Time Series Forecasting-Rose Wine

Problem Statement:

ABC Estate Wines has been a leader in the rose wine industry for many years, offering high-quality wines to consumers all around the world. As the company continues to expand its reach and grow its customer base, it is essential to analyze market trends and forecast future sales to ensure continued success.

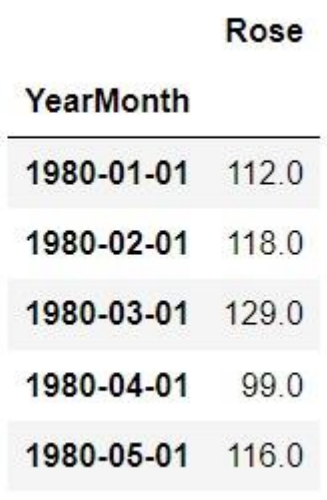
In this report, we will focus on analyzing the sales data for rose wine in the 20th century. As an analyst for ABC Estate Wines, I have been tasked with reviewing this data to identify patterns, trends, and opportunities for growth in the wine market.This knowledge will help us to make informed decisions about how to position our products in the market, optimize our sales strategies, and forecast future sales trends.

Overall, this report aims to provide valuable insights into the wine market and how ABC Estate Wines can continue to succeed in this highly competitive industry.

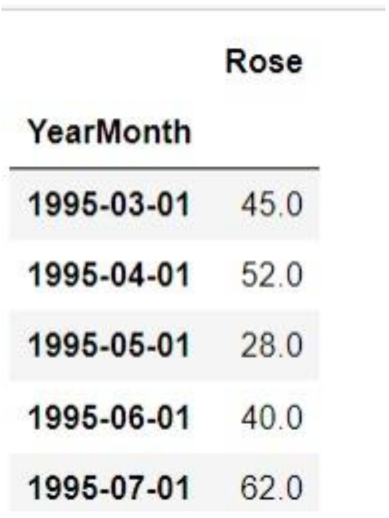
1. Read the data as an appropriate Time Series data and plot the data.

Data set is read using the pandas library.

Rows of data set;



	Rose
YearMonth	
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

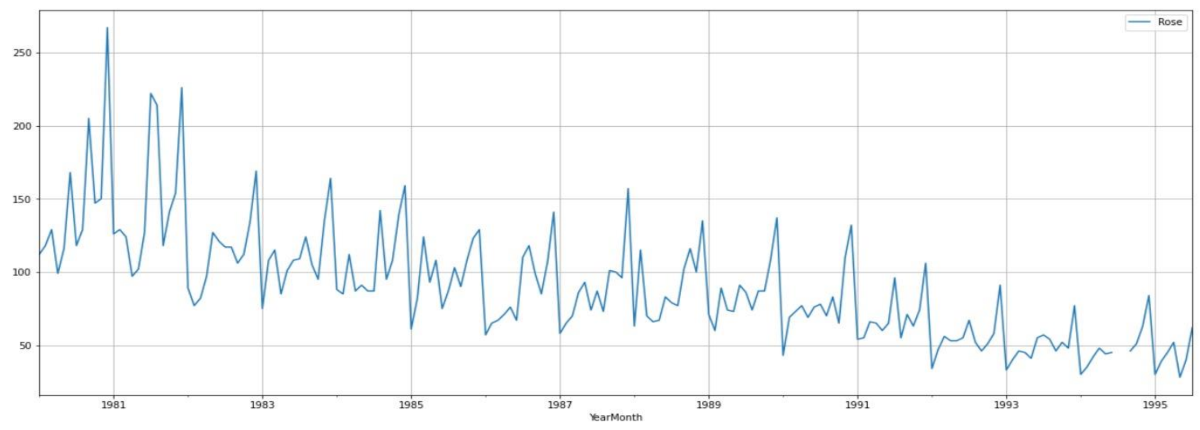


	Rose
YearMonth	
1995-03-01	45.0
1995-04-01	52.0
1995-05-01	28.0
1995-06-01	40.0
1995-07-01	62.0

Number of Rows and Columns of Dataset:

The dataset has 187 rows and 1 column.

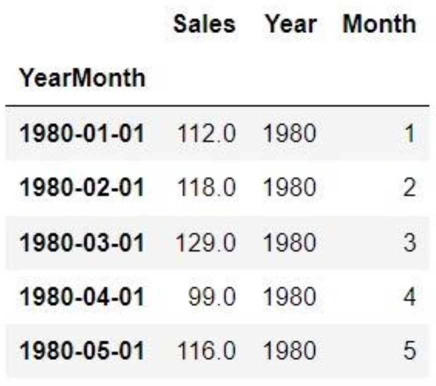
Plot of the dataset:



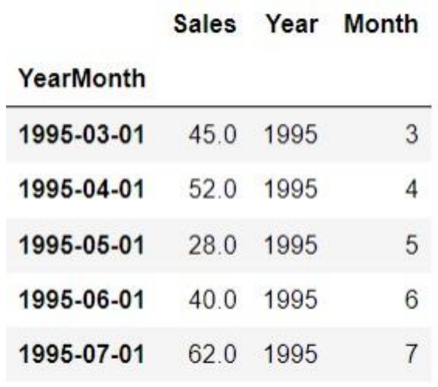
Post Ingestion of Dataset:

We have divided the dataset further by extraction month and year columns from the YearMonth column and renamed the sparkling column name to Sales for better analysis od the dataset.

Rows of new data set;



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



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

Number of Rows and Columns of Dataset: The dataset has 187 rows and 3 column.

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

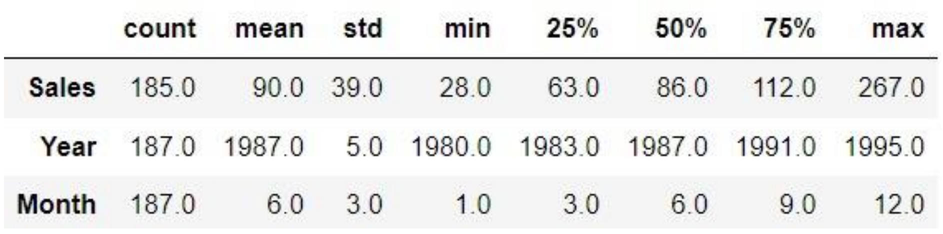
Data Type;

Index: DateTime

Sales: integer

Month: integer Year: integer

Statistical summary:

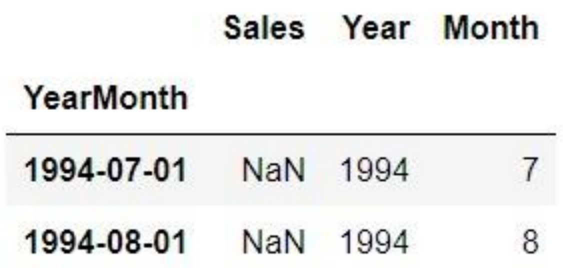


	count	mean	std	min	25%	50%	75%	max
Sales	185.0	90.0	39.0	28.0	63.0	86.0	112.0	267.0
Year	187.0	1987.0	5.0	1980.0	1983.0	1987.0	1991.0	1995.0
Month	187.0	6.0	3.0	1.0	3.0	6.0	9.0	12.0

Null Value:

There are 2 null values present in sales the dataset.

We found the values for the months of July & August were missing for the year 1994.



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

We tried following approaches to impute the data, these were as below.

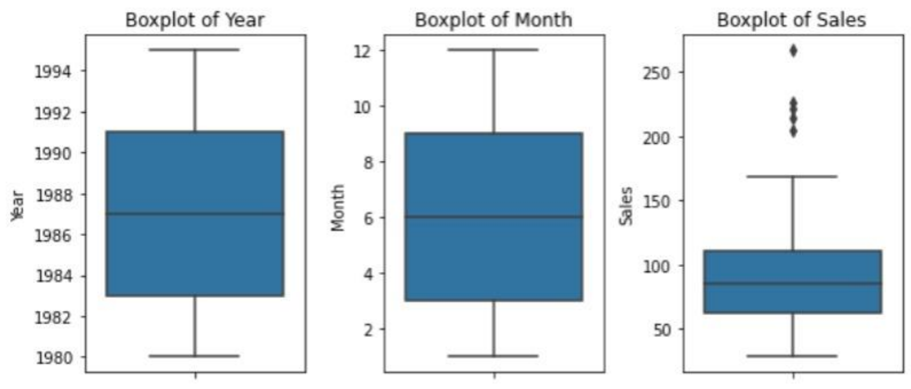
Mean - Before & After

Treating null values is very important to do further analysis.

In this approach, instead of taking means for the 7th months across all the years, we just took mean of the 7th months values from a year before and a year after the missing value.

Similar steps were taken for 8th month.

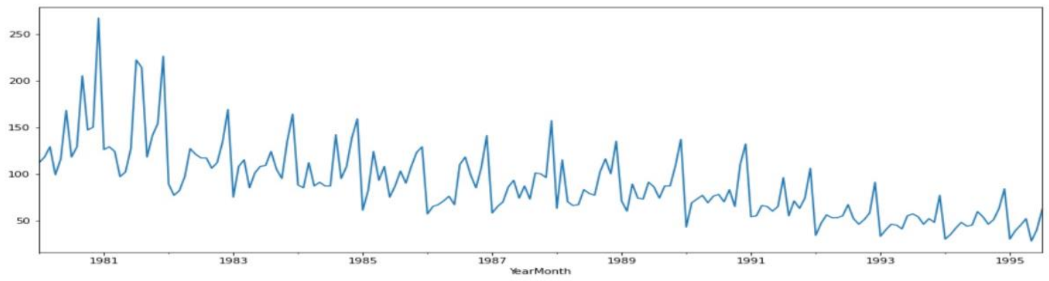
Boxplot of dataset:



The box plot shows:

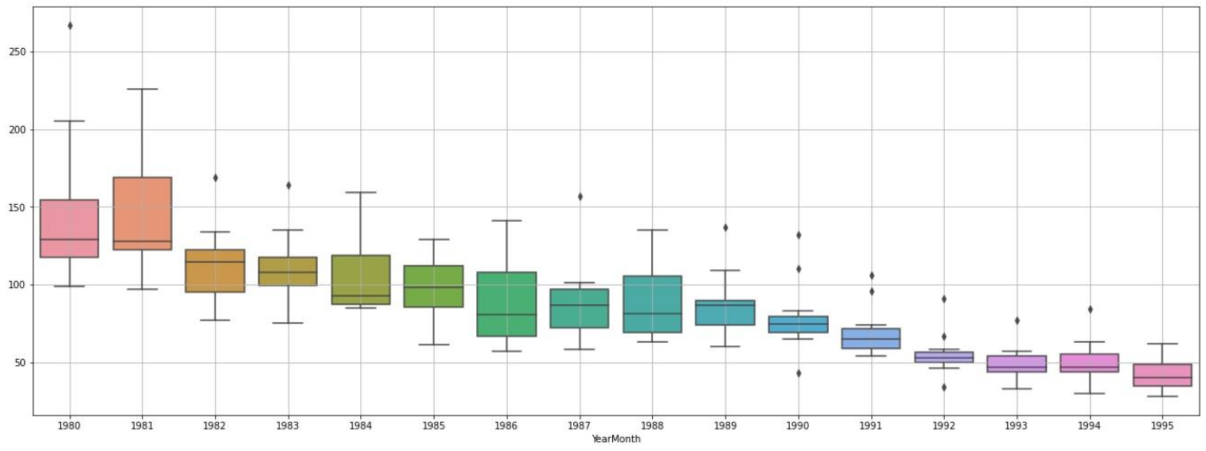
● Sales boxplot has outliers we can treat them but we are choosing not to treat them as they do not give much effect on the time series model.

Line plot of sales:



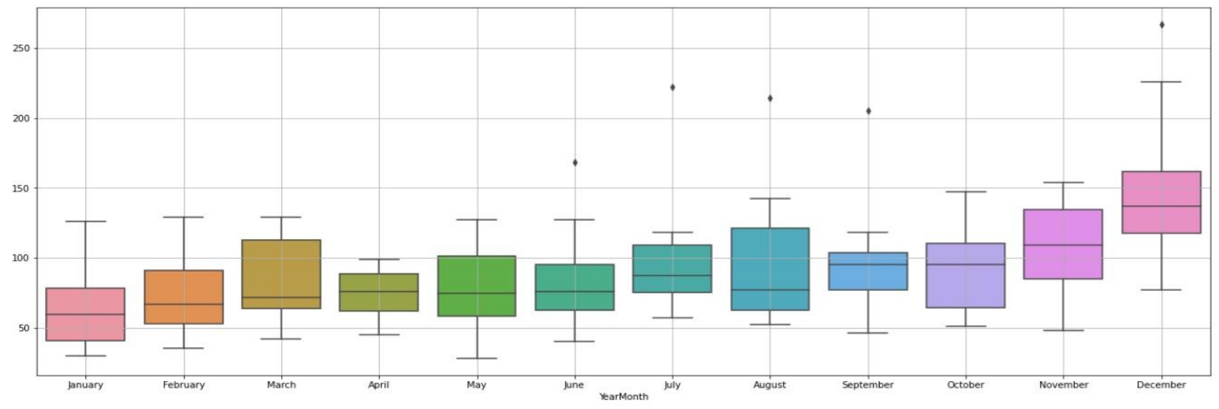
The line plot shows the patterns of trend and seasonality and also shows that there was a peak in the year 1981.

Boxplot Yearly:



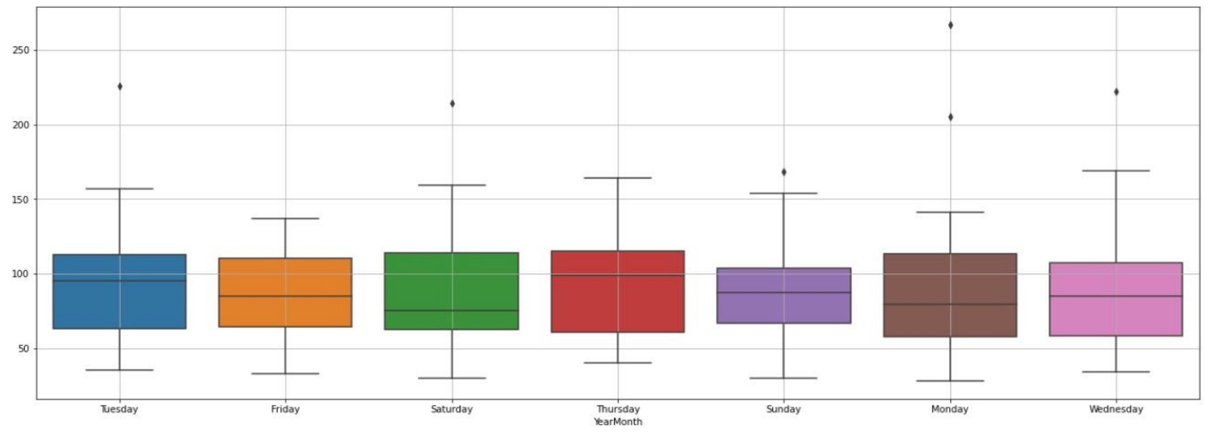
This yearly box plot shows there is consistency over the years and there was a peak in 1980-1981. Outliers are present in almost all years.

Boxplot Monthly:



The plot shows that sales are highest in the month of December and lowest in the month of January. Sales are consistent from January to July then from august the sales start to increase. Outliers are present in June, July, august, September and December.

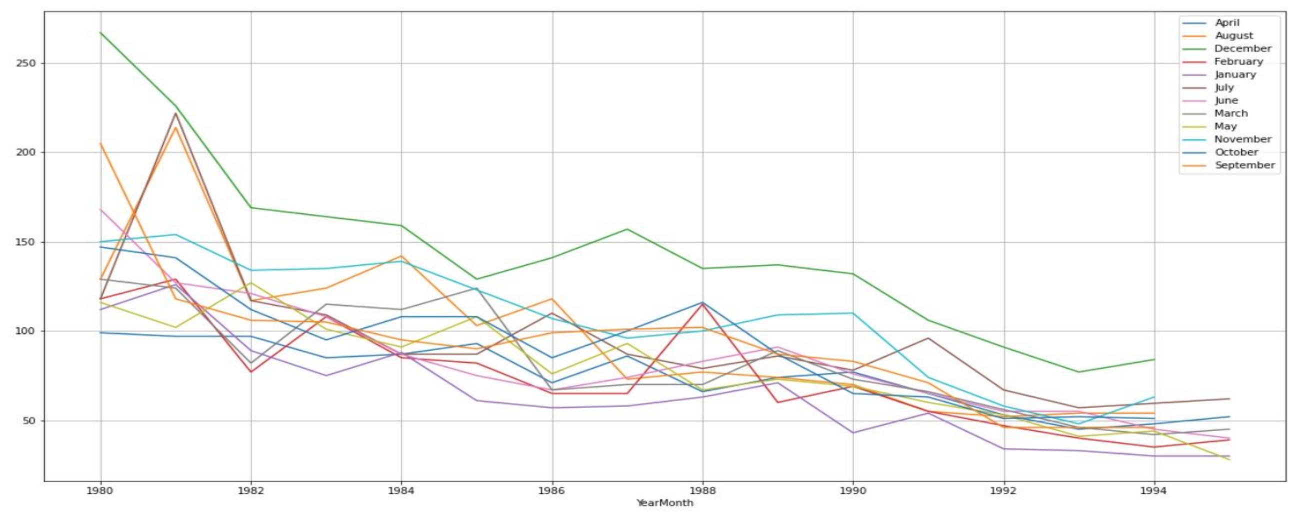
Boxplot Weekday vise:



Tuesday has more sales than other days and Wednesday has the lowest sales of the week.

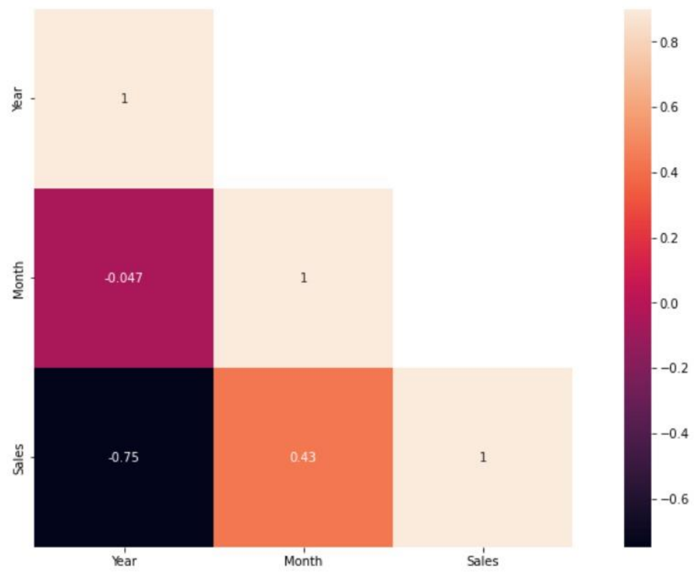
Outliers are present on all days except Friday and Thursday.

Graph of Monthly Sales over the years:



This plot shows that December has the highest sales over the years and the year 1981 was the year with the highest number of sales.

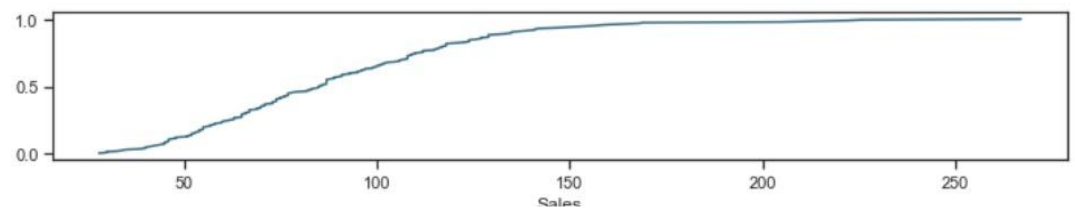
Correlation plot



This heat map shows that there was little correlation between Sales and the Years data, there significantly more correlation between the month and Sales columns. Clearly indicating a seasonal pattern in our Sales data. Certain months have higher sales, while certain months have lesser.

Plot ECDF: Empirical Cumulative Distribution Function

This graph shows the distribution of data.



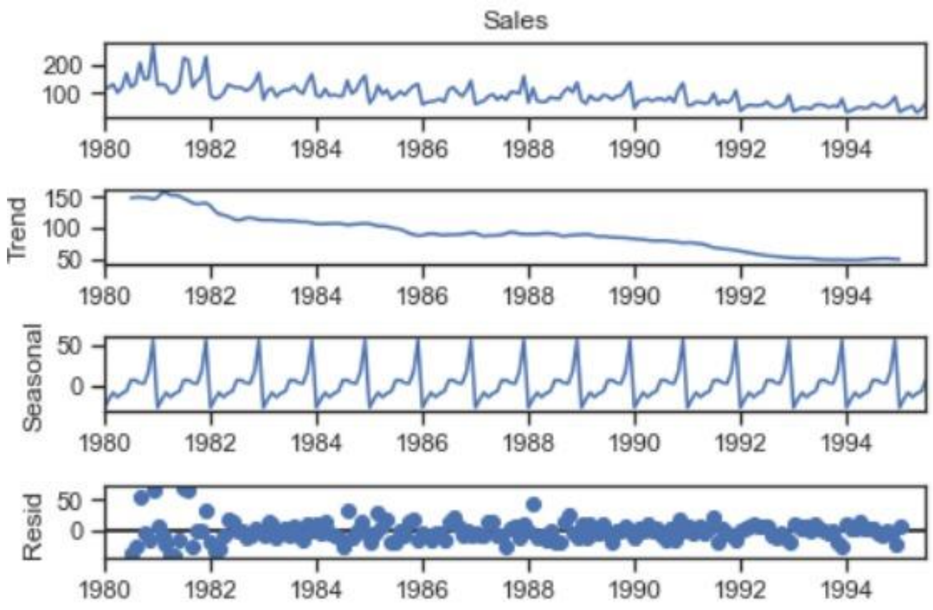
This plot shows:

● 50% sales has been less 100

● Highest vales is 250

● Aprox 90% sales has been less than 150

Decomposition –Additive



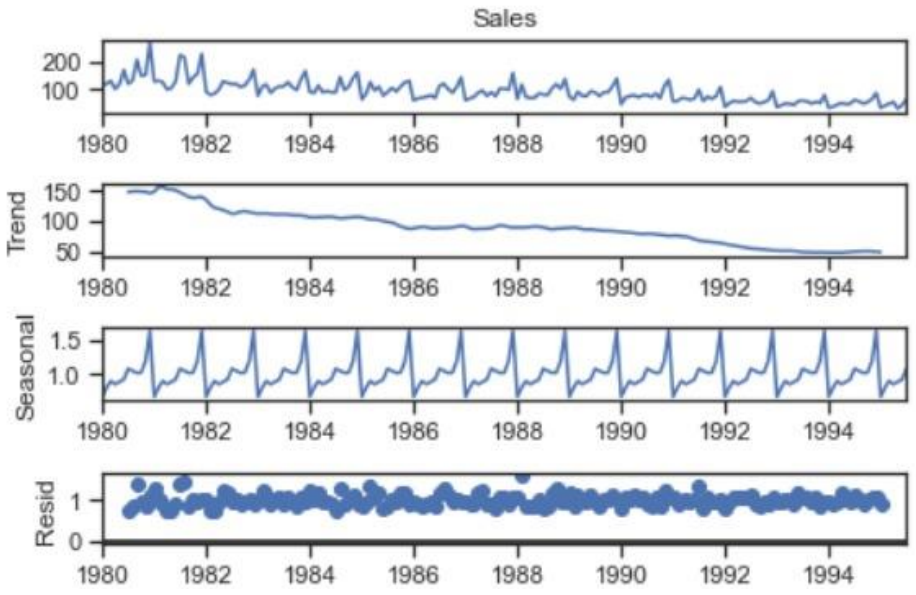
The plots show:

● Peak year 1981

● It also shows that the trend has declined over the year after 1981 ● Residue is spread and is not in a straight line.

● Both trend and seasonality are present.

Decomposition-Multiplicative



The plots show:

● Peak year 1981

● It also shows that the trend has declined over the year after 1981.

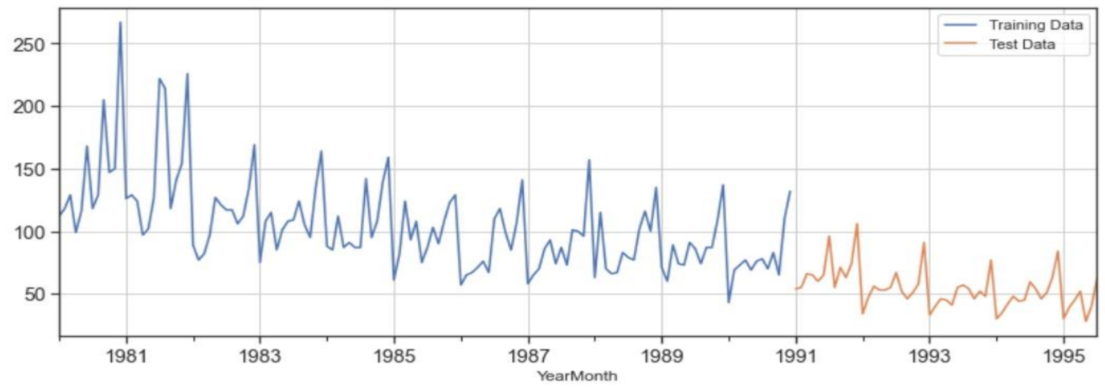
● Residue is spread and is in approx a straight line.

● Both trend and seasonality are present.

● Reside is 0 to 1, while for additive is 0 to 50.

● So multiplicative model is selected owing to a more stable residual plot and lower range of residuals.

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

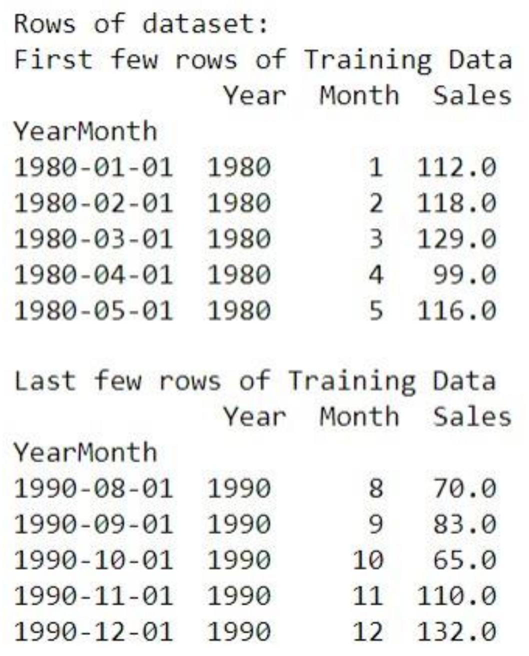


Data split from 1980-1990 is training data, then 1991 to 1995 is training data.

Rows and Columns:

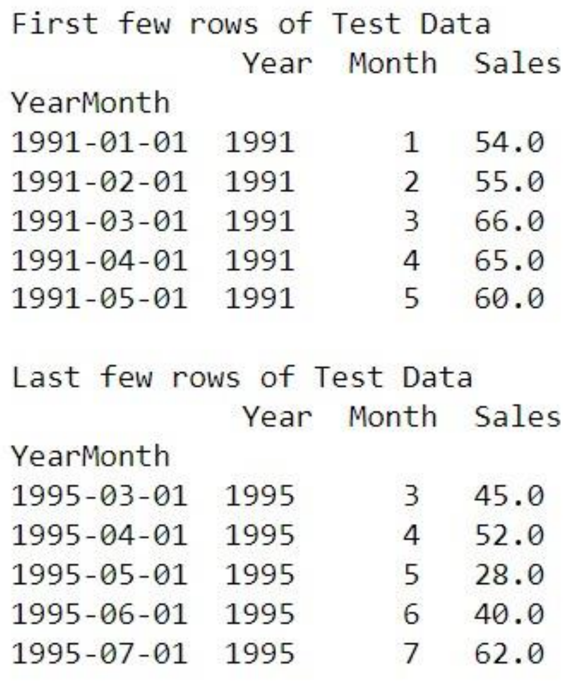
train dataset has 132 rows and 3 columns. test dataset has 55 and 3 columns.

Few Rows of datasets:



```
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
```

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

● Model 1:Linear Regression

● Model 2: Naive Approach

● Model 3: Simple Average

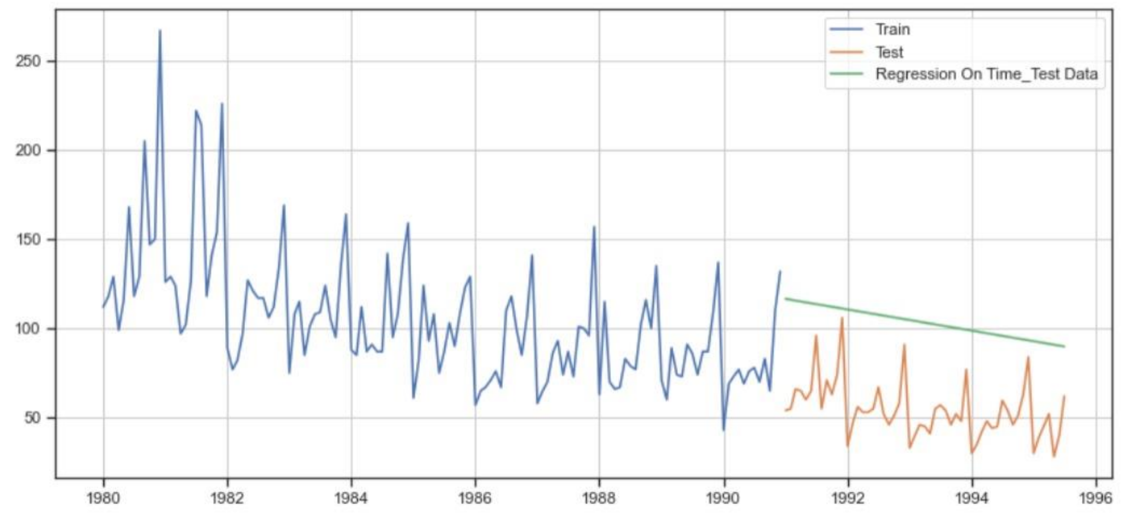
● Model 4: Moving Average(MA)

● Model 5: Simple Exponential Smoothing

● Model 6: Double Exponential Smoothing (Holt's Model)

● Model 7: Triple Exponential Smoothing (Holt - Winter's Model)

Model 1: Linear Regression

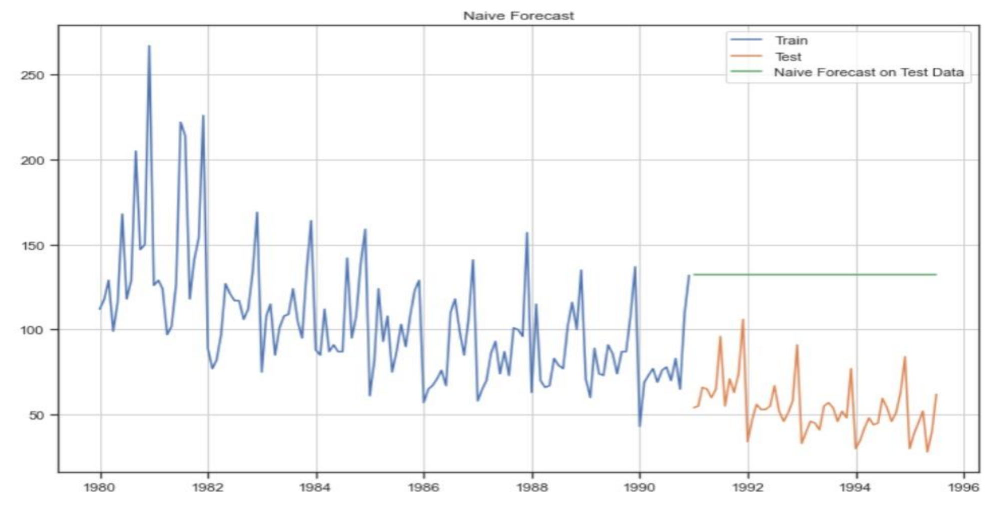


The green line indicates the predictions made by the model, while the orange values are the actual test values. It is clear the predicted values are very far off from the actual values

Model was evaluated using the RMSE metric. Below is the RMSE calculated for this model.

Linear Regression 51.080941

Model 2: Naive Approach:

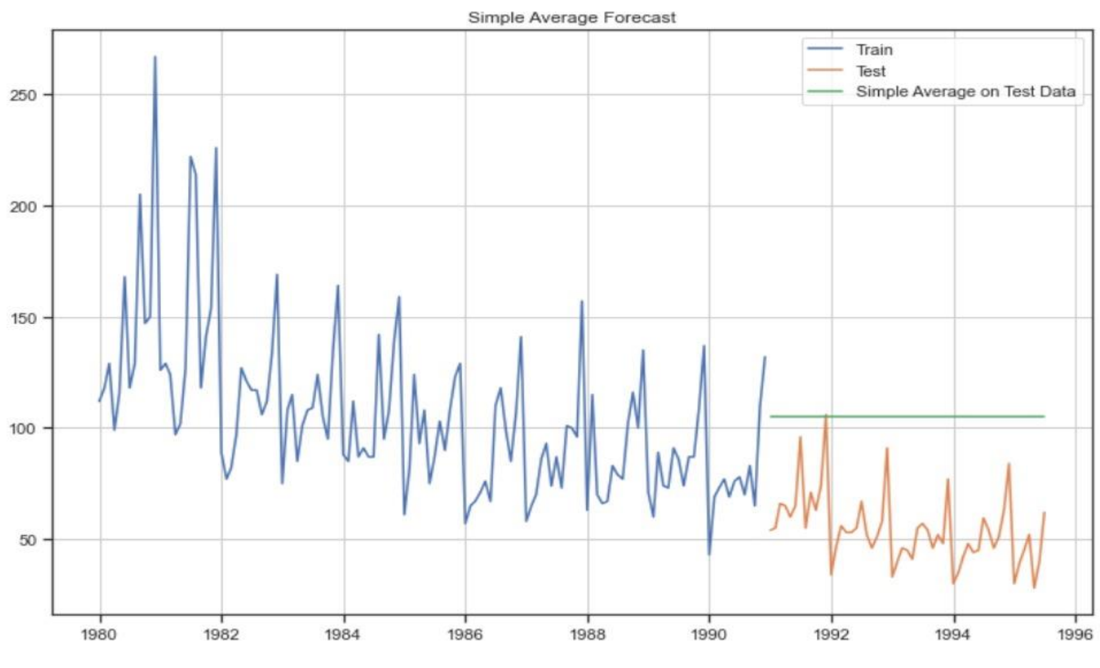


The green line indicates the predictions made by the model, while the orange values are the actual test values. It is clear the predicted values are very far off from the actual values

Model was evaluated using the RMSE metric. Below is the RMSE calculated for this model.

Naive Model 79.304391

Method 3: Simple Average

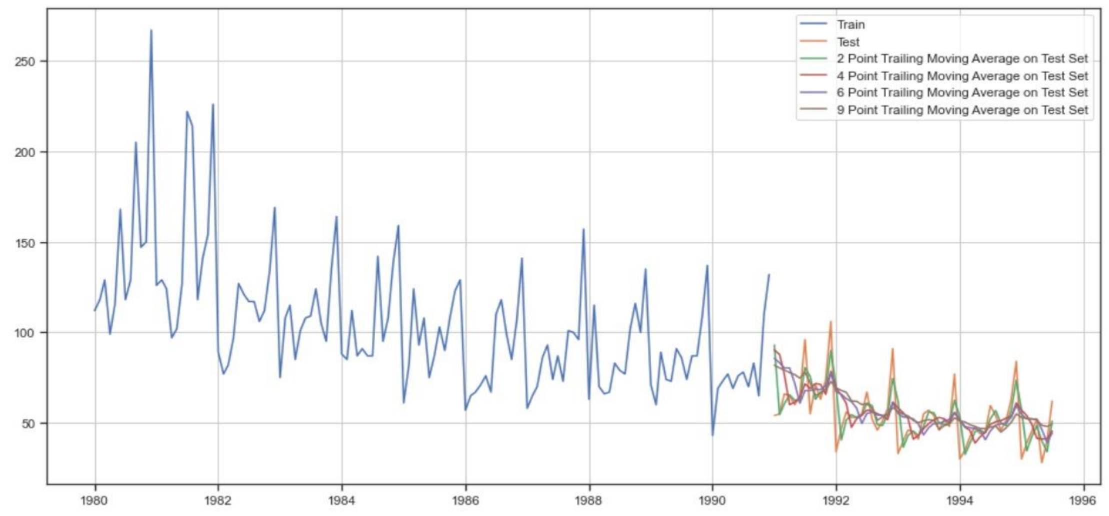


The green line indicates the predictions made by the model, while the orange values are the actual test values. It is clear the predicted values are very far off from the actual values

Model was evaluated using the RMSE metric. Below is the RMSE calculated for this model.

Simple Average Model 53.049755

Method 4: Moving Average(MA)



Model was evaluated using the RMSE metric. Below is the RMSE calculated for this model.

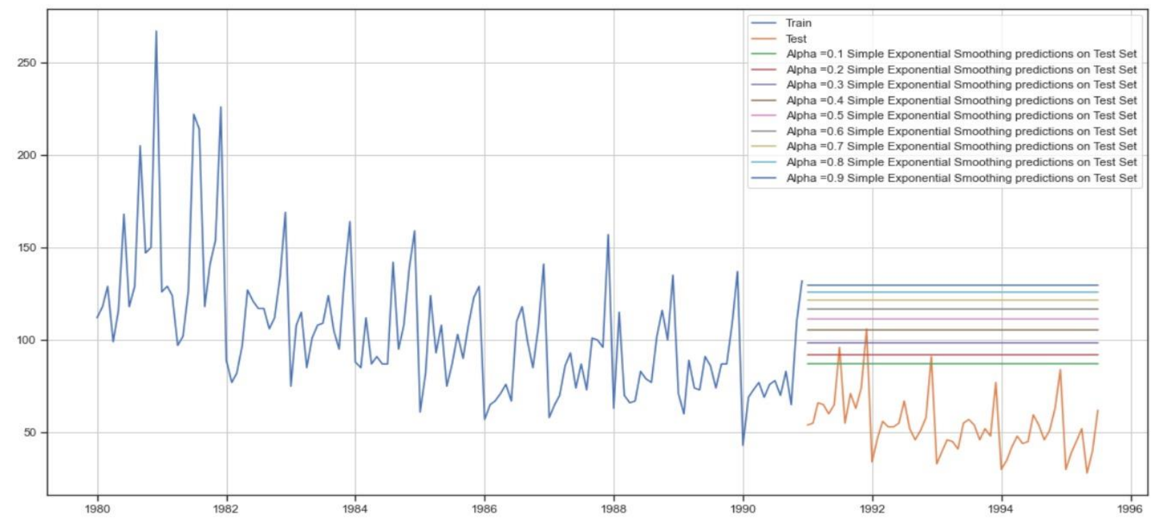
2 pointTrailingMovingAverage 11.589082 4 pointTrailingMovingAverage 14.506190

6 pointTrailingMovingAverage 14.558008

9 pointTrailingMovingAverage 14.797139

We created multiple moving average models with rolling windows varying from 2 to 9. Rolling average is a better method than simple average as it takes into account only the previous n values to make the prediction, where n is the rolling window defined. This takes into account the recent trends and is in general more accurate. Higher the rolling window, smoother will be its curve, since more values are being taken into account.

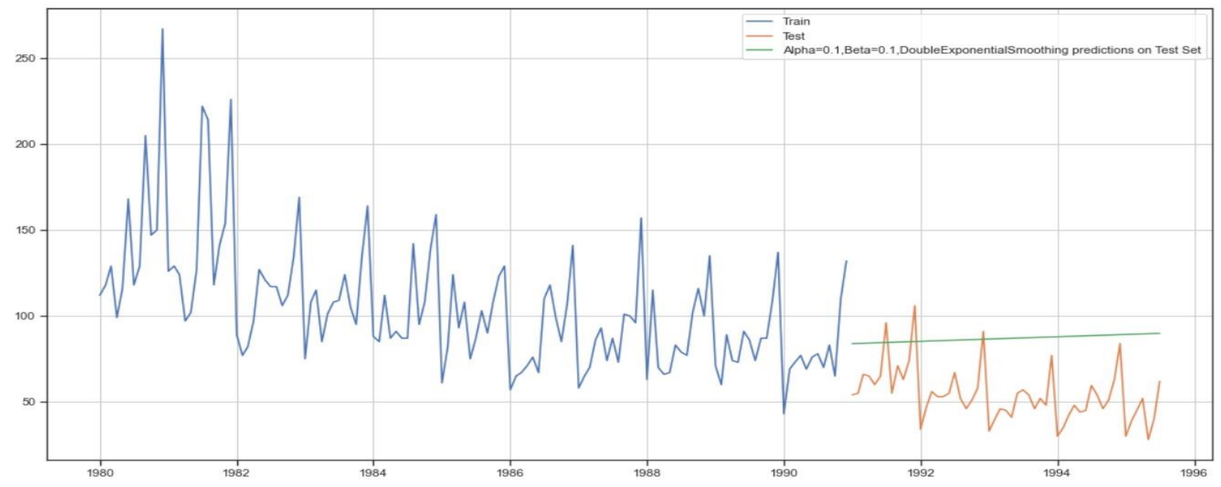
Method 5: Simple Exponential Smoothing



Model was evaluated using the RMSE metric. Below is the RMSE calculated for this model.

Alpha=0.1,SimpleExponentialSmoothing 36.429535

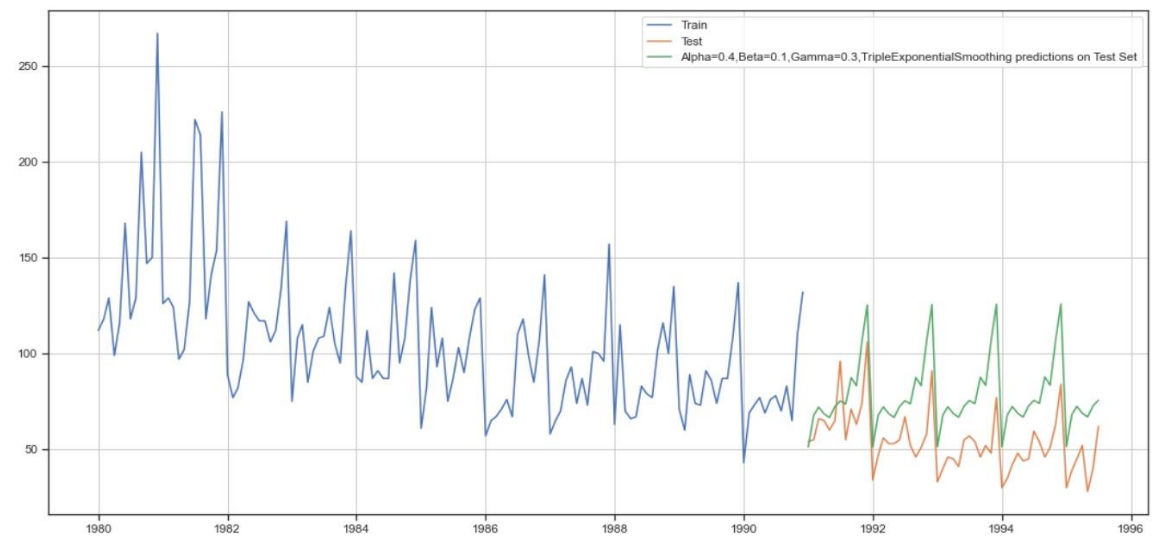
Method 6: Double Exponential Smoothing (Holt's Model)



Model was evaluated using the RMSE metric. Below is the RMSE calculated for this model.

Alpha Value = 0.1, beta value = 0.1, DoubleExponentialSmoothing36.510010

Method 7: Triple Exponential Smoothing (Holt - Winter's Model)



Output for best alpha, beta and gamma values is shown by the green color line in the above plot. Best model had both multiplicative trend as well as seasonality.

So far this is the best model

Model was evaluated using the RMSE metric. Below is the RMSE calculated for this model.

Alpha=0.4,Beta=0.1,Gamma=0.3,TripleExponentialSmoothing 8.992350

5 Check for the stationarity of the data on which the model is being built on using appropriate statistical tests and also mention the hypothesis for the statistical test. If the data is found to be non-stationary, take appropriate steps to make it stationary. Check the new data for stationarity and comment. Note: Stationarity should be checked at alpha = 0.05.

Check for stationarity of the whole Time Series data.

The Augmented Dickey-Fuller test is an unit root test which determines whether there is a unit root and subsequently whether the series is non-stationary.

The hypothesis in a simple form for the ADF test is:

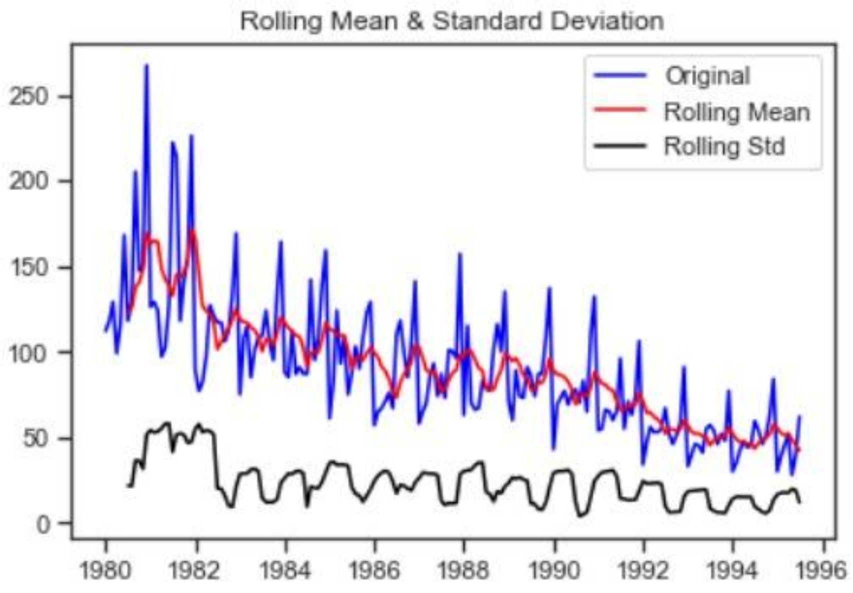
● H0 : The Time Series has a unit root and is thus non-stationary.

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

We would want the series to be stationary for building ARIMA models and thus we would want the p-value of this test to be less than the α value.

We see that at 5% significant level the Time Series is non-stationary.

Plot 20: dickey fuller test

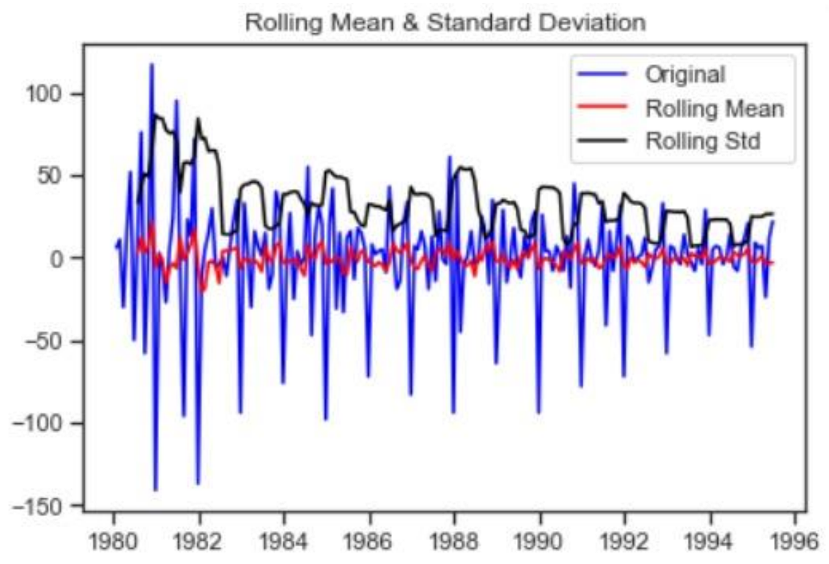


Results of Dickey-Fuller Test:

Test Statistic -1.892338 p-value 0.335674

we failed to reject the null hypothesis, which implies the Series is not stationary in nature.

In order to try and make the series stationary we used the differencing approach. We used .diff() function on the existing series without any argument, implying the default diff value of 1 and also dropped the NaN values, since differencing of order 1 would generate the first value as NaN which need to be dropped



Results of Dickey-Fuller Test:

Test Statistic -8.032729e+00 p-value 1.938803e-12

the null hypothesis that the series is not stationary at difference = 1 was rejected, which implied that the series has indeed become stationary after we performed the differencing.

We could now proceed ahead with ARIMA/ SARIMA models, since we had made the series stationary.

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

AUTO - ARIMA model

We employed a for loop for determining the optimum values of p,d,q, where p is the order of the AR (Auto-Regressive) part of the model, while q is the order of the MA (Moving Average) part of the model. d is the differencing that is required to make the series stationary. p,q values in the range of (0,4) were given to the for loop, while a fixed value of 1 was given for d, since we had already determined d to be 1, while checking for stationarity using the ADF test.

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)

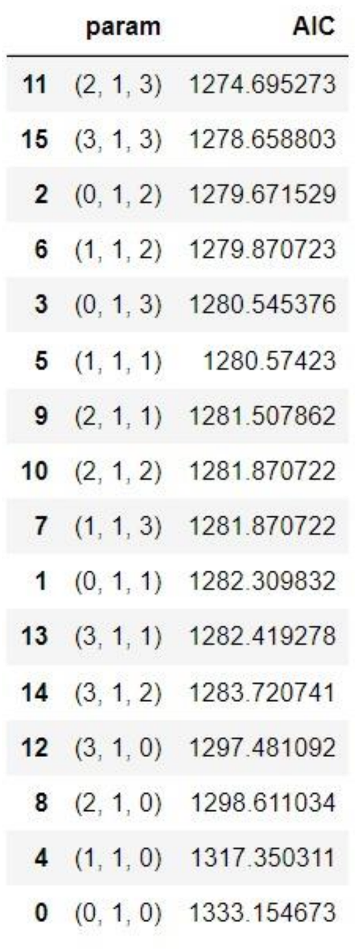
Model: (3, 1, 0)

Model: (3, 1, 1)

Model: (3, 1, 2)

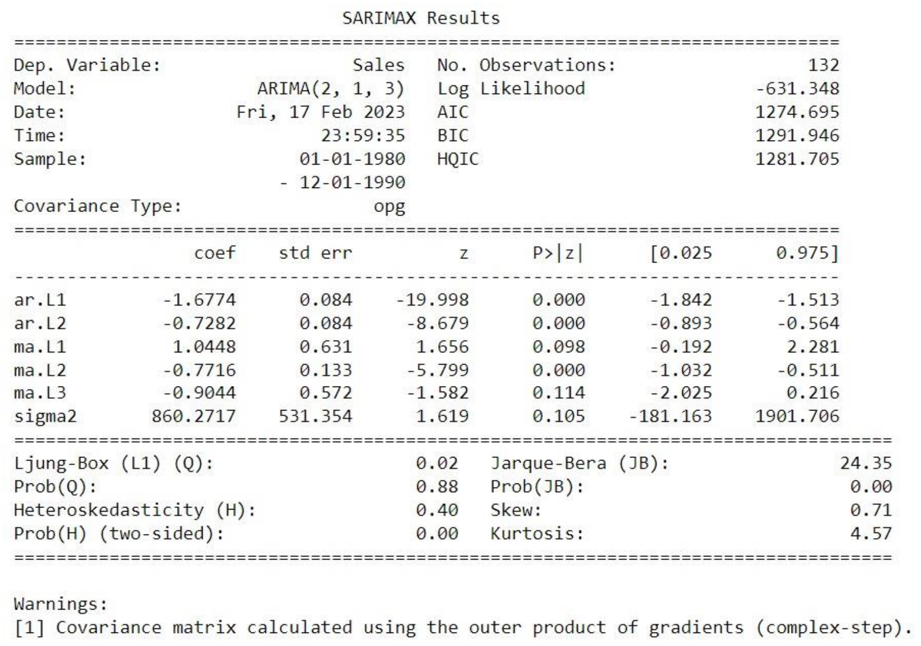
Model: (3, 1, 3)

Akaike information criterion (AIC) value was evaluated for each of these models and the model with least AIC value was selected.



	param	AIC
11	(2, 1, 3)	1274.695273
15	(3, 1, 3)	1278.658803
2	(0, 1, 2)	1279.671529
6	(1, 1, 2)	1279.870723
3	(0, 1, 3)	1280.545376
5	(1, 1, 1)	1280.57423
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

the summary report for the ARIMA model with values (p=2,d=1,q=3).



```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Sales   No. Observations:                  132
Model:                 ARIMA(2, 1, 3)   Log Likelihood                -631.348
Date:                Fri, 17 Feb 2023   AIC                           1274.695
Time:                        23:59:35   BIC                           1291.946
Sample:                    01-01-1980   HQIC                          1281.705
                         - 12-01-1990
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.6774      0.084    -19.998      0.000      -1.842      -1.513
ar.L2         -0.7282      0.084     -8.679      0.000      -0.893      -0.564
ma.L1          1.0448      0.631      1.656      0.098      -0.192       2.281
ma.L2         -0.7716      0.133     -5.799      0.000      -1.032      -0.511
ma.L3         -0.9044      0.572     -1.582      0.114      -2.025       0.216
sigma2       860.2717    531.354      1.619      0.105    -181.163    1901.706
===================================================================================
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):                24.35
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).
```

RMSE values are as below:

36.42079120523518

AUTO- SARIMA Model

A similar for loop like AUTO_ARIMA with below values was employed, resulting in the models shown below.

p = q = range(0, 4) d= range(0,2) D = range(0,2) pdq = list(itertools.product(p, d, q)) model_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, D, q))]

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)

Akaike information criterion (AIC) value was evaluated for each of these models and the model with least AIC value was selected. Here only the top 5 models are shown.