Time Series Forecasting-Rose Project Report

Mr. Akash Kamble

June_A Batch

Program: DSBA

❖ Table of Contents:

❖ List of Figures:

❖ List of Tables:

1. Time Series Forecasting for Rose Wine Sales

Executive Summary

Intend of the study is to use different time series forecasting methods to forecast the sales for Rose wines of ABC company.

Introduction

We will built various models, evaluate the model on basis of RMSE score and finally choose the best model.

Sample of the dataset

YearMonth	Rose
1980-01-01	112.0
1980-02-01	118.0
1980-03-01	129.0
1980-04-01	99.0
1980-05-01	116.0

Table 1.1: Data Sample

Dataset has records of monthly Rose wine sales from year 1980 to 1995.

Exploratory Data Analysis (EDA)

- Let's check for missing values in the data frame
  - There are missing values in the data set.

- Checking for outliers present in the data

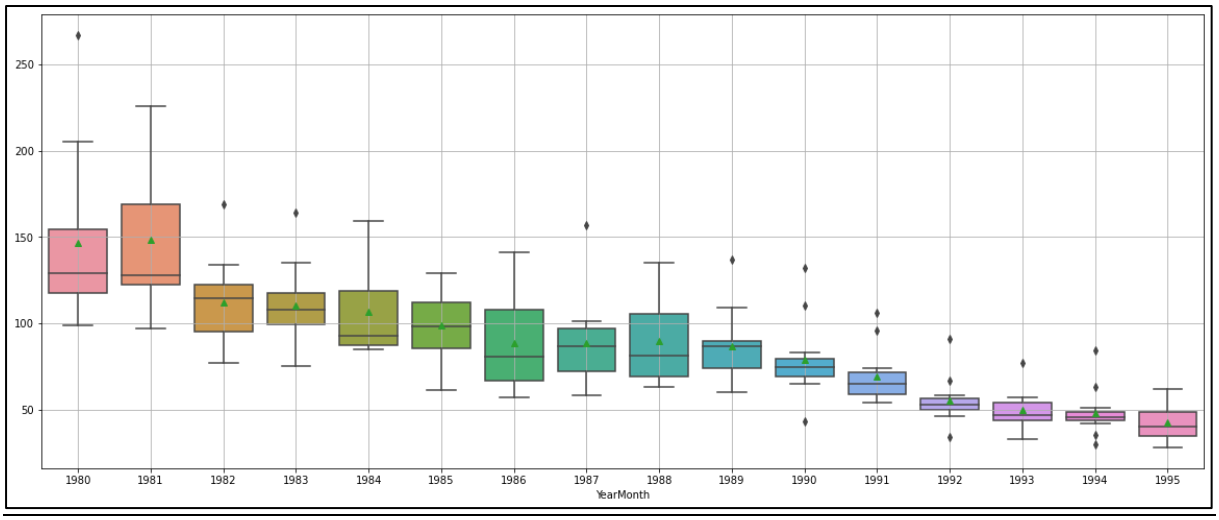

Fig 1.1: Boxplot for each year

From above boxplots,

- There are outliers present in the data.
- Box plot is just to understand the data distribution and not for any treatment.

Problem Statement 1:

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

Q1.1. Read the data as an appropriate Time Series data and plot the data.

- Read the data as an appropriate time series data:

YearMonth	Rose
1980-01-01	112.0
1980-02-01	118.0
1980-03-01	129.0
1980-04-01	99.0
1980-05-01	116.0

Table 1.1: Data Sample

- Plotting the data:

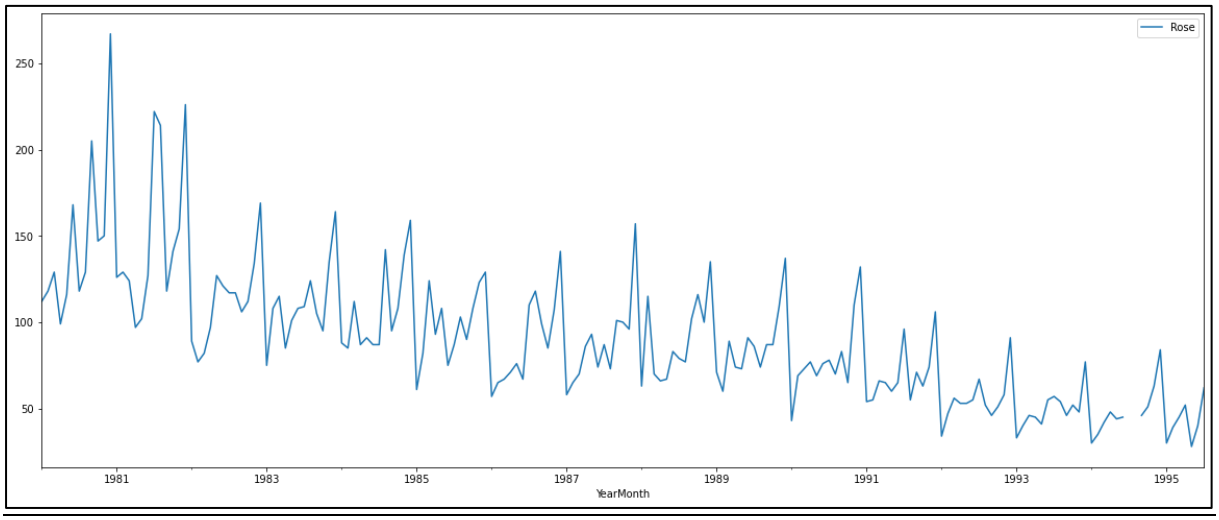

Fig 1.2: Series Plot

- Data values are stored in correct time order and there are some missing values (they are treated in further section).
- There is strong downtrend trend component present in the dataset.
- Seasonal trend is present.
- It can be inferred by looking at the data that the series is multiplicative in nature.

Q1.2 Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.

Exploratory Data Analysis (EDA)

- **Treating Missing Values**:
  - There are missing values present in the data set**.**
  - Therefore, we should interpolate the values
  - Following is the graph after treating the missing values

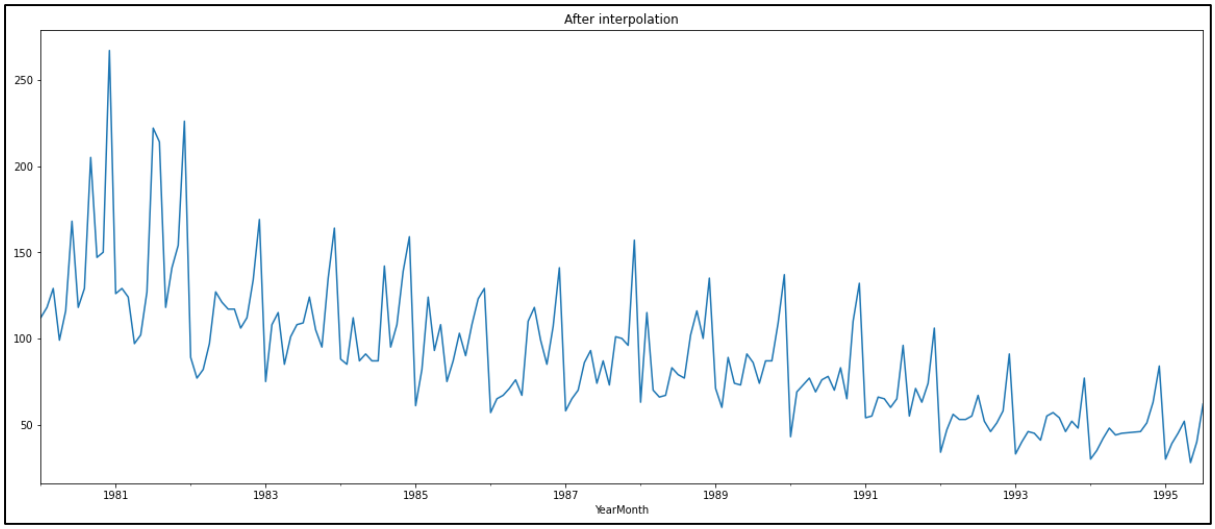

Fig 1.2.1: Series Plot after treatment of missing values

- **Yearly Sales Boxplot**:

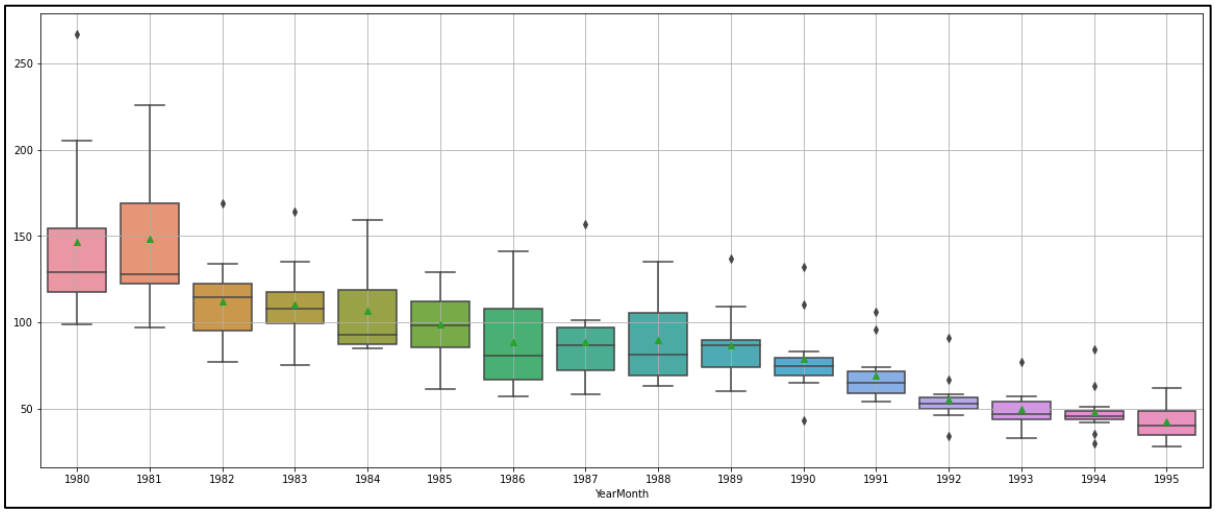

Fig 1.3: Yearly Sales Boxplot

- There are outliers present in yearly sales data, same is the reason for spikes in the line plot.
- There is clearly downtrend present in the series.

- **Monthly Sales Boxplot**:

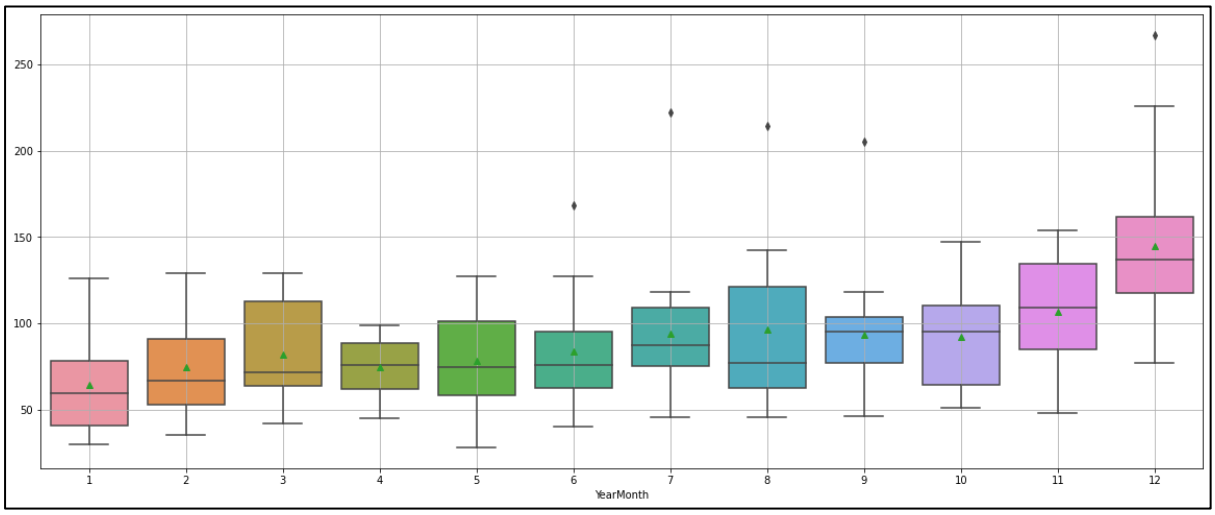

Fig 1.4: Monthly Sales Boxplot

- There seems to be higher sales in last two months of the year.
- Highest sales are contributed by December month for each year, followed by November.
- **Monthly Sales lineplot**:

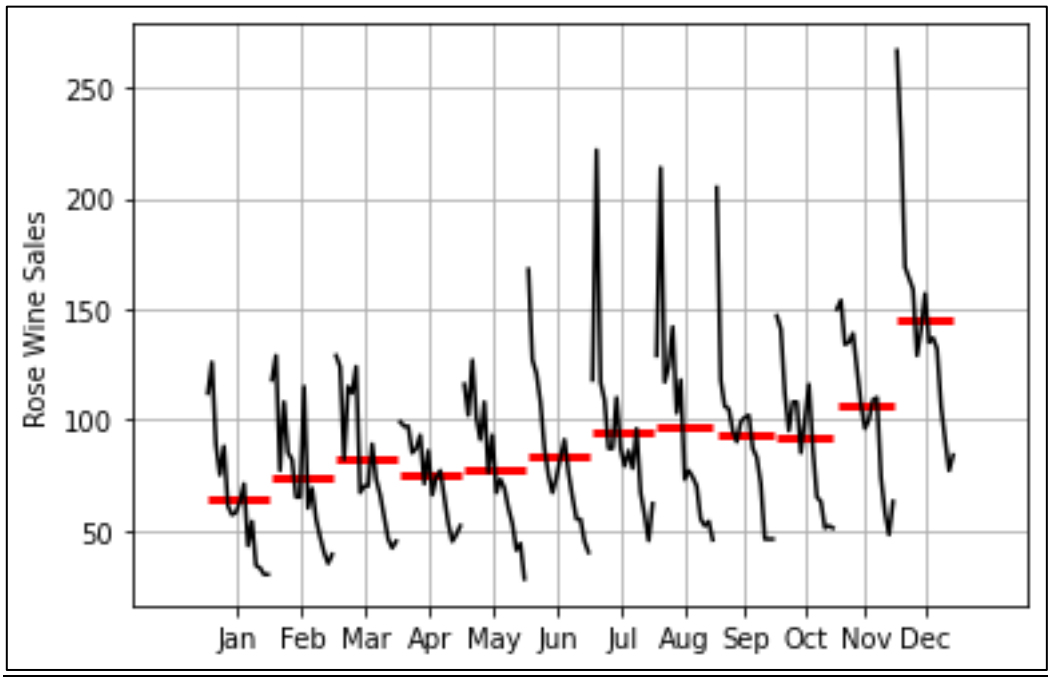

Fig 1.5: Mean of monthly wine sales

- There is too much of variance in monthly sales of rose wine.

- Cross tab for monthly and yearly wine sales:
  - Cross Tab

YearMonth	1	2	3	4	5	6	7	8	9	10	11	12
YearMonth												
1980	112.0	118.0	129.0	99.0	116.0	168.0	118.000000	129.000000	205.0	147.0	150.0	267.0
1981	126.0	129.0	124.0	97.0	102.0	127.0	222.000000	214.000000	118.0	141.0	154.0	226.0
1982	89.0	77.0	82.0	97.0	127.0	121.0	117.000000	117.000000	106.0	112.0	134.0	169.0
1983	75.0	108.0	115.0	85.0	101.0	108.0	109.000000	124.000000	105.0	95.0	135.0	164.0
1984	88.0	85.0	112.0	87.0	91.0	87.0	87.000000	142.000000	95.0	108.0	139.0	159.0
1985	61.0	82.0	124.0	93.0	108.0	75.0	87.000000	103.000000	90.0	108.0	123.0	129.0
1986	57.0	65.0	67.0	71.0	76.0	67.0	110.000000	118.000000	99.0	85.0	107.0	141.0
1987	58.0	65.0	70.0	86.0	93.0	74.0	87.000000	73.000000	101.0	100.0	96.0	157.0
1988	63.0	115.0	70.0	66.0	67.0	83.0	79.000000	77.000000	102.0	116.0	100.0	135.0
1989	71.0	60.0	89.0	74.0	73.0	91.0	86.000000	74.000000	87.0	87.0	109.0	137.0
1990	43.0	69.0	73.0	77.0	69.0	76.0	78.000000	70.000000	83.0	65.0	110.0	132.0
1991	54.0	55.0	66.0	65.0	60.0	65.0	96.000000	55.000000	71.0	63.0	74.0	106.0
1992	34.0	47.0	56.0	53.0	53.0	55.0	67.000000	52.000000	46.0	51.0	58.0	91.0
1993	33.0	40.0	46.0	45.0	41.0	55.0	57.000000	54.000000	46.0	52.0	48.0	77.0
1994	30.0	35.0	42.0	48.0	44.0	45.0	45.333333	45.666667	46.0	51.0	63.0	84.0
1995	30.0	39.0	45.0	52.0	28.0	40.0	62.000000	NaN	NaN	NaN	NaN	NaN

Table 1.2: Cross Tab

- Plotting the Cross Tab:

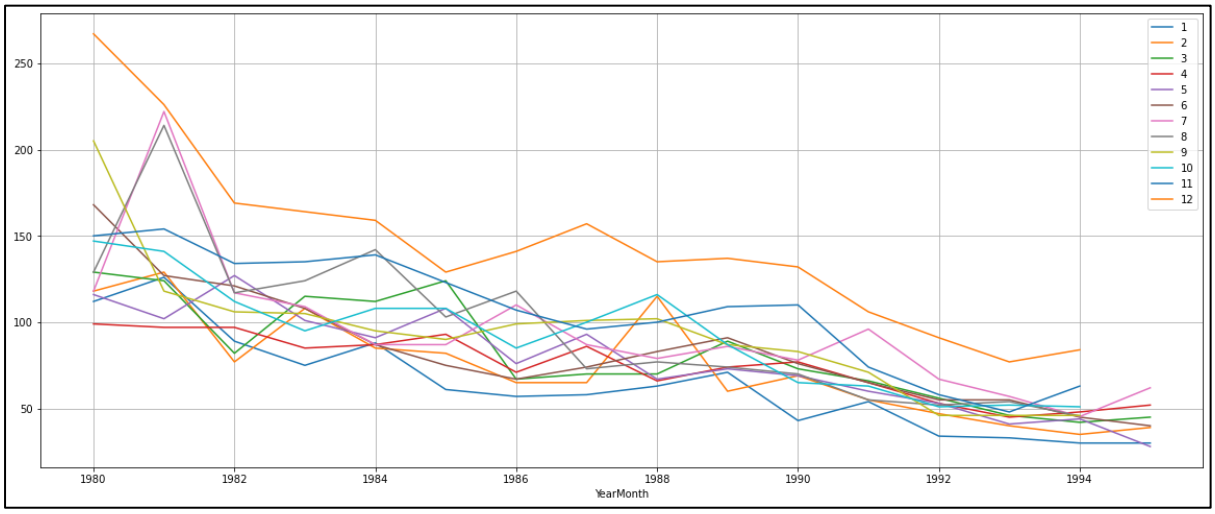

Fig 1.6: Plotting the cross tab

- We can infer that December month is the highest wine selling month throughout the study period.

- Percentage change in Sales of Rose Wine:

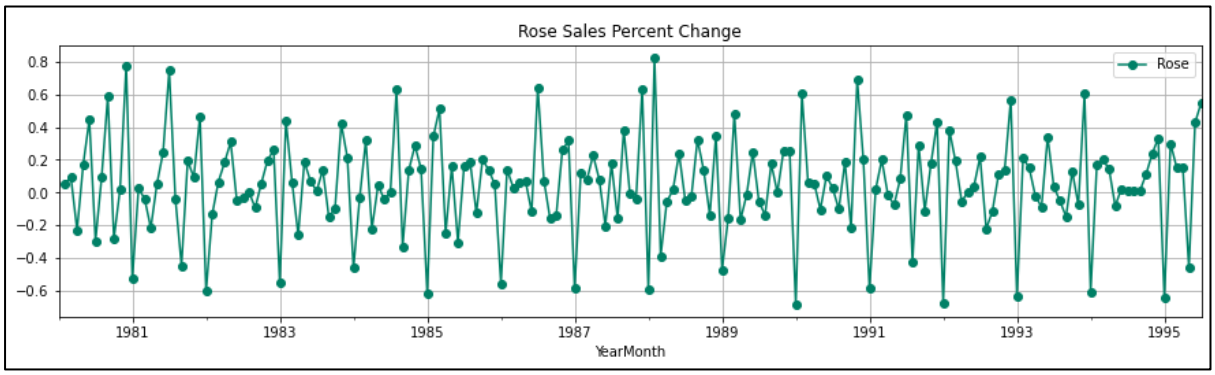

Fig 1.7: Percentage change in wine sales

- We can infer that there is lot of variation in percentage change in sales of Rose wines.

- Decomposition of Time Series:

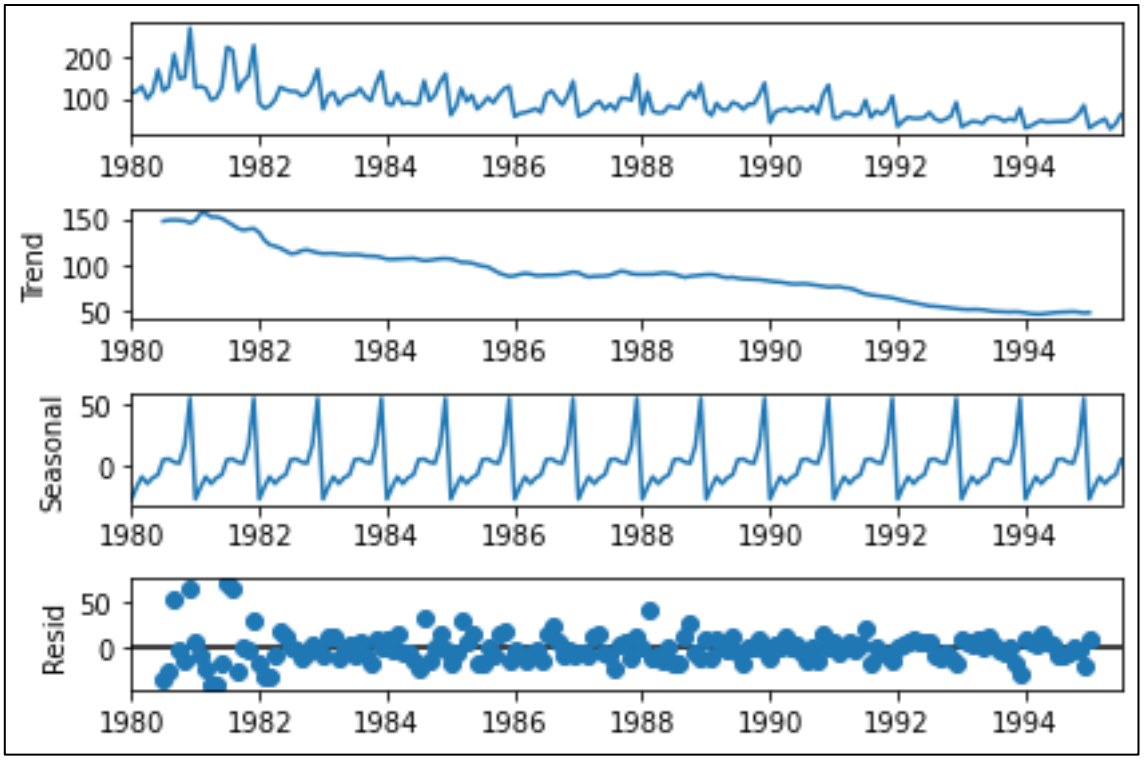

Fig 1.8: Additive Decomposition

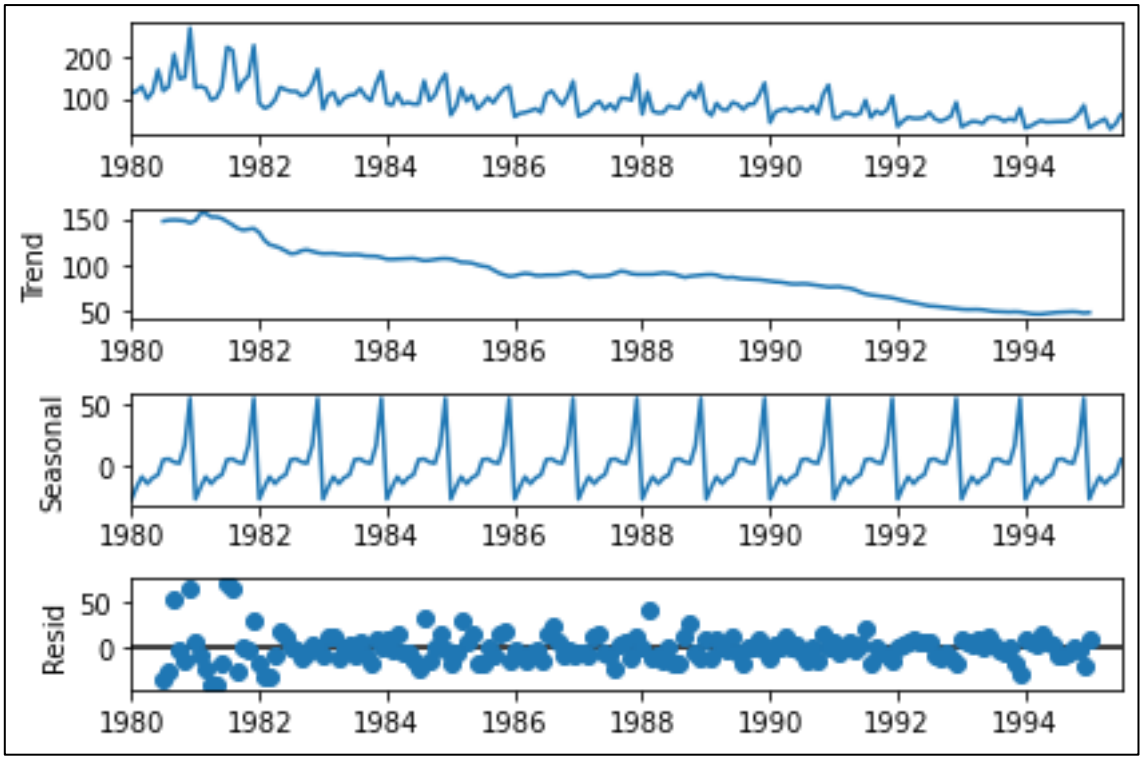

Fig 1.9: Multiplicative Decomposition

- There is a clear downtrend throughout the series.
- There is seasonal component present in the data set.
- Here in both the cases i.e. additive and multiplicative, the residuals are similar. We can say Multiplicative decomposition fits the best looking at the original series.

Q1.3 Split the data into training and test. The test data should start in 1991.

- **Training Dataset:**

First few rows of Training Data

YearMonth	Rose
1980-01-01	112.0
1980-02-01	118.0
1980-03-01	129.0
1980-04-01	99.0
1980-05-01	116.0

Last few rows of Training Data

YearMonth	Rose
1990-08-01	70.0
1990-09-01	83.0
1990-10-01	65.0
1990-11-01	110.0
1990-12-01	132.0

Table 1.3: Training Data Set

- **Testing Dataset:**

First few rows of Test Data

YearMonth	Rose
1991-01-01	54.0
1991-02-01	55.0
1991-03-01	66.0
1991-04-01	65.0
1991-05-01	60.0

Last few rows of Test Data

YearMonth	Rose
1995-03-01	45.0
1995-04-01	52.0
1995-05-01	28.0
1995-06-01	40.0
1995-07-01	62.0

Table 1.4: Testing Dataset

- **Test Train Graph**

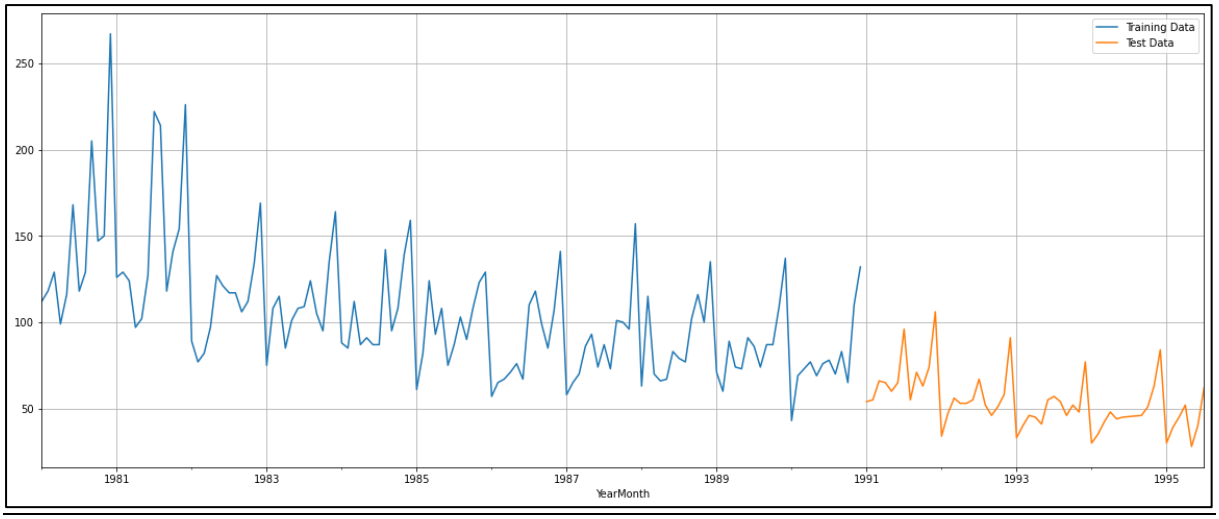

Fig 1.10: Test Train Graph

Q1.4 Build all the exponential smoothing models on the training data and evaluate the model using RMSE on the test data. Other additional models such as regression, naïve forecast models, simple average models, moving average models should also be built on the training data and check the performance on the test data using RMSE.

1. Linear Regression

- **Step 1:** Import necessary libraries
  from sklearn.linear_model import LinearRegression

- **Step 2:** Apply logistic regression by using function
  - LR_model = LinearRegression()

- **Step 3:** Forecasting on test data:

YearMonth	Rose	time	RegOnTime
1991-01-01	54.0	1	137.321144
1991-02-01	55.0	2	136.826766
1991-03-01	66.0	3	136.332388
1991-04-01	65.0	4	135.838010
1991-05-01	60.0	5	135.343632

Table 1.5: Forecasting on test data

- **Step 4:** Plot the Forecasts:

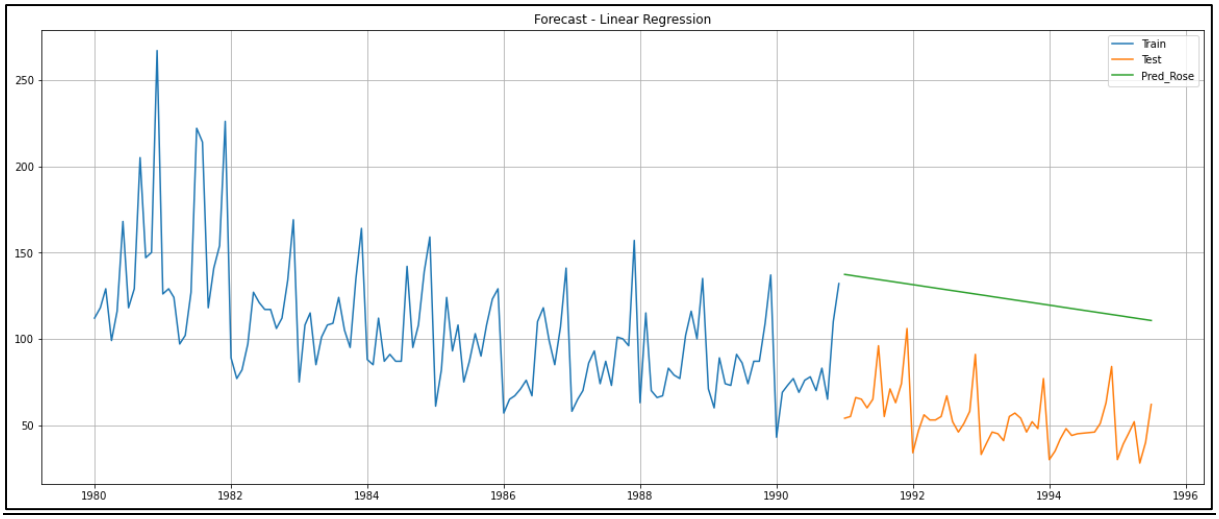

Fig 1.11: Forecast – Linear Regression

- **Step 5:** Calculate the RMSE

  - RMSE Score: 71.59

2. Naïve Approach:

- **Step 1:** Import necessary libraries
  No specific library required

- **Step 2:** Apply Naïve model by using function
  No specific function required

- **Step 3:** Forecasting on test data:

```
YearMonth
1991-01-01    132.0
1991-02-01    132.0
1991-03-01    132.0
1991-04-01    132.0
1991-05-01    132.0
Freq: MS, Name: Naive, dtype: float64
```

Table 1.6: Forecasting on test data

- **Step 4:** Plot the Forecasts:

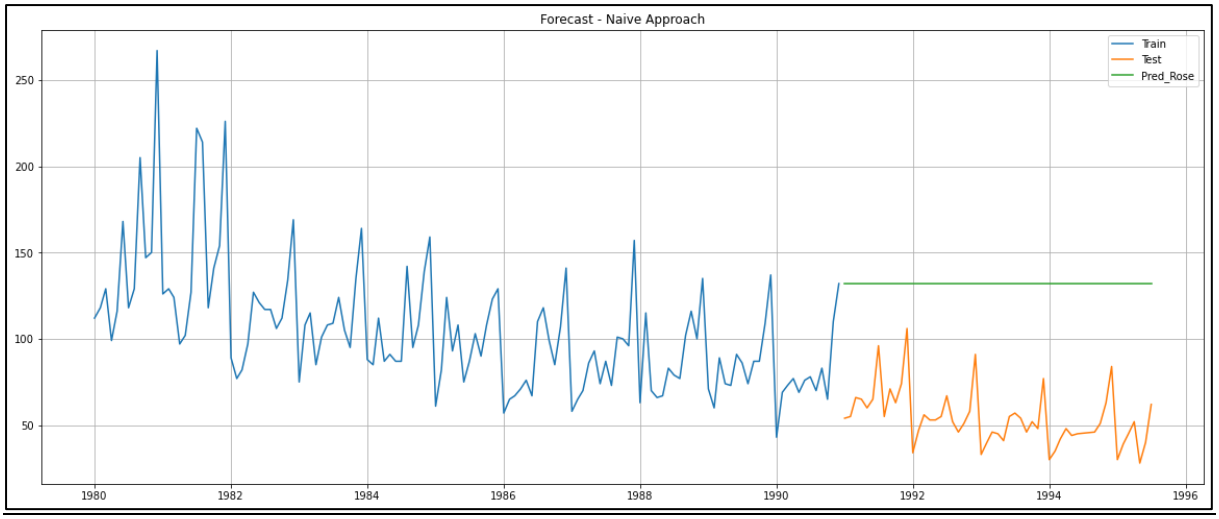

Fig 1.12: Forecast – Naïve Approach

- **Step 5:** Calculate the RMSE

  - RMSE Score: 79.71

3. Simple Average Model

- **Step 1:** Import necessary libraries
  No specific library required

- **Step 2:** Apply Simple Average by using function
  No specific function required

- **Step 3:** Forecasting on test data:

YearMonth	Rose	mean_forecast
1991-01-01	54.0	53.854545
1991-02-01	55.0	53.854545
1991-03-01	66.0	53.854545
1991-04-01	65.0	53.854545
1991-05-01	60.0	53.854545

Table 1.7: Forecasting on test data

- **Step 4:** Plot the Forecasts:

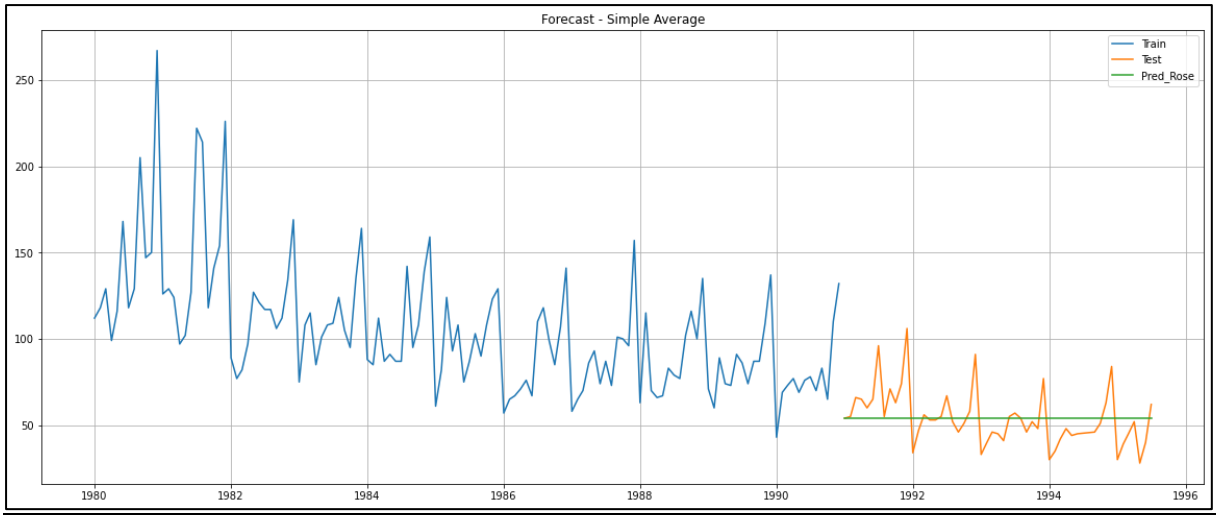

Fig 1.13: Forecast – Simple Average

- **Step 5:** Calculate the RMSE
  - RMSE Score: 15.75

4. Moving Average Model

- **Step 1:** Import necessary libraries
  No specific library required

- **Step 2:** Apply Moving Average by using function
  Moving_Average['Trailing_2'] = Moving_Average['Rose'].rolling(2).mean()
  Moving_Average['Trailing_4'] = Moving_Average[' Rose '].rolling(4).mean()
  Moving_Average['Trailing_6'] = Moving_Average['Rose'].rolling(6).mean()
  Moving_Average['Trailing_8'] = Moving_Average['Rose'].rolling(8).mean()
- **Step 3:** Forecasting on test data:

YearMonth	Rose	Trailing_2	Trailing_4	Trailing_6	Trailing_8
1980-01-01	112.0	NaN	NaN	NaN	NaN
1980-02-01	118.0	115.0	NaN	NaN	NaN
1980-03-01	129.0	123.5	NaN	NaN	NaN
1980-04-01	99.0	114.0	114.50	NaN	NaN
1980-05-01	116.0	107.5	115.50	NaN	NaN
1980-06-01	168.0	142.0	128.00	123.666667	NaN
1980-07-01	118.0	143.0	125.25	124.666667	NaN
1980-08-01	129.0	123.5	132.75	126.500000	123.625
1980-09-01	205.0	167.0	155.00	139.166667	135.250
1980-10-01	147.0	176.0	149.75	147.166667	138.875

Table 1.8: Forecasting on test data

- **Step 4:** Plot the Forecasts:

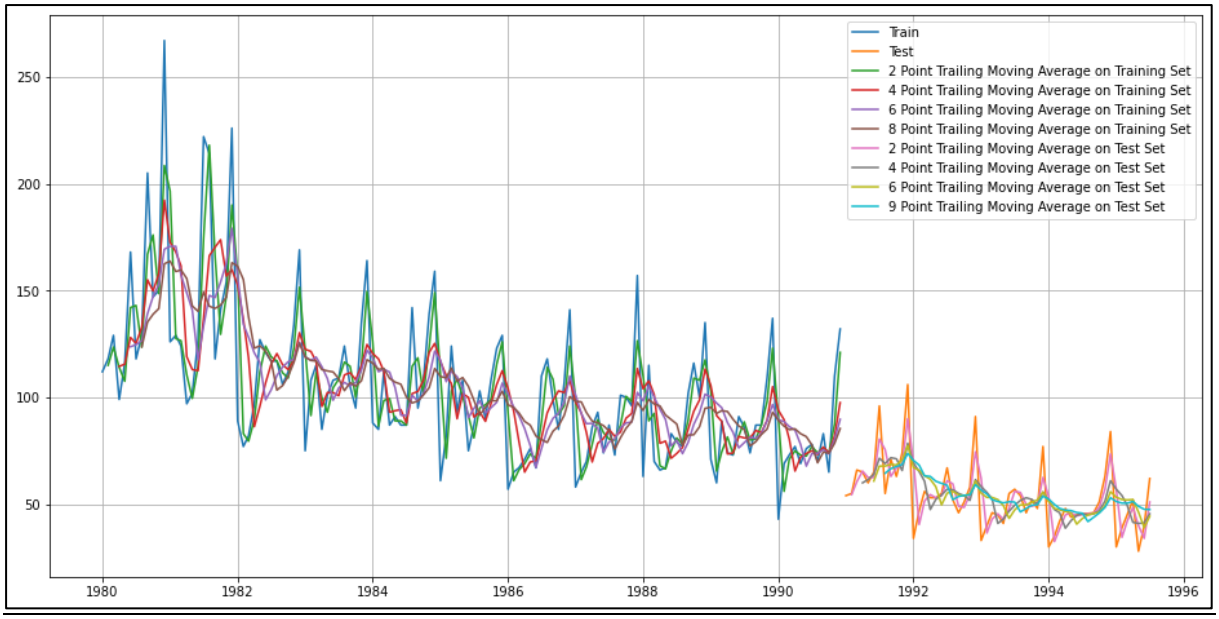

Fig 1.14: Forecast – Trailing Moving Average

- **Step 5:** Calculate the RMSE

2pointTrailingMovingAverage	10.402622
4pointTrailingMovingAverage	13.176123
6pointTrailingMovingAverage	13.299874
8pointTrailingMovingAverage	13.997297

Table 1.9: RMSE on various trailing moving average

5. Simple Moving Average Model

- **Step 1:** Import necessary libraries
  from statsmodels.tsa.api import SimpleExpSmoothing

- **Step 2:** Apply Simple Moving Average by using function
  SES_model = SimpleExpSmoothing(SES_train['Rose'])
  SES_model_autofit = SES_model.fit(optimized=True)

- **Step 3:** Forecasting on test data:

YearMonth	Rose	Predict
1991-01-01	54.0	87.104999
1991-02-01	55.0	87.104999
1991-03-01	66.0	87.104999
1991-04-01	65.0	87.104999
1991-05-01	60.0	87.104999

Table 1.10: Forecast on test dataset

- **Step 4:** Plot the Forecasts:

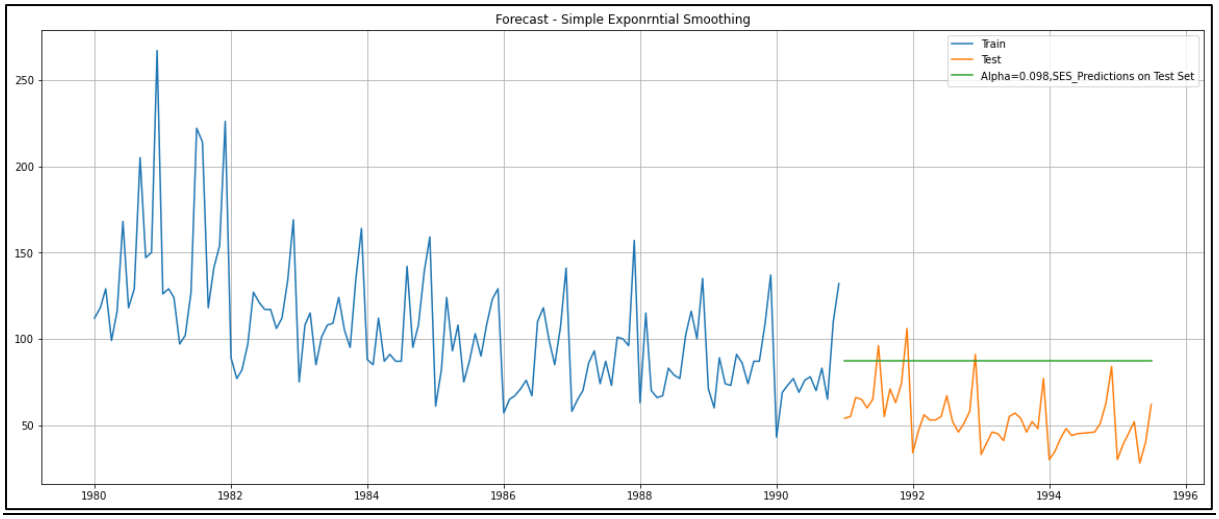

Fig 1.15: Forecast – Trailing Moving Average

- **Step 5:** Calculate the RMSE
  - RMSE Score: 36.79

- **Step 6:** Iterate through different parameters (alpha, beta, gamma)

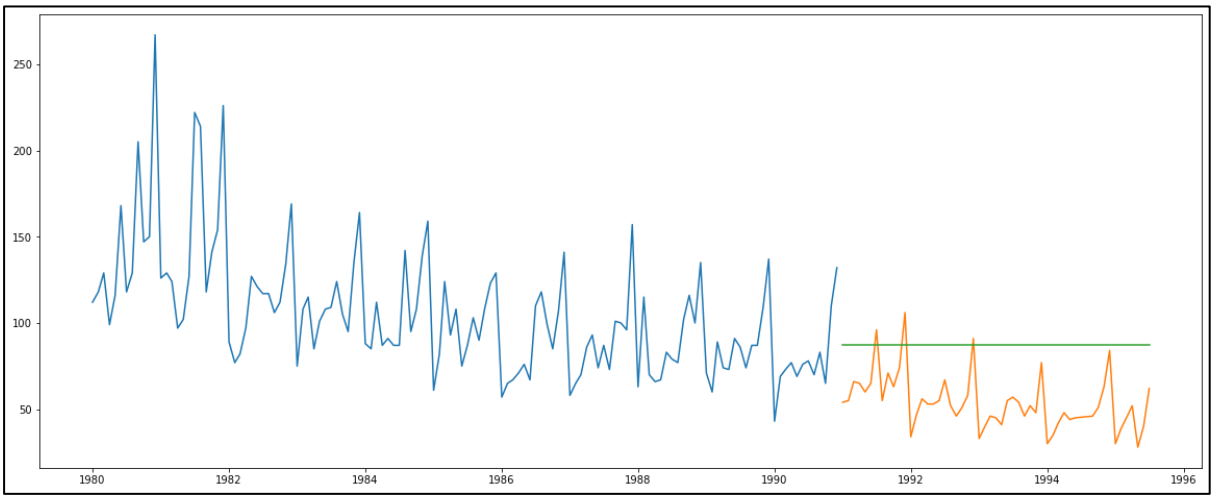

Fig 1.16: Forecast – Trailing Moving Average with different params

  - New RMSE Score: 36.43

6. Double Exponential Smoothing Model

- **Step 1:** Import necessary libraries
  from statsmodels.tsa.api import Holt

- **Step 2:** Apply Double Exponential Smoothing Model by using function
  DES_model = Holt(DES_train['Rose'])
  DES_model_autofit = DES_model.fit(optimized=True)

- **Step 3:** Forecasting on test data:

YearMonth	Rose	Predict
1991-01-01	54.0	72.063249
1991-02-01	55.0	71.568871
1991-03-01	66.0	71.074493
1991-04-01	65.0	70.580115
1991-05-01	60.0	70.085736

Table 1.12: Forecasting on test data

- **Step 4:** Plot the Forecasts:

Fig 1.19: Forecast – Double exponential Smoothing

- **Step 5:** Calculate the RMSE
  - RMSE Score: 84.29

- **Step 6: Iterate through different parameters (alpha, beta, gamma)**

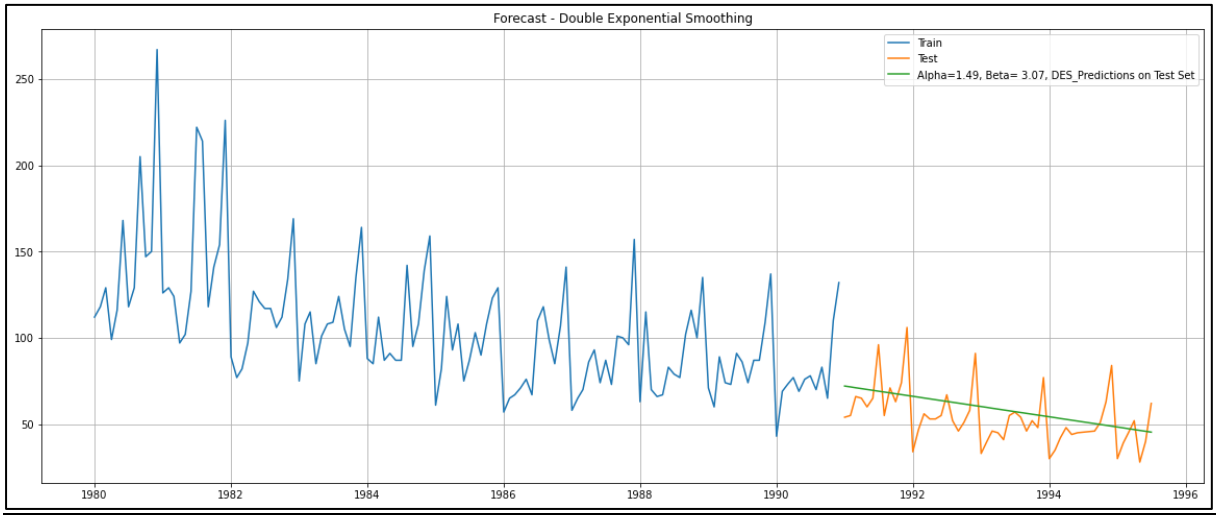

Fig 1.20: Forecast – Double exponential Smoothing with different params

  - New RMSE Score: 15.26

7. Triple Exponential Smoothing Model

- **Step 1:** Import necessary libraries
  from statsmodels.tsa.api import ExponentialSmoothing

- **Step 2:** Apply Triple Exponential Smoothing Model by using function
  TES_model = ExponentialSmoothing(TES_train['Rose'],trend='additive',seasonal='multiplicative')

  TES_model_autofit = TES_model.fit(optimized=True)

- **Step 3:** Forecasting on test data:

YearMonth	Rose	Predict
1991-01-01	54.0	56.411071
1991-02-01	55.0	63.801331
1991-03-01	66.0	69.500435
1991-04-01	65.0	60.599603
1991-05-01	60.0	67.889542

Table 1.13: Forecasting on test data

- **Step 4:** Plot the Forecasts:

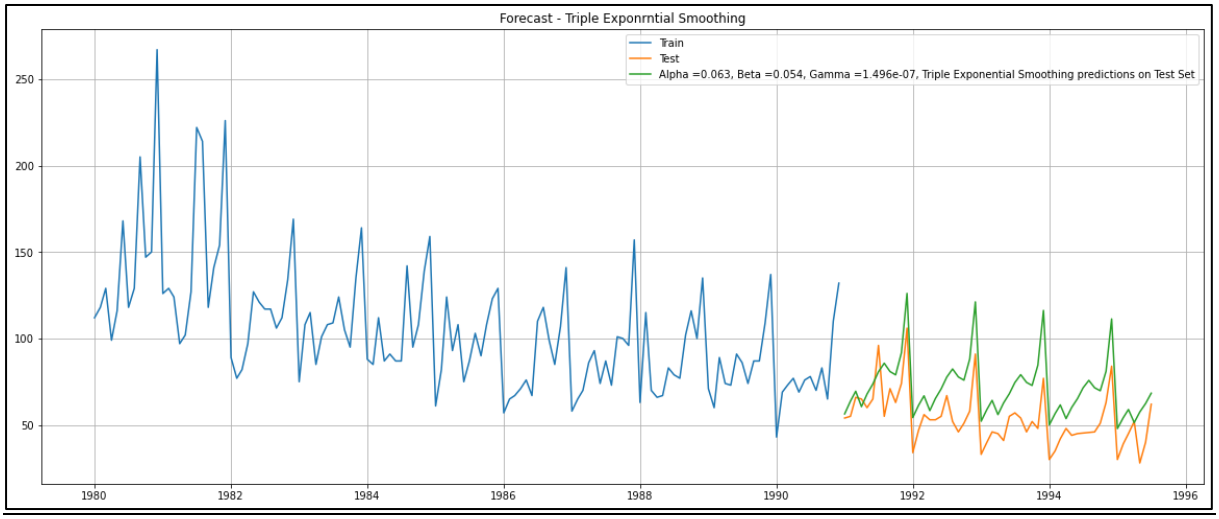

Fig 1.21: Forecast – Triple exponential Smoothing

- **Step 5:** Calculate the RMSE
  - RMSE Score: 20.37

- **Step 6: Iterate through different parameters (alpha, beta, gamma)**

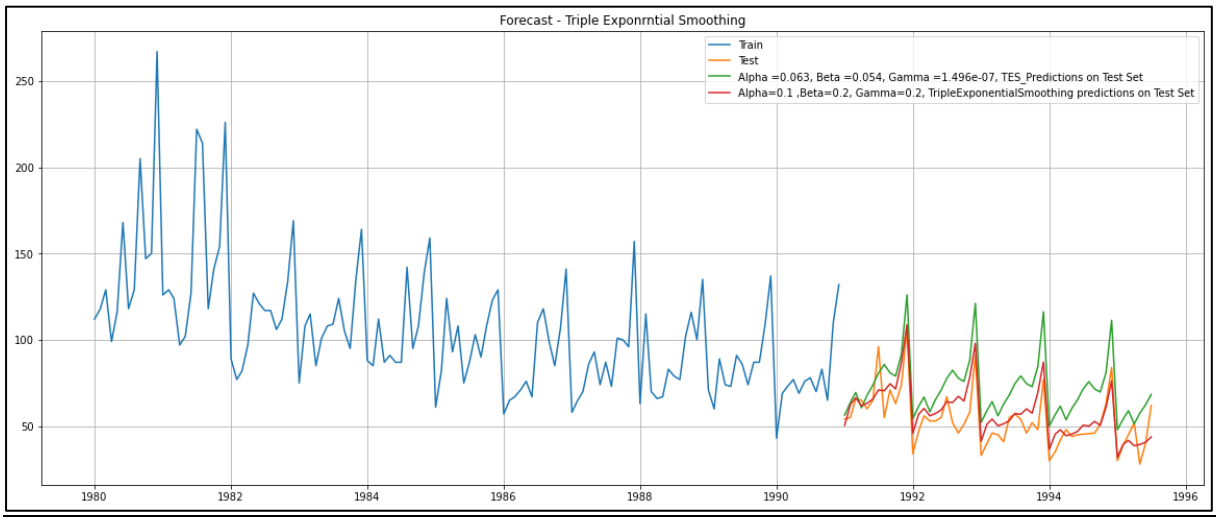

Fig 1.22: Forecast – Triple exponential Smoothing with different params

  - New RMSE Score: 9.22

Q1.5 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.

- Since ARIMA/SARIMA model requires a stationary series, a formal stationarity test needs to be applied to the time series under consideration.
- Augmented Dickey-Fuller Test: A formal test to check whether time series data follows stationary process.
    1. H0: Time series is non-stationary
    2. H1: Time series is stationary
- Following is the result of AD-Fuller Test:

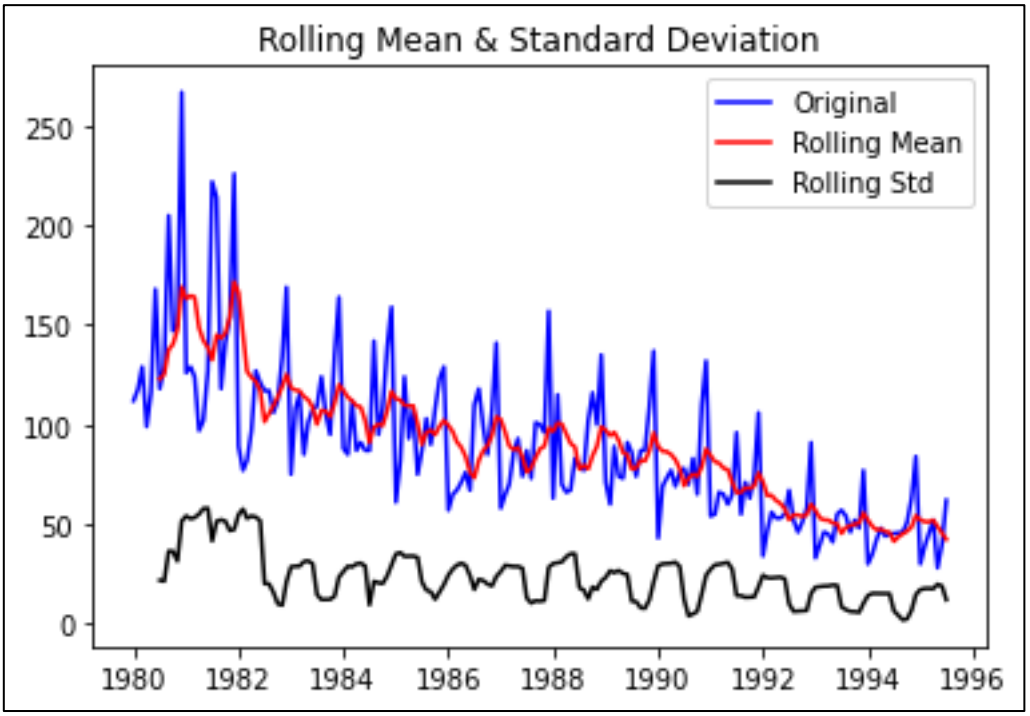

Fig 1.23: Rolling mean & STD deviation

Results of Dickey-Fuller Test:	
Test Statistic	-1.876699
p-value	0.343101
#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	

Table 1.14: AD Fuller Test Result

- P-value is 0.24 which is greater than level of significance (0.05). Therefore, we failed to reject the null hypothesis. Thus, the series is non-stationary.
- Now, we need to take 1st order difference so as to convert the non-stationary series into stationary series.

- Following is the result of AD fuller test on 1st order difference.

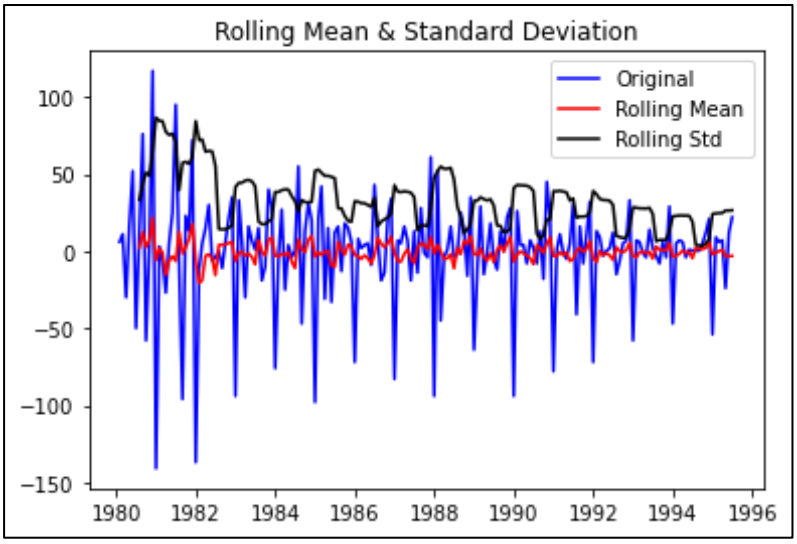

Fig 1.24: Rolling mean & STD deviation with order 1 differentiation

Results of Dickey-Fuller Test:	
Test Statistic	-8.044392e+00
p-value	1.810895e-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	

Table 1.15: AD Fuller Test Result

- P-value is 1.81e-12 which is lesser than level of significance (0.05). Therefore, we reject the null hypothesis. Thus, the series is now stationary.

Q1.6 Build an automated version of the ARIMA/SARIMA model in which the parameters are selected using the lowest Akaike Information Criteria (AIC) on the training data and evaluate this model on the test data using RMSE?

➢ **Automated ARIMA Model**

- **Step 1:** Import necessary libraries
  from statsmodels.tsa.arima.model import ARIMA

- **Step 2:** Apply automated ARIMA by selecting range of p, d & q values
  - Range of 'p': 0 to 2 (2 is inclusive)
  - Range of 'd': 1 (as 1st differential of series is stationary)
  - Range of 'q': 0 to 2 (2 is inclusive)

- **Step 3:** Sort AIC values in ascending order

	param	AIC
2	(0, 1, 2)	1279.671529
5	(1, 1, 2)	1279.870723
4	(1, 1, 1)	1280.574230
7	(2, 1, 1)	1281.507862
8	(2, 1, 2)	1281.870722
1	(0, 1, 1)	1282.309832
6	(2, 1, 0)	1298.611034
3	(1, 1, 0)	1317.350311
0	(0, 1, 0)	1333.154673

Table 1.16: AIC Values

- **Step 4:** Selecting the best parameter with lowest AIC

  p: 0 (Partial autocorrelation)

  d: 1 (degree of differentiation)

  q: 2 (Autocorrelation)

- **Step 5:** ARIMAX Result:

SARIMAX Results

Dep. Variable:	Rose	No. Observations:	132
Model:	ARIMA(0, 1, 2)	Log Likelihood	-636.836
Date:	Sun, 20 Feb 2022	AIC	1279.672
Time:	22:16:48	BIC	1288.297
Sample:	01-01-1980	HQIC	1283.176
	- 12-01-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.6970	0.072	-9.689	0.000	-0.838	-0.556
ma.L2	-0.2042	0.073	-2.794	0.005	-0.347	-0.061
sigma2	965.8407	88.305	10.938	0.000	792.766	1138.915

Ljung-Box (L1) (Q):	0.14	Jarque-Bera (JB):	39.24
Prob(Q):	0.71	Prob(JB):	0.00
Heteroskedasticity (H):	0.36	Skew:	0.82
Prob(H) (two-sided):	0.00	Kurtosis:	5.13

Table 1.17: ARIMAX Result

- **Step 6:** Residual Diagnostics:

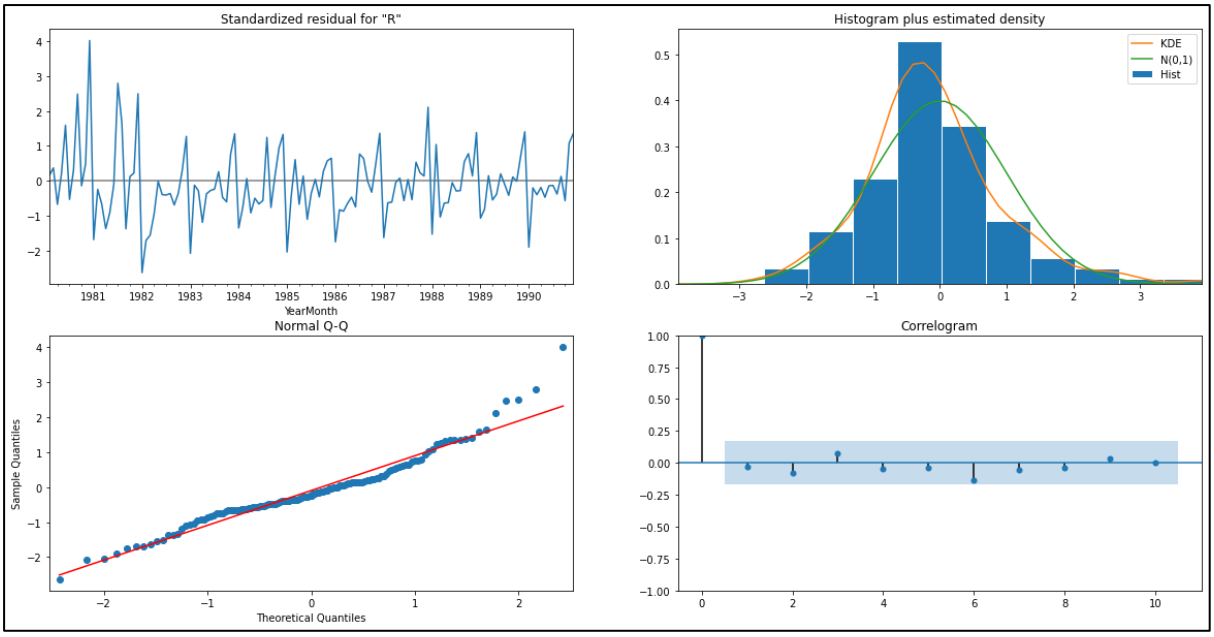

Fig 1.25: Residual Diagnostics

- **Step 7:** Calculate the RMSE
  - RMSE Score: 37.31

➢ Automated SARIMA Model

- **Step 1:** Import necessary libraries
  import statsmodels.api as sm

- **Step 2:** Apply automated SARIMA by selecting range of p, d & q values
  - Range of 'p': 0 to 2 (2 is inclusive)
  - Range of 'd': 1 (as 1st differential of series is stationary)
  - Range of 'q': 0 to 2 (2 is inclusive)

- **Step 3:** Sort AIC values in ascending order

	param	seasonal	AIC
26	(0, 1, 2)	(2, 1, 2, 12)	774.969119
53	(1, 1, 2)	(2, 1, 2, 12)	776.940109
80	(2, 1, 2)	(2, 1, 2, 12)	776.996101
17	(0, 1, 1)	(2, 1, 2, 12)	782.153872
79	(2, 1, 2)	(2, 1, 1, 12)	783.703652

Table 1.17: AIC Values

- **Step 4:** Selecting the best parameter with lowest AIC
  (p,d,q)(P,D,Q) = (0,1,2)(2,1,2,12)

- **Step 5:** SARIMAX Result:

SARIMAX Results

Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(0, 1, 2)x(2, 1, 2, 12)	Log Likelihood	-380.485
Date:	Sun, 20 Feb 2022	AIC	774.969
Time:	22:17:51	BIC	792.622
Sample:	0	HQIC	782.094
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.9524	0.184	-5.166	0.000	-1.314	-0.591
ma.L2	-0.0764	0.126	-0.605	0.545	-0.324	0.171
ar.S.L12	0.0480	0.177	0.271	0.786	-0.299	0.395
ar.S.L24	-0.0419	0.028	-1.513	0.130	-0.096	0.012
ma.S.L12	-0.7526	0.301	-2.503	0.012	-1.342	-0.163
ma.S.L24	-0.0721	0.204	-0.354	0.723	-0.471	0.327
sigma2	187.8660	45.275	4.149	0.000	99.129	276.603

Ljung-Box (L1) (Q):	0.06	Jarque-Bera (JB):	4.86
Prob(Q):	0.81	Prob(JB):	0.09
Heteroskedasticity (H):	0.91	Skew:	0.41
Prob(H) (two-sided):	0.79	Kurtosis:	3.77

Table 1.18: SARIMAX Result

- **Step 6:** Residual Diagnostics:

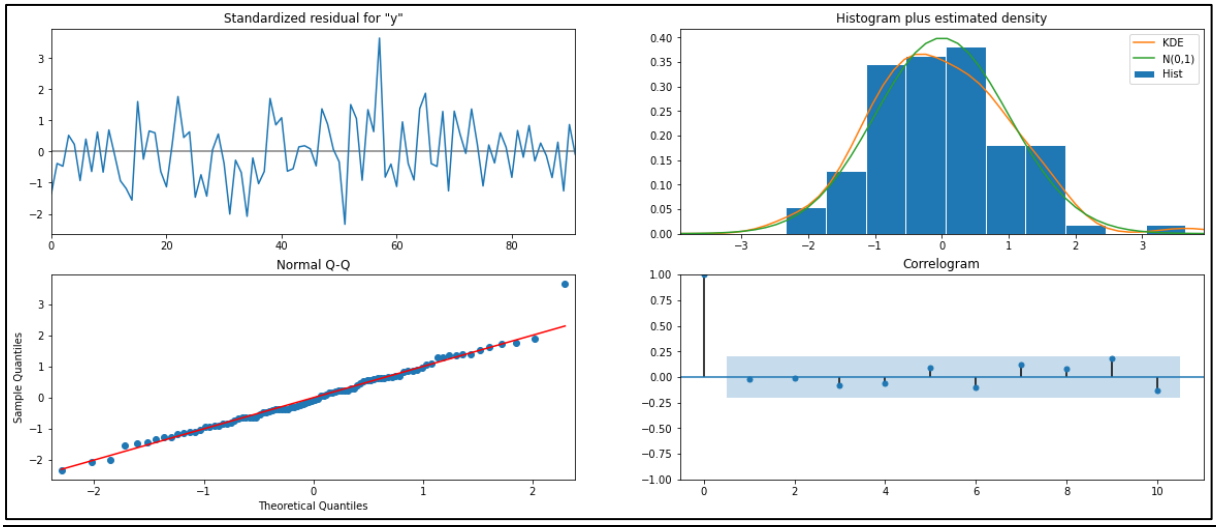

Fig 1.25: Residual Diagnostics

- **Step 7:** Calculate the RMSE
  - RMSE Score: 16.50

Q1.7 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.

➢ **Cut-off based ARIMA Model**

- **Step 1:** Import necessary libraries

  import statsmodels.api as sm

- **Step 2:** Decide the cut-off points of ACF and PACF

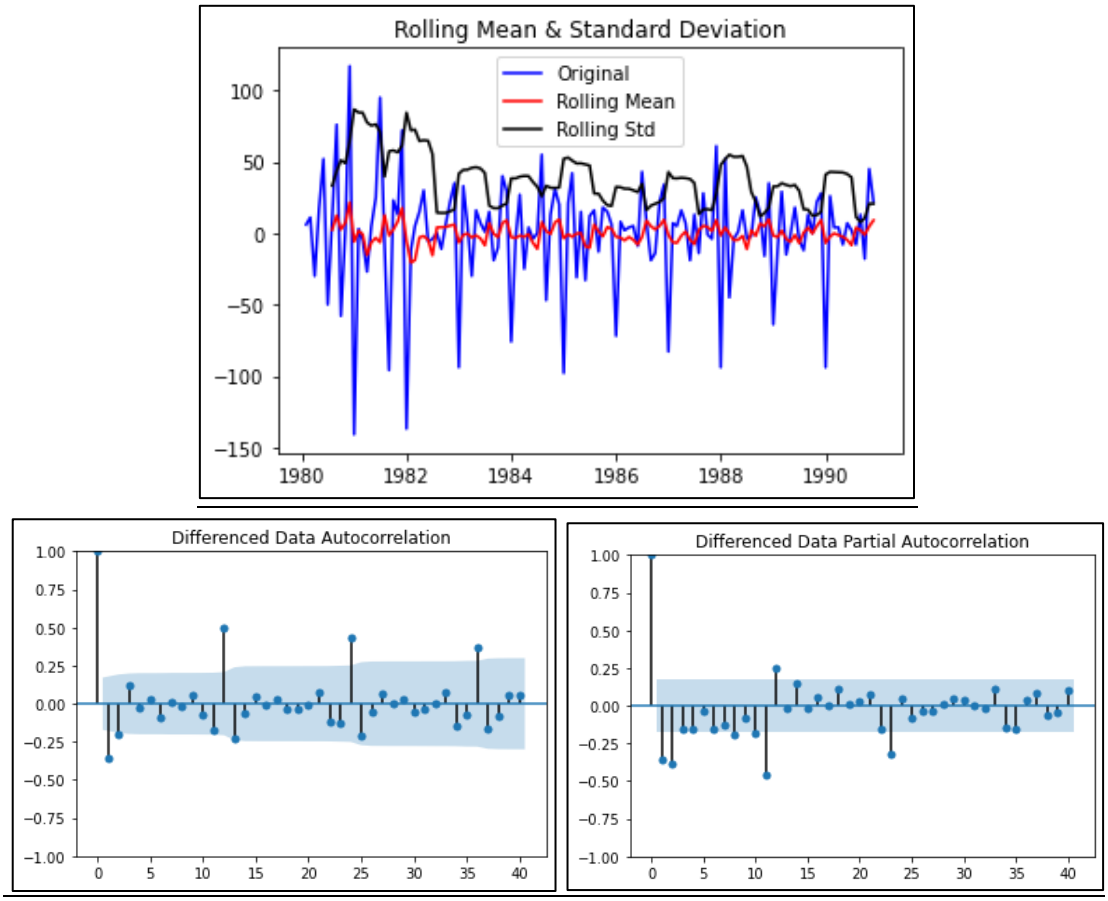

Fig 1.25: Rolling mean, std. deviation, PACF and ACF plots

- It can be inferred that there is a seasonality after every 12 months which is recurring in nature.
- Here, p =2 as 3 lag are falling outside the significance blue band.
- d=1 as order of differencing is 1 as original series was non-stationary but became stationary after differencing)
- q=2 as two lags are falling above the significance blue band. Seasonality after 12 lags is seen in plots.
- ACF & PACF plot are done using 95% confidence interval bands.
- Therefore, (p,d,q)(P,D,Q) = (2,1,2)(2,1,0,12)

- **Step 3:** Building model

```
manual_ARIMA = ARIMA(train_a_sarima['Rose'], order=(2,1,2),freq='MS')

results_manual_ARIMA = manual_ARIMA.fit()
```

- **Step 4:** ARIMAX Result:

SARIMAX Results

Dep. Variable:	Rose	No. Observations:	132
Model:	ARIMA(2, 1, 2)	Log Likelihood	-635.935
Date:	Sun, 20 Feb 2022	AIC	1281.871
Time:	22:16:47	BIC	1296.247
Sample:	01-01-1980	HQIC	1287.712
	- 12-01-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.4540	0.469	-0.969	0.333	-1.372	0.464
ar.L2	0.0001	0.170	0.001	0.999	-0.334	0.334
ma.L1	-0.2541	0.459	-0.554	0.580	-1.154	0.646
ma.L2	-0.5984	0.430	-1.390	0.164	-1.442	0.245
sigma2	952.1601	91.424	10.415	0.000	772.973	1131.347

Ljung-Box (L1) (Q):	0.02	Jarque-Bera (JB):	34.16
Prob(Q):	0.88	Prob(JB):	0.00
Heteroskedasticity (H):	0.37	Skew:	0.79
Prob(H) (two-sided):	0.00	Kurtosis:	4.94

- Table 1.18: ARIMAX Result

- **Step 6:** Residual Diagnostics:

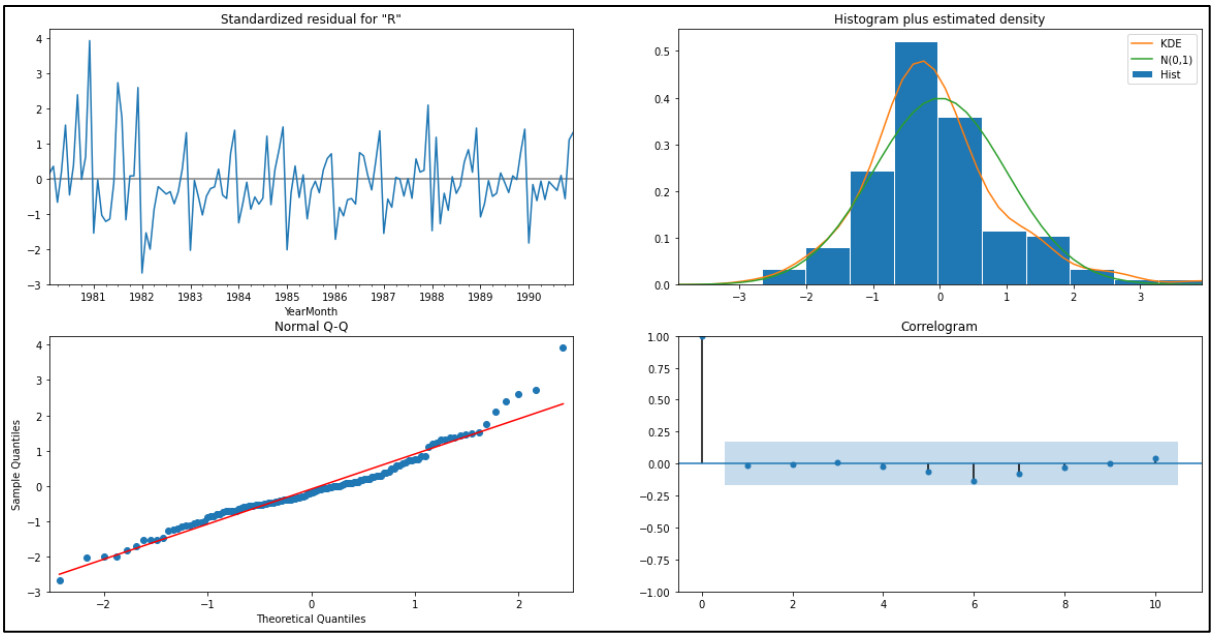

Table 1.26: Residual Diagnostics

- **Step 7:** Calculate the RMSE
  - RMSE Score: 36.87

➢ Cut-off based SARIMA Model

- **Step 1:** Import necessary libraries

```
import statsmodels.api as sm
```

- **Step 2:** Decide the cut-off points of ACF and PACF

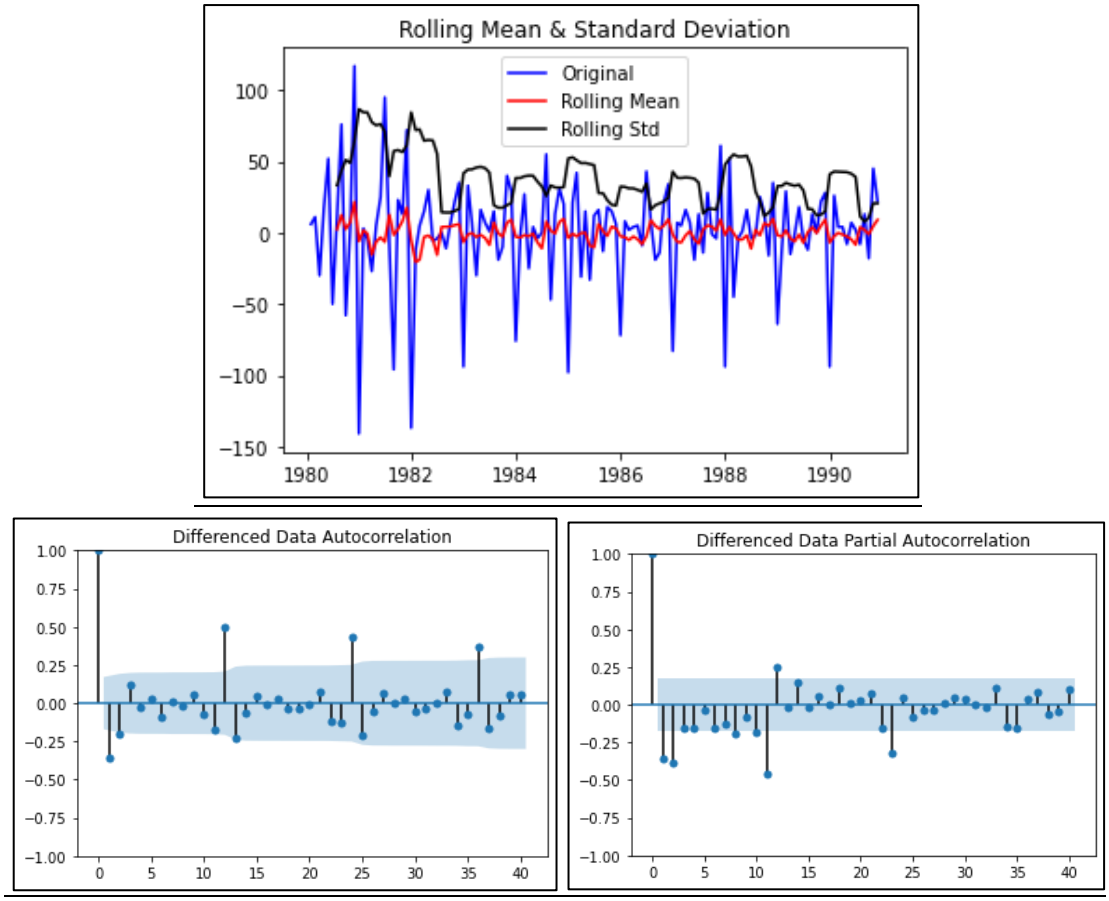

Fig 1.25: Rolling mean, std. deviation, PACF and ACF plots

- It can be inferred that there is a seasonality after every 12 months which is recurring in nature.
- Here, p =2 as 3 lag are falling outside the significance blue band.
- d=1 as order of differencing is 1 as original series was non-stationary but became stationary after differencing)
- q=2 as two lags are falling above the significance blue band. Seasonality after 12 lags is seen in plots.
- ACF & PACF plot are done using 95% confidence interval bands.
- Therefore, (p,d,q)(P,D,Q) = (2,1,2)(2,1,0,12)

- **Step 3:** Building model

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

- **Step 4:** SARIMAX Result:

SARIMAX Results

Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(2, 1, 2)x(2, 1, [], 12)	Log Likelihood	-390.390
Date:	Sun, 20 Feb 2022	AIC	794.780
Time:	22:16:50	BIC	812.508
Sample:	0	HQIC	801.938
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.2172	2.170	-0.100	0.920	-4.470	4.036
ar.L2	0.0743	0.202	0.369	0.712	-0.321	0.470
ma.L1	-0.7312	2.160	-0.339	0.735	-4.964	3.502
ma.L2	-0.3219	2.250	-0.143	0.886	-4.732	4.088
ar.S.L12	-0.3877	0.086	-4.491	0.000	-0.557	-0.218
ar.S.L24	-0.1506	0.063	-2.387	0.017	-0.274	-0.027
sigma2	235.8675	42.249	5.583	0.000	153.061	318.674

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	1.22
Prob(Q):	0.98	Prob(JB):	0.54
Heteroskedasticity (H):	0.87	Skew:	0.19
Prob(H) (two-sided):	0.69	Kurtosis:	3.42

Table 1.19: SARIMAX Result

- **Step 6:** Residual Diagnostics:

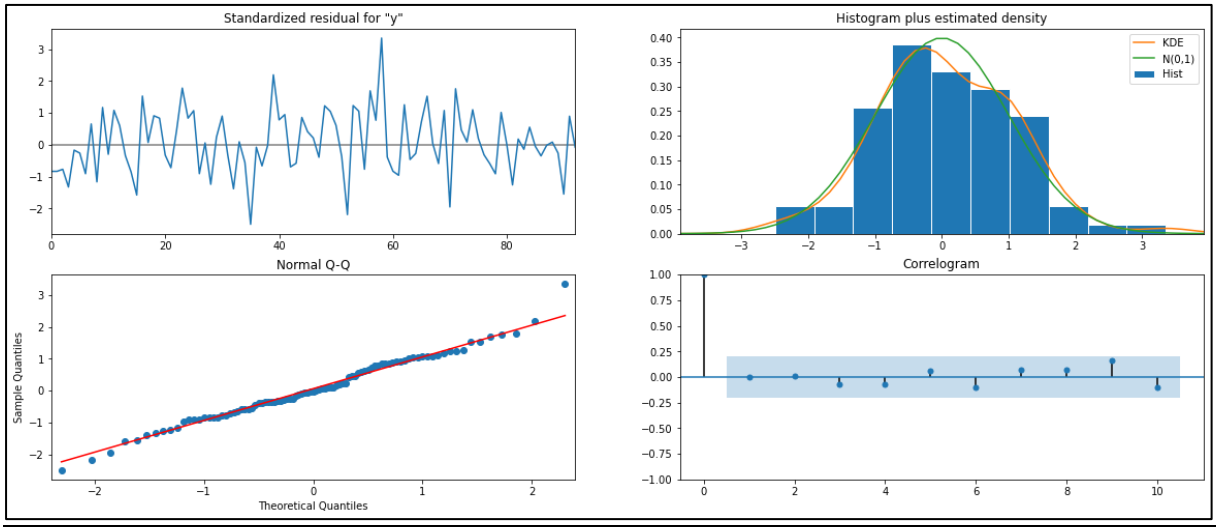

Fig 1.27: Residual Diagnostics

- **Step 7:** Calculate the RMSE
  - RMSE Score: 17.91

Q1.8 Build a table with all the models built along with their corresponding parameters and the respective RMSE values on the test data.

	Test RMSE
Alpha=0.1 ,Beta=0.2, Gamma=0.2, TripleExponentialSmoothing	9.223504
2pointTrailingMovingAverage	10.402622
4pointTrailingMovingAverage	13.176123
6pointTrailingMovingAverage	13.299874
8pointTrailingMovingAverage	13.997297
Alpha=1.49, Beta= 3.07, Double Exp. Smoothing	15.268947
Simple Average	15.759783
(0, 1, 2)(2, 1, 2, 12) , SARIMA iteration Model	16.500245
(2,1,2)(2,1,0,12) SARIMA Manual Model	17.914615
Alpha =0.063, Beta =0.054, Gamma =1.496e-07, Triple Exp. Smoothing	20.370572
Alpha=0.075,SimpleExponentialSmoothing	36.432370
Alpha=0.098, Simple Exp. Smoothing	36.796242
(2,1,2) ARIMA Manual Model	36.871197
(0,1,2) ARIMA Manual Model	37.306480
LR_Model	71.596828
Naive_Model	79.718773
Alpha=0.1,Beta=0.1,DoubleExponentialSmoothing	84.295759

Table 1.20: Models with RMSE values

- We can conclude from above that Tripple Exponential Smoothing model with alpha=0.1, beta=0.2 and gamma=0.2 is the best suited with RMSE value of 9.22.

Q1.9 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.

8. Triple Exponential Smoothing Model

- **Step 1:** Import necessary libraries
  from statsmodels.tsa.api import ExponentialSmoothing

- **Step 2:** Apply Triple Exponential Smoothing Model by using function
  TES_model = ExponentialSmoothing(TES_train['Rose'],trend='additive',seasonal='Multiplicative')

  TES_model_autofit = TES_model.fit(optimized=True)

- **Step 3:** Forecasting on test data:

YearMonth	Rose	Predict
1991-01-01	54.0	56.411071
1991-02-01	55.0	63.801331
1991-03-01	66.0	69.500435
1991-04-01	65.0	60.599603
1991-05-01	60.0	67.889542

Table 1.13: Forecasting on test data

- **Step 4:** Plot the Forecasts:

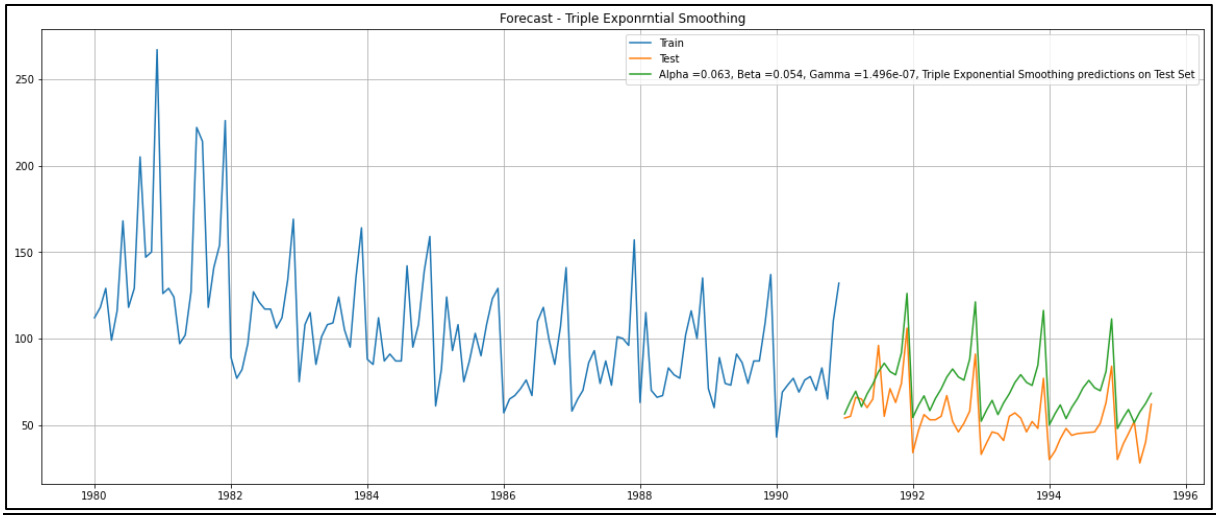

Fig 1.21: Forecast – Triple exponential Smoothing

- **Step 5:** Calculate the RMSE
  - RMSE Score: 20.37

- **Step 6: Iterate through different parameters (alpha, beta, gamma)**

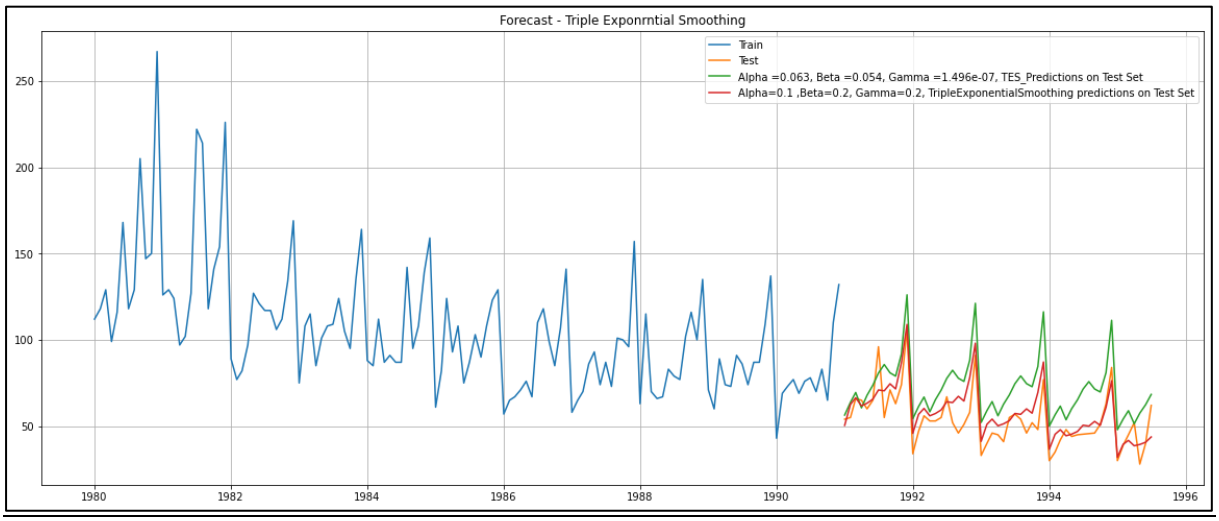

Fig 1.22: Forecast – Triple exponential Smoothing with different params

  - New RMSE Score: 9.22

- **Step 7:** Predictions

1995-08-01	47.607992
1995-09-01	48.284484
1995-10-01	50.279657
1995-11-01	58.461229
1995-12-01	82.116597
1996-01-01	31.696348
1996-02-01	39.431810
1996-03-01	45.360026
1996-04-01	46.803005
1996-05-01	40.722234
1996-06-01	46.994555
1996-07-01	54.041101

Freq: MS, dtype: float64

Table 1.21: Forecast for the next 12 months

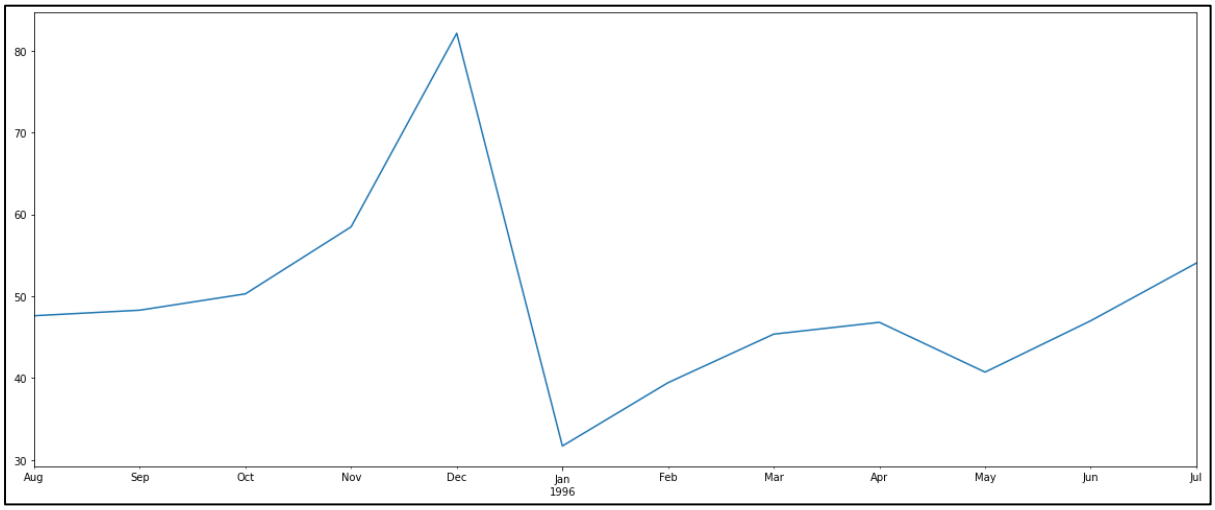

Fig 1.28: Predictions

- **Step 8:** Forecast for the next 12 months

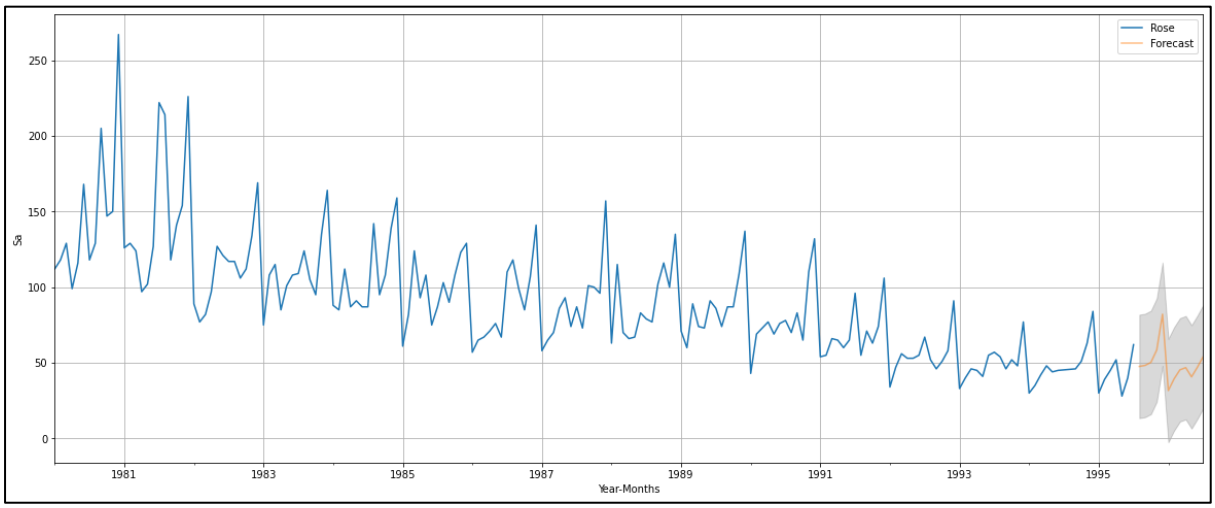

Fig 1.29: Forecast for the next 12 months

- Sa;es for the forecasted is more or less similar to the sales of year 1995.

Q1.10 Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales.

- Triple exponential smoothing model seems to be best suited for series with random fluctuations.
- In the month of December, rose wine sales peaked which is due to festive season (might be Christmas).
- Trend component in the series shows downtrend trend throughout the year.
- Trend component is strong and have heavy weightage in the prediction.

Suggestions for the ABC Company:

1. Looking at the sales we can say that there is continuous downtrend.
   So, company should decide whether to continue such variety of wine. Because managing a product takes too much of money and efforts and with such a low sales it becomes very difficult to be profitable.
2. Company should see improve the sales in the 1st three quarter of the year so as to gain more revenue.
3. Brand building is another key to enhance the revenue.

End of the Report