Time Series Analysis of U.S. Air Carrier Traffic Dataset

Prepared by:

Shakti Singh Rathore

Bablu Prajapti

Problem Statement:

Analyze the Data of United States Air Carrier Traffic and Forecast the Total number of miles traveled by paying passengers (Revenue Passenger Mile) For next 12 Months.

Dataset Description:

The dataset used in this project is publically available and was created by U.S. from January 2000 to February 2020 based on various criteria like Geographic Area, Schedule Type,ServiceClass(Passenger or Cargo) Operating Statistics(RPM or RTM). We have two feature in the dataset year and Total RPM so based on this data we have to forecast the total value of RPM of upcoming 12 Months.

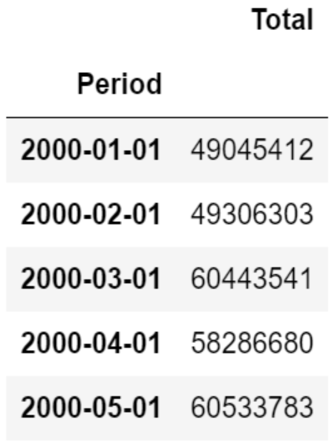
Feature Information:

Period: Indicate the Time Period of Dataset

Total: indicate the Total RPM of the period.

Analysis:

We use python language for analysis of our data. First Load the Data in Jupyter note book so we can See first glance of dataset .



Period	Total
2000-01-01	49045412
2000-02-01	49306303
2000-03-01	60443541
2000-04-01	58286680
2000-05-01	60533783

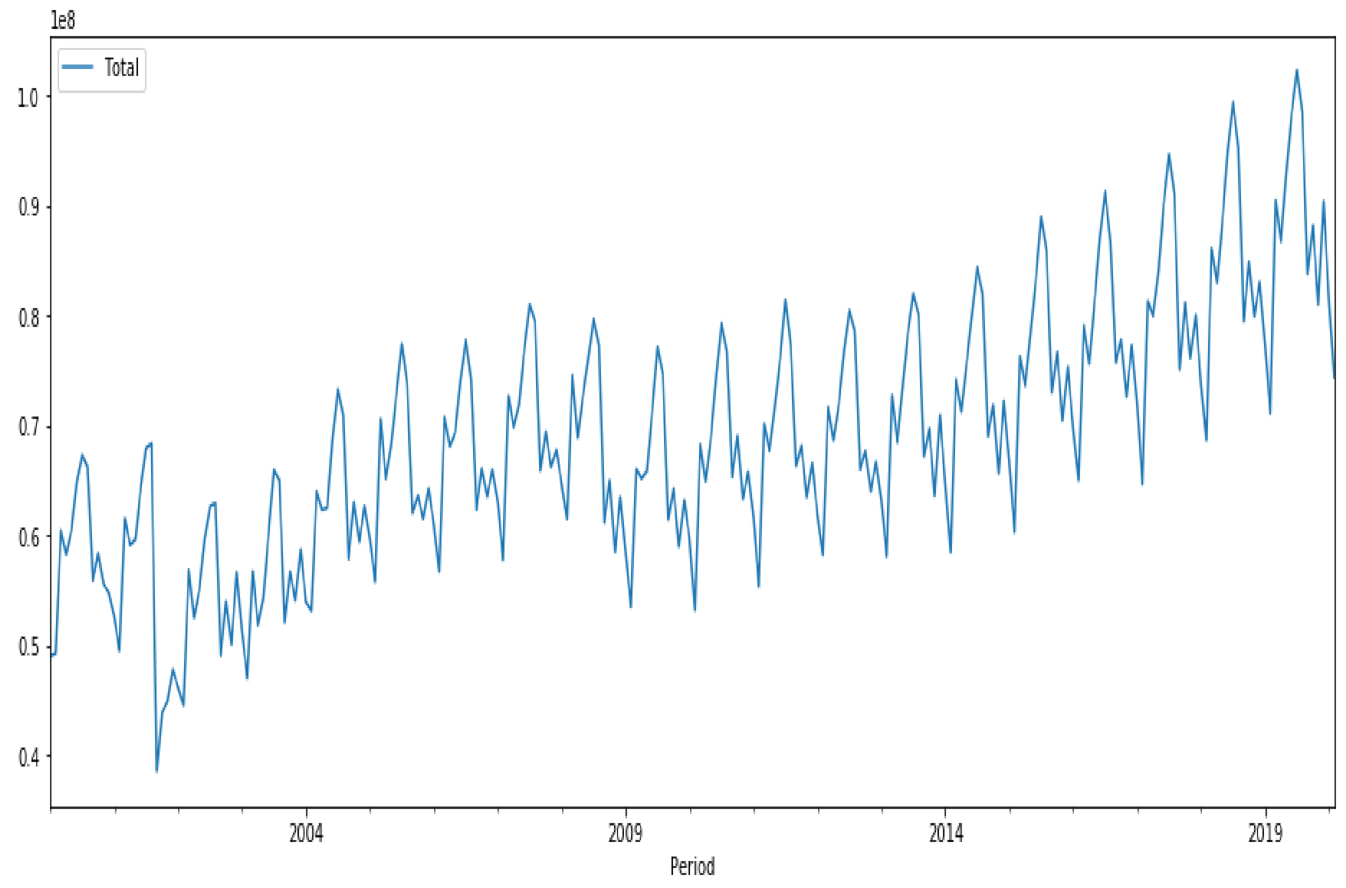
So we can see time period and total RPM shown with respect to each other.

Visualization:

Visualization of data is an imperative aspect of data science. It helps to understand data and also to explain the data to another person.

Line Chart:

By looking at the line chart we can say that seasonality and little bit of trend present in the data.



Pre-Processing of the Data:

Checking For Stationary:

•Until unless your time series is stationary, you cannot build a time series model.

Testing For Stationary

Using ADF Test

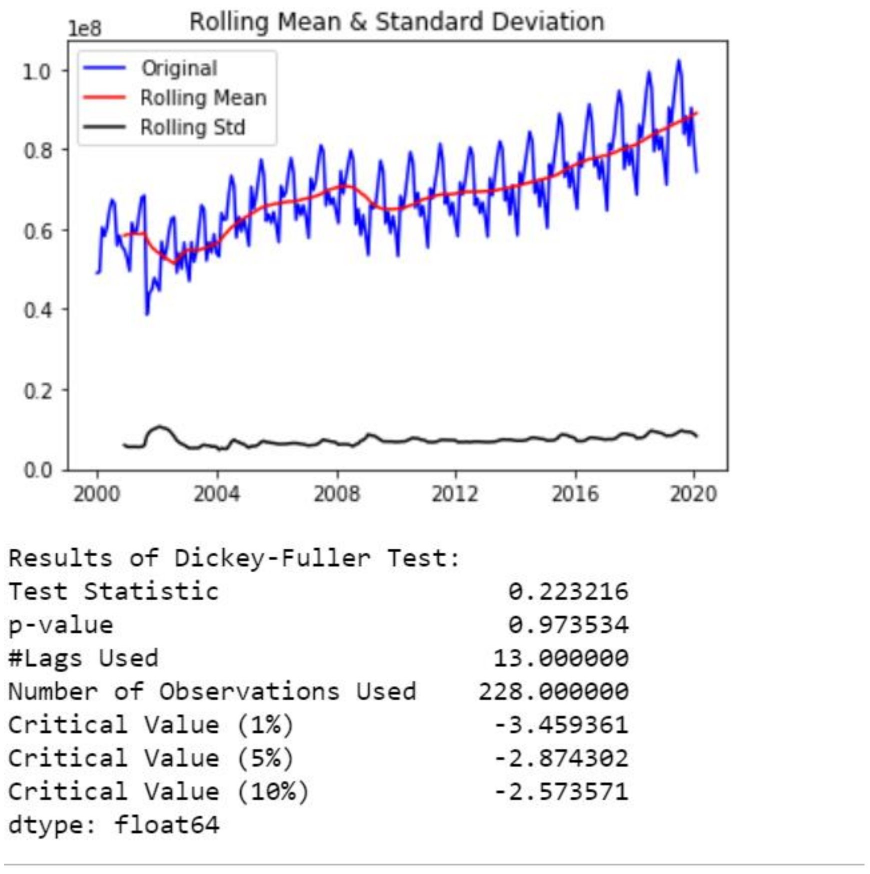
(Smaller p-values suggest that data is stationary)

H0: series is non-stationary

H1: series is stationary

If Data is non-stationary our first goal is to make data Stationary, data with trend and seasonality are always non-stationary data On the other hand, time series with white noise and with cyclic behavior will be considered as stationary.

Using ADF Test:

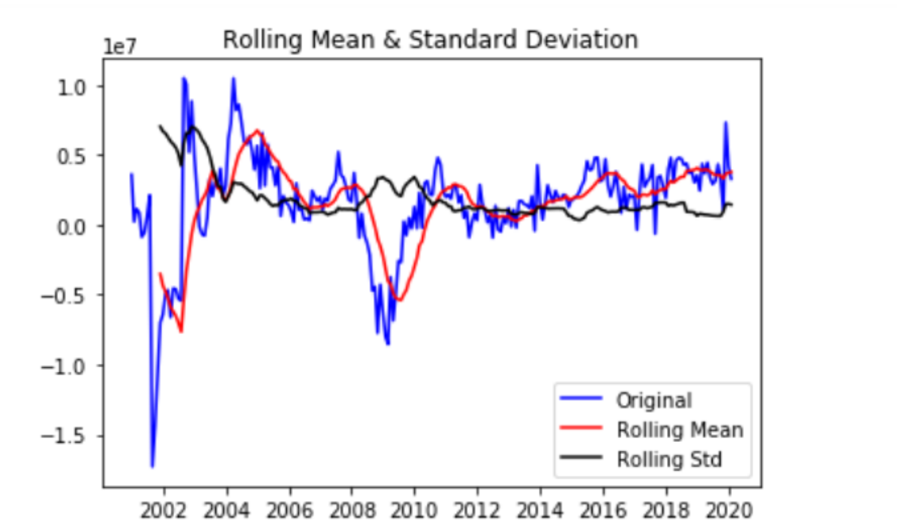


```
Results of Dickey-Fuller Test:
Test Statistic                   0.223216
p-value                          0.973534
#Lags Used                      13.000000
Number of Observations Used    228.000000
Critical Value (1%)             -3.459361
Critical Value (5%)             -2.874302
Critical Value (10%)            -2.573571
dtype: float64
```

So as we can see that variance are not constant with time and P-value are higher than .05 so we can say that data is non-stationary.

Making Data Stationary Using Differencing:

We can make data stationary by taking difference of legs here legs mean previous data points in the data sets.



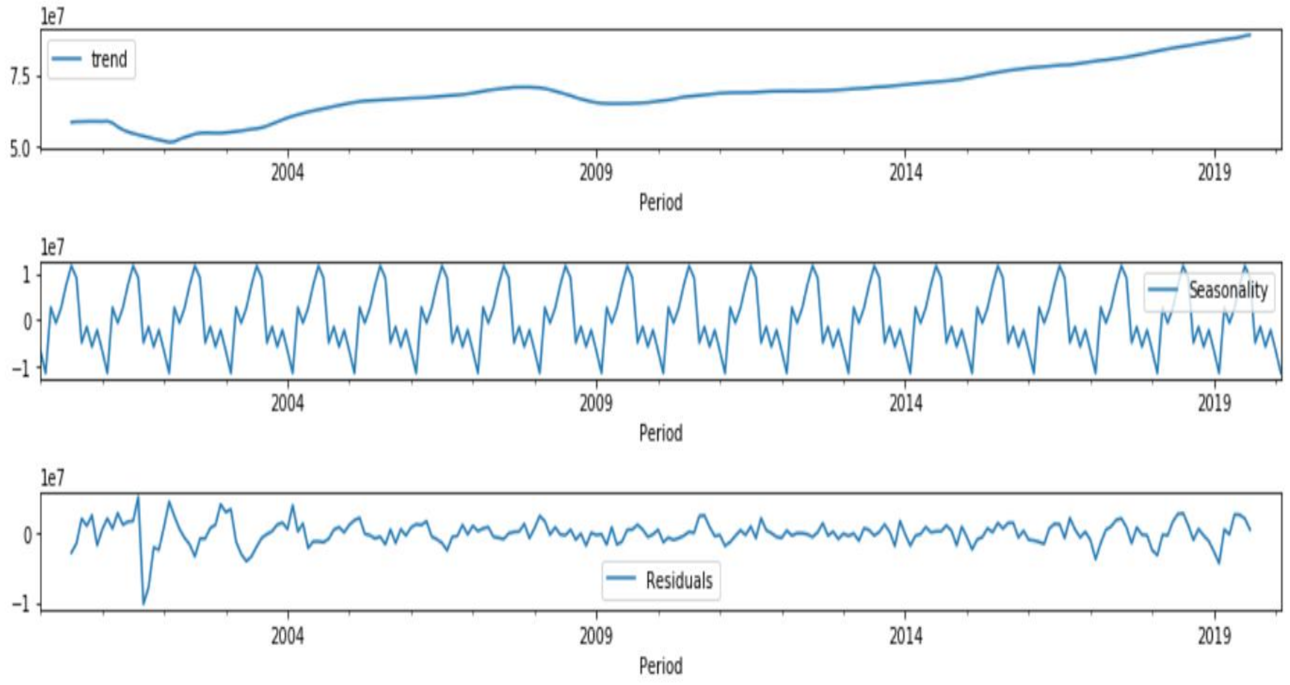
```
Results of Dickey-Fuller Test:
Test Statistic                  -2.923398
p-value                          0.042694
#Lags Used                      13.000000
Number of Observations Used    216.000000
Critical Value (1%)             -3.460992
Critical Value (5%)             -2.875016
Critical Value (10%)            -2.573952
dtype: float64
```

By looking at P-value we can say that our data is stationary.

Decomposition:

Time series decomposition involves thinking of a series as a combination of level, trend, seasonality, and noise components. Decomposition provides a useful abstract model for thinking about time series generally and for better understanding problems during time series analysis and forecasting.

Here we use Additive Model. The additive model is useful when the seasonal variation is relatively constant over time.



Smoothing:

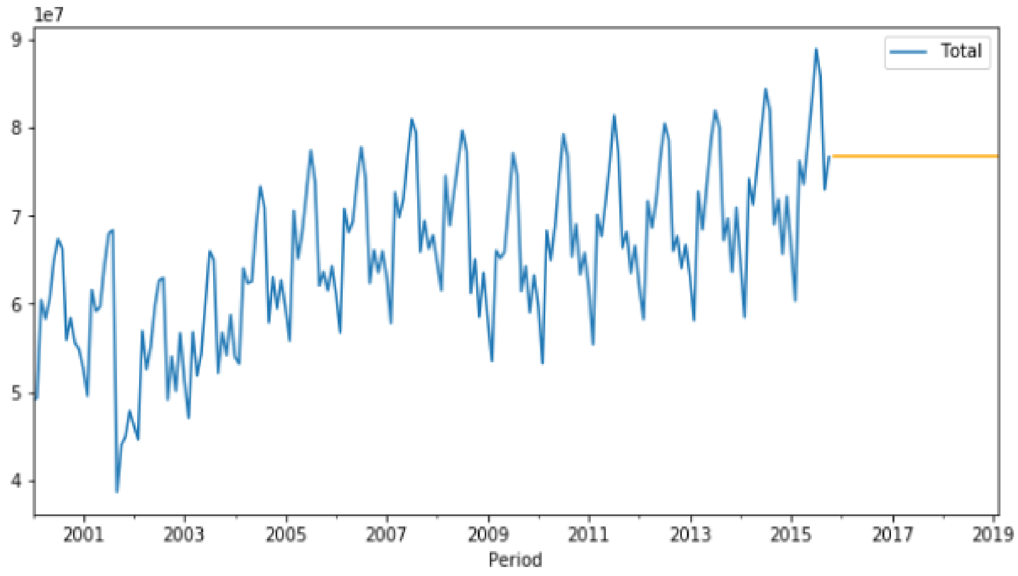
Smoothing is usually done to help us better see the patterns trends for example , in time series generally smooth out the irregular roughness to see a clear signal.

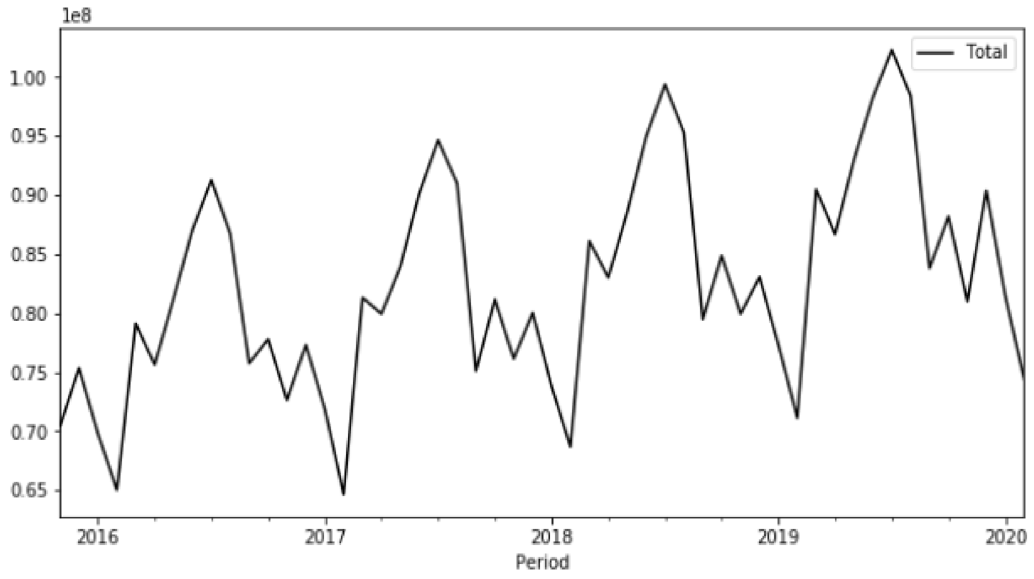
For seasonal data we might smooth out the seasonality so that we can identify the trend.

Simple exponential smoothing (SES):

The simplest of the exponentially smoothing method is naturally called simple exponentially smoothing. This method is suitable for forecasting the data with no clear trend or seasonality

In the below image we apply simple exponential smoothing but it gives less accurate results to us because seasonality is present in the dataset.

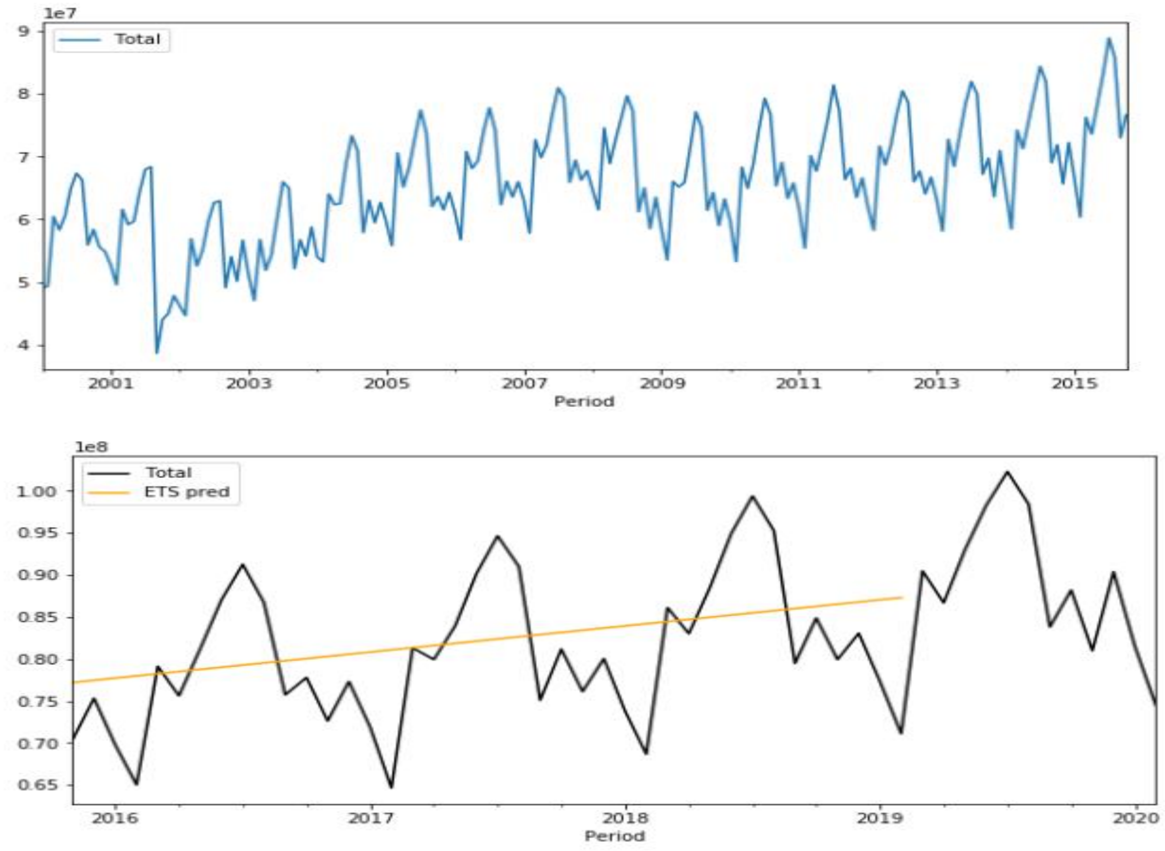




Simple Exponentially Smoothing

Exponential Smoothing:

Exponential smoothing is a rule of thumb technique for smoothing time series data using the exponential window function. Whereas in the simple moving average the past observations are weighted equally, exponential functions are used to assign exponentially decreasing weights over time

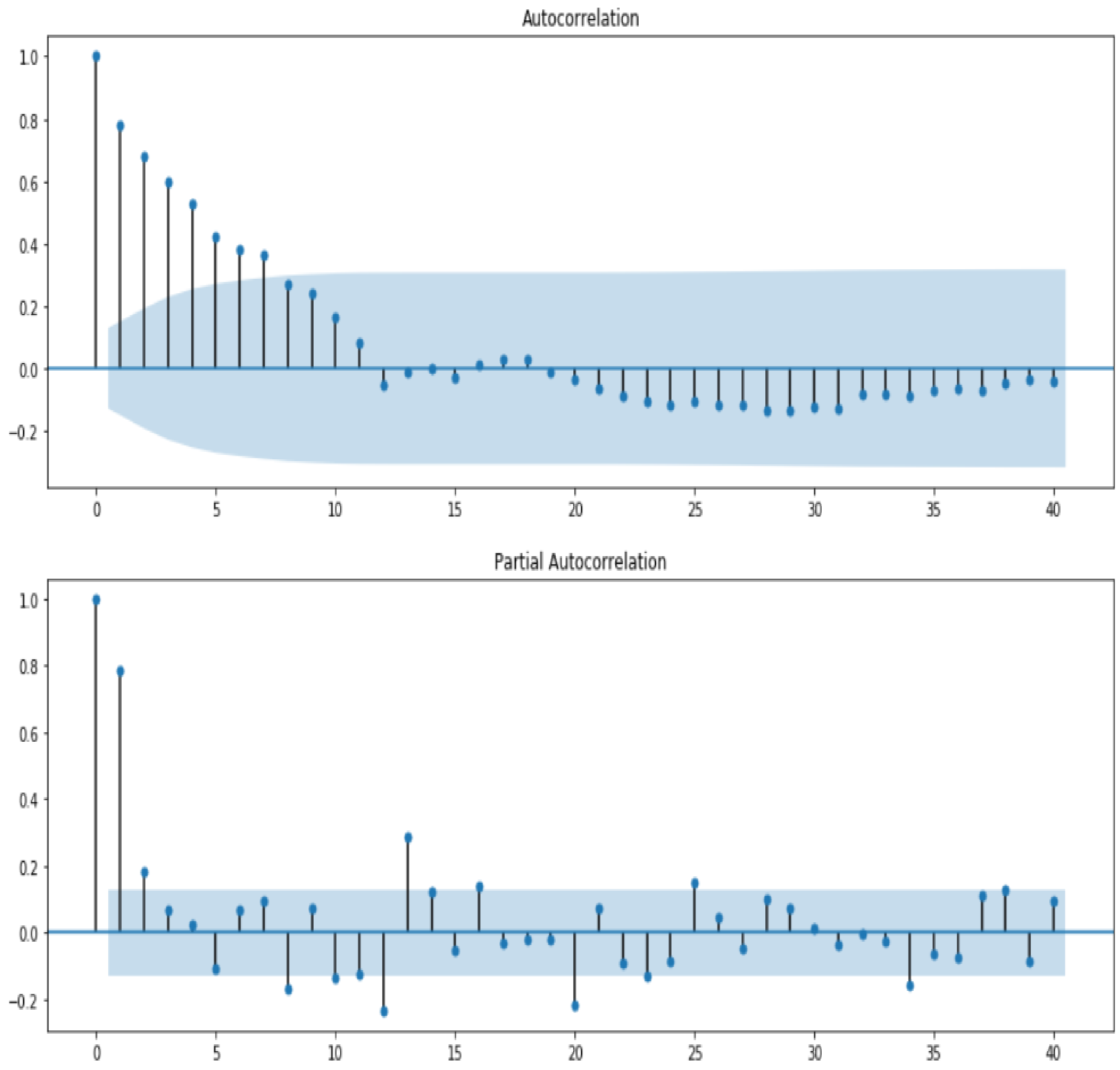


Exponential Smoothing

ACF and PACF Plots:

ACF is an (complete) auto-correlation function which gives us values of auto-correlation of any series with its lagged values. We plot these values along with the confidence band. We have an ACF plot. In simple terms, it describes how well the present value of the series is related with its past values. A time series can have components like trend, seasonality, cyclic and residual. ACF considers all these components while finding correlations hence it's a 'complete auto-correlation plot'.

PACF is a partial auto-correlation function. Basically instead of finding correlations of present with lags like ACF, it finds correlation of the residuals (which remains after removing the effects which are already explained by the earlier lag(s)) with the next lag value hence 'partial' and not 'complete' as we remove already found variations before we find the next correlation. So if there is any hidden information in the residual which can be modeled by the next lag, we might get a good correlation and we will keep that next lag as a feature while modeling.



ACF and PACF plots

Modeling:

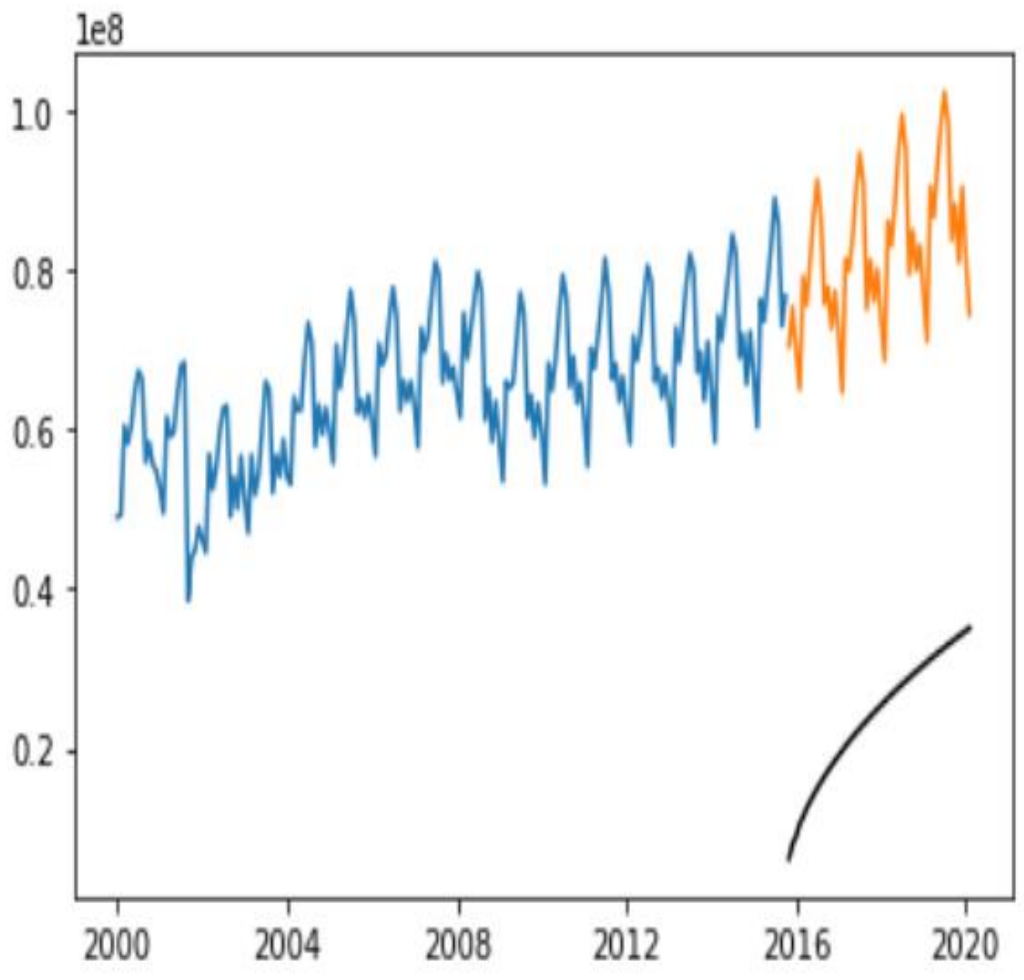
Autoregressive Integrated Moving Average (ARIMA)

Autoregressive Integrated Moving Average, or ARIMA, is one of the most widely used forecasting methods for univariate time series data forecasting.
Although the method can handle data with a trend, it does not support time series with a seasonal component.

This method has three variables to account for:

- P = Periods to lag
- D = number of differencing transformations required by the time series to get stationary.
- Q = this variable denotes the lag of the error component

We can see that ARIMA Model does not perform well because data contain seasonality ARIMA model capture only trend.



ARIMA MODEL

Seasonal Autoregressive Integrated Moving Average (SARIMA)

Seasonal Autoregressive Integrated Moving Average, SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component.

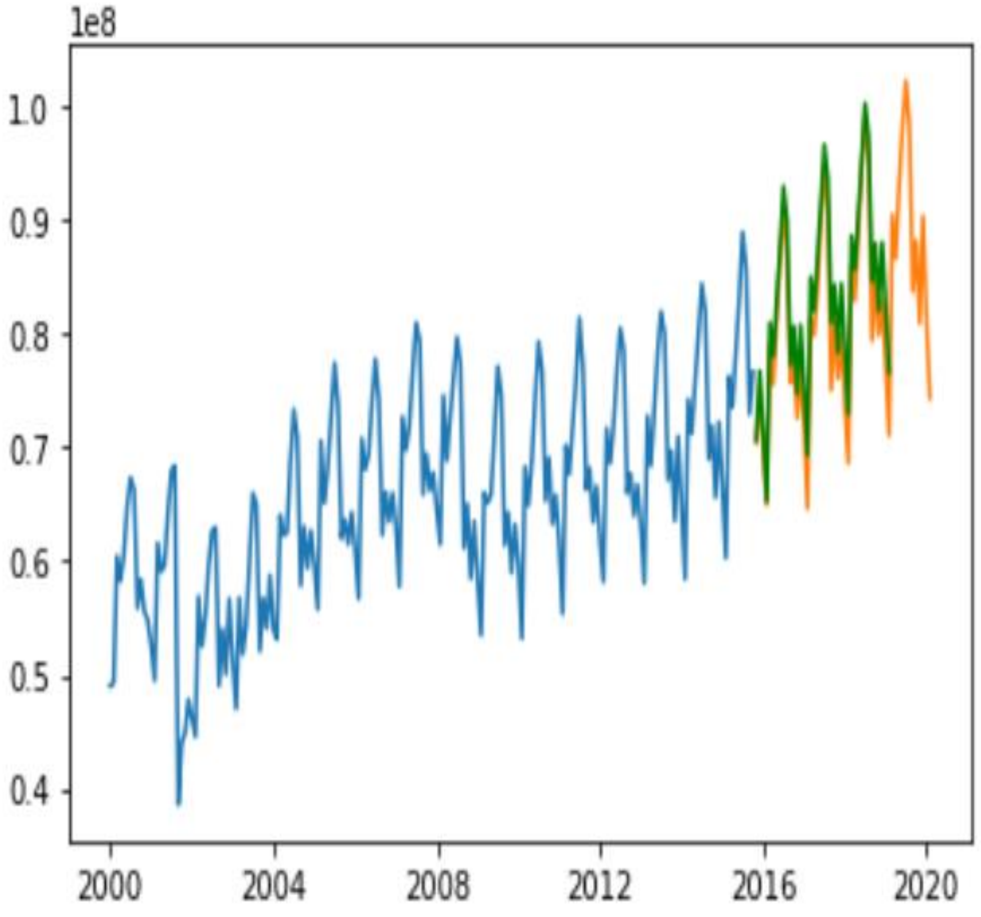
It adds three new hyper parameters to specify the auto regression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality.

This method has six variables to account for:

- p: Trend auto regression order.
- d: Trend difference order.
- q: Trend moving average order.

- P: Seasonal autoregressive order.
- D: Seasonal difference order.
- Q: Seasonal moving average order.
- m: The number of time steps for a single seasonal period.

Now SARIMA perform well on this data because SARIMA capture trend as well as seasonality.

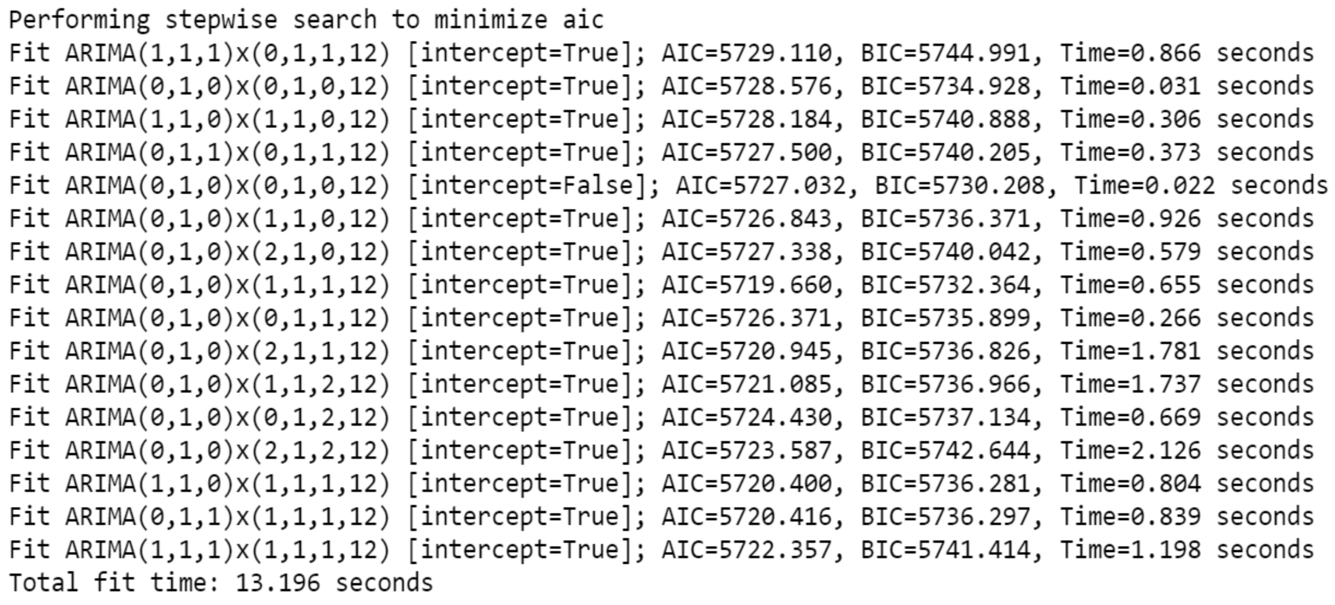


SARIMA MODEL

AUTOARIMA

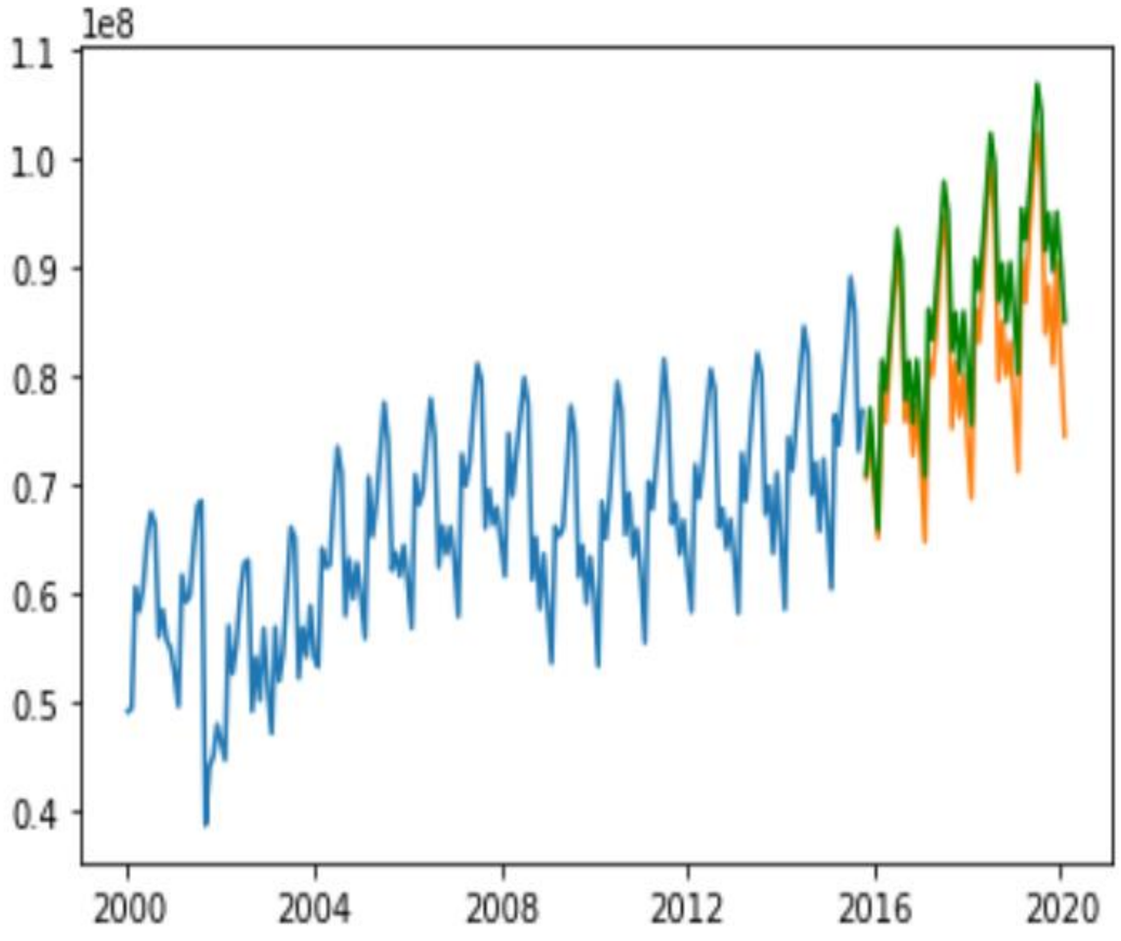
Although ARIMA and SARIMA is a very powerful model for forecasting time series data the data preparation and parameter tuning end up being very time consuming .Before implementing ARIMA you need to make the series stationary and determine the value of p and q using plots but AUTOARIMA Model automate this process itself

AUTOARIMA calculates best parameter values
Based on the AIC and BIC Values.



```
Performing stepwise search to minimize aic
Fit ARIMA(1,1,1)x(0,1,1,12) [intercept=True]; AIC=5729.110, BIC=5744.991, Time=0.866 seconds
Fit ARIMA(0,1,0)x(0,1,0,12) [intercept=True]; AIC=5728.576, BIC=5734.928, Time=0.031 seconds
Fit ARIMA(1,1,0)x(1,1,0,12) [intercept=True]; AIC=5728.184, BIC=5740.888, Time=0.306 seconds
Fit ARIMA(0,1,1)x(0,1,1,12) [intercept=True]; AIC=5727.500, BIC=5740.205, Time=0.373 seconds
Fit ARIMA(0,1,0)x(0,1,0,12) [intercept=False]; AIC=5727.032, BIC=5730.208, Time=0.022 seconds
Fit ARIMA(0,1,0)x(1,1,0,12) [intercept=True]; AIC=5726.843, BIC=5736.371, Time=0.926 seconds
Fit ARIMA(0,1,0)x(2,1,0,12) [intercept=True]; AIC=5727.338, BIC=5740.042, Time=0.579 seconds
Fit ARIMA(0,1,0)x(1,1,1,12) [intercept=True]; AIC=5719.660, BIC=5732.364, Time=0.655 seconds
Fit ARIMA(0,1,0)x(0,1,1,12) [intercept=True]; AIC=5726.371, BIC=5735.899, Time=0.266 seconds
Fit ARIMA(0,1,0)x(2,1,1,12) [intercept=True]; AIC=5720.945, BIC=5736.826, Time=1.781 seconds
Fit ARIMA(0,1,0)x(1,1,2,12) [intercept=True]; AIC=5721.085, BIC=5736.966, Time=1.737 seconds
Fit ARIMA(0,1,0)x(0,1,2,12) [intercept=True]; AIC=5724.430, BIC=5737.134, Time=0.669 seconds
Fit ARIMA(0,1,0)x(2,1,2,12) [intercept=True]; AIC=5723.587, BIC=5742.644, Time=2.126 seconds
Fit ARIMA(1,1,0)x(1,1,1,12) [intercept=True]; AIC=5720.400, BIC=5736.281, Time=0.804 seconds
Fit ARIMA(0,1,1)x(1,1,1,12) [intercept=True]; AIC=5720.416, BIC=5736.297, Time=0.839 seconds
Fit ARIMA(1,1,1)x(1,1,1,12) [intercept=True]; AIC=5722.357, BIC=5741.414, Time=1.198 seconds
Total fit time: 13.196 seconds
```

The blue line show train data points, yellow line shows test data points and green line show the forecast value of dataset and in this dataset AUTOARIMA perform well as compare to other models.



AUTOARIMA MODEL

Conclusion:

In this Project, we consider the U.S. Air Traffic Dataset from 2000 to 2020. After removing seasonality and trend from the dataset we fit different models based on minimum AIC, BIC Selection criteria The AUTOARIMA (0, 1, 0)*(1, 1, 1, 12) was chosen by AIC Criterion was good at diagnostics and was good to do the predications.