IOD Capstone Project Final Report

Predicting Future Stock Prices Using Machine Learning

Chang-Joon Lee

1. Introduction

1.1 Problem Statement

In this project, we want to **predict future stock prices using trends from previous years**.

Investing in stocks, or share trading, is one of the primary and arguably one of the most popular forms of investment for many investors. According to Statista Research Department[1], the total size of equity market capitalisation (the total value of all shares traded on the equity market) of the New York Stock Exchange, which is the largest stock exchange in the world, is just over 24.1 trillion U.S. dollars as of August 2022. This number goes to show the sheer amount of money and the number of investors that are invested in the stock market.

Share trading is a popular option for many because it can yield higher investment return for relatively less capital in comparison to other investments, for example, land property. However, investing in stocks also comes with higher uncertainties, and therefore relatively higher risk for investors than other forms of investment.

The issue is that price movements of stocks exhibit noises, fluctuations, and volatility, due to several independent and dependent variables that influence the values of stocks in the market. Because of this complexity, it is very challenging to correctly predict their future values with great precision. And the issue is compounded especially in our current, post-pandemic economic climate, where the recession is looming, and the falling confidence of many investors could potentially lead to a major collapse in the stock market.

Given the current state, the ability to precisely forecast future stock prices becomes more vital for the investors, at the very least, to minimise their potential loss, and at best, identify the most opportune time to maximise their profits by buying low and selling high.

1.2 Business Question by Our Stakeholders

Our stakeholders are **retail investors** (i.e., non-professional traders) who generally trade in smaller volumes and with less capital than other stockbrokers and professional traders.

The problem stated in the previous section is more profound for our stakeholders, because they typically lack the required background knowledge and experience, and time to fully digest the available financial and economic data and confidently let the data dictate their investment decisions. So, the ideal solution for our stakeholders will be the ability to **easily and accurately forecast future stock prices** without the need to directly extract insights from complex data.

Therefore, the business question set out by our stakeholders is:

"How much **more investment return** can we provide to our stakeholders by providing **easy** and **accurate** future stock price forecast?"

In terms of accuracy, the aim is to be able to have an average error of less than 1%. For example, if the stock is forecasted to be priced at $100 and the actual price is $105 (error of 5%), the loss in potential profit gain is $5. If one has invested $10,000 in the stock, then this loss scales to $500. Thus, the error in our forecast becomes more profound as the total amount of investment increases. This may not seem critical, but since our stakeholders typically have limited capital, we would like to minimise their potential loss as much as possible.

1.3 Modelling Objective

The business question stated above can be translated into the following modelling objectives:

> "Develop an **app** that can **automate** the process of:
>
> 1. **Gathering** financial and economic **data**
> 2. **Forecast** future stock prices based on the data"

More specifically, we aim to make mid-term (30-days) forecasts and a series of mid-term forecasts to show how the stock prices changes long-term (2-years).

To achieve our objectives, we will need to acquire the required data from several sources. The types of data required and where to acquire them are detailed in the next section.

2. Data Analysis

2.1 Data Acquisition and Pre-Processing

The required data are acquired using several Application Programming Interfaces (APIs) as shown on the next page (Table 1). The use of APIs automates the process of acquiring the most up-to-date data and therefore eliminates the need for our stakeholders to manually search for the data.

The most up-to-date and complete historical data were acquired. Because the API providers are active and are well-known financial institutions, the quality of data is very accurate (cross-checked with multiple sources) and are updated regularly.

Some data are only given monthly, quarterly or annually. Transforming these data into a daily format result in a significant number of null (empty) values. We used a combination of forward fill and backward fill methods (available in the Python Pandas library) to replace the null values. In using this method, we are assuming that a value on a particular day carries on to the next day and onwards until there is a new value recorded. For example, we assume a quarterly value for GDP recorded on 1 Jan 1990 remains unchanged and carries on until a new value is recorded on 1 April 1990 and so on. Using backward fill method will fill null values that are before the first recorded value. Using this method assumes that the values before the first recorded value did not change.

The time period for the data after acquiring and cleaning is from 1 Jan 1990 to 31 Dec 2019 (30 years).

The process of acquiring and cleaning the data are fully automated using user-defined custom helper functions in the source code.

The final dataset after cleaning consists of ~10,000 daily records with 26 features.

Table 1. A list of data acquired and their respective source(s).

Data Type	Data	Source	Notes
Stock Market Indices	S&P 500	Yahoo Finance API[2]	Target Variable to Forecast. Includes daily open, high, low, and close values, and volume of stocks traded.
	Dow Jones		Only used daily close values.
	Nikkei (Tokyo)		
	FTSE 100 (London)		
	Hang Seng (Hong Kong)		
	U.S. Treasury Bond		
Commodity	Gold	NASDAQ[3] & Alpha Vantage[4] APIs	
	Silver		
	Crude Oil		
Economic Indicator	Gross Domestic Product	NASDAQ[3] & Alpha Vantage[4] APIs	For USA. Given quarterly.
	Unemployment		For USA. Given monthly.
	Consumer Price Index		For USA. Indicates inflation. Given monthly.
	Median Income		For USA. Given annually.
Technical Indicator	Volatility Index	Yahoo Finance API[2]	
	Moving Average	N/A	Manually generated based on the S&P 500 data.
	Bollinger Bands		
	Relative Strength Index		
Investor Sentiment	University of Michigan Consumer Sentiment	Federal Reserve Economic Data API[5]	
	New York Times News Sentiment	New York Times API[6]	Based on custom-built sentiment analysis.

2.2 Exploratory Data Analysis (EDA)

Here, we will discuss some of the highlights of the EDA.

2.2.1 S&P 500

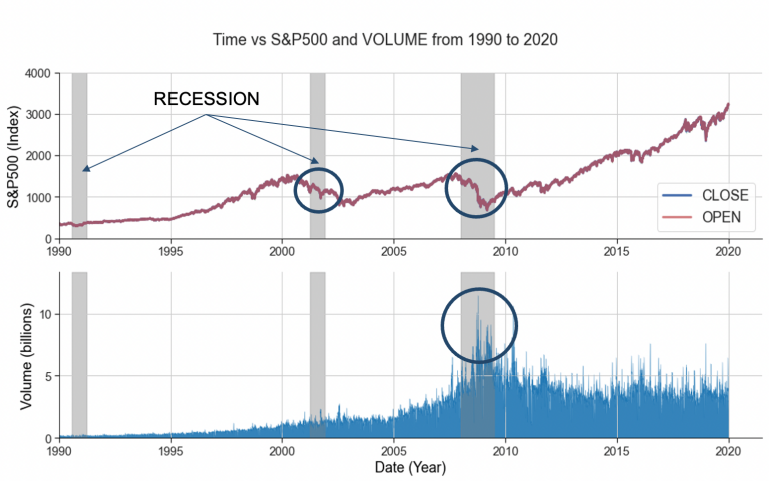

Figure 1. Change in S&P 500 Open and Close Values Over Time (top) and Change in the Volume of Stocks Traded Over Time (bottom). The grey stripes represent periods of recession.

Figure 1 above shows that there is a clear decline in the values during the recession (grey striped periods), especially during 2001 – 2002, and 2007 – 2009. The decline is most profound during 2007 – 2009, which is also known as the Great Recession, when a significant number of economies around the world collapsed. The volume of stocks traded also illustrates the impact of the Great Recession. At its peak, a record number of stocks were traded (~11 billion), suggesting that many investors panic sold given the massive economic uncertainties of the time.

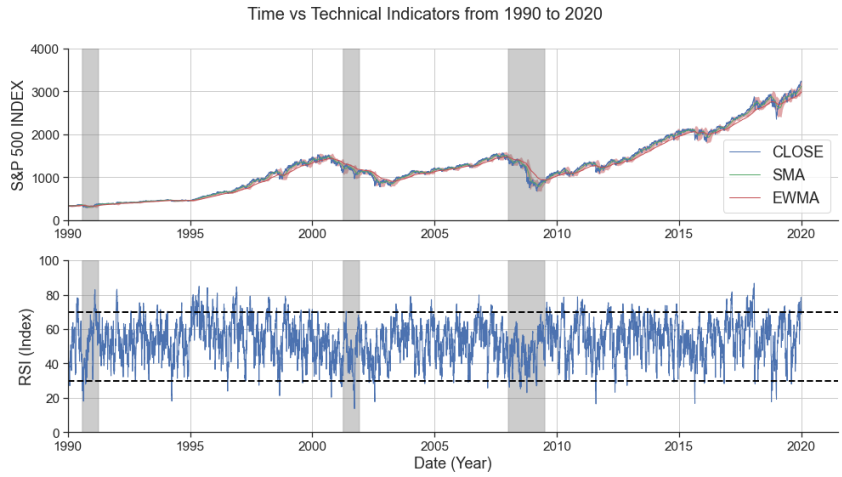

Figure 2. Change in S&P 500 Technical Indicators – Simple 50-day Moving Average (SMA) and 200-day Exponential Weighted Moving Average (EWMA) (top) and Change in Relative Strength Index (RSI) Over Time (bottom). The grey stripes represent periods of recession.

Both SMA and EWMA shown in Figure 2 closely resemble the pattern of S&P 500 index, as they should.

The most interesting observation is the behaviour of RSI. Usually, if the RSI goes below 30, it indicates an oversold (sold below actual value) market, whereas the RSI going above 70 indicates overbought (bough above actual value) market. According to Figure 2, the S&P 500 stocks were bought at higher price than its actual value during 1995 – 1996, and during 2017 – 2018. Both periods are few years removed from the recession. Conversely, the stocks were typically sold at a lower price than its actual value during recession periods, which again illustrates how investors tend to panic sell when there are uncertainties.

2.2.2 Other Stock Indices, Economic Indicators, Commodity Prices, and Consumer Sentiments.

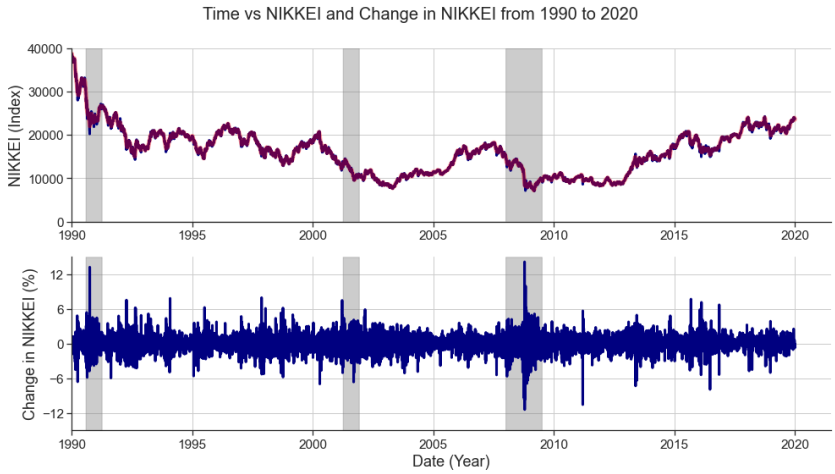

Figure 3. Change in Nikkei Index Over Time (top) and Its Daily Percentage Change (bottom). The grey stripes represent periods of recession.

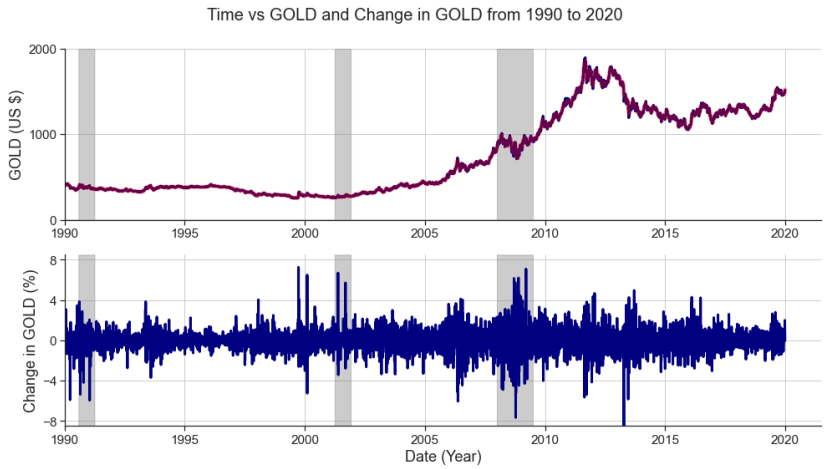

Figure 4. Change in the Price of Gold Over Time (top) and Its Percentage Change (bottom). The grey stripes represent periods of recession.

Most other stock market indices followed a similar pattern as we have seen in Figure 1. However, the Nikkei Index was different (Figure 3) as it started from its peak in 1990 following the peak of Japanese economic bubble and began to decline once the bubble had burst. It showed a sign of recovery during the 2000s until the Great Recession hit the world economy. The extent of damage caused by the Great

Recession on the Japanese economy and its major stock market seems to be on par with the US and other world economies. Interestingly, the price of gold (Figure 4) and other commodities did not seem to suffer as much during the Great Recession (± 12% change in stock market indices compared to ± 8% change in gold price). This data reinforces the idea that investors seek lower risk investments during the recession such as the US Treasury Bonds (guaranteed by the US government) and basic commodities like gold (mostly regarded as the most stable and consistent in terms of value).

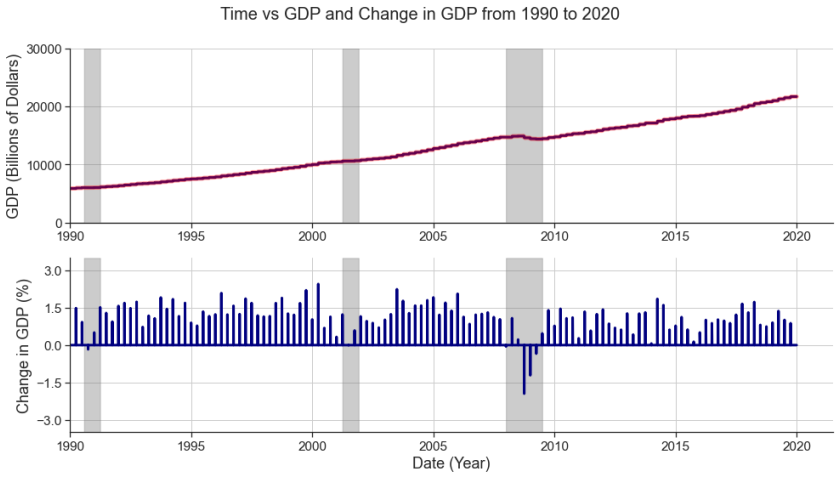

Figure 5. Change in GDP (USA) Over Time (top) and Its Percentage Change (bottom). The grey stripes represent periods of recession.

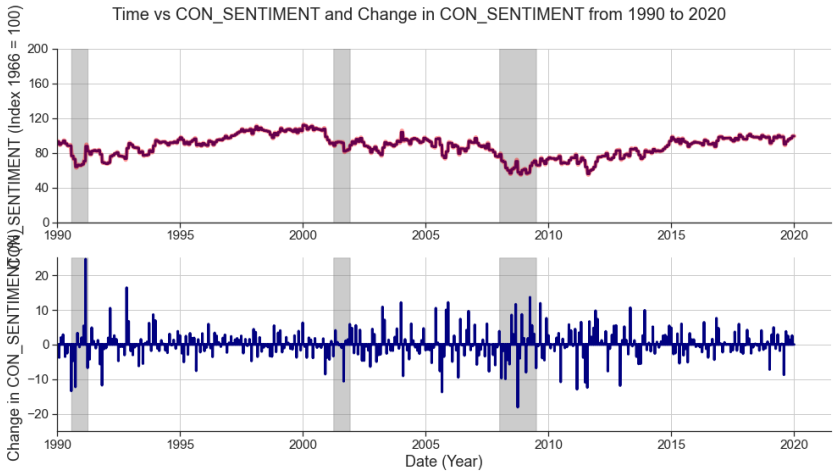

Figure 6. Change in the Consumer Sentiment Over Time (top) and Its Monthly Percentage Change (bottom). The grey stripes represent periods of recession.

Figures 5 and 6 further illustrate the blunt impact of the Great Recession. The GDP for USA in general, increased every quarter by around 1 – 1.5%. It was only during the Great Recession that the GDP for USA failed to grow, when it fell from its previous value by more than -1.5% (Figure 5). The consumer sentiment in Figure 6 supports the observations made from previous figures, that the economic uncertainty was unlike any other previously experienced and thus many investors lost confidence in the market (as can be seen by its lowest point in Figure 6), leading them away from investing.

3. Machine Learning Modelling

3.1 Feature Engineering

3.1.1. Feature Correlations

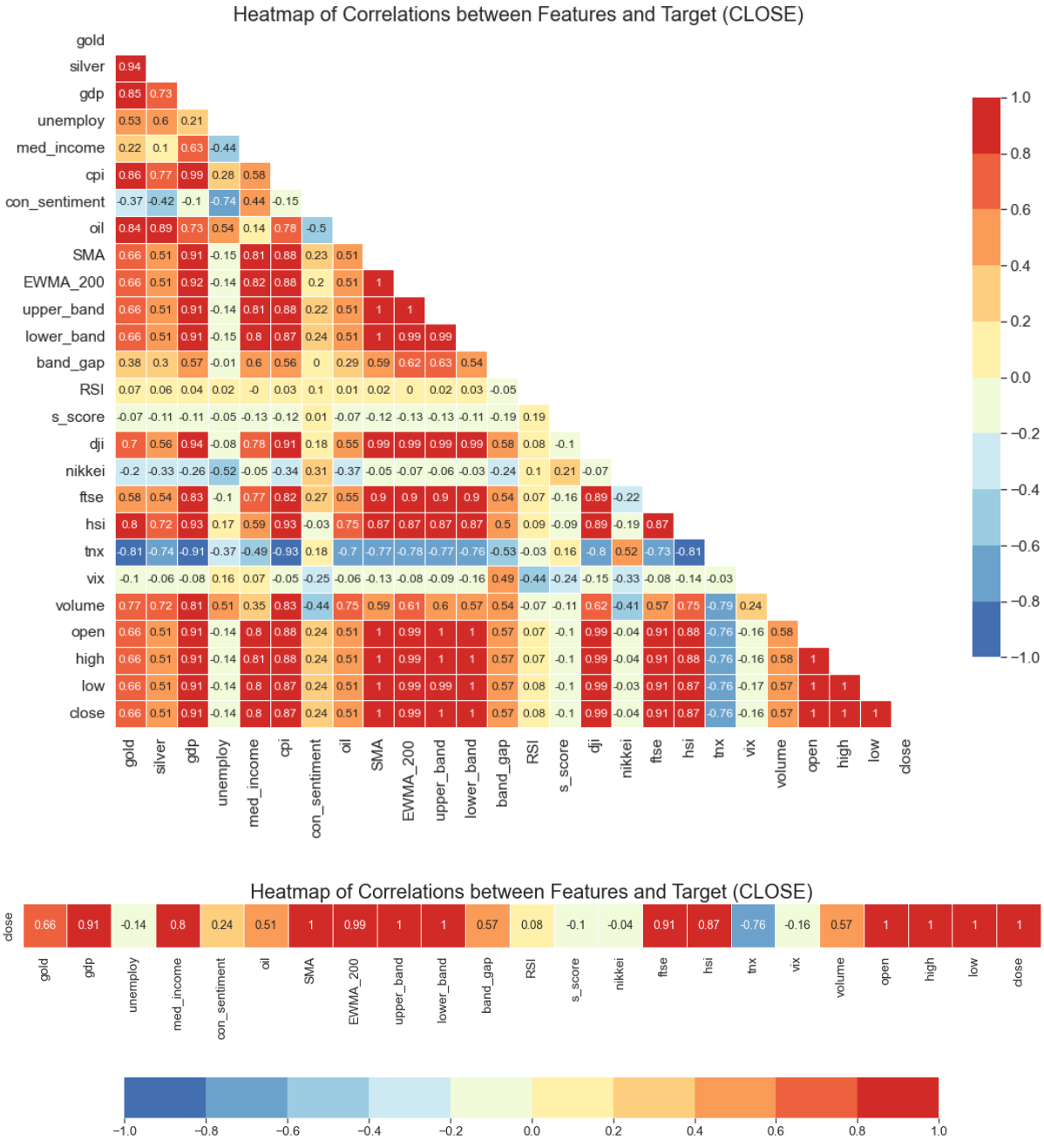

Figure 7. Heatmap of All Feature Correlations (top) and Heatmap of Correlations between Selected Features and Target (Close) (bottom).

From Figure 7, we found that some features have very strong correlations (> 0.95) with features other than the target variable, which is the "close" value of S&P 500. These features are:

- Gold and silver (0.94)
- GDP and CPI (0.99)
- Dow Jones and S&P 500 technical indicators (0.99)

To avoid multicollinearity, we have decided to remove silver, CPI, and Dow Jones variables from the dataset.

3.1.2. Lag Features and Multi-Output Targets

Because our objective is to make a 30-day forecast based on the previous 14 days of records, we need to create additional "lag features" and "multi-output" targets.

This means that for a feature X, we need to set 14 previous records of X as its own feature. So, in a single row, there will be additional 14 lag features named X_t, X_{t-1}, X_{t-2}, …, X_{t-14}. Here, a lag is a fixed amount (in our case, day) of passing time. Note that in this case, the previous records of the target variable also become features. In other words, previous 14 records of the closing value of S&P 500 become additional features. Figure 8 below shows that there is a very strong positive autocorrelation in the data, meaning that past values (up to 48 lag features tested) have a very strong influence on the current value.

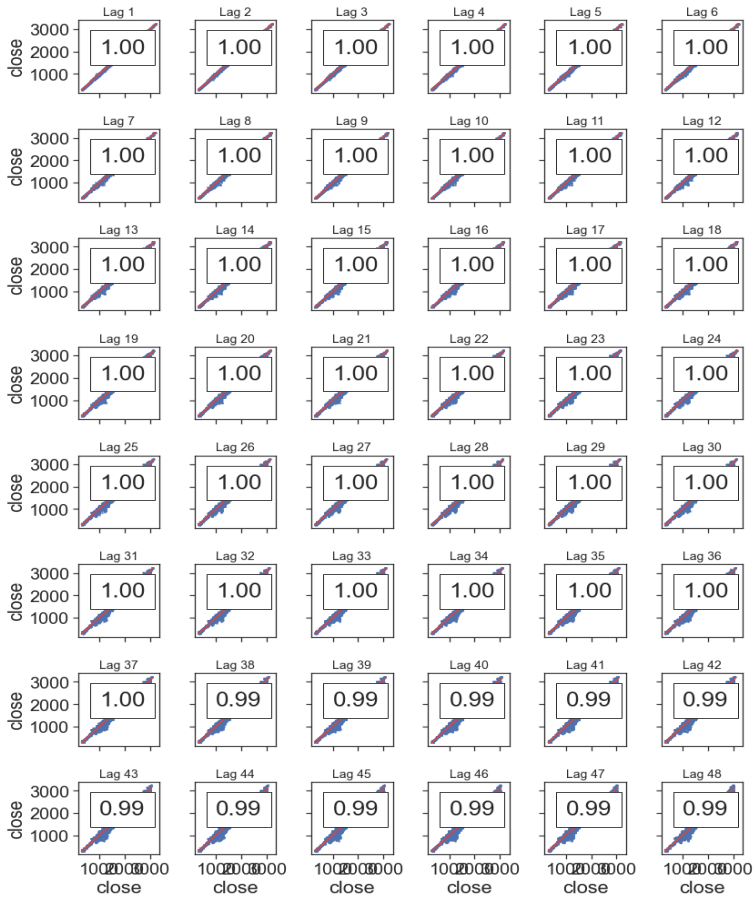

Figure 8. Lag Plots Showing Correlation Between the Current Close Value of S&P 500 and Its Past Values.

Unlike other conventional machine learning problems where there is only one target variable, here we will have 30 targets, each representing future values of the close value of S&P 500 for the next 30 days. So, each row of the target will contain 30 variables, named Y_t, Y_{t+1}, Y_{t+2}, …, Y_{t+30}.

Therefore, after feature engineering, we had **308 training features** (14 x 22 original features) and **30 output variables**.

3.1.3. Train and Test Datasets

The sequence of the data must be maintained since we are dealing with a time-series. Rather than randomly shuffling the data and splitting into train and test datasets, we have chosen to **train** the model using the data from **1990 to 2017** and **test** the model using the data from **2018 to 2019**.

All values were standardised prior to training and transformed back to its original scale after prediction.

3.2 Machine Learning Algorithms

We tested a range of different machine learning algorithms to get a sense of how accurate our forecasts can be. The tested algorithms are:

1. AutoRegression (AR)
2. AutoRegressive Integrated Moving Average (ARIMA)
3. Univariate and Multivariate Linear Regression (LR)
4. Nonlinear Models (Support Vector Regressor (SVR), Random Forest Regressor (RF), XGBoost Regressor (XGB))
5. Hybrid Models (combining LR with SVR, RF and XGB)
6. Deep Learning Models (Multi-Layer Perceptron (MLP), Long-Short Term Memory (LSTM))

We used root square mean error (RMSE) as our scoring metrics. The lower the value of RMSE the more accurate the forecast is to the actual values.

Simple models such as AR and ARIMA models, and LR models took no longer than 1 – 2 minutes to train. More complex models on average took around 10 minutes to fully train.

3.2.1 AR and ARIMA models

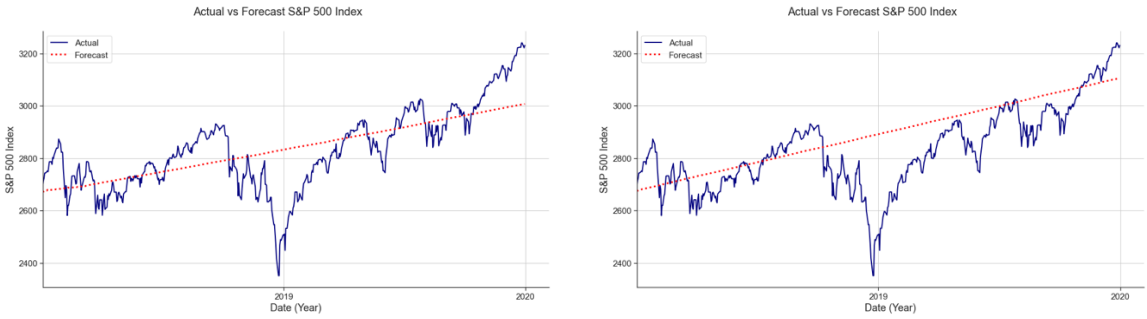

Figure 9. 2-Years Forecast Using AR (left) and Using ARIMA (right).

Both AR and ARIMA models performed similarly, giving a basic forecast for the next 2 years as a single straight trendline. AR and ARIMA models had a train RMSE of 1.69 and 1.91, respectively. However, their test RMSE were **111.42** and **129.83**, respectively, indicating over-fitting.

3.2.2 LR models

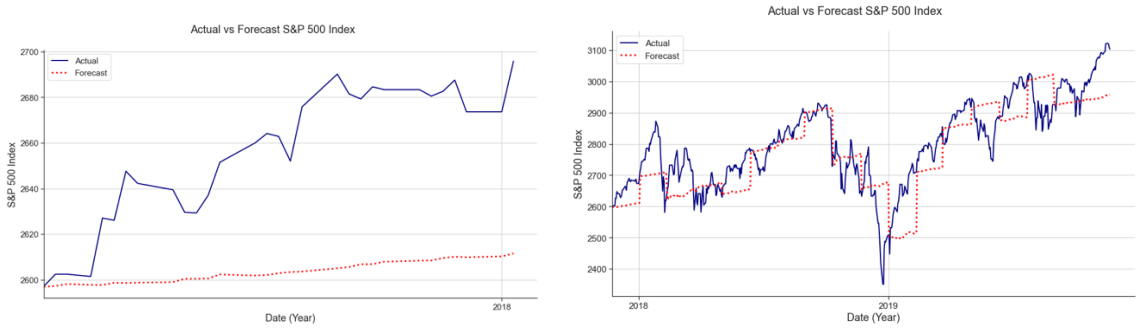

Figure 10. First 30-Day Forecast (left) and 2-Years Forecast Using LR Model.

The first 30-days forecast using a univariate LR model was not very accurate as shown in Figure 10, with a test RMSE of 57.39. The 2-years forecast had a test RMSE of **85.61**. Training of the model resulted in RMSE of 36.9, indicating less over-fitting than AR and ARIMA models. Including multiple features for training did not significantly improve the overall test RMSE.

3.2.3 Nonlinear models

The optimum hyperparameters for each nonlinear model were searched for using GridSearch method.

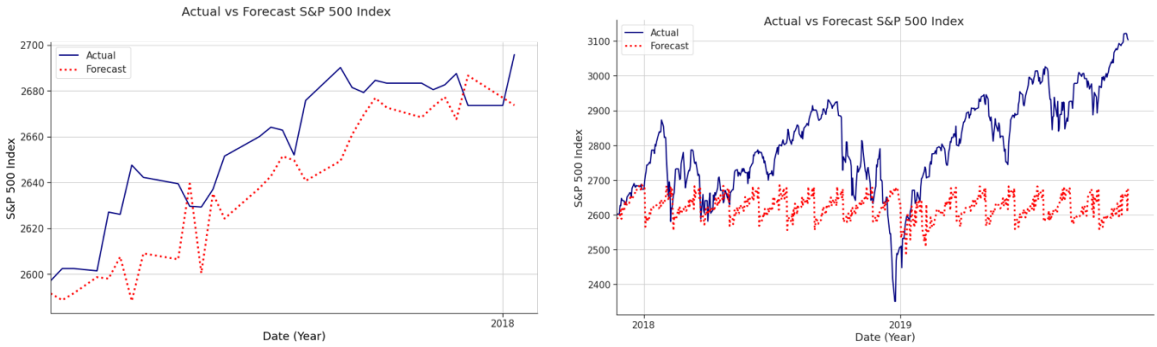

Figure 11. First 30-Day Forecast (left) and 2-Years Forecast Using RF Model.

Unlike the LR model, the first 30-days forecast using a nonlinear model performed quite well, with a test RMSE of 22.35 (Figure 11, result for RF model which was the best-performing nonlinear model). However, the 2-years forecast had a test RMSE of **225.56**. Other nonlinear models similarly performed poorly over 2-year period. From Figure 11, we can see that nonlinear models failed to learn the general trend (or non-stationary component of the time-series) and basically remains stationary over time. Training of the model resulted in RMSE of 7.94, indicating a very severe over-fitting.

3.2.4 Hybrid models

The idea behind hybrid models is to combine the best of two worlds – in our case, linear and nonlinear

models. First, we employ a simple linear model to learn the trend of the time-series. Then, we employ a second nonlinear model to learn the residuals that resulted from the first model. The residuals of the first model represent any seasonality, cyclicity or irregular patterns remaining in the time-series after the trend component has been removed.

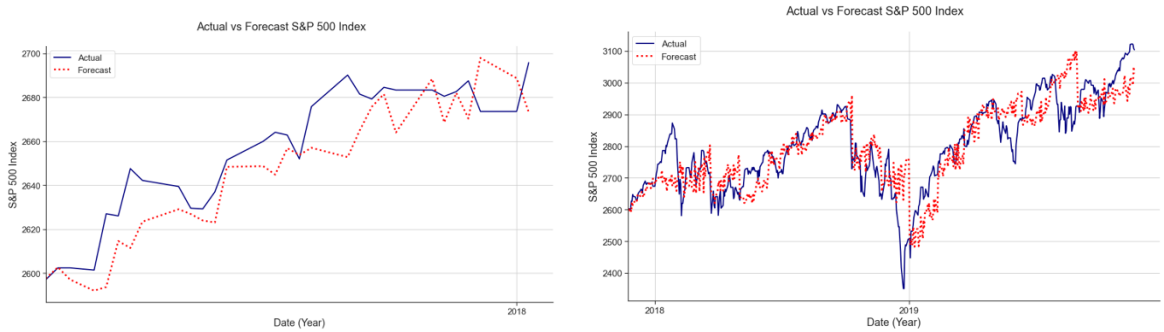

Figure 12. First 30-Day Forecast (left) and 2-Years Forecast Using Hybrid LR + RF Model.

Combining LR and RF models significantly improved both the first 30-days forecast and the overall two-years forecast, with a test RMSE of 16.44 and **81.24**, respectively. Figure 12 shows that much of the errors that were observed in nonlinear models have been eliminated. There are still some misses, but this is likely because the model requires re-training.

3.2.5 Deep Learning models

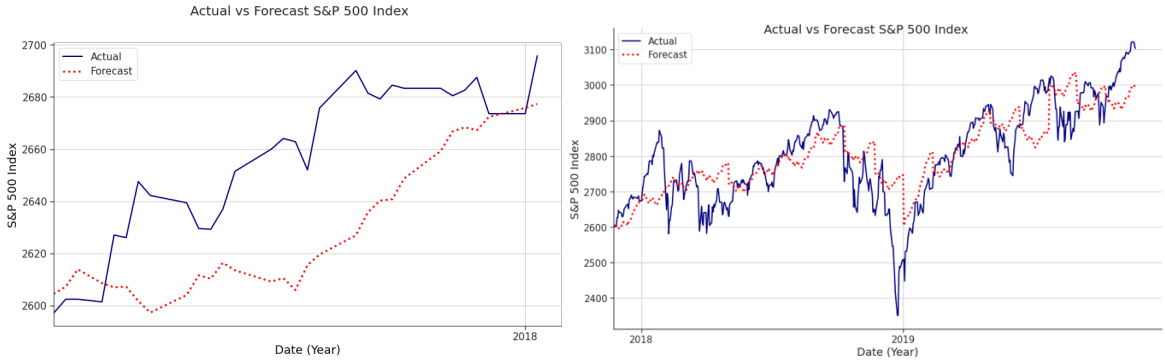

Figure 13. First 30-Day Forecast (left) and 2-Years Forecast Using MLP.

Deep learning models using Keras and TensorFlow libraries were trained and tested using Google Colab.

After searching for the best hyperparameters and neural network architecture, the final MLP model consisted of 3 hidden layers with 10 (first layer), 50 (second layer), and 20 (third layer) nodes.

The MLP model performed quite favourably compared to the best hybrid model, capturing the overall trend for the 2-years period as well as accurately forecasting the first 30 days (Figure 13). The test RMSE was 33.94 for the 30-days forecast, and **84.52** for the 2-years forecast. The train RMSE was 24.55.

Unexpectedly, the LSTM model performed very poorly with a 30-days forecast RMSE of 138.09 and a

2-years forecast RMSE of 136.4. The issue was that the datasets could not be standardised and transformed back to its original scale because of the specific data shape requirement set by the LSTM model. Therefore, unlike other models, the model was trained on unstandardised values, which led to greater error.

3.3 Summary of Results and Best Model Chosen

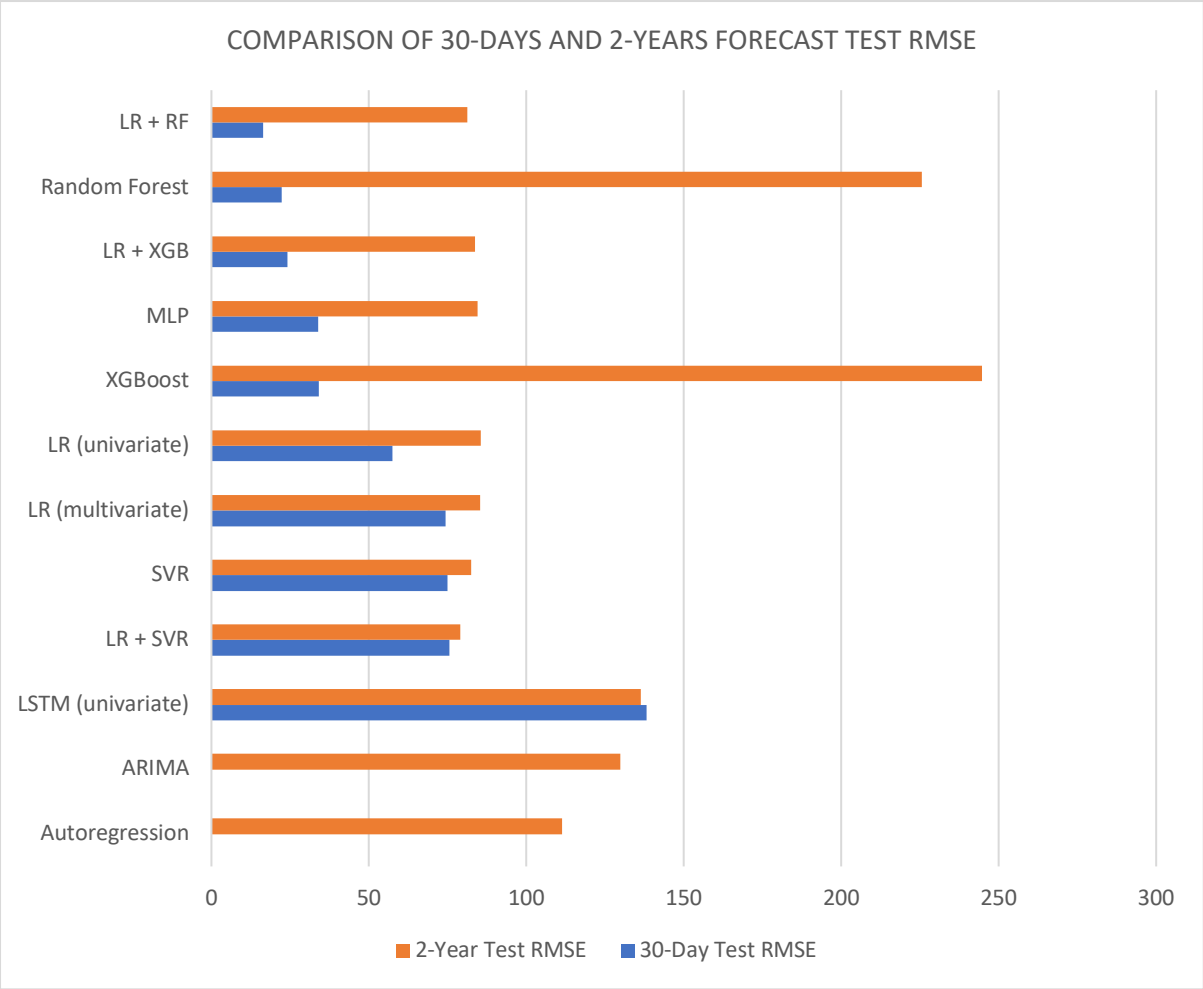

Figure 14. Summary of Test RMSE Scores for Each Model.

According to Figure 14, the best overall model is the **hybrid LR + RF model**.

4. Conclusions and Recommendations

4.1 Conclusions on Modelling Objectives

We have satisfactorily built a predictive model that can accurately make mid-term and long-term forecasts. The final chosen model is hybrid LR + RF model with a 30-days forecast RMSE of 16.44 and a 2-years forecast RMSE of 81.24. The mean average percentage error for the model is around 0.01,

meaning that the average error between the forecasted values and the actual values were around 1%. Therefore, we have met the objective of building an accurate predictive model.

We have also streamlined the process of acquiring data and transforming the data to be used by the model by incorporating APIs and user-defined helper functions. This means that there is minimum input and more importantly minimum coding experience required by our stakeholders when they use the model on their own. Therefore, we have also met the objective of automatically acquiring the required data.

4.2 Conclusions on Business Objective

The business question set out by our stakeholders was:

"How much **more investment return** can we provide to our stakeholders by providing **easy** and **accurate** future stock price forecast?"

We answer this question with a following scenario:

In the late 2018 there was a sudden and sharp fall in S&P 500 index, from around 2600 to around 2400. However, in a couple of months the index value recovered back to its original position. This would have been a perfect opportunity to buy low and sell high, if we knew what was likely to happen at the time of stock prices falling. Specifically, if we assume that the S&P 500 index value represents the dollar value (i.e., $2400 and $2600), then we would have potentially gained around 10% return on our investment.

A simple forecast based on linear regression or AR would suggest that no significant increase in stock price is likely over the next few months, so given the uncertainties, it would recommend not to buy any stocks. If we followed this recommendation, then we would have lost the opportunity to gain 10% return.

However, our model forecasted that the stock price will go up in the next two months, so it would recommend buying more stocks. Following this recommendation, we would have gained 10% return on our investment.

In other words, if our stakeholders invested $10,000 upon following our model recommendation, then they would have gained a profit of $1,000 during the two months span. This may not sound much, but for context, one would expect to gain around $14 for the same amount of investment if the money was deposited into a 30-day term deposit (1.7% per annum, based on 2018 interest rate).

So, we are confident that our stakeholders will be able to make better investment decisions and hopefully make more investment returns by using our model forecasts.

4.3 Recommendations

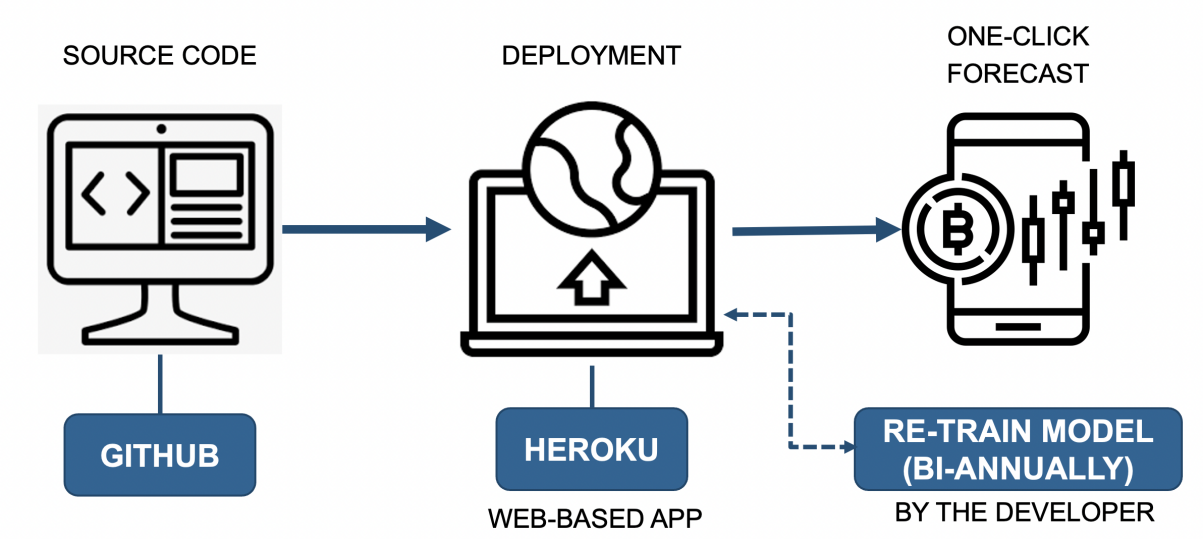

Figure 15. Next Step in Our App Building Process.

The source code is made available in GitHub Repository[7] for our stakeholders to use.

The next step is to deploy our model as a web-based app. We recommend using Heroku for deployment. Once deployed, the goal is to be able to provide our stakeholders what we call a "one-click forecast" where the model will automatically acquire the required data and make forecasts after a single click. We will need to continually re-train the model every 6-months or more frequently during the time of recession to ensure the accuracy of model forecasts meet our stakeholders' expectation.

In the meantime, it is also recommended we continue to explore more finetuning of model hyperparameters to see if we can improve the overall accuracy than currently reported in this report.

References

1. Statista Research Department, https://www.statista.com/statistics/270126/largest-stock-exchange-operators-by-market-capitalization-of-listed-companies/ (Oct 2022)
2. Yahoo Finance API Library, https://pypi.org/project/yfinance/ (2022)
3. NASDAQ API Library, https://data.nasdaq.com/tools/api (2022)
4. Alpha Vantage API Library, https://www.alphavantage.co/documentation/ (2022)
5. Federal Reserve Economic Data, https://fred.stlouisfed.org/ (2022)
6. New York Times API, https://developer.nytimes.com/apis (2022)
7. Chang-Joon Lee GitHub Repository, https://github.com/oneway1225/IOD-Capstone-Project (Oct 2022)

Source code for some of the helper functions used in the project are from:

- QuantInsti, https://blog.quantinsti.com/build-technical-indicators-in-python/ (2022) Used for Generating Technical Indicators
- Kaggle, https://www.kaggle.com/code/ryanholbrook/time-series-as-features (2022) Used for generating lag features

These codes have been modified to suit the purpose of the project.

Appendix – Definitions for Some of the Economic and Technical Indicators

- S&P 500, Dow Jones, Nikkei, FTSE 100, and Hang Seng Index are major stock market indices indicating how well the stocks around the world are performing.

- **Chicago Board Options Exchange (CBOE) Volatility Index (VIX)**: A popular measure of the stock market's expectation of volatility based on S&P 500 index options.

- **Treasury Yield 10 Years Bond**: Effective annual interest rate that the US government pays on one of its debt obligations, expressed as a percentage. It is the annual return investors can expect from holding a US government security with a given maturity. Because they are backed by the US government, Treasury securities are seen as a safer investment relative to stocks.

- **Gross Domestic Product (GDP):** GDP measures the output of all goods and services in an economy. As the stock market rises and falls, so too, does sentiment in the economy, either negatively or positively. In a bull market (stock prices rising), consumers and companies have more wealth and confidence, leading to more spending and higher GDP.

- **Consumer Price Index (CPI):** An index that measures the monthly change in prices paid by U.S. consumers. It is one of the most popular measures of inflation and deflation.

- **University of Michigan Consumer Sentiment Survey:** Illustrates the average U.S. consumer's confidence level. The higher the confidence level, the more consumers are willing to spend money, and therefore increases stock prices, especially those stocks of car manufacturers, home builders, and other retailers.

- **Moving Average:** The mean or average of the specified data for a given set of consecutive periods. As new data becomes available, the mean of the data is computed by dropping the oldest value and adding the latest one. It can smoothen the short-term fluctuations and reduce the temporal variation in data.

- **Bollinger's Band:** A volatility or standard deviation-based oscillator which comprises of three components – middle band (moving average line), upper, and lower bands (usually two standard deviations away from the middle band). As the volatility of the stock prices changes, the gap between the bands also changes. During more volatile markets the gap widens and amid low volatility conditions, the gap contracts.

- **Relative Strength Index:** A momentum oscillator to indicate overbought (overvalued owing to excessive buying at unjustifiably high prices) and oversold (sold at a price below its true value) conditions in the market. It oscillates between 0 and 100 and its values are below a certain level. Typically, if the RSI line goes below 30, it indicates an oversold market whereas the RSI going over 70 indicates overbought conditions.