TIME SERIES ANALYSIS

ON

HCLTECH NIFTY 50 DATASET

USING ARIMA MODELLING

ABSTRACT

This project focuses on the application of time series analysis using ARIMA (AutoRegressive Integrated Moving Average) and Auto-ARIMA models to predict stock market trends within the Nifty 50 index, specifically focusing on the HCL Technologies (HCLTECH) stock. Time series analysis plays a crucial role in understanding and forecasting stock price movements, which are influenced by various economic, financial, and market-specific factors. The project aims to provide insights into the effectiveness of ARIMA and Auto-ARIMA models in capturing the underlying patterns and volatility of the stock market.

The methodology involves collecting historical stock price data for HCLTECH from the Nifty 50 index, preprocessing the data to handle missing values and anomalies, and transforming it into a suitable format for time series analysis. The ARIMA model, a classical time series model, will be implemented to capture the autoregressive and moving average components of the data. Additionally, the Auto-ARIMA model, which automates the process of identifying optimal model parameters, will be employed to enhance the accuracy of predictions.

The project will focus on evaluating the performance of the ARIMA and Auto-ARIMA models by conducting thorough model validation and comparison. Various metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) will be utilized to quantify the accuracy of the predictions. The project will also explore the models' ability to capture different aspects of stock price movements, including short-term fluctuations and long-term trends.

Through this project, we aim to provide valuable insights into the application of time series analysis in the context of stock market prediction. The results obtained from the ARIMA and Auto-ARIMA models will offer a basis for understanding their effectiveness in capturing the complexities of the Nifty 50 index, with a specific focus on HCLTECH stock. This project's outcomes can serve as a foundation for further research in utilizing advanced forecasting techniques and models to enhance stock market predictions and inform investment decisions.

ABOUT THE DATASET

The dataset contains the price history and trading volumes of the fifty stocks in the index NIFTY 50 from NSE (National Stock Exchange) India. All datasets are at a day-level with pricing and trading values split across .cvs files for each stock along with a metadata file with some macro-information about the stocks itself. The data spans from 1st January, 2000 to 30th April, 2021.

Some of the relevant columns of the dataset are:

1. **Date**: This is the datetime column.
2. **Symbol**: This column contains the Company name which is HCLTECH in the scenario.
3. **Series**: This column is described as the type of security provided which is 'EQ' in this scenario.
4. **Prev Close**: This column is described as the Previous Day's Closing Price. **This column is also our target column in the dataset.**
5. **Open**: This column is described as the Open Price of the Day.
6. **High**: This column is described as the Highest Price in Day.
7. **Low**: This column is described as the Lowest Price in Day.
8. **Last**: This column is described as the Last Traded Price in Day.
9. **VWAP**: This column is described as the Volume Weighted Average Price.

Thus, these are some of the relevant Columns in our dataset.

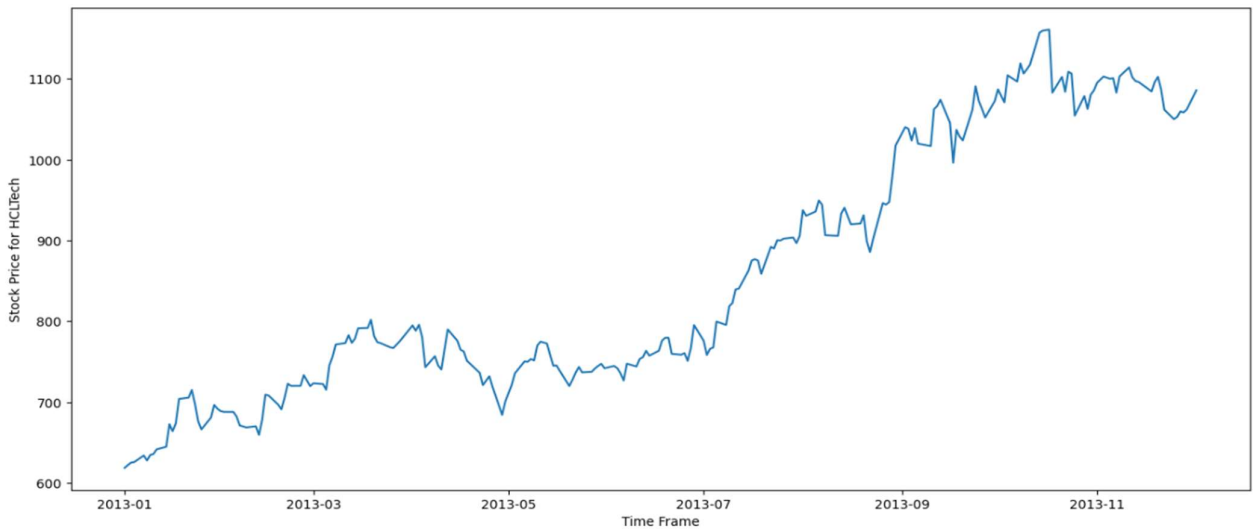
SOFTWARES USED

1. Jupyter Notebook
2. Python

SPECIAL PACKAGES USED

1. **Pandas**: Used for Data Manipulation and Analysis.
2. **Numpy**: For Numerical Operations.
3. **Seaborn and Matplotlib**: For Data Visualizations.
4. **Seasonal_decompose from Statsmodels**: Used to Decompose the Time Series into Seasonal,Trend, etc., components.
5. **adfuller from Statsmodels**: Used to perform Augmented Dickey Fuller Test.
6. **acf, pacf from Statsmodels**: Used for obtaining Auto-Correlation and Partial Auto-Correlation charts.
7. **ARIMA from Statsmodels**: Used to perform ARIMA Modelling in the Time Series Data.
8. **pmdarima**: Used to perform Auto-ARIMA to get the best parameters for ARIMA Modelling.

A Specific part(truncated) of the Time Series data I have worked on is represented in graphical format as follows:



INTRODUCTION TO TIME-SERIES ANALYSIS

A **Time-Series data** is a series of data points or observations recorded at different or regular time intervals. In general, a time series is a sequence of data points taken at equally spaced time intervals. The frequency of recorded data points may be hourly, daily, weekly, monthly, quarterly or annually.

Time-Series Forecasting is the process of using a statistical model to predict future values of a time-series based on past results.

A time series analysis encompasses statistical methods for analyzing time series data. These methods enable us to extract meaningful statistics, patterns and other characteristics of the data. Time series are visualized with the help of line charts. So, time series analysis involves understanding inherent aspects of the time series data so that we can create meaningful and accurate forecasts.

Applications of time series are used in statistics, finance or business applications. A very common example of time series data is the daily closing value of the stock index like NASDAQ or Dow Jones. Other common applications of time series are sales and demand forecasting, weather forecasting, econometrics, signal processing, pattern recognition and earthquake prediction.

COMPONENTS OF A TIME-SERIES

1. **Trend** - The trend shows a general direction of the time series data over a long period of time. A trend can be increasing(upward), decreasing(downward), or horizontal(stationary).

2. **Seasonality** - The seasonality component exhibits a trend that repeats with respect to timing, direction, and magnitude. Some examples include an increase in water consumption in summer due to hot weather conditions.

3. **Cyclical Component** - These are the trends with no set repetition over a particular period of time. A cycle refers to the period of ups and downs, booms and slums of a time series, mostly observed in business cycles.

These cycles do not exhibit a seasonal variation but generally occur over a time period of 3 to 12 years depending on the nature of the time series.

4. **Irregular Variation** - These are the fluctuations in the time series data which become evident when trend and cyclical variations are removed. These variations are unpredictable, erratic, and may or may not be random.

5. **ETS Decomposition** - ETS Decomposition is used to separate different components of a time series. The term ETS stands for Error, Trend and Seasonality.

PATTERNS IN A TIME SERIES

Any time series visualization may consist of the following components:

Base Level + **Trend** + **Seasonality** + **Error**

Trend: A **trend** is observed when there is an increasing or decreasing slope observed in the time series.

Seasonality: A **seasonality** is observed when there is a distinct repeated pattern observed between regular intervals due to seasonal factors. It could be because of the month of the year, the day of the month, weekdays or even time of the day.

However, it is not mandatory that all time series must have a trend and/or seasonality. A time series may not have a distinct trend but have a seasonality and vice-versa.

Cyclic behaviour: Another important thing to consider is the **cyclic behaviour**. It happens when the rise and fall pattern in the series does not happen in fixed calendar-based intervals. We should not confuse 'cyclic' effect with 'seasonal' effect. If the patterns are not of fixed calendar based frequencies, then it is cyclic. Because, unlike the seasonality, cyclic effects are typically influenced by the business and other socio-economic factors.

ADDITIVE AND MULTIPLICATIVE TIME SERIES

We may have different combinations of trends and seasonality. Depending on the nature of the trends and seasonality, a time series can be modeled as an additive or multiplicative time series. Each observation in the series can be expressed as either a sum or a product of the components.

Additive Time Series:

Value = Base Level + Trend + Seasonality + Error

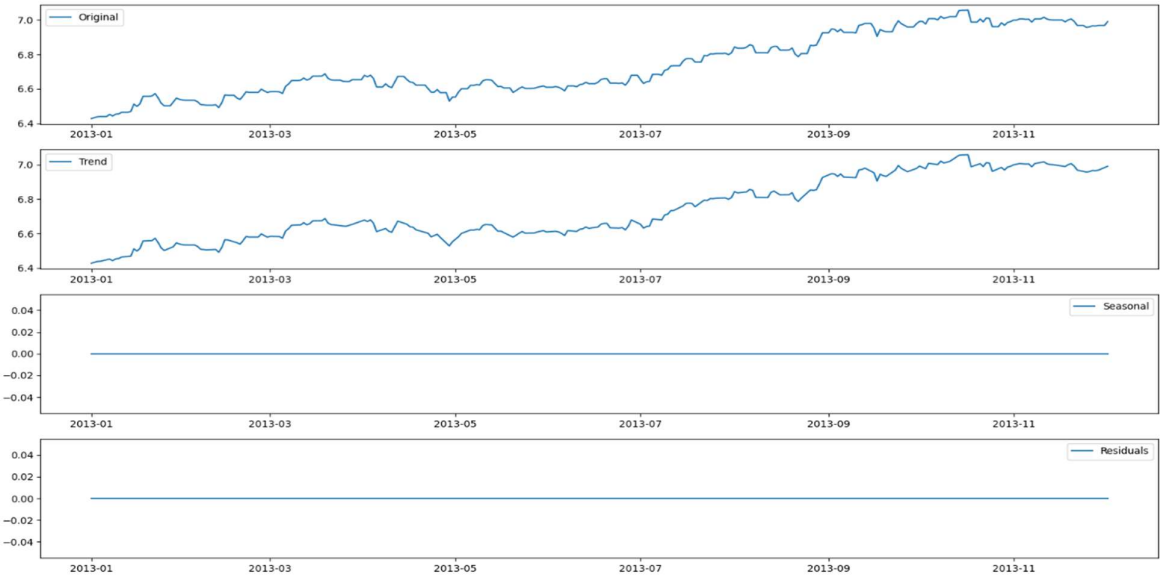
Multiplicative Time Series:

Value = Base Level x Trend x Seasonality x Error

DECOMPOSITION OF A TIME SERIES

Decomposition of a time series can be performed by considering the series as an additive or multiplicative combination of the base level, trend, seasonal index and the residual term.

The seasonal_decompose in statsmodels implements this conveniently.



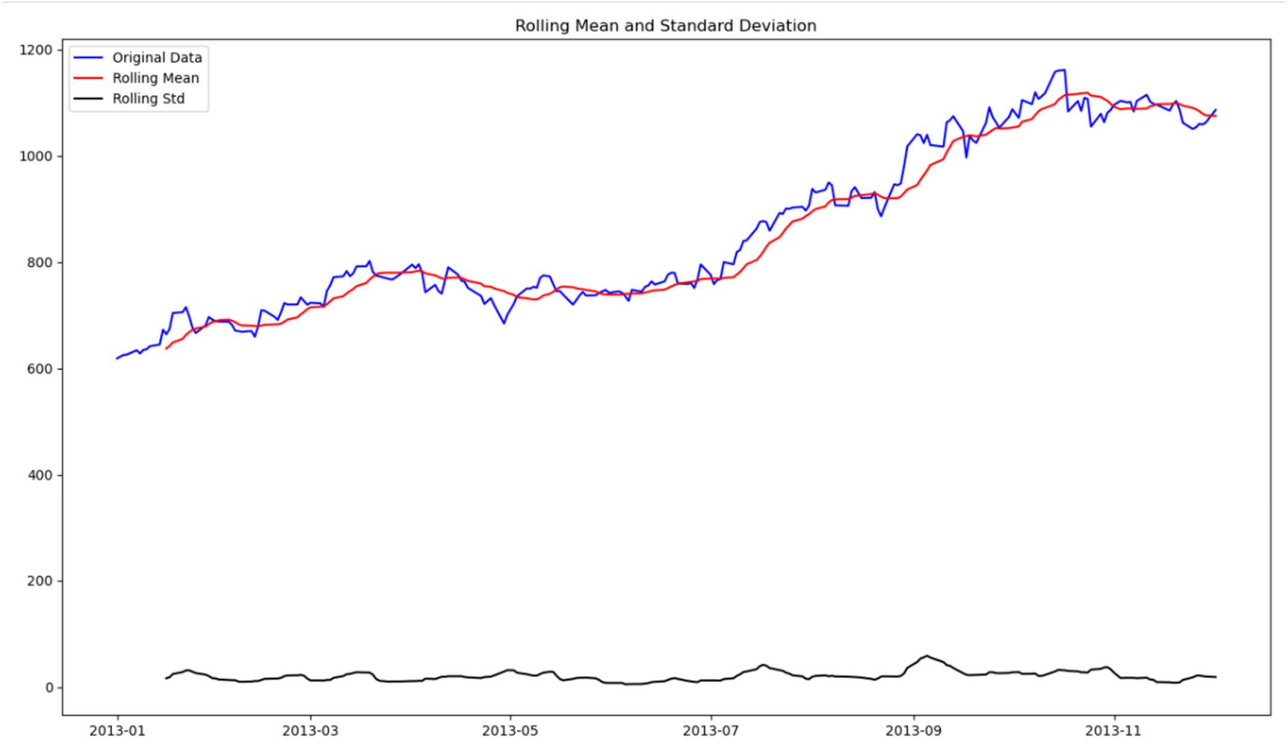

```
Out[69]: (230,)
```

Here, there is a decomposed truncated Time Series which we have obtained after decomposition.

STATIONARY AND NON-STATIONARY TIME SERIES

Stationarity is a property of a time series. A **stationary series** is one where the values of the series is not a function of time. So, the values are independent of time.Hence the statistical properties of the series like mean, variance and autocorrelation are constant over time. Autocorrelation of the series is nothing but the correlation of the series with its previous values.

A stationary time series is independent of seasonal effects as well.



This is the obtained Time Series of the specific truncated segment. It is clear that the Time Series is not Stationary because the Rolling Mean is not constant over the time interval.

So we need to make the Time Series stationary in order to employ a model.

Thus we take the log values of the data we are provided and apply differencing on the data in order to get a differenced Time Series to obtain Stationarity.

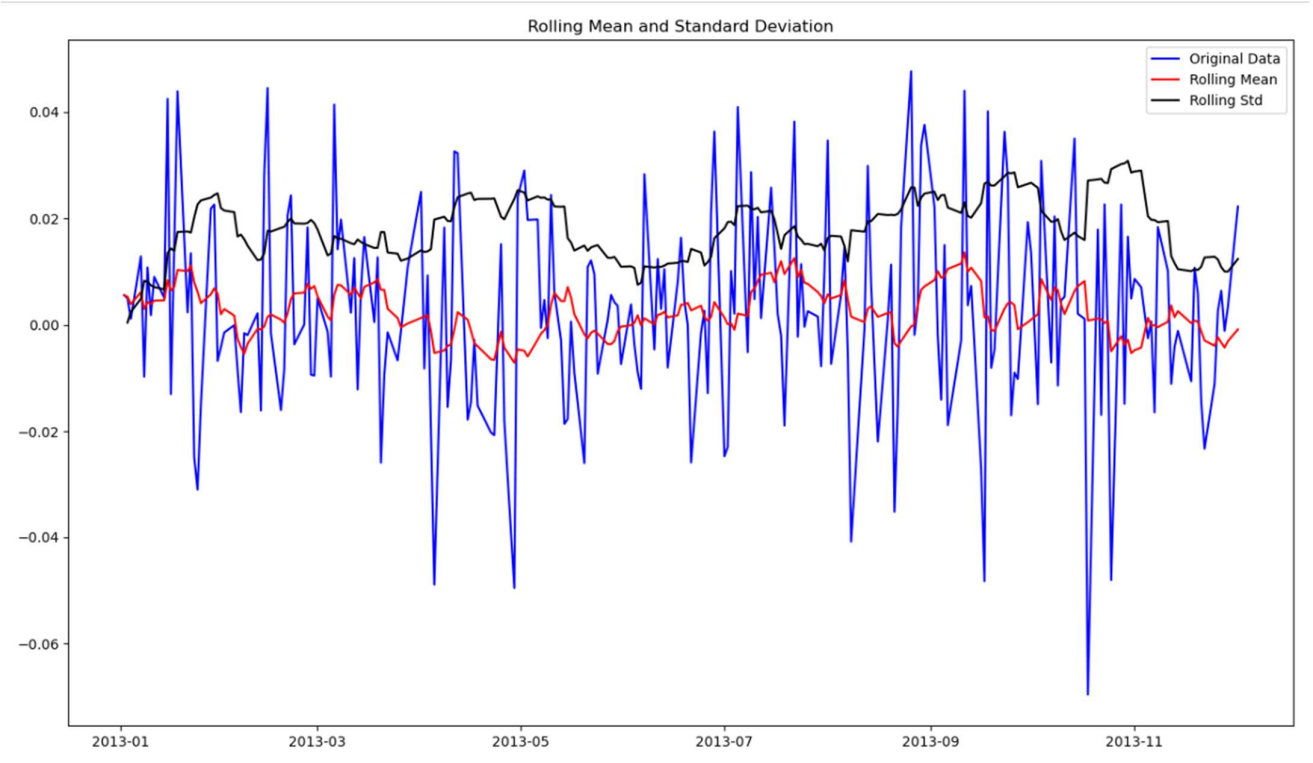
Reasons to convert a non-stationary series into stationary one before forecasting:

There are reasons why we want to convert a non-stationary series into a stationary one. These are given below:

1. Forecasting a stationary series is relatively easy and the forecasts are more reliable.

2. An important reason is, autoregressive forecasting models are essentially linear regression models that utilize the lag(s) of the series itself as predictors.

3. We know that linear regression works best if the predictors (X variables) are not correlated against each other. So, stationarizing the series solves this problem since it removes any persistent autocorrelation, thereby making the predictors(lags of the series) in the forecasting models nearly independent.



Thus, this is how it looked after differencing the log Time Series.

The Time Series is surely Stationary.

AUTOCORRELATION AND PARTIAL AUTOCORRELATION FUNCTIONS

Autocorrelation is simply the correlation of a series with its own lags. If a series is significantly autocorrelated, that means, the previous values of the series (lags) may be helpful in predicting the current value.

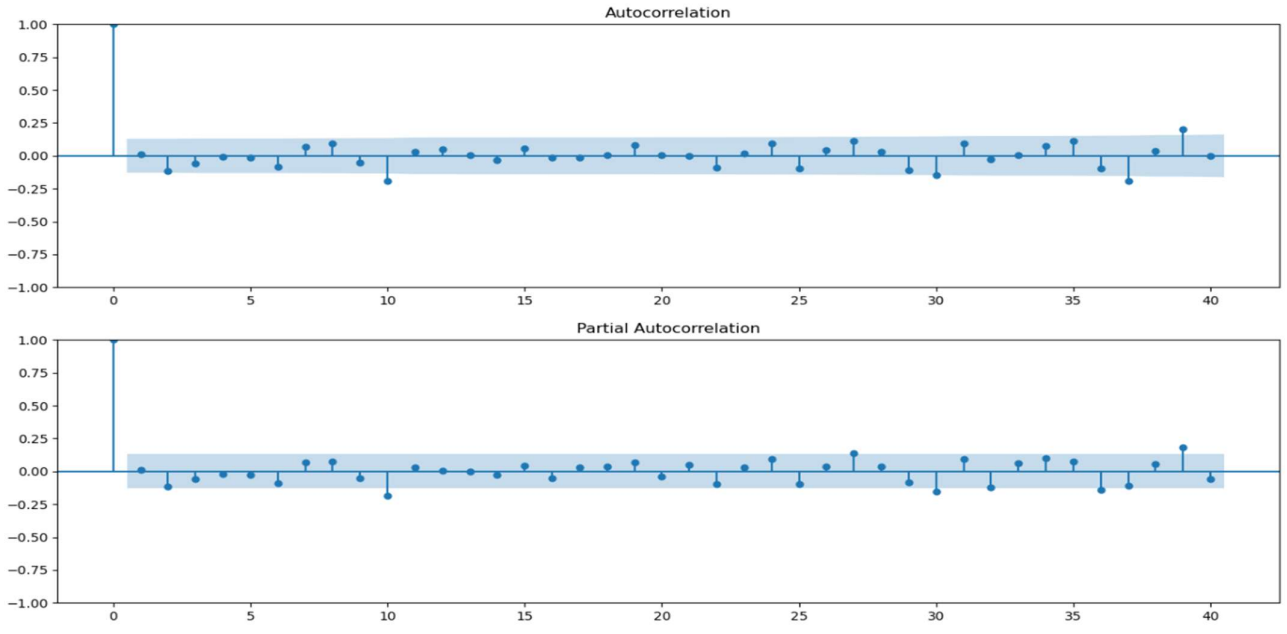
Partial Autocorrelation also conveys similar information but it conveys the pure correlation of a series and its lag, excluding the correlation contributions from the intermediate lags.

Computation of Partial Autocorrelation Function:

The partial autocorrelation function of lag (k) of a series is the coefficient of that lag in the autoregression equation of Y. The autoregressive equation of Y is nothing but the linear regression of Y with its own lags as predictors.

For example, if Y_t is the current series and Y_t-1 is the lag 1 of Y, then the partial autocorrelation of lag 3 (Y_t-3) is the coefficient α3 of Y_t-3 in the following equation:

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \alpha_3 Y_{t-3}$$



Thus these are the Auto-Correlation and Partial Auto-Correlation plots.

ARIMA MODEL

The ARIMA model (an acronym for Auto-Regressive Integrated Moving Average), essentially creates a linear equation which describes and forecasts your time series data. This equation is generated through three separate parts which can be described as:

- **AR** — Auto-Regression: Equation terms created based on past data points
- **I** — Integration or differencing: accounting for overall "trend" in the data
- **MA** — Moving Average: Equation terms of error or noise based on past data points

Together, these three parts make up the AR-I-MA model.

The AR and MA aspects of ARIMA actually come from standalone models that can describe trends of more simplified time series data. With ARIMA modeling, we essentially have the power to use a combination of these two models along with differencing (the "I") to allow for simple or complex time series analysis.

DETAILS OF ARIMA MODEL

The ARIMA model is almost always represented as ARIMA(p, d, q) where each of the letters corresponds to one of the three parts described above. These three letters represent parameters that you will have to provide, and are described as follows:

- **p** determines the number of **autoregressive (AR) terms**
- **d** determines the order of **differencing**
- **q** determines the number of **moving average (MA) terms**

Auto-Regressive and Moving Average parts

The ARIMA model is recursive in nature and thus relies on past calculations. This recursive nature comes directly from the AR and MA equation terms that are added to the model.

The p value, or AR part, essentially describes how reliant our data points are on past data points. If p=1 then the model's output for a specific time relies directly on what the output was for the time before. If p=2, then the output would rely on the outputs from the last two time periods.

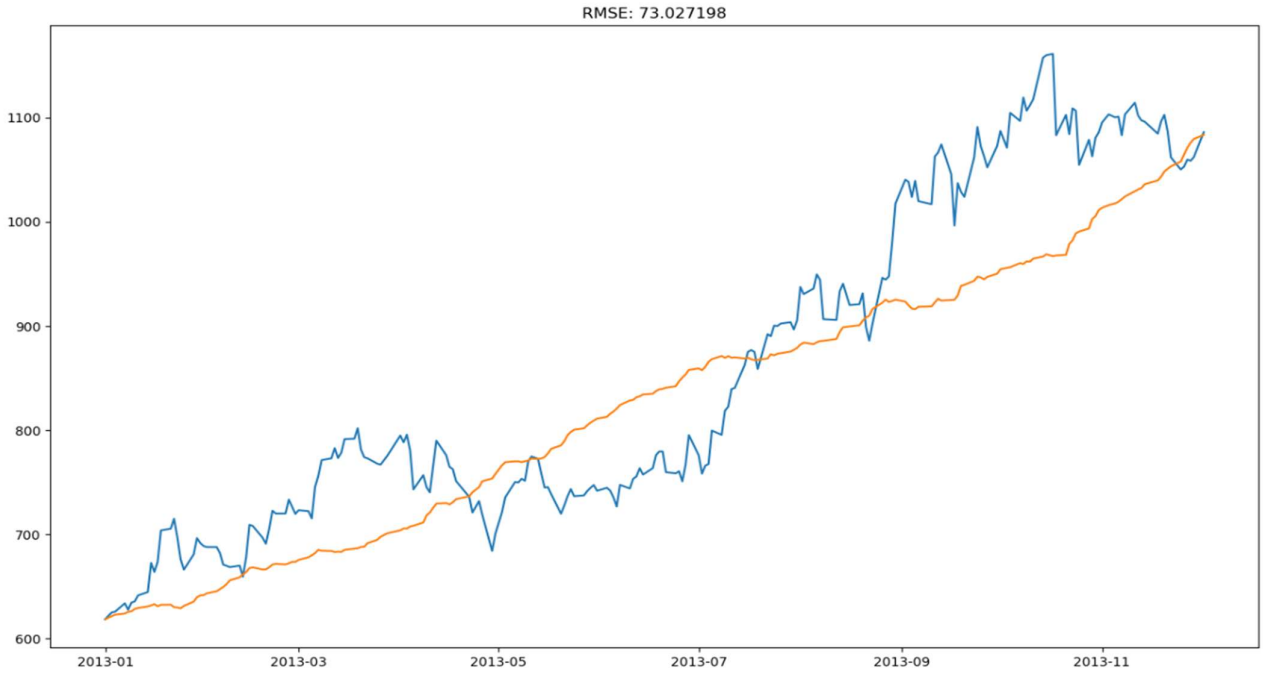
Similarly, the q value, or MA part, uses the same recursive concept. The difference is that q describes how related your current output is to its past error or noise calculations. So, if q=1, then your current output would rely on the past time period's noise calculation. For q=2, our output would rely on the noise from the last two time periods.

Integration part

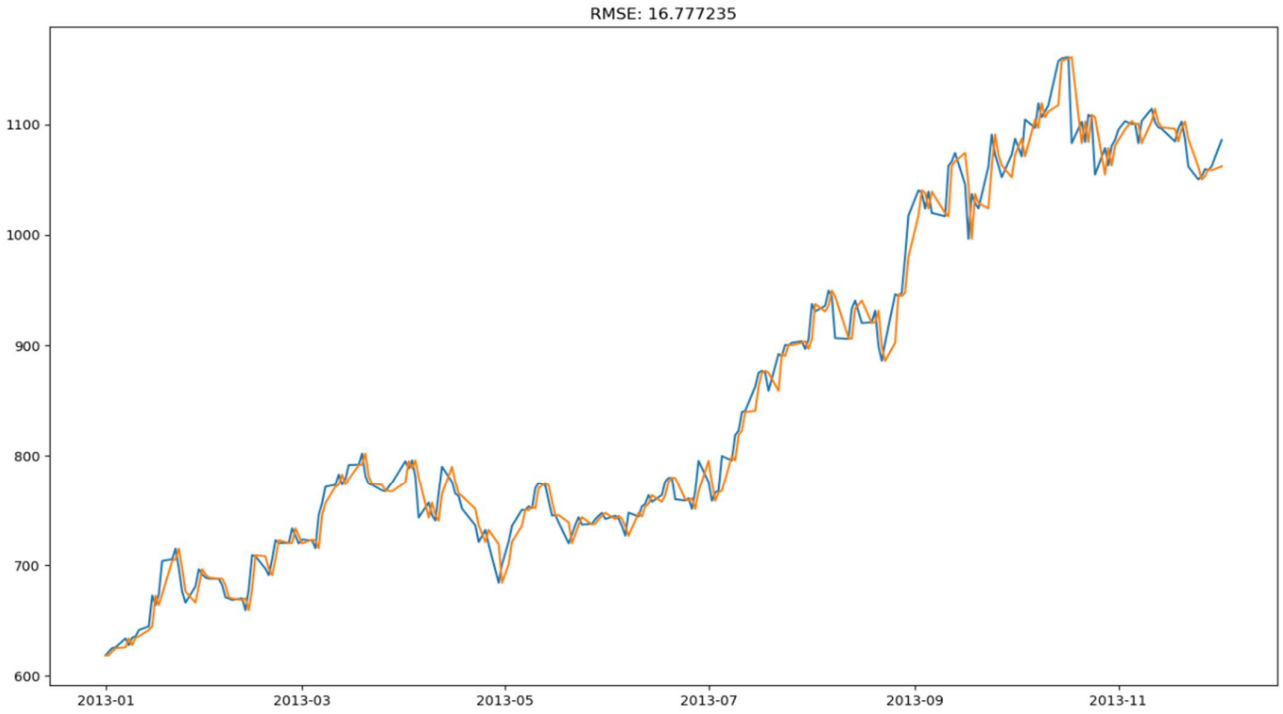
This part of the model accounts for general trends that occur throughout the time series data. The d value refers to how many times we would need to take the derivative of our time series trend to get a flat line (or constant).

If we were analyzing this data with an ARIMA model, we would likely use d=1 to account for its linear trend. If the trend were quadratic, we would probably have to use d=2.

Now, after applying ARIMA to our Time Series, we have obtained this graph which explains the data quite well and suitable for further forecasting.



But, there is still scope of developing our model using Hyperparameter tuning by employing Auto-ARIMA into our Time Series.



Thus, Auto-ARIMA has helped us to get even a better model by tuning the parameters of our ARIMA model to suitable values.

RESULTS AND CONCLUSIONS

- From Acf and Pacf plots, we got the optimum values for p,d,q which were used in the ARIMA model as parameters.

- We used log-transformation and did a differencing which made our Time Series stationary.

- We did a Hypothesis Testing of the stationarity of our Time Series to cross-validate the stationarity.

```
Test Statistic: -11.814335558344519
p-value: 8.70994137094202e-22
The Time Series is Stationary.
```

- Our ARIMA model with (p,d,q) values as (2,0,2) where p and q values are obtained from acf and pacf plots and d value being the lag value, gave an accuracy of 78.13%.

- Auto-ARIMA model gave us the best values of p,d and q which are 0,1 and 0 respectively.

```
Performing stepwise search to minimize aic
 ARIMA(3,1,3)(0,0,0)[0] intercept   : AIC=-1158.020, Time=0.35 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=-1167.424, Time=0.03 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=-1165.456, Time=0.03 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=-1165.465, Time=0.08 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=-1165.525, Time=0.02 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=-1163.485, Time=0.11 sec

Best model:  ARIMA(0,1,0)(0,0,0)[0] intercept
Total fit time: 0.643 seconds

Out[100]: ARIMA(order=(0, 1, 0), scoring_args={}, suppress_warnings=True)
```

- Auto-ARIMA helped us to find the best model which perfectly captures the trend and unexpected fluctuations also, which was not obtained during simple ARIMA as simple ARIMA model could not explain those unexpected fluctuations as seen in the graphs.

- Finally, with the help of Auto-ARIMA, our Time Series model explains 98.8% of the variation in the data and is perfect for forecasting.

THANK YOU