**Business Report**

**Project – Time series Forecasting**

**Created by Amit Jain**

Contents

[List of Figure 5](#_Toc112500642)

[Rose.csv 7](#_Toc112500643)

[1. Read the data as an appropriate Time Series data and plot the data. 7](#_Toc112500644)

[Plot for Rose wine Sales data 8](#_Toc112500645)

[2. Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition. 9](#_Toc112500646)

[Box Plot for Year on year sales data 9](#_Toc112500647)

[Monthly plot 10](#_Toc112500648)

[Plot a month plot of the give Time Series: 11](#_Toc112500649)

[Plot the Time Series according to different months for different years 12](#_Toc112500650)

[Yearly Plot: 13](#_Toc112500651)

[Quarterly plot – 14](#_Toc112500652)

[Daily plot 15](#_Toc112500653)

[Decade Plot 16](#_Toc112500654)

[Sales data without Seasonality component: 18](#_Toc112500655)

[Multiplicative Model for Rose data problem 19](#_Toc112500656)

[3. Split the data into training and test. The test data should start in 1991. 20](#_Toc112500657)

[4. Build all the exponential smoothing models on the training data and evaluate the model using RMSE on the test data. Other additional models such as regression, naïve forecast models, simple average models, moving average models should also be built on the training data and check the performance on the test data using RMSE 22](#_Toc112500658)

[Model1: Linear Regression 22](#_Toc112500659)

[Model2 – Naïve model 23](#_Toc112500660)

[Model 3 – Simple Average 24](#_Toc112500661)

[Model4- Moving Average – 25](#_Toc112500662)

[Model -5- Exponential Smoothing 27](#_Toc112500663)

[Method 6: Double Exponential Smoothing (Holt's Model) 30](#_Toc112500664)

[Method 7: Triple Exponential Smoothing (Holt - Winter's Model) 32](#_Toc112500665)

[5. Check for the stationarity of the data on which the model is being built on using appropriate statistical tests and also mention the hypothesis for the statistical test. If the data is found to be non-stationary, take appropriate steps to make it stationary. Check the new data for stationarity and comment. 35](#_Toc112500666)

[6. Build an automated version of the ARIMA/SARIMA model in which the parameters are selected using the lowest Akaike Information Criteria (AIC) on the training data and evaluate this model on the test data using RMSE 37](#_Toc112500667)

[ARIMA Model: 37](#_Toc112500668)

[SARIMA Model: 37](#_Toc112500669)

[Auto ARIMA Model: 38](#_Toc112500670)

[Auto SARIMA: 41](#_Toc112500671)

[7. Build ARIMA/SARIMA models based on the cut-off points of ACF and PACF on the training data and evaluate this model on the test data using RMSE. 43](#_Toc112500672)

[Build Manual ARIMA Model 44](#_Toc112500673)

[Build Manual SARIMA Model: 46](#_Toc112500674)

[8. Build a table with all the models built along with their corresponding parameters and the respective RMSE values on the test data 48](#_Toc112500675)

[9. Based on the model-building exercise, build the most optimum model(s) on the complete data and predict 12 months into the future with appropriate confidence intervals/bands. 49](#_Toc112500676)

[10. Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales. 51](#_Toc112500677)

[Sparkling.csv 53](#_Toc112500678)

[1. Read the data as an appropriate Time Series data and plot the data. 53](#_Toc112500679)

[Plot for Rose wine Sales data 54](#_Toc112500680)

[2. Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition. 55](#_Toc112500681)

[Box Plot for Year on year sales data 55](#_Toc112500682)

[Monthly plot 56](#_Toc112500683)

[Plot a month plot of the give Time Series: 57](#_Toc112500684)

[Plot the Time Series according to different months for different years 58](#_Toc112500685)

[Yearly Plot: 59](#_Toc112500686)

[Quarterly plot – 60](#_Toc112500687)

[Daily plot 61](#_Toc112500688)

[Decade Plot 62](#_Toc112500689)

[Sales data without Seasonality component: 64](#_Toc112500690)

[Multiplicative Model for Rose data problem 65](#_Toc112500691)

[3. Split the data into training and test. The test data should start in 1991. 66](#_Toc112500692)

[4. Build all the exponential smoothing models on the training data and evaluate the model using RMSE on the test data. Other additional models such as regression, naïve forecast models, simple average models, moving average models should also be built on the training data and check the performance on the test data using RMSE 68](#_Toc112500693)

[Model1: Linear Regression 68](#_Toc112500694)

[Model2 – Naïve model 69](#_Toc112500695)

[Model 3 – Simple Average 70](#_Toc112500696)

[Model4- Moving Average – 71](#_Toc112500697)

[Model -5- Exponential Smoothing 74](#_Toc112500698)

[Method 6: Double Exponential Smoothing (Holt's Model) 77](#_Toc112500699)

[Method 7: Triple Exponential Smoothing (Holt - Winter's Model) 79](#_Toc112500700)

[5. Check for the stationarity of the data on which the model is being built on using appropriate statistical tests and also mention the hypothesis for the statistical test. If the data is found to be non-stationary, take appropriate steps to make it stationary. Check the new data for stationarity and comment. 83](#_Toc112500701)

[6. Build an automated version of the ARIMA/SARIMA model in which the parameters are selected using the lowest Akaike Information Criteria (AIC) on the training data and evaluate this model on the test data using RMSE 85](#_Toc112500702)

[ARIMA Model: 85](#_Toc112500703)

[SARIMA Model: 85](#_Toc112500704)

[Auto ARIMA Model: 86](#_Toc112500705)

[Auto SARIMA: 89](#_Toc112500706)

[7. Build ARIMA/SARIMA models based on the cut-off points of ACF and PACF on the training data and evaluate this model on the test data using RMSE. 91](#_Toc112500707)

[Build Manual ARIMA Model 92](#_Toc112500708)

[Build Manual SARIMA Model: 94](#_Toc112500709)

[8. Build a table with all the models built along with their corresponding parameters and the respective RMSE values on the test data 96](#_Toc112500710)

[9. Based on the model-building exercise, build the most optimum model(s) on the complete data and predict 12 months into the future with appropriate confidence intervals/bands. 97](#_Toc112500711)

[10. Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales. 99](#_Toc112500712)

# **List of Figure**

[Figure 1 Time series Plot 8](#_Toc112502798)

[Figure 2 Box plot for year to year sales 9](#_Toc112502799)

[Figure 3 Time series Plot 10](#_Toc112502800)

[Figure 4 Month Plot of the Given time series 11](#_Toc112502801)

[Figure 5 yearly sales across Month 12](#_Toc112502802)

[Figure 6 Sum of Sales Each year Plot 13](#_Toc112502803)

[Figure 7 Quarterly sales for each year 14](#_Toc112502804)

[Figure 8 Daily Sales for all years 15](#_Toc112502805)

[Figure 9 Decade Sales for all data 16](#_Toc112502806)

[Figure 10 Additive Decomposition 17](#_Toc112502807)

[Figure 11 Sales data without Seasonality 18](#_Toc112502808)

[Figure 12 Multiplicative Sales Decomposition 19](#_Toc112502809)

[Figure 13 Plot Train and test data 21](#_Toc112502810)

[Figure 14 Linear regression Prediction Plot 22](#_Toc112502811)

[Figure 15 Prediction Plot for Naïve based Model 23](#_Toc112502812)

[Figure 16 Prediction Plot for Simple Average 24](#_Toc112502813)

[Figure 17 Moving Average for 2,3,4 and 6 MA 25](#_Toc112502814)

[Figure 18 Plot MA on test data 25](#_Toc112502815)

[Figure 19 Building all Models on Test data 26](#_Toc112502816)

[Figure 20 SES Model on Auto parameters 28](#_Toc112502817)

[Figure 21 SES Model on Corrected parameters 29](#_Toc112502818)

[Figure 22 DES model 31](#_Toc112502819)

[Figure 23 TES Model on Auto Parameters 32](#_Toc112502820)

[Figure 24 TES Model on Corrected parameters 33](#_Toc112502821)

[Figure 25 SES, DES and TES Model together comparison on Test data 34](#_Toc112502822)

[Figure 26 Stationary Time series 36](#_Toc112502823)

[Figure 27 Diagnostic Plot for Auto ARIMA 40](#_Toc112502824)

[Figure 28 Auto SARIMA Model Diagnostic Plot 42](#_Toc112502825)

[Figure 29 Manual ARIMA ACF/PACF Plot 44](#_Toc112502826)

[Figure 30 Manual ARIMA Diagnostic Plot 46](#_Toc112502827)

[Figure 31 Manual SARIMA ACF/PACF Plot 46](#_Toc112502828)

[Figure 32 Manual SAIRMA Diagnostic Plot 47](#_Toc112502829)

[Figure 33 Best Model TES plot for next 12 Months predictions 49](#_Toc112502830)

[Figure 34 TES Model next 12 Month Predictions with Confidence Band 50](#_Toc112502831)

[Figure 35 Time series Plot 54](#_Toc112502832)

[Figure 36 Box plot for year to year sales 55](#_Toc112502833)

[Figure 37 Time series Plot 56](#_Toc112502834)

[Figure 38 Month Plot of the Given time series 57](#_Toc112502835)

[Figure 39 yearly sales across Month 58](#_Toc112502836)

[Figure 40 Sum of Sales Each year Plot 59](#_Toc112502837)

[Figure 41 Quarterly sales for each year 60](#_Toc112502838)

[Figure 42 Daily Sales for all years 61](#_Toc112502839)

[Figure 43 Decade Sales for all data 62](#_Toc112502840)

[Figure 44 Additive Decomposition 63](#_Toc112502841)

[Figure 45 Sales data without Seasonality 64](#_Toc112502842)

[Figure 46 Multiplicative Sales Decomposition 65](#_Toc112502843)

[Figure 47 Plot Train and test data 67](#_Toc112502844)

[Figure 48 Linear regression Prediction Plot 68](#_Toc112502845)

[Figure 49 Prediction Plot for Naïve based Model 69](#_Toc112502846)

[Figure 50 Prediction Plot for Simple Average 70](#_Toc112502847)

[Figure 51 Moving Average for 2,3,4 and 6 MA 71](#_Toc112502848)

[Figure 52 Plot MA on Full data 72](#_Toc112502849)

[Figure 53 Building all Models on Test data 73](#_Toc112502850)

[Figure 54 SES Model on Auto parameters 75](#_Toc112502851)

[Figure 55 SES Model on Corrected parameters 76](#_Toc112502852)

[Figure 56 DES model 78](#_Toc112502853)

[Figure 57 TES Model on Auto Parameters 80](#_Toc112502854)

[Figure 58 TES Model on Corrected parameters 81](#_Toc112502855)

[Figure 59 SES, DES and TES Model together comparison on Test data 82](#_Toc112502856)

[Figure 60 Stationary Time series 84](#_Toc112502857)

[Figure 61 Diagnostic Plot for Auto ARIMA 88](#_Toc112502858)

[Figure 62 Auto SARIMA Model Diagnostic Plot 90](#_Toc112502859)

[Figure 63 Manual ARIMA ACF/PACF Plot 92](#_Toc112502860)

[Figure 64 Manual ARIMA Diagnostic Plot 93](#_Toc112502861)

[Figure 65 Manual SARIMA ACF/PACF Plot 94](#_Toc112502862)

[Figure 66 Manual SAIRMA Diagnostic Plot 95](#_Toc112502863)

[Figure 67 Best Model TES plot for next 12 Months predictions 97](#_Toc112502864)

[Figure 68 TES Model next 12 Month Predictions with Confidence Band 98](#_Toc112502865)

# **Rose.csv**

Introduction: This report explains the business requirements and provide the detailed solution based on the data provided for each problem statement. given in the assignment. Also, the purpose of this exercise is to execute various Timeseries forecasting learning techniques and building various models over Timeseries data, combine all predictions and find out the model with best prediction or accuracy. Timeseries data is unlikely normal data used in supervised/unsupervised learning, where we have dependent /independent variables.

In TS data, we need to use predictions on Same field, using Old data set in ordered Time manner.

# **Read the data as an appropriate Time Series data and plot the data.**

Time Series is a sequence of observations recorded at regular time intervals.

|  |  |  |
| --- | --- | --- |
|  | **YearMonth** | **Rose** |
| **0** | 1980-01-01 | 112.0 |
| **1** | 1980-02-01 | 118.0 |
| **2** | 1980-03-01 | 129.0 |
| **3** | 1980-04-01 | 99.0 |
| **4** | 1980-05-01 | 116.0 |
| **...** | ... | ... |
| **182** | 1995-03-01 | 45.0 |
| **183** | 1995-04-01 | 52.0 |
| **184** | 1995-05-01 | 28.0 |
| **185** | 1995-06-01 | 40.0 |
| **186** | 1995-07-01 | 62.0 |

187 rows × 2 columns

Given data is not time. So, we parse the date range and create a timestamp.

# **Plot for Rose wine Sales data**

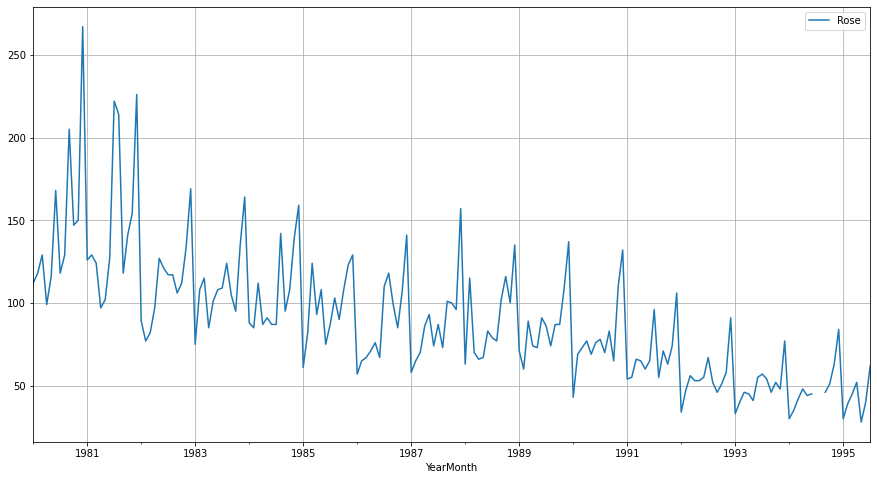


Figure Time series Plot

Describer for Rose Wine Sample data:

count 185.000000

mean 90.394595

std 39.175344

min 28.000000

25% 63.000000

50% 86.000000

75% 112.000000

max 267.000000

Insights:

* 1. Data consist of 187 data points
  2. It seems to be contained seasonality
  3. We also notice the fluctuations in the trend in the initial years and slowly decreasing the following years.
  4. Minimum sales for the data in Any month are 28 and Max sales of Rose wines in any month is 267
  5. Year 1981 has highest sales and 1995 has lowest sales

# **Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.**

# **Box Plot for Year on year sales data**

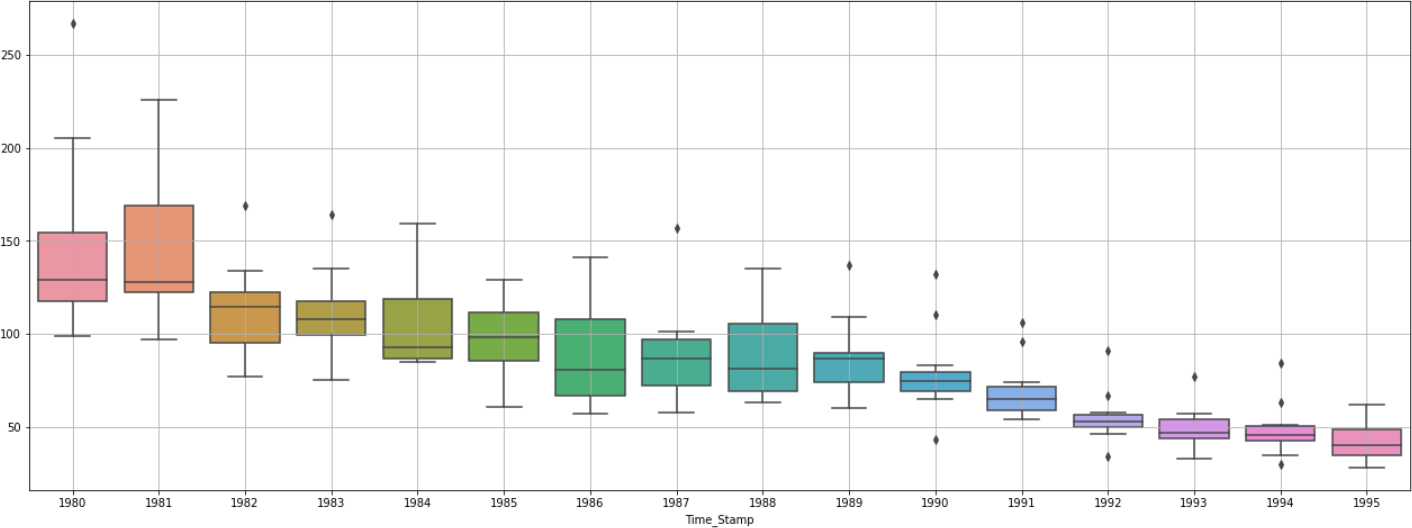


Figure Box plot for year to year sales

Insights:

1. We see the Big increase in Sales for Rose wine in year 1981 and soon after that started decreasing Also there were years with Rise and fall equally. But gradually Sales goes down.
2. Boxplot helps to check the outliers in each year and month and we see there are outliers in almost all the year as per the box plot.
3. Average sales are lowest in the year of 1995

# **Monthly plot**

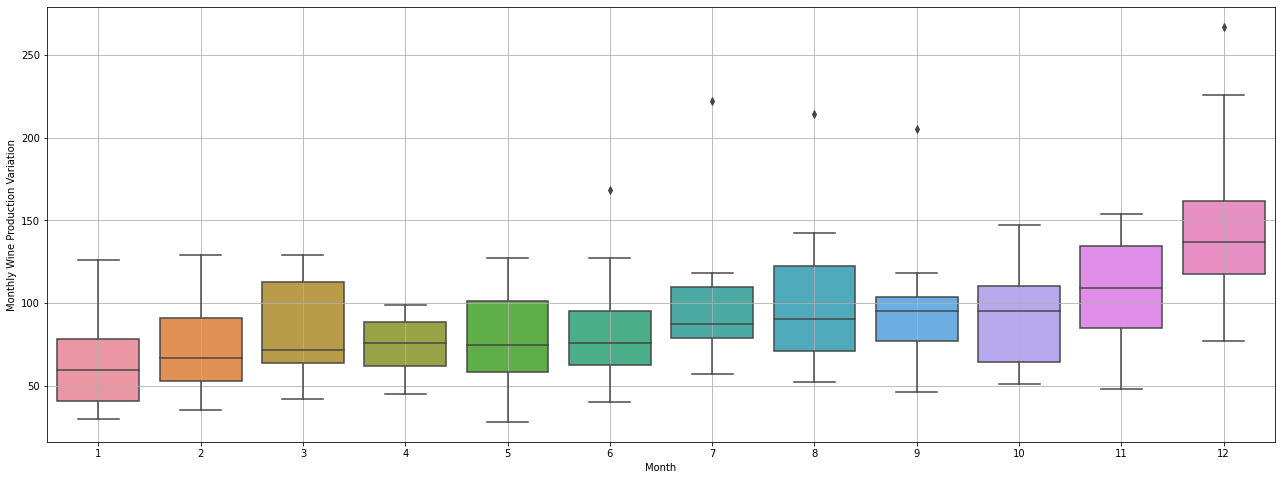


Figure Time series Plot

Insights:

* 1. The box plot for various months is plotted
  2. Monthly plot contains outliers in the month of June, July, August, September and December.
  3. There are Highest sales in the Month of December followed by 2nd highest Sales in November Month, which indicates year end party and vacation celebration sales

# **Plot a month plot of the give Time Series:**

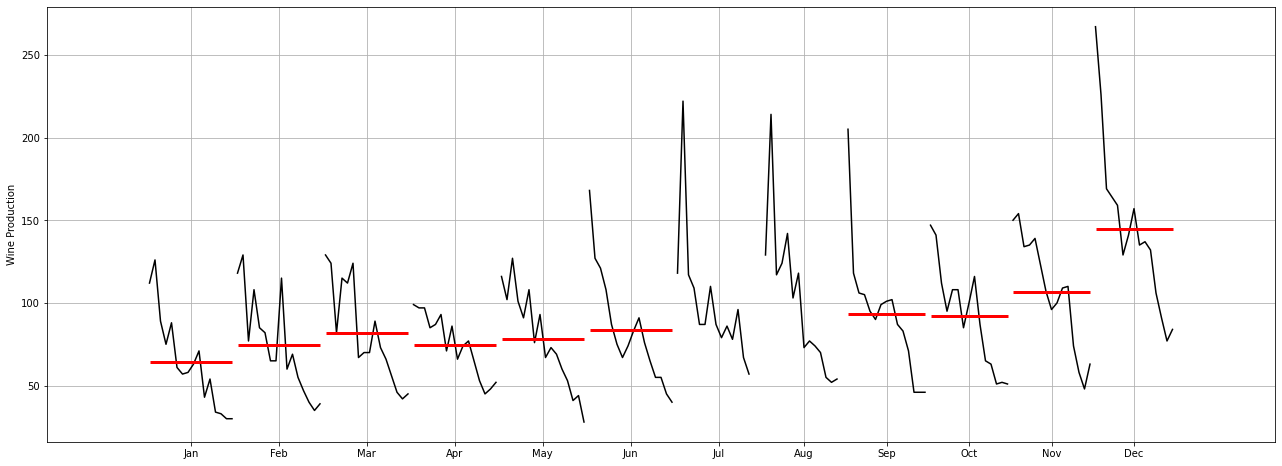


Figure Month Plot of the Given time series

Insights:

1. We have accumulated all year’s data for each month and plotted it against each month.
2. We see that there are High Sales in each month at the month Start and then Fall down till Mid of the month,
3. We also see small increase in Sales in every Mid of the Month and then again goes down till Month End.
4. It clearly indicates, when Anyone gets the Salary at the Start of the month and sometimes biweekly, then Sales for Month Start is always High and small Rise in Mid of the Month.

# **Plot the Time Series according to different months for different years**

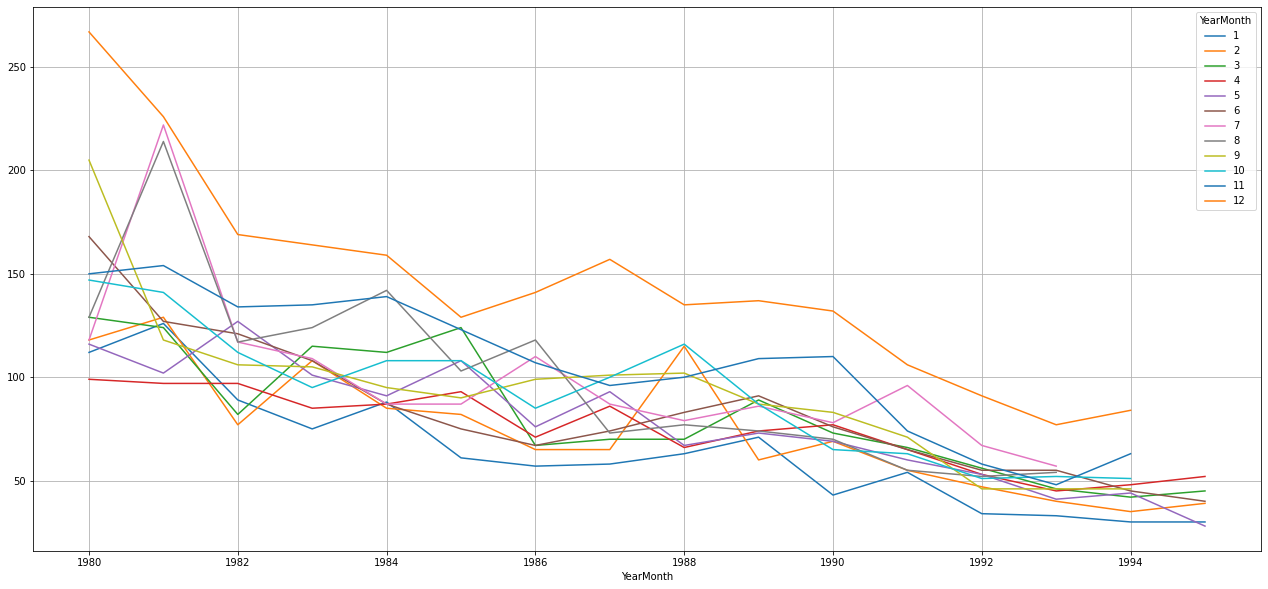


Figure yearly sales across Month

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **YearMonth** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| **YearMonth** |  |  |  |  |  |  |  |  |  |  |  |  |
| **1980** | 112.0 | 118.0 | 129.0 | 99.0 | 116.0 | 168.0 | 118.0 | 129.0 | 205.0 | 147.0 | 150.0 | 267.0 |
| **1981** | 126.0 | 129.0 | 124.0 | 97.0 | 102.0 | 127.0 | 222.0 | 214.0 | 118.0 | 141.0 | 154.0 | 226.0 |
| **1982** | 89.0 | 77.0 | 82.0 | 97.0 | 127.0 | 121.0 | 117.0 | 117.0 | 106.0 | 112.0 | 134.0 | 169.0 |
| **1983** | 75.0 | 108.0 | 115.0 | 85.0 | 101.0 | 108.0 | 109.0 | 124.0 | 105.0 | 95.0 | 135.0 | 164.0 |
| **1984** | 88.0 | 85.0 | 112.0 | 87.0 | 91.0 | 87.0 | 87.0 | 142.0 | 95.0 | 108.0 | 139.0 | 159.0 |
| **1985** | 61.0 | 82.0 | 124.0 | 93.0 | 108.0 | 75.0 | 87.0 | 103.0 | 90.0 | 108.0 | 123.0 | 129.0 |
| **1986** | 57.0 | 65.0 | 67.0 | 71.0 | 76.0 | 67.0 | 110.0 | 118.0 | 99.0 | 85.0 | 107.0 | 141.0 |
| **1987** | 58.0 | 65.0 | 70.0 | 86.0 | 93.0 | 74.0 | 87.0 | 73.0 | 101.0 | 100.0 | 96.0 | 157.0 |
| **1988** | 63.0 | 115.0 | 70.0 | 66.0 | 67.0 | 83.0 | 79.0 | 77.0 | 102.0 | 116.0 | 100.0 | 135.0 |
| **1989** | 71.0 | 60.0 | 89.0 | 74.0 | 73.0 | 91.0 | 86.0 | 74.0 | 87.0 | 87.0 | 109.0 | 137.0 |
| **1990** | 43.0 | 69.0 | 73.0 | 77.0 | 69.0 | 76.0 | 78.0 | 70.0 | 83.0 | 65.0 | 110.0 | 132.0 |
| **1991** | 54.0 | 55.0 | 66.0 | 65.0 | 60.0 | 65.0 | 96.0 | 55.0 | 71.0 | 63.0 | 74.0 | 106.0 |
| **1992** | 34.0 | 47.0 | 56.0 | 53.0 | 53.0 | 55.0 | 67.0 | 52.0 | 46.0 | 51.0 | 58.0 | 91.0 |
| **1993** | 33.0 | 40.0 | 46.0 | 45.0 | 41.0 | 55.0 | 57.0 | 54.0 | 46.0 | 52.0 | 48.0 | 77.0 |
| **1994** | 30.0 | 35.0 | 42.0 | 48.0 | 44.0 | 45.0 | NaN | NaN | 46.0 | 51.0 | 63.0 | 84.0 |
| **1995** | 30.0 | 39.0 | 45.0 | 52.0 | 28.0 | 40.0 | 62.0 | NaN | NaN | NaN | NaN | NaN |

Insights:

1. December records have the high number of rose wine sales in each year.
2. May, January have low number of wine sales.
3. There are 2 Months data not available in July and August 1994 and then no data after July 1995.

# **Yearly Plot:**

aggregate the time series from an annual perspective and summing up the observations

|  |  |  |
| --- | --- | --- |
|  | **YearMonth** | **Rose** |
| **0** | 1980-12-31 | 1758.0 |
| **1** | 1981-12-31 | 1780.0 |
| **2** | 1982-12-31 | 1348.0 |
| **3** | 1983-12-31 | 1324.0 |
| **4** | 1984-12-31 | 1280.0 |

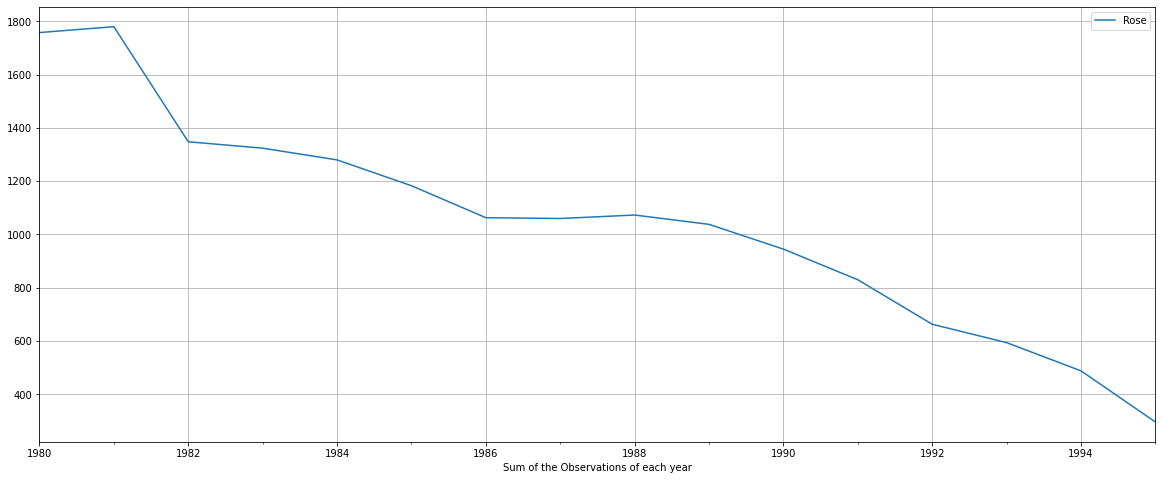


Figure Sum of Sales Each year Plot

Insights:

1. The plot shows that in 1982 there is a fall in the wine sales and there is a steep downfall is observed.
2. The resampled yearly or annual series have smoothened out the seasonality and have only been able to capture the year on year trend where there was.

# **Quarterly plot –**

aggregate the time series from a quarterly perspective and sum the observations of each quarter.

|  |  |  |
| --- | --- | --- |
|  | **YearMonth** | **Rose** |
| **0** | 1980-03-31 | 359.0 |
| **1** | 1980-06-30 | 383.0 |
| **2** | 1980-09-30 | 452.0 |
| **3** | 1980-12-31 | 564.0 |
| **4** | 1981-03-31 | 379.0 |

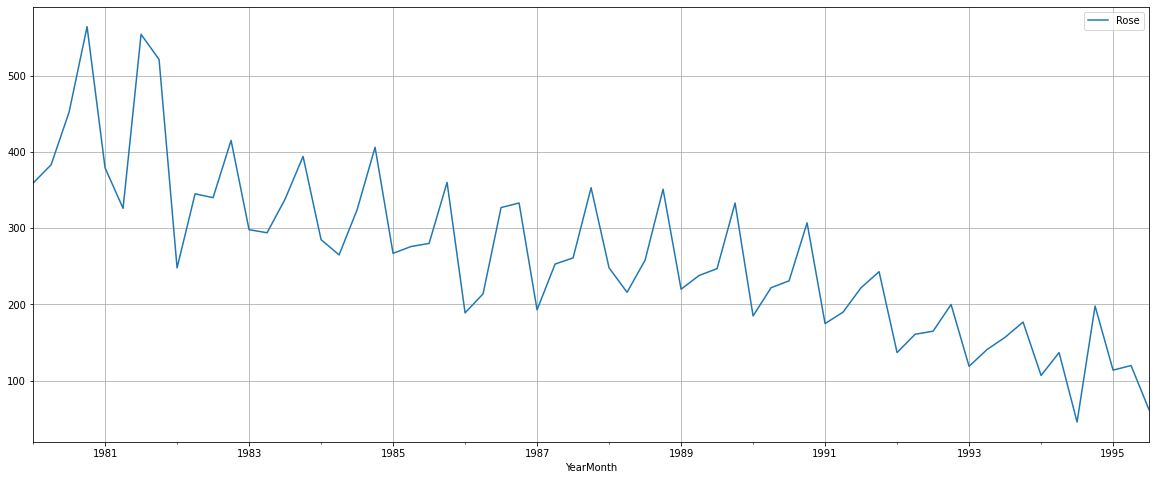


Figure Quarterly sales for each year

Insights:

1. There is some rise is found in Year 1984 and 1988, but ultimately that also goes down.
2. We see that the year on year quarterly series represents the year on year monthly series. The quarterly series is able to catch the seasonality in the data.

# **Daily plot**

aggregate the data from a daily perspective

|  |  |
| --- | --- |
| YearMonth | Rose |
| 0 | 1980-01-01 | 112.0 |
| 1 | 1980-01-02 | 0.0 |
| 2 | 1980-01-03 | 0.0 |
| 3 | 1980-01-04 | 0.0 |
| 4 | 1980-01-05 | 0.0 |
| ... | ... | ... |
| 5656 | 1995-06-27 | 0.0 |
| 5657 | 1995-06-28 | 0.0 |
| 5658 | 1995-06-29 | 0.0 |
| 5659 | 1995-06-30 | 0.0 |
| 5660 | 1995-07-01 | 62.0 |

5661 rows × 2 columns

The values which the original series cannot provide is taken as 0 by python if we try to resample the data on a daily basis.

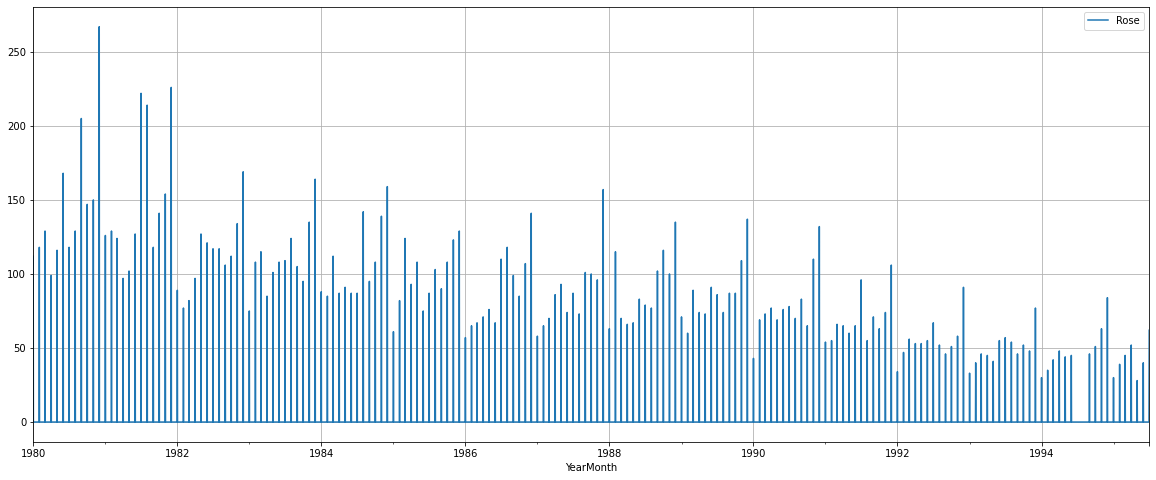


Figure Daily Sales for all years

The above graph fails to give us a proper understanding of our data. Thus, resampling the data to intervals where a number of observations are 0 is not a good idea as that does not give us an understanding of the performance of the time series.

To get a very high-level overview of the trend of the Time Series Data (if Trend is present) can be understood by resampling the data keeping the intervals very large.

# **Decade Plot**

|  | **YearMonth** | **Rose** |
| --- | --- | --- |
| **0** | 1980-12-31 | 1758.0 |
| **1** | 1990-12-31 | 12094.0 |
| **2** | 2000-12-31 | 2871.0 |

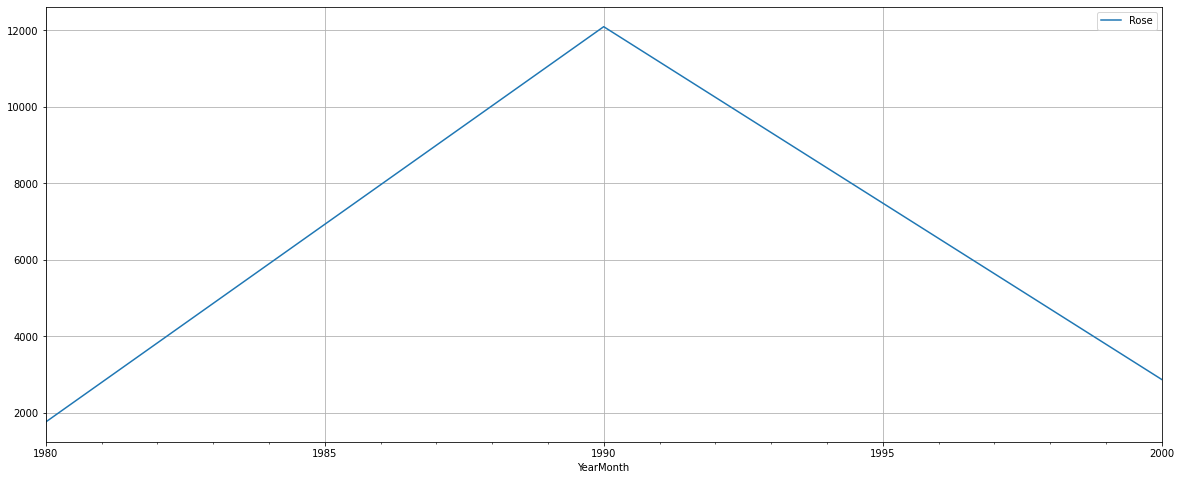


Figure Decade Sales for all data

If we take the resampling period to be 10 years or a decade, we see that the seasonality present has been smoothed over and it is only giving an estimate of the trend.

**Decompose the Time Series : Additive Model**

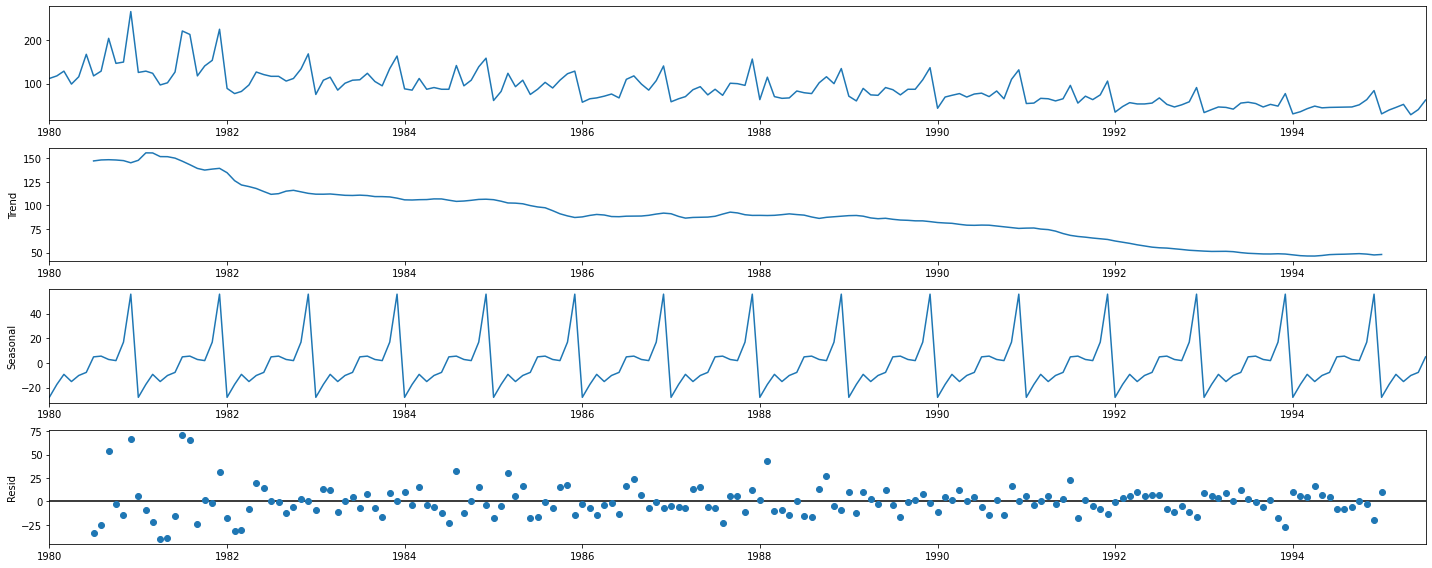


Figure Additive Decomposition

Insights:

1. We have built 2 Models of the data Additive trend as well as Multiplicative Trend .
2. From the ‘additive’ decomposition, there is seasonality in the data. Which is At the Start of the year Sales goes down, and at the end of the year, sales goes High for that year.
3. Sales trend is going down year by year

Trend

YearMonth

1980-01-01 NaN

1980-02-01 NaN

1980-03-01 NaN

1980-04-01 NaN

1980-05-01 NaN

1980-06-01 NaN

1980-07-01 147.083333

1980-08-01 148.125000

1980-09-01 148.375000

1980-10-01 148.083333

1980-11-01 147.416667

1980-12-01 145.125000

Name: trend, dtype: float64

Seasonality

YearMonth

1980-01-01 -27.908647

1980-02-01 -17.435632

1980-03-01 -9.285830

1980-04-01 -15.098330

1980-05-01 -10.196544

1980-06-01 -7.678687

1980-07-01 4.896908

1980-08-01 5.499686

1980-09-01 2.774686

1980-10-01 1.871908

1980-11-01 16.846908

1980-12-01 55.713575

Name: seasonal, dtype: float64

Residual

YearMonth

1980-01-01 NaN

1980-02-01 NaN

1980-03-01 NaN

1980-04-01 NaN

1980-05-01 NaN

1980-06-01 NaN

1980-07-01 -33.980241

1980-08-01 -24.624686

1980-09-01 53.850314

1980-10-01 -2.955241

1980-11-01 -14.263575

1980-12-01 66.161425

Name: resid, dtype: float64

# **Sales data without Seasonality component:**

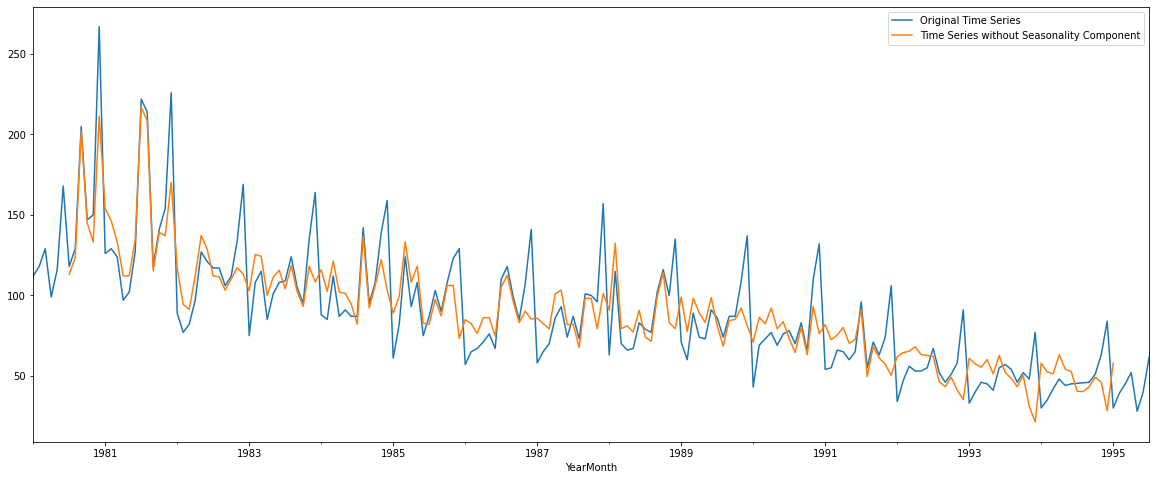


Figure Sales data without Seasonality

YearMonth

1980-01-01 NaN

1980-02-01 NaN

1980-03-01 NaN

1980-04-01 NaN

1980-05-01 NaN

1980-06-01 NaN

1980-07-01 113.103092

1980-08-01 123.500314

1980-09-01 202.225314

1980-10-01 145.128092

1980-11-01 133.153092

1980-12-01 211.286425

dtype: float64

# **Multiplicative Model for Rose data problem**

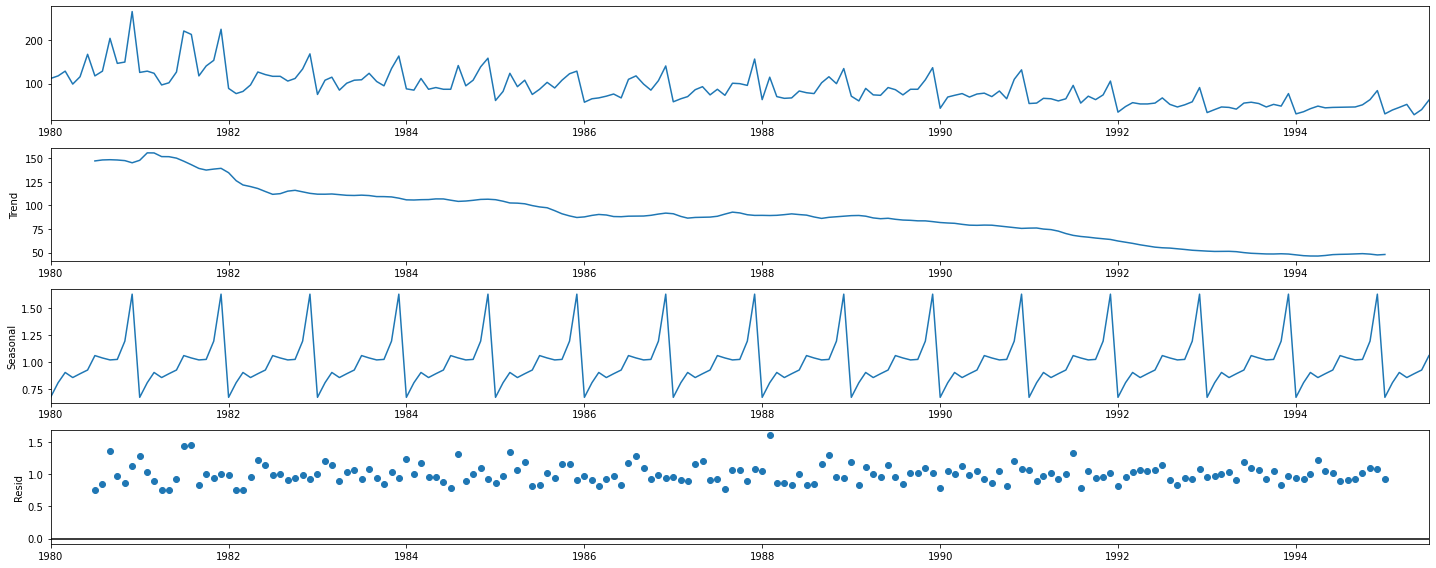


Figure Multiplicative Sales Decomposition

Since Residual is more close to a single line, we would choose Additive Decomposition over Multiplicative decomposing.

# **Split the data into training and test. The test data should start in 1991.**

We have split the data in train and test data set,

Data before 1991 is considered as train set and data after 1991 is considered as test data set:

After splitting this is the final shape of the Train and Test data

(132, 1)

(55, 1)

**First few rows of Training Data**

Rose

YearMonth

1980-01-01 112.0

1980-02-01 118.0

1980-03-01 129.0

1980-04-01 99.0

1980-05-01 116.0

**Last few rows of Training Data**

Rose

YearMonth

1990-08-01 70.0

1990-09-01 83.0

1990-10-01 65.0

1990-11-01 110.0

1990-12-01 132.0

First few rows of Test Data

Rose

YearMonth

1991-01-01 54.0

1991-02-01 55.0

1991-03-01 66.0

1991-04-01 65.0

1991-05-01 60.0

Last few rows of Test Data

Rose

YearMonth

1995-03-01 45.0

1995-04-01 52.0

1995-05-01 28.0

1995-06-01 40.0

1995-07-01 62.0



Figure Plot Train and test data

It is difficult to predict the future if the past is not happened. From the above split, we are predicting similar to the past data.

# **Build all the exponential smoothing models on the training data and evaluate the model using RMSE on the test data. Other additional models such as regression, naïve forecast models, simple average models, moving average models should also be built on the training data and check the performance on the test data using RMSE**

# **Model1: Linear Regression**

For this particular linear regression, we are going to regress the 'Rose' variable against the order of the occurrence. For this we need to modify our training data before fitting it into a linear regression.

Regress the “Rose” variable against the order of occurrence. We have also generated the numerical instance order for both training and test set . Linear Regression is built on the training and test dataset



Figure Linear regression Prediction Plot

We have evaluated the Model based on RMSE parameter and put this in a Data frame, which we would use later for Comparing multiple models

Model evaluation :

|  | **index** | **Test RMSE** |
| --- | --- | --- |
| **0** | RegressionOnTime | 15.268955 |

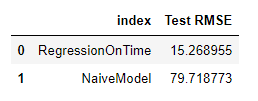
# **Model2 – Naïve model**

For this particular naive model, we say that the prediction for tomorrow is the same as today and the prediction for day after tomorrow is tomorrow and since the prediction of tomorrow is same as today, therefore the prediction for day after tomorrow is also today.



Figure Prediction Plot for Naïve based Model

Model Evaluation



# **Model 3 – Simple Average**

For this particular simple average method, we will forecast by using the average of the training values

|  |  |  |
| --- | --- | --- |
|  | **Rose** | **mean\_forecast** |
| **YearMonth** |  |  |
| **1991-01-01** | 54.0 | 104.939394 |
| **1991-02-01** | 55.0 | 104.939394 |
| **1991-03-01** | 66.0 | 104.939394 |
| **1991-04-01** | 65.0 | 104.939394 |
| **1991-05-01** | 60.0 | 104.939394 |

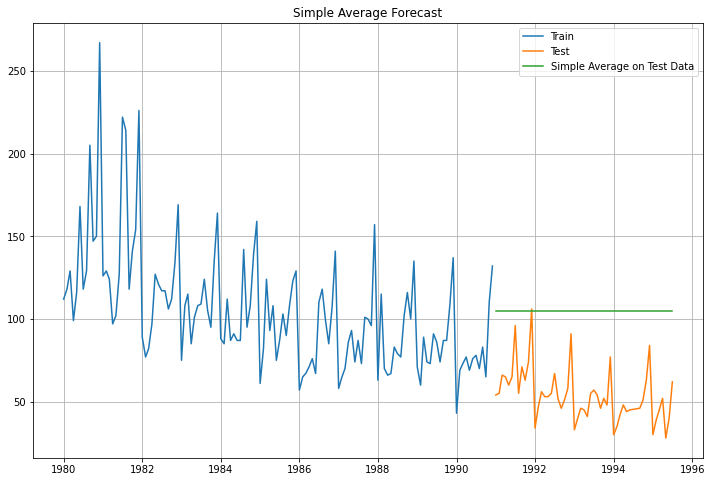


Figure Prediction Plot for Simple Average

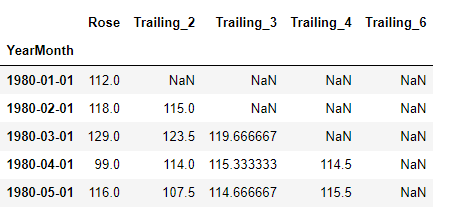
Model Evaluation:

|  |  |
| --- | --- |
|  | **Test RMSE** |
| **RegressionOnTime** | 15.268955 |
| **NaiveModel** | 79.718773 |
| **SimpleAverageModel** | 53.460570 |

# **Model4- Moving Average –**

For the moving average model, we are going to calculate rolling means (or moving averages) for different intervals. The best interval can be determined by the maximum accuracy (or the minimum error) over here.

Let’s take moving average of previous 2, 3, 4 and 6 data elements:



Plot the predictions based on Moving averages:

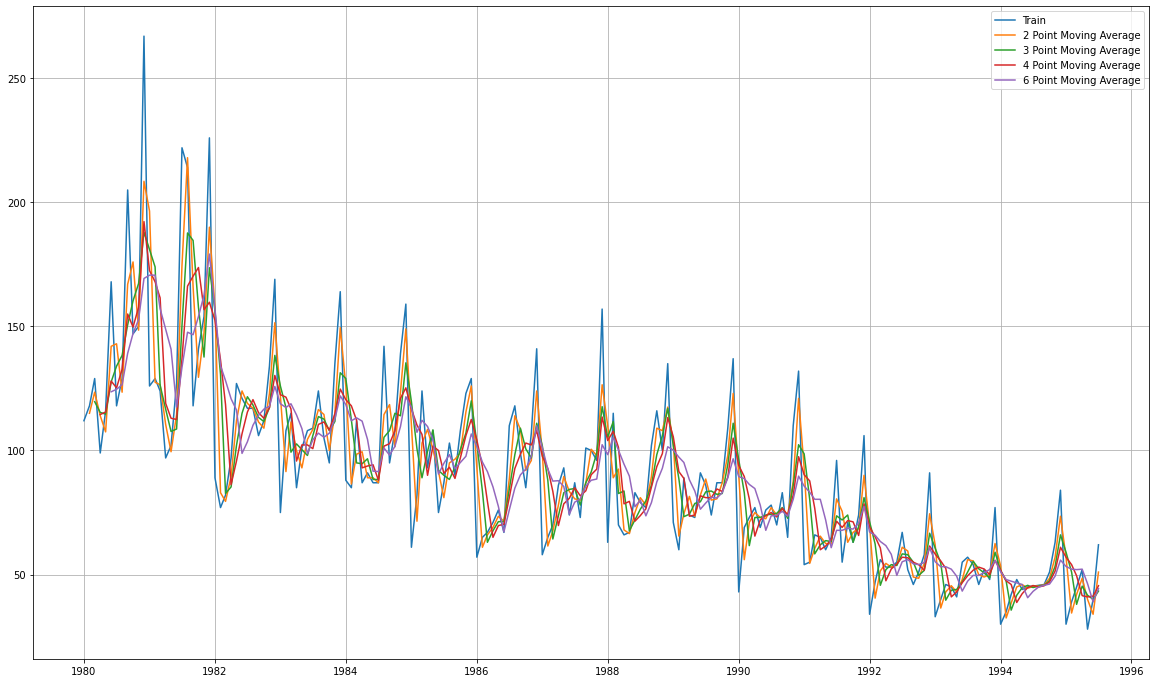


Figure Moving Average for 2,3,4 and 6 MA

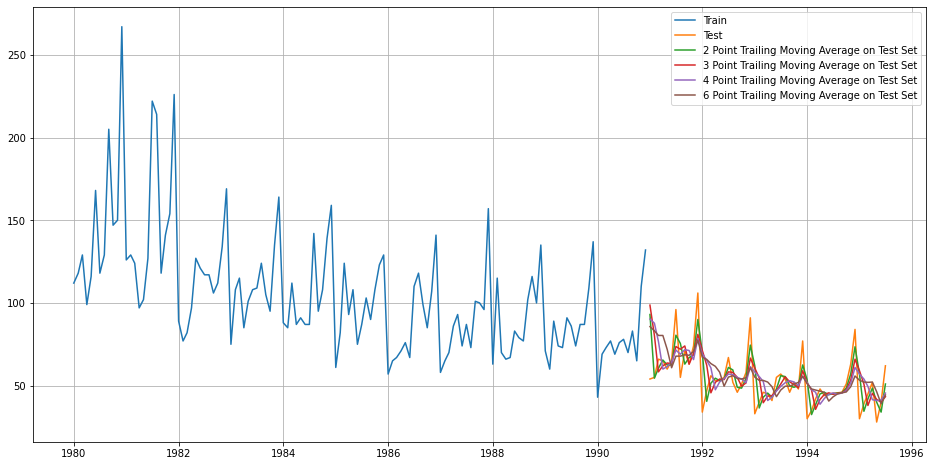
Let us split the data into train and test and plot this Time Series. The window of the moving average is need to be carefully selected as too big a window will result in not having any test set as the whole series might get averaged over. 

Figure Plot MA on test data

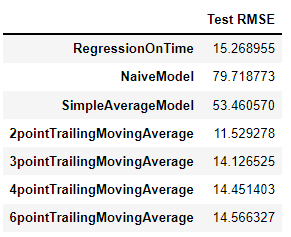
**Model Evaluation:**

For 2 point Moving Average Model forecast on the Training Data, RMSE is 11.529

For 3 point Moving Average Model forecast on the Training Data, RMSE is 14.127

For 4 point Moving Average Model forecast on the Training Data, RMSE is 14.451

For 6 point Moving Average Model forecast on the Training Data, RMSE is 14.566



The Best Model for Moving average is 2 points trailing moving average with lowest RMSE of 11.52, We would choose this model for predicting the Test model, if we are to choose the best one from above all models:

Before we go on to build the various Exponential Smoothing models, let us plot all the models and compare the Time Series plots

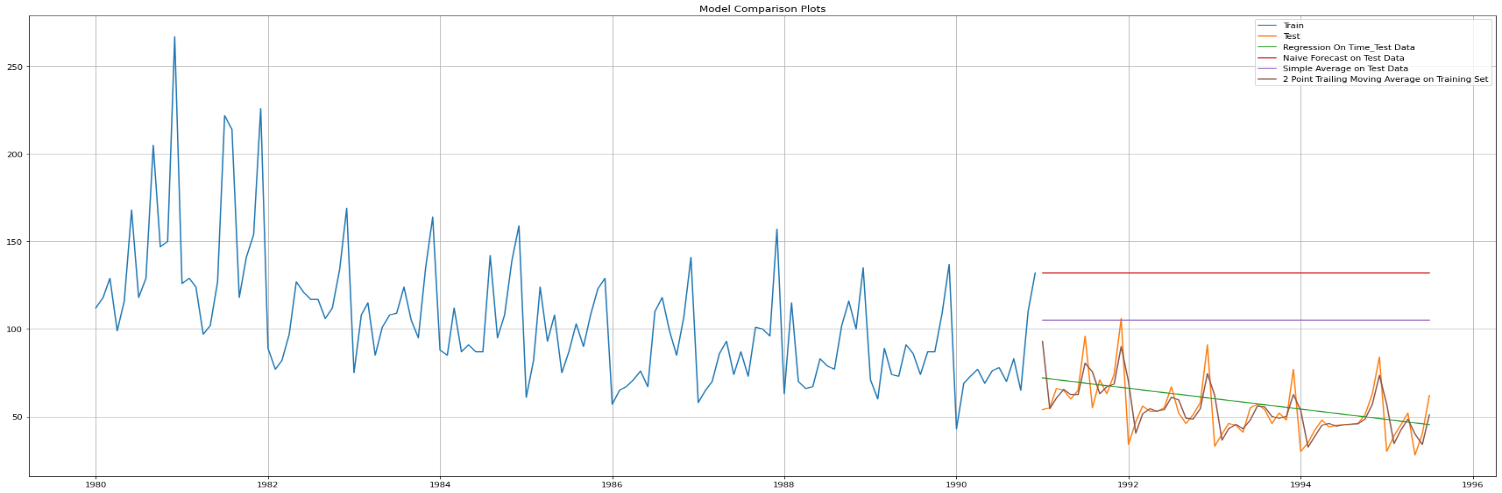


Figure Building all Models on Test data

# **Model -5- Exponential Smoothing**

Exponential smoothing methods consist of flattening time series data. Exponential smoothing averages or exponentially weighted moving averages consist of forecast based on previous periods data with exponentially declining influence on the older observations.

Exponential smoothing methods consist of special case exponential moving with notation ETS (Error, Trend, Seasonality) where each can be none(N), additive (N), additive damped (Ad), Multiplicative (M) or multiplicative damped (Md). One or more parameters control how fast the weights decay. These parameters have values between 0 and 1

First Model in Exponential smoothing is SES (Simple Exponential Smoothing) .

SES - ETS(A, N, N) - Simple Exponential Smoothing with additive errors

The simplest of the exponentially smoothing methods is naturally called simple exponential smoothing (SES). This method is suitable for forecasting data with no clear trend or seasonal pattern.

In Single ES, the forecast at time (t + 1) is given by Winters,1960

𝐹𝑡+1=𝛼𝑌𝑡+(1−𝛼)𝐹𝑡

Parameter 𝛼 is called the smoothing constant and its value lies between 0 and 1. Since the model uses only one smoothing constant, it is called Single Exponential Smoothing.

Note: Here, there is both trend and seasonality in the data. So, we should have directly gone for the Triple Exponential Smoothing but Simple Exponential Smoothing and the Double Exponential Smoothing models are built over here to get an idea of how the three types of models compare in this case.

SimpleExpSmoothing class must be instantiated and passed the training data.

The fit() function is then called providing the fit configuration, the alpha value, smoothing\_level. If this is omitted or set to None, the model will automatically optimize the value

Auto parameters:

{'smoothing\_level': 0.09874989743650385,

'smoothing\_trend': nan,

'smoothing\_seasonal': nan,

'damping\_trend': nan,

'initial\_level': 134.38699692184085,

'initial\_trend': nan,

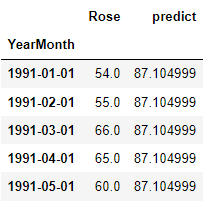
'initial\_seasons': array([], dtype=float64),

'use\_boxcox': False,

'lamda': None,

'remove\_bias': False}

Perform Prediction on test data set and these are the result look like:



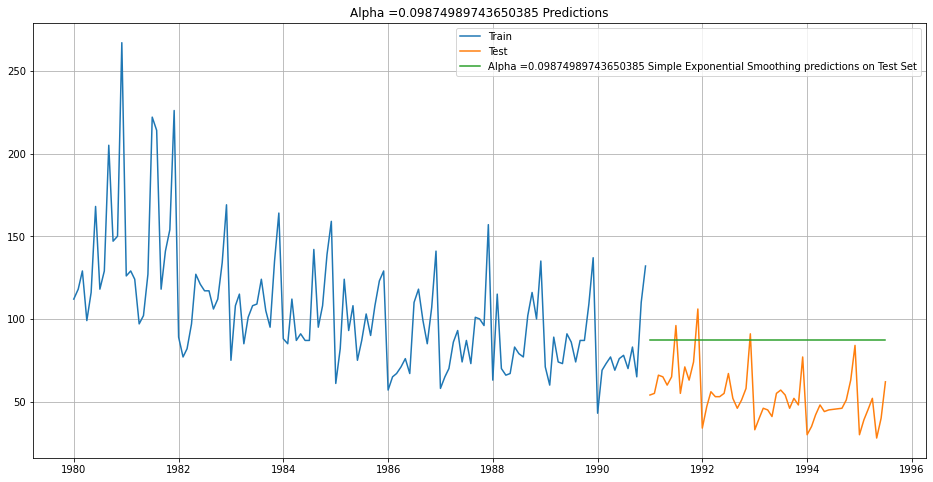
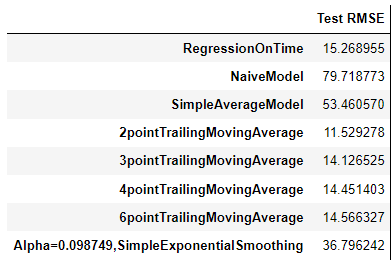


Figure SES Model on Auto parameters

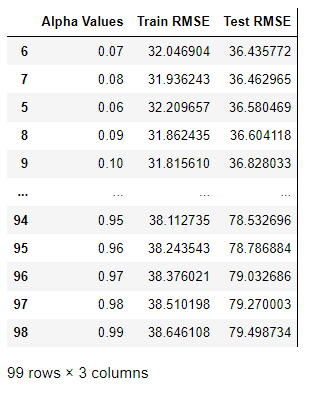
Model Evaluation for 𝛼 = 0.09874989743650385 : Simple Exponential Smoothing For Alpha =0.09874989743650385 Simple Exponential Smoothing Model forecast on the Test Data, RMSE is 36.796



Setting different alpha values. The higher the alpha value more weightage is given to the more recent observation. That means, what happened recently will happen again.

We will run a loop with different alpha values to understand which particular value works best for alpha on the test set. We have given a range of values between 0.01 to 1 with the interval of 0.01 so , Alpha values like 0.01, 0.02, 0.03 … upto 1.

We tried all 100 possible values and calculated the RMSE value for that, and chosen best Alpha value for Lowest RMSE value



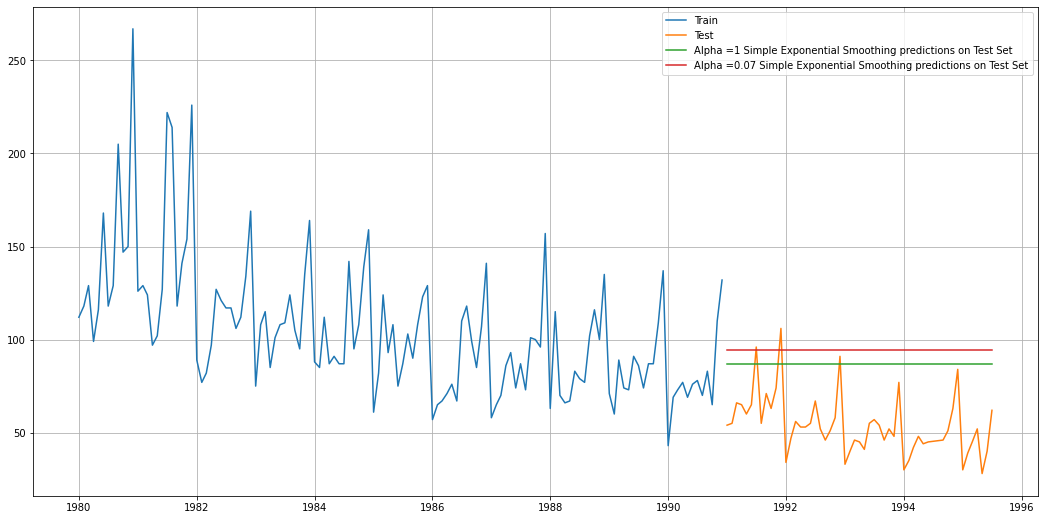
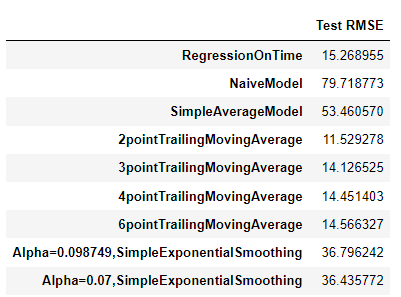


Figure SES Model on Corrected parameters

Model evaluation:



# **Method 6: Double Exponential Smoothing (Holt's Model)**

Holt - ETS(A, A, N) - Holt's linear method with additive errors , Double Exponential Smoothing.

One of the drawbacks of the simple exponential smoothing is that the model does not do well in the presence of the trend. This model is an extension of SES known as Double Exponential model which estimates two smoothing parameters. This is applicable when data has Trend but no seasonality. In this model two separate components are considered: **Level** and **Trend**. **Level** is the local mean.

One smoothing parameter α corresponds to the level series.

A second smoothing parameter β corresponds to the trend series.

Double Exponential Smoothing uses two equations to forecast future values of the time series, one for forecasting the short-term average value or level and the other for capturing the trend.

Intercept or Level equation, 𝐿𝑡 is given by: 𝐿𝑡=𝛼𝑌𝑡+(1−𝛼)𝐹𝑡

Trend equation is given by 𝑇𝑡=𝛽(𝐿𝑡−𝐿𝑡−1)+(1−𝛽)𝑇𝑡−1

Here, 𝛼 and 𝛽 are the smoothing constants for level and trend, respectively,

0 < 𝛼 < 1 and 0 < 𝛽 < 1.

The forecast at time t + 1 is given by

𝐹𝑡+1=𝐿𝑡+𝑇𝑡

𝐹𝑡+𝑛=𝐿𝑡+𝑛𝑇𝑡

Two parameters 𝛼 and 𝛽 are estimated in this model. Level and Trend are accounted for in this model

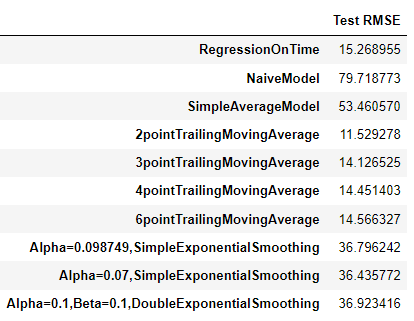
Similar to Above SES model, we will calculate Alpha and beta values by using the for loops and calculate the RMSE value. Then we would choose Alpha and beta value for the Lowest RMSE.





Figure DES model

We see that the double exponential smoothing is picking up the trend component along with the level component as well.



Inference

Here, we see that the Double Exponential Smoothing has actually done well when compared to the Simple Exponential Smoothing. This is because of the fact that the Double Exponential Smoothing model has picked up the trend component as well.

The Holt's model in Python has certain other options of exponential trends or whether the smoothing parameters should be damped. You can try these out later to check whether you get a better forecast.

# **Method 7: Triple Exponential Smoothing (Holt - Winter's Model)**

Three parameters 𝛼 , 𝛽 and 𝛾 are estimated in this model. Level, Trend and Seasonality are accounted for in this model. First of all build the model with default parameters:

{'smoothing\_level': 0.07736040004765096,

'smoothing\_trend': 0.03936496779735522,

'smoothing\_seasonal': 0.0008375039104357999,

'damping\_trend': nan,

'initial\_level': 156.90674503596637,

'initial\_trend': -0.9061396720042346,

'initial\_seasons': array([0.7142168 , 0.80982439, 0.88543128, 0.77363782, 0.87046319,

0.94699283, 1.04196135, 1.11012703, 1.04835489, 1.0276963 ,

1.19783562, 1.6514144 ]),

'use\_boxcox': False,

'lamda': None,

'remove\_bias': False}

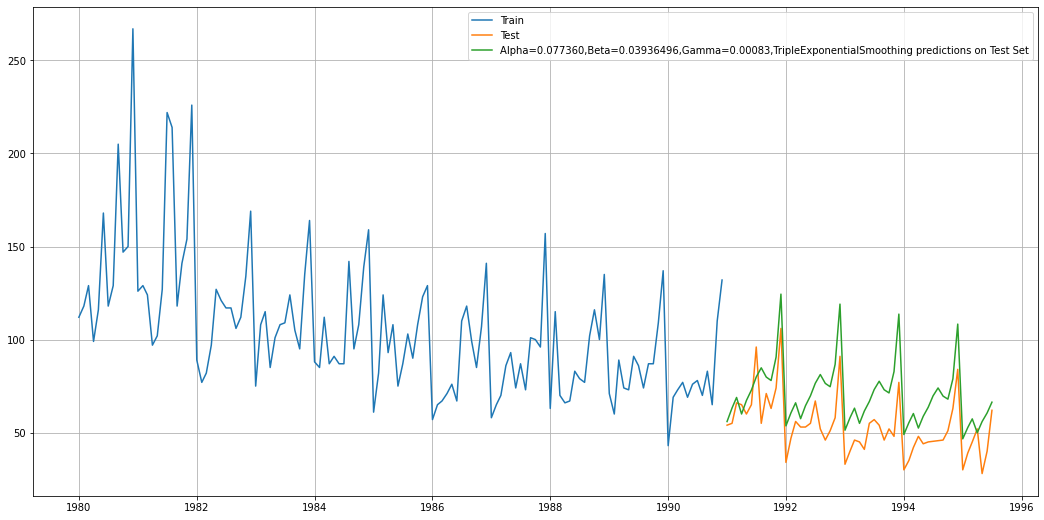


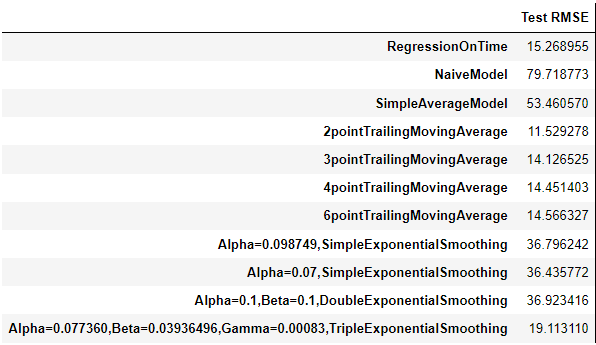
Figure TES Model on Auto Parameters

We see that the Triple Exponential Smoothing is picking up the seasonal component as well.

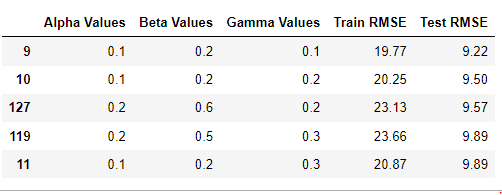
Inference

Triple Exponential Smoothing has performed the best on the test as expected since the data had both trend and seasonality.

But we see that our triple exponential smoothing is under forecasting. Let us try to tweak some of the parameters in order to get a better forecast on the test set.



Auto generated Alpha , beta and game values and decide best based on lowest RMSE values



Final values are chosen based as 0.1, 0.2 and0.1 on lowest RMSE value of 9.22 .

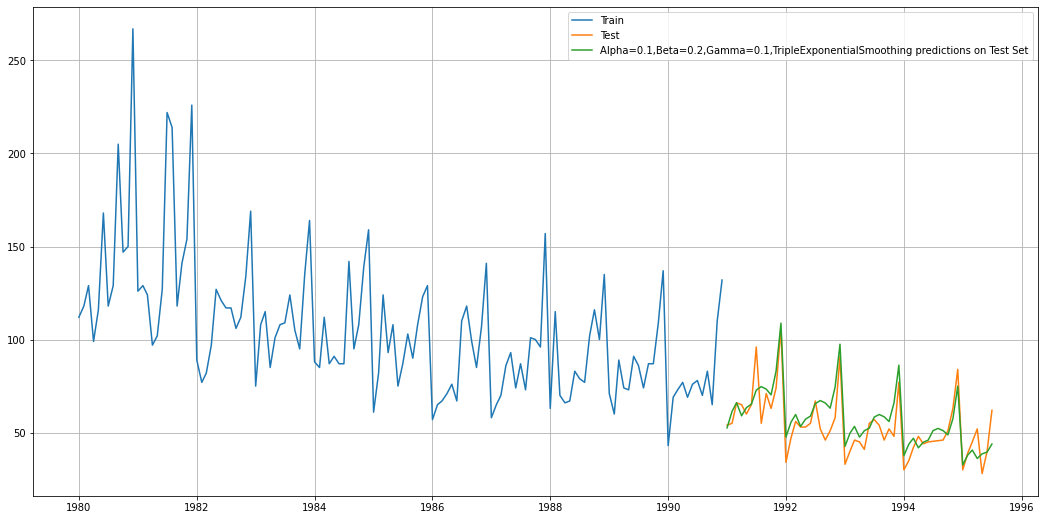
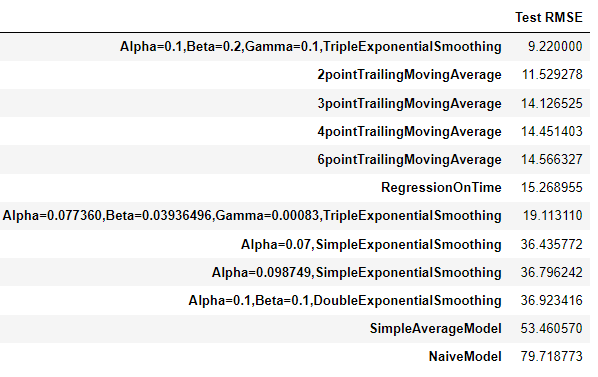


Figure TES Model on Corrected parameters

Compare the Model generated so far:



We see that the best model is the Triple Exponential Smoothing with multiplicative seasonality with the parameters 𝛼 = 0.1 , 𝛽 = 0.2 and 𝛾 = 0.1.

For this data, we had both trend and seasonality so by definition Triple Exponential Smoothing is supposed to work better than the Simple Exponential Smoothing as well as the Double Exponential Smoothing.

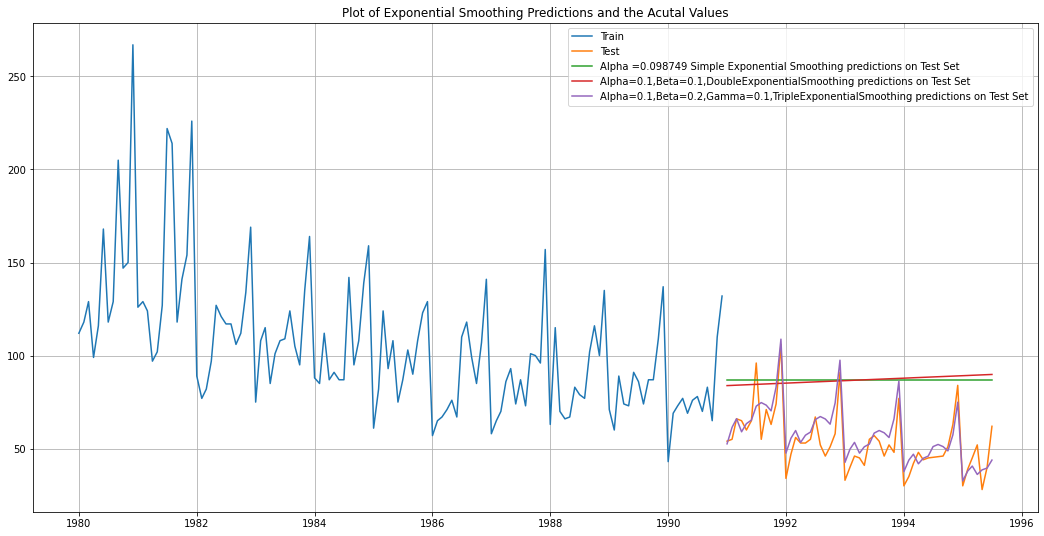


Figure SES, DES and TES Model together comparison on Test data

In this particular we have built several models and went through a model building exercise. This particular exercise has given us an idea as to which particular model gives us the least error on our test set for this data. But in Time Series Forecasting, we need to be very vigil about the fact that after we have done this exercise we need to build the model on the whole data. Remember, the training data that we have used to build the model stops much before the data ends. In order to forecast using any of the models built, we need to build the models again (this time on the complete data) with the same parameters. For this particular mentored learning session, we will go ahead and build only the top 1 model which gave us the best accuracy (least RMSE). The two models to be built on the whole data are the following:

Alpha=0.4,Beta=0.1,Gamma=0.3,TripleExponentialSmoothing

Alpha=0.111,Beta=0.049,Gamma=0.362,TripleExponentialSmoothing

# **Check for the stationarity of the data on which the model is being built on using appropriate statistical tests and also mention the hypothesis for the statistical test. If the data is found to be non-stationary, take appropriate steps to make it stationary. Check the new data for stationarity and comment.**

Note: Stationarity should be checked at alpha = 0.05

A time series has stationarity when the observations are not dependent on the time. Statistical properties of these time series will not change with time thus they will have constant mean, variance, and covariance.

The time series which have trends or with seasonality, are not stationary. Because trends will have a change in the movement of data concerning time which will cause the change in mean over time. Whereas seasonality occurs when the pattern in time series shows a variation for a regular time interval which will cause the variance to change over time.

Stationarity of time series can be detected by: Visually Plotting the time series and check for trend or seasonality. By Splitting time series into the different partitions and compare the statistical inference.

We can also perform Augmented Dickey-Fuller test to check the stationarity.

The Augmented Dickey-Fuller test is a unit root test which determines whether there is a unit root and subsequently whether the series is non-stationary.

**The hypothesis in a simple form for the ADF test is:**

**𝐻0H0 : The Time Series has a unit root and is thus non-stationary.**

**𝐻1H1 : The Time Series does not have a unit root and is thus stationary.**

We would want the series to be stationary for building ARIMA models and thus we would want the p-value of this test to be less than the 𝛼α value. We have performed ADF test and following are the results for the same:

DF test statistic is -2.240

DF test p-value is **0.4671371627793189**

Number of lags used 13

Since P value for above test is greater than 0.05 , which is 0.4671371627793189, so we are fail to reject the Null hypothesis and we accept that it is a Non stationary Time series. We see that at 5% significant level the Time Series is non-stationary.

There are various ways that Python allows us to select the appropriate number of lags at which we check whether the Time Series is stationary. To know more about the how to select the various ways, please refer to the link over ℎ𝑒𝑟𝑒.

Let us take one level of differencing to see whether the series becomes stationary.

DF test statistic is -8.162

DF test p-value is 3.015976115827045e-11

Number of lags used 12

Since P value is way less than 0.05 so going back by one level of differencing will make time series as Stationary. Now, let us go ahead and plot the stationary series.

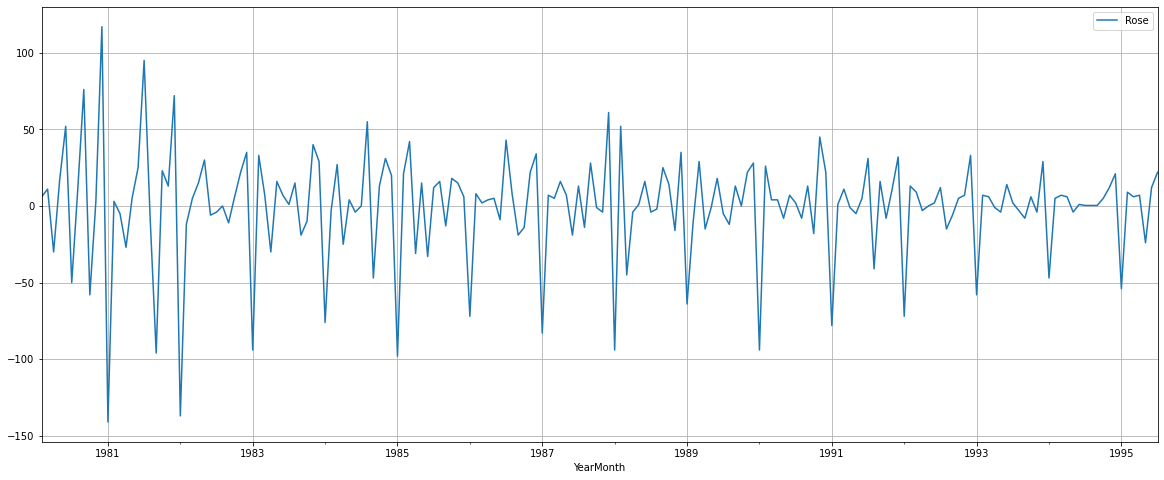


Figure Stationary Time series

Also, if the series is non-stationary, stationaries the Time Series by taking a difference of the Time Series. Then we can use this particular differenced series to train the ARIMA models. We do not need to worry about stationarity for the Test Data because we are not building any models on the Test Data, we are evaluating our models over there. You can look at other kinds of transformations as part of making the time series stationary like taking logarithms

# **Build an automated version of the ARIMA/SARIMA model in which the parameters are selected using the lowest Akaike Information Criteria (AIC) on the training data and evaluate this model on the test data using RMSE**

# **ARIMA Model:**

An ARIMA and SARIMA models are class of statistical models for analyzing and forecasting time series data. Lets break it down :

* AR: Autoregression. A model that uses the dependent relationship between an observation and some number of lagged observations.
* I: Integrated. The use of differencing of raw observations in order to make the time series stationary.
* MA: Moving Average. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

The parameters of the ARIMA model are defined as follows:

* p: The number of lag observations included in the model, also called the lag order.
* d: The number of times that the raw observations are differenced, also called the degree of differencing.
* q: The size of the moving average window, also called the order of moving average.

The main assumption of AR model is that the time series data is stationary.

A stationary time series is one whose statistical properties such as mean, variance, autocorrelation, etc. are all constant over time.

When the time series data is not stationary, then we convert the non-stationary data before applying AR models. Method we used for making timeseries as Stationary is : Taking the difference between consecutive observations, we also call it a lag-1 difference. For time series with a seasonal component, the lag may be expected to be the period (width) of the seasonality.

White noise of the residuals:

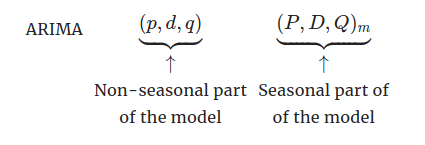
White noise is a process of residuals ϵt that are uncorrelated and follow normal distribution with mean 0 and constant standard deviation. In AR models, one of the main assumptions is the errors follow a white noise.

# **SARIMA Model:**

The difference between ARIMA and SARIMA (SARIMAX) is about the seasonality of the dataset. if your data is seasonal, like it happens after a certain period of time. then we will use SARIMA.

SARIMA stands for Seasonal-ARIMA and it includes seasonality contribution to the forecast. The importance of seasonality is quite evident and ARIMA fails to encapsulate that information implicitly.

The Autoregressive (AR), Integrated (I), and Moving Average (MA) parts of the model remain as that of ARIMA. The addition of Seasonality adds robustness to the SARIMA model. It’s represented as:

[Source](https://otexts.com/fpp2/seasonal-arima.html)

where m is the number of observations per year. We use the uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model.

Similar to ARIMA, the P,D,Q values for seasonal parts of the model can be deduced from the ACF and PACF plots of the data. Let’s implement SARIMA for the same Catfish sales model.

Both ARIMA and SARIMA can be build using Automated way of generating values p,d and q value or manual way of building ACF / PACF graph and observe the numbers for p , d and q.

# **Auto ARIMA Model:**

Since we have already seen that Stationarity can be achieved with Lag of 1 . So d value is 1. And We have taken chosen default value for p and q between 0 and 4, And then generated all possible combination for p,d and q. There are the parameters we have chosen for testing our ARIMA Model and gathered AIC value for all the models:

Examples of the parameter combinations for the Model

Model: (0, 1, 0)

Model: (0, 1, 1)

Model: (0, 1, 2)

Model: (0, 1, 3)

Model: (1, 1, 0)

Model: (1, 1, 1)

Model: (1, 1, 2)

Model: (1, 1, 3)

Model: (2, 1, 0)

Model: (2, 1, 1)

Model: (2, 1, 2)

Model: (2, 1, 3)

Model: (3, 1, 0)

Model: (3, 1, 1)

Model: (3, 1, 2)

Model: (3, 1, 3)

Following are the AIC values for above listed all parameters , when we fit Timeseries data into ARIMA model :

ARIMA(0, 1, 0) - AIC:1333.1546729124348

ARIMA(0, 1, 1) - AIC:1282.3098319748315

ARIMA(0, 1, 2) - AIC:1279.6715288535752

ARIMA(0, 1, 3) - AIC:1280.5453761734652

ARIMA(1, 1, 0) - AIC:1317.3503105381526

ARIMA(1, 1, 1) - AIC:1280.574229538006

ARIMA(1, 1, 2) - AIC:1279.870723423191

ARIMA(1, 1, 3) - AIC:1281.8707223309964

ARIMA(2, 1, 0) - AIC:1298.6110341604958

ARIMA(2, 1, 1) - AIC:1281.5078621868606

ARIMA(2, 1, 2) - AIC:1281.8707222264304

ARIMA(2, 1, 3) - AIC:1274.6951493753345

ARIMA(3, 1, 0) - AIC:1297.481091727174

ARIMA(3, 1, 1) - AIC:1282.4192776271934

ARIMA(3, 1, 2) - AIC:1283.720740597716

ARIMA(3, 1, 3) - AIC:1278.6699617388035

Then we have sported the data, based on AIC value and least value of AIC have following records:

|  |  |  |
| --- | --- | --- |
|  | param | AIC |
| 11 | (2, 1, 3) | 1274.695149 |
| 15 | (3, 1, 3) | 1278.669962 |
| 2 | (0, 1, 2) | 1279.671529 |
| 6 | (1, 1, 2) | 1279.870723 |
| 3 | (0, 1, 3) | 1280.545376 |

Lets build the SARIMAX report for the Best parameter (2,1,3)

SARIMAX Results

==============================================================================

Dep. Variable: Rose No. Observations: 132

Model: ARIMA(2, 1, 3) Log Likelihood -631.348

Date: Thu, 25 Aug 2022 AIC 1274.695

Time: 22:11:02 BIC 1291.946

Sample: 01-01-1980 HQIC 1281.705

- 12-01-1990 Covariance Type: opg

==============================================================================

coef std err z P>|z| [0.025 0.975]

------------------------------------------------------------------------------

ar.L1 -1.6777 0.084 -20.037 0.000 -1.842 -1.514

ar.L2 -0.7285 0.084 -8.701 0.000 -0.893 -0.564

ma.L1 1.0445 0.650 1.606 0.108 -0.230 2.319

ma.L2 -0.7724 0.134 -5.772 0.000 -1.035 -0.510

ma.L3 -0.9049 0.590 -1.533 0.125 -2.061 0.252

sigma2 858.9672 547.433 1.569 0.117 -213.981 1931.916

==============================================================================

Ljung-Box (L1) (Q): 0.02 Jarque-Bera (JB): 24.48

Prob(Q): 0.88 Prob(JB): 0.00

Heteroskedasticity (H): 0.40 Skew: 0.71

Prob(H) (two-sided): 0.00 Kurtosis: 4.57

==============================================================================

Warnings:

[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Lets Build the Diagnostic Plot:

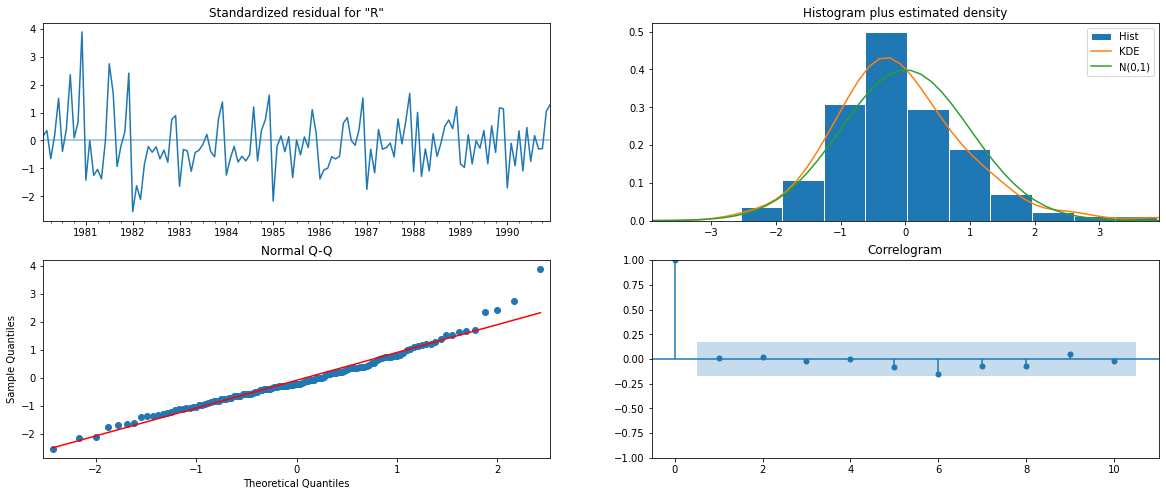


Figure Diagnostic Plot for Auto ARIMA

Lets find the RMSE Value for Auto ARIMA with p, d, q value of 2,1,3:

|  |  |
| --- | --- |
|  | **RMSE** |
| **Auto\_ARIMA(2,1,3)** | 36.809324 |

# **Auto SARIMA:**

Similar to Auto Arima Model, We have taken all possible combination of p,d,q and P,D,Q values for SARIMA Model, along with the Seasonality parameter. We have taken these combinations:

Examples of the parameter combinations for the Model are

Model: (0, 1, 1)(0, 0, 1, 6)

Model: (0, 1, 2)(0, 0, 2, 6)

Model: (0, 1, 3)(0, 0, 3, 6)

Model: (1, 1, 0)(1, 0, 0, 6)

Model: (1, 1, 1)(1, 0, 1, 6)

Model: (1, 1, 2)(1, 0, 2, 6)

Model: (1, 1, 3)(1, 0, 3, 6)

Model: (2, 1, 0)(2, 0, 0, 6)

Model: (2, 1, 1)(2, 0, 1, 6)

Model: (2, 1, 2)(2, 0, 2, 6)

Model: (2, 1, 3)(2, 0, 3, 6)

Model: (3, 1, 0)(3, 0, 0, 6)

Model: (3, 1, 1)(3, 0, 1, 6)

Model: (3, 1, 2)(3, 0, 2, 6)

Model: (3, 1, 3)(3, 0, 3, 6)

Then we fit our TS data into SARIMA model to calculate the AIC value and sorted AIC values. Following is Sorted AIC value achieved with p,d,q values:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **param** | **seasonal** | **AIC** |
| **187** | (2, 1, 3) | (2, 0, 3, 6) | 951.744297 |
| **59** | (0, 1, 3) | (2, 0, 3, 6) | 952.073632 |
| **251** | (3, 1, 3) | (2, 0, 3, 6) | 952.582102 |
| **191** | (2, 1, 3) | (3, 0, 3, 6) | 953.205616 |
| **123** | (1, 1, 3) | (2, 0, 3, 6) | 953.684951 |

So the Best parameter for SARIMA Model will be (2,1,3) (2,0,3,6) . RMSE and MEP values are:

RMSE: 27.124535166501488

MAPE: 55.23994932700531

SARIMAX Results

==========================================================

Dep. Variable: Rose No. Observations: 132

Model: SARIMAX(2, 1, 3)x(2, 0, 3, 6) Log Likelihood -464.872

Date: Thu, 25 Aug 2022 AIC 951.744

Time: 22:13:31 BIC 981.349

Sample: 01-01-1980 HQIC 963.750

- 12-01-1990

Covariance Type: opg

==============================================================================

coef std err z P>|z| [0.025 0.975]

------------------------------------------------------------------------------

ar.L1 -0.5028 0.083 -6.082 0.000 -0.665 -0.341

ar.L2 -0.6628 0.084 -7.919 0.000 -0.827 -0.499

ma.L1 -0.3714 1869.343 -0.000 1.000 -3664.216 3663.474

ma.L2 0.2033 1175.137 0.000 1.000 -2303.023 2303.430

ma.L3 -0.8320 1555.238 -0.001 1.000 -3049.043 3047.379

ar.S.L6 -0.0838 0.049 -1.720 0.085 -0.179 0.012

ar.S.L12 0.8099 0.052 15.465 0.000 0.707 0.913

ma.S.L6 0.1701 0.248 0.687 0.492 -0.315 0.656

ma.S.L12 -0.5646 0.199 -2.837 0.005 -0.955 -0.174

ma.S.L18 0.1710 0.143 1.198 0.231 -0.109 0.451

sigma2 260.7967 4.88e+05 0.001 1.000 -9.55e+05 9.56e+05

==============================================================================

Ljung-Box (L1) (Q): 0.72 Jarque-Bera (JB): 4.77

Prob(Q): 0.40 Prob(JB): 0.09

Heteroskedasticity (H): 0.54 Skew: -0.36

Prob(H) (two-sided): 0.06 Kurtosis: 3.73

==============================================================================

Warnings: [1] Covariance matrix calculated using the outer product of gradients (complex-step). [2] Covariance matrix is singular or near-singular, with condition number 5.98e+14. Standard errors may be unstable.

**Diagnostics Plot:**

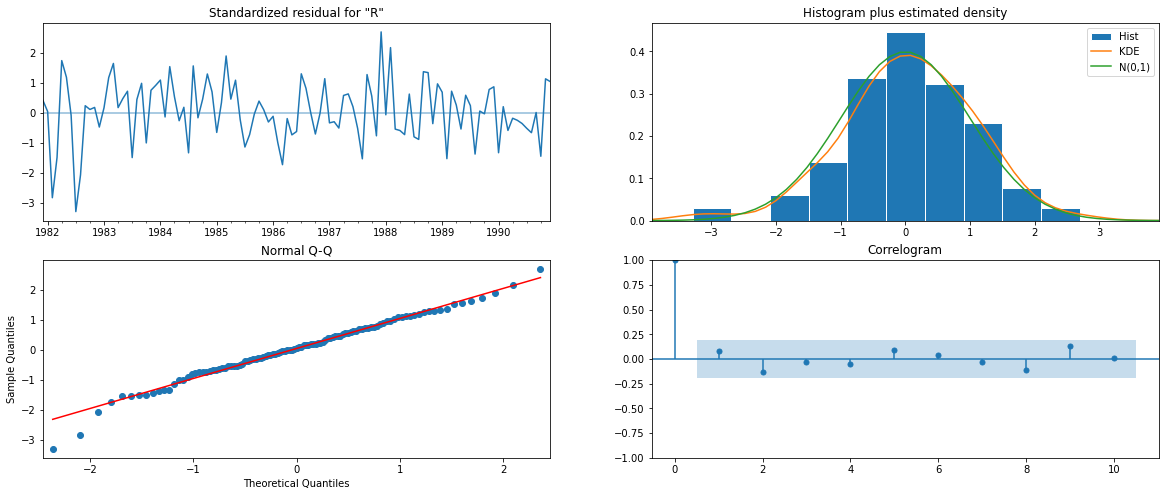


Figure Auto SARIMA Model Diagnostic Plot

# **Build ARIMA/SARIMA models based on the cut-off points of ACF and PACF on the training data and evaluate this model on the test data using RMSE.**

In the above question we build Automated version of ARIMA/SARIMA model, in that we have used all possible combinations of p,d and q values and generated Chosen AIC, RMSE and MEP parameters to identify, which combination of pdq is good for the timeseries data.

One problem with Auto version is, it needs lot of computations and we should have good memory and processing power in the system for iterating all possible values. So to avoid that scenario , we can use building ACF/PACF graph for generating Manual version of ARIMA/SARIMA Model.

Important Component of Automated version of ARIMA Model :

**Auto-Correlation Function (ACF)** or correlogram : A plot of auto-correlation of different lags is called ACF. The plot summarizes the correlation of an observation with lag values. The x-axis shows the lag and the y-axis shows the correlation coeﬃcient between -1 and 1 for negative and positive correlation.

**Partial Auto-Correlation Function (PACF)** Autocorrelation Function (ACF) : A plot of partial auto-correlation for different values of lags is called PACF.The plot summarizes the correlations for an observation with lag values that is not accounted for by prior lagged observations.

Both plots are drawn as bar charts showing the 95% and 99% conﬁdence intervals as horizontal lines. Bars that cross these conﬁdence intervals are therefore more signiﬁcant and worth noting.

Some useful patterns you may observe on these plots are:

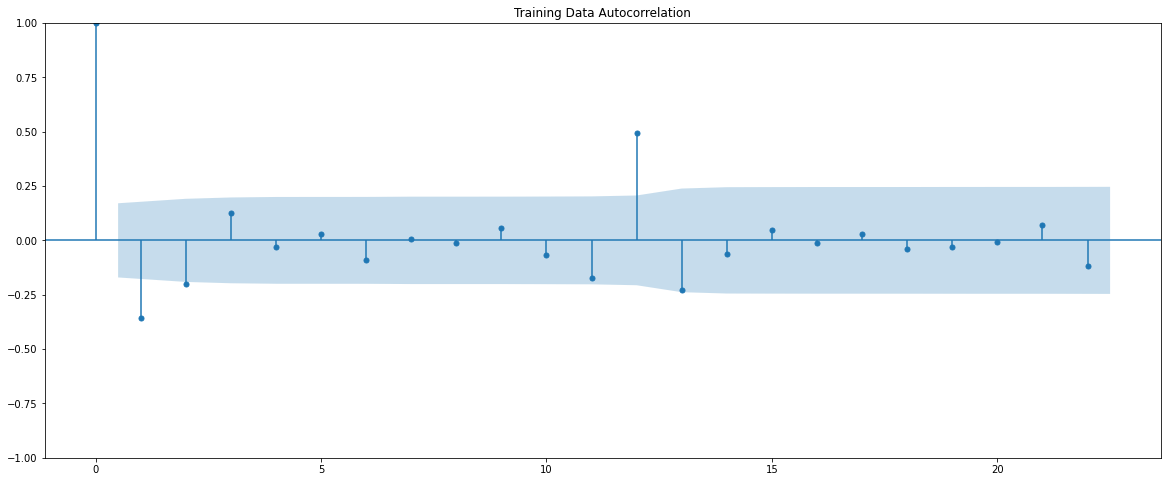
The number of lags is p when:

* The partial auto-correlation, | ρpk | > 1.96 / n−−√ for first p values and cuts off to zero. The auto-correlation function, ρk decreases exponentially.
* The model is AR of order p when the PACF cuts-off after a lag p.
* The model is MA of order p when the ACF cuts-off after a lag q.
* The model is a mix of AR and MA if both the PACF and ACF trail oﬀ and cuts-off at p and q respectively.

For an ARIMA (p,d,q) process, it becomes non-stationary to stationary after differencing it for d times

# **Build Manual ARIMA Model**

ACF graph for ARIMA Model:



PACF Graph for Arima Model

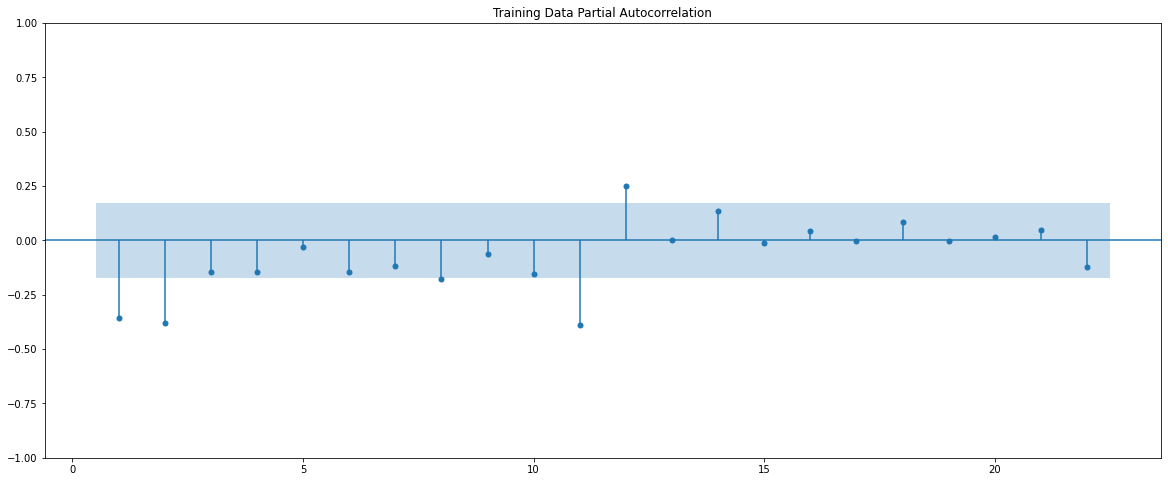


Figure Manual ARIMA ACF/PACF Plot

Here, we have taken alpha=0.05.

* The Auto-Regressive parameter in an ARIMA model is 'p' which comes from the significant lag before which the PACF plot cuts-off to 2.
* The Moving-Average parameter in an ARIMA model is 'q' which comes from the significant lag before the ACF plot cuts-off to 2.

By looking at the above plots, we will take the value of p and q to be 2 and 2 respectively.

SARIMAX Results

==============================================================================

Dep. Variable: Rose No. Observations: 132

Model: ARIMA(2, 1, 2) Log Likelihood -635.935

Date: Thu, 25 Aug 2022 AIC 1281.871

Time: 22:11:03 BIC 1296.247

Sample: 01-01-1980 HQIC 1287.712

- 12-01-1990 Covariance Type: opg

==============================================================================

coef std err z P>|z| [0.025 0.975]

------------------------------------------------------------------------------

ar.L1 -0.4540 0.469 -0.969 0.333 -1.372 0.464

ar.L2 0.0001 0.170 0.001 0.999 -0.334 0.334

ma.L1 -0.2541 0.459 -0.554 0.580 -1.154 0.646

ma.L2 -0.5984 0.430 -1.390 0.164 -1.442 0.245

sigma2 952.1601 91.424 10.415 0.000 772.973 1131.347

===================================================================================

Ljung-Box (L1) (Q): 0.02 Jarque-Bera (JB): 34.16

Prob(Q): 0.88 Prob(JB): 0.00

Heteroskedasticity (H): 0.37 Skew: 0.79

Prob(H) (two-sided): 0.00 Kurtosis: 4.94

==================================================================

Warnings: [1] Covariance matrix calculated using the outer product of gradients (complex-step)

Diagnostic Plot:

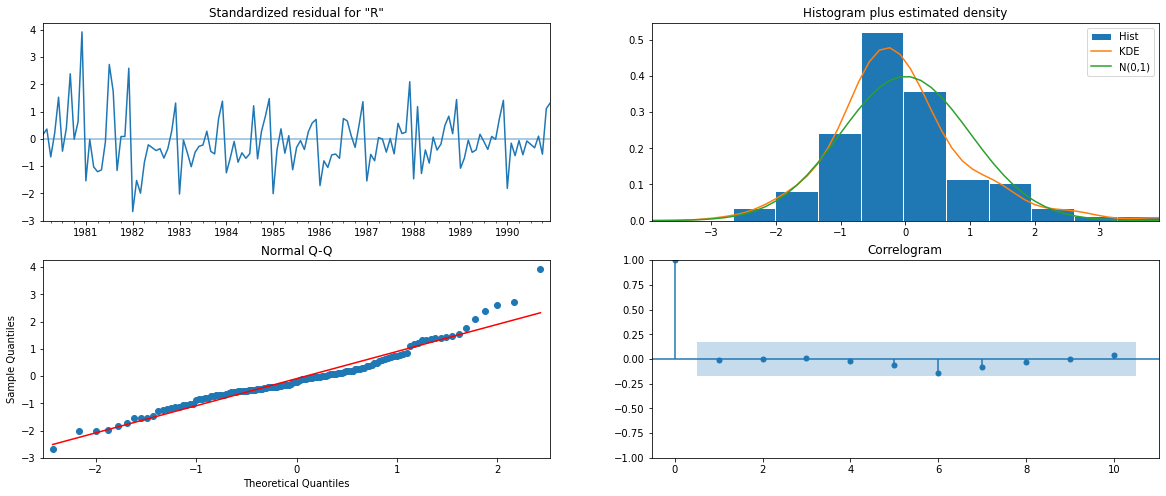


Figure Manual ARIMA Diagnostic Plot

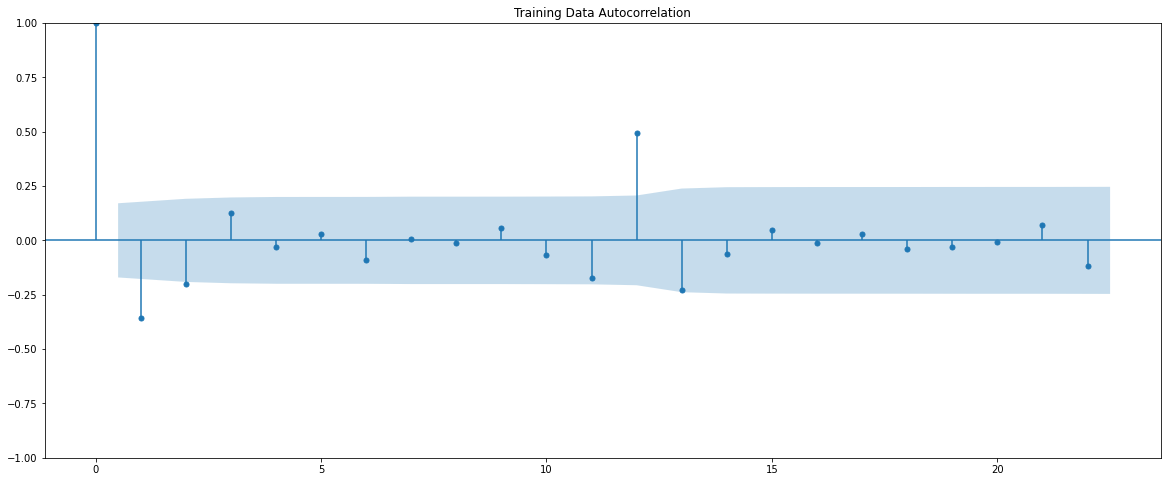
#### Predict on the Test Set using this model and evaluate the model

RMSE: 36.87119662176807

MAPE: 76.05621272229534

# **Build Manual SARIMA Model:**

We will star from where we left in ARIMA Model, from the above manual version of ARMIA model, we had chosen value for p,d,q as (2,1,2). So we will choose same for SARIMA Model and then add the parameter for Seasonality . Lets plot ACF and PACF plot one more time:



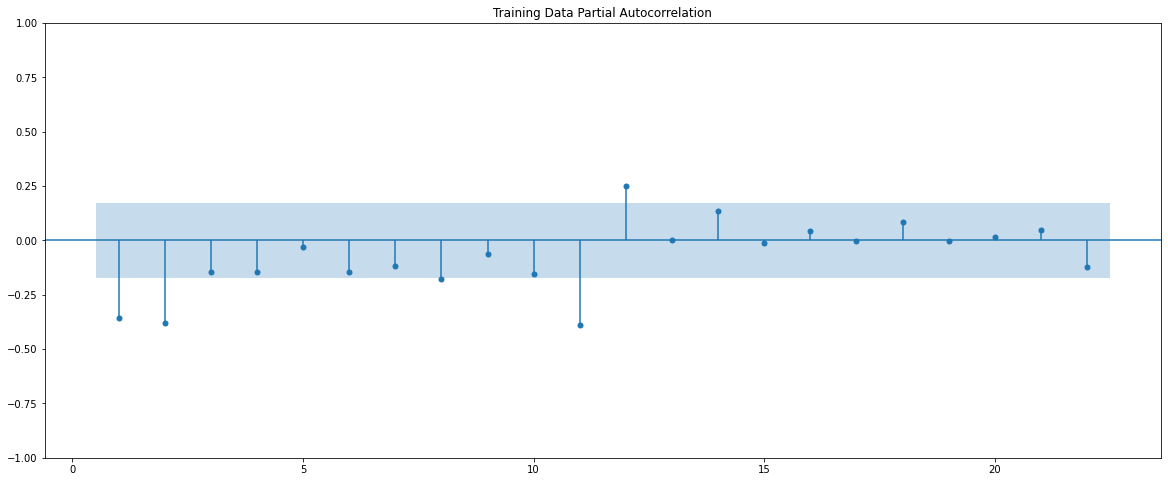


Figure Manual SARIMA ACF/PACF Plot

Here, we have taken alpha=0.05. We can not see that there is a seasonality. and P value and Q value would be 0 from above graphs.

We are going to take the seasonal period as 3 or its multiple e.g. 6. We are taking the p value to be 3 and the q value also to be 3 as the parameters same as the ARIMA model.

The Auto-Regressive parameter in an SARIMA model is 'P' which comes from the significant lag after which the PACF plot cuts-off to 0.

The Moving-Average parameter in an SARIMA model is 'Q' which comes from the significant lag after which the ACF plot cuts-off to 0.

So final values are as follows: (2,1,2)(0,1,0,6) . Lets build the Sarimax result:

SARIMAX Results

===========================================================================

Dep. Variable: Rose No. Observations: 132

Model: SARIMAX(2, 1, 2)x(0, 1, [], 6) Log Likelihood -634.370

Date: Thu, 25 Aug 2022 AIC 1278.739

Time: 22:11:04 BIC 1292.759

Sample: 01-01-1980 HQIC 1284.434

- 12-01-1990

Covariance Type: opg

==============================================================================

coef std err z P>|z| [0.025 0.975]

------------------------------------------------------------------------------

ar.L1 1.1123 0.055 20.150 0.000 1.004 1.220

ar.L2 -0.3756 0.029 -13.012 0.000 -0.432 -0.319

ma.L1 -1.9944 0.114 -17.561 0.000 -2.217 -1.772

ma.L2 1.0000 0.114 8.786 0.000 0.777 1.223

sigma2 1774.6506 0.000 1.39e+07 0.000 1774.650 1774.651

===================================================================

Ljung-Box (L1) (Q): 0.32 Jarque-Bera (JB): 4.04

Prob(Q): 0.57 Prob(JB): 0.13

Heteroskedasticity (H): 0.37 Skew: -0.24

Prob(H) (two-sided): 0.00 Kurtosis: 3.76

==============================================================

Warnings: [1] Covariance matrix calculated using the outer product of gradients (complex-step). [2] Covariance matrix is singular or near-singular, with condition number 1.8e+22. Standard errors may be unstable.

Diagnostics Plot:

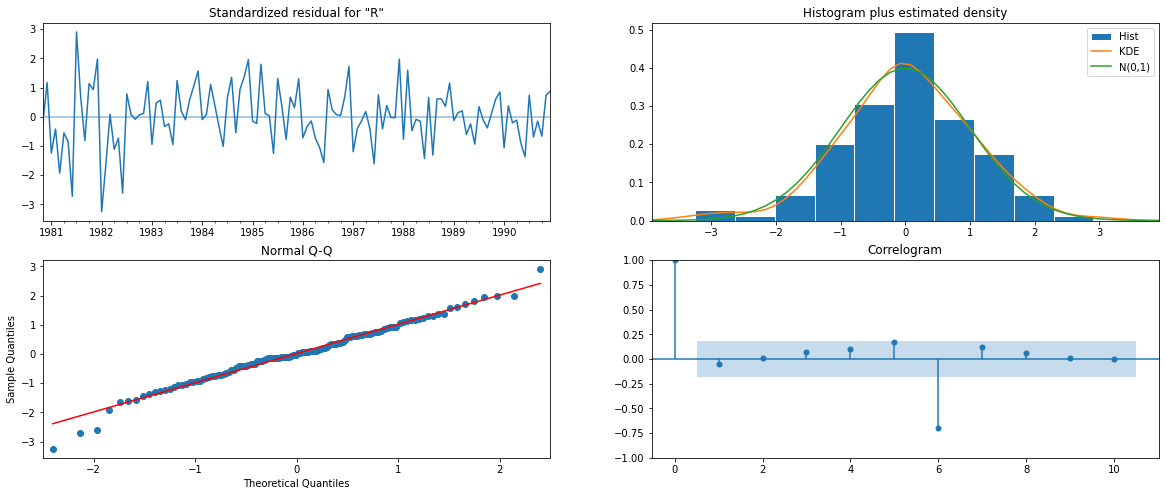


Figure Manual SAIRMA Diagnostic Plot

Calculate the RMSE and MAPE value for test data for Manual SARIMA Model:

RMSE: 31.65138712114397

MAPE: 51.711250860674

# **Build a table with all the models built along with their corresponding parameters and the respective RMSE values on the test data**

We have built several Models, with different parameters, and calculated the RMSE values for all the parameters . After combining all RMSE values for all the models, following is the table for that:

|  |  |
| --- | --- |
| **Model name with Parameters** | **Test RMSE** |
| Alpha=0.1,Beta=0.2,Gamma=0.1,TripleExponentialSmoothing | 9.22 |
| 2pointTrailingMovingAverage | 11.529278 |
| 3pointTrailingMovingAverage | 14.126525 |
| 4pointTrailingMovingAverage | 14.451403 |
| 6pointTrailingMovingAverage | 14.566327 |
| RegressionOnTime | 15.268955 |
| Alpha=0.077360,Beta=0.03936496,Gamma=0.00083,TripleExponentialSmoothing | 19.11311 |
| Auto\_SARIMA(2,1,3)(2, 0, 3, 6) | 27.124535 |
| manual\_SARIMA(2,1,2)(0,1,0,6) | 31.651387 |
| Alpha=0.07,SimpleExponentialSmoothing | 36.435772 |
| Alpha=0.098749,SimpleExponentialSmoothing | 36.796242 |
| Auto\_ARIMA(2,1,3) | 36.809324 |
| Manual\_ARIMA(2,1,2) | 36.871197 |
| Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing | 36.923416 |
| SimpleAverageModel | 53.46057 |
| NaiveModel | 79.718773 |

# **Based on the model-building exercise, build the most optimum model(s) on the complete data and predict 12 months into the future with appropriate confidence intervals/bands.**

Based on the table above, we have see that best model so far given least value of RMSE is Triple exponential Model with parameters of Alpha=0.1,Beta=0.2,Gamma=0.1. It’s RMSE values is only 9.22 as compared to the worst RMSE value of 79.718773 of Naïve based Model .

Lets build the Triple exponential Smoothing model based on above parameters and fit full data set , we were using only test data for the predictions earlier. We have also calculated the RMSE value for full data set .

RMSE of the Full Model 17.023705516791587

Also as required, we have predicted nest 12 months of data for using above model:

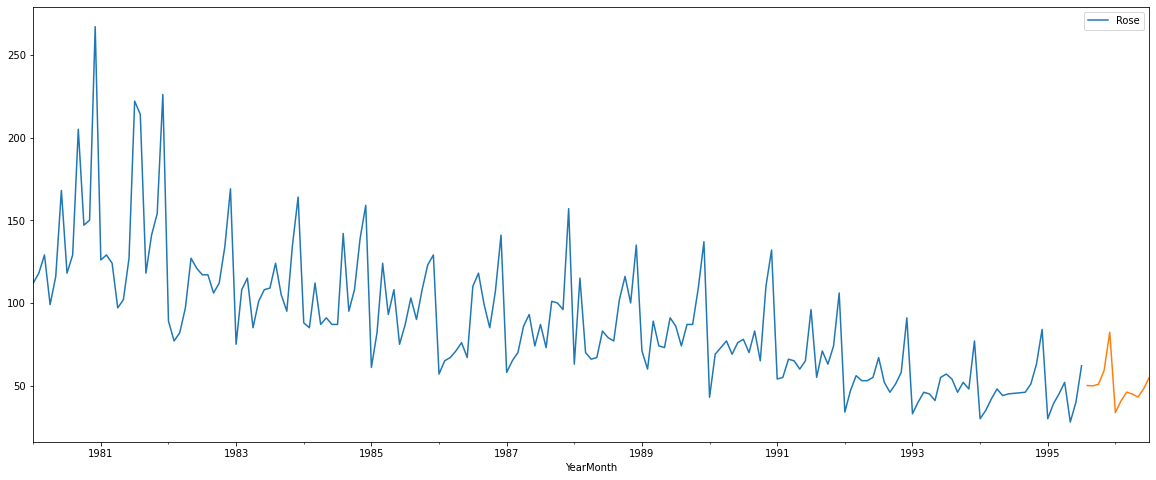


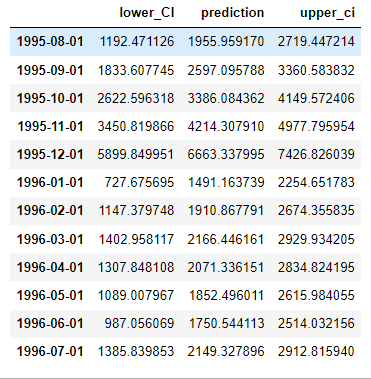
Figure Best Model TES plot for next 12 Months predictions

Plotting the forecast with the confidence band.

**Confidence band for forecasting**: A confidence band is used in statistical analysis to represent the uncertainty in an estimate of a curve or function based on limited or noisy data. The 95% confidence bands enclose the area that you can be 95% sure contains the true curve. It gives you a visual sense of how well your data define the best-fit curve. It is closely related to the 95% prediction bands , which enclose the area that you expect to enclose 95% of future data points. This includes both the uncertainty in the true position of the curve (enclosed by the confidence bands), and also accounts for scatter of data around the curve. Therefore, prediction bands are always wider than confidence bands

For building the Confidence band we need to find lower and upper values of actual predictions with 95% confidentiality . In our case

Following are the Table for Predictions of upper and lower values along with Predicted values:



Plot the time series with next 12 months of unseen data :

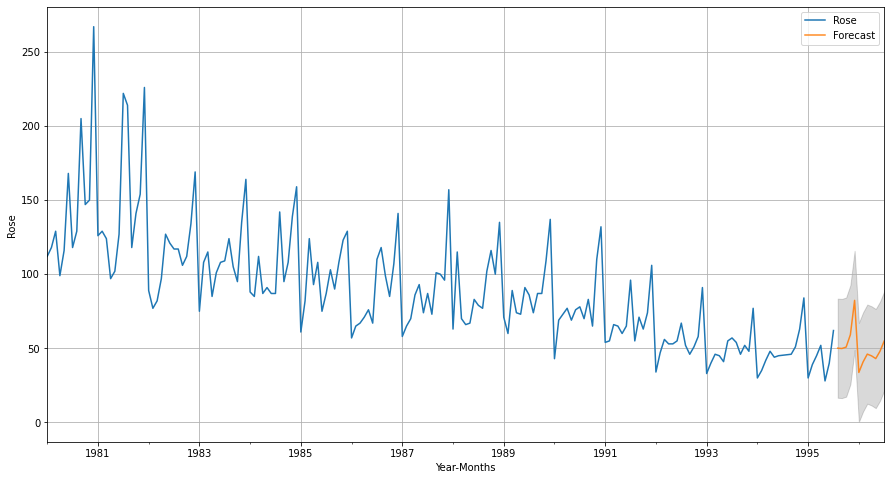


Figure TES Model next 12 Month Predictions with Confidence Band

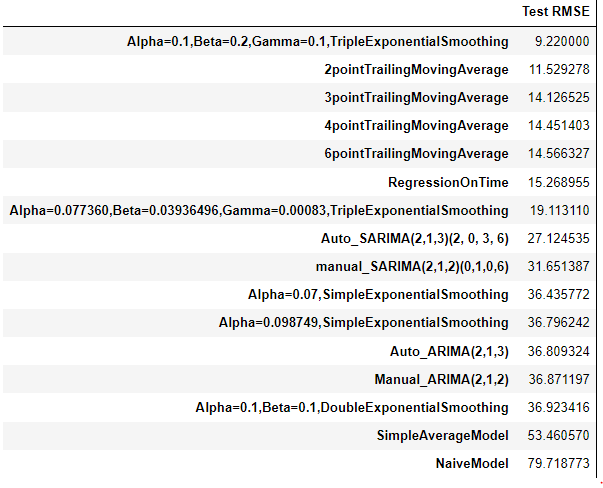
# **Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales.**

Analysis Rose wine productions Sales data . These are the observations about the data:

1. Data consist of 187 data points
2. It seems to be contained seasonality
3. We also notice the fluctuations in the trend in the initial years and slowly decreasing the following years.
4. Minimum sales for the data in any month is 28 and Max sales of Rose wines in any month is 267
5. Year 1981 has highest sales and 1995 has lowest sales
6. We see the Big increase in Sales for Rose wine in year 1981 and soon after that started decreasing Also there were years with Rise and fall equally. But gradually Sales goes down.
7. We see there are outliers in almost all the year as per the box plot.
8. Average sales is lowest in the year of 1995
9. Monthly plot contains outliers in the month of June, July, August, September and December.
10. There are Highest sales in the Month of December followed by 2nd highest Sales in November Month, which indicates year end party and vacation celebration sales
11. We see that there are High Sales in each month at the month Start and then Fall down till Mid of the month,
12. We also see small increase in Sales in every Mid of the Month and then again goes down till Month End.
13. It clearly indicates, when Anyone gets the Salary at the Start of the month and sometimes biweekly, then Sales for Month Start is always High and small Rise in Mid of the Month.
14. There are 2 Months data not available in July and August 1994 and then no data after July 1995.
15. The plot shows that in 1982 there is a fall in the wine sales and there is a steep downfall is observed.
16. The resampled yearly or annual series have smoothened out the seasonality and have only been able to capture the year on year trend where there was.
17. There is some rise is found in Year 1984 and 1988, but ultimately that also goes down.
18. We see that the year on year quarterly series represents the year on year monthly series. The quarterly series is able to catch the seasonality in the data.
19. If we take the resampling period to be 10 years or a decade, we see that the seasonality present has been smoothed over and it is only giving an estimate of the trend.
20. Sales trend is going down year by year
21. We have done up sampling of the data for Quarterly, yearly and a Decade sale, there are seasonality in the data, and it gets flatter out when we Up sample the data
22. We have also done Down sampling of the data for analyzing Daily Sales.
23. Correlogram, histogram, residual and quartiles were plotted.

Forecasting analysis:

We have built 17 Models on our data and Compared all of them based on RMSE value for each Model: Results with the parameters used for comparisons are as follows:



It is clear that Alpha=0.1,Beta=0.2,Gamma=0.1,TripleExponentialSmoothing has the lower RMSE of 9.22 and Naive Model has the highest RMSE value of 79.718

Based on above listed table, we have found that triple exponential Model doing the best prediction , based on Lowest RMSE value of 9.22 as compared to other models.

**Recommendations:**

From the forecast it is being predicted that the sales will increase in the next 12 months and efforts should be made to keep inventory as per the predicted forecast. Also as observed before sales will increase till December and then there is a likely chance of sharp drop in January followed by a very gradual increase till July 1996. This is similar observations during Monthly and yearly plot with seasonality .

Also, wine producing company should take efforts or run marketing campaign for promoting wine productions at the start of the year and especially in the period of Jan-July 1996 Months.

Company should also analyze further various business methods like marketing, optimization of raw materials and for better profitability.

# **Sparkling.csv**

Introduction: This report explains the business requirements and provide the detailed solution based on the data provided for each problem statement. given in the assignment. Also, the purpose of this exercise is to execute various Timeseries forecasting learning techniques and building various models over Timeseries data, combine all predictions and find out the model with best prediction or accuracy. Timeseries data is unlikely normal data used in supervised/unsupervised learning, where we have dependent /independent variables.

In TS data, we need to use predictions on Same field, using Old data set in ordered Time manner.

# **Read the data as an appropriate Time Series data and plot the data.**

Time Series is a sequence of observations recorded at regular time intervals.

| **YearMonth** | **Sparkling** |
| --- | --- |
| **0** | 1980-01-01 | 1686 |
| **1** | 1980-02-01 | 1591 |
| **2** | 1980-03-01 | 2304 |
| **3** | 1980-04-01 | 1712 |
| **4** | 1980-05-01 | 1471 |
| **...** | ... | ... |
| **182** | 1995-03-01 | 1897 |
| **183** | 1995-04-01 | 1862 |
| **184** | 1995-05-01 | 1670 |
| **185** | 1995-06-01 | 1688 |
| **186** | 1995-07-01 | 2031 |

187 rows × 2 columns

Given data is not time. So, we parse the date range and create a timestamp.

# **Plot for Rose wine Sales data**

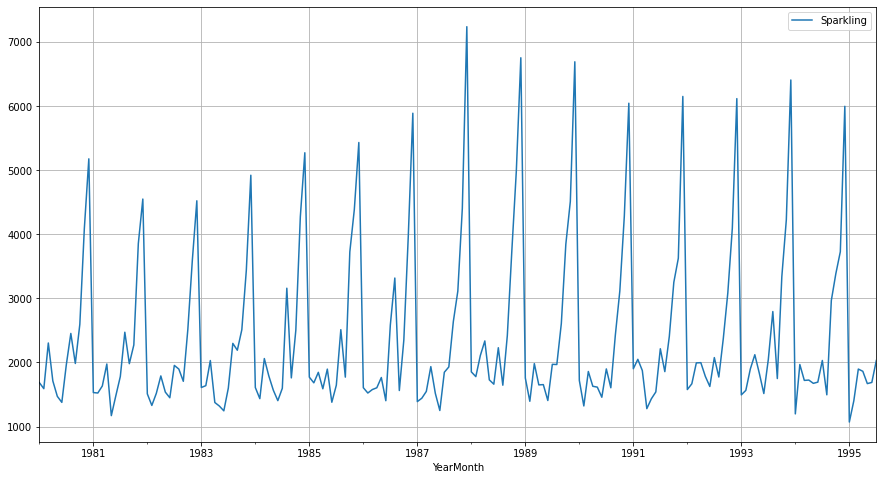


Figure Time series Plot

Describer for Sparkling Wine Sample data : Sparkling

count 187.000000

mean 2402.417112

std 1295.111540

min 1070.000000

25% 1605.000000

50% 1874.000000

75% 2549.000000

max 7242.000000

Insights:

1. Data consist of 187 data points
2. It seems to be contained seasonality
3. We also notice that there is small portion of the trend in data but altogether its not visible very clearly by plotting it in years.
4. Minimum sales for the data in Any month are 1070 and Max sales of Rose wines in any month is 7242
5. Year 1988 has highest sales
6. We also notice that 1995 has lowest sales , but this is not correct because we don’t have full data point for this year

# **Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.**

# **Box Plot for Year on year sales data**

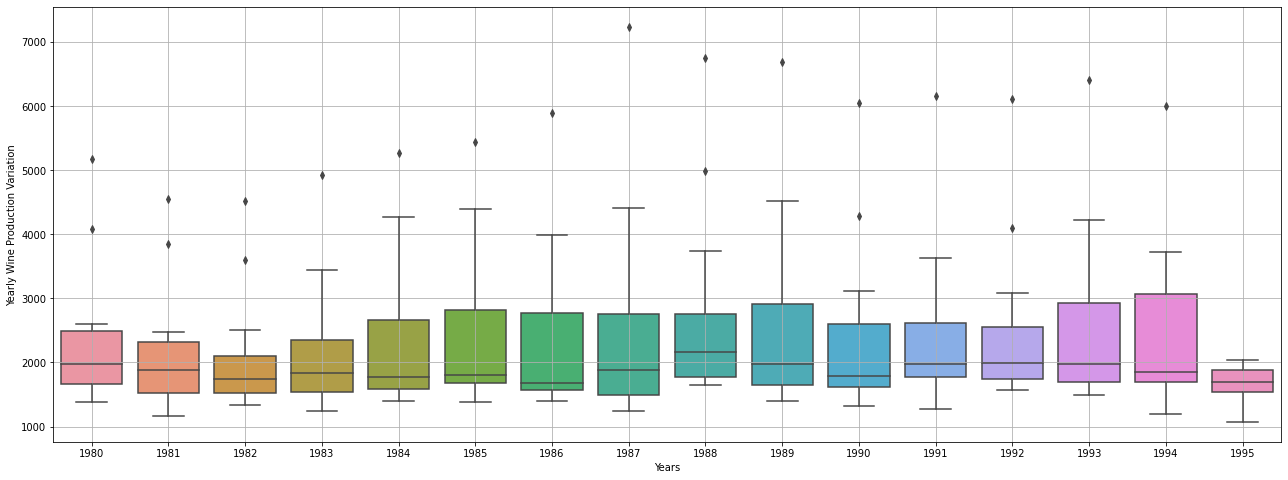


Figure Box plot for year to year sales

Insights:

1. We see the decrease in Sales for Sparkling wine in initial first 3-4 years , then again Sales increased , which remain almost constant and no big increase in sales .
2. Boxplot helps to check the outliers in each year and month and we see there are outliers in almost all the year as per the box plot.
3. Average sales are lowest in the year of 1995

# **Monthly plot**

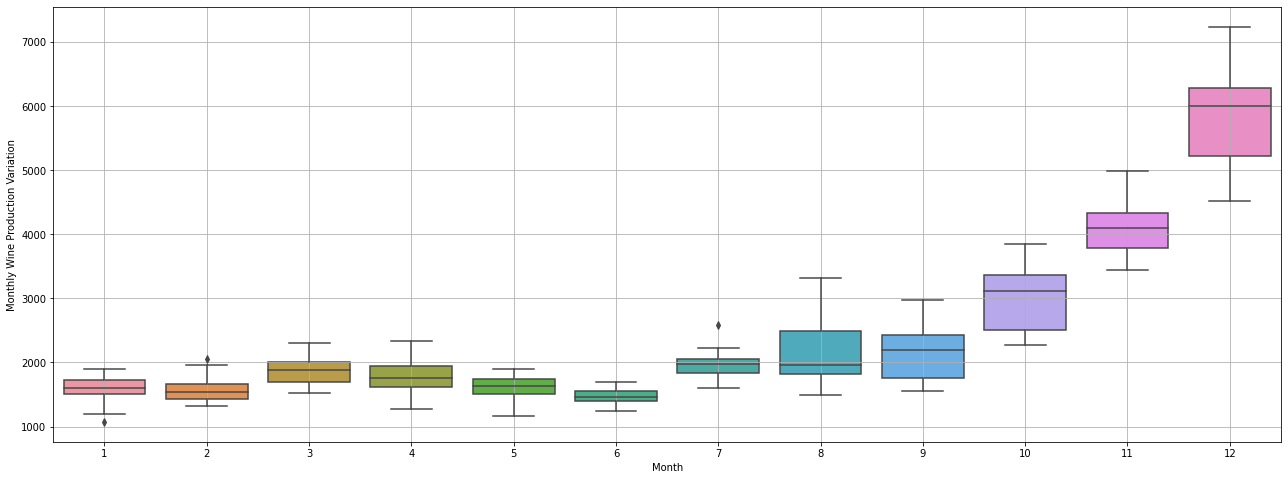


Figure Time series Plot

Insights:

1. The box plot for various months is plotted
2. Monthly plot contains outliers in the month of February and July month Sales
3. There are Highest sales in the Month of December followed by 2nd highest Sales in November Month, which indicates year end party and vacation celebration sales
4. As year end comes, sales start growing and it is about similar sales , which is very less for the first 6 months of any year

# **Plot a month plot of the give Time Series:**

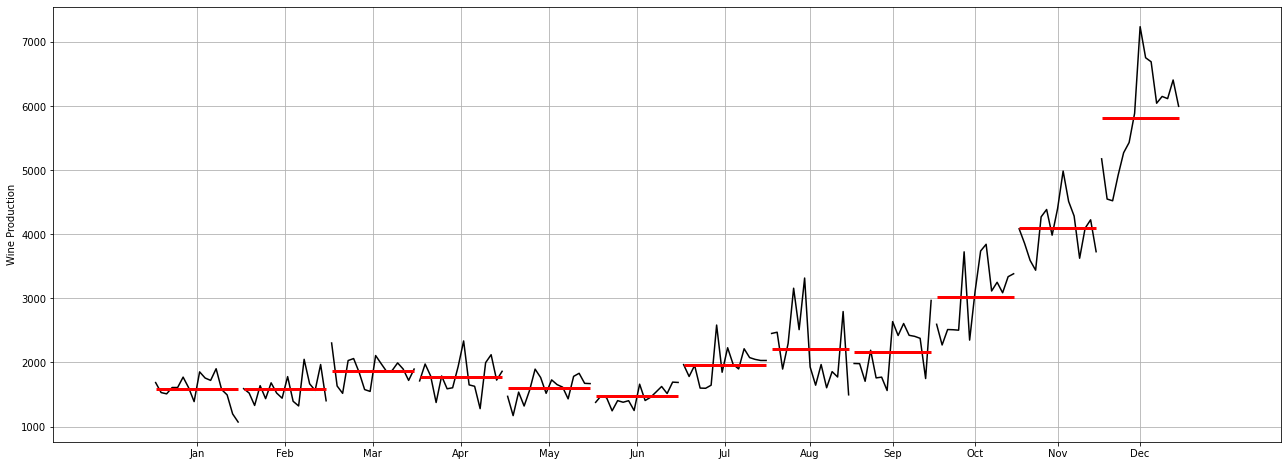


Figure Month Plot of the Given time series

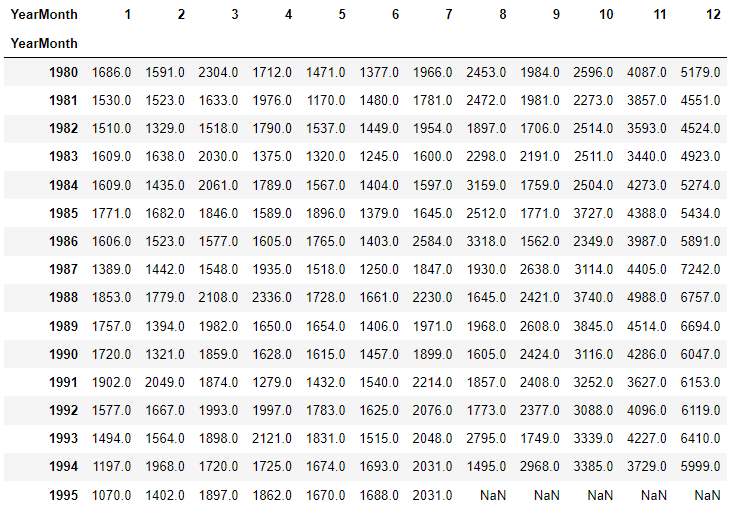
Insights:

1. We have accumulated all year’s data for each month and plotted it against each month.
2. We see that there are High Sales in each month at the Mid of the month and then Fall down till end of the month.
3. We also see decrease and lowest Sales in every Start of the month and then slightly cover up , goes highest in Mid of the month.-

# **Plot the Time Series according to different months for different years**



Figure yearly sales across Month



Insights:

1. December records have the high number of Sparkling wine sales in each year.
2. January have low number of wine sales.
3. There are 5 Months data not available for year 1995. So we have all data point available from Start of Year 1980 till July 1995.

# **Yearly Plot:**

aggregate the time series from an annual perspective and summing up the observations

|  |  |  |
| --- | --- | --- |
|  | **YearMonth** | **Sparkling** |
| **0** | 1980-12-31 | 28406 |
| **1** | 1981-12-31 | 26227 |
| **2** | 1982-12-31 | 25321 |
| **3** | 1983-12-31 | 26180 |
| **4** | 1984-12-31 | 28431 |

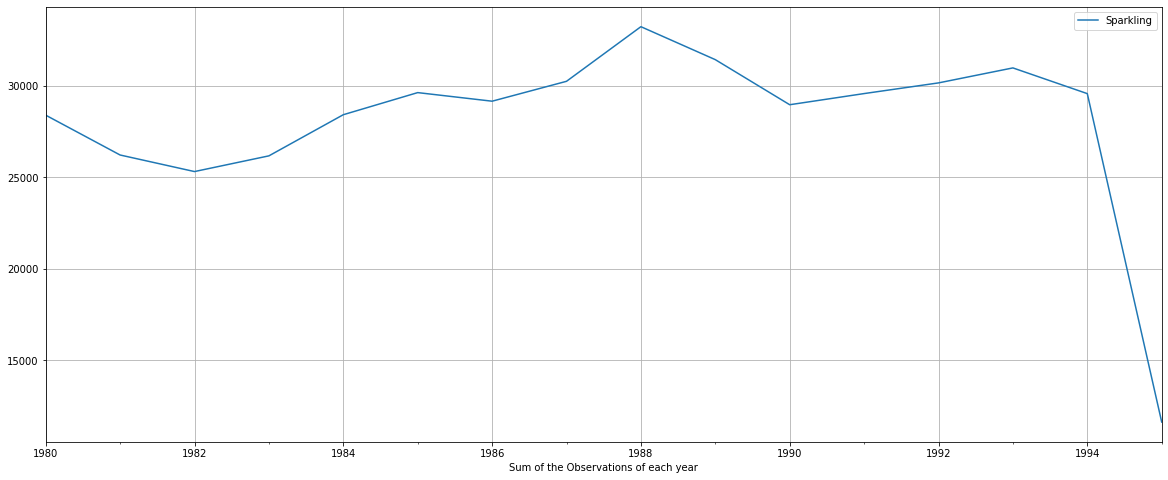


Figure Sum of Sales Each year Plot

Insights:

1. The plot shows that in 1982 there is a fall in the wine sales
2. Then sales goes steady will 1987
3. Highest sales observed in year 1988 , then goes down again for 89 and 90s.
4. After 1994, Sales goes down rapidly, but this is all because we don’t have data for full year 1995.
5. The resampled yearly or annual series have smoothened out the seasonality and have only been able to capture the year on year trend where there was.

# **Quarterly plot –**

aggregate the time series from a quarterly perspective and sum the observations of each quarter.

|  |  |
| --- | --- |
|  | **Sparkling** |
| **YearMonth** |  |
| **1980-03-31** | 5581 |
| **1980-06-30** | 4560 |
| **1980-09-30** | 6403 |
| **1980-12-31** | 11862 |
| **1981-03-31** | 4686 |

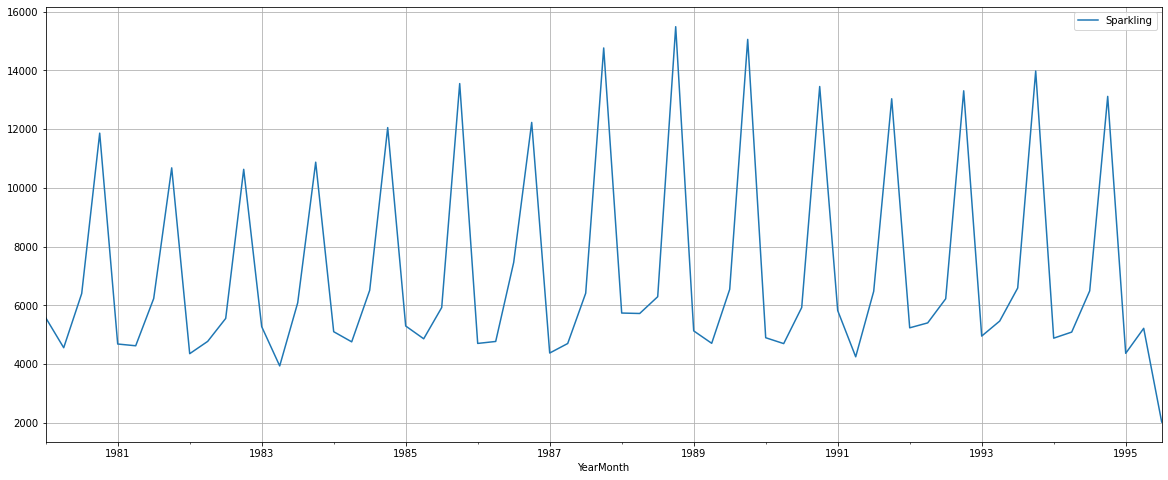


Figure Quarterly sales for each year

Insights:

1. We see that the year on year quarterly series represents the year on year monthly series. The quarterly series is able to catch the seasonality in the data.

# **Daily plot**

aggregate the data from a daily perspective

|  |
| --- |
|  |
| **YearMonth** | **Sparkling** |
| **1980-01-01** | 1686 |
| **1980-01-02** | 0 |
| **1980-01-03** | 0 |
| **1980-01-04** | 0 |
| **1980-01-05** | 0 |
| **...** | ... |
| **1995-06-27** | 0 |
| **1995-06-28** | 0 |
| **1995-06-29** | 0 |
| **1995-06-30** | 0 |
| **1995-07-01** | 2031 |

5661 rows × 1 columns

The values which the original series cannot provide is taken as 0 by python if we try to resample the data on a daily basis.

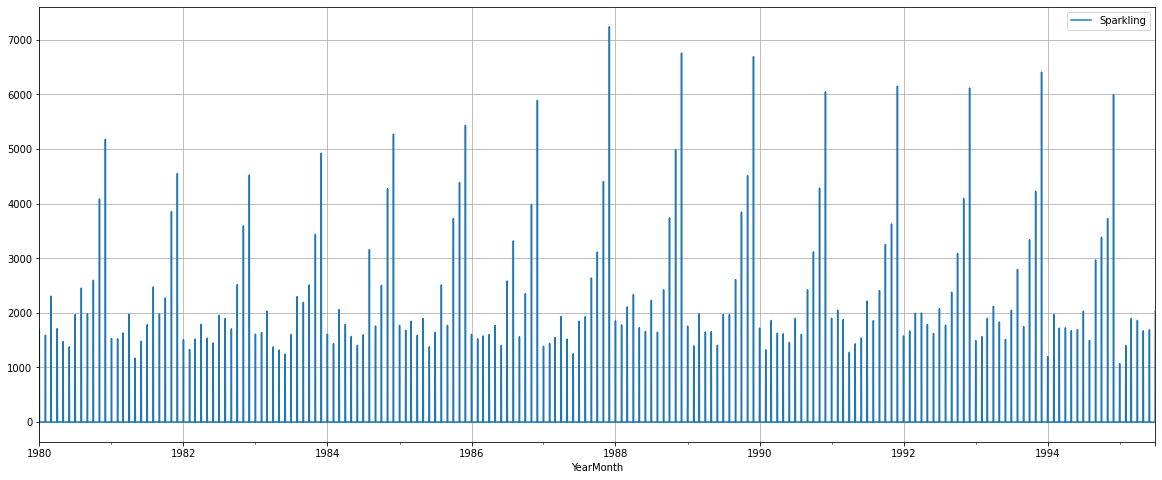


Figure Daily Sales for all years

The above graph fails to give us a proper understanding of our data. Thus, resampling the data to intervals where a number of observations are 0 is not a good idea as that does not give us an understanding of the performance of the time series.

To get a very high-level overview of the trend of the Time Series Data (if Trend is present) can be understood by resampling the data keeping the intervals very large.

# **Decade Plot**

|  | **Sparkling** |
| --- | --- |
| **YearMonth** |  |
| **1980-12-31** | 28406 |
| **1990-12-31** | 288893 |
| **2000-12-31** | 131953 |

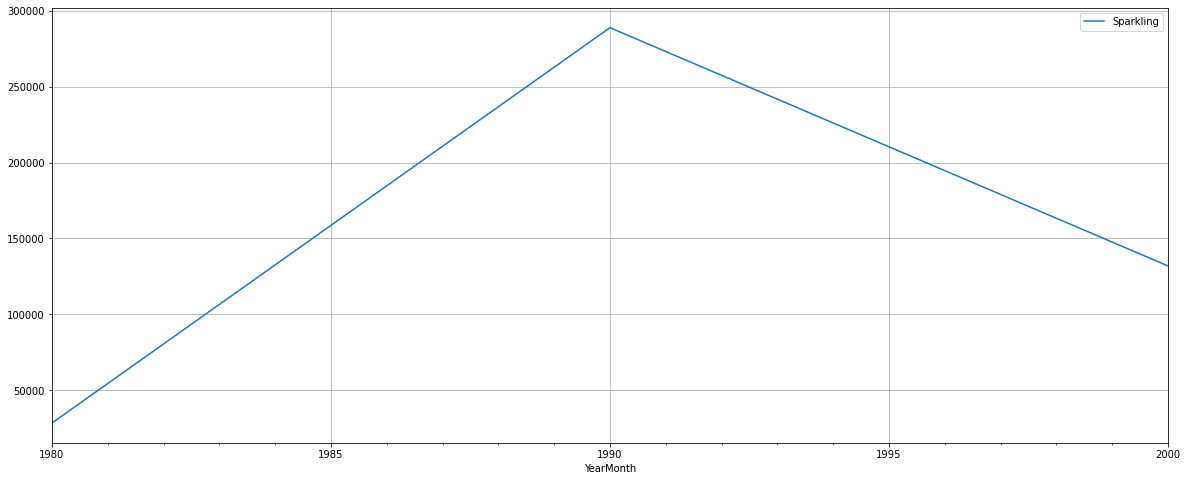


Figure Decade Sales for all data

If we take the resampling period to be 10 years or a decade, we see that the seasonality present has been smoothed over and it is only giving an estimate of the trend.

There are always Increase in decade data, which gets flattened out in future decades

**Decompose the Time Series : Additive Model**

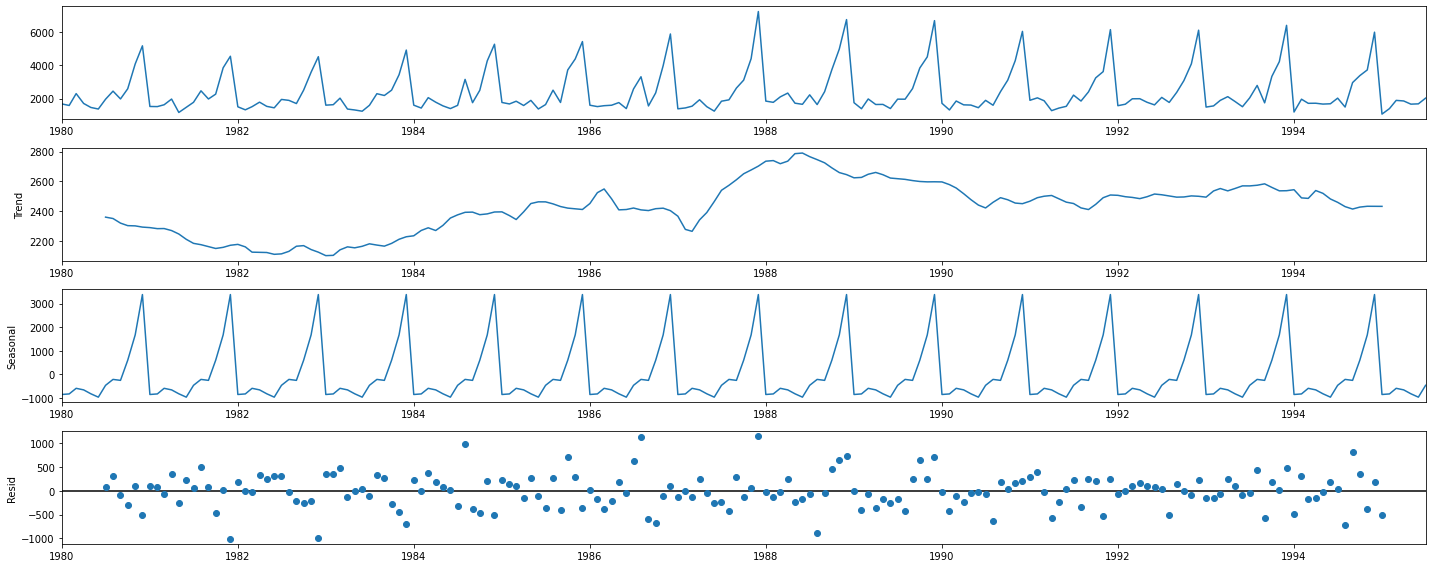


Figure Additive Decomposition

Insights:

1. We have built 2 Models of the data Additive trend as well as Multiplicative Trend .
2. From the ‘additive’ decomposition, there is seasonality in the data. Which is at the End of the year when Sales goes High, and at the Start of the year, sales goes down for following years.
3. Sales trend is almost constant for 2nd half of the data and don’t see , a big difference in total Sales

Trend

YearMonth

1980-01-01 NaN

1980-02-01 NaN

1980-03-01 NaN

1980-04-01 NaN

1980-05-01 NaN

1980-06-01 NaN

1980-07-01 2360.666667

1980-08-01 2351.333333

1980-09-01 2320.541667

1980-10-01 2303.583333

1980-11-01 2302.041667

1980-12-01 2293.791667

Name: trend, dtype: float64

Seasonality

YearMonth

1980-01-01 -854.260599

1980-02-01 -830.350678

1980-03-01 -592.356630

1980-04-01 -658.490559

1980-05-01 -824.416154

1980-06-01 -967.434011

1980-07-01 -465.502265

1980-08-01 -214.332821

1980-09-01 -254.677265

1980-10-01 599.769957

1980-11-01 1675.067179

1980-12-01 3386.983846

Name: seasonal, dtype: float64

Residual

YearMonth

1980-01-01 NaN

1980-02-01 NaN

1980-03-01 NaN

1980-04-01 NaN

1980-05-01 NaN

1980-06-01 NaN

1980-07-01 70.835599

1980-08-01 315.999487

1980-09-01 -81.864401

1980-10-01 -307.353290

1980-11-01 109.891154

1980-12-01 -501.775513

Name: resid, dtype: float64

# **Sales data without Seasonality component:**

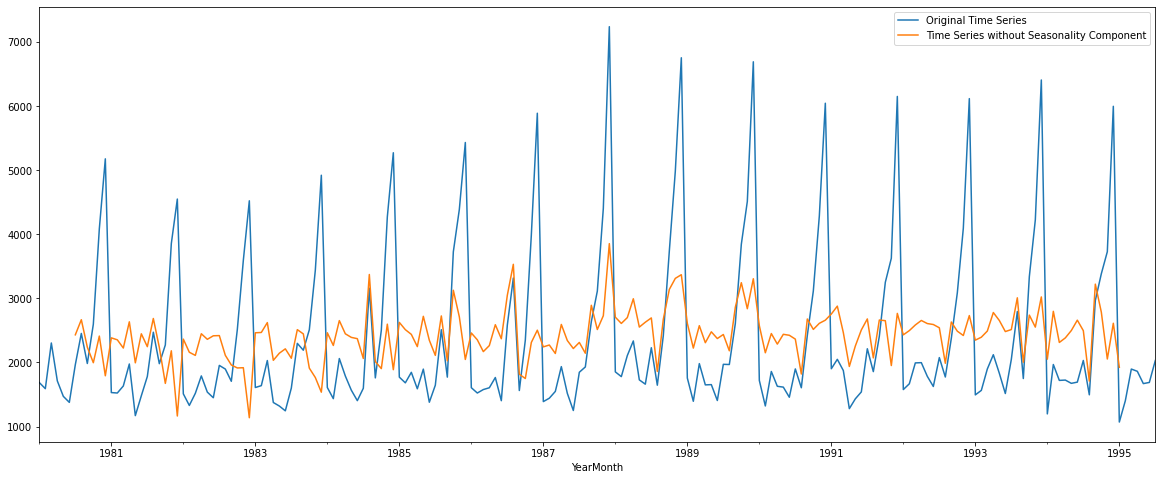


Figure Sales data without Seasonality

# **Multiplicative Model for Rose data problem**

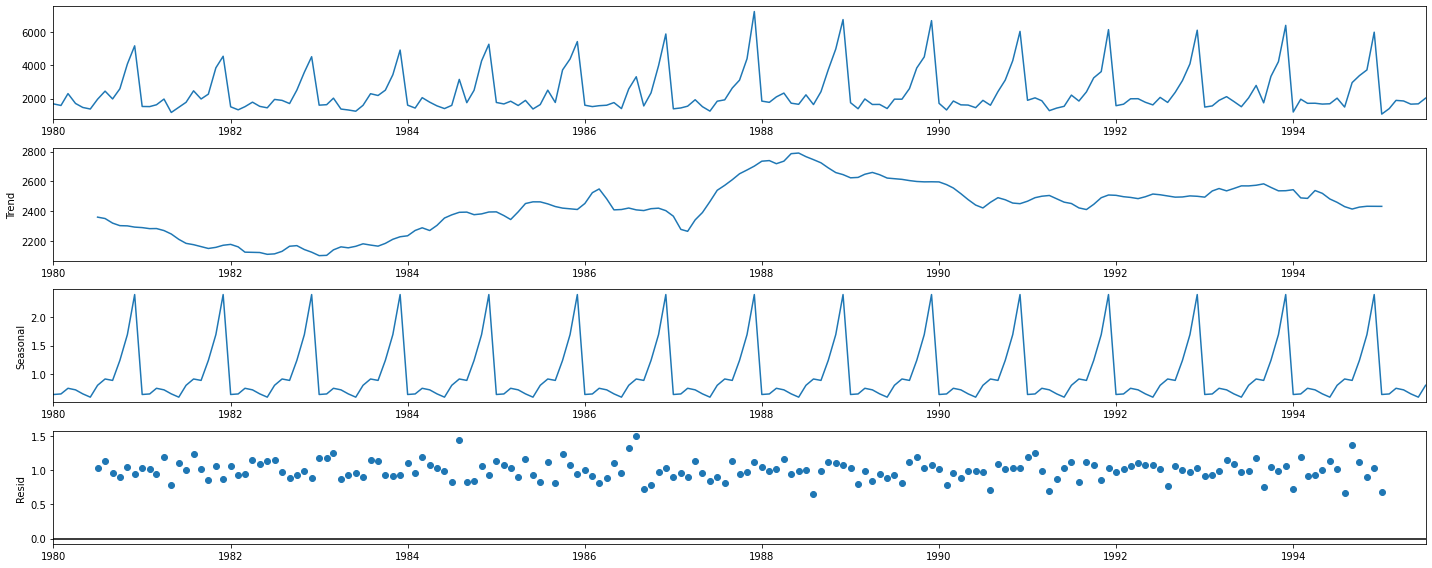


Figure Multiplicative Sales Decomposition

Since Residual is more close to a single line, we would choose Multiplicative Decomposition over Additive decomposing.

# **Split the data into training and test. The test data should start in 1991.**

We have split the data in train and test data set,

Data before 1991 is considered as train set and data after 1991 is considered as test data set:

After splitting this is the final shape of the Train and Test data

(132, 1)

(55, 1)

First few rows of Training Data

Sparkling

YearMonth

1980-01-01 1686

1980-02-01 1591

1980-03-01 2304

1980-04-01 1712

1980-05-01 1471

Last few rows of Training Data

Sparkling

YearMonth

1990-08-01 1605

1990-09-01 2424

1990-10-01 3116

1990-11-01 4286

1990-12-01 6047

First few rows of Test Data

Sparkling

YearMonth

1991-01-01 1902

1991-02-01 2049

1991-03-01 1874

1991-04-01 1279

1991-05-01 1432

Last few rows of Test Data

Sparkling

YearMonth

1995-03-01 1897

1995-04-01 1862

1995-05-01 1670

1995-06-01 1688

1995-07-01 2031

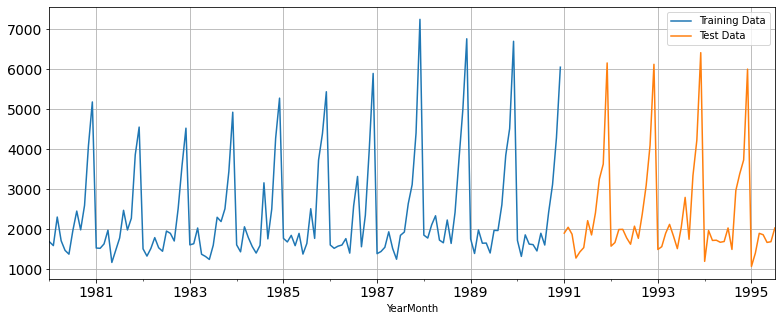


Figure Plot Train and test data

It is difficult to predict the future if the past is not happened. From the above split, we are predicting similar to the past data.

# **Build all the exponential smoothing models on the training data and evaluate the model using RMSE on the test data. Other additional models such as regression, naïve forecast models, simple average models, moving average models should also be built on the training data and check the performance on the test data using RMSE**

# **Model1: Linear Regression**

For this particular linear regression, we are going to regress the ‘Sparkling’ variable against the order of the occurrence. For this we need to modify our training data before fitting it into a linear regression.

Regress the “Sparkling’” variable against the order of occurrence. We have also generated the numerical instance order for both training and test set . Linear Regression is built on the training and test dataset

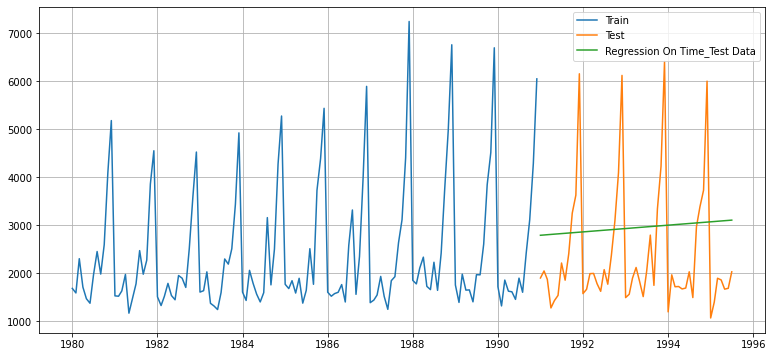


Figure Linear regression Prediction Plot

We have evaluated the Model based on RMSE parameter and put this in a Data frame, which we would use later for Comparing multiple models

Model evaluation :

|  | **Test RMSE** |
| --- | --- |
| **RegressionOnTime** | 1389.135175 |

# **Model2 – Naïve model**

For this particular naive model, we say that the prediction for tomorrow is the same as today and the prediction for day after tomorrow is tomorrow and since the prediction of tomorrow is same as today, therefore the prediction for day after tomorrow is also today.

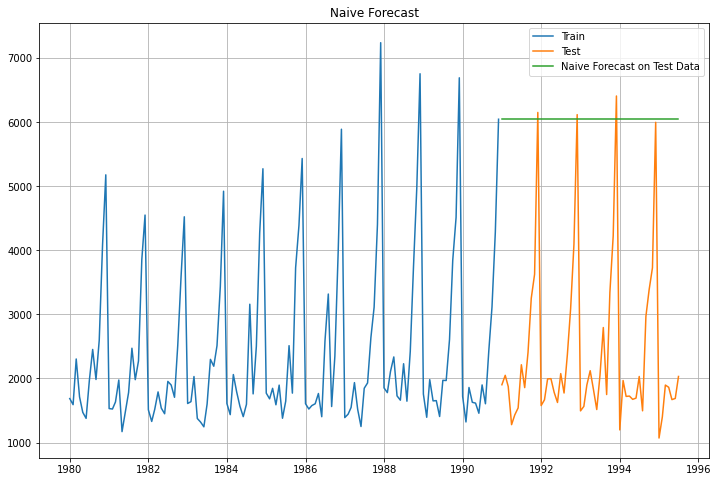


Figure Prediction Plot for Naïve based Model

Model Evaluation

|  | **Test RMSE** |
| --- | --- |
| **RegressionOnTime** | 1389.135175 |
| **NaiveModel** | 3864.279352 |

# **Model 3 – Simple Average**

For this particular simple average method, we will forecast by using the average of the training values.

|  |  |  |
| --- | --- | --- |
|  | **Sparkling** | **mean\_forecast** |
| **YearMonth** |  |  |
| **1991-01-01** | 1902 | 2403.780303 |
| **1991-02-01** | 2049 | 2403.780303 |
| **1991-03-01** | 1874 | 2403.780303 |
| **1991-04-01** | 1279 | 2403.780303 |
| **1991-05-01** | 1432 | 2403.780303 |

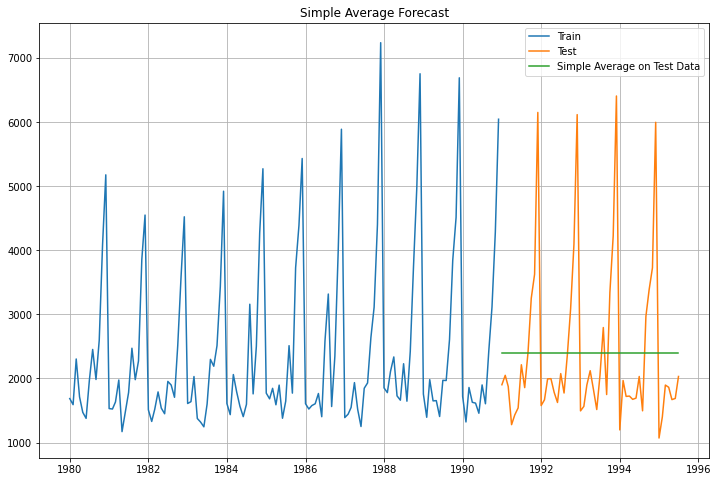


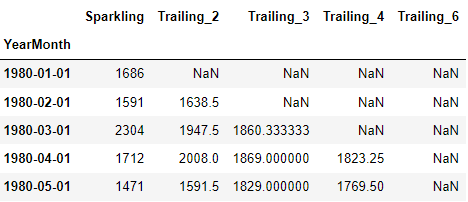
Figure Prediction Plot for Simple Average

| Model Evaluation: | **Test RMSE** |
| --- | --- |
| **RegressionOnTime** | 1389.135175 |
| **NaiveModel** | 3864.279352 |
| **SimpleAverageModel** | 1275.081804 |

# **Model4- Moving Average –**

For the moving average model, we are going to calculate rolling means (or moving averages) for different intervals. The best interval can be determined by the maximum accuracy (or the minimum error) over here.

Let’s take moving average of previous 2, 3, 4 and 6 data elements:



Plot the predictions based on Moving averages:

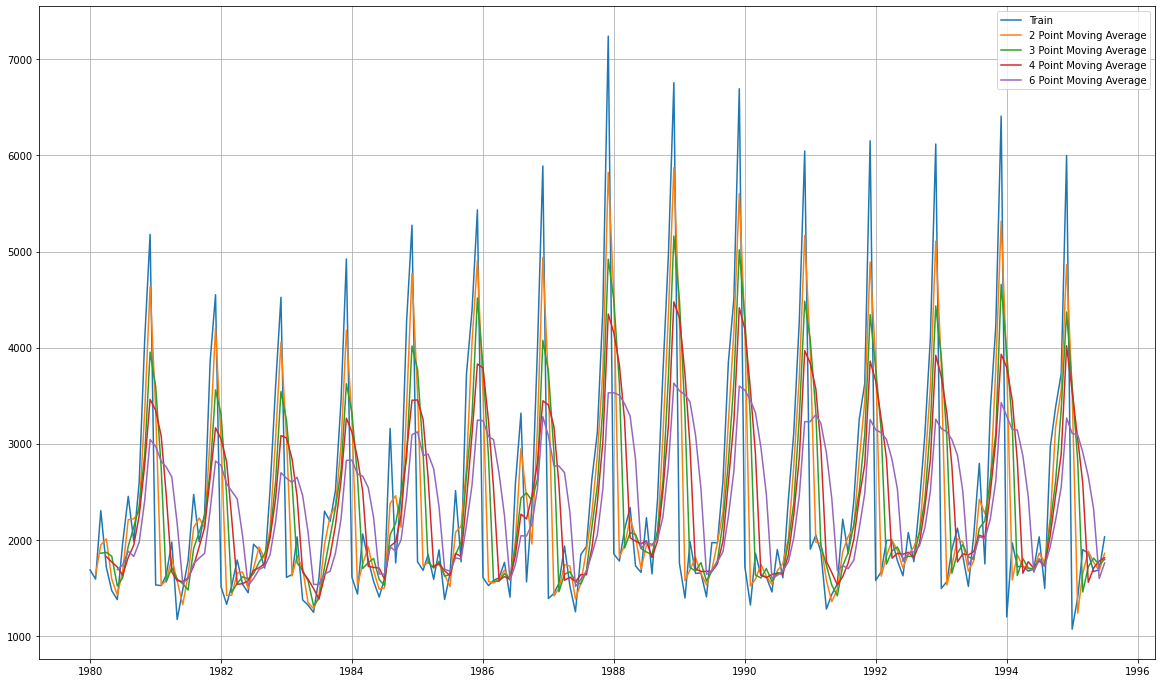


Figure Moving Average for 2,3,4 and 6 MA

Let us split the data into train and test and plot this Time Series. The window of the moving average is need to be carefully selected as too big a window will result in not having any test set as the whole series might get averaged over.

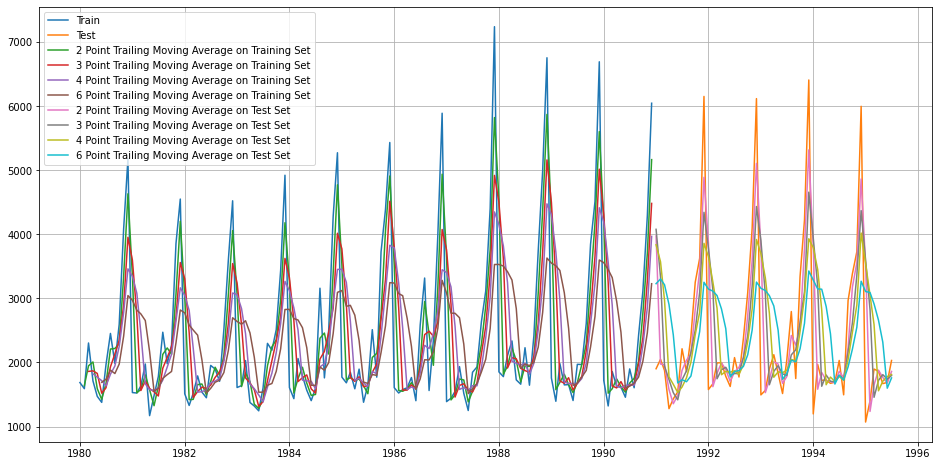


Figure Plot MA on Full data

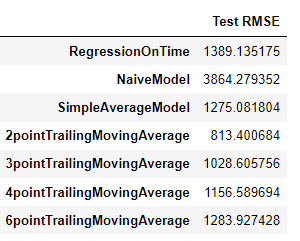
**Model Evaluation:**

For 2 point Moving Average Model forecast on the Training Data, RMSE is 813.401

For 3 point Moving Average Model forecast on the Training Data, RMSE is 1028.606

For 4 point Moving Average Model forecast on the Training Data, RMSE is 1156.590

For 6 point Moving Average Model forecast on the Training Data, RMSE is 1283.927



The Best Model for Moving average is 2 points trailing moving average with lowest RMSE of 813.401, We would choose this model for predicting the Test model, if we are to choose the best one from above all models:

Before we go on to build the various Exponential Smoothing models, let us plot all the models and compare the Time Series plots

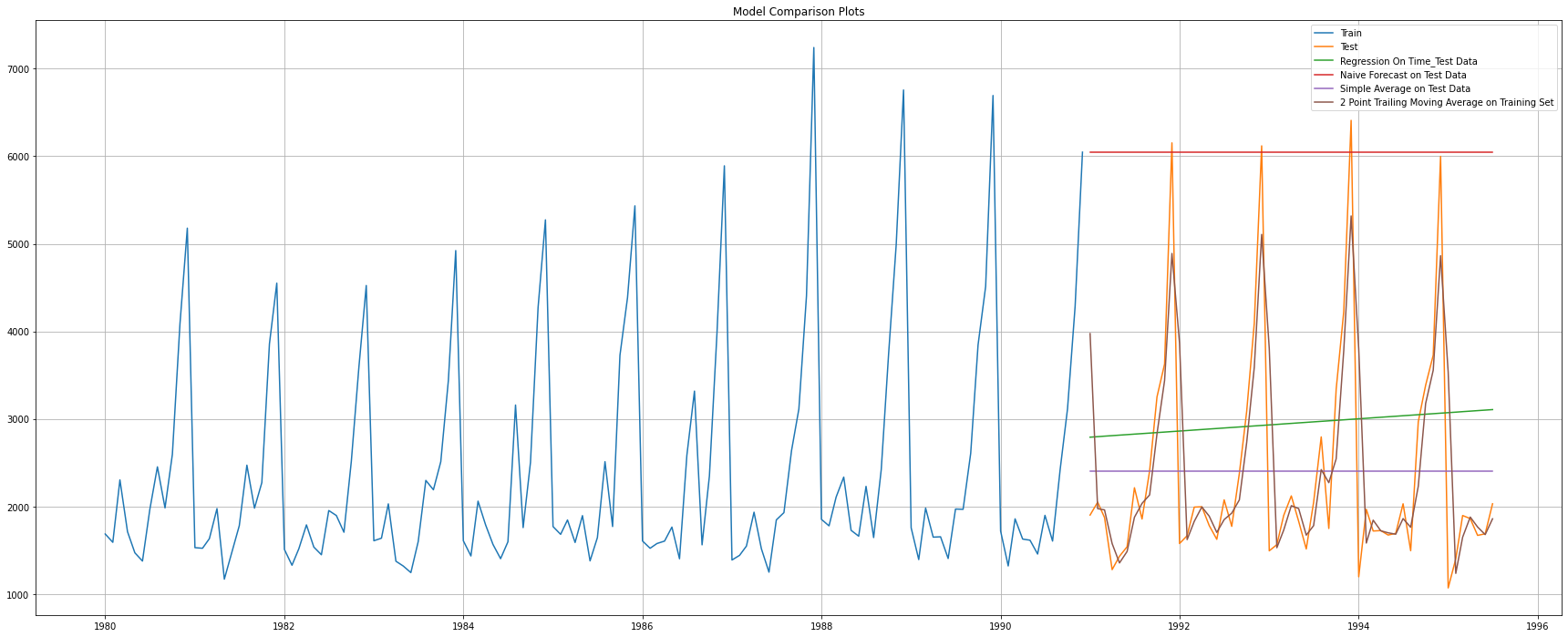


Figure Building all Models on Test data

# **Model -5- Exponential Smoothing**

Exponential smoothing methods consist of flattening time series data. Exponential smoothing averages or exponentially weighted moving averages consist of forecast based on previous periods data with exponentially declining influence on the older observations.

Exponential smoothing methods consist of special case exponential moving with notation ETS (Error, Trend, Seasonality) where each can be none(N), additive (N), additive damped (Ad), Multiplicative (M) or multiplicative damped (Md). One or more parameters control how fast the weights decay. These parameters have values between 0 and 1

First Model in Exponential smoothing is SES (Simple Exponential Smoothing) .

SES - ETS(A, N, N) - Simple Exponential Smoothing with additive errors

The simplest of the exponentially smoothing methods is naturally called simple exponential smoothing (SES). This method is suitable for forecasting data with no clear trend or seasonal pattern.

In Single ES, the forecast at time (t + 1) is given by Winters,1960

𝐹𝑡+1=𝛼𝑌𝑡+(1−𝛼)𝐹𝑡

Parameter 𝛼 is called the smoothing constant and its value lies between 0 and 1. Since the model uses only one smoothing constant, it is called Single Exponential Smoothing.

Note: Here, there is both trend and seasonality in the data. So, we should have directly gone for the Triple Exponential Smoothing but Simple Exponential Smoothing and the Double Exponential Smoothing models are built over here to get an idea of how the three types of models compare in this case.

SimpleExpSmoothing class must be instantiated and passed the training data.

The fit() function is then called providing the fit configuration, the alpha value, smoothing\_level. If this is omitted or set to None, the model will automatically optimize the value

Auto parameters:

{'smoothing\_level': 0.04960659880745982,

'smoothing\_trend': nan,

'smoothing\_seasonal': nan,

'damping\_trend': nan,

'initial\_level': 1818.5047538435374,

'initial\_trend': nan,

'initial\_seasons': array([], dtype=float64),

'use\_boxcox': False,

'lamda': None,

'remove\_bias': False}

Perform Prediction on test data set and these are the result look like:

|  |  |  |
| --- | --- | --- |
|  | **Sparkling** | **predict** |
| **YearMonth** |  |  |
| **1991-01-01** | 1902 | 2724.929339 |
| **1991-02-01** | 2049 | 2724.929339 |
| **1991-03-01** | 1874 | 2724.929339 |
| **1991-04-01** | 1279 | 2724.929339 |
| **1991-05-01** | 1432 | 2724.929339 |

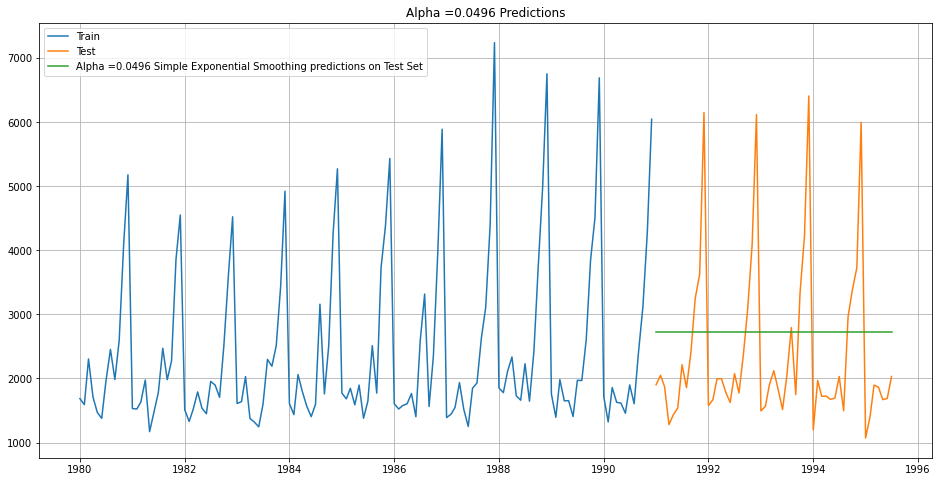
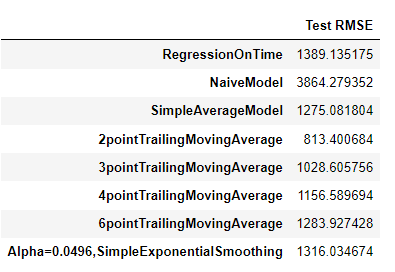
****

Figure SES Model on Auto parameters

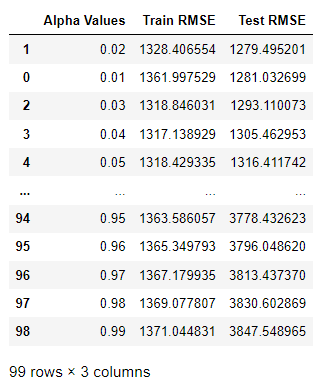
Model Evaluation for 𝛼 = 0.09874989743650385 : Simple Exponential Smoothing For Alpha =0.09874989743650385 Simple Exponential Smoothing Model forecast on the Test Data, RMSE is 36.796



**Setting different alpha values**. The higher the alpha value more weightage is given to the more recent observation. That means, what happened recently will happen again.

We will run a loop with different alpha values to understand which particular value works best for alpha on the test set. We have given a range of values between 0.01 to 1 with the interval of 0.01 so , Alpha values like 0.01, 0.02, 0.03 … up to 1.

We tried all 100 possible values and calculated the RMSE value for that, and chosen best Alpha value for Lowest RMSE value



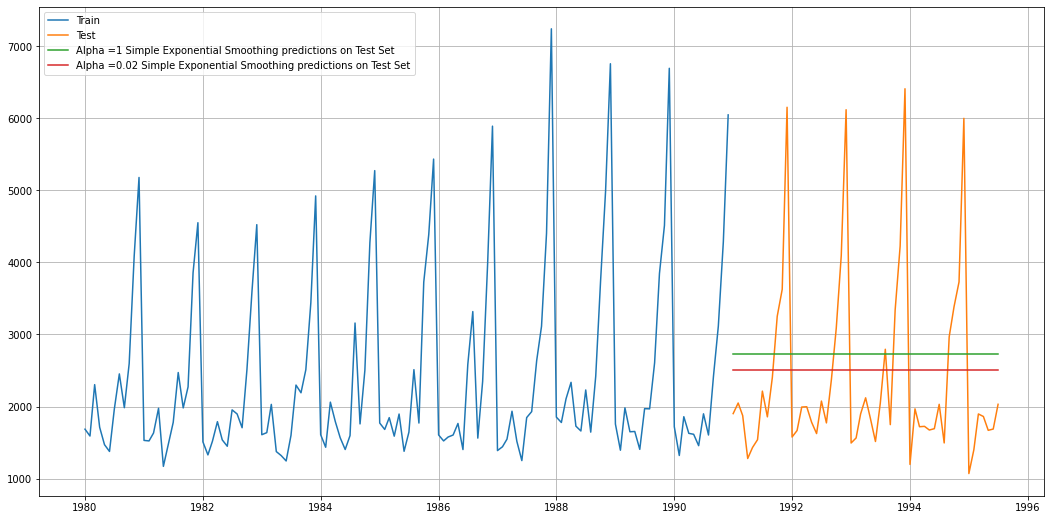
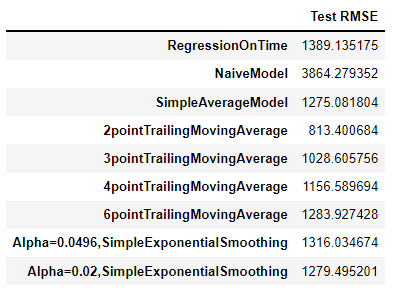


Figure SES Model on Corrected parameters

Model evaluation:



# **Method 6: Double Exponential Smoothing (Holt's Model)**

Holt - ETS(A, A, N) - Holt's linear method with additive errors , Double Exponential Smoothing.

One of the drawbacks of the simple exponential smoothing is that the model does not do well in the presence of the trend. This model is an extension of SES known as Double Exponential model which estimates two smoothing parameters. This is applicable when data has Trend but no seasonality. In this model two separate components are considered: **Level** and **Trend**. **Level** is the local mean.

One smoothing parameter α corresponds to the level series.

A second smoothing parameter β corresponds to the trend series.

Double Exponential Smoothing uses two equations to forecast future values of the time series, one for forecasting the short-term average value or level and the other for capturing the trend.

Intercept or Level equation, 𝐿𝑡 is given by: 𝐿𝑡=𝛼𝑌𝑡+(1−𝛼)𝐹𝑡

Trend equation is given by 𝑇𝑡=𝛽(𝐿𝑡−𝐿𝑡−1)+(1−𝛽)𝑇𝑡−1

Here, 𝛼 and 𝛽 are the smoothing constants for level and trend, respectively,

0 < 𝛼 < 1 and 0 < 𝛽 < 1.

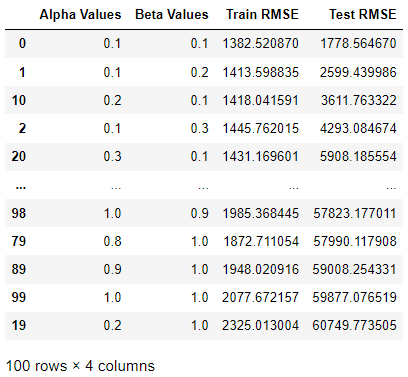
The forecast at time t + 1 is given by

𝐹𝑡+1=𝐿𝑡+𝑇𝑡

𝐹𝑡+𝑛=𝐿𝑡+𝑛𝑇𝑡

Two parameters 𝛼 and 𝛽 are estimated in this model. Level and Trend are accounted for in this model

Similar to Above SES model, we will calculate Alpha and beta values by using the for loops and calculate the RMSE value. Then we would choose Alpha and beta value for the Lowest RMSE.



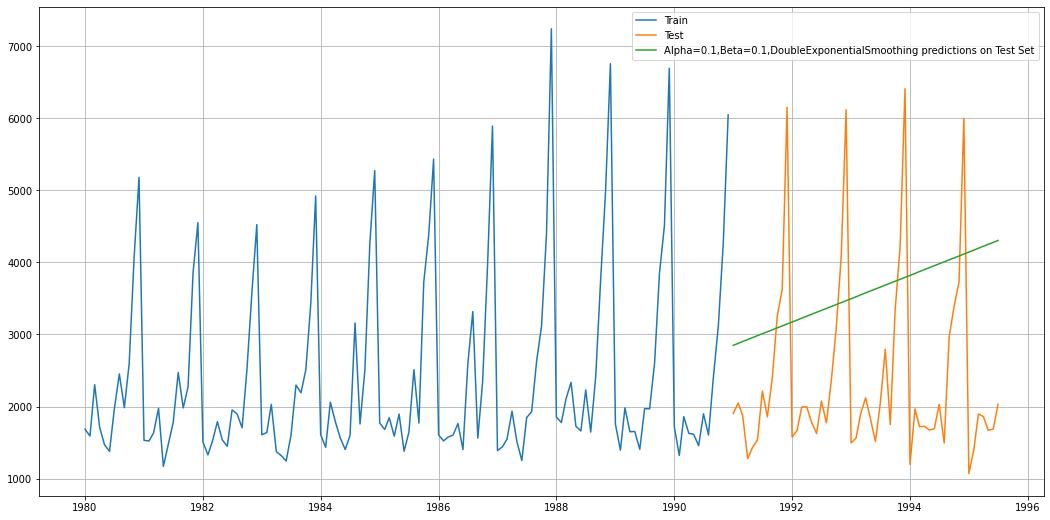


Figure DES model

We see that the double exponential smoothing is picking up the trend component along with the level component as well.



Inference

Here, we see that the Double Exponential Smoothing has actually done well when compared to the Simple Exponential Smoothing. This is because of the fact that the Double Exponential Smoothing model has picked up the trend component as well.

The Holt's model in Python has certain other options of exponential trends or whether the smoothing parameters should be damped. You can try these out later to check whether you get a better forecast.

# **Method 7: Triple Exponential Smoothing (Holt - Winter's Model)**

Three parameters 𝛼 , 𝛽 and 𝛾 are estimated in this model. Level, Trend and Seasonality are accounted for in this model. First of all build the model with default parameters:

{'smoothing\_level': 0.11119949831569428,

'smoothing\_trend': 0.049430920023313805,

'smoothing\_seasonal': 0.3620525701498937,

'damping\_trend': nan,

'initial\_level': 2356.5264391986907,

'initial\_trend': -9.443690175376352,

'initial\_seasons': array([0.71325627, 0.68332509, 0.90537798, 0.80561841, 0.65639659,

0.65451508, 0.88690241, 1.13423953, 0.91927727, 1.21396745,

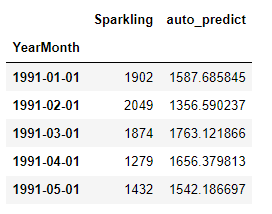
1.86941738, 2.3734461 ]),

'use\_boxcox': False,

'lamda': None,

'remove\_bias': False}

And predicted values look like this, as compare to Actual values



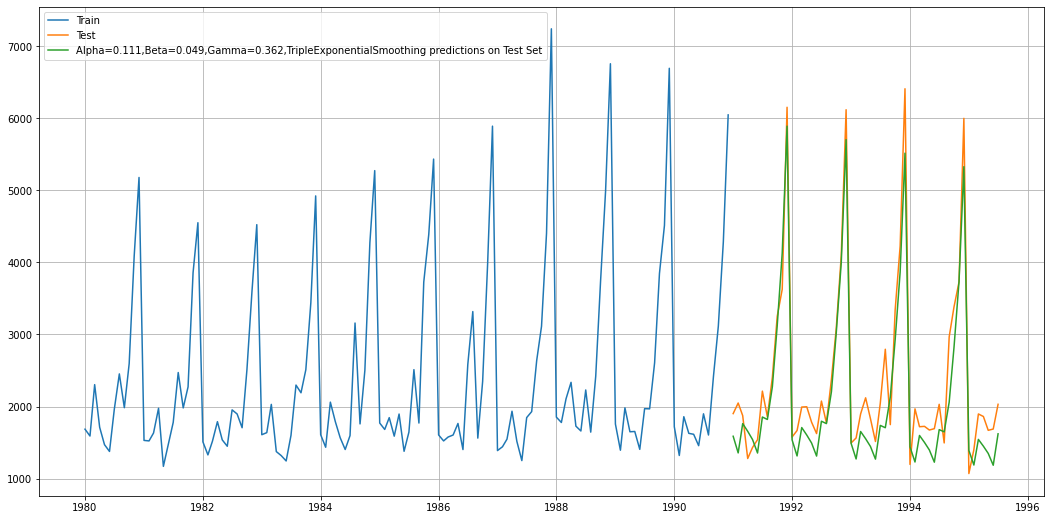
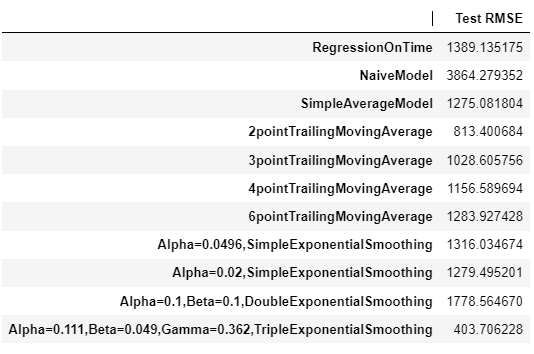


Figure TES Model on Auto Parameters

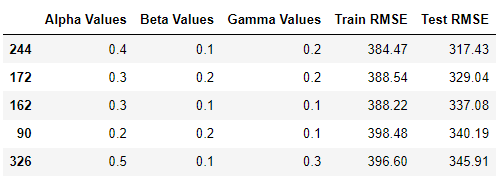
We see that the Triple Exponential Smoothing is picking up the seasonal component as well.

Inference

Triple Exponential Smoothing has performed the best on the test as expected since the data had both trend and seasonality. But we see that our triple exponential smoothing is under forecasting. Let us try to tweak some of the parameters in order to get a better forecast on the test set.



Auto generated Alpha , beta and game values and decide best based on lowest RMSE values



Final values are chosen based as 0.4, 0.1 and 0.2 on lowest RMSE value of 317.43.

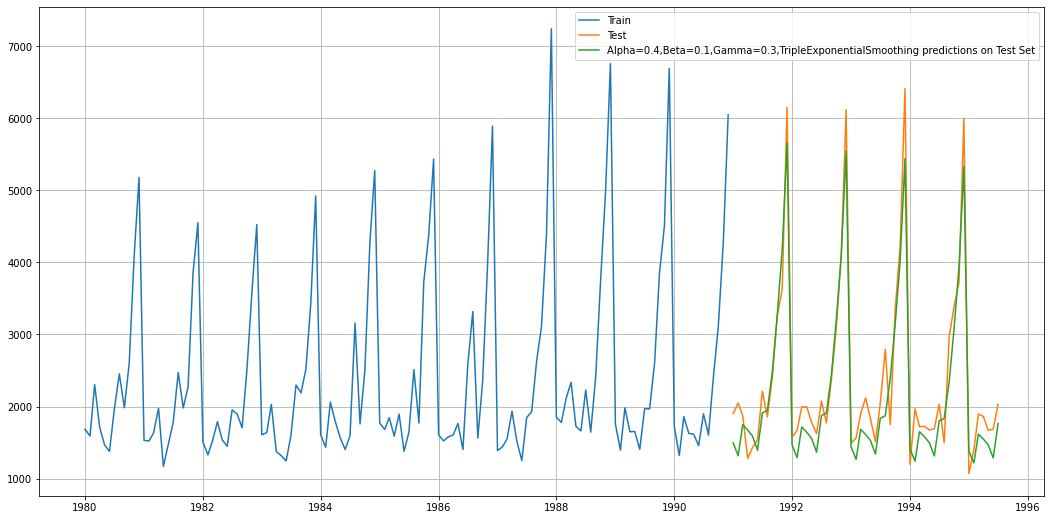
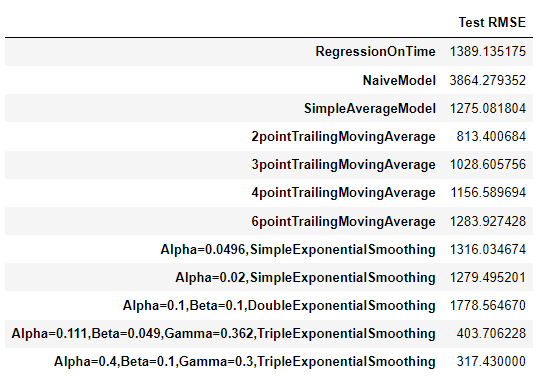


Figure TES Model on Corrected parameters

Compare the Model generated so far:



We see that the best model is the Triple Exponential Smoothing with multiplicative seasonality with the parameters 𝛼 = 0.4 , 𝛽 = 0.1 and 𝛾 = 0.3.

For this data, we had both trend and seasonality so by definition Triple Exponential Smoothing is supposed to work better than the Simple Exponential Smoothing as well as the Double Exponential Smoothing.

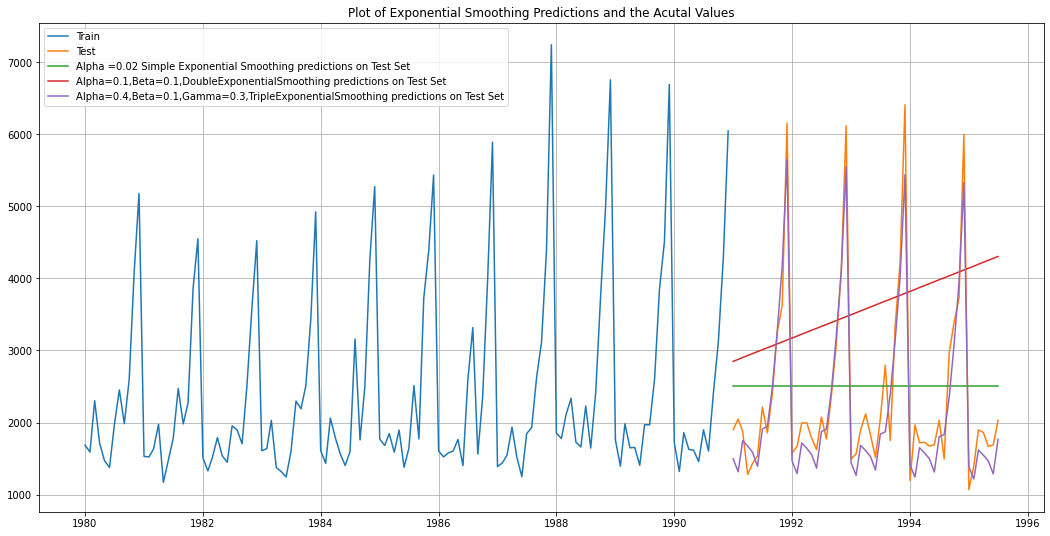


Figure SES, DES and TES Model together comparison on Test data

In this particular we have built several models and went through a model building exercise. This particular exercise has given us an idea as to which particular model gives us the least error on our test set for this data. But in Time Series Forecasting, we need to be very vigil about the fact that after we have done this exercise we need to build the model on the whole data. Remember, the training data that we have used to build the model stops much before the data ends. In order to forecast using any of the models built, we need to build the models again (this time on the complete data) with the same parameters. For this particular mentored learning session, we will go ahead and build only the top 1 model which gave us the best accuracy (least RMSE).

The two models to be built on the whole data are the following:

Alpha=0.4,Beta=0.1,Gamma=0.3,TripleExponentialSmoothing

Alpha=0.111,Beta=0.049,Gamma=0.362,TripleExponentialSmoothing

# **Check for the stationarity of the data on which the model is being built on using appropriate statistical tests and also mention the hypothesis for the statistical test. If the data is found to be non-stationary, take appropriate steps to make it stationary. Check the new data for stationarity and comment.**

Note: Stationarity should be checked at alpha = 0.05

A time series has stationarity when the observations are not dependent on the time. Statistical properties of these time series will not change with time thus they will have constant mean, variance, and covariance.

The time series which have trends or with seasonality, are not stationary. Because trends will have a change in the movement of data concerning time which will cause the change in mean over time. Whereas seasonality occurs when the pattern in time series shows a variation for a regular time interval which will cause the variance to change over time.

Stationarity of time series can be detected by: Visually Plotting the time series and check for trend or seasonality. By Splitting time series into the different partitions and compare the statistical inference.

We can also perform Augmented Dickey-Fuller test to check the stationarity.

The Augmented Dickey-Fuller test is a unit root test which determines whether there is a unit root and subsequently whether the series is non-stationary.

**The hypothesis in a simple form for the ADF test is:**

**𝐻0H0 : The Time Series has a unit root and is thus non-stationary.**

**𝐻1H1 : The Time Series does not have a unit root and is thus stationary.**

We would want the series to be stationary for building ARIMA models and thus we would want the p-value of this test to be less than the 𝛼α value. We have performed ADF test and following are the results for the same:

DF test statistic is -1.798

DF test p-value is 0.7055958459932058

Number of lags used 12

Since P value for above test is greater than 0.05 , which is 0.4671371627793189, so we are fail to reject the Null hypothesis and we accept that it is a Non stationary Time series. We see that at 5% significant level the Time Series is non-stationary.

There are various ways that Python allows us to select the appropriate number of lags at which we check whether the Time Series is stationary. To know more about the how to select the various ways, please refer to the link over ℎ𝑒𝑟𝑒.

Let us take one level of differencing to see whether the series becomes stationary.

DF test statistic is -44.912

DF test p-value is 0.0

Number of lags used 10

Since P value is way less than 0.05 so going back by one level of differencing will make time series as Stationary. Now, let us go ahead and plot the stationary series.

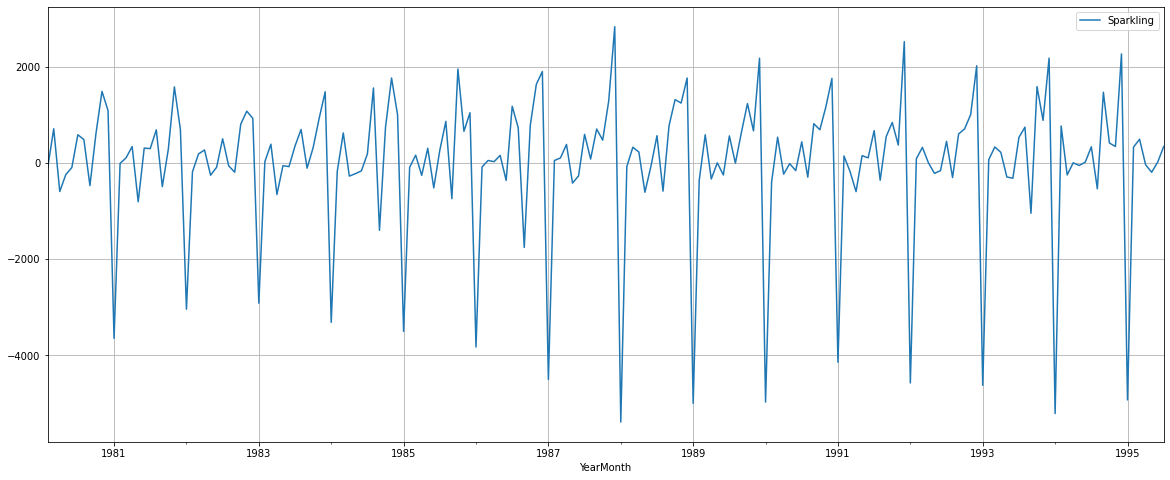


Figure Stationary Time series

Also, if the series is non-stationary, stationaries the Time Series by taking a difference of the Time Series. Then we can use this particular differenced series to train the ARIMA models. We do not need to worry about stationarity for the Test Data because we are not building any models on the Test Data, we are evaluating our models over there. You can look at other kinds of transformations as part of making the time series stationary like taking logarithms

# **Build an automated version of the ARIMA/SARIMA model in which the parameters are selected using the lowest Akaike Information Criteria (AIC) on the training data and evaluate this model on the test data using RMSE**

# **ARIMA Model:**

An ARIMA and SARIMA models are class of statistical models for analyzing and forecasting time series data. Lets break it down :

* AR: Autoregression. A model that uses the dependent relationship between an observation and some number of lagged observations.
* I: Integrated. The use of differencing of raw observations in order to make the time series stationary.
* MA: Moving Average. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

The parameters of the ARIMA model are defined as follows:

* p: The number of lag observations included in the model, also called the lag order.
* d: The number of times that the raw observations are differenced, also called the degree of differencing.
* q: The size of the moving average window, also called the order of moving average.

The main assumption of AR model is that the time series data is stationary.

A stationary time series is one whose statistical properties such as mean, variance, autocorrelation, etc. are all constant over time.

When the time series data is not stationary, then we convert the non-stationary data before applying AR models. Method we used for making timeseries as Stationary is : Taking the difference between consecutive observations, we also call it a lag-1 difference. For time series with a seasonal component, the lag may be expected to be the period (width) of the seasonality.

White noise of the residuals:

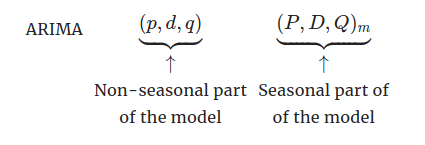
White noise is a process of residuals ϵt that are uncorrelated and follow normal distribution with mean 0 and constant standard deviation. In AR models, one of the main assumptions is the errors follow a white noise.

# **SARIMA Model:**

The difference between ARIMA and SARIMA (SARIMAX) is about the seasonality of the dataset. if your data is seasonal, like it happens after a certain period of time. then we will use SARIMA.

SARIMA stands for Seasonal-ARIMA and it includes seasonality contribution to the forecast. The importance of seasonality is quite evident and ARIMA fails to encapsulate that information implicitly.

The Autoregressive (AR), Integrated (I), and Moving Average (MA) parts of the model remain as that of ARIMA. The addition of Seasonality adds robustness to the SARIMA model. It’s represented as:

[Source](https://otexts.com/fpp2/seasonal-arima.html)

where m is the number of observations per year. We use the uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model.

Similar to ARIMA, the P,D,Q values for seasonal parts of the model can be deduced from the ACF and PACF plots of the data. Let’s implement SARIMA for the same Catfish sales model.

Both ARIMA and SARIMA can be build using Automated way of generating values p,d and q value or manual way of building ACF / PACF graph and observe the numbers for p , d and q.

# **Auto ARIMA Model:**

Since we have already seen that Stationarity can be achieved with Lag of 1 . So d value is 1. And We have taken chosen default value for p and q between 0 and 4, And then generated all possible combination for p,d and q. There are the parameters we have chosen for testing our ARIMA Model and gathered AIC value for all the models:

Examples of the parameter combinations for the Model

Model: (0, 1, 0)

Model: (0, 1, 1)

Model: (0, 1, 2)

Model: (0, 1, 3)

Model: (1, 1, 0)

Model: (1, 1, 1)

Model: (1, 1, 2)

Model: (1, 1, 3)

Model: (2, 1, 0)

Model: (2, 1, 1)

Model: (2, 1, 2)

Model: (2, 1, 3)

Model: (3, 1, 0)

Model: (3, 1, 1)

Model: (3, 1, 2)

Model: (3, 1, 3)

Following are the AIC values for above listed all parameters , when we fit Timeseries data into ARIMA model :

ARIMA(0, 1, 0) - AIC:2267.6630357855465

ARIMA(0, 1, 1) - AIC:2263.060015591336

ARIMA(0, 1, 2) - AIC:2234.4083231280847

ARIMA(0, 1, 3) - AIC:2233.994857754891

ARIMA(1, 1, 0) - AIC:2266.6085393190087

ARIMA(1, 1, 1) - AIC:2235.7550946815218

ARIMA(1, 1, 2) - AIC:2234.5272004521566

ARIMA(1, 1, 3) - AIC:2235.6078038022133

ARIMA(2, 1, 0) - AIC:2260.36574396809

ARIMA(2, 1, 1) - AIC:2233.77762634562

ARIMA(2, 1, 2) - AIC:2213.509212685558

ARIMA(2, 1, 3) - AIC:2232.8363546880437

ARIMA(3, 1, 0) - AIC:2257.72337899794

ARIMA(3, 1, 1) - AIC:2235.4986052031873

ARIMA(3, 1, 2) - AIC:2230.8080878060878

ARIMA(3, 1, 3) - AIC:2221.459263352943

Then we have sported the data, based on AIC value and least value of AIC have following records:

|  | **param** | **AIC** |
| --- | --- | --- |
| **10** | (2, 1, 2) | 2213.509213 |
| **15** | (3, 1, 3) | 2221.459263 |
| **14** | (3, 1, 2) | 2230.808088 |
| **11** | (2, 1, 3) | 2232.836355 |
| **9** | (2, 1, 1) | 2233.777626 |

Lets build the SARIMAX report for the Best parameter (2,1,3)

SARIMAX Results

==============================================================================

Dep. Variable: Sparkling No. Observations: 132

Model: ARIMA(2, 1, 2) Log Likelihood -1101.755

Date: Thu, 25 Aug 2022 AIC 2213.509

Time: 22:13:59 BIC 2227.885

Sample: 01-01-1980 HQIC 2219.351

- 12-01-1990

Covariance Type: opg

==============================================================================

coef std err z P>|z| [0.025 0.975]

------------------------------------------------------------------------------

ar.L1 1.3121 0.046 28.781 0.000 1.223 1.401

ar.L2 -0.5593 0.072 -7.740 0.000 -0.701 -0.418

ma.L1 -1.9917 0.109 -18.217 0.000 -2.206 -1.777

ma.L2 0.9999 0.110 9.109 0.000 0.785 1.215

sigma2 1.099e+06 1.99e-07 5.51e+12 0.000 1.1e+06 1.1e+06

===================================================================================

Ljung-Box (L1) (Q): 0.19 Jarque-Bera (JB): 14.46

Prob(Q): 0.67 Prob(JB): 0.00

Heteroskedasticity (H): 2.43 Skew: 0.61

Prob(H) (two-sided): 0.00 Kurtosis: 4.08

===================================================================================

Warnings:

[1] Covariance matrix calculated using the outer product of gradients (complex-step).

[2] Covariance matrix is singular or near-singular, with condition number 6.98e+27. Standard errors may be unstable.

Lets Build the Diagnostic Plot:



Figure Diagnostic Plot for Auto ARIMA

Lets find the RMSE Value for Auto ARIMA with p, d, q value of 2,1,2:

RMSE: 1299.980240900083

MAPE: 47.099874793454504

# **Auto SARIMA:**

Similar to Auto Arima Model, We have taken all possible combination of p,d,q and P,D,Q values for SARIMA Model, along with the Seasonality parameter. We have taken these combinations:

Examples of the parameter combinations for the Model are

Model: (0, 1, 1)(0, 0, 1, 6)

Model: (0, 1, 2)(0, 0, 2, 6)

Model: (0, 1, 3)(0, 0, 3, 6)

Model: (1, 1, 0)(1, 0, 0, 6)

Model: (1, 1, 1)(1, 0, 1, 6)

Model: (1, 1, 2)(1, 0, 2, 6)

Model: (1, 1, 3)(1, 0, 3, 6)

Model: (2, 1, 0)(2, 0, 0, 6)

Model: (2, 1, 1)(2, 0, 1, 6)

Model: (2, 1, 2)(2, 0, 2, 6)

Model: (2, 1, 3)(2, 0, 3, 6)

Model: (3, 1, 0)(3, 0, 0, 6)

Model: (3, 1, 1)(3, 0, 1, 6)

Model: (3, 1, 2)(3, 0, 2, 6)

Model: (3, 1, 3)(3, 0, 3, 6)

Then we fit our TS data into SARIMA model to calculate the AIC value and sorted AIC values. Following is Sorted AIC value achieved with p,d,q values:

|  | **param** | **seasonal** | **AIC** |
| --- | --- | --- | --- |
| **187** | (2, 1, 3) | (2, 0, 3, 6) | 1632.621824 |
| **59** | (0, 1, 3) | (2, 0, 3, 6) | 1633.327862 |
| **251** | (3, 1, 3) | (2, 0, 3, 6) | 1634.474632 |
| **63** | (0, 1, 3) | (3, 0, 3, 6) | 1635.054387 |
| **123** | (1, 1, 3) | (2, 0, 3, 6) | 1635.424423 |

So the Best parameter for SARIMA Model will be (2,1,3) (2,0,3,6) . RMSE and MEP values are:

RMSE: 743.2229288149066

MAPE: 32.480506654342875

SARIMAX Results

=====================================================

Dep. Variable: Sparkling No. Observations: 132

Model: SARIMAX(2, 1, 3)x(2, 0, 3, 6) Log Likelihood -805.311

Date: Thu, 25 Aug 2022 AIC 1632.622

Time: 22:17:47 BIC 1662.227

Sample: 01-01-1980 HQIC 1644.628

- 12-01-1990

Covariance Type: opg

==============================================================================

coef std err z P>|z| [0.025 0.975]

------------------------------------------------------------------------------

ar.L1 -1.1886 0.032 -37.062 0.000 -1.251 -1.126

ar.L2 -0.9414 0.045 -20.726 0.000 -1.030 -0.852

ma.L1 0.4320 0.488 0.885 0.376 -0.525 1.388

ma.L2 -0.1071 0.340 -0.315 0.753 -0.774 0.560

ma.L3 -0.8582 0.569 -1.509 0.131 -1.973 0.256

ar.S.L6 -0.0007 0.028 -0.026 0.979 -0.056 0.055

ar.S.L12 1.0464 0.020 52.773 0.000 1.008 1.085

ma.S.L6 -0.7316 0.667 -1.097 0.273 -2.039 0.576

ma.S.L12 -1.1091 0.328 -3.384 0.001 -1.751 -0.467

ma.S.L18 0.2487 0.495 0.502 0.616 -0.722 1.219

sigma2 7.147e+04 1.32e-05 5.42e+09 0.000 7.15e+04 7.15e+04

===================================================================================

Ljung-Box (L1) (Q): 0.01 Jarque-Bera (JB): 21.89

Prob(Q): 0.91 Prob(JB): 0.00

Heteroskedasticity (H): 1.42 Skew: 0.34

Prob(H) (two-sided): 0.30 Kurtosis: 5.09

===================================================================================

Warnings:

[1] Covariance matrix calculated using the outer product of gradients (complex-step).

[2] Covariance matrix is singular or near-singular, with condition number 1.26e+26. Standard errors may be unstable.

**Diagnostics Plot:**

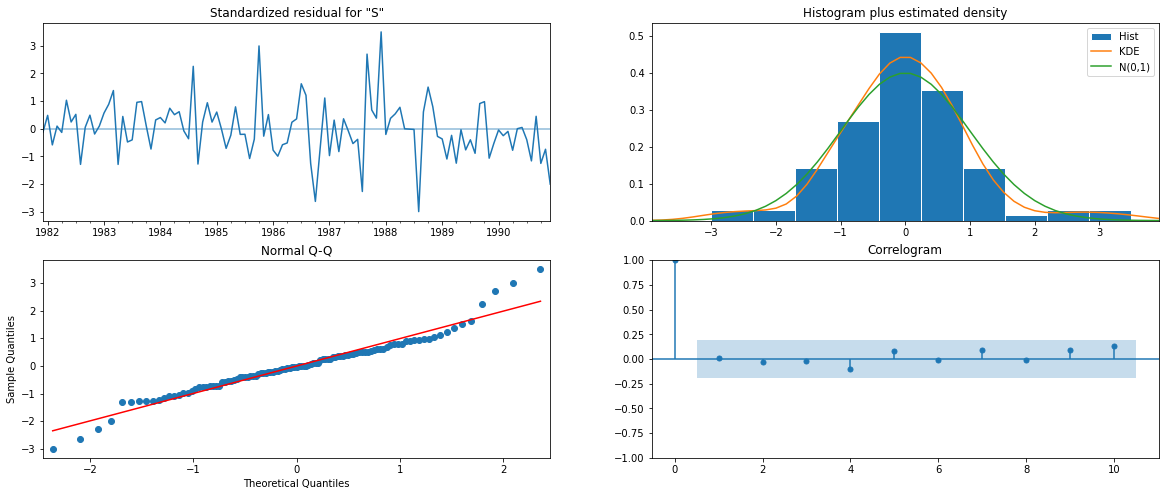


Figure Auto SARIMA Model Diagnostic Plot

# **Build ARIMA/SARIMA models based on the cut-off points of ACF and PACF on the training data and evaluate this model on the test data using RMSE.**

In the above question we build Automated version of ARIMA/SARIMA model, in that we have used all possible combinations of p,d and q values and generated Chosen AIC, RMSE and MEP parameters to identify, which combination of pdq is good for the timeseries data.

One problem with Auto version is, it needs lot of computations and we should have good memory and processing power in the system for iterating all possible values. So to avoid that scenario , we can use building ACF/PACF graph for generating Manual version of ARIMA/SARIMA Model.

Important Component of Automated version of ARIMA Model :

**Auto-Correlation Function (ACF)** or correlogram : A plot of auto-correlation of different lags is called ACF. The plot summarizes the correlation of an observation with lag values. The x-axis shows the lag and the y-axis shows the correlation coeﬃcient between -1 and 1 for negative and positive correlation.

**Partial Auto-Correlation Function (PACF)** Autocorrelation Function (ACF) : A plot of partial auto-correlation for different values of lags is called PACF.The plot summarizes the correlations for an observation with lag values that is not accounted for by prior lagged observations.

Both plots are drawn as bar charts showing the 95% and 99% conﬁdence intervals as horizontal lines. Bars that cross these conﬁdence intervals are therefore more signiﬁcant and worth noting.

Some useful patterns you may observe on these plots are:

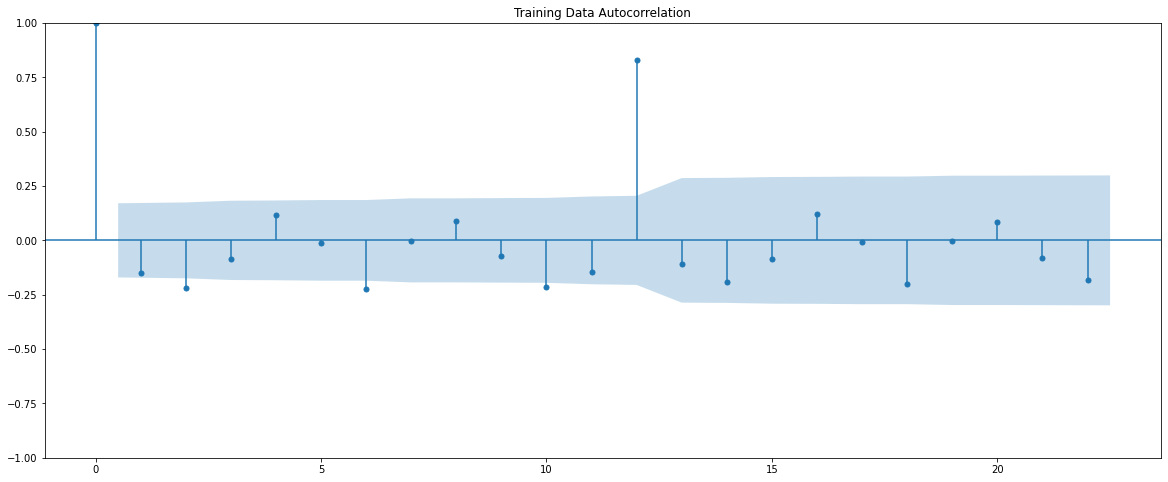
The number of lags is p when:

* The partial auto-correlation, | ρpk | > 1.96 / n−−√ for first p values and cuts off to zero. The auto-correlation function, ρk decreases exponentially.
* The model is AR of order p when the PACF cuts-off after a lag p.
* The model is MA of order p when the ACF cuts-off after a lag q.
* The model is a mix of AR and MA if both the PACF and ACF trail oﬀ and cuts-off at p and q respectively.

For an ARIMA (p,d,q) process, it becomes non-stationary to stationary after differencing it for d times

# **Build Manual ARIMA Model**

ACF graph for ARIMA Model:



PACF Graph for Arima Model

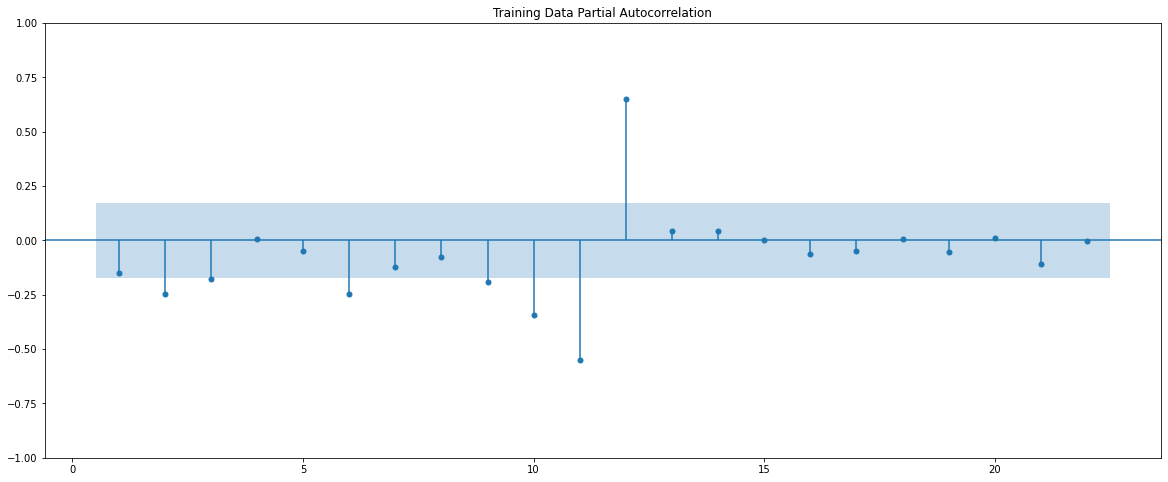


Figure Manual ARIMA ACF/PACF Plot

Here, we have taken alpha=0.05.

* The Auto-Regressive parameter in an ARIMA model is 'p' which comes from the significant lag before which the PACF plot cuts-off to 0.
* The Moving-Average parameter in an ARIMA model is 'q' which comes from the significant lag before the ACF plot cuts-off to 0.

By looking at the above plots, we will take the value of p and q to be 0 and 0 respectively.

SARIMAX Results

==============================================================================

Dep. Variable: Sparkling No. Observations: 132

Model: ARIMA(0, 1, 0) Log Likelihood -1132.832

Date: Thu, 25 Aug 2022 AIC 2267.663

Time: 22:14:01 BIC 2270.538

Sample: 01-01-1980 HQIC 2268.831

- 12-01-1990

Covariance Type: opg

==============================================================================

coef std err z P>|z| [0.025 0.975]

------------------------------------------------------------------------------

sigma2 1.885e+06 1.29e+05 14.658 0.000 1.63e+06 2.14e+06

===================================================================================

Ljung-Box (L1) (Q): 3.07 Jarque-Bera (JB): 198.83

Prob(Q): 0.08 Prob(JB): 0.00

Heteroskedasticity (H): 2.46 Skew: -1.92

Prob(H) (two-sided): 0.00 Kurtosis: 7.65

===================================================================================

Warnings:

[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Diagnostic Plot:

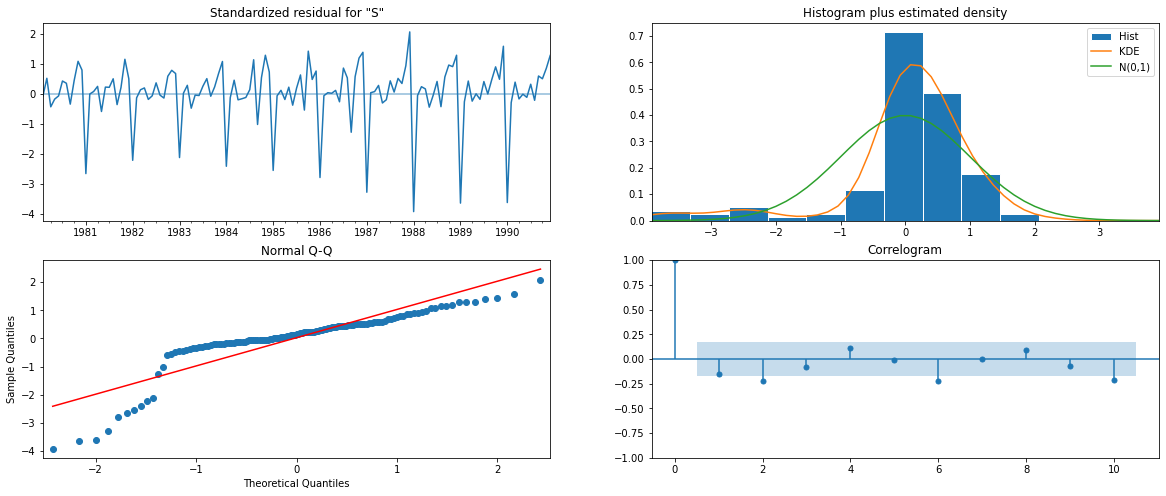


Figure Manual ARIMA Diagnostic Plot

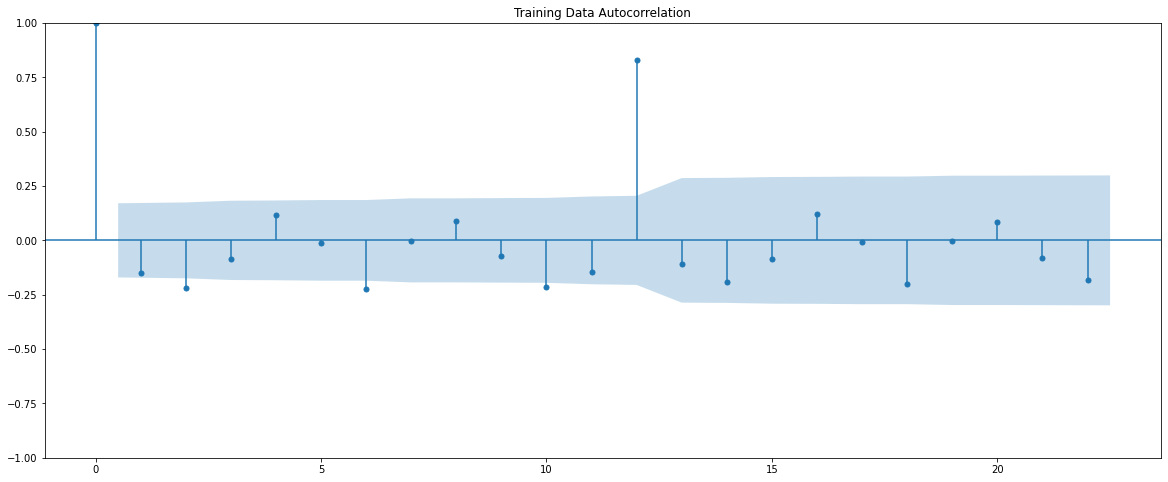
#### Predict on the Test Set using this model and evaluate the model

RMSE: 3864.2793518443914

MAPE: 201.32764950352743

# **Build Manual SARIMA Model:**

We will star from where we left in ARIMA Model, from the above manual version of ARMIA model, we had chosen value for p,d,q as (0,1,0). So we will choose same for SARIMA Model and then add the parameter for Seasonality . Lets plot ACF and PACF plot one more time:



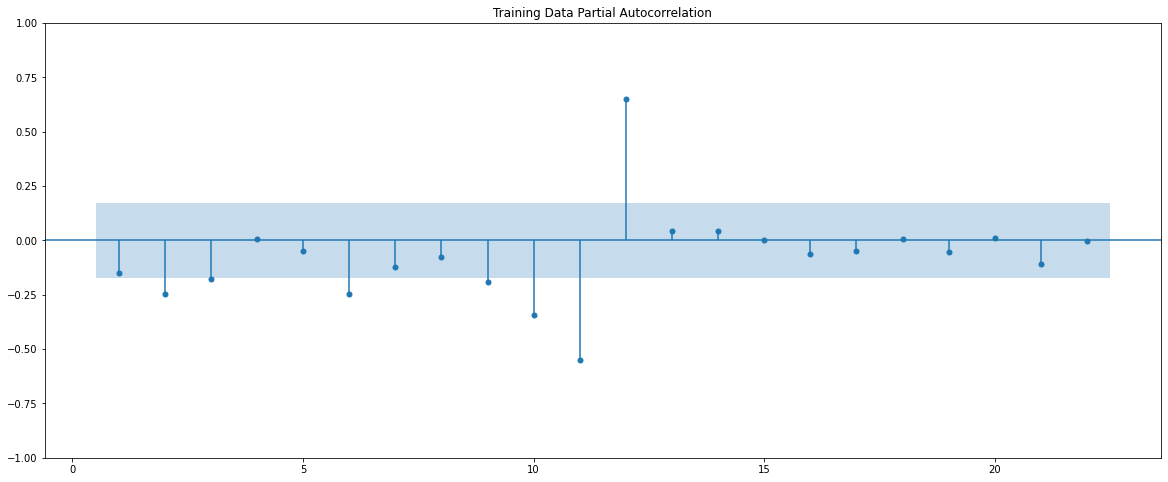


Figure Manual SARIMA ACF/PACF Plot

Here, we have taken alpha=0.05. We are going to take the seasonal period as 3 or its multiple e.g. 6. We are taking the p value to be 3 and the q value also to be 3 as the parameters same as the ARIMA model.

The Auto-Regressive parameter in an SARIMA model is 'P' which comes from the significant lag after which the PACF plot cuts-off to 2.

The Moving-Average parameter in an SARIMA model is 'Q' which comes from the significant lag after which the ACF plot cuts-off to 2.

The Moving-Average parameter in an SARIMA model is 'Q' which comes from the significant lag after which the ACF plot cuts-off to 0.

So final values are as follows: (0,1,0) (2, 1, 2, 6). Lets build the Sarimax result:

SARIMAX Results

================================================

Dep. Variable: Sparkling No. Observations: 132

Model: SARIMAX(0, 1, 0)x(2, 1, [1, 2], 6) Log Likelihood -854.548

Date: Thu, 25 Aug 2022 AIC 1719.096

Time: 22:14:02 BIC 1732.689

Sample: 01-01-1980 HQIC 1724.611

- 12-01-1990

Covariance Type: opg

==============================================================================

coef std err z P>|z| [0.025 0.975]

------------------------------------------------------------------------------

ar.S.L6 -1.1528 0.267 -4.322 0.000 -1.676 -0.630

ar.S.L12 -0.1397 0.271 -0.515 0.607 -0.672 0.392

ma.S.L6 0.1665 0.290 0.573 0.566 -0.403 0.736

ma.S.L12 -0.4794 0.077 -6.211 0.000 -0.631 -0.328

sigma2 2.47e+05 2.44e+04 10.103 0.000 1.99e+05 2.95e+05

===================================================================================

Ljung-Box (L1) (Q): 15.56 Jarque-Bera (JB): 41.05

Prob(Q): 0.00 Prob(JB): 0.00

Heteroskedasticity (H): 1.33 Skew: 0.71

Prob(H) (two-sided): 0.39 Kurtosis: 5.61

===================================================================================

Warnings:

[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Diagnostics Plot:



Figure Manual SAIRMA Diagnostic Plot

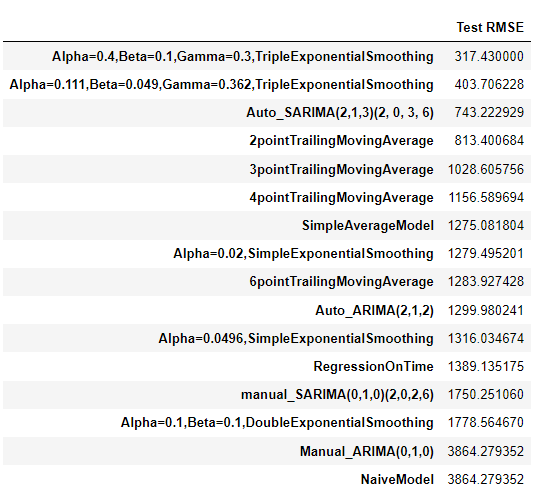
Calculate the RMSE and MAPE value for test data for Manual SARIMA Model:

RMSE: 1750.2510595324502

MAPE: 82.17368719075145

# **Build a table with all the models built along with their corresponding parameters and the respective RMSE values on the test data**

We have built several Models, with different parameters, and calculated the RMSE values for all the parameters . After combining all RMSE values for all the models, following is the table for that:



# **Based on the model-building exercise, build the most optimum model(s) on the complete data and predict 12 months into the future with appropriate confidence intervals/bands.**

Based on the table above, we have see that best model so far given least value of RMSE is Triple exponential Model with parameters of Alpha=0.4,Beta=0.1,Gamma=0.3. It’s RMSE values is only 317.4300 as compared to the worst RMSE value of 3864.279 of Naïve based Model .

Lets build the Triple exponential Smoothing model based on above parameters and fit full data set , we were using only test data for the predictions earlier. We have also calculated the RMSE value for full data set .

RMSE of the Full Model 388.5276378831874

Also as required, we have predicted nest 12 months of data for using above model:

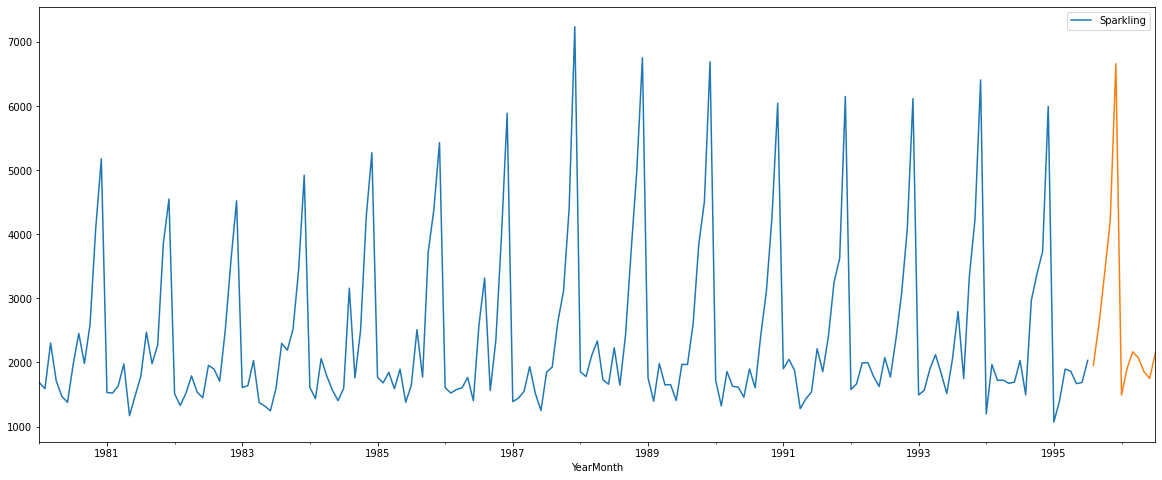


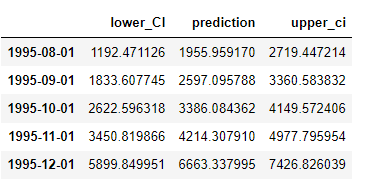
Figure Best Model TES plot for next 12 Months predictions

Plotting the forecast with the confidence band.

**Confidence band for forecasting**: A confidence band is used in statistical analysis to represent the uncertainty in an estimate of a curve or function based on limited or noisy data. The 95% confidence bands enclose the area that you can be 95% sure contains the true curve. It gives you a visual sense of how well your data define the best-fit curve. It is closely related to the 95% prediction bands , which enclose the area that you expect to enclose 95% of future data points. This includes both the uncertainty in the true position of the curve (enclosed by the confidence bands), and also accounts for scatter of data around the curve. Therefore, prediction bands are always wider than confidence bands

For building the Confidence band we need to find lower and upper values of actual predictions with 95% confidentiality . In our case

Following are the Table for Predictions of upper and lower values along with Predicted values:



Plot the time series with next 12 months of unseen data :

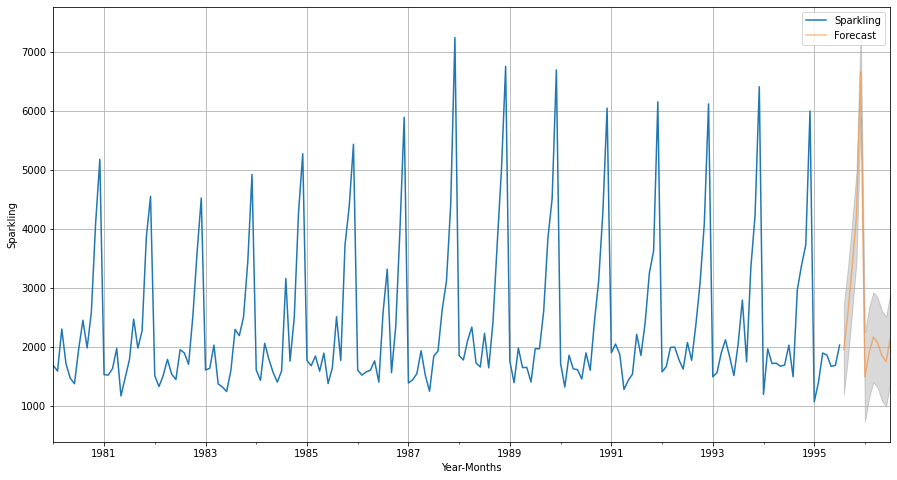


Figure TES Model next 12 Month Predictions with Confidence Band

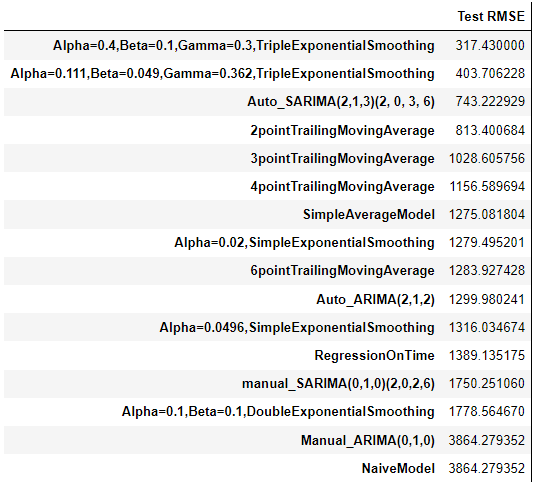
# **Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales.**

Analysis about the data:

1. This is a Sales data for Sparkling wine product of Wine manufacturing company , with the mean sales of 2302.417 in per Month and Minimum Sales of 1070 . whereas max Sales of 7242 in any Month.
2. Data consist of 187 data points
3. It seems to be contained seasonality
4. We also notice that there is small portion of the trend in data but altogether its not visible very clearly by plotting it in years.
5. Minimum sales for the data in Any month are 1070 and Max sales of Rose wines in any month is 7242
6. Year 1988 has highest sales
7. We also notice that 1995 has lowest sales , but this is not correct because we don’t have full data point for this year
8. We have also seen that all year sales have outlier Sales in a few months of each year.
9. We have done upsampling of the data for Quarterly, yearly and a Decade sales, There are seasonality in the data, and it gets flatternout when we Upsample the data
10. We have also done Downsampling of the data for analyzing Daily Sales.
11. We see the decrease in Sales for Sparkling wine in initial first 3-4 years , then again Sales increased , which remain almost constant and no big increase in sales .
12. Boxplot helps to check the outliers in each year and month and we see there are outliers in almost all the year as per the box plot.
13. Average sales are lowest in the year of 1995
14. Monthly plot contains outliers in the month of February and July month Sales
15. There are Highest sales in the Month of December followed by 2nd highest Sales in November Month, which indicates year end party and vacation celebration sales
16. As yearend comes, sales start growing and it is about similar sales , which is very less for the first 6 months of any year
17. We have accumulated all year’s data for each month and plotted it against each month.
18. We see that there are High Sales in each month at the Mid of the month and then Fall down till end of the month.
19. We also see decrease and lowest Sales in every Start of the month and then slightly cover up , goes highest in Mid of the month.-
20. January have low number of wine sales.
21. Highest sales observed in year 1988 , then goes down again for 89 and 90s.
22. After 1994, Sales goes down rapidly, but this is all because we don’t have data for full year 1995.
23. The resampled yearly or annual series have smoothened out the seasonality and have only been able to capture the year on year trend where there was.
24. The quarterly series is able to catch the seasonality in the data.
25. If we take the resampling period to be 10 years or a decade, we see that the seasonality present has been smoothed over and it is only giving an estimate of the trend.
26. There are always Increase in decade data, which gets flattened out in future decades
27. We have built 2 Models of the data Additive trend as well as Multiplicative Trend .
28. From the ‘additive’ decomposition, there is seasonality in the data. Which is at the End of the year when Sales goes High, and at the Start of the year, sales goes down for following years.
29. Sales trend is almost constant for 2nd half of the data and don’t see , a big difference in total Sales

Forecasting analysis:

We have built 17 Models on our data and Compared all of them based on RMSE value for each Model: Results with the parameters used for comparisons are as follows:



#### It is clear that Alpha=0.4,Beta=0.1,Gamma=0.3,TripleExponentialSmoothing has the lower RMSE and NaiveModel has the hieghest RMSE value of 3864.27935

Based on above listed table, we have found that Alpha=0.1,Beta=0.2,Gamma=0.1,TripleExponentialSmoothing has the lower RMSE of 9.22 and NaiveModel has the highest RMSE value of 79.718

To find the most optimum model, we run the model on the full data . We predict for the next 12 months for next years. RMSE of the Full Model 388.5276378831874

**Recommendations:**

From the forecast it is being predicted that the sales will remain steady and it will follow same pattern, as it has been doing for previous years, where Year end will be good and Start of new year will be bad for the sales.

Company should start promotions and give more deals, to increase sales for all the months, and not only for the end of the year.

Also as observed before sales will increase till December and then there is a likely chance of sharp drop in January followed by a very gradual increase in the middle of 1996. This is similar observations during Monthly and yearly plot with seasonality as well.

Also, wine producing company should take efforts or run marketing campaign for promoting wine productions at the start of the year and especially in the period of Jan-July 1996 Months.

Company should also analyze further various business methods like marketing, optimization of raw materials and for better profitability.