,** we can also compute this from the Confusion Matrix:

$$\frac{TP + TN}{TP + FP + FN + TN}$$

**Misclassification Rate** or error rate of each algorithm, calculated as the count of incorrect predictions on the total predictions made:

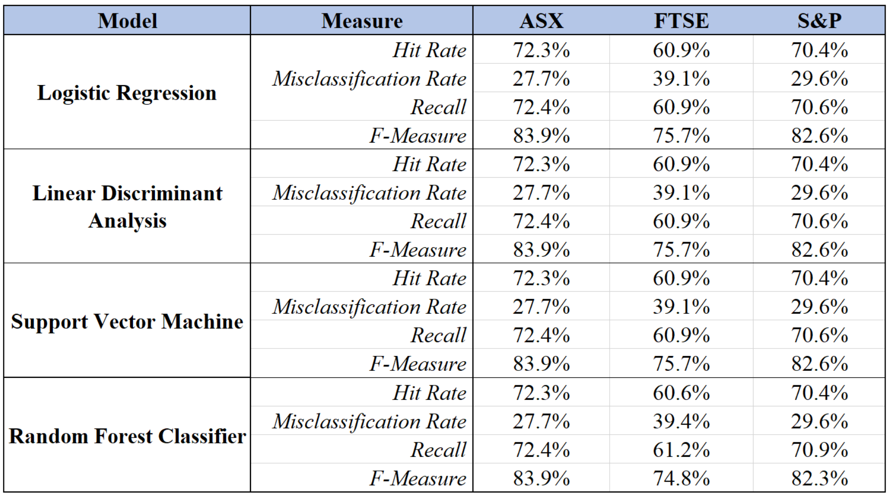
$$\frac{FP + FN}{TP + FP + FN + TN}$$

**Recall:** calculated as total of correct predictions of out the total positive predictions, aiming for the highest possible value:

$$\frac{TP}{TP + TN}$$

**F-Measure:** This measure facilitates comparison between models by, computed as:

$$\frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$



| Model | Measure | ASX | FTSE | S&P |
|---|---|---|---|---|
| **Logistic Regression** | *Hit Rate* | 72.3% | 60.9% | 70.4% |
| | *Misclassification Rate* | 27.7% | 39.1% | 29.6% |
| | *Recall* | 72.4% | 60.9% | 70.6% |
| | *F-Measure* | 83.9% | 75.7% | 82.6% |
| **Linear Discriminant Analysis** | *Hit Rate* | 72.3% | 60.9% | 70.4% |
| | *Misclassification Rate* | 27.7% | 39.1% | 29.6% |
| | *Recall* | 72.4% | 60.9% | 70.6% |
| | *F-Measure* | 83.9% | 75.7% | 82.6% |
| **Support Vector Machine** | *Hit Rate* | 72.3% | 60.9% | 70.4% |
| | *Misclassification Rate* | 27.7% | 39.1% | 29.6% |
| | *Recall* | 72.4% | 60.9% | 70.6% |
| | *F-Measure* | 83.9% | 75.7% | 82.6% |
| **Random Forest Classifier** | *Hit Rate* | 72.3% | 60.6% | 70.4% |
| | *Misclassification Rate* | 27.7% | 39.4% | 29.6% |
| | *Recall* | 72.4% | 61.2% | 70.9% |
| | *F-Measure* | 83.9% | 74.8% | 82.3% |

Figure 12: Phase 1 Model Tabulated Results

**Discussion** All four of the algorithms are approximately equivalent in performance across each of the computed metrics for each index when predicting price direction. However, it is clear that further investigation into the input specifications for each model to optimise performance. Next, for these supervised machine learning algorithms to be successful in ingesting new data rapidly and creating classifications, the time complexity of each algorithm must be addressed. Thus, the Support Vector Machine, Random Forest Classifier are not an appropriate solution, however care must be taken to satisfy the assumptions of the Logisitic Regression and Linear Discriminant Analysis. A factor in slowing down each algorithm, or increasing the time complexity, is the quantity of data relied upon to make the next prediction at time, $t$. To reduce the amount of data ingested by each model, will improve the speed of each model, thus investigation into the optimal quantity of data required to achieve a desired level of accuracy before it decays is required. Lastly, for solving this problem using price direction, at this periodicity and low trade activity may cause the data may remain stationary for multiple time periods, where this occurs the price direction is not binary, and therefore must be assumed to be positive or be omitted. This assumption is limiting and reduces the viability of the application of this model to data where close price may remain unchanged between periods.

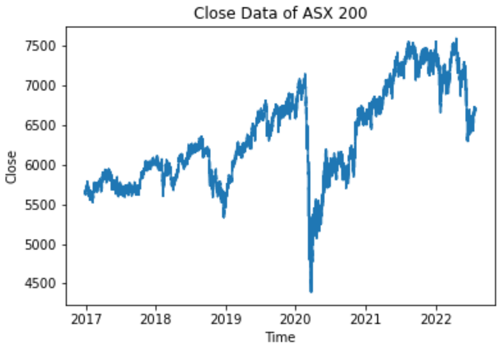
### 7.4.3 William Reynolds: Introductory Analysis

#### Methodology

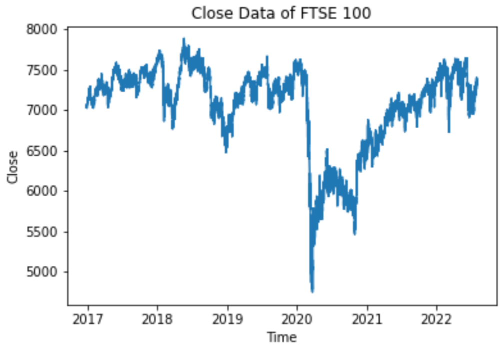
1. Plotting the data was the one of the first tools used in the project. Although large, plotting each index would allow instantaneous recognition of any obvious or insightful patterns into the data over a large timescale. This idea could then be expanded upon and therefore plots of smaller subsets of the time-series data analysed. The basic theory to plot the data is to input each of the index data files and then, set the time as the index of the data. This makes plotting straightforward as plotting the closing price variable will automatically be plotted against the index, time.

2. The first model utilised after the initial visual analysis was the Random Forest Classifier model. The idea of this model was to instead of looking to quantifiably predict future prices, start simple and instead look to predict the movement itself. The first instance of the model would be single directional. That is, the model would take on values of 1 and 0, yet the data had the option of moving upwards, downwards, and unchanged. The idea was to look at the difference between close and opening prices. If the closing price was greater than the opening, this would indicate an increase in the time duration and would be indicated with a value 1, an unchanged result or decrease would then be represented by a 0. As the model is essentially predicting whether the close price would be higher than the opening. The factors that could then be chosen to be used to make these predictions include: the volume, opening, close, interest and lagged movements from previous time periods and the opening of the current time period.

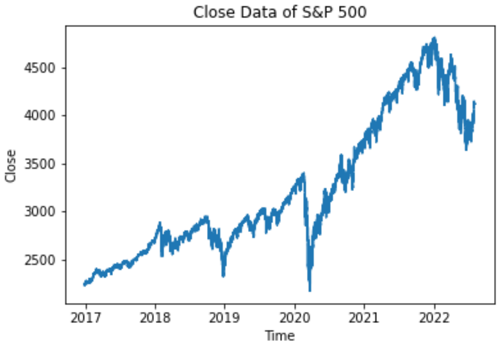
**Results** The results of the basic analysis are provided below:



(a)



(b)



(c)

The table illustrates the results from the initial Random Forest Classifier Model using various samples of the data Key:

| | **S&P 500** | **FTSE 100** | **ASX 200** |
|---|---|---|---|
| Correct Increase | 40600 | 3789 | 13756 |
| Missed Increase | 11429 | 34846 | 21680 |
| Correct No Increase | 15047 | 44010 | 36635 |
| Missed No Increase | 49780 | 4475 | 22969 |
| **Meaningful Accuracy** | 0.45 | 0.46 | 0.37 |

**Correct Increase** = Prediction 1, Actual 1
**Missed Increase** = Prediction 0, Actual 1
**Correct No Increase** = Prediction 0, Actual 0
**Missed No Increase** = Prediction 1, Actual 0

Meaningful Accuracy was defined as the ratio of correct increase predictions divided by the number of correct increase predictions plus the number of missed no increase. This is because correct increase prediction can create a profitable scenario, while a missed no increase would create a loss scenario

### Discussion

1. For the visualisation of the data, there is no identifiable surprise which was the assumption considering this would have indicated easily profitable opportunities. However, visualising the data given is a good idea to understand the kinds of analysis that can be performed, for example the data clearly is not stationary. Therefore, looking at differenced data would be the logical choice moving forward for analysis. It also serves as verifying there were no simple solutions to the problem given, as this would both save everyone time and achieve the clients goals.

2. The results from the Random Forest classification model are somewhat misleading but understandably need further development. The reason the results are misleading is although there are a significant number of missed increases, these are simply missed signals and do not indicate a loss in a trading scenario, as the trader would simply wait until the model predicts the next increase to sell.

   Despite this, the accuracy still leaves a lot left to be desired, particularly due to the large number of incorrect no-increases, as for a trader this would be indicate a loss-making decision, they would receive the prediction of a movement and therefore sell yet the actuality is there was no change. Despite its shortcomings this basic random forest classifier model formed the starting point for the analysis models. The Phase One models look to further develop the idea of predicting movements opposed to quantifying values, yet improve upon some of the shortcomings of this algorithm.