PGP - DSBA

Time Series Forecasting

Project Report – December 2022

Shruti Jha
12-11-2022

Great Learning
POWER AHEAD

Contents

List of Tables

List of Figures

ABC Estate Wines – Sales Forecasting

Introduction

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

Analysis to be performed on each of the two data sets separately.

1. Sparkling Wine Sales

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

The CSV file is loaded using pandas function read_csv() to perform analysis.

1.1.1 Sample of dataset

Here are the top 5 rows (sample) of the dataset:

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

Table 1. 1: Dataset Sample

- Dataset has 2 variables, showing the time component of the data and the sale of sparkling wine over the month and years.

1.1.2 Plotting data

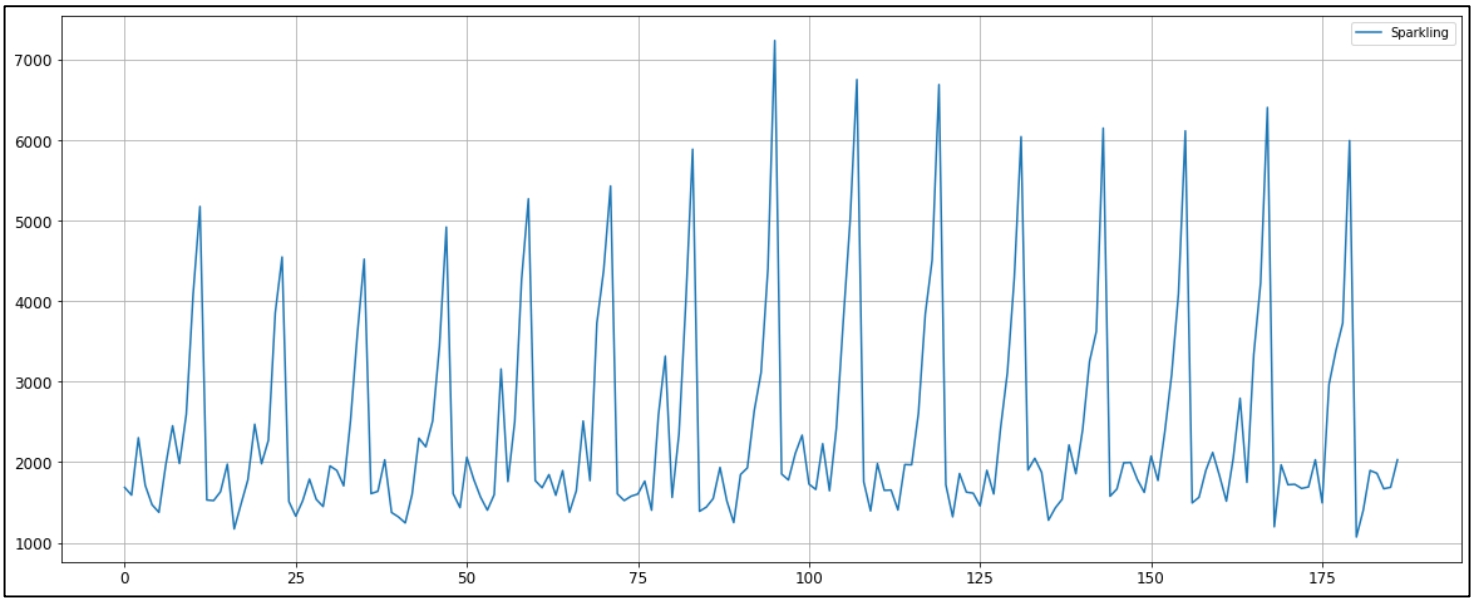


Figure 1. 1: Plotting Original Data

- Though the above plot looks like a Time Series plot, notice that the x-axis is not time. In order to make x-axis as time series, we passed the date range manually through 'date_range' command in pandas.
- After appending the time series, named 'Time_Stamp' column in our dataset, we assigned time series as index of our dataset.
- This is how the plotting of modified dataset looks:

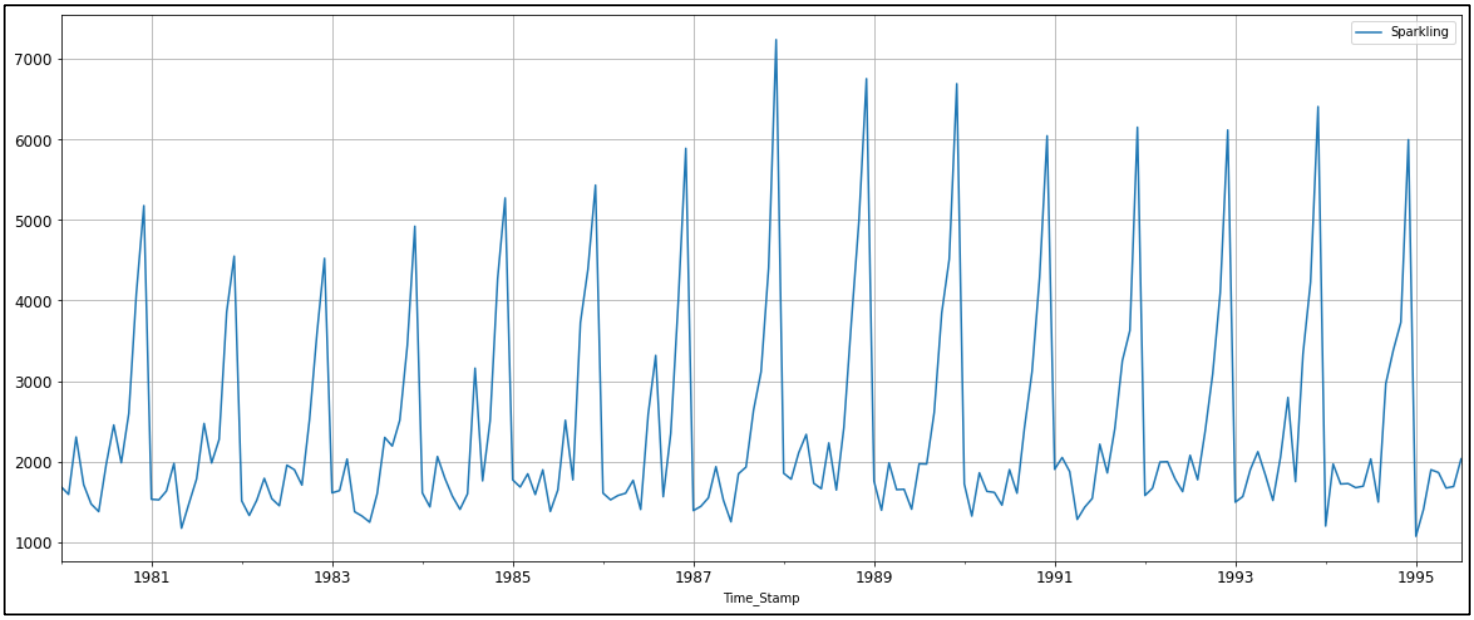


Figure 1. 2: Plotting Modified Data with Time_Stamp

- As we can observe now, x-axis is a time series, showcasing observations of wine sales over the years.
- There are trend and seasonality present in the time series. The magnitude of seasonality is changing with time, hence the seasonality is multiplicative.

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

1.2.1 Types of variables in the dataset

```
DatetimeIndex: 187 entries, 1980-01-31 to 1995-07-31
Data columns (total 1 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Sparkling  187 non-null    int64
dtypes: int64(1)
memory usage: 2.9 KB
```

- There are 187 observations in our data, ranging from January 1980 till July 1995.
- The 'Sparkling' column contains the wine sales volume for each month during the given period.
- There are no missing values present in the dataset.

1.2.2 Data Description

	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 1. 2: Data Description

- There are a total of 187 records, indicating that the data is of monthly frequency.
- The average sparkling wine sale of 187 months was 2402.42.
- The minimum sale was of 1070 wines and the maximum sale was of 7242 wines.
- The data description of a time series data doesn't paint a correct picture, as the factors like trend, seasonality and certain spikes in the outcome are not well accounted for.

1.2.3 Exploratory Data Analysis

Yearly Sales Bar Plot

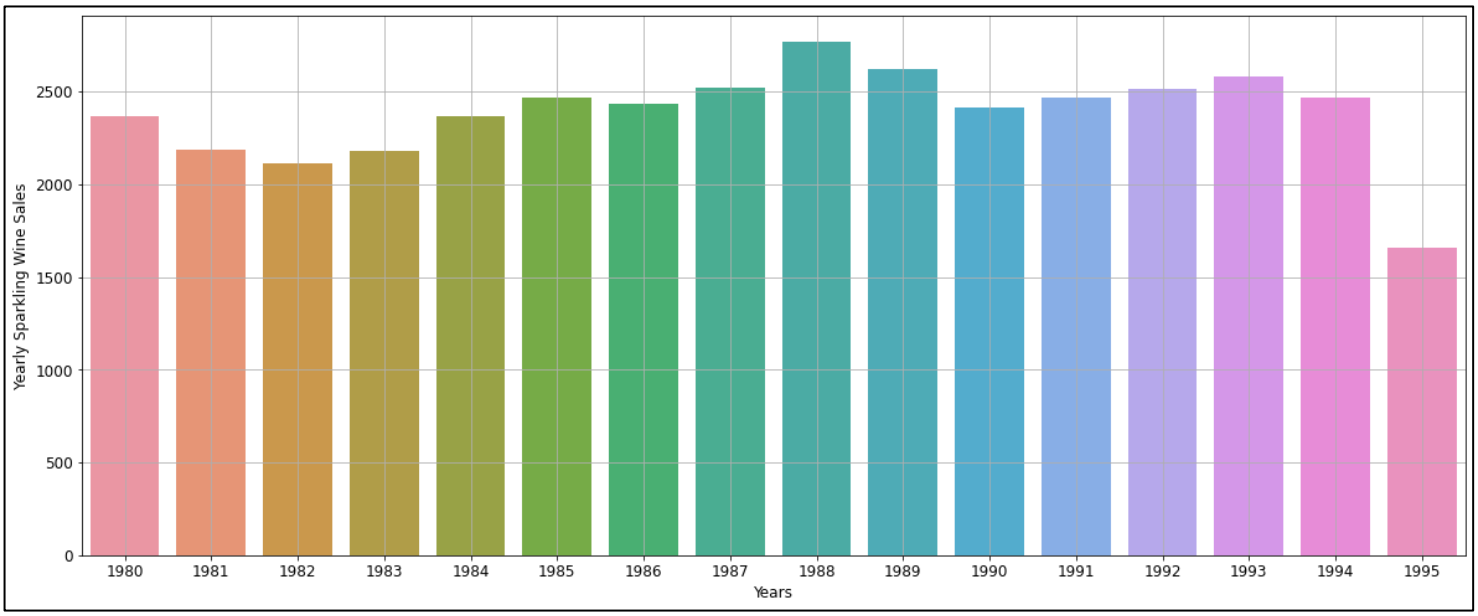


Figure 1. 3: Yearly Sales – bar plot

- From the yearly sales plot, we can see that year 1988 marks the highest sales of sparkling wine.
- Although, the lowest sales appear to be in the year 1995, but we only have 7 months data from that year. After that, year 1982 records minimum sales.

Yearly Sales Box Plot

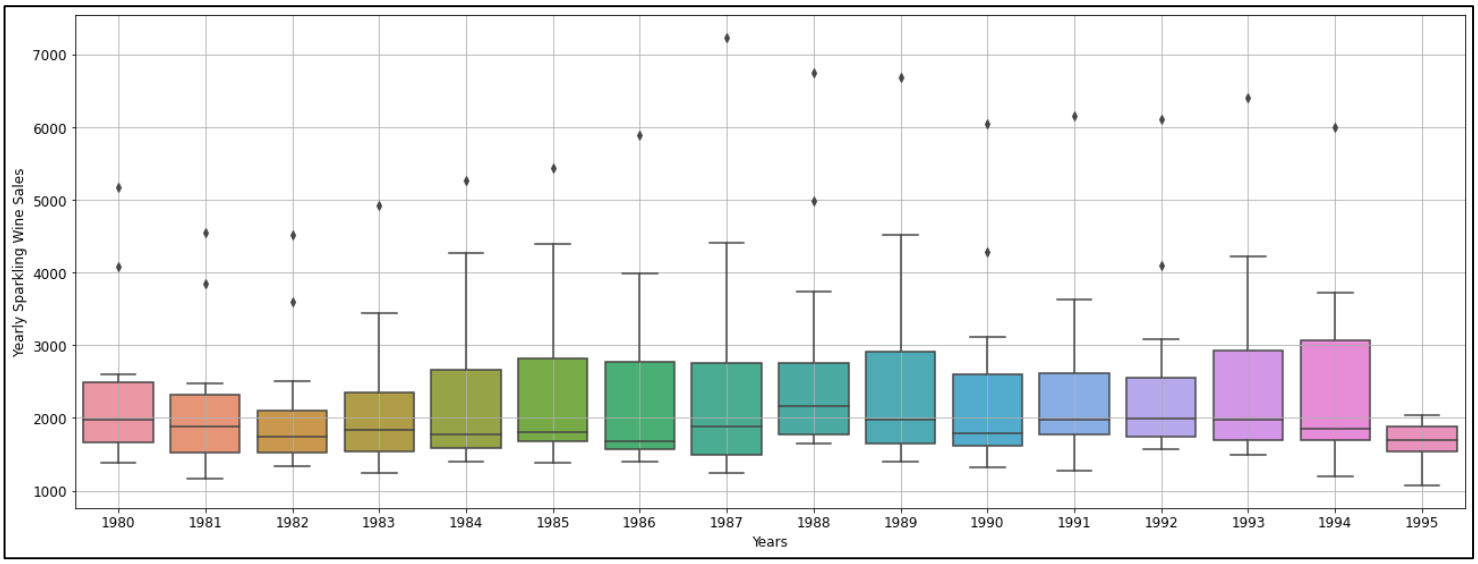


Figure 1. 4: Yearly Sales – box plot

- From the box plot we can observe that the year 1987 seems to be the most inconsistent in terms of wine sales. The median and upper limit are quite far from the outliers present in that year, we can say that the company was not prepared for that much of a high sales.
- We can say that company had least inconsistent sales in the years 1981, 1982 and 1995.

Monthly Sales Bar Plot

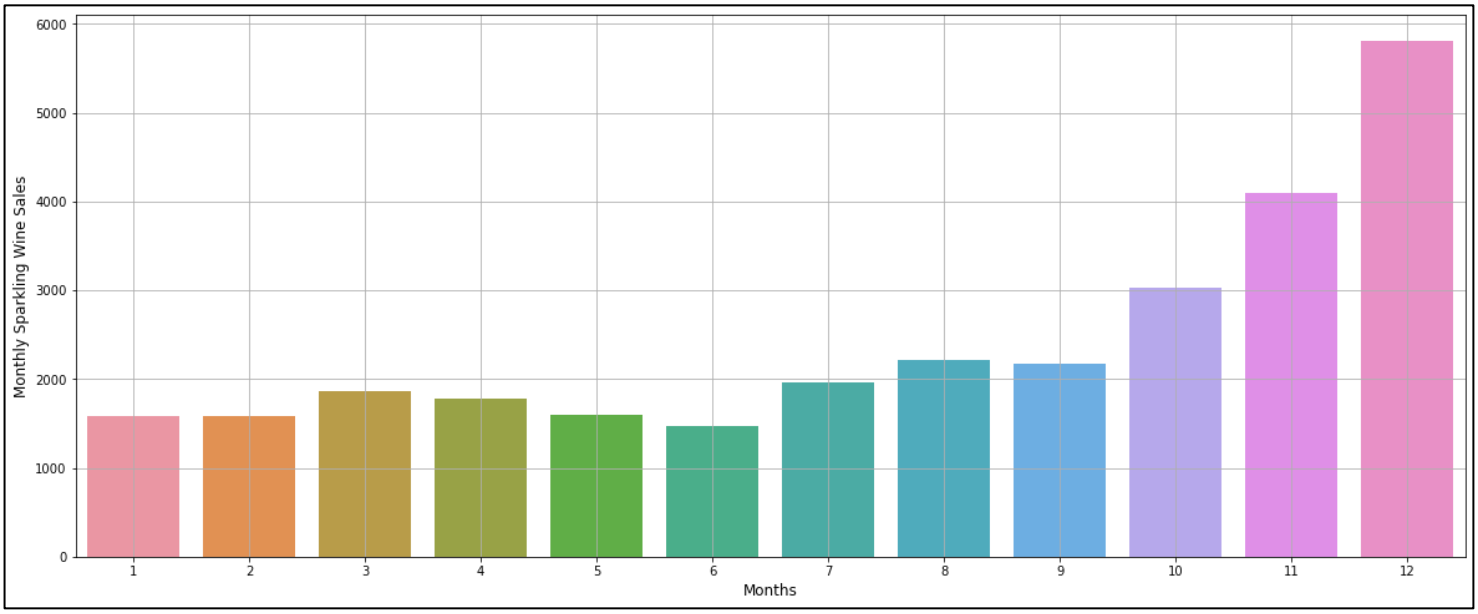


Figure 1. 5: Monthly Sales – bar plot

- Overall, December month accounts for highest sales of sparkling wine, provided that more people celebrate holiday season.
- The minimum wine sales observed in June month.
- The busiest time for the wine company is fourth quarter, given the festive season. The leanest period is second quarter, when the sale is not much.

Monthly Sales Box Plot

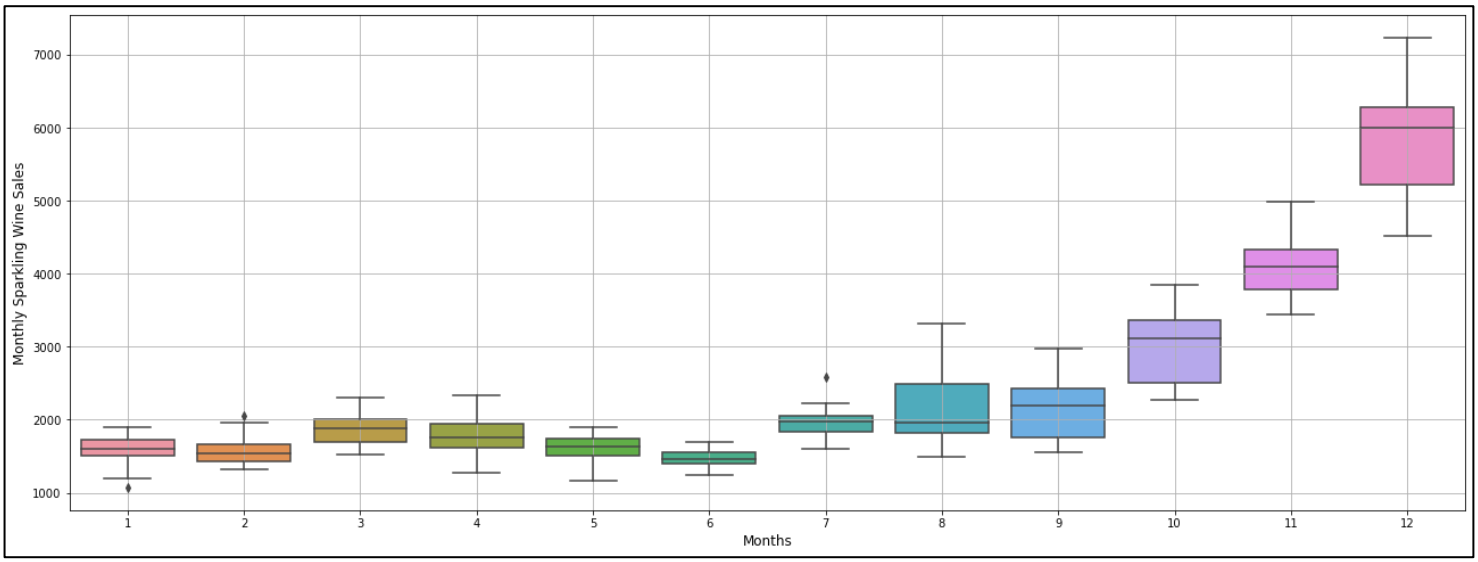


Figure 1. 6: Monthly Sales – box plot

- The most consistent month in terms of wine sales has been December.
- We can observe outliers in January, February and July months, where the least consistent month with sales is July.

Monthplot

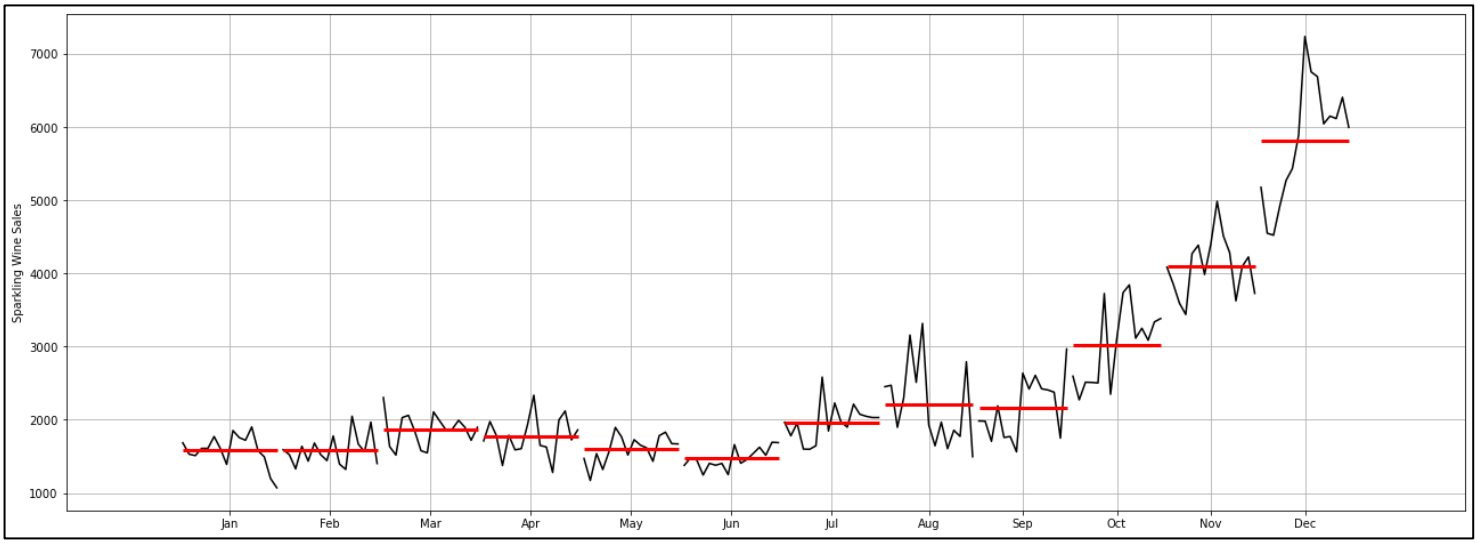


Figure 1. 7: Monthplot

- The high fluctuations can be observed in August and December months.
- June is the month with very low fluctuations.

Month on Month Sales Comparison

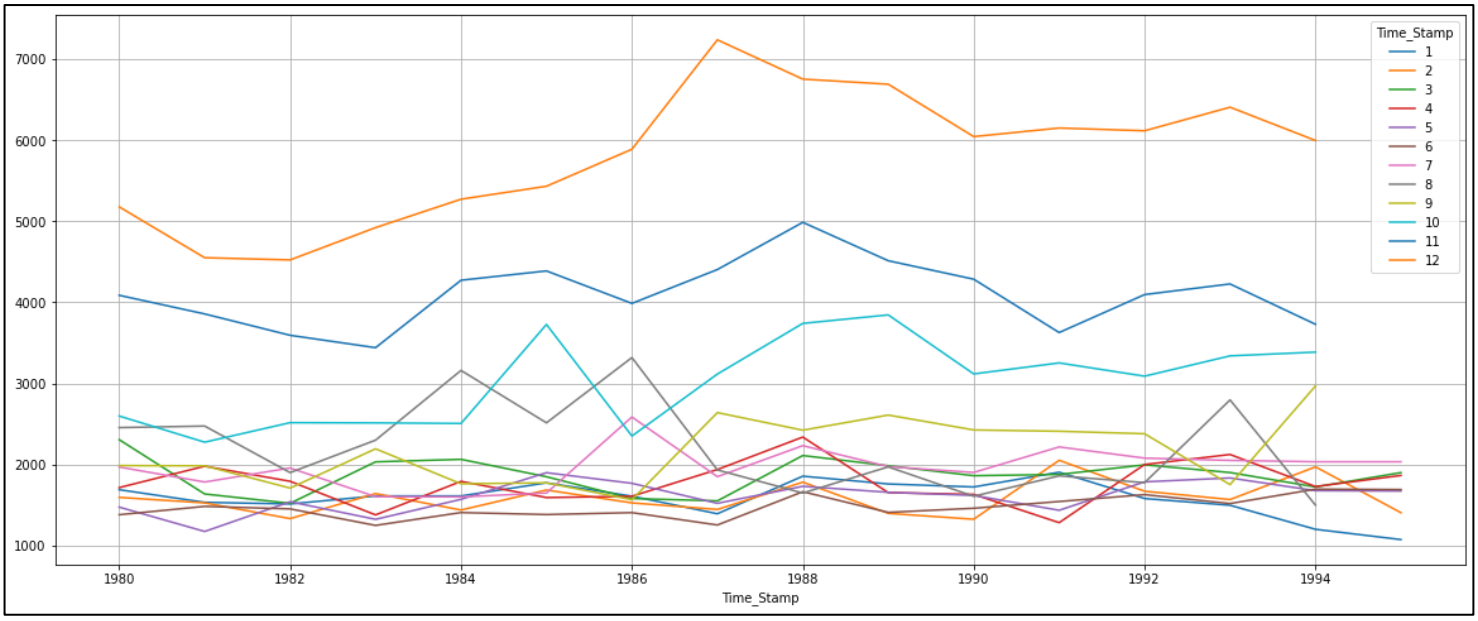


Figure 1. 8: Month on Month Sales Comparison

- December marks the highest number of sales.
- August onwards we can see increasing pattern in sales.
- Sales for rest all months are quite close to each other, with some spikes here and there.

Empirical Cumulative Distribution Plot

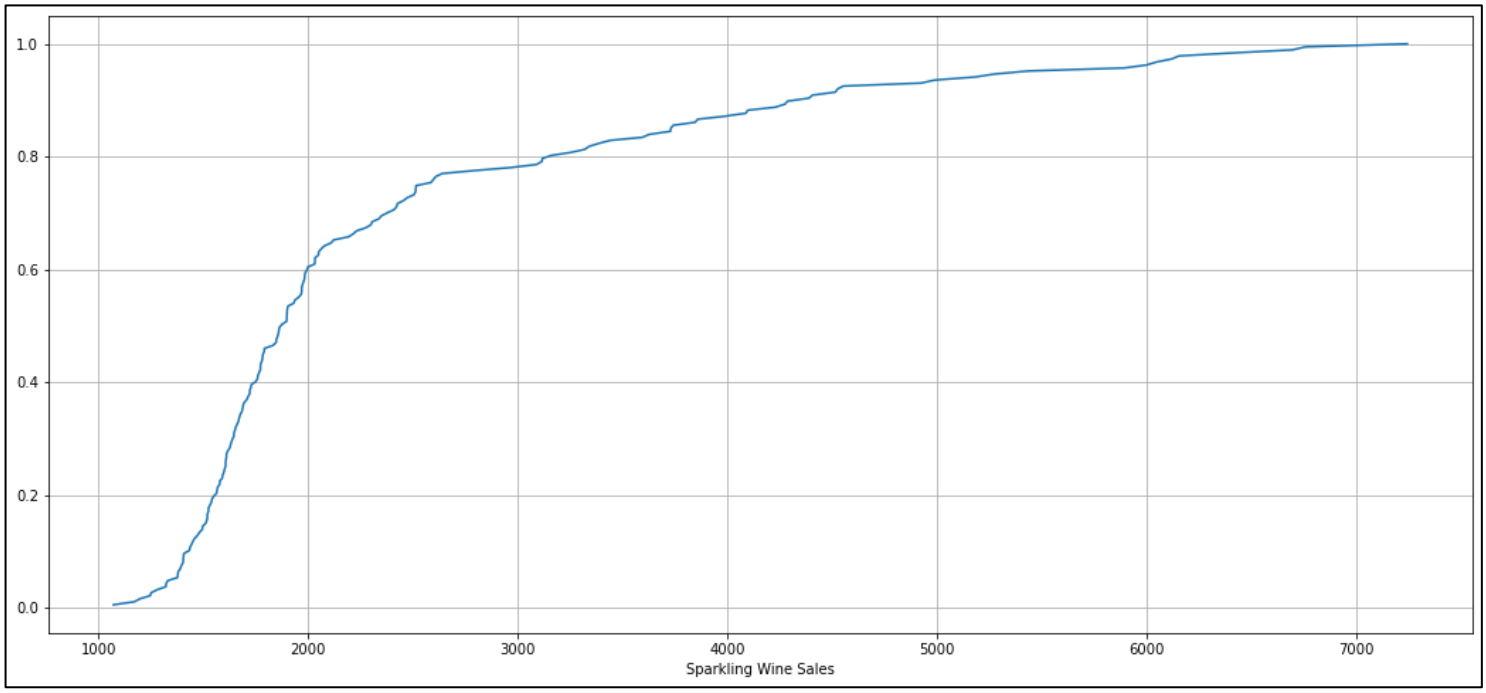


Figure 1. 9: Empirical Cumulative Distribution Plot

- The above plot shows the distributions of entire sales.
- 60% of the sales lying between 1600 to 3100. 20% values are lying above 3500. 20% of values are lying below 1500.

1.2.4 Decomposition of the Data

- A time series is composed of trend, seasonality and residuals. To analyse these components separately, the time series can be decomposed.
- There are 2 types of decomposition models:
    - Additive model – Useful when the seasonality variation is relatively constant over time.
    - Multiplicative model – Useful when the magnitude of seasonality is varying over time.
- Here is the visual representation of decomposed time series using **additive model**:

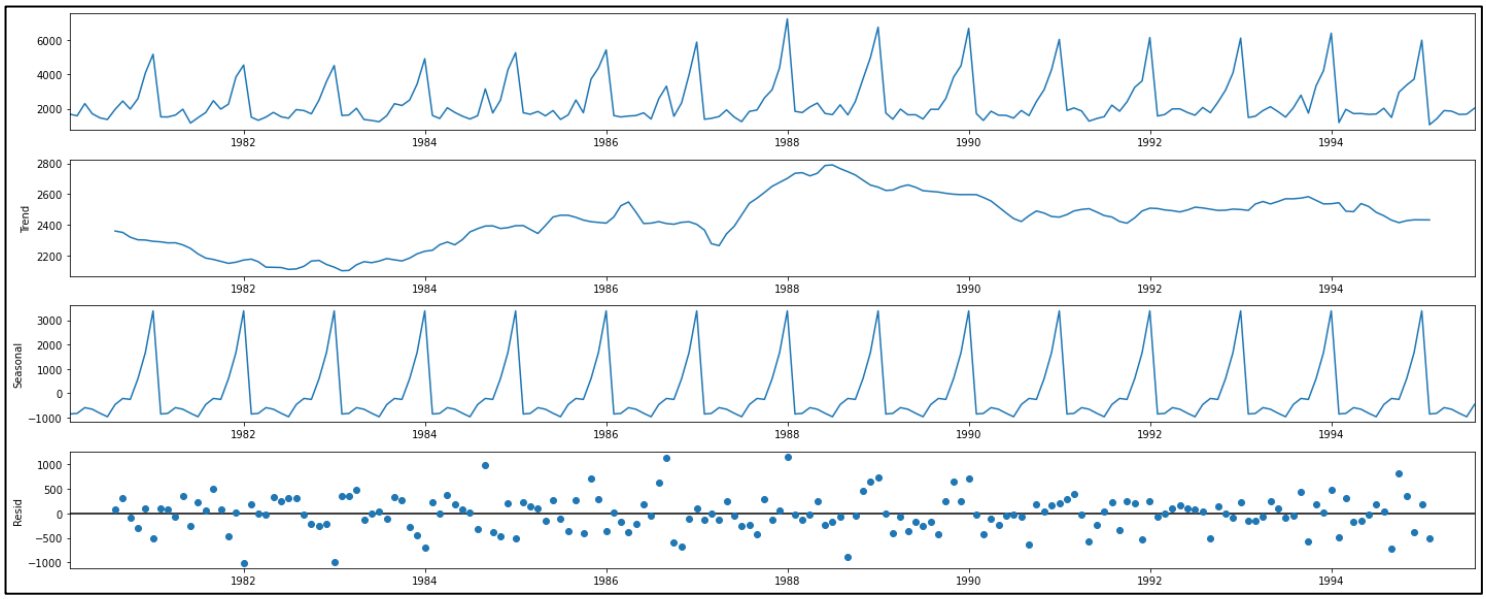


Figure 1. 10: Decomposition Plot - Additive

- First panel has the original data.
- Second panel shows the trend of the data. In this case, the trend is of mix nature. Starts with decreasing, then increasing and then stagnant with some spikes in between.
- Third panel shows seasonality. We can see that every year we have a repeated pattern.
- Forth panel has the errors. The errors/ residuals are showing some patterns.
- The errors are ranging from -1000 to +1000.
- Here is the visual representation of decomposed time series using **Multiplicative model**:

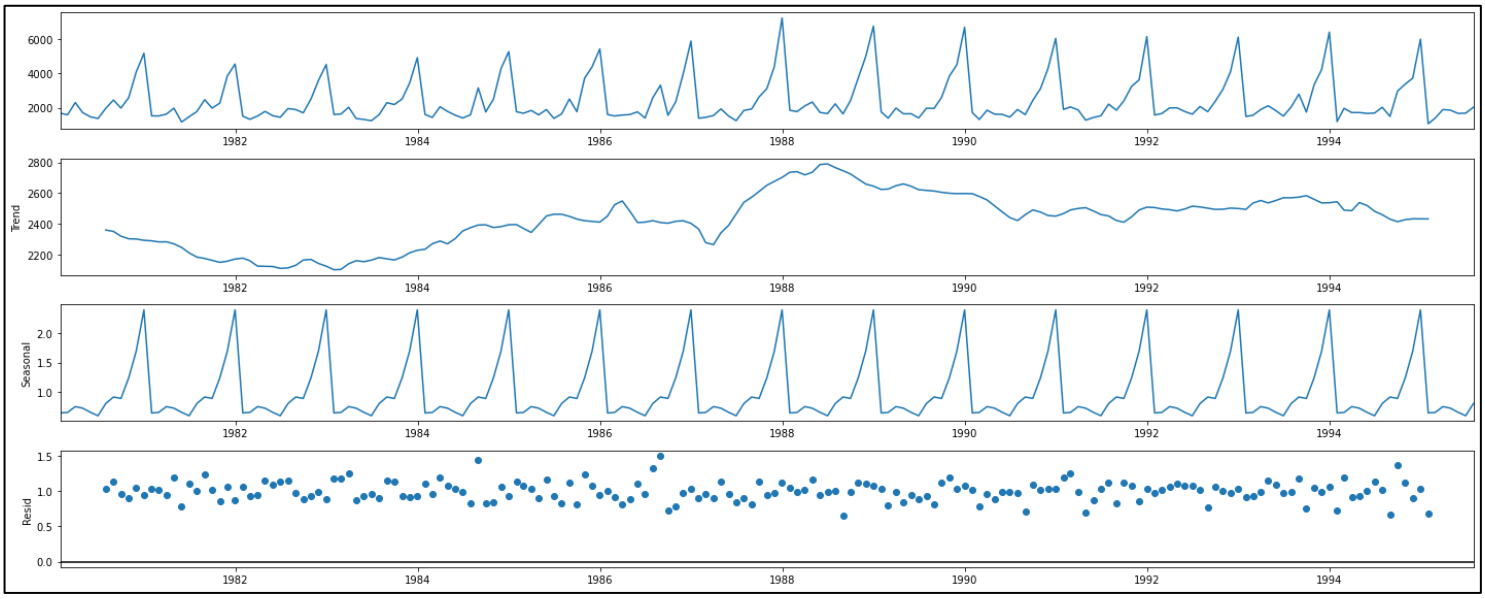


Figure 1. 11: Decomposition Plot - Multiplicative

- For the multiplicative series, we see that a lot of residuals are located around 1 and not showing outliers effect as seen in additive series.
- If we decompose a multiplicative time series using additive model, the errors continue to bear the elements of seasonality.
- In this case a multiplicative decomposition is the better choice, as the errors don't have seasonal element.

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

- In order to run multiple analytics models, we have split the data into train and test sets. Train set will be used to build the model on; and model performance can be evaluated on the test set.
- The time series has been split into train and test sets. Train set has data from 1980 till 1990 (132 entries) and test set has data starting from 1991 till 1995 (55 entries).
- This is how the split time series looks:

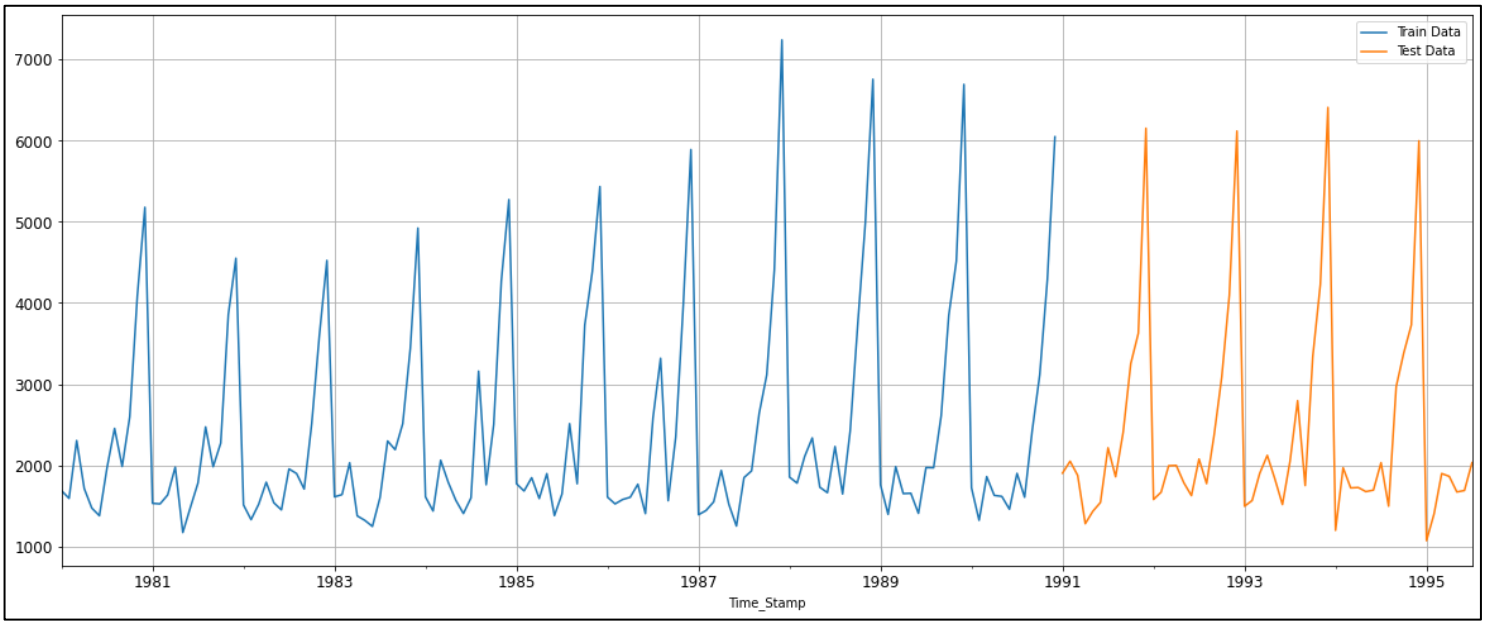


Figure 1. 12: Split Time Series

- Please note that the break between the blue and orange line is not missing data. The break is because two different sets are plotted together and test set is not picking up where train set stopped, rather it is starting from the first sales value in the test data.

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

1.4.1 Linear Regression Model

For this particular linear regression, we have regressed the 'Sparkling' (wine sale) variable against the order of the occurrence. For this we modified the training data before fitting it into a linear regression. We have generated a numerical time instance order for both the training and test set and added these values in the respective sets. This is how the sample of data looks like for linear regression:

First few rows of Training Data

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

Last few rows of Training Data

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

First few rows of Test Data

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

Last few rows of Test Data

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

Now that our training and test data has been modified, let us go ahead use Linear Regression to build the model on the training data and evaluate the model on the test data using Root Mean Square Error (RMSE). The lesser the RMSE, the better the model performs.

For Linear Regression forecast on the train data, **RMSE is 1389.135**. This is how the forecast appear against actual values:

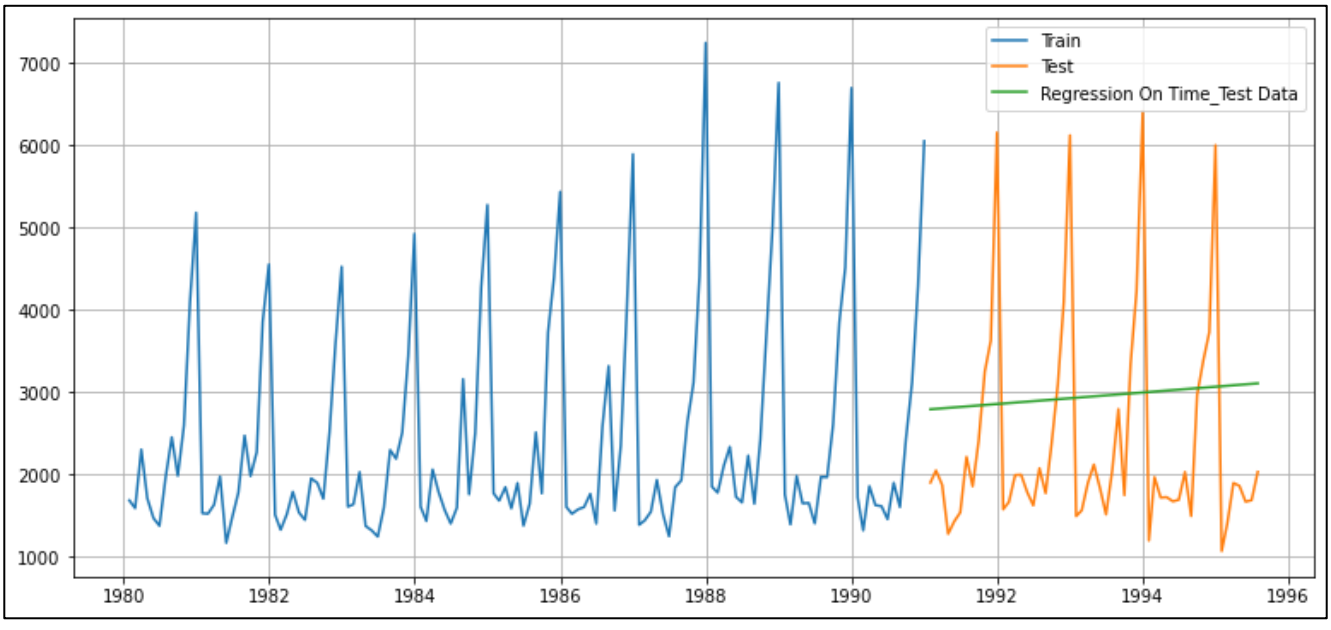


Figure 1. 13: Forecast using Linear Regression

- The training data is represented by the blue line, test data is represented by orange line.

- The green line in the graph represents the forecast made using linear regression.
- As we can see that this forecast does capture the trend of the data, but seasonality component remains missing. This indicates that linear regression is not fit for the data with seasonality, as it gives only a best fit line as an outcome.

1.4.2 Naïve Forecast Model

For this particular naive model, we can 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.

In other words, naïve forecast method used the last value in the time series to forecast the future values, and keeps it constant throughout the length of the forecast. In the training dataset, the last sales value is 6047. The naïve method built the model using 6047 as the forecast value for the length of test data.

For Naïve Forecast on the train data, **RMSE is 3864.279**. This is how the forecast appear against actual values:

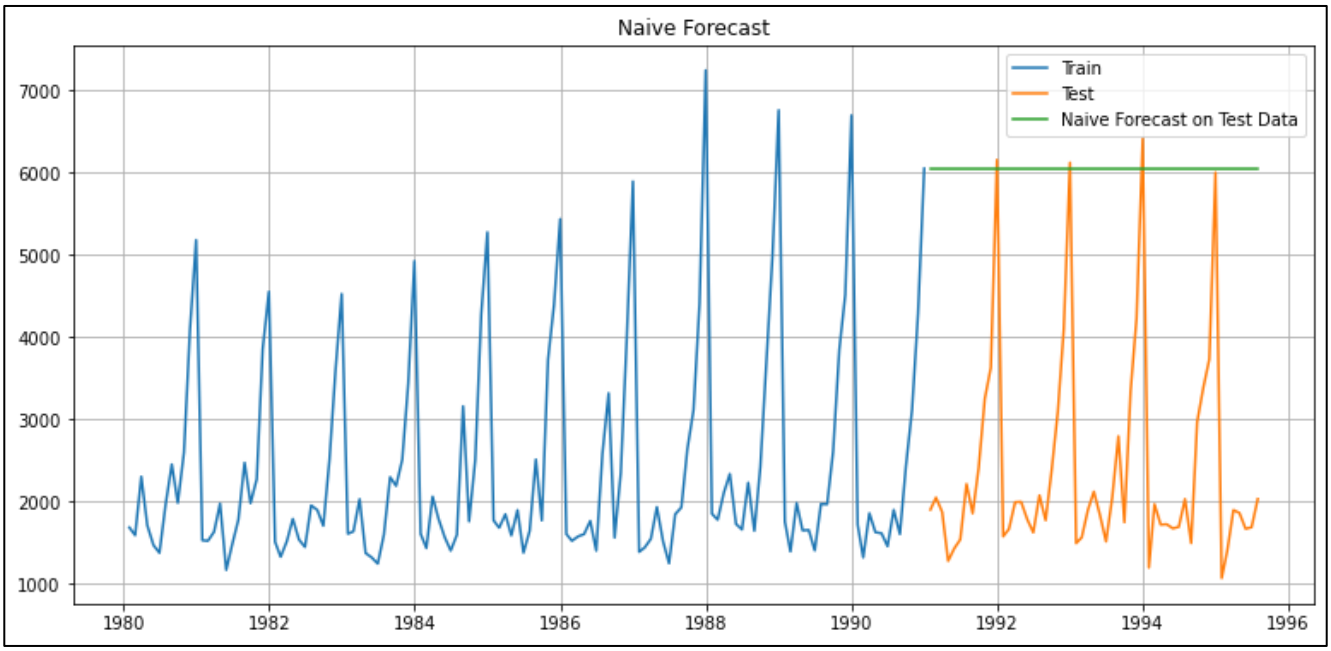


Figure 1. 14: Forecast using Naïve Model

- The RMSE value is very high than that of linear regression.
- The forecast line in green is a flat line, discounting trend and seasonality present in our data.
- This model doesn't seem to be optimum at all for forecasting wine sales.

1.4.3 Simple Average Model

For this particular simple average method, we forecasted by using the average of the training values. Since the average is impacted by outliers (abrupt changes in the time series), simple average is better than naïve forecast method where only the last value is considered for forecast.

For Simple Average Forecast on the train data, **RMSE is 1275.082**. This is how the forecast appear against actual values:

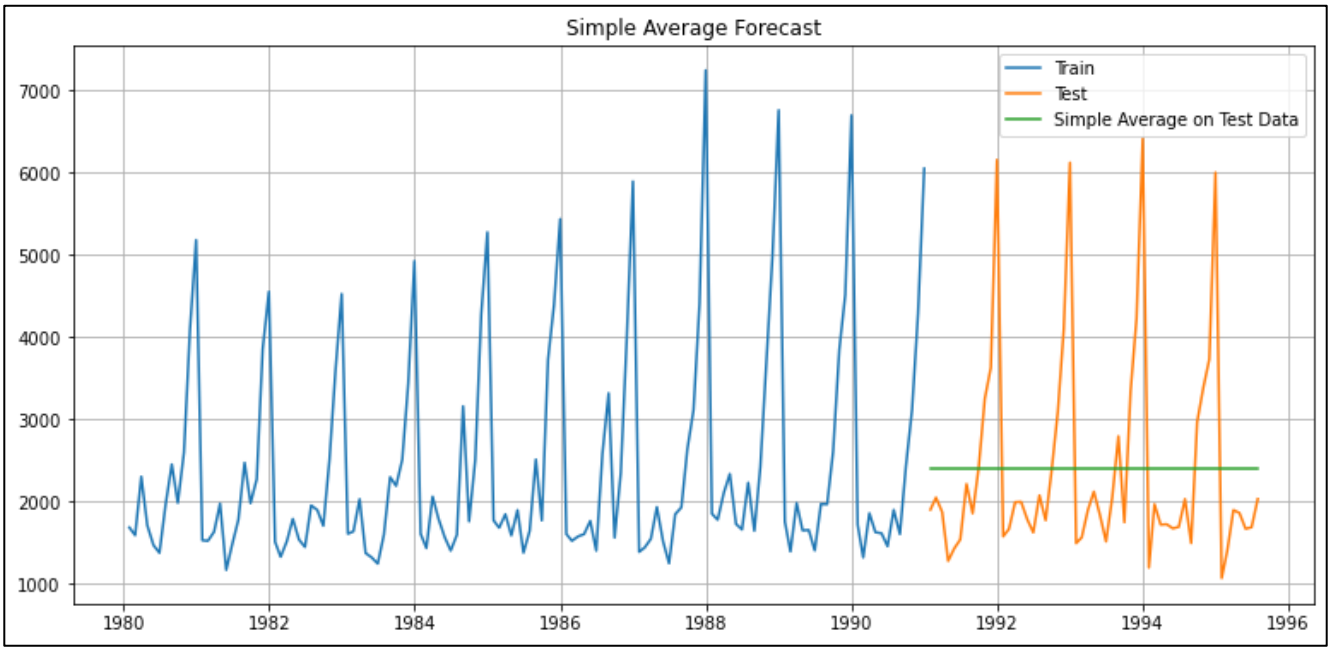


Figure 1. 15: Forecast using Simple Average

- The RMSE is the lowest among all the models so far.
- Although, the forecast line in green is again a flat line, discounting trend and seasonality present in our data.
- Since the model is using one static value of the average of training data as forecast, it doesn't seem to be optimum for our data.

1.4.4 Moving Average Model

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 (RMSE). The different intervals we are considering in this case are:

A. 2-point trailing moving average – average of 2 months data
B. 4-point trailing moving average – average of 4 months data
C. 6-point trailing moving average – average of 6 months data
D. 9-point trailing moving average – average of 9 months data

For Moving Average, we are going to average over the entire data. And later split the data into train and test after running the model.

This is how the moving average appear against the original data:

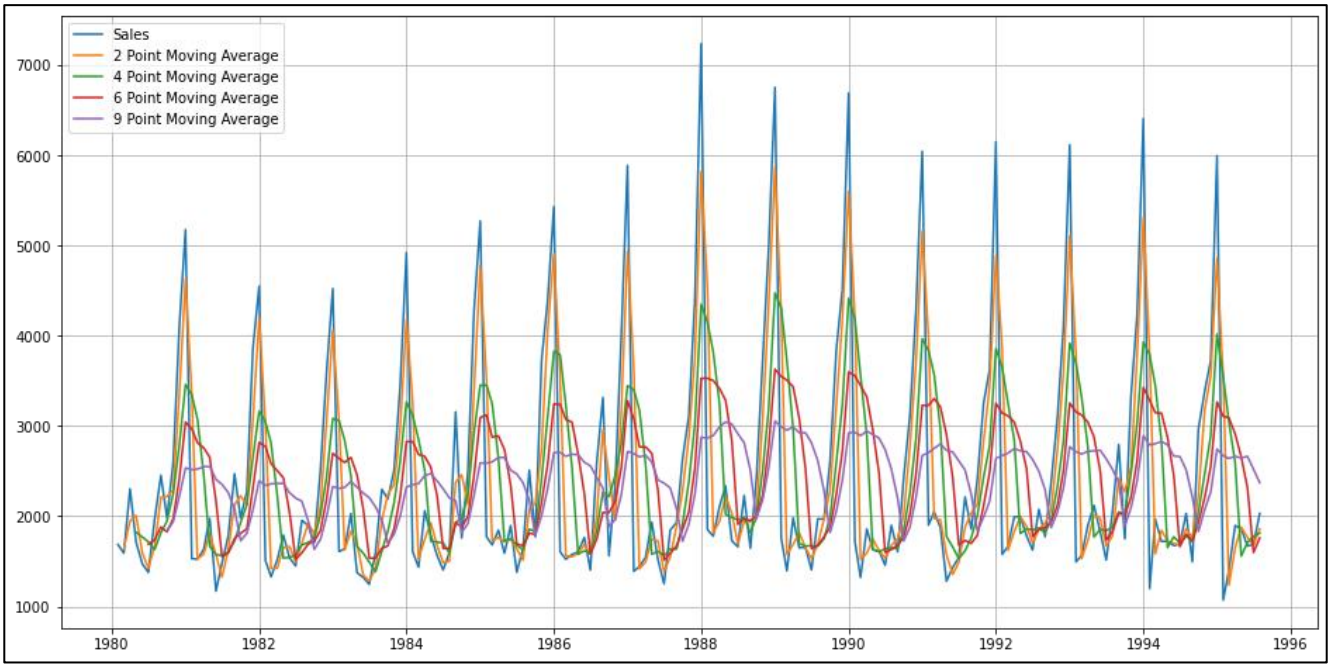


Figure 1. 16: Forecast using Moving Average

After splitting the data into train (1980 – 1990) and test (1991 – 1995), plotting this Time Series:

2-point trailing moving average

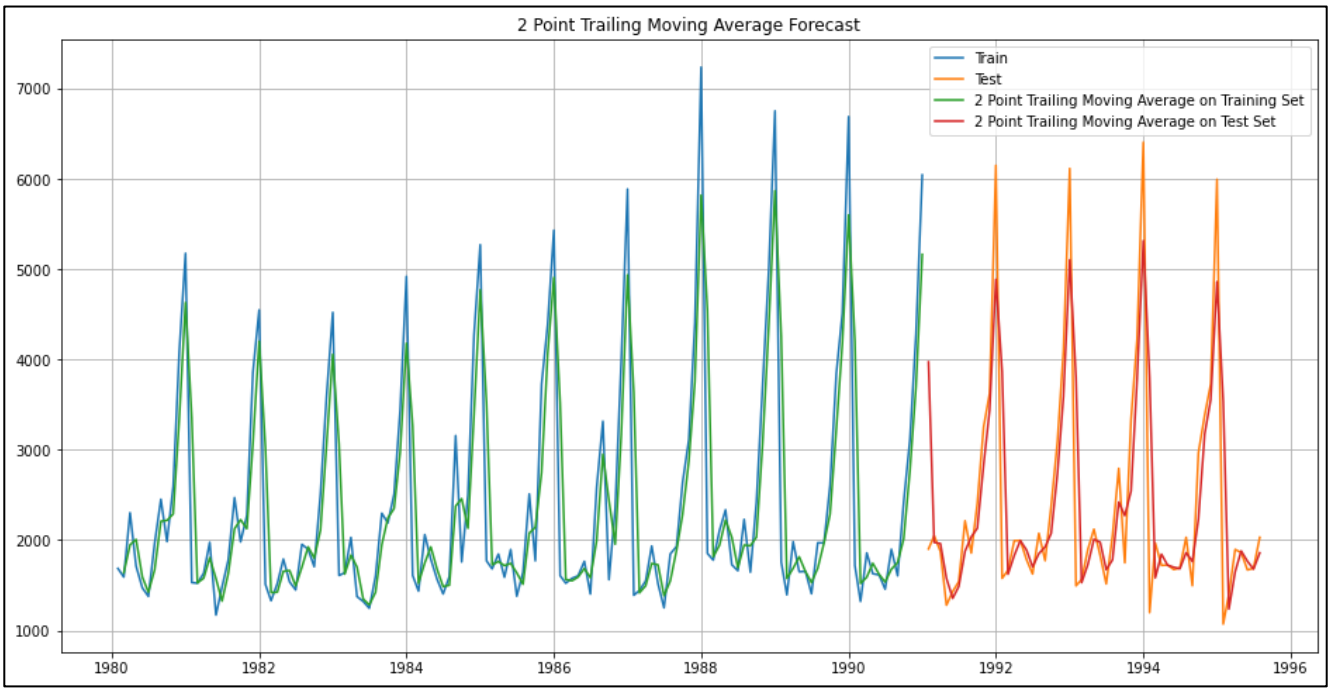


Figure 1. 17: Forecast using 2 Point Moving Average

4-point trailing moving average

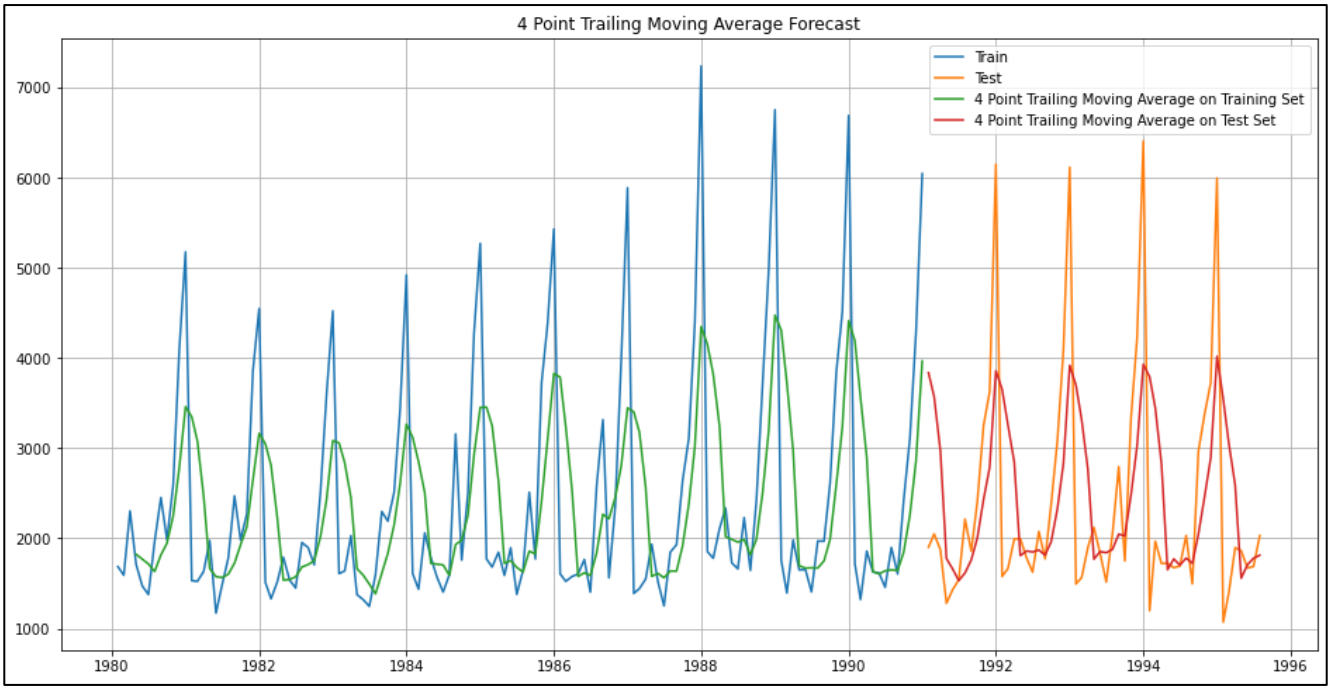


Figure 1. 18: Forecast using 4 Point Moving Average

6-point trailing moving average

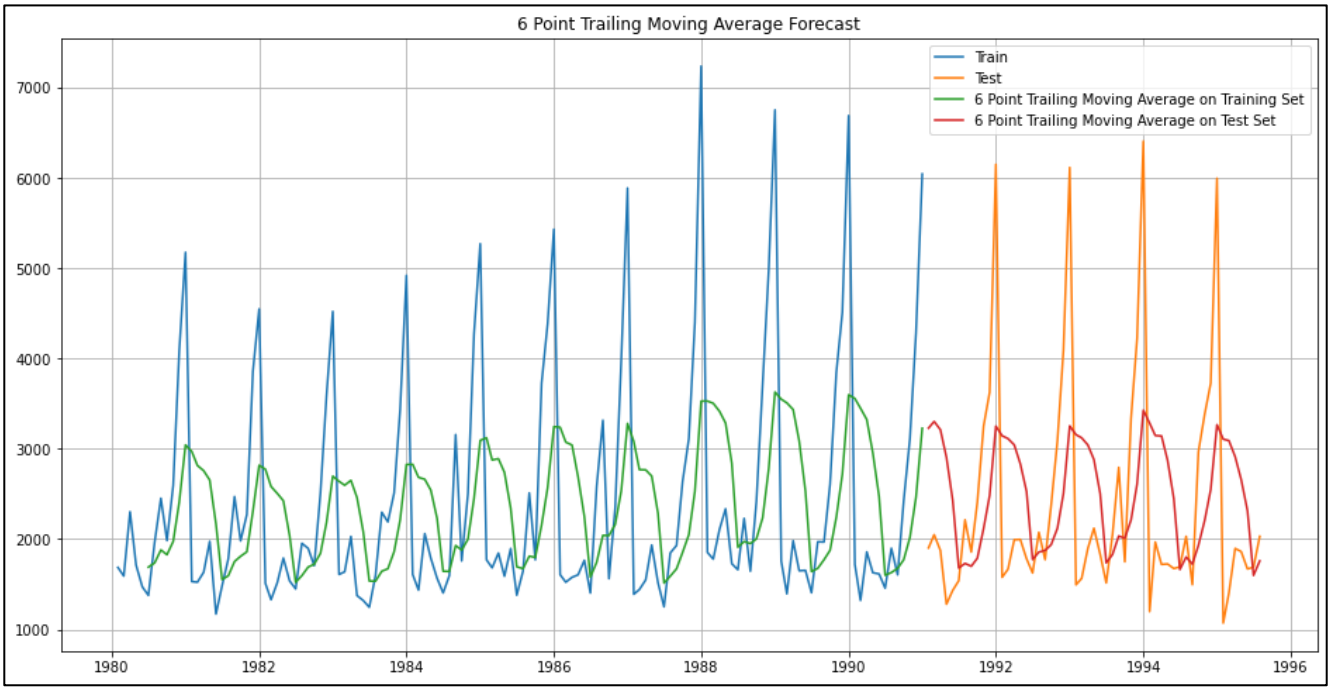


Figure 1. 19: Forecast using 6 Point Moving Average

9-point trailing moving average

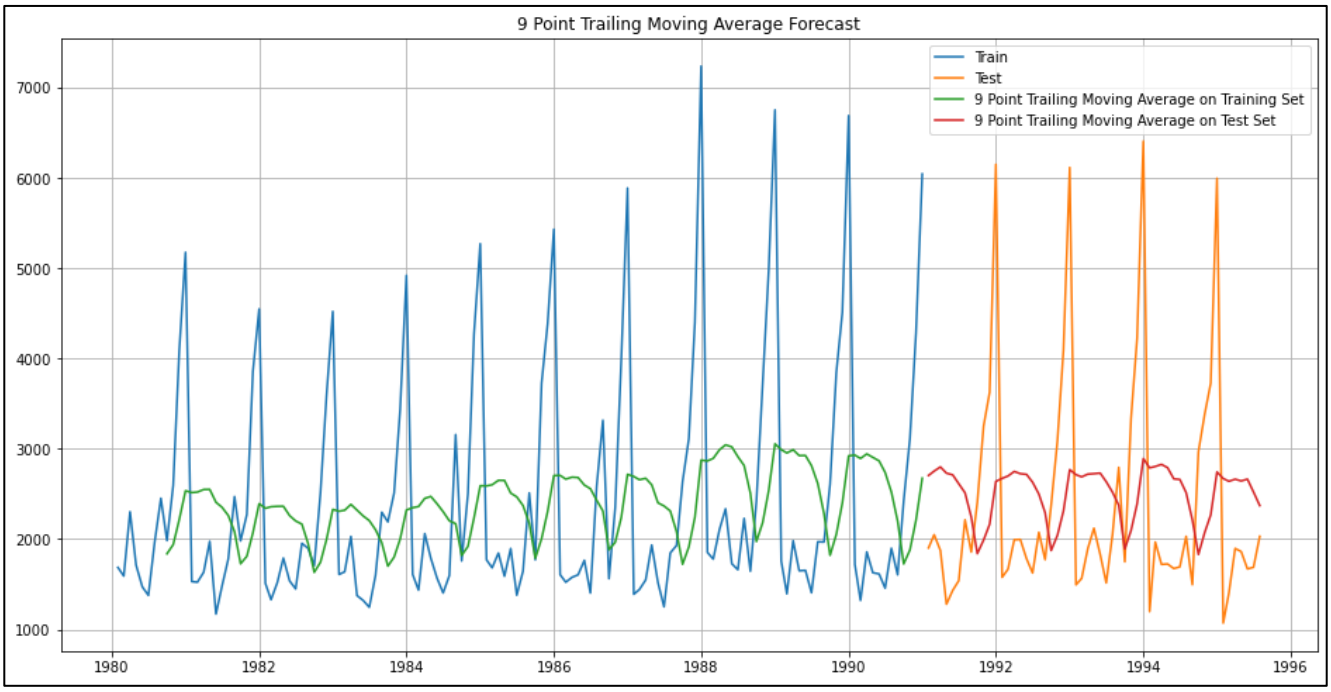


Figure 1. 20: Forecast using 9 Point Moving Average

For Moving Average Forecast on the train data various RMSE values are as follows:

- 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**
- For 6 point Moving Average Model forecast on the Training Data, **RMSE is 1283.927**
- For 9 point Moving Average Model forecast on the Training Data, **RMSE is 1346.278**

- The RMSE is the lowest for 2-point trailing moving average.
- From the graphs also we can observe that 2-point trailing moving average forecast is quite closer to our test data. Trend and seasonality is also captured well by this model.
- Hence, 2-point trailing moving average model is the best model we have so far for forecasting sparkling wine sales.

Exponential Smoothing Methods

- Exponential smoothing method considers the weighted averages of the past observations.
- There are 3 parameters; out of these 3, one or more parameters control the weighted averages.
    1. Alpha - Level of the time series
    2. Beta – Trend of the time series
    3. Gamma – Seasonality of the time series

1.4.5 Simple Exponential Smoothing Model

Simple exponential smoothing method for forecasting only works when there is no trend or seasonality in the time series. It only accounts for the Alpha (level) of the time series.

Since we have trend and seasonality in our time series, this model is right away not very useful. But let's build and compare the outcomes of this model with the rest of the models.

For Alpha = 0.07 Simple exponential smoothing Forecast on the train data, **RMSE is 1338.012**. This is how the forecast appear against actual values:

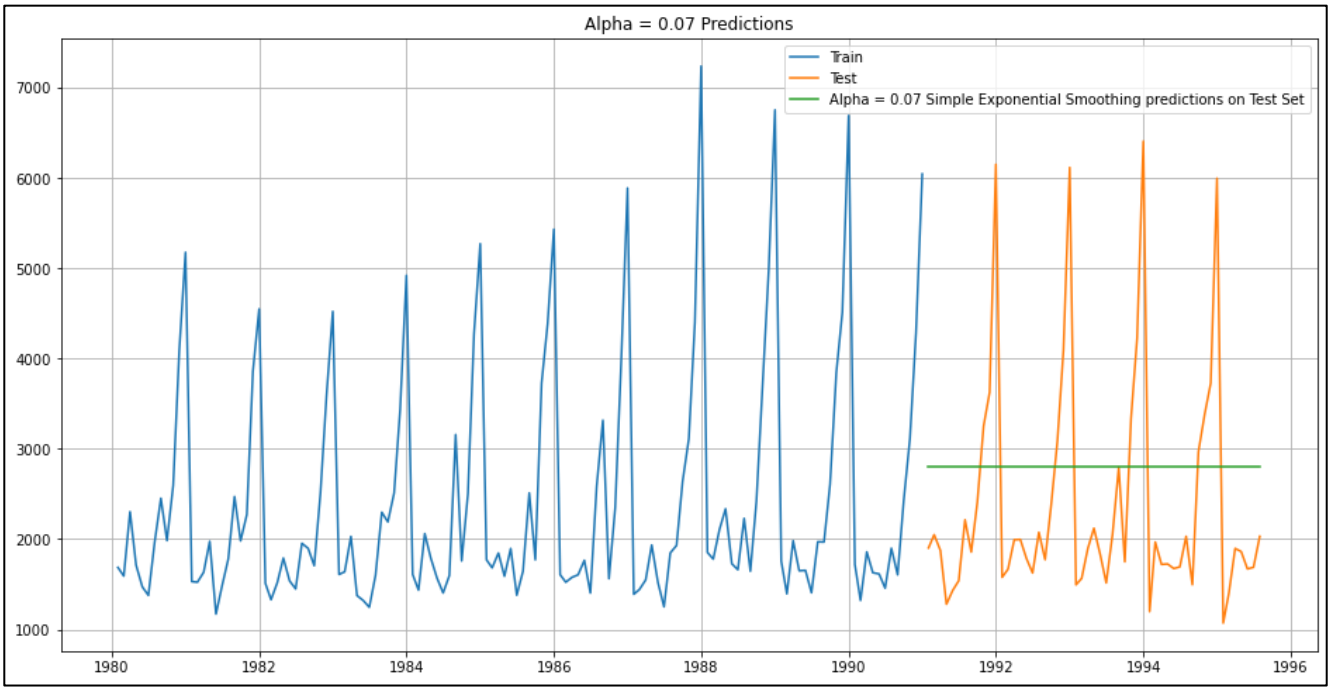


Figure 1. 21: Simple Exponential Smoothing – Alpha 0.07

Setting different alpha values:

- We ran a loop with different alpha values to understand which particular value works best for alpha on the test set.
- The higher the alpha value more weightage is given to the more recent observation. That means, what happened recently will happen again.
- Alpha value of 0.02 returns the lowest **RMSE of 1278.497**. So, 0.02 is the optimum alpha value in terms of SES model. Alpha = 0.02 (close to 0) interprets that forecasts are far from the actual data points.
- This is how the forecast appear with Alpha as 0.07 and 0.02, against the actual values:

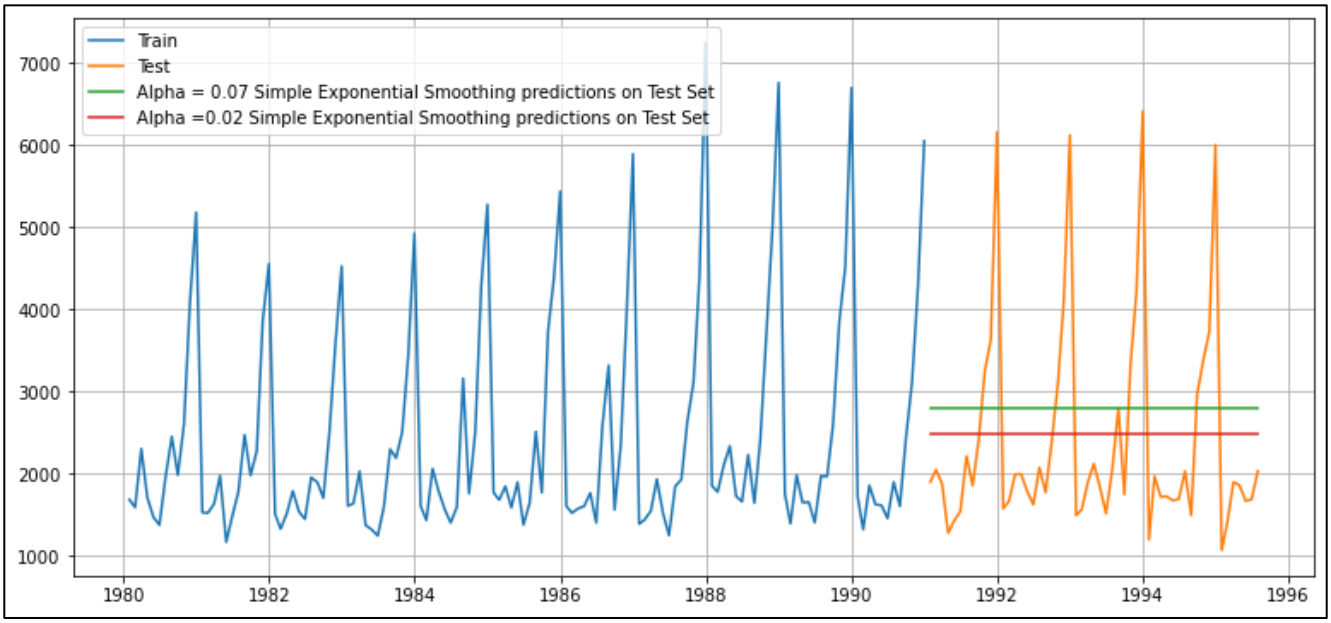


Figure 1. 22: Simple Exponential Smoothing – Alpha 0.07 & 0.02

- The RMSE of Simple Exponential Smoothing method at Alpha = 0.02 is low as compared to most of the models we applied so far.
- As we can see from the graph, this method is returning a static forecast for different Alpha values. That is because it does not factor trend and seasonality. Hence, it is not optimum for our data.

1.4.6 Double Exponential Smoothing Model

Double exponential smoothing method for forecasting works when there is no seasonality in the time series. It only accounts for the Alpha (level) and Beta (trend) of the time series.

Since we have seasonality also present in our time series, this model may not be very useful. But let's build and compare the outcomes of this model with the rest of the models.

For Alpha = 0.66 and Beta = 9.966 e-05 (0.00), Double exponential smoothing Forecast on the train data, **RMSE is 3949.993**. This is how the forecast appear against actual values:

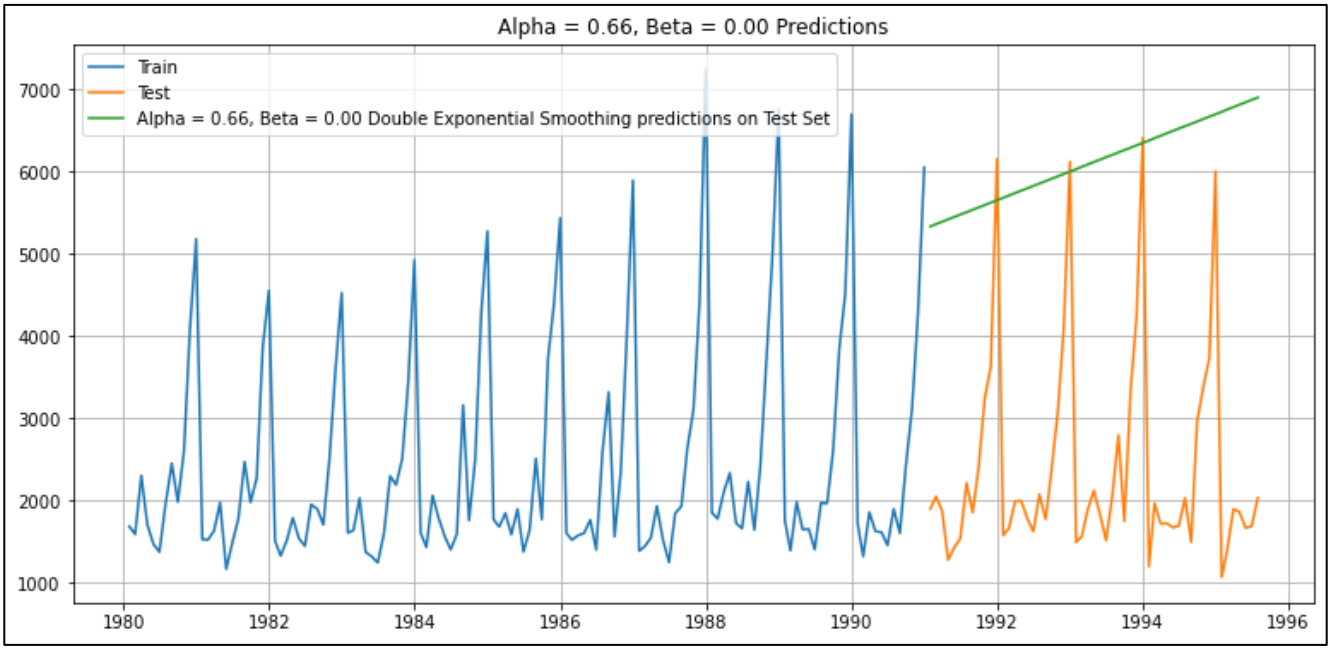


Figure 1. 23: Double Exponential Smoothing – Alpha 0.66 & Beta 0.00

Setting different alpha and beta values:

- We ran a loop with different Alpha and Beta values to understand which particular values work best for alpha and beta on the test set.
- Alpha = 0.1 and Beta = 0.1 return a lower **RMSE of 1777.735**. So, 0.1 is the optimum Alpha and Beta values in terms of DES model.
    - Alpha = 0.1 (close to 0) interprets that forecast is far from the actual data points.
    - Beta = 0.1 (close to 0) interprets that forecast is giving more weightage to older trend.
- This is how the forecast appear with Alpha as 0.1 and Beta as 0.1, against the actual values:

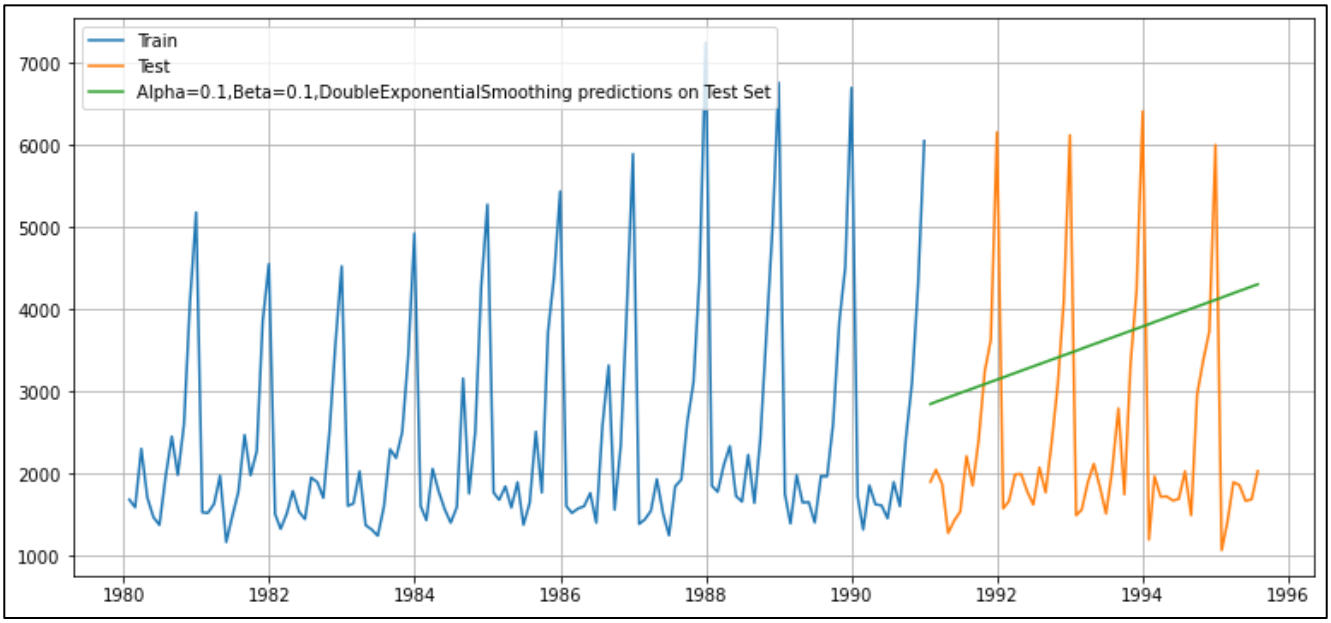


Figure 1. 24: Double Exponential Smoothing – Alpha 0.1 & Beta 0.1

- The RMSE of Double Exponential Smoothing method at optimum Alpha and Beta value of 0.1 is one of the highest among all the models we have applied so far.
- As we can see from the graph, this method is taking trend into consideration but discounting seasonality, which is very crucial in our time series.
- Hence, Double Exponential Smoothing model is not optimum for our data.

1.4.7 Triple Exponential Smoothing Model

Triple exponential smoothing method for forecasting works when there is both, trend and seasonality, present in the time series. It accounts for the Alpha (level), Beta (trend) and Gamma (seasonality) of the time series.

This model can be very useful for our trend and seasonality laced time series. Let's build the model and compare if it works better than the rest of the models.

Since we have seasonality component involved, the model is built using 2 techniques:

1. Additive seasonality
2. Multiplicative seasonality

Additive seasonality

The model is fit using the brute force method to choose the best parameters automatically.

For Alpha = 0.1, Beta = 0.01 and Gamma = 0.509, Triple exponential smoothing (additive) forecast on the train data, **RMSE is 379.696**. This is how the forecast appear against actual values:

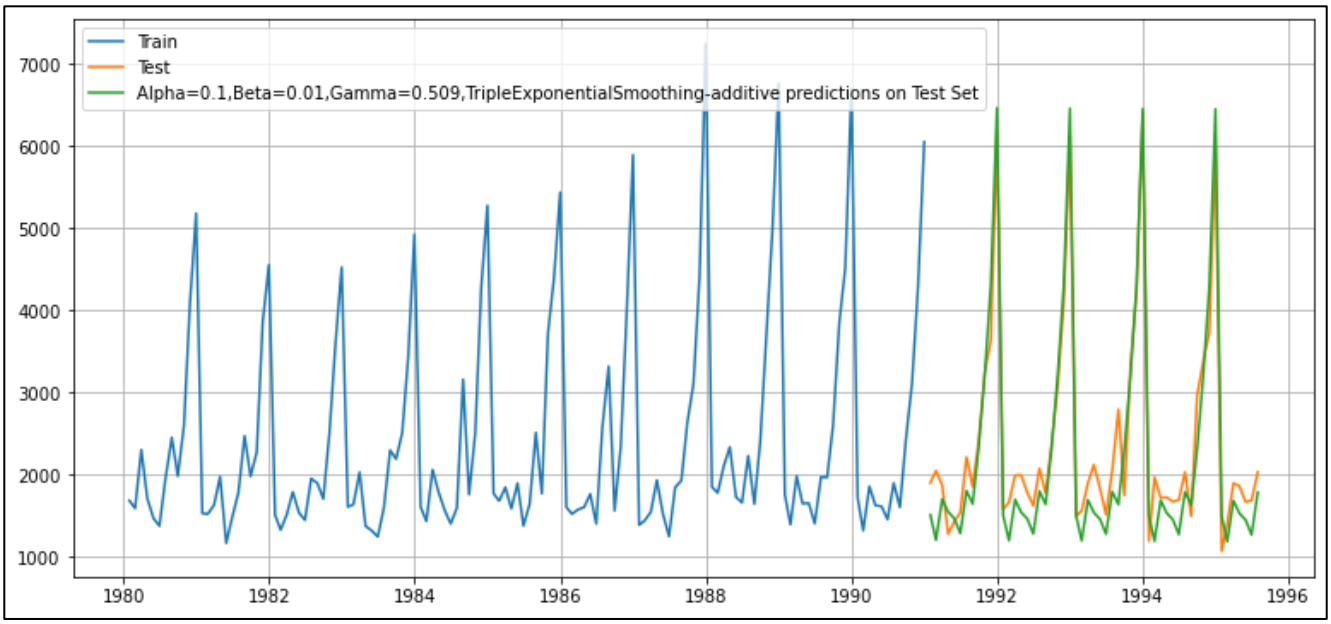


Figure 1. 25: Triple Exponential Smoothing_Additive – Alpha 0.1, Beta 0.01 & Gamma = 0.509

Setting different alpha, beta and gamma values:

- We ran a loop with different Alpha, Beta and Gamma values to understand which particular values work best for alpha, beta and gamma on the test set.
- Alpha = 0.1, Beta = 0.4 and Gamma = 0.1 return a lower **RMSE of 342.935**. So, 0.1 is the optimum Alpha & Gamma and 0.4 is the optimum Beta values in terms of TES-additive model.
    - Alpha = 0.1 (close to 0) interprets that forecast is far from the actual data points.
    - Beta = 0.4 (close to 0) interprets that forecast is giving more weightage to older trend.
    - Gamma = 0.4 (close to 0) interprets that forecast is giving more weightage to older seasonality.
- This is how the forecast appear with Alpha as 0.1, Beta as 0.4 and Gamma as 0.1, against the actual values:

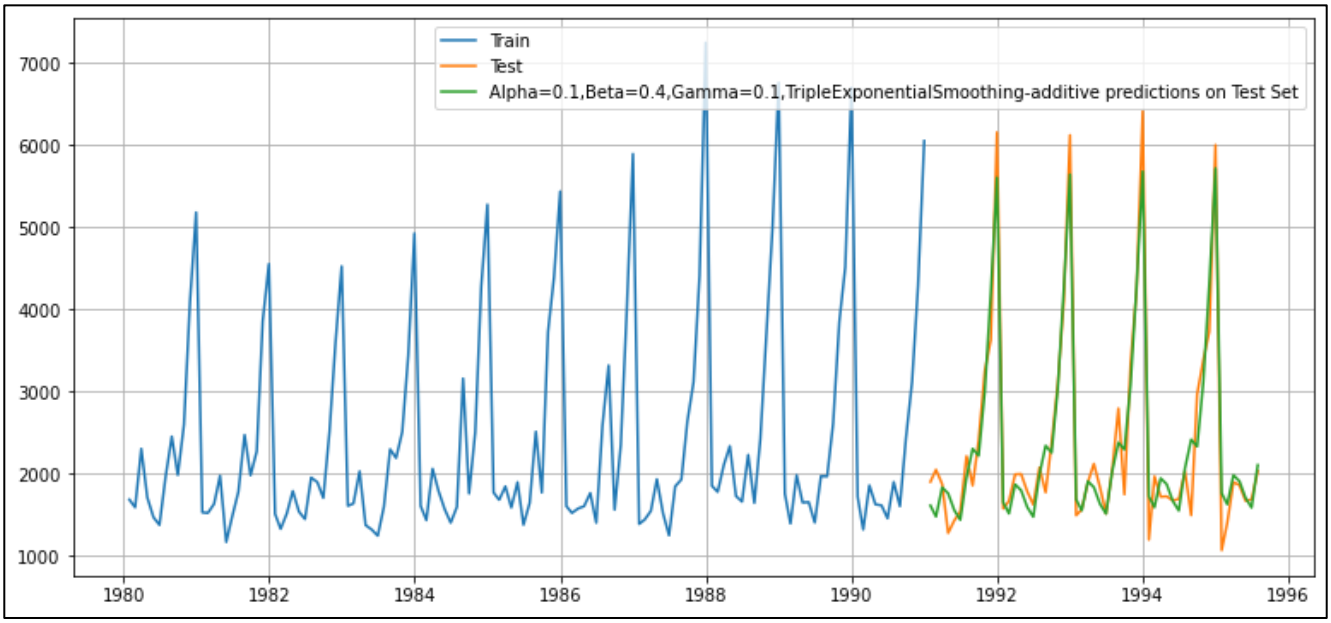


Figure 1. 26: Triple Exponential Smoothing_Additive – Alpha 0.1, Beta 0.4 & Gamma = 0.1

Multiplicative seasonality

The model is fit using the brute force method to choose the best parameters automatically.

For Alpha = 0.1, Beta = 0.049 and Gamma = 0.36, Triple exponential smoothing (multiplicative) forecast on the train data, **RMSE is 406.510**. This is how the forecast appear against actual values:

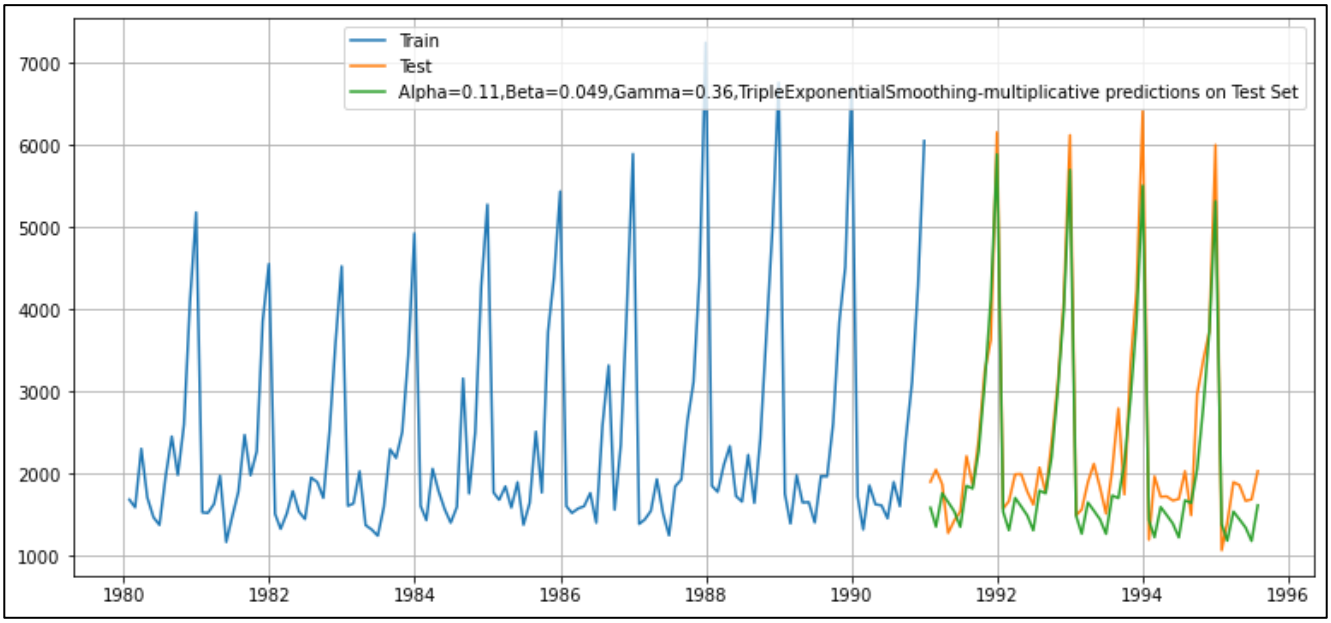


Figure 1. 27: Triple Exponential Smoothing_Multiplicative – Alpha 0.1, Beta 0.049 & Gamma = 0.36

Setting different alpha, beta and gamma values:

- We ran a loop with different Alpha, Beta and Gamma values to understand which particular values work best for alpha, beta and gamma on the test set.
- Alpha = 0.4, Beta = 0.1 and Gamma = 0.2 return the lowest **RMSE of 317.434**. So, 0.4 is the optimum Alpha, 0.1 is the optimum Beta and 0.2 is the optimum Gamma value in terms of TES-multiplicative model.

- Alpha = 0.4 (close to 0) interprets that forecast is far from the actual data points.
- Beta = 0.1 (close to 0) interprets that forecast is giving more weightage to older trend.
- Gamma = 0.2 (close to 0) interprets that forecast is giving more weightage to older seasonality.

- This is how the forecast appear with Alpha as 0.4, Beta as 0.1 and Gamma as 0.2, against the actual values:

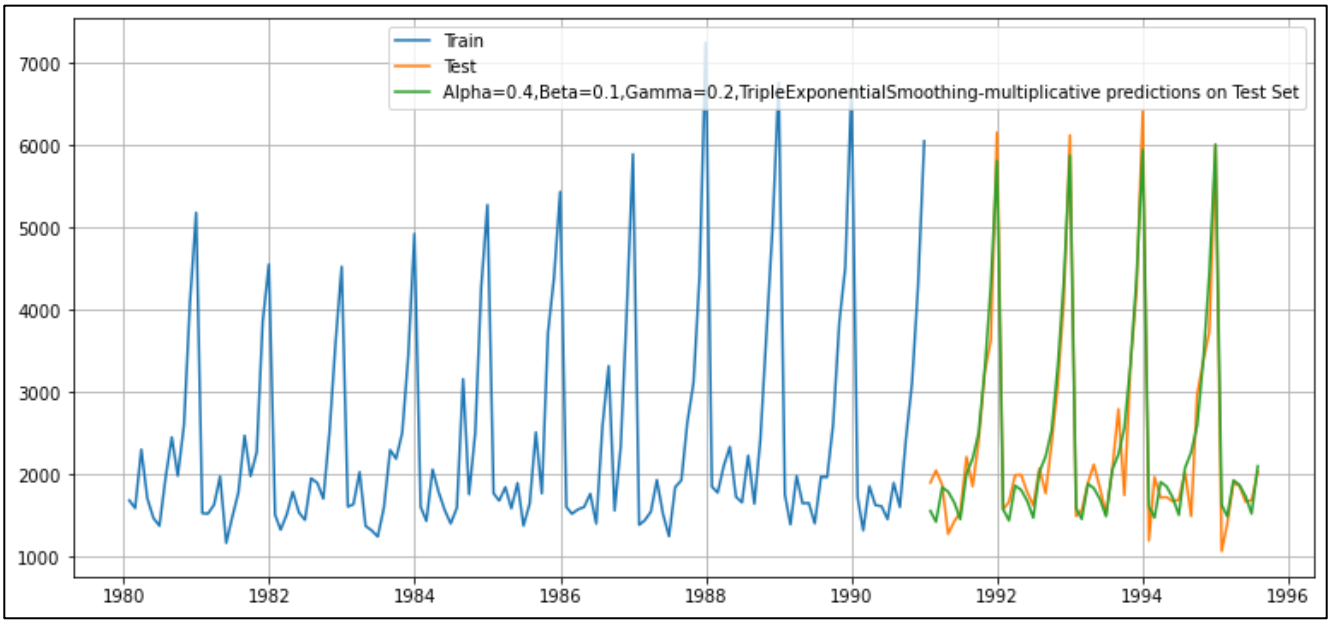
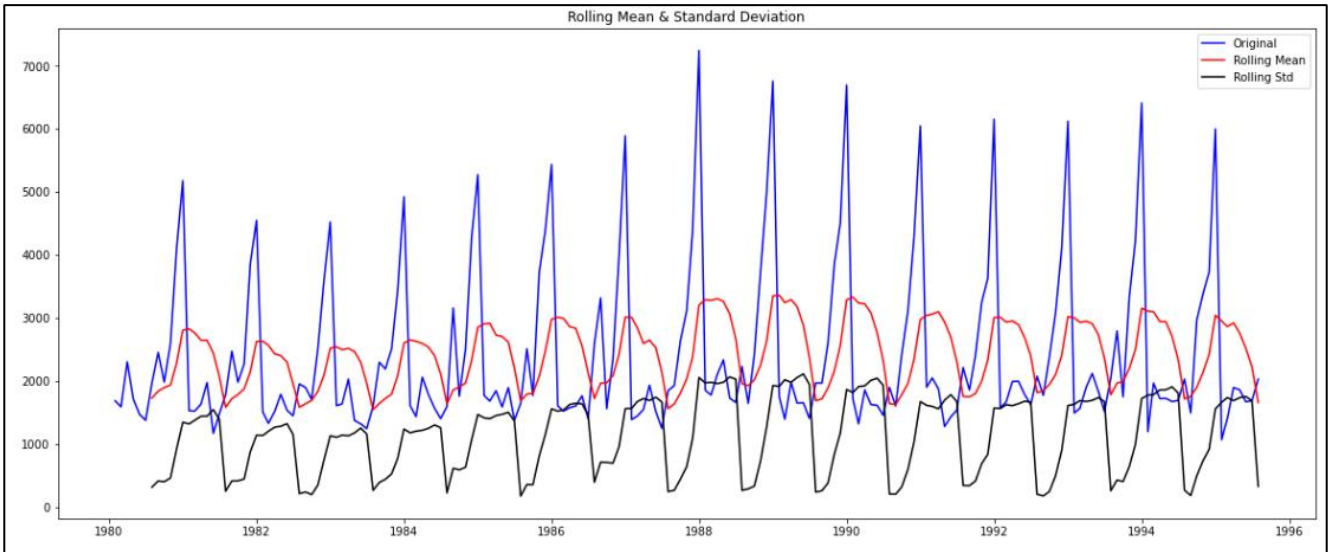


Figure 1. 28: Triple Exponential Smoothing_Multiplicative – Alpha 0.4, Beta 0.1 & Gamma = 0.2

- From the Triple Exponential Smoothing – Multiplicative model we can observe that the forecast is fairly close to the test set values.
- The RMSE is also the lowest among all the models ran so far.
- This could be the most optimum model to forecast for the sparkling wine sales date.

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

- The Augmented Dickey-Fuller (ADF) test is a unit root test which determines whether there is a unit root in the time series and subsequently whether the series is non-stationary or not.
- The hypothesis in a simple form for the ADF test is:
  - H_0: The Time Series has a unit root and is thus non-stationary.
  - H_1: The Time Series does not have a unit root and is thus stationary.
- We would want the series to be stationary for building ARIMA models and thus we would want the p-value of this test to be less than the alpha value of 0.05.
- First, we check the data stationarity on the entire time series data:

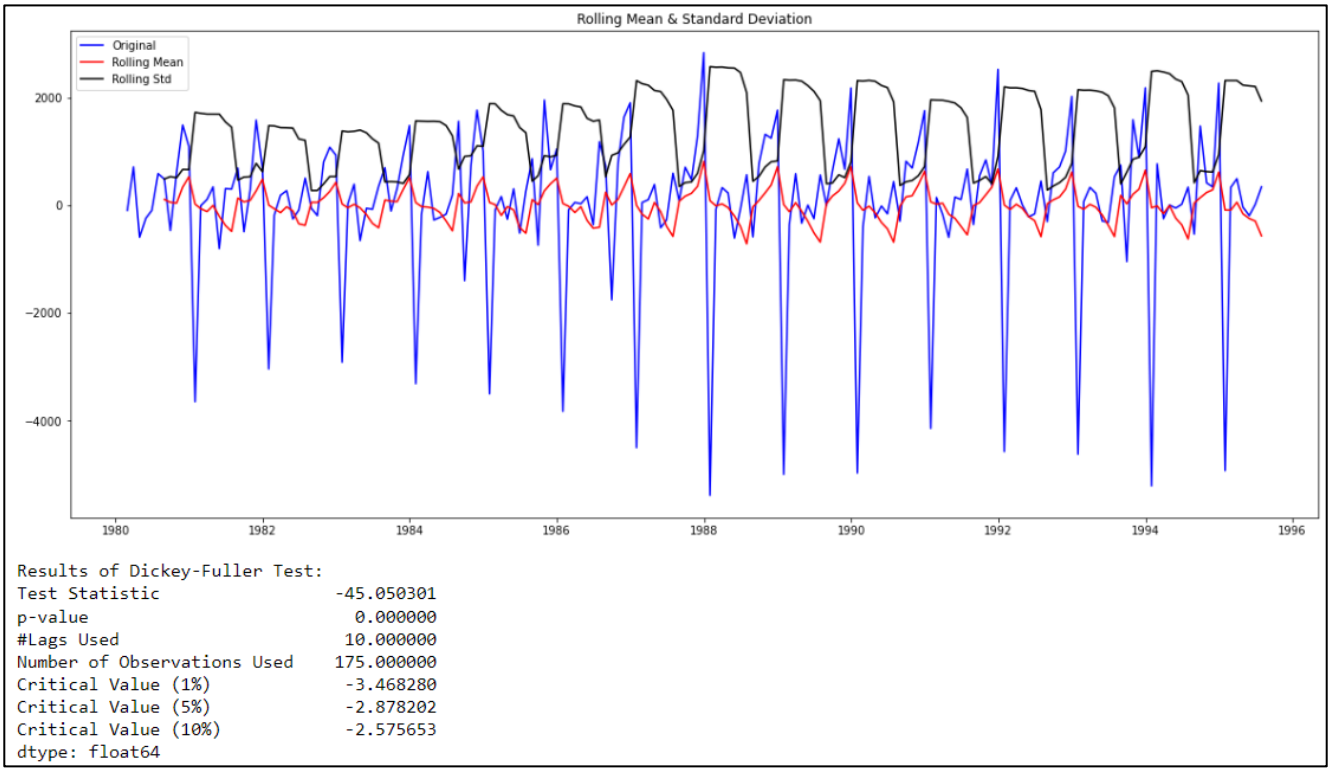



```
Results of Dickey-Fuller Test:
Test Statistic                  -1.360497
p-value                          0.601061
#Lags Used                      11.000000
Number of Observations Used    175.000000
Critical Value (1%)             -3.468280
Critical Value (5%)             -2.878202
Critical Value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

Figure 1. 29: ADF Test – Original Data

- Since p-value 0.60 > 0.05 (failed to reject H_0), we can say that at 5% significance level the time series is non-stationary.

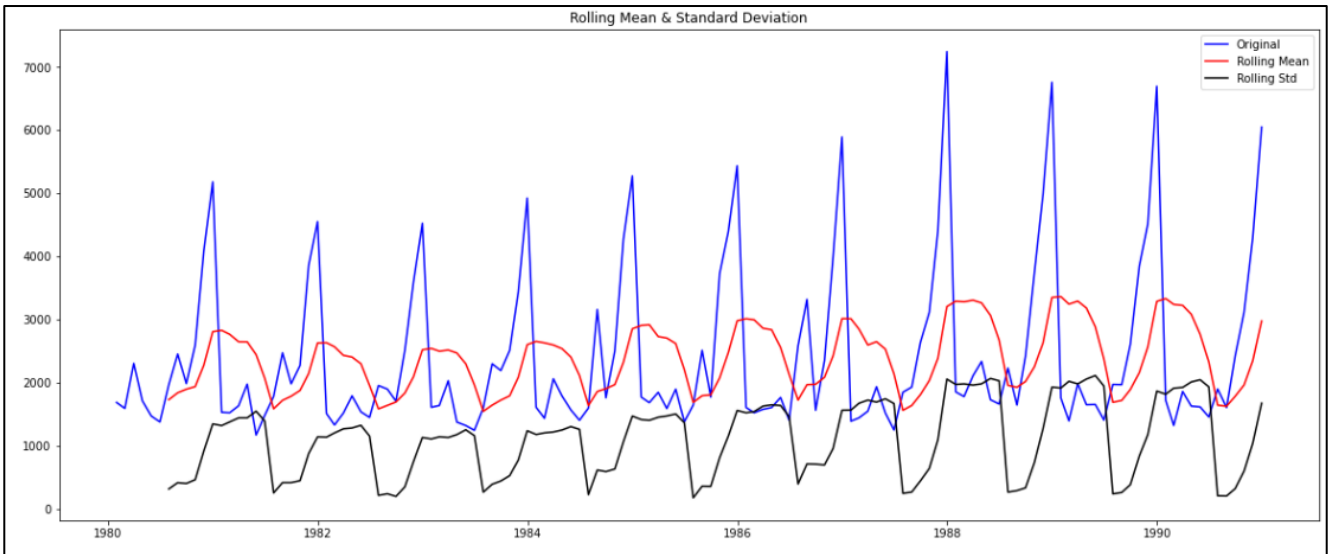
- Let us take a difference of order 1 and check whether the time series becomes stationary or not:




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

Figure 1. 30: ADF Test – Original Data with Differencing

- Now that the p-value 0.00 < 0.05, we reject the null hypothesis. Thus, the time series is now stationary.
- Now, we check the data stationarity on the train data:

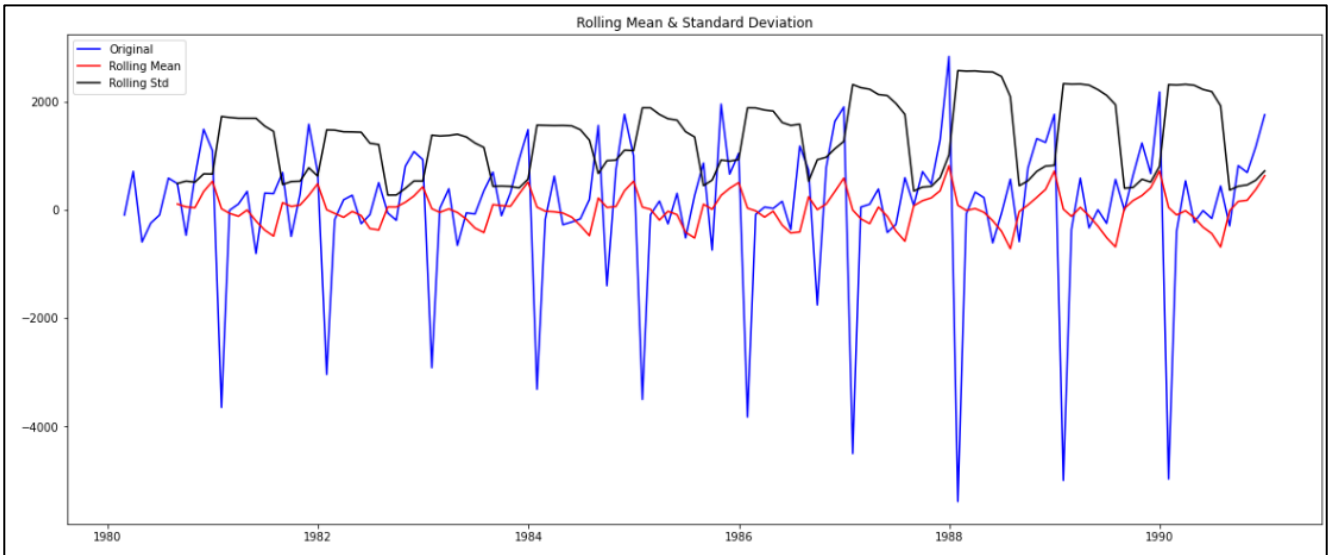



```
Results of Dickey-Fuller Test:
Test Statistic                -1.208926
p-value                         0.669744
#Lags Used                     12.000000
Number of Observations Used   119.000000
Critical Value (1%)            -3.486535
Critical Value (5%)            -2.886151
Critical Value (10%)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

Figure 1. 31: ADF Test – Train Data

- Since p-value 0.66 > 0.05 (failed to reject H_0), we can say that at 5% significance level the training time series is non-stationary.

- Let us take a difference of order 1 and check whether the time series becomes stationary or not:




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

Figure 1. 32: ADF Test – Training Data with Differencing

- Now that the p-value 2.28e-12 < 0.05, we reject the null hypothesis. Thus, the training time series is now stationary as well.

Note: If the series is non-stationary, stationarize the Time Series by taking a difference of the Time Series. Then we can use this particular differenced series to train the ARIMA models. We do not need to worry about stationarity for the Test Data because we are not building any models on the Test Data, we are only evaluating our models over there.

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

1.6.1 Automated ARIMA Model - ARIMA(p,d,q)

- ARIMA – Auto Regressive Integrated Moving Average.
- One of the fundamental assumptions of ARIMA model is that the time series is supposed to be stationary, meaning no having no trend.
- As we checked using ADF test in the previous segment, our time series was non-stationary. But it converted to be stationary at 1 level of differencing. Hence, we can use train data with 1 level of differencing to run ARIMA model.
- An ARIMA model consists of the Auto Regressive (AR) part and the Moving Average (MA) part, after we have made the time series stationary.
- The AR is also denoted as 'q'; MA is also denoted as 'p'; and differencing is also denoted as 'd'.
- To find the minimum Akaike Information Criteria (AIC) value we ran different combination of 'pdq':
    - Where, value of 'p' and 'q' ranges from 0 to 3.
    - 'd' = 1 remains constant as we made the time series stationary at 1 level of differencing.
- As we can see from the below table, the lowest AIC value is achieved at (p,d,q) as (2,1,2).

	param	AIC
10	(2, 1, 2)	2213.509217
15	(3, 1, 3)	2221.451977
14	(3, 1, 2)	2230.757294
11	(2, 1, 3)	2232.983058
9	(2, 1, 1)	2233.777626
3	(0, 1, 3)	2233.994858
2	(0, 1, 2)	2234.408323
6	(1, 1, 2)	2234.5272
13	(3, 1, 1)	2235.498987
7	(1, 1, 3)	2235.60781
5	(1, 1, 1)	2235.755095
12	(3, 1, 0)	2257.723379
8	(2, 1, 0)	2260.365744
1	(0, 1, 1)	2263.060016
4	(1, 1, 0)	2266.608539
0	(0, 1, 0)	2267.663036

Table 1. 3: AIC - ARIMA

- Here is the ARIMA model with automated values of (p,d,q):

SARIMAX Results

Dep. Variable:	Sparkling	No. Observations:	132
Model:	ARIMA(2, 1, 2)	Log Likelihood	-1101.755
Date:	Tue, 13 Dec 2022	AIC	2213.509
Time:	14:56:50	BIC	2227.885
Sample:	01-31-1980	HQIC	2219.351
	- 12-31-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	1.3121	0.046	28.786	0.000	1.223	1.401
ar.L2	-0.5593	0.072	-7.731	0.000	-0.701	-0.417
ma.L1	-1.9916	0.110	-18.184	0.000	-2.206	-1.777
ma.L2	0.9999	0.110	9.093	0.000	0.784	1.215
sigma2	1.099e+06	2e-07	5.49e+12	0.000	1.1e+06	1.1e+06

Ljung-Box (L1) (Q):	0.19	Jarque-Bera (JB):	14.46
Prob(Q):	0.67	Prob(JB):	0.00
Heteroskedasticity (H):	2.43	Skew:	0.61
Prob(H) (two-sided):	0.00	Kurtosis:	4.08

- From this model we can infer that, we are making more errors in MA version, since the value of 'str err' is more for MA.
- As per the p-value, that is less than 0.05 for all levels of AR and MA, we cannot identify the significance.
- MA L1 is the most significant and AR L2 is the least significant, as per the value of 'coef'.
- Predict on the Test Set using this model and evaluate the forecast. For Automated_ARIMA(2,1,2) forecast on the train data, **RMSE is 1299.980**. This is how the forecast appear against actual values:

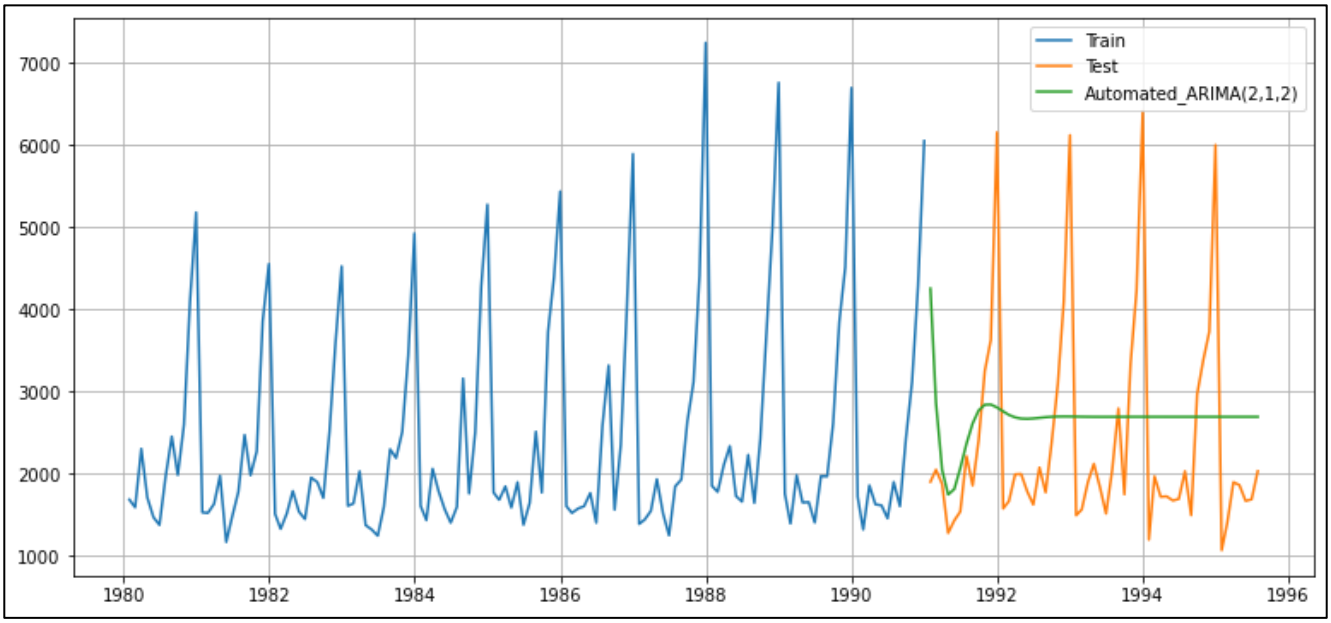


Figure 1. 33: Automated_ARIMA(2,1,2)

- From the Automated_ARIMA model we can observe that the forecast is not considering seasonality. And trend components is also quite constant towards the end.
- The RMSE is also not the lowest among all the models ran so far.
- This not optimum model to forecast for the sparkling wine sales date.

1.6.2 Automated SARIMA Model – SARIMA(p,d,q) (P,D,Q,F)

- SARIMA – Seasonal Auto Regressive Integrated Moving Average.
- For a Seasonal ARIMA / SARIMA model, we have to take care of 4 parameters such as AR (p), MA (q), seasonal AR (P) and seasonal MA (Q).
- With correct differencing (d) and seasonal differencing (D).
- Also, seasonal frequency (F) indicates the seasonal effects over a particular period.
- As we checked using ADF test in the previous segment, our time series was non-stationary. But it converted to be stationary at 1 level of differencing. Hence, we can use train data with 1 level of differencing to run SARIMA model.
- To find the minimum Akaike Information Criteria (AIC) value we ran different combination of 'pdq' and 'PDQF':
    - Where, value of 'p', 'P', 'q' and 'Q' ranges from 0 to 3.
    - 'd' = 1 remains constant as we made the time series stationary at 1 level of differencing.
    - 'D' = 0, as we have already stationarized the data once.
    - 'F' = 4, as from the below Autocorrelation plot, we can observe a pattern at each 4th occurrence.
- As we can see from the below table, the lowest AIC value is achieved at (p,d,q)(P,D,Q,F) as (0, 1, 3) (3, 0, 3, 4).

	param	seasonal	AIC
63	(0, 1, 3)	(3, 0, 3, 4)	1710.552848
127	(1, 1, 3)	(3, 0, 3, 4)	1711.542457
191	(2, 1, 3)	(3, 0, 3, 4)	1714.121986
255	(3, 1, 3)	(3, 0, 3, 4)	1714.727577
251	(3, 1, 3)	(2, 0, 3, 4)	1714.874679

Table 1. 4: AIC - SARIMA

- Here is the SARIMA model with automated values of (p,d,q)(P,D,Q,F):

SARIMAX Results

Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(0, 1, 3)x(3, 0, 3, 4)	Log Likelihood	-845.276
Date:	Tue, 13 Dec 2022	AIC	1710.553
Time:	15:00:08	BIC	1738.002
Sample:	0	HQIC	1721.694
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.7687	0.100	-7.685	0.000	-0.965	-0.573
ma.L2	-0.1863	0.161	-1.154	0.249	-0.503	0.130
ma.L3	0.1081	0.127	0.854	0.393	-0.140	0.356
ar.S.L4	-0.0034	0.012	-0.295	0.768	-0.026	0.019
ar.S.L8	-0.0236	0.010	-2.439	0.015	-0.043	-0.005
ar.S.L12	1.0406	0.009	117.682	0.000	1.023	1.058
ma.S.L4	-0.1269	0.139	-0.915	0.360	-0.399	0.145
ma.S.L8	-0.1061	0.124	-0.857	0.391	-0.349	0.136
ma.S.L12	-0.7675	0.098	-7.817	0.000	-0.960	-0.575
sigma2	1.271e+05	1.9e-06	6.7e+10	0.000	1.27e+05	1.27e+05

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	40.06
Prob(Q):	0.95	Prob(JB):	0.00
Heteroskedasticity (H):	2.75	Skew:	0.82
Prob(H) (two-sided):	0.00	Kurtosis:	5.37

- From this model we can infer that, ar.S.L4 is the least significant and ar.S.L12 is the most significant variable.
- We ran the diagnostic plot:

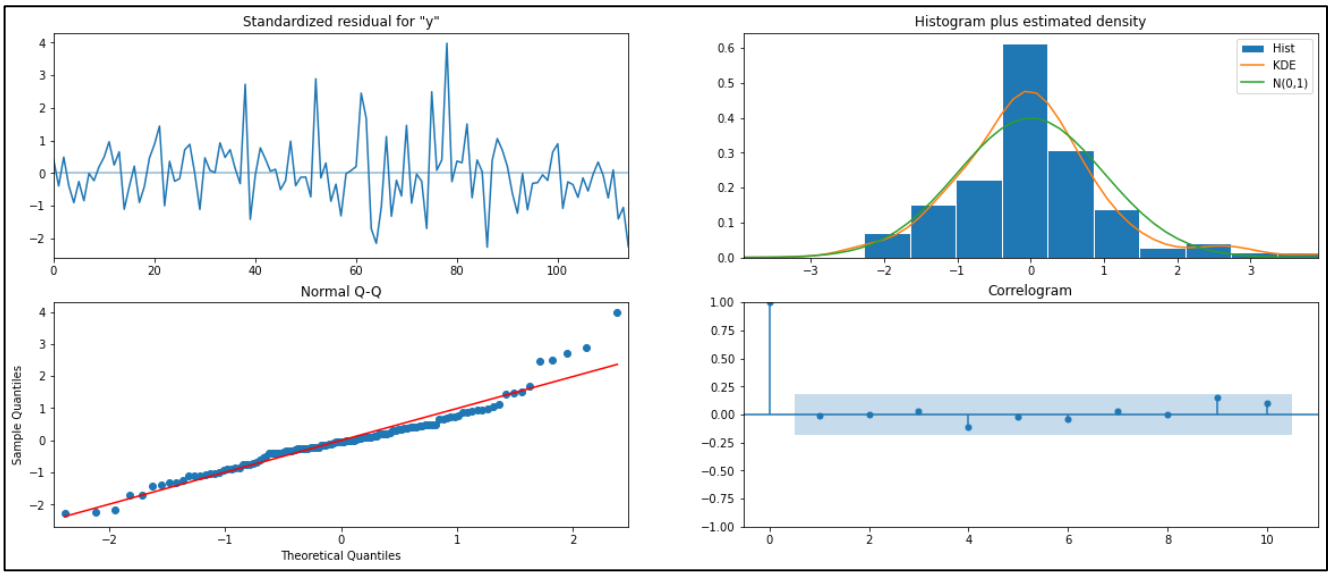


Figure 1. 34: Automated SARIMA Mode Diagnostic Plot

- In the Normal Q-Q plot, forecasted values (blue dots) are fairly close to the actual values.
- In Correlogram, all the data points are within the significance zone. This indicates that we have considered adequate amount of correlation. Hence, significantly model has performed well, using the optimum value of p and q.

- Predict on the Test Set using this model and evaluate the forecast. For Automated_SARIMA(0, 1, 3) (3, 0, 3, 4) forecast on the train data, **RMSE is 564.925**. This is how the forecast appear against actual values:

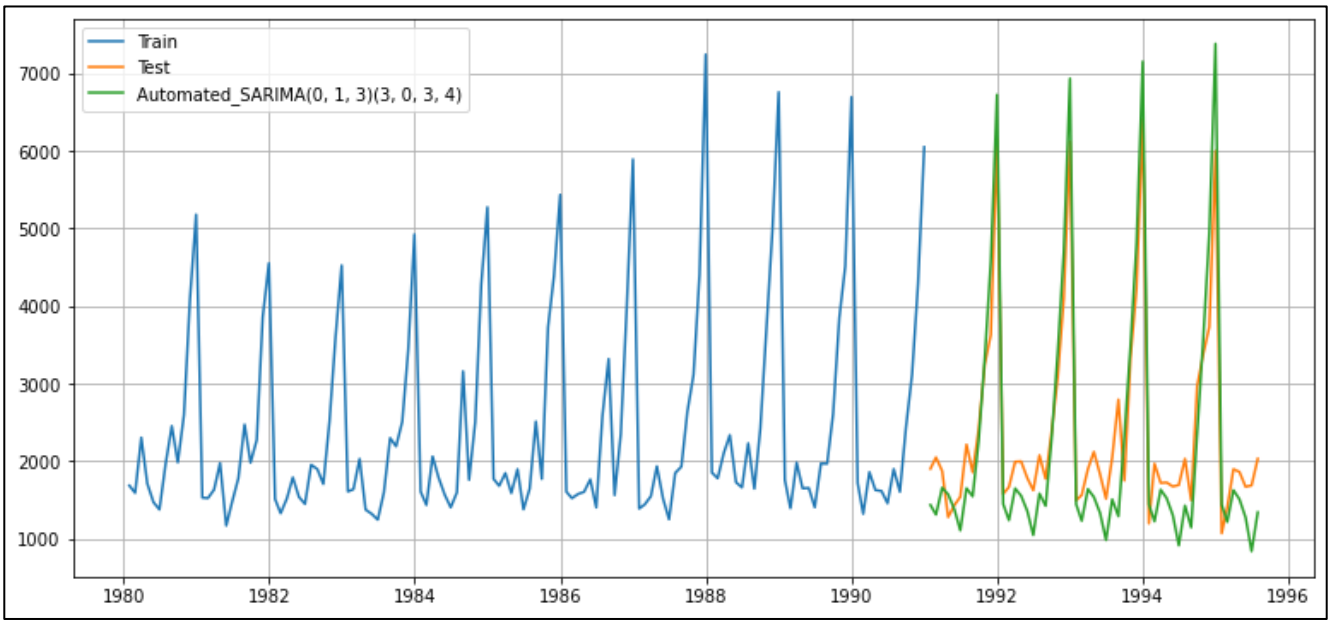


Figure 1. 35: Automated_SARIMA(0, 1, 3) (3, 0, 3, 4)

- From the Automated_SARIMA model we can observe that the forecast has accounted for seasonality and trend very well.
- The RMSE is low as compared to automate ARIMA model. Simply because SARIMA is 'Seasonal' ARIMA, better fit for time series with seasonality and trend component.

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

1.7.1 Manual ARIMA Model - ARIMA(p,d,q)

- For manual ARIMA model, we set the value of AR (p) and MA (q) using Autocorrelation function (ACF) and Partial Autocorrelation function (PACF) plots, respectively.
- These plots are created using the difference train data of level 1 (d = 1).

ACF Plot

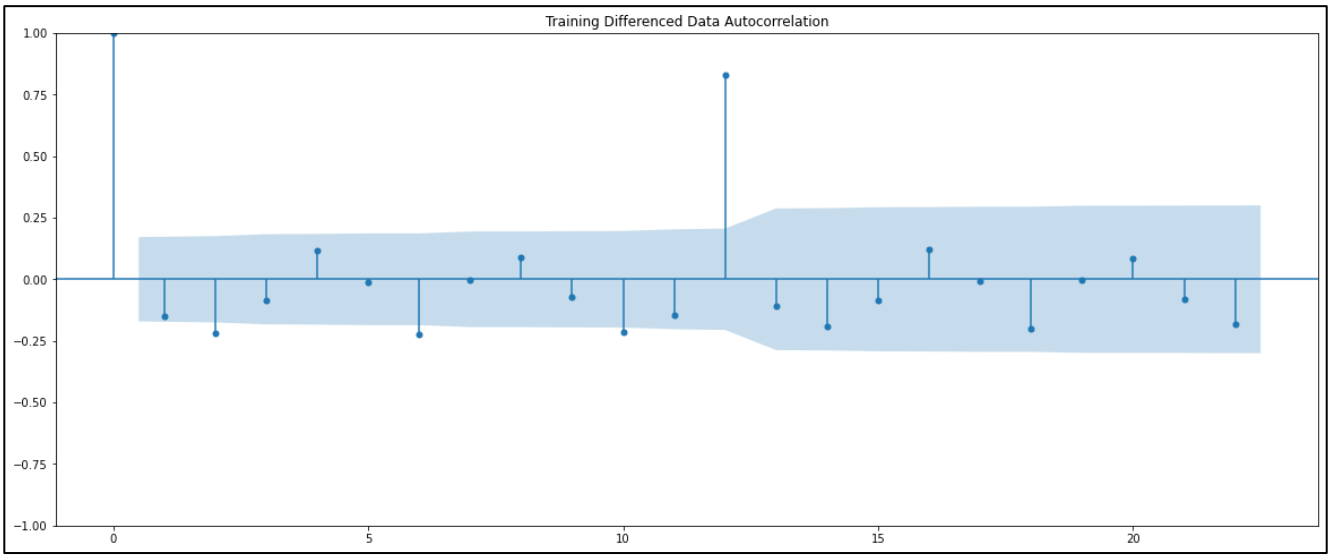


Figure 1. 36: ACF Plot

PACF Plot

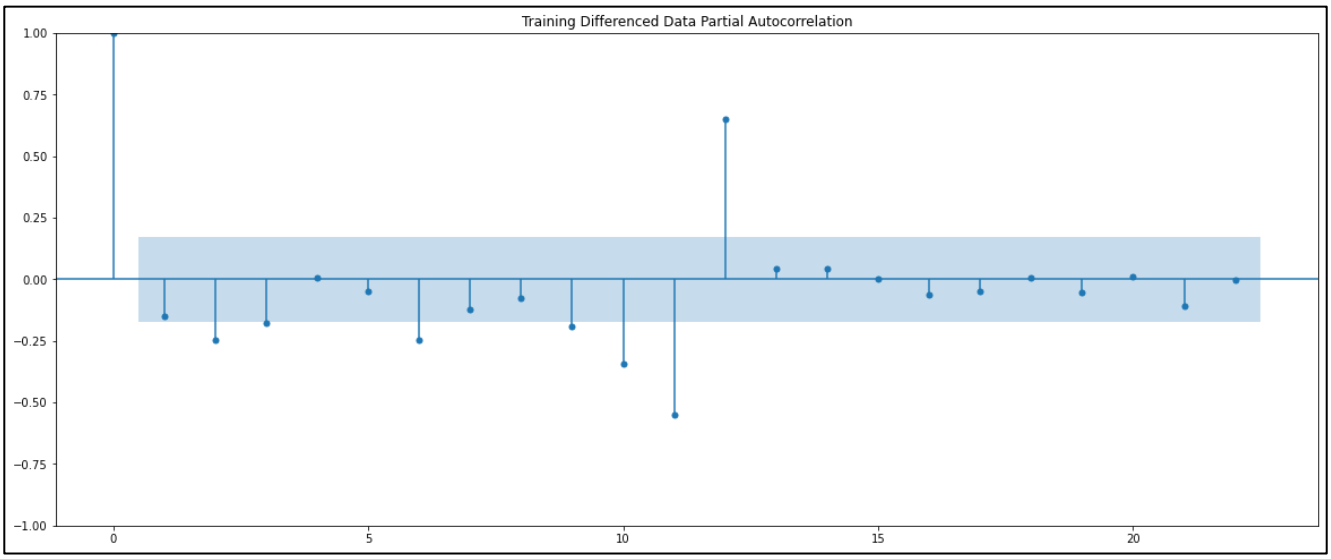


Figure 1. 37: PACF Plot

- Here, we have taken alpha=0.05.
- The Moving-Average parameter in an ARIMA model is 'q' which comes from the significant lag before the ACF plot cuts-off to 0.
- The Auto-Regressive parameter in an ARIMA model is 'p' which comes from the significant lag before which the PACF plot cuts-off to 0.
- By looking at the above plots, we can say that both the PACF and ACF plot cuts-off at lag 0.

- Here is the manual ARIMA model with values of (0,1,0):

SARIMAX Results

Dep. Variable:	Sparkling	No. Observations:	132
Model:	ARIMA(0, 1, 0)	Log Likelihood	-1132.832
Date:	Tue, 13 Dec 2022	AIC	2267.663
Time:	15:00:09	BIC	2270.538
Sample:	01-31-1980	HQIC	2268.831
	- 12-31-1990		
Covariance Type:	opg		

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

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

- There is no variable information given as the value of p and q both is 0.
- Predict on the Test Set using this model and evaluate the forecast. For Automated_ARIMA(0,1,0) forecast on the train data, **RMSE is 3864.279**. This is how the forecast appear against actual values:

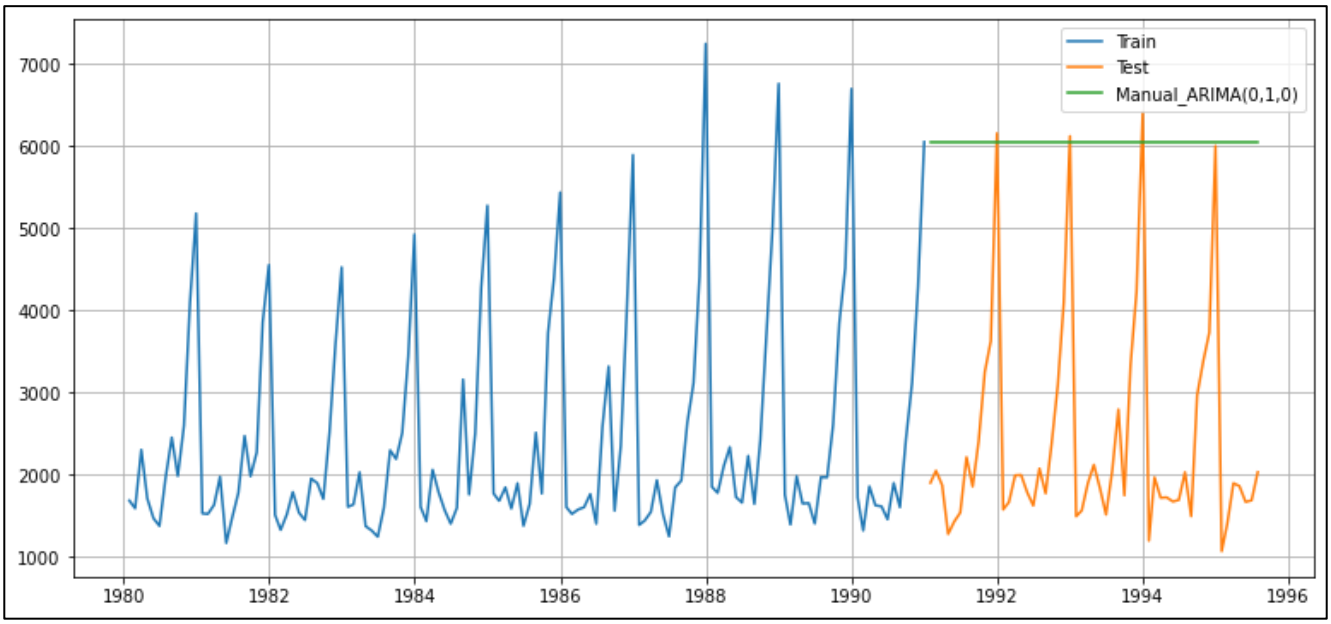


Figure 1. 38: Manual_ARIMA(0,1,0)

- From the Manual_ARIMA model we can observe that the forecast is not considering seasonality and trend. Giving a static forecast.
- The RMSE is also not the lowest among all the models ran so far.
- This not optimum model to forecast for the sparkling wine sales date.

1.7.2 Manual SARIMA Model - SARIMA(p,d,q) (P,D,Q,F)

- For manual SARIMA model, we have taken alpha=0.05.
- We are going to take the seasonal period as 4. We are taking values of AR (p) and MA (q) to be 0 and differencing (d) to be 1, as the parameters same as the manual_ARIMA model.

ACF Plot – original training data

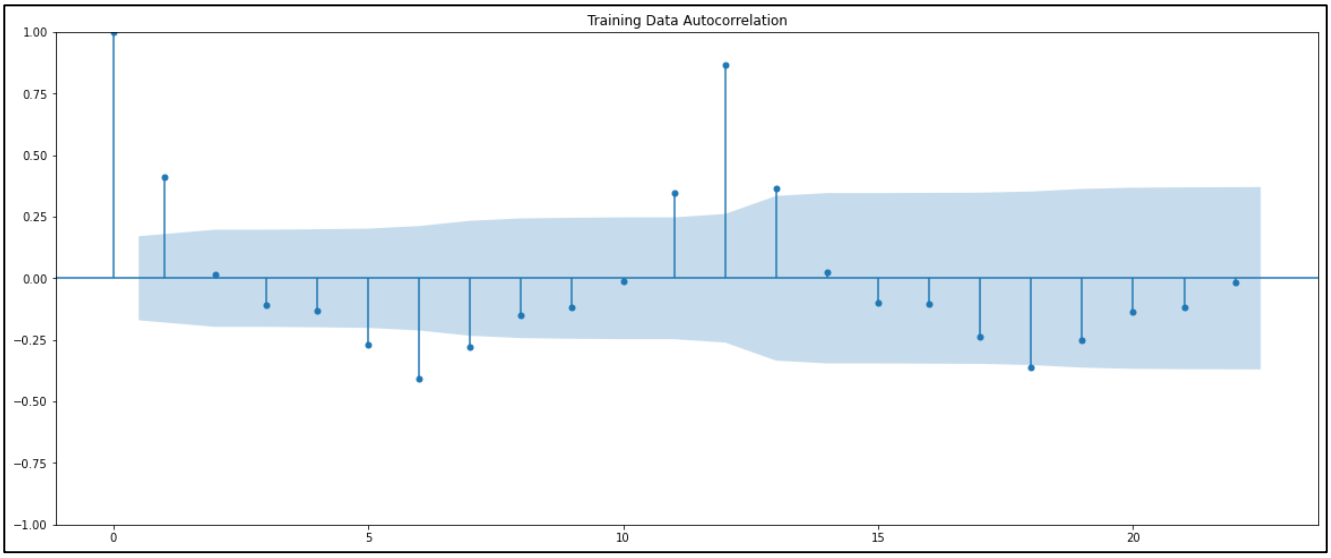


Figure 1. 39: ACF Plot – original training data

PACF Plot – original training data

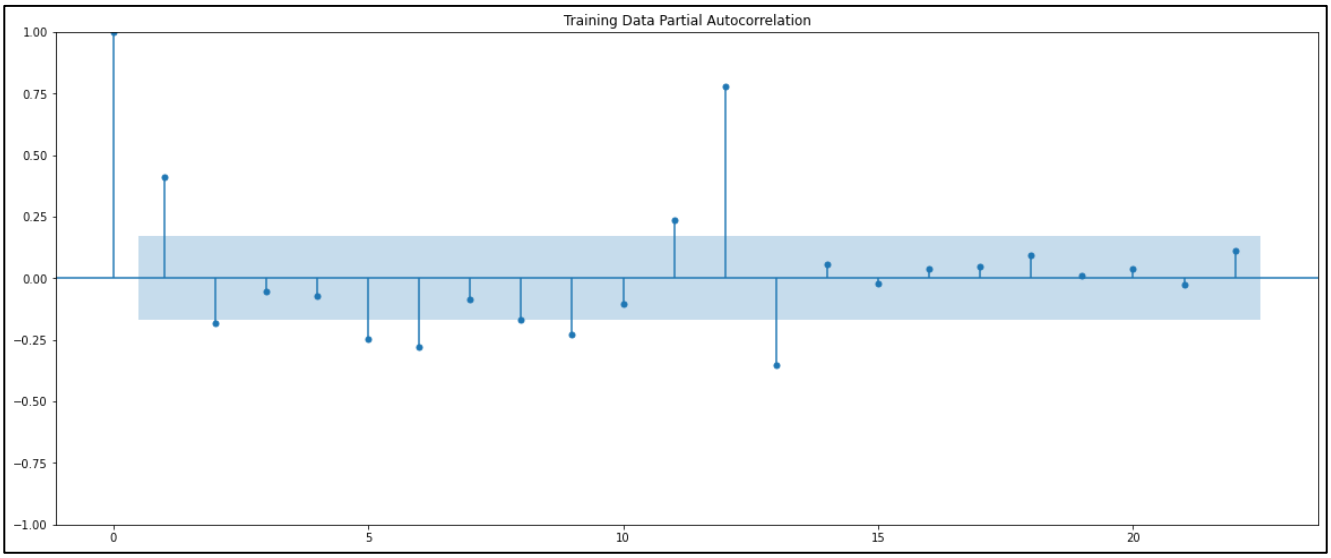


Figure 1. 40: PACF Plot – original training data

- The Auto-Regressive parameter in a SARIMA model is 'P' which comes from the significant lag after which the PACF plot cuts-off to 2.
- The Moving-Average parameter in a SARIMA model is 'Q' which comes from the significant lag after which the ACF plot cuts-off to 1.
- To determine the value of 'P' and 'Q', non-stationarized data is considered. Because the data stationarizing was already performed for the value of 'p' and 'q'.
- Hence, the value of 'D' remains constant as 0.

- 'F' = 4, as from the Autocorrelation plot, we can observe a pattern at each 4th occurrence.
- Here is manual SARIMA model with values of (0,1,0) (2, 0, 1, 4):

SARIMAX Results

Dep. Variable:	Sparkling	No. Observations:	132
Model:	SARIMAX(0, 1, 0)x(2, 0, [1], 4)	Log Likelihood	-1043.193
Date:	Wed, 14 Dec 2022	AIC	2094.386
Time:	18:48:04	BIC	2105.635
Sample:	01-31-1980	HQIC	2098.955
	- 12-31-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.S.L4	0.6214	0.193	3.220	0.001	0.243	1.000
ar.S.L8	0.3867	0.176	2.191	0.028	0.041	0.733
ma.S.L4	-0.9469	0.163	-5.792	0.000	-1.267	-0.626
sigma2	1.862e+06	1.06e-07	1.75e+13	0.000	1.86e+06	1.86e+06

Ljung-Box (L1) (Q):	3.20	Jarque-Bera (JB):	26.94
Prob(Q):	0.07	Prob(JB):	0.00
Heteroskedasticity (H):	2.11	Skew:	-0.46
Prob(H) (two-sided):	0.02	Kurtosis:	5.10

- From this model we can infer that, ma.S.L4 is the most significant and ar.S.L8 is the least significant variable.
- We ran the diagnostic plot:

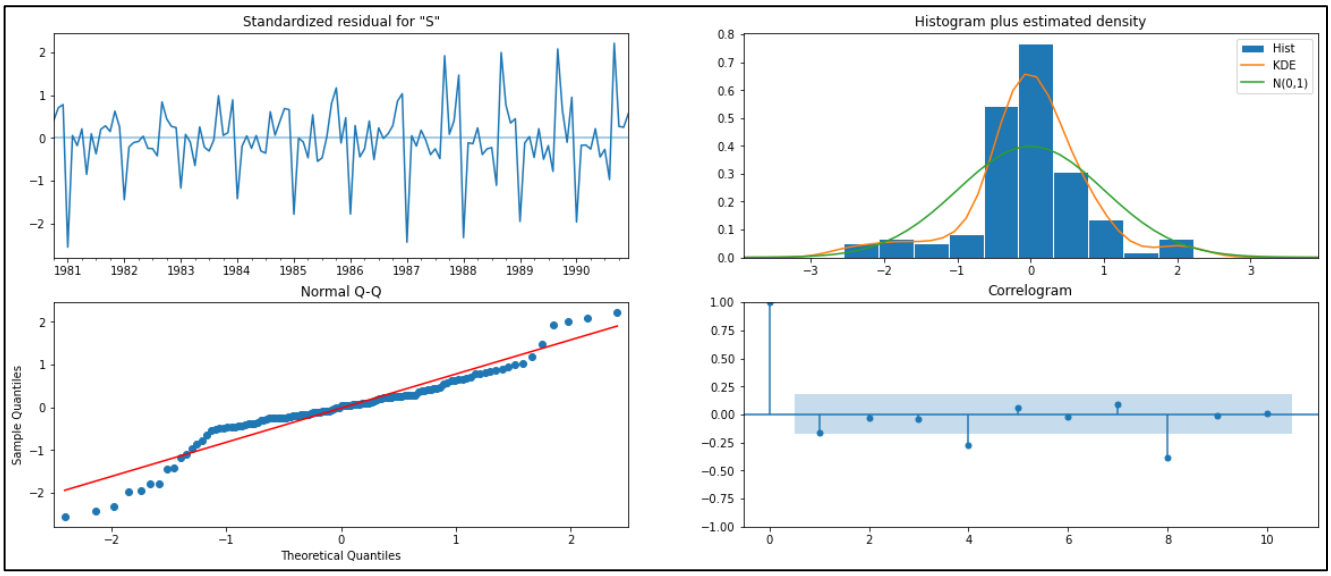


Figure 1. 41: Manual SARIMA Mode Diagnostic Plot

- In the Normal Q-Q plot, forecasted values (blue dots) are falling far from the actual values.
- In Correlogram, not all the data points are within the significance zone. This indicates that manual SARIMA model did not consider the adequate amount of correlation. Hence, significantly model has not performed well, using the optimum value of parameters.
- Predict on the Test Set using this model and evaluate the forecast. For Manual_SARIMA(0,1,0) (2, 0, 1, 4) forecast on the train data, **RMSE is 2812.584**. This is how the forecast appear against actual values:

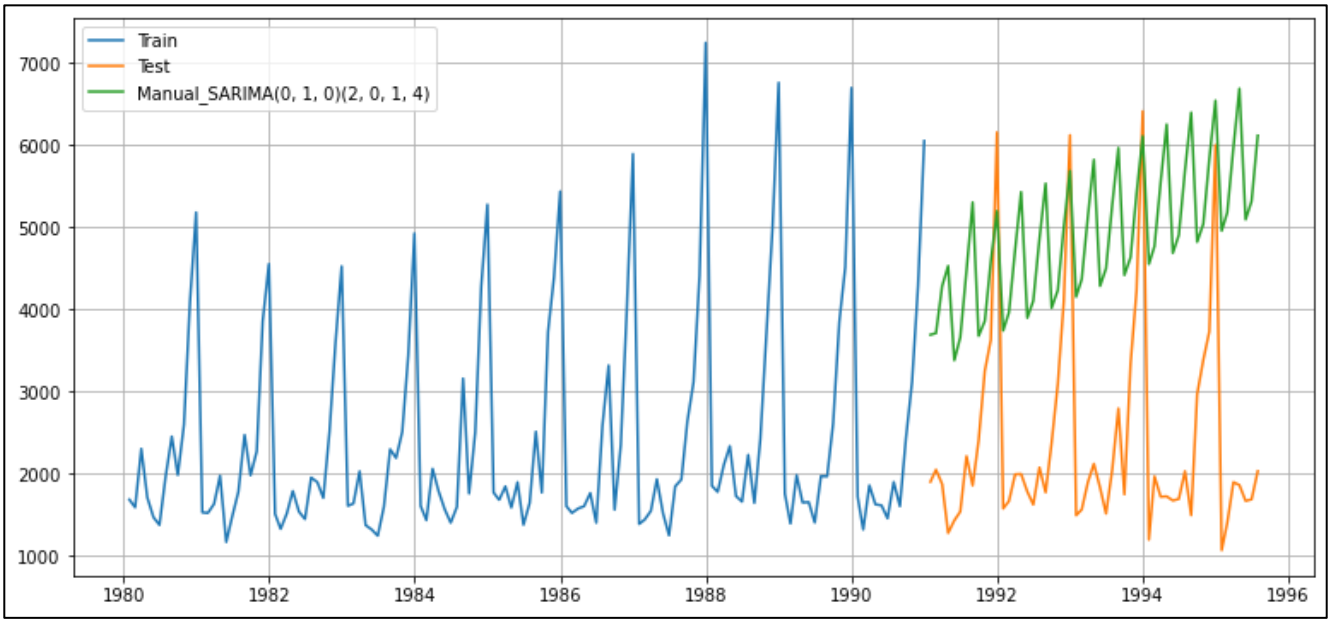


Figure 1. 42: Manual_SARIMA(0,1,0) (2, 0, 1, 4)

- From the Manual_SARIMA model we can observe that the forecast has accounted for trend well but seasonality seems to be smoothened out as compared to the test data.
- The RMSE is also high as compared to the Automated SARIMA model.
- This model is not optimum to be used for the forecast of our data.

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

Here is the table with RMSE values of all the models built using training data and applied on test data, in lowest first order:

	Test RMSE
Alpha=0.4,Beta=0.1,Gamma=0.2,TripleExponentialSmoothing_Multiplicative	317.434
Alpha=0.1,Beta=0.4,Gamma=0.1,TripleExponentialSmoothing_Additive	342.935
Alpha=0.1,Beta=0.01,Gamma=0.509,TripleExponentialSmoothing_Additive	379.696
Alpha=0.11,Beta=0.049,Gamma=0.36,TripleExponentialSmoothing_Multiplicative	406.510
Automated_SARIMA(0, 1, 3)(3, 0, 3, 4)	564.925
2pointTrailingMovingAverage	813.401
4pointTrailingMovingAverage	1156.590
SimpleAverageModel	1275.000
Alpha=0.02,SimpleExponentialSmoothing	1278.498
6pointTrailingMovingAverage	1283.927
Automated_ARIMA(2,1,2)	1299.980
Alpha=0.07,SimpleExponentialSmoothing	1338.012
9pointTrailingMovingAverage	1346.278
RegressionOnTime	1389.135
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	1777.735
Manual_SARIMA(0, 1, 0)(2, 0, 1, 4)	2812.584
NaiveModel	3864.279
Manual_ARIMA(0,1,0)	3864.279
Alpha=0.66,Beta=0.00,DoubleExponentialSmoothing	3949.993

Table 1. 5: RMSE Table

From this table, we see that the best model is the Triple Exponential Smoothing with multiplicative seasonality with the parameters Alpha = 0.4, Beta = 0.1 and Gamma = 0.2.

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

Triple Exponential Smoothing with multiplicative seasonality with the parameters Alpha = 0.4, Beta = 0.1 and Gamma = 0.2, provides the least RMSE value. Hence, the most optimum model for our data.

- We built this model on the complete data and the **RMSE comes to 376.775**.
- Calculated the predictions for 12 months into the future, with the upper and lower confidence bands at 95% confidence level.
- This is the sample of forecasted data:

	lower_CI	prediction	upper_ci
1995-08-31	1322.989	2063.449	2803.909
1995-09-30	1838.948	2579.408	3319.867
1995-10-31	2676.195	3416.655	4157.115
1995-11-30	3564.018	4304.478	5044.937
1995-12-31	5864.418	6604.877	7345.337

Table 1. 6: Wine Sales Forecast with CI

- This is how the forecasted values appear visually:

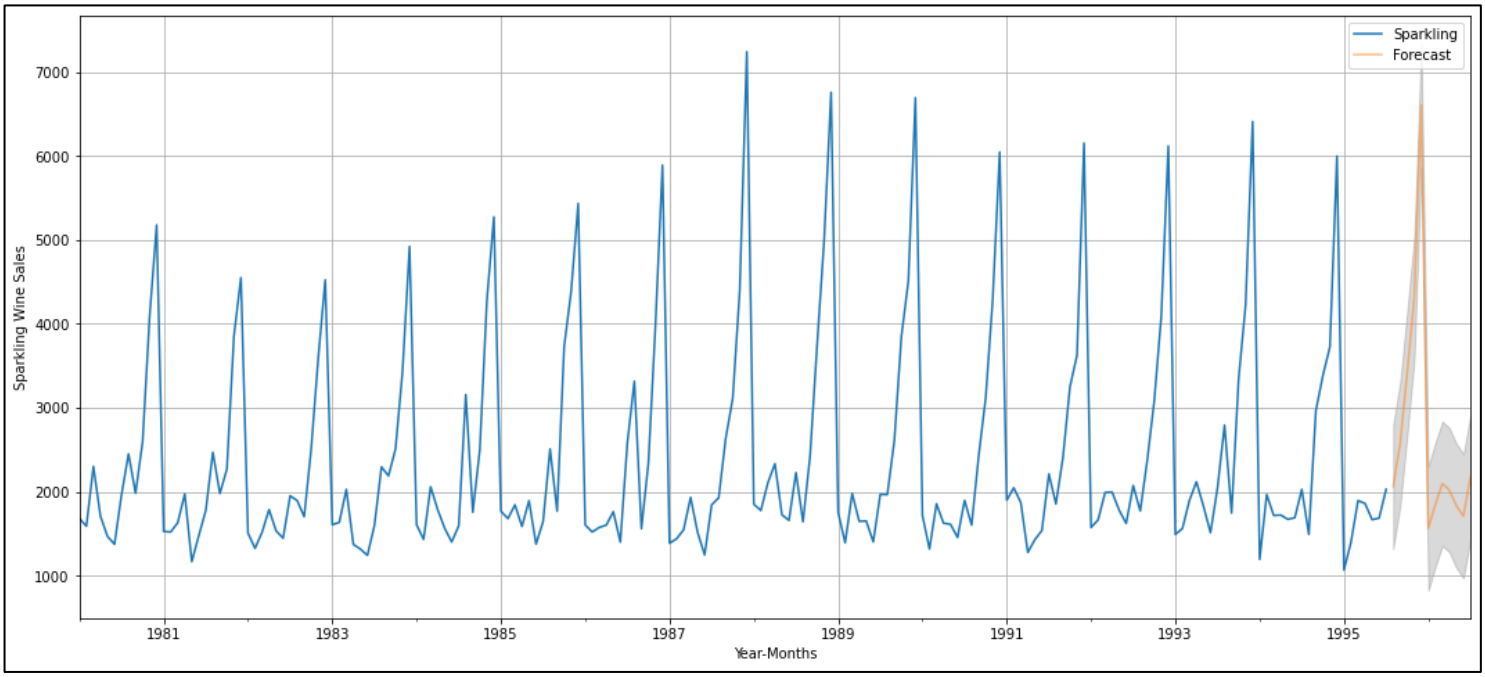


Figure 1. 43: Forecast on Compete Data with Confidence Interval

- The forecasts that we calculated are not definite values, as the future is unpredictable. It is best to provide forecasts in a range, to account for any unprecedented event that might occur in future.
- We have taken a confidence interval of 95% to present our forecasts. The orange line in the graph is the forecasts and the grey area around it is the confidence interval, showing the upper and lower limit to expect as the forecast.

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

- The model we have built, takes in account the trend and seasonality present in our data to make the forecasts.
- Giving a clear indication to the business as to what to expect in terms of sparkling wine sales.
- As per the forecast, the trend is increasing and seasonality is also multiplying.
- Which is a good signal for the wine sales.
- Company can expect the sales more than 7000 in the busiest season.
- As the company is approaching a busy season in 1995 Q4, they can plan well and stock up the wines to meet higher sales.
- Now is the perfect time to roll out the advertisements and/or marketing campaigns to boost the sales even further.

2. Rose Wine Sales

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

The CSV file is loaded using pandas function read_csv() to perform analysis.

2.1.1 Sample of dataset

Here are the top 5 rows (sample) of the dataset:

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

Table 2. 1: Dataset Sample

- Dataset has 2 variables, showing the time component of the data and the sale of rose wine over the month and years.

2.1.2 Plotting data

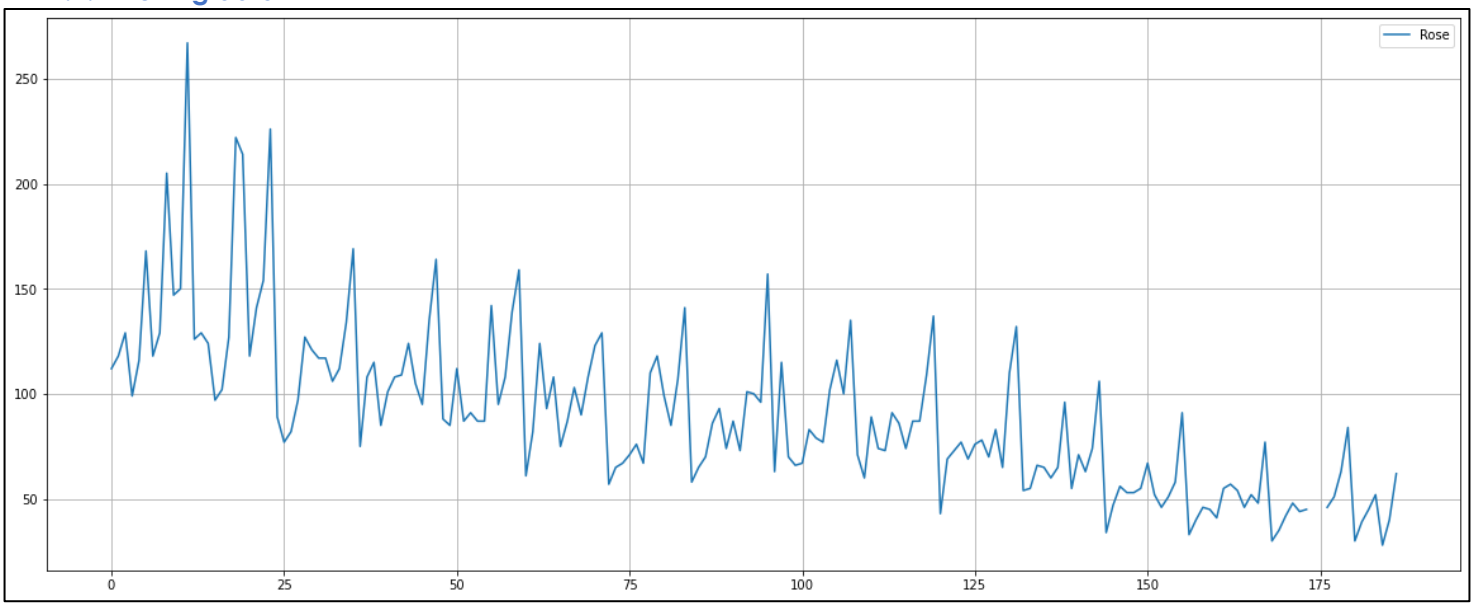


Figure 2. 1: Plotting Original Data

- Though the above plot looks like a Time Series plot, notice that the x-axis is not time. In order to make x-axis as time series, we passed the date range manually through 'date_range' command in pandas.
- After appending the time series, named 'Time_Stamp' column in our dataset, we assigned time series as index of our dataset.
- This is how the plotting of modified dataset looks:

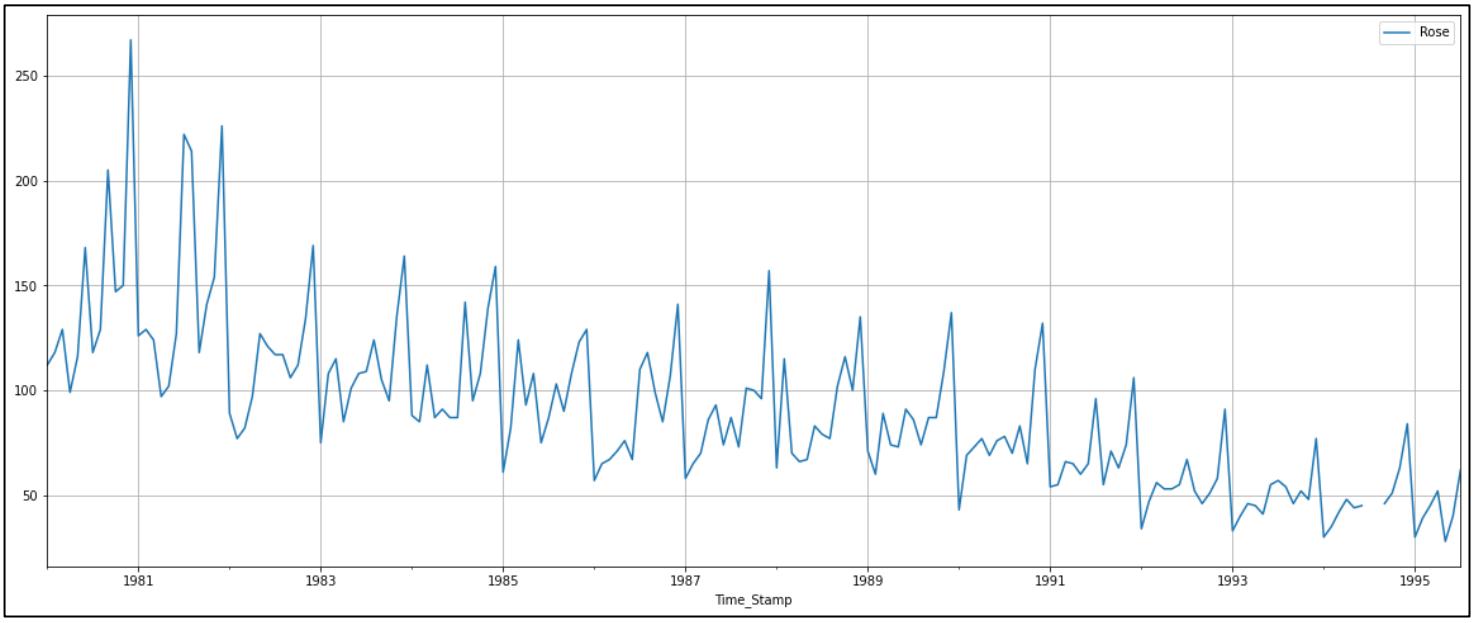


Figure 2. 2: Plotting Modified Data with Time_Stamp

- As we can observe now, x-axis is a time series, showcasing observations of wine sales over the years.
- There is trend and seasonality present in this time series. The magnitude of seasonality is changing with time, hence the seasonality is multiplicative. The trend is downward.

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

2.2.1 Types of variables in the dataset

```
DatetimeIndex: 187 entries, 1980-01-31 to 1995-07-31
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Rose    185 non-null    float64
dtypes: float64(1)
memory usage: 2.9 KB
```

- There are 187 observations in our data, ranging from January 1980 till July 1995.
- The 'Rose' column contains the wine sales volume for each month during the given period. There are only 185 values present in this column, indicating the time series is missing 2 values.

2.2.2 Missing Values

- The sales values for July and August, 1994 are missing from our data. To maintain the sequential nature of our time series, we are not removing the time period that is missing values, we rather assign values to it.

Time_Stamp	Rose
1994-07-31	NaN
1994-08-31	NaN

Table 2. 2: Missing Values

- As we observed seasonality in our data, it is a good practice to impute missing values using past season's data. As such, we used the average sales value of June, July, August and September from 1993 to impute the missing sales numbers for July and August, 1994.
- Here is how the values populated in 1994:

Time_Stamp	Rose
1994-01-31	30.0
1994-02-28	35.0
1994-03-31	42.0
1994-04-30	48.0
1994-05-31	44.0
1994-06-30	45.0
1994-07-31	53.0
1994-08-31	53.0
1994-09-30	46.0
1994-10-31	51.0
1994-11-30	63.0
1994-12-31	84.0

Table 2. 3: Monthly Data - 1994

2.2.3 Data Description

	Rose
count	187.000000
mean	89.994652
std	39.154573
min	28.000000
25%	62.500000
50%	85.000000
75%	111.000000
max	267.000000

Table 2. 4: Data Description

- There are a total of 187 records, indicating that the data is of monthly frequency.
- The average rose wine sale of 187 months was 89.99.
- The minimum sale was of 29 wines and the maximum sale was of 267 wines.
- The data description of a time series data doesn't paint a correct picture, as the factors like trend, seasonality and certain spikes in the outcome are not well accounted for.

2.2.4 Exploratory Data Analysis

Yearly Sales Bar Plot

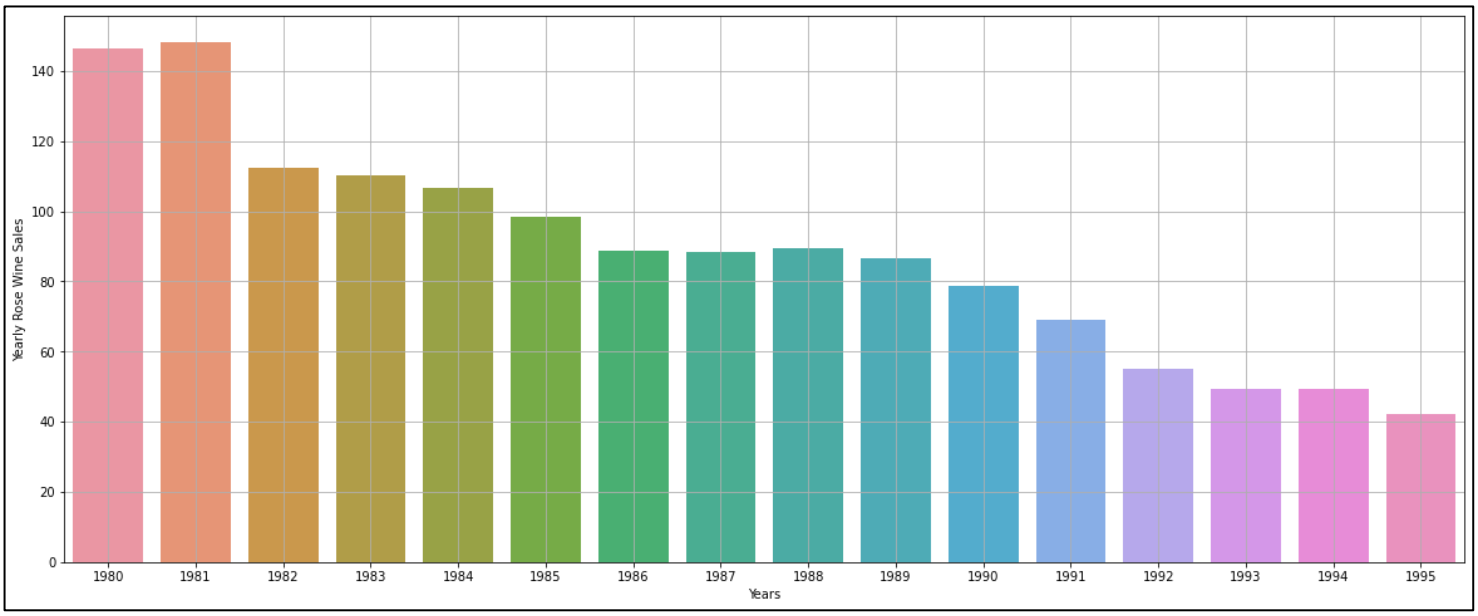


Figure 2. 3: Yearly Sales – bar plot

- From the yearly sales plot, we can see that year 1981 marks the highest sales of rose wine.
- Although, the lowest sales appear to be in the year 1995, but we only have 7 months data from that year. After that, years 1993 and 1994 record minimum sales of 594 each.

Yearly Sales Box Plot

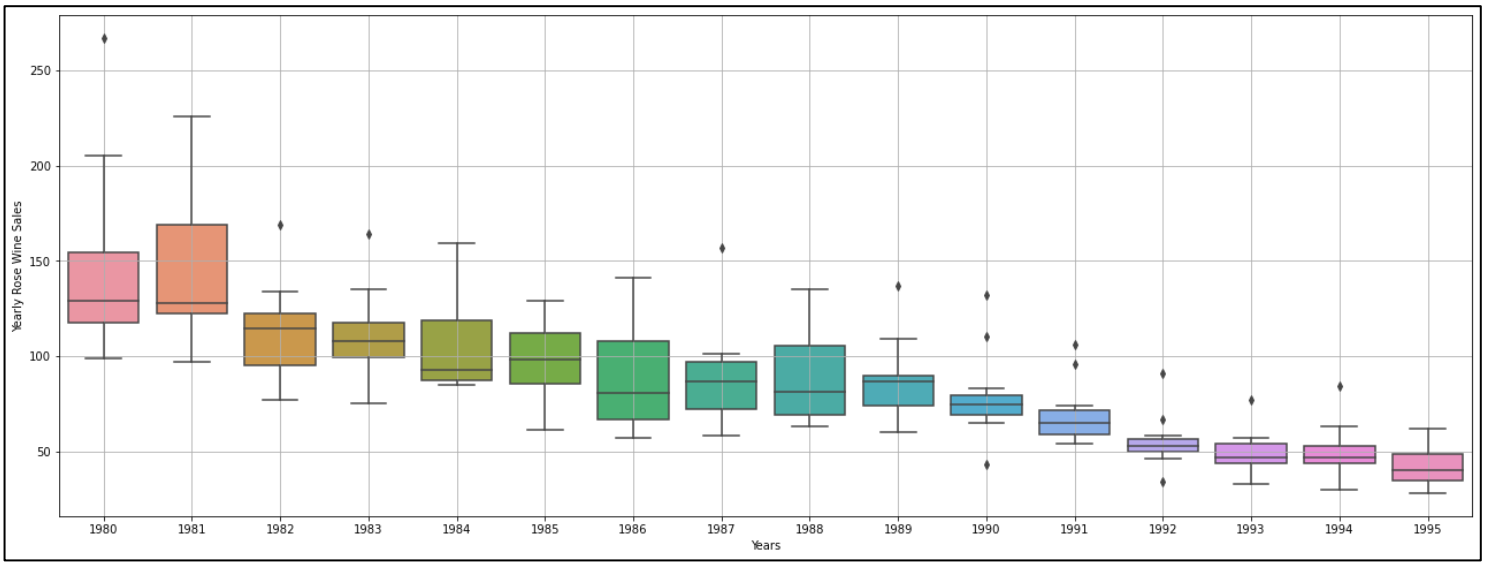


Figure 2. 4: Yearly Sales – box plot

- From the box plot we can observe that the year 1980 seems to be the most inconsistent in terms of wine sales. The median and upper limit are quite far from the outliers present in that year; we can say that the company was not prepared for that much of a high sale.
- We can say that company had least inconsistent sales in the years 1985 and 1986.

Monthly Sales Bar Plot

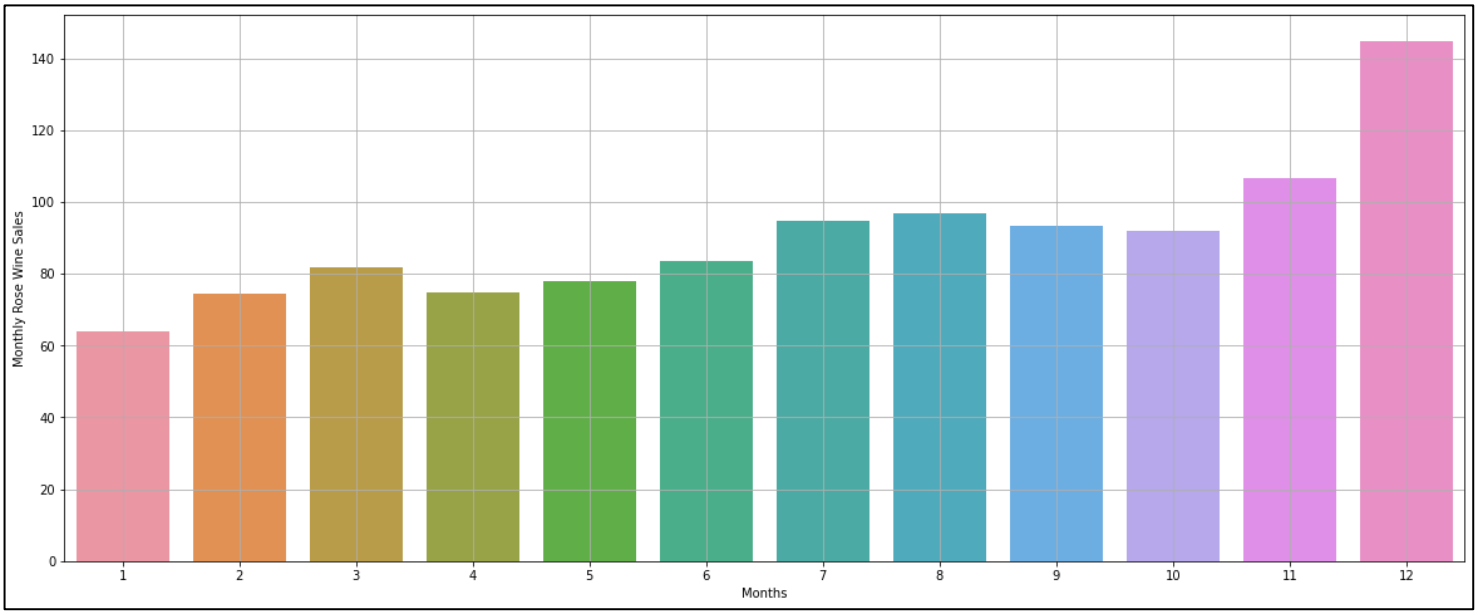


Figure 2. 5: Monthly Sales – bar plot

- Overall, December month accounts for highest sales of rose wine, provided that more people celebrate festive season.
- The minimum wine sales observed in month of January.
- The busiest time for the wine company is fourth quarter. The leanest period is first and second quarter, when the sale is not a much.

Monthly Sales Box Plot

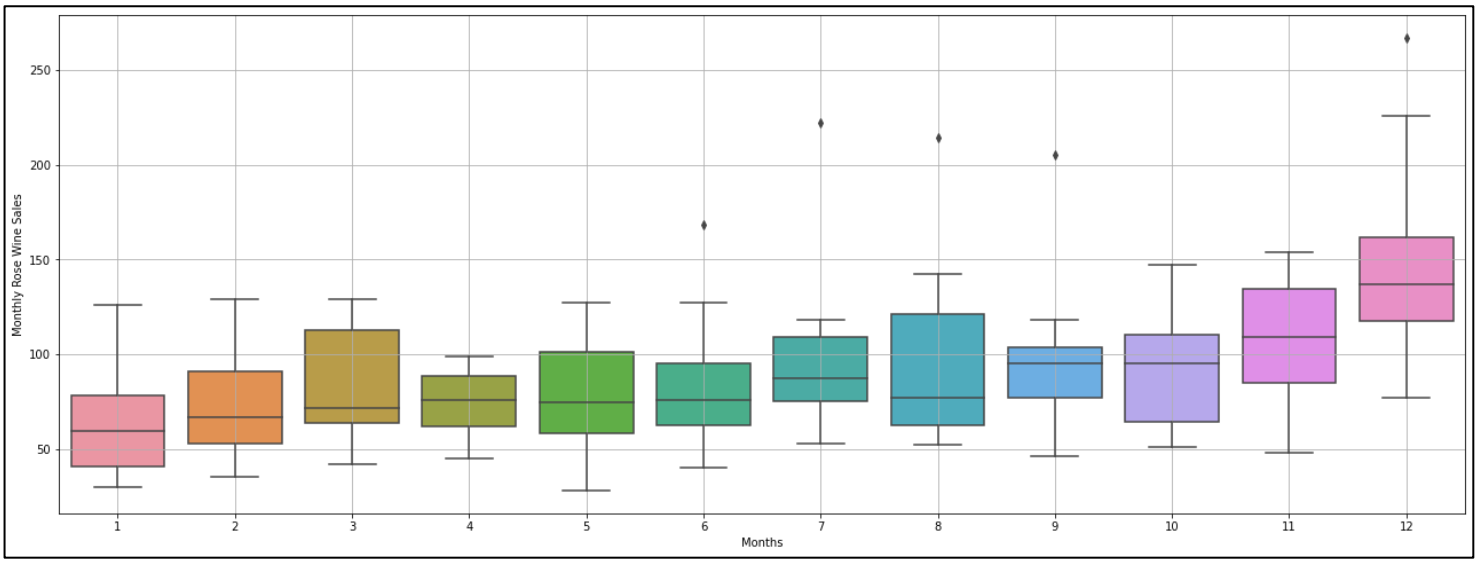


Figure 2. 6: Monthly Sales – box plot

- The most consistent month in terms of wine sales has been November, when the demand is more and those demands are met adequately.
- We can observe outliers in June, July, August, September and December months, where the least consistent month with sales is July.

Monthplot

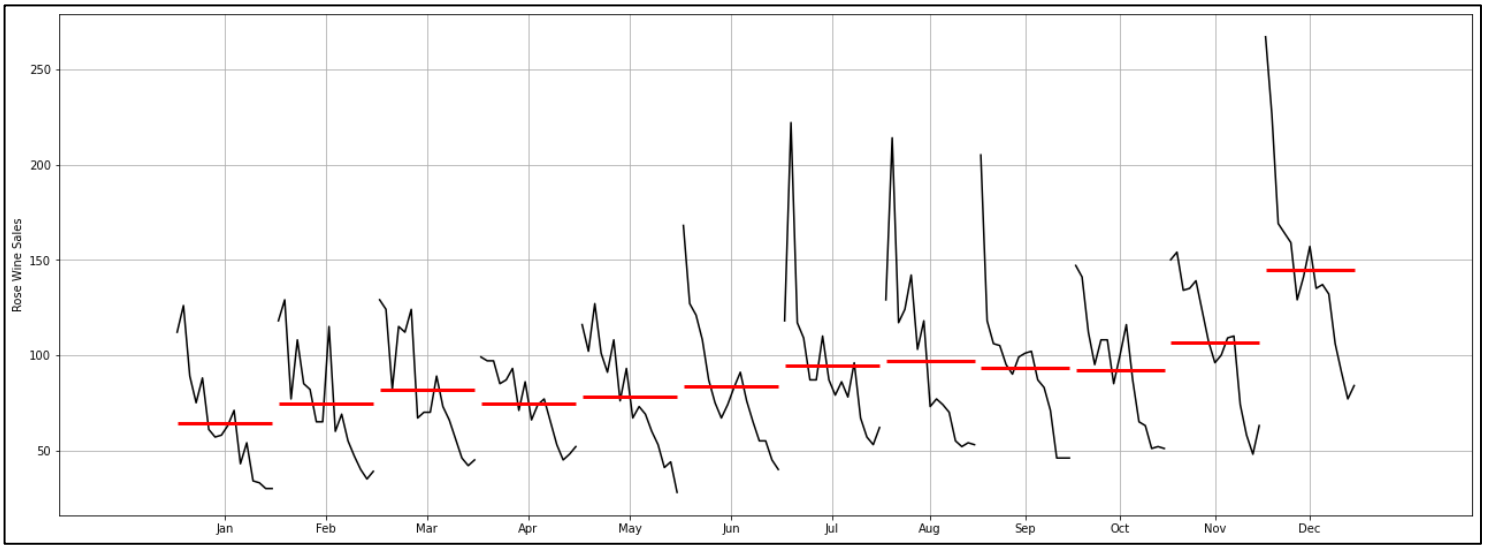


Figure 2. 7: Monthplot

- The high fluctuations can be observed in July, August and December months.
- June and September are the months with very low fluctuations.

Month on Month Sales Comparison

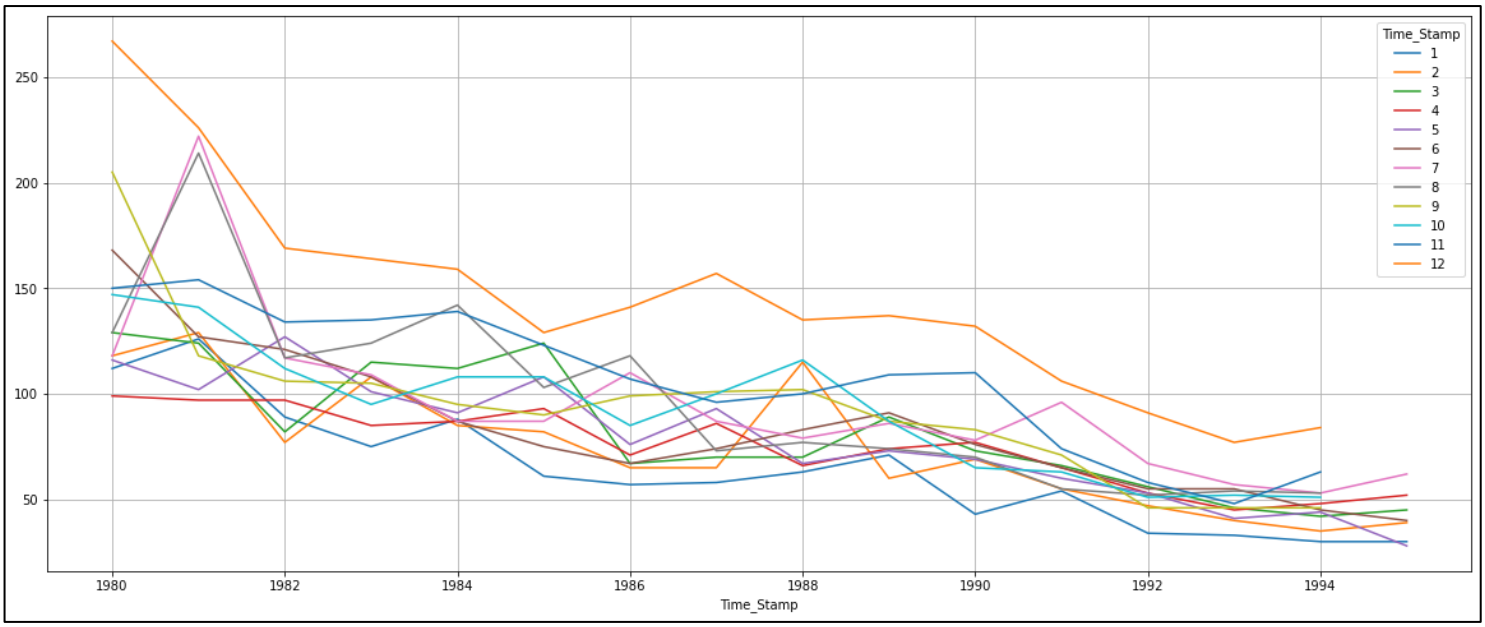


Figure 2. 8: Month on Month Sales Comparison

- December marks the highest number of sales.
- Sales for rest all months are overlapping each other, with some spikes here and there.

Empirical Cumulative Distribution Plot

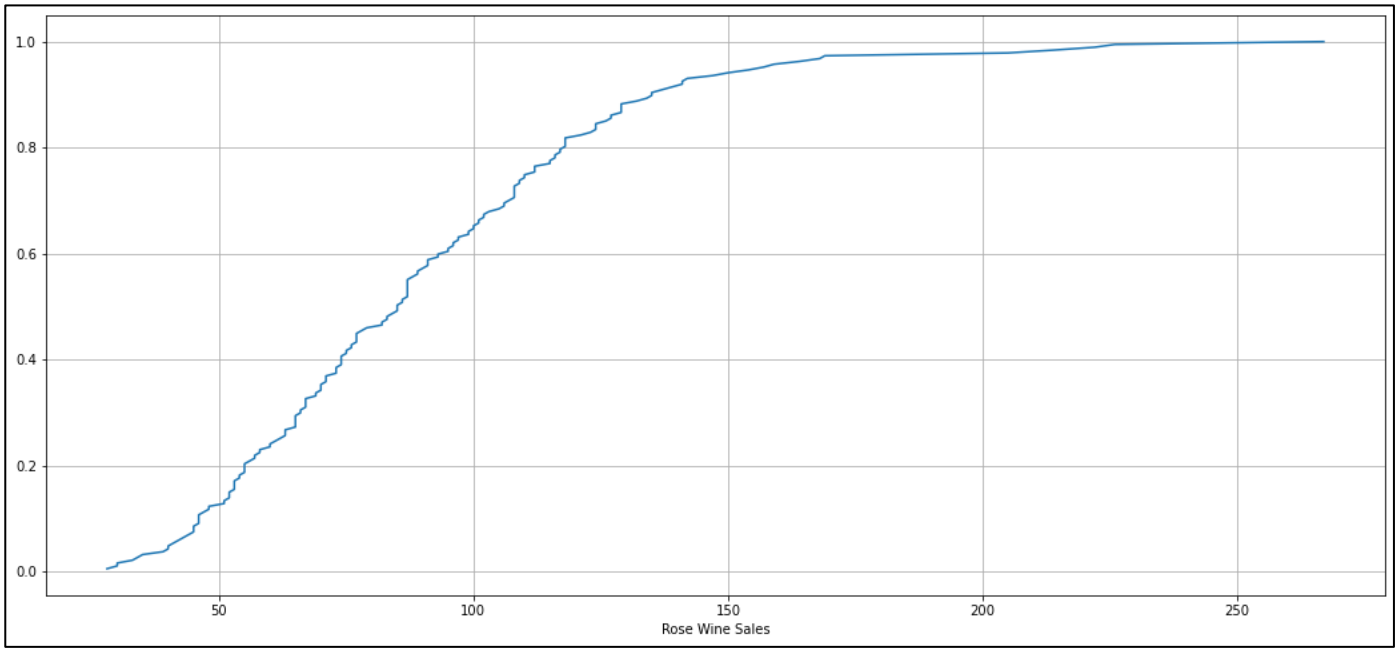


Figure 2. 9: Empirical Cumulative Distribution Plot

- The above plot shows the distributions of entire sales.
- 60% of the sales lying between 60 to 120. 20% values are lying above 120. 20% of values are lying below 60.

2.2.5 Decomposition of the Data

- A time series is composed of trend, seasonality and residuals. To analyse these components separately, the time series can be decomposed.
- There are 2 types of decomposition models:
  - Additive model – Useful when the seasonality variation is relatively constant over time.
  - Multiplicative model – Useful when the magnitude of seasonality is varying over time.
- Here is the visual representation of decomposed time series using **additive model**:

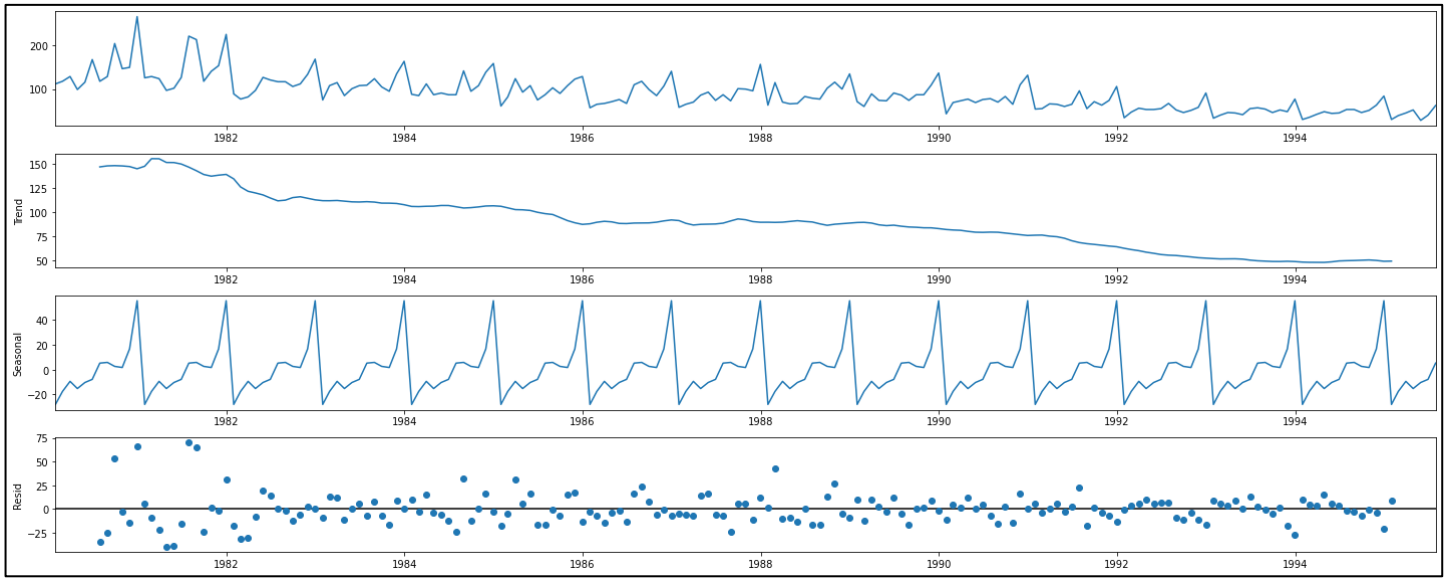


Figure 2. 10: Decomposition Plot - Additive

- First panel has the original data.
- Second panel shows the trend of the data. In this case, the trend downwards.
- Third panel shows seasonality. We can see that every year we have a repeated pattern.
- Forth panel has the errors. The errors/ residuals are showing some patterns.
- The errors are ranging from -25 to +75.
- Here is the visual representation of decomposed time series using **Multiplicative model**:

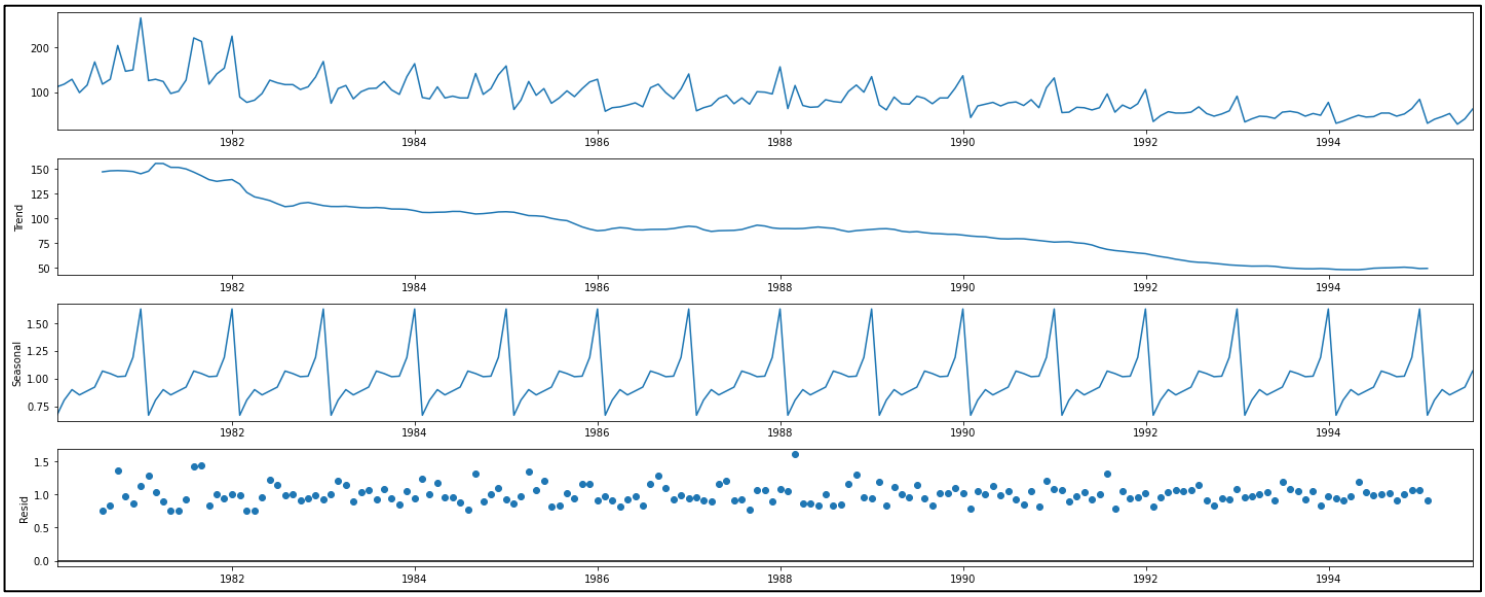


Figure 2. 11: Decomposition Plot - Multiplicative

- For the multiplicative series, we see that a lot of residuals are located around 1 and not showing outliers effect as seen in additive series.
- If we decompose a multiplicative time series using additive model, the errors continue to bear the elements of seasonality.
- In this case a multiplicative decomposition is the better choice, as the errors don't have seasonal elements.

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

- In order to run multiple analytics models, we have split the data into train and test sets. Train set will be used to build the model on; and model performance can be evaluated on the test set.
- The time series has been split into train and test sets. Train set has data from 1980 till 1990 (132 entries) and test set has data starting from 1991 till 1995 (55 entries).
- This is how the split time series looks:

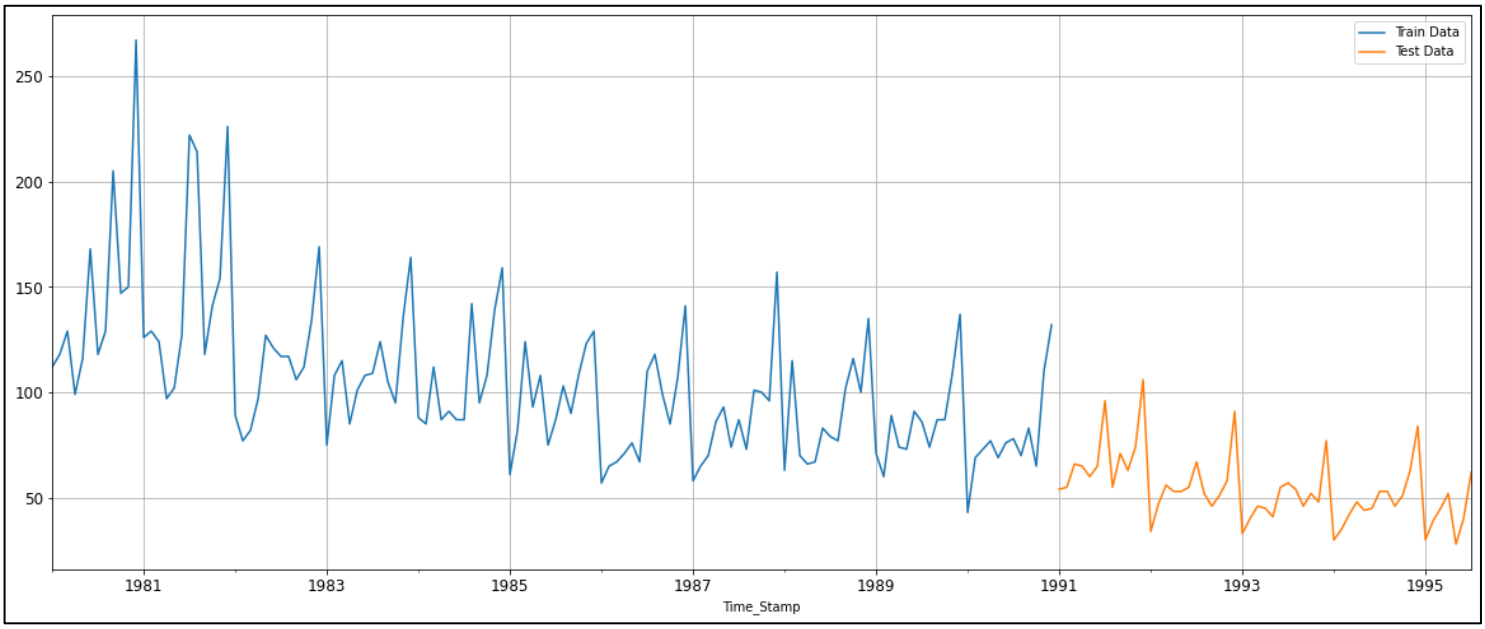


Figure 2. 12: Split Time Series

- Please note that the break between the blue and orange line is not missing data. The break is because two different sets are plotted together and test set is not picking up where train set stopped, rather it is starting from the first sales value in the test data.

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

2.4.1 Linear Regression Model

For this particular linear regression, we have regressed the 'Rose' (wine sale) variable against the order of the occurrence. For this we modified the training data before fitting it into a linear regression. We have generated a numerical time instance order for both the training and test set and added these values in the respective sets. This is how the sample of data looks like for linear regression:

First few rows of Training Data

Time_Stamp	Rose	time
1980-01-31	112.0	1
1980-02-29	118.0	2
1980-03-31	129.0	3
1980-04-30	99.0	4
1980-05-31	116.0	5

Last few rows of Training Data

Time_Stamp	Rose	time
1990-08-31	70.0	128
1990-09-30	83.0	129
1990-10-31	65.0	130
1990-11-30	110.0	131
1990-12-31	132.0	132

First few rows of Test Data

Time_Stamp	Rose	time
1991-01-31	54.0	133
1991-02-28	55.0	134
1991-03-31	66.0	135
1991-04-30	65.0	136
1991-05-31	60.0	137

Last few rows of Test Data

Time_Stamp	Rose	time
1995-03-31	45.0	183
1995-04-30	52.0	184
1995-05-31	28.0	185
1995-06-30	40.0	186
1995-07-31	62.0	187

Now that our training and test data has been modified, let us go ahead use Linear Regression to build the model on the training data and evaluate the model on the test data using Root Mean Square Error (RMSE). The lesser the RMSE, the better the model performs.

For Linear Regression forecast on the train data, **RMSE is 15.237**. This is how the forecast appear against actual values:

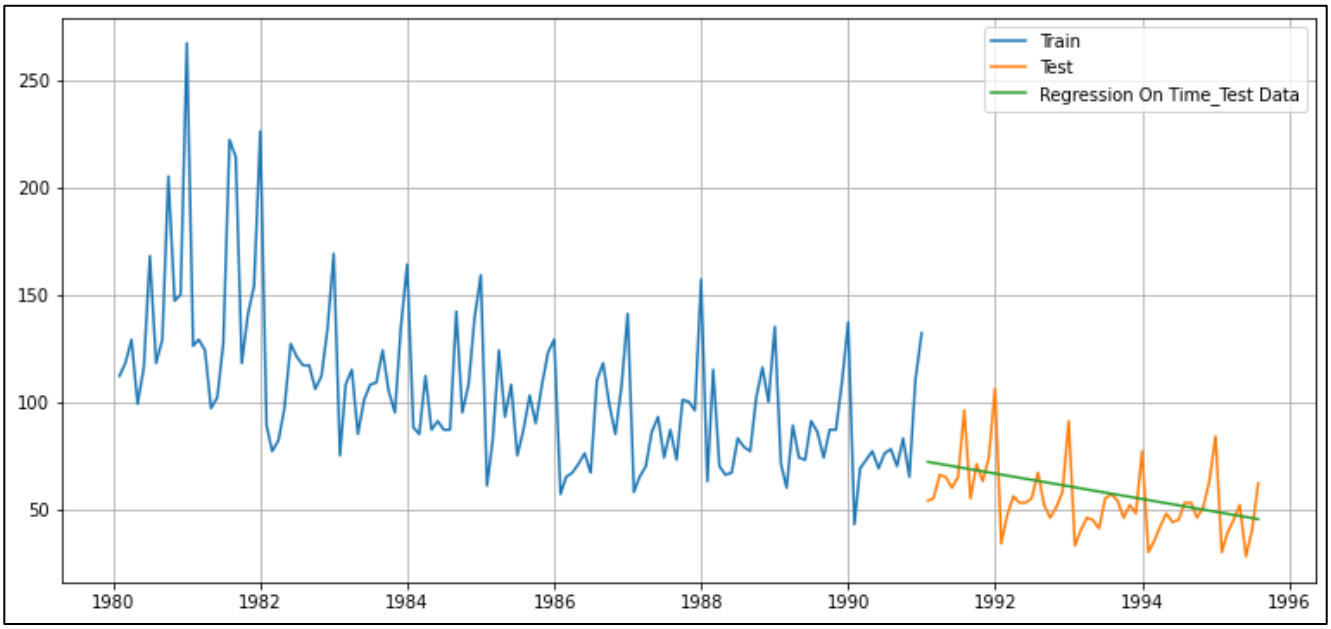


Figure 2. 13: Forecast using Linear Regression

- The training data is represented by the blue line, test data is represented by orange line.

- The green line in the graph represents the forecast made using linear regression.
- As we can see that this forecast does capture the trend of the data, but seasonality component remains missing. This indicates that linear regression is not fit for the data with seasonality, as it gives only a best fit line as an outcome.

2.4.2 Naïve Forecast Model

For this particular naive model, we can 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.

In other words, naïve forecast method used the last value in the time series to forecast the future values, and keeps it constant throughout the length of the forecast. In the training dataset, the last sales value is 132. The naïve method built the model using 132 as the forecast value for the length of test data.

For Naïve Forecast on the train data, **RMSE is 79.435**. This is how the forecast appear against actual values:

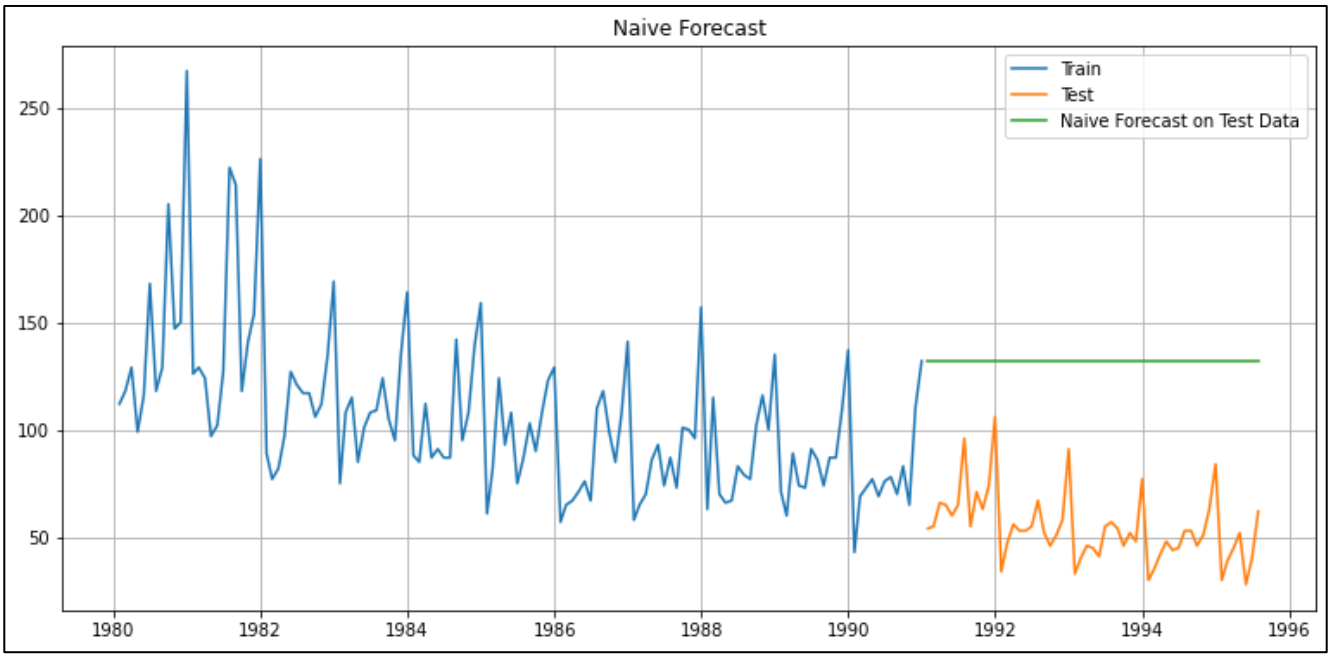


Figure 2. 14: Forecast using Naïve Model

- The RMSE value is very high than that of linear regression.
- The forecast line in green is a flat line, discounting trend and seasonality present in our data.
- This model doesn't seem to be optimum at all for forecasting wine sales.

2.4.3 Simple Average Model

For this particular simple average method, we forecasted by using the average of the training values. Since the average is impacted by outliers (abrupt changes in the time series), simple average is better than naïve forecast method where only the last value is considered for forecast.

For Simple Average Forecast on the train data, **RMSE is 53**. This is how the forecast appear against actual values:

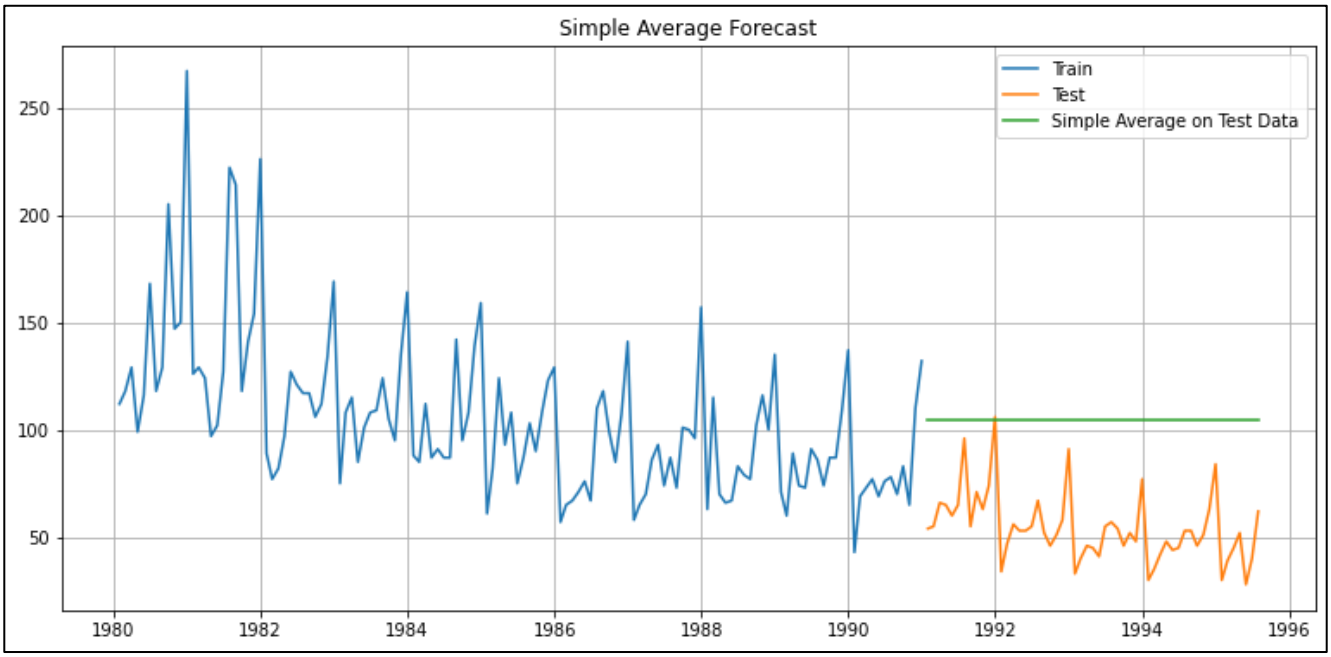


Figure 2. 15: Forecast using Simple Average

- The RMSE is lower than that of naïve forecast model.
- The forecast line in green is again a flat line, discounting trend and seasonality present in our data.
- Since the model is using one static value of the average of training data as forecast, it doesn't seem to be optimum for our data.

2.4.4 Moving Average Model

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 (RMSE). The different intervals we are considering in this case are:

A. 2-point trailing moving average – average of 2 months data
B. 4-point trailing moving average – average of 4 months data
C. 6-point trailing moving average – average of 6 months data
D. 9-point trailing moving average – average of 9 months data

For Moving Average, we are going to average over the entire data. And later split the data into train and test after running the model.

This is how the moving average appear against the original data:

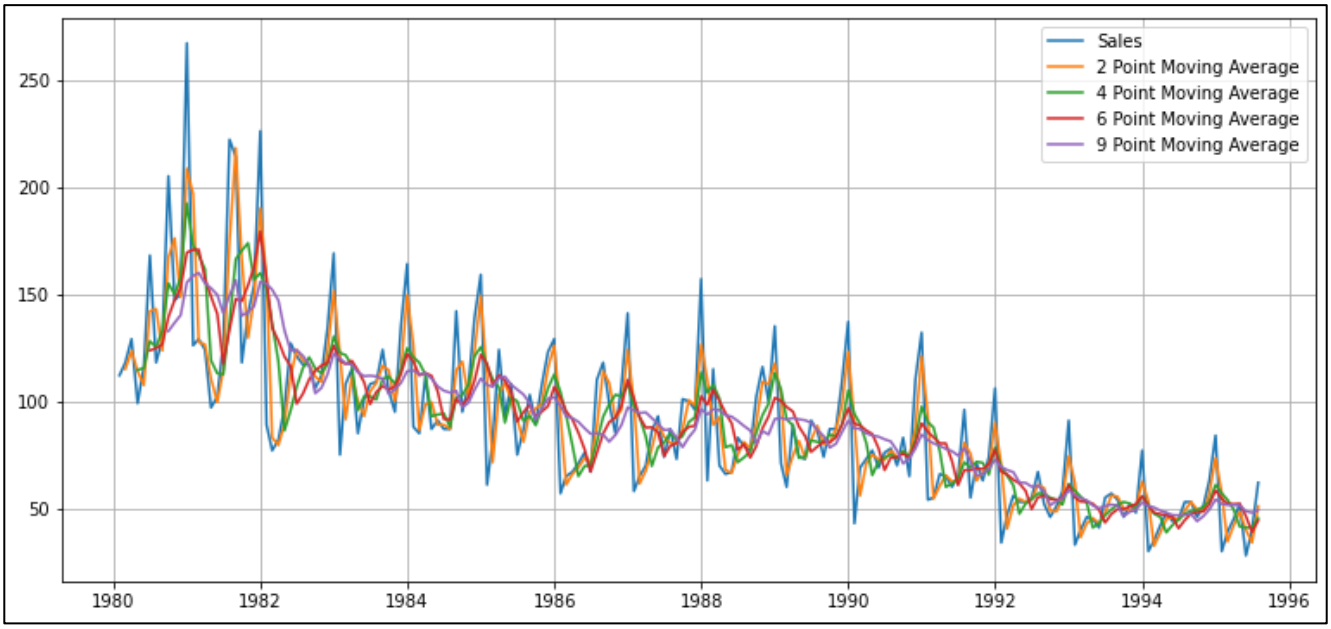


Figure 2. 16: Forecast using Moving Average

After splitting the data into train (1980 – 1990) and test (1991 – 1995), plotting this Time Series:

2-point trailing moving average

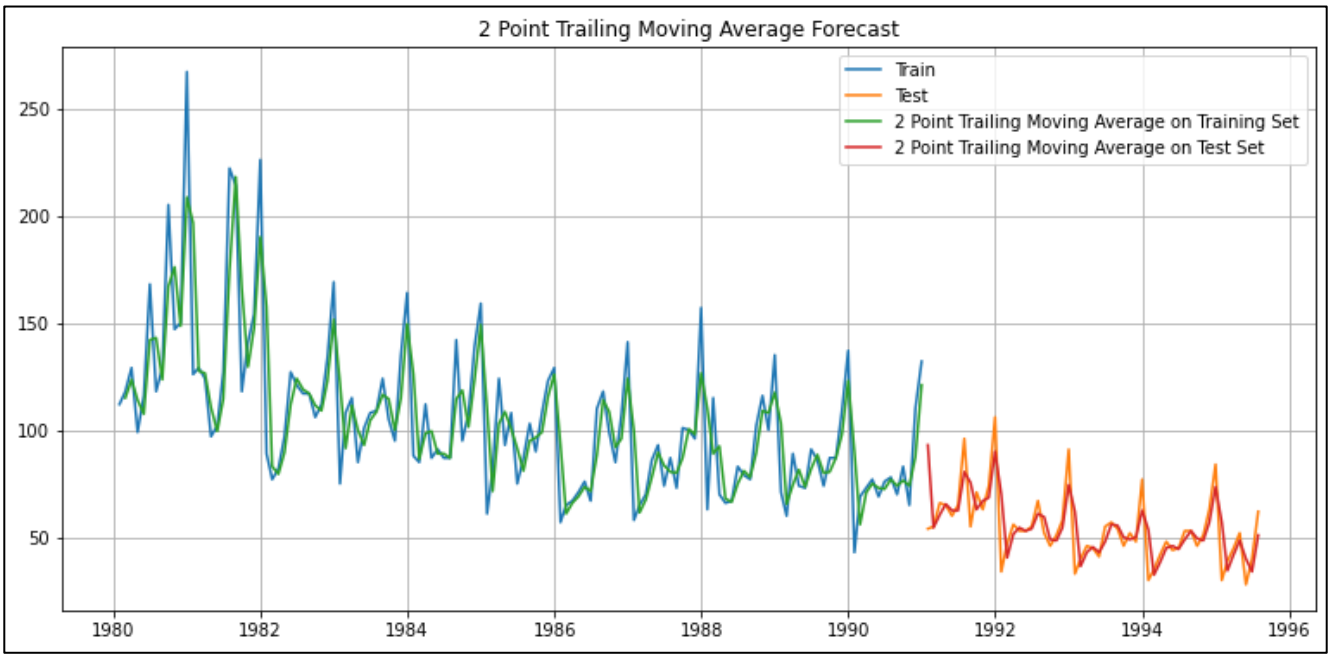


Figure 2. 17: Forecast using 2 Point Moving Average

4-point trailing moving average

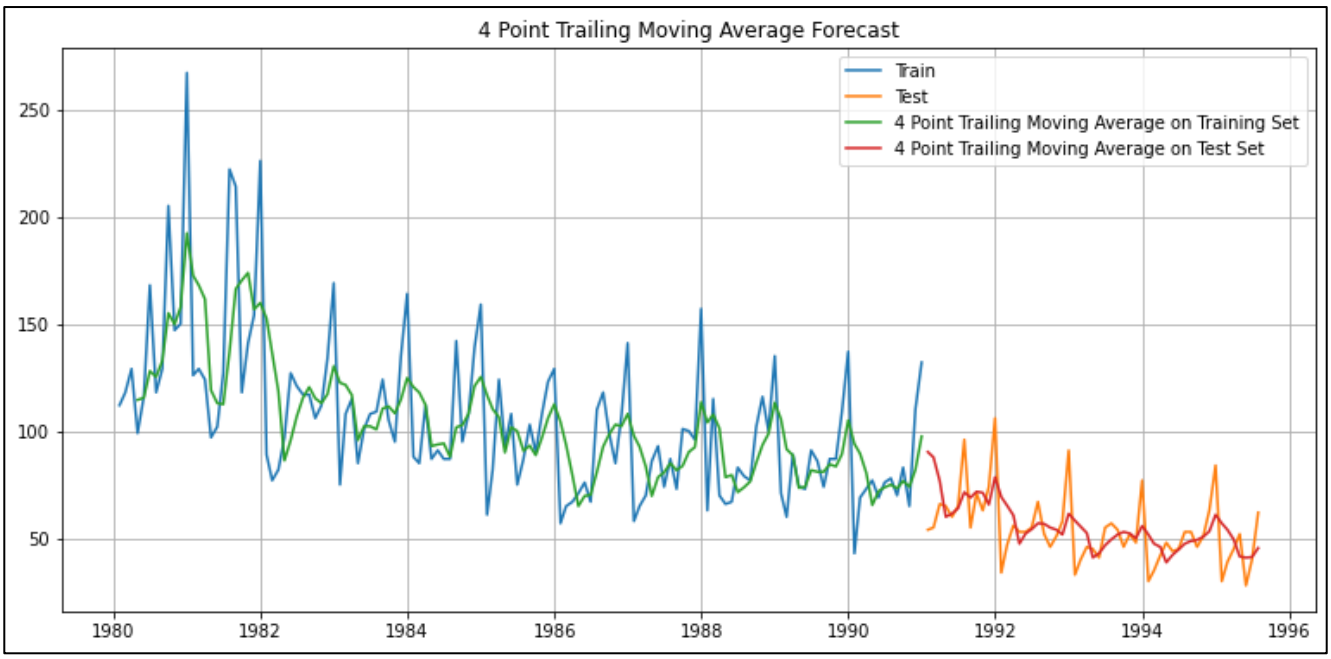


Figure 2. 18: Forecast using 4 Point Moving Average

6-point trailing moving average

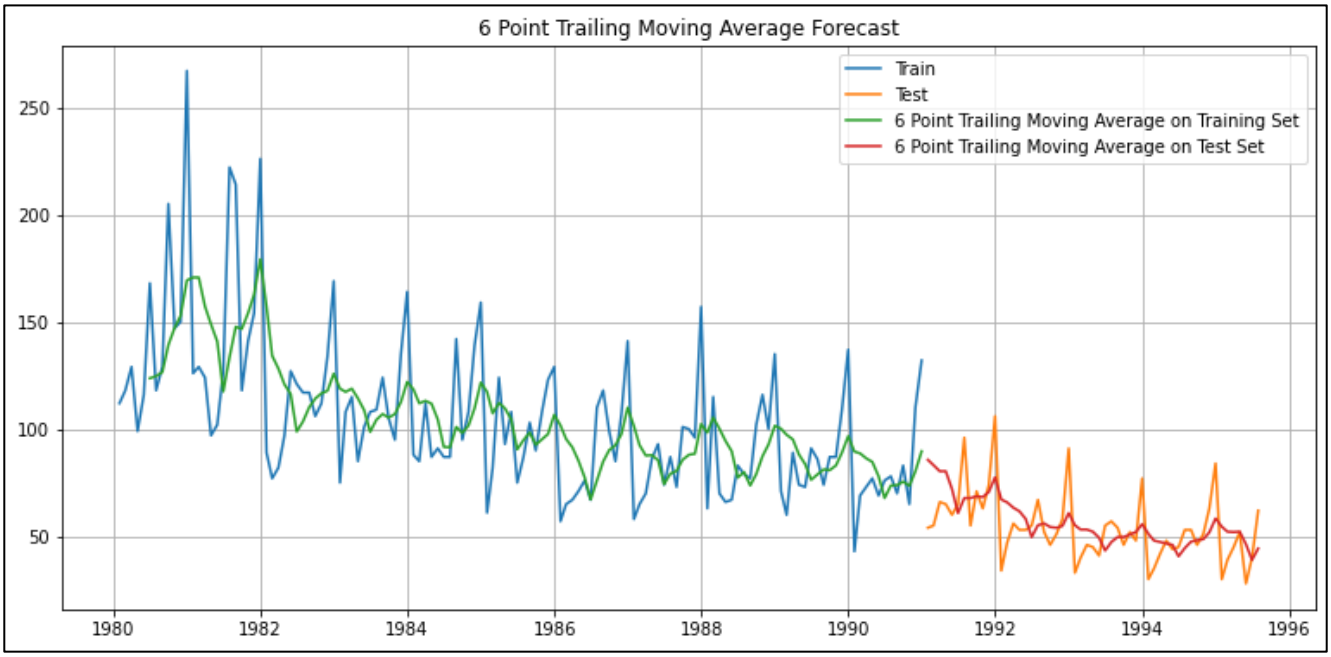


Figure 2. 19: Forecast using 6 Point Moving Average

9-point trailing moving average

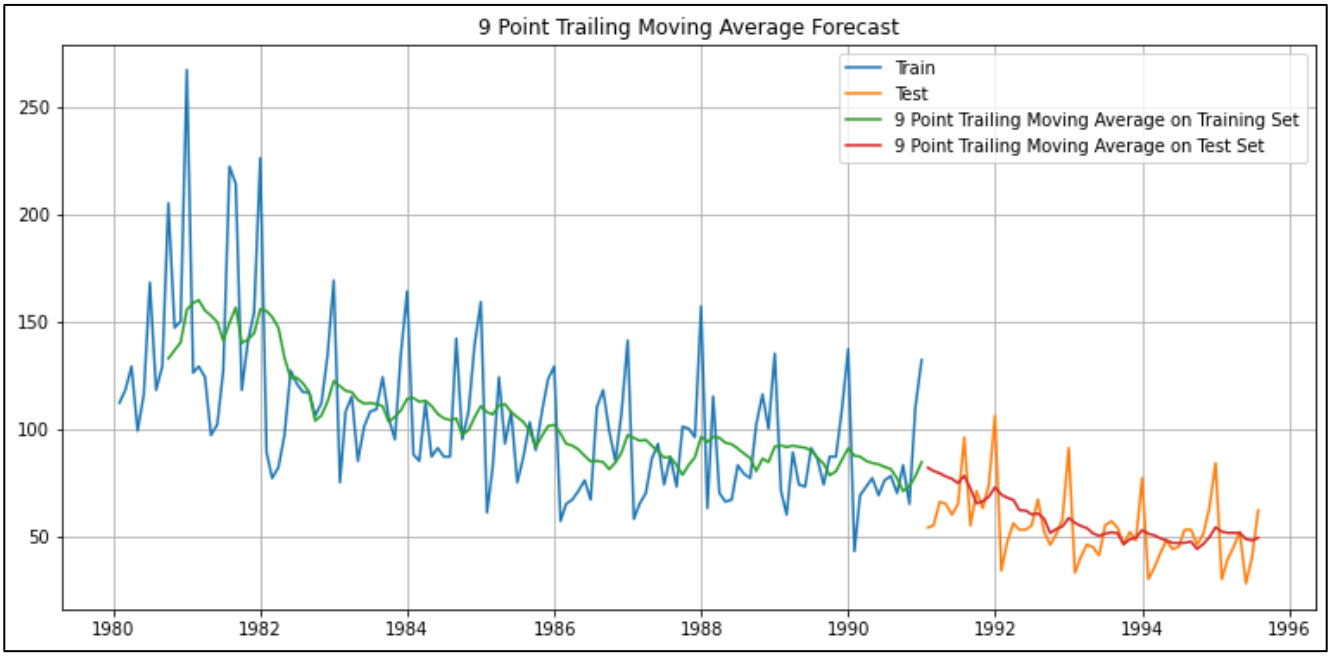


Figure 2. 20: Forecast using 9 Point Moving Average

For Moving Average Forecast on the train data various RMSE values are as follows:

- For 2 point Moving Average Model forecast on the Training Data, **RMSE is 11.551**
- For 4 point Moving Average Model forecast on the Training Data, **RMSE is 14.453**
- For 6 point Moving Average Model forecast on the Training Data, **RMSE is 14.533**
- For 9 point Moving Average Model forecast on the Training Data, **RMSE is 14.738**

- The RMSE is the lowest for 2-point trailing moving average.
- From the graphs also we can observe that 2-point trailing moving average forecast is quite closer to the test data. Trend and seasonality are also captured well by this model.

- Hence, 2-point trailing moving average model is the best model we have so far for forecasting rose wine sales.

Exponential Smoothing Methods

- Exponential smoothing method considers the weighted averages of the past observations.
- There are 3 parameters; out of these 3, one or more parameters control the weighted averages.
    1. Alpha - Level of the time series
    2. Beta – Trend of the time series
    3. Gamma – Seasonality of the time series

2.4.5 Simple Exponential Smoothing Model

Simple exponential smoothing method for forecasting only works when there is no trend or seasonality in the time series. It only accounts for the Alpha (level) of the time series.

Since we have trend and seasonality in our time series, this model is right away not very useful. But let's build and compare the outcomes of this model with the rest of the models.

For Alpha = 0.099 Simple exponential smoothing Forecast on the train data, **RMSE is 36.515**. This is how the forecast appear against actual values:

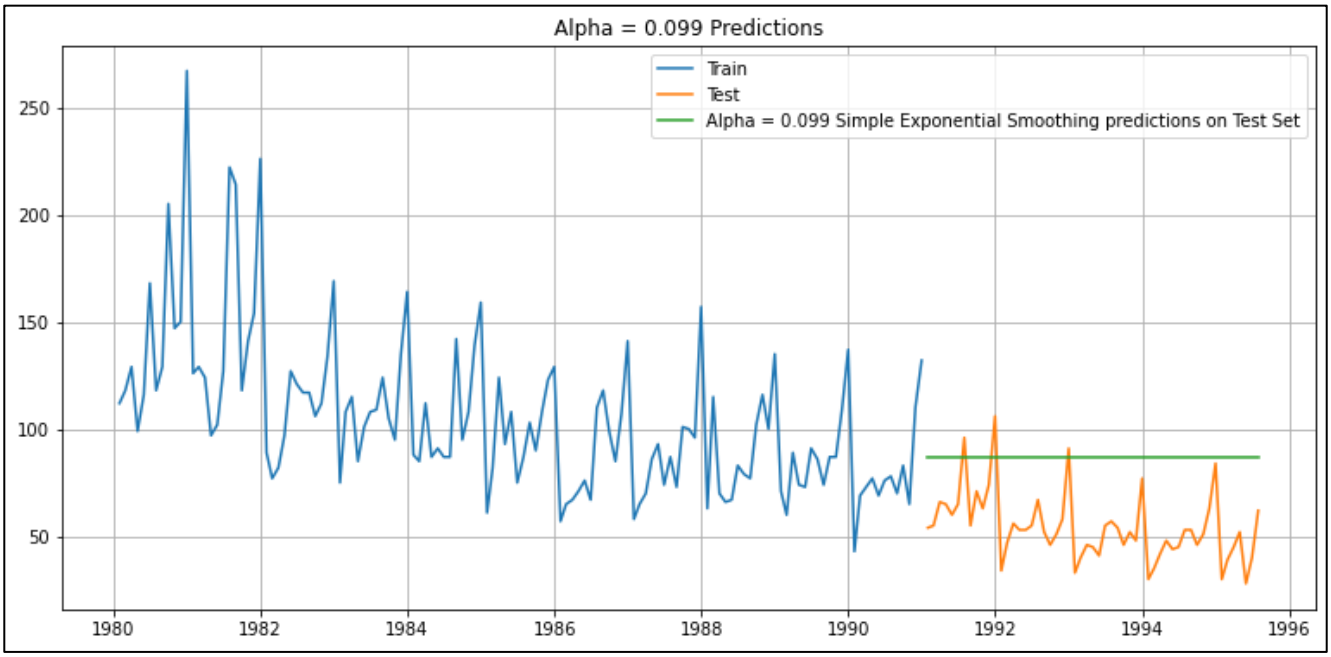


Figure 2. 21: Simple Exponential Smoothing – Alpha 0.099

Setting different alpha values:

- We ran a loop with different alpha values to understand which particular value works best for alpha on the test set.
- The higher the alpha value more weightage is given to the more recent observation. That means, what happened recently will happen again.
- Alpha value of 0.07 returns the lowest **RMSE of 36.154**. So, 0.07 is the optimum alpha value in terms of SES model. Alpha = 0.07 (close to 0) interprets that forecasts are far from the actual data points.
- This is how the forecast appear with Alpha as 0.99 and 0.07, against the actual values:

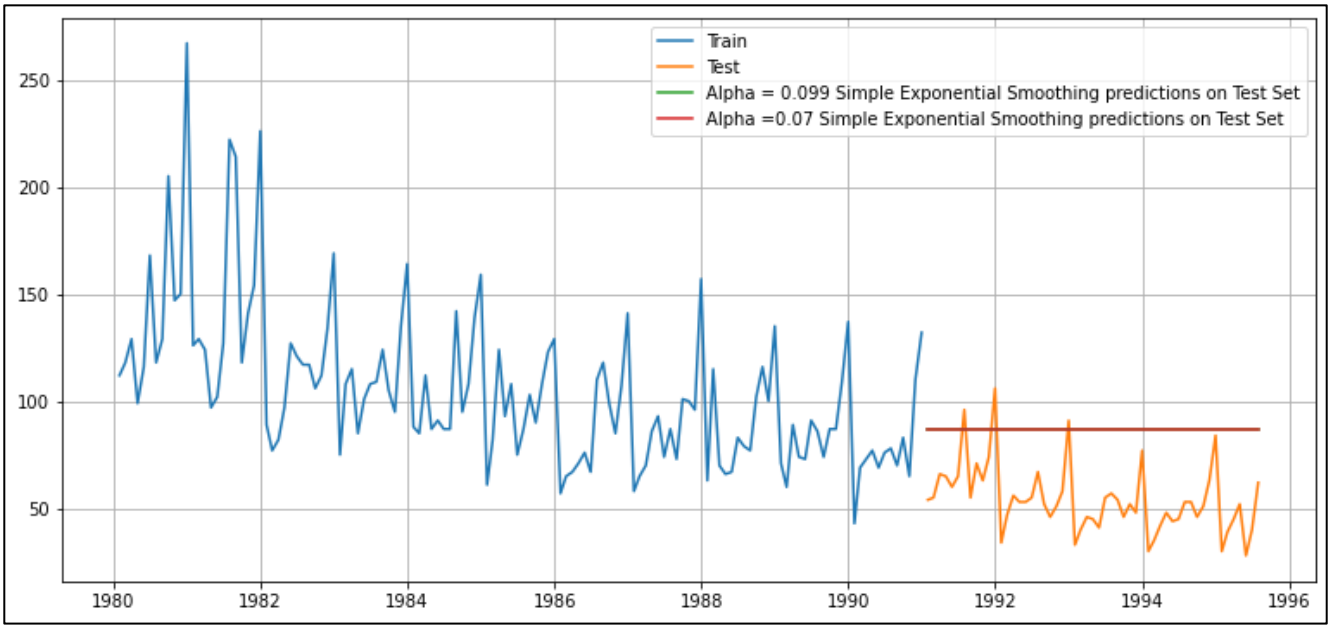


Figure 2. 22: Simple Exponential Smoothing – Alpha 0.099 & 0.07

- The RMSE of Simple Exponential Smoothing method at Alpha = 0.07 is slightly lower than Alpha = 0.099.
- As we can see from the graph, this method is returning a static forecast for different Alpha values. That is because it does not factor trend and seasonality. Hence, it is not optimum for our data.

2.4.6 Double Exponential Smoothing Model

Double exponential smoothing method for forecasting works when there is no seasonality in the time series. It only accounts for the Alpha (level) and Beta (trend) of the time series.

Since we have seasonality also present in our time series, this model may not be very useful. But let's build and compare the outcomes of this model with the rest of the models.

For Alpha = 1.908e-08 (0.0) and Beta = 7.302e-09 (0.00), Double exponential smoothing Forecast on the train data, **RMSE is 15.237**. This is how the forecast appear against actual values:

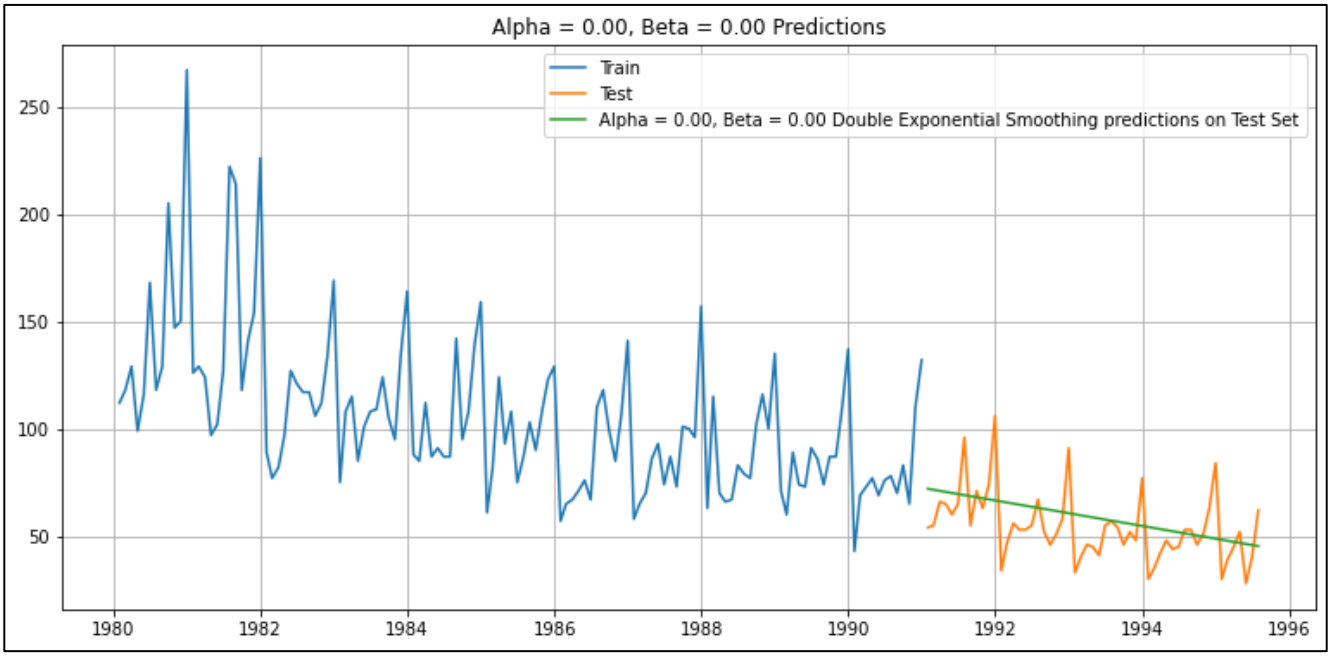


Figure 2. 23: Double Exponential Smoothing – Alpha 0.0 & Beta 0.0

Setting different alpha and beta values:

- We ran a loop with different Alpha and Beta values to understand which particular values work best for alpha and beta on the test set.
- Alpha = 0.1 and Beta = 0.1 return a lower **RMSE of 36.588**. So, 0.1 is the optimum Alpha and Beta values in terms of DES model.
    - Alpha = 0.1 (close to 0) interprets that forecast is far from the actual data points.
    - Beta = 0.1 (close to 0) interprets that forecast is giving more weightage to older trend.
- This is how the forecast appear with Alpha as 0.1 and Beta as 0.1, against the actual values:

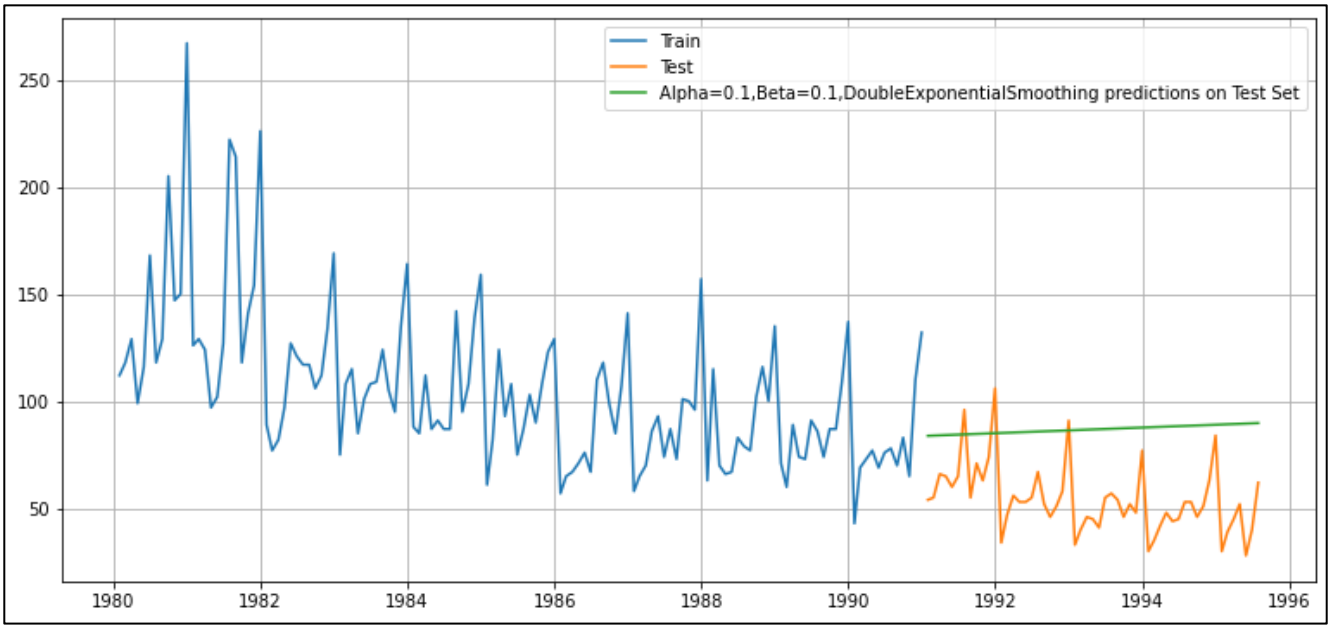


Figure 2. 24: Double Exponential Smoothing – Alpha 0.1 & Beta 0.1

- The RMSE of Double Exponential Smoothing method at optimum Alpha and Beta value of 0.1 is one of the highest among all the models we have applied so far.
- As we can see from the graph, this method is taking trend into consideration but discounting seasonality, which is very crucial in our time series.
- Hence, Double Exponential Smoothing model is not optimum for our data.

2.4.7 Triple Exponential Smoothing Model

Triple exponential smoothing method for forecasting works when there is both, trend and seasonality, present in the time series. It accounts for the Alpha (level), Beta (trend) and Gamma (seasonality) of the time series.

This model can be very useful for our trend and seasonality laced time series. Let's build the model and compare if it works better than the rest of the models.

Since we have seasonality component involved, the model is built using 2 techniques:

3. Additive seasonality
4. Multiplicative seasonality

Additive seasonality

The model is fit using the brute force method to choose the best parameters automatically.

For Alpha = 0.088, Beta = 6.730e-05 (0.0) and Gamma = 0.004, Triple exponential smoothing (additive) forecast on the train data, **RMSE is 14.133**. This is how the forecast appear against actual values:

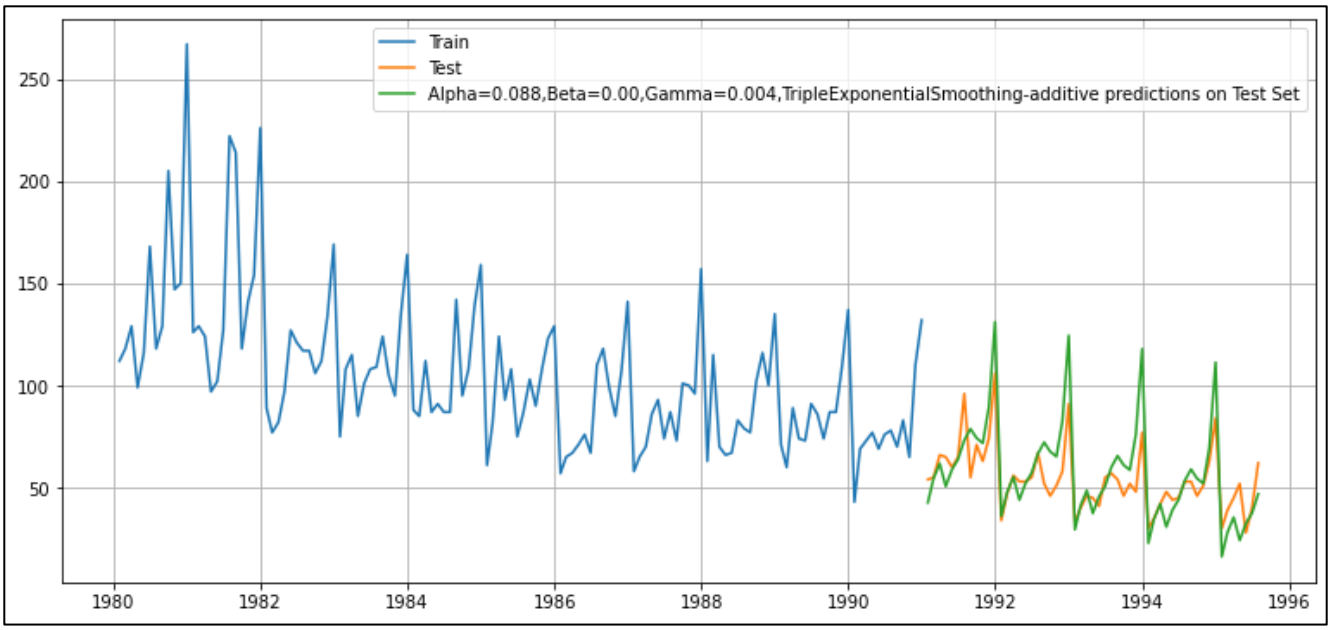


Figure 2. 25: Triple Exponential Smoothing_Additive – Alpha 0.088, Beta 0.0 & Gamma = 0.004

Setting different alpha, beta and gamma values:

- We ran a loop with different Alpha, Beta and Gamma values to understand which particular values work best for alpha, beta and gamma on the test set.
- Alpha = 0.1, Beta = 0.4 and Gamma = 0.3 return a lower **RMSE of 12.096**. So, 0.1 is the optimum Alpha, 0.4 is the optimum value for Beta and 0.3 is the optimum value for Gamma in terms of TES-additive model.
    - Alpha = 0.1 (close to 0) interprets that forecast is far from the actual data points.
    - Beta = 0.4 (close to 0) interprets that forecast is giving more weightage to older trend.
    - Gamma = 0.3 (close to 0) interprets that forecast is giving more weightage to older seasonality.

- This is how the forecast appear with Alpha as 0.1, Beta as 0.4 and Gamma as 0.3, against the actual values:

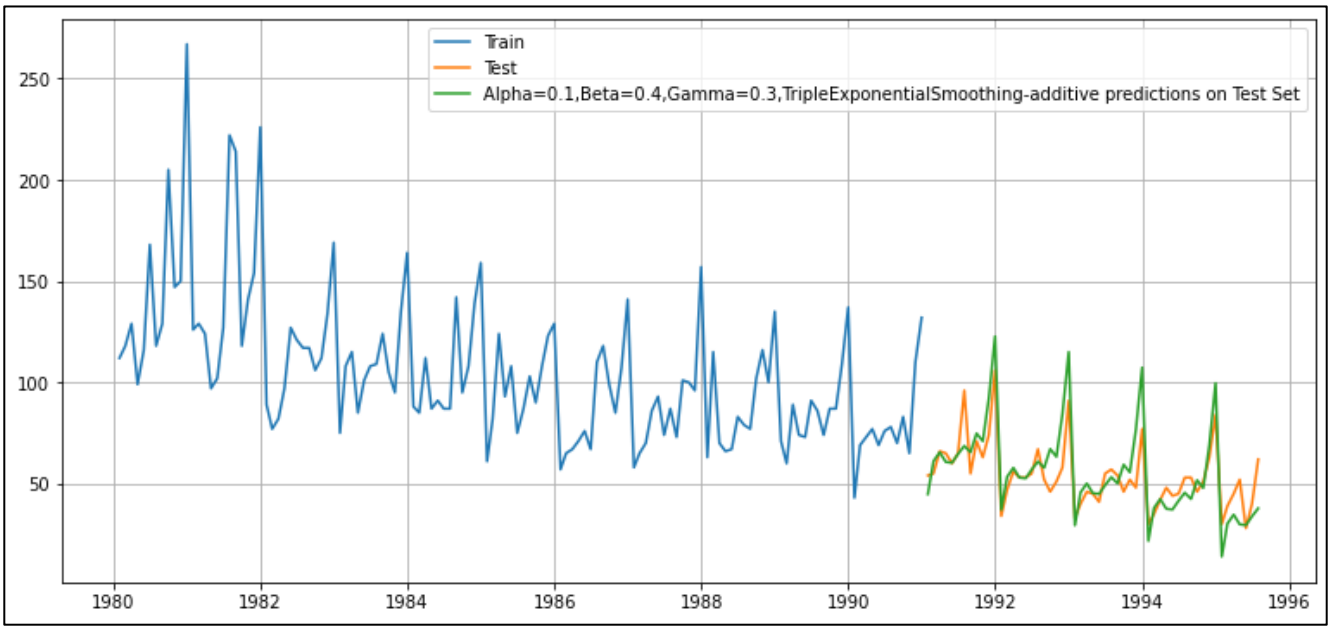


Figure 2. 26: Triple Exponential Smoothing_Additive – Alpha 0.1, Beta 0.4 & Gamma = 0.3

Multiplicative seasonality

The model is fit using the brute force method to choose the best parameters automatically.

For Alpha = 0. 07, Beta = 0.046 and Gamma = 8.35e-07 (0.0), Triple exponential smoothing (multiplicative) forecast on the train data, **RMSE is 19.863**. This is how the forecast appear against actual values:

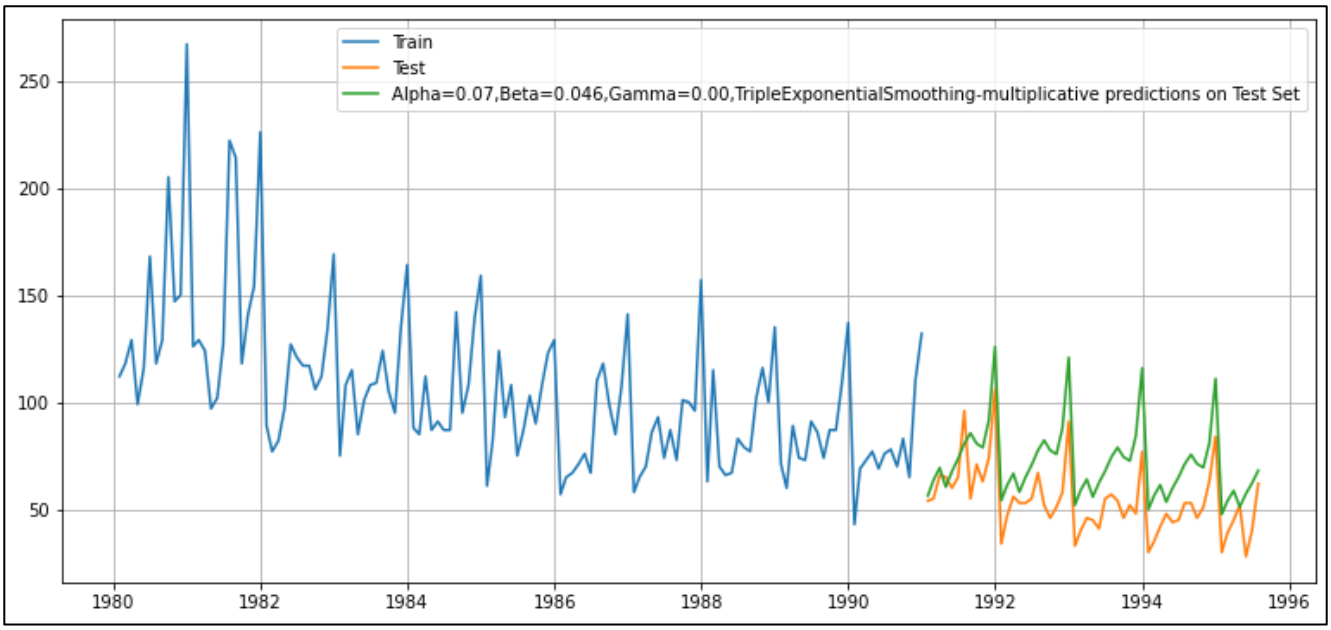


Figure 2. 27: Triple Exponential Smoothing_Multiplicative – Alpha 0.07, Beta 0.046 & Gamma = 0.0

Setting different alpha, beta and gamma values:

- We ran a loop with different Alpha, Beta and Gamma values to understand which particular values work best for alpha, beta and gamma on the test set.

- Alpha = 0.1, Beta = 0.2 and Gamma = 0.1 return the lowest **RMSE of 9.152**. So, 0.1 is the optimum Alpha and Gamma and 0.2 is the optimum Beta value in terms of TES-multiplicative model.
    - Alpha = 0.1 (close to 0) interprets that forecast is far from the actual data points.
    - Beta = 0.2 (close to 0) interprets that forecast is giving more weightage to older trend.
    - Gamma = 0.1 (close to 0) interprets that forecast is giving more weightage to older seasonality.
- This is how the forecast appear with Alpha as 0.1, Beta as 0.2 and Gamma as 0.1, against the actual values:

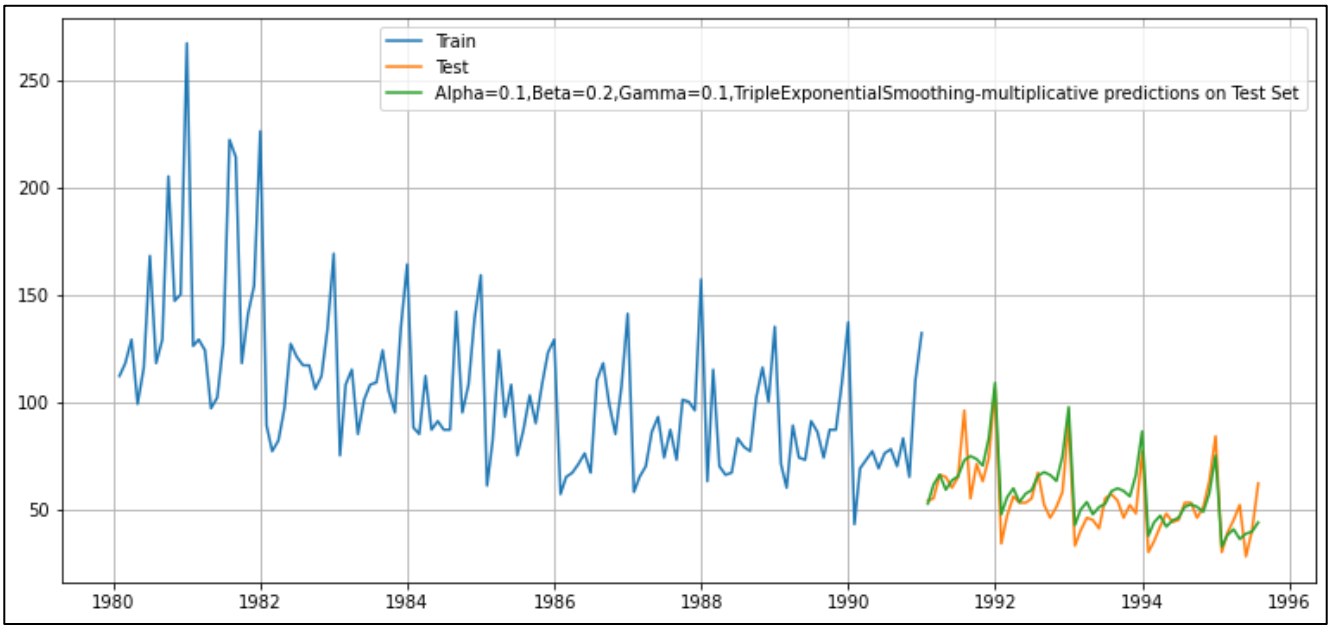
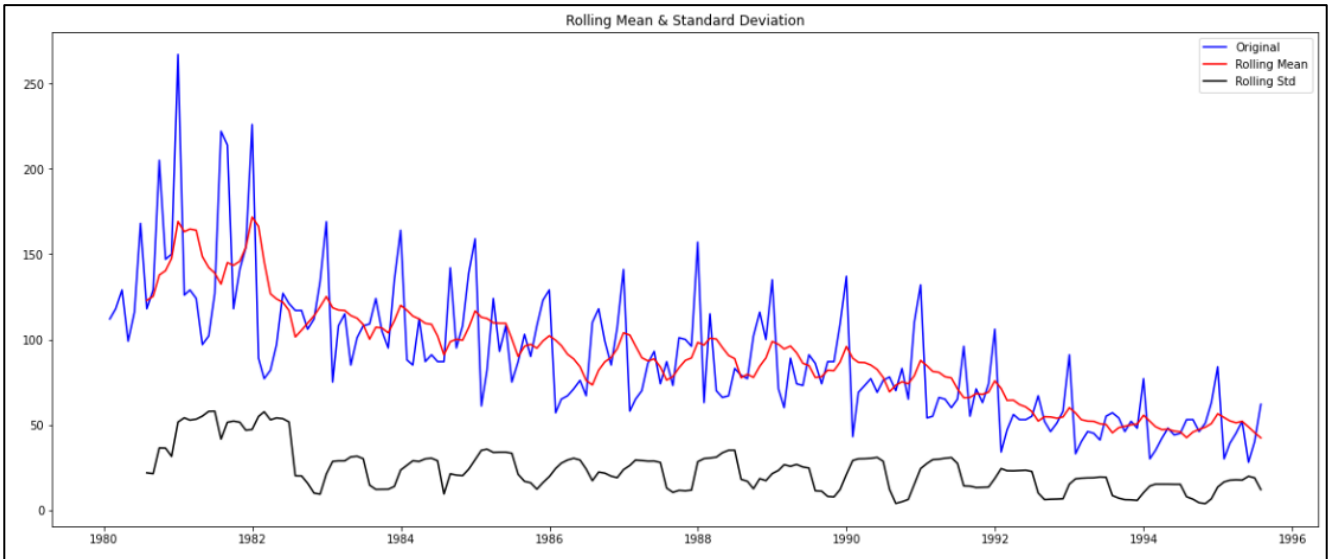


Figure 2. 28: Triple Exponential Smoothing_Multiplicative – Alpha 0.1, Beta 0.2 & Gamma = 0.1

- From the Triple Exponential Smoothing – Multiplicative model we can observe that the forecast is fairly close to the test set values.
- The RMSE is also the lowest among all the models ran so far.
- This could be the most optimum model to forecast for the rose wine sales date.

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

- The Augmented Dickey-Fuller (ADF) test is a unit root test which determines whether there is a unit root in the time series and subsequently whether the series is non-stationary or not.
- The hypothesis in a simple form for the ADF test is:
  - H_0: The Time Series has a unit root and is thus non-stationary.
  - H_1: The Time Series does not have a unit root and is thus stationary.
- We would want the series to be stationary for building ARIMA models and thus we would want the p-value of this test to be less than the alpha value of 0.05.
- First, we check the data stationarity on the entire time series data:

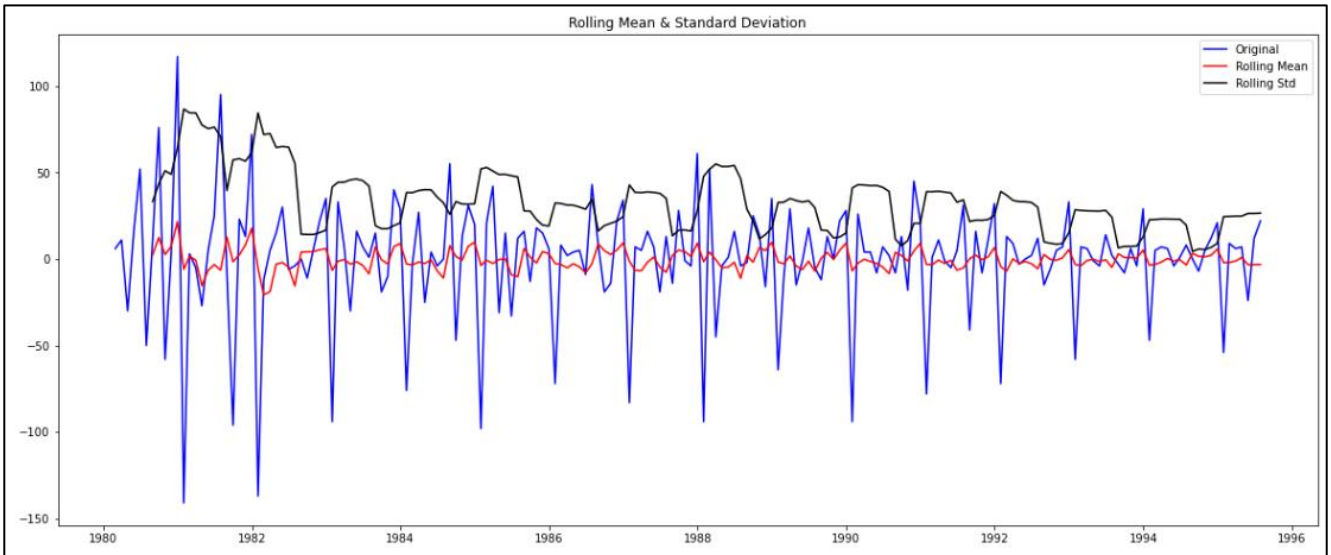



```
Results of Dickey-Fuller Test:
Test Statistic                  -1.893949
p-value                          0.334913
#Lags Used                      13.000000
Number of Observations Used    173.000000
Critical Value (1%)             -3.468726
Critical Value (5%)             -2.878396
Critical Value (10%)            -2.575756
dtype: float64
```

Figure 2. 29: ADF Test – Original Data

- Since p-value 0.3 > 0.05 (failed to reject H_0), we can say that at 5% significance level the time series is non-stationary.

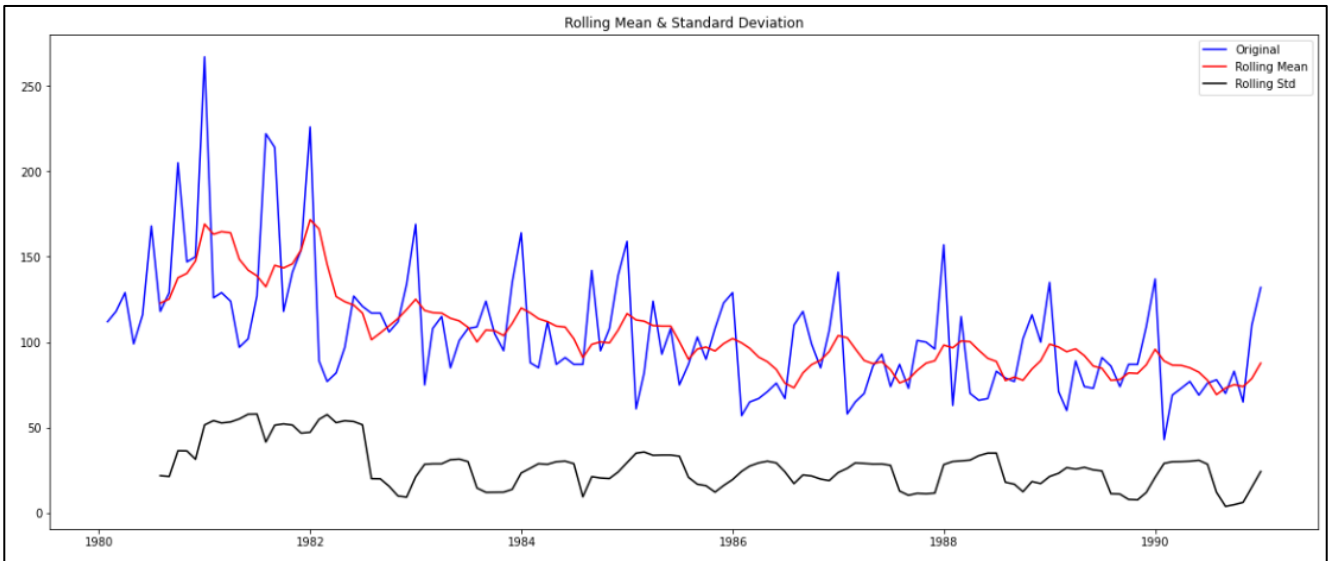
- Let us take a difference of order 1 and check whether the time series becomes stationary or not:




```
Results of Dickey-Fuller Test:
Test Statistic                -8.041739e+00
p-value                        1.839230e-12
#Lags Used                     1.200000e+01
Number of Observations Used    1.730000e+02
Critical Value (1%)           -3.468726e+00
Critical Value (5%)           -2.878396e+00
Critical Value (10%)          -2.575756e+00
dtype: float64
```

Figure 2. 30: ADF Test – Original Data with Differencing

- Now that the p-value 1.83e-12 < 0.05, we reject the null hypothesis. Thus, the time series is now stationary.
- Now, we check the data stationarity on the train data:




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

Figure 2. 31: ADF Test – Train Data

- Since p-value 0.2 > 0.05 (failed to reject H_0), we can say that at 5% significance level the training time series is non-stationary.

- Let us take a difference of order 1 and check whether the time series becomes stationary or not:

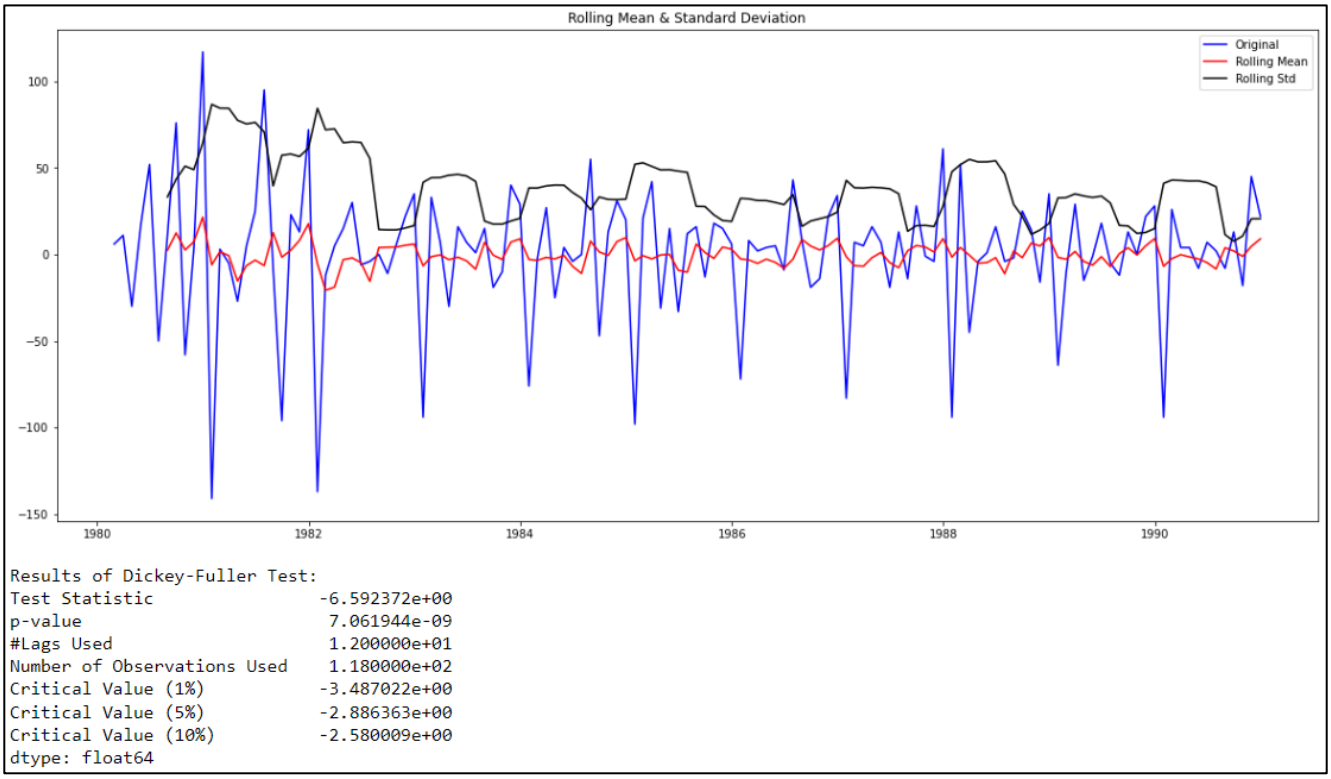


Figure 2. 32: ADF Test – Training Data with Differencing

- Now that the p-value 7.06e-09 < 0.05, we reject the null hypothesis. Thus, the training time series is now stationary as well.

Note: If the series is non-stationary, stationarize the Time Series by taking a difference of the Time Series. Then we can use this particular differenced series to train the ARIMA models. We do not need to worry about stationarity for the Test Data because we are not building any models on the Test Data, we are only evaluating our models over there.

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

2.6.1 Automated ARIMA Model - ARIMA(p,d,q)

- ARIMA – Auto Regressive Integrated Moving Average.
- One of the fundamental assumptions of ARIMA model is that the time series is supposed to be stationary, meaning no having no trend.
- As we checked using ADF test in the previous segment, our time series was non-stationary. But it converted to be stationary at 1 level of differencing. Hence, we can use train data with 1 level of differencing to run ARIMA model.
- An ARIMA model consists of the Auto Regressive (AR) part and the Moving Average (MA) part, after we have made the time series stationary.
- The AR is also denoted as 'q'; MA is also denoted as 'p'; and differencing is also denoted as 'd'.
- To find the minimum Akaike Information Criteria (AIC) value we ran different combination of 'pdq':
    - Where, value of 'p' and 'q' ranges from 0 to 3.
    - 'd' = 1 remains constant as we made the time series stationary at 1 level of differencing.
- As we can see from the below table, the lowest AIC value is achieved at (p,d,q) as (2,1,3).

	param	AIC
11	(2, 1, 3)	1274.695319
15	(3, 1, 3)	1278.654399
2	(0, 1, 2)	1279.671529
6	(1, 1, 2)	1279.870723
3	(0, 1, 3)	1280.545376
5	(1, 1, 1)	1280.57423
9	(2, 1, 1)	1281.507862
10	(2, 1, 2)	1281.870722
7	(1, 1, 3)	1281.870722
1	(0, 1, 1)	1282.309832
13	(3, 1, 1)	1282.419278
14	(3, 1, 2)	1283.720741
12	(3, 1, 0)	1297.481092
8	(2, 1, 0)	1298.611034
4	(1, 1, 0)	1317.350311
0	(0, 1, 0)	1333.154673

Table 2. 5: AIC - ARIMA

- Here is the ARIMA model with automated values of (p,d,q):

SARIMAX Results

Dep. Variable:	Rose	No. Observations:	132
Model:	ARIMA(2, 1, 3)	Log Likelihood	-631.348
Date:	Wed, 14 Dec 2022	AIC	1274.695
Time:	23:53:34	BIC	1291.947
Sample:	01-31-1980	HQIC	1281.705
	- 12-31-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-1.6780	0.084	-20.029	0.000	-1.842	-1.514
ar.L2	-0.7287	0.084	-8.697	0.000	-0.893	-0.565
ma.L1	1.0447	0.616	1.695	0.090	-0.163	2.253
ma.L2	-0.7716	0.132	-5.856	0.000	-1.030	-0.513
ma.L3	-0.9044	0.558	-1.620	0.105	-1.999	0.190
sigma2	858.9120	517.873	1.659	0.097	-156.100	1873.924

Ljung-Box (L1) (Q):	0.02	Jarque-Bera (JB):	24.43
Prob(Q):	0.88	Prob(JB):	0.00
Heteroskedasticity (H):	0.40	Skew:	0.71
Prob(H) (two-sided):	0.00	Kurtosis:	4.57

- From this model we can infer that, we are making more errors in MA version, since the value of 'str err' is more for MA.
- AR L1 is the most significant variable, as per the value of coef.
- Predict on the Test Set using this model and evaluate the forecast. For Automated_ARIMA(2,1,3) forecast on the train data, **RMSE is 36.536**. This is how the forecast appear against actual values:

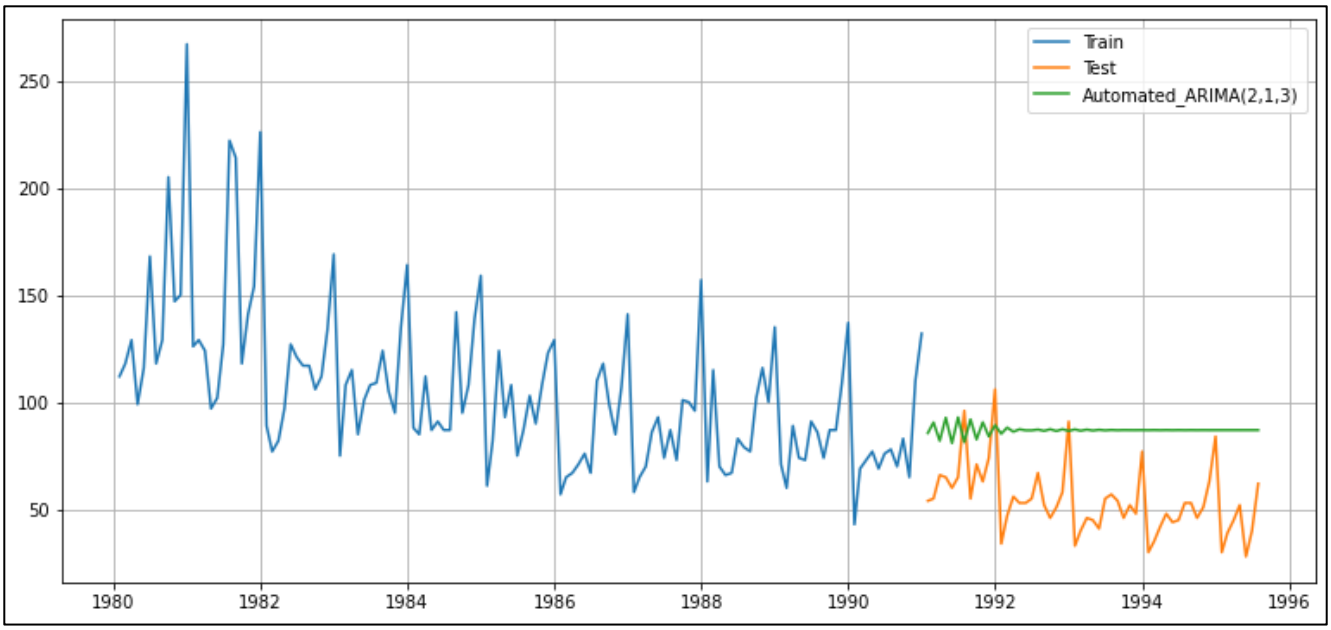


Figure 2. 33: Automated_ARIMA(2,1,3)

- From the Automated_ARIMA model we can observe that the forecast is considering very less seasonality. And trend component is also quite constant towards the end.
- The RMSE is also not the lowest among all the models ran so far.
- This not optimum model to forecast for the rose wine sales date.

2.6.2 Automated SARIMA Model – SARIMA(p,d,q) (P,D,Q,F)

- SARIMA – Seasonal Auto Regressive Integrated Moving Average.
- For a Seasonal ARIMA / SARIMA model, we have to take care of 4 parameters such as AR (p), MA (q), seasonal AR (P) and seasonal MA (Q).
- With correct differencing (d) and seasonal differencing (D).
- Also, seasonal frequency (F) indicates the seasonal effects over a particular period.
- As we checked using ADF test in the previous segment, our time series was non-stationary. But it converted to be stationary at 1 level of differencing. Hence, we can use train data with 1 level of differencing to run SARIMA model.
- To find the minimum Akaike Information Criteria (AIC) value we ran different combination of 'pdq' and 'PDQF':
    - Where, value of 'p', 'P', 'q' and 'Q' ranges from 0 to 3.
    - 'd' = 1 remains constant as we made the time series stationary at 1 level of differencing.
    - 'D' = 0, as we have already stationarized the data once.
    - 'F' = 9, as from the below Autocorrelation plot, we can observe a pattern at each 9th occurrence.
- As we can see from the below table, the lowest AIC value is achieved at (p,d,q)(P,D,Q,F) as (3, 1, 3) (3, 0, 3, 9).

	param	seasonal	AIC
255	(3, 1, 3)	(3, 0, 3, 9)	914.734527
127	(1, 1, 3)	(3, 0, 3, 9)	917.317497
191	(2, 1, 3)	(3, 0, 3, 9)	919.150305
119	(1, 1, 3)	(1, 0, 3, 9)	926.35959
63	(0, 1, 3)	(3, 0, 3, 9)	926.727658

Table 2. 6: AIC - SARIMA

- Here is the SARIMA model with automated values of (p,d,q)(P,D,Q,F):

SARIMAX Results

Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(3, 1, 3)x(3, 0, 3, 9)	Log Likelihood	-444.367
Date:	Tue, 13 Dec 2022	AIC	914.735
Time:	16:03:52	BIC	948.602
Sample:	0 - 132	HQIC	928.441
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.0165	0.113	0.146	0.884	-0.204	0.237
ar.L2	0.6847	0.094	7.316	0.000	0.501	0.868
ar.L3	-0.3016	0.131	-2.308	0.021	-0.558	-0.046
ma.L1	-0.9992	173.643	-0.006	0.995	-341.333	339.335
ma.L2	-0.9992	110.132	-0.009	0.993	-216.853	214.855
ma.L3	1.0001	209.396	0.005	0.996	-409.408	411.408
ar.S.L9	0.7445	0.075	9.899	0.000	0.597	0.892
ar.S.L18	-0.6624	0.069	-9.594	0.000	-0.798	-0.527
ar.S.L27	0.5522	0.057	9.613	0.000	0.440	0.665
ma.S.L9	-0.5894	0.177	-3.329	0.001	-0.936	-0.242
ma.S.L18	0.4654	0.187	2.495	0.013	0.100	0.831
ma.S.L27	-0.4143	0.162	-2.554	0.011	-0.732	-0.096
sigma2	337.6596	7.07e+04	0.005	0.996	-1.38e+05	1.39e+05

Ljung-Box (L1) (Q):	0.64	Jarque-Bera (JB):	8.43
Prob(Q):	0.42	Prob(JB):	0.01
Heteroskedasticity (H):	0.59	Skew:	0.58
Prob(H) (two-sided):	0.13	Kurtosis:	3.83

- From this model we can infer that, AR L1 is the least significant and MA L3 is the most significant. variable.
- We ran the diagnostic plot:

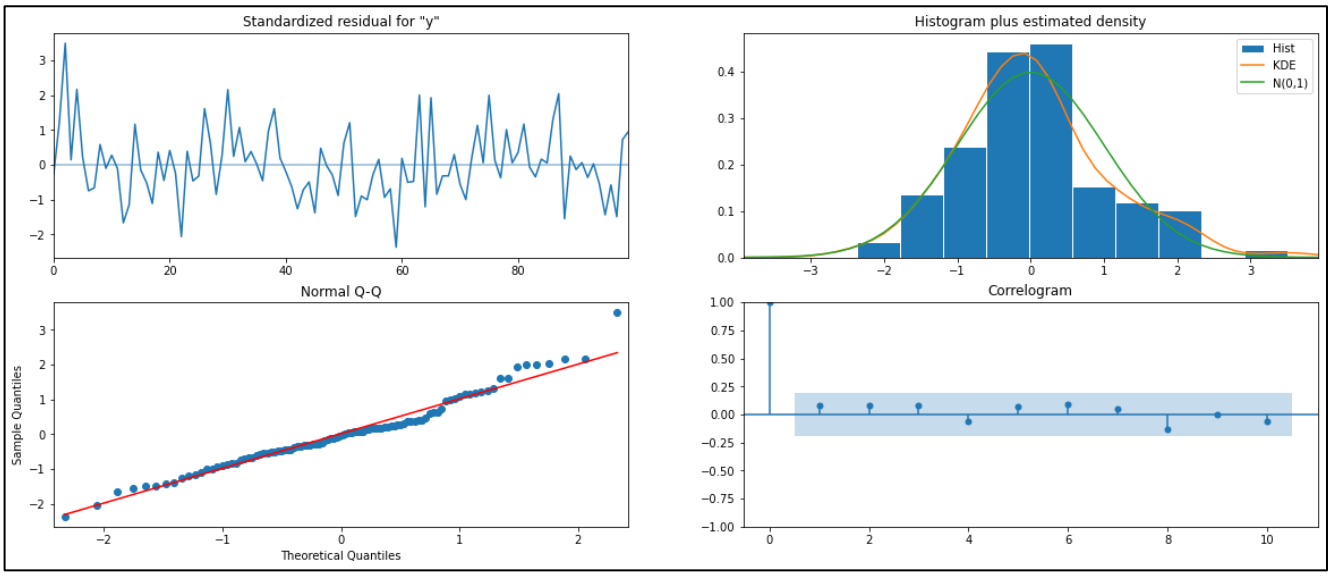


Figure 2. 34: Automated SARIMA Mode Diagnostic Plot

- In the Normal Q-Q plot, forecasted values (blue dots) are fairly close to the actual values.
- In Correlogram, all the data points are within the significance zone. This indicates that we have considered adequate amount of correlations. Hence, significantly model has performed well, using the optimum value of p and q.
- Predict on the Test Set using this model and evaluate the forecast. For Automated_SARIMA (3, 1, 3)(3, 0, 3, 9) forecast on the train data, **RMSE is 30.894**. This is how the forecast appear against actual values:

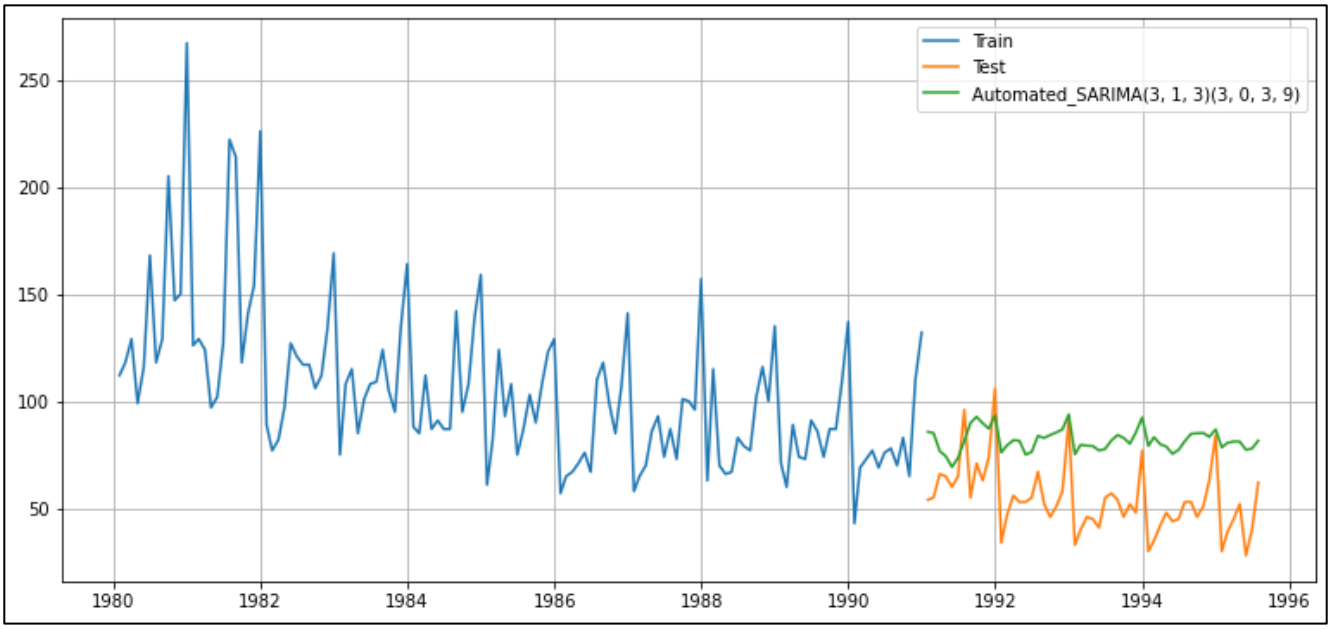


Figure 2. 35: Automated_SARIMA(3, 1, 3)(3, 0, 3, 9)

- From the Automated_SARIMA model we can observe that the forecast has accounted for trend very well but seasonality is not well defined as compared to the test data.
- The RMSE is low as compared to automate ARIMA model. Simply because SARIMA is 'Seasonal' ARIMA, better fit for time series with seasonality and trend component.

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

2.7.1 Manual ARIMA Model - ARIMA(p,d,q)

- For manual ARIMA model, we set the value of AR (p) and MA (q) using Autocorrelation function (ACF) and Partial Autocorrelation function (PACF) plots, respectively.
- These plots are created using train data of level 1 difference (d = 1).

ACF Plot

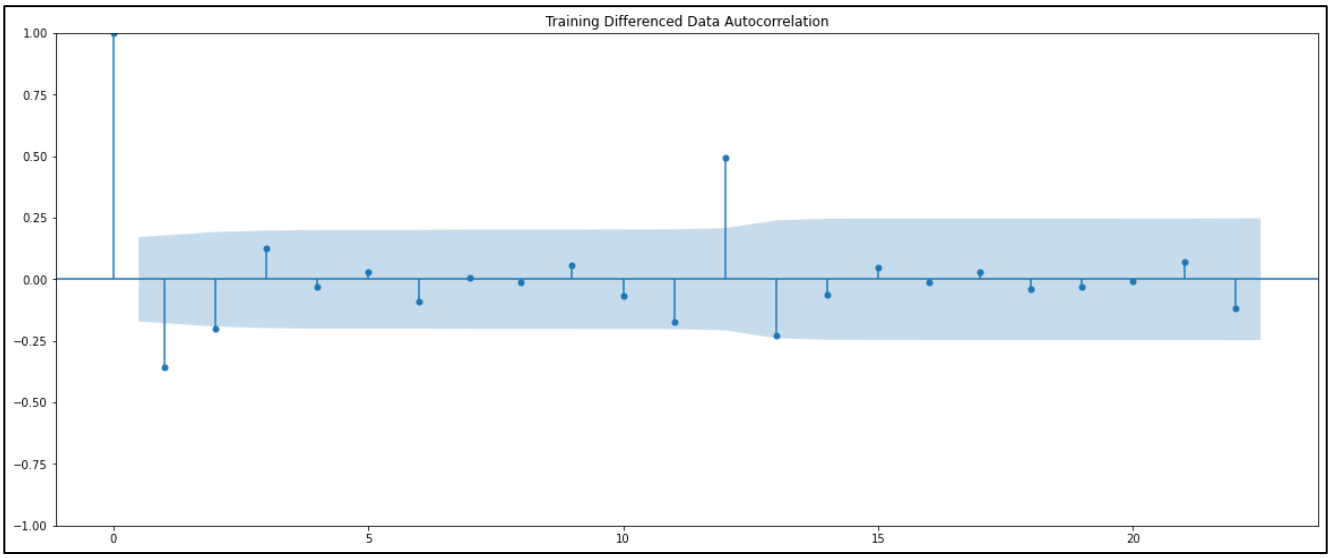


Figure 2. 36: ACF Plot

PACF Plot

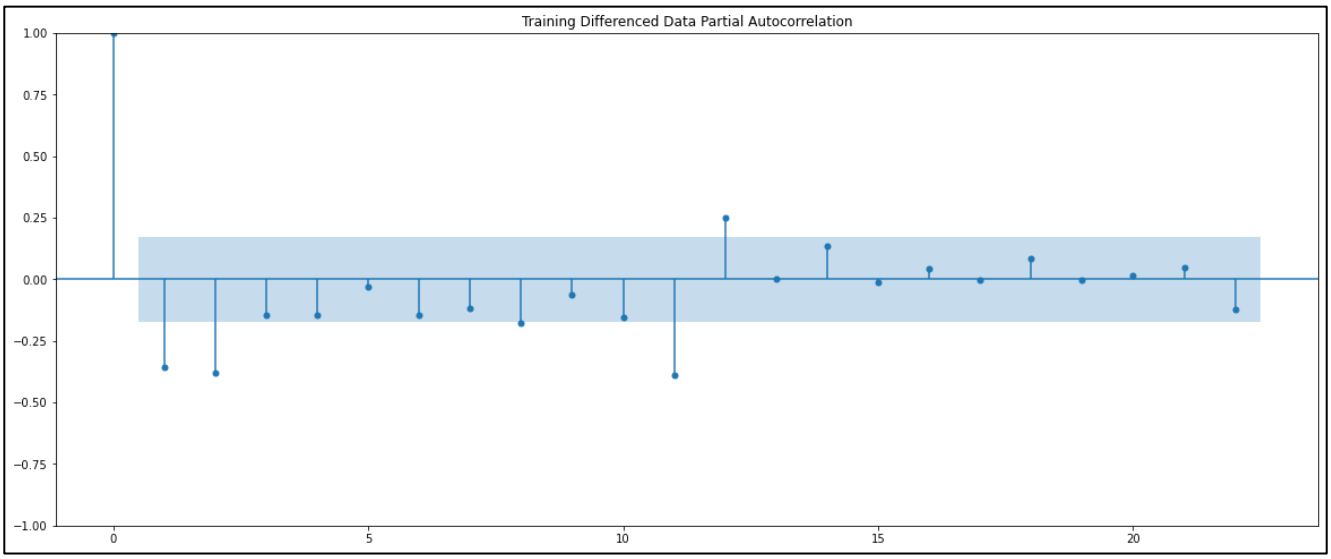


Figure 2. 37: PACF Plot

- Here, we have taken alpha=0.05.
- The Auto-Regressive parameter in an ARIMA model is 'p' which comes from the significant lag before which the PACF plot cuts-off to 2.
- The Moving-Average parameter in an ARIMA model is 'q' which comes from the significant lag before the ACF plot cuts-off to 2.
- By looking at the above plots, we can say that both the PACF and ACF plot cuts-off at lag 2.

- Here is the manual ARIMA model with values of (2,1,2):

SARIMAX Results

Dep. Variable:	Rose	No. Observations:	132
Model:	ARIMA(2, 1, 2)	Log Likelihood	-635.935
Date:	Thu, 15 Dec 2022	AIC	1281.871
Time:	00:10:50	BIC	1296.247
Sample:	01-31-1980 - 12-31-1990	HQIC	1287.712
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

- MA L2 is the most significant and AR L2 is the least significant variable as per coef values.
- Predict on the Test Set using this model and evaluate the forecast. For Automated_ARIMA(2,1,2) forecast on the train data, **RMSE is 36.590**. This is how the forecast appear against actual values:

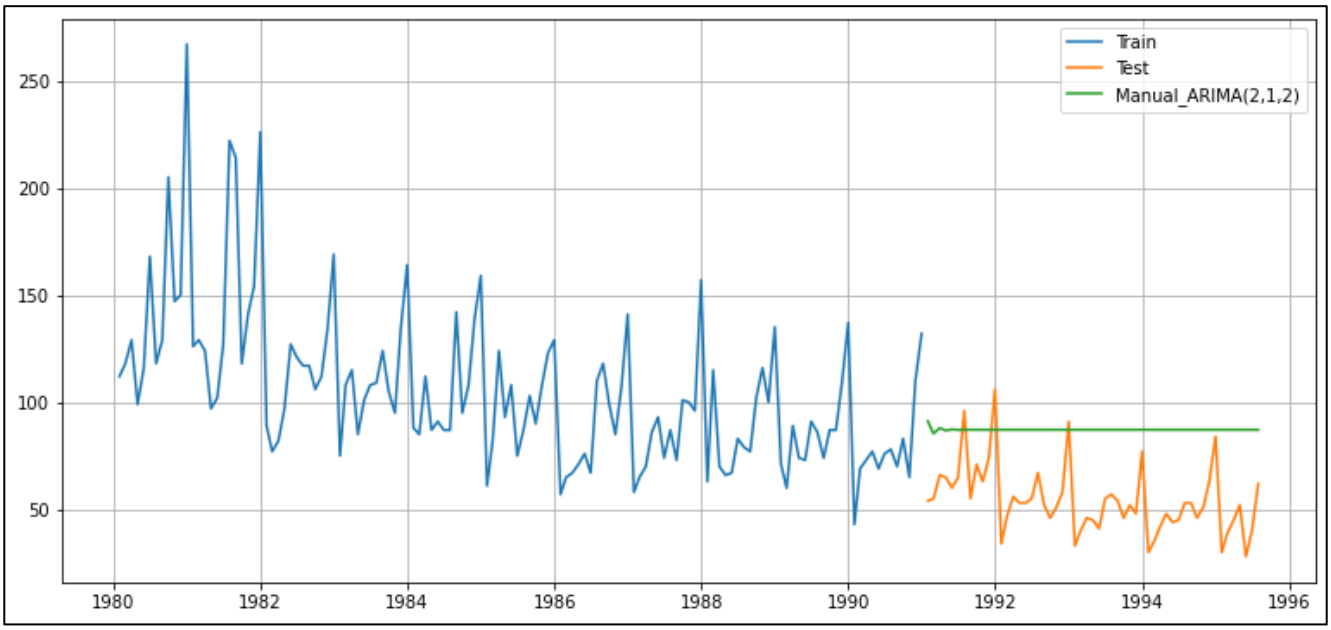


Figure 2. 38: Manual_ARIMA(2,1,2)

- From the Manual_ARIMA model we can observe that the forecast is not considering seasonality and trend. Giving a static forecast.
- The RMSE is also not the lowest among all the models ran so far.
- This not optimum model to forecast for the rose wine sales date.

2.7.2 Manual SARIMA Model - SARIMA(p,d,q) (P,D,Q,F)

- For manual SARIMA model, we have taken alpha=0.05.
- We are going to take the seasonal period as 9. We are taking values of AR (p) and MA (q) to be 2 and differencing (d) to be 1, as the parameters same as the manual_ARIMA model.

ACF Plot – original training data

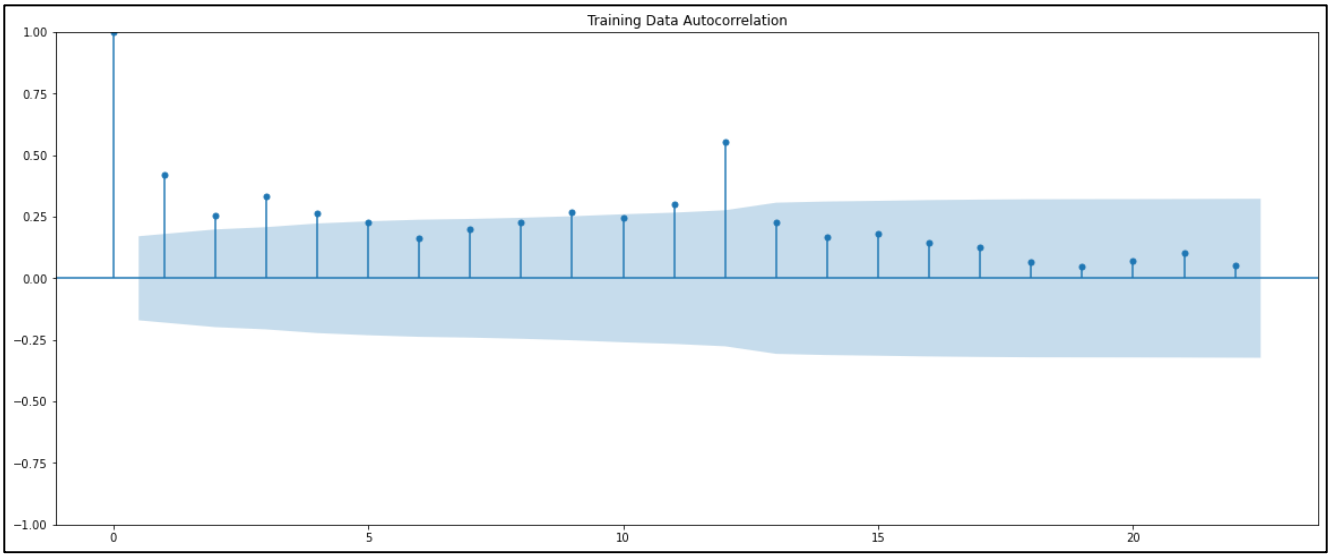


Figure 2. 39: ACF Plot – original training data

PACF Plot – original training data

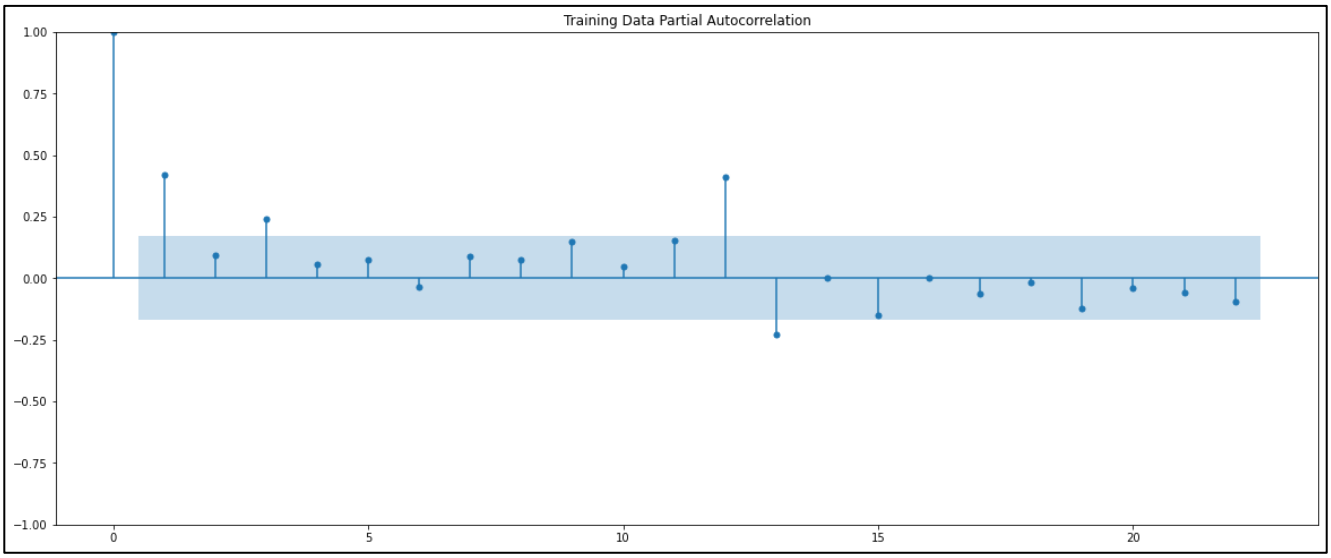


Figure 2. 40: PACF Plot – original training data

- The Auto-Regressive parameter in a SARIMA model is 'P' which comes from the significant lag after which the PACF plot cuts-off to 1.
- The Moving-Average parameter in a SARIMA model is 'Q' which comes from the significant lag after which the ACF plot cuts-off to 5.
- To determine the value of 'P' and 'Q', non-stationarized data is considered. Because the data stationarizing was already performed for the value of 'p' and 'q'.
- Hence, the value of 'D' remains constant as 0.

- 'F' = 9, as from the Autocorrelation plot, we can observe a pattern at each 9th occurrence.
- Here is manual SARIMA model with values of (2,1,2) (1, 0, 5, 9):

SARIMAX Results

Dep. Variable:	Rose	No. Observations:	132
Model:	SARIMAX(2, 1, 2)x(1, 0, [1, 2, 3, 4, 5], 9)	Log Likelihood	-368.914
Date:	Thu, 15 Dec 2022	AIC	759.828
Time:	00:17:55	BIC	786.435
Sample:	01-31-1980 - 12-31-1990	HQIC	770.517
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.9608	0.142	6.754	0.000	0.682	1.240
ar.L2	-0.2311	0.123	-1.871	0.061	-0.473	0.011
ma.L1	-1.9971	33.667	-0.059	0.953	-67.984	63.990
ma.L2	0.9975	33.572	0.030	0.976	-64.802	66.797
ar.S.L9	0.6883	0.079	8.725	0.000	0.534	0.843
ma.S.L9	-0.7418	48.049	-0.015	0.988	-94.917	93.433
ma.S.L18	-0.2911	40.669	-0.007	0.994	-80.001	79.419
ma.S.L27	0.1589	50.879	0.003	0.998	-99.561	99.879
ma.S.L36	1.2252	103.581	0.012	0.991	-201.791	204.241
ma.S.L45	-0.4980	32.182	-0.015	0.988	-63.573	62.577
sigma2	157.4324	1.16e+04	0.014	0.989	-2.25e+04	2.28e+04

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	4.14
Prob(Q):	0.99	Prob(JB):	0.13
Heteroskedasticity (H):	0.86	Skew:	0.53
Prob(H) (two-sided):	0.70	Kurtosis:	2.73

- From this model we can infer that, MA L1 is the most significant and MA.S.L27 is the least significant variable.
- We ran the diagnostic plot:

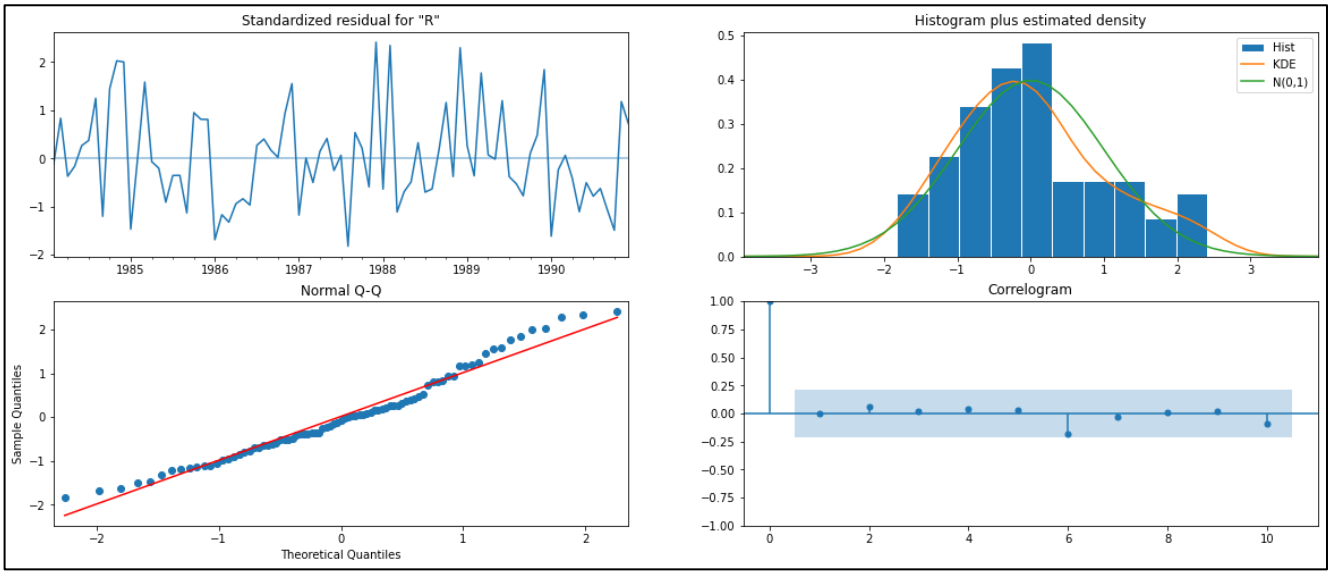


Figure 2. 41: Manual SARIMA Mode Diagnostic Plot

- In the Normal Q-Q plot, forecasted values (blue dots) are falling a little far from the actual values.
- In Correlogram, not all the data points are within the significance zone, data point at 6 is breaching the significance level just a bit. This indicates that manual SARIMA model did not consider the adequate amount of correlation. Hence, significantly model has not performed well, using the optimum value of parameters.
- Predict on the Test Set using this model and evaluate the forecast. For Manual_SARIMA (2, 1, 2)(1, 0, 5, 9) forecast on the train data, **RMSE is 34.210**. This is how the forecast appear against actual values:

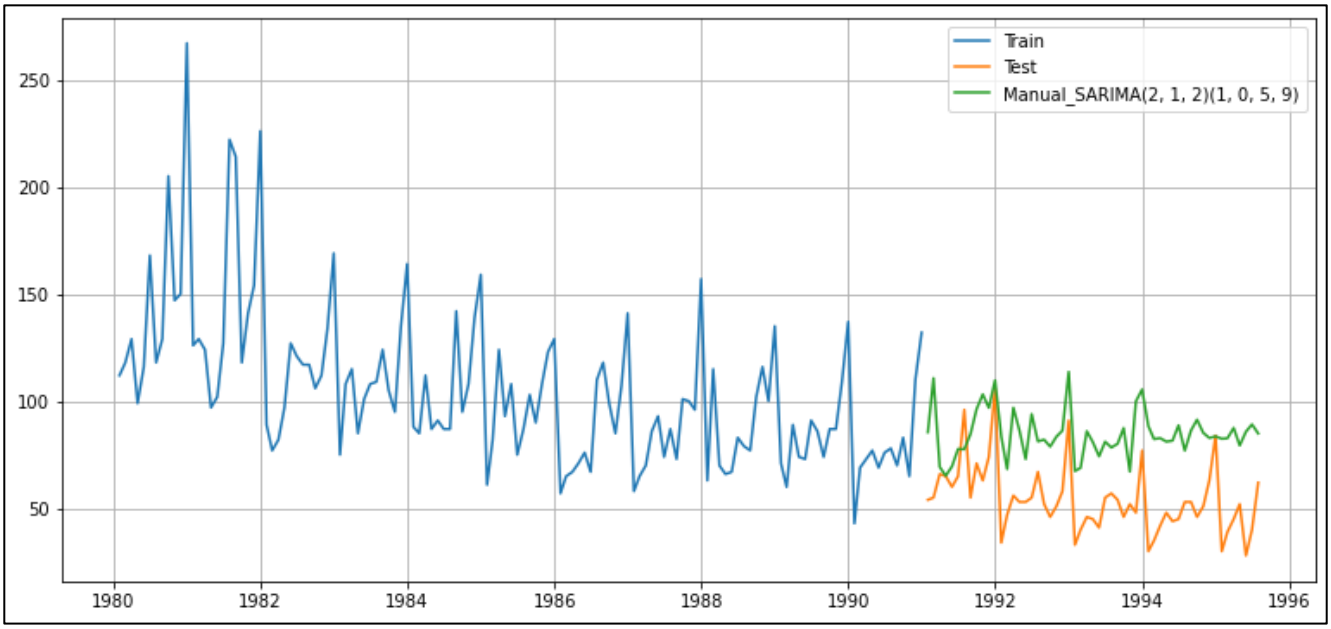


Figure 2. 42: Manual_SARIMA (2, 1, 2)(1, 0, 5, 9)

- From the Manual_SARIMA model we can observe that the forecast has accounted for trend well but seasonality seems to be smoothened out as compared to the test data.
- The RMSE is also high as compared to the Automated SARIMA model.
- This model is not optimum to be used for the forecast of our data.

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

Here is the table with RMSE values of all the models built using training data and applied on test data, in lowest first order:

	Test RMSE
Alpha=0.1,Beta=0.2,Gamma=0.1,TripleExponentialSmoothing_Multiplicative	9.152
2pointTrailingMovingAverage	11.551
Alpha=0.1,Beta=0.4,Gamma=0.3,TripleExponentialSmoothing_Additive	12.096
Alpha=0.1=088,Beta=0.00,Gamma=0.004,TripleExponentialSmoothing_Additive	14.133
4pointTrailingMovingAverage	14.453
6pointTrailingMovingAverage	14.533
9pointTrailingMovingAverage	14.738
RegressionOnTime	15.237
Alpha=0.00,Beta=0.00,DoubleExponentialSmoothing	15.237
Alpha=0.07,Beta=0.046,Gamma=0.00,TripleExponentialSmoothing_Multiplicative	19.863
Automated_SARIMA(3, 1, 3)(3, 0, 3, 9)	30.894
Manual_SARIMA(2, 1, 2)(1, 0, 5, 9)	34.210
Alpha=0.07,SimpleExponentialSmoothing	36.154
Alpha=0.099,SimpleExponentialSmoothing	36.515
Automated_ARIMA(2,1,3)	36.536
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	36.588
Manual_ARIMA(2,1,2)	36.590
SimpleAverageModel	53.000
NaiveModel	79.435

Table 2. 7: RMSE Table

From this table, we see that the best model is the Triple Exponential Smoothing with multiplicative seasonality with the parameters Alpha = 0.1, Beta = 0.2 and Gamma = 0.1.

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

Triple Exponential Smoothing with multiplicative seasonality with the parameters Alpha = 0.1, Beta = 0.2 and Gamma = 0.1, provides the least RMSE value. Hence, the most optimum model for our data.

- We built this model on the complete data and the **RMSE comes to 17.013**.
- Calculated the predictions for 12 months into the future, with the upper and lower confidence bands at 95% confidence level.
- This is the sample of forecasted data:

	lower_CI	prediction	upper_ci
1995-08-31	18.055911	51.482458	84.909004
1995-09-30	16.865352	50.291899	83.718445
1995-10-31	17.712838	51.139385	84.565932
1995-11-30	26.001769	59.428315	92.854862
1995-12-31	49.136196	82.562742	115.989289

Table 2. 8: Wine Sales Forecast with CI

- This is how the forecasted values appear visually:

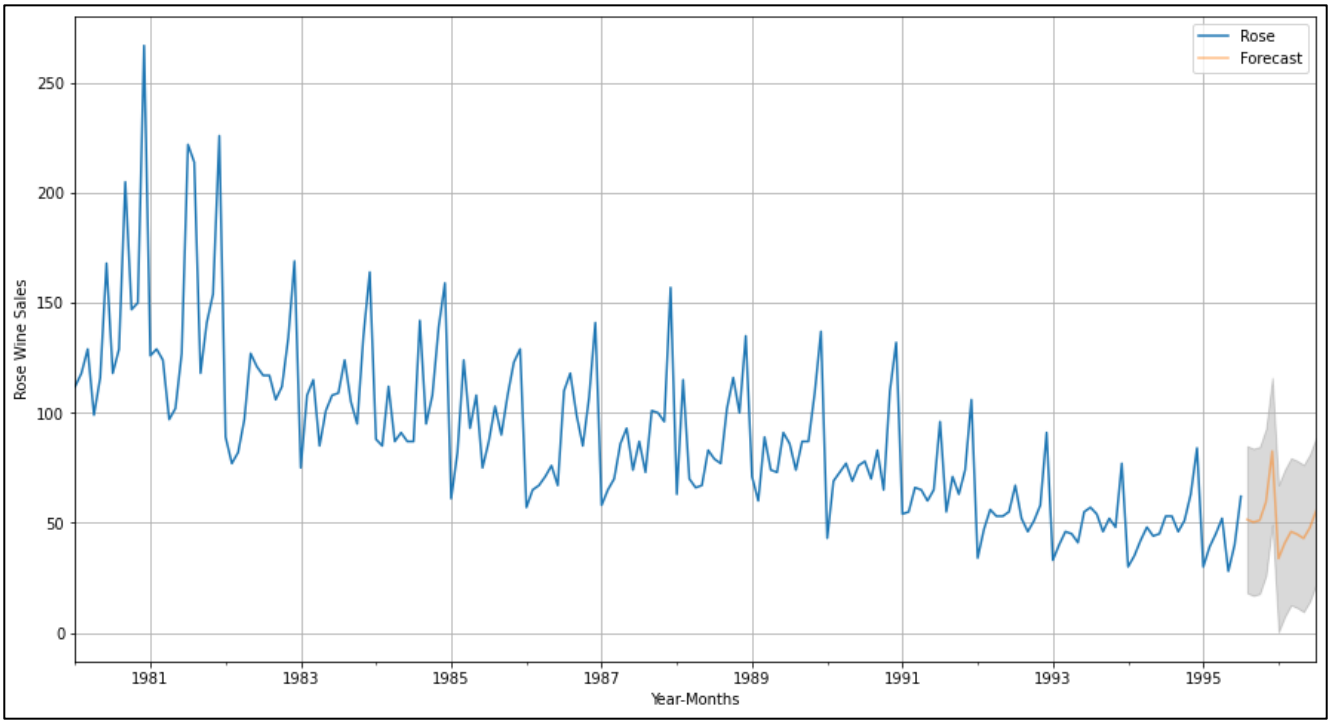


Figure 2. 43: Forecast on Compete Data with Confidence Interval

- The forecasts that we calculated are not definite values, as the future is unpredictable. It is best to provide forecasts in a range, to account for any unprecedented event that might occur in future.
- We have taken a confidence interval of 95% to present our forecasts. The orange line in the graph is the forecasts and the grey area around it is the confidence interval, showing the upper and lower limit to expect as the forecast.

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

- The model we have built, takes in account the trend and seasonality present in our data to make the forecasts.
- Giving a clear indication to the business as to what to expect in terms of rose wine sales.
- As per the forecast, the trend is decreasing and seasonality is also multiplying, taking the forecast downwards.
- Company can expect the sales close to 120 in the busiest season.
- After July 1995, company is approaching festive season. This would be the perfect time to roll out the advertisements and/or marketing campaigns to boost the sales.
- Looking at the past trends, ABC Estate Wines is not performing well in terms of Rose wine sales and has a downward trend.
- A through market study/ customer survey can give them a clear picture as to why the sale is low.