

Learning for Life



BUSINESS REPORT TIME SERIES FORECASTING

ROSE WINE SALES TIME SERIES

PREPARED BY - RUPESH KUMAR / SUBMISSION DATE : 19/12/2021

Table of Contents

Questions	Description	Page No.
Problem : 2	Rose Wine Sales Time Series -For this particular assignment, the data of different types of wine sales in the 20th century is to be analysed. Both of these data are from the same company but of different wines. As an analyst in the ABC Estate Wines, you are tasked to analyse and forecast Wine Sales in the 20th century.	1
Problem : 2	Executive Summary &. Introduction	1
1	Read the data as an appropriate Time Series data and plot the data.	1-4
2	Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.	4-12
3	Split the data into training and test. The test data should start in 1991.	13-14
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 and simple average models. should also be built on the training data and check the performance on the test data using RMSE.	14-27
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 $\alpha = 0.05$.	27-28
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.	29-34
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.	35-40
8	Build a table (create a data frame) with all the models built along with their corresponding parameters and the respective RMSE values on the test data.	41
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.	42-44
10	Comment on the model thus built and report your findings and suggest the measures that the company should be taking for future sales.	44-46

List of Figures

Fig.No.	Figure Name	Page No.
1	Fig : 1 Plot of the Original Data	2
2	Plot of the Data with Time Stamp	4
3	Year on Year box-plot for theRose Wine Sales.	7
4	Monthly Box-Plot for the Rose Wine Sales Taking all the Years into Account	8
5	Month-plot of Rose Wine Sales Time Series	9
6	Time Series Plot for different months for different years.	10
7	Additive Decomposition	11
8	Multiplicative Decomposition	12
9	Plot of Train & Test Data.	14
10	Prediction on Test Dataset of Linear Regression Model	15
11	Prediction on Test Dataset of Navie Forecast Model	17
12	Prediction on Test Dataset of Simple Average Model	18
13	Whole Data With Moving Average	19
14	Moving Average on both the Training and Test data	20
15	Prediction on Test Dataset of Simple Exponential Model	21
16	Prediction on Test Dataset of Simple Exponential Model at $\alpha = 0.3$	22
17	Prediction on Test Dataset of Double Exponential Model at $\alpha = 0.3$ and $\beta = 0.3$	24
18	Prediction on Test Dataset of Triple Exponential Model at $\alpha = 0.10053559166352141$, $\beta = 2.9667113597565187e-06$, $\gamma = 3.752823638129799e-07$	25
19	Prediction on Test Dataset of Triple Exponential Model at $\alpha = 0.3$ $\beta = 0.4$ and $\gamma = 0.3$	26
20	Check for stationarity of the whole Time Series data. - Dickey-Fuller test	27
21	Check for stationarity after differencing of order 1 on whole Time Series data. - Dickey-Fuller test	28
22	Check for stationarity of the Train Time Series data. - Dickey-Fuller test	29
23	Check for stationarity after differencing of order 1 on Train Time Series data. - Dickey-Fuller tes	30
24	ACF plot of the original Data	32
25	Diagnostics Plot of Auto_SARIMA (0, 1, 2)(2, 0, 2, 12)	34

List of Figures

Fig.No.	Figure Name	Page No.
26	ACF / PACF Plot for ARIMA Model	35
27	Checking the Stationarity by taking the seasonal differencing of order 12- Dickey-Fuller Test	37
28	Seasonal Diff (12) ACF / PACF Plot for SARIMA Model	38
29	Diagnostics Plot of Manual_SARIMA (4, 1, 2)(2, 0, 1, 12)	40
30	Plot of the forecast on full data along with the confidence band	43

List of Tables

Table No.	Table Name	Page No.
1	Records of the Dataset Head & Tail	1
2	Time Stamp	2
3	Records of the Dataset Head & Tail with Time Stamp	3
4	Head of the Final Dataset for Time Series Forecasting of Sparkling Wine Sales	3
5	Summary of the Dataset	4
6	Appropriateness of Datatypes & Information of the Dataframe	5
7	Checking Null Values	5
8	Null Values Entries	5
9	Checking the Imputation of Null Values	6
10	Shape of the Dataset	6
11	Pivot Table of different months for different years.	9
12	Values for Trend , Seasonality & Residuals of Additive Decomposition Model	11
13	Values for Trend , Seasonality & Residuals of Multiplicative Decomposition Model	12
14	Checking the Records of the Train & Test Data.	13
15	Shape of the Train Data	13
16	Shape of the Test Data	14
17	Records of Training & Test Data with Time Instances for Linear Regression Model	15
18	Prediction on Test Data of Navie Model	16
19	Prediction on Test Data of Simple Average Model	18
20	Records of Dataset with Rolling Mean	19
21	Prediction on Test Data of Simple Exponential Model	21
22	Alpha Values for Train & Test RMSE Simple Exponential Smoothing Model	22
23	Alpha & Beta Values for Train & Test RMSE Double Exponential Smoothing Model	23
24	Prediction on Test Data of Triple Exponential Model	25
25	Alpha ,Beta and Gamma Values for Train & Test RMSE Triple Exponential Smoothing Model	26
26	Dickey - Fuller Test Result on WholeTS Data	28
27	Dickey - Fuller Test Result on WholeTS Data with differencing of order 1	28

List of Tables

Table No.	Table Name	Page No.
28	Dickey - Fuller Test Result on Train TS Data	29
29	Dickey - Fuller Test Result on Train TS Data with differencing of order 1	30
30	Combinations & AIC Values for Auto_ARIMA Model	31
31	Result Summary of Auto_ARIMA (0,1,2)	31
32	Combinations & AIC Values for Auto_SRIMA Model With Seasonality as 12	33
33	Result Summary of Auto_SARIMA(0, 1, 2)(2, 0, 2, 12)	33
34	SummaryFrame of Auto_SARIMA(0, 1, 2)(2, 0, 2, 12) at alpha = 0.05	34
35	Result Manual_ARIMA (4,1,2) Model	36
36	Dickey - Fuller Test Result on Train Data with seasonal diff of order 12	37
37	Result Manual_SARIMA (4,1,2) (2,0,1,12) Model	39
38	SummaryFrame of Manual_SARIMA(4,1,2) (2,0,1,12) Model at alpha = 0.05	40
39	Result of All Models With Parameter and Test RMSE	41
40	Result of Top 5 Models With Parameter and Test RMSE	41
41	Result Full Model TripleExponentialSmoothing Model with Alpha=0.100,Beta=2.966e06,Gamma=3.7528e07	42
42	Confidence Interval of Full Model for Forecast	43

Problem Statement of Rose Wine Sales Time Series :

For this particular assignment, the data of different types of wine sales in the 20th century is to be analysed. Both of these data are from the same company but of different wines. As an analyst in the ABC Estate Wines, you are tasked to analyse and forecast Wine Sales in the 20th century.

Executive Summary

ABC Estate wines have the data of different types of wine sales in the 20th century Both of these data are from the same company but of different wines **Sparkling Wine & Rose Wine Sales**. As an analyst in the ABC Estate Wines, Our tasked to analyse and forecast different types of Wine Sales in the 20th century. Here we are analyse and forecast **Rose Wine Sales** in the 20th century.

Introduction

The purpose of this whole exercise is to explore the dataset , analyse and forecast **Rose Wine Sales** in the 20th century.Here we perform the exploratory data analysis & apply various time series forecasting models like **Linear Regression , Navie Forecast ,Simple Average , Moving Average and various kind of exponential smoothing models like (Simple , Double , Triple Exponential) and ARIMA / SARIMA** models on the Rose Wine Sales dataset and check their **RMSE** on the test data , model which gives the least **RMSE** will be the final model for us to analyse and forecast the Rose Wine Sales in the 20th century.

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

Checking the Records of the Dataset -
Head of the Dataset - First 10 Records of the Dataset.
Tail of the Dataset - Last 10 Records of the Dataset.

YearMonth Rose			YearMonth Rose		
0	1980-01	112.0	177	1994-10	51.0
1	1980-02	118.0	178	1994-11	63.0
2	1980-03	129.0	179	1994-12	84.0
3	1980-04	99.0	180	1995-01	30.0
4	1980-05	116.0	181	1995-02	39.0
5	1980-06	168.0	182	1995-03	45.0
6	1980-07	118.0	183	1995-04	52.0
7	1980-08	129.0	184	1995-05	28.0
8	1980-09	205.0	185	1995-06	40.0
9	1980-10	147.0	186	1995-07	62.0

Tab:1 Records of the Dataset Head & Tail

Plot of the Data

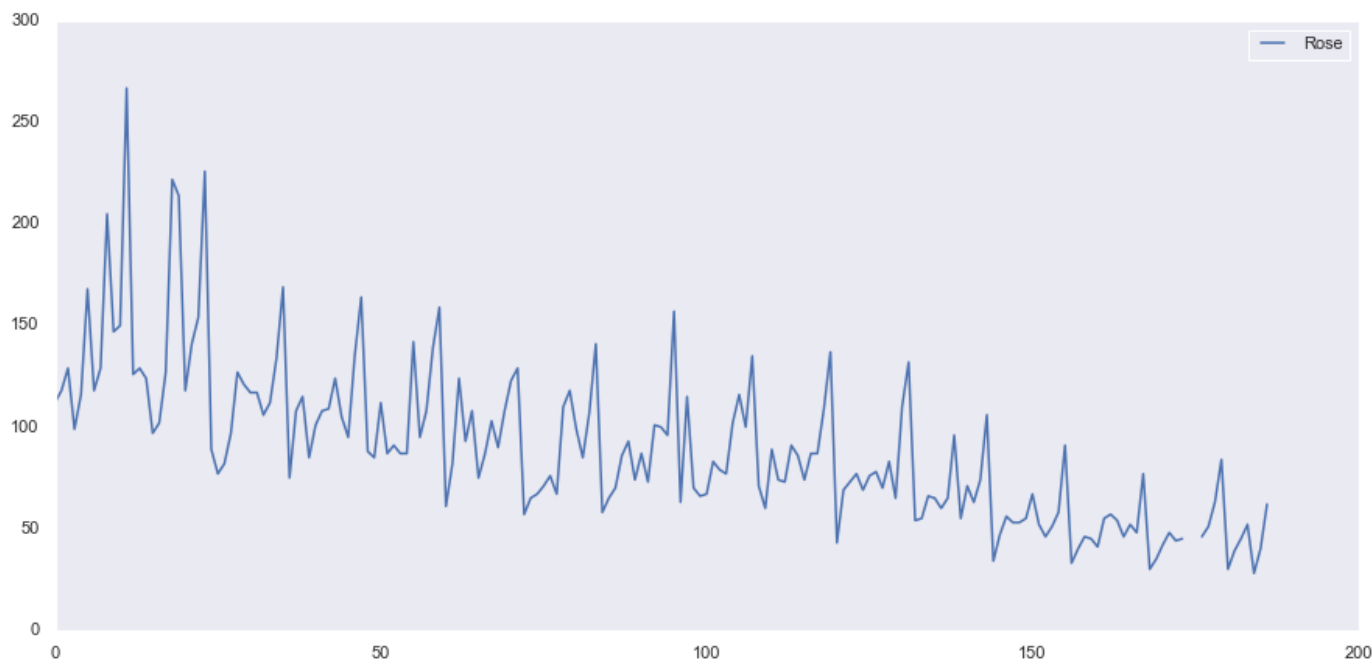


Fig : 1 Plot of the Original Data

Note :

Though the above plot looks like a Time Series plot, notice that the X-Axis is not time. In order to make the X-Axis as a Time Series, we need to pass the date range manually through a command in Pandas.

Adding the Time Stamp into Dataset.

Note :

The time stamps is defined as a monthly Time Series after looking at the data. Data given to us is start from Jan 1980 to July 1995 , In order to make the X-Axis as a Time Series, we need to pass the date range from start= '1/1/1980', end='8/1/1995' as we know that the when we using this data_range function always creates the time stamp with the last day of that month,when we include this into dataset we will see that the first observation of the time stamp will be 31 Jan 1980 , so in order to include the July we will write the end period as end='8/1/1995' now it automatically take the last day of July 31 for this dataset,as we know the frequency is "monthly" so we set it freq = "M".

```
DatetimeIndex(['1980-01-31', '1980-02-29', '1980-03-31', '1980-04-30',
              '1980-05-31', '1980-06-30', '1980-07-31', '1980-08-31',
              '1980-09-30', '1980-10-31',
              ...,
              '1994-10-31', '1994-11-30', '1994-12-31', '1995-01-31',
              '1995-02-28', '1995-03-31', '1995-04-30', '1995-05-31',
              '1995-06-30', '1995-07-31'],
              dtype='datetime64[ns]', length=187, freq='M')
```

Tab:2 Time Stamp

Observation:

Last date of every month is being used here from 31 Jan 1980 till 31 July 1995.

CHECKING THE RECORDS OF THE DATASET AFTER ADDING TIME_STAMP.

Checking the Records of the Dataset -
Head of the Dataset - First 10 Records of the Dataset.
Tail of the Dataset - Last 10 Records of the Dataset.

	YearMonth	Rose	Time_Stamp		YearMonth	Rose	Time_Stamp
0	1980-01	112.0	1980-01-31	177	1994-10	51.0	1994-10-31
1	1980-02	118.0	1980-02-29	178	1994-11	63.0	1994-11-30
2	1980-03	129.0	1980-03-31	179	1994-12	84.0	1994-12-31
3	1980-04	99.0	1980-04-30	180	1995-01	30.0	1995-01-31
4	1980-05	116.0	1980-05-31	181	1995-02	39.0	1995-02-28
5	1980-06	168.0	1980-06-30	182	1995-03	45.0	1995-03-31
6	1980-07	118.0	1980-07-31	183	1995-04	52.0	1995-04-30
7	1980-08	129.0	1980-08-31	184	1995-05	28.0	1995-05-31
8	1980-09	205.0	1980-09-30	185	1995-06	40.0	1995-06-30
9	1980-10	147.0	1980-10-31	186	1995-07	62.0	1995-07-31

Tab:3 Records of the Dataset Head & Tail with Time Stamp

Final Dataset for Time Series Forecasting

Rose Wine Sales	
Time_Stamp	
1980-01-31	112.0
1980-02-29	118.0
1980-03-31	129.0
1980-04-30	99.0
1980-05-31	116.0

Tab:4 Head of the Final Dataset for Time Series Forecasting of Rose Wine Sales

Note:
In the final dataset we have Time Stamp as index and records of the Rose Wine Sales , we also drop the year-month column as it is not useful for us. Now we successfully creates the Time Series object, let us go ahead and analyze the Time Series plot that we got.

Plot of the Data After Adding Time_Stamp.

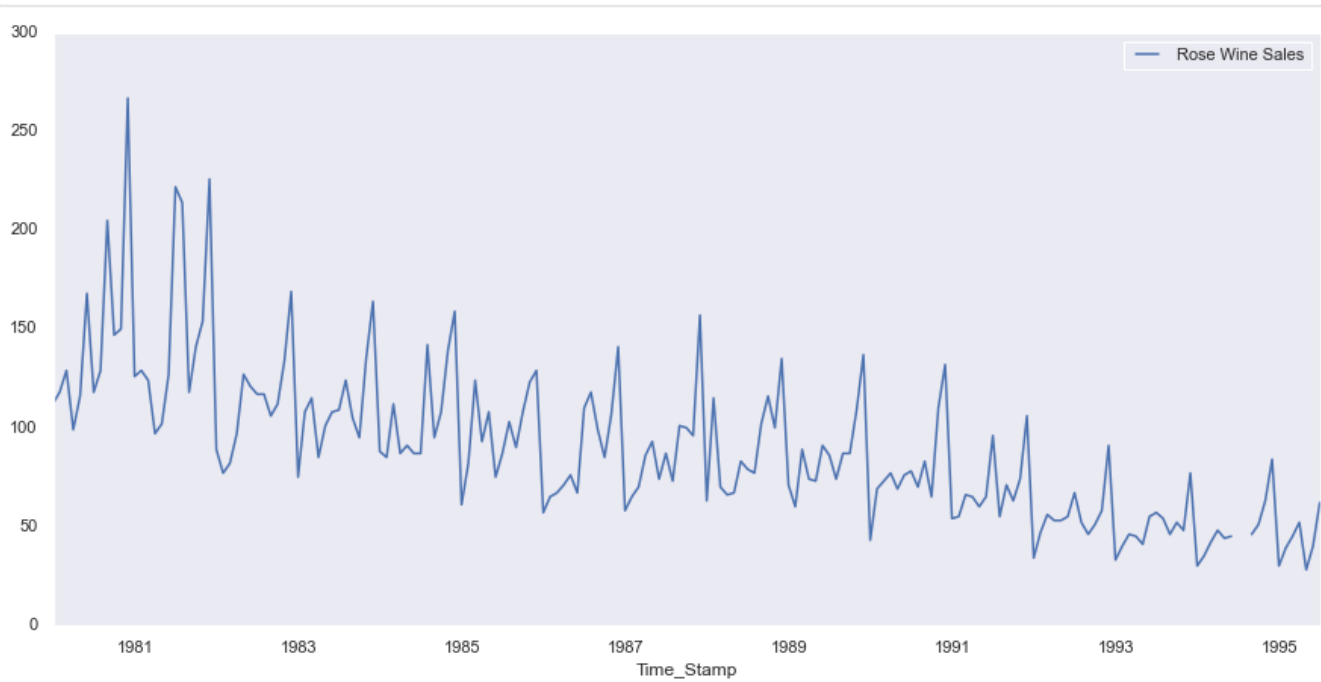


Fig : 2 Plot of the Data with Time Stamp

Insights

We can see that there is a downward trend in the series with a seasonal pattern associated as well. Moreover we found that some data is missing from the series too. We will check the missing values and impute them as well by suitable method.

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

Exploratory Data Analysis -

Checking the Summary of the Dataset.

The describe () method computes and displays summary statistics for a Python dataframe. From the above table we can infer the count, mean, std , 25% , 50% ,75% and min & max values of the Rose Wine Sales column present in the dataset.

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

Tab:5 Summary of the Dataset

Insights

- Rose Wine Sales ranges from a minimum of 28 to maximum of 267 .
- Mean of the Rose Wine Sales is around 90.394595.
- Standard Deviation of the Rose Wine Sales is 39.175344.
- 25% , 50% (median) and 75 % of Rose Wine Sales are 63 ,86 and 112.

Checking the Appropriateness of Data-types & Information of the Dataframe.

The info() function is used to print a concise summary of a DataFrame. This method prints information about a DataFrame including the index d-type and column d-types, non-null values and memory usage.

S.No.	Features / Columns	Non-Null Count	Dtype	Memory Usage
1	Rose Wine Sales	185 non-null	float64	2.9KB

Tab:6 Appropriateness of Datatypes & Information of the Dataframe

Insights

From the above results we can see that there is 2 null values present in the dataset.Their are total 187 entries of Rose wines Sales as per Monthly frequency in this dataset,indexed from 1980-01-31 to 1995-07-31.Rose Wine Sales column have d-type of float64. Memory used by the dataset: 2.9 KB.

Checking the Null Values in the Dataset.

S.No.	Features / Columns	Null Count
1	Rose Wine Sales	2

Tab:7 Checking Null Values

Insights

There is 2 Null Values Present in the Dataset.So we need to impute the missing value present in the data with some meaningful value by using suitable null value imputation method.

Observation

We found in year 1994 there are 2 missing value for 1994-07-31 and 1994-08-31

Rose Wine Sales	
Time_Stamp	
1994-01-31	30.0
1994-02-28	35.0
1994-03-31	42.0
1994-04-30	48.0
1994-05-31	44.0
1994-06-30	45.0
1994-07-31	NaN
1994-08-31	NaN
1994-09-30	46.0
1994-10-31	51.0
1994-11-30	63.0
1994-12-31	84.0

Tab:8 Null Values Entries

By using Forward Fillna Method (ffill()) function is used to fill the missing value in the dataframe. 'ffill' stands for 'forward fill' and will propagate last valid observation forward) we impute the null values present in the rose wine sales dataset.

Rose Wine Sales	
Time_Stamp	
1994-01-31	30.0
1994-02-28	35.0
1994-03-31	42.0
1994-04-30	48.0
1994-05-31	44.0
1994-06-30	45.0
1994-07-31	45.0
1994-08-31	45.0
1994-09-30	46.0
1994-10-31	51.0
1994-11-30	63.0
1994-12-31	84.0

Result

We successfully impute the null values by forward fillna method. Now we donot have any null values in the data.
Now we donot have any null values in year 1994 i.e- .1994-07-31 and 1994-08-31

Tab:9 Checking the Imputation of Null Values

Checking the Shape of the Dataframe.

No. of Rows	No. of Columns
187	1

Tab:10 Shape of the Dataset

Insights

The Rose.csv data set has 187 observations (rows) and 1 variable (column named as Rose Wine Sales) in the dataset.

Data Visualization of the Time Series

Note

A box-plot gives a nice summary of one or several numeric variables. The line that divides the box into 2 parts represents the median of the data. The end of the box shows the upper and lower quartiles. The extreme lines show the highest and lowest value excluding outliers.

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

Year on Year boxplot for the Rose Wine Sales.

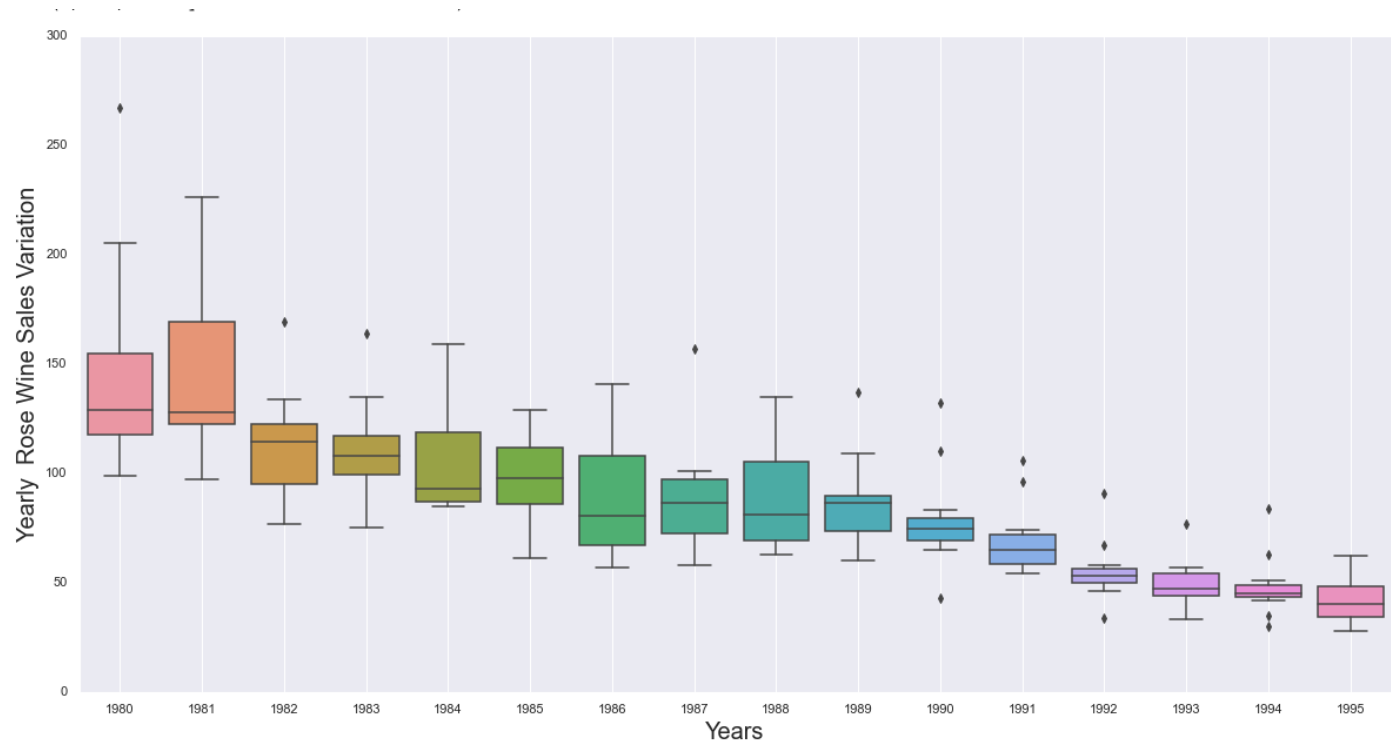


Fig : 3 Year on Year boxplot for theRose Wine Sales.

Insights

- As we got to know from the Time Series plot, the box-plots over here also indicates a measure of trend being present. Also, we see that the Rose Wine Sales have outliers for the years.The yearly boxplots also shows that the Sales have decreased year after year.
- Box-plot of Year 1980 and 1981 have max median value, from this we can clearly infer that year 1981 have maximum Rose Wine Sales.
- Box-plot of Year 1995 have min median value,we can clearly infer that year 1995 have minimum Rose Wine Sales.

Monthly Box-Plot for the Rose Wine Sales Taking all the Years into Account

Note

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

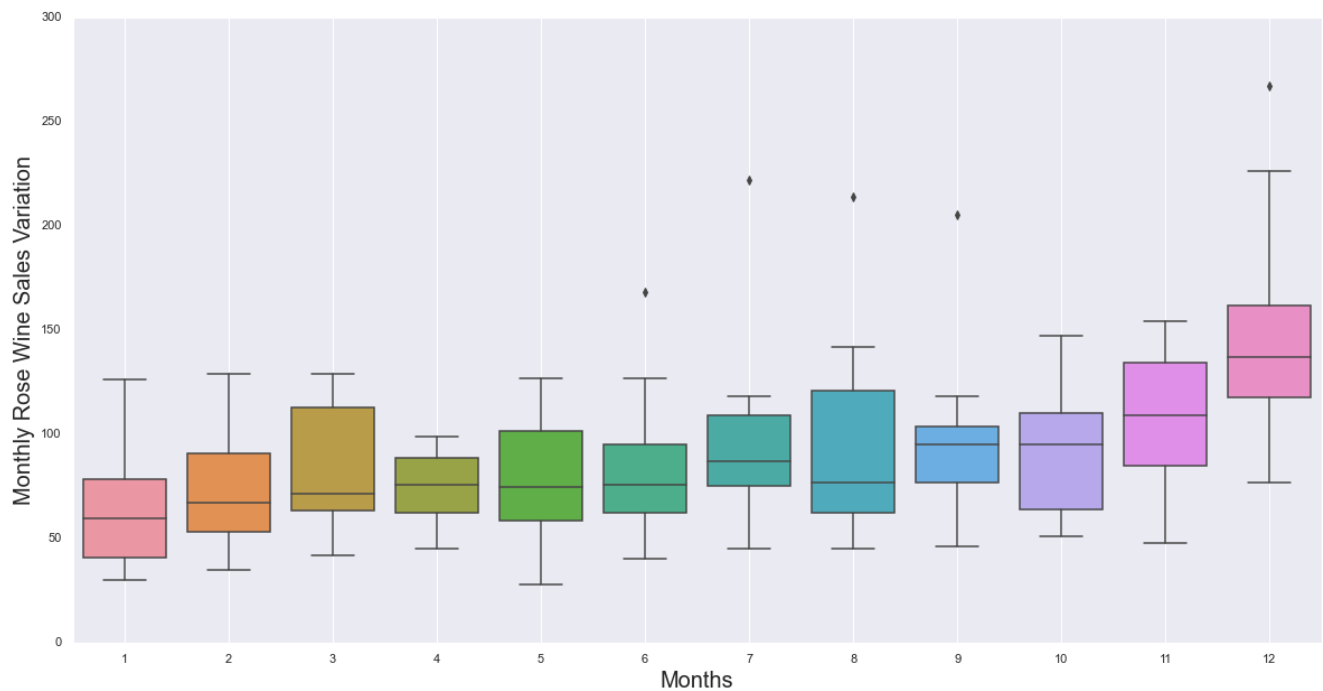


Fig : 4 Monthly Box-Plot for the Rose Wine Sales Taking all the Years into Account

Insights

- The Box-Plots for the monthly Rose Wine Sales for different years very few outliers in the month 6 , 7, 8,9 and 12 show outliers , rest doesn't show any outliers.
- From September to December the Rose Wine Sales increasing , so this the period where the Rose Wine Sales is highest.
- December is the month of highest Rose Wine Sales every year.whereas jan is the month of lowest Rose wine sales.
- There is seasonality also every year from September to December the Rose Wine Sales increasing.

Monthplot of Rose Wine Sales Time Series

Note

This plot shows the variations of the Rose Wine Sales values across the months & this red line is the mean value of the Rose Wine Sales for every month.

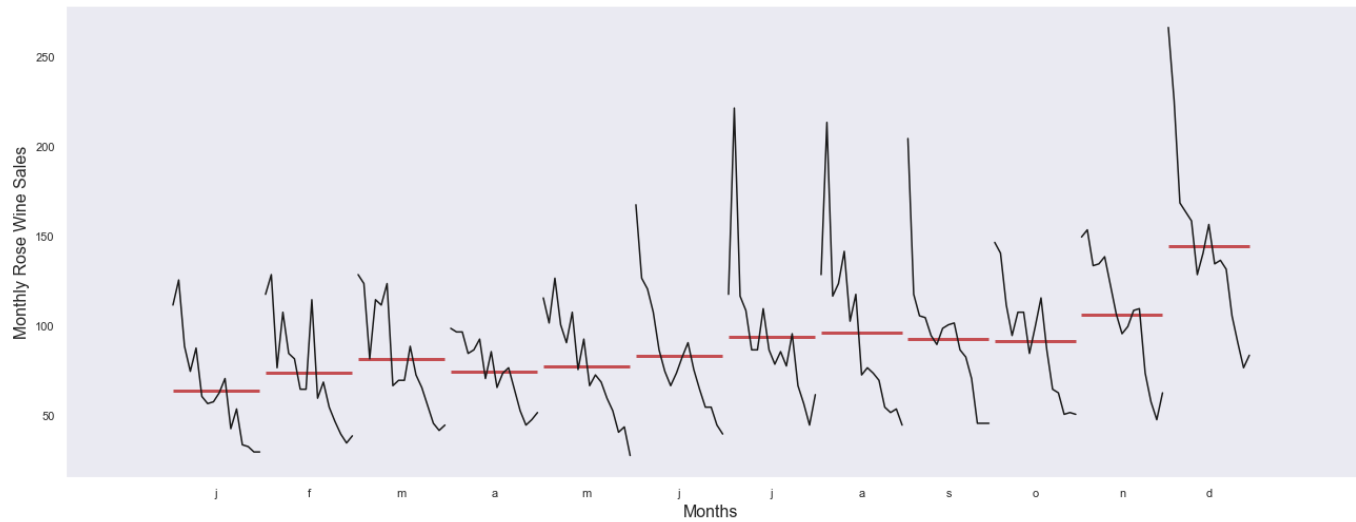


Fig : 5 Monthplot of Rose Wine Sales Time Series

Insights

- As noticed in the above box-plot we get same result from here too. From September to December Rose Wine Sales goes on increasing.
- December month have the highest sales of the Rose Wine while Jan month have low sales of the Rose Wine.

Time Series Plot for different months for different years.

Time_Stamp	1	2	3	4	5	6	7	8	9	10	11	12
Time_Stamp												
1980	112.0	118.0	129.0	99.0	116.0	168.0	118.0	129.0	205.0	147.0	150.0	267.0
1981	126.0	129.0	124.0	97.0	102.0	127.0	222.0	214.0	118.0	141.0	154.0	226.0
1982	89.0	77.0	82.0	97.0	127.0	121.0	117.0	117.0	106.0	112.0	134.0	169.0
1983	75.0	108.0	115.0	85.0	101.0	108.0	109.0	124.0	105.0	95.0	135.0	164.0
1984	88.0	85.0	112.0	87.0	91.0	87.0	87.0	142.0	95.0	108.0	139.0	159.0
1985	61.0	82.0	124.0	93.0	108.0	75.0	87.0	103.0	90.0	108.0	123.0	129.0
1986	57.0	65.0	67.0	71.0	76.0	67.0	110.0	118.0	99.0	85.0	107.0	141.0
1987	58.0	65.0	70.0	86.0	93.0	74.0	87.0	73.0	101.0	100.0	96.0	157.0
1988	63.0	115.0	70.0	66.0	67.0	83.0	79.0	77.0	102.0	116.0	100.0	135.0
1989	71.0	60.0	89.0	74.0	73.0	91.0	86.0	74.0	87.0	87.0	109.0	137.0
1990	43.0	69.0	73.0	77.0	69.0	76.0	78.0	70.0	83.0	65.0	110.0	132.0
1991	54.0	55.0	66.0	65.0	60.0	65.0	96.0	55.0	71.0	63.0	74.0	106.0
1992	34.0	47.0	56.0	53.0	53.0	55.0	67.0	52.0	46.0	51.0	58.0	91.0
1993	33.0	40.0	46.0	45.0	41.0	55.0	57.0	54.0	46.0	52.0	48.0	77.0
1994	30.0	35.0	42.0	48.0	44.0	45.0	45.0	45.0	46.0	51.0	63.0	84.0
1995	30.0	39.0	45.0	52.0	28.0	40.0	62.0	NaN	NaN	NaN	NaN	NaN

Tab:11 Pivot Table of different months for different years.

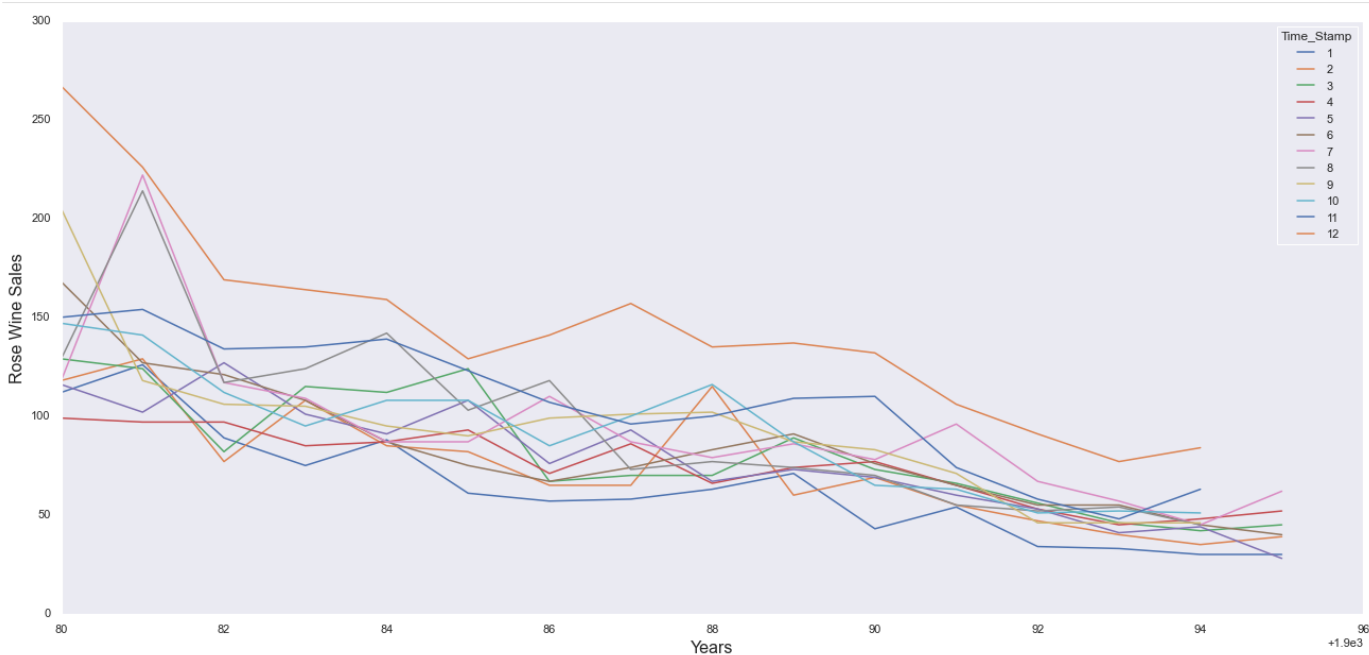


Fig : 6 Time Series Plot for different months for different years.

Note

This plot gives us information about the monthly trend across the years. Here in this plot every line is a month tells us about the sales of Rose Wines of each month across the year. This is way to show year on year monthly trend.

Insights

- From the above plot we clearly infer that December month have highest sales of Rose Wine.
- Jan month have the lowest sales of the Rose Wine.

Decomposition of the Rose Wine Time Series

Additive Decomposition Model -

Additive model analysis is a newly emerged approach for time-series modeling. ... Under this setting, the given time-series would be decomposed into four components: trend, seasonality, cyclic patterns, and a random component. The formula is as follows: $y(t)=g(t)+s(t)+h(t)+\epsilon(t)$.

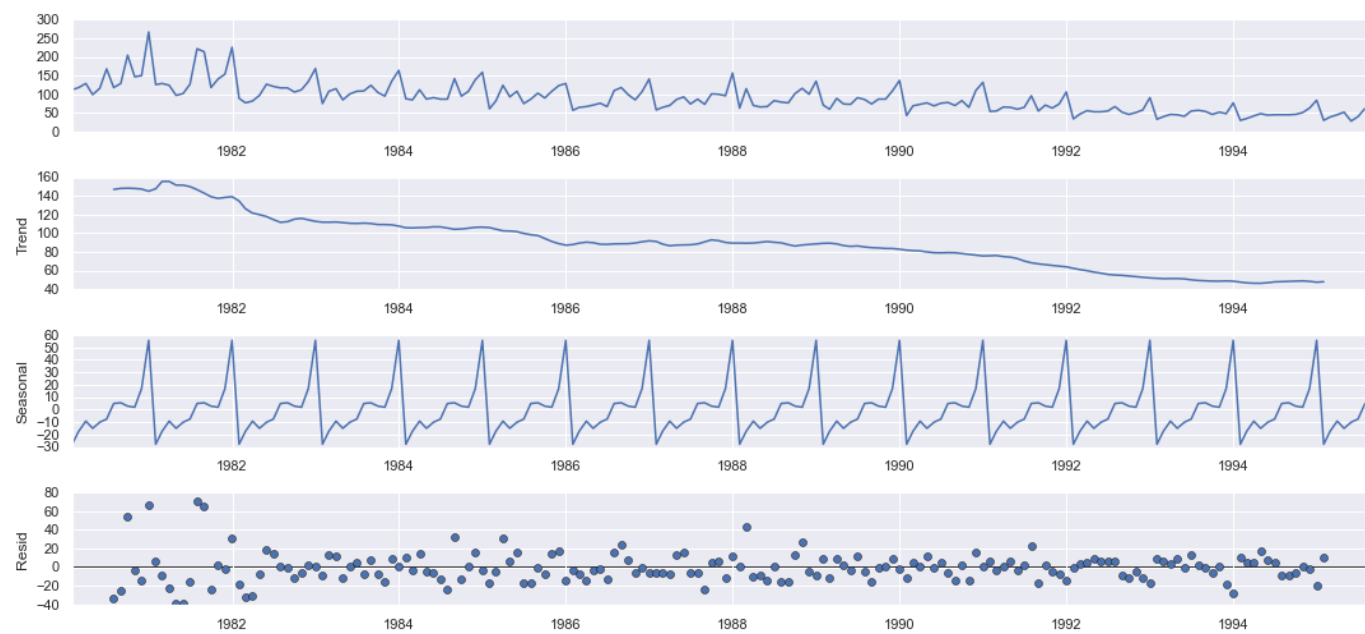


Fig : 7 Additive Decomposition

Insights

- As per the 'Additive' Decomposition Model, we see that there is a down-trend in the years of the data. There is a seasonality as well.
- Errors are not randomly distributed ,showing some kind of pattern.
- We see that the residuals are located around 0 from the plot of the residuals in the decomposition.
- Even the peaks of the original time series is not constant having change in the values.

Values for Trend , Seasonality & Residuals of Additive Decomposition Model

Trend				Residual	
Time_Stamp		Time_Stamp		Time_Stamp	
1980-01-31	NaN	1980-01-31	-27.903092	1980-01-31	NaN
1980-02-29	NaN	1980-02-29	-17.431663	1980-02-29	NaN
1980-03-31	NaN	1980-03-31	-9.279878	1980-03-31	NaN
1980-04-30	NaN	1980-04-30	-15.092378	1980-04-30	NaN
1980-05-31	NaN	1980-05-31	-10.190592	1980-05-31	NaN
1980-06-30	NaN	1980-06-30	-7.672735	1980-06-30	NaN
1980-07-31	147.083333	1980-06-30	-7.672735	1980-07-31	-33.963575
1980-08-31	148.125000	1980-07-31	4.880241	1980-08-31	-24.585797
1980-09-30	148.375000	1980-08-31	5.460797	1980-09-30	53.844759
1980-10-31	148.083333	1980-09-30	2.780241	1980-10-31	-2.960797
1980-11-30	147.416667	1980-10-31	1.877464	1980-11-30	-14.269130
1980-12-31	145.125000	1980-11-30	16.852464	1980-12-31	66.155870
1981-01-31	147.750000	1980-12-31	55.719130	1981-01-31	6.153092
1981-02-28	155.625000	1981-01-31	-27.903092	1981-02-28	-9.193337
1981-03-31	155.541667	1981-02-28	-17.431663	1981-03-31	-22.261789
1981-04-30	151.666667	1981-03-31	-9.279878	1981-04-30	-39.574289
1981-05-31	151.583333	1981-04-30	-15.092378	1981-05-31	-39.392741
1981-06-30	150.041667	1981-05-31	-10.190592	1981-06-30	-15.368932
1981-07-31	146.791667	1981-06-30	-7.672735	1981-07-31	70.328092
1981-08-31	143.083333	1981-07-31	4.880241	1981-08-31	65.455870
1981-09-30	139.166667	1981-08-31	5.460797	1981-09-30	-23.946908
1981-10-31	137.416667	1981-09-30	2.780241	1981-10-31	1.705870
1981-11-30	138.458333	1981-10-31	1.877464	1981-11-30	-1.310797
1981-12-31	139.250000	1981-11-30	16.852464	1981-12-31	31.030870
		1981-12-31	55.719130		
Name: trend, dtype: float64		Name: seasonal, dtype: float64		Name: resid, dtype: float64	

Tab:12 Values for Trend , Seasonality & Residuals of Additive Decomposition Model

Multiplicative Decomposition Model

Multiplicative Model: $Y_t = T_t * S_t * I_t$ It is considered when the resultant time series is the product of the components. A series may be considered multiplicative series when the seasonal fluctuations increase as trend increases. A multiplicative time series can be transformed into an additive series by taking log transformation i.e. $\log(Y_t) = \log(T_t) + \log(S_t) + \log(I_t)$

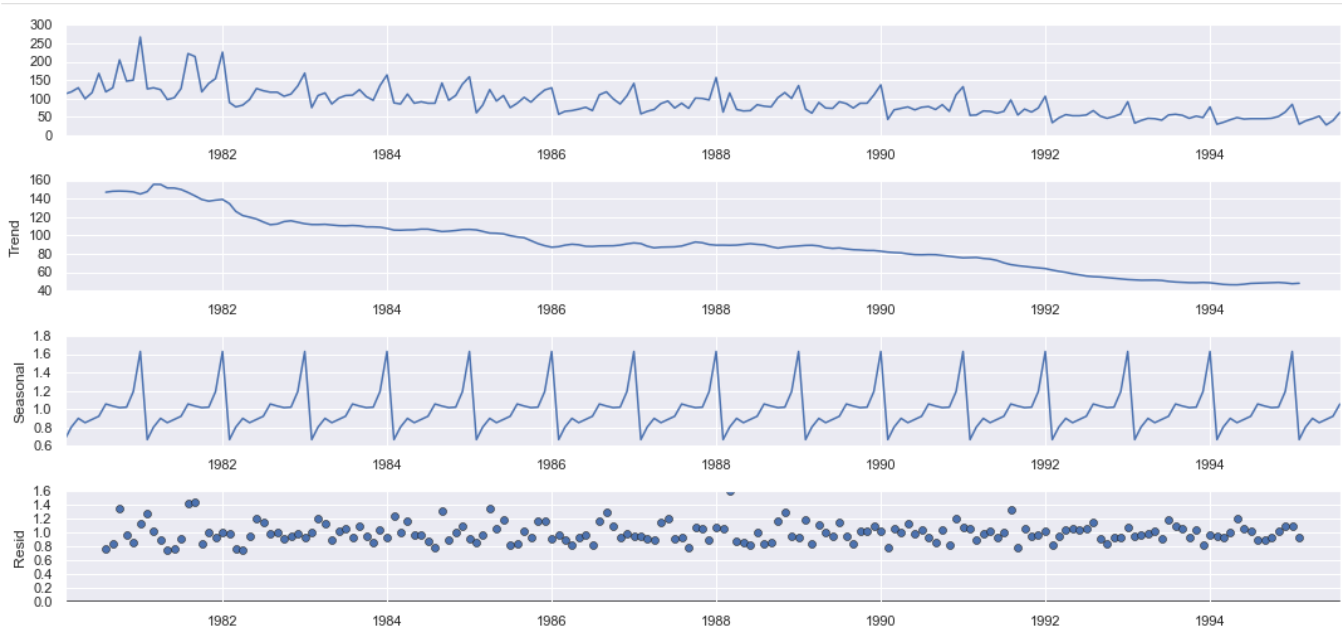


Fig : 8 Multiplicative Decomposition

Insights

As per the 'Multiplicative' Decomposition Model, we see that there is a pronounced down-trend in the years of the data. There is a seasonality as well.

For the multiplicative series, we see that residuals are located around 1.

Values for Trend , Seasonality & Residuals of Multiplicative Decomposition Model

Trend		Seasonality		Residual	
Time_Stamp		Time_Stamp		Time_Stamp	
1980-01-31	NaN	1980-01-31	0.670182	1980-01-31	NaN
1980-02-29	NaN	1980-02-29	0.806224	1980-02-29	NaN
1980-03-31	NaN	1980-03-31	0.901278	1980-03-31	NaN
1980-04-30	NaN	1980-04-30	0.854154	1980-04-30	NaN
1980-05-31	NaN	1980-05-31	0.889531	1980-05-31	NaN
1980-06-30	NaN	1980-06-30	0.924099	1980-06-30	NaN
1980-07-31	147.083333	1980-07-31	1.057682	1980-07-31	0.758514
1980-08-31	148.125000	1980-08-31	1.035066	1980-08-31	0.841382
1980-09-30	148.375000	1980-09-30	1.017753	1980-09-30	1.357534
1980-10-31	148.083333	1980-10-31	1.022688	1980-10-31	0.970661
1980-11-30	147.416667	1980-11-30	1.192494	1980-11-30	0.853274
1980-12-31	145.125000	1980-12-31	1.628848	1980-12-31	1.129506
Name: trend, dtype: float64		Name: seasonal, dtype: float64		Name: resid, dtype: float64	

Tab:13 Values for Trend , Seasonality & Residuals of Multiplicative Decomposition Model

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

Splitting of the data into Train and Test.

Note

Training Data is till the end of 1990. Test Data is from the beginning of 1991 to the last time stamp provided.

Checking the Records of the Train & Test Data.

First few rows of Training Data		First few rows of Test Data	
Rose Wine Sales		Rose Wine Sales	
Time_Stamp		Time_Stamp	
1980-01-31	112.0	1991-01-31	54.0
1980-02-29	118.0	1991-02-28	55.0
1980-03-31	129.0	1991-03-31	66.0
1980-04-30	99.0	1991-04-30	65.0
1980-05-31	116.0	1991-05-31	60.0
Last few rows of Training Data		Last few rows of Test Data	
Rose Wine Sales		Rose Wine Sales	
Time_Stamp		Time_Stamp	
1990-08-31	70.0	1995-03-31	45.0
1990-09-30	83.0	1995-04-30	52.0
1990-10-31	65.0	1995-05-31	28.0
1990-11-30	110.0	1995-06-30	40.0
1990-12-31	132.0	1995-07-31	62.0

Tab:14 Checking the Records of the Train & Test Data.

Checking the Shape of the Train & Test Data.

Shape attribute tells us number of observations and variables we have in the data set. It is used to check the dimension of data.

No. of Rows	No. of Columns
132	1

Tab:15 Shape of the Train Data

No. of Rows	No. of Columns
55	1

Tab:16 Shape of the Test Data

Plot of Train & Test Data.

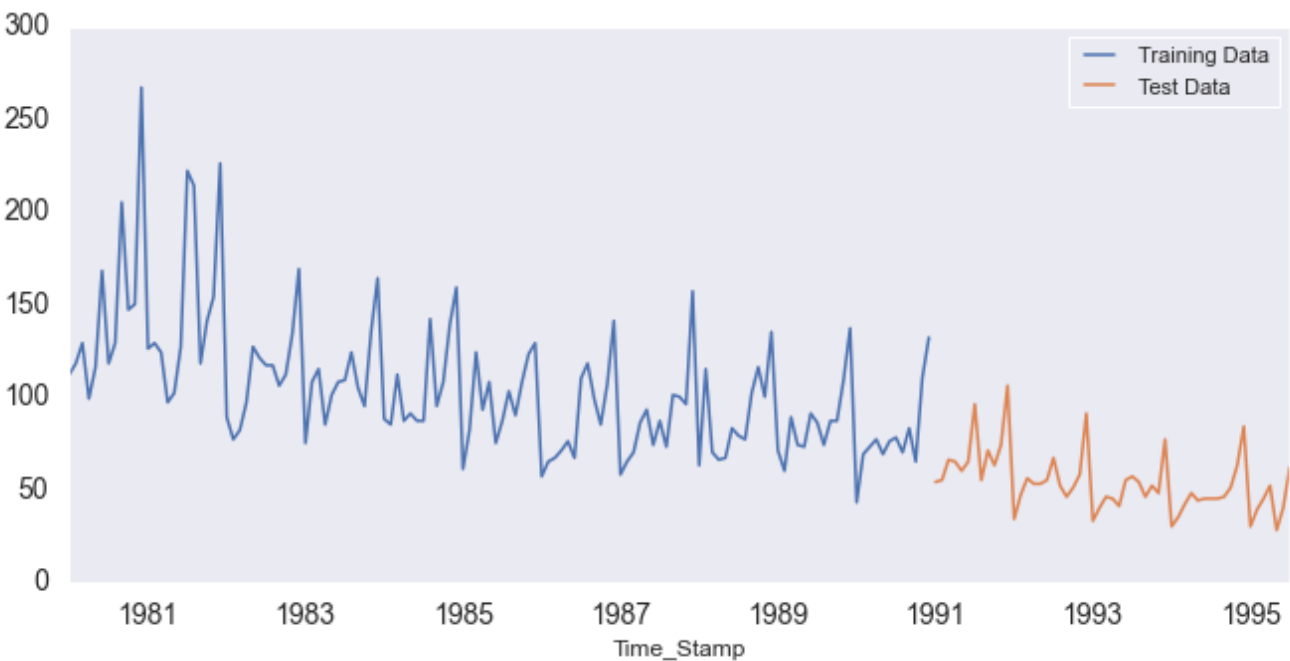


Fig : 9 Plot of Train & Test Data.

Insights

- Blue color represents the train data ,training Data is till the end of 1990.
- Orange color represents the test data ,test Data is from the beginning of 1991 to the last time stamp provided.

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 and simple average models. should also be built on the training data and check the performance on the test data using RMSE.

Model 1: Linear Regression

Note

For this particular linear regression, we are going to regress the 'Rose Wine Sales' variable against the order of the occurrence. For this we need to modify our training data before fitting it into a linear regression.

Observation

We see that we have successfully the generated the numerical time instance order for both the training and test set. Now we will add these values in the training and test set.

First few rows of Training Data			First few rows of Test Data		
Time_Stamp	Rose Wine Sales	time	Time_Stamp	Rose Wine Sales	time
1980-01-31	112.0	1	1991-01-31	54.0	133
1980-02-29	118.0	2	1991-02-28	55.0	134
1980-03-31	129.0	3	1991-03-31	66.0	135
1980-04-30	99.0	4	1991-04-30	65.0	136
1980-05-31	116.0	5	1991-05-31	60.0	137
Last few rows of Training Data			Last few rows of Test Data		
Time_Stamp	Rose Wine Sales	time	Time_Stamp	Rose Wine Sales	time
1990-08-31	70.0	128	1995-03-31	45.0	183
1990-09-30	83.0	129	1995-04-30	52.0	184
1990-10-31	65.0	130	1995-05-31	28.0	185
1990-11-30	110.0	131	1995-06-30	40.0	186
1990-12-31	132.0	132	1995-07-31	62.0	187

Tab:17 Records of Training & Test Data with Time Instances for Linear Regression Model

Note

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

Building A Linear Regression Model -

Invoke the Linear Regression function (from sklearn.linear_model import LinearRegression) fit the function on the train & test data and build the linear regression model. In this problem we are advised to build linear regression model and check the performance of Predictions on Test sets using RMSE.

Prediction on Test Dataset

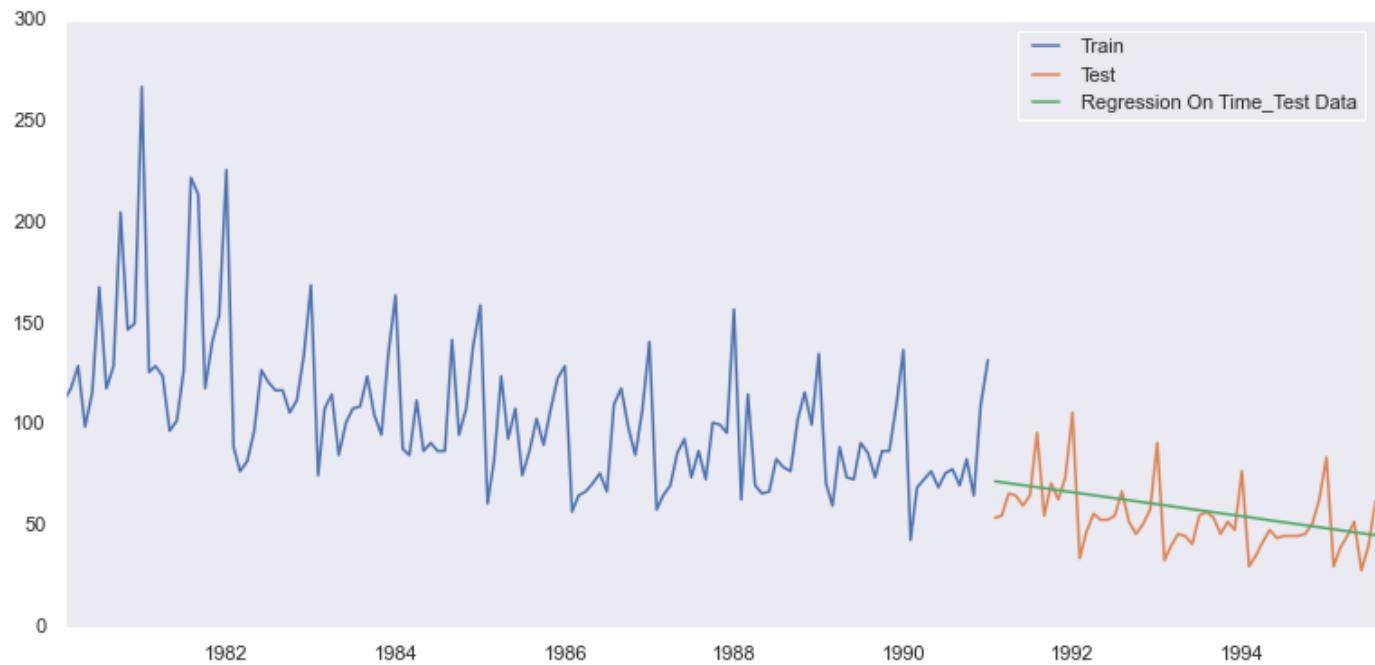


Fig : 10 Prediction on Test Dataset of Linear Regression Model

Observation

Here we get a linear line following the previous patterns.

Model Evaluation Linear Regression

RMSE - The root mean square error (RMSE) for a regression model is similar to the standard deviation (SD) for the ideal measurement model. The SD estimates the deviation from the sample mean x. The RMSE estimates the deviation of the actual y-values from the regression line.

Root mean square error, which is a metric that tells us the average distance between the predicted values from the model and the actual values in the dataset. The lower the RMSE, the better a given model is able to “fit” a dataset.

Test Data - **RMSE of Linear Regression** Model is - 15.276

Model 2: Naive Model , { Navie Approach: $\hat{y}_{t+1}=y_t$ }

Note

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

Building Navie Model

`NaiveModel_test['naive'] = np.asarray(train['Rose Wine Sales'])
[len(np.asarray(train['Rose Wine Sales']))-1]`

Prediction on Test Dataset

	Rose Wine Sales	naive
Time_Stamp		
1991-01-31	54.0	132.0
1991-02-28	55.0	132.0
1991-03-31	66.0	132.0
1991-04-30	65.0	132.0
1991-05-31	60.0	132.0

Tab:18 Prediction on Test Data of Navie Model

Observation

We get the same value of Navie forecast for the entire test data.

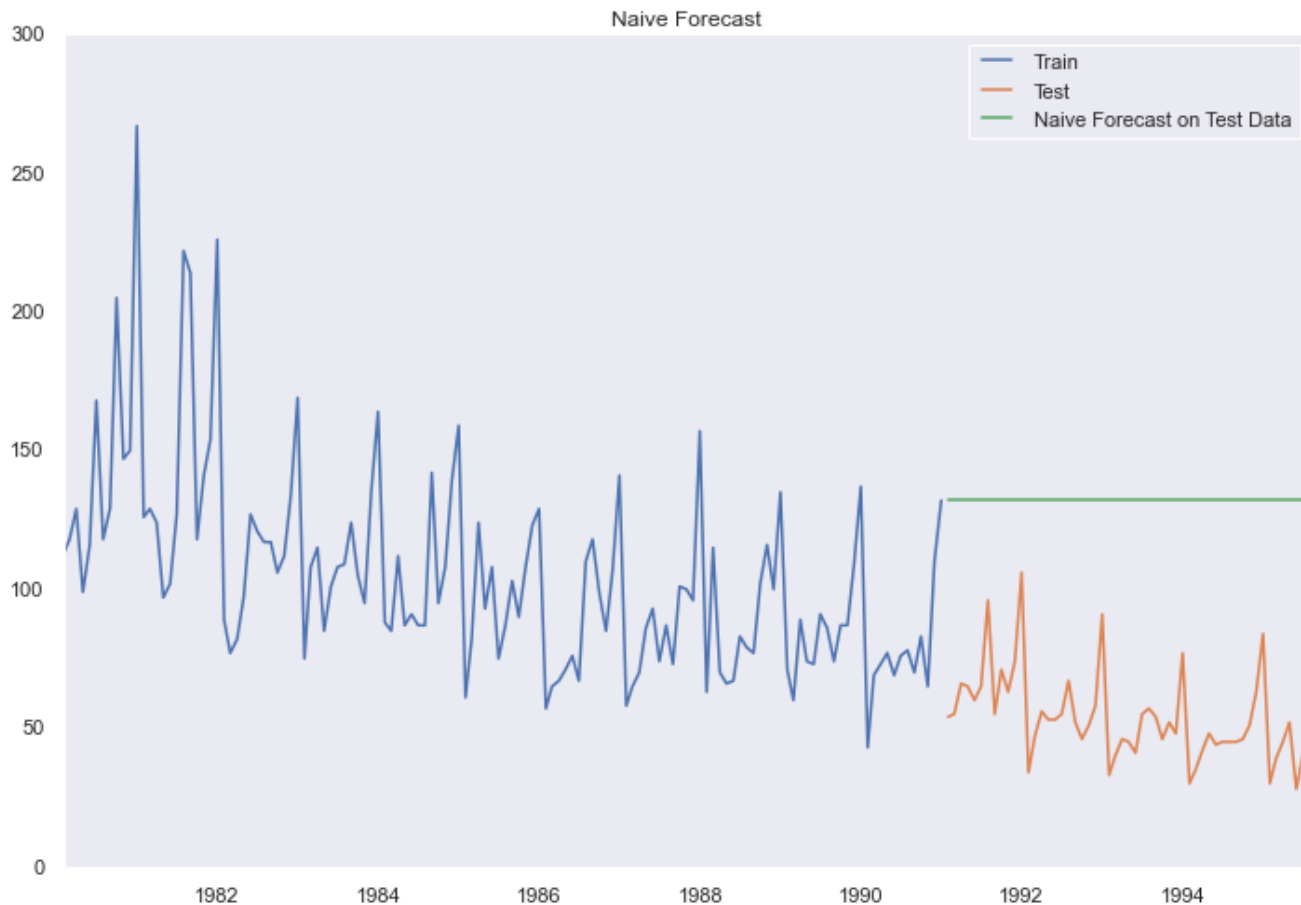


Fig : 11 Prediction on Test Dataset of Navie Forecast Model

Observation

The last observation of the train data is constructed the forecast for entire test data. That's why we are seeing a straight line here. i.e. The Rose Wine Sales will be like the recent past.

Model Evaluation Navie Forecast

Test Data - **RMSE of Navie Forecast Model is - 79.739**

Model 3 : Simple Average Model

Note

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

Building Simple Average Model

```
SimpleAverage_test['mean_forecast']= train['Rose Wine Sales'].mean()
```

Prediction on Test Dataset

Rose Wine Sales		mean_forecast
Time_Stamp		
1991-01-31	54.0	104.939394
1991-02-28	55.0	104.939394
1991-03-31	66.0	104.939394
1991-04-30	65.0	104.939394
1991-05-31	60.0	104.939394

Tab:19 Prediction on Test Data of Simple Average Model

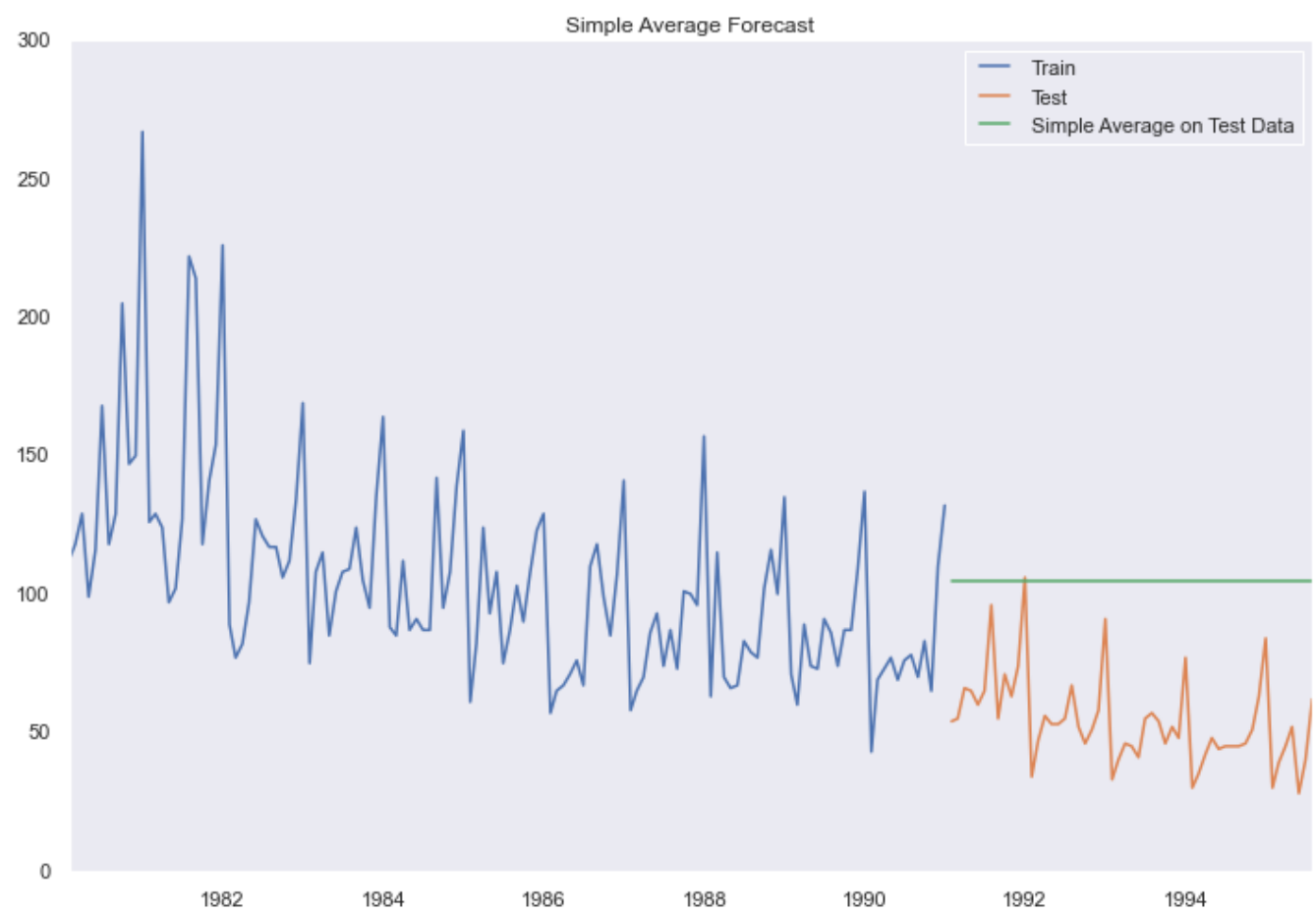


Fig : 12 Prediction on Test Dataset of Simple Average Model

Model Evaluation Simple Average

Test Data - **RMSE of Simple Average Model** is - 53.481

Model 4 : Moving Average(MA)

Note

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.

Building Moving Average Model

```
MovingAverage['Trailing_2']= MovingAverage['Rose Wine Sales'].rolling(2).mean()  
MovingAverage['Trailing_4']= MovingAverage['Rose Wine Sales'].rolling(4).mean()  
MovingAverage['Trailing_6']= MovingAverage['Rose Wine Sales'].rolling(6).mean()  
MovingAverage['Trailing_9']= MovingAverage['Rose Wine Sales'].rolling(9).mean()  
  
MovingAverage.head()
```

	Rose Wine Sales	Trailing_2	Trailing_4	Trailing_6	Trailing_9
Time_Stamp					
1980-01-31	112.0	NaN	NaN	NaN	NaN
1980-02-29	118.0	115.0	NaN	NaN	NaN
1980-03-31	129.0	123.5	NaN	NaN	NaN
1980-04-30	99.0	114.0	114.5	NaN	NaN
1980-05-31	116.0	107.5	115.5	NaN	NaN

Tab:20 Records of Dataset with Rolling Mean

Plotting of the Whole Data With Moving Average

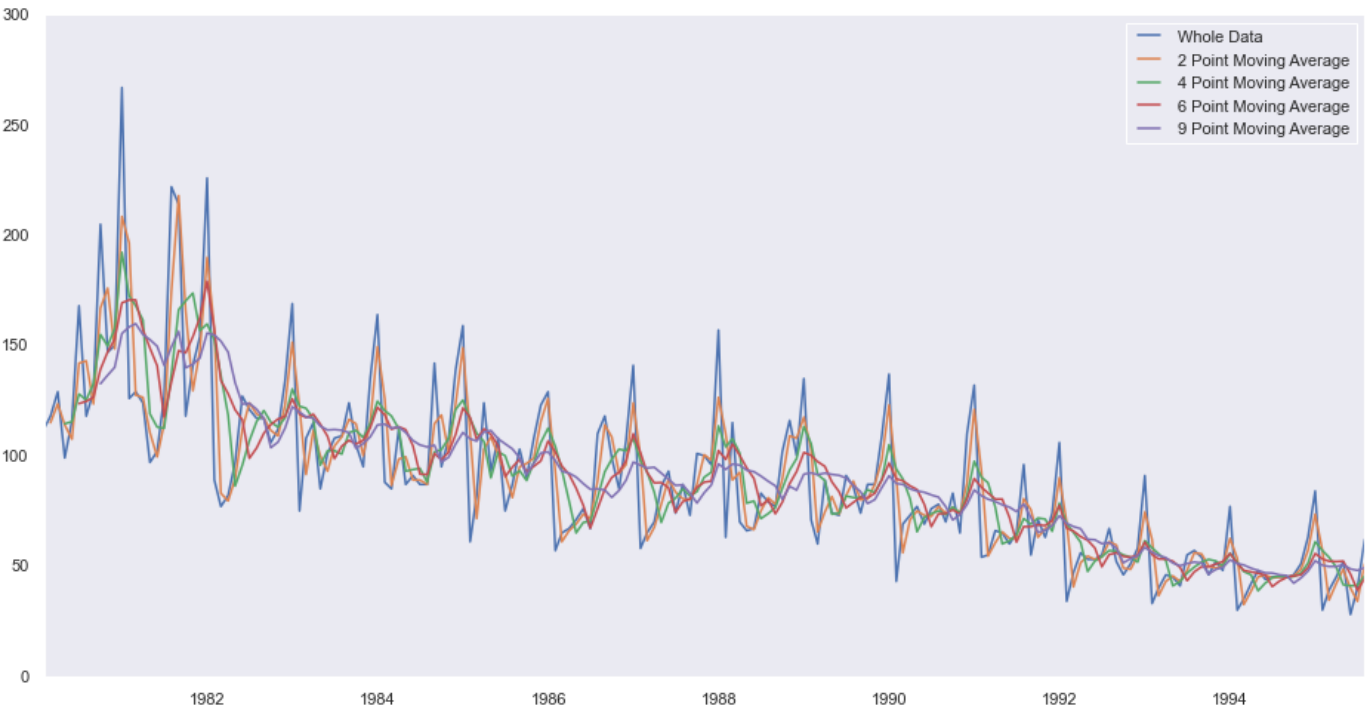


Fig : 13 Whole Data With Moving Average

Observation

2 Point Moving Average curve is copying the original data as it is replicating the original data.

Plotting Moving Average on both the Training and Test data

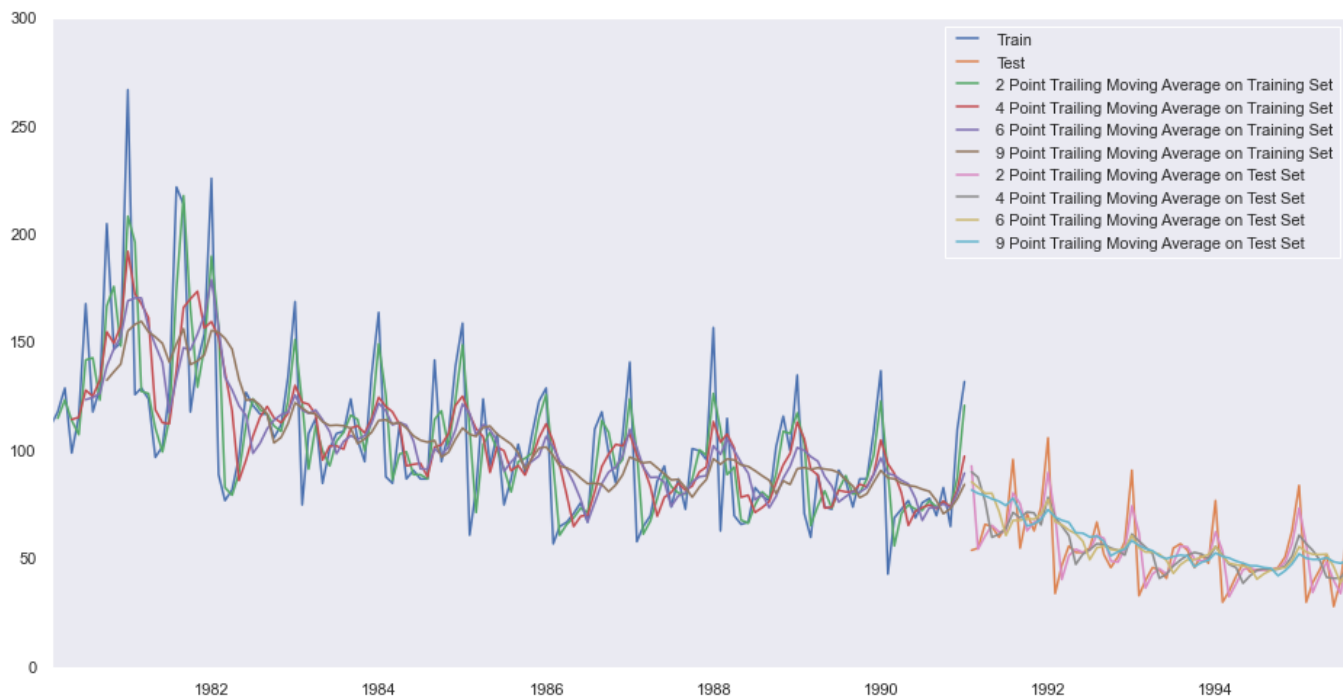


Fig : 14 Moving Average on both the Training and Test data

Test Data - RMSE With Moving Average Model

- For 2 point Moving Average Model forecast on the Test Data, RMSE is 11.529
- For 4 point Moving Average Model forecast on the Test Data, RMSE is 14.455
- For 6 point Moving Average Model forecast on the Test Data, RMSE is 14.572
- For 9 point Moving Average Model forecast on the Test Data, RMSE is 14.731

Result

RMSE of the 2 Point Moving Average Model is the least - 11.529 among the 4 , 6 , 9 Point Moving Average Model.

Model 5 : Simple Exponential Smoothing Model

Single Exponential Smoothing, SES for short, also called Simple Exponential Smoothing, is a time series forecasting method for univariate data without a trend or seasonality. It requires a single parameter, called alpha (α), also called the smoothing factor or smoothing coefficient.

Building Simple Exponential Smoothing Model- Autofit Method

```
model_SES= SimpleExpSmoothing(SES_train['Rose Wine Sales'])
```

```
model_SES_autofit= model_SES.fit(optimized=True)
```

Checking the Parameter

```
{'smoothing_level': 0.09874989207824814,  
'smoothing_trend': nan,  
'smoothing_seasonal': nan,  
'damping_trend': nan,  
'initial_level': 134.3869755697016,  
'initial_trend': nan,  
'initial_seasons': array([], dtype=float64),  
'use_boxcox': False,  
'lamda': None,  
'remove_bias': False}
```

Insights

Here in the simple exponential smoothing model we get value of alpha = 0.09874989207824814

Prediction on Test Dataset

	Rose Wine Sales	predict
Time_Stamp		
1991-01-31	54.0	87.104999
1991-02-28	55.0	87.104999
1991-03-31	66.0	87.104999
1991-04-30	65.0	87.104999
1991-05-31	60.0	87.104999

Tab:21 Prediction on Test Data of Simple Exponential Model

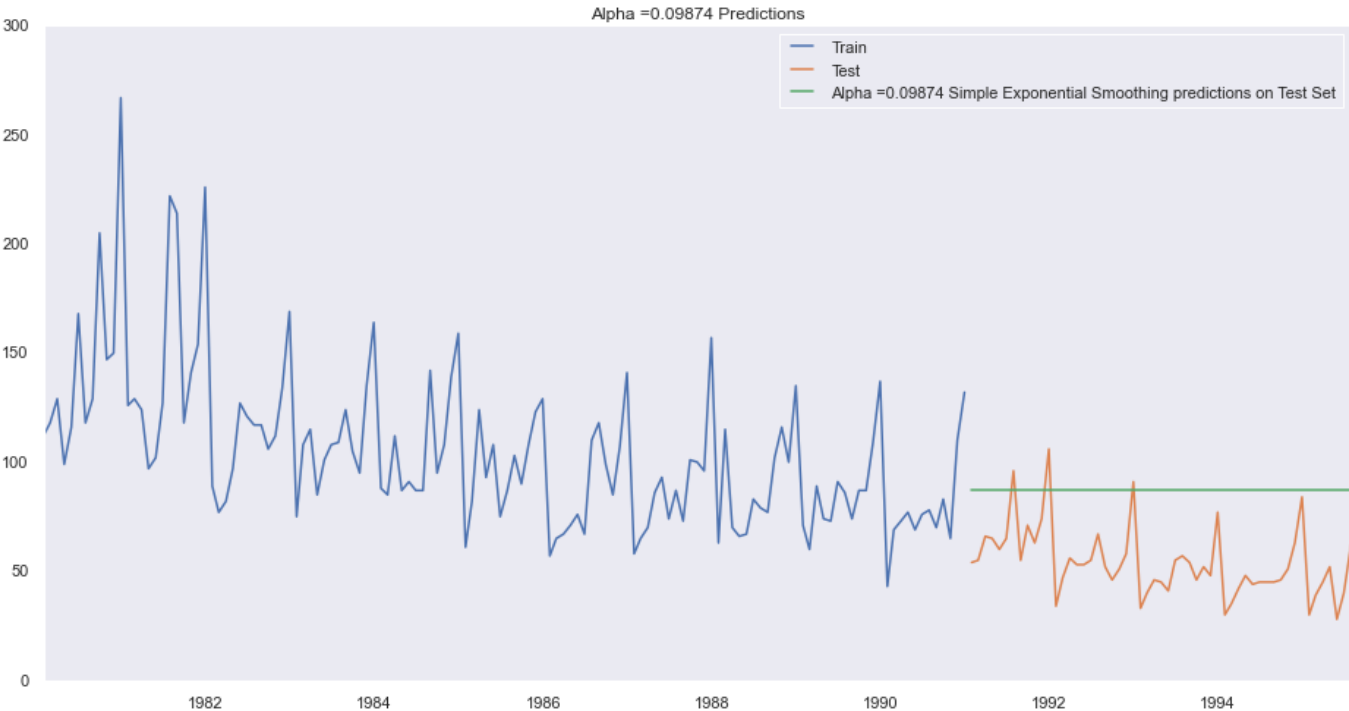


Fig : 15 Prediction on Test Dataset of Simple Exponential Model

Test Data - RMSE with Simple Exponential Smoothing Model

For Alpha =0.09874 Simple Exponential Smoothing Model forecast on the Test Data, RMSE is 36.817

Simple Exponential Smoothing Model by Setting different alpha values - Brute Force Meshod

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

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

	Alpha Values	Train RMSE	Test RMSE
0	0.3	32.470164	47.525251
1	0.4	33.035130	53.787686
2	0.5	33.682839	59.661932
3	0.6	34.441171	64.991324
4	0.7	35.323261	69.718108
5	0.8	36.334596	73.793865
6	0.9	37.482782	77.159094

Tab:22 Alpha Values for Train & Test RMSE Simple Exponential Smoothing Model

Insights

Here after comparing different alpha values we get the least RMSE on Test data at alpha=0.3.So, we can take alpha=0.3 for the model.

We will take alpha = 0.3 to built our simple exponential smoothing model .

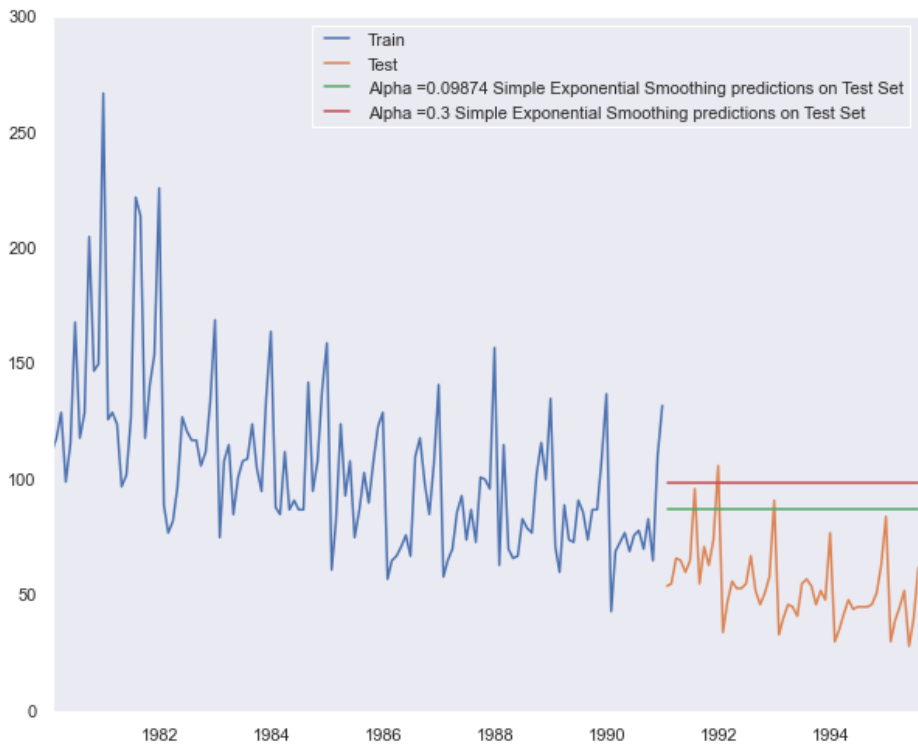


Fig : 16 Prediction on Test Dataset of Simple Exponential Model at alpha = 0.3

Test Data - RMSE with Simple Exponential Smoothing Model with aplha =0.3

For Alpha =0.3 Simple Exponential Smoothing Model forecast on the Test Data, RMSE is 47.525251

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

Double Exponential Smoothing model is suitable to model the time series with trend but without seasonality. In the model there are two kinds of smoothed quantities: smoothed signal and smoothed trend. ... The Holt's linear exponential smoothing displays a constant trend indefinitely into the future.

Double exponential smoothing employs a level component and a trend component at each period. Double exponential smoothing uses two weights, (also called smoothing parameters), to update the components at each period.

Note

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

Building Double Exponential Smoothing Model (Holt's Model) by Setting different alpha & beta values -Brute Force Method (For Codes please check code file.)

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

	Alpha Values	Beta Values	Train RMSE	Test RMSE
0	0.3	0.3	35.944983	265.591922
8	0.4	0.3	36.749123	339.330850
1	0.3	0.4	37.393239	358.775361
16	0.5	0.3	37.433314	394.296935
24	0.6	0.3	38.348984	439.320331

Tab:23 Alpha & Beta Values for Train & Test RMSE Double Exponential Smoothing Model

Insights

Here after comparing different alpha and beta values we get the least RMSE on Test data at alpha=0.3. and beta =0.3 So, we can take alpha=0.3 and beta = 0.3 for the model.

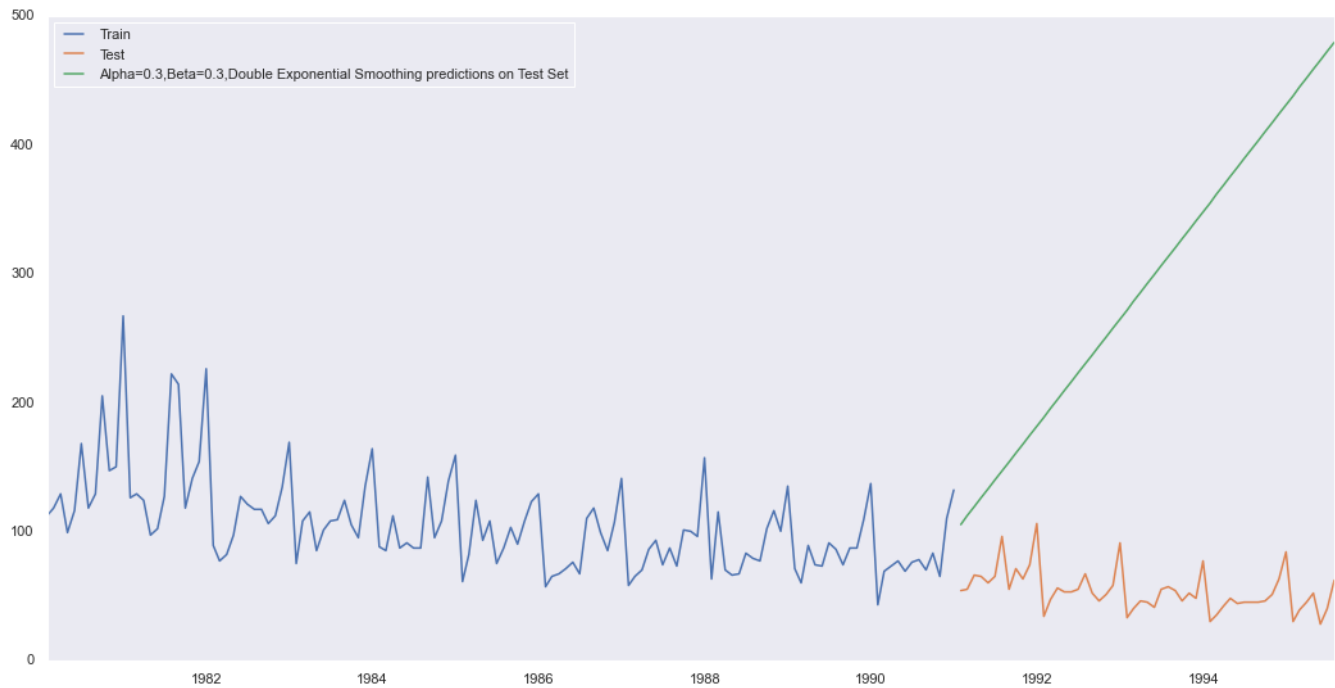


Fig : 17 Prediction on Test Dataset of Double Exponential Model at alpha = 0.3 and beta = 0.3

Test Data - RMSE with Double Exponential Smoothing Model with alpha = 0.3 and beta = 0.3

For Alpha = 0.3 and Beta = 0.3 Double Exponential Smoothing Model forecast on the Test Data, RMSE is 265.591922

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

Triple exponential smoothing is used to handle the time series data containing a seasonal component. This method is based on three smoothing equations: stationary component, trend, and seasonal. Both seasonal and trend can be additive or multiplicative. ... Seasonal change smoothing factor.

Note

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

Building Triple Exponential Smoothing Model (Holt's Winter Model) - Autofit Method

```
model_TES = ExponentialSmoothing(TES_train['Rose Wine Sales'], trend='additive', seasonal='multiplicative', freq='M')
```

```
model_TES_autofit = model_TES.fit()
```

Checking the Parameter

```
{'smoothing_level': 0.10053559166352141,  
'smoothing_trend': 2.9667113597565187e-06,  
'smoothing_seasonal': 3.752823638129799e-07,  
'damping_trend': nan,  
'initial_level': 49.81115356815166,  
'initial_trend': -0.19155645429914447,  
'initial_seasons': array([2.19563917, 2.47993924, 2.71080105, 2.37691247, 2.66805867,  
      2.8714592 , 3.16165653, 3.38100476, 3.16286393, 3.11258111,  
      3.61951388, 4.95479995]),  
'use_boxcox': False,  
'lamda': None,  
'remove_bias': False}
```

Insights

Here in the Triple Exponential Smoothing model we get value of alpha = 0.10053559166352141 , beta = 2.9667113597565187e-06, gamma = 3.752823638129799e-07

Prediction on Test Dataset

Rose Wine Sales		auto_predict
Time_Stamp		
1991-01-31	54.0	54.078681
1991-02-28	55.0	60.605971
1991-03-31	66.0	65.728614
1991-04-30	65.0	57.177535
1991-05-31	60.0	63.670076

Tab:24 Prediction on Test Data of Triple Exponential Model

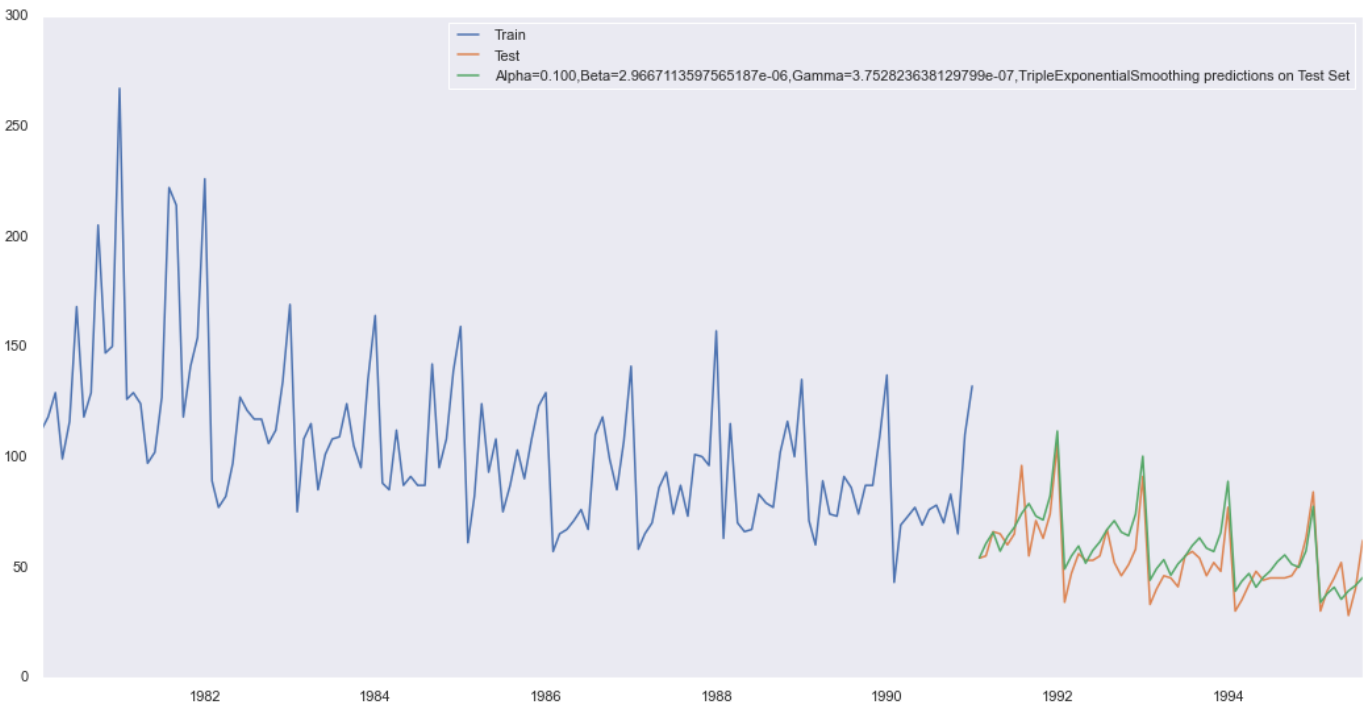


Fig : 18 Prediction on Test Dataset of Triple Exponential Model at alpha = 0.10053559166352141 , beta = 2.9667113597565187e-06, gamma = 3.752823638129799e-07

Test Data - RMSE with Triple Exponential Smoothing Model with alpha = 0.10053559166352141 , beta = 2.9667113597565187e-06, gamma = 3.752823638129799e-07

For alpha = 0.10053559166352141 , beta = 2.9667113597565187e-06, gamma = 3.752823638129799e-07.Triple Exponential Smoothing Model forecast on the Test Data, RMSE is 9.790.

Triple Exponential Smoothing Model by Setting different alpha , beta and gamma values - Brute Force Meshod

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

	Alpha Values	Beta Values	Gamma Values	Train RMSE	Test RMSE
8	0.3	0.4	0.3	28.111886	10.951007
1	0.3	0.3	0.4	27.399095	11.185630
69	0.4	0.3	0.8	32.601491	12.613096
16	0.3	0.5	0.3	29.087520	14.395909
131	0.5	0.3	0.6	32.144773	16.703934

Tab:25 Alpha ,Beta and Gamma Values for Train & Test RMSE Triple Exponential Smoothing Model

Insights

Here after comparing different alpha , beta and gamma values we get the least RMSE on Test data at alpha=0.3 , beta = 0.4 and gamma = 0.3. So, we can take alpha=0.3, beta = 0.3 and gamma = 0.3 for the model.

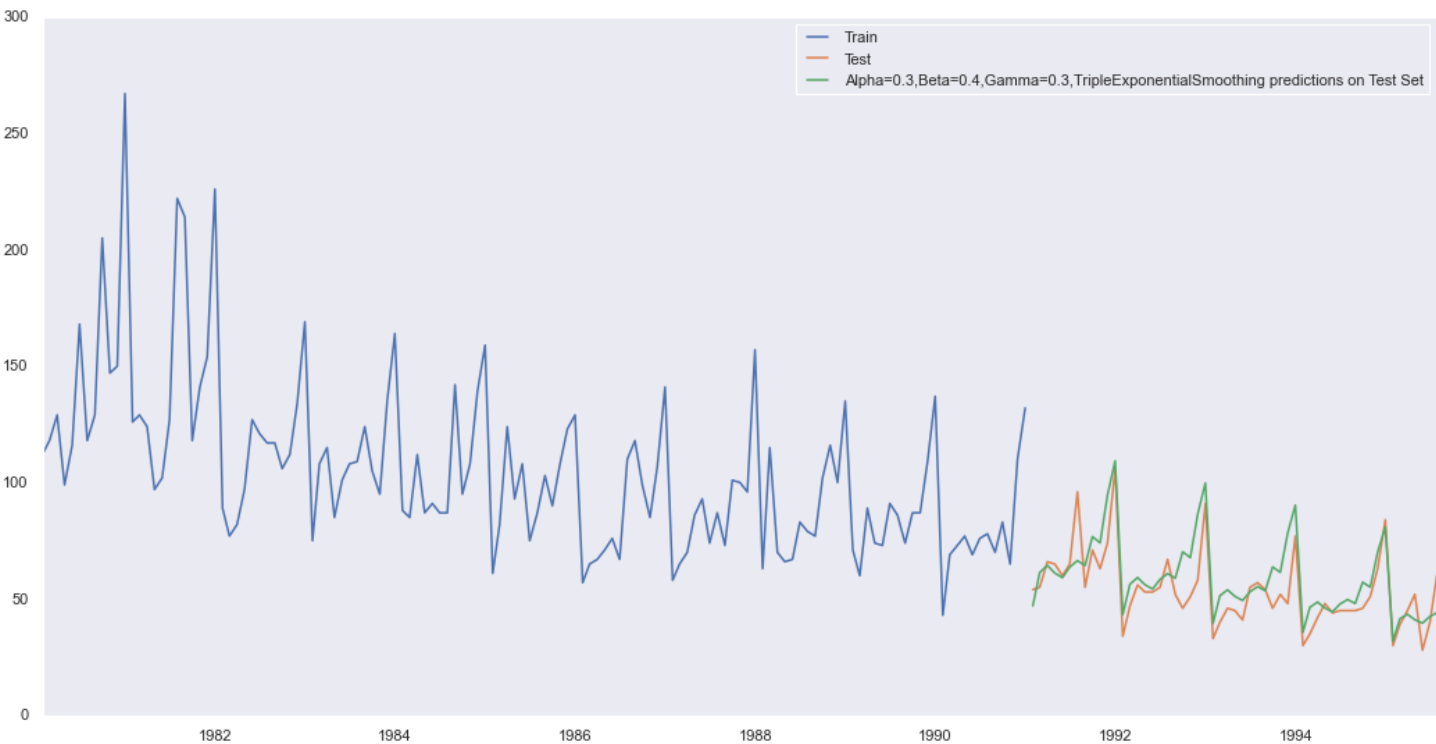


Fig : 19 Prediction on Test Dataset of Triple Exponential Model at alpha =0.3 beta = 0.4 and gamma = 0.3

**Test Data - RMSE with Triple Exponential Smoothing Model with $\alpha = 0.3$
 $\beta = 0.4$ and $\gamma = 0.3$.**

For $\alpha = 0.3$ $\beta = 0.4$ and $\gamma = 0.3$. Triple Exponential Smoothing Model forecast on the Test Data, RMSE is 10.951007

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 $\alpha = 0.05$.

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

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

- H0**
: The Time Series has a unit root and is thus non-stationary.
- H1**
: The Time Series does not have a unit root and is thus stationary.

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

Check for stationarity of the whole Time Series data. - Dickey-Fuller test

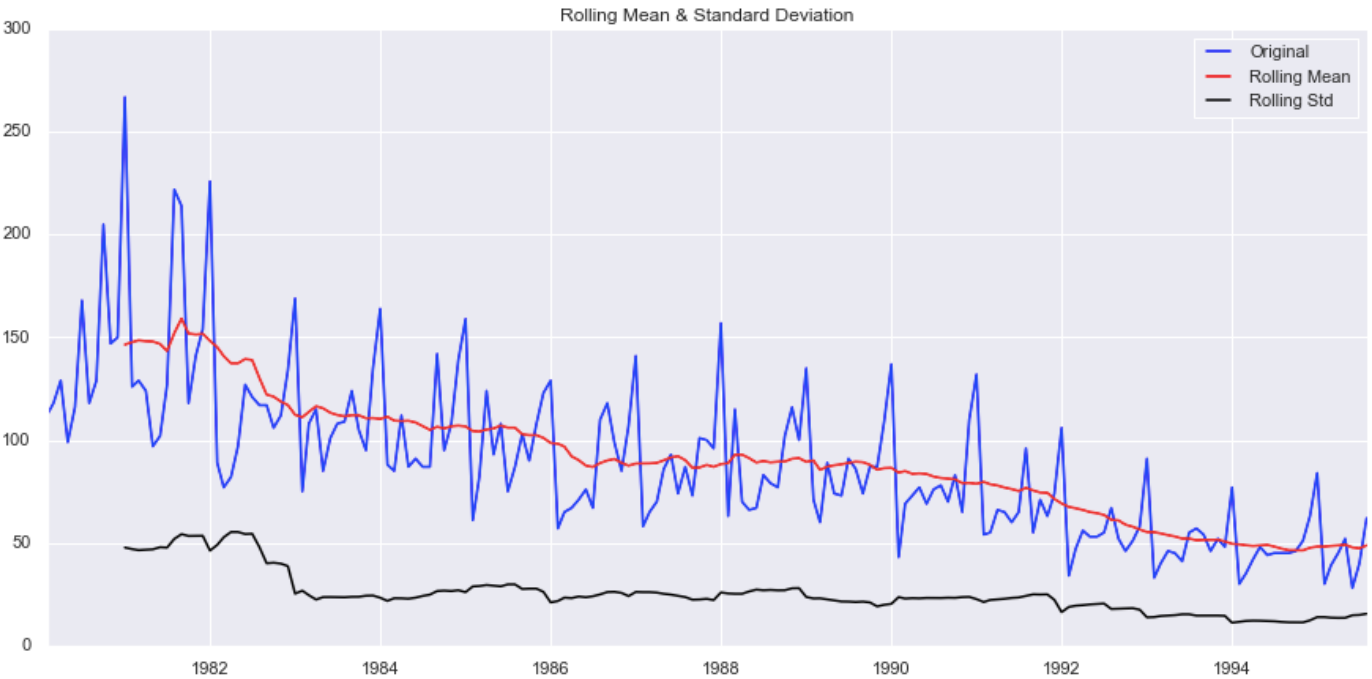


Fig : 20 Check for stationarity of the whole Time Series data. - Dickey-Fuller test

Results of Dickey-Fuller Test:

Test Statistic	-1.874856
p-value	0.343981
#Lags Used	13.000000
Number of Observations Used	173.000000
Critical Value (1%)	-3.468726
Critical Value (5%)	-2.878396
Critical Value (10%)	-2.575756
dtype: float64	

Conclusion :

On comparing the p-value , we found p-value is greater than the 5% significant level ,hence we fail to reject the null hypothesis & reached on the conclusion that he Time Series is non-stationary. Let us take a difference of order 1 and check whether the Time Series is stationary or not.

Tab:26 Dickey - Fuller Test Result on WholeTS Data

Let us take a difference of order 1 and check whether the Time Series is stationary or not.

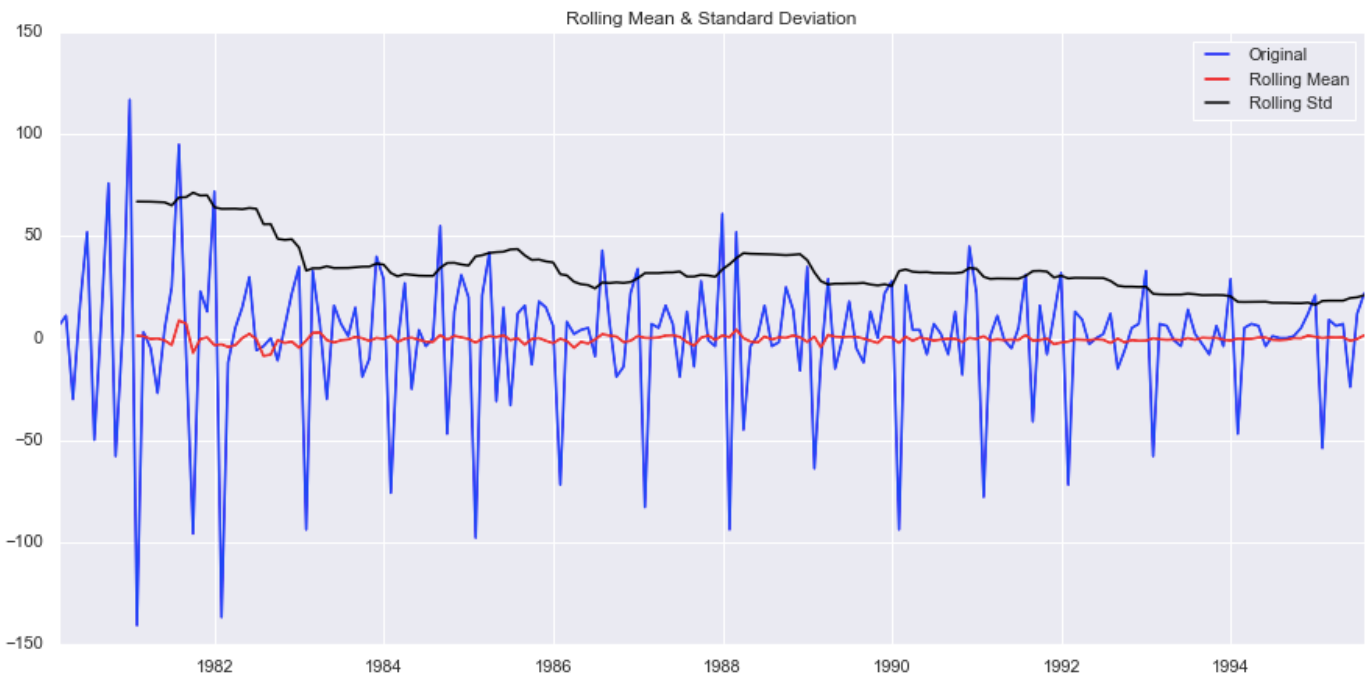


Fig : 21 Check for stationarity after differencing of order 1 on whole Time Series data. - Dickey-Fuller test

Results of Dickey-Fuller Test:

Test Statistic	-8.044139e+00
p-value	1.813580e-12
#Lags Used	1.200000e+01
Number of Observations Used	1.730000e+02
Critical Value (1%)	-3.468726e+00
Critical Value (5%)	-2.878396e+00
Critical Value (10%)	-2.575756e+00
dtype: float64	

Tab:27 Dickey - Fuller Test Result on WholeTS Data with differencing of order 1

Conclusion :

On comparing the p-value , we found p-value is less than the 5% significant level ,hence we reject the null hypothesis & reached on the conclusion that he Time Series is stationary with difference of order 1.We see that at $\alpha = 0.05$ the Time Series is indeed stationary.

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

Automated version of an ARIMA model with the lowest Akaike Information Criteria (AIC).

An ARIMA model is a class of statistical models for analyzing and forecasting time series data. ... ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average. It is a generalization of the simpler AutoRegressive Moving Average and adds the notion of integration

Checking the Stationarity of the Train Data

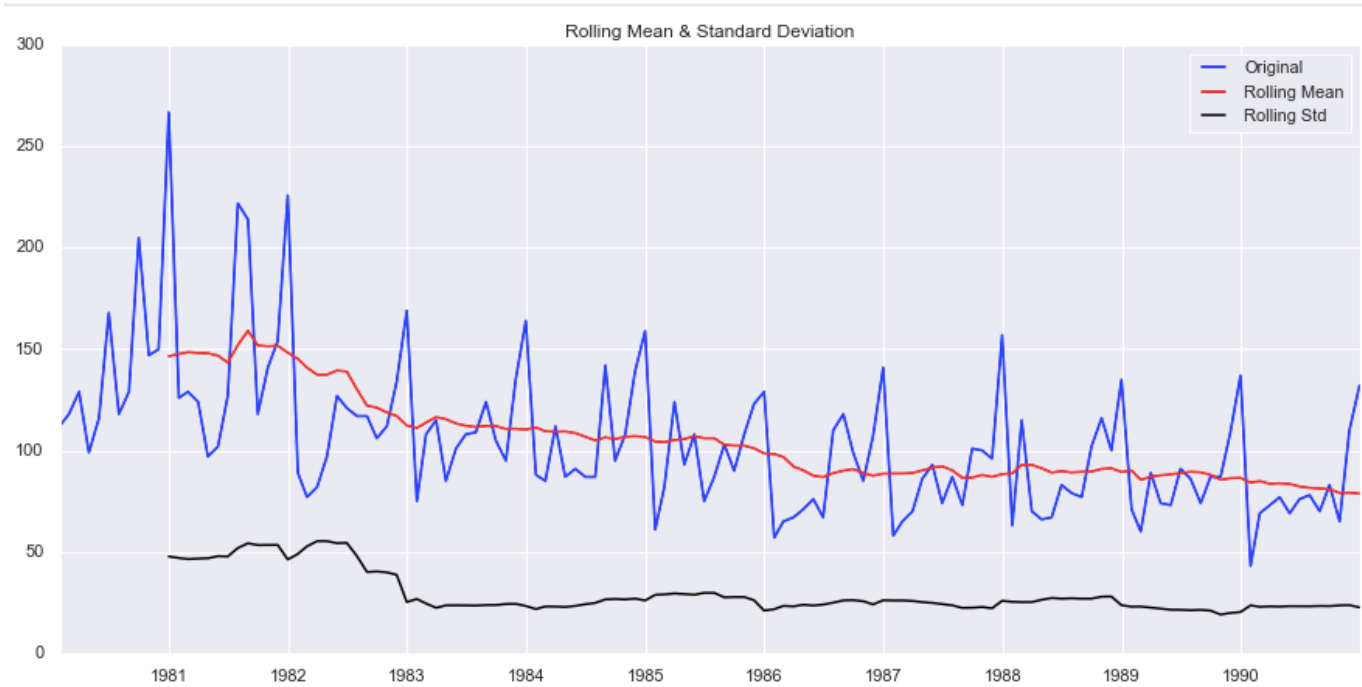


Fig : 22 Check for stationarity of the Train Time Series data. - Dickey-Fuller test

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

Tab:28 Dickey - Fuller Test Result on Train TS Data

Conclusion :

On comparing the p-value , we found p-value is greater than the 5% significant level ,hence we fail to reject the null hypothesis & reached on the conclusion that he Time Series is non-stationary. Let us take a difference of order 1 and check whether the Time Series is stationary or not.

Let us take a difference of order 1 and check whether the Train Time Series is stationary or not.

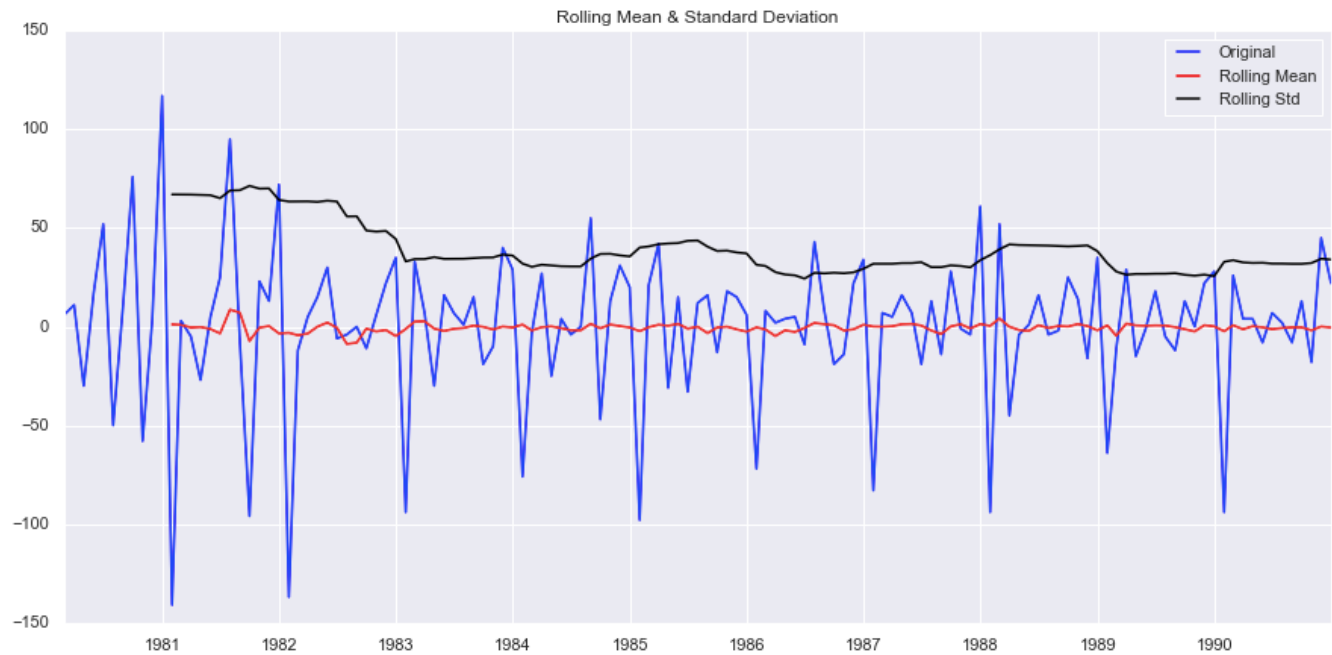


Fig : 23 Check for stationarity after differencing of order 1 on Train Time Series data. - Dickey-Fuller test

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

Tab:29 Dickey - Fuller Test Result on Train TS Data with differencing of order 1

Conclusion :

On comparing the p-value , we found p-value is less than the 5% significant level ,hence we reject the null hypothesis & reached on the conclusion that he Time Series is stationary with difference of order 1.We see that at $\alpha = 0.05$ the Time Series is indeed stationary.

Note

If the series is non-stationary, stationarize the Time Series by taking a difference of the Time Series. Then we can use this particular differenced series to train the ARIMA models. We do not need to worry about stationarity for the Test Data because we are not building any models on the Test Data, we are evaluating our models over there. You can look at other kinds of transformations as part of making the time series stationary like taking logarithms.Here we see that differencing of order 1 makes the series stationary so value of d in ARIMA model will kept as 1 as we need to take a difference of the series to make it stationary.we are running loop which helps us in getting a combination of different parameters of p and q in the range of 0 and 2 for our ARIMA model. Let's check the combinations and will select that combination which have lowest AIC value for ARIMA model.

	param	AIC
2	(0, 1, 2)	1276.835373
5	(1, 1, 2)	1277.359224
4	(1, 1, 1)	1277.775750
7	(2, 1, 1)	1279.045689
8	(2, 1, 2)	1279.298694
1	(0, 1, 1)	1280.726183
6	(2, 1, 0)	1300.609261
3	(1, 1, 0)	1319.348311
0	(0, 1, 0)	1335.152658

Conclusion :

From the Combination and AIC value table we get best combination for ARIMA model which have least AIC , combination of different parameters of p and q are (0,1,2) here , p=0 ,d=1, and q=2 , this combination has lowest AIC of 2210.621575 among all the combinations. So , we can take (0,1,2) combination to built our ARIMA model.

Tab: 30 Combinations & AIC Values for Auto_ARIMA Model

Building Automated version of an ARIMA model with the lowest Akaike Information Criteria (AIC).

```
auto_ARIMA = ARIMA(train['Rose Wine Sales'], order=(0,1,2),freq='M')
results_auto_ARIMA = auto_ARIMA.fit()
```

Auto_ARIMA Model Results

ARIMA Model Results						
Dep. Variable:	D.Rose Wine Sales	No. Observations:	131			
Model:	ARIMA(0, 1, 2)	Log Likelihood	-634.418			
Method:	css-mle	S.D. of innovations	30.167			
Date:	Mon, 20 Dec 2021	AIC	1276.835			
Time:	02:27:24	BIC	1288.336			
Sample:	02-29-1980	HQIC	1281.509			
	- 12-31-1990					
	coef	std err	z	P> z	[0.025	0.975]
const	-0.4885	0.085	-5.742	0.000	-0.655	-0.322
ma.L1.D.Rose Wine Sales	-0.7601	0.101	-7.499	0.000	-0.959	-0.561
ma.L2.D.Rose Wine Sales	-0.2398	0.095	-2.518	0.012	-0.427	-0.053
Roots						
	Real	Imaginary	Modulus	Frequency		
MA.1	1.0001	+0.0000j	1.0001	0.0000		
MA.2	-4.1695	+0.0000j	4.1695	0.5000		

Tab:31 Result Summary of Auto_ARIMA (0,1,2)

Conclusion

By looking the ARIMA model summary we found that the p-values of all the components of the ARIMA model are significant.Hence the value which we get for p,d & q from lowest AIC are significant.

Test Data - RMSE of Auto_ARIMA(0,1,2) Model with $p=0$ $d=1$ and $q=2$ is 15.624861973

Automated version of the SARIMA model with the lowest Akaike Information Criteria (AIC).

A seasonal autoregressive integrated moving average (SARIMA) model is one step different from an ARIMA model based on the concept of seasonal trends. SARIMA accepts an additional set of parameters $(P,D,Q)_m$ that specifically describe the seasonal components of the model. Here P , D and Q represent the seasonal regression, differencing and moving average coefficients, and m represents the number of data points (rows) in each seasonal cycle. Seasonal Autoregressive Integrated Moving Average, SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component.

Plotting ACF plot to understand the seasonal parameter for the SARIMA model.

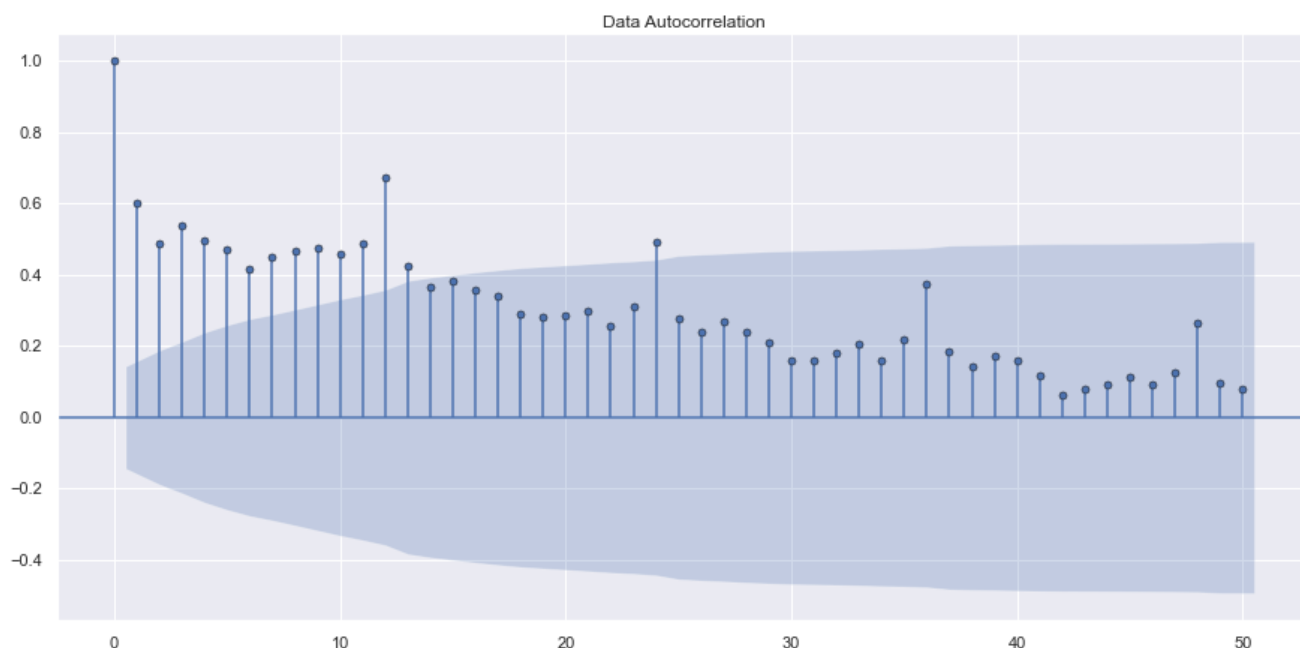


Fig : 24 ACF plot of the original Data

Conclusion

By looking the ACF plot we found a pattern that every 12th lag is significant and repeating itself in the same pattern so building a SARIMA model we take seasonality of 12 and built the model. We will run our auto SARIMA models by setting seasonality of 12.

Checking for the combination of different parameters of $(p\ d\ q)$ and (P,D,Q) by setting the seasonality as 12 for the auto SARIMA model.

Conclusion :

From the Combination and AIC value table we get best combination for SARIMA model which have least AIC , combination of (p d q) and (P,D,Q) by setting the seasonality as 6 for the auto SARIMA model are (0, 1, 2) (2, 0, 2, 12) which have least AIC of 887.937509. So , we can take (0, 1, 2) (2, 0, 2, 12) combination to built our Auto_SARIMA model.

	param	seasonal	AIC
26	(0, 1, 2)	(2, 0, 2, 12)	887.937509
53	(1, 1, 2)	(2, 0, 2, 12)	889.903852
80	(2, 1, 2)	(2, 0, 2, 12)	890.668798
69	(2, 1, 1)	(2, 0, 0, 12)	896.518161
78	(2, 1, 2)	(2, 0, 0, 12)	897.346444

Tab:32 Combinations & AIC Values for Auto_SRIMA Model With Seasonality as 12

Building Automated version of SARIMA model with the lowest Akaike Information Criteria (AIC).

```
auto_SARIMA_12 = sm.tsa.statespace.SARIMAX(train['Rose Wine Sales'].values,
```

```
    order=(0, 1, 2),
    seasonal_order=(2, 0, 2, 12),
    enforce_stationarity=False,
    enforce_invertibility=False)
```

```
results_auto_SARIMA_12 = auto_SARIMA_12.fit(maxiter=1000)
```

Auto_SARIMA Model Results

```

=====
SARIMAX Results
=====
Dep. Variable:                y      No. Observations:      132
Model:                SARIMAX(0, 1, 2)x(2, 0, 2, 12)      Log Likelihood      -436.969
Date:                Mon, 20 Dec 2021      AIC      887.938
Time:                02:27:59      BIC      906.448
Sample:                0      HQIC      895.437
                    - 132
Covariance Type:                opg
=====
              coef      std err          z      P>|z|      [0.025      0.975]
-----
ma.L1          -0.8427      189.960      -0.004      0.996     -373.157      371.472
ma.L2          -0.1573      29.844      -0.005      0.996     -58.650      58.335
ar.S.L12         0.3467       0.079       4.375      0.000       0.191       0.502
ar.S.L24         0.3023       0.076       3.996      0.000       0.154       0.451
ma.S.L12         0.0767       0.133       0.577      0.564      -0.184       0.337
ma.S.L24        -0.0726       0.146      -0.498      0.618      -0.358       0.213
sigma2         251.3137     4.77e+04       0.005      0.996    -9.33e+04     9.38e+04
=====
Ljung-Box (L1) (Q):                0.10      Jarque-Bera (JB):                2.33
Prob(Q):                0.75      Prob(JB):                0.31
Heteroskedasticity (H):            0.88      Skew:                0.37
Prob(H) (two-sided):            0.70      Kurtosis:              3.03
=====

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

Tab:33 Result Summary of Auto_SARIMA(0, 1, 2)(2, 0, 2, 12)

Conclusion

By looking the SARIMA model summary we found that the coff. for all the components and p-values of the components like - ma.L1 ,ma.L2 , ma.S.L12 and ma.S.L24 of the SARIMA model are more than 0.05 so these are insignificant values.

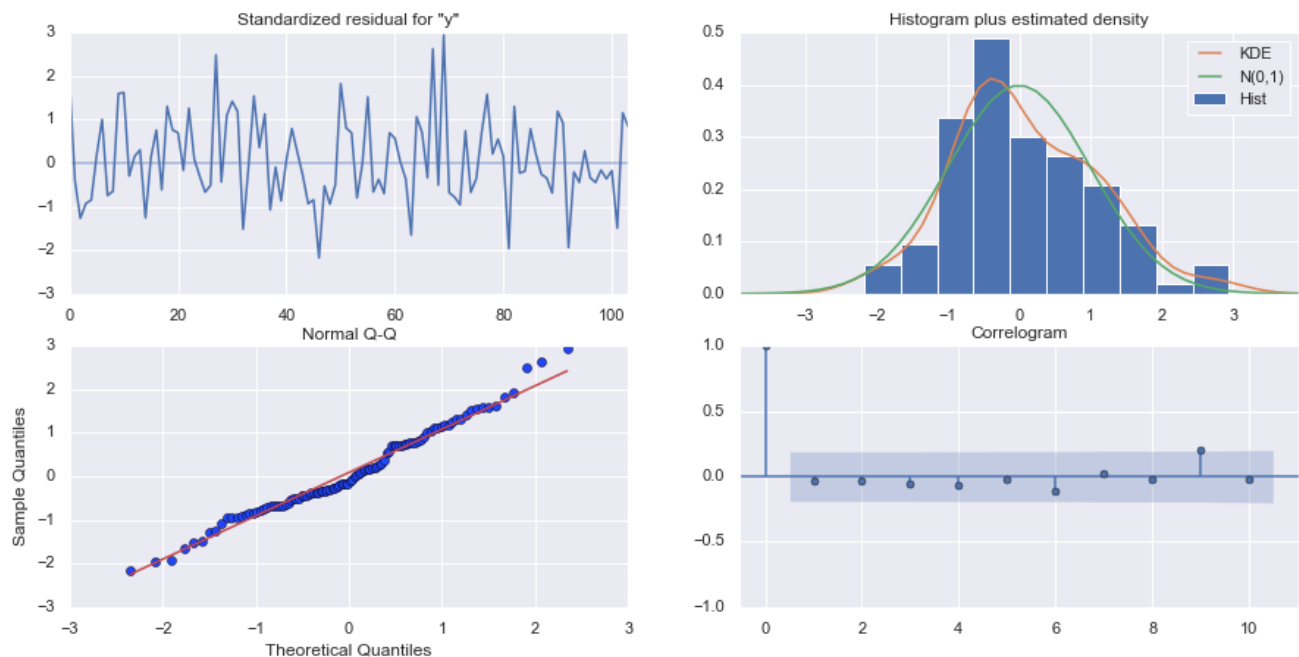


Fig : 25 Diagnostics Plot of Auto_SARIMA (0, 1, 2)(2, 0, 2, 12)

Insights

- Form Standardize residual for "Y" we infer that is no pattern found in the error part.
- By looking at the diagnostics plots like (Histogram and Normal Q-Q) we infer that errors are very slightly left skew distributed.

Summary Frame with alpha =0.05

y	mean	mean_se	mean_ci_lower	mean_ci_upper
0	62.867264	15.928501	31.647976	94.086553
1	70.541190	16.147659	38.892360	102.190020
2	77.356411	16.147657	45.707585	109.005236
3	76.208814	16.147657	44.559988	107.857639
4	72.747398	16.147657	41.098572	104.396224

Tab:34 SummaryFrame of Auto _SARIMA(0, 1, 2)(2, 0, 2, 12) at alpha = 0.05

Observation:

From the above table at 95 % of confidence interval we have the mean value , mean_std , and lower limit & upper limit for forecast of Auto _SARIMA(0, 1, 2)(2, 0, 2, 12) at alpha = 0.05

Test Data - RMSE of Auto_SARIMA(0, 1, 2)(2, 0, 2, 12) Model is 26.949019122116677

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 model based on the cut-off points of ACF and PACF.

ACF and PACF plots to get the values for p and q.

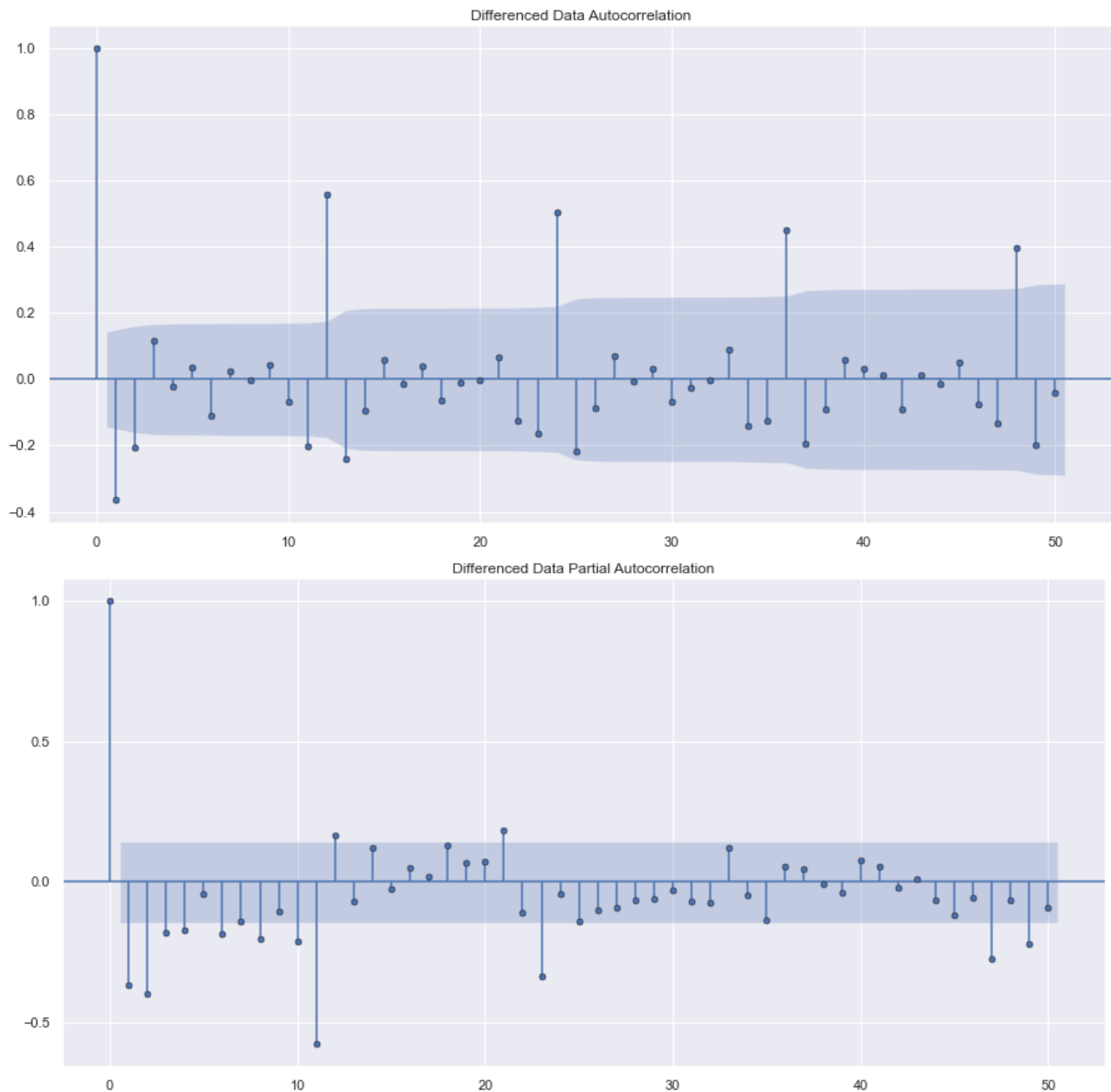


Fig : 26 ACF / PACF Plot for ARIMA Model

Insights

- 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 2.
- The value for $d = 1$, as we see differencing of order 1 makes the series stationary .

Building ARIMA model based on the cut-off points of ACF and PACF.

```
manual_ARIMA = ARIMA(train['Rose Wine Sales'].astype('float64'), order=(4,1,2),freq='M')
```

```
results_manual_ARIMA = manual_ARIMA.fit()
```

Manual_ARIMA Model Results

ARIMA Model Results						
Dep. Variable:	D.Rose Wine Sales	No. Observations:	131			
Model:	ARIMA(4, 1, 2)	Log Likelihood	-633.876			
Method:	css-mle	S.D. of innovations	29.793			
Date:	Mon, 20 Dec 2021	AIC	1283.753			
Time:	02:28:00	BIC	1306.754			
Sample:	02-29-1980	HQIC	1293.099			
	- 12-31-1990					
	coef	std err	z	P> z	[0.025	0.975]
const	-0.1905	0.576	-0.331	0.741	-1.319	0.938
ar.L1.D.Rose Wine Sales	1.1685	0.087	13.391	0.000	0.997	1.340
ar.L2.D.Rose Wine Sales	-0.3562	0.132	-2.693	0.007	-0.616	-0.097
ar.L3.D.Rose Wine Sales	0.1855	0.132	1.402	0.161	-0.074	0.445
ar.L4.D.Rose Wine Sales	-0.2227	0.091	-2.443	0.015	-0.401	-0.044
ma.L1.D.Rose Wine Sales	-1.9506	nan	nan	nan	nan	nan
ma.L2.D.Rose Wine Sales	1.0000	nan	nan	nan	nan	nan
Roots						
	Real	Imaginary	Modulus	Frequency		
AR.1	1.1027	-0.4116j	1.1770	-0.0569		
AR.2	1.1027	+0.4116j	1.1770	0.0569		
AR.3	-0.6863	-1.6643j	1.8002	-0.3122		
AR.4	-0.6863	+1.6643j	1.8002	0.3122		
MA.1	0.9753	-0.2209j	1.0000	-0.0355		
MA.2	0.9753	+0.2209j	1.0000	0.0355		

Tab:35 Result Manual_ARIMA (4,1,2) Model

Conclusion

By looking the Manual_ARIMA model summary we found that the coff. for all the components and p-values of the component ar.L3.D.Rose Wine Sales of the SARIMA model have p-value more than 0.05, it is insignificant.

Test Data - RMSE of Manual_ARIMA (4,1,2) Model with p =4 d = 1 and q=2 is 33.96947074765847

Manual SARIMA model based on the cut-off points of ACF and PACF.

From the original sparkling wine sales time series we see that there is a trend and a seasonality. So, now we take a seasonal differencing and check the series. Here we take diff order of 12 because we have seasonality of that order.

Checking the Stationarity on train data by taking the seasonal differencing of order 12- Dickey-Fuller Test

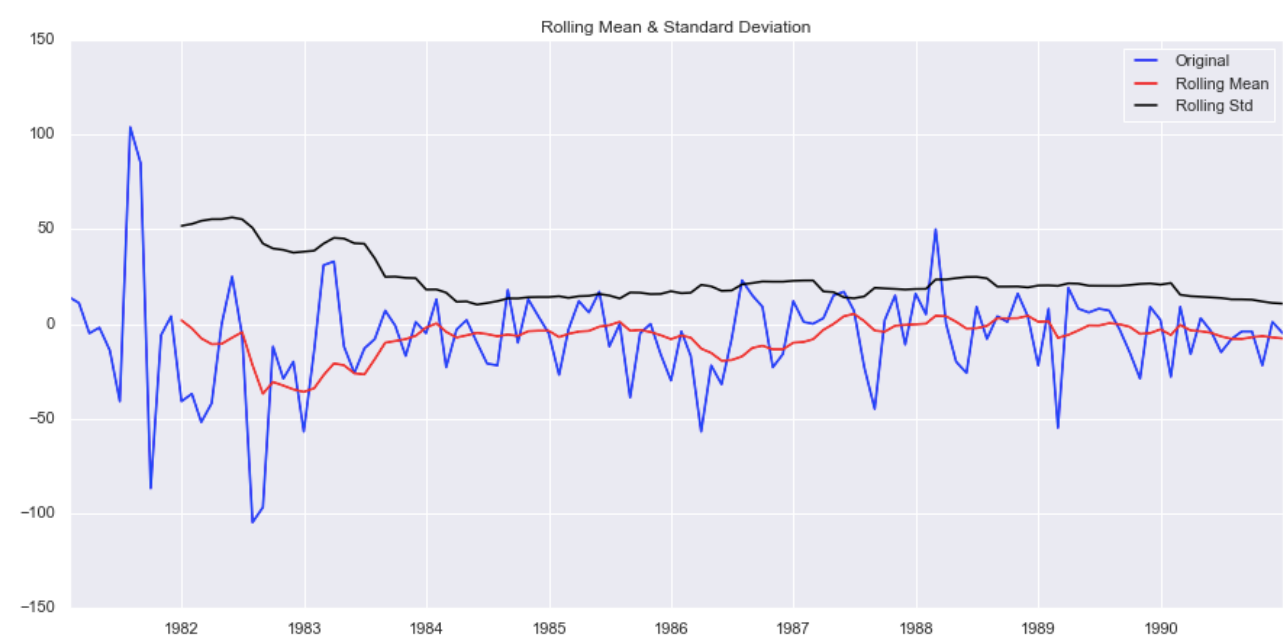


Fig : 27 Checking the Stationarity by taking the seasonal differencing of order 12- Dickey-Fuller Test

```
Results of Dickey-Fuller Test:
Test Statistic      -3.619482
p-value             0.005399
#Lags Used          11.000000
Number of Observations Used 108.000000
Critical Value (1%) -3.492401
Critical Value (5%) -2.888697
Critical Value (10%) -2.581255
dtype: float64
```

Tab: 36 Dickey - Fuller Test Result on Train Data with seasonal diff of order 12

Result

As the looking ate the p-value we conclude that the series is stationary.We donot want further differencing of seasonal differenced series as we found it stationary so we get value of D=0.

Ploting ACF AND PACF Plots with Seasonal Diff to get the Value of P AND Q.

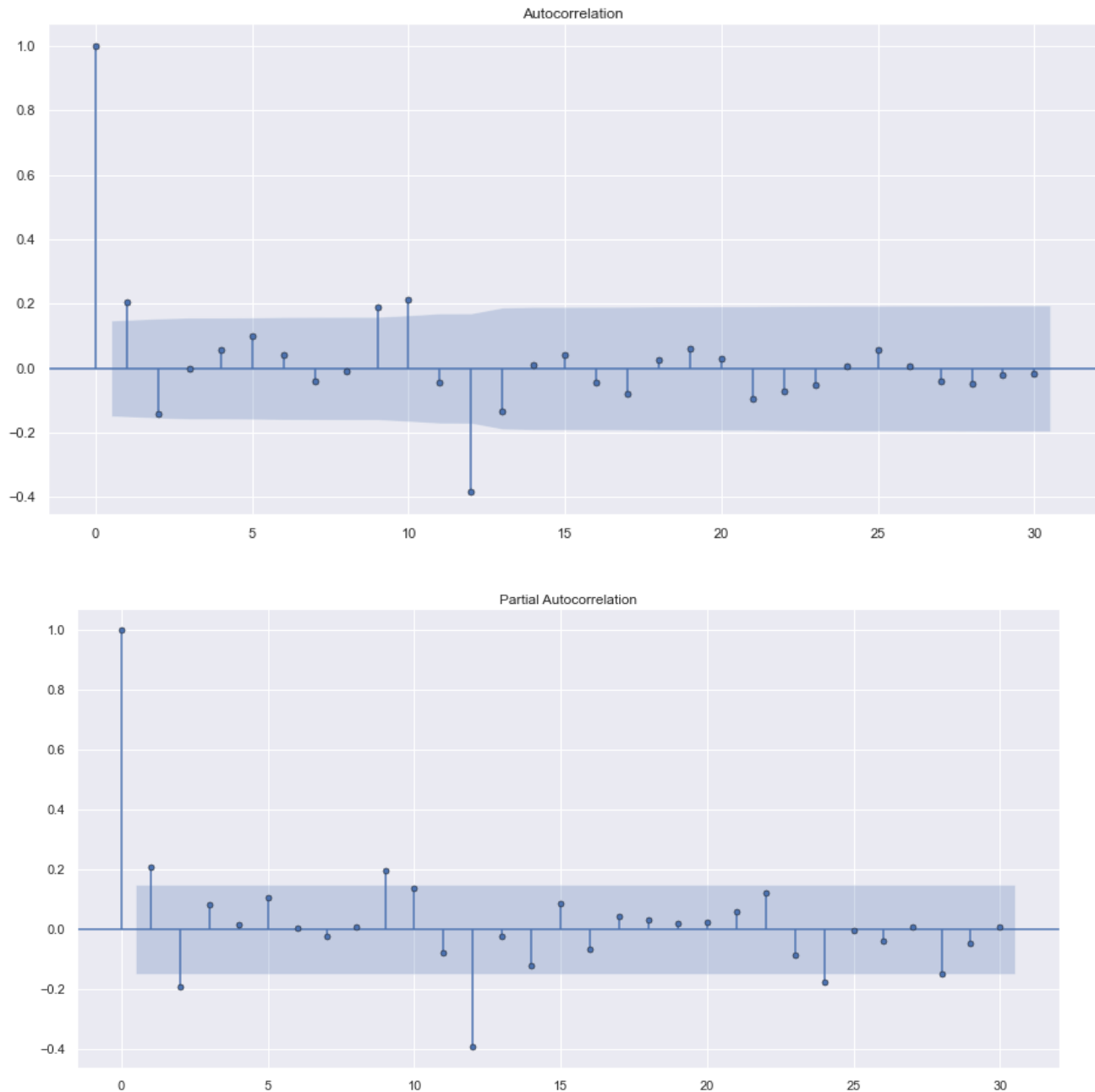


Fig : 28 Seasonal Diff (12) ACF / PACF Plot for SARIMA Model

Observation

Here, we have taken $\alpha=0.05$. We are going to take the seasonal period as 12. We will keep the p , d and q parameters same as the Manual 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. We get value of $P=2$.

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. We get value of $Q=1$.

Building the Manual SARIMA Model at Seasonality 12.

```

manual_SARIMA_12 = sm.tsa.statespace.SARIMAX(train['Rose Wine Sales'].values,
order=(4, 1, 2),
seasonal_order=(2, 0, 1, 12),
enforce_stationarity=False,
enforce_invertibility=False)
results_manual_SARIMA_12 = manual_SARIMA_12.fit(maxiter=1000)

```

Manual_SARIMA (4,1,2)(2,0,1,12)Model Results

```

=====
SARIMAX Results
=====
Dep. Variable:                y      No. Observations:      132
Model:          SARIMAX(4, 1, 2)x(2, 0, [1], 12)      Log Likelihood      -430.931
Date:                Mon, 20 Dec 2021      AIC      881.862
Time:                02:28:03      BIC      908.210
Sample:              0      HQIC      892.534
                  - 132
Covariance Type:          opg
=====
              coef      std err          z      P>|z|      [0.025      0.975]
-----
ar.L1          -0.7465         0.297      -2.518      0.012      -1.328      -0.165
ar.L2           0.0212         0.153       0.138      0.890      -0.279      0.322
ar.L3          -0.0780         0.137      -0.568      0.570      -0.347      0.191
ar.L4          -0.0634         0.085      -0.750      0.453      -0.229      0.102
ma.L1          -0.1243      174.221      -0.001      0.999     -341.591      341.343
ma.L2          -0.8758     152.517      -0.006      0.995     -299.803      298.052
ar.S.L12        0.3365         0.080       4.222      0.000       0.180      0.493
ar.S.L24        0.2796         0.072       3.878      0.000       0.138      0.421
ma.S.L12        0.1347         0.154       0.874      0.382      -0.168      0.437
sigma2         242.4467     4.22e+04       0.006      0.995     -8.25e+04      8.3e+04
=====
Ljung-Box (L1) (Q):              0.05      Jarque-Bera (JB):              3.06
Prob(Q):              0.83      Prob(JB):              0.22
Heteroskedasticity (H):          0.76      Skew:              0.39
Prob(H) (two-sided):          0.42      Kurtosis:              3.30
=====

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

```

Tab: 37 Result Manual_SARIMA (4,1,2) (2,0,1,12) Model**Conclusion**

By looking the Manual_SARIMA model summary we found that the coeff. for all the components and p-values of the component ar.L2 , ar.L3 , ar.L4 , ma.L1, ma.L2, ma.S.L12 components of the Manual_SARIMA model have p-value more than 0.05, it is insignificant.

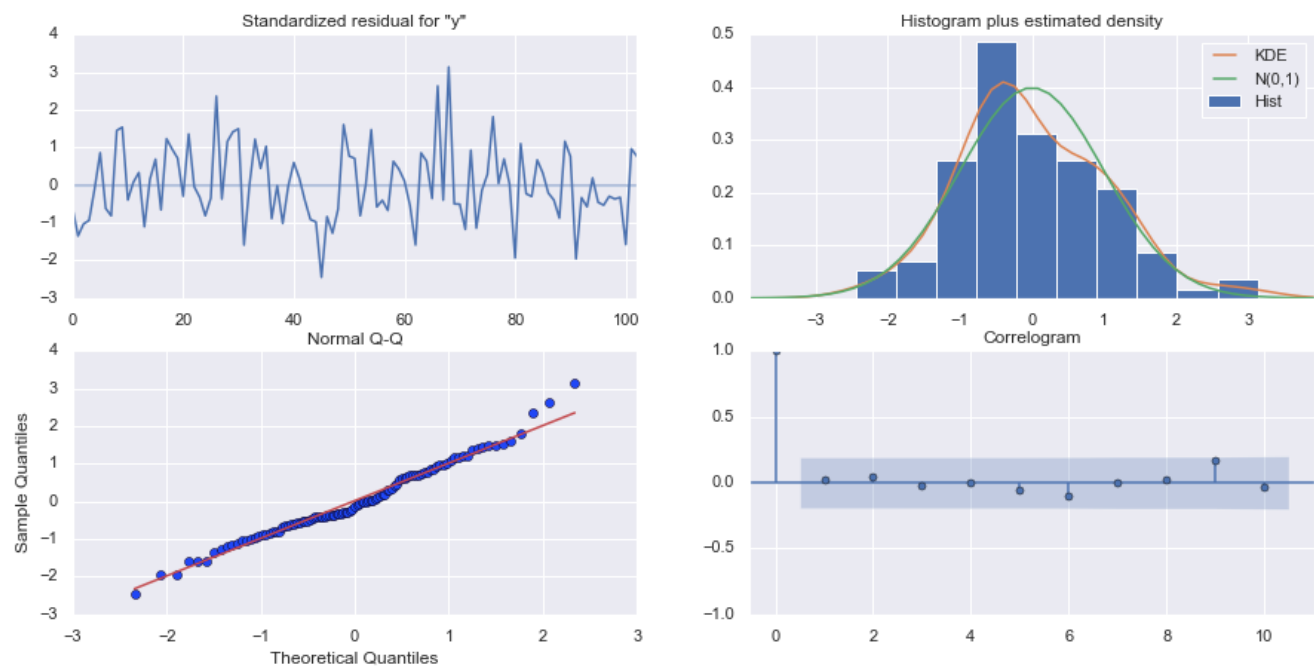


Fig : 29 Diagnostics Plot of Manual_SARIMA (4, 1, 2)(2, 0, 1, 12)

Insights

- Form Standardize residual for "Y" we infer that is no pattern found in the error part.
- By looking at the diagnostics plots like (Histogram and Normal Q-Q) we infer that errors are almost normally distributed.
- From correlogram we infer that correlation coeff. of error terms is zero and we can see that none of the errors are significant.(i.e No error term crossing the cut-off).
- The model Diagnostics Plot of Manual_SARIMA (4, 1, 2)(2, 0, 1, 12) looks fine.

Summary Frame with alpha =0.05

y	mean	mean_se	mean_ci_lower	mean_ci_upper
0	62.030432	15.642372	31.371947	92.688918
1	69.142418	15.790820	38.192979	100.091857
2	76.645851	15.822683	45.633962	107.657740
3	77.846289	15.822648	46.834469	108.858110
4	73.041577	15.842418	41.991008	104.092146

Tab: 38 SummaryFrame of Manual_SARIMA(4,1,2) (2,0,1,12) Model at alpha = 0.0 5

Observation:

From the above table at 95 % of confidence interval we have the mean value , mean_std , and lower limit & upper limit for forecast of Manual_SARIMA(4,1,2) (2,0,1,12) Model at alpha = 0.05

Test Data - RMSE of Manual_SARIMA(3, 1, 2)(0, 0, 0, 12) Model is 28.09706375062283

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
Regression On Time	15.275732
Naive Model	79.738550
Simple Average Model	53.480857
2 Point Trailing Moving Average	11.529409
4 Point Trailing Moving Average	14.455221
6 PointTrailing Moving Average	14.572009
9 PointTrailing Moving Average	14.731209
Alpha=0.09874,Simple Exponential Smoothing	36.816904
Alpha=0.3,Simple Exponential Smoothing	47.525251
Alpha=0.3,Beta=0.3,DoubleExponentialSmoothing	265.591922
Alpha=0.100,Beta=2.966e-06,Gamma=3.7528e-07,TripleExponentialSmoothing	9.790009
Alpha=0.3,Beta=0.4,Gamma=0.3,TripleExponentialSmoothing	10.951007
Auto_ARIMA(0,1,2)	15.624862
Auto_SARIMA(0,1,2)(2,0,2,12)	26.949019
Manual_ARIMA(4,1,2)	33.969471
Manual SARIMA(4,1,2)(2,0,1,12)	28.097064

Tab: 39 Result of All Models With Parameter and Test RMSE

	Test RMSE
Alpha=0.100,Beta=2.966e-06,Gamma=3.7528e-07,TripleExponentialSmoothing	9.790009
Alpha=0.3,Beta=0.4,Gamma=0.3,TripleExponentialSmoothing	10.951007
2 Point Trailing Moving Average	11.529409
4 Point Trailing Moving Average	14.455221
6 PointTrailing Moving Average	14.572009

Tab:40 Result of Top 5 Models With Parameter and Test RMSE

Result

From the above table , we found **TripleExponentialSmoothing Model with Alpha=0.100,Beta=2.966e06,Gamma=3.7528e07**, Model have the least RMSE of 9.790009 on the Test Data. So , we conclude that **TripleExponentialSmoothing Model with Alpha=0.100,Beta=2.966e06,Gamma=3.7528e07**, will be our finalised model on the complete data to predict 12 months into the future.

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.

Building the most optimum model on the Full Data - From the above Test RMSE results we conclude that **TripleExponentialSmoothing Model with Alpha=0.100,Beta=2.966e06,Gamma=3.7528e07** is the best model with least RMSE among all the models.We finalise TripleExponentialSmoothing Model with Alpha=0.100,Beta=2.966e06,Gamma=3.7528e07 is best model to built on full data to predict 12 months into the future with appropriate confidence intervals/bands.

```
fullmodel1 = ExponentialSmoothing(df,
                                trend='additive',
                                seasonal='multiplicative').fit(smoothing_level=0.10053559166352141,
                                                                smoothing_trend=0.0000029667113597565187,
                                                                smoothing_seasonal=0.0000003752823638129799)
```

Full Model Results

ExponentialSmoothing Model Results			
Dep. Variable:	Rose Wine Sales	No. Observations:	187
Model:	ExponentialSmoothing	SSE	48465.592
Optimized:	True	AIC	1071.253
Trend:	Additive	BIC	1122.950
Seasonal:	Multiplicative	AICC	1075.324
Seasonal Periods:	12	Date:	Mon, 20 Dec 2021
Box-Cox:	False	Time:	02:28:04
Box-Cox Coeff.:	None		
	coeff	code	optimized
smoothing_level	0.1005356	alpha	False
smoothing_trend	2.9667e-06	beta	False
smoothing_seasonal	3.7528e-07	gamma	False
initial_level	54.527548	l.0	True
initial_trend	-0.2075527	b.0	True
initial_seasons.0	1.9793311	s.0	True
initial_seasons.1	2.2473756	s.1	True
initial_seasons.2	2.4791514	s.2	True
initial_seasons.3	2.2152075	s.3	True
initial_seasons.4	2.4217390	s.4	True
initial_seasons.5	2.6232273	s.5	True
initial_seasons.6	2.9426181	s.6	True
initial_seasons.7	3.0311218	s.7	True
initial_seasons.8	2.8639735	s.8	True
initial_seasons.9	2.8292914	s.9	True
initial_seasons.10	3.2846493	s.10	True
initial_seasons.11	4.5275808	s.11	True

Tab:41 Result Full Model TripleExponentialSmoothing
Model with Alpha=0.100,Beta=2.966e06,Gamma=3.7528e07

Conclusion

By looking the Full model summary we found that the coff. for all the components and summary give us information about the trend , seasonality and seasonal period of the Rose wine Sales time series.

Calculating the Confidence Bands at 95% confidence level

One assumption that we have made over here while calculating the confidence bands is that the standard deviation of the forecast distribution is almost equal to the residual standard deviation.

	lower_CI	prediction	upper_ci
1995-08-31	15.560545	47.199078	78.837612
1995-09-30	12.363386	44.001920	75.640453
1995-10-31	11.243310	42.881843	74.520377
1995-11-30	17.463155	49.101689	80.740222
1995-12-31	35.103821	66.742355	98.380888

Tab:42 Confidence Interval of Full Model for Forecast

Observation:

From the above table at 95 % of confidence interval we have the mean value , mean_std , and lower limit & upper limit for forecast of Full Model TripleExponentialSmoothing Model with Alpha=0.100,Beta=2.966e06,Gamma=3.7528e07 at alpha = 0.05

RMSE of Full Model TripleExponentialSmoothing Model with Alpha=0.100,Beta=2.966e06,Gamma=3.7528e07 is RMSE: 16.098890944223697

Plot of the forecast on full data along with the confidence band

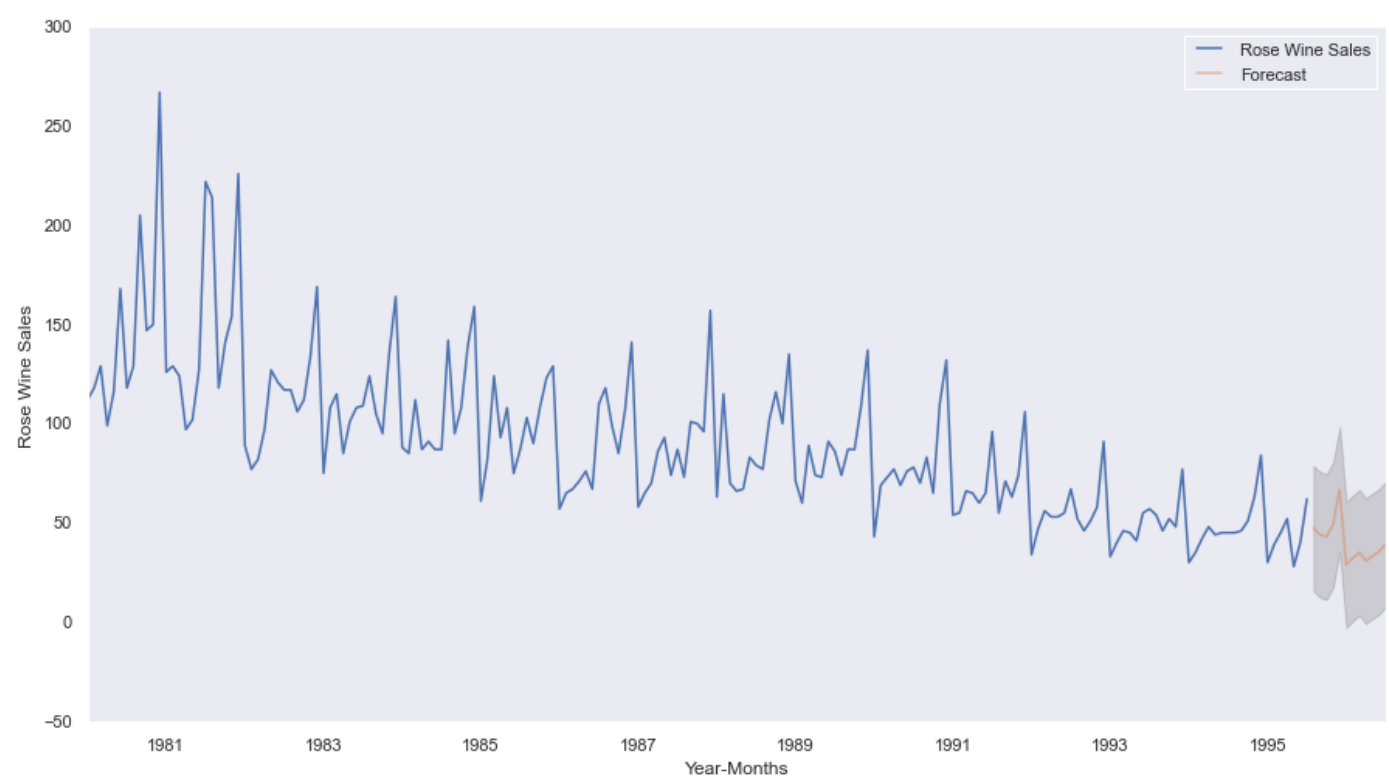


Fig : 30 Plot of the forecast on full data along with the confidence band

Insights :

From the the above plot we infer that with 95% of the confidence level we found that forecast also follows the same pattern as original Rose wine sales series follows.

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

The purpose of this whole exercise is to explore the dataset , analyse and forecast Rose Wine Sales in the 20th century. Here we perform the exploratory data analysis & apply various time series forecasting models like Linear Regression , Navie Forecast , Simple Average , Moving Average and various kind of exponential smoothing models like (Simple , Double , Triple Exponential) and ARIMA / SARIMA models on the Rose Wine Sales dataset and check their RMSE on the test data , model which gives the least RMSE will be the final model for us to analyse and forecast the Rose Wine Sales in the 20th century.

Insights of EDA / Data Visualization and Time Series Forecasting Models :

We can see that there is a downward trend in the series with a seasonal pattern associated as well. Moreover we found that some data is missing from the series too. We will check the missing values and impute them as well by suitable method.

Description of the Original Rose Wine Sales Time Series

- Rose Wine Sales ranges from a minimum of 28 to maximum of 267.
- Mean of the Rose Wine Sales is around 90.394595.
- Standard Deviation of the Rose Wine Sales is 39.175344.
- 25% , 50% (median) and 75 % of Rose Wine Sales are 63 ,86 and 112.

Information about the Original Sparkling Wine Sales Time Series

- From the above results we can see that there is 2 null values present in the dataset.
- There are total 187 entries of Rose wines Sales as per Monthly frequency in this dataset, indexed from 1980-01-31 to 1995-07-31.
- Rose Wine Sales column have d-type of float64.
- Memory used by the dataset: 2.9 KB.

There is 2 Null Values Present in the Dataset. So we need to impute the missing value present in the data with some meaningful value by using suitable null value imputation method. In this problem we impute the null values by forwardfill null value method. We successfully impute the null values by forward fillna method. Now we donot have any null values in the data.

The Rose.csv data set has 187 observations (rows) and 1 variable (column named as Rose Wine Sales) in the dataset.

Year on Year boxplot for the Rose Wine Sales.

- As we got to know from the Time Series plot, the box-plots over here also indicates a measure of trend being present. Also, we see that the Rose Wine Sales have outliers for the years. The yearly boxplots also shows that the Sales have decreased year after year.
- Box-plot of Year 1980 and 1981 have max median value, from this we can clearly infer that year 1981 have maximum Rose Wine Sales.
- Box-plot of Year 1995 have min median value, we can clearly infer that year 1995 have minimum Rose Wine Sales.

Monthly Box-Plot for the Rose Wine Sales Taking all the Years into Account

- The Box-Plots for the monthly Rose Wine Sales for different years very few outliers in the month 6, 7, 8, 9 and 12 show outliers, rest doesn't show any outliers.
- From September to December the Rose Wine Sales increasing, so this the period where the Rose Wine Sales is highest.
- December is the month of highest Rose Wine Sales every year whereas Jan is the month of lowest Rose wine sales.
- There is seasonality also every year from September to December the Rose Wine Sales increasing.

Month-Plot of Rose Wine Sales Time Series

- As noticed in the above box-plot we get same result from here too. From September to December Rose Wine Sales goes on increasing.
- December month have the highest sales of the Rose Wine while Jan month have low sales of the Rose Wine.

Time Series Plot for different months for different years

- This plot gives us information about the monthly trend across the years. Here in this plot every line is a month tells us about the sales of Rose Wines of each month across the year. This is way to show year on year monthly trend.
- From the above plot we clearly infer that December month have highest sales of Rose Wine.
- Jan month have the lowest sales of the Rose Wine.

Here we apply various time series forecasting models like Linear Regression, Navie Forecast, Simple Average, Moving Average and various kind of exponential smoothing models like (Simple, Double, Triple Exponential) and ARIMA / SARIMA models on the Sparkling Wine Sales dataset and check their RMSE on the test data. After comparing the TEST RMSE of all the model that we built. We come to know that the TEST RMSE of **Alpha=0.100, Beta=2.966e-06, Gamma=3.7528e-07, TripleExponentialSmoothingModel** is least among all the models with different parameters. So we take **Alpha=0.100, Beta=2.966e-06, Gamma=3.7528e-07, TripleExponentialSmoothing Model** to build complete data and predict 12 months into the future with appropriate confidence intervals/bands.

RMSE of the Full Model is - 16.098890944223697

From the Plot of the forecast on full data along with the confidence band we infer that with 95% of the confidence level we found that forecast also follows the same pattern as original Rose wine sales series follows going downwards.

Recommendations:

The ABC Estate Wines company should focus on key strengths & develop marketing strategies to promote Rose Wine Sales . As we From Sept to Dec the Rose wine sales are increasing and highest in December but in the month of Jan it was low so wine company can run various offers during this period to boost their rose wine sales as this time is new year weekend so company can give various offers like buy 1 get 1 or some interesting gifts on purchase of rose wine to attract more customers. Such activities can help wine comapny to increase their sales of rose wine in jan month.

