TSF CODED PROJECT

Business Report

DSBA

Submitted By: Maheep Singh
Batch : PGP-DSBA (PGPDSBA.O.AUG24.A)

Table of Contents

List of Figures

List of Tables

Business Context & Data Dictionary

Context

As an analyst at ABC Estate Wines, we are presented with historical data encompassing the sales of different types of wines throughout the 20th century. These datasets originate from the same company but represent sales figures for distinct wine varieties. Our objective is to delve into the data, analyze trends, patterns, and factors influencing wine sales over the course of the century. By leveraging data analytics and forecasting techniques, we aim to gain actionable insights that can inform strategic decision-making and optimize sales strategies for the future.

Objective

The primary objective of this project is to analyze and forecast wine sales trends for the 20th century based on historical data provided by ABC Estate Wines. We aim to equip ABC Estate Wines with the necessary insights and foresight to enhance sales performance, capitalize on emerging market opportunities, and maintain a competitive edge in the wine industry.

Data Description

The data provided is a times series for 2 types of wines, Rose & Sparkling, from Jan. 1980 to July 1995.

Data Dictionary:

1. Rose Wine Sales over time: -
    - **YearMonth**: Year & Month of sale
    - **Rose**: No. of Rose wine units sold for the month
2. Sparkling Wine Sales over time: -
    - **YearMonth**: Year & Month of Sale
    - **Sparkling**: No. of Sparkling wine units sold for the month

Rubric Question 1: Define the Problem and Perform Exploratory Data Analysis

Define the Problem

Data on wine sales from the 20^{th} century are available from ABC Estate Wines, a wine producing firm, and should be examined. With the provided information, an estimate of wine sales in the 20th century must be forecasted, followed by insights & recommendations to enhance sales performance & capitalize on emerging market opportunities, thus, maintaining a competitive edge in the wine industry.

Data Overview – Rose Wine

- **Load dataset** & display top & bottom 5 rows: -

First few rows of Data

	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

Last few rows of Data

	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

Figure 1: Rose | Sample of top & bottom 5 rows of the dataset

- **Creating a Time-stamp Index**: -
  - Create a Date-Time Series **'TimeStamp' to be added** to the original table, ranging from **1/1/1980 to 8/1/1995**.
  - The newly created Series is added to the table & then column **'YearMonth' can be removed**.
  - **'TimeStamp'** column is set as index for the table.
  - Rename the column **'Rose'** as '**RoseWine_Sales'** to denote the units of wine sold for the Rose brand.
  - Below are the top & bottom 5 rows of the table post above changes: -

TimeStamp	RoseWine_Sales
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

TimeStamp	RoseWine_Sales
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

Figure 2: Rose | Sample of top & bottom 5 rows of the dataset (post time-series index creation)

- There are a total of **187 rows & 1 column** in the dataset by checking the shape of the dataset: -

```
Shape of dataset:  (187, 1)
```

Figure 3: Rose | Shape of dataset

- **Checking datatypes**: -

```
<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   RoseWine_Sales  185 non-null    float64
dtypes: float64(1)
```

Figure 4: Rose | Checking datatypes

– There is 1 column of float64 (numeric) datatype in the dataset.

- **Statistical Summary of the dataset**: -

	count	mean	std	min	25%	50%	75%	max
RoseWine_Sales	185.0	90.394595	39.175344	28.0	63.0	86.0	112.0	267.0

Figure 5: Rose | Statistical Summary of the dataset

Type	Columns	Observations & Insights
Numerical	RoseWine_Sales	✓ No. of wine units sold ranges between 28 to 267 per month ✓ Mean is ~90 units per month & Median is 86 units per month ✓ Standard Deviation is ~39 units per month

Table 1: Rose | Statistical Summary

- **Check & Treat Missing/Null Values**: -
  - Upon checking, 2 missing/null values were found: -

```
Null values:-

RoseWine_Sales    2
dtype: int64
```

	RoseWine_Sales
TimeStamp	
1994-07-31	NaN
1994-08-31	NaN

Figure 6: Rose | Null-value check

- Since, we are dealing with time series data, dropping any missing value would mean loss of significant information in time. Instead, we shall use **Interpolation** to impute missing values.
- **Spline interpolation** is a method used to estimate values between known data points. It fits a smooth curve (a spline) through the data points, making sure the curve is continuous and differentiable. This method is used when once wants to generate a smooth curve that passes through or near all the points.
- Use **Interpolate with Spline method (order 3)** to impute missing values. Below is the Null-value check result post imputation: -

```
Null-value check post imputation:-
RoseWine_Sales    0
dtype: int64
```

	RoseWine_Sales
TimeStamp	

Figure 7: Rose | Null-value check (post imputation)

Data Overview – Sparkling Wine

- **Load dataset** & display top & bottom 5 rows: -

First few rows of Data

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

Last few rows of Data

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

Figure 8: Sparkling | Sample of top & bottom 5 rows of the dataset

- **Creating a Time-stamp Index**: -
  - Create a Date-Time Series **'TimeStamp' to be added** to the original table, ranging from **1/1/1980 to 8/1/1995**.
  - The newly created Series is added to the table & then column **'YearMonth' can be removed**.
  - **'TimeStamp'** column is set as index for the table.
  - Rename the column **'Sparkling** as '**SparklingWine_Sales'** to denote the units of wine sold for the Sparkling brand.
  - Below are the top & bottom 5 rows of the table post above changes: -

TimeStamp	SparklingWine_Sales
1980-01-31	1686
1980-02-29	1591
1980-03-31	2304
1980-04-30	1712
1980-05-31	1471

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

Figure 9: Sparkling | Sample of top & bottom 5 rows of the dataset (post time-series index creation)

- There are a total of **187 rows & 1 column** in the dataset by checking the shape of the dataset: -

```
Shape of dataset:  (187, 1)
```

Figure 10: Sparkling | Shape of dataset

- **Checking datatypes**: -

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

Figure 11: Sparkling | Checking datatypes

– There is 1 column of float64 (numeric) datatype in the dataset.

- **Statistical Summary of the dataset**: -

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

Figure 12: Sparkling | Statistical Summary of the dataset

Type	Columns	Observations & Insights
Numerical	SparklingWine_Sales	✓ No. of wine units sold ranges between 1070 to 7242 per month ✓ Mean is ~2402 units per month & Median is 1874 units per month ✓ Standard Deviation is ~1295 units per month

Table 2: Sparkling | Statistical Summary

- **Check & Treat Missing/Null Values**: -
  - Upon checking, no missing/null values were found: -

```
Null values:-

SparklingWine_Sales    0
dtype: int64
```

	SparklingWine_Sales
TimeStamp	

Figure 13: Sparkling | Null-value check

Exploratory Data Analysis – Rose Wine

- **Plot data for wine-sales over time (month-year): -**

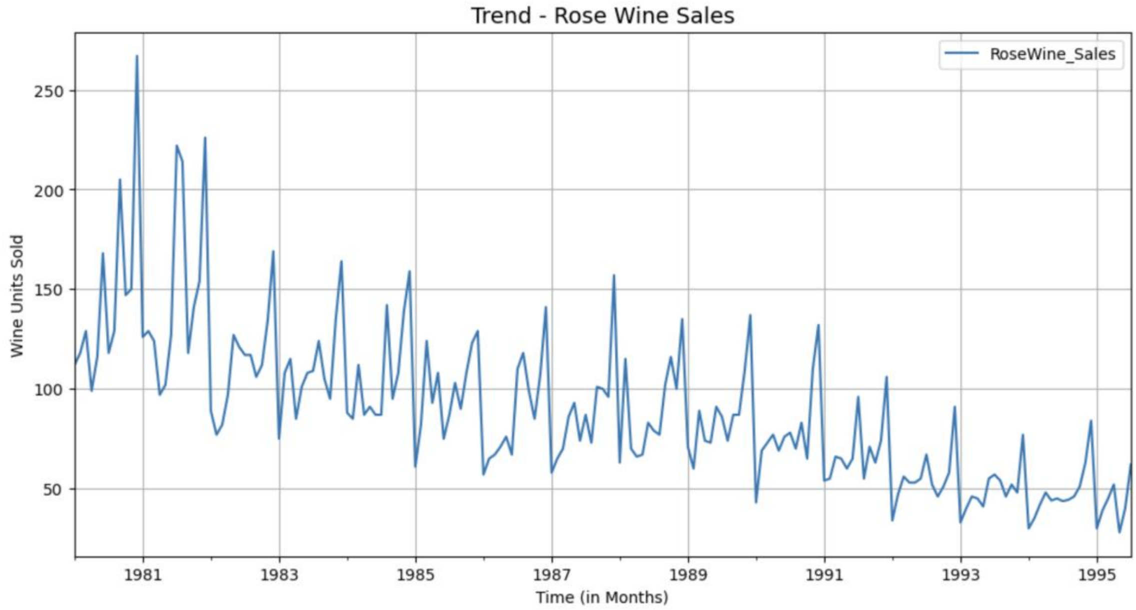


Figure 14: Rose | Trend – Month-Year Sales

- **Observations: -**
  - The line plot shows the monthly sales trend of Rose wine from 1980 to 1995.
  - **Overall Trend**: There's a **clear downward trend** in sales over the 15-year period. Sales start high in the early 1980s and steadily decline through the mid-1990s. This suggests a **gradual loss of consumer interest** or possible **market shrinkage over time**
  - **Seasonality:** There are regular, recurring peaks and troughs within each year. **Spikes tend to occur at similar intervals (e.g., once every 12 months)**, suggesting annual seasonality — possibly related to summer months when this wine might be more popular. As sales decline toward the 1990s, **seasonality becomes less pronounced, possibly due to lower overall sales volume**.

- **Plot a boxplot to understand the sales across different years: -**

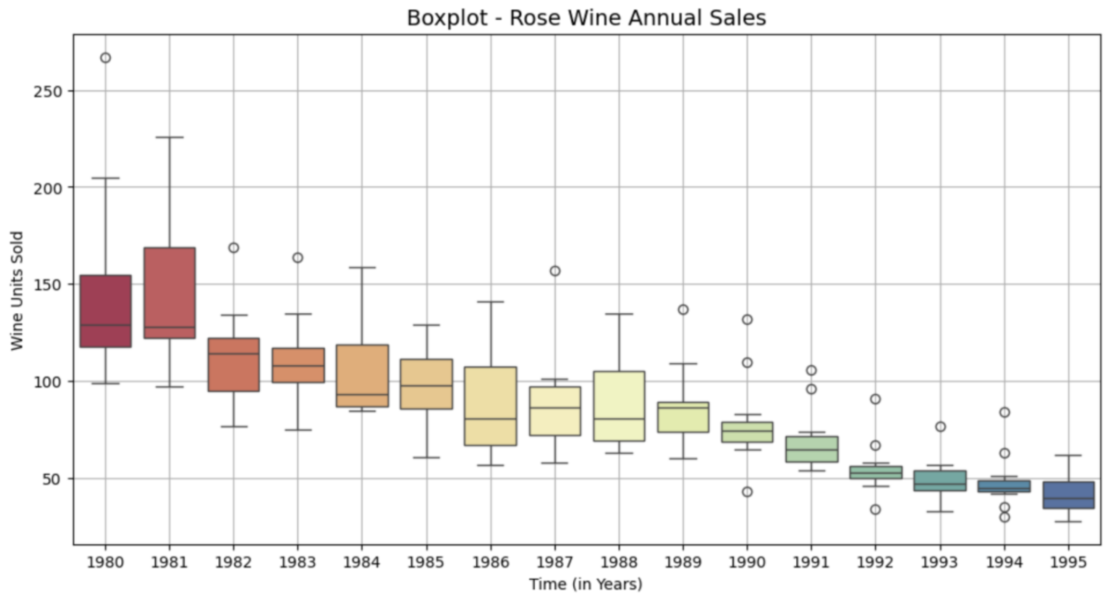


Figure 15: Rose | Boxplot of Wine sales across various years

- **Observations: -**
  - There's a **strong negative trend** in both the **median** and **range** of wine sales.
  - **Early years show strong performance with high variability, while later years show lower and more stable (but worse) sales.**
  - This suggests both **declining demand** and possibly **reduced market efforts** or **external challenges** impacting performance over time.
  - **High outliers in early 1980s** (especially 1980–1981) show months where sales spiked to **above 200 or even 250 units**. **Fewer or no high outliers** post-1990, meaning Rosé wine no longer achieved high sales even occasionally.

▪ **Plot a boxplot to understand the sales at month level: -**

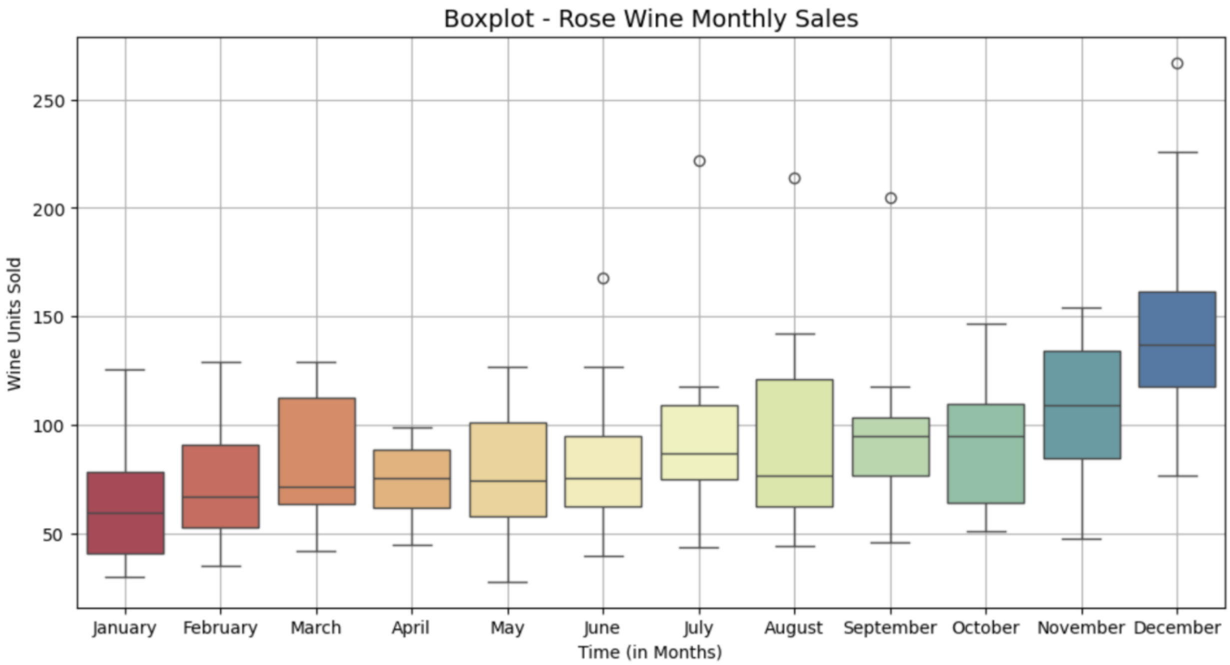


Figure 16: Rose | Boxplot – Monthly Wine sales

- **Observations: -**
  - **Wine sales are seasonal**, with demand surging in **summer** and again during the **year-end holiday season**.
  - **January has the lowest** wine sales while **December sees the greatest**. The sales modestly grow from January to August and then sharply climb after that
  - **Sales are lowest in the early months of the year.**
  - **High variability** in peak months (e.g., December) suggests some years had **exceptionally strong performances**.

- **Plot a time series MonthPlot to understand the spread: -**

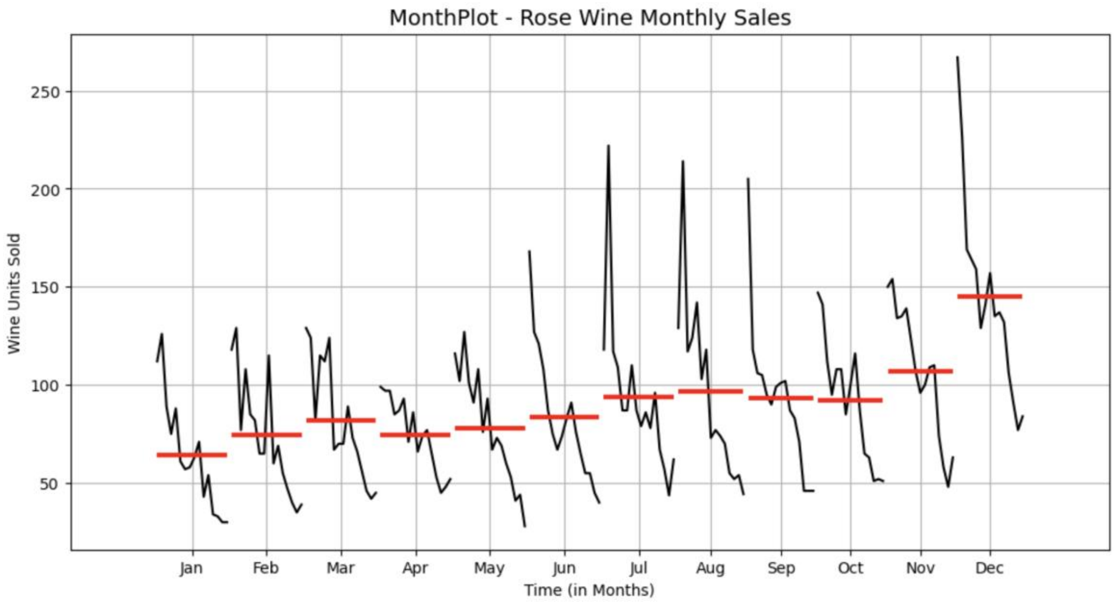


Figure 17: Rose | MonthPlot – Monthly Wine sales

- **Observations: -**
  - **MonthPlot** of wine monthly sales is a time series visualization that highlights **seasonality across individual months**. It plots **monthly data points** year by year (thin black lines) and overlays **monthly means** with **thick red horizontal lines**.
  - As clearly evident, **there is strong seasonality in wine sales.**
  - **December is the clear peak month, followed by July–August and November.**
  - **January to March are lowest-performing and stable months.**
  - **High variability** in peak months (e.g., December) suggests some years had **exceptionally strong performances**.
  - Some months (like July, November, December) show **taller black line peaks**, indicating more **volatile sales patterns**. Other months (like January–March) show **tight clustering**, implying **more stable but lower sales**.

- **Plot a trend for each month sales across all years: -**

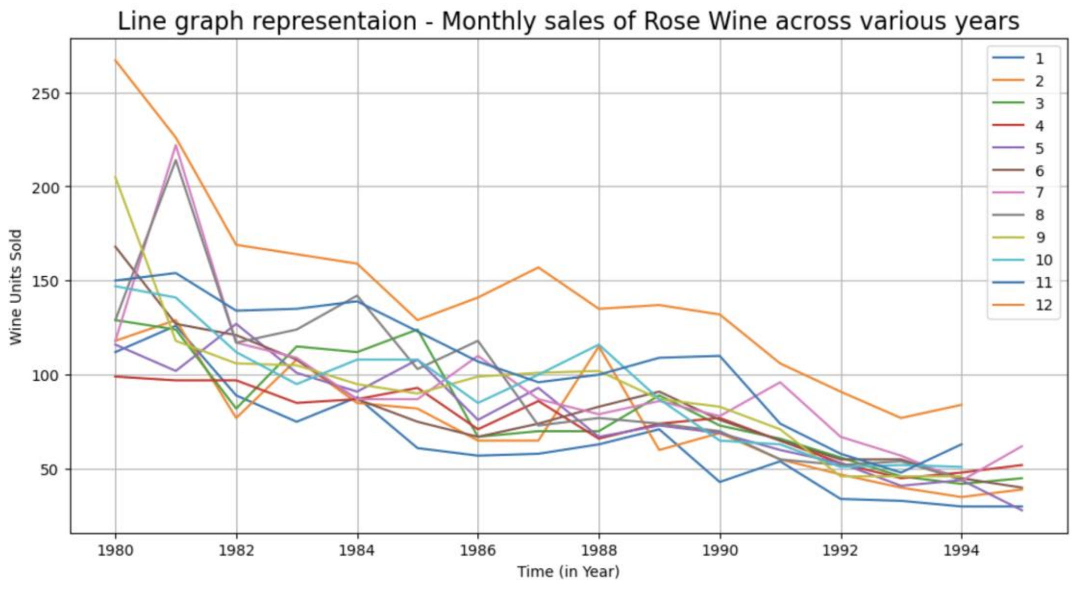


Figure 18: Rose | Trend – Each month sales across years

- **Observations: -**
  - **Long-term decline in wine sales across all months. The decline is steepest in the early 1980s**, followed by a gradual tapering into the 1990s.
  - **December is the strongest month** consistently, followed by summer and fall months.
  - **Seasonal effects become less pronounced over time.**
  - As time progresses, monthly lines **converge toward lower values**, indicating **flattening of seasonality** and a **shrinking overall market**.

- **Plot the Empirical Cumulative Distribution Function: -**

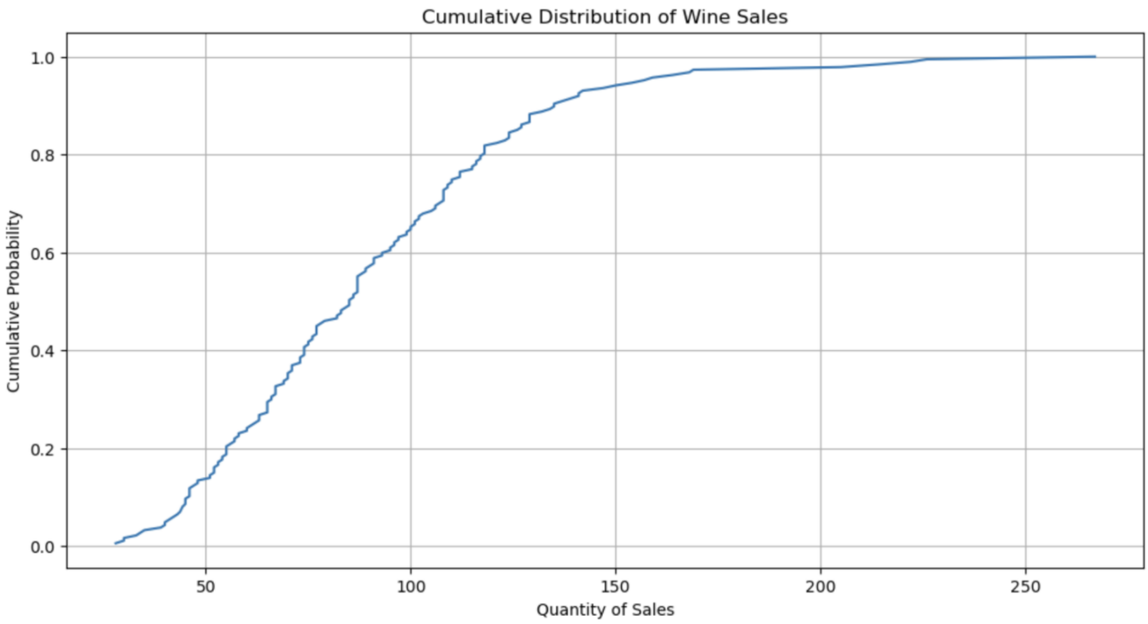


Figure 19: Rose | Empirical Cumulative Distribution of Wine sales

- **Observations: -**
  - **The majority of wine sales were moderate (70–130 units)** as it clear that the bulk of sales were in the range of 70 to 100 units.
  - **Extreme sales (above 200 units)** were **rare** and represent **outliers or special events** (e.g., holiday boosts).

- **Plot the average sales per month and the month-on-month percentage change of sales: -**

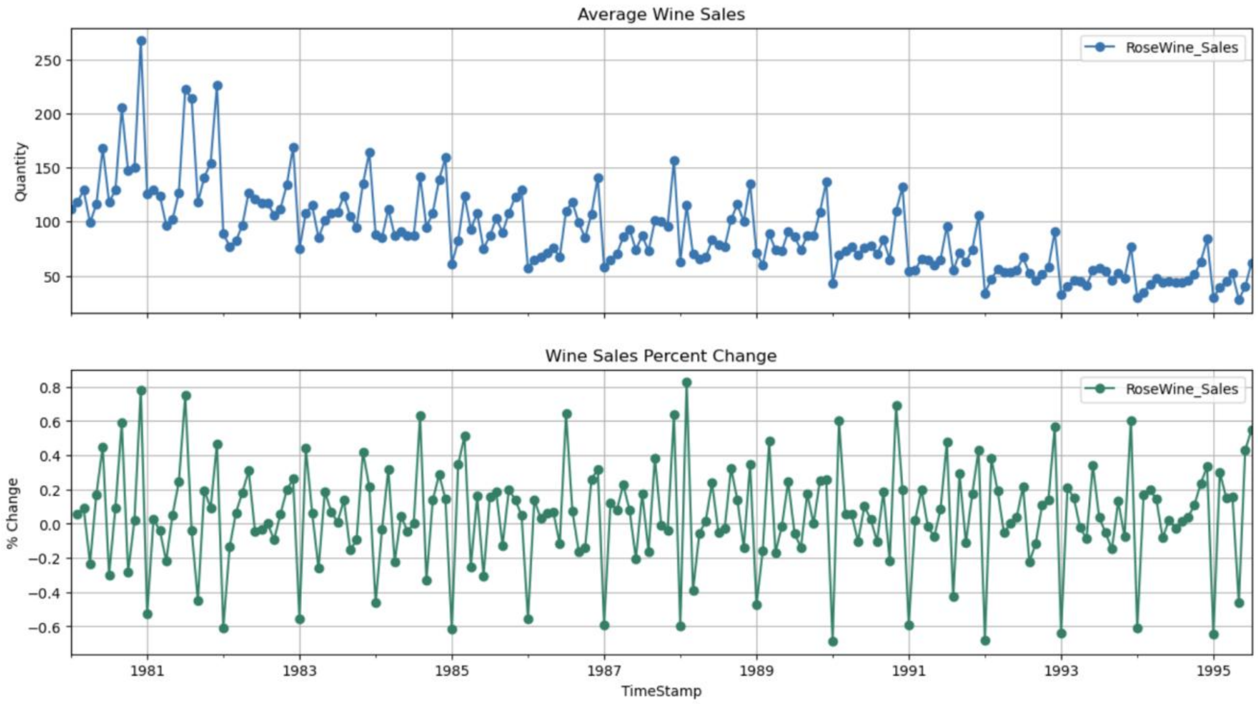


Figure 20: Rose | Trend – Average Wine sales & Wine percent change

- **Observations: -**
  - **Sales Trend (Top Plot):** Confirms a declining trend in absolute sales, particularly after 1985. Early years had high spikes, potentially due to product launches, promotions, or market entry effects.
  - **Volatility (Bottom Plot): Sales percent change is highly unstable**.
  - **Sales are declining**, but **volatility remains high**, suggesting the company may need to: -
    - ✓ Improve sales forecasting and inventory planning
    - ✓ Investigate causes of month-to-month fluctuations
    - ✓ Stabilize demand with consistent marketing and customer engagement strategies

Exploratory Data Analysis – Sparkling Wine

- **Plot data for wine-sales over time (month-year): -**

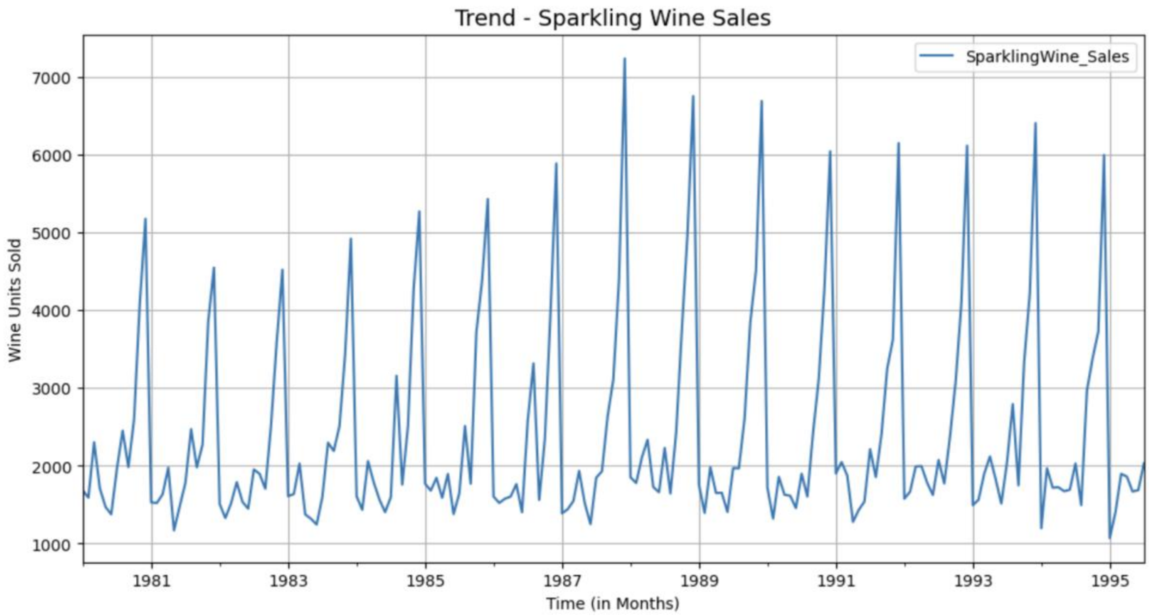


Figure 21: Sparkling | Trend – Month-Year Sales

- **Observations: -**
  - The line plot shows the monthly sales trend of Rose wine from 1980 to 1995.
  - **Overall Trend**: **Off-peak sales are stable**, suggesting a reliable base of regular consumers. There has been a constant pattern of sales with seasonality. **Over the years, the sales have consistent**.
  - **Seasonality: Sales are highly seasonal**, with massive spikes at the end of each year. These sharp peaks suggest a strong **seasonal surge**, likely due to **New Year's, Christmas, and other festive occasions** where sparkling wine is popular. Seasonality is **highly consistent** across all years.
  - The **height of annual peaks increases over time**, especially from 1984 onward.

- **Plot a boxplot to understand the sales across different years: -**

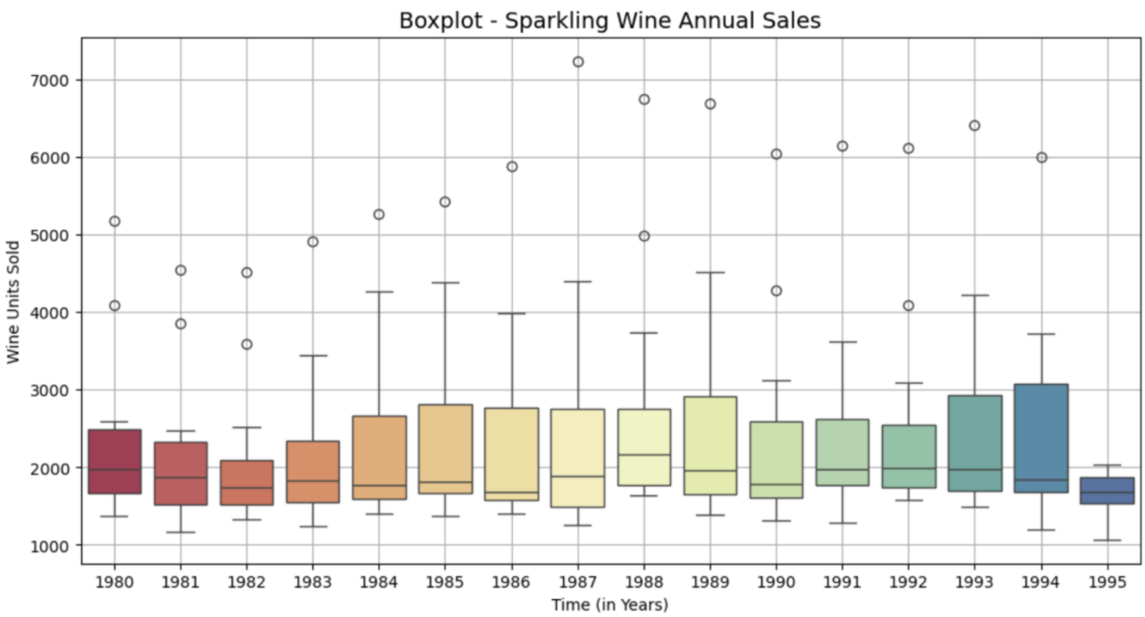


Figure 22: Sparkling | Boxplot of Wine sales across various years

- **Observations: -**
  - Sales of wine have **remained constant** over the years.
  - The presence of **more extreme high outliers in later years** confirms that **peak seasonal demand is increasing.**
  - Sales became **more seasonally polarized**, but the brand maintains a **strong year-round presence.**

▪ **Plot a boxplot to understand the sales at month level: -**

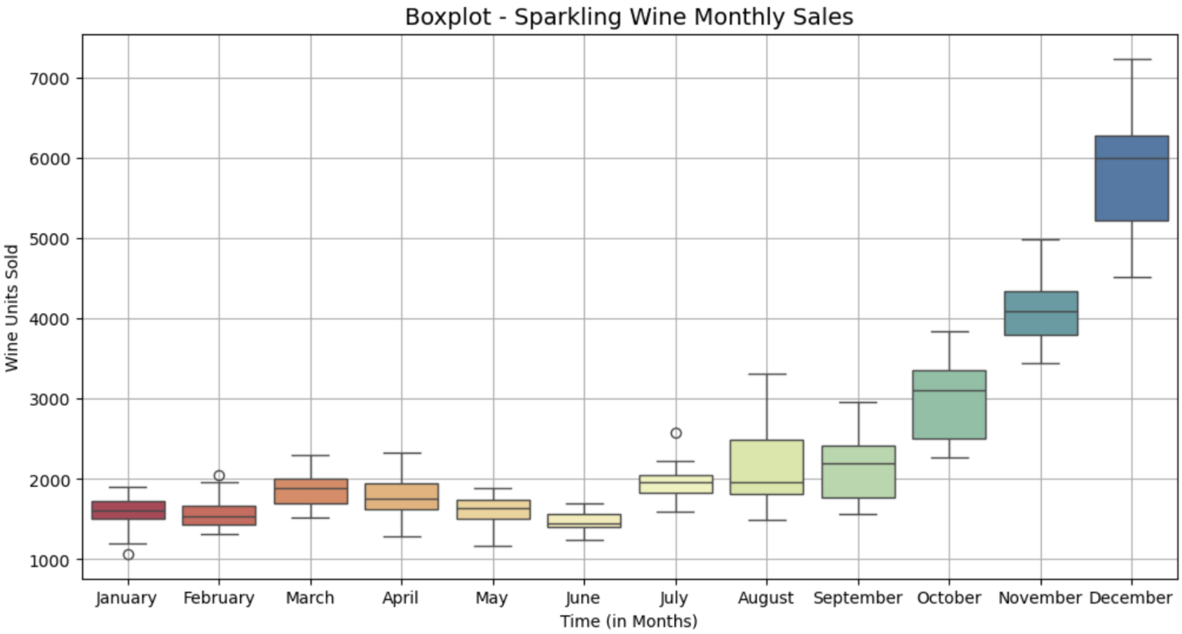


Figure 23: Sparkling | Boxplot – Monthly Wine sales

- **Observations: -**
  - **The pattern is strongly seasonal,** with a sharp ramp-up from **September to December** and a drop back to baseline in January. The transition is quite stark, showing a classic **holiday-driven product cycle**.
  - **Stable, low-variation sales are observed during the rest of the year,** suggesting predictable off-season demand.
  - **Sales are lowest in the early months of the year.**

- **Plot a time series MonthPlot to understand the spread: -**

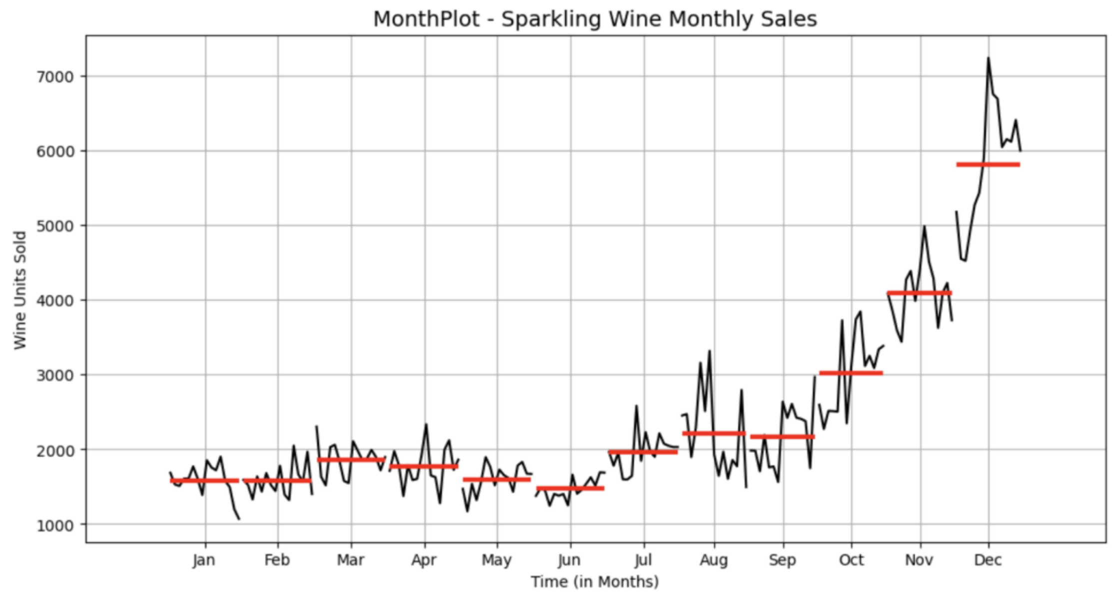


Figure 24: Sparkling | MonthPlot – Monthly Wine sales

- **Observations: -**
  - **MonthPlot** of wine monthly sales is a time series visualization that highlights **seasonality across individual months**. It plots **monthly data points** year by year (thin black lines) and overlays **monthly means** with **thick red horizontal lines**.
  - As clearly evident, **there is strong seasonality in wine sales.**
  - **Wine is a highly seasonal product,** with a clear **spike in December** and **stable, lower demand during other months.**
  - **January to July** suggests stable baseline demand**, August to November** suggests gradual **build-up** & in **December** there is a peak demand.
  - **January to June are lowest-performing and stable months.**
  - **High variability** in the peak month (December) suggesting **sharp and sustained sales spike** (average sales exceed **6,000 units**).

- **Plot a trend for each month sales across all years: -**

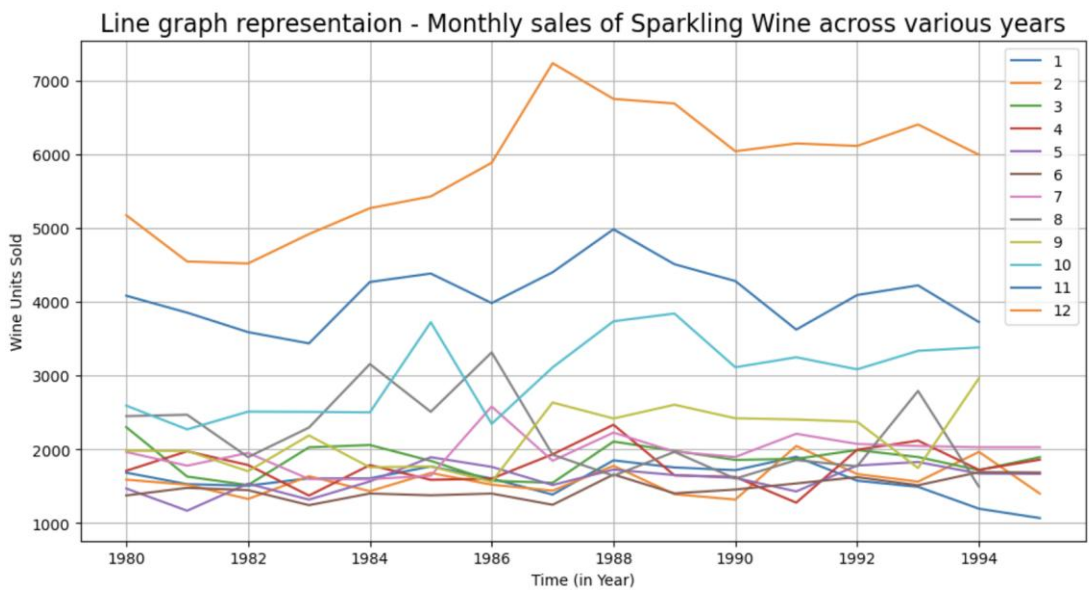


Figure 25: Sparkling | Trend – Each month sales across years

- **Observations: -**
  - Wine sales are **heavily skewed toward Q4**, especially **December.**
  - There's a **year-over-year increase in peak month performance,** while **off-season months remain stable.**
  - **Seasonal effects become less pronounced over time.**
  - As time progresses, monthly lines **converge toward lower values**, indicating **flattening of seasonality** and a **shrinking overall market**.

- **Plot the Empirical Cumulative Distribution Function: -**

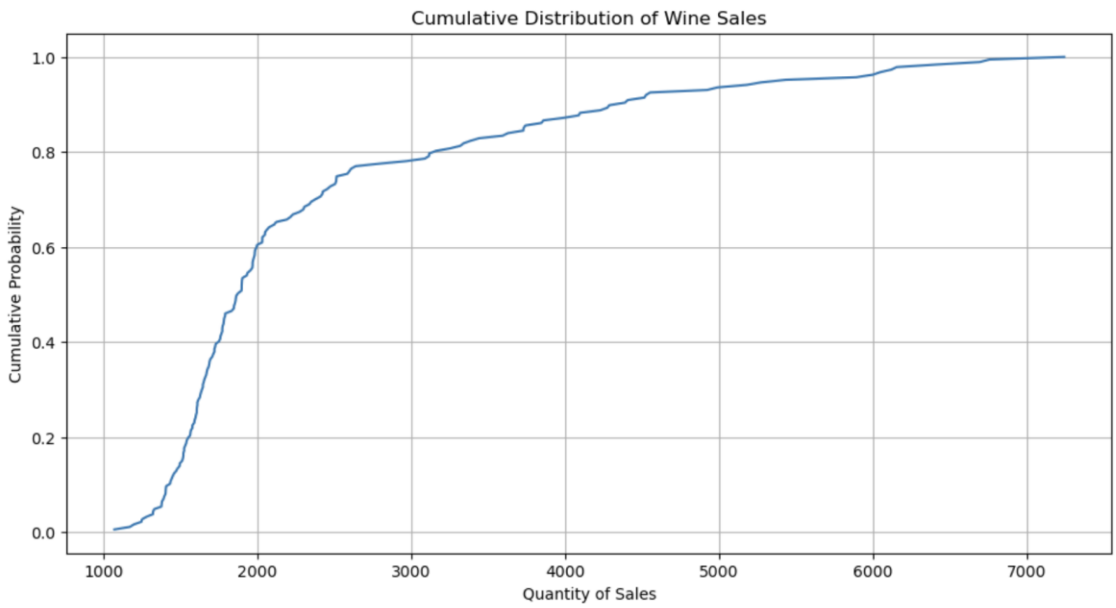


Figure 26: Sparkling | Empirical Cumulative Distribution of Wine sales

- **Observations: -**
  - 80% of the units sold are less than 4000. Only 20% of sales involved more than 3000 items. Therefore, it is clear that the **bulk of sales are in the range of 1000 to 3000 units.**
  - **Strong concentration of months with modest, stable demand**, mostly, **the non-holiday months.**
  - **Top 10–15% of months** had sales over **5,000 units**, reaching up to **7,200 units**. These values most likely correspond to **December** month.
  - The lowest recorded monthly sales were around **1,000 units**, meaning, **even the weakest months had decent sales**, indicating a **strong baseline demand**.

- **Plot the average sales per month and the month-on-month percentage change of sales: -**

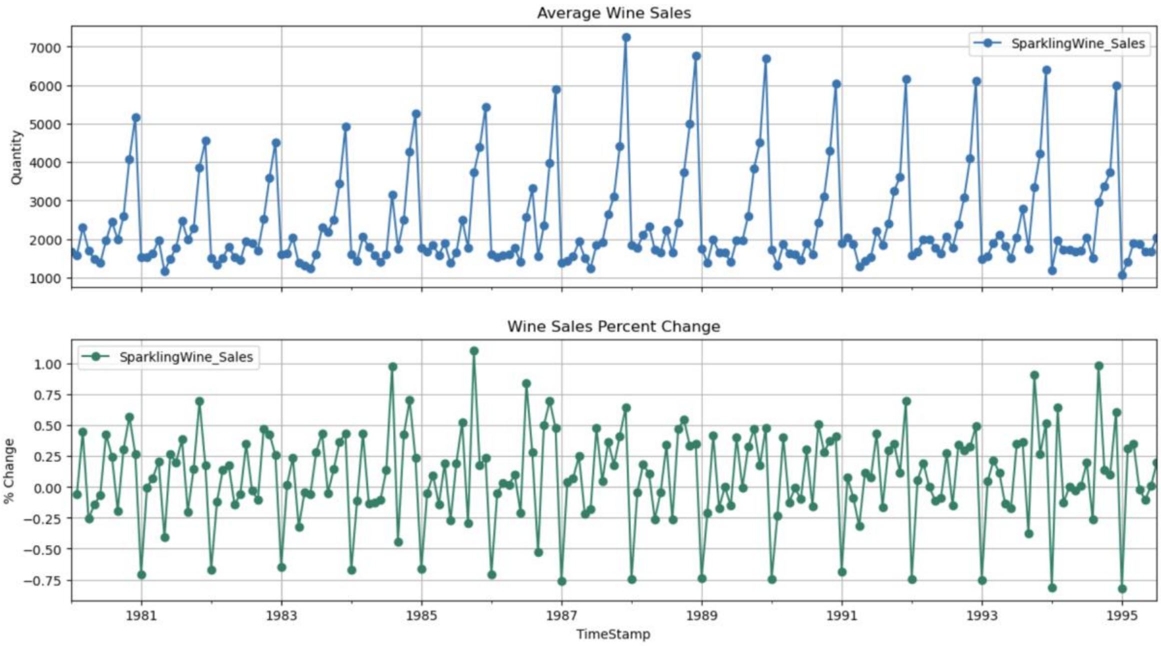


Figure 27: Sparkling | Trend – Average Wine sales & Wine percent change

- **Observations: -**
  - **Sales Trend (Top Plot):** Strong **recurring annual spikes**, consistently occurring around **December. Off-peak months** maintain a **steady baseline. Peak heights show slight upward trend in mid-80s**, but generally remain strong and stable throughout the years. This suggests **robust seasonal demand pattern,** but **not declining** like Rose Wines
  - **Volatility (Bottom Plot): Volatility is seasonal, not erratic.** Sudden swings are predictable, mostly in the month of December.
  - **To summarize: -**
    - ✓ **Trend**: Stable with strong seasonal peaks in sales. Consistent sales with no signs of decline
    - ✓ **Volatility**: High, but predictable and seasonally driven.
    - ✓ **Off-season Stability:** Strong baseline demand (steady throughout year)

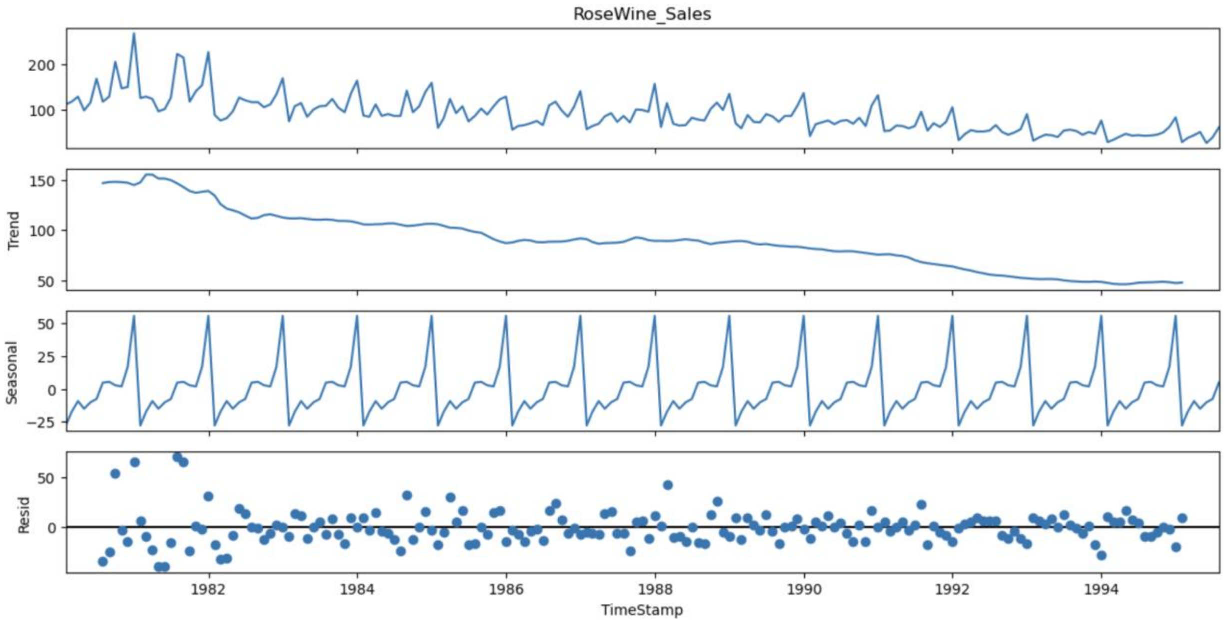
Perform Decomposition – Rose Wine

Additive Decomposition

- Additive decomposition is a method used in time series analysis to break down a time series into three main components, assuming their effects add together linearly: -

  Time Series = Trend + Seasonality + Residual (or Noise)

- Using **seasonal_decompose()** function in python we decompose the times series into 3 components below: -



Trend

TimeStamp	
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

Seasonality

TimeStamp	
1980-01-31	-27.892252
1980-02-29	-17.422041
1980-03-31	-9.268252
1980-04-30	-15.080752
1980-05-31	-10.178966
1980-06-30	-7.661109
1980-07-31	4.803952
1980-08-31	5.426762
1980-09-30	2.791081
1980-10-31	1.888304
1980-11-30	16.863304
1980-12-31	55.729970

Residual

TimeStamp	
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	-33.887286
1980-08-31	-24.551762
1980-09-30	53.833919
1980-10-31	-2.971637
1980-11-30	-14.279970
1980-12-31	66.145030

Figure 28: Rose | Additive Decomposition of time series

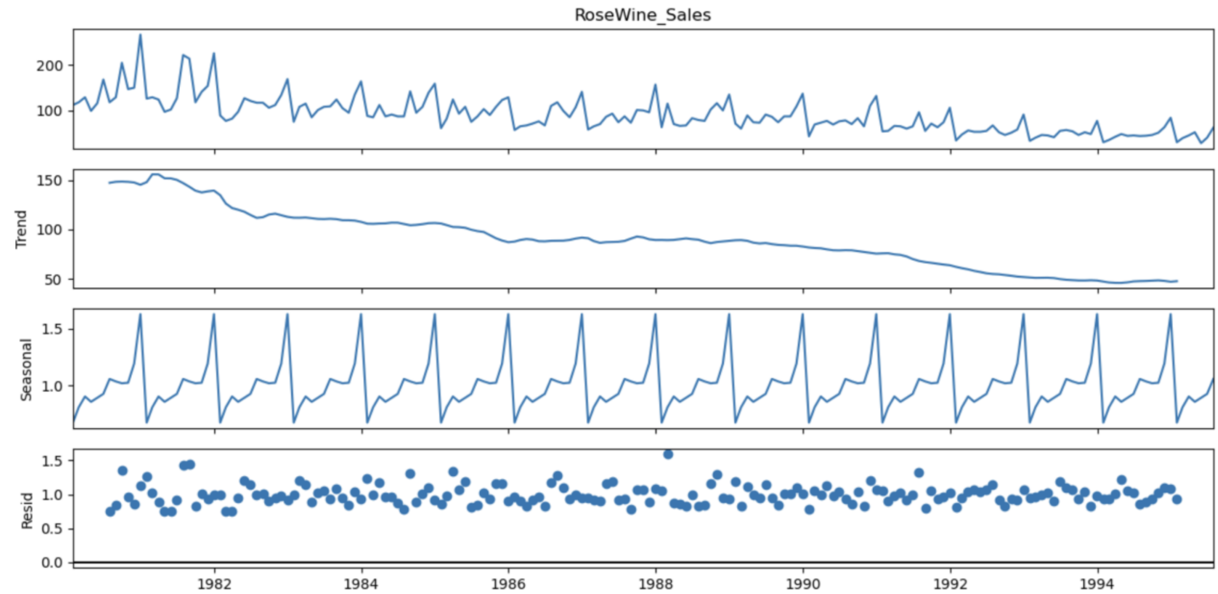
- **Observations: -**
  - **Trend Component:** There is a **clear downward trend**. Starts around 150+ units and steadily declines below 100 by the 1990s i.e. progressive loss in popularity or market share
  - **Seasonal Component:** There is very **clear and consistent seasonality. Peaks every 12 months**, indicating a strong annual cycle (possibly linked to summer or holidays). Height of peaks and troughs remains relatively **constant over time.**
  - **Residual Component: No strong patterns in the residuals,** though there are a few spikes (potential one-off events or anomalies).

Multiplicative Decomposition

- Multiplicative decomposition is a time series analysis method where the components of the series are assumed to interact multiplicatively, rather than additively. It’s useful when seasonal fluctuations grow or shrink in proportion to the trend: -

Time Series = Trend X Seasonality X Residual (or Noise)

- Using **seasonal_decompose()** function in python we decompose the times series into 3 components below: -




```
Trend
 TimeStamp
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
```

```
Seasonality
 TimeStamp
1980-01-31    0.670320
1980-02-29    0.806375
1980-03-31    0.901501
1980-04-30    0.854411
1980-05-31    0.889760
1980-06-30    0.924324
1980-07-31    1.056071
1980-08-31    1.034338
1980-09-30    1.017959
1980-10-31    1.022915
1980-11-30    1.192781
1980-12-31    1.629244
```

```
Residual
 TimeStamp
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    0.759671
1980-08-31    0.841974
1980-09-30    1.357259
1980-10-31    0.970446
1980-11-30    0.853069
1980-12-31    1.129231
```

Figure 29: Rose | Multiplicative Decomposition of time series

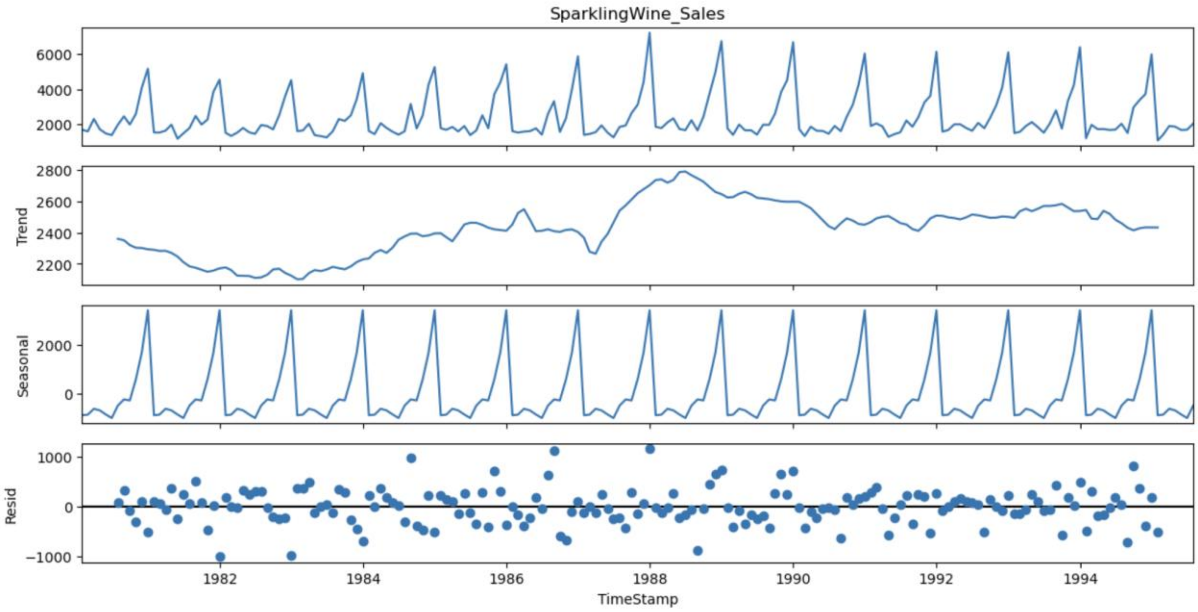
- **Observations: -**
  - **Trend Component:** Similar **declining trend,** as in the additive model, with sales drop from around 150 units to below 100, over time.
  - **Seasonal Component: Seasonal peaks and dips remain consistent** in pattern but apply **proportionally** to the trend.
  - **Residual Component: No strong patterns in the residuals,** though it has been observed that the **seasonal fluctuation of residuals is under control, relative to additive decomposition.**
  - The size of the seasonal variations doesn't change on comparison, but the **residuals are tightly controlled by the multiplicative** decomposition.

Perform Decomposition – Sparkling Wine

Additive Decomposition

- Additive decomposition is a method used in time series analysis to break down a time series into three main components, assuming their effects add together linearly: -

  Time Series = Trend + Seasonality + Residual (or Noise)
- Using **seasonal_decompose()** function in python we decompose the times series into 3 components below: -



Trend

TimeStamp	
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

Seasonality

TimeStamp	
1980-01-31	-854.260599
1980-02-29	-830.350678
1980-03-31	-592.356630
1980-04-30	-658.490559
1980-05-31	-824.416154
1980-06-30	-967.434011
1980-07-31	-465.502265
1980-08-31	-214.332821
1980-09-30	-254.677265
1980-10-31	599.769957
1980-11-30	1675.067179
1980-12-31	3386.983846

Residual

TimeStamp	
1980-01-31	NaN
1980-02-29	NaN
1980-03-31	NaN
1980-04-30	NaN
1980-05-31	NaN
1980-06-30	NaN
1980-07-31	70.835599
1980-08-31	315.999487
1980-09-30	-81.864401
1980-10-31	-307.353290
1980-11-30	109.891154
1980-12-31	-501.775513

Figure 30: Sparkling | Additive Decomposition of time series

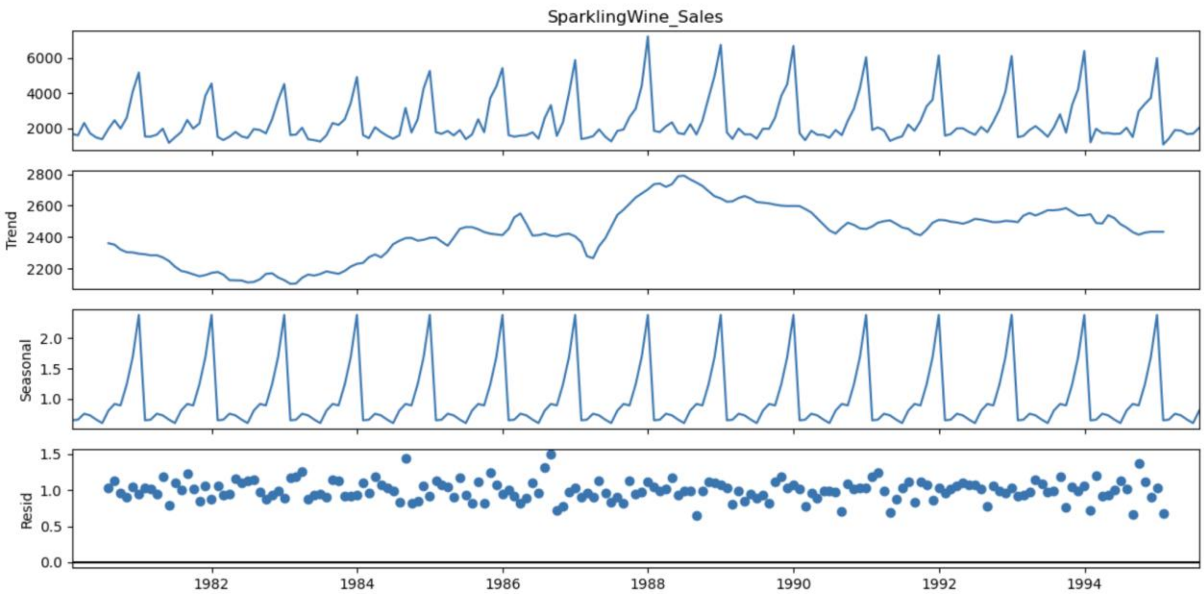
- **Observations: -**
  - **Trend Component:** There seems to be slight growth in initial years and then **plateauing in baseline.**
  - **Seasonal Component:** There is very **clear and consistent seasonality. One major peak (likely December),** with a **gradual rise** starting around **September–October.**
  - **Residual Component:** There seems to be some **significant scatter,** especially in early and peak months, suggesting that that additive decomposition might **not fully capture** the increasing peak amplitudes over time.

Multiplicative Decomposition

- Multiplicative decomposition is a time series analysis method where the components of the series are assumed to interact multiplicatively, rather than additively. It’s useful when seasonal fluctuations grow or shrink in proportion to the trend: -

Time Series = Trend X Seasonality X Residual (or Noise)

- Using **seasonal_decompose()** function in python we decompose the times series into 3 components below: -



Trend TimeStamp		Seasonality TimeStamp		Residual TimeStamp	
1980-01-31	NaN	1980-01-31	0.649843	1980-01-31	NaN
1980-02-29	NaN	1980-02-29	0.659214	1980-02-29	NaN
1980-03-31	NaN	1980-03-31	0.757440	1980-03-31	NaN
1980-04-30	NaN	1980-04-30	0.730351	1980-04-30	NaN
1980-05-31	NaN	1980-05-31	0.660609	1980-05-31	NaN
1980-06-30	NaN	1980-06-30	0.603468	1980-06-30	NaN
1980-07-31	2360.666667	1980-07-31	0.809164	1980-07-31	1.029230
1980-08-31	2351.333333	1980-08-31	0.918822	1980-08-31	1.135407
1980-09-30	2320.541667	1980-09-30	0.894367	1980-09-30	0.955954
1980-10-31	2303.583333	1980-10-31	1.241789	1980-10-31	0.907513
1980-11-30	2302.041667	1980-11-30	1.690158	1980-11-30	1.050423
1980-12-31	2293.791667	1980-12-31	2.384776	1980-12-31	0.946770

Figure 31: Sparkling | Multiplicative Decomposition of time series

- **Observations: -**
  - **Trend Component:** Similar **plateauing in baseline** observed, as in the additive model.
  - **Seasonal Component: Seasonal peaks and dips remain consistent**. In high-trend periods, sales **increase proportionally** during seasonal peaks; while, in low-trend periods, seasonal effects **shrink proportionally**.
  - **Residual Component: Less scatter than the additive residuals**, especially, in high-variance months.
  - The **residuals are relatively less scattered in the multiplicative** decomposition, indicating that the **multiplicative** model does a **better job isolating trend and seasonality**, leaving less unexplained variation.

Rubric Question 2: Data Preprocessing

Missing-value Check & Treatment – Rose Wine

- Please refer the link – [Check & Treat Missing/Null Values]() section.
- 2 missing values observed which we treated using Spline Interpolation.

Missing-value Check & Treatment – Sparkling Wine

- Please refer the link – [Check & Treat Missing/Null Values]() section.
- No missing values observed.

Visualize the Processed Data – Rose/Sparkling Wine

- Please refer the link – [Exploratory Data Analysis – Rose Wine]() section for Rose Wine.
- Please refer the link – [Exploratory Data Analysis – Sparkling Wine]() section for Sparkling Wine.

Train-Test Split – Rose/Sparkling Wine

- Train and Test data are split in the ratio of 70:30. Data up to 1991 is included in the training data, while data from 1991 through 1995 is used for testing.
- Below are the top & bottom 5 rows for both, Rose & Sparkling wine, Train & Test datasets below: -

First few rows of Training Data

TimeStamp	RoseWine_Sales
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

First few rows of Test Data

TimeStamp	RoseWine_Sales
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

First few rows of Training Data

TimeStamp	SparklingWine_Sales
1980-01-31	1686
1980-02-29	1591
1980-03-31	2304
1980-04-30	1712
1980-05-31	1471

First few rows of Test Data

TimeStamp	SparklingWine_Sales
1991-01-31	1902
1991-02-28	2049
1991-03-31	1874
1991-04-30	1279
1991-05-31	1432

Last few rows of Training Data

TimeStamp	RoseWine_Sales
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

Last few rows of Test Data

TimeStamp	RoseWine_Sales
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

Last few rows of Training Data

TimeStamp	SparklingWine_Sales
1990-08-31	1605
1990-09-30	2424
1990-10-31	3116
1990-11-30	4286
1990-12-31	6047

Last few rows of Test Data

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

Figure 32: Rose & Sparkling | Train-Test Data Split

- Below is the shape of Test-Train datasets for both Rose & Sparkling Wine: -

```
Shape of Train data:  (132, 1)
Shape of Test data:  (55, 1)
Total Observations in dataset:  187
```

Figure 33: Rose | Shape of Train-Test datasets

```
Shape of Train data:  (132, 1)
Shape of Test data:  (55, 1)
Total Observations in dataset:  187
```

Figure 34: Sparkling | Shape of Train-Test datasets

- Below is LinePlot capturing the Train-Test Split for both Rose & Sparkling Wine: -

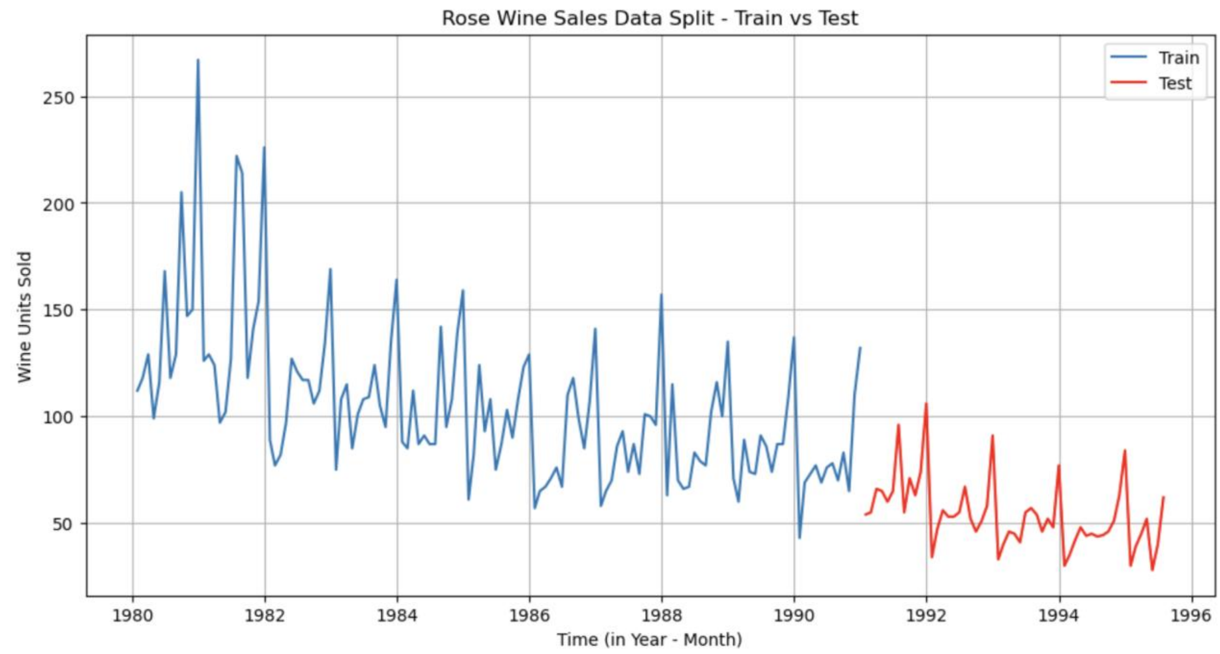


Figure 35: Rose | LinePlot of Train-Test data split

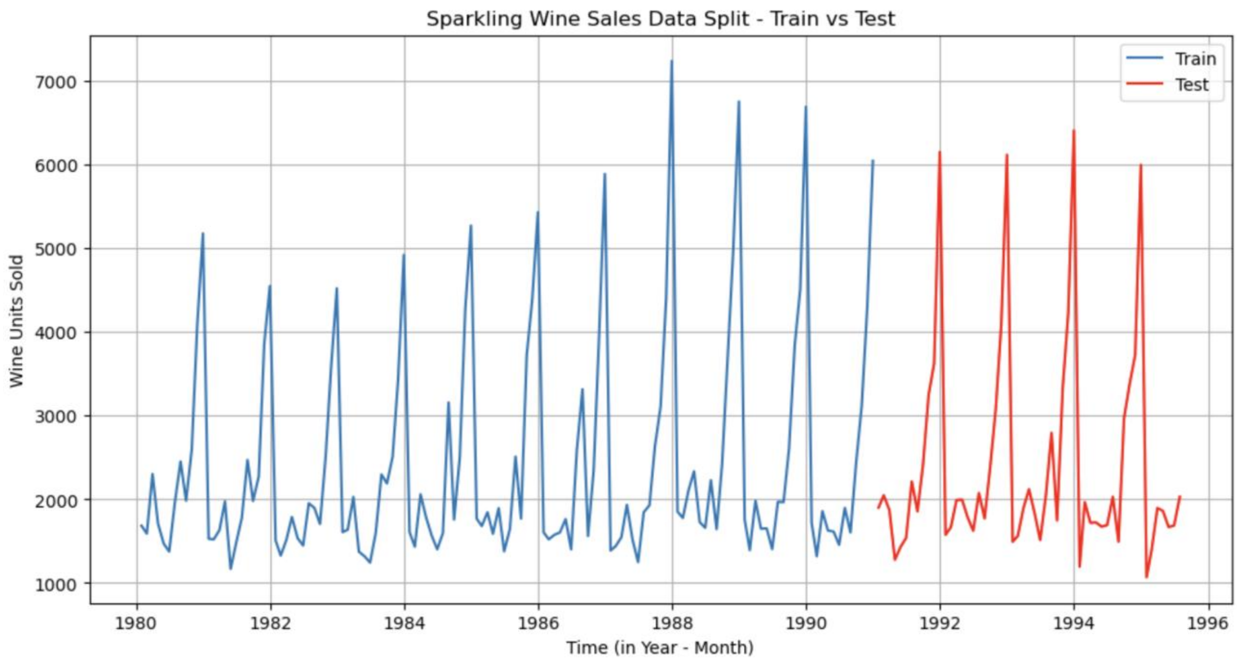


Figure 36: Sparkling | LinePlot of Train-Test data split

Rubric Question 3: Model Building – Original Data

Linear Regression – Rose Wine

- Let's regress the 'RoseWine_Sales' variable against the order of the occurrence.
- Below Linear Regression model is built: -

```
▾ LinearRegression
LinearRegression()
```

Figure 37: Rose | Linear Regression Model

- Below are the predicted values on the top 5 rows of test dataset: -

TimeStamp	RoseWine_Sales	Time	RegressionOnTime_pred
1991-01-31	54.0	132	72.557644
1991-02-28	55.0	133	72.063266
1991-03-31	66.0	134	71.568888
1991-04-30	65.0	135	71.074511
1991-05-31	60.0	136	70.580133

Figure 38: Rose | Predicted values of test dataset

- Below is the plot of regressed predicted values against train & test dataset: -

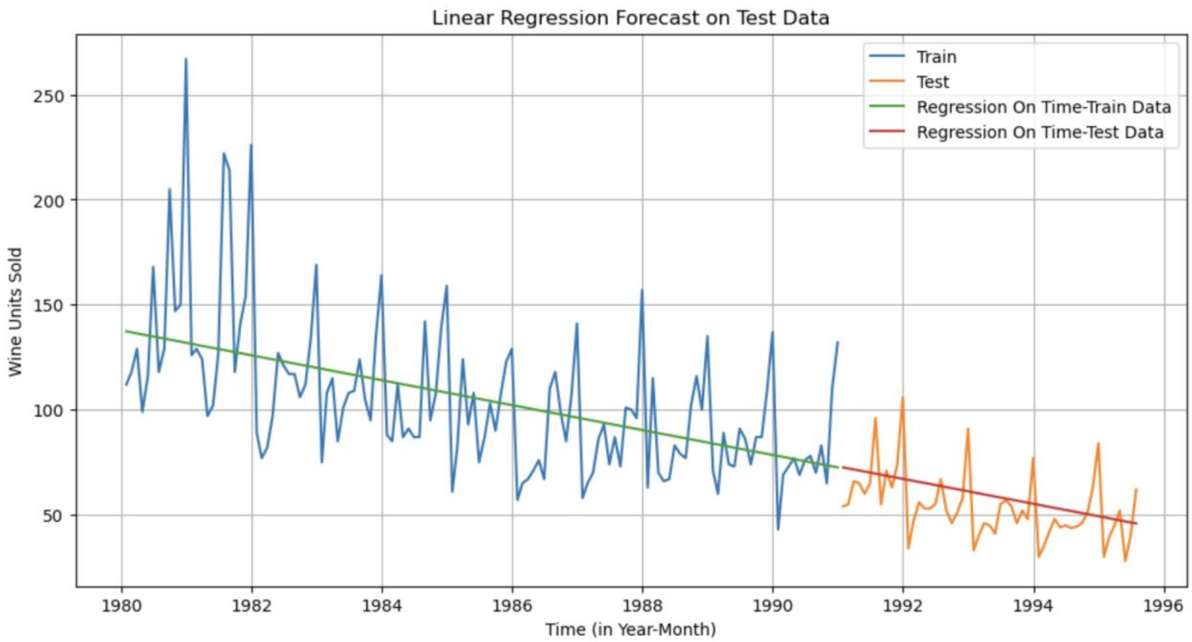


Figure 39: Rose | Linear Regression Plot on Train-Test dataset

- **Model Performance** – Below is the RMSE value of the Linear Regression Model: -

```
For RegressionOnTime forecast on the Test Data,  RMSE is 15.457
```

Figure 40: Rose | Linear Regression Model RMSE

- **Observations**: -
  - Clearly, the model has a **forecasted a falling trend**.
  - Though, the trends have been picked up by the model, but it **failed to forecast the seasonality** in the model.
  - The root means squared error (**RMSE**) for the linear regression model is **15.457.**

Linear Regression – Sparkling Wine

- Let's regress the 'SparklingWine_Sales' variable against the order of the occurrence.
- Below Linear Regression model is built: -

```
▾ LinearRegression
LinearRegression()
```

Figure 41: Sparkling | Linear Regression Model

- Below are the predicted values on the top 5 rows of test dataset: -

TimeStamp	SparklingWine_Sales	Time	RegressionOnTime_pred
1991-01-31	1902	132	2785.819435
1991-02-28	2049	133	2791.652093
1991-03-31	1874	134	2797.484752
1991-04-30	1279	135	2803.317410
1991-05-31	1432	136	2809.150069

Figure 42: Sparkling | Predicted values of test dataset

- Below is the plot of regressed predicted values against train & test dataset: -

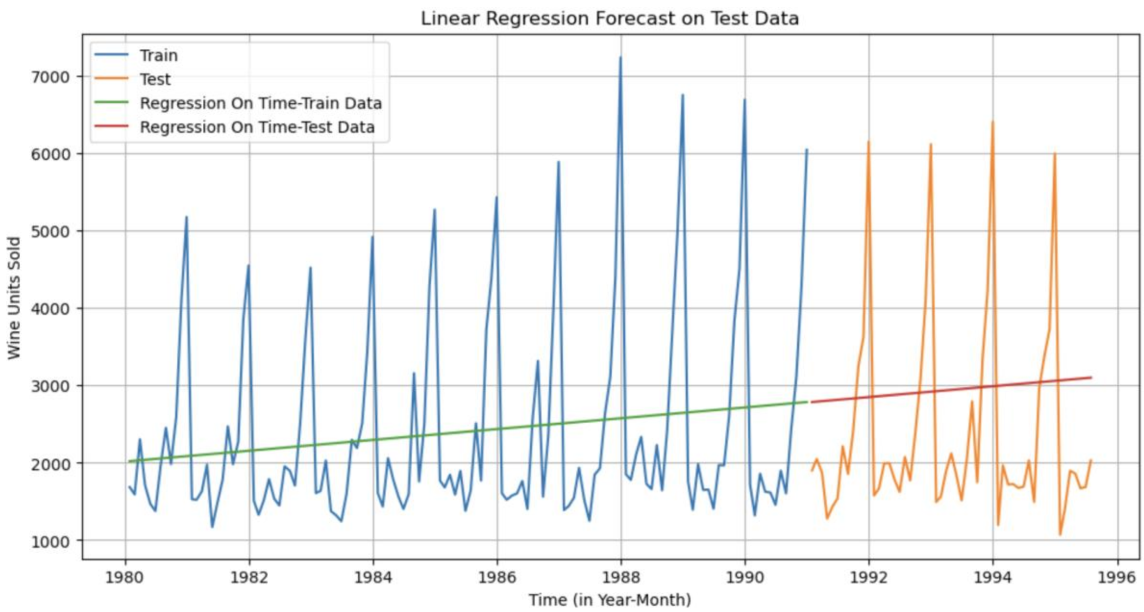


Figure 43: Sparkling | Linear Regression Plot on Train-Test dataset

- **Model Performance** – Below is the RMSE value of the Linear Regression Model: -

```
For RegressionOnTime forecast on the Test Data,  RMSE is 1386.836
```

Figure 44: Sparkling | Linear Regression Model RMSE

- **Observations**: -
  - Clearly, the model has a **forecasted a slightly upward trend**.
  - Though, the trends have been picked up by the model, but it **failed to forecast the seasonality** in the model.
  - The root means squared error (**RMSE**) for the linear regression model is **1386.836.**

Naive Forecast – Rose Wine

- In this model, the prediction for tomorrow is the same as today and the prediction for day after tomorrow is tomorrow and since the prediction of tomorrow is same as today, therefore the prediction for day after tomorrow is also today.
- Below are the predicted values on the top 5 rows of test dataset :-

TimeStamp	RoseWine_Sales	NaiveModel_pred
1991-01-31	54.0	132.0
1991-02-28	55.0	132.0
1991-03-31	66.0	132.0
1991-04-30	65.0	132.0
1991-05-31	60.0	132.0

Figure 45: Rose | Predicted values of test dataset

- Below is the plot of the forecasted values against the train dataset: -

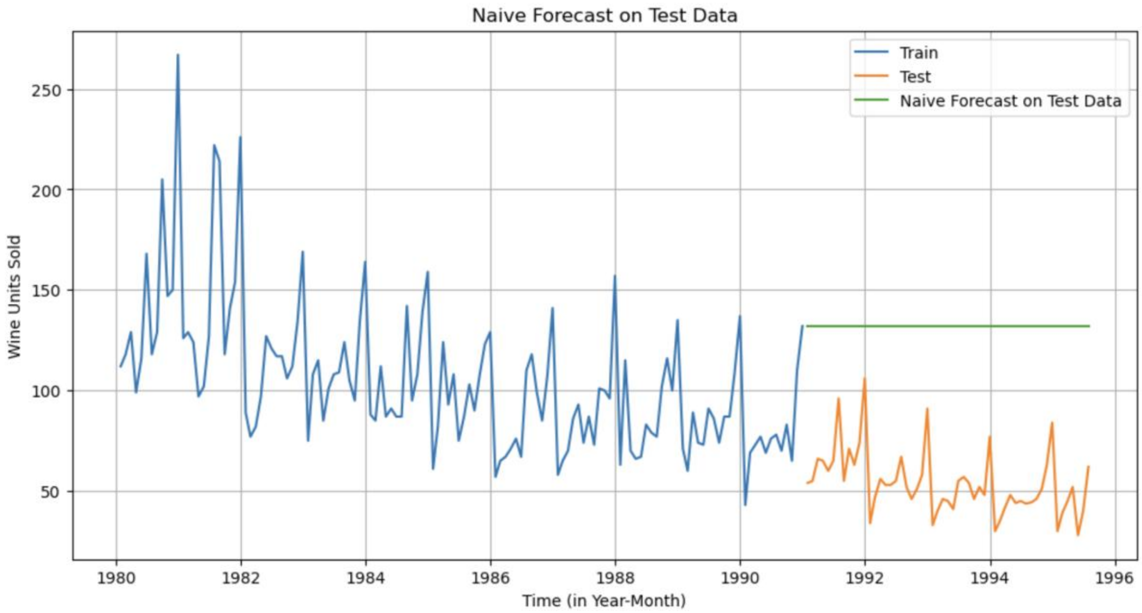


Figure 46: Rose | Naive Forecast Plot on Test dataset

- **Model Performance** – Below is the RMSE value of the Naive Forecast Model: -

```
For Naive forecast on the Test Data,  RMSE is 79.778
```

Figure 47: Rose | Naive Forecast RMSE

- **Observations**: -
  - Clearly, the model has **failed to forecast, both seasonality & trend**, with this model.
  - The root means squared error (**RMSE**) for the Naive Forecast model is **79.778**, which is significantly higher than the Linear Regression Model due to obvious reasons as highlighted above.

Naive Forecast – Sparkling Wine

- In this model, the prediction for tomorrow is the same as today and the prediction for day after tomorrow is tomorrow and since the prediction of tomorrow is same as today, therefore the prediction for day after tomorrow is also today.
- Below are the predicted values on the top 5 rows of test dataset :-

TimeStamp	SparklingWine_Sales	NaiveModel_pred
1991-01-31	1902	6047
1991-02-28	2049	6047
1991-03-31	1874	6047
1991-04-30	1279	6047
1991-05-31	1432	6047

Figure 48: Sparkling | Predicted values of test dataset

- Below is the plot of the forecasted values against the train dataset: -

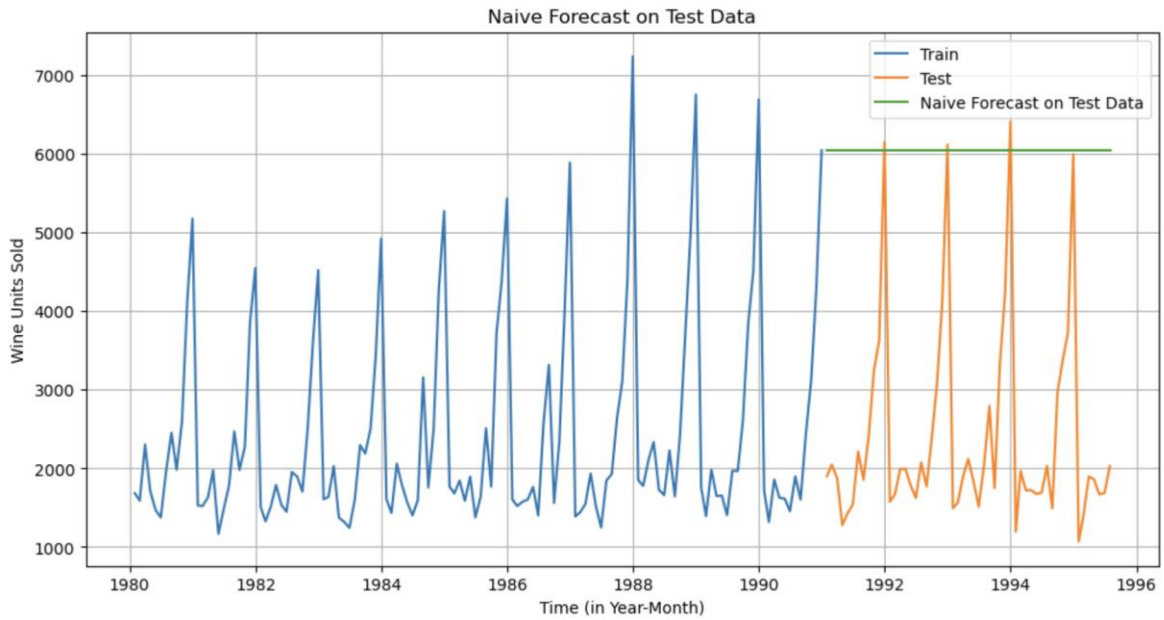


Figure 49: Sparkling | Naive Forecast Plot on Test dataset

- **Model Performance** – Below is the RMSE value of the Naive Forecast Model: -

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

Figure 50: Sparkling | Naive Forecast RMSE

- **Observations**: -
  - Clearly, the model has **failed to forecast, both seasonality & trend**, with this model.
  - The root means squared error (**RMSE**) for the Naive Forecast model is **3864.279**, which is significantly higher than the Linear Regression Model due to obvious reasons as highlighted above.

Simple Average Model – Rose Wine

- In this model, we forecast by using the average of the training values.
- Below are the predicted values on the top 5 rows of test dataset :-

TimeStamp	RoseWine_Sales	SimpleAverage_pred
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

Figure 51: Rose | Predicted values of test dataset

- Below is the plot of the forecasted values against the train dataset: -

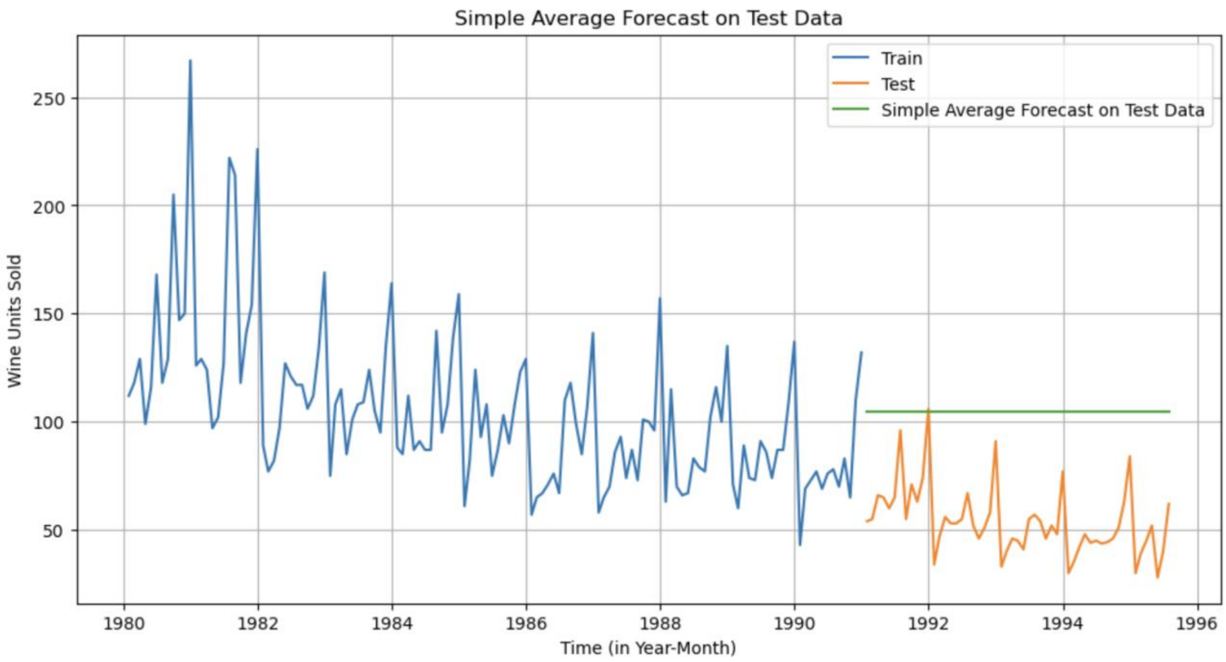


Figure 52: Rose | Simple Average Forecast Plot on Test dataset

- **Model Performance** – Below is the RMSE value of the Simple Average Forecast Model: -

```
For Simple Average forecast on the Test Data,  RMSE is 53.522
```

Figure 53: Rose | Simple Average Forecast RMSE

- **Observations**: -
  - Clearly, the model has **failed to forecast, both seasonality & trend**, with this model.
  - The root means squared error (**RMSE**) for the Simple Average Forecast model is **53.522**, which is significantly higher than the Linear Regression Model (but less than Naive Forecast) due to obvious reasons as highlighted above.

Simple Average Model – Sparkling Wine

- In this model, we forecast by using the average of the training values.
- Below are the predicted values on the top 5 rows of test dataset :-

TimeStamp	SparklingWine_Sales	SimpleAverage_pred
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

Figure 54: Sparkling | Predicted values of test dataset

- Below is the plot of the forecasted values against the train dataset: -

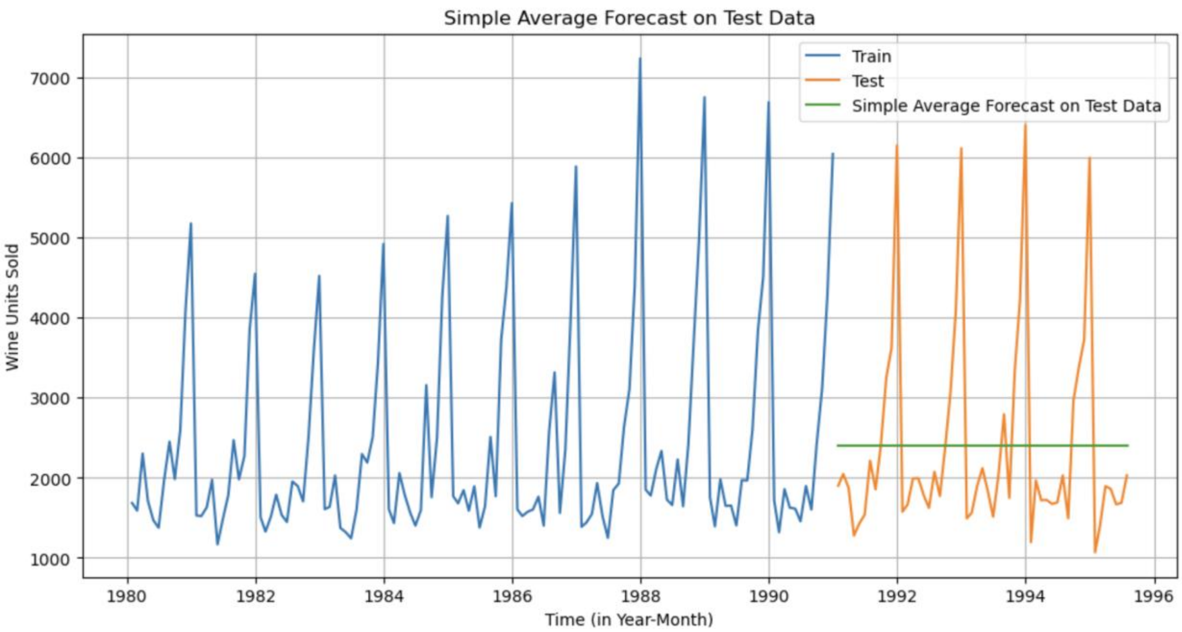


Figure 55: Sparkling | Simple Average Forecast Plot on Test dataset

- **Model Performance** – Below is the RMSE value of the Simple Average Forecast Model: -

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

Figure 56: Sparkling | Simple Average Forecast RMSE

- **Observations**: -
  - Clearly, the model has **failed to forecast, both seasonality & trend**, with this model.
  - There may (or may not) be a marginally upward trend but the model shows no trend which coincidently matches with the overall trend.
  - The root means squared error (**RMSE**) for the Simple Average Forecast model is **1275.082**, which is lower than both, Linear Regression Model & Naive Forecast Model, due to reasons highlighted above.

Moving Average Model – Rose Wine

- In this model, we calculate the rolling means (or moving averages) for different intervals. The best interval can be determined by the maximum accuracy (or the minimum error).
- Below are the predicted values on the top 5 rows of test dataset for various intervals (2, 4, 6, 9) :-

TimeStamp	RoseWine_Sales	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	123.666667	NaN
1980-07-31	118.0	143.0	125.25	124.666667	NaN
1980-08-31	129.0	123.5	132.75	126.500000	NaN
1980-09-30	205.0	167.0	155.00	139.166667	132.666667

Figure 57: Rose | Predicted values of test dataset

- Below is the plot of the Moving Average values against the entire dataset: -

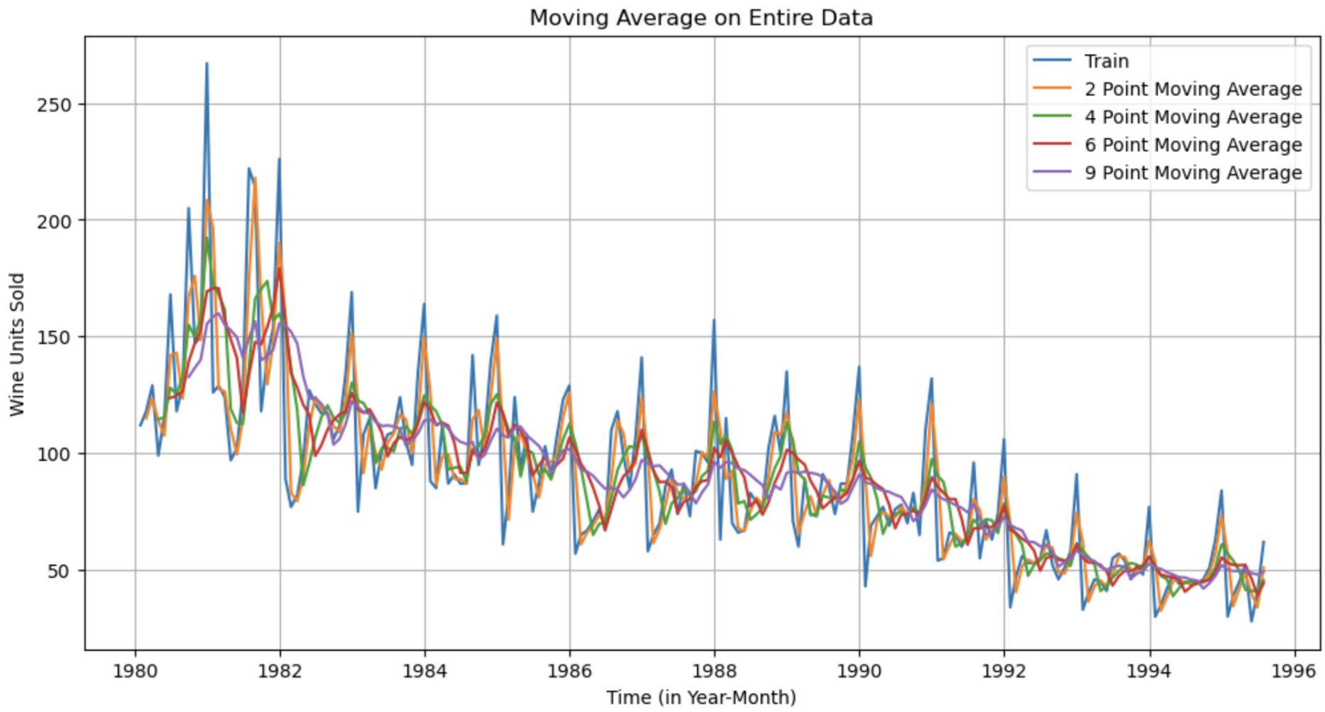


Figure 58: Rose | Moving Average values against the entire dataset

- For better visibility, we plot below the Moving Average values against the entire dataset separately for each interval: -

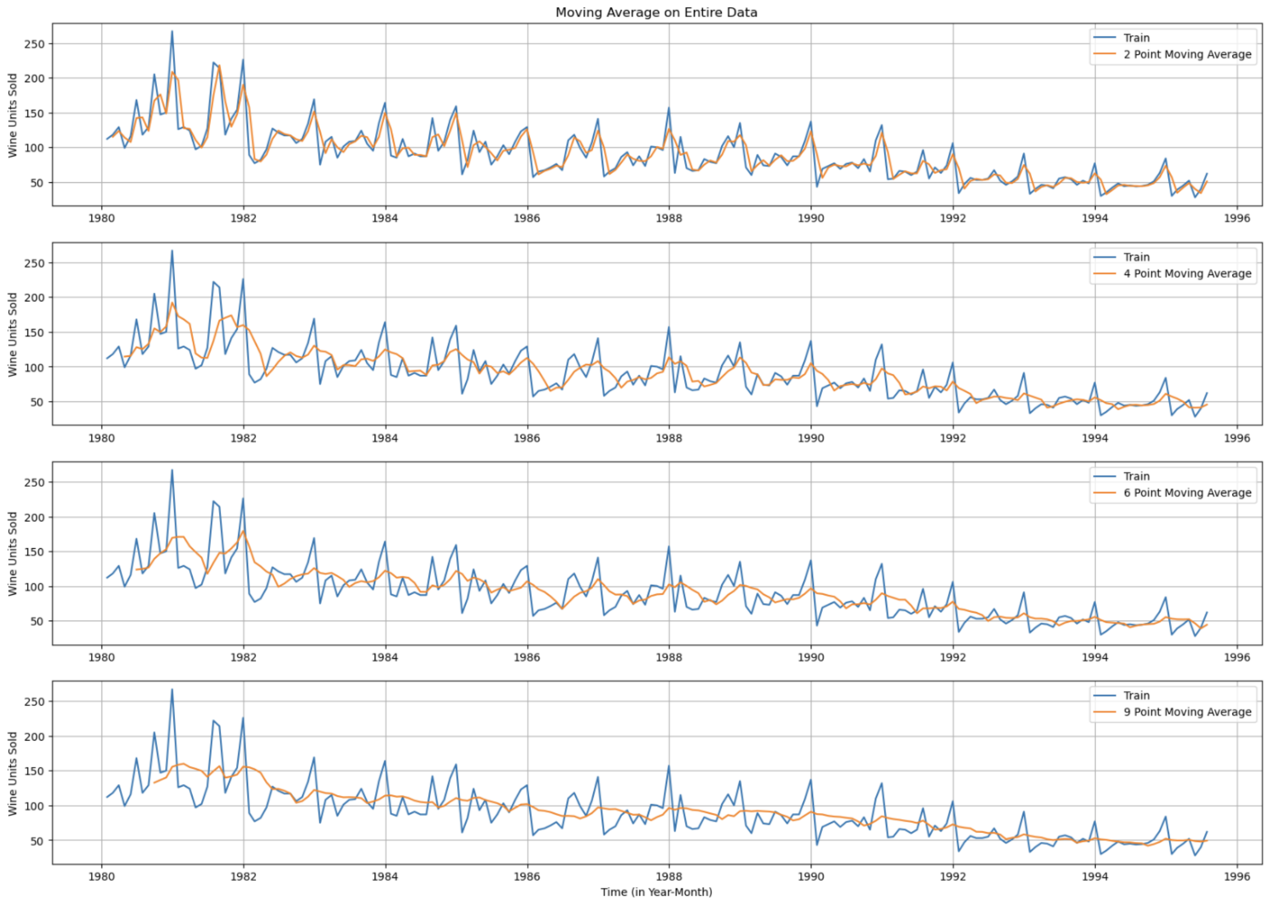


Figure 59: Rose | Moving Average values against the entire dataset (by Interval)

- Below is the plot of the forecasted values against the train dataset for different intervals: -

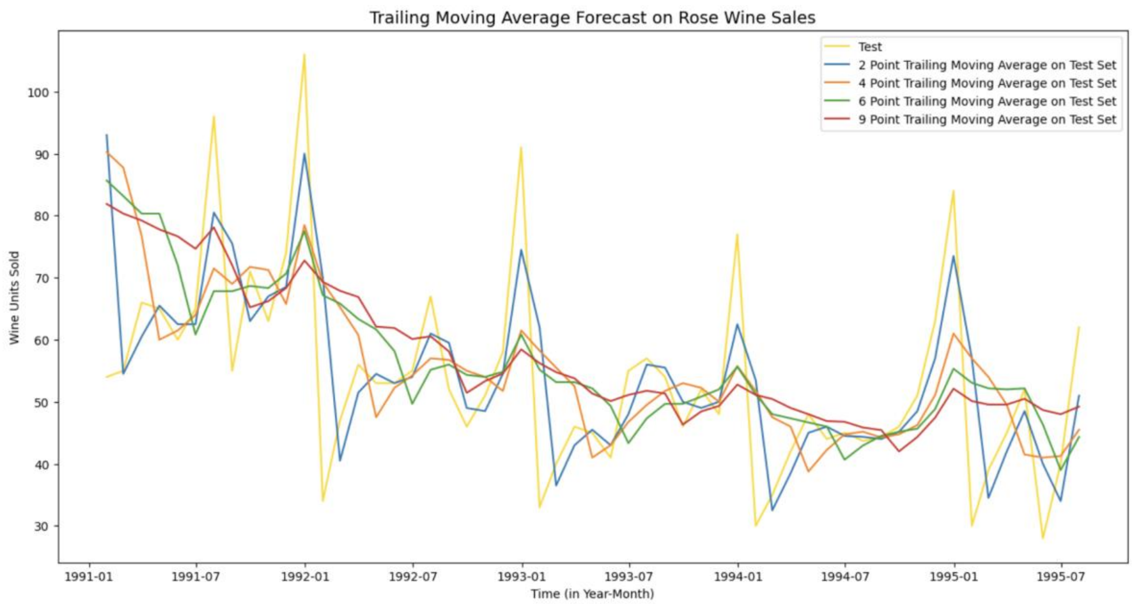


Figure 60: Rose | Moving Average Forecast Plot on Test dataset (various intervals)

- **Model Performance** – Below are the RMSE values for various intervals of the Moving Average Forecast Model: -

```
For 2 point Moving Average Model forecast on the Training Data,  RMSE is 11.530
For 4 point Moving Average Model forecast on the Training Data,  RMSE is 14.462
For 6 point Moving Average Model forecast on the Training Data,  RMSE is 14.587
For 9 point Moving Average Model forecast on the Training Data,  RMSE is 14.740
```

Figure 61: Rose | Moving Average Forecast RMSE (various intervals)

- **Observations**: -
  - **2-point trailing moving average model has the lowest RMSE** against all other intervals. Hence, we choose 2-point trailing moving average model for further evaluation.
  - The model was able to capture, **both seasonality & trend**, in the forecast. Seems to be better than the models built so far.
  - The root means squared error (**RMSE**) for the 2-Point Trailing Moving Average Forecast model is **11.530**, which is **lowest** than all the **models built so far**.

Moving Average Model – Sparkling Wine

- In this model, we calculate the rolling means (or moving averages) for different intervals. The best interval can be determined by the maximum accuracy (or the minimum error).
- Below are the predicted values on the top 5 rows of test dataset for various intervals (2, 4, 6, 9) :-

TimeStamp	SparklingWine_Sales	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	1690.166667	NaN
1980-07-31	1966	1671.5	1631.50	1736.833333	NaN
1980-08-31	2453	2209.5	1816.75	1880.500000	NaN
1980-09-30	1984	2218.5	1945.00	1827.166667	1838.222222

Figure 62: Sparkling | Predicted values of test dataset

- Below is the plot of the Moving Average values against the entire dataset: -

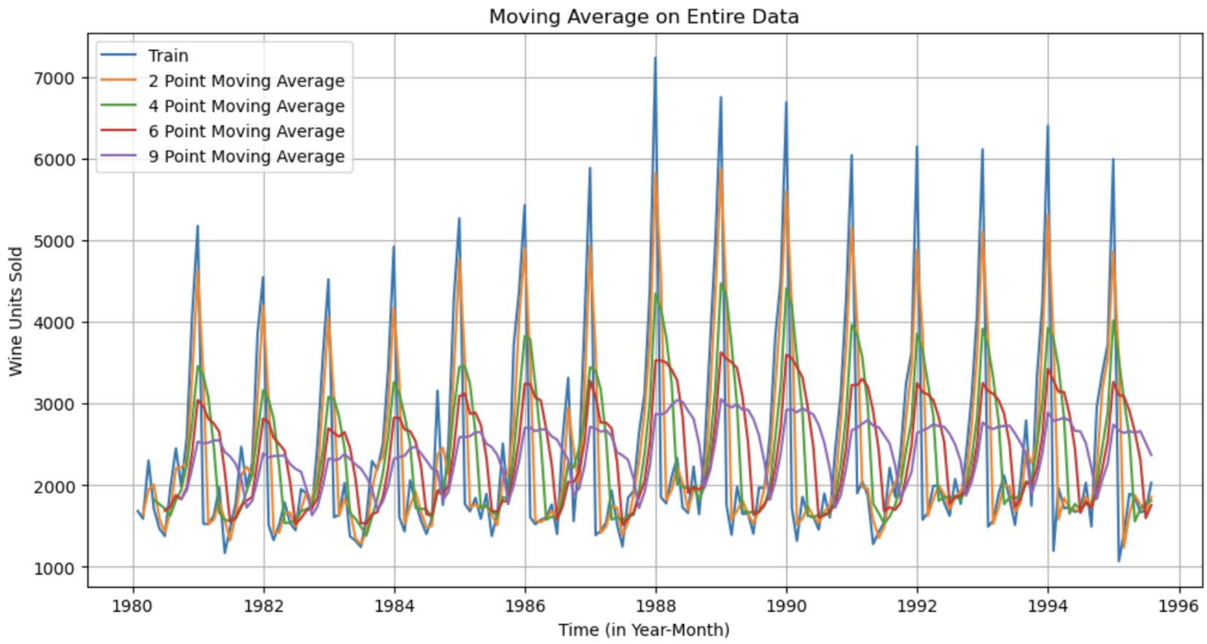


Figure 63: Sparkling | Moving Average values against the entire dataset

- For better visibility, we plot below the Moving Average values against the entire dataset separately for each interval: -

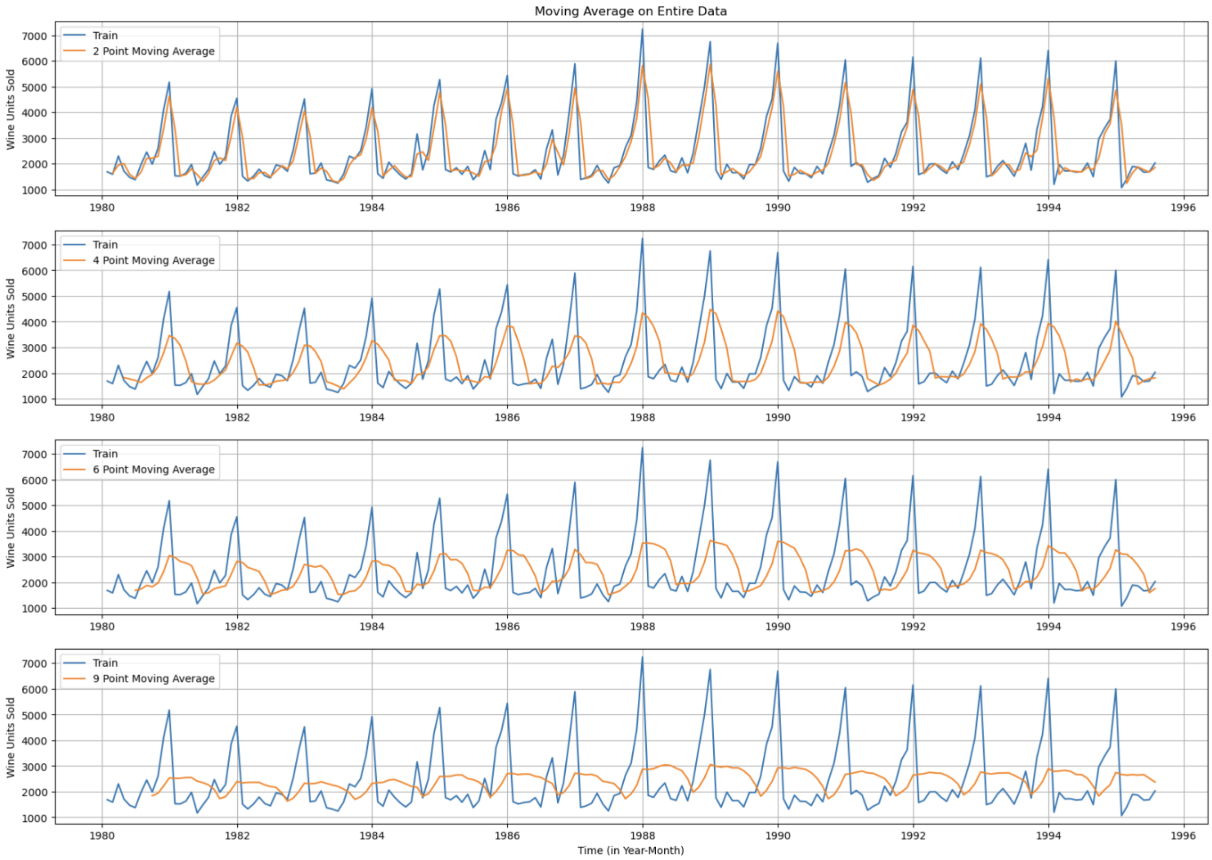


Figure 64: Sparkling | Moving Average values against the entire dataset (by Interval)

- Below is the plot of the forecasted values against the train dataset for different intervals: -

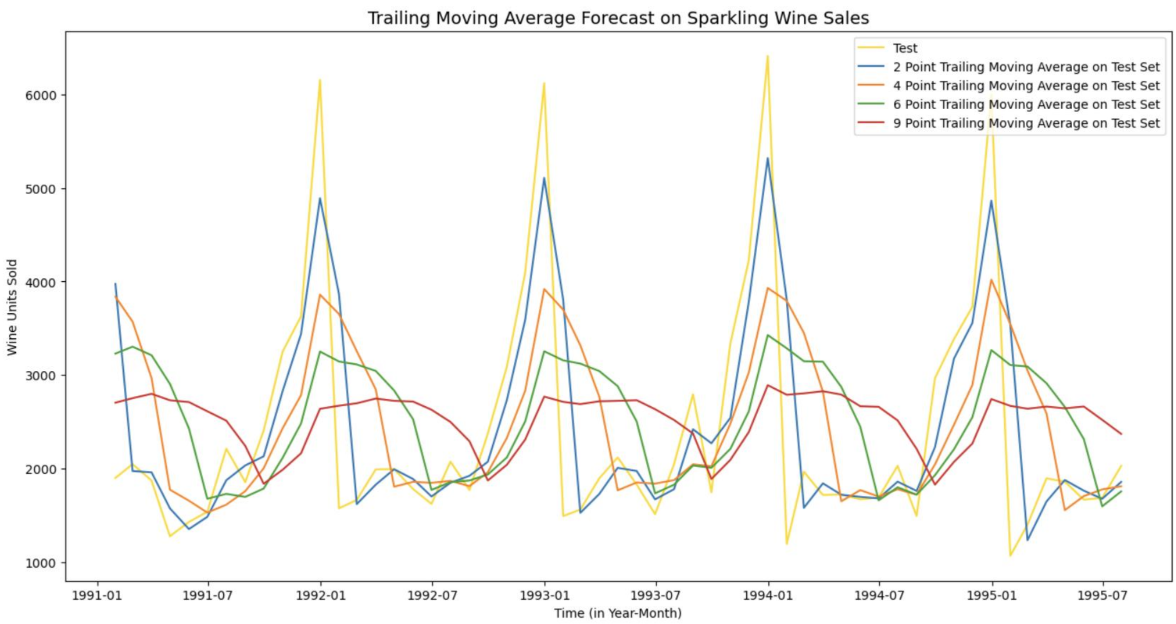


Figure 65: Sparkling | Moving Average Forecast Plot on Test dataset (various intervals)

- **Model Performance** – Below are the RMSE values for various intervals of the Moving Average Forecast Model: -

```
For 2 point Moving Average Model forecast on the Training Data,  RMSE is 813.401
For 4 point Moving Average Model forecast on the Training Data,  RMSE is 1156.590
For 6 point Moving Average Model forecast on the Training Data,  RMSE is 1283.927
For 9 point Moving Average Model forecast on the Training Data,  RMSE is 1346.278
```

Figure 66: Sparkling | Moving Average Forecast RMSE (various intervals)

- **Observations**: -
  - **2-point trailing moving average model has the lowest RMSE** against all other intervals. Hence, we choose 2-point trailing moving average model for further evaluation.
  - The model was able to capture, **both seasonality & trend**, in the forecast. Seems to be better than the models built so far.
  - The root means squared error (**RMSE**) for the 2-Point Trailing Moving Average Forecast model is **813.401**, which is **lowest** than all the **models built so far**.

Comparison Plots for Simple Models – Rose/Sparkling Wine

- Before we go ahead & built slightly more advanced models, lets compare the plot of the forecasted values for simple models built so far i.e. for Linear Regression Model, Naive Forecast Model, Simple Average Model, Moving Average Model (2-point trailing) :-
- Below are the comparison plots for both the wines: -

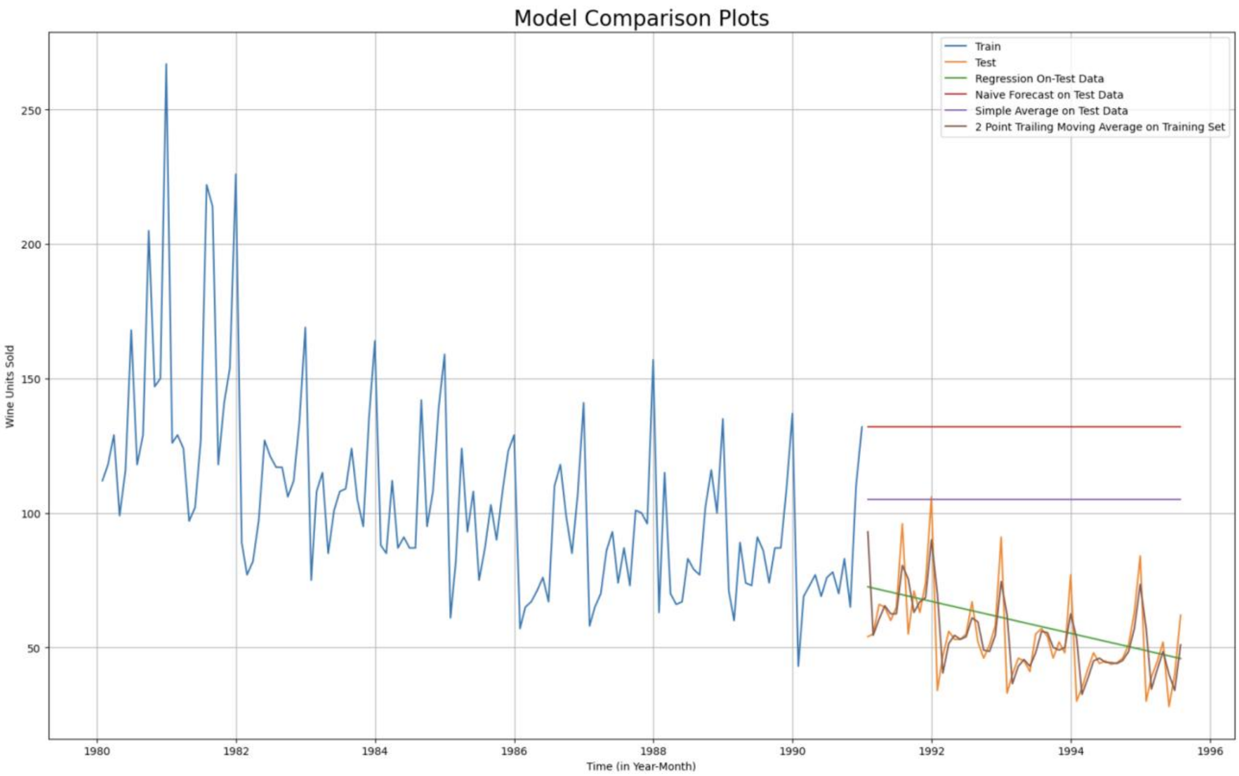


Figure 67: Rose | Comparison Plot for Simple Models

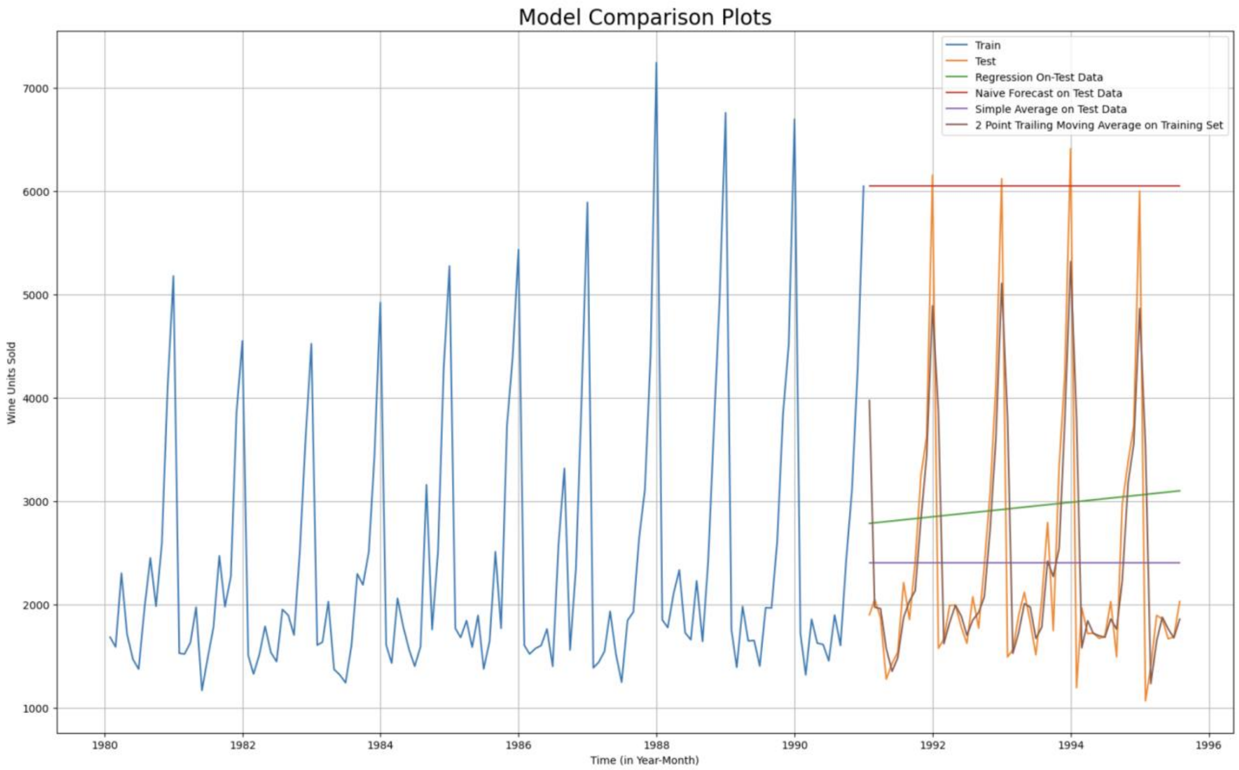


Figure 68: Sparkling | Comparison Plot for Simple Models

- **Observations (for both Rose & Sparkling Wines)**: -
  - Linear Regression Model – Captured the trend component, but the seasonality component has been missed.
  - Naive Forecast & Simple Average Model – Failed to capture, both, trend & seasonality components.
  - Moving Average (2-point trailing) – Captured both, trend & seasonality components.

Simple Exponential Smoothing Model (Autofit) – Rose Wine

- The simplest of the exponentially smoothing methods is called simple exponential smoothing (SES). This method is suitable for forecasting data with no clear trend or seasonal pattern. In Single ES, the forecast at time (t + 1) is given by: -
  $F_{t+1} = \alpha Y_t + (1 - \alpha)F_t$
- Parameter 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.
- We will build the **autofit model** first wherein the most **optimised hyperparameters** are selected using SimpleExpSmoothing() function in python.

```
{'smoothing_level': 0.12362013444181875,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 112.0,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Figure 69: Rose | SES Model (Autofit) Hyperparameters

- Below are the predicted values on the top 5 rows of test dataset: -

TimeStamp	RoseWine_Sales	SES_pred
1991-01-31	54.0	87.983765
1991-02-28	55.0	87.983765
1991-03-31	66.0	87.983765
1991-04-30	65.0	87.983765
1991-05-31	60.0	87.983765

Figure 70: Rose | Predicted values of test dataset for SES (Autofit)

- Below is the plot of the forecasted values against the train dataset: -

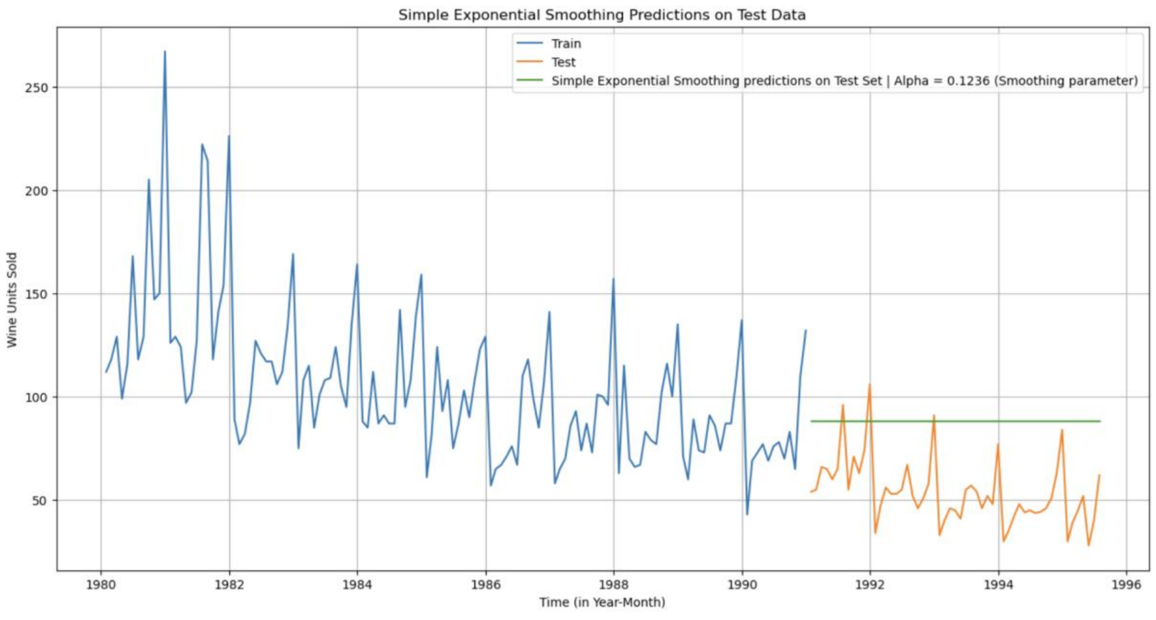


Figure 71: Rose | SES predicted values against the entire dataset

- **Model Performance** – Below are the RMSE values for SES Forecast Model: -

```
For Simple Exponential Smoothing Model forecast (Alpha = 0.1236) on the Test Data,  RMSE is 37.655
```

Figure 72: Rose | SES Forecast RMSE

- **Observations**: -
  - The more recent observation is given more weight with higher alpha value. This implies that the recent events will repeat again.
  - Clearly, the model was **not able to capture seasonality & trend**, in the forecast as this model **only takes into account the Level**.
  - The root means squared error (**RMSE**) for SES (autofit) Forecast model is **37.655**, which is **higher** than Moving Average Model.

Simple Exponential Smoothing Model (Autofit) – Sparkling Wine

- The simplest of the exponentially smoothing methods is called simple exponential smoothing (SES). This method is suitable for forecasting data with no clear trend or seasonal pattern. In Single ES, the forecast at time (t + 1) is given by: -
  $F_{t+1} = \alpha Y_t + (1-\alpha)F_t$
- Parameter 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.
- We will build the **autofit model** first wherein the most **optimised hyperparameters** are selected using SimpleExpSmoothing() function in python.

```
{'smoothing_level': 0.03953488372093023,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 1686.0,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Figure 73: Sparkling | SES Model (Autofit) Hyperparameters

- Below are the predicted values on the top 5 rows of test dataset: -

TimeStamp	SparklingWine_Sales	SES_pred
1991-01-31	1902	2676.676366
1991-02-28	2049	2676.676366
1991-03-31	1874	2676.676366
1991-04-30	1279	2676.676366
1991-05-31	1432	2676.676366

Figure 74: Sparkling | Predicted values of test dataset for SES (Autofit)

- Below is the plot of the forecasted values against the train dataset: -

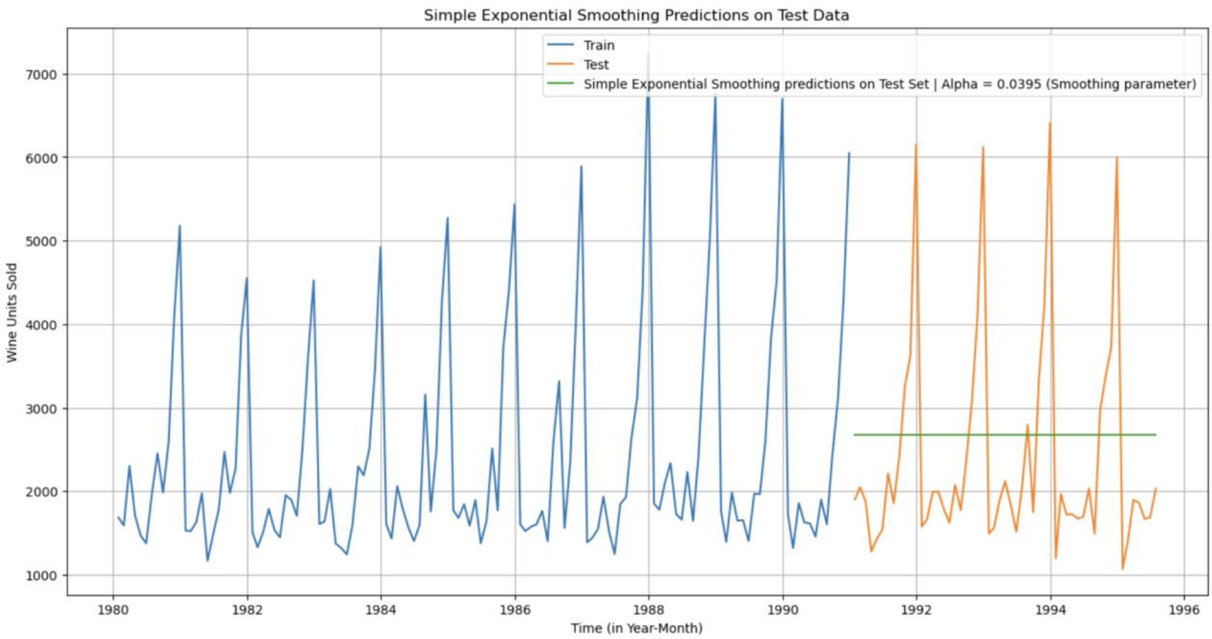


Figure 75: Sparkling | SES predicted values against the entire dataset

- **Model Performance** – Below are the RMSE values for SES Forecast Model: -

```
For Simple Exponential Smoothing Model forecast (Alpha = 0.0395) on the Test Data,  RMSE is 1304.927
```

Figure 76: Sparkling | SES Forecast RMSE

- **Observations**: -
  - The more recent observation is given more weight with higher alpha value. This implies that the recent events will repeat again.
  - Clearly, the model was **not able to capture seasonality & trend**, in the forecast as this model **only takes into account the Level**.
  - The root means squared error (**RMSE**) for SES (autofit) Forecast model is **1304.927**, which is **higher** than Moving Average Model.

Simple Exponential Smoothing Model (Manual) – Rose Wine

- The simplest of the exponentially smoothing methods is called simple exponential smoothing (SES). This method is suitable for forecasting data with no clear trend or seasonal pattern. In Single ES, the forecast at time (t + 1) is given by: -
  $F_{t+1} = \alpha Y_t + (1 - \alpha) F_t$
- Parameter 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.
- We will build the **model manually** now wherein we will calculate the RMSE for all alpha values in the range of 0.1 to 0.1 (step size 0.05) and **use the lowest one for model building: -**

	Alpha Values	Test RMSE
0.10	0.10	36.890375
0.15	0.15	38.784334
0.20	0.20	41.423877
0.25	0.25	44.422545
0.30	0.30	47.566302

Figure 77: Rose | Alpha against RMSE values in SES (manual) Model

- Below is the plot of the forecasted values against the train dataset: -

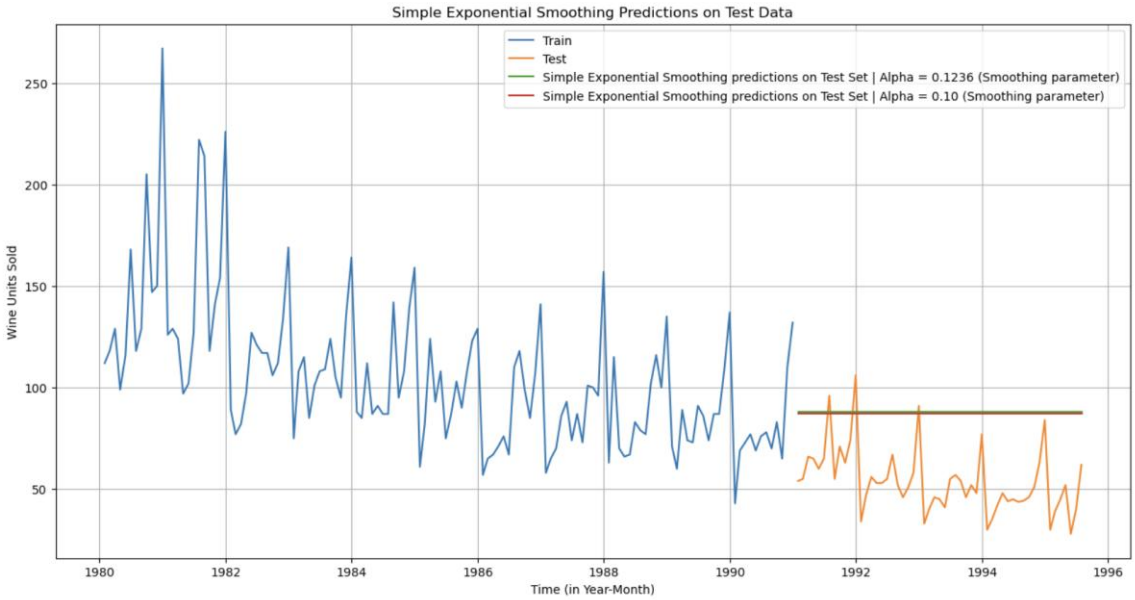


Figure 78: Rose | SES (manual) predicted values against the entire dataset

- **Model Performance** – Below are the RMSE values for SES Forecast Model (manual): -

```
For Simple Exponential Smoothing Model forecast (Alpha = 0.10) on the Test Data,  RMSE is 36.890
```

Figure 79: Rose | SES (manual) Forecast RMSE

- **Observations**: -
  - The more recent observation is given more weight with higher alpha value. This implies that the recent events will repeat again.
  - Clearly, the model was **not able to capture seasonality & trend**, in the forecast as this model **only takes into account the Level**.
  - The root means squared error (**RMSE**) for SES (manual) Forecast model is **36.890**, which is **slightly better** than the Autofit SES model but not as good as the Moving Average Model.
  - The **Simple Exponential Smoothing with alpha= 0.10 (Manual)** is taken as the best model against the Autofit versions as it has the lowest test RMSE.

Simple Exponential Smoothing Model (Manual) – Sparkling Wine

- The simplest of the exponentially smoothing methods is called simple exponential smoothing (SES). This method is suitable for forecasting data with no clear trend or seasonal pattern. In Single ES, the forecast at time (t + 1) is given by: -
  $F_{t+1} = \alpha Y_t + (1 - \alpha) F_t$
- Parameter 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.
- We will build the **model manually** now wherein we will calculate the RMSE for all alpha values in the range of 0.1 to 0.1 (step size 0.05) and **use the lowest one for model building**: -

	Alpha Values	Test RMSE
0.10	0.10	1375.393398
0.15	0.15	1466.203651
0.20	0.20	1595.206839
0.25	0.25	1755.488175
0.30	0.30	1935.507132

Figure 80: Sparkling | Alpha against RMSE values in SES (manual) Model

- Below is the plot of the forecasted values against the train dataset: -

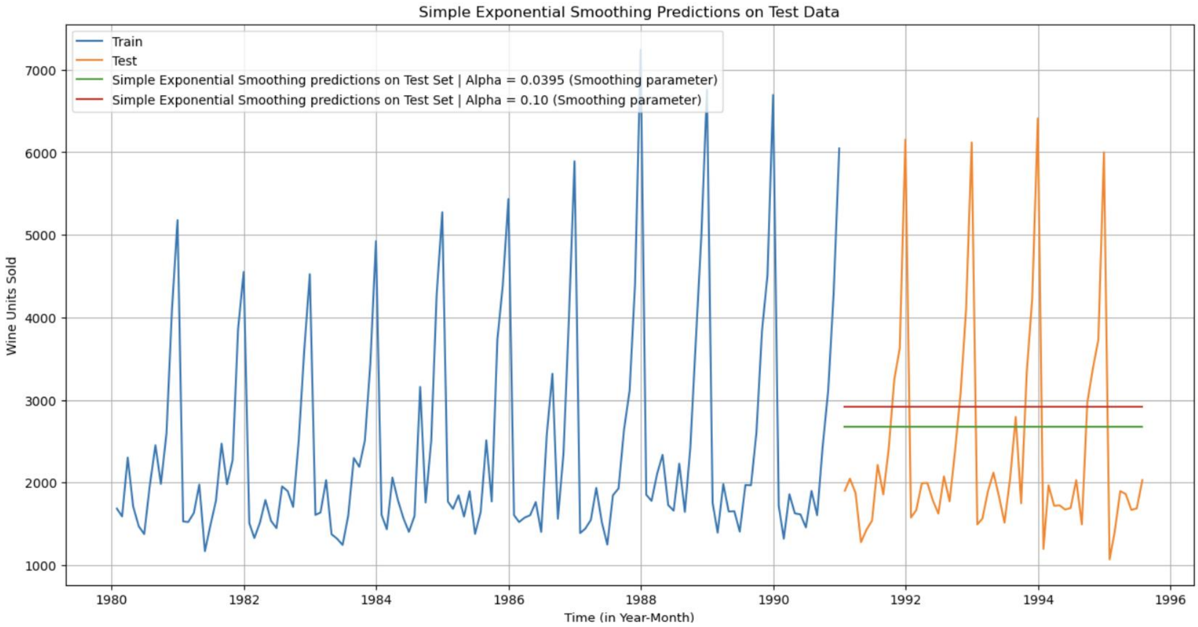


Figure 81: Sparkling | SES (manual) predicted values against the entire dataset

- **Model Performance** – Below are the RMSE values for SES Forecast Model (manual): -

```
For Simple Exponential Smoothing Model forecast (Alpha = 0.10) on the Test Data,  RMSE is 1375.393
```

Figure 82: Sparkling | SES (manual) Forecast RMSE

- **Observations**: -
  - The more recent observation is given more weight with higher alpha value. This implies that the recent events will repeat again.
  - Clearly, the model was **not able to capture seasonality & trend**, in the forecast as this model **only takes into account the Level**.
  - The root means squared error (**RMSE**) for SES (manual) Forecast model is **1375.393**, which is **not as good as the Autofit** SES model.
  - The **Simple Exponential Smoothing with alpha= 0.0395 (Autofit)** is taken as the best model against the Manual version as it has the lowest test RMSE.

Double Exponential Smoothing Model (Autofit) – Rose Wine

- This model is an extension of SES and estimates two smoothing parameters.
- It is applicable when the data has trend but no seasonality.
- Two separate components are considered: Level and Trend. Level is the local mean.
- One smoothing parameter α corresponds to the level series. A second smoothing parameter β corresponds to the trend series.

Intercept or Level equation, L_t is given by: $L_t = \alpha Y_t + (1-\alpha)F_t$

Trend equation is given by $T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$

- α and β are the smoothing constants for level and trend, respectively | 0 <α < 1 and 0 < β < 1.
- The forecast at time t + 1 is given by: -

$F_{t+1} = L_t + T_t$

$F_{t+n} = L_t + nT_t$

- We will **autofit model** first wherein the most **optimised hyperparameters** are selected using Holt() function in python -

```
{'smoothing_level': 1.4901161193847656e-08,
 'smoothing_trend': 1.6610391146660035e-10,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 137.81553690867275,
 'initial_trend': -0.4943781897068274,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Figure 83: Rose | DES Model (Autofit) Hyperparameters

- Below are the predicted values on the top 5 rows of test dataset: -

TimeStamp	RoseWine_Sales	DES_pred
1991-01-31	54.0	72.063238
1991-02-28	55.0	71.568859
1991-03-31	66.0	71.074481
1991-04-30	65.0	70.580103
1991-05-31	60.0	70.085725

Figure 84: Rose | Predicted values of test dataset for DES (Autofit)

- Below is the plot of the forecasted values against the train dataset: -

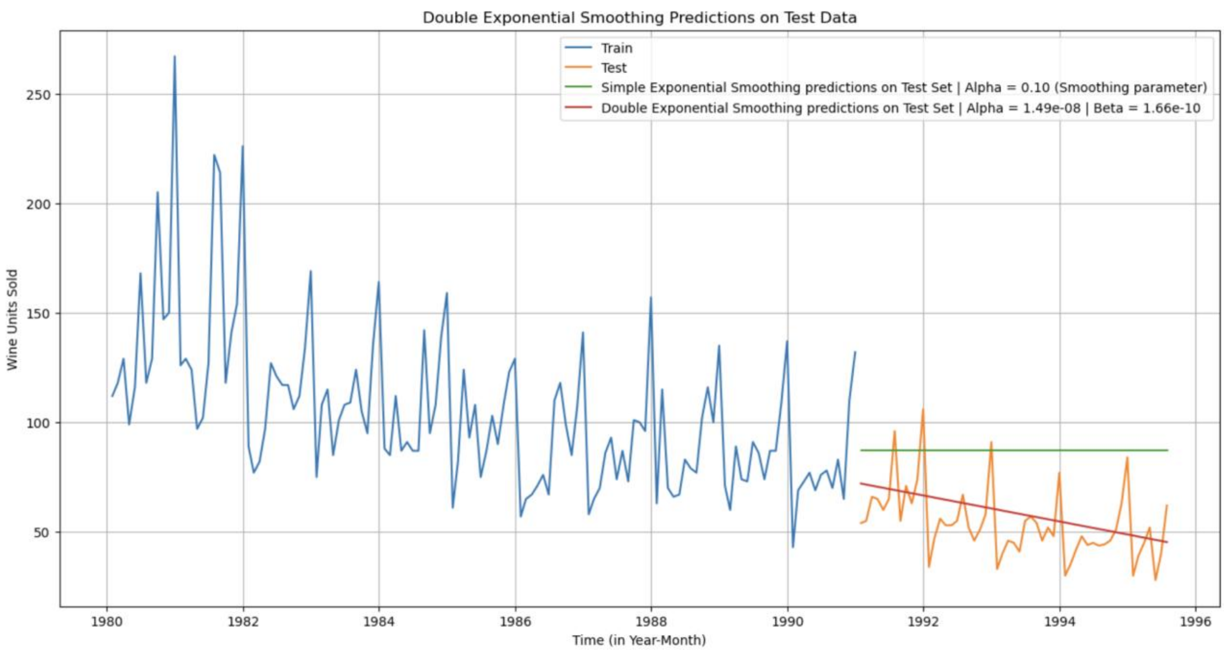


Figure 85: Rose | DES (Autofit) predicted values against the entire dataset

- **Model Performance** – Below are the RMSE values for DES Forecast Model (Autofit): -

```
For Double Exponential Smoothing Model forecast (Alpha = 1.49e-08, Beta = 1.66e-10) on the Test Data,  RMSE is 15.291
```

Figure 86: Rose | DES (Autofit) Forecast RMSE

- **Observations**: -
  - The model **captures the trend but fails to capture the seasonality**, which is expected in line with the definition of Double Exponential Smoothing Model.
  - Clearly, the model is **better than the Simple Exponential Smoothing Model** since **the RMSE has almost halved**.
  - The root means squared error (**RMSE**) for DES (Autofit) Forecast model is **15.291**, which is **lowest,** relative to **all the models but Moving Average (which captures seasonality also).**

Double Exponential Smoothing Model (Autofit) – Sparkling Wine

- This model is an extension of SES and estimates two smoothing parameters.
- It is applicable when the data has trend but no seasonality.
- Two separate components are considered: Level and Trend. Level is the local mean.
- One smoothing parameter α corresponds to the level series. A second smoothing parameter β corresponds to the trend series.

Intercept or Level equation, L_t is given by: $L_t = \alpha Y_t + (1 - \alpha) F_t$

Trend equation is given by $T_t = \beta(L_t - L_{t-1}) + (1 - \beta) T_{t-1}$

- α and β are the smoothing constants for level and trend, respectively | 0 <α < 1 and 0 < β < 1.
- The forecast at time t + 1 is given by: -
  $F_{t+1} = L_t + T_t$
  $F_{t+n} = L_t + nT_t$

- We will **autofit model** first wherein the most **optimised hyperparameters** are selected using Holt() function in python -

```
{'smoothing_level': 0.6885714285714285,
 'smoothing_trend': 9.999999999999999e-05,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 1686.0,
 'initial_trend': -95.0,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Figure 87: Sparkling | DES Model (Autofit) Hyperparameters

- Below are the predicted values on the top 5 rows of test dataset: -

	SparklingWine_Sales	DES_pred
TimeStamp		
1991-01-31	1902	5221.278699
1991-02-28	2049	5127.886554
1991-03-31	1874	5034.494409
1991-04-30	1279	4941.102264
1991-05-31	1432	4847.710119

Figure 88: Sparkling | Predicted values of test dataset for DES (Autofit)

- Below is the plot of the forecasted values against the train dataset: -

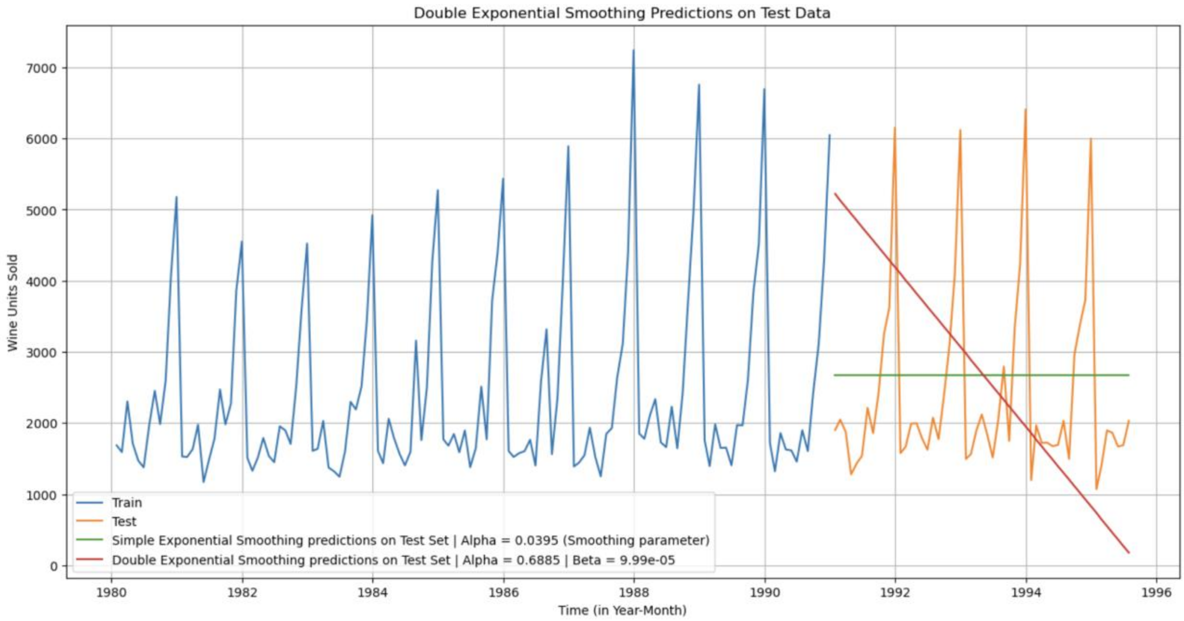


Figure 89: Sparkling | DES (Autofit) predicted values against the entire dataset

- **Model Performance** – Below are the RMSE values for DES Forecast Model (Autofit): -

```
For Double Exponential Smoothing Model forecast (Alpha = 0.6885, Beta = 9.99e-05) on the Test Data,  RMSE is 2007.239
```

Figure 90: Sparkling | DES (Autofit) Forecast RMSE

- **Observations**: -
  - Though the model has **tried to capture the trend** but it seems that, given high spikes in the data, it was **not able to properly capture the trend**. The overall trend is more or less constant, but the model is predicting a falling trend.
  - The model is **not expected to capture seasonality**, as per the definition of Double Exponential Smoothing Model.
  - The model is **not better than the Simple Exponential Smoothing Model** since **the RMSE is higher** due to wrong prediction in the trend.
  - The root means squared error (**RMSE**) for DES (Autofit) Forecast model is **2007.239**, which is **not as good as other models.**

Double Exponential Smoothing Model (Manual) – Rose Wine

- This model is an extension of SES and estimates two smoothing parameters.
- It is applicable when the data has trend but no seasonality.
- Two separate components are considered: Level and Trend. Level is the local mean.
- One smoothing parameter α corresponds to the level series. A second smoothing parameter β corresponds to the trend series.

Intercept or Level equation, L_t is given by: $L_t = \alpha Y_t + (1-\alpha)F_t$

Trend equation is given by $T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$

- α and β are the smoothing constants for level and trend, respectively | 0 <α < 1 and 0 < β < 1.
- The forecast at time t + 1 is given by: -
  $F_{t+1} = L_t + T_t$
  $F_{t+n} = L_t + nT_t$

- We will build the **model manually** now wherein we will calculate the RMSE for all Alpha & Beta values in the range of 0.1 to 0.1 (step size 0.05) for both smoothing parameters and **use the lowest one for model building**: -

	Alpha	Beta	Test RMSE
0.35	0.05	0.35	15.670743
0.30	0.05	0.30	18.480365
0.15	0.05	0.15	22.659851
0.05	0.05	0.05	30.907779
0.40	0.05	0.40	31.137095

Figure 91: Rose | Alpha, Beta against RMSE values in DES (manual) Model

- Below is the plot of the forecasted values against the train dataset: -

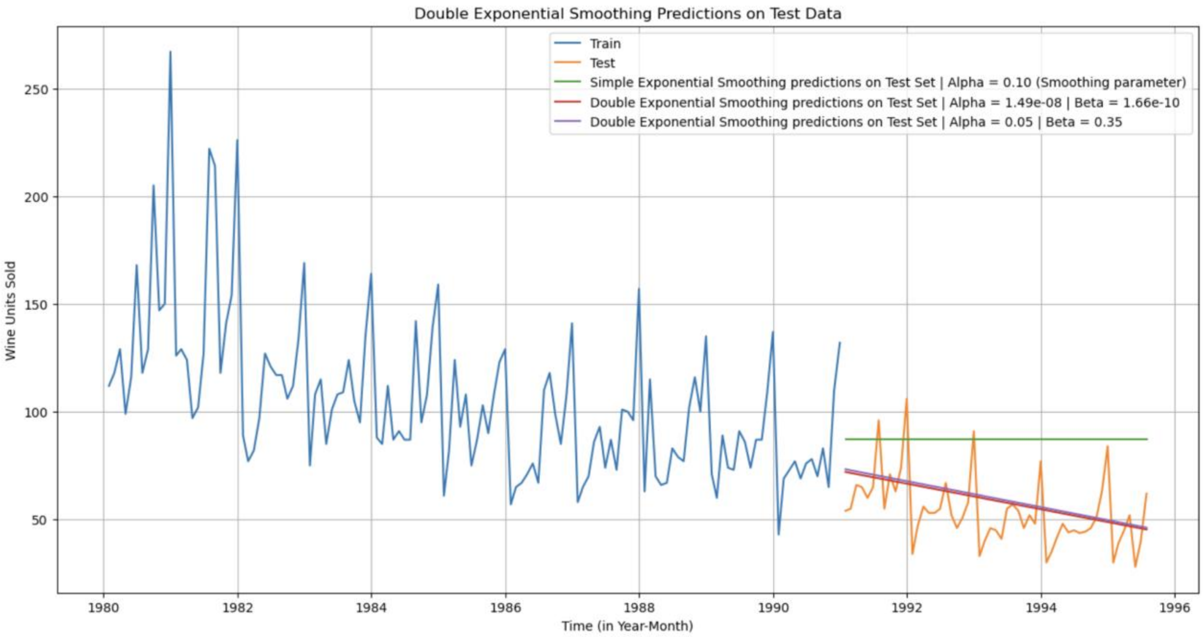


Figure 92: Rose | DES (Manual) predicted values against the entire dataset

- **Model Performance** – Below are the RMSE values for DES Forecast Model (Manual): -

```
For Double Exponential Smoothing Model forecast (Alpha = 0.05, Beta = 0.35) on the Test Data,  RMSE is 15.671
```

Figure 93: Rose | DES (Manual) Forecast RMSE

- **Observations**: -
  - The model **captures the trend but fails to capture the seasonality**, which is expected in line with the definition of Double Exponential Smoothing Model.
  - Clearly, the model is **better than the Simple Exponential Smoothing Model** since **the RMSE has almost halved**.
  - The root means squared error (**RMSE**) for DES (Manual) Forecast model is **15.671**, which is **comparable with the autofit model** but is better than **all the models except Moving Average (which captures seasonality also).**
  - The **Double Exponential Smoothing with Alpha 1.49e-08 & Beta 1.66e-10 (Autofit version)** is taken as the best model against the Manual version as it has the lowest test RMSE

Double Exponential Smoothing Model (Manual) – Sparkling Wine

- This model is an extension of SES and estimates two smoothing parameters.
- It is applicable when the data has trend but no seasonality.
- Two separate components are considered: Level and Trend. Level is the local mean.
- One smoothing parameter α corresponds to the level series. A second smoothing parameter β corresponds to the trend series.

Intercept or Level equation, L_t is given by: $L_t = \alpha Y_t + (1-\alpha)F_t$

Trend equation is given by $T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$

- α and β are the smoothing constants for level and trend, respectively | 0 <α < 1 and 0 < β < 1.
- The forecast at time t + 1 is given by: -

$F_{t+1} = L_t + T_t$

$F_{t+n} = L_t + nT_t$

- We will build the **model manually** now wherein we will calculate the RMSE for all Alpha & Beta values in the range of 0.1 to 0.1 (step size 0.05) for both smoothing parameters and **use the lowest one for model building**: -

	Alpha	Beta	Test RMSE
0.05	0.05	0.05	1418.407668
0.20	0.05	0.20	1443.099273
0.15	0.05	0.15	1457.041594
0.10	0.05	0.10	1466.899629
0.35	0.05	0.35	1547.022626

Figure 94: Sparkling | Alpha, Beta against RMSE values in DES (manual) Model

- Below is the plot of the forecasted values against the train dataset: -

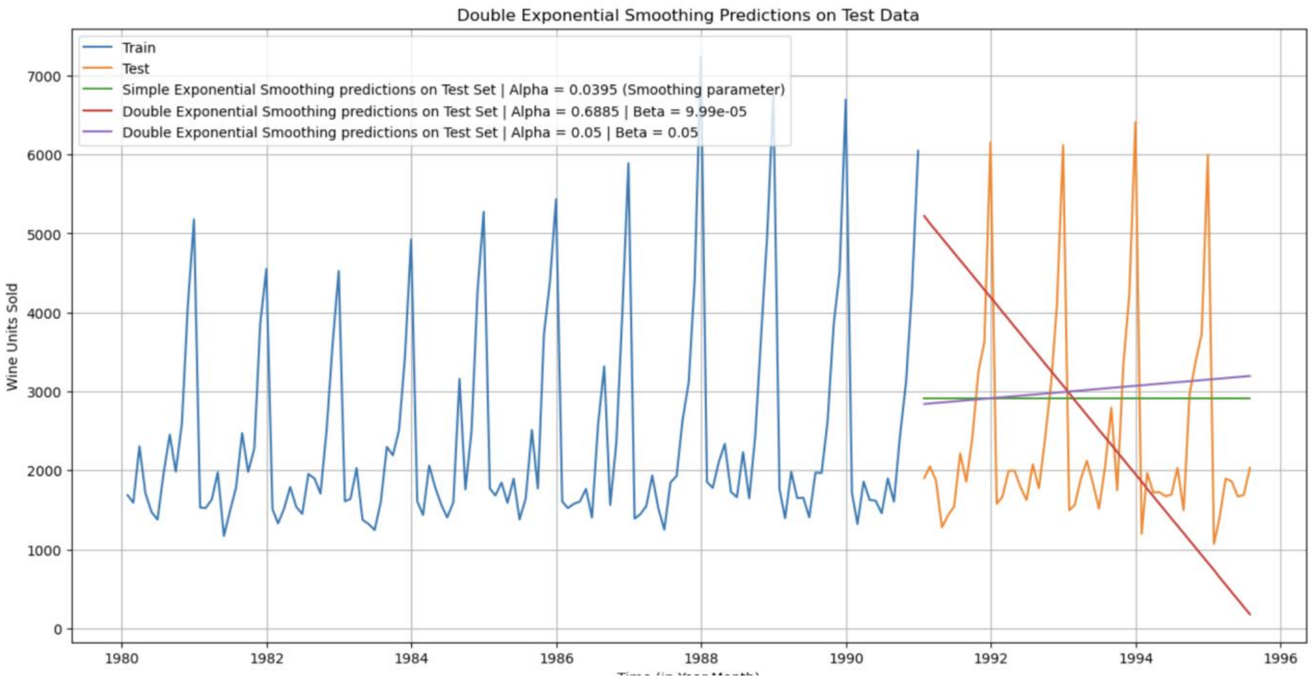


Figure 95: Sparkling | DES (Manual) predicted values against the entire dataset

- **Model Performance** – Below are the RMSE values for DES Forecast Model (Manual): -

```
For Double Exponential Smoothing Model forecast (Alpha = 0.05, Beta = 0.05) on the Test Data,  RMSE is 1418.408
```

Figure 96: Sparkling | DES (Manual) Forecast RMSE

- **Observations**: -
  - The model is able **capture the trend to some extent** which is showing a slight increasing trend.
  - However, it **fails to capture the seasonality**, which is expected in line with the definition of Double Exponential Smoothing Model.
  - The root means squared error (**RMSE**) for DES (Manual) Forecast model is **1418.408**, which is **not as good as other models.**
  - The **Double Exponential Smoothing with Alpha 0.05 & Beta 0.05 (Manual version)** is taken as the best model against the Autofit version as it has the lowest test RMSE

Triple Exponential Smoothing Model (Autofit) – Rose Wine

- This model is an extension of DES and estimates three smoothing parameters.
- It is applicable when data has both Trend and seasonality.
- Three separate components are considered: Level, Trend and Seasonality.
- One smoothing parameter α corresponds to the Level series. A second smoothing parameter β corresponds to the trend series. A third smoothing parameter γ corresponds to the seasonality series.
- $0 < \alpha < 1$, $0 < \beta < 1$, $0 < \gamma < 1$

- We will **autofit model** first wherein the most **optimised hyperparameters** are selected using ExponentialSmoothing() function in python.
- However, we will **try different combinations** to obtain the most optimised hyperparameters giving the lowest RMSE: -
  - Combination 1: Trend – Additive | Seasonality – Additive
  - Combination 2: Trend – Additive | Seasonality – Multiplicative
  - Combination 3: Trend – Multiplicative | Seasonality – Additive
  - Combination 4: Trend – Multiplicative | Seasonality – Multiplicative
  - Combination 5: Trend – No Trend | Seasonality – Multiplicative
  - Combination 6: Trend – No Trend | Seasonality – Additive
- As evident below, the 1st **combination, where both Trend & Seasonality are Additive** in nature, gives the lowest RMSE. So, we will use this to obtain the most optimum hyperoperators to build the model.

	TES_Type	RMSE
0	RMSE (Add|Add)	14.292378
2	RMSE (Mul|Add)	18.600040
3	RMSE (Mul|Mul)	20.064614
1	RMSE (Add|Mul)	20.232984
5	RMSE (NA|Add)	26.761311
4	RMSE (NA|Mul)	27.865038

Figure 97: Rose | TES RMSE with different combinations

- We will **autofit model** first wherein the most **optimised hyperparameters** are selected as below: -

```
{'smoothing_level': 0.08954054664605082,
 'smoothing_trend': 0.0002400108693915795,
 'smoothing_seasonal': 0.003466872515750747,
 'damping_trend': nan,
 'initial_level': 146.5570157826235,
 'initial_trend': -0.547196983509005,
 'initial_seasons': array([-31.17478463, -18.74839869, -10.76961776, -21.36741017,
        -12.63775539,  -7.27430333,   2.61279801,   8.69603625,
          4.79381122,   2.96110122,  21.05738849,  63.18279918]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Figure 98: Rose | TES Model (Autofit) Hyperparameters

- Below are the predicted values on the top 5 rows of test dataset: -

	RoseWine_Sales	TES_pred_aa	TES_pred_am	TES_pred_ma	TES_pred_mm	TES_pred_nm	TES_pred_na	TES_pred
TimeStamp								
1991-01-31	54.0	42.684928	56.321655	43.031585	55.663816	58.487979	46.311523	42.684928
1991-02-28	55.0	54.564005	63.664690	55.146566	62.993228	65.963835	58.644004	54.564005
1991-03-31	66.0	61.995209	69.374024	62.896673	68.738503	71.690637	66.560704	61.995209
1991-04-30	65.0	50.852018	60.435528	52.007396	59.835212	63.485656	55.874847	50.852018
1991-05-31	60.0	59.034271	67.758341	60.570926	67.118704	70.690366	64.576703	59.034271

Figure 99: Rose | Predicted values of test dataset for TES (Autofit)

- Below is the plot of the forecasted values against the train dataset: -

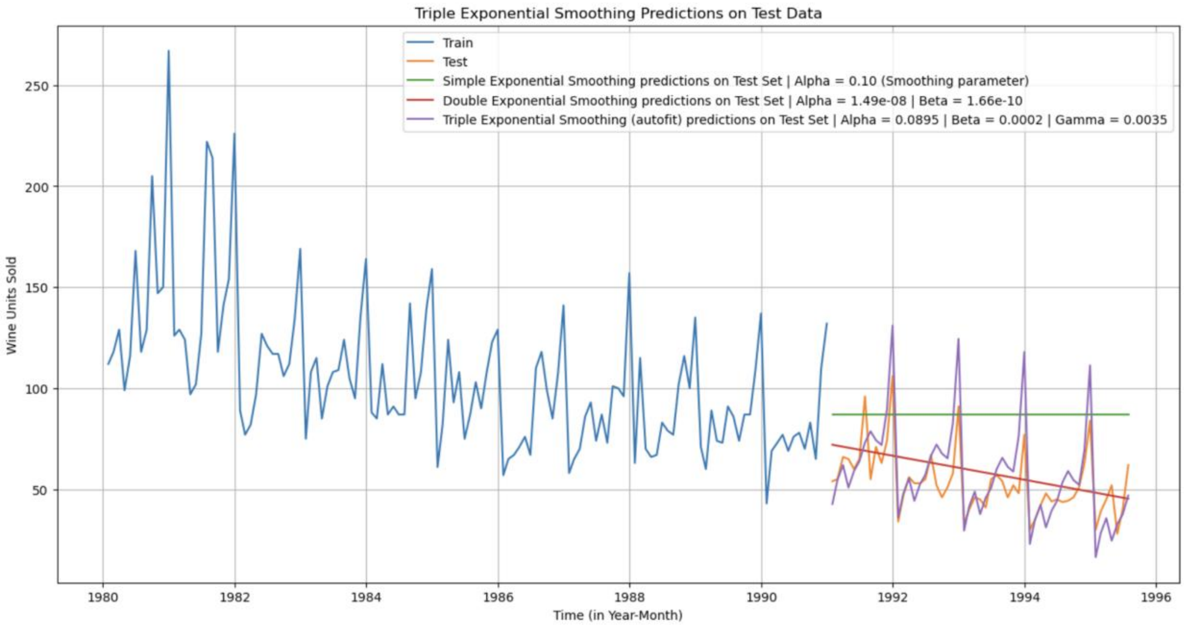


Figure 100: Rose | TES (Autofit) predicted values against the entire dataset

- **Model Performance** – Below are the RMSE values for TES Forecast Model (Autofit): -

```
For Triple Exponential Smoothing Model forecast (Alpha = 0.0895, Beta = 0.0002 & Gamma = 0.0035) on the Test Data,  RMSE is 14.292
```

Figure 101: Rose | TES (Autofit) Forecast RMSE

- **Observations**: -
  - The model **captures both the trend & seasonality** pretty well as evident in the chart.
  - Clearly, the model is **better than both the Simple Exponential Smoothing & Double Exponential Smoothing Models**.
  - The root means squared error (**RMSE**) for TES (Autofit) Forecast model is **14.292**, which is **lowest,** relative to **all the models but Moving Average (2-point trailing).**

Triple Exponential Smoothing Model (Autofit) – Sparkling Wine

- This model is an extension of DES and estimates three smoothing parameters.
- It is applicable when data has both Trend and seasonality.
- Three separate components are considered: Level, Trend and Seasonality.
- One smoothing parameter α corresponds to the Level series. A second smoothing parameter β corresponds to the trend series. A third smoothing parameter γ corresponds to the seasonality series.
- $0 < \alpha < 1, 0 < \beta < 1, 0 < \gamma < 1$

- We will **autofit model** first wherein the most **optimised hyperparameters** are selected using ExponentialSmoothing() function in python.
- However, we will **try different combinations** to obtain the most optimised hyperparameters giving the lowest RMSE: -
  - Combination 1: Trend – Additive | Seasonality – Additive
  - Combination 2: Trend – Additive | Seasonality – Multiplicative
  - Combination 3: Trend – Multiplicative | Seasonality – Additive
  - Combination 4: Trend – Multiplicative | Seasonality – Multiplicative
  - Combination 5: Trend – No Trend | Seasonality – Multiplicative
  - Combination 6: Trend – No Trend | Seasonality – Additive
- As evident below, the **5th combination, where we have No Trend but Multiplicative Seasonality**, gives the lowest RMSE. So, we will use this to obtain the most optimum hyperoperators to build the model.

	TES_Type	RMSE
4	RMSE (NA|Mul)	318.831018
5	RMSE (NA|Add)	358.884513
0	RMSE (Add|Add)	378.951023
3	RMSE (Mul|Mul)	380.398478
2	RMSE (Mul|Add)	381.190534
1	RMSE (Add|Mul)	404.286809

Figure 102: Sparkling | TES RMSE with different combinations

- We will **autofit model** first wherein the most **optimised hyperparameters** are selected as below: -

```
{'smoothing_level': 0.1571861945453721,
 'smoothing_trend': nan,
 'smoothing_seasonal': 0.3988473432692238,
 'damping_trend': nan,
 'initial_level': 2339.583573821,
 'initial_trend': nan,
 'initial_seasons': array([0.71154022, 0.68368103, 0.89128632, 0.80410254, 0.66158018,
        0.66379696, 0.87673153, 1.1097104 , 0.91191736, 1.17017652,
        1.8024179 , 2.26524454]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Figure 103: Sparkling | TES Model (Autofit) Hyperparameters

- Below are the predicted values on the top 5 rows of test dataset: -

TimeStamp	SparklingWine_Sales	TES_pred_aa	TES_pred_am	TES_pred_ma	TES_pred_mm	TES_pred_nm	TES_pred_na	TES_pred
1991-01-31	1902	1490.402890	1587.497468	1483.304864	1591.299973	1574.383785	1558.682382	1574.383785
1991-02-28	2049	1204.525152	1356.394925	1199.007966	1360.408886	1334.761769	1261.960669	1334.761769
1991-03-31	1874	1688.734182	1762.929755	1682.755193	1767.949510	1766.428452	1748.901681	1766.428452
1991-04-30	1279	1551.226125	1656.165933	1546.660734	1661.619432	1653.470080	1601.375681	1653.470080
1991-05-31	1432	1461.197883	1542.002730	1456.387757	1547.414170	1557.888218	1511.761770	1557.888218

Figure 104: Sparkling | Predicted values of test dataset for TES (Autofit)

- Below is the plot of the forecasted values against the train dataset: -

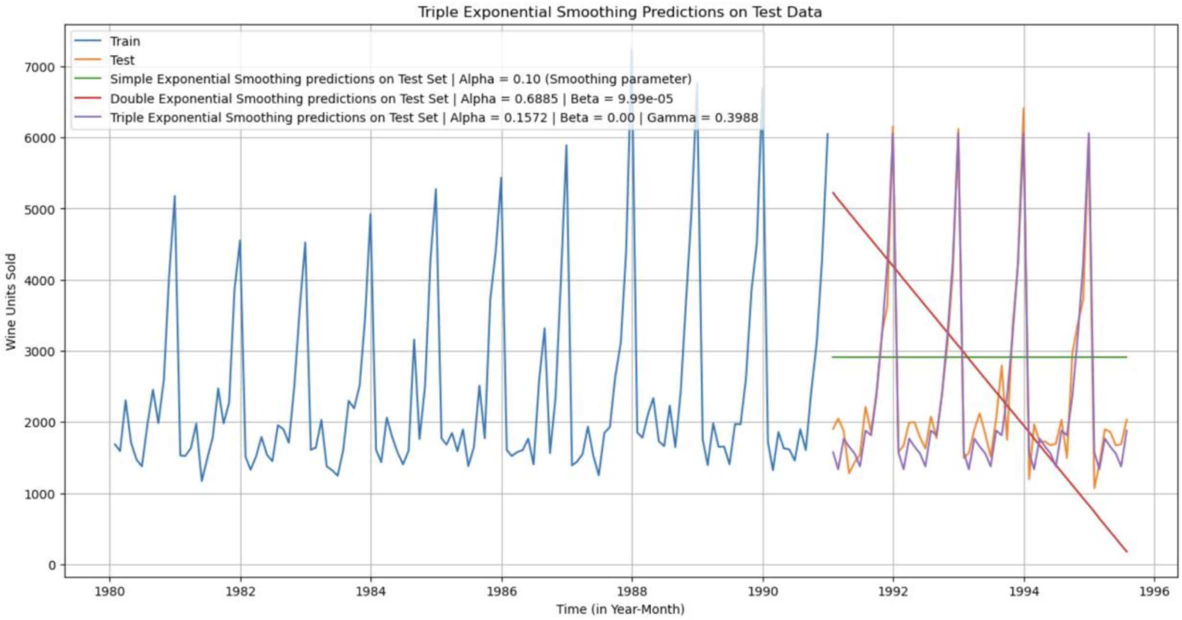


Figure 105: Sparkling | TES (Autofit) predicted values against the entire dataset

- **Model Performance** – Below are the RMSE values for TES Forecast Model (Autofit): -

```
For Triple Exponential Smoothing Model forecast (Alpha = 0.1572, Beta = 0.00 & Gamma = 0.3988) on the Test Data,  RMSE is 318.831
```

Figure 106: Sparkling | TES (Autofit) Forecast RMSE

- **Observations**: -
  - The model **captures both the trend & seasonality** very well as evident in the chart.
  - Clearly, the model is **better than both the Simple Exponential Smoothing & Double Exponential Smoothing Models**.
  - The root means squared error (**RMSE**) for TES (Autofit) Forecast model is **318.831**, which is **lowest, relative to all other models. This is the best model so far.**

Triple Exponential Smoothing Model (Manual) – Rose Wine

- This model is an extension of DES and estimates three smoothing parameters.
- It is applicable when data has both Trend and seasonality.
- Three separate components are considered: Level, Trend and Seasonality.
- One smoothing parameter α corresponds to the Level series. A second smoothing parameter β corresponds to the trend series. A third smoothing parameter γ corresponds to the seasonality series.
- $0 < \alpha < 1, 0 < \beta < 1, 0 < \gamma < 1$

- We will build the **model manually** now wherein we will calculate the RMSE for all Alpha, Beta & Gamma values in the range of 0.1 to 0.1 (step size 0.05) for all smoothing parameters and **use the lowest one for model building**: -

	Alpha	Beta	Gamma	Test RMSE
0.10	0.1	0.20	0.10	9.263811
0.10	0.1	0.15	0.10	9.313443
0.15	0.1	0.15	0.15	9.341160
0.15	0.1	0.20	0.15	9.373094
0.20	0.1	0.15	0.20	9.431296

Figure 107: Rose | Alpha, Beta, Gamma against RMSE values in TES (manual) Model

- Below is the plot of the forecasted values against the train dataset: -

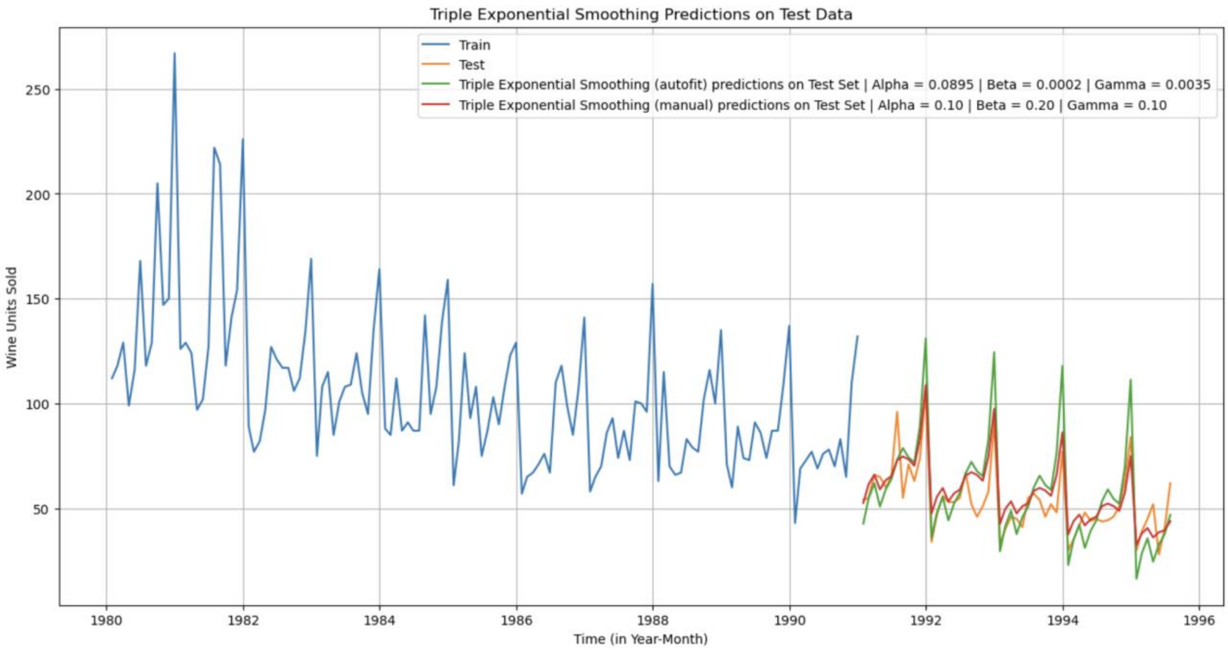


Figure 108: Rose | TES (Manual) predicted values against the entire dataset

- **Model Performance** – Below are the RMSE values for TES Forecast Model (Manual): -

```
For Triple Exponential Smoothing Model forecast (Alpha = 0.10, Beta = 0.20 & Gamma = 0.10) on the Test Data,  RMSE is 9.264
```

Figure 109: Rose | TES (Manual) Forecast RMSE

- **Observations**: -
  - The model **captures both the trend & seasonality** very well as evident in the chart.
  - The root means squared error (**RMSE**) for TES (Manual) Forecast model is **9.264.831**, which is **lowest, relative to all other models. This is the best model so far.**
  - **Triple Exponential Smoothing Model with Alpha 0.10, Beta 0.20 & Gamma 0.10 (Manual version)** is taken as the **best model against the Autofit version** as it has the lowest test RMSE

Triple Exponential Smoothing Model (Manual) – Sparkling Wine

- This model is an extension of DES and estimates three smoothing parameters.
- It is applicable when data has both Trend and seasonality.
- Three separate components are considered: Level, Trend and Seasonality.
- One smoothing parameter α corresponds to the Level series. A second smoothing parameter β corresponds to the trend series. A third smoothing parameter γ corresponds to the seasonality series.
- 0 < α <1, 0 < β <1, 0 < γ <1

- We will build the **model manually** now wherein we will calculate the RMSE for all Alpha, Beta & Gamma values in the range of 0.1 to 0.1 (step size 0.05) for all smoothing parameters and **use the lowest one for model building**: -

	Alpha	Beta	Gamma	Test RMSE
0.25	0.1	0.00	0.25	302.911684
0.25	0.1	0.04	0.25	302.911684
0.25	0.1	0.06	0.25	302.911684
0.25	0.1	0.08	0.25	302.911684
0.25	0.1	0.02	0.25	302.911684

Figure 110: Sparkling | Alpha, Beta, Gamma against RMSE values in TES (manual) Model

- Below is the plot of the forecasted values against the train dataset: -

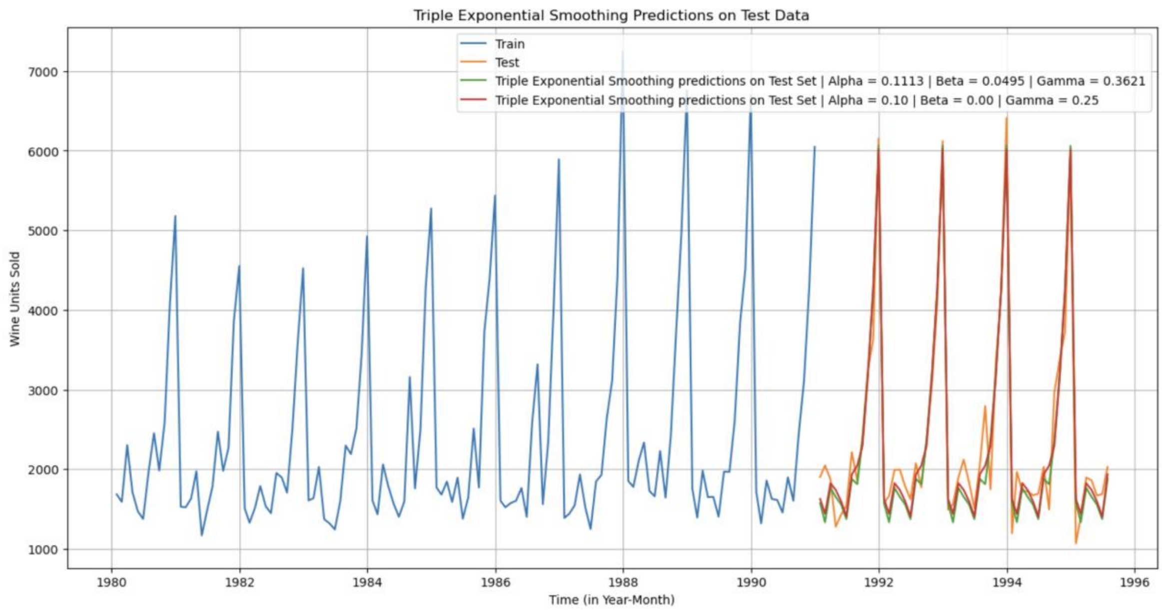


Figure 111: Sparkling | TES (Manual) predicted values against the entire dataset

- **Model Performance** – Below are the RMSE values for TES Forecast Model (Manual): -

```
For Triple Exponential Smoothing Model forecast (Alpha = 0.10, Beta = 0.00 & Gamma = 0.25) on the Test Data,  RMSE is 302.912
```

Figure 112: Sparkling | TES (Manual) Forecast RMSE

- **Observations**: -
  - The model **captures both the trend & seasonality** very well as evident in the chart.
  - The root means squared error (**RMSE**) for TES (Manual) Forecast model is **302.912**, which is **lowest, relative to all other models. This is the best model so far.**
  - **Triple Exponential Smoothing Model with Alpha 0.10, Beta 0.00 & Gamma 0.25 (Manual version)** is taken as the **best model against the Autofit version** as it has the lowest test RMSE

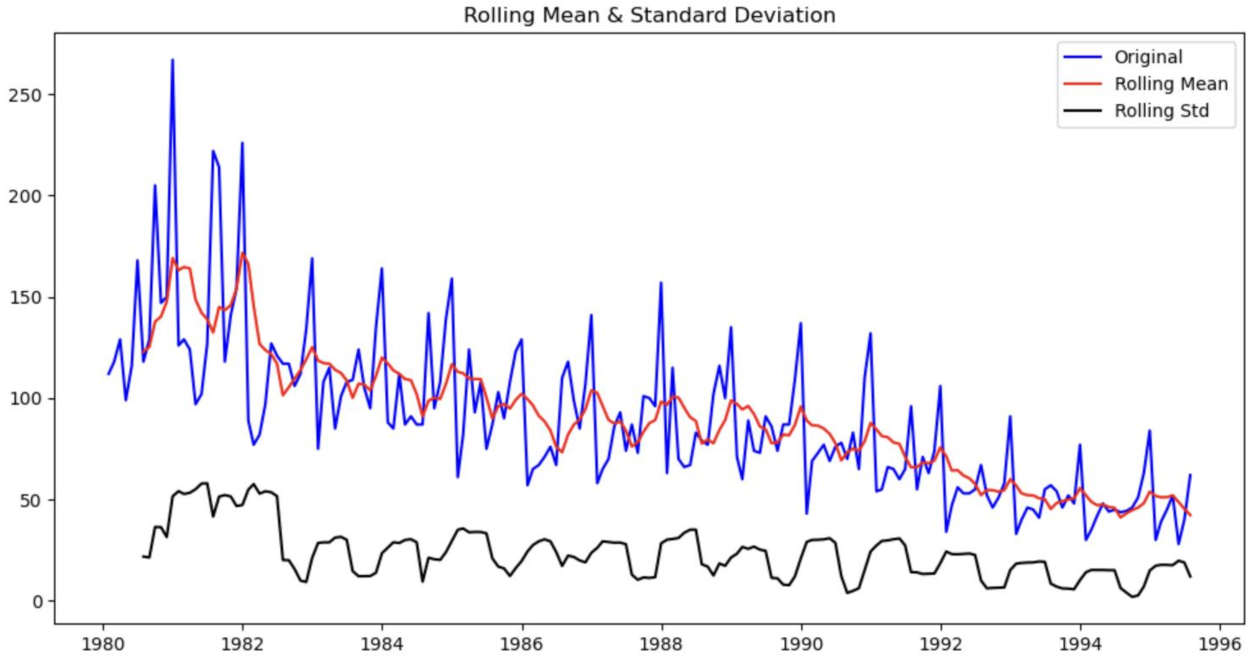
Rubric Question 4: Check for Stationarity

Checking Stationarity of Entire Data – Rose Wine

- A Time Series is considered to be stationary when **statistical properties such as the variance and (auto) correlation are constant over time.** Stationary Time Series allows us to think of the statistical properties of the time series as not changing in time, which enables us to build appropriate statistical models for forecasting based on past data.
- Stationarity means that the **autocorrelation of lag 'k' depends on k, but not on time t.**
- The **Augmented Dickey-Fuller test** is a unit root test which determines whether there is a unit root and subsequently whether the series is non-stationary.
- Framing the **hypothesis**: -

  H0: The Time Series has a unit root and is thus non-stationary.

  H1: The Time Series does not have a unit root and is thus stationary.
- The series have to be **stationary for building ARIMA/SARIMA** models and thus we would want the **p-value of this test to be less than the α** value.
- Below is the output of the test: -

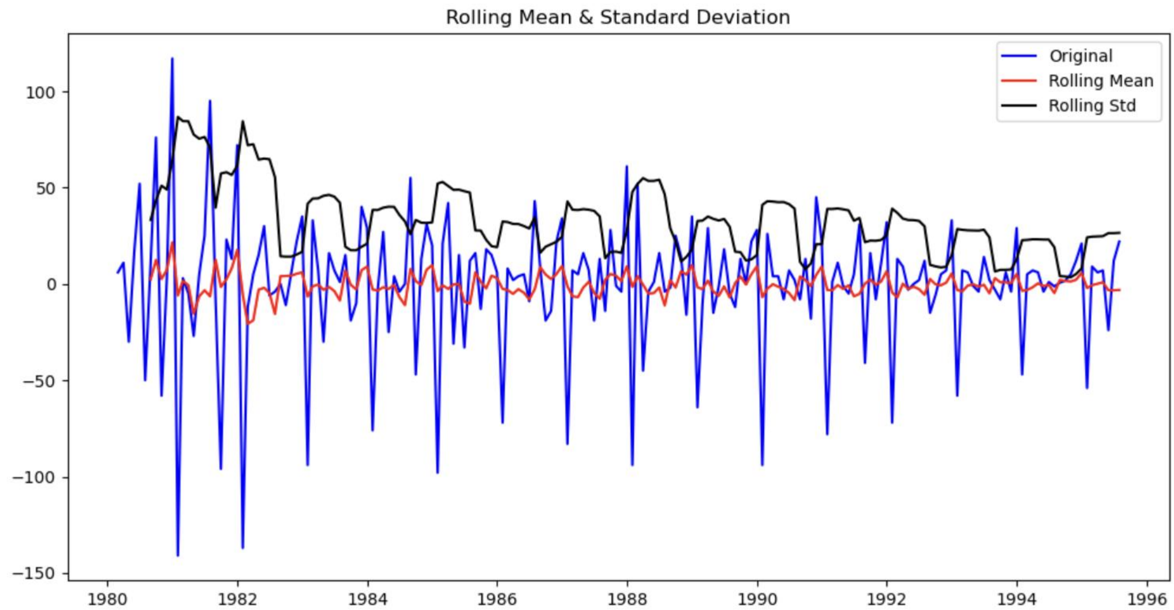



```
Results of Dickey-Fuller Test:
Test Statistic                  -1.873273
p-value                          0.344737
#Lags Used                      13.000000
Number of Observations Used    173.000000
Critical Value (1%)             -3.468726
Critical Value (5%)             -2.878396
Critical Value (10%)            -2.575756
dtype: float64
```

Figure 113: Rose | Test for Stationarity of the entire dataset

- We see that at **5% significant level the Time Series is non-stationary as p-value is 0.3447** which is **more than alpha value (0.05)**, therefore we **fail to reject the null hypothesis.**
- Also, by looking at the plot, it is evident that rolling std. deviation is not constant over time.

- Let us take **one level of differencing** to see whether the series becomes stationary & **carry out the test again.** Below is the output**: -**




```
Results of Dickey-Fuller Test:
Test Statistic                -8.043385e+00
p-value                        1.821604e-12
#Lags Used                     1.200000e+01
Number of Observations Used    1.730000e+02
Critical Value (1%)           -3.468726e+00
Critical Value (5%)           -2.878396e+00
Critical Value (10%)          -2.575756e+00
dtype: float64
```

Figure 114: Rose | Test for Stationarity of the entire dataset (d=1)

- We see that at 5% significant level the **Time Series becomes stationary as p-value is 1.821e-12 which is less than alpha value (0.05)**, therefore we reject the null hypothesis. We can see that the provided **time series becomes stationary with differencing**.
- Also, by looking at the plot, it is evident that rolling std. deviation is constant over time.

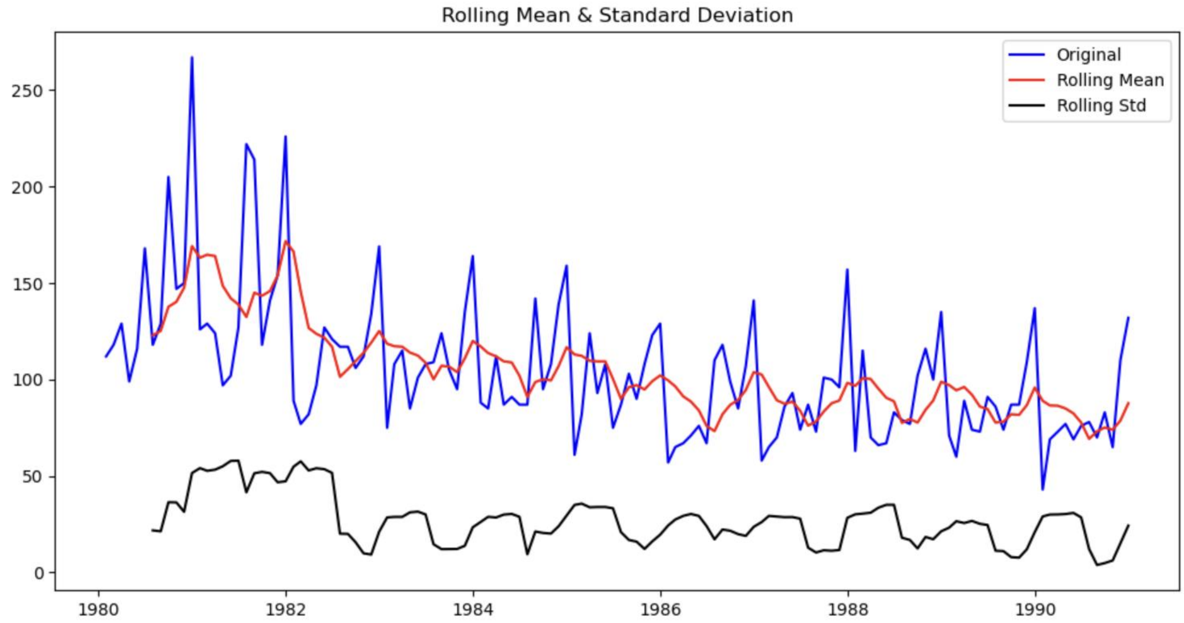
Checking Stationarity of Training Data – Rose Wine

- Framing the **hypothesis**: -

  H0: The Time Series has a unit root and is thus non-stationary.

  H1: The Time Series does not have a unit root and is thus stationary.

- Below is the output of the test: -

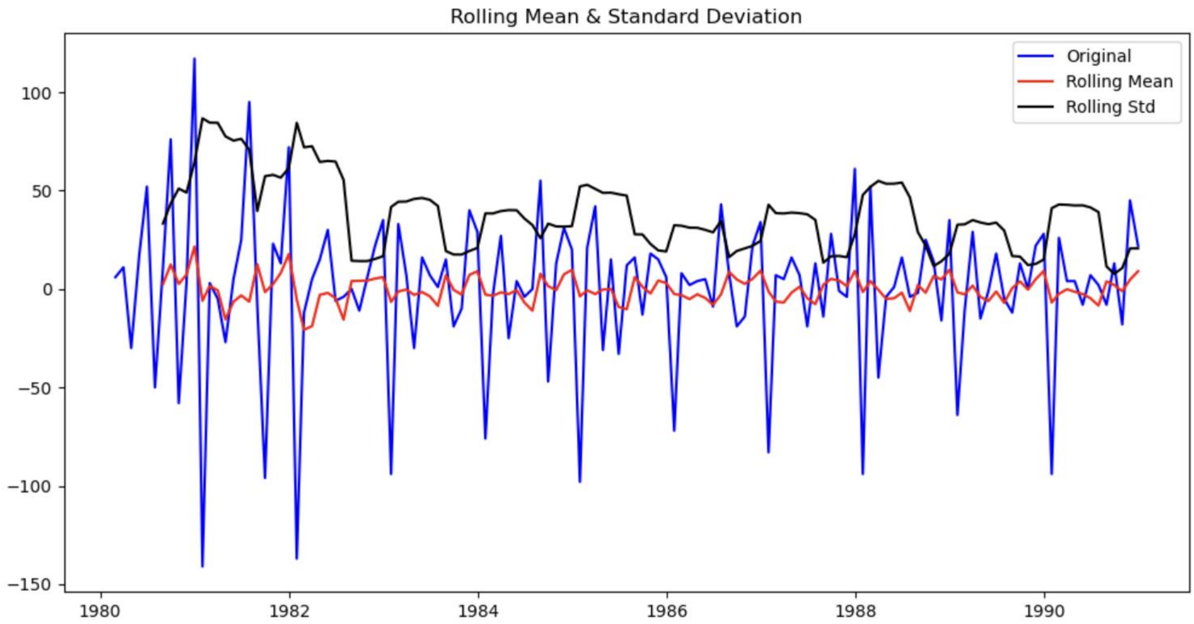



```
Results of Dickey-Fuller Test:
Test Statistic                 -2.164250
p-value                         0.219476
#Lags Used                     13.000000
Number of Observations Used   118.000000
Critical Value (1%)            -3.487022
Critical Value (5%)            -2.886363
Critical Value (10%)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

Figure 115: Rose | Test for Stationarity of the training dataset

- We see that at **5% significant level the Time Series is non-stationary as p-value is 0.2195** which is **more than alpha value (0.05)**, therefore we **fail to reject the null hypothesis.**
- Also, by looking at the plot, it is evident that rolling std. deviation is not constant over time.

- Let us take **one level of differencing** to see whether the series becomes stationary & **carry out the test again.** Below is the output**: -**




```
Results of Dickey-Fuller Test:
Test Statistic                -6.592372e+00
p-value                        7.061944e-09
#Lags Used                     1.200000e+01
Number of Observations Used    1.180000e+02
Critical Value (1%)           -3.487022e+00
Critical Value (5%)           -2.886363e+00
Critical Value (10%)          -2.580009e+00
dtype: float64
```

Figure 116: Rose | Test for Stationarity of the training dataset (d=1)

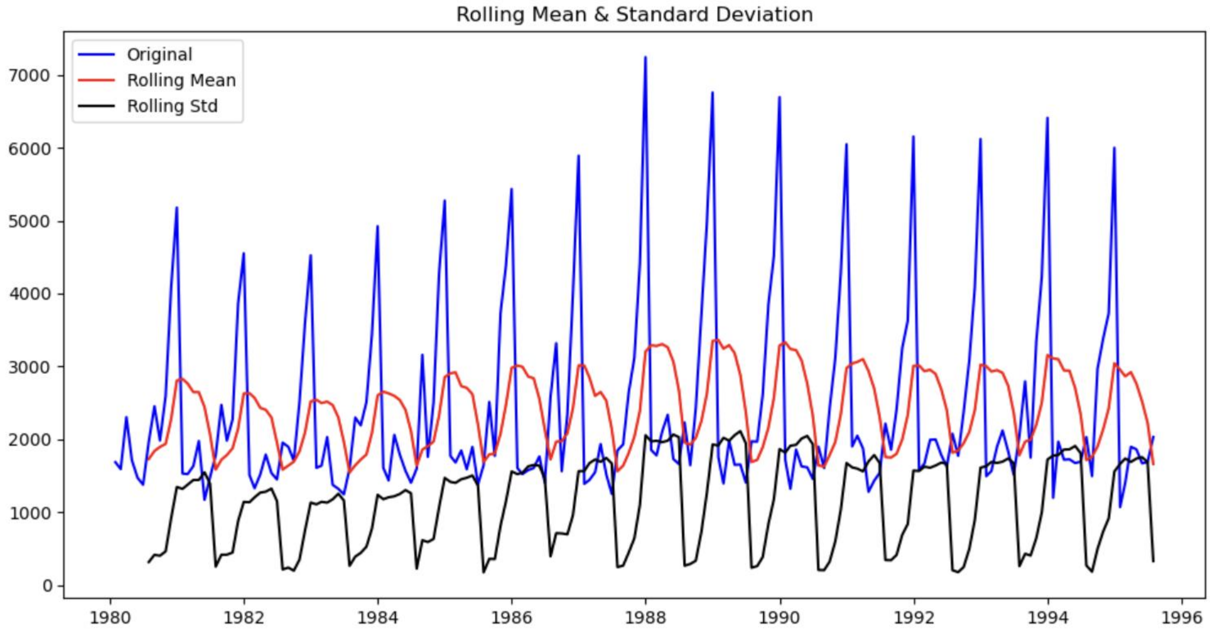
- We see that at 5% significant level the **Time Series becomes stationary as p-value is 7.062e-9 which is less than alpha value (0.05)**, therefore we reject the null hypothesis. We can see that the provided **time series becomes stationary with differencing**.
- Also, by looking at the plot, it is evident that rolling std. deviation is constant over time.

Checking Stationarity of Entire Data – Sparkling Wine

- A Time Series is considered to be stationary when **statistical properties such as the variance and (auto) correlation are constant over time.** Stationary Time Series allows us to think of the statistical properties of the time series as not changing in time, which enables us to build appropriate statistical models for forecasting based on past data.
- Stationarity means that the **autocorrelation of lag 'k' depends on k, but not on time t.**
- The **Augmented Dickey-Fuller test** is a unit root test which determines whether there is a unit root and subsequently whether the series is non-stationary.
- Framing the **hypothesis**: -

  H0: The Time Series has a unit root and is thus non-stationary.

  H1: The Time Series does not have a unit root and is thus stationary.
- The series have to be **stationary for building ARIMA/SARIMA** models and thus we would want the **p-value of this test to be less than the α** value.
- Below is the output of the test: -

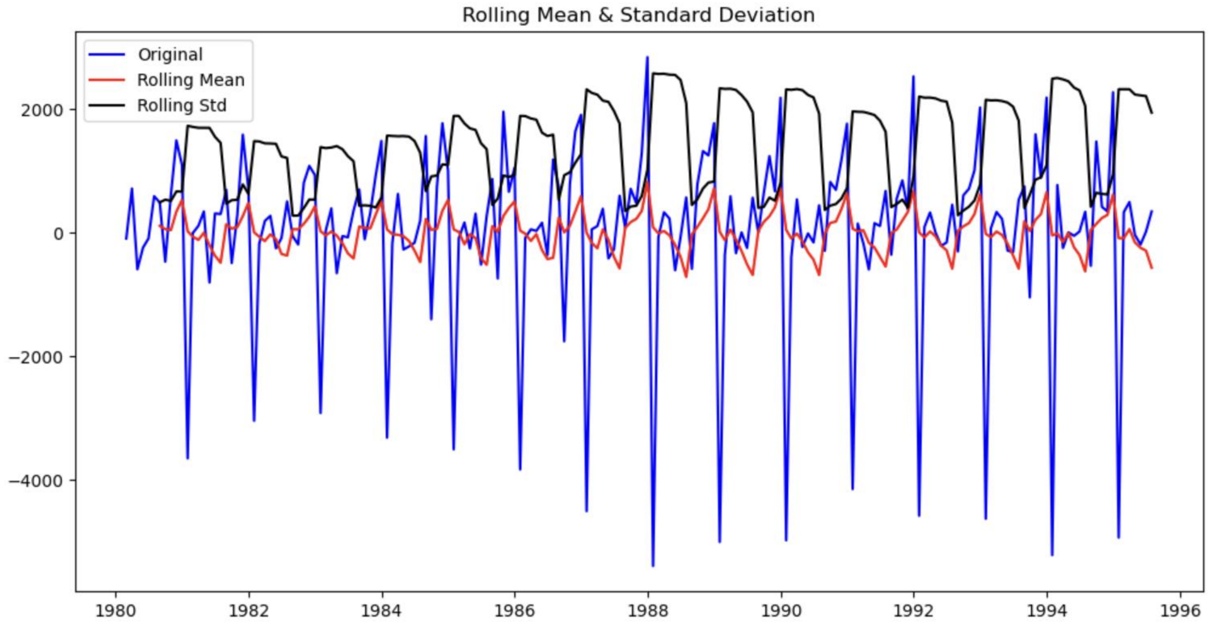



```
Results of Dickey-Fuller Test:
Test Statistic                  -1.360497
p-value                          0.601061
#Lags Used                      11.000000
Number of Observations Used    175.000000
Critical Value (1%)             -3.468280
Critical Value (5%)             -2.878202
Critical Value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

Figure 117: Sparkling | Test for Stationarity of the entire dataset

- We see that at **5% significant level the Time Series is non-stationary as p-value is 0.6011** which is **more than alpha value (0.05)**, therefore we **fail to reject the null hypothesis.**
- Also, by looking at the plot, it is evident that rolling std. deviation is not constant over time.

- Let us take **one level of differencing** to see whether the series becomes stationary & **carry out the test again.** Below is the output**: -**




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

Figure 118: Sparkling | Test for Stationarity of the entire dataset (d=1)

- We see that at 5% significant level the **Time Series becomes stationary as p-value is 0.0000 which is less than alpha value (0.05)**, therefore we reject the null hypothesis. We can see that the provided **time series becomes stationary with differencing**.
- Also, by looking at the plot, it is evident that rolling std. deviation is constant over time.

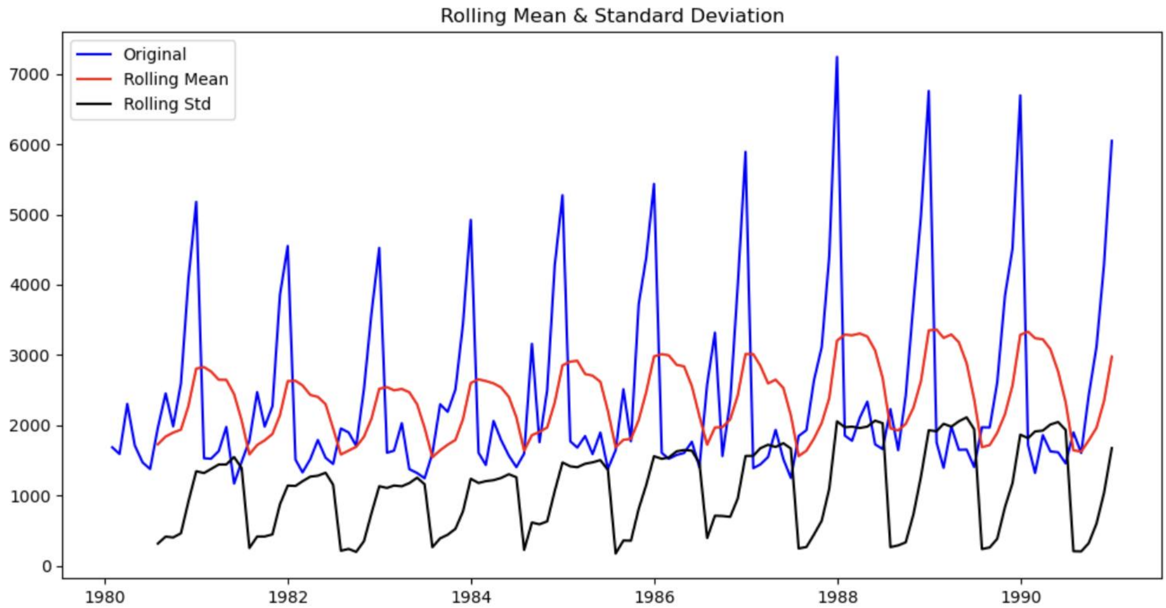
Checking Stationarity of Training Data – Sparkling Wine

- Framing the **hypothesis**: -

  H0: The Time Series has a unit root and is thus non-stationary.

  H1: The Time Series does not have a unit root and is thus stationary.

- Below is the output of the test: -

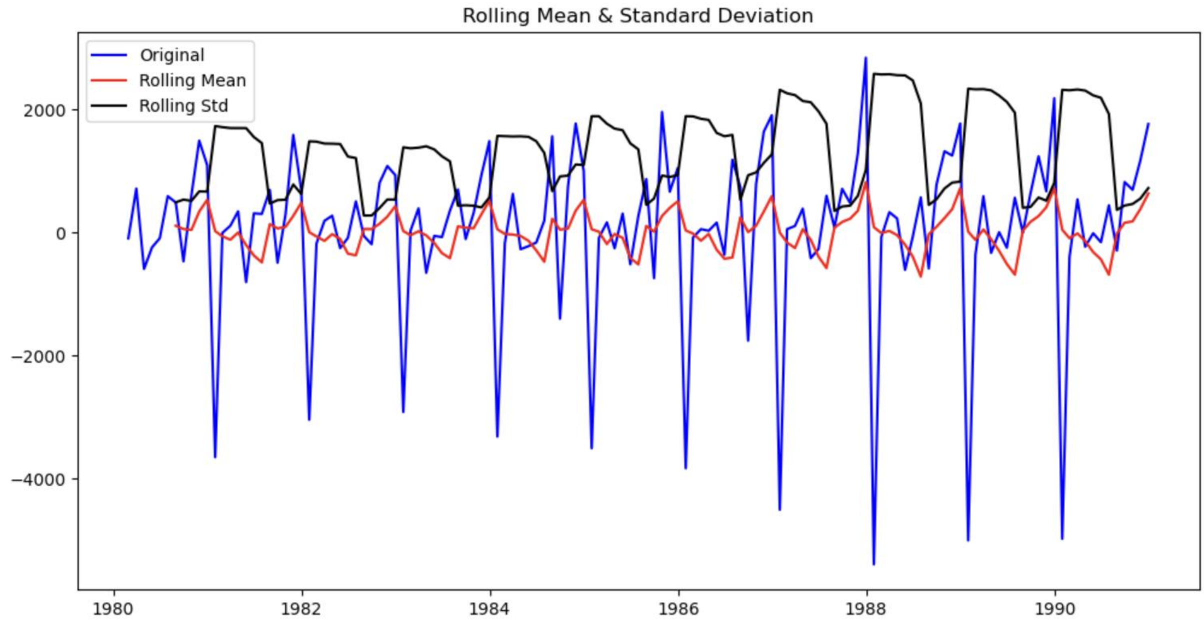



```
Results of Dickey-Fuller Test:
Test Statistic                 -1.208926
p-value                         0.669744
#Lags Used                     12.000000
Number of Observations Used   119.000000
Critical Value (1%)            -3.486535
Critical Value (5%)            -2.886151
Critical Value (10%)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

Figure 119: Sparkling | Test for Stationarity of the training dataset

- We see that at **5% significant level the Time Series is non-stationary as p-value is 0.6697** which is **more than alpha value (0.05)**, therefore we **fail to reject the null hypothesis.**
- Also, by looking at the plot, it is evident that rolling std. deviation is not constant over time.

- Let us take **one level of differencing** to see whether the series becomes stationary & **carry out the test again.** Below is the output**: -**




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

Figure 120: Sparkling | Test for Stationarity of the training dataset (d=1)

- We see that at 5% significant level the **Time Series becomes stationary as p-value is 2.2801e-12 which is less than alpha value (0.05)**, therefore we reject the null hypothesis. We can see that the provided **time series becomes stationary with differencing**.
- Also, by looking at the plot, it is evident that rolling std. deviation is constant over time.

Rubric Question 5: Model Building – Stationary Data

- As per the Augmented Dicky-Fuller test, we observed that the **time series data by itself is not stationary**, however, it becomes **stationary when differencing is done**. The same thing is also observed with Training data.
- Therefore, for training the **models**, it can be **built with order of difference d=1**.

Generate ACF & PACF Plots – Rose/Sparkling Wine

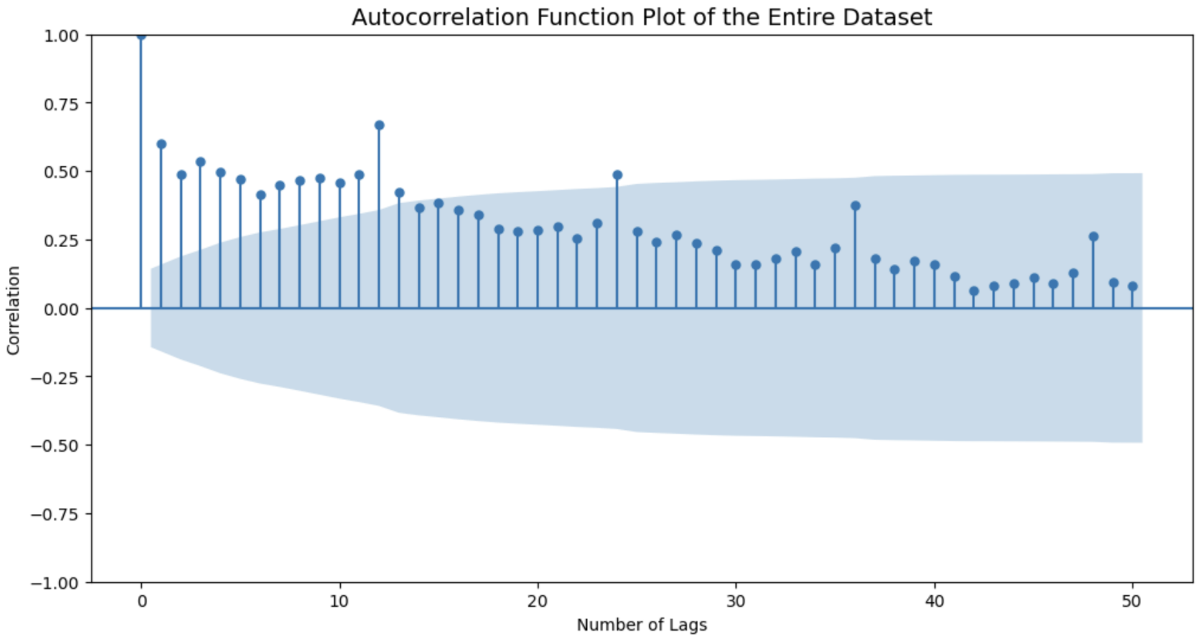


Figure 121: Rose | ACF Plot

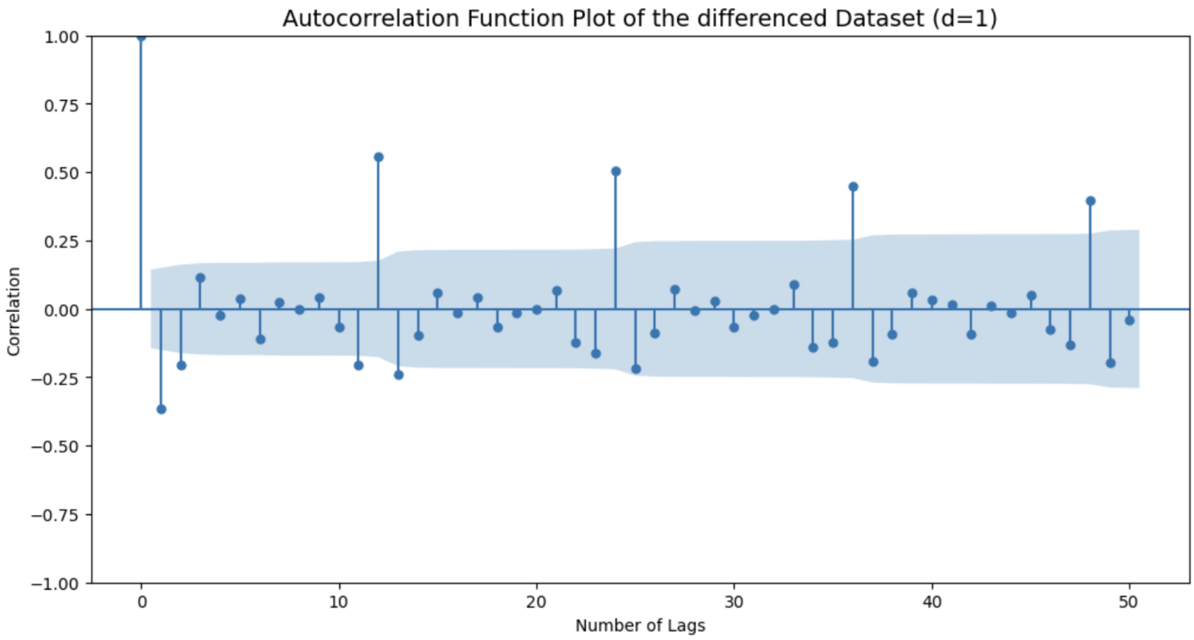


Figure 122: Rose | ACF Plot (d=1)

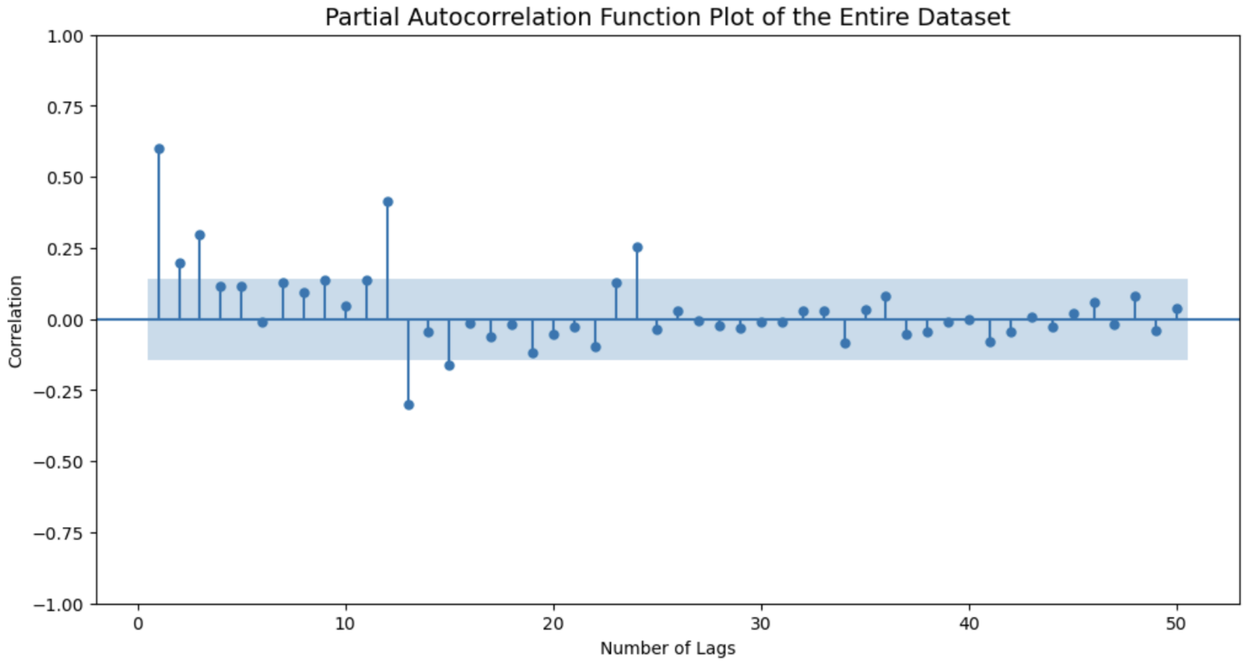


Figure 123: Rose | PACF Plot

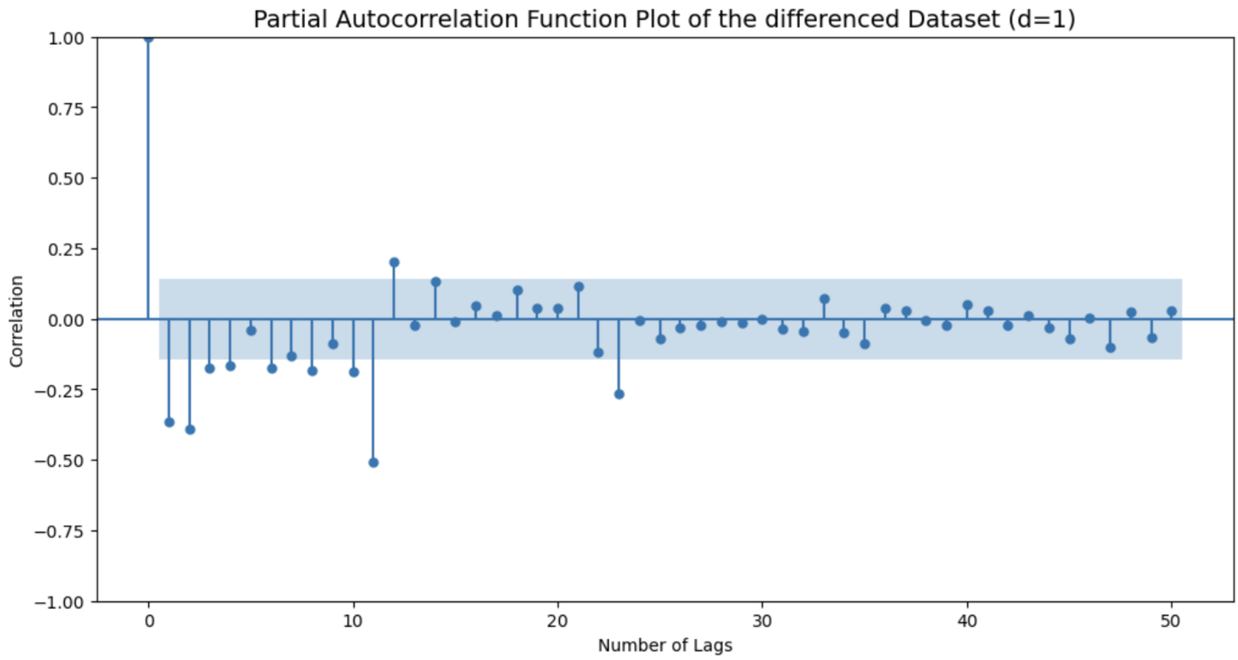


Figure 124: Rose | PACF Plot (d=1)

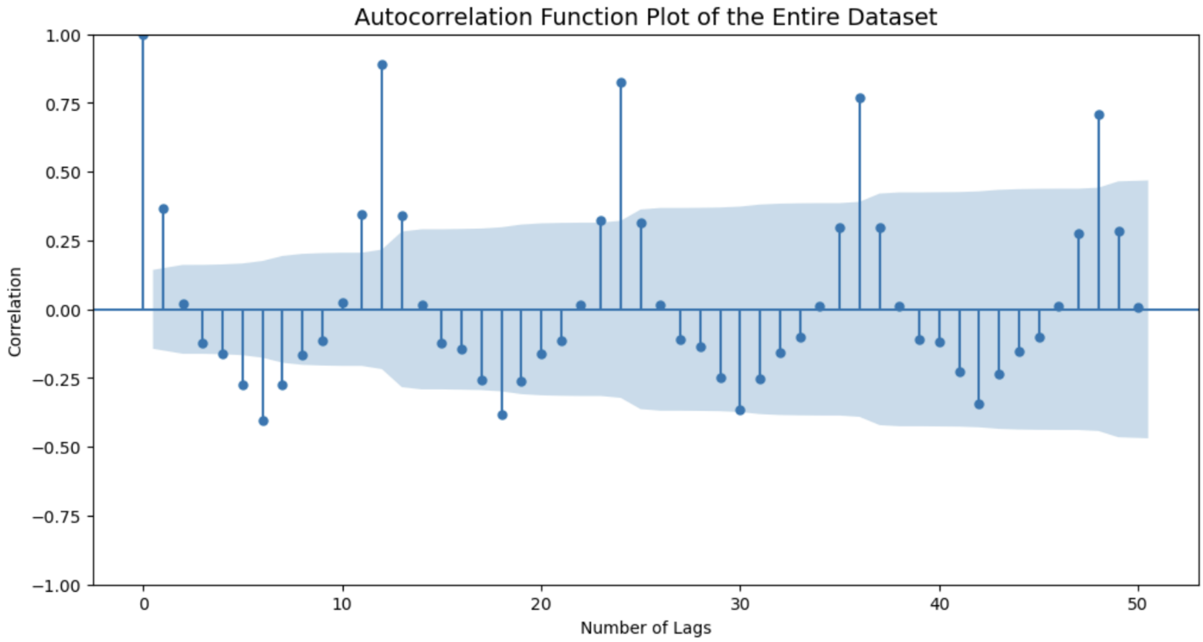


Figure 125: Sparkling | ACF Plot

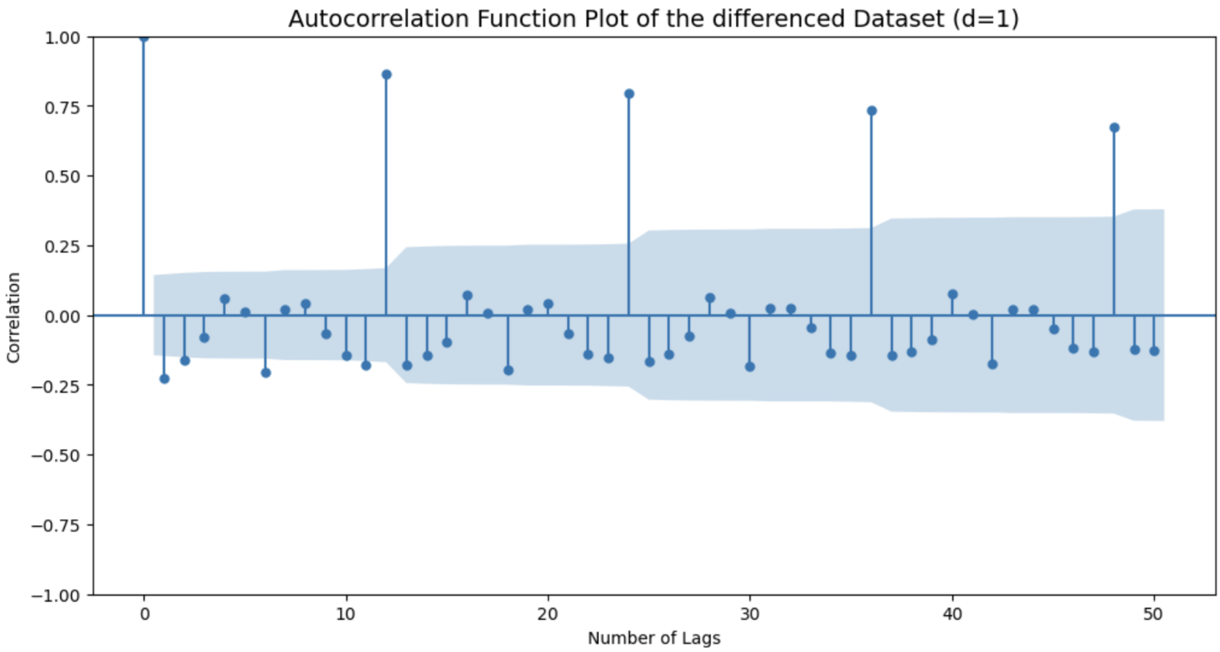


Figure 126: Sparkling | ACF Plot (d=1)

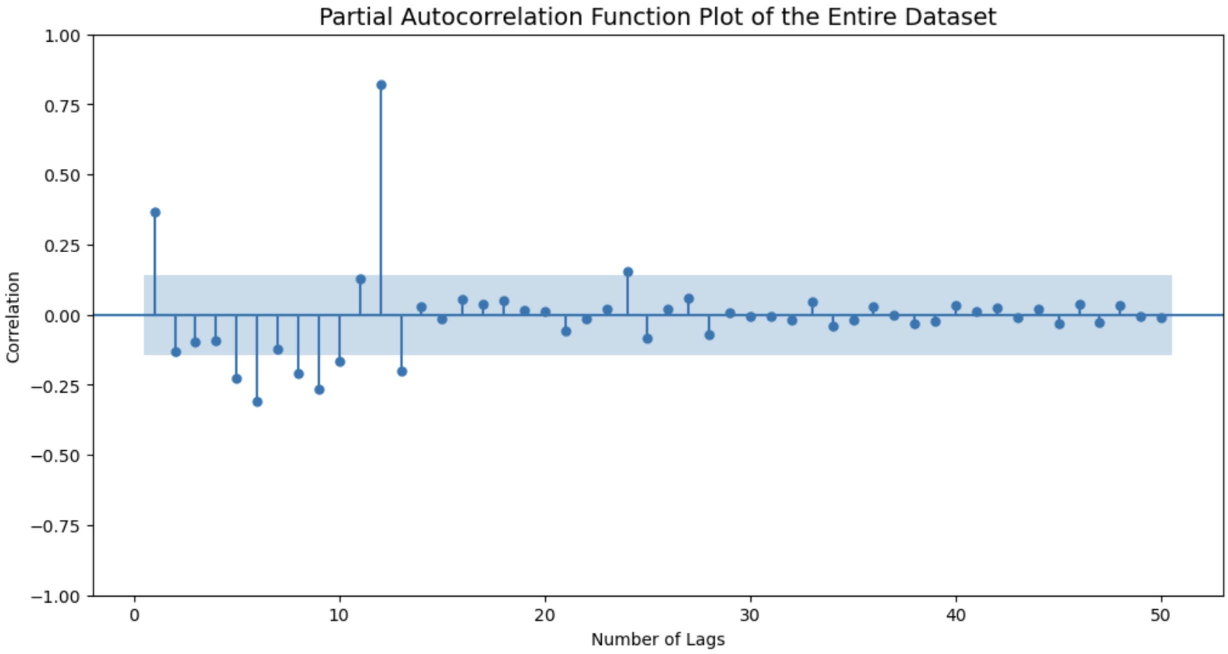


Figure 127: Sparkling | PACF Plot

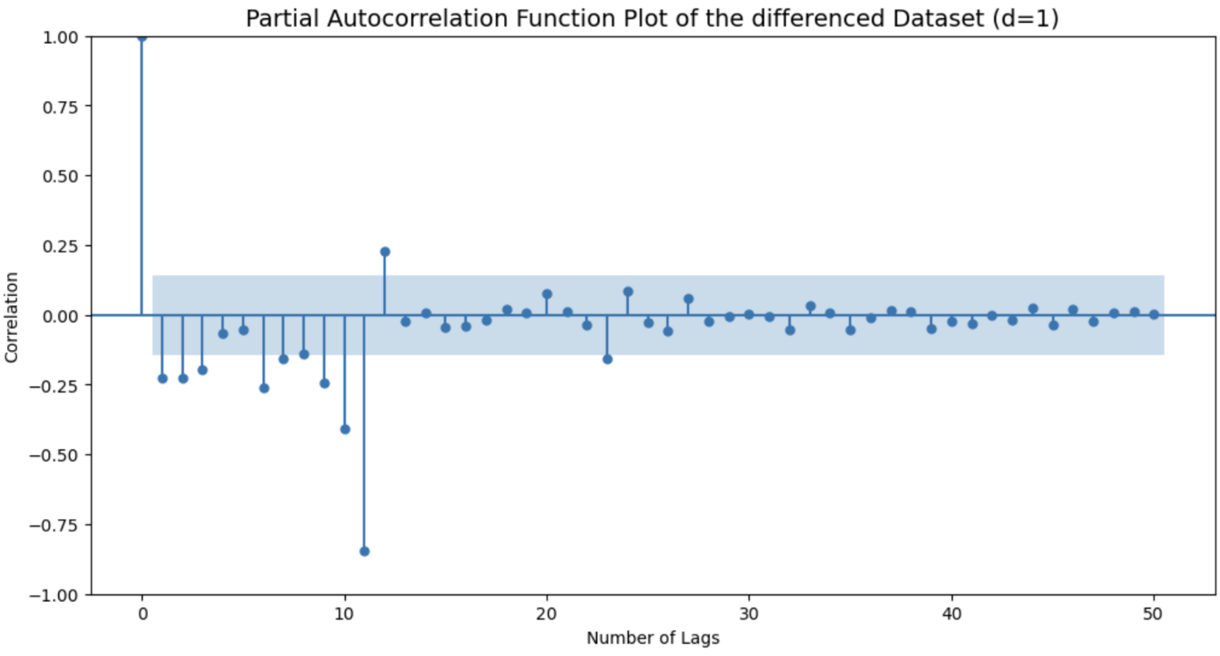


Figure 128: Sparkling | PACF Plot (d=1)

- **Observations: -**
    - **Rose | ACF:** Time series is highly autocorrelated, especially at short lags. There is clear and strong seasonality at 12-month intervals, supporting what we've seen in the decomposition.
    - **Rose | ACF (d=1):** Most autocorrelations are now within the confidence interval & the series has been made stationary in mean, as desired. Seasonality remains and would need to be addressed (via SARIMA). ACF does not gradually taper off, which supports the conclusion that the trend was successfully removed.
    - **Rose | PACF:** First lag has a strong partial correlation (~0.65) and is well outside the confidence interval. Indicates weaker AR relationships beyond lag 1. Indicates evidence of seasonality (12). Classic sign of a pure AR (1) process, where the PACF drops sharply after the first lag.
    - **Rose | PACF (d=1):** Lag 1 shows a strong, significant partial autocorrelation, suggesting a strong AR (1) component even after removing trend. Sharp drop after lag 1, i.e. is a classic indicator of a pure AR (1) process.
    - **Sparkling | ACF:** High autocorrelation at lag 1 (~0.85). There is Clear spikes around multiples of 12. Autocorrelation doesn't immediately cut off, suggesting presence of trend and non-stationarity, meaning differencing is likely needed.
    - **Sparkling | ACF (d=1):** Most autocorrelations decay quickly and are near zero that confirms that differencing (d = 1) successfully removed the trend and stabilized the mean of the time series. Clear peaks around every 12 lags, suggesting strong annual seasonality. First lag (lag 1) just above the confidence interval.
    - **Sparkling | PACF:** Lag 1 is moderately significant. Clear indicator of strong annual seasonality. Beyond lag 1 and lag 12, most partial autocorrelations are statistically insignificant.
    - **Sparkling | PACF (d=1):** Lag 1 shows a very high partial autocorrelation (~0.9). PACF cuts off after lag 1.

Auto ARIMA – Rose Wine

- Autoregression means **regression of a variable on itself** which means **Autoregressive models use previous time period values to predict the current time period values.** One of the fundamental **assumptions** of an AR model is that the **time series is assumed to be a stationary** process**.**
- **An ARIMA model is characterized by 3 terms: p, d, q: -**
  - p is the order of the Auto Regressive (AR) term
  - q is the order of the Moving Average (MA) term
  - d is the number of differencing required to make the time series stationary
- For the **selection criteria of p, d, q** the ARIMA model is built by using **automated model parameters with lowest Akaike Information Criteria** by using the ARIMA () function in python**.**
- Since, we know 1st order differencing makes the series stationary, we consider **d =1** & try different combinations of **p & q in the range of 0 to 5.**
- Below is the output of top combinations whose model gave the least AIC: -

	param	AIC
13	(2, 1, 3)	1274.695440
23	(4, 1, 3)	1278.451412
18	(3, 1, 3)	1278.657996
14	(2, 1, 4)	1278.768576
9	(1, 1, 4)	1279.605264

Figure 129: Rose | Auto ARIMA pdq-AIC combinations

- As evident above, (2,1,3) gives the most optimum model. Building the model with this combination: -

```
                               SARIMAX Results
==============================================================================
Dep. Variable:          RoseWine_Sales   No. Observations:                  132
Model:                  ARIMA(2, 1, 3)   Log Likelihood                -631.348
Date:                 Sun, 06 Apr 2025   AIC                           1274.695
Time:                         14:11:42   BIC                           1291.947
Sample:                     01-31-1980   HQIC                          1281.705
                          - 12-31-1990
Covariance Type:                   opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.6781      0.084    -20.035      0.000      -1.842      -1.514
ar.L2         -0.7291      0.084     -8.706      0.000      -0.893      -0.565
ma.L1          1.0444      0.618      1.691      0.091      -0.166       2.255
ma.L2         -0.7721      0.132     -5.871      0.000      -1.030      -0.514
ma.L3         -0.9047      0.560     -1.616      0.106      -2.002       0.193
sigma2       859.4180    519.391      1.655      0.098    -158.570    1877.406
===================================================================================
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):                24.51
Prob(Q):                              0.88   Prob(JB):                         0.00
Heteroskedasticity (H):               0.40   Skew:                             0.71
Prob(H) (two-sided):                  0.00   Kurtosis:                         4.57
===================================================================================
```

Figure 130: Rose | Auto ARIMA Model Summary (2,1,3)

- **Observations: -**
  - p=2: Two autoregressive (AR) terms
  - d=1: First differencing to achieve stationarity
  - q=3: Three moving average (MA) terms
  - AR terms (L1 & L2) are highly significant ($p < 0.001$)
  - MA terms: L2 is strongly significant, L1 and L3 are not significant at 95% confidence.
  - The model fits the data well.
  - No autocorrelation in residuals (Ljung-Box $p > 0.05$).

- Below is the Diagnostic Plot: -

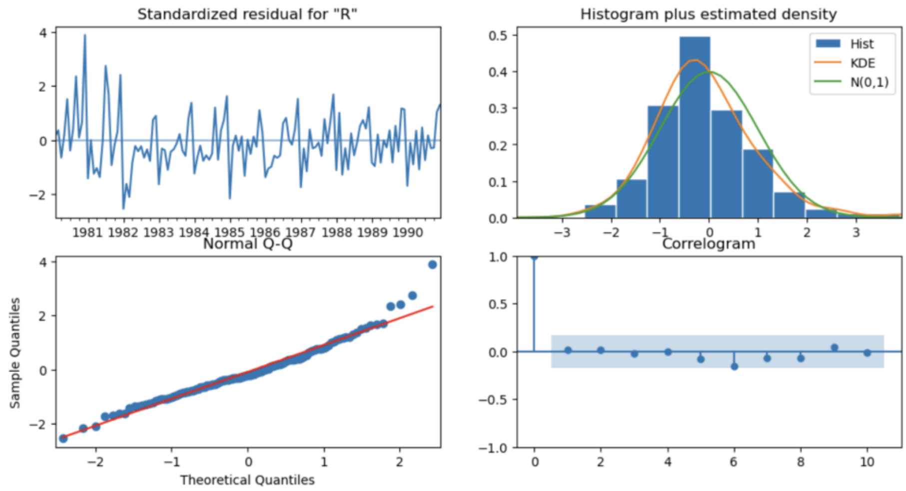


Figure 131: Rose | Auto ARIMA Diagnostic Plot

- **Observations: -**
  - **Standardized Residuals (Time Series Plot):** Residuals fluctuate around zero, as expected. Some spikes (e.g., 1980–1981), but no sustained structure.
  - **Histogram with KDE and Normal Curve:** The residual distribution is approximately normal (though, slightly right skewed), reasonably bell-shaped and can be acceptable in forecasting context
  - **Q-Q Plot:** Points lie mostly on the red line, indicating approximate normality.
  - **Correlogram:** Residuals are not auto-correlated, meaning model has captured the temporal structure well.

- Below is the Plot forecasting predicted future values: -

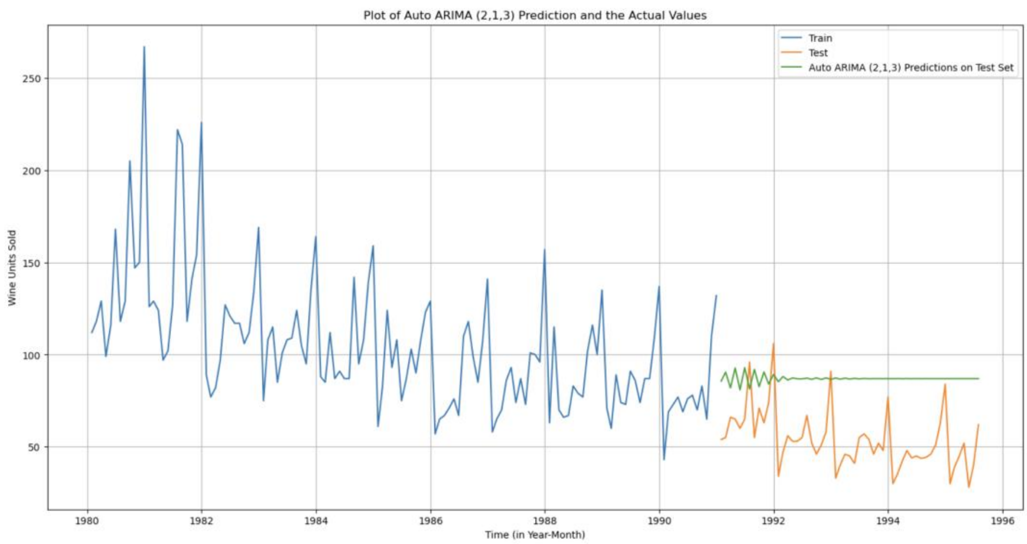


Figure 132: Rose | Auto ARIMA Forecast Plot

- **Model Evaluation –** Below is the RMSE for Auto ARIMA (2,1,3) Model: -

```
For Auto ARIMA (2,1,3) on the Test Data,  RMSE is 36.875
```

Figure 133: Rose | Auto ARIMA RMSE

- **Observations: -**
  - Clearly, the model **is unable to capture the entire characteristics** of the test data as ARIMA models perform well only on non-seasonal time series.
  - Not surprisingly, the **RMSE** of the aforementioned ARIMA model is **higher than the majority of previously constructed models**.
  - The root means squared error (**RMSE**) of test data for the **Auto ARIMA model (2,1,3)** is **36.875.**

Auto ARIMA – Sparkling Wine

- Autoregression means **regression of a variable on itself** which means **Autoregressive models use previous time period values to predict the current time period values.** One of the fundamental **assumptions** of an AR model is that the **time series is assumed to be a stationary** process**.**
- **An ARIMA model is characterized by 3 terms: p, d, q: -**
  - p is the order of the Auto Regressive (AR) term
  - q is the order of the Moving Average (MA) term
  - d is the number of differencing required to make the time series stationary
- For the **selection criteria of p, d, q** the ARIMA model is built by using **automated model parameters with lowest Akaike Information Criteria** by using the ARIMA () function in python**.**
- Since, we know 1st order differencing makes the series stationary, we consider **d =1** & try different combinations of **p & q in the range of 0 to 5.**
- Below is the output of top combinations whose model gave the least AIC: -

	param	AIC
12	(2, 1, 2)	2213.509213
24	(4, 1, 4)	2214.059617
19	(3, 1, 4)	2220.209574
18	(3, 1, 3)	2221.436801
23	(4, 1, 3)	2222.904096

Figure 134: Sparkling | Auto ARIMA pdq-AIC combinations

- As evident above, (2,1,2) gives the most optimum model. Building the model with this combination: -

```
                               SARIMAX Results
==============================================================================
Dep. Variable:     SparklingWine_Sales   No. Observations:                  132
Model:                  ARIMA(2, 1, 2)   Log Likelihood               -1101.755
Date:                 Sun, 06 Apr 2025   AIC                           2213.509
Time:                         14:07:06   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    -18.216      0.000      -2.206      -1.777
ma.L2          0.9999      0.110      9.108      0.000       0.785       1.215
sigma2      1.099e+06   1.99e-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
===================================================================================
```

Figure 135: Sparkling | Auto ARIMA Model Summary (2,1,2)

- **Observations: -**
  - p=2: Two autoregressive (AR) terms
  - d=1: First differencing to achieve stationarity
  - q=2: Two moving average (MA) terms
  - All AR and MA terms are statistically significant. The model successfully captures both short-term autoregressive and moving average structure
  - The model fits the data well.
  - No autocorrelation in residuals (Ljung-Box $p > 0.05$).

- Below is the Diagnostic Plot: -

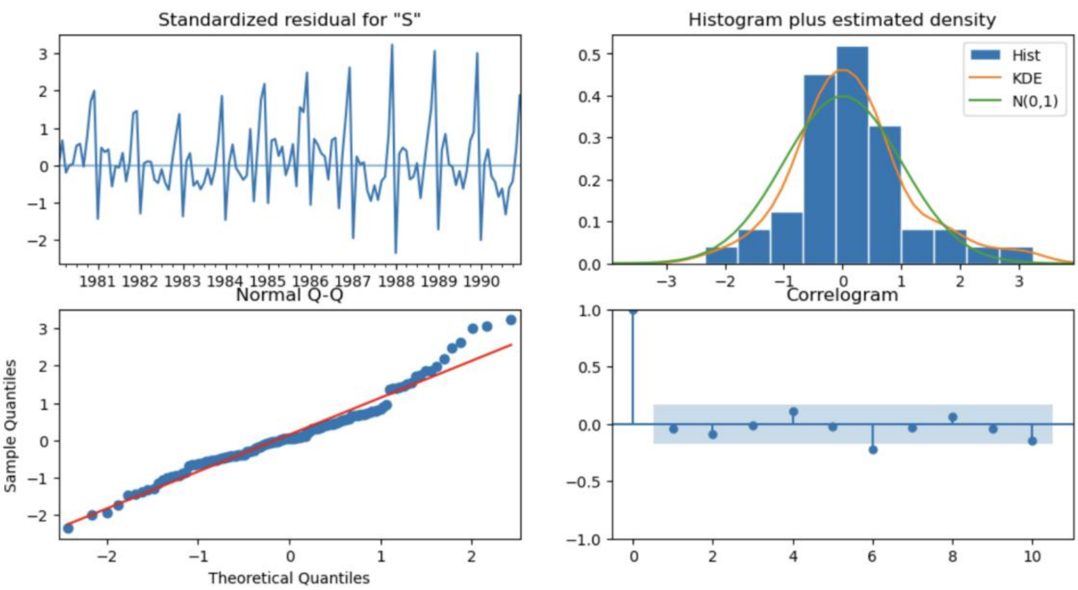


Figure 136: Sparkling | Auto ARIMA Diagnostic Plot

- **Observations: -**
  - **Standardized Residuals (Time Series Plot):** Residuals fluctuate around zero, as expected. Some spikes (e.g., 1980–1981), but no sustained structure.
  - **Histogram with KDE and Normal Curve:** Residuals are approximately bell-shaped, cantered around zero. Slight right skew, but generally a good match to normal distribution.
  - **Q-Q Plot:** Most points lie close to the red line, which suggests the residuals are approximately normally distributed.
  - **Correlogram:** Residuals are not auto-correlated, meaning model has captured the temporal structure well.

- Below is the Plot forecasting predicted future values: -

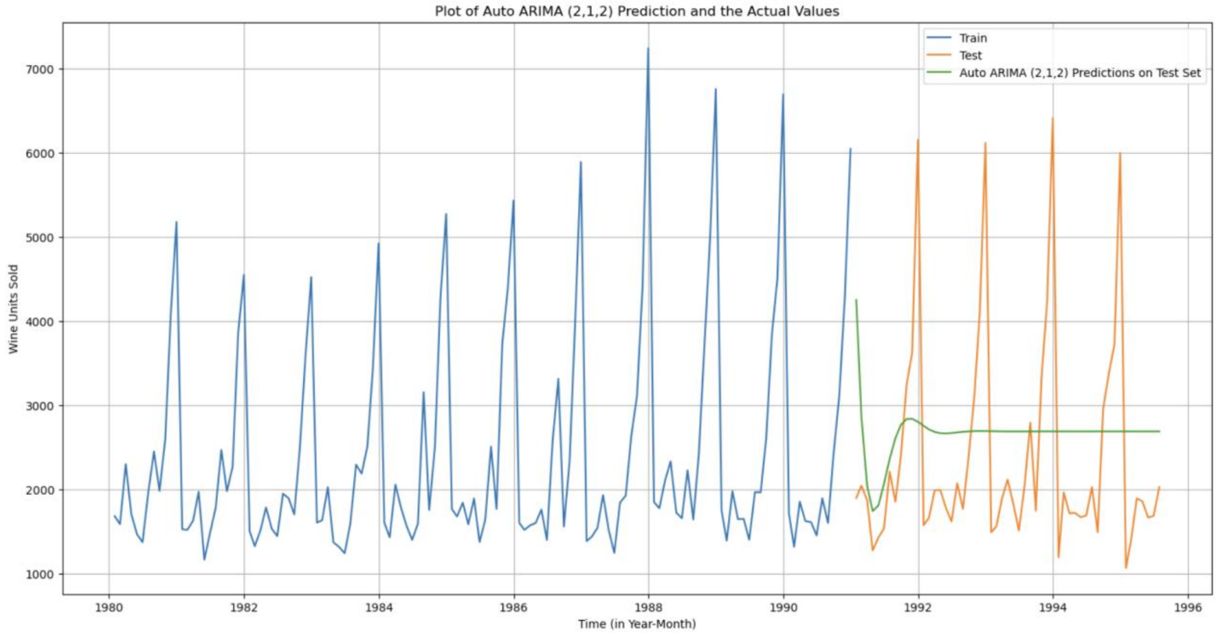


Figure 137: Sparkling | Auto ARIMA Forecast Plot

- **Model Evaluation –** Below is the RMSE for Auto ARIMA (2,1,2) Model: -

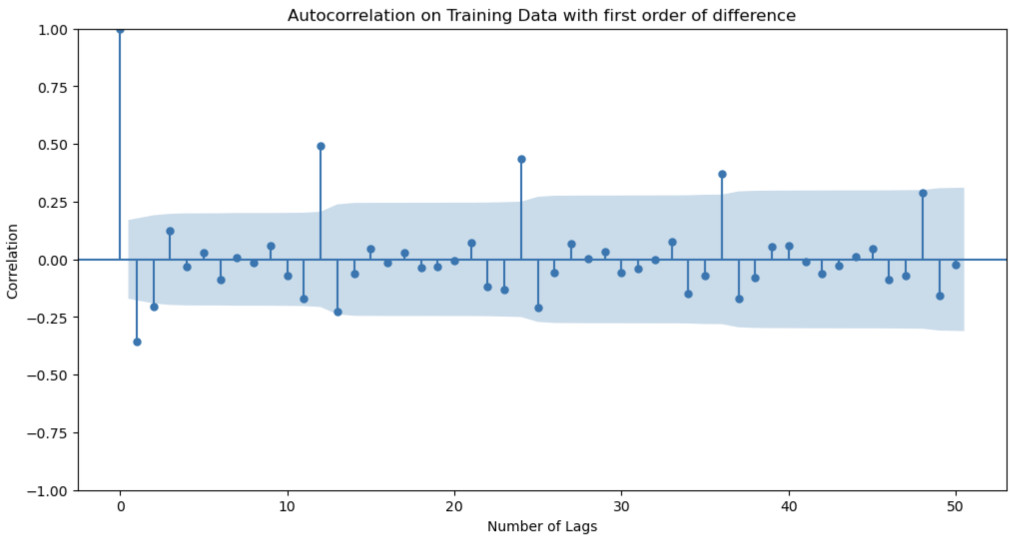
```
For Auto ARIMA (2,1,2) on the Test Data,  RMSE is 1299.980
```

Figure 138: Sparkling | Auto ARIMA RMSE

- **Observations: -**
  - Clearly, the model **is unable to capture the entire characteristics** of the test data as ARIMA models perform well only on non-seasonal time series.
  - Not surprisingly, the **RMSE** of the aforementioned ARIMA model is **higher than the majority of previously constructed models**.
  - The root means squared error (**RMSE**) of test data for the **Auto ARIMA model (2,1,2)** is **1299.980.**

Manual ARIMA – Rose Wine

- Autoregression means **regression of a variable on itself** which means **Autoregressive models use previous time period values to predict the current time period values.** One of the fundamental **assumptions** of an AR model is that the **time series is assumed to be a stationary** process**.**
- **An ARIMA model is characterized by 3 terms: p, d, q: -**
  - p is the order of the Auto Regressive (AR) term
  - q is the order of the Moving Average (MA) term
  - d is the number of differencing required to make the time series stationary
- Since, we know 1^{st} order differencing makes the series stationary, we consider **d =1**
- The parameters **p & q can be determined by looking at the PACF & ACF** plots respectively.
- **Autocorrelation function (ACF)** - At lag *k*, this is the correlation between series values that are *k* intervals apart.
- **Partial autocorrelation function (PACF) -** At lag *k*, this is the correlation between series values that are *k* intervals apart, accounting for the values of the intervals between.
- The Auto-Regressive parameter in an ARIMA model is p which comes from the significant lag after which the PACF plot cuts-off below the confidence interval. The Moving-Average parameter in an ARIMA model is 'q' which comes from the significant lag after which the ACF plot cuts-off below the confidence interval.
- Below are the ACF/PACF plots (on train data with d=1): -



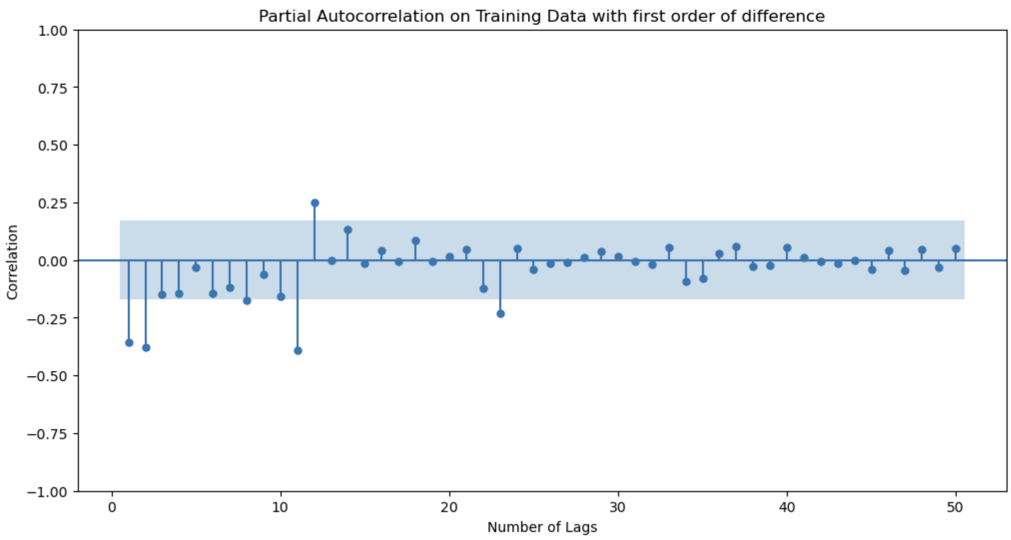


Figure 139: Rose | ACF/PACF Plots on Train data (d=1)

- By looking at ACF/PACF plots, we observe: -
  - ACF shows significant lags up to ~3, suggesting an MA (2) or MA (3)
  - PACF has strong spikes at lags 1 and 2, implying an AR (2) component
  - By manual inspection, suggested parameters are (2,1,2) or (2,1,3)
  - Since, we have already built a model with (2,1,3), we shall go ahead & now **build for (2,1,2):** -

```
                               SARIMAX Results
==============================================================================
Dep. Variable:         RoseWine_Sales   No. Observations:                  132
Model:                 ARIMA(2, 1, 2)   Log Likelihood                -635.935
Date:                Sun, 06 Apr 2025   AIC                           1281.871
Time:                        14:11:47   BIC                           1296.247
Sample:                    01-31-1980   HQIC                          1287.712
                         - 12-31-1990
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.4540      0.469     -0.969      0.333      -1.372       0.464
ar.L2          0.0001      0.170      0.001      0.999      -0.334       0.334
ma.L1         -0.2541      0.459     -0.554      0.580      -1.154       0.646
ma.L2         -0.5984      0.430     -1.390      0.164      -1.442       0.245
sigma2       952.1601     91.424     10.415      0.000     772.973    1131.347
===================================================================================
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):                34.16
Prob(Q):                              0.88   Prob(JB):                         0.00
Heteroskedasticity (H):               0.37   Skew:                             0.79
Prob(H) (two-sided):                  0.00   Kurtosis:                         4.94
===================================================================================
```

Figure 140: Rose | Manual ARIMA Model Summary (2,1,2)

- **Observations: -**
  - p=2: Two autoregressive (AR) terms
  - d=1: First differencing to achieve stationarity
  - q=2: Two moving average (MA) terms
  - None of the AR or MA terms are statistically significant (p-values > 0.05).
  - Suggests the ARIMA (2,1,2) model is overparameterized (might be modelling noise rather than signal) i.e. overfitting.
  - No autocorrelation in residuals (Ljung-Box p > 0.05).

- Below is the Diagnostic Plot: -

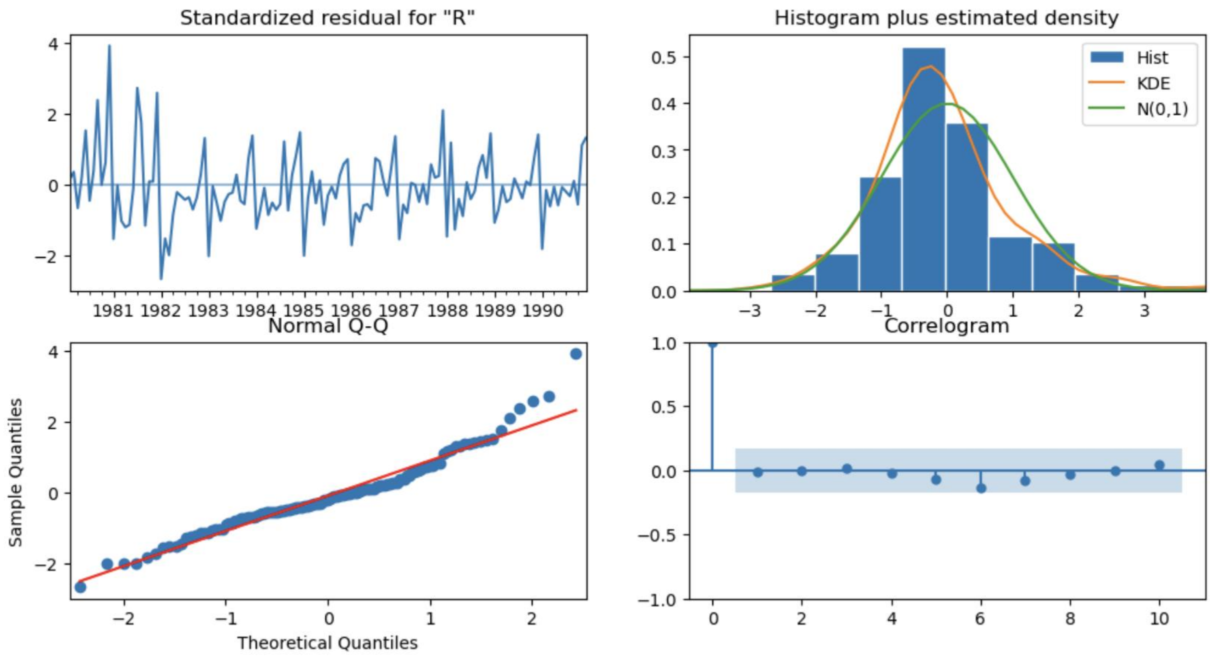


Figure 141: Rose | Manual ARIMA Diagnostic Plot

- **Observations: -**
  - **Standardized Residuals (Time Series Plot):** Residuals fluctuate around zero, as expected. Some spikes (e.g., 1980–1981), but no sustained structure.
  - **Histogram with KDE and Normal Curve:** The residual distribution is approximately normal (though, slightly left skewed), reasonably bell-shaped and can be acceptable in forecasting context
  - **Q-Q Plot:** Most points lie close to the red line, supporting the normality assumption.
  - **Correlogram:** Residuals are mildly autocorrelated, meaning the model has captured the time-dependence structure well.

- Below is the Plot forecasting predicted future values: -

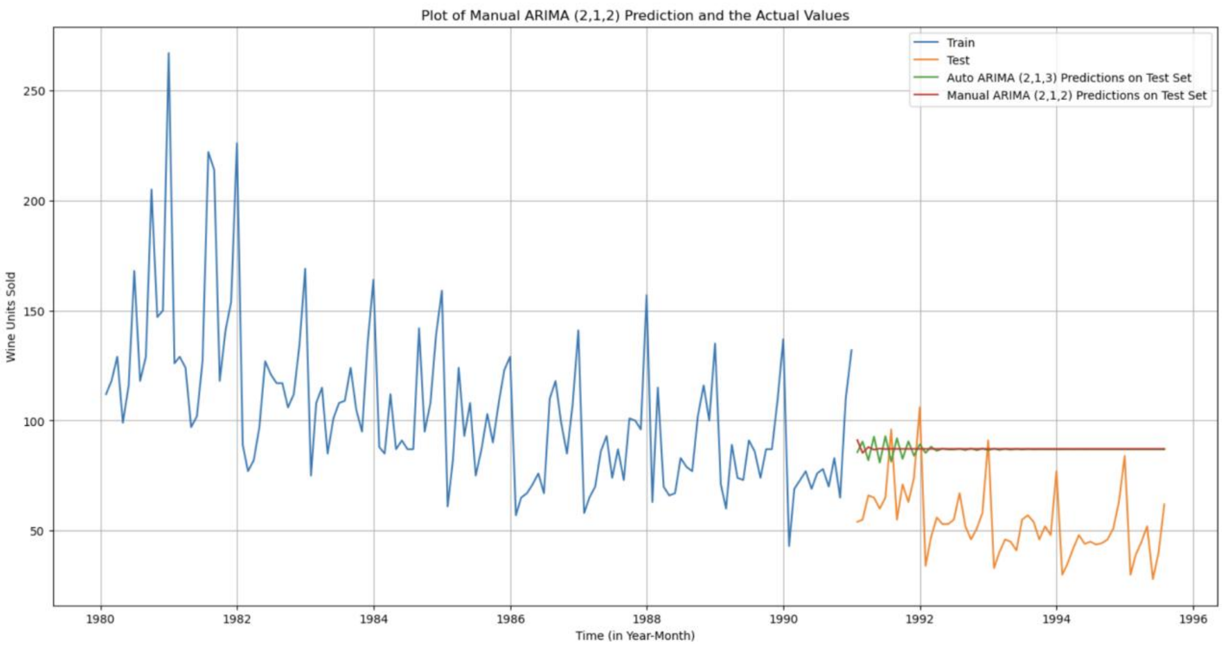


Figure 142: Rose | Manual ARIMA Forecast Plot

- **Model Evaluation –** Below is the RMSE for Auto ARIMA (2,1,2) Model: -

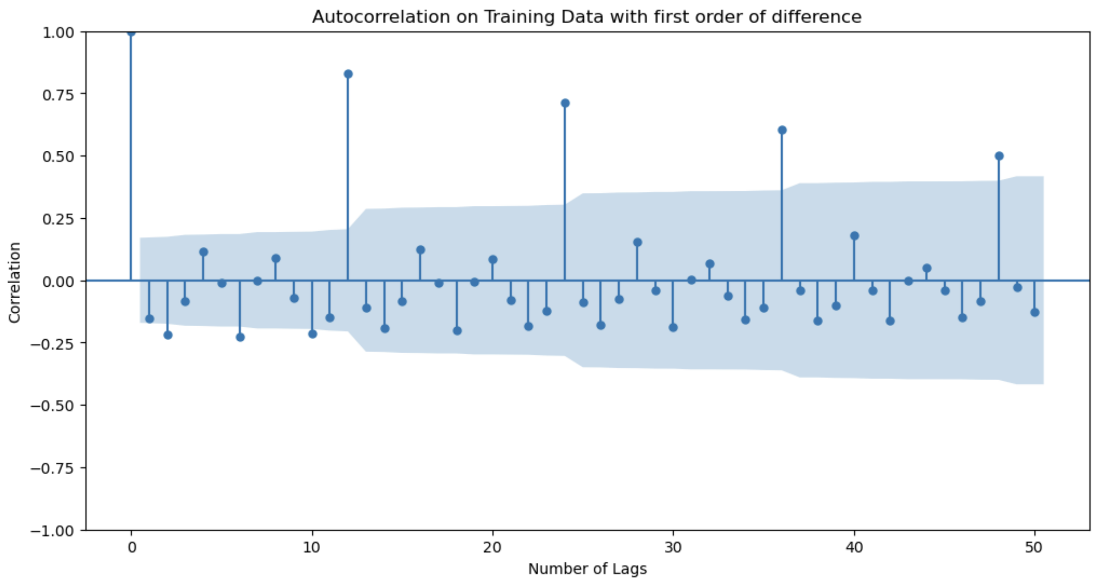
```
For Manual ARIMA (2,1,2) on the Test Data,  RMSE is 36.933
```

Figure 143: Rose | Manual ARIMA RMSE

- **Observations: -**
  - Clearly, the model **is unable to capture the entire characteristics** of the test data as ARIMA models perform well only on non-seasonal time series.
  - Not surprisingly, the **RMSE** of the aforementioned ARIMA model is **higher than the majority of previously constructed models**.
  - The root means squared error (**RMSE**) of test data for the **Manual ARIMA model (2,1,2)** is **36.933**, which is relatively weaker than the Auto ARIMA Model

Manual ARIMA – Sparkling Wine

- Autoregression means **regression of a variable on itself** which means **Autoregressive models use previous time period values to predict the current time period values.** One of the fundamental **assumptions** of an AR model is that the **time series is assumed to be a stationary** process**.**
- **An ARIMA model is characterized by 3 terms: p, d, q: -**
  - p is the order of the Auto Regressive (AR) term
  - q is the order of the Moving Average (MA) term
  - d is the number of differencing required to make the time series stationary
- Since, we know 1st order differencing makes the series stationary, we consider **d =1**
- The parameters **p & q can be determined by looking at the PACF & ACF** plots respectively.
- **Autocorrelation function (ACF)** - At lag *k*, this is the correlation between series values that are *k* intervals apart.
- **Partial autocorrelation function (PACF) -** At lag *k*, this is the correlation between series values that are *k* intervals apart, accounting for the values of the intervals between.
- The Auto-Regressive parameter in an ARIMA model is p which comes from the significant lag after which the PACF plot cuts-off below the confidence interval. The Moving-Average parameter in an ARIMA model is 'q' which comes from the significant lag after which the ACF plot cuts-off below the confidence interval.
- Below are the ACF/PACF plots (on train data with d=1): -



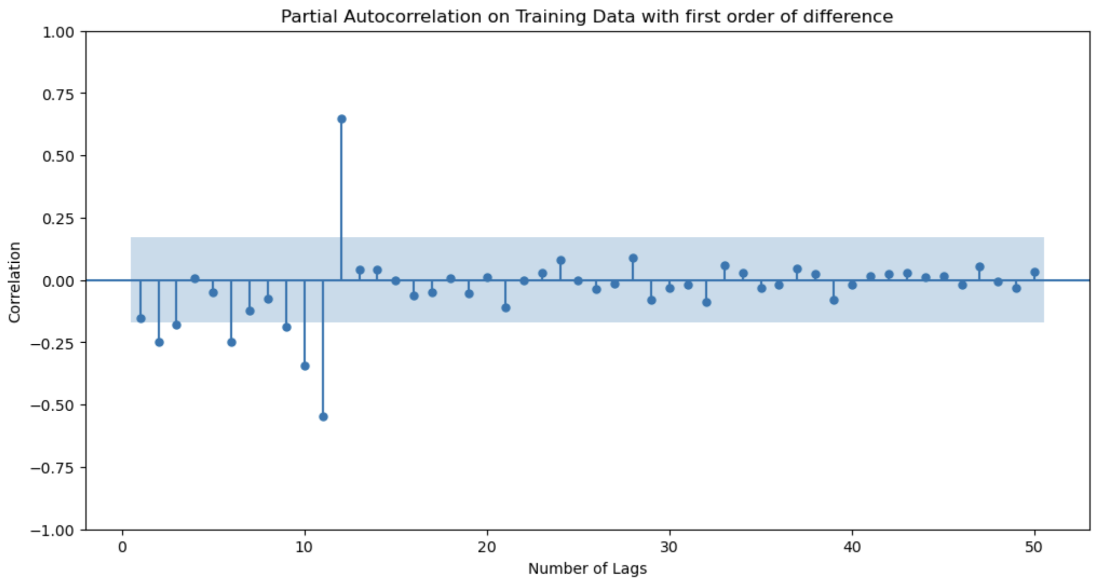


Figure 144: Sparkling | ACF/PACF Plots on Train data (d=1)

- By looking at ACF/PACF plots, we observe: -
  - ACF shows significant lags up to ~3, suggesting an MA (3)
  - PACF has strong spikes at lags 4, implying an AR (4) component
  - By manual inspection, suggested parameters are (4,1,3)
  - We shall go ahead & now **build for (4,1,3): -**

```
                               SARIMAX Results
==============================================================================
Dep. Variable:     SparklingWine_Sales   No. Observations:                  132
Model:                  ARIMA(4, 1, 3)   Log Likelihood               -1103.452
Date:                 Tue, 08 Apr 2025   AIC                           2222.904
Time:                         19:07:17   BIC                           2245.906
Sample:                     01-31-1980   HQIC                          2232.251
                          - 12-31-1990
Covariance Type:                   opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.4570      0.192      2.383      0.017       0.081       0.833
ar.L2         -0.9591      0.325     -2.953      0.003      -1.596      -0.323
ar.L3          0.4640      0.179      2.593      0.010       0.113       0.815
ar.L4          0.0227      0.314      0.072      0.942      -0.592       0.637
ma.L1         -0.9190      0.360     -2.553      0.011      -1.624      -0.214
ma.L2          0.9084      0.360      2.526      0.012       0.203       1.613
ma.L3         -0.9894      0.195     -5.072      0.000      -1.372      -0.607
sigma2      1.494e+06   2.93e-07    5.1e+12      0.000    1.49e+06    1.49e+06
===================================================================================
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):                 2.80
Prob(Q):                              0.89   Prob(JB):                         0.25
Heteroskedasticity (H):               2.73   Skew:                             0.27
Prob(H) (two-sided):                  0.00   Kurtosis:                         3.46
===================================================================================
```

Figure 145: Sparkling | Manual ARIMA Model Summary (4,1,3)

  - **Observations: -**
    - p=4: Four autoregressive (AR) terms
    - d=1: First differencing to achieve stationarity
    - q=3: Three moving average (MA) terms
    - model fits well, and diagnostics are mostly acceptable
- Below is the Diagnostic Plot: -

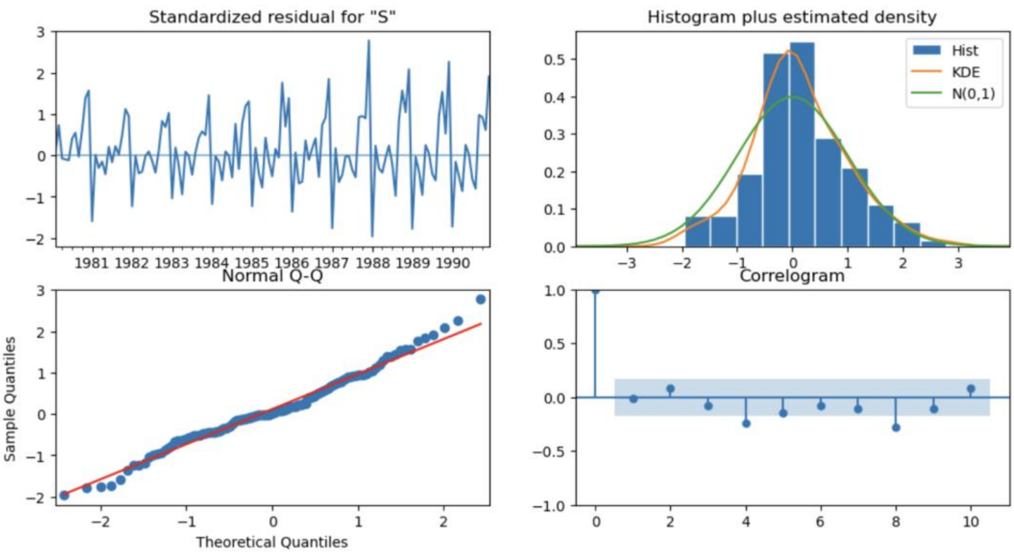


Figure 146: Sparkling | Manual ARIMA Diagnostic Plot

- **Observations: -**
  - **Standardized Residuals (Time Series Plot):** Residuals fluctuate around zero, as expected.
  - **Histogram with KDE and Normal Curve:** Residuals are approximately normally distributed.
  - **Q-Q Plot:** Good fit to normality, with mild kurtosis.
  - **Correlogram:** All autocorrelations are within the confidence band, meaning the model has captured the time-dependence structure well.

- Below is the Plot forecasting predicted future values: -

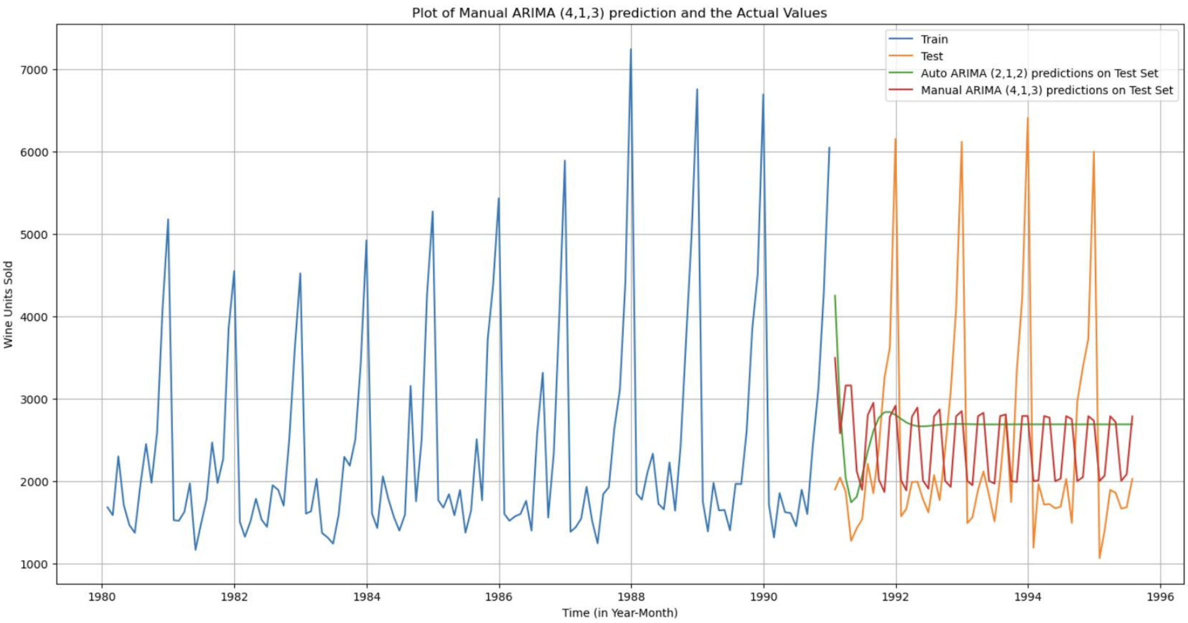


Figure 147: Sparkling | Manual ARIMA Forecast Plot

- **Model Evaluation –** Below is the RMSE for Manual ARIMA (4,1,3) Model: -

```
For Manual ARIMA (4,1,3) on the Test Data,  RMSE is 1232.597
```

Figure 148: Sparkling | Manual ARIMA RMSE

- **Observations: -**
  - Clearly, the model **is unable to capture the entire characteristics** of the test data as ARIMA models perform well only on non-seasonal time series.
  - Not surprisingly, the **RMSE** of the aforementioned ARIMA model is **higher than the majority of previously constructed models**.
  - The root means squared error (**RMSE**) of test data for the **Manual ARIMA model (4,1,3)** is **1232.597**, which is relatively weaker than the Auto ARIMA Model

Auto SARIMA – Rose Wine

- SARIMA models are an extension of ARIMA for a time series data with defined seasonality. SARIMA models use seasonal differencing which is similar to regular differencing. An ARIMA model is characterized by 7 terms: p, d, q, P, D, Q, F: -
  - p is the order of the Auto Regressive (AR) term
  - q is the order of the Moving Average (MA) term
  - d is the number of differencing required to make the time series stationary
  - P is the order of the Seasonal Auto Regressive (AR) term
  - Q is the order of the Seasonal Moving Average (MA) term
  - D is the number of seasonal differencing required to make the time series stationary
  - F is the seasonal frequency of the time series
- For the **selection criteria of p, d, q, P, D, Q, F** the ARIMA model is built by using **automated model parameters with lowest Akaike Information Criteria** by using the SARIMAX () function in python**.**
- Since, we know 1st order differencing makes the series stationary, we consider **d =1**.
- We shall use **D=0** as we have shown already that the series is stationary at d=1.
- By looking at the ACF/PACF plots, it is clear that the pattern (spike) is repeating every 12 months. So, **F = 12.**
- We will try different combinations of **p, P & q, Q in the range of 0 to 4.**
- Below is the output of top combinations whose model gave the least AIC: -

	param	seasonal	AIC
222	(3, 1, 1)	(3, 0, 2, 12)	774.400286
238	(3, 1, 2)	(3, 0, 2, 12)	774.880935
220	(3, 1, 1)	(3, 0, 0, 12)	775.426699
221	(3, 1, 1)	(3, 0, 1, 12)	775.495330
252	(3, 1, 3)	(3, 0, 0, 12)	775.561018

Figure 149: Rose | Auto SARIMA param-AIC combinations

- As evident above, (3,1,1) x (3,0,2,12) | (p,d,q) x (P,D,Q,F), gives the most optimum model. Building the model with this combination: -

```
                                        SARIMAX Results
================================================================================================
Dep. Variable:                              RoseWine_Sales   No. Observations:              132
Model:             SARIMAX(3, 1, 1)x(3, 0, [1, 2], 12)   Log Likelihood            -377.200
Date:                                  Tue, 08 Apr 2025   AIC                        774.400
Time:                                          19:43:30   BIC                        799.618
Sample:                                      01-31-1980   HQIC                       784.578
                                           - 12-31-1990
Covariance Type:                                    opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.0464      0.126      0.367      0.714      -0.202       0.294
ar.L2         -0.0060      0.120     -0.050      0.960      -0.241       0.229
ar.L3         -0.1808      0.098     -1.838      0.066      -0.374       0.012
ma.L1         -0.9370      0.067    -13.905      0.000      -1.069      -0.805
ar.S.L12       0.7639      0.165      4.640      0.000       0.441       1.087
ar.S.L24       0.0840      0.159      0.527      0.598      -0.229       0.397
ar.S.L36       0.0727      0.095      0.764      0.445      -0.114       0.259
ma.S.L12      -0.4969      0.250     -1.988      0.047      -0.987      -0.007
ma.S.L24      -0.2191      0.210     -1.044      0.296      -0.630       0.192
sigma2       192.1534     39.627      4.849      0.000     114.485     269.822
===================================================================================
Ljung-Box (L1) (Q):                   0.30   Jarque-Bera (JB):                 1.64
Prob(Q):                              0.58   Prob(JB):                         0.44
Heteroskedasticity (H):               1.11   Skew:                             0.33
Prob(H) (two-sided):                  0.77   Kurtosis:                         3.03
===================================================================================
```

Figure 150: Rose | Auto SARIMA Model Summary (3,1,1) x (3,0,2,12)

- **Observations: -**
  - p=3: Three autoregressive (AR) terms
  - d=1: First differencing to achieve stationarity
  - q=1: One moving average (MA) terms
  - P=3: Three seasonal autoregressive (AR) terms
  - D=0: No seasonal differencing
  - Q=2: Two seasonal moving average (MA) terms
  - F=12: Twelve as the seasonal frequency of the time series
  - Non-seasonal MA (1) and seasonal AR (12) & SMA (12) are significant.
  - Higher-order seasonal terms (24, 36) are not contributing much
  - The model fits the data well.
  - Seasonal terms dominate; many AR terms are not significant
  - No autocorrelation in residuals (Ljung-Box p > 0.05).

▪ Below is the Diagnostic Plot: -

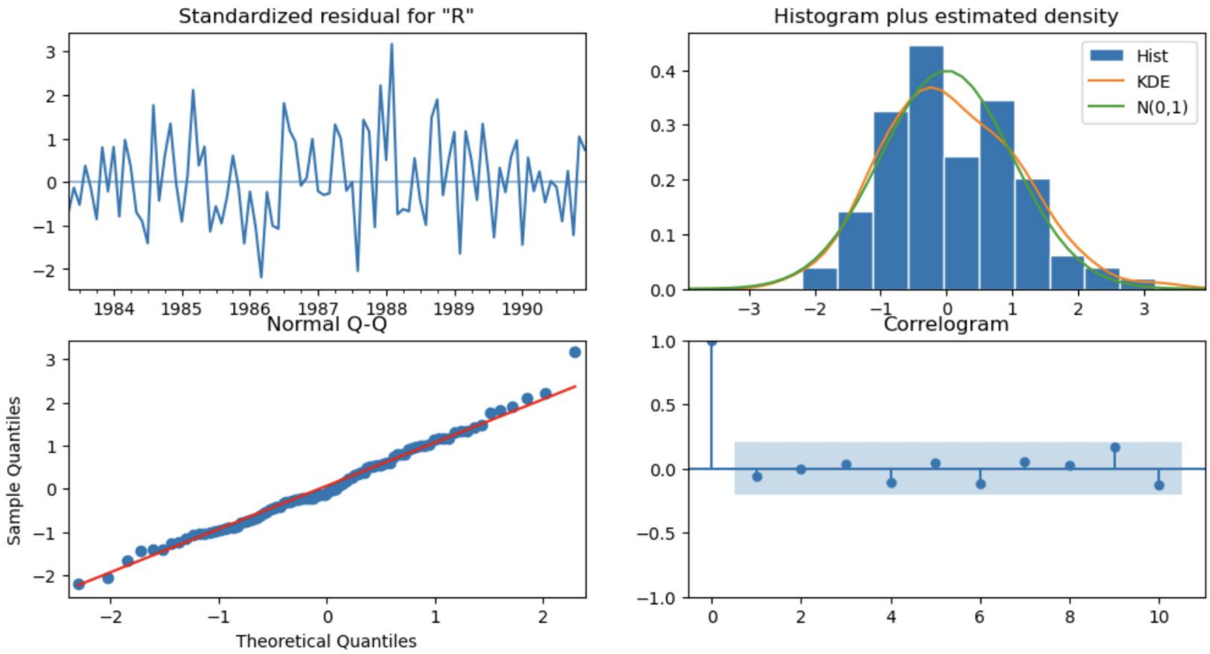


Figure 151: Rose | Auto SARIMA Diagnostic Plot

- **Observations: -**
  - **Standardized Residuals (Time Series Plot):** Residuals fluctuate randomly around zero with no visible trend.
  - **Histogram with KDE and Normal Curve:** Residuals are approximately normally distributed.
  - **Q-Q Plot:** Residuals follow a nearly normal distribution, good enough for forecasting
  - **Correlogram:** Residuals are not auto-correlated, meaning model has captured the temporal structure well.

- Below is the Plot forecasting predicted future values: -

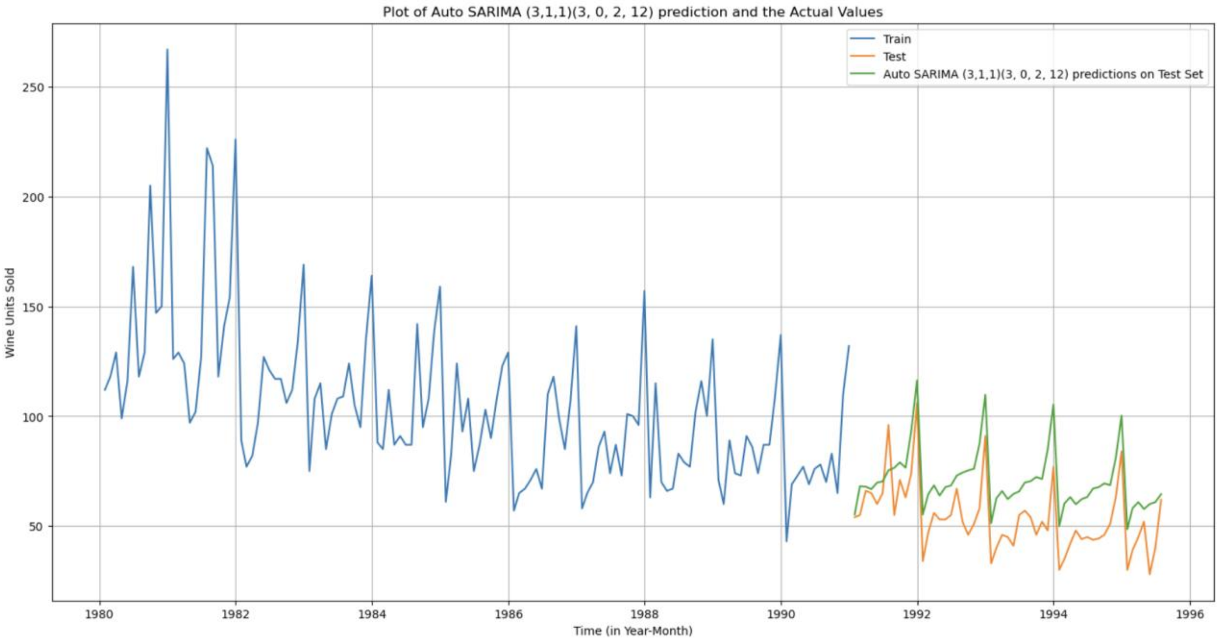


Figure 152: Rose | Auto SARIMA Forecast Plot

- **Model Evaluation –** Below is the RMSE for Auto SARIMA (3,1,1) x (3,0,2,12) Model: -

```
For Auto SARIMA (3,1,1)(3, 0, 2, 12) forecast on the Test Data,  RMSE is 18.947
```

Figure 153: Rose | Auto SARIMA RMSE

- **Observations: -**
  - Clearly, the SARIMA model performs well on seasonal time series. It is due to this reason it is able **to capture the entire characteristics** of the test data.
  - Not surprisingly, the **RMSE** of the aforementioned ARIMA model has **nearly halved relative to ARIMA models**, since it decently captures the seasonality as well.
  - The root means squared error (**RMSE**) of test data for the **Auto SARIMA model (3,1,1) x (3,0,2,12)** is **18.947.**

Auto SARIMA – Sparkling Wine

- SARIMA models are an extension of ARIMA for a time series data with defined seasonality. SARIMA models use seasonal differencing which is similar to regular differencing. An ARIMA model is characterized by 7 terms: p, d, q, P, D, Q, F: -
  - p is the order of the Auto Regressive (AR) term
  - q is the order of the Moving Average (MA) term
  - d is the number of differencing required to make the time series stationary
  - P is the order of the Seasonal Auto Regressive (AR) term
  - Q is the order of the Seasonal Moving Average (MA) term
  - D is the number of seasonal differencing required to make the time series stationary
  - F is the seasonal frequency of the time series
- For the **selection criteria of p, d, q, P, D, Q, F** the ARIMA model is built by using **automated model parameters with lowest Akaike Information Criteria** by using the SARIMAX () function in python**.**
- Since, we know 1st order differencing makes the series stationary, we consider **d =1**.
- We shall use **D=0** as we have shown already that the series is stationary at d=1.
- By looking at the ACF/PACF plots, it is clear that the pattern (spike) is repeating every 12 months. So, **F = 12.**
- We will try different combinations of **p, P & q, Q in the range of 0 to 4.**
- Below is the output of top combinations whose model gave the least AIC: -

	param	seasonal	AIC
0	(3, 1, 2)	(3, 0, 0, 12)	1403.770703
0	(2, 1, 0)	(3, 0, 0, 12)	1411.944973
0	(2, 1, 0)	(3, 0, 1, 12)	1413.101762
0	(1, 1, 2)	(3, 0, 0, 12)	1413.810247
0	(2, 1, 0)	(3, 0, 2, 12)	1414.199455

Figure 154: Sparkling | Auto SARIMA param-AIC combinations

- As evident above, (3,1,2) x (3,0,0,12) | (p,d,q) x (P,D,Q,F), gives the most optimum model. Building the model with this combination: -

```
                                      SARIMAX Results
==========================================================================================
Dep. Variable:                   SparklingWine_Sales   No. Observations:                132
Model:             SARIMAX(3, 1, 2)x(3, 0, [], 12)   Log Likelihood                -684.617
Date:                               Tue, 08 Apr 2025   AIC                           1387.235
Time:                                       20:16:10   BIC                           1409.931
Sample:                                   01-31-1980   HQIC                          1396.395
                                        - 12-31-1990
Covariance Type:                                 opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.5375      0.338     -1.588      0.112      -1.201       0.126
ar.L2          0.0255      0.187      0.137      0.891      -0.340       0.391
ar.L3          0.0784      0.130      0.604      0.546      -0.176       0.333
ma.L1         -0.1876      0.326     -0.575      0.565      -0.827       0.452
ma.L2         -0.6876      0.271     -2.534      0.011      -1.219      -0.156
ar.S.L12       0.5713      0.103      5.541      0.000       0.369       0.773
ar.S.L24       0.2606      0.117      2.223      0.026       0.031       0.490
ar.S.L36       0.2125      0.111      1.915      0.056      -0.005       0.430
sigma2      1.682e+05   2.52e+04      6.673      0.000    1.19e+05    2.18e+05
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                 8.81
Prob(Q):                              0.99   Prob(JB):                         0.01
Heteroskedasticity (H):               1.17   Skew:                             0.36
Prob(H) (two-sided):                  0.67   Kurtosis:                         4.33
===================================================================================
```

Figure 155: Rose | Auto SARIMA Model Summary (3,1,2) x (3,0,0,12)

- **Observations: -**
  - p=3: Three autoregressive (AR) terms
  - d=1: First differencing to achieve stationarity
  - q=2: Two moving average (MA) terms
  - P=3: Three seasonal autoregressive (AR) terms
  - D=0: No seasonal differencing
  - Q=0: No seasonal moving average (MA) terms
  - F=12: Twelve as the seasonal frequency of the time series
  - 3 of 8 parameters are significant. The model seems to be overparameterized.
  - The model fits the data well.
  - Seasonal terms dominate.

- Below is the Diagnostic Plot: -

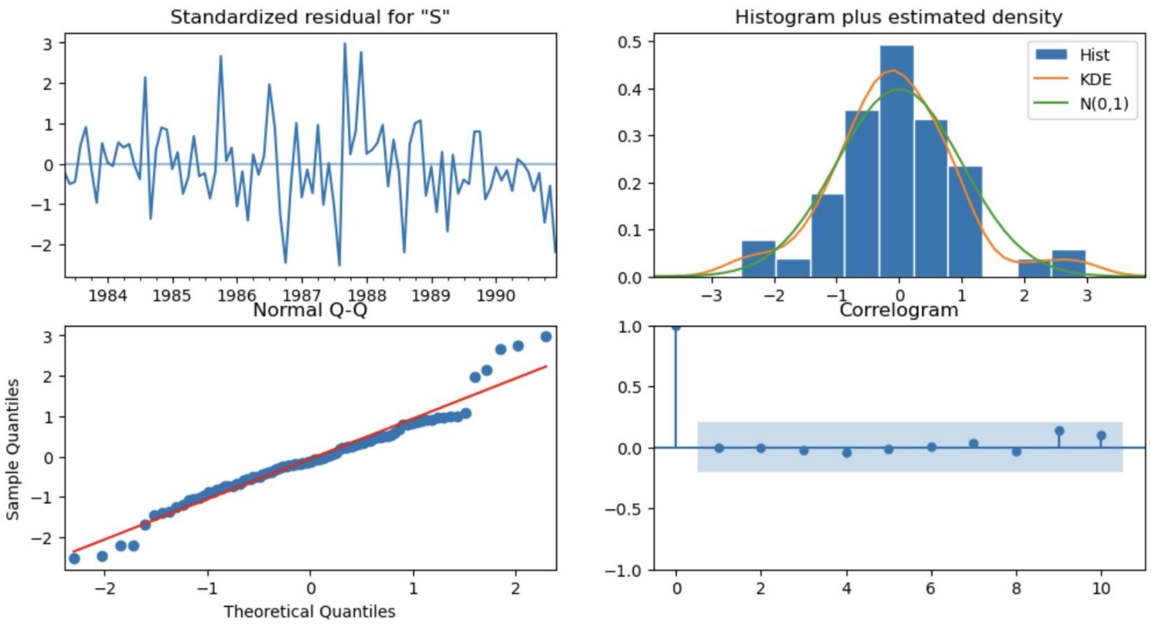


Figure 156: Sparkling | Auto SARIMA Diagnostic Plot

- **Observations: -**
  - **Standardized Residuals (Time Series Plot):** Residuals are cantered around zero.
  - **Histogram with KDE and Normal Curve:** Residuals are approximately normally distributed.
  - **Q-Q Plot:** Good fit to normal distribution, with slight kurtosis
  - **Correlogram:** Residuals are not auto-correlated, meaning model has captured the temporal structure well.

- Below is the Plot forecasting predicted future values: -

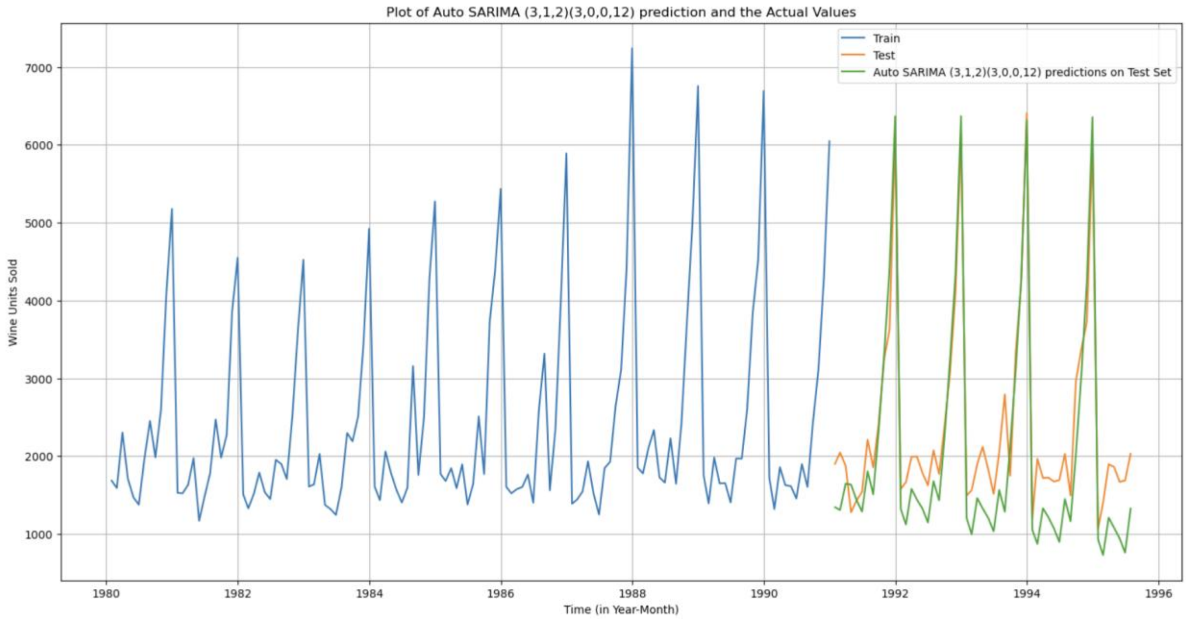


Figure 157: Sparkling | Auto SARIMA Forecast Plot

- **Model Evaluation –** Below is the RMSE for Auto SARIMA (3,1,2) x (3,0,0,12) Model: -

```
For Auto SARIMA (3,1,2)(3,0,0,12) forecast on the Test Data,  RMSE is 543.242
```

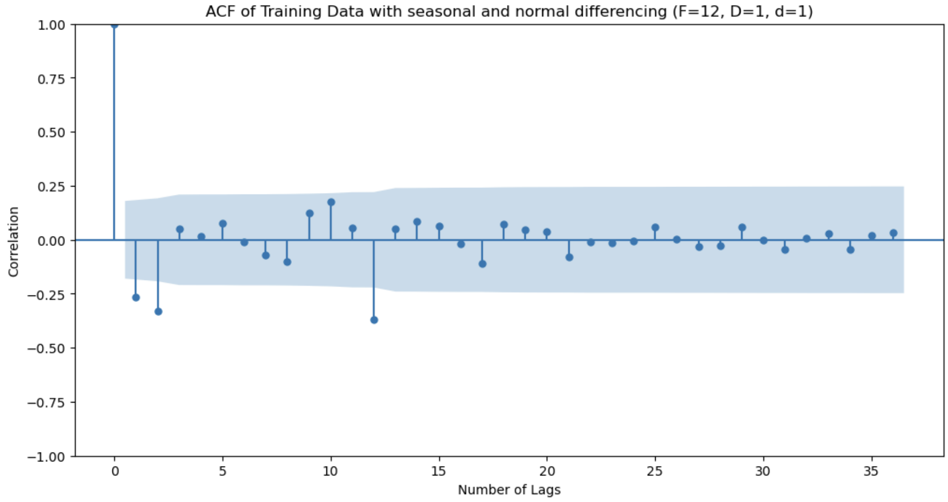
Figure 158: Sparkling | Auto SARIMA RMSE

- **Observations: -**
  - Clearly, the SARIMA model performs well on seasonal time series. It is due to this reason it is able **to capture the entire characteristics** of the test data.
  - Not surprisingly, the **RMSE** of the aforementioned ARIMA model has **nearly halved relative to ARIMA models**, since it decently captures the seasonality as well.
  - The root means squared error (**RMSE**) of test data for the **Auto SARIMA model (3,1,2) x (3,0,0,12)** is **543.242.**

Manual SARIMA – Rose Wine

- SARIMA models are an extension of ARIMA for a time series data with defined seasonality. SARIMA models use seasonal differencing which is similar to regular differencing. An ARIMA model is characterized by 7 terms: p, d, q, P, D, Q, F: -
  - p is the order of the Auto Regressive (AR) term
  - q is the order of the Moving Average (MA) term
  - d is the number of differencing required to make the time series stationary
  - P is the order of the Seasonal Auto Regressive (AR) term
  - Q is the order of the Seasonal Moving Average (MA) term
  - D is the number of seasonal differencing required to make the time series stationary
  - F is the seasonal frequency of the time series
- Since, we know 1st order differencing makes the series stationary, we consider **d =1**. Considering seasonality, we will take **D=1.**
- By examining the ACF/PACF plots earlier, we have already established there seems to be **seasonality of 12**, so we take **F=12.**
- Remaining **parameters to be determined by looking at the PACF & ACF** plots respectively.
- **Autocorrelation function (ACF)** - At lag *k*, this is the correlation between series values that are *k* intervals apart.
- **Partial autocorrelation function (PACF) -** At lag *k*, this is the correlation between series values that are *k* intervals apart, accounting for the values of the intervals between.
- The Auto-Regressive parameter in an ARIMA model is p which comes from the significant lag after which the PACF plot cuts-off below the confidence interval. The Moving-Average parameter in an ARIMA model is 'q' which comes from the significant lag after which the ACF plot cuts-off below the confidence interval.

- Below are the ACF/PACF plots (on train data with d=1, D=1, F=12): -



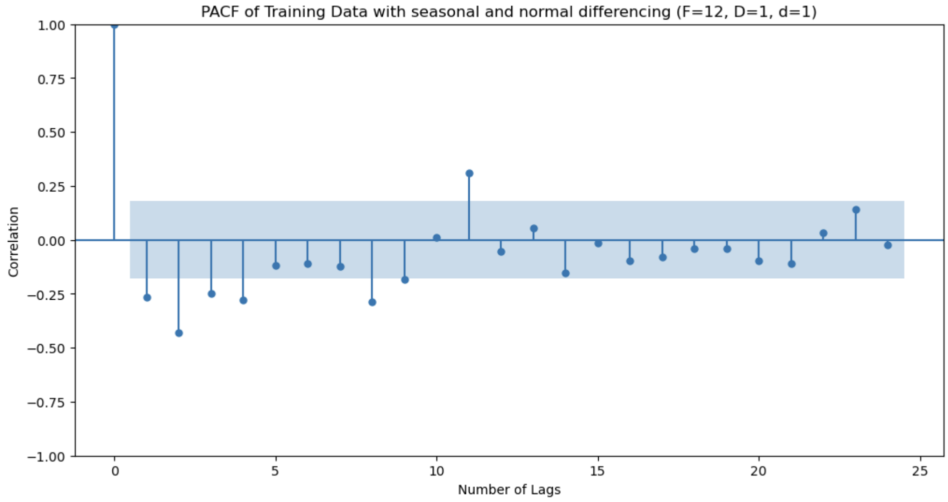


Figure 159: Rose | ACF/PACF Plots on Train data (d=1, D=1, F=12)

- By looking at ACF/PACF plots, we observe: -
  - PACF: Till lag 4 is significant before cut-off, so AR (i.e. p) = 4.
  - PACF: At seasonal lag of 12, it cuts off, so Seasonal AR (i.e. P) = 0.
  - ACF: Till lag 1 and 2 are significant before it cuts off, so MA (i.e. q) = 2.
  - ACF: At seasonal lag of 12, a significant lag is can be seen and no significant seasonal lags after 24, so Seasonal AR (i.e. Q) = 1.
  - By manual inspection, suggested parameters for SARIMA model are **(4, 1, 2) x (0, 1, 1, 12)**, as inferred from the ACF and PACF plots.

- We shall go ahead & now **build for (4, 1, 2) x (0, 1, 1, 12)**: -

```
                                    SARIMAX Results
==========================================================================================
Dep. Variable:                         RoseWine_Sales   No. Observations:                  132
Model:             SARIMAX(4, 1, 2)x(0, 1, [1], 12)   Log Likelihood                -446.102
Date:                                Tue, 08 Apr 2025   AIC                            908.203
Time:                                        19:43:36   BIC                            929.358
Sample:                                    01-31-1980   HQIC                           916.774
                                         - 12-31-1990
Covariance Type:                                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.8046      0.119     -6.778      0.000      -1.037      -0.572
ar.L2          0.0387      0.140      0.276      0.783      -0.237       0.314
ar.L3         -0.2310      0.147     -1.568      0.117      -0.520       0.058
ar.L4         -0.1875      0.108     -1.741      0.082      -0.398       0.024
ma.L1          0.1434   1542.193    9.3e-05      1.000   -3022.499    3022.786
ma.L2         -0.8566   1321.046     -0.001      0.999   -2590.059    2588.346
ma.S.L12      -0.5406      0.085     -6.385      0.000      -0.707      -0.375
sigma2       296.7691   4.58e+05      0.001      0.999   -8.97e+05    8.97e+05
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):                 0.03
Prob(Q):                              0.94   Prob(JB):                         0.98
Heteroskedasticity (H):               0.55   Skew:                            -0.02
Prob(H) (two-sided):                  0.08   Kurtosis:                         3.07
===================================================================================
```

Figure 160: Rose | Manual SARIMA Model Summary (4, 1, 2) x (0, 1, 1, 12)

- **Observations: -**
  - p=4: Four autoregressive (AR) terms
  - d=1: First differencing to achieve stationarity
  - q=2: Two moving average (MA) terms
  - P=0: No seasonal autoregressive (AR) terms
  - D=1: One seasonal differencing
  - Q=1: One seasonal moving average (MA) terms
  - F=12: Twelve as the seasonal frequency of the time series
  - Only 1 out 3 of 7 parameters are not significant.
  - The model fits the data well.
- Below is the Diagnostic Plot: -

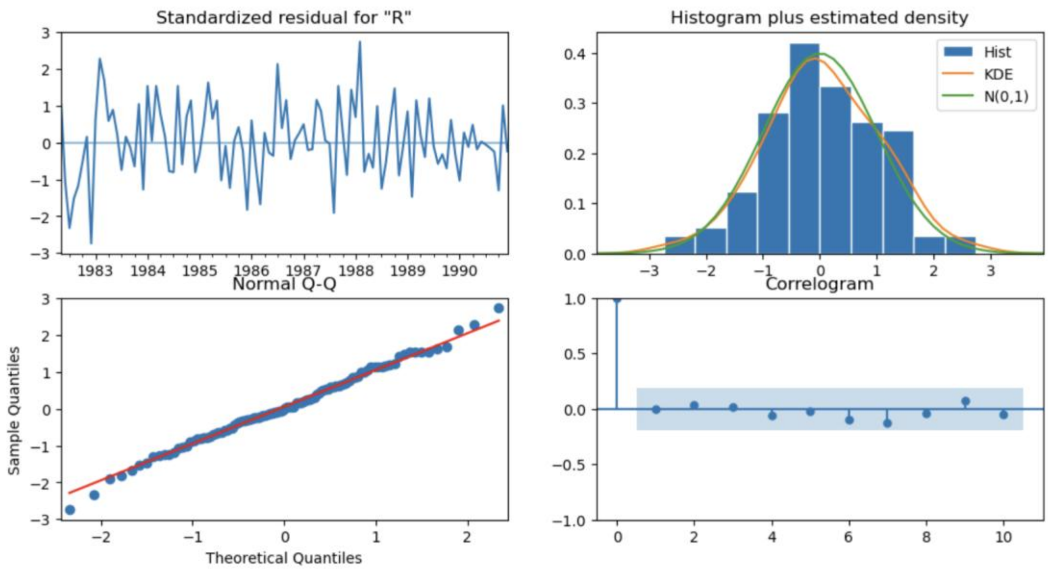


Figure 161: Rose | Manual SARIMA Diagnostic Plot

- **Observations: -**
  - **Standardized Residuals (Time Series Plot):** Residuals fluctuate randomly around zero.
  - **Histogram with KDE and Normal Curve:** Residuals are approximately normally distributed.
  - **Q-Q Plot:** Residuals match a normal distribution.
  - **Correlogram:** Residuals are not autocorrelated, meaning the model has not captured the time-dependence structure well.

- Below is the Plot forecasting predicted future values: -

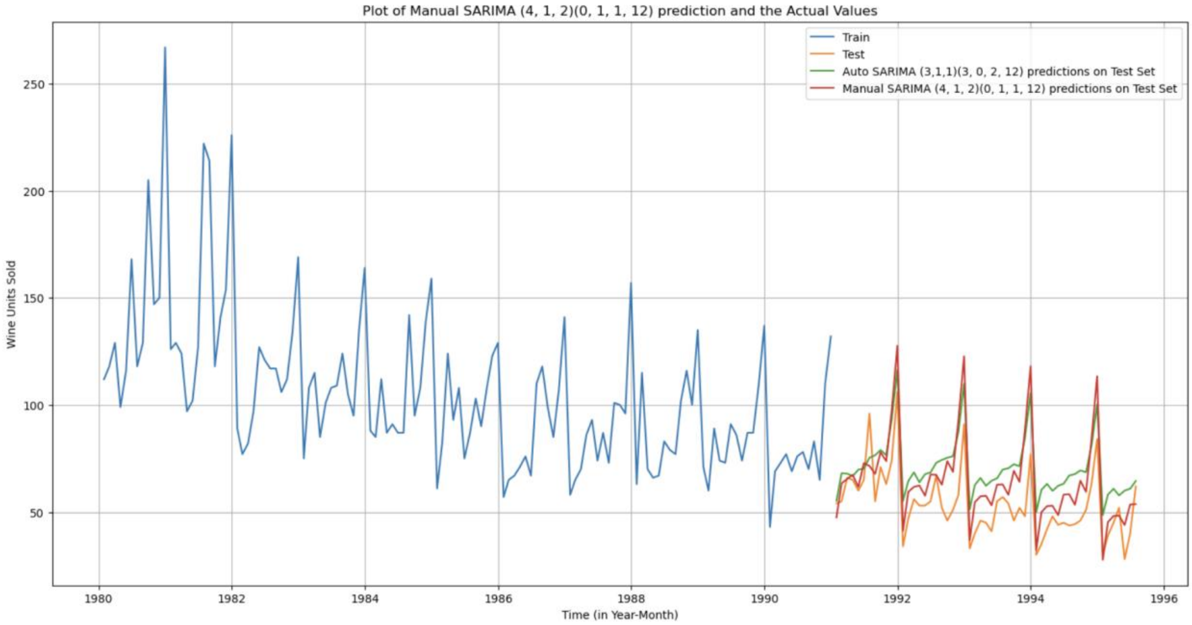


Figure 162: Rose | Manual SARIMA Forecast Plot

- **Model Evaluation –** Below is the RMSE for Auto SARIMA (4, 1, 2) x (0, 1, 1, 12) Model: -

```
For Manual SARIMA (4, 1, 2)(0, 1, 1, 12) forecast on the Test Data,  RMSE is 15.946
```

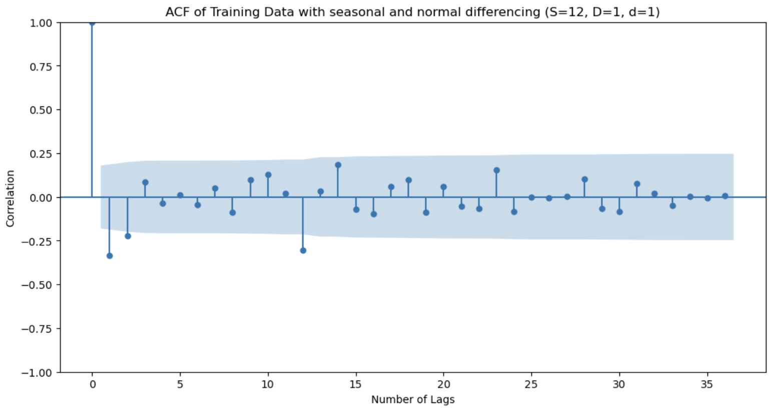
Figure 163: Rose | Manual SARIMA RMSE

- **Observations: -**
  - Clearly, the SARIMA model performs well on seasonal time series. It is due to this reason it is able **to capture the entire characteristics** of the test data.
  - Not surprisingly, the **RMSE** of the aforementioned ARIMA model has **nearly halved relative to ARIMA models**, since it decently captures the seasonality as well.
  - The root means squared error (**RMSE**) of test data for the **Manual SARIMA model (4, 1, 2) x (0, 1, 1, 12)** is **15.946.**

Manual SARIMA – Sparkling Wine

- SARIMA models are an extension of ARIMA for a time series data with defined seasonality. SARIMA models use seasonal differencing which is similar to regular differencing. An ARIMA model is characterized by 7 terms: p, d, q, P, D, Q, F: -
  - p is the order of the Auto Regressive (AR) term
  - q is the order of the Moving Average (MA) term
  - d is the number of differencing required to make the time series stationary
  - P is the order of the Seasonal Auto Regressive (AR) term
  - Q is the order of the Seasonal Moving Average (MA) term
  - D is the number of seasonal differencing required to make the time series stationary
  - F is the seasonal frequency of the time series
- Since, we know 1st order differencing makes the series stationary, we consider **d =1**. Considering seasonality, we will take **D=1.**
- By examining the ACF/PACF plots earlier, we have already established there seems to be **seasonality of 12**, so we take **F=12.**
- Remaining **parameters to be determined by looking at the PACF & ACF** plots respectively.
- **Autocorrelation function (ACF)** - At lag *k*, this is the correlation between series values that are *k* intervals apart.

- **Partial autocorrelation function (PACF) -** At lag *k*, this is the correlation between series values that are *k* intervals apart, accounting for the values of the intervals between.
- The Auto-Regressive parameter in an ARIMA model is p which comes from the significant lag after which the PACF plot cuts-off below the confidence interval. The Moving-Average parameter in an ARIMA model is 'q' which comes from the significant lag after which the ACF plot cuts-off below the confidence interval.
- Below are the ACF/PACF plots (on train data with d=1, D=1, F=12): -



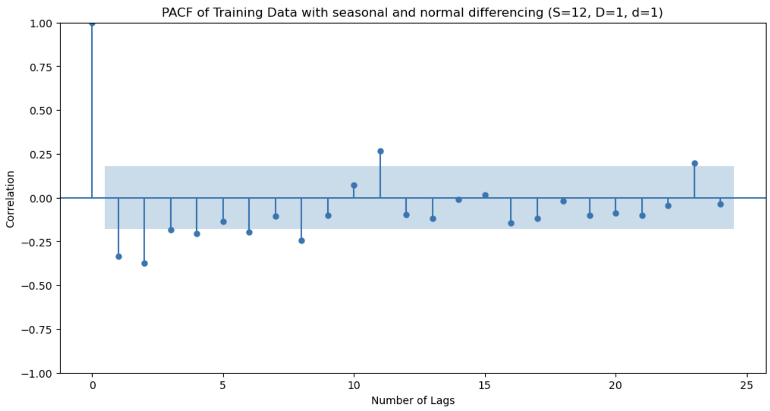


Figure 164: Sparkling | ACF/PACF Plots on Train data (d=1, D=1, F=12)

- By looking at ACF/PACF plots, we observe: -
  - PACF: Till lag 4 is significant before cut-off, so AR (i.e. p) = 4.
  - PACF: At seasonal lag of 12, it cuts off, so Seasonal AR (i.e. P) = 0.
  - ACF: Till lag 1 and 2 are significant before it cuts off, so MA (i.e. q) = 2.
  - ACF: At seasonal lag of 12, a significant lag is can be seen and no significant seasonal lags after 24, so Seasonal AR (i.e. Q) = 1.
  - By manual inspection, suggested parameters for SARIMA model are **(4, 1, 2) x (0, 1, 1, 12)**, as inferred from the ACF and PACF plots.

- We shall go ahead & now **build for (4, 1, 2) x (0, 1, 1, 12)**: -

```
                                      SARIMAX Results
==============================================================================================
Dep. Variable:                     SparklingWine_Sales   No. Observations:                  132
Model:             SARIMAX(4, 1, 2)x(0, 1, [1], 12)   Log Likelihood                -771.377
Date:                             Tue, 08 Apr 2025   AIC                           1558.755
Time:                                     20:46:02   BIC                           1579.910
Sample:                                 01-31-1980   HQIC                          1567.325
                                      - 12-31-1990
Covariance Type:                               opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.1794      0.593     -0.303      0.762      -1.341       0.982
ar.L2         -0.0852      0.193     -0.442      0.659      -0.463       0.293
ar.L3          0.0187      0.114      0.163      0.870      -0.205       0.242
ar.L4         -0.1634      0.151     -1.086      0.278      -0.458       0.132
ma.L1         -0.5369      0.625     -0.859      0.391      -1.763       0.689
ma.L2         -0.3071      0.588     -0.522      0.602      -1.460       0.846
ma.S.L12      -0.4833      0.087     -5.582      0.000      -0.653      -0.314
sigma2      1.614e+05   2.28e+04      7.078      0.000    1.17e+05    2.06e+05
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):                14.39
Prob(Q):                              0.94   Prob(JB):                         0.00
Heteroskedasticity (H):               1.21   Skew:                             0.56
Prob(H) (two-sided):                  0.58   Kurtosis:                         4.44
===================================================================================
```

Figure 165: Sparkling | Manual SARIMA Model Summary (4, 1, 2) x (0, 1, 1, 12)

- **Observations: -**
  - p=4: Four autoregressive (AR) terms
  - d=1: First differencing to achieve stationarity
  - q=2: Two moving average (MA) terms
  - P=0: No seasonal autoregressive (AR) terms
  - D=1: One seasonal differencing
  - Q=1: One seasonal moving average (MA) terms
  - F=12: Twelve as the seasonal frequency of the time series
  - Only the seasonal MA term is significant. Others may be overfitting.
  - The model may be overfitting.

- Below is the Diagnostic Plot: -

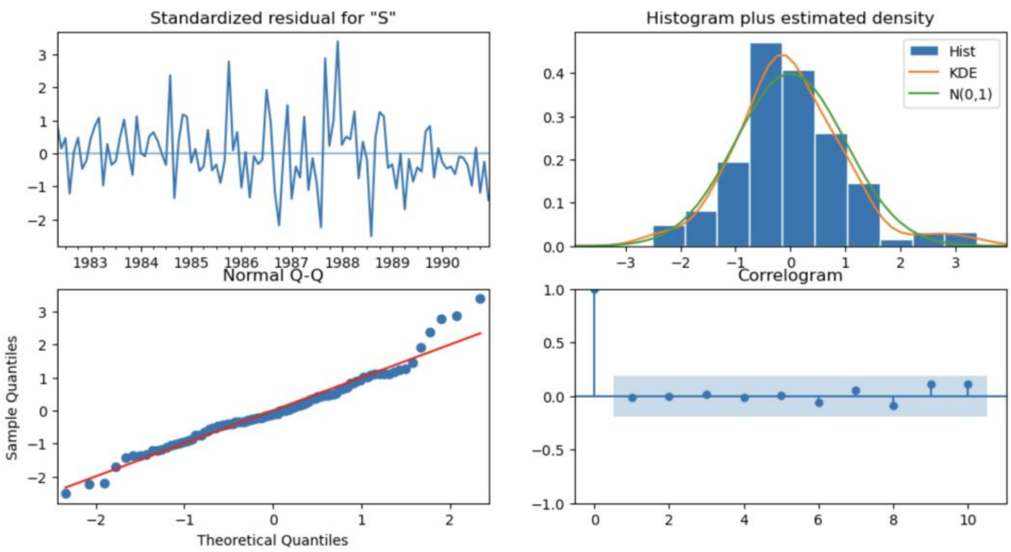


Figure 166: Sparkling | Manual SARIMA Diagnostic Plot

- **Observations: -**
  - **Standardized Residuals (Time Series Plot):** Residuals fluctuate around zero, without visible trends or non-constant variance.
  - **Histogram with KDE and Normal Curve:** Residuals are approximately normally distributed.
  - **Q-Q Plot:** Residuals closely follow a normal distribution, with small tail deviations.
  - **Correlogram:** All autocorrelation lags lie within the 95% confidence bounds, meaning the model has not captured the time-dependence structure well.

▪ Below is the Plot forecasting predicted future values: -

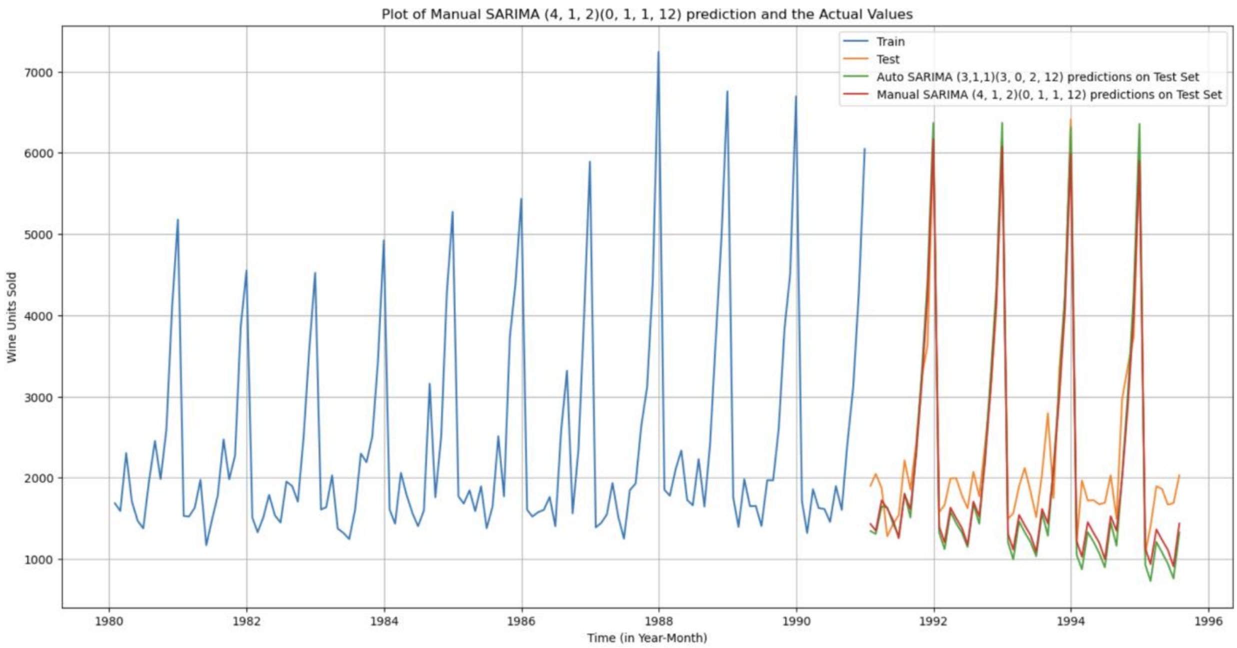


Figure 167: Sparkling | Manual SARIMA Forecast Plot

▪ **Model Evaluation –** Below is the RMSE for Manual SARIMA (4, 1, 2) x (0, 1, 1, 12) Model: -

```
For Manual SARIMA (4,1,2)(0,1,1,12) forecast on the Test Data,  RMSE is 468.682
```

Figure 168: Sparkling | Manual SARIMA RMSE

▪ **Observations: -**
  - Clearly, the SARIMA model performs well on seasonal time series. It is due to this reason it is able **to capture the entire characteristics** of the test data.
  - Not surprisingly, the **RMSE** of the aforementioned ARIMA model shows **significant reduction relative to ARIMA models** (almost one-third), since it captures the seasonality as well.
  - The root means squared error (**RMSE**) of test data for the **Auto SARIMA model (4, 1, 2) x (0, 1, 1, 12)** is **468.682,** which is second to Triple Exponential Smoothing Models only.

Rubric Question 6: Compare the Performance of Models

RMSE Comparison Table of all the models – Rose/Sparkling Wine

- Below is the table showing the Test **RMSE** of all the models built so far for **Rose Wine** (in ascending order): -

	RMSE
Triple Exponential Smoothing (manual) | Alpha = 0.10 | Beta = 0.20 | Gamma = 0.10	9.263811
Moving Average - 2 point Trailing	11.530180
Triple Exponential Smoothing (autofit) | Alpha = 0.0895 | Beta = 0.0002 | Gamma = 0.0035	14.292378
Moving Average - 4 point Trailing	14.462330
Moving Average - 6 point Trailing	14.586916
Moving Average - 9 point Trailing	14.740112
Double Exponential Smoothing (autofit) | Alpha = 1.49e-08 | Beta = 1.66e-10	15.291448
Linear Regression	15.457444
Double Exponential Smoothing (manual) | Alpha = 0.05 | Beta = 0.35	15.670743
Manual SARIMA (4, 1, 2)(0, 1, 1, 12)	15.945927
Auto SARIMA (3,1,1)(3, 0, 2, 12)	18.946850
Auto ARIMA (2,1,3)	36.874717
Simple Exponential Smoothing (manual) | Alpha = 0.10	36.890375
Manual ARIMA (2,1,2)	36.933459
Simple Exponential Smoothing (autofit) | Alpha = 0.1236	37.654503
Simple Average	53.521557
Naive Model	79.778066

Figure 169: Rose | RMSE Model Comparison Table

- **Model Selection: -**

– From the table it is very clear that the **Triple Exponential Smoothing (manual) with Alpha (level) = 0.10, Beta (trend) = 0.20, Gamma (seasonality) = 0.10** is the best model on account of the following reasons: -

  - ✓ **Lowest RMSE of 9.263** – Root Mean Square Error is the most direct indicator of prediction accuracy & since this model has the lowest RMSE of all, significantly better than the rest.
  - ✓ **Captures Level, Trend, and Seasonality** – Triple Exponential Smoothing (Holt-Winters) is ideal for seasonal time series like Rose wine sales.
  - ✓ **Manual tuning beats automatic fitting** – The manually tuned version outperforms the auto-fit version of the same model (RMSE = 14.29).

- Below is the table showing the Test **RMSE** of all the models built so far for **Sparkling Wine** (in ascending order): -

	RMSE
Triple Exponential Smoothing (manual) | Alpha = 0.10 | Beta = 0.00 | Gamma = 0.25	302.911684
Triple Exponential Smoothing (autofit) | Alpha = 0.1572 | Beta = 0.00 | Gamma = 0.3988	318.831018
Manual SARIMA (4,1,2)(0,1,1,12)	468.681644
Auto SARIMA (3,1,2)(3,0,0,12)	543.242254
Moving Average - 2 point Trailing	813.400684
Moving Average - 4 point Trailing	1156.589694
Manual ARIMA (4,1,3)	1232.597198
Simple Average	1275.081804
Moving Average - 6 point Trailing	1283.927428
Auto ARIMA (2,1,2)	1299.979756
Simple Exponential Smoothing (autofit) | Alpha = 0.0395	1304.927405
Moving Average - 9 point Trailing	1346.278315
Simple Exponential Smoothing (manual) | Alpha = 0.10	1375.393398
Linear Regression	1386.836243
Double Exponential Smoothing (manual) | Alpha = 0.05 | Beta = 0.05	1418.407668
Double Exponential Smoothing (autofit) | Alpha = 0.6885 | Beta = 9.99e-05	2007.238526
Naive Model	3864.279352

Figure 170: Sparkling | RMSE Model Comparison Table

- **Model Selection: -**

− From the table it is very clear that the **Triple Exponential Smoothing (manual) with Alpha (level) = 0.10, Beta (trend) = 0.00, Gamma (seasonality) = 0.25** is the best model on account of the following reasons: -

  - ✓ **Lowest RMSE of 302.912**– Root Mean Square Error is the most direct indicator of prediction accuracy & since this model has the lowest RMSE of all, significantly better than the rest.
  - ✓ **Captures Seasonality, Ignores Trend** – Beta is 0, meaning no trend modelling, which is ideal if your data shows strong seasonality but no clear upward/downward trend over time. Gamma of 0.25 effectively captures seasonal fluctuations, which is crucial in Sparkling Wine sales, as they often spike during holidays/festive seasons.
  - ✓ **Manual tuning beats automatic fitting** – The manually tuned version outperforms the auto-fit version of the same model (RMSE = 318.83).

Rebuild the Best Model (whole dataset) & Forecast for next 12 months – Rose Wine

- The most optimum model selected – **Triple Exponential Smoothing (manual), with Alpha (level) = 0.10, Beta (trend) = 0.20, Gamma (seasonality) = 0.10,** as parameters.
- Building the model again on the whole dataset: -

```
{'smoothing_level': 0.1,
 'smoothing_trend': 0.2,
 'smoothing_seasonal': 0.1,
 'damping_trend': nan,
 'initial_level': 137.08025526287202,
 'initial_trend': 1.1822177142197932,
 'initial_seasons': array([0.80846632, 0.87922805, 0.95915429, 0.83360616, 0.92786195,
        1.03980828, 1.16713904, 1.23152331, 1.12739185, 1.11144107,
        1.25453673, 1.77741085]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Figure 171: Rose | TES Optimum Model

- **Model Evaluation** – Below is the **RMSE** of the model on the whole dataset: -

```
For Total Exponential Smoothing forecast on the Full Data (0.1,0.2,0.1),  RMSE is 17.031
```

Figure 172: Rose | RMSE of TES Optimum Model

- Below are the **forecasted values** for **next 12 months**: -

1995-08-31	49.823594
1995-09-30	49.797715
1995-10-31	50.777794
1995-11-30	59.151832
1995-12-31	82.319635
1996-01-31	33.709173
1996-02-29	40.798303
1996-03-31	46.111860
1996-04-30	44.967472
1996-05-31	43.122837
1996-06-30	48.062507
1996-07-31	54.780756

Figure 173: Rose | Forecast for next 12 months (TES Optimum Model)

- Below is the forecast **time series plot for next 12 months**: -

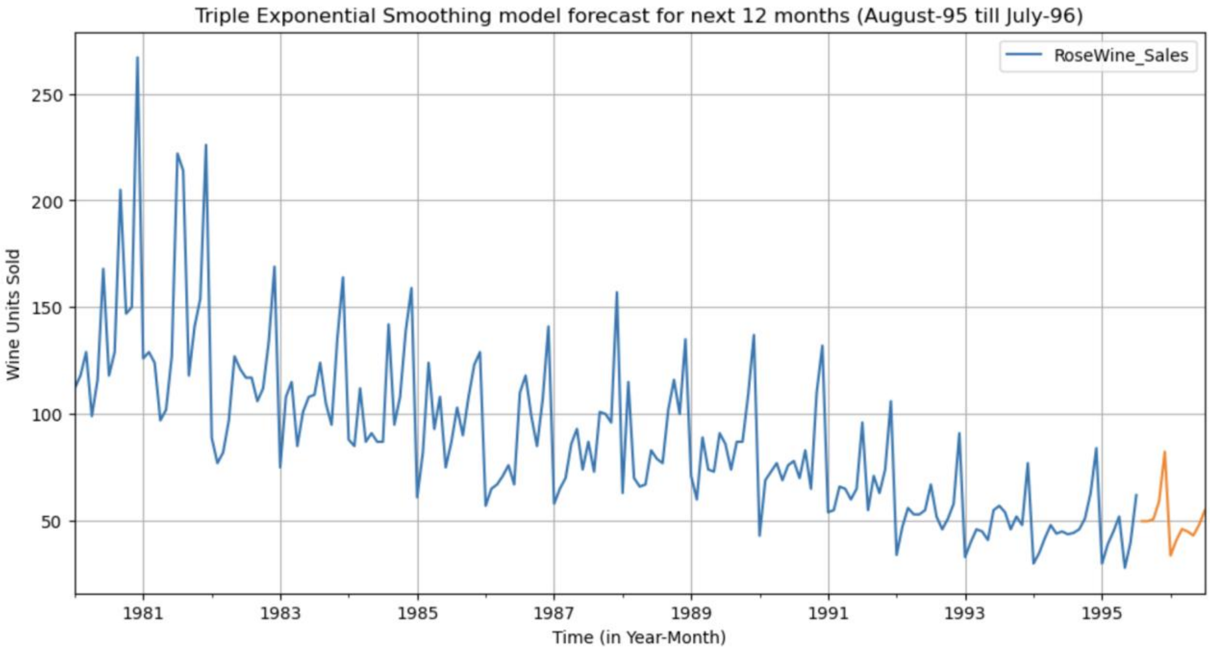


Figure 174: Rose | Time Series Forecast Plot for next 12 months (TES Optimum Model)

- Below are the **forecasted values** for **next 12 months** with **upper and lower confidence bands at 95% confidence level**: -

	lower_ci	prediction	upper_ci
1995-08-31	16.361652	49.823594	83.285537
1995-09-30	16.335773	49.797715	83.259658
1995-10-31	17.315852	50.777794	84.239736
1995-11-30	25.689890	59.151832	92.613774
1995-12-31	48.857692	82.319635	115.781577
1996-01-31	0.247231	33.709173	67.171115
1996-02-29	7.336360	40.798303	74.260245
1996-03-31	12.649917	46.111860	79.573802
1996-04-30	11.505530	44.967472	78.429415
1996-05-31	9.660895	43.122837	76.584779
1996-06-30	14.600565	48.062507	81.524449
1996-07-31	21.318814	54.780756	88.242698

Figure 175: Rose | Forecast for next 12 months with CI Bands (TES Optimum Model)

- Below is the **plot of the full data with forecast** along with the **confidence band (95%)**: -

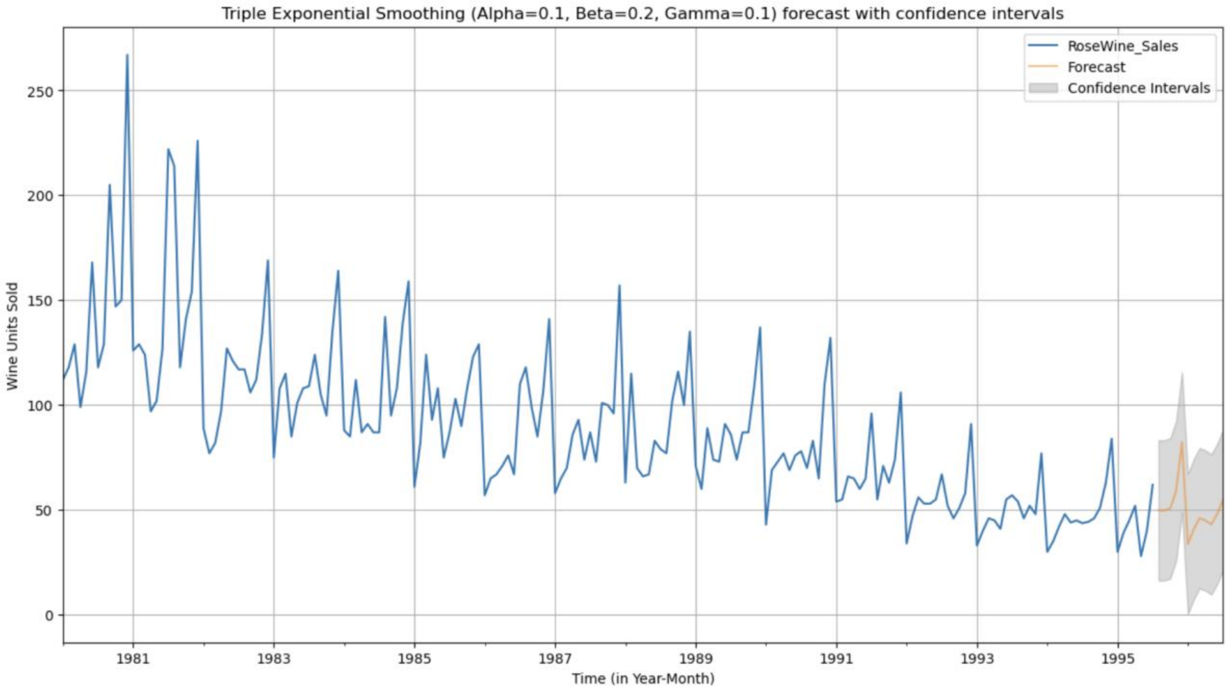


Figure 176: Rose | Time Series Forecast Plot for next 12 months (TES Optimum Model) with CI Bands

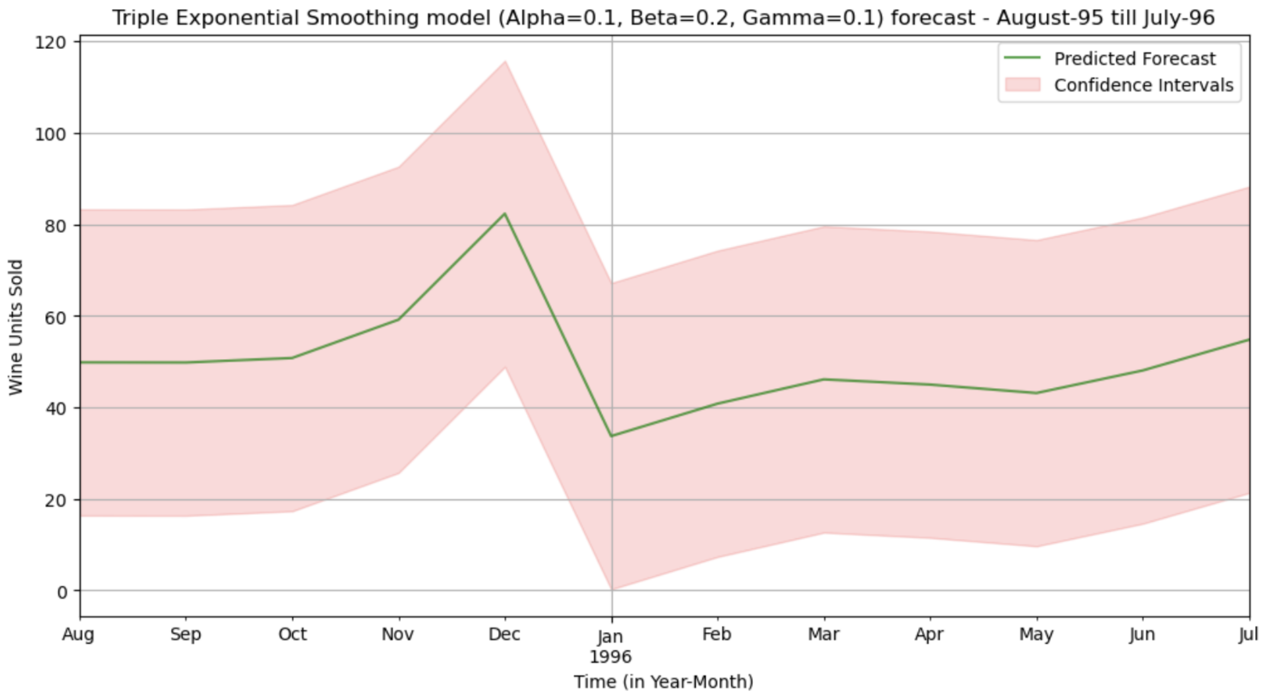


Figure 177: Rose | Time Series Forecast Plot for next 12 months (TES Optimum Model) with CI Bands (Focused-view)

Rebuild the Best Model (whole dataset) & Forecast for next 12 months – Sparkling Wine

- The most optimum model selected – **Triple Exponential Smoothing (manual), with Alpha (level) = 0.10, Beta (trend) = 0.00, Gamma (seasonality) = 0.25,** as parameters.
- Building the model again on the whole dataset: -

```
{'smoothing_level': 0.1,
 'smoothing_trend': 0.0,
 'smoothing_seasonal': 0.25,
 'damping_trend': nan,
 'initial_level': 2356.3655442540025,
 'initial_trend': 0.8359448201869769,
 'initial_seasons': array([0.79086386, 0.7652567 , 0.94969844, 0.87656335, 0.81371731,
        0.74266526, 0.96739807, 1.19908029, 1.06268518, 1.33785076,
        1.99168205, 2.61707748]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Figure 178: Sparkling | TES Optimum Model

- **Model Evaluation** – Below is the **RMSE** of the model on the whole dataset: -

```
For Total Exponential Smoothing (Alpha = 0.10 | Beta = 0.00 | Gamma = 0.25) forecast on the Full Data,  RMSE is 354.356
```

Figure 179: Sparkling | RMSE of TES Optimum Model

- Below are the **forecasted values** for **next 12 months**: -

1995-08-31	1966.341594
1995-09-30	2375.376435
1995-10-31	3216.738557
1995-11-30	3993.855527
1995-12-31	6085.378194
1996-01-31	1402.901270
1996-02-29	1633.345905
1996-03-31	1870.459606
1996-04-30	1826.695007
1996-05-31	1676.516930
1996-06-30	1583.246265
1996-07-31	2014.109714

Figure 180: Sparkling | Forecast for next 12 months (TES Optimum Model)

▪ Below is the forecast **time series plot for next 12 months**: -

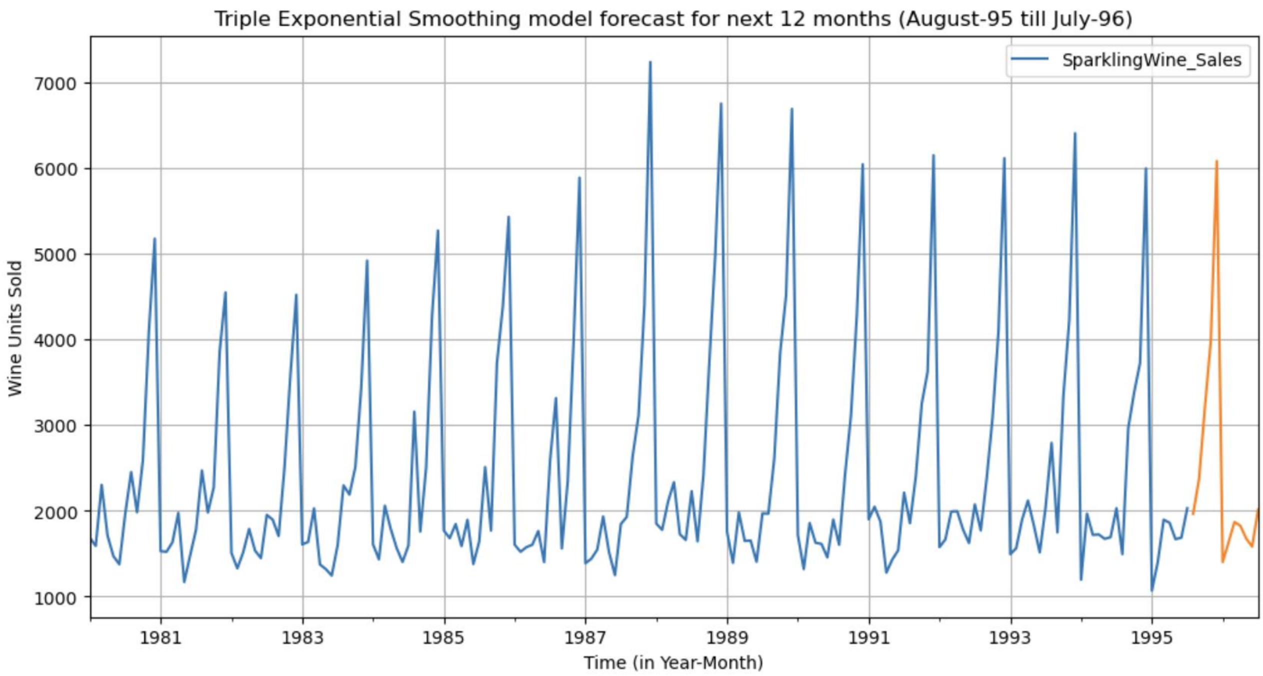


Figure 181: Sparkling | Time Series Forecast Plot for next 12 months (TES Optimum Model)

▪ Below are the **forecasted values** for **next 12 months** with **upper and lower confidence bands at 95% confidence level**: -

	lower_ci	prediction	upper_ci
1995-08-31	1270.699014	1966.341594	2661.984175
1995-09-30	1679.733854	2375.376435	3071.019015
1995-10-31	2521.095976	3216.738557	3912.381137
1995-11-30	3298.212947	3993.855527	4689.498108
1995-12-31	5389.735613	6085.378194	6781.020774
1996-01-31	707.258689	1402.901270	2098.543850
1996-02-29	937.703324	1633.345905	2328.988485
1996-03-31	1174.817026	1870.459606	2566.102187
1996-04-30	1131.052426	1826.695007	2522.337587
1996-05-31	980.874350	1676.516930	2372.159510
1996-06-30	887.603684	1583.246265	2278.888845
1996-07-31	1318.467134	2014.109714	2709.752294

Figure 182: Sparkling | Forecast for next 12 months with CI Bands (TES Optimum Model)

- Below is the **plot of the full data with forecast** along with the **confidence band (95%)**: -

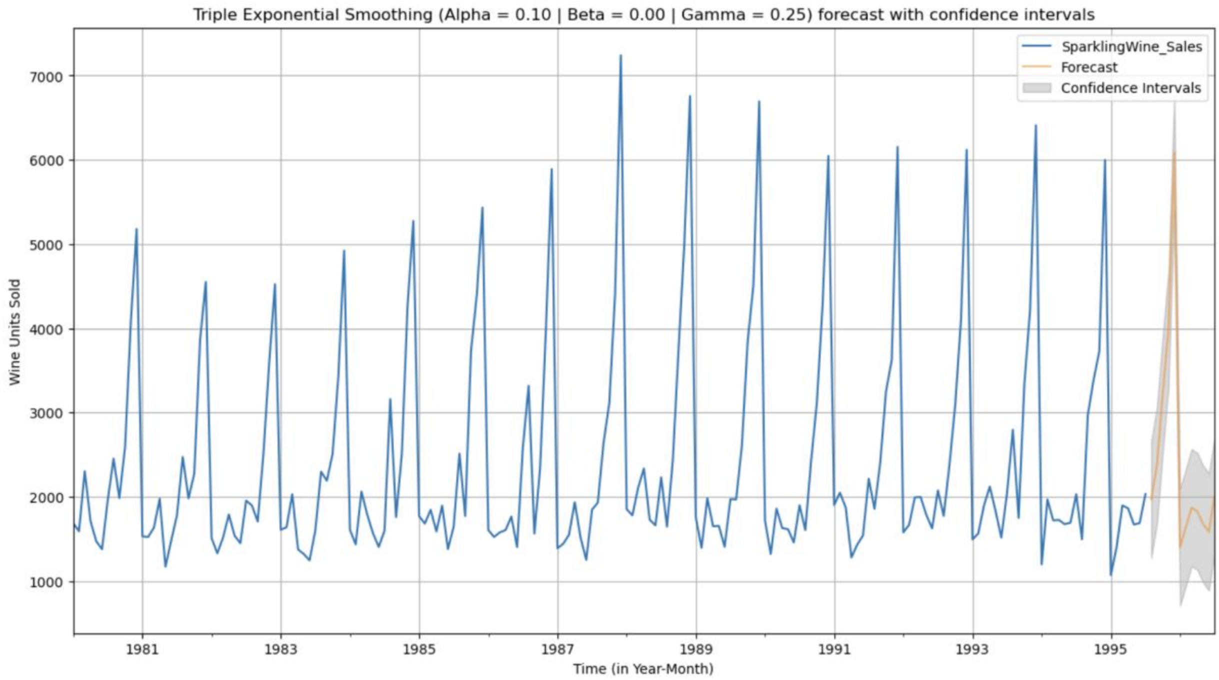


Figure 183: Sparkling | Time Series Forecast Plot for next 12 months (TES Optimum Model) with CI Bands

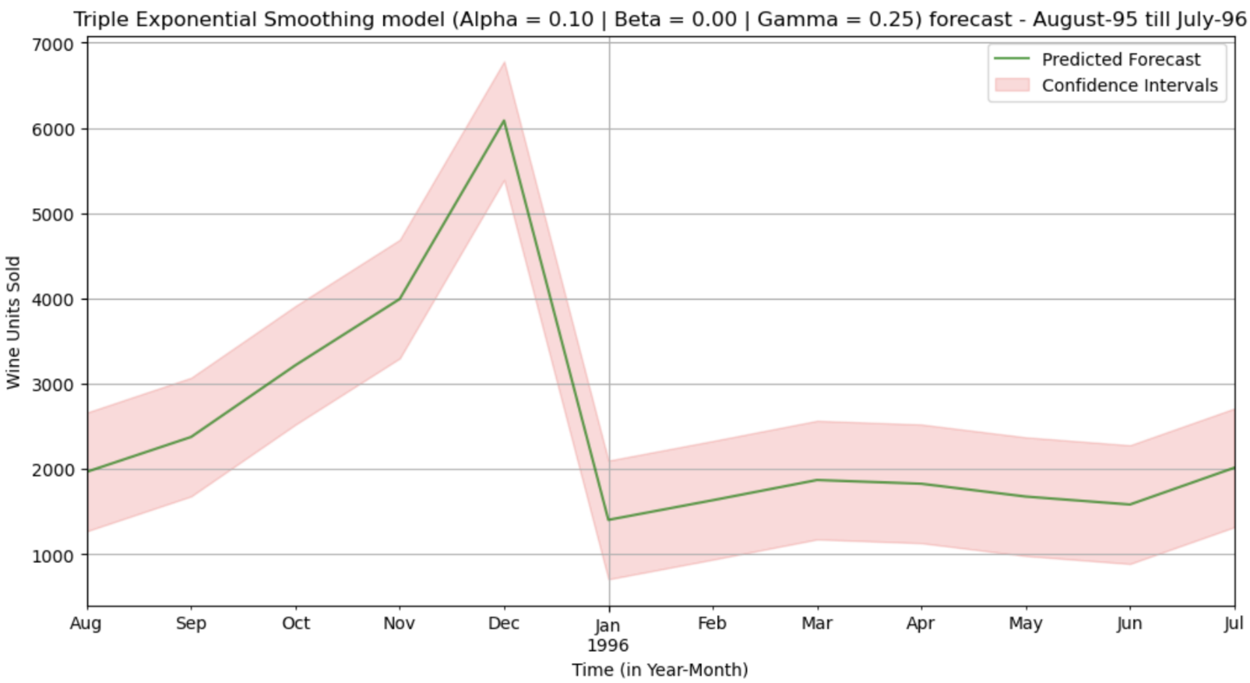


Figure 184: Sparkling | Time Series Forecast Plot for next 12 months (TES Optimum Model) with CI Bands (Focused-view)

Rubric Question 7: Actionable Insights & Recommendations

Rose Wine

- **Model Performance: -**
  - The Triple Exponential Smoothing (manual) model (Alpha=0.1, Beta=0.2, Gamma=0.1) achieved the **lowest RMSE of 9.26**, significantly outperforming other methods, including ARIMA and SARIMA.
  - RMSE for the final model on the whole dataset is also decent at 17.031.
  - This implies that the TES model **captured trend, seasonality, and level** components very well for Rose Wine Sales.
- **Historical Insights: -**
  - Please refer the link → **EDA-Rose** section for details.
- **Forecast Insights: -**
  - Sales are **expected to increase steadily from August 1995 to December 1995**, peaking in December (82.3 units).
  - **A sharp drop** occurs **in January 1996 (33.7 units)**, likely due to post-holiday seasonality.
  - **Gradual recovery** follows from **February to July 1996**, ending at **54.8 units** in July.
  - **Strong seasonality** is evident, especially with high **December** demand, possibly due to **holiday-related consumption.**
  - **Lower sales in early-year months** like January and February are consistent with **seasonal dips**.
  - The **Confidence Interval** (shaded areas) shows increasing uncertainty in the forecast, especially around **December (1995)** with wider interval (48.9 to 115.8) indicating **high variability**, likely due to irregular year-end spikes; and, **Early 1996 with** extremely wide CI in **Jan 1996 (0.24 to 67.17)**, implying **forecast instability** or **volatile historical data** during this time.
  - **Flattening Pattern:** Over the long term, there appears to be a **gradual decline in overall Rose wine sales.** The **last few years appear to have flattened out,** with less volatility but no strong upward movement, implying a sign of **possible market saturation or shifting preferences.** Such sustained **sales decline could lead to lower revenues and excess capacity.**
- **Business Recommendations: -**
  1. **Customer Insight & Market Research:**
     - Conduct surveys or analytics to understand why sales are falling: Is it price, taste, branding, or competition?
     - Explore if customers are switching to other beverages (e.g., craft beer, cocktails, non-alcoholic options).
     - Investigate factors causing wide forecast intervals in some months.
  2. **Review Pricing Strategy:**
     - If demand is price-sensitive, consider value-tier offerings or targeted discounts.
  3. **Product Innovation:**
     - Introduce new wine variants or rebrand the older wine.
     - Consider limited-edition seasonal wines, packaging changes, or smaller serving options.
  4. **Promotions & Marketing:**
     - Capitalize on the holiday spike with bundled offers or gift packs.
     - Develop strategies to reduce the post-holiday drop, e.g., loyalty discounts or “New Year, New Wine” campaigns.
  5. **Diversify Sales Channels:**
     - Invest in online sales, subscription models, or export to emerging markets.
     - Collaborate with restaurants or events for promotions and exposure.
  6. **Inventory Management & Cash Flow Planning:**
     - Stock up for December: Prepare for high demand; consider bulk purchasing or ramping up production.
     - Reduce stock in Jan–Feb: Sales drop significantly; avoid overstocking during this low-demand period.
     - Expect a revenue peak in Q4 (Oct–Dec) and a dip in Q1 (Jan–Mar); align financial strategies accordingly.
     - Save on logistics and warehousing during the low-sales months.

Sparkling Wine

- **Model Performance: -**
  - The Triple Exponential Smoothing (manual) model (Alpha=0.1, Beta=0.0, Gamma=0.25) achieved the **lowest RMSE of 302.91**, significantly outperforming other methods, including ARIMA and SARIMA.
  - RMSE for the final model on the whole dataset is also decent at 354.356.
  - This implies that the TES model **captured seasonality and level** components very well, while **ignoring the trend** component (as evident in the chart of whole dataset) for Sparkling Wine Sales.
- **Historical Insights: -**
  - Please refer the link → **EDA-Sparkling** section for details.
- **Forecast Insights: -**
  - **No Clear Long-Term Decline**. Unlike Rose wine, **overall volumes remain steady**, with peak sales reaching **above 6000–7000 units annually**.
  - The chart shows a **repeating annual spike** (likely in December), with a **relatively stable baseline** throughout the years.
  - **Monthly sales** between peaks hover around **2000 units,** with **minor fluctuations** over the years.
  - **Forecasted values** for Aug–Nov 1995 lies within this **stable range** (1966 to 3993 units).
  - **Strong and Predictable Seasonality,** wherein every year, a sharp **peak occurs in December**, probably due to **holiday celebrations** and New Year's festivities.
  - After the December peak, **sales plunge drastically in January** (forecast: 1402 units), before slowly recovering through the year.
  - High Uncertainty in December with high Confidence intervals (5389-6781). This indicates **volatility in historical holiday season sales** as customers may buy significantly more or slightly less depending on external factors (weather, economy, etc.).
  - **Post-peak months (Jan–May)** show **less forecast variability**, suggesting more predictable off-peak sales.
- **Business Recommendations: -**
  1. **Prepare for December Spike & Post-Holiday Strategy:**
     - Ensure high inventory and staffing for the November–December surge.
     - Invest in promotions, gift packaging, and festive marketing to maximize the holiday rush.
     - Investigate factors causing wide forecast intervals in some months.
     - Offer post-New Year discounts or smaller-size bottles to sustain engagement.
     - Target events like Valentine’s Day or Mother’s Day with themed promotions to stimulate sales in quieter months.
  2. **Stabilize Non-Seasonal Sales:**
     - Sparkling wine is often seen as a celebration drink. Try repositioning it for regular occasions; e.g., brunch, everyday luxury.
     - Develop new use cases (cocktail mixers, sparkling mocktails) to appeal to younger consumers.
  3. **Explore Export & Online Channels:**
     - The predictability of seasonal demand makes it ideal for planning subscription boxes or online pre-orders. Build a DTC platform offering seasonal boxes, special occasion kits, and birthday or holiday gifting subscriptions.
     - International markets with different seasonal patterns may help balance demand across the year.
  4. **Diversify Usage Occasions Beyond Holidays:**
     - Partner with lifestyle influencers to promote “Sparkling Moments” year-round.
     - Launch marketing campaigns cantered on non-traditional holidays (e.g., “Treat Yourself Tuesdays,” “Summer Sparkle Nights”).
  5. **Product Line Innovation:**
     - Expand product lines to create new revenue streams to serve the ever-evolving consumer preferences (Flavoured sparkling wines, Low- or no-alcohol options, Premium offerings with unique packaging or limited editions)
  6. **Experiential Marketing & Brand Engagement:**
     - Highlight eco-friendly packaging, local sourcing, or low-carbon footprint production. Position Sparkling wine as not just a luxury but a conscious indulgence.
     - Host or sponsor events, tasting tours, or immersive virtual experiences.
     - Create a brand story that connects emotionally with customers (e.g., family legacy, vineyard tradition, or modern lifestyle).