Time Series Case Study of JNJ Stock Price

November 26, 2022

Contents

1 Introduction

To protect the economy from the influence of COVID-19, the U.S. Federal Reserve announced a quantitative easing plan on March 15, 2020. Quantitative easing is a monetary policy in that a central bank buys long term securities to increase the money supply and encourages investments. It has influenced many companies in the U.S. stock market throughout 2020 and 2021.

In this project, we leveraged the historical stock price of Johnson & Johnson from Yahoo Finance. from 2016 to 2021 to characterize the stock price movement by creating two time series models: ARIMA and VAR respectively and one supervised learning model using gradient boosted trees.

We first divided the time series data set into two sub-periods, one before March, 2020 denoted before Q.E. period, and one after March, 2020 denoted as during Q.E. period. Further analysis is done on the whole data set (from 2016 to 2021), the before Q.E. data set (2016 to March, 2020), and during Q.E. data set (after March, 2020) respectively. We then implemented two traditional time series models, the ARIMA model and VAR model, and one supervised learning model, gradient boosted trees, on the entire data set and on each sub-period. In the time series models, for each data set period, we used the first $\frac{2}{3}$ (66%) of the data as a training set and the remaining $\frac{1}{3}$ (33%) of the data as a testing set for ARIMA and VAR model. For gradient boosted trees, we use used the first 50% of the data as training set, 20% of the data as validation set, and the remaining 30% of the data as the test set. We followed the general procedure of time series modelling, which will be discussed in later sections, to construct the ARIMA model, VAR model and gradient boosted trees model. Afterwards, predictions were made based on the model constructed.

By comparing the difference in predictions across the three models, we may conclude our findings about the J&J stock price movement characteristics for different periods, based on which, we may make some causal inferences. Also, traditional time series models and the supervised learning model were compared. All in all, our report can be used as a guideline for time series model constructions, and could provide some insights into choosing between traditional time series and supervised learning models under specific circumstances.

2 Methodology

In this section, we will go through the process of constructing each model respectively. This section could serve as a handbook for time series model construction for further reference in the future.

2.1 Time Series Overview

A time series is a sequence of numbers in chronological order. Time series analysis is to use this series of numbers, applied with statistical models, to predict future developments. A time series can be decomposed into four components:

- **Secular Trend**: movements along the term
- **Seasonal Variations**: seasonal changes
- **Cyclical Fluctuations**: periodical but not seasonal variations
- **Irregular Variations**: other nonrandom sources of variations of series.

Time series models are implemented to characterize these four components.

2.2 ARIMA Model

ARIMA is short for Autoregressive Integrated Moving Average Model. It is the most commonly used basic model for time series analysis. The ARIMA model can be seen as a class of models. It is a generalization of the simpler Auto-regressive Moving Average Model and adds the notion of integration. The model characteristics can be decomposed to three components:

- **AR (Autoregressive Model)**: A model correlates the output variable with the linear combination of its own past values with certain lags applied
- **I (Intergrated)**: A difference of a certain order applied to the raw observations (e.g. subtracting each observation from observation that is one-time-step ahead) in order to obtain stationarity
- **MA (Moving-average Model)**: A model that correlates the output variable with the current residual error and residual error from the lagged observations

Each of these components are explicitly specified in the model as a parameter. A standard notation is used of ARIMA(p,d,q) where the parameters are defined as follows:

- **p (Lag Order)**: The number of lag observations included in the model
- **d (Order of Differencing)**: The number of times that the raw observations are differenced
- **q (Order of Moving Average)**: The size of the moving average window

We built three separate ARIMA models by choosing appropriate parameters to model the three periods[1] defined in previous section respectively.

[1]The three periods are the complete time series data from 2016 to 2021, the before Q.E. data from 2016 to March, 2020, and the during Q.E. data from March, 2020 to October, 2021

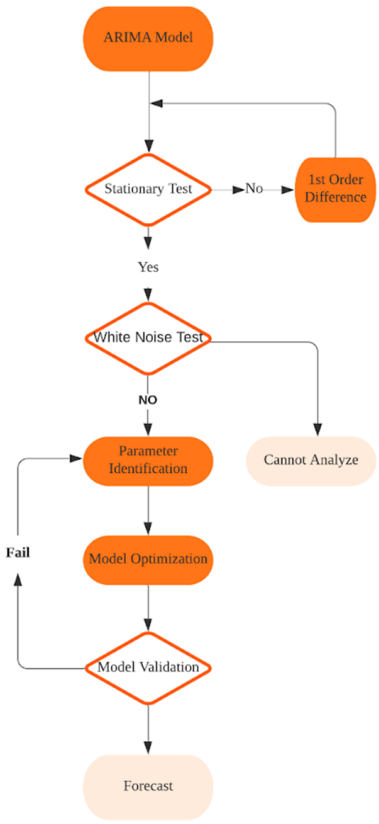

Figure 1: ARIMA Model Construction Procedure

Figure 1 is a flow chart illustrating the general procedure for ARIMA model construction, which will be discussed in detail step by step.

2.2.1 Stationary Test

Intuitively, the stationarity is the requirement that the fitted curve obtained from the sample time series can continue along the existing trend in the future period. It is a basic assumption on which time series analysis is based. Only predictions based on stationary time series are valid. Therefore, we would first implement a stationary test to check whether the time series is stationary or not, which are usually done in two ways:

- **Sequence Plot (Subjective)**: If the curve in the sequence plot shows a trend that fluctuates up and down around the value of 0, and the amplitude of fluctuation is consistent, then it's stationary
- **Unit Root Test (Objective)**: Implementing a hypothesis test to check whether a unit root exists or not

By observations from the sequence plot, if the fitted curve fluctuates up and down around the value of 0, and the amplitude of fluctuation is consistent, then the time series is stationary. If the curve has no definite trend, and the mean value and variance fluctuate greatly, then it is not a stationary series. Observations of sequence plot depends on visual judgment and personal experience. Different people may have different judgments based on the sequence plot. Therefore, we commonly use the unit root test.

There are several unit root tests available, each unit root test is a hypothesis test. Commonly used unit root tests include:

- **ADF (Augmented Dickey-Fuller) Test**: The most commonly used test. Null Hypothesis: Unit root exists, non-stationary time series; Alternative Hypothesis: Unit root does not exist, stationary time series.
- **PP (Phillips-Perron) Test**: Supplementary to the ADF test. Same Null and Alternative Hypothesis as the ADF test.
- **DF-GLS (Dickey-Fuller with GLS Detrending) Test** : Most efficient test. A 'quasi-difference' is performed on the data to be tested, and then the original sequence is de-trended using the quasi-difference data before testing. Same Null and Alternative Hypothesis as the ADF test.
- **KPSS (Kwiatkowski–Phillips–Schmidt–Shin) Test**: Null Hypothesis: Unit root does not exist, stationary time series; Alternative Hypothesis: Unit root exists, non-stationary time series.

In addition to these tests above, there are several other tests, such as Zivot-Andrew Test and Variance Ratio Test, that could be implemented to test the stationarity. The core idea here is to calculate the test statistics and the P-value in order to decide whether to reject the null hypothesis or not.

If the time series is non-stationary, we have to difference the data in order to stabilize the time series. Differencing stabilizes the mean of a time series by removing some of its variation features, thereby eliminating (or reducing) the trend and seasonality of the time series. There are mainly three options for differencing:

- **Random Walk Model**: A time series consisting of changes in the original series of consecutive observations.
- **Second Order Differencing**: A time series consisting of changes in the original series of consecutive observations (changes in the random walk model).
- **Seasonal Differencing**: Differencing between an observation and the corresponding observation one year ago.

After implementing these differencing methods, stationary tests are performed again to check whether the differenced time series are stationary or not. If yes, then we could proceed to the next step, if not, higher order differencing may be implemented until stationarity is obtained.

2.2.2 White Noise Test

After obtaining a stationary time series, the next step is to determine whether the data is white noise or not. For white noise (also known as pure random sequence), there is no correlation across different terms in the sequence, and the sequence is undergoing completely disordered random fluctuation, so the analysis of the sequence can be terminated. White noise sequence is a stationary sequence with no information to extract, thus, it's meaningless to study. There are several ways to check whether the time series is a white noise or not:

- **LBQ (Ljung-Box/Q) Test**: Most commonly used. Null Hypothesis: Time series data are independent, it's a white noise; Alternative Hypothesis: Time series data are not independent, it's not a white noise.
- **ACF/PACF (Autocorrelation/Partial Autocorrelation) Plot**: If the ACF/PACF plot shows a trailing or truncating of zero order, then the time series is a white noise.
- **BG (Breusch-Godfrey) Test**: Null Hypothesis: There is no serial correlation, the time series is a white noise; Alternative Hypothesis: there exists serial correlation, the time series is not a white noise.
- **DW (Durbin Watson) Test**: Only applicable for first order autocorrelation. If the test statistic d is close to 2, then there is no autocorrelation, thus, the time series is a white noise.

After implementing the white noise test, if the time series is not a white noise, we can proceed to the next step. If it's a white-noise, then we may conclude that the data is not analytical using time series models.

2.2.3 Parameter Identification

After verifying the stationary time series is non-white noise, the next step is to select the appropriate parameters for our model, that is, the appropriate number of lags: p and q.

This is usually done by directly observing the ACF (Autocorrelation) and PACF (Partial Autocorrelation) plots. ACF/PACF illustrates a characteristic of 'Cuts off' and 'Tail Off'. 'Cuts off' means that the value of the autocorrelation becomes zero abruptly as the number of lags increases, and 'tails off' means that the value of the autocorrelation decays to zero asymptotically (usually exponentially) as the number of lags increases. We could do some preliminary judgments on the models we will be using and identifying the corresponding parameters following the rules in table 1[2]:

	ARIMA(p, d, 0)	**ARIMA(0, d, q)**	**ARIMA(p, d, q)**
ACF	Tails Off	Cuts off after lag q	Tails Off
PACF	Cuts off after lag p	Tails Off	Tails Off

Table 1: Decision Rules for ARIMA Model Parameters

[2]The parameter d in ARIMA model is the order of difference as we explained in previous section

Note that sometimes the ACF/PACF plots may be ambiguous to decide the parameters' values and which model to implement. In this case, we would pick several candidate models for further analysis to help make decisions. For instance, if we observe a 'cuts off' on ACF after lag q, a 'cuts off' on PACF after lag p, and the 'tails off' is not obvious in both plots, then we could pick ARIMA(p, d, 0), ARIMA(0, d, q), and ARIMA(p, d, q) as candidates.

2.2.4 Model Optimization

After obtaining the candidates for our model, we will pick the optimal one. There are several indices we could refer to help with our assessment:

- **AIC (Akaike Information Criterion)**: Rewards goodness of fit assessed by the likelihood function, and includes a penalty that is an increasing function of the number of estimated parameters
- **BIC (Bayesian Information Criterion)**: An increasing function of error variance and number of parameters estimated, which results in a higher penalty than AIC does.
- **HQIC (Hannan–Quinn Information Criterion)**: Alternative to AIC and BIC.

We use AIC, BIC and HQIC as selection criteria. The idea behind the construction of these statistics can be seen as a trade-off between the goodness of fit and simplicity of the model. It is to impose a 'penalty' according to the number of independent variables while considering the residual error of fitting. In this case, a lower value of these criteria is preferred. Sometimes these criteria do not agree on the same model. Usually, we choose the model where the most criteria agree or with the lowest BIC value. However, we should always be aware that these criteria cannot explain the accuracy of a certain model, that is to say, for the three models, we cannot guarantee that the model with the minimum AIC, BIC or HQIC can describe the data best. A supplementary way to help make the selection is to look at the forecast accuracy on the testing set. We would use our testing set to test the forecasting accuracy. The model with the best overall performance in AIC, BIC, HQIC and forecasting accuracy will be selected as our final model.

2.2.5 Model Validation

Lastly, we will do a model validation test on the optimal model we selected. The main idea is to look at the residual to see if it's a white noise. If it's not, then there is still some information included in the residual term, and we need to make some adjustments to our model. There are basically two ways to test it:

- **LBQ (Ljung-box/Q) Test**: Test whether the residual is white noise or not.
- **Normal QQ-plot**: If most of the scatter points fall on the standard noraml KDE (kernel density estimation) line, then the residual is white noise.

The LBQ test on the residual follows the same pattern as we introduced before. Sometimes, it's more direct to check whether the residuals are standard normal distributed or not. If so, then the residual must be a white noise. However, if it's not, it could still be a white noise but not Gaussian white noise. We could either use LBQ test or Normal QQ-plot to check whether the residual is white noise or not, while if the residual is not Gaussian white noise, we have to do the LBQ test

to check whether the residual is ordinary white noise or not. If the residuals are white noise, the information contained in the time series data has been fully extracted by our model and the model passes the validation test. If not, we have to go back to parameter identification section and make adjustments until we validate the model.

2.2.6 Prediction

After validating the model, we could do some simple forecasts based on the historical data. The difference in these predictions obtained from different models will be discussed in later sections.

2.3 VAR Model

The vector auto-regression model, which is also known as VAR, is a forecasting algorithm that can be used when two or more time series influence each other. It captures the relationship between multiple time series as the variables evolve over time, including the lagged values and the error term. Therefore, before establishing the VAR model, the only information required is in regard to the variables, which could potentially be hypothesized to affect each other over time.

In our case, we applied VAR model to the open, high, low, close price for Johnson & Johnson's stock, which are correlated with each other by nature, and we built three separate VAR models by choosing appropriate parameters to model the three periods defined in the previous section respectively.

2.3.1 Data Examination

We first examined the four sequence plots for open, high, low, close price regarding the entire time scope as seen in Figure 2 below. Through observing the sequence trends, we noticed that each of the series indeed had a similar pattern over the years, indicating potential correlation between variables.

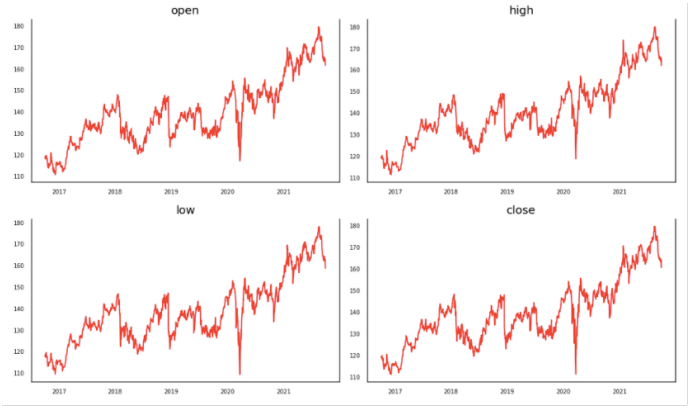

Figure 2: Sequence Plots for Johnson & Johnson's Stock Price: Open, High, Low, Close

Therefore, we considered building the VAR model starting off by examining the time series step by step with certain procedures. The flow chart, Figure 3, illustrates the general process for VAR model construction, and this is followed in later sections.

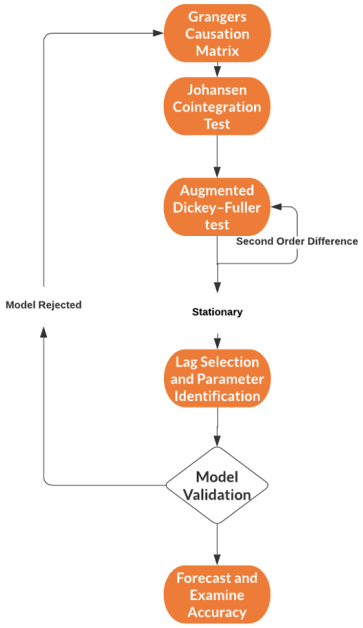

Figure 3: Flow Chart for Constructing a VAR Model

2.3.2 Granger's Causality Test and Johanson Cointegration Test

Unlike ARIMA model, the construction of VAR model began by testing causation and relationship between different time series using Granger's Causality Test and Johanson Cointegration Test.

For Granger's Causality Test, it is used to determine whether one time series is correlated with another: variable evolving over time can Granger-cause another evolving variable, as the prediction of variable's value is based on its own past values and the past values of another variable, meaning that this is better than using solely its own past values. Therefore, the null hypothesis of Granger's Causality Test states that the coefficients of past values in the regression equation is zero.

Also, Johanson Cointegration Test establishes the presence of a statistically significant connection between different time series. When two or more time series are cointegrated, it means that they have a long-run, statistically significant relationship, which creates the basis for establishing VAR model.

2.3.3 Stationary Test

Similar to ARIMA model, stationarity is the rule of basis for conducting time series analysis and the unit root test, specifically the Augmented Dickey–Fuller test, was used to check for the stationarity of VAR model. We difference the time series until it is stationary following the same pattern as that of the ARIMA model.

2.3.4 Parameter Identification

After passing the stationarity test, the construction of VAR model can then move on to the parameter identification stage, which determines the order of the VAR model, also known as the number of lags. Similar to ARIMA model, there are four statistics serving as the selection criteria for the right lag order of the VAR model, including AIC (Akaike Information Criterion), BIC (Bayesian Information Criterion), FPE (Akaike's Final Prediction Error) and HQIC (Hannan–Quinn Information Criterion). Here, the Final Prediction Error (FPE) statistic estimates the model fitting error and model quality when it is tested with respect to different data and to predict new outputs. Similarly, these statistics indicate a trade-off between the goodness of fit and the simplicity of the model, and a lower value of BIC is selected.

2.3.5 Model Validation

After setting up the VAR model and determining its parameters, the model validation should be performed by checking for serial correlation of residuals using Durbin Watson Statistic. If there exists any correlation left in the residuals, this means that there is some pattern in the time series model that is still left to be explained. Therefore, the typical course of action is to either increase the order of the model or to induce more predictors into the system, or even to look for a different algorithm to model the given time series.

Using the Durbin Watson's Statistics which ranges from 0 to 4 and checking serial correlation of residuals, the closer it is to 2, meaning that there is no significant serial correlation. The closer it is to 0 indicates that there is a positive serial correlation, and vice versa for the closer it is to 4.

2.3.6 Prediction

Having the model built up and validated, predictions can be done using the VAR model. In order to forecast, the VAR model expects up to the order of lag of observations from the past data. The forecasting results are generated, but it is based on the scale of the training data used by the model. So, to bring it back up to its original scale, we need to de-difference it as many times we had previously differenced the original input data.

2.4 Gradient Boosted Tree

The gradient boosted trees model, is a supervised machine learning model that is based on decision trees. Decision trees use input features to split the data from the target variable into different categories. Starting from a single input feature called a node, the data is split into subsets, where the number of subsets and splitting criteria obeys a set of rules chosen by the algorithm. When the data is split to subsets, each split is called a branch. After splitting the data, if the algorithm determines that one of the subset categorizations does not benefit from being split into further

subsets using more input features, then that subset is retained and is called a leaf. A loss function, which determines how accurate the subset is to the actual data, is used to determine if a subset should be a leaf. On the contrary, if splitting the subset into further subsets improves the subset categorizations, then another input feature is used to create a node and branches that will split the designated subset. The creation of nodes and splitting of subsets will continue until all the data is in leafs. The decision tree can also be pruned by the algorithm, which will stop after a certain number of branches, leafs, nodes have been reached or some other parameter conditions satisfied.

Gradient boosted trees improve upon the decision tree algorithm by using an ensemble of decision trees, where the output of the previous decision trees are used as the input for the next decision tree. This is achieved by using the residuals from the previous tree as the input for the next tree. The residuals of a decision tree is the data from the leaf subsets that have incorrect predictions for the target variable and is calculated using the loss function. So, the model is able to identify weak decision trees and use their results to improve the next iteration of the decision tree. The result is that the final decision tree is an aggregate of all the previous iterations of the decision trees and predicts the value of the target variable much more accurately than using a single decision tree. One of the drawbacks of using supervised learning methods based on decision trees for time series regression data is that the model will never predict values outside the training set, as the decision trees can not properly split those subsets of data with values outside the training set.

2.4.1 Temporal Feature Engineering

In the dataset for the Johnson & Johnson stock price, there are only two variables of interest, the close price and the date corresponding to the close. The only immediate feature of the dataset is the date, and the only target variable is the close price. So, implementing a gradient boosted trees model on the unaltered dataset will result in all future predictions being linear and inaccurate due to the lack of features. To potentially improve the results of the model, a variety of other features can be engineered from the date feature. These temporal features are year, season, month, week, weekday and the time difference between subsequent dates. The year feature is simply the value of the year in Gregorian calendar of the year in the date feature. The season can alternatively be thought of as a quarter, since it captures three-month periods of the year as a number between one and four. The month feature indicates the month of the year as a number between one and twelve. The week feature indicates the number of weeks since the beginning of the year, and as such is a number between one and fifty-two. The weekday is simply a number from zero to six, to indicate which day of the week it is. The time difference feature is the time difference between adjacent data points in the date feature, it has units of seconds.

The inclusion of these temporal features allows for the model to identify common trends based on the cyclical nature of the majority of the features. For example, a common phenomenon in the stock market, is the “Januaray Effect” where stock prices typically experience higher growth compared to the other months. Incorporating the month feature allows for the model to potentially capture the January effect and use it when making predictions about the close price. Another important consideration when using cyclical features as in the case of the season, month, week and weekday features is that the transition from the end of the feature domain to the beginning of the feature domain should be continuous. For example, when the date transitions from December to January, the transition is smooth, and in general stock prices do not change dramatically from December

31st to January 1st. However, with the current encoding of the month feature, there is a dramatic transition from December to January as the value of the feature changes from twelve to one. This sharp transition in values must be fixed for all cyclical features so that the transition is smooth. This is where cyclical feature encoding is necessary.

2.4.2 Cyclical Feature Encoding

Cyclical feature encoding transforms the feature values into two separate features using a cosine or sine transformation, the values of the combination of these two trigonometrically transformed features are cyclical. For cyclical features this method greatly improves the model compared to using the base features and typically improves the model results compared to using a one-hot-encoder. The feature sine transformation obeys the following formula,

$$x_{i,\sin} = \sin\left(\frac{2\pi x_i}{\max(x)}\right) \tag{1}$$

and the feature cosine transformation obeys,

$$x_{i,\cos} = \cos\left(\frac{2\pi x_i}{\max(x)}\right) \tag{2}$$

where $x_{i,\sin}$ and $x_{i,\cos}$ are the sine and cosine transformed i^{th} value of the feature respectively, x_i is the i^{th} value of the original feature and $\max(x)$ is the maximum value in the feature set.

The effect of this transformation on the feature can be seen by plotting the various components of the feature. The following is an example of cyclically feature encoding an hour feature, which takes values from zero to twenty-three.

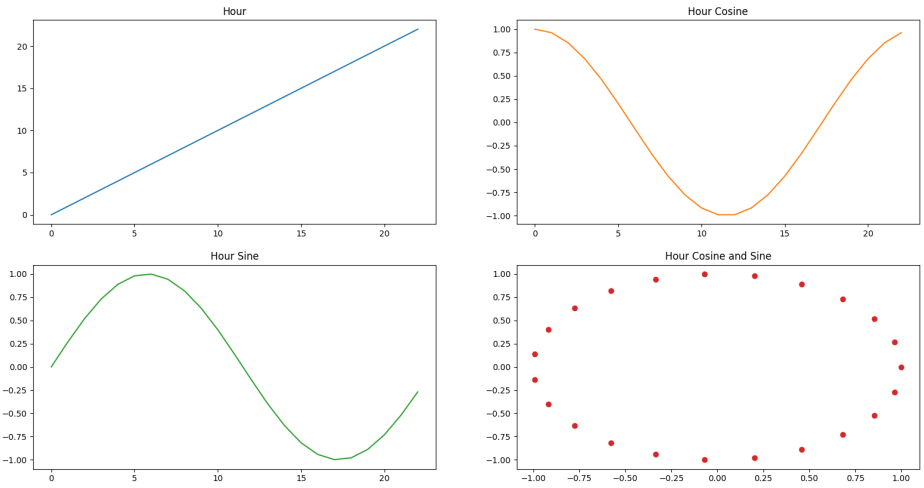

Figure 4: Cyclically encoding the hour feature using sine and cosine transforms.

The upper left plot shows the original values of the hour feature with respect to the number of data point index, unsurprisingly it is a perfect linear relationship since the data point index starts at zero and the feature value starts at zero. The upper right plot is the value of the cosine transformed feature with respect to the data point index, we can see that this plot closely resembles a cosine curve, as expected. The bottom left plot is the value of the sine transformed feature with respect to the data point index, we can see that this plot closely resembles a sine curve, as expected. Lastly, the bottom right plot is the value of the cosine transformed feature with respect to the sine transformed feature, this is where it is clearly obvious that the combination of the trigonometrically transformed features encodes a cyclical process into the original feature with the same period as the original feature.

Applying these methods to the season, month, week and weekday features in the data set for the Johnson & Johnson stock price allows for the model to interpret the cyclical nature of these features and make better predictions on the close price.

2.4.3 Financial Feature Engineering

To further improve upon the number of features available for the model to learn the close price, a variety of financial technical indicators were calculated from the close price and used to calculate future close prices. Special consideration was made when implementing these features in the model to ensure that there was not any look ahead bias. The choice of the technical indicators was chosen on the criteria of familiarity/common usage in the market, ability to calculate using only close price and the independence of the indicators to avoid multicollinearity or at least understand their correlation. The financial technical indicators chosen were the simple moving average (SMA) 20-day, the exponential moving average (EMA) 20-day, the moving average convergence/divergence (MACD) and the relative strength index (RSI). In addition to these technical indicators, a sliding window method was applied that uses a set number of the previous close prices to predict the current close price.

2.4.4 Sliding Window Method

The sliding window method uses a set number of previous target variable data points as features for the data point of the target variable currently being predicted. In the case of the Johnson & Johnson stock price data set, this was using a chosen number of previous close prices as features for predicting the current close price. The idea behind the sliding window method is that the value of the current target variable being predicted should be similar to the last few previous target variable values. This sliding window method for close prices was used in conjunction with the technical indicators to provide features based on the close price instead of just the date. The drawback of using features based on the target variable is that look ahead bias can be easily overlooked and accidentally implemented. As well, the model must predict only a single or handful of close prices before incorporating those newly predicted values as features and then training the model again and predicting another single or handful of close prices. This cycle of predicting and retraining the model causes the model to take much longer to predict the test set data compared to predicting the test set all at once, the other issue is that the predictions are further used to create more predictions and so if the model has a positive or negative feedback loop, the model will begin to predict either constant values for the target variable or extremely large/small target variable values. The benefit is that the model has a larger variety of features to use when making predictions on the target

variable. Below is a visualization of the sliding window method, showing how the sliding window chooses a set number of previous target variables to make predictions about the current target variable.

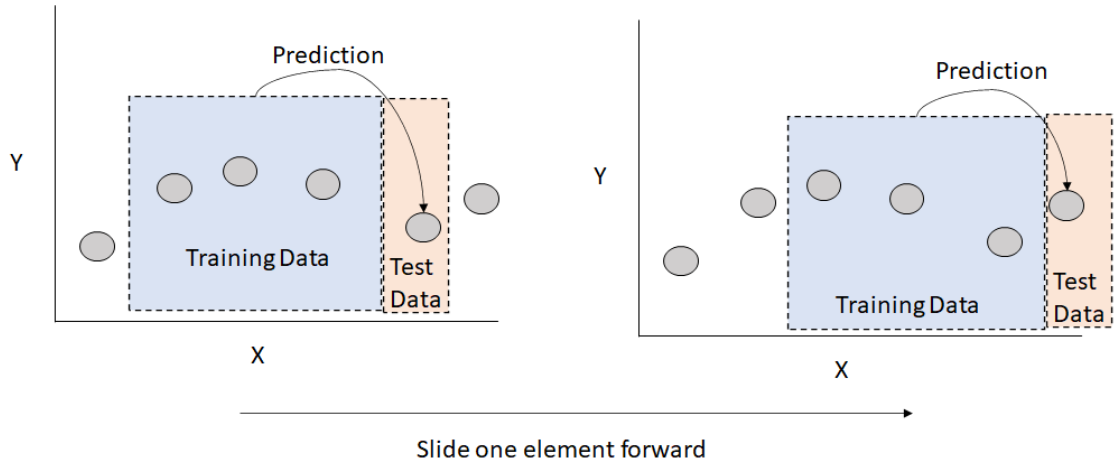

Figure 5: Visualization of the sliding window method. The advantage of the method is that previous target variable values can be used to make predictions about the future without incorporating look ahead bias.

2.4.5 Grid Search Hyperparameter Tuning

When implementing the gradient boosted trees model for the close price prediction in the Johnson & Johnson stock price dataset, the data was split into a training set, validation set and test set.

After engineering all the features for the model, the model parameters must be tuned to ensure it fits the training data accurately without overfitting. This is where hyperparameter tuning is important. Hyperparameter tuning searches the space of different parameters available for the model, attempting to find the set of parameters that does not overfit to the training set and accurately fits to the validation set. One such method is grid search, where a user-defined range of values are given for each parameter to be tuned, and the grid search method tests all the different combinations of parameter values to find the set of parameters that tune the model the best. The parameters that the grid search method were applied to for gradient boosted trees were gamma, learning rate (eta), max depth and number of trees. The parameter gamma controls the loss function value in each node, that is required to create a branch/split the subset further. The learning rate parameter controls how much of the residuals in the previous decision tree are input into the next iteration of the decision tree. Decreasing the learning rate can help to prevent overfitting to the training set. The max depth parameter controls how many times subsequent nodes can branch. The number of trees parameter controls how many decision trees are used in the gradient boosting algorithm. Below is a visualization of how the grid search hyperparameter tuning algorithm works.

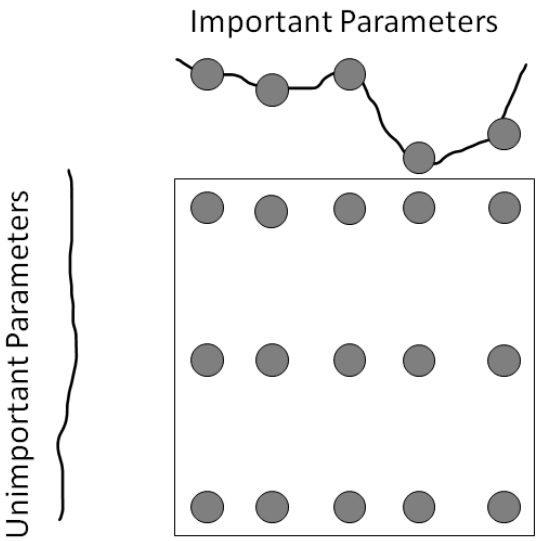

Figure 6: Visualization of the grid search hyperparameter tuning algorithm. The hope of the algorithm is that one of the parameter set combinations will find the minimum of the loss function.

3 Results

In this section, we will display step-wise outputs for our modelling procedure to show in detail how each model is constructed.

3.1 ARIMA Model

3.1.1 Preliminary Data Processing

We first divided the complete data set into two sub-periods, below are the sequence plots for the complete data set, the before Q.E. period, and the during Q.E. period.

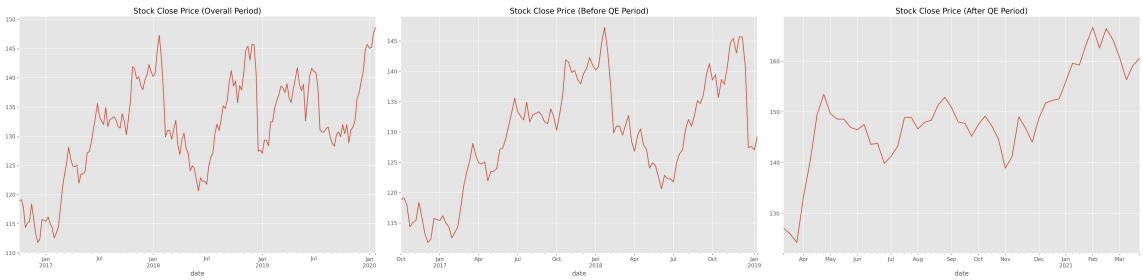

Figure 7: Sequence Plots

3.1.2 Stationary Test

Observing the sequence plots, we may notice that the original time series may be non-stationary as the amplitude of fluctuation is not consistent, and does not fluctuates up and down around the value of 0. We did ADF Test to formally check the stationarity. As shown in the ADF Test output, we notice that the p-value is greater than 0.05, indicating that we don't have enough evidence to reject the null hypothesis that the time series is non-stationary.

ADF Test (p-value)		
Overall Period	Before QE Period	After QE Period
0.26	0.24	0.2

Figure 8: ADF Test Output (Original Time Series)

To stabilize the time series, we applied a 1^{st} order difference to the original time series and redo the ADF Test. The test results below shows that the times series becomes stationary after applying 1^{st} order difference to it. At this stage, we identify the parameter $d = 1$ in the ARIMA model.

ADF Test (p-value)		
Overall Period	Before QE Period	After QE Period
9.25E-20	5.39E-14	1.64E-02

Figure 9: ADF Test Output (After 1^{st} order Difference)

3.1.3 White Noise Test

To check whether the stationary time series or not, we applied Ljung-box Test. As shown in the following output, the P-values are all smaller than 0.05, which means the three time series are not White Noise and we may proceed to the next step.

Ljung-box Test (p-value)		
Overall Period	Before QE Period	After QE Period
2.16E-02	1.90E-02	4.87E-05

Figure 10: Ljung-Box Test Output

3.1.4 Parameter Identification

To identify the corresponding parameters for the time series model, we plotted the ACF/PACF plots for each time series respectively. As shown in the graphs below, we observe a 'cuts off' after lag 1 in both ACF and PACF plots, which indicates the corresponding parameters $p = 1$ and $q = 1$.

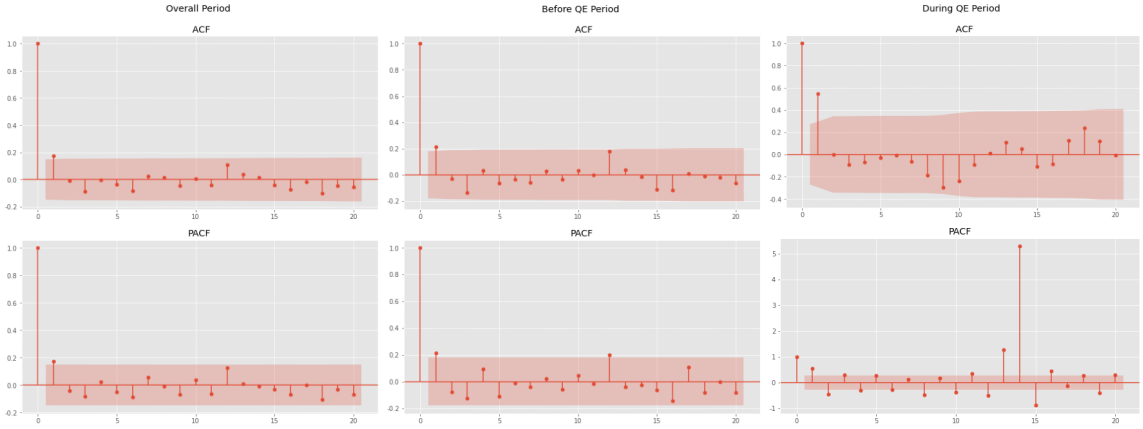

Figure 11: ACF/PACF Plot

Note that the 'tails off' characteristic is not very obvious. In this case, we developed three candidate models for further investigation: ARIMA(1, 1, 0)[3], ARIMA(0, 1, 1)[4] and ARIMA(1, 1, 1).

3.1.5 Model Optimization

We select the model with the best overall performance in AIC, BIC and HQIC as our final model for each time period. Below are the summary tables of AIC, BIC and HQIC values for each candidate model for each time period.

Overall Period			
	AIC	BIC	HQIC
ARIMA(1, 1, 0)	4940.08	4955.49	4945.87
ARIMA(0, 1, 1)	4941.17	4956.58	4946.96
ARIMA(1, 1, 1)	4939.95	4960.50	4947.67

Before QE Period			
	AIC	BIC	HQIC
ARIMA(1, 1, 0)	3245.09	3259.38	3250.56
ARIMA(0, 1, 1)	3244.74	3259.03	3250.21
ARIMA(1, 1, 1)	3245.90	3264.95	3253.19

During QE Period			
	AIC	BIC	HQIC
ARIMA(1, 1, 0)	1654.30	1666.20	1659.02
ARIMA(0, 1, 1)	1654.50	1666.41	1659.22
ARIMA(1, 1, 1)	1653.15	1669.03	1659.44

Figure 12: AIC, BIC, and HQIC for Candidate Models

Note that all criteria agree that ARIMA(0, 1, 1) should be selected for the before Q.E. Period.

[3]Also know as AR(1) with 1^{st} order differencing
[4]Also known as MA(1) with 1^{st} order differencing

However, when we select the model for the complete data set and the during Q.E. Period, AIC does not agree on the model selected based on BIC and HQIC. Since BIC and HQIC of ARIMA(1, 1, 0) are the lowest and AIC of this model is relatively low (not the lowest but close to the lowest value), we select ARIMA(1, 1, 0) as our model.

Also, we would like to look at the forecast accuracy on the testing set as supplementary tests to help us select the final models. Below are the real data versus predicted data for each model in different time periods. Note that all these three candidate models show a similar forecast accuracy on the corresponding testing set for each time periods.

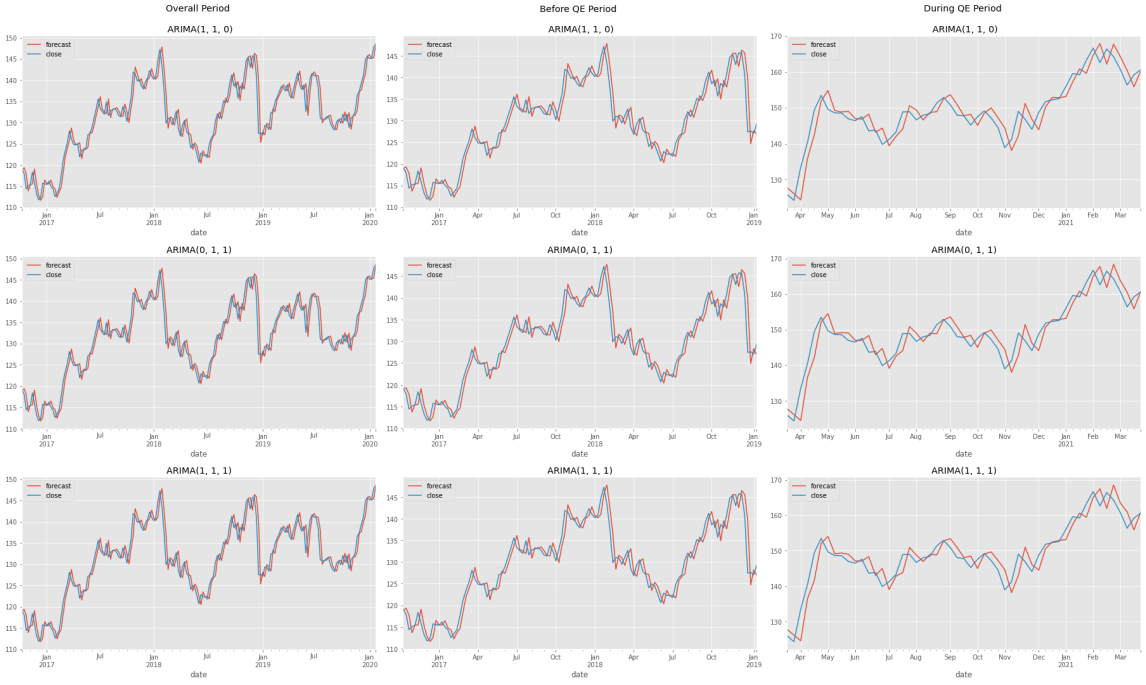

Figure 13: Real v.s. Predicted for Candidate Models

Combined with AIC, BIC and HQIC selection criteria, we choose ARIMA(1, 1, 0) as our final model for the complete data set from 2016 to 2021 and the During Q.E. Period from March, 2020 to October, 2021, and ARIMA(0, 1, 1) as our final model for the Before Q.E. Period as summarized below.

3.1.6 Model Validation

After obtaining our final model, we will implement validation test to check whether the residual for each model is a white noise or not. Below is the Normal QQ-plot of the residual for each model. As we can see from the graph, which means the residual follows a standard normal distribution, thus, is Gaussian white noise.

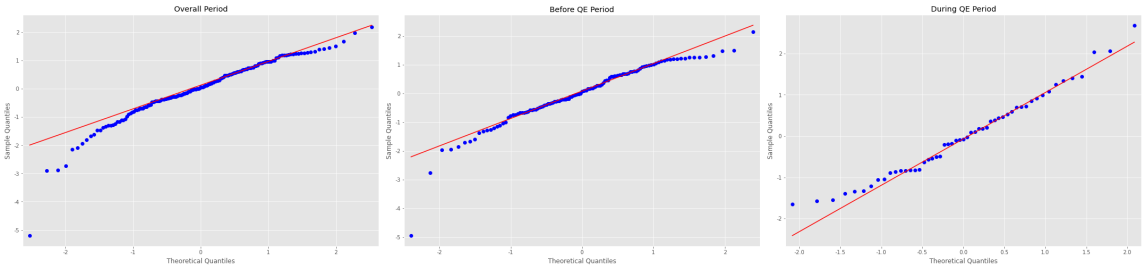

Figure 14: Normal QQ-Plot of the Residual

Alternatively, we could do Ljung-box test to the residual of each model respectively following the same pattern as we did before. As shown in the outputs below, the P-value for each Ljung-box Test of different lags are all greater than 0.05, which means we cannot reject the null hypothesis that it is a white noise. At this point, we may conclude that the information contained in the time series data has been fully extracted by our model, and our models pass the validation test.

Ljung-box Test (p-value)		
Overall Period	Before QE Period	After QE Period
0.92	0.91	0.78

Figure 15: Ljun-box Test on the residual

3.1.7 Prediction

After validating the final models, we did some predictions on the future stock price movements using different models in corresponding periods.

The following graph illustrates stock price predictions generated by the ARIMA(1, 1, 0) model based on the complete data set.

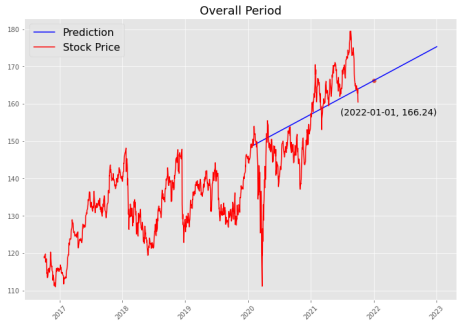

Figure 16: Prediction Based on Complete Data Set

The following graphs show the predictions of the stock price generated by the ARIMA(0, 1, 1) model based on the Before Q.E. Period data set and the predictions of the stock price generated by the ARIMA(1, 1, 0) model based on the During Q.E. Period data set.

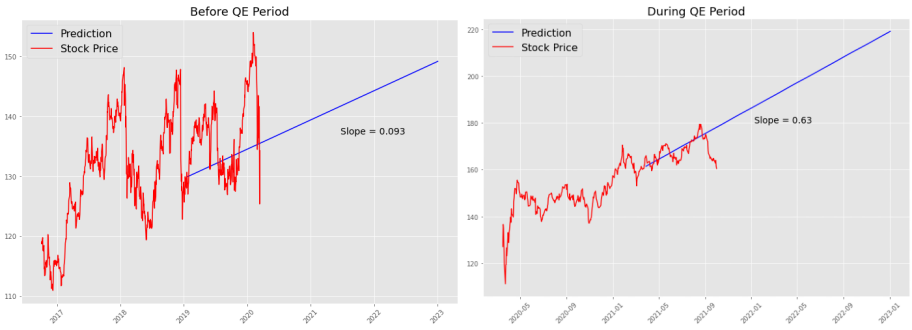

Figure 17: Prediction Based on Before and During QE Period Data Set

Difference across the predictions and inferences will be discussed in later sections.

3.2 VAR Model

3.2.1 Preliminary Data Processing

We divided the complete data set into two sub-periods, below are the sequence plots for the Before Q.E. period and the During Q.E. period. The complete time period was shown in Figure 2 in previous section. Note that we observe that the open, high, low and close prices exhibit potential serial correlation, which will be examined in the next step.

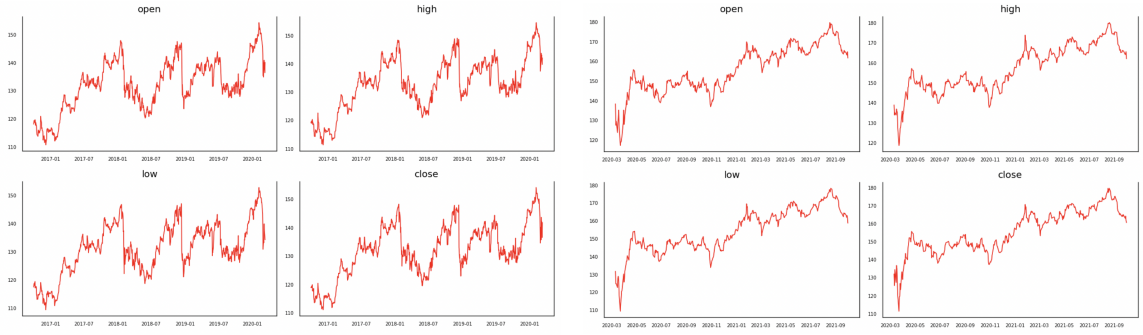

Figure 18: Sequence Plots for the Before Q.E. period (left) and the During Q.E. period (right)

3.2.2 Granger's Causality Test and Johanson Cointegration Test

As shown in the matrix below, the rows represented the response variables and the columns were the predictor series. Looking into the P-Values, we could observe that all the time series variables

in the system were interchangeably causing each other. This made the system of multi-time series a good candidate for using VAR models to forecast.

	open_x	high_x	low_x	close_x
open_y	1.0	0.0000	0.0000	0.0
high_y	0.0	1.0000	0.0000	0.0
low_y	0.0	0.0000	1.0000	0.0
close_y	0.0	0.0014	0.0002	1.0

	open_x	high_x	low_x	close_x
open_y	1.0000	0.0	0.000	0.0
high_y	0.0216	1.0	0.000	0.0
low_y	0.0000	0.0	1.000	0.0
close_y	0.1481	0.0	0.005	1.0

	open_x	high_x	low_x	close_x
open_y	1.0000	0.0	0.000	0.0
high_y	0.0216	1.0	0.000	0.0
low_y	0.0000	0.0	1.000	0.0
close_y	0.1481	0.0	0.005	1.0

Figure 19: Granger's Causality Test for the Entire Period, the Before Q.E. Period (Left) and the During Q.E. Period (Right)

Also, based on the test results for Johanson Cointegration Test, we found that each two of the open, high, low, close price have a long-run statistical relationship.

Johanson Cointegration Test			
	Overall Period	Before QE Period	After QE Period
Open VS High	2.78E-03	7.07E-03	1.17E-02
Open VS Close	2.92E-03	7.56E-03	1.16E-02
Open VS Low	1.16E-02	7.60E-03	1.72E-02

Figure 20: Johanson Cointegration Test for the Entire Period, the Before Q.E. Period and the During Q.E. Period between each two variables

3.2.3 Stationary Test

We then proceeded to the stationary test, we applied ADF tests to all time series for all periods. The test results were shown in Figure 21 below. We determined all four sets (open, high, low, close) of time series price data were stationary after 1^{st} order differencing was applied.

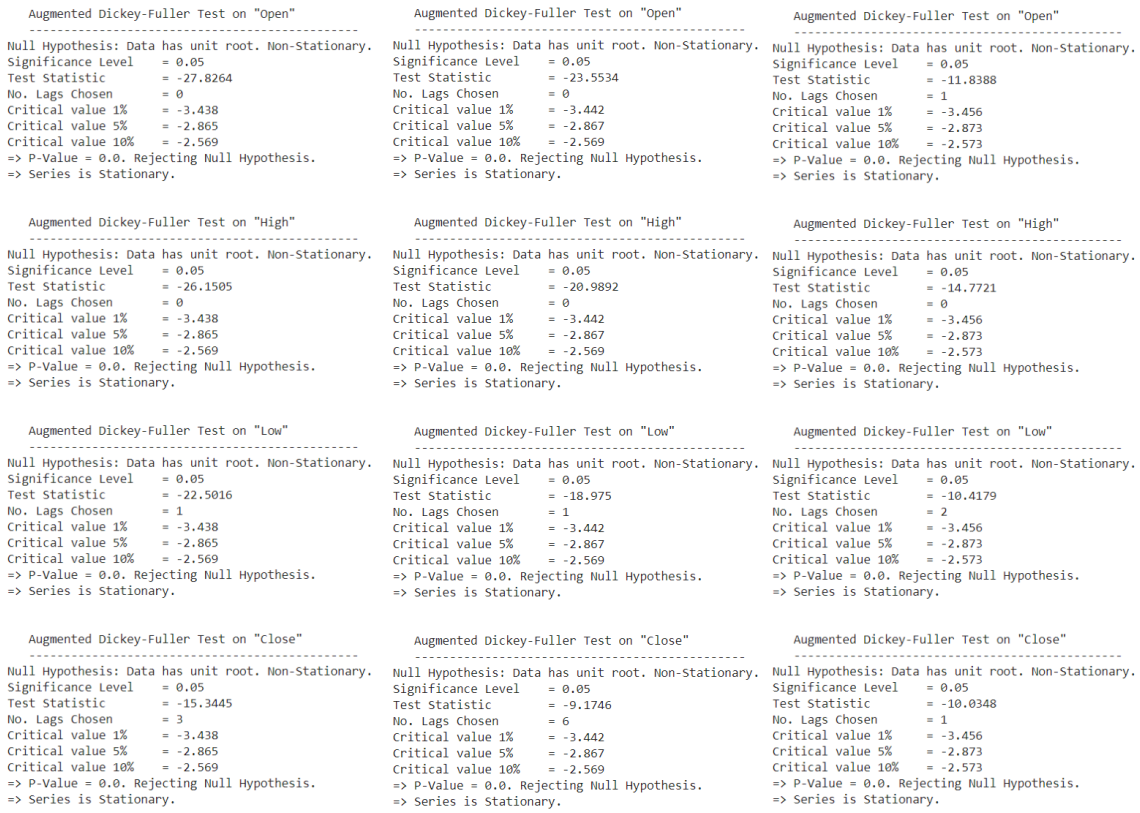

Figure 21: Stationary Tests for the Entire Period, the Before Q.E. Period and the During Q.E. Period

3.2.4 Parameter Identification

We iteratively fit increasing orders of the VAR model and picked the order that gave the least AIC or BIC or FPE or HQIC value. In our case, we determined the order that generated the minimum BIC, which provided the least number of lag order comparing to other selection criteria to prevent over-fitting. Based on the tables shown below, we determined our final models for the three periods: VAR(3) for the complete time series; VAR(3) for the before Q.E. period; and VAR(1) for the During Q.E. period.

VAR Order Selection (* highlights the minimums)

	AIC	BIC	FPE	HQIC
0	-0.01411	0.008908	0.9860	-0.005276
1	-1.639	-1.524	0.1941	-1.595
2	-2.079	-1.872	0.1250	-2.000
3	-2.359	-2.060*	0.09452	-2.244
4	-2.445	-2.054	0.08673	-2.295
5	-2.529	-2.046	0.07972	-2.344
6	-2.569	-1.994	0.07660	-2.348
7	-2.618	-1.950	0.07296	-2.362*
8	-2.651	-1.892	0.07057	-2.360
9	-2.649	-1.798	0.07071	-2.323
10	-2.654	-1.711	0.07037	-2.292
11	-2.670*	-1.635	0.06925*	-2.273
12	-2.666	-1.538	0.06960	-2.233
13	-2.657	-1.437	0.07024	-2.188
14	-2.635	-1.323	0.07181	-2.131
15	-2.617	-1.213	0.07308	-2.078
16	-2.591	-1.095	0.07501	-2.017
17	-2.581	-0.9926	0.07584	-1.971
18	-2.577	-0.8970	0.07613	-1.932
19	-2.561	-0.7891	0.07737	-1.881
20	-2.536	-0.6719	0.07937	-1.821

VAR Order Selection (* highlights the minimums)

	AIC	BIC	FPE	HQIC
0	0.1765	0.2076	1.193	0.1887
1	-1.663	-1.507	0.1896	-1.602
2	-2.082	-1.803	0.1246	-1.973
3	-2.397	-1.992*	0.09103	-2.239
4	-2.470	-1.942	0.08459	-2.264
5	-2.565	-1.912	0.07692	-2.310*
6	-2.577	-1.800	0.07599	-2.274
7	-2.626	-1.724	0.07240	-2.274
8	-2.658	-1.632	0.07015	-2.257
9	-2.648	-1.498	0.07087	-2.198
10	-2.644	-1.370	0.07114	-2.146
11	-2.663	-1.264	0.06983	-2.117
12	-2.670*	-1.146	0.06941*	-2.075
13	-2.653	-1.005	0.07061	-2.009
14	-2.623	-0.8509	0.07282	-1.931
15	-2.597	-0.7013	0.07473	-1.857
16	-2.587	-0.5662	0.07559	-1.798
17	-2.575	-0.4298	0.07657	-1.737
18	-2.570	-0.3008	0.07701	-1.684
19	-2.544	-0.1507	0.07910	-1.609
20	-2.510	0.007951	0.08196	-1.526

VAR Order Selection (* highlights the minimums)

	AIC	BIC	FPE	HQIC
0	1.456	1.514	4.289	1.479
1	0.4020	0.6904*	1.495	0.5182
2	0.1985	0.7175	1.220	0.4076
3	-0.006071*	0.7436	0.9944*	0.2959*
4	0.01421	0.9946	1.015	0.4091
5	0.07811	1.289	1.083	0.5660
6	0.08631	1.528	1.093	0.6671
7	0.1298	1.802	1.144	0.8035
8	0.04272	1.946	1.051	0.8093
9	0.05087	2.185	1.062	0.9104
10	0.01018	2.375	1.024	0.9626
11	0.02649	2.622	1.045	1.072
12	0.05714	2.883	1.083	1.195
13	0.09287	3.149	1.129	1.324
14	0.06604	3.353	1.107	1.390
15	0.04093	3.559	1.089	1.458
16	0.01938	3.768	1.076	1.529
17	0.07806	4.057	1.154	1.681
18	0.07979	4.290	1.170	1.776
19	0.1085	4.549	1.221	1.897
20	0.02081	4.692	1.137	1.903

Figure 22: VAR Model Parameter for the Entire Period, the Before Q.E. Period and the During Q.E. Period

3.2.5 Model Validation

Based on our output, our model passed the Durbin Watson's Statistics, as shown by the test statistics for open, high, low, close price in the table below. All test statistics were very close to 2, which indicated there was no significant serial correlation in residuals, which meant they were white noise. Therefore, we had captured all information and the VAR model is valid.

Durbin Watson Test			
	Overall Period	Before QE Period	After QE Period
Open	2.04	2.10	2.24
High	2.00	2.03	2.15
Low	2.01	2.04	2.09
Close	1.98	1.99	2.02

Figure 23: Durbin Watson's Statistics

3.2.6 Predictions

Figure 24 shows the prediction of the stock price generated by the VAR model, as the corresponding forecast versus actual value of open, high, low, close price for three different time periods are plotted respectively. The difference in these predictions will be discussed in later sections.

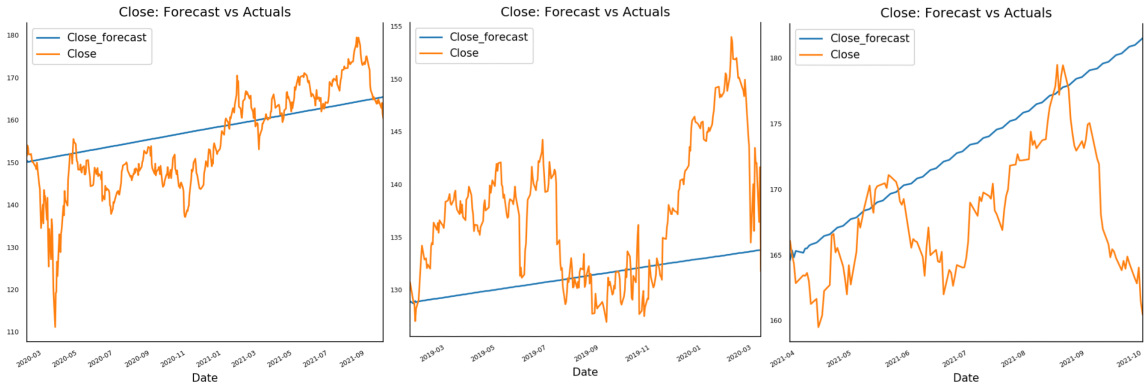

Figure 24: VAR Model Predictions for the Entire Period, the Before Q.E. Period and the During Q.E. Period

3.3 Gradient Boosted Trees

Before using the XGBoost python package gradient boosted trees model to make predictions on the close price for the Johnson & Johnson stock, the data was partitioned into a 50% training set, 30% validation set and 20% test set. The training of the model was also split into two separate categorizes, one variation of the model was trained only on the temporal features discussed in the section 2.4.1 and the other variation of the model was trained on the temporal features and the financial features in the section 2.4.3.

The financial feature variation used a sliding window with a size of seven, and so used the seven previous close price values to predict the next close price. The number of trees was chosen to be one hundred, and all the other parameters were left with their default value. These parameters were chosen as using the grid search hyperparameter tuning method always caused a positive feedback loop when the model predicted close prices using the sliding window method, leading to close price values significantly lower than the validation or test set. The sliding window size was chosen as it minimized the mean squared error on the training and validation set given the set of parameters.

The variation with only temporal features used the grid search hyperparameter tuning method to find parameters that minimized the mean square error on the training and validation set. The parameters are a gamma of 0.01, a learning rate of 0.05, a max depth of 10 and the number of decision trees as 100.

3.3.1 Feature Importance

The importance of the features used to predict the close price target variable in the gradient boosted trees model were plotted for both variations of model training. The model had the following feature importance plots.

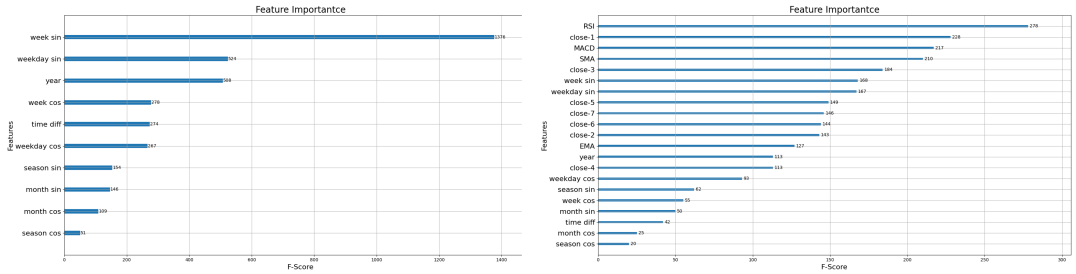

Figure 25: Feature Importance plot for the variation of the model that includes only temporal features on the left and the variation that includes both temporal and financial features on the right. The features are ranked in order of importance, with the feature contributing most to the model at the top and the feature contributing the least at the bottom.

There is difficulty when calculating the feature importance in the financial features variation due to the sliding window method as the model is trained and predicts every data point individually and so technically, every time the model is trained a new feature importance plot should be generated. However, this would lead to the same number of feature importance plots as there are data points in the validation and test set, obviously this is not possible to plot and so only the feature importance plot for the first data point has been shown. An alternative could be to plot the f-score for each data point and for each feature, however the resulting plot would likely be unreadable, and the feature importance plots are not the main subject of investigation in this project. Observing the current feature importance plot for the variation with financial features, it can be seen that the first five features are all the financial features and that for the majority of features, the temporal features are less important for calculating the close price in the model compared to the financial features. This is not necessarily surprising, given that the financial features are directly calculated from previous close price data.

In the feature importance plot with only temporal features, it can be seen that the week and weekday are the most important features when determining the close price, followed by the year. The general trend of the feature importance is that the short period temporal features are more important than the longer period temporal features, this would make sense as there is much more variation per period in the short term temporal features compared to the long term temporal features.

3.3.2 Close Price Prediction

The prediction of the future close price for the Johnson & Johnson stock was performed using the two different variations of the gradient boosted trees model that use different feature sets. The

following close prices were predicted for the validation and test set.

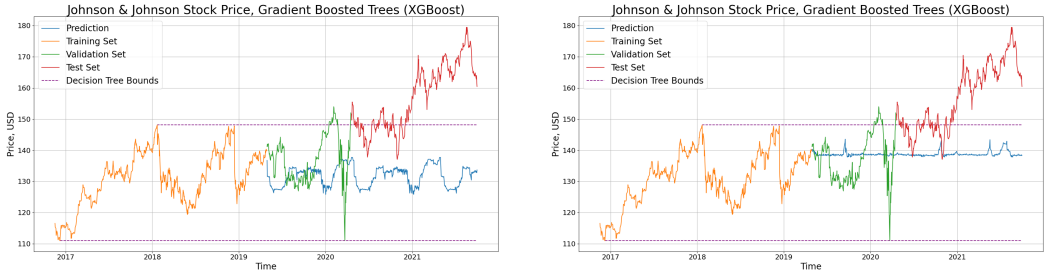

Figure 26: Close price prediction on the validation and test set for the entire Johnson & Johnson data set using only the temporal features variation on the left and the financial feature variation on the right. The actual close price values for the training, validation and test set are shown along with the gradient boosted trees model prediction. The maximum and minimum of the training set are also shown, as the model will never predict a close price outside those bounds due to the alogorithm of decision tree methods.

As can be seen in the plot of the close price prediction for the financial feature variation that the prediction is nearly a straight line with only small variations throughout the prediction period. As described in the methodology section, this effect is likely due to a negative feedback loop from the sliding window method and that causes the predictions to be constant. So, this variation of the model is not particularly powerful in making predictions about future close prices, as it will likely always choose a price near this constant line. As well, this variation is not capturing the structure and variations of the stock price seen in the validation and training set. The mean squared error for this prediction is 283.13.

For the close price prediction using the variation with only temporal features, it can be seen that this variation predicts much more structure within the close price for the validation and test set, although it is still not fitting the actual validation and test set data very accurately it provides some variation in the predicted value of the future close price. The mean squared error for this prediction is 491.10. Although this is much larger than the variation that uses financial features, the overall close price prediction structure is much more interesting.

3.3.3 Quantitative Easing

The prediction of the close price for the Johnson & Johnson stock was also performed on the data set containing only data before the start of quantitative easing on March 15, 2020 and the data set containing only data during quantitative easing. The close price for both data sets were predicted using both variations of the model.

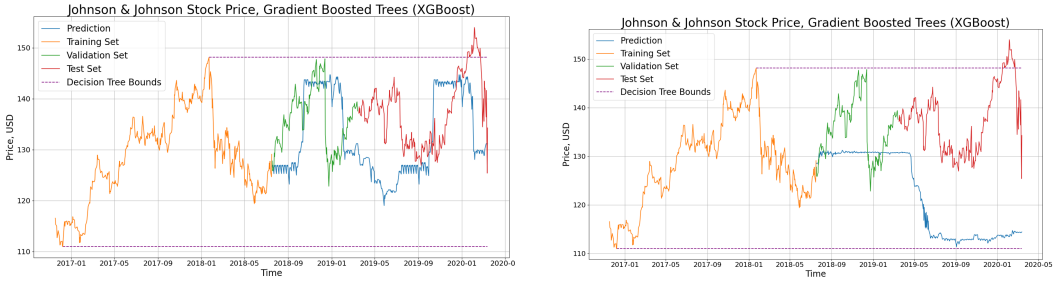

Figure 27: Close price prediction on the validation and test set for the Johnson & Johnson data set before quantitative easing, using only the temporal features variation on the left and the financial feature variation on the right. The actual close price values for the training, validation and test set are shown along with the gradient boosted trees model prediction. The maximum and minimum of the training set are also shown, as the model will never predict a close price outside those bounds due to the nature of decision tree methods.

From the plot of the prediction of the close price before quantitative easing for the variation that uses financial features it can be seen that the prediction is relatively constant until near the beginning of the test set where the price drops significantly. This is quite surprising given that the training set does not include any sharp drops, and so we would not expect the model to predict behaviour that is significantly different from the training set. As well as the fact that in the plot over the entire data set, the model was nearly constant and did not have any steep drops. The mean squared error for the prediction before quantitative easing is 338.22.

For the variation that uses only temporal features, we can see that the predicted close price fits the validation and test set quite well compared to its prediction on the entire dataset. This is not particularly surprising given that the model was hyperparameter tuned to the training set from the entire data set. So, some of the validation and test set from the data set before quantitative easing will contain the same data from the training set from the entire data set. The mean squared error for the prediction before quantitative easing is 107.26.

The prediction of the close price during quantitative easing is given below.

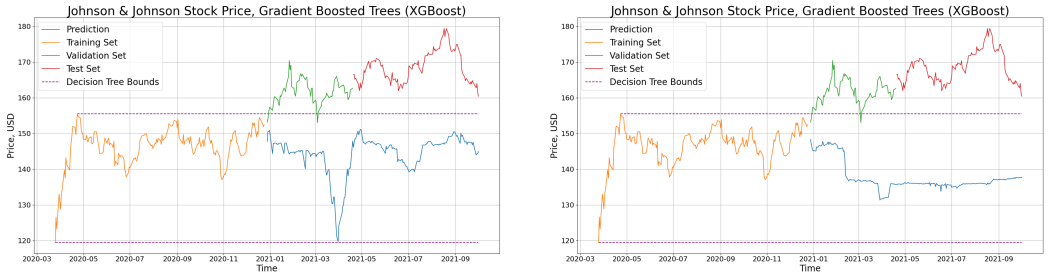

Figure 28: Close price prediction on the validation and test set for the Johnson & Johnson data set during quantitative easing, using only the temporal features variation on the left and the financial feature variation on the right. The actual close price values for the training, validation and test set are shown along with the gradient boosted trees model prediction. The maximum and minimum of the training set are also shown, as the model will never predict a close price outside those bounds due to the nature of decision tree methods.

As can be seen both variations of the model perform quite poorly, this is not unexpected given that the actual close price in the validation and test set are above the maximum value in the training set and so neither model will not be able to predict close prices above that maximum value in the training set. Focusing on the variation that uses financial features, we can see that there is a sharp drop in the close price prediction as was also seen in the data set from before quantitative easing, however this sharp drop is much less steep and does not occur near the partition of the validation and test set. The mean squared error for the prediction during quantitative easing is 847.64.

In the variation that uses only temporal features, we can see that in the close price prediction, the model attempts to replicate the sharp decrease and increase that occurs at the beginning of the training set. However, we know that this price decrease is from the introduction of the novel coronavirus and so should not be an annually repeating structure in the close price. It is interesting that the sharp drop in both variations of the model during quantitative easing occur at roughly the same location, indicating that both models are trying to predict a similar structure but have different methods of manifesting this change. In the variation that only uses temporal features, the mean squared error for the prediction during quantitative easing is 500.72.

4 Analysis and Interpretation

In this section, we will comment on our results obtained previously, analysis the output and make some inferences based on it.

4.1 Prediction of Stock Price

The prediction of the future stock price generated by three models are shown in the Part 3. Based on the complete data set, the prediction of the stock price on January 1st, 2022 is $166.24 in the ARIMA model, $164.44 in the VAR model, and $126.16 in the gradient boosted trees. The prediction generated by the gradient boosted trees is much lower than the predictions generated by

the other two models, since gradient boosted trees cannot predict values outside the decision tree boundary.

4.2 Time Series before and during Q.E.

Based on the predictions of the stock price generated by the ARIMA and VAR model based on the before Q.E. data set and during Q.E. data set, we found that the slope of the predictions of the time series before the Quantitative Easing is much lower than the slope of the predictions of the time series during the Quantitative Easing, so we can conclude that the Quantitative Easing caused that the Johnson & Johnson's stock price increased faster than before. Since the prediction of gradient boosted trees is not a straight line, so we do not use this model in this part of our analysis.

4.3 Pros and Cons of Three Models

ARIMA model is the most basic and popular time series model. It took the least execution time among three models. However, it has some disadvantages. First, the traditional model identification techniques are difficult to understand. ARIMA model requires lots of tests to find the best model, and this process is subjective. This is also a problem of the VAR model. Second, the long-term forecast of the ARIMA model eventually becomes a straight line.

VAR model can analyze multiple time series and show the correlation. For example, VAR model can be used to analyze the time series of GDP and interest rate. The VAR model can show the correlation between them. However, the VAR model is like a 'black box'. It can show the correlation among variables, but it cannot explain the relationship economically since VAR model does not refer to any economic theory framework.

Comparing with ARIMA and VAR model, the gradient boosted trees model predict subtle features, but it cannot predict values outside decision tree boundary. In addition, gradient boosted trees does not require the tests like the ARIMA and VAR models, but it requires feature engineering. With regard to the execution time, the gradient boosted trees is computational most expensive among three models.

We use Mean Square Error (MSE) to assess the forecast accuracy. Mean squared error shows the average squared difference between our prediction and the actual value, thus, the lower the MSE value, the more accurate the model fits the testing set data. The following table shows the mean squared error of our final models for the complete data set, Before Q.E. Period, and During Q.E. Period. In our case study, the MSE of the VAR model is a little bit lower than the MSE of the ARIMA model, and they are closed. But the MSE of gradient boosted trees is much higher because it cannot predict values outside the decision tree boundary, which causes more error than the other two models. The variaiton of the gradient boosted trees model that only used temporal features was used to calculate the mean squared error in the table below.

Mean Squared Error (MSE)			
	Overall Period	Before QE Period	After QE Period
ARIMA	69.47	52.27	30.24
VAR	61.65	44.96	26.81
Gradient Boosted Trees	283.13	107.26	500.73

Figure 29: Mean Squared Error for Each Time Period

5 Conclusion

We selected ARIMA(1, 1, 0), VAR(3) for the complete time series; ARIMA(0, 1, 1), VAR(3) for the before Q.E. period; and ARIMA(1, 1, 0), VAR(1)for the During Q.E. period. And we trained the gradient boosted trees by using the complete data set and the divided periods data set (before and during the Q.E. period). The parameters of gradient boosted trees model with only temporal features are a gamma of 0.01, a learning rate of 0.05, a max depth of 10 and the number of decision trees as 100.

The time series models developed based on the complete data set predict that the stock price of Johnson & Johnson will be around $165 on January 1st, 2022. The time series models developed based on divided time period data sets (before and during Q.E. periods) show that the stock price of Johnson & Johnson increases faster during the Quantitative Easing than before.

Comparing the ARIMA, VAR, and gradient boosted trees, we conclude that ARIMA and VAR perform better than gradient boosted trees in this case study because the mean squared error of ARIMA and VAR is much lower than the mean squared error of gradient boosted trees, and the execution time of ARIMA and VAR is less than the execution time of gradient boosted trees. In addition, VAR model can analyze multiple time series, but ARIMA model can only analyze a single time series.