BUSINESS REPORT

Time Series Forecasting

Dhruv Dosad

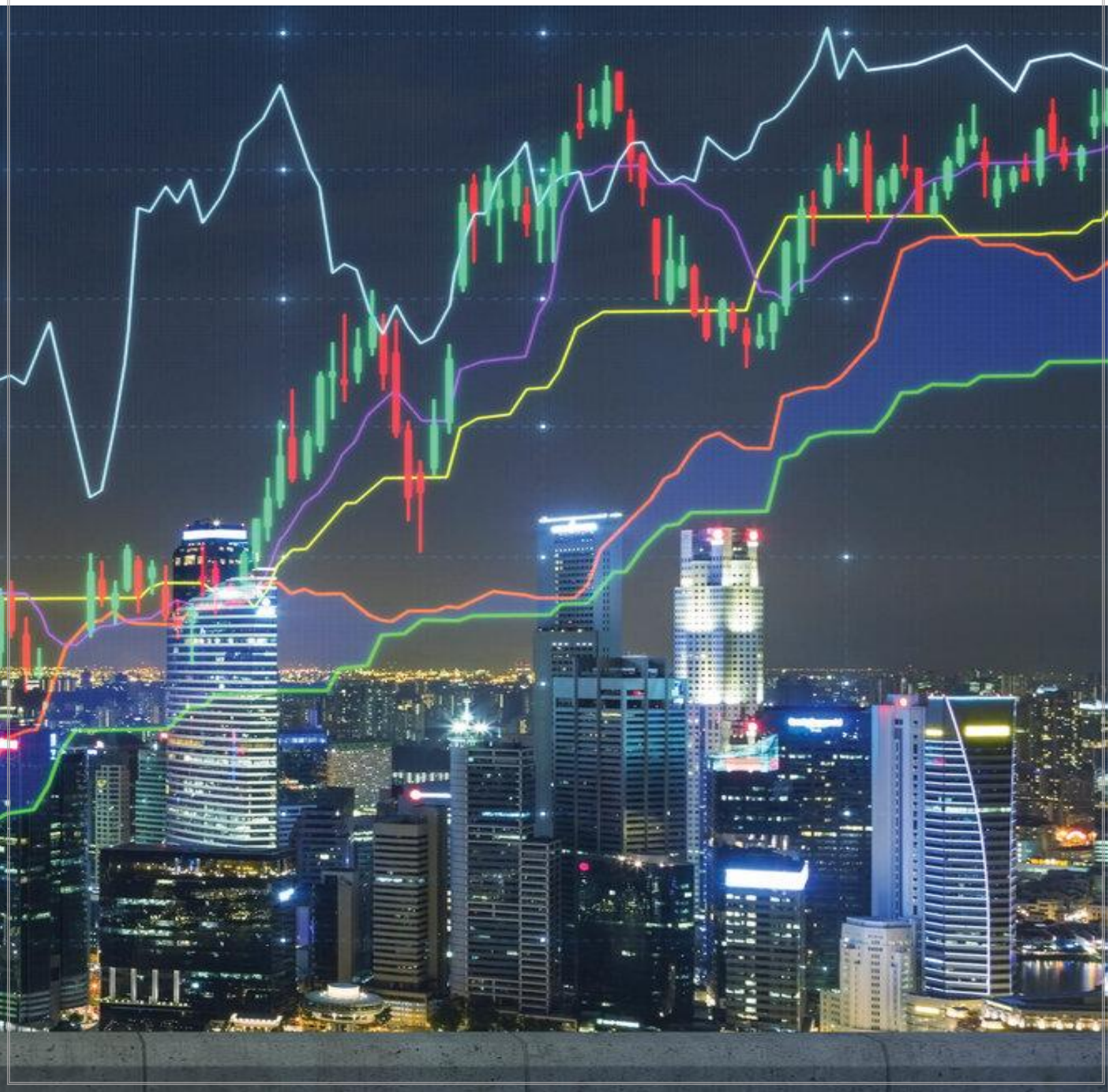

TABLE OF CONTENTS

Problem I: Sparkling Wine Sales	01–42
1.1 Read the data as an appropriate Time Series data and plot the data.	01
1.2 Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.	03
1.3 Split the data into training and test. The test data should start in 1991.	10
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.	11
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.	25
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.	27
1.7 Build a table with all the models built along with their corresponding parameters and the respective RMSE values on the test data.	36
1.8 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.	37
1.9 Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales.	41
Problem II: Rose Wine Sales	43–82
2.1 Read the data as an appropriate Time Series data and plot the data.	43
2.2 Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.	44

2.3 Split the data into training and test. The test data should start in 1991.	52
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.	53
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.	65
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.	67
2.7 Build a table with all the models built along with their corresponding parameters and the respective RMSE values on the test data.	76
2.8 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.	77
2.9 Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales.	81

List of Figures

Fig 1.1 Time Series Plot	02
Fig 1.2 Histogram & Boxplot	03
Fig 1.3 Spread of Sales Across Different Years	04
Fig 1.4 Spread of Sales Across Different Months	04
Fig 1.5 Distribution of time series across different months	05
Fig 1.6 Year–on–Year Monthly Comparison	06
Fig 1.7 Empirical Cumulative Distribution Plot	07
Fig 1.8 Year–on–Year Average Sales	07
Fig 1.9 Decomposed Time Series– Additive	08
Fig 1.10 Histogram of Residuals – Additive Decomposition	08
Fig 1.11 Decomposed Time Series– Multiplicative	09
Fig 1.12 Residuals Histogram– Multiplicative Decomposition	09
Fig 1.13 Train & Test Split Time Series	11
Fig 1.14 Time Series Plot: Linear Regression	12
Fig 1.15 Time Series Plot: Naïve	13

Fig 1.16 Time Series Plot: Simple Average	14
Fig 1.17 Time Series Plot: Moving Average on Whole data	15
Fig 1.18 Time Series Plot: Moving Average	16
Fig 1.19 Time Series Plot: 2–Point Moving Average	16
Fig 1.20 Time Series Plot: Simple Exponential Smoothing Alpha = 0.07	18
Fig 1.21 Time Series Plot: Simple Exponential Smoothing Alpha = 0.02	19
Fig 1.22 Time Series Plot: Double Exponential Smoothing Alpha = 0.665, Beta= 0.0001	20
Fig 1.23 Time Series Plot: Double Exponential Smoothing Alpha = 0.02, Beta= 0.38	21
Fig 1.24 Time Series Plot: Triple Exponential Smoothing Alpha = 0.111, Beta= 0.049, Gamma= 0.362	22
Fig 1.25 Time Series Plot: Triple Exponential Smoothing Alpha = 0.01, Beta= 0.04, Gamma= 0.25	24
Fig 1.26 Time Series Plot: Model Comparisions	25
Fig 1.27 Stationarity of Whole Data Using AD Fuller Test	26
Fig 1.28 Stationarity of Whole Data Using AD Fuller Test at Differencing of Order 1	26
Fig 1.29 Stationarity of Training Data Using AD Fuller Test	27
Fig 1.30 Stationarity of Training Data Using AD Fuller Test at Differencing of Order 1	27
Fig 1.31 Autocorrelation Plot	28
Fig 1.32 Differenced Autocorrelation Plot	28
Fig 1.33 Diagnostic Plot: Automated ARIMA (2, 1, 2)	29
Fig 1.34 Time Series Plot: Automated ARIMA (2, 1, 2)	30
Fig 1.35 Differenced Autocorrelation Plot : SARIMA	31
Fig 1.36 Diagnostic Plot: Automated SARIMA (1, 1, 2)(1, 0, 2, 12)	32
Fig 1.37 Time Series Plot: Automated SARIMA (1, 1, 2)(1, 0, 2, 12)	33
Fig 1.38 Time Series Plot: Train Data	33
Fig 1.39 Time Series Plot: Test Data	34
Fig 1.40 Stationarity of Differenced Training Data Using AD Fuller Test (D=1)	34
Fig 1.41 Diagnostic Plot: Automated SARIMA(0, 0, 2)(0, 1, 2, 12)	35
Fig 1.42 Time Series Plot: Automated SARIMA(0, 0, 2)(0, 1, 2, 12)	36
Fig 1.43 Forecasted Plot : Triple Exponential Model with Alpha = 0.01, Beta = 0.04, Gamma = 0.25	38
Fig 1.44 Stationarity of Differenced Data Using AD Fuller (D=12)	39
Fig 1.45 Diagnostic Plot: Automated SARIMA(0, 0, 2)(0, 1, 2, 12)	39
Fig 1.46 Forecasted Plot : SARIMA(0, 0, 2)(0, 1, 2, 12)	40
Fig 2.1a Time Series Plot	44
Fig 2.1b Time Series Plot Post Null Treatment	45
Fig 2.2 Histogram & Boxplot	46
Fig 2.3 Spread of Sales Across Different Years	46
Fig 2.4 Spread of Sales Across Different Months	46
Fig 2.5 Distribution of time series across different months	47
Fig 2.6 Year–on–Year Monthly Comparison	48
Fig 2.7 Empirical Cumulative Distribution Plot	48
Fig 2.8 Year–on–Year Average Sales	49
Fig 2.9 Decomposed Time Series– Additive	49
Fig 2.10 Histogram of Residuals – Additive Decomposition	50

Fig 2.11 Decomposed Time Series– Multiplicative	50
Fig 2.12 Residuals Histogram— Multiplicative Decomposition	51
Fig 2.13 Train & Test Split Time Series	52
Fig 2.14 Time Series Plot: Linear Regression	53
Fig 2.15 Time Series Plot: Naïve	54
Fig 2.16 Time Series Plot: Simple Average	55
Fig 2.17 Time Series Plot: Moving Average on Whole data	56
Fig 2.18 Time Series Plot: Moving Average	56
Fig 2.19 Time Series Plot: 2–Point Moving Average	57
Fig 2.20 Time Series Plot: Simple Exponential Smoothing Alpha = 0.099	58
Fig 2.21 Time Series Plot: Simple Exponential Smoothing Alpha = 0.07	59
Fig 2.22 Time Series Plot: Double Exponential Smoothing Alpha = 1.49*10^–8, Beta = 5.44*10^–9	60
Fig 2.23 Time Series Plot: Double Exponential Smoothing Alpha = 0.04 & Beta = 0.47	61
Fig 2.24 Time Series Plot: Triple Exponential Smoothing Alpha = 0.077, Beta= 0.039, Gamma= 0.0008	62
Fig 2.25 Time Series Plot: Triple Exponential Smoothing Alpha = 0.04, Beta = 0.52, Gamma = 0.10	63
Fig 2.26 Time Series Plot: Model Comparisions	64
Fig 2.27 Stationarity of Whole Data Using AD Fuller Test	65
Fig 2.28 Stationarity of Whole Data Using AD Fuller Test at Differencing of Order 1	66
Fig 2.29 Stationarity of Training Data Using AD Fuller Test	66
Fig 2.30 Stationarity of Training Data Using AD Fuller Test at Differencing of Order 1	67
Fig 2.31 Autocorrelation Plot	67
Fig 2.32 Differenced Autocorrelation Plot	68
Fig 2.33 Diagnostic Plot: Automated ARIMA (0, 1, 2)	69
Fig 2.34 Time Series Plot: Automated ARIMA (0, 1, 2)	70
Fig 2.35 Differenced Autocorrelation Plot : SARIMA	71
Fig 2.36 Diagnostic Plot: Automated SARIMA (0, 1, 2)(2, 0, 2, 12)	72
Fig 2.37 Time Series Plot: Automated SARIMA (0, 1, 2)(2, 0, 2, 12)	72
Fig 2.38 Time Series Plot: Train Data	73
Fig 2.39 Time Series Plot: Test Data	73
Fig 2.40 Stationarity of Differenced Training Data Using AD Fuller Test (D=1)	74
Fig 2.41 Diagnostic Plot: Automated SARIMA(0, 1, 2)(2, 1, 2, 12)	75
Fig 2.42 Time Series Plot: Automated SARIMA (0, 1, 2)(2, 1, 2, 12)	75
Fig 2.43 Forecasted Plot: Triple Exponential Model with Alpha = 0.04, Beta = 0.52, Gamma = 0.10	79
Fig 2.44 Stationarity of Differenced Data Using AD Fuller (D=12)	79
Fig 2.45 Diagnostic Plot: Automated SARIMA(0, 1, 2)(2, 1, 2, 12)	80
Fig 2.46 Forecasted Plot: Automated SARIMA(0, 1, 2)(2, 1, 2, 12)	81

List of Tables

Table 1.1 First 5 Samples of the Dataset	01
Table 1.2 Last 5 Samples of the Dataset	01
Table 1.3 First 5 Samples of the Converted Dataset	01
Table 1.4 Info of the Dataset	02
Table 1.5 Descriptive Statistics	03
Table 1.6 Year–on–Year Monthly Sales	06
Table 1.7 Decomposed Time Series Components	10
Table 1.8 Dimensions of Original, Test & Train Data	10
Table 1.9 Sample of Training Data	11
Table 1.10 Sample of Test Data	11
Table 1.11 Sample of LinearRegression Test & Train Data	12
Table 1.12 Model Performance Summary – Linear Regression	12
Table 1.13 Samples of Train & Test data– Naive Model	13
Table 1.14 Model Performance Summary – Naïve	13
Table 1.15 Samples of Train & Test Data for Simple Average	14
Table 1.16 Model Performance Summary – Simple Average	14
Table 1.17 Moving Average Sample on Training Data	15
Table 1.18 Model Performance Summary – Moving Averages	16
Table 1.19 Autofill Simple Exponential Smoothing Optimal Parameters	17
Table 1.20 Model Performance Summary – Simple Exponential Smoothing Alpha = 0.07	18
Table 1.21 Brute Force Simple Exponential Smoothing Parameters	18
Table 1.22 Model Performance Summary – Simple Exponential Smoothing Alpha = 0.02	19
Table 1.23 Autofill Double Exponential Smoothing Optimal Parameters	20
Table 1.24 Model Performance Summary – Double Exponential Smoothing Alpha = 0.665, Beta= 0.0001	20
Table 1.25 Brute Force Double Exponential Smoothing Parameters	21
Table 1.26 Model Performance Summary – Double Exponential Smoothing Alpha = 0.02, Beta= 0.38	21
Table 1.27 Autofill Triple Exponential Smoothing Parameters	22
Table 1.28 Model Performance Summary – Triple Exponential Smoothing Alpha = 0.111, Beta= 0.049, Gamma= 0.362	23
Table 1.29 Brute Force Triple Exponential Smoothing Parameters	23
Table 1.30 Model Performance Summary – Triple Exponential Smoothing Alpha = 0.01, Beta= 0.04, Gamma= 0.25	24
Table 1.31 ARIMA AIC Parameters	29
Table 1.32 Auto ARIMA Model Summary	29
Table 1.33 Model Performance Summary – Automated ARIMA (2, 1, 2)	30
Table 1.34 SARIMA AIC Parameters without Seasoning	31
Table 1.35 Auto SARIMA without Differencing Model Summary	32
Table 1.36 Model Performance Summary – Automated SARIMA (1, 1, 2)(1, 0, 2, 12)	33
Table 1.37 SARIMA AIC Parameters with Seasoning	35
Table 1.38 Auto SARIMA with Differencing Model Summary	35

Table 1.39 Model Performance Summary – Automated SARIMA(0, 0, 2)(0, 1, 2, 12)	36
Table 1.40 Model Performance Summary – Consolidated	36
Table 1.41 Best Performing Models	37
Table 1.42 Forecast Results – Triple Exponential Model with Alpha = 0.01, Beta = 0.04, Gamma = 0.25	38
Table 1.43 Auto SARIMA Forecast Model Summary	39
Table 1.44 Forecast Results – SARIMA(0, 0, 2)(0, 1, 2, 12)	40
Table 2.1 First 5 and Last 5 Samples of the Dataset	43
Table 2.2 First 5 Samples of the Converted Dataset	43
Table 2.3 Info of the Dataset	44
Table 2.4 Missing Values Before & After Treatment	45
Table 2.5 Descriptive Statistics	45
Table 2.6 Year–on–Year Monthly Sales	47
Table 2.7 Decomposed Time Series Components	51
Table 2.8 Dimensions of Original, Test & Train Data	52
Table 2.9 Sample of Training Data	52
Table 2.10 Sample of Test Data	52
Table 2.11 Sample of LinearRegression Test & Train Data	53
Table 2.12 Model Performance Summary – Linear Regression	53
Table 2.13 Samples of Train & Test data– Naive Model	54
Table 2.14 Model Performance Summary – Naïve	54
Table 2.15 Samples of Train & Test Data for Simple Average	55
Table 2.16 Model Performance Summary – Simple Average	55
Table 2.17 Moving Average Sample on Training Data	56
Table 2.18 Model Performance Summary – Moving Averages	57
Table 2.19 Autofill Simple Exponential Smoothing Optimal Parameters	58
Table 2.20 Model Performance Summary – Simple Exponential Smoothing Alpha = 0.099	58
Table 2.21 Brute Force Simple Exponential Smoothing Parameters	59
Table 2.22 Model Performance Summary – Simple Exponential Smoothing Alpha = 0.07	59
Table 2.23 Autofill Double Exponential Smoothing Optimal Parameters	60
Table 2.24 Model Performance Summary – Double Exponential Smoothing Alpha = 1.49*10^–8, Beta = 5.44*10^–9	60
Table 2.25 Brute Force Double Exponential Smoothing Parameters	61
Table 2.26 Model Performance Summary – Double Exponential Smoothing Alpha = 0.04, Beta= 0.47	61
Table 2.27 Autofill Triple Exponential Smoothing Parameters	62
Table 2.28 Model Performance Summary – Triple Exponential Smoothing Alpha = 0.077, Beta= 0.039, Gamma= 0.0008	63
Table 2.29 Brute Force Triple Exponential Smoothing Parameters	63
Table 2.30 Model Performance Summary – Triple Exponential Smoothing Alpha = 0.04, Beta = 0.52, Gamma = 0.10	64
Table 2.31 ARIMA AIC Parameters	68
Table 2.32 Auto ARIMA Model Summary	69
Table 2.33 Model Performance Summary – Automated ARIMA (0, 1, 2)	70

Table 2.34 SARIMA AIC Parameters without Seasoning	71
Table 2.35 Auto SARIMA without Differencing Model Summary	72
Table 2.36 Model Performance Summary – Automated SARIMA (0, 1, 2)(2, 0, 2, 12)	73
Table 2.37 SARIMA AIC Parameters with Seasoning	74
Table 2.38 Auto SARIMA with Differencing Model Summary	75
Table 2.39 Model Performance Summary – Automated SARIMA (0, 1, 2)(2, 1, 2, 12)	76
Table 2.40 Model Performance Summary – Consolidated	76
Table 2.41 Best Performing Models	77
Table 2.42 Forecast Results – Triple Exponential Model with Alpha = 0.04, Beta = 0.52, Gamma = 0.10	78
Table 2.43 Auto SARIMA Forecast Model Summary	80
Table 2.44 Forecast Results – Automated SARIMA(0, 1, 2)(2, 1, 2, 12)	80

Problem 1: Sparkling Wine Sales

Problem Statement:

As an analyst in the ABC Estate Wines, your task is to analyse and forecast Wine Sales in the 20th century.

Data set for the Problem: Sparkling.csv

Data Dictionary:

YearMonth : Month & Year of the sale

Sparkling: Total Number of Sparkling Wine sales in particular Month–Year

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

Basic Information about the dataset

- **Sample of the dataset:** First & last 5 values 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 First 5 Samples of the Dataset

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

Table 1.2 Last 5 Samples of the Dataset

- Converting the **YearMonth** Column to **DatetimeIndex & droping** default index. Sample:

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

Table 1.3 First 5 Samples of the Converted Dataset

- **Information about the dataset:**

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

Table 1.4 Info of the Dataset

- The DataFrame has **187 entries** with a **DatetimeIndex** ranging from **January 1980 to July 1995.**
- The **'Sparkling'** column is of **integer type** (int64), and it has **187 non-null values.**

- **Time Series Plot:**

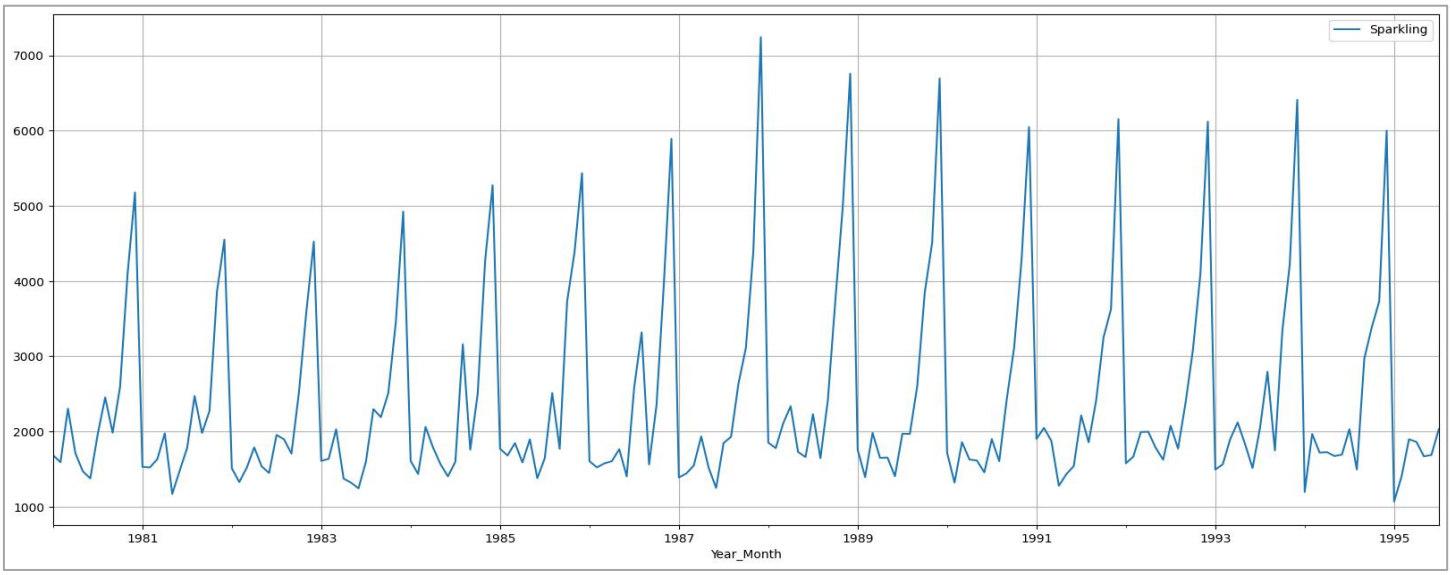

Fig 1.1 Time Series Plot

- **Increasing Trend:** The **sales of Sparkling** wine have been steadily **increasing** over the years, indicating a positive trend in customer demand.
- **Seasonal Patterns:** We observe that there are specific periods each year when sales spike, **especially during November and December.** These peaks might be due to the holiday season, when people tend to buy more Sparkling products for celebrations
- Historical data will enable accurate forecasting for better planning.

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

- **Missing Values:**
  - There are no missing values in the dataset
- **Duplicate Values:**
  - The dataset shows 11 Duplicates rows for Sparkling wine sales, however, when checked further, these were the same no of sales at different year. Hence, we conclude that **there are no duplicate values**
- **Descriptive Statistics:**

	count	mean	std	min	25%	50%	75%	max
Sparkling	187.0	2402.417112	1295.11154	1070.0	1605.0	1874.0	2549.0	7242.0

Table 1.5 Descriptive Statistics

  - The dataset contains **187 observations** of sparkling wine sales.
  - On **average**, there are approx. **2402 sales**
  - **Half** of the sales **fall below 1874**, indicating a **balanced distribution** around the median value.
  - The sales data has a **moderate level of variability**, with a **standard deviation of about 1295**.
  - The **minimum** recorded sales for sparkling products are **1070**, while the **maximum** is **7242**.

- **Histogram & Boxplot**

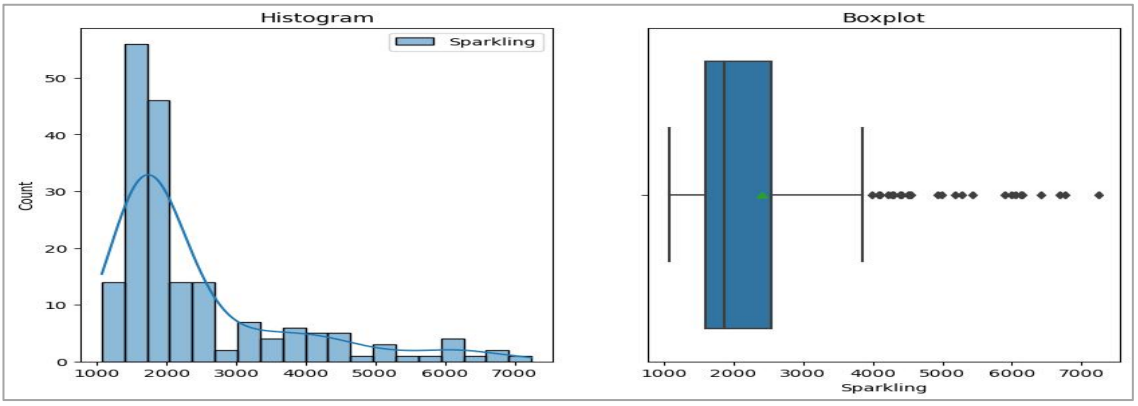

Fig 1.2 Histogram & Boxplot

  - The dataset is right skewed with the presence of outliers on the right tail

- **Spread of Sales: Year-on-Year Boxplot**

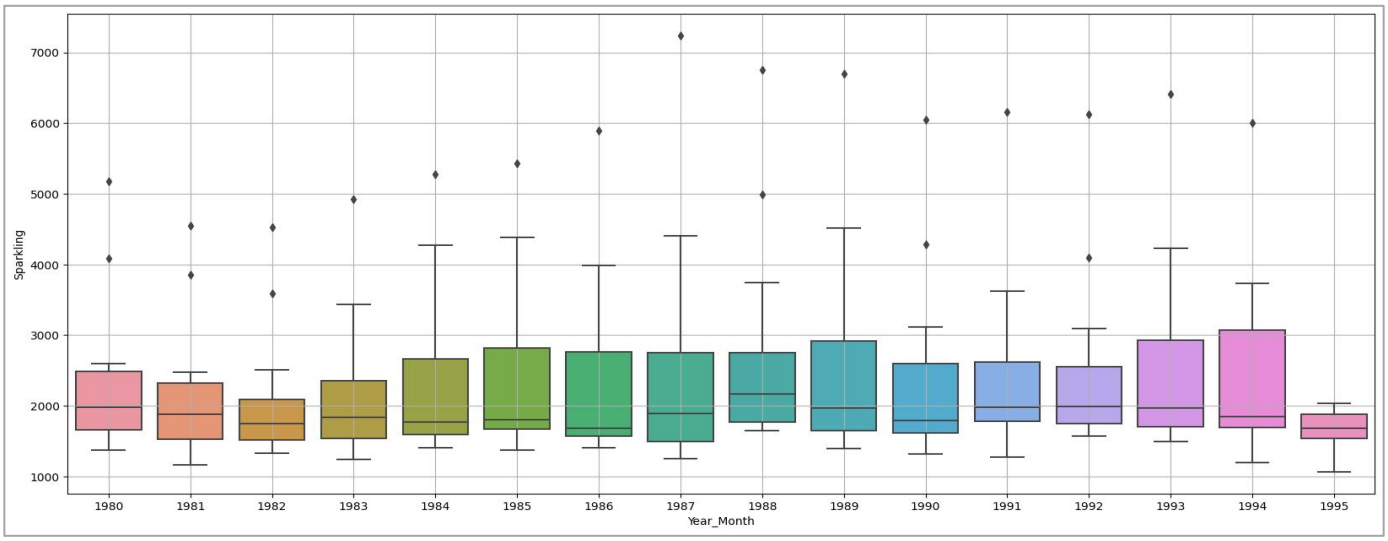

Fig 1.3 Spread of Sales Across Different Years

- Sales of Sparkling wine has been **increasing**, with median sales rising over time.
- The dataset is **skewed** with the presence of **outliers on the right tail.**
- There is significant variability in sales from year to year, with a large interquartile range.
- There are a few outliers in the data, which could be due to special promotions, holidays or other occasions

➢ **Spread of Sales: Month–on– Month**

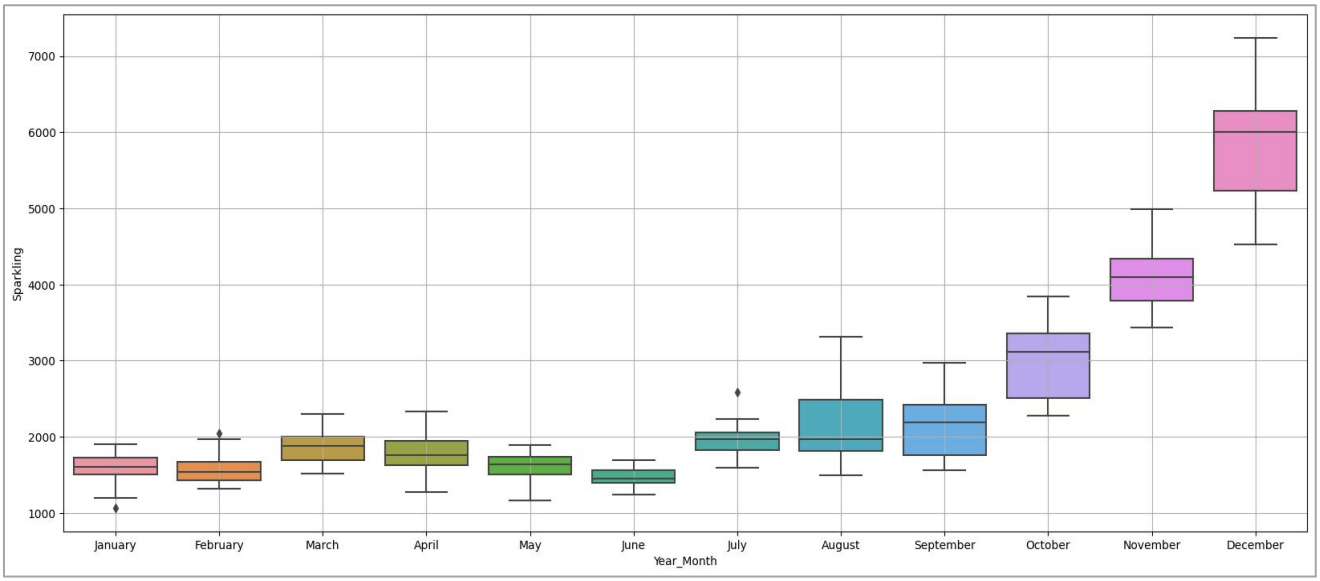

Fig 1.4 Spread of Sales Across Different Months

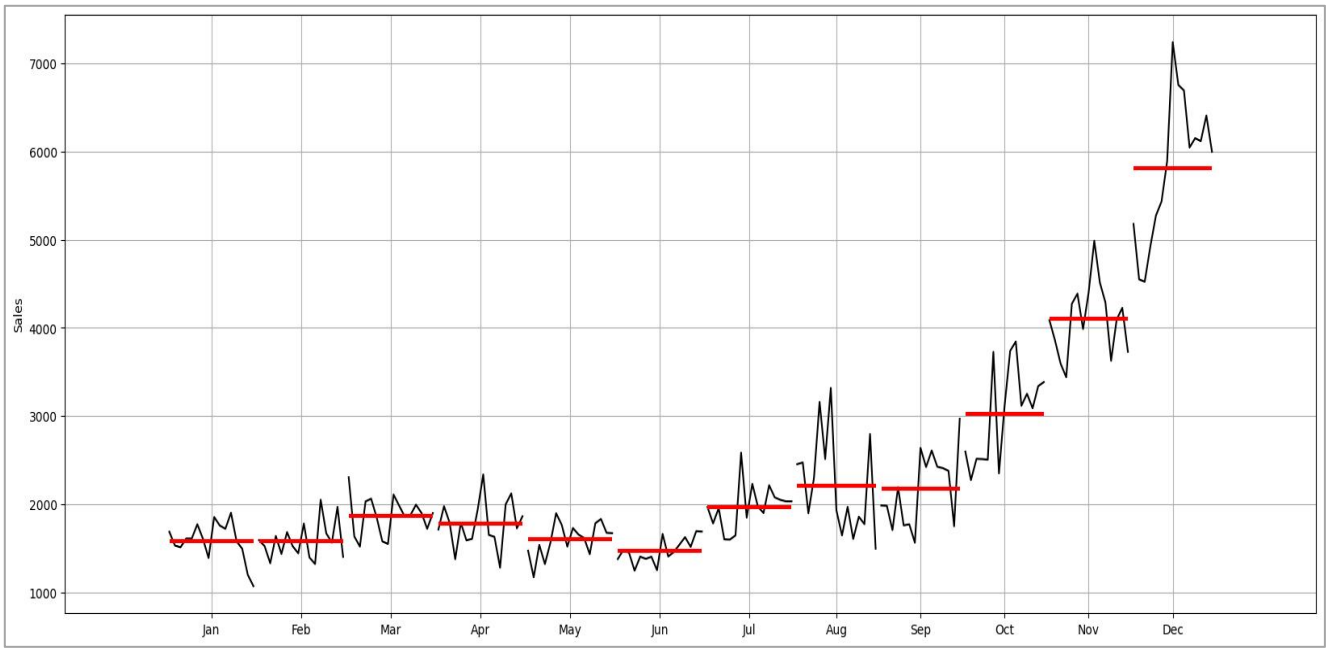

Fig 1.5 Distribution of time series across different months

- The **Month** Plot provides insights into the distribution of the Time Series data across different months, with the **red line representing the median value**.
- Upon analyzing the Month Plot, we can observe a **constant (stable) trend** that remains consistent across all the years for each month, except for December. Additionally, the plot shows clear **seasonal patterns** across the months.
- In **December**, the sales initially **dip**, then experience an **increase**, followed by another drop, eventually stabilizing over the years. This pattern in December's sales aligns with the overall trend observed in the entire time series.
- The evidence suggests that **December** might be a significant month influencing the trend in the time series, given that it has the **maximum sales** compared to other months.

➢ **Spread of Sales: Year-on-Year & Monthly Comparison:**

Year_Month	1	2	3	4	5	6	7	8	9	10	11	12
Year_Month												
1980	1686.0	1591.0	2304.0	1712.0	1471.0	1377.0	1966.0	2453.0	1984.0	2596.0	4087.0	5179.0
1981	1530.0	1523.0	1633.0	1976.0	1170.0	1480.0	1781.0	2472.0	1981.0	2273.0	3857.0	4551.0
1982	1510.0	1329.0	1518.0	1790.0	1537.0	1449.0	1954.0	1897.0	1706.0	2514.0	3593.0	4524.0
1983	1609.0	1638.0	2030.0	1375.0	1320.0	1245.0	1600.0	2298.0	2191.0	2511.0	3440.0	4923.0
1984	1609.0	1435.0	2061.0	1789.0	1567.0	1404.0	1597.0	3159.0	1759.0	2504.0	4273.0	5274.0
1985	1771.0	1682.0	1846.0	1589.0	1896.0	1379.0	1645.0	2512.0	1771.0	3727.0	4388.0	5434.0
1986	1606.0	1523.0	1577.0	1605.0	1765.0	1403.0	2584.0	3318.0	1562.0	2349.0	3987.0	5891.0
1987	1389.0	1442.0	1548.0	1935.0	1518.0	1250.0	1847.0	1930.0	2638.0	3114.0	4405.0	7242.0
1988	1853.0	1779.0	2108.0	2336.0	1728.0	1661.0	2230.0	1645.0	2421.0	3740.0	4988.0	6757.0
1989	1757.0	1394.0	1982.0	1650.0	1654.0	1406.0	1971.0	1968.0	2608.0	3845.0	4514.0	6694.0
1990	1720.0	1321.0	1859.0	1628.0	1615.0	1457.0	1899.0	1605.0	2424.0	3116.0	4286.0	6047.0
1991	1902.0	2049.0	1874.0	1279.0	1432.0	1540.0	2214.0	1857.0	2408.0	3252.0	3627.0	6153.0
1992	1577.0	1667.0	1993.0	1997.0	1783.0	1625.0	2076.0	1773.0	2377.0	3088.0	4096.0	6119.0
1993	1494.0	1564.0	1898.0	2121.0	1831.0	1515.0	2048.0	2795.0	1749.0	3339.0	4227.0	6410.0
1994	1197.0	1968.0	1720.0	1725.0	1674.0	1693.0	2031.0	1495.0	2968.0	3385.0	3729.0	5999.0
1995	1070.0	1402.0	1897.0	1862.0	1670.0	1688.0	2031.0	NaN	NaN	NaN	NaN	NaN

Table 1.6 Year–on–Year Monthly Sales

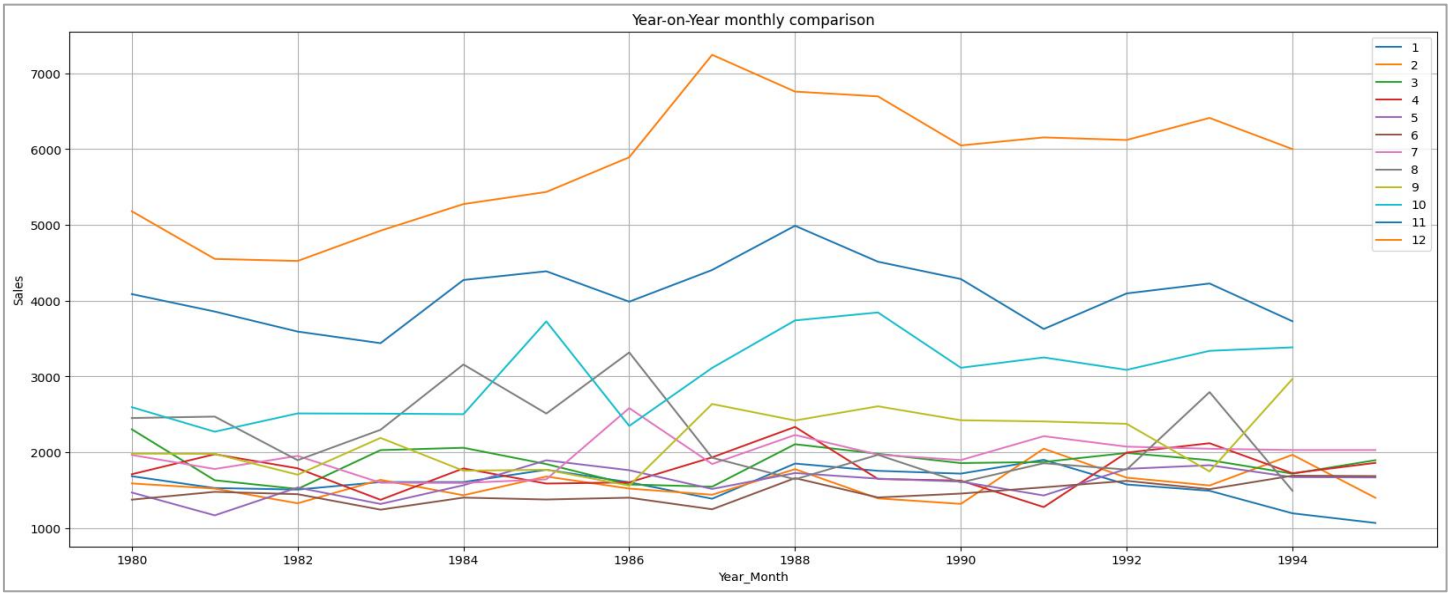

Fig 1.6 Year–on–Year Monthly Comparison

- The above plots show us the behavior of the Sparkling sales across various months.
- The sales are **highest in December**. The sales appear to **drop** in the month of **January** and are **stable till July**, with **seasonal patterns across the years**.
- There is **significant variability** in sales from month to month
- There are a few outliers in the data, which could be due to special promotions, holidays or other occasions

- **Empirical Cumulative Distribution Plot**

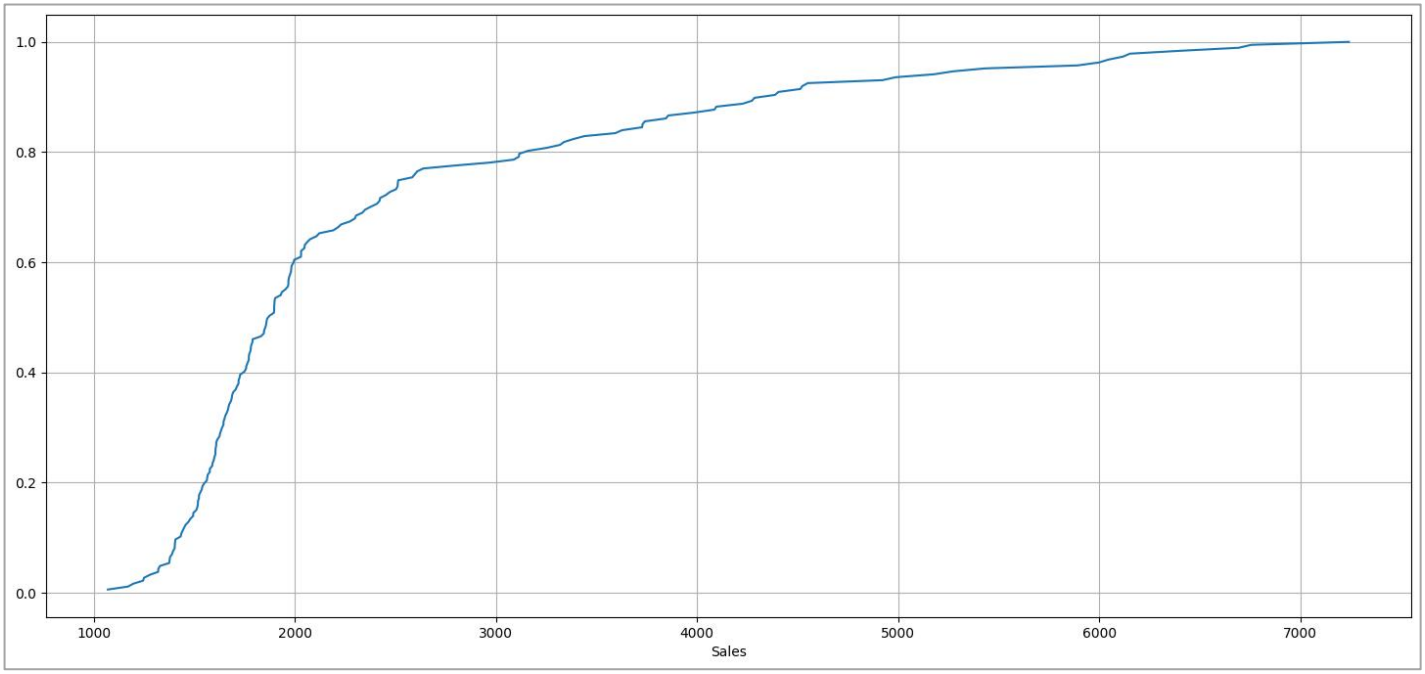

Fig 1.7 Empirical Cumulative Distribution Plot

- This graph tells us what **percentage** of data points refer to what number of Sales.
- The distribution of sales is skewed to the right. This means that there are more sales at the lower end of the distribution than at the higher end.

- **Average Sparkling Sales**

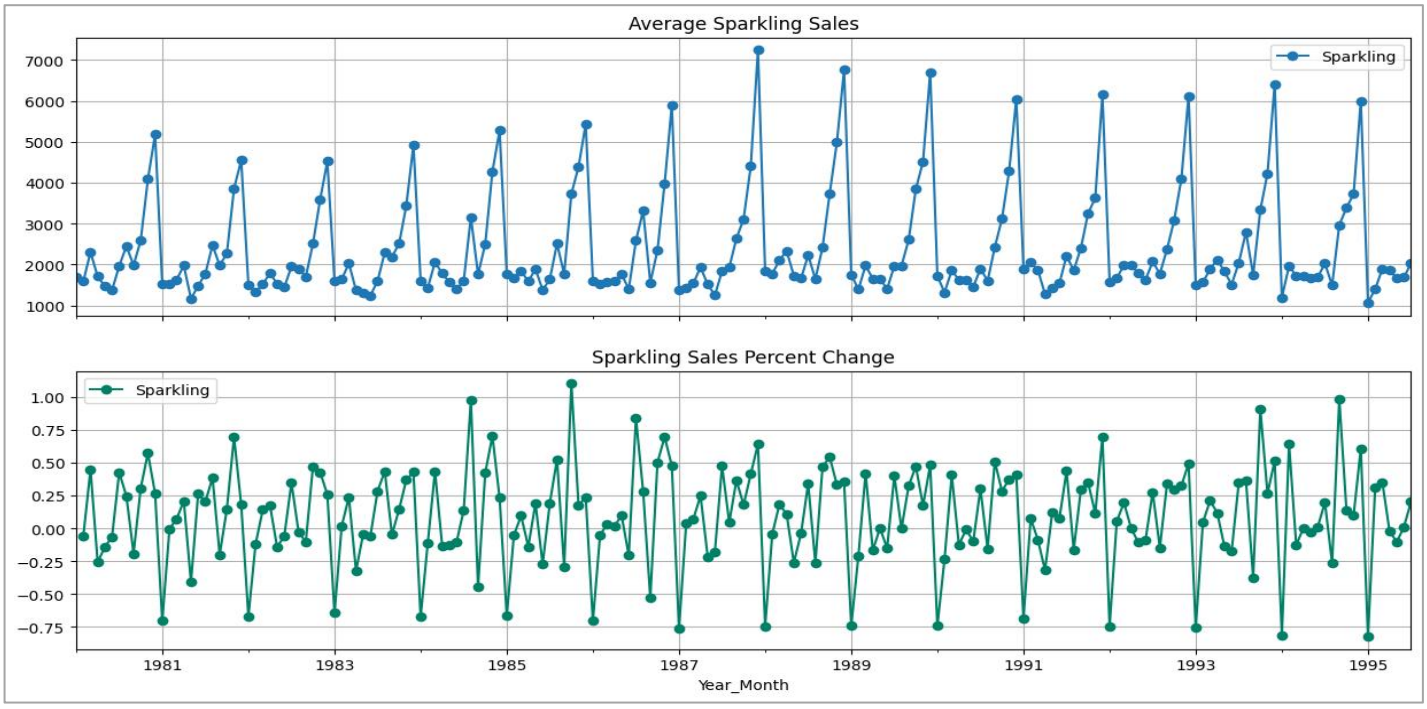

Fig 1.8 Year–on–Year Average Sales

- The **average sales are increasing** year–on–year. This is evident from the fact that the line graph is generally increasing.
- The year–on–year percentage change in sales is **positive** for most years, but there are some years with negative changes increasing, but there are some periods where it decreases.
- There is a seasonal pattern in sales, with sales being highest in December.

Decomposition:

➢ Additive Decomposition:

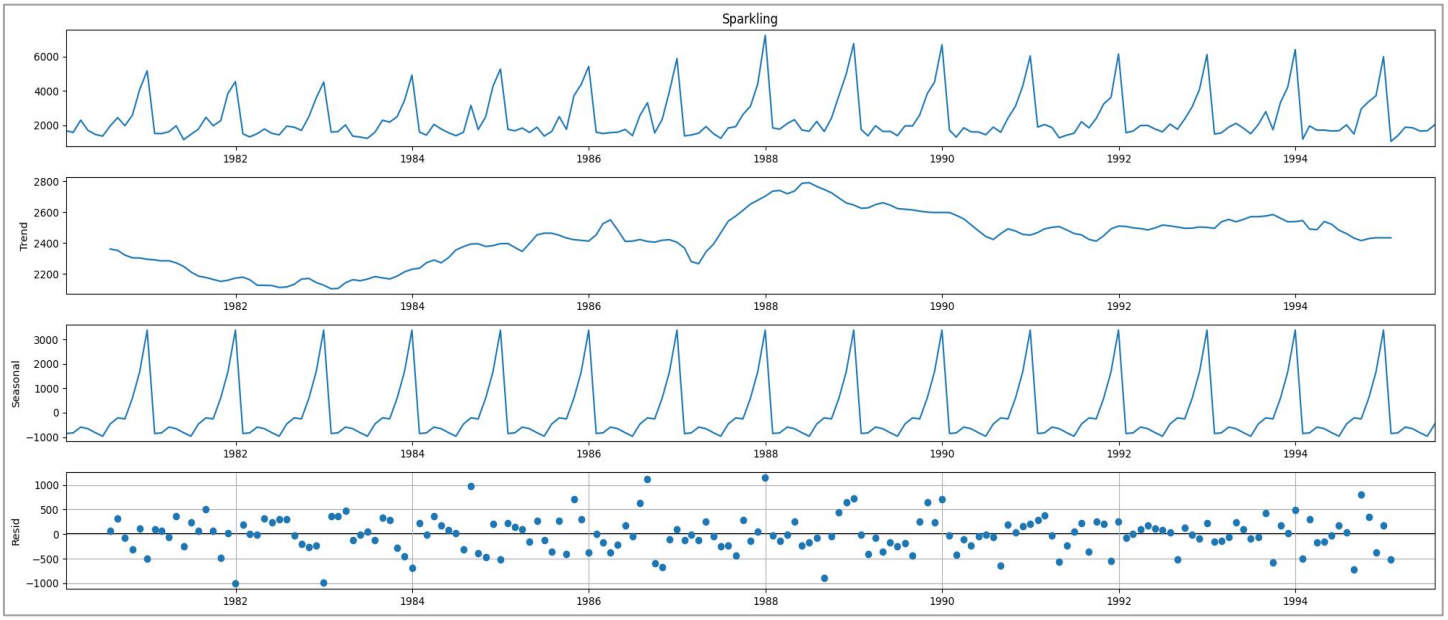

Fig 1.9 Decomposed Time Series– Additive

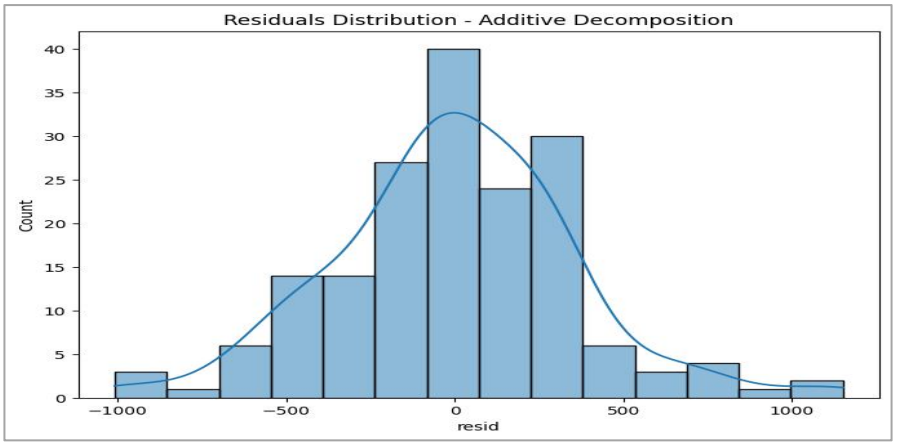

Fig 1.10 Residuals Histogram– Additive Decomposition

▪ Test for Normality

We will use **the Shapiro Wilk Test** for Normality. Let's define the Null & alternate hypothesis: –

Ho: The residuals are normally distributed

Ha: The residuals are not normally distributed

p—value of the Shapiro–Wilk Test on the residuals **= 0.03**

Since the p–value < 0.05 – We Reject the null hypothesis.

Hence Residuals are **not normally distributed at 95% confidence level**. The time series is **not an additive** time series.

- **Multiplicative Decomposition:**

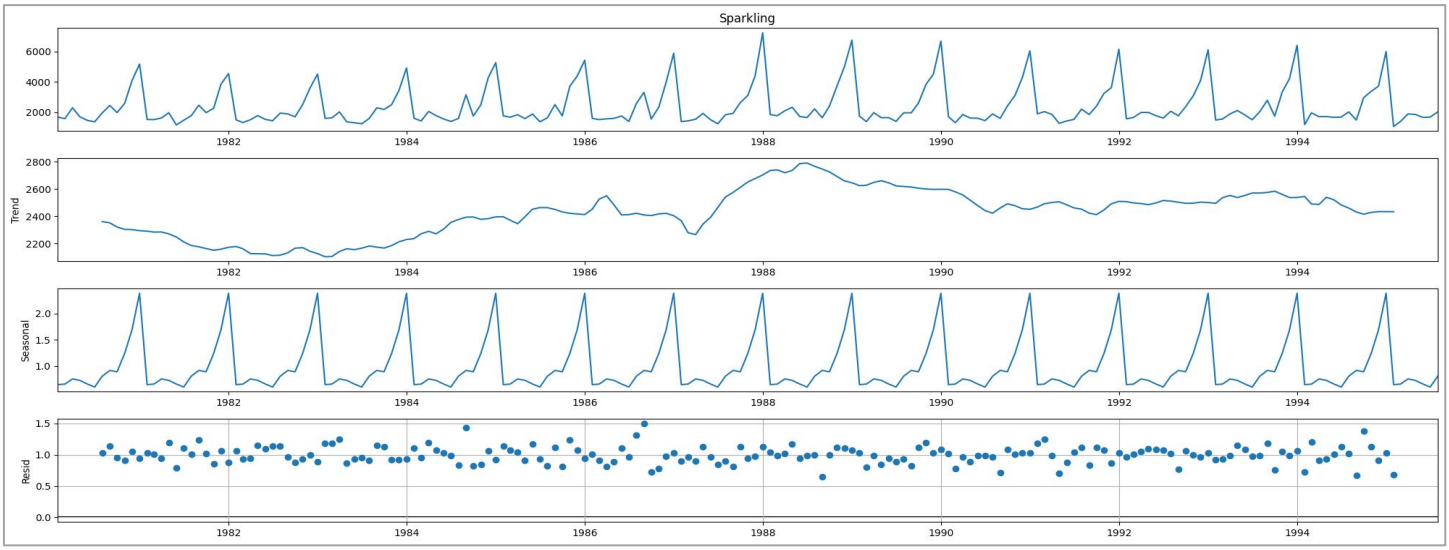

Fig 1.11 Decomposed Time Series– Multiplicative

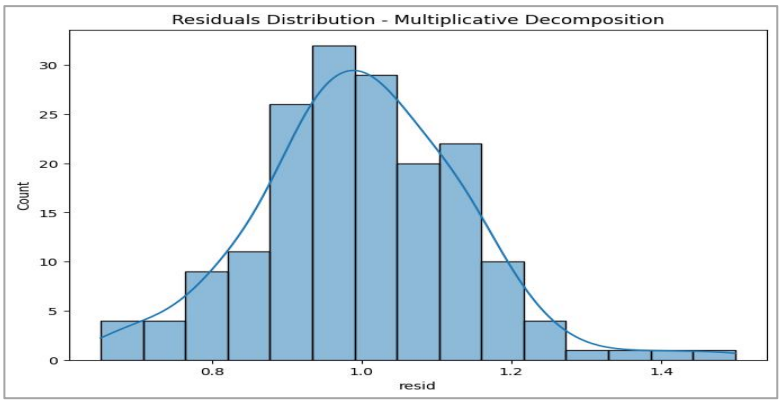

Fig 1.12 Residuals Histogram— Multiplicative Decomposition

- **Test for Normality**

We will use **the Shapiro Wilk Test** for Normality. Let's define the Null & alternate hypothesis: –

Ho: The residuals are normally distributed

Ha: The residuals are not normally distributed

p—value of the Shapiro–Wilk Test on the residuals = **0.08**

Since the p–value > 0.05: we **fail to reject the Null hypothesis**

Residuals are **normally distributed at 95% confidence level.** The time series is a **multiplicative** time series.

- **Time series components for Multiplicative:**

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

```
Seasonality
 Year_Month
1980-01-31     0.649843
1980-02-29     0.659214
1980-03-31     0.757440
1980-04-30     0.730351
1980-05-31     0.660609
1980-06-30     0.603468
1980-07-31     0.809164
1980-08-31     0.918822
1980-09-30     0.894367
1980-10-31     1.241789
1980-11-30     1.690158
1980-12-31     2.384776
Name: seasonal, dtype: float64
```

```
Residual
 Year_Month
1980-01-31          NaN
1980-02-29          NaN
1980-03-31          NaN
1980-04-30          NaN
1980-05-31          NaN
1980-06-30          NaN
1980-07-31     1.029230
1980-08-31     1.135407
1980-09-30     0.955954
1980-10-31     0.907513
1980-11-30     1.050423
1980-12-31     0.946770
Name: resid, dtype: float64
```

Table 1.7 Decomposed Time Series Components

Based on the decomposed data provided for both additive and multiplicative decomposition, we can see that the **Time Series is Multiplicative**

- The **Trend** appears to be increasing over time with some fluctuations, indicating a **positive growth pattern.**
- The **seasonality** shows both positive and negative values, suggesting regular **cycles or seasonal effects.**
- The **residual** component includes random fluctuations and unexplained variance in the time series.

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

- The data was split into a train and test set.
- The splitting was done chronologically, with data from the year **1991** forming the **test set.**
- The **train** set contains **132 records.**
- The **test** set contains **55 records.**

```
Dimentions of Original Dataset: (187, 1)
Dimentions of Training data: (132, 1)
Dimentions of Training data: (55, 1)
```

Table 1.8 Dimensions of Original, Train & Test Data

- **Training data sample**

First few rows of Train

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

Last few rows of Train

Year_Month	Sparkling
1990-08-31	1605
1990-09-30	2424
1990-10-31	3116
1990-11-30	4286
1990-12-31	6047

Table 1.9 Sample of Training Data

- **Test data sample**

First few rows of Test

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

Last few rows of Test

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

Table 1.10 Sample of Test Data

- **Train Test Split Plot**

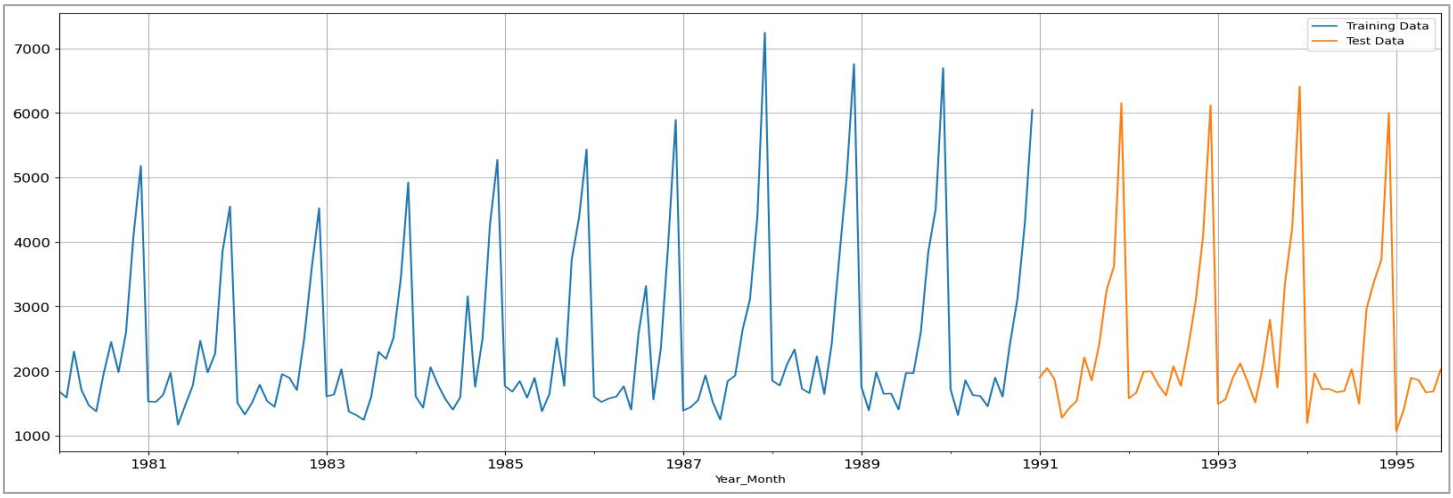

Fig 1.13 Train & Test Split Time Series

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

Building different models and comparing the accuracy metrics.

- **Linear Regression Model**

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

We generated the **numerical time instance order** for both the training and test set. Sample of the Train & Test data

First few rows of Training Data

Year_Month	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

First few rows of Test Data

Year_Month	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

Table 1.11 Sample of LinearRegression Train & Test Data

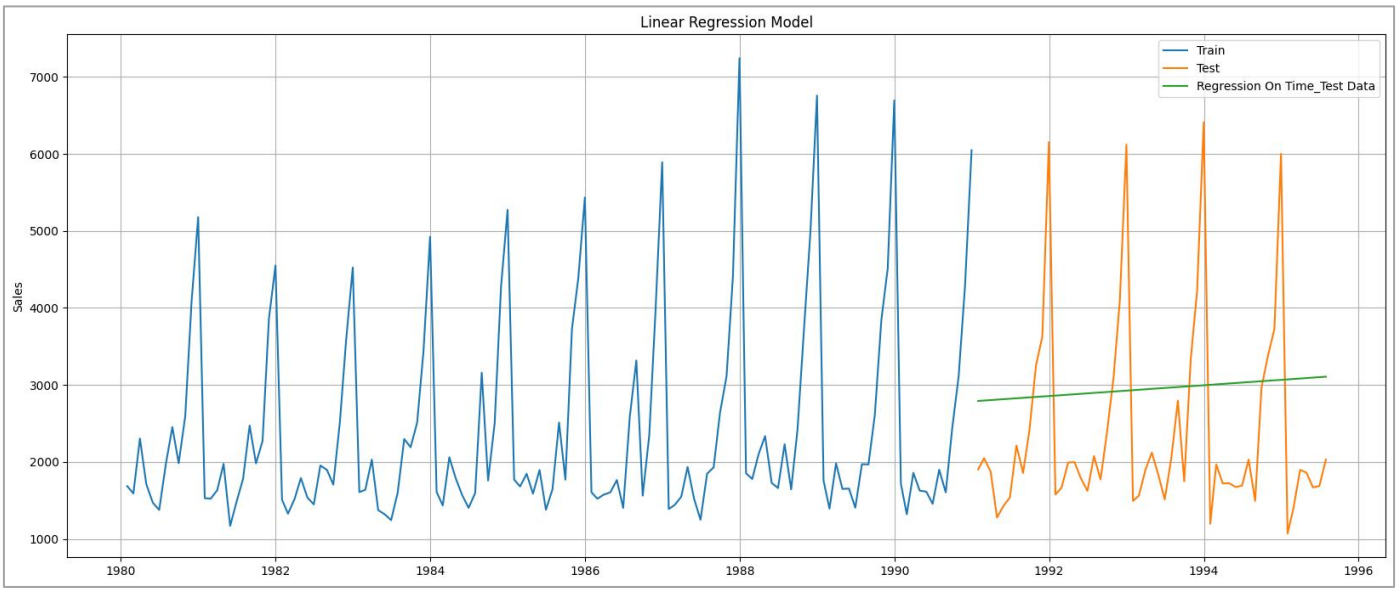

Fig 1.14 Time Series Plot: Linear Regression

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Linear Regression	1279.32	1389.14	59.35

Table 1.12 Model Performance Summary – Linear Regression

- Linear regression **captures the trend** but **not the seasonality.**
- **Test RMSE is 1389.14**, **MAPE is 59.35** for Linear Regression, indicating **difficulty** in handling seasonality.

➢ Naïve Forecast Model

The Naive model **predicts tomorrow's value** based on **today's observation,** and the **day after tomorrow's** prediction is also the **same as today's value.**

Samples of Train & test data after we trained on Naive model:

Year_Month	
1980-01-31	6047
1980-02-29	6047
1980-03-31	6047
1980-04-30	6047
1980-05-31	6047
Name: naive, dtype: int64	

Year_Month	
1991-01-31	6047
1991-02-28	6047
1991-03-31	6047
1991-04-30	6047
1991-05-31	6047
Name: naive, dtype: int64	

Table 1.13 Samples of Train & Test Data — Naive Model

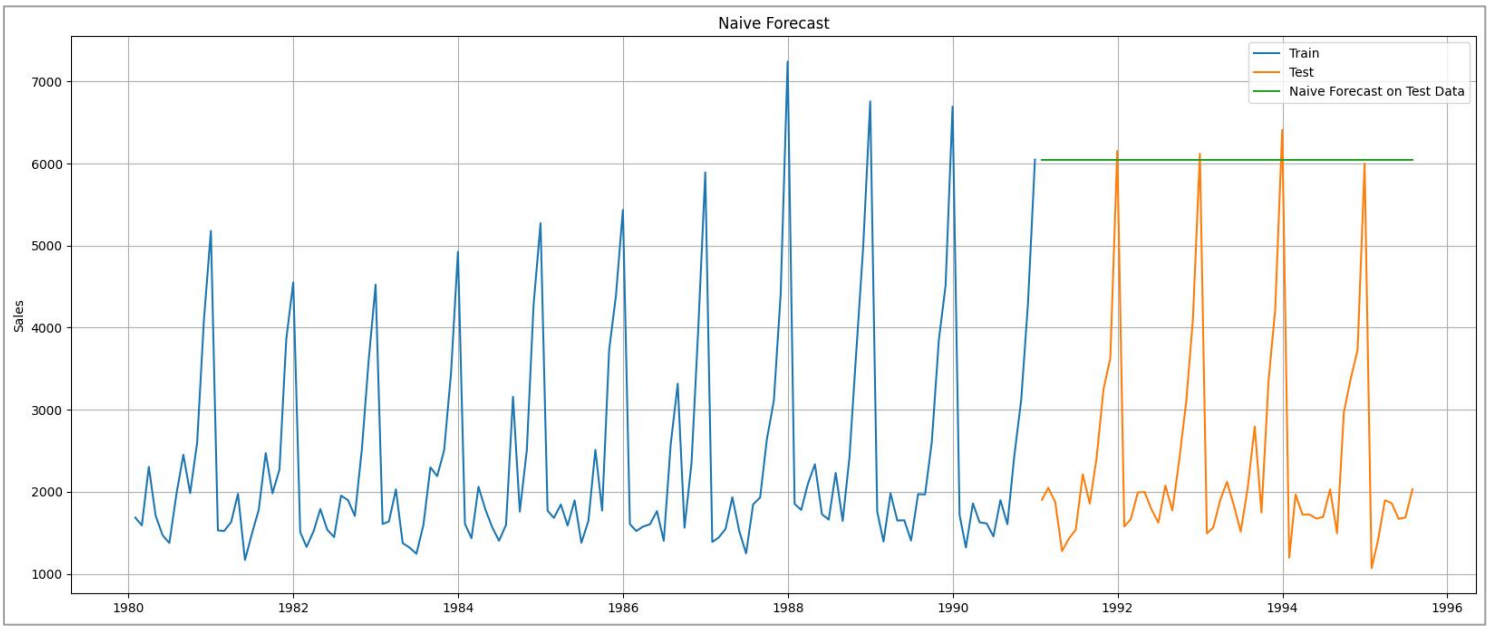

Fig 1.15 Time Series Plot: Naïve

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Naïve Forecast	3867.70	3864.28	201.33

Table 1.14 Model Performance Summary — Naïve

- Naive approach **ignores trend and seasonality** as it forecasts the last observed value
- **Test RMSE is 3864.28**, **MAPE is 201.33** for Naive forecast, **showing significant errors** due to the lack of trend and seasonality capture.

➢ Simple Average Model:

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

Samples of Train & test data for Simple Average:

Year_Month	Sparkling	simpleAvg
1980-01-31	1686	2403.780303
1980-02-29	1591	2403.780303
1980-03-31	2304	2403.780303
1980-04-30	1712	2403.780303
1980-05-31	1471	2403.780303

Year_Month	Sparkling	mean_forecast
1991-01-31	1902	2403.780303
1991-02-28	2049	2403.780303
1991-03-31	1874	2403.780303
1991-04-30	1279	2403.780303
1991-05-31	1432	2403.780303

Table 1.15 Samples of Train & Test Data for Simple Average

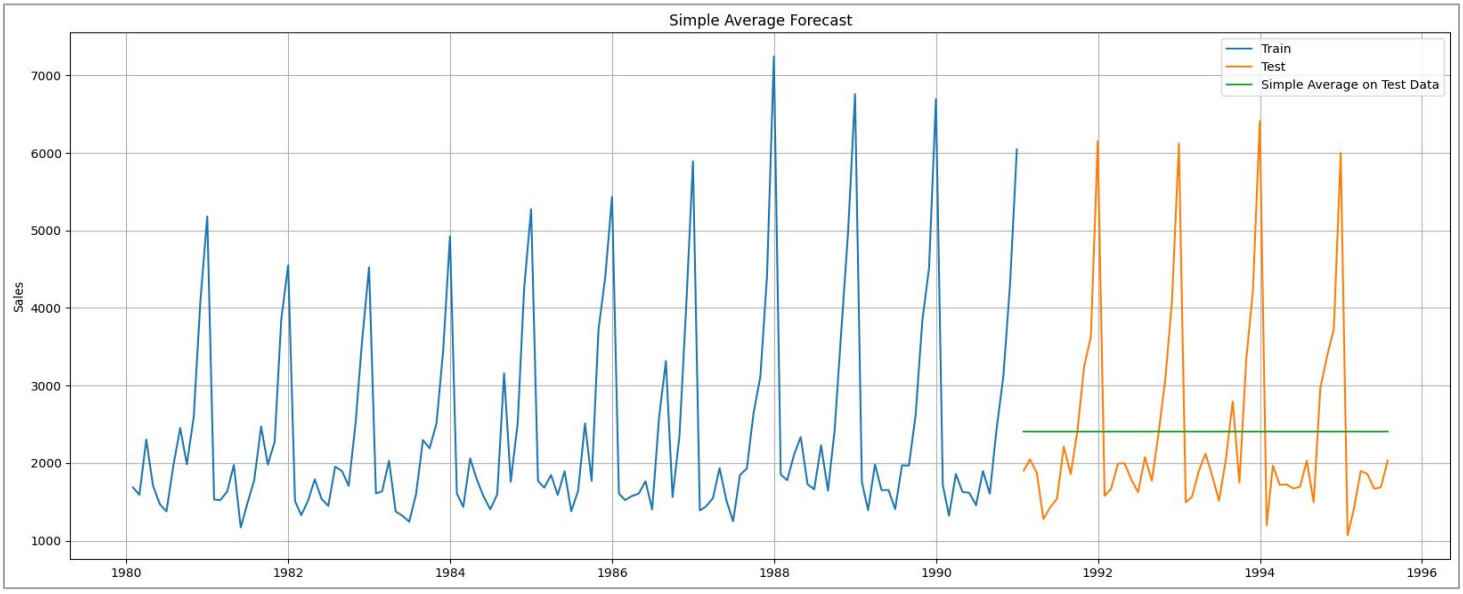

Fig 1.16 Time Series Plot: Simple Average

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Simple Average	1298.48	1275.08	39.16

Table 1.16 Model Performance Summary – Simple Average

- Simple Average Model **forecasts the mean** of the training data. It **ignores** both the **trend and seasonality.**
- **Test RMSE is 1275.08, MAPE is 39.16**. Errors are significant due to the lack of trend and seasonality capture.
- **Performs better than linear regression** in this case as test data has a constant trend different from the training data's underlying trend.

➢ 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) over here.

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

Moving Average Sample on Training data

Year_Month	Sparkling	Trailing_2	Trailing_4	Trailing_6	Trailing_9
1980-01-31	1686	NaN	NaN	NaN	NaN
1980-02-29	1591	1638.5	NaN	NaN	NaN
1980-03-31	2304	1947.5	NaN	NaN	NaN
1980-04-30	1712	2008.0	1823.25	NaN	NaN
1980-05-31	1471	1591.5	1769.50	NaN	NaN
1980-06-30	1377	1424.0	1716.00	1690.17	NaN
1980-07-31	1966	1671.5	1631.50	1736.83	NaN
1980-08-31	2453	2209.5	1816.75	1880.50	NaN
1980-09-30	1984	2218.5	1945.00	1827.17	1838.22
1980-10-31	2596	2290.0	2249.75	1974.50	1939.33

Table 1.17 Moving Average Sample on Training data

Moving Average Sample plot on Whole data

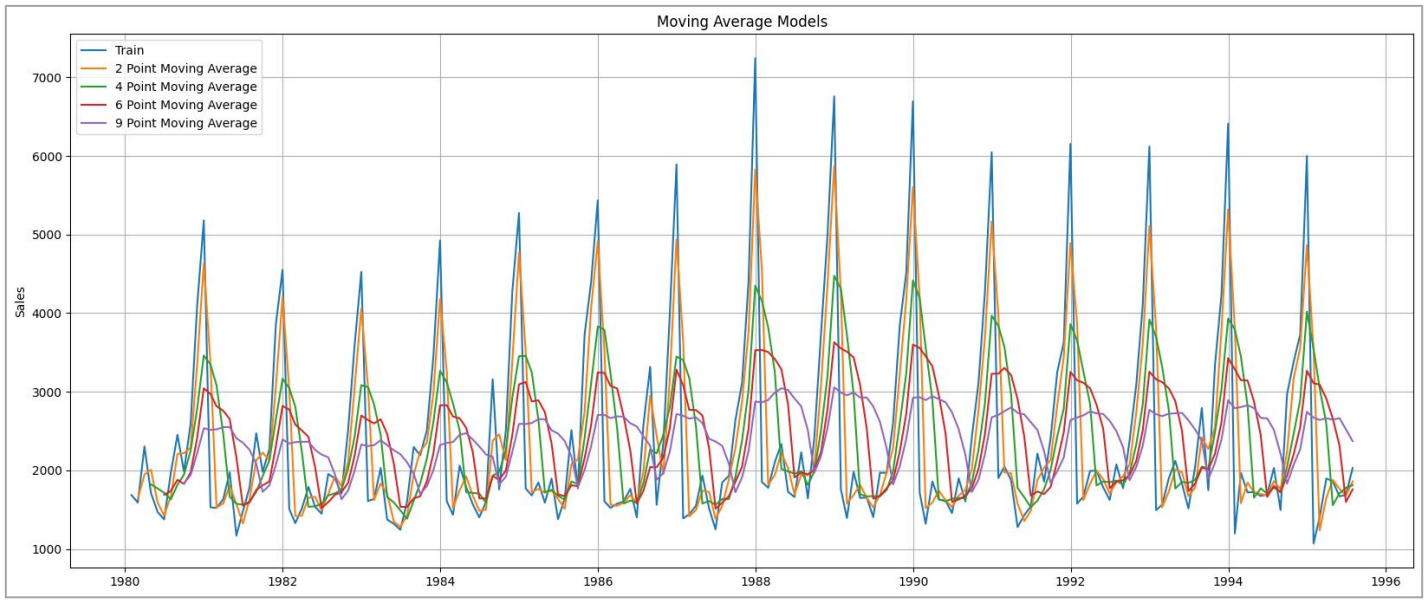

Fig 1.17 Time Series Plot: Moving Average on Whole data

Moving Average Sample plot on Test data

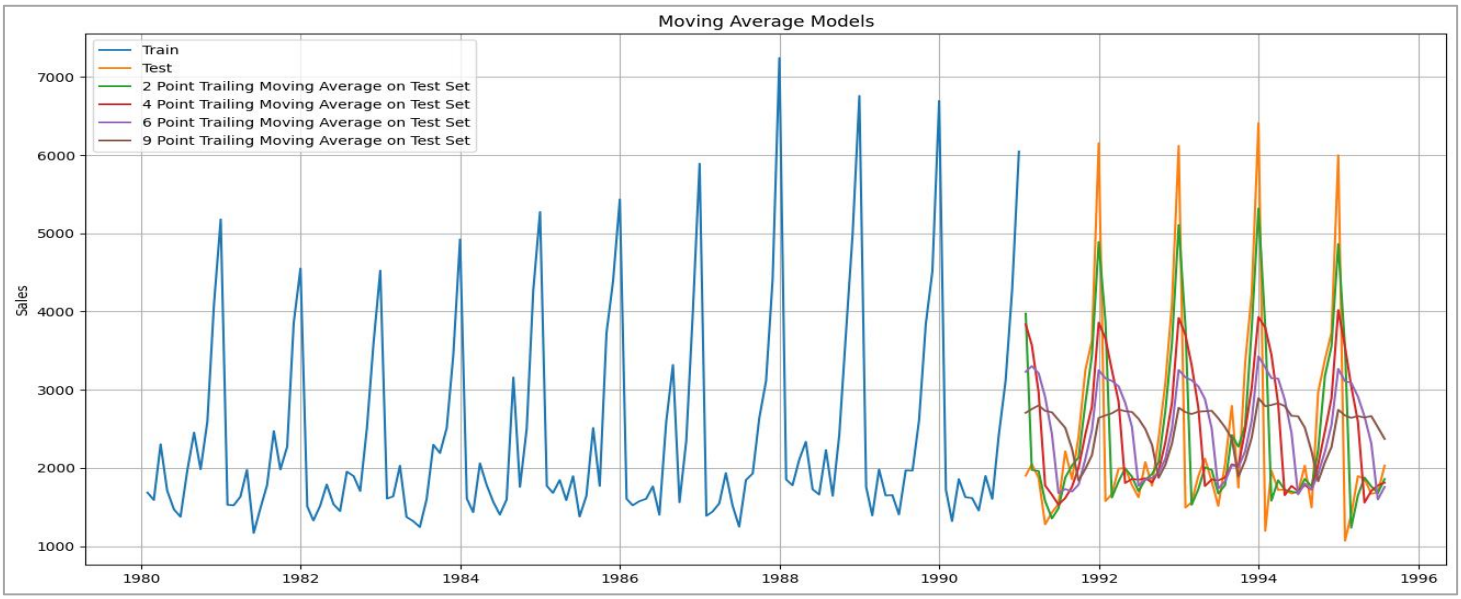

Fig 1.18 Time Series Plot: Moving Average

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
2 Point Moving Average	706.18	813.40	24.71
4 Point Moving Average	1125.87	1156.59	41.08
6 Point Moving Average	1275.45	1283.93	48.4
9 Point Moving Average	1372.84	1346.28	50.07

Table 1.18 Model Performance Summary – Moving Averages

The **2-point Trailing Moving Average** model is selected as it has the **least errors (RMSE & MAPE)** among all the moving average models evaluated.

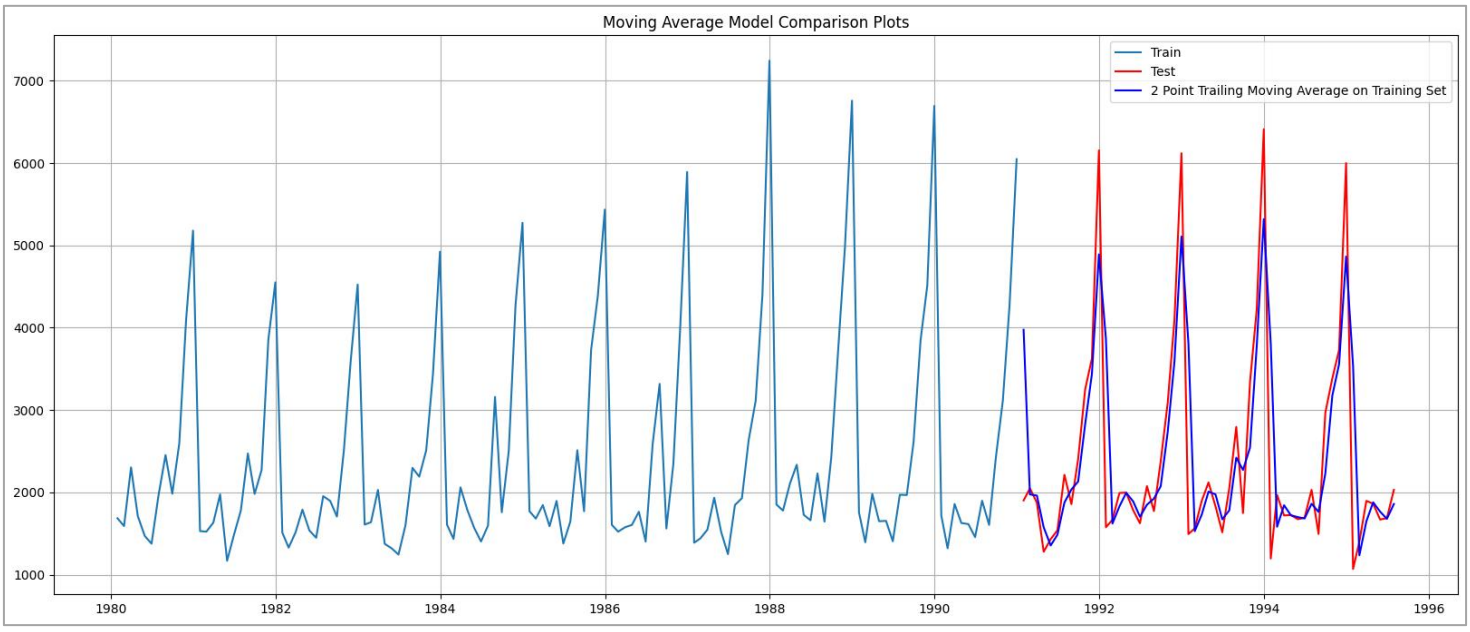

Fig 1.19 Time Series Plot: 2-Point Moving Average

For the **moving average models,** here are the insights based on their RMSE and MAPE values:

- **2–Point Moving Average:** RMSE = **813.40**, MAPE = **24.71.** It performs **relatively well** in **capturing** the **trend** and **seasonality** but still has room for improvement as **struggles** to capture the **sharp fluctuations** in the sales pattern over time
- **4–Point Moving Average**: RMSE = 1156.59, MAPE = 41.08. It performs slightly worse than the 2–Point model, but it still shows better accuracy than simpler models like the Simple Average.
- **6–Point Moving Average**: RMSE = 1283.93, MAPE = 48.4. It performs reasonably well but is not as accurate as the 2–Point or 4–Point models.
- **9–Point Moving Average**: RMSE = 1346.28, MAPE = 50.07. It shows reasonable performance but is not as good as the 2–Point model.
- Overall, the **2–Point Moving Average model** stands out as the best–performing model with the **lowest RMSE and MAPE** values. However, there is still room for improvement in all the models to better capture the trend `seasonality and reduce errors in the forecasts.

➢ Simple Exponential Smoothing (SES)

This method is suitable for forecasting data with **no clear trend or seasonal pattern.** It gives more weight to recent observations, which means that recent data points have a stronger influence on the forecast than older ones. This approach allows the model to capture short–term trends and adapt quickly to changes in the data.

Parameter **Alpha (α)** is called the smoothing constant and its value lies between **0 and 1.** Since the model uses only one smoothing constant, it is called Single Exponential Smoothing

▪ SES: Auto Fill Method

The autofit model finds the most optimal parameters according to python while fitting on the train data.

Simple Exponential Smoothing optimal parameters: –

Smoothing Level (alpha) = 0.07

Initial Level = 1763.93.

```
{'smoothing_level': 0.07028781460389563,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 1763.9269926897732,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Table 1.19 Autofill Simple Exponential Smoothing Optimal Parameters

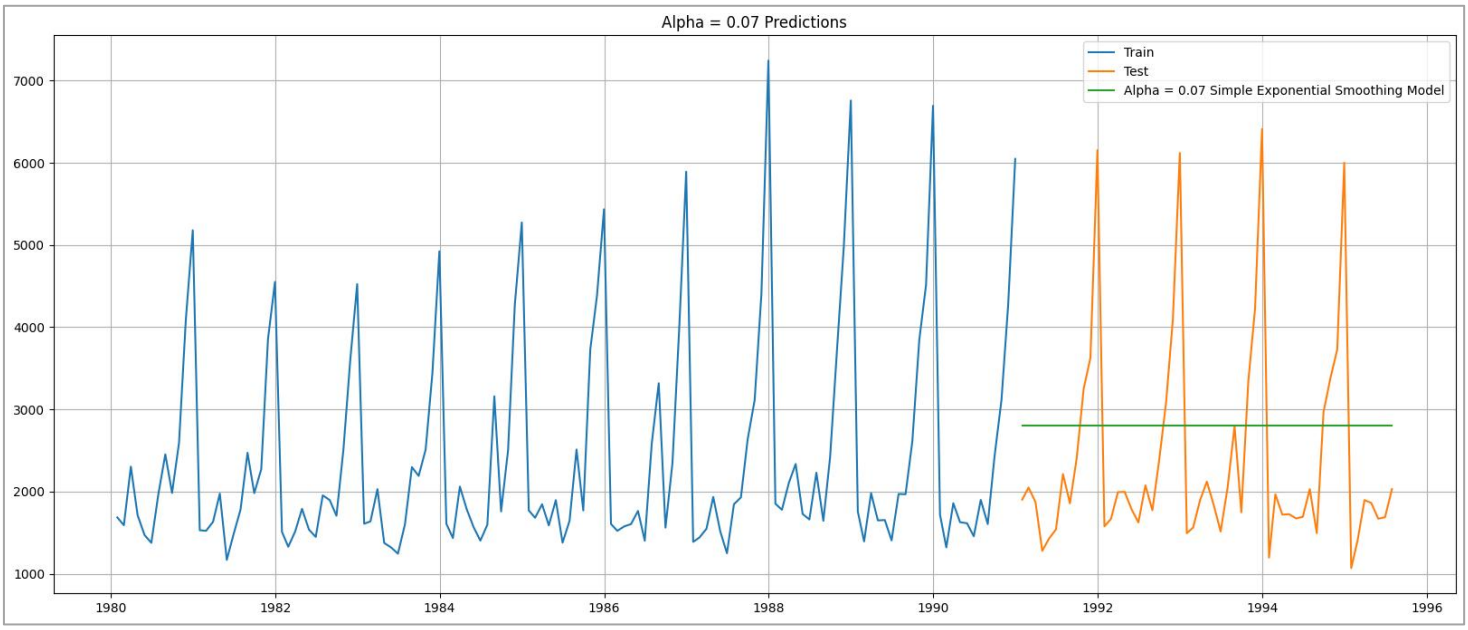

Fig 1.20 Time Series Plot: Simple Exponential Smoothing Alpha = 0.07

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Simple Exponential Smoothing Alpha = 0.07	1322.90	1338.00	53.88

Table 1.20 Model Performance Summary – Simple Exponential Smoothing Alpha = 0.07

- **The Autofill Simple exponential smoothing** model provides one-step-ahead forecast. It **ignores** both the **trend and seasonality** in the data.
- Test RMSE is **1338** and MAPE is **53.88,** indicating **poor performance** in capturing underlying patterns.
- The low smoothing parameter **(0.07)** implies a heavy reliance on past averages. This makes it less accurate compared to more sophisticated methods.

- **SES: Brute Force Method**

The brute force model **tests various smoothing parameter** values to find the best ones for accurate test data forecasting. Below is the table for various parameters, **sorted** with **least Test RMSE** on top.

	Alpha	Train RMSE	Test RMSE
1	0.02	1346.258628	1278.497798
0	0.01	1397.988872	1286.648058
2	0.03	1329.877089	1292.565292
3	0.04	1324.937340	1305.283075
4	0.05	1324.401979	1316.359347

Table 1.21 Brute Force Simple Exponential Smoothing Parameters

Best **Alpha** for Simple Exponential Smoothing: **0.02**

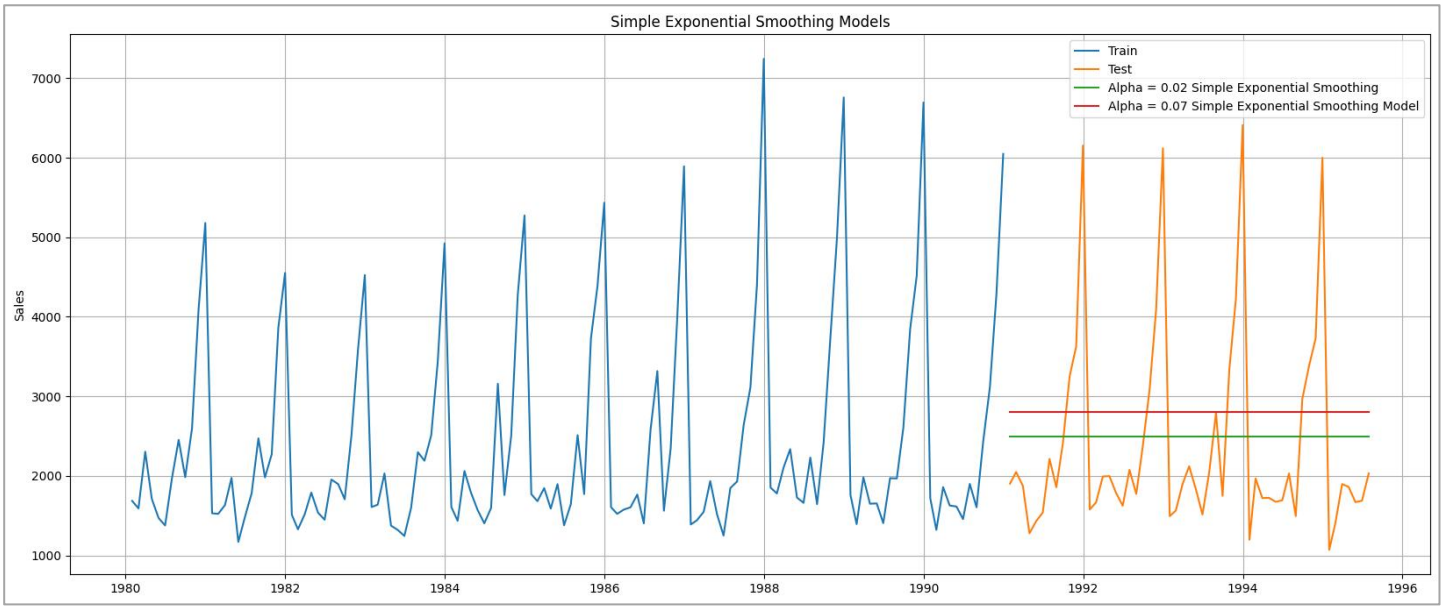

Fig 1.21 Time Series Plot: Simple Exponential Smoothing Alpha = 0.02

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Simple Exponential Smoothing Alpha = 0.02	1346.26	1278.50	42.41

Table 1.22 Model Performance Summary – Simple Exponential Smoothing Alpha = 0.02

- The **Brute Force simple exponential smoothing** model generates one-step-ahead forecasts, **overlooking both trend and seasonality** in the time series.
- The model exhibits an RMSE **of 1278.50** and **MAPE of 42.41**, signifying its **inability** to effectively capture the underlying **trend and seasonal patterns.**
- The small smoothing parameter **(0.02)** indicates that the **model relies** heavily on **past data averages** rather than recent observations, making its accuracy comparable to the simple average model.

➢ Double Exponential Smoothing (DES)

This method is applicable when data has Trend but no seasonality.

▪ DES: Auto Fill Method

The autofit model finds the most optimal parameters according to python while fitting on the train data.

Double Exponential Smoothing optimal parameters: –

Smoothing Level (Alpha) = 0.665

Smoothing Trend (beta) = 0.0001

```
Holt model Exponential Smoothing Estimated Parameters

{'smoothing_level': 0.6649999999999999, 'smoothing_trend': 0.0001, 'smoothing_seasonal': nan,
```

Table 1.23 Autofill Double Exponential Smoothing Optimal Parameters

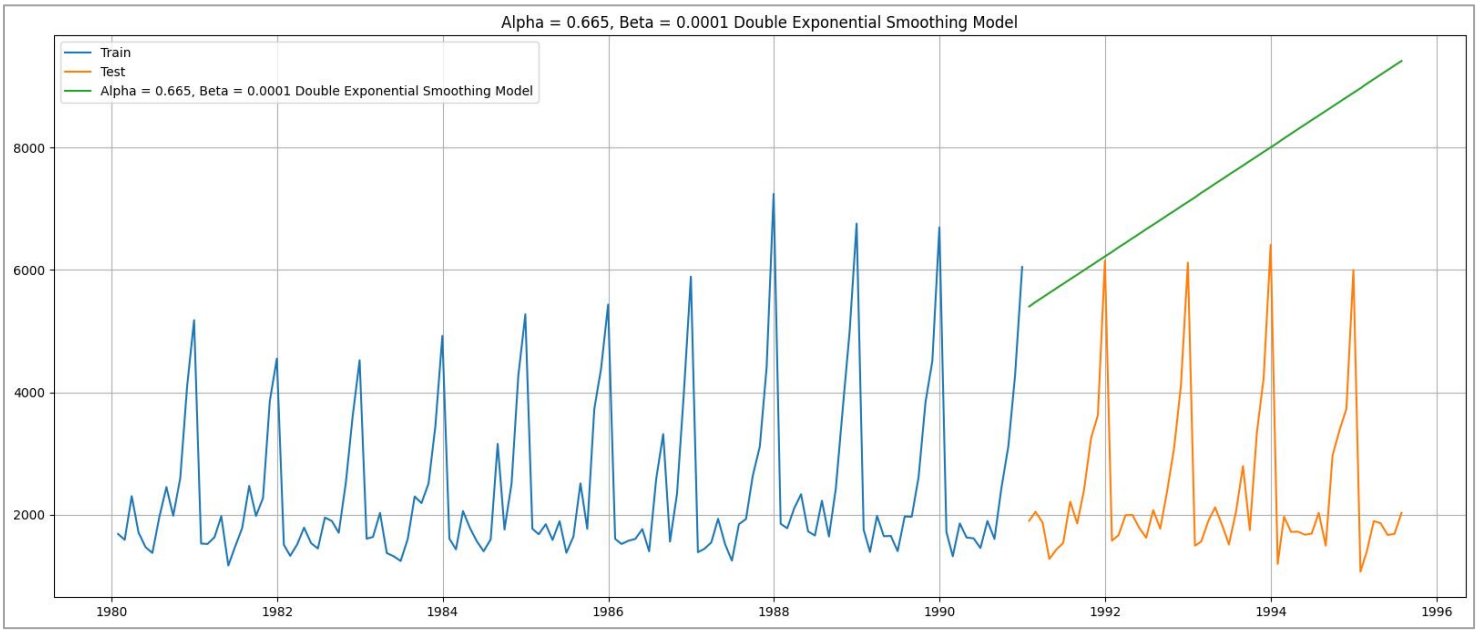

Fig 1.22 Time Series Plot: Double Exponential Smoothing Alpha = 0.665, Beta= 0.0001

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Double Exponential Smoothing Alpha = 0.665, Beta= 0.0001	1339.50	5291.88	268.91

Table 1.24 Model Performance Summary – Double Exponential Smoothing Alpha = 0.665, Beta= 0.0001

- The **Autofill double exponential smoothing** model forecasts the **trend** but **disregards** the **seasonality** of the time series.
- The **RMSE is 5291.88**, and the **MAPE is 268.91** for the Double exponential smoothing model. While the error is comparably lower in the Train set, the **erro**r is **very high on test**, indicating its **failure to capture the seasonality i**n the data.
- The level smoothing parameter being close to 1 suggests it is **highly influenced** by **recent observations**, **leading to higher forecasts due to sales spikes at the year-end.**
- Similarly, with beta close to 0, the model **captures the historical trend** of the particular level and **extrapolates** it.

- **DES: Brute Force Method**

The brute force model **tests various smoothing parameter** values to find the best ones for accurate test data forecasting. Below is the table for various parameters, **sorted** with **least Test RMSE** on top.

	Alpha	Beta	Train RMSE	Test RMSE
136	0.02	0.38	1398.025311	1275.874751
135	0.02	0.37	1398.309816	1276.128575
106	0.02	0.08	1509.840203	1276.557274
258	0.03	0.61	1415.807906	1278.165214
201	0.03	0.04	1492.652360	1279.190198

Table 1.25 Brute Force Double Exponential Smoothing Parameters

Since **alpha = 0.02** and **beta = 0.38** yield the **least test RMSE**, indicating the best fit for our test data, we select them to build our double exponential smoothing model.

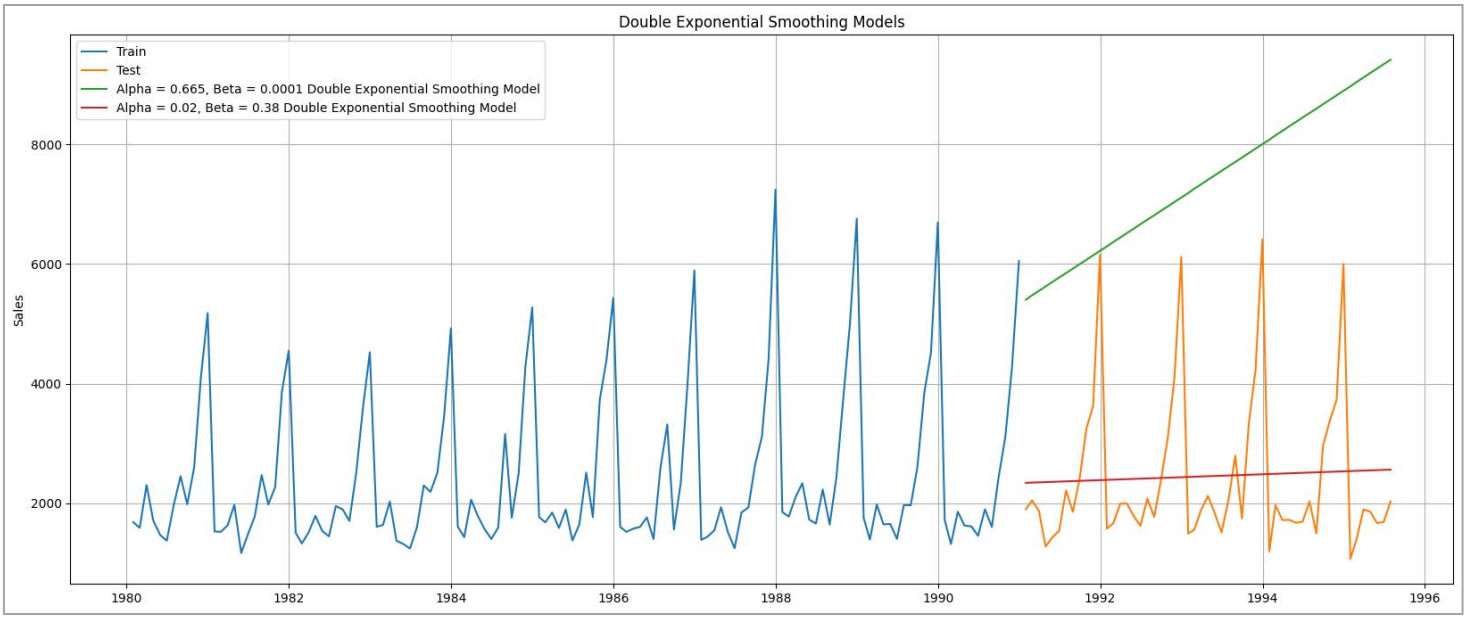

Fig 1.23 Time Series Plot: Double Exponential Smoothing Alpha = 0.02, Beta= 0.38

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Double Exponential Smoothing Alpha = 0.02, Beta= 0.38	1398.03	1275.87	40.97

Table 1.26 Model Performance Summary – Double Exponential Smoothing Alpha = 0.02, Beta= 0.38

- The **Brute Force double exponential smoothing** model effectively **captures the trend** but **overlooks the seasonality** of the time series.
- This performs **better than auto fill method** (Double Exponential Smoothing alpha= 0.665, Beta = 0.0001)

- The model **exhibits a high RMSE** of 1275.87 and **MAPE of 40.97**, indicating its **failure** to **capture the seasonality** in the data.
- The level smoothing parameter (**alpha**) is **close to 0 (0.02),** suggesting that the level forecast is primarily **based on the past data rather than recent observations.**
- The trend smoothing parameter (**beta**) is **0.38,** which lies between 0 and 1, indicating a **balanced approach** that **considers both historical data and recent observations** to forecast the trend.

➢ Triple Exponential Smoothing (TES)

The triple exponential smoothing model is suitable for time series with both **trend and seasonality**.

Since the trend variation is linear and the seasonal decomposition suggests a multiplicative time series, we use a triple exponential smoothing model with additive trend and multiplicative seasonality.

▪ TES: Auto Fill Method

The autofit model finds the most optimal parameters according to python while fitting on the train data.

Triple Exponential Smoothing optimal parameters: –

Smoothing Level (**Alpha**) = **0.111**

Smoothing Trend (**Beta**) = **0.049**

Smoothing Seasonal (**Gamma**) = **0.362**

```
Holt Winters model Exponential Smoothing Estimated Parameters

{'smoothing_level': 0.11119949831569428, 'smoothing_trend': 0.049430920023313805, 'smoothing_seasonal': 0.3620525701498937
```

Table 1.27 Autofill Triple Exponential Smoothing Parameters

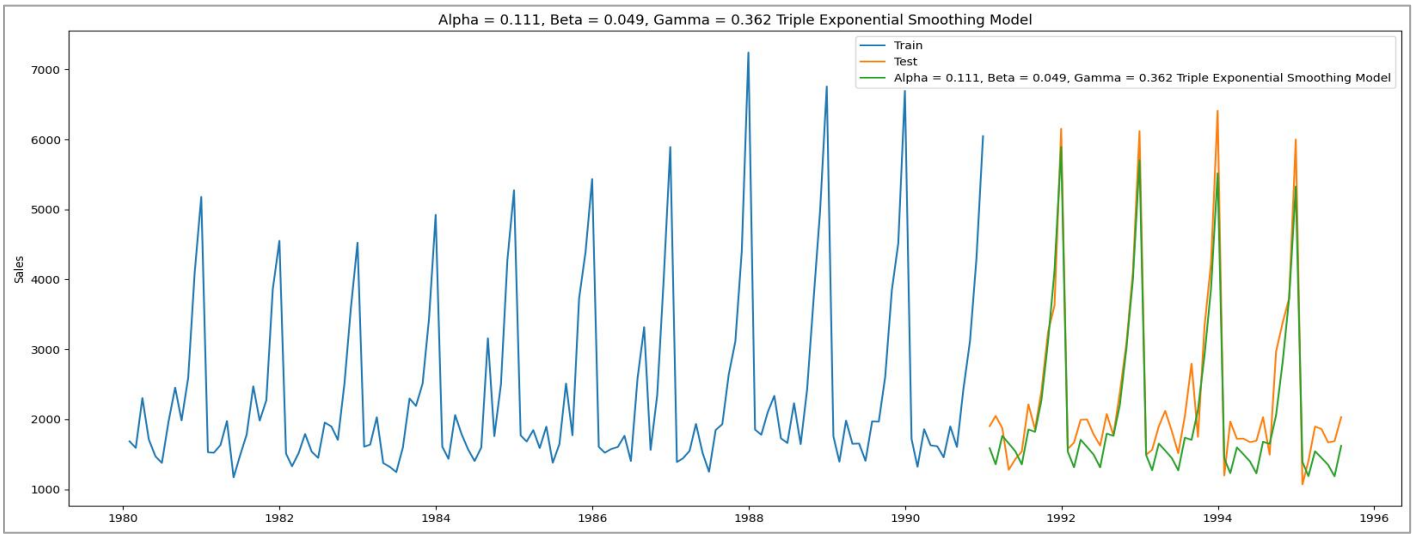

Fig 1.24 Time Series Plot: Triple Exponential Smoothing Alpha = 0.111, Beta= 0.049, Gamma= 0.362

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Triple Exponential Smoothing Alpha = 0.111, Beta= 0.049, Gamma= 0.362	355.77	403.71	48.37

Table 1.28 Model Performance Summary – Triple Exponential Smoothing Alpha = 0.111, Beta= 0.049, Gamma= 0.362

- The **Autofill triple Exponential Smoothing** model **captures** both the **trend and seasonality** of the time series.
- The test **RMSE** is **403.71,** and the **MAPE is 48.37**. This model has the lowest error among all the evaluated models.
- The values of **alpha** and **beta**, being **close to 0**, indicate that the **forecast relies** heavily on **historical data** to build the model.
- On the other hand, the **gamma** value has a **moderate value**, indicating that it considers more than just the recent past data but less than the entire historical data.

- **TES: Brute Force Method**

The brute force model **tests various smoothing parameter** values to find the best ones for accurate test data forecasting. Below is the table for various parameters, **sorted** with **least Test RMSE** on top.

	Alpha	Beta	Gamma	Train RMSE	Test RMSE
41	0.01	0.04	0.25	470.837600	302.728640
40	0.01	0.04	0.22	481.966174	303.298599
42	0.01	0.04	0.28	461.026879	303.899551
1163	0.04	0.07	0.25	366.400702	305.639893
1164	0.04	0.07	0.28	365.138125	305.990053

Table 1.29 Brute Force Triple Exponential Smoothing Parameters

Since **alpha = 0.01, beta = 0.04, gamma = 0.25** yield the **least test RMSE**, indicating the best fit for our test data, we select them to build our Triple Exponential Smoothing model.

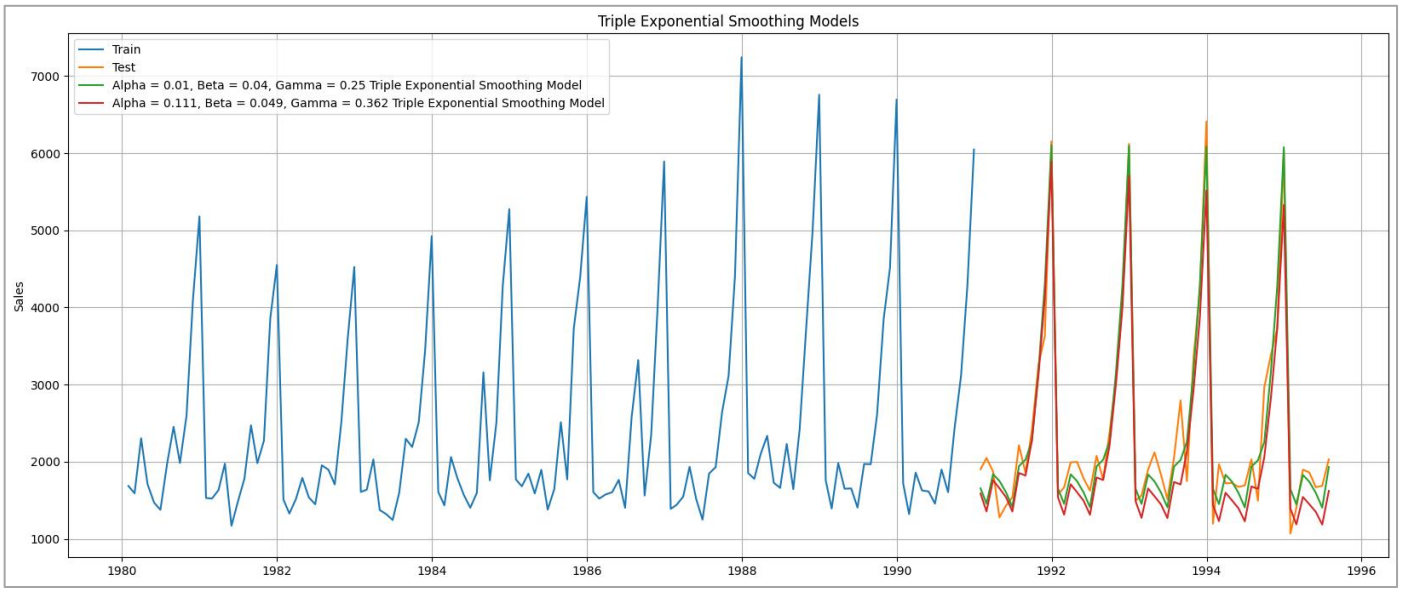

Fig 1.25 Time Series Plot: Triple Exponential Smoothing Alpha = 0.01, Beta= 0.04, Gamma= 0.25

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Triple Exponential Smoothing Alpha = 0.01, Beta= 0.04, Gamma= 0.25	470.84	302.73	49.86

Table 1.30 Model Performance Summary – Triple Exponential Smoothing Alpha = 0.01, Beta= 0.04, Gamma= 0.25

- The **Brute Force Triple Exponential smoothing** model **captures** both the **trend and seasonality** of the time series effectively.
- With **RMSE of 302.73** and **MAPE** of **49.86,** this model exhibits the **best accuracy among all the evaluated models so far.**
- The values of **alpha** and **beta** being **close to 0** imply that the model heavily **relies** on **historical data** to make forecasts.
- On the other hand, the **moderate value** of **gamma** indicates that it strikes a **balance** between **recent past** data and the **entire historical past** to **capture seasonality**.

➢ **Plotting the prediction of all the Models built so far**

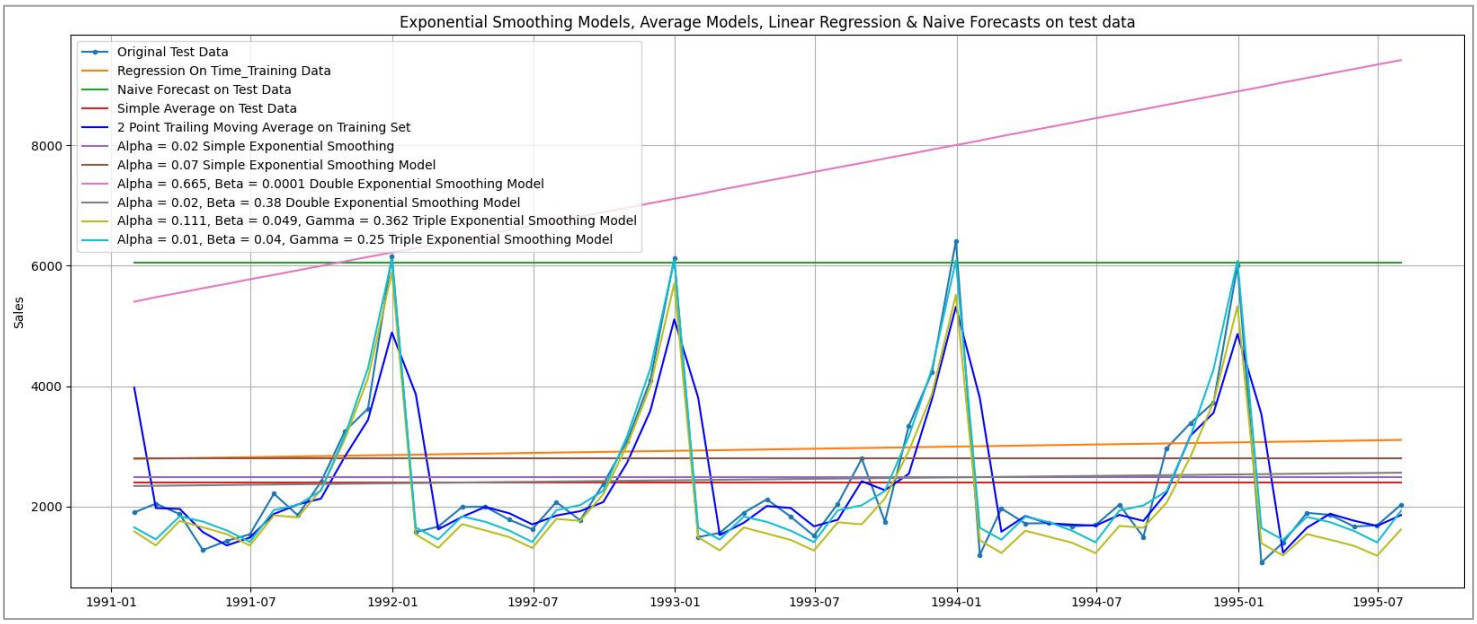

Fig 1.26 Time Series Plot: Model Comparisions

- The **triple exponential smoothing** models **performed best** followed by the **2–point moving average** model for **capturing** both **trend** and **seasonality** in the time series.
- The **Brute Force triple exponential smoothing** models show the **best accuracy** among all models evaluated, with the lowest Root Mean Square Error **(RMSE) of 302.73** and Mean Absolute Percentage Error (**MAPE**) of **49.86**.
- The **moving average** model, although effective to some extent, is **not well–suited** for capturing significant changes in the time series due to its averaging nature.
- The rest of the other models are also **not suitable for prediction** as they **do not capture** both the **Trend & Seasonality** required for the time series.

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.

➢ Stationarity of the whole Time Series data

The **Augmented Dickey–Fuller (ADF) Test** is used to check the stationarity of a time series. The test formulates the following null and alternative hypotheses:

Null Hypothesis (Ho): The time series is non-stationary.

Alternative Hypothesis (Ha): The time series is stationary.

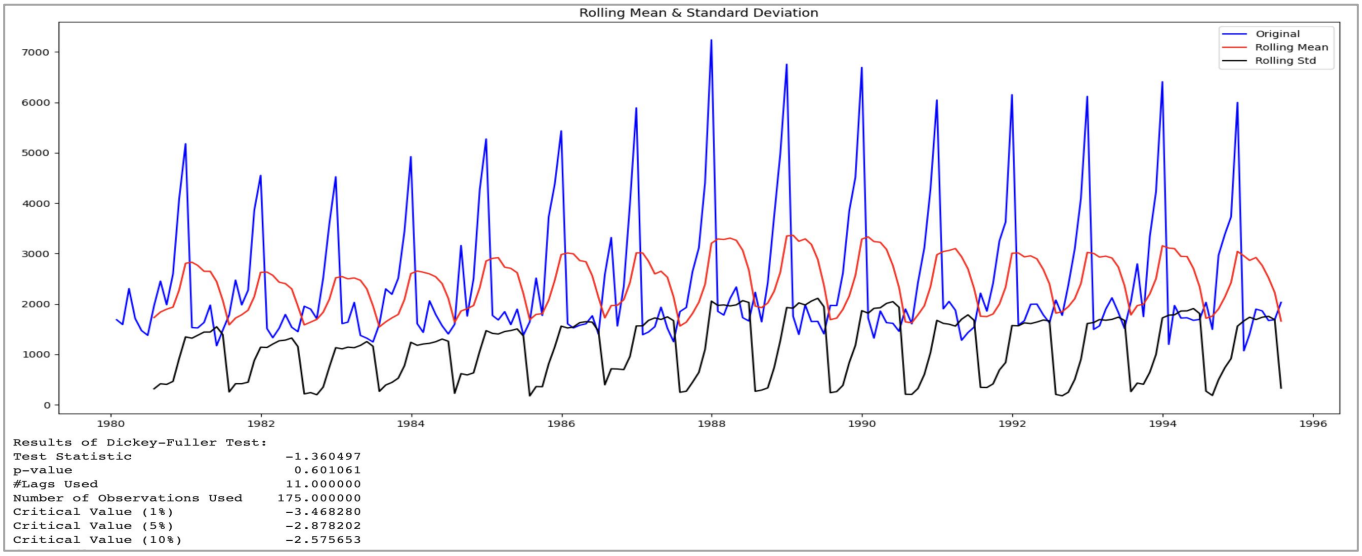

Fig 1.27 Stationarity of Whole Data Using AD Fuller Test

- We see that at **5% significant level** the Time Series is **non-stationary.**
- Let us take a **difference of order 1** and check whether the Time Series is stationary or not

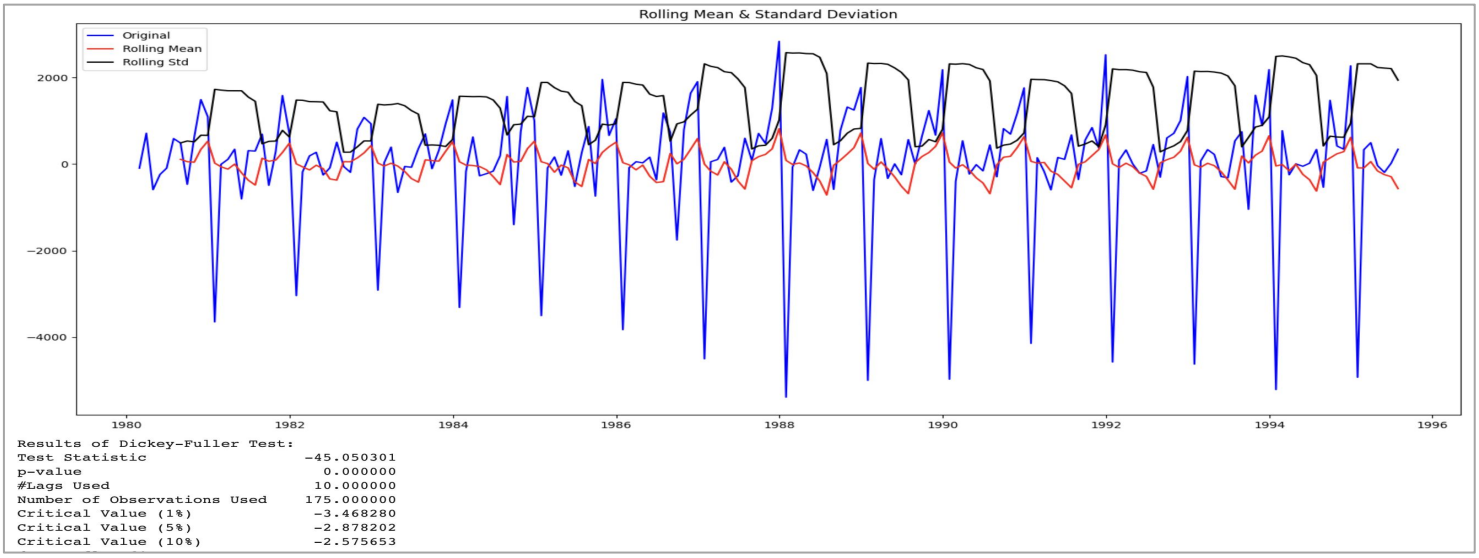

Fig 1.28 Stationarity of Whole Data Using AD Fuller Test at Differencing of Order 1

- At **difference of order 1,** the series have become **stationary at ﷺ = 0.05.**

➢ Stationarity of the Training Data Time Series

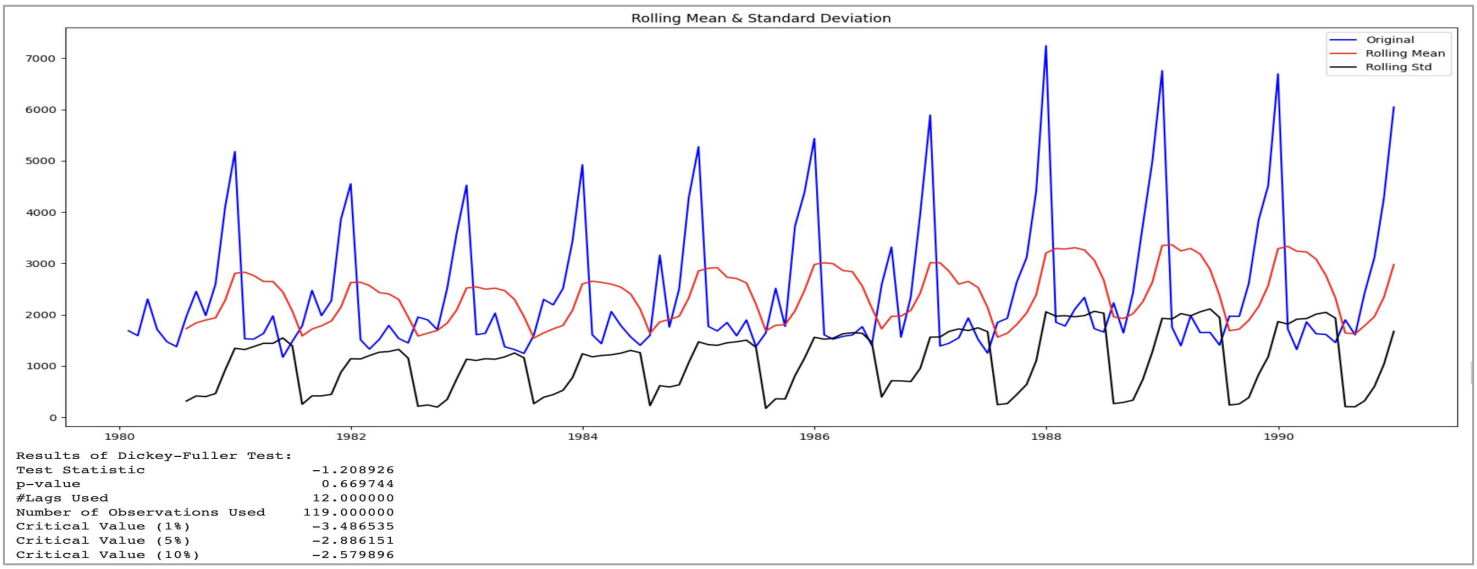

Fig 1.29 Stationarity of Training Data Using AD Fuller Test

- At **5% significant level** the Time Series is **non-stationary.**
- Taking First order difference using diff function

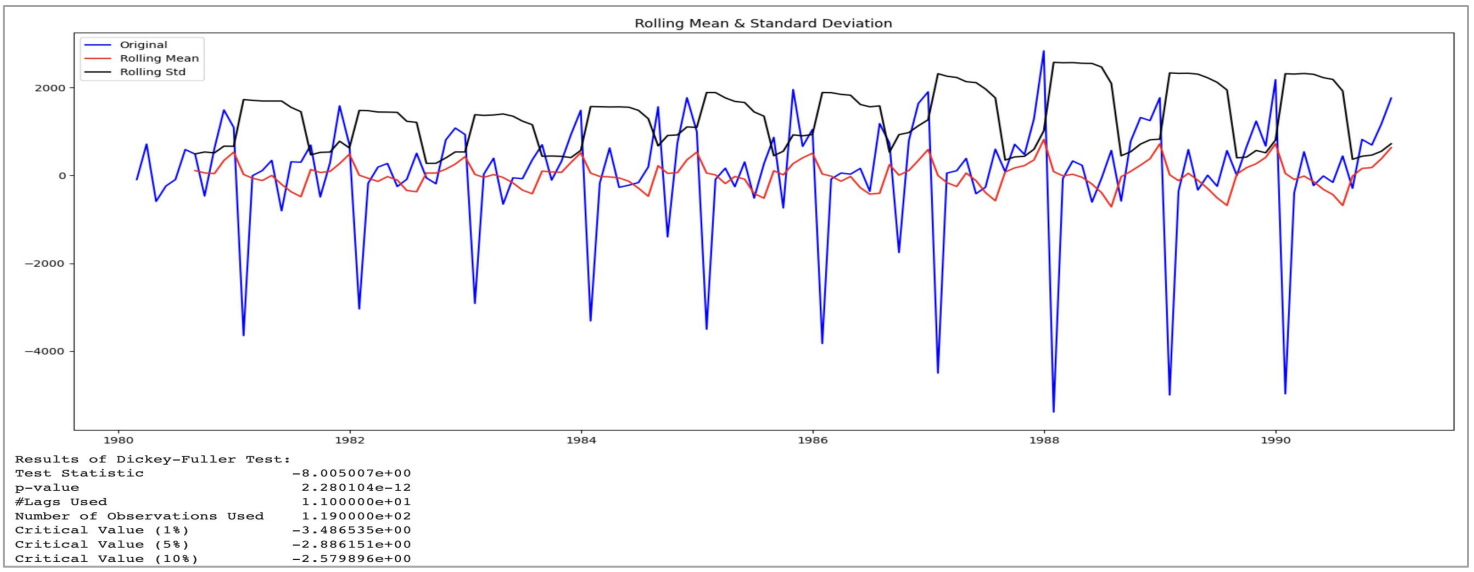

Fig 1.30 Stationarity of Training Data Using AD Fuller Test at Differencing of Order 1

- We see that at **difference of order 1**, the series have become **stationary at ** = 0.05.**

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.

➢ **Autocorrelation Plots**

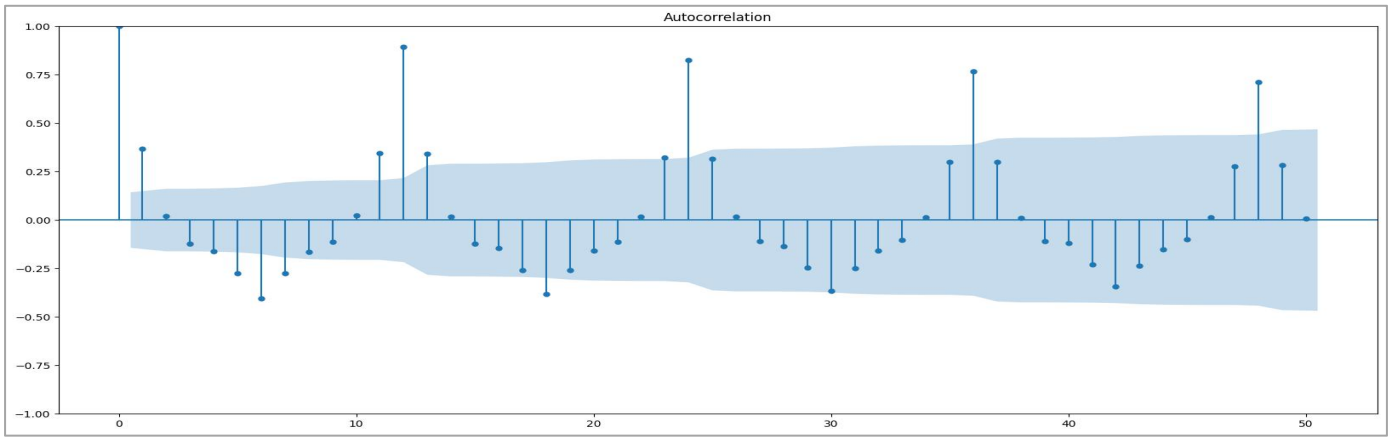

Fig 1.31 Autocorrelation Plot

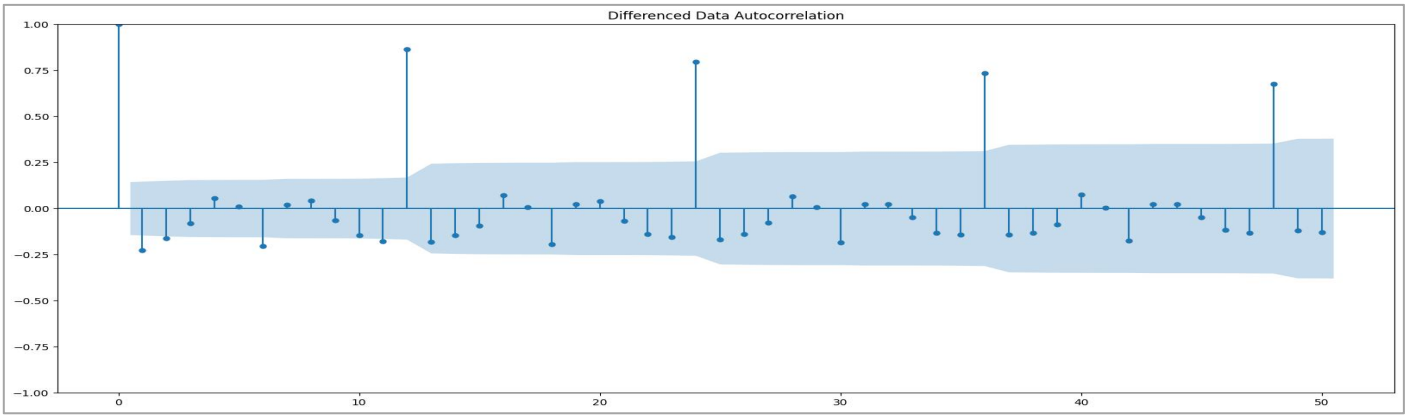

Fig 1.32 Differenced Autocorrelation Plot

- From the above plots, we can say that there is **seasonality** in the data. This would be more useful when building SARIMA model

➢ Automated version of an ARIMA model

ARIMA: – Auto Regressive Integrated Moving Average is a way of modeling time series data for **forecasting or predicting future data points**. Improving AR Models by making Time Series stationary through **Moving Average Forecasts**

ARIMA models consist of 3 components: –

AR model: The data is modelled based on past observations.

Integrated component: Whether the data needs to be differenced/transformed.

MA model: Previous forecast errors are incorporated into the model.

The best parameters are selected in accordance with the lowest Akaike Information Criteria (AIC).

- **ARIMA Model building to estimate best 'p', 'd', 'q' parameters (Lowest AIC Approach)**

	param	AIC
8	(2, 1, 2)	2213.509213
7	(2, 1, 1)	2233.777626
2	(0, 1, 2)	2234.408323
5	(1, 1, 2)	2234.527200
4	(1, 1, 1)	2235.755095
6	(2, 1, 0)	2260.365744
1	(0, 1, 1)	2263.060016
3	(1, 1, 0)	2266.608539
0	(0, 1, 0)	2267.663036

Table 1.31 ARIMA AIC Parameters

- For p = 2, d = 1, q = 2 , we obtained the **lowest AIC** value, we used it to build our ARIMA model.

ARIMA Results

SARIMAX Results

Dep. Variable:	Sparkling	No. Observations:	132
Model:	ARIMA(2, 1, 2)	Log Likelihood	-1101.755
Date:	Fri, 04 Aug 2023	AIC	2213.509
Time:	09:47:25	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	28.782	0.000	1.223	1.401
ar.L2	-0.5593	0.072	-7.740	0.000	-0.701	-0.418
ma.L1	-1.9917	0.109	-18.215	0.000	-2.206	-1.777
ma.L2	0.9999	0.110	9.108	0.000	0.785	1.215
sigma2	1.099e+06	2e-07	5.51e+12	0.000	1.1e+06	1.1e+06

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

Table 1.32 Auto ARIMA Model Summary

Diagnostic Plot

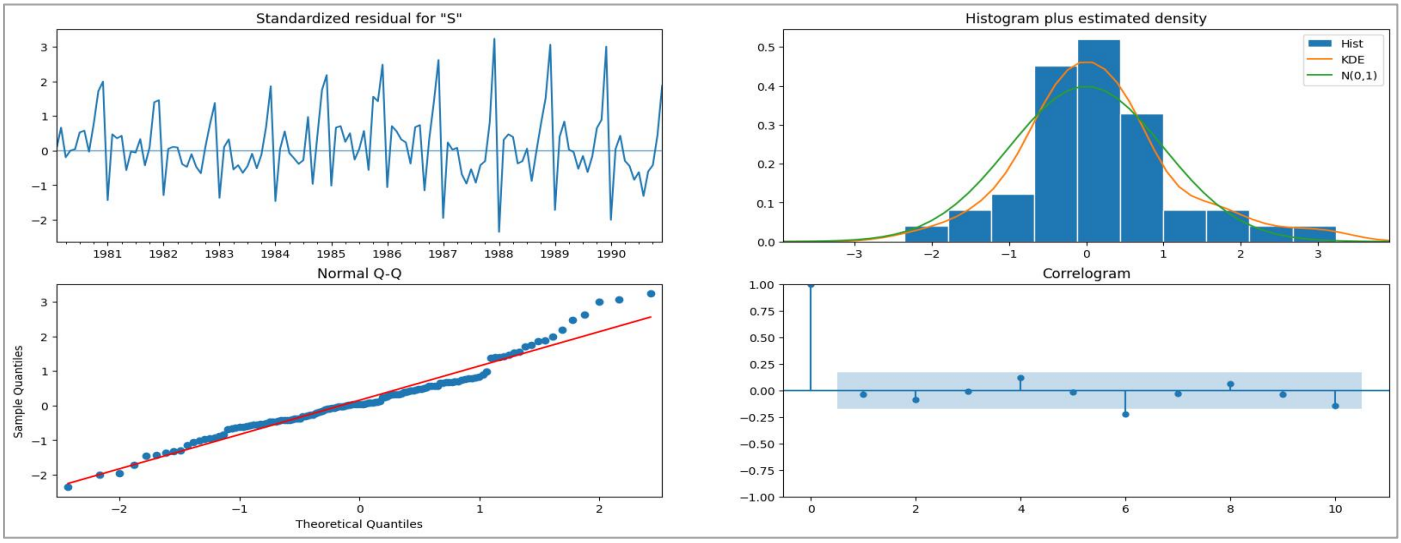

Fig 1.33 Diagnostic Plot: Automated ARIMA (2, 1, 2)

Forecast on the test data

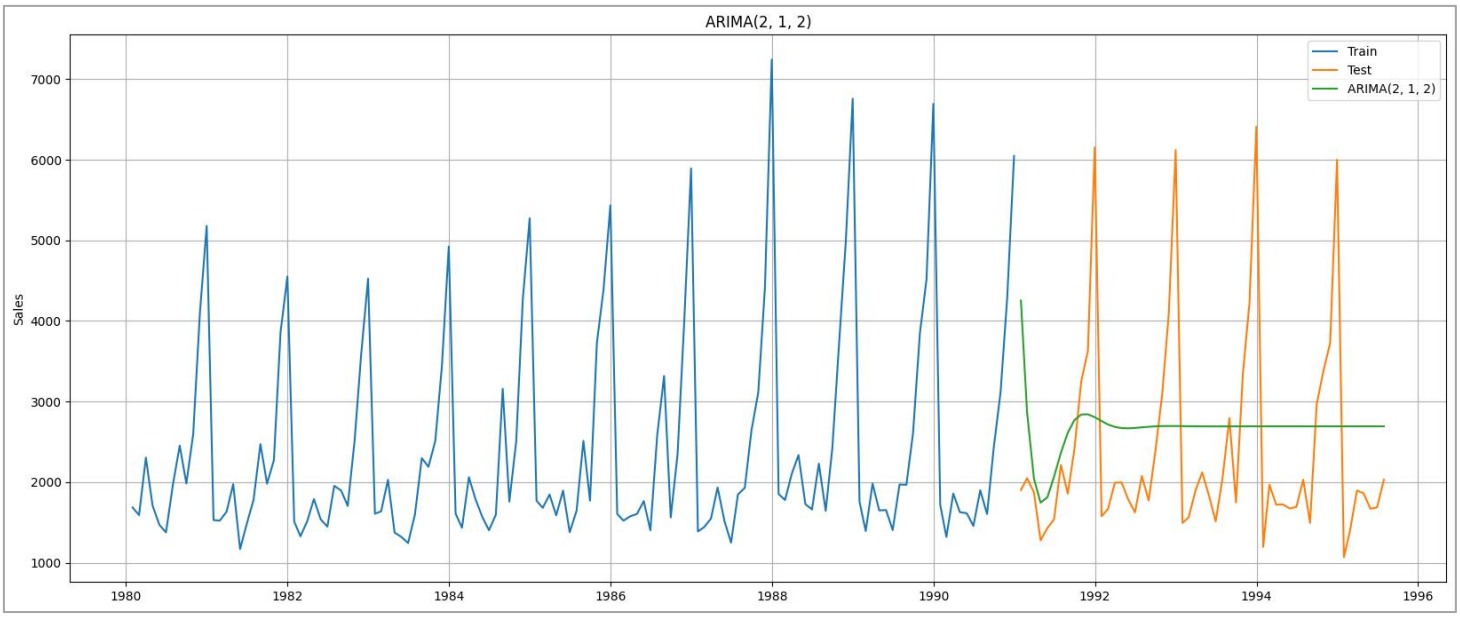

Fig 1.34 Time Series Plot: Automated ARIMA (2, 1, 2)

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Automated ARIMA (2, 1, 2)	1076.51	1299.98	49.42

Table 1.33 Model Performance Summary – Automated ARIMA (2, 1, 2)

- The **Auto ARIMA model** aims to **capture** the **underlying trend** in the data but **does not consider the seasonality component.**
- The model's performance is evaluated with a **Root Mean Square Error of 1299.98** and a **Mean Absolute Percentage Error of 49.42.** The model performed poorer on Test data as compared to Training

➢ Automated version of a SARIMA model

The ARIMA models can be extended/improved to handle seasonal components of a data series.

The seasonal autoregressive moving average model is given by **SARIMA (p, d, q)(P, D, Q)F**

The above model consists of:

- **Autoregressive** and **moving average components (p, q)**
- **Seasonal autoregressive** and **moving average components (P, Q)**
- The **ordinary** and **seasonal difference components** of order **'d' and 'D'**
- **Seasonal frequency 'F'**

The value for the parameters **(p,d,q) and (P, D, Q)** can be decided by comparing different values for each and taking the lowest AIC value for the model build.

The value for **F** can be consolidated by **ACF plot**

- **Without Seasonal Differencing (D = 0):**

Let us look at the differenced ACF plot again to understand the seasonal parameter for the SARIMA model

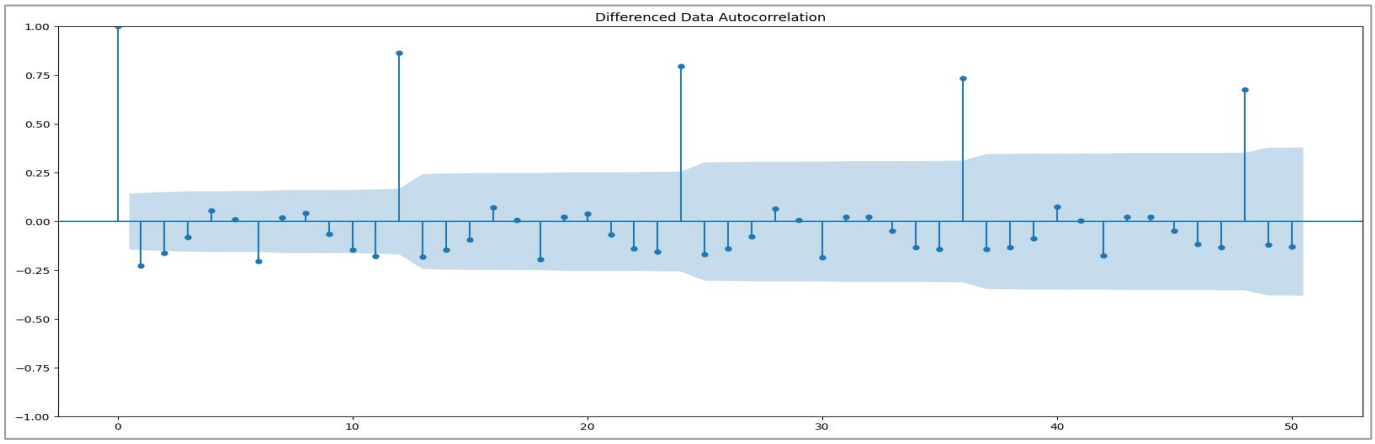

Fig 1.35 Differenced Autocorrelation Plot: SARIMA

- **S=12** is chosen for seasonal differencing as it is **significant**, and the **ACF plot at S=12 does not taper off.**
- This indicates the **presence of seasonality**, and applying seasonal differencing to the original series can improve the model's performance.
- **d = 1** to make the time series stationary
- Seasonal differencing **not yet applied** to make the time series stationary **D = 0**

To find the values of p, q, P & Q, we iterated values between and computed AIC values for each combination. Lowest 5 shown below. We built the model on the values: p = 1, d =1, q = 2, P = 1, D = 0, Q = 2, S = 12

	param	seasonal	AIC
50	(1, 1, 2)	(1, 0, 2, 12)	1555.584247
53	(1, 1, 2)	(2, 0, 2, 12)	1555.934563
26	(0, 1, 2)	(2, 0, 2, 12)	1557.121563
23	(0, 1, 2)	(1, 0, 2, 12)	1557.160507
77	(2, 1, 2)	(1, 0, 2, 12)	1557.340404

Table 1.34 SARIMA AIC Parameters without Seasoning

SARIMA Results

SARIMAX Results

Dep. Variable:	Sparkling	No. Observations:	132
Model:	SARIMAX(1, 1, 2)x(1, 0, 2, 12)	Log Likelihood	-770.792
Date:	Fri, 04 Aug 2023	AIC	1555.584
Time:	09:48:40	BIC	1574.095
Sample:	01-31-1980	HQIC	1563.083
	- 12-31-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.6282	0.255	-2.464	0.014	-1.128	-0.128
ma.L1	-0.1040	0.225	-0.463	0.644	-0.545	0.337
ma.L2	-0.7276	0.154	-4.736	0.000	-1.029	-0.427
ar.S.L12	1.0439	0.014	72.839	0.000	1.016	1.072
ma.S.L12	-0.5550	0.098	-5.663	0.000	-0.747	-0.363
ma.S.L24	-0.1354	0.120	-1.133	0.257	-0.370	0.099
sigma2	1.506e+05	2.03e+04	7.401	0.000	1.11e+05	1.9e+05

Ljung-Box (L1) (Q):	0.04	Jarque-Bera (JB):	11.72
Prob(Q):	0.84	Prob(JB):	0.00
Heteroskedasticity (H):	1.47	Skew:	0.36
Prob(H) (two-sided):	0.26	Kurtosis:	4.48

Table 1.35 Auto SARIMA without Differencing Model Summary

Diagnostic Plot

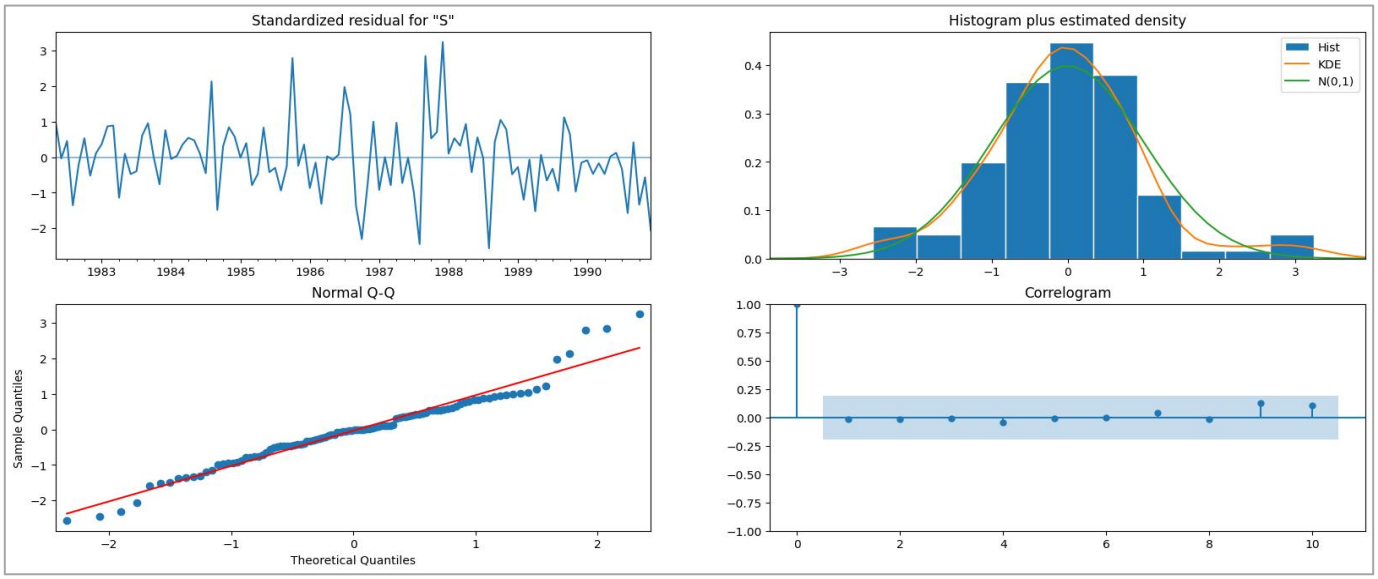

Fig 1.36 Diagnostic Plot: Automated SARIMA (1, 1, 2)(1, 0, 2, 12)

Forecast on the test data

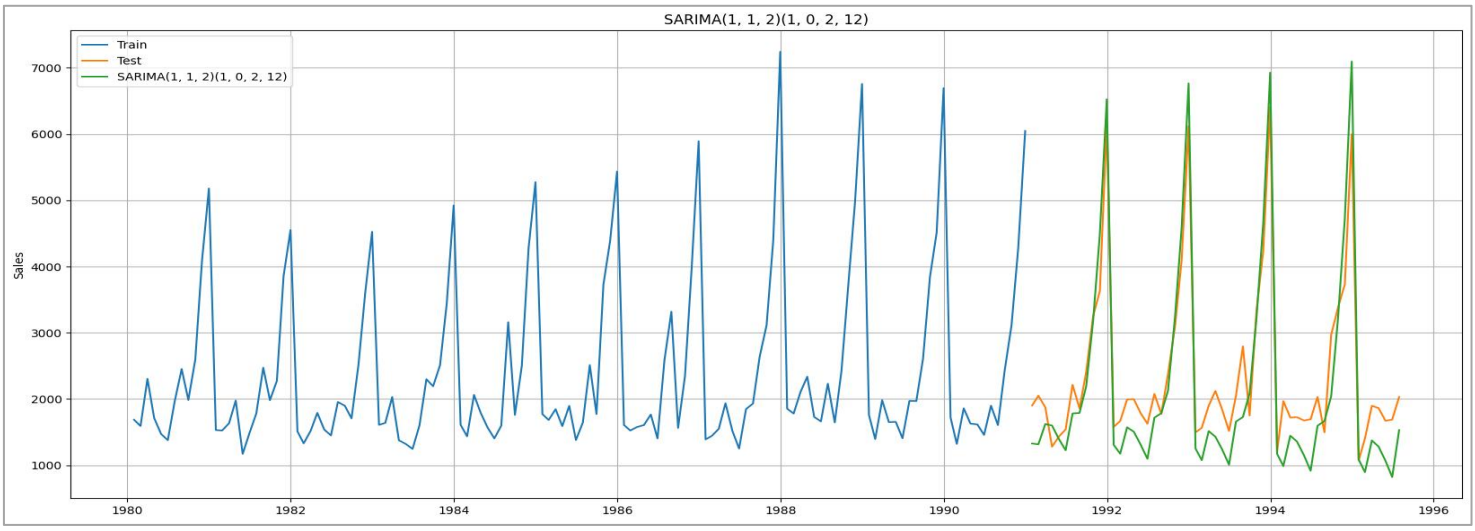

Fig 1.37 Time Series Plot: Automated SARIMA (1, 1, 2)(1, 0, 2, 12)

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Automated SARIMA (1, 1, 2)(1, 0, 2, 12)	592.29	528.66	59.91

Table 1.36 Model Performance Summary – Automated SARIMA (1, 1, 2)(1, 0, 2, 12)

- The **Automated SARIMA (1, 1, 2)(1, 0, 2, 12)** aims to **capture** the **underlying trend** in the data as well as **the seasonality component.**
- The model's performance is evaluated with a **Root Mean Square Error of 528.66** and a **Mean Absolute Percentage Error of 59.91.** The model **performed well** as compared to **ARIMA, best after Triple Smoothening Methods.**

- **With Seasonal Differencing (D = 1):**

As noticed from the monthly plot shown below, the time series is relatively constant for each month except for December, which shows a pattern. Since December has the highest sales, it has a significant impact on the time series.

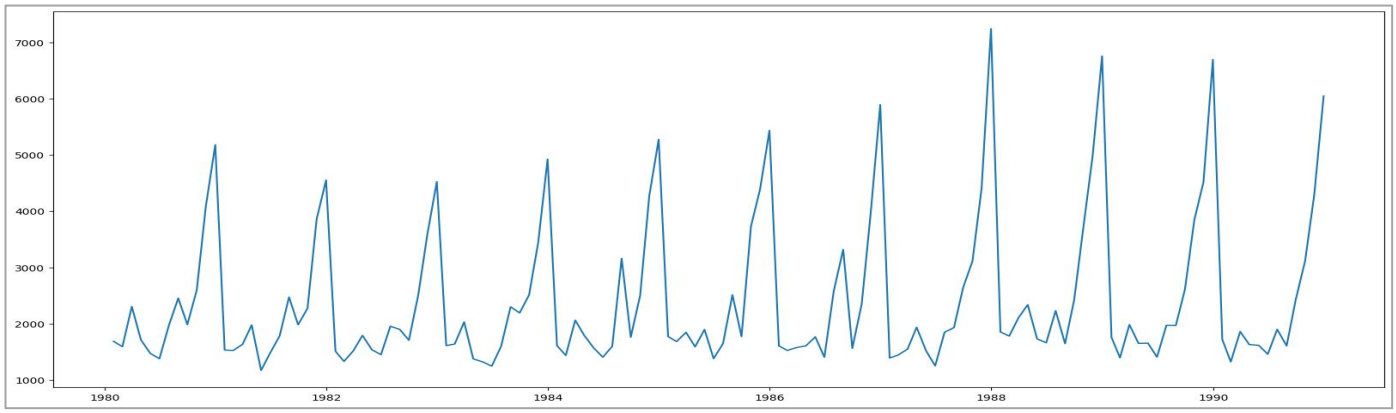

Fig 1.38 Time Series Plot: Train Data

To address this, we **apply a 12-month seasonal difference** to the data and examine if it makes the training data stationary.

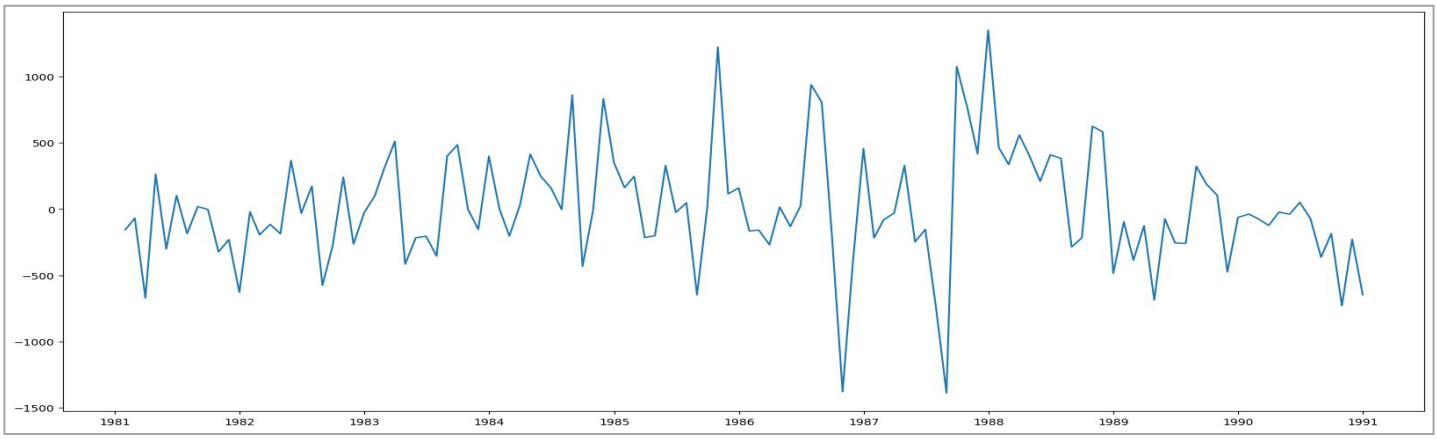

Fig 1.39 Time Series Plot: Test Data

The time series looks stationary. Let's check for **stationarity** using **Augmented Dickey – Fuller Test.**

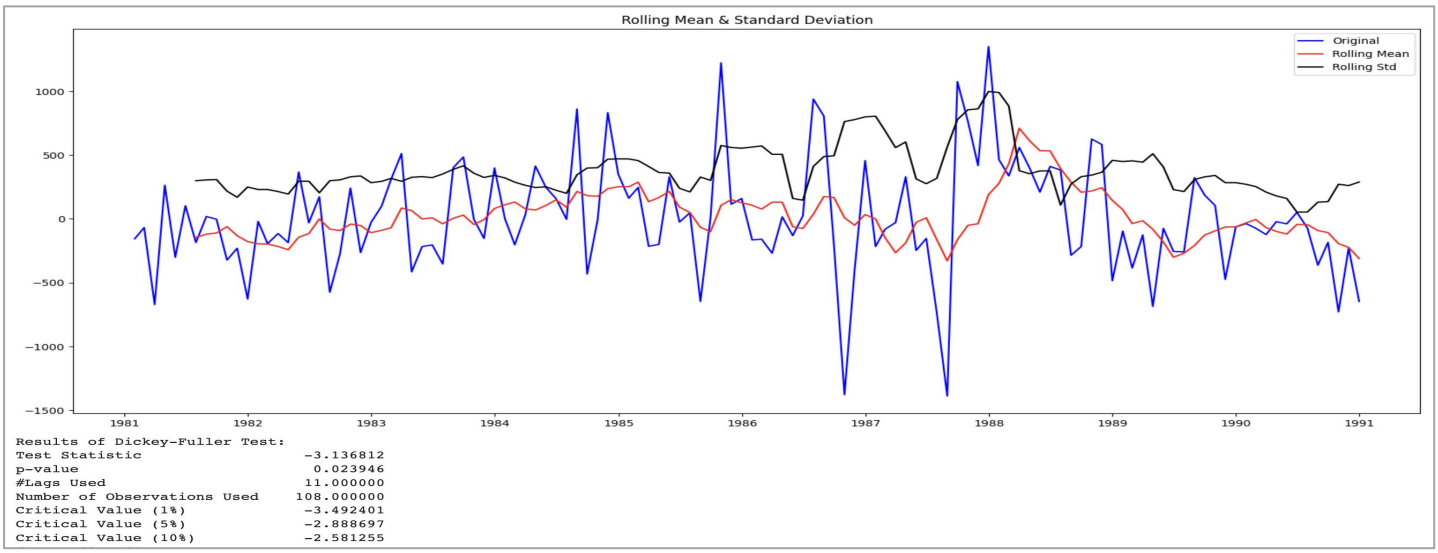

Fig 1.40 Stationarity of Differenced Training Data Using AD Fuller Test (D=1)

- The **p-value is less than 0.05**, leading us to **reject the null hypothesis (Ho)** at 95% confidence level and conclude that the **time series is stationary**.
- Therefore, we can build our model **with d = 0 (regular differencing)** and **D = 1 (seasonal differencing),** as applying only seasonal differencing makes the time series stationary and prevents over-differencing.

To find the values of p, q, P & Q, we iterated values between and computed AIC values for each combination. Lowest 5 shown below. We built the model on the values: p = 0, d =0, q = 2, P = 0, D =1, Q = 2, S = 12

	param	seasonal	AIC
20	(0, 0, 2)	(0, 1, 2, 12)	1397.037021
47	(1, 0, 2)	(0, 1, 2, 12)	1397.239335
74	(2, 0, 2)	(0, 1, 2, 12)	1397.783884
23	(0, 0, 2)	(1, 1, 2, 12)	1398.924576
50	(1, 0, 2)	(1, 1, 2, 12)	1399.001907

Table 1.37 SARIMA AIC Parameters with Seasoning

SARIMA Results

SARIMAX Results

Dep. Variable:	Sparkling	No. Observations:	132
Model:	SARIMAX(0, 0, 2)x(0, 1, 2, 12)	Log Likelihood	-693.519
Date:	Fri, 04 Aug 2023	AIC	1397.037
Time:	09:49:40	BIC	1409.700
Sample:	01-31-1980	HQIC	1402.150
	- 12-31-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	0.2746	0.106	2.583	0.010	0.066	0.483
ma.L2	-0.0647	0.116	-0.555	0.579	-0.293	0.164
ma.S.L12	-0.3647	0.098	-3.715	0.000	-0.557	-0.172
ma.S.L24	-0.0430	0.139	-0.309	0.757	-0.316	0.230
sigma2	1.808e+05	2.28e+04	7.917	0.000	1.36e+05	2.26e+05

Ljung-Box (L1) (Q):	0.04	Jarque-Bera (JB):	17.47
Prob(Q):	0.85	Prob(JB):	0.00
Heteroskedasticity (H):	0.68	Skew:	0.65
Prob(H) (two-sided):	0.29	Kurtosis:	4.68

Table 1.38 Auto SARIMA with Differencing Model Summary

Diagnostic Plot

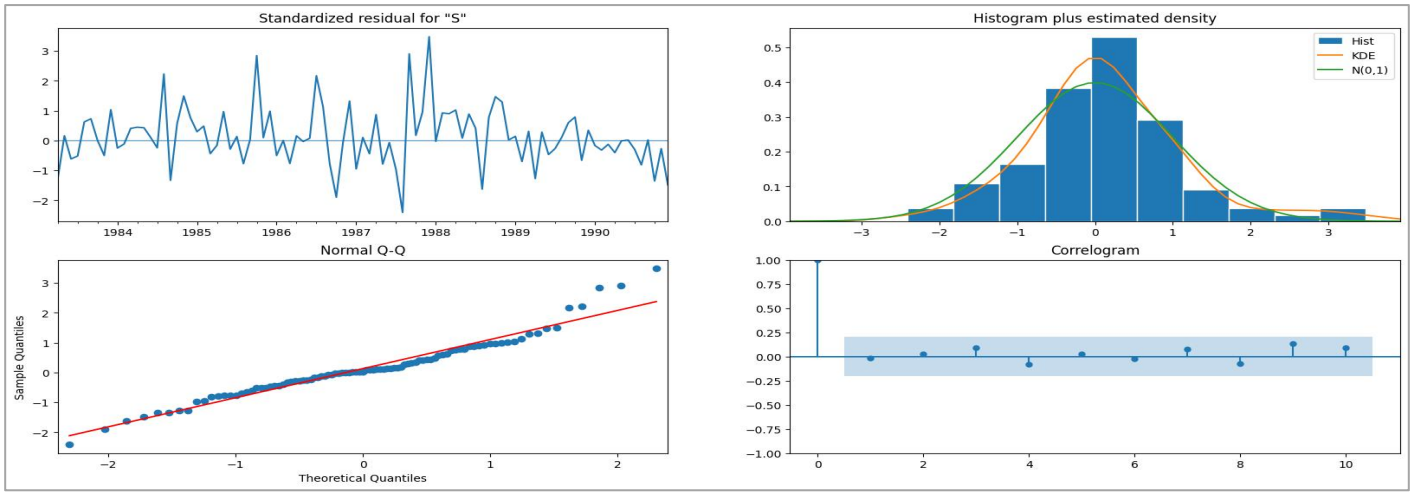

Fig 1.41 Diagnostic Plot: Automated SARIMA(0, 0, 2)(0, 1, 2, 12)

Forecast on the test data

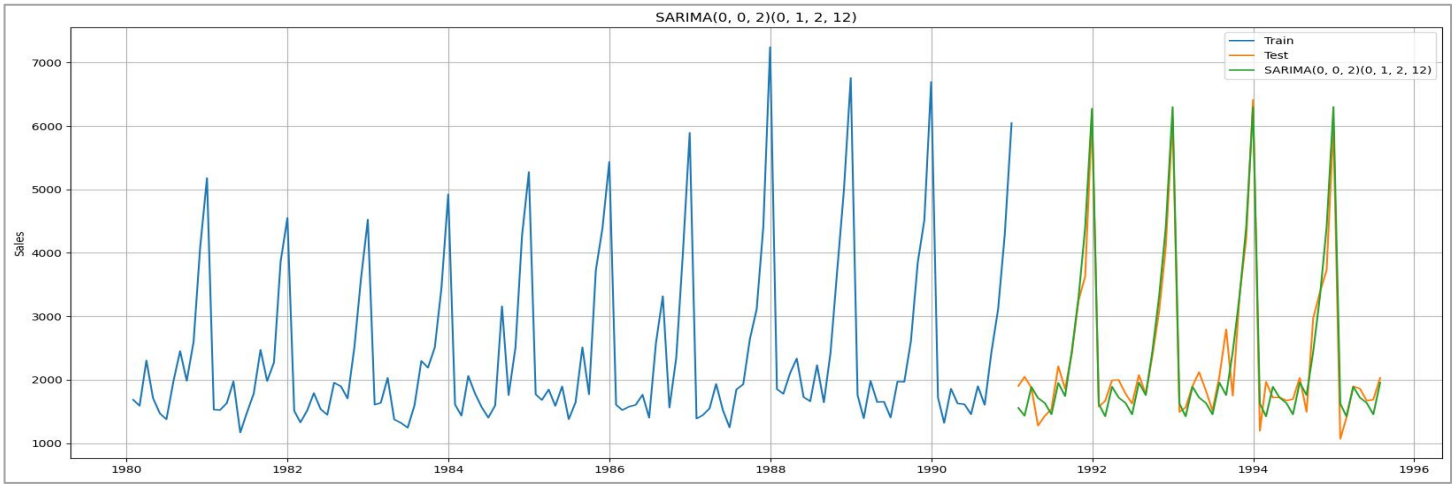

Fig 1.42 Time Series Plot: Automated SARIMA(0, 0, 2)(0, 1, 2, 12)

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Automated SARIMA(0, 0, 2)(0, 1, 2, 12)	861.69	317.06	51.69

Table 1.39 Model Performance Summary – Automated SARIMA(0, 0, 2)(0, 1, 2, 12)

- The SARIMA model **successfully captures** both the **trend and seasonality** in the data.
- The **Root Mean Square Error is 317.06,** and the **Mean Absolute Percentage Error is 51.69** for the **automated SARIMA** model **with seasonal differencing.**
- This model performs **better than** the model **without seasonal differencing,** indicating that **incorporating seasonal differencing improves the accuracy** of the forecast.
- The model is the second **best performed so far**

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

Final Results Table:

Model Name	Train_RMSE	Test_RMSE	MAPE
Alpha = 0.01, Beta = 0.04, Gamma = 0.25 Triple Exponential Smoothing	470.84	302.73	49.86
Automated SARIMA(0, 0, 2)(0, 1, 2, 12)	861.69	317.06	51.69
Alpha = 0.111, Beta = 0.049, Gamma = 0.362 Triple Exponential Smoothing	355.77	403.71	48.37
Automated SARIMA(1, 1, 2)(1, 0, 2, 12)	592.29	528.66	59.91
2 point Trailing Moving Average	706.18	813.4	24.71
Simple Average Model	1298.48	1275.08	39.16
Alpha = 0.02, Beta = 0.38 Double Exponential Smoothing	1398.03	1275.87	40.97
Alpha = 0.02 Simple Exponential Smoothing	1346.26	1278.5	42.41
Automated ARIMA(2, 1, 2)	1076.51	1299.98	49.42
Alpha = 0.07 Simple Exponential Smoothing	1322.9	1338.0	53.88
Linear Regression On Time	1279.32	1389.14	59.35
NaiveModel	3867.7	3864.28	201.33
Alpha=0.665, Beta = 0.0001 Double Exponential Smoothing	1339.5	5291.88	268.91

Table 1.40 Model Performance Summary – Consolidated

- **Model Performance:** After evaluating various forecasting models, the top-performing ones are:
  - **Tuned Triple Exponential Model** (Alpha = 0.01, Beta = 0.04, Gamma = 0.25): It shows the **best accuracy** with a Test RMSE of 302.73 and MAPE of 49.86.
  - **Automated SARIMA with Seasonal Differencing** (SARIMA(0, 0, 2)(0, 1, 2, 12)): It also **performs well**, with a Test RMSE of 317.06 and MAPE of 51.69.
  - **Triple Exponential Smoothing** (Alpha = 0.111, Beta = 0.049, Gamma = 0.362): This model is the third-best with a Test RMSE of 403.71 and MAPE of 48.37.
  - On the other hand, the Alpha=0.665, Beta = 0.0001 **Double Exponential Smoothing model** performs **poorly** with a high Test RMSE of 5291.88 and MAPE of 268.91.

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

- **The most optimum models for forecasting are as shown below:**

Final Results Table:

Model Name	Train_RMSE	Test_RMSE	MAPE
Alpha = 0.01, Beta = 0.04, Gamma = 0.25 Triple Exponential Smoothing	470.84	302.73	49.86
Automated SARIMA(0, 0, 2)(0, 1, 2, 12)	861.69	317.06	51.69
Alpha = 0.111, Beta = 0.049, Gamma = 0.362 Triple Exponential Smoothing	355.77	403.71	48.37
Automated SARIMA(1, 1, 2)(1, 0, 2, 12)	592.29	528.66	59.91
2 point Trailing Moving Average	706.18	813.4	24.71

Table 1.41 Best Performing Models

- **Tuned Triple Exponential Model** with Alpha = 0.01, Beta = 0.04, and Gamma = 0.25 having **test RMSE 302.73 and MAPE 49.96**
- **Automated SARIMA with seasonal differencing** – SARIMA(0, 0, 2)(0, 1, 2, 12) having **test RMSE 317.06 and MAPE 51.69**
- **Triple Exponential Smoothing** with Alpha = 0.111, Beta = 0.049, and Gamma = 0.362, which has **test RMSE 403.71** and **MAPE 48.37.**

We'll forecast on Top 2 Models

- **Forecasting on Tuned Triple Exponential Model with Alpha = 0.01, Beta = 0.04, Gamma = 0.25**

To **forecast 12 months** into the future, we build the model on the **full data first** before forecasting

RMSE Full Model = 416.50

- **Assumption**: Forecast distribution's standard deviation ≈ Residual standard deviation.

- **Purpose**: Helps estimate uncertainty in the forecast.
- **Use**: Construct confidence intervals with a specified level of confidence.

Forecast Results: –

	lower_ci	prediction	upper_ci
1995-08-31	1200.848838	1996.325100	2791.801361
1995-09-30	1592.834233	2388.310494	3183.786755
1995-10-31	2482.423911	3277.900172	4073.376433
1995-11-30	3277.056202	4072.532463	4868.008724
1995-12-31	5380.157921	6175.634182	6971.110443
1996-01-31	625.254007	1420.730268	2216.206529
1996-02-29	838.088822	1633.565084	2429.041345
1996-03-31	1075.186663	1870.662925	2666.139186
1996-04-30	1030.764395	1826.240656	2621.716917
1996-05-31	885.435997	1680.912258	2476.388519
1996-06-30	794.407425	1589.883686	2385.359947
1996-07-31	1249.849273	2045.325534	2840.801795

Table 1.42 Forecast Results – Triple Exponential Model with Alpha = 0.01, Beta = 0.04, Gamma = 0.25

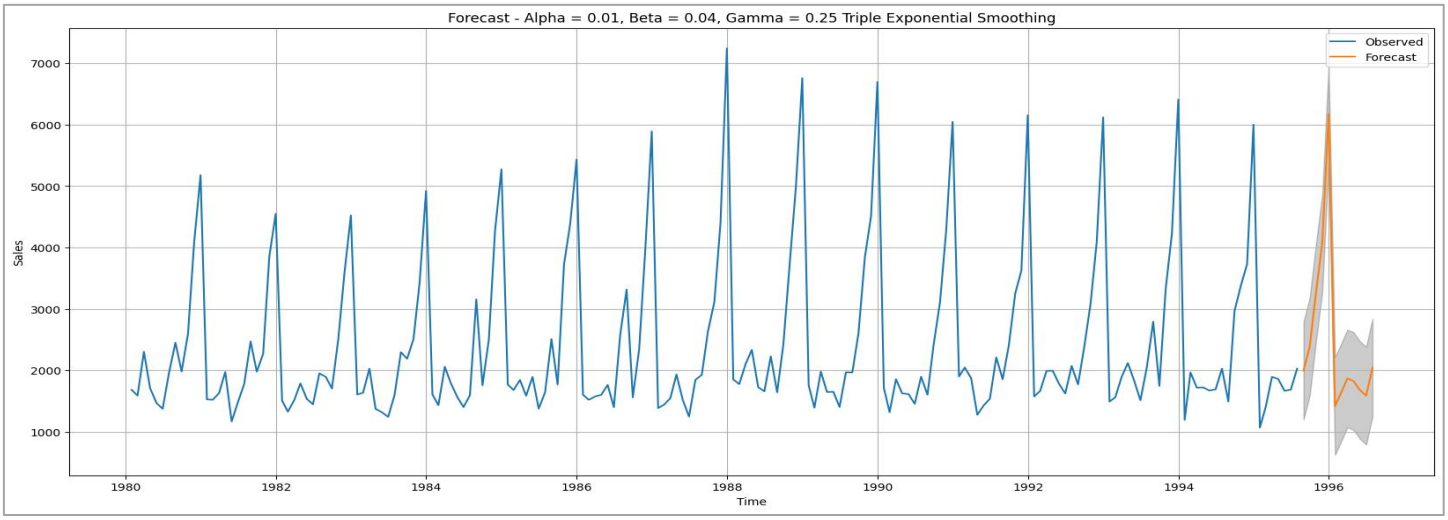

Fig 1.43 Forecasted Plot: Triple Exponential Model with Alpha = 0.01, Beta = 0.04, Gamma = 0.25

➢ Forecasting on Automated SARIMA with seasonal differencing – SARIMA(0, 0, 2)(0, 1, 2, 12)

- **Stationarity Check on Full data:**

After applying seasonal differencing (D=12), at **p–value < 0.05** we **reject the Null Hypothesis** & conclude that the **full data** is also **Stationary at 95% confidence level.**

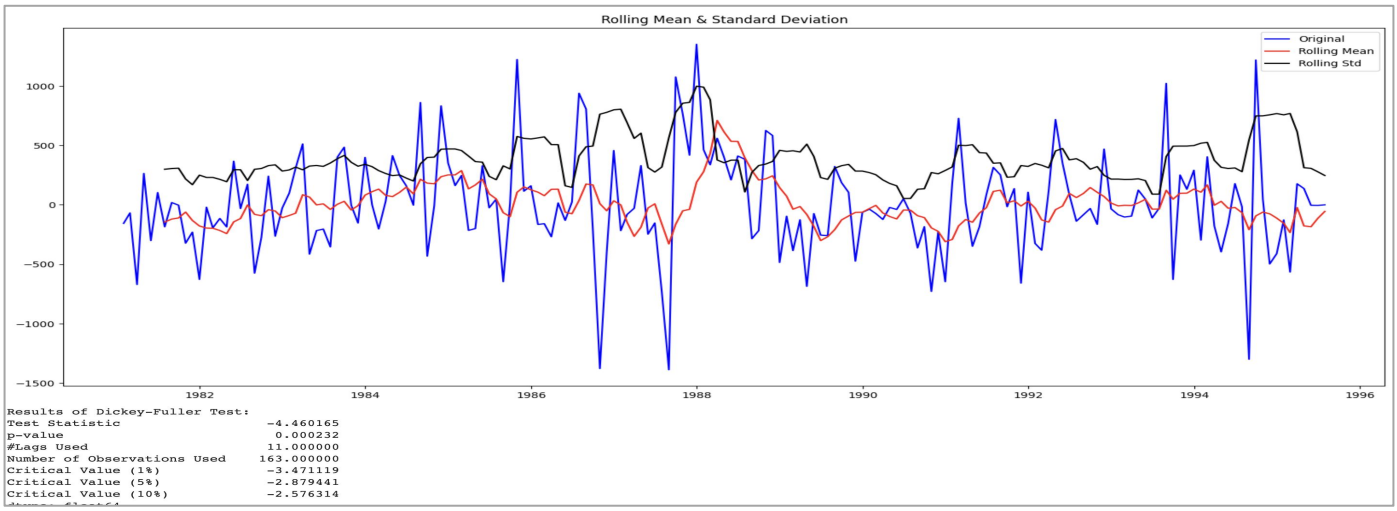

Fig 1.44 Stationarity of Differenced Data Using AD Fuller (D=12)

SARIMAX Results

Dep. Variable:	Sparkling	No. Observations:	187
Model:	SARIMAX(1, 1, 2)x(1, 0, 2, 12)	Log Likelihood	-1173.413
Date:	Fri, 04 Aug 2023	AIC	2360.827
Time:	09:49:50	BIC	2382.309
Sample:	01-31-1980	HQIC	2369.551
	- 07-31-1995		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.6610	0.242	-2.734	0.006	-1.135	-0.187
ma.L1	-0.2739	0.200	-1.369	0.171	-0.666	0.118
ma.L2	-0.8113	0.227	-3.578	0.000	-1.256	-0.367
ar.S.L12	1.0157	0.012	84.456	0.000	0.992	1.039
ma.S.L12	-1.3873	0.338	-4.102	0.000	-2.050	-0.724
ma.S.L24	-0.1461	0.146	-1.001	0.317	-0.432	0.140
sigma2	5.948e+04	1.84e+04	3.231	0.001	2.34e+04	9.56e+04

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	27.47
Prob(Q):	0.96	Prob(JB):	0.00
Heteroskedasticity (H):	1.03	Skew:	0.52
Prob(H) (two-sided):	0.93	Kurtosis:	4.76

Table 1.43 Auto SARIMA Forecast Model Summary

Diagnostic Plot

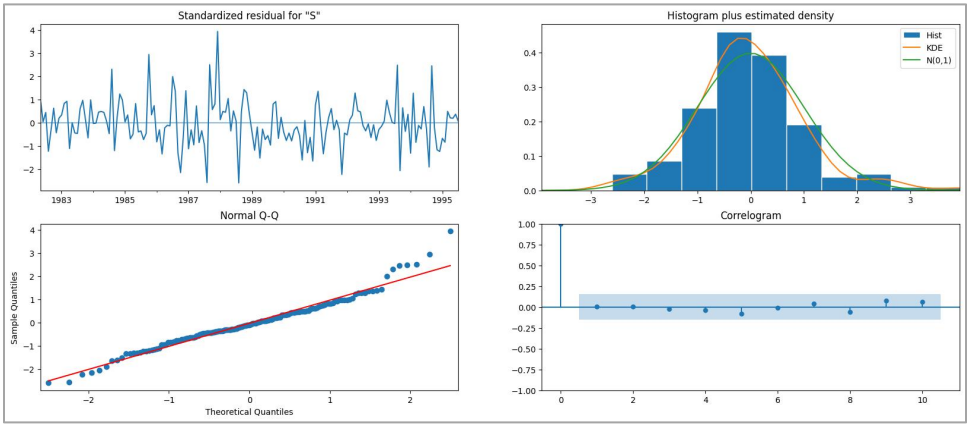

Fig 1.45 Diagnostic Plot: Automated SARIMA(0, 0, 2)(0, 1, 2, 12)

Forecast Results: –

Sparkling	mean	mean_se	mean_ci_lower	mean_ci_upper
1995-08-31	1836.367093	379.710188	1092.148801	2580.585385
1995-09-30	2489.589237	384.474504	1736.033056	3243.145418
1995-10-31	3324.586557	384.580305	2570.823009	4078.350105
1995-11-30	4020.224547	386.338744	3263.014522	4777.434571
1995-12-31	6289.999972	386.393292	5532.683035	7047.316909
1996-01-31	1244.690294	387.303811	485.588774	2003.791815
1996-02-29	1533.142520	387.532185	773.593395	2292.691645
1996-03-31	1821.702218	388.159318	1060.923933	2582.480502
1996-04-30	1788.492996	388.499401	1027.048163	2549.937830
1996-05-31	1627.566794	389.018579	865.104390	2390.029197
1996-06-30	1563.321091	389.414529	800.082639	2326.559542
1996-07-31	2000.702985	389.889175	1236.534244	2764.871726

Table 1.44 Forecast Results – SARIMA(0, 0, 2)(0, 1, 2, 12)

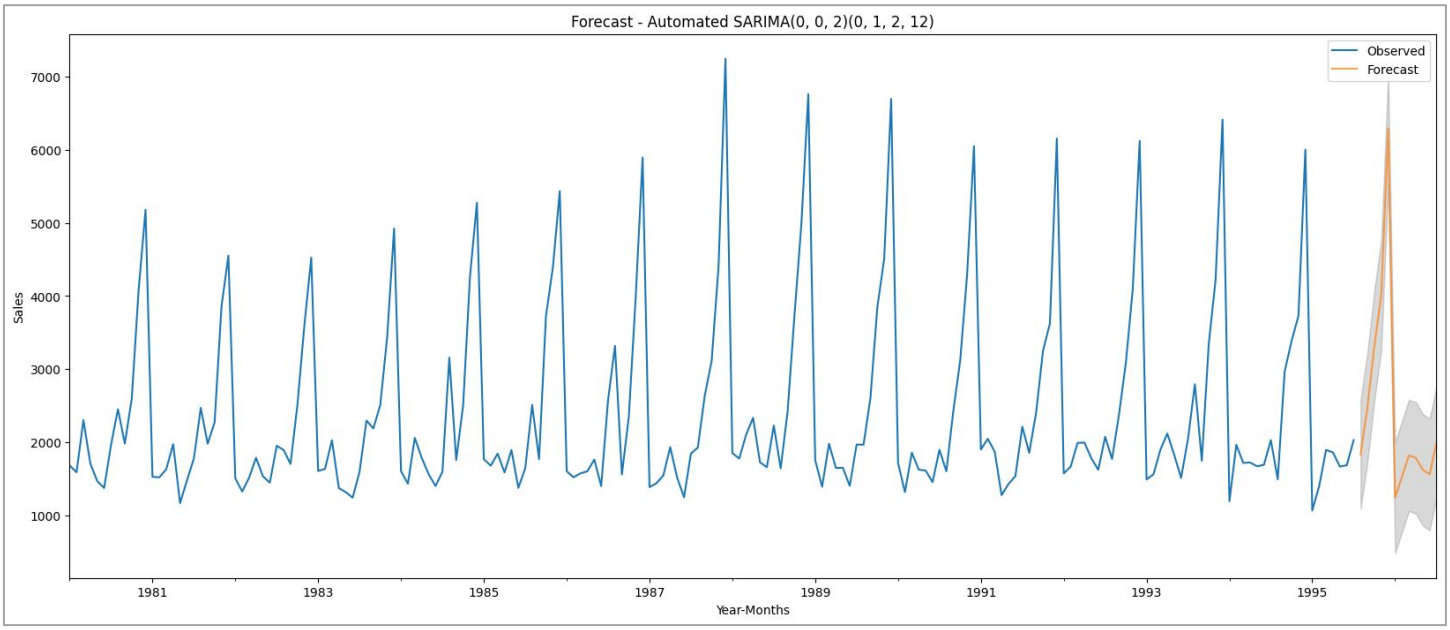

Fig 1.46 Forecasted Plot : SARIMA(0, 0, 2)(0, 1, 2, 12)

- The predictions from both the models consistently **indicate** that the **sales will exhibit a steady trend** and **seasonality** similar to previous years, with a **slight upswing** during the **holiday season (November–December)** compared to the previous year.

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

Forecasting Insights:

- **Data Analysis:**
  - The sales of Sparkling products have shown a **steady increasing trend** over the years, indicating a **growth pattern** in customer demand.
  - **Seasonal patterns** are evident in the data, with **sales spiking during November and December,** which can be attributed to the holiday season when customers tend to buy more Sparkling products for celebrations.

- **Time Series Characteristics:**
  - Based on the decomposed data provided, the time series is observed to be of a **multiplicative nature.**
  - The **trend** appears to be **increasing** over time with some fluctuations, suggesting a **growth pattern**.
  - **Seasonality** shows both positive and negative values, indicating regular cycles or seasonal effects.
  - The residual component includes random fluctuations and unexplained variance in the time series.

- **Model Performance**
  - After evaluating various forecasting models, the top-performing ones are:
    - **Tuned Triple Exponential Model** (Alpha = 0.01, Beta = 0.04, Gamma = 0.25): It shows the **best accuracy** with a Test RMSE of 302.73 and MAPE of 49.86.
    - **Automated SARIMA with Seasonal Differencing** (SARIMA(0, 0, 2)(0, 1, 2, 12)): It also **performs well**, with a Test RMSE of 317.06 and MAPE of 51.69.
    - **Triple Exponential Smoothing** (Alpha = 0.111, Beta = 0.049, Gamma = 0.362): This model is the third-best with a Test RMSE of 403.71 and MAPE of 48.37.
    - On the other hand, the Alpha=0.665, Beta = 0.0001 **Double Exponential Smoothing model** performs **poorly** with a high Test RMSE of 5291.88 and MAPE of 268.91.

- **Predictions Model**
  - For comprehensive forecasting and predicting 12 months ahead, we built the following models:
    - **Tuned Triple Exponential Model** with Alpha = 0.01, Beta = 0.04, and Gamma = 0.25, the RMSE of the Full Model is **416.50.**
    - **Automated SARIMA with Seasonal Differencing** (SARIMA(0, 0, 2)(0, 1, 2, 12)): It also performs well, with a full model RMSE of **539.99**

➢ Measures for Future Sales

- Based on the analysis, the company can take the following measures to improve future sales:
  - **Capitalize on Seasonal Trends:** With observed seasonal patterns during November and December, the company should plan production and marketing efforts to meet increased demand during holiday seasons.
  - **Inventory Management:** Implement effective inventory management to avoid stockouts during peak periods and minimize excess inventory during slower periods.
  - **Pricing Strategy:** Utilize dynamic pricing to adjust prices during peak and off-peak periods, attracting more customers and optimizing revenue.
  - **Customer Engagement**: Strengthen customer relationships through personalized offers, loyalty programs, and active engagement to foster repeat purchases.

Problem 2: Rose Wine Sales

Problem Statement:

As an analyst in the ABC Estate Wines, your task is to analyse and forecast Wine Sales in the 20th century.

Data set for the Problem: Rose.csv

Data Dictionary:

YearMonth : Month & Year of the sale

Rose: Total Number of Rose Wine sales in particular Month–Year

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

Basic Information about the dataset

- **Sample of the dataset:** First & last 5 values of the dataset:

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

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

Table 2.1 First 5 and Last 5 Samples of the Dataset

- Converting the **YearMonth** Column to **DatetimeIndex & dropping** default index. Sample:

Year_Month	Rose
1980-01-31	112.0
1980-02-29	118.0
1980-03-31	129.0
1980-04-30	99.0
1980-05-31	116.0

Table 2.2 First 5 Samples of the Converted Dataset

- **Information about the dataset:**

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

Table 2.3 Info of the Dataset

- The DataFrame has **187 entries** with a **DatetimeIndex** ranging from **January 1980 to July 1995.**
- The **'Rose'** column is of **integer type** (int64), and it has **187 non-null values.**

- **Time Series Plot:**

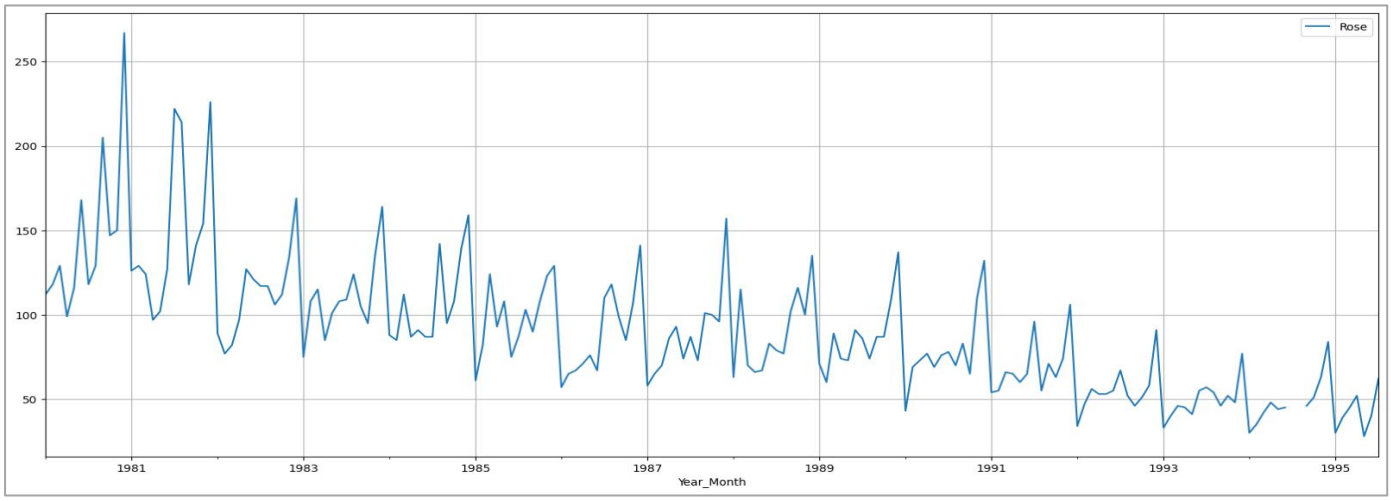

Fig 2.1a Time Series Plot

- **Decreasing Trend:** The data shows a **consistent decrease in Rose wine sales** over the years, suggesting a declining pattern in customer demand for this type of wine.
- **Seasonal Patterns:** Despite the overall decreasing trend, there are still **seasonal variations** in the data, with sales peaking in certain months and dropping in others.
- **Missing Data:** There are **missing values** in the **year 1994** and which will be handled accordingly

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

- **Missing Values:**
  - There are 2 Null values in the dataset, for 07–1994 & 08–1994, which need to be corrected. Let's check the Null values
  - Interpolate the missing values using cubic interpolation.

Year_Month	Rose
1994-07-31	NaN
1994-08-31	NaN

Year_Month	Rose
1994-07-31	45.270014
1994-08-31	44.504928

Table 2.4 Missing Values **Before** & **After** Treatment

- **Time Series Plot post Null Treatment:**

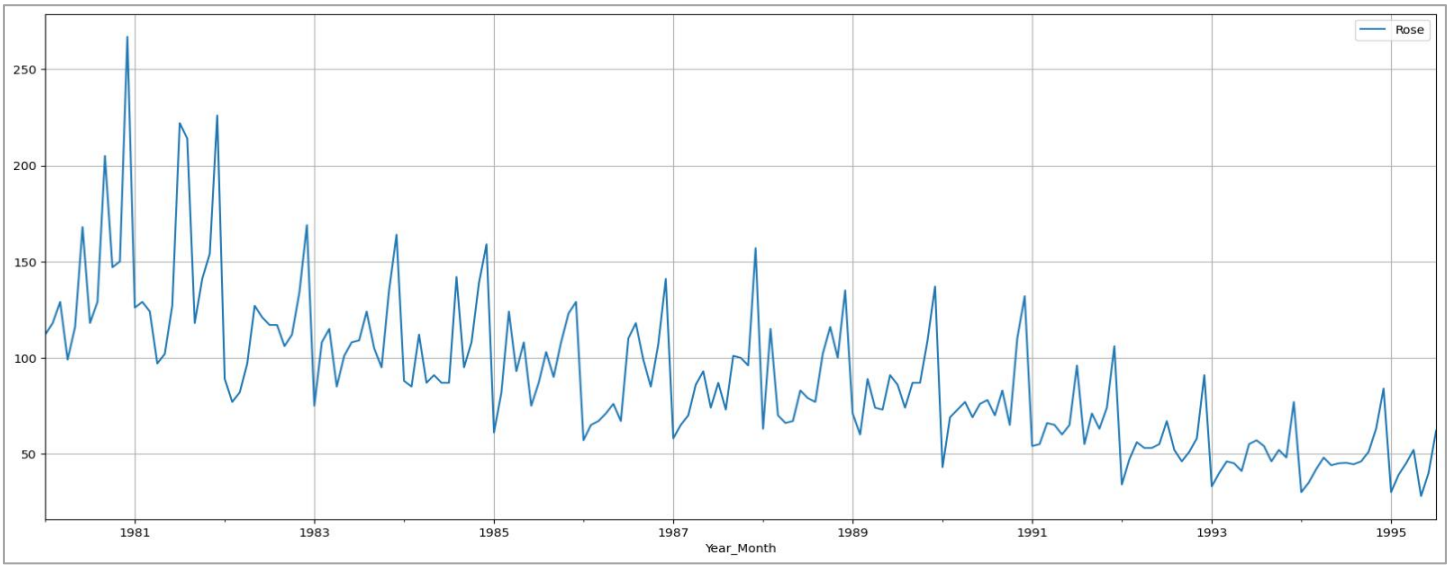

Fig 2.1b Time Series Plot Post Null Treatment

- **Duplicate Values:**
  - The dataset shows 90 Duplicates rows for Rose wine sales, however, when checked further, these were the same no of sales at different year. Hence, we conclude that **there are no duplicate values**
- **Descriptive Statistics:**

	count	mean	std	min	25%	50%	75%	max
Rose	187.0	89.907887	39.245847	28.0	62.5	85.0	111.0	267.0

Table 2.5 Descriptive Statistics

  - The dataset contains **187 observations** of Rose product sales.
  - It shows **significant variability**, with sales ranging from a **minimum of 28 units** to a **maximum of 267 units.**
  - On **average**, the monthly sales of Rose wine are **approximately 89.91 units**
  - The data is **right-skewed,** with the **mean 89.91** being **higher than** the **median 85**

- **Histogram & Boxplot:**

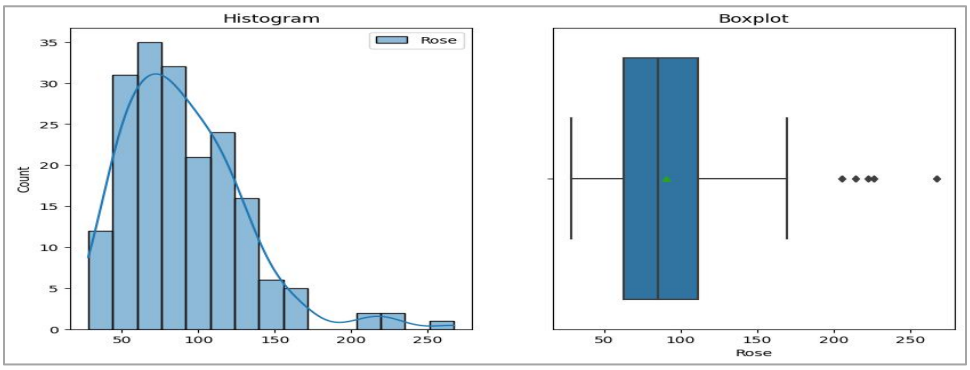

Fig 2.2 Histogram & Boxplot

  - The dataset is **right skewed** with the presence of outliers on the right tail

- **Spread of Sales: Year–on–Year Boxplot**

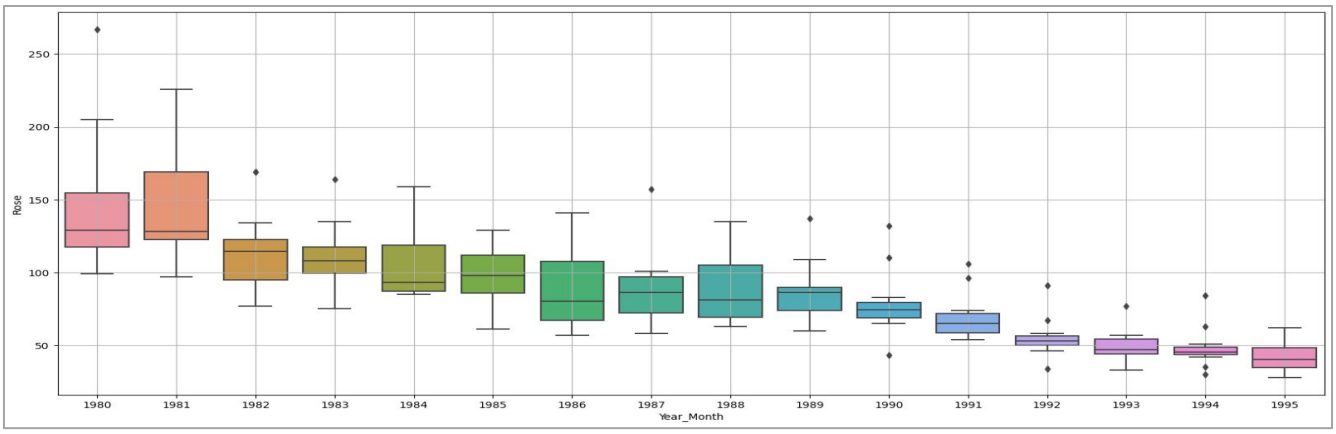

Fig 2.3 Spread of Sales Across Different Years

  - Sales of Rose products have been **declining** over time.
  - The dataset is **skewed** with the presence of **outliers on the right tail.**
  - There is significant variability in sales from year to year
  - There are a few outliers in the data, which could be due to special promotions, holidays or other occasions

- **Spread of Sales: Month–on– Month**

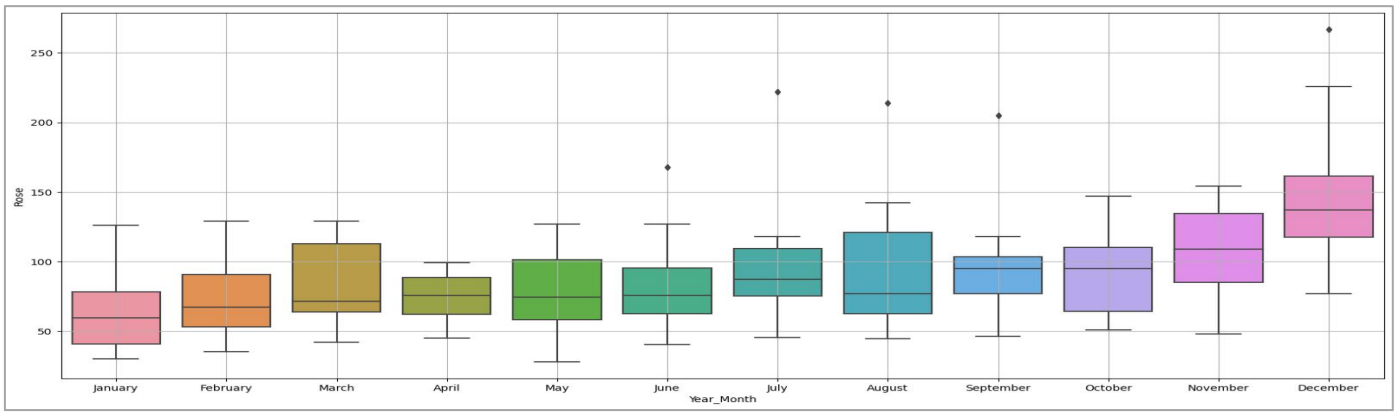

Fig 2.4 Spread of Sales Across Different Months

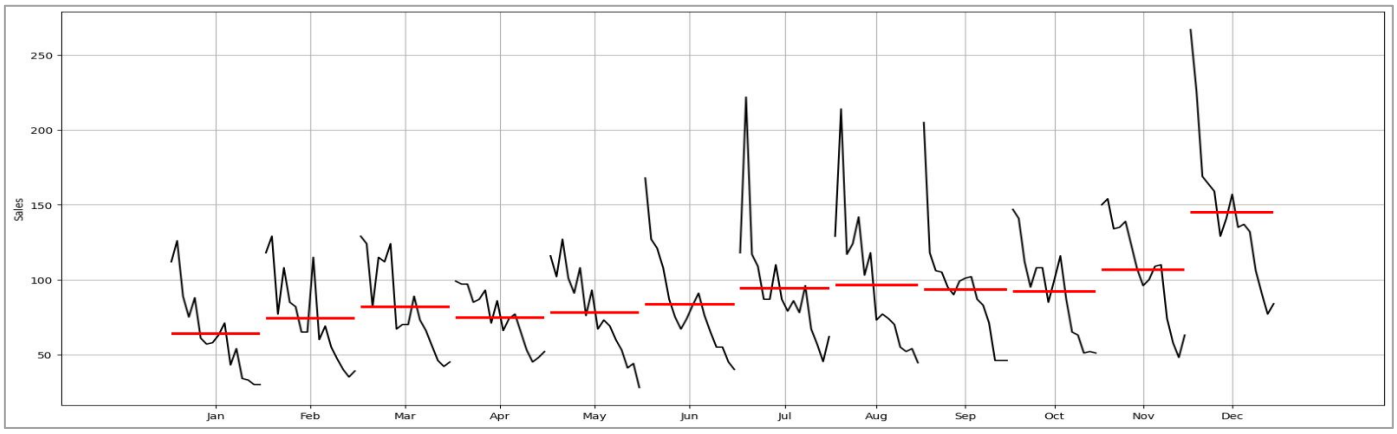

Fig 2.5 Distribution of time series across different months

- The Month Plot **demonstrates a downward trend** across all years for each month, indicating **consistent seasonality** throughout the months.
- Sales of Rose wine **exhibit seasonality**, with a significant **increase from August to December,** peaking in **December** due to the winter season and Christmas Holidays.
- There is a sharp **drop** in sales in **January** (post the holiday season), and stability can be seen in the mid-months (March to August).
- **Outliers** on the maximum side likely correspond to **December sales.**

➢ **Spread of Sales: Year-on-Year & Monthly Comparison:**

Year_Month	1	2	3	4	5	6	7	8	9	10	11	12
Year_Month												
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	45.270014	44.504928	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 2.6 Year-on-Year Monthly Sales

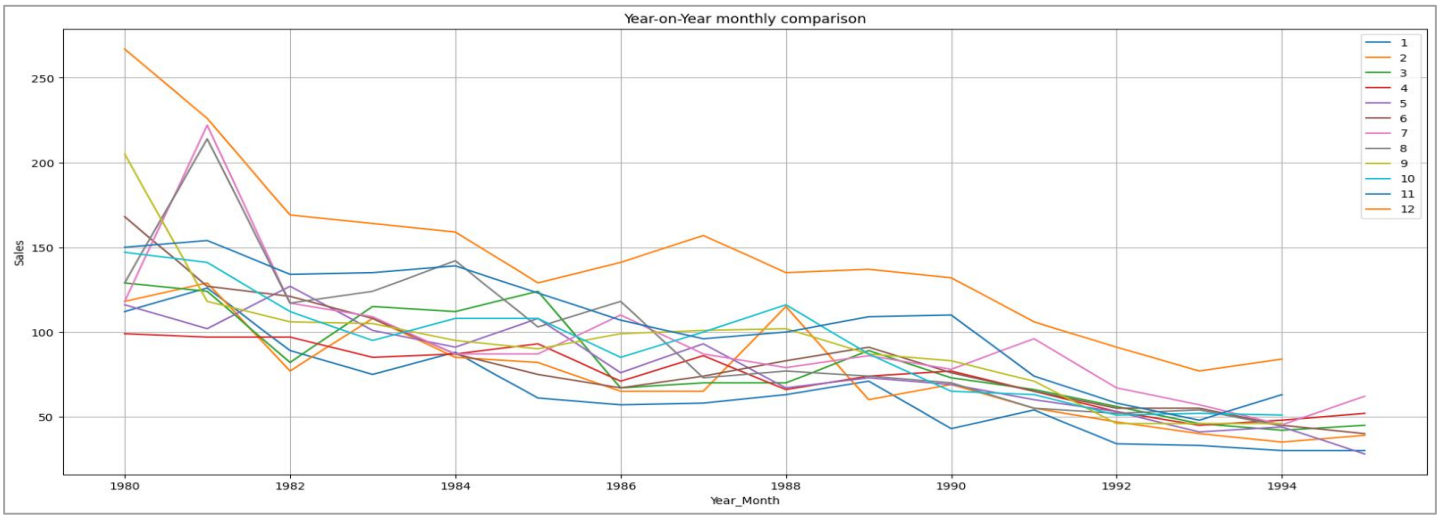

Fig 2.6 Year–on–Year Monthly Comparison

- **December** consistently shows the **highest sales across all the years,** while **January** has the l**owest sales** for most years.
- A noticeable **downward trend** can be observed across all months over the years.
- The outliers in the monthly boxplot are likely from the years 1980 or 1981.

➢ Empirical Cumulative Distribution Plot

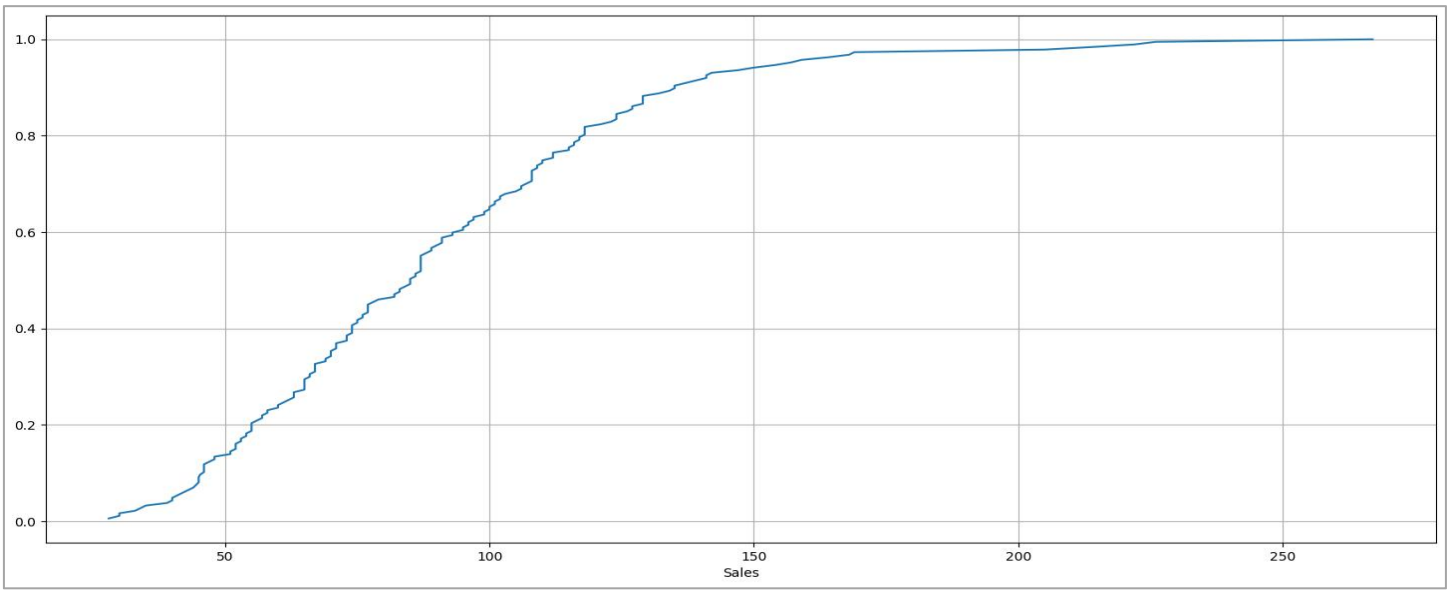

Fig 2.7 Empirical Cumulative Distribution Plot

- This graph tells us what **percentage** of data points refer to what number of Sales.

➢ **Average Rose Sales**

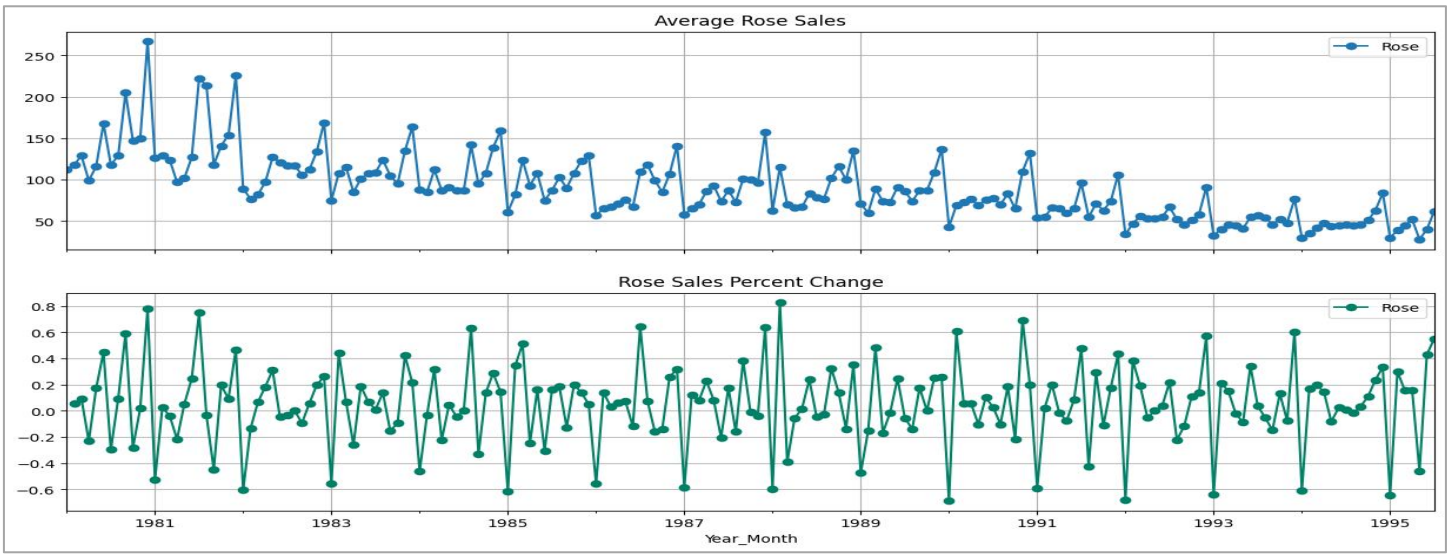

Fig 2.8 Year–on–Year Average Sales

- The **average sales are declining** year–on–year. This is evident from the fact that the line graph is generally decreasing.
- There is a seasonal pattern in sales, with sales being highest in December.

Decomposition:

➢ **Additive Decomposition:**

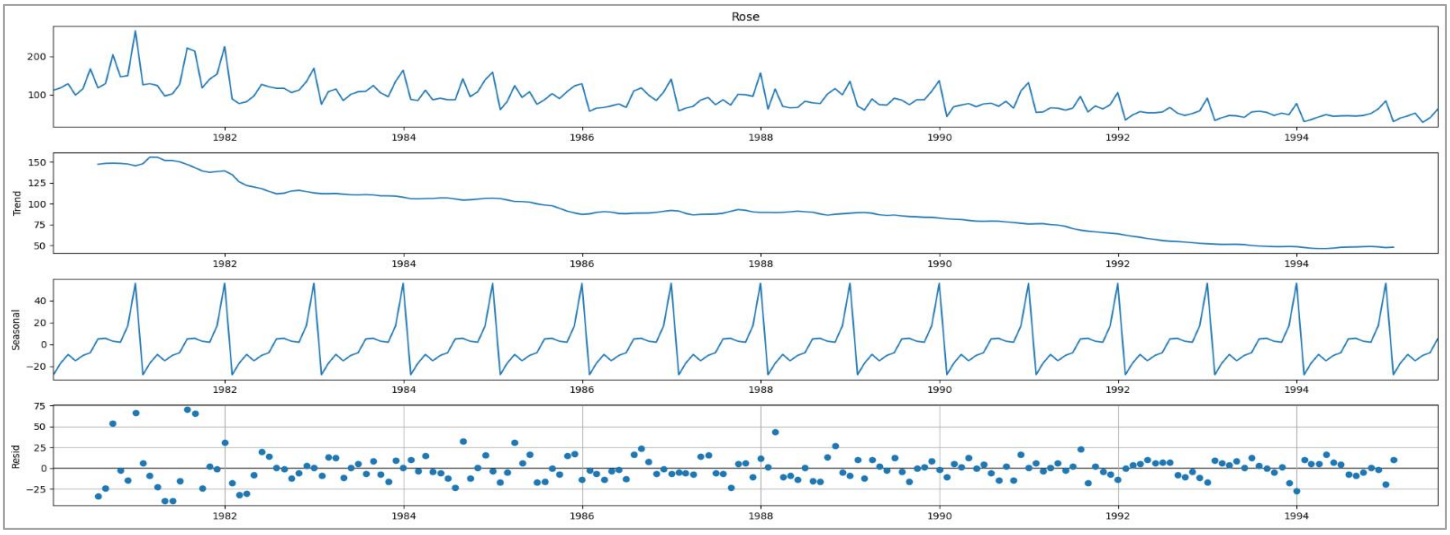

Fig 2.9 Decomposed Time Series– Additive

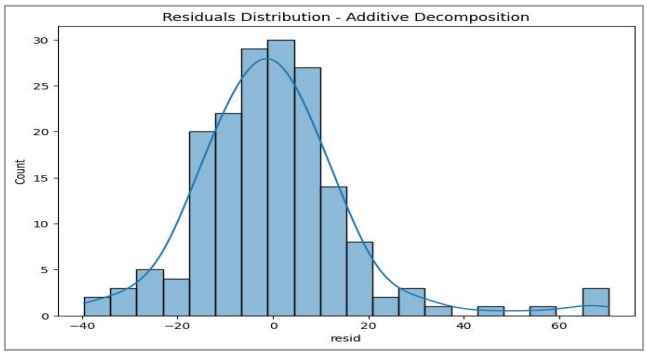

Fig 2.10 Residuals Histogram– Additive Decomposition

- **Test for Normality**

We will use **the Shapiro Wilk Test** for Normality. Let's define the Null & alternate hypothesis: –

Ho: The residuals are normally distributed

Ha: The residuals are not normally distributed

p–value of the Shapiro–Wilk Test on the residuals **= ~7.984797711912961e–09**

Since the p–value < 0.05 – We Reject the null hypothesis.

Hence Residuals are **not normally distributed at 95% confidence level**. The time series is **not an additive** time series.

➢ **Multiplicative Decomposition:**

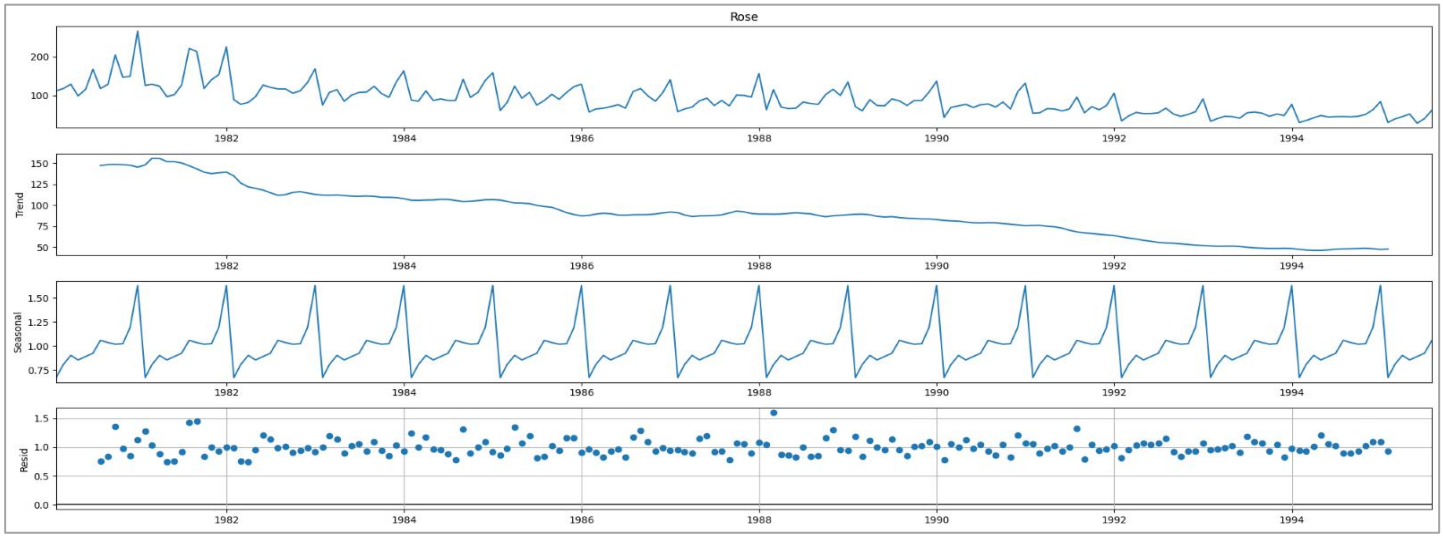

Fig 2.11 Decomposed Time Series– Multiplicative

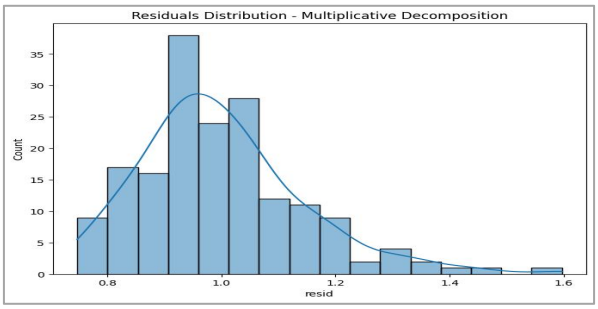

Fig 2.12 Residuals Histogram— Multiplicative Decomposition

- **Test for Normality**

We will use **the Shapiro Wilk Test** for Normality. Let's define the Null & alternate hypothesis: –

Ho: The residuals are normally distributed

Ha: The residuals are not normally distributed

p—value of the Shapiro–Wilk Test on the residuals = **6.26639530310058e–06**

Since the p–value < 0.05: we **fail to reject the Null hypothesis**

Residuals are not **normally distributed at 95% confidence level.** It cannot be determined from this that the time series is a pure multiplicative time series either.

➢ **Time series components for Multiplicative:**

Trend	
Year_Month	
1980-01-31	NaN
1980-02-29	NaN
1980-03-31	NaN
1980-04-30	NaN
1980-05-31	NaN
1980-06-30	NaN
1980-07-31	147.083333
1980-08-31	148.125000
1980-09-30	148.375000
1980-10-31	148.083333
1980-11-30	147.416667
1980-12-31	145.125000
Name: trend, dtype: float64	

Seasonality	
Year_Month	
1980-01-31	0.670198
1980-02-29	0.806223
1980-03-31	0.901304
1980-04-30	0.854185
1980-05-31	0.889558
1980-06-30	0.924126
1980-07-31	1.058081
1980-08-31	1.034408
1980-09-30	1.017778
1980-10-31	1.022716
1980-11-30	1.192528
1980-12-31	1.628895
Name: seasonal, dtype: float64	

Residual	
Year_Month	
1980-01-31	NaN
1980-02-29	NaN
1980-03-31	NaN
1980-04-30	NaN
1980-05-31	NaN
1980-06-30	NaN
1980-07-31	0.758228
1980-08-31	0.841918
1980-09-30	1.357501
1980-10-31	0.970636
1980-11-30	0.853249
1980-12-31	1.129473
Name: resid, dtype: float64	

Table 2.7 Decomposed Time Series Components

- The time series shows a clear **downward trend** across the years, with a sharper **dip observed after 1991 compared to before 1991**.
- **Seasonality** is present in the data, as **sales pick up** in the **ending months of the year.**
- While the time series exhibits characteristics closer to a multiplicative nature, it cannot be definitively classified as either an additive or multiplicative time series

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

- The data was split into a train and test set.
- The splitting was done chronologically, with data from the year **1991** forming the **test set.**
- The **train** set contains **132 records,** while the **test** set contains **55 records.**

```
Dimentions of Original Dataset: (187, 1)
Dimentions of Training data: (132, 1)
Dimentions of Training data: (55, 1)
```

Table 2.8 Dimensions of Original, Train & Test Data

➢ **Training data sample**

First few rows of Train

Year_Month	Rose
1980-01-31	112.0
1980-02-29	118.0
1980-03-31	129.0
1980-04-30	99.0
1980-05-31	116.0

Last few rows of Train

Year_Month	Rose
1990-08-31	70.0
1990-09-30	83.0
1990-10-31	65.0
1990-11-30	110.0
1990-12-31	132.0

Table 2.9 Sample of Training Data

➢ **Test data sample**

First few rows of Test

Year_Month	Rose
1991-01-31	54.0
1991-02-28	55.0
1991-03-31	66.0
1991-04-30	65.0
1991-05-31	60.0

Last few rows of Test

Year_Month	Rose
1995-03-31	45.0
1995-04-30	52.0
1995-05-31	28.0
1995-06-30	40.0
1995-07-31	62.0

Table 2.10 Sample of Test Data

➢ **Train Test Split Plot**

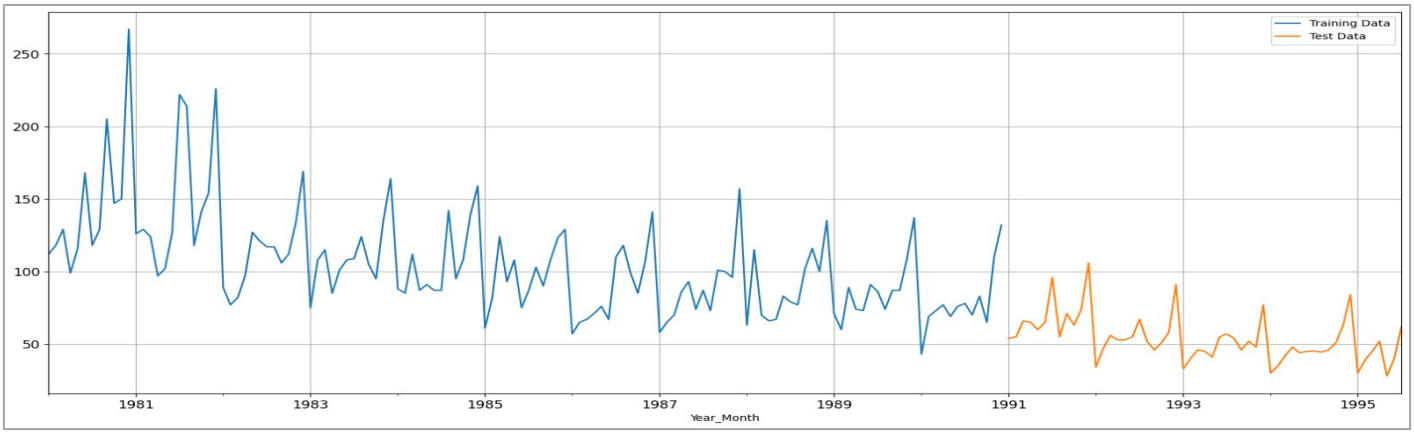

Fig 2.13 Train & Test Split Time Series

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

Building different models and comparing the accuracy metrics.

➢ Linear Regression Model

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

We generated the **numerical time instance order** for both the training and test set. Sample of the Train & Test data

First few rows of Training Data

Year_Month	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

First few rows of Test Data

Year_Month	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

Table 2.11 Sample of LinearRegression Train & Test Data

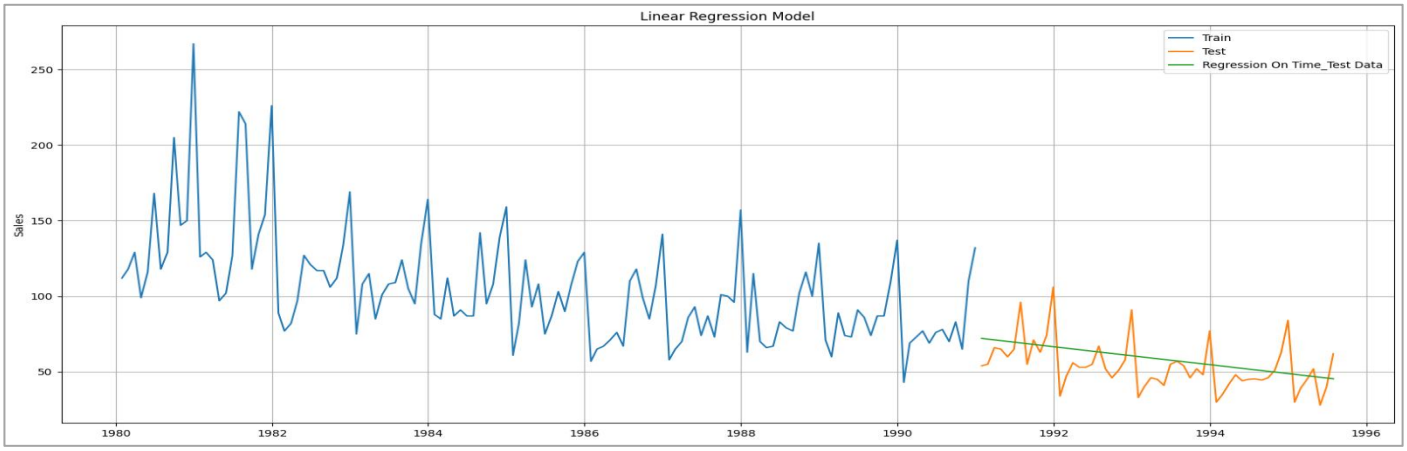

Fig 2.14 Time Series Plot: Linear Regression

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Linear Regression	30.72	15.28	25.01

Table 2.12 Model Performance Summary – Linear Regression

- Linear regression **captures the downward trend** but **not the seasonality.**
- **Test RMSE is 15.28**, **MAPE is 25.01** for Linear Regression, indicating **difficulty** in handling seasonality.

➢ Naïve Forecast Model

The Naive model **predicts tomorrow's value** based on **today's observation,** and the **day after tomorrow's** prediction is also the **same as today's value.**

Samples of Train & test data after we trained on Naive model:

Year_Month	
1980-01-31	132.0
1980-02-29	132.0
1980-03-31	132.0
1980-04-30	132.0
1980-05-31	132.0
Name: naive, dtype: float64	

Year_Month	
1991-01-31	132.0
1991-02-28	132.0
1991-03-31	132.0
1991-04-30	132.0
1991-05-31	132.0
Name: naive, dtype: float64	

Table 2.13 Samples of Train & Test Data – Naive Model

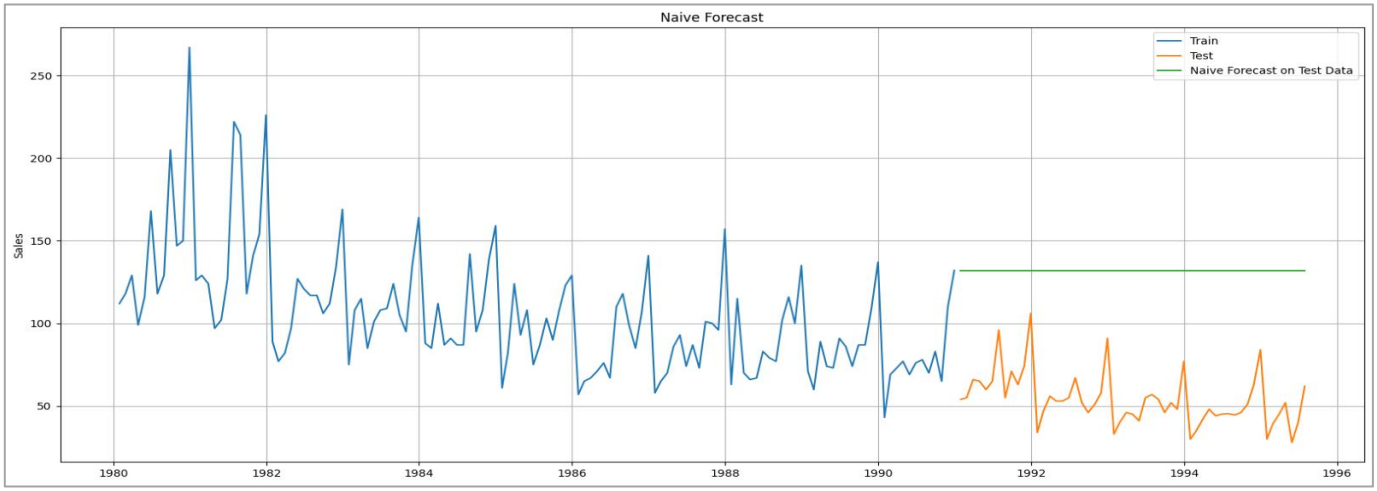

Fig 2.15 Time Series Plot: Naïve

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Naïve Forecast	45.06	79.74	164.99

Table 2.14 Model Performance Summary – Naïve

- **Naive** approach **ignores trend and seasonality** as it forecasts the last observed value
- **Test RMSE is 79.74**, **MAPE is 164.99 for Naive forecast, showing significant errors** due to the lack of trend and seasonality capture.

➢ Simple Average Model:

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

Samples of Train & test data for Simple Average:

Year_Month	Rose	simpleAvg
1980-01-31	112.0	104.939394
1980-02-29	118.0	104.939394
1980-03-31	129.0	104.939394
1980-04-30	99.0	104.939394
1980-05-31	116.0	104.939394

Year_Month	Rose	mean_forecast
1991-01-31	54.0	104.939394
1991-02-28	55.0	104.939394
1991-03-31	66.0	104.939394
1991-04-30	65.0	104.939394
1991-05-31	60.0	104.939394

Table 2.15 Samples of Train & Test Data for Simple Average

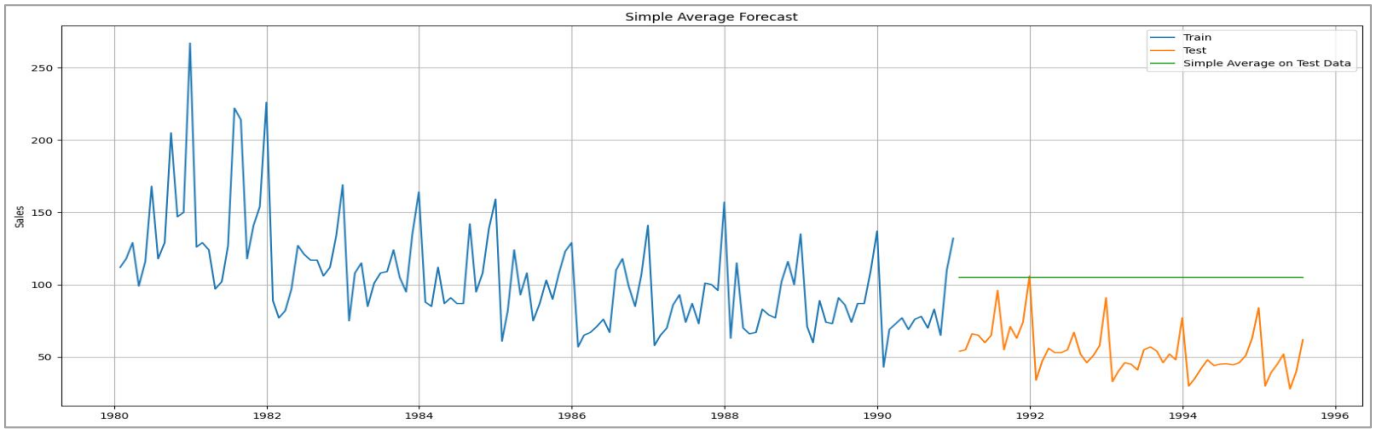

Fig 2.16 Time Series Plot: Simple Average

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Simple Average	36.03	53.49	110.70

Table 2.16 Model Performance Summary – Simple Average

- Simple Average Model **forecasts the mean** of the training data. It **ignores** both the **trend and seasonality.**
- **Test RMSE is 53.49, MAPE is 110.76**. Errors are significant due to the lack of trend and seasonality capture.
- **Performs better than Naïve,** however, is not good enough for predictions

➢ 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) over here.

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

Moving Average Sample on Training data

Year_Month	Rose	Trailing_2	Trailing_4	Trailing_6	Trailing_9
1980-01-31	112.0	NaN	NaN	NaN	NaN
1980-02-29	118.0	115.0	NaN	NaN	NaN
1980-03-31	129.0	123.5	NaN	NaN	NaN
1980-04-30	99.0	114.0	114.50	NaN	NaN
1980-05-31	116.0	107.5	115.50	NaN	NaN
1980-06-30	168.0	142.0	128.00	123.67	NaN
1980-07-31	118.0	143.0	125.25	124.67	NaN
1980-08-31	129.0	123.5	132.75	126.50	NaN
1980-09-30	205.0	167.0	155.00	139.17	132.67
1980-10-31	147.0	176.0	149.75	147.17	136.56

Table 2.17 Moving Average Sample on Training data

Moving Average Sample plot on Whole data

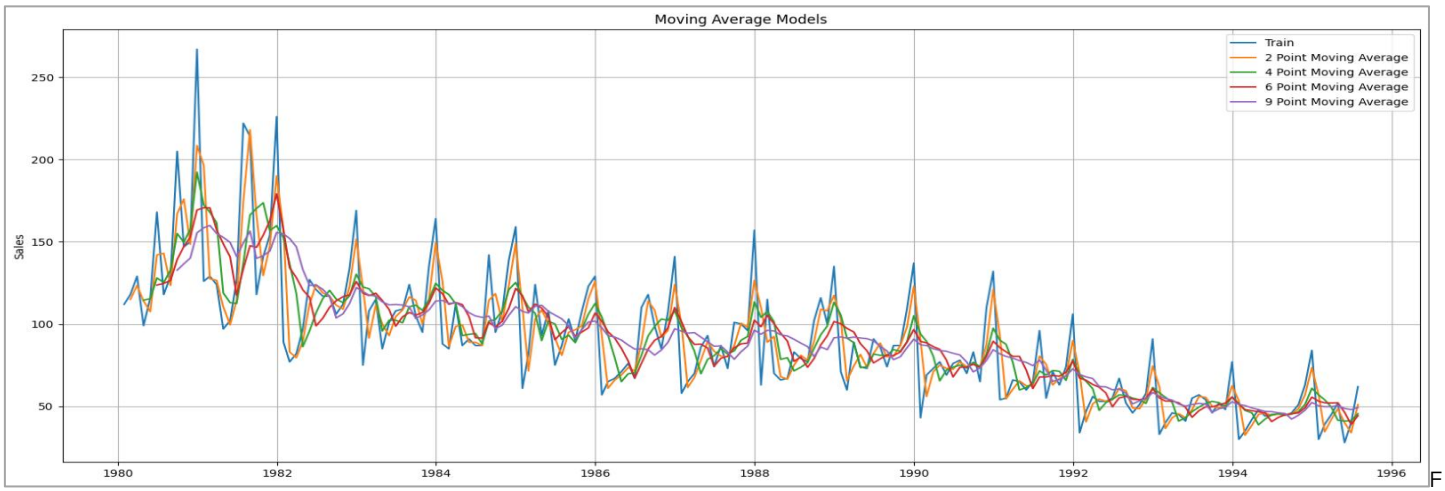

Fig 2.17 Time Series Plot: Moving Average on Whole data

Moving Average Sample plot on Test data

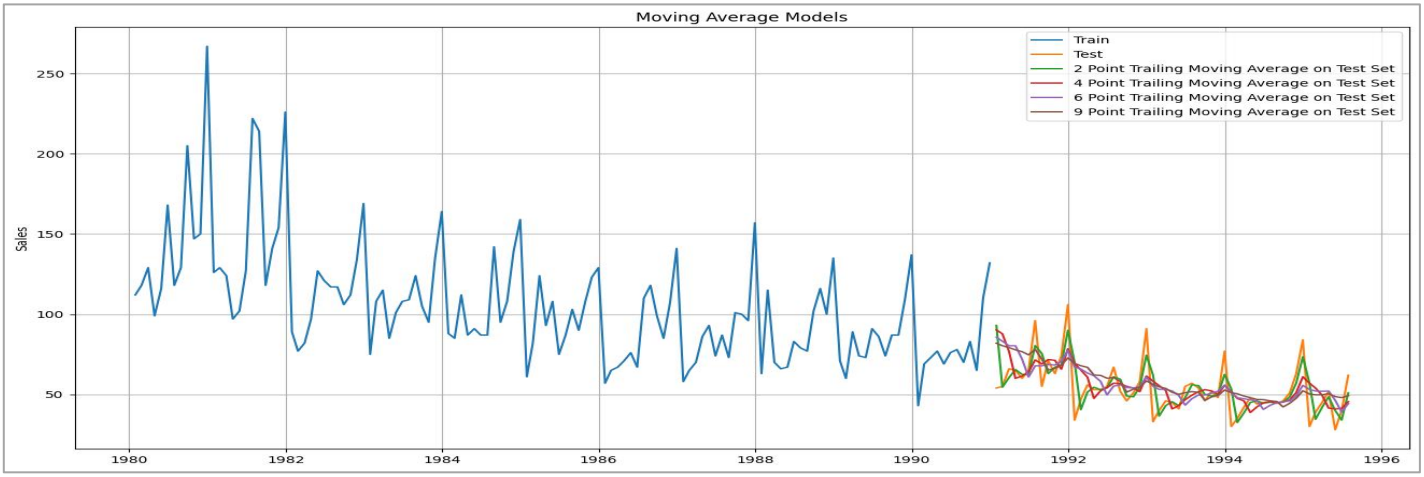

Fig 2.18 Time Series Plot: Moving Average

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
2 Point Moving Average	19.67	11.53	15.73
4 Point Moving Average	26.2	14.46	21.28
6 Point Moving Average	28.52	14.57	22.34
9 Point Moving Average	30.23	14.73	22.69

Table 2.18 Model Performance Summary – Moving Averages

The **2-point Trailing Moving Average** model is selected as it has the **least errors (RMSE & MAPE)** among all the moving average models evaluated.

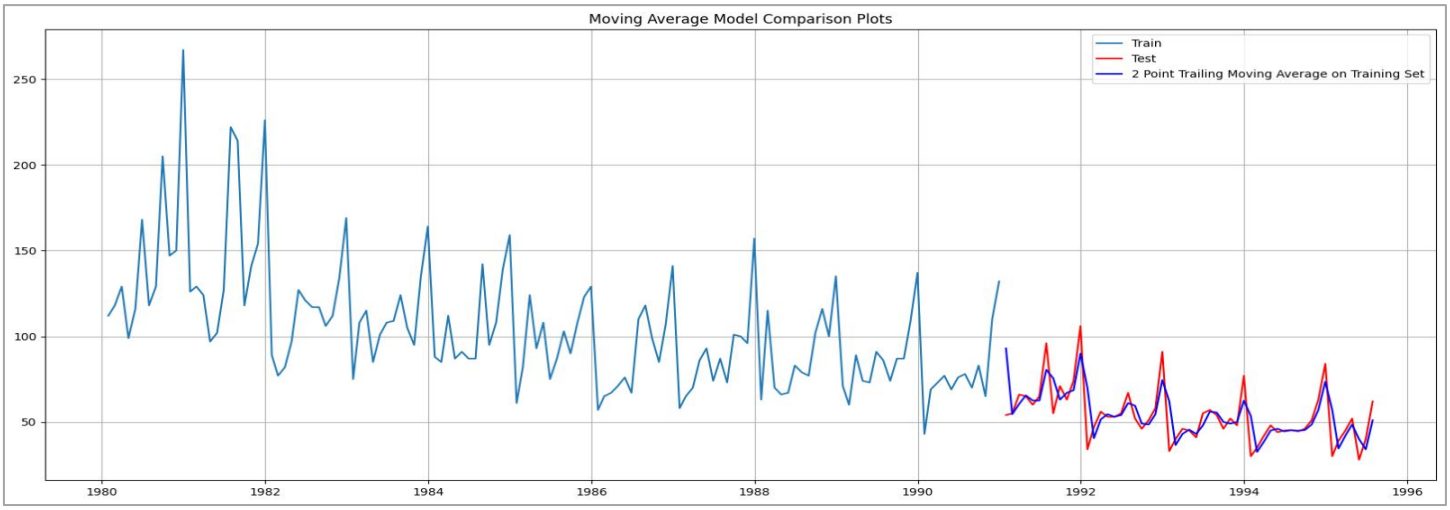

Fig 2.19 Time Series Plot: 2-Point Moving Average

For the moving average models, here are the insights based on their RMSE and MAPE values:

- **2-Point Moving Average: RMSE = 19.67, Test RMSE = 11.53, Test MAPE = 15.73.** It **performs relatively well** in capturing the **trend and seasonality** but still has room for improvement as it struggles to capture the sharp fluctuations in the sales pattern over time.
- **4-Point Moving Average: RMSE = 26.2, Test RMSE = 14.46, Test MAPE = 21.28.** It performs **slightly worse than the 2-Point** model but still shows better accuracy than simpler models like the Simple Average.
- **6-Point Moving Average: RMSE = 28.52, Test RMSE = 14.57, Test MAPE = 22.34.** It did not perform reasonably well and is not as accurate as the 2-Point or 4-Point models.
- **9-Point Moving Average: RMSE = 30.23, Test RMSE = 14.73, Test MAPE = 22.69.** It is again not as good as the 2-Point model.
- Overall, the **2-Point Moving Average** model stands out as the **best-performing model** so far among all with the lowest Test RMSE and Test MAPE values.
- However, there is still room for improvement in all the models to better capture the trend and seasonality and reduce errors in the forecasts.

➢ Simple Exponential Smoothing (SES)

This method is suitable for forecasting data with **no clear trend or seasonal pattern.** It gives more weight to recent observations, which means that recent data points have a stronger influence on the forecast than older ones. This approach allows the model to capture short–term trends and adapt quickly to changes in the data.

Parameter **Alpha (α)** is called the smoothing constant and its value lies between **0 and 1.** Since the model uses only one smoothing constant, it is called Single Exponential Smoothing

▪ SES: Auto Fill Method

The autofit model finds the most optimal parameters according to python while fitting on the train data.

Simple Exponential Smoothing optimal parameters: –

Smoothing Level (alpha) = 0. 099

Initial Level = 134.39

```
{'smoothing_level': 0.09874920899865502,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 134.3871074301239,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Table 2.19 Autofill Simple Exponential Smoothing Optimal Parameters

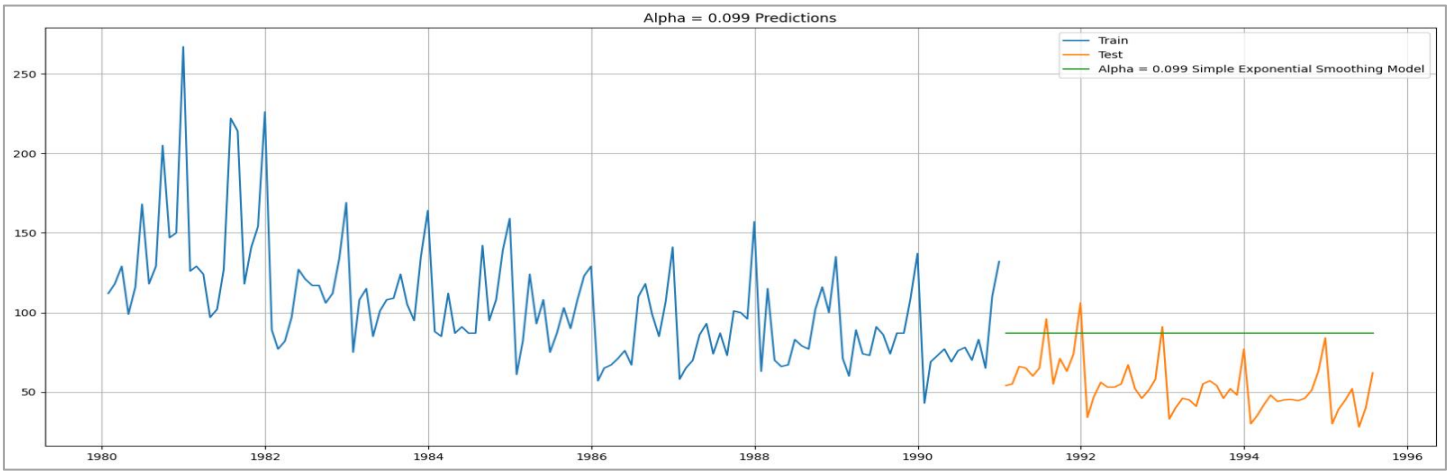

Fig 2.20 Time Series Plot: Simple Exponential Smoothing Alpha = 0.099

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Simple Exponential Smoothing Alpha = 0.099	31.50	36.82	76.00

Table 2.20 Model Performance Summary – Simple Exponential Smoothing Alpha = 0.099

- **The Autofill Simple exponential smoothing** model provides one-step-ahead forecast. It **ignores** both the **trend and seasonality** in the data.
- Test **RMSE is 36.82** and **MAPE is 76.00,** indicating **poor performance** in capturing underlying patterns.
- The low smoothing parameter **(0.099)** implies a heavy reliance on past averages. This makes it less accurate compared to more sophisticated methods.

- **SES: Brute Force Method**

The brute force model **tests various smoothing parameter** values to find the best ones for accurate test data forecasting. Below is the table for various parameters, **sorted** with **least Test RMSE** on top.

	Alpha	Train RMSE	Test RMSE
6	0.07	32.649443	36.460430
7	0.08	32.477045	36.488207
5	0.06	32.880735	36.602736
8	0.09	32.348486	36.629498
9	0.10	32.253385	36.853442

Table 2.21 Brute Force Simple Exponential Smoothing Parameters

Best **Alpha** for Simple Exponential Smoothing: **0.07**

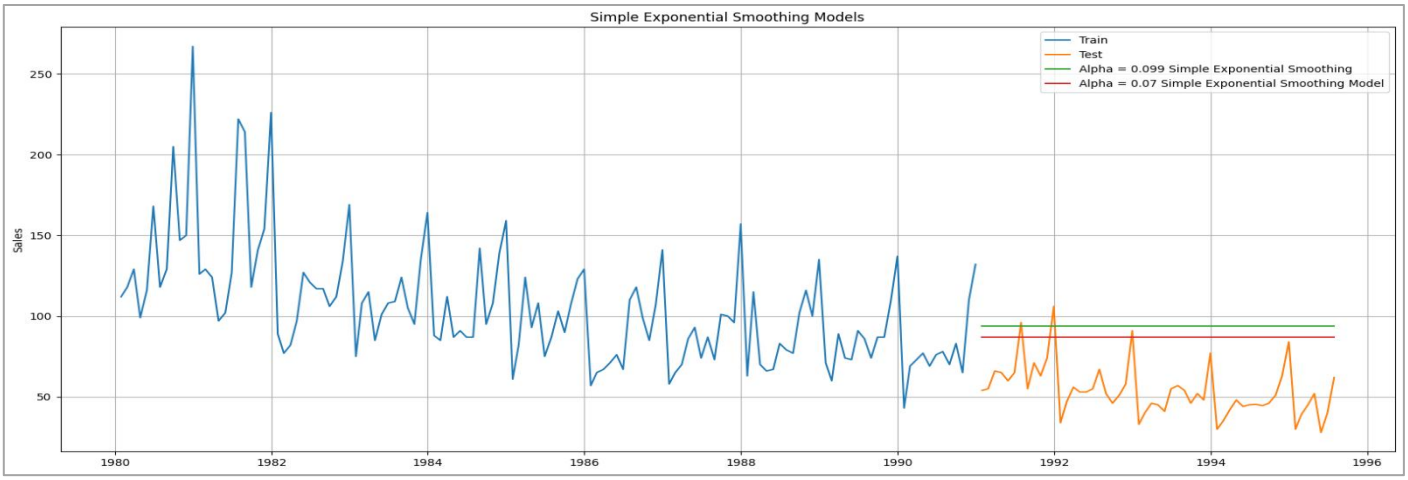

Fig 2.21 Time Series Plot: Simple Exponential Smoothing Alpha = 0.07

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Simple Exponential Smoothing Alpha = 0.07	32.65	36.46	88.74

Table 2.22 Model Performance Summary – Simple Exponential Smoothing Alpha = 0.07

- The **Brute Force simple exponential smoothing** model generates one-step-ahead forecasts, **overlooking both trend and seasonality** in the time series.
- The model exhibits an **RMSE of 36.46** and **MAPE of 88.74**, signifying its **inability** to effectively capture the underlying **trend and seasonal patterns.**
- The small smoothing parameter **(0.07)** indicates that the **model relies** heavily on **past data averages** rather than recent observations, making its accuracy comparable to the simple average model.

➢ Double Exponential Smoothing (DES)

This method is applicable when data has Trend but no seasonality.

▪ DES: Auto Fill Method

The autofit model finds the most optimal parameters according to python while fitting on the train data.

Double Exponential Smoothing optimal parameters: –

Smoothing Level (Alpha) = 1.49*10^–8

Smoothing Trend (beta) = 5.44*10^–9

```
Holt model Exponential Smoothing Estimated Parameters

{'smoothing_level': 1.4901161193847656e-08, 'smoothing_trend': 5.448169774560283e-09, 'smoothing_seasonal': nan,
```

Table 2.23 Autofill Double Exponential Smoothing Optimal Parameters

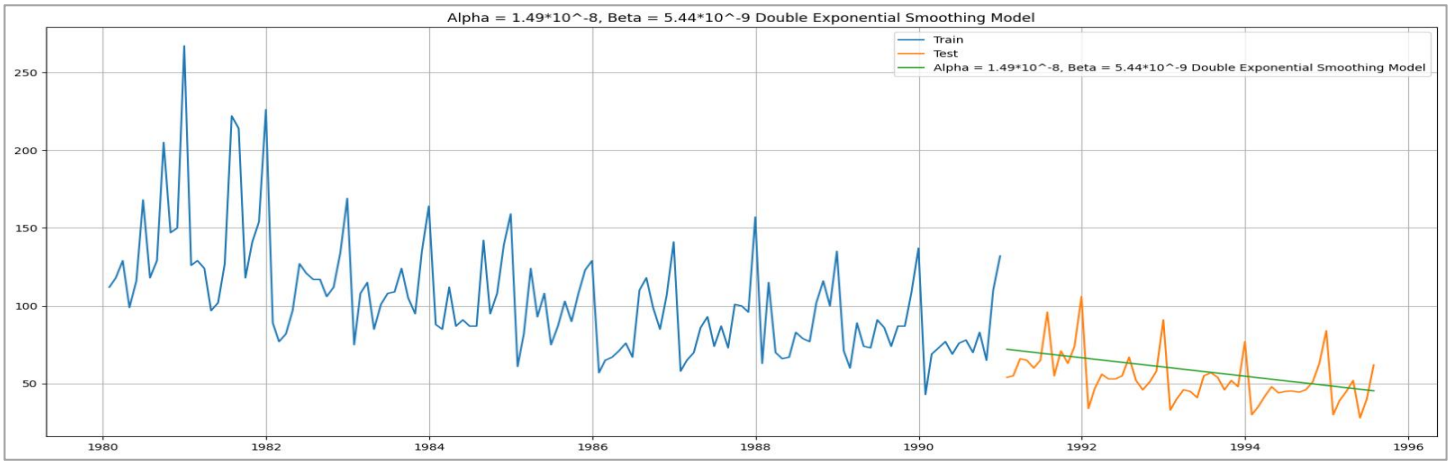

Fig 2.22 Time Series Plot: Double Exponential Smoothing Alpha = 1.49*10^–8, Beta = 5.44*10^–9

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Double Exponential Smoothing Alpha = 1.49*10^–8, Beta = 5.44*10^–9	30.72	15.28	30.14

Table 2.24 Model Performance Summary – Double Exponential Smoothing Alpha = 1.49*10^–8, Beta = 5.44*10^–9

- The **Autofill double exponential smoothing** model forecasts the **trend** but **disregards** the **seasonality** of the time series.
- The **RMSE is 15.28**, and the **MAPE is 30.14** for the Double exponential smoothing model. While the error is comparable to the Linear Regression Model
- **Alpha & beta close to 0** means it has used the **whole historical data to forecast** the time series.

- **DES: Brute Force Method**

The brute force model **tests various smoothing parameter** values to find the best ones for accurate test data forecasting. Below is the table for various parameters, **sorted** with **least Test RMSE** on top.

	Alpha	Beta	Train RMSE	Test RMSE
343	0.04	0.47	39.202418	14.459497
222	0.03	0.25	46.040023	15.021584
223	0.03	0.26	45.711843	15.231184
262	0.03	0.65	41.690382	15.330784
398	0.05	0.03	59.667569	15.363394

Table 2.25 Brute Force Double Exponential Smoothing Parameters

Since **Alpha = 0.04 & Beta = 0.47** yield the **least test RMSE**, indicating the best fit for our test data, we select them to build our double exponential smoothing model.

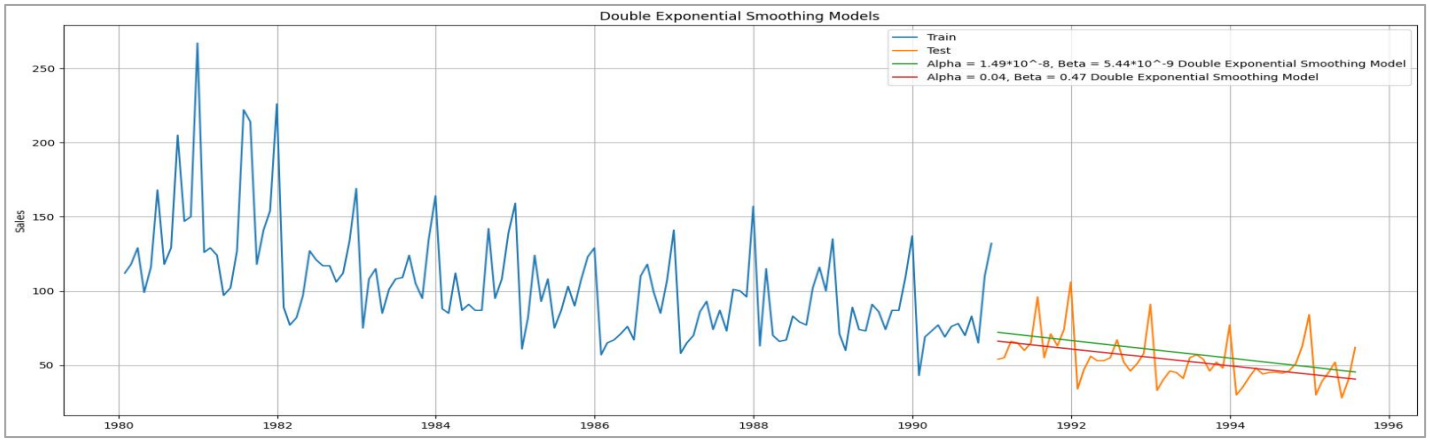

Fig 2.23 Time Series Plot: Double Exponential Smoothing Alpha = 0.04 & Beta = 0.47

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Double Exponential Smoothing Alpha = 0.04 & Beta = 0.47	39.20	14.46	34.86

Table 2.26 Model Performance Summary – Double Exponential Smoothing Alpha = 0.04, Beta= 0.47

- The **Brute Force double exponential smoothing** model effectively **captures the trend** but **overlooks the seasonality** of the time series.

- This performs slightly **better than auto fill method** (Double Exponential Smoothing **Alpha** = 1.49*10^−8, **Beta** = 5.44*10^−9
- The model **exhibits RMSE** of **14.46** and **MAPE of 34.86.**

➢ **Triple Exponential Smoothing (TES)**

- The triple exponential smoothing model is suitable for time series with both **trend and seasonality**.
- Since the trend variation is linear and the seasonal decomposition suggests a multiplicative time series, we use a triple exponential smoothing model with additive trend and multiplicative seasonality.

▪ **TES: Auto Fill Method**

The autofit model finds the most optimal parameters according to python while fitting on the train data.

Triple Exponential Smoothing optimal parameters: –

Smoothing Level (**Alpha**) = **0.077**

Smoothing Trend (**Beta**) = **0.039**

Smoothing Seasonal (**Gamma**) = **0.0008**

```
Holt Winters model Exponential Smoothing Estimated Parameters

{'smoothing_level': 0.07736040004765096, 'smoothing_trend': 0.03936496779735522, 'smoothing_seasonal': 0.0008375039104357999,
```

Table 2.27 Autofill Triple Exponential Smoothing Parameters

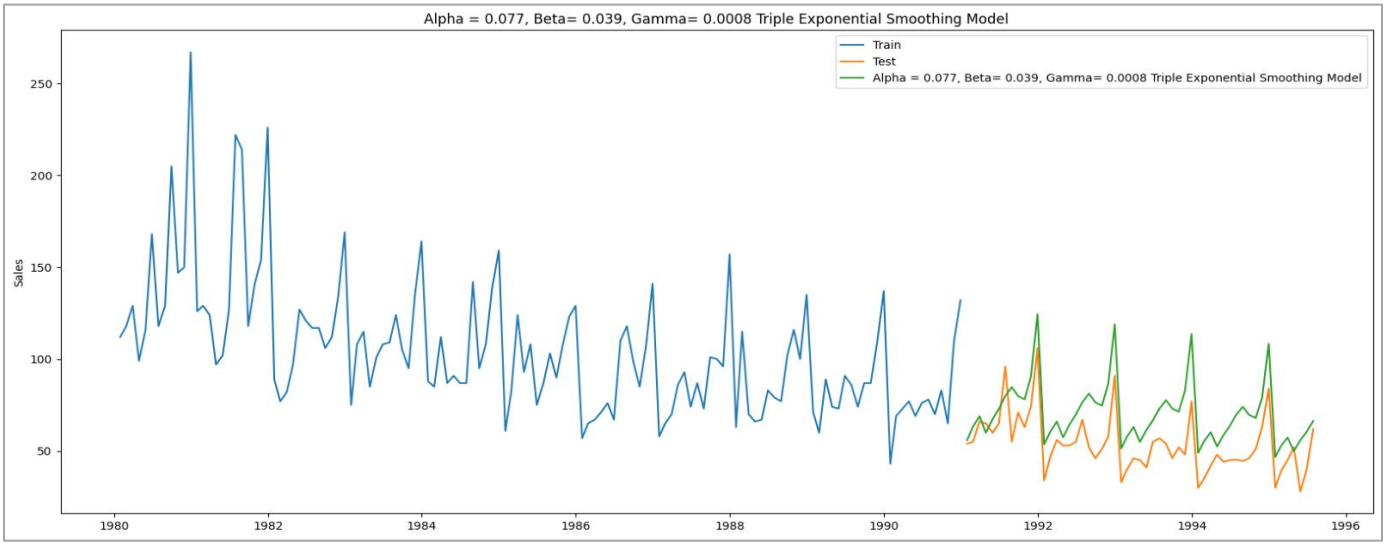

Fig 2.24 Time Series Plot: Triple Exponential Smoothing Alpha = 0.077, Beta= 0.039, Gamma= 0.0008

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Triple Exponential Smoothing Alpha = 0.077, Beta= 0.039, Gamma= 0.0008	18.41	19.15	47.83

Table 2.28 Model Performance Summary – Triple Exponential Smoothing Alpha = 0.077, Beta= 0.039, Gamma= 0.0008

- The **Autofill triple Exponential Smoothing** model **captures** both the **trend and seasonality** of the time series.
- The test **RMSE** is 19.15**,** and the **MAPE is 47.83**. However, the error is higher than Double exponential Smoothening or LinearRegression model
- The values of **Alpha, Beta and Gamma** being **close to 0**, indicate that the **forecast relies** heavily on **historical data** to build the model giving a smoother prediction.

- **TES: Brute Force Method**

The brute force model **tests various smoothing parameter** values to find the best ones for accurate test data forecasting. Below is the table for various parameters, **sorted** with **least Test RMSE** on top.

	Alpha	Beta	Gamma	Train RMSE	Test RMSE
1653	0.04	0.52	0.10	21.687974	8.233321
106	0.01	0.10	0.22	28.206290	8.284979
1654	0.04	0.52	0.13	21.655908	8.435349
72	0.01	0.07	0.19	30.190955	8.451800
1625	0.04	0.49	0.25	21.740051	8.473643

Table 2.29 Brute Force Triple Exponential Smoothing Parameters

Since **Alpha = 0.04, Beta = 0.52, Gamma = 0.10** yield the **least test RMSE**, indicating the best fit for our test data, we select them to build our Triple Exponential Smoothing model.

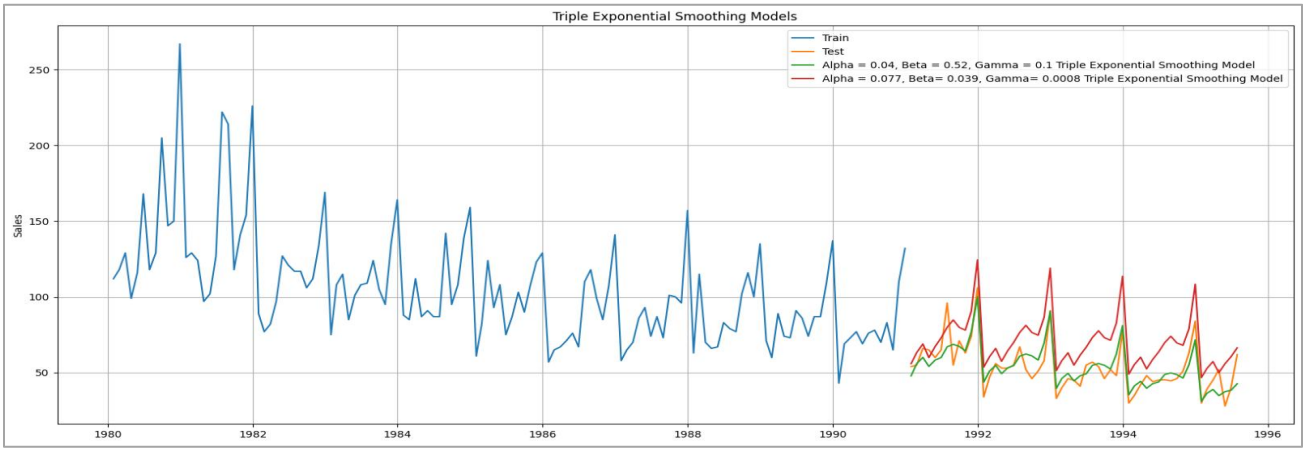

Fig 2.25 Time Series Plot: Triple Exponential Smoothing Alpha = 0.04, Beta = 0.52, Gamma = 0.10

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Triple Exponential Smoothing Alpha = 0.04, Beta = 0.52, Gamma = 0.10	21.69	8.22	30.67

Table 2.30 Model Performance Summary – Triple Exponential Smoothing Alpha = 0.04, Beta = 0.52, Gamma = 0.10

- The **Brute Force Triple Exponential smoothing** model **captures** both the **trend and seasonality.**
- With **RMSE of 8.23** and **MAPE** of **30.73,** this model exhibits the **best accuracy among all the evaluated models so far.**
- The values of **alpha** and **beta** being **close to 0** imply that the model heavily **relies** on **historical data** to make forecasts.

➢ **Plotting the prediction of all the Models built so far**

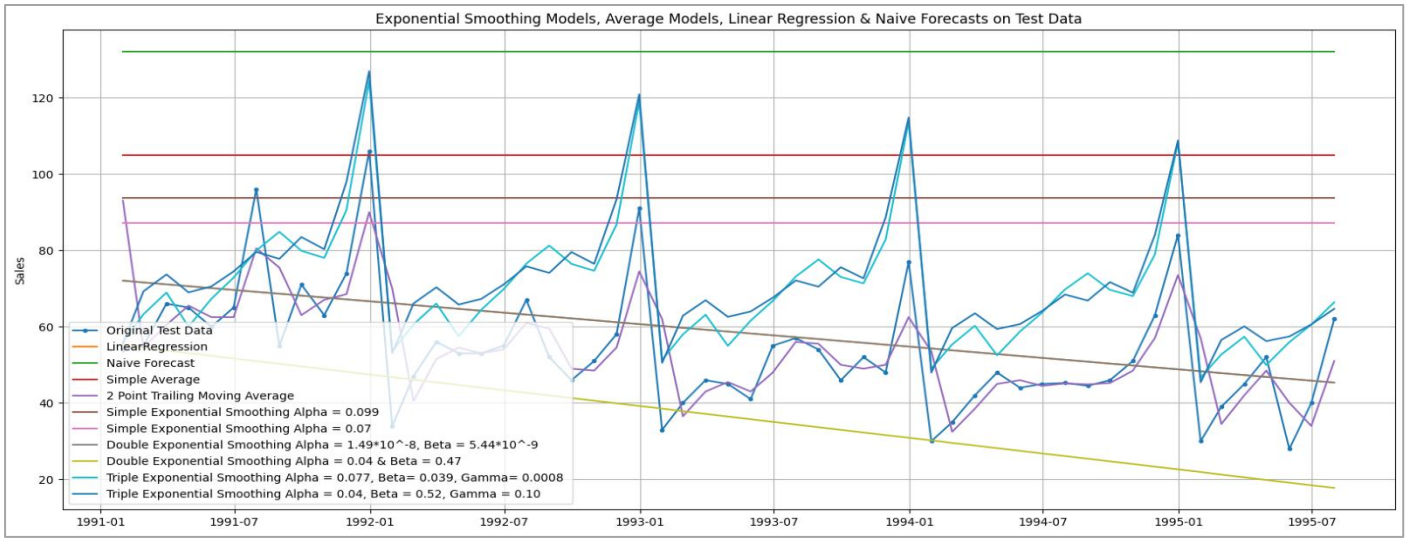

Fig 2.26 Time Series Plot: Model Comparisions

- The **Brute Force triple exponential smoothing** models show the **best accuracy** among all models evaluated, with the lowest Root Mean Square Error **(RMSE) of 8.22** and Mean Absolute Percentage Error (**MAPE**) of **30.67**
- This is followed by the **2-point moving average** model, capturing both trend and seasonality in the time series.
- The rest of the other models are also **not suitable for prediction** as they **do not capture** both the **Trend & Seasonality well** required for the time series.

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.

➢ Stationarity of the whole Time Series data

The **Augmented Dickey-Fuller (ADF) Test** is used to check the stationarity of a time series. The test formulates the following null and alternative hypotheses:

Null Hypothesis (Ho): The time series is non-stationary.

Alternative Hypothesis (Ha): The time series is stationary.

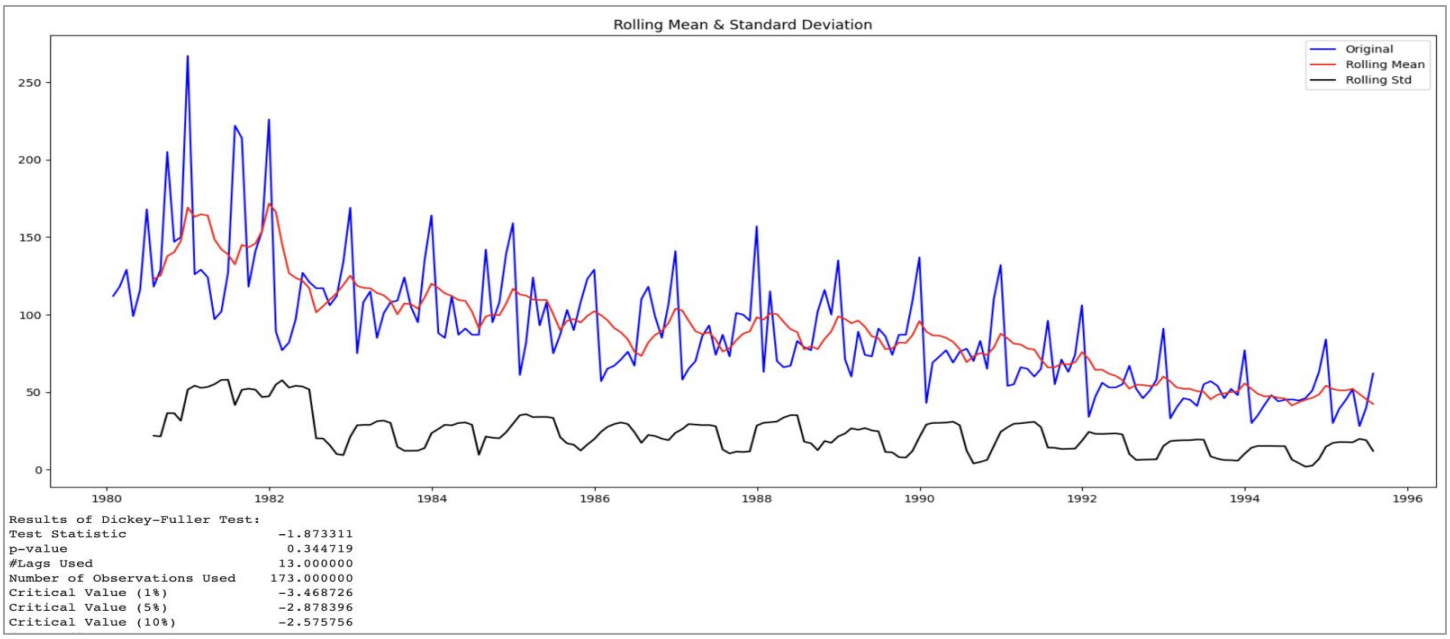

Fig 2.27 Stationarity of Whole Data Using AD Fuller Test

- We see that at **5% significant level** the Time Series is **non-stationary.**
- Let us take a **difference of order 1** and check whether the Time Series is stationary or not

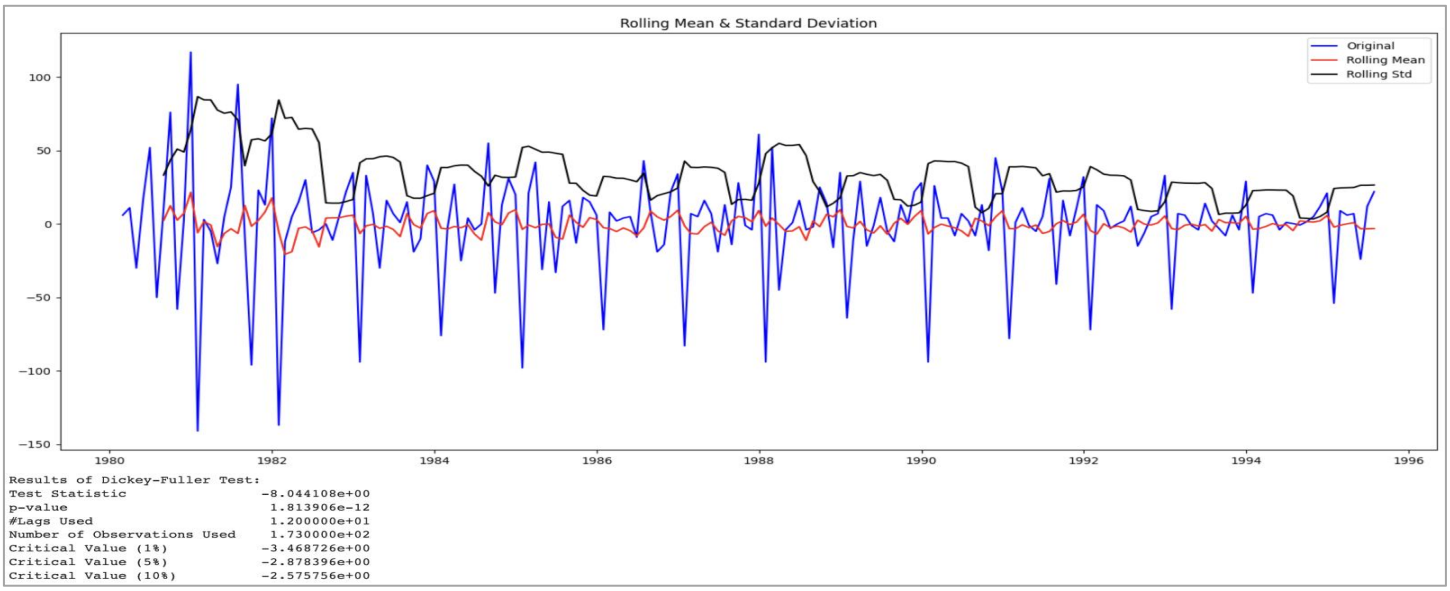

Fig 2.28 Stationarity of Whole Data Using AD Fuller Test at Differencing of Order 1

- At **difference of order 1,** the series have become **stationary at ** = 0.05.**

➢ Stationarity of the Training Data Time Series

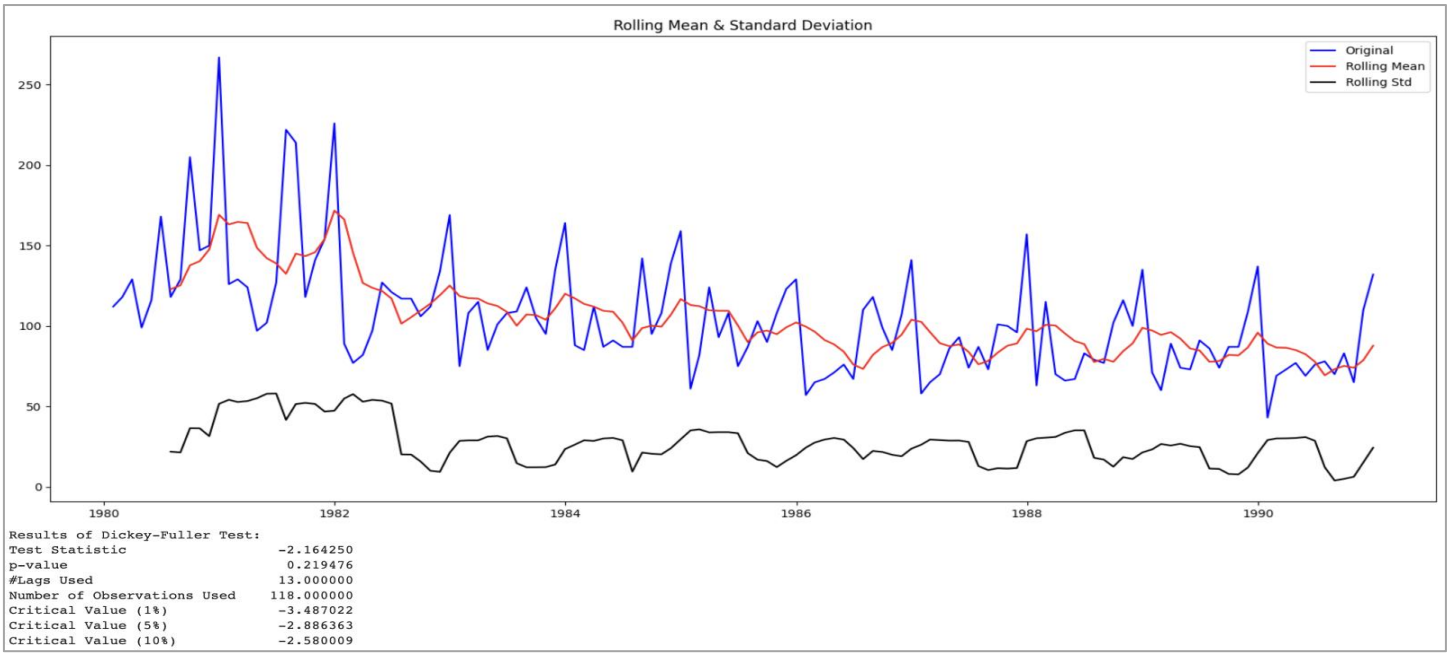

Fig 2.29 Stationarity of Training Data Using AD Fuller Test

- At **5% significant level** the Time Series is **non–stationary.**
- Taking First order difference using diff function

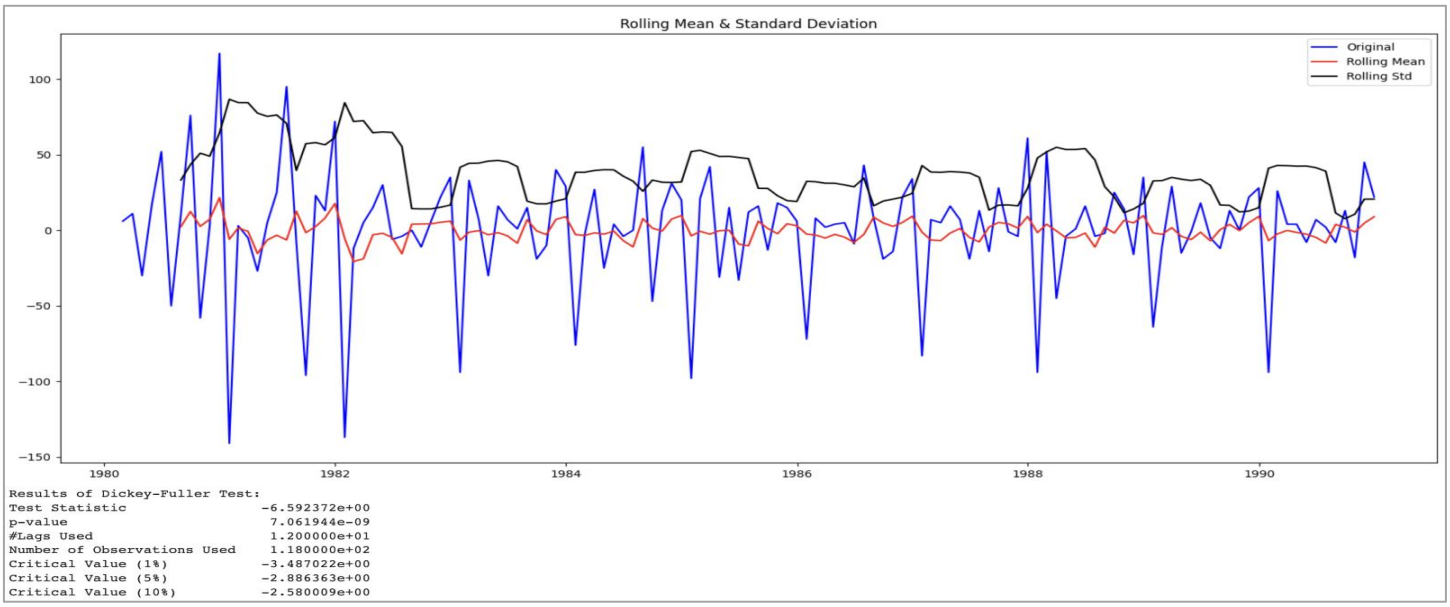

Fig 2.30 Stationarity of Training Data Using AD Fuller Test at Differencing of Order 1

- We see that at **difference of order 1**, the series have become **stationary at ** = 0.05.**

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.

➢ Autocorrelation Plots

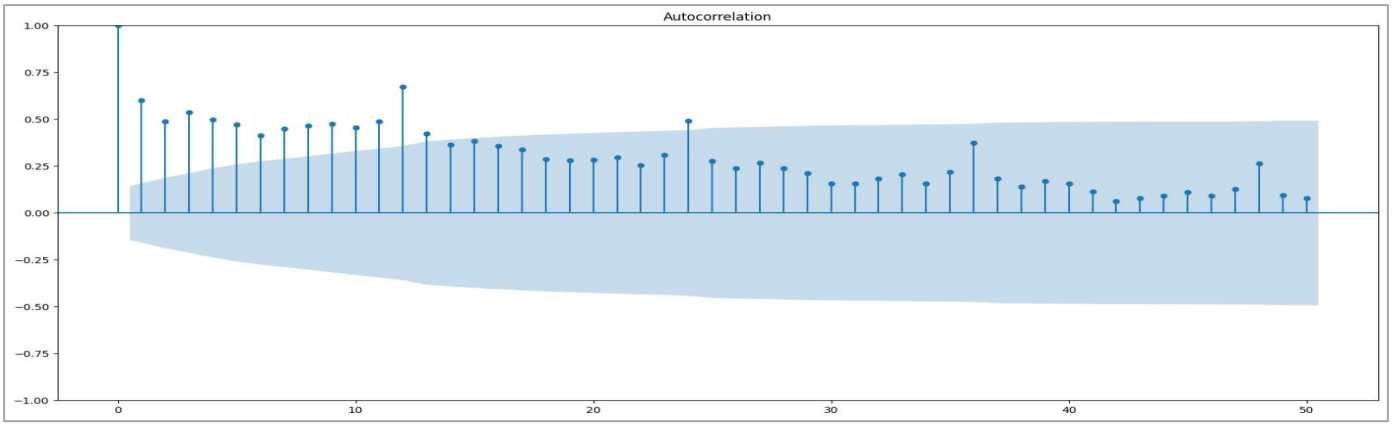

Fig 2.31 Autocorrelation Plot

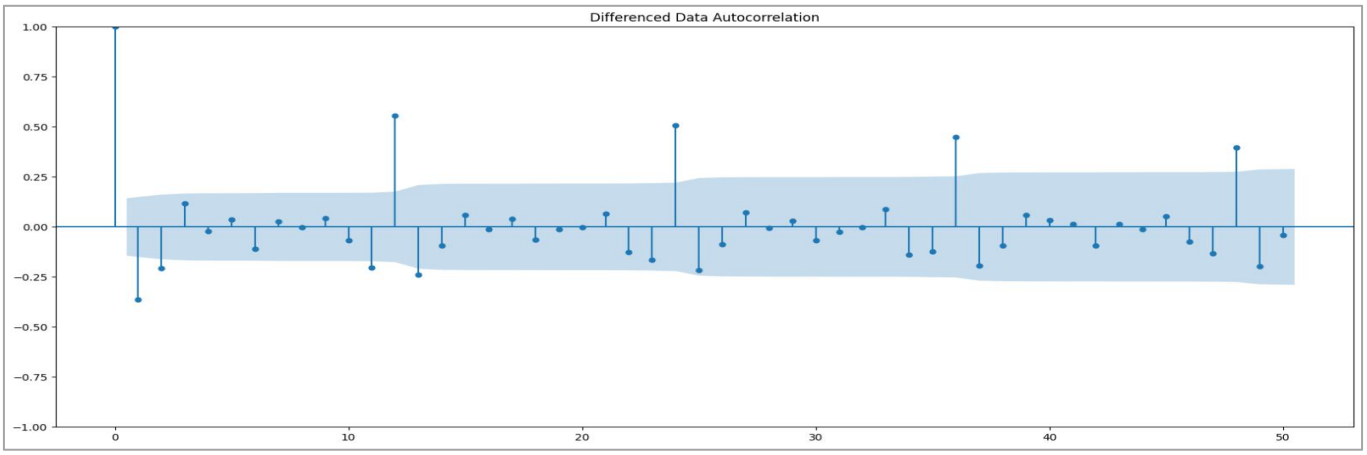

Fig 2.32 Differenced Autocorrelation Plot

- From the above plots, we can say that there is **seasonality** in the data. This would be more useful when building SARIMA model

➢ Automated version of an ARIMA model

ARIMA: – Auto Regressive Integrated Moving Average is a way of modeling time series data for **forecasting or predicting future data points**. Improving AR Models by making Time Series stationary through **Moving Average Forecasts**

ARIMA models consist of 3 components: –

AR model: The data is modelled based on past observations.

Integrated component: Whether the data needs to be differenced/transformed.

MA model: Previous forecast errors are incorporated into the model.

The best parameters are selected in accordance with the lowest Akaike Information Criteria (AIC).

- **ARIMA Model building to estimate best 'p', 'd', 'q' parameters (Lowest AIC Approach)**

	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 2.31 ARIMA AIC Parameters

- For p = 0, d = 1, q = 2 , we obtained the **lowest AIC** value, we used it to build our ARIMA model.

ARIMA Results

SARIMAX Results

Dep. Variable:	Rose	No. Observations:	132
Model:	ARIMA(0, 1, 2)	Log Likelihood	-636.836
Date:	Wed, 02 Aug 2023	AIC	1279.672
Time:	11:09:40	BIC	1288.297
Sample:	01-31-1980	HQIC	1283.176
	- 12-31-1990		
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 2.32 Auto ARIMA Model Summary

Diagnostic Plot

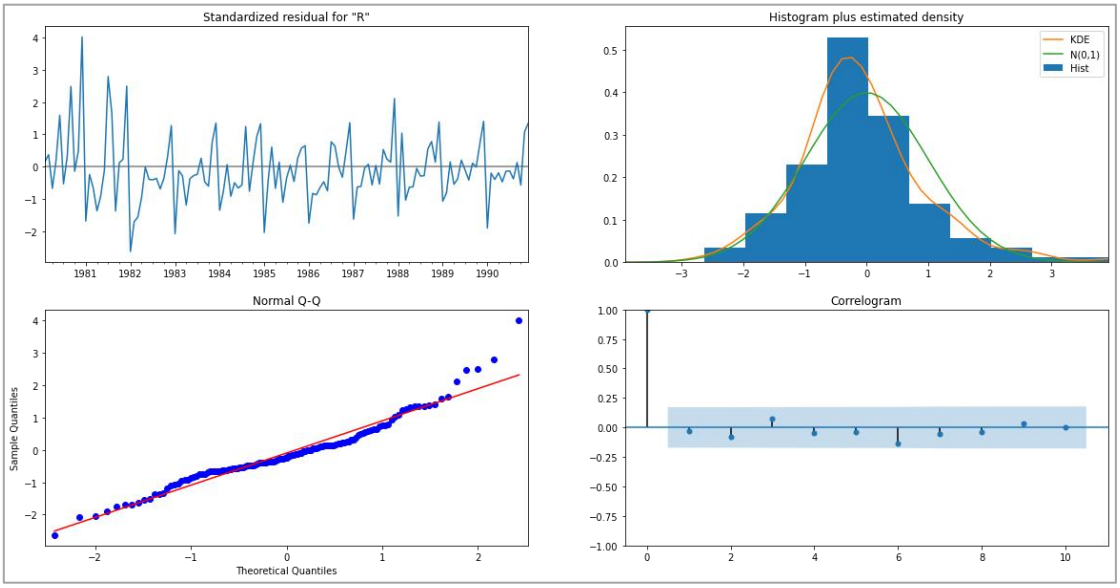

Fig 2.33 Diagnostic Plot: Automated ARIMA (0, 1, 2)

Forecast on the test data

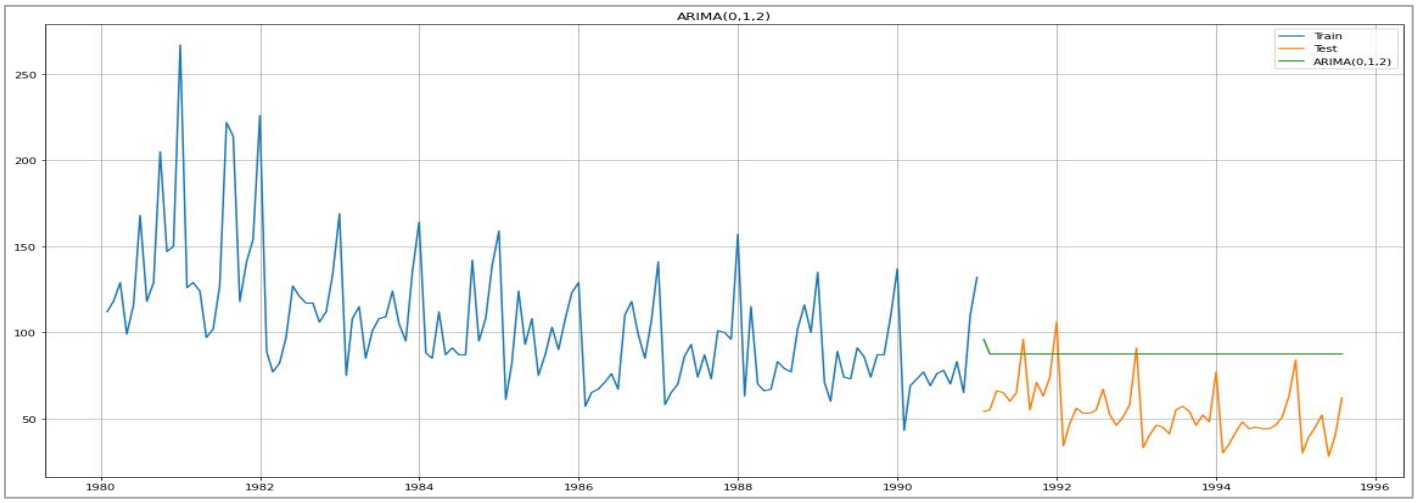

Fig 2.34 Time Series Plot: Automated ARIMA (0, 1, 2)

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Automated ARIMA (0, 1, 2)	31.25	37.37	77.18

Table 2.33 Model Performance Summary – Automated ARIMA (0, 1, 2)

- The **Auto ARIMA model** aims to **capture** the **underlying trend** in the data but **does not consider the seasonality component.**
- The model's performance is evaluated with a **Root Mean Square Error of 37.37** and a **Mean Absolute Percentage Error of 77.18.** The model performed poorer on Test data as compared to Training

➢ Automated version of a SARIMA model

The ARIMA models can be extended/improved to handle seasonal components of a data series.

The seasonal autoregressive moving average model is given by **SARIMA (p, d, q)(P, D, Q)F**

The above model consists of:

- **Autoregressive** and **moving average components (p, q)**
- **Seasonal autoregressive** and **moving average components (P, Q)**
- The **ordinary** and s**easonal difference components** of order **'d' and 'D'**
- **Seasonal frequency 'F'**

The value for the parameters **(p,d,q) and (P, D, Q)** can be decided by comparing different values for each and taking the lowest AIC value for the model build.

The value for **F** can be consolidated by **ACF plot**

- **Without Seasonal Differencing (D = 0):**

Let us look at the differenced ACF plot again to understand the seasonal parameter for the SARIMA model

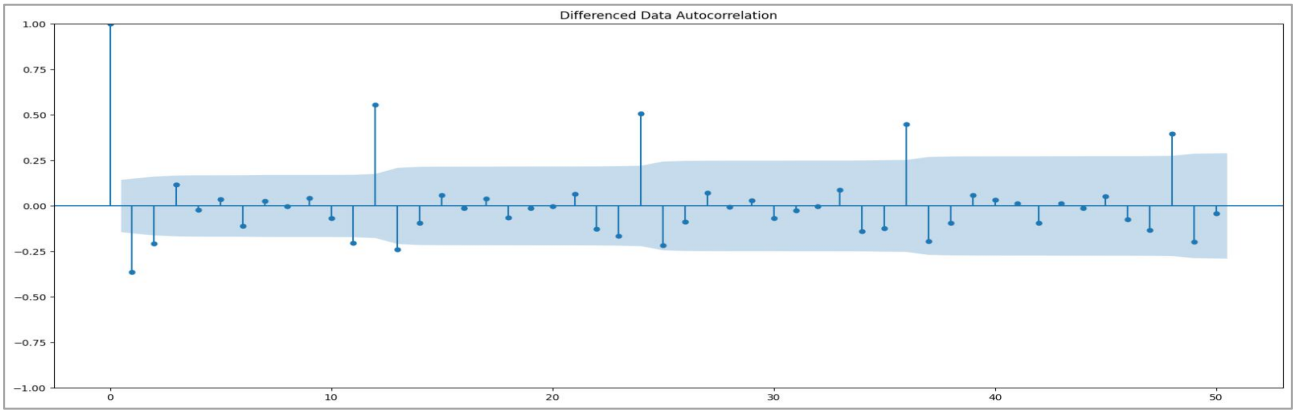

Fig 2.35 Differenced Autocorrelation Plot: SARIMA

- **S=12** is chosen for seasonal differencing as it is **significant**, and the **ACF plot at S=12 does not taper off.**
- This indicates the **presence of seasonality**, and applying seasonal differencing to the original series can improve the model's performance.
- **d = 1** to make the time series stationary
- Seasonal differencing **not yet applied** to make the time series stationary **D = 0**

To find the values of p, q, P & Q, we iterated values between and computed AIC values for each combination. Lowest 5 shown below. We built the model on the values: p = 0, d =1, q = 2, P = 2, D = 0, Q = 2, S = 12

	param	seasonal	AIC
26	(0, 1, 2)	(2, 0, 2, 12)	887.937509
53	(1, 1, 2)	(2, 0, 2, 12)	889.901198
80	(2, 1, 2)	(2, 0, 2, 12)	890.668798
69	(2, 1, 1)	(2, 0, 0, 12)	896.518161
78	(2, 1, 2)	(2, 0, 0, 12)	897.346445

Table 2.34 SARIMA AIC Parameters without Seasoning

SARIMA Results

SARIMAX Results

Dep. Variable:	Rose	No. Observations:	132
Model:	SARIMAX(0, 1, 2)x(2, 0, 2, 12)	Log Likelihood	-436.969
Date:	Mon, 28 Nov 2022	AIC	887.938
Time:	12:35:33	BIC	906.448
Sample:	01-31-1980	HQIC	895.437
	- 12-31-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.8427	189.867	-0.004	0.996	-372.976	371.290
ma.L2	-0.1573	29.829	-0.005	0.996	-58.621	58.307
ar.S.L12	0.3467	0.079	4.375	0.000	0.191	0.502
ar.S.L24	0.3023	0.076	3.996	0.000	0.154	0.451
ma.S.L12	0.0767	0.133	0.577	0.564	-0.184	0.337
ma.S.L24	-0.0726	0.146	-0.498	0.618	-0.358	0.213
sigma2	251.3137	4.77e+04	0.005	0.996	-9.33e+04	9.38e+04

Ljung-Box (L1) (Q):	0.10	Jarque-Bera (JB):	2.33
Prob(Q):	0.75	Prob(JB):	0.31
Heteroskedasticity (H):	0.88	Skew:	0.37
Prob(H) (two-sided):	0.70	Kurtosis:	3.03

Table 2.35 Auto SARIMA without Differencing Model Summary

Diagnostic Plot

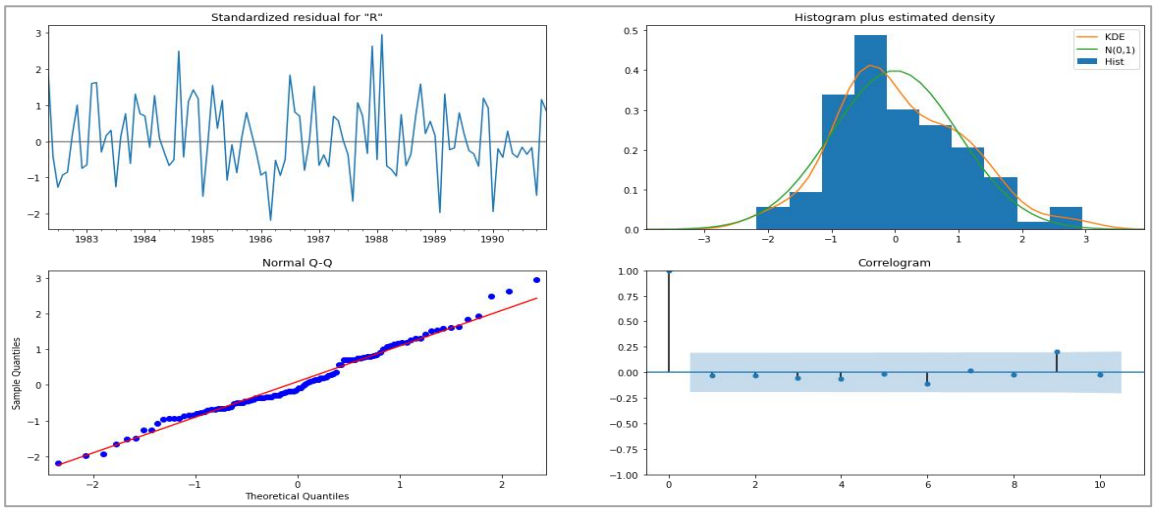

Fig 2.36 Diagnostic Plot: Automated SARIMA (0, 1, 2)(2, 0, 2, 12)

Forecast on the test data

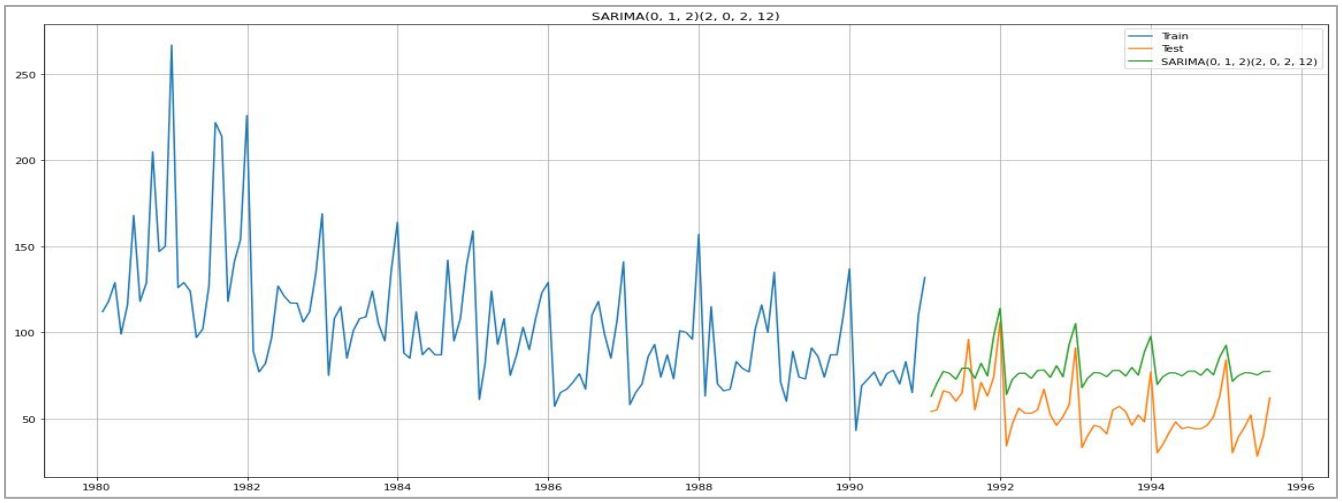

Fig 2.37 Time Series Plot: Automated SARIMA (0, 1, 2)(2, 0, 2, 12)

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Automated SARIMA (0, 1, 2)(2, 0, 2, 12)	32.23	26.95	59.95

Table 2.36 Model Performance Summary – Automated SARIMA (0, 1, 2)(2, 0, 2, 12)

- The **Automated SARIMA (0, 1, 2)(2, 0, 2, 12) aims** to **capture** the **underlying trend** in the data as well as **the seasonality component.**
- The model's performance is evaluated with a **Root Mean Square Error of 26.95** and a **Mean Absolute Percentage Error of 59.95.**
- The error is still **high** as the model fails to capture the **increasing** trend in the time series while in actual it is decreasing. Also, the seasonality was not captured well.

- **With Seasonal Differencing (D = 1):**

As noticed from the monthly plot shown below, the time series is relatively constant for each month except for December, which shows a pattern. Since December has the highest sales, it has a significant impact on the time series.

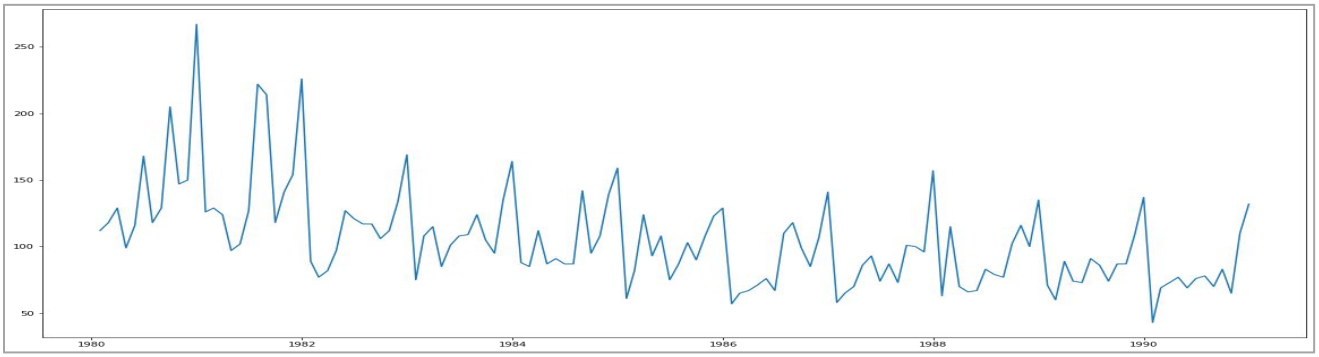

Fig 2.38 Time Series Plot: Train Data

To address this, we **apply a 12–month seasonal difference** to the data and examine if it makes the training data stationary.

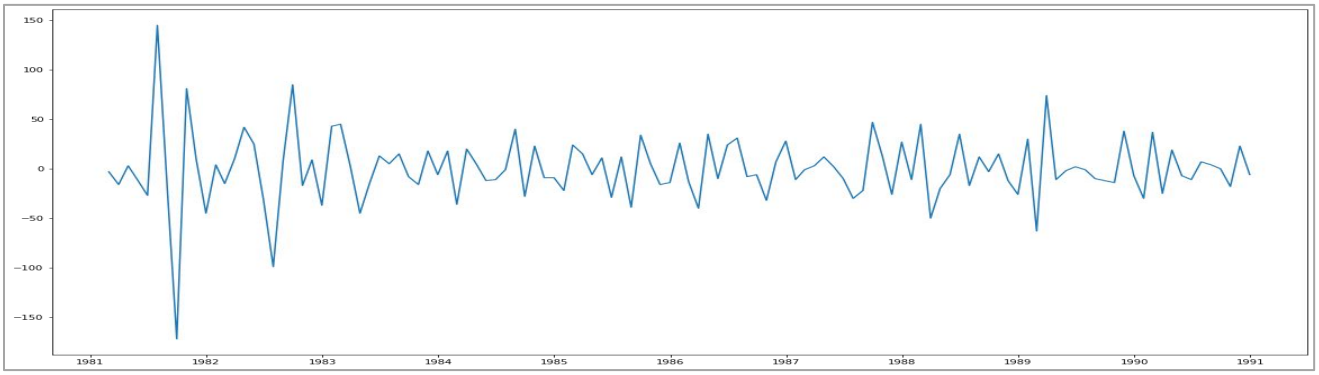

Fig 2.39 Time Series Plot: Test Data

The time series looks stationary. Let's check for **stationarity** using **Augmented Dickey – Fuller Test.**

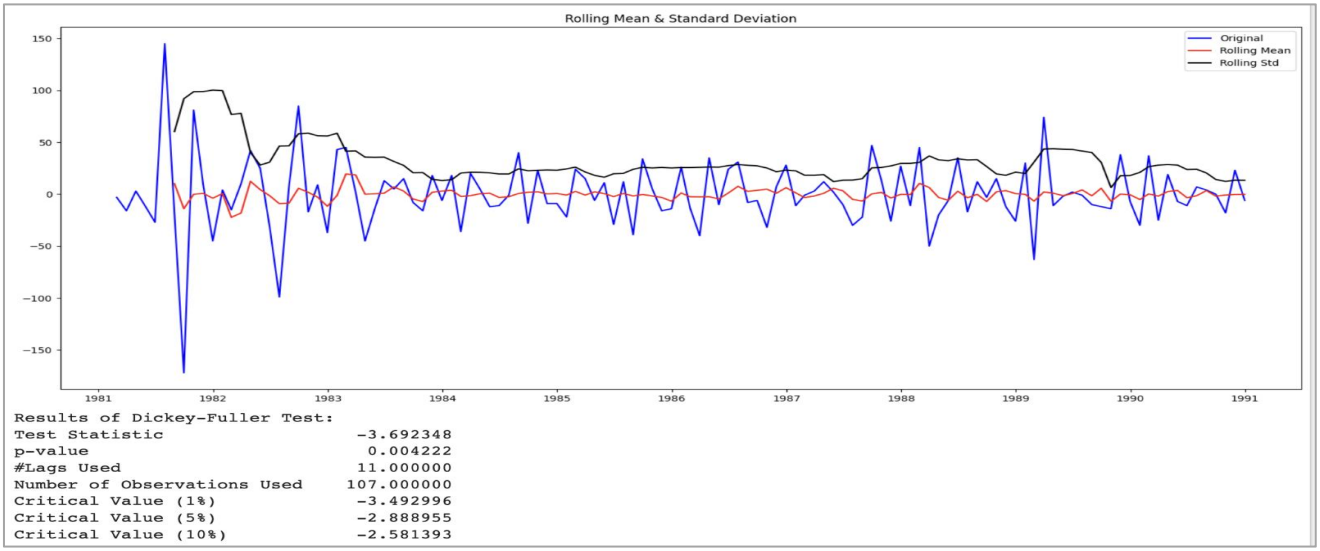

Fig 2.40 Stationarity of Differenced Training Data Using AD Fuller Test (D=1)

- The **p–value is less than 0.05**, leading us to **reject the null hypothesis (Ho)** at 95% confidence level and conclude that the **time series is stationary**.
- Therefore, we can build our model **with d = 0 (regular differencing)** and **D = 1 (seasonal differencing),** as applying only seasonal differencing makes the time series stationary and prevents over–differencing.

To find the values of p, q, P & Q, we iterated values between and computed AIC values for each combination. Lowest 5 shown below. We built the model on the values: p = 0, d =1, q = 2, P = 2, D =1, Q = 2, S = 12

	param	seasonal	AIC
26	(0, 1, 2)	(2, 1, 2, 12)	774.969120
53	(1, 1, 2)	(2, 1, 2, 12)	776.940108
80	(2, 1, 2)	(2, 1, 2, 12)	776.996100
17	(0, 1, 1)	(2, 1, 2, 12)	782.153872
79	(2, 1, 2)	(2, 1, 1, 12)	783.703652

Table 2.37 SARIMA AIC Parameters with Seasoning

SARIMA Results

SARIMAX Results

Dep. Variable:	Rose	No. Observations:	132
Model:	SARIMAX(0, 1, 2)x(2, 1, 2, 12)	Log Likelihood	-380.485
Date:	Sun, 06 Aug 2023	AIC	774.969
Time:	07:14:14	BIC	792.622
Sample:	01-31-1980	HQIC	782.094
	- 12-31-1990		
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	0.0479	0.177	0.271	0.786	-0.299	0.394
ar.S.L24	-0.0419	0.028	-1.513	0.130	-0.096	0.012
ma.S.L12	-0.7525	0.301	-2.503	0.012	-1.342	-0.163
ma.S.L24	-0.0722	0.204	-0.354	0.723	-0.472	0.327
sigma2	187.8724	45.274	4.150	0.000	99.137	276.608

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 2.38 Auto SARIMA with Differencing Model Summary

Diagnostic Plot

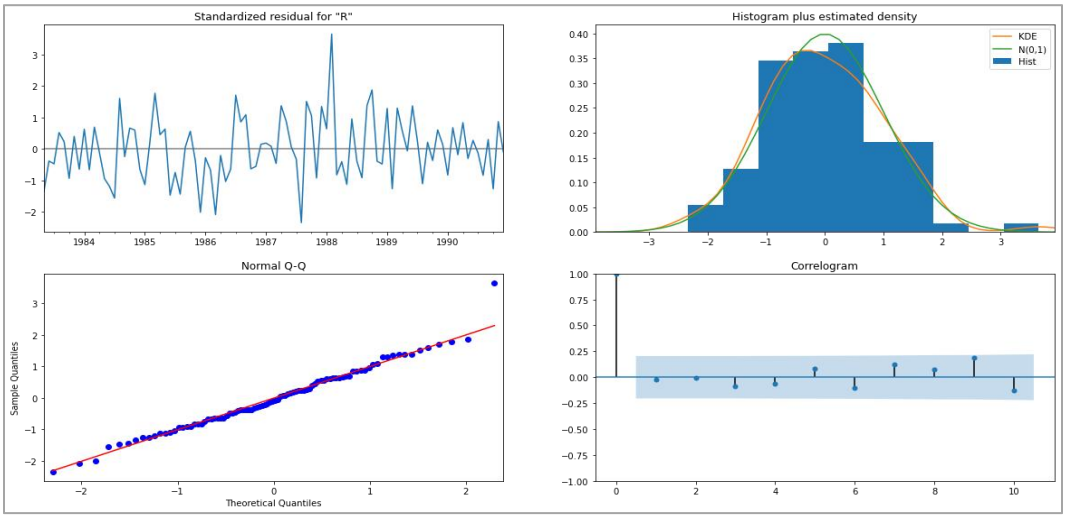

Fig 2.41 Diagnostic Plot: Automated SARIMA(0, 1, 2)(2, 1, 2, 12)

Forecast on the test data

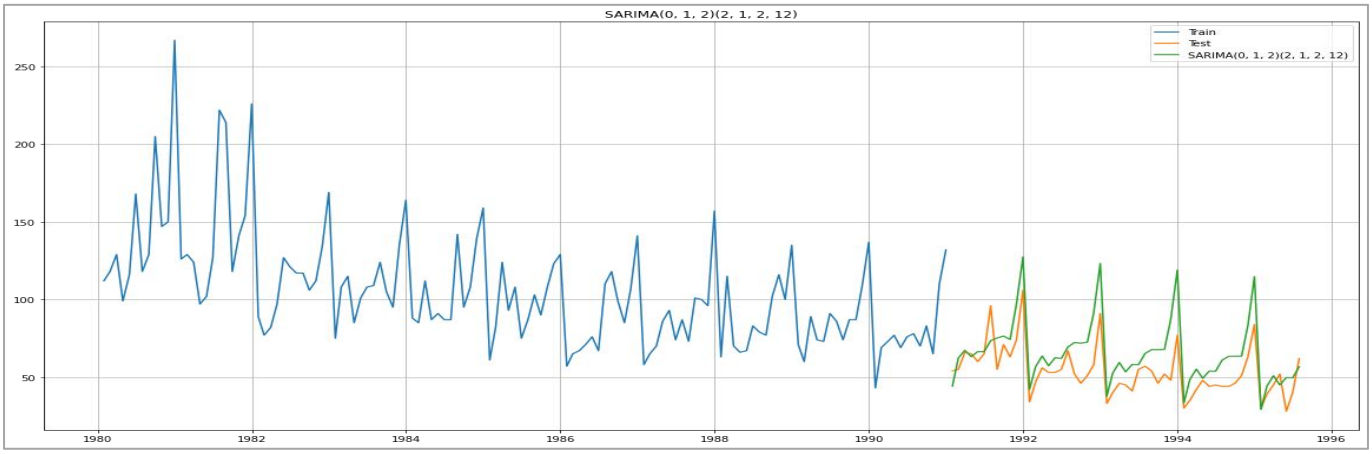

Fig 2.42 Time Series Plot: Automated SARIMA (0, 1, 2)(2, 1, 2, 12)

Model Performance			
Model	**Train RMSE**	**Test RMSE**	**Test MAPE**
Automated SARIMA (0, 1, 2)(2, 1, 2, 12)	39.18	16.52	44.49

Table 2.39 Model Performance Summary – Automated SARIMA (0, 1, 2)(2, 1, 2, 12)

- The **Automated SARIMA (0, 1, 2)(2, 1, 2, 12)** model **successfully captures** both the **trend and seasonality** in the data.
- The **Root Mean Square Error is 16.52,** and the **Mean Absolute Percentage Error is 44.49** for the **automated SARIMA** model **with seasonal differencing.**
- This model performs **better than** the model **without seasonal differencing,** indicating that **incorporating seasonal differencing improves the accuracy** of the forecast.

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

Final Results Table:

Model Name	Train_RMSE	Test_RMSE	MAPE
Alpha = 0.04, Beta = 0.52, Gamma = 0.10 Triple Exponential Smoothing	21.69	8.22	30.67
2 point Trailing Moving Average	19.67	11.53	15.73
Alpha = 0.04 & Beta = 0.47 Double Exponential Smoothing Model	39.2	14.46	34.86
Alpha = 1.49*10^-8, Beta = 5.44*10^-9 Double Exponential Smoothing Model	30.72	15.28	30.14
Linear Regression On Time	30.72	15.28	25.01
Automated SARIMA(0, 1, 2)(2, 1, 2, 12)	39.18	16.52	44.49
Alpha = 0.111, Beta = 0.049, Gamma = 0.362 Triple Exponential Smoothing	18.41	19.15	47.83
Automated SARIMA (0, 1, 2)(2, 0, 2, 12)	32.23	26.95	59.95
Automated SARIMA(0, 1, 2)(2, 0, 2, 12)	32.23	26.95	59.95
Alpha = 0.07 Simple Exponential Smoothing	32.65	36.46	88.74
Alpha = 0.099 Simple Exponential Smoothing	31.5	36.82	76.0
Automated ARIMA(0, 1, 2)	31.25	37.33	77.04
Simple Average Model	36.03	53.49	110.7
NaiveModel	45.06	79.74	164.99

Table 2.40 Model Performance Summary – Consolidated

- **Model Performance:** After evaluating various forecasting models, the top-performing ones are:
  - **Triple Exponential Smoothing Model (Alpha = 0.04, Beta = 0.52, Gamma = 0.10):** This model exhibits the highest accuracy, with a Train RMSE of 21.69, Test RMSE of 8.22, and MAPE of 30.67.
  - **2-Point Trailing Moving Average:** This model performs well, with a Train RMSE of 19.67, Test RMSE of 11.53, and MAPE of 15.73.

- **Double Exponential Smoothing Model (Alpha = 0.04 & Beta = 0.47):** While not as accurate as the top model, it still shows decent performance, with a Train RMSE of 39.20, Test RMSE of 14.46, and MAPE of 34.86.
- **Double Exponential Smoothing Model (Alpha = 1.49*10^-8, Beta = 5.44*10^-9):** This model exhibits moderate accuracy, with a Train RMSE of 30.72, Test RMSE of 15.28, and MAPE of 30.14.
- **Linear Regression on Time:** This model performs similarly to the previous one, with a Train RMSE of 30.72, Test RMSE of 15.28, and MAPE of 25.01.
- **Automated SARIMA(0, 1, 2)(2, 1, 2, 12):** This model shows reasonable accuracy, with a Train RMSE of 39.18, Test RMSE of 16.52, and MAPE of 44.49.
- **Triple Exponential Smoothing Model (Alpha = 0.111, Beta = 0.049, Gamma = 0.362):** Another triple exponential model with a slightly lower performance, having a Train RMSE of 18.41, Test RMSE of 19.15, and MAPE of 47.83.

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

The most optimum models for forecasting are as shown below:

Final Results Table:

Model Name	Train_RMSE	Test_RMSE	MAPE
Alpha = 0.04, Beta = 0.52, Gamma = 0.10 Triple Exponential Smoothing	21.69	8.22	30.67
2 point Trailing Moving Average	19.67	11.53	15.73
Alpha = 0.04 & Beta = 0.47 Double Exponential Smoothing Model	39.2	14.46	34.86
Alpha = 1.49*10^-8, Beta = 5.44*10^-9 Double Exponential Smoothing Model	30.72	15.28	30.14
Linear Regression On Time	30.72	15.28	25.01
Automated SARIMA(0, 1, 2)(2, 1, 2, 12)	39.18	16.52	44.49

Table 2.41 Best Performing Models

Optimal Model: Tuned Triple Exponential Smoothing

- Alpha = 0.04, Beta = 0.52, Gamma = 0.10
- Best accuracy in capturing trend and seasonality.

2-Point Trailing Moving Average:

- Used for analysis, provides only 1-step forecast.
- Inadequate for capturing seasonality during forecasts.

Linear Regression & Double Exponential Smoothing:

- Accurately captures trend but fails to capture seasonality.
- Better accuracy due to the slowly flattening trend.

Automated SARIMA(0, 1, 2)(2, 1, 2, 12)

- Shows reasonable accuracy
- Captures both Trend and seasonality

Recommendation

- For comprehensive forecasting, we will build **Automated SARIMA(0, 1, 2)(2, 1, 2, 12)** model and **Triple Exponential Smoothing** (Alpha = 0.04, Beta = 0.52, Gamma = 0.10) model.
- Capable of predicting **12 months** ahead.
- Capture both **trend and seasonality** effectively

We'll forecast on above 2 Models

➢ Forecasting on Tuned Triple Exponential Model with Alpha = 0.04, Beta = 0.52, Gamma = 0.10

To **forecast 12 months** into the future, we build the model on the **full data first** before forecasting

RMSE Full Model = **18.07**

- **Assumption**: Forecast distribution's standard deviation ≈ Residual standard deviation.
- **Purpose**: Helps estimate uncertainty in the forecast.
- **Use**: Construct confidence intervals with a specified level of confidence.

Forecast Results: –

	lower_ci	prediction	upper_ci
1995-08-31	17.729526	53.245088	88.760651
1995-09-30	17.942683	53.458245	88.973807
1995-10-31	18.939987	54.455549	89.971111
1995-11-30	28.133732	63.649294	99.164856
1995-12-31	53.560644	89.076206	124.591768
1996-01-31	1.059142	36.574704	72.090266
1996-02-29	8.638256	44.153818	79.669380
1996-03-31	14.478913	49.994476	85.510038
1996-04-30	13.522119	49.037681	84.553243
1996-05-31	11.919426	47.434988	82.950550
1996-06-30	17.462925	52.978487	88.494049
1996-07-31	25.609165	61.124727	96.640289

Table 2.42 Forecast Results – Triple Exponential Model with Alpha = 0.04, Beta = 0.52, Gamma = 0.10

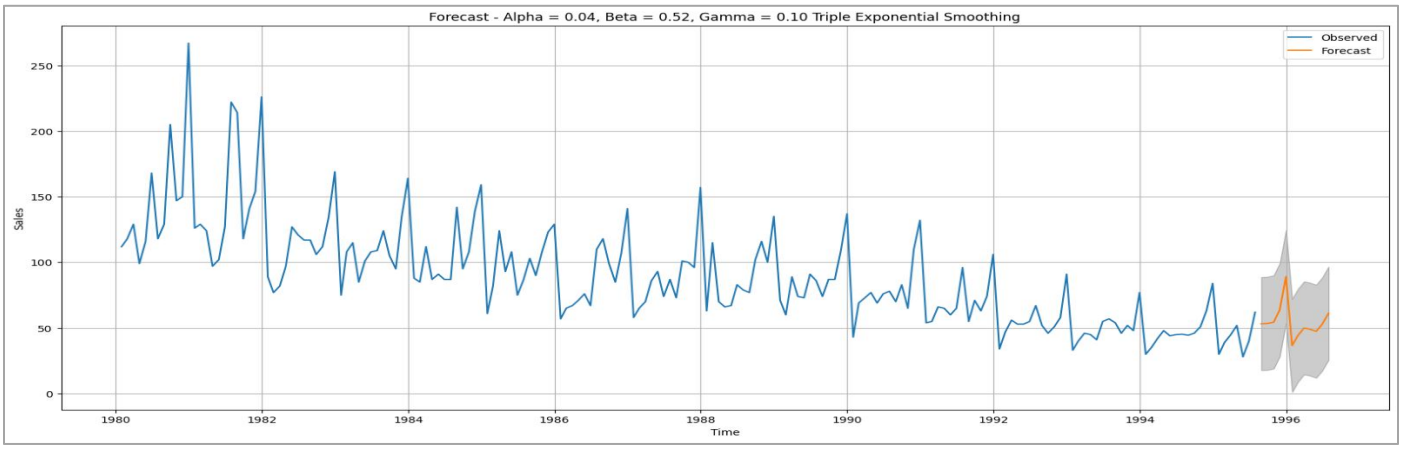

Fig 2.43 Forecasted Plot: Triple Exponential Model with Alpha = 0.04, Beta = 0.52, Gamma = 0.10

➢ Forecasting on Automated SARIMA with seasonal differencing – SARIMA (0, 1, 2)(2, 1, 2, 12)

Stationarity Check on Full data:

After applying seasonal differencing (D=12), at **p–value < 0.05** we **reject the Null Hypothesis** & conclude that the **full data** is also **Stationary at 95% confidence level.**

Let's now forecast by first building the model on whole data. The RMSE on the full data with seasonal differencing is **33.47.**

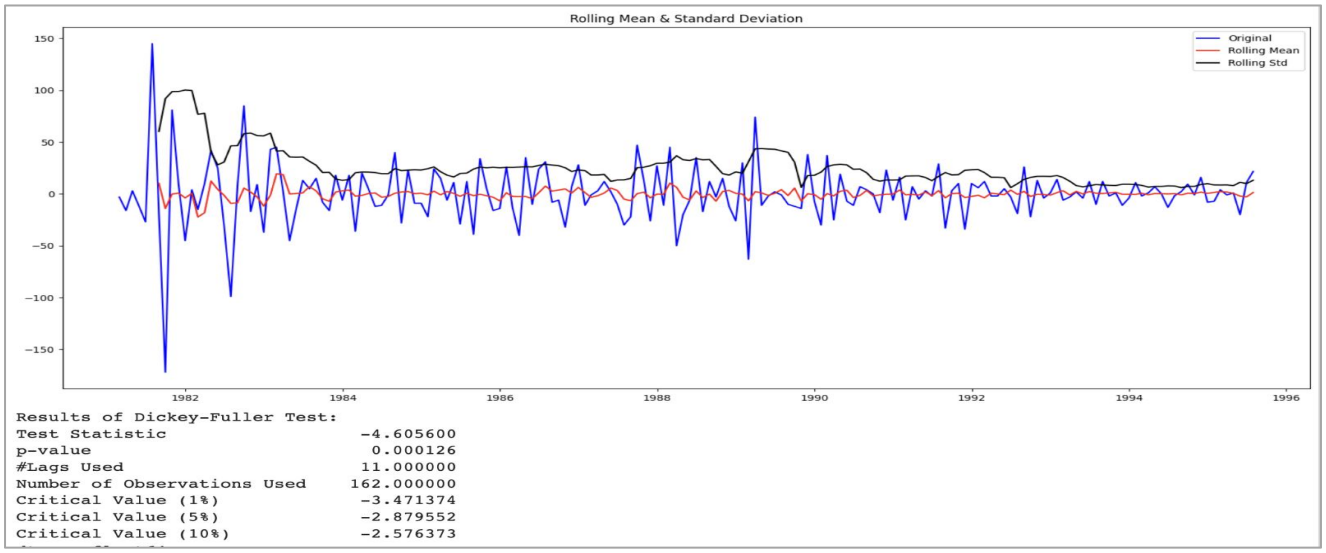

Fig 2.44 Stationarity of Differenced Data Using AD Fuller (D=12)

Model Results

SARIMAX Results

Dep. Variable:	Rose	No. Observations:	187
Model:	SARIMAX(0, 1, 2)x(2, 1, 2, 12)	Log Likelihood	-588.612
Date:	Sun, 06 Aug 2023	AIC	1191.223
Time:	07:44:38	BIC	1212.156
Sample:	01-31-1980	HQIC	1199.729
	- 07-31-1995		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.8254	0.080	-10.332	0.000	-0.982	-0.669
ma.L2	-0.0807	0.086	-0.933	0.351	-0.250	0.089
ar.S.L12	0.0635	0.160	0.398	0.691	-0.249	0.376
ar.S.L24	-0.0340	0.019	-1.790	0.074	-0.071	0.003
ma.S.L12	-0.6954	0.207	-3.361	0.001	-1.101	-0.290
ma.S.L24	-0.0547	0.150	-0.365	0.715	-0.348	0.239
sigma2	166.1069	17.904	9.278	0.000	131.016	201.198

Ljung-Box (L1) (Q):	0.07	Jarque-Bera (JB):	8.27
Prob(Q):	0.79	Prob(JB):	0.02
Heteroskedasticity (H):	0.51	Skew:	0.33
Prob(H) (two-sided):	0.02	Kurtosis:	3.95

Table 2.43 Auto SARIMA Forecast Model Summary

Diagnostic Plot

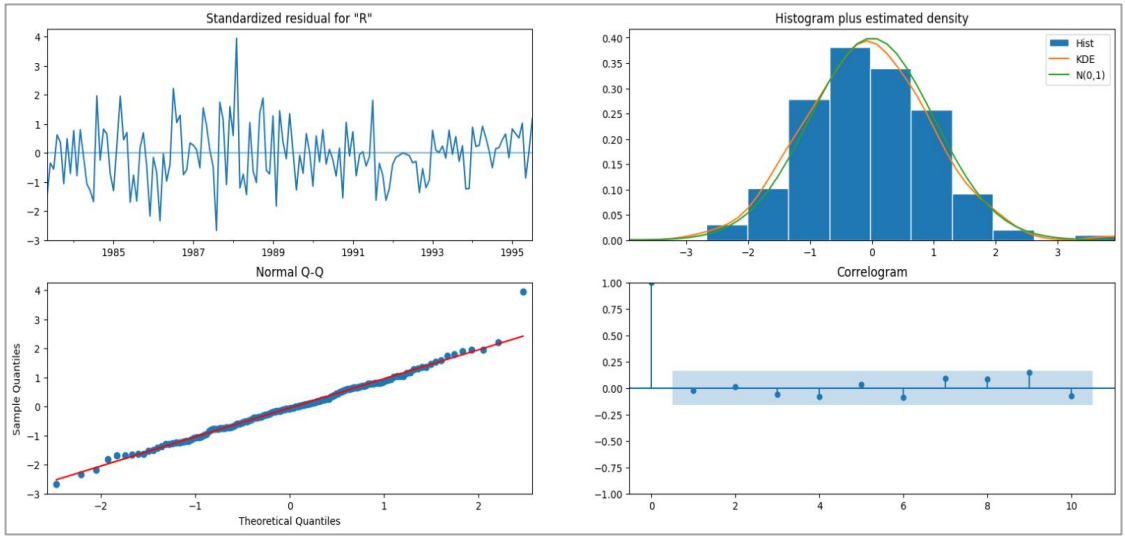

Fig 2.45 Diagnostic Plot: Automated SARIMA(0, 1, 2)(2, 1, 2, 12)

Forecast Results: –

Rose	mean	mean_se	mean_ci_lower	mean_ci_upper
1995-08-31	42.985812	12.890666	17.720571	68.251053
1995-09-30	43.514729	13.085857	17.866921	69.162537
1995-10-31	45.493227	13.141787	19.735798	71.250656
1995-11-30	57.522291	13.197489	31.655688	83.388895
1995-12-31	84.992662	13.252978	59.017303	110.968021
1996-01-31	20.574774	13.308022	-5.508469	46.658018
1996-02-29	30.225068	13.363002	4.034066	56.416070
1996-03-31	36.974571	13.417674	10.676413	63.272728
1996-04-30	38.519865	13.472149	12.114938	64.924791
1996-05-31	29.045411	13.526459	2.534038	55.556784
1996-06-30	36.323622	13.580559	9.706215	62.941029
1996-07-31	49.468464	13.634457	22.745419	76.191509

Table 2.44 Forecast Results – Automated SARIMA(0, 1, 2)(2, 1, 2, 12)

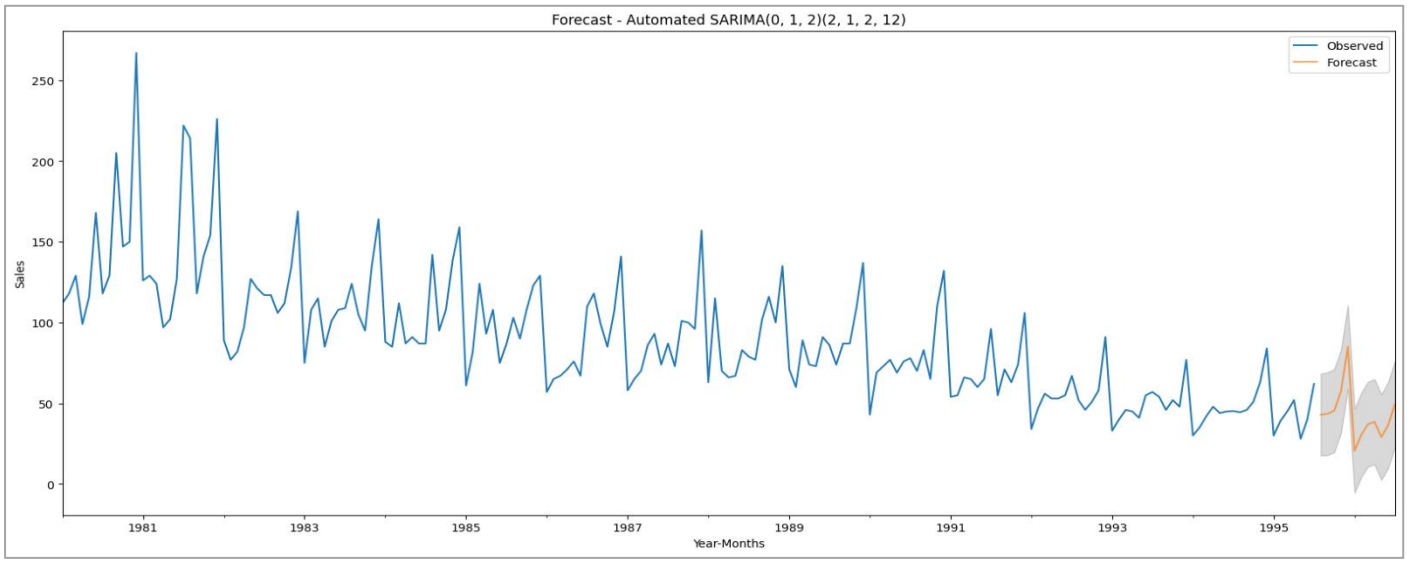

Fig 2.46 Forecasted Plot: Automated SARIMA(0, 1, 2)(2, 1, 2, 12)

- The predictions from both the models consistently **indicate** that the **sales will exhibit a steady trend** and **seasonality** similar to previous years, with a **slight upswing** during the **holiday season (November–December)** compared to the previous year.

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

Forecasting Insights:

- **Data Analysis:**
  - The data shows a consistent **decreasing trend** in Rose wine sales over the years, indicating a **declining pattern** in customer demand for this type of wine.
  - Despite the overall decreasing trend, there are still **seasonal variations** in the data, with sales **peaking** in certain months (towards the end of each year) and **dropping** in others.

- **Time Series Characteristics:**
  - The time series exhibits a clear **downward trend** across the years, with a sharper dip observed after 1991 compared to before 1991.
  - **Seasonality** is present in the data, as sales pick up in the ending months of the year.
  - The time series cannot be definitively classified as either an additive or multiplicative time series, but it leans closer to a multiplicative nature.

- **Model Performance:**
  - The **Tuned Triple Exponential Smoothing model with Alpha = 0.04, Beta = 0.52, and Gamma = 0.10** shows the **best accuracy** in capturing **trend and seasonality.**
  - The **2-Point Trailing Moving Average** model is mainly used for analysis and provides only a 1-step forecast, making it inadequate for capturing seasonality during forecasts.
  - **Linear Regression & Double Exponential Smoothing** accurately **capture trend** but **fail to capture seasonality**. However, they show better accuracy due to the slowly flattening trend.
  - The **Automated SARIMA(0, 1, 2)(2, 1, 2, 12)** model shows **reasonable accuracy** and **captures both trend and seasonality.**

- **Prediction Model:**

For comprehensive forecasting and predicting 12 months ahead, we built the following models:

  - **Seasonality** is present in the data, as sales pick up in the ending months of the year.
  - **Automated SARIMA(0, 1, 2)(2, 1, 2, 12)**: It captures both trend and seasonality effectively. The **RMSE** of the Full Model is **18.07.**
  - **Tuned Triple Exponential Smoothing (Alpha = 0.04, Beta = 0.52, Gamma = 0.10):** It exhibits the best accuracy in capturing trend and seasonality. The **RMSE** of the Full Model is **33.47.**

- **Measures for Future Sales:**

A **critical decision** must be taken to **either discontinue** the Rose wine or **undertake product** and process **enhancements** to boost sales. To improve future sales, ABC Estate Wines should consider the following strategies:-

  - **Customer Engagement: Strengthen** customer relationships through **personalized offers, loyalty programs, and active engagement** to foster repeat business.
    - Marketing promotions, Sponsoring small & large events having target audience
    - Advertisements strategies such as launching non-alcoholic beverages with the same brand name to again popularity
  - **Capitalize on Seasonal Trends:** Plan production and marketing efforts to meet increased demand during holiday seasons, especially in November and December.
  - **Inventory Management:** Implement effective inventory management to avoid stockouts during peak periods and minimize excess inventory during slower periods.
  - **Pricing Strategy:** Utilize dynamic pricing to adjust prices during peak and off-peak periods, attracting more customers and optimizing revenue.
  - **Competitor benchmarking:** Figure out competitor businesses for similar wines to understand the demand of rose wine in the market and make necessary changes.