TIME SERIES FORECASTING BUSINESSS REPORT

THANUSRI A

14-04-2024

PROBLEM

Define the problem and perform Exploratory Data Analysis
Read the data as an appropriate time series data - Plot the data - Perform EDA - Perform Decomposition

Data Pre-processing
Missing value treatment - Visualize the processed data - Train-test split

Model Building - Original Data
Build forecasting models - Linear regression - Simple Average - Moving Average - Exponential Models (Single, Double, Triple) - Check the performance of the models built

Check for Stationarity
Check for stationarity - Make the data stationary (if needed)

Model Building - Stationary Data
Generate ACF & PACF Plot and find the AR, MA values. - Build different ARIMA models - Auto ARIMA - Manual ARIMA - Build different SARIMA models - Auto SARIMA - Manual SARIMA - Check the performance of the models built

Compare the performance of the models
Compare the performance of all the models built - Choose the best model with proper rationale - Rebuild the best model using the entire data - Make a forecast for the next 12 months

TIME SERIES FORECASTING ON SPARKLING DATA SET

The top five rows of the data set after making the Yearmonth column as index is as follows:

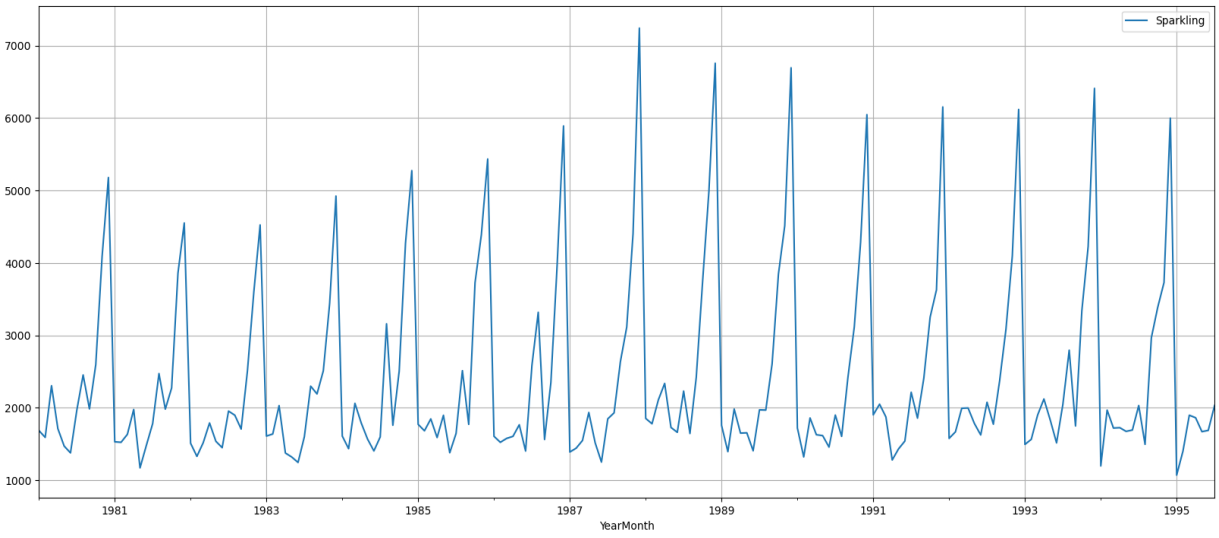
YearMonth	Sparkling
1980-01-01	1686
1980-02-01	1591
1980-03-01	2304
1980-04-01	1712
1980-05-01	1471

The last five rows is :

YearMonth	Sparkling
1995-03-01	1897
1995-04-01	1862
1995-05-01	1670
1995-06-01	1688
1995-07-01	2031

- The data set contains 187 rows and 1 column.

The graphical representation of the ‘Sparkling’ dataset is as follows:



Now we separate the year and month columns from the Yearmonth column, the first few rows of the dataset now looks like

YearMonth	Sparkling	Year	Month
1980-01-01	1686	1980	1
1980-02-01	1591	1980	2
1980-03-01	2304	1980	3
1980-04-01	1712	1980	4
1980-05-01	1471	1980	5

Renaming the Sparkling as Sales, the first five rows of the data set looks like

YearMonth	Sales	Year	Month
1980-01-01	1686	1980	1
1980-02-01	1591	1980	2
1980-03-01	2304	1980	3
1980-04-01	1712	1980	4
1980-05-01	1471	1980	5

The last five rows are as follows:

YearMonth	Sales	Year	Month
1995-03-01	1897	1995	3
1995-04-01	1862	1995	4
1995-05-01	1670	1995	5
1995-06-01	1688	1995	6
1995-07-01	2031	1995	7

- Now we have 187 rows and 3 columns
- The info of the data set

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 187 entries, 1980-01-01 to 1995-07-01
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Sales   187 non-null    int64
 1   Year    187 non-null    int32
 2   Month   187 non-null    int32
dtypes: int32(2), int64(1)
memory usage: 4.4 KB
```

- All the columns have integer data type

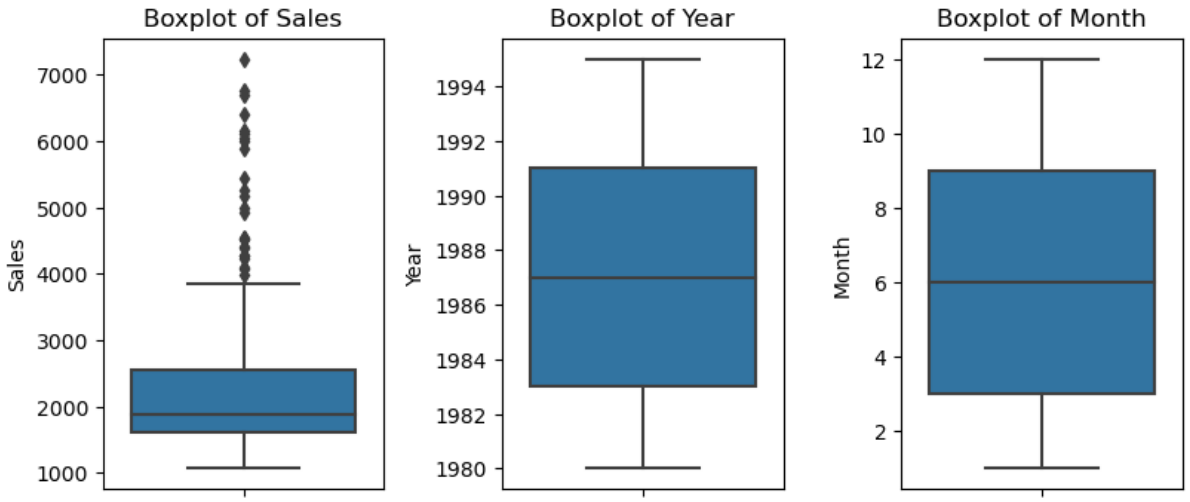
The data summary looka like:

	count	mean	std	min	25%	50%	75%	max
Sales	187.0	2402.0	1295.0	1070.0	1605.0	1874.0	2549.0	7242.0
Year	187.0	1987.0	5.0	1980.0	1983.0	1987.0	1991.0	1995.0
Month	187.0	6.0	3.0	1.0	3.0	6.0	9.0	12.0

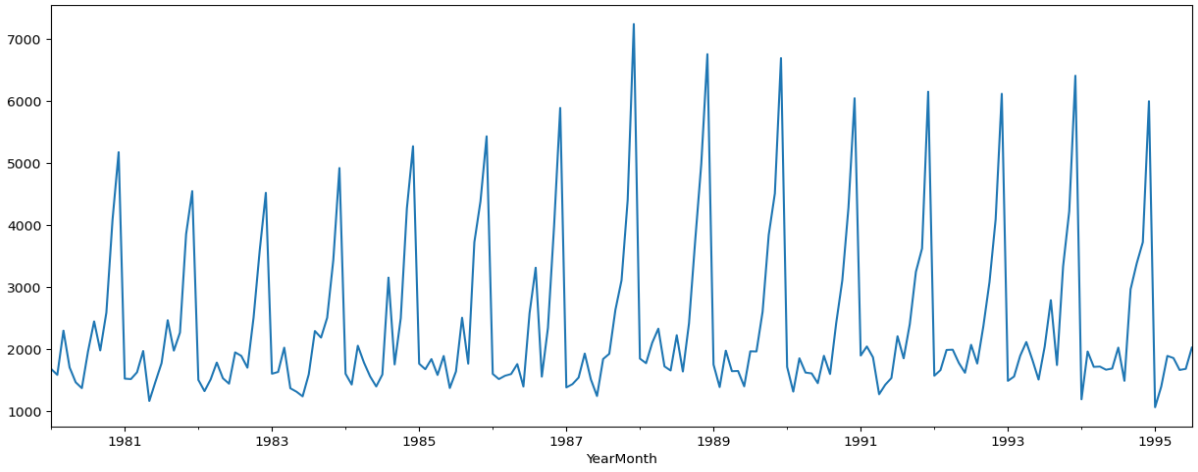
- The average sales of Sparkling wine is 2402
- The minimum sales of Sparkling wine is 1070
- The maximum sales of Sparkling wine is 7242

There are no null values present.

The boxplots of the features present is as follows

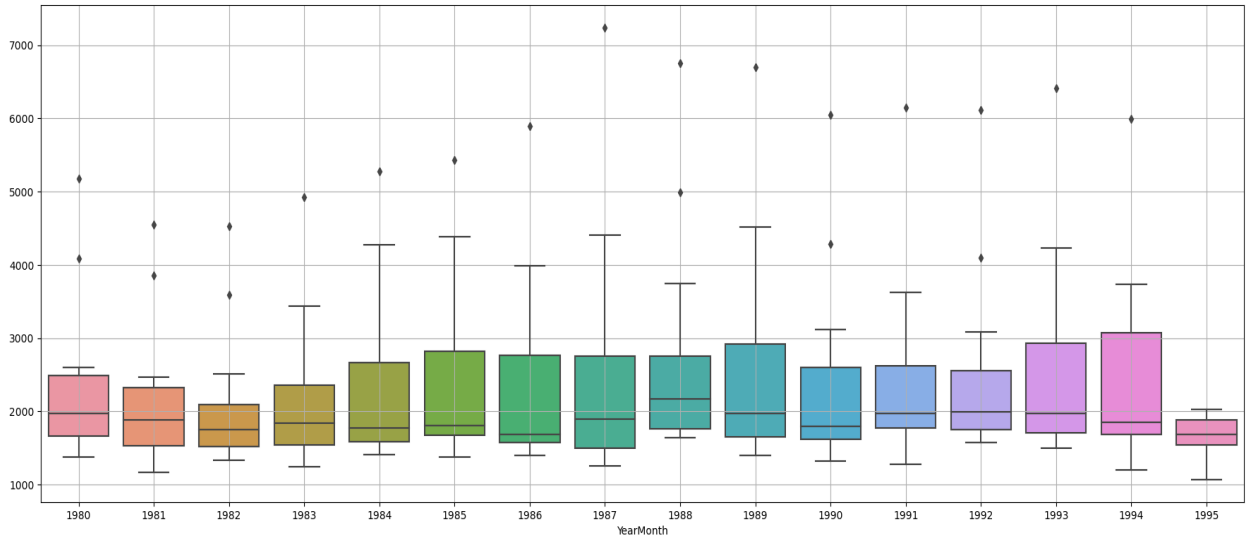


The sales plot is as follows:



The yearly boxplot is :

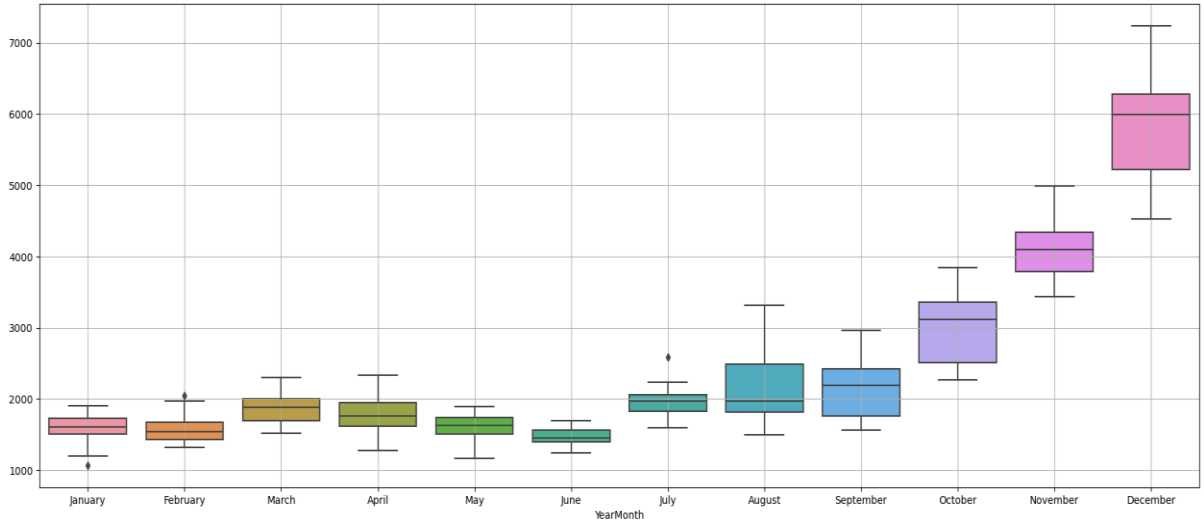
Now, let us plot a box and whisker (1.5* IQR) plot to understand the spread of the data and check for outliers in each year, if any-



As we got to know from the Time Series plot, the box plots over here also indicates a measure of trend being present. Also, we see that Sales of Sparkling Wine has some outliers for certain years.

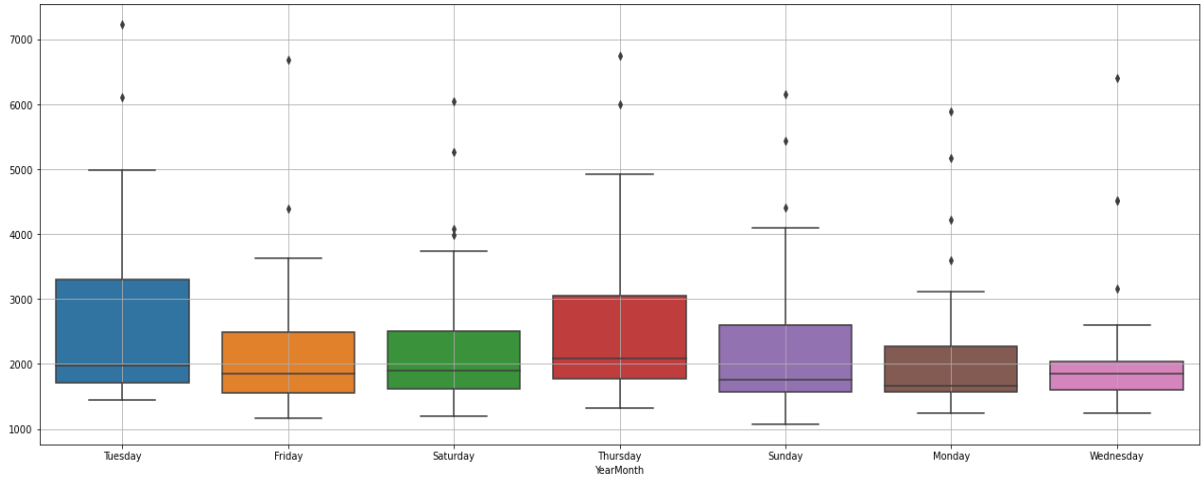
The monthly boxplot is :

Since this is a monthly data, let us plot a box and whisker (1.5* IQR) plot to understand the spread of the data and check for outliers for every month across all the years, if any.

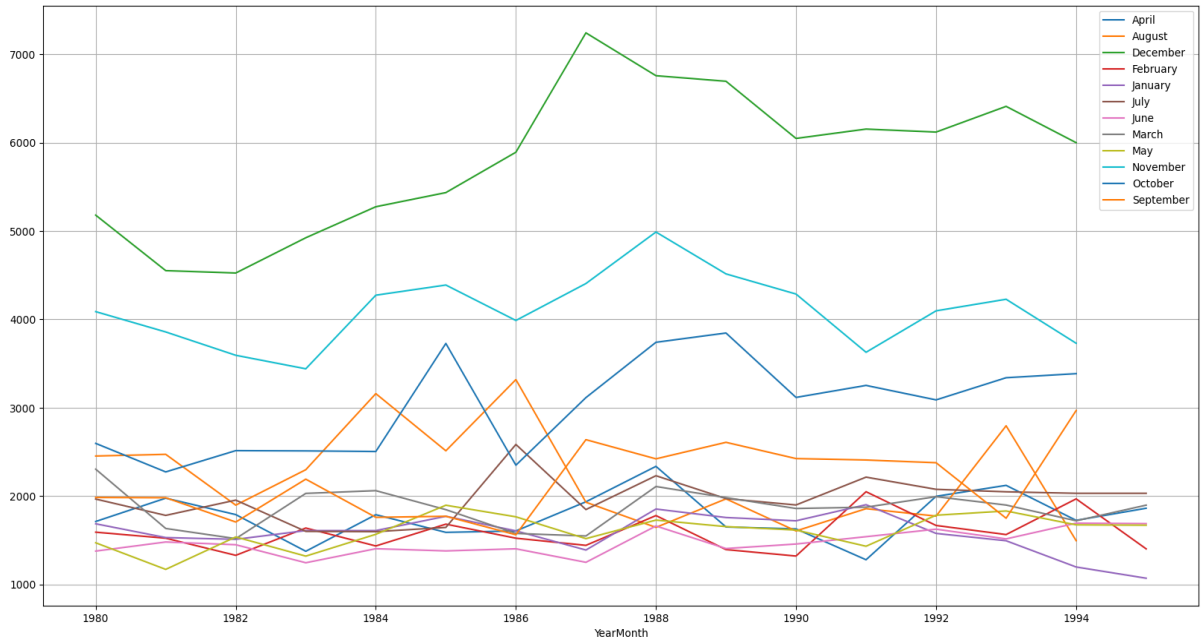


The highest such numbers are being recorded in the month of December across various years

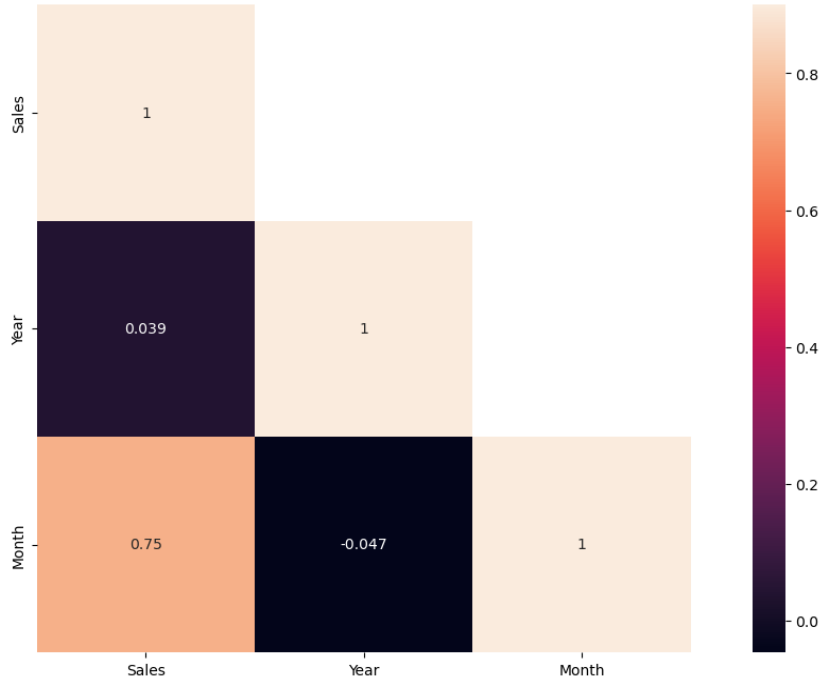
The weekdays boxplot looks like



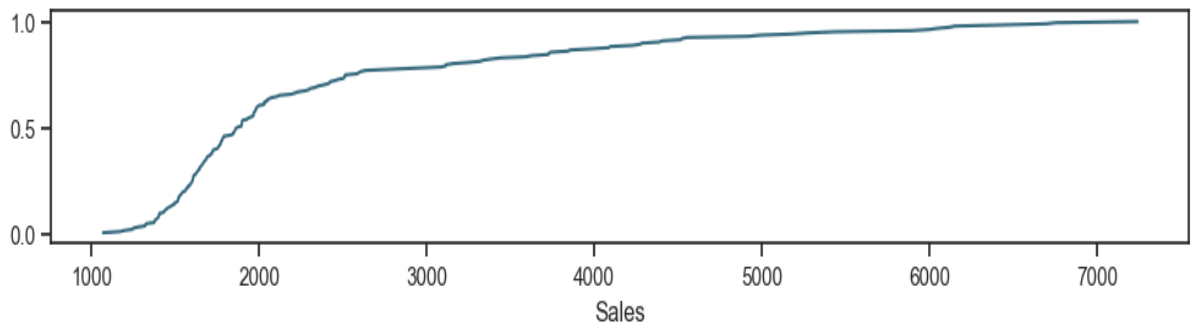
The graph of monthly sales across years is



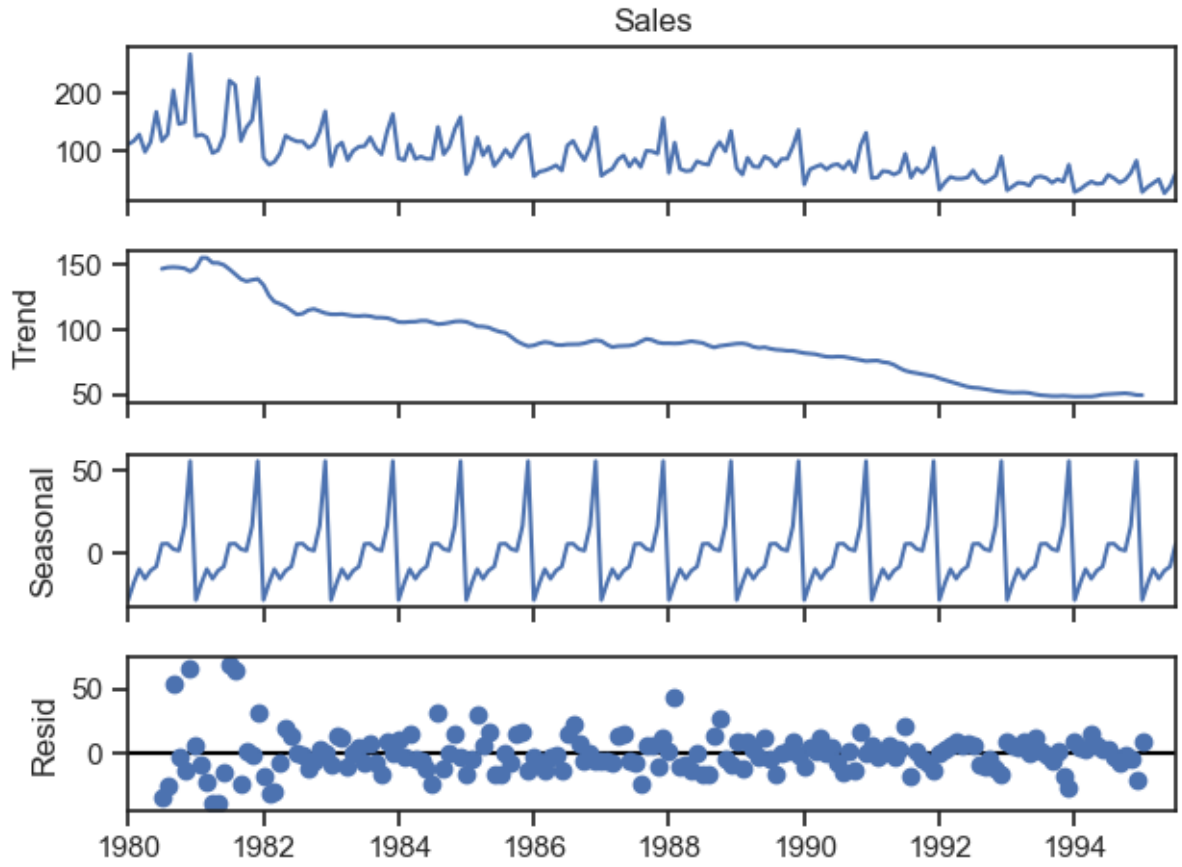
The correlation plot is as follows



The plot of empirical cummulative distribution function



The decomposition of the sales into trend, seasonality and residuals in additive mode



- We see that the residuals are located around 0 from the plot of the residuals in the decomposition.
- Also there is a trend which keeps on changing.
- Also there are no outliers in the dataset.

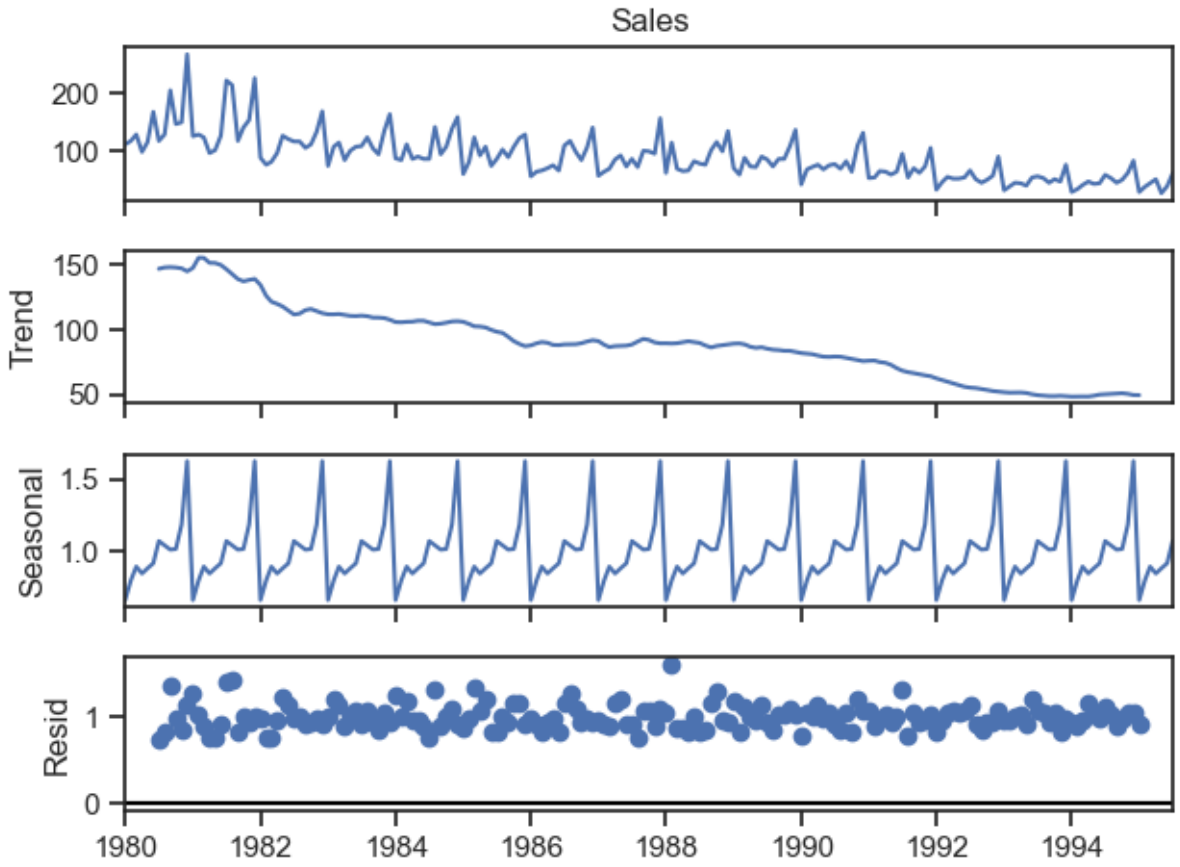
The values after decompostion are

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

Seasonality
 YearMonth
1980-01-01    -854.260599
1980-02-01    -830.350678
1980-03-01    -592.356630
1980-04-01    -658.490559
1980-05-01    -824.416154
1980-06-01    -967.434011
1980-07-01    -465.502265
1980-08-01    -214.332821
1980-09-01    -254.677265
1980-10-01     599.769957
1980-11-01    1675.067179
1980-12-01    3386.983846
Name: seasonal, dtype: float64

Residual
 YearMonth
1980-01-01           NaN
1980-02-01           NaN
1980-03-01           NaN
1980-04-01           NaN
1980-05-01           NaN
1980-06-01           NaN
1980-07-01     70.835599
1980-08-01    315.999487
1980-09-01    -81.864401
1980-10-01   -307.353290
1980-11-01    109.891154
1980-12-01   -501.775513
Name: resid, dtype: float64
```

The decomposition of sales into trend, seasonality and residuals in multiplicative mode



- As per the 'additive' decomposition, we see that there is a pronounced trend in the earlier years of the data. There is seasonality as well.
- Also there are no outliers in the dataset.
- The trend keeps on changing.

The values are

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

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

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

Now we split the data into train and test series, In time series forecasting the splitting of train and test data set is not random .

After splitting at 1991, the shape of the train and test data is

```
Shape of datasets:
train dataset:  (132, 3)
test dataset:  (55, 3)
```

The top five rows and bottom five rows of the train and test data set is as follows :

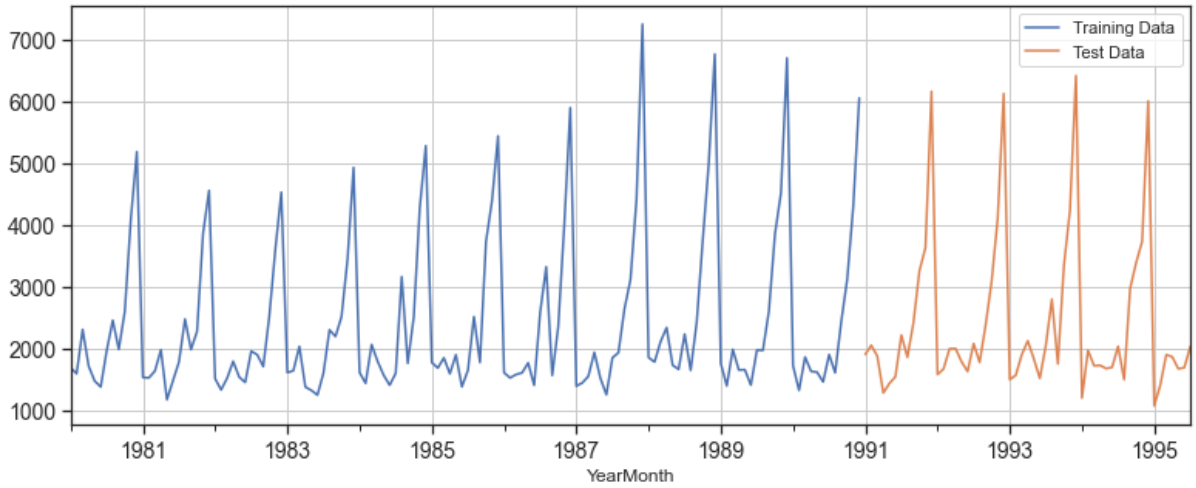
```
Rows of dataset:
First few rows of Training Data
             Sales  Year  Month
YearMonth
1980-01-01   1686  1980      1
1980-02-01   1591  1980      2
1980-03-01   2304  1980      3
1980-04-01   1712  1980      4
1980-05-01   1471  1980      5

Last few rows of Training Data
             Sales  Year  Month
YearMonth
1990-08-01   1605  1990      8
1990-09-01   2424  1990      9
1990-10-01   3116  1990     10
1990-11-01   4286  1990     11
1990-12-01   6047  1990     12

First few rows of Test Data
             Sales  Year  Month
YearMonth
1991-01-01   1902  1991      1
1991-02-01   2049  1991      2
1991-03-01   1874  1991      3
1991-04-01   1279  1991      4
1991-05-01   1432  1991      5

Last few rows of Test Data
             Sales  Year  Month
YearMonth
1995-03-01   1897  1995      3
1995-04-01   1862  1995      4
1995-05-01   1670  1995      5
1995-06-01   1688  1995      6
1995-07-01   2031  1995      7
```

The graphical plot representing the train and test data set is as follows :



It is difficult to predict the future observations if such an instance have not happened in the past. From our train-test split we are predicting likewise behavior as compared to the past years.

LINEAR REGRESSION

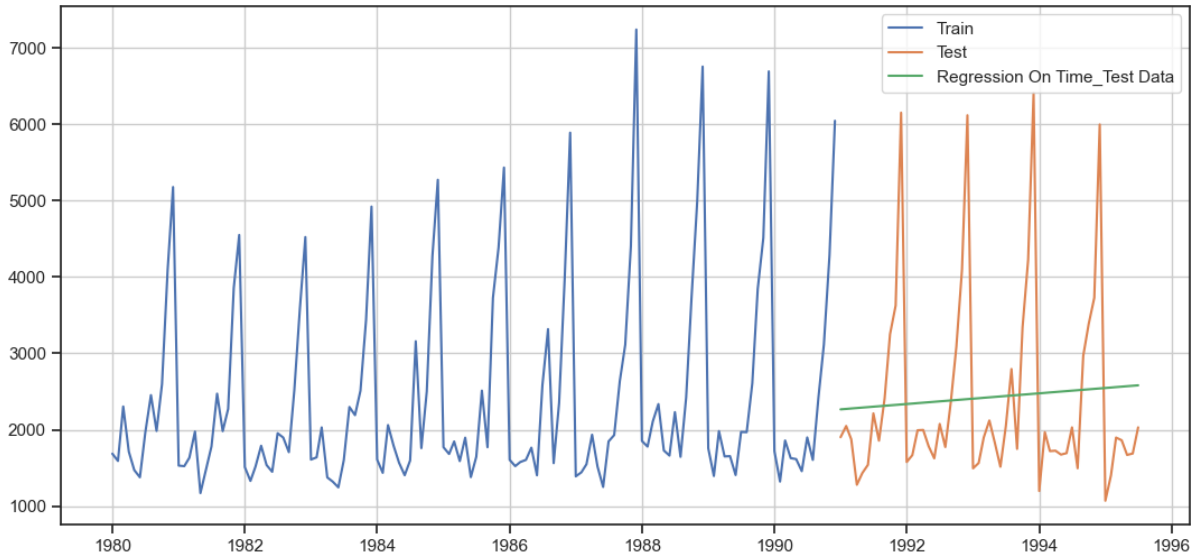
For this particular linear regression, we are going to regress the 'Sales' variable against the order of the occurrence.

Train and test time instance is as follows:

```
Training Time instance
 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 3
3, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63,
64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94,
95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 12
0, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132]
Test Time instance
 [43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 7
3, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97]
```

Now that our training and test data has been modified, let us go ahead use $LinearRegression$ to build the model on the training data and test the model on the test data

After applying the linear regression, the forecasted plot looks like



The RMSE value on applying on test data is

	Test RMSE
Linear Regression	1275.867052

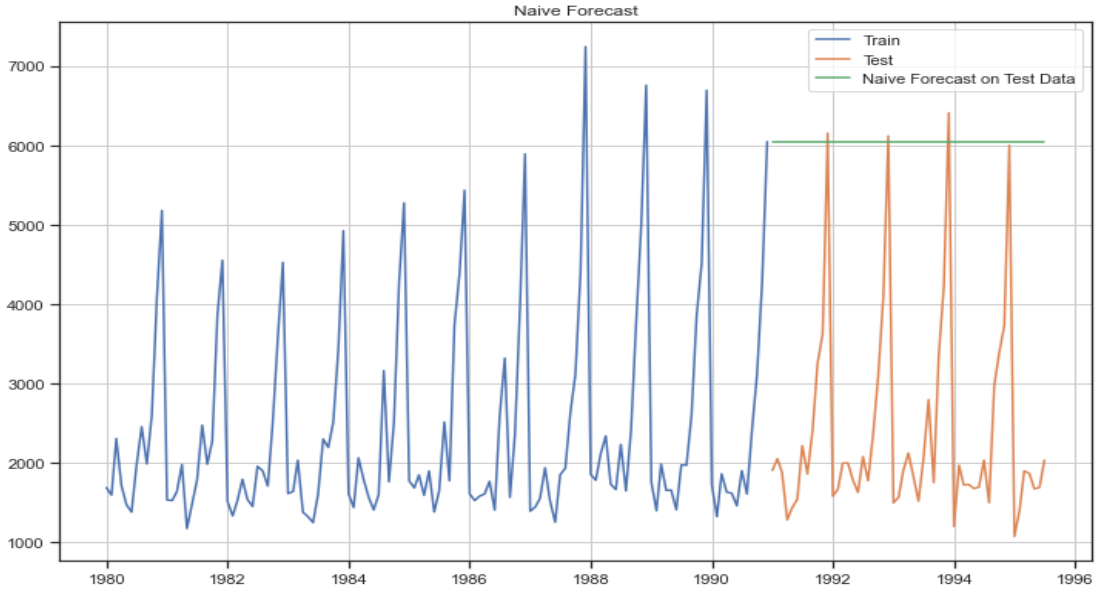
NAÏVE APPROACH

For this particular naive model, we say that the prediction for tomorrow is the same as today and the prediction for day after tomorrow is tomorrow and since the prediction of tomorrow is same as today, therefore the prediction for day after tomorrow is also today.

After applying the naïve approach, we get

```
YearMonth
1991-01-01    6047
1991-02-01    6047
1991-03-01    6047
1991-04-01    6047
1991-05-01    6047
Name: naive, dtype: int64
```

After applying the naïve approach, the forecasted plot looks like



The RMSE values are

	Test RMSE
Linear Regression	1275.867052
Naive Model	3864.279352

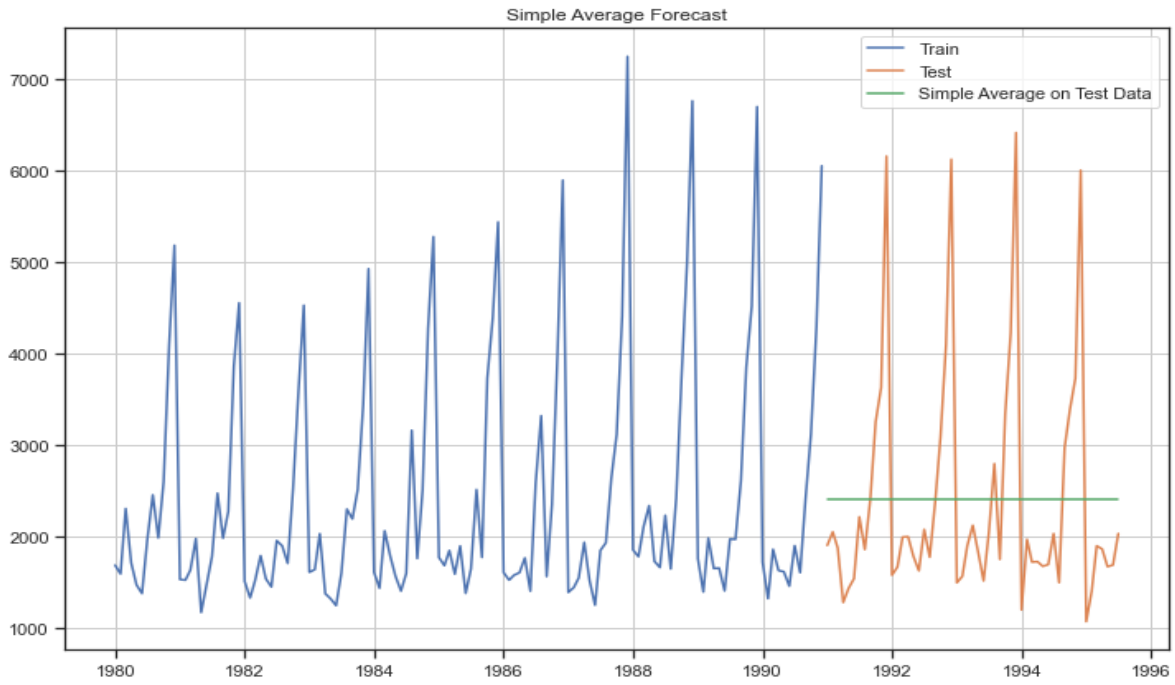
SIMPLE AVERAGE

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

After applying the simple average method, the forecasted mean is

YearMonth	Sales	Year	Month	mean_forecast
1991-01-01	1902	1991	1	2403.780303
1991-02-01	2049	1991	2	2403.780303
1991-03-01	1874	1991	3	2403.780303
1991-04-01	1279	1991	4	2403.780303
1991-05-01	1432	1991	5	2403.780303

The forecasted plot is



The RMSE value on applying on test data

	Test RMSE
Linear Regression	1275.867052
Naive Model	3864.279352
Simple Average Model	1275.081804

MOVING AVERAGE

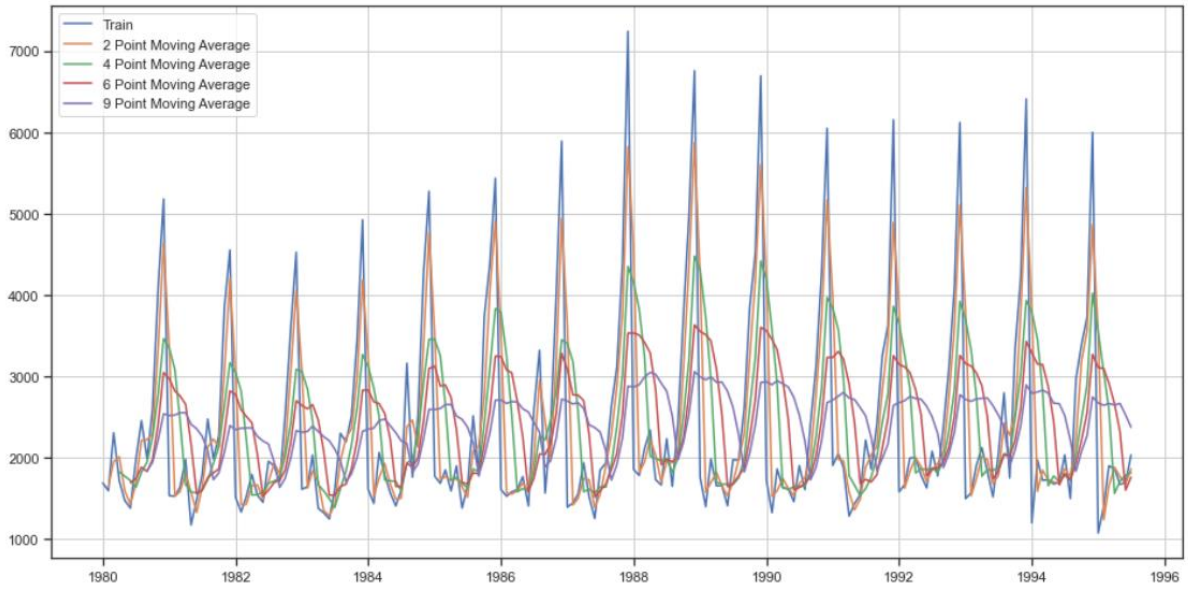
For the moving average model, we are going to calculate rolling means (or moving averages) for different intervals. The best interval can be determined by the maximum accuracy (or the minimum error) over here.

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

The moving average is calculated for different rolling values, It is as follows

YearMonth	Sales	Year	Month	Trailing_2	Trailing_4	Trailing_6	Trailing_9
1980-01-01	1686	1980	1	NaN	NaN	NaN	NaN
1980-02-01	1591	1980	2	1638.5	NaN	NaN	NaN
1980-03-01	2304	1980	3	1947.5	NaN	NaN	NaN
1980-04-01	1712	1980	4	2008.0	1823.25	NaN	NaN
1980-05-01	1471	1980	5	1591.5	1769.50	NaN	NaN

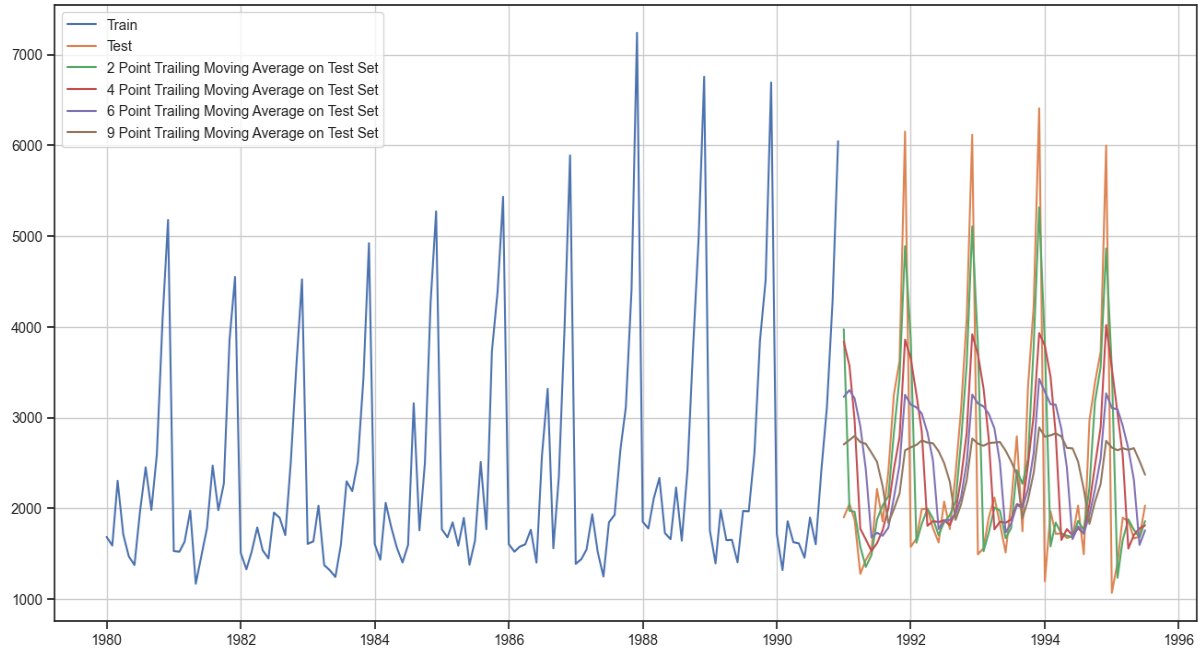
The forecasted plot looks like



The shape of the trainling moving average train and test data is

```
(55, 7)
(132, 7)
```

Before we go on to build the various Exponential Smoothing models, let us plot all the models and compare the Time Series plots.



The RMSE values are as follows :

	Test RMSE
Linear Regression	1275.867052
Naive Model	3864.279352
Simple Average Model	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315

SIMPLE EXPONENTIAL SMOOTHING

The parameters are

```
{'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}
```

The forecasted values are

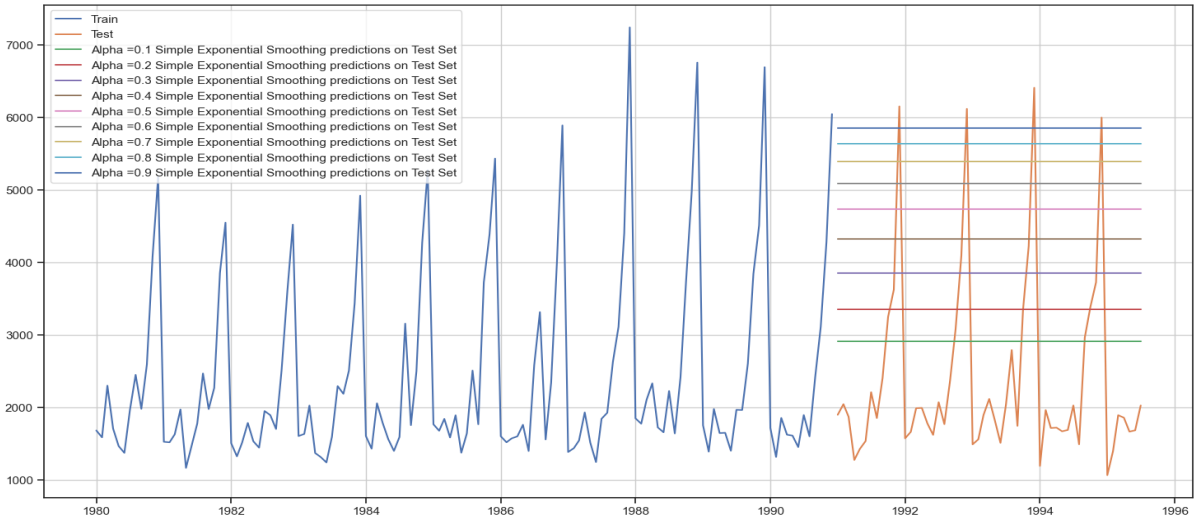
YearMonth	Sales	Year	Month	predict
1991-01-01	1902	1991	1	2676.676366
1991-02-01	2049	1991	2	2676.676366
1991-03-01	1874	1991	3	2676.676366
1991-04-01	1279	1991	4	2676.676366
1991-05-01	1432	1991	5	2676.676366

Setting different alpha values.

Remember, the higher the alpha value more weightage is given to the more recent observation. That means, what happened recently will happen again.

We will run a loop with different alpha values to understand which particular value works best for alpha on the test set

Now for different values of alpha we plot the forecasted plot



The RMSE values are

	Alpha Values	Train RMSE	Test RMSE
0	0.1	1333.873836	1375.393398
1	0.2	1356.042987	1595.206839
2	0.3	1359.511747	1935.507132
3	0.4	1352.588879	2311.919615
4	0.5	1344.004369	2666.351413
5	0.6	1338.805381	2979.204388
6	0.7	1338.844308	3249.944092
7	0.8	1344.462091	3483.801006
8	0.9	1355.723518	3686.794285

Note that we have low RMSE value for when alpha=0.1

The RMSE values till now for different methods is

	Test RMSE
Linear Regression	1275.867052
Naive Model	3864.279352
Simple Average Model	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315
Alpha=0.1,SimpleExponentialSmoothing	1375.393398

DOUBLE EXPONENTIAL SMOOTHING (HOLT'S MODEL)

Two parameters α and β are estimated in this model. Level and Trend are accounted for in this model.

The parameters are

```
{'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}
```

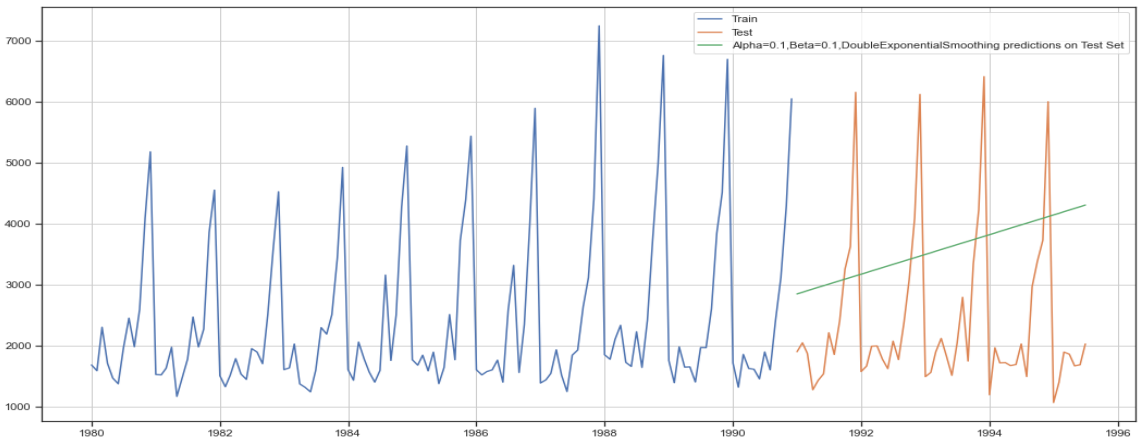
The predicted values are as follows

YearMonth	Sales	Year	Month	predict
1991-01-01	1902	1991	1	5221.278699
1991-02-01	2049	1991	2	5127.886554
1991-03-01	1874	1991	3	5034.494409
1991-04-01	1279	1991	4	4941.102264
1991-05-01	1432	1991	5	4847.710119

For different values of aplha and beta, we fing the RMSE values and sort them to find the low RMSE value.

	Alpha Values	Beta Values	Train RMSE	Test RMSE
0	0.1	0.1	1382.520870	1778.564670
1	0.1	0.2	1413.598835	2599.439986
2	0.1	0.3	1445.762015	4293.084674
3	0.1	0.4	1480.897776	6039.537339
4	0.1	0.5	1521.108657	7390.522201
...	...	...	...	...
95	1.0	0.6	1753.402326	49327.087977
96	1.0	0.7	1825.187155	52655.765663
97	1.0	0.8	1902.013709	55442.273880
98	1.0	0.9	1985.368445	57823.177011
99	1.0	1.0	2077.672157	59877.076519

The forecasted plot is as follows:



The RMSE values till now for different methods is

	Test RMSE
Linear Regression	1275.867052
Naive Model	3864.279352
Simple Average Model	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315
Alpha=0.1,SimpleExponentialSmoothing	1375.393398
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	1778.564670

TRIPLE EXPONENTIAL SMOOTHING (HOLT - WINTER'S MODEL)

Three parameters , β and γ are estimated in this model. Level, Trend and Seasonality are accounted for in this model.

The parameters for when trend and seasonality being additive, additive

```
{'smoothing_level': 0.11127218025549082,
 'smoothing_trend': 0.01236078344884856,
 'smoothing_seasonal': 0.4607177625005678,
 'damping_trend': nan,
 'initial_level': 2356.5781955695325,
 'initial_trend': -0.018629382309754305,
 'initial_seasons': array([-636.2336622 , -722.98363397, -398.64361027, -473.43090136,
        -808.42531193, -815.35036274, -384.23072634,   72.99502455,
        -237.44284242,  272.32573256, 1541.37815463, 2590.07742631]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

The parameters for when trend and seasonality being additive, multiplicative

```
{'smoothing_level': 0.11101220760873363,
 'smoothing_trend': 0.04931229141613127,
 'smoothing_seasonal': 0.3624630075688972,
 'damping_trend': nan,
 'initial_level': 2356.4944083975206,
 'initial_trend': -9.84005678473666,
 'initial_seasons': array([0.71400191, 0.68242746, 0.90425704, 0.80517582, 0.65565206,
        0.65356211, 0.88613212, 1.13347123, 0.91881202, 1.21182696,
        1.87089064, 2.37424138]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

The parameters for when trend and seasonality being multiplicative, multiplicative

```
{'smoothing_level': 0.11105950016225202,
 'smoothing_trend': 0.04935435931484549,
 'smoothing_seasonal': 0.3622188768302799,
 'damping_trend': nan,
 'initial_level': 2357.161176620859,
 'initial_trend': 0.999697325206386,
 'initial_seasons': array([0.73111105, 0.69874416, 0.90049926, 0.81021521, 0.66840549,
       0.6691344 , 0.87891607, 1.11663153, 0.91684784, 1.17900619,
       1.82762886, 2.31595687]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

The parameters for when trend and seasonality being multiplicative, additive

```
{'smoothing_level': 0.11107142866032142,
 'smoothing_trend': 0.012341269861019839,
 'smoothing_seasonal': 0.4609259258337408,
 'damping_trend': nan,
 'initial_level': 2356.5416666887654,
 'initial_trend': 1.003171553167385,
 'initial_seasons': array([-636.18663196, -722.94704863, -398.70746525, -473.40538195,
       -808.38454863, -815.31163196, -384.22829861,   72.94878474,
       -237.3949653 ,  272.33420139, 1541.2821181 , 2590.00086809]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

The forecasted values of all the above parameters

YearMonth	Sales	Year	Month	predict_ta_sa	predict_ta_sm	predict_tm_sm	predict_tm_sa
1991-01-01	1902	1991	1	1509.969093	1586.782642	1591.473305	1511.756579
1991-02-01	2049	1991	2	1205.343244	1355.896477	1360.492250	1227.680658
1991-03-01	1874	1991	3	1702.386113	1762.095344	1768.009110	1713.866726
1991-04-01	1279	1991	4	1548.514691	1655.471900	1661.678682	1578.252255
1991-05-01	1432	1991	5	1467.824074	1541.320914	1547.447439	1490.255624

The RMSE values after applying smoothing

	Test RMSE
Linear Regression	1275.867052
Naive Model	3864.279352
Simple Average Model	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315
Alpha=0.1,SimpleExponentialSmoothing	1375.393398
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	1778.564670
Alpha=0.08621,Beta=1.3722,Gamma=0.4763,TrippleExponentialSmoothing_Auto_Fit	1304.927405

For different values of alpha, beta, gamma we forecast the values, the top five rows of the data looks like

YearMonth	Sales	Year	Month	(predict_ta_sa, 0.1, 0.1, 0.1)	(predict_ta_sa, 0.1, 0.1, 0.2)	(predict_ta_sa, 0.1, 0.1, 0.30000000000000004)	(predict_ta_sa, 0.1, 0.1, 0.4)	(predict_ta_sa, 0.1, 0.1, 0.5)	(predict_ta_sa, 0.1, 0.1, 0.6)
1991-01-01	1902	1991	1	1671.894991	1540.529588	1472.827405	1444.947521	1440.100315	1446.456719
1991-02-01	2049	1991	2	1535.938082	1354.094081	1236.723426	1163.127303	1118.381068	1091.681321
1991-03-01	1874	1991	3	1882.992874	1728.658127	1644.294990	1605.772780	1593.658780	1593.602194
1991-04-01	1279	1991	4	1798.243923	1638.281580	1535.922824	1469.062420	1424.230588	1393.229741
1991-05-01	1432	1991	5	1576.572747	1470.697707	1394.544409	1347.223962	1324.218679	1318.006765

5 rows × 3461 columns

After sorting through RMSE values

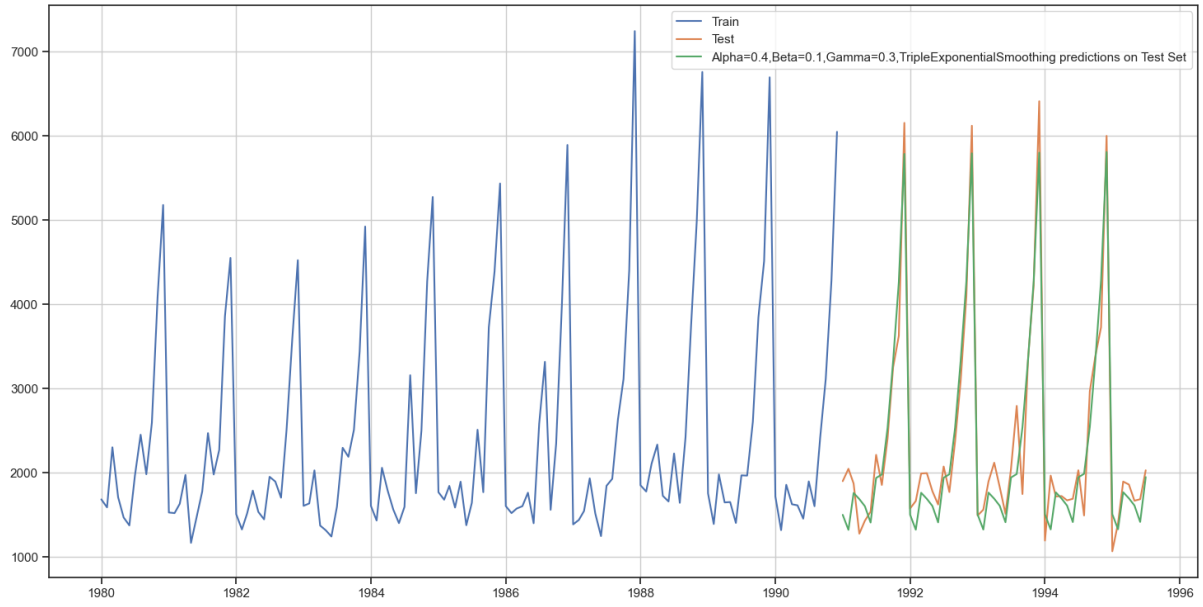
	Alpha Values	Beta Values	Gamma Values	Train RMSE	Test RMSE	Method
1301	0.4	0.1	0.2	384.467709	317.434302	ta_sm
2245	0.4	0.1	0.3	381.106645	326.579641	tm_sm
1211	0.3	0.2	0.2	388.544148	329.037543	ta_sm
1200	0.3	0.1	0.1	388.220071	337.080969	ta_sm
1110	0.2	0.2	0.1	398.482510	340.186457	ta_sm

The RMSE table now looks like

	Test RMSE
Linear Regression	1275.867052
Naive Model	3864.279352
Simple Average Model	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315
Alpha=0.1,SimpleExponentialSmoothing	1375.393398
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	1778.564670
Alpha=0.08621,Beta=1.3722,Gamma=0.4763,TrippleExponentialSmoothing_Auto_Fit	1304.927405
Alpha=0.4,Beta=0.1,Gamma=0.2,TripleExponentialSmoothing	317.434302

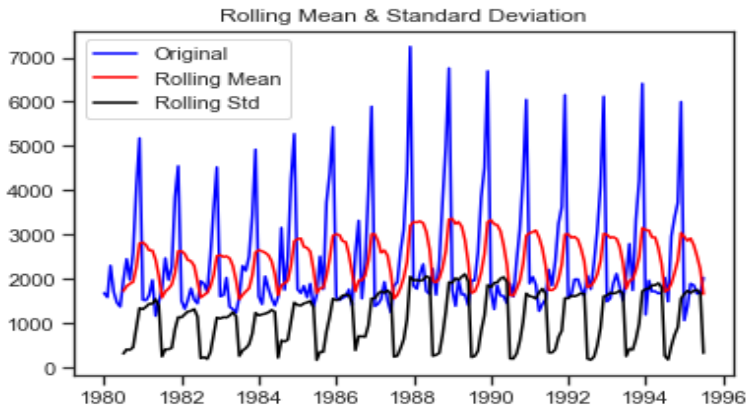
- Note that the RMSE value obtained is much lesser than the RMSE of the other methods

The forecasted plot looks like



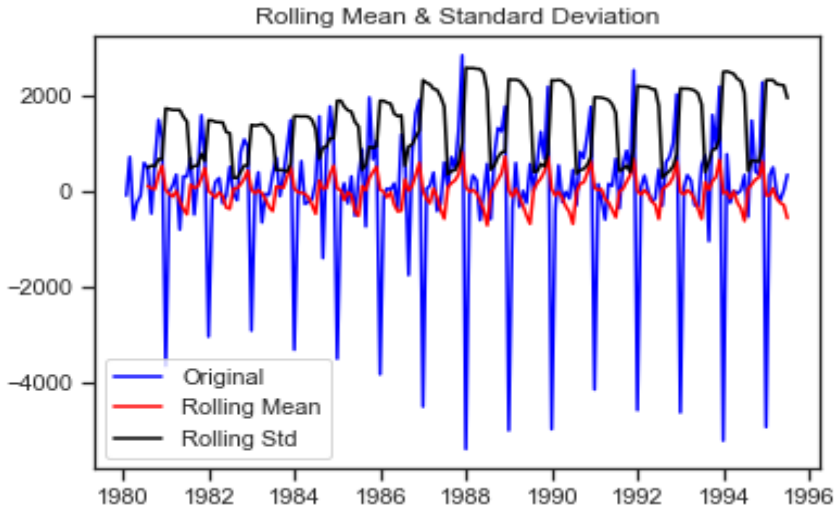
Now we check for stationarity at alpha=0.05

The plot for rolling mean and standard deviation is as follows




```
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
```

- The p value is 0.6 which is greater than 0.05
- So we accept the null hypothesis i.e the plot is not stationary
- The above plot is not stationary, so we have to make it stationary by differencing
- After differencing the plot looks like




```
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
```

BUILDING OF ARIMA MODEL

Some parameter combinations for the model are as follows

```
Model: (0, 1, 1)
Model: (0, 1, 2)
Model: (0, 1, 3)
Model: (1, 1, 0)
Model: (1, 1, 1)
Model: (1, 1, 2)
Model: (1, 1, 3)
Model: (2, 1, 0)
Model: (2, 1, 1)
Model: (2, 1, 2)
Model: (2, 1, 3)
Model: (3, 1, 0)
Model: (3, 1, 1)
Model: (3, 1, 2)
Model: (3, 1, 3)
```

Now we get the Akaike Information Criteria (AIC) for all the above combinations of the models

```
ARIMA(0, 1, 0) - AIC:2267.6630357855465
ARIMA(0, 1, 1) - AIC:2263.060015591336
ARIMA(0, 1, 2) - AIC:2234.4083231278064
ARIMA(0, 1, 3) - AIC:2233.9948577600953
ARIMA(1, 1, 0) - AIC:2266.6085393190087
ARIMA(1, 1, 1) - AIC:2235.7550946736415
ARIMA(1, 1, 2) - AIC:2234.5272004521285
ARIMA(1, 1, 3) - AIC:2235.607812292888
ARIMA(2, 1, 0) - AIC:2260.365743968097
ARIMA(2, 1, 1) - AIC:2233.7776262364514
ARIMA(2, 1, 2) - AIC:2213.5092170036614
```

```
C:\Users\THANUSRI\anaconda31\Lib\site-packages\statsmodels\base\model.py:607: ConvergenceWarning: Maximum Likelihood optimizati
on failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

```
ARIMA(2, 1, 3) - AIC:2232.982762467654
ARIMA(3, 1, 0) - AIC:2257.72337899794
ARIMA(3, 1, 1) - AIC:2235.4989865071907
```

```
C:\Users\THANUSRI\anaconda31\Lib\site-packages\statsmodels\base\model.py:607: ConvergenceWarning: Maximum Likelihood optimizati
on failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

```
ARIMA(3, 1, 2) - AIC:2230.792548127145
ARIMA(3, 1, 3) - AIC:2221.456643202785
```

```
C:\Users\THANUSRI\anaconda31\Lib\site-packages\statsmodels\base\model.py:607: ConvergenceWarning: Maximum Likelihood optimizati
on failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

now we sort in decreasing order of AIC values

	param	AIC
10	(2, 1, 2)	2213.509217
15	(3, 1, 3)	2221.456643
14	(3, 1, 2)	2230.792548
11	(2, 1, 3)	2232.982762
9	(2, 1, 1)	2233.777626
3	(0, 1, 3)	2233.994858
2	(0, 1, 2)	2234.408323
6	(1, 1, 2)	2234.527200
13	(3, 1, 1)	2235.498987
7	(1, 1, 3)	2235.607812
5	(1, 1, 1)	2235.755095
12	(3, 1, 0)	2257.723379
8	(2, 1, 0)	2260.365744
1	(0, 1, 1)	2263.060016
4	(1, 1, 0)	2266.608539
0	(0, 1, 0)	2267.663036

Arima summary for the first model i.e (2,1,2) is

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Sales   No. Observations:                  132
Model:                 ARIMA(2, 1, 2)   Log Likelihood               -1101.755
Date:                Thu, 11 Apr 2024   AIC                           2213.509
Time:                        21:20:00   BIC                           2227.885
Sample:                    01-01-1980   HQIC                          2219.351
                         - 12-01-1990
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          1.3121      0.046     28.786      0.000       1.223       1.401
ar.L2         -0.5593      0.072     -7.731      0.000      -0.701      -0.417
ma.L1         -1.9916      0.110    -18.184      0.000      -2.206      -1.777
ma.L2          0.9999      0.110      9.093      0.000       0.784       1.215
sigma2      1.099e+06      2e-07   5.49e+12      0.000     1.1e+06     1.1e+06
===================================================================================
Ljung-Box (L1) (Q):                   0.19   Jarque-Bera (JB):                14.46
Prob(Q):                              0.67   Prob(JB):                         0.00
Heteroskedasticity (H):               2.43   Skew:                             0.61
Prob(H) (two-sided):                  0.00   Kurtosis:                         4.08
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 2.78e+28. Standard errors may be unstable.
```

Now we evaluate this model on test data using RMSE

	Test RMSE
Linear Regression	1275.867052
Naive Model	3864.279352
Simple Average Model	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315
Alpha=0.1,SimpleExponentialSmoothing	1375.393398
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	1778.564670
Alpha=0.08621,Beta=1.3722,Gamma=0.4763,TrippleExponentialSmoothing_Auto_Fit	1304.927405
Alpha=0.4,Beta=0.1,Gamma=0.2,TripleExponentialSmoothing	317.434302
Auto_ARIMA	1299.979692

BUILDING OF SARIMA MODEL

Some parameter combinations for the model are as follows

```
Examples of some parameter combinations for Model...
Model: (0, 1, 1)(0, 0, 1, 12)
Model: (0, 1, 2)(0, 0, 2, 12)
Model: (0, 1, 3)(0, 0, 3, 12)
Model: (1, 1, 0)(1, 0, 0, 12)
Model: (1, 1, 1)(1, 0, 1, 12)
Model: (1, 1, 2)(1, 0, 2, 12)
Model: (1, 1, 3)(1, 0, 3, 12)
Model: (2, 1, 0)(2, 0, 0, 12)
Model: (2, 1, 1)(2, 0, 1, 12)
Model: (2, 1, 2)(2, 0, 2, 12)
Model: (2, 1, 3)(2, 0, 3, 12)
Model: (3, 1, 0)(3, 0, 0, 12)
Model: (3, 1, 1)(3, 0, 1, 12)
Model: (3, 1, 2)(3, 0, 2, 12)
Model: (3, 1, 3)(3, 0, 3, 12)
```

Now we get the Akaike Information Criteria (AIC) for all the above combinations of the models and we sort in decreasing order of AIC values. The top five rows looks like

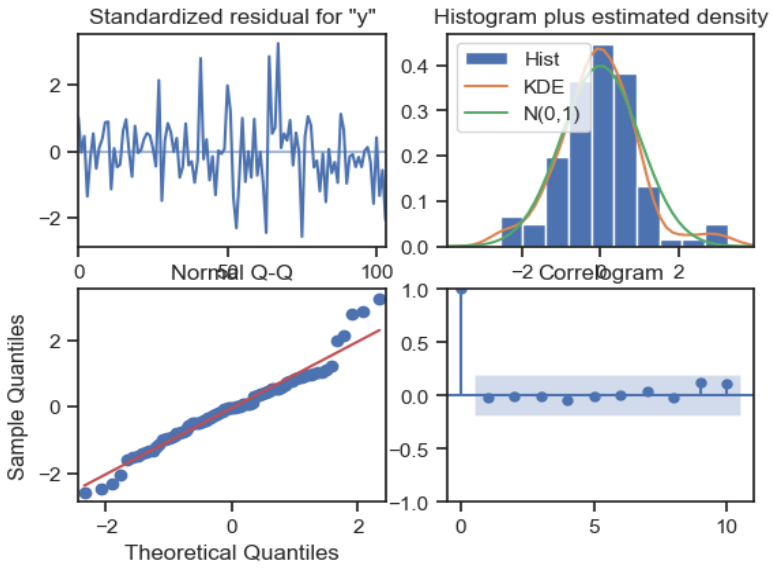
	param	seasonal	AIC
83	(1, 1, 1)	(0, 0, 3, 12)	12.000000
183	(2, 1, 3)	(1, 0, 3, 12)	20.000000
215	(3, 1, 1)	(1, 0, 3, 12)	208.047774
91	(1, 1, 1)	(2, 0, 3, 12)	259.695513
251	(3, 1, 3)	(2, 0, 3, 12)	349.867995

The SARIMAX results is as follows

```
                                SARIMAX Results
==========================================================================================
Dep. Variable:                                 y   No. Observations:                  132
Model:             SARIMAX(1, 1, 2)x(1, 0, 2, 12)   Log Likelihood                -770.792
Date:                            Sat, 08 Jul 2023   AIC                           1555.584
Time:                                    08:47:07   BIC                           1574.095
Sample:                                         0   HQIC                          1563.083
                                              - 132
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.6281      0.255     -2.463      0.014      -1.128      -0.128
ma.L1         -0.1041      0.225     -0.463      0.643      -0.545       0.337
ma.L2         -0.7276      0.154     -4.734      0.000      -1.029      -0.426
ar.S.L12       1.0439      0.014     72.842      0.000       1.016       1.072
ma.S.L12      -0.5551      0.098     -5.663      0.000      -0.747      -0.363
ma.S.L24      -0.1355      0.120     -1.133      0.257      -0.370       0.099
sigma2      1.506e+05   2.03e+04      7.400      0.000    1.11e+05     1.9e+05
===================================================================================
Ljung-Box (L1) (Q):                   0.04   Jarque-Bera (JB):                11.72
Prob(Q):                              0.84   Prob(JB):                         0.00
Heteroskedasticity (H):               1.47   Skew:                             0.36
Prob(H) (two-sided):                  0.26   Kurtosis:                         4.48
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

The plot diagonastics

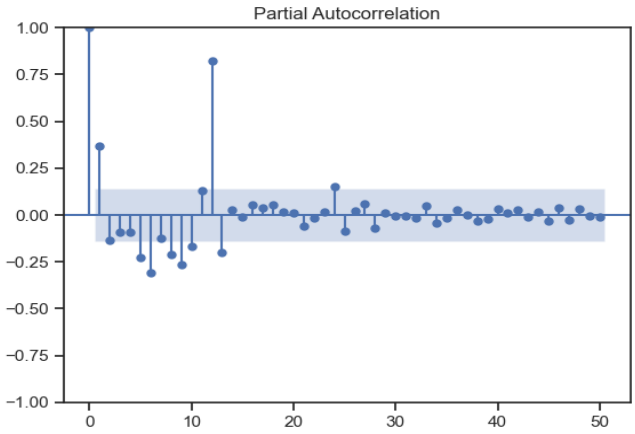


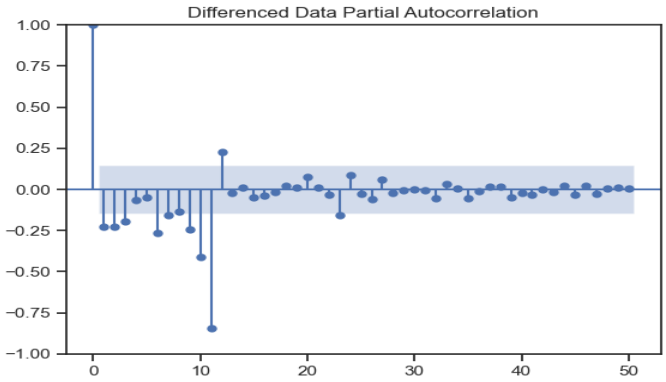
Now we evaluate this model on test data using RMSE

	Test RMSE
Linear Regression	1275.867052
Naive Model	3864.279352
Simple Average Model	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315
Alpha=0.1,SimpleExponentialSmoothing	1375.393398
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	1778.564670
Alpha=0.08621,Beta=1.3722,Gamma=0.4763,TrippleExponentialSmoothing_Auto_Fit	1304.927405
Alpha=0.4,Beta=0.1,Gamma=0.2,TripleExponentialSmoothing	317.434302
Auto_ARIMA	1299.979692
(1,1,1),(2,0,3,12),Auto_SARIMA	528.590608

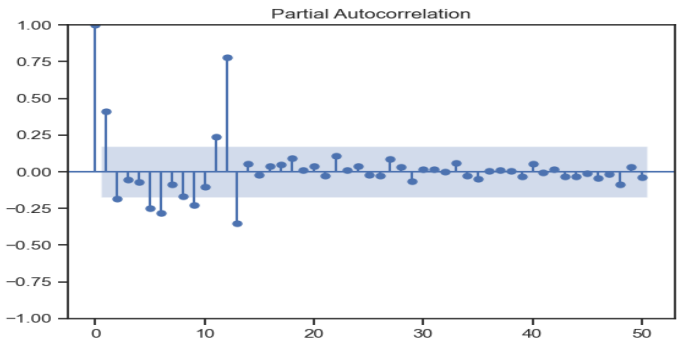
MANUAL SARIMA

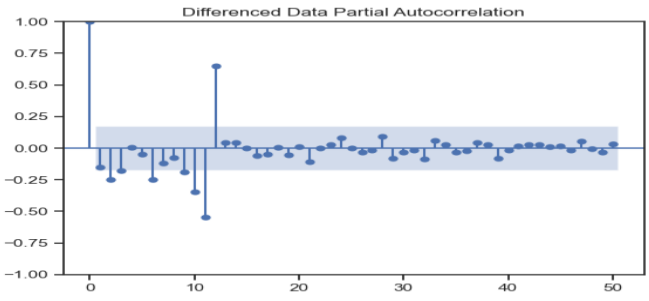
The ACF and PACF plots are as follows





After differencing to convert it from non stationary to stationary





The SARIMAX results for manual arima model

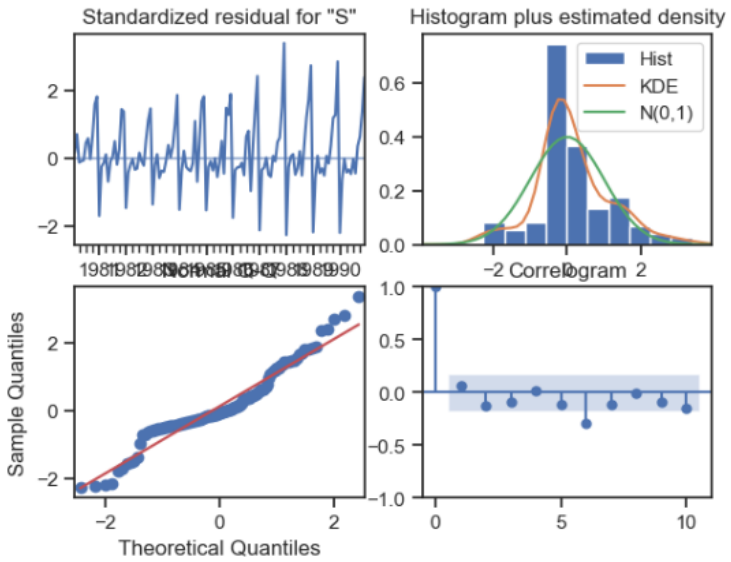
SARIMAX Results

Dep. Variable:	Sales	No. Observations:	132
Model:	ARIMA(1, 1, 1)	Log Likelihood	-1114.878
Date:	Thu, 11 Apr 2024	AIC	2235.755
Time:	21:26:55	BIC	2244.381
Sample:	01-01-1980 - 12-01-1990	HQIC	2239.260
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.4494	0.043	10.366	0.000	0.364	0.534
ma.L1	-0.9996	0.102	-9.811	0.000	-1.199	-0.800
sigma2	1.401e+06	7.57e-08	1.85e+13	0.000	1.4e+06	1.4e+06

Ljung-Box (L1) (Q):	0.50	Jarque-Bera (JB):	10.42
Prob(Q):	0.48	Prob(JB):	0.01
Heteroskedasticity (H):	2.64	Skew:	0.46
Prob(H) (two-sided):	0.00	Kurtosis:	4.03

The plot diagonastics



The RMSE value is

	Test RMSE
Linear Regression	1275.867052
Naive Model	3864.279352
Simple Average Model	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315
Alpha=0.1,SimpleExponentialSmoothing	1375.393398
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	1778.564670
Alpha=0.08621,Beta=1.3722,Gamma=0.4763,TrippleExponentialSmoothing_Auto_Fit	1304.927405
Alpha=0.4,Beta=0.1,Gamma=0.2,TripleExponentialSmoothing	317.434302
Auto_ARIMA	1299.979692
(1,1,1),(2,0,3,12),Auto_SARIMA	528.590608
ARIMA(3,1,3)	1319.936732

MANUAL SARIMA MODEL

The SARIMAX results

SARIMAX Results

Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(1, 1, 1)x(1, 1, 1, 12)	Log Likelihood	-882.088
Date:	Thu, 11 Apr 2024	AIC	1774.175
Time:	21:26:57	BIC	1788.071
Sample:	0	HQIC	1779.818
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.1957	0.104	1.878	0.060	-0.009	0.400
ma.L1	-0.9404	0.053	-17.897	0.000	-1.043	-0.837
ar.S.L12	0.0711	0.242	0.294	0.769	-0.404	0.546
ma.S.L12	-0.5035	0.221	-2.277	0.023	-0.937	-0.070
sigma2	1.51e+05	1.33e+04	11.371	0.000	1.25e+05	1.77e+05

Ljung-Box (L1) (Q):	0.01	Jarque-Bera (JB):	45.66
Prob(Q):	0.93	Prob(JB):	0.00
Heteroskedasticity (H):	2.61	Skew:	0.82
Prob(H) (two-sided):	0.00	Kurtosis:	5.56

The RMSE values for the models

	Test RMSE
Linear Regression	1275.867052
Naive Model	3864.279352
Simple Average Model	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315
Alpha=0.1,SimpleExponentialSmoothing	1375.393398
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	1778.564670
Alpha=0.08621,Beta=1.3722,Gamma=0.4763,TrippleExponentialSmoothing_Auto_Fit	1304.927405
Alpha=0.4,Beta=0.1,Gamma=0.2,TripleExponentialSmoothing	317.434302
Auto_ARIMA	1299.979779
(1,1,1),(2,0,3,12),Auto_SARIMA	528.607001
ARIMA(3,1,3)	1319.936734
(1,1,1)(1,1,1,12),Manual_SARIMA	359.612450

Now we sort the above table in increasing order of RMSE values

	Test RMSE
Alpha=0.4,Beta=0.1,Gamma=0.2,TripleExponentialSmoothing	317.434302
(1,1,1)(1,1,1,12),Manual_SARIMA	359.612449
(1,1,1),(2,0,3,12),Auto_SARIMA	528.590608
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
Simple Average Model	1275.081804
Linear Regression	1275.867052
6pointTrailingMovingAverage	1283.927428
Auto_ARIMA	1299.979692
Alpha=0.08621,Beta=1.3722,Gamma=0.4763,TrippleExponentialSmoothing_Auto_Fit	1304.927405
ARIMA(3,1,3)	1319.936732
9pointTrailingMovingAverage	1346.278315
Alpha=0.1,SimpleExponentialSmoothing	1375.393398
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	1778.564670
Naive Model	3864.279352

- From the above table we understood that the 'Triple exponential smoothing' gives less RMSE value i.e 317.43

Now we build the most optimum model on the complete data and predict 12 months into the future with appropriate confidence intervals/bands.

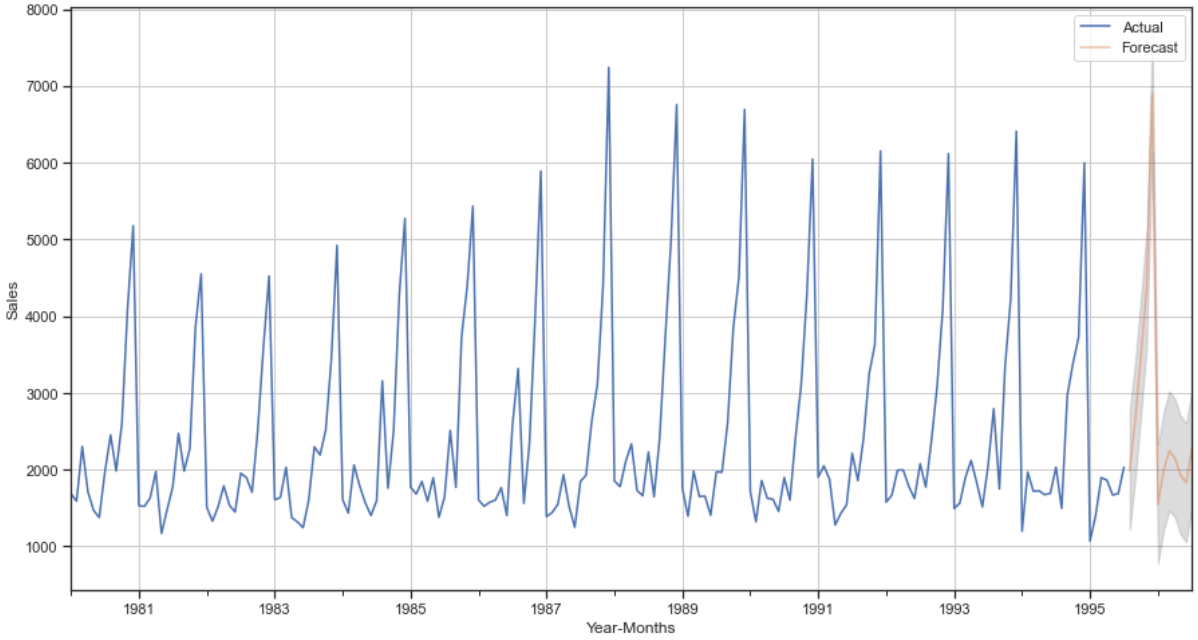
The sales predictions after applying the smoothing is as follows

	Sales_Predictions
1995-08-01	1988.782193
1995-09-01	2652.762887
1995-10-01	3483.872246
1995-11-01	4354.989747
1995-12-01	6900.103171
1996-01-01	1546.800546
1996-02-01	1981.361768
1996-03-01	2245.459724
1996-04-01	2151.066942
1996-05-01	1929.355815
1996-06-01	1830.619260
1996-07-01	2272.156151

Now we calculate the upper and lower confidence bands at 95% confidence level

	lower_CI	prediction	upper_ci
1995-08-01	1213.490105	1988.782193	2764.074282
1995-09-01	1877.470798	2652.762887	3428.054975
1995-10-01	2708.580157	3483.872246	4259.164335
1995-11-01	3579.697659	4354.989747	5130.281836
1995-12-01	6124.811083	6900.103171	7675.395260

The final optimum forecasted plot is as follows



TIME SERIES FORECASTING ON ROSE DATA SET

The top five rows of the dataset

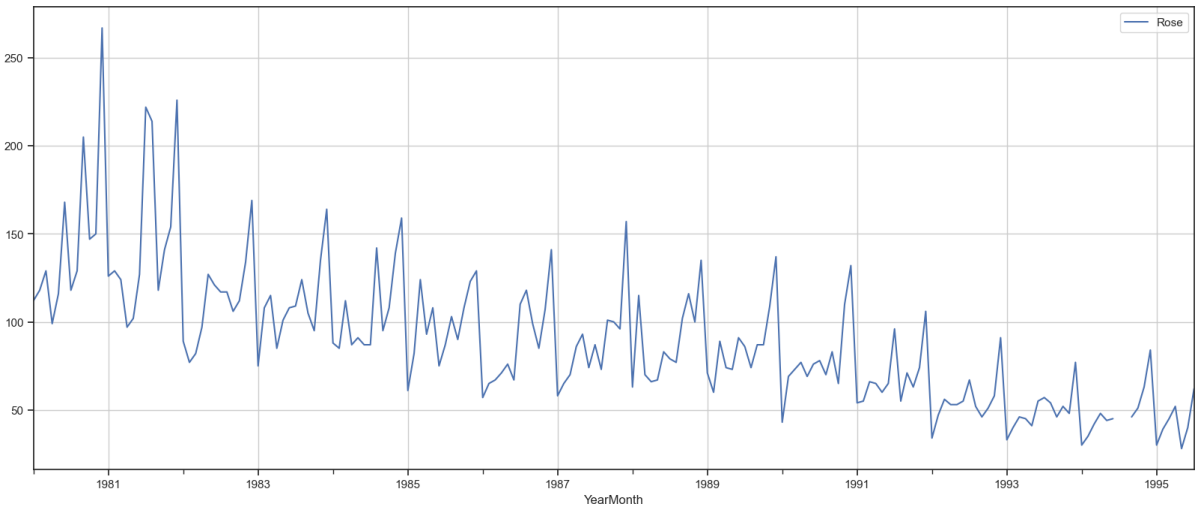
YearMonth	Rose
1980-01-01	112.0
1980-02-01	118.0
1980-03-01	129.0
1980-04-01	99.0
1980-05-01	116.0

The last five rows of the dataset

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

- There are 187 rows and 1 column present in the ‘Rose’ dataset

The graphical representation of the ‘Rose’ dataset is as follows:



Now we separate the year and month columns from the Yearmonth column, the first few rows of the dataset now looks like

YearMonth	Rose	Year	Month
1980-01-01	112.0	1980	1
1980-02-01	118.0	1980	2
1980-03-01	129.0	1980	3
1980-04-01	99.0	1980	4
1980-05-01	116.0	1980	5

Renaming the Rose as Sales, the first five rows of the data set looks like

YearMonth	Sales	Year	Month
1980-01-01	112.0	1980	1
1980-02-01	118.0	1980	2
1980-03-01	129.0	1980	3
1980-04-01	99.0	1980	4
1980-05-01	116.0	1980	5

The last five rows of the dataset are

YearMonth	Sales	Year	Month
1995-03-01	45.0	1995	3
1995-04-01	52.0	1995	4
1995-05-01	28.0	1995	5
1995-06-01	40.0	1995	6
1995-07-01	62.0	1995	7

- Now we have 187 rows and 3 columns
- The info of the data set

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 187 entries, 1980-01-01 to 1995-07-01
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Sales   185 non-null    float64
 1   Year    187 non-null    int32  
 2   Month   187 non-null    int32  
dtypes: float64(1), int32(2)
memory usage: 4.4 KB
```

The data summary looks like

	count	mean	std	min	25%	50%	75%	max
Sales	185.0	90.0	39.0	28.0	63.0	86.0	112.0	267.0
Year	187.0	1987.0	5.0	1980.0	1983.0	1987.0	1991.0	1995.0
Month	187.0	6.0	3.0	1.0	3.0	6.0	9.0	12.0

- The average sales is 90
- The minimum sales is 28
- The maximum sales is 267

- There are 2 null values present in the dataset

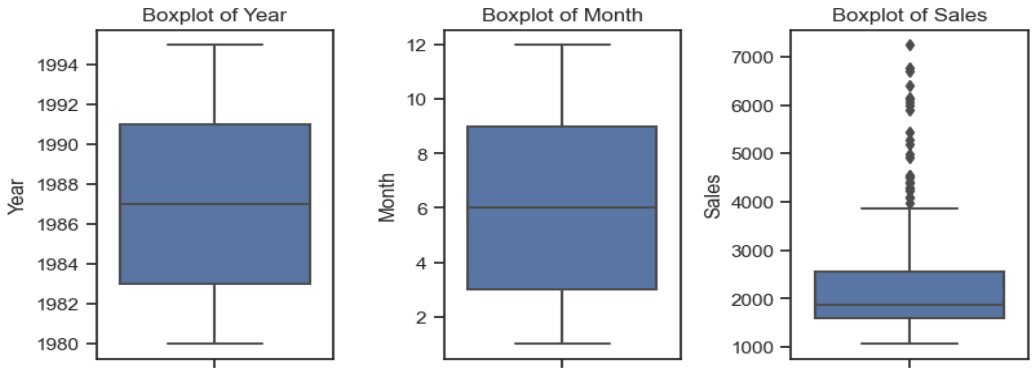
```
Sales    2
Year     0
Month    0
dtype: int64
```

The null value is present in the following dates

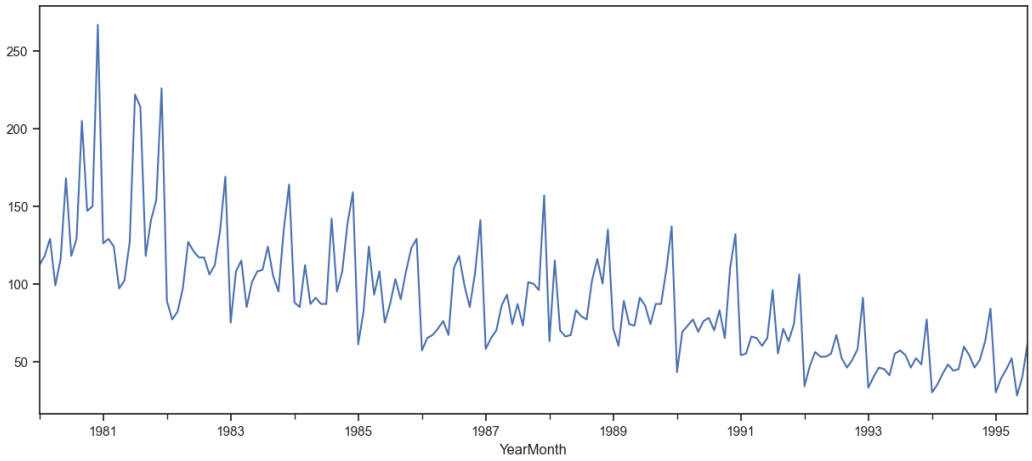
	Sales	Year	Month
YearMonth			
1994-07-01	NaN	1994	7
1994-08-01	NaN	1994	8

- Now we calculate the mean sales value for the all the seventh months of all the years present
- The mean for seventh month is 59.5
- And also we calculate the mean sales value for the all the eighth months of all the years present
- The mean for eighth month is 54
- Now we impute the missing values with the means calculated.
- There are no null values present in any features now.

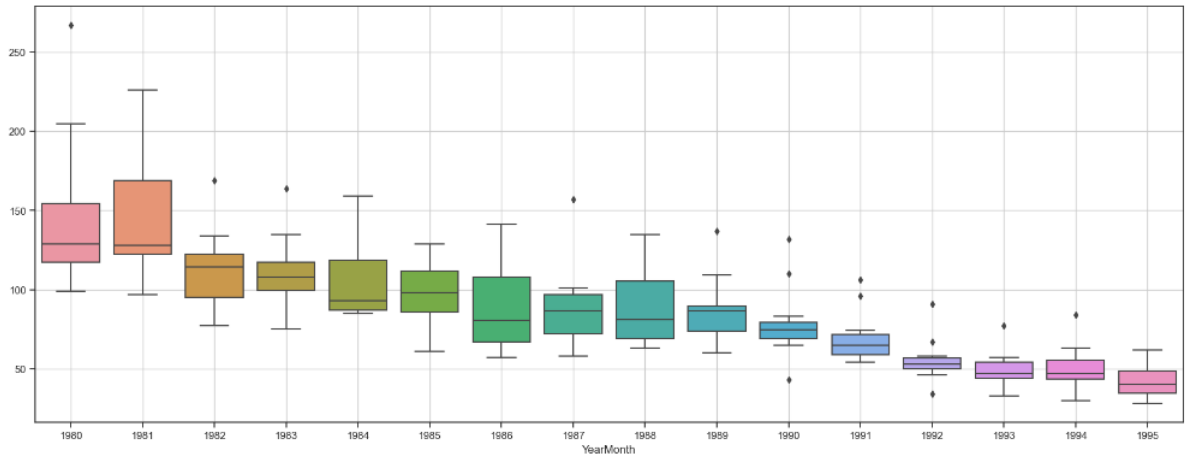
The boxplots of the features present is as follows



The Time series plot we got is as follows:



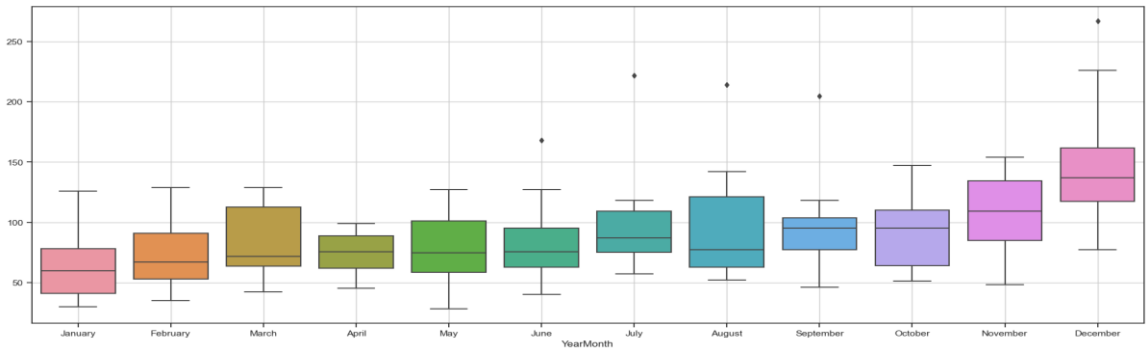
The yearly boxplot is



As we got to know from the Time Series plot, the box plots over here also indicates a measure of trend being present. Also, we see that Sales of Rose Wine has some outliers for certain years.

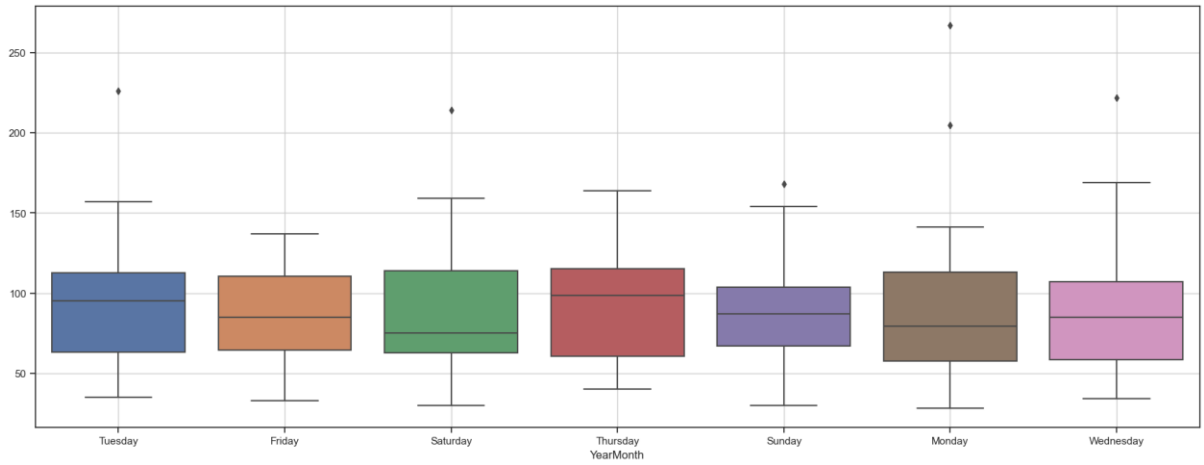
The monthly boxplot is

Since this is a monthly data, let us plot a box and whisker (1.5* IQR) plot to understand the spread of the data and check for outliers for every month across all the years, if any.

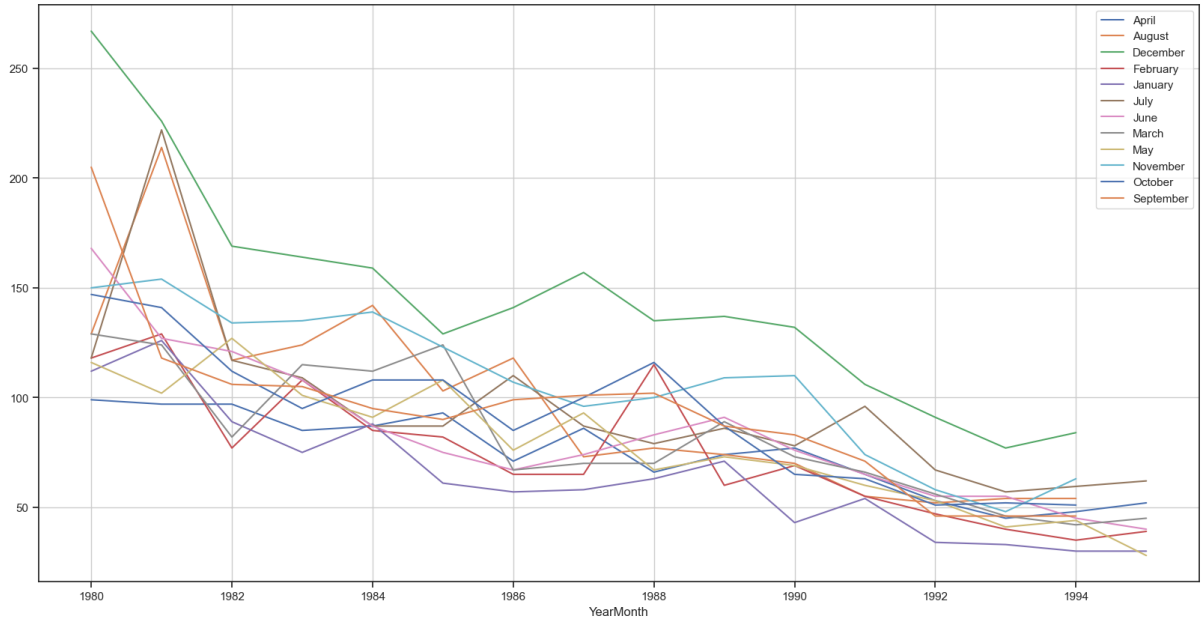


The highest such numbers are being recorded in the month of December across various years.

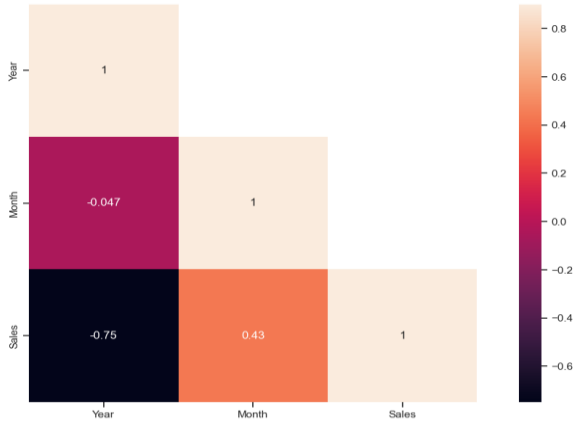
The weekly boxplot is



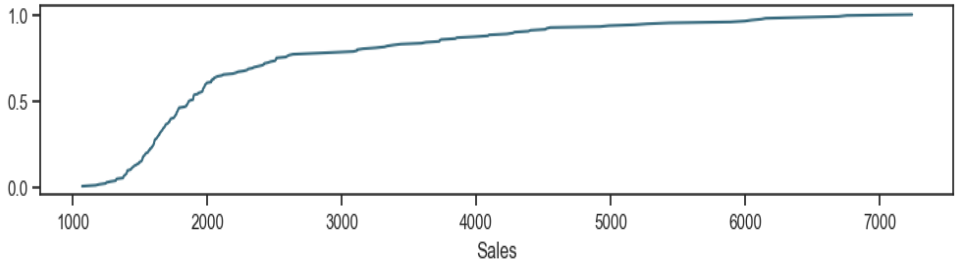
The graph of monthly sales across years is



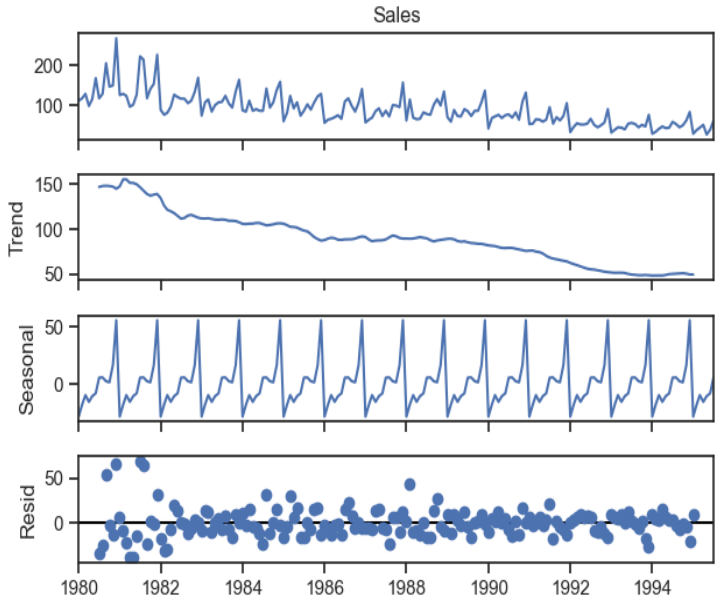
The correlation plot is as follows



The plot of empirical cummulative distribution function



The decomposition of the sales into trend, seasonality and residuals in additive mode



We see that the residuals are located around 0 from the plot of the residuals in the decomposition.

Also there is a trend.

Also there are no outliers in the dataset.

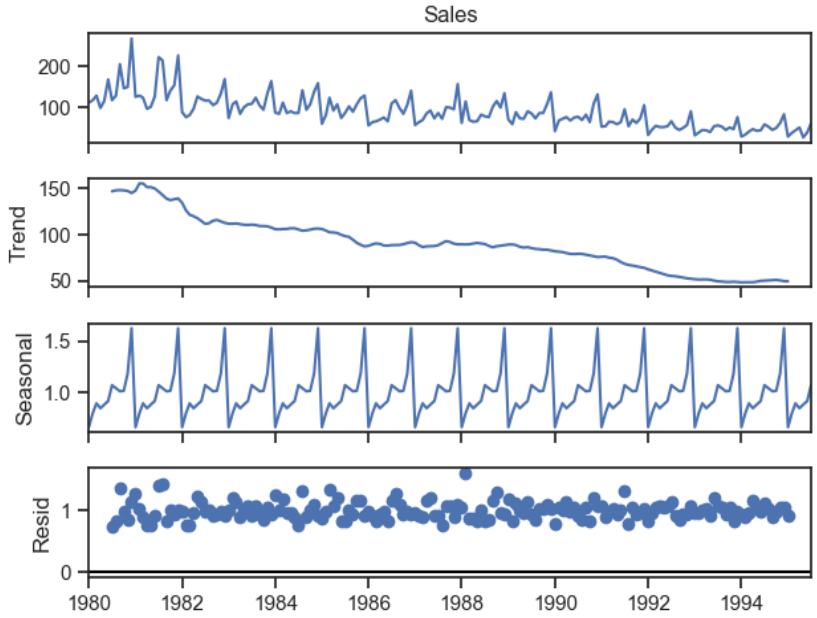
The values after decompostion are

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

Seasonality
 YearMonth
1980-01-01   -28.031994
1980-02-01   -17.543105
1980-03-01    -9.418105
1980-04-01   -15.230605
1980-05-01   -10.328819
1980-06-01    -7.810962
1980-07-01     5.718006
1980-08-01     5.931895
1980-09-01     2.651339
1980-10-01     1.748562
1980-11-01    16.723562
1980-12-01    55.590228
Name: seasonal, dtype: float64

Residual
 YearMonth
1980-01-01          NaN
1980-02-01          NaN
1980-03-01          NaN
1980-04-01          NaN
1980-05-01          NaN
1980-06-01          NaN
1980-07-01   -34.801339
1980-08-01   -25.056895
1980-09-01    53.973661
1980-10-01    -2.831895
1980-11-01   -14.140228
1980-12-01    66.284772
Name: resid, dtype: float64
```

The decomposition of the sales into trend, seasonality and residuals in multiplicative mode



As per the 'additive' decomposition, we see that there is a pronounced trend in the earlier years of the data. There is seasonality as well.

Also there are no outliers in the dataset.

The values after decomposition are

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

Seasonality
 YearMonth
1980-01-01    0.668577
1980-02-01    0.804550
1980-03-01    0.898744
1980-04-01    0.851237
1980-05-01    0.886934
1980-06-01    0.921546
1980-07-01    1.074644
1980-08-01    1.044683
1980-09-01    1.015406
1980-10-01    1.020108
1980-11-01    1.189232
1980-12-01    1.624338
Name: seasonal, dtype: float64

Residual
 YearMonth
1980-01-01         NaN
1980-02-01         NaN
1980-03-01         NaN
1980-04-01         NaN
1980-05-01         NaN
1980-06-01         NaN
1980-07-01    0.746542
1980-08-01    0.833636
1980-09-01    1.360672
1980-10-01    0.973117
1980-11-01    0.855614
1980-12-01    1.132642
Name: resid, dtype: float64
```

Now we split the data into train and test series, In time series forecasting the splitting of train and test data set is not random .

After splitting at 1991, the shape of the train and test data is

```
Shape of datasets:
train dataset:  (132, 3)
test dataset:  (55, 3)
```

The top five rows and bottom five rows of the train and test data set is as follows :

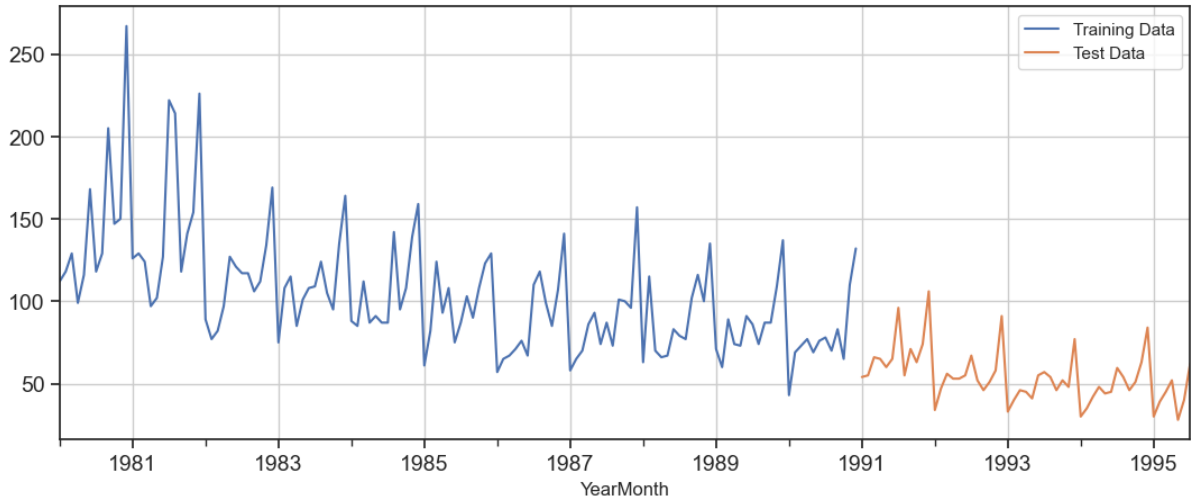
```
Rows of dataset:
First few rows of Training Data
            Year  Month  Sales
YearMonth
1980-01-01  1980      1  112.0
1980-02-01  1980      2  118.0
1980-03-01  1980      3  129.0
1980-04-01  1980      4   99.0
1980-05-01  1980      5  116.0

Last few rows of Training Data
            Year  Month  Sales
YearMonth
1990-08-01  1990      8   70.0
1990-09-01  1990      9   83.0
1990-10-01  1990     10   65.0
1990-11-01  1990     11  110.0
1990-12-01  1990     12  132.0

First few rows of Test Data
            Year  Month  Sales
YearMonth
1991-01-01  1991      1   54.0
1991-02-01  1991      2   55.0
1991-03-01  1991      3   66.0
1991-04-01  1991      4   65.0
1991-05-01  1991      5   60.0

Last few rows of Test Data
            Year  Month  Sales
YearMonth
1995-03-01  1995      3   45.0
1995-04-01  1995      4   52.0
1995-05-01  1995      5   28.0
1995-06-01  1995      6   40.0
1995-07-01  1995      7   62.0
```

The graphical plot representing the train and test data set is as follows :



It is difficult to predict the future observations if such an instance have not happened in the past. From our train-test split we are predicting likewise behavior as compared to the past years.

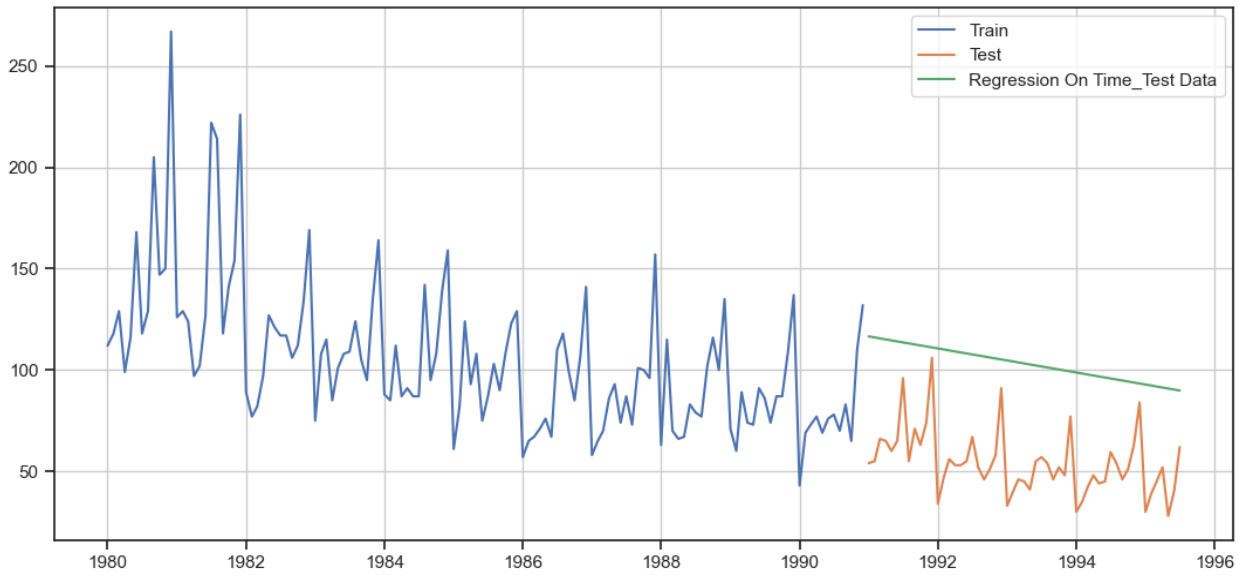
LINEAR REGRESSION

For this particular linear regression, we are going to regress the 'Sales' variable against the order of the occurrence.

Train and test time instance is as follows:

```
Training Time instance
 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 3
4, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65,
66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97,
98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123,
124, 125, 126, 127, 128, 129, 130, 131, 132]
Test Time instance
 [43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 7
4, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97]
```

Now that our training and test data has been modified, let us go ahead use $LinearRegression$ to build the model on the training data and test the model on the test data After applying the linear regression, the forecasted plot looks like



The RMSE value on applying on test data is

	Test RMSE
Linear Regression	51.080941

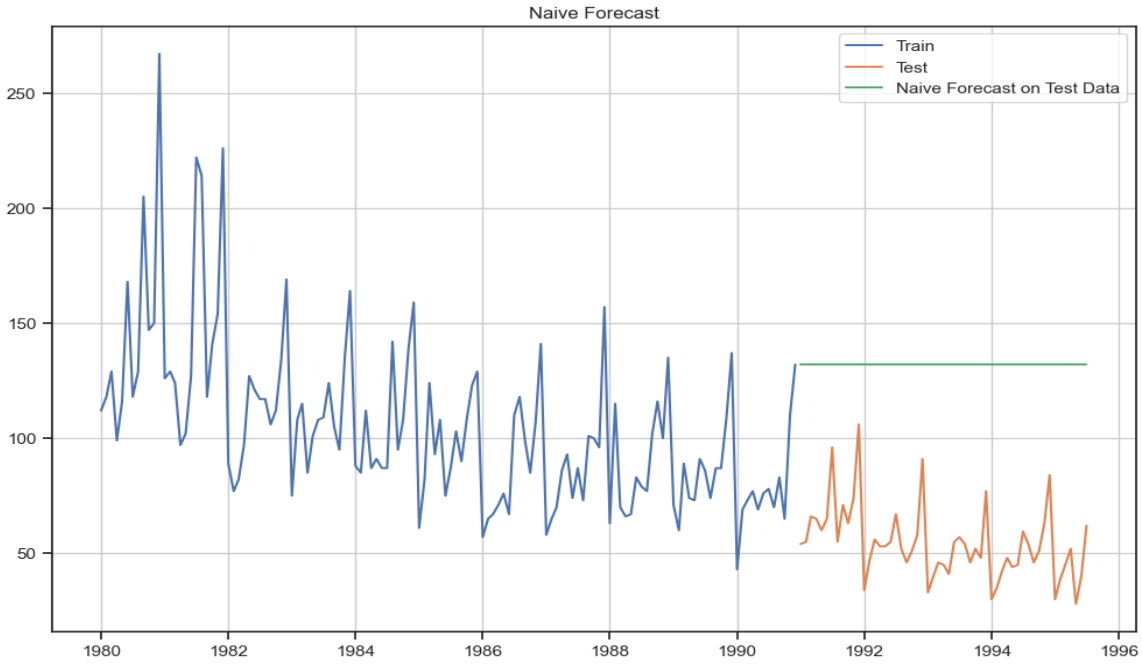
NAÏVE APPROACH

For this particular naive model, we say that the prediction for tomorrow is the same as today and the prediction for day after tomorrow is tomorrow and since the prediction of tomorrow is same as today, therefore the prediction for day after tomorrow is also today.

After applying the naïve approach, we get

```
YearMonth
1991-01-01    132.0
1991-02-01    132.0
1991-03-01    132.0
1991-04-01    132.0
1991-05-01    132.0
Name: naive, dtype: float64
```

After applying the naïve approach, the forecasted plot looks like



The RMSE values are

	Test RMSE
Linear Regression	51.080941
Naive Model	79.304391

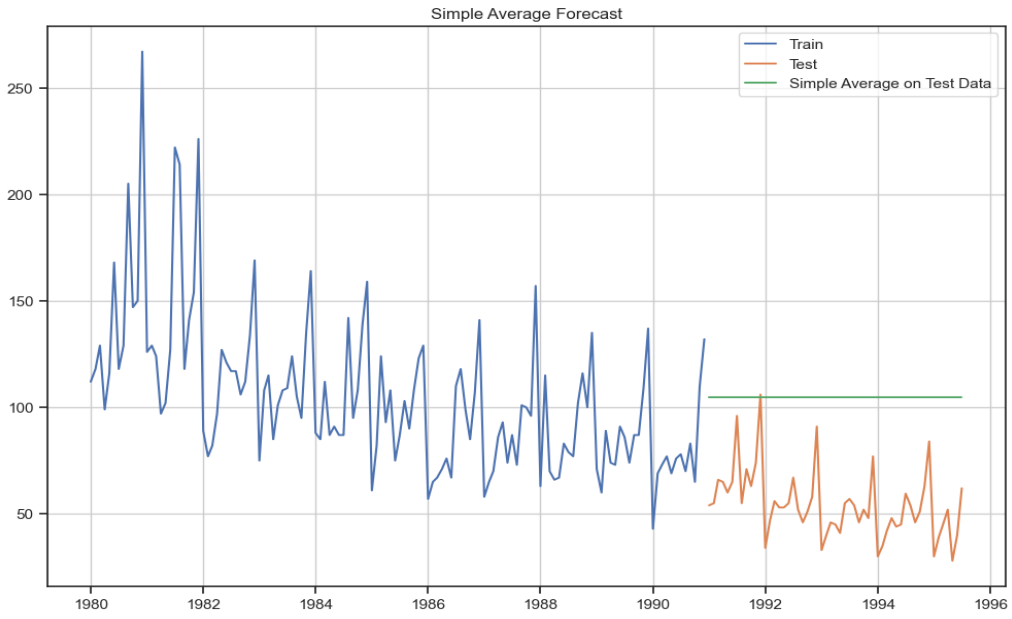
SIMPLE AVERAGE

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

After applying the simple average method, the forecasted mean is

YearMonth	Year	Month	Sales	mean_forecast
1991-01-01	1991	1	54.0	104.939394
1991-02-01	1991	2	55.0	104.939394
1991-03-01	1991	3	66.0	104.939394
1991-04-01	1991	4	65.0	104.939394
1991-05-01	1991	5	60.0	104.939394

The forecasted plot is



The RMSE value on applying on test data

	Test RMSE
Linear Regression	51.080941
Naive Model	79.304391
Simple Average Model	53.049755

MOVING AVERAGE

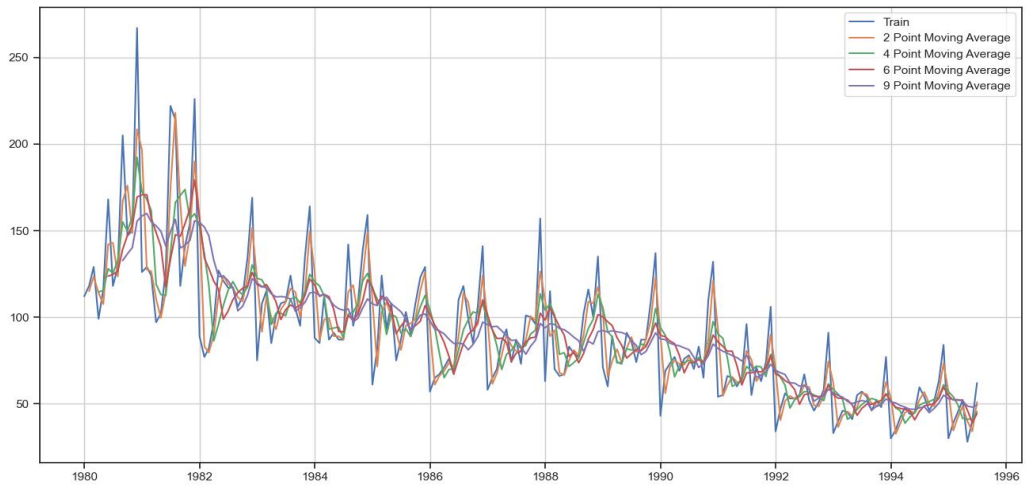
For the moving average model, we are going to calculate rolling means (or moving averages) for different intervals. The best interval can be determined by the maximum accuracy (or the minimum error) over here.

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

The moving average is calculated for different rolling values, It is as follows

YearMonth	Year	Month	Sales	Trailing_2	Trailing_4	Trailing_6	Trailing_9
1980-01-01	1980	1	112.0	NaN	NaN	NaN	NaN
1980-02-01	1980	2	118.0	115.0	NaN	NaN	NaN
1980-03-01	1980	3	129.0	123.5	NaN	NaN	NaN
1980-04-01	1980	4	99.0	114.0	114.5	NaN	NaN
1980-05-01	1980	5	116.0	107.5	115.5	NaN	NaN

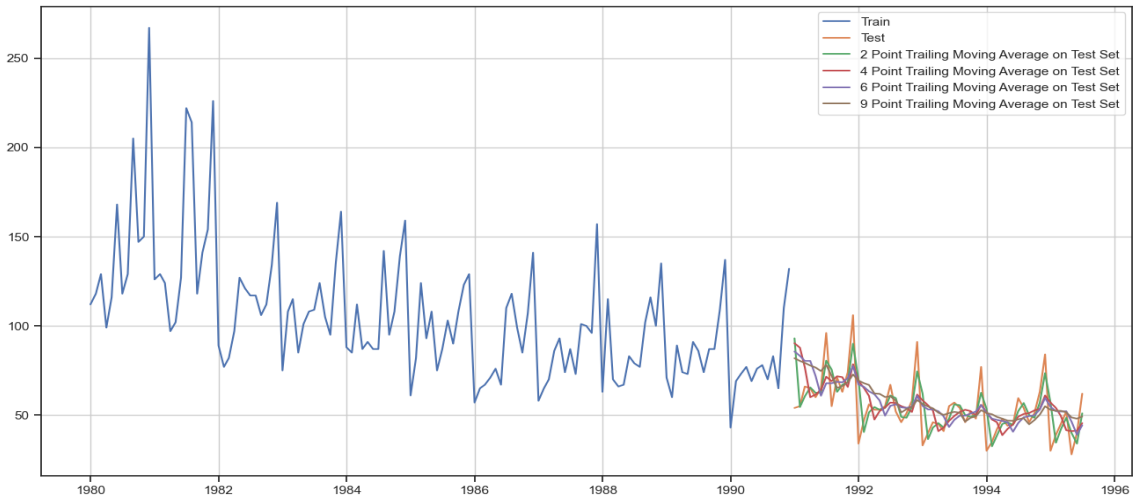
The forecasted plot looks like



The shape of the trainling moving average train and test data is

```
(55, 7)
(132, 7)
```

Before we go on to build the various Exponential Smoothing models, let us plot all the models and compare the Time Series plots.



The RMSE values are as follows :

	Test RMSE
Linear Regression	51.080941
Naive Model	79.304391
Simple Average Model	53.049755
2pointTrailingMovingAverage	11.589082
4pointTrailingMovingAverage	14.506190
6pointTrailingMovingAverage	14.558008
9pointTrailingMovingAverage	14.797139

SIMPLE EXPONENTIAL SMOOTHING

The parameters are

```
{'smoothing_level': 0.12362013660706869,
 '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}
```

The forecasted values are

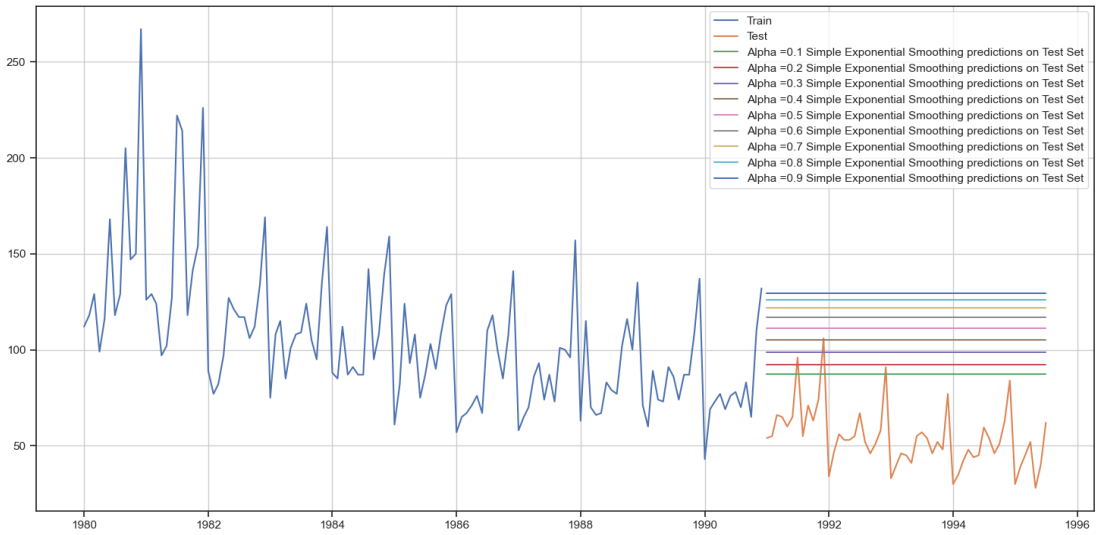
YearMonth	Year	Month	Sales	predict
1991-01-01	1991	1	54.0	87.983766
1991-02-01	1991	2	55.0	87.983766
1991-03-01	1991	3	66.0	87.983766
1991-04-01	1991	4	65.0	87.983766
1991-05-01	1991	5	60.0	87.983766

Setting different alpha values.

Remember, the higher the alpha value more weightage is given to the more recent observation. That means, what happened recently will happen again.

We will run a loop with different alpha values to understand which particular value works best for alpha on the test set

Now for different values of alpha we plot the forecasted plot



The RMSE values are

	Alpha Values	Train RMSE	Test RMSE
0	0.1	31.815610	36.429535
1	0.2	31.979391	40.957988
2	0.3	32.470164	47.096522
3	0.4	33.035130	53.356493
4	0.5	33.682839	59.229384
5	0.6	34.441171	64.558022
6	0.7	35.323261	69.284383
7	0.8	36.334596	73.359904
8	0.9	37.482782	76.725002

Note that we have low RMSE value for when alpha=0.1

The RMSE values till now for different methods is

	Test RMSE
Linear Regression	51.080941
Naive Model	79.304391
Simple Average Model	53.049755
2pointTrailingMovingAverage	11.589082
4pointTrailingMovingAverage	14.506190
6pointTrailingMovingAverage	14.558008
9pointTrailingMovingAverage	14.797139
Alpha=0.1,SimpleExponentialSmoothing	36.429535

DOUBLE EXPONENTIAL SMOOTHING (HOLT'S MODEL)

Two parameters α and β are estimated in this model. Level and Trend are accounted for in this model.

The parameters are

```
{'smoothing_level': 0.16213319620268435,
 'smoothing_trend': 0.13152157897780353,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 112.0,
 'initial_trend': 6.0,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

The predicted values are as follows

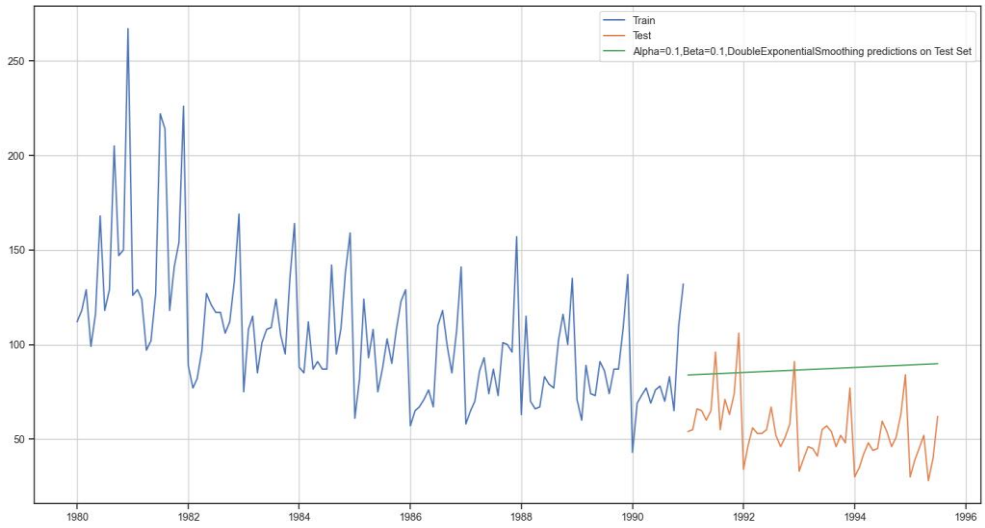
YearMonth	Year	Month	Sales	predict
1991-01-01	1991	1	54.0	87.249992
1991-02-01	1991	2	55.0	88.152721
1991-03-01	1991	3	66.0	89.055450
1991-04-01	1991	4	65.0	89.958179
1991-05-01	1991	5	60.0	90.860909

For different values of aplha and beta, we fing the RMSE values and sort them to find the low RMSE value.

	Alpha Values	Beta Values	Train RMSE	Test RMSE
0	0.1	0.1	34.439111	36.510010
1	0.1	0.2	33.450729	48.221436
2	0.1	0.3	33.145789	77.649847
3	0.1	0.4	33.262191	99.064536
4	0.1	0.5	33.688415	123.742433
...	...	...	...	...
95	1.0	0.6	51.831610	801.137173
96	1.0	0.7	54.497039	841.349112
97	1.0	0.8	57.365879	853.421959
98	1.0	0.9	60.474309	834.167545
99	1.0	1.0	63.873454	779.536777

100 rows × 4 columns

The forecasted plot is as follows:



The RMSE values till now for different methods is

	Test RMSE
Linear Regression	51.080941
Naive Model	79.304391
Simple Average Model	53.049755
2pointTrailingMovingAverage	11.589082
4pointTrailingMovingAverage	14.506190
6pointTrailingMovingAverage	14.558008
9pointTrailingMovingAverage	14.797139
Alpha=0.1,SimpleExponentialSmoothing	36.429535
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	36.510010

TRIPLE EXPONENTIAL SMOOTHING (HOLT - WINTER'S MODEL)

Three parameters , β and γ are estimated in this model. Level, Trend and Seasonality are accounted for in this model

The parameters for when trend and seasonality being additive, additive

```
{'smoothing_level': 0.08830330642635406,
 'smoothing_trend': 6.730635331927582e-05,
 'smoothing_seasonal': 0.004455138229351625,
 'damping_trend': nan,
 'initial_level': 146.88752868155674,
 'initial_trend': -0.5492163940406024,
 'initial_seasons': array([-31.12207537, -18.81171138, -10.86052241, -21.52235816,
       -12.68359535,  -7.17529564,   2.7456236 ,   8.84900094,
         4.85724354,   2.9520333 ,  21.05004912,  63.29916317]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

The parameters for when trend and seasonality being additive, multiplicative

```
{'smoothing_level': 0.07132109562890512,
 'smoothing_trend': 0.04553831096563722,
 'smoothing_seasonal': 8.356711212063695e-07,
 'damping_trend': nan,
 'initial_level': 134.25655591779326,
 'initial_trend': -0.8038265942903572,
 'initial_seasons': array([0.83746068, 0.94985307, 1.03812083, 0.90732186, 1.02043162,
        1.11131741, 1.22228039, 1.30104211, 1.23132915, 1.20610008,
        1.40577823, 1.93832412]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

The parameters for when trend and seasonality being multiplicative, multiplicative

```
{'smoothing_level': 0.0521696988178152,
 'smoothing_trend': 0.03546119906270268,
 'smoothing_seasonal': 0.00028476357235519487,
 'damping_trend': nan,
 'initial_level': 165.34400052966296,
 'initial_trend': 0.9923066985722868,
 'initial_seasons': array([0.68749747, 0.77992196, 0.85424762, 0.74551695, 0.83903295,
        0.91333126, 1.00288146, 1.06901269, 1.01412988, 0.99063168,
        1.15690782, 1.59275092]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

The parameters for when trend and seasonality being multiplicative, additive

```
{'smoothing_level': 0.04392379552557389,
 'smoothing_trend': 2.2382727020656378e-05,
 'smoothing_seasonal': 0.0005301252122432588,
 'damping_trend': nan,
 'initial_level': 141.25278376466085,
 'initial_trend': 0.993889634188071,
 'initial_seasons': array([-22.44377304, -10.02696656,  -1.98981311, -12.58528373,
        -3.74636822,   1.74412094,  11.67398645,  17.78894186,
        13.81217532,  11.92521254,  30.01867872,  72.29048742]),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

The forecasted values of all the above parameters

YearMonth	Year	Month	Sales	predict_ta_sa	predict_ta_sm	predict_tm_sm	predict_tm_sa
1991-01-01	1991	1	54.0	42.672382	56.334597	55.894208	43.085644
1991-02-01	1991	2	55.0	54.439917	63.692059	63.236692	55.102106
1991-03-01	1991	3	66.0	61.841877	69.388935	69.062862	62.741356
1991-04-01	1991	4	65.0	50.636896	60.452304	60.106162	51.750473
1991-05-01	1991	5	60.0	58.918913	67.770362	67.444092	60.196334

The RMSE values after applying smoothing

	Test RMSE
Linear Regression	51.080941
Naive Model	79.304391
Simple Average Model	53.049755
2pointTrailingMovingAverage	11.589082
4pointTrailingMovingAverage	14.506190
6pointTrailingMovingAverage	14.558008
9pointTrailingMovingAverage	14.797139
Alpha=0.1,SimpleExponentialSmoothing	36.429535
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	36.510010
Alpha=0.08621,Beta=1.3722,Gamma=0.4763,TrippleExponentialSmoothing_Auto_Fit	37.192624

For different values of alpha, beta, gamma we forecast the values, the top five rows of the data looks like

YearMonth	Year	Month	Sales	(predict_ta_sa, 0.1, 0.1, 0.1)	(predict_ta_sa, 0.1, 0.1, 0.2)	(predict_ta_sa, 0.1, 0.1, 0.30000000000000004)	(predict_ta_sa, 0.1, 0.1, 0.4)	(predict_ta_sa, 0.1, 0.1, 0.5)	(predict_ta_sa, 0.1, 0.1, 0.6)	(predict_ta_sa, 0.1, 0.1, 0.7000000000000001)	...
1991-01-01	1991	1	54.0	45.711834	46.537302	46.559436	46.071952	45.225493	44.012323	42.371394	...
1991-02-01	1991	2	55.0	56.369270	60.659980	62.645947	63.356292	63.447907	63.255359	62.959944	...
1991-03-01	1991	3	66.0	63.004762	65.794341	66.979401	67.649481	68.330584	69.025477	69.443834	...
1991-04-01	1991	4	65.0	51.663022	58.369250	62.190538	64.397061	65.989684	67.481555	68.978770	...
1991-05-01	1991	5	60.0	58.931424	61.246579	62.077338	62.127541	62.025325	62.111373	62.439476	...

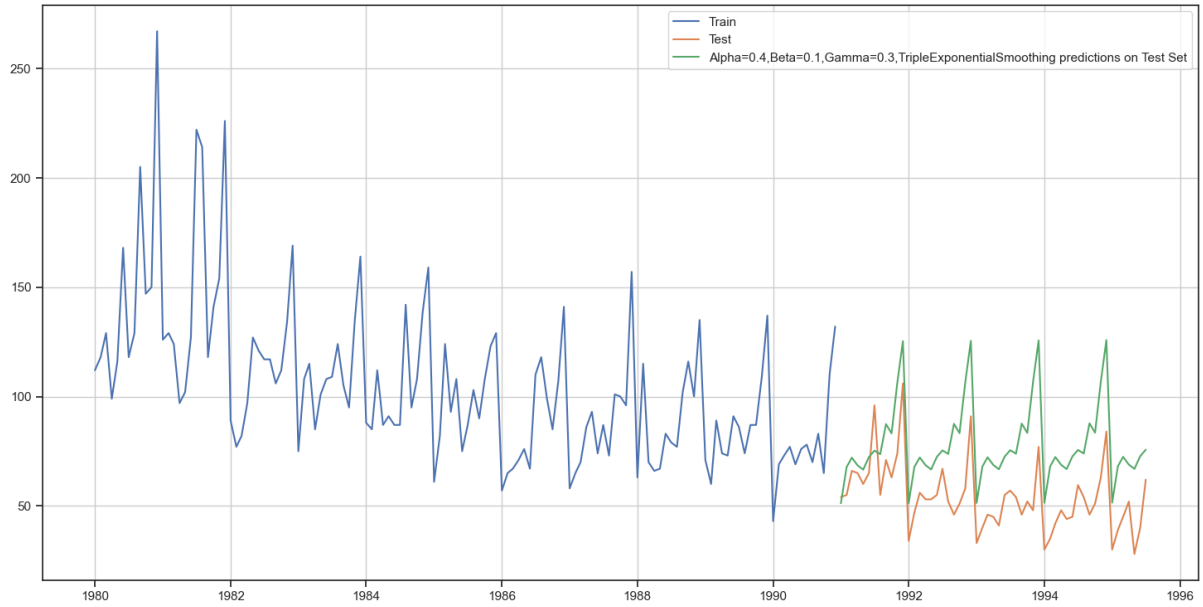
5 rows × 3461 columns

After sorting through RMSE values

	Test RMSE
Linear Regression	51.080941
Naive Model	79.304391
Simple Average Model	53.049755
2pointTrailingMovingAverage	11.589082
4pointTrailingMovingAverage	14.506190
6pointTrailingMovingAverage	14.558008
9pointTrailingMovingAverage	14.797139
Alpha=0.1,SimpleExponentialSmoothing	36.429535
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	36.510010
Alpha=0.08621,Beta=1.3722,Gamma=0.4763,TrippleExponentialSmoothing_Auto_Fit	37.192624
Alpha=0.2,Beta=0.7,Gamma=0.2,TripleExponentialSmoothing	8.992350

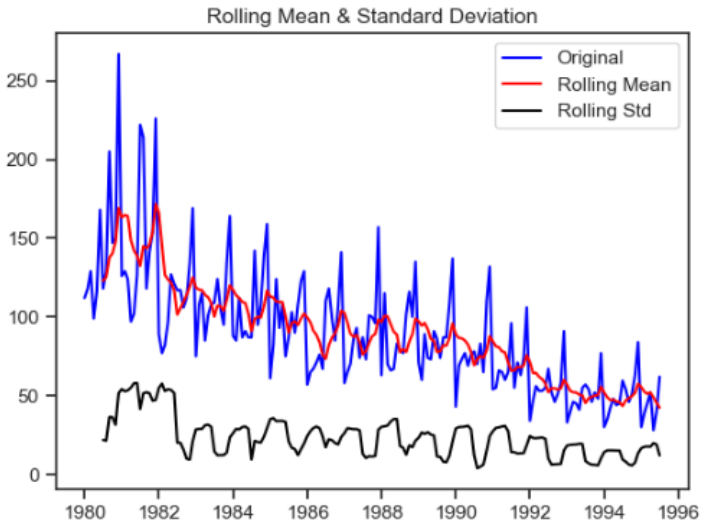
Note that the RMSE value obtained is much lesser than the RMSE of the other methods

The forecasted plot looks like



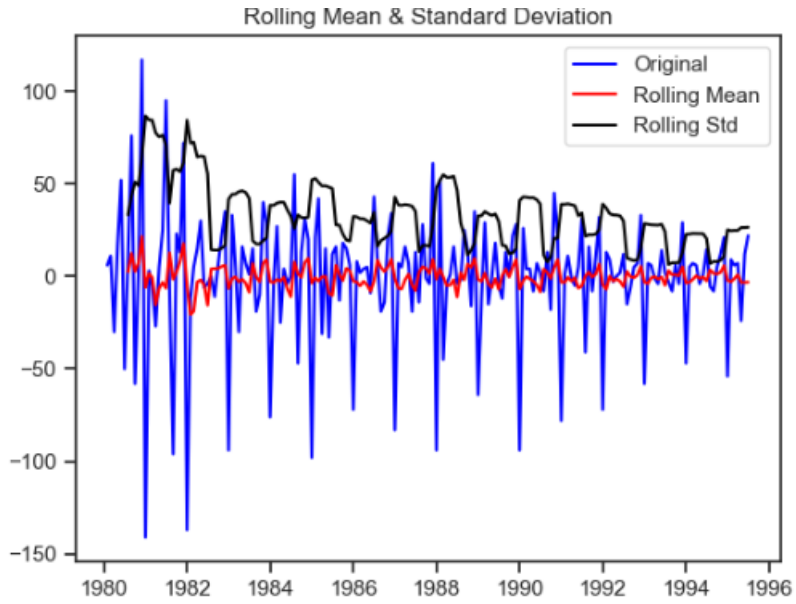
Now we check for stationarity at alpha=0.05

The plot for rolling mean and standard deviation is as follows




```
Results of Dickey-Fuller Test:
Test Statistic                  -1.892338
p-value                          0.335674
#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
```

- The p value is 0.33 which is greater than 0.05
- So we accept the null hypothesis i.e the plot is not stationary
- The above plot is not stationary, so we have to make it stationary by differencing
- After differencing the plot looks like




```
Results of Dickey-Fuller Test:
Test Statistic                -8.032729e+00
p-value                        1.938803e-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
```

BUILDING OF ARIMA MODEL

Some parameter combinations for the model are as follows

```
Some parameter combinations for the Model...
Model: (0, 1, 1)
Model: (0, 1, 2)
Model: (0, 1, 3)
Model: (1, 1, 0)
Model: (1, 1, 1)
Model: (1, 1, 2)
Model: (1, 1, 3)
Model: (2, 1, 0)
Model: (2, 1, 1)
Model: (2, 1, 2)
Model: (2, 1, 3)
Model: (3, 1, 0)
Model: (3, 1, 1)
Model: (3, 1, 2)
Model: (3, 1, 3)
```

Now we get the Akaike Information Criteria (AIC) for all the above combinations of the models

```
ARIMA(0, 1, 0) - AIC:1333.1546729124348
ARIMA(0, 1, 1) - AIC:1282.3098319748315
ARIMA(0, 1, 2) - AIC:1279.6715288535752
ARIMA(0, 1, 3) - AIC:1280.5453761734668
ARIMA(1, 1, 0) - AIC:1317.3503105381526
ARIMA(1, 1, 1) - AIC:1280.5742295380076
ARIMA(1, 1, 2) - AIC:1279.870723423191
ARIMA(1, 1, 3) - AIC:1281.8707223309993
ARIMA(2, 1, 0) - AIC:1298.6110341604908
ARIMA(2, 1, 1) - AIC:1281.5078621868597
ARIMA(2, 1, 2) - AIC:1281.8707222264168
```

```
C:\Users\THANUSRI\anaconda31\Lib\site-packages\statsmodels\base\model.py:607: ConvergenceWarning: Maximum Likelihood optimizati
on failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

```
ARIMA(2, 1, 3) - AIC:1274.6956920197993
ARIMA(3, 1, 0) - AIC:1297.4810917271707
ARIMA(3, 1, 1) - AIC:1282.4192776272025
ARIMA(3, 1, 2) - AIC:1283.720740597717
ARIMA(3, 1, 3) - AIC:1278.6543716254587
```

```
C:\Users\THANUSRI\anaconda31\Lib\site-packages\statsmodels\base\model.py:607: ConvergenceWarning: Maximum Likelihood optimizati
on failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

now we sort in decreasing order of AIC values

	param	AIC
11	(2, 1, 3)	1274.695692
15	(3, 1, 3)	1278.654372
2	(0, 1, 2)	1279.671529
6	(1, 1, 2)	1279.870723
3	(0, 1, 3)	1280.545376
5	(1, 1, 1)	1280.574230
9	(2, 1, 1)	1281.507862
10	(2, 1, 2)	1281.870722
7	(1, 1, 3)	1281.870722
1	(0, 1, 1)	1282.309832
13	(3, 1, 1)	1282.419278
14	(3, 1, 2)	1283.720741
12	(3, 1, 0)	1297.481092
8	(2, 1, 0)	1298.611034
4	(1, 1, 0)	1317.350311
0	(0, 1, 0)	1333.154673

Arima summary for the first model i.e (2,1,3) is

SARIMAX Results

Dep. Variable:	Sales	No. Observations:	132
Model:	ARIMA(2, 1, 3)	Log Likelihood	-631.348
Date:	Thu, 11 Apr 2024	AIC	1274.696
Time:	21:28:21	BIC	1291.947
Sample:	01-01-1980 - 12-01-1990	HQIC	1281.706
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-1.6778	0.084	-20.027	0.000	-1.842	-1.514
ar.L2	-0.7286	0.084	-8.698	0.000	-0.893	-0.564
ma.L1	1.0444	0.602	1.734	0.083	-0.136	2.225
ma.L2	-0.7722	0.130	-5.923	0.000	-1.028	-0.517
ma.L3	-0.9046	0.546	-1.658	0.097	-1.974	0.165
sigma2	859.1645	505.933	1.698	0.089	-132.445	1850.774

Ljung-Box (L1) (Q):	0.02	Jarque-Bera (JB):	24.47
Prob(Q):	0.88	Prob(JB):	0.00
Heteroskedasticity (H):	0.40	Skew:	0.71
Prob(H) (two-sided):	0.00	Kurtosis:	4.57

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Now we evaluate this model on test data using RMSE

	Test RMSE
Linear Regression	51.080941
Naive Model	79.304391
Simple Average Model	53.049755
2pointTrailingMovingAverage	11.589082
4pointTrailingMovingAverage	14.506190
6pointTrailingMovingAverage	14.558008
9pointTrailingMovingAverage	14.797139
Alpha=0.1,SimpleExponentialSmoothing	36.429535
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	36.510010
Alpha=0.08621,Beta=1.3722,Gamma=0.4763,TrippleExponentialSmoothing_Auto_Fit	37.192624
Alpha=0.2,Beta=0.7,Gamma=0.2,TripleExponentialSmoothing	8.992350
Auto_ARIMA	36.415314

BUILDING OF SARIMA MODEL

```
Examples of some parameter combinations for Model..
Model: (0, 1, 1)(0, 0, 1, 12)
Model: (0, 1, 2)(0, 0, 2, 12)
Model: (0, 1, 3)(0, 0, 3, 12)
Model: (1, 1, 0)(1, 0, 0, 12)
Model: (1, 1, 1)(1, 0, 1, 12)
Model: (1, 1, 2)(1, 0, 2, 12)
Model: (1, 1, 3)(1, 0, 3, 12)
Model: (2, 1, 0)(2, 0, 0, 12)
Model: (2, 1, 1)(2, 0, 1, 12)
Model: (2, 1, 2)(2, 0, 2, 12)
Model: (2, 1, 3)(2, 0, 3, 12)
Model: (3, 1, 0)(3, 0, 0, 12)
Model: (3, 1, 1)(3, 0, 1, 12)
Model: (3, 1, 2)(3, 0, 2, 12)
Model: (3, 1, 3)(3, 0, 3, 12)
```

Now we get the Akaike Information Criteria (AIC) for all the above combinations of the models and we sort in decreasing order of AIC values. The top five rows looks like

	param	seasonal	AIC
222	(3, 1, 1)	(3, 0, 2, 12)	774.400287
238	(3, 1, 2)	(3, 0, 2, 12)	774.880938
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

The SARIMAX results is as follows

SARIMAX Results

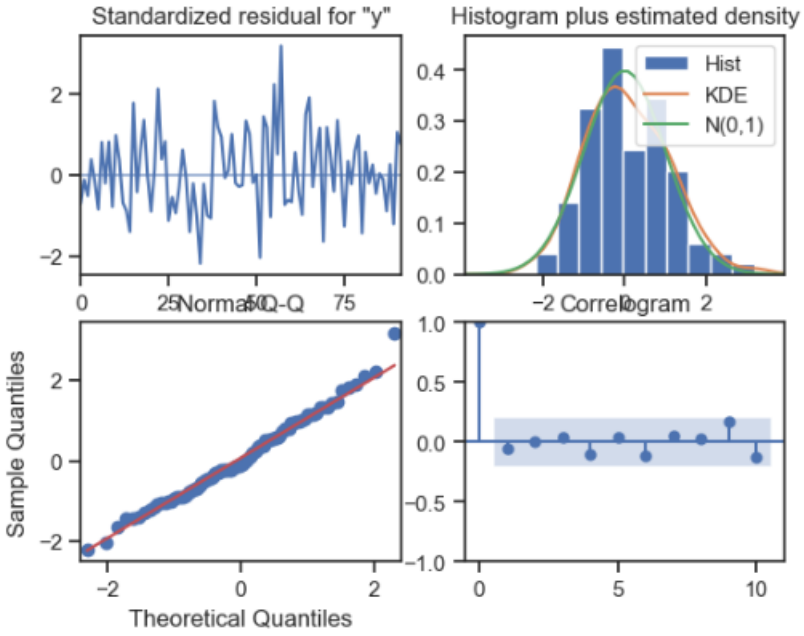
Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(3, 1, 1)x(3, 0, [1, 2], 12)	Log Likelihood	-377.200
Date:	Thu, 11 Apr 2024	AIC	774.400
Time:	21:33:39	BIC	799.618
Sample:	0 - 132	HQIC	784.578
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.0464	0.126	0.367	0.714	-0.201	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	-13.907	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	192.1320	39.623	4.849	0.000	114.473	269.791

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

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

The plot diagonastics

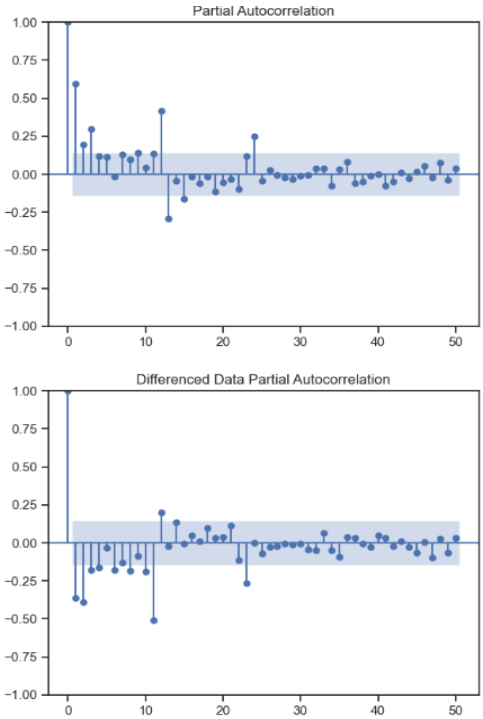


Now we evaluate this model on test data using RMSE

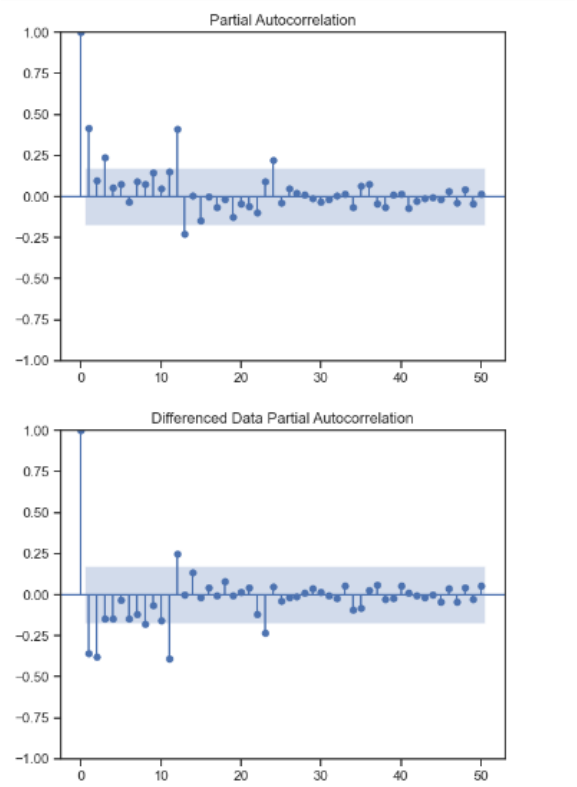
	Test RMSE
Linear Regression	51.080941
Naive Model	79.304391
Simple Average Model	53.049755
2pointTrailingMovingAverage	11.589082
4pointTrailingMovingAverage	14.506190
6pointTrailingMovingAverage	14.558008
9pointTrailingMovingAverage	14.797139
Alpha=0.1,SimpleExponentialSmoothing	36.429535
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	36.510010
Alpha=0.08621,Beta=1.3722,Gamma=0.4763,TrippleExponentialSmoothing_Auto_Fit	37.192624
Alpha=0.2,Beta=0.7,Gamma=0.2,TripleExponentialSmoothing	8.992350
Auto_ARIMA	36.415314
(3,1,1),(3,0,2,12),Auto_SARIMA	18.534956

MANUAL ARIMA

The ACF and PACF plots are as follows



After differencing to convert it from non stationary to stationary

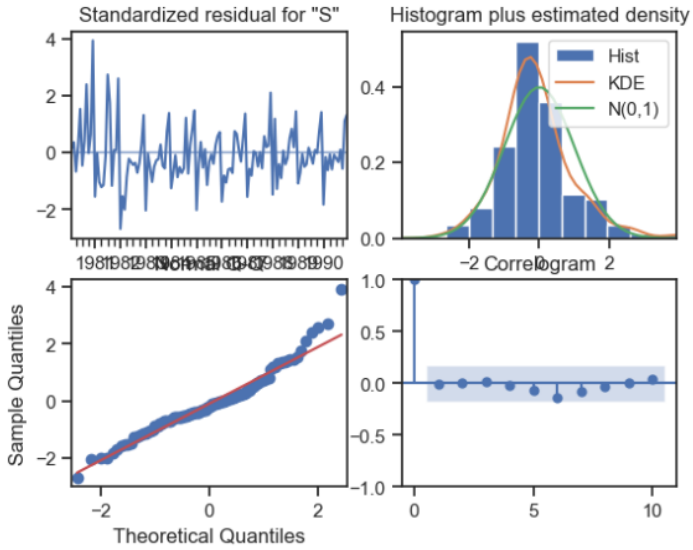


The SARIMAX results for manual arima model

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Sales   No. Observations:                  132
Model:                 ARIMA(2, 1, 2)   Log Likelihood                -635.935
Date:                Thu, 11 Apr 2024   AIC                           1281.871
Time:                        21:33:55   BIC                           1296.247
Sample:                  01-01-1980   HQIC                          1287.712
                       - 12-01-1990
Covariance Type:                 opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.4540      0.469     -0.969      0.333      -1.372       0.464
ar.L2          0.0001      0.170      0.001      0.999      -0.334       0.334
ma.L1         -0.2541      0.459     -0.554      0.580      -1.154       0.646
ma.L2         -0.5984      0.430     -1.390      0.164      -1.442       0.245
sigma2       952.1601     91.424     10.415      0.000     772.973    1131.347
===================================================================================
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):                34.16
Prob(Q):                              0.88   Prob(JB):                         0.00
Heteroskedasticity (H):               0.37   Skew:                             0.79
Prob(H) (two-sided):                  0.00   Kurtosis:                         4.94
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

The plot diagonastics



The RMSE value is

	Test RMSE
Linear Regression	51.080941
Naive Model	79.304391
Simple Average Model	53.049755
2pointTrailingMovingAverage	11.589082
4pointTrailingMovingAverage	14.506190
6pointTrailingMovingAverage	14.558008
9pointTrailingMovingAverage	14.797139
Alpha=0.1,SimpleExponentialSmoothing	36.429535
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	36.510010
Alpha=0.08621,Beta=1.3722,Gamma=0.4763,TrippleExponentialSmoothing_Auto_Fit	37.192624
Alpha=0.2,Beta=0.7,Gamma=0.2,TripleExponentialSmoothing	8.992350
Auto_ARIMA	36.415314
(3,1,1),(3,0,2,12),Auto_SARIMA	18.534956
ARIMA(3,1,3)	36.473225

MANUAL SARIMA MODEL

The SARIMAX results

SARIMAX Results

Dep. Variable:	y	No. Observations:	132
Model:	SARIMAX(2, 1, 2)x(2, 1, 2, 12)	Log Likelihood	-538.016
Date:	Thu, 11 Apr 2024	AIC	1094.031
Time:	21:34:09	BIC	1119.044
Sample:	0	HQIC	1104.188
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.5493	0.228	-2.410	0.016	-0.996	-0.103
ar.L2	-0.0744	0.099	-0.752	0.452	-0.268	0.119
ma.L1	-0.1702	0.216	-0.787	0.431	-0.594	0.254
ma.L2	-0.6695	0.228	-2.939	0.003	-1.116	-0.223
ar.S.L12	-1.0137	0.524	-1.935	0.053	-2.040	0.013
ar.S.L24	-0.1004	0.175	-0.573	0.567	-0.444	0.243
ma.S.L12	0.2914	49.414	0.006	0.995	-96.558	97.141
ma.S.L24	-0.7078	35.084	-0.020	0.984	-69.472	68.056
sigma2	430.2727	2.11e+04	0.020	0.984	-4.08e+04	4.17e+04

Ljung-Box (L1) (Q):	0.02	Jarque-Bera (JB):	27.15
Prob(Q):	0.90	Prob(JB):	0.00
Heteroskedasticity (H):	0.33	Skew:	0.26
Prob(H) (two-sided):	0.00	Kurtosis:	5.28

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

The RMSE values for the models

	Test RMSE
Linear Regression	51.080941
Naive Model	79.304391
Simple Average Model	53.049755
2pointTrailingMovingAverage	11.589082
4pointTrailingMovingAverage	14.506190
6pointTrailingMovingAverage	14.558008
9pointTrailingMovingAverage	14.797139
Alpha=0.1,SimpleExponentialSmoothing	36.429535
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	36.510010
Alpha=0.08621,Beta=1.3722,Gamma=0.4763,TrippleExponentialSmoothing_Auto_Fit	37.192624
Alpha=0.2,Beta=0.7,Gamma=0.2,TripleExponentialSmoothing	8.992350
Auto_ARIMA	36.415314
(3,1,1),(3,0,2,12),Auto_SARIMA	18.534956
ARIMA(3,1,3)	36.473225
(2,1,2)(2,1,2,12),Manual_SARIMA	14.977177

Now we sort the above table in increasing order of RMSE values

	Test RMSE
Alpha=0.2,Beta=0.7,Gamma=0.2,TripleExponentialSmoothing	8.992350
2pointTrailingMovingAverage	11.589082
4pointTrailingMovingAverage	14.506190
6pointTrailingMovingAverage	14.558008
9pointTrailingMovingAverage	14.797139
(2,1,2)(2,1,2,12),Manual_SARIMA	14.977177
(3,1,1),(3,0,2,12),Auto_SARIMA	18.534956
Auto_ARIMA	36.415314
Alpha=0.1,SimpleExponentialSmoothing	36.429535
ARIMA(3,1,3)	36.473225
Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing	36.510010
Alpha=0.08621,Beta=1.3722,Gamma=0.4763,TrippleExponentialSmoothing_Auto_Fit	37.192624
Linear Regression	51.080941
Simple Average Model	53.049755
Naive Model	79.304391

- From the above table we understood that the ‘Triple exponential smoothing’ gives less RMSE value i.e 8.992

Now we build the most optimum model on the complete data and predict 12 months into the future with appropriate confidence intervals/bands.

The sales predictions after applying the smoothing is as follows

	Sales_Predictions
1995-08-01	1961.135966
1995-09-01	2755.845496
1995-10-01	4350.148046
1995-11-01	6467.199378
1995-12-01	11562.330237
1996-01-01	3274.957784
1996-02-01	4445.687620
1996-03-01	5816.912912
1996-04-01	6431.148531
1996-05-01	6324.013903
1996-06-01	6307.062803
1996-07-01	8098.263479

Now we calculate the upper and lower confidence bands at 95% confidence level

	lower_CI	prediction	upper_ci
1995-08-01	716.392152	1961.135966	3205.879779
1995-09-01	1511.101682	2755.845496	4000.589309
1995-10-01	3105.404233	4350.148046	5594.891860
1995-11-01	5222.455564	6467.199378	7711.943191
1995-12-01	10317.586423	11562.330237	12807.074050

The final optimum forecasted plot is as follows