TIME SERIES FORECASTING

PROJECT

ABC ESTATE WINES

CASE - SPARKLING WINE

SABITA NAIR PANCHAL

BATCH - DSBA (FEBRUARY, 2021)

SUBMISSION DATE - 10/10/2021

TABLE OF CONTENTS

Executive Summary:

ABC Estate Wines have asked for an analysis into the sales pattern of their brand 'Sparkling' wine. A data of monthly Sparkling wine sales from January, 1980 to July 1995 has been shared. Basis this, the company is also expecting a future forecast of the sales of this brand of wine into the next 12 months.

1.1 Reading the Time series:

The data of the 'Sparkling' wine sales is a classic example of a univariate time series with monthly frequency. The first five rows of the dataset is given in the table below:

Table: 1.1

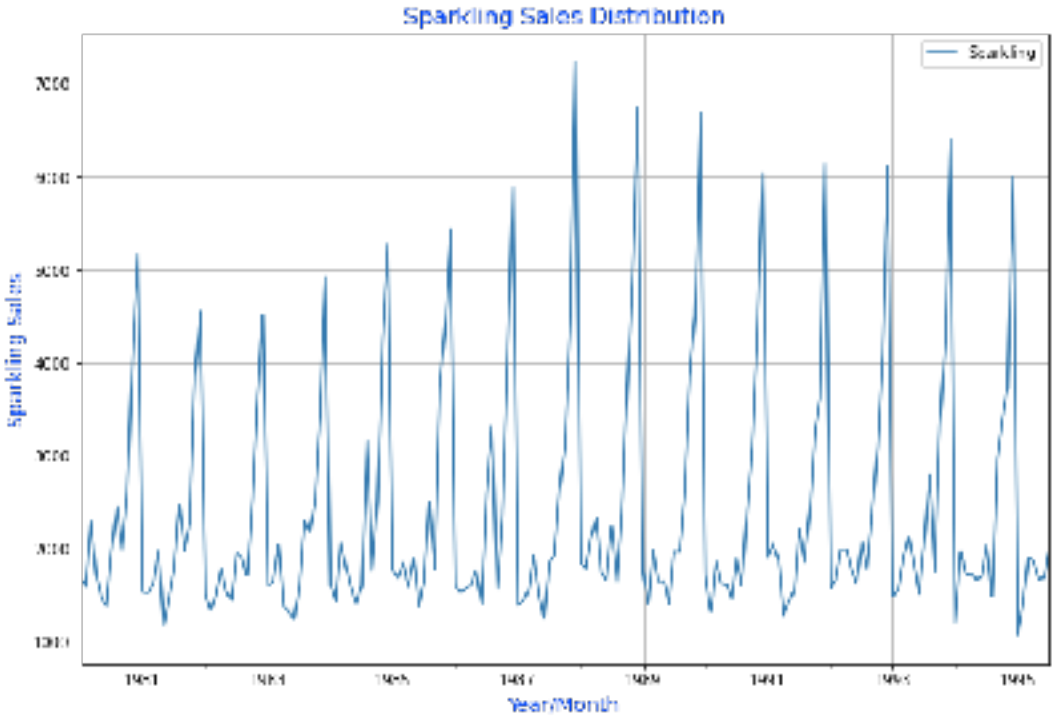
YearMonth	Sparkling
1980-01-01	1686
1980-02-01	1591
1980-03-01	2304
1980-04-01	1712
1980-05-01	1471

There are 187 rows in the dataset, and no null/missing values. The data description is given below:

YearMonth	represents the month and year (Parsing was done to convert this into DateTimeIndex, essential for Time series analysis)
Sparkling	represents the monthly sales of the 'Sparkling' brand of wine

A time series plot was used to graphically represent the distribution of 'Sparkling' wine sales over the years (as shown in figure below).

Figure: 1.1



Inferences:

- The data is contiguous, without any change in time sequence
- There are no apparent missing values.
- Although there is a high and low movement of the sales numbers, there is no evident trend. Sales seem to have shown a steady increase between 1983 and 1988. Post that, there has been a decline till 1991, and thereafter it has been fairly steady except a slight spurt in 1994.
- The seasonality factor is very evident here - every year has seen a sharp spike in sales around the same time - this can only be attributed

to a seasonal component. The seasonal pattern is very similar across the years.

1.2 Exploratory data analysis:

—> Statistical summary of the data:

Statistical summary of the dataset was checked using the describe() function, the results are given below:

Table: 1.2.1

	Sparkling
count	187
mean	2402.417112
std	1295.11154
min	1070
25%	1605
50%	1874
75%	2549
max	7242

Inferences: The following can be inferred from the above statistics:

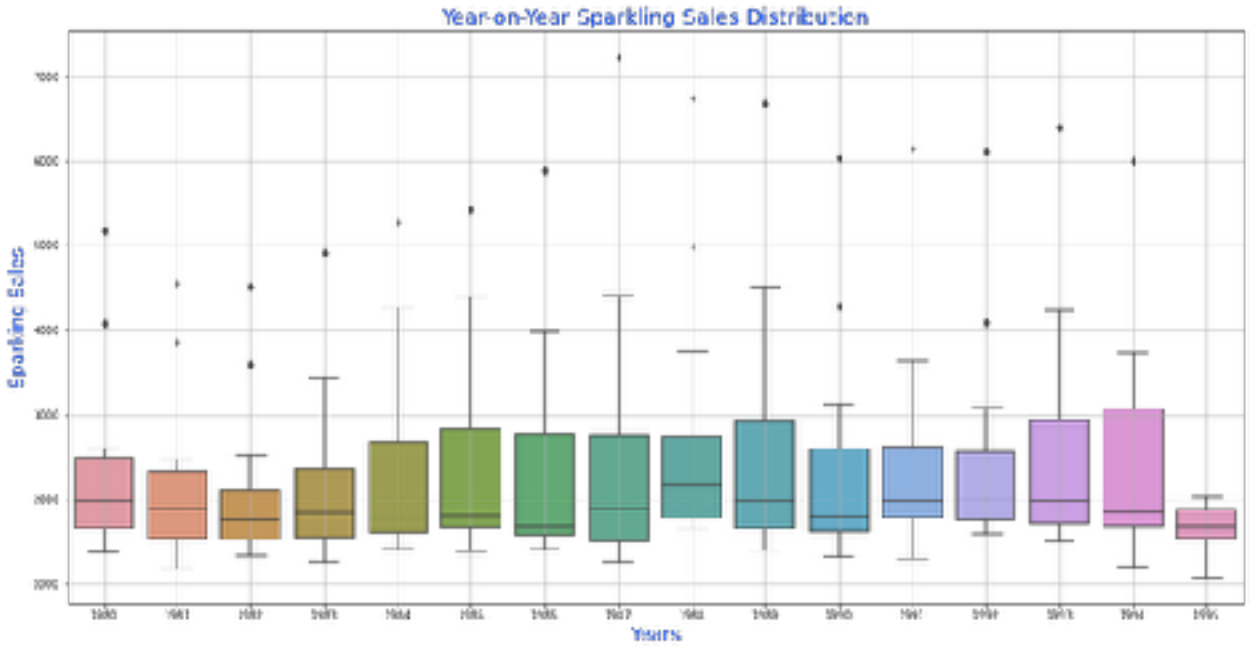
- The average monthly sales of 'Sparkling' wine is approx. 2402.
- The minimum sales registered in a month is 1070, while the maximum sales registered is 7242
- The mean of the observations (2402) is fairly than the median (1874) - proving that the wine sales figures do not follow a normal distribution, I.e., they are not centered around the mean value.

- Also, a standard deviation of 1295 shows that there is high variation in the data.

Univariate analysis:

—> Year-on-year spread of ‘Sparkling’ sales:

Figure: 1.2.1



Inferences: This plot reveals the distribution of sales of ‘Sparkle’ wine per year. We can observe the following:

- The largest sales distribution is evident in the years 1987 and 1989. The least distribution was in years 1980, 1981 and 1982. A spurt in sales was seen after that, however, sales distribution was sparse again in the years 1990 and 1992.

- The year 1995 will not be considered as we only have data for the first seven months of the year.

- The median sales value for many years is centered around the 2000 mark (1980, 1989, 1991, 1992 and 1993), showing that across those years many months in that year have recorded sales centered around that month.

- The least sales recorded for any year was in 1995 (approx. 1100), and the highest sales was in 1987 (over 7200 - albeit an outlier).

—> Pivot table of month-wise and year-wise sales:

Table: 1.2.2

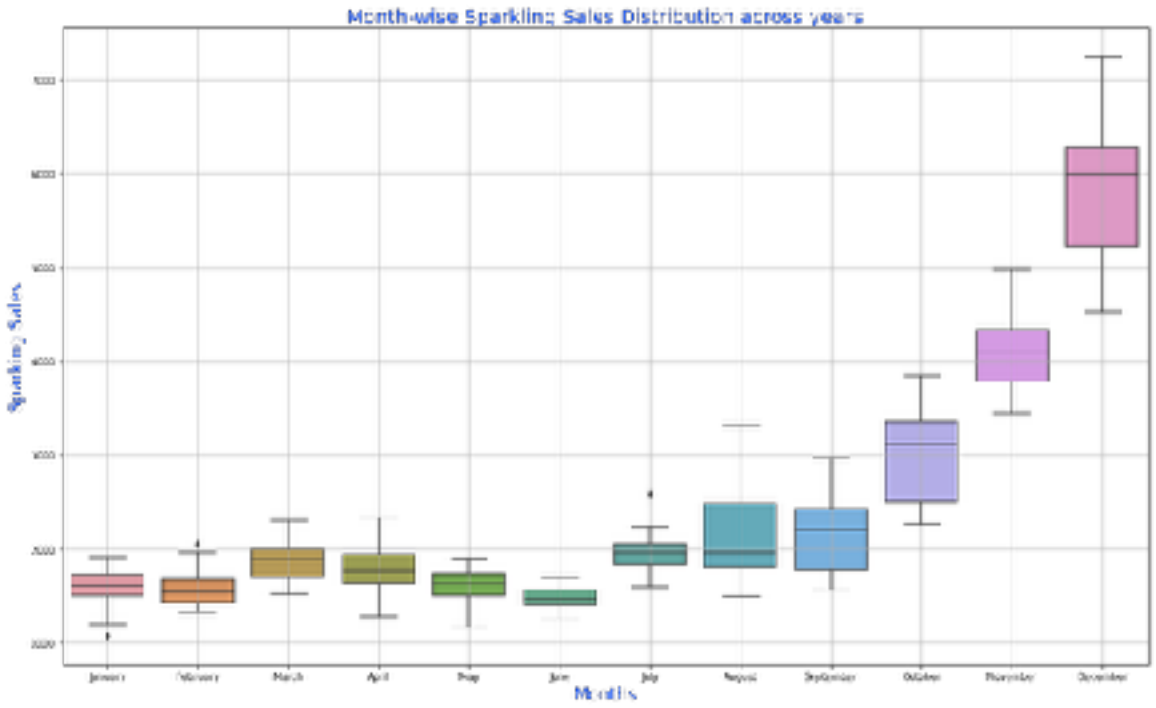
Year Month	April	Aug	Dec	Feb	Jan	July	June	March	May	Nov	Oct	Sept
1980	1712	2453	5179	1591	1686	1966	1377	2304	1471	4087	2596	1984
1981	1976	2472	4551	1523	1530	1781	1480	1633	1170	3857	2273	1981
1982	1790	1897	4524	1329	1510	1954	1449	1518	1537	3593	2514	1706
1983	1375	2298	4923	1638	1609	1600	1245	2030	1320	3440	2511	2191
1984	1789	3159	5274	1435	1609	1597	1404	2061	1567	4273	2504	1759
1985	1589	2512	5434	1682	1771	1645	1379	1846	1896	4388	3727	1771
1986	1605	3318	5891	1523	1606	2584	1403	1577	1765	3987	2349	1562
1987	1935	1930	7242	1442	1389	1847	1250	1548	1518	4405	3114	2638
1988	2336	1645	6757	1779	1853	2230	1661	2108	1728	4988	3740	2421
1989	1650	1968	6694	1394	1757	1971	1406	1982	1654	4514	3845	2608
1990	1628	1605	6047	1321	1720	1899	1457	1859	1615	4286	3116	2424
1991	1279	1857	6153	2049	1902	2214	1540	1874	1432	3627	3252	2408
1992	1997	1773	6119	1667	1577	2076	1625	1993	1783	4096	3088	2377
1993	2121	2795	6410	1564	1494	2048	1515	1898	1831	4227	3339	1749
1994	1725	1495	5999	1968	1197	2031	1693	1720	1674	3729	3385	2968
1995	1862	NaN	NaN	1402	1070	2031	1688	1897	1670	NaN	NaN	NaN

Inferences: The pivot-table reveals the following:

- The year with the highest total sales is 1988 (33,246) selling approx. 2770 units of wine every month
- The least total sales was recorded in 1982 (25,321), selling approx. 2110 every month. (we have discounted 1995, since we only have data for the first seven months).
- The month with highest sum of sales across the years is December, selling over 87,100 units of wine, averaging 5813 per year.
- The month with least total sales across years is June (23,572), averaging about 1473 per year.

—> Monthly sales distribution across years:

Figure: 1.2.2

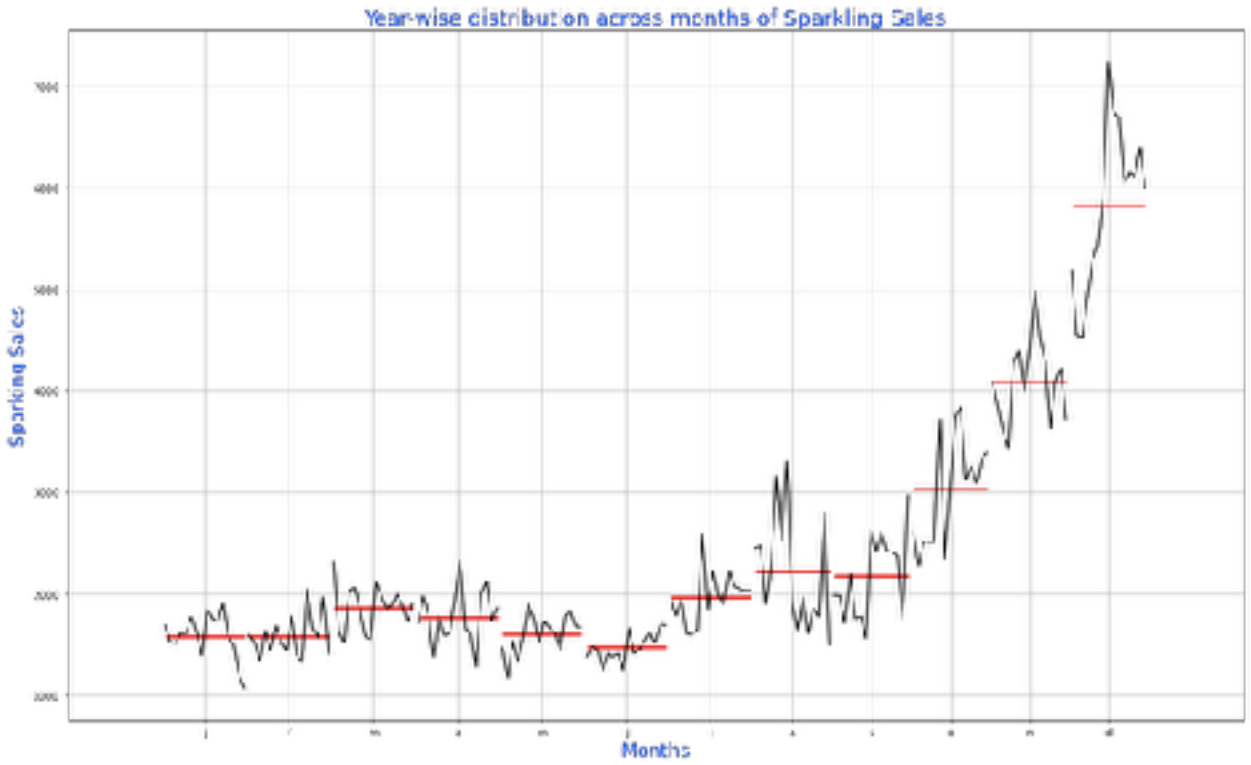


Inferences: This box-plot corroborates the observations of the pivot-table above, viz:

- December and June are the months with the highest and lowest distribution of wine sales respectively.

—> Year-wise sales distribution across months:

Figure: 1.2.3



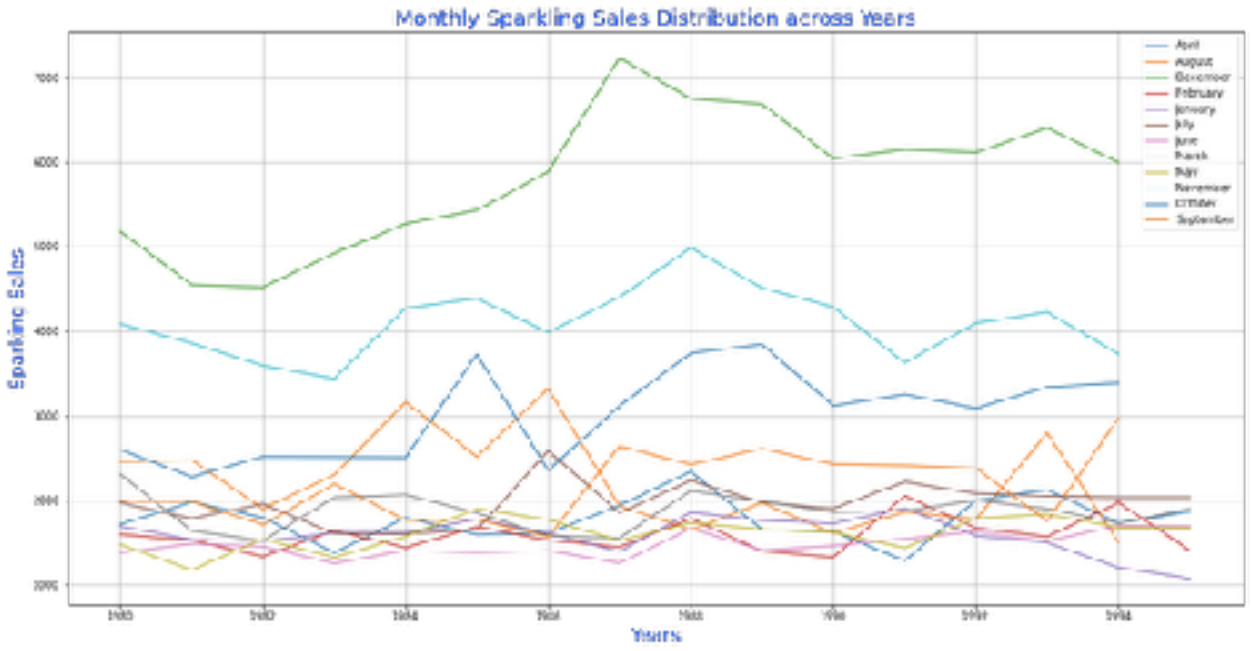
Inferences: This month-plot shows the year-wise distribution of wine sales for each month. It reinforces our inferences from the pivot-table, as given below:

- December is the month that has recorded the highest sales across years.
- June has consistently recorded the least sales across the years.

- 1995 is the year that gave the lowest sales figure (approx. 1100).
- Across most months, there seems to be a drop in sales towards 1995.

—> Monthly sales distribution across years:

Figure: 1.2.4



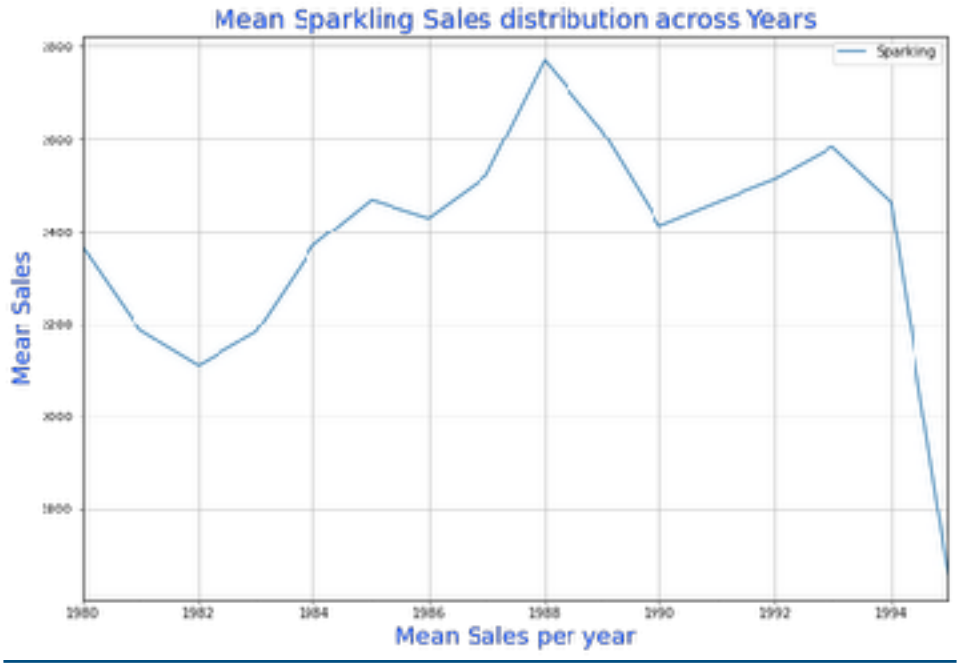
Inferences: the following data can be inferred from the above line-plot:

- The months of December, followed by November and then October, have contributed to the highest sales figures of ‘Sparkling’ wine.
- The months with least sales seem to be June, January, February and May.
- For most months, there seems to be a spurt in sales between the years 1986 and 1989.

- Between 1981 and 1983, there was an evident drop in sales across months.
- 1995 also shows a downward trend.

—> Mean ‘Sparkling’ sales across years:

Figure: 1.2.5

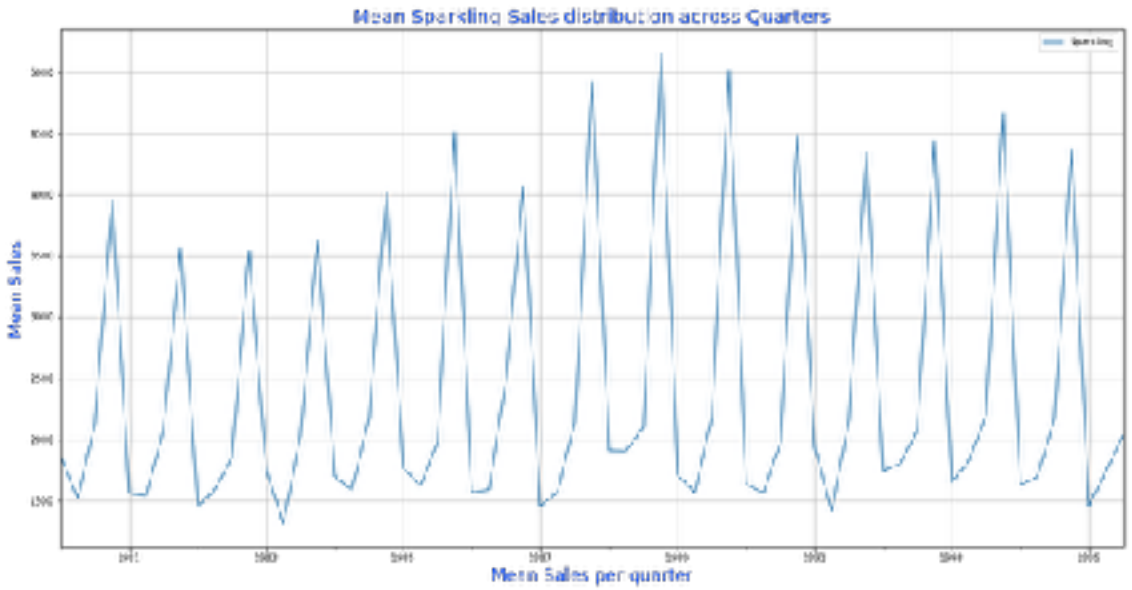


Inferences: This plot corroborates what has been inferred from the previous plots and the pivot-table:

- After an initial drop in sales in 1982, the sales figures peaked till about 1988, then dropped again in 1990, went up till 1993, and then shows a downward trend again.
- We can observe that there is no steady trend in sales across the years, it keeps rising and falling.

—> Mean ‘Sparkling’ sales across quarters:

Figure: 1.2.6



Inferences:

- The quarterly sales reveal a very strong seasonality in sales distribution. We can clearly see that every year begins with very low sales, then peaks to the maximum during the 4th quarter (October, November and December).
- The seasonality has a very strong, steady pattern, which is similar across the years.

- The trend component, as we saw in the previous plot, is not steady, and keeps rising and falling.

Outlier check:

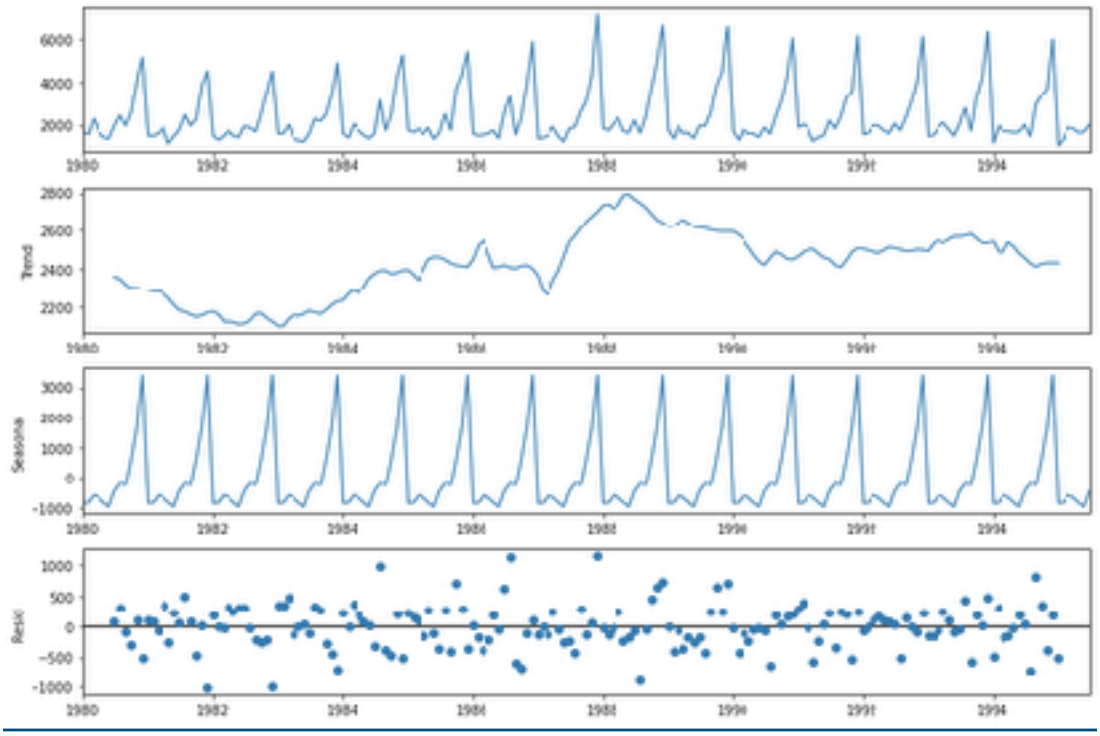
As seen in Figure: 1.2.1, there are a few outliers in the data, but they are not substantial or numerous enough to warrant an outlier treatment. Hence, outliers have not been treated here.

Time Series decomposition:

Decomposition aids to segregate the trend, seasonality and the residuals/ error elements in the data. Both additive and multiplicative decomposition was performed on the given data using the seasonal_decompose() function, as given below:

—> Additive decomposition:

Figure: 1.2.7

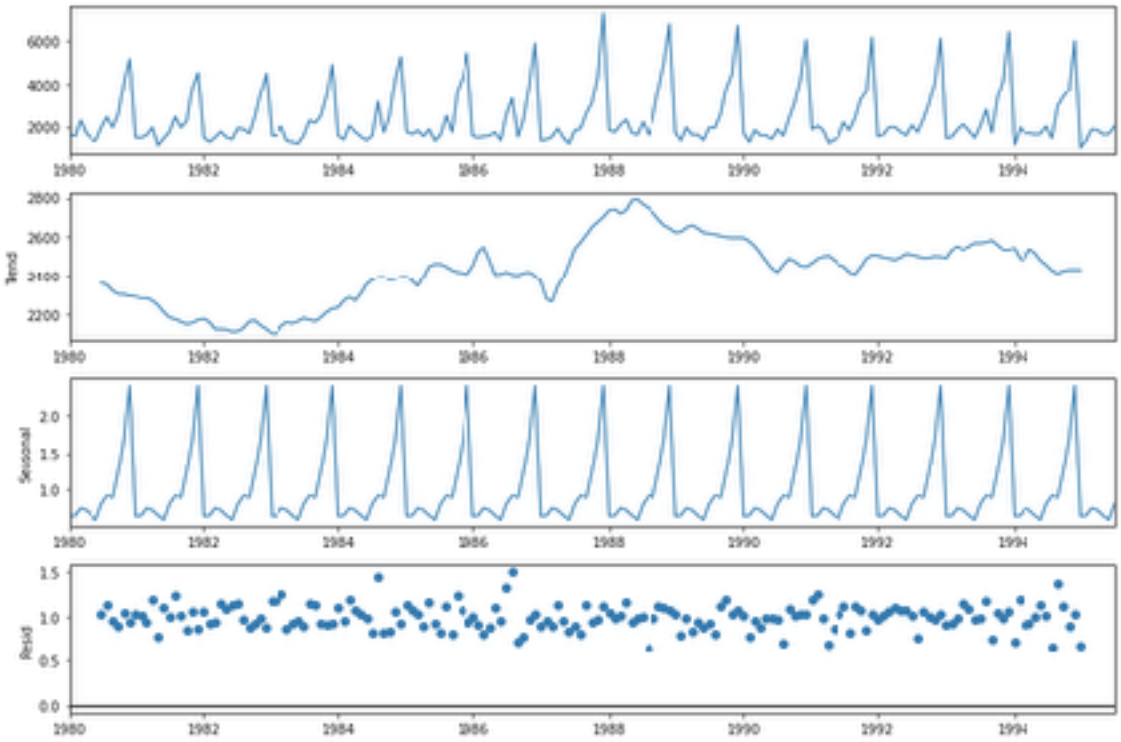


Inferences:

- We can see that there is lack of a specific pattern in the trend component. However, seasonality is very evident in the data.
- The scale of residuals varies from -1000 to +1000. Based on the plotting of the residuals, we can infer that error component is not randomly distributed - it still carries some information/pattern in it.
- Thus, we will not choose the additive decomposition to analyze this time series.

—> Multiplicative decomposition:

Figure: 1.2.8



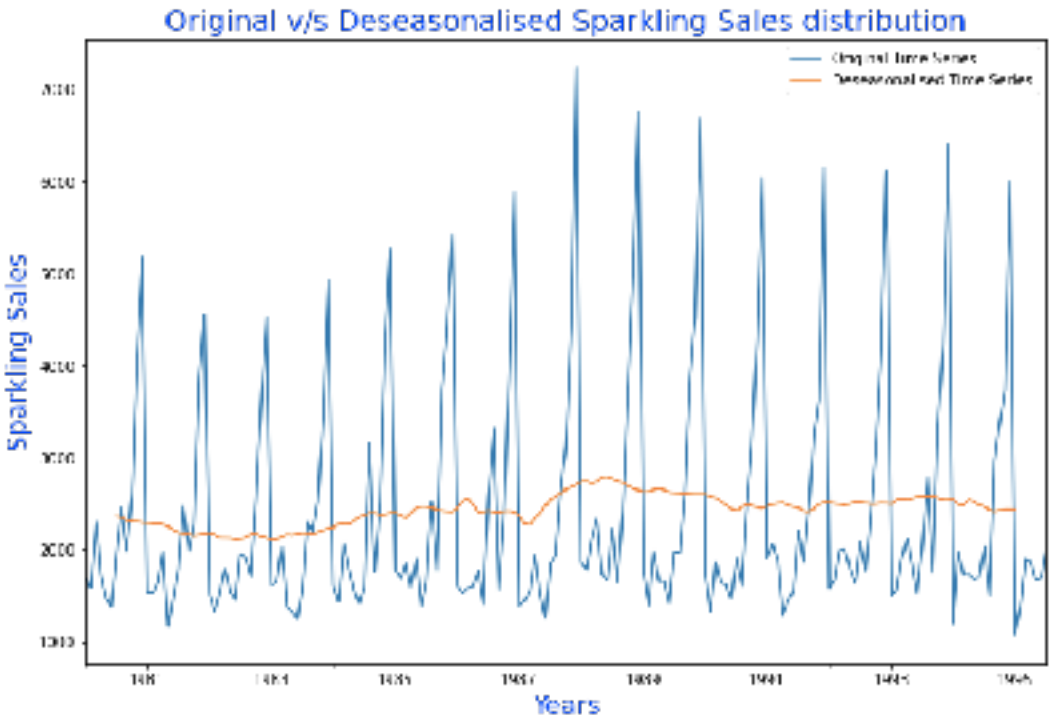
Inferences:

- Here, the residuals are centered around the point 1.0. Thus we can effectively say that error component is purely random, and does not conceal any information in it.
- Thus we will select Multiplicative decomposition to proceed with the time series analysis and forecasting.

—> Plotting of de-seasonalised time series:

To verify the choice of decomposition, a plotting of the deseasonalised data time series was done (with only trend and residual elements), as shown in graph below:

Figure: 1.2.9



Inferences:

- This plot reveals beyond doubt that once seasonality is removed, data is clearly represented in just one wavy line. Thus, we can affirm that seasonality is a major component of this data.

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

1.3 Splitting data into train and

The dataset was split using the following criteria:

Set:	Criteria:	Observations:
Train	Data uptil 1991	132
Test	Data from 1991 onwards	55

The first five rows of the training and test set are given below:

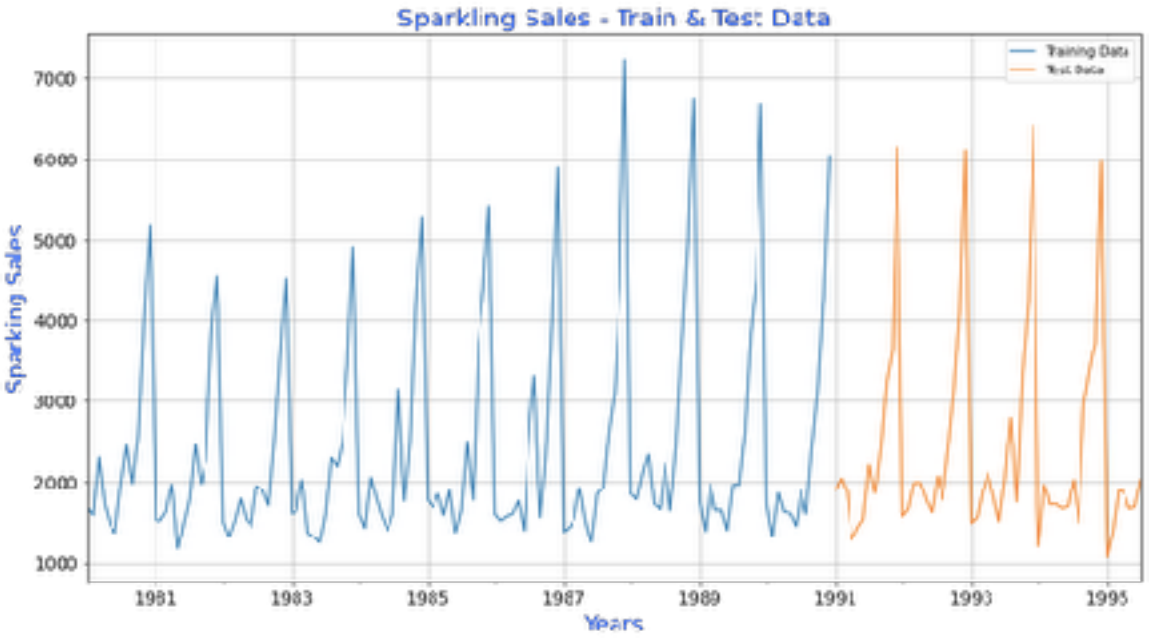
Table: 1.3

Training set:	
YearMonth	Sparkling
1980-01-01	1686

1980-02-01	1591
1980-03-01	2304
1980-04-01	1712
1980-05-01	1471

Post splitting, the train and test data was plotted to cross-check the split, as given in figure overleaf:

Figure: 1.3



1.4 Exponential Smoothing:

- Exponential smoothing is a forecasting technique which is an extension of the weighted moving average method (wherein higher weights are given to more recent observations).
- There are various exponential smoothing methods, some of which are:
    1. Single exponential smoothing (SES) - where trend and seasonality are absent
    2. Double exponential smoothing (DES) - where only trend is present
    3. Triple exponential smoothing (TES) - where both trend and seasonality are present
- In this case, since there is the presence of very strong seasonality, we will utilize the TES method first.

—> Model evaluation metric:

- Throughout this case, the evaluation criteria used is the RMSE (root mean squared error). All models have been compared for effectiveness using their RMSEs.
- RMSE is a robust model evaluation metric. It is the square root of mean of squares of all error terms. It is derived as follows:

$$rmse = \sqrt{\frac{\sum_{t=1}^{n} (Y_t - \hat{Y}_t)^2}{n}}$$

where: n = no. of observations

t = time period

Y_t = actual value of observation

$Yhat_t$ = forecasted value of observation

1.4.1 Triple Exponential Smoothing (TES) using Holt's Winter method:

Using the ExponentialSmoothing() function from statsmodels.tsa.api, the TES model was built for the time series.

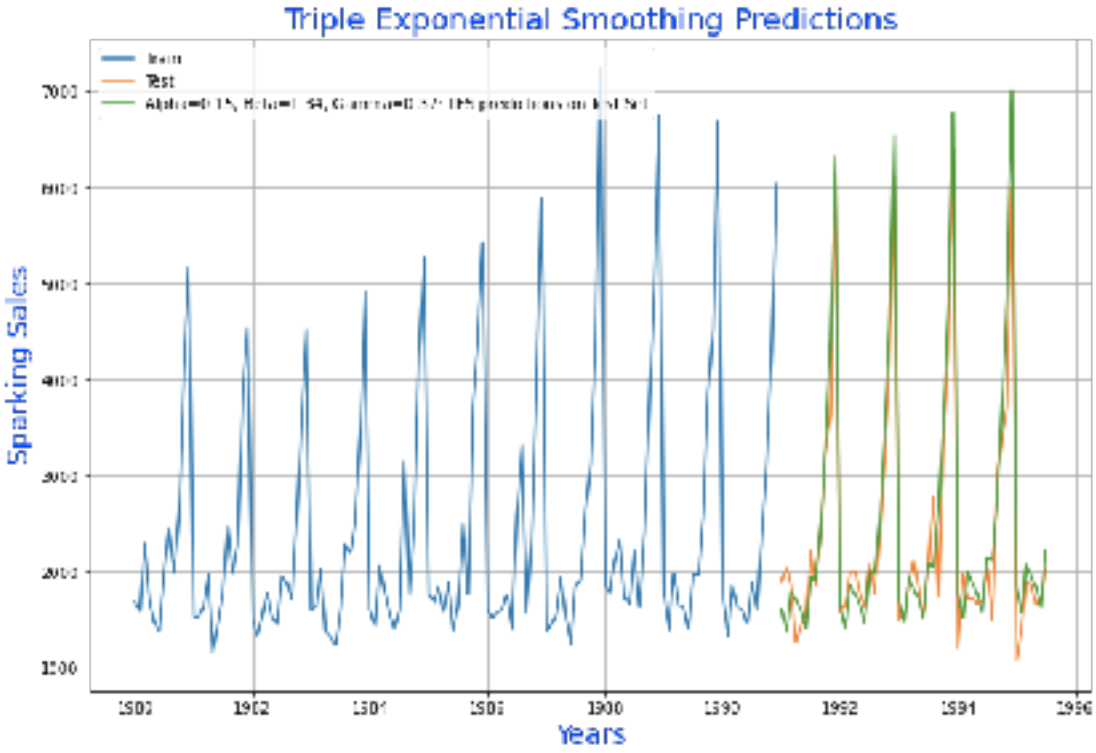
The best parameters for the automated TES model are given below:

Parameter:	Description:	Value:
alpha	smoothing level	0.1533
beta	smoothing slope	1.34E-20
gamma	smoothing seasonal	0.3690

Test predictions:

The TES model built was used to predict sales on the test data, which is plotted in the graph overleaf:

Figure: 1.4.1



Observations:

The forecast on test data (green line) seems to be fairly accurate, it is very close and almost mimics the test data (orange line).

Model evaluation:

RMSE for TES was computed, and shared in a DataFrame, to be used for later comparison with RMSEs of other models, as given below:

Model:	Test RMSE:
TES: Alpha=0.15, Beta=1.34, Gamma=0.37	392.93

1.4.2 Triple Exponential Smoothing (TES) with tweaked parameters:

The TES model was rebuilt with 'trend' parameter as 'Additive', to check if it would have a bearing on the final forecast effectiveness.

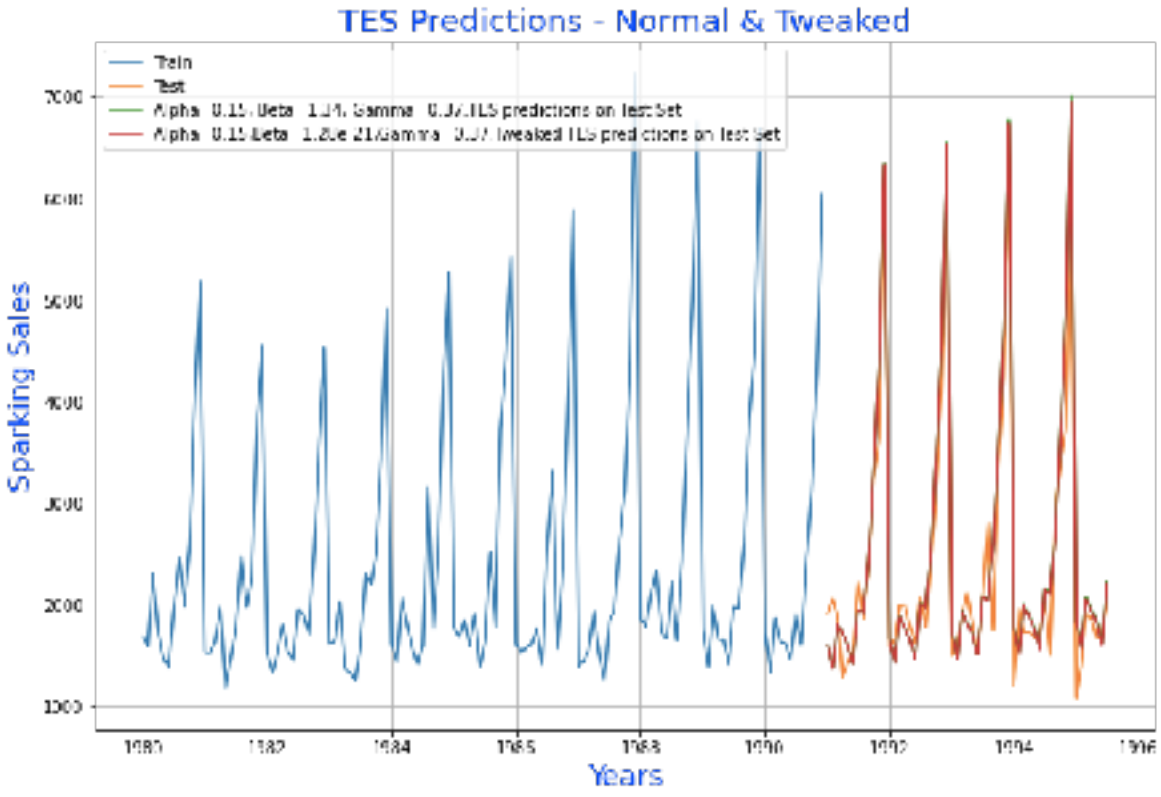
The best parameters for the tweaked TES model are given below:

Parameter:	Description:	Value:
alpha	smoothing level	0.1542
beta	smoothing slope	1.27E-21
gamma	smoothing seasonal	0.3713

Test predictions:

The tweaked TES model was used to predict sales on the test data. A comparison plot of the normal and tweaked TES against test data is given below:

Figure: 1.4.2



Observations:

The tweaked TES forecasts on the test data are almost identical to the previous TES forecasts - the red line (tweaked TES forecasts) almost completely cover the green line (previous TES forecasts).

Model evaluation:

RMSE for tweaked TES was computed, as given below:

Model:	Test RMSE:
TES_tweaked: Alpha=0.15, Beta=1.28e-21, Gamma=0.37	383.14

1.4.3 Double Exponential Smoothing (DES) using Holt's method:

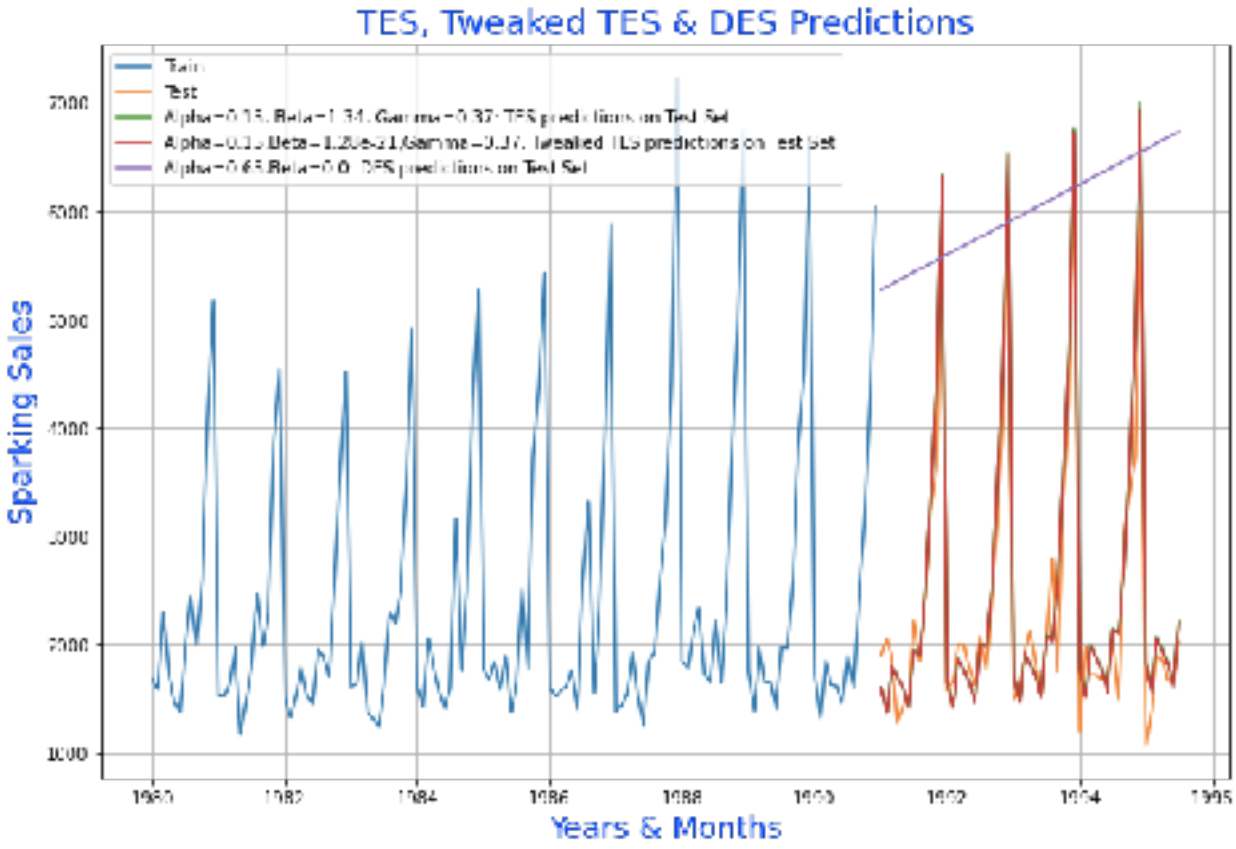
Despite the presence of seasonality in the data, the DES model was built using the Holt() function from statsmodels.tsa.api (this exercise was done for sheer comparison sake), which gave the following best parameters:

Parameter:	Description:	Value:
alpha	smoothing level	0.6478
beta	smoothing slope	0.0

Test predictions:

The DES model test predictions were plotted in a comparison plot along with the test predictions of previous models, as given overleaf:

Figure: 1.4.3



Observations:

The DES forecasts on the test data is a straight-line prediction (as seen in purple line). This is very unrealistic prediction, proving that the DES model is not apt in this case.

Model evaluation:

RMSE for DES is as given below:

Model:	Test RMSE:
DES: Alpha=0.65,Beta=0.0	3851.28

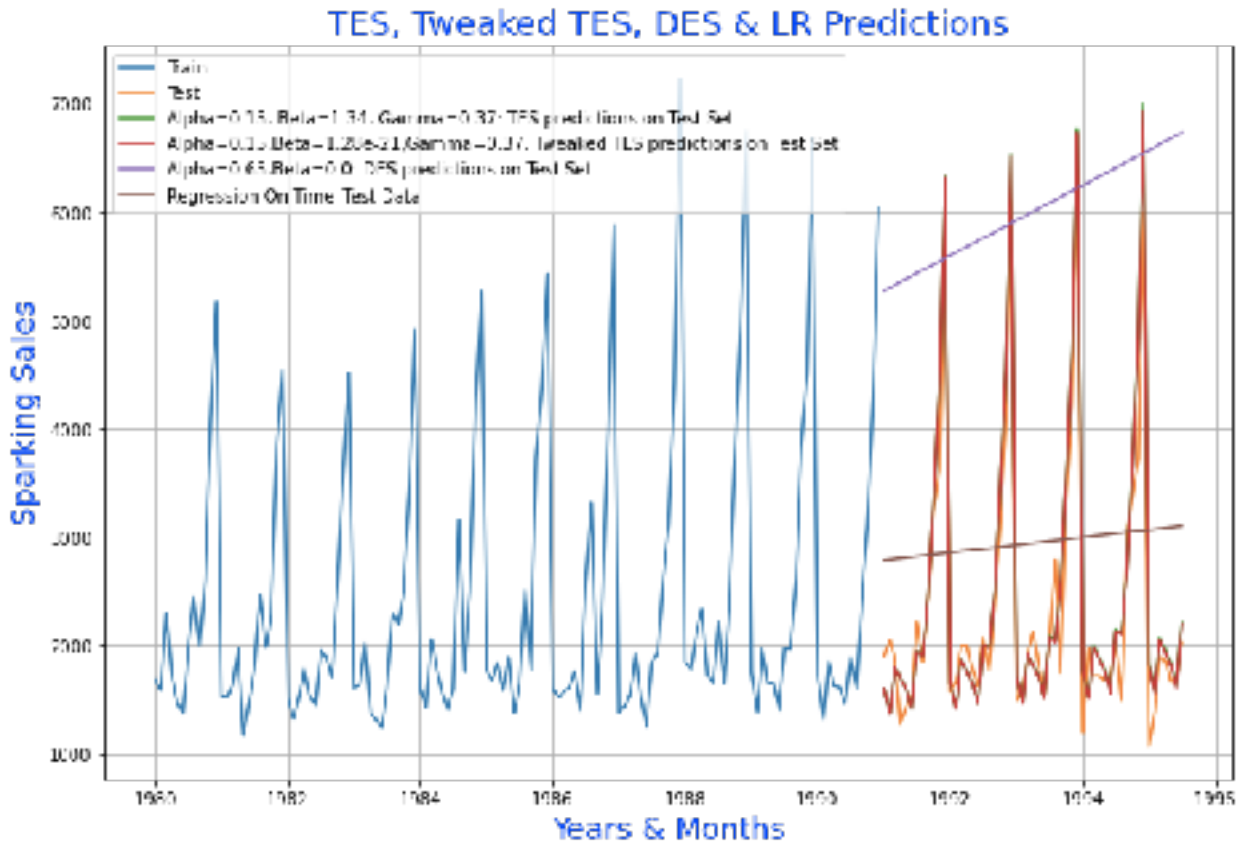
1.4.4 Linear Regression model:

- The Linear regression (LR) model was used for forecasting test data using the *LinearRegression()* function from *sklearn.linear_model*.
- For running the LR model, an time instance (cardinal numbering indexing) had to be given to the train ans test set, since the LR model cannot run with the DateTimeIndex.

Test predictions:

The LR model test predictions were plotted in a comparison plot along with the test predictions of previous models, as given below:

Figure: 1.4.4



Observations:

The LR forecasts on the test data is a straight-line prediction (as seen in brown line). This is a very unrealistic prediction, proving that the LR model is not apt in this case.

Model evaluation:

RMSE for LR model is as given below:

Model:	Test RMSE:
Linear regression	1389.14

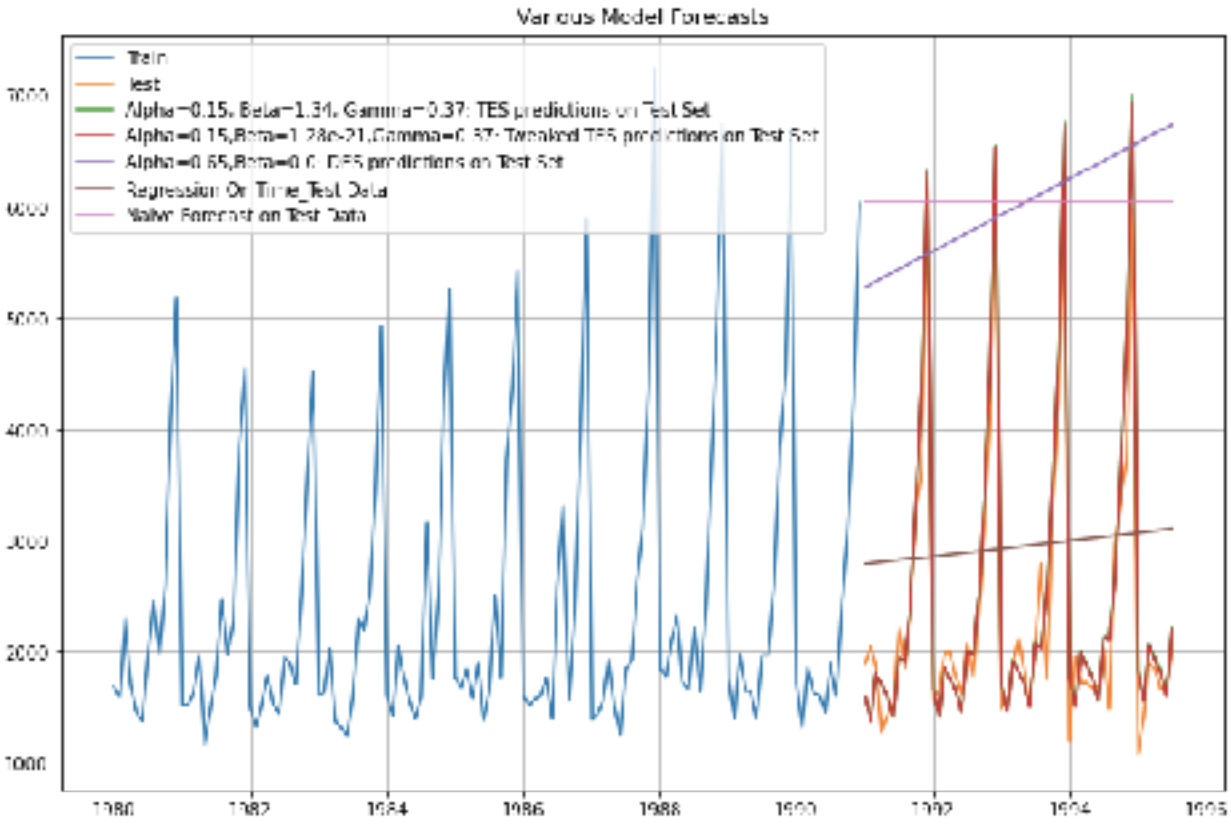
1.4.5 Naive approach model:

The Naive approach was used to make predictions on the test data, which essentially uses the final observation as the prediction.

Test predictions:

The Naive approach test predictions are plotted against other model predictions in the plot overleaf:

Figure: 1.4.5



Observations:

The Naive approach forecasts on the test data is again a straight-line prediction (as shown by pink line). This is a highly unrealistic prediction, as it does not even capture the trend element.

Model evaluation:

RMSE for Naive model is given below:

Model:	Test RMSE:
Naive approach	3864.28

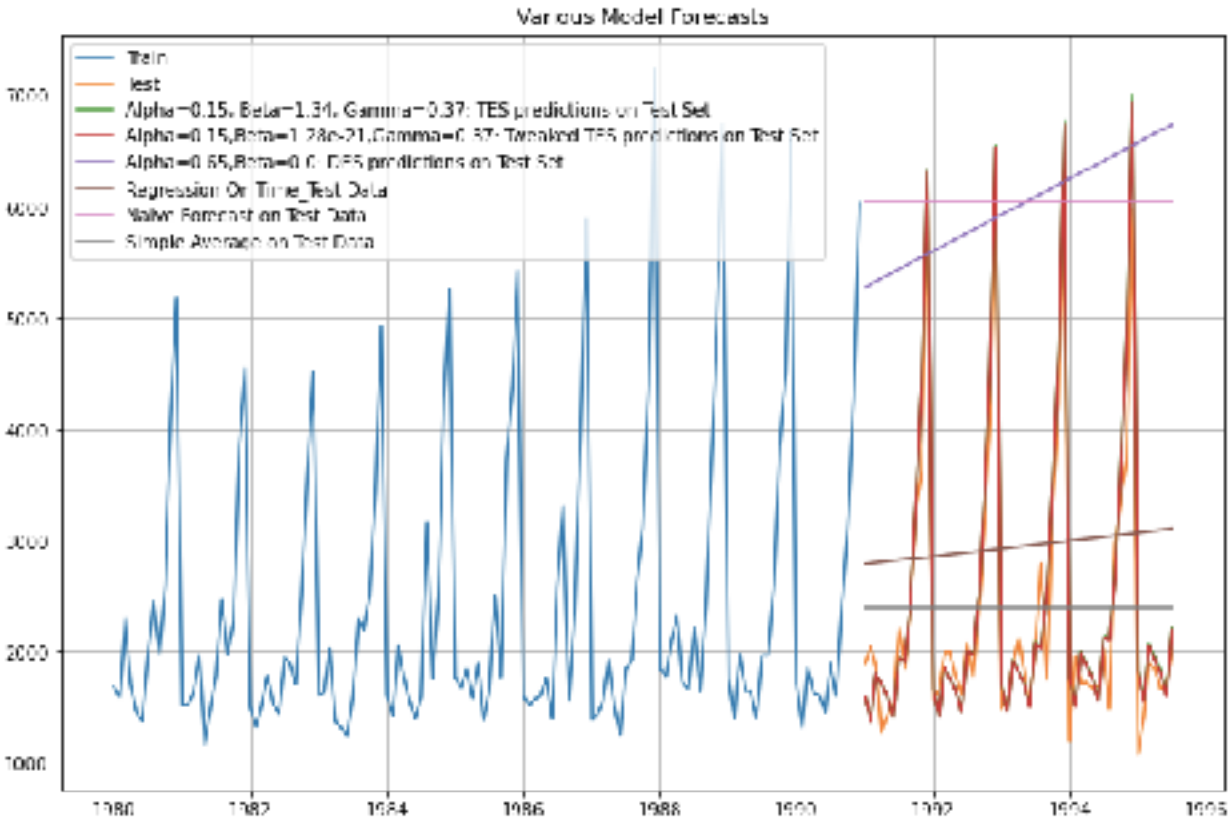
1.4.6 Simple Average method:

The Simple average (SA) method was used to make predictions on the test data, which essentially uses the mean of the train data to make predictions.

Test predictions:

The SA test predictions are plotted against other model predictions in the plot below:

Figure: 1.4.6



Observations:

The SA predictions on the test data is again an absolute straight-line prediction (as shown by grey line), that does not even capture the trend element.

Model evaluation:

RMSE for Naive model is given below:

Model:	Test RMSE:
Simple Average	1275.08

1.4.7 Moving Average method:

The Moving average (MA) method uses the mean of a specified number of sliding window / trailing observations to make predictions on the test data. Here, we have taken four different trailing windows for mean computation - 2, 4, 6 and 9, in order to test out their effectiveness.

Test predictions:

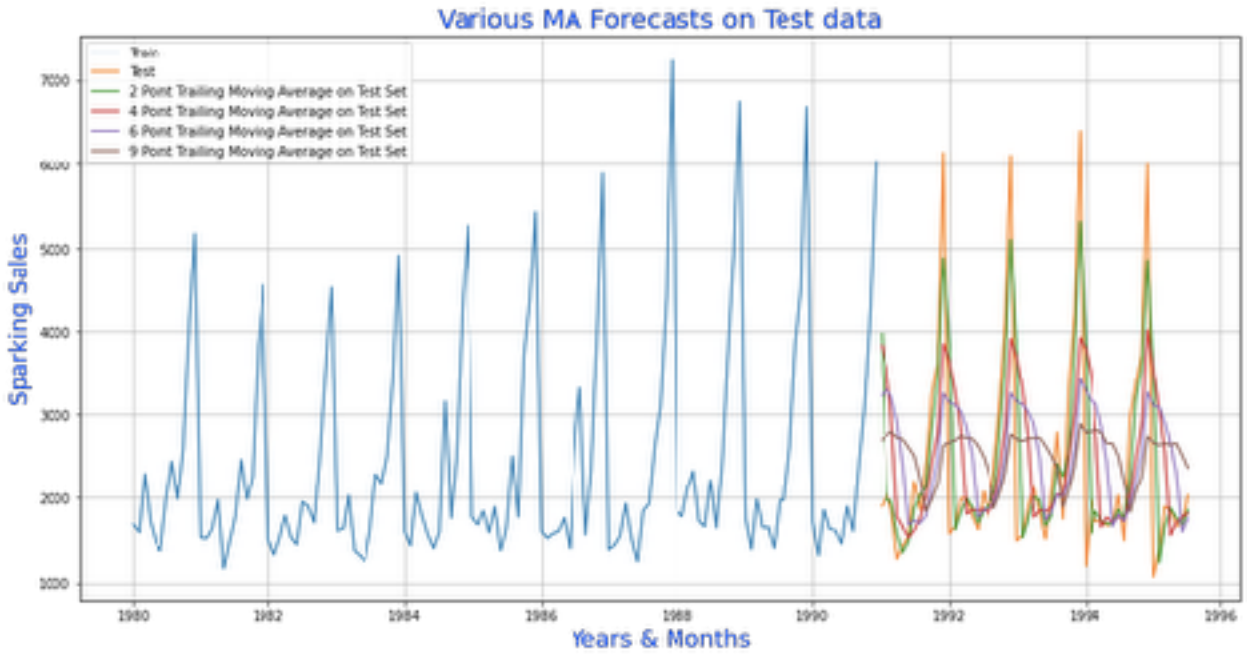
The below table shows the first ten MA predictions for the various trailing windows:

Table: 1.4.7

YearMonth	Sparkling	Trailing_2	Trailing_4	Trailing_6	Trailing_9
1980-01-01	1686	NaN	NaN	NaN	NaN
1980-02-01	1591	1638.5	NaN	NaN	NaN
1980-03-01	2304	1947.5	NaN	NaN	NaN
1980-04-01	1712	2008	1823.25	NaN	NaN
1980-05-01	1471	1591.5	1769.5	NaN	NaN
1980-06-01	1377	1424	1716	1690.17	NaN
1980-07-01	1966	1671.5	1631.5	1736.83	NaN
1980-08-01	2453	2209.5	1816.75	1880.50	NaN
1980-09-01	1984	2218.5	1945	1827.17	1838.22
1980-10-01	2596	2290	2249.75	1974.50	1939.33

The MA test predictions are plotted against other model predictions in the plot below:

Figure: 1.4.7



Observations:

Out of the MA predictions on the test data, the 2-point trailing average (demonstrated by green line) gives the most accurate prediction, closest to the test data as compared to the other three trailing windows.

Model evaluation:

RMSE for the four MA models are given below:

Model:	Test RMSE:
2-point MA	813.40
4-point MA	1156.59
6-point MA	1283.93
9-point MA	1346.28

1.5 Checking stationarity of the data:

A stationary time series is one where the mean and the variance of the series is constant over a period of time, and the correlation between two observations depends only on the distance/lag between them.

Stationarity check on train data:

- Checking the stationarity of the train data is an essential step before proceeding to build more complex forecast models like ARIMA and SARIMA.
- Stationarity need not be checked for the test data, since models will only be trained on the train data. Therefore, test data features will not affect the predictions.
- To check for stationarity, the Augmented Dickey-Fuller (ADF) test was used on the train, using the adfuller() function from statsmodels.tsa.stattools.
- With the derived ADF test statistics, hypothesis testing was done to come to a conclusion. The hypothesis were:

  H_0 = Series is non-stationary

  H_1 = Series is stationary

- The ADF test gave the following results:

```
Test statistic is -2.062
Test p-value is 0.567
Number of lags used 12
```

Conclusion:

At 95% confidence interval: **p-value (0.567) > alpha (0.05)**

Therefore, the time series is **non-stationary**.

Differencing the series:

Level-1 differencing was done on train data to make the non-stationary time series stationary, using the diff() function.

Post level-1 differencing, the following test stats were derived:

```
Test statistic is -7.968
Test p-value is 8.479E-11
Number of lags used 11
```

Conclusion:

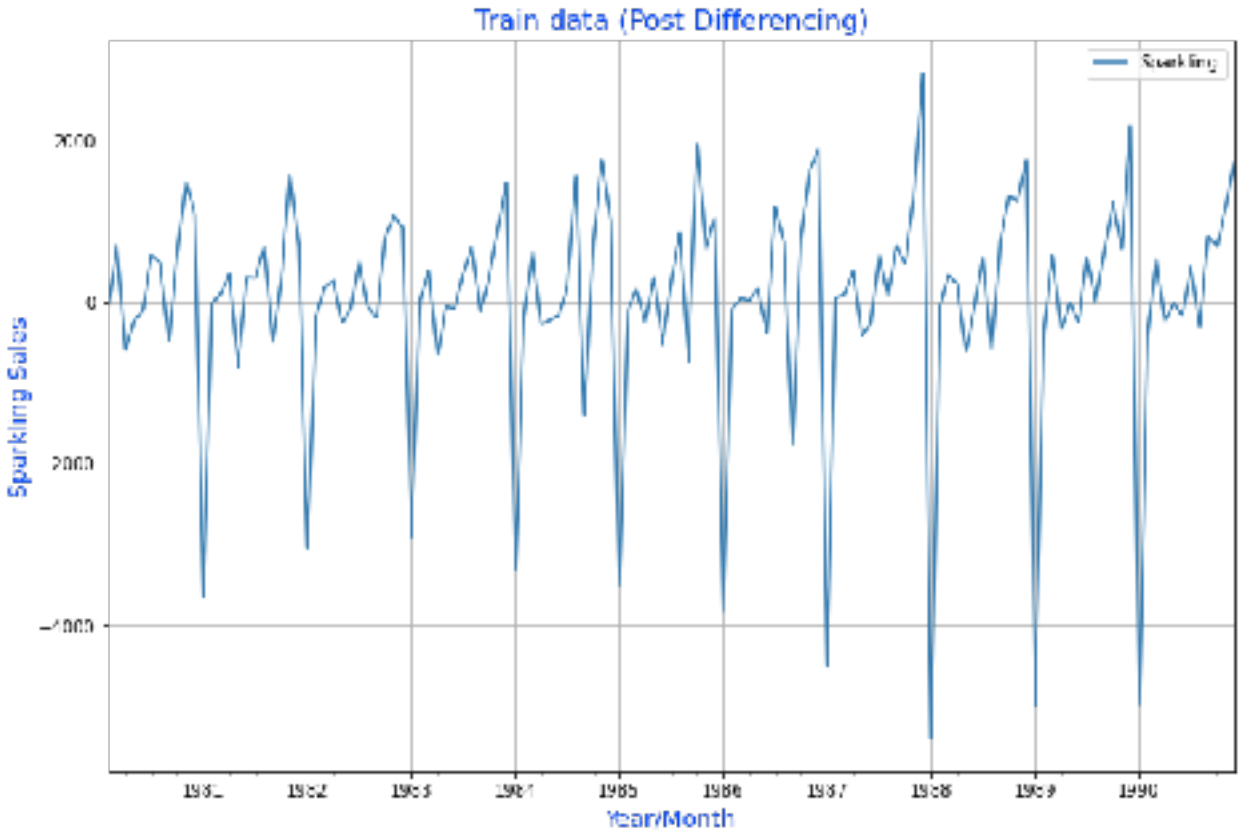
At 95% confidence interval: **p-value (8.479E-11) < alpha (0.05)**

Therefore, the time series is now **stationary.**

Plotting the differenced time series:

The time series post differencing, which is stationary in nature, is as shown overleaf:

Figure: 1.5


Train data (Post Differencing)
Sparkling
Sparkling Sales
Year/Month

1.6 Building automated ARIMA/SARIMA model:

- Auto-regressive Integrated Moving Average (ARIMA) models are powerful models that can make future forecasts on time series data.
- However, in this case, wherein the series demonstrates a very strong seasonality, a simple ARIMA will not suffice.
- Here we will employ the SARIMA (Seasonal Auto-regressive Integrated Moving Average) model, so that the seasonal adjustments are fully accounted for in the future predictions.
- The SARIMA can be defined as a function of the ARIMA, being represented as **SARIMA(p,d,q)(P,D,Q)[m]**, which is equal to:

ARIMA (p,d,q) * ARIMA (P,D,Q)[m]

where: p = no. of auto-regressive components

d = no. of differencing done

q = no. of moving average components

P = seasonal auto-regressive component

D = seasonal differencing

Q = seasonal moving average components

m = frequency of observations

Parameter generation:

- Using the 'itertools' function, random parameter combinations for the SARIMA model were automatically generated.
- With the sm.tsa.statespace.SARIMAX() function from statsmodels.api, various combinations of the random parameters were tested to a maximum iteration of 1000, to find the best parameters to build the automated model.

Scoring and selecting the best SARIMA model:

- The Akaike Information Criterion (AIC) metric was used to select the best parameter. The parameters with least AIC values was chosen to build the model with.
- Post running the iteration with random parameters, a data frame was generated with the least-AIC parameters, as shown below:

Table: 1.6

Parameter	Seasonal parameter	AIC
(3, 1, 2)	(3, 0, 0, 12)	1387.23
(3, 1, 1)	(3, 0, 0, 12)	1387.79
(3, 1, 2)	(3, 0, 1, 12)	1388.60
(3, 1, 1)	(3, 0, 1, 12)	1388.68
(3, 1, 3)	(3, 0, 0, 12)	1389.14

- Based on this information, SARIMA(3,1,2)(3,0,0,12) was chosen as the best parameter for automated SARIMA generation.

Generating the auto-SARIMA model:

The auto-SARIMA model was built and the following model diagnostics was generated.

Figure: 1.6.1

SARIMAX Results

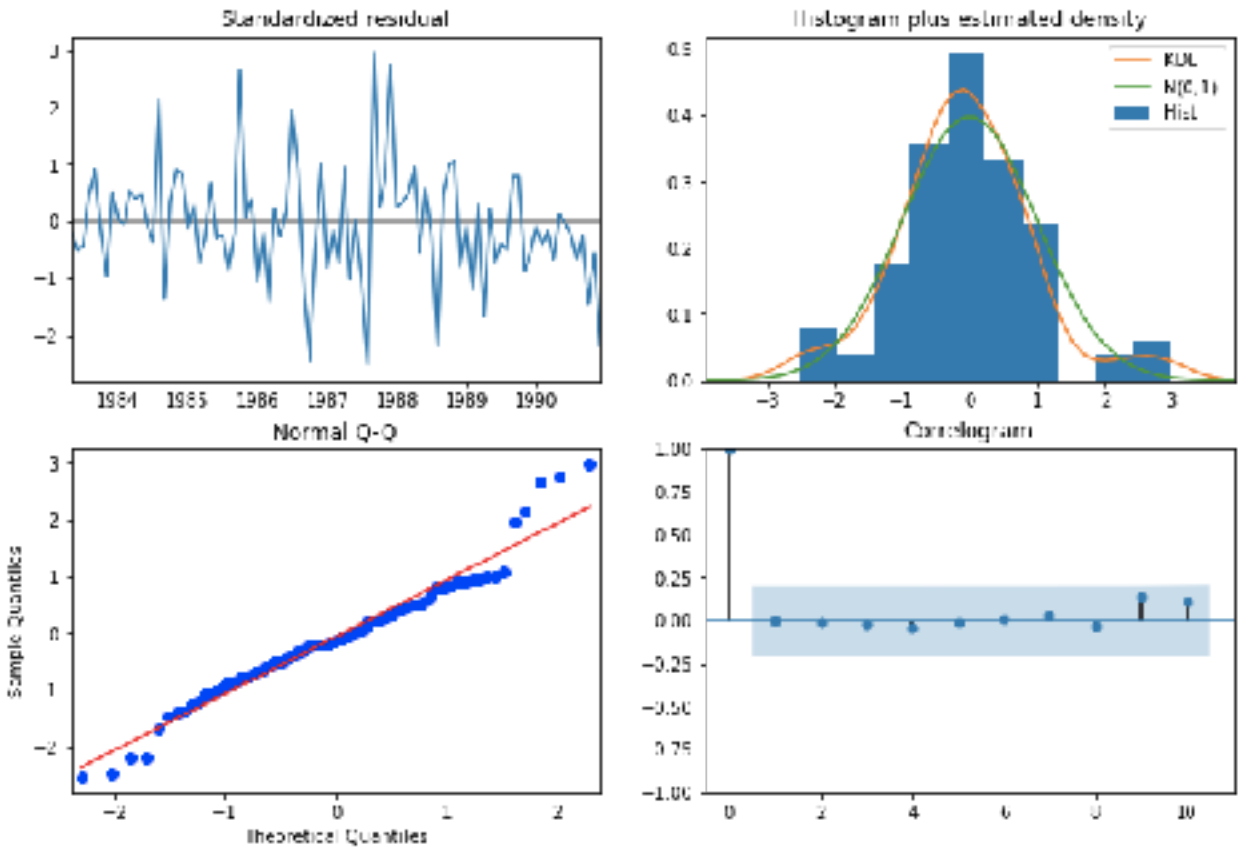
Dep. Variable:		Sparkling	No. Observations:		132	
Model:	SARIMAX(3, 1, 2)x(3, 0, [], 12)		Log Likelihood		-684.617	
Date:	Sat, 09 Oct 2021		AIC		1387.235	
Time:	07:01:11		BIC		1409.931	
Sample:	01-01-1980		HQIC		1396.395	
	- 12-01-1990					
Covariance Type:	opg					

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.5372	0.339	-1.586	0.113	-1.201	0.126
ar.L2	0.0256	0.187	0.137	0.891	-0.340	0.392
ar.L3	0.0785	0.130	0.604	0.546	-0.176	0.333
ma.L1	-0.1878	0.326	-0.575	0.565	-0.828	0.452
ma.L2	-0.6875	0.272	-2.531	0.011	-1.220	-0.155
ar.S.L12	0.5713	0.103	5.542	0.000	0.369	0.773
ar.S.L24	0.2605	0.117	2.222	0.026	0.031	0.490
ar.S.L36	0.2126	0.111	1.916	0.055	-0.005	0.430
sigma2	1.682e+05	2.52e+04	6.671	0.000	1.19e+05	2.18e+05

Ljung-Box (Q):	27.31	Jarque-Bera (JB):	8.81
Prob(Q):	0.94	Prob(JB):	0.01
Heteroskedasticity (H):	1.17	Skew:	0.36
Prob(H) (two-sided):	0.67	Kurtosis:	4.33

A diagnostic plot was drawn to check the nature of the residuals in the model, as shown overleaf:

Figure: 1.6.2



Evaluating the auto-SARIMA model:

For the model evaluation, two metrics were used:

- RMSE - root mean square error (square root of mean of squares of all error terms)
- MAPE - mean absolute percentage error (mean of absolute difference between error terms, represented as a percentage)

The evaluation results for the auto-SARIMA is given below:

	Test RMSE	MAPE
SARIMA(3,1,2)(3,0,0,12)	543.04	23.23

1.7 Building manual SARIMA model:

In order to build manual SARIMA models, the parameters p,d,q and seasonal parameters P,D,Q will have to be manually selected. For this purpose, the following were plotted:

- ***ACF (auto-correlation function)*** - measures correlation of current observations with past observations
- ***PACF (partial auto-correlation function)*** - measures correlation between current and k-lagged series by removing intermediate observations.

ACF and PACF plots:

Figure: 1.7.1

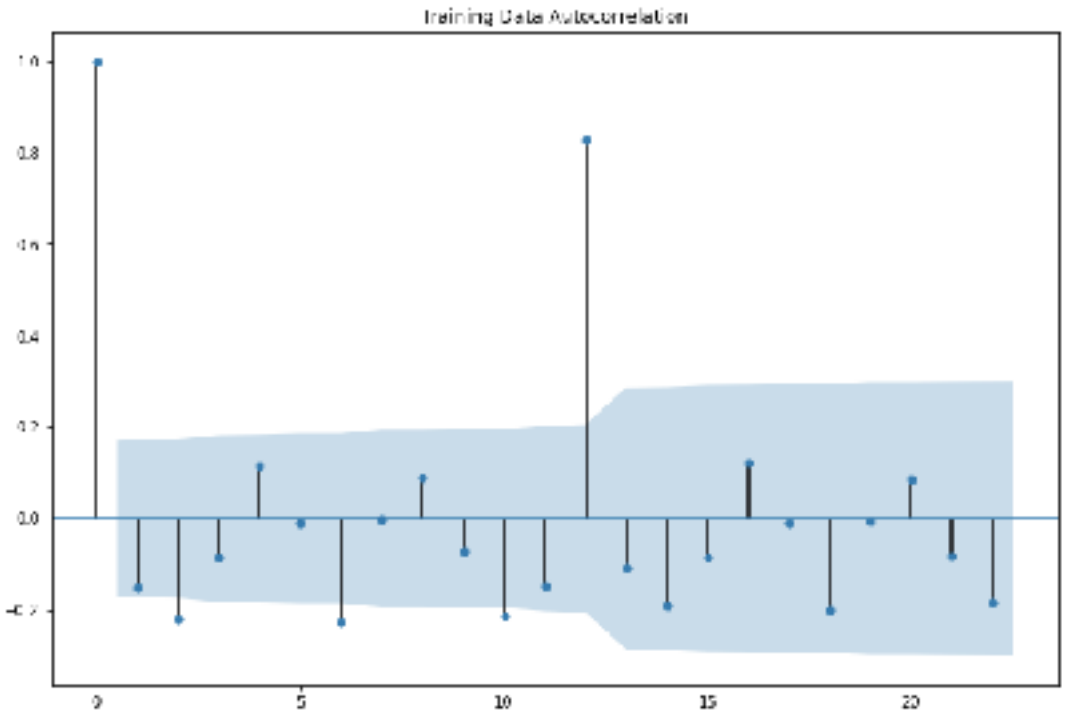
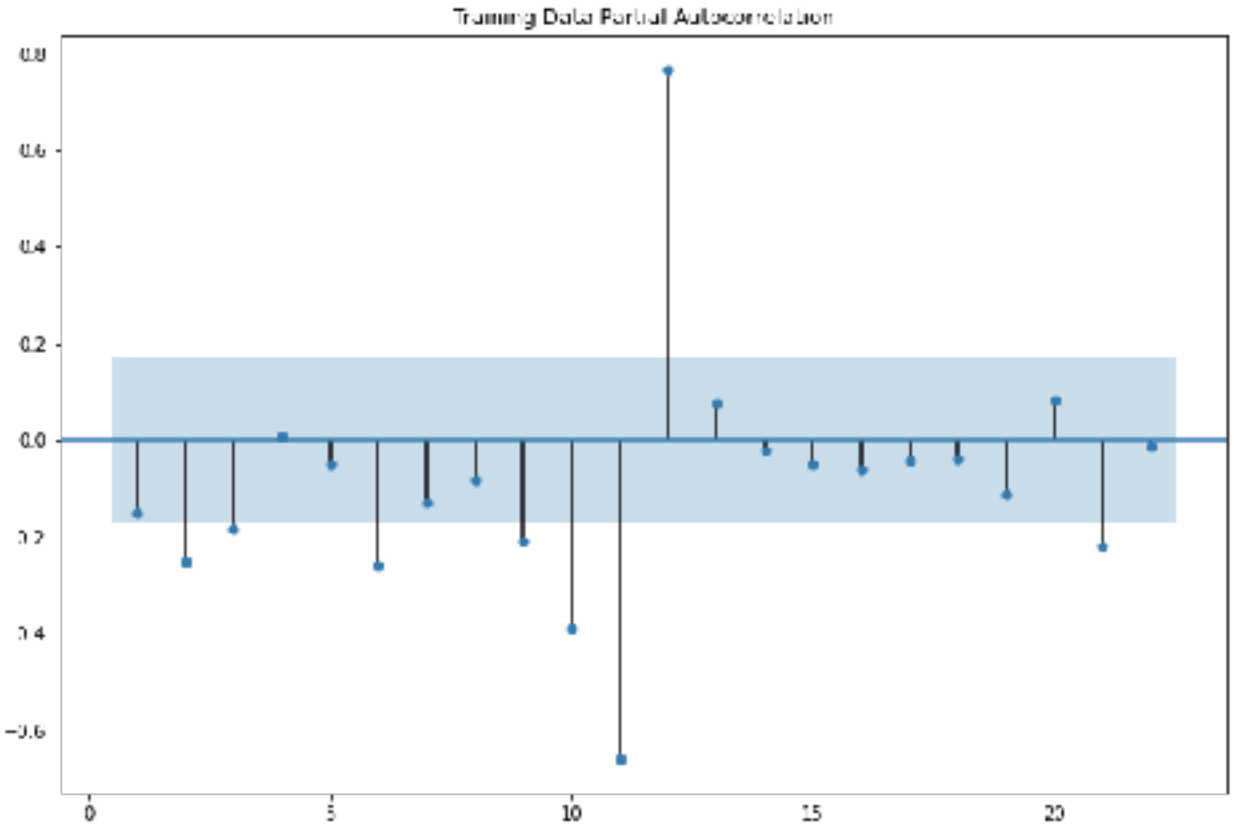


Figure: 1.7.2



As per the ACF and PACF plots, we can derive the following values:

Parameters	p=0, d=1, q=0
Seasonal parameters	P=0, D=1, Q=0, m=12

The manual SARIMA model was built and test data forecasted using the above selected parameters.

The evaluation results for the manual-SARIMA is given below:

	Test RMSE	MAPE
SARIMA(0,1,0)(0,1,0,12)	3864.279	201.32

Additional SARIMA models with random parameters:

Three additional models were built using the following:

Parameters	Seasonal parameters
(1,1,1)	(0,0,0,12)
(3,1,2)	(0,0,1,12)
(3,1,2)	(0,0,1,6)

1.8 Model evaluation:

The RMSE values were compared to decide on the best model to use for making the future sales predictions. The least RMSE suggests the most effective model. Tabular comparison of model-wise RMSE values is given below:

Table: 1.8

Model	Test RMSE
TES: Alpha=0.15, Beta=1.34, Gamma=0.37	393
TES_tweaked: Alpha=0.15, Beta=1.28e-21, Gamma=0.37	383
DES: Alpha=0.65,Beta=0.0	3851
LR RSME	1389
Naive RSME	3864
SA RSME	1275
2-point MA	813
4-point MA	1157
6-point MA	1284
9-point MA	1346
SARIMA(3,1,2)(3,0,0,12)	543
SARIMA(0,1,0)(0,0,0,12)	3864
SARIMA(1,1,1)(0,0,0,12)	1325
SARIMA(3,1,2)(0,0,1,12)	1207
SARIMA(3,1,2)(0,0,1,6)	1269

Conclusion:

Based on the RMSE values, the best models are:

- TES (Alpha=0.15, Beta=1.34, Gamma=0.37)
- SARIMA (2,1,2) (1,1,1,12)

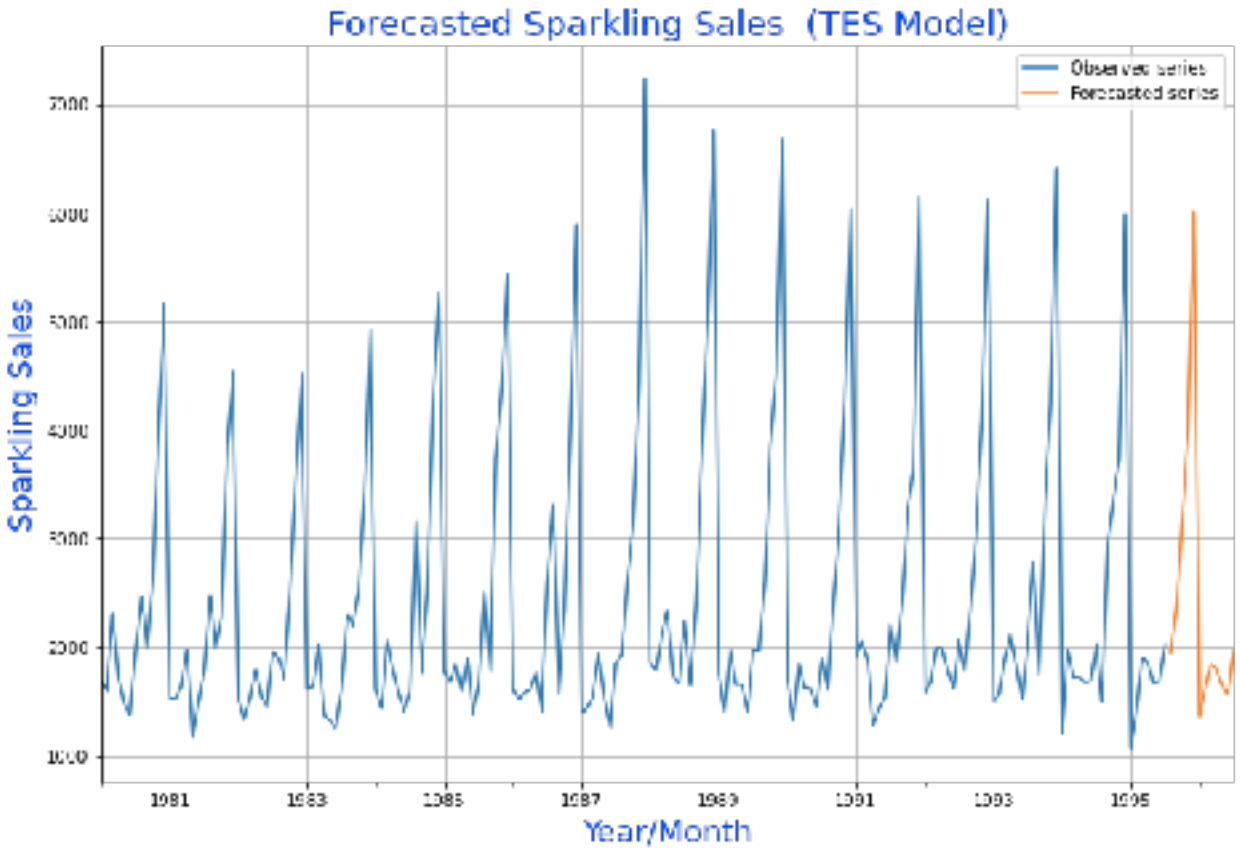
1.9 Forecasting future sales:

- Since the Moving Average method is good for predictions on the test data, it is not really ideal in making future forecasts. Hence we will only use the TES and SARIMA model for forecasting future sales.
- The time frame for which forecast is to be done is 12 months, i.e. from 01-08-1995 to 01-07-1996.

—> **Forecasts with TES**:

The TES predictions for the future sale of Rose wine for the next 12 months is as plotted below:

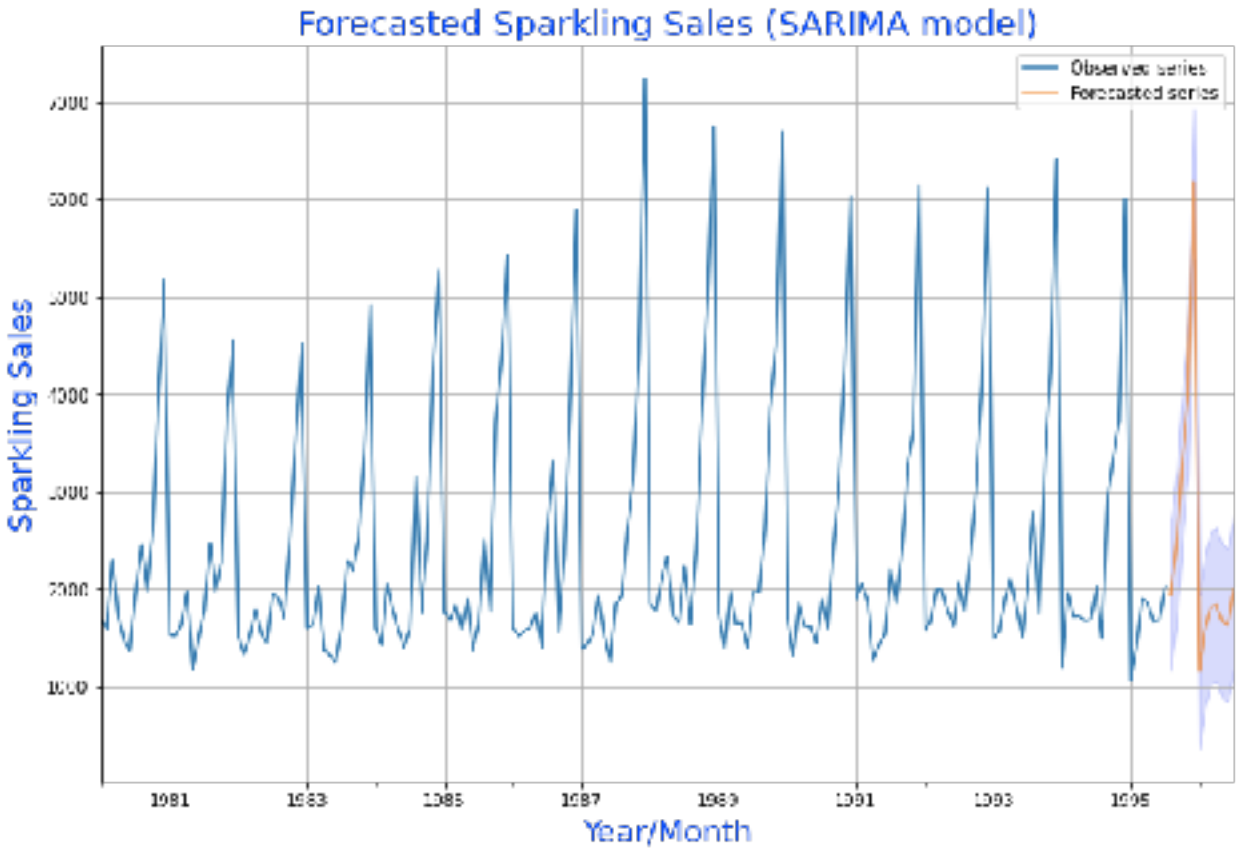
Figure: 1.9.1



—> **Forecasts with SARIMA**:

The SARIMA predictions for the future sale of Rose wine for the next 12 months (with confidence interval marked in light blue) is as plotted below:

Figure: 1.9.2



1.10 Comments and business insights:

The following can be observed from the two different forecasts made:

—> **Observations on TES forecasts**:

The 12-month future forecast exhibits a steady trend, while still maintaining the seasonality of the observed time series.

—> **Observations on SARIMA forecasts**:

The 12-month forecasted series demonstrates a slightly higher sales figure, neither promising a wide increment or decline in sales of 'Sparkling' wine. It has also maintained the seasonal component of the observed data series.

—> Business insights:

- Data has shown a very steady trend in the sale of 'Sparkling' wine. Thus, ABC Estate wines must think of strategies to boost sales.
- The company needs to analyze the reasons behind this stagnation.
- They can opt for a customer survey to gauge the exact reasons for the stagnant sales.
- Data shows a steep seasonal spike during the final quarter of the year, perhaps due to festivities like Christmas and New Year, when people indulge in wine.

- The company can promote sales by offering many schemes and slash prices during the festive season.
- For the periods when sales are very slow, especially the first and second quarter, certain discounts and other offers can be given.