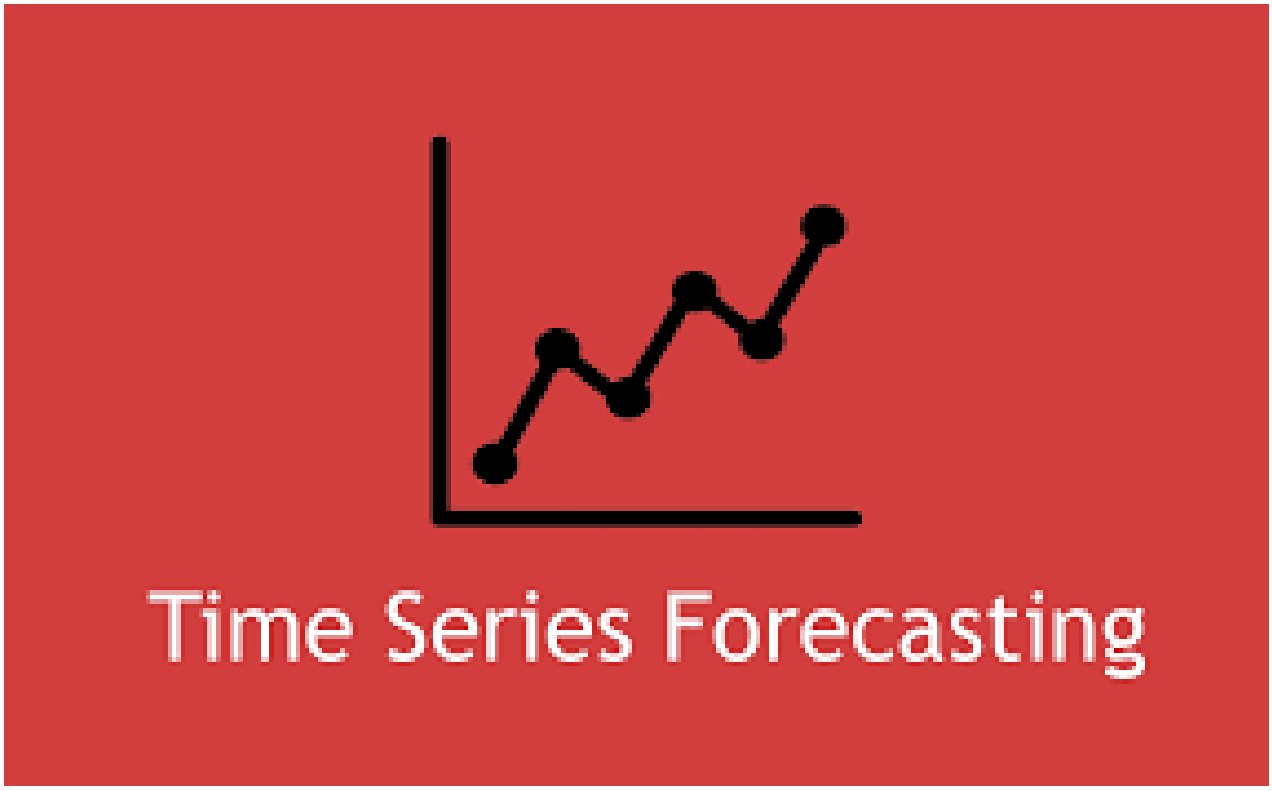
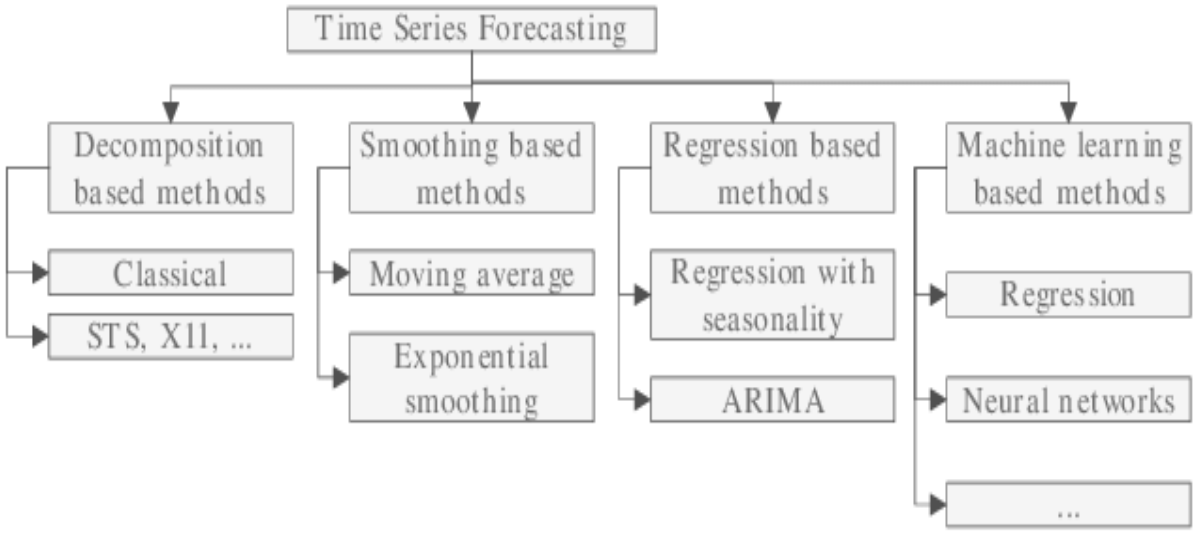
PARAKANDLA. MANISHA – PGP DSBA ONLINE

PROJECT: TIME SERIES FORECASTING (20-02-2022)





This Business report will provide the detailed explanation of how we performed analysis according to the problem statement given in the assignment. It will also provide the relative resolution and explanation with regards to the problem statement.

Contents

List of Figures:

TIME SERIES ANALYSIS ON WINESALES WITH SPARKLING DATASET

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 Wine Sales in the 20th century.

Data set for the Problem: **Sparkling.csv**

First, we import all the necessary libraries such as seaborn, pandas, sklearn etc to perform our analysis.
Next, we import the dataset **Sparkling.csv**

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

When we read the data using pandas read_csv function, the sample of the data looks like:

	YearMonth	Sparkling
0	1980-01	1686
1	1980-02	1591
2	1980-03	2304
3	1980-04	1712
4	1980-05	1471

We need to transform this data into appropriate time series data using pandas date_range function with (start='1/1/1980', end='8/1/1995', freq='M').

After that, I have created timestamp for this data and made timestamp as index for the data and dropped the YearMonth column. Now, the data looks like:

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

SPARKLING WINE YEAR WISE SALE:

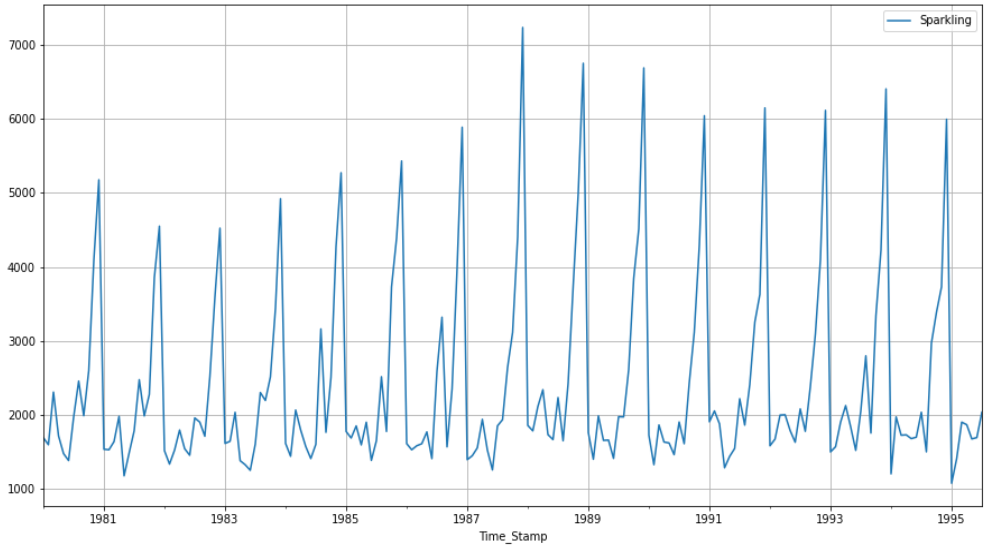


Fig1. Sparkling year wise sales

→We observe that, Sparkling wine sales show no much trend in the yearly sale.

→The seasonality seems to have a pattern on yearly basis.

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

The Shape of the dataset is (187,1).

1.There are 187 observations which represent the monthly sales of respective wines from the year 1980 to July 1995.

2.The data has two variables the YearMonth of sales and the sales for the respective month of the year.

3.There are no null values present in the data.

4.Checking the info of the data:

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 187 entries, 1980-01-31 to 1995-07-31
Data columns (total 1 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Sparkling  187 non-null    int64
dtypes: int64(1)
memory usage: 2.9 KB
```

5.Description of the data:

	Sparkling
count	187.000000
mean	2402.417112
std	1295.111540
min	1070.000000
25%	1605.000000
50%	1874.000000
75%	2549.000000
max	7242.000000

Boxplot for yearly/Monthly sales for rose and sparkling wine:

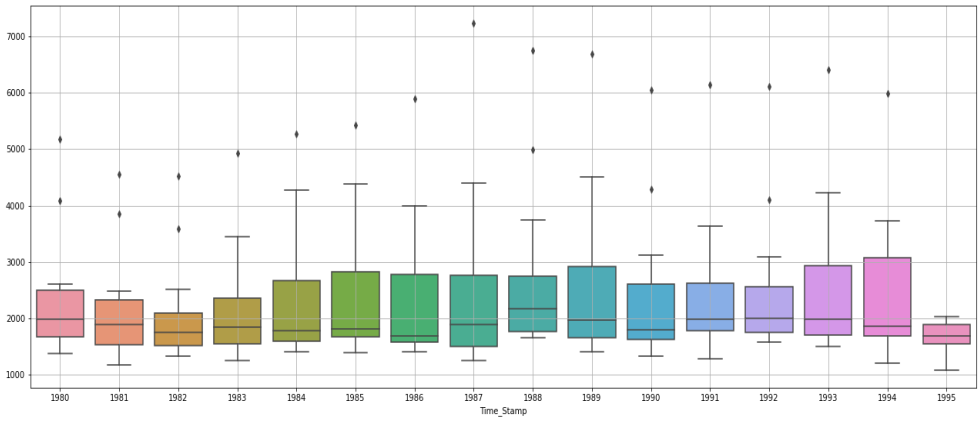


Fig2. Boxplot for yearly sales

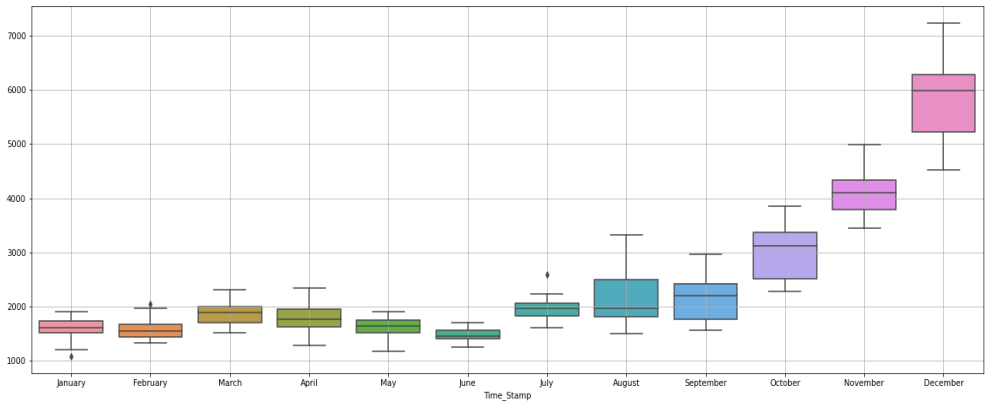


Fig3. Boxplot for Monthly sales

For the yearly Boxplot, we can see that the boxplots do not indicate any trend. The sales of sparkling wine having outliers for almost all the years except 1955.

For the Monthly Boxplot, we can observe that there is an increase in the sale. The sale of December month has the highest value. There are only few outliers present in the Monthly Boxplot.

Cumulative % and Month on Month % sales plots of Sparkling wine:

Time series month plot to understand the spread of Sales across different years and within different months across years.

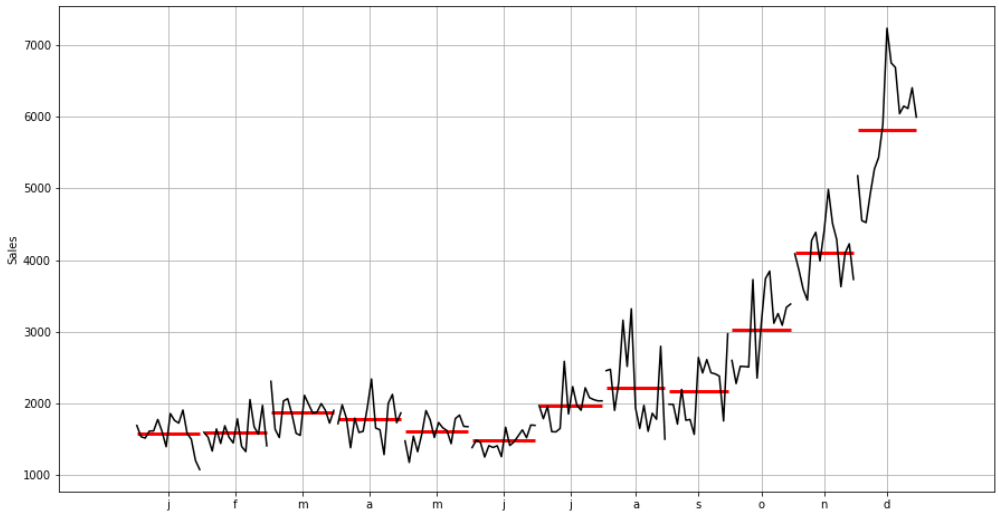


Fig4.Timeseries month plot for spread of sales

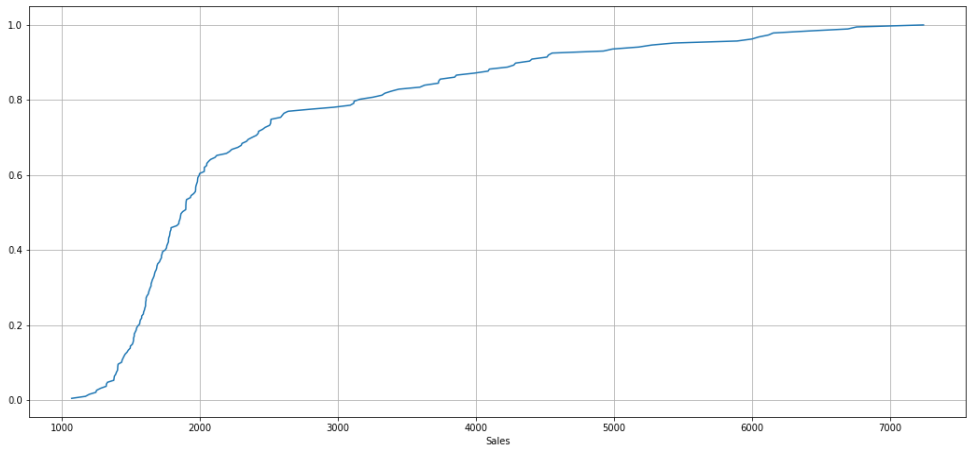


Fig5. Empirical Cumulative Distribution Curve

The ECD curve tells us what percentage of data points refer to what number of sales.

Line plot for monthly sales for Sparkling wine:

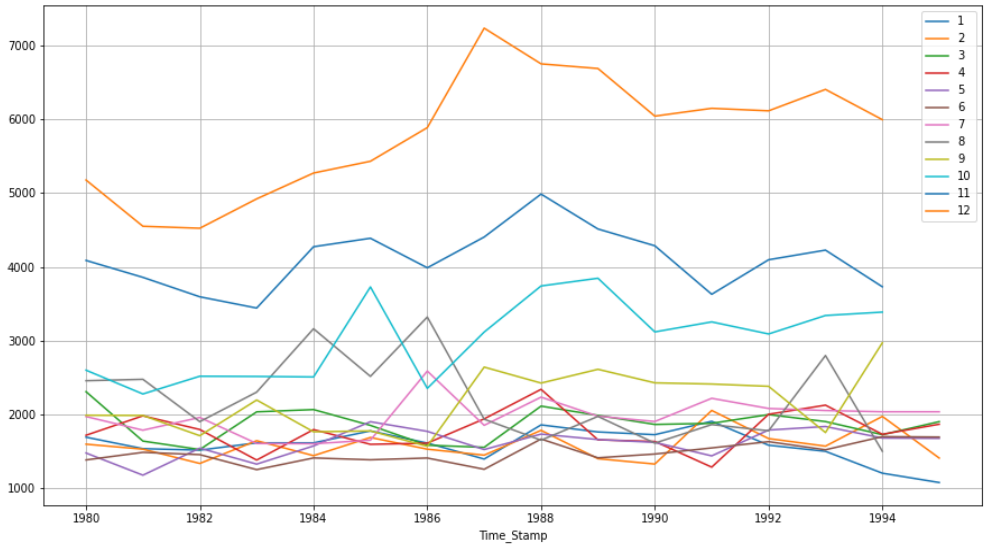
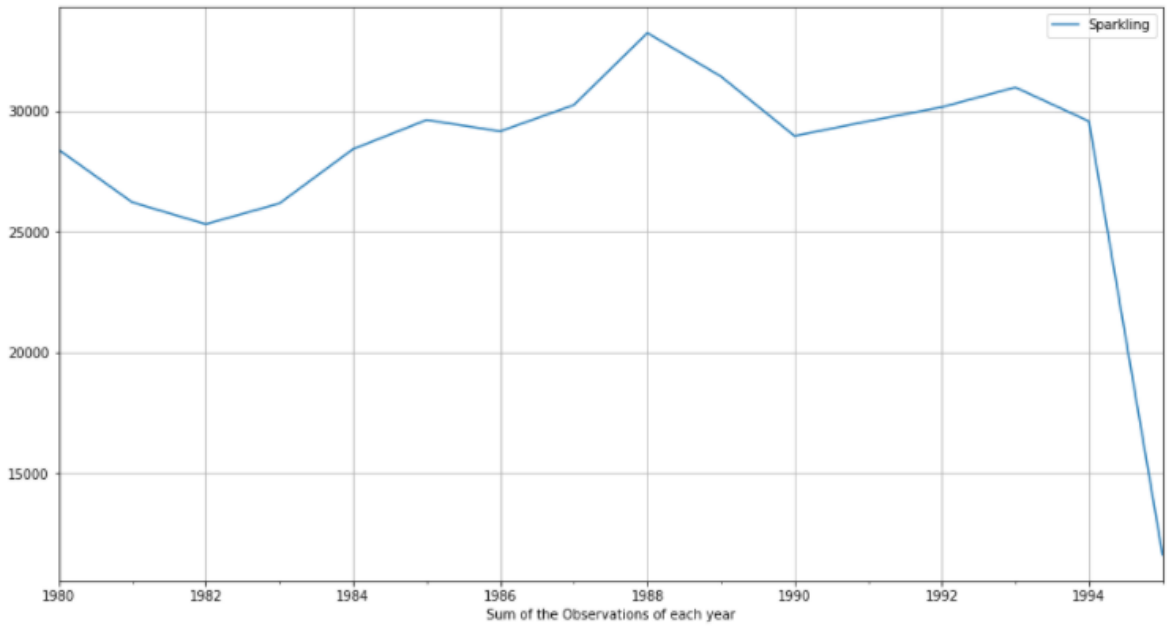


Fig6. Line plot for monthly sales

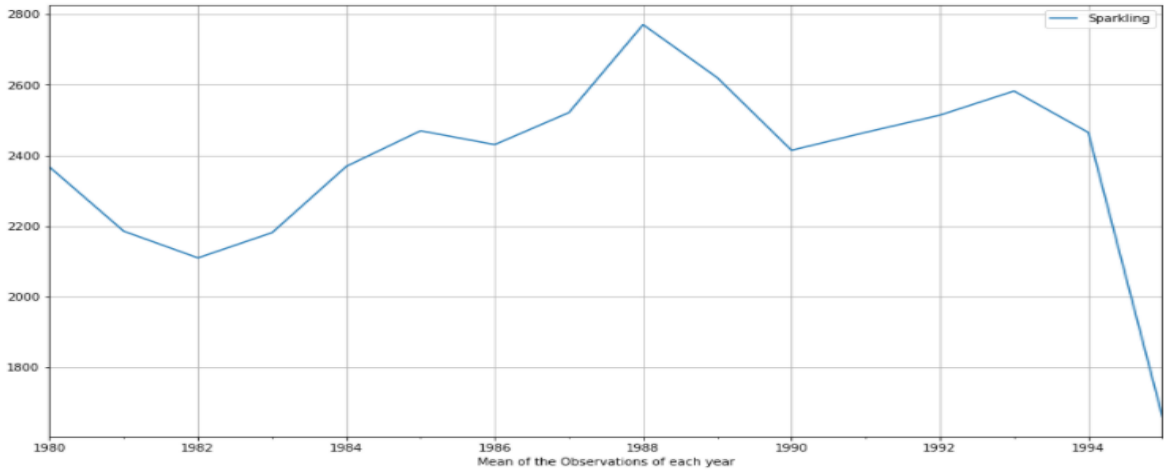
We can observe that, the line plot for monthly sales of sparkling wine shows that the December month the highest sale and August, January, June and February show lower sale values.

Read the monthly data into a quarterly and yearly format. Compare the time series plot and draw inferences.

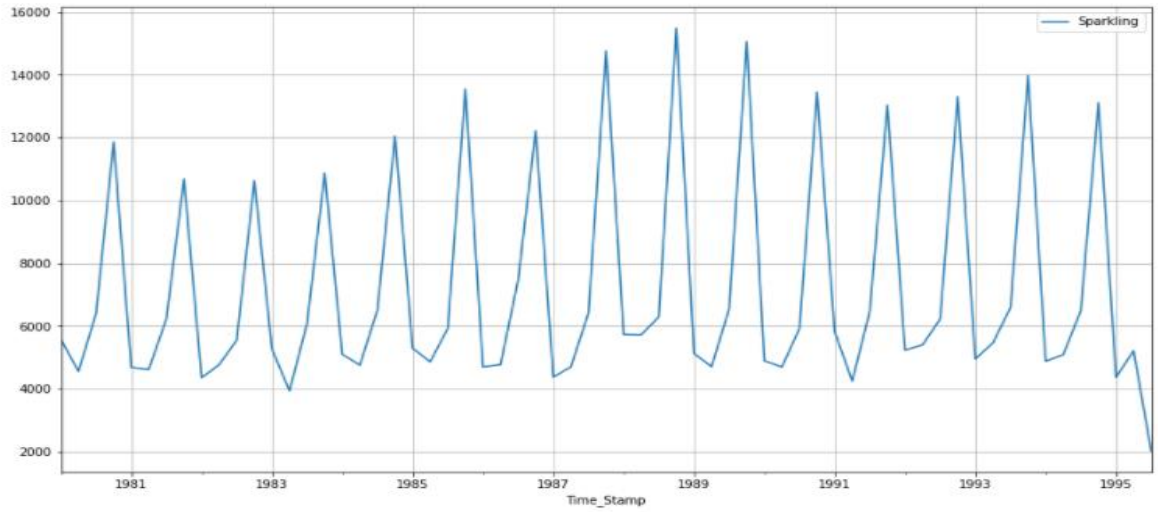
Sum of observations of each year:

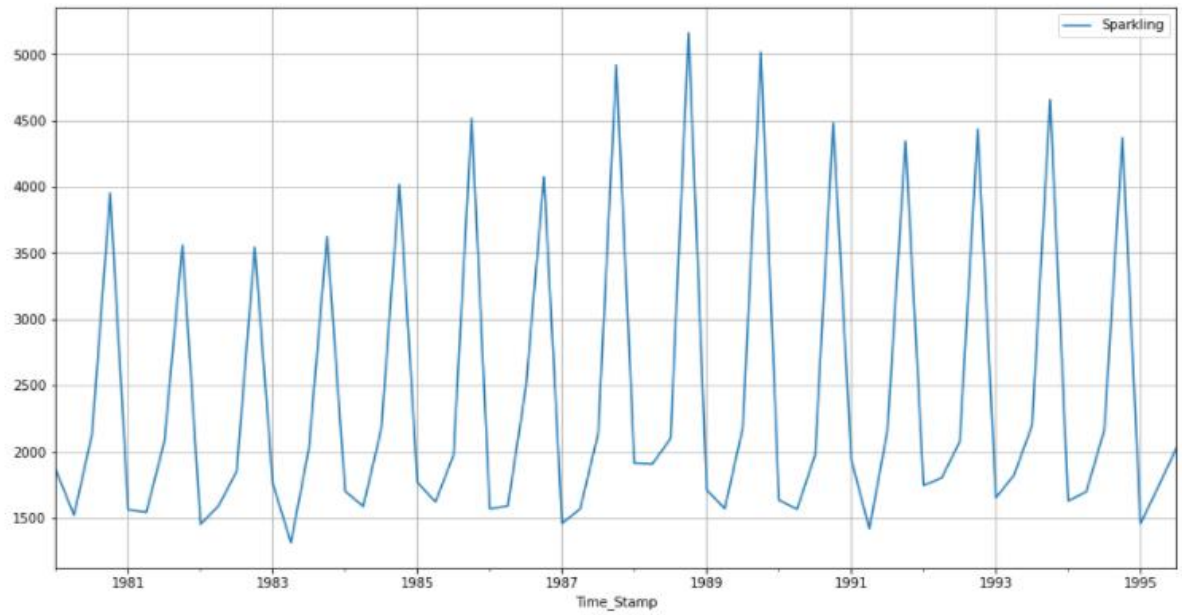


Mean observations of each year:

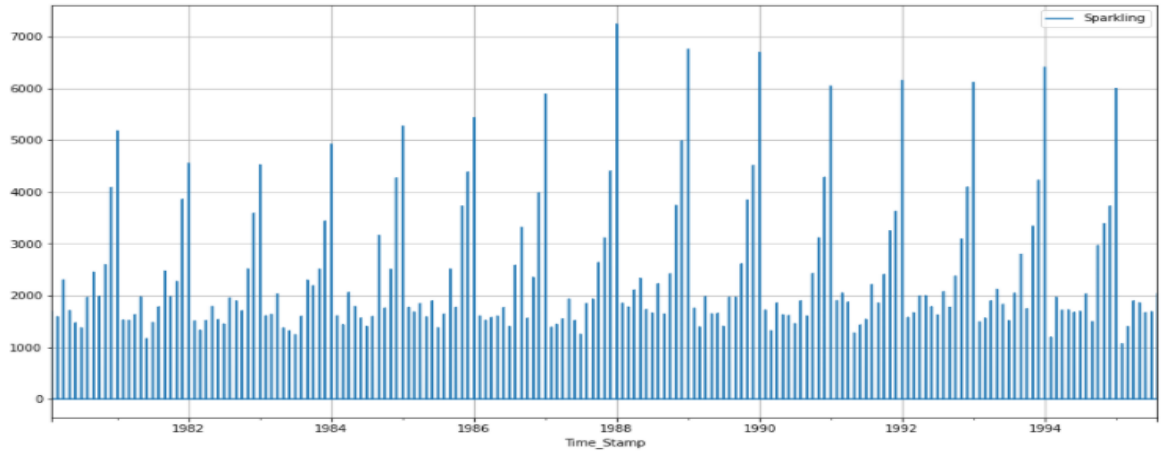


Quarterly plots:

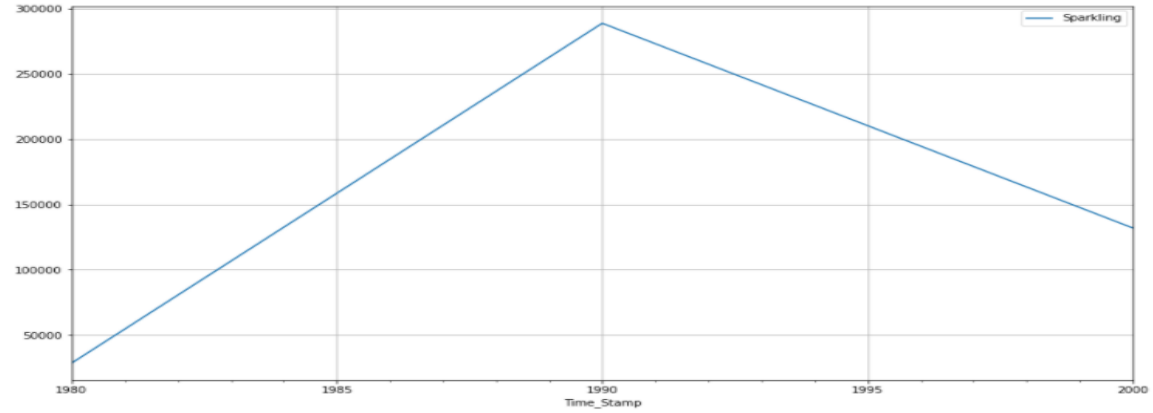




Daily plot:



Decade Plot:



Average sales & Percentage change of Sales with respect to time:

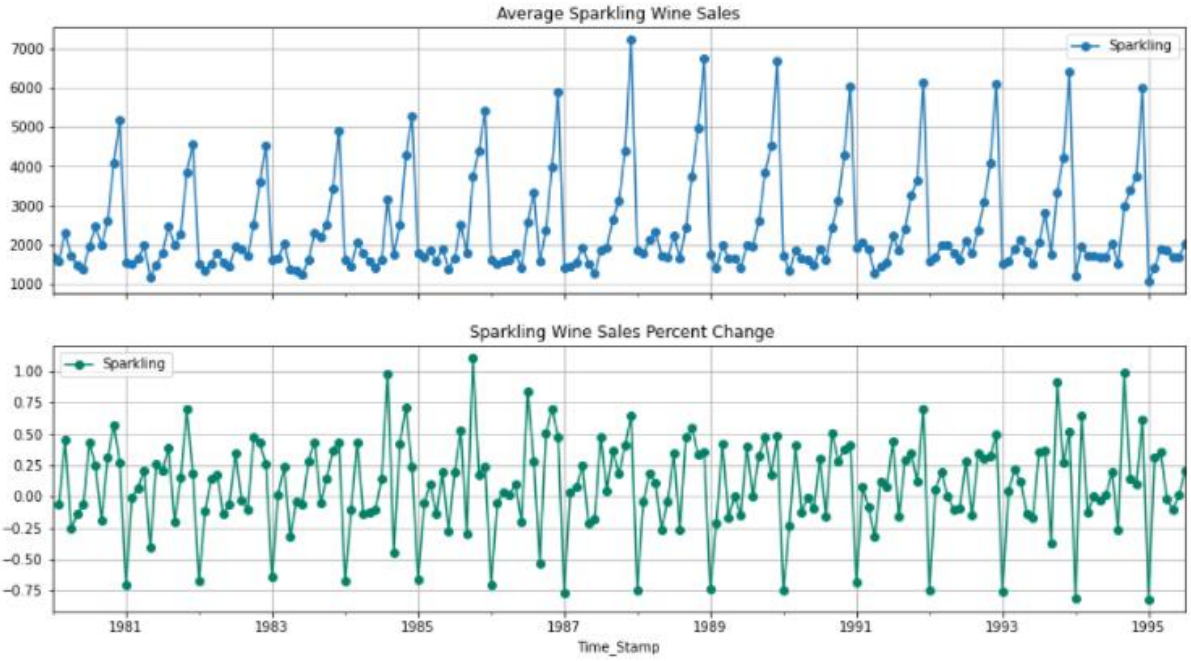


Fig7. Average sales & Percentage change of Sales

The median values are stable from January to June and has an increasing trend from July to December. January to December months. The Average sales value does not show a trend.

Additive Decomposition of Sparkling wine sales:

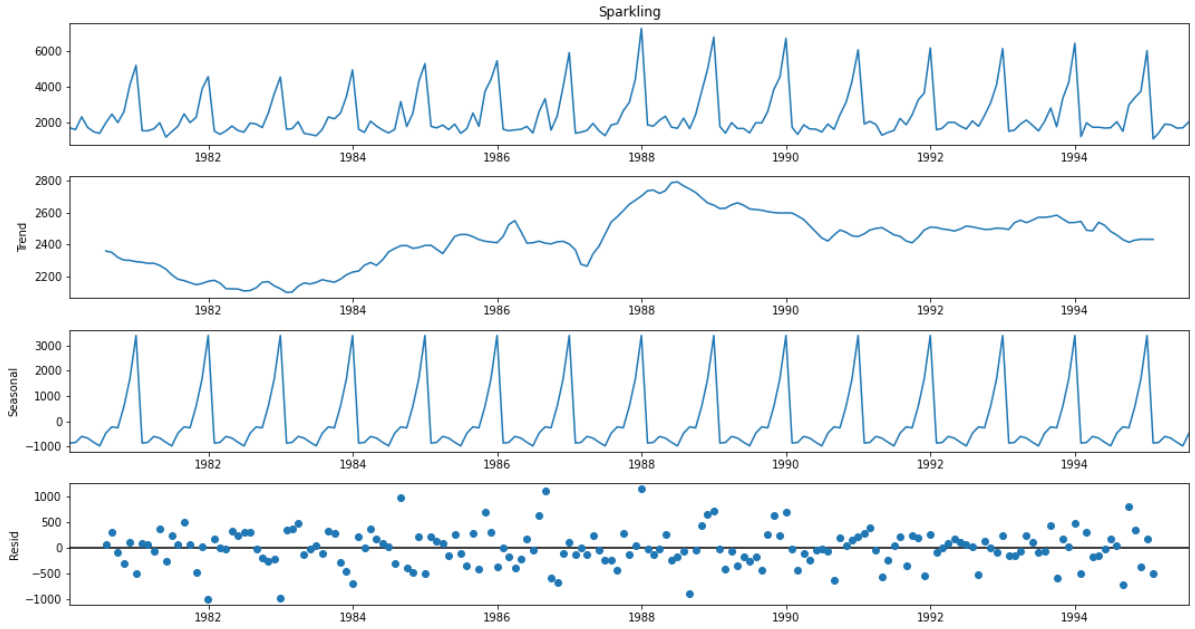


Fig8. Additive Decomposition

We can see that the residuals are located around 0 from the plot of the residuals in the decomposition.

Multiplicative Decomposition of Sparkling wine sales:

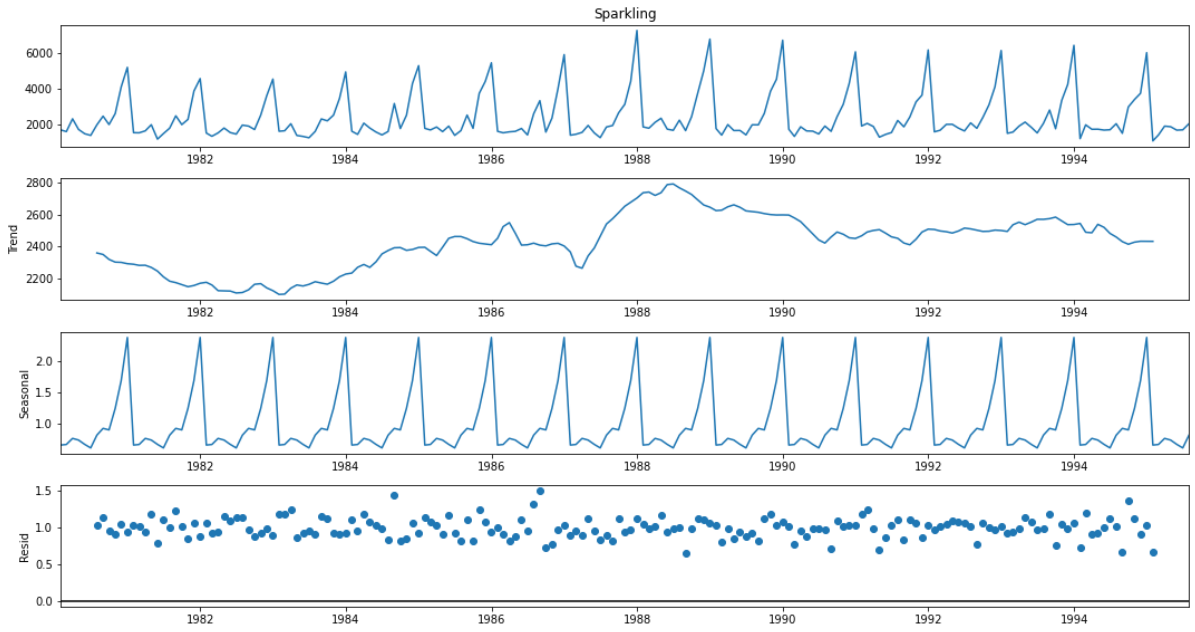


Fig9. Multiplicative Decomposition

For Multiplicative decomposition, we can see that a lot of residuals are located around Trend

Time_Stamp

1980-01-31 NaN
1980-02-29 NaN
1980-03-31 NaN
1980-04-30 NaN
1980-05-31 NaN
1980-06-30 NaN
1980-07-31 2360.666667
1980-08-31 2351.333333
1980-09-30 2320.541667
1980-10-31 2303.583333
1980-11-30 2302.041667
1980-12-31 2293.791667
Name: trend, dtype: float64

Seasonality
Time_Stamp
1980-01-31 -854.260599
1980-02-29 -830.350678
1980-03-31 -592.356630
1980-04-30 -658.490559
1980-05-31 -824.416154
1980-06-30 -967.434011
1980-07-31 -465.502265
1980-08-31 -214.332821
1980-09-30 -254.677265
1980-10-31 599.769957
1980-11-30 1675.067179
1980-12-31 3386.983846
Name: seasonal, dtype: float64

Residual
Time_Stamp
1980-01-31 NaN
1980-02-29 NaN
1980-03-31 NaN
1980-04-30 NaN
1980-05-31 NaN
1980-06-30 NaN
1980-07-31 70.835599
1980-08-31 315.999487
1980-09-30 -81.864401
1980-10-31 -307.353290
1980-11-30 109.891154

1980-12-31 -501.775513
Name: resid, dtype: float64

Summary of Sparkling:
→Sparkling dataset doesn't show a visible trend however it shows seasonality, also if observed from additive decomposition the residual is catching some pattern.
→Multiplicative decomposition on the other hand seems to dictate on the series as the scale of the residual plot had decreased considerably.
→Monthly bar plots showed that the sales are higher towards the last months than earlier.

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

The train data of Sparkling has been split up to the year 1990 and has 132 data points.

The test data has been split from the year 1991 a 3. Split the data into training and test. The test data should start in 1991 and has 55 data points.

From train-test split we will be predicting the future sales in comparison with past years' sale.

Shape of train data: (132,1)

Shape of test data: (55,1)

First/last few rows of training and testing data:

First few rows of Training Data Sparkling
Time_Stamp
1980-01-31 1686
1980-02-29 1591
1980-03-31 2304
1980-04-30 1712
1980-05-31 1471

Last few rows of Training Data Sparkling
Time_Stamp
1990-08-31 1605
1990-09-30 2424
1990-10-31 3116
1990-11-30 4286
1990-12-31 6047

First few rows of Test Data Sparkling
Time_Stamp
1991-01-31 1902
1991-02-28 2049
1991-03-31 1874
1991-04-30 1279
1991-05-31 1432

Last few rows of Test Data Sparkling

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

Plot for train-test data:

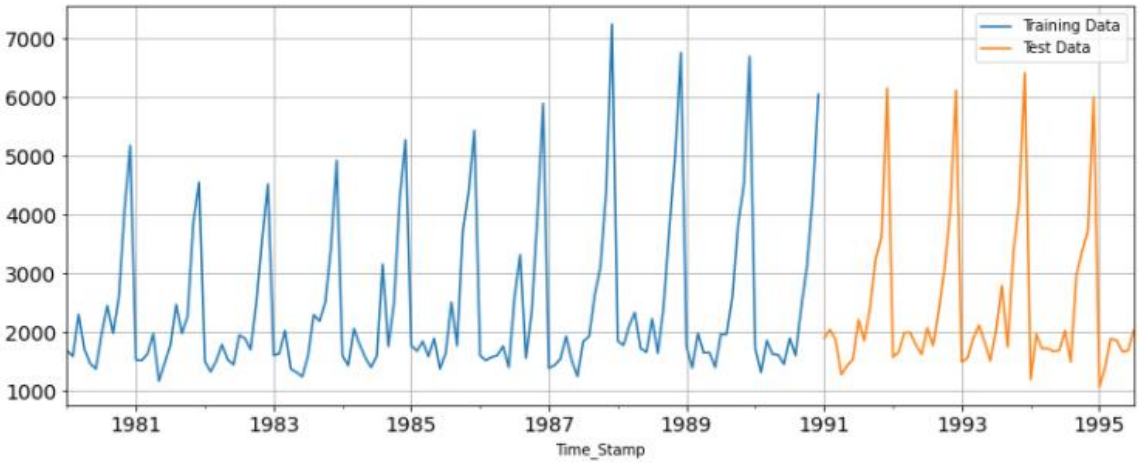


Fig10. Train-test split

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

MODEL1: LINEAR REGRESSION

For Linear Regression, we regress the sales variable against the order of the occurrence.

Then we generate the numerical time instance order for both train and test set.

For linear regression the equation will be y=a+b(time)

```
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, 4
1, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 6
0, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 7
9, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 9
8, 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, 12
9, 130, 131, 132]
Test Time instance
 [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, 8
1, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97]
```

We will add these values in the training and test set.

Hence, the train and test are thus modified to perform Linear Regression.

First few rows of Training Data Sparkling time

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

Last few rows of Training Data Sparkling time

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

First few rows of Test Data Sparkling time

Time_Stamp		
1991-01-31	1902	43
1991-02-28	2049	44
1991-03-31	1874	45
1991-04-30	1279	46
1991-05-31	1432	47

Last few rows of Test Data Sparkling time

Time_Stamp		
1995-03-31	1897	93
1995-04-30	1862	94
1995-05-31	1670	95
1995-06-30	1688	96
1995-07-31	2031	97

Predictions on test data:

```
[array([2266.71282822, 2272.54548672, 2278.37814522, 2284.21080373,
       2290.04346223, 2295.87612073, 2301.70877923, 2307.54143774,
       2313.37409624, 2319.20675474, 2325.03941325, 2330.87207175,
       2336.70473025, 2342.53738875, 2348.37004726, 2354.20270576,
       2360.03536426, 2365.86802276, 2371.70068127, 2377.53333977,
       2383.36599827, 2389.19865677, 2395.03131528, 2400.86397378,
       2406.69663228, 2412.52929078, 2418.36194929, 2424.19460779,
       2430.02726629, 2435.85992479, 2441.6925833 , 2447.5252418 ,
       2453.3579003 , 2459.19055881, 2465.02321731, 2470.85587581,
       2476.68853431, 2482.52119282, 2488.35385132, 2494.18650982,
       2500.01916832, 2505.85182683, 2511.68448533, 2517.51714383,
       2523.34980233, 2529.18246084, 2535.01511934, 2540.84777784,
       2546.68043634, 2552.51309485, 2558.34575335, 2564.17841185,
       2570.01107035, 2575.84372886, 2581.67638736])]
```

Predictions on train data:

```
[array([2021.74117111, 2027.57382961, 2033.40648811, 2039.23914662,
       2045.07180512, 2050.90446362, 2056.73712213, 2062.56978063,
       2068.40243913, 2074.23509763, 2080.06775614, 2085.90041464,
       2091.73307314, 2097.56573164, 2103.39839015, 2109.23104865,
       2115.06370715, 2120.89636565, 2126.72902416, 2132.56168266,
       2138.39434116, 2144.22699966, 2150.05965817, 2155.89231667,
       2161.72497517, 2167.55763367, 2173.39029218, 2179.22295068,
       2185.05560918, 2190.88826769, 2196.72092619, 2202.55358469,
       2208.38624319, 2214.2189017 , 2220.0515602 , 2225.8842187 ,
       2231.7168772 , 2237.54953571, 2243.38219421, 2249.21485271,
       2255.04751121, 2260.88016972, 2266.71282822, 2272.54548672,
       2278.37814522, 2284.21080373, 2290.04346223, 2295.87612073,
       2301.70877923, 2307.54143774, 2313.37409624, 2319.20675474,
       2325.03941325, 2330.87207175, 2336.70473025, 2342.53738875,
       2348.37004726, 2354.20270576, 2360.03536426, 2365.86802276,
       2371.70068127, 2377.53333977, 2383.36599827, 2389.19865677,
       2395.03131528, 2400.86397378, 2406.69663228, 2412.52929078,
       2418.36194929, 2424.19460779, 2430.02726629, 2435.85992479,
       2441.6925833 , 2447.5252418 , 2453.3579003 , 2459.19055881,
       2465.02321731, 2470.85587581, 2476.68853431, 2482.52119282,
       2488.35385132, 2494.18650982, 2500.01916832, 2505.85182683,
       2511.68448533, 2517.51714383, 2523.34980233, 2529.18246084,
       2535.01511934, 2540.84777784, 2546.68043634, 2552.51309485,
       2558.34575335, 2564.17841185, 2570.01107035, 2575.84372886,
       2581.67638736, 2587.50904586, 2593.34170437, 2599.17436287,
       2605.00702137, 2610.83967987, 2616.67233838, 2622.50499688,
       2628.33765538, 2634.17031388, 2640.00297239, 2645.83563089,
       2651.66828939, 2657.50094789, 2663.3336064 , 2669.1662649 ,
       2674.9989234 , 2680.8315819 , 2686.66424041, 2692.49689891,
       2698.32955741, 2704.16221591, 2709.99487442, 2715.82753292,
       2721.66019142, 2727.49284992, 2733.32550843, 2739.15816693,
       2744.99082543, 2750.82348394, 2756.65614244, 2762.48880094,
       2768.32145944, 2774.15411795, 2779.98677645, 2785.81943495])]
```

Linear Regression train test plot:

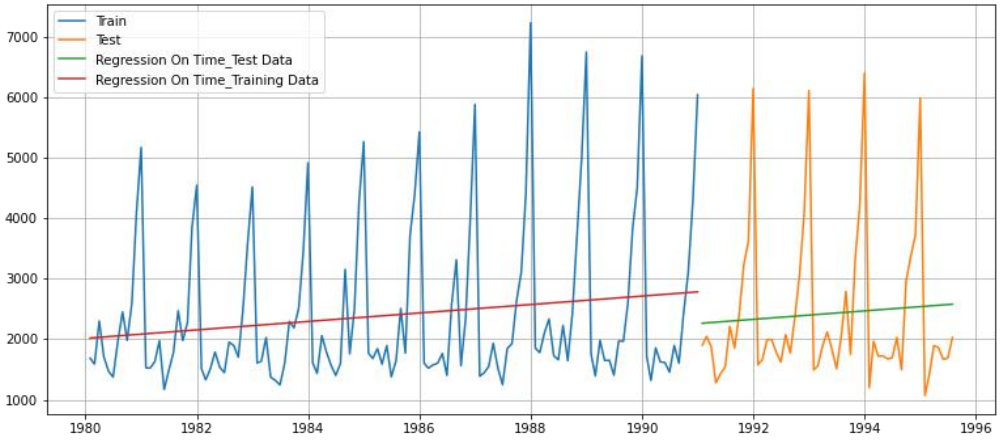


Fig11. Linear Regression train test

The line in the plot showing the upward trend for test data.

```
Regression On Time forecast on the Training Data:
RMSE is 1279.322

Regression On Time forecast on the Test Data:
RMSE is 1275.867
```

The RMSE for the linear regression model generated for test data

	Test RMSE
RegressionOnTime	1275.867052

MODEL2: NAÏVE FORECAST MODEL

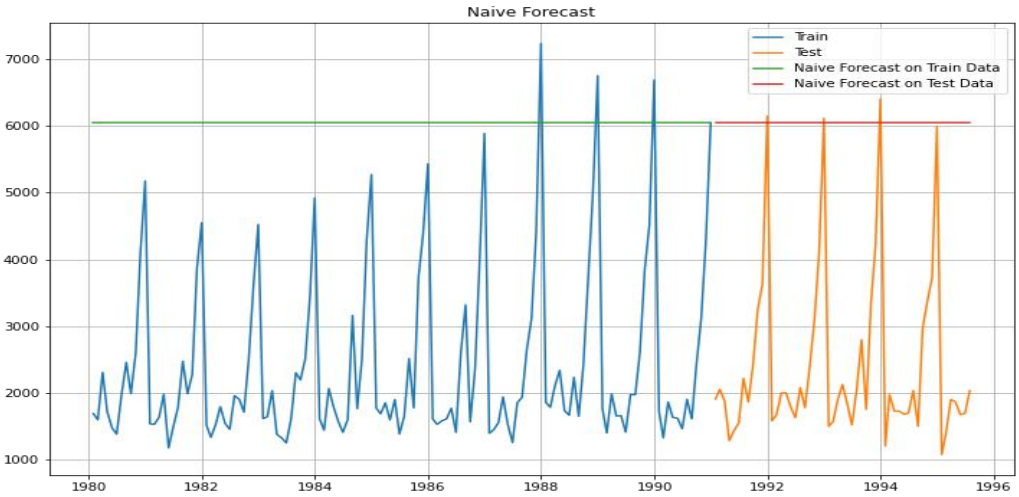


Fig12. Train-Test Naïve

For the Naïve model, we observe that the red line in the plot shows a straight line which predicts sale of tomorrow is same as today. And the prediction for day after tomorrow is same as tomorrow.

Hence, it applies to all the future years.

```
Naive Model forecast on the Training Data,
RMSE is 3867.701
```

```
Naive Model forecast on the Test Data,
RMSE is 3864.279
```

The RMSE for the Naive model generated for test data: 3864.279352

MODEL3: SIMPLE AVERAGE MODEL

In Simple Average method, we will forecast the data using the average of the training values.

From the plot below, we observe that the red line is straight and shows the simple average forecasting.

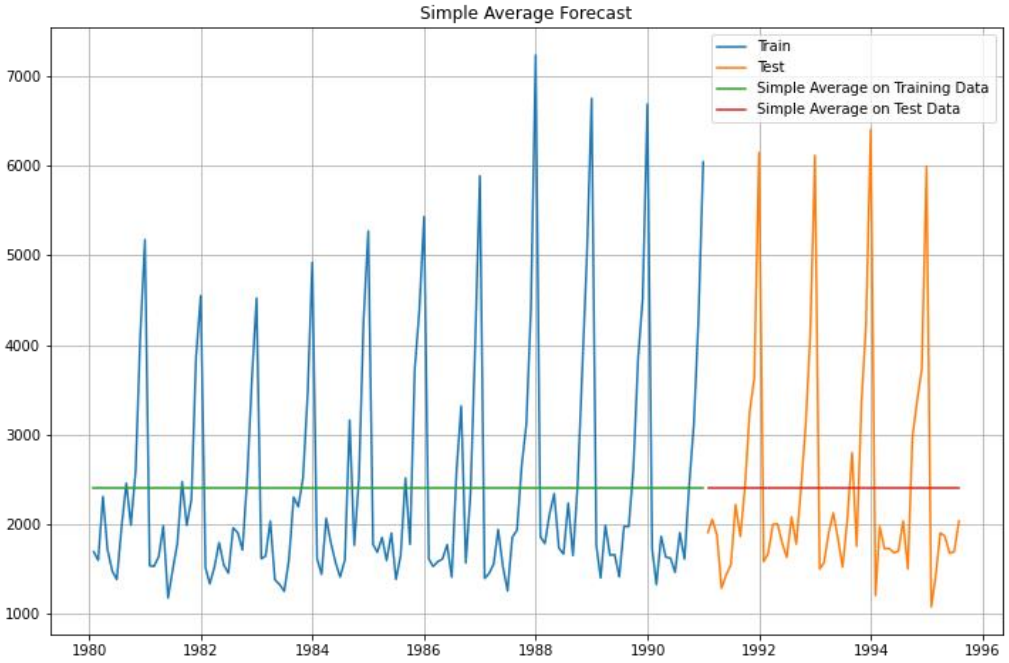


Fig13. SA-train-test

```
Simple Average Model forecast on the Training Data:
RMSE is 1298.484
```

```
Simple Average forecast on the Test Data:
RMSE is 1275.082
```

The RMSE for the Simple Average model generated for test data: 1275.081804

MODEL4: MOVING AVERAGE MODEL

In Moving Average Model, we compute moving averages for 2,4,6,9 point intervals.

Then the best interval is determined by the maximum accuracy.

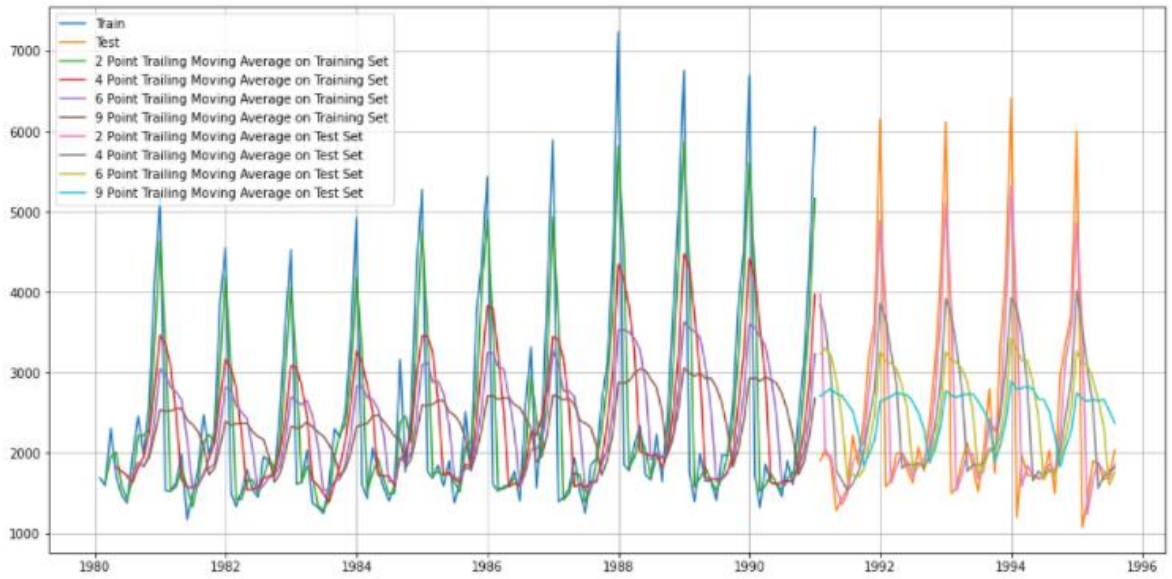


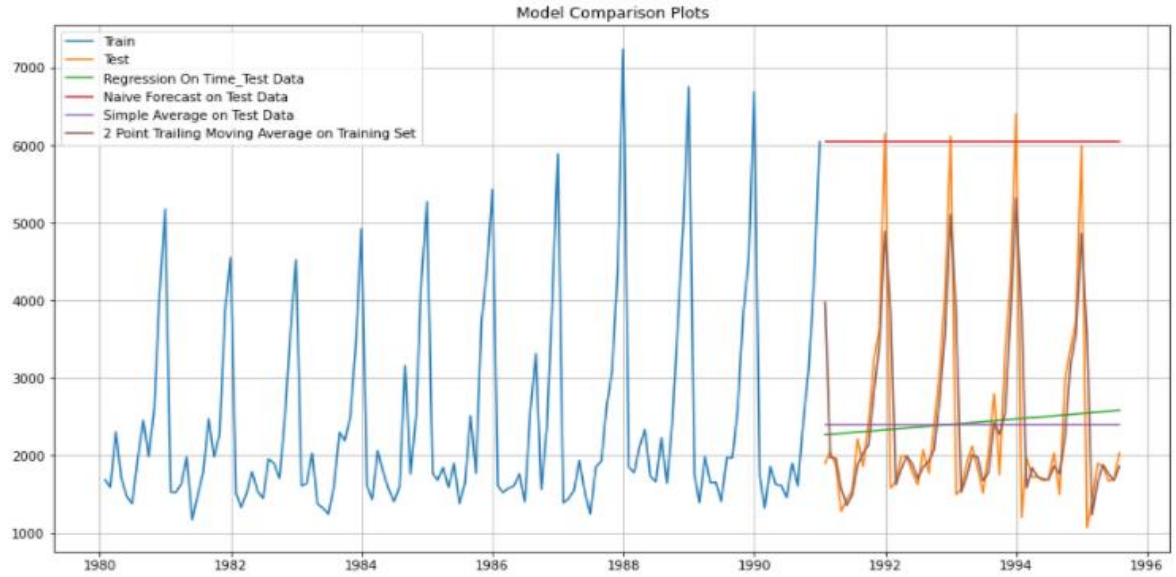
Fig14.MA Train-Test

```
2 point Moving Average Model forecast on the Testing Data:
RMSE is 813.401
4 point Moving Average Model forecast on the Testing Data:
RMSE is 1156.590
6 point Moving Average Model forecast on the Testing Data:
RMSE is 1283.927
9 point Moving Average Model forecast on the Testing Data:
RMSE is 1346.278
```

	Test RMSE
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315

From the above table, we see that 2point trailing moving average has the least score.

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



MODEL5: SIMPLE EXPONENTIAL SMOOTHING

Simple Exponential Smoothing, is **a time series forecasting method for univariate data without a trend or seasonality**. It requires a single parameter, called alpha (a), also called the smoothing factor or smoothing coefficient. This method is suitable for forecasting data with no clear trend or seasonal pattern.

It requires a single parameter, called *alpha* (*a*), also called the smoothing factor or smoothing coefficient.
This parameter controls the rate at which the influence of the observations at prior time steps decay exponentially. Alpha is often set to a value between 0 and 1. Large values mean that the model pays attention mainly to the most recent past observations, whereas smaller values mean more of the history is taken into account when making a prediction.

A value close to 1 indicates fast learning (that is, only the most recent values influence the forecasts), whereas a value close to 0 indicates slow learning (past observations have a large influence on forecasts).

Hyperparameters:

Alpha: Smoothing factor for the level.

```
                    SimpleExpSmoothing Model Results
==============================================================================
Dep. Variable:              Sparkling   No. Observations:                  132
Model:            SimpleExpSmoothing   SSE                      228338410.448
Optimized:                      True   AIC                           1899.987
Trend:                          None   BIC                           1905.753
Seasonal:                       None   AICC                          1900.302
Seasonal Periods:               None   Date:                 Sun, 20 Feb 2022
Box-Cox:                       False   Time:                         17:12:15
Box-Cox Coeff.:                 None
==============================================================================
                       coeff                 code              optimized
------------------------------------------------------------------------------
smoothing_level            0.0496066                alpha                 True
initial_level              1818.5048                  1.0                 True
------------------------------------------------------------------------------
```

Parameters:

```
{'smoothing_level': 0.04960659884563118,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 1818.5047543457245,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Predict on train:

Time_Stamp	Sparkling	predict
1980-01-31	1686	1818.504754
1980-02-29	1591	1811.931644
1980-03-31	2304	1800.971977
1980-04-30	1712	1825.925486
1980-05-31	1471	1820.274030

Predict on Test:

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

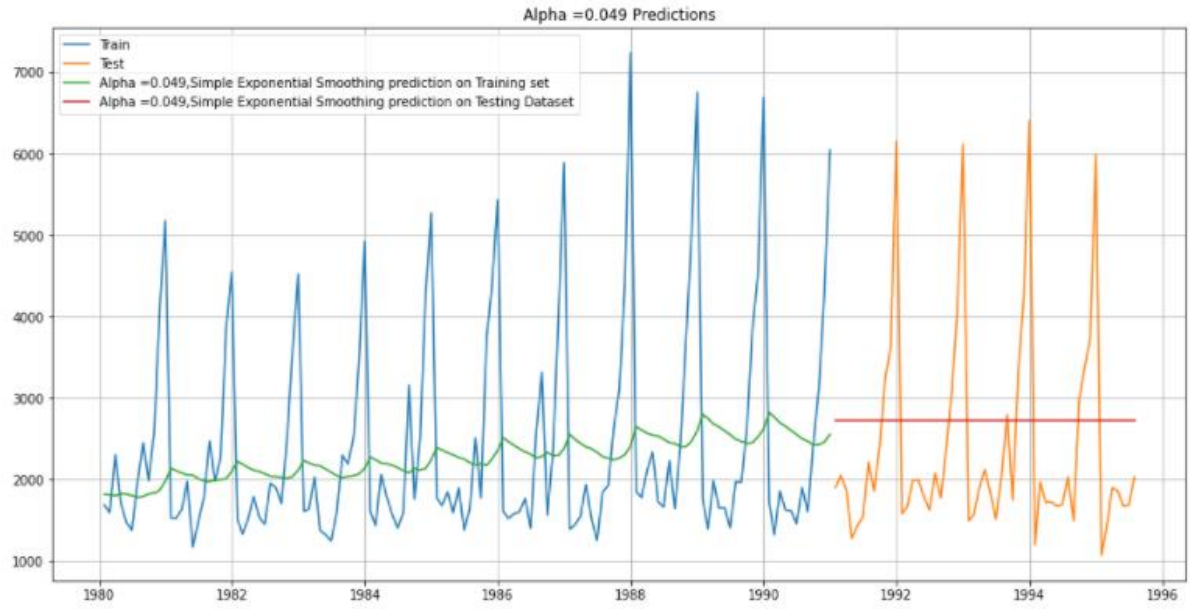


Fig15. SES Train-Test

```
Alpha =0.049 Simple Exponential Smoothing Model forecast on the Training Data,
RMSE is 1315.232
```

```
Alpha =0.049 Simple Exponential Smoothing Model forecast on the Training Data,
RMSE is 1316.035
```

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

MODEL6: SIMPLE EXPONENTIAL SMOOTHING WITH ALPHA IN RANGE OF 0.01 TO 0.1

In Simple Exponential Smoothing Model, we will run a loop with different alpha values to understand which particular value is best.

Alpha value ranges from 0.01 to 0.1

	Alpha Values	Test RMSE	Train RMSE
0	0.01	1276.251337	1302.063355
1	0.02	1283.553056	1303.192007
2	0.03	1294.721648	1305.212814
3	0.04	1305.943195	1308.368577
4	0.05	1316.543221	1312.159247
...	...	...	...
94	0.95	3778.432623	1363.586031
95	0.96	3796.048620	1365.349773
96	0.97	3813.437370	1367.179921
97	0.98	3830.602869	1369.077800
98	0.99	3847.548965	1371.044831

99 rows × 3 columns

SimpleExpSmoothing Model Results

Dep. Variable:	Sparkling	No. Observations:	132
Model:	SimpleExpSmoothing	SSE	223788705.587
Optimized:	True	AIC	1897.330
Trend:	None	BIC	1903.096
Seasonal:	None	AICC	1897.645
Seasonal Periods:	None	Date:	Sun, 20 Feb 2022
Box-Cox:	False	Time:	17:12:17
Box-Cox Coeff.:	None		

	coeff	code	optimized
smoothing_level	0.0100000	alpha	False
initial_level	2357.6819	l.0	True

The RMSE for the Simple Exponential smoothing model (Alpha=0.01) generated for test data: 1276.251337

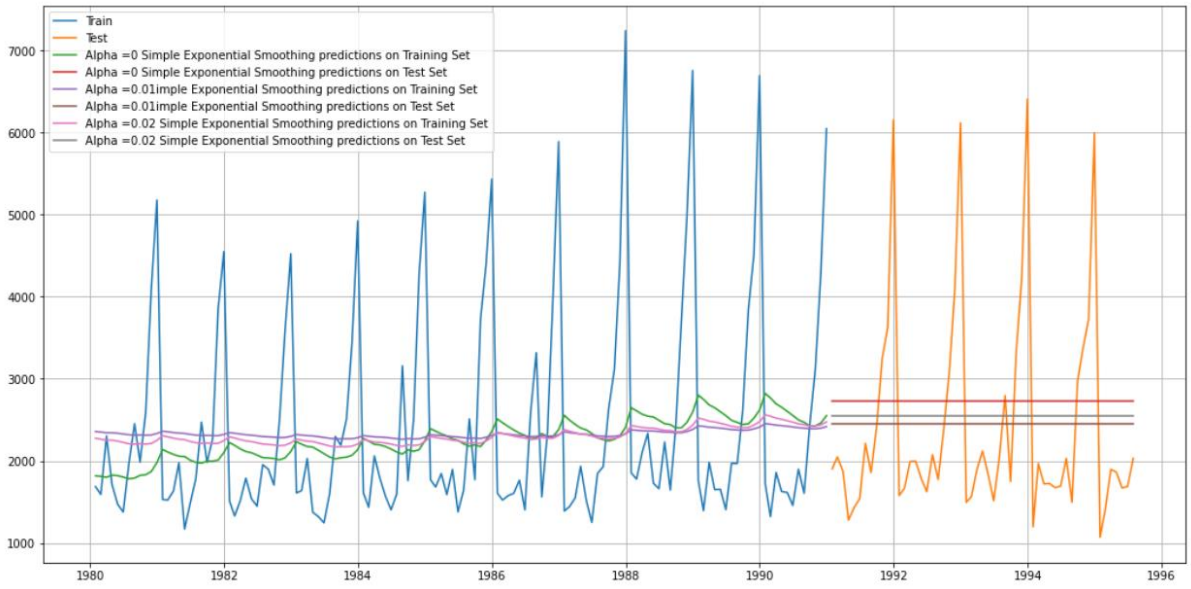


Fig16. SES (Alpha) Train-Test

MODEL7: DOUBLE EXPONENTIAL SMOOTHING

Double Exponential Smoothing is an extension to Exponential Smoothing that explicitly adds support for trends in the univariate time series.

In addition to the alpha parameter for controlling smoothing factor for the level, an additional smoothing factor is added to control the decay of the influence of the change in trend called beta(*b*).

The method supports trends that change in different ways: an additive and a multiplicative, depending on whether the trend is linear or exponential respectively.

Double Exponential Smoothing with an additive trend is classically referred to as Holt's linear trend model, named for the developer of the method Charles Holt.

- **Additive Trend**: Double Exponential Smoothing with a linear trend.
- **Multiplicative Trend**: Double Exponential Smoothing with an exponential trend.

For longer range (multi-step) forecasts, the trend may continue on unrealistically. As such, it can be useful to dampen the trend over time.

Holt Model Results

Dep. Variable:	Sparkling	No. Observations:	132
Model:	Holt	SSE	240445440.041
Optimized:	True	AIC	1910.807
Trend:	Additive	BIC	1922.338
Seasonal:	None	AICC	1911.479
Seasonal Periods:	None	Date:	Sun, 20 Feb 2022
Box-Cox:	False	Time:	17:12:18
Box-Cox Coeff.:	None		

	coeff	code	optimized
smoothing_level	0.6885714	alpha	True
smoothing_trend	0.0001	beta	True
initial_level	1686.0000	l.0	True
initial_trend	-95.000000	b.0	True

Parameters formatted:

	name	param	optimized
smoothing_level	alpha	0.688571	True
smoothing_trend	beta	0.000100	True
initial_level	l.0	1686.000000	True
initial_trend	b.0	-95.000000	True

Predictions on Training data:

Time_Stamp	Sparkling	(predict, 0.68, 0.0)
1980-01-31	1686	1591.000000
1980-02-29	1591	1561.420827
1980-03-31	2304	1486.796779
1980-04-30	1712	1954.564417
1980-05-31	1471	1692.589636

Predictions on Testing data:

Time_Stamp	Sparkling	(predict, 0.68, 0.0)
1991-01-31	1902	5221.278699
1991-02-28	2049	5127.886554
1991-03-31	1874	5034.494409
1991-04-30	1279	4941.102264
1991-05-31	1432	4847.710119

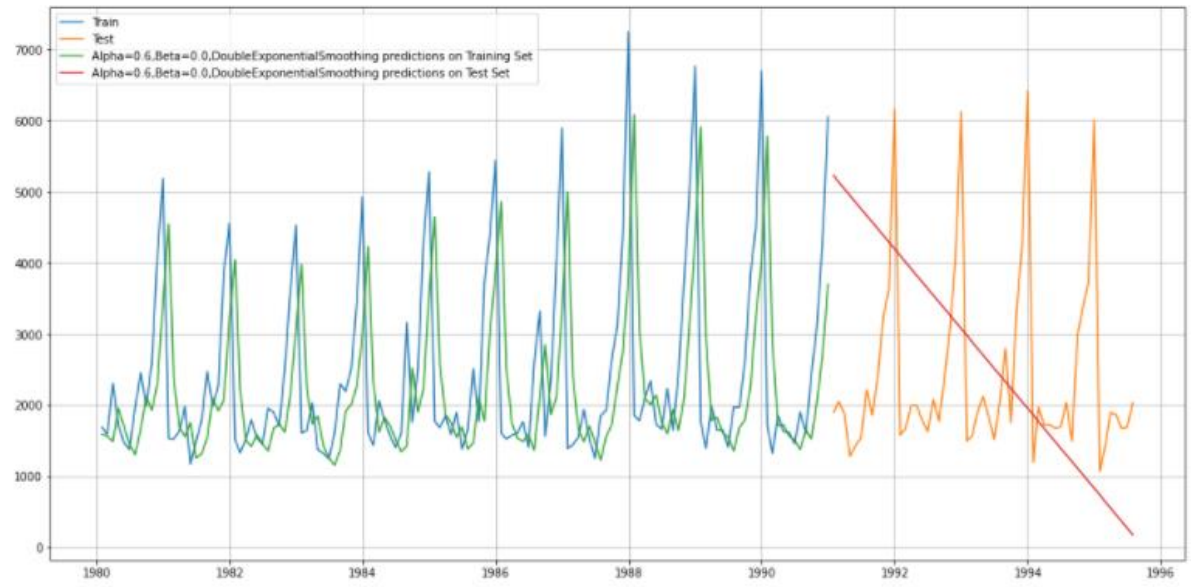


Fig17. DES Train-Test

```
Alpha=0.68 and Beta=0 Double Exponential Smoothing Model forecast on the Training Data,  RMSE is 1349.650
```

```
Alpha=0.68 and Beta=0 Double Exponential Smoothing Model forecast on the Testing Data,  RMSE is 2007.239
```

MODEL8: DOUBLE EXPONENTIAL SMOOTHING IN RANGE 0.01 TO 1

In Double Exponential Smoothing Model, we will run a loop with different alpha, beta values to understand which particular value is best.

	Alpha Values	Beta Values	Train RMSE	Test RMSE
0	0.3	0.3	1590.151686	18259.110704
8	0.4	0.3	1568.527729	23878.496940
1	0.3	0.4	1680.813420	26069.841401
16	0.5	0.3	1530.223975	27095.532414
24	0.6	0.3	1506.223119	29070.722592
...	...	...	...	...
39	0.7	1.0	1816.804334	57297.154185
62	1.0	0.9	1985.350646	57823.177011
47	0.8	1.0	1872.547666	57990.117908
55	0.9	1.0	1947.975429	59008.254331
63	1.0	1.0	2077.639251	59877.076519

64 rows × 4 columns

Holt Model Results

Dep. Variable:	Sparkling	No. Observations:	132
Model:	Holt	SSE	333772874.885
Optimized:	True	AIC	1954.098
Trend:	Additive	BIC	1965.630
Seasonal:	None	AICC	1954.770
Seasonal Periods:	None	Date:	Sun, 20 Feb 2022
Box-Cox:	False	Time:	17:12:21
Box-Cox Coeff.:	None		

	coeff	code	optimized
smoothing_level	0.3000000	alpha	False
smoothing_trend	0.3000000	beta	False
initial_level	1427.2832	l.0	True
initial_trend	119.70551	b.0	True

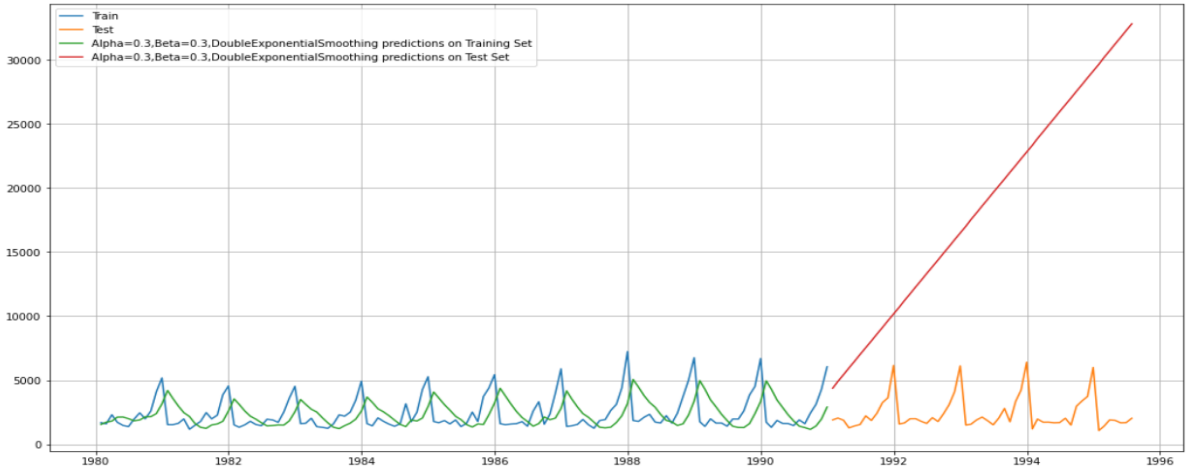


Fig18. DES(Alpha,beta) Train-Test

MODEL9: TRIPLE EXPONENTIAL SMOOTHING

Triple Exponential Smoothing is an extension of Exponential Smoothing that explicitly adds support for seasonality to the univariate time series.

This method is sometimes called Holt-Winters Exponential Smoothing, named for two contributors to the method: Charles Holt and Peter Winters.

In addition to the alpha and beta smoothing factors, a new parameter is added called *gamma* (*g*) that controls the influence on the seasonal component.

As with the trend, the seasonality may be modeled as either an additive or multiplicative process for a linear or exponential change in the seasonality.

- **Additive Seasonality**: Triple Exponential Smoothing with a linear seasonality.
- **Multiplicative Seasonality**: Triple Exponential Smoothing with an exponential seasonality.

Triple exponential smoothing is the most advanced variation of exponential smoothing and through configuration, it can also develop double and single exponential smoothing models.

Being an adaptive method, Holt-Winter's exponential smoothing allows the level, trend and seasonality patterns to change over time.

In Triple Exponential smoothing we have three parameters:

Alpha, Beta, Gamma

Smoothing level value represents Alpha

Smoothing trend value represents Beta

Smoothing Seasonality value represents Gamma

Parameters:

```
{'smoothing_level': 0.11251389383851898,
 'smoothing_trend': 0.037513905124479975,
 'smoothing_seasonal': 0.493687892134789,
 'damping_trend': nan,
 'initial_level': 1640.1903994601003,
 'initial_trend': -2.883527455748493,
 'initial_seasons': array([  45.90357842,  -48.98920376,  662.93554561,   72.68
953516,
        -168.88494907, -262.45259553,  326.06601139,  813.23429222,
         344.33124094,  956.08513572, 2446.81371315, 3538.45996852]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Alpha: 0.112, Beta: 0.037 and Gamma:0.493, Triple Exponential Smoothing Model forecast o n the Training Data, RMSE is **376.297**

Alpha: 0.112, Beta: 0.037 and Gamma:0.493, Triple Exponential Smoothing Model forecast o n the Test Data, RMSE is **473.954**

ExponentialSmoothing Model Results

Dep. Variable:	Sparkling	No. Observations:	132
Model:	ExponentialSmoothing	SSE	18691172.409
Optimized:	True	AIC	1597.620
Trend:	Additive	BIC	1643.745
Seasonal:	Additive	AICC	1603.673
Seasonal Periods:	12	Date:	Sun, 20 Feb 2022
Box-Cox:	False	Time:	17:12:21
Box-Cox Coeff.:	None		

	coeff	code	optimized
smoothing_level	0.1125139	alpha	True
smoothing_trend	0.0375139	beta	True
smoothing_seasonal	0.4936879	gamma	True
initial_level	1640.1904	l.0	True
initial_trend	-2.8835275	b.0	True
initial_seasons.0	45.903578	s.0	True
initial_seasons.1	-48.989204	s.1	True
initial_seasons.2	662.93555	s.2	True
initial_seasons.3	72.689535	s.3	True
initial_seasons.4	-168.88495	s.4	True
initial_seasons.5	-262.45260	s.5	True
initial_seasons.6	326.06601	s.6	True
initial_seasons.7	813.23429	s.7	True
initial_seasons.8	344.33124	s.8	True
initial_seasons.9	956.08514	s.9	True
initial_seasons.10	2446.8137	s.10	True
initial_seasons.11	3538.4600	s.11	True

Prediction on Train data:

Time_Stamp	Sparkling	auto_predict
1980-01-31	1686	1683.210450
1980-02-29	1591	1585.759778
1980-03-31	2304	2295.424490
1980-04-30	1712	1703.329904
1980-05-31	1471	1459.954082

Prediction on Test data:

Time_Stamp	Sparkling	auto_predict
1991-01-31	1902	1474.614638
1991-02-28	2049	1169.444327
1991-03-31	1874	1658.498607
1991-04-30	1279	1504.366522
1991-05-31	1432	1417.164025

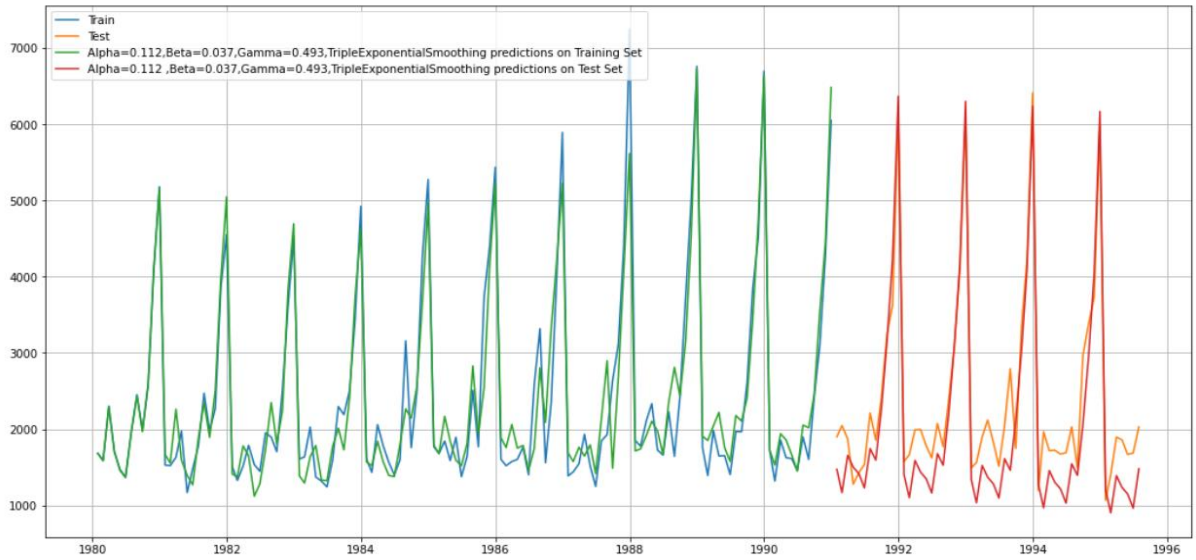


Fig19. TES Train-Test

MODEL10: TRIPLE EXPONENTIAL SMOOTHING IN RANGE 0.3 TO 1.1

In Triple Exponential Smoothing Model, we will run a loop with different alpha, beta and gamma values to understand which particular set of value is best.

The results are stored in a data frame as shown below,

	Alpha Values	Beta Values	Gamma Values	Train RMSE	Test RMSE
280	0.7	0.6	0.3	533.517574	387.732328
328	0.8	0.4	0.3	522.400400	389.253701
178	0.5	0.9	0.5	554.997699	389.813457
290	0.7	0.7	0.5	590.842363	390.441654
168	0.5	0.8	0.3	512.031228	393.196534
...	...	...	...	...	...
489	1.0	0.8	0.4	835.728124	29308.267401
496	1.0	0.9	0.3	852.142556	35779.064449
497	1.0	0.9	0.4	874.232337	36190.851129
505	1.0	1.0	0.4	895.557121	41717.834760
504	1.0	1.0	0.3	914.734293	45252.888467

512 rows × 5 columns

ExponentialSmoothing Model Results

Dep. Variable:	Sparkling	No. Observations:	132
Model:	ExponentialSmoothing	SSE	27763707.765
Optimized:	True	AIC	1649.850
Trend:	Additive	BIC	1695.975
Seasonal:	Additive	AICC	1655.903
Seasonal Periods:	12	Date:	Sun, 20 Feb 2022
Box-Cox:	False	Time:	17:12:37
Box-Cox Coeff.:	None		

	coeff	code	optimized
smoothing_level	0.4000000	alpha	False
smoothing_trend	0.4000000	beta	False
smoothing_seasonal	0.3000000	gamma	False
initial_level	1974.1946	l.0	True
initial_trend	-100.45222	b.0	True
initial_seasons.0	-177.21396	s.0	True
initial_seasons.1	-169.72289	s.1	True
initial_seasons.2	370.12574	s.2	True
initial_seasons.3	304.65498	s.3	True
initial_seasons.4	18.220438	s.4	True
initial_seasons.5	28.691688	s.5	True
initial_seasons.6	532.43062	s.6	True
initial_seasons.7	1058.1493	s.7	True
initial_seasons.8	542.85307	s.8	True
initial_seasons.9	1010.5632	s.9	True
initial_seasons.10	2283.6426	s.10	True
initial_seasons.11	3262.4551	s.11	True

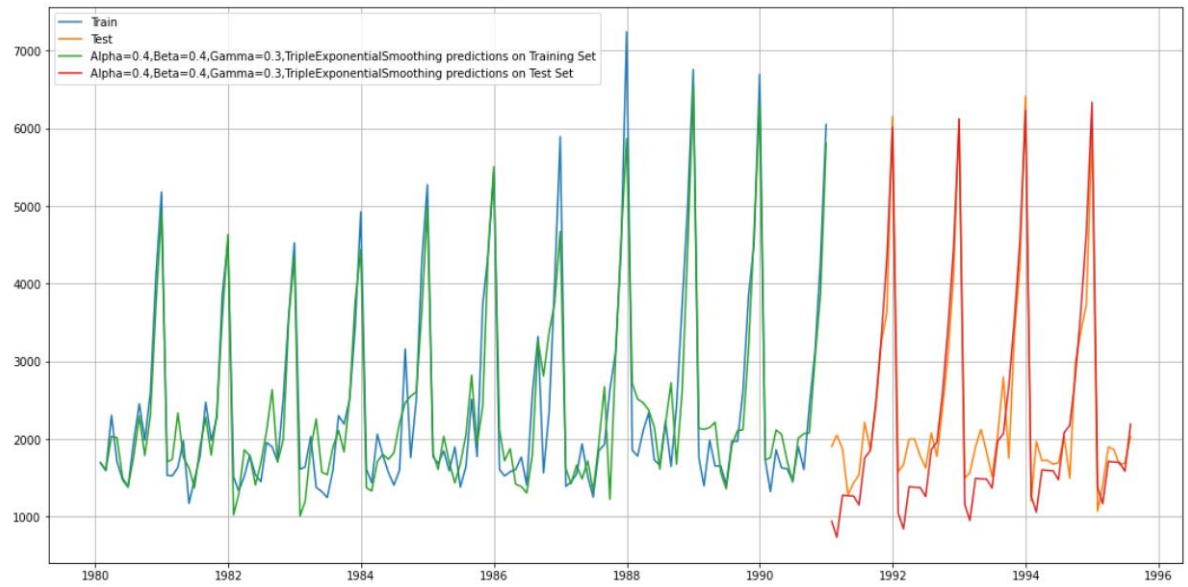


Fig20. TES (Alpha, beta, gamma) Train-Test

The RMSE for the Triple Exponential smoothing model (Alpha=0.4, beta=0.4, gamma=0.3) generated for test data: 462.8880831

Results of RMSE values of the models on Test data:

	Test RMSE
Alpha=0.4, Beta=0.4 ,Gamma=0.3,TripleExponentialSmoothing	462.880831
Alpha: 0.112,Beta: 0.037 and Gamma:0.493,TripleExponentialSmoothing	473.954384
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
SimpleAverageModel	1275.081804
RegressionOnTime	1275.867052
Alpha=0.01,SimpleExponentialSmoothing	1276.251337
6pointTrailingMovingAverage	1283.927428
Alpha=0,SimpleExponentialSmoothing	1316.034674
9pointTrailingMovingAverage	1346.278315
Alpha=0.64 and Beta=0,DoubleExponentialSmoothing	2007.238526
NaiveModel	3864.279352
Alpha=0.3,Beta=0.3,DoubleExponentialSmoothing	18259.110704

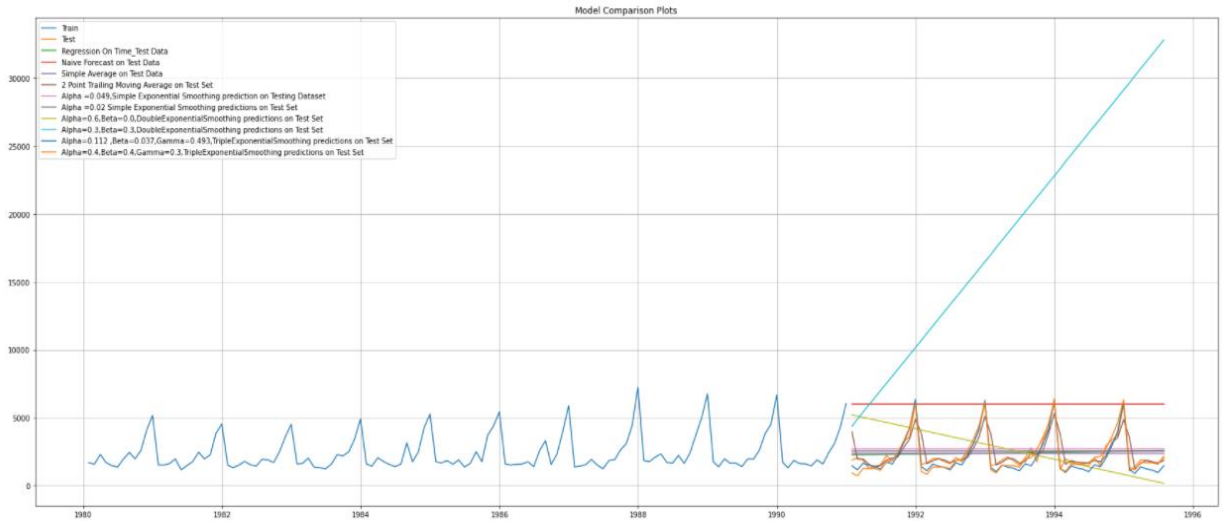
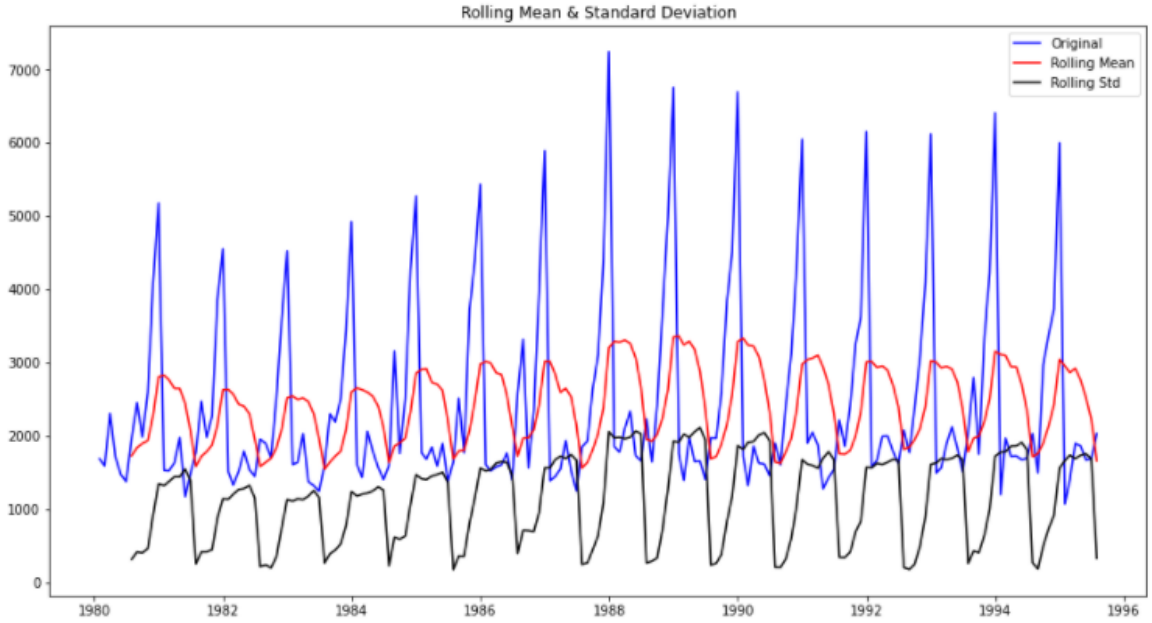


Fig21. Model Comparison Plot

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.




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

We see that at 5% significant level the Time Series is non-stationary.

The null hypothesis for ADF test (H0) is that the time series is non-stationary.

The alternate hypothesis for ADF test (H1) is that time series is stationary.

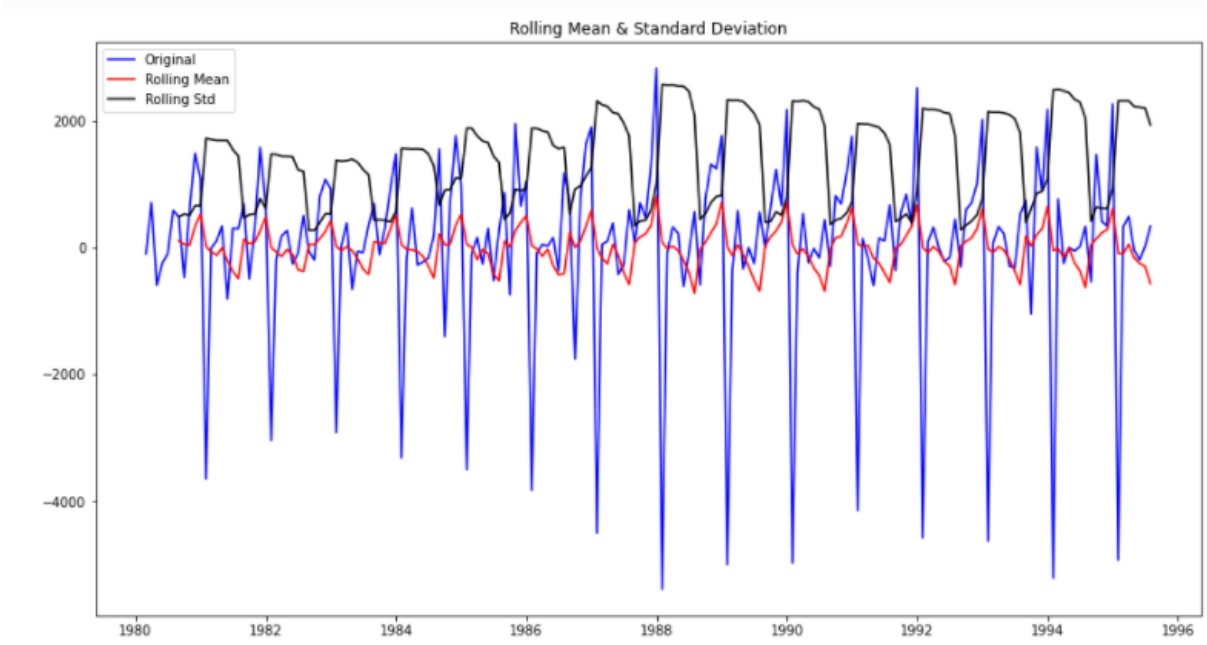
since the p-value of the ADF test is greater than the critical value at 5%, we cannot reject the null hypothesis

Thus, the given time given series is non stationary

To check the stationarity of Sparkling data, we need to check if the alpha value is less than 0.05.

From the above result, we can see that alpha = 0.60 which is higher than 0.05.

Hence, we fail to reject the null hypothesis.




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

After taking a difference of order 1, we see that at α = 0.05 the Time Series is indeed stationary.

Therefore, we apply a difference of 1 and check for stationarity.

Now, the result for alpha value is less than 0.05.

Hence the null hypothesis is rejected and the data is stationary.

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

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.

MODEL11: AUTOMATED ARIMA BASED ON AIC CRITERIA

ARIMA stands for Auto Regressive Integrated Moving Average. It is a class of model that captures a suite of different standard temporal structures in time series data.

ARIMA description, capturing the key aspects of the model itself. Briefly, they are:

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

Each of these components are explicitly specified in the model as a parameter. A standard notation is used of ARIMA (p, d, q) where the parameters are substituted with integer values to quickly indicate the specific ARIMA model being used.

The parameters of the ARIMA model are defined as follows:

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

```
Some parameter combinations for the Model...
Model: (0, 1, 1)
Model: (0, 1, 2)
Model: (0, 1, 3)
Model: (1, 1, 0)
Model: (1, 1, 1)
Model: (1, 1, 2)
Model: (1, 1, 3)
Model: (2, 1, 0)
Model: (2, 1, 1)
Model: (2, 1, 2)
Model: (2, 1, 3)
Model: (3, 1, 0)
Model: (3, 1, 1)
Model: (3, 1, 2)
Model: (3, 1, 3)
```

```
ARIMA(0, 1, 0) - AIC:2267.6630357855465
ARIMA(0, 1, 1) - AIC:2263.060015591336
ARIMA(0, 1, 2) - AIC:2234.408323130674
ARIMA(0, 1, 3) - AIC:2233.9948577624864
ARIMA(1, 1, 0) - AIC:2266.6085393190087
ARIMA(1, 1, 1) - AIC:2235.7550946742404
ARIMA(1, 1, 2) - AIC:2234.527200452325
ARIMA(1, 1, 3) - AIC:2235.607812178285
ARIMA(2, 1, 0) - AIC:2260.365743968097
ARIMA(2, 1, 1) - AIC:2233.7776262581274
ARIMA(2, 1, 2) - AIC:2213.5092126110794
ARIMA(2, 1, 3) - AIC:2232.9861669928914
ARIMA(3, 1, 0) - AIC:2257.72337899794
ARIMA(3, 1, 1) - AIC:2235.498607037124
ARIMA(3, 1, 2) - AIC:2230.753752431717
ARIMA(3, 1, 3) - AIC:2221.4513541225133
```

The table showing AIC values arranged in descending order with continuous combinations of p, d and q:

	param	AIC
10	(2, 1, 2)	2213.509213
15	(3, 1, 3)	2221.451354
14	(3, 1, 2)	2230.753752
11	(2, 1, 3)	2232.986167
9	(2, 1, 1)	2233.777626
3	(0, 1, 3)	2233.994858
2	(0, 1, 2)	2234.408323
6	(1, 1, 2)	2234.527200
13	(3, 1, 1)	2235.498607
7	(1, 1, 3)	2235.607812
5	(1, 1, 1)	2235.755095
12	(3, 1, 0)	2257.723379
8	(2, 1, 0)	2260.365744
1	(0, 1, 1)	2263.060016
4	(1, 1, 0)	2266.608539
0	(0, 1, 0)	2267.663036

```
                               SARIMAX Results
==========================================================================================
Dep. Variable:                      Sparkling   No. Observations:                  132
Model:                 ARIMA(2, 1, 2)   Log Likelihood               -1101.755
Date:                Sun, 20 Feb 2022   AIC                           2213.509
Time:                        17:12:42   BIC                           2227.885
Sample:                    01-31-1980   HQIC                          2219.351
                         - 12-31-1990
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          1.3121      0.046     28.781      0.000       1.223       1.401
ar.L2         -0.5593      0.072     -7.740      0.000      -0.701      -0.418
ma.L1         -1.9917      0.109    -18.214      0.000      -2.206      -1.777
ma.L2          0.9999      0.110      9.108      0.000       0.785       1.215
sigma2      1.099e+06      2e-07   5.51e+12      0.000     1.1e+06     1.1e+06
===================================================================================
Ljung-Box (L1) (Q):                   0.19   Jarque-Bera (JB):                14.46
Prob(Q):                              0.67   Prob(JB):                         0.00
Heteroskedasticity (H):               2.43   Skew:                             0.61
Prob(H) (two-sided):                  0.00   Kurtosis:                         4.08
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 5.11e+27. Standard errors may be unstable.
```

Plot diagnostics:

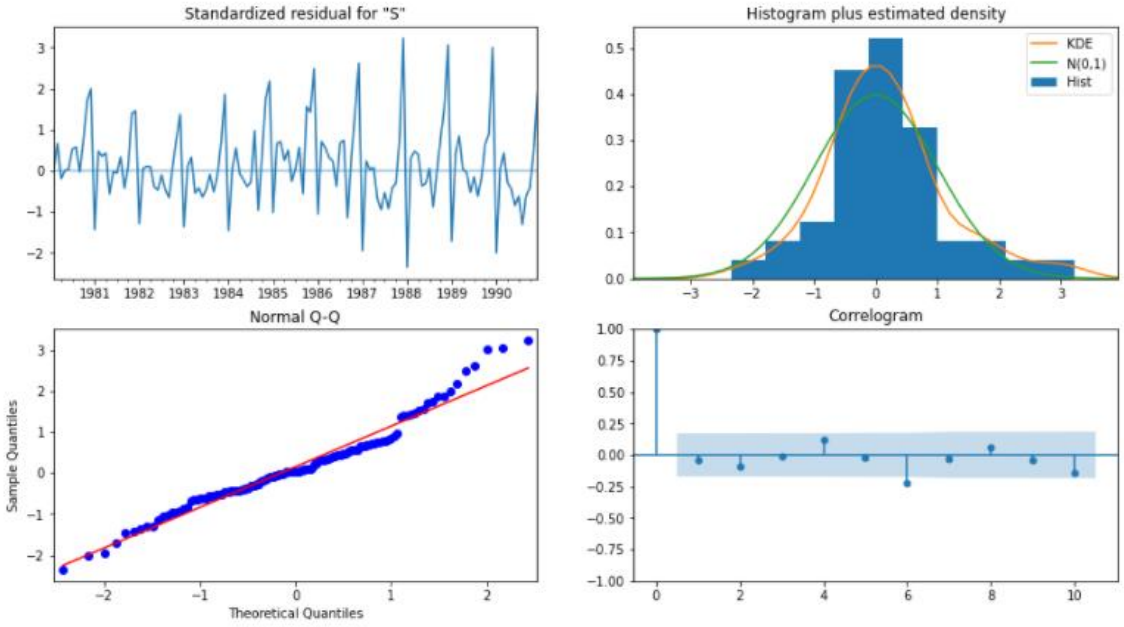


Fig22. Plot Diagnostics

We ran the automated ARIMA model for Sparkling Sales and sorted AIC values output from lowest to highest.

We then proceeded to build the ARIMA model with the lowest Akaike Information Criteria.

The ARIMA model is built with the best parameters based on the least AIC value in the above table.

RMSE for the autofit ARIMA model: 1299.979523620647

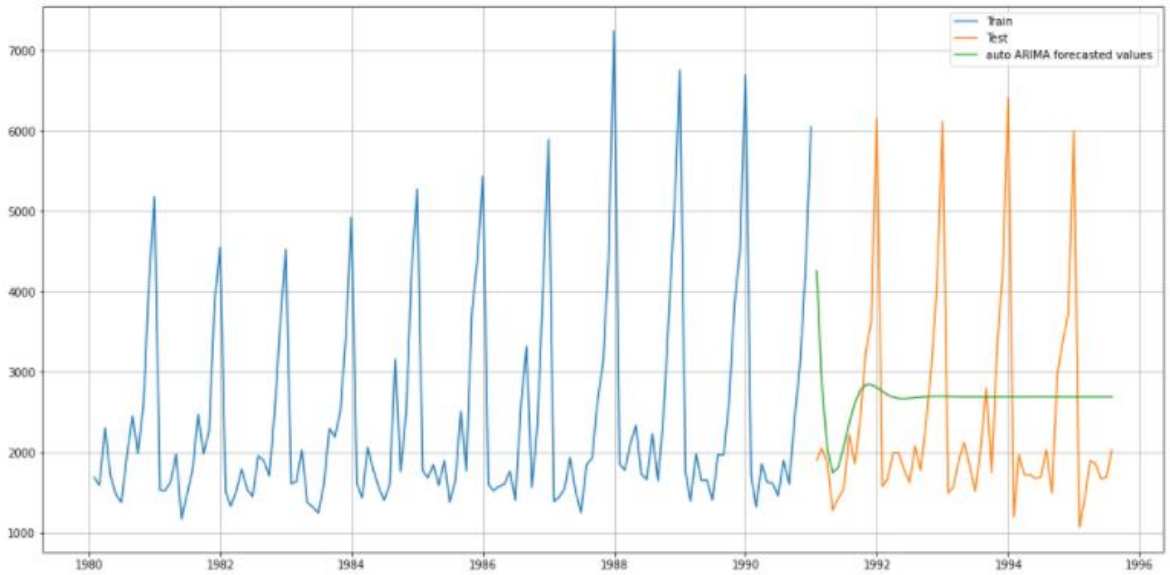


Fig23. Autofit Arima

MODEL12: AUTOMATED SARIMA MODEL WITH SEASONALITY 6 & 12:

A seasonal autoregressive integrated moving average (SARIMA) model is one step different from an ARIMA model based on the concept of seasonal trends. In many time series data, frequent seasonal effects come into play. Take for example the average temperature measured in a location with four seasons. There will be a seasonal effect on a yearly basis, and the temperature in this particular season will definitely have a strong correlation with the temperature measured last year in the same season.

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.

Configuring a SARIMA requires selecting hyperparameters for both the trend and seasonal elements of the series.

Trend Elements

There are three trend elements that require configuration.

They are the same as the ARIMA model; specifically:

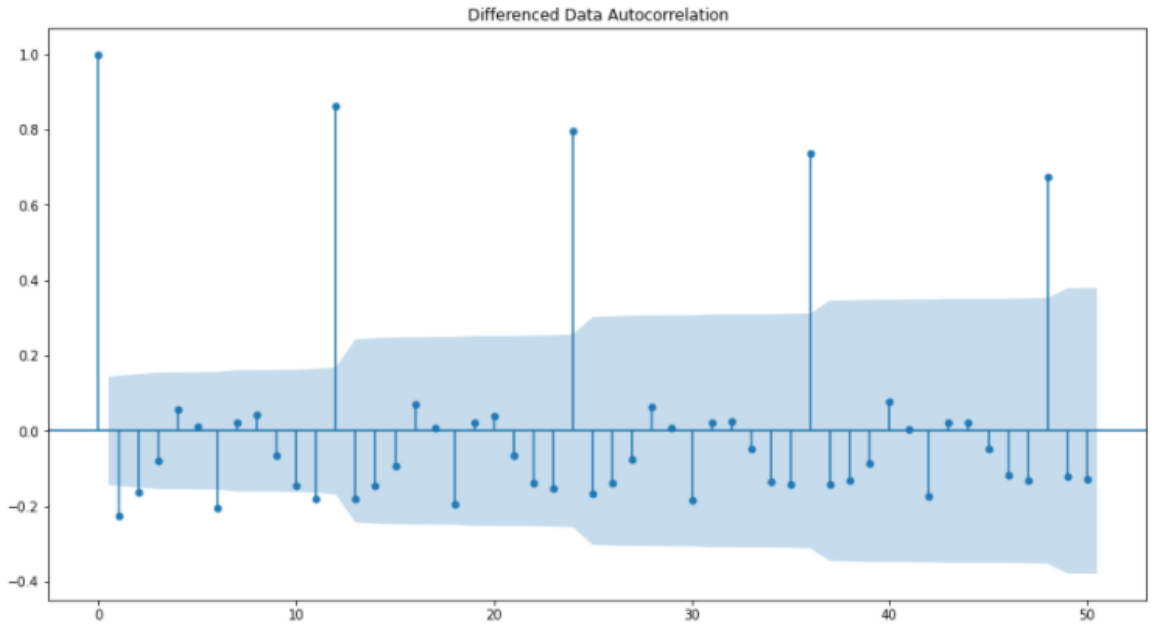
- **p**: Trend autoregression order.
- **d**: Trend difference order.
- **q**: Trend moving average order.

Seasonal Elements

There are four seasonal elements that are not part of ARIMA that must be configured; they are:

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

Let us look at the ACF plot once more to understand the seasonal parameter for the SARIMA model.



We see that there can be a seasonality of 6 as well as 12. We will run our auto SARIMA models by setting seasonality both as 6 and 12.

```
Examples of some parameter combinations for Model...
Model: (0, 1, 1)(0, 0, 1, 6)
Model: (0, 1, 2)(0, 0, 2, 6)
Model: (1, 1, 0)(1, 0, 0, 6)
Model: (1, 1, 1)(1, 0, 1, 6)
Model: (1, 1, 2)(1, 0, 2, 6)
Model: (2, 1, 0)(2, 0, 0, 6)
Model: (2, 1, 1)(2, 0, 1, 6)
Model: (2, 1, 2)(2, 0, 2, 6)
```

	param	seasonal	AIC
53	(1, 1, 2)	(2, 0, 2, 6)	1727.678698
26	(0, 1, 2)	(2, 0, 2, 6)	1727.888804
80	(2, 1, 2)	(2, 0, 2, 6)	1729.321425
17	(0, 1, 1)	(2, 0, 2, 6)	1741.647354
44	(1, 1, 1)	(2, 0, 2, 6)	1743.379778

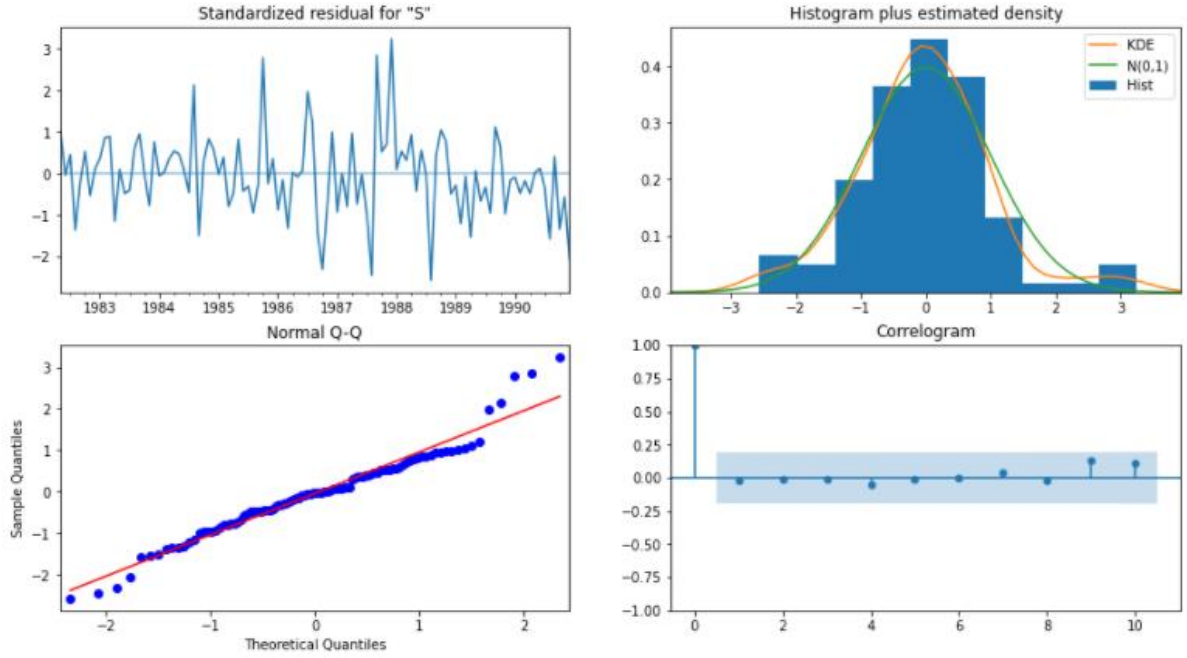
SARIMAX Results

Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(1, 1, 2)x(2, 0, 2, 6)	Log Likelihood	-855.839
Date:	Wed, 23 Feb 2022	AIC	1727.679
Time:	14:26:21	BIC	1749.707
Sample:	0 - 132	HQIC	1736.621
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.6449	0.286	-2.257	0.024	-1.205	-0.085
ma.L1	-0.1068	0.250	-0.428	0.669	-0.596	0.383
ma.L2	-0.7006	0.202	-3.471	0.001	-1.096	-0.305
ar.S.L6	-0.0045	0.027	-0.165	0.869	-0.057	0.049
ar.S.L12	1.0361	0.018	56.083	0.000	1.000	1.072
ma.S.L6	0.0676	0.152	0.444	0.657	-0.231	0.366
ma.S.L12	-0.6123	0.093	-6.590	0.000	-0.794	-0.430
sigma2	1.448e+05	1.71e+04	8.466	0.000	1.11e+05	1.78e+05

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

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



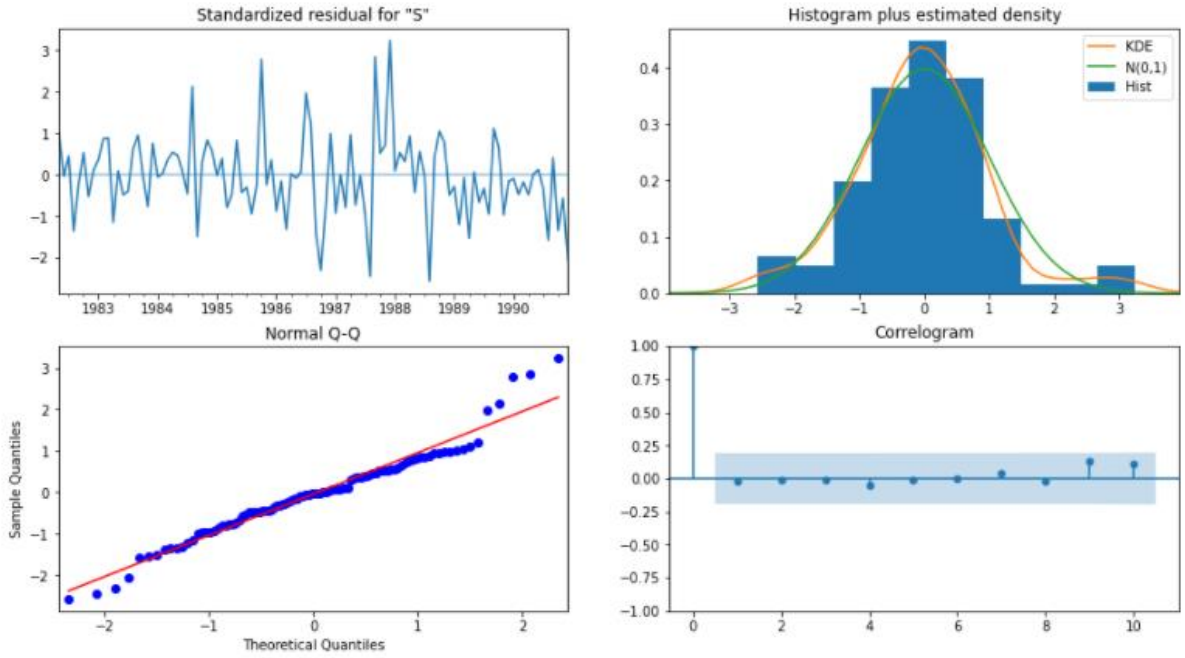
RMSE for the autofit ARIMA model: 626.9256927299741

Setting the seasonality as 12 for the second iteration of the auto SARIMA model:

```
Examples of some parameter combinations for Model...
Model: (0, 1, 1)(0, 0, 1, 12)
Model: (0, 1, 2)(0, 0, 2, 12)
Model: (1, 1, 0)(1, 0, 0, 12)
Model: (1, 1, 1)(1, 0, 1, 12)
Model: (1, 1, 2)(1, 0, 2, 12)
Model: (2, 1, 0)(2, 0, 0, 12)
Model: (2, 1, 1)(2, 0, 1, 12)
Model: (2, 1, 2)(2, 0, 2, 12)
```

Sorted the AIC values output from lowest to highest, we then proceed to build the SARIMA MODEL with the lowest Akaike Information Criteria.

	param	seasonal	AIC
50	(1, 1, 2)	(1, 0, 2, 12)	1555.584247
53	(1, 1, 2)	(2, 0, 2, 12)	1555.929654
26	(0, 1, 2)	(2, 0, 2, 12)	1557.121565
23	(0, 1, 2)	(1, 0, 2, 12)	1557.160507
77	(2, 1, 2)	(1, 0, 2, 12)	1557.340405



SARIMAX Results

Dep. Variable:	Sparkling	No. Observations:	132
Model:	SARIMAX(1, 1, 2)x(1, 0, 2, 12)	Log Likelihood	-770.792
Date:	Sun, 20 Feb 2022	AIC	1555.584
Time:	17:13:23	BIC	1574.095
Sample:	01-31-1980 - 12-31-1990	HQIC	1563.084
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.6283	0.254	-2.474	0.013	-1.126	-0.131
ma.L1	-0.1033	0.224	-0.462	0.644	-0.541	0.335
ma.L2	-0.7284	0.152	-4.779	0.000	-1.027	-0.430
ar.S.L12	1.0439	0.014	72.746	0.000	1.016	1.072
ma.S.L12	-0.5547	0.098	-5.653	0.000	-0.747	-0.362
ma.S.L24	-0.1356	0.120	-1.133	0.257	-0.370	0.099
sigma2	1.506e+05	2.04e+04	7.401	0.000	1.11e+05	1.91e+05

Ljung-Box (L1) (Q):	0.04	Jarque-Bera (JB):	11.66
Prob(Q):	0.84	Prob(JB):	0.00
Heteroskedasticity (H):	1.47	Skew:	0.36
Prob(H) (two-sided):	0.26	Kurtosis:	4.47

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

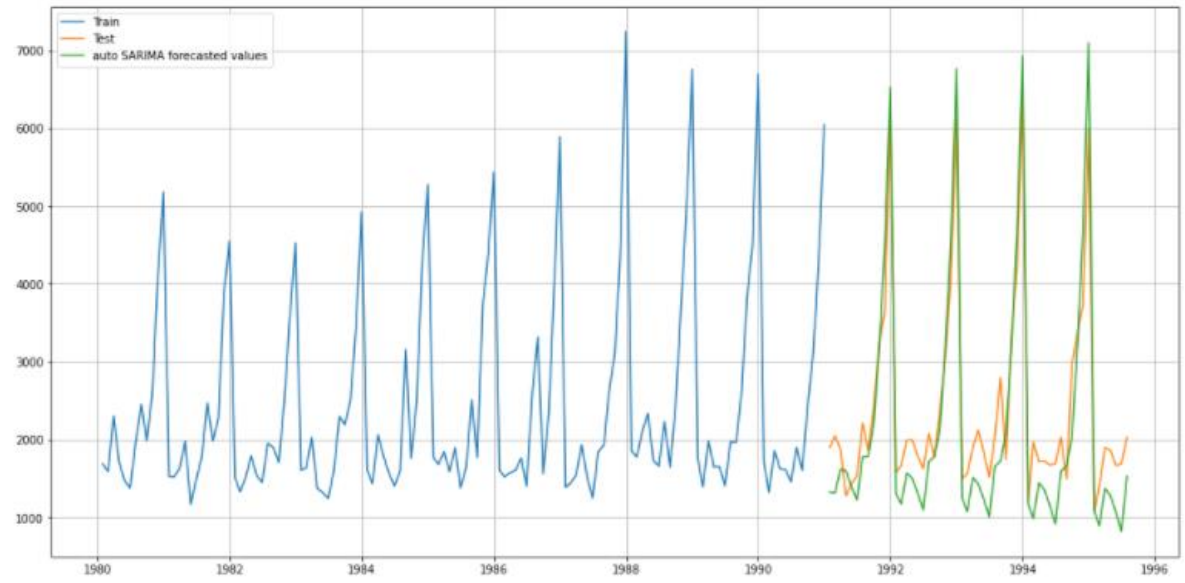


Fig24. Automated Sarima

RMSE for the autofit SARIMA model: 528.389740110892

Inference on Model diagnostics confirms that the model residuals are normally distributed.

Standardized residual: Do not display any obvious seasonality

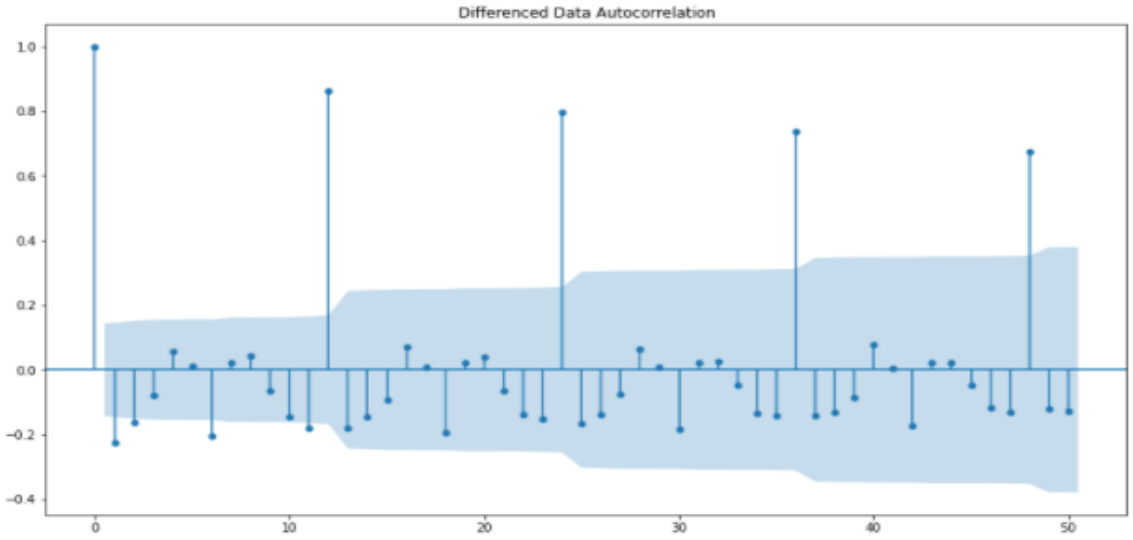
Histogram plus estimated density: The KDE plot of the residuals is similar with the normal distribution. Hence the model residuals are normally distributed based.

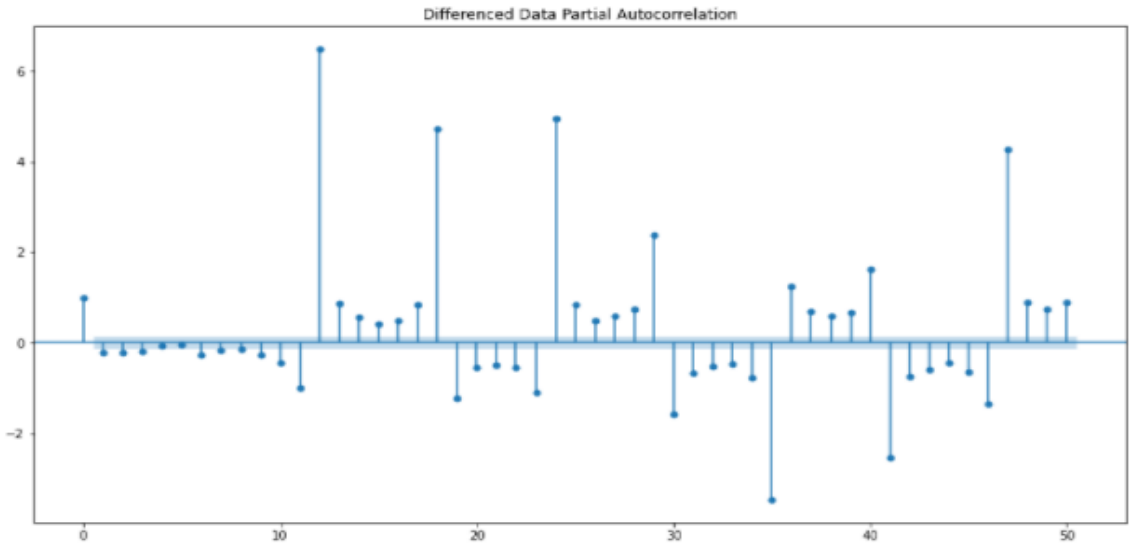
Normal Q-Q plot: There is an ordered distribution of residuals (blue dots) following the linear trend of samples taken from a standard normal distribution.

Correlogram: The time series residuals have low correlation with lagged versions itself.

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.

MODEL13: MANUAL ARIMA WITH CUT-OFF VALUES FROM ACF AND PACF





The p value from PACF is 3 as there are 3 significant values above the cut-off

The q value from ACF is 2 as there are 2 significant values above the cut-off

The d values is 1 as we need single order differencing to make the series stationary

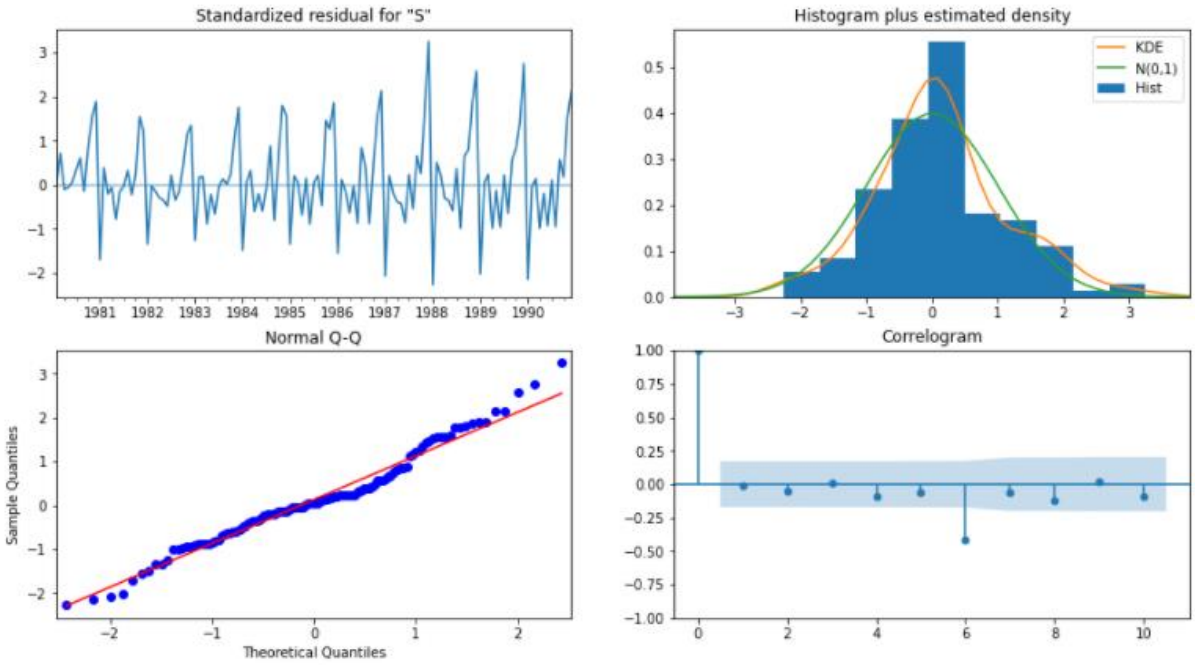
SARIMAX Results

Dep. Variable:	Sparkling	No. Observations:	132
Model:	ARIMA(3, 1, 2)	Log Likelihood	-1109.377
Date:	Sun, 20 Feb 2022	AIC	2230.754
Time:	17:13:24	BIC	2248.005
Sample:	01-31-1980 - 12-31-1990	HQIC	2237.764
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.4330	0.042	-10.396	0.000	-0.515	-0.351
ar.L2	0.3259	0.112	2.903	0.004	0.106	0.546
ar.L3	-0.2411	0.071	-3.415	0.001	-0.380	-0.103
ma.L1	0.0194	0.127	0.152	0.879	-0.230	0.269
ma.L2	-0.9804	0.135	-7.243	0.000	-1.246	-0.715
sigma2	1.267e+06	1.94e-07	6.52e+12	0.000	1.27e+06	1.27e+06

Ljung-Box (L1) (Q):	0.02	Jarque-Bera (JB):	4.60
Prob(Q):	0.89	Prob(JB):	0.10
Heteroskedasticity (H):	2.72	Skew:	0.37
Prob(H) (two-sided):	0.00	Kurtosis:	3.54

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 2.3e+29. Standard errors may be unstable.



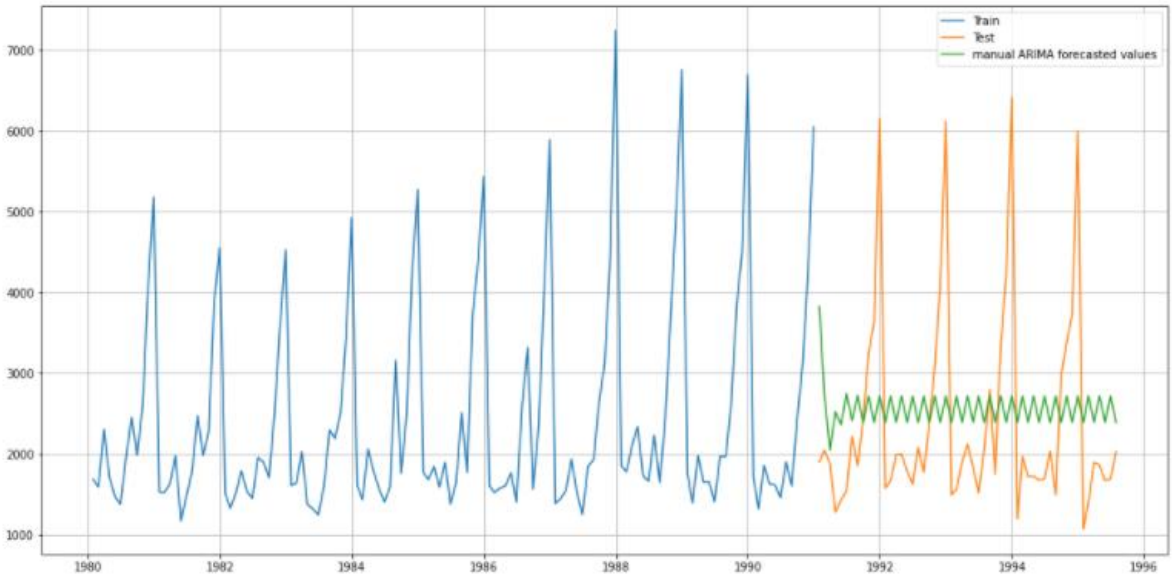


Fig25. Manual Arima

RMSE for the manual ARIMA model: 1283.3528418167834

Manual ARIMA is built based on ACF plot and PACF plot.

Hence, we choose AR parameter value as p and moving average parameter value to be q.

MODEL14: MANUAL SARIMA

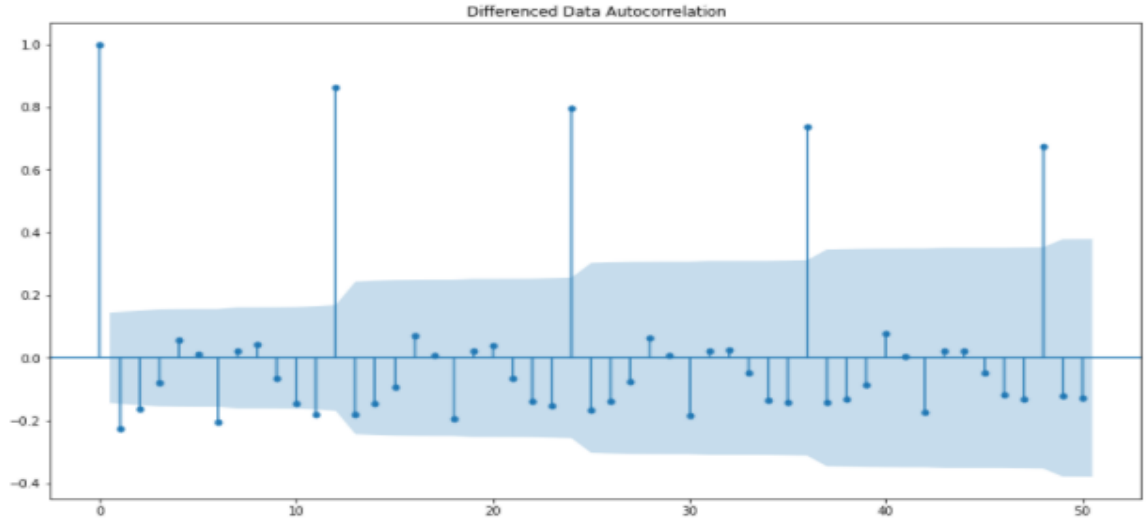
SARIMAX Results

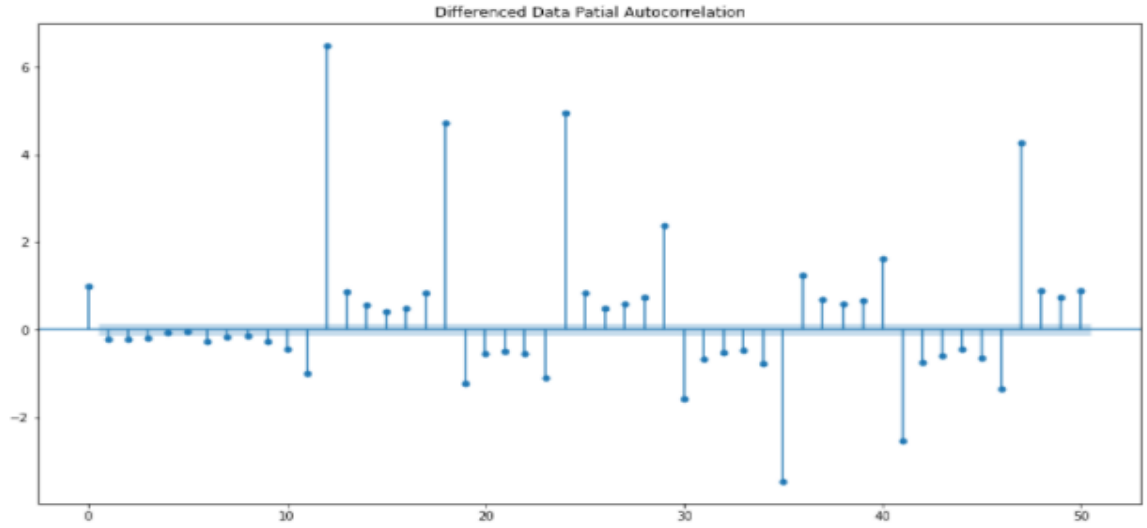
Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(3, 1, 2)x(3, 1, 2, 12)	Log Likelihood	-598.630
Date:	Sun, 20 Feb 2022	AIC	1219.260
Time:	17:13:29	BIC	1245.462
Sample:	0	HQIC	1229.765
	- 132		
Covariance Type:	opg		

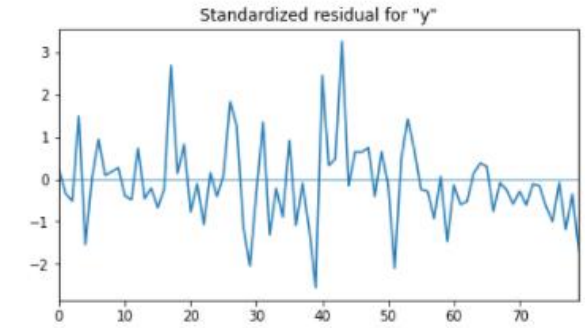
	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.7556	0.151	-5.013	0.000	-1.051	-0.460
ar.L2	0.1168	0.185	0.633	0.527	-0.245	0.479
ar.L3	-0.0520	0.143	-0.365	0.715	-0.332	0.228
ma.L1	0.0331	0.191	0.173	0.862	-0.341	0.407
ma.L2	-0.9670	0.156	-6.197	0.000	-1.273	-0.661
ar.S.L12	-0.7538	0.496	-1.520	0.128	-1.725	0.218
ar.S.L24	-0.6371	0.351	-1.818	0.069	-1.324	0.050
ar.S.L36	-0.2469	0.151	-1.641	0.101	-0.542	0.048
ma.S.L12	0.3719	0.491	0.758	0.448	-0.590	1.333
ma.S.L24	0.3467	0.365	0.949	0.343	-0.370	1.063
sigma2	1.79e+05	1.67e-06	1.07e+11	0.000	1.79e+05	1.79e+05

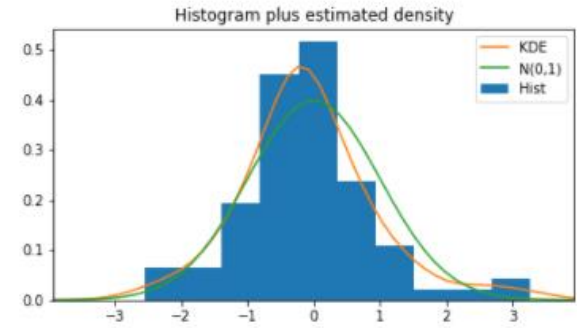
Ljung-Box (L1) (Q):	0.01	Jarque-Bera (JB):	13.16
Prob(Q):	0.93	Prob(JB):	0.00
Heteroskedasticity (H):	0.66	Skew:	0.62
Prob(H) (two-sided):	0.29	Kurtosis:	4.55

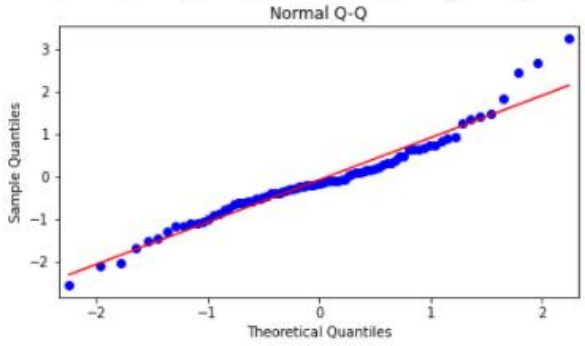
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 1.31e+28. Standard errors may be unstable.

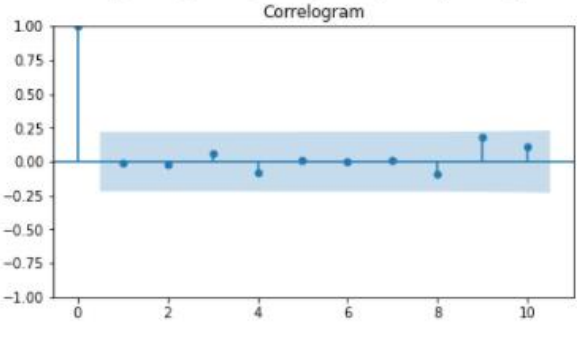

Differenced Data Autocorrelation


Differenced Data Patial Autocorrelation


Standardized residual for "y"


Histogram plus estimated density
KDE
N(0,1)
Hist


Normal Q-Q
Sample Quantiles
Theoretical Quantiles


Correlogram

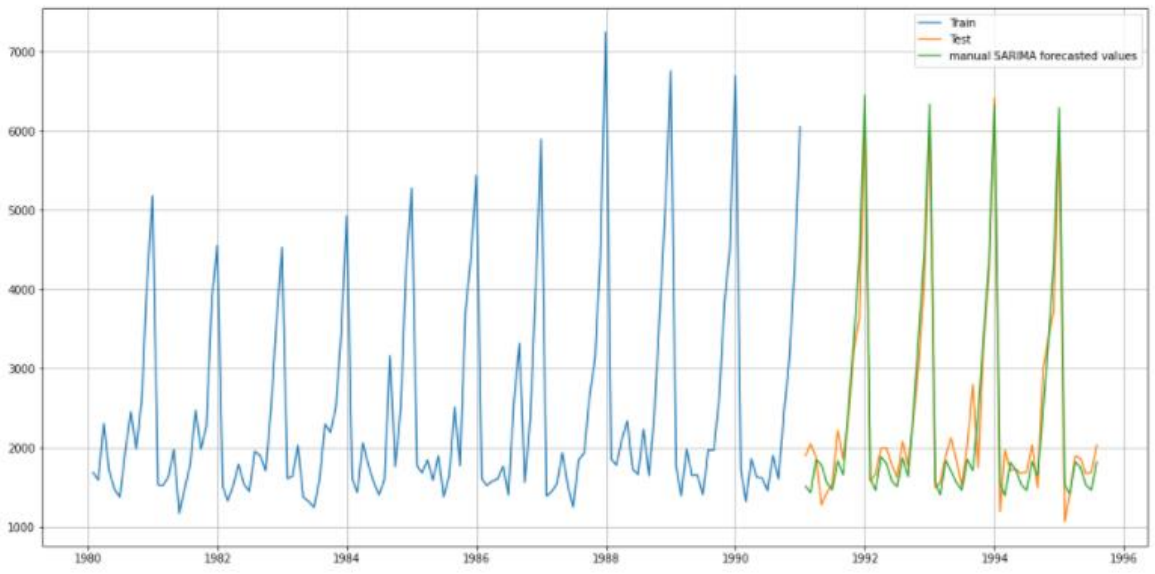


Fig26. Manual Sarima

RMSE for the manual SARIMA model: 329.55690868118654

Manual SARIMA is built based on ACF plot and PACF plot.

Hence, we choose AR parameter value as p. moving average parameter value to be q and d(difference) value to be 1.

We then derive the seasonal parameters based on the seasonal cut-off.

Inference on Model diagnostics confirms that the model residuals are normally distributed.

Standardized residual: Do not display any obvious seasonality

Histogram plus estimated density: The KDE plot of the residuals is similar with the normal distribution. Hence the model residuals are normally distributed based.

Normal Q-Q plot: There is an ordered distribution of residuals (blue dots) following the linear trend of samples taken from a standard normal distribution.

Correlogram: The time series residuals have low correlation with lagged versions itself.

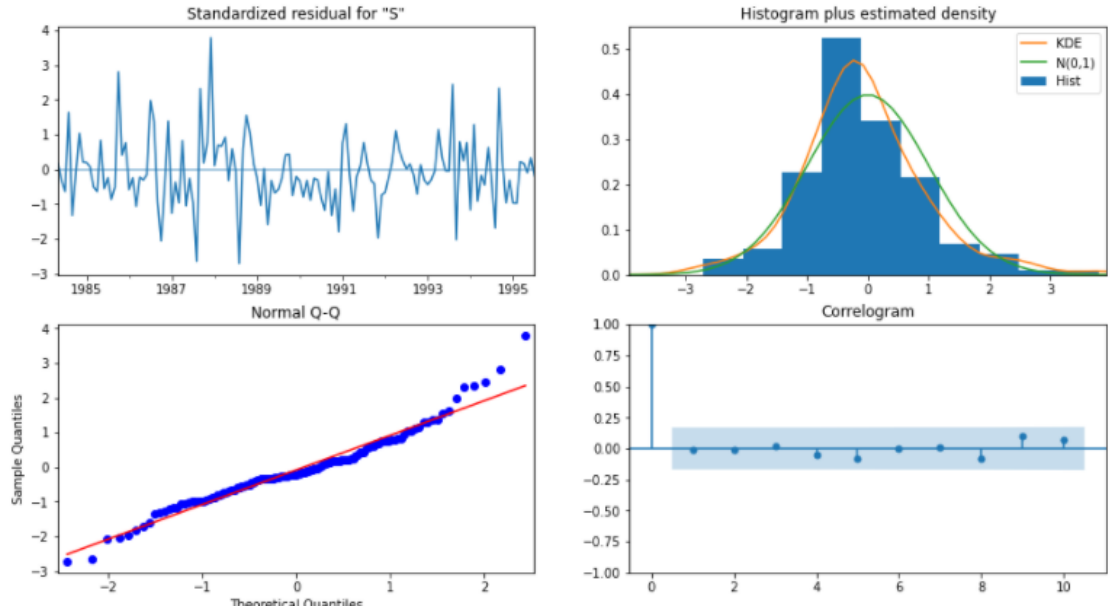
8. Build a table (create a data frame) with all the models built along with their corresponding parameters and the respective RMSE values on the test data.

	Test RMSE
RegressionOnTime	1275.867052
NaiveModel	3864.279352
SimpleAverageModel	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315
Alpha=0,SimpleExponentialSmoothing	1316.034674
Alpha=0.01,SimpleExponentialSmoothing	1276.251337
Alpha=0.64 and Beta=0,DoubleExponentialSmoothing	2007.238526
Alpha=0.3,Beta=0.3,DoubleExponentialSmoothing	18259.110704
Alpha: 0.112,Beta: 0.037 and Gamma:0.493,TripleExponentialSmoothing	473.954384
Alpha=0.4, Beta=0.4 ,Gamma=0.3,TripleExponentialSmoothing	462.880831
automated ARIMA(2,1,2)	1299.979524
automatedSARIMA(1,1,2)(2,0,2,6)	626.925693
automated SARIMA(1,1,2)*(1,0,2,12)	528.389740
manual ARIMA(3,1,2)	1283.352842
manual SARIMA(3,1,2)(3,1,2,12)	329.556909

	Test RMSE
manual SARIMA(3,1,2)(3,1,2,12)	329.556909
Alpha=0.4, Beta=0.4 ,Gamma=0.3,TripleExponentialSmoothing	462.880831
Alpha: 0.112,Beta: 0.037 and Gamma:0.493,TripleExponentialSmoothing	473.954384
automated SARIMA(1,1,2)*(1,0,2,12)	528.389740
automatedSARIMA(1,1,2)(2,0,2,6)	626.925693
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
SimpleAverageModel	1275.081804
RegressionOnTime	1275.867052
Alpha=0.01,SimpleExponentialSmoothing	1276.251337
manual ARIMA(3,1,2)	1283.352842
6pointTrailingMovingAverage	1283.927428
automated ARIMA(2,1,2)	1299.979524
Alpha=0,SimpleExponentialSmoothing	1316.034674
9pointTrailingMovingAverage	1346.278315
Alpha=0.64 and Beta=0,DoubleExponentialSmoothing	2007.238526
NaiveModel	3864.279352
Alpha=0.3,Beta=0.3,DoubleExponentialSmoothing	18259.110704

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.

Optimum Model on Complete Dataset:



SARIMAX Results

Dep. Variable:	Sparkling	No. Observations:	187
Model:	SARIMAX(3, 1, 2)x(3, 1, 2, 12)	Log Likelihood	-1000.243
Date:	Sun, 20 Feb 2022	AIC	2022.487
Time:	17:13:38	BIC	2054.445
Sample:	01-31-1980 - 07-31-1995	HQIC	2035.473
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.8611	0.090	-9.546	0.000	-1.038	-0.684
ar.L2	0.0117	0.129	0.090	0.928	-0.242	0.265
ar.L3	-0.0767	0.102	-0.754	0.451	-0.276	0.123
ma.L1	0.0322	0.120	0.269	0.788	-0.203	0.267
ma.L2	-0.9678	0.098	-9.839	0.000	-1.161	-0.775
ar.S.L12	-0.6104	0.392	-1.556	0.120	-1.379	0.158
ar.S.L24	-0.4985	0.231	-2.162	0.031	-0.950	-0.047
ar.S.L36	-0.2472	0.109	-2.263	0.024	-0.461	-0.033
ma.S.L12	0.1232	0.395	0.311	0.755	-0.652	0.898
ma.S.L24	0.2490	0.266	0.937	0.349	-0.272	0.770
sigma2	1.562e+05	1.31e-06	1.19e+11	0.000	1.56e+05	1.56e+05

Ljung-Box (L1) (Q):	0.01	Jarque-Bera (JB):	26.96
Prob(Q):	0.92	Prob(JB):	0.00
Heteroskedasticity (H):	0.56	Skew:	0.59
Prob(H) (two-sided):	0.05	Kurtosis:	4.84

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 5.83e+26. Standard errors may be unstable.

Forecasting 12 months into the future with the complete model

Sparkling	mean	mean_se	mean_ci_lower	mean_ci_upper
1995-08-31	1868.722447	396.502573	1091.591683	2645.853210
1995-09-30	2511.337191	401.850000	1723.725664	3298.948718
1995-10-31	3272.700171	402.696945	2483.428663	4061.971679
1995-11-30	3874.461237	403.111966	3084.376302	4664.546173
1995-12-31	6099.009090	403.131652	5308.885571	6889.132609

Sparkling	mean	mean_se	mean_ci_lower	mean_ci_upper
1996-03-31	1872.458742	404.471778	1079.708625	2665.208859
1996-04-30	1851.465489	404.492868	1058.674035	2644.256942
1996-05-31	1719.880268	405.073050	925.951679	2513.808856
1996-06-30	1631.715759	405.103224	837.728030	2425.703488
1996-07-31	2038.441588	405.645169	1243.391665	2833.491510

RMSE of the Full Model 578.96457095304

From the Previous answer we observe that Manual Sarima has the least RMSE score.

It falls under most optimum model compared to other models.

Sparkling	mean	mean_se	mean_ci_lower	mean_ci_upper
1995-08-31	1868.722447	396.502573	1091.591683	2645.853210
1995-09-30	2511.337191	401.850000	1723.725664	3298.948718
1995-10-31	3272.700171	402.696945	2483.428663	4061.971679
1995-11-30	3874.461237	403.111966	3084.376302	4664.546173
1995-12-31	6099.009090	403.131652	5308.885571	6889.132609
1996-01-31	1191.793876	403.840266	400.281498	1983.306253
1996-02-29	1557.080087	403.852106	765.544505	2348.615669
1996-03-31	1872.458742	404.471778	1079.708625	2665.208859
1996-04-30	1851.465489	404.492868	1058.674035	2644.256942
1996-05-31	1719.880268	405.073050	925.951679	2513.808856
1996-06-30	1631.715759	405.103224	837.728030	2425.703488
1996-07-31	2038.441588	405.645169	1243.391665	2833.491510

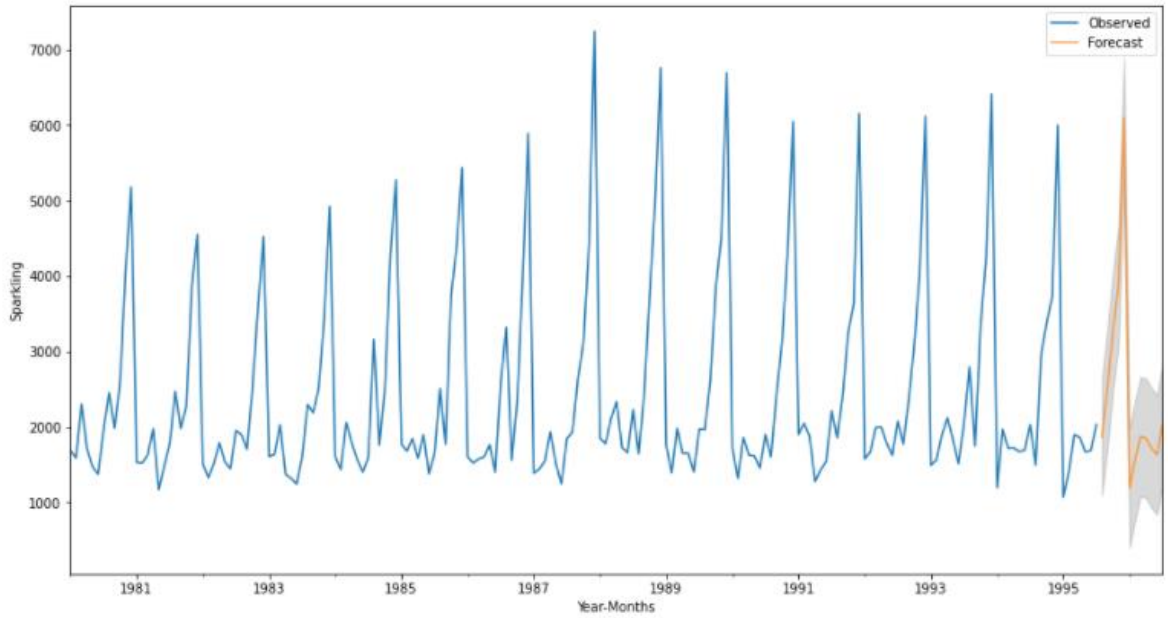


Fig27. Final Model

Inference on Model diagnostics confirms that the model residuals are normally distributed.

Standardized residual: Do not display any obvious seasonality

Histogram plus estimated density: The KDE plot of the residuals is similar with the normal distribution. Hence the model residuals are normally distributed based.

Normal Q-Q plot: There is an ordered distribution of residuals (blue dots) following the linear trend of samples taken from a standard normal distribution.

Correlogram: The time series residuals have low correlation with lagged versions itself.

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

Time series analysis involves understanding various aspects about the inherent nature of the series so that you are better informed to create meaningful and accurate forecasts**.**

Any time series may be split into the following components:

Base Level + Trend + Seasonality + Error

Observations:

Sparkling sales shows stabilized values.

December month shows the highest sales.

The models are built and are chosen based on the least RMSE score.

The sales of Sparkling wine is seasonal and also had trend. Therefore, the company cannot have the same stock throughout the year.

The company should use prediction results to plan about future stock.

Inference:

The models are built considering the Trend and Seasonality in to account and we see from the output plot that the future prediction is in line with the trend and seasonality in the previous years.

The company should use the prediction results and capitalize on the high demand seasons and ensure to source and supply the high demand.

The company should use the prediction results to plan the low demand seasons to stock as per the demand.

Products that are discounted should be highlighted so consumers can see the savings prominently Discounts can compel consumers to buy.

As we know how the seasonality is in the prediction company cannot have the same stock through the year.

You should create a dynamic consumer experience with fresh point -of-sale materials and well stocked displays.

Displays need to look fresh and interesting and tell a compelling story about why the consumer should purchase the product.

Seasonal memberships and discounts can be introduced. Consumers get very excited about savings and appreciate discounts being passed on.

Many prominent retailers also have loyalty programs or club member cards that create excitement. A club -member price brings consumers back and improve sales

Events and tastings help draw consumers to your store and generate sales.
Retailers with economies of scale successfully sample consumers on more profitable wines.

Some even comparison -taste customers on national brands that are more expensive to demonstrate they are offering a less expensive but superior product. And bringing in celebrities, sommeliers or trade reps for tastings can help create excitement and drive traffic.