



DIGITAL SIGNAL PROCESSING LAB MINI-PROJECT REPORT

Contributors to Project:
Sudarshan H V (EE18B034)
Yashas S V(EE18B040)

Course Number:EE2004 (DSP Lab)
Course Instructor:Dr Pooja Vyavahare

Objective:

Using the ARIMA model to learn forecasting of economic data, with SENSEX data as the example dataset.

Introduction:

Data Analysis has become an important aspect of today's world, with many approaches being introduced on a day to day basis. One such model is the fairly popular ARIMA model. In this project report, we will analyze how we can use and implement this model in MATLAB. Also, we will examine the final results of this model and draw inferences on the usability of the model as well. This project abstract is detailed in nature, mainly because of the fact that we did not find comprehensive and well-documented resources on the implementation of our project.

Literature Review:

In this section, we would like to mention the main inspiration and guiding lights for our projects. In all fairness, we have referred close to 50 resources, but will mention the main three resources here and will be listing out other notable resources in a text file in our project folder.

For theory, we have referred to the books "**Time Series**" by Peter.J.Brockwell and Richard Davis and the book "**Time Series Analysis and its Applications**" by Robert.H.Shumway. We have mainly referred to the initial two chapters in both textbooks to understand the theory behind the ARMA models.

For implementation inspiration, we have used github repositories by various users, through which we understood the implementation of the individual MATLAB functions involved in our project.

Short Theory:

Before we begin with the actual project methodology, it is important to be clear with the basic theory behind implementing the model we have chosen. The ARIMA model falls under a set of predictive analysis tools for using Time Series.

“A time series is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data.”

ARIMA, short for 'Auto Regressive Integrated Moving Average' is actually a class of models that 'explains' a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values.

A nonseasonal ARIMA model is classified as an "ARIMA(p,d,q)" model, where:

- **p** is the number of autoregressive terms,
- **d** is the number of nonseasonal differences needed for stationarity, and
- **q** is the number of lagged forecast errors in the prediction equation.

The forecasting equation is constructed as follows. First, let y denote the d^{th} difference of Y , which means:

$$\text{If } d=0: y_t = Y_t$$

$$\text{If } d=1: y_t = Y_t - Y_{t-1}$$

$$\text{If } d=2: y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$$

Note that the second difference of Y (the $d=2$ case) is not the difference from 2 periods ago. Rather, it is the *first-difference-of-the-first difference*, which is the discrete analog of a second derivative, i.e., the local acceleration of the series rather than its local trend.

In terms of y , the general forecasting equation is:

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q}$$

We need to find these coefficients in the equation. And for that, we will be making use of a variety of MATLAB command which are mentioned in the Key Commands section below.

One example below

ARIMA(0,1,0) = random walk: If the series Y is not stationary, the simplest possible model for it is a random walk model, which can be considered as a limiting case of an AR(1) model in which the autoregressive coefficient is equal to 1, i.e., a series with infinitely slow mean reversion. The prediction equation for this model can be written as:

$$\hat{Y}_t - Y_{t-1} = \mu$$

$$\hat{Y}_t = \mu + Y_{t-1}$$

...where the constant term is the average period-to-period change (i.e. the long-term drift) in Y . This model could be fitted as a *no-intercept regression model* in which the first difference of Y is the dependent variable. Since it includes (only) a nonseasonal difference and a constant term, it is classified as an "ARIMA(0,1,0) model with constant." The random-walk-*without*-drift model would be an ARIMA(0,1,0) model *without* constant

Method:

The key commands list will be given at the end of the report in the Appendix section. For the purpose of brevity, we have decided to keep the above-mentioned topic as a separate section, instead of keeping it along with the Method.

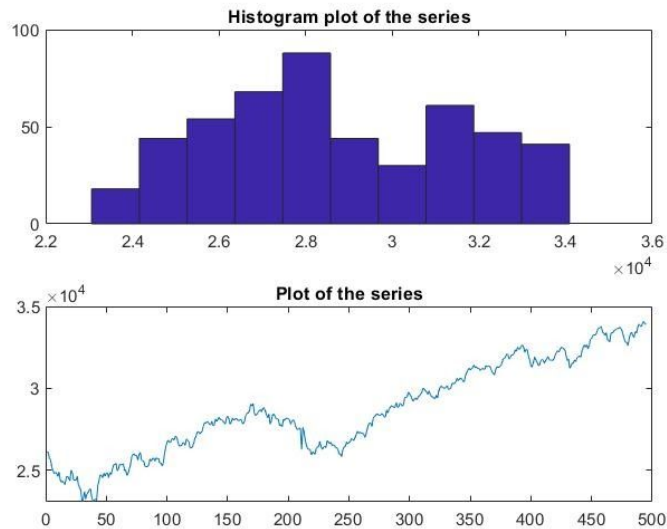
Our project mainly consists of seven phases:

- 1) Selecting a dataset and extracting the required data.
- 2) Converting the given data into a stationary time series
- 3) Fitting a theoretical ARIMA model based on the analysis.
- 4) Using AIC BIC tests, to examine all possible models and get a better fitting model, if it exists.
- 5) Predicting the data using the two models obtained previously.
- 6) Forecasting data using the two models.
- 7) Using an additional forecasting technique known as the "Monte Carlo" technique.

STEP 1:






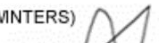






This step involves finding a dataset for SENSEX stocks and converting it into a usable format. We found our dataset after extensive searching in dataset archives, and it has been kept for reference in our Project folder. There were some missing entries in the database, which was beyond our control. So, we decided to compile the remaining data entries, to get our final dataset.

In the program, we extracted the data for the years 2016 and 2017 for training the ARIMA model. The plot of the data that was extracted is shown below.



STEP 2:

Next, we need to check whether the time series is stationary or not. A stationary time series has a constant mean and variance and has no seasonality. There are two ways to check this. The first way is to observe the plot. Various types of trends and seasonalities are shown in the image below.

	Nonseasonal	Additive Seasonal	Multiplicative Seasonal
Constant Level (SIMPLE)	 NN	 NA	 NM
Linear Trend	 LN	 LA	 LM
Damped Trend (0.95)	 DN	 DA	 DM
Exponential Trend (1.05)	 EN	 EA	 EM

The second method is to use the Augmented Dickey-Fuller Test. (ADF Test). This test returns two values, **H** and **P-value**. H is either 0 or 1, depending on whether the time series is non-stationary or stationary. P-value is a measure of stationarity of the time series. For example, if the critical p-value is set to be 5%, it means that if p-value is less than 0.05, then the chances of the time series being stationary is high. This critical p-value is set to 5% by convention, and can be set even to 1%. The results of the ADF test on original data are shown below.

In case our time series is found to have non-stationarity or seasonality, we have two methods to remove them.

1)First differencing method: Here, we take a time series, and we take the first difference. Mathematically this would be represented as $x(t)=x(t+1)-x(t)$. In this case, the process is called as first order differencing. If the same process is repeated 'd' times, it is known as d^{th} order differencing. This value 'd' is also fed as the second parameter in the arima function (the order the process 'I') also known as **degree of differencing**

2)Seasonal difference method: This process is mainly done to remove seasonality. If we find the periodicity of seasonality by observing the graph, we do the following Mathematical operation:

$x(t)=x(T+t)-x(t)$, where T is the seasonality period.

For our dataset, we have found that first order differencing is sufficient to eliminate the stationarity. The plots of the time series after the above operation is shown below, along with the ADF test results.

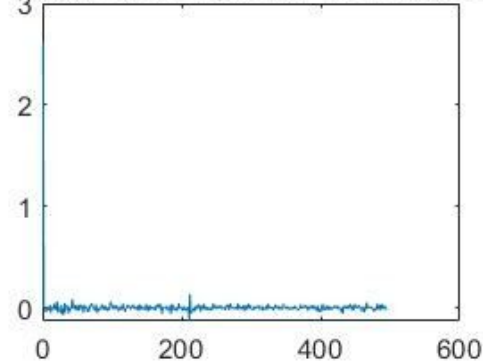
ADF TEST RESULT FOR 0 DIFFERENCING:

0 0.9698

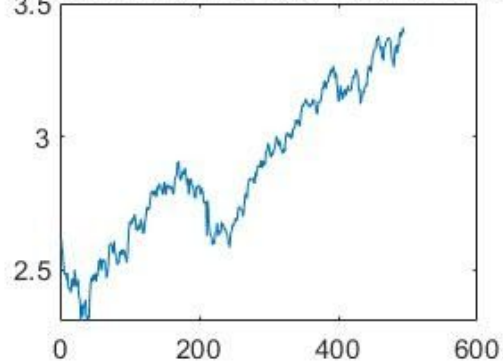
ADF TEST RESULT FOR 1 DIFFERENCING:

1.0000 0.0010

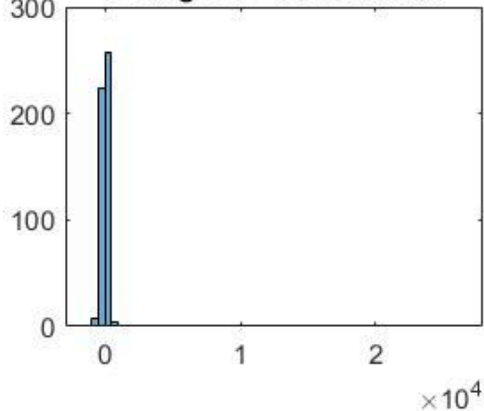
Plot of Opening data - 1 Difference



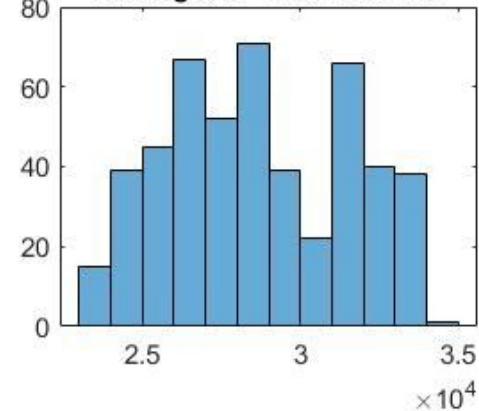
Plot of Opening data - 0 Difference



Histogram - 1 Difference



Histogram - 0 Difference



STEP 3:

The next step after attaining a stationary time series, is to estimate the order of AR and MA processes. The best way to estimate the above is by observing the ACF (Auto-Correlation Factor) and PACF (Partial AutoCorrelation Factor) plots of the time series.

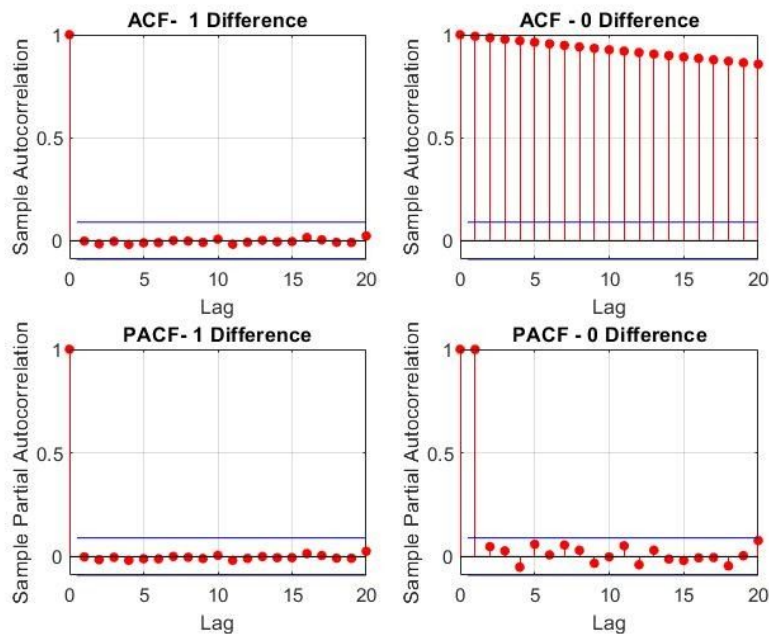
Conditional Mean Model	ACF	PACF
$AR(p)$	Tails off gradually	Cuts off after p lags
$MA(q)$	Cuts off after q lags	Tails off gradually
$ARMA(p,q)$	Tails off gradually	Tails off gradually

Using the ACF PACF plots, we can easily observe purely AR ($q=0$) or purely MA ($p=0$) processes. In case of mixed ARMA(p,q) processes, we can use an advanced method known as EACF (Extended Autocorrelation Factor) table. Basically, it is a table of X's and O's. And the left

uppermost O in the triangle of Os indicates the order of the ARMA process. An illustrative table for the ARMA(2,1) process is shown below. (Column no = 2, Row no = 1).

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	x	x	o	o	o	o	o	o	o	o	o	o	o
1	x	x	o	o	o	o	o	o	o	o	o	o	o	o
2	x	x	x	o	o	o	o	o	o	o	o	o	o	o
3	x	x	o	o	o	o	o	o	o	o	o	o	o	o
4	x	o	x	o	o	o	o	o	o	o	o	o	o	o
5	x	o	o	o	o	x	o	o	o	o	o	o	o	o
6	x	o	o	x	o	x	o	o	o	o	o	o	o	o
7	x	x	x	x	x	x	o	o	o	o	o	o	o	o

The ACF and PACF plots for our data indicated that the best predictive model would be the ARIMA(0,1,0) models. This is because after differencing, there were no significant lags in either the ACF or the PACF plots. Note that without differencing the ACF plot remains constant, which indicates non-stationarity.



STEP 4:

Even though we can predict the best-fit ARMA models using the techniques mentioned above, we still need to verify the best-fit models using AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) parameters.

AIC and BIC are both penalized-likelihood criteria. They are sometimes used for choosing best predictor subsets. The AIC or BIC for a model is usually written in the form $[-2\log L + kp]$, where L is the likelihood function, p is the number of parameters in the model, and k is 2 for AIC and $\log(n)$ for BIC.

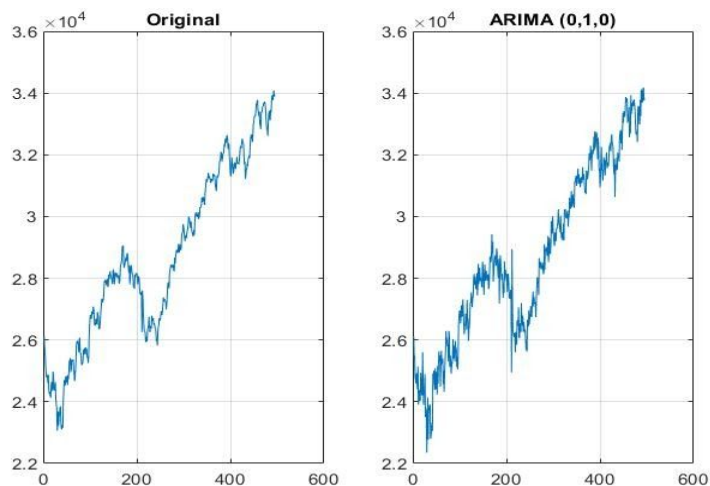
In short, the lower the AIC and BIC values, the more simple and better fitting a model is likely to be. Based on our calculations, the ARIMA(2,1,2) model is the best-fit for the data.

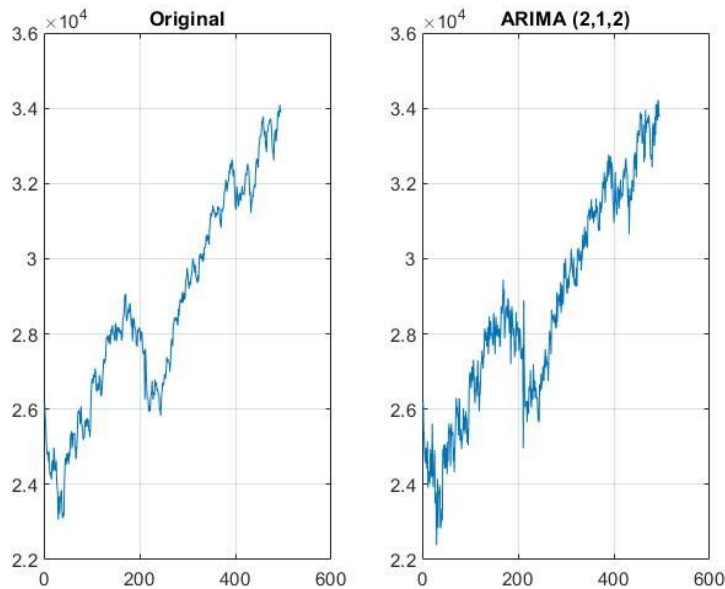
Var1	Var2	Var3
"arima(0,1,0) "	6766.5	6774.9
"arima(1,1,1) "	6764.6	6771.6
"arima(1,1,2) "	6763.3	6770.4
"arima(1,2,1) "	6764.5	6771.5
"arima(1,2,2) "	6764.4	6771.5
"arima(2,1,1) "	6763.4	6770.4
"arima(2,1,2) "	6762.2	6769.2
"arima(2,2,1) "	6764.1	6771.1
"arima(2,2,2) "	6764	6771.1

Var 2 represents AIC values and Var 3 represents BIC values.

STEP 5:

Now, we have our predicted best-fit model (from Step 3) and our practical best-fit model (from Step 4). We need to use the **infer** function in MATLAB, to predict the residuals of the time series. **Residual is nothing but the difference between the predicted and actual value.** We will predict the data for the years 2016 and 2017, as it is the training data. Then we will plot them to confirm that our models are predicting accurately compared to our original data.





STEP 6:

After setting up our models, now we feed our ARIMA models and the data to the forecast function in MATLAB, so that we can get a linear plot of the forecasted data.

We have to keep in mind that the forecasted data comes with a 95% confidence interval. This confidence interval is the interval where there is a 95% probability of the actual future data to appear. This particular interval is given by $[Y - 1.96 \cdot \text{YMSE}, Y + 1.96 \cdot \text{YMSE}]$, where Y represents the value of the forecasted data point and YMSE represents the Mean Squared Error at the point.

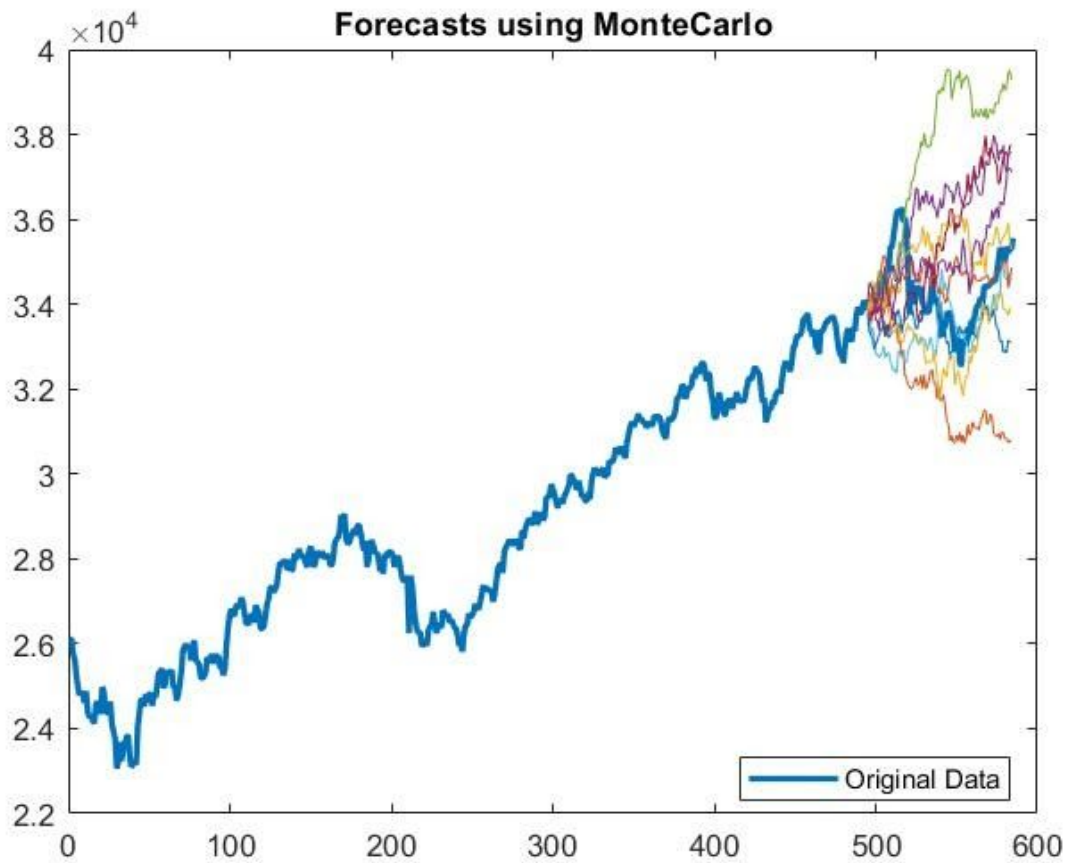
(The actual results for our dataset is shown in the results section).

STEP 7:

The last step in our project is to implement a Monte-Carlo simulation.

“Monte Carlo simulation is a technique used to understand the impact of risk and uncertainty in financial, project management, cost, and other forecasting models. A **Monte Carlo simulator** helps one visualize most or all of the potential outcomes to have a better idea regarding the risk of a decision.”

The reason why we used this simulation is to understand the various paths predictions can take and the risks associated with the predictions. Basically, this simulation takes into accounts various random factors.



Though it was not necessary to include this in our project, we felt that this function describes the fact that however accurate predictions might be, there are still chances of many factors coming into play which may not have been foreseen.

Key Commands:

In this section we have listed out the key commands that we have used to implement our project method.

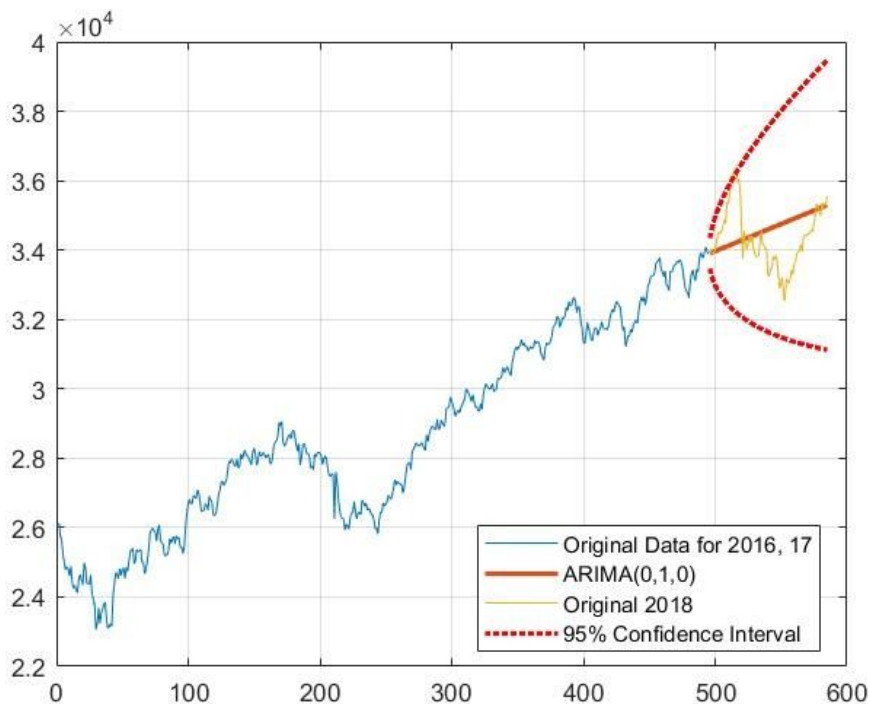
The Econometrics ToolBox has to be installed for many of the commands below

- **table2array**: Converts table data type to array data type
- **mean(x)**: This calculates the mean of the array x
- **var(x)**: This calculates the variance of the array x.
- **autocorr(data)**: Gives the autocorrelation plot of the column vector 'data'
- **parcorr(data)**: Gives the partial autocorrelation plot of the column vector 'data'
- **adftest(data)**: This function returns the h and p values after performing the Augmented Dickey-Fuller Test on the column vector 'data'

- **arima(p,d,q)**: Creates an Arima model with p^{th} order AR process, d^{th} order differencing and q^{th} order MA process.
- **estimate(Md, data)**: This function takes an ARIMA model Md and the data vector, and outputs a model which has calculated all the coefficients for estimating the ARIMA model (which was mentioned in the Theory Section).
- **infer(Md, data)**: This function takes the model output from the previous command along with data and predicts the residual and the log-likelihood of the data.
- **aicbic(Loglike, parameters, n)**: Calculates AIC and BIC model based on log-likelihood of data, number of parameters, and the number of data entries.
- **forecast(Md, n, 'Y0', data)**: Used to forecast n number of future data entries based on model Md and input 'data' vector
- **simulate(Md, n, 'NumPath', n1, 'Y0', data)**: Plots n1 number of possible plots based on a model Md and input 'data' vector.

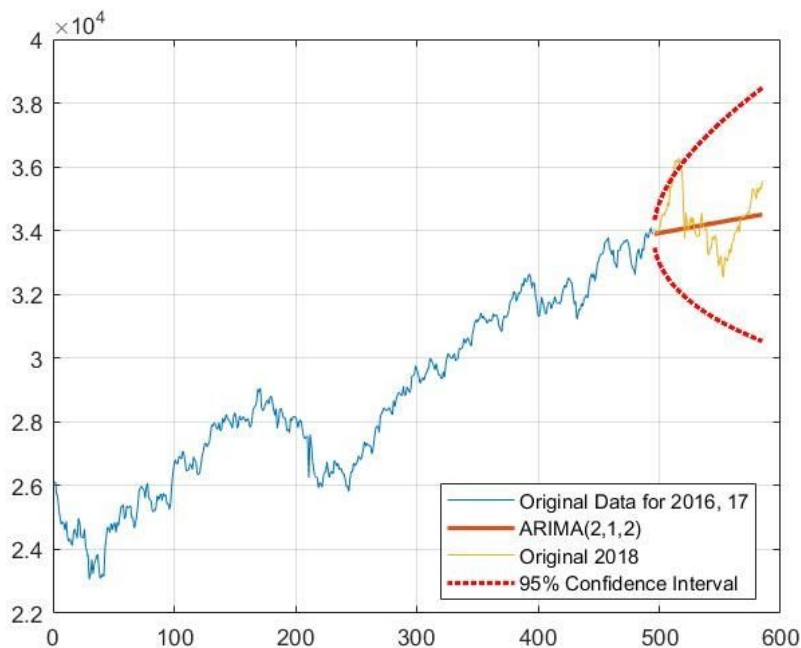
Experimental results:

These were the results of the forecasted data for the ARIMA(0,1,0) and ARIMA (2,1,2) models along with the Confidence Intervals. And we confirmed that the original data of the year 2018 was well within the forecasted confidence intervals.



In the first graph, The original data is well within the confidence interval. Also, the forecasted line pretty much fits the actual data very well. But, the important point to note is that the confidence interval widens as more data is forecasted, which is naturally correct. What the above graph tells us is that the more into the future we try to forecast, the more uncertain our predictions are.

The second graph is more interesting as the graph has gone out of the confidence interval once. So, we learn that predictive modeling along with a trial and test method works best to find out which model works best.



Conclusion:

The main conclusions that we have drawn are:

1) ARIMA model depends heavily on past data, hence prediction is fairly accurate only for a short time into the future, say around one month to three months at maximum. The confidence interval plots are indicative of the error margins.

2) There are better models in Machine Learning like Neural Networks and Support Vector Machines.

3) Most of the stock market datasets have AR and MA orders of 2 or 3 at maximum. It is unlikely to see higher orders of AR and MA.

4) First order differencing is the most effective and quickest method to remove stationarity. There are several methods of removing stationarity like curve-fitting, polynomial regression and so on, but these methods should be tried if and only if the stationarity cannot be attained by differencing.

5) ARMA models can only take stationary time series as inputs, but ARIMA models can take non-stationary inputs as well. There are other advanced techniques like ARIMAX and SARIMA which can account for various trends and multiplicative seasonalities. ARIMA model accounts for simple differencing and it is NOT possible to remove other types of complex non-stationarity.

We conclude this report by stating that the ARIMA(0,1,0) model gives the best fit for our SENSEX data based on the Confidence Interval readings obtained.