greatlearning
Power Ahead

TIME SERIES FORECASTING

Mohamed Rifaz Ali K S

PGP-DSBA Online

February' 22

greatlearning
Power Ahead

Contents

List of Tables

List of Figures

Problem: Wine Sales Forecasting Project

Executive Summary

The data of different types of wine sales in the 20th century is to be analyzed. We are provided with the 2 different types of wines named 'Sparkling' and 'Rose'. As an analyst in the ABC Estate Wines, we are asked to analyze and forecast Wine Sales in the 20th century.

Introduction

The purpose of this problem is to analyze and forecast Wine Sales in the 20th century. We are provided with the monthly sales data of 'Sparkling' and 'Rose' wines from 1980 January to 1995 July. As an analyst in the ABC Estate Wines, we are asked to analyze and forecast Wine Sales in the 20th century for the future 12 months. To accomplish this, we use different types of Time Series forecasting models such as Linear Regression, Naïve, Simple Average, Moving Average, Exponential Smoothing and ARIMA/SARIMA models and evaluate the best model using RMSE model and fit the model with original data and predict 12 months into the future. We will analyze the Sparkling wine sales and Rose wine sales individually.

Sparkling Wine Sales

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

Read the Sparkling Time Series Data

There are 187 records and two columns YearMonth and Sales. While reading the Time Series data, let's read the data's using **'parse_dates=True'** parameter and set the index column as 'YearMonth'. By this, we will use the lot of Time Series functionalities offered by the pandas directly.

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

Table 1.Sparkling Wine Sales Data

Now, we have our data ready for the Time Series Analysis.

Plot the data

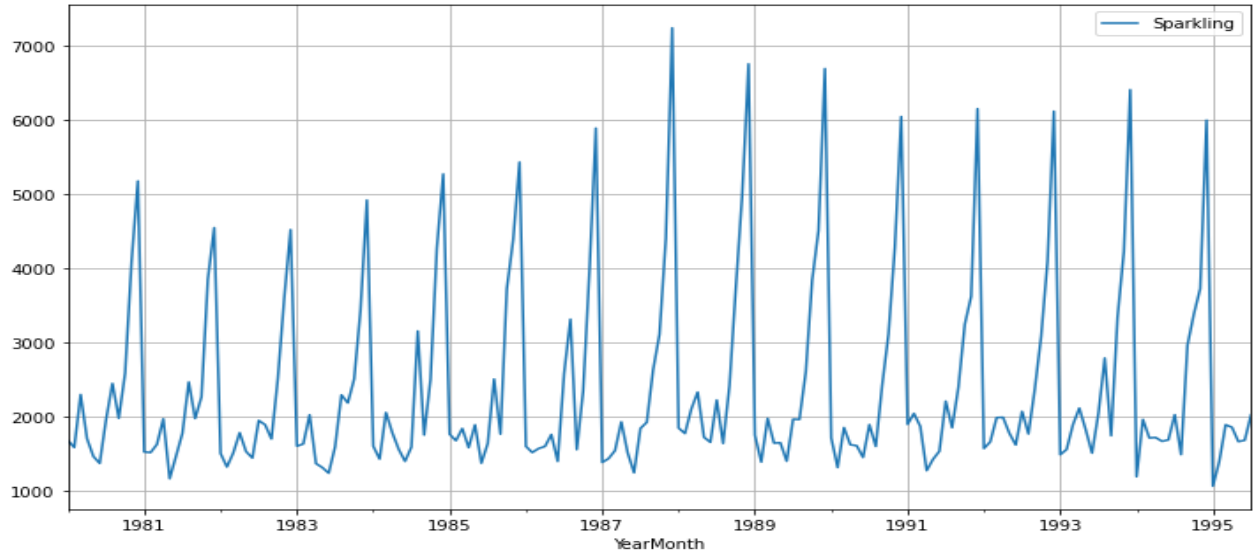


Figure 1. Sparkling Time Series plot

As it is Time Series forecasting analysis, data should be continuous without any null values or break in the data. Also, it has to be sequential. Upon checking, there are no null values present in the given dataset and good for our analysis.

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

Exploratory Data Analysis

	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

Table 2.Sparkling Wine Sales Summary

The average unit sale of Sparkling wine is 2402, whereas the minimum sale and maximum sale is 1070 and 7242 units respectively.

Yearly Boxplot

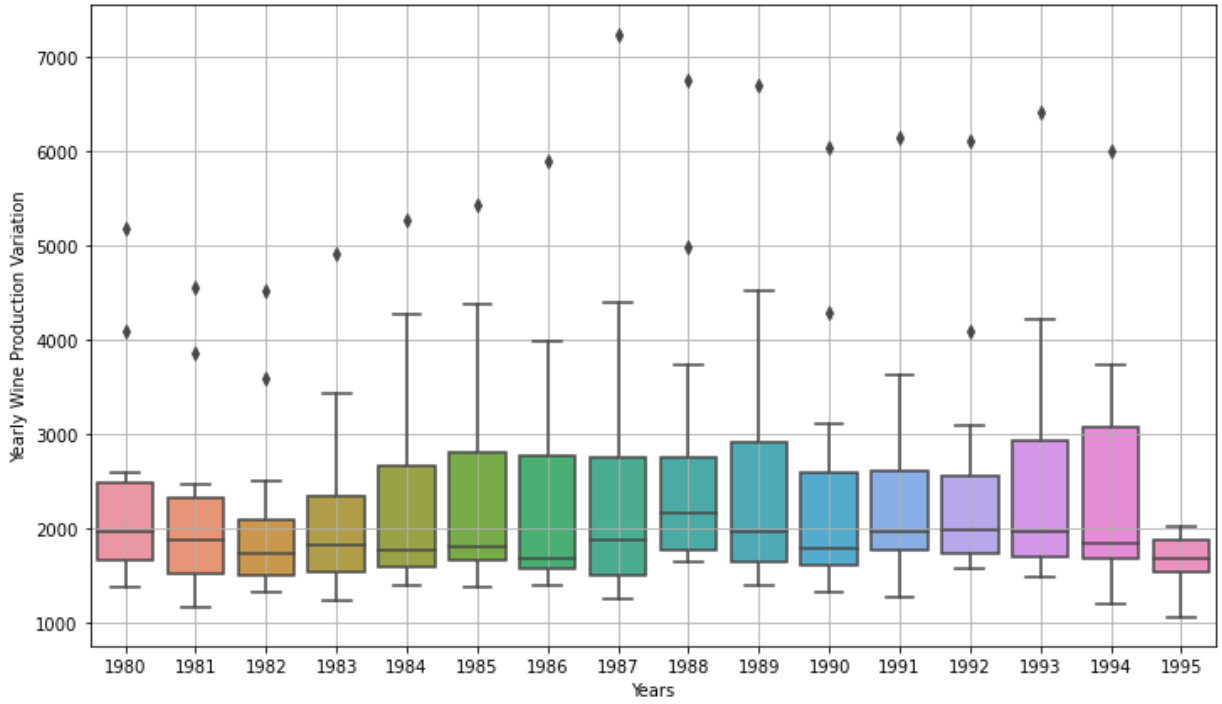


Figure 2. Sparkling Yearly Boxplot

We see, sparkling wine sales is high from the year 1984 to 1990 and 1993/1994.

Monthly Boxplot

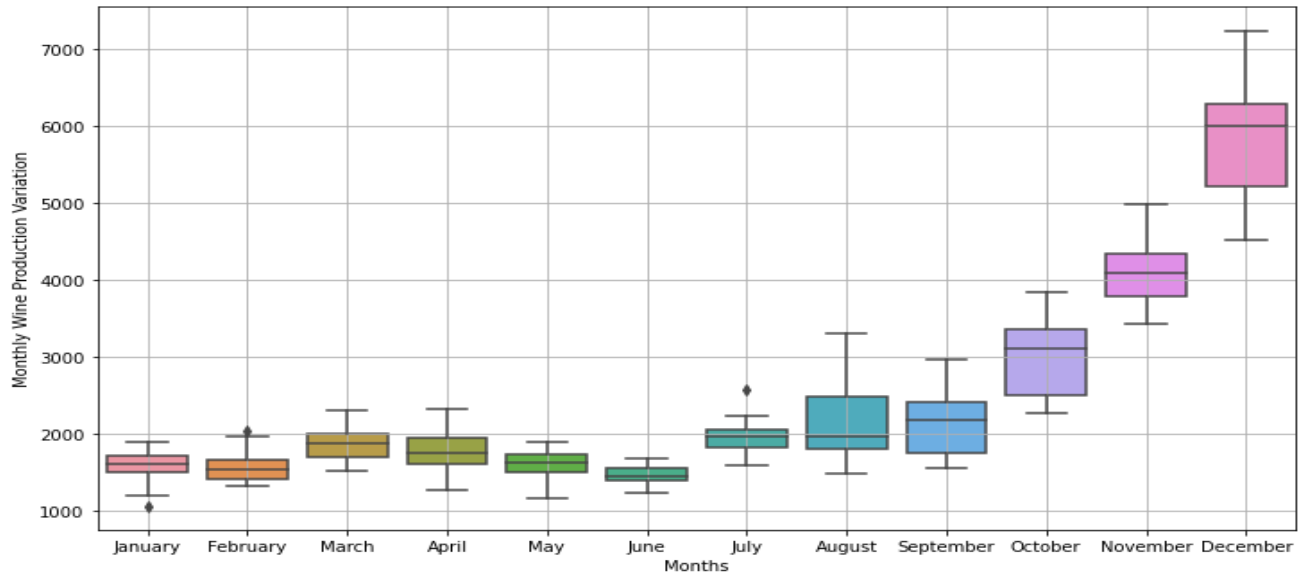


Figure 3. Sparkling Monthly Boxplot

The Sparkling wine sale is at its peak during the last 3 months of the year. i.e., when the winter starts or festival season starts.

Sparkling Time Series Plot

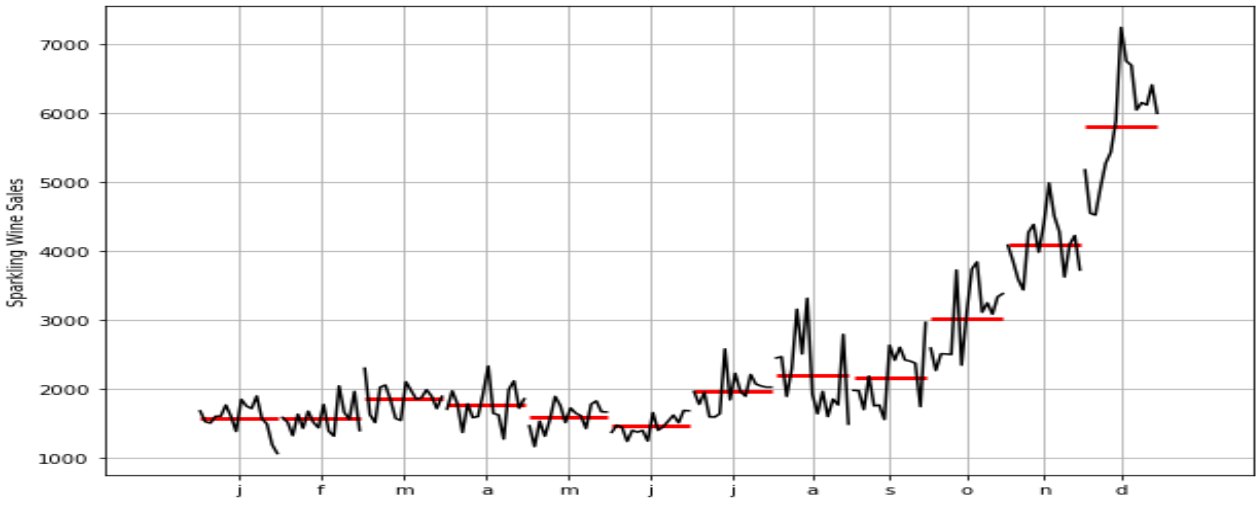


Figure 4. Sparkling Monthly Time Series plot

From the time series plot, we observed that Sparkling sales is exponentially high on October, November and December of every year.

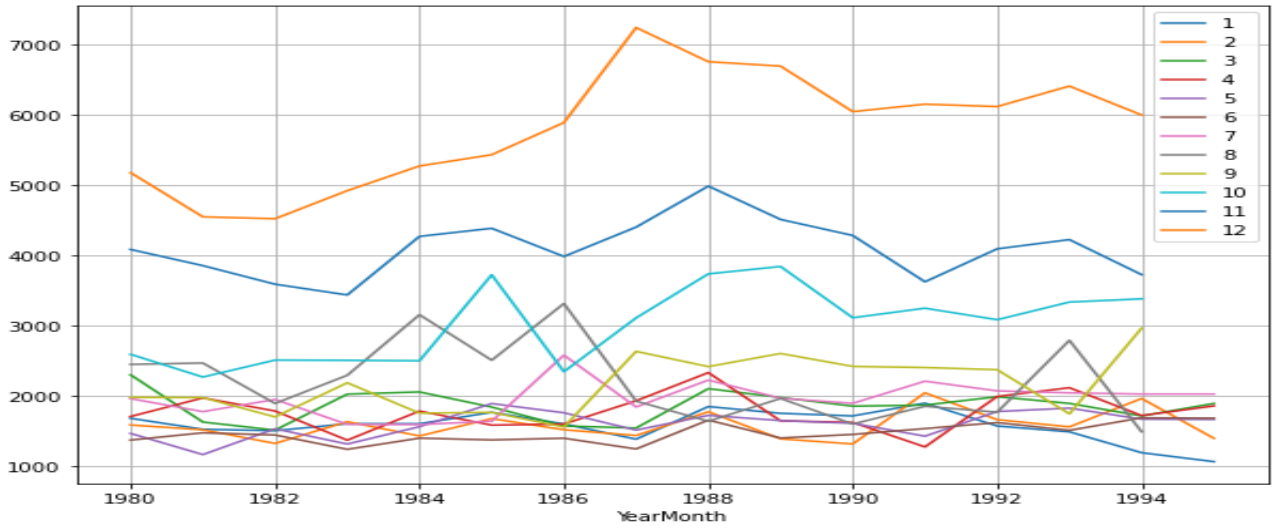


Figure 5. Sparkling Monthly Sales Across Years plot

Even across the years, the sparkling wine sales are high during the last 3 months of the year. Otherwise, sales are almost consistent and within 2500 units.

Sum of Sales at the end of every Year

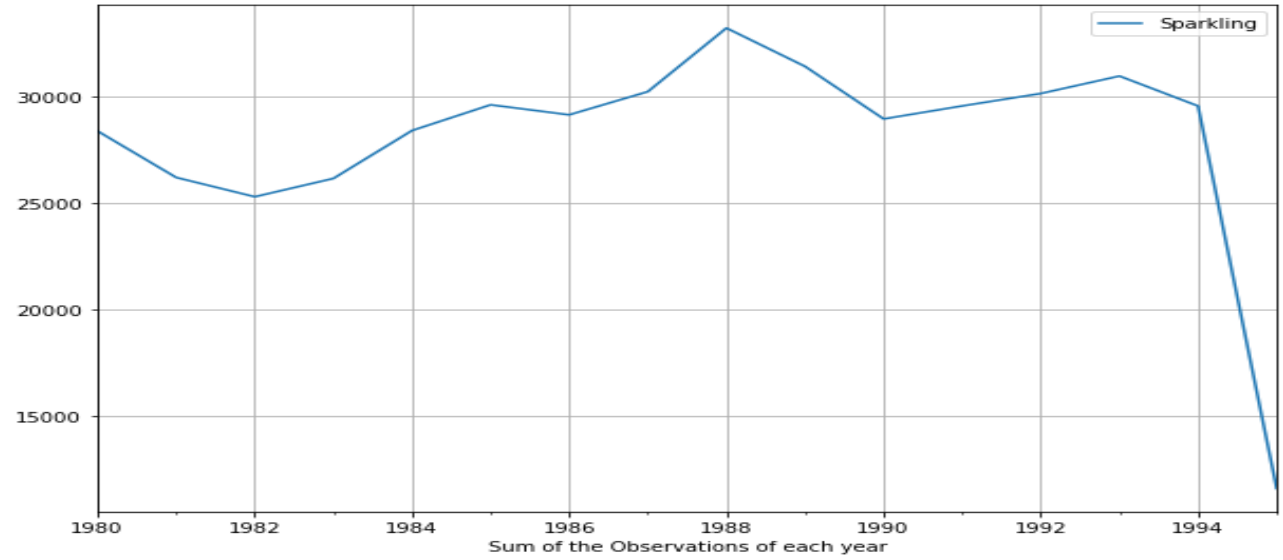


Figure 6. Sparkling Year End Sum of Sales

The sum of sales is high on the year 1988 and on the year 1994 it is abruptly coming down and it could be due to the data's available only till the month of 1994 July. As we observed, the sales is picking up during Q4 of the year. So, possibly the curve would be flatten out if we have the data's for the rest of the year 1994.

Average of Sales at the end of every Year

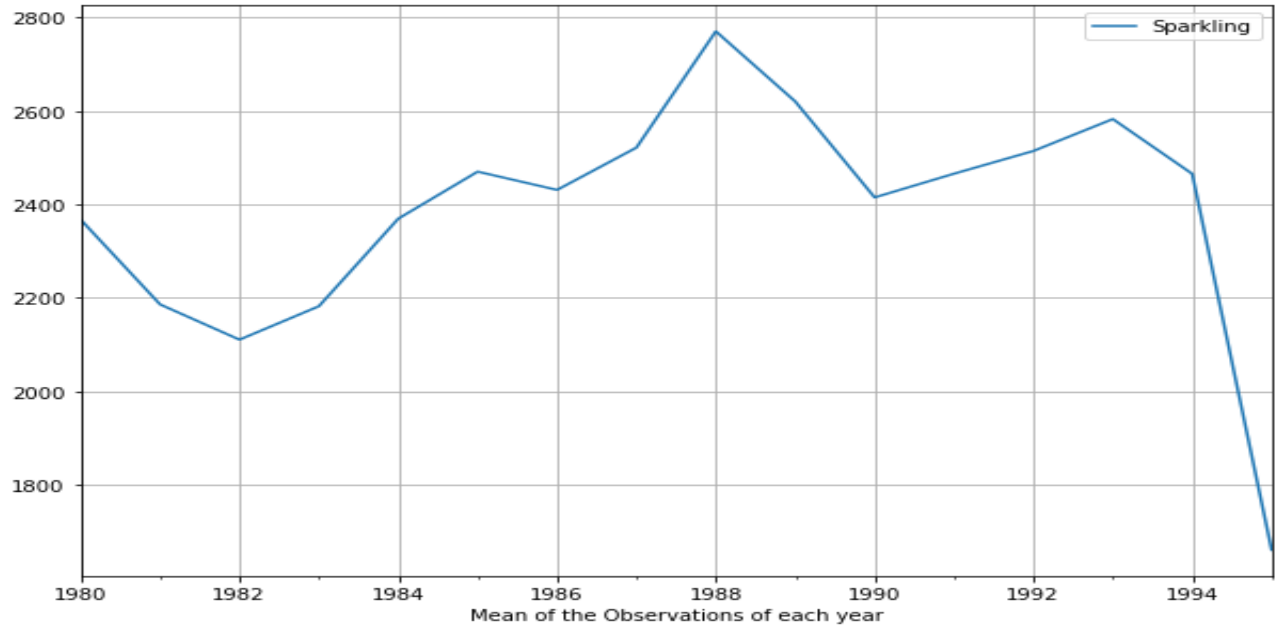


Figure 7. Sparkling Year End Average Sales

Average Quarterly Sales

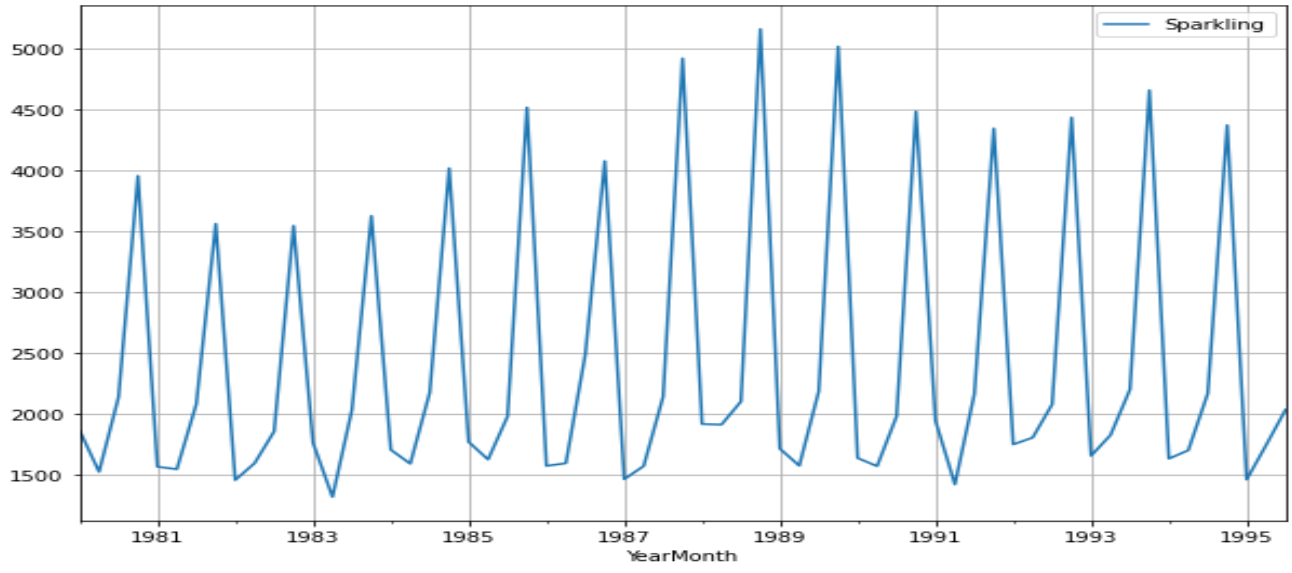


Figure 8. Sparkling Quarterly Average Sales

It clearly indicates that there is a sharp spike when sales approaching the year end.

Average Sales and Percentage Change in Sales

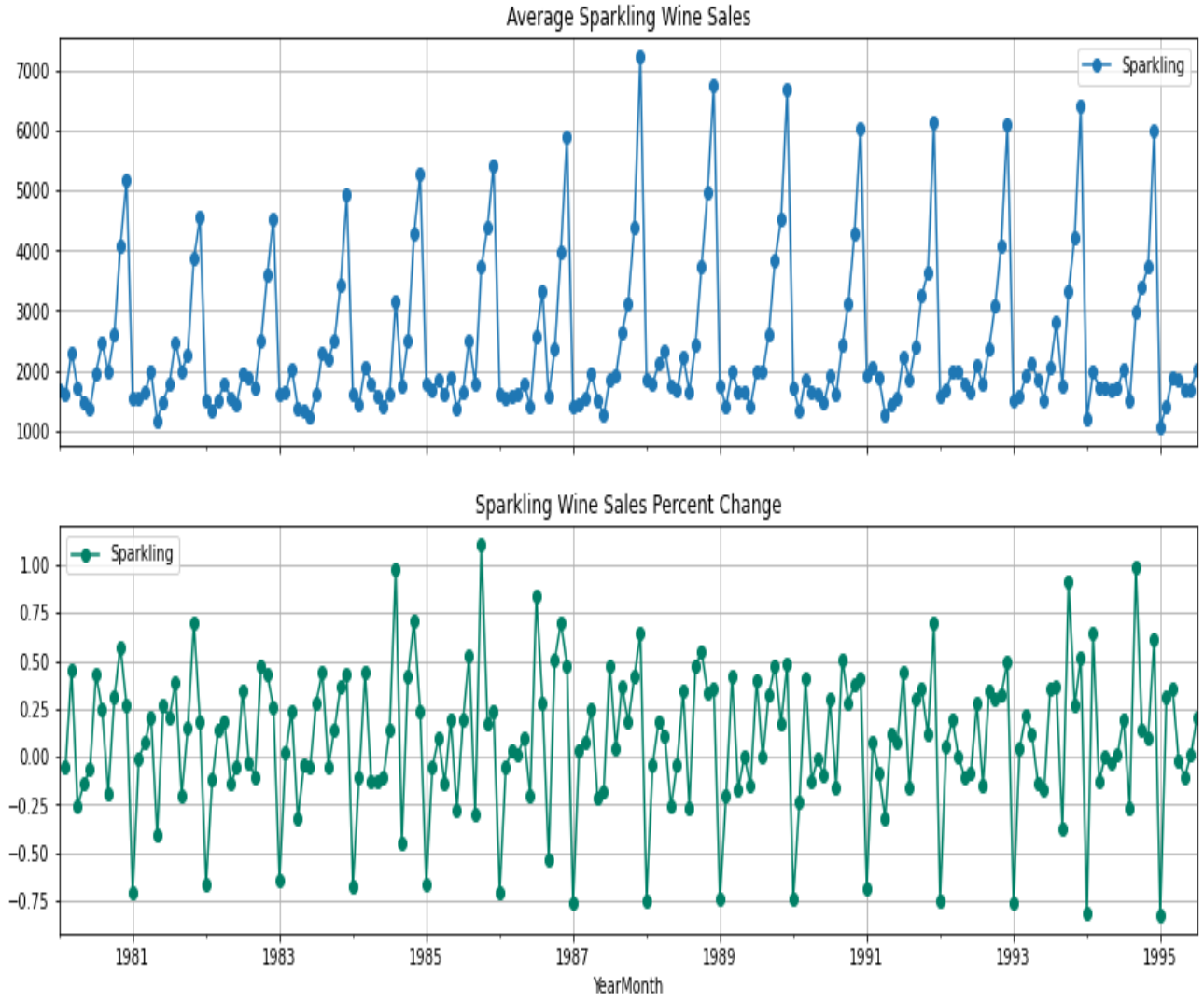


Figure 9. Sparkling Average and Percentage Changes Sales

- The average sale during the year end is 5K and gone up to 7K after 1988.
- On each year, there is a 50% increase in the sales. During the year end of 1986, the sales percentage change gone above 100%. Few times have touched the 100% change in the sales percentage.

Additive Model Decomposition

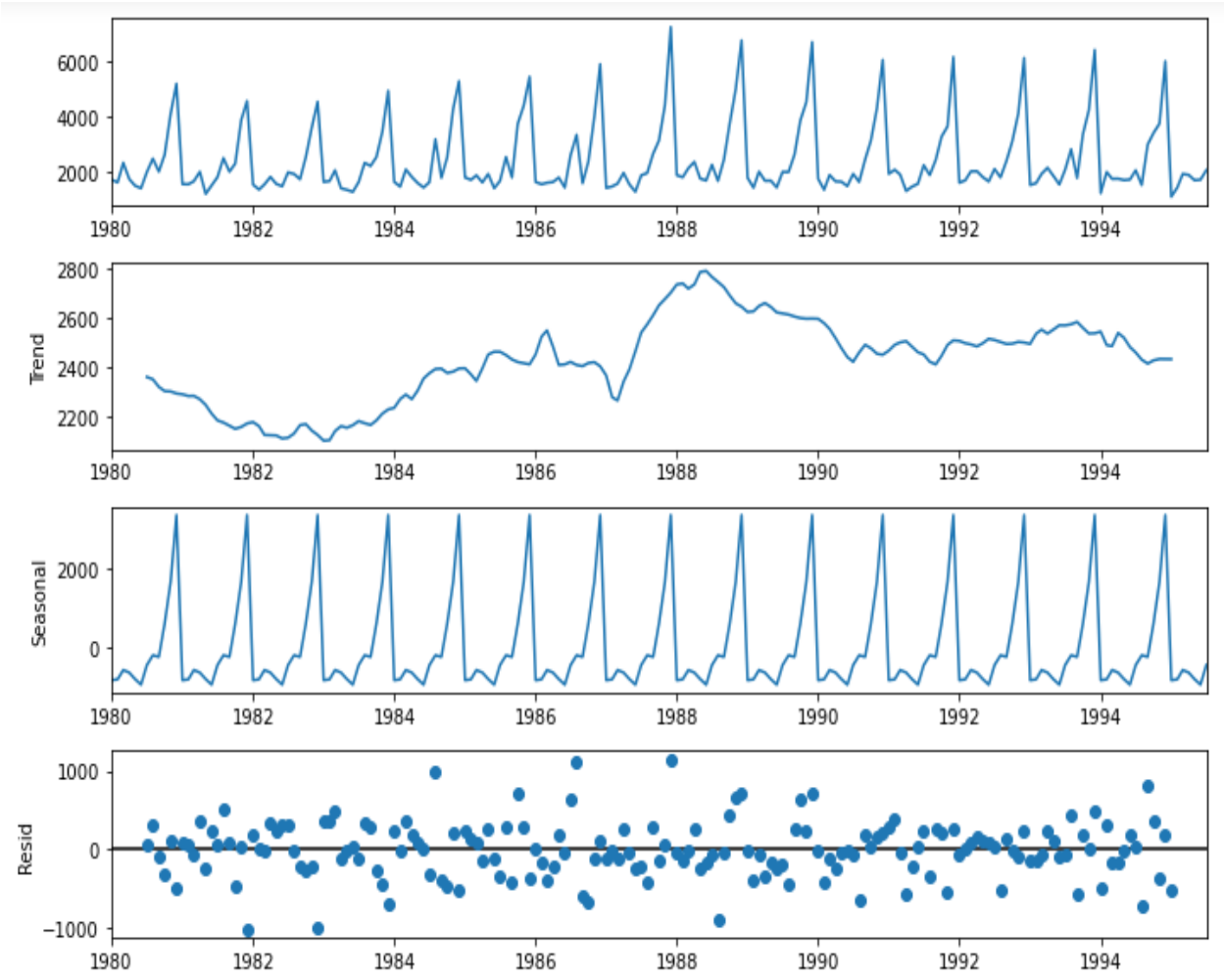


Figure 10. Sparkling Time Series Decomposition

There is an increasing trend after 1983 and the sales coming down after 1986. Abruptly, there is an exponential increase of sales on 1987 year end and it is consistent till 1990 and started flattening out from 1990 to 1994. Seasonality is present in the Sparkling dataset. The residual forms a slight pattern for both 'additive' as well as 'multiplicative' method. However, the seasonality does not go up with the trend assuming this is possibly an 'additive' method.

Time Series without Seasonality plots

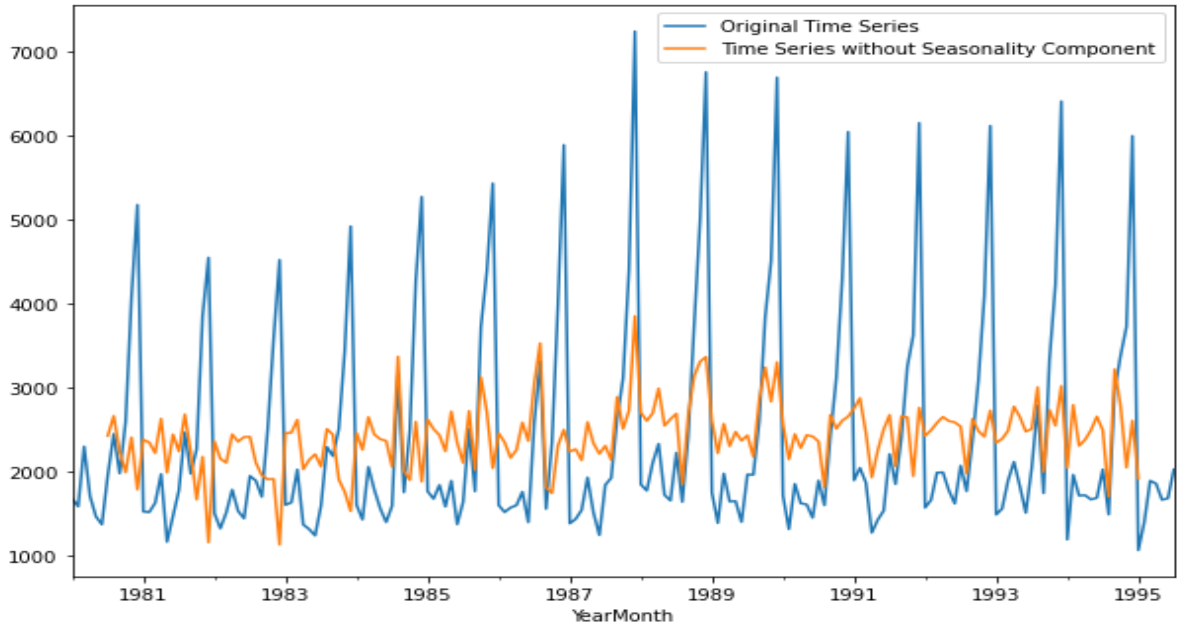


Figure 11. Sparkling Time Series without Seasonality

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

Splitting the Data

Let's split the training data till 1990 December and test data start from beginning January 1991.

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

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

Table 3. Sparkling Training and Test Data

In training, there are 132 records and in testing we got 55 records.

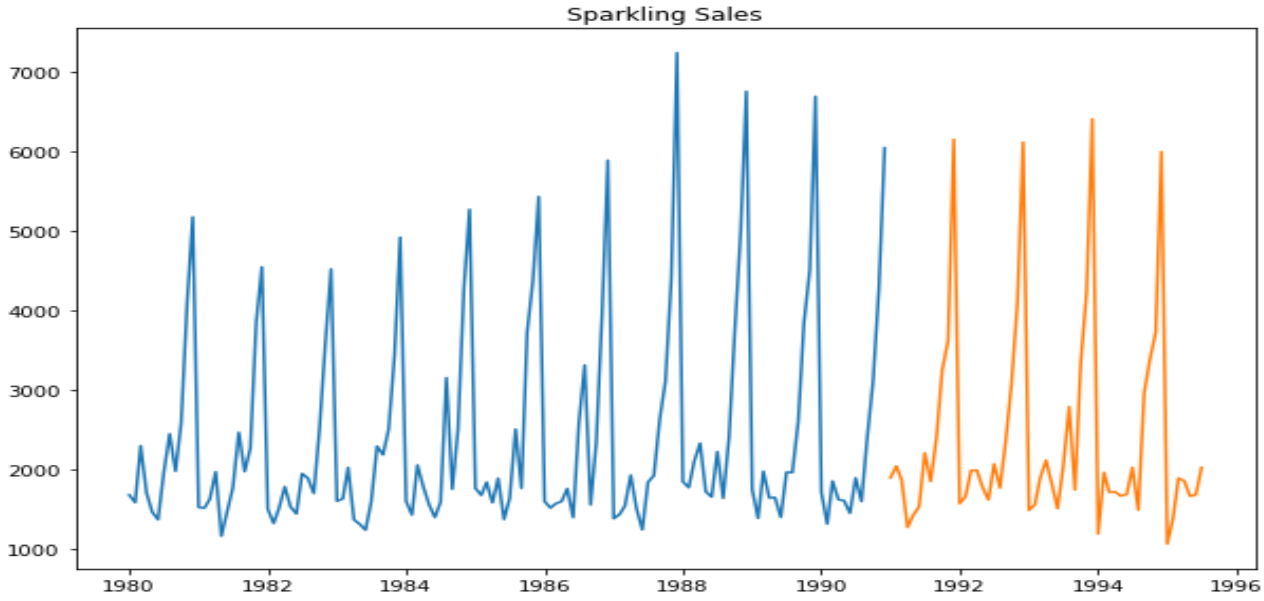


Figure 12.Sparkling Training and Testing Plot

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.

Building Models

Linear Regression

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

```
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, 3
4, 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, 18
3, 184, 185, 186, 187]
```

First few rows of Training Data

YearMonth	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

Last few rows of Training Data

YearMonth	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

First few rows of Test Data

YearMonth	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 Test Data

YearMonth	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

Now that our training and test data has been modified, let us go ahead use LinearRegression to build the model on the training data and test the model on the test data. Let's instantiate the Linear Regression Model using the below function and fit the model using training time and Sparkling Sales data.

```
lr = LinearRegression()

lr.fit(LinearRegression_train[['time']],LinearRegression_train['Sparkling'].values)
```

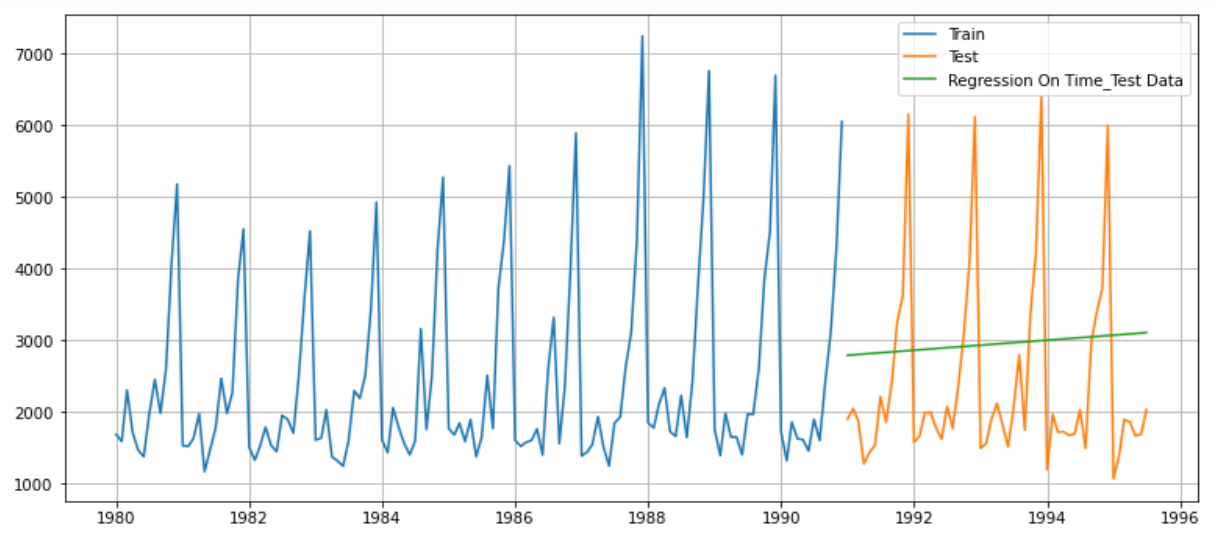


Figure 13.Sparkling Linear Regression Prediction Plot

From the above plot we can conclude that Linear Regression model doesn't predict well with the test data. However, let's continue with the model evaluation and store the results in the dataframe.

Model Evaluation on Test Data

For RegressionOnTime forecast on the Test Data, RMSE is 1389.135

	Test RMSE
RegressionOnTime	1389.135175

Table 4. Sparkling Linear Regression RMSE Table

Naïve Approach

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

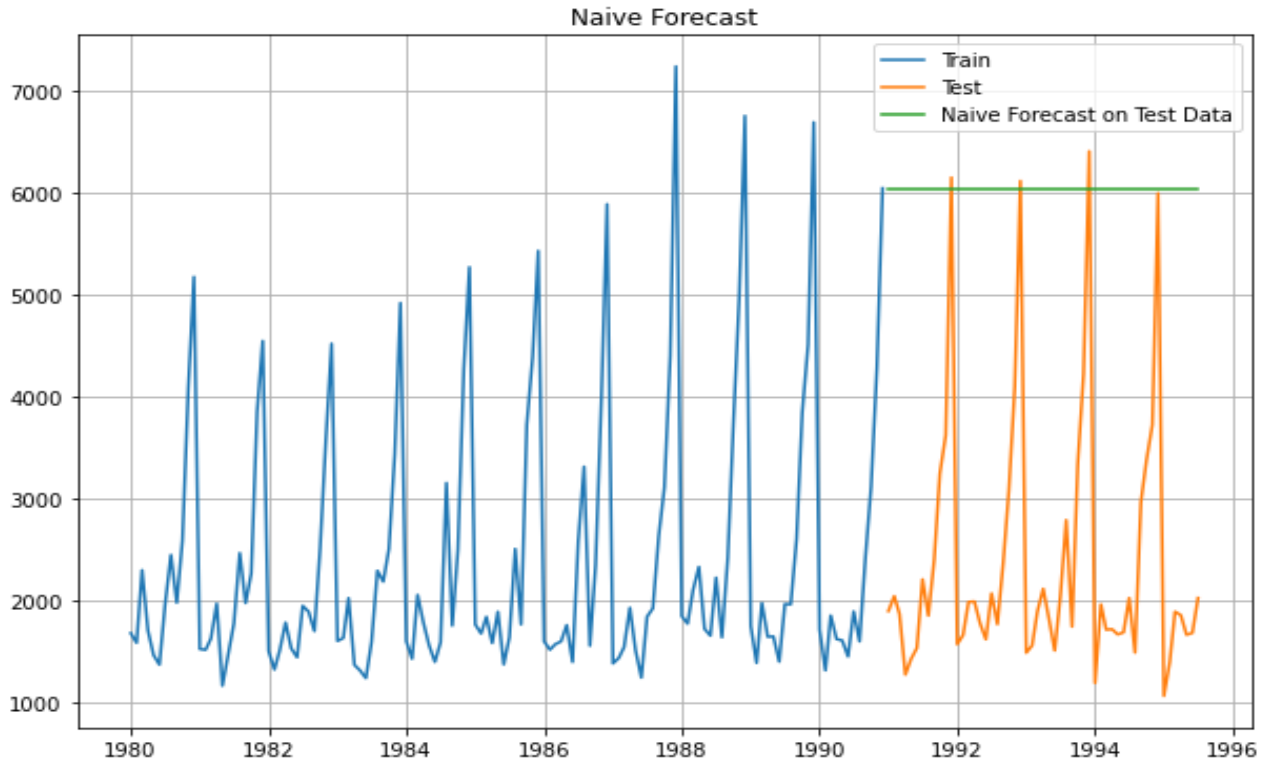


Figure 14.Sparkling Naïve method Prediction Plot

From the above plot we can conclude that Naïve approach doesn't predict well with the test data. It predicts as a straight line. However, let's continue with the model evaluation and store the results in the dataframe.

Model Evaluation on Test Data

For Naive forecast on the Test Data, RMSE is 3864.279

	Test RMSE
RegressionOnTime	1389.135175
NaiveModel	3864.279352

Table 5. Sparkling Naïve RMSE Table

Simple Average

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

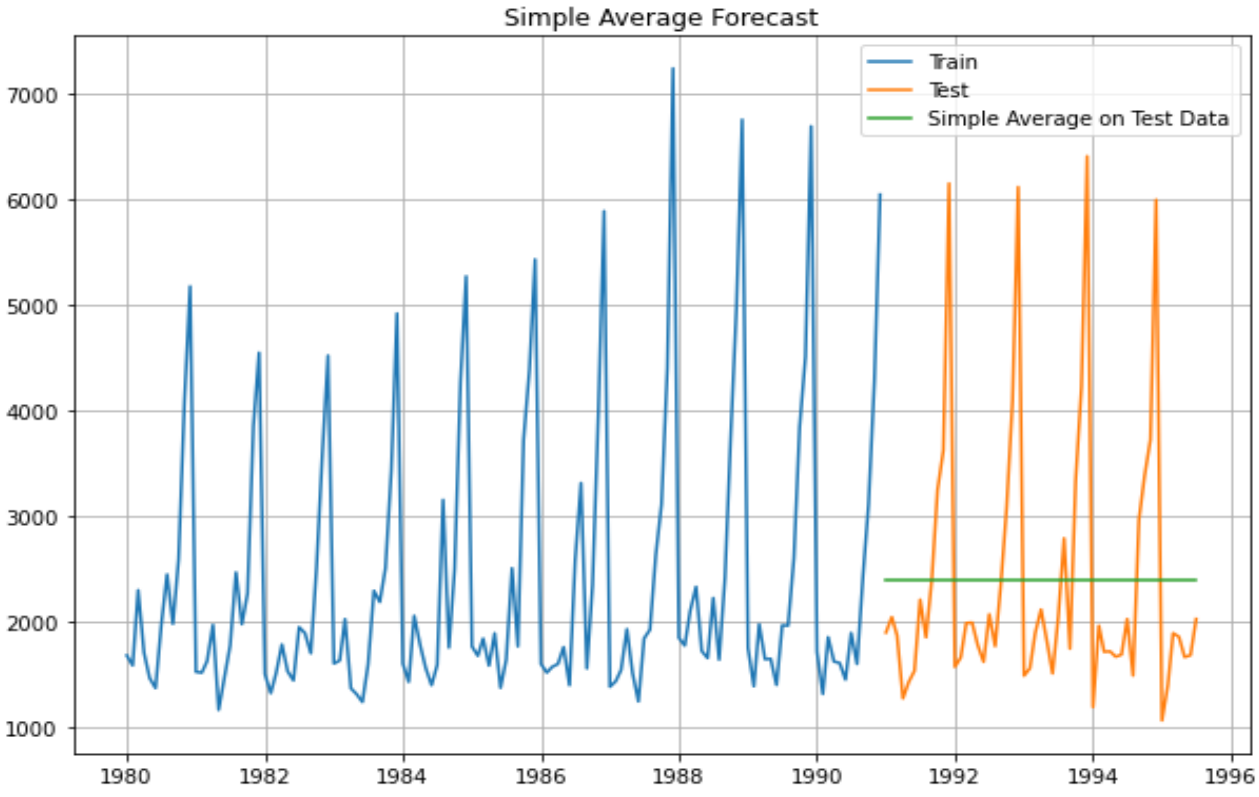


Figure 15.Sparkling Simple Average Prediction Plot

From the above plot we can conclude that Simple Average approach doesn't predict well with the test data. It predicts as a straight line. However, let's continue with the model evaluation and store the results in the dataframe.

Model Evaluation on Test Data

For Simple Average forecast on the Test Data, RMSE is 1275.082

	Test RMSE
RegressionOnTime	1389.135175
NaiveModel	3864.279352
SimpleAverageModel	1275.081804

Table 6. Sparkling Simple Average RMSE Table

Moving Average (MA)

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

For Moving Average, we are going to average over the entire data.

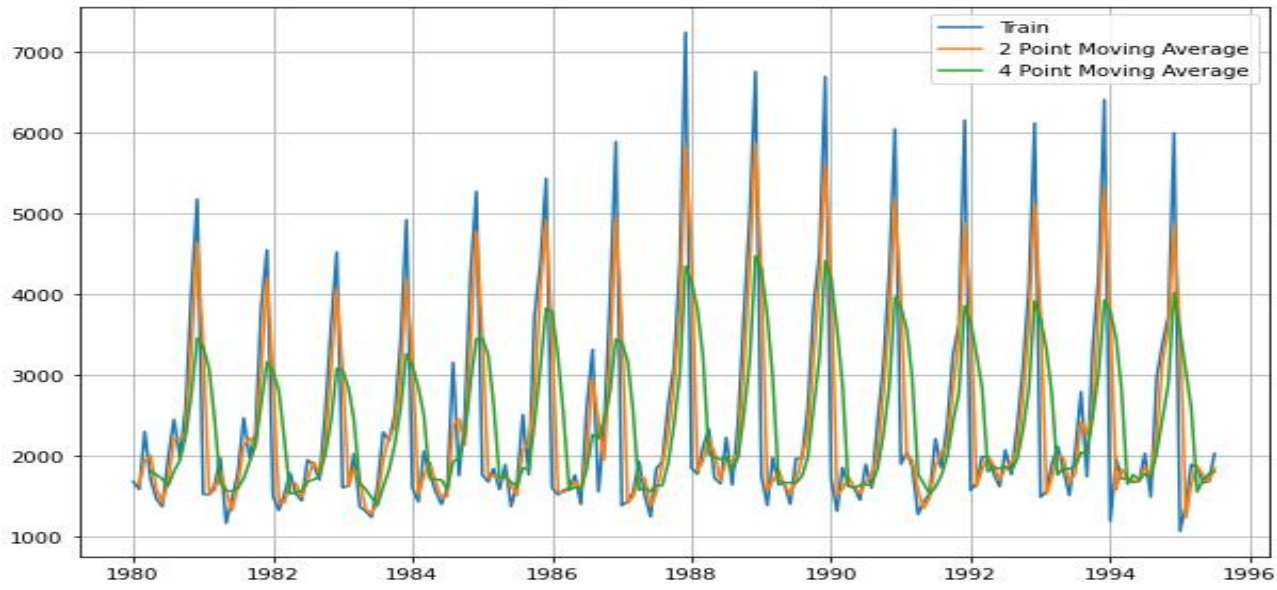


Figure 16.Sparkling Moving Average for whole data

From the above plot we can conclude that Moving Average approach is the best model built so far. Precisely, '2 point moving average' predicts well with the test data. Let's train the model using the Moving Average dataset and predict the test data.

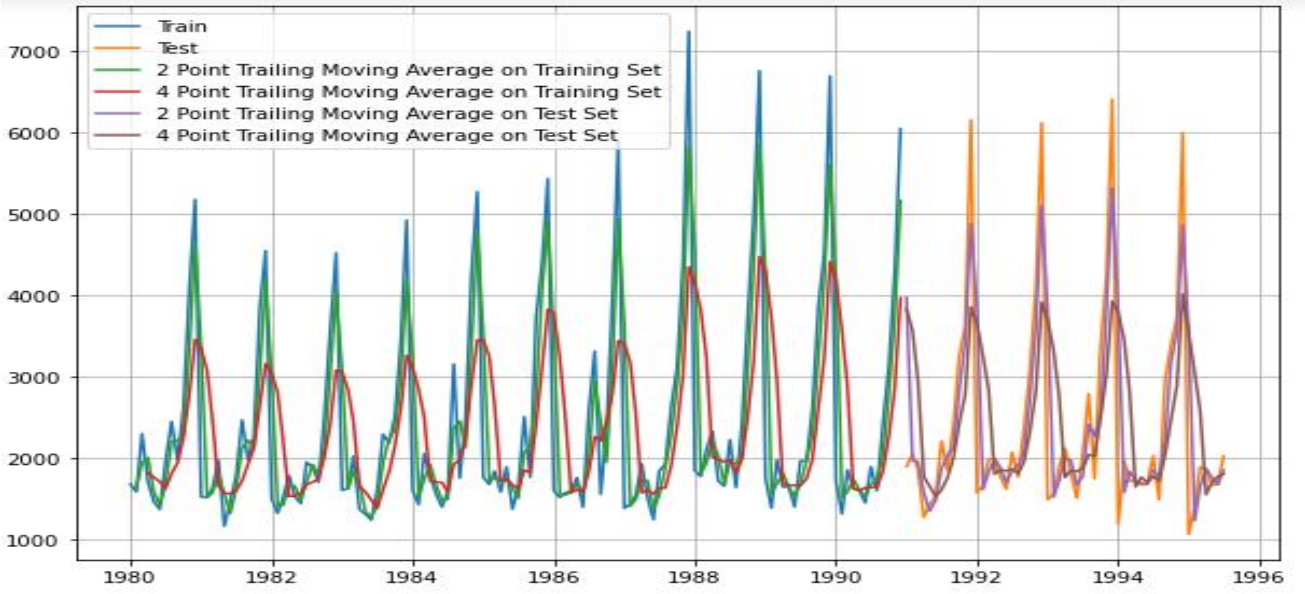


Figure 17.Sparkling Moving Average Prediction Plot

Even in the testing records, Moving Average model performs well.

Let's continue with the model evaluation and store the results in the dataframe.

Model Evaluation on Test Data

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

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

	Test RMSE
RegressionOnTime	1389.135175
NaiveModel	3864.279352
SimpleAverageModel	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694

Table 7. Sparkling Moving Average RMSE Table

So far, we have got the best RMSE value for '2 point rolling average' method.

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

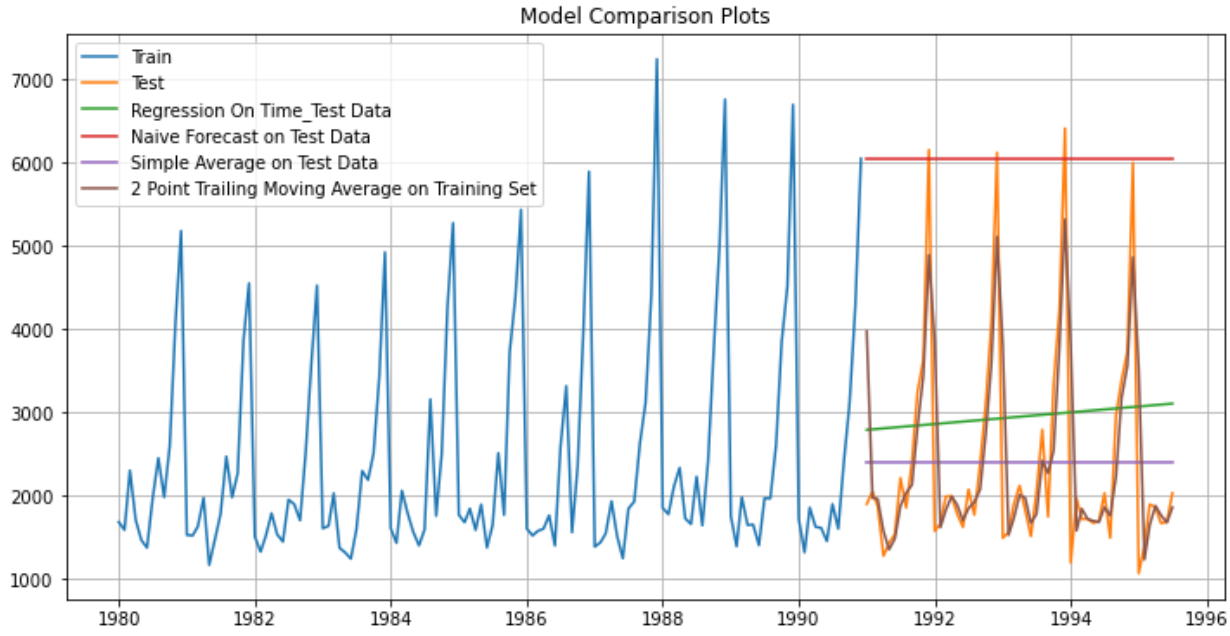


Figure 18.Model Comparison plots

The above plot clearly shows that the 2 point trailing Moving Average method (*curve in brown color*) predicts the test data well.

Simple Exponential Smoothing

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

Let's instantiate the SES SimpleExpSmoothing() function importing from statsmodels library.

```
model_SES =
SimpleExpSmoothing(SES_train['Sparkling'],initialization_method='estimated')

model_SES_autofit = model_SES.fit(optimized=True)
```

Fitting the Simple Exponential Smoothing model and asking python to choose the optimal parameters

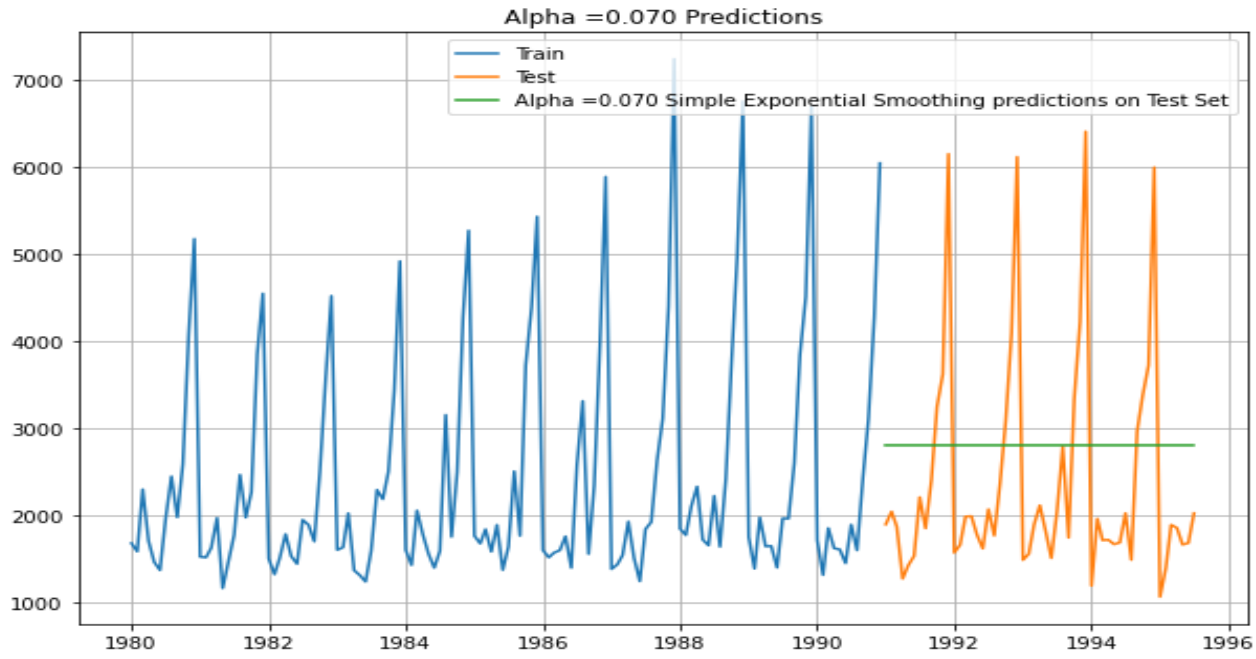


Figure 19.Sparkling auto alpha value SES Prediction Plot

From the above plot we can conclude that SES model approach doesn't predict well with the test data. It predicts as a straight line. However, let's continue with the model evaluation and store the results in the dataframe.

Model Evaluation on Test Data

For Alpha =0.070 Simple Exponential Smoothing Model forecast on the Test Data, RMSE is 1338.008

	Test RMSE
RegressionOnTime	1389.135175
NaiveModel	3864.279352
SimpleAverageModel	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
Alpha=0.070,SimpleExponentialSmoothing	1338.007771

Table 8. Sparkling auto SES RMSE Table

Setting different alpha values. Remember, the higher the alpha value more weightage is given to the more recent observation. That means, what happened recently will happen again. We will run a loop with different alpha values to understand which particular value works best for alpha on the test set.

Since the best fit alpha is 0.070, let's run the loop with the lower values such as 0.001, 0.01, 0.05, 0.1, 0.2, and 0.3 and capture the test data RMSE values in the dataframe.

	Alpha Values	Train RMSE	Test RMSE
1	0.010	1397.988872	1286.648058
2	0.050	1324.401979	1316.359347
3	0.100	1336.428478	1375.393335
0	0.001	1549.799408	1497.004890
4	0.200	1356.950475	1595.206839
5	0.300	1359.953398	1935.507132

Table 9. Sparkling tuning SES RMSE Table

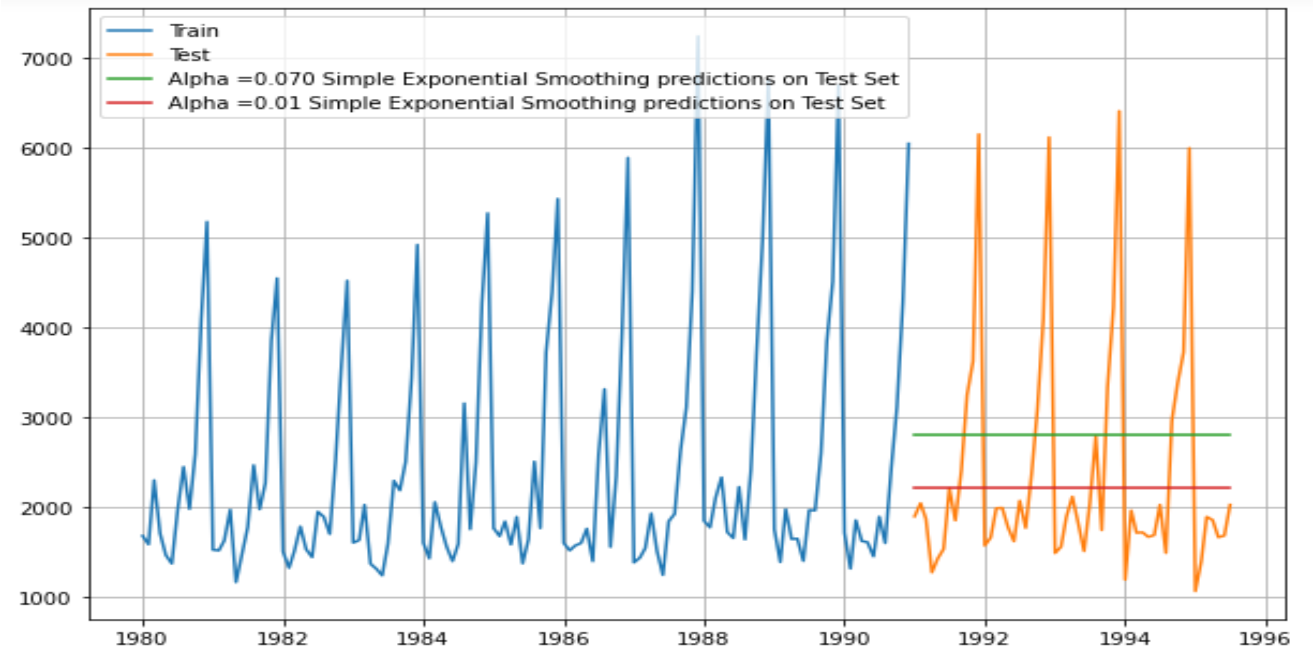


Figure 20. Sparkling SES Prediction Plot

As we presumed, SES is not the good fit for our Sparkling data.

	Test RMSE
RegressionOnTime	1389.135175
NaiveModel	3864.279352
SimpleAverageModel	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
Alpha=0.070,SimpleExponentialSmoothing	1338.007771
Alpha=0.01,SimpleExponentialSmoothing	1286.648058

Table 10. Sparkling SES RMSE Table

Double Exponential Smoothing

Double Exponential Smoothing model is suitable to model the time series with trend but without seasonality. The Holt's linear exponential smoothing displays a constant trend indefinitely into the future. But, the trend and seasonality is present in our Sparkling data. Empirical evidence shows that the Holt's linear method tends to over-forecast.

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

Let's instantiate the Holt() function from statsmodels library.

```
model_DES = Holt(DES_train['Sparkling'])
```

Let's run the loop with the values from 0.1 to 1.0 for both α and β and capture the test data RMSE values in the dataframe.

	Alpha Values	Beta Values	Train RMSE	Test RMSE
0	0.1	0.1	1382.520870	1778.564670
1	0.1	0.2	1413.598835	2599.439986
10	0.2	0.1	1418.041591	3611.763322
2	0.1	0.3	1445.762015	4293.084674
20	0.3	0.1	1431.169601	5908.185554

Table 11. Sparkling tuning DES RMSE Table

The best RMSE values of DES are 0.1 for both α and β.

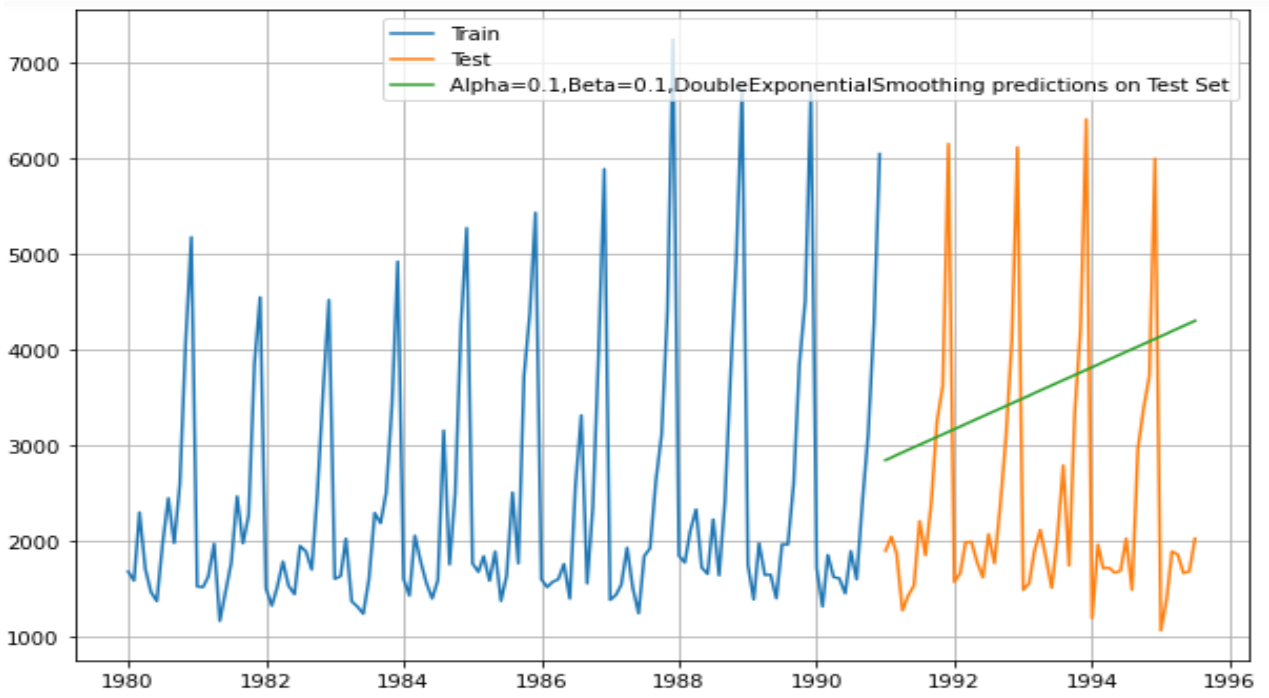


Figure 21.Sparkling DES Prediction Plot

From the above plot we can conclude that DES model approach doesn't predict well with the test data. It predicts as a straight line. However, let's continue with the model evaluation and store the results in the dataframe.

	Test RMSE
RegressionOnTime	1389.135175
NaiveModel	3864.279352
SimpleAverageModel	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
Alpha=0.070,SimpleExponentialSmoothing	1338.007771
Alpha=0.01,SimpleExponentialSmoothing	1286.648058
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	1778.564670

Table 12. Sparkling DES RMSE Table

Triple Exponential Smoothing

Triple exponential smoothing is used to handle the time series data containing a seasonal component. This method is based on three smoothing equations: stationary component, trend, and seasonal. Both seasonal and trend can be additive or multiplicative. Our sparkling data has both trend and seasonality and possibly the good fit for this data.

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

Let's instantiate the ExponentialSmoothing () function from statsmodels library.

```
model_TES =
ExponentialSmoothing(TES_train['Sparkling'],trend='additive',seasonal='multip
licative')

model_TES_autofit = model_TES.fit()
```

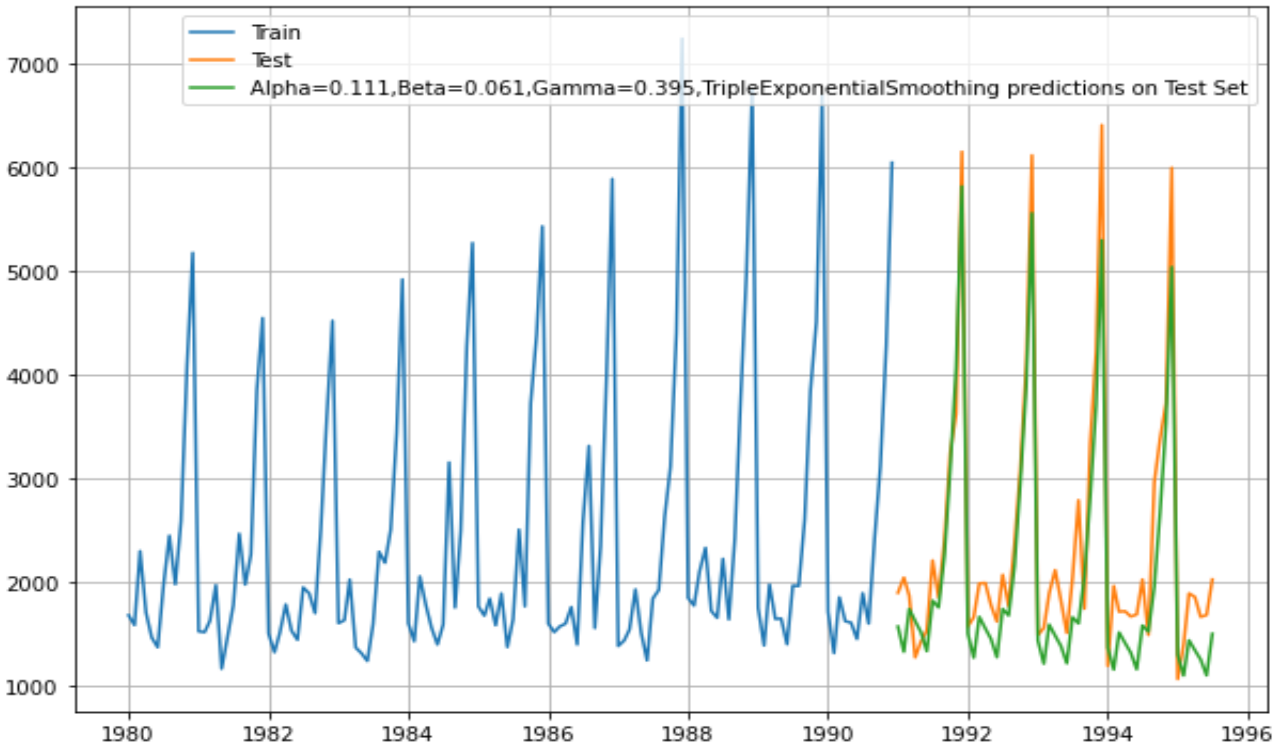


Figure 22.Sparkling auto TES values Prediction Plot

As we presumed, the above plot is good fit for our business problem.

Model Evaluation on Test Data

For Alpha=0.111, Beta=0.061,Gamma=0.395, Triple Exponential Smoothing Model forecast on the Test Data, RMSE is 469.592

	Test RMSE
RegressionOnTime	1389.135175
NaiveModel	3864.279352
SimpleAverageModel	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
Alpha=0.070,SimpleExponentialSmoothing	1338.007771
Alpha=0.01,SimpleExponentialSmoothing	1286.648058
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	1778.564670
Alpha=0.111,Beta=0.061,Gamma=0.395,TripleExponentialSmoothing	469.591516

Table 13. Sparkling auto TES RMSE Table

Setting different Alpha, Beta and Gamma values. Let's run the loop with values and capture the test data RMSE values in the dataframe.

	Alpha Values	Beta Values	Gamma Values	Train RMSE	Test RMSE
301	0.4	0.1	0.2	389.772245	336.715250
211	0.3	0.2	0.2	395.529174	350.145204
110	0.2	0.2	0.1	405.333164	352.571689
200	0.3	0.1	0.1	394.630053	352.607849
20	0.1	0.3	0.1	414.423963	354.534561

Table 14. Sparkling tuning TES RMSE Table

Upon tuning the α, β and γ values, the root mean square error is coming down compared to the autofit parameters, which is good sign of our model would perform well with the production records.

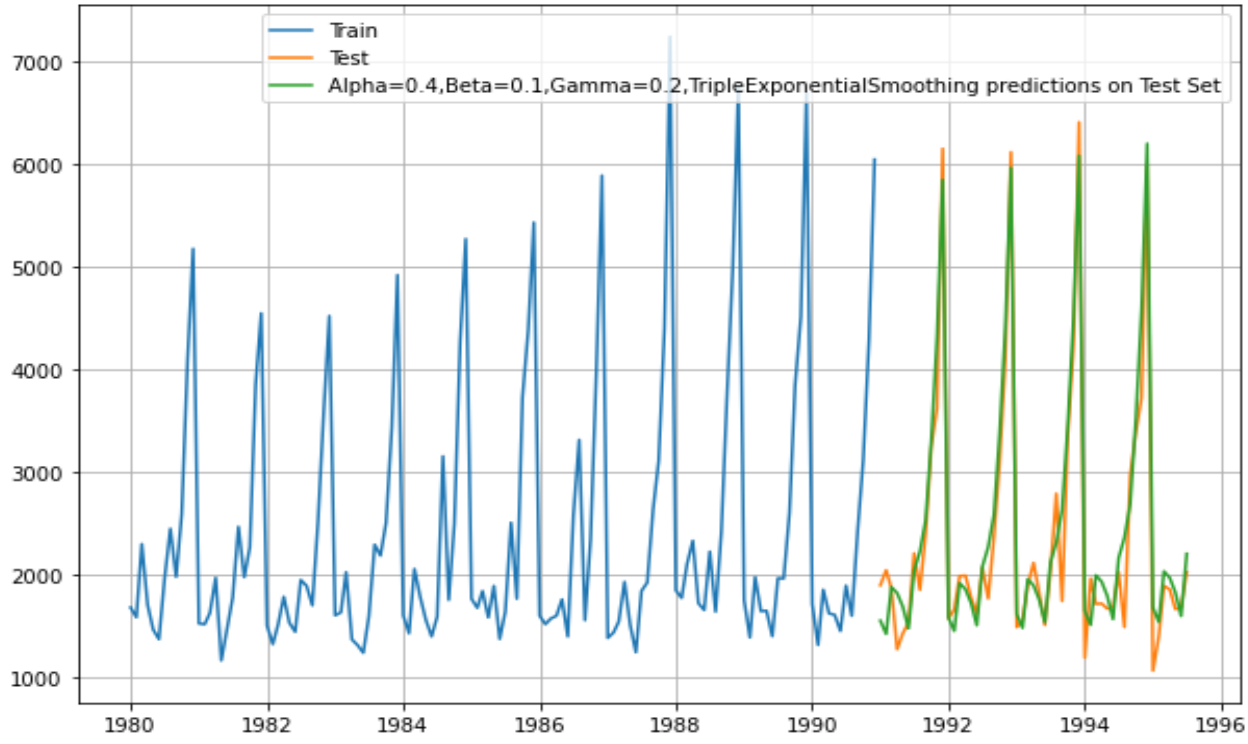


Figure 23.Sparkling TES Prediction Plot

RMSE Summary of the Models for Sparkling Test data

	Test RMSE
RegressionOnTime	1389.135175
NaiveModel	3864.279352
SimpleAverageModel	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
Alpha=0.070,SimpleExponentialSmoothing	1338.007771
Alpha=0.01,SimpleExponentialSmoothing	1286.648058
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	1778.564670
Alpha=0.111,Beta=0.061,Gamma=0.395,TripleExponentialSmoothing	469.591516
Alpha=0.4,Beta=0.1,Gamma=0.2,TripleExponentialSmoothing	336.715250

Table 15. Sparkling TES RMSE Table

Optimized Model for Sparkling Data

	Test RMSE
Alpha=0.4,Beta=0.1,Gamma=0.2,TripleExponentialSmoothing	336.715250
Alpha=0.111,Beta=0.061,Gamma=0.395,TripleExponentialSmoothing	469.591516
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
SimpleAverageModel	1275.081804
Alpha=0.01,SimpleExponentialSmoothing	1286.648058
Alpha=0.070,SimpleExponentialSmoothing	1338.007771
RegressionOnTime	1389.135175
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	1778.564670
NaiveModel	3864.279352

Table 16. Best Sparkling RMSE Table

From the above table, TES model with the tuning parameters results out the low RMSE values. Also, the curve of TES model series is cope with the test data correctly.

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.

Stationarity Check using Dickey Fuller Test

Auto-regressive (AR) and moving average (MA) models are popular models that are frequently used for forecasting. AR and MA models are combined to create models such as auto-regressive moving average (ARMA) and auto-regressive integrated moving average (ARIMA) models. ARIMA models are basically regression models; auto-regression means regression of a variable on itself measured at different time periods.

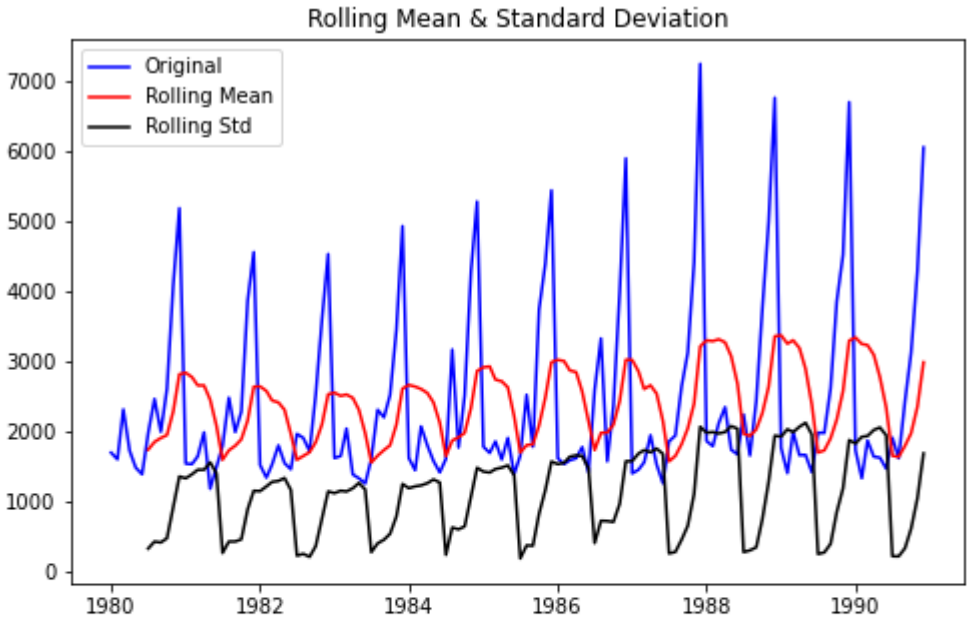
The main assumption of AR model is that the time series data is stationary.

A stationary time series is one whose statistical properties such as mean, variance, autocorrelation, etc. are all constant over time. When the time series data is not stationary, then we convert the non-stationary data before applying AR models.

The Augmented Dickey Fuller Test (ADF) is unit root test for stationarity. The null hypothesis is that time series is non-stationary. Alternative hypothesis is that time series is stationary.

From statsmodels library, lets import the adfuller function to perform ADF test.

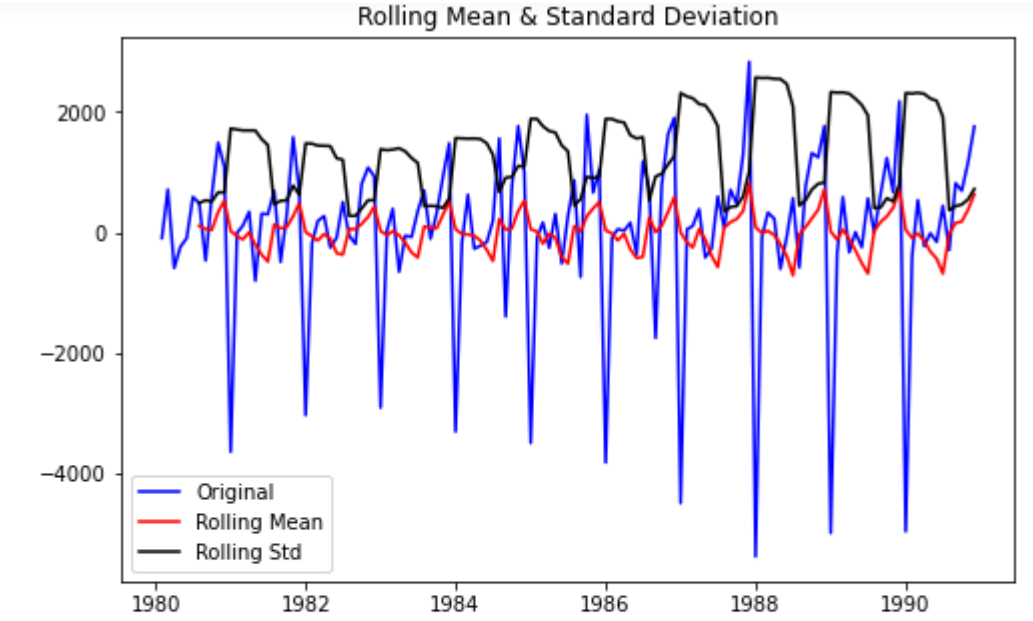
```
dftest = adfuller(timeseries, autolag='AIC')

test_stationarity(train['Sparkling'])
```




```
Results of Dickey-Fuller Test:
Test Statistic                  -2.061798
p-value                          0.567411
#Lags Used                      12.000000
Number of Observations Used    119.000000
Critical Value (1%)             -4.036934
Critical Value (5%)             -3.448049
Critical Value (10%)            -3.149068
dtype: float64
```

Figure 24.ADF Test for original Sparkling train data

Here, the p-value is greater than 0.05(alpha value). So, it is failed to reject the null hypothesis. i.e., data is not stationary. Taking the difference between consecutive observations is called a lag-1 difference to make it stationary. Let's take the first order difference in the data and run the ADF test again.




```
Results of Dickey-Fuller Test:
Test Statistic                -7.967842e+00
p-value                        8.479211e-11
#Lags Used                     1.100000e+01
Number of Observations Used    1.190000e+02
Critical Value (1%)           -4.036934e+00
Critical Value (5%)           -3.448049e+00
Critical Value (10%)          -3.149068e+00
dtype: float64
```

Figure 25.ADF Test after lag-1 difference in Sparkling train data

Now, the p-value is close to 0 and rejected the null hypothesis test. i.e., data is stationary and now we can apply the ARIMA models.

Stationary plot for Sparkling train data

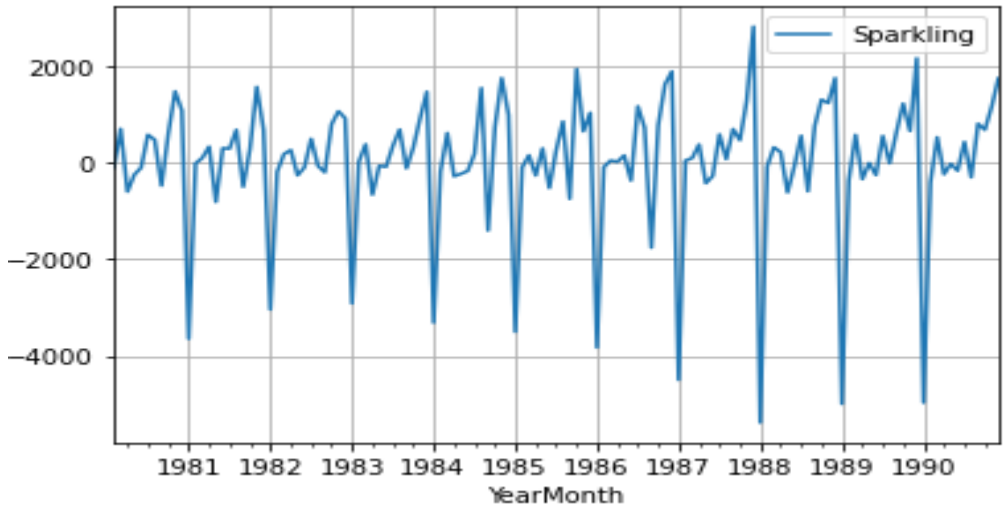


Figure 26.Stationary Sparkling Train data plot

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.

Building auto ARIMA model using AIC values

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

Specifically for ARIMA model, ARIMA (p, d, q) means that it you are describing some response variable (Y) by combining a 'p' order Auto-Regressive model and a 'q' order Moving Average model. A good way to think about it is (AR, I, MA).

Through 'itertools' python package, let's create the series of loop for ARIMA model parameters.

- p and q values from 0 to 3
- d=1 because it becomes non-stationary to stationary after differencing it with 1 time itself.

Let's import the ARIMA function from statsmodels.tsa.arima.model time series library and run through the different combinations of pq values with d=1

We have got the best AIC values for ARIMA(2,1,2) model.

	param	AIC
10	(2, 1, 2)	2213.509212
15	(3, 1, 3)	2221.464720
14	(3, 1, 2)	2230.908233

Table 17. AIC table for Sparkling AUTO ARIMA

Fit this model and see the summary below.

SARIMAX Results

Dep. Variable:	y	No. Observations:	132
Model:	ARIMA(2, 1, 2)	Log Likelihood	-1116.419
Date:	Wed, 09 Feb 2022	AIC	2242.837
Time:	16:15:46	BIC	2257.213
Sample:	0	HQIC	2248.679
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.5166	0.114	4.539	0.000	0.294	0.740
ar.L2	-0.1848	0.206	-0.896	0.370	-0.589	0.219
ma.L1	-1.9946	0.108	-18.483	0.000	-2.206	-1.783
ma.L2	0.9969	0.107	9.298	0.000	0.787	1.207
sigma2	1.338e+06	1.67e-07	8.03e+12	0.000	1.34e+06	1.34e+06

Ljung-Box (L1) (Q):	0.05	Jarque-Bera (JB):	11.09
Prob(Q):	0.82	Prob(JB):	0.00
Heteroskedasticity (H):	2.76	Skew:	0.40
Prob(H) (two-sided):	0.00	Kurtosis:	4.18

Figure 27.Statistical summary for Sparkling AUTO ARIMA

From the statistical summary, 2nd component of auto regression is not significant.

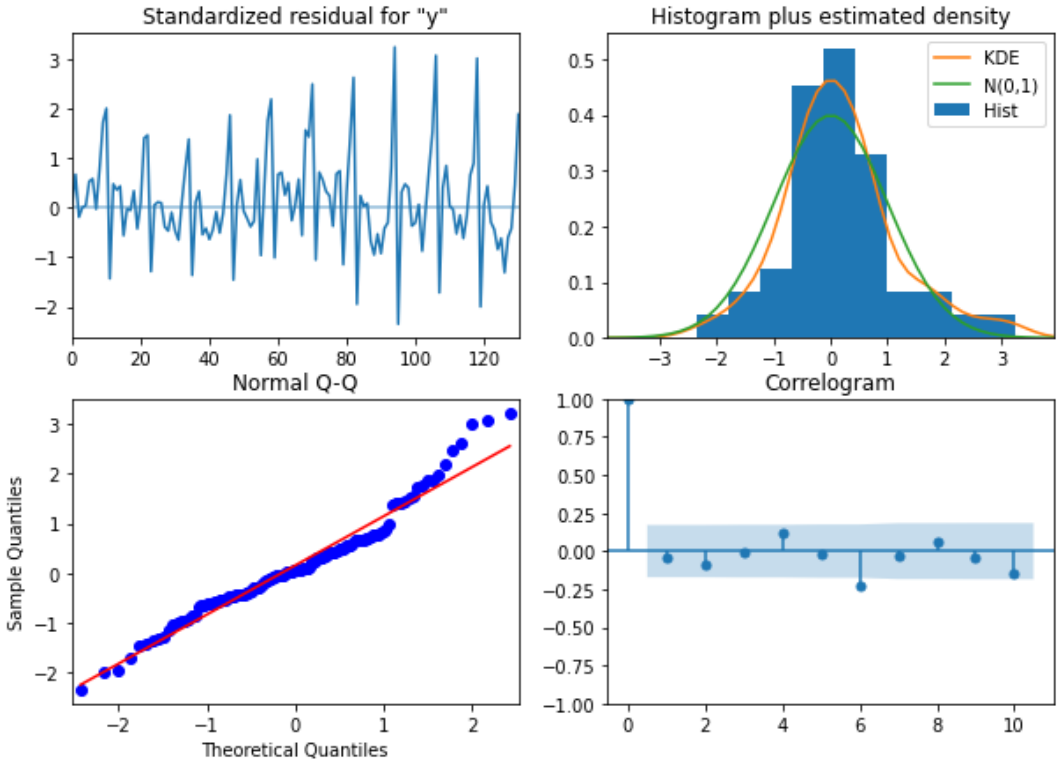


Figure 28. Sparkling Diagnostics plot for AUTO ARIMA

- The residuals of histogram states that they are not uniformly distributed.
- The Q-Q plot indicates that quantiles are not coming from the normal distribution as they are not aligned with the line.
- The Correlogram shows the autocorrelation of the residuals and there is one component is significant and it exceeds the confidence intervals.

Model Evaluation on Test Data

RMSE values for the ARIMA(2,1,2) shown below.

	Test RMSE
AUTO ARIMA(2,1,2)	1299.979779

Table 18. Test Data RMSE for AUTO ARIMA(2,1,2)

Building auto SARIMA model using AIC values

Let's include the seasonality component in the ARIMA model and see it if it brings out the best RMSE values.

- pq/PQ values from 0 to 3
- d=1 because it becomes non-stationary to stationary after differencing it with 1 time itself.
- D=0 because the seasonality component is stationary after running with ADF test.
- Here the data provided is monthly, thus M=12.

Through 'itertools' python package, let's create the series of loop for SARIMA model parameters.

Let's import the SARIMAX function from statsmodels time series library and run through the different combinations of pq values with d=1 and PD values with D=0

	param	seasonal	AIC
236	(3, 1, 2)	(3, 0, 0, 12)	1387.234718
220	(3, 1, 1)	(3, 0, 0, 12)	1387.788332
237	(3, 1, 2)	(3, 0, 1, 12)	1388.602196
221	(3, 1, 1)	(3, 0, 1, 12)	1388.681484
252	(3, 1, 3)	(3, 0, 0, 12)	1389.142084

Table 19. AIC Table for AUTO SARIMA(3,1,2)(3,0,0,12)

Fit this model and see the summary below.

SARIMAX Results

Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(3, 1, 2)x(3, 0, [], 12)	Log Likelihood	-684.617
Date:	Thu, 10 Feb 2022	AIC	1387.235
Time:	10:49:00	BIC	1409.931
Sample:	0	HQIC	1396.395
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.5374	0.338	-1.588	0.112	-1.201	0.126
ar.L2	0.0256	0.187	0.137	0.891	-0.340	0.392
ar.L3	0.0785	0.130	0.604	0.546	-0.176	0.333
ma.L1	-0.3365	0.294	-1.143	0.253	-0.913	0.240
ma.L2	-0.7979	0.344	-2.322	0.020	-1.471	-0.124
ar.S.L12	0.5713	0.103	5.541	0.000	0.369	0.773
ar.S.L24	0.2606	0.117	2.223	0.026	0.031	0.490
ar.S.L36	0.2126	0.111	1.916	0.055	-0.005	0.430
sigma2	1.449e+05	2.95e+04	4.911	0.000	8.71e+04	2.03e+05

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	8.80
Prob(Q):	0.99	Prob(JB):	0.01
Heteroskedasticity (H):	1.17	Skew:	0.36
Prob(H) (two-sided):	0.67	Kurtosis:	4.33

Figure 29.Statistical summary for Sparkling AUTO SARIMA

From the statistical summary, seasonality AR components are significant and other components are not significant since the p-value >0.05.

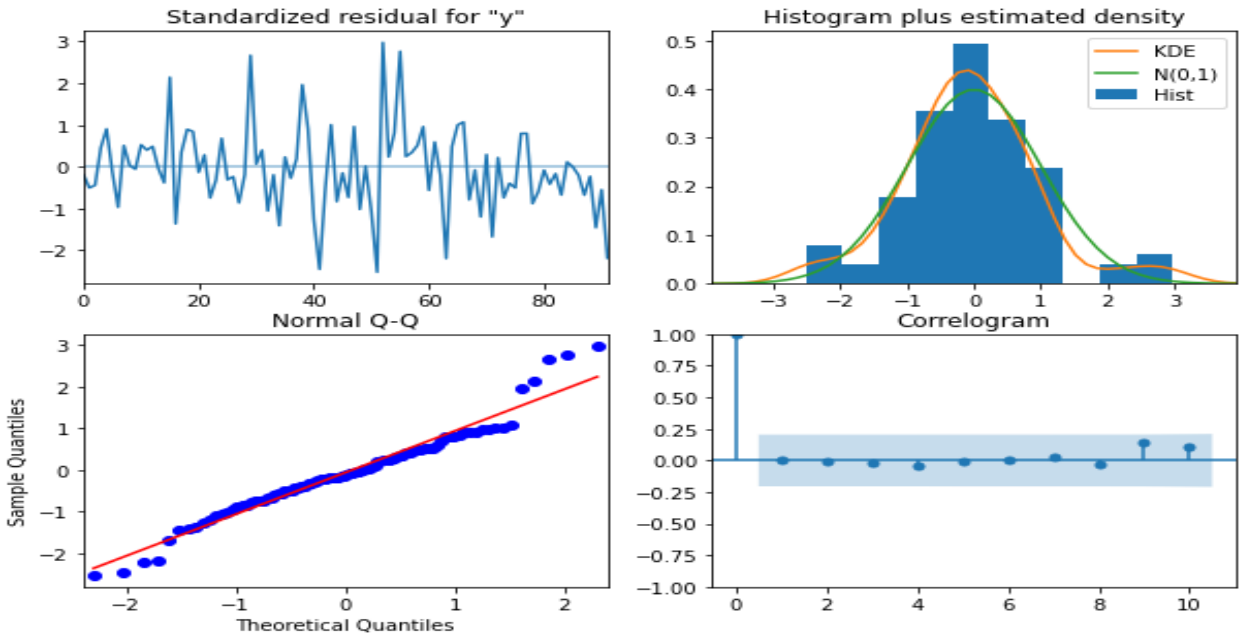


Figure 30. Sparkling Diagnostics plot for AUTO SARIMA

- The residuals of histogram states that they are slightly uniformly distributed.
- The Q-Q plot indicates that quantiles are coming from the normal distribution as they are almost aligned with the line. Only few points are moved away from the line.
- The Correlogram shows the autocorrelation of the residuals and none of the terms are exceeding the confidence intervals.

Model Evaluation on Test Data

RMSE values for the AUTO SARIMA(3, 1, 2)(3, 0, 0, 12) shown below.

	Test RMSE
AUTO SARIMA(3, 1, 2)(3, 0, 0, 12)	543.170722

Table 20. Test Data RMSE for AUTO SARIMA(3,1,2)(3,0,0,12)

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.

Building ARIMA model using ACF and PACF plots

Train data becomes stationary after taking the 1st order difference. So, plot the same using the train.diff() values. Hence, **d=1**.

p=0(AR)

To estimate the amount of AR terms, you need to look at the PACF plot. First, ignore the value at lag 0. It will always show a perfect correlation, since we are estimating the correlation between today's values with itself. Note that there is a blue area in the plot, representing the confidence interval. To estimate how much AR terms you should use, start counting how many "lollipop" are above or below the confidence interval before the next one enter the blue area.

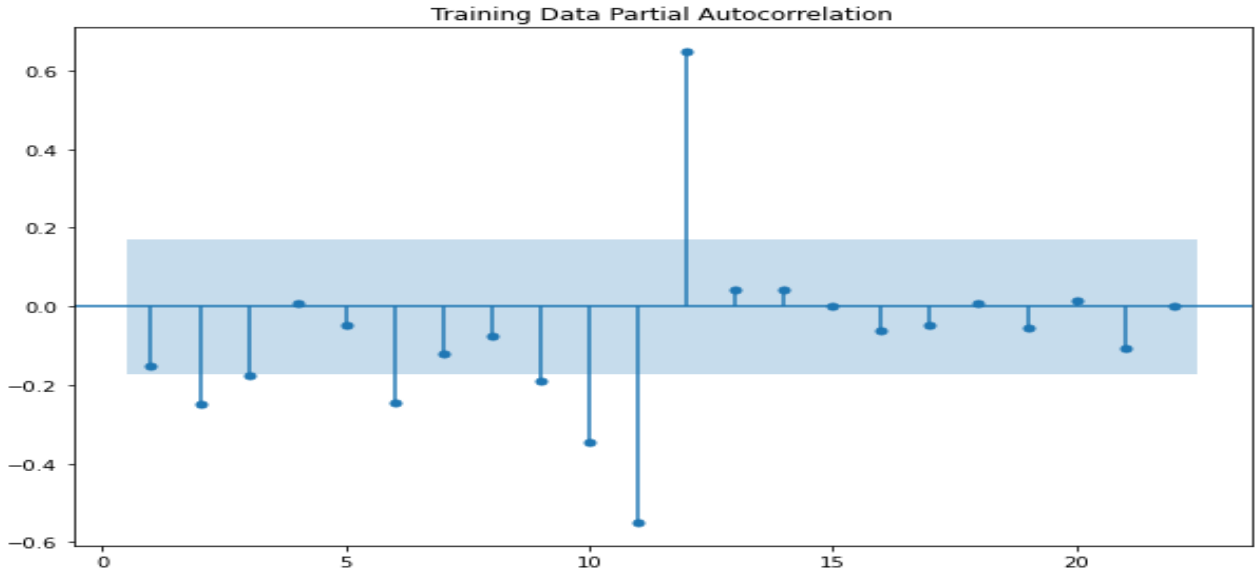


Figure 31. Sparkling Partial Autocorrelation plot

So, looking at the PACF plot above, we can estimate to use 0 AR terms for our model, since there is no lag and the 1st component itself within the blue area.

q=0(MA)

To estimate the amount of MA terms, this time you will look at ACF plot. The same logic is applied here: how much lollipops are above or below the confidence interval before the next lollipop enters the blue area?

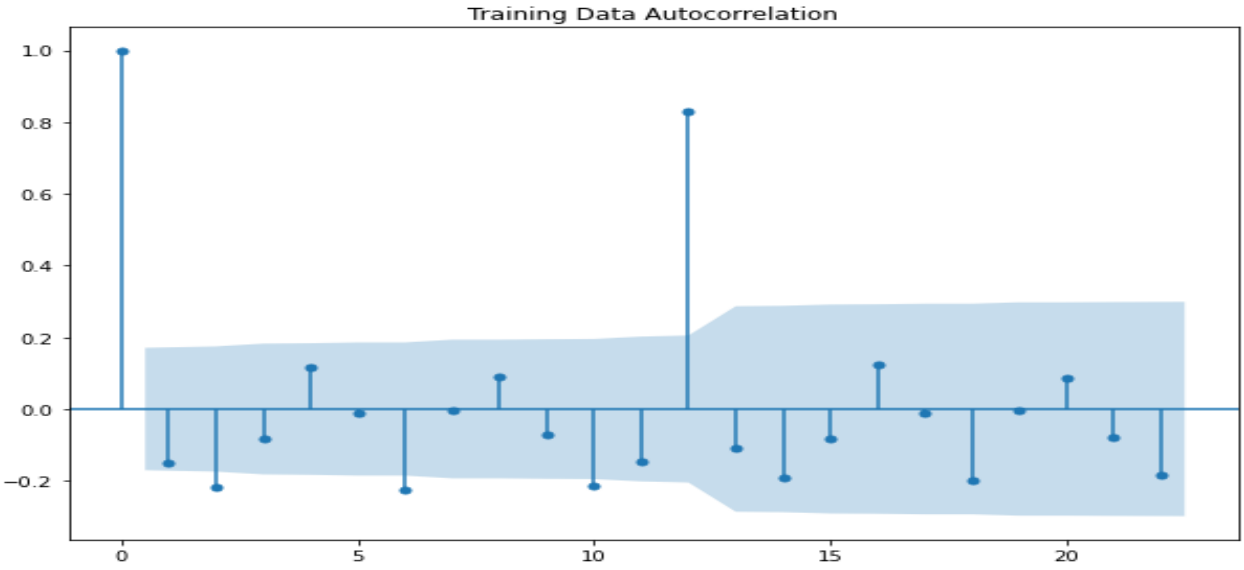


Figure 32. Sparkling Autocorrelation plot

In our example, we can estimate 0 MA terms, since there is no lag and the 1st component itself within the blue area.

Let's build and fit the ARIMA model using the parameters identified

```
manual_ARIMA = ARIMA(train['Sparkling'].values, order=(0,1,0))
```

SARIMAX Results

Dep. Variable:	y	No. Observations:	132
Model:	ARIMA(0, 1, 0)	Log Likelihood	-1132.832
Date:	Thu, 10 Feb 2022	AIC	2267.663
Time:	11:06:50	BIC	2270.538
Sample:	0	HQIC	2268.831
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
sigma2	1.885e+06	1.29e+05	14.658	0.000	1.63e+06	2.14e+06

Ljung-Box (L1) (Q):	3.07	Jarque-Bera (JB):	198.83
Prob(Q):	0.08	Prob(JB):	0.00
Heteroskedasticity (H):	2.46	Skew:	-1.92
Prob(H) (two-sided):	0.00	Kurtosis:	7.65

Figure 33.Statistical summary for Sparkling manual ARIMA

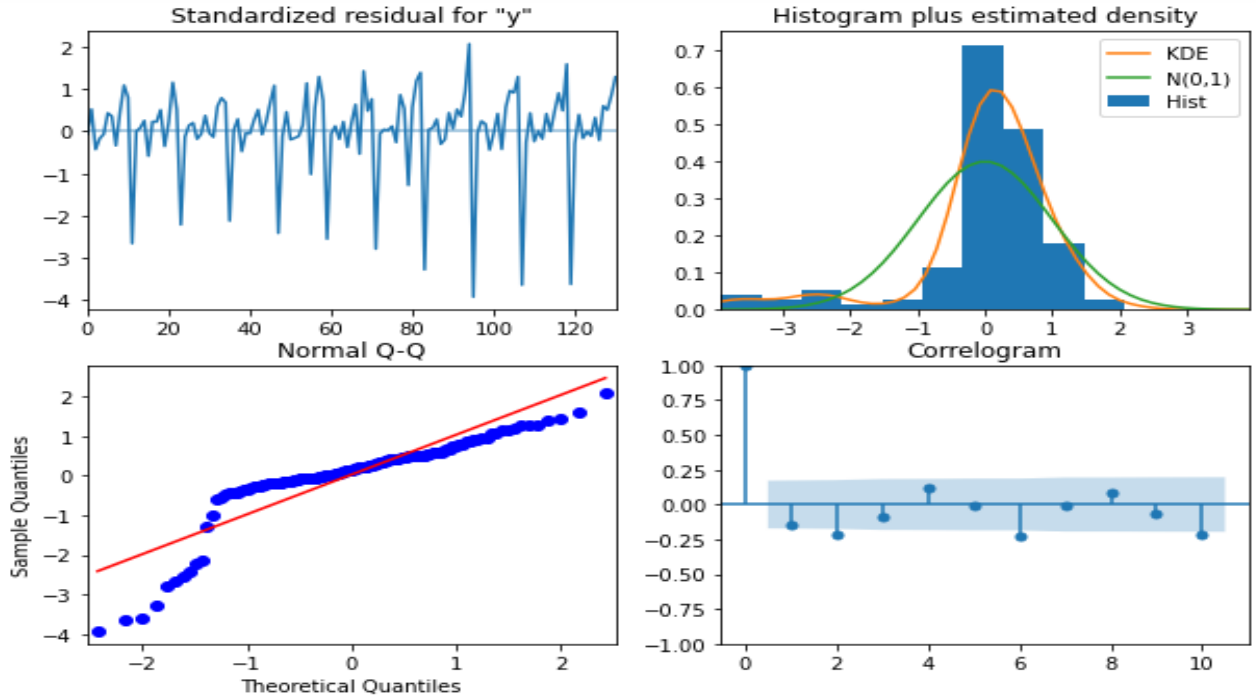


Figure 34. Sparkling Diagnostics plot for Manual ARIMA

The Correlogram shows the autocorrelation of the residuals and few terms are outside the confidence intervals. i.e., it forms the pattern. Not the good fit for our business problem.

Model Evaluation on Test Data

RMSE values for the ARIMA(0, 1, 0) shown below and so far the worst RMSE values.

	Test RMSE
Manual ARIMA(0,1,0)	3864.279352

Table 21. Test Data RMSE for Manual ARIMA(0,1,0)

Building SARIMA model using ACF and PACF plots

The process is quite similar to non-seasonal AR, and you will still be using the ACF and PACF function for that. To estimate the amount of AR terms, you will look one more time to the PACF function. Now, instead of count how many lollipops are out of the confidence interval, you will count how many **seasonal lollipops** are out.

The Q order can be calculated from the Autocorrelation (ACF) plot. Autocorrelation is the correlation of a single time series with a lagged copy of itself.

From the above graph, we note that the maximum lag with a value out the confidence intervals is 8, thus Q = 1(MA)

In the PACF graph the maximum lag with a value out the confidence intervals (in light blue) is 1, thus we can set P = 1(AR)

D=0 because the seasonality component is stationary after running with ADF test.

M=12. M indicates the periodicity, i.e. the number of periods in season, such as 12 for monthly data.

Let's build and fit the SARIMA model using the parameters identified

```
manual_SARIMA = sm.tsa.statespace.SARIMAX(train['Sparkling'],
order=(0,1,0), seasonal_order=(1, 0, 1, 12),
enforce_stationarity=False, enforce_invertibility=False)
```

SARIMAX Results

Dep. Variable:	Sparkling	No. Observations:	132
Model:	SARIMAX(0, 1, 0)x(1, 0, [1], 12)	Log Likelihood	-900.495
Date:	Thu, 10 Feb 2022	AIC	1806.991
Time:	11:31:44	BIC	1815.303
Sample:	01-01-1980 - 12-01-1990	HQIC	1810.365
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.S.L12	1.0325	0.019	52.957	0.000	0.994	1.071
ma.S.L12	-0.5384	0.078	-6.896	0.000	-0.691	-0.385
sigma2	2.463e+05	2.34e+04	10.520	0.000	2e+05	2.92e+05

Ljung-Box (L1) (Q):	19.69	Jarque-Bera (JB):	31.97
Prob(Q):	0.00	Prob(JB):	0.00
Heteroskedasticity (H):	1.88	Skew:	0.66
Prob(H) (two-sided):	0.05	Kurtosis:	5.18

Figure 35.Statistical summary for Sparkling Manual SARIMA

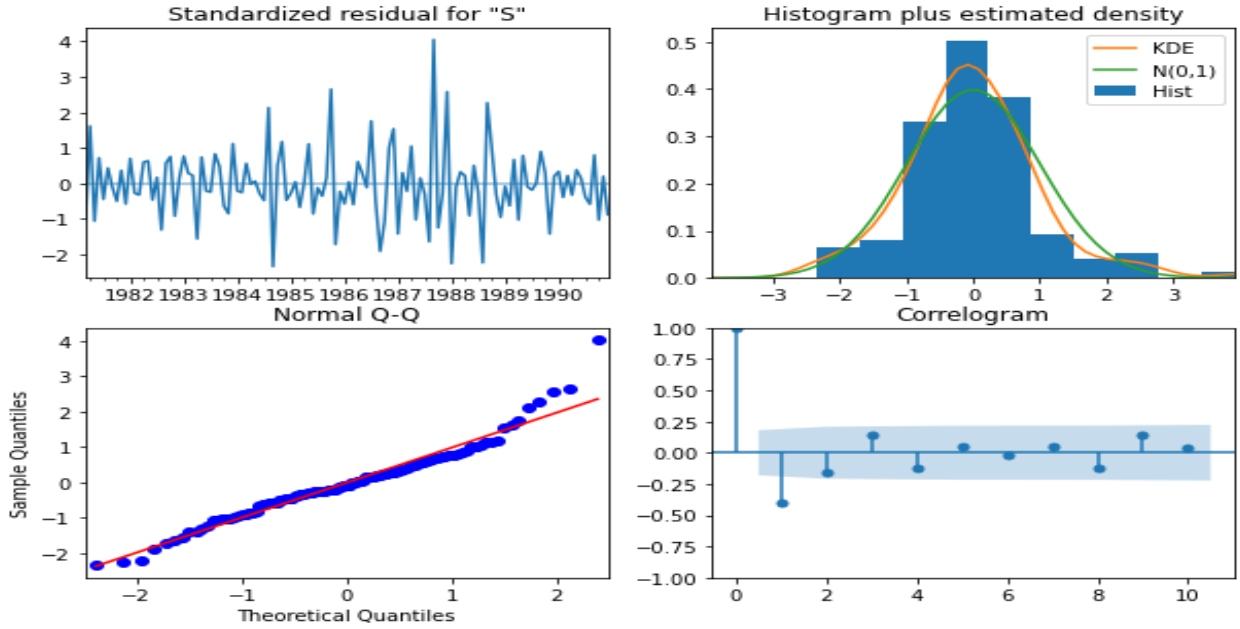


Figure 36. Sparkling Diagnostics plot for Manual SARIMA

The Correlogram shows the autocorrelation of the residuals and there is one component is significant and it exceeds the confidence intervals. So, the residuals are forming some kind of pattern which is not a good fit.

Model Evaluation on Test Data

RMSE values for the SARIMA (0, 1, 0)(1,0,1,12) shown below.

	Test RMSE
Manual SARIMA(0,1,0)(1, 0, 1, 12)	1787.706703

Table 22. Test Data RMSE for Manual SARIMA(0,1,0)(1,0,1,12)

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

Summary of all the Models

	Test RMSE
RegressionOnTime	1389.135175
NaiveModel	3864.279352
SimpleAverageModel	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
Alpha=0.070,SimpleExponentialSmoothing	1338.007771
Alpha=0.01,SimpleExponentialSmoothing	1286.648058
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	1778.564670
Alpha=0.111,Beta=0.061,Gamma=0.395,TripleExponentialSmoothing	469.591516
Alpha=0.4,Beta=0.1,Gamma=0.2,TripleExponentialSmoothing	336.715250
AUTO ARIMA(2,1,2)	1299.979779
AUTO SARIMA(3, 1, 2)(3, 0, 0, 12)	543.170722
Manual ARIMA(0,1,0)	3864.279352
Manual SARIMA(0,1,0)(1, 0, 1, 12)	1787.706703

Table 23. Test Data RMSE summary for all models

By looking at the RMSE values for the test records, we can say Triple Exponential Smoothing, AUTO SARIMA and 2 point Trailing MA models yields the lower RMSE values.

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.

	Test RMSE
Alpha=0.4,Beta=0.1,Gamma=0.2,TripleExponentialSmoothing	336.715250
Alpha=0.111,Beta=0.061,Gamma=0.395,TripleExponentialSmoothing	469.591516
AUTO SARIMA(3, 1, 2)(3, 0, 0, 12)	543.170722
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
SimpleAverageModel	1275.081804
Alpha=0.01,SimpleExponentialSmoothing	1286.648058
AUTO ARIMA(2,1,2)	1299.979779
Alpha=0.070,SimpleExponentialSmoothing	1338.007771
RegressionOnTime	1389.135175
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	1778.564670
Manual SARIMA(0,1,0)(1, 0, 1, 12)	1787.706703
NaiveModel	3864.279352
Manual ARIMA(0,1,0)	3864.279352

Table 24.Ordered Test Data RMSE summary for all models

We see that the best model is the Triple Exponential Smoothing with multiplicative seasonality with the parameters α = 0.4, β = 0.1 and γ = 0.2. Let's fit this model with the original data

```
fullmodel1 =
ExponentialSmoothing(df,trend='additive',seasonal='multiplicative').fi
t(smoothing_level=0.4,smoothing_trend=0.1,smoothing_seasonal=0.2)
```

RMSE for the full model is **376.7746**

Forecast the future for next 5 years (approximately)

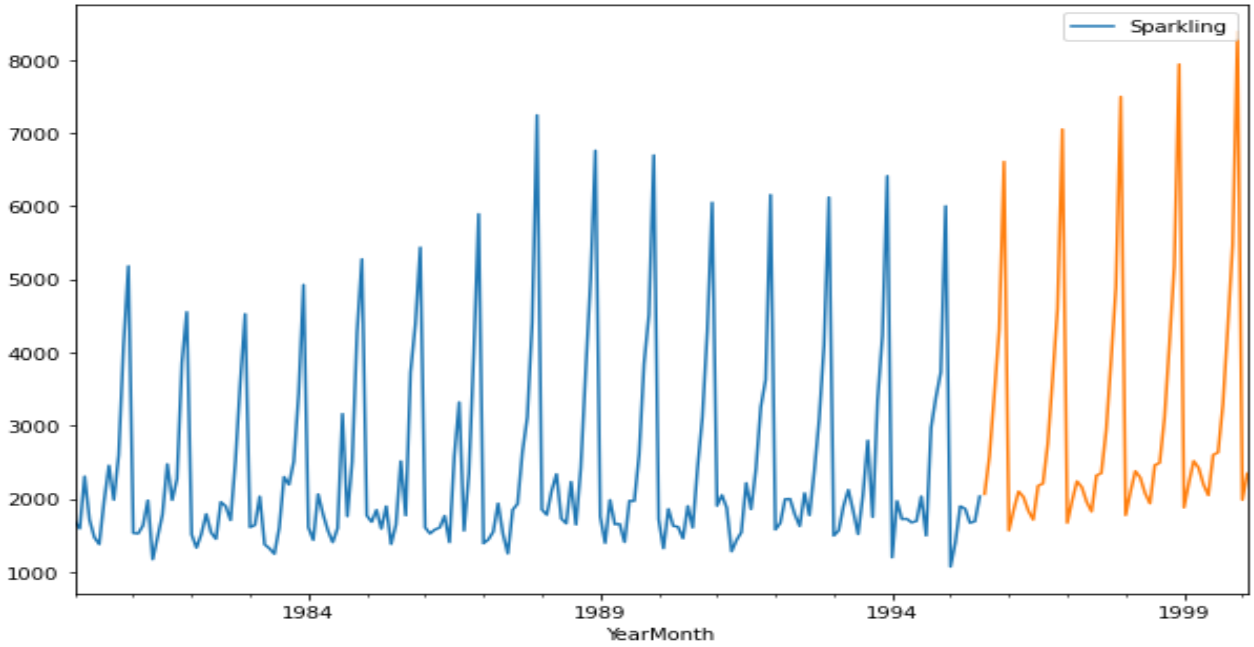


Figure 37. Sparkling Forecast data for next 5 years

Forecasting 12 months into the future

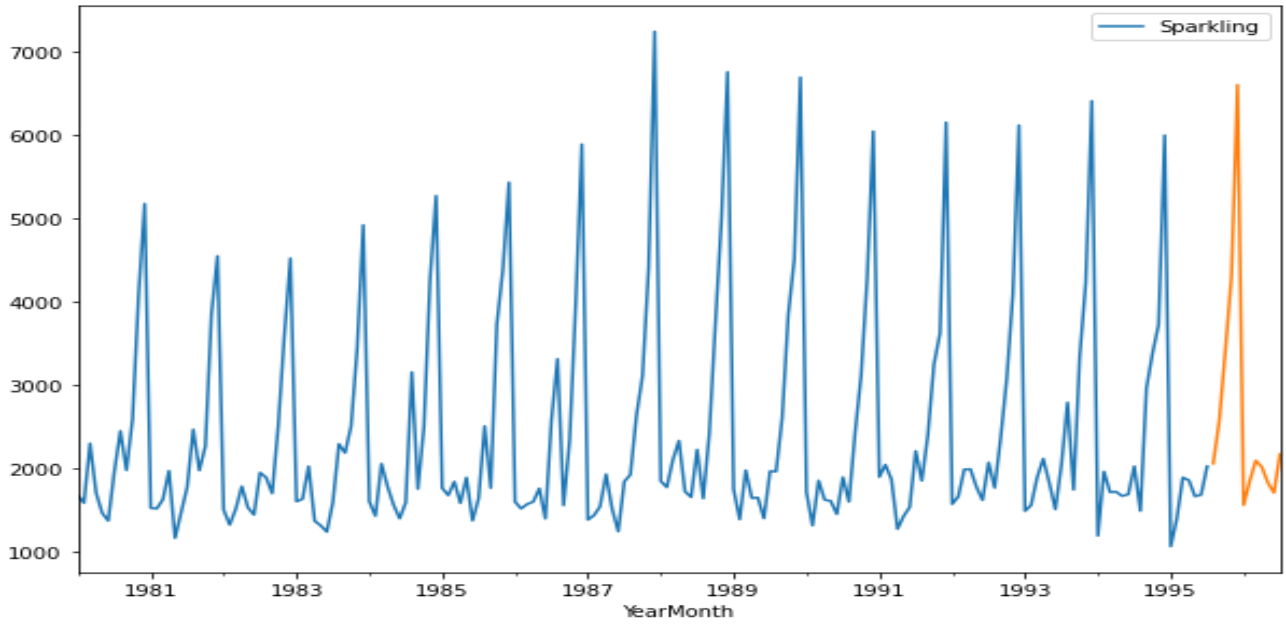


Figure 38. Sparkling Forecast data for next 12 months

Forecasting 12 months into the future with Confidence Intervals

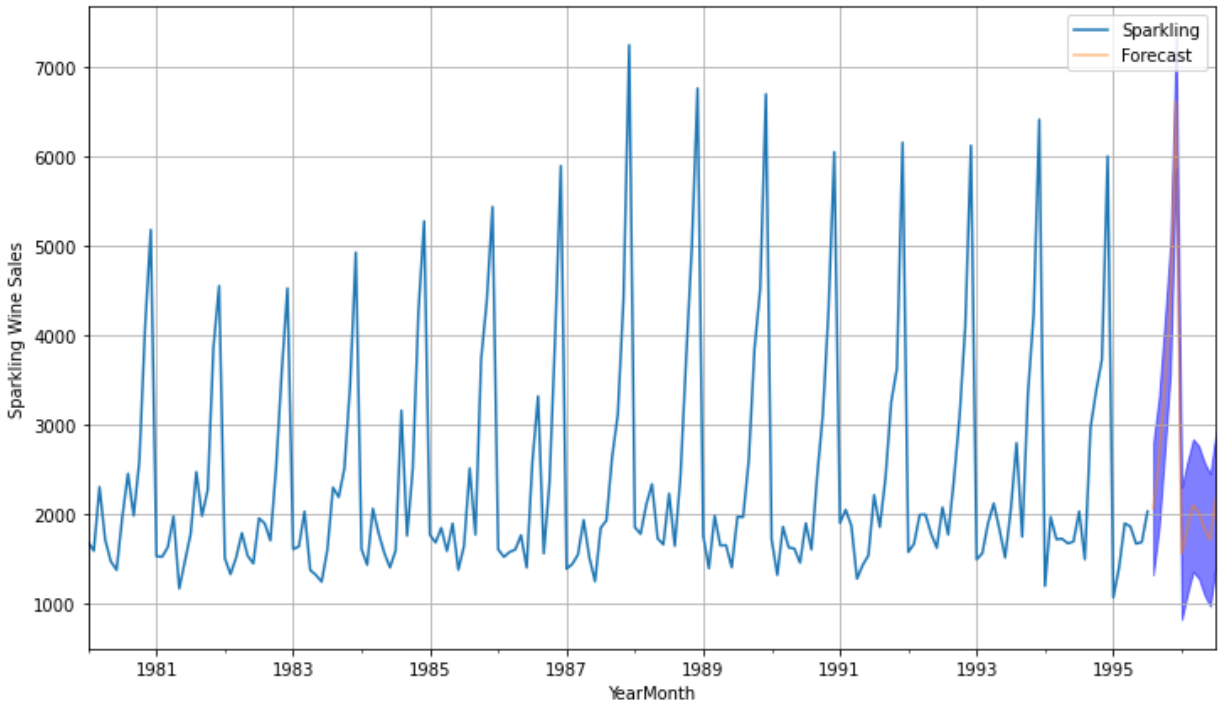


Figure 39. Sparkling Forecast data for next 12 months with Confidence Intervals

Forecasted 12 months values

1995-08-01	2063.448928
1995-09-01	2579.407431
1995-10-01	3416.654241
1995-11-01	4304.476935
1995-12-01	6604.875995
1996-01-01	1564.539687
1996-02-01	1849.759839
1996-03-01	2098.878757
1996-04-01	2022.428764
1996-05-01	1834.540611
1996-06-01	1712.408856
1996-07-01	2176.425389

Table 26.Forecasted values(12 months)

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

Recommendations and Insights

1. Among all the models builds, Triple Exponential Smoothing model outperforms all the other models. TES model is used to handle the time series data containing trend and seasonality. This model is smoothening out the level, trend and seasonal components. Our sparkling data has both trend and seasonality and it is the good fit for this data.
2. We observe the very sharp increase in units of sales during October, November and December. There is an abrupt decrease in sales starting from January and it is almost stationary till June. Again, the sales are picking up from July onwards. Especially on December, sales picked up quite busy. So, advise the production unit to run with minimum production Q1 & Q2 and increase the production a bit high on Q3 and very high on Q4. This way, we may meet our customers' demands and increase the sales even high on holiday seasons.
3. On 1988, sum of the sales is quite high compared to the other years. Try to get the insights and see what has changed from 1988 to 1999 in the wine, sales, marketing etc. This may help the estate to consider the similar options in forthcoming months.
4. Conduct the Wine direct-to-survey to ask about the improvements can be made to the existing flavors and understand their preferences.

Rose Wine Sales

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

Read the Rose Time Series Data

There are 187 records and two columns YearMonth and Sales. While reading the Time Series data, let's read datas using **'parse_dates=True'** parameter and set the index column as YearMonth. By this, we will use the lot of Time Series functionalities offered by the pandas directly.

YearMonth	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 27.Rose Wine Sales Data

Now, we have our data ready for the Time Series Analysis.

Plot the data

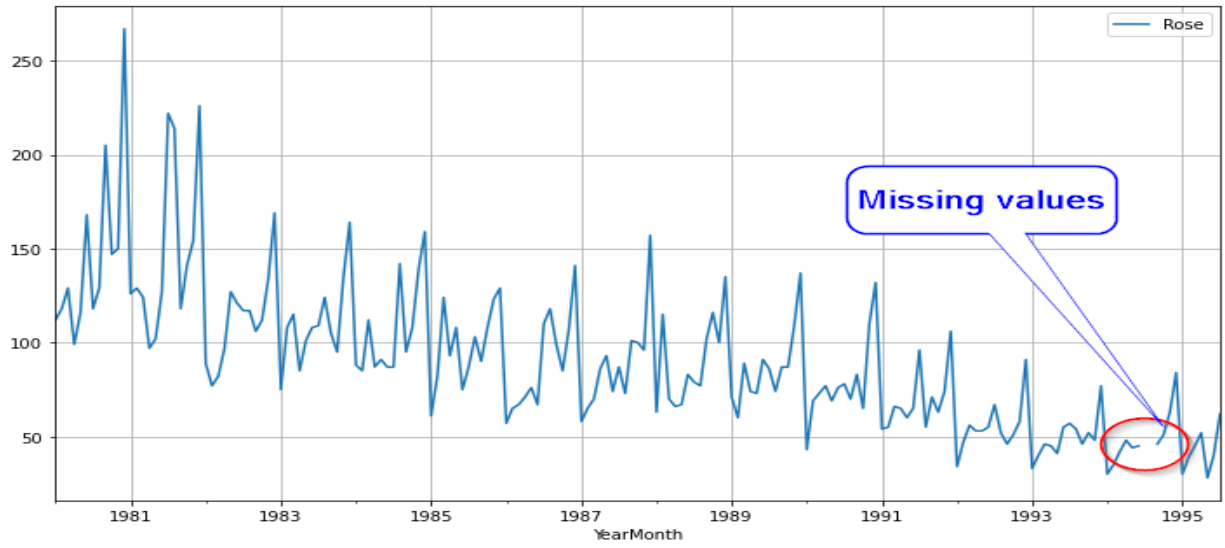


Figure 40. Rose Time Series plot

As it is Time Series forecasting analysis, data should be continuous without any null values or break in the data. Also, it has to be sequential. Upon checking, there are 2 null values present in the given dataset as shown below.

YearMonth	Rose
1994-07-01	NaN
1994-08-01	NaN

Table 28.Rose missing Sales Data

We will impute using the interpolate spline method for these 2 records.

YearMonth	Rose
1994-01-01	30.000000
1994-02-01	35.000000
1994-03-01	42.000000
1994-04-01	48.000000
1994-05-01	44.000000
1994-06-01	45.000000
1994-07-01	46.153199
1994-08-01	47.211982
1994-09-01	46.000000
1994-10-01	51.000000
1994-11-01	63.000000
1994-12-01	84.000000

Table 29.Rose Data after imputation

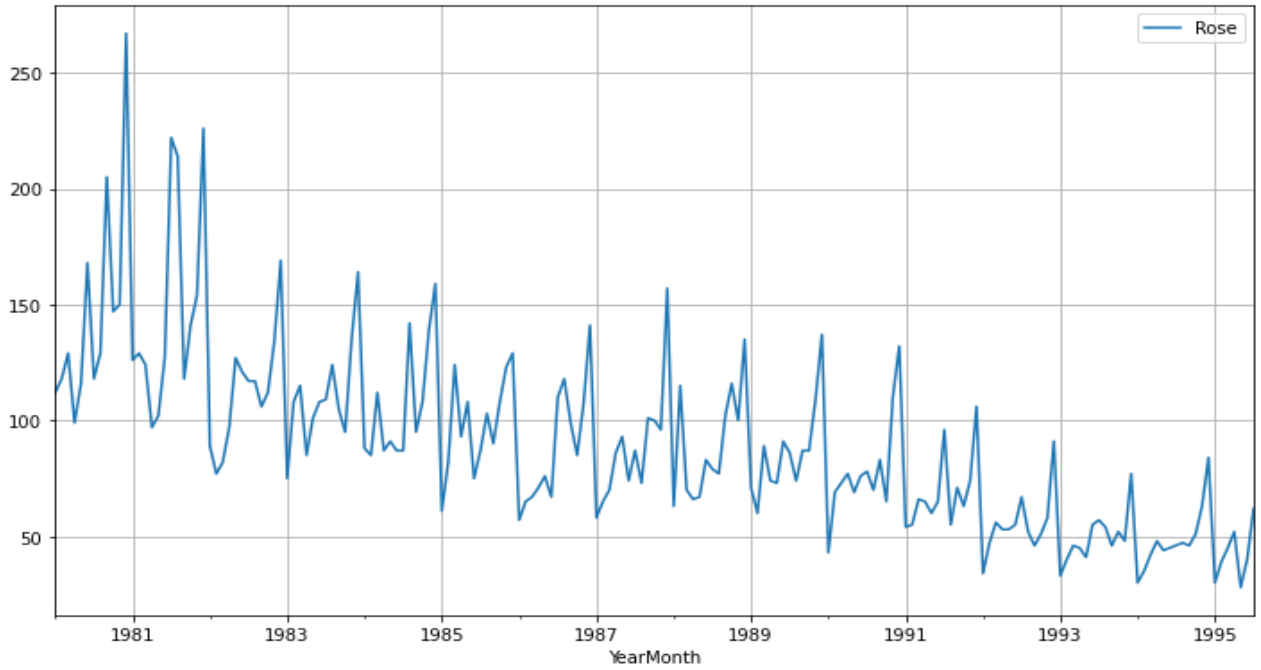


Figure 41. Rose Time Series plot after imputing missing values

Now, we see that there is a continuous in the time series plot.

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

Exploratory Data Analysis

	Rose
count	187.000000
mean	89.927087
std	39.224153
min	28.000000
25%	62.500000
50%	85.000000
75%	111.000000
max	267.000000

Table 30.Rose Data after imputation

- The average unit sale of Rose wine is ~90.
- The minimum sale and maximum sale is 28 and 267 units respectively.

Yearly Boxplot

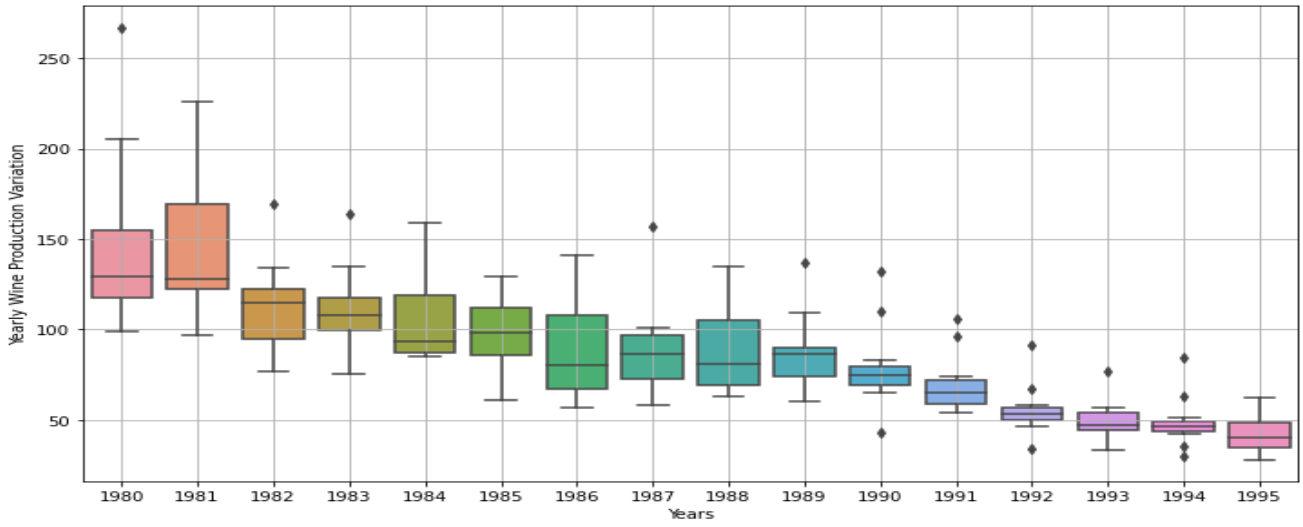


Figure 42. Rose Yearly Boxplot

From the above plot, we observed that there is a high unit of sales on 1980 and 1981. Post then, the rose wine sales starts decreasing gradually.

Monthly Boxplot

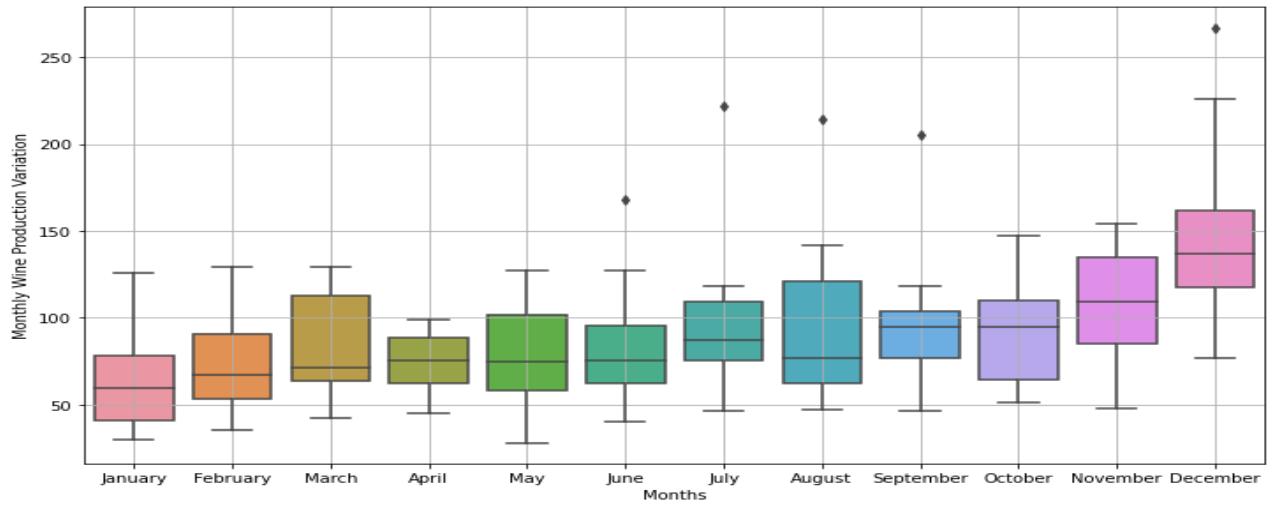


Figure 43. Rose Monthly Boxplot

The Rose wine sale is at its peak during the month of November and December of the year. Otherwise, there is an approximate average sale of 100 units from March to October.

Plot a time series month plot to understand the spread of wine sales across different years and within different months across years.

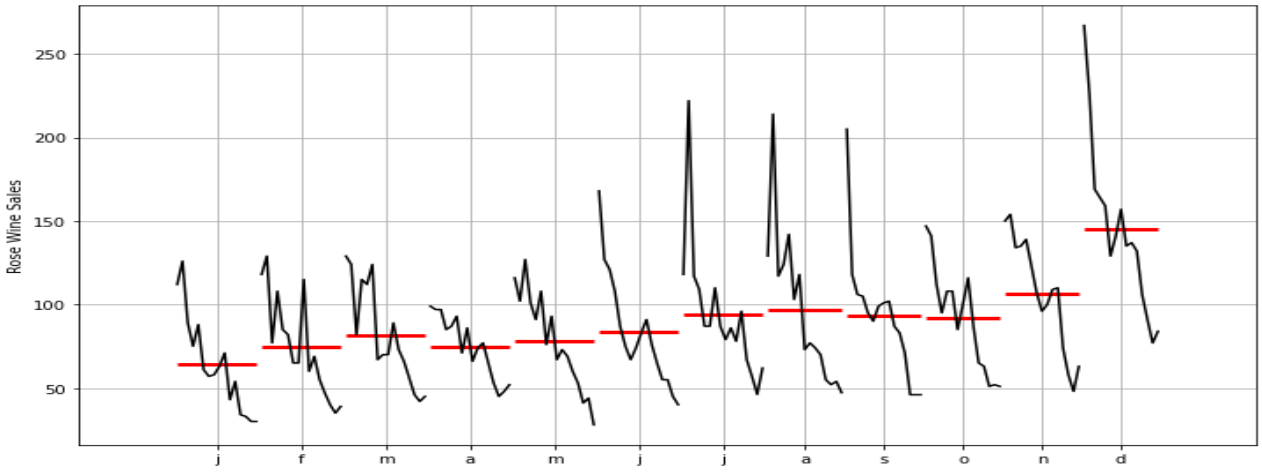


Figure 44. Rose Monthly Time Series plot

From the time series plot, we observed that Rose sales of above 200 units on the month of July, August and September. On December, the sales are gone above 250 units.

Observe, the trend keeps coming down over the years in all the months.

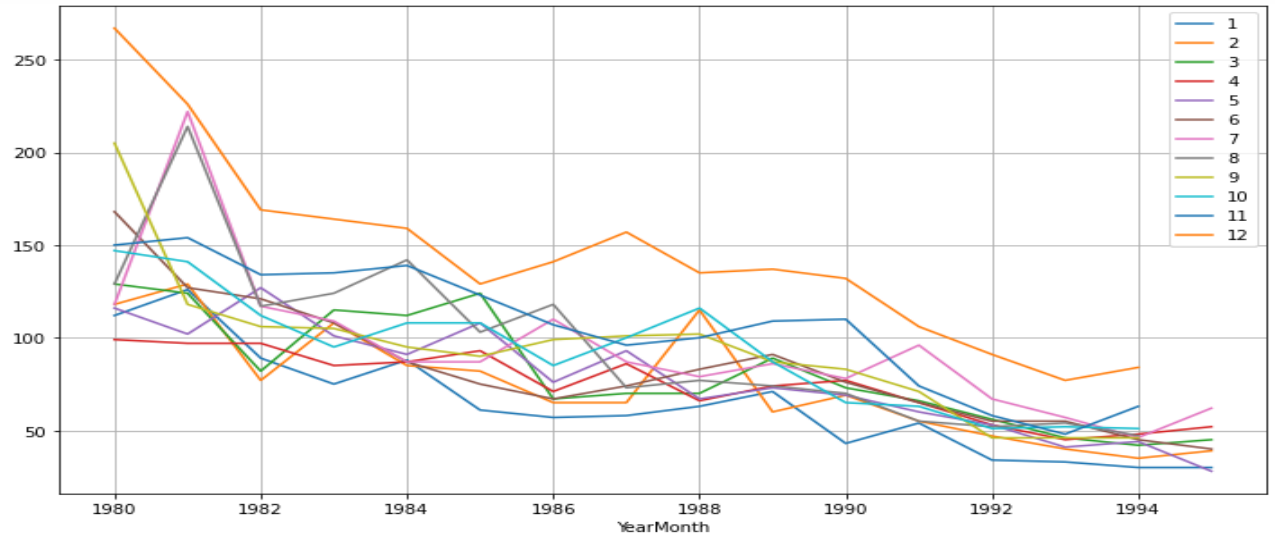


Figure 45. Rose Monthly Sales across Years plot

Even across the year, the sale of rose wine is exponentially going down is clearly visible in this graph.

Sum of Sales at the end of every Year

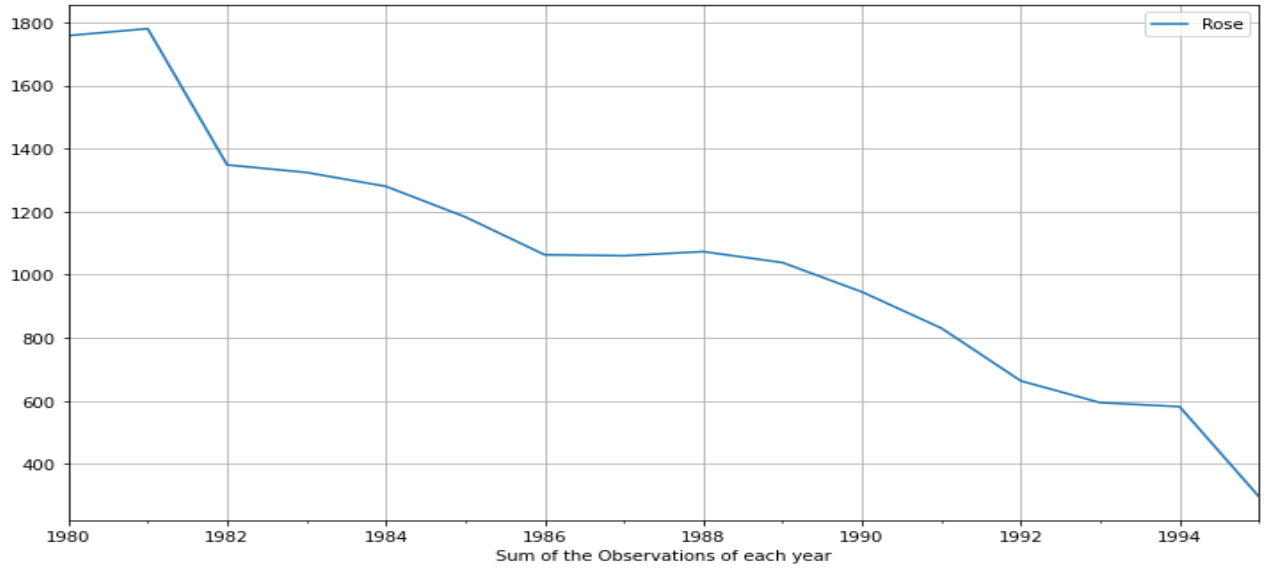


Figure 46. Rose Year End Sum of Sales

The sum of sales is high on the year 1980 and on the year 1991 onwards it is abruptly coming down. It's almost from 1800 to 300 units which is really a bad sign for this rose wine would go out of the market soon.

Average of Sales at the end of every Year

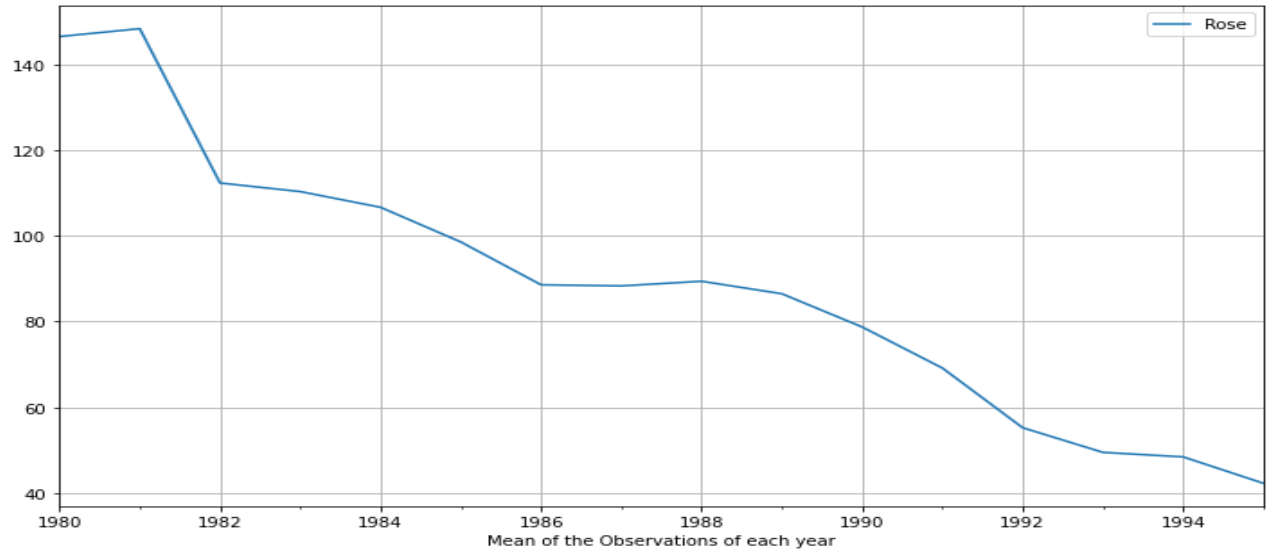


Figure 47. Rose Year End Average Sales

The average of the rose wine sales is 150 units on the year 1980 and reached out to 40 units on 1994.

Between these years, the sales kept coming down.

Average Quarterly Sales

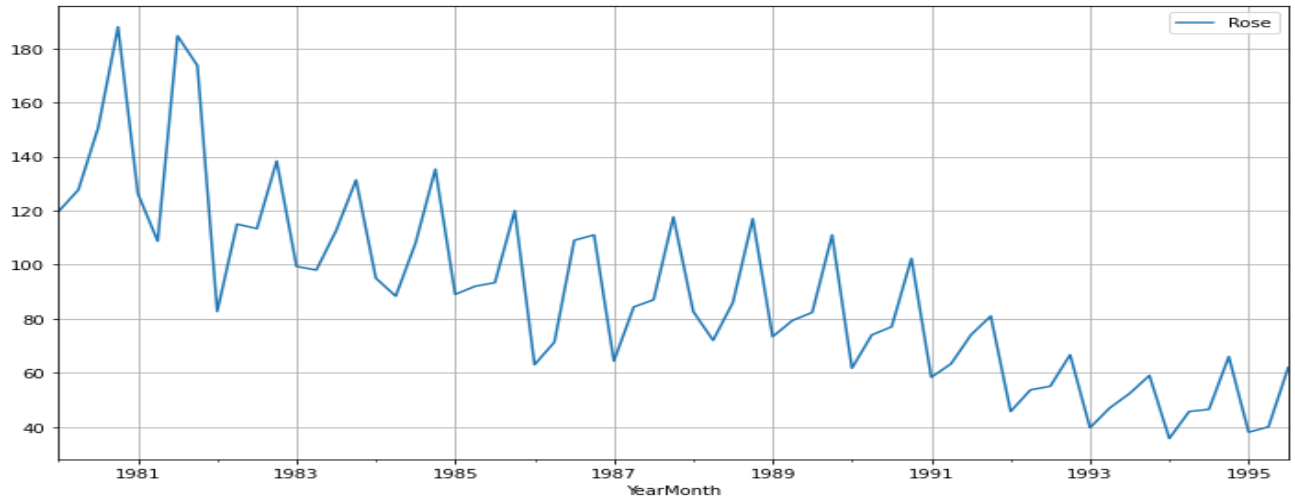


Figure 48. Rose Quarterly Average Sales

Over the years, from the end of July to October, there is a slight consistency and again there is a sharp spike on end of Q4. But, the sales kept coming down between the years 1981 and 1995.

Average Sales and Percentage Change in Sales

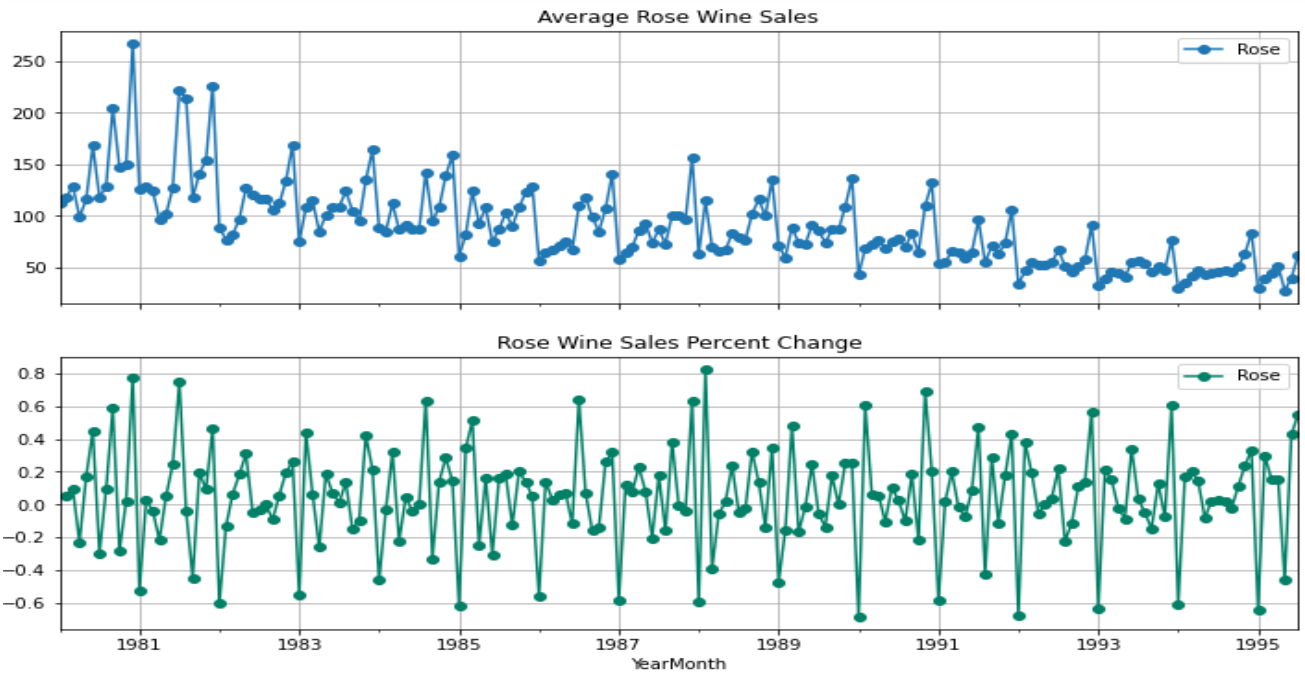


Figure 49. Rose Average and Percentage Changes Sales

At very few times, the percentage of sales varies from 60 to 80 percent. Otherwise, it is just under the 20 percent.

Multiplicative Model Decomposition

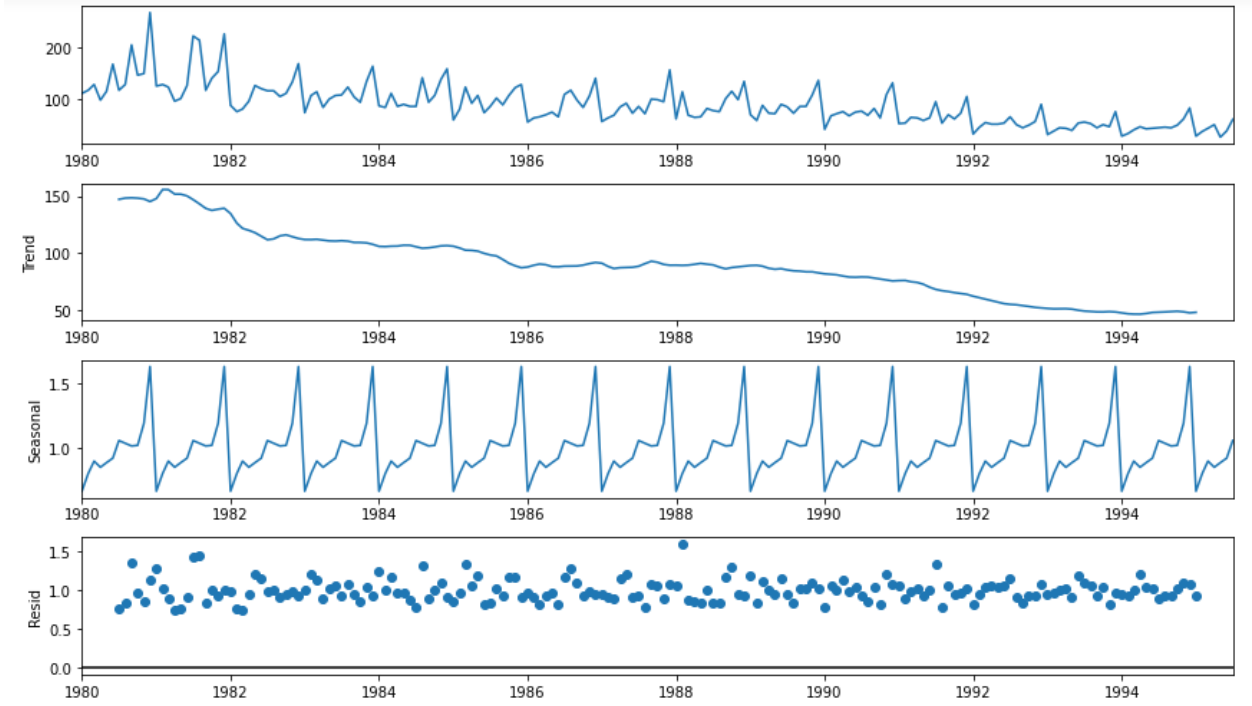


Figure 50. Rose Time Series Decomposition

There is a decreasing trend after 1981 and the sales kept coming down exponentially till 1994 July. Seasonality is present in the Rose dataset. Both the trend and seasonality gets decreases together over the years. So, this is possibly a 'multiplicative' method.

Time Series without Seasonality plots

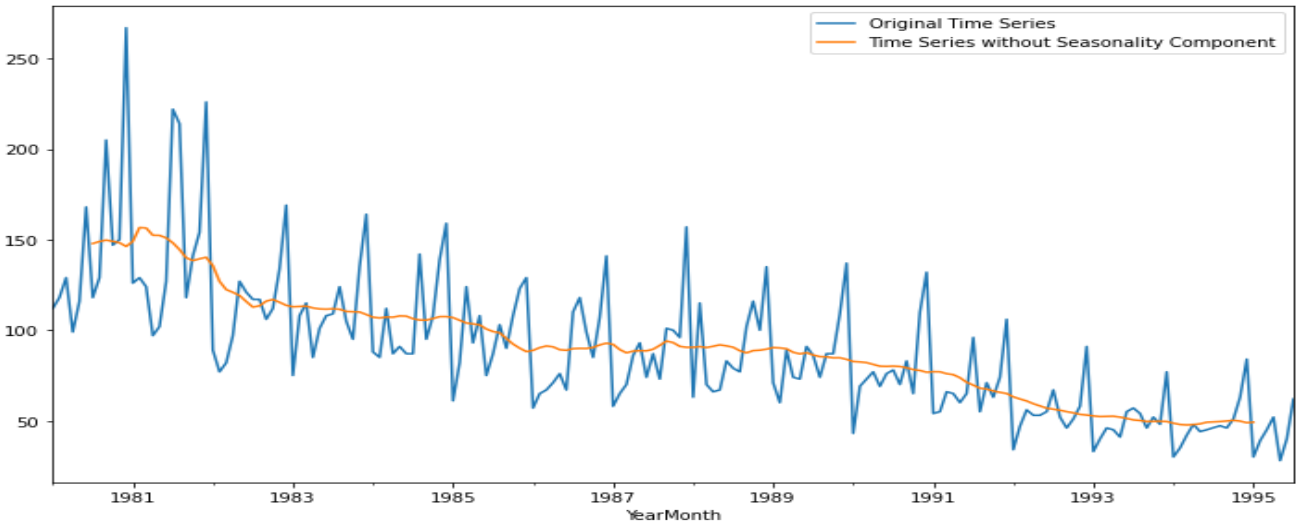


Figure 51. Rose Time Series without Seasonality

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

Splitting the Data

Let's split the training data till 1990 December and test data start from beginning January 1991.

YearMonth	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

YearMonth	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

Table 31. Rose Training and Test Data

In training, there are 132 records and in testing we got 55 records.

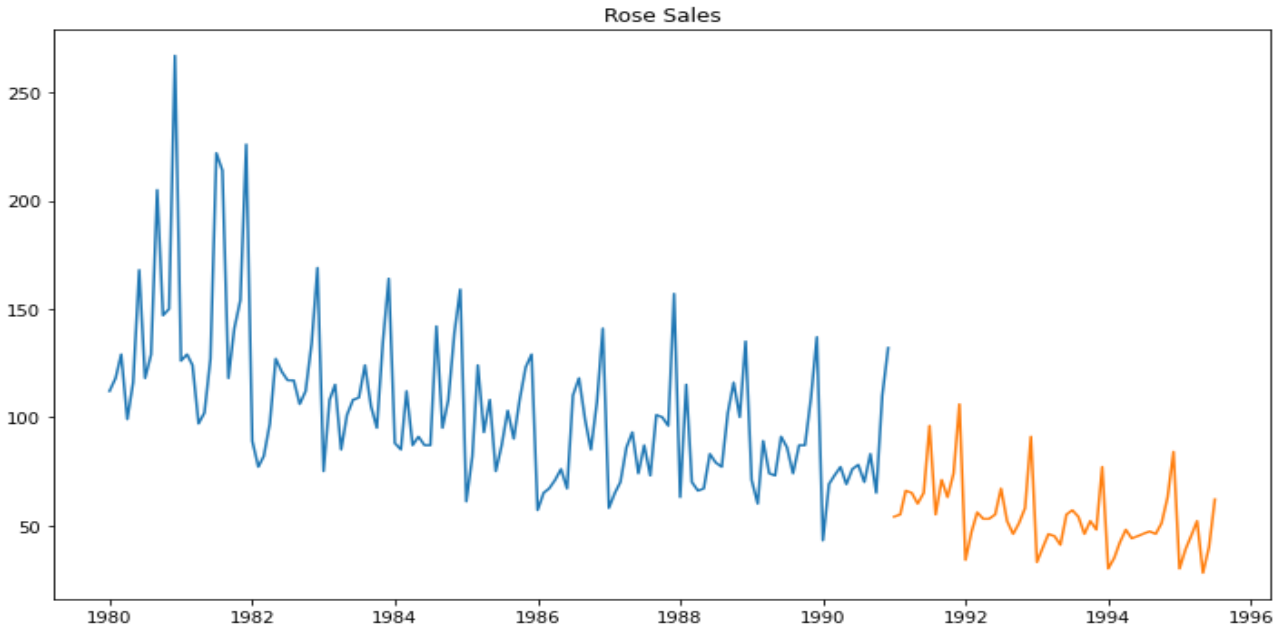


Figure 52.Rose Training and Testing Plot

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.

Building Models

Linear Regression

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

```
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, 3
4, 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, 18
3, 184, 185, 186, 187]
```

First few rows of Training Data

YearMonth	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

Last few rows of Training Data

YearMonth	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

First few rows of Test Data

YearMonth	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 Test Data

YearMonth	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

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

Let's instantiate the Linear Regression Model using the below function and fit the model using training time and Rose Sales data.

```
lr = LinearRegression()

lr.fit(LinearRegression_train[['time']],LinearRegression_train['Rose'].values)
```

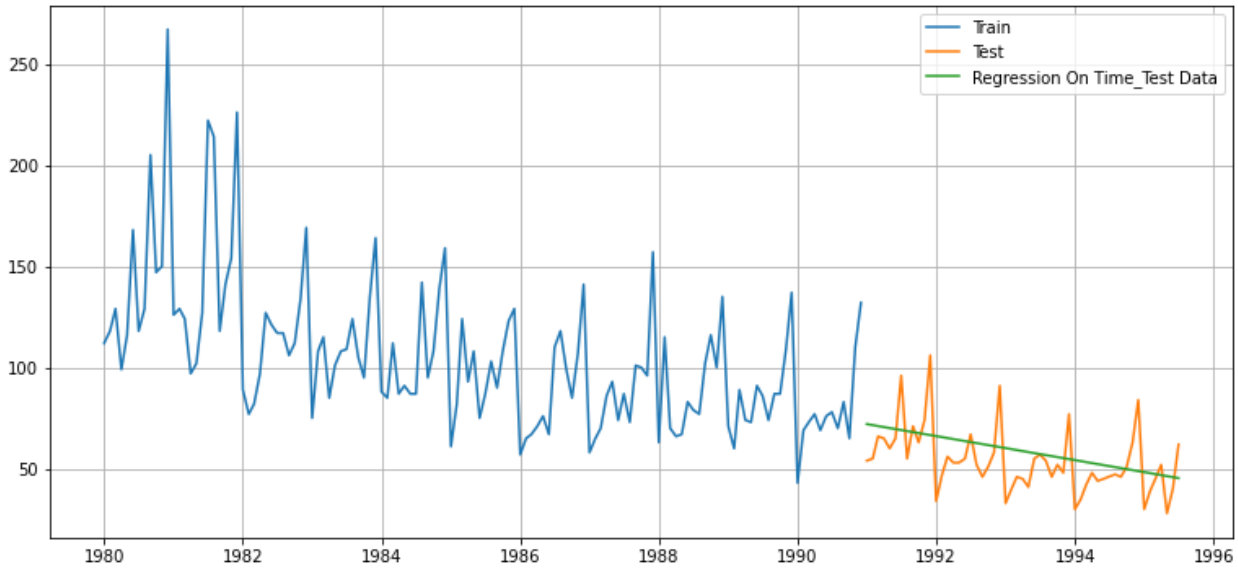


Figure 53.Rose Linear Regression Prediction Plot

From the above plot we can conclude that Linear Regression model doesn't predict well with the test data. However, let's continue with the model evaluation and store the results in the dataframe.

Model Evaluation on Test Data

For RegressionOnTime forecast on the Test Data, RMSE is 15.255

	Test RMSE
RegressionOnTime	15.255492

Table 32. Rose Linear Regression RMSE Table

Naïve Approach

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

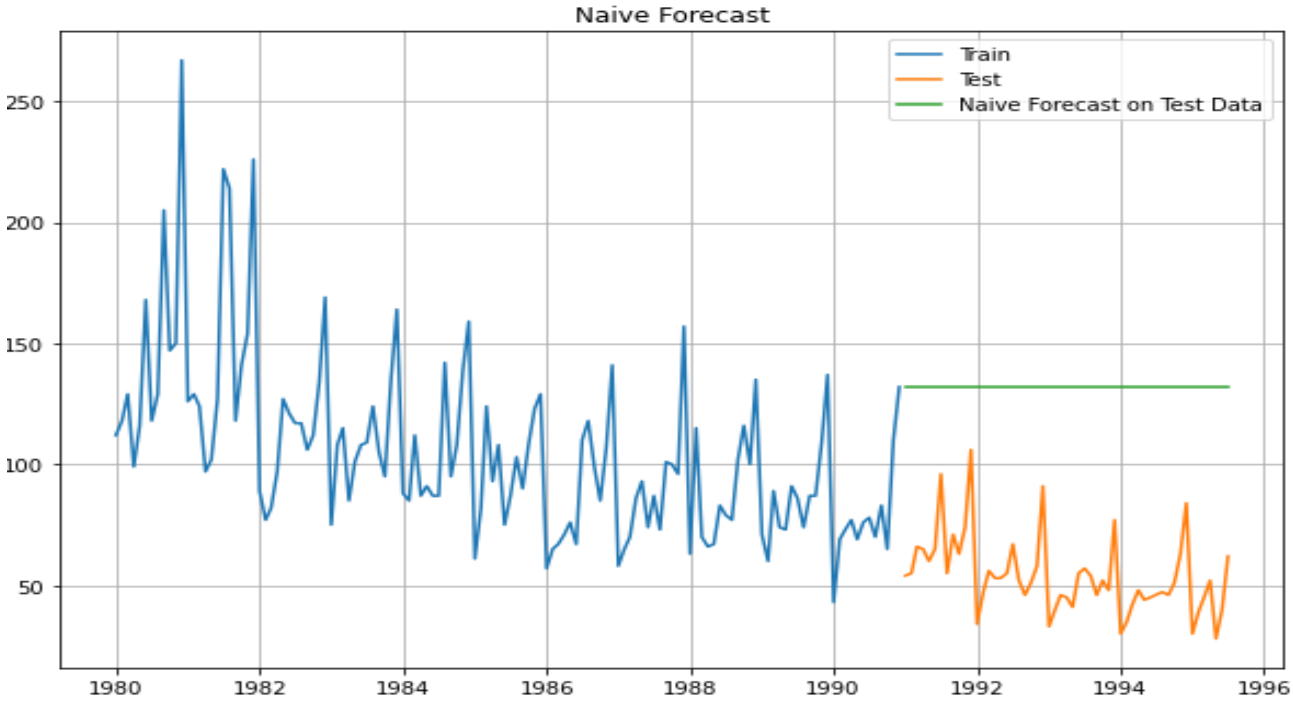


Figure 54.Rose Naïve method Prediction Plot

From the above plot we can conclude that Naïve approach doesn't predict well with the test data. It predicts as a straight line. However, let's continue with the model evaluation and store the results in the dataframe.

Model Evaluation on Test Data

For Naive forecast on the Test Data, RMSE is `79.672`

	Test RMSE
RegressionOnTime	15.255492
NaiveModel	79.672475

Table 33. Rose Naïve RMSE Table

Simple Average

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

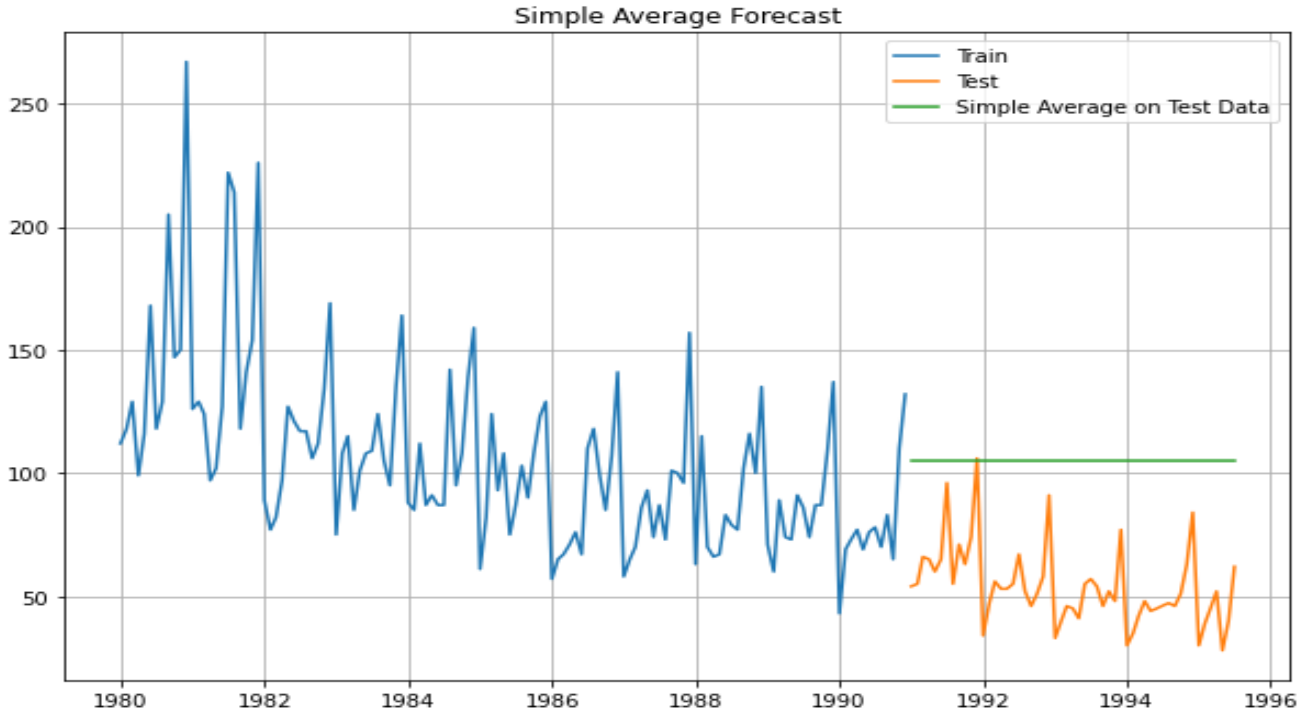


Figure 55.Rose Simple Average Prediction Plot

From the above plot we can conclude that Simple Average approach doesn't predict well with the test data. It predicts as a straight line. However, let's continue with the model evaluation and store the results in the dataframe.

Model Evaluation on Test Data

For Simple Average forecast on the Test Data, RMSE is `53.413`

	Test RMSE
RegressionOnTime	15.255492
NaiveModel	79.672475
SimpleAverageModel	53.413298

Table 34. Rose Simple Average RMSE Table

Moving Average(MA)

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

For Moving Average, we are going to average over the entire data.

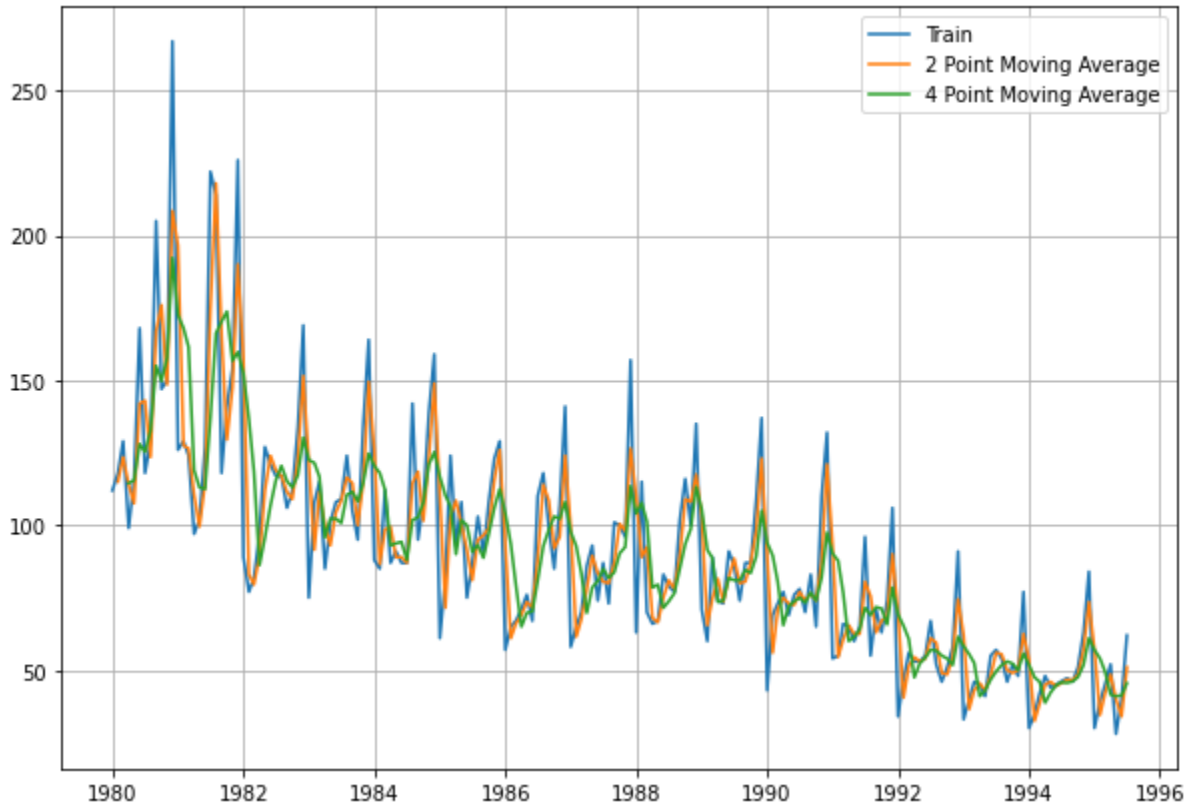


Figure 56.Rose Moving Average for whole data

From the above plot we can conclude that Moving Average approach is the best model built so far. Precisely, '2 point moving average' predicts well with the test data. Let's train the model using the Moving Average dataset and predict the test data.

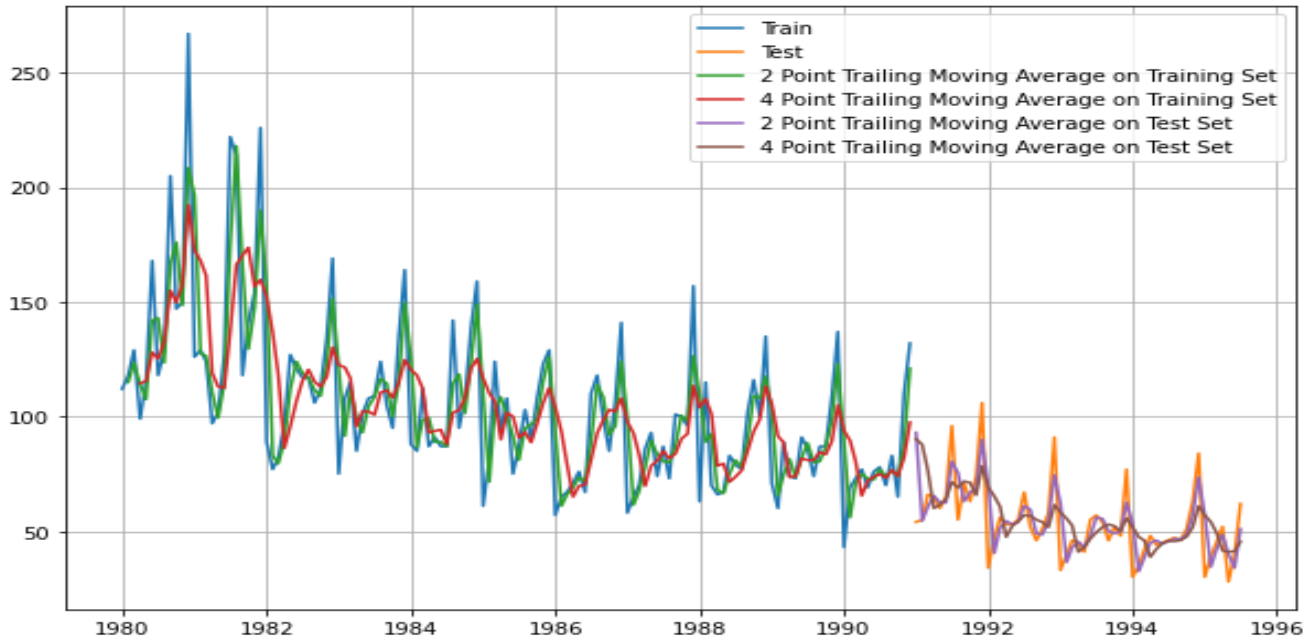


Figure 57.Rose Moving Average Prediction Plot

Even in the testing records, MA model performs well.

Let's continue with the model evaluation and store the results in the dataframe.

Model Evaluation on Test Data

For 2 point Moving Average Model forecast on the Training Data, RMSE is 11.530

For 4 point Moving Average Model forecast on the Training Data, RMSE is 14.444

	Test RMSE
RegressionOnTime	15.255492
NaiveModel	79.672475
SimpleAverageModel	53.413298
2pointTrailingMovingAverage	11.529985
4pointTrailingMovingAverage	14.444375

Table 35. Rose Moving Average RMSE Table

So far, we have got the best RMSE value for 2 point rolling average method.

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

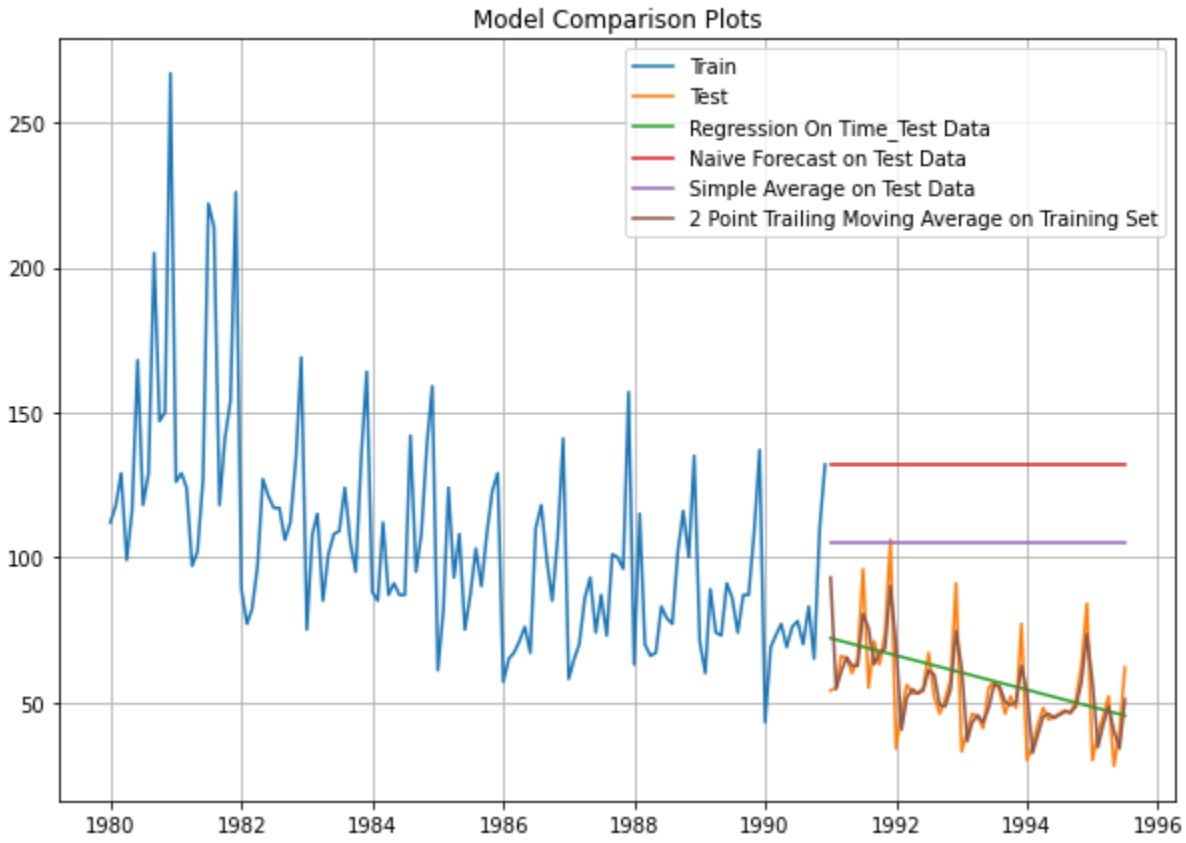


Figure 58.Model Comparison plots

Simple Exponential Smoothing

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

Let's instantiate the SES SimpleExpSmoothing() function importing from statsmodels library.

```
rmodel_SES =
SimpleExpSmoothing(rSES_train['Rose'],initialization_method='estimated')

rmodel_SES_autofit = rmodel_SES.fit(optimized=True)
```

Fitting the Simple Exponential Smoothing model and asking python to choose the optimal parameters

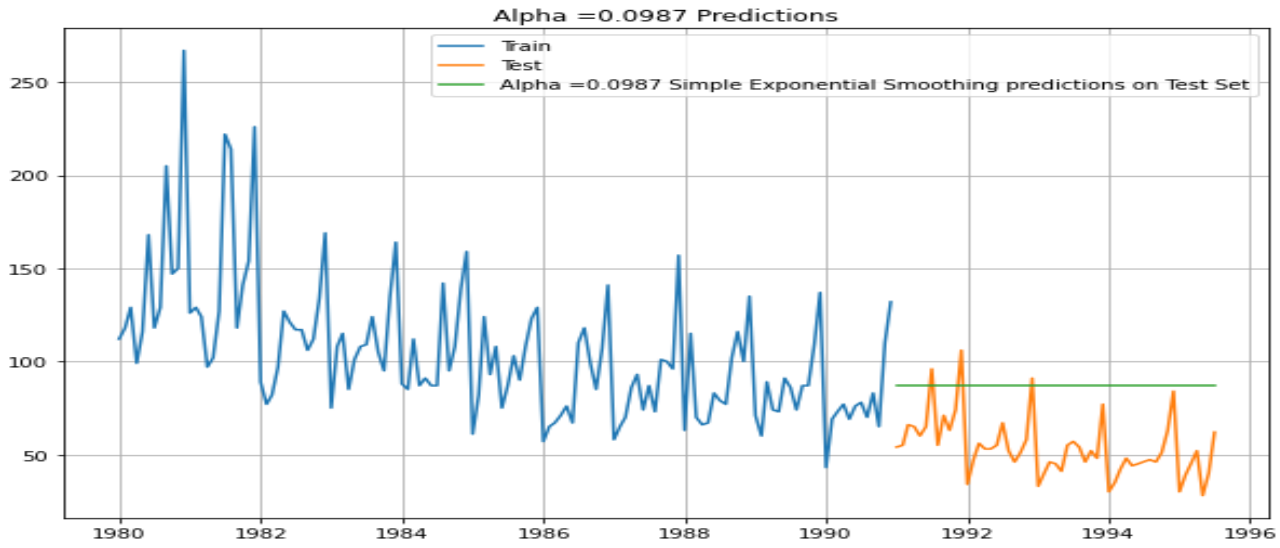


Figure 59.Rose auto alpha values SES Prediction Plot

From the above plot we can conclude that SES model approach doesn't predict well with the test data. It predicts as a straight line. However, let's continue with the model evaluation and store the results in the dataframe.

Model Evaluation on Test Data

For Alpha =0.0987 Simple Exponential Smoothing Model forecast on the Test Data, RMSE is `36.748`

	Test RMSE
RegressionOnTime	15.255492
NaiveModel	79.672475
SimpleAverageModel	53.413298
2pointTrailingMovingAverage	11.529985
4pointTrailingMovingAverage	14.444375
Alpha=0.0987,SimpleExponentialSmoothing	36.748401

Table 36.Rose auto SES RMSE table

Setting different alpha values. Remember, the higher the alpha value more weightage is given to the more recent observation. That means, what happened recently will happen again. We will run a loop with different alpha values to understand which particular value works best for alpha on the test set.

Since the best fit alpha is 0.0987, let's run the loop with the lower values such as 0.01, 0.05, 0.1 and 0.2 and capture the test data RMSE values in the dataframe.

	Alpha Values	Train RMSE	Test RMSE
2	0.10	32.253385	36.780184
1	0.05	33.193280	36.950813
3	0.20	32.155991	41.314129
0	0.01	35.889833	47.459871

Table 37.Rose tuning SES RMSE table

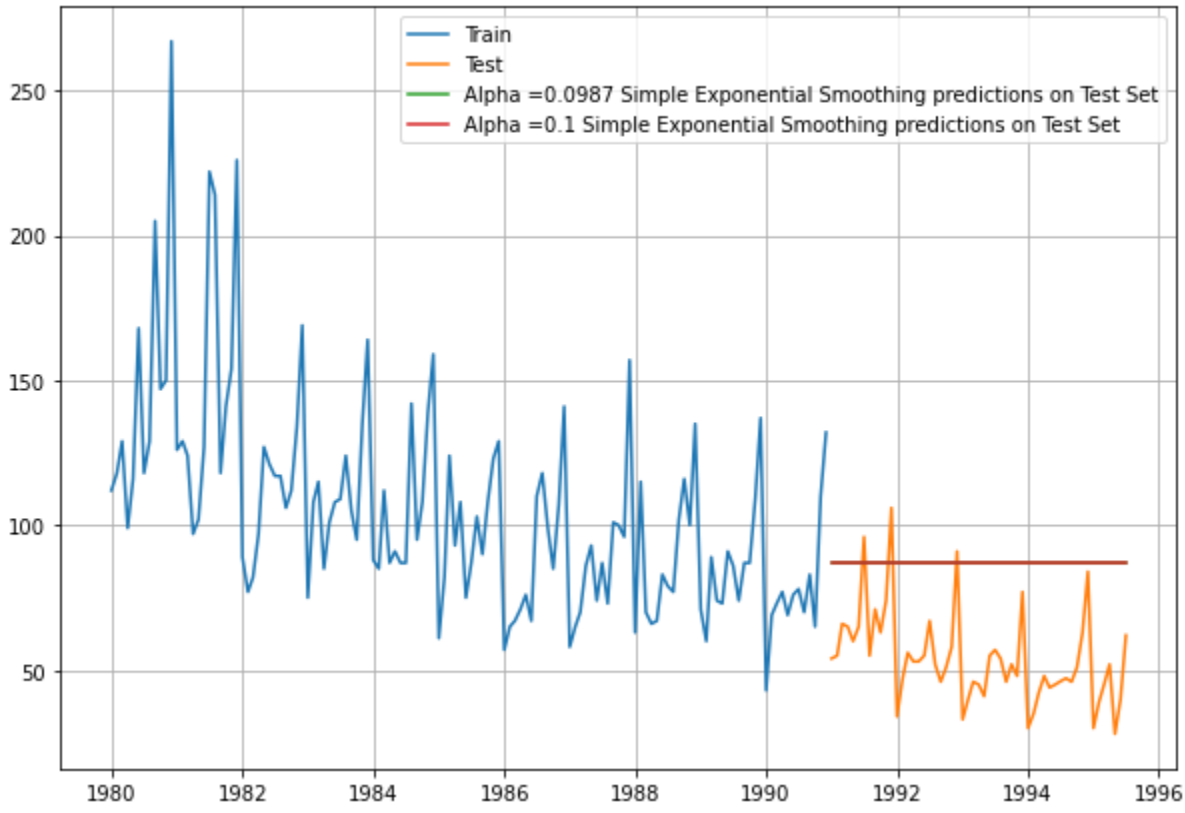


Figure 60.Rose SES Prediction Plot

As we presumed, SES is not the good fit for our Rose data.

	Test RMSE
RegressionOnTime	15.255492
NaiveModel	79.672475
SimpleAverageModel	53.413298
2pointTrailingMovingAverage	11.529985
4pointTrailingMovingAverage	14.444375
Alpha=0.0987,SimpleExponentialSmoothing	36.748401
Alpha=0.1,SimpleExponentialSmoothing	36.780184

Table 38.Rose SES RMSE table

Double Exponential Smoothing

Double Exponential Smoothing model is suitable to model the time series with trend but without seasonality. The Holt's linear exponential smoothing displays a constant trend indefinitely into the future. But, the trend and seasonality is present in our Sparkling data. Empirical evidence shows that the Holt's linear method tends to over-forecast.

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

Let's instantiate the Holt() function from statsmodels library

```
rmodel_DES = Holt(rDES_train['Rose'])
```

Let's run the loop with the values from 0.1 to 1.0 for both α and β and capture the test data RMSE values in the dataframe.

	Alpha Values	Beta Values	Train RMSE	Test RMSE
0	0.1	0.1	34.439111	36.874004
1	0.1	0.2	33.450729	48.634943
10	0.2	0.1	33.097427	65.677796
2	0.1	0.3	33.145789	78.100383
20	0.3	0.1	33.611269	98.598321

Table 39.Rose tuning DES RMSE table

The best RMSE values of DES are 0.1 for both α and β.

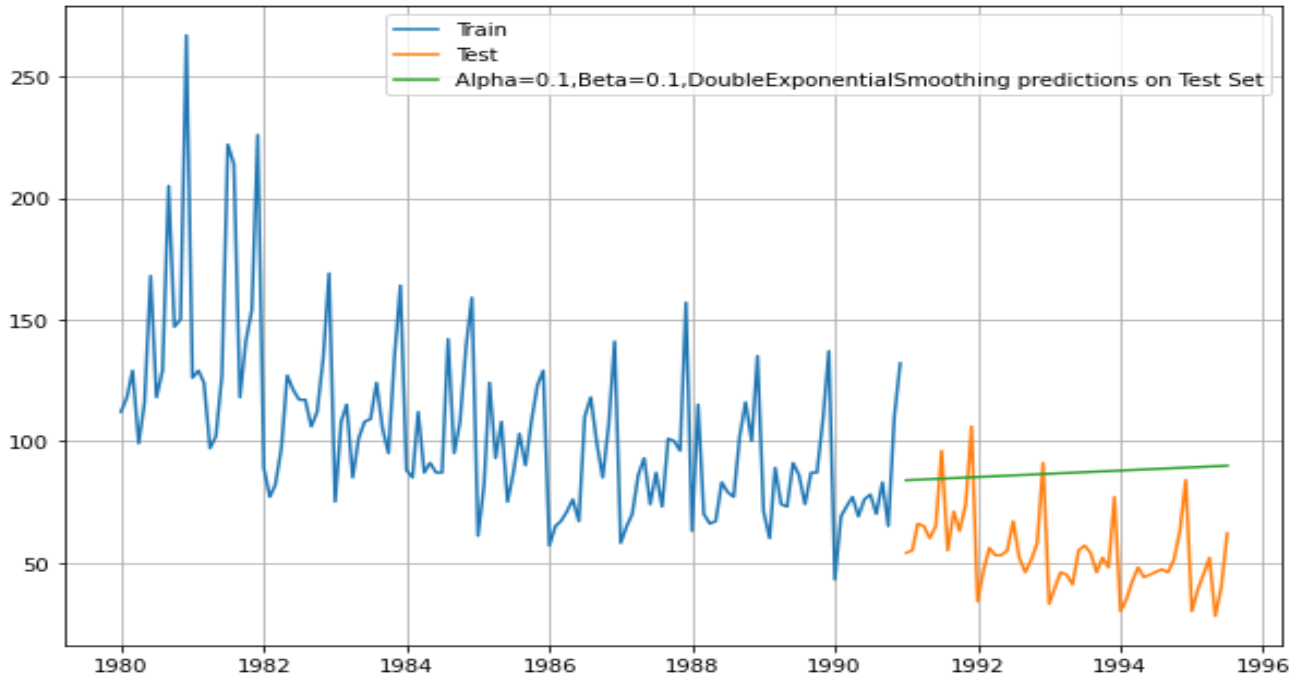


Figure 61.Rose DES Prediction Plot

From the above plot we can conclude that DES model approach doesn't predict well with the test data. It predicts as a straight line. However, let's continue with the model evaluation and store the results in the dataframe.

	Test RMSE
RegressionOnTime	15.255492
NaiveModel	79.672475
SimpleAverageModel	53.413298
2pointTrailingMovingAverage	11.529985
4pointTrailingMovingAverage	14.444375
Alpha=0.0987,SimpleExponentialSmoothing	36.748401
Alpha=0.1,SimpleExponentialSmoothing	36.780184
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	36.874004

Table 40.Rose DES RMSE table

Triple Exponential Smoothing

Triple exponential smoothing is used to handle the time series data containing a seasonal component. This method is based on three smoothing equations: stationary component, trend, and seasonal. Both seasonal and trend can be additive or multiplicative. Our sparkling data has both trend and seasonality and possibly the good fit for this data.

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

Let's instantiate the ExponentialSmoothing () function from statsmodels library

```
rmodel_TES =
ExponentialSmoothing(rTES_train['Rose'],trend='additive',seasonal='multiplica
tive')

rmodel_TES_autofit = rmodel_TES.fit()
```

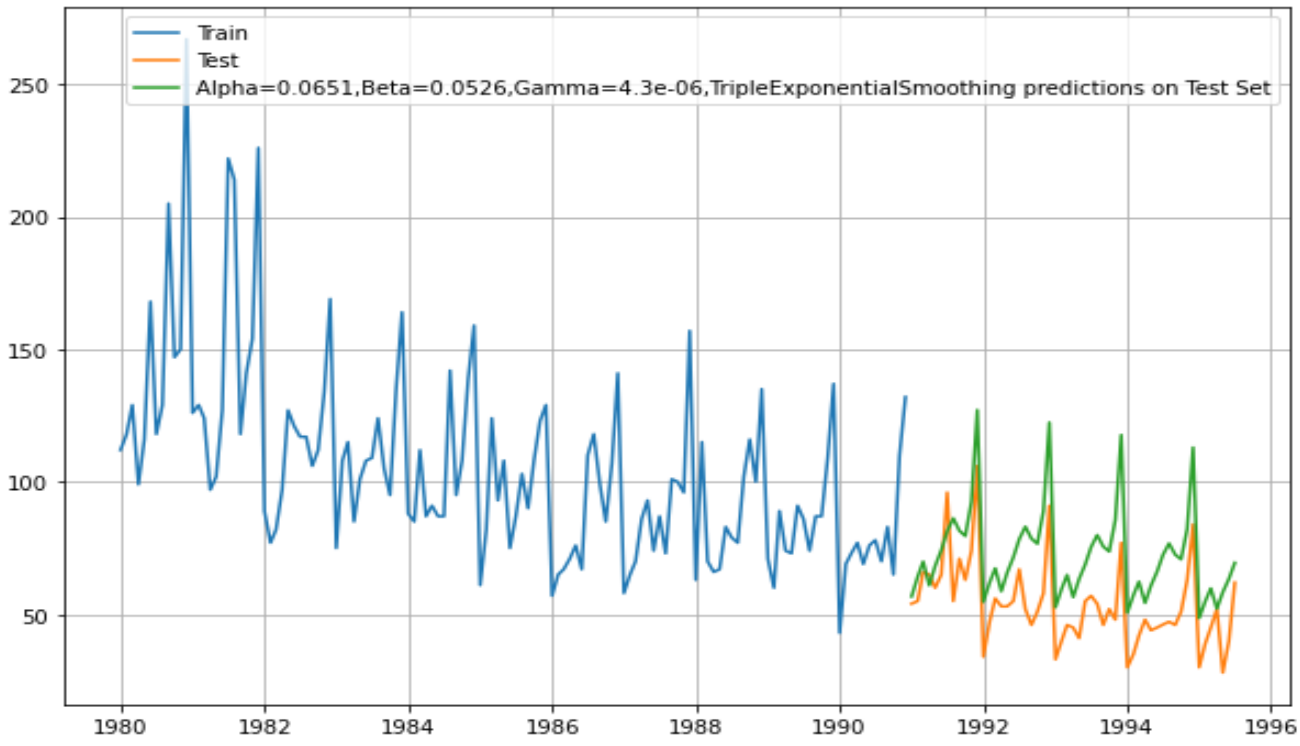


Figure 62.Rose auto TES values Prediction Plot

As we presumed, the above plot is good fit for our business problem

Model Evaluation on Test Data

For Alpha=0.0651, Beta=0.0526,Gamma=4.3e-06, Triple Exponential Smoothing Model forecast on the Test Data, RMSE is 21.038

	Test RMSE
RegressionOnTime	15.255492
NaiveModel	79.672475
SimpleAverageModel	53.413298
2pointTrailingMovingAverage	11.529985
4pointTrailingMovingAverage	14.444375
Alpha=0.0987,SimpleExponentialSmoothing	36.748401
Alpha=0.1,SimpleExponentialSmoothing	36.780184
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	36.874004
Alpha=0.0651,Beta=0.0526,Gamma=4.3e-06,TripleExponentialSmoothing	21.037637

Table 41.Rose auto TES RMSE table

Setting different Alpha, Beta and Gamma values. Let's run the loop with values and capture the test data RMSE values in the dataframe.

	Alpha Values	Beta Values	Gamma Values	Train RMSE	Test RMSE
11	0.1	0.2	0.2	24.365597	9.628012
12	0.1	0.2	0.3	23.969166	9.923487
10	0.1	0.2	0.1	25.529854	9.940633
142	0.2	0.5	0.3	27.631767	10.053020
151	0.2	0.6	0.2	28.289836	10.059361

Table 42.Rose tuning TES RMSE table

Upon tuning the α, β and γ values, the root mean square error is coming down compared to the autofit parameters, which is good sign of our model would perform well with the production records

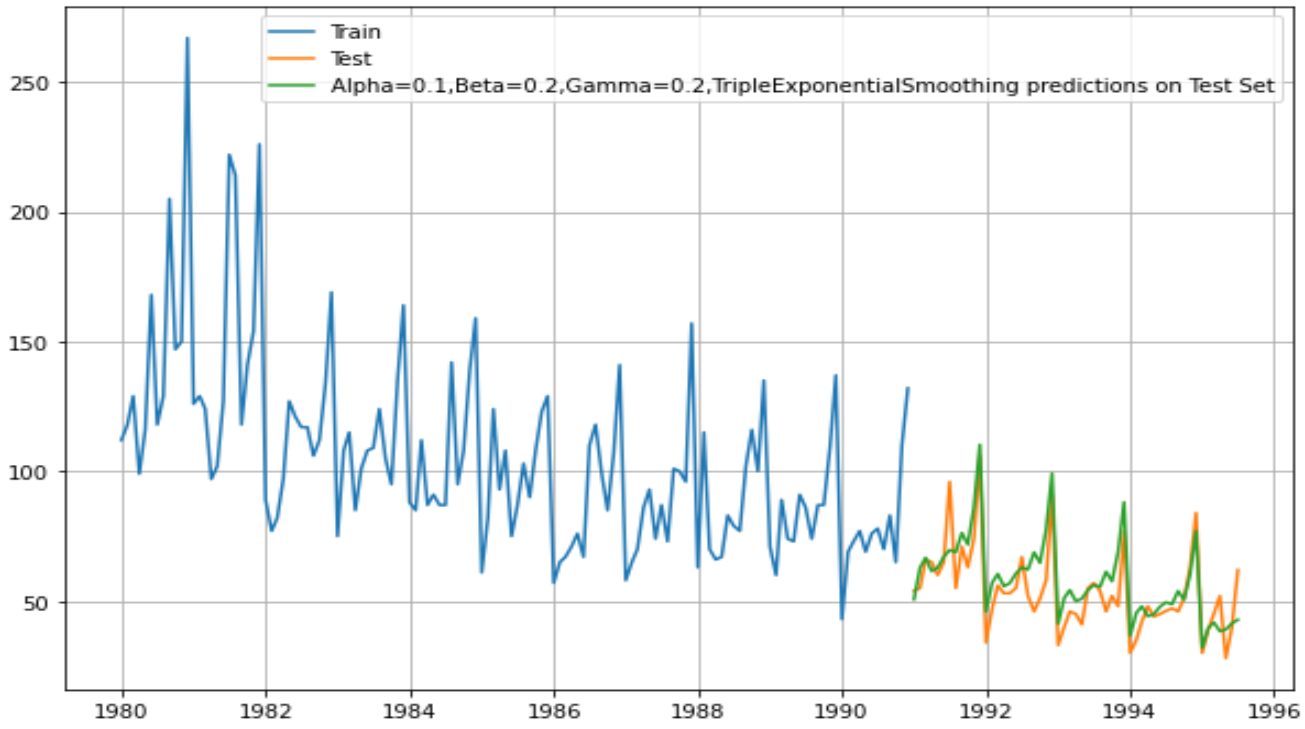


Figure 63.Rose TES Prediction Plot

RMSE Summary of the Models for Rose Test data

	Test RMSE
RegressionOnTime	15.255492
NaiveModel	79.672475
SimpleAverageModel	53.413298
2pointTrailingMovingAverage	11.529985
4pointTrailingMovingAverage	14.444375
Alpha=0.0987,SimpleExponentialSmoothing	36.748401
Alpha=0.1,SimpleExponentialSmoothing	36.780184
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	36.874004
Alpha=0.0651,Beta=0.0526,Gamma=4.3e-06,TripleExponentialSmoothing	21.037637
Alpha=0.1,Beta=0.2,Gamma=0.2,TripleExponentialSmoothing	9.628012

Table 43. Rose TES RMSE table

Optimized Model for Rose Data

	Test RMSE
Alpha=0.1,Beta=0.2,Gamma=0.2,TripleExponentialSmoothing	9.628012
2pointTrailingMovingAverage	11.529985
4pointTrailingMovingAverage	14.444375
RegressionOnTime	15.255492
Alpha=0.0651,Beta=0.0526,Gamma=4.3e-06,TripleExponentialSmoothing	21.037637
Alpha=0.0987,SimpleExponentialSmoothing	36.748401
Alpha=0.1,SimpleExponentialSmoothing	36.780184
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	36.874004
SimpleAverageModel	53.413298
NaiveModel	79.672475

Table 44.Best Rose RMSE table

From the above table, TES model with the tuning parameters results out the low RMSE values. Also, the curve of TES model series is cope with the test data correctly.

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.

Stationarity Check using Dickey Fuller Test

Auto-regressive (AR) and moving average (MA) models are popular models that are frequently used for forecasting. AR and MA models are combined to create models such as auto-regressive moving average (ARMA) and auto-regressive integrated moving average (ARIMA) models. ARIMA models are basically regression models; auto-regression means regression of a variable on itself measured at different time periods.

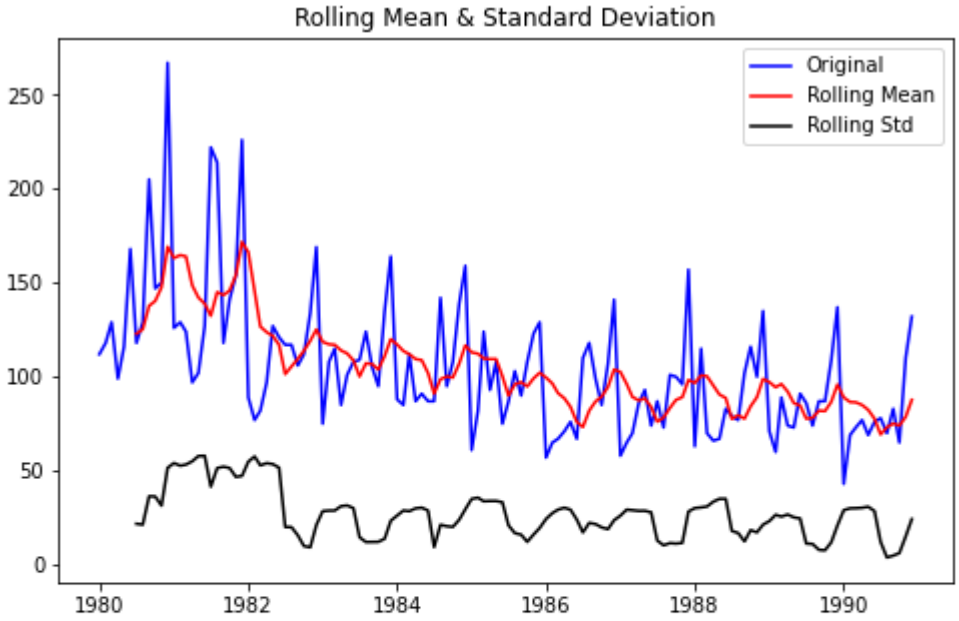
The main assumption of AR model is that the time series data is stationary.

A stationary time series is one whose statistical properties such as mean, variance, autocorrelation, etc. are all constant over time. When the time series data is not stationary, then we convert the non-stationary data before applying AR models.

The Augmented Dickey Fuller Test (ADF) is unit root test for stationarity. The null hypothesis is that time series is non-stationary. Alternative hypothesis is that time series is stationary.

From statsmodels library, lets import the adfuller function to perform ADF test.

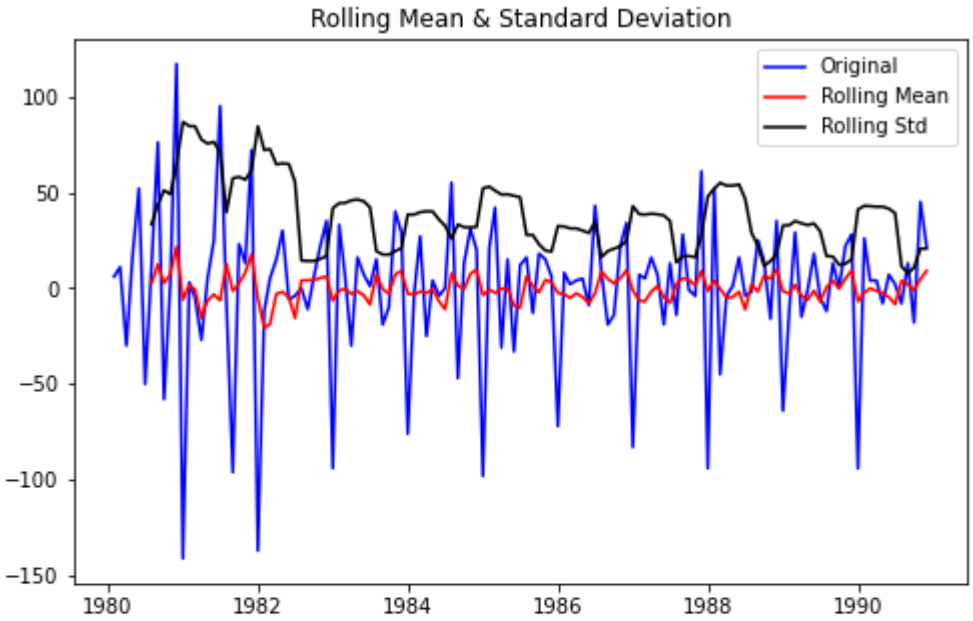
```
dftest = adfuller(timeseries, autolag='AIC')

test_stationarity(train['Rose'])
```




```
Results of Dickey-Fuller Test:
Test Statistic                  -1.686149
p-value                          0.756909
#Lags Used                      13.000000
Number of Observations Used    118.000000
Critical Value (1%)             -4.037614
Critical Value (5%)             -3.448373
Critical Value (10%)            -3.149257
dtype: float64
```

Figure 64.ADF Test for original Rose train data

Here, the p-value is greater than 0.05(alpha value). So, it is failed to reject the null hypothesis. i.e., data is not stationary. Taking the difference between consecutive observations is called a lag-1 difference to make it stationary. Let's take the first order difference in the data and run the ADF test again.




```
Results of Dickey-Fuller Test:
Test Statistic                -6.804433e+00
p-value                        3.894831e-08
#Lags Used                     1.200000e+01
Number of Observations Used    1.180000e+02
Critical Value (1%)           -4.037614e+00
Critical Value (5%)           -3.448373e+00
Critical Value (10%)          -3.149257e+00
dtype: float64
```

Figure 65.ADF Test after lag-1 difference in Rose train data

Now, the p-value is close to 0 and rejected the null hypothesis test. i.e., data is stationary and now we can apply the ARIMA models.

Stationary plot for Rose train data

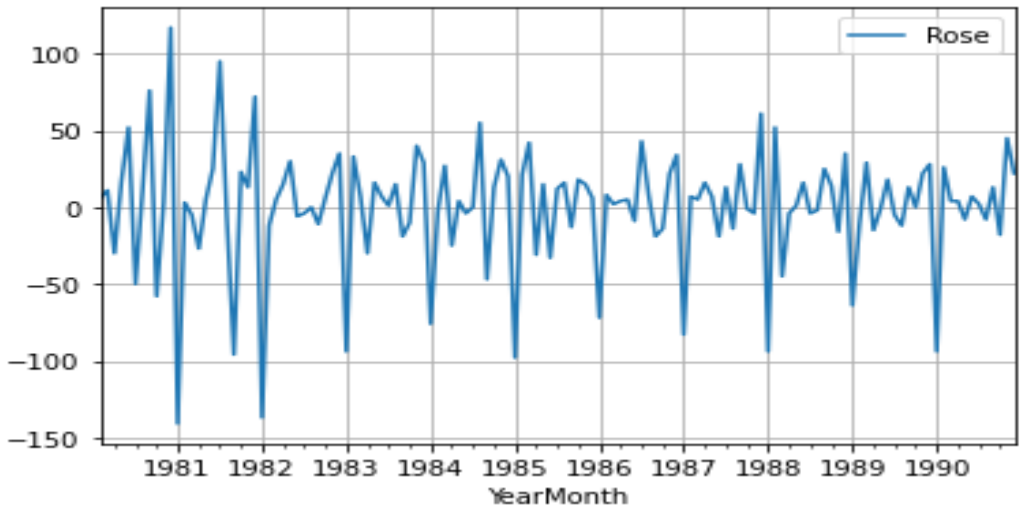


Figure 66.Stationary Rose Train data plot

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.

Building auto ARIMA model using AIC values

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

Specifically for ARIMA model, ARIMA (p, d, q) means that it you are describing some response variable (Y) by combining a 'p' order Auto-Regressive model and a 'q' order Moving Average model. A good way to think about it is (AR, I, MA).

Through 'itertools' python package, let's create the series of loop for ARIMA model parameters.

- p and q values from 0 to 3
- d=1 because it becomes non-stationary to stationary after differencing it with 1 time itself.

Let's import the ARIMA function from statsmodels.tsa.arima.model time series library and run through the different combinations of pq values with d=1

We have got the best AIC values for ARIMA(2,1,3) model.

	param	AIC
11	(2, 1, 3)	1274.695098
15	(3, 1, 3)	1278.658314
2	(0, 1, 2)	1279.671529

Table 45. AIC table for Rose AUTO ARIMA

Fit this model and see the summary below.

SARIMAX Results

Dep. Variable:	Rose	No. Observations:	132			
Model:	ARIMA(2, 1, 3)	Log Likelihood	-631.348			
Date:	Thu, 10 Feb 2022	AIC	1274.695			
Time:	12:20:51	BIC	1291.946			
Sample:	01-01-1980	HQIC	1281.705			
	- 12-01-1990					
Covariance Type:	opg					

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-1.6780	0.084	-20.018	0.000	-1.842	-1.514
ar.L2	-0.7288	0.084	-8.694	0.000	-0.893	-0.564
ma.L1	1.0448	0.649	1.609	0.108	-0.228	2.317
ma.L2	-0.7717	0.134	-5.751	0.000	-1.035	-0.509
ma.L3	-0.9045	0.589	-1.536	0.125	-2.059	0.250
sigma2	859.1032	546.561	1.572	0.116	-212.137	1930.343

Ljung-Box (L1) (Q):	0.02	Jarque-Bera (JB):	24.45
Prob(Q):	0.88	Prob(JB):	0.00
Heteroskedasticity (H):	0.40	Skew:	0.71
Prob(H) (two-sided):	0.00	Kurtosis:	4.57

Figure 67. Statistical summary for Rose AUTO ARIMA

From the statistical summary, 1st & 3rd component of Moving Average is not significant.

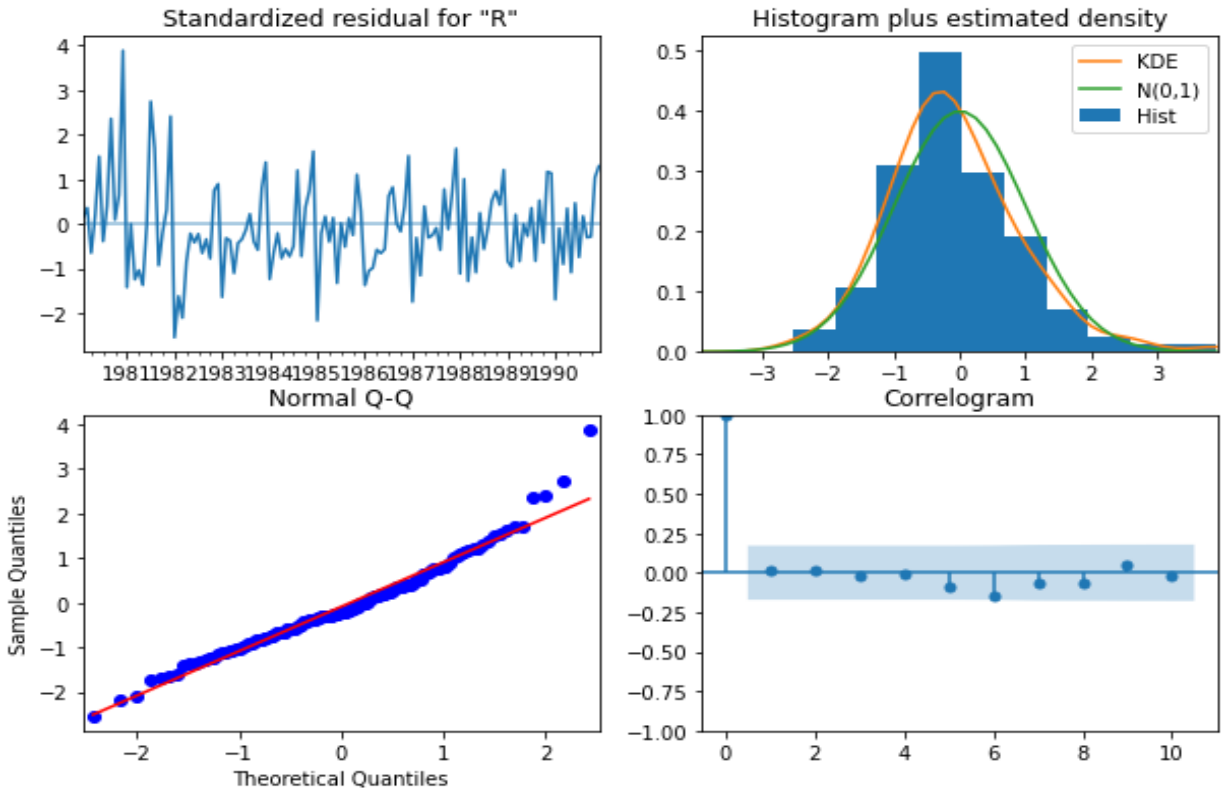


Figure 68. Rose Diagnostics plot for AUTO ARIMA

- The residuals of histogram states that they are almost uniformly distributed.
- The Q-Q plot indicates that quantiles are coming from the normal distribution as they are aligned with the line.
- The Correlogram shows the autocorrelation of the residuals and there is none of the components are significant.

Model Evaluation on Test Data

RMSE values for the ARIMA(2,1,3) shown below.

	Test RMSE
AUTO ARIMA(2,1,3)	36.768358

Table 46. Test Data RMSE for AUTO ARIMA(2,1,3)

Building auto SARIMA model using AIC values

Let's include the seasonality component in the ARIMA model and see it if it brings out the best RMSE values.

- pq/PQ values from 0 to 3
- d=1 because it becomes non-stationary to stationary after differencing it with 1 time itself.
- D=0 because the seasonality component is stationary after running with ADF test.
- Here the data provided is monthly, thus M=12.

Through 'itertools' python package, let's create the series of loop for SARIMA model parameters.

Let's import the SARIMAX function from statsmodels time series library and run through the different combinations of pq values with d=1 and PD values with D=0

	param	seasonal	AIC
222	(3, 1, 1)	(3, 0, 2, 12)	774.400285
238	(3, 1, 2)	(3, 0, 2, 12)	774.880937
220	(3, 1, 1)	(3, 0, 0, 12)	775.426699
221	(3, 1, 1)	(3, 0, 1, 12)	775.495330
252	(3, 1, 3)	(3, 0, 0, 12)	775.561018

Table 47. AIC table for Rose AUTO ARIMA

Fit this model and see the summary below.

SARIMAX Results

Dep. Variable:	Rose	No. Observations:	132
Model:	SARIMAX(3, 1, 1)x(3, 0, [1, 2], 12)	Log Likelihood	-391.557
Date:	Thu, 10 Feb 2022	AIC	803.114
Time:	12:35:37	BIC	828.332
Sample:	01-01-1980	HQIC	813.292
	- 12-01-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.6115	0.124	-4.918	0.000	-0.855	-0.368
ar.L2	-0.3310	0.160	-2.073	0.038	-0.644	-0.018
ar.L3	-0.1995	0.106	-1.886	0.059	-0.407	0.008
ma.L1	-1.0000	516.355	-0.002	0.998	-1013.038	1011.038
ar.S.L12	0.7440	0.192	3.881	0.000	0.368	1.120
ar.S.L24	0.1256	0.167	0.753	0.451	-0.201	0.452
ar.S.L36	0.0482	0.061	0.795	0.426	-0.071	0.167
ma.S.L12	-1.0414	0.500	-2.083	0.037	-2.021	-0.062
ma.S.L24	-0.3816	0.277	-1.379	0.168	-0.924	0.161
sigma2	144.9404	7.48e+04	0.002	0.998	-1.47e+05	1.47e+05

Ljung-Box (L1) (Q):	0.75	Jarque-Bera (JB):	0.84
Prob(Q):	0.39	Prob(JB):	0.66
Heteroskedasticity (H):	0.84	Skew:	0.20
Prob(H) (two-sided):	0.63	Kurtosis:	2.74

Figure 69.Statistical summary for Rose AUTO SARIMA

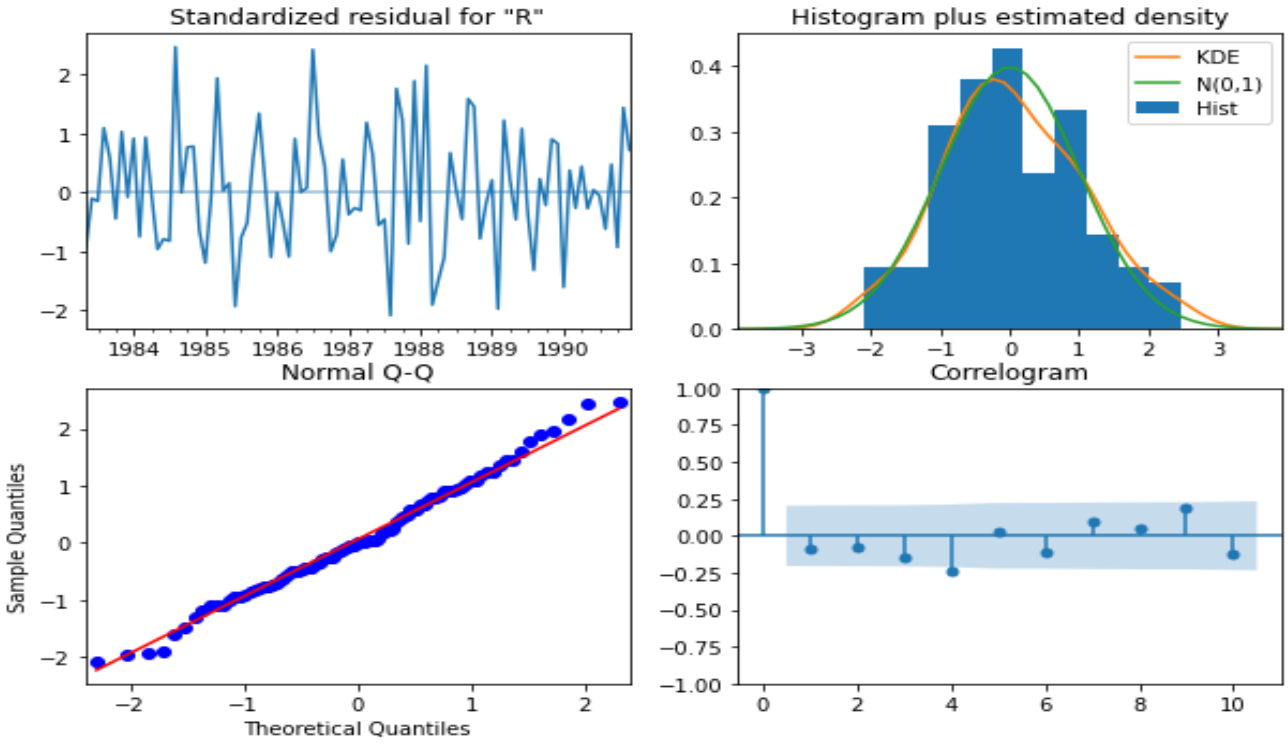


Figure 70. Rose Diagnostics plot for AUTO SARIMA

- The residuals of histogram states that they are not uniformly distributed.

- The Q-Q plot indicates that quantiles are coming from the normal distribution as they are almost aligned with the line. Only few points are moved away from the line.
- The Correlogram shows the autocorrelation of the residuals and none of the terms are exceeding the confidence intervals.

Model Evaluation on Test Data

RMSE values for the AUTO SARIMA(3, 1, 1)(3, 0, 2, 12) shown below.

	Test RMSE
AUTO SARIMA(3,1,1)(3,0,2,12)	57.956065

Table 48. Test Data for AUTO SARIMA

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.

Building ARIMA model using ACF and PACF plots

Train data becomes stationary after taking the 1st order difference. So, plot the same using the train.diff() values. Hence, **d=1**.

p=2(AR)

To estimate the amount of AR terms, you need to look at the PACF plot. First, ignore the value at lag 0. It will always show a perfect correlation, since we are estimating the correlation between today's values with itself. Note that there is a blue area in the plot, representing the confidence interval. To estimate how much AR terms you should use, start counting how many "lollipop" are above or below the confidence interval before the next one enter the blue area.

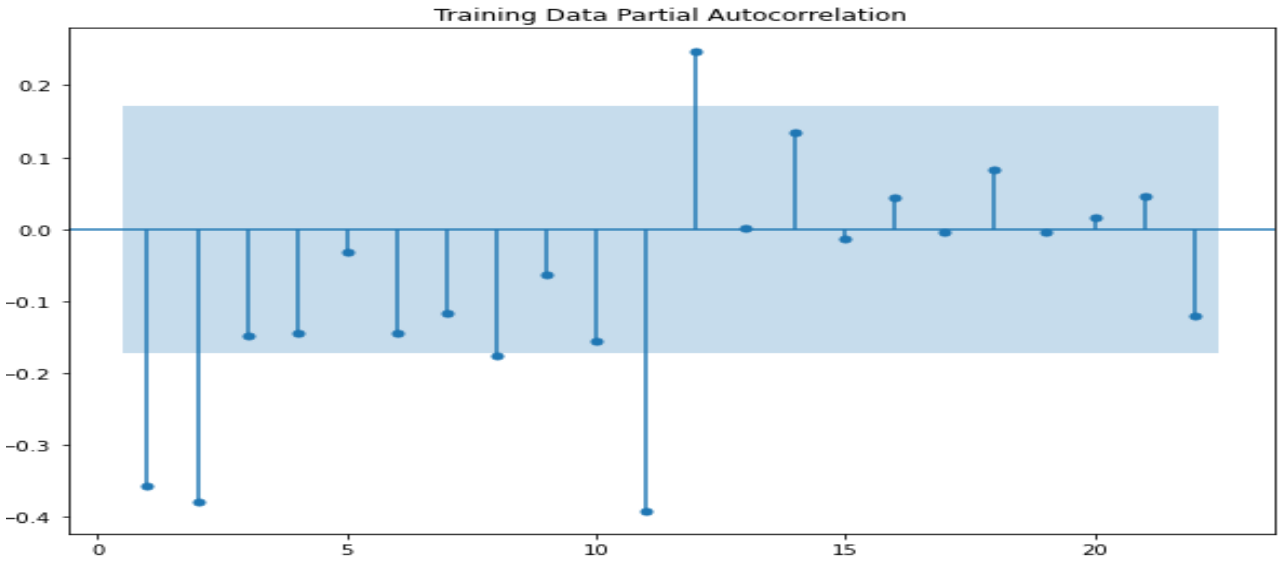


Figure 71. Rose Partial Autocorrelation plot

So, looking at the PACF plot above, we can estimate to use **2** AR terms for our model, since there is a cut-off after 2 terms.

q=2(MA)

To estimate the amount of MA terms, this time you will look at ACF plot. The same logic is applied here: how much lollipops are above or below the confidence interval before the next lollipop enters the blue area?

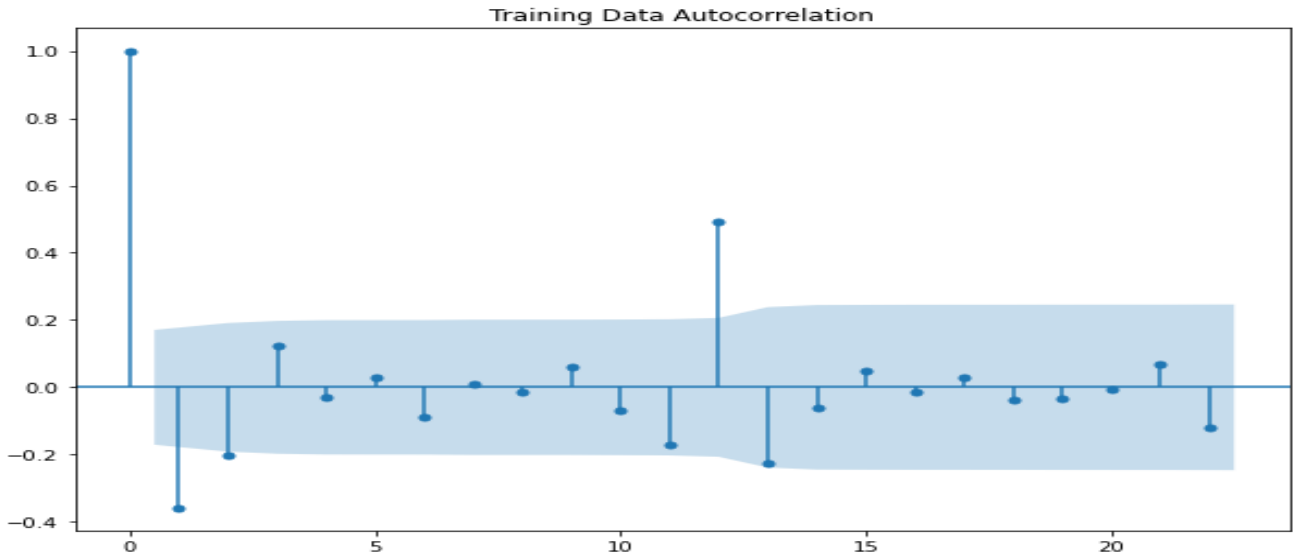


Figure 72. Rose Autocorrelation plot

In our example, we can estimate 2 MA terms, because there is no lag and the 1st component itself within the blue area.

Let's build and fit the ARIMA model using the parameters identified

```
rmanual_ARIMA = ARIMA(rtrain['Rose'], order=(2,1,2))
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                   Rose   No. Observations:                  132
Model:                 ARIMA(2, 1, 2)   Log Likelihood                -635.935
Date:                Thu, 10 Feb 2022   AIC                           1281.871
Time:                        12:46:16   BIC                           1296.247
Sample:                    01-01-1980   HQIC                          1287.712
                         - 12-01-1990
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.4540      0.469     -0.969      0.333      -1.372       0.464
ar.L2          0.0001      0.170      0.001      0.999      -0.334       0.334
ma.L1         -0.2541      0.459     -0.554      0.580      -1.154       0.646
ma.L2         -0.5984      0.430     -1.390      0.164      -1.442       0.245
sigma2       952.1601     91.424     10.415      0.000     772.973    1131.347
===================================================================================
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):                34.16
Prob(Q):                              0.88   Prob(JB):                         0.00
Heteroskedasticity (H):               0.37   Skew:                             0.79
Prob(H) (two-sided):                  0.00   Kurtosis:                         4.94
===================================================================================
```

Figure 73.Statistical summary for Rose Manual SARIMA

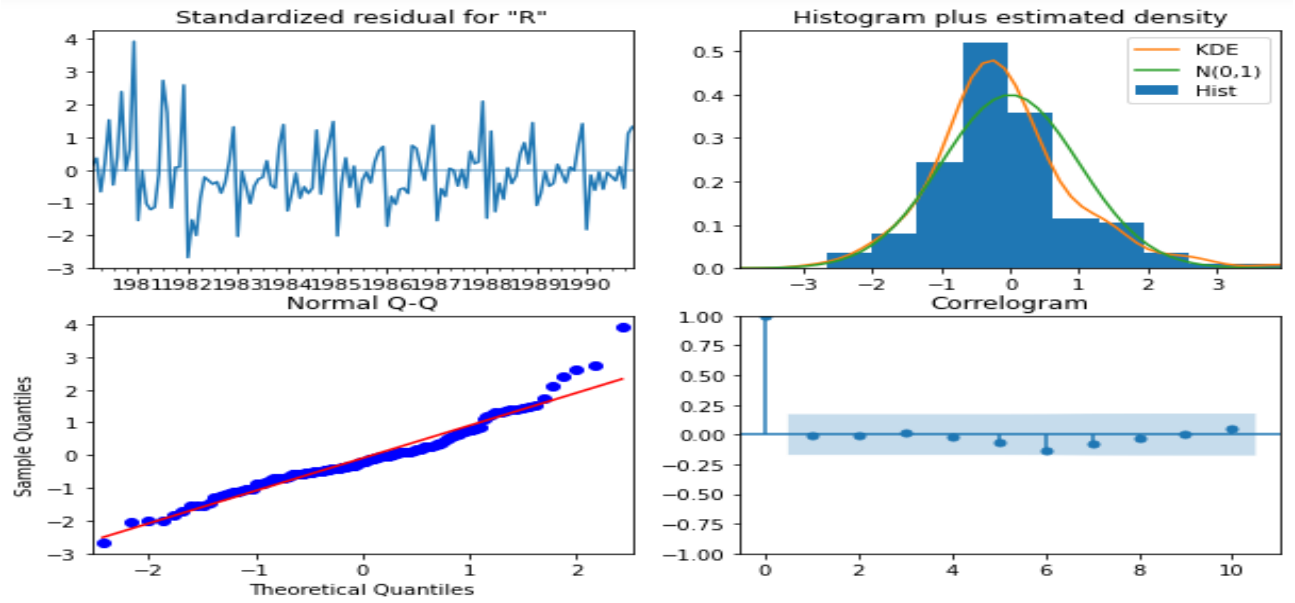


Figure 74. Rose Diagnostics plot for Manual SARIMA

The Normality Q-Q plot does not form the straight line, appears to be a not good fit for this problem.

Model Evaluation on Test Data

RMSE values for the ARIMA(2, 1, 2) shown below and so far the worst RMSE values.

	Test RMSE
Manual ARIMA(2,1,2)	36.82342

Table 49. Test RMSE for Rose Manual ARIMA(2,1,2)

Building SARIMA model using ACF and PACF plots

The process is quite similar to non-seasonal AR, and you will still be using the ACF and PACF function for that. To estimate the amount of AR terms, you will look one more time to the PACF function. Now, instead of count how many lollipops are out of the confidence interval, you will count how many **seasonal lollipops** are out.

The Q order can be calculated from the Autocorrelation (ACF) plot. Autocorrelation is the correlation of a single time series with a lagged copy of itself.

From the above graph, we note that the maximum lag with a value out the confidence intervals is 8, thus Q = 1(MA)

In the PACF graph the maximum lag with a value out the confidence intervals (in light blue) is 1, thus we can set P = 1(AR)

D=0 because the seasonality component is stationary after running with ADF test.

M=12. M indicates the periodicity, i.e. the number of periods in season, such as 12 for monthly data.

Let's build and fit the SARIMA model using the parameters identified

```
rmanual_SARIMA =
sm.tsa.statespace.SARIMAX(rtrain['Rose'],order=(2,1,2),seasonal_order=
(1, 0, 1, 12),enforce_stationarity=False,enforce_invertibility=False)
```

SARIMAX Results

Dep. Variable:	Rose	No. Observations:	132
Model:	SARIMAX(2, 1, 2)x(1, 0, [1], 12)	Log Likelihood	-515.095
Date:	Thu, 10 Feb 2022	AIC	1044.191
Time:	12:47:40	BIC	1063.466
Sample:	01-01-1980	HQIC	1052.016
	- 12-01-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.5632	0.165	-3.407	0.001	-0.887	-0.239
ar.L2	-0.0749	0.079	-0.946	0.344	-0.230	0.080
ma.L1	-0.1990	0.176	-1.129	0.259	-0.544	0.146
ma.L2	-0.7187	0.176	-4.087	0.000	-1.063	-0.374
ar.S.L12	0.8728	0.030	28.952	0.000	0.814	0.932
ma.S.L12	-1.2602	0.254	-4.962	0.000	-1.758	-0.763
sigma2	234.5599	70.190	3.342	0.001	96.991	372.129

Ljung-Box (L1) (Q):	0.01	Jarque-Bera (JB):	39.67
Prob(Q):	0.92	Prob(JB):	0.00
Heteroskedasticity (H):	0.46	Skew:	0.53
Prob(H) (two-sided):	0.02	Kurtosis:	5.66

Figure 75.Statistical summary for Rose Manual SARIMA

From the above statistical summary, the 2nd Auto Regression and 1st Moving Average term is not significant. All other terms are less than 0.05.

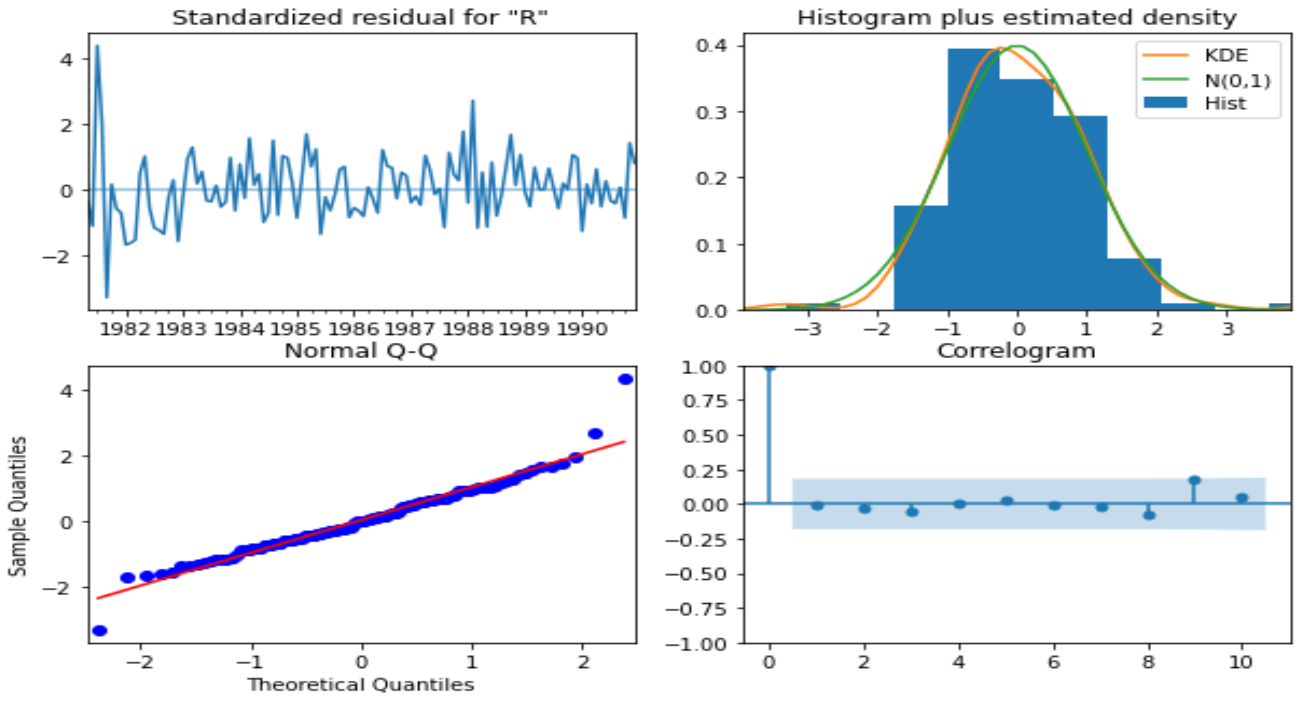


Figure 76. Rose Diagnostics plot for Manual SARIMA

- The residuals of histogram states that they are uniformly distributed.
- The Q-Q plot indicates that quantiles are coming from the normal distribution as they are almost aligned with the line.
- The Correlogram shows the autocorrelation of the residuals and none of the terms are exceeding the confidence intervals.
- Appears to be good fit for this Rose Wine sales data.

Model Evaluation on Test Data

RMSE values for the SARIMA (2, 1, 2)(1,0,1,12) shown below.

	Test RMSE
Manual SARIMA(2,1,2)(1, 0, 1, 12)	21.492688

Table 50.Test Data RMSE for Rose Manual SARIMA (2,1,2)(1,0,1,12)

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

Summary of all the Models

	Test RMSE
RegressionOnTime	15.255492
NaiveModel	79.672475
SimpleAverageModel	53.413298
2pointTrailingMovingAverage	11.529985
4pointTrailingMovingAverage	14.444375
Alpha=0.0987,SimpleExponentialSmoothing	36.748401
Alpha=0.1,SimpleExponentialSmoothing	36.780184
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	36.874004
Alpha=0.0651,Beta=0.0526,Gamma=4.3e-06,TripleExponentialSmoothing	21.037637
Alpha=0.1,Beta=0.2,Gamma=0.2,TripleExponentialSmoothing	9.628012
AUTO ARIMA(2,1,3)	36.768358
AUTO SARIMA(3,1,1)(3,0,2,12)	57.956065
Manual ARIMA(2,1,2)	36.823420
Manual SARIMA(2,1,2)(0, 1, 0, 12)	21.492688

Table 51. RMSE summary table for all models

By looking at the RMSE values for the test records, we can say Triple Exponential Smoothing, AUTO SARIMA and 2 point Trailing MA models yields the lower RMSE values.

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.

	Test RMSE
Alpha=0.1,Beta=0.2,Gamma=0.2,TripleExponentialSmoothing	9.628012
2pointTrailingMovingAverage	11.529985
4pointTrailingMovingAverage	14.444375
RegressionOnTime	15.255492
Alpha=0.0651,Beta=0.0526,Gamma=4.3e-06,TripleExponentialSmoothing	21.037637
Manual SARIMA(2,1,2)(0, 1, 0, 12)	21.492688
Alpha=0.0987,SimpleExponentialSmoothing	36.748401
AUTO ARIMA(2,1,3)	36.768358
Alpha=0.1,SimpleExponentialSmoothing	36.780184
Manual ARIMA(2,1,2)	36.823420
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	36.874004
SimpleAverageModel	53.413298
AUTO SARIMA(3,1,1)(3,0,2,12)	57.956065
NaiveModel	79.672475

Table 52. Ordered RMSE Summary Table for all models

We see that the best model is the Triple Exponential Smoothing with multiplicative seasonality with the parameters α = 0.4, β = 0.1 and γ = 0.2. Lets fit this model with the original data

```
fullmodel1 =
ExponentialSmoothing(df,trend='additive',seasonal='multiplicative').fi
t(smoothing_level=0.4,smoothing_trend=0.1,smoothing_seasonal=0.2)
```

RMSE for the full model is **376.7746**

Forecast the future for next 5 years (approximately)

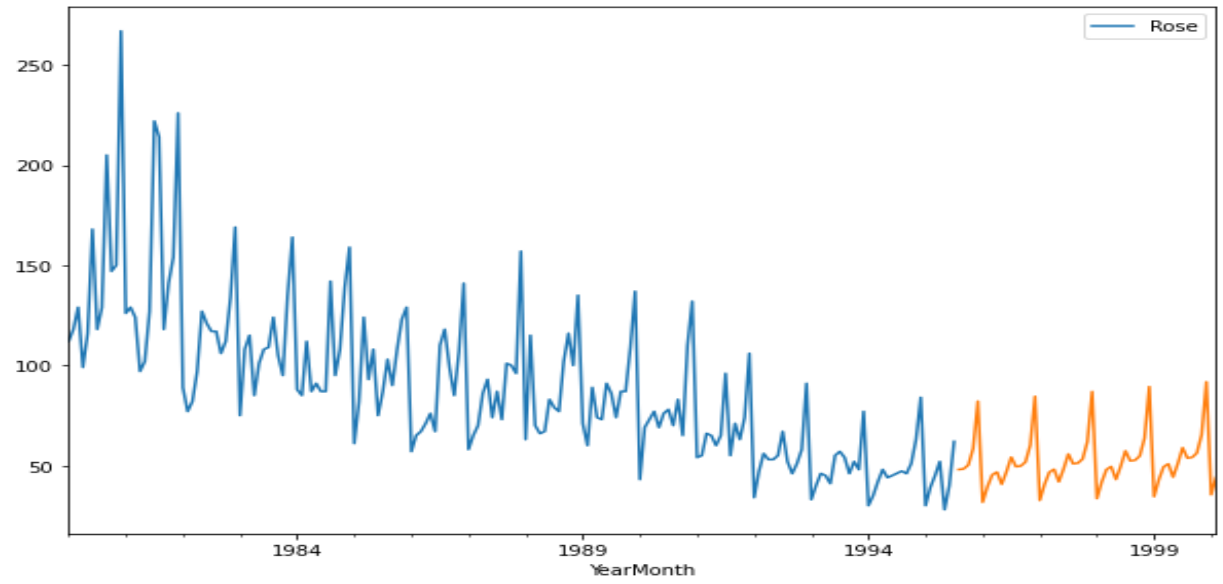


Figure 77.Forecast Rose data for next 5 years

Forecasting 12 months into the future

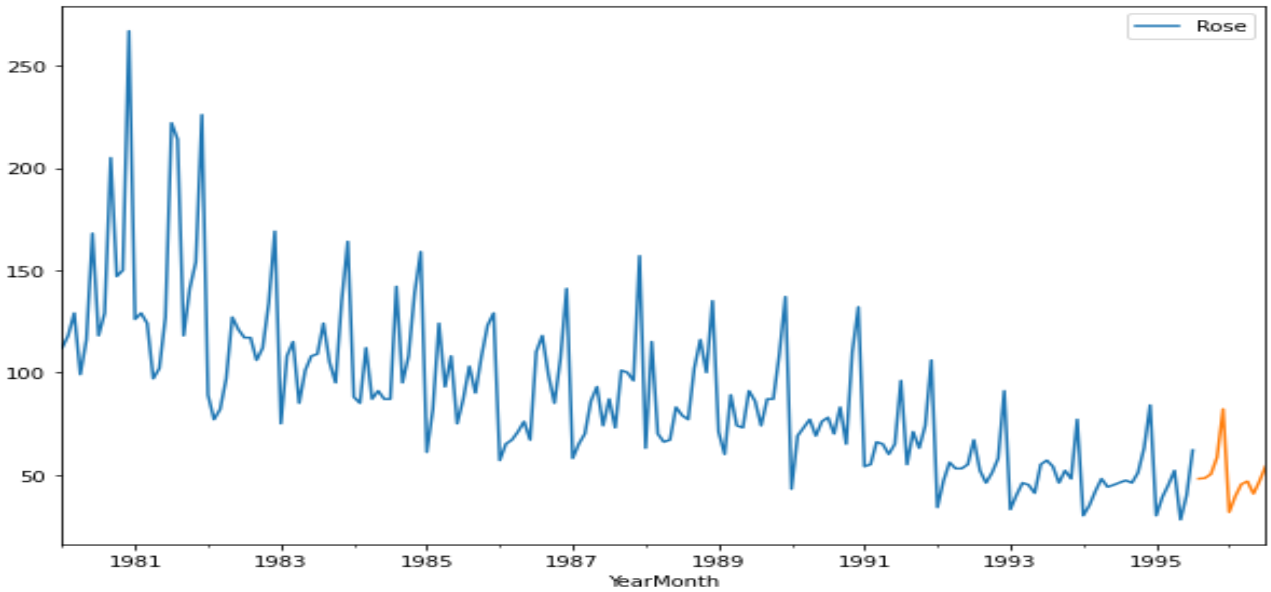


Figure 78.Forecast Rose data for next 12 months

Forecasting 12 months into the future with Confidence Intervals

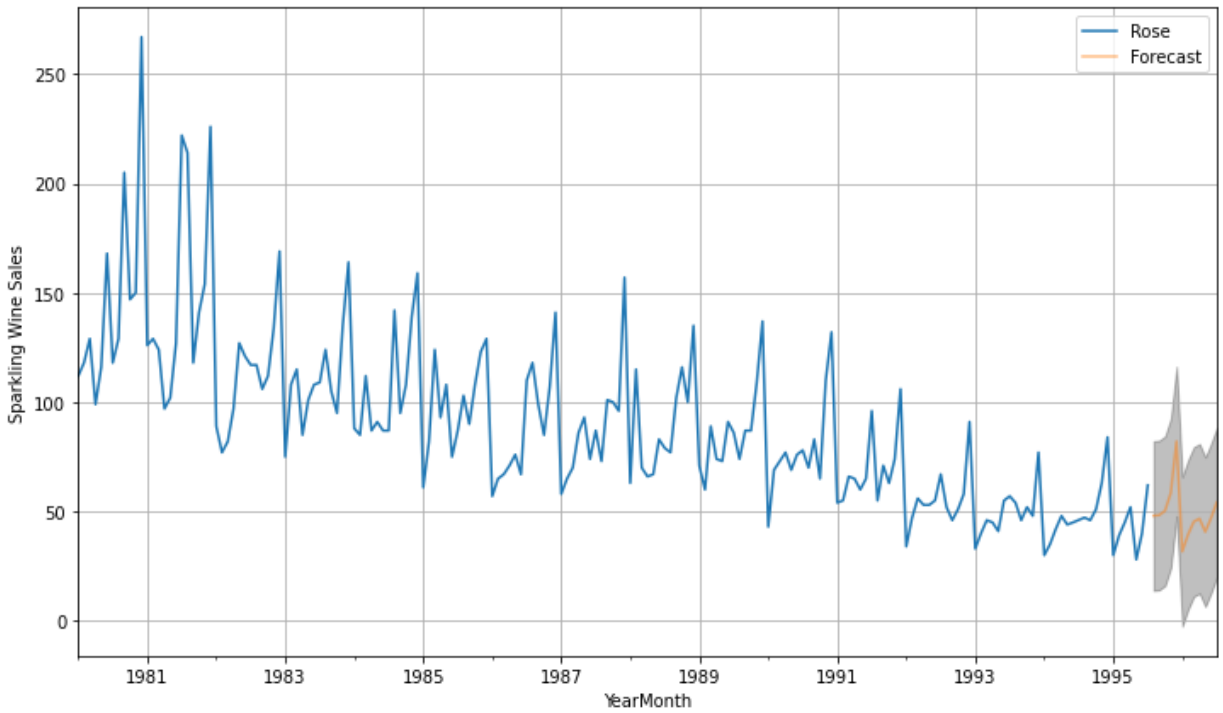


Figure 79.Forecast Rose data for next 12 months with Confidence Intervals

Forecasted 12 months values

1995-08-01	48.027146
1995-09-01	48.319651
1995-10-01	50.294057
1995-11-01	58.455624
1995-12-01	82.095526
1996-01-01	31.683779
1996-02-01	39.404477
1996-03-01	45.318818
1996-04-01	46.745747
1996-05-01	40.681298
1996-06-01	46.928740
1996-07-01	54.163611

Table 53.Forecasted values(12 months)

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

Recommendations and Insights

1. Among all the models builds, Triple Exponential Smoothing model outperforms all the other models. Next goes with the Moving Average models and the SARIMA models. The more complex the modell, the better it would work with the real-time data. Our Rose wine data is clearly showing up the decreasing trend and seasonality. TES model is used to handle the time series data containing trend and seasonality. This model is smoothening out the level, trend, seasonal components and it is the good fit for this data.
2. Over the years, there is no increase in the units of Rose wine sales rather it kept coming down to the worst in 1995 compared to 1980. The company should considering the removal of Rose Wine from the market. Because, there is no sign of sales being picked up over the years and it might not add any profit to the business improvements.
3. Conduct the survey to investigate the quality, price, taste and what kind of improvements can be made to the Rose wine. Alternatively, the estate can introduce the new type of wine with a new name accounting all the wrongdoing in the Rose wine.

THE END!