Project - Time Series Forecasting

Name: Varsha Srinivasan

Table of Contents

LIST OF TABLES

LIST OF FIGURES

PROBLEM 1 - Rose

Problem Statement

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 Rose Wine Sales in the 20th century.

Introduction

The purpose of this whole exercise is to perform exploratory data analysis and perform Time Series Forecasting using Exponential Smoothing models, Regression, Naïve Forecast models, Simple Average models, Moving Average models, ARIMA and SARIMA models(using cut-off points of AIC,ACF and PACF plots) to forecast the sales of Rose wine.

Data Description

1. YearMonth: The Year and the Month on which its corresponding units of Rose Wine is sold.
2. Rose: Units of Rose Wine sold.

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

Sample of the dataset:

	YearMonth	Rose
0	1980-01	112.0
1	1980-02	118.0
2	1980-03	129.0
3	1980-04	99.0
4	1980-05	116.0

	YearMonth	Rose
182	1995-03	45.0
183	1995-04	52.0
184	1995-05	28.0
185	1995-06	40.0
186	1995-07	62.0

Table 1: Sample of the Dataset1

The data is read from the excel file and the above tables shows the first and last 5 rows of the dataset.There are 187 rows in the dataframe. The Rose is the variable to be forecasted . YearMonth denotes the year and month values ranging from Jan 1980 to July 1995.

There are no duplicates in the datatset.

```
There are 0 duplicates in the dataset
```

The initial datatype of the columns before indexing are:

```
YearMonth     object
Rose         float64
dtype: object
```

YearMonth column is converted into a Time Stamp index using to_datetime function and YearMonth is dropped.

Time_Stamp	Rose
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

Table 2:TimeStamped Dataset

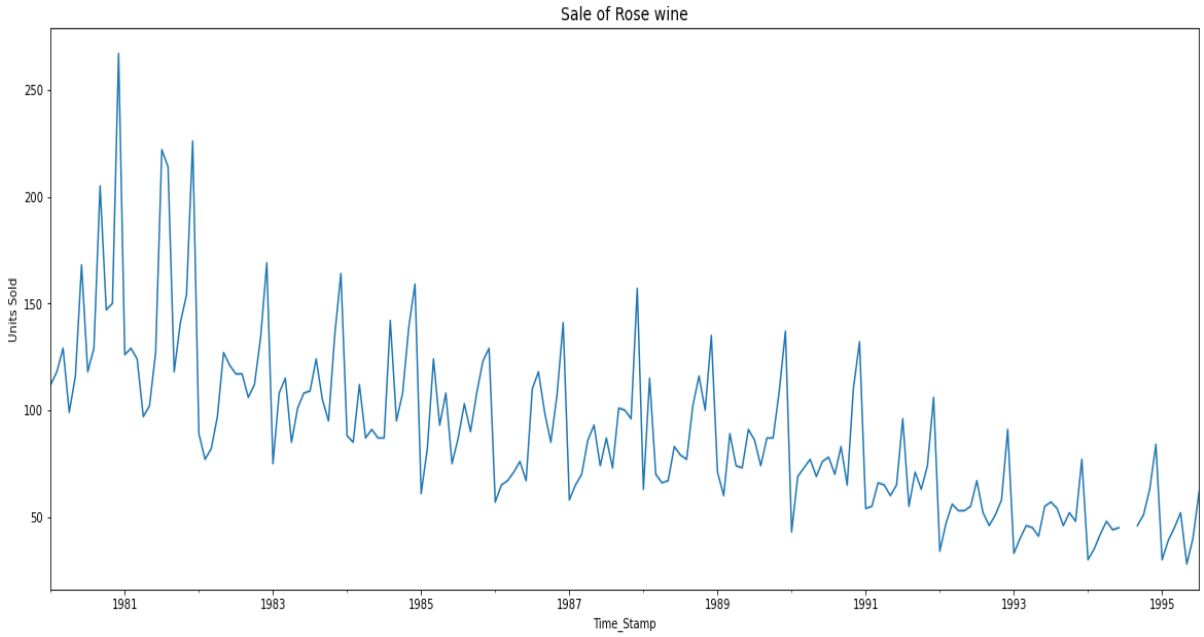

Figure 1: Rose Sales Plot

From the pot we can see a downwards trend and seasonality too. We can see a disconnect in the graph in 1994 . This might be because of missing values.

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

There are 0 duplicates in the dataset.

Data Type and Missing Values

The initial datatype of the columns after indexing are:

```
Rose          float64
dtype: object
```

	Rose
count	185.000000
mean	90.394595
std	39.175344
min	28.000000
25%	63.000000
50%	86.000000
75%	112.000000
max	267.000000

Time_Stamp	Rose
1994-07-01	NaN
1994-08-01	NaN

Table 3: Rose-Data Type and Missing Values

There are 2 null values which can also be inferred from the output of isnull function.

It is imputed with interpolation function with linear method. The values after interpolation are:

Time_Stamp	Rose
1994-07-01	45.333333

Time_Stamp	Rose
1994-08-01	45.666667

Table 4: Imputed Values

Both values are taken as 45 as the decimals doesn't give any information when calculating units sold.

Monthly Sales

Monthly Sales across years

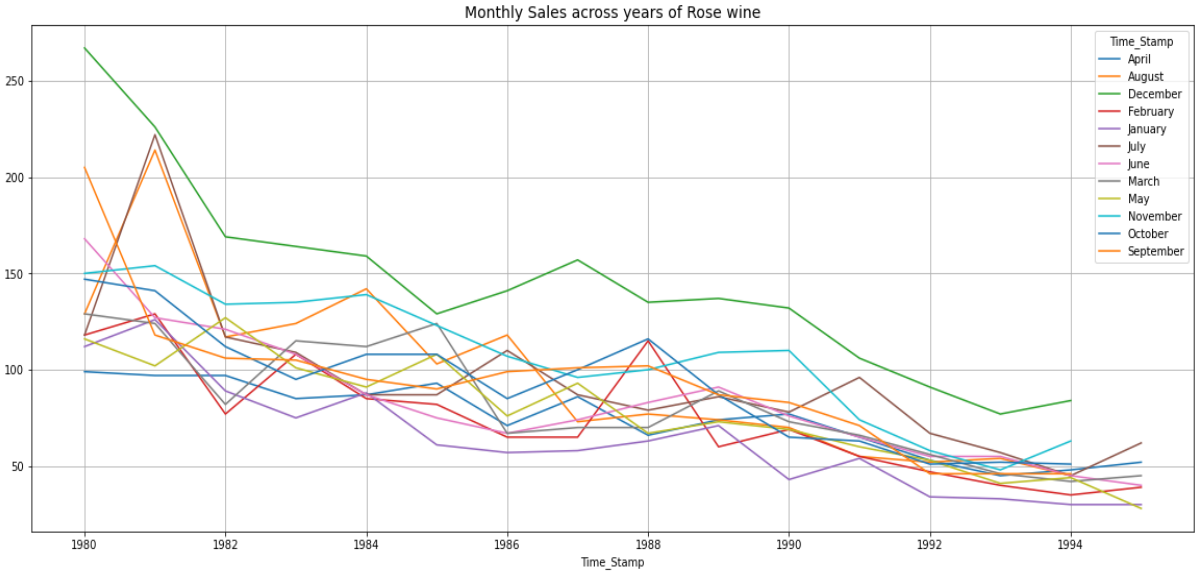

Figure 2:Monthly Sales across years - Rose

From the plot we can infer that December month has the highest sales across all years and January month has the lowest sales across most of the years.

Monthly Sales Sum Barplot of Rose wine

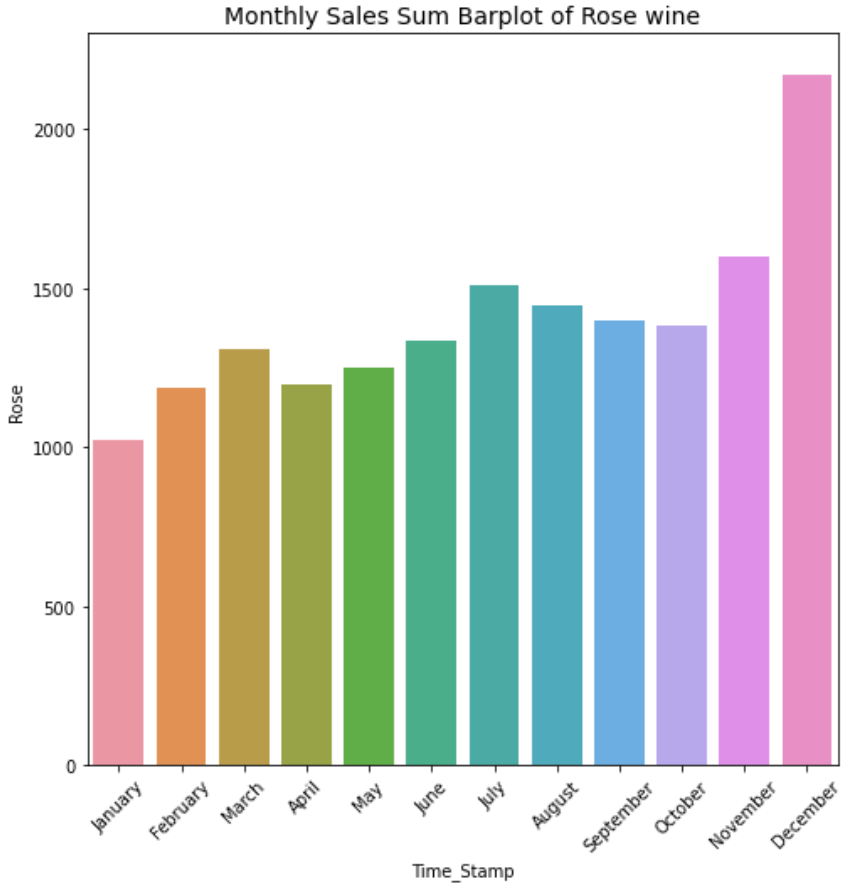

Figure 3:Rose-Barplot-Monthly

December has highest sales combining all the years going above 2000 units sold while January has lowest sales combined with almost 1000 units sold.

Barplot is plotted using sum of the month values from all the years.

Boxplot of Monthly Sales of Rose wine

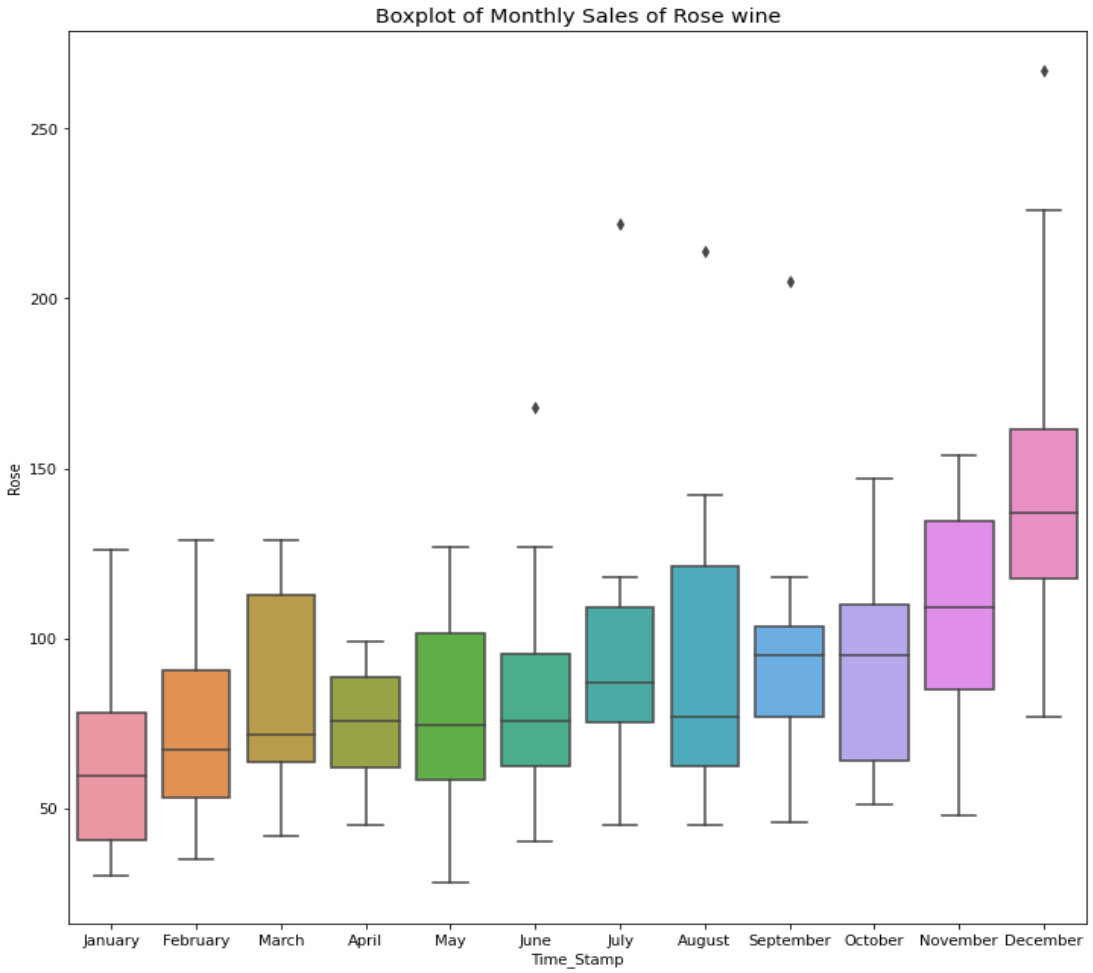

Figure 4:Rose-Boxplot-Monthly

We can see from the boxplot that there is an increasing trend in the sales from the increasing median in the subsequent months.

Month Plot

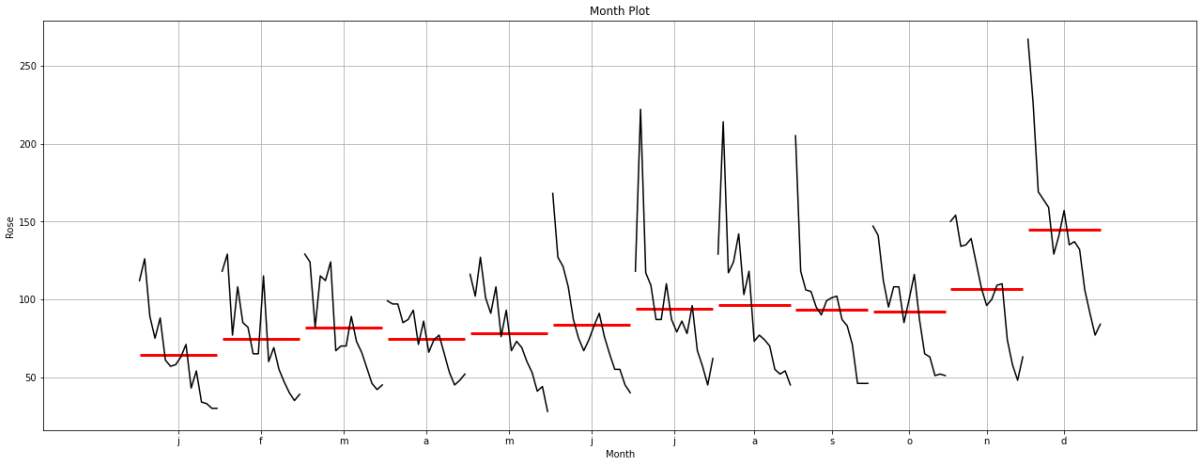

Figure 5:Rose-Monthplot

Month wise there is an increasing trend but year wise there is an decreasing trend in the units sold.

Quarterly Sales

Quarterly Sales across years of Rose wine

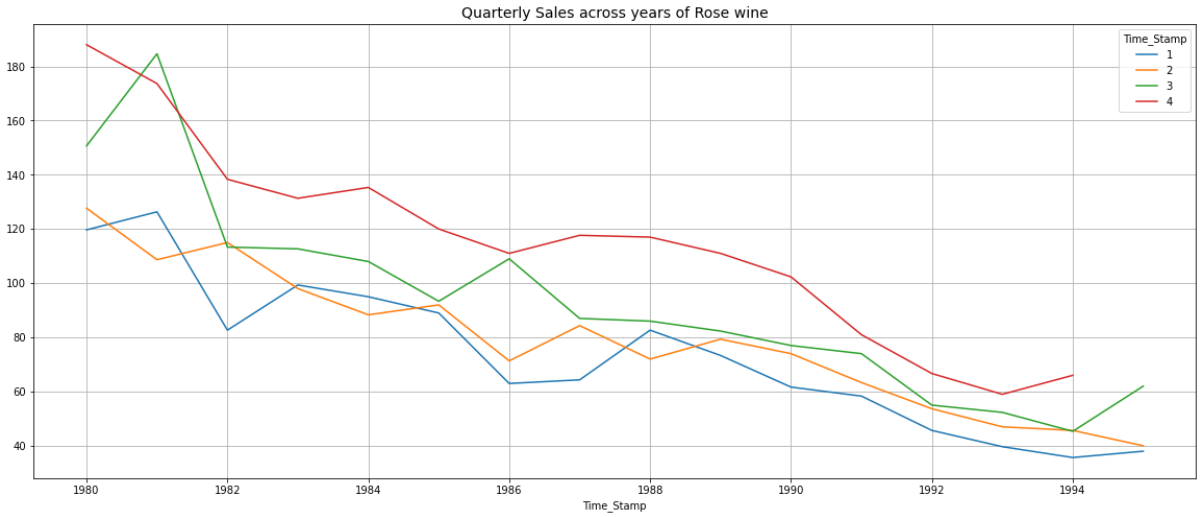

Figure 6:Rose-Quarterly Sales

Fourth quarter has the highest sales across all the years denoted by the red line. First quarter has the lowest sales across most of the years denoted by the blue line.

Barplot of Quarterly Sales Sum of Rose wine

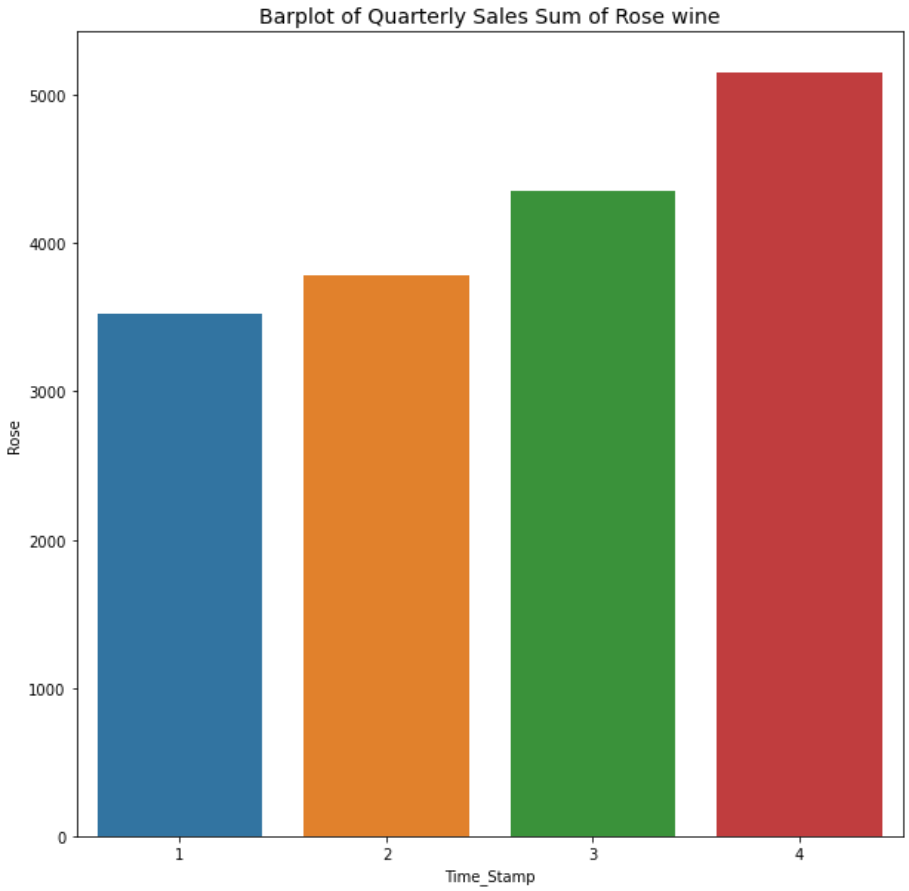

Figure 7:Rose-Quarterly-Barplot

Fourth quarter has the highest sum of sales across all the years with above 5000 units sold. First quarter has the lowest sum of sales across most of the years with almost 3500 units sold.

Boxplot of Quarterly Sales of Rose wine

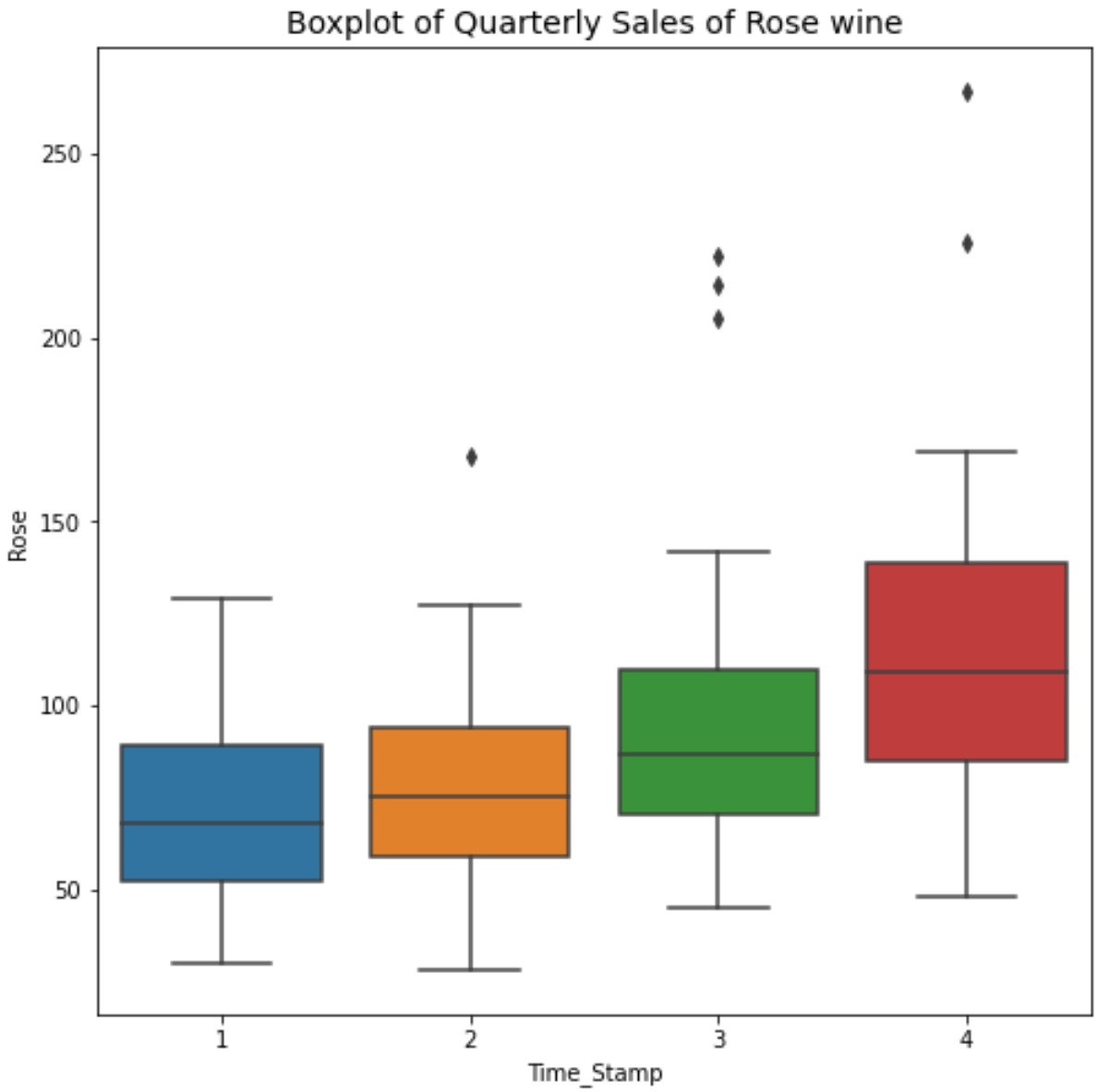

Figure 8:Rose-Quarterly-Boxplot

There is an increasing(upward) trend which we can see from the median values of individual boxplot with subsequent quarters.

Yearly Sales

Time_Stamp	Rose
1980	146.500000
1981	148.333333
1982	112.333333
1983	110.333333
1984	106.666667
1985	98.583333
1986	88.583333
1987	88.333333
1988	89.416667
1989	86.500000
1990	78.750000
1991	69.166667
1992	55.250000
1993	49.500000
1994	48.166667
1995	42.285714

Table 5:Rose-Yearly sales

This table gives the mean value of the units of wine sold for each year. There is a decreasing trend in the sales yearwise.

Barplot of Yearly Sales Sum of Rose wine

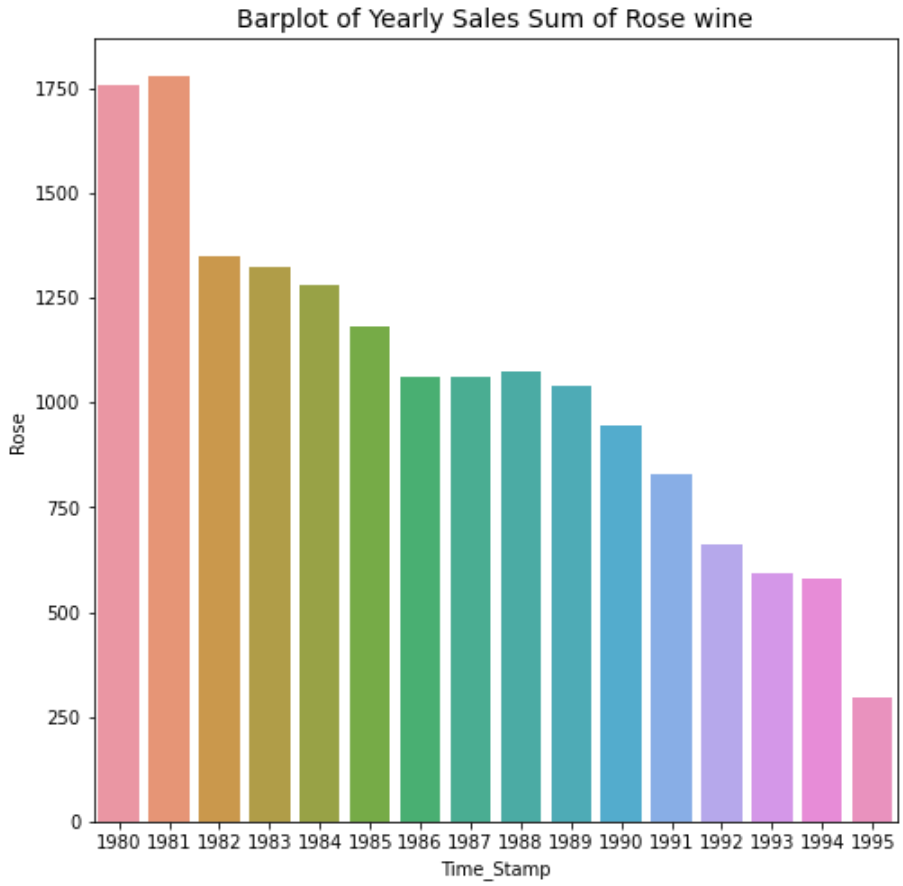

Figure 9:Rose-Barplot-Yearly

1981 has the highest sales amongst all years. 1995 has the lowest sales amongst all years.

Boxplot of Yearly Sales of Rose wine

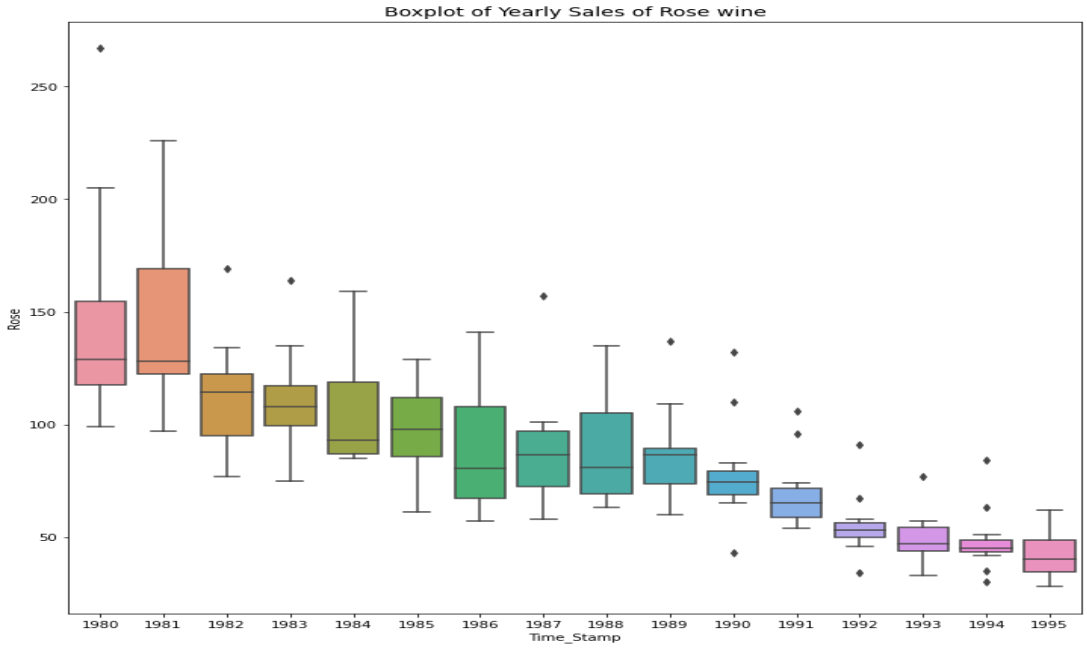

Figure 10:Rose-Boxplot-Yearly

There is a decreasing trend year-wise as shown by the medians of boxplots. The maximum sale ever happened is in 1980 which is above 250 units.

Decomposition

Additive Decomposition of Rose wine

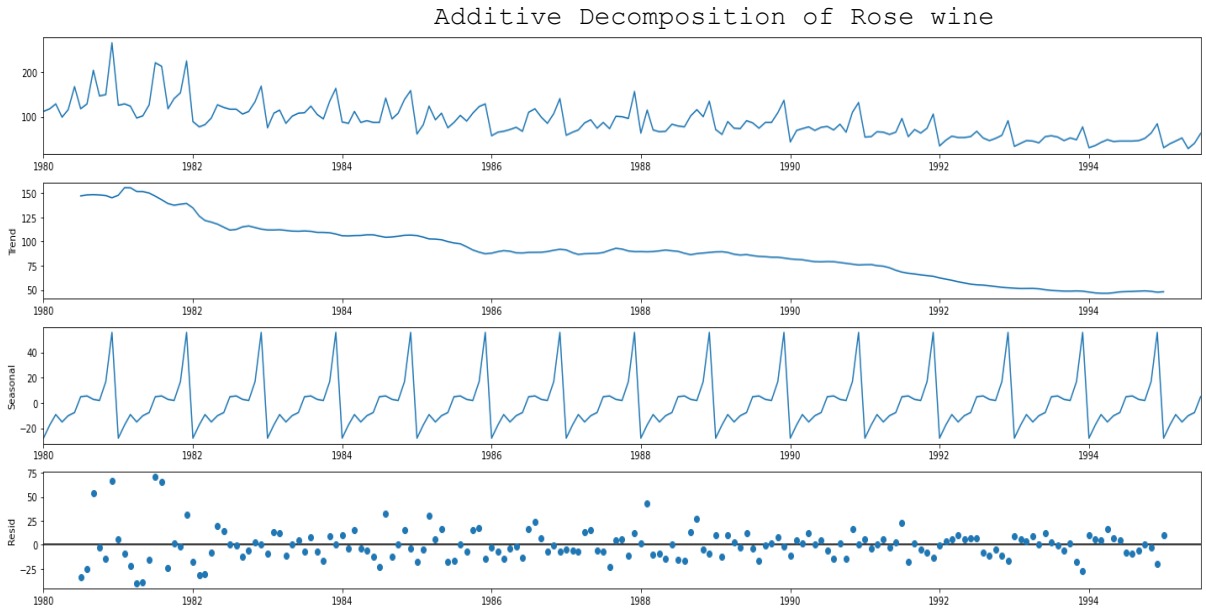

Figure 11:Rose Add Decompostion

Residuals have a pattern and it is around 0. So additive decomposition is suitable for the data.

The trend,seasonality and residual for 12 months

```
Additive Trend
 Time_Stamp
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
```

```
Additive Seasonality
 Time_Stamp
1980-01-01   -27.903092
1980-02-01   -17.431663
1980-03-01    -9.279878
1980-04-01   -15.092378
1980-05-01   -10.190592
1980-06-01    -7.672735
1980-07-01     4.880241
1980-08-01     5.460797
1980-09-01     2.780241
1980-10-01     1.877464
1980-11-01    16.852464
1980-12-01    55.719130
Name: seasonal, dtype: float64
```

```
Additive Residual
 Time_Stamp
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   -33.963575
1980-08-01   -24.585797
1980-09-01    53.844759
1980-10-01    -2.960797
1980-11-01   -14.269130
1980-12-01    66.155870
Name: resid, dtype: float64
```

Multiplicative Decomposition of Rose wine

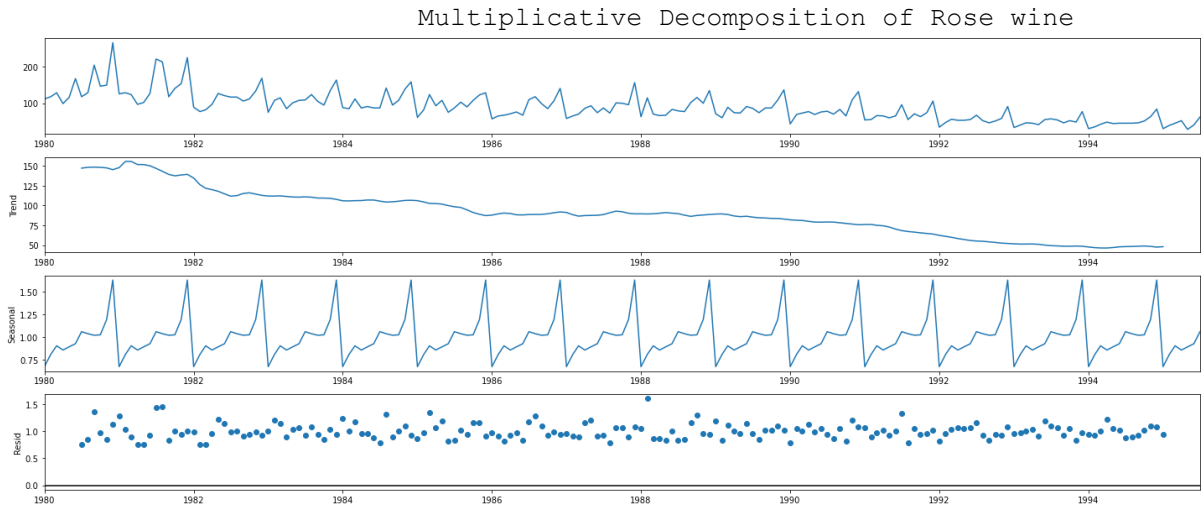

Figure 12:Rose -Multipl Decomp

Residuals have a pattern and it is around 1. So additive decomposition is not suitable for the data.

The trend ,seasonality and residuals for first 12 months is.

```
Multiplicative Trend
 Time_Stamp
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
```

```
Multiplicative Seasonality
 Time_Stamp
1980-01-01    0.670182
1980-02-01    0.806224
1980-03-01    0.901278
1980-04-01    0.854154
1980-05-01    0.889531
1980-06-01    0.924099
1980-07-01    1.057682
1980-08-01    1.035066
1980-09-01    1.017753
1980-10-01    1.022688
1980-11-01    1.192494
1980-12-01    1.628848
Name: seasonal, dtype: float64
```

```
Multiplicative Residual
 Time_Stamp
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    0.758514
1980-08-01    0.841382
1980-09-01    1.357534
1980-10-01    0.970661
1980-11-01    0.853274
1980-12-01    1.129506
Name: resid, dtype: float64
```

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

The first 5 and last 5 rows of train data shows that it starts from January 1980 and ends at December 1990.

Time_Stamp	Rose
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

Time_Stamp	Rose
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

Table 6:Rose -Train Data

The first 5 and last 5 rows of test data shows that it starts from January 1991 and ends at July 1995.

Time_Stamp	Rose
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

Time_Stamp	Rose
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

Table 7:Rose Test Data

1.4. 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.

Simple Exponential Smoothing Auto Fit Model

Simple Exponential Smoothing(SES) 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.

A SES model is built on train data with initialization_method value as estimated and the following parameters with values.

initialization_method - Method for initialize the recursions.

Use_brute=True -> Search for good starting values using a brute force (grid) optimizer

Optimized=True -> Estimate model parameters by maximizing the log-likelihood.

```
{'smoothing_level': 0.09874983698117956,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 134.38702481818487,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

The model built is used to forecast for next 55 months which is the length of test data.

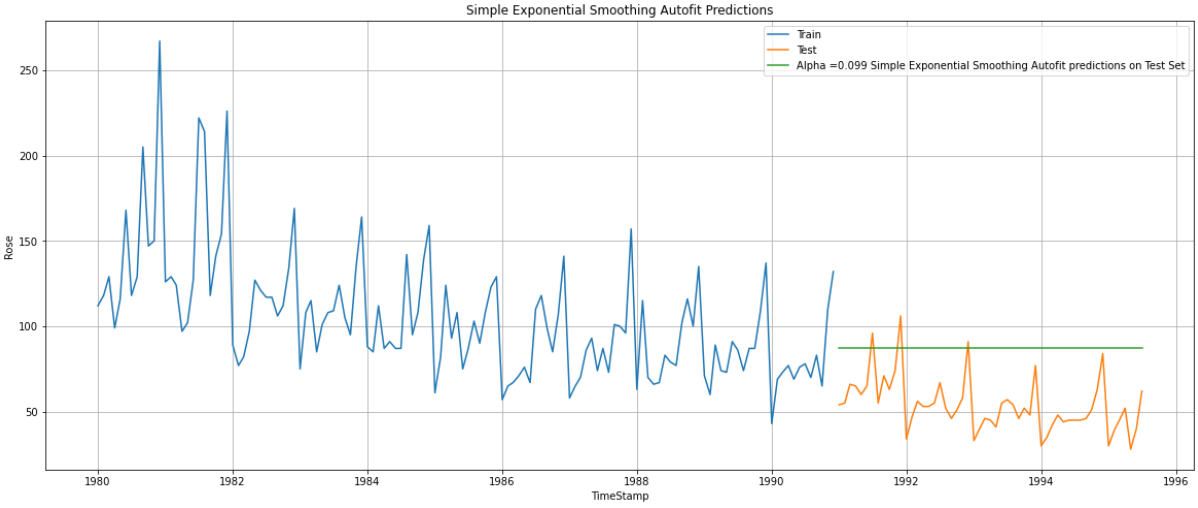

Figure 13:Rose SES

RMSE is calculated on test data.

SES Autofit RMSE: 36.82

Double Exponential Smoothing - Holt Autofit Model

Double exponential smoothing(DES) employs a level component and a trend co mponent at each period

A DES model is built on train data with initialization_method value as estimated , use_brute=True and Optimized=True. The other parameters and the values are:

```
{'smoothing_level': 1.4901161193847656e-08, 'smoothing_trend': 1.6610391146
660035e-10, 'smoothing_seasonal': nan, 'damping_trend': nan, 'initial_level
': 137.81553690867275, 'initial_trend': -0.4943781897068274, 'initial_seaso
ns': array([], dtype=float64), 'use_boxcox': False, 'lamda': None, 'remove_
bias': False}
```

The model built is used to forecast for next 55 months which is the length of test data.

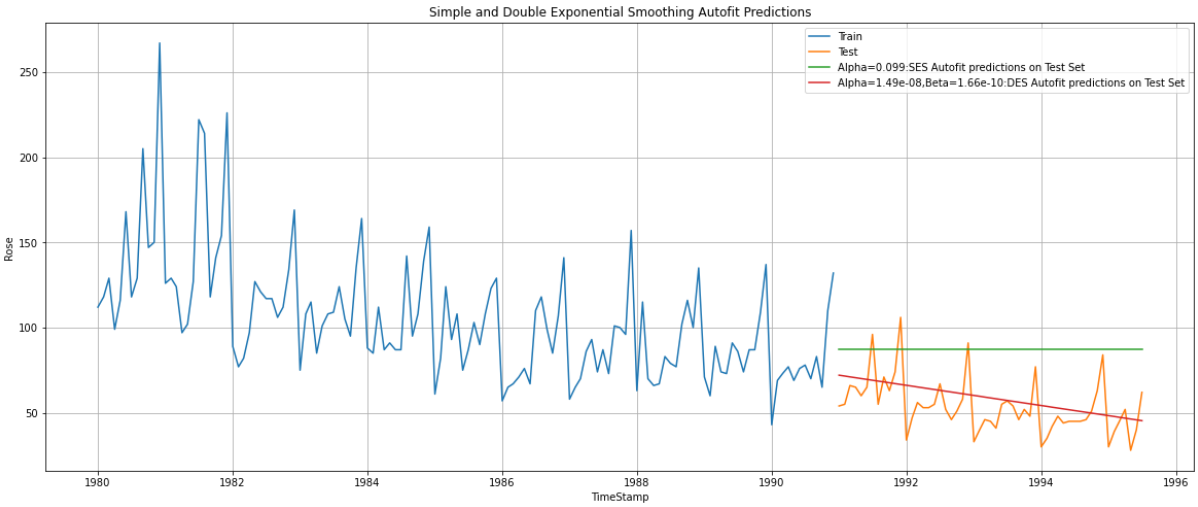

Figure 14:Rose-DES

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

RMSE is calculated on test data.

DES Autofit RMSE: 15.28

Triple Exponential Smoothing - ETS(A, A, A) - Holt Winter's linear method with additive errors Autofit Model

Triple exponential smoothing(TES) 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

In this model both trend and seasonality is chosen as additive. A DES model is built on train data with initialization_method value as estimated , use_brute=True and Optimized=True. The other parameters and the values are:

```
{'smoothing_level': 0.08954054664605082, 'smoothing_trend': 0.0002400108693
915795, 'smoothing_seasonal': 0.003466872515750747, 'damping_trend': nan, '
initial_level': 146.5570157826235, 'initial_trend': -0.547196983509005, 'in
itial_seasons': array([-31.17478463, -18.74839869, -10.76961776, -21.367410
17,
       -12.63775539,  -7.27430333,   2.61279801,   8.69603625,
         4.79381122,   2.96110122,  21.05738849,  63.18279918]), 'use_boxco
x': False, 'lamda': None, 'remove_bias': False}
```

The model built is used to forecast for next 55 months which is the length of test data.

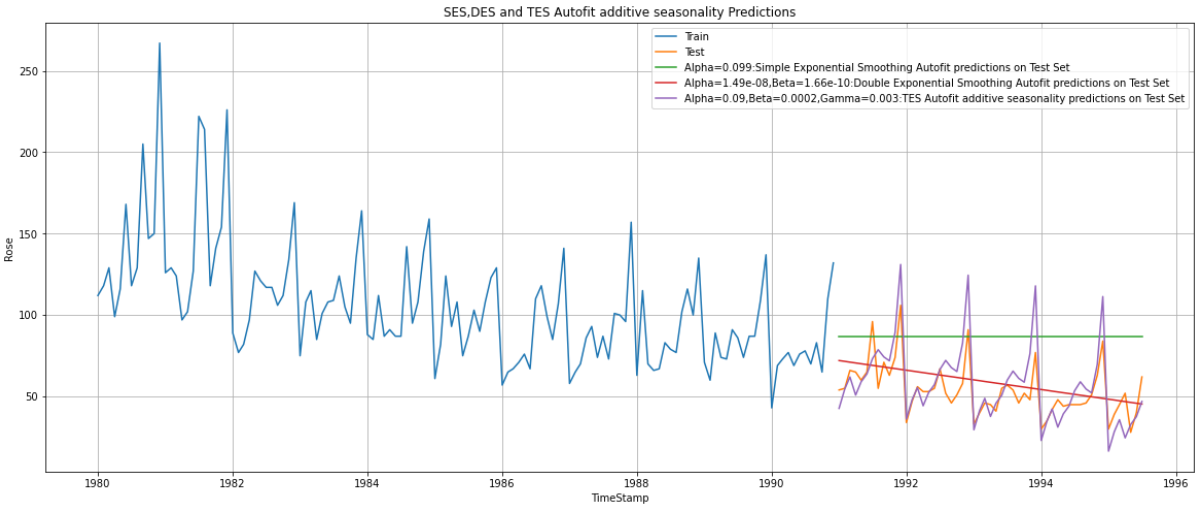

Figure 15:Rose TES add seas

RMSE is calculated on test data.

TES_A Autofit RMSE: 14.26

Triple Exponential Smoothing - ETS(A, A, M) - Holt Winter's linear method with multiplicative errors

In this model trend is additive and seasonality is chosen as multiplicative. A DES model is built on train data with initialization_method value as estimated , use_brute=True and Optimized=True. The other parameters and the values are:

The model built is used to forecast for next 55 months which is the length of test data.

```
{'smoothing_level': 0.0715106306609405, 'smoothing_trend': 0.04529179757535
142, 'smoothing_seasonal': 7.244325029450242e-05, 'damping_trend': nan, 'in
itial_level': 130.40839142502193, 'initial_trend': -0.77985743179386, 'init
ial_seasons': array([0.86218996, 0.977675  , 1.0687727 , 0.93403881, 1.0506
25  ,
       1.14410977, 1.25836944, 1.33937772, 1.26778766, 1.24131254,
       1.44724625, 1.99553681]), 'use_boxcox': False, 'lamda': None, 'remov
e_bias': False}
```

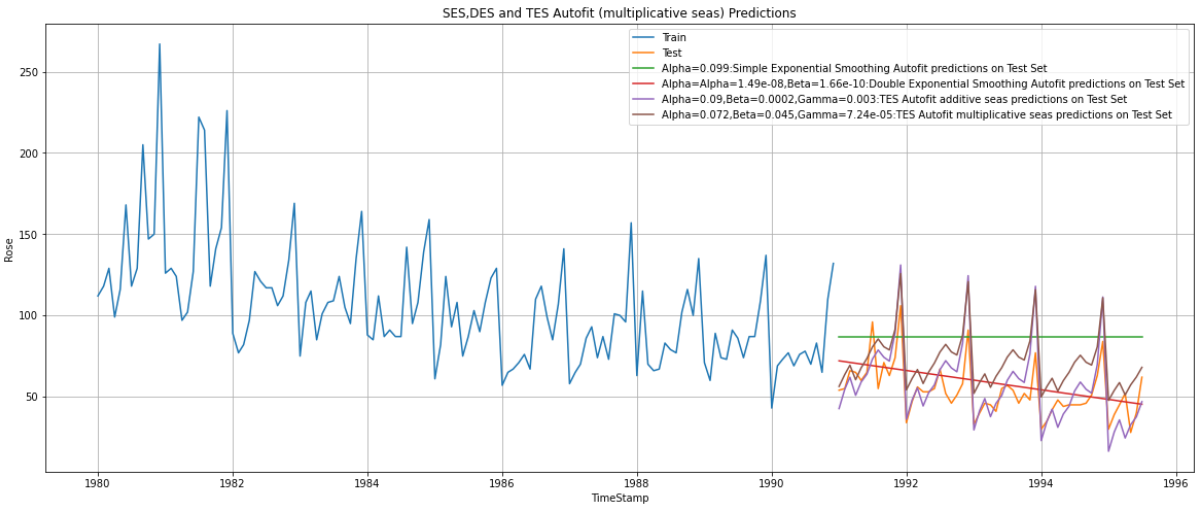

Figure 16:Rose TES multipl seas

RMSE is calculated on test data.

TES_m Autofit RMSE: 20.18

Iterative Method for Simple Exponential Smoothing

A SES model is built on train data with initialization_method value as estimated , use_brute=True and Optimized=False. The value of smoothing_level(alpha) is taken from 0.1 to 1 and its corresponding RMSE value is calculated as shown below.

	Alpha Values	Test RMSE
0	0.1	36.848684
1	0.2	41.382452
2	0.3	47.525251
3	0.4	53.787686
4	0.5	59.661932
5	0.6	64.991324
6	0.7	69.718108
7	0.8	73.793865
8	0.9	77.159094
9	1.0	79.738550

We can see that alpha as 0.1 as the least RMSE.

Plot the prediction of the model with smoothing level value as 0.1.

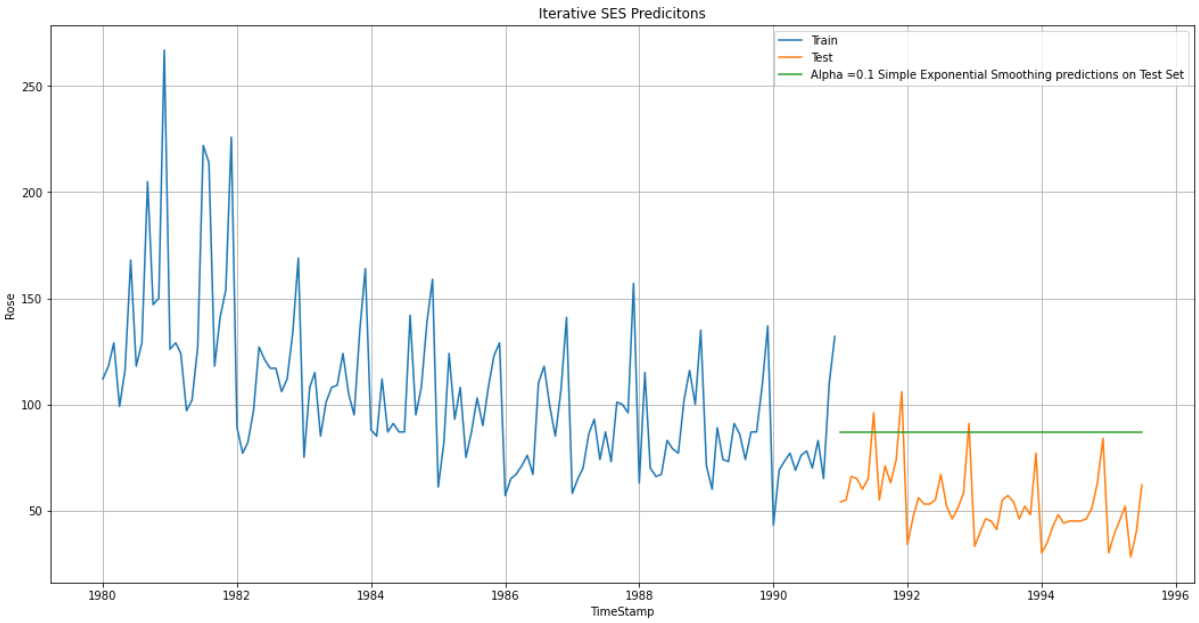

Figure 17:Rose Iterative SES

The RMSE of the iterative SES is 36.85

Iterative Method for Double Exponential Smoothing

A DES model is built on train data with initialization_method value as estimated , use_brute=True and Optimized=False. The values of smoothing_level(alpha) and smoothing_trend(beta) is taken from 0.1 to 1 and its corresponding RMSE value is calculated .The first 10 rows with least RMSE is shown below.

	Alpha Values	Beta Values	Test RMSE
0	0.1	0.1	36.900871
1	0.1	0.2	48.657789
10	0.2	0.1	65.754759
2	0.1	0.3	78.150329
20	0.3	0.1	98.676734
3	0.1	0.4	99.681595
11	0.2	0.2	114.058193
4	0.1	0.5	124.206311
30	0.4	0.1	129.002112
40	0.5	0.1	155.382399

The model with smoothing level as 0.1 and smoothing trend as 0.1 has the least RMSE value 36.9.

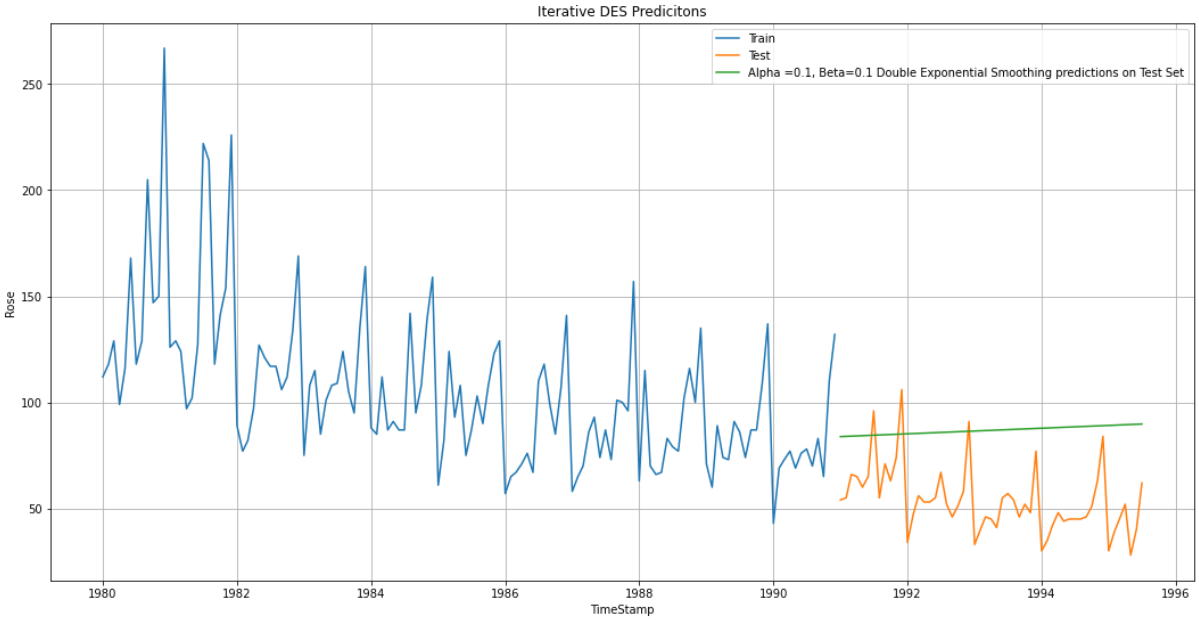

Figure 18:Rose Iter DES

The RMSE of Iterative DES is 36.9.

Iterative Method - Triple Exponential Smoothing - ETS(A, A, A)

A TES model with trend and seasonality as additive is built on train data with initialization_method value as estimated , use_brute=True and Optimized=False. The values of smoothing_level(alpha) ,smoothing_trend(beta) and smoothing_seasonal(gamma)is taken from 0.1 to 1 and its corresponding RMSE value is calculated. First 5 rows of the data which shows the 5 least RMSE ones are shown below.

	Alpha Values	Beta Values	Gamma Values	Test RMSE
32	0.1	0.4	0.3	11.973601
22	0.1	0.3	0.3	12.031354
12	0.1	0.2	0.3	12.076021
13	0.1	0.2	0.4	12.079340
23	0.1	0.3	0.4	12.218737

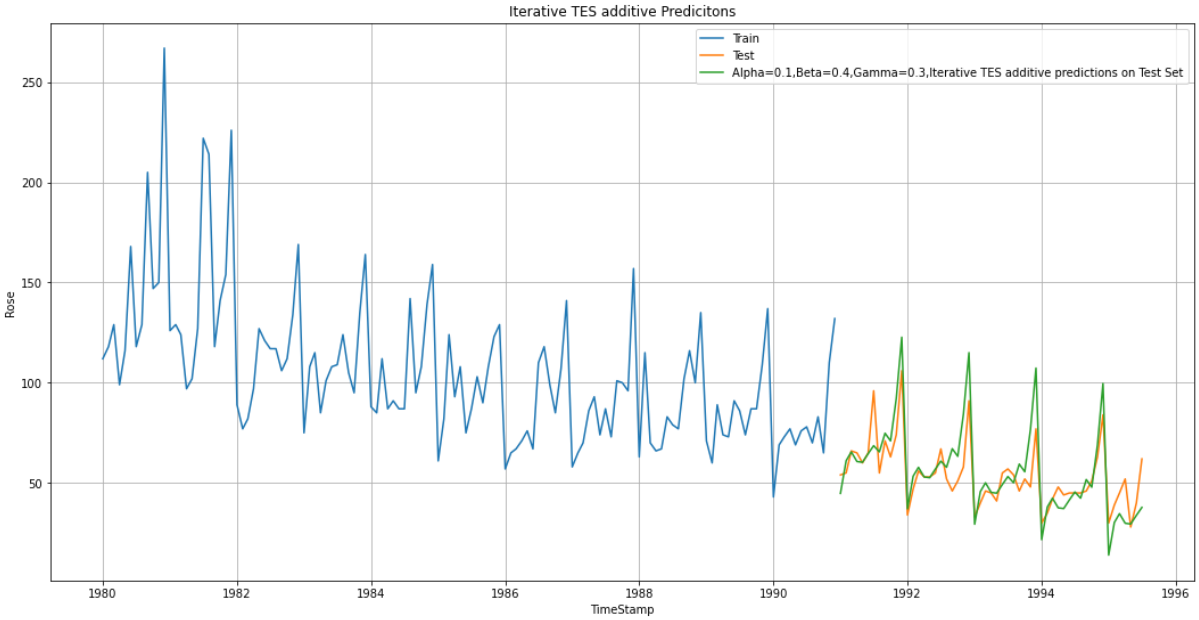

Figure 19:Rose Iter TES add seas

The RMSE of iterative TES additive is 11.97.

Iterative Method - Triple Exponential Smoothing - ETS(A, A, M)

A TES model with trend as additive and seasonality as multiplicative is built on train data with initialization_method value as estimated , use_brute=True and Optimized=False. The values of smoothing_level(alpha) ,smoothing_trend(beta) and smoothing_seasonal(gamma)is taken from 0.1 to 1 and its corresponding RMSE value is calculated. First 5 rows of the data which shows the 5 least RMSE ones are shown below.

	Alpha Values	Beta Values	Gamma Values	Test RMSE
10	0.1	0.2	0.1	9.236464
11	0.1	0.2	0.2	9.505572
151	0.2	0.6	0.2	9.561249
142	0.2	0.5	0.3	9.884070
12	0.1	0.2	0.3	9.895300

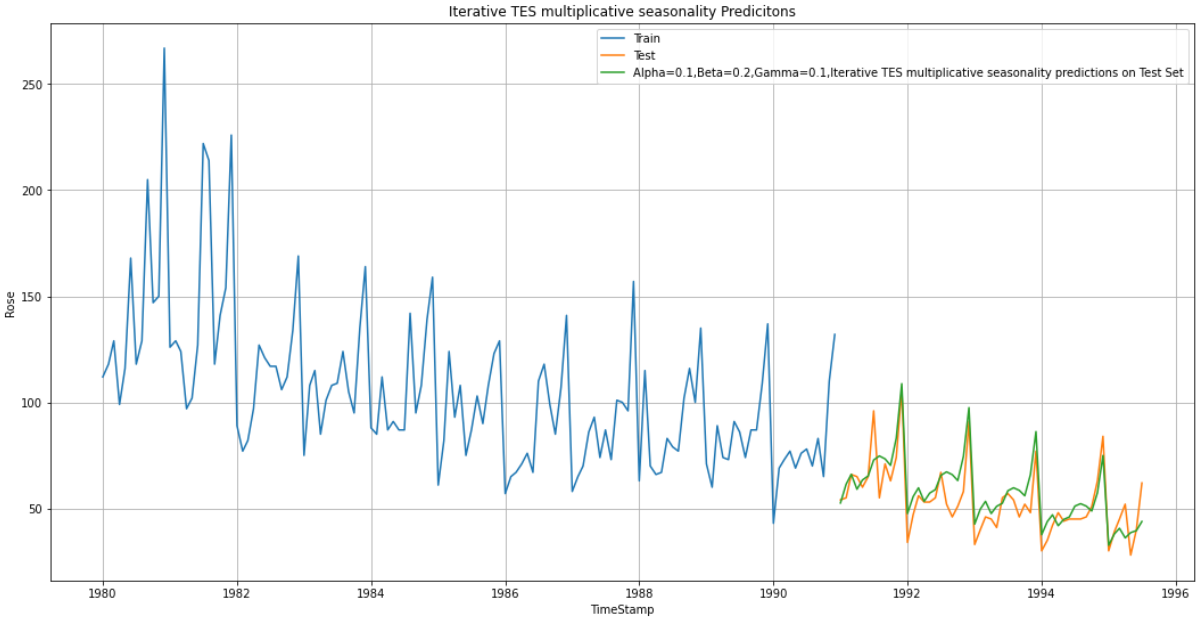

Figure 20:Rose Iter multiplicative seas

The RMSE of iterative TES multiplicative is 9.24.

Linear Regression Model

Linear regression uses the relationship between the data-points to draw a straight line through all them. This line can be used to predict future values.

The training and testing time instance is created.

```
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, 4
0, 41, 42, 43, 44, 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]
```

The instances are added to the train and test data as column ‘time’ and treated as an independent variable.

First few rows of Training Data

Time_Stamp	Rose	time
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	5

First few rows of Test Data

Time_Stamp	Rose	time
1991-01-01	54.0	133
1991-02-01	55.0	134
1991-03-01	66.0	135
1991-04-01	65.0	136
1991-05-01	60.0	137

Last few rows of Training Data

Time_Stamp	Rose	time
1990-08-01	70.0	128
1990-09-01	83.0	129
1990-10-01	65.0	130
1990-11-01	110.0	131
1990-12-01	132.0	132

Last few rows of Test Data

Time_Stamp	Rose	time
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

Rose column is treated as a dependent variable. Both the train independent and dependent variable are fitted on the Linear Regression model from sklearn library.

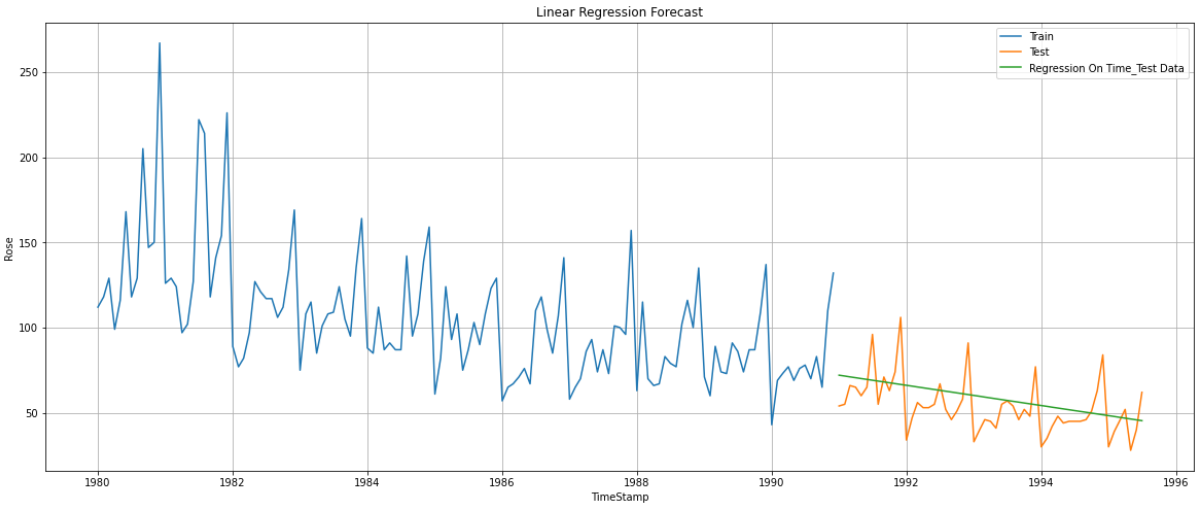

Figure 21:Rose-Linear Regr Plot

The fitted model is sued to predict on test data and the output of Rose wine sales is used to calculate RMSE along with actual Rose wine sales.
The RMSE of Linear Regression forecast on the Test Data is 15.280

Naive Forecast Model:

Naïve forecasting is the technique in which the last period's sales are used for the next period's forecast without predictions or adjusting the factors. Forecasts produced using a naïve approach are equal to the final observed value.

The last 5 rows of the train data are:

Time_Stamp	Rose
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

The last value is 132 which is treated as the forecasted value for the future time period as shown below.

First few rows of Test Data

Time_Stamp	Rose	naive
1991-01-01	54.0	132.0
1991-02-01	55.0	132.0
1991-03-01	66.0	132.0
1991-04-01	65.0	132.0
1991-05-01	60.0	132.0

Last few rows of Test Data

Time_Stamp	Rose	naive
1995-03-01	45.0	132.0
1995-04-01	52.0	132.0
1995-05-01	28.0	132.0
1995-06-01	40.0	132.0
1995-07-01	62.0	132.0

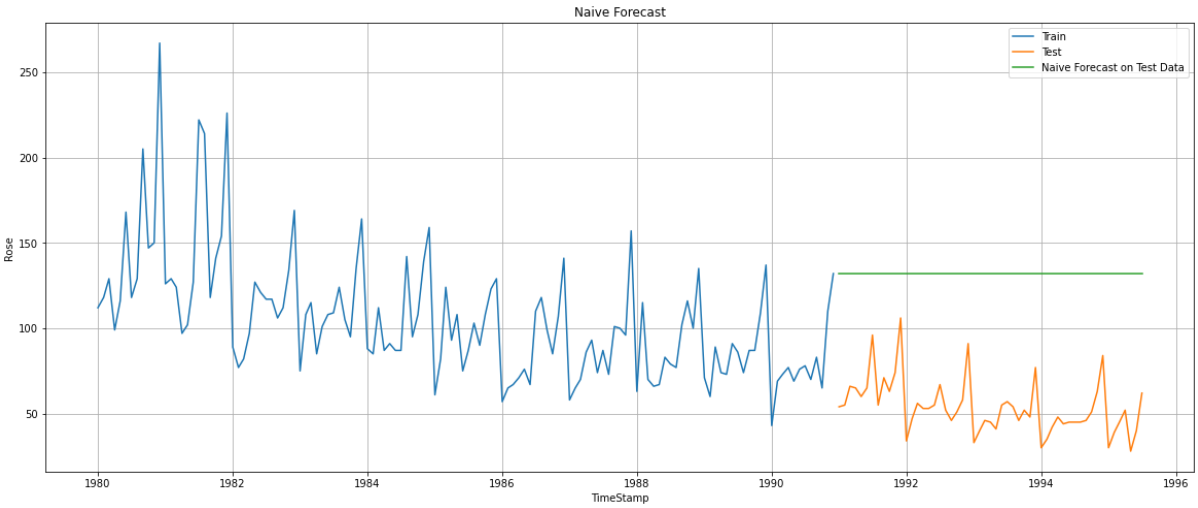

Figure 22:Rose-Naive Forecast plot

The predicted test output along with the actual test output is used for calculating RMSE.

The RMSE for NaiveModel forecast on the Test Data is 79.74

Simple Average Model

In the Simple Average model ,forecast is equal to the average of historical data.
The average of the train data is taken and added as the forecast of Test data as shown below.

Time_Stamp	Rose	mean_forecast
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

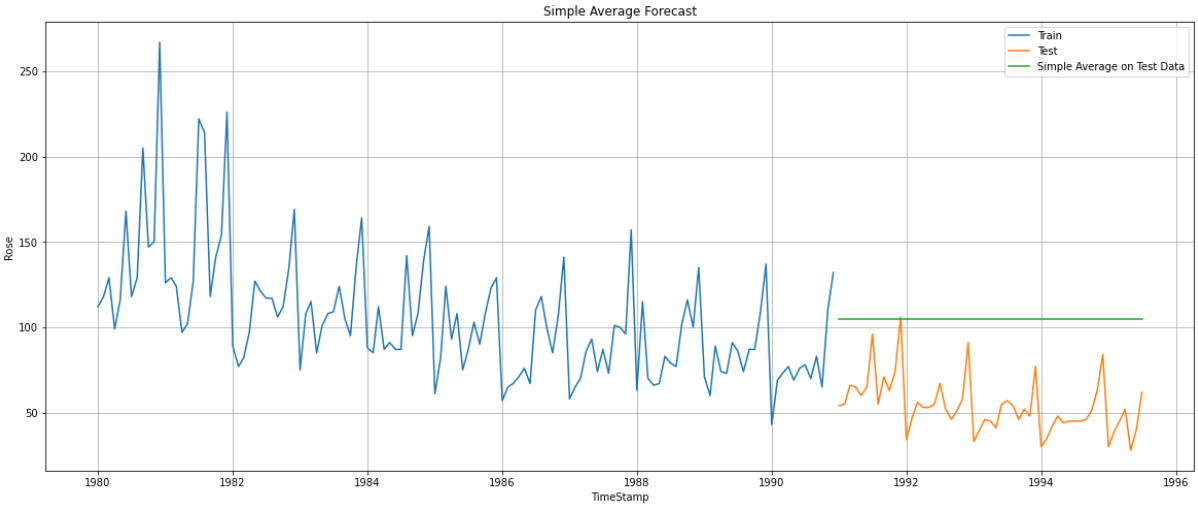

Figure 23:Rose-Simple Average Plot

The predicted test output along with the actual test output is used for calculating RMSE.

The RMSE for Simple Average forecast on the Test Data is 53.48

Moving Average Forecast Model

Moving Average Forecast Model takes an average of a set of numbers in a given range while moving the range.

Moving Average is calculated on train data for which the following values are given to rolling function.

2-takes moving average of 2 months of data

3-takes moving average of 3 months of data

6-takes moving average of 6 months of data

12-takes moving average of 12 months of data

The first 5 rows of training data is the following:

Time_Stamp	Rose	Trailing_2	Trailing_3	Trailing_6	Trailing_12
1980-01-01	112.0	NaN	NaN	NaN	NaN
1980-02-01	118.0	115.0	NaN	NaN	NaN
1980-03-01	129.0	123.5	119.666667	NaN	NaN
1980-04-01	99.0	114.0	115.333333	NaN	NaN
1980-05-01	116.0	107.5	114.666667	NaN	NaN

Table 8:Rose-MA-Train head

The last 5 rows of training data is the following:

Time_Stamp	Rose	Trailing_2	Trailing_3	Trailing_6	Trailing_12
1990-08-01	70.0	74.0	74.666667	73.833333	81.250000
1990-09-01	83.0	76.5	77.000000	75.500000	80.916667
1990-10-01	65.0	74.0	72.666667	73.500000	79.083333
1990-11-01	110.0	87.5	86.000000	80.333333	79.166667
1990-12-01	132.0	121.0	102.333333	89.666667	78.750000

Table 9:Rose-MA-train tail

The value 121 is taken as the forecasted value for the test for 2 point Moving Average.

The value 102 is taken as the forecasted value for the test for 3 point Moving Average.(102.333333 rounded off)

The value 90 is taken as the forecasted value for the test for 6 point Moving Average. (89.666667 rounded off)

The value 79 is taken as the forecasted value for the test for 12 point Moving Average. (78.750000 rounded off)

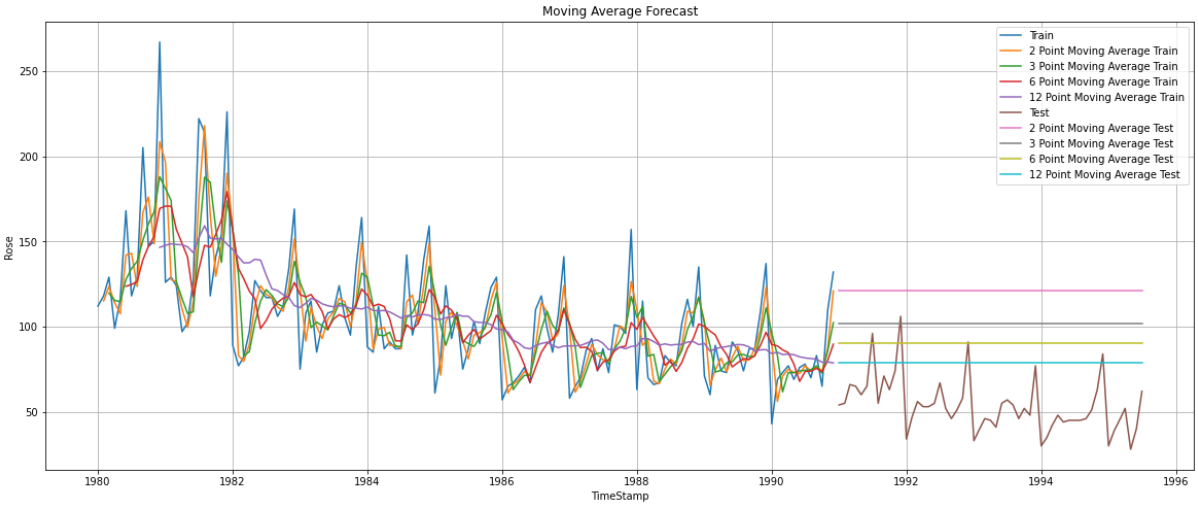

Figure 24:Rose-Moving Average plot

The RMSE for 2 point Moving Average forecast on the Test Data is 68.99
The RMSE for 3 point Moving Average forecast on the Test Data is 50.68
The RMSE for 6 point Moving Average forecast on the Test Data is 39.45
The RMSE for 12 point Moving Average forecast on the Test Data is 29.7

1.5. Check for the stationarity of the data on which the model is bei ng built on using appropriate statistical tests and also mention the hypothesis for the statistical test. If the data is found to be non-stat ionary, take appropriate steps to make it stationary. Check the new data for stationarity and comment. Note: Stationarity should be checked at alpha = 0.05.

Augmented Dickey Fuller test (ADF Test) is a common statistical test used to test whether a given Time series is stationary or not. It is one of the most commonly used statistical test when it comes to analyzing the stationary of a series.
The hypothesis for the statistical test is:

H0-Null Hypothesis: Time series is non-stationary

H1-Alternate Hypothesis: Time series is stationary

The ADF Test is conducted on the train data

```
Results of Augmented Dickey-Fuller Test:
Test Statistic                  -2.164250
p-value                          0.219476
#Lags Used                      13.000000
Number of Observations Used    118.000000
Critical Value (1%)             -3.487022
Critical Value (5%)             -2.886363
Critical Value (10%)            -2.580009
dtype: float64
```

The p-value obtained by the test should be less than the significance level (say 0.05) to reject the Null hypothesis or it fails to reject the Null hypothesis.

p value obtained from the ADF test is 0.219 which is greater than 0.05 . Hence we fail to reject the Null Hypothesis and so we can say that data is non-stationary.

To convert the data into a stationary one, the difference of a Dataframe value with the value in the previous row is taken and remove missing values. The ADF Test is taken again on the modified train data.

```
Results of Augmented Dickey-Fuller Test:
Test Statistic                -6.592372e+00
p-value                        7.061944e-09
#Lags Used                     1.200000e+01
Number of Observations Used    1.180000e+02
Critical Value (1%)           -3.487022e+00
Critical Value (5%)           -2.886363e+00
Critical Value (10%)          -2.580009e+00
dtype: float64
```

After modifying, the p-value 7.061943750942e-09 obtained by the test is less than 0.05. Now the data has been converted into a stationary one.

1.6. 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.

ARIMA Automated Train Model

Auto Regressive Integrated Moving Average (ARIMA) models are applied on time series data when the current value is assumed to be correlated to past values and past prediction errors. Therefore, these models are used in defining current value as a linear combination of past values and past prediction errors. Here, we have defined a few terms that would be useful in understanding ARIMA models in detail. ARIMA models can only be applied only on stationary time series data.

The Akaike information criterion (AIC) is an estimator of out-of-sample prediction error and thereby relative quality of statistical models for a given set of data. The least the AIC the better the model is .

The p,q value is taken from 0 to 3.'d' is taken as 1.

	param	AIC
6	(1, 1, 2)	1278.055800
10	(2, 1, 2)	1279.366570
15	(3, 1, 3)	1281.216982
2	(0, 1, 2)	1281.561210
7	(1, 1, 3)	1284.818744
3	(0, 1, 3)	1285.199785
11	(2, 1, 3)	1288.617424
13	(3, 1, 1)	1297.035418
14	(3, 1, 2)	1297.622101
9	(2, 1, 1)	1297.860333
5	(1, 1, 1)	1315.793087
1	(0, 1, 1)	1330.864202
12	(3, 1, 0)	1348.180931
8	(2, 1, 0)	1364.690255
4	(1, 1, 0)	1407.410114
0	(0, 1, 0)	1453.686410

Table 10:Rose-Automated ARIMA AIC

The model has the parameter 'order' which has its values in the form of (p,d,q) where

p: Trend autoregression order.

d: Trend difference order.

q: Trend moving average order.

The p value is taken as 1, q as 2 and d as 1.(1,1,2) as it has the least AIC value 1278.06.

ARIMA is built using stationary data after dropping its NA values since it reduces the AIC value .

enforce_stationarity→Whether or not to transform the AR parameters to enforce stationarity in the autoregressive component of the model.

enforce_invertibility→Whether or not to transform the MA parameters to enforce invertibility in the moving average component of the model.

enforce_stationarity and enforce_ invertibility is given as false.

The ARIMA model is built with those values and RMSE is calculated on test data.

The Summary of ARIMA model is

SARIMAX Results

Dep. Variable:	Rose	No. Observations:	131
Model:	ARIMA(1, 1, 2)	Log Likelihood	-635.028
Date:	Sun, 05 Jun 2022	AIC	1278.056
Time:	20:07:51	BIC	1289.526
Sample:	02-01-1980	HQIC	1282.716
	- 12-01-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.2083	0.070	2.980	0.003	0.071	0.345
ma.L1	-1.9937	0.701	-2.844	0.004	-3.368	-0.620
ma.L2	0.9972	0.702	1.420	0.156	-0.379	2.374
sigma2	928.0469	628.591	1.476	0.140	-303.969	2160.063

Ljung-Box (L1) (Q):	0.02	Jarque-Bera (JB):	3.06
Prob(Q):	0.90	Prob(JB):	0.22
Heteroskedasticity (H):	0.36	Skew:	0.10
Prob(H) (two-sided):	0.00	Kurtosis:	3.73

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

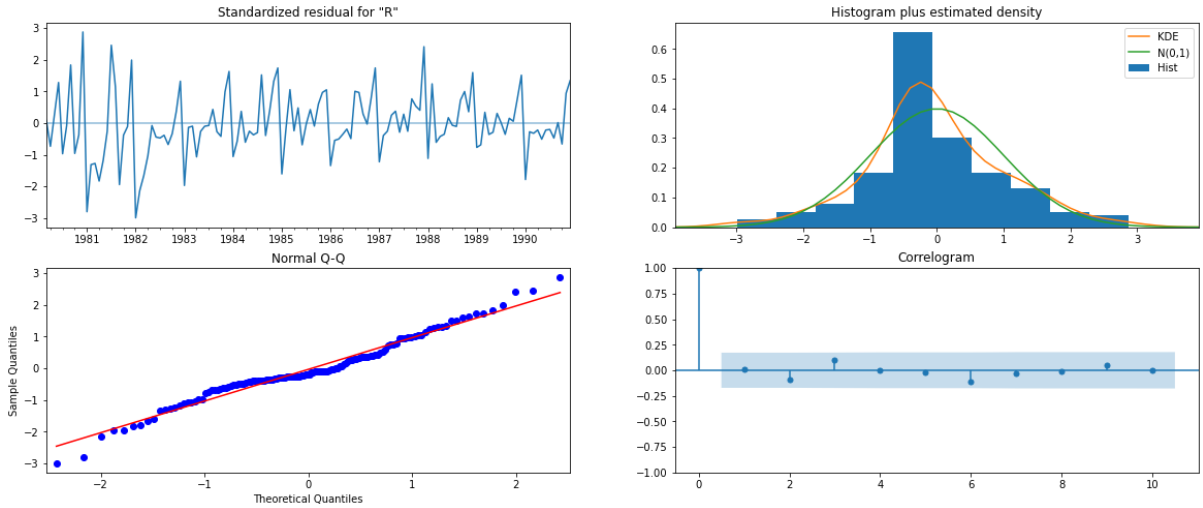

Figure 25:Rose -Automated ARIMA diagnostics

The diagnostics look good here.

RMSE of Automated ARIMA model on Test data is: 56.93

SARIMA Automated Train Model

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

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

Autocorrelation Function (ACF)

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 Autocorrelation Function (PACF)

A plot of partial auto-correlation for different values of lags is called PACF.

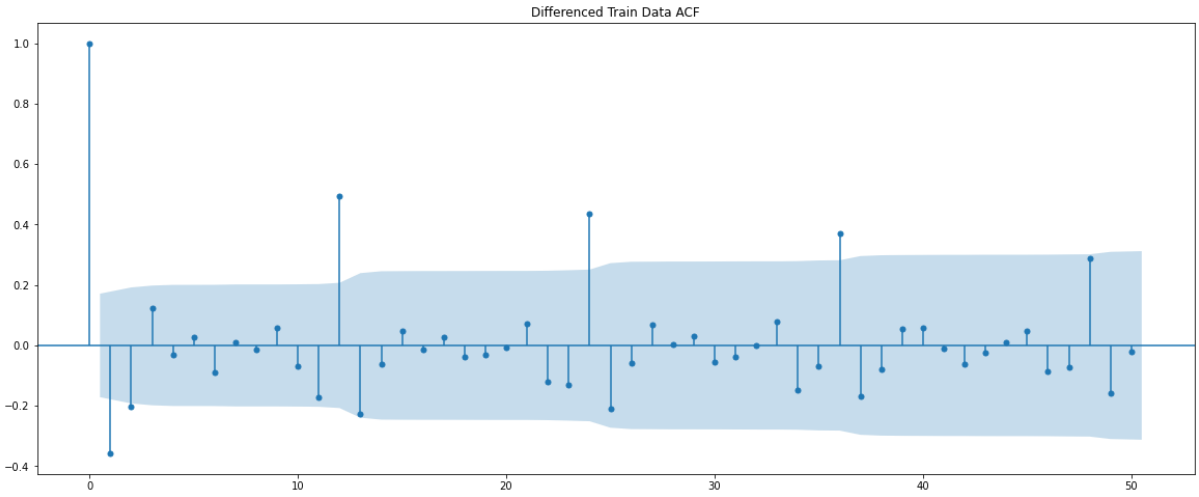

Figure 26:Rose-Auto SARIMA ACF

The model has the parameter 'seasonal order' which has its values in the form of (P, D, Q, s):

P: Seasonal autoregressive order.

D: Seasonal difference order.

Q: Seasonal moving average order.

s: The number of time steps for a single seasonal period.

's' is determined from acf plot

The P,Q value is taken from 0 to 2. D is as 0 and 1. 'd' is taken as 1. From the above ACF plot we can say that ,Seasonality after every 12th lag is visible. We will run our auto SARIMA models by setting seasonality as 12. SARIMA is built using stationary data after dropping its NA values since it reduces the AIC value.

	param	seasonal	AIC
53	(0, 1, 2)	(2, 1, 2, 12)	775.378229
107	(1, 1, 2)	(2, 1, 2, 12)	777.069145
161	(2, 1, 2)	(2, 1, 2, 12)	779.068914
41	(0, 1, 2)	(0, 1, 2, 12)	783.535250
47	(0, 1, 2)	(1, 1, 2, 12)	783.577724

Table 11:Rose-Automated SARIMA AIC

The parameters in the first row has the least AIC so its taken to build the SARIMA model. ‘p’ is taken as 0, d as 1, q as 2, P as 2, D as 1, Q as 2 and s as 12.

SARIMA is built using stationary data after dropping its NA values since it reduces the AIC value to 775.38.

enforce_stationarity→Whether or not to transform the AR parameters to enforce stationarity in the autoregressive component of the model.

enforce_invertibility→Whether or not to transform the MA parameters to enforce invertibility in the moving average component of the model.

enforce_stationarity and enforce_ invertibility is given as false.

The SARIMA model is built with those values and RMSE is calculated on test data.

The Summary of SARIMA model built is:

SARIMAX Results

Dep. Variable:	y	No. Observations:	131
Model:	SARIMAX(0, 1, 2)x(2, 1, 2, 12)	Log Likelihood	-380.689
Date:	Sun, 05 Jun 2022	AIC	775.378
Time:	21:05:31	BIC	792.954
Sample:	0	HQIC	782.469
	- 131		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-1.8916	0.054	-34.843	0.000	-1.998	-1.785
ma.L2	0.9077	0.055	16.410	0.000	0.799	1.016
ar.S.L12	0.0334	0.130	0.258	0.797	-0.221	0.288
ar.S.L24	-0.0639	0.040	-1.610	0.107	-0.142	0.014
ma.S.L12	-0.7857	0.268	-2.929	0.003	-1.311	-0.260
ma.S.L24	-0.0653	0.168	-0.389	0.697	-0.394	0.264
sigma2	212.7953	45.697	4.657	0.000	123.231	302.359

Ljung-Box (L1) (Q):	0.13	Jarque-Bera (JB):	1.60
Prob(Q):	0.72	Prob(JB):	0.45
Heteroskedasticity (H):	0.85	Skew:	0.29
Prob(H) (two-sided):	0.66	Kurtosis:	3.29

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

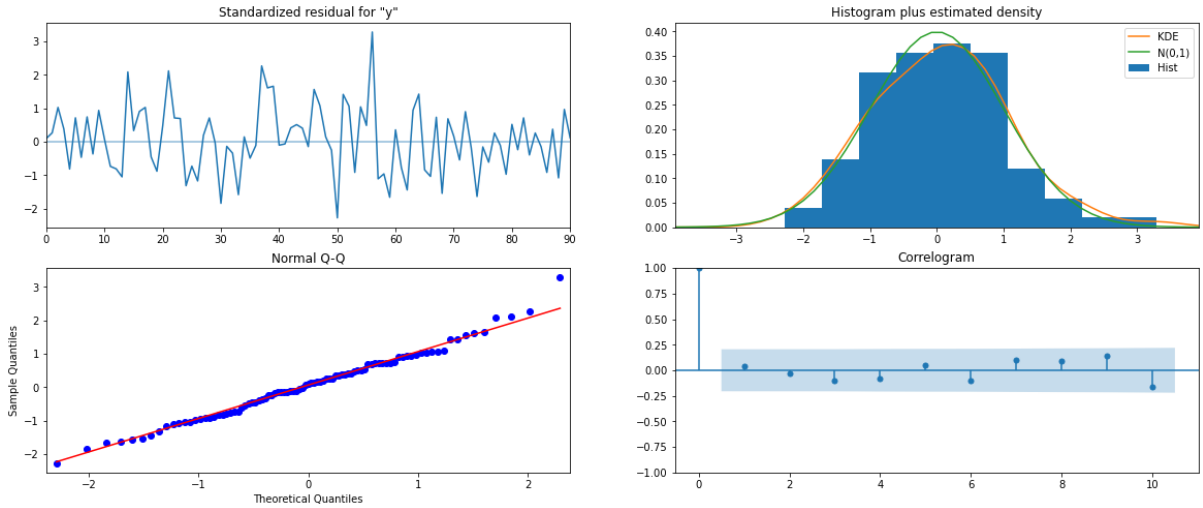

Figure 27:Rose-Automated SARIMA diagnostics

RMSE of Automated SARIMA model on Test data is: 61.44

1.7. 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.

Manual ARIMA Model

ACF is used for identifying the value of q and PACF is used for identifying the value of p.

The p value of the Augmented Dickey-Fuller Test on Train data: 0.219476 .'p' is not less than 0.05 so the data isnt stationary.

The p value of the Augmented Dickey-Fuller Test on Train data: 7.061943750942e-09. The data after first difference is stationary as the p value is less than 0.05. Thus we can consider the value of d as 1.

ACF and PACF plotted with differenced data is below:

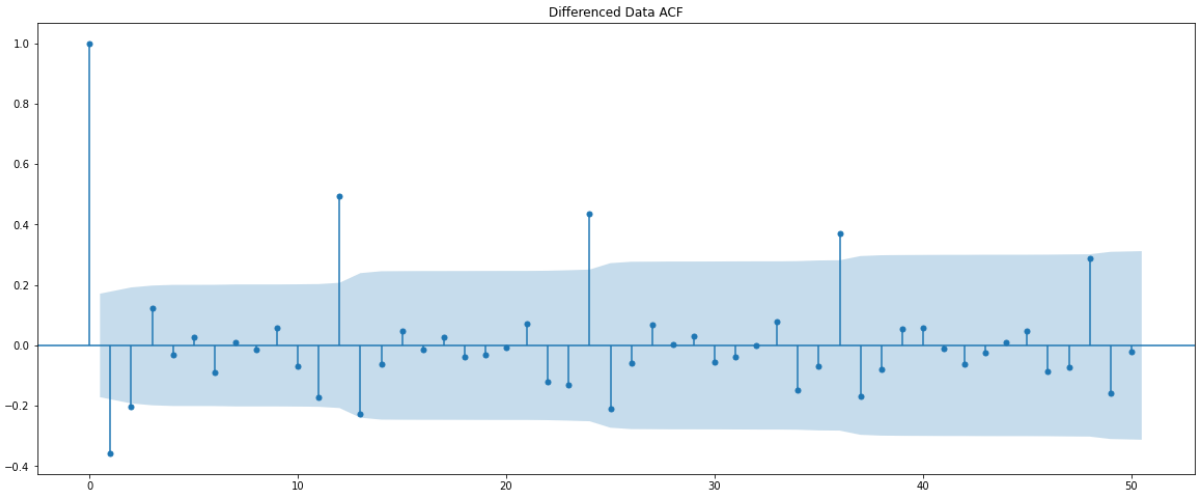

Figure 28:Rose-Manual ARIMA ACF

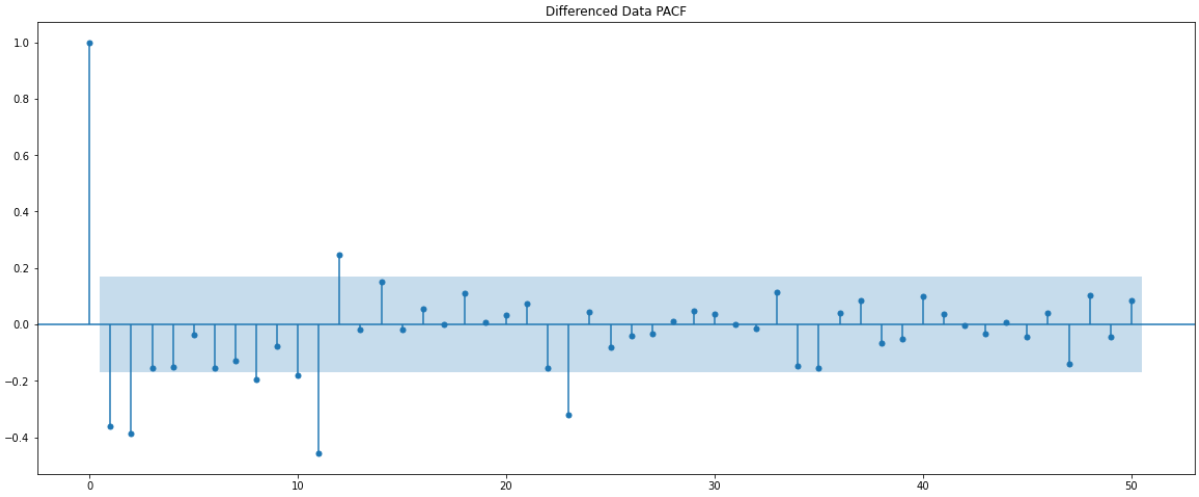

Figure 29:Rose-Manual ARIMA PACF

From the ACF and PACF models we can take the value of p and q as 2. Since difference of order 1 has been taken on the data to make it stationary ,d=1.

enforce_stationarity→Whether or not to transform the AR parameters to enforce stationarity in the autoregressive component of the model.

enforce_invertibility→Whether or not to transform the MA parameters to enforce invertibility in the moving average component of the model.

enforce_stationarity and enforce_ invertibility is given as false.

ARIMA is built using stationary data after dropping its NA values .The ARIMA model is built with those values and RMSE is calculated on test data.

The summary of Manual ARIMA model is:

SARIMAX Results

Dep. Variable:	y	No. Observations:	131
Model:	ARIMA(2, 1, 2)	Log Likelihood	-634.683
Date:	Sun, 05 Jun 2022	AIC	1279.367
Time:	21:12:39	BIC	1293.704
Sample:	0	HQIC	1285.192
	- 131		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.2191	0.069	3.172	0.002	0.084	0.354
ar.L2	-0.0740	0.084	-0.879	0.380	-0.239	0.091
ma.L1	-1.9952	1.443	-1.383	0.167	-4.823	0.833
ma.L2	0.9986	1.445	0.691	0.490	-1.834	3.831
sigma2	917.4069	1295.528	0.708	0.479	-1621.781	3456.595

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	3.00
Prob(Q):	0.96	Prob(JB):	0.22
Heteroskedasticity (H):	0.37	Skew:	0.07
Prob(H) (two-sided):	0.00	Kurtosis:	3.73

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

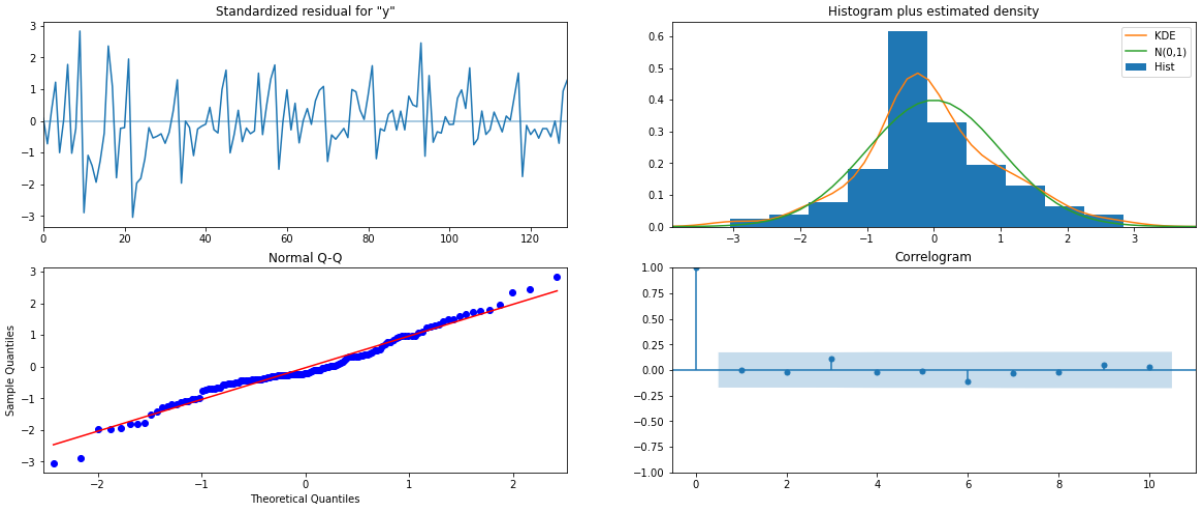

Figure 30:Rose-Manual ARIMA diagnostics

The diagnostics are good here.

RMSE of Manual ARIMA model on Test data is: 56.97

Manual SARIMA Model:

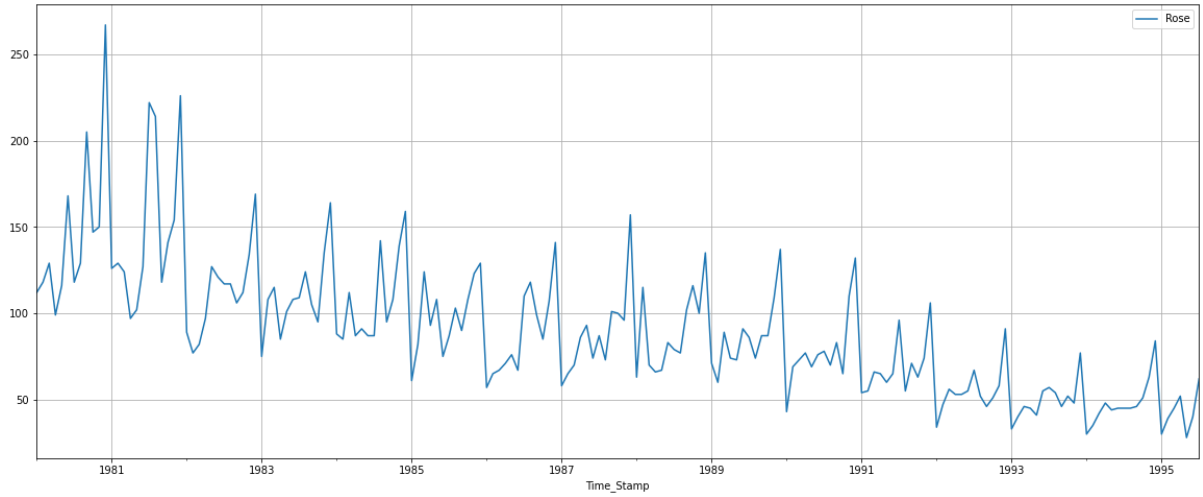

Figure 31:Rose-Manual SARIMA plot 1

We see that there is a trend and a seasonality. So, now we take a seasonal differencing and check the series.

Since the seasonality parameter is 12 we can plot the graph for the difference of order 12 with NA values dropped.

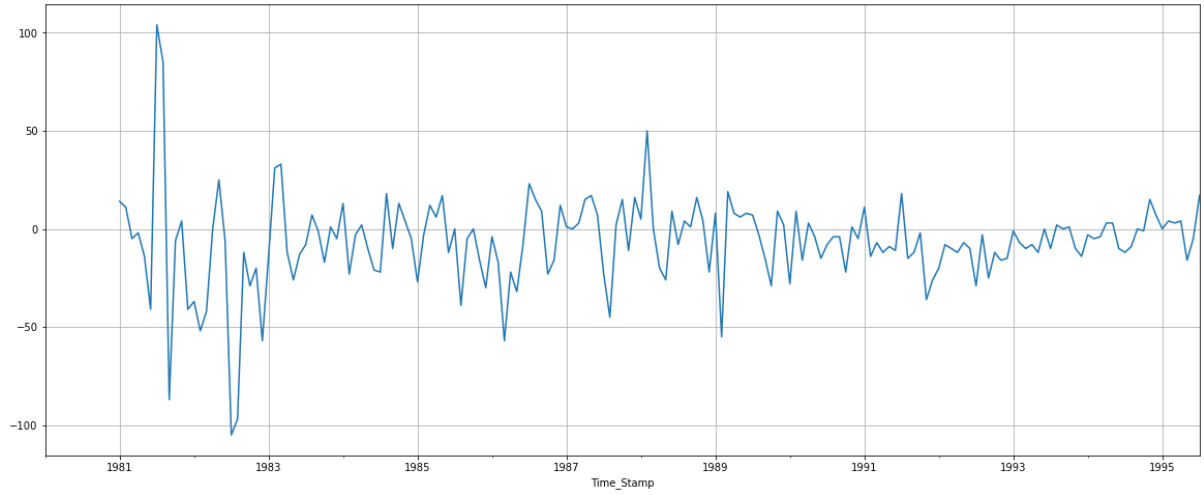

Figure 32:Rose-Manual SARIMA plot 2

As there is a slight trend in the graph we can take a differencing of first order on the seasonally differenced series .

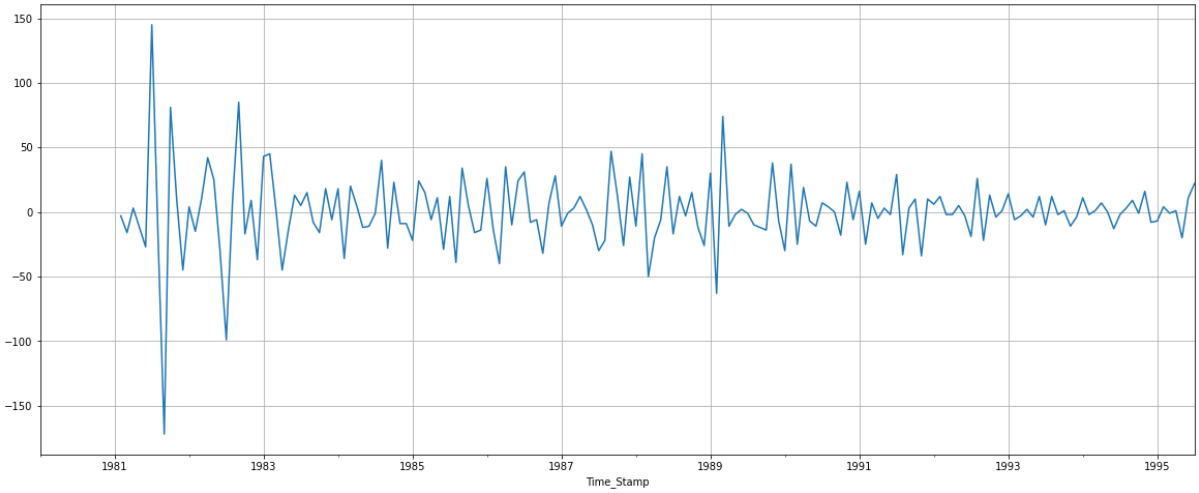

Figure 33:Rose-Manual SARIMA plot 3

Now we see that there is almost no trend present in the data. Seasonality is only present in the data.

Let us go ahead and check the stationarity of the above series before fitting the SARIMA model.

```
Results of Augmented Dickey-Fuller Test:
Test Statistic                  -3.692348
p-value                          0.004222
#Lags Used                      11.000000
Number of Observations Used    107.000000
Critical Value (1%)             -3.492996
Critical Value (5%)             -2.888955
Critical Value (10%)            -2.581393
dtype: float64
```

The series is stationary.

The first difference of seasonal differenced train data is used to plot ACF and PACF as below:

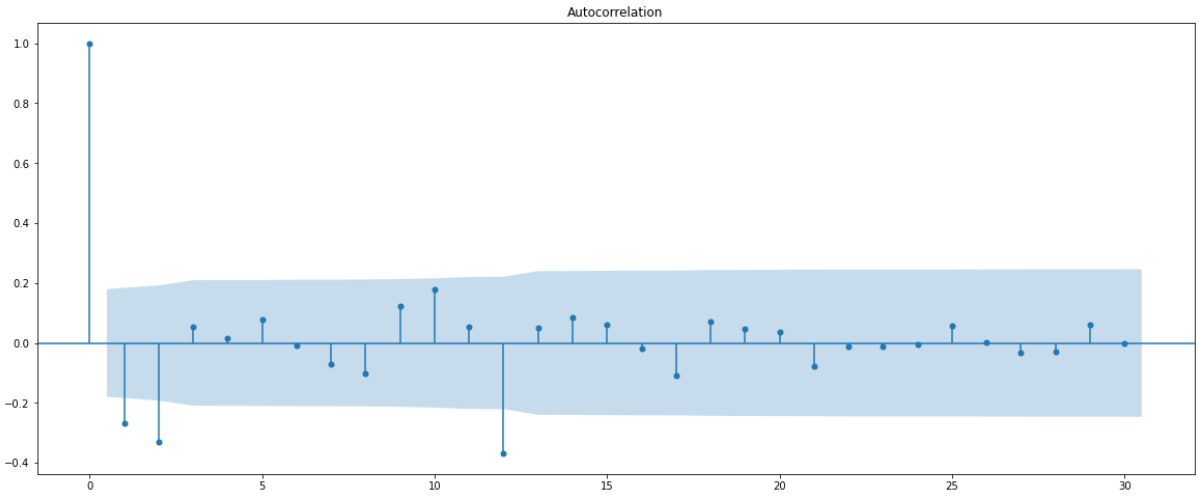

Figure 34:Rose-Manual SARIMA ACF

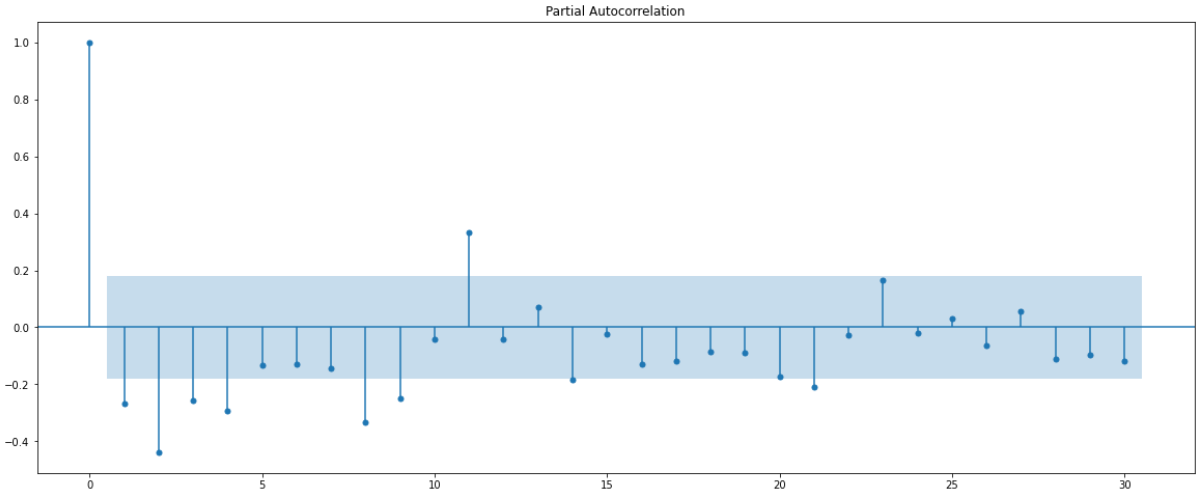

Figure 35:Rose-Manual SARIMA PACF

Here, we have taken alpha=0.05.

We are going to take the seasonal period as 12. We will keep the p(1) and q(1) 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.

By looking at the plots we see that the ACF and the PACF do not directly cut-off to 0.

Hence P=4 and Q=2. As we have taken a differencing of first order on the seasonally differenced series D is taken as 1.

'p','q','d' has the same value as the one calculated in the ARIMA model.

Order=(2,1,2) and Seasonal_order=(4,1,2,12) are the parameters to be passed to the SARIMA model along with the difference data after dropping NA values.

enforce_stationarity→Whether or not to transform the AR parameters to enforce stationarity in the autoregressive component of the model.

enforce_invertibility→Whether or not to transform the MA parameters to enforce invertibility in the moving average component of the model.

enforce_stationarity and enforce_ invertibility is given as false.

The model is built with these values and RMSE is calculated.

SARIMAX Results

Dep. Variable:	y	No. Observations:	131
Model:	SARIMAX(2, 1, 2)x(4, 1, 2, 12)	Log Likelihood	-283.397
Date:	Sun, 05 Jun 2022	AIC	588.794
Time:	21:22:38	BIC	613.209
Sample:	0	HQIC	598.468
	- 131		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.0906	0.161	-0.563	0.574	-0.406	0.225
ar.L2	-0.0851	0.160	-0.532	0.595	-0.399	0.229
ma.L1	-2.0887	2.661	-0.785	0.432	-7.304	3.127
ma.L2	1.0881	2.874	0.379	0.705	-4.545	6.721
ar.S.L12	-0.8413	0.225	-3.732	0.000	-1.283	-0.399
ar.S.L24	-0.1871	0.225	-0.830	0.406	-0.629	0.255
ar.S.L36	0.0257	0.111	0.232	0.816	-0.191	0.242
ar.S.L48	-5.569e-05	0.016	-0.003	0.997	-0.032	0.032
ma.S.L12	0.0661	0.396	0.167	0.867	-0.710	0.842
ma.S.L24	-0.5773	0.366	-1.577	0.115	-1.295	0.140
sigma2	176.7965	487.188	0.363	0.717	-778.075	1131.668

Ljung-Box (L1) (Q):	0.01	Jarque-Bera (JB):	4.58
Prob(Q):	0.92	Prob(JB):	0.10
Heteroskedasticity (H):	0.78	Skew:	0.54
Prob(H) (two-sided):	0.55	Kurtosis:	3.68

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

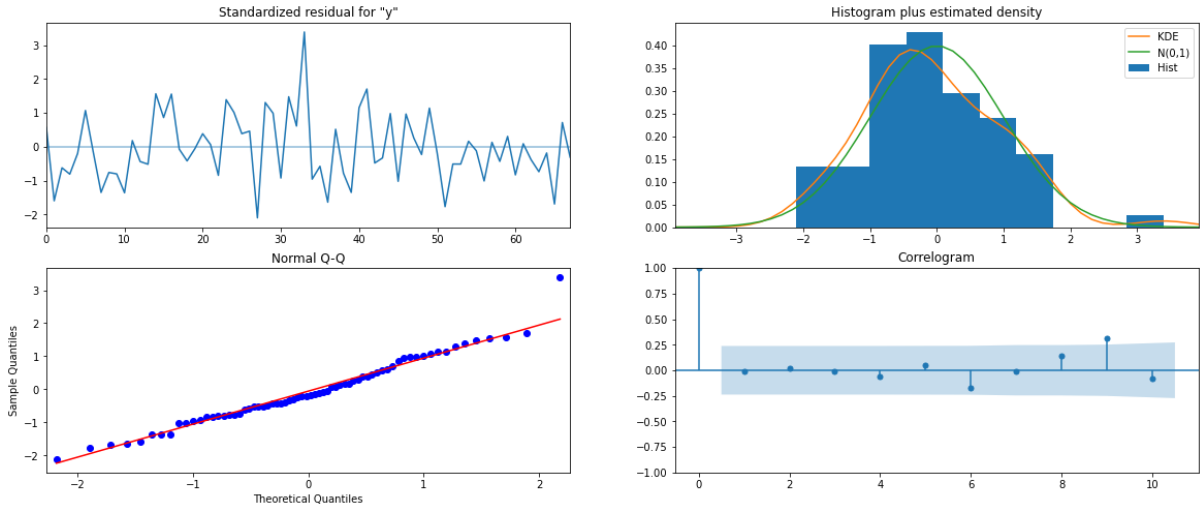

Figure 36:Rose-Manual SARIMA diagnostics

RMSE of Manual SARIMA model on Test data is: 60.24

1.8.Build a table (create a data frame) with all the models built along with their corresponding parameters and the respective RMSE values on the test data.

The model with its parameter and its corresponding RMSE value is shown below in ascending order of RMSE value from which we can see that Iterative Triple Exponential Smoothing with additive trend and Multiplicative Seasonality has the least RMSE value of 9.24. Hence it's the optimum model.

	Test RMSE
Alpha=0.1,Beta=0.2,Gamma=0.1,TES Iterative Multiplicative Seas	9.24
Alpha=0.1,Beta=0.4,Gamma=0.3,TES Iterative Additive Seas	11.97
Alpha=0.09,Beta=0.0002,Gamma=0.003:TES Autofit	14.26
Alpha=1.49e-08,Beta=1.66e-10:DES Autofit	15.28
RegressionOnTime	15.28
Alpha=0.072,Beta=0,.045,Gamma=7.24e-05 TES Autofit	20.18
12pointTrailingMovingAverage	29.70
Alpha=0.099,SES Autofit	36.82
Alpha=0.1 SES Iterative	36.85
Alpha=0.1 , Beta=0.1,DES Iterative	36.90
6pointTrailingMovingAverage	39.45
3pointTrailingMovingAverage	50.68
SimpleAverageModel	52.58
ARIMA Automated(1,1,2)	56.93
ARIMA Manual(2,1,2)	56.97
SARIMA Manual(2,1,2)(4,1,2,12)	60.24
SARIMA Automated(0,1,2)(2,1,2,12)	61.44
2pointTrailingMovingAverage	68.99
NaiveModel	79.74

Table 12:Rose-Models-RMSE

1.9.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

The optimum model Iterative Triple Exponential Smoothing with additive trend and Multiplicative Seasonality is built and forecasted for 12 months into the future.
The paramters are as below:
{'smoothing_level': 0.1, 'smoothing_trend': 0.2, 'smoothing_seasonal': 0.1, 'damping_trend': nan, 'initial_level': 145.27499999999992, 'initial_trend': 0.7643939393939481,

'initial_seasons': array([0.75572235, 0.80417408, 0.89051255, 0.75964172, 0.88067767,
0.92553586, 1.08905952, 1.13538639, 1.03030222, 0.96585715,
1.13788391, 1.62524659]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}

The forecast is shown with 95% confidence interval band.

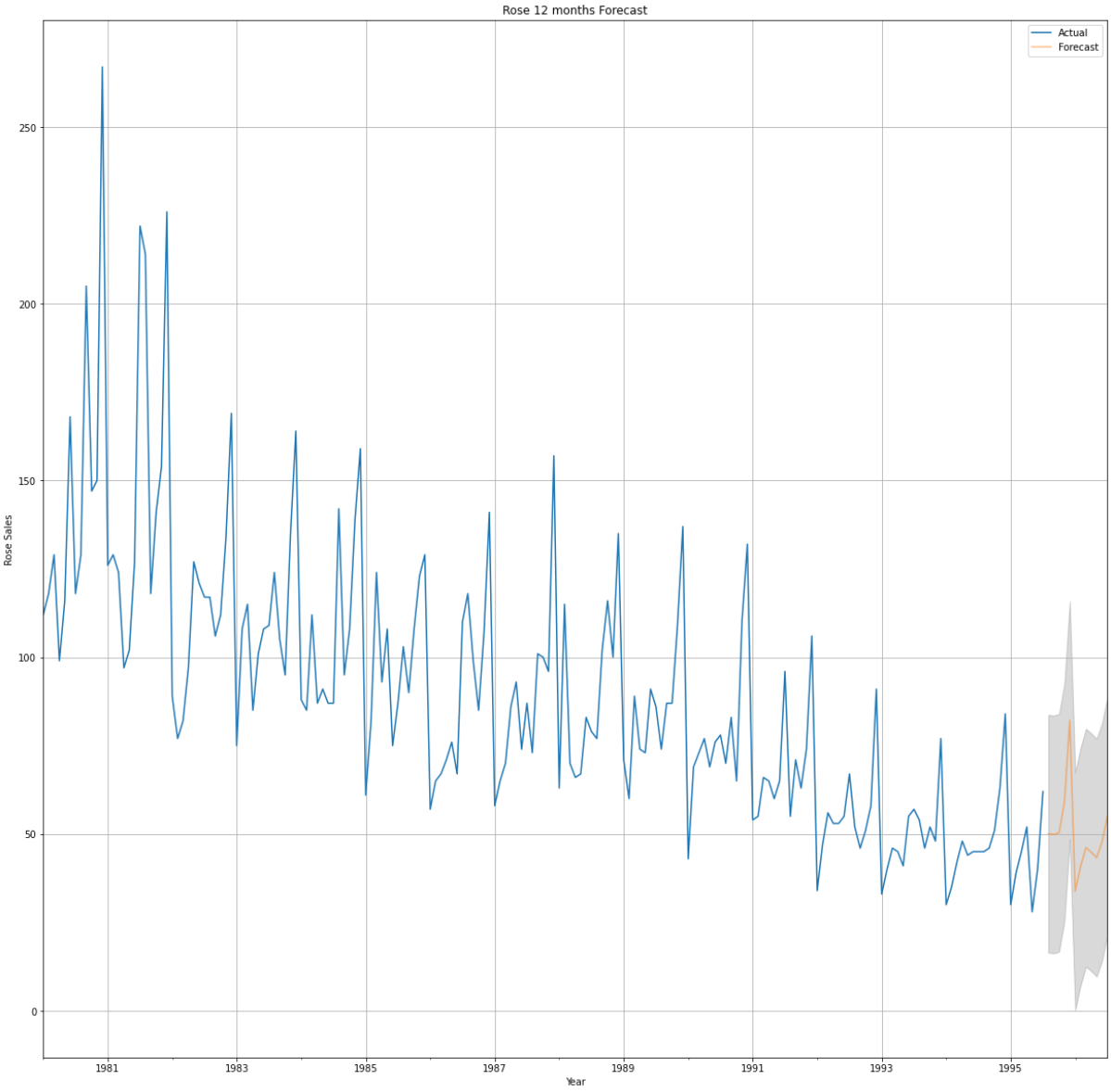

Figure 37:Rose-12 months Forecast

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

1. First read the data as a time series and plotted it on a graph to show how sales for Sparkling wines over the years.

2. Then performed some exploratory data analyses on the data sets, creating various types of charts for analyze the sales. The missing values are also imputed.

3. I split the data into test(data from the year 1991) and train(data before the year 1991).

4. Next I built the following models :
- Simple Exponential Smoothing Model
- Iterative Simple Exponential Smoothing Model
- Double Exponential Smoothing Model
- Iterative Double Exponential Smoothing Model
- Triple Exponential Smoothing Model
- Iterative Triple Exponential Smoothing Model.
- Linear Regression Model
- Naïve Approach
- Simple Average Model
- Moving Average Model

For all the above models RMSE value was calculated to understand the performance.

5. The stationarity of the data was checked by stating hypothesis for statistical testing and using ADF Test.

6. From here, we build ARIMA and SARIMA models, but first we examine the dataset. If the series is not stationary, we take the first difference of the series and converted into a stationary series.

7. The ARIMA/SARIMA models are built using AIC scores, we select the parameter with the least AIC and the model is built with it. RMSE is also calculated to check the performance.

8. The ARIMA/SARIMA models are built manually by calculating value of p,q,P,Q,s,d,D from ACF , PACF graph. RMSE is also calculated to check the performance.

9. Finally, we take the model with minimum RMSE value and build the most optimum model on the complete data .The sales for the next 12 months in future with 95% confidence intervals is predicted.

Recommendations

--> Fourth quarter has the highest sales among other quarter. So the company can stock up the wines in the second quarter itself to prepare themselves to supply the high demand in the fourth quarter.

--> Proper branding advertising in leading newspaper and magazines can be done. Social Media Advertising can also be done to improve the sales in the first 3 quarters.

--> First quarter has the lowest sales . So coupons and differs can be offered to boost up the sales.

--> Over the years sales are decreasing this can be due to more competition with new companies. As time progresses the wine company must bring in unique tastes.

--> Further information like age group,location of the customers can be analyzed to improve the model performance and get a better understanding of the Rose wine sales.

PROBLEM 2 - Sparkling

Problem Statement

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 Sparkling Wine Sales in the 20th century.

Introduction

The purpose of this whole exercise is to perform exploratory data analysis and perform Time Series Forecasting using Exponential Smoothing models, Regression, Naïve Forecast models, Simple Average models, Moving Average models, ARIMA and SARIMA models(using cut-off points of AIC,ACF and PACF plots) to forecast the sales of Sparkling wine.

Data Description

1. YearMonth: The Year and the Month on which its corresponding units of Rose Wine is sold.
2. Sparkling: Units of Sparkling Wine sold.

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

Sample 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

	YearMonth	Sparkling
182	1995-03	1897
183	1995-04	1862
184	1995-05	1670
185	1995-06	1688
186	1995-07	2031

Table 13: Sample of the Dataset2

The data is read from the excel file and the above tables shows the first and last 5 rows of the dataset.There are 187 rows in the dataframe. The Sparkling is the variable to be forecasted . YearMonth denotes the year and month values ranging from Jan 1980 to July 1995.

There are no duplicates in the datatset.

```
There are 0 duplicates in the dataset
```

The initial datatype of the columns before indexing are:

```
YearMonth    object
Sparkling     int64
dtype: object
```

YearMonth column is converted into a Time Stamp index using to_datetime function and YearMonth is dropped.

Time_Stamp	Sparkling
1980-01-01	1686
1980-02-01	1591
1980-03-01	2304
1980-04-01	1712
1980-05-01	1471

Figure 38:Sparkling-Timestamped data

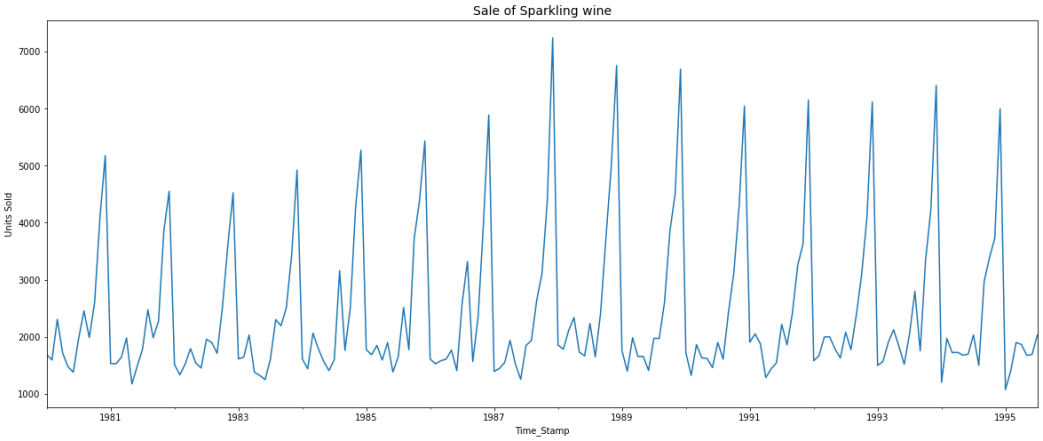

Figure 39:Sales of Sparkling plot

From the pot we can see there is no trend but seasonality is present. There is no missing value

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

	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

Figure 40:Sparkling description

The minimum sales was 1070 units of wine and maximum was 7242 units of wine through all the years.

There are no null values in the dataset.

```
Sparkling    0
dtype: int64
```

Monthly Sales

Monthly Sales across years

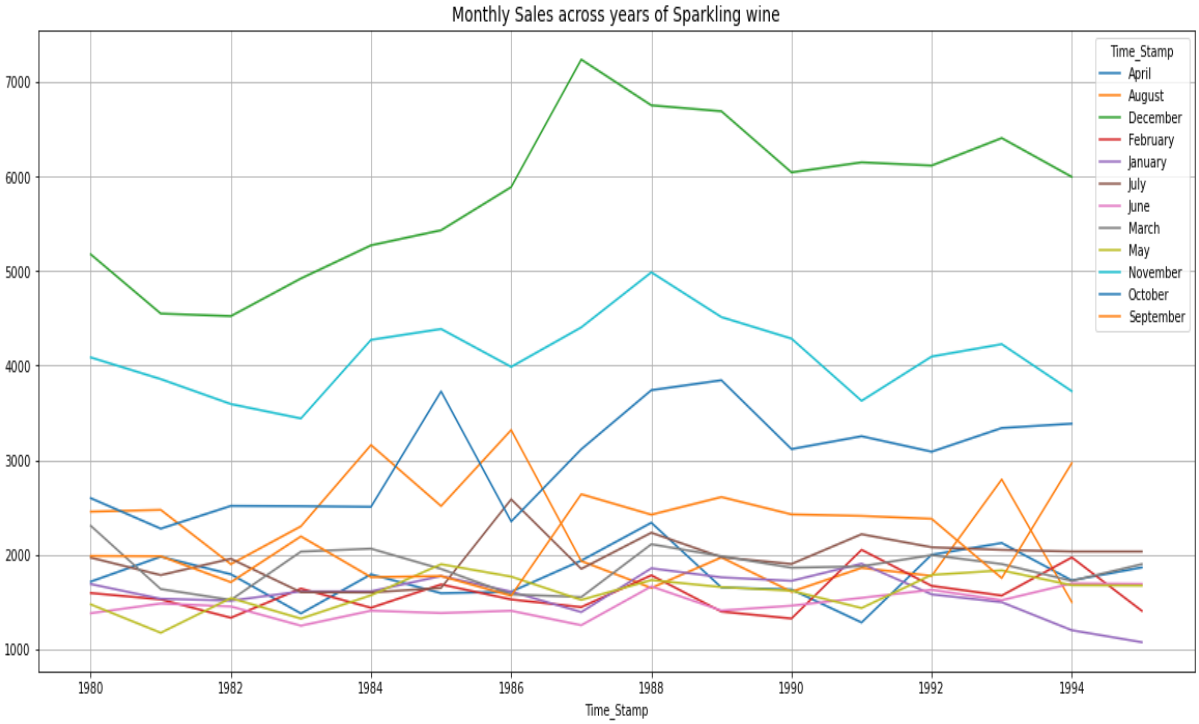

Figure 41:Sparkling Monthly sales over years

From the plot we can infer that December month has the highest sales across all years and November month has the second highest sales across most of the years.

Monthly Sales Sum Barplot of Sparkling wine

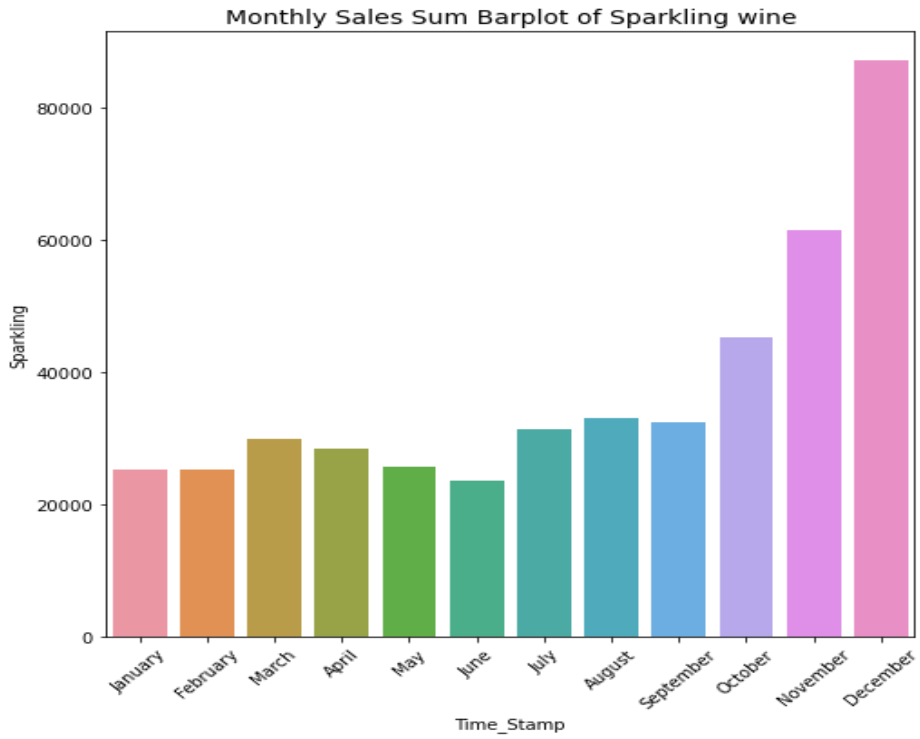

Figure 42:Sparkling Barplot Monthly sales

December has the highest sales of Sparkling wine with above 80000 units combining all the years. June has the lowest sales of Sparkling wine with slightly above 20000 units combining all the years.

Boxplot of Monthly Sales of Sparkling wine

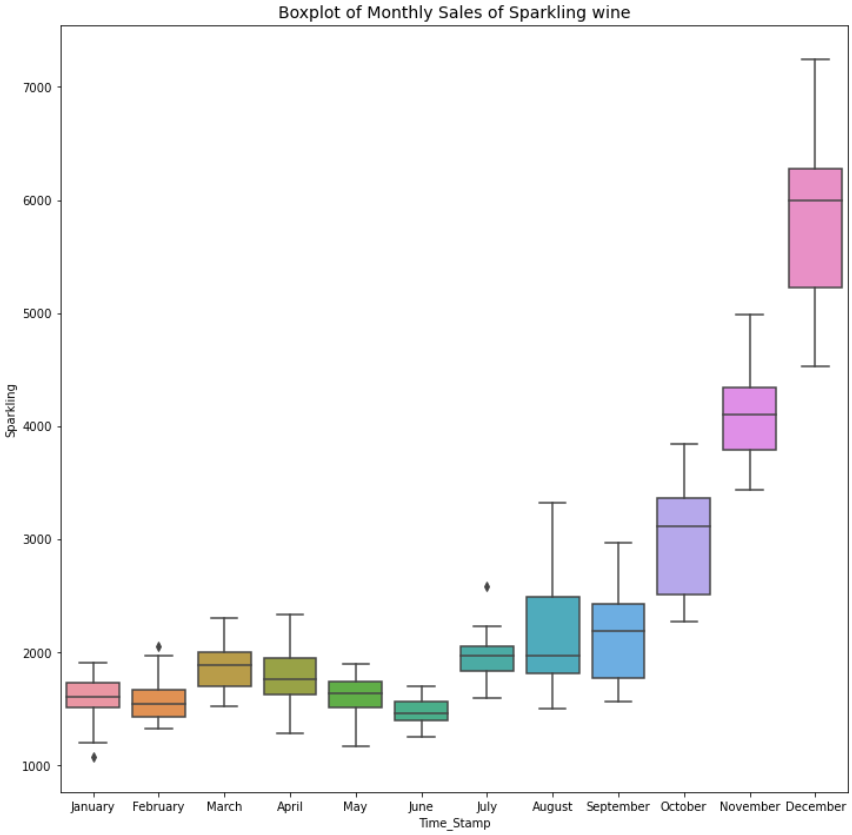

Figure 43:Sparkling Boxplot Monthly sales

There is an increasing trend month wise with January has the sale with the least value of almost 1000.

Month Plot

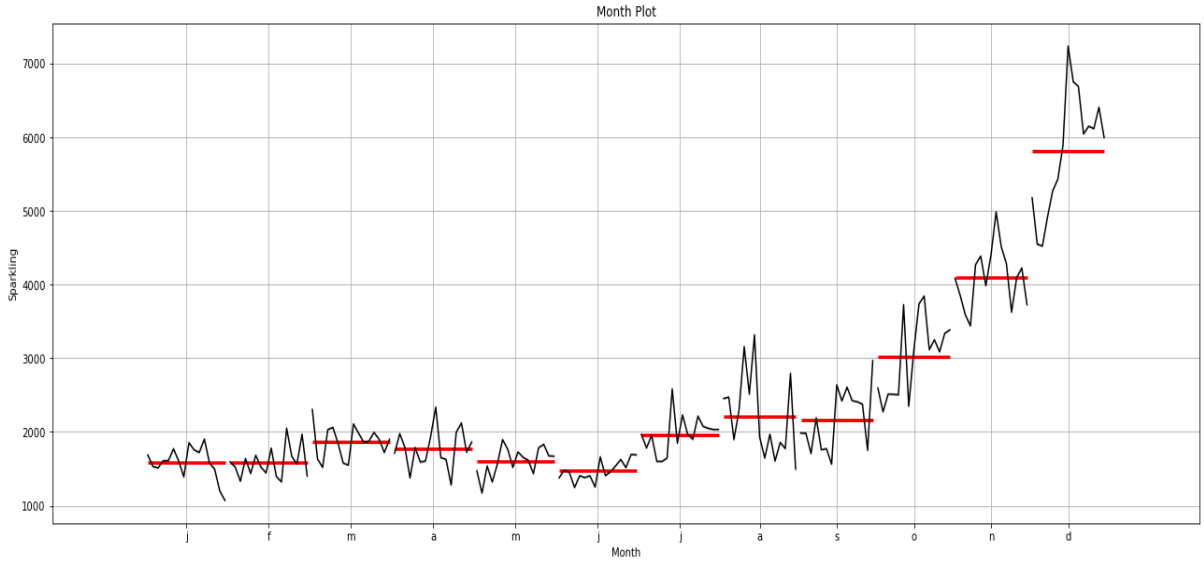

Figure 44:Sparkling Monthplot

The months are marked orderly in x-axis. There is an increasing trend with the months.

Quarterly Sales

Quarterly Sales across years of Sparkling wine

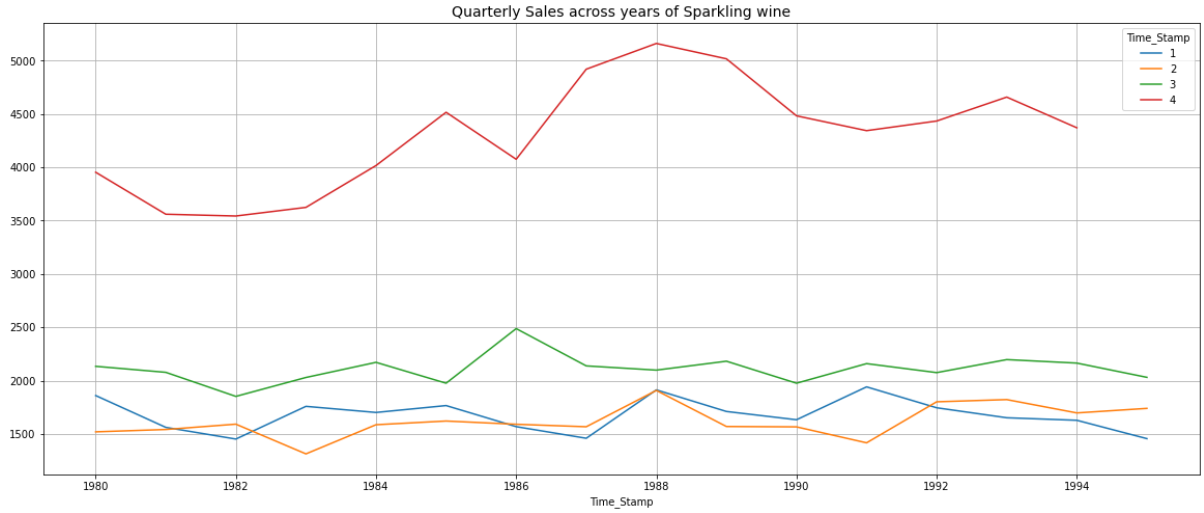

Figure 45:Sparkling Quarterly sales over years

The fourth quarter has the most sales of wine throughout all the years. Third quarter has the second highest sales of wine throughout all the years

Barplot of Quarterly Sales Sum of Sparkling wine

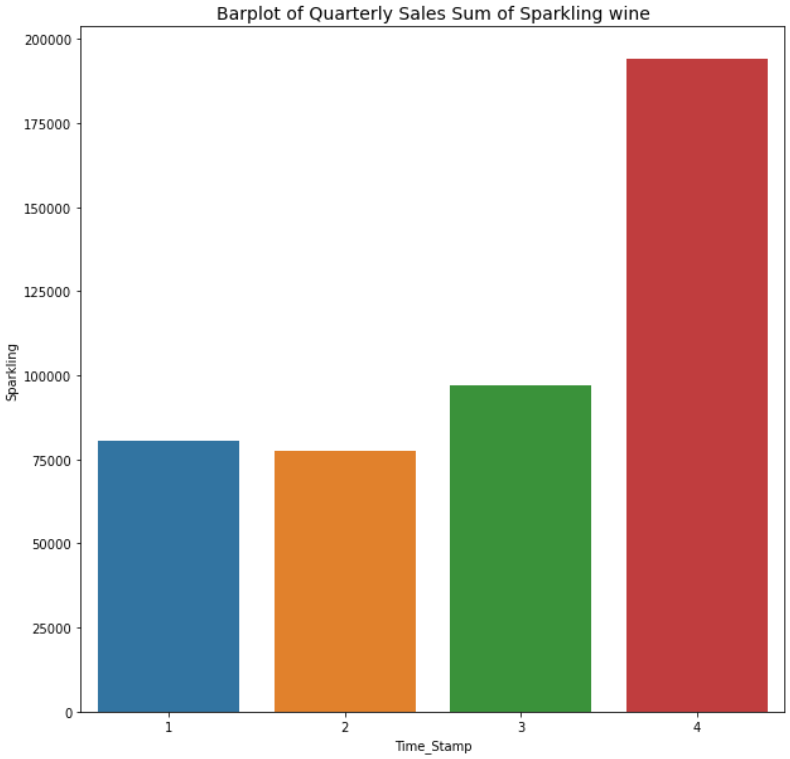

Figure 46:Sparkling Barplot Quarterly

4th quarter has the highest sales with nearly 190000 units of wine combining all the years. 2nd quarter has lesser sales than the first quarter combining all the years.

Boxplot of Quarterly Sales of Sparkling wine

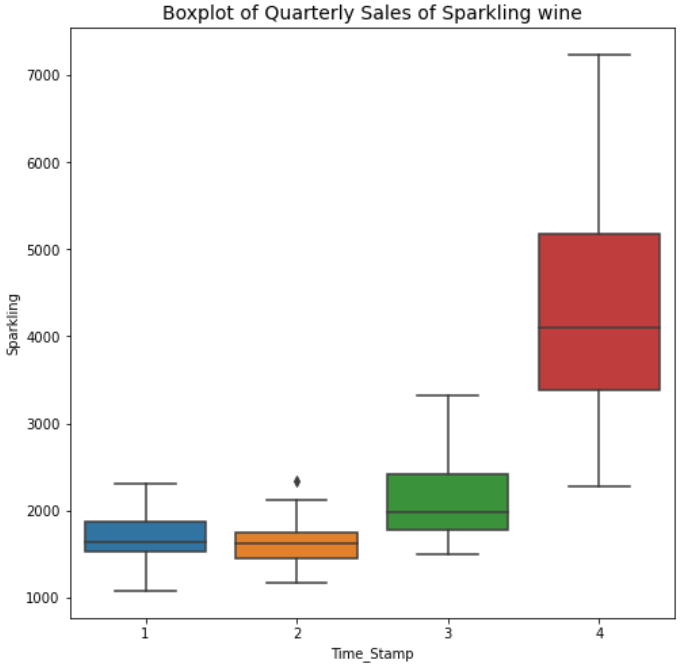

Figure 47:Sparkling Quarterly Boxplot

The sales are increasing with each subsequent quarter throughout all the years.

Yearly Sales

Time_Stamp	Sparkling
1980	2367.166667
1981	2185.583333
1982	2110.083333
1983	2181.666667
1984	2369.250000
1985	2470.000000
1986	2430.833333
1987	2521.500000
1988	2770.500000
1989	2620.250000
1990	2414.750000
1991	2465.583333
1992	2514.250000
1993	2582.583333
1994	2465.333333
1995	1660.000000

Figure 48:Sparkling Yearly Sales

Above are the average sales each year.

Barplot of Yearly Sales Sum of Sparkling wine

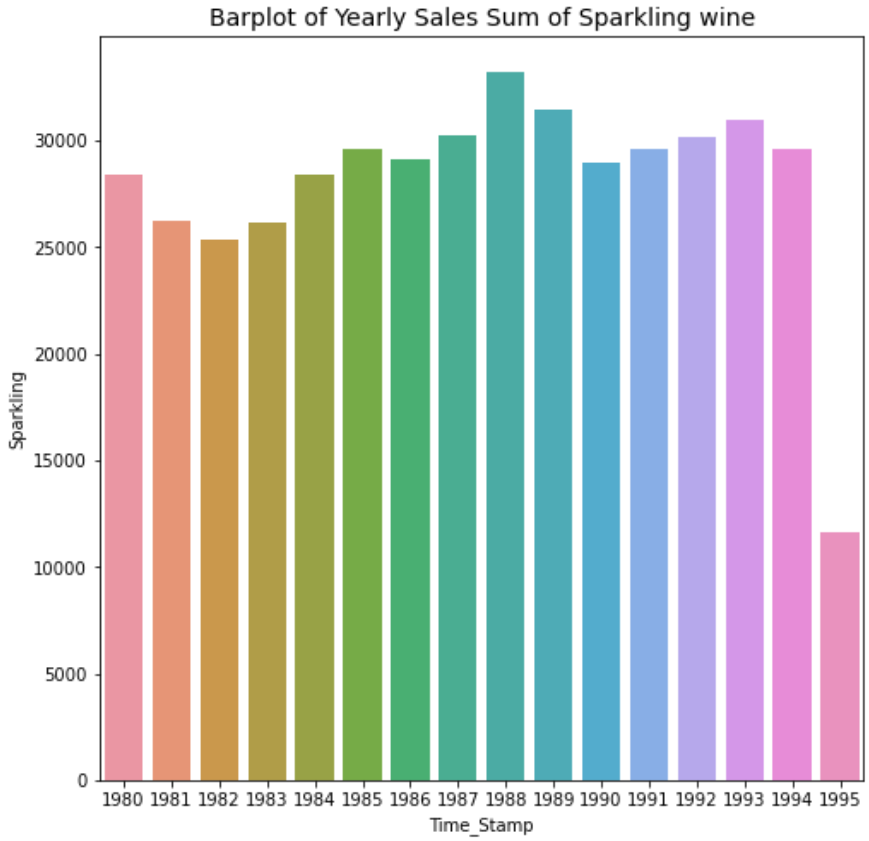

Figure 49:Sparkling Barplot Yearly

1988 has the highest sale of all the years.

Boxplot of Yearly Sales of Sparkling wine

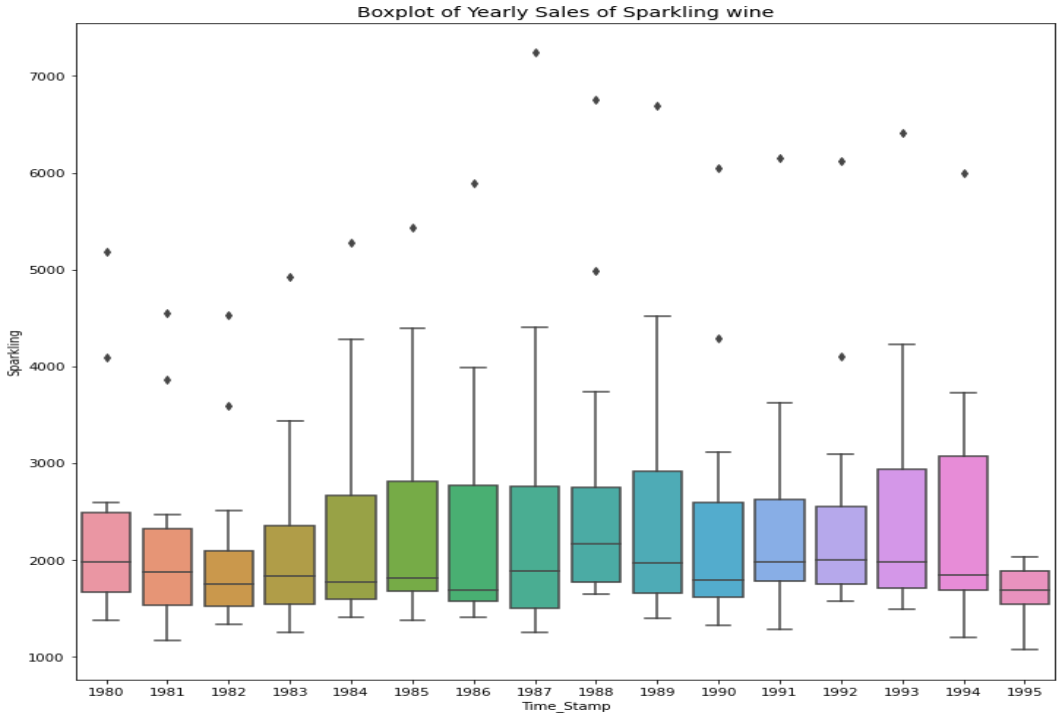

Figure 50:Sparkling Boxplot Yearly

No trend is visible is seen in the data. 1995 has the lowest sale of wine.

Decomposition

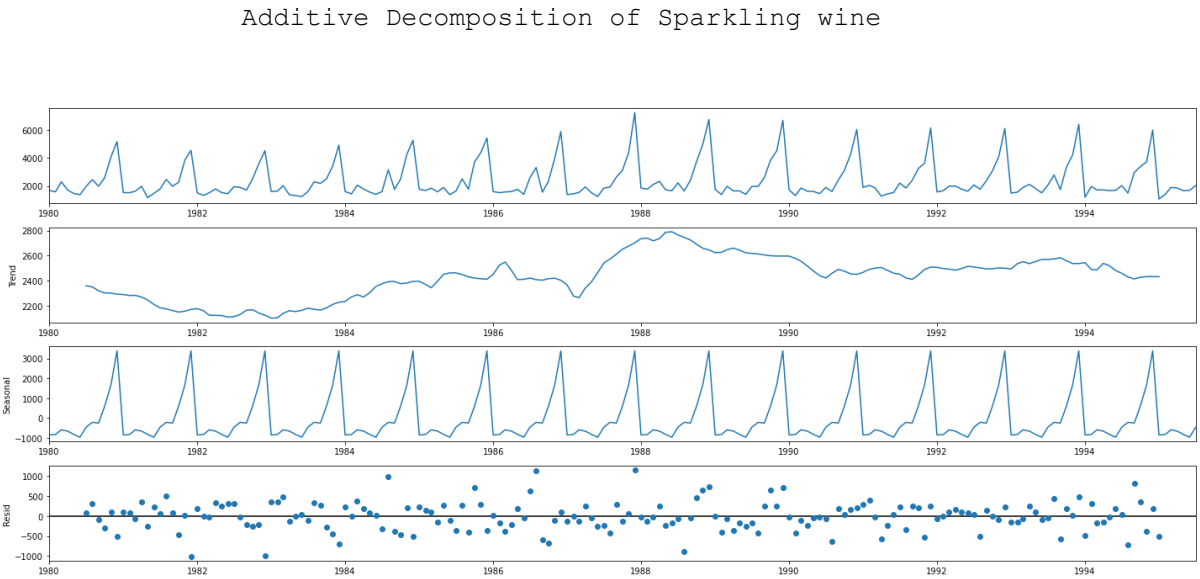

Figure 51:Sparkling Additive Decomp

We see that the residuals are located around 0 from the residual and patter is present. The residuals are ranging from -1000 to +1000.

so further decomposing to multiplicative model to minimize the residuals. There is seasonality and we don't observe pronounced trend.

The first 12 months trend, seasonality and residual values of Sparkling wine sales dataset are shown below:

```
Additive Seasonality
 Time_Stamp
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
```

```
Additive Trend
 Time_Stamp
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
```

```
Additive Residual
 Time_Stamp
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
```

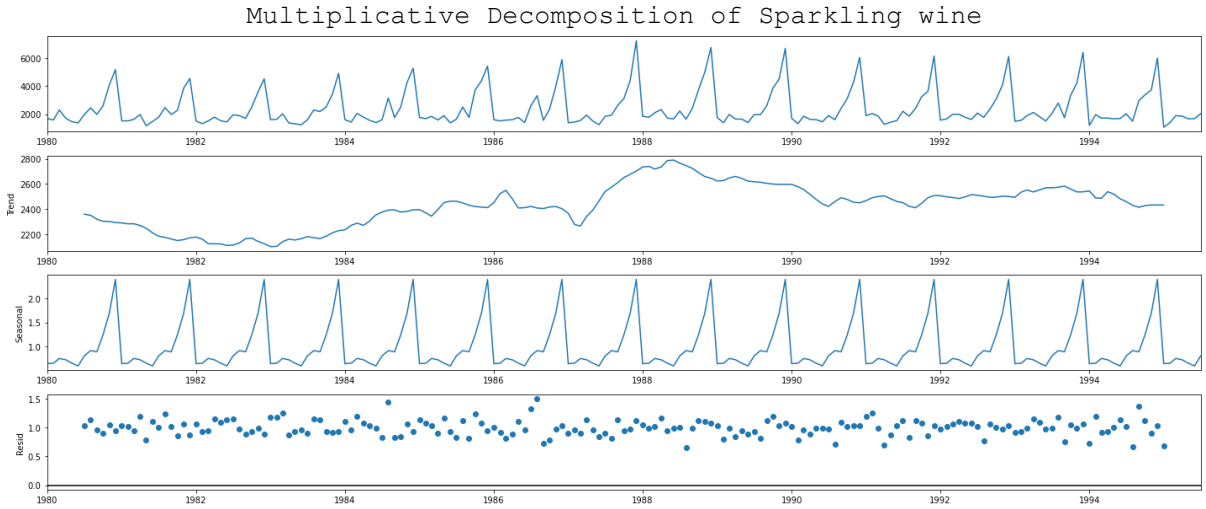

Figure 52:Sparkling Mulitpl Decomp

In the multiplicative decomposition a lot of residuals are located around 1 and it has pattern too. There is seasonality and we don't observe pronounced trend.

Since both additive and multiplicative has pattern in its residuals we can choose additive has the simpler and suitable decomposition method.

The first 12 months multiplicative trend, seasonality and residual values of Sparkling wine sales dataset are shown below:

```
Multiplicative Trend
 Time_Stamp
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
```

```
Multiplicative Seasonality
 Time_Stamp
1980-01-01    0.649843
1980-02-01    0.659214
1980-03-01    0.757440
1980-04-01    0.730351
1980-05-01    0.660609
1980-06-01    0.603468
1980-07-01    0.809164
1980-08-01    0.918822
1980-09-01    0.894367
1980-10-01    1.241789
1980-11-01    1.690158
1980-12-01    2.384776
Name: seasonal, dtype: float64
```

```
Multiplicative Residual
 Time_Stamp
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    1.029230
1980-08-01    1.135407
1980-09-01    0.955954
1980-10-01    0.907513
1980-11-01    1.050423
1980-12-01    0.946770
Name: resid, dtype: float64
```

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

The first 5 and last 5 rows of train data shows that it starts from January 1980 and ends at December 1990.

Time_Stamp	Sparkling
1980-01-01	1686
1980-02-01	1591
1980-03-01	2304
1980-04-01	1712
1980-05-01	1471

Time_Stamp	Sparkling
1990-08-01	1605
1990-09-01	2424
1990-10-01	3116
1990-11-01	4286
1990-12-01	6047

Table 14:Sparkling Train Data

The first 5 and last 5 rows of test data shows that it starts from January 1991 and ends at July 1995.

Time_Stamp	Sparkling
1991-01-01	1902
1991-02-01	2049
1991-03-01	1874
1991-04-01	1279
1991-05-01	1432

Time_Stamp	Sparkling
1995-03-01	1897
1995-04-01	1862
1995-05-01	1670
1995-06-01	1688
1995-07-01	2031

Table 15:Sparkling Test Data

2.4. 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.

Simple Exponential Smoothing Auto Fit Model

Simple Exponential Smoothing(SES) 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.

A SES model is built on train data with initialization_method value as estimated and the following parameters with values.

initialization_method - Method for initialize the recursions.

Use_brute=True -> Search for good starting values using a brute force (grid) optimizer

Optimized=True -> Estimate model parameters by maximizing the log-likelihood.

```
{'smoothing_level': 0.07029120765764557,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 1764.0137060346985,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

The model built is used to forecast for next 55 months which is the length of test data.

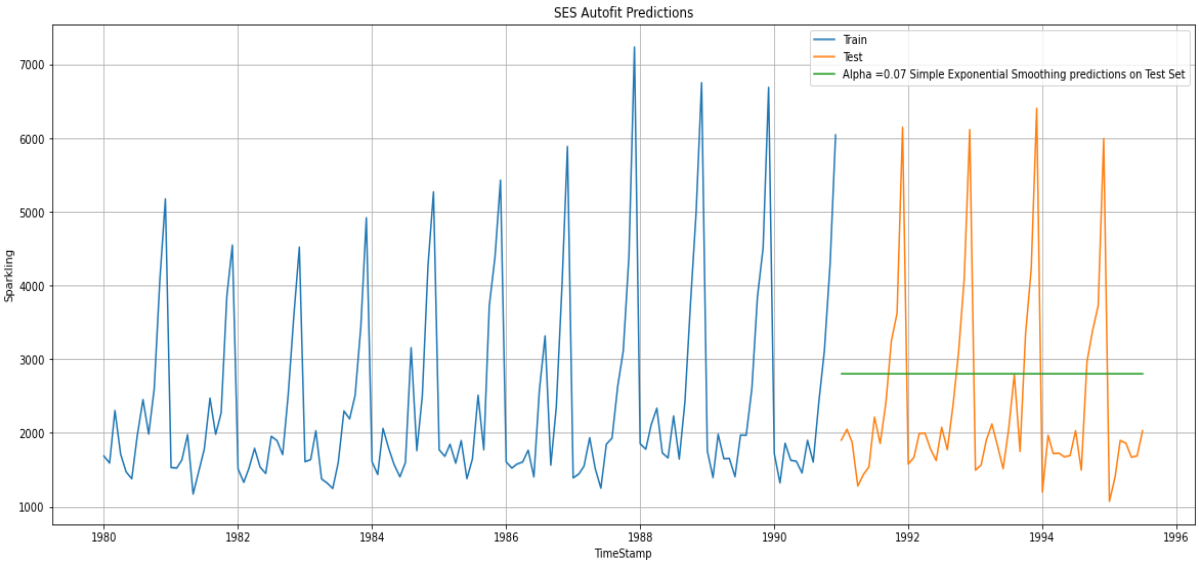

Figure 53:Sparkling SES plot

RMSE is calculated on test data.

SES Autofit RMSE on Test data: 1338.01

<u>Double Exponential Smoothing - Holt Autofit Model</u>

Double exponential smoothing(DES) employs a level component and a trend component at each period.

A DES model is built on train data with initialization_method value as estimated , use_brute=True and Optimized=True. The other parameters and the values are:

```
{'smoothing_level': 0.6649999999999999, 'smoothing_trend': 0.0001, 'smoothing_seasonal': nan, 'damping_trend': nan, 'initial_level': 1502.1999999999991, 'initial_trend': 74.87272727272739, 'initial_seasons': array([], dtype=float64), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

In [381]:

The model built is used to forecast for next 55 months which is the length of test data.

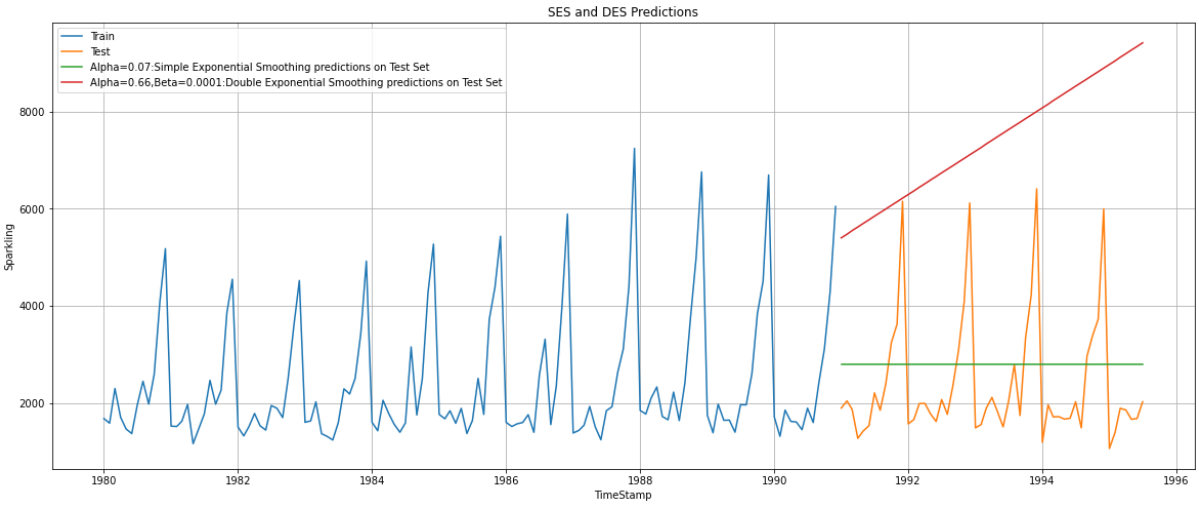

Figure 54:Sparkling DES plot

RMSE is calculated on test data.

DES Autofit RMSE on Test Data: 5291.88

Triple Exponential Smoothing - ETS(A, A, A) - Holt Winter's linear method with additive errors Autofit Model

Triple exponential smoothing(TES) 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

In this model both trend and seasonality is chosen as additive. A DES model is built on train data with initialization_method value as estimated , use_brute=True and Optimized=True. The other parameters and the values are:

```
{'smoothing_level': 0.11127227248079453, 'smoothing_trend': 0.0123608043050
88534, 'smoothing_seasonal': 0.46071766688111543, 'damping_trend': nan, 'in
itial_level': 2356.577980956387, 'initial_trend': -0.10243675533021725, 'in
itial_seasons': array([-636.23319334, -722.9832009 , -398.64410813, -473.43
045416,
       -808.42473284, -815.34991402, -384.23065038,   72.99484403,
       -237.44226045,  272.32608272, 1541.37737052, 2590.07692296]), 'use_b
oxcox': False, 'lamda': None, 'remove_bias': False}
```

In [386]:

The model built is used to forecast for next 55 months which is the length of test data.

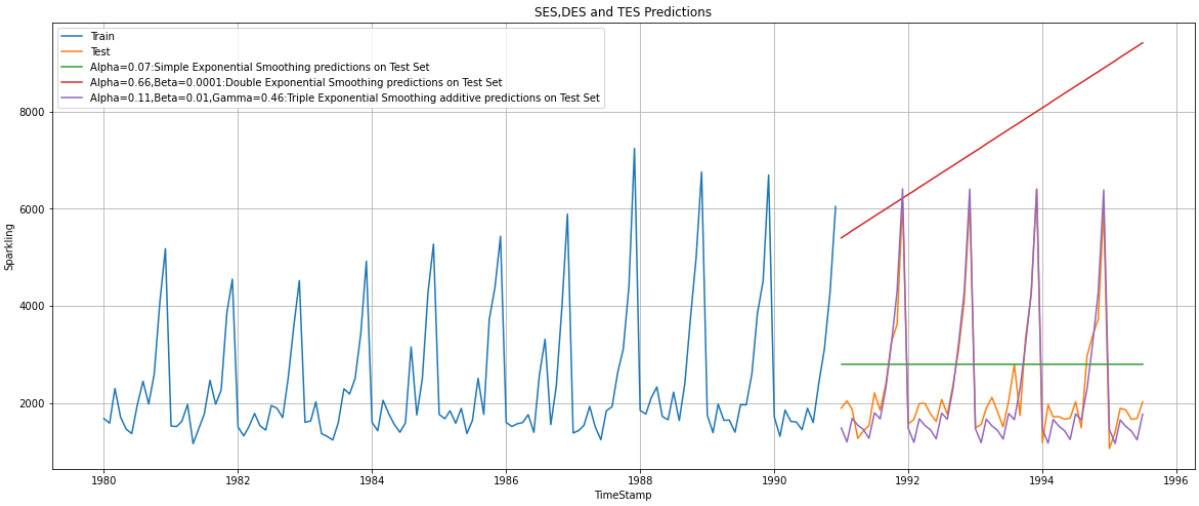

Figure 55:Sparkling TES add seas plot

RMSE is calculated on test data.

TES Additive Autofit RMSE on Test data: 378.95

Triple Exponential Smoothing - ETS(A, A, M) - Holt Winter's linear method with multiplicative errors

In this model trend is additive and seasonality is chosen as multiplicative. A DES model is built on train data with initialization_method value as estimated ,

use_brute=True and Optimized=True. The other parameters and the values are:

The model built is used to forecast for next 55 months which is the length of test data.

```
{'smoothing_level': 0.11133818361298699, 'smoothing_trend': 0.0495051310195
09915, 'smoothing_seasonal': 0.3620795793580111, 'damping_trend': nan, 'ini
tial_level': 2356.4967888704355, 'initial_trend': -10.187944726007238, 'ini
tial_seasons': array([0.71296382, 0.68242226, 0.90755008, 0.80515228, 0.655
97218,
       0.65414505, 0.88617935, 1.13345121, 0.92046306, 1.21337874,
       1.87340336, 2.37811768]), 'use_boxcox': False, 'lamda': None, 'remov
e_bias': False}
```

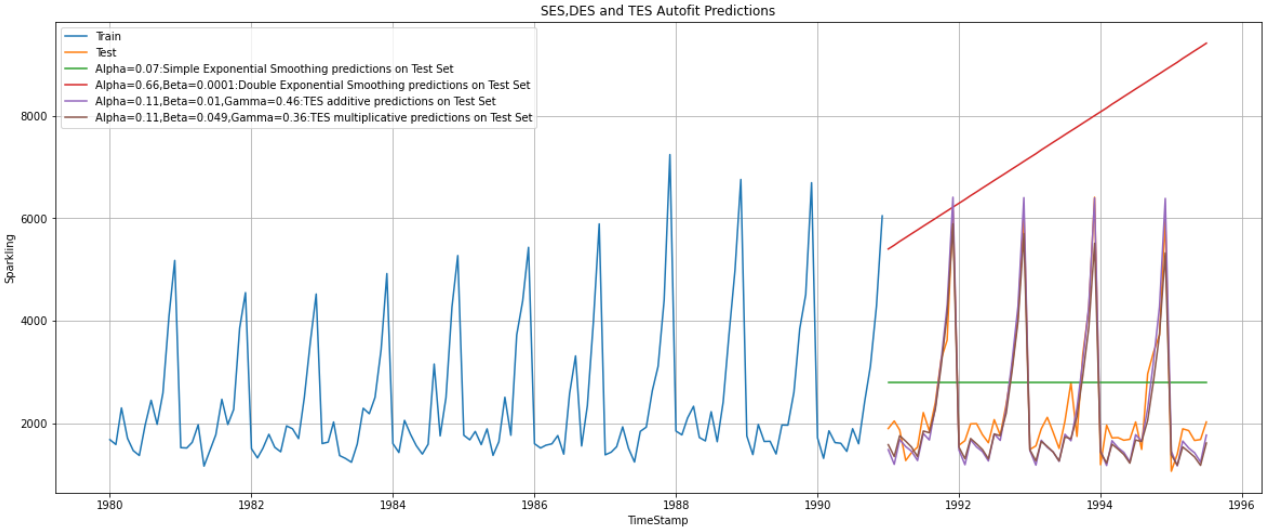

Figure 56:Sparkling TES multipl seas plot

RMSE is calculated on test data.

TES multiplicative autofit RMSE on Test data : 404.29

Both the Triple Exponential Smoothing models are picking up the seasonal component as well which can be inferred from the graph.

Iterative Method for Simple Exponential Smoothing

A SES model is built on train data with initialization_method value as estimated , use_brute=True and Optimized=False. The value of smoothing_level(alpha) is taken from 0.1 to 1 and its corresponding RMSE value is calculated as shown below.

	Alpha Values	Test RMSE
0	0.1	1375.393335
1	0.2	1595.206839
2	0.3	1935.507132
3	0.4	2311.919615
4	0.5	2666.351413
5	0.6	2979.204388
6	0.7	3249.944092
7	0.8	3483.801006
8	0.9	3686.794285
9	1.0	3864.279352

We can see that alpha as 0.1 as the least RMSE.

Plot the prediction of the model with smoothing level value as 0.1.

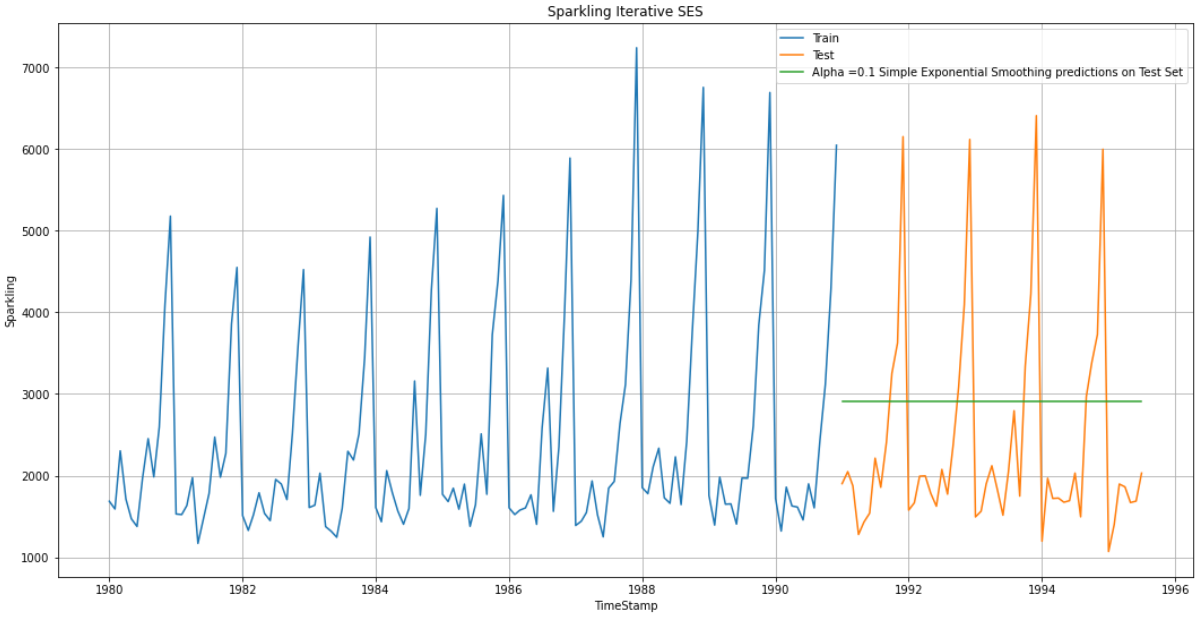

Figure 57:Sparkling Iterative SES

Iterative SES RMSE on Test data : 1375.39

Iterative Method for Double Exponential Smoothing

A DES model is built on train data with initialization_method value as estimated , use_brute=True and Optimized=False. The values of smoothing_level(alpha) and smoothing_trend(beta) is taken from 0.1 to 1 and

its corresponding RMSE value is calculated .The first 10 rows with least RMSE is shown below.

	Alpha Values	Beta Values	Test RMSE
0	0.1	0.1	1777.734773
1	0.1	0.2	2599.314701
10	0.2	0.1	3611.766690
2	0.1	0.3	4287.469279
20	0.3	0.1	5908.185555
3	0.1	0.4	6044.157399
11	0.2	0.2	6878.567126
4	0.1	0.5	7386.659388
30	0.4	0.1	8039.102035
5	0.1	0.6	8553.627687

The model with smoothing level as 0.1 and smoothing trend as 0.1 has the least RMSE value 1777.73

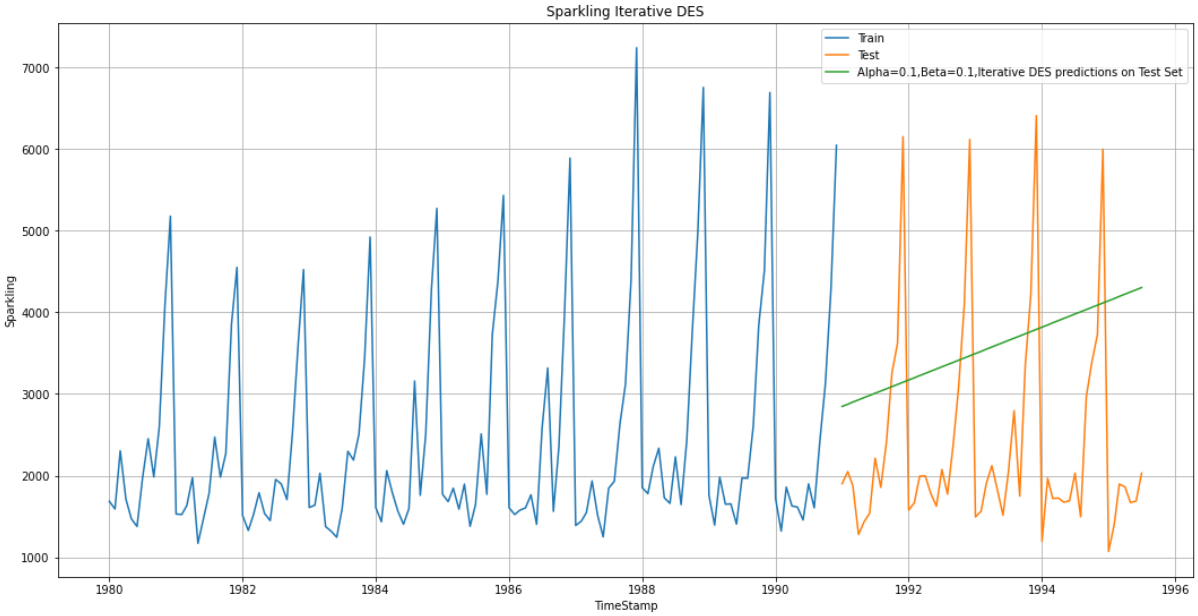

Figure 58:Sparkling Iter DES plot

Iterative SES RMSE on Test data : 1777.73

Iterative Method - Triple Exponential Smoothing - ETS(A, A, A)

A TES model with trend and seasonality as additive is built on train data with initialization_method value as estimated , use_brute=True and Optimized=False. The values of smoothing_level(alpha) ,smoothing_trend(beta) and smoothing_seasonal(gamma)is taken from 0.1 to

1 and its corresponding RMSE value is calculated. First 5 rows of the data which shows the 5 least RMSE ones are shown below.

	Alpha Values	Beta Values	Gamma Values	Test RMSE
30	0.1	0.4	0.1	342.934716
110	0.2	0.2	0.1	343.121437
156	0.2	0.6	0.7	348.792360
200	0.3	0.1	0.1	390.834811
20	0.1	0.3	0.1	391.304169

The model with smoothing level as 0.1,smoothing trend as 0.4 and smoothing seasonal 0.1 has the least RMSE value 342.93

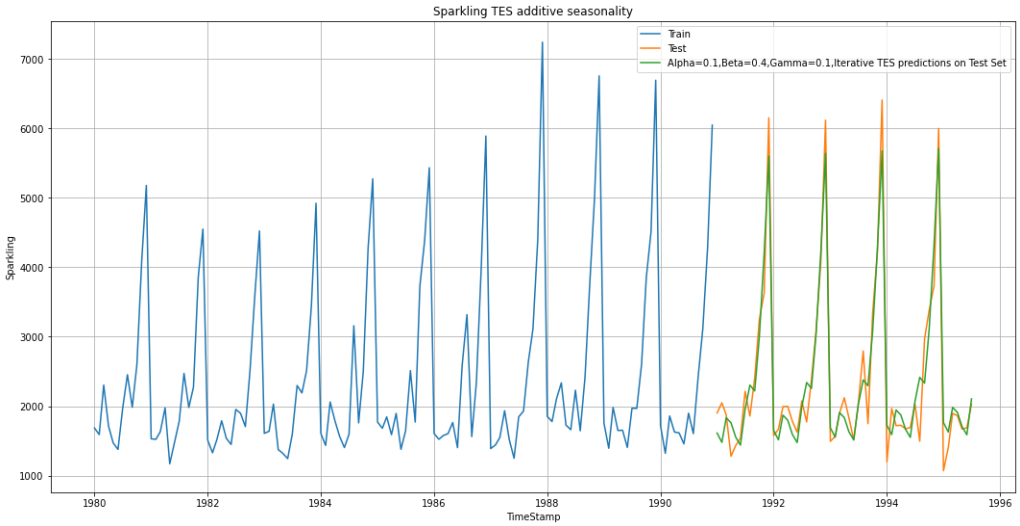

Figure 59:Sparkling-Iter TES add seas

Iterative TES Additive RMSE on Test data : 342.93

Iterative Method - Triple Exponential Smoothing - ETS(A, A, M)

A TES model with trend as additive and seasonality as multiplicative is built on train data with initialization_method value as estimated , use_brute=True and Optimized=False. The values of smoothing_level(alpha) ,smoothing_trend(beta) and smoothing_seasonal(gamma)is taken from 0.1 to

1 and its corresponding RMSE value is calculated. First 5 rows of the data which shows the 5 least RMSE ones are shown below.

	Alpha Values	Beta Values	Gamma Values	Test RMSE
301	0.4	0.1	0.2	317.434302
211	0.3	0.2	0.2	329.037543
200	0.3	0.1	0.1	337.080969
110	0.2	0.2	0.1	340.186457
402	0.5	0.1	0.3	345.913415

The model with smoothing level as 0.4,smoothing trend as 0.1 and smoothing seasonal 0.2 has the least RMSE value 317.43

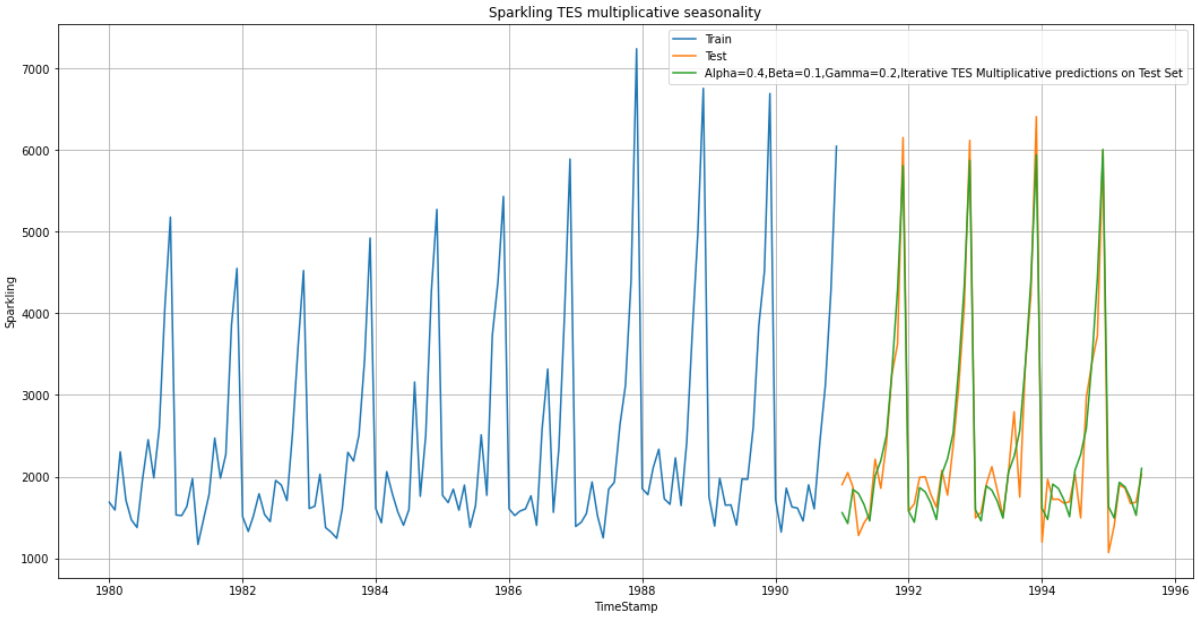

Figure 60:Sparkling Iterative TES multipl seas plot

The RMSE of iterative TES multiplicative is 317.43

Linear Regression Model

Linear regression uses the relationship between the data-points to draw a straight line through all them. This line can be used to predict future values.

The training and testing time instance is created.

```
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, 44, 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]
```

The instances are added to the train and test data as column 'time' and treated as an independent variable.

First few rows of Training Data

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

First few rows of Test Data

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

Last few rows of Training Data

Time_Stamp	Sparkling	time
1990-08-01	1605	128
1990-09-01	2424	129
1990-10-01	3116	130
1990-11-01	4286	131
1990-12-01	6047	132

Last few rows of Test Data

Time_Stamp	Sparkling	time
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

Table 16:Sparkling LR Train Data

Sparkling column is treated as a dependent variable. Both the train independent and dependent variable are fitted on the Linear Regression model from sklearn library.

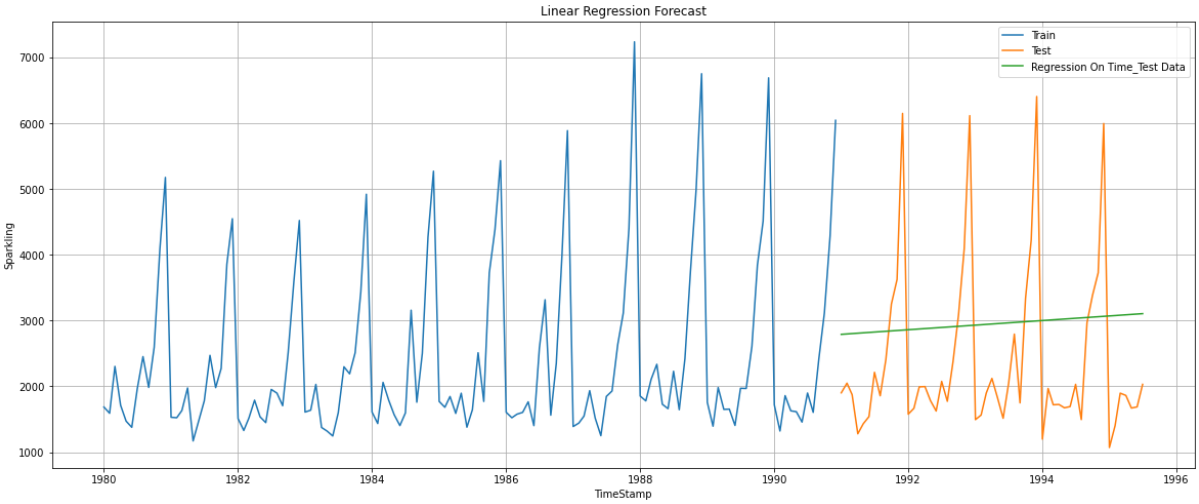

Figure 61:Sparkling-Linear Regr plot

The fitted model is sued to predict on test data and the output of Rose wine sales is used to calculate RMSE along with actual Rose wine sales.

The RMSE of Linear Regression forecast on the Test Data is 1389.14

Naive Forecast Model:

Naïve forecasting is the technique in which the last period's sales are used for the next period's forecast without predictions or adjusting the factors. Forecasts produced using a naïve approach are equal to the final observed value.

The last 5 rows of the train data are:

	Sparkling
Time_Stamp	
1990-08-01	1605
1990-09-01	2424
1990-10-01	3116
1990-11-01	4286
1990-12-01	6047

The last value is 6047 which is treated as the forecasted value for the future time period as shown below.

The first 5 rows of test data are:

Time_Stamp	Sparkling	naive
1991-01-01	1902	6047
1991-02-01	2049	6047
1991-03-01	1874	6047
1991-04-01	1279	6047
1991-05-01	1432	6047

The last 5 rows of test data are:

Time_Stamp	Sparkling	naive
1995-03-01	1897	6047
1995-04-01	1862	6047
1995-05-01	1670	6047
1995-06-01	1688	6047
1995-07-01	2031	6047

Table 17:Naive Forecast Value Test

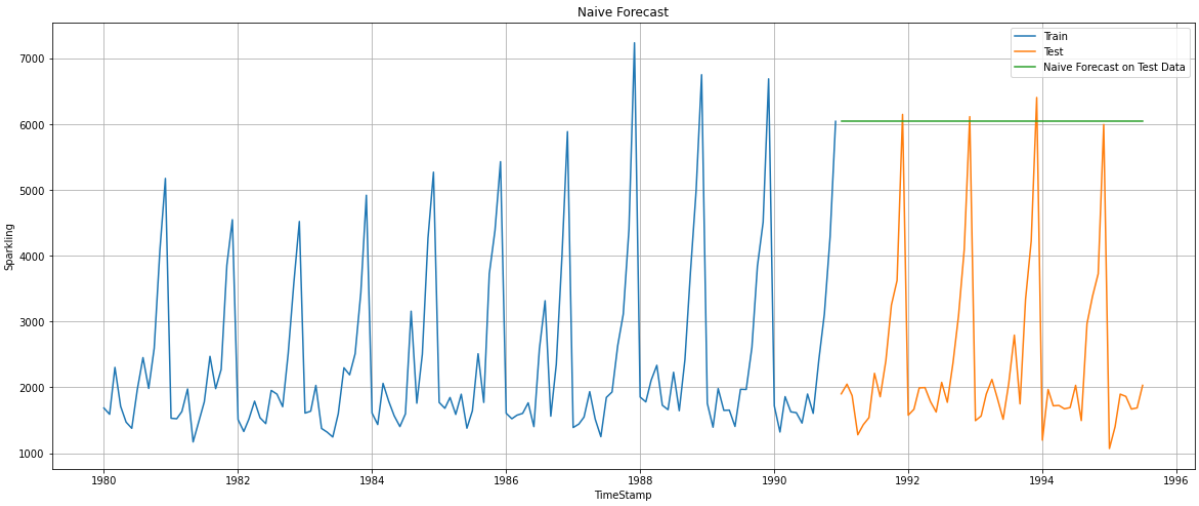

Figure 62:Sparkling-Naive Forecast plot

The predicted test output along with the actual test output is used for calculating RMSE.

The RMSE for NaiveModel forecast on the Test Data 3864.28

Simple Average Model

In the Simple Average model ,forecast is equal to the average of historical data.
The average of the train data is taken and added as the forecast of Test data after converting into an integer number(mean -> 2403.780303030303) as shown below.

Time_Stamp	Sparkling	mean_forecast
1991-01-01	1902	2403
1991-02-01	2049	2403
1991-03-01	1874	2403
1991-04-01	1279	2403
1991-05-01	1432	2403

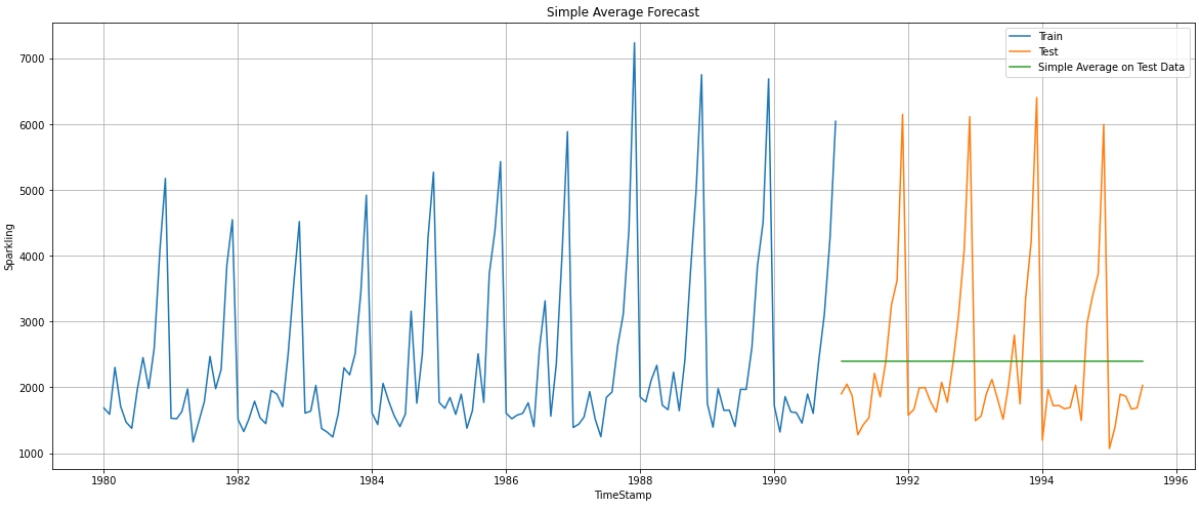

Figure 63:Sparkling-SA plot

The predicted test output along with the actual test output is used for calculating RMSE.

The RMSE for Simple Average forecast on the Test Data 1275.08.

Moving Average Forecast Model

Moving Average Forecast Model takes an average of a set of numbers in a given range while moving the range.

Moving Average is calculated on train data for which the following values are given to rolling function.

2-takes moving average of 2 months of data

3-takes moving average of 3 months of data

6-takes moving average of 6 months of data

12-takes moving average of 12 months of data

The first 5 rows of training data is the following:

Time_Stamp	Sparkling	Trailing_2	Trailing_3	Trailing_6	Trailing_12
1980-01-01	1686	NaN	NaN	NaN	NaN
1980-02-01	1591	1638.5	NaN	NaN	NaN
1980-03-01	2304	1947.5	1860.333333	NaN	NaN
1980-04-01	1712	2008.0	1869.000000	NaN	NaN
1980-05-01	1471	1591.5	1829.000000	NaN	NaN

Figure 64:Sparkling-MA-Train head

The last 5 rows of training data is the following:

Time_Stamp	Sparkling	Trailing_2	Trailing_3	Trailing_6	Trailing_12
1990-08-01	1605	1752.0	1653.666667	1677.166667	2563.750000
1990-09-01	2424	2014.5	1976.000000	1771.333333	2548.416667
1990-10-01	3116	2770.0	2381.666667	2019.333333	2487.666667
1990-11-01	4286	3701.0	3275.333333	2464.500000	2468.666667
1990-12-01	6047	5166.5	4483.000000	3229.500000	2414.750000

Figure 65:Sparkling-MA-Train tail

The value 5166 is taken as the forecasted value for the test for 2 point Moving Average. (5166.5 rounded off)

The value 4483 is taken as the forecasted value for the test for 3 point Moving Average.

The value 3229 is taken as the forecasted value for the test for 6 point Moving Average. (3229.500000 rounded off)

The value 2414 is taken as the forecasted value for the test for 12 point Moving Average. (2414.750000 rounded off)

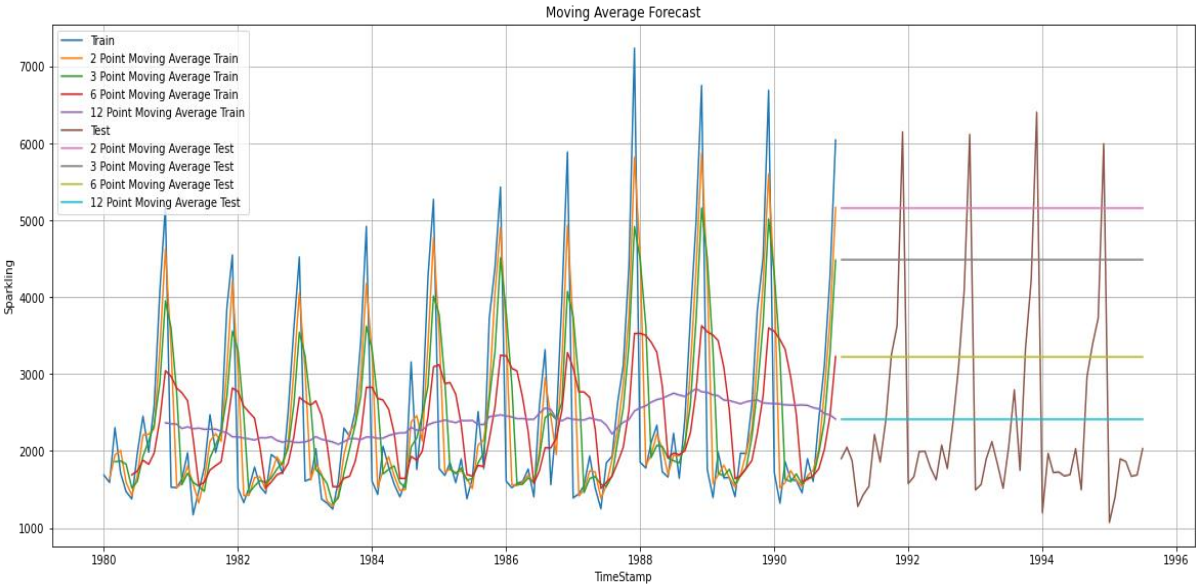

Figure 66:Sparkling-Mov Avg-plot

The RMSE for 2 point Moving Average Model forecast on the Test Data is 3046.52
The RMSE for 3 point Moving Average Model forecast on the Test Data is 2443.0
The RMSE for 6 point Moving Average Model forecast on the Test Data is 1521.34
The RMSE for 12 point Moving Average Model forecast on the Test Data is 1275.16

2.5. 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.

Augmented Dickey Fuller test (ADF Test) is a common statistical test used to test whether a given Time series is stationary or not. It is one of the most commonly used statistical test when it comes to analyzing the stationary of a series.
The hypothesis for the statistical test is:

H0-Null Hypothesis: Time series is non-stationary

H1-Alternate Hypothesis: Time series is stationary

The ADF Test is conducted on the train data

```
Results of Augmented 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
```

The p-value obtained by the test should be less than the significance level (say 0.05) to reject the Null hypothesis or it fails to reject the Null hypothesis.

p value obtained from the ADF test is 0.6697 which is greater than 0.05 . Hence we fail to reject the Null Hypothesis and so we can say that data is non-stationary.

To convert the data into a stationary one, the difference of a Dataframe value with the value in the previous row is taken and remove missing values. The ADF Test is taken again on the modified train data.

```
Results of Augmented 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
```

After modifying, the p-value 2.280104e-12 obtained by the test is less than 0.05. Now the data has been converted into a stationary one.

2.6. 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.

ARIMA Automated Train Model

Auto Regressive Integrated Moving Average (ARIMA) models are applied on time series data when the current value is assumed to be correlated to past values and past prediction errors. Therefore, these models are used in defining current value as a linear combination of past values and past prediction errors. Here, we have defined a few terms that would be useful in understanding ARIMA models in detail. ARIMA models can only be applied only on stationary time series data.

The Akaike information criterion (AIC) is an estimator of out-of-sample prediction error and thereby relative quality of statistical models for a given set of data. The least the AIC the better the model is .

The p,q value is taken from 0 to 3.‘d’ is taken as 1.

	param	AIC
7	(1, 1, 3)	2226.476259
10	(2, 1, 2)	2226.977906
3	(0, 1, 3)	2227.387719
6	(1, 1, 2)	2227.537319
15	(3, 1, 3)	2237.423464
14	(3, 1, 2)	2245.170645
13	(3, 1, 1)	2248.538423
9	(2, 1, 1)	2250.756824
11	(2, 1, 3)	2252.951943
2	(0, 1, 2)	2253.555026
5	(1, 1, 1)	2256.778569
1	(0, 1, 1)	2257.658950
12	(3, 1, 0)	2285.470080
8	(2, 1, 0)	2308.148040
4	(1, 1, 0)	2328.441974
0	(0, 1, 0)	2359.049380

The model has the parameter ‘order’ which has its values in the form of (p,d,q) where

p: Trend autoregression order.

d: Trend difference order.

q: Trend moving average order.

The p value is taken as 1, q as 2 and d as 1.(1,1,2) as it has the least AIC value.

ARIMA is built using stationary data after dropping its NA values since it reduces the AIC value .

enforce_stationarity→Whether or not to transform the AR parameters to enforce stationarity in the autoregressive component of the model.

enforce_invertibility→Whether or not to transform the MA parameters to enforce invertibility in the moving average component of the model.

enforce_stationarity and enforce_ invertibility is given as false.

The ARIMA model is built with those values and RMSE is calculated on test data.

The Summary of ARIMA model is

SARIMAX Results

Dep. Variable:	Sparkling	No. Observations:	131
Model:	ARIMA(1, 1, 3)	Log Likelihood	-1108.238
Date:	Sun, 05 Jun 2022	AIC	2226.476
Time:	16:10:53	BIC	2240.814
Sample:	02-01-1980	HQIC	2232.302
	- 12-01-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.2073	0.283	0.733	0.463	-0.347	0.761
ma.L1	-1.6953	0.378	-4.479	0.000	-2.437	-0.953
ma.L2	0.4026	0.713	0.565	0.572	-0.995	1.800
ma.L3	0.2948	0.348	0.848	0.397	-0.387	0.977
sigma2	1.374e+06	8.22e-07	1.67e+12	0.000	1.37e+06	1.37e+06

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	10.64
Prob(Q):	0.99	Prob(JB):	0.00
Heteroskedasticity (H):	2.71	Skew:	0.38
Prob(H) (two-sided):	0.00	Kurtosis:	4.18

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 4.16e+28. Standard errors may be unstable.

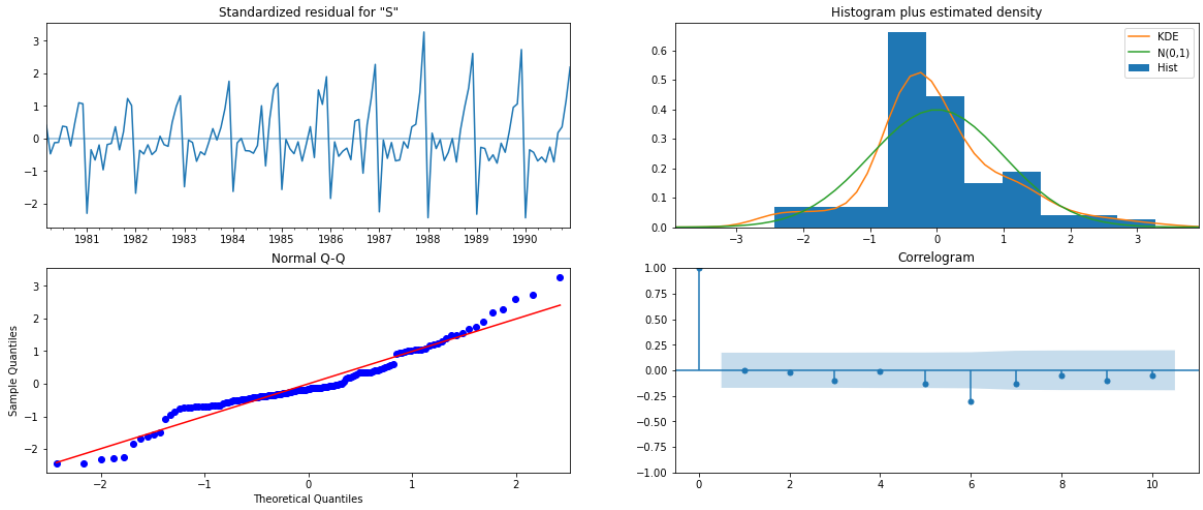

Figure 67:Sparkling Automated ARIMA diagnostics

The diagnostics look good here.

RMSE of Automated ARIMA model on Test data is: 2763.74

SARIMA Automated Train Model

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

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

Autocorrelation Function (ACF)

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 Autocorrelation Function (PACF)

A plot of partial auto-correlation for different values of lags is called PACF.

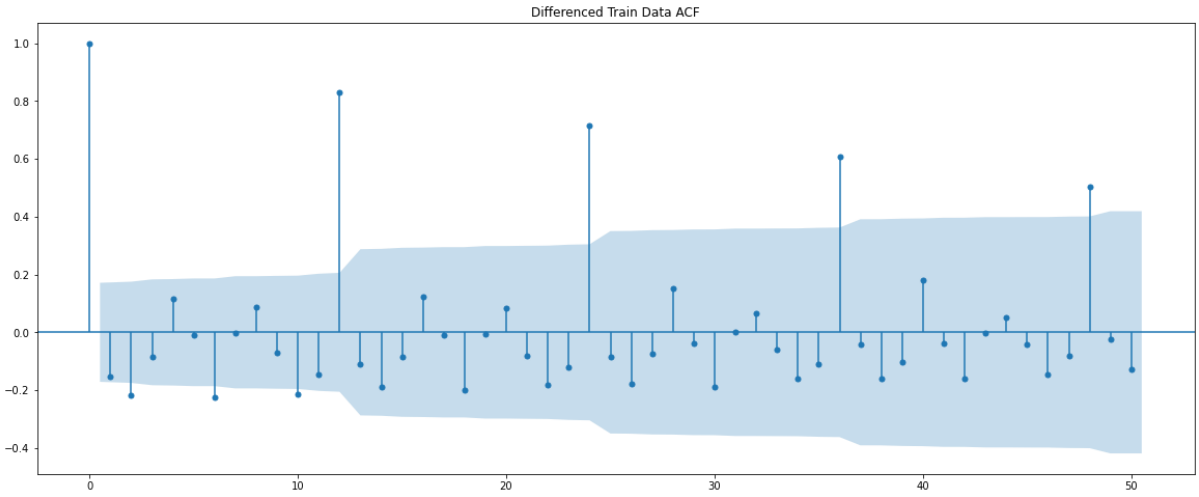

Figure 68:Sparkling-Automated SARIMA-ACF

The model has the parameter 'seasonal order' which has its values in the form of (P, D, Q, s):

P: Seasonal autoregressive order.

D: Seasonal difference order.

Q: Seasonal moving average order.

s: The number of time steps for a single seasonal period.

's' is determined from acf plot

The P,Q value is taken from 0 to 2. D is as 0 and 1. 'd' is taken as 1. From the above ACF plot we can say that ,Seasonality after every 12th lag is visible. We will run our auto SARIMA models by setting seasonality as 12. SARIMA is built using stationary data after dropping its NA values since it reduces the AIC value.

	param	seasonal	AIC
95	(1, 1, 2)	(0, 1, 2, 12)	1375.249142
155	(2, 1, 2)	(1, 1, 2, 12)	1377.868484
107	(1, 1, 2)	(2, 1, 2, 12)	1378.240131
53	(0, 1, 2)	(2, 1, 2, 12)	1378.379011
161	(2, 1, 2)	(2, 1, 2, 12)	1379.843553

The parameters in the first row has the least AIC so its taken to build the SARIMA model. P is taken as 1, d as 1, q as 1, P as 0, D as 1, Q as 2 and s as 12.

SARIMA is built using stationary data after dropping its NA values since it reduces the AIC value.

enforce_stationarity→Whether or not to transform the AR parameters to enforce stationarity in the autoregressive component of the model.

enforce_invertibility→Whether or not to transform the MA parameters to enforce invertibility in the moving average component of the model.

enforce_stationarity and enforce_ invertibility is given as false.

The SARIMA model is built with those values and RMSE is calculated on test data.

The Summary of SARIMA model built is:

SARIMAX Results

Dep. Variable:	Sparkling	No. Observations:	131
Model:	SARIMAX(1, 1, 2)x(0, 1, 2, 12)	Log Likelihood	-681.625
Date:	Sun, 05 Jun 2022	AIC	1375.249
Time:	16:36:51	BIC	1390.314
Sample:	02-01-1980	HQIC	1381.327
	- 12-01-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.1996	0.139	1.435	0.151	-0.073	0.472
ma.L1	-1.9384	0.067	-28.901	0.000	-2.070	-1.807
ma.L2	0.9429	0.068	13.955	0.000	0.810	1.075
ma.S.L12	-0.4224	0.086	-4.930	0.000	-0.590	-0.254
ma.S.L24	-0.0223	0.150	-0.149	0.882	-0.316	0.272
sigma2	1.8e+05	2.42e+04	7.434	0.000	1.33e+05	2.28e+05

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	33.78
Prob(Q):	0.99	Prob(JB):	0.00
Heteroskedasticity (H):	0.83	Skew:	0.89
Prob(H) (two-sided):	0.60	Kurtosis:	5.40

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

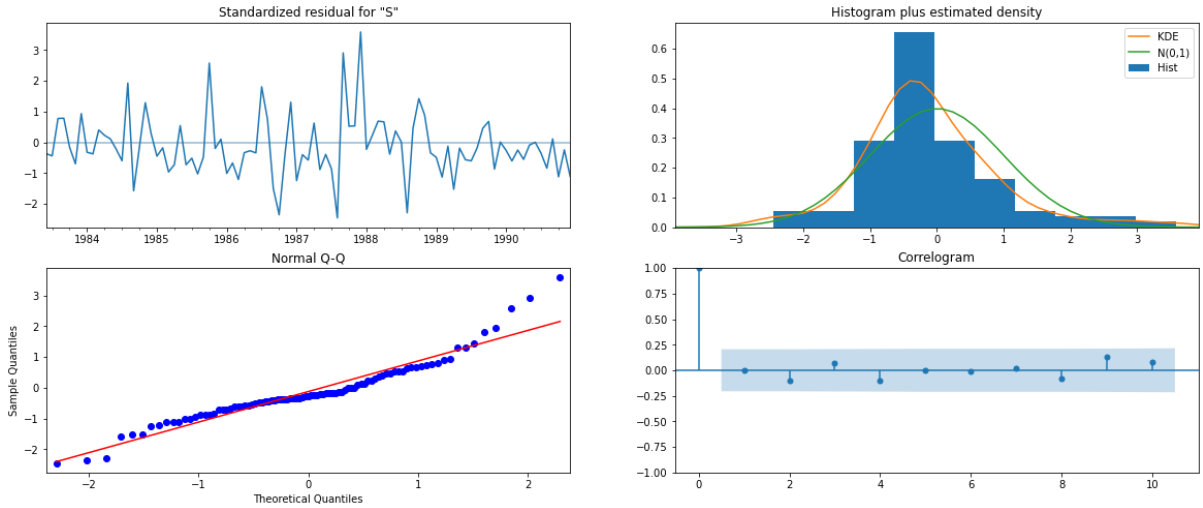

Figure 69:Sparkling-Automated SARIMA diagnostics

The diagnostics look good here.

RMSE of Automated SARIMA model on Test data is: 2866.01

2.7. 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.

Manual ARIMA Model

ACF is used for identifying the value of q and PACF is used for identifying the value of p.

The p value of the Augmented Dickey-Fuller Test on Train data: 0.669744

'p' is not less than 0.05 so the data isnt stationary.

The p value of the Augmented Dickey-Fuller Test on Train first difference data: 2.280104355826159e-12.

The data after first difference is stationary as the p value is less than 0.05. Thus we can consider the value of d as 1.

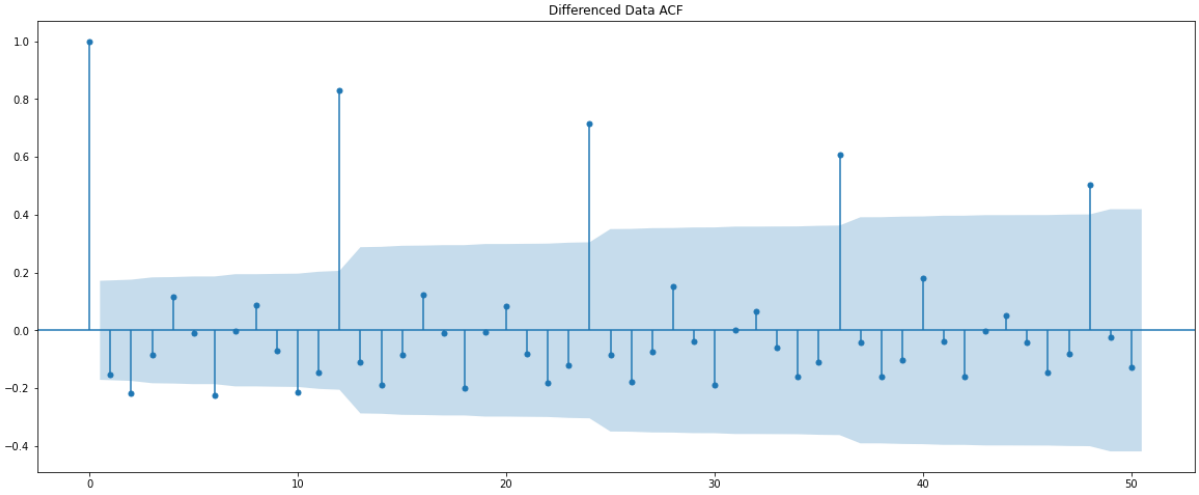

Figure 70:Sparkling-Manual ARIMA ACF

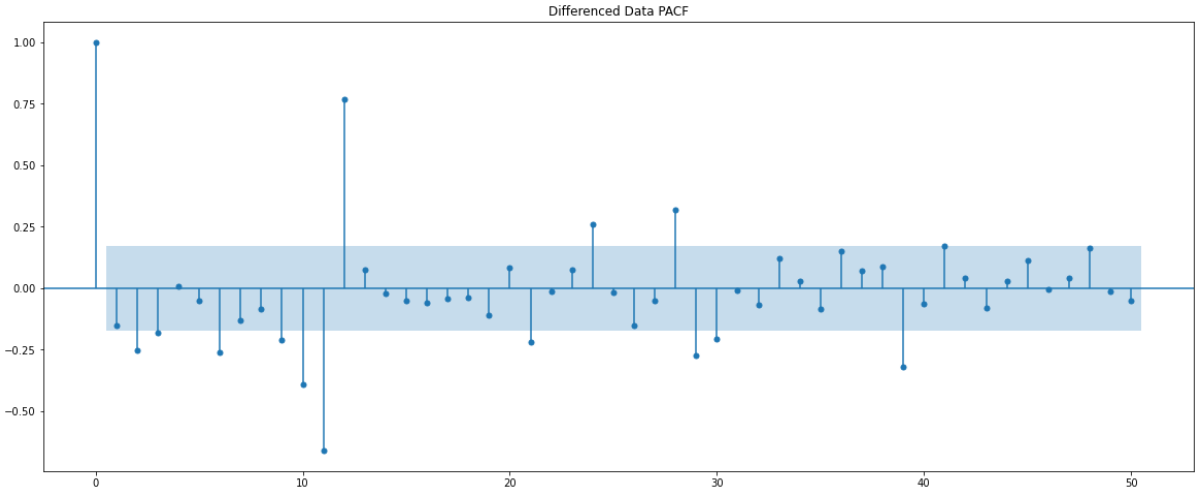

Figure 71:Sparkling-Manual ARIMA-PACF

From the ACF and PACF models we can take the value of p and q as 0. Since difference of order 1 has been taken on the data to make it stationary ,d=1.

enforce_stationarity→Whether or not to transform the AR parameters to enforce stationarity in the autoregressive component of the model.

enforce_invertibility→Whether or not to transform the MA parameters to enforce invertibility in the moving average component of the model.

enforce_stationarity and enforce_ invertibility is given as false.

ARIMA is built using stationary data after dropping its NA values .The ARIMA model is built with those values and RMSE is calculated on test data.

The summary of Manual ARIMA model is:

SARIMAX Results

Dep. Variable:	Sparkling	No. Observations:	131
Model:	ARIMA(0, 1, 0)	Log Likelihood	-1178.525
Date:	Sun, 05 Jun 2022	AIC	2359.049
Time:	17:04:25	BIC	2361.917
Sample:	02-01-1980 - 12-01-1990	HQIC	2360.215
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
sigma2	4.35e+06	3.19e+05	13.642	0.000	3.72e+06	4.97e+06

Ljung-Box (L1) (Q):	29.52	Jarque-Bera (JB):	101.79
Prob(Q):	0.00	Prob(JB):	0.00
Heteroskedasticity (H):	2.57	Skew:	-1.15
Prob(H) (two-sided):	0.00	Kurtosis:	6.67

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

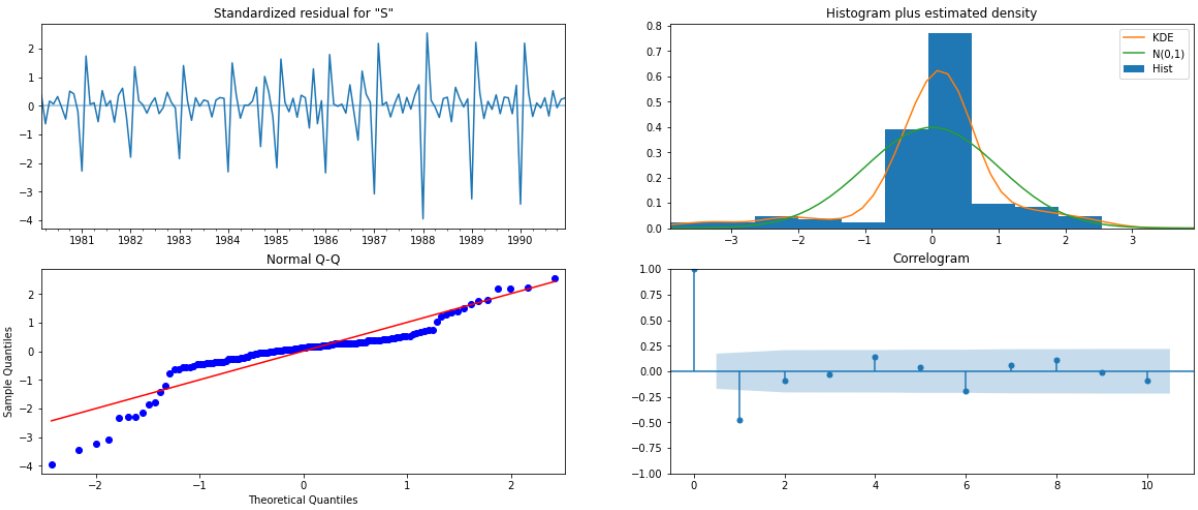

Figure 72:Sparkling-Manual ARIMA diagnostics

The diagnostics is okay here.

RMSE of Manual ARIMA model on Test data is: 1425.85

Manual SARIMA Model:

Since the seasonality parameter is 12 we can plot the graph for the difference of order 12 with NA values dropped.

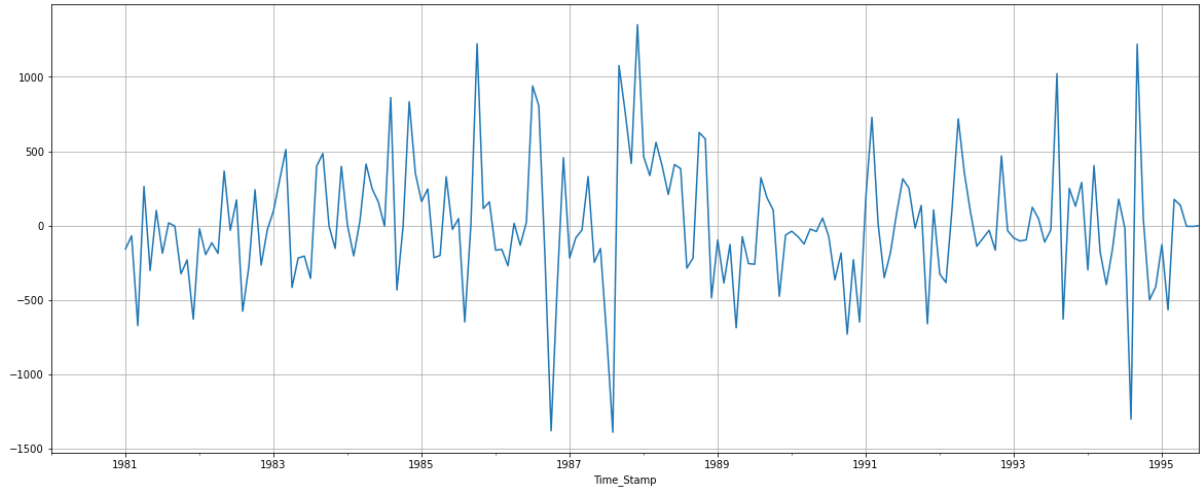

Figure 73:Sparkling-Manual SARIMA plot 1

As there is a slight trend in the graph we can take a differencing of first order on the seasonally differenced series .

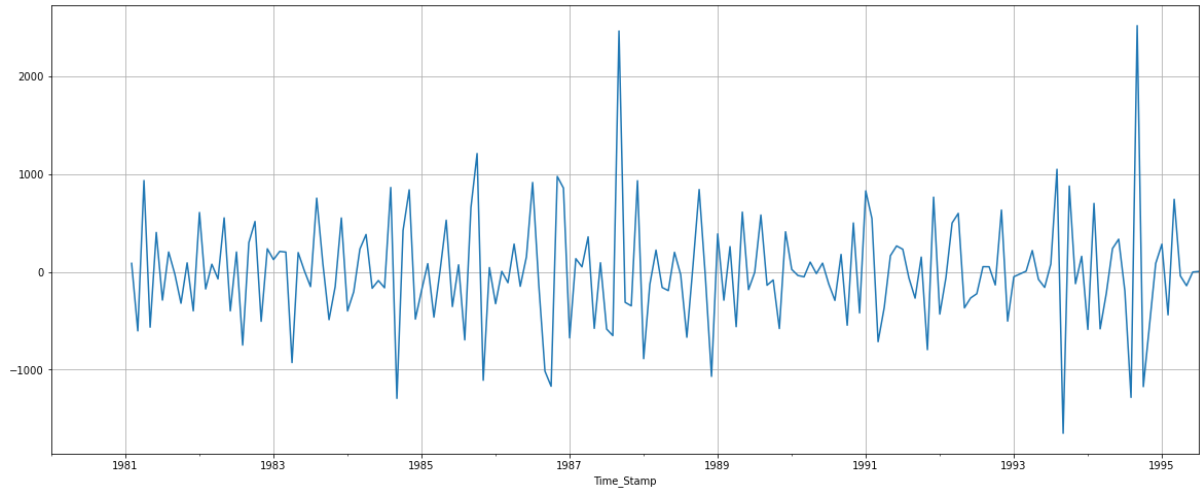

Figure 74:Sparkling-Manual SARIMA plot 2

Now we see that there is almost no trend present in the data. Seasonality is only present in the data.

Let us go ahead and check the stationarity of the above series before fitting the SARIMA model.

```
Results of Augmented Dickey-Fuller Test:
Test Statistic                  -3.342905
p-value                          0.013066
#Lags Used                      10.000000
Number of Observations Used    108.000000
Critical Value (1%)             -3.492401
Critical Value (5%)             -2.888697
Critical Value (10%)            -2.581255
dtype: float64
```

The first difference of seasonal differenced train data is used to plot ACF and PACF as below:

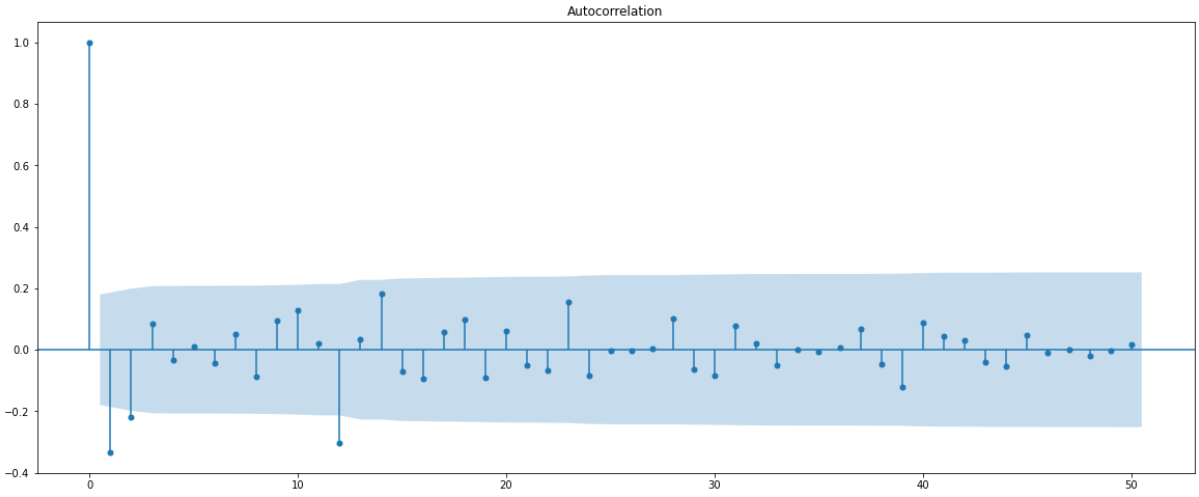

Figure 75:Sparkling-Manual SARIMA ACF

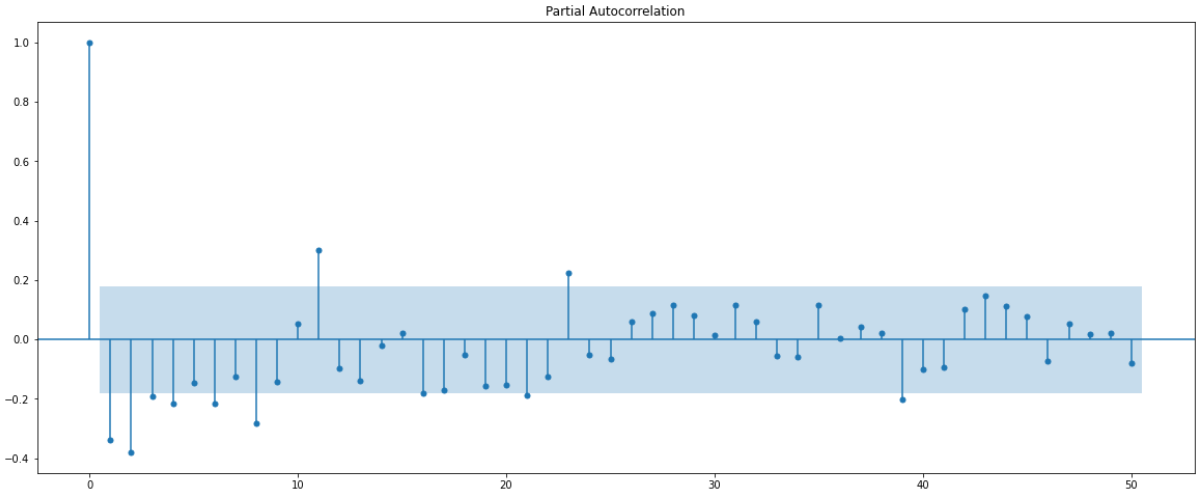

Figure 76:Sparkling-Manual SARIMA PACF

Here, we have taken alpha=0.05.

We are going to take the seasonal period as 12. We will keep the p(1) and q(1) 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.

By looking at the plots we see that the ACF and the PACF do not directly cut-off to 0.

Hence P=4 and Q=2. As we have taken a differencing of first order on the seasonally differenced series D is taken as 1.

‘p’,’q’,’d’ has the same value as the one calculated in the ARIMA model.

Order=(0,1,0) and Seasonal_order=(4,1,2,12) are the parameters to be passed to the SARIMA model along with the difference data after dropping NA values.

enforce_stationarity→Whether or not to transform the AR parameters to enforce stationarity in the autoregressive component of the model.

enforce_invertibility→Whether or not to transform the MA parameters to enforce invertibility in the moving average component of the model.

enforce_stationarity and enforce_ invertibility is given as false.

The model is built with these values and RMSE is calculated.

SARIMAX Results

Dep. Variable:	y	No. Observations:	131
Model:	SARIMAX(0, 1, 0)x(4, 1, [1, 2], 12)	Log Likelihood	-573.025
Date:	Sun, 05 Jun 2022	AIC	1160.050
Time:	19:00:01	BIC	1175.790
Sample:	0	HQIC	1166.302
	- 131		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.S.L12	-0.3749	0.462	-0.811	0.417	-1.281	0.531
ar.S.L24	0.4354	0.441	0.988	0.323	-0.428	1.299
ar.S.L36	0.0473	0.189	0.251	0.802	-0.322	0.417
ar.S.L48	-0.0134	0.143	-0.094	0.925	-0.293	0.267
ma.S.L12	0.1119	1.448	0.077	0.938	-2.725	2.949
ma.S.L24	-0.7546	1.336	-0.565	0.572	-3.373	1.864
sigma2	7.069e+05	8.03e+05	0.880	0.379	-8.68e+05	2.28e+06

Ljung-Box (L1) (Q):	20.58	Jarque-Bera (JB):	6.11
Prob(Q):	0.00	Prob(JB):	0.05
Heteroskedasticity (H):	0.39	Skew:	-0.20
Prob(H) (two-sided):	0.03	Kurtosis:	4.39

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

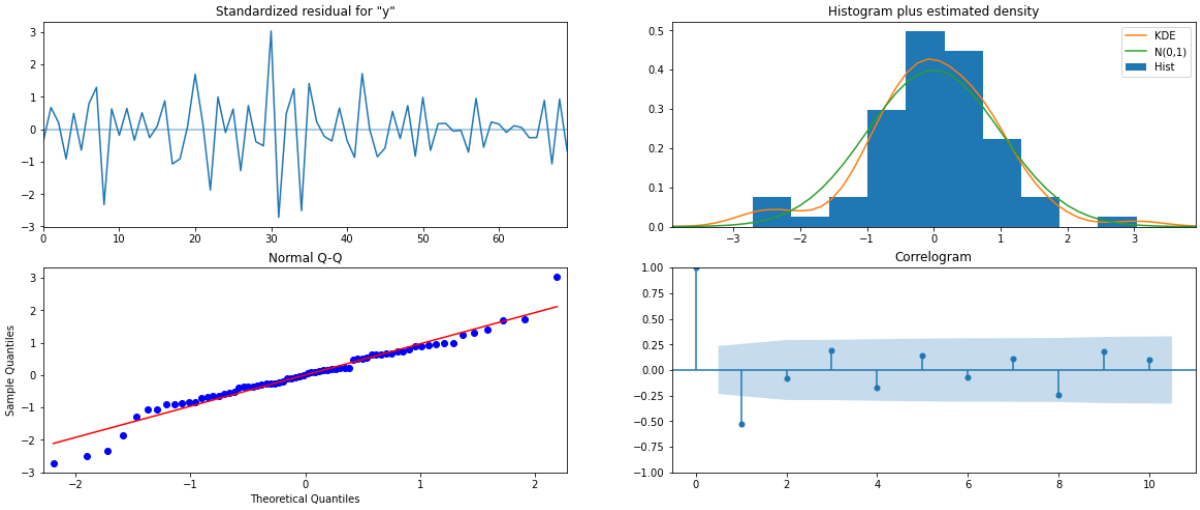

Figure 77:Sparkling-Manual SARIMA diagnostics

The diagnostics look good here.

RMSE of Manual SARIMA model on Test data is: 3437.38

2.8.Build a table (create a data frame) with all the models built along with their corresponding parameters and the respective RMSE values on the test data.

The model with its parameter and its corresponding RMSE value is shown below in ascending order of RMSE value from which we can see that Iterative Triple Exponential Smoothing with additive trend and Multiplicative Seasonality has the least RMSE value. Hence it's the optimum model.

	Test RMSE
Alpha=0.4,Beta=0.1,Gamma=0.2,TES Iterative Multiplicative Seas	317.43
Alpha=0.1,Beta=0.4,Gamma=0.1,TES Iterative Additive Seas	342.93
Alpha=0.11,Beta=0.01,Gamma=0.46:TES additive	378.95
Alpha=0.11,Beta=0.049,Gamma=0.36:TES Multiplicative	404.29
SimpleAverageModel	1275.08
12pointTrailingMovingAverage	1275.16
Alpha=0.07,SES	1338.01
Alpha=0.1 SES Iterative	1375.39
RegressionOnTime	1389.14
ARIMA Manual(0,1,0)	1425.85
6pointTrailingMovingAverage	1521.34
Alpha=0.1 , Beta=0.1,DES Iterative	1777.73
3pointTrailingMovingAverage	2443.00
ARIMA Automated(1,1,3)	2763.74
SARIMA Automated (1,1,2)(0,1,2,12)	2866.01
2pointTrailingMovingAverage	3046.52
SARIMA Manual(0,1,0)(4,1,2,12)	3437.38
NaiveModel	3864.28
Alpha=0.66,Beta=0.0001:DES	5291.88

Table 18:Sparkling-Model-RMSE

2.9.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

The optimum model Iterative Triple Exponential Smoothing with additive trend and Multiplicative Seasonality is built and forecasted for 12 months into the future.
The paramters are as below:
{'smoothing_level': 0.4, 'smoothing_trend': 0.1,
'smoothing_seasonal': 0.2, 'damping_trend': nan, 'initial_level':
2356.541666666665, 'initial_trend': -9.181060606060463,
'initial_seasons': array([0.71166877, 0.67309316, 0.81943184,
0.78429538, 0.63424785,
0.63175794, 0.82647725, 1.0318111 , 0.89263071, 1.1231428 ,
1.69872589, 2.17271729]), 'use_boxcox': False, 'lamda': None,
'remove_bias': False}

The forecast is shown with 95% confidence interval band.

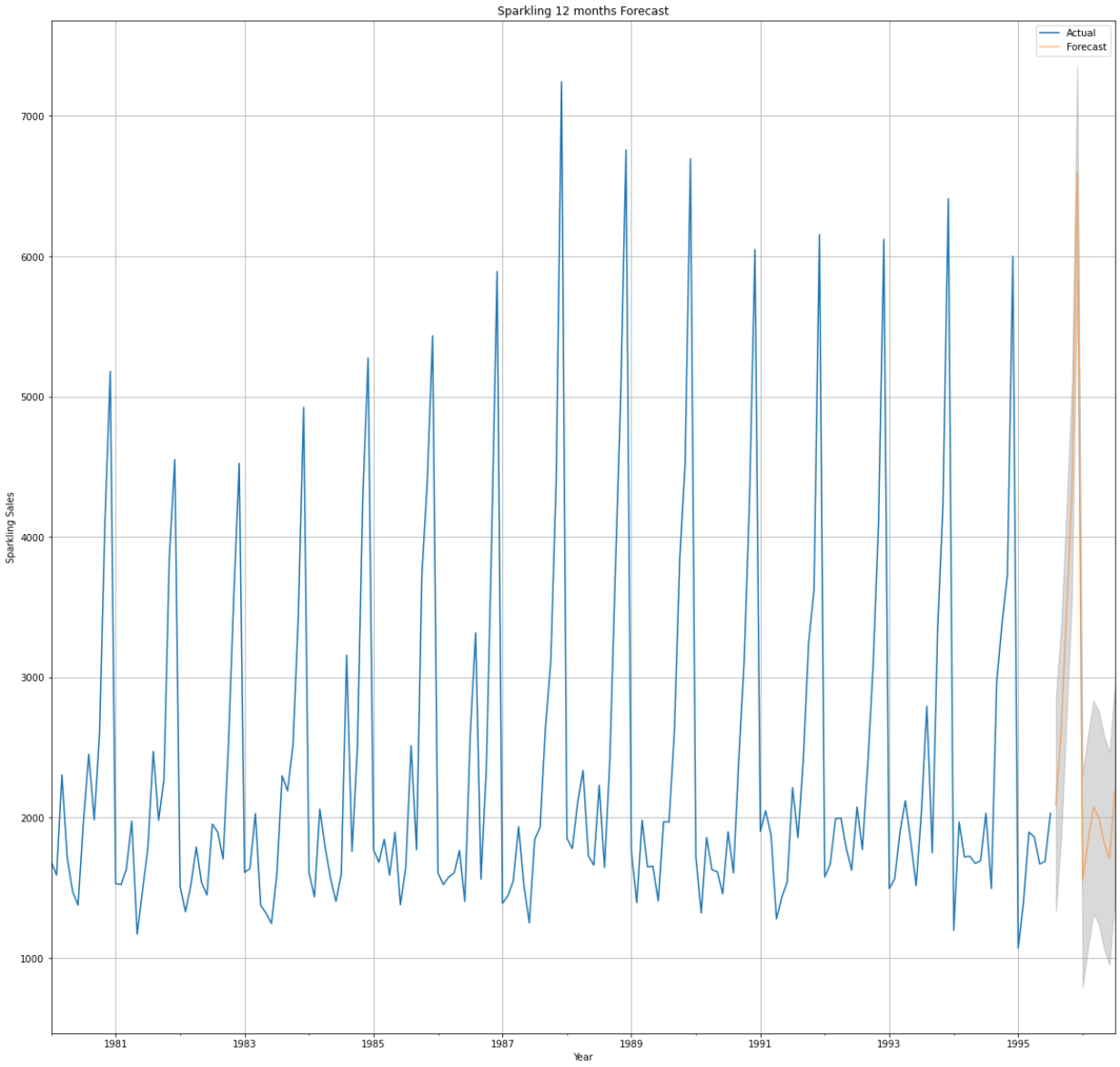

Figure 78:Sparkling Optimum Model Plot

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

1. First ,read the data as a time series and plotted it on a graph to show how sales for Sparkling wines over the years.

2. Then performed some exploratory data analyses on the data sets, creating various types of charts for analyze the sales.

3. I split the data into test(data from the year 1991) and train(data before the year 1991).

4. Next I built the following models :
- Simple Exponential Smoothing Model
- Iterative Simple Exponential Smoothing Model
- Double Exponential Smoothing Model
- Iterative Double Exponential Smoothing Model
- Triple Exponential Smoothing Model
- Iterative Triple Exponential Smoothing Model.
- Linear Regression Model
- Naïve Approach
- Simple Average Model
- Moving Average Model

For all the above models RMSE value was calculated to understand the performance.

5. The stationarity of the data was checked by stating hypothesis for statistical testing and using ADF Test.

6. From here, we build ARIMA and SARIMA models, but first we examine the dataset. If the series is not stationary, we take the first difference of the series and converted into a stationary series.

7. The ARIMA/SARIMA models are built using AIC scores, we select the parameter with the least AIC and the model is built with it. RMSE is also calculated to check the performance.

8. The ARIMA/SARIMA models are built manually by calculating value of p,q,P,Q,s,d,D from ACF , PACF graph. RMSE is also calculated to check the performance.

9. Finally, we take the model with minimum RMSE value and build the most optimum model on the complete data .The sales for the next 12 months in future with 95% confidence intervals is predicted.

Recommendations

--> Fourth quarter has the highes sales among other quarter. So the company can stock up the wines in the second quarter itself to prepare themselves to supply the high demand in the fourth quarter.

--> Proper branding advertising in leading newspaper and magazines can be done. Social Media Advertising can also be done to improve the sales in the first 3 quarters.

--> Second quarter has the lowest sales . So coupons and differs can be offered to boost up the sales.

--> The quality and the taste of the wines can be improved.

--> Further information like age group,location of the customers can be analyzed to improve the model performance and get a better understanding of the Sparkling wine sales.

----------X----------X---------X----------