Great Learning
& UT Austin

Time Series Forecasting

Project Submission

Submitted by- Gunjar Fuley
Phone-9938126651
Batch- PGPDSBA Online Nov_A 2020

Problem 1 for the Data Set: Shoesales

You are an analyst in the IJK shoe company and you are expected to forecast the sales of the pairs of shoes for the upcoming 12 months from where the data ends. The data for the pair of shoe sales have been given to you from January 1980 to July 1995.

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

   The data was uploaded to the Jupyter Notebook. The data was retrieved using CSV file. The name of the file was 'Shoe-Sales.csv'. The data has two columns i.e. 'Shoe_Sales' and 'YearMonth'. As understood from name only, the column 'YearMonth' has time therefore it was made as index column while reading the data. Also, parse_dates was used to tell Python that the column 'YearMonth' has time data.

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

   After loading the CSV file, it was understood that the dataset has 187 values, out of which there are 0 NULL values.

YearMonth	Shoe_Sales
1980-01-01	85
1980-02-01	89
1980-03-01	109
1980-04-01	95
1980-05-01	91

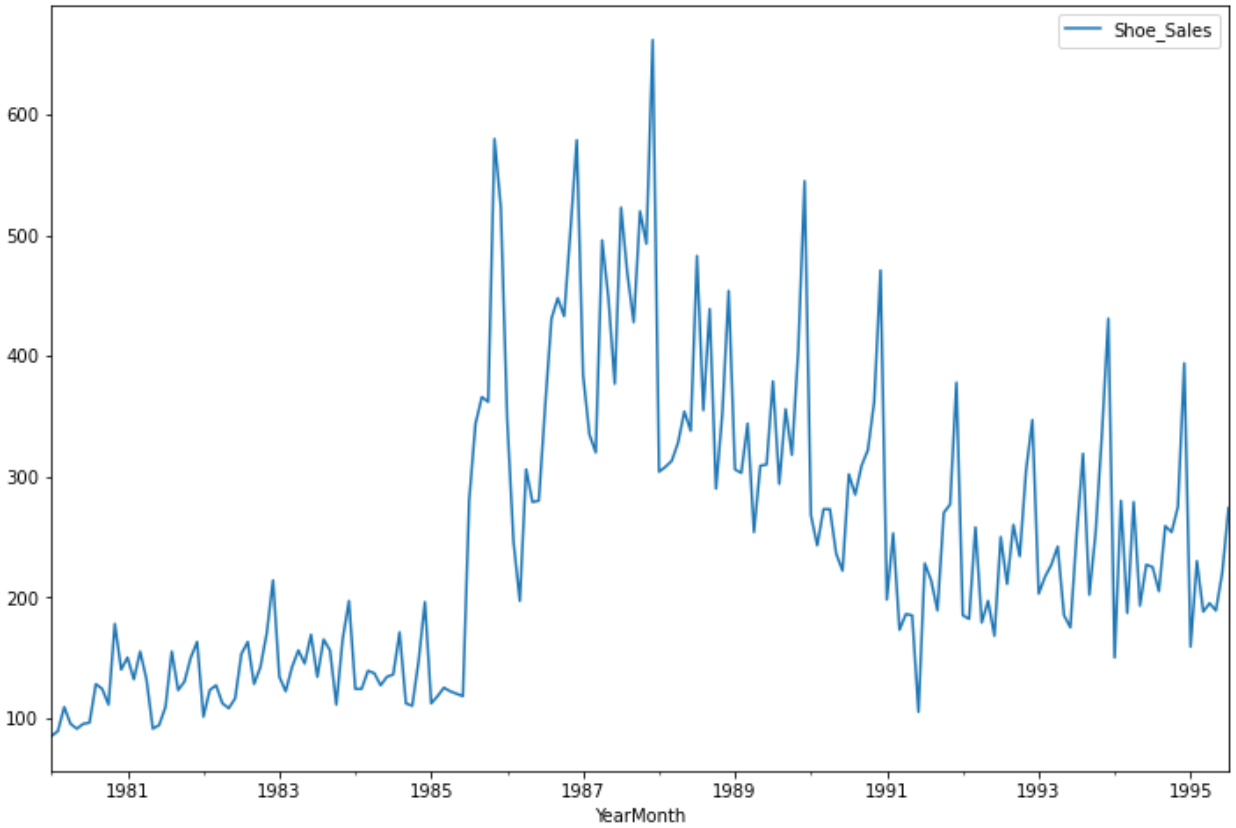
   Above are initial 5 data points.

YearMonth	Shoe_Sales
1995-03-01	188
1995-04-01	195
1995-05-01	189
1995-06-01	220
1995-07-01	274

Above are the last 5 data points.
Here we can easily understand that this is monthly sales data of a shoes company. The data was collected on 1st day of every month from 01-January-1980 to 01-July-1995.

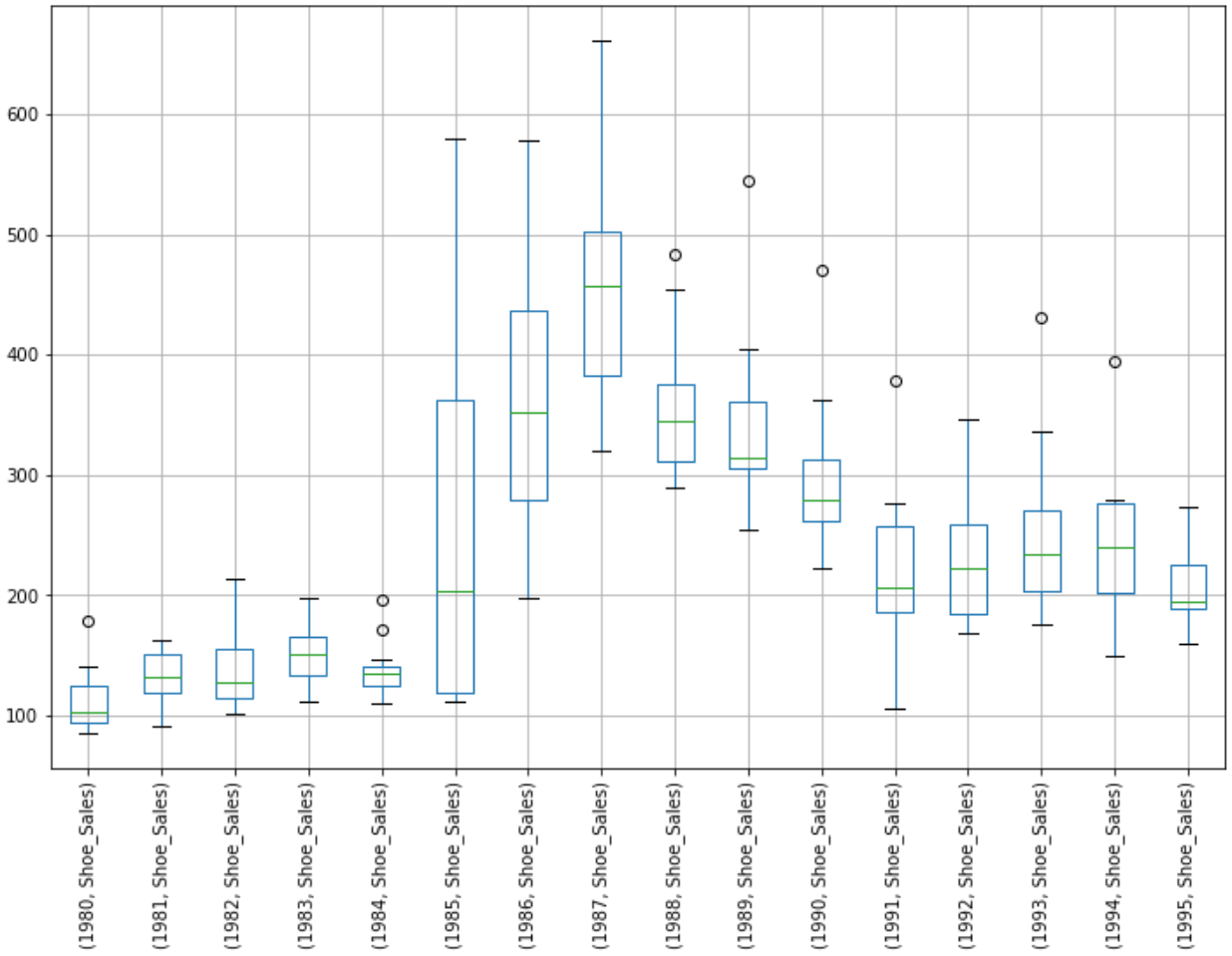
	Shoe_Sales
count	187.000
mean	245.636
std	121.391
min	85.000
25%	143.500
50%	220.000
75%	315.500
max	662.000

Above is the descriptive statistics of the dataset. We can see that the mean sale of 187 months is 245.63. Also, we can see that the minimum value is 85 and maximum value is 662.

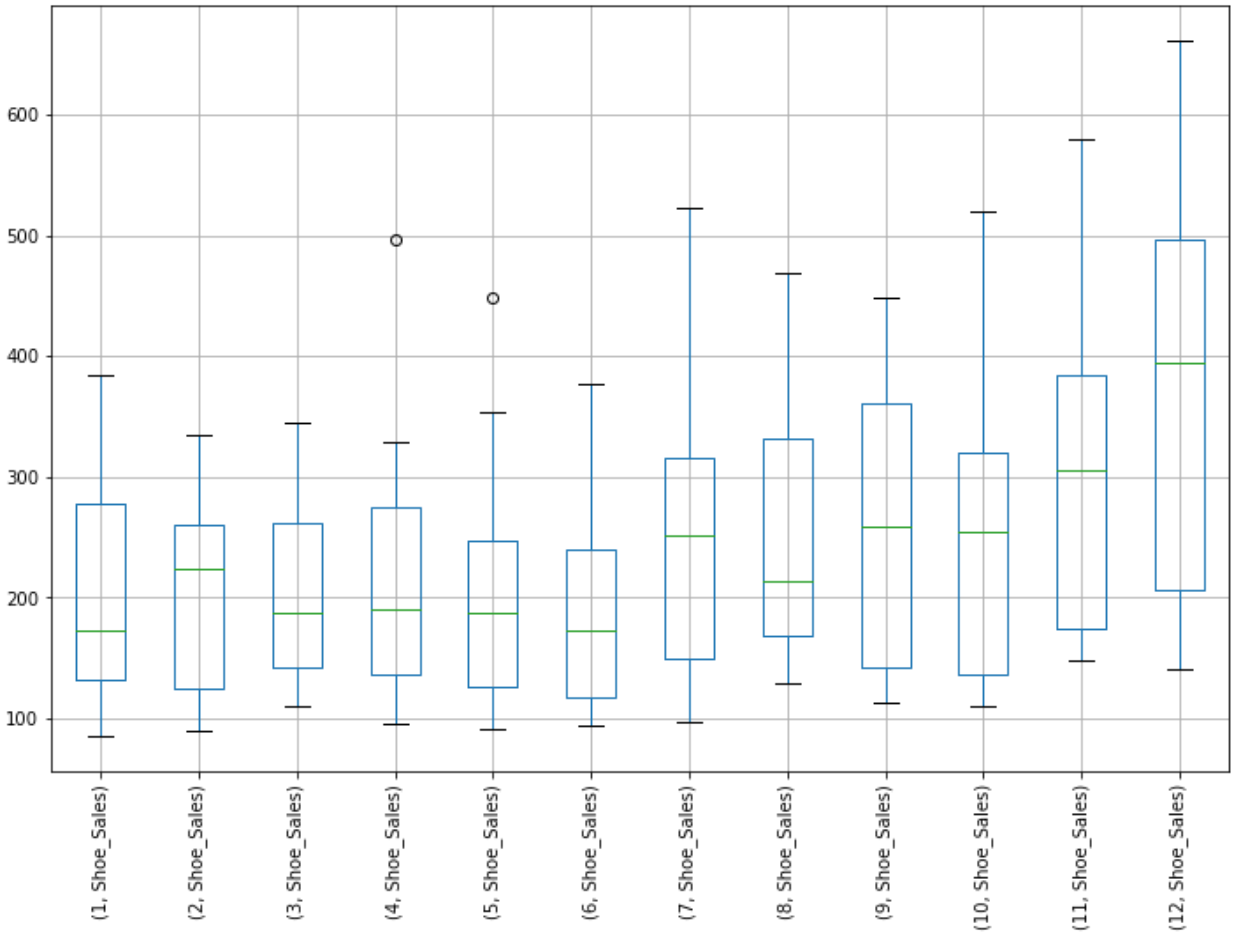


Above is the graph for the dataset. Here we cannot clearly see the trend and seasonality. Also, we can clearly see that the sales have increased during the time period between 1985 and 1991.

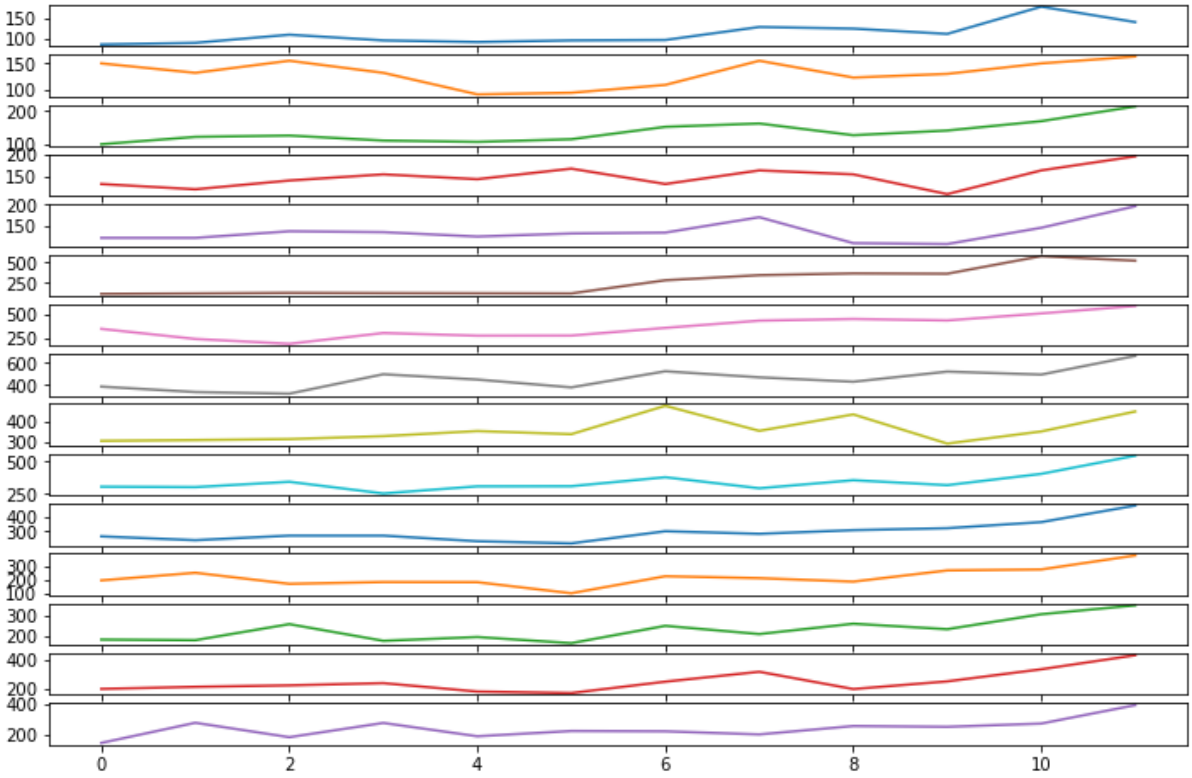
2. Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.



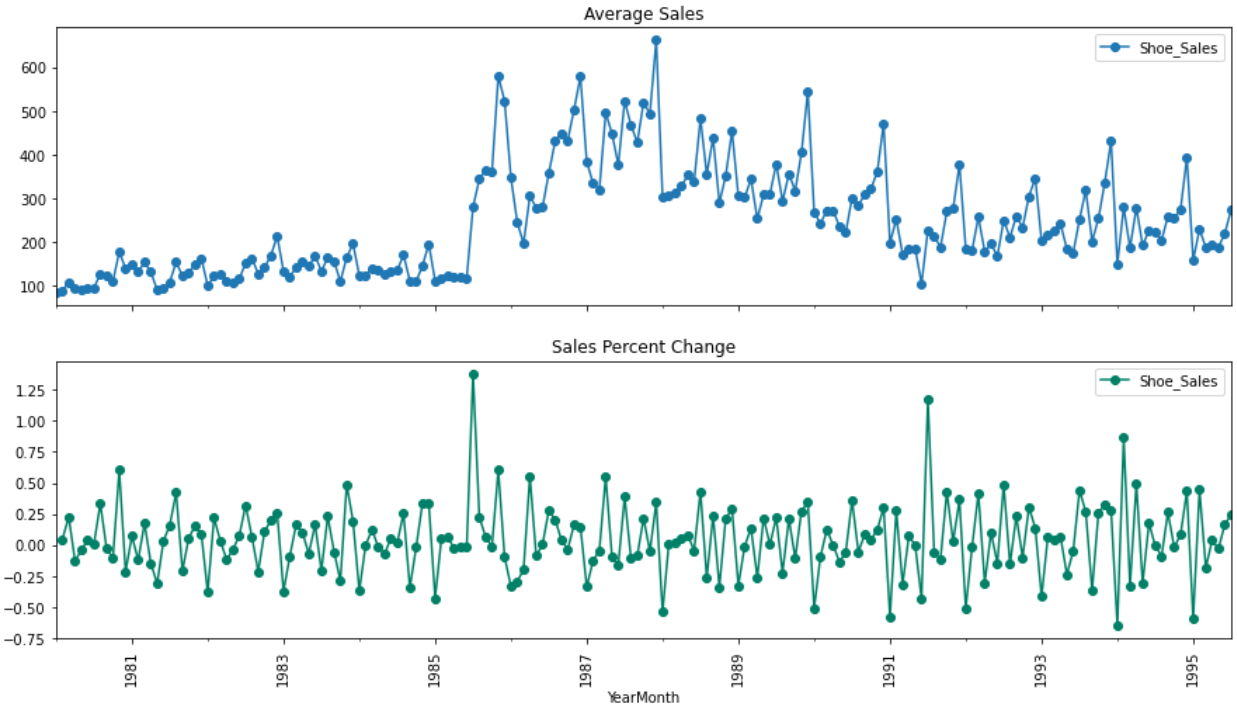
Above is the yearly sales report for the company. Here we can see that the outliers are present for the years 1980, 1984, 1988, 1989, 1990, 1991, 1993 and 1994. The overall shoe sales is highest for year 1987 followed by 1986.



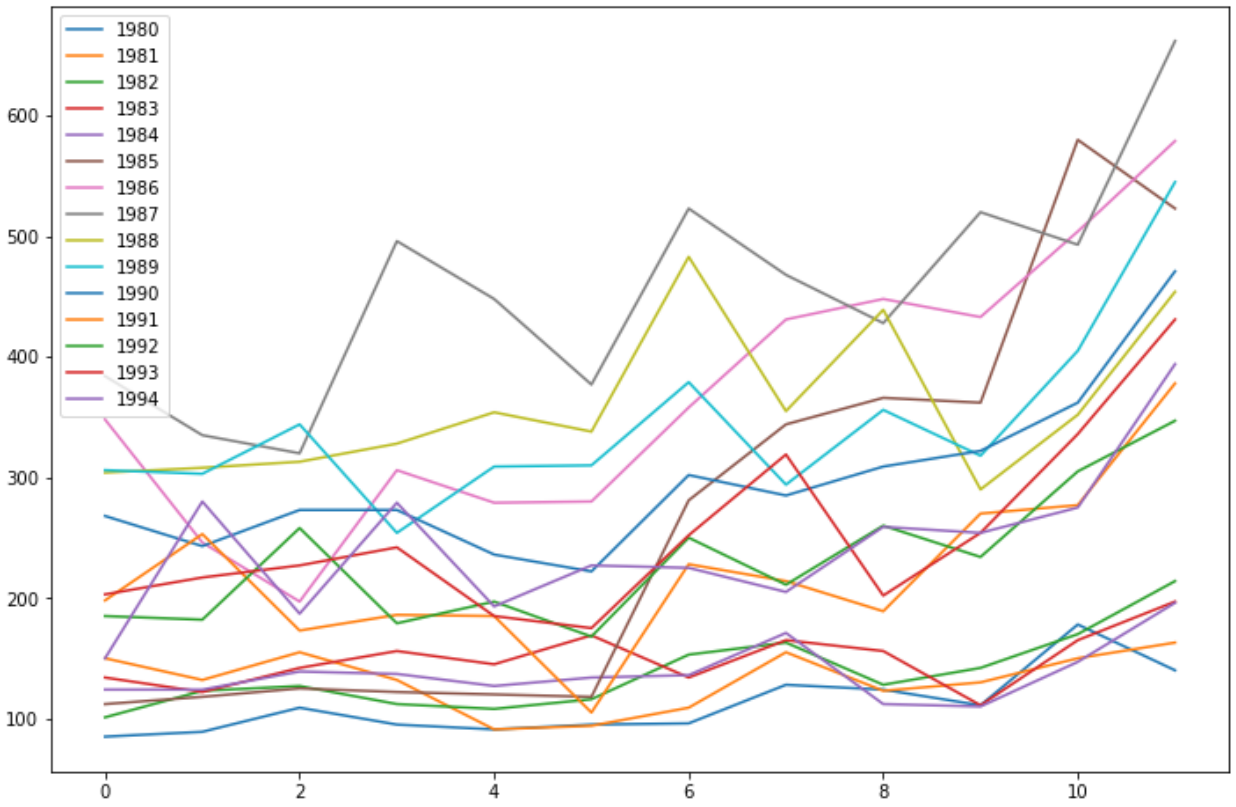
Above is the monthly sales data between years 1980 and 1995. We can see that the monthly trend for all the months. December is the best performing month followed by November. Also, outlier presence can be seen in April and May sales data.



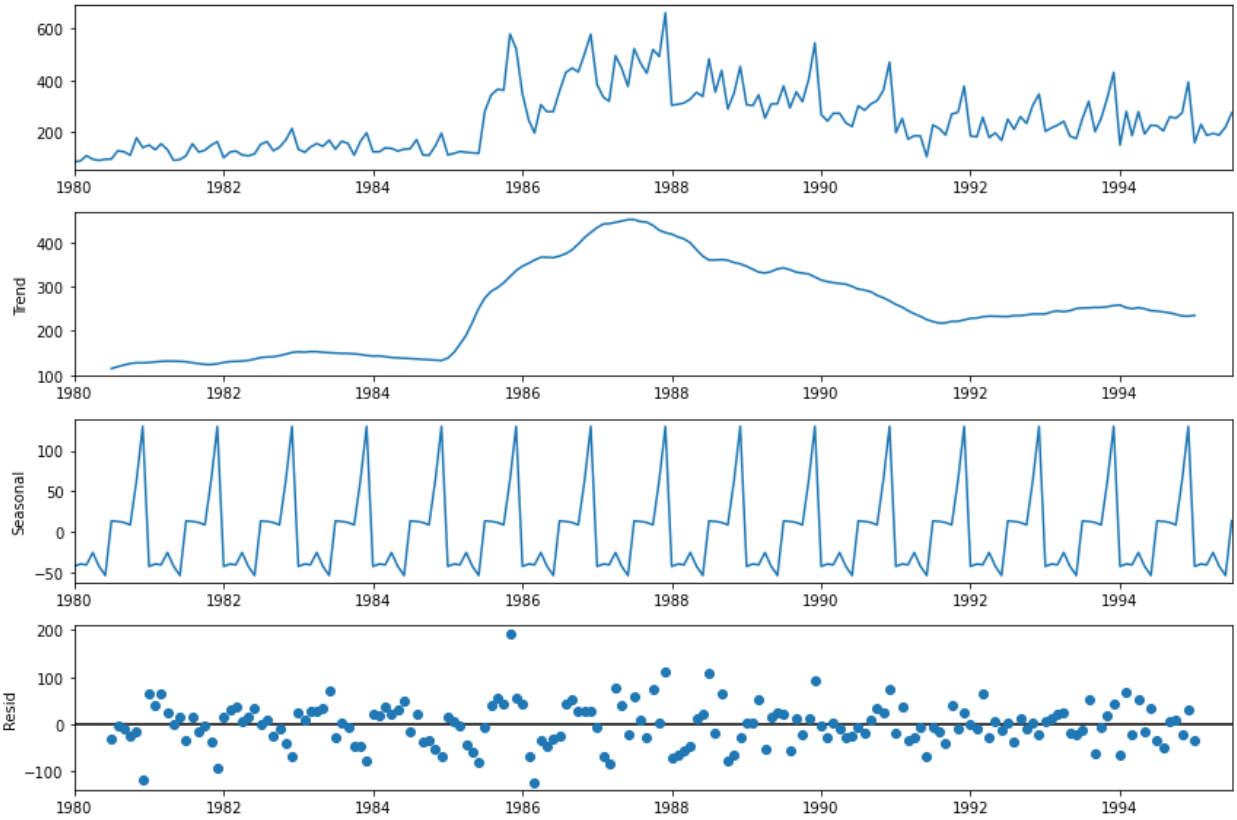
The above graph represents the sales for the years from 1980 and 1994. It is hard to understand the seasonality or trend from this figure.



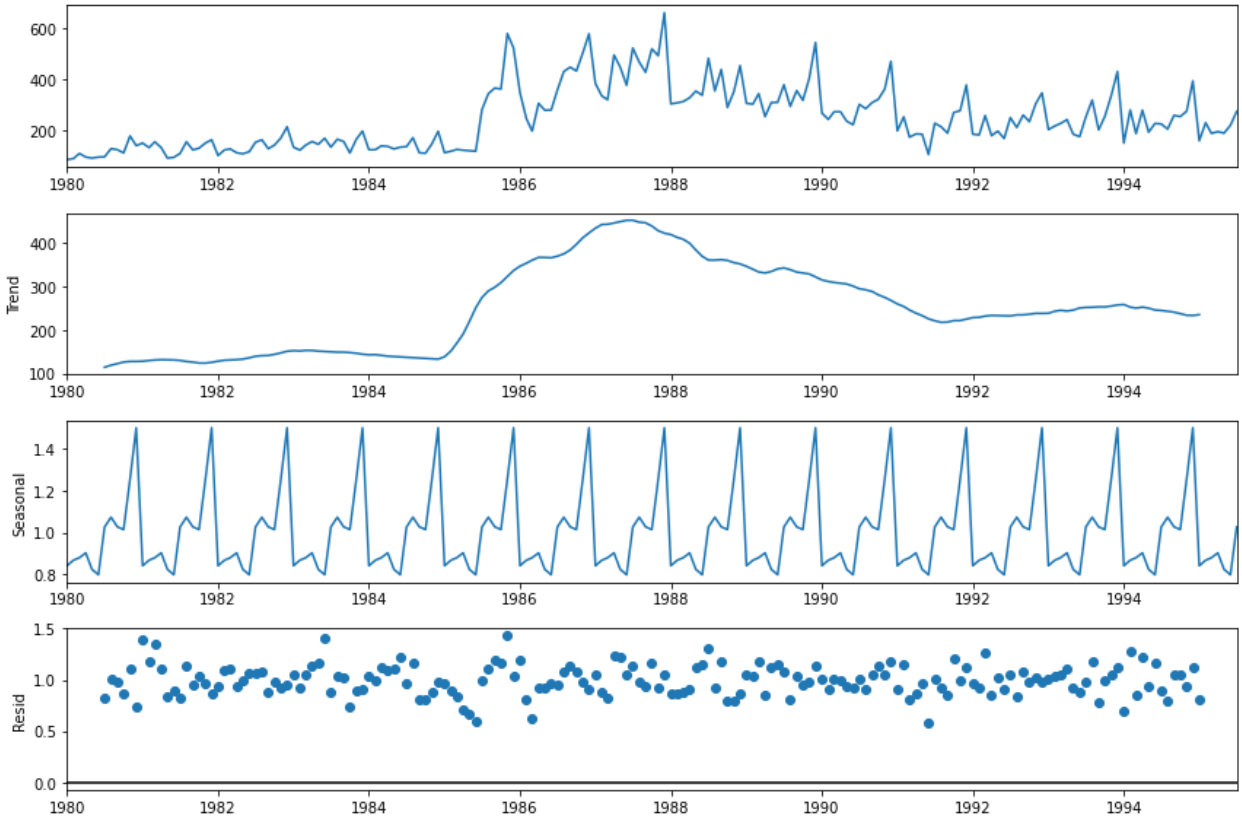
Above figure represents the yearly average sales of the shoes and the percentage change in the sales. Interestingly, we can see overall increase in the sales between 1985 to 1989. But the percentage change in sales remains constant.



Above figure represents the sales figure of the shoes for the company from 1980 till 1994.



Above is the additive decomposition of the time series dataset. Here we can clearly see the trend as well as seasonality in the data. According to the trend the overall sales increased during the time period between 1986 and 1988. In case of seasonality, it is constant all across. The residue or the error component is random in nature which is good to go. If there would have been some pattern in the residue then we must have to go towards the multiplicative decomposition.



Above is the multiplicative decomposition of the time series. Here again we can see that there is clear visible seasonality in the data. The residue here is again randomly distributed. There's no trend visible in the multiplicative decomposition as well.

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

```
(132, 1)
(55, 1)
```

The data was split into two train and test. After splitting the data the train has 132 values and test has 55 values.

YearMonth	Shoe_Sales
1980-01-01	85
1980-02-01	89
1980-03-01	109
1980-04-01	95
1980-05-01	91

Above is the head of the train data which starts with data from 1-January-1980.

YearMonth	Shoe_Sales
1991-01-01	198
1991-02-01	253
1991-03-01	173
1991-04-01	186
1991-05-01	185

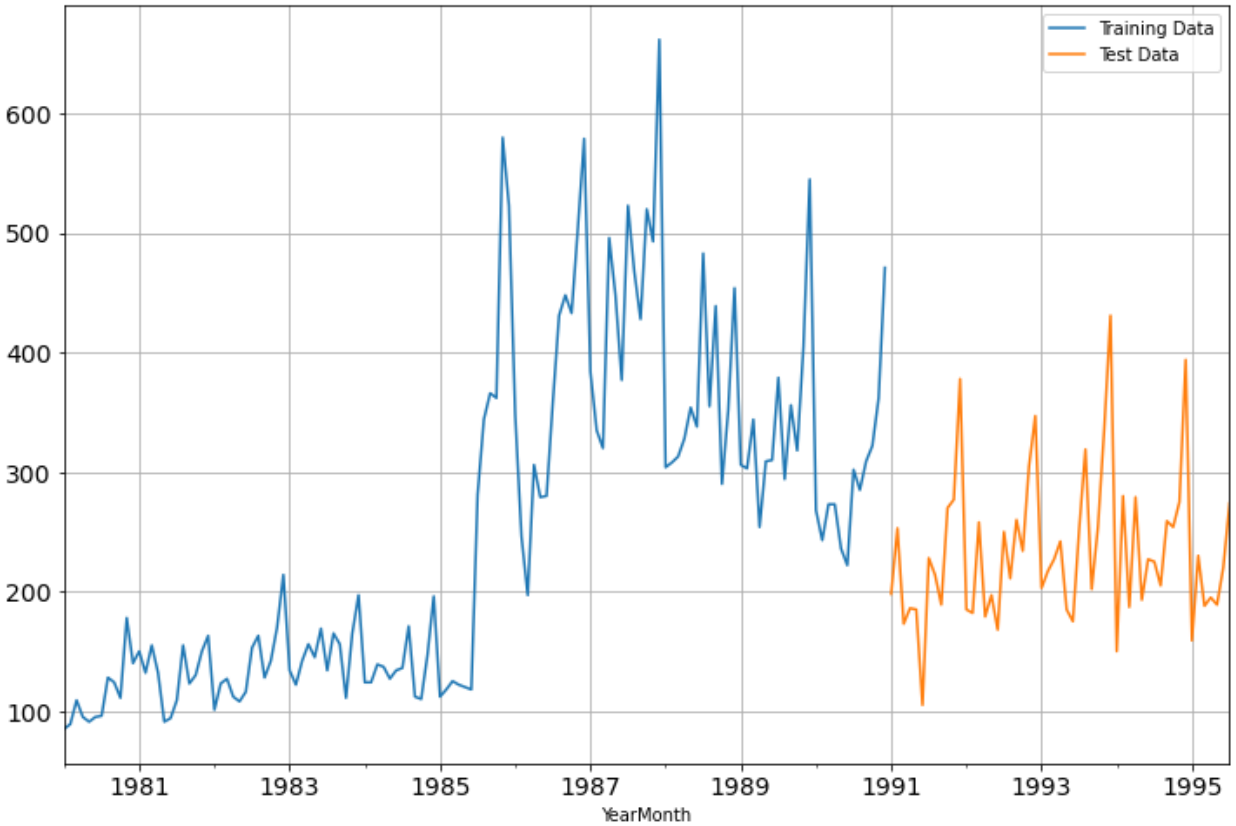
Above is the head of the test data which starts at 01-January-1991.

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 132 entries, 1980-01-01 to 1990-12-01
Data columns (total 1 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Shoe_Sales  132 non-null    int64
dtypes: int64(1)
memory usage: 2.1 KB
```

Above figure confirms that all the values in the train data are non-NULL.

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 55 entries, 1991-01-01 to 1995-07-01
Data columns (total 1 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Shoe_Sales  55 non-null     int64
dtypes: int64(1)
memory usage: 880.0 bytes
```

Above figure confirms that all the values in the test data are non-NULL.



In the above figure of entire time series, the train data till 1991 is colored in blue and the test data is colored in orange.

4. **Build various 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, simple average models etc. should also be built on the training data and check the performance on the test data using RMSE.

   Simple Exponential Smoothing (Level, No Seasonality, No Trend)

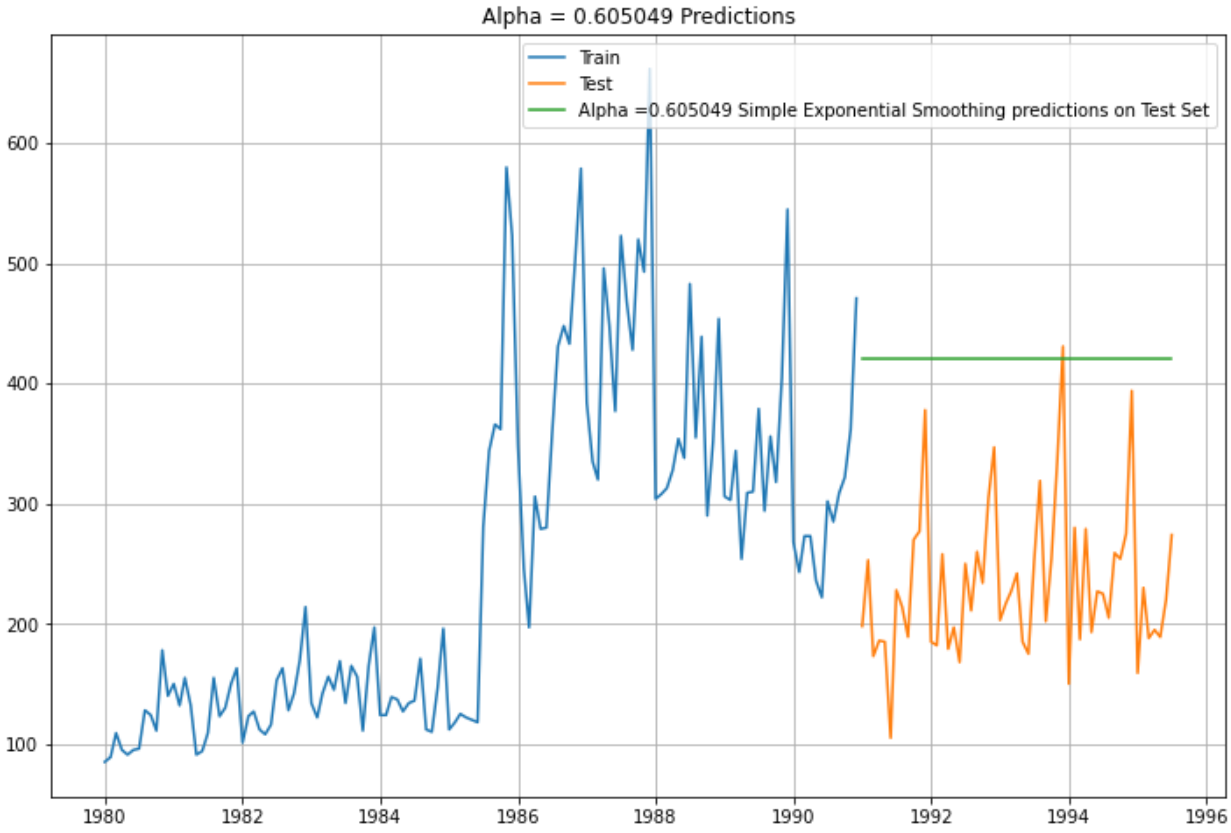
   The class was created on the train dataset for applying the Simple Exponential Smoothing. Fitting was done using fit() function. Here, the optimum parameters were chosen by the Python. Then, the parameters were checked using 'params' function.

```
{'smoothing_level': 0.605049221658923,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 88.83028430097019,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Above are the parameters for SES. The value of Alpha (Smoothing_Level) is 0.605049. The fitted model on the training set was used to forecast on the test set.

1991-01-01	420.229857
1991-02-01	420.229857
1991-03-01	420.229857
1991-04-01	420.229857
1991-05-01	420.229857
1991-06-01	420.229857
1991-07-01	420.229857
1991-08-01	420.229857
1991-09-01	420.229857
1991-10-01	420.229857
1991-11-01	420.229857
1991-12-01	420.229857
1995-01-01	420.229857
1995-02-01	420.229857
1995-03-01	420.229857
1995-04-01	420.229857
1995-05-01	420.229857
1995-06-01	420.229857
1995-07-01	420.229857

Above are the forecasted values on the test set. Value is constant i.e. 420.229857.



Green colored line in the above figure is the estimated value which is 420.229857 for the test time period from 01-January-1991 to 01-July-1995. Orange line is the actual value.

Therefore, we cannot consider this model as the gap between the actual and predicted is very high.

```
SES RMSE: 196.404836419672
SES RMSE (calculated using statsmodels): 196.404836419672
```

The Root Mean Square Error deviation for Simple Exponential Smoothing is 196.404.

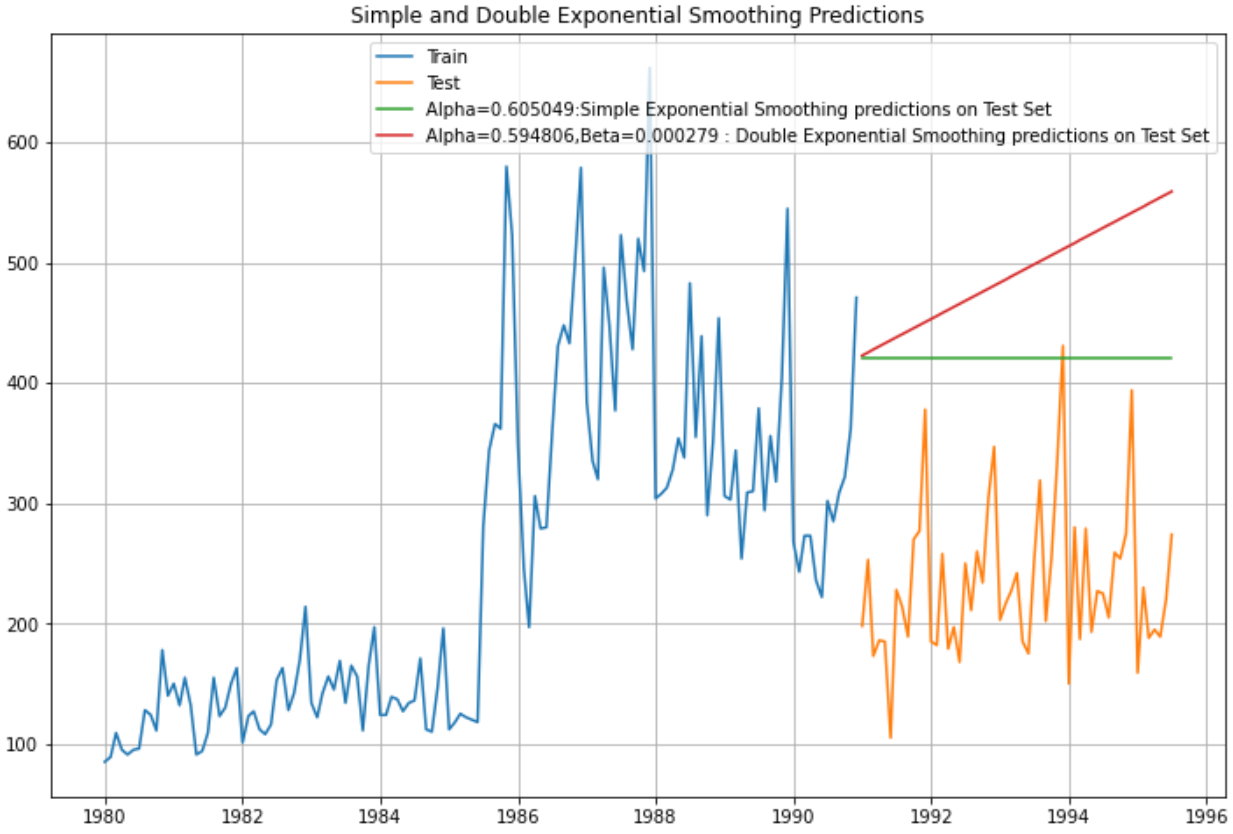
Double Exponential Smoothing (Level, Trend, No Seasonality)

The class was created on the train dataset for applying the Holt model Exponential Smoothing. Fitting was done using fit() function. Here, the optimum parameters were chosen by the Python. Then, the parameters were checked using 'params' function.

```
==Holt model Exponential Smoothing Estimated Parameters ==

{'smoothing_level': 0.5948061323729839, 'smoothing_trend': 0.000279646480657923, 'smoothing_seasonal': nan, 'damping_trend
': nan, 'initial_level': 82.93815017865691, 'initial_trend': 2.5254544148321547, 'initial_seasons': array([], dtype=float6
4), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

Value of Alpha (Smoothing_Level) is 0.594806 Value of Beta (Smoothing_Trend) is 0.000279.



Red colored line in the above figure represents the estimated values for the test time period from 01-January-1991 to 01-July-1995. Orange line is the actual value. We cannot consider this model as the gap between the actual and predicted is very high.

```
DES RMSE: 266.16120808183047
```

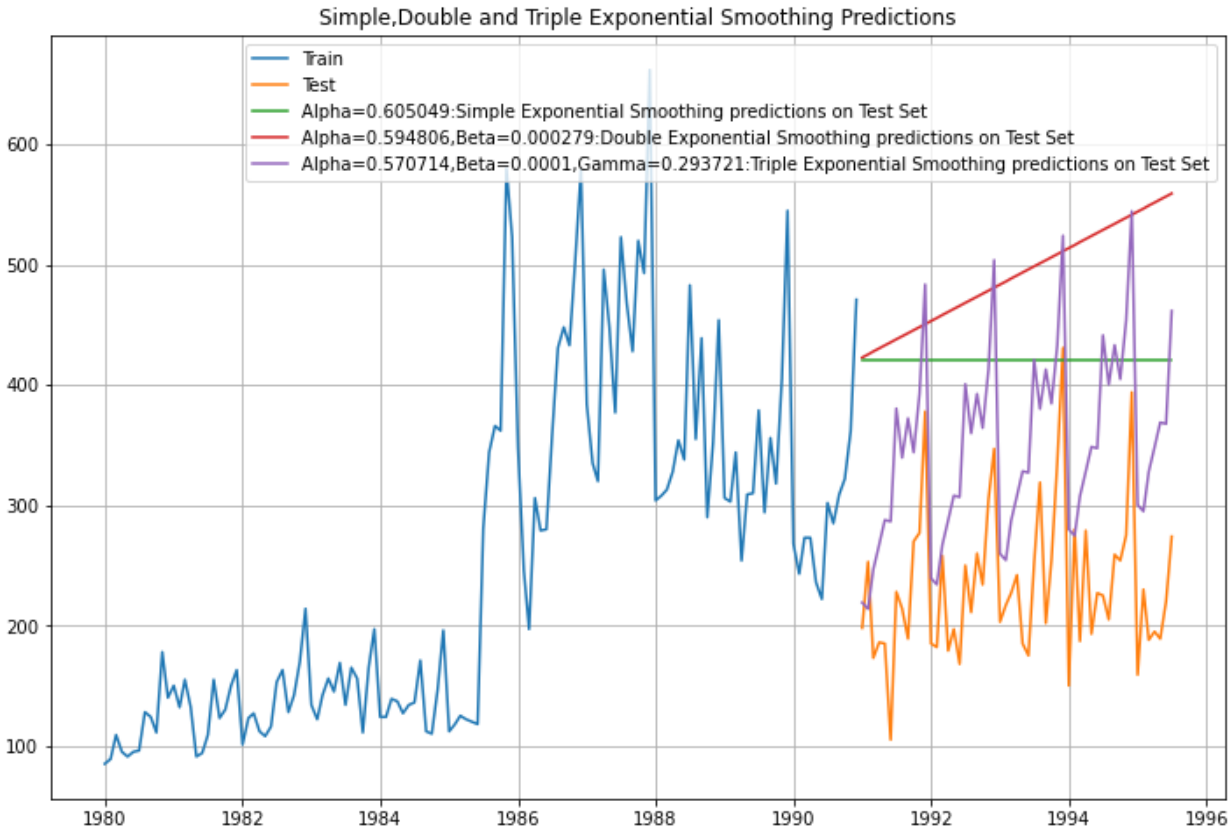
The Root Mean Square Error deviation for Double Exponential Smoothing is 266.161.

Triple Exponential Smoothing (Level, Trend, Seasonality- Additive)
The class was created on the train dataset for applying the Holt Winters model Exponential Smoothing. Fitting was done using fit() function. Here, the optimum parameters were chosen by the Python. Then, the parameters were checked using 'params' function. Here, the seasonality is additive.

```
==Holt Winters model Exponential Smoothing Estimated Parameters ==

{'smoothing_level': 0.5707142857142857, 'smoothing_trend': 0.0001, 'smoothing_seasonal': 0.29372180451127816, 'damping_tren
d': nan, 'initial_level': 116.47499999999994, 'initial_trend': 1.6939393939394016, 'initial_seasons': array([-11.20138889,
-14.06597222,   1.11111111,  -5.25347222,
       -21.42013889, -11.18055556, -10.83680556,  18.14236111,
        -2.53472222, -12.53472222,  28.90277778,  40.87152778]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

Value of Alpha: 0.5707 Value of Beta: 0.0001 Value of Gamma: 0.2937



Violet colored line in the above figure represents the estimated values for the test time period from 01-January-1991 to 01-July-1995. Orange line is the actual value. This model has performed better than the SES and DES. However, we cannot consider this model as the gap between the actual and predicted is very high.

```
TES RMSE: 128.99252592312354
```

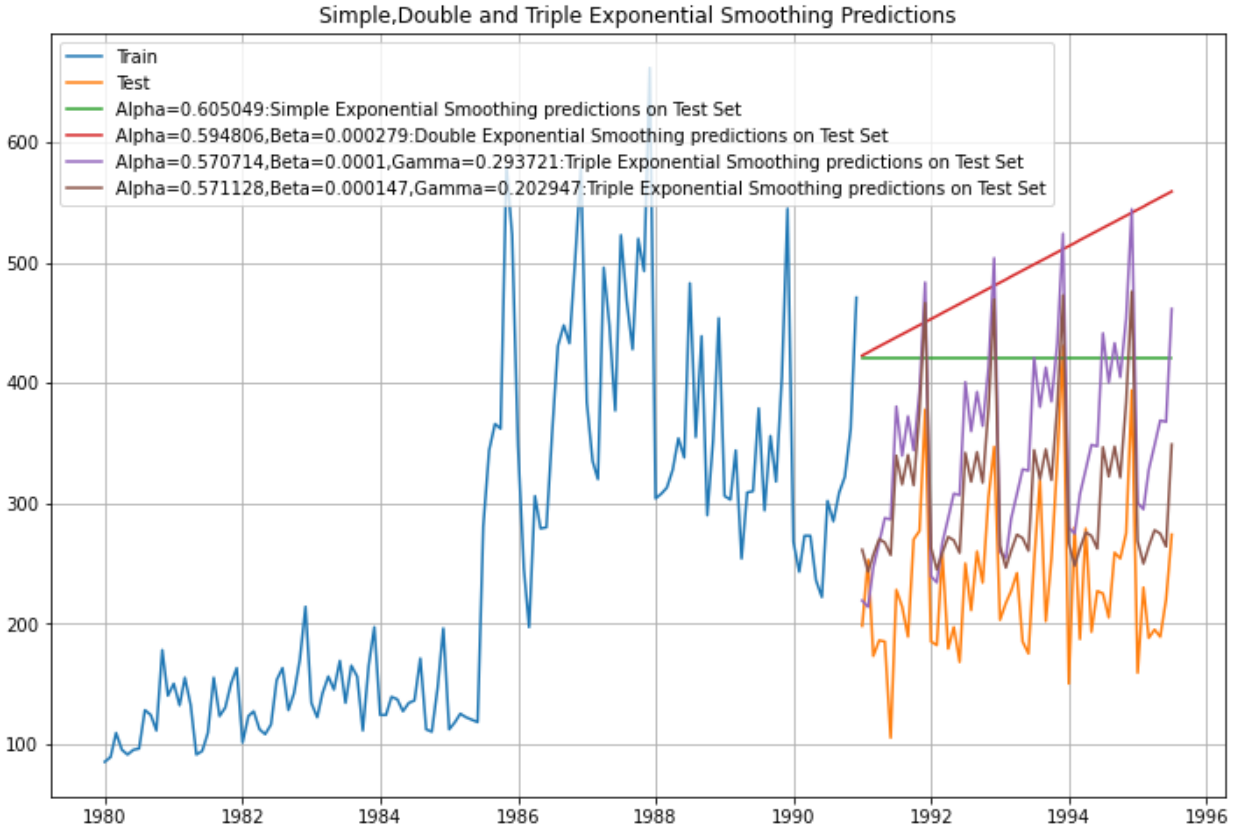
The Root Mean Square Error deviation for Triple Exponential Smoothing is 128.992.

Triple Exponential Smoothing (Level, Trend, Seasonality- Multiplicative)
The class was created on the train dataset for applying the Holt Winters model Exponential Smoothing. Fitting was done using fit() function. Here, the optimum parameters were chosen by the Python. Then, the parameters were checked using 'params' function. Here, the seasonality is multiplicative.

```
==Holt Winters model Exponential Smoothing Estimated Parameters ==

{'smoothing_level': 0.5711286329525818, 'smoothing_trend': 0.00014781930867568429, 'smoothing_seasonal': 0.2029473370607799
4, 'damping_trend': nan, 'initial_level': 116.35529208070726, 'initial_trend': 0.11219854465675648, 'initial_seasons': arra
y([1.05679343, 1.01130311, 1.2337466 , 1.40663129, 1.32162715,
       1.07936886, 1.18018187, 1.50183082, 1.72369093, 1.4704132 ,
       1.75485304, 1.92101444]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

Value of Alpha: 0.571128 Value of Beta: 0.000147 Value of Gamma: 0.202947



Brown colored line in the above figure represents the estimated values for the test time period from 01-January-1991 to 01-July-1995. Orange line is the actual value. This model has performed better than the SES, DES and TES (additive). We might consider this model as the gap between the actual and predicted is less.

```
TES_Multiplicative RMSE: 83.734048494837
```

The Root Mean Square Error deviation for Triple Exponential Smoothing- Multiplicative is 83.7340.

	Test RMSE
Alpha=0.605049,SES	196.404836
Alpha=1,Beta=0.0189:DES	266.161208
Alpha=0.570714,Beta=0.0001,Gamma=0.293721:TES(Additive)	128.992526
Alpha=0.571128,Beta=0.000147,Gamma=0.202947,Gamma=0:TES(Multiplicative)	83.734048

We can see that the Triple Exponential Smoothing Multiplicative model has performed the best with least RMSE deviation value i.e. 83.734048.

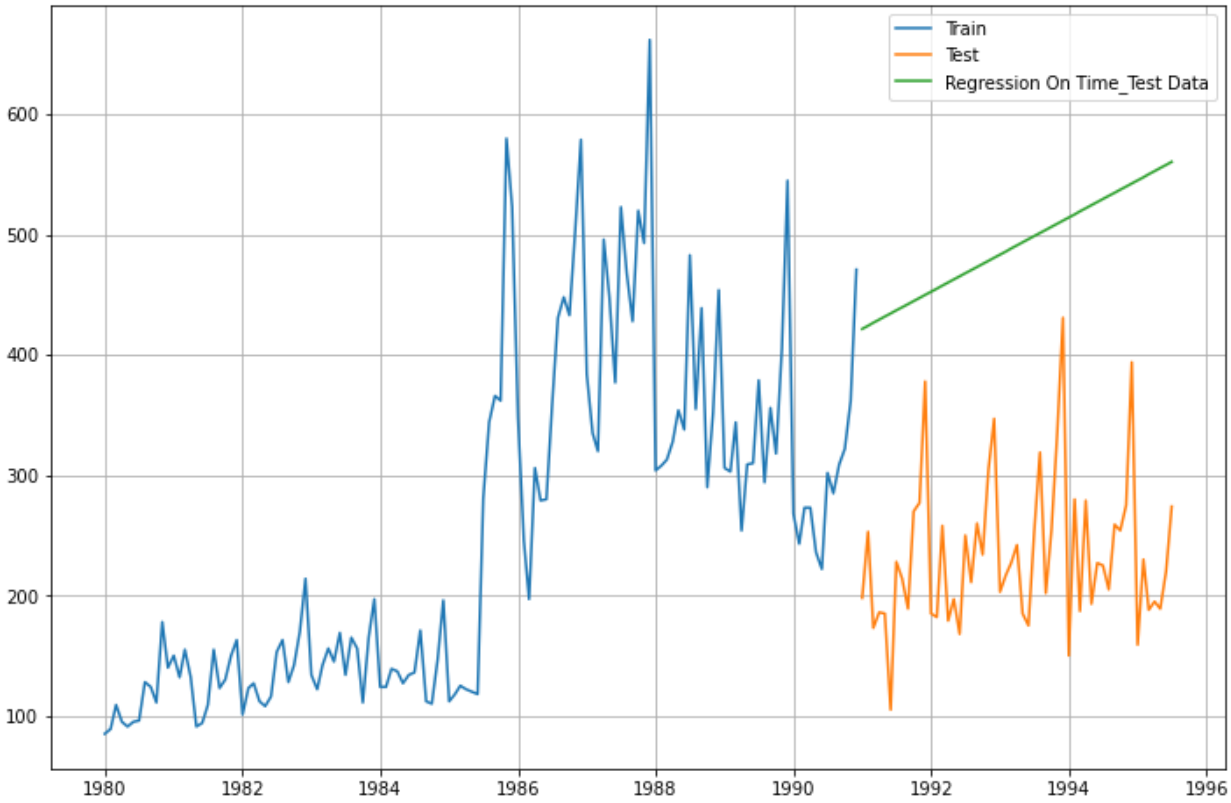
Building different models and comparing the accuracy metrics:

Linear Regression

For Linear Regression, We will take YearMonth as the independent variable and Shoe_Sales as dependent variable.

```
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
 [133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 1
57, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181,
182, 183, 184, 185, 186, 187]
```

Above figure represents the training time instance as well as the test time instance. We have generated the numerical time instance order for both the training and test dataset. Further, we shall add them to training and test set respectively. After the training and test data has been modified. Linear Regression was used to build the model on the training data and test the model on the test data.



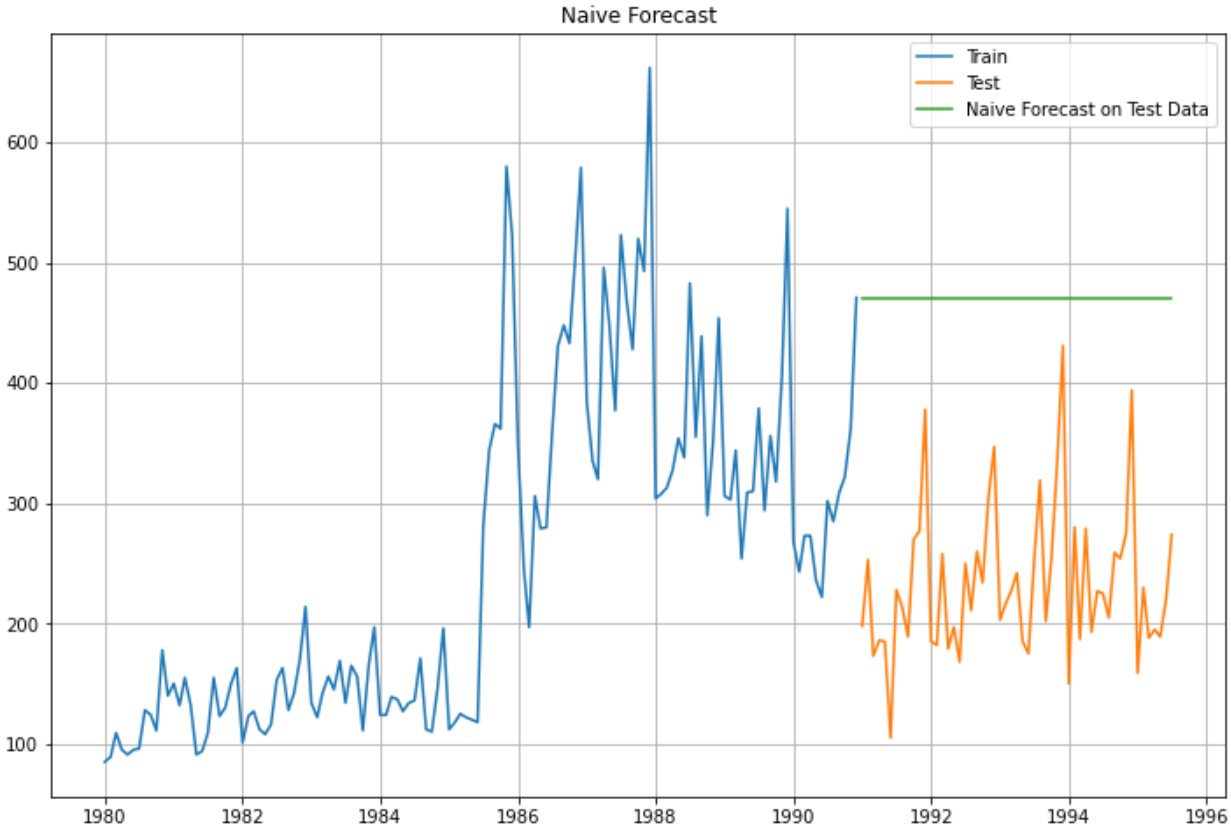
Green colored line in the above figure represents the estimated values for the test time period from 01-January-1991 to 01-July-1995. Orange line is the actual value. We cannot consider this model as the gap between the actual and predicted is very high.

```
For RegressionOnTime forecast on the Test Data,  RMSE is 266.276
```

The Root Mean Square Error deviation for Linear Regression is 266.276.

Naive Bayes

Naïve Bayes model was applied to the dataset.



Green colored line in the above figure represents the estimated values for the test time period from 01-January-1991 to 01-July-1995. Orange line is the actual value. We cannot consider this model as the gap between the actual and predicted is very high.

```
For RegressionOnTime forecast on the Test Data,  RMSE is 245.121
```

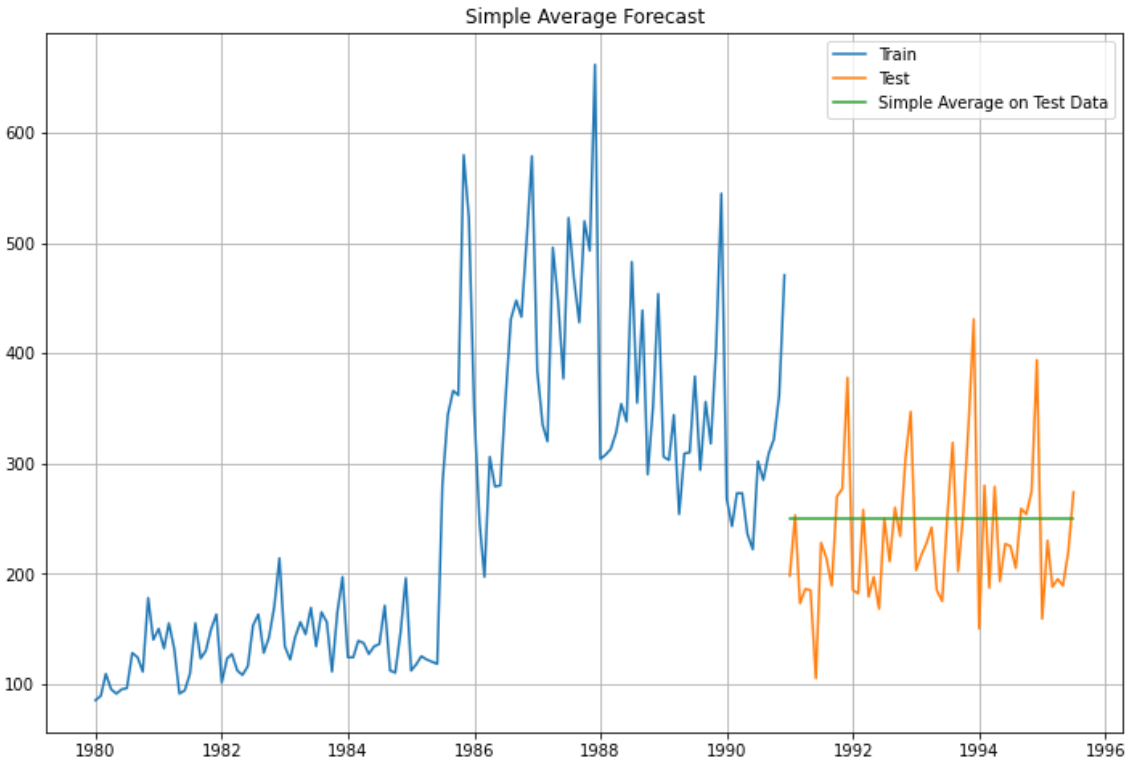
The Root Mean Square Error deviation for Naïve Bayes is 245.121

Simple Average

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

YearMonth	Shoe_Sales	mean_forecast
1991-01-01	198	250.575758
1991-02-01	253	250.575758
1991-03-01	173	250.575758
1991-04-01	186	250.575758
1991-05-01	185	250.575758

The above figure represents the forecast values for the test period which is constant i.e. 250.575758.



Green colored line in the above figure is the estimated value which is 250.575758. Orange line is the actual value. In this model the gap between the actual and predicted isn't that much. But, the predicted line is flat that cannot be used for prediction.

```
For Simple Average forecast on the Test Data,  RMSE is 63.985
```

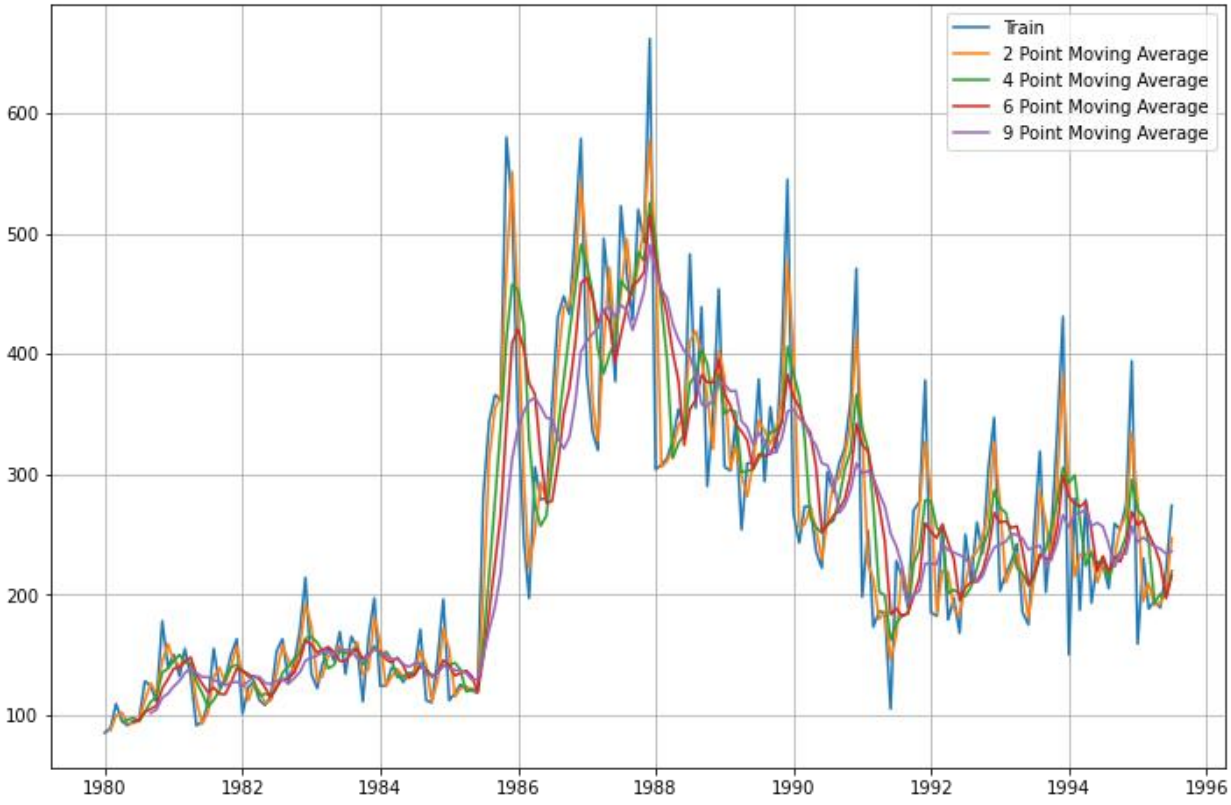
The Root Mean Square Error deviation for Simple Average Method is 63.985. Considering the RMSE value the has performed the best amongst all the models applied on the test dataset.

Moving Average

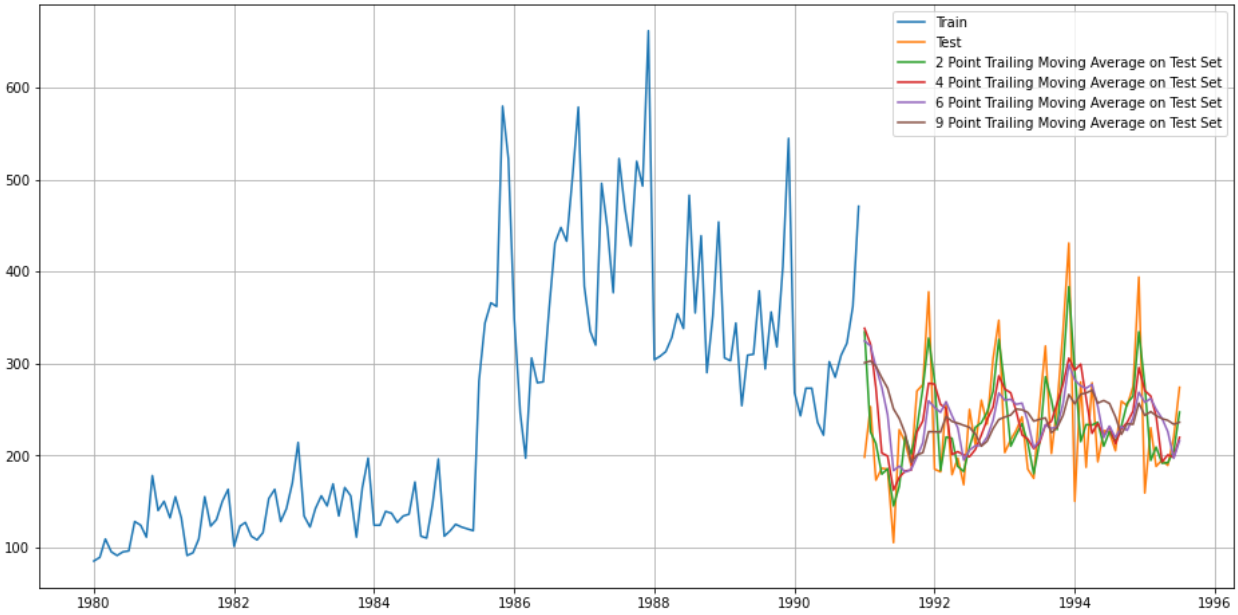
For the moving average model, we are going to calculate rolling means (or moving averages) for different intervals. The best interval can be determined by the maximum accuracy (or the minimum error) over here. For Moving Average, we are going to average over the entire data.

YearMonth	Shoe_Sales	Trailing_2	Trailing_4	Trailing_6	Trailing_9
1980-01-01	85	NaN	NaN	NaN	NaN
1980-02-01	89	87.0	NaN	NaN	NaN
1980-03-01	109	99.0	NaN	NaN	NaN
1980-04-01	95	102.0	94.5	NaN	NaN
1980-05-01	91	93.0	96.0	NaN	NaN

Above figure shows the head() for the Trailing Moving Average for intervals 2,4,6 and 9.



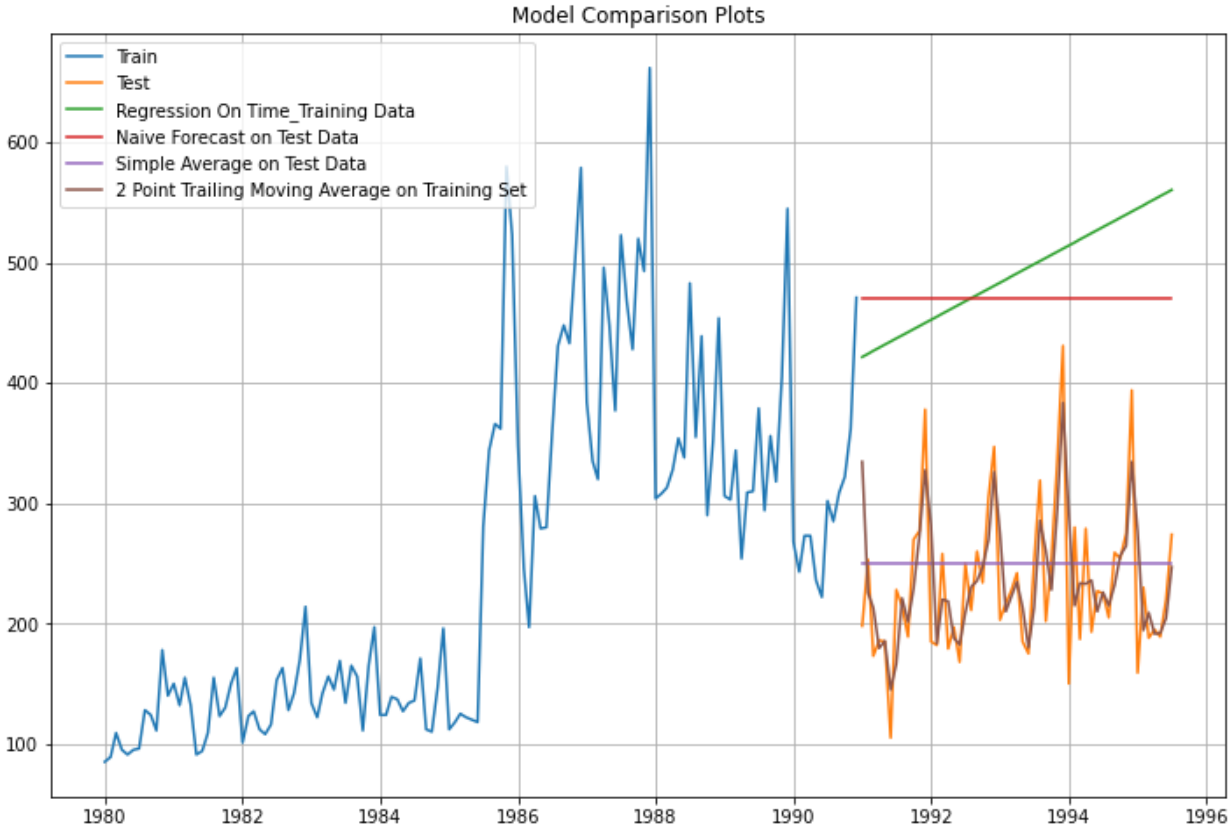
Above diagram is plotted on the whole data. As the Moving Average was applied on the whole data for training therefore its performance against the test data is not visible. Therefore, we need to divide the data into the train as well as the test dataset to understand the performance of the model.



After splitting the data into train and test, the above figure shows the prediction against the test dataset. As the trailing moving average is very close to each other therefore it is hard to understand from the figure that which one among these have performed the best. Therefore, we need to rely on deviation RMSE values to understand the performance of these trailing moving average lines.

```
For 2 point Moving Average Model forecast on the Training Data,  RMSE is 45.949
For 4 point Moving Average Model forecast on the Training Data,  RMSE is 57.873
For 6 point Moving Average Model forecast on the Training Data,  RMSE is 63.457
For 9 point Moving Average Model forecast on the Training Data,  RMSE is 67.724
```

Above figure gives the RMSE values for all the trailing moving average prediction line. We can understand that the best performance is by the 2 point Moving Average Model forecast that is 45.949.



Above figure is the Model Comparison plot to confirm that the 2 point Trailing Moving Average has performed best and prediction is very close to the actual test data.

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.**

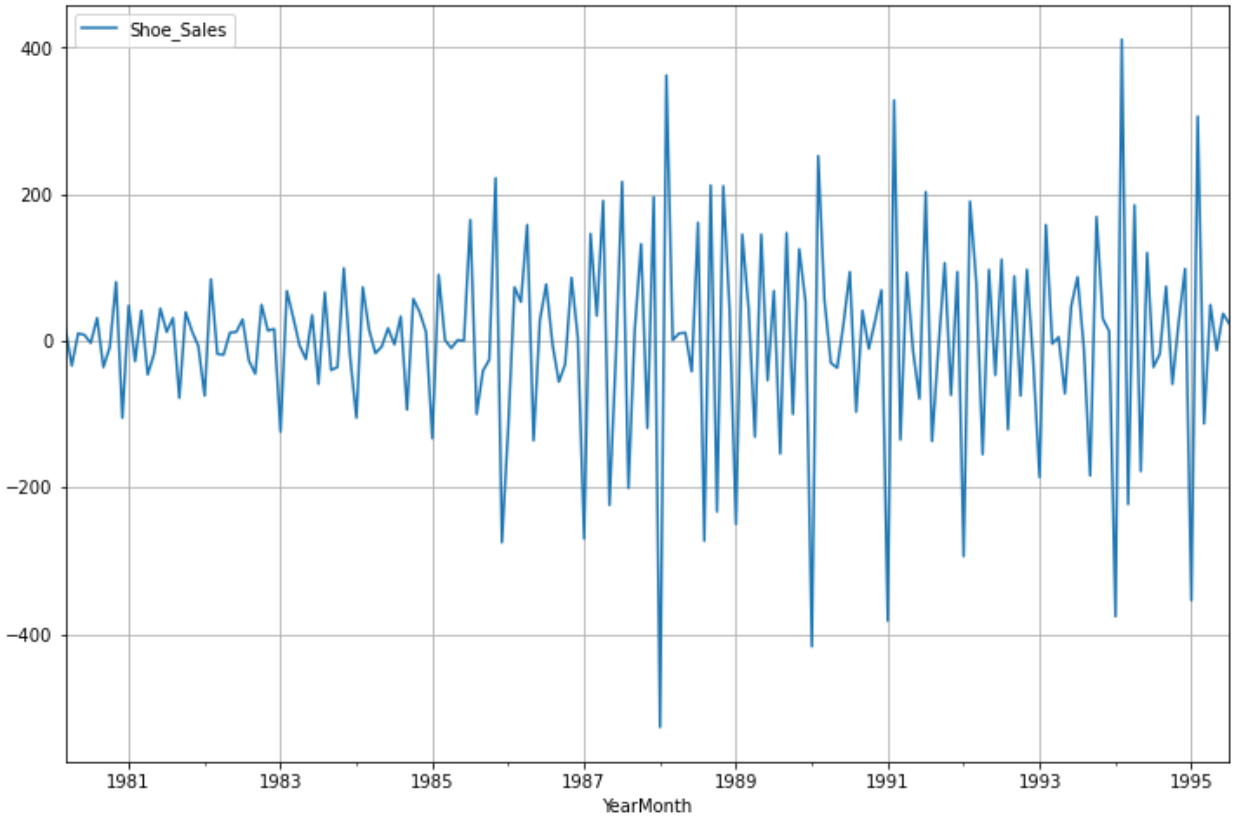
Null Hypothesis (H_0): The Time Series has a unit root and is thus non-stationary.
Alternate Hypothesis (H_1): The Time Series does not have a unit root and is thus stationary.
The python library used is the adfuller from statsmodel.

```
DataFrame test statistic is -1.577
DataFrame test p-value is 0.8014186234536529
Number of lags used 13
```

The p-value here is 0.8014 which is greater than 0.05, so we cannot reject the Null Hypothesis. Therefore, we can conclude that the series is not stationary.
To make the time series stationary, we need to take the first order difference of the time-series. Also, we need to drop the null values present in the data.

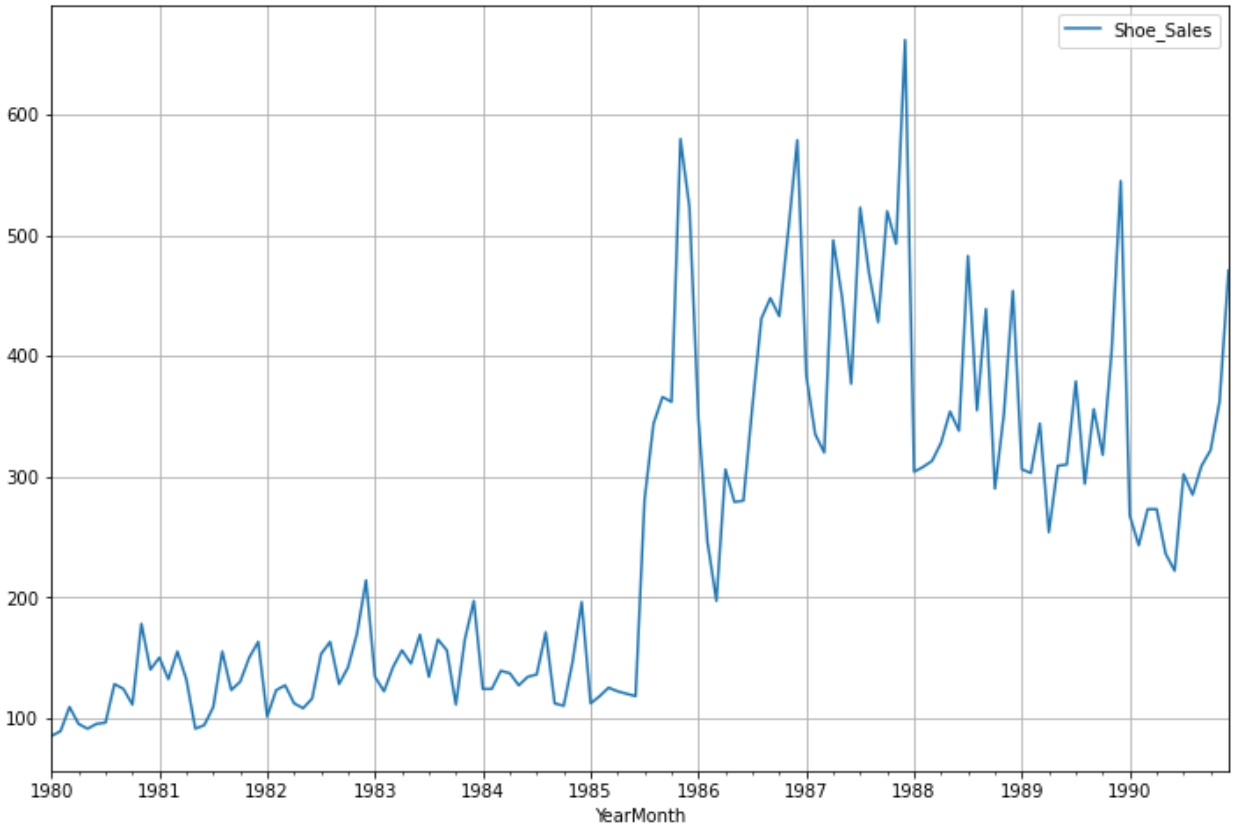
```
DF test statistic is -3.532
DF test p-value is 0.03611703400135922
Number of lags used 12
```

Now, if we check the p-value then it is 0.03611 which is less than 0.05.
Hence, we can reject the Null Hypothesis and accept the alternate Hypothesis.



Above is the plot of the series, we can see that now it is stationary. A stationary series has property of a constant mean and a constant variance which can be seen clearly in the plot above.

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**



Above plot shows the training data. Here we can clearly understand that the training data is not stationary. Therefore, we have to make it stationary for applying the ARIMA/ SARIMA.

```
DataFrame test statistic is -1.749
DataFrame test p-value is 0.7287654522797273
Number of lags used 13
```

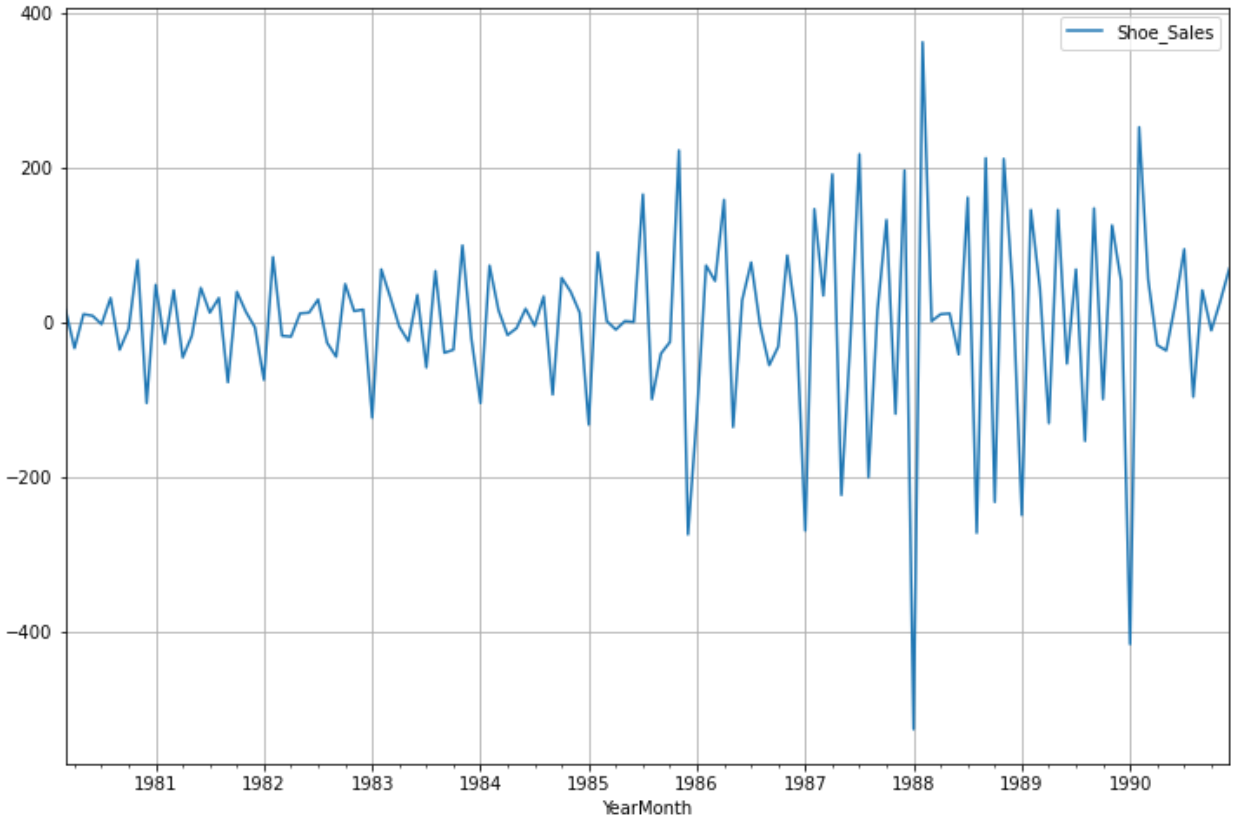
The training data is non-stationary at 95% confidence level. Let us take a first level of differencing to make the Time Series stationary.

```
DataFrame test statistic is -3.181
DataFrame test p-value is 0.0882258925591975
Number of lags used 13
```

The training data is non-stationary at 95% confidence level. Let us take a second level of differencing to make the Time Series stationary.

```
DataFrame test statistic is -6.564
DataFrame test p-value is 1.3362283516594626e-07
Number of lags used 13
```

After the second level of differencing the training data has p-value lower than 0.05. Therefore, we can reject the null hypothesis and go with alternate hypothesis.



Above plot the training dataset after making it stationary. Now it is good to go for ARIMA/ SARIMA modelling.

Auto ARIMA

```
Examples of the par
Model: (0, 2, 0)
Model: (0, 2, 1)
Model: (0, 2, 2)
Model: (0, 2, 3)
Model: (1, 2, 0)
Model: (1, 2, 1)
Model: (1, 2, 2)
Model: (1, 2, 3)
Model: (2, 2, 0)
Model: (2, 2, 1)
Model: (2, 2, 2)
Model: (2, 2, 3)
Model: (3, 2, 0)
Model: (3, 2, 1)
Model: (3, 2, 2)
Model: (3, 2, 3)
```

The above loop helps us in getting a combination of different parameters of p and q in the range of 0 and 2. We have kept the value of d as 2 as we need to take a difference of the series to make it stationary.

```
ARIMA(0, 2, 0) - AIC:1894.29072879556
ARIMA(0, 2, 1) - AIC:1745.4141816798729
ARIMA(0, 2, 2) - AIC:1614.1526200043695
ARIMA(0, 2, 3) - AIC:1516.670058295184
ARIMA(1, 2, 0) - AIC:1786.052880627531
ARIMA(1, 2, 1) - AIC:1667.154779794173
ARIMA(1, 2, 2) - AIC:1568.0286156894736
ARIMA(1, 2, 3) - AIC:1511.5579647953296
ARIMA(2, 2, 0) - AIC:1727.8901467216583
ARIMA(2, 2, 1) - AIC:1628.9244397209254
ARIMA(2, 2, 2) - AIC:1550.279581343846
ARIMA(2, 2, 3) - AIC:1569.0915480922906
ARIMA(3, 2, 0) - AIC:1703.5893067845086
ARIMA(3, 2, 1) - AIC:1606.116538301513
ARIMA(3, 2, 2) - AIC:1632.3773834249255
ARIMA(3, 2, 3) - AIC:1554.2856947389482
```

We have executed a loop within the pdq parameters defined by itertools. The parameters were used from the loop. We have appended the AIC values and the model parameters to the previously created data frame. For easier understanding, we shall be sorting of the AIC values.

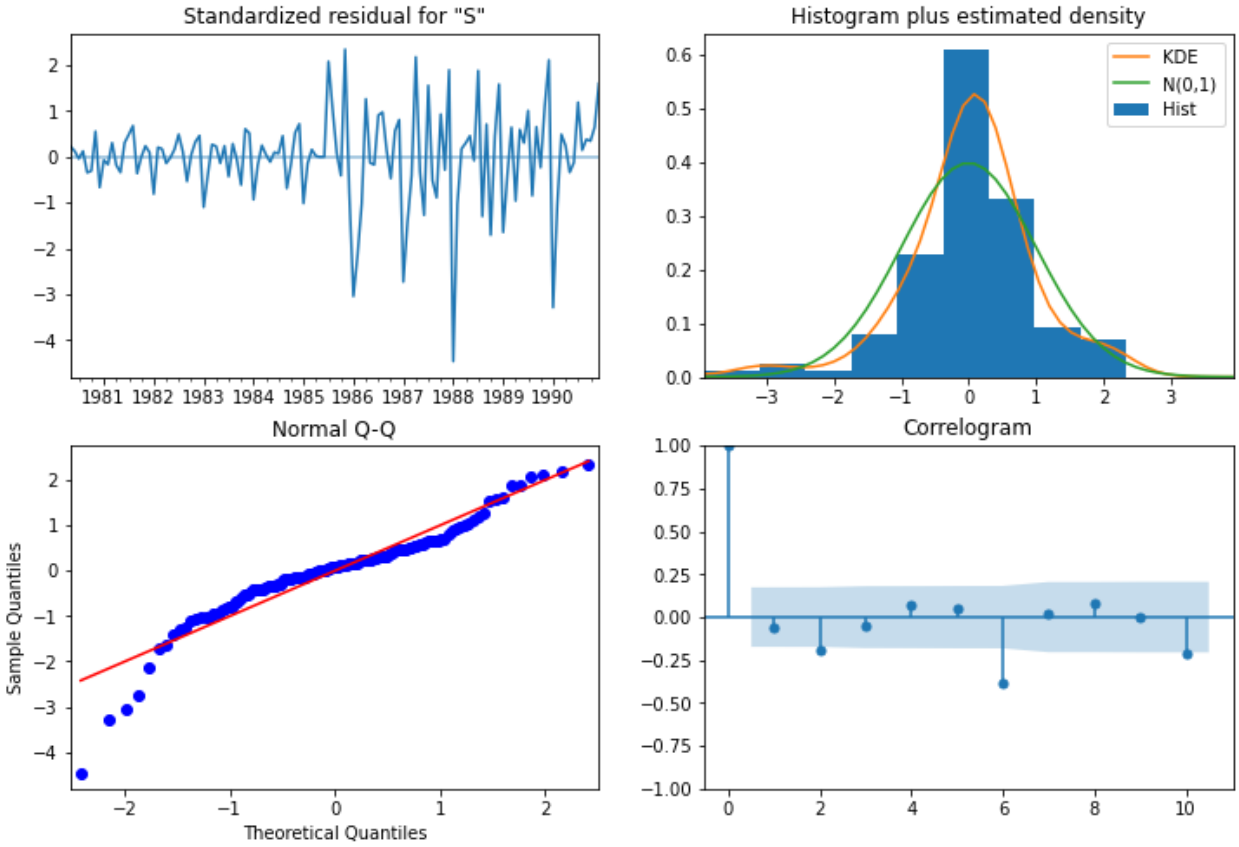
	param	AIC
7	(1, 2, 3)	1511.557965
3	(0, 2, 3)	1516.670058
10	(2, 2, 2)	1550.279581
15	(3, 2, 3)	1554.285695
6	(1, 2, 2)	1568.028616

Above figure has sorted AIC values. Here the best parameter is [pdq] = [1,2,3]

```
                               SARIMAX Results
==============================================================================
Dep. Variable:             Shoe_Sales   No. Observations:                  130
Model:                 ARIMA(1, 2, 3)   Log Likelihood                -750.779
Date:                Thu, 06 Jan 2022   AIC                           1511.558
Time:                        23:21:02   BIC                           1525.818
Sample:                    03-01-1980   HQIC                          1517.352
                         - 12-01-1990
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.2365      0.071     -3.317      0.001      -0.376      -0.097
ma.L1         -2.9980      0.085    -35.267      0.000      -3.165      -2.831
ma.L2          2.9965      0.167     17.905      0.000       2.668       3.325
ma.L3         -0.9985      0.084    -11.824      0.000      -1.164      -0.833
sigma2      5555.3225   9.02e-05   6.16e+07      0.000    5555.322    5555.323
===================================================================================
Ljung-Box (L1) (Q):                   0.48   Jarque-Bera (JB):                95.78
Prob(Q):                              0.49   Prob(JB):                         0.00
Heteroskedasticity (H):              11.04   Skew:                            -1.01
Prob(H) (two-sided):                  0.00   Kurtosis:                         6.72
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 2.63e+22. Standard errors may be unstable.
```

With the best parameters the ARIMA model was fitted. Above is the summary of it.

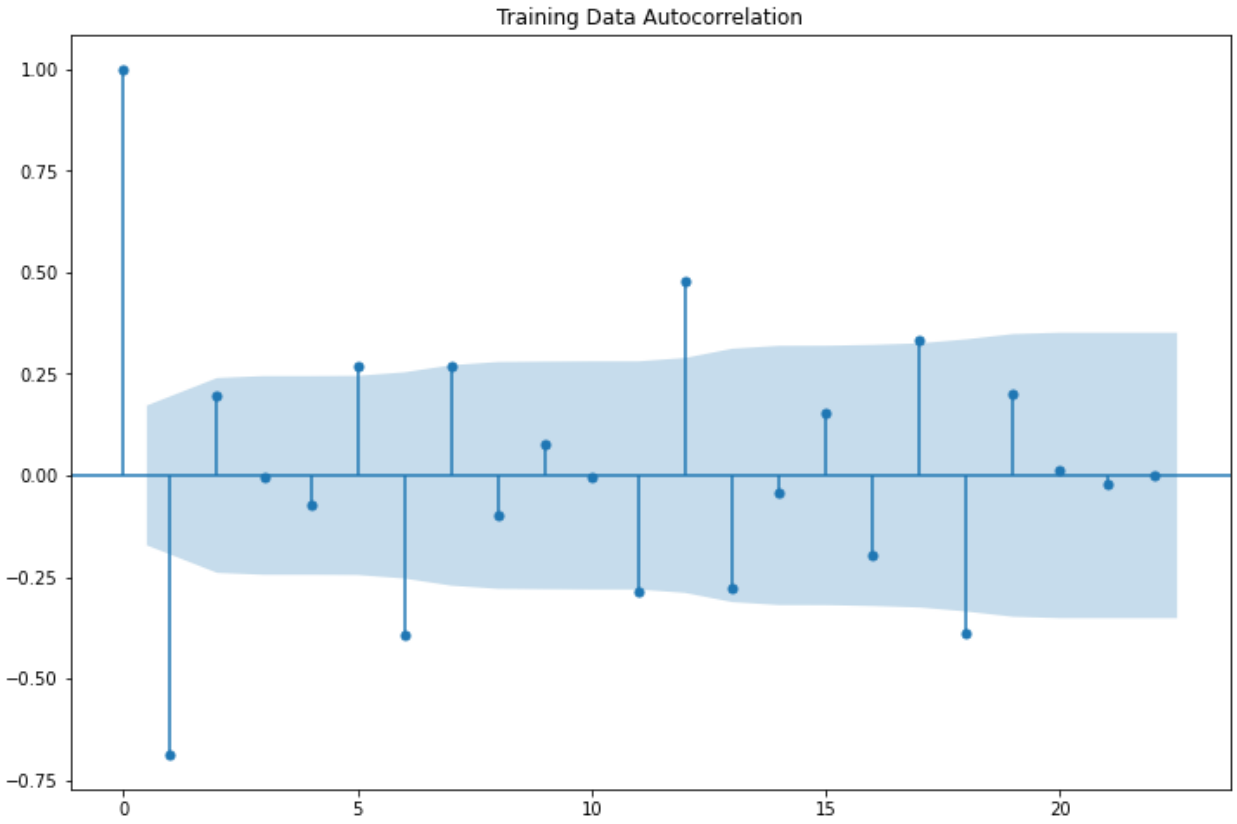


The diagnostic function was used on the train data. Above are the results for the same. Now comes the phase where we shall check the prediction on the test data and get the accuracy of the model.

```
RMSE: 243.9313191324045
MAPE: 101.05189061285644
```

Above is the deviation RMSE & MAPE values against the test data.

Auto-SARIMA



Here we shall take the seasonality component as 6. We shall ignore the first series in the above figure because of its high correlation with original series. But from Lag1, we can clearly see that every 6th Lag is significant in the above diagram. Therefore, seasonality is 6.

```
Examples of the parameter combinations for the Model are
Model: (0, 2, 1)(0, 0, 1, 6)
Model: (0, 2, 2)(0, 0, 2, 6)
Model: (0, 2, 3)(0, 0, 3, 6)
Model: (1, 2, 0)(1, 0, 0, 6)
Model: (1, 2, 1)(1, 0, 1, 6)
Model: (1, 2, 2)(1, 0, 2, 6)
Model: (1, 2, 3)(1, 0, 3, 6)
Model: (2, 2, 0)(2, 0, 0, 6)
Model: (2, 2, 1)(2, 0, 1, 6)
Model: (2, 2, 2)(2, 0, 2, 6)
Model: (2, 2, 3)(2, 0, 3, 6)
Model: (3, 2, 0)(3, 0, 0, 6)
Model: (3, 2, 1)(3, 0, 1, 6)
Model: (3, 2, 2)(3, 0, 2, 6)
Model: (3, 2, 3)(3, 0, 3, 6)
```

Above are the examples of the parameter combinations for the model. We shall choose the one with lowest AIC value.

```
SARIMA(0, 2, 0)x(0, 0, 0, 6) - AIC:1880.446416345086
SARIMA(0, 2, 0)x(0, 0, 1, 6) - AIC:1787.448866231284
SARIMA(0, 2, 0)x(0, 0, 2, 6) - AIC:1688.3935430691865
SARIMA(0, 2, 0)x(0, 0, 3, 6) - AIC:1602.971130876005
SARIMA(0, 2, 0)x(1, 0, 0, 6) - AIC:1792.411326872125
SARIMA(0, 2, 0)x(1, 0, 1, 6) - AIC:1760.144637538666
SARIMA(0, 2, 0)x(1, 0, 2, 6) - AIC:1676.2589807842692
SARIMA(0, 2, 0)x(1, 0, 3, 6) - AIC:1594.404042489281
SARIMA(0, 2, 0)x(2, 0, 0, 6) - AIC:1691.2638766302512
SARIMA(0, 2, 0)x(2, 0, 1, 6) - AIC:1688.147229084905
SARIMA(0, 2, 0)x(2, 0, 2, 6) - AIC:1676.4461571331508
SARIMA(0, 2, 0)x(2, 0, 3, 6) - AIC:1595.9273286052407
SARIMA(0, 2, 0)x(3, 0, 0, 6) - AIC:1605.3156605806305
SARIMA(0, 2, 0)x(3, 0, 1, 6) - AIC:1607.302598760014
SARIMA(0, 2, 0)x(3, 0, 2, 6) - AIC:1608.9990657617163
SARIMA(0, 2, 0)x(3, 0, 3, 6) - AIC:1592.8289968767522
SARIMA(0, 2, 1)x(0, 0, 0, 6) - AIC:1719.145986831026
.
.
.
.
SARIMA(3, 2, 3)x(1, 0, 3, 6) - AIC:1243.3112032640865
SARIMA(3, 2, 3)x(2, 0, 0, 6) - AIC:1287.2231132473285
SARIMA(3, 2, 3)x(2, 0, 1, 6) - AIC:1287.2796045099867
SARIMA(3, 2, 3)x(2, 0, 2, 6) - AIC:1280.363377685908
SARIMA(3, 2, 3)x(2, 0, 3, 6) - AIC:1215.9197514280074
SARIMA(3, 2, 3)x(3, 0, 0, 6) - AIC:1225.9763211144018
SARIMA(3, 2, 3)x(3, 0, 1, 6) - AIC:1223.6290610207702
SARIMA(3, 2, 3)x(3, 0, 2, 6) - AIC:1224.52189324241
SARIMA(3, 2, 3)x(3, 0, 3, 6) - AIC:1220.1023874824762
```

Automated SARIMA was applied to the training data.

	param	seasonal	AIC
119	(1, 2, 3)	(1, 0, 3, 6)	1210.851397
251	(3, 2, 3)	(2, 0, 3, 6)	1215.919751
183	(2, 2, 3)	(1, 0, 3, 6)	1216.950935
59	(0, 2, 3)	(2, 0, 3, 6)	1219.433624
255	(3, 2, 3)	(3, 0, 3, 6)	1220.102387

Above are the parameters sorted from low to high according to the AIC.

Therefore, p=1 d=2 q=3 P=1 D=0 Q=3 are the best parameters according to the model.

SARIMAX Results

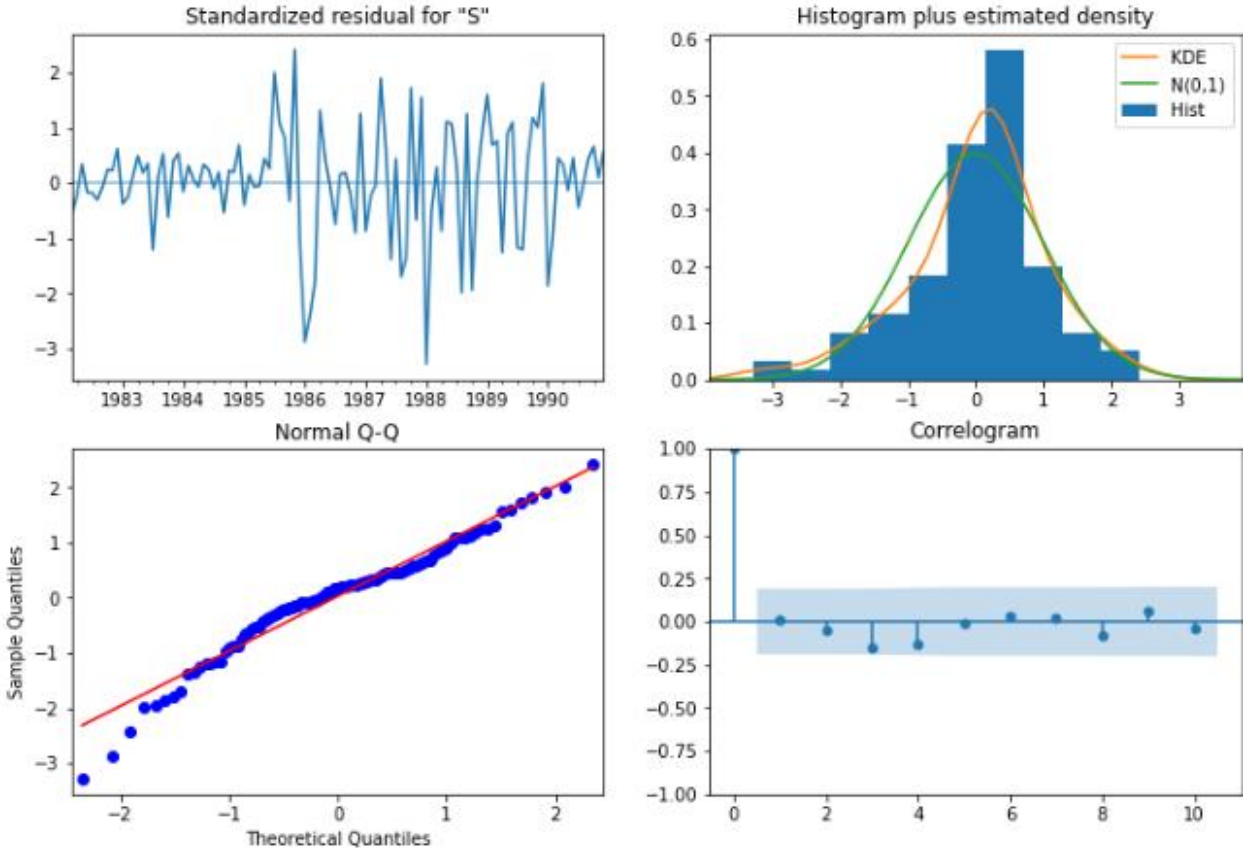
Dep. Variable:	Shoe_Sales	No. Observations:	130
Model:	SARIMAX(1, 2, 3)x(1, 0, 3, 6)	Log Likelihood	-596.426
Date:	Fri, 07 Jan 2022	AIC	1210.851
Time:	11:27:48	BIC	1234.822
Sample:	03-01-1980 - 12-01-1990	HQIC	1220.567
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.3239	0.094	-3.434	0.001	-0.509	-0.139
ma.L1	-2.9465	0.316	-9.322	0.000	-3.566	-2.327
ma.L2	2.8978	0.622	4.661	0.000	1.679	4.116
ma.L3	-0.9513	0.309	-3.079	0.002	-1.557	-0.346
ar.S.L6	-1.0315	0.038	-26.811	0.000	-1.107	-0.956
ma.S.L6	1.1490	0.336	3.423	0.001	0.491	1.807
ma.S.L12	0.4111	0.163	2.524	0.012	0.092	0.730
ma.S.L18	0.3488	0.108	3.220	0.001	0.136	0.561
sigma2	3183.2667	1561.911	2.038	0.042	121.977	6244.557

Ljung-Box (L1) (Q):	0.01	Jarque-Bera (JB):	11.61
Prob(Q):	0.93	Prob(JB):	0.00
Heteroskedasticity (H):	6.25	Skew:	-0.62
Prob(H) (two-sided):	0.00	Kurtosis:	4.04

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

After applying the best parameters to the model, we get the above model.

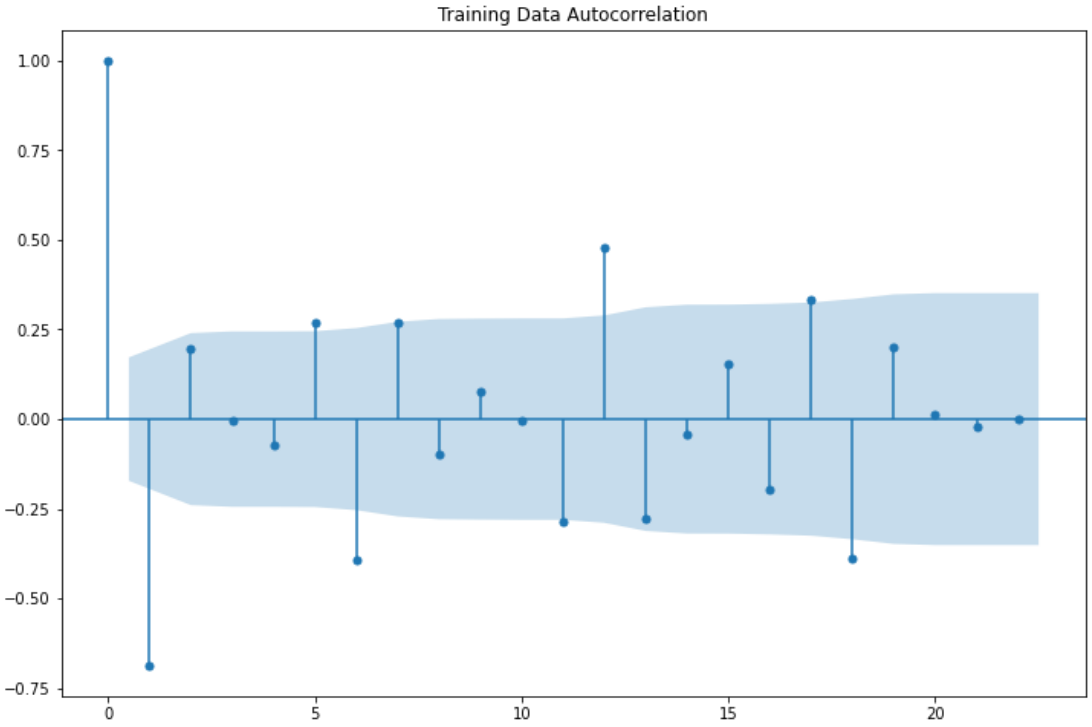


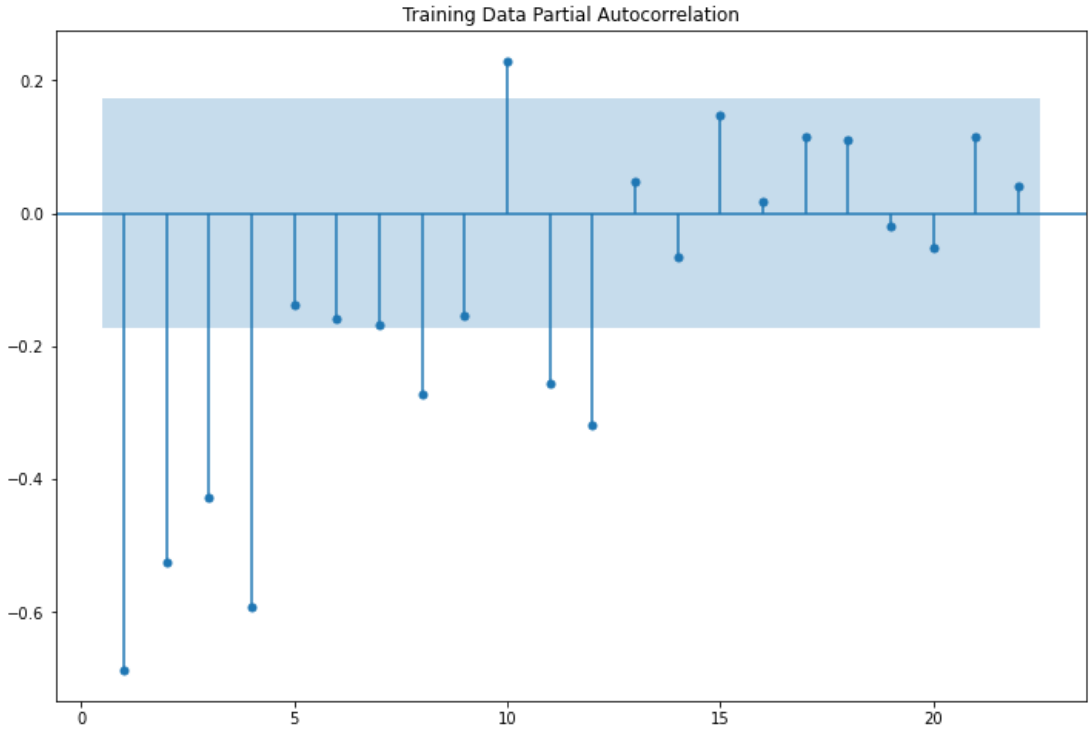
The diagnostic function was applied to the output.

```
RMSE: 270.7848081031495
MAPE: 105.69636589335285
```

Above is the deviation RMSE & MAPE values against the test data.

7. **Build ARIMA/SARIMA models based on the cut-off points of ACF and PACF on the training data and evaluate this model on the test data using RMSE.**
   ARIMA by looking at ACF & PACF





Here, we have taken alpha=0.05.

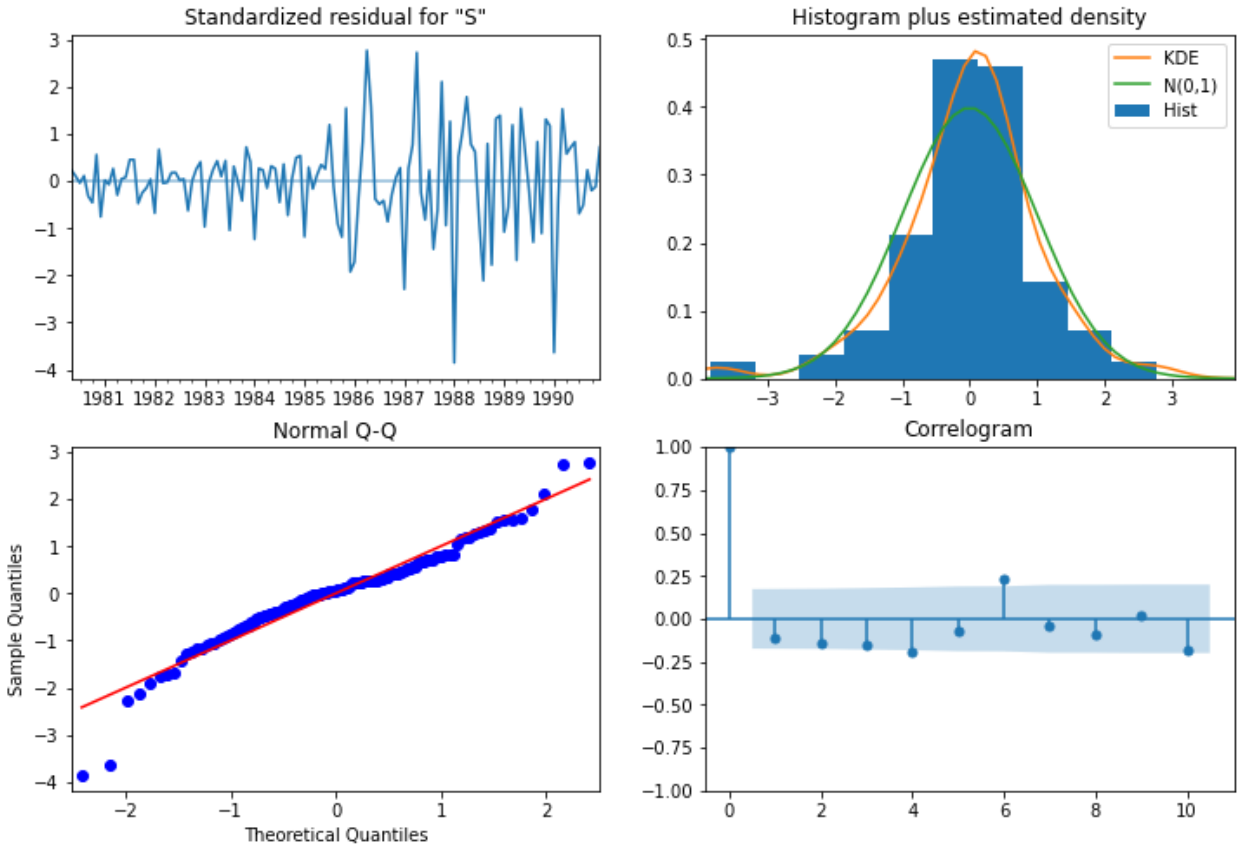
- The Auto-Regressive parameter in an ARIMA model is 'p' which comes from the significant lag before which the PACF plot cuts-off to 4.
- The Moving-Average parameter in an ARIMA model is 'q' which comes from the significant lag before the ACF plot cuts-off to 1.
- The difference value remains the same as 2.

By looking at the above plots, we will take the value of p and q to be 4 and 1 respectively.

```
                               SARIMAX Results
==============================================================================
Dep. Variable:             Shoe_Sales   No. Observations:                  130
Model:                 ARIMA(4, 2, 1)   Log Likelihood                -770.175
Date:                Thu, 06 Jan 2022   AIC                           1552.351
Time:                        23:57:28   BIC                           1569.463
Sample:                    03-01-1980   HQIC                          1559.303
                         - 12-01-1990
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.5262      0.064    -23.888      0.000      -1.651      -1.401
ar.L2         -1.5482      0.097    -15.962      0.000      -1.738      -1.358
ar.L3         -1.1761      0.102    -11.575      0.000      -1.375      -0.977
ar.L4         -0.5887      0.065     -9.098      0.000      -0.716      -0.462
ma.L1         -1.0000      0.104     -9.631      0.000      -1.203      -0.796
sigma2      9003.0492   1.15e-05   7.81e+08      0.000    9003.049    9003.049
===================================================================================
Ljung-Box (L1) (Q):                   1.52   Jarque-Bera (JB):                40.23
Prob(Q):                              0.22   Prob(JB):                         0.00
Heteroskedasticity (H):              11.00   Skew:                            -0.60
Prob(H) (two-sided):                  0.00   Kurtosis:                         5.47
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 2.64e+24. Standard errors may be unstable.
```

Based on the parameters identified (4,2,1), the fitting on the model was built. Above is the summary of the same.



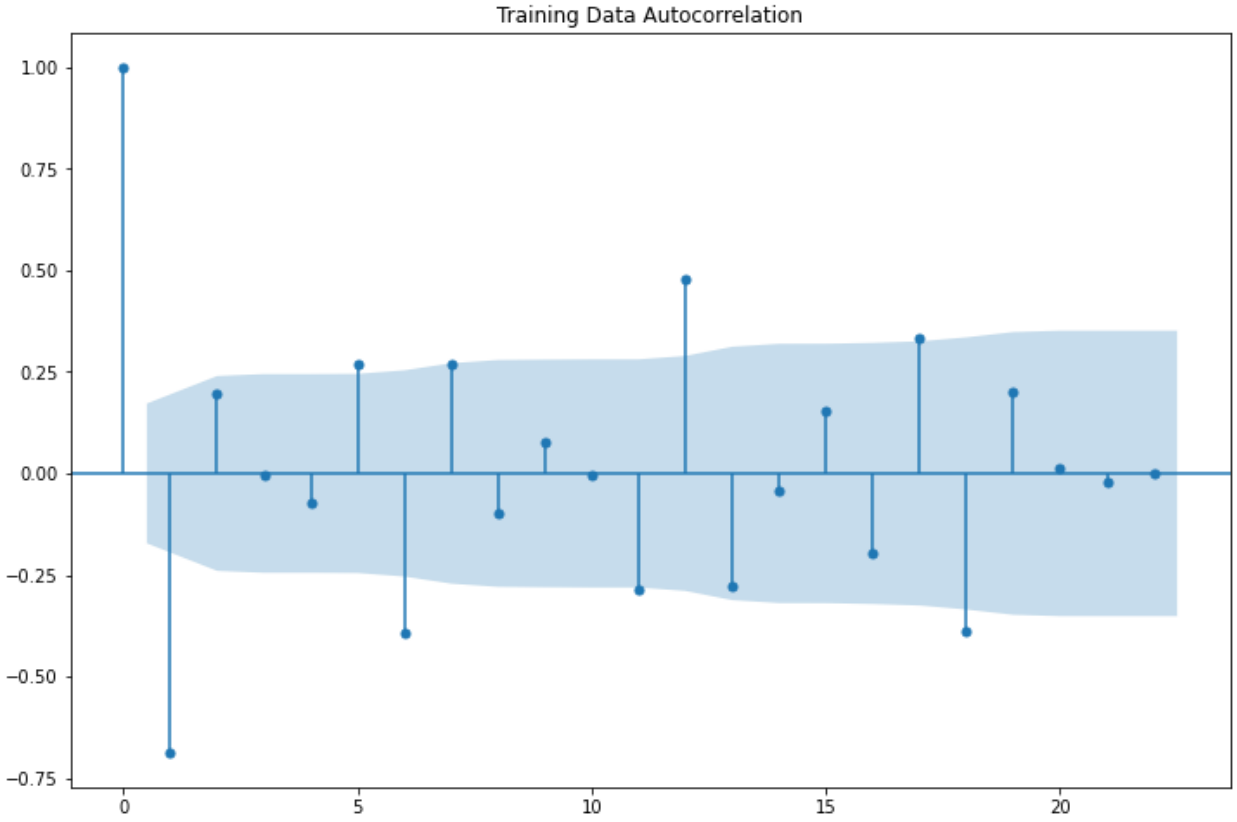
The diagnostic function was used on the train data. Above are the results for the same.

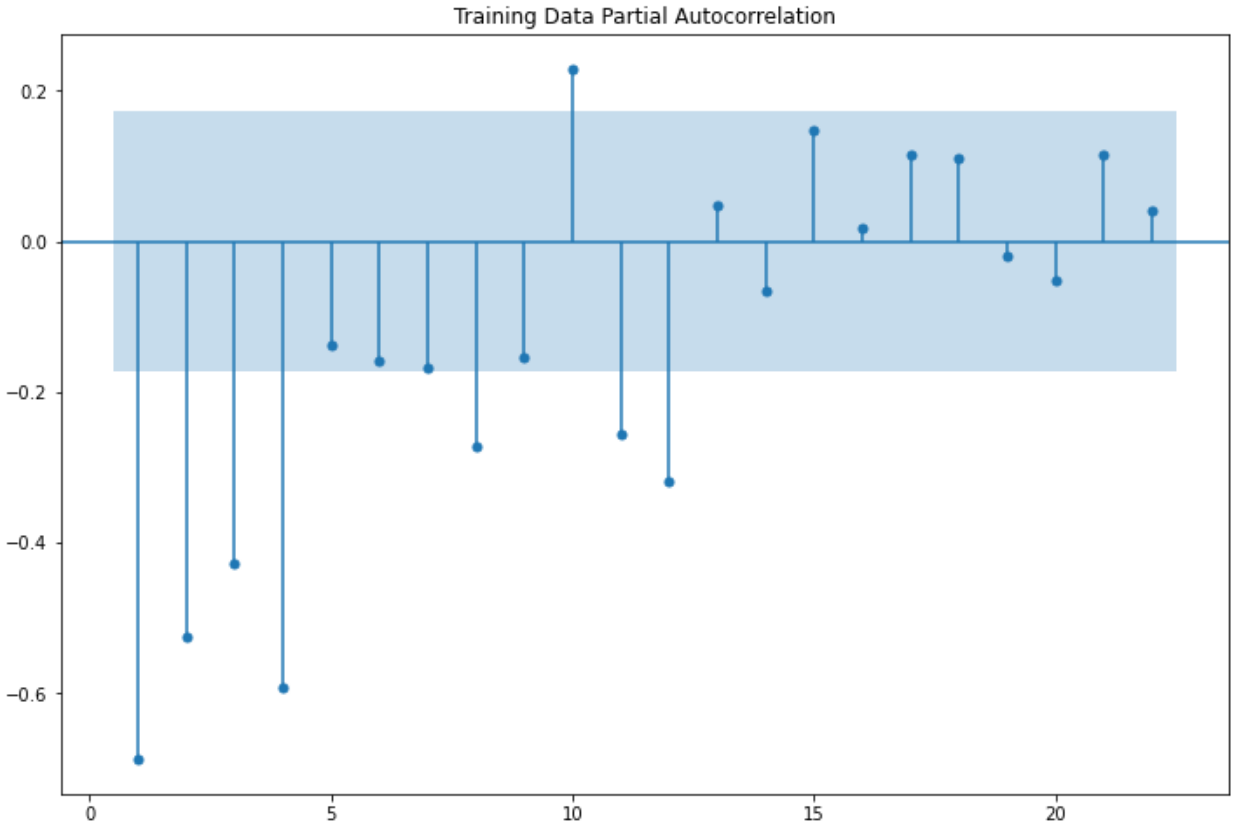
Now comes the phase where we shall check the prediction on the test data and get the accuracy of the model.

```
RMSE: 224.55680513863803
MAPE: 92.50022397627056
```

Above is the deviation RMSE & MAPE values against the test data.

SARIMA by looking at ACF & PACF


Training Data Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
0
5
10
15
20


Training Data Partial Autocorrelation
0.2
0.0
−0.2
−0.4
−0.6
0
5
10
15
20

Here, we have taken alpha=0.05.

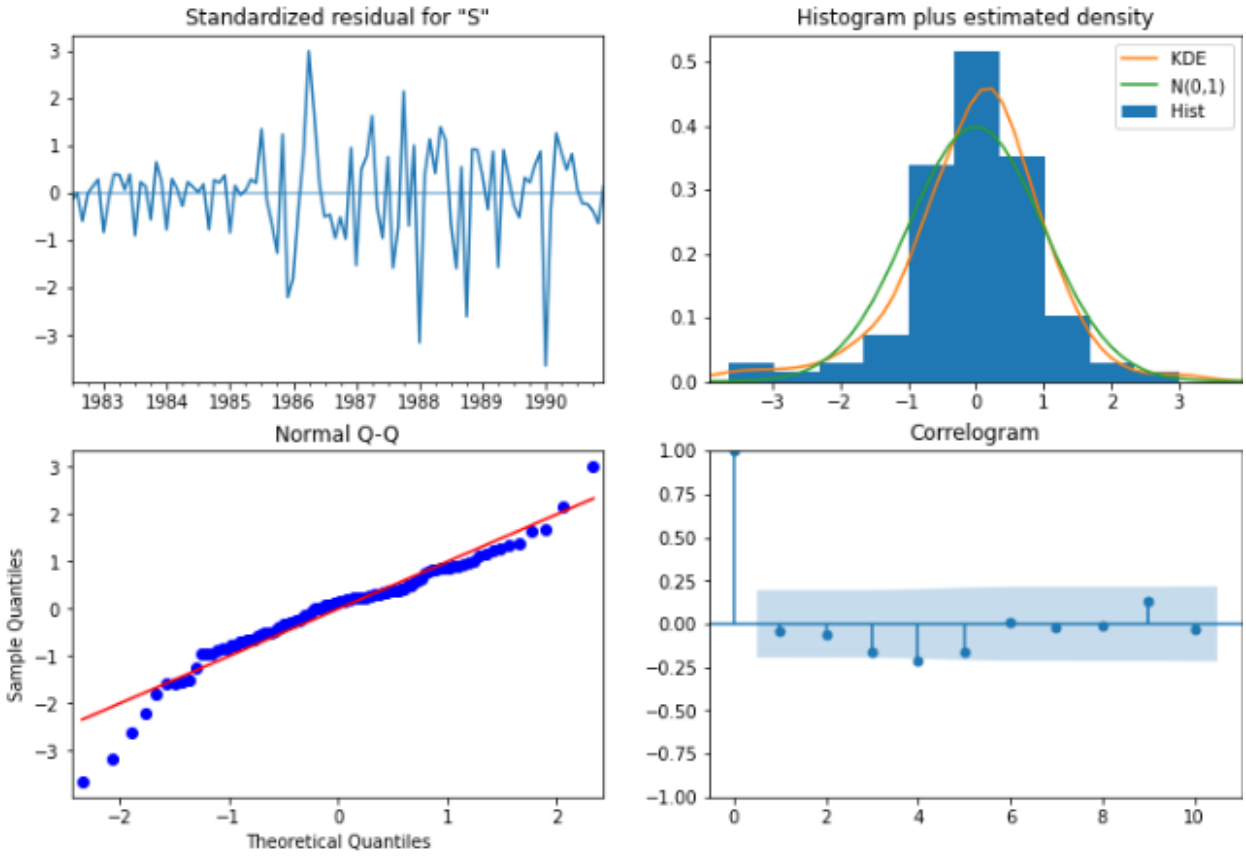
We are going to take the seasonal period as 6. We are taking the p value to be 4 and the q value also to be 1 as the parameters same as the ARIMA model.

- The Auto-Regressive parameter in an SARIMA model is 'P' which comes from the significant lag after which the PACF plot cuts-off to 0.
- The Moving-Average parameter in an SARIMA model is 'Q' which comes from the significant lag after which the ACF plot cuts-off to 4.
- Here the value of D=0

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                             Shoe_Sales   No. Observations:                130
Model:             SARIMAX(4, 2, 1)x(0, 0, [1, 2, 3, 4], 6)   Log Likelihood        -609.218
Date:                               Fri, 07 Jan 2022   AIC                           1238.437
Time:                                       11:28:11   BIC                           1264.686
Sample:                                   03-01-1980   HQIC                          1249.066
                                        - 12-01-1990
Covariance Type:                                opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.5860      0.070    -22.565      0.000      -1.724      -1.448
ar.L2         -1.6339      0.118    -13.824      0.000      -1.866      -1.402
ar.L3         -1.3245      0.124    -10.705      0.000      -1.567      -1.082
ar.L4         -0.6950      0.089     -7.807      0.000      -0.869      -0.520
ma.L1         -1.0000    824.542     -0.001      0.999   -1617.072    1615.072
ma.S.L6      -82.7761   1.84e+04     -0.005      0.996   -3.61e+04     3.6e+04
ma.S.L12     -23.9611   5165.445     -0.005      0.996   -1.01e+04    1.01e+04
ma.S.L18     -31.1694   6845.219     -0.005      0.996   -1.34e+04    1.34e+04
ma.S.L24      -2.9780    578.605     -0.005      0.996   -1137.023    1131.067
sigma2         1.2559   1131.383      0.001      0.999   -2216.214    2218.726
===================================================================================
Ljung-Box (L1) (Q):                   0.19   Jarque-Bera (JB):                27.96
Prob(Q):                              0.66   Prob(JB):                         0.00
Heteroskedasticity (H):               6.92   Skew:                            -0.68
Prob(H) (two-sided):                  0.00   Kurtosis:                         5.18
===================================================================================

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

The model was applied to the parameters.



The diagnostic function was used on the train data. Above are the results for the same.

```
RMSE: 248.28653369654742
MAPE: 102.89823060791588
```

Above is the deviation RMSE & MAPE values against the test data.

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

	Test RMSE
Alpha=0.605049,SES	196.404836
Alpha=1,Beta=0.0189:DES	266.161208
Alpha=0.570714,Beta=0.0001,Gamma=0.293721:TES(Additive)	128.992526
Alpha=0.571128,Beta=0.000147,Gamma=0.202947,Gamma=0:TES(Multiplicative)	83.734048
RegressionOnTime	266.276472
NaiveModel	245.121306
SimpleAverageModel	63.984570
2pointTrailingMovingAverage	45.948736
4pointTrailingMovingAverage	57.872686
6pointTrailingMovingAverage	63.456893
9pointTrailingMovingAverage	67.723648
ARIMA(1,2,3) Lowest AIC	243.931319
ARIMA(4,2,1)By looking ACF and PACF	224.556805
SARIMA(1,2,3)(1,0,3,6)Lowest AIC	270.784808
SARIMA(4,2,1)(0,0,4,6)By looking ACF and PACF	248.286534

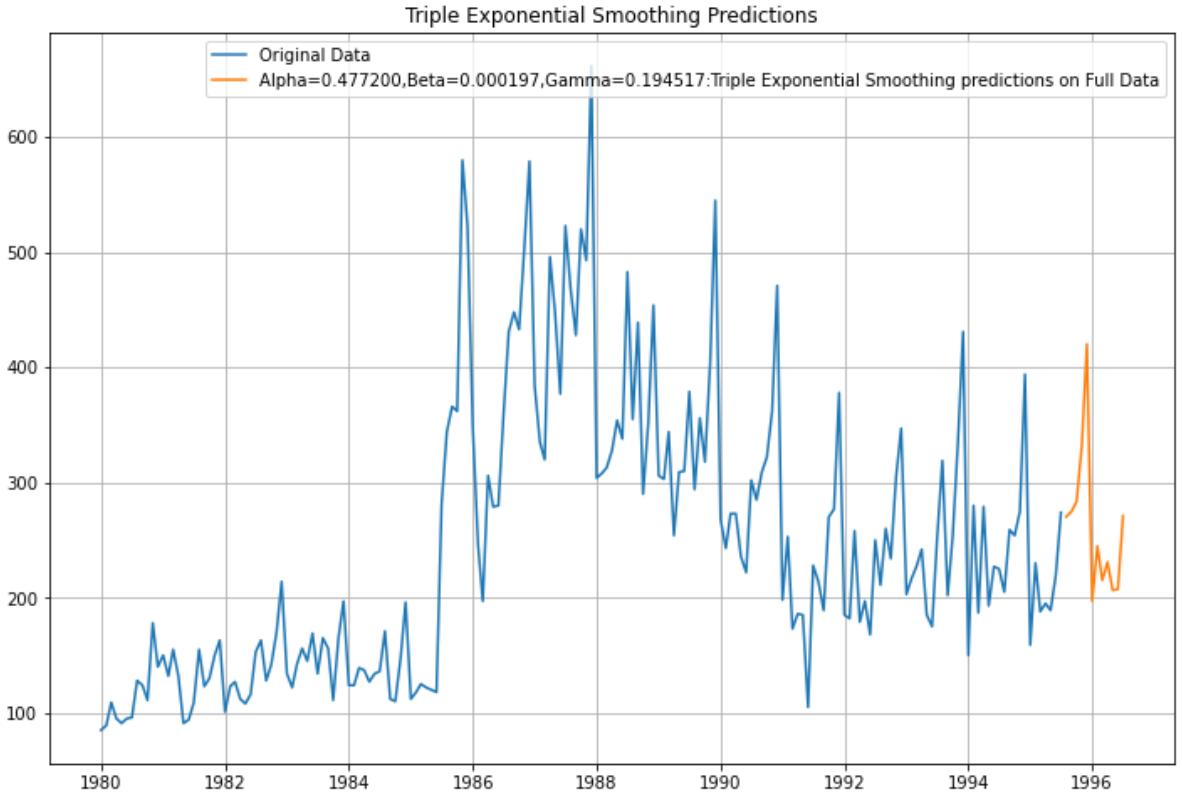
9. **Based on the model-building exercise, build the most optimum model(s) on the complete data and predict 12 months into the future with appropriate confidence intervals/bands.**

Based on the model building exercise, we shall select two models-

- Triple Exponential Smoothing
  This model was considered because it gives very good RMSE value i.e. 83.734048. This model was applied on the Full data and the values were predicted for the upcoming 12 months. From 01-August-1995 to 01-July-1996.

```
1995-08-01    270.249393
1995-09-01    274.818008
1995-10-01    283.391171
1995-11-01    331.225724
1995-12-01    420.357117
1996-01-01    197.049744
1996-02-01    244.959895
1996-03-01    215.137908
1996-04-01    230.916631
1996-05-01    206.490242
1996-06-01    207.435175
1996-07-01    270.997005
Freq: MS, dtype: float64
```

Above figure gives the predicted values for the next 12 months.



The above plot shows the original data and the predicted data for next 12 months.

- ARIMA (4,2,1) By looking ACF and PACF

  Firstly the entire series was checked for stationarity.

```
DataFrame test statistic is -1.577
DataFrame test p-value is 0.8014186234536529
Number of lags used 13
```

  We can see that according to p-value, the series is not stationary.

```
DF test statistic is -3.532
DF test p-value is 0.03611703400135922
Number of lags used 12
```

First order differencing was done to make the series stationary.

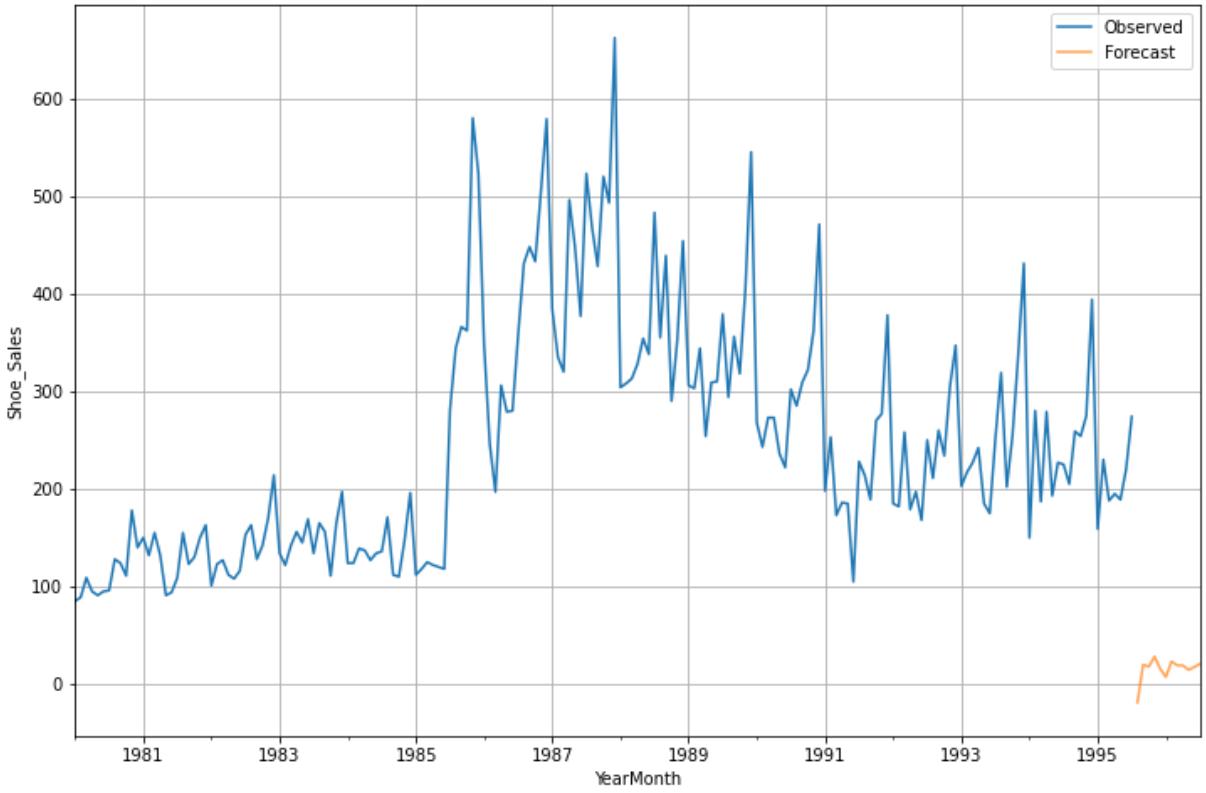
```
                               SARIMAX Results
==============================================================================
Dep. Variable:             Shoe_Sales   No. Observations:                  186
Model:                 ARIMA(4, 2, 1)   Log Likelihood               -1076.086
Date:                Fri, 07 Jan 2022   AIC                           2164.173
Time:                        11:49:24   BIC                           2183.462
Sample:                    02-01-1980   HQIC                          2171.991
                         - 07-01-1995
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.1747      0.059    -19.955      0.000      -1.290      -1.059
ar.L2         -1.0242      0.093    -11.024      0.000      -1.206      -0.842
ar.L3         -0.7684      0.095     -8.106      0.000      -0.954      -0.583
ar.L4         -0.3725      0.065     -5.691      0.000      -0.501      -0.244
ma.L1         -1.0000     48.114     -0.021      0.983     -95.301      93.301
sigma2      6665.6153   3.21e+05      0.021      0.983   -6.22e+05    6.35e+05
===================================================================================
Ljung-Box (L1) (Q):                   0.12   Jarque-Bera (JB):                82.74
Prob(Q):                              0.72   Prob(JB):                         0.00
Heteroskedasticity (H):               7.05   Skew:                            -1.15
Prob(H) (two-sided):                  0.00   Kurtosis:                         5.34
===================================================================================

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

The ARIMA model was applied to the entire dataset. Above is the summary.

Shoe_Sales	mean	mean_se	mean_ci_lower	mean_ci_upper
1995-08-01	-18.833899	81.866634	-179.289552	141.621755
1995-09-01	19.926323	83.029861	-142.809213	182.661860
1995-10-01	17.876591	84.407877	-147.559809	183.312991
1995-11-01	28.326744	86.032906	-140.294653	196.948141
1995-12-01	15.841750	90.689143	-161.905704	193.589205
1996-01-01	7.285493	94.574416	-178.076957	192.647943
1996-02-01	23.200987	94.713019	-162.433120	208.835093
1996-03-01	19.311566	97.502465	-171.789754	210.412887
1996-04-01	19.148811	99.862664	-176.578413	214.876035
1996-05-01	14.625026	102.426679	-186.127576	215.377628
1996-06-01	17.509332	103.936956	-186.203358	221.222022
1996-07-01	20.671513	105.378834	-185.867206	227.210232

Above are the predictions by the model for the next 12 months.



The above graph represents the original data along with forecast of the next 12 months.

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

Recommendation

- The management should rely on Triple Exponential Smoothing method for predicting the future sales because it has given the best predictions amongst all the models.
- The management should introduce good combo offers in the month of November and December. The sales for December have been historically very good.
- There is strong seasonality according to past performance; therefore offseason discount schemes should be launched to boost the sales in these non- performing months.

Problem 2 for the Data Set: Softdrink

You are an analyst in the RST soft drink company and you are expected to forecast the sales of the production of the soft drink for the upcoming 12 months from where the data ends. The data for the production of soft drink has been given to you from January 1980 to July 1995.

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

   The data was uploaded to the Jupyter Notebook. The data was retrieved using CSV file. The name of the file was ' SoftDrink.csv'. The data has two columns i.e. 'SoftDrink' and 'YearMonth'. As understood from name only, the column 'YearMonth' has time therefore it was made as index column while reading the data. Also, parse_dates was used to tell Python that the column 'YearMonth' has time data.

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

   After loading the CSV file, it was understood that the dataset has 187 values, out of which there are 0 NULL values.

YearMonth	SoftDrinkProduction
1980-01-01	1954
1980-02-01	2302
1980-03-01	3054
1980-04-01	2414
1980-05-01	2226

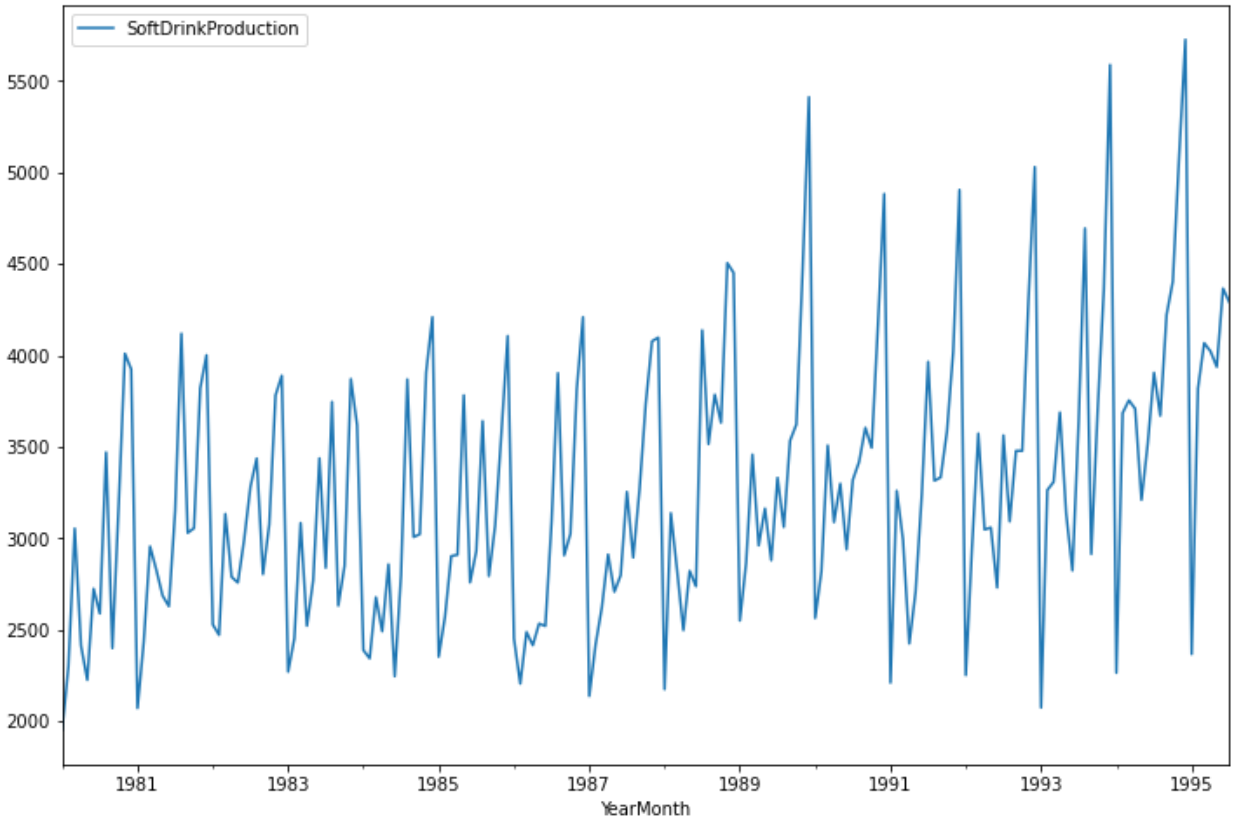
   Above are initial 5 data points.

YearMonth	SoftDrinkProduction
1995-03-01	4067
1995-04-01	4022
1995-05-01	3937
1995-06-01	4365
1995-07-01	4290

Above are the last 5 data points.
Here we can easily understand that this is monthly production data of a soft drink company. The data was collected on 1st day of every month from 01-January-1980 to 01-July-1995.

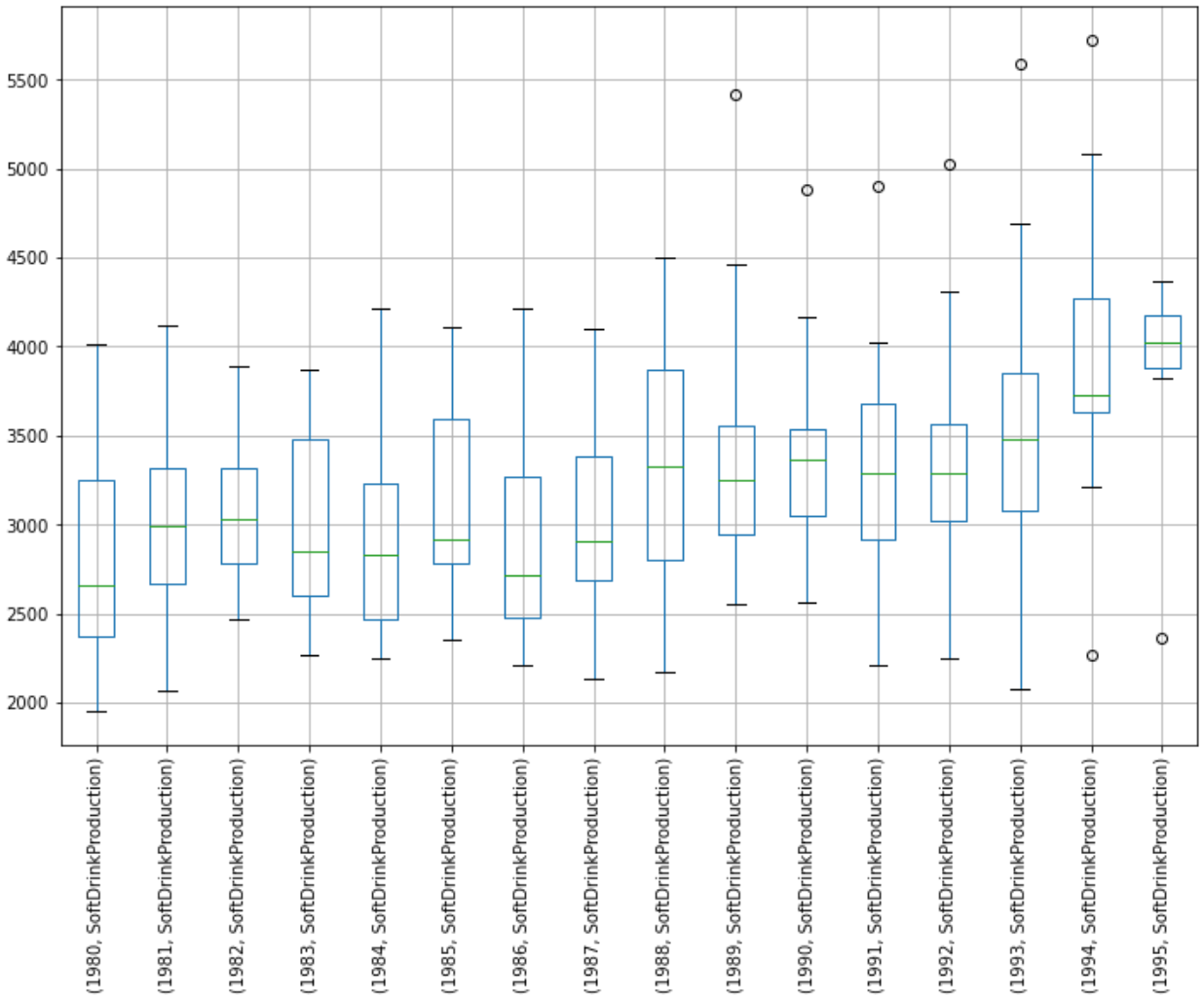
	SoftDrinkProduction
count	187.000
mean	3262.610
std	728.357
min	1954.000
25%	2748.000
50%	3134.000
75%	3741.000
max	5725.000

Above is the descriptive statistics of the dataset. We can see that the mean production of 187 months is 3262.61. Also, we can see that the minimum value is 1954 and maximum value is 5725.

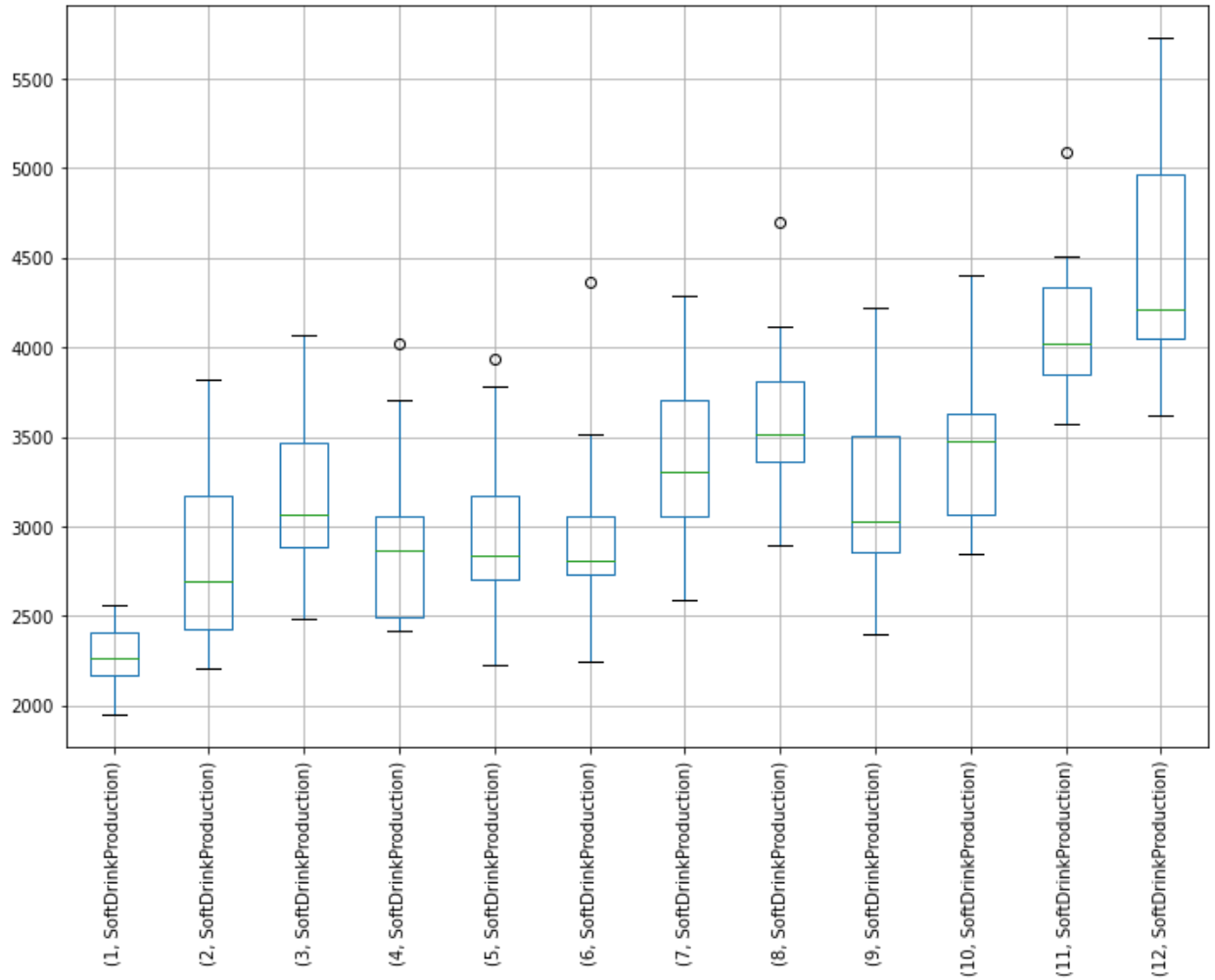


Above is the graph for the dataset. Here we cannot clearly see the trend and seasonality. Also, we can clearly see that the production have increased during the time period between 1990 and 1995.

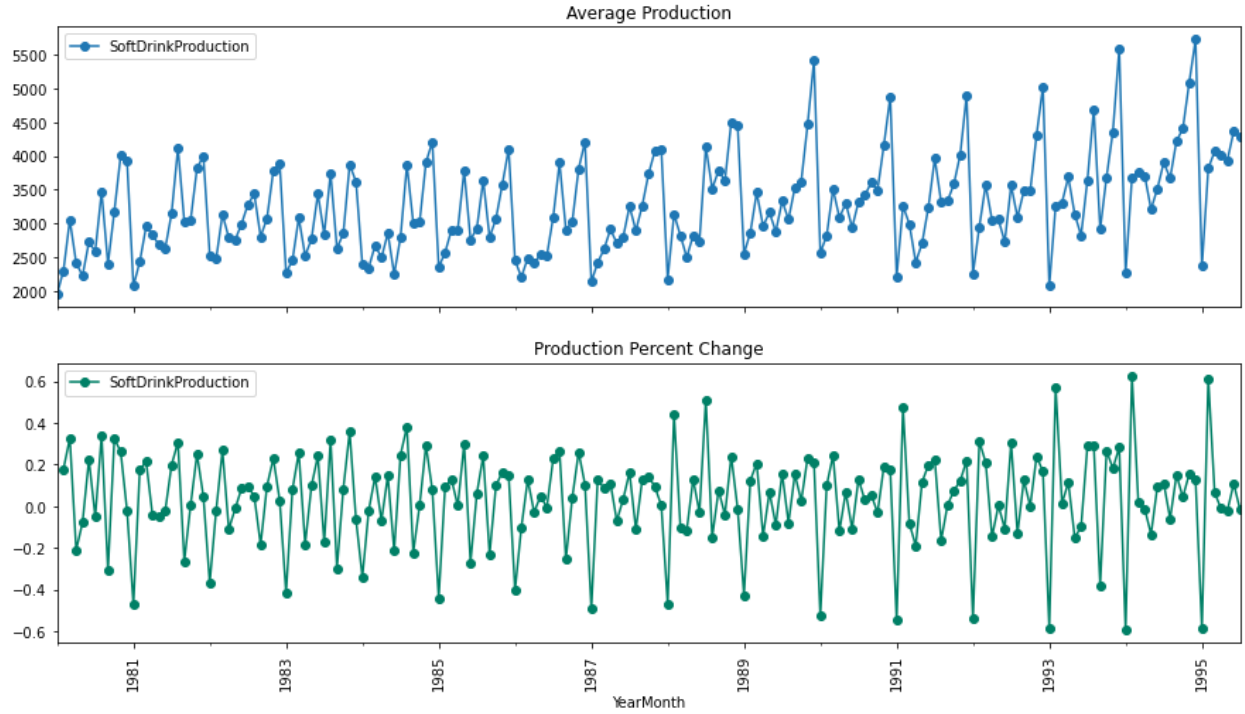
2. **Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.**



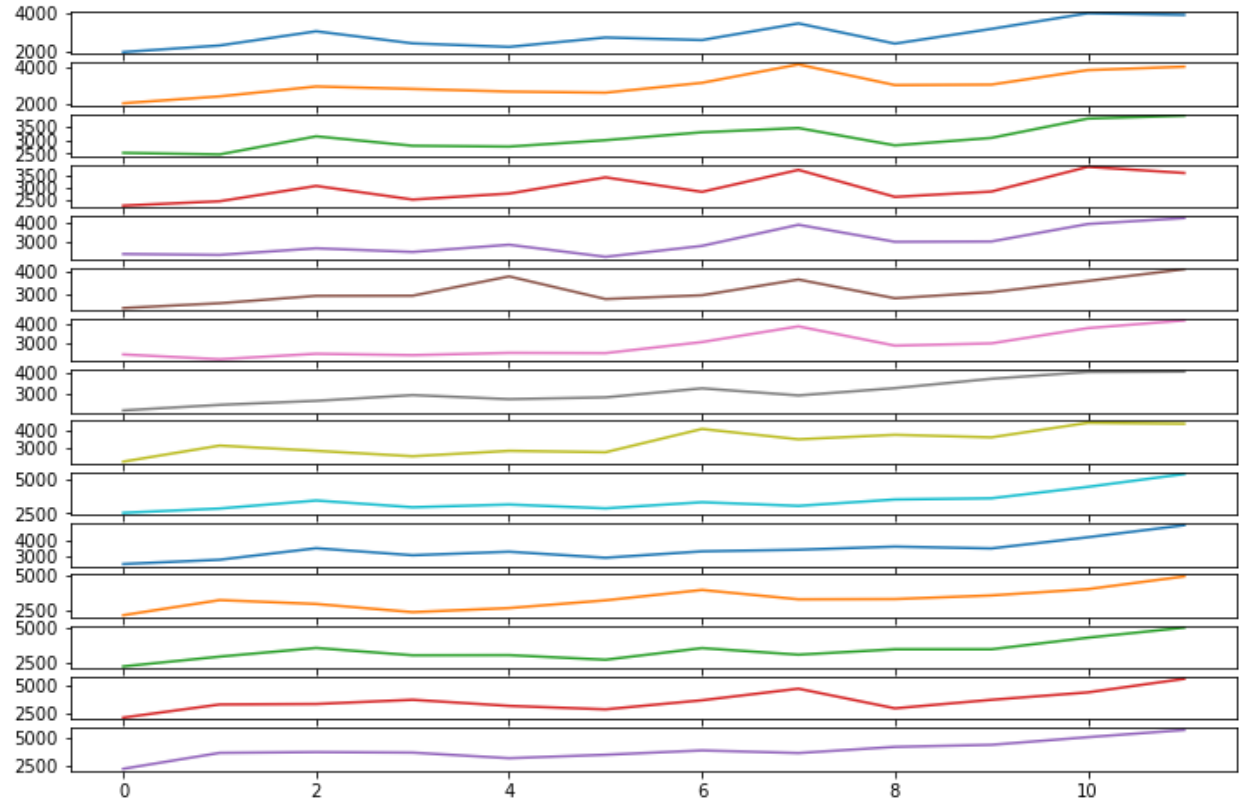
Above is the yearly production report for the company. Here we can see that the outliers are present for the years 1989, 1990, 1991, 1992, 1993 and 1994. The overall production is highest for year 1994 followed by 1993.



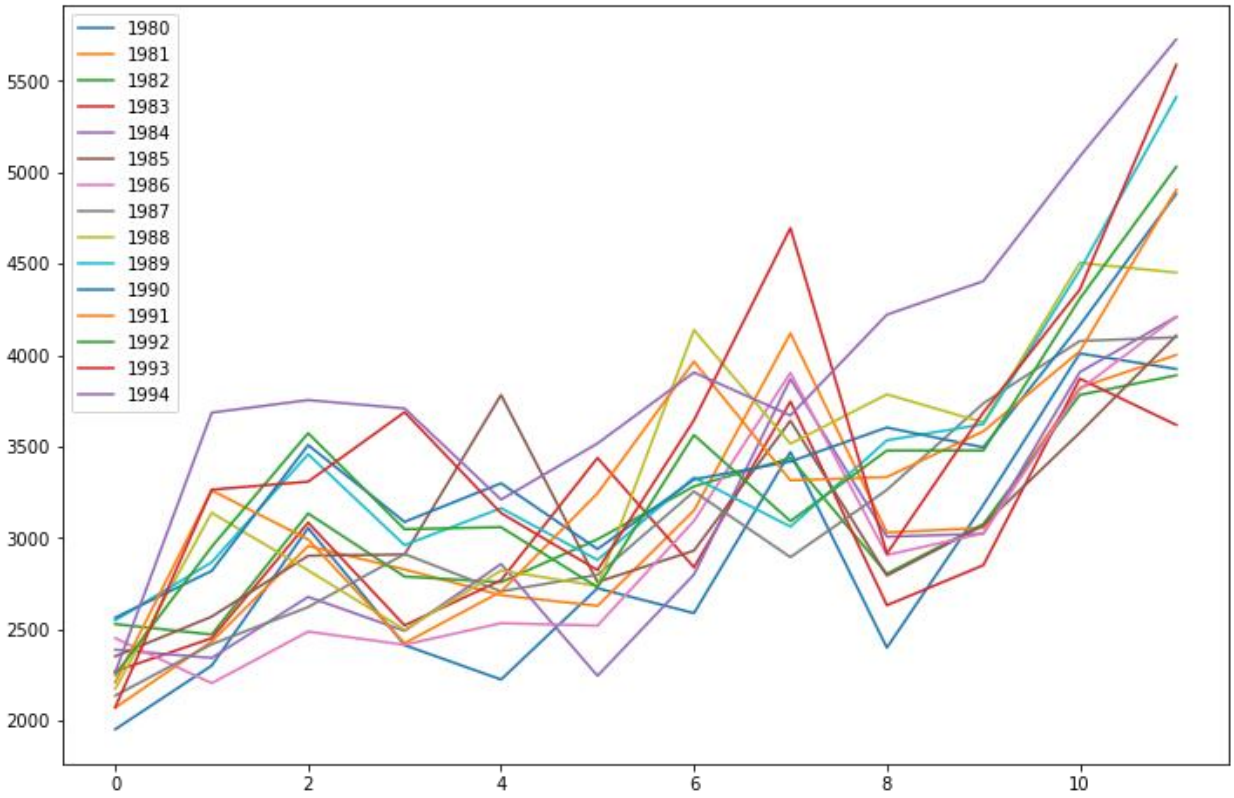
Above is the monthly production data between years 1980 and 1995. We can see that the monthly trend for all the months. December is the best performing month followed by November. Also, outlier presence can be seen in April, May, June, August and November production data.



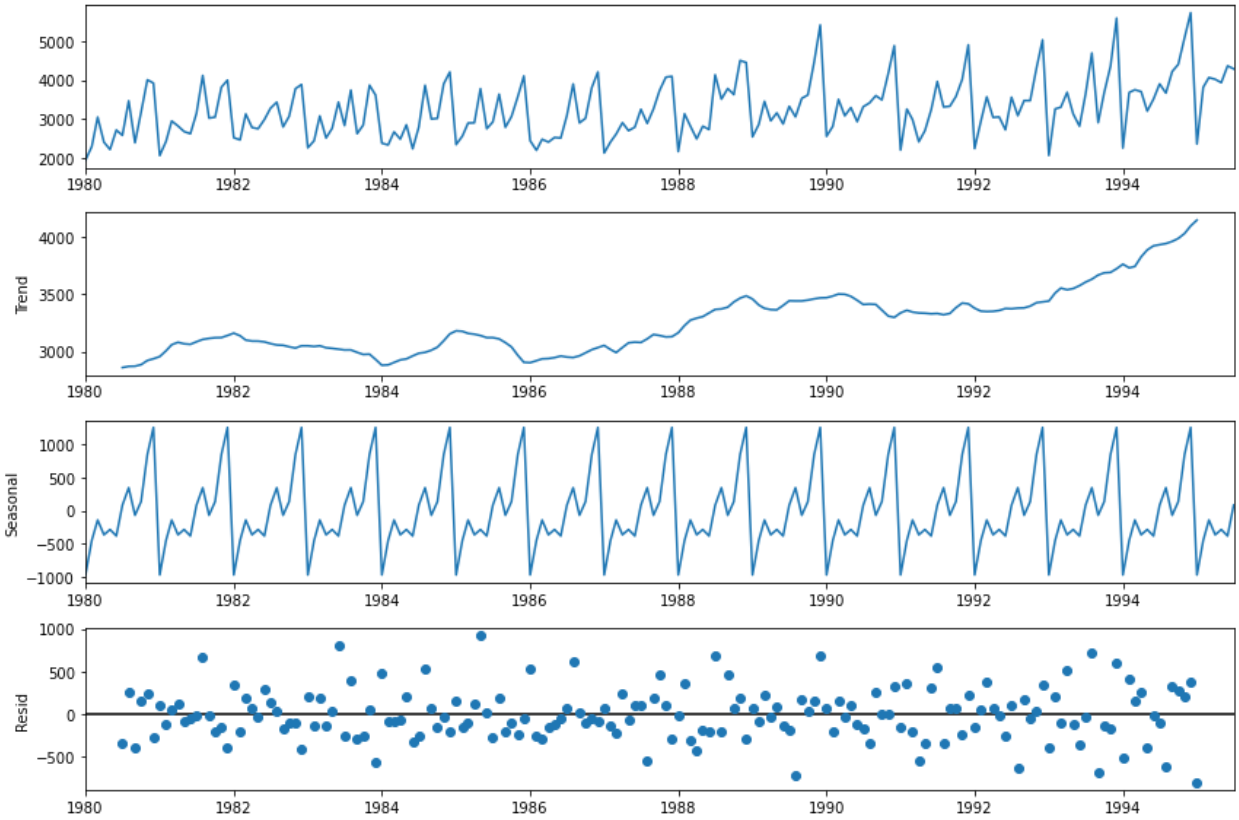
Above figure represents the yearly average production of the soft drink and the percentage change in the production. Interestingly, we can see overall increase in the production between years 1990 to 1995. But the percentage change in production remains constant.



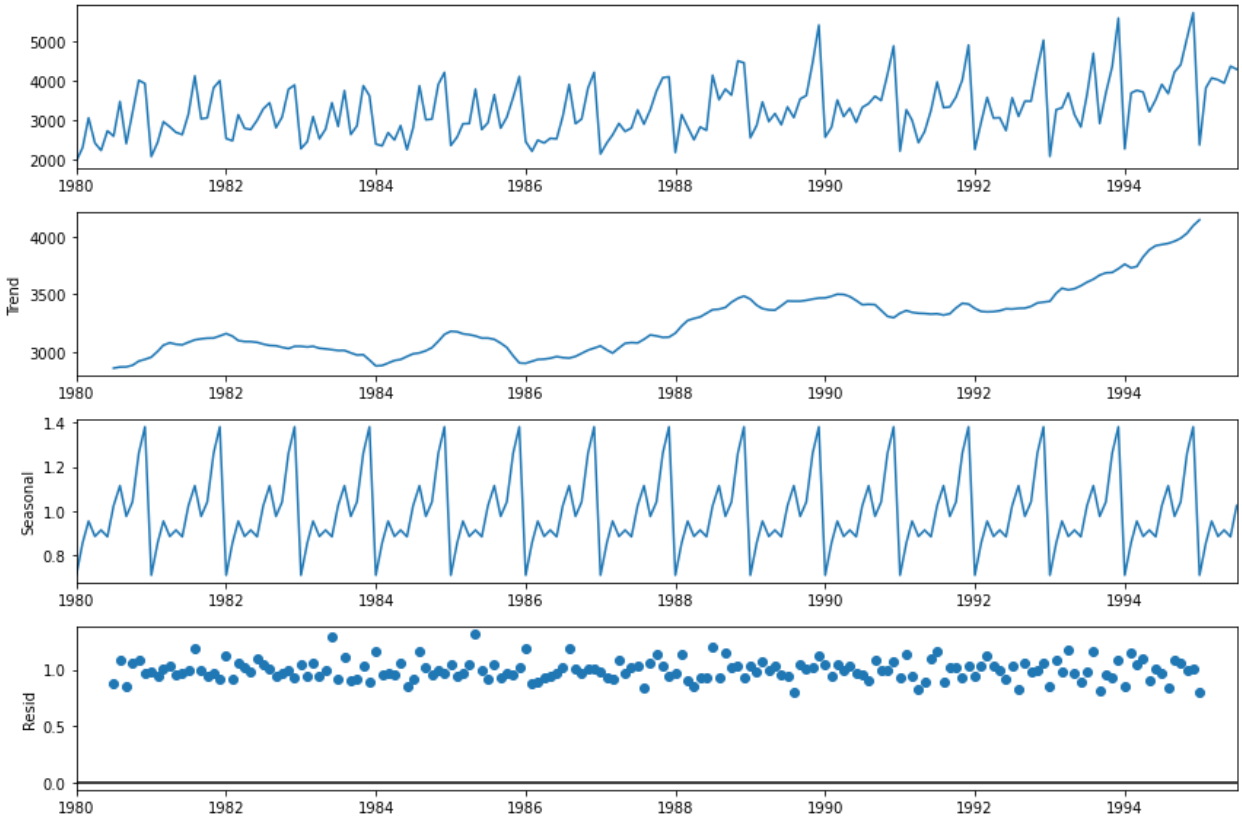
The above graph represents the production for the years from 1980 and 1994. It is hard to understand the seasonality or trend from this figure.



From above graph we can see some seasonality in the data.



Above is the additive decomposition of the time series dataset. Here we can clearly see the trend as well as seasonality in the data. According to the trend the overall production increased during the time period between 1992 and 1995. In case of seasonality, it is constant all across. The residue or the error component is random in nature which is good to go. If there would have been some pattern in the residue then we must have to go towards the multiplicative decomposition.



Above is the multiplicative decomposition of the time series. Here again we can see that there is clear visible seasonality in the data. The residue here is again randomly distributed. There's some trend visible in the multiplicative decomposition residue.

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

```
(132, 1)
(55, 1)
```

The data was split into two train and test. After splitting the data the train has 132 values and test has 55 values.

YearMonth	SoftDrinkProduction
1980-01-01	1954
1980-02-01	2302
1980-03-01	3054
1980-04-01	2414
1980-05-01	2226

Above is the head of the train data which starts with data from 1-January-1980.

YearMonth	SoftDrinkProduction
1991-01-01	2211
1991-02-01	3260
1991-03-01	2992
1991-04-01	2425
1991-05-01	2707

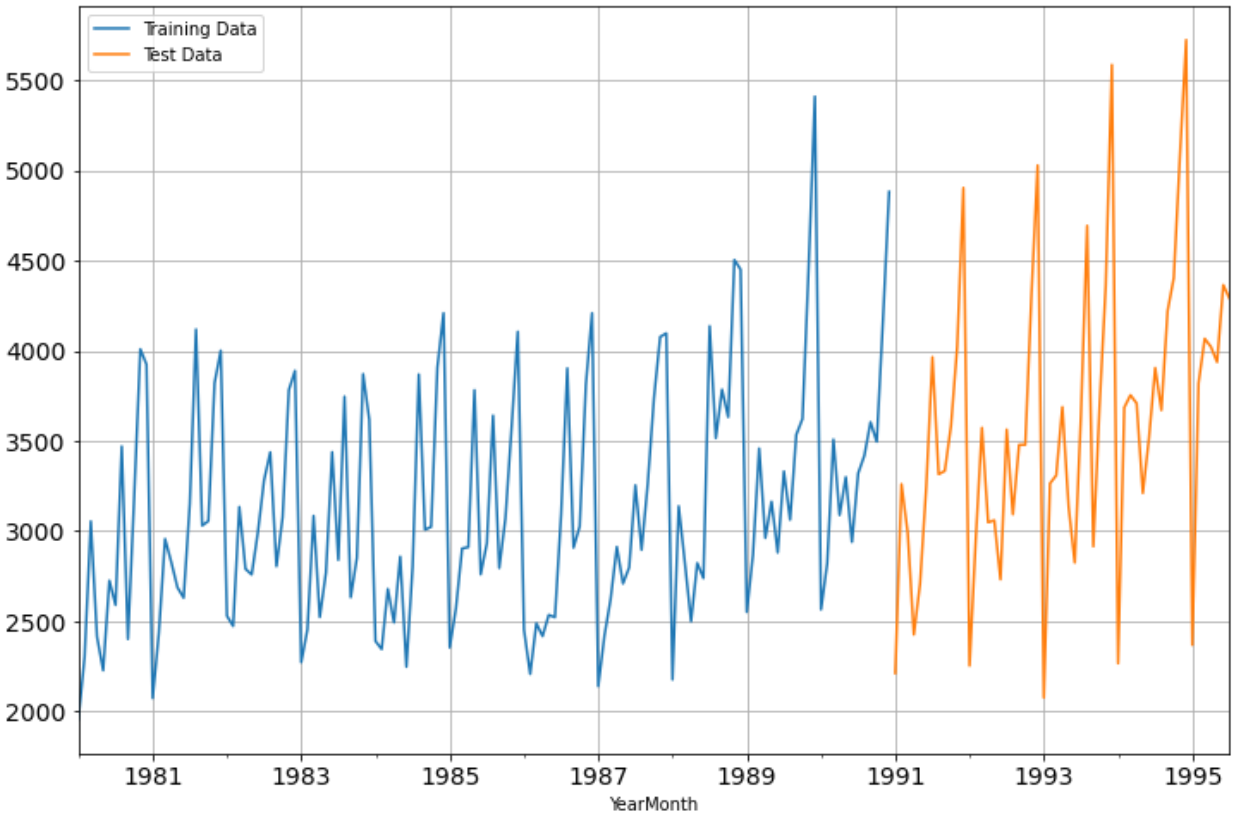
Above is the head of the test data which starts at 01-January-1991.

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 132 entries, 1980-01-01 to 1990-12-01
Data columns (total 1 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   SoftDrinkProduction  132 non-null    int64
dtypes: int64(1)
memory usage: 2.1 KB
```

Above figure confirms that all the values in the train data are non-NULL.

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 55 entries, 1991-01-01 to 1995-07-01
Data columns (total 1 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   SoftDrinkProduction  55 non-null     int64
dtypes: int64(1)
memory usage: 880.0 bytes
```

Above figure confirms that all the values in the test data are non-NULL.



In the above figure of entire time series, the train data till 1991 is colored in blue and the test data is colored in orange.

4. **Build various 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, simple average models etc. should also be built on the training data and check the performance on the test data using RMSE.

   Simple Exponential Smoothing (Level, No Seasonality, No Trend)

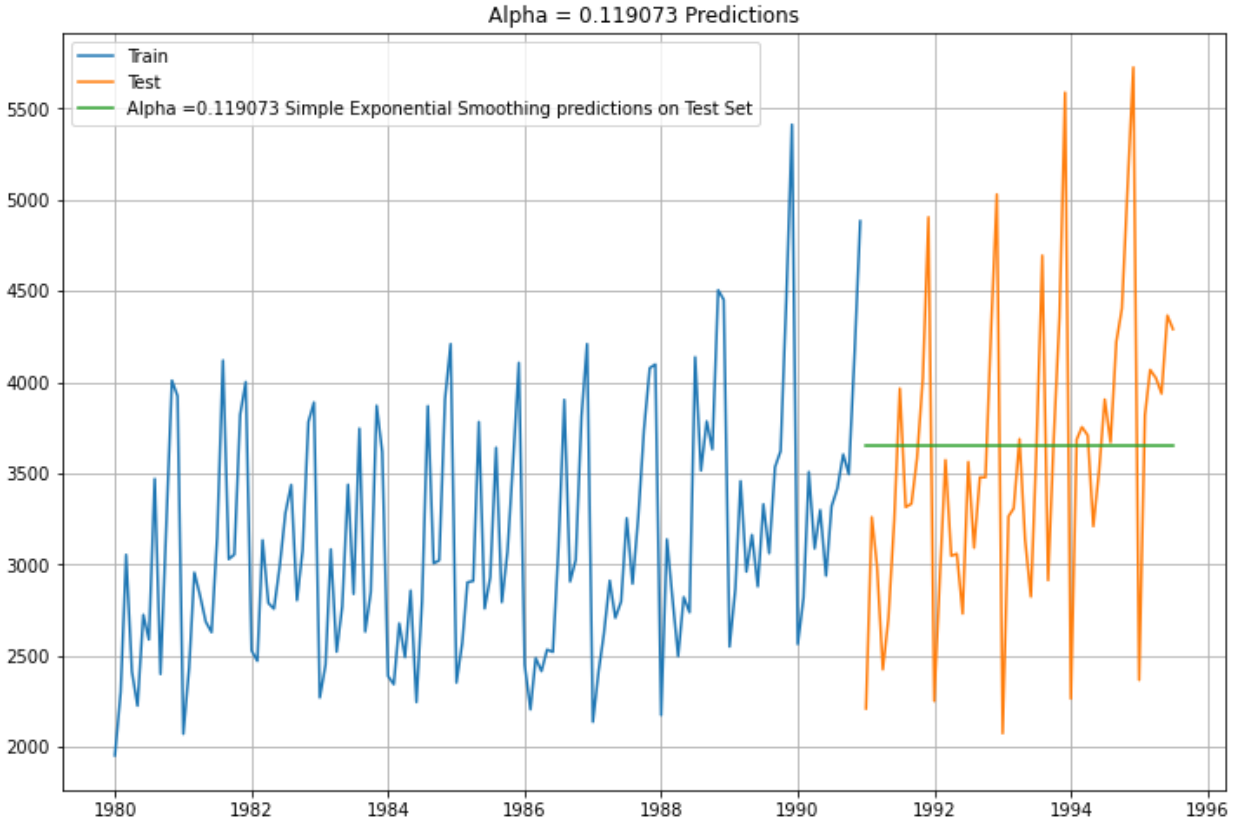
   The class was created on the train dataset for applying the Simple Exponential Smoothing. Fitting was done using fit() function. Here, the optimum parameters were chosen by the Python. Then, the parameters were checked using 'params' function.

```
{'smoothing_level': 0.11907309094689855,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 2573.0166666666655,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

Above are the parameters for SES. The value of Alpha (Smoothing_Level) is 0.119073.
The fitted model on the training set was used to forecast on the test set.

```
1991-01-01    3656.847753
1991-02-01    3656.847753
1991-03-01    3656.847753
1991-04-01    3656.847753
1991-05-01    3656.847753
1991-06-01    3656.847753
1991-07-01    3656.847753
1991-08-01    3656.847753
1991-09-01    3656.847753
1991-10-01    3656.847753
1991-11-01    3656.847753
1991-12-01    3656.847753
1992-01-01    3656.847753
1992-02-01    3656.847753
1992-03-01    3656.847753
1992-04-01    3656.847753
1992-05-01    3656.847753
1992-06-01    3656.847753
1992-07-01    3656.847753
1992-08-01    3656.847753
1992-09-01    3656.847753
1992-10-01    3656.847753
```

Above are the forecasted values on the test set. Value is constant i.e. 3656.847753.



Green colored line in the above figure is the estimated value which is 3656.847753 for the test time period from 01-January-1991 to 01-July-1995. Orange line is the actual value. We cannot consider this model as the gap between the actual and predicted is very high.

```
SES RMSE: 809.5016403931278
SES RMSE (calculated using statsmodels): 809.5016403931279
```

The Root Mean Square Error deviation for Simple Exponential Smoothing is 809.5016.

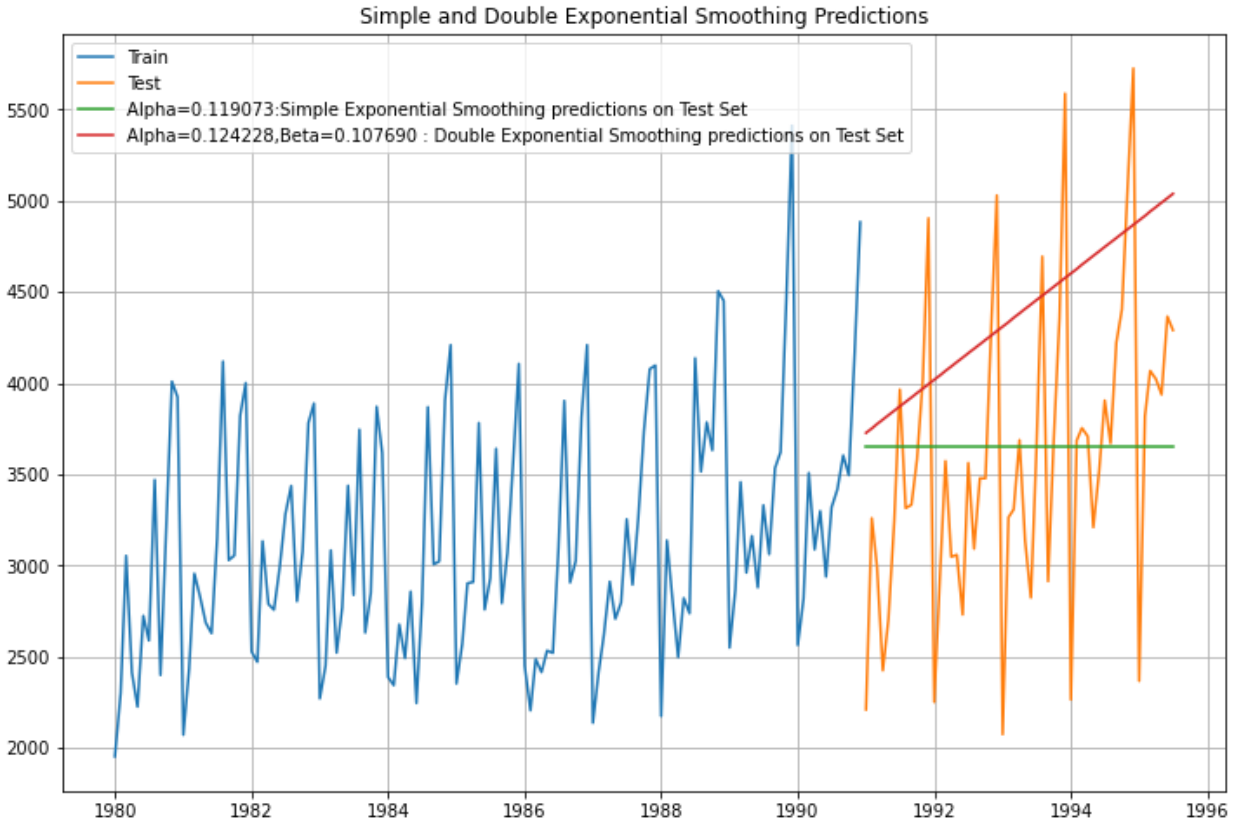
Double Exponential Smoothing (Level, Trend, No Seasonality)

The class was created on the train dataset for applying the Holt model Exponential Smoothing. Fitting was done using fit() function. Here, the optimum parameters were chosen by the Python. Then, the parameters were checked using 'params' function.

```
==Holt model Exponential Smoothing Estimated Parameters ==

{'smoothing_level': 0.1242286864966588, 'smoothing_trend': 0.10769076164072929, 'smoothing_seasonal': nan, 'damping_trend':
nan, 'initial_level': 2142.9200400852947, 'initial_trend': 42.27465415028941, 'initial_seasons': array([], dtype=float64),
'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

Value of Alpha (Smoothing_Level) is 0.124228 Value of Beta (Smoothing_Trend) is 0.107690.



Red colored line in the above figure represents the estimated values for the test time period from 01-January-1991 to 01-July-1995. Orange line is the actual value. We cannot consider this model as the gap between the actual and predicted is very high.

```
DES RMSE: 1074.3291531501832
```

The Root Mean Square Error deviation for Double Exponential Smoothing is 1074.32.

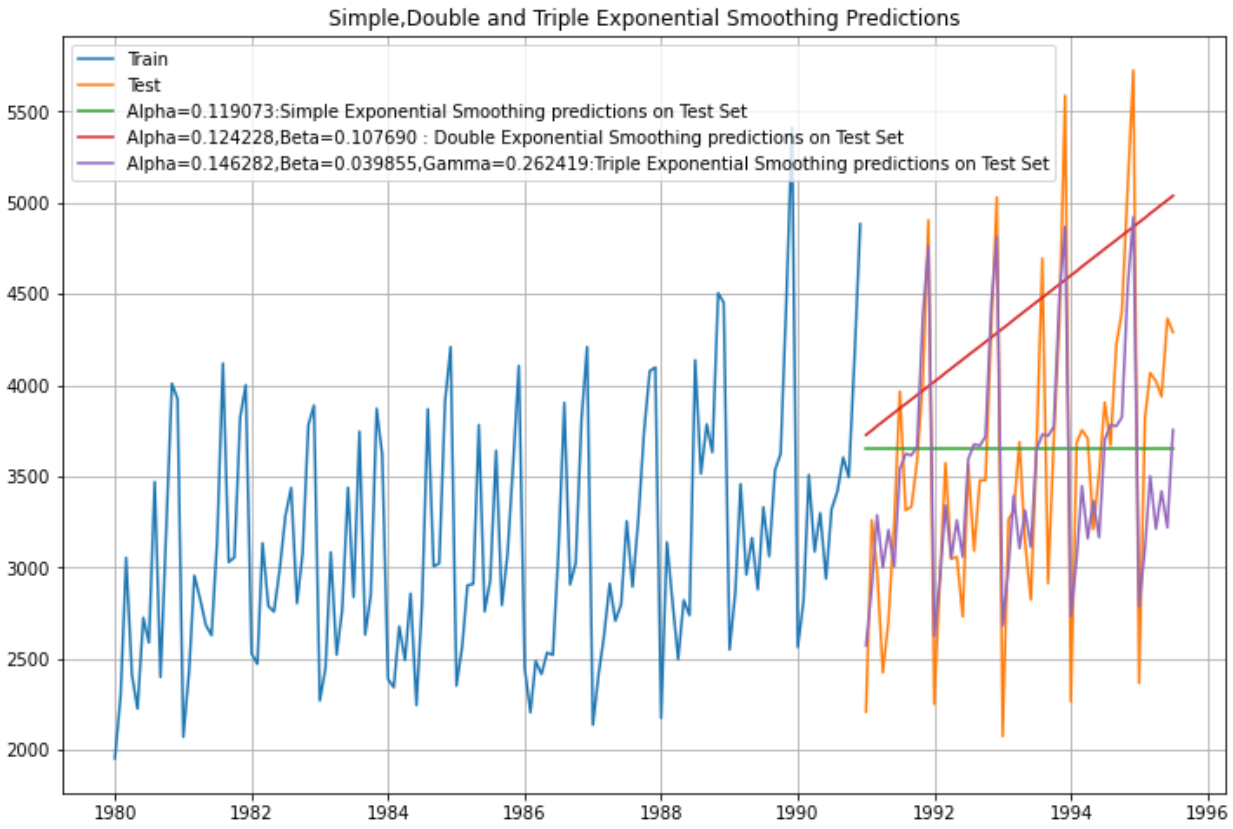
Triple Exponential Smoothing (Level, Trend, Seasonality- Additive)

The class was created on the train dataset for applying the Holt Winters model Exponential Smoothing. Fitting was done using fit() function. Here, the optimum parameters were chosen by the Python. Then, the parameters were checked using 'params' function. Here, the seasonality is additive.

```
==Holt Winters model Exponential Smoothing Estimated Parameters ==

{'smoothing_level': 0.14628214287204402, 'smoothing_trend': 0.03985523474431963, 'smoothing_seasonal': 0.2624197351602548,
'damping_trend': nan, 'initial_level': 2803.214611111109, 'initial_trend': 7.179638888889087, 'initial_seasons': array([-68
7.29896528, -582.87175694,  -55.66104861, -365.74079861,
       -253.26738194, -196.41738194,  -32.54725694,  690.31611806,
       -282.20021528,   44.75545139,  867.40386806,  853.53236806]), 'use_boxcox': False, 'lamda': None, 'remove_bias': Fal
se}
```

Value of Alpha: 0.146282 Value of Beta: 0.039855 Value of Gamma: 0.262419



Violet colored line in the above figure represents the estimated values for the test time period from 01-January-1991 to 01-July-1995. Orange line is the actual value. This model has performed better than the SES and DES.

```
TES RMSE: 458.9653920540907
```

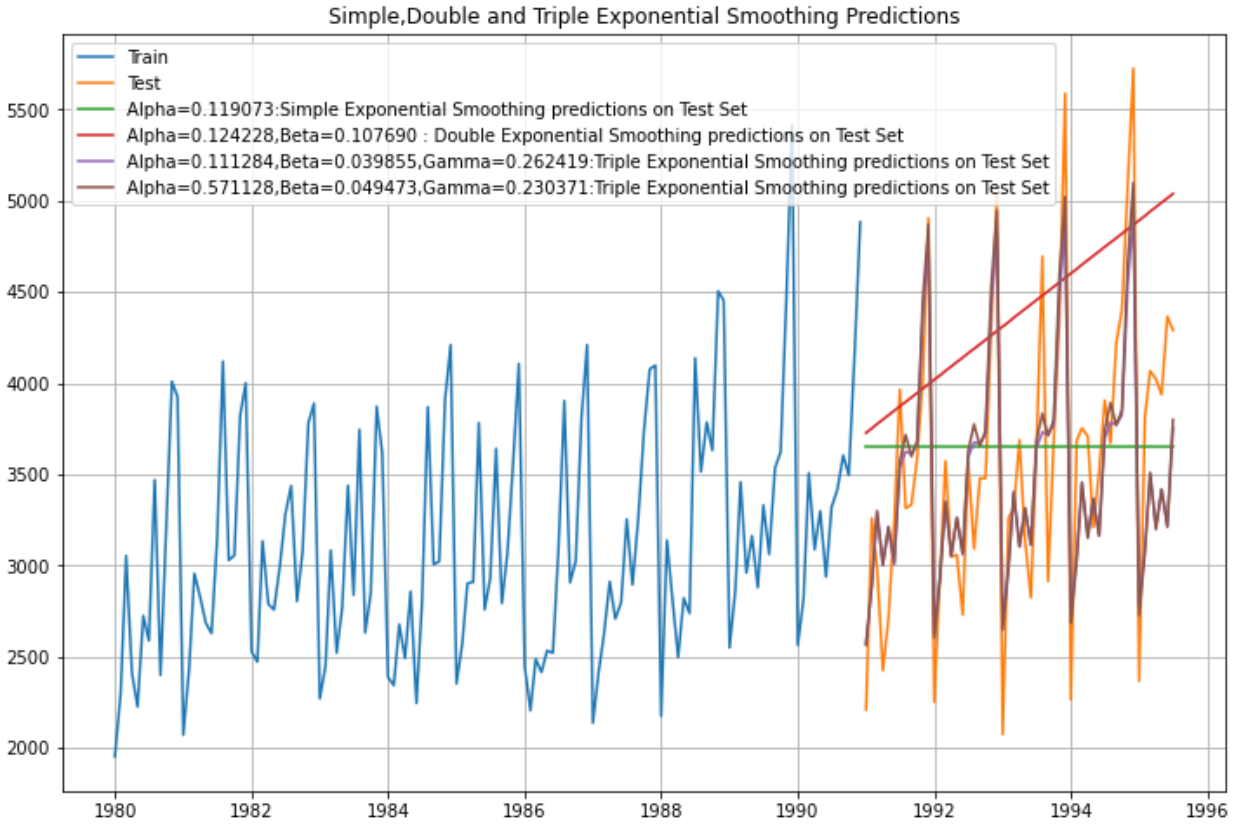
The Root Mean Square Error deviation for Triple Exponential Smoothing is 458.965.

Triple Exponential Smoothing (Level, Trend, Seasonality- Multiplicative)
The class was created on the train dataset for applying the Holt Winters model Exponential Smoothing. Fitting was done using fit() function. Here, the optimum parameters were chosen by the Python. Then, the parameters were checked using 'params' function. Here, the seasonality is multiplicative.

```
==Holt Winters model Exponential Smoothing Estimated Parameters ==

{'smoothing_level': 0.11128429736328378, 'smoothing_trend': 0.04947326762762311, 'smoothing_seasonal': 0.23037194388521623,
'damping_trend': nan, 'initial_level': 2803.0168193984414, 'initial_trend': 10.486286228443715, 'initial_seasons': array
([0.80284001, 0.86968748, 1.08266033, 0.93954787, 0.96331944,
       0.98854326, 1.0654188 , 1.28504436, 1.0083707 , 1.0929922 ,
       1.36460606, 1.41709466]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

Value of Alpha: 0.111284 Value of Beta: 0.049473 Value of Gamma: 0.230371.



Brown colored line in the above figure represents the estimated values for the test time period from 01-January-1991 to 01-July-1995. Orange line is the actual value. This model has performed better than the SES, DES and TES (additive).

```
TES_Multiplicative RMSE: 447.7225807439294
```

The Root Mean Square Error deviation for Triple Exponential Smoothing- Multiplicative is 447.7225.

	Test RMSE
Alpha=0.119073,SES	809.501640
Alpha=1,Beta=0.0189:DES	1074.329153
Alpha=0.570714,Beta=0.0001,Gamma=0.293721:TES(Additive)	458.965392
Alpha=0.571128,Beta=0.000147,Gamma=0.202947,Gamma=0:TES(Multiplicative)	447.722581

We can see that the Triple Exponential Smoothing Multiplicative model has performed the best.

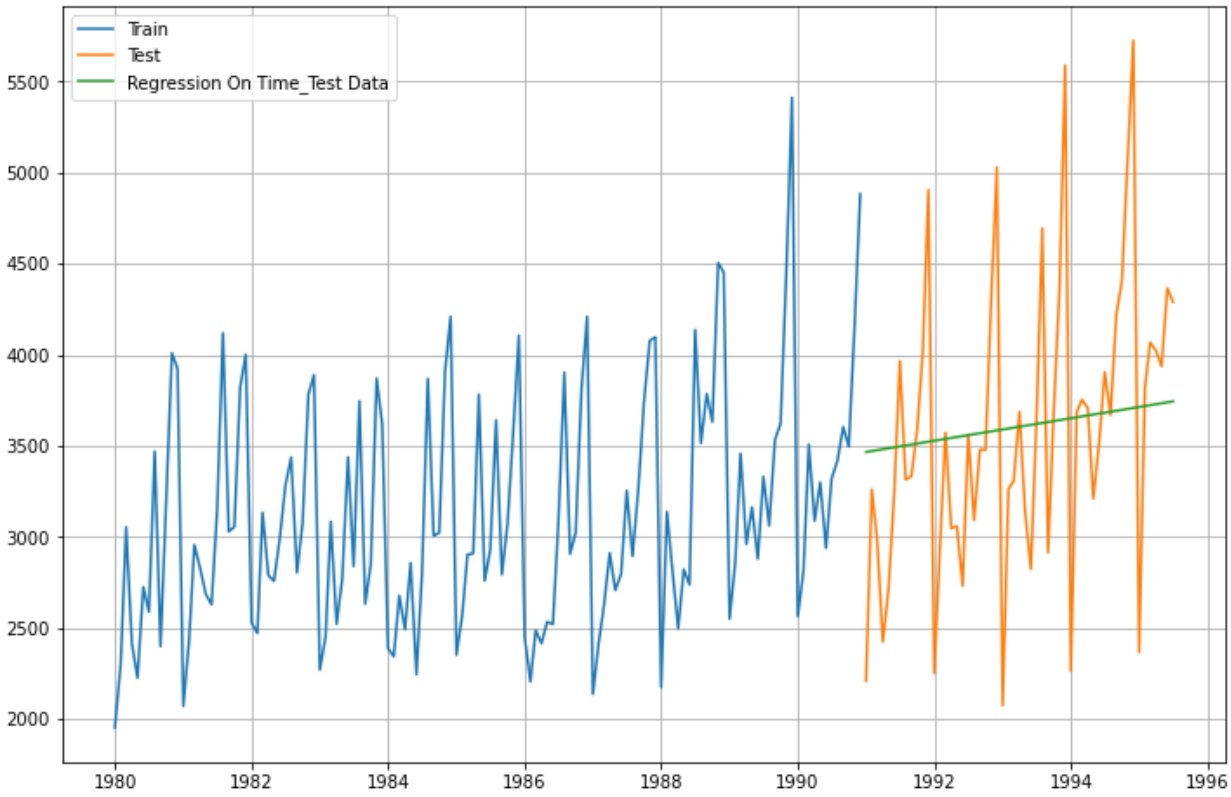
Building different models and comparing the accuracy metrics:

Linear Regression

For Linear Regression, We will take YearMonth as the independent variable and SoftDrinkProduction as dependent variable.

```
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
 [133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 1
57, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181,
182, 183, 184, 185, 186, 187]
```

Above figure represents the training time instance as well as the test time instance. We have generated the numerical time instance order for both the training and test dataset. Further, we shall add them to training and test set respectively. After the training and test data has been modified. Linear Regression was used to build the model on the training data and test the model on the test data.



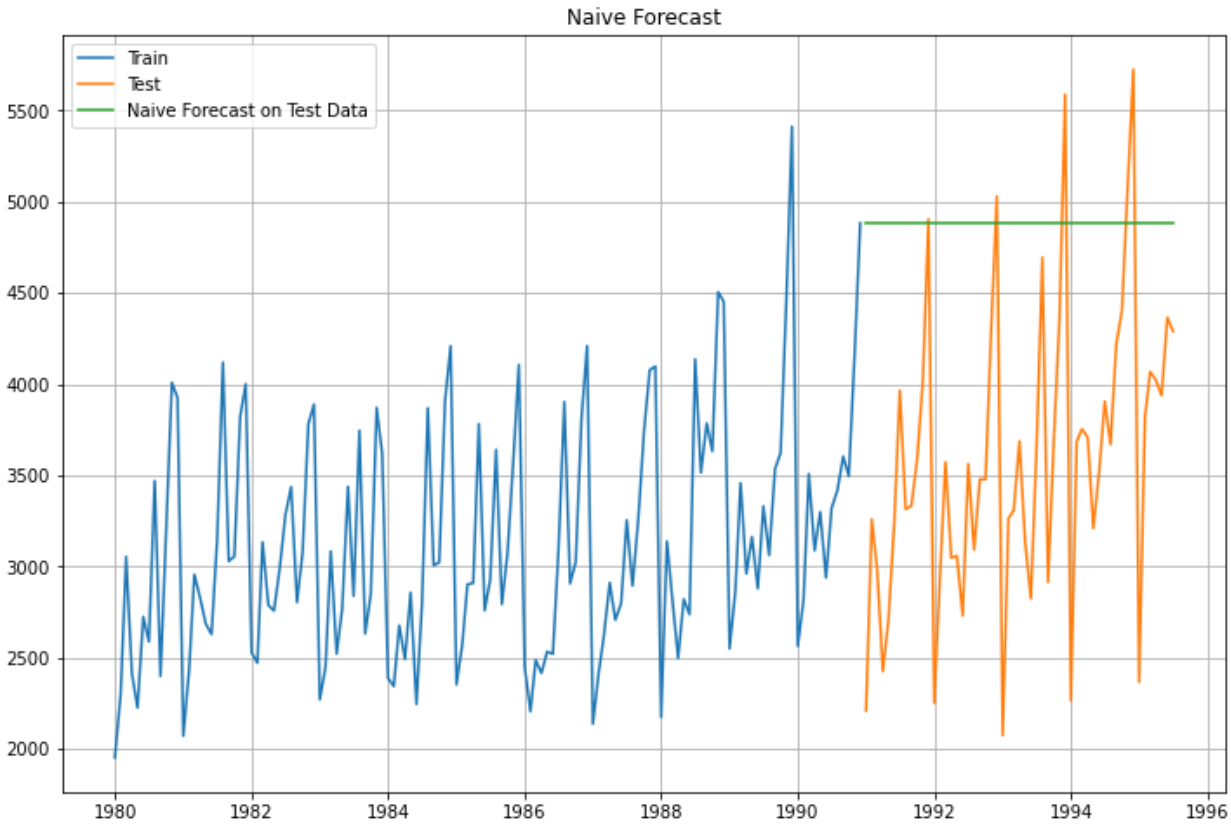
Green colored line in the above figure represents the estimated values for the test time period from 01-January-1991 to 01-July-1995. Orange line is the actual value.

```
For RegressionOnTime forecast on the Test Data,  RMSE is 775.808
```

The Root Mean Square Error deviation for Linear Regression is 775.808.

Naive Bayes

Naïve Bayes model was applied to the dataset.



Green colored line in the above figure represents the estimated values for the test time period from 01-January-1991 to 01-July-1995. Orange line is the actual value.

```
For RegressionOnTime forecast on the Test Data,  RMSE is 1519.259
```

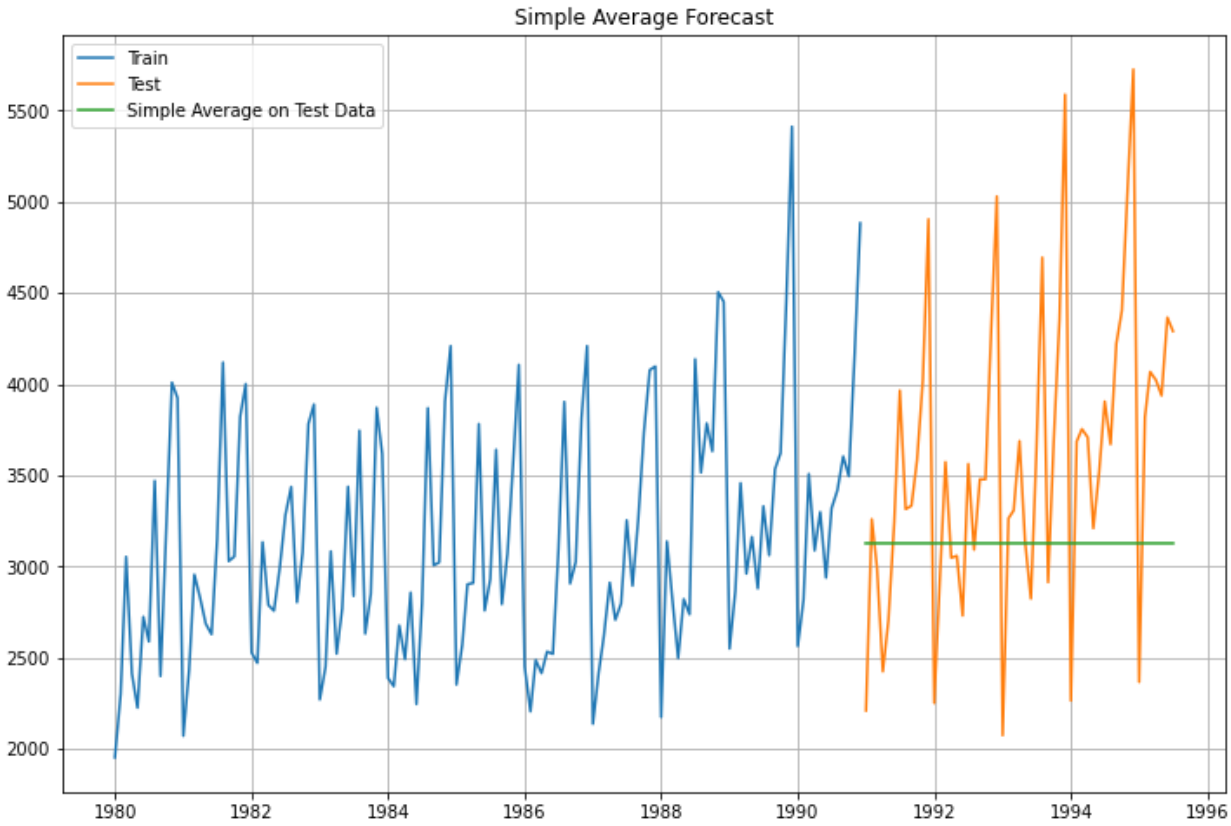
The Root Mean Square Error deviation for Naïve Bayes is 1519.259

Simple Average

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

YearMonth	SoftDrinkProduction	mean_forecast
1991-01-01	2211	3124.166667
1991-02-01	3260	3124.166667
1991-03-01	2992	3124.166667
1991-04-01	2425	3124.166667
1991-05-01	2707	3124.166667

The above figure represents the forecast values for the test period which is constant i.e. 3124.167.



Green colored line in the above figure is the estimated value which is 3124.167. Orange line is the actual value. In this model the gap between the actual and predicted isn't that much.

```
For Simple Average forecast on the Test Data,  RMSE is 934.353
```

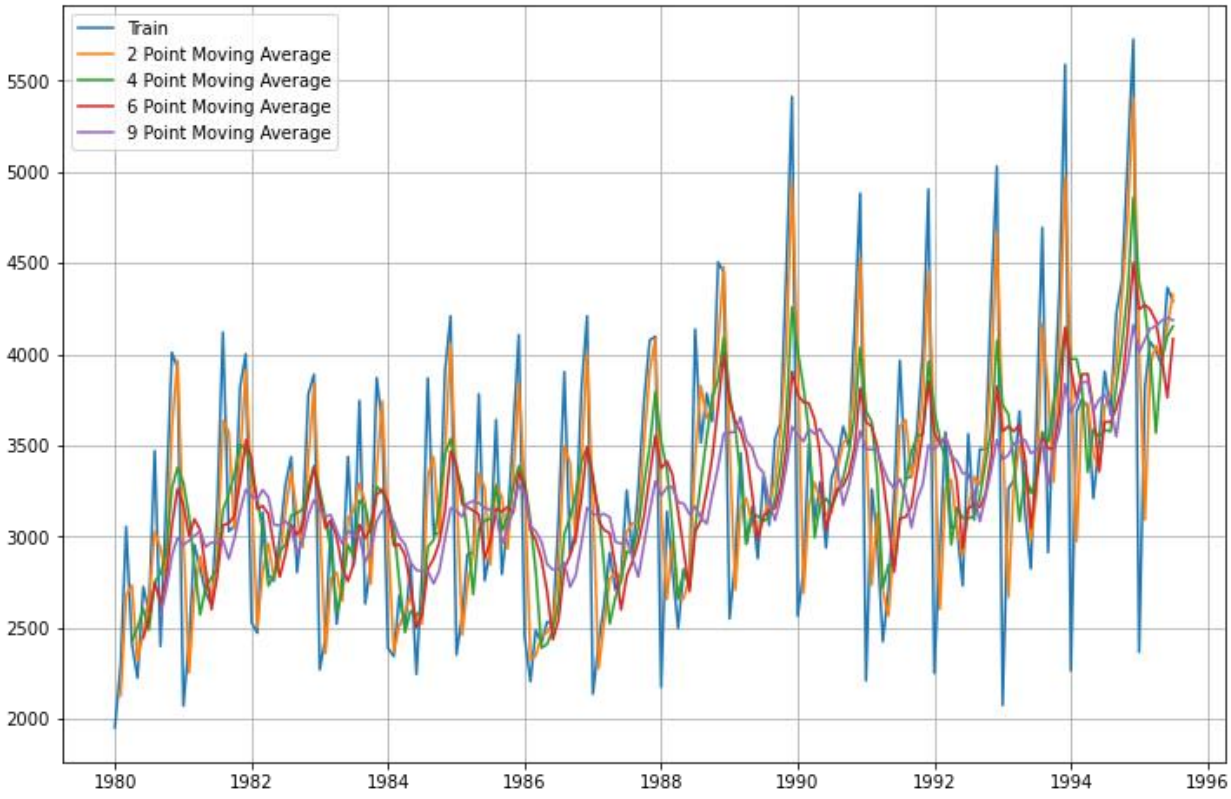
The Root Mean Square Error deviation for Simple Average Method is 934.353.

Moving Average

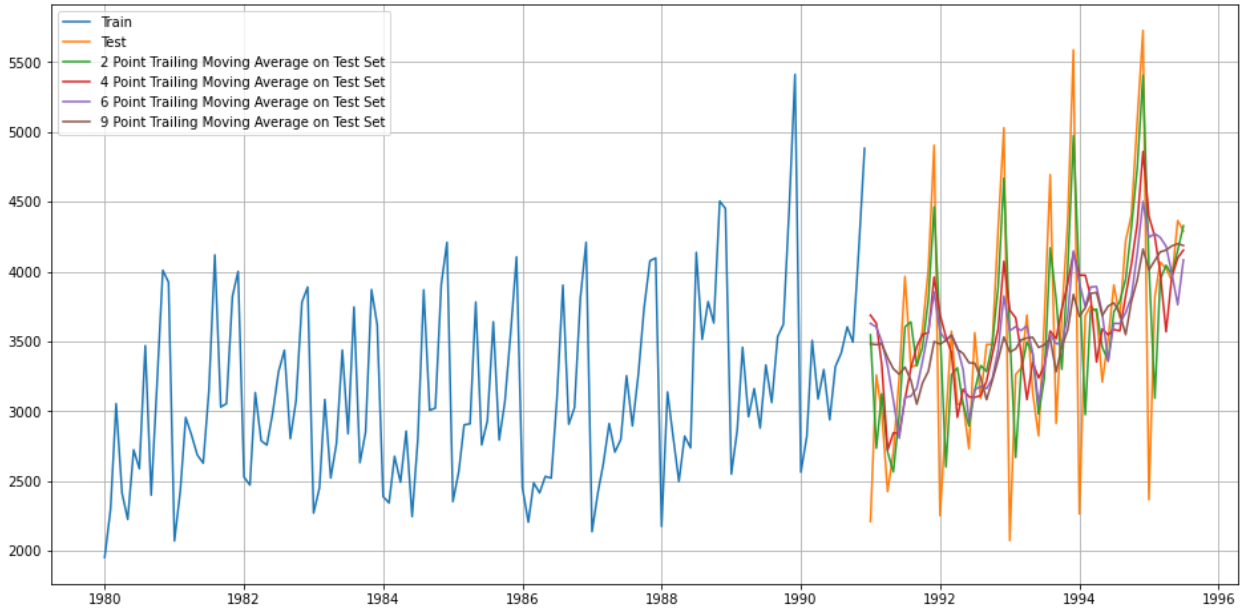
For the moving average model, we are going to calculate rolling means (or moving averages) for different intervals. The best interval can be determined by the maximum accuracy (or the minimum error) over here. For Moving Average, we are going to average over the entire data.

YearMonth	SoftDrinkProduction	Trailing_2	Trailing_4	Trailing_6	Trailing_9
1980-01-01	1954	NaN	NaN	NaN	NaN
1980-02-01	2302	2128.0	NaN	NaN	NaN
1980-03-01	3054	2678.0	NaN	NaN	NaN
1980-04-01	2414	2734.0	2431.0	NaN	NaN
1980-05-01	2226	2320.0	2499.0	NaN	NaN

Above figure shows the head() for the Trailing Moving Average for intervals 2,4,6 and 9.



Above diagram is plotted on the whole data. As the Moving Average was applied on the whole data for training therefore its performance against the test data is not visible. Therefore, we need to divide the data into the train as well as the test dataset to understand the performance of the model.

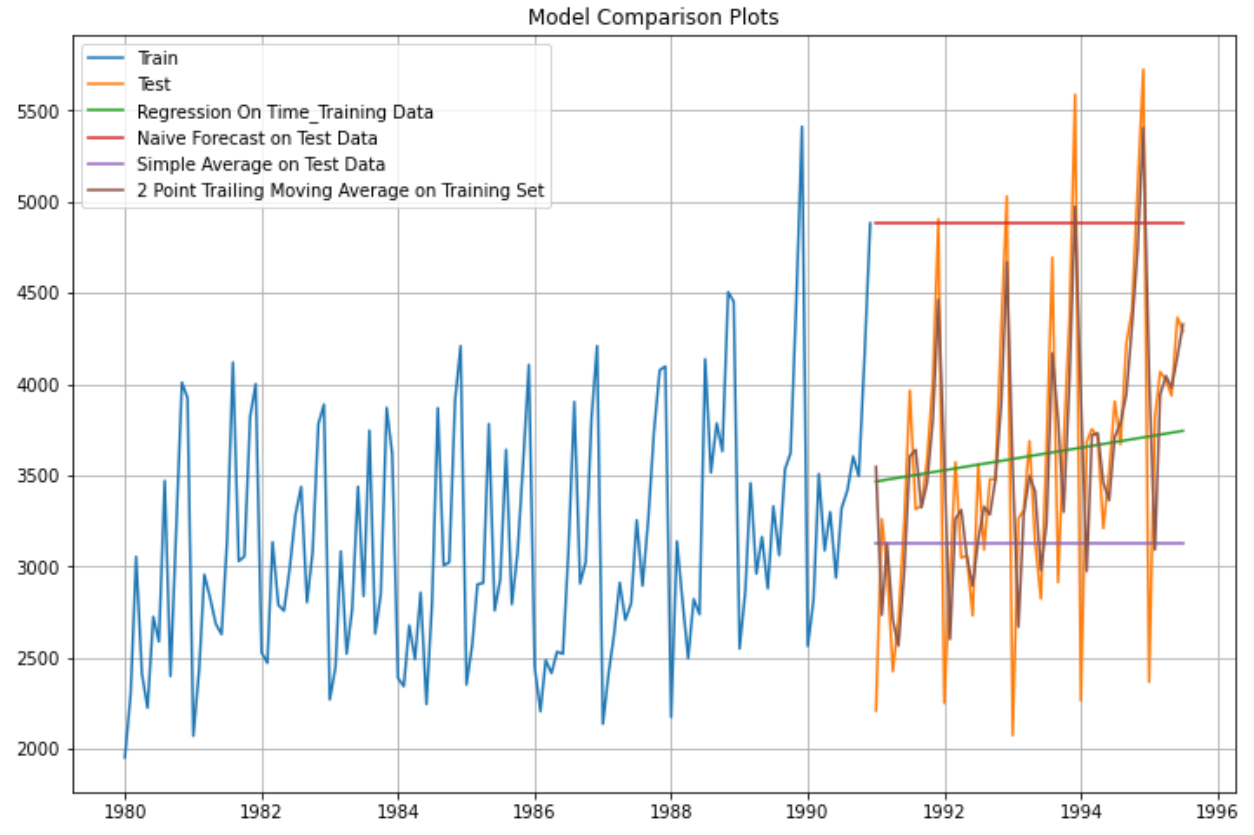


After splitting the data into train and test, the above figure shows the prediction against the test dataset. As the trailing moving average is very close to each other therefore it is hard to understand from the figure that which one among these have performed the best. Therefore, we need to rely on deviation RMSE values to understand the performance of these trailing moving average lines.

```
For 2 point Moving Average Model forecast on the Training Data,  RMSE is 556.725
For 4 point Moving Average Model forecast on the Training Data,  RMSE is 687.182
For 6 point Moving Average Model forecast on the Training Data,  RMSE is 710.514
For 9 point Moving Average Model forecast on the Training Data,  RMSE is 735.890
```

Above figure gives the RMSE values for all the trailing moving average prediction line. We can understand that the best performance is by the 2 point Moving Average Model forecast that is 556.725.

Model Comparison



Above plot represents the train dataset as well as the all the models built for prediction.

	Test RMSE
Alpha=0.119073,SES	809.501640
Alpha=1,Beta=0.0189:DES	1074.329153
Alpha=0.570714,Beta=0.0001,Gamma=0.293721:TES(Additive)	458.965392
Alpha=0.571128,Beta=0.000147,Gamma=0.202947,Gamma=0:TES(Multiplicative)	447.722581
RegressionOnTime	775.807810
NaiveModel	1519.259233
SimpleAverageModel	934.353358
2pointTrailingMovingAverage	556.725418
4pointTrailingMovingAverage	687.181726
6pointTrailingMovingAverage	710.513877
9pointTrailingMovingAverage	735.889827

Above table represents the RMSE values of all the model built on training data and tested on the test data. Here, we can see that the least Root Mean Square Error deviation for Triple Exponential Smoothing (Multiplicative) is the best with 447.722.

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.**

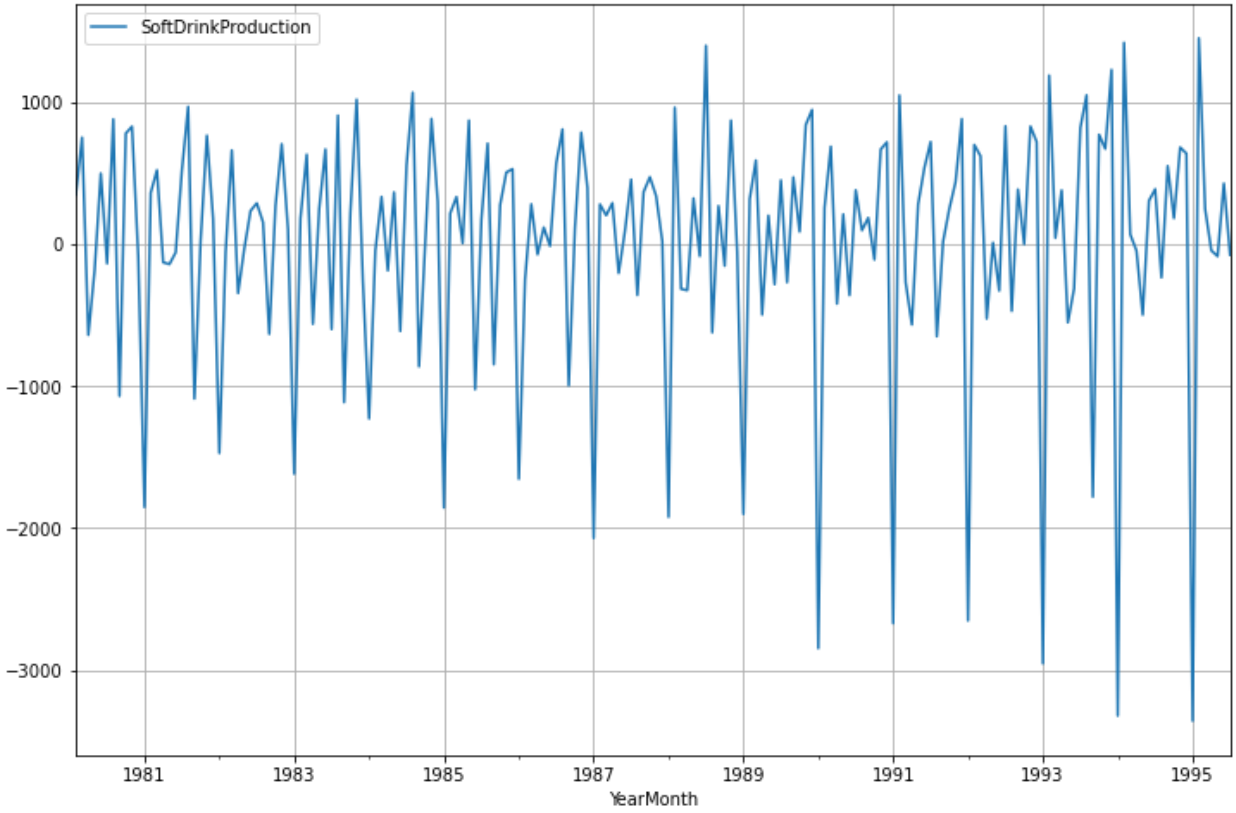
Null Hypothesis (H_0): The Time Series has a unit root and is thus non-stationary.
Alternate Hypothesis (H_1): The Time Series does not have a unit root and is thus stationary.
The python library used is the adfuller from statsmodel.

```
DataFrame test statistic is -0.425
DataFrame test p-value is 0.9861019765758295
Number of lags used 12
```

The p-value here is 0.9860 which is greater than 0.05, so we cannot reject the Null Hypothesis. Therefore, we can conclude that the series is not stationary.
To make the time series stationary, we need to take the first order difference of the time-series. Also, we need to drop the null values present in the data.

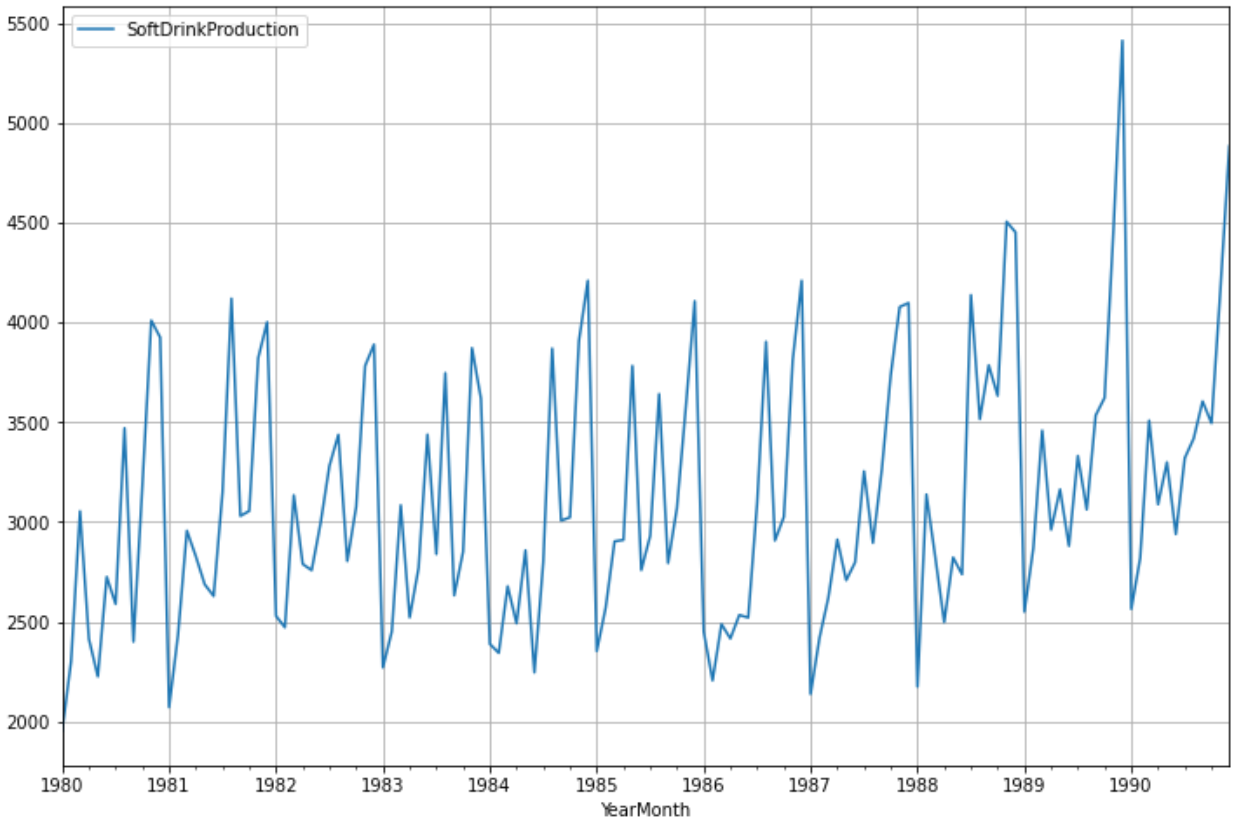
```
DF test statistic is -9.481
DF test p-value is 3.053709292535948e-14
Number of lags used 11
```

Now, if we check the p-value is less than 0.05.
Hence, we can reject the Null Hypothesis and accept the alternate Hypothesis.



Above is the plot of the series, we can see that now it is stationary. A stationary series has property of a constant mean and a constant variance which can be seen clearly in the plot above.

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.**



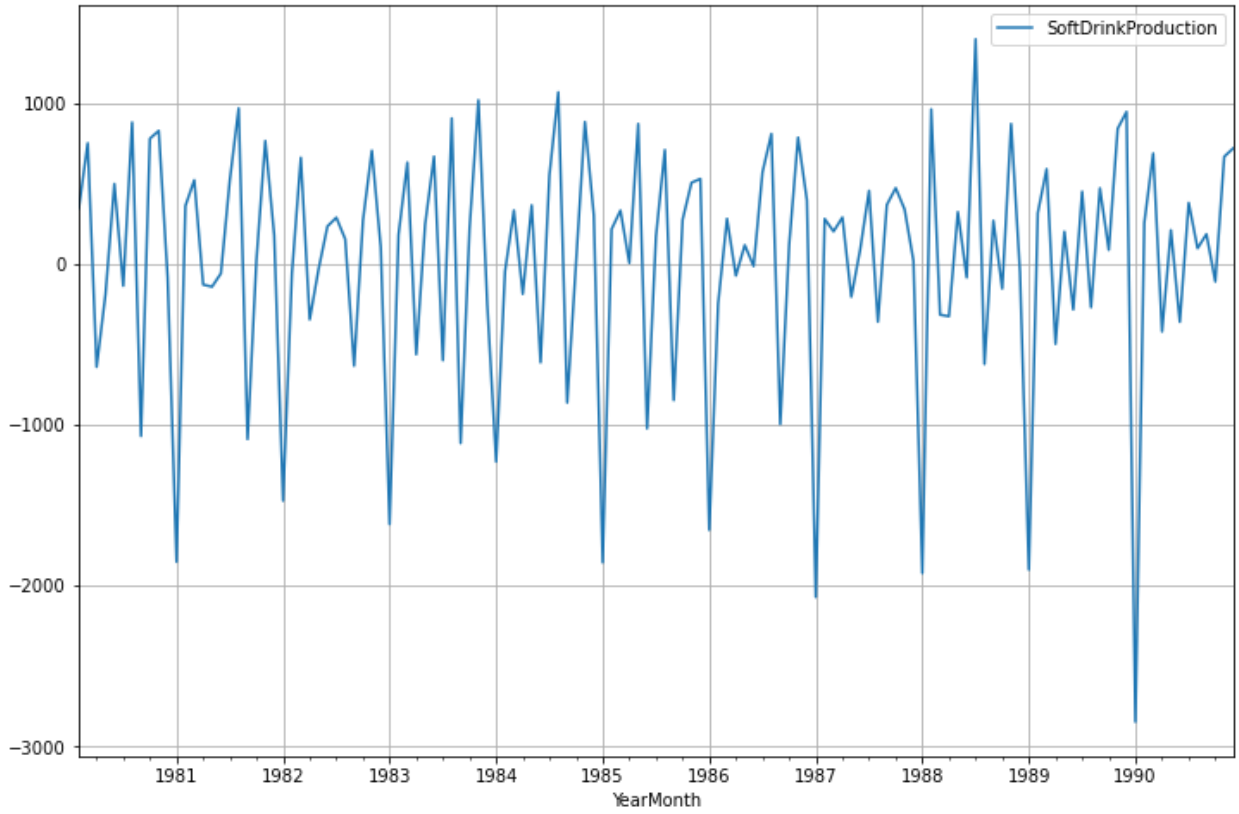
Above plot shows the training data. Here we can clearly understand that the training data is not stationary. Therefore, we have to make it stationary for applying the ARIMA/ SARIMA.

```
DataFrame test statistic is -1.649
DataFrame test p-value is 0.7726647141271693
Number of lags used 12
```

The training data is non-stationary at 95% confidence level. Let us take a first level of differencing to make the Time Series stationary.

```
DataFrame test statistic is -7.271
DataFrame test p-value is 3.4205181049970745e-09
Number of lags used 11
```

After the first level of differencing the training data has p-value lower than 0.05. Therefore, we can reject the null hypothesis and go with alternate hypothesis.



Above plot the training dataset after making it stationary. Now it is good to go for ARIMA/ SARIMA modelling.

Auto ARIMA

```
Examples of the parameter combinations for the Model
Model: (0, 1, 0)
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)
```

The above loop helps us in getting a combination of different parameters of p and q in the range of 0 and 3. We have kept the value of d as 1 as we need to take a difference of the series to make it stationary.

```
ARIMA(0, 1, 0) - AIC:2103.7338336875
ARIMA(0, 1, 1) - AIC:2069.5996302114518
ARIMA(0, 1, 2) - AIC:2056.4892632434694
ARIMA(0, 1, 3) - AIC:2056.831789419076
ARIMA(1, 1, 0) - AIC:2097.8721216490417
ARIMA(1, 1, 1) - AIC:2061.523083934958
ARIMA(1, 1, 2) - AIC:2056.7156820689206
ARIMA(1, 1, 3) - AIC:2058.7121590365855
ARIMA(2, 1, 0) - AIC:2073.234860536023
ARIMA(2, 1, 1) - AIC:2059.100671812967
ARIMA(2, 1, 2) - AIC:2058.712702099876
ARIMA(2, 1, 3) - AIC:2057.0898772431656
ARIMA(3, 1, 0) - AIC:2070.3653671411857
ARIMA(3, 1, 1) - AIC:2058.304546211261
ARIMA(3, 1, 2) - AIC:2060.6799657021306
ARIMA(3, 1, 3) - AIC:2059.6037668767667
```

We have executed a loop within the pdq parameters defined by itertools. The parameters were used from the loop. We have appended the AIC values and the model parameters to the previously created data frame. For easier understanding, we shall be sorting of the AIC values.

	param	AIC
2	(0, 1, 2)	2056.489263
6	(1, 1, 2)	2056.715682
3	(0, 1, 3)	2056.831789
11	(2, 1, 3)	2057.089877
13	(3, 1, 1)	2058.304546

Above figure has sorted AIC values. Here the best parameter is [pdq] = [0,1,2]

SARIMAX Results

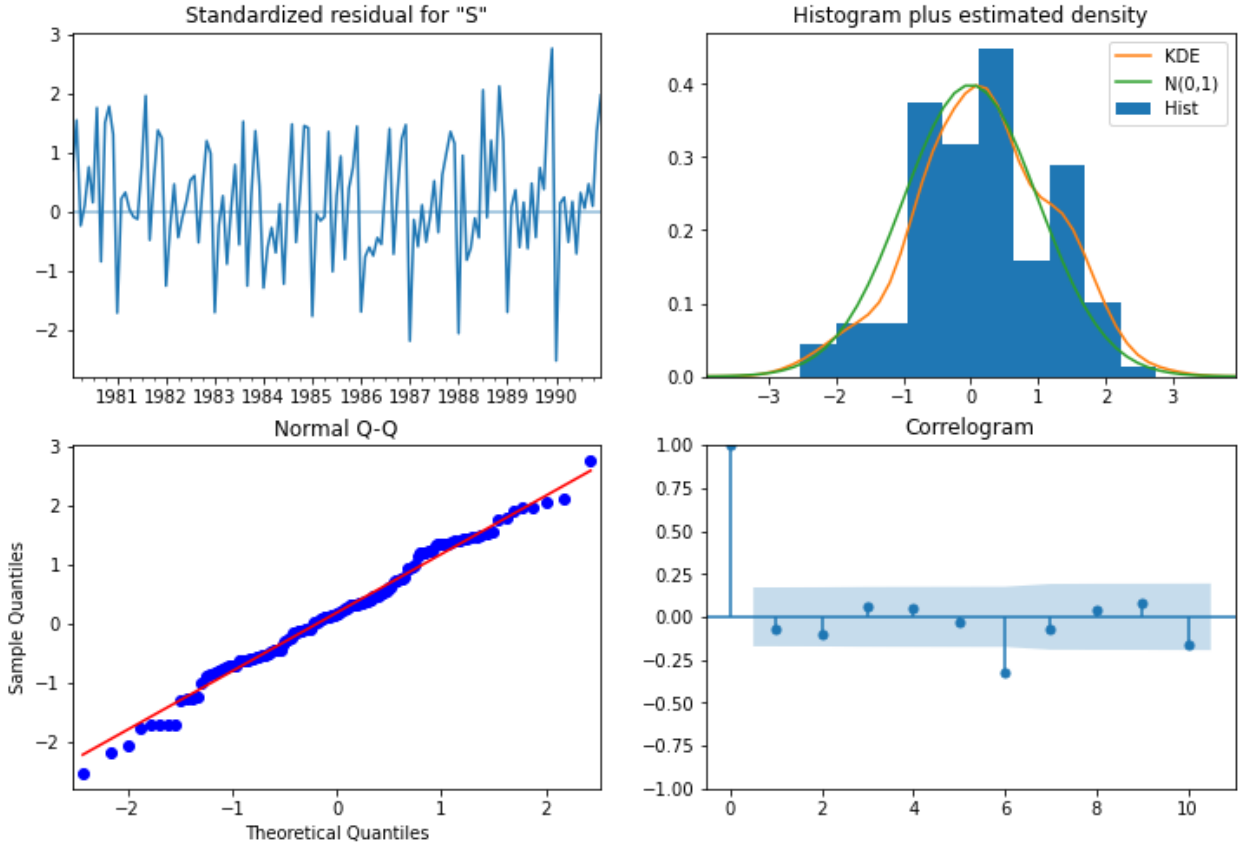
Dep. Variable:	SoftDrinkProduction	No. Observations:	132
Model:	ARIMA(0, 1, 2)	Log Likelihood	-1025.245
Date:	Sat, 08 Jan 2022	AIC	2056.489
Time:	12:39:15	BIC	2065.115
Sample:	01-01-1980 - 12-01-1990	HQIC	2059.994
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.5407	0.085	-6.392	0.000	-0.707	-0.375
ma.L2	-0.3913	0.113	-3.475	0.001	-0.612	-0.171
sigma2	3.572e+05	4.62e+04	7.725	0.000	2.67e+05	4.48e+05

Ljung-Box (L1) (Q):	0.61	Jarque-Bera (JB):	0.39
Prob(Q):	0.44	Prob(JB):	0.82
Heteroskedasticity (H):	1.31	Skew:	-0.13
Prob(H) (two-sided):	0.37	Kurtosis:	2.91

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

With the best parameters the ARIMA model was fitted. Above is the summary of it.



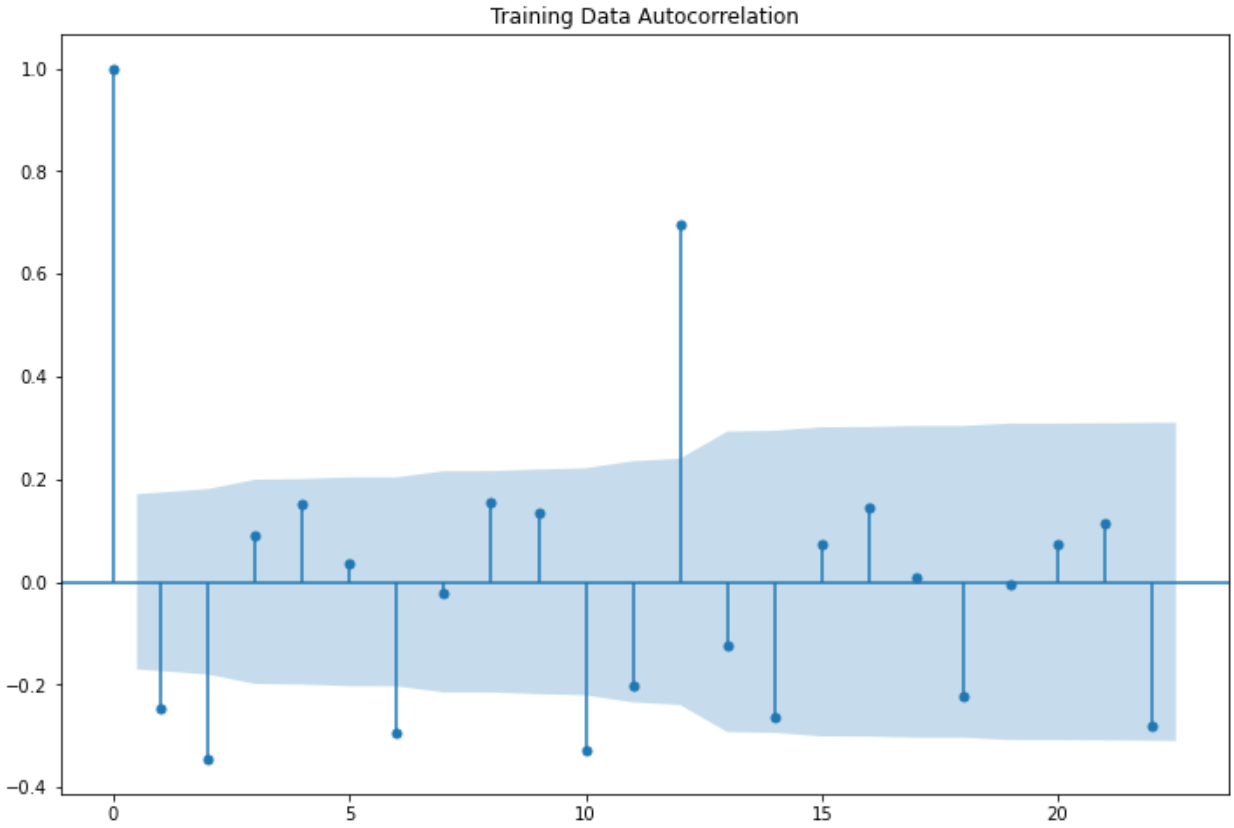
The diagnostic function was used on the train data. Above are the results for the same.

Now comes the phase where we shall check the prediction on the test data and get the accuracy of the model.

```
RMSE: 831.6158521331566
MAPE: 18.494207973397934
```

Above is the deviation RMSE & MAPE values against the test data.

Auto-SARIMA



Here we shall take the seasonality component as 6. We shall ignore the first series in the above figure because of its high correlation with original series. But from Lag1, we can clearly see that 6^{th} Lag and 12^{th} Lag is significant in the above diagram. Therefore, seasonality is 6.

```
Examples of the parameter combinations for the Model are
Model: (0, 1, 1)(0, 0, 1, 6)
Model: (0, 1, 2)(0, 0, 2, 6)
Model: (0, 1, 3)(0, 0, 3, 6)
Model: (1, 1, 0)(1, 0, 0, 6)
Model: (1, 1, 1)(1, 0, 1, 6)
Model: (1, 1, 2)(1, 0, 2, 6)
Model: (1, 1, 3)(1, 0, 3, 6)
Model: (2, 1, 0)(2, 0, 0, 6)
Model: (2, 1, 1)(2, 0, 1, 6)
Model: (2, 1, 2)(2, 0, 2, 6)
Model: (2, 1, 3)(2, 0, 3, 6)
Model: (3, 1, 0)(3, 0, 0, 6)
Model: (3, 1, 1)(3, 0, 1, 6)
Model: (3, 1, 2)(3, 0, 2, 6)
Model: (3, 1, 3)(3, 0, 3, 6)
```

Above are the examples of the parameter combinations for the model. We shall choose the one with lowest AIC value.

```
SARIMA(0, 1, 0)x(0, 0, 0, 6) - AIC:2088.4633983660715
SARIMA(0, 1, 0)x(0, 0, 1, 6) - AIC:1991.6865911398422
SARIMA(0, 1, 0)x(0, 0, 2, 6) - AIC:1835.7619130468058
SARIMA(0, 1, 0)x(0, 0, 3, 6) - AIC:1746.6922157437966
SARIMA(0, 1, 0)x(1, 0, 0, 6) - AIC:2001.2435317896568
SARIMA(0, 1, 0)x(1, 0, 1, 6) - AIC:1938.055937089127
SARIMA(0, 1, 0)x(1, 0, 2, 6) - AIC:1819.7378453343078
SARIMA(0, 1, 0)x(1, 0, 3, 6) - AIC:1714.4573257054753
SARIMA(0, 1, 0)x(2, 0, 0, 6) - AIC:1790.6600051597607
SARIMA(0, 1, 0)x(2, 0, 1, 6) - AIC:1790.4200189619835
SARIMA(0, 1, 0)x(2, 0, 2, 6) - AIC:1765.5917652590535
SARIMA(0, 1, 0)x(2, 0, 3, 6) - AIC:1681.213865056899
SARIMA(0, 1, 0)x(3, 0, 0, 6) - AIC:1704.3326684928627
SARIMA(0, 1, 0)x(3, 0, 1, 6) - AIC:1702.2015504499402
```

.
.
.
.

```
SARIMA(3, 1, 3)x(0, 0, 2, 6) - AIC:1752.931895454675
SARIMA(3, 1, 3)x(0, 0, 3, 6) - AIC:1665.7090027468123
SARIMA(3, 1, 3)x(1, 0, 0, 6) - AIC:1867.6433030524186
SARIMA(3, 1, 3)x(1, 0, 1, 6) - AIC:1823.9344391808315
SARIMA(3, 1, 3)x(1, 0, 2, 6) - AIC:1702.4858544721117
SARIMA(3, 1, 3)x(1, 0, 3, 6) - AIC:1625.7914773153766
SARIMA(3, 1, 3)x(2, 0, 0, 6) - AIC:1706.447090947792
SARIMA(3, 1, 3)x(2, 0, 1, 6) - AIC:1710.1353616789306
SARIMA(3, 1, 3)x(2, 0, 2, 6) - AIC:1675.8110678712685
SARIMA(3, 1, 3)x(2, 0, 3, 6) - AIC:1593.616772683024
SARIMA(3, 1, 3)x(3, 0, 0, 6) - AIC:1624.232637276705
SARIMA(3, 1, 3)x(3, 0, 1, 6) - AIC:1625.6820905891523
SARIMA(3, 1, 3)x(3, 0, 2, 6) - AIC:1607.325798186081
SARIMA(3, 1, 3)x(3, 0, 3, 6) - AIC:1589.7609615873516
```

Automated SARIMA was applied to the training data.

	param	seasonal	AIC
255	(3, 1, 3)	(3, 0, 3, 6)	1589.760962
59	(0, 1, 3)	(2, 0, 3, 6)	1590.244407
191	(2, 1, 3)	(3, 0, 3, 6)	1591.011959
123	(1, 1, 3)	(2, 0, 3, 6)	1591.284675
63	(0, 1, 3)	(3, 0, 3, 6)	1592.243831

Above are the parameters sorted from low to high according to the AIC.
Therefore, p=3 d=1 q=3 P=3 D=0 Q=3 at seasonality = 6 are the best parameters according to the model.

SARIMAX Results

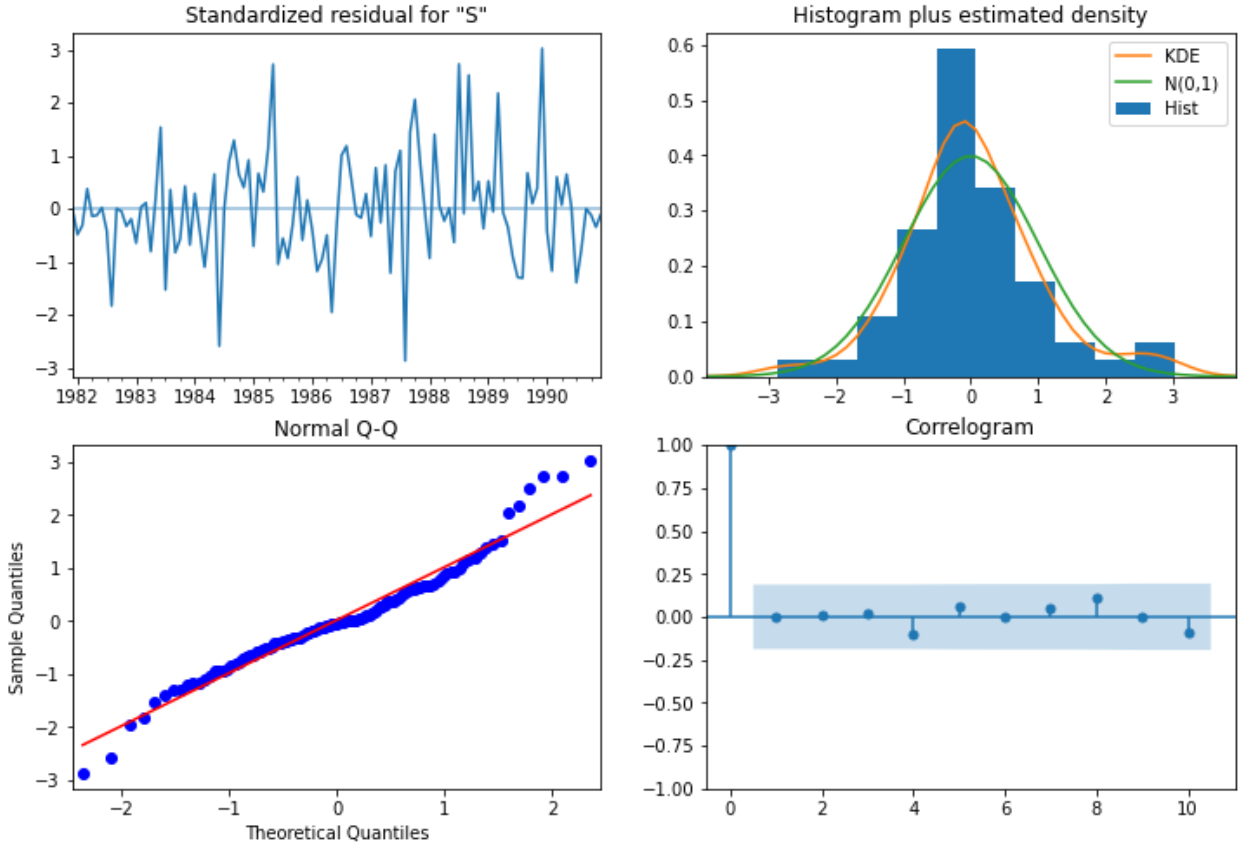
Dep. Variable:	SoftDrinkProduction	No. Observations:	132
Model:	SARIMAX(3, 1, 3)x(3, 0, 3, 6)	Log Likelihood	-781.880
Date:	Sun, 09 Jan 2022	AIC	1589.761
Time:	12:06:36	BIC	1624.748
Sample:	01-01-1980 - 12-01-1990	HQIC	1603.950
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.8582	0.139	6.167	0.000	0.585	1.131
ar.L2	-1.0002	0.104	-9.654	0.000	-1.203	-0.797
ar.L3	0.2193	0.130	1.684	0.092	-0.036	0.475
ma.L1	-1.7147	0.121	-14.228	0.000	-1.951	-1.479
ma.L2	1.7309	0.205	8.445	0.000	1.329	2.133
ma.L3	-0.9211	0.158	-5.813	0.000	-1.232	-0.611
ar.S.L6	-0.1182	0.577	-0.205	0.838	-1.249	1.013
ar.S.L12	1.0021	0.035	28.753	0.000	0.934	1.070
ar.S.L18	0.1367	0.586	0.233	0.816	-1.012	1.285
ma.S.L6	0.0531	0.580	0.091	0.927	-1.084	1.190
ma.S.L12	-0.6039	0.138	-4.390	0.000	-0.874	-0.334
ma.S.L18	-0.1249	0.389	-0.321	0.748	-0.888	0.638
sigma2	8.982e+04	5.15e-06	1.75e+10	0.000	8.98e+04	8.98e+04

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	10.59
Prob(Q):	0.98	Prob(JB):	0.01
Heteroskedasticity (H):	1.77	Skew:	0.38
Prob(H) (two-sided):	0.09	Kurtosis:	4.33

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 1.64e+27. Standard errors may be unstable.

After applying the best parameters to the model, we get the above model.



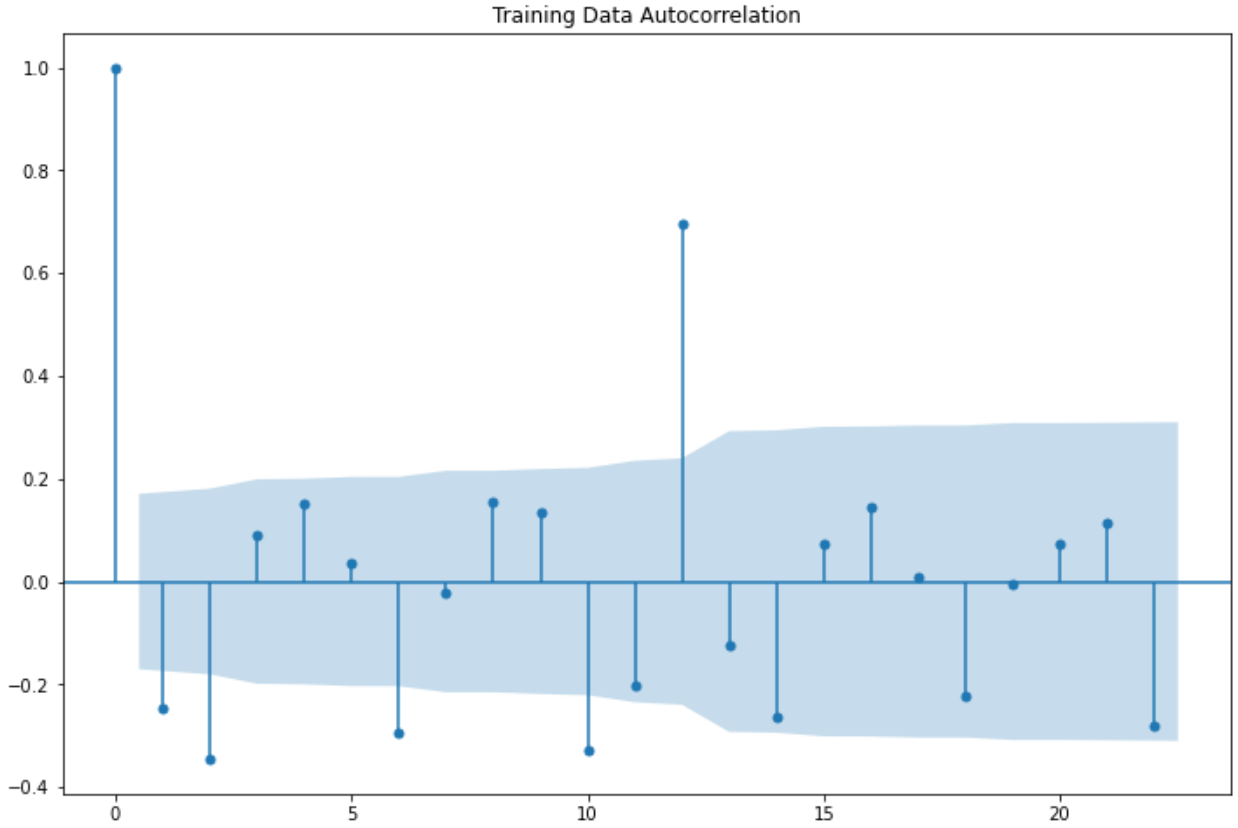
The diagnostic function was applied to the output.

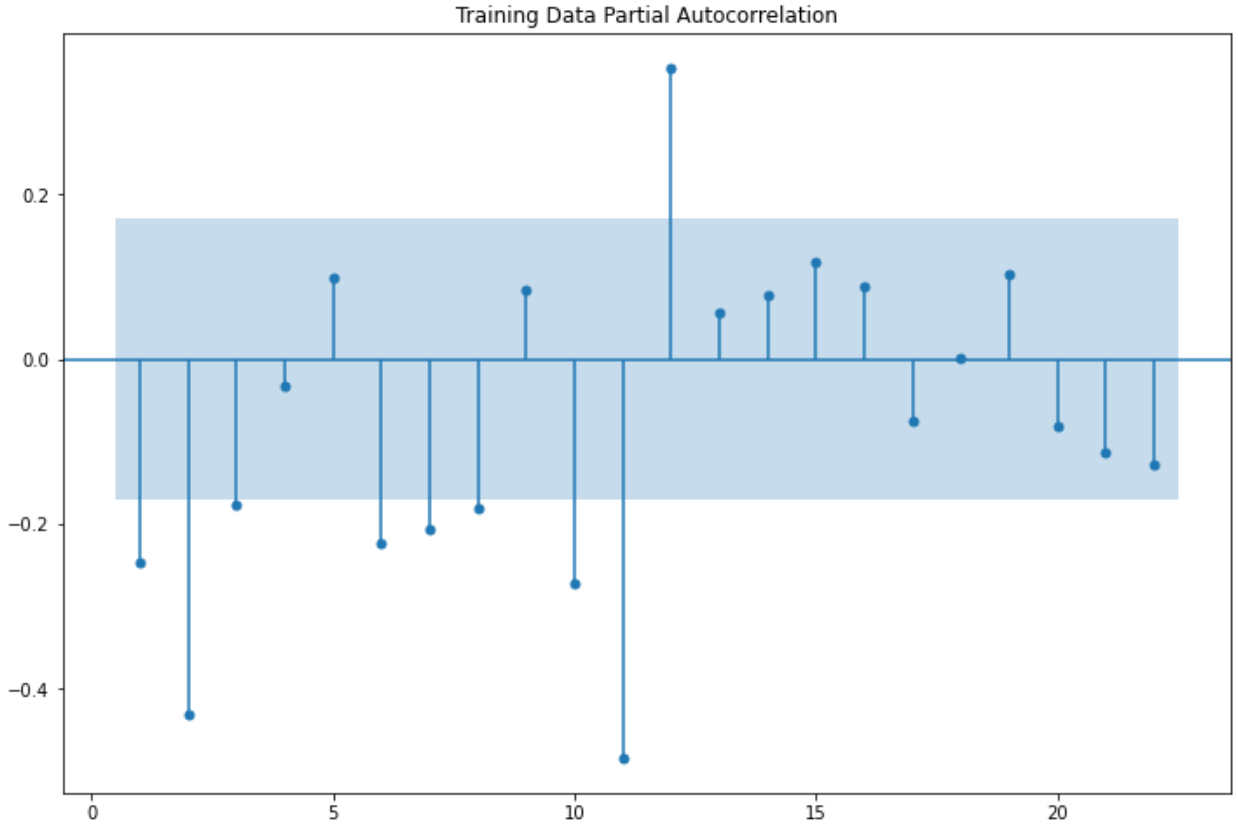
SoftDrinkProduction	mean	mean_se	mean_ci_lower	mean_ci_upper
1991-01-01	2810.754060	302.172609	2218.506630	3403.001490
1991-02-01	3144.759464	305.528186	2545.935224	3743.583704
1991-03-01	3279.034107	307.965981	2675.431875	3882.636338
1991-04-01	2957.418178	318.955464	2332.276957	3582.559399
1991-05-01	3330.305230	326.691973	2690.000730	3970.609730

```
RMSE: 428.68312547435875
MAPE: 10.86586083696877
```

Above is the deviation RMSE & MAPE values against the test data.

7. **Build ARIMA/SARIMA models based on the cut-off points of ACF and PACF on the training data and evaluate this model on the test data using RMSE.**
   ARIMA by looking at ACF & PACF


Training Data Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20


Training Data Partial Autocorrelation
0.2
0.0
−0.2
−0.4
0
5
10
15
20

Here, we have taken alpha=0.05.

- The Auto-Regressive parameter in an ARIMA model is 'p' which comes from the significant lag before which the PACF plot cuts-off to 3.
- The Moving-Average parameter in an ARIMA model is 'q' which comes from the significant lag before the ACF plot cuts-off to 2.
- The difference value remains the same as 1

SARIMAX Results

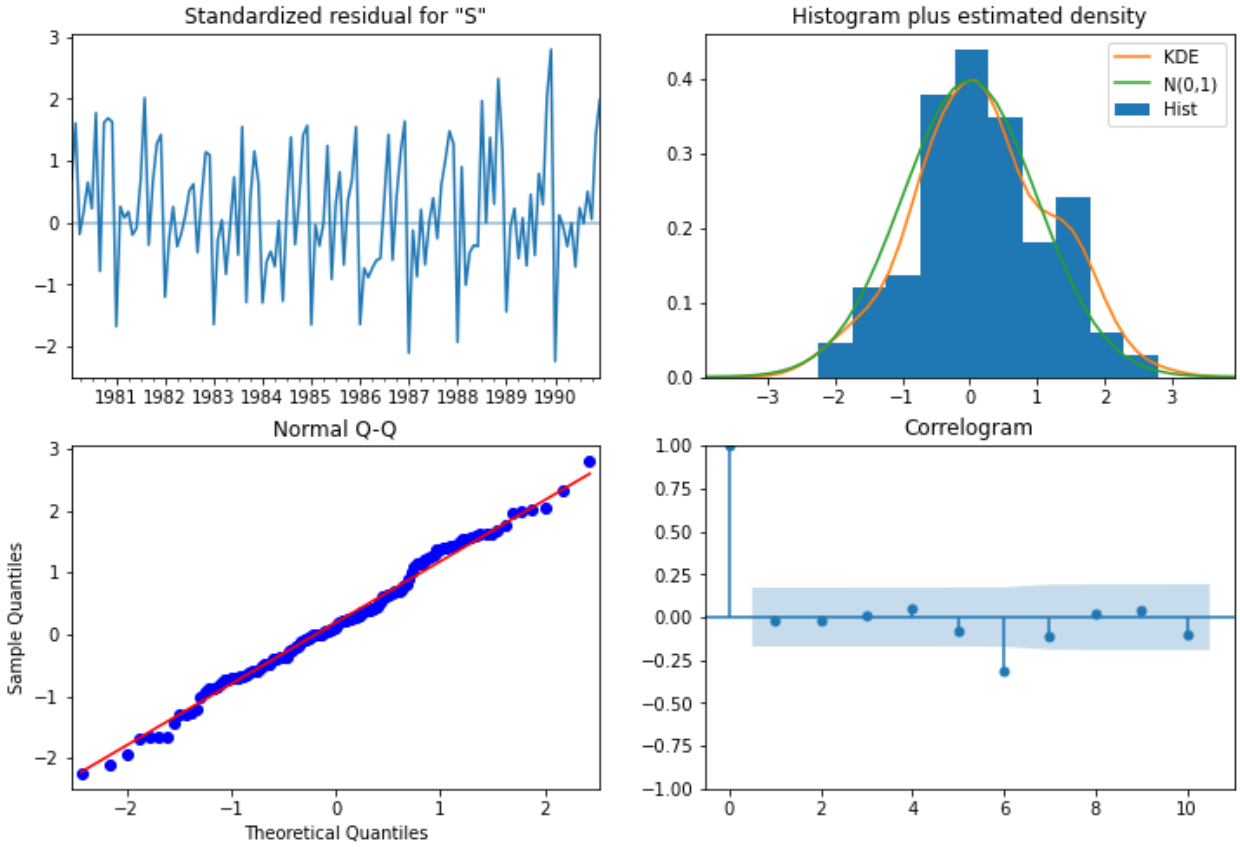
Dep. Variable:	SoftDrinkProduction	No. Observations:	132
Model:	ARIMA(3, 1, 2)	Log Likelihood	-1024.340
Date:	Sun, 09 Jan 2022	AIC	2060.680
Time:	11:43:31	BIC	2077.931
Sample:	01-01-1980	HQIC	2067.690
	- 12-01-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.3216	0.460	-0.700	0.484	-1.223	0.579
ar.L2	-0.0001	0.189	-0.001	0.999	-0.371	0.370
ar.L3	-0.0227	0.218	-0.104	0.917	-0.449	0.404
ma.L1	-0.2633	0.453	-0.581	0.561	-1.151	0.624
ma.L2	-0.6405	0.400	-1.599	0.110	-1.425	0.144
sigma2	3.514e+05	4.97e+04	7.066	0.000	2.54e+05	4.49e+05

Ljung-Box (L1) (Q):	0.07	Jarque-Bera (JB):	0.43
Prob(Q):	0.80	Prob(JB):	0.81
Heteroskedasticity (H):	1.29	Skew:	0.04
Prob(H) (two-sided):	0.40	Kurtosis:	2.73

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

Based on the parameters identified (3,1,2), the fitting on the model was built. Above is the summary of the same.

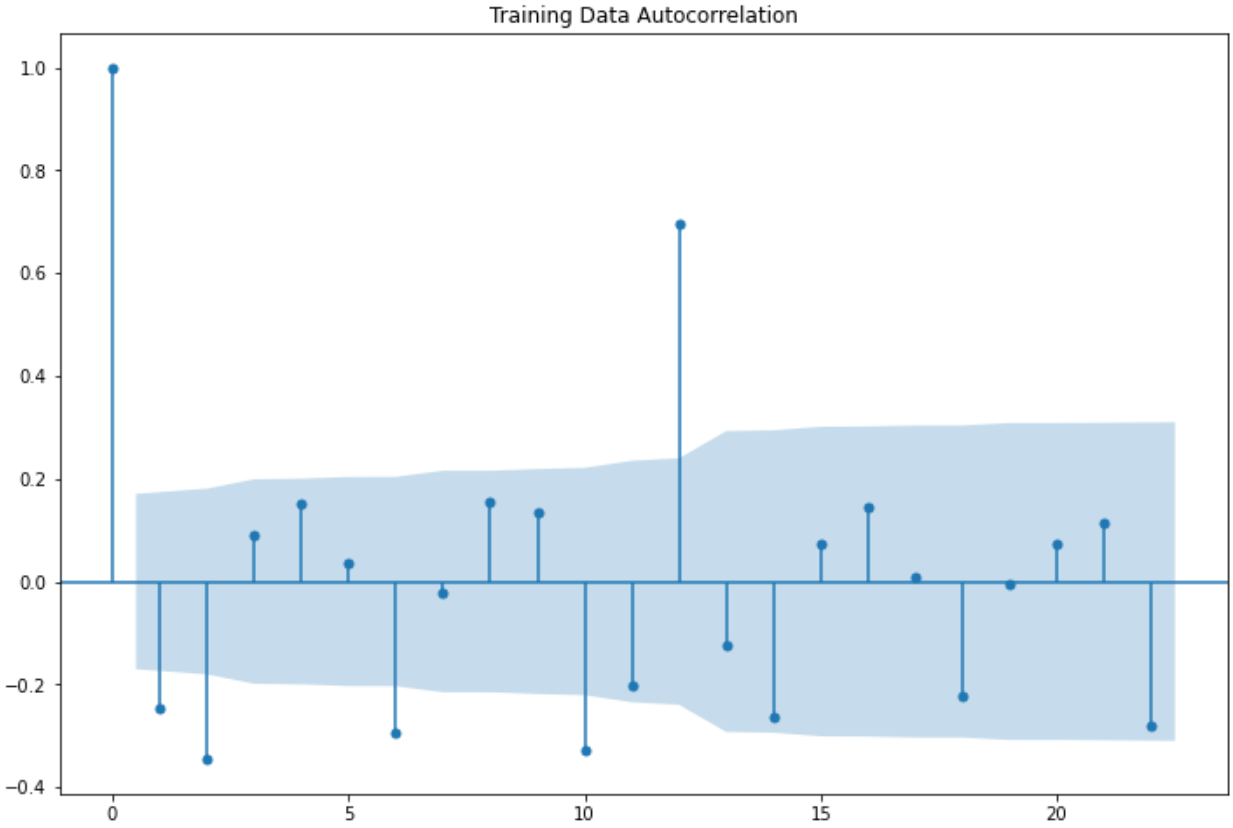


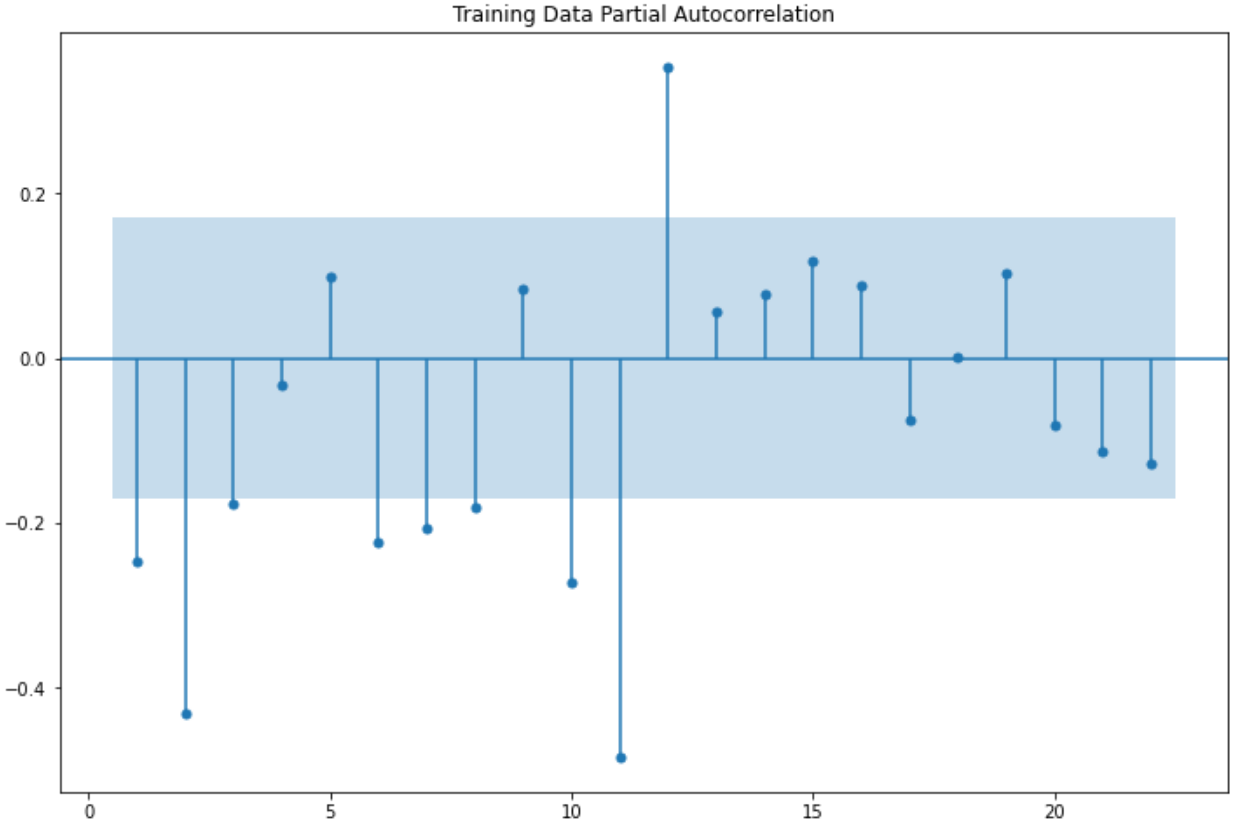
The diagnostic function was used on the train data. Above are the results for the same. Now comes the phase where we shall check the prediction on the test data and get the accuracy of the model.

```
RMSE: 822.217444645557
MAPE: 18.363068954479175
```

Above is the deviation RMSE & MAPE values against the test data.

SARIMA by looking at ACF & PACF


Training Data Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20


Training Data Partial Autocorrelation
0.2
0.0
−0.2
−0.4
0
5
10
15
20

Here, we have taken alpha=0.05.

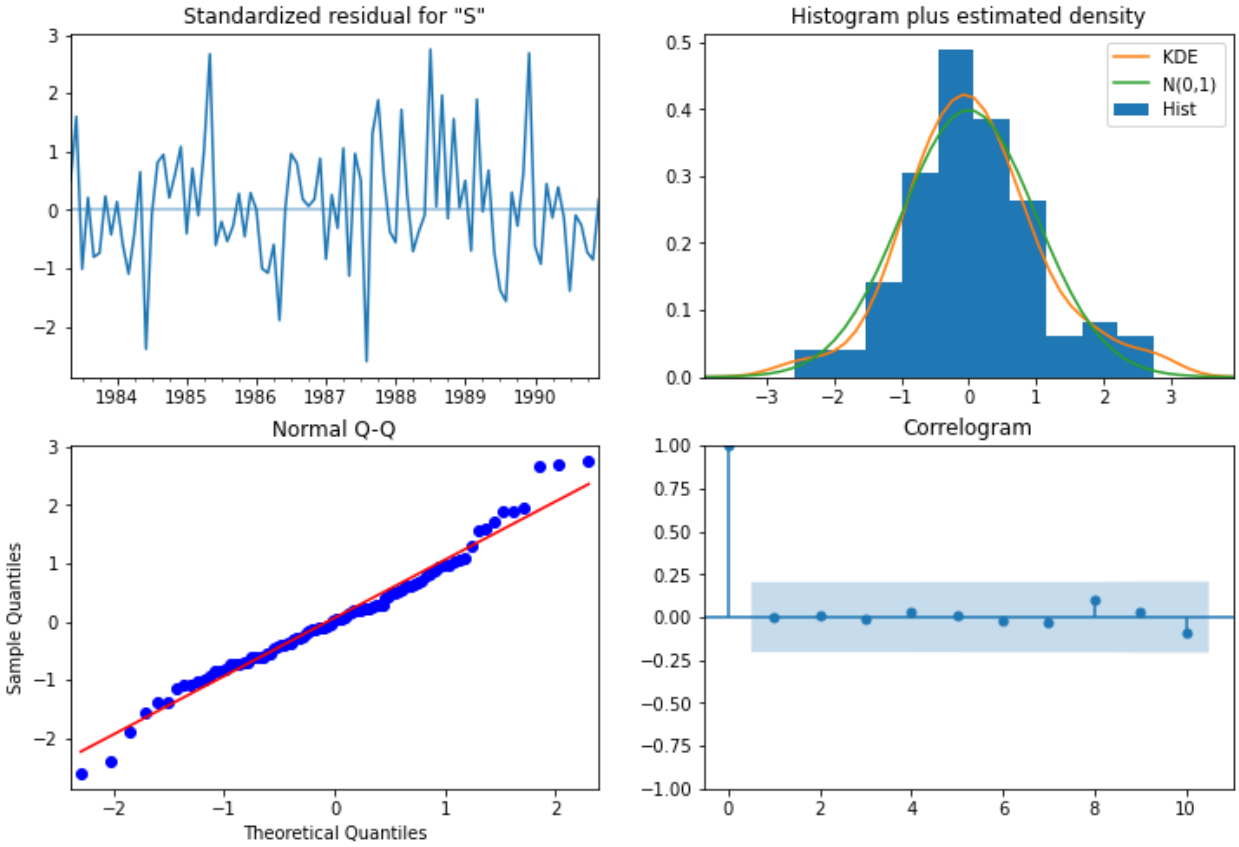
We are going to take the seasonal period as 6. We are taking the p value to be 3 and the q value also to be 2 as the parameters same as the ARIMA model.

- The Auto-Regressive parameter in an SARIMA model is 'P' which comes from the significant lag after which the PACF plot cuts-off to 6.
- The Moving-Average parameter in an SARIMA model is 'Q' which comes from the significant lag after which the ACF plot cuts-off to 0.
- Here the value of D=0

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                SoftDrinkProduction   No. Observations:                  132
Model:             SARIMAX(3, 1, 2)x(6, 0, [], 6)   Log Likelihood                -667.816
Date:                            Sun, 09 Jan 2022   AIC                           1359.632
Time:                                    13:55:40   BIC                           1389.893
Sample:                                01-01-1980   HQIC                          1371.845
                                     - 12-01-1990
Covariance Type:                               opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.2520      1.160     -0.217      0.828      -2.526       2.022
ar.L2          0.0041      0.179      0.023      0.982      -0.347       0.356
ar.L3          0.0856      0.188      0.454      0.650      -0.284       0.455
ma.L1         -0.5488      1.190     -0.461      0.645      -2.882       1.784
ma.L2         -0.3275      1.044     -0.314      0.754      -2.374       1.719
ar.S.L6       -0.1520      0.144     -1.058      0.290      -0.433       0.129
ar.S.L12       0.4010      0.118      3.395      0.001       0.170       0.632
ar.S.L18       0.0024      0.136      0.018      0.986      -0.264       0.269
ar.S.L24       0.3115      0.134      2.319      0.020       0.048       0.575
ar.S.L30       0.1474      0.173      0.853      0.394      -0.191       0.486
ar.S.L36       0.2343      0.115      2.031      0.042       0.008       0.460
sigma2      1.17e+05   1.88e+04      6.216      0.000    8.01e+04    1.54e+05
==========================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                 3.47
Prob(Q):                              1.00   Prob(JB):                         0.18
Heteroskedasticity (H):               1.38   Skew:                             0.31
Prob(H) (two-sided):                  0.37   Kurtosis:                         3.72
==========================================================================================

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

The model was applied to the parameters.



The diagnostic function was used on the train data. Above are the results for the same.

```
RMSE: 452.7660044226601
MAPE: 11.652814042195676
```

Above is the deviation RMSE & MAPE values against the test data.

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

	Test RMSE
Alpha=0.119073,SES	809.501640
Alpha=1,Beta=0.0189:DES	1074.329153
Alpha=0.570714,Beta=0.0001,Gamma=0.293721:TES(Additive)	458.965392
Alpha=0.571128,Beta=0.000147,Gamma=0.202947,Gamma=0:TES(Multiplicative)	447.722581
RegressionOnTime	775.807810
NaiveModel	1519.259233
SimpleAverageModel	934.353358
2pointTrailingMovingAverage	556.725418
4pointTrailingMovingAverage	687.181726
6pointTrailingMovingAverage	710.513877
9pointTrailingMovingAverage	735.889827
ARIMA(0,1,2) Lowest AIC	831.615852
ARIMA(3,1,2)By looking ACF and PACF	822.217445
SARIMA(3,1,3)(3,0,3,6)Lowest AIC	428.683125
SARIMA(3,1,2)(6,0,0,6)By looking ACF and PACF	452.766004

The best performing models are SARIMA – by lowest **Akaike Information Criterion (AIC)** with RMSE value 428.6831 which is followed by **Triple Exponential Smoothing (Multiplicative)** with RMSE value 447.7225.

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

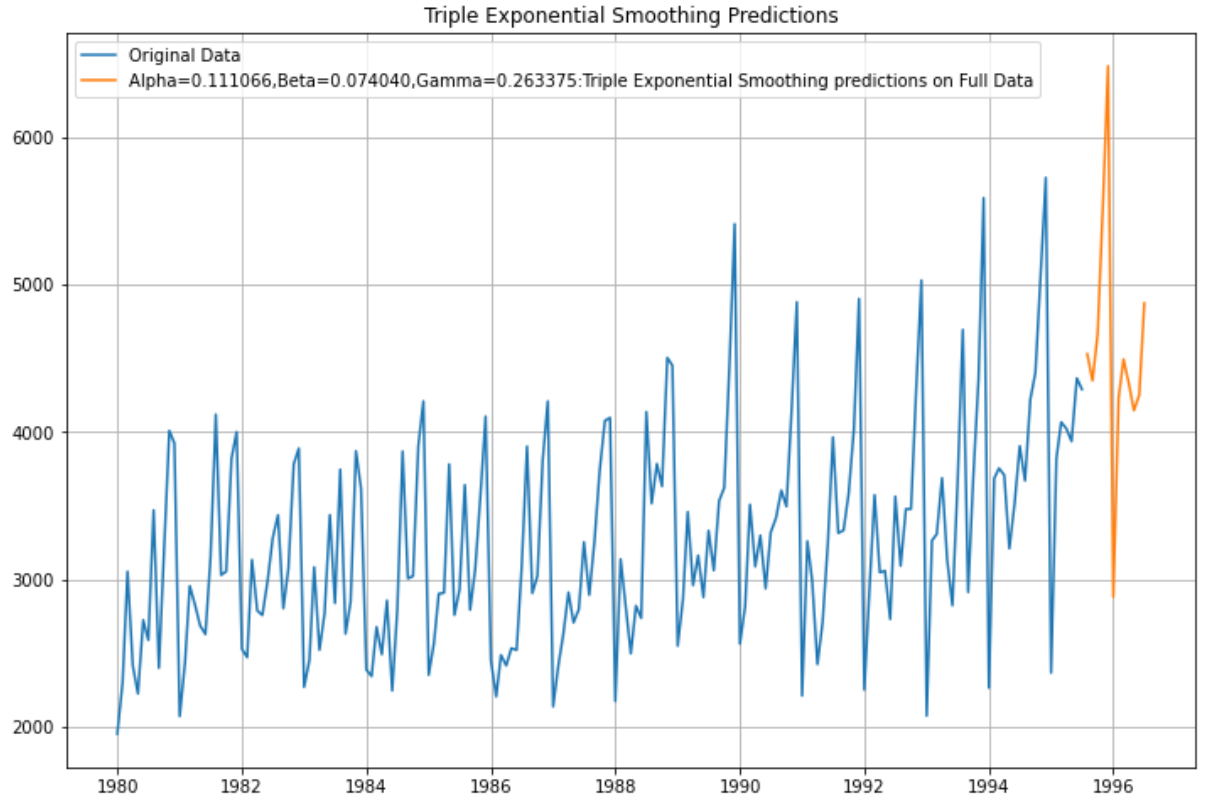
Based on the model building exercise, we shall select two models-

- **Triple Exponential Smoothing (Multiplicative)**

  This model was applied on the Full data and the values were predicted for the upcoming 12 months. From 01-August-1995 to 01-July-1996.

```
1995-08-01    4529.234491
1995-09-01    4348.812375
1995-10-01    4665.193018
1995-11-01    5539.806082
1995-12-01    6481.529179
1996-01-01    2881.680844
1996-02-01    4231.098900
1996-03-01    4492.998841
1996-04-01    4328.110489
1996-05-01    4146.669529
1996-06-01    4250.042687
1996-07-01    4873.894549
Freq: MS, dtype: float64
```

  Above figure gives the predicted values for the next 12 months.



  The above plot shows the original data and the predicted data for next 12 months.

- **SARIMA(3,1,3)(3,0,3,6) Lowest AIC**

  Here, we have a scenario where our training data was stationary but our full data was not stationary. So, we will use the same parameters as our training data but with adding a level of differencing which is needed for the data to be stationary. SARIMA using the lowest AIC parameters achieved were applied.

SARIMAX Results

Dep. Variable:	SoftDrinkProduction			No. Observations:		187
Model:	SARIMAX(3, 1, 3)x(3, 0, 3, 6)			Log Likelihood		-1191.191
Date:	Sun, 09 Jan 2022			AIC		2408.382
Time:	14:40:42			BIC		2448.680
Sample:	01-01-1980			HQIC		2424.741
	- 07-01-1995					
Covariance Type:	opg					

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.8239	0.663	-1.242	0.214	-2.124	0.476
ar.L2	-0.1856	0.635	-0.292	0.770	-1.430	1.059
ar.L3	-0.1577	0.097	-1.620	0.105	-0.348	0.033
ma.L1	-0.0610	0.674	-0.090	0.928	-1.382	1.260
ma.L2	-0.5702	0.354	-1.610	0.107	-1.264	0.124
ma.L3	-0.0057	0.516	-0.011	0.991	-1.016	1.005
ar.S.L6	-0.0059	0.708	-0.008	0.993	-1.394	1.383
ar.S.L12	1.0167	0.023	44.471	0.000	0.972	1.062
ar.S.L18	0.0209	0.720	0.029	0.977	-1.390	1.432
ma.S.L6	-0.0791	0.705	-0.112	0.911	-1.462	1.304
ma.S.L12	-0.6806	0.126	-5.403	0.000	-0.927	-0.434
ma.S.L18	-0.0350	0.483	-0.073	0.942	-0.982	0.912
sigma2	1.142e+05	1.11e+04	10.308	0.000	9.25e+04	1.36e+05

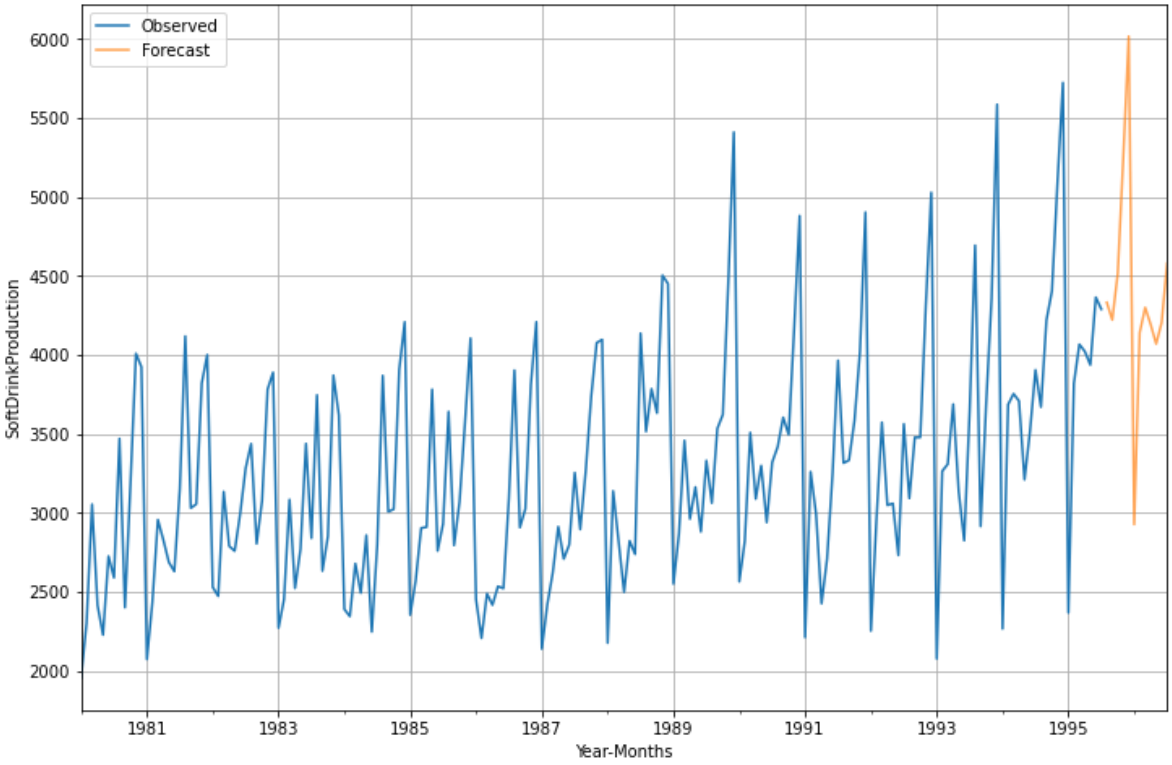
Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	19.35
Prob(Q):	0.97	Prob(JB):	0.00
Heteroskedasticity (H):	1.60	Skew:	0.51
Prob(H) (two-sided):	0.08	Kurtosis:	4.35

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

The SARIMA model was applied to the entire dataset. Above is the summary.

SoftDrinkProduction	mean	mean_se	mean_ci_lower	mean_ci_upper
1995-08-01	4332.475336	338.018526	3669.971198	4994.979473
1995-09-01	4221.920105	340.252490	3555.037479	4888.802731
1995-10-01	4516.842340	341.559592	3847.397842	5186.286839
1995-11-01	5245.001223	343.621182	4571.516082	5918.486363
1995-12-01	6018.332460	352.826077	5326.806056	6709.858864
1996-01-01	2926.501317	355.694775	2229.352368	3623.650266
1996-02-01	4140.655859	357.525834	3439.918101	4841.393618
1996-03-01	4300.039035	360.355047	3593.756120	5006.321949
1996-04-01	4192.893898	365.521500	3476.484923	4909.302874
1996-05-01	4070.052108	368.632006	3347.546652	4792.557564
1996-06-01	4204.666554	372.634081	3474.317175	4935.015933
1996-07-01	4580.472272	375.831783	3843.855514	5317.089031

Above are the predictions by the model for the next 12 months.



The above graph represents the original data along with forecast of the next 12 months.

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

Recommendation

- The management should rely on SARIMA model with the parameters achieved using the Akaike information criterion for predicting the future production because it has given the best predictions amongst all the models and the future production prediction follows the seasonality trend as well.
- Production is very low in the first half of every year. January month has the lowest production followed by June. Therefore, company should try searching for new markets where soft drinks are in demand even in the offseason for increasing the sales and subsequently the production.
- There is strong seasonality according to past performance; therefore offseason performance schemes for people associated with production should be launched to boost the production in these non- performing months.