greatlearning

Learning for Life

BUSINESS REPORT TIME SERIES FORECASTING

SPARKLING WINE SALES TIME SERIES

PREPARED BY - RUPESH KUMAR / SUBMISSION DATE : 19/12/2021

Table of Contents

List of Figures

List of Figures

List of Tables

List of Tables

Problem Statement of Sparkling Wine Sales Time Series :

For this particular assignment, the data of different types of wine sales in the 20th century is to be analysed. Both of these data are from the same company but of different wines. As an analyst in the ABC Estate Wines, you are tasked to analyse and forecast Wine Sales in the 20th century.

Executive Summary

ABC Estate wines have the data of different types of wine sales in the 20th century Both of these data are from the same company but of different wines **Sparkling Wine & Rose Wine** Sales. As an analyst in the ABC Estate Wines, Our tasked to analyse and forecast different types of Wine Sales in the 20th century. Here we are analyse and forecast **Sparkling Wine Sales** in the 20th century.

Introduction

The purpose of this whole exercise is to explore the dataset , analyse and forecast **Sparkling Wine Sales** in the 20th century.Here we perform the exploratory data analysis & apply various time series forecasting models like **Linear Regression , Navie Forecast ,Simple Average , Moving Average and various kind of exponential smoothing models like (Simple , Double , Triple Exponential) and ARIMA / SARIMA** models on the Sparkling Wine Sales dataset and check their **RMSE** on the test data , model which gives the least **RMSE** will be the final model for us to analyse and forecast the Sparkling Wine Sales in the 20th century.

1. Read the data as an appropriate Time Series data and plot the data.

Checking the Records of the Dataset -
Head of the Dataset - First 10 Records of the Dataset.
Tail of the Dataset - Last 10 Records of the Dataset.

	YearMonth	Sparkling
0	1980-01	1686
1	1980-02	1591
2	1980-03	2304
3	1980-04	1712
4	1980-05	1471
5	1980-06	1377
6	1980-07	1966
7	1980-08	2453
8	1980-09	1984
9	1980-10	2596

	YearMonth	Sparkling
177	1994-10	3385
178	1994-11	3729
179	1994-12	5999
180	1995-01	1070
181	1995-02	1402
182	1995-03	1897
183	1995-04	1862
184	1995-05	1670
185	1995-06	1688
186	1995-07	2031

Tab:1 Records of the Dataset Head & Tail

Plot of the Data

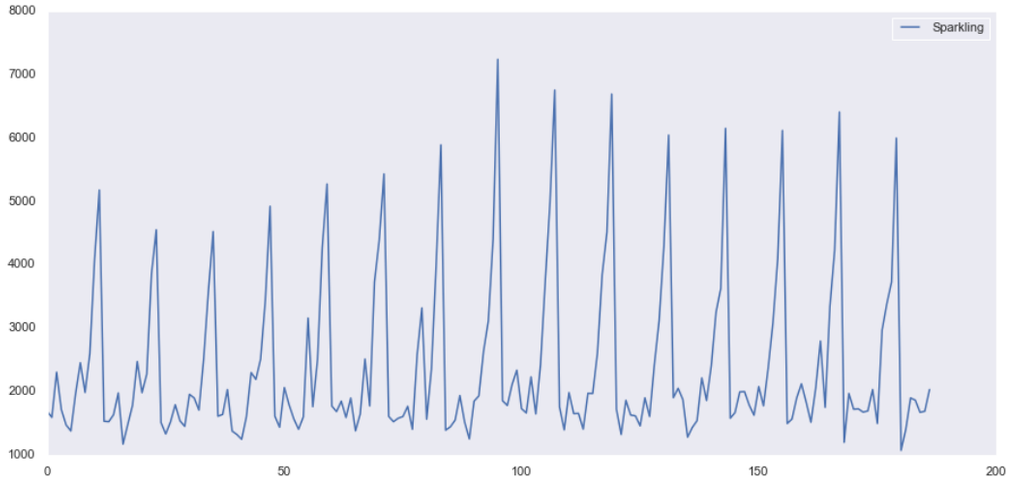

Fig : 1 Plot of the Original Data

Note :

Though the above plot looks like a Time Series plot, notice that the X-Axis is not time. In order to make the X-Axis as a Time Series, we need to pass the date range manually through a command in Pandas.

Adding the Time Stamp into Dataset.

Note :

The time stamps is defined as a monthly Time Series after looking at the data.Data given to us is start from Jan 1980 to July 1995 , In order to make the X-Axis as a Time Series, we need to pass the date range from start= '1/1/1980', end='8/1/1995' as we know that the when we using this data_range function always creates the time stamp with the last day of that month,when we include this into dataset we will see that the first observation of the time stamp will be 31 Jan 1980 , so in order to include the July we will write the end period as end='8/1/1995' now it automatically take the last day of July 31 for this dataset,as we know the frequency is "monthly" so we set it freq = "M".

```
DatetimeIndex(['1980-01-31', '1980-02-29', '1980-03-31', '1980-04-30',
               '1980-05-31', '1980-06-30', '1980-07-31', '1980-08-31',
               '1980-09-30', '1980-10-31',
               ...
               '1994-10-31', '1994-11-30', '1994-12-31', '1995-01-31',
               '1995-02-28', '1995-03-31', '1995-04-30', '1995-05-31',
               '1995-06-30', '1995-07-31'],
              dtype='datetime64[ns]', length=187, freq='M')
```

Tab:2 Time Stamp

Observation:

Last date of every month is being used here from 31 Jan 1980 till 31 July 1995.

Checking the Records of the Dataset after adding Time_Stamp.

Checking the Records of the Dataset -
Head of the Dataset - First 10 Records of the Dataset.
Tail of the Dataset - Last 10 Records of the Dataset.

	YearMonth	Sparkling	Time_Stamp
0	1980-01	1686	1980-01-31
1	1980-02	1591	1980-02-29
2	1980-03	2304	1980-03-31
3	1980-04	1712	1980-04-30
4	1980-05	1471	1980-05-31
5	1980-06	1377	1980-06-30
6	1980-07	1966	1980-07-31
7	1980-08	2453	1980-08-31
8	1980-09	1984	1980-09-30
9	1980-10	2596	1980-10-31

	YearMonth	Sparkling	Time_Stamp
177	1994-10	3385	1994-10-31
178	1994-11	3729	1994-11-30
179	1994-12	5999	1994-12-31
180	1995-01	1070	1995-01-31
181	1995-02	1402	1995-02-28
182	1995-03	1897	1995-03-31
183	1995-04	1862	1995-04-30
184	1995-05	1670	1995-05-31
185	1995-06	1688	1995-06-30
186	1995-07	2031	1995-07-31

Tab:3 Records of the Dataset Head & Tail with Time Stamp

Final Dataset for Time Series Forecasting

	Sparkling Wine Sales
Time_Stamp	
1980-01-31	1686
1980-02-29	1591
1980-03-31	2304
1980-04-30	1712
1980-05-31	1471

Tab:4 Head of the Final Dataset for Time Series Forecasting of Sparkling Wine Sales

Note:
In the final dataset we have Time Stamp as index and records of the Sparkling Wine Sales , we also drop the year-month column as it is not useful for us. Now we successfully creates the Time Series object, let us go ahead and analyze the Time Series plot that we got.

Plot of the Data After Adding Time_Stamp.

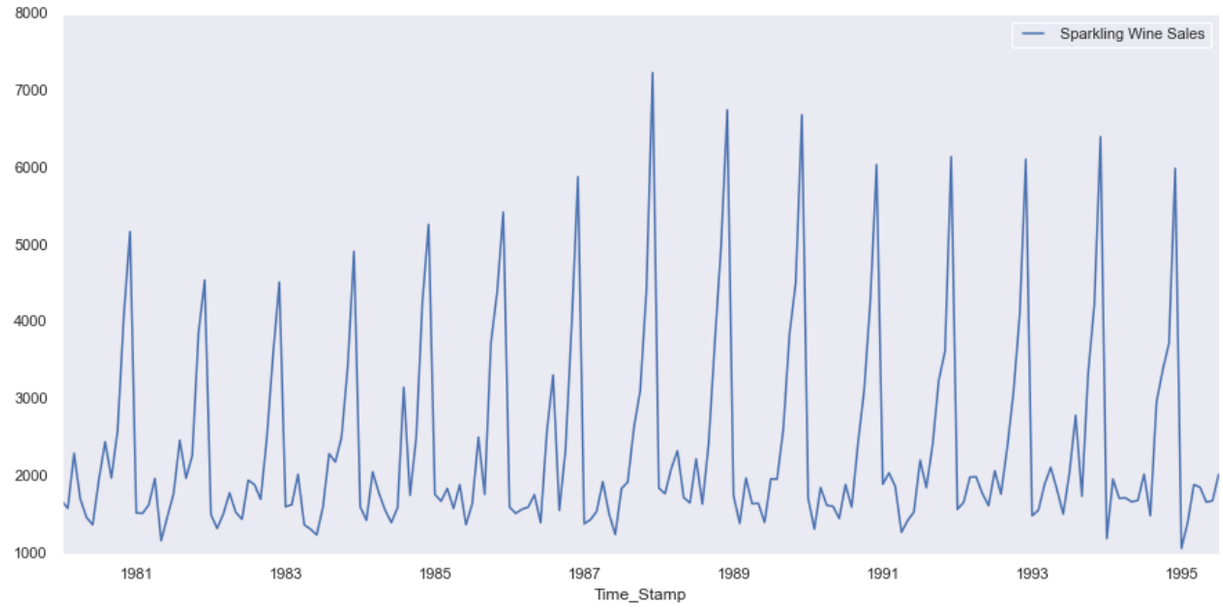

Fig : 2 Plot of the Data with Time Stamp

Insights

We notice that there is some kind of increasing trend in the initial years which stabilizes as the years (or more specifically the months in each of the years) progresses. There is some kind of seasonality associated in the data as well.

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

Exploratory Data Analysis -

Checking the Summary of the Dataset.

The describe ( ) method computes and displays summary statistics for a Python dataframe.From the above table we can infer the count,mean, std , 25% , 50% ,75% and min & max values of the Sparkling Wine Sales column present in the dataset.

Statistical Summary	Values
count	187.0
mean	2402.417112
std	1295.11154
min	1070.0
25%	1605.0
50%	1874.0
75%	2549.0
max	7242.0

Tab:5 Summary of the Dataset

Insights

- Sparkling Wine Sales ranges from a minimum of 1070 to maximum of 7242.
- Mean of the Sparkling Wine Sales is around 2402.417112.
- Standard Deviation of the Sparkling Wine Sales is 1295.111540.
- 25% , 50% (median) and 75 % of Sparkling Wine Sales are 1605 , 1874 and 2549.

Checking the Appropriateness of Data-types & Information of the Dataframe.

The info( ) function is used to print a concise summary of a DataFrame. This method prints information about a DataFrame including the index d-type and column d-types, non-null values and memory usage.

S.No.	Features / Columns	Non-Null Count	Dtype	Memory Usage
1	Sparkling Wine Sales	187 non-null	int64	2.9KB

Tab:6 Appropriateness of Datatypes & Information of the Dataframe

Insights

From the above results we can see that there is no null values present in the dataset.Their are total 187 entries of Sparkling wines Sales as per Monthly frequency in this dataset,indexed from 1980-01-31 to 1995-07-31.Sparkling Wine Sales column have d-type of int64. Memory used by the dataset: 2.9 KB.

Checking the Null Values in the Dataset.

S.No.	Features / Columns	Null Count
1	Sparkling Wine Sales	0

Tab:7 Checking Null Values

Insights

There is No Null Present in the Dataset.

Checking the Shape of the Dataframe.

No. of Rows	No. of Columns
187	1

Tab:8 Shape of the Dataset

Insights

The Sparkling.csv data set have 187 observations (rows) and 1 variable (column named as Sparkling Wine Sales) in the dataset.

Data Visualization of the Time Series

Note

A box-plot gives a nice summary of one or several numeric variables. The line that divides the box into 2 parts represents the median of the data. The end of the box shows the upper and lower quartiles. The extreme lines show the highest and lowest value excluding outliers.

Now, let us plot a box and whisker (1.5* IQR) plot to understand the spread of the data and check for outliers in each year, if any.

Year on Year boxplot for the Sparkling Wine Sales.

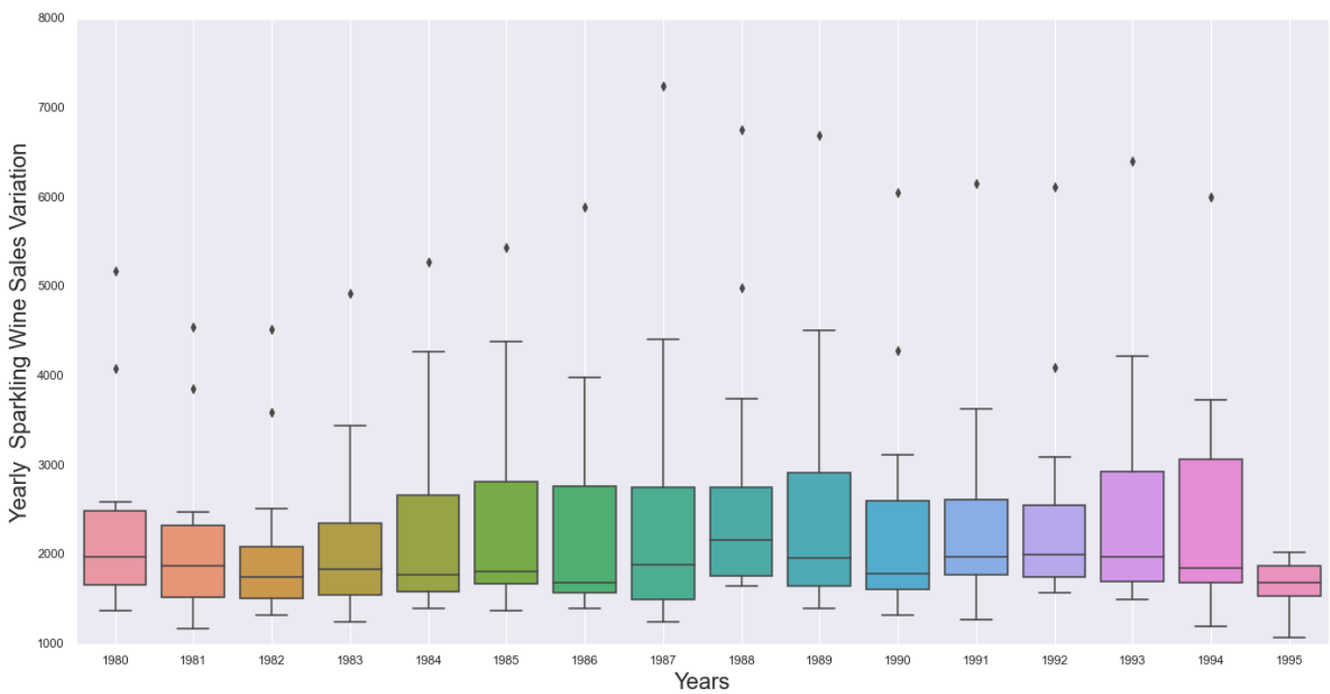

Fig : 3 Year on Year boxplot for the Sparkling Wine Sales.

Insights

- As we got to know from the Time Series plot, the box-plots over here also indicates a measure of trend being present. Also, we see that the Sparkling Wine Sales have outliers for the years.
- Box-plot of Year 1988 have max median value,we can clearly infer that year 1988 have maximum Sparkling Wine Sales.
- Box-plot of Year 1995 have min median value,we can clearly infer that year 1995 have minimum Sparkling Wine Sales.

Monthly Box-Plot for the Sparkling Wine Sales Taking all the Years into Account

Note

Since this is a monthly data, let us plot a box and whisker (1.5* IQR) plot to understand the spread of the data and check for outliers for every month across all the years, if any.

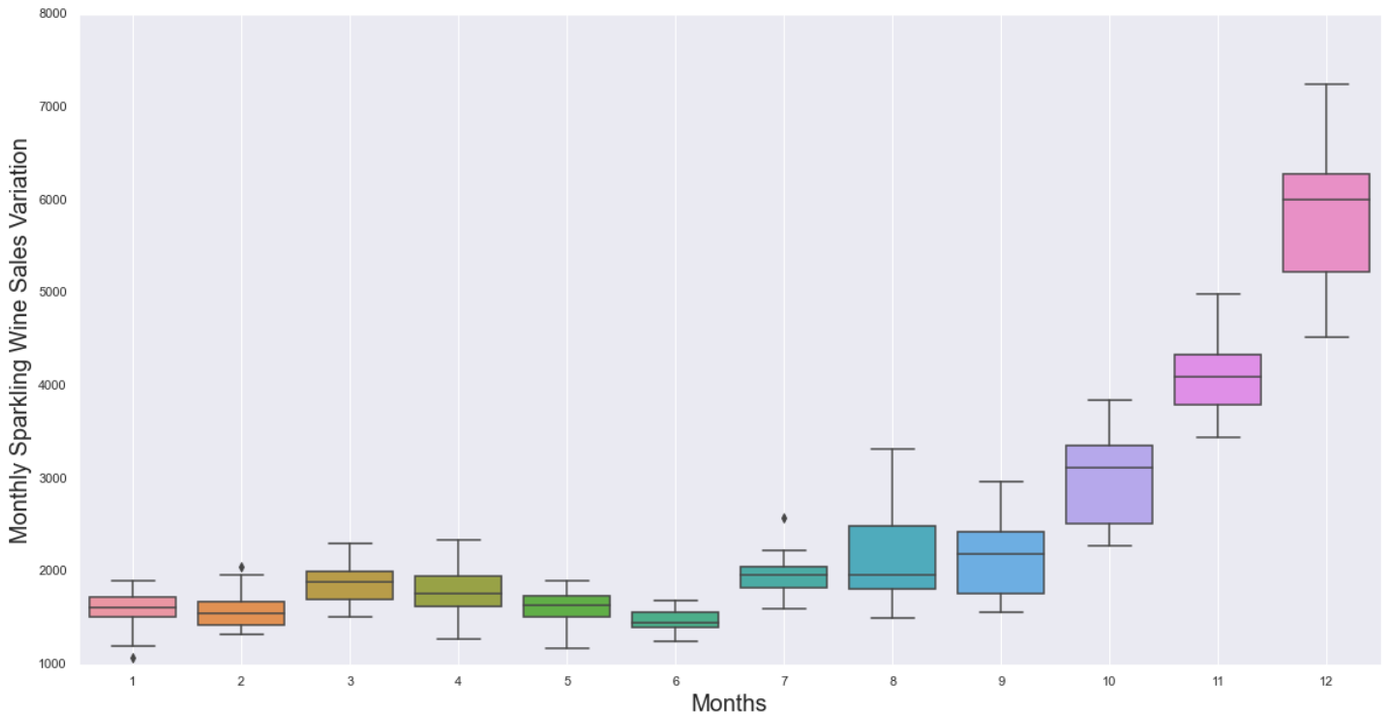

Fig : 4 Monthly Box-Plot for the Sparkling Wine Sales Taking all the Years into Account

Insights

- The Box-Plots for the monthly Sparkling Wine Sales for different years doesn't show too much outliers only month 1 , 2 & 7 show a few outliers , rest doesn't show any outliers.
- From September to December the Sparkling Wine Sales increasing , so this the period where the Sparkling Wine Sales is highest.
- There is seasonality also every year from September to December the Sparkling Wine Sales increasing.
- In June month we have lowest sales of the Sparkling Wine.

Monthplot of Sparkling Wine Sales Time Series

Note

This plot shows the variations of the Sparkling Wine Sales values across the months & this red line is the mean value of the Sparkling Wine Sales for every month.

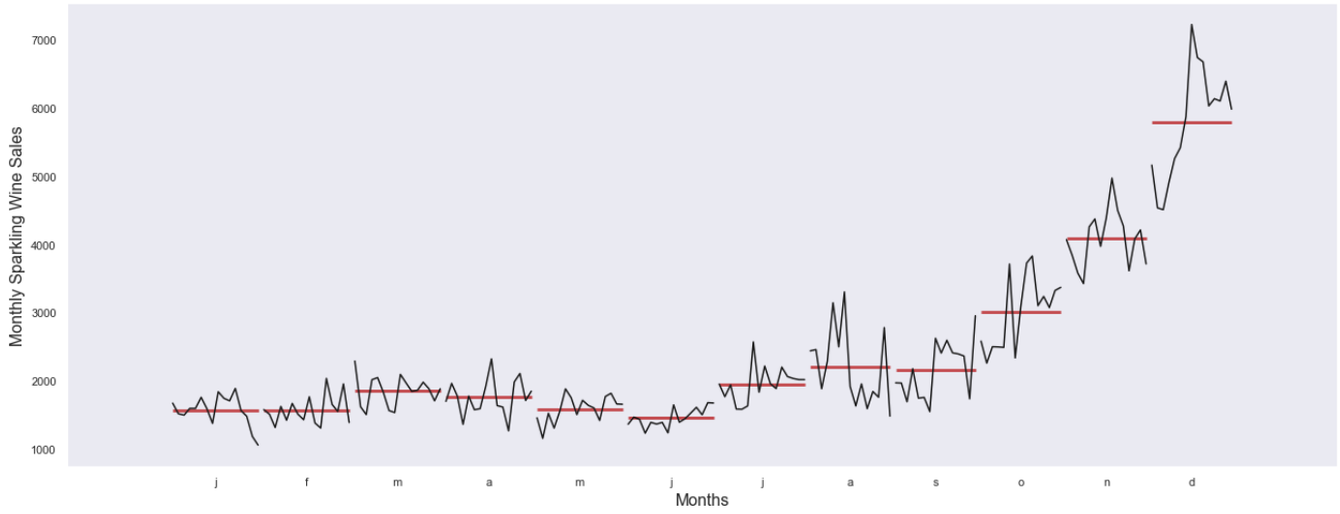

Fig : 5 Monthplot of Sparkling Wine Sales Time Series

Insights

- As noticed in the above box-plot we get same result from here too. From September to December Sparkling Wine Sales goes on increasing.
- December month have the highest sales of the Sparkling Wine while June month have lowest sales of the Sparkling Wine.

Time Series Plot for different months for different years.

Time_Stamp	1	2	3	4	5	6	7	8	9	10	11	12
Time_Stamp												
1980	1686.0	1591.0	2304.0	1712.0	1471.0	1377.0	1966.0	2453.0	1984.0	2596.0	4087.0	5179.0
1981	1530.0	1523.0	1633.0	1976.0	1170.0	1480.0	1781.0	2472.0	1981.0	2273.0	3857.0	4551.0
1982	1510.0	1329.0	1518.0	1790.0	1537.0	1449.0	1954.0	1897.0	1706.0	2514.0	3593.0	4524.0
1983	1609.0	1638.0	2030.0	1375.0	1320.0	1245.0	1600.0	2298.0	2191.0	2511.0	3440.0	4923.0
1984	1609.0	1435.0	2061.0	1789.0	1567.0	1404.0	1597.0	3159.0	1759.0	2504.0	4273.0	5274.0
1985	1771.0	1682.0	1846.0	1589.0	1896.0	1379.0	1645.0	2512.0	1771.0	3727.0	4388.0	5434.0
1986	1606.0	1523.0	1577.0	1605.0	1765.0	1403.0	2584.0	3318.0	1562.0	2349.0	3987.0	5891.0
1987	1389.0	1442.0	1548.0	1935.0	1518.0	1250.0	1847.0	1930.0	2638.0	3114.0	4405.0	7242.0
1988	1853.0	1779.0	2108.0	2336.0	1728.0	1661.0	2230.0	1645.0	2421.0	3740.0	4988.0	6757.0
1989	1757.0	1394.0	1982.0	1650.0	1654.0	1406.0	1971.0	1968.0	2608.0	3845.0	4514.0	6694.0
1990	1720.0	1321.0	1859.0	1628.0	1615.0	1457.0	1899.0	1605.0	2424.0	3116.0	4286.0	6047.0
1991	1902.0	2049.0	1874.0	1279.0	1432.0	1540.0	2214.0	1857.0	2408.0	3252.0	3627.0	6153.0
1992	1577.0	1667.0	1993.0	1997.0	1783.0	1625.0	2076.0	1773.0	2377.0	3088.0	4096.0	6119.0
1993	1494.0	1564.0	1898.0	2121.0	1831.0	1515.0	2048.0	2795.0	1749.0	3339.0	4227.0	6410.0
1994	1197.0	1968.0	1720.0	1725.0	1674.0	1693.0	2031.0	1495.0	2968.0	3385.0	3729.0	5999.0
1995	1070.0	1402.0	1897.0	1862.0	1670.0	1688.0	2031.0	NaN	NaN	NaN	NaN	NaN

Tab:9 Pivot Table of different months for different years.

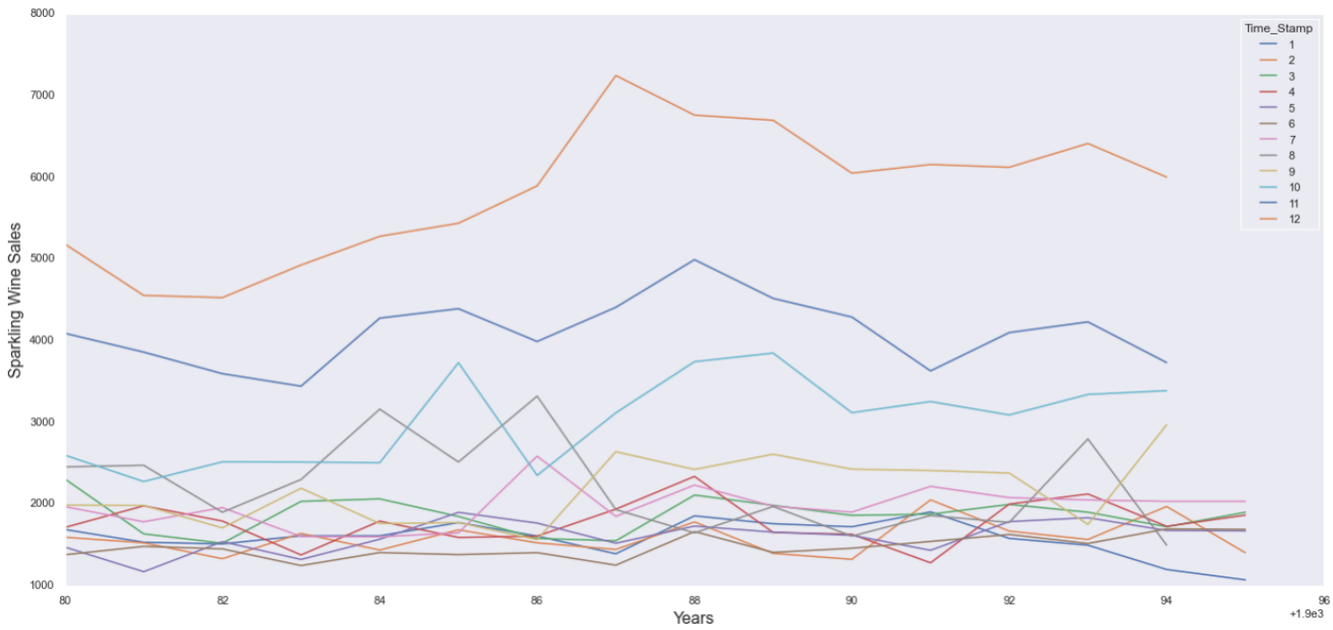

Fig : 6 Time Series Plot for different months for different years.

Insights

- From the above plot we clearly infer that December month have highest sales of Sparking Wine.
- June month have the lowest sales of the Sparkling Wine.

Decomposition of the Sparkling Wine Time Series

Additive Decomposition Model -

Additive model analysis is a newly emerged approach for time-series modeling. ... Under this setting, the given time-series would be decomposed into four components: trend, seasonality, cyclic patterns, and a random component. The formula is as follows: $y(t)=g(t)+s(t)+h(t)+\epsilon(t)$.

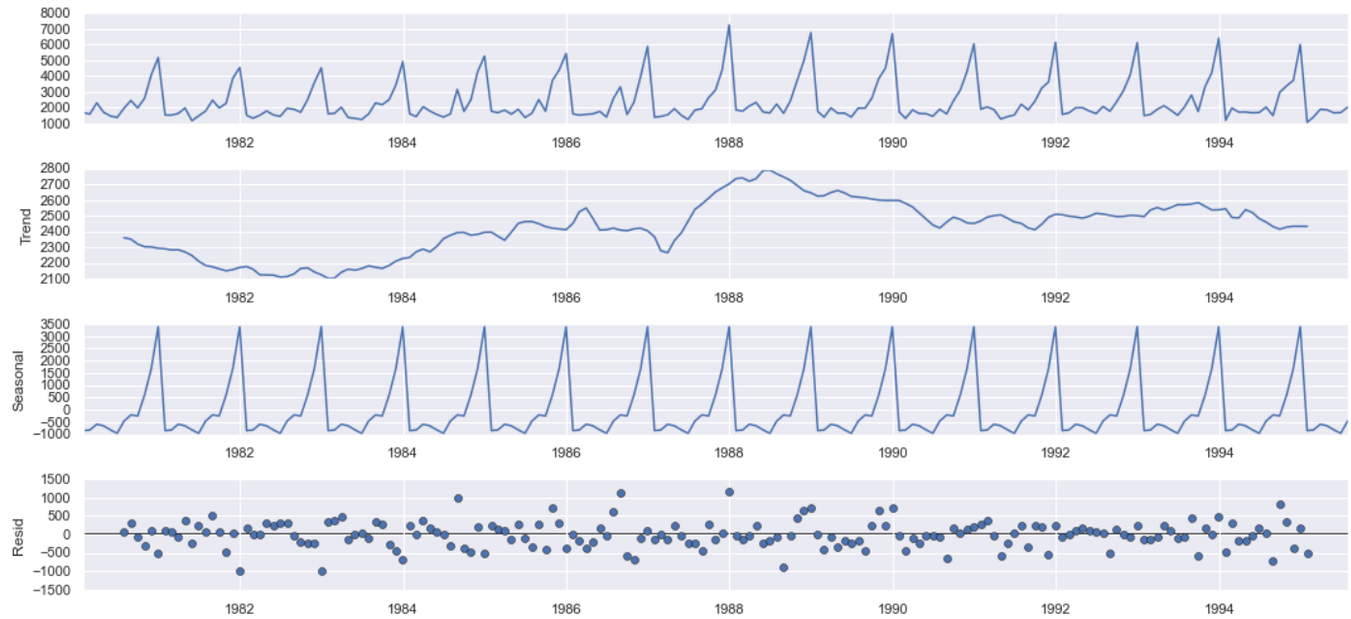

Fig : 7 Additive Decomposition

Insights

- As per the 'Additive' Decomposition Model, we see that there is a pronounced trend in the earlier years of the data. There is a seasonality as well.
- Errors are not randomly distributed ,showing some kind of pattern , as we noticed that errors increases as year increases after 1985 onwards errors are high , which is not a good which is errors are not random.
- We see that the residuals are located around 0 from the plot of the residuals in the decomposition.
- Even the peaks of the original time series is not constant having change in the values.

Values for Trend , Seasonality & Residuals of Additive Decomposition Model

Trend

Time_Stamp	
1980-01-31	NaN
1980-02-29	NaN
1980-03-31	NaN
1980-04-30	NaN
1980-05-31	NaN
1980-06-30	NaN
1980-07-31	2360.666667
1980-08-31	2351.333333
1980-09-30	2320.541667
1980-10-31	2303.583333
1980-11-30	2302.041667
1980-12-31	2293.791667
1981-01-31	2290.375000
1981-02-28	2283.458333
1981-03-31	2284.125000
1981-04-30	2270.541667
1981-05-31	2247.500000
1981-06-30	2211.750000
1981-07-31	2184.750000
1981-08-31	2175.833333
1981-09-30	2162.958333
1981-10-31	2150.416667
1981-11-30	2157.958333
1981-12-31	2171.958333

Name: trend, dtype: float64

Seasonality

Time_Stamp	
1980-01-31	-854.260599
1980-02-29	-830.350678
1980-03-31	-592.356630
1980-04-30	-658.490559
1980-05-31	-824.416154
1980-06-30	-967.434011
1980-07-31	-465.502265
1980-08-31	-214.332821
1980-09-30	-254.677265
1980-10-31	599.769957
1980-11-30	1675.067179
1980-12-31	3386.983846
1981-01-31	-854.260599
1981-02-28	-830.350678
1981-03-31	-592.356630
1981-04-30	-658.490559
1981-05-31	-824.416154
1981-06-30	-967.434011
1981-07-31	-465.502265
1981-08-31	-214.332821
1981-09-30	-254.677265
1981-10-31	599.769957
1981-11-30	1675.067179
1981-12-31	3386.983846

Name: seasonal, dtype: float64

Residual

Time_Stamp	
1980-01-31	NaN
1980-02-29	NaN
1980-03-31	NaN
1980-04-30	NaN
1980-05-31	NaN
1980-06-30	NaN
1980-07-31	70.835599
1980-08-31	315.999487
1980-09-30	-81.864401
1980-10-31	-307.353290
1980-11-30	109.891154
1980-12-31	-501.775513
1981-01-31	93.885599
1981-02-28	69.892345
1981-03-31	-58.768370
1981-04-30	363.948892
1981-05-31	-253.083846
1981-06-30	235.684011
1981-07-31	61.752265
1981-08-31	510.499487
1981-09-30	72.718932
1981-10-31	-477.186624
1981-11-30	23.974487
1981-12-31	-1007.942179

Name: resid, dtype: float64

Tab:10 Values for Trend , Seasonality & Residuals of Additive Decomposition Model

Multiplicative Decomposition Model

Multiplicative Model: Yt = Tt * St * It is considered when the resultant time series is the product of the components. A series may be considered multiplicative series when the seasonal fluctuations increase as trend increases. A multiplicative time series can be transformed into an additive series by taking log transformation i.e. log(Yt) = log(Tt) + log(St) + log(It)

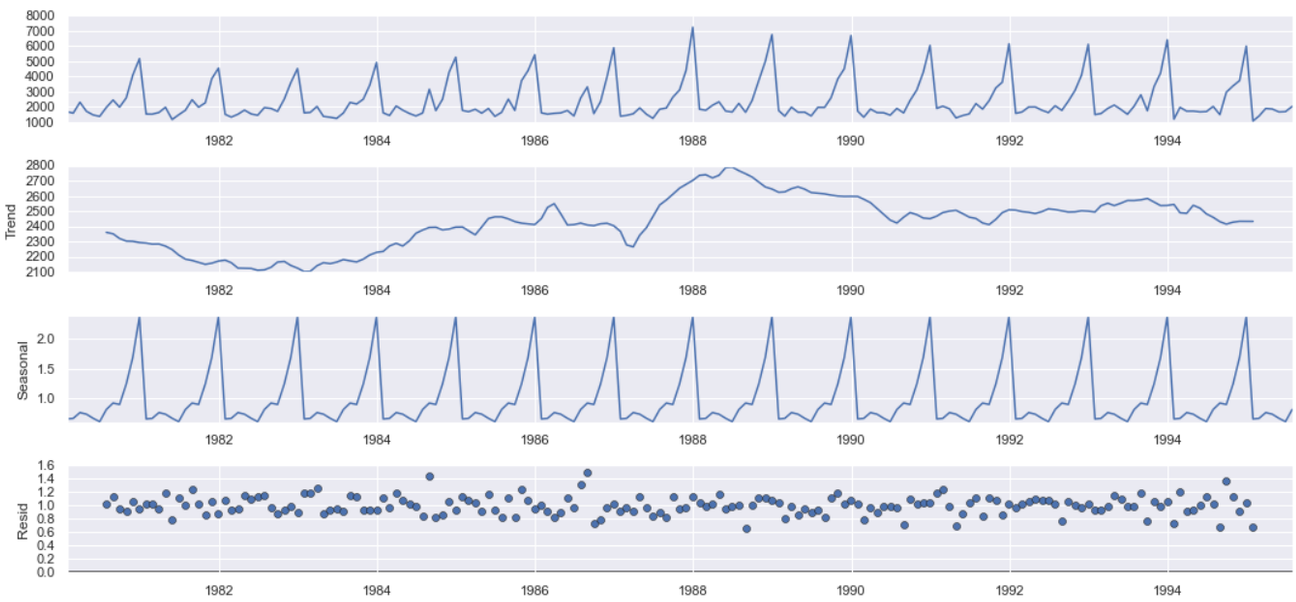

Fig : 8 Multiplicative Decomposition

Insights

- As per the 'Multiplicative' Decomposition Model, we see that there is a pronounced trend in the earlier years of the data. There is a seasonality as well.
- For the multiplicative series, we see that a lot of residuals are located around 1.

Values for Trend , Seasonality & Residuals of Multplicative Decomposition Model

Trend

Time_Stamp	
1980-01-31	NaN
1980-02-29	NaN
1980-03-31	NaN
1980-04-30	NaN
1980-05-31	NaN
1980-06-30	NaN
1980-07-31	2360.666667
1980-08-31	2351.333333
1980-09-30	2320.541667
1980-10-31	2303.583333
1980-11-30	2302.041667
1980-12-31	2293.791667

Name: trend, dtype: float64

Seasonality

Time_Stamp	
1980-01-31	0.649843
1980-02-29	0.659214
1980-03-31	0.757440
1980-04-30	0.730351
1980-05-31	0.660609
1980-06-30	0.603468
1980-07-31	0.809164
1980-08-31	0.918822
1980-09-30	0.894367
1980-10-31	1.241789
1980-11-30	1.690158
1980-12-31	2.384776

Name: seasonal, dtype: float64

Residual

Time_Stamp	
1980-01-31	NaN
1980-02-29	NaN
1980-03-31	NaN
1980-04-30	NaN
1980-05-31	NaN
1980-06-30	NaN
1980-07-31	1.029230
1980-08-31	1.135407
1980-09-30	0.955954
1980-10-31	0.907513
1980-11-30	1.050423
1980-12-31	0.946770

Name: resid, dtype: float64

Tab:11 Values for Trend , Seasonality & Residuals of Multiplicative Decomposition Model

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

Spliting of the data into Train and Test.

Note

Training Data is till the end of 1990. Test Data is from the beginning of 1991 to the last time stamp provided.

Checking the Records of the Train & Test Data.

First few rows of Training Data

Time_Stamp	Sparkling Wine Sales
1980-01-31	1686
1980-02-29	1591
1980-03-31	2304
1980-04-30	1712
1980-05-31	1471

Last few rows of Training Data

Time_Stamp	Sparkling Wine Sales
1990-08-31	1605
1990-09-30	2424
1990-10-31	3116
1990-11-30	4286
1990-12-31	6047

First few rows of Test Data

Time_Stamp	Sparkling Wine Sales
1991-01-31	1902
1991-02-28	2049
1991-03-31	1874
1991-04-30	1279
1991-05-31	1432

Last few rows of Test Data

Time_Stamp	Sparkling Wine Sales
1995-03-31	1897
1995-04-30	1862
1995-05-31	1670
1995-06-30	1688
1995-07-31	2031

Tab:12 Checking the Records of the Train & Test Data.

Checking the Shape of the Train & Test Data.

Shape attribute tells us number of observations and variables we have in the data set. It is used to check the dimension of data.

No. of Rows	No. of Columns
132	1

Tab:13 Shape of the Train Data

No. of Rows	No. of Columns
55	1

Tab:14 Shape of the Test Data

Plot of Train & Test Data.

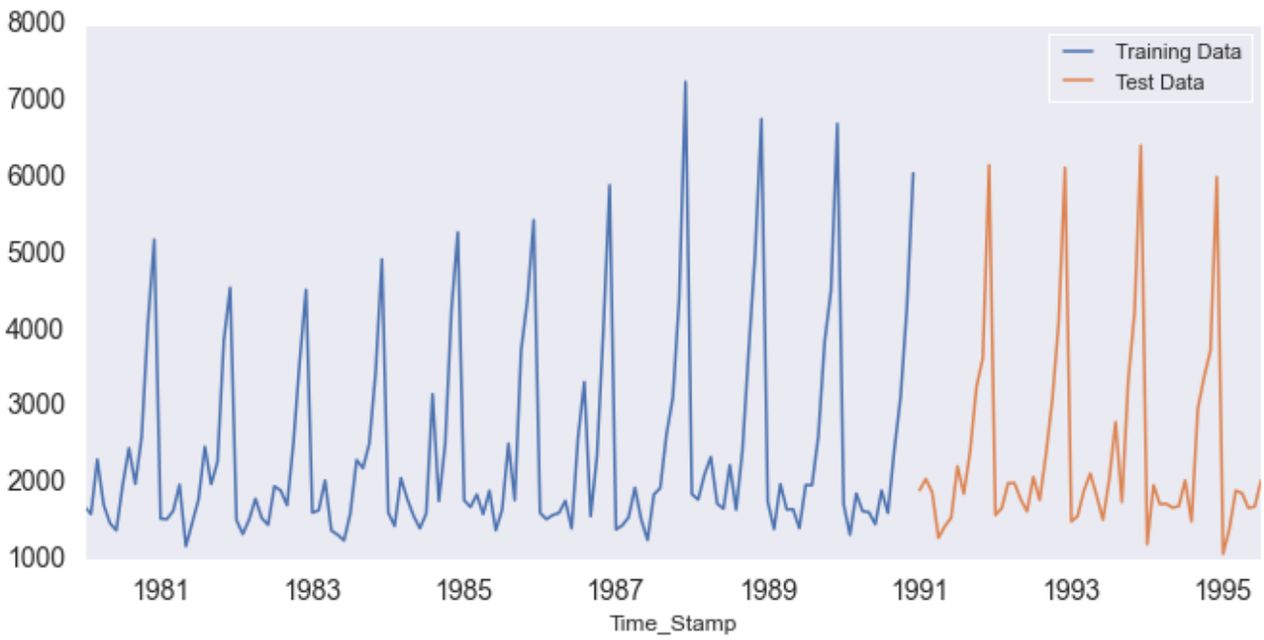

Fig : 9 Plot of Train & Test Data.

Insights

- Blue color represents the train data ,training Data is till the end of 1990.
- Orange color represents the test data ,test Data is from the beginning of 1991 to the last time stamp provided.

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

Model 1: Linear Regression

Note

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

```
Training Time instance
 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 4
4, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,
86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121,
122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132]
Test Time instance
 [133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165,
166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187]
```

Tab:15 Training & Test Time Instances for Linear Regression Model

Observation

We see that we have successfully the generated the numerical time instance order for both the training and test set. Now we will add these values in the training and test set.

First few rows of Training Data

Time_Stamp	Sparkling Wine Sales	time
1980-01-31	1686	1
1980-02-29	1591	2
1980-03-31	2304	3
1980-04-30	1712	4
1980-05-31	1471	5

Last few rows of Training Data

Time_Stamp	Sparkling Wine Sales	time
1990-08-31	1605	128
1990-09-30	2424	129
1990-10-31	3116	130
1990-11-30	4286	131
1990-12-31	6047	132

First few rows of Test Data

Time_Stamp	Sparkling Wine Sales	time
1991-01-31	1902	133
1991-02-28	2049	134
1991-03-31	1874	135
1991-04-30	1279	136
1991-05-31	1432	137

Last few rows of Test Data

Time_Stamp	Sparkling Wine Sales	time
1995-03-31	1897	183
1995-04-30	1862	184
1995-05-31	1670	185
1995-06-30	1688	186
1995-07-31	2031	187

Tab:16 Records of Training & Test Data with Time Instances for Linear Regression Model

Note

Now that our training and test data has been modified, let us go ahead use Linear Regression to build the model on the training data and test the model on the test data.

Building A Linear Regression Model -

Invoke the Linear Regression function (from sklearn.linear_model import LinearRegression) fit the function on the train & test data and build the linear regression model. In this problem we are advised to build linear regression model and check the performance of Predictions on Test sets using RMSE.

Prediction on Test Dataset

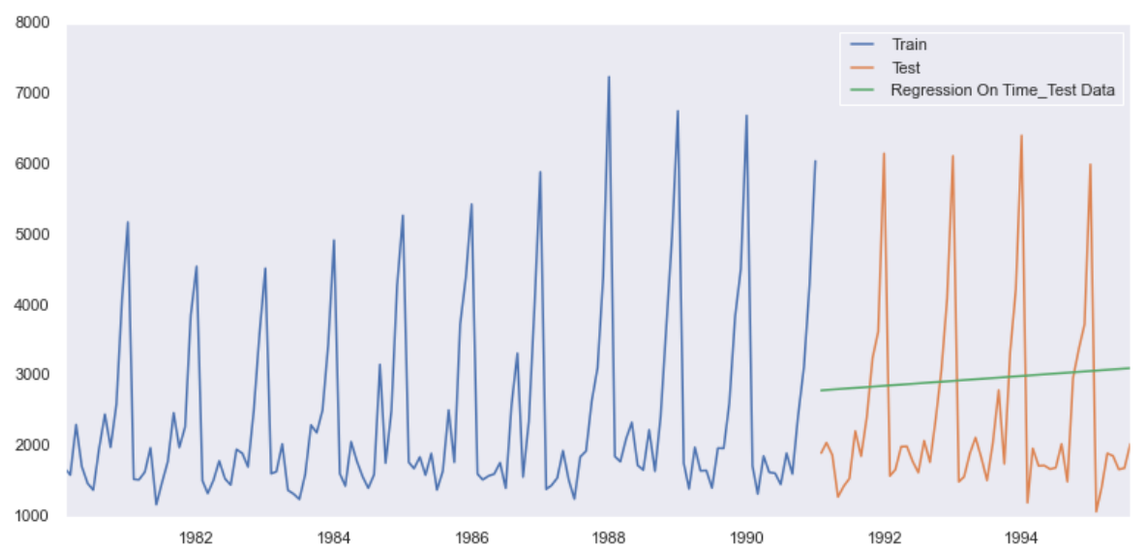

Fig : 10 Prediction on Test Dataset of Linear Regression Model

Observation

Here we get a linear line following the previous patterns.

Model Evaluation Linear Regression

RMSE - The root mean square error (RMSE) for a regression model is similar to the standard deviation (SD) for the ideal measurement model. The SD estimates the deviation from the sample mean x. The RMSE estimates the deviation of the actual y-values from the regression line.

Root mean square error, which is a metric that tells us the average distance between the predicted values from the model and the actual values in the dataset. The lower the RMSE, the better a given model is able to "fit" a dataset.

Test Data - **RMSE** of **Linear Regression** Model is - **1389.135**

Model 2: Naive Model , { Navie Approach: ^yt+1=yt }

Note

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.

Building Navie Model

```
NaiveModel_test['naive'] = np.asarray(train['Sparkling Wine Sales'])
[len(np.asarray(train['Sparkling Wine Sales']))-1]
```

Prediction on Test Dataset

Time_Stamp	Sparkling Wine Sales	naive
1991-01-31	1902	6047
1991-02-28	2049	6047
1991-03-31	1874	6047
1991-04-30	1279	6047
1991-05-31	1432	6047

Tab:17 Prediction on Test Data of Navie Model

Observation

We get the same value of Navie forecast for the entire test data.

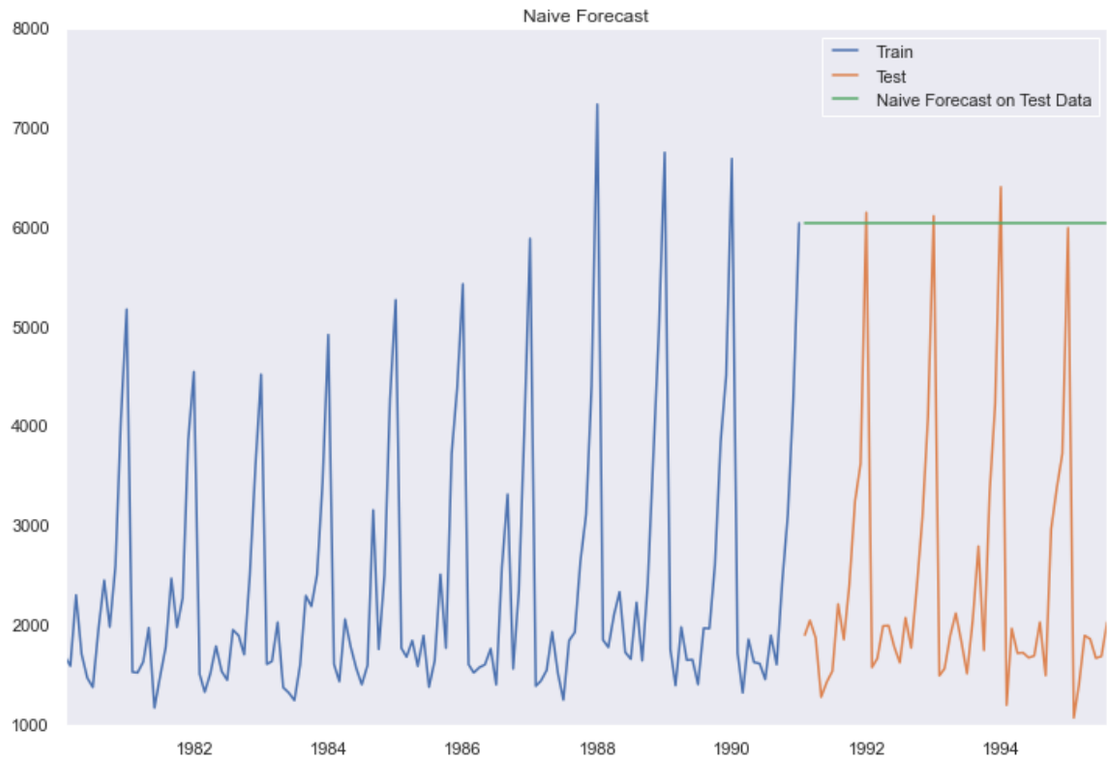

Fig : 11 Prediction on Test Dataset of Navie Forecast Model

Observation

The last observation of the train data is constructed the forecast for entire test data.That's why we are seeing a straight line here. i.e. The Sparkling Wine Sales will be like the recent past.

Model Evaluation Navie Forecast

Test Data - **RMSE** of **Navie Forecast** Model is - **3864.279**

Model 3 : Simple Average Model

Note

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

Building Simple Average Model

SimpleAverage_test['mean_forecast']= train['Sparkling Wine Sales'].mean()

Prediction on Test Dataset

Time_Stamp	Sparkling Wine Sales	mean_forecast
1991-01-31	1902	2403.780303
1991-02-28	2049	2403.780303
1991-03-31	1874	2403.780303
1991-04-30	1279	2403.780303
1991-05-31	1432	2403.780303

Tab:18 Prediction on Test Data of Simple Average Model

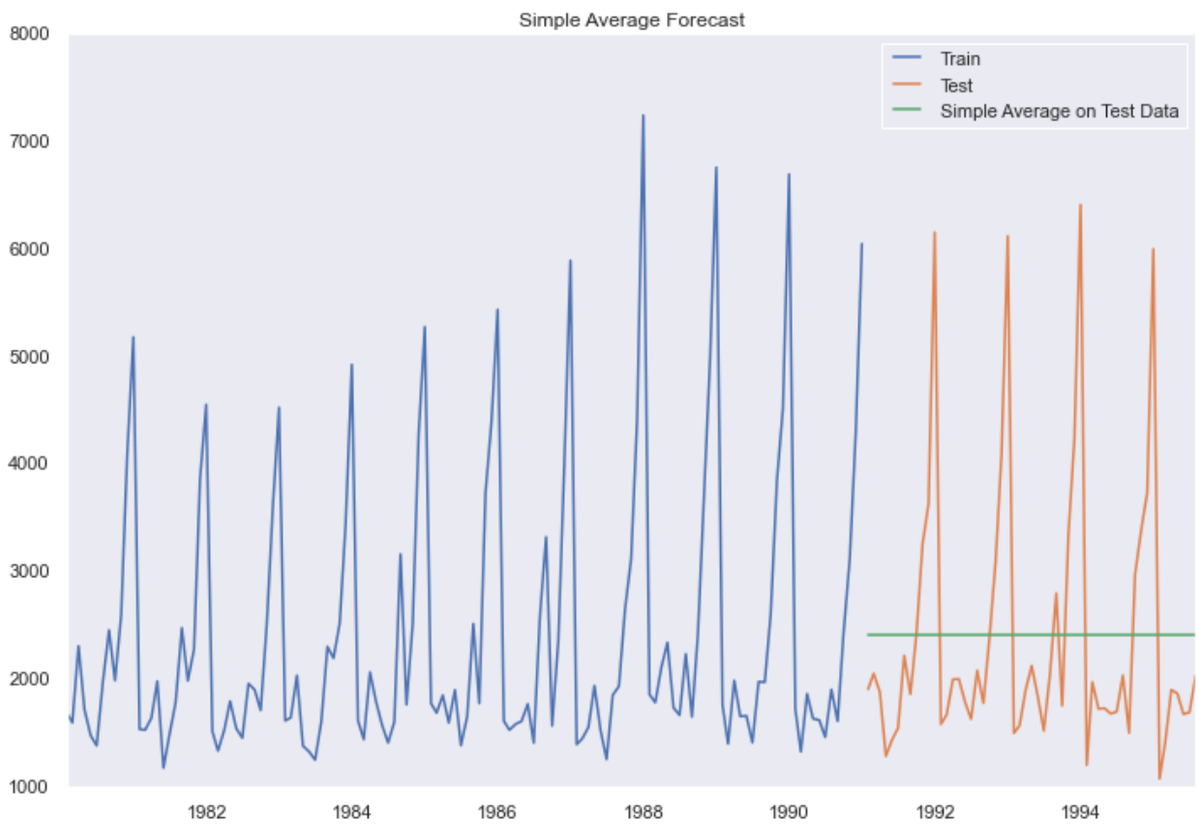

Fig : 12 Prediction on Test Dataset of Simple Average Model

Model Evaluation Simple Average

Test Data - **RMSE** of **Simple Average Model** is - **1275.082**

Model 4 : Moving Average(MA)

Note

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.

Building Moving Average Model

```
MovingAverage['Trailing_2']= MovingAverage['Sparkling Wine Sales'].rolling(2).mean()
MovingAverage['Trailing_4']= MovingAverage['Sparkling Wine Sales'].rolling(4).mean()
MovingAverage['Trailing_6']= MovingAverage['Sparkling Wine Sales'].rolling(6).mean()
MovingAverage['Trailing_9']= MovingAverage['Sparkling Wine Sales'].rolling(9).mean()

MovingAverage.head()
```

Time_Stamp	Sparkling Wine Sales	Trailing_2	Trailing_4	Trailing_6	Trailing_9
1980-01-31	1686	NaN	NaN	NaN	NaN
1980-02-29	1591	1638.5	NaN	NaN	NaN
1980-03-31	2304	1947.5	NaN	NaN	NaN
1980-04-30	1712	2008.0	1823.25	NaN	NaN
1980-05-31	1471	1591.5	1769.50	NaN	NaN

Tab:19 Records of Dataset with Rolling Mean

Plotting of the Whole Data With Moving Average

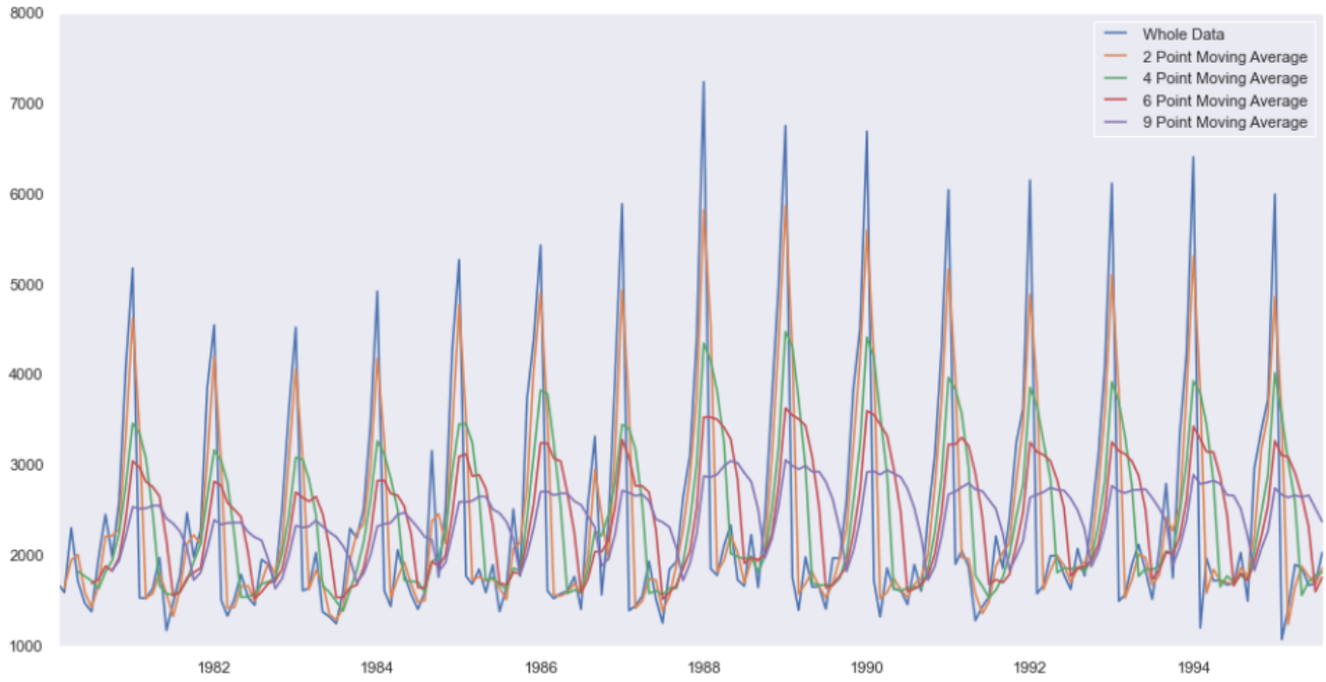

Fig : 13 Whole Data With Moving Average

Observation

2 Point Moving Average curve is copying the original data as it is replicating the original data.

Plotting Moving Average on both the Training and Test data

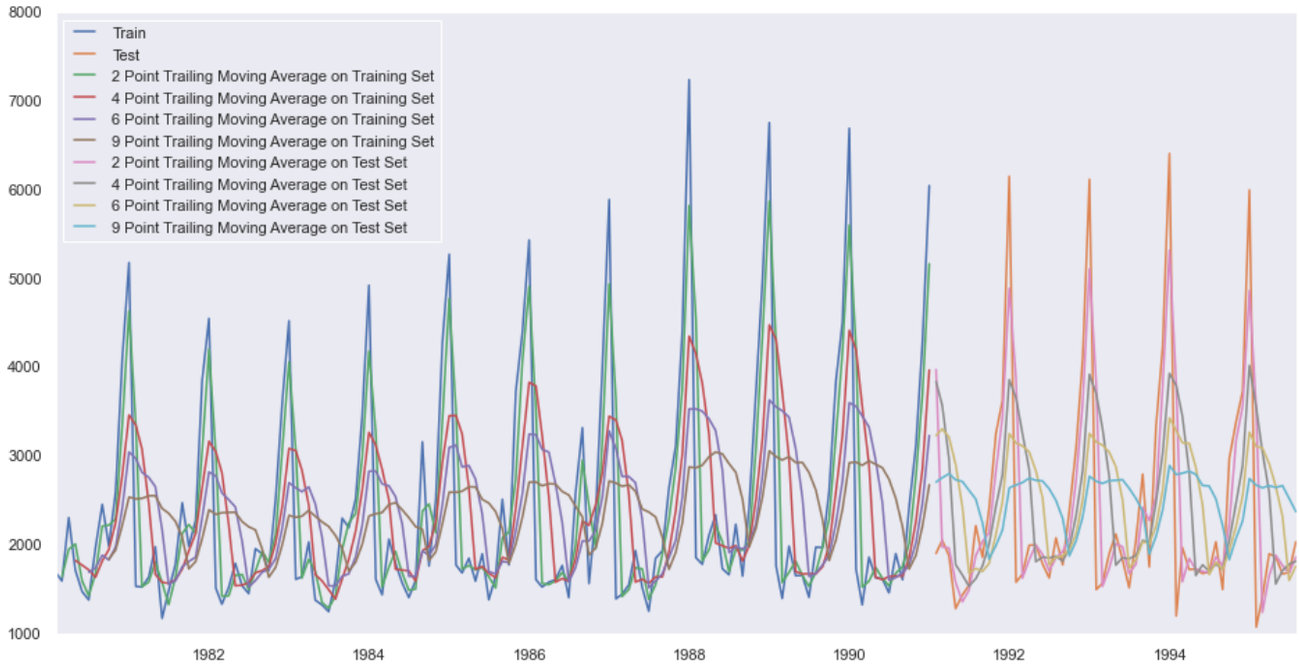

Fig : 14 Moving Average on both the Training and Test data

Test Data - RMSE With Moving Average Model

- For 2 point Moving Average Model forecast on the Test Data, RMSE is 813.401
- For 4 point Moving Average Model forecast on the Test Data, RMSE is 1156.590
- For 6 point Moving Average Model forecast on the Test Data, RMSE is 1283.927
- For 9 point Moving Average Model forecast on the Test Data, RMSE is 1346.278

Result

RMSE of the 2 Point Moving Average Model is the least - 813.401 among the 4 , 6 , 9 Point Moving Average Model.

Model 5 : Simple Exponential Smoothing Model

Single Exponential Smoothing, SES for short, also called Simple Exponential Smoothing, is a time series forecasting method for univariate data without a trend or seasonality. It requires a single parameter, called alpha (a), also called the smoothing factor or smoothing coefficient.

Building Simple Exponential Smoothing Model- Autofit Method

```
model_SES= SimpleExpSmoothing(SES_train['Sparkling Wine Sales'])
model_SES_autofit= model_SES.fit(optimized=True)
```

Checking the Parameter

{'smoothing_level': 0.049606598807459296,
'smoothing_trend': nan,
'smoothing_seasonal': nan,
'damping_trend': nan,
'initial_level': 1818.5047538435304,
'initial_trend': nan,
'initial_seasons': array([], dtype=float64),
'use_boxcox': False,
'lamda': None,
'remove_bias': False}

Insights

Here in the simple exponential smoothing model we get value of alpha = 0.049606598807459296

Prediction on Test Dataset

Time_Stamp	Sparkling Wine Sales	predict
1991-01-31	1902	2724.929339
1991-02-28	2049	2724.929339
1991-03-31	1874	2724.929339
1991-04-30	1279	2724.929339
1991-05-31	1432	2724.929339

Tab:20 Prediction on Test Data of Simple Exponential Model

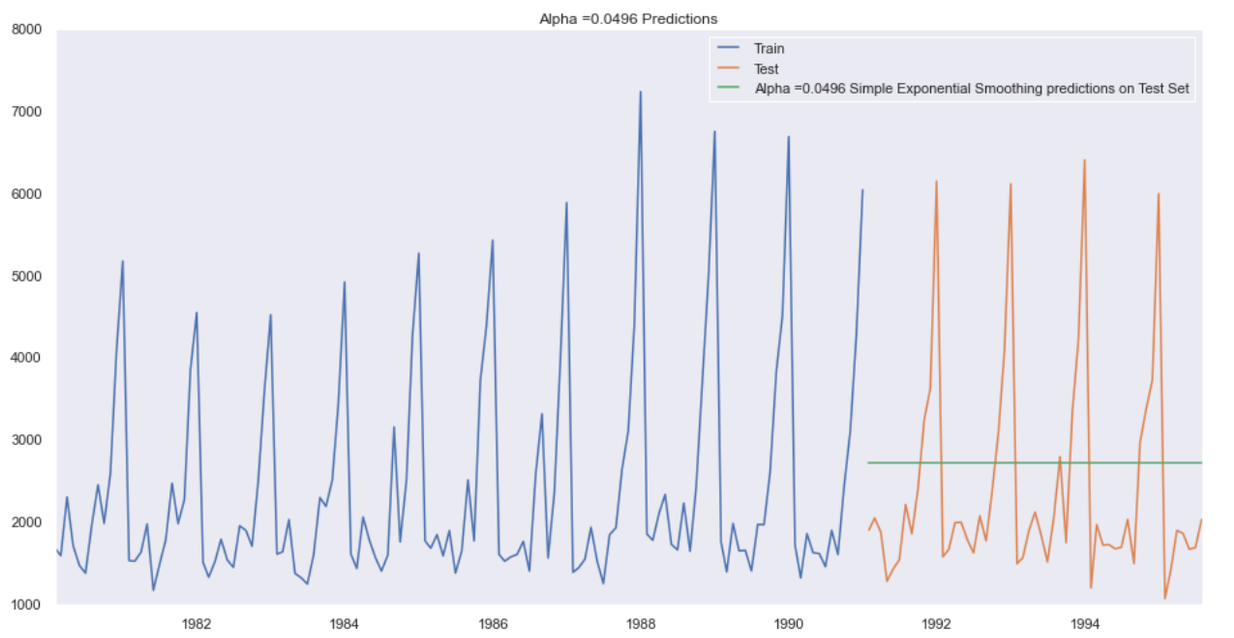

Fig : 15 Prediction on Test Dataset of Simple Exponential Model

Test Data - RMSE with Simple Exponential Smoothing Model

For Alpha =0.0496 Simple Exponential Smoothing Model forecast on the Test Data, RMSE is 1316.035

Simple Exponential Smoothing Model by Setting different alpha values - Brute Force Meshod

Remember, 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.

	Alpha Values	Train RMSE	Test RMSE
0	0.3	1359.511747	1935.507132
1	0.4	1352.588879	2311.919615
2	0.5	1344.004369	2666.351413
3	0.6	1338.805381	2979.204388
4	0.7	1338.844308	3249.944092
5	0.8	1344.462091	3483.801006
6	0.9	1355.723518	3686.794285

Tab:21 Alpha Values for Train & Test RMSE Simple Exponential Smoothing Model

Insights

Here after comparing different alpha values we get the least RMSE on Test data at alpha=0.3.So, we can take alpha=0.3 for the model.

We will take alpha = 0.3 to built our simple exponential smoothing model .

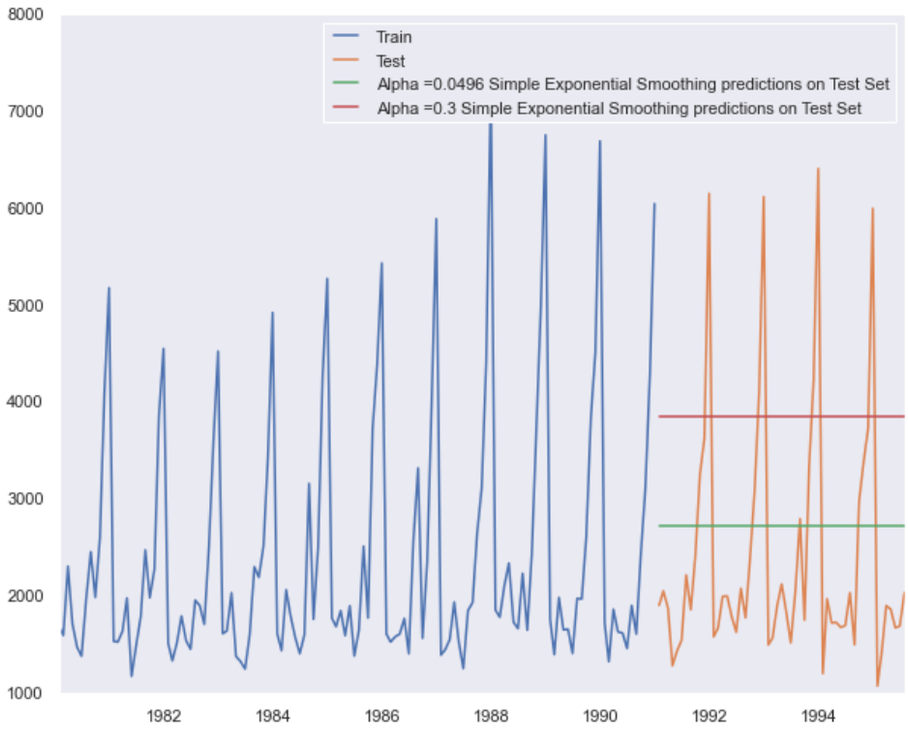

Fig : 16 Prediction on Test Dataset of Simple Exponential Model at alpha = 0.3

Test Data - RMSE with Simple Exponential Smoothing Model with aplha =0.3

For Alpha =0.3 Simple Exponential Smoothing Model forecast on the Test Data, RMSE is 1935.507132

Model 6 : Double Exponential Smoothing (Holt's Model)

Double Exponential Smoothing model is suitable to model the time series with trend but without seasonality. In the model there are two kinds of smoothed quantities: smoothed signal and smoothed trend. ... The Holt's linear exponential smoothing displays a constant trend indefinitely into the future.

Double exponential smoothing employs a level component and a trend component at each period. Double exponential smoothing uses two weights, (also called smoothing parameters), to update the components at each period.

Note

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

Building Double Exponential Smoothing Model (Holt's Model) by Setting different alpha & beta values -Brute Force Method (For Codes please check code file.)

We will run a loop with different alpha and beta values to understand which particular value works best for alpha and beta on the test set.

	Alpha Values	Beta Values	Train RMSE	Test RMSE
0	0.3	0.3	1592.292788	18259.110704
8	0.4	0.3	1569.338606	23878.496940
1	0.3	0.4	1682.573828	26069.841401
16	0.5	0.3	1530.575845	27095.532414
24	0.6	0.3	1506.449870	29070.722592

Tab:22 Alpha & Beta Values for Train & Test RMSE Double Exponential Smoothing Model

Insights

Here after comparing different alpha and beta values we get the least RMSE on Test data at alpha=0.3. and beta =0.3 So, we can take alpha=0.3 and beta = 0.3 for the model.

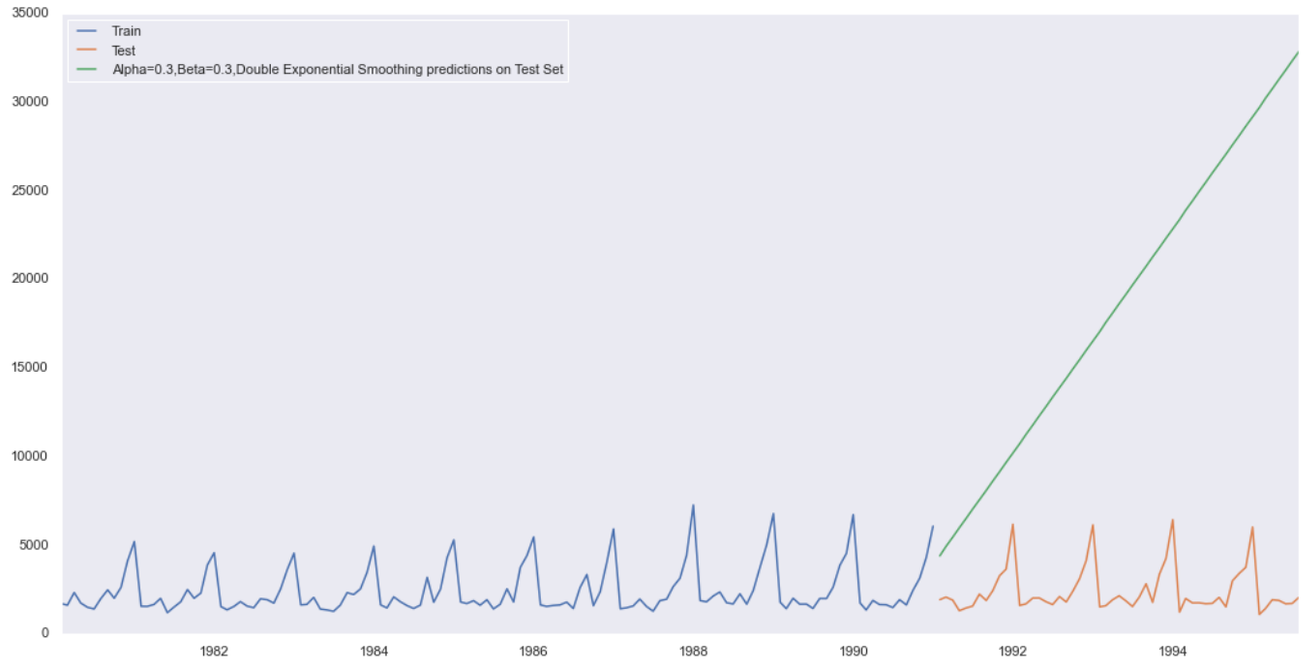

Fig : 17 Prediction on Test Dataset of Double Exponential Model at alpha = 0.3 and beta = 0.3

Test Data - RMSE with Double Exponential Smoothing Model with aplha =0.3 and beta = 0.3

For Alpha =0.3 and Beta = 0.3 Double Exponential Smoothing Model forecast on the Test Data, RMSE is 18259.110704

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

Triple exponential smoothing is used to handle the time series data containing a seasonal component. This method is based on three smoothing equations: stationary component, trend, and seasonal. Both seasonal and trend can be additive or multiplicative. ... Seasonal change smoothing factor.

Note

Three parameters α,β and γ are estimated in this model.Level, Trend and Seasonality are accounted for in this model.

Building Triple Exponential Smoothing Model (Holt's Winter Model) - Autofit Method

```
model_TES = ExponentialSmoothing(TES_train['Sparkling Wine Sales'],trend='additive',seasonal='multiplicative',freq='M')

model_TES_autofit = model_TES.fit()
```

Checking the Parameter

{'smoothing_level': 0.11107193639676448,
'smoothing_trend': 0.06170661554551025,
'smoothing_seasonal': 0.39507938025204126,
'damping_trend': nan,
'initial_level': 1640.0001849429073,
'initial_trend': -15.111380527844805,
'initial_seasons': array([1.03314765, 0.98921937, 1.40520416, 1.20124958, 0.93920975,
0.95169819, 1.29579418, 1.68037583, 1.35792845, 1.79419758,
2.82688557, 3.60017043]),
'use_boxcox': False,
'lamda': None,
'remove_bias': False}

Insights

Here in the Triple Exponential Smoothing model we get value of alpha = 0.11107193639676448 , beta = 0.06170661554551025 , gamma = 0.39507938025204126

Prediction on Test Dataset

Time_Stamp	Sparkling Wine Sales	auto_predict
1991-01-31	1902	1577.246443
1991-02-28	2049	1333.624267
1991-03-31	1874	1746.041391
1991-04-30	1279	1630.569083
1991-05-31	1432	1523.309244

Tab:23 Prediction on Test Data of Triple Exponential Model

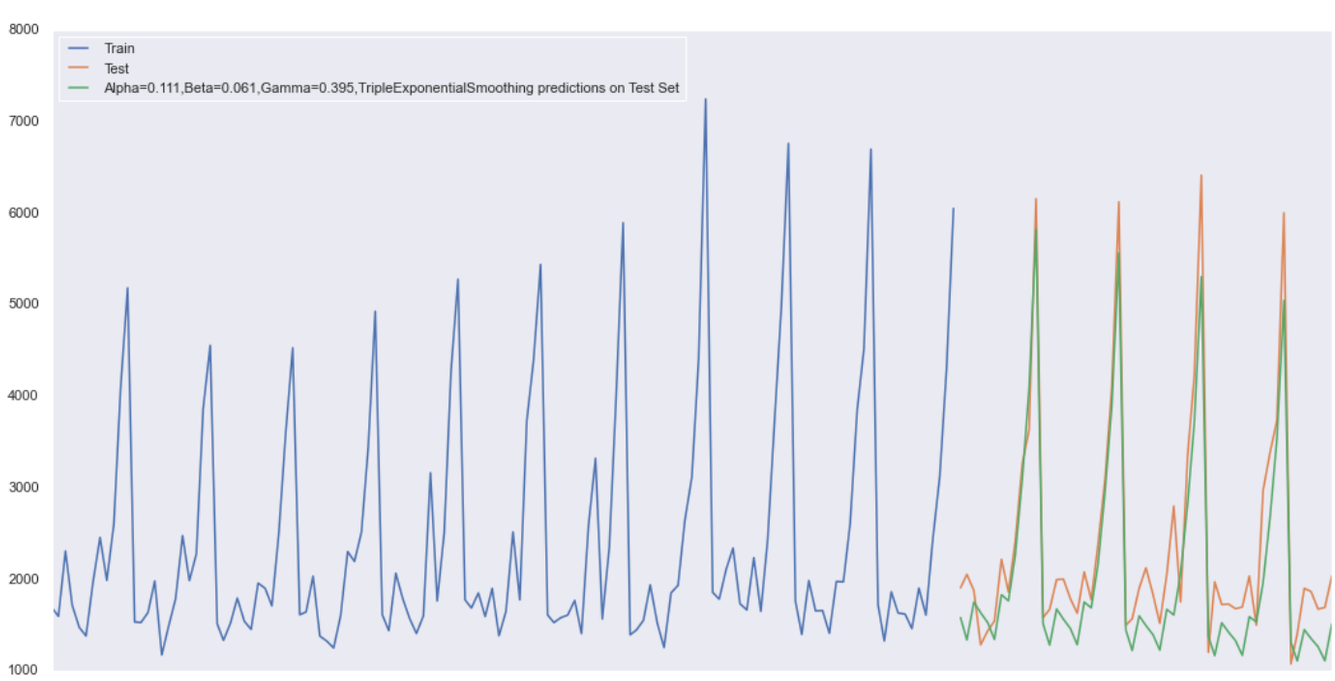

Fig : 18 Prediction on Test Dataset of Triple Exponential Model at alpha =0.111 beta = 0.0617 and gamma = 0.395

Test Data - RMSE with Triple Exponential Smoothing Model with alpha =0.111 beta = 0.0617 and gamma = 0.395 .

For alpha =0.111 beta = 0.0617 and gamma = 0.395 Triple Exponential Smoothing Model forecast on the Test Data, RMSE is 469.591666.

Triple Exponential Smoothing Model by Setting different alpha , beta and gamma values - Brute Force Meshod

We will run a loop with different alpha , beta and gamma values to understand which particular value works best for alpha, beta and gamma on the test set.

	Alpha Values	Beta Values	Gamma Values	Train RMSE	Test RMSE
0	0.3	0.3	0.3	404.513320	392.786198
8	0.3	0.4	0.3	424.828055	410.854547
65	0.4	0.3	0.4	435.553595	421.409170
296	0.7	0.8	0.3	700.317756	518.188752
130	0.5	0.3	0.5	498.239915	542.175497

Tab:24 Alpha ,Beta and Gamma Values for Train & Test RMSE Triple Exponential Smoothing Model

Insights

Here after comparing different alpha , beta and gamma values we get the least RMSE on Test data at alpha=0.3 , beta = 0.3 and gamma = 0.3. So, we can take alpha=0.3, beta = 0.3 and gamma = 0.3 for the model.

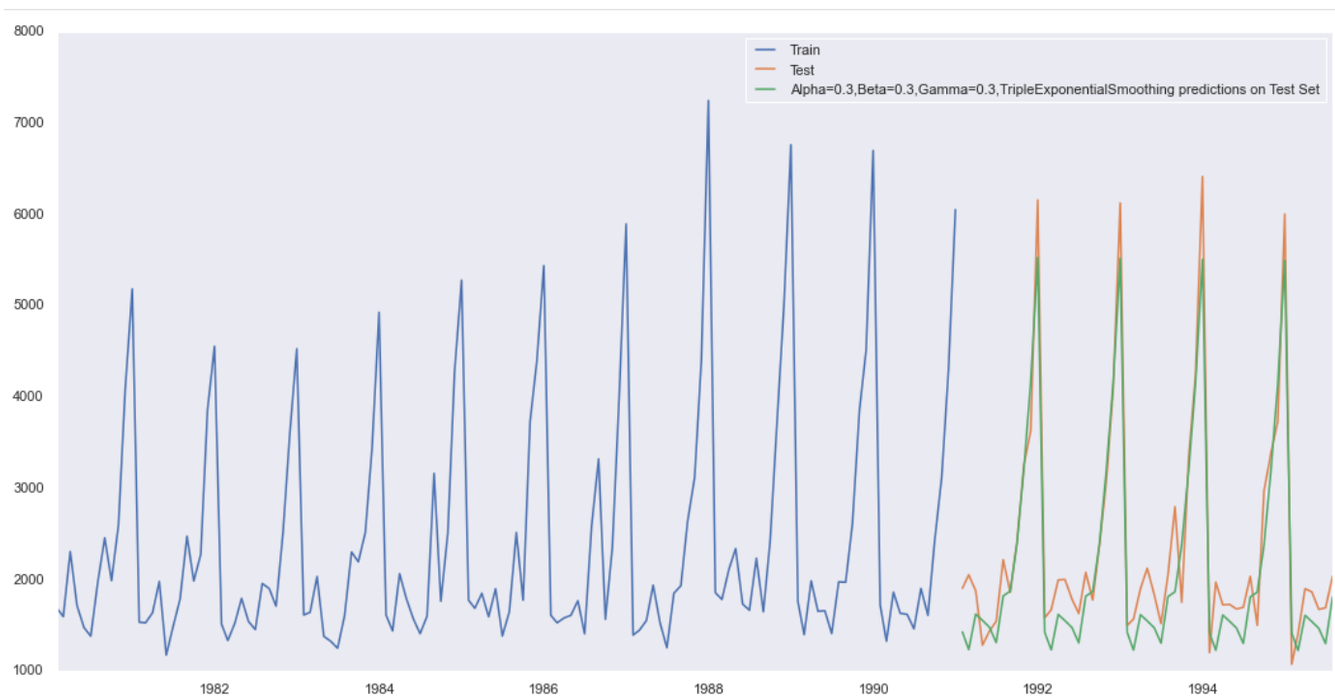

Fig : 19 Prediction on Test Dataset of Triple Exponential Model at alpha =0.3 beta = 0.3 and gamma = 0.3

Test Data - RMSE with Triple Exponential Smoothing Model with alpha =0.3 beta = 0.3 and gamma = 0.3 .

For alpha =0.3 beta = 0.3 and gamma = 0.3 . Triple Exponential Smoothing Model forecast on the Test Data, RMSE is 392.786198.

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. Note: Stationarity should be checked at alpha = 0.05.

The **Augmented Dickey-Fuller test** is an 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:

H0
: The Time Series has a unit root and is thus non-stationary.
H1
: 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 α
value.

Check for stationarity of the whole Time Series data. - Dickey-Fuller test

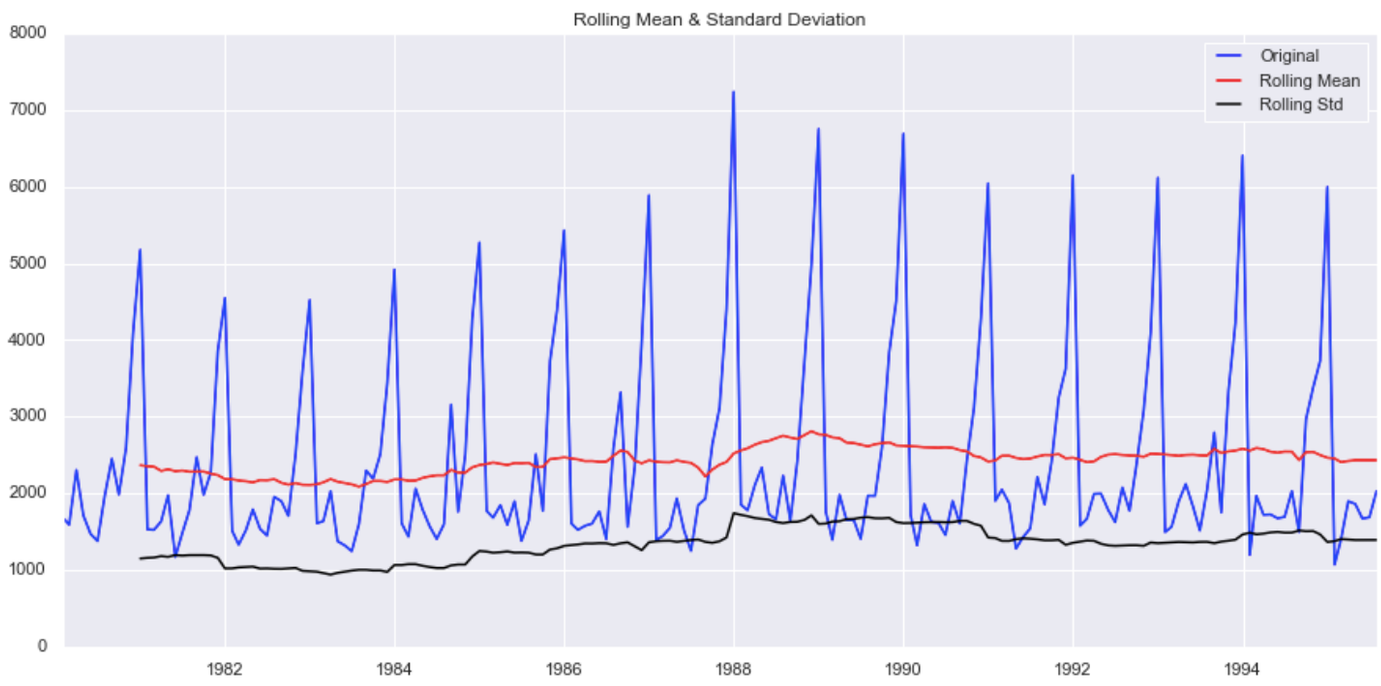

Fig : 20 Check for stationarity of the whole Time Series data. - Dickey-Fuller test

```
Results of Dickey-Fuller Test:
Test Statistic                  -1.360497
p-value                          0.601061
#Lags Used                      11.000000
Number of Observations Used    175.000000
Critical Value (1%)             -3.468280
Critical Value (5%)             -2.878202
Critical Value (10%)            -2.575653
dtype: float64
```

Tab:25 Dickey - Fuller Test Result on WholeTS Data

Conclusion :

On comparing the p-value , we found p-value is greater than the 5% significant level ,hence we fail to reject the null hypothesis & reached on the conclusion that he Time Series is non-stationary. Let us take a difference of order 1 and check whether the Time Series is stationary or not.

Let us take a difference of order 1 and check whether the Time Series is stationary or not.

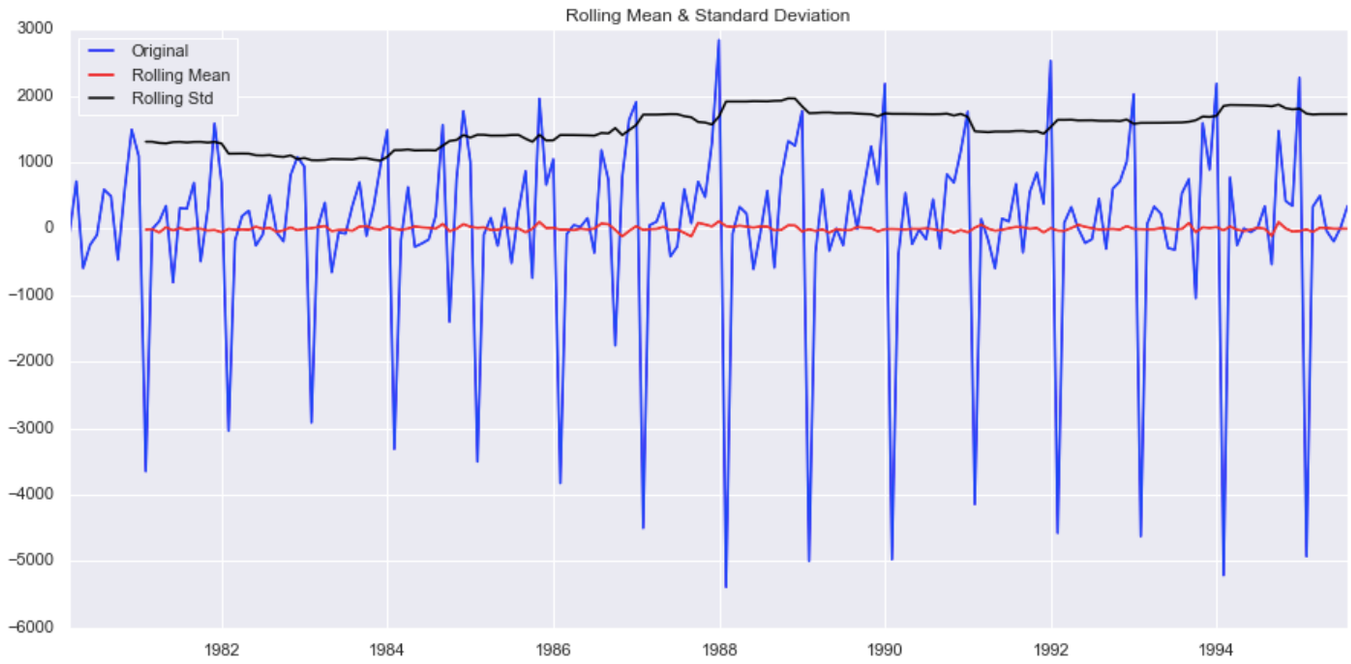

Fig : 21 Check for stationarity after differencing of order 1 on whole Time Series data. - Dickey-Fuller test

```
Results of Dickey-Fuller Test:
Test Statistic                -45.050301
p-value                         0.000000
#Lags Used                     10.000000
Number of Observations Used   175.000000
Critical Value (1%)            -3.468280
Critical Value (5%)            -2.878202
Critical Value (10%)           -2.575653
dtype: float64
```

Tab:26 Dickey - Fuller Test Result on WholeTS Data with differencing of order 1

Conclusion :

On comparing the p-value , we found p-value is less than the 5% significant level ,hence we reject the null hypothesis & reached on the conclusion that he Time Series is stationary with difference of order 1.We see that at $\alpha = 0.05$ the Time Series is indeed stationary.

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.

Automated version of an ARIMA model with the lowest Akaike Information Criteria (AIC).

An ARIMA model is a class of statistical models for analyzing and forecasting time series data. ... ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average. It is a generalization of the simpler AutoRegressive Moving Average and adds the notion of integration

Checking the Stationarity of the Train Data

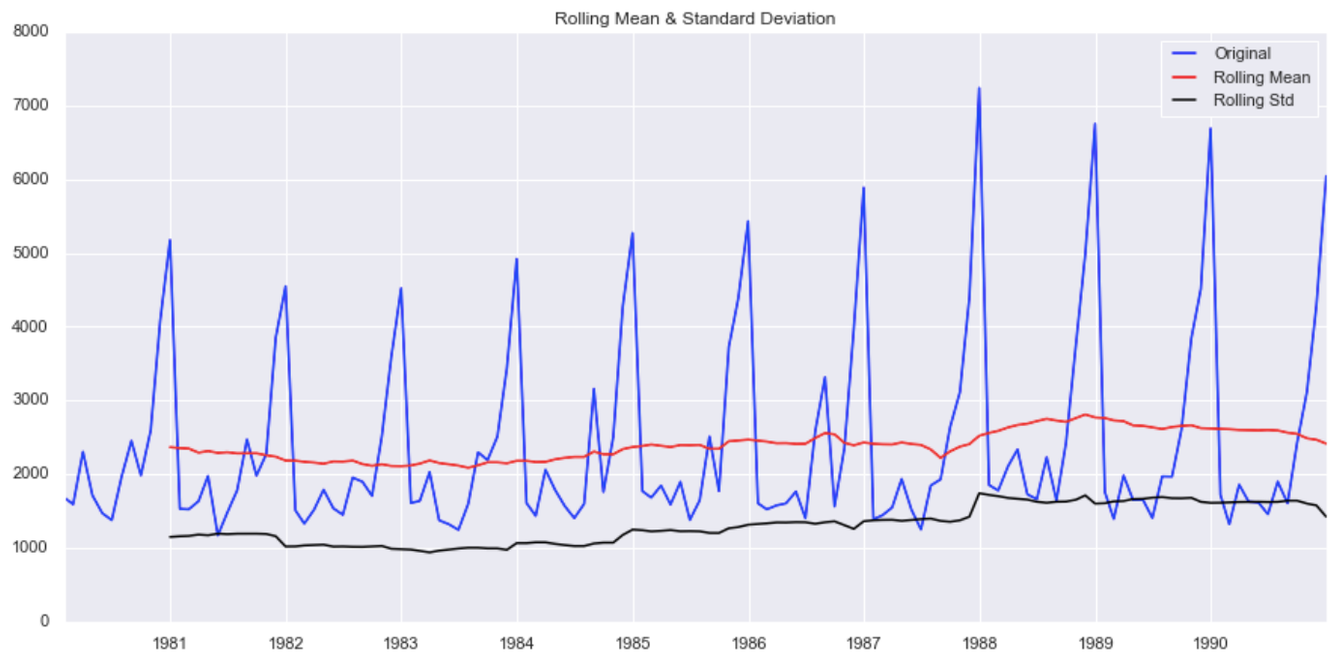

Fig : 22 Check for stationarity of the Train Time Series data. - Dickey-Fuller test

```
Results of Dickey-Fuller Test:
Test Statistic                  -1.208926
p-value                          0.669744
#Lags Used                      12.000000
Number of Observations Used    119.000000
Critical Value (1%)             -3.486535
Critical Value (5%)             -2.886151
Critical Value (10%)            -2.579896
dtype: float64
```

Tab:27 Dickey - Fuller Test Result on Train TS Data

Conclusion :

On comparing the p-value , we found p-value is greater than the 5% significant level ,hence we fail to reject the null hypothesis & reached on the conclusion that he Time Series is non-stationary.
Let us take a difference of order 1 and check whether the Time Series is stationary or not.

Let us take a difference of order 1 and check whether the Train Time Series is stationary or not.

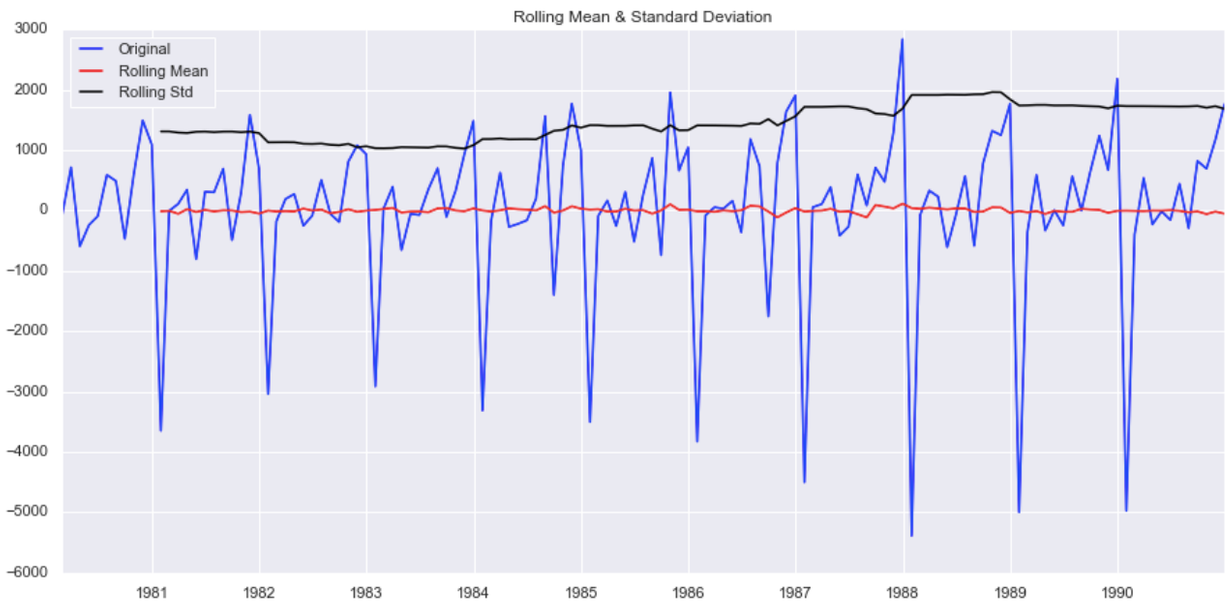

Fig : 23 Check for stationarity after differencing of order 1 on Train Time Series data. - Dickey-Fuller test

```
Results of Dickey-Fuller Test:
Test Statistic                -8.005007e+00
p-value                        2.280104e-12
#Lags Used                     1.100000e+01
Number of Observations Used    1.190000e+02
Critical Value (1%)           -3.486535e+00
Critical Value (5%)           -2.886151e+00
Critical Value (10%)          -2.579896e+00
dtype: float64
```

Tab:28 Dickey - Fuller Test Result on Train TS Data with differencing of order 1

Conclusion :

On comparing the p-value , we found p-value is less than the 5% significant level ,hence we reject the null hypothesis & reached on the conclusion that he Time Series is stationary with difference of order 1.We see that at α = 0.05 the Time Series is indeed stationary.

Note

If the series is non-stationary, stationarize 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.Here we see that differencing of order 1 makes the series stationary so value of d in ARIMA model will kept as 1 as we need to take a difference of the series to make it stationary.we are running loop which helps us in getting a combination of different parameters of p and q in the range of 0 and 2 for our ARIMA model. Let's check the combinations and will select that combination which have lowest AIC value for ARIMA model.

	param	AIC
8	(2, 1, 2)	2210.621575
7	(2, 1, 1)	2232.360490
2	(0, 1, 2)	2232.783098
5	(1, 1, 2)	2233.597647
4	(1, 1, 1)	2235.013945
6	(2, 1, 0)	2262.035601
1	(0, 1, 1)	2264.906437
3	(1, 1, 0)	2268.528061
0	(0, 1, 0)	2269.582796

Tab:29 Combinations & AIC Values for Auto_ARIMA Model

Conclusion :

From the Combination and AIC value table we get best combination for ARIMA model which have least AIC , combination of different parameters of p and q are (2,1,2) here , p=2 ,d=1, and q=2 , this combination has lowest AIC of 2210.621575 among all the combinations. So , we can take (2,1,2) combination to built our ARIMA model.

Building Automated version of an ARIMA model with the lowest Akaike Information Criteria (AIC).

```
auto_ARIMA = ARIMA(train['Sparkling Wine Sales'], order=(2,1,2),freq='M')

results_auto_ARIMA = auto_ARIMA.fit()
```

Auto_ARIMA Model Results

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:     D.Sparkling Wine Sales   No. Observations:                  131
Model:                 ARIMA(2, 1, 2)   Log Likelihood               -1099.311
Method:                       css-mle   S.D. of innovations            1013.266
Date:                Sun, 19 Dec 2021   AIC                            2210.622
Time:                        00:04:17   BIC                            2227.873
Sample:                    02-29-1980   HQIC                           2217.632
                         - 12-31-1990
==================================================================================================
                                     coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------------
const                              5.5850      0.518     10.774      0.000       4.569       6.601
ar.L1.D.Sparkling Wine Sales       1.2701      0.075     17.046      0.000       1.124       1.416
ar.L2.D.Sparkling Wine Sales      -0.5601      0.074     -7.616      0.000      -0.704      -0.416
ma.L1.D.Sparkling Wine Sales      -1.9967      0.042    -47.076      0.000      -2.080      -1.914
ma.L2.D.Sparkling Wine Sales       0.9967      0.042     23.455      0.000       0.913       1.080
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.1339           -0.7070j            1.3362           -0.0887
AR.2            1.1339           +0.7070j            1.3362            0.0887
MA.1            1.0002           +0.0000j            1.0002            0.0000
MA.2            1.0031           +0.0000j            1.0031            0.0000
-----------------------------------------------------------------------------
```

Tab:30 Result Summary of Auto _ARIMA (2,1,2)

Conclusion

By looking the ARIMA model summary we found that the p-values of all the components of the ARIMA model are significant.Hence the value which we get for p,d & q from lowest AIC are significant.

Test Data - RMSE of Auto_ARIMA(2,1,2) Model with p =2 d = 1 and q=2 is 1374.336485

Automated version of the SARIMA model with the lowest Akaike Information Criteria (AIC).

A seasonal autoregressive integrated moving average (SARIMA) model is one step different from an ARIMA model based on the concept of seasonal trends.SARIMA accepts an additional set of parameters $(P,D,Q)m$ that specifically describe the seasonal components of the model. Here P, D and Q represent the seasonal regression, differencing and moving average coefficients, and m represents the number of data points (rows) in each seasonal cycle.Seasonal Autoregressive Integrated Moving Average, SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component.

Plotting ACF plot to understand the seasonal parameter for the SARIMA model.

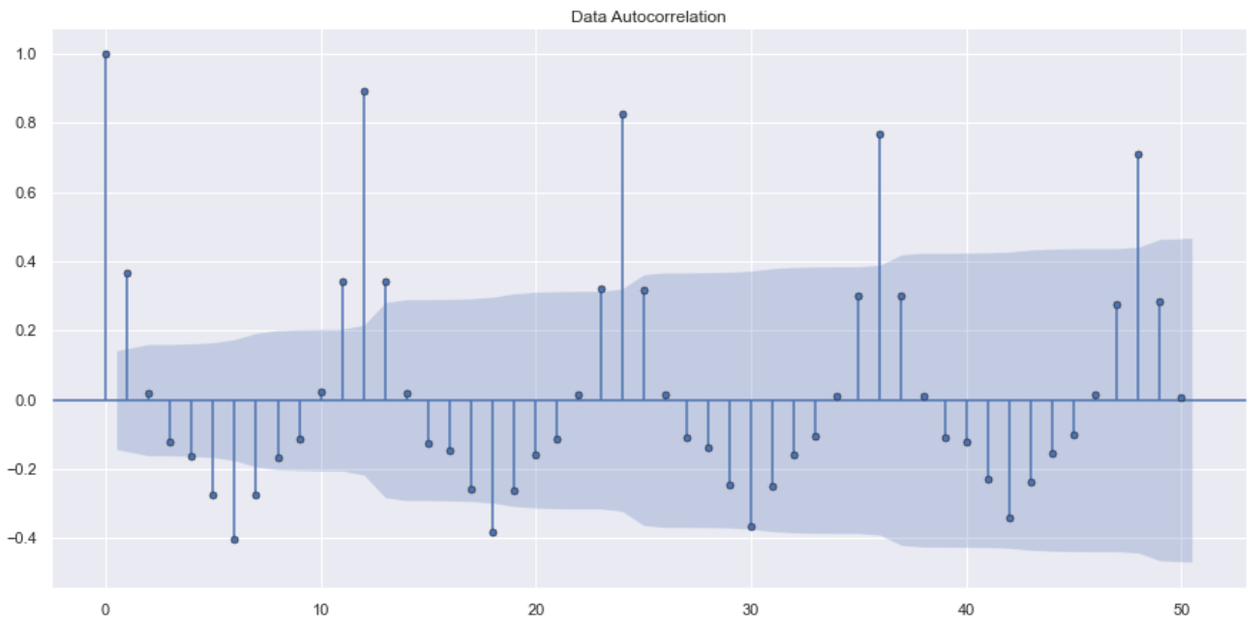

Fig : 24 ACF plot of the original Data

Conclusion

By looking the ACF plot we found a pattern that every 6th and 12th lag is significant and repeating itself in the same pattern so building a SARIMA model we take seasonality of 6 as well as 12 and built the model.We see that there can be a seasonality of 6 and 12. We will run our auto SARIMA and manual SARIMA models by setting seasonality both as 6 and 12.

Checking for the combination of different parameters of (p d q) and (P,D,Q) by setting the seasonality as 6 for the auto SARIMA model.

	param	seasonal	AIC
53	(1, 1, 2)	(2, 0, 2, 6)	1727.678699
26	(0, 1, 2)	(2, 0, 2, 6)	1727.888803
80	(2, 1, 2)	(2, 0, 2, 6)	1729.363547
17	(0, 1, 1)	(2, 0, 2, 6)	1741.696451
44	(1, 1, 1)	(2, 0, 2, 6)	1743.374729

Tab:31 Combinations & AIC Values for Auto_SRIMA Model With Seasonality as 6

Conclusion :

From the Combination and AIC value table we get best combination for SARIMA model which have least AIC , combination of (p d q) and (P,D,Q) by setting the seasonality as 6 for the auto SARIMA model are (1, 1, 2) (2, 0, 2, 6) which have least AIC of 1727.678699. So , we can take (1, 1, 2) (2, 0, 2, 6) combination to built our Auto_SARIMA model.

Building Automated version of SARIMA model with the lowest Akaike Information Criteria (AIC).

```
auto_SARIMA_6 = sm.tsa.statespace.SARIMAX(train['Sparkling Wine Sales'].values,
                                          order=(1, 1, 2),
                                          seasonal_order=(2, 0, 2, 6),
                                          enforce_stationarity=False,
                                          enforce_invertibility=False)
results_auto_SARIMA_6 = auto_SARIMA_6.fit(maxiter=1000)
```

Auto_SARIMA Model Results

SARIMAX Results

Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(1, 1, 2)x(2, 0, 2, 6)	Log Likelihood	-855.839
Date:	Sun, 19 Dec 2021	AIC	1727.679
Time:	00:04:52	BIC	1749.707
Sample:	0	HQIC	1736.621
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.6449	0.286	-2.257	0.024	-1.205	-0.085
ma.L1	-0.1068	0.250	-0.428	0.669	-0.596	0.383
ma.L2	-0.7006	0.202	-3.471	0.001	-1.096	-0.305
ar.S.L6	-0.0045	0.027	-0.165	0.869	-0.057	0.049
ar.S.L12	1.0361	0.018	56.073	0.000	1.000	1.072
ma.S.L6	0.0676	0.152	0.444	0.657	-0.231	0.366
ma.S.L12	-0.6122	0.093	-6.588	0.000	-0.794	-0.430
sigma2	1.448e+05	1.71e+04	8.465	0.000	1.11e+05	1.78e+05

Ljung-Box (L1) (Q):	0.09	Jarque-Bera (JB):	25.23
Prob(Q):	0.77	Prob(JB):	0.00
Heteroskedasticity (H):	2.63	Skew:	0.47
Prob(H) (two-sided):	0.00	Kurtosis:	5.09

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

Tab:32 Result Summary of Auto _SARIMA(1, 1, 2)(2, 0, 2, 6)

Conclusion

By looking the SARIMA model summary we found that the coff. for all the components and p-values of the components like - ma.L1 , ar.S.L6 and ma.S.L6 of the SARIMA model are more than 0.05 so these are insignificant values.

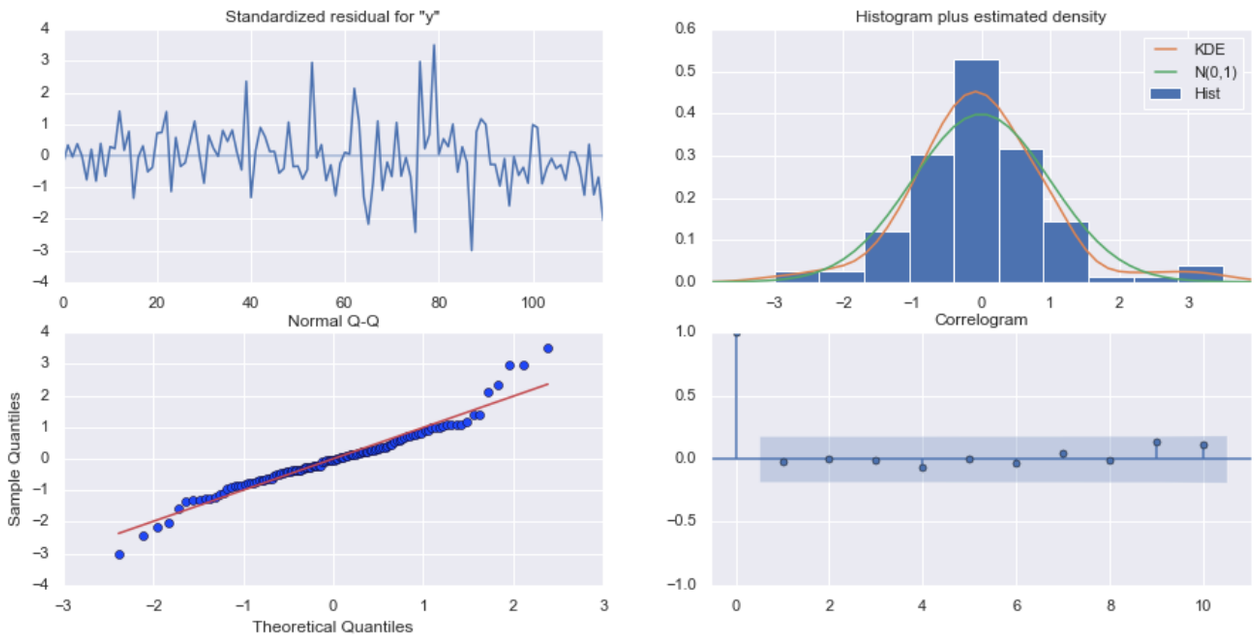

Fig : 25 Diagnostics Plot of Auto_SARIMA (1, 1, 2)(2, 0, 2, 6)

Insights

- Form Standardize residual for "Y" we infer that is no pattern found in the error part.
- By looking at the diagnostics plots like (Histogram and Normal Q-Q) we infer that errors are almost normally distributed.
- From correlogram we infer that correlation coff. of error terms is zero and we can see that none of the errors are significant.(i.e No error term crossing the cut-off).

Summary Frame with alpha =0.05

y	mean	mean_se	mean_ci_lower	mean_ci_upper
0	1330.386482	380.571295	584.480451	2076.292513
1	1177.274343	392.122746	408.727883	1945.820802
2	1625.927381	392.316725	857.000729	2394.854033
3	1546.300445	397.717185	766.789087	2325.811804
4	1308.731674	398.936056	526.831371	2090.631977

Tab:33 SummaryFrame of Auto _SARIMA(1, 1, 2)(2, 0, 2, 6) at alpha = 0.05

Observation:

From the above table at 95 % of confidence interval we have the mean value , mean_std , and lower limit & upper limit for forecast of Auto _SARIMA(1, 1, 2)(2, 0, 2, 6) at alpha = 0.05

Test Data - RMSE of Auto_SARIMA(1, 1, 2)(2, 0, 2, 6) Model is 626.9586777963591

Checking for the combination of different parameters of (p d q) and (P,D,Q) by setting the seasonality as 12 for the auto SARIMA model.

	param	seasonal	AIC
50	(1, 1, 2)	(1, 0, 2, 12)	1555.584247
53	(1, 1, 2)	(2, 0, 2, 12)	1555.934563
26	(0, 1, 2)	(2, 0, 2, 12)	1557.121563
23	(0, 1, 2)	(1, 0, 2, 12)	1557.160507
77	(2, 1, 2)	(1, 0, 2, 12)	1557.340402

Tab:34 Combinations & AIC Values for Auto_SRIMA Model With Seasonality as 12

Conclusion :

From the Combination and AIC value table we get best combination for SARIMA model which have least AIC , combination of (p d q) and (P,D,Q) by setting the seasonality as 12 for the auto SARIMA model are (1, 1, 2) (1, 0, 2, 12) which have least AIC of 1555.584247. So , we can take (1, 1, 2) (1, 0, 2, 12) combination to built our Auto_SARIMA model.

Building Automated version of SARIMA model with the lowest Akaike Information Criteria (AIC).

```
auto_SARIMA_12 = sm.tsa.statespace.SARIMAX(train['Sparkling Wine Sales'].values,
                                           order=(1, 1, 2),
                                           seasonal_order=(1, 0, 2, 12),
                                           enforce_stationarity=False,
                                           enforce_invertibility=False)
results_auto_SARIMA_12 = auto_SARIMA_12.fit(maxiter=1000)
```

Auto_SARIMA Model Results

SARIMAX Results

Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(1, 1, 2)x(1, 0, 2, 12)	Log Likelihood	-770.792
Date:	Sun, 19 Dec 2021	AIC	1555.584
Time:	00:05:38	BIC	1574.095
Sample:	0	HQIC	1563.083
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.6281	0.255	-2.463	0.014	-1.128	-0.128
ma.L1	-0.1041	0.225	-0.463	0.643	-0.545	0.337
ma.L2	-0.7276	0.154	-4.734	0.000	-1.029	-0.426
ar.S.L12	1.0439	0.014	72.843	0.000	1.016	1.072
ma.S.L12	-0.5550	0.098	-5.663	0.000	-0.747	-0.363
ma.S.L24	-0.1355	0.120	-1.134	0.257	-0.370	0.099
sigma2	1.506e+05	2.03e+04	7.400	0.000	1.11e+05	1.9e+05

Ljung-Box (L1) (Q):	0.04	Jarque-Bera (JB):	11.72
Prob(Q):	0.84	Prob(JB):	0.00
Heteroskedasticity (H):	1.47	Skew:	0.36
Prob(H) (two-sided):	0.26	Kurtosis:	4.48

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

Tab:35 Result Summary of Auto _SARIMA(1, 1, 2)(1, 0, 2, 12)

Conclusion

By looking the SARIMA model summary we found that the coff. for all the components and p-values of the components like - ma.L1 and ma.S.L24 of the SARIMA model are more than 0.05 so these are insignificant values.

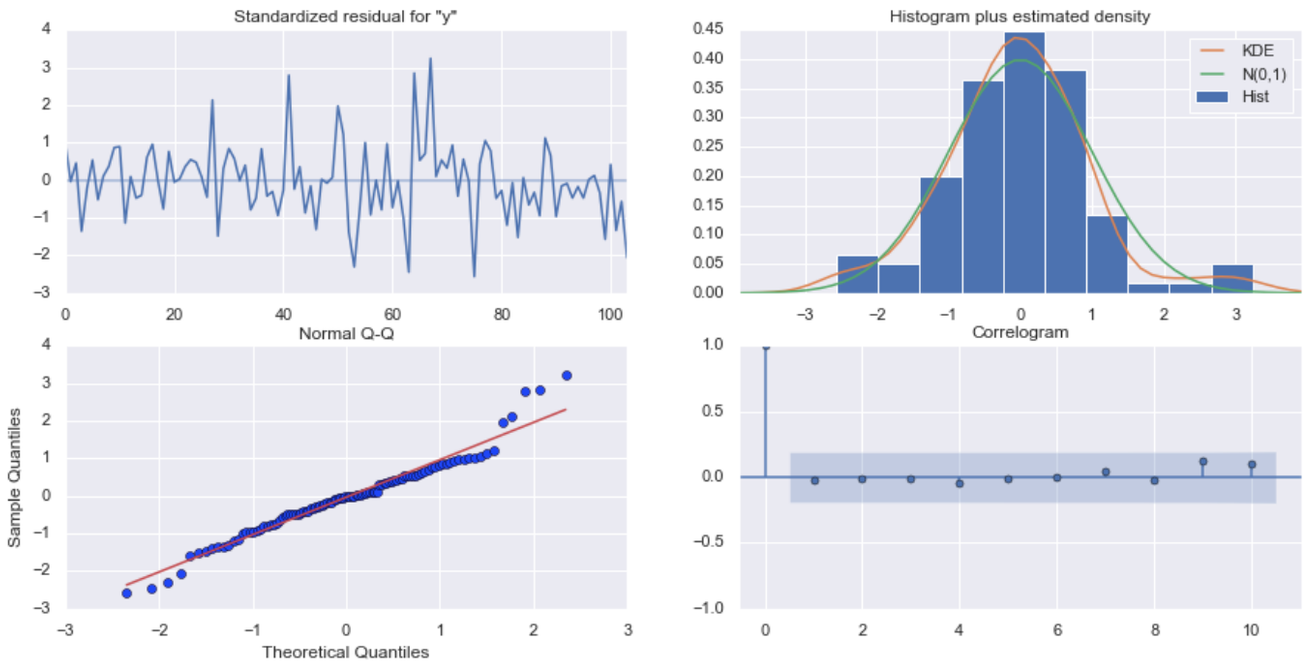

Fig : 26 Diagnostics Plot of Auto_SARIMA (1, 1, 2)(1, 0, 2, 12)

Insights

- Form Standardize residual for "Y" we infer that is no pattern found in the error part.
- By looking at the diagnostics plots like (Histogram and Normal Q-Q) we infer that errors are almost normally distributed.
- From correlogram we infer that correlation coff. of error terms is zero and we can seee that none of the errors are significant.(i.e No error term crossing the cut-off).

Summary Frame with alpha =0.05

y	mean	mean_se	mean_ci_lower	mean_ci_upper
0	1327.403511	388.342697	566.265811	2088.541210
1	1315.128897	402.006691	527.210261	2103.047533
2	1621.616642	402.000289	833.710554	2409.522731
3	1598.882450	407.237670	800.711284	2397.053615
4	1392.710461	407.967866	593.108136	2192.312786

Tab:36 SummaryFrame of Auto _SARIMA(1, 1, 2)(1, 0, 2, 12) at alpha = 0.05

Observation:

From the above table at 95 % of confidence interval we have the mean value , mean_std , and lower limit & upper limit for forecast of Auto _SARIMA(1, 1, 2)(1, 0, 2, 12) at alpha = 0.05

Test Data - RMSE of Auto_SARIMA(1, 1, 2)(1, 0, 2, 12) Model is 528.591167953671

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.

ARIMA model based on the cut-off points of ACF and PACF.

ACF and PACF plots to get the values for p and q.

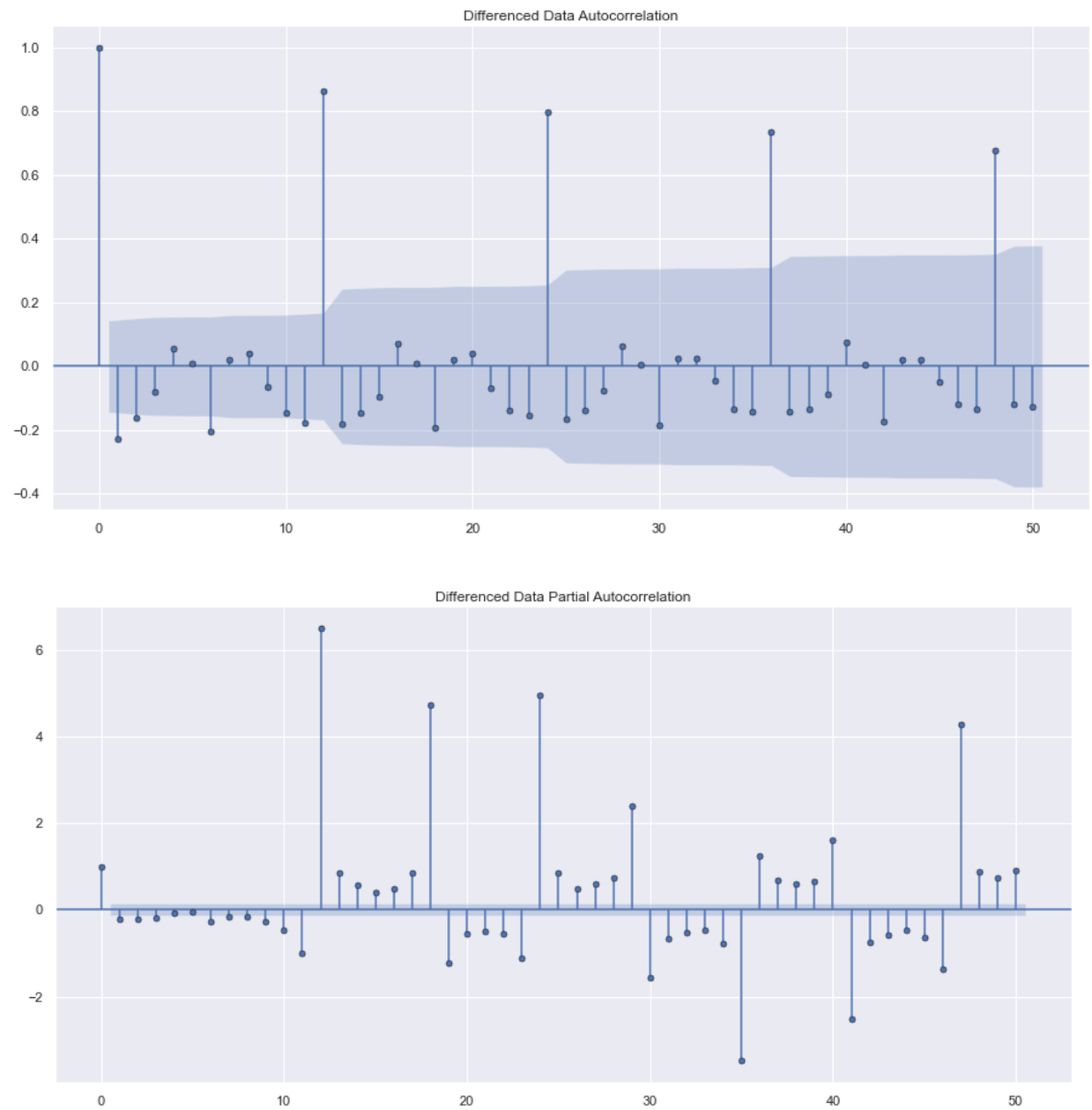

Fig : 27 ACF / PACF Plot for ARIMA Model

Insights

- The Auto-Regressive parameter in an ARIMA model is 'p' which comes from the significant lag before which the PACF plot cuts-off to 3.
- The Moving-Average parameter in an ARIMA model is 'q' which comes from the significant lag before the ACF plot cuts-off to 2.
- The value for d =1 , as we see differencing of order 1 makes the series stationary .

Building ARIMA model based on the cut-off points of ACF and PACF.

```
manual_ARIMA = ARIMA(train['Sparkling Wine Sales'].astype('float64'),
order=(3,1,2),freq='M')

results_manual_ARIMA = manual_ARIMA.fit()
```

Manual_ARIMA Model Results

ARIMA Model Results

Dep. Variable:	D.Sparkling Wine Sales	No. Observations:	131
Model:	ARIMA(3, 1, 2)	Log Likelihood	-1107.464
Method:	css-mle	S.D. of innovations	1106.284
Date:	Sun, 19 Dec 2021	AIC	2228.928
Time:	00:05:40	BIC	2249.054
Sample:	02-29-1980 - 12-31-1990	HQIC	2237.106

	coef	std err	z	P>|z|	[0.025	0.975]
const	5.9847	3.643	1.643	0.100	-1.156	13.126
ar.L1.D.Sparkling Wine Sales	-0.4419	1.77e-05	-2.49e+04	0.000	-0.442	-0.442
ar.L2.D.Sparkling Wine Sales	0.3080	6.82e-05	4516.858	0.000	0.308	0.308
ar.L3.D.Sparkling Wine Sales	-0.2501	5.47e-05	-4573.432	0.000	-0.250	-0.250
ma.L1.D.Sparkling Wine Sales	-0.0009	0.019	-0.045	0.965	-0.039	0.037
ma.L2.D.Sparkling Wine Sales	-0.9991	0.019	-51.736	0.000	-1.037	-0.961

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	-1.0000	-0.0000j	1.0000	-0.5000
AR.2	1.1157	-1.6594j	1.9996	-0.1558
AR.3	1.1157	+1.6594j	1.9996	0.1558
MA.1	1.0000	+0.0000j	1.0000	0.0000
MA.2	-1.0009	+0.0000j	1.0009	0.5000

Tab:37 Result Manual_ARIMA (3,1,2) Model

Conclusion

By looking the Manual_ARIMA model summary we found that the coff. for all the components and p-values of the component ma.L1.D.Sparkling Wine Sales of the SARIMA model have p-value more than 0.05 so ma.L1.D.Sparkling Wine Sales have insignificant values.

Test Data - RMSE of Manual_ARIMA (3,1,2) Model with p =3 d = 1 and q=2 is 1379.0900674284858

Manual SARIMA model based on the cut-off points of ACF and PACF.

From the original sparkling wine sales time series we see that there is a trend and a seasonality. So, now we take a seasonal differencing and check the series.Here we take diff order of 6 because we have seasonality of that order.

Checking the Stationarity on train data by taking the seasonal differencing of order 6- Dickey-Fuller Test

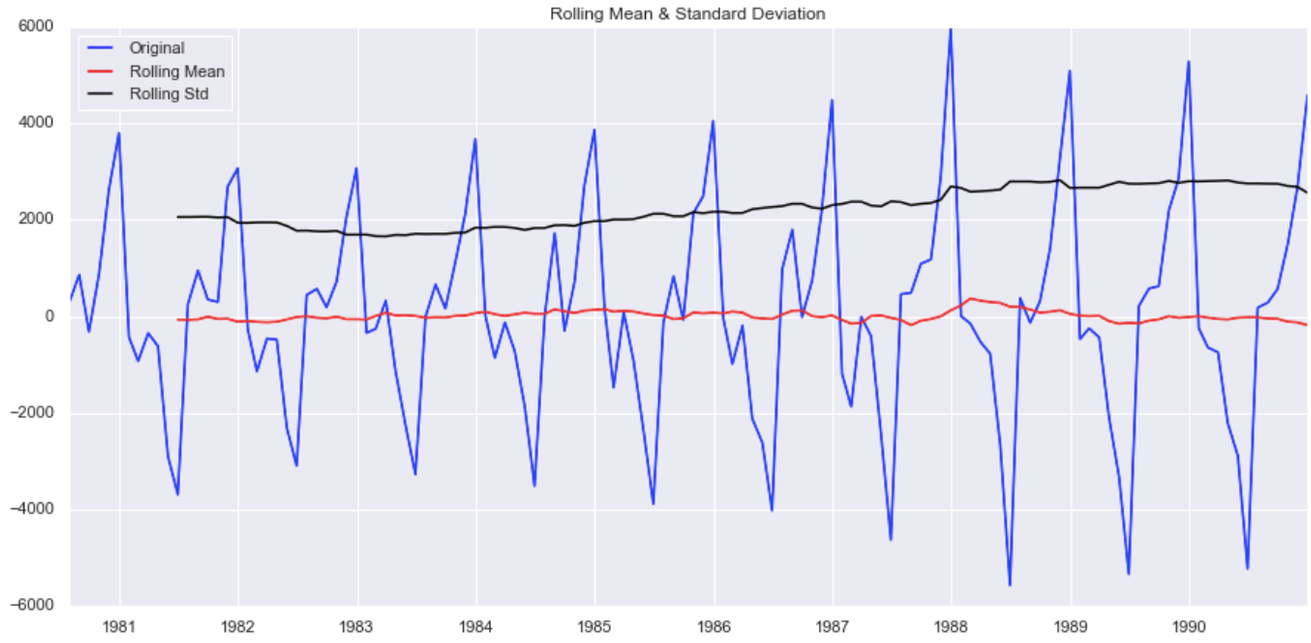

Fig : 28 Checking the Stationarity by taking the seasonal differencing of order 6- Dickey-Fuller Test

```
Results of Dickey-Fuller Test:
Test Statistic                -8.181919e+00
p-value                        8.088278e-13
#Lags Used                     6.000000e+00
Number of Observations Used    1.190000e+02
Critical Value (1%)           -3.486535e+00
Critical Value (5%)           -2.886151e+00
Critical Value (10%)          -2.579896e+00
dtype: float64
```

Tab:38 Dickey - Fuller Test Result on Train Data with seasonal diff of order 6

Result

By the looking at the p-value we conclude that the series is stationary.We donot want further differencing of seasonal differenced series as we found it stationary so we get value of D=0.

Ploting ACF AND PACF Plots with Seasonal Diff of 6 to get the Value of P AND Q.

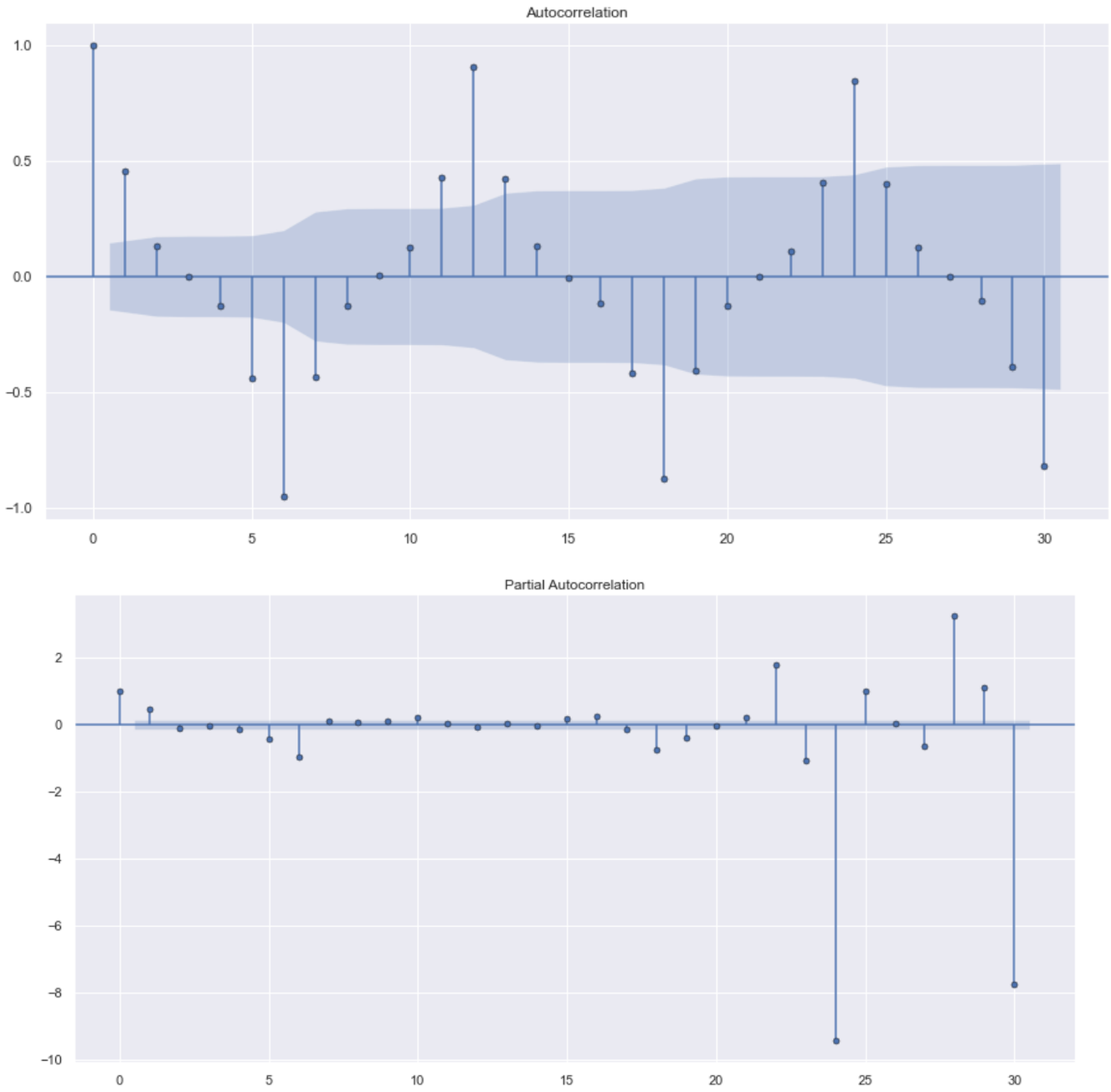

Fig : 29 Seasonal Diff (6) ACF / PACF Plot for SARIMA Model

Conclusion:

Here, we have taken alpha=0.05.

We are going to take the seasonal period as 6. We will keep the p ,d and q parameters same as the Mannual 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.We get value of P = 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. We get value of Q=1 Remember to check the ACF and the PACF plots only at multiples of 6 (since 6 is the seasonal period).

Building the Manual SARIMA Model at Seasonality 6

```
manual_SARIMA_6 = sm.tsa.statespace.SARIMAX(train['Sparkling Wine Sales'].values,
                                            order=(3, 1, 2),
                                            seasonal_order=(2, 0, 1, 6),
                                            enforce_stationarity=False,
                                            enforce_invertibility=False)
results_manual_SARIMA_6 = manual_SARIMA_6.fit(maxiter=1000)
```

Manual_SARIMA (3,1,2)(2,0,1,6)Model Results

SARIMAX Results

Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(3, 1, 2)x(2, 0, [1], 6)	Log Likelihood	-865.222
Date:	Sun, 19 Dec 2021	AIC	1748.443
Time:	00:05:42	BIC	1773.226
Sample:	0	HQIC	1758.503
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.5337	0.197	-2.704	0.007	-0.921	-0.147
ar.L2	0.0669	0.108	0.617	0.537	-0.146	0.279
ar.L3	-0.0168	0.086	-0.195	0.845	-0.186	0.152
ma.L1	-0.1670	0.213	-0.784	0.433	-0.585	0.251
ma.L2	-0.8330	0.197	-4.220	0.000	-1.220	-0.446
ar.S.L6	-0.0299	0.055	-0.545	0.586	-0.138	0.078
ar.S.L12	0.9500	0.032	29.733	0.000	0.887	1.013
ma.S.L6	0.0796	0.139	0.573	0.567	-0.193	0.352
sigma2	1.712e+05	1.21e-06	1.41e+11	0.000	1.71e+05	1.71e+05

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	7.27
Prob(Q):	0.97	Prob(JB):	0.03
Heteroskedasticity (H):	1.96	Skew:	0.14
Prob(H) (two-sided):	0.04	Kurtosis:	4.19

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 4.22e+27. Standard errors may be unstable.

Tab:39 Result Manual_SARIMA (3,1,2) (2,0,1,6) Model

Conclusion

By looking the model summary we found that the p-values of the components like ar.L2 ,ar.L3 , ma.L1 , ar.S.L6 and ma.S.L6 is more than 0.05.

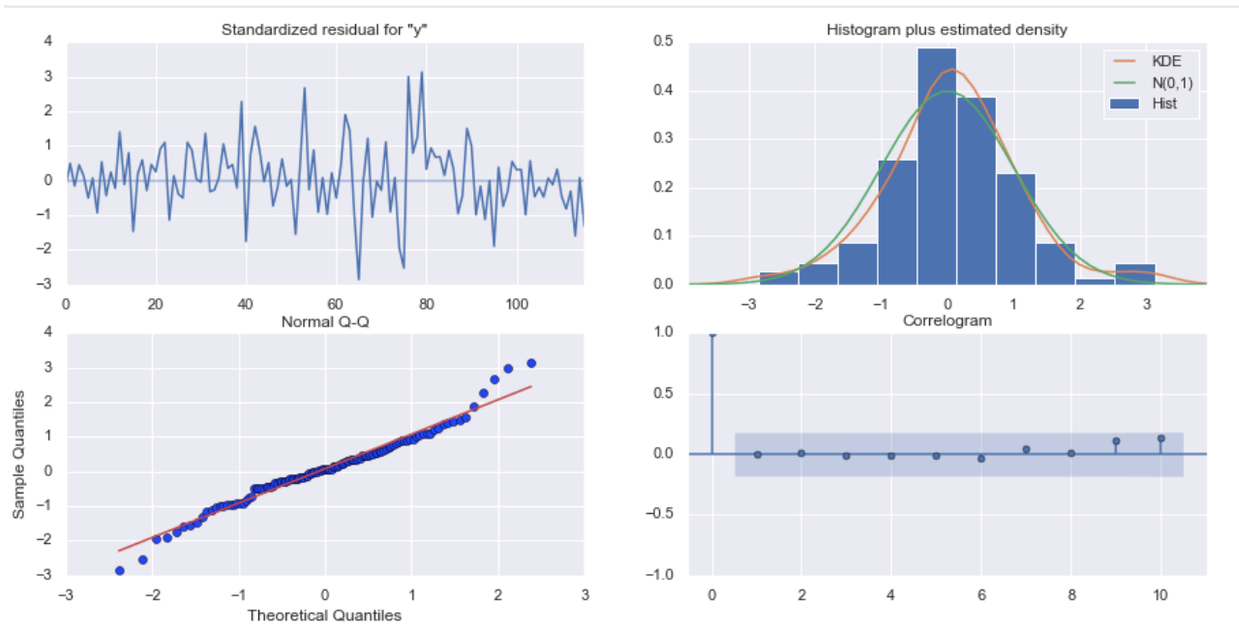

Fig : 30 Diagnostics Plot of Manual_SARIMA (3, 1, 2)(2, 0, 1, 6)

Insights

- Form Standardize residual for "Y" we infer that is no pattern found in the error part.
- By looking at the diagnostics plots like (Histogram and Normal Q-Q) we infer that errors are almost normally distributed.
- From correlogram we infer that correlation coff. of error terms is zero and we can seee that none of the errors are significant.(i.e No error term crossing the cut-off).

Results

The model diagnostics plot looks okay.

Summary Frame with alpha =0.05

	y	mean	mean_se	mean_ci_lower	mean_ci_upper
0	1566.159861	415.445780	751.901096	2380.418627
1	1446.132287	434.636978	594.259464	2298.005110
2	1828.559188	435.951872	974.109220	2683.009155
3	1627.470666	436.709747	771.535291	2483.406041
4	1567.439846	436.854451	711.220856	2423.658836

Tab:40 SummaryFrame of Manual _SARIMA(3,1,2) (2,0,1,6) Model at alpha = 0.05

Observation:

From the above table at 95 % of confidence interval we have the mean value , mean_std , and lower limit & upper limit for forecast of Manual_SARIMA(3,1,2) (2,0,1,6) Model at alpha = 0.05

Test Data - RMSE of Manual_SARIMA(3, 1, 2)(2, 0, 1, 6) Model is 343.35524809778116

Build a version of the Manual SARIMA model for which the best parameters are selected by looking at the ACF and the PACF plots. - Seasonality at 12.

From the original sparkling wine sales time series we see that there is a trend and a seasonality. So, now we take a seasonal differencing and check the series.Here we take diff order of 12 because we have seasonality of that order.

Checking the Stationarity on train data by taking the seasonal differencing of order 12- Dickey-Fuller Test

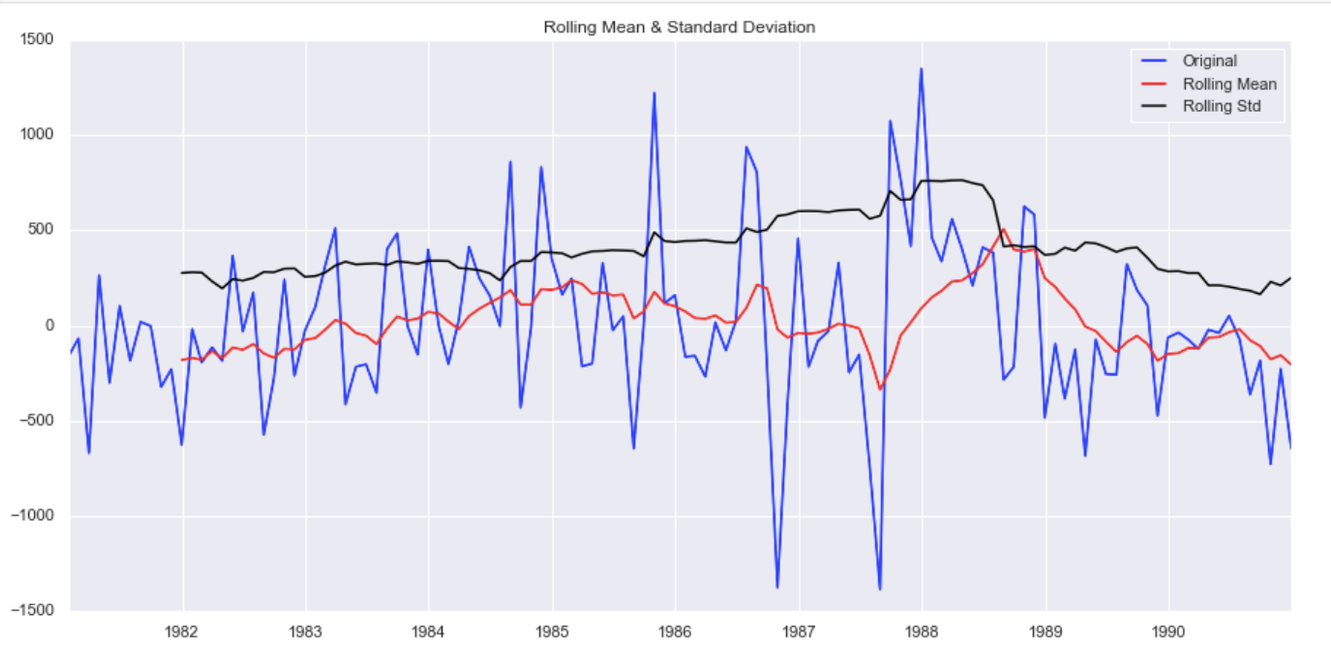

Fig : 31 Checking the Stationarity by taking the seasonal differencing of order 12- Dickey-Fuller Test

```
Results of Dickey-Fuller Test:
Test Statistic                  -3.136812
p-value                          0.023946
#Lags Used                      11.000000
Number of Observations Used    108.000000
Critical Value (1%)             -3.492401
Critical Value (5%)             -2.888697
Critical Value (10%)            -2.581255
dtype: float64
```

Tab:41 Dickey - Fuller Test Result on Train Data with seasonal diff of order 12

Result

As the looking ate the p-value we conclude that the series is stationary.We donot want further differencing of seasonal differenced series as we found it stationary so we get value of D=0.

Ploting ACF AND PACF Plots with Seasonal Diff to get the Value of P AND Q.

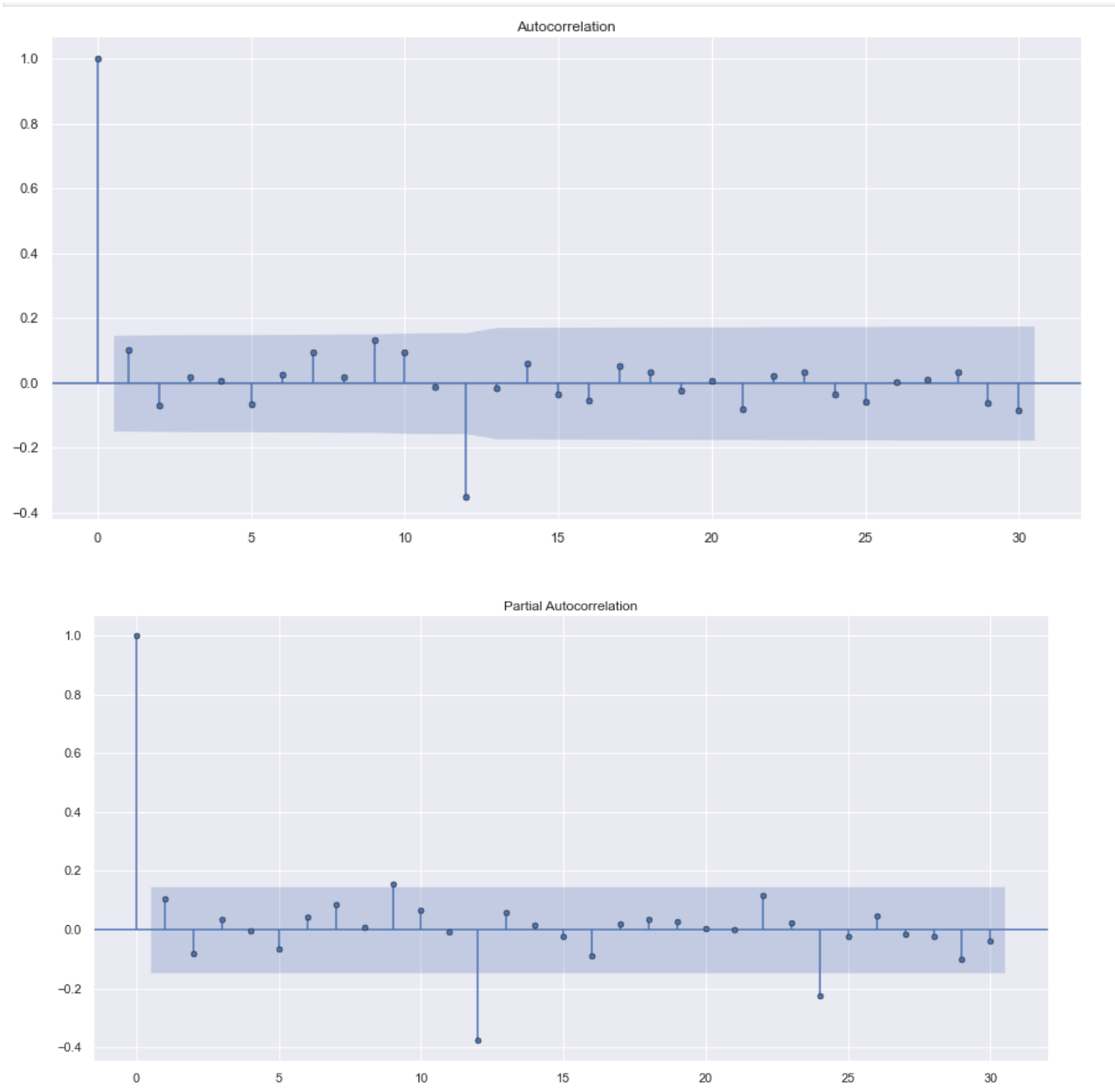

Fig : 32 Seasonal Diff (12) ACF / PACF Plot for SARIMA Model

Observation

Here, we have taken alpha=0.05. We are going to take the seasonal period as 12. We will keep the p ,d and q parameters same as the Mannual 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.We get value of P = 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. We get value of Q= 0.

Building the Manual SARIMA Model at Seasonality 12.

```
manual_SARIMA_12 = sm.tsa.statespace.SARIMAX(train['Sparkling Wine Sales'].values,
                                             order=(3, 1, 2),
                                             seasonal_order=(0, 0, 0, 12),
                                             enforce_stationarity=False,
                                             enforce_invertibility=False)
results_manual_SARIMA_12 = manual_SARIMA_12.fit(maxiter=1000)
```

Manual_SARIMA (3,1,2)(0,0,0,12)Model Results

SARIMAX Results

Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(3, 1, 2)	Log Likelihood	-1087.657
Date:	Sun, 19 Dec 2021	AIC	2187.315
Time:	00:05:44	BIC	2204.427
Sample:	0	HQIC	2194.267
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.4328	0.210	-2.066	0.039	-0.843	-0.022
ar.L2	0.3206	0.167	1.926	0.054	-0.006	0.647
ar.L3	-0.2487	0.258	-0.964	0.335	-0.754	0.257
ma.L1	0.0129	0.196	0.066	0.948	-0.371	0.396
ma.L2	-0.9693	0.176	-5.500	0.000	-1.315	-0.624
sigma2	1.768e+06	1.28e-07	1.38e+13	0.000	1.77e+06	1.77e+06

Ljung-Box (L1) (Q):	0.02	Jarque-Bera (JB):	3.70
Prob(Q):	0.89	Prob(JB):	0.16
Heteroskedasticity (H):	2.59	Skew:	0.38
Prob(H) (two-sided):	0.00	Kurtosis:	3.34

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 4.45e+28. Standard errors may be unstable.

Tab:42 Result Manual_SARIMA (3,1,2) (0,0,0,12) Model

Conclusion

By looking the model summary we found that the p-values of the components like ar.L3 , and ma.L1 is more than 0.05.

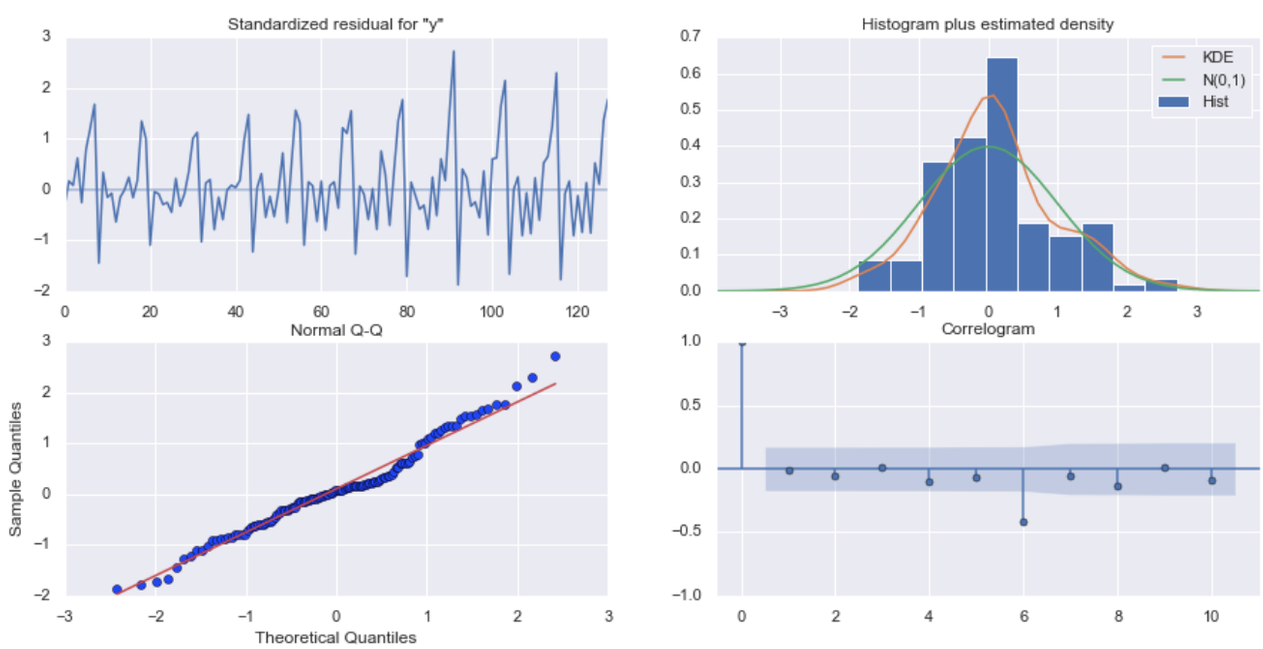

Fig : 33 Diagnostics Plot of Manual_SARIMA (3, 1, 2)(0, 0, 0, 12)

Results

The model Diagnostics Plot of Manual_SARIMA (3, 1, 2)(0, 0, 0, 12) looks poor as compared to the Diagnostics Plot of Manual_SARIMA (3, 1, 2)(2, 0, 1, 6).

Summary Frame with alpha =0.05

	y	mean	mean_se	mean_ci_lower	mean_ci_upper
0		3769.532791	1330.849865	1161.114987	6377.950595
1		2745.173158	1537.441023	-268.155875	5758.502190
2		2020.267505	1544.907583	-1007.695718	5048.230728
3		2572.088046	1547.606616	-461.165183	5605.341275
4		2355.645527	1548.248218	-678.865219	5390.156273

Tab:43 SummaryFrame of Manual _SARIMA(3,1,2) (0,0,0,12) Model at alpha = 0.05

Observation:

From the above table at 95 % of confidence interval we have the mean value , mean_std , and lower limit & upper limit for forecast of Manual_SARIMA(3,1,2) (0,0,0,12) Model at alpha = 0.05

Test Data - RMSE of Manual_SARIMA(3, 1, 2)(0, 0, 0, 12) Model is 1282.4983982484705

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

	Test RMSE
Regression On Time	1389.135175
Naive Model	3864.279352
Simple Average Model	1275.081804
2 Point Trailing Moving Average	813.400684
4 Point Trailing Moving Average	1156.589694
6 PointTrailing Moving Average	1283.927428
9 PointTrailing Moving Average	1346.278315
Alpha=0.0496,Simple Exponential Smoothing	1316.034674
Alpha=0.3,Simple Exponential Smoothing	1935.507132
Alpha=0.3,Beta=0.3,DoubleExponentialSmoothing	18259.110704
Alpha=0.111,Beta=0.061,Gamma=0.395,TripleExponentialSmoothing	469.591666
Alpha=0.3,Beta=0.3,Gamma=0.3,TripleExponentialSmoothing	392.786198
Auto_ARIMA(2,1,2)	1374.336485
Auto_SARIMA(1,1,2)(2,0,2,6)	626.958678
Auto_SARIMA(1,1,2)(1,0,2,12)	528.591168
Manual_ARIMA(3,1,2)	1379.090067
Manual SARIMA(3,1,2)(2,0,1,6)	343.355248
Manual SARIMA(3,1,2)(0,0,0,12)	1282.498398

Tab:44 Result of All Models With Parameter and Test RMSE

	Test RMSE
Manual SARIMA(3,1,2)(2,0,1,6)	343.355248
Alpha=0.3,Beta=0.3,Gamma=0.3,TripleExponentialSmoothing	392.786198
Alpha=0.111,Beta=0.061,Gamma=0.395,TripleExponentialSmoothing	469.591666
Auto_SARIMA(1,1,2)(1,0,2,12)	528.591168
Auto_SARIMA(1,1,2)(2,0,2,6)	626.958678

Tab:45 Result of Top 5 Models With Parameter and Test RMSE

Result

From the above table , we found **Manual SARIMA(3,1,2)(2,0,1,6)** Model have the least RMSE of **343.355248** on the Test Data. So , we conclude that **Manual SARIMA(3,1,2)(2,0,1,6) Model** will be our finalised model on the complete data to predict 12 months into the future.

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.

Building the most optimum model on the Full Data - From the above Test RMSE results we conclude that Manual SARIMA(3,1,2)(2,0,1,6) is the best model with least RMSE among all the models.We finalise Manual SARIMA(3,1,2)(2,0,1,6) is best model to built on full data to predict 12 months into the future with appropriate confidence intervals/bands.

```
full_data_model = sm.tsa.statespace.SARIMAX(df['Sparkling Wine Sales'],
                                            order=(3,1,2),
                                            seasonal_order=(2, 0, 1, 6),
                                            enforce_stationarity=False,
                                            enforce_invertibility=False)
results_full_data_model = full_data_model.fit(maxiter=1000)
```

Full Model Results

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                 Sparkling Wine Sales   No. Observations:                  187
Model:             SARIMAX(3, 1, 2)x(2, 0, [1], 6)   Log Likelihood               -1307.874
Date:                             Sun, 19 Dec 2021   AIC                           2633.747
Time:                                     00:05:47   BIC                           2662.022
Sample:                                 01-31-1980   HQIC                          2645.220
                                      - 07-31-1995
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -2.2752      0.064    -35.547      0.000      -2.401      -2.150
ar.L2         -1.6992      0.125    -13.648      0.000      -1.943      -1.455
ar.L3         -0.3862      0.067     -5.745      0.000      -0.518      -0.254
ma.L1          1.9589      0.100     19.537      0.000       1.762       2.155
ma.L2          0.9982      0.103      9.709      0.000       0.797       1.200
ar.S.L6       -0.0524      0.040     -1.312      0.189      -0.131       0.026
ar.S.L12       0.9314      0.031     29.857      0.000       0.870       0.993
ma.S.L6        0.3539      0.090      3.951      0.000       0.178       0.529
sigma2      2.437e+05   8.16e-07   2.99e+11      0.000    2.44e+05    2.44e+05
===================================================================================
Ljung-Box (L1) (Q):                   2.93   Jarque-Bera (JB):                21.11
Prob(Q):                              0.09   Prob(JB):                         0.00
Heteroskedasticity (H):               1.30   Skew:                             0.47
Prob(H) (two-sided):                  0.32   Kurtosis:                         4.45
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 6.26e+26. Standard errors may be unstable.
```

Tab:46 Result Full SARIMA Model(3,1,2) (2,0,1,6) Model

Conclusion

By looking the Full model summary we found that the coff. for all the components and p-values of the component ar.S.L6 is insignificant.

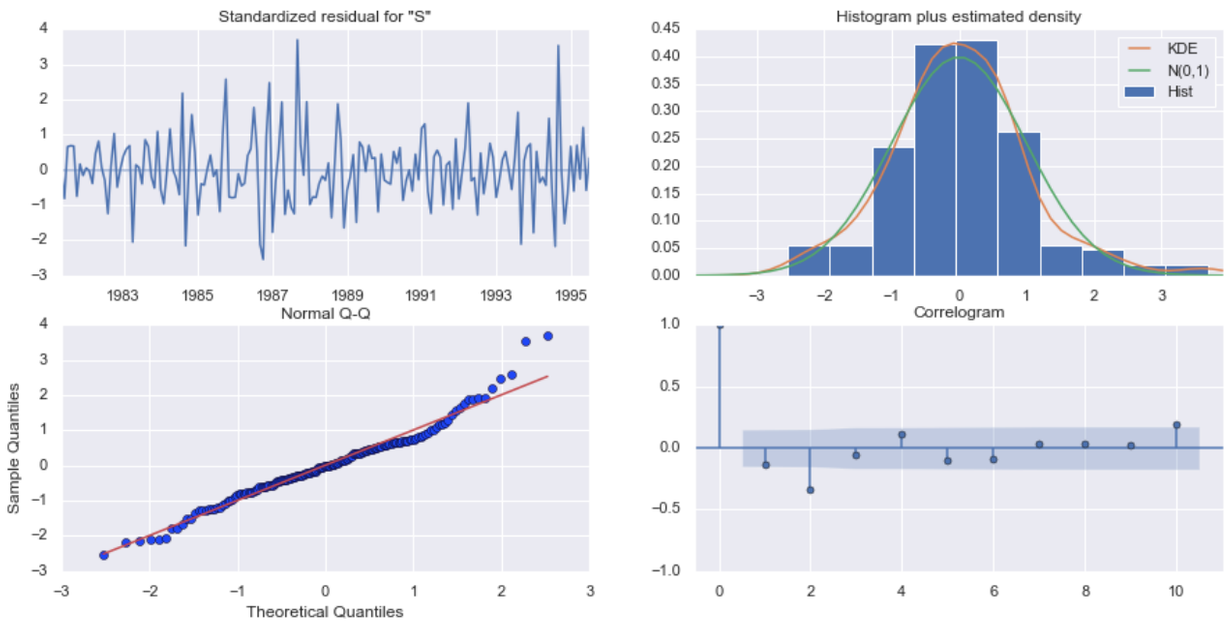

Fig : 34 Diagnostics Plot of Full SARIMA Model (3,1,2) (2,0,1,6)

Results

The model Diagnostics Plot of Full SARIMA Model (3, 1, 2)(2, 0, 1, 6) looks okay.

Summary Frame of Full Model with alpha =0.05 appropriate confidence intervals/bands to predict 12 months into the future

Sparkling Wine Sales	mean	mean_se	mean_ci_lower	mean_ci_upper
1995-08-31	1931.746888	495.998331	959.608022	2903.885753
1995-09-30	2870.384593	598.268696	1697.799495	4042.969691
1995-10-31	3656.218245	692.895693	2298.167642	5014.268848
1995-11-30	3862.088083	799.162541	2295.758284	5428.417881
1995-12-31	6009.725671	862.484226	4319.287650	7700.163691

Sparkling Wine Sales	mean	mean_se	mean_ci_lower	mean_ci_upper
1996-03-31	2278.298766	1174.522437	-23.722909	4580.320441
1996-04-30	1893.001580	1258.137207	-572.902034	4358.905195
1996-05-31	2096.400050	1351.609881	-552.706638	4745.506737
1996-06-30	1554.347009	1428.416961	-1245.298791	4353.992808
1996-07-31	2580.567060	1506.185391	-371.502060	5532.636180

Tab : 47 Records of Summary Frame of Full Model with alpha =0.05 appropriate confidence intervals/bands.

Observation:

From the above table at 95 % of confidence interval we have the mean value , mean_std , and lower limit & upper limit for forecast of Full_SARIMA(3,1,2) (2,0,1,6) Model at alpha = 0.05

RMSE of Full_SARIMA(3, 1, 2)(2, 0, 1, 6) Model is 861.8998226009788

Plot of the forecast on full data along with the confidence band

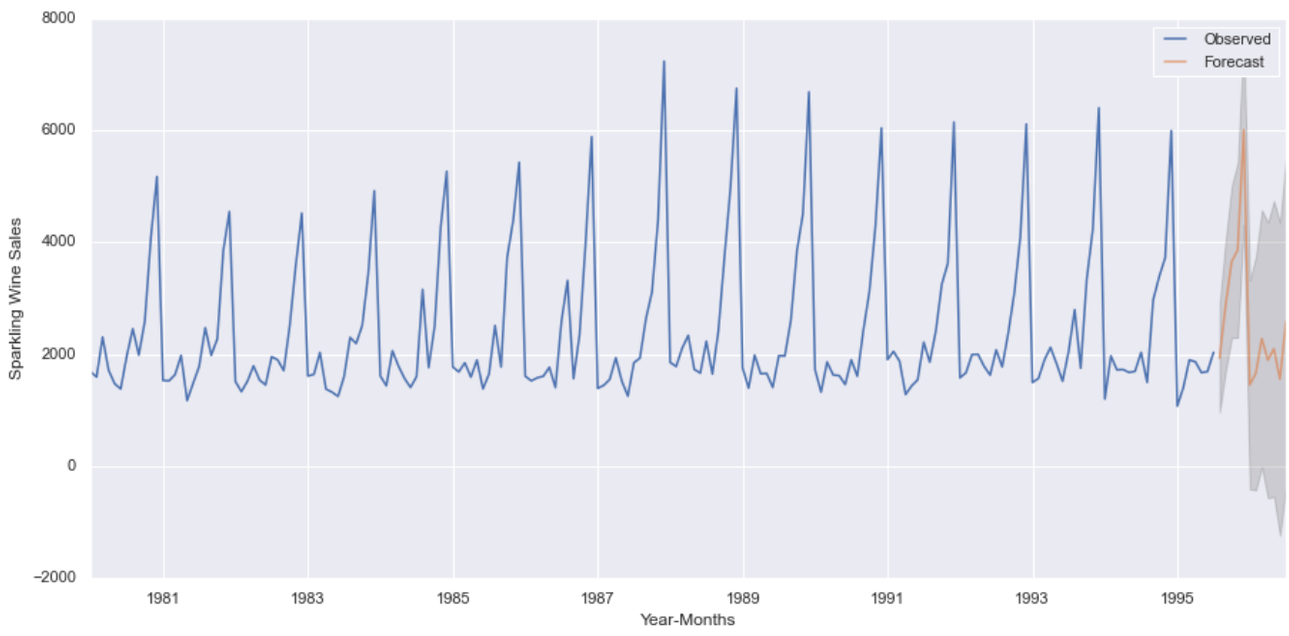

Fig : 35 Plot of the forecast on full data along with the confidence band

Insights :

From the the above plot we infer that with 95% of the confidence level we found that forecast also follows the same pattern as original sparkling wine sales series follows.

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

The purpose of this whole exercise is to explore the dataset , analyse and forecast Sparkling Wine Sales in the 20th century.Here we perform the exploratory data analysis & apply various time series forecasting models like Linear Regression , Navie Forecast ,Simple Average , Moving Average and various kind of exponential smoothing models like (Simple , Double , Triple Exponential) and ARIMA / SARIMA models on the Sparkling Wine Sales dataset and check their RMSE on the test data , model which gives the least RMSE will be the final model for us to analyse and forecast the Sparkling Wine Sales in the 20th century.

Insights of EDA / Data Visualization and Time Series Forecasting Models :

We notice that in original sparkling wine time series there is some kind of **increasing trend in the initial years** which stabilizes as the years (or more specifically the months in each of the years) progresses and after 1988 sales decreases . There is some kind of **seasonality** associated in the data as well.

Description of the Original Sparkling Wine Sales Time Series

- Sparkling Wine Sales ranges from a minimum of 1070 to maximum of 7242.
- Mean of the Sparkling Wine Sales is around 2402.417112.
- Standard Deviation of the Sparkling Wine Sales is 1295.111540.
- 25% , 50% (median) and 75 % of Sparkling Wine Sales are 1605 , 1874 and 2549.

Information about the Original Sparkling Wine Sales Time Series

From the above results we can see that there is no null values present in the dataset.Their are total 187 entries of Sparkling wines Sales as per Monthly frequency in this dataset,indexed from 1980-01-31 to 1995-07-31.Sparkling Wine Sales column have d-type of int64. Memory used by the dataset: 2.9 KB.

The Sparkling.csv data set has 187 observations (rows) and 1 variable (column named as Sparkling Wine Sales) in the dataset.

Year on Year boxplot for the Sparkling Wine Sales Insights

- As we got to know from the Time Series plot, the box-plots over here also indicates a measure of trend being present. Also, we see that the Sparkling Wine Sales have outliers for the years.
- Box-plot of Year 1988 have max median value,we can clearly infer that year 1988 have maximum Sparkling Wine Sales.
- Box-plot of Year 1995 have min median value,we can clearly infer that year 1995 have minimum Sparkling Wine Sales.

Monthly Box-Plot for the Sparkling Wine Sales Taking all the Years into Account Insights

- **T**he Box-Plots for the monthly Sparkling Wine Sales for different years doesn't show too much outliers only month 1 , 2 & 7 show outliers , rest doesn't show any outliers.
- From September to December the Sparkling Wine Sales increasing , so this the period where the Sparkling Wine Sales is highest.
- There is seasonality also every year from September to December the Sparkling Wine Sales increasing.
- In June month we have lowest sales of the Sparkling Wine.

Month-plot of Sparkling Wine Sales Time Series Insights

- As noticed in the above box-plot we get same result from here too. From September to December Sparkling Wine Sales goes on increasing.
- December month have the highest sales of the Sparkling Wine while June month have lowest sales of the Sparkling Wine.

Plot for different months for different years of Original Time Series Insights

This plot gives us information about the monthly trend across the years.Here in this plot every line is a month tells us about the sales of Sparkling Wines of each month across the year.This is way to show year on year monthly trend.

- From the above plot we clearly infer that December month have highest sales of Sparking Wine.
- June month have the lowest sales of the Sparkling Wine.

Here we apply various time series forecasting models like Linear Regression , Navie Forecast ,Simple Average , Moving Average and various kind of exponential smoothing models like (Simple , Double , Triple Exponential) and ARIMA / SARIMA models on the Sparkling Wine Sales dataset and check their RMSE on the test data , After comparing the TEST RMSE of all the model that we built. We come know that the TEST RMSE of **Manual SARIMA(3,1,2)(2,0,1,6)** is least among all the models with different parameters. So we take **Manual SARIMA(3,1,2)(2,0,1,6) model** to built complete data and predict 12 months into the future with appropriate confidence intervals/bands.

RMSE of the Full Model is - 861.8998226009788

From the Plot of the forecast on full data along with the confidence band we infer that with 95% of the confidence level we found that forecast also follows the same pattern as original sparkling wine sales series follows.

Recommendations:

The ABC Estate Wines company should focus on key strengths & develop marketing strategies to promote Sparkling Wine Sales . As we From Sept to Dec the Sparkling wine sales are higher.But in the month of May & June it was low showing less people consumes wine in this period so wine company can run various offers during this period to boost their sparkling wine sales.ABC Estate wines company gives various offers like buy 1 get 1 or some interesting gifts on purchase of sparkling wine to attract more customers.Introduce some new flavour of sparkling wine can attract consumers which leads to increase in their wine sales during this period.