Murali b

TIMESERIES FORECASTING

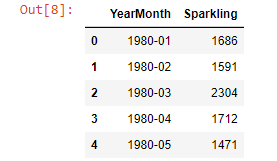
Contents Page Number

1.1 Head of the data 3  
1.2 Tail of the data 3  
1.3 Shape of the data 3  
1.4 Null values 4  
1.5 Adding the timestamp to the dataframe 4  
1.6 Setting the timestamp 5  
1.7 Plotting the graph of the dataset 5  
1.8 Description of the dataset 6  
1.9 Yearly boxplot 6  
1.10 Monthly boxplot 7  
1.11 Monthly sale across the years 8  
1.12 Empirical Cumulative distribution graph 8  
1.13 Average sparkling sales 9  
1.14 Additive Decomposition 10  
1.15 Multiplicative decomposition 11  
1.16 First and Five rows of train and test data 15  
1.17 Optimal parameters for SES 17  
1.18 Graph of forecast values for SES 19  
1.19 Test RMSE value of SES 19  
1.20 Optimal parameters for DES 19  
1.21 Graph of forecasted values for DES 20  
1.22 Test RMSE value of SES and DES 21  
1.23 Optimal parameters for TES 21  
1.24 Graph for forecasted value for SES, DES, TES 22  
1.25 RMSE values for SES, DES, TES 22  
1.26 Graph of regression model forecasting 24  
1.27 RMSE values of regression and other models 24  
1.28 Graph of Naïve approach forecasting 25  
1.29 RMSE values of Naïve approach 26  
1.30 Graph of simple Average model 27  
1.31 RMSE values of Simple Average model 27  
1.32 Graph of Moving Average model 28  
1.33 RMSE values for 2,4,9 point Moving Average 29  
1.34 RMSE values for Moving Average model 29  
1.35 ADF test using statsmodel 30  
1.36 ADF test with difference at one level 30  
1.37 Plot to check stationarity 31  
1.38 Checking training dataset stationarity 31  
1.39 Training dataset stationarity using difference 32  
1.40 Various ARIMA combination 32  
1.41 Sorting the ARIMA values 33  
1.42 Automated ARIMA result 33  
1.43 RMSE values for Automated ARIMA 34  
1.44 ACF plot showcasing seasonality 34  
1.45 Automated SARIMA AIC values 35  
1.46 SARIMAX result 36  
1.47 Plot of SARIMA 36  
1.48 SARIMA predicted summary frame 37  
1.49 RMSE value for Automated SARIMA 37  
1.50 SARIMA values based on AIC values 38  
1.51 SARIMA at 12 result 39  
1.52 Plot diagnostic for SARIMA at 12 39  
1.53 RMSE values for Automated SARIMA at 12 40  
Problem:

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.

**SPARKING** DATASET:

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

Head of the data:   
   
1.1 Head of the data

Tail of the data:   
Table

Description automatically generated  
1.2 Tail of the data  
  
Shape of the data :   
  
1.3 Shape of the data

Checking for Null values:   
Text

Description automatically generated  
1.4 Null values  
We can see that there are no Null values present in Sparkling dataset.

Adding the time stamp to the dataframe  
Text, letter

Description automatically generated  
  
Table

Description automatically generated  
1.5 Adding the timestamp to the dataframe   
  
Table

Description automatically generated  
After setting the timestamp at the Index in the dataset after dropping YearMonth column then it will be:  
Table

Description automatically generated  
1.6 Setting the timestamp

Plotting the data :   
  
Chart, histogram

Description automatically generated  
1.7 Plotting graph of the dataset

From the above plot, we can observe that there was a slight upward trend in the year 1988 along with a seasonal pattern.

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

Describe the data :   
Graphical user interface, application

Description automatically generated  
1.8 Description of the dataset

Yearly Boxplot :  
Chart, box and whisker chart

Description automatically generated

We can see in the yearly boxplot that how the sales were decreased and increased over the past few years and the highest number of sales was done in the year 1989.   
  
1.9 Yearly boxplot showcasing the sale of Sparkling wine

Monthly Boxplot :   
Chart, box and whisker chart

Description automatically generated  
1.10 Monthly boxplot showcasing the sale of Sparking wine

We can see from the above monthly boxplot, that the sale of Sparking wine was highest in the month of December.

Monthly sales across the years :   
Application, table, Excel

Description automatically generated

Monthly sales across the years plot :   
Chart, line chart

Description automatically generated  
1.11 Graph showcasing the sale of Sparkling wine across the year for every month

As per the above plot, we can observe that the highest sale was in the month of December in 1987.

Empirical Cumulative Distribution :

Chart, line chart

Description automatically generated  
1.12 Empirical Cumulative Distributive Graph

Empirical Cumulative Distribution will explain about the % of the data points referring to what number of sales.

Average Sparkling sales per month:   
A picture containing graphical user interface

Description automatically generated  
1.13 Graph showcasing average sparkling sales and its percentage change over the years.

The above plot shows the average sparking sales per month as well as the month on month % change in Sparking sales. We can see that the average Sparkling sale seems to be highest in the year 1987-88.

With the Exploratory data analysis, we can hereby see that the data starts from year 1980 and will end in the year 1998. There is no Null values present in this dataframe and already we have created a date timestamp and then we had replaced it as index by dropping the YearMonth column and we can see that there are 187 rows and 2 columns.

Decomposition:

Additive Decomposition:

Graphical user interface

Description automatically generated  
1.14 Additive Decomposition

Table

Description automatically generated

Graphical user interface, text, table

Description automatically generated

Graphical user interface, text, table

Description automatically generated

The above output, we can see the top 10 head details of the trend, seasonal, residual.

By doing additive decomposition on the dataset, we can see that there are residuals located around 0 from the plot which in turn gives pattern that is said to be non-essential for model building.

Multiplicative Decomposition :

Graphical user interface, histogram

Description automatically generated  
1.15 Multiplicative Decomposition

Text, table

Description automatically generated  
  
Graphical user interface, text, table

Description automatically generated  
  
Table

Description automatically generated

The above output, we can see the top 10 head details of the trend, seasonal, residual.

By performing multiplicative decomposition on the dataset, we can see now that the residuals are located near 1. So, we can use multiplicative decomposition for building our models.

Plotting the Autocorrelation and partial autocorrelation function plots:   
Timeline

Description automatically generated  
Timeline

Description automatically generated  
From the above all the plots, we can see that there seems to be a seasonality in the data.

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

Let’s split the data into train and test and will plot the train and test data as train data will be till end of year 1990 whereas test data will start from year 1991.

Train head:  
Table

Description automatically generated

Train Tail:  
Table

Description automatically generated  
Test head:  
Table

Description automatically generated  
  
Test tail:  
Table

Description automatically generated

1.16 First and five rows of training and test data  
  
Train shape details:  
  
  
Test shape details:



Checking for stationarity of Train data time series:  
Graphical user interface, histogram

Description automatically generated

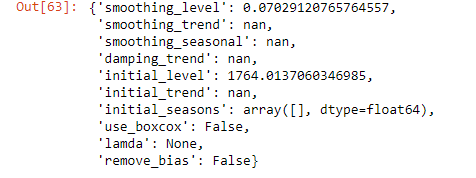
From the above plot, we can see that the series is not stationary at 𝛼 = 0.05 value.

Graphical user interface

Description automatically generatedWe can observe that after taking a difference of order 1then the series have become stationary at 𝛼 = 0.05 value.   
Info of the train data:   
Text, letter

Description automatically generated

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

Simple Exponential smoothing model :  
  
  
1.17 Optimal parameters for single exponential smoothing

SimpleExpSmoothing class must be instantiated and should be passed to the the training data. SES is a time series forecasting method for univariate data without trend or seasonality.

The fit() function is then called providing the fit configuration, the alpha value, smoothing\_level. If this is omitted or set to None, the model will automatically optimize the value. So in order to build an SES model, we should first autofit the smoothing model and by using autofit.params function we shall ask Python to choose the optimal parameters and we get the optimal parameter for SES.

SES predict:  
Table

Description automatically generated

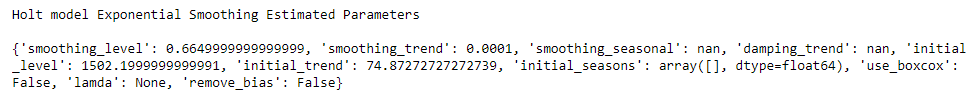
Plotting the Training data, Test data and the forecasted values  
Chart, line chart

Description automatically generated  
1.18 Graph of forecasted values for simple exponential smoothing  
  
From the above graph, we can see that the optimized smoothing level seems to be 0.07 and it shows the forecasted values for Train and test dataset using SES.

RMSE value :   
Shape

Description automatically generated

1.19 Test RMSE value of SES

Double Exponential smoothing:  
  
  
1.20 Optimal parameters for double exponential smoothing

DES uses two weights alpha and beta to update the components at each period by employing a level component and a trend component at each period. The above output shows the optimal parameters for DES.

DES predict:  
Table

Description automatically generated

Plotting the Training data, Test data and the forecasted values   
Chart, histogram

Description automatically generated  
1.21 Graph of forecasted values for double exponential smoothing

The above plot depicts that the double exponential smoothing forecast values using both train and test datasets is gradually picking up the trend component along with the level component. And we can see here that the smoothing level has been optimized to be alpha = 0.663 , beta= 9.966

RMSE value:   
Graphical user interface

Description automatically generated  
1.22 Test RMSE value of SES and DES

Inferences which can be drawn here is, we can see that the SES has really done well when compared to the DES. This is because the DES model was unable to pick up the trend component available in the data.

Triple Exponential Smoothing:  
A picture containing text

Description automatically generated  
1.23 Optimal parameters for Triple Exponential Smoothing   
  
The above output shows the TES estimated parameters forecast based on the test set and can be found that the optimal smoothing level to be at alpha = 11.127 , beta = 0.123 and gamma = 0.460

TES predict:  
Table

Description automatically generated

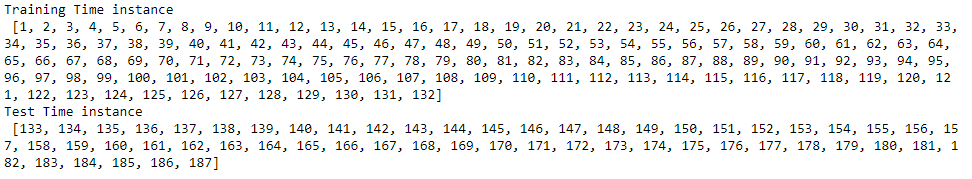
TES is generally used to handle the time series data which contains seasonal component and is based on alpha, beta and gamma and also seasonal and trend can be either additive or multiplicative.

Plotting the Training data, Test data and the forecasted values:  
Chart, histogram

Description automatically generated  
1.24 Graph for forecasted values for single , Double and Triple Exponential Smoothing.  
  
The above plot shows that the TES is picking up the seasonal component as well as the trend level of the data.

RMSE value:  
Graphical user interface, text

Description automatically generated  
1.25 RMSE values for Single, Double and Triple Exponential Smoothing   
  
From the above output, we can see TES model shows very low RMSE value when compared to SES , DES and so TES is a very good model which can be used for forecasting.

Linear Regression:   
  
Train and test time instances:  


Table

Description automatically generated  
  
Chart, histogram

Description automatically generated  
1.26 Graph of regression model forecasting  
  
This method is used for predicting a future response based on the response history and the transfer of dynamics from the relevant predictors. So, for a regression model, to build the model we will take both the train and test data and fit it using Linear Regression. Here, from the above graph we can see that the regression model is not performing a very good forecasting when compared to the remaining models.

Test data – RMSE:  
Graphical user interface, text

Description automatically generated  
1.27 RMSE values of regression along with other models built.   
  
The RMSE values seems to be very high when compared to SES , DES and TES. So we can see from the above output, that RMSE value for regression is very low when compared to DES so regression is not the best fit model for forecasting.

Naive Approach:   
Text

Description automatically generated  
  
Chart, histogram

Description automatically generated  
1.28 Graph of Naïve approach forecasting  
  
So for Naive approach 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. The above plot shows that the Naïve forecast on the test data is really very high when compared to the actual data.

Test data – RMSE  
Graphical user interface, text

Description automatically generated  
1.29 RMSE values of naïve approach along with previously built models

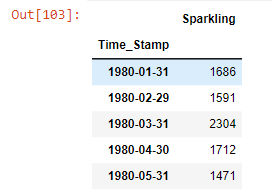
Simple Average:   
Table

Description automatically generated  
  
Chart, histogram

Description automatically generated  
1.30 Graph of Simple Average model  
  
For Simple average method, we will forecast by using the average of the training values and the above plot shows that the forecast sale is really very low when compared to the other models. SMA is calculated by adding up the last n period’s values and then dividing that number by n. So basically, the moving average value is considered as the forecast for the next period.

Test data – RMSE:  
Graphical user interface, text, application

Description automatically generated  
1.31 RMSE values of Simple Average model  
  
The RMSE value for SAM is very low when compared to Naives and Regression models and it’s more than TES.

Moving Average:  
  
Chart

Description automatically generated  
1.32 Graph of moving average model trailing points  
  
Chart

Description automatically generated  
  
MA is basically a technique that calculates the overall trend in a dataset. From the above graph we can see that the 2-point trailing MA is really performing well on test data when compared to the other trailing points.

RMSE value:  
Text

Description automatically generated  
1.33 RMSE values for 2,4 and 9 point moving average  
  
Table

Description automatically generated with medium confidence  
1.34 RMSE values for Moving Average model and all the previously built models   
  
When compared to 4-, 6- and 9-point MA RMSE we see that the RMSE for 2-point MA has a really very low value. When compared to the previously built models we see that the MA has the 2nd lowest RMSE value and remains high when compared to TES.

1. 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 a unit root test which determines whether there is a unit root and subsequently whether the series is non-stationary.

The hypothesis 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.

ADF test using stats model:  
A picture containing text

Description automatically generated  
1.35 ADF Test using statsmodel   
  
We see that at 5% significant level the Time Series is non-stationary and to make the data stationary, we will use 1 level of differencing and return the code.

ADF test with difference at one level:  
  
  
1.36 ADF test with difference at one level   
  
Graphical user interface, chart, line chart

Description automatically generated  
1.37 Plot to check stationarity   
  
By using one level differencing, we can now see that the data is stationary from the above plot.

Checking the stationarity on the training dataset:  
  
Graphical user interface, histogram

Description automatically generated  
1.38 Checking Training dataset stationarity  
  
We see that at 5% significant level the Time Series is non-stationary. Let us take a difference of order 1 and check whether the Time Series is stationary or not.

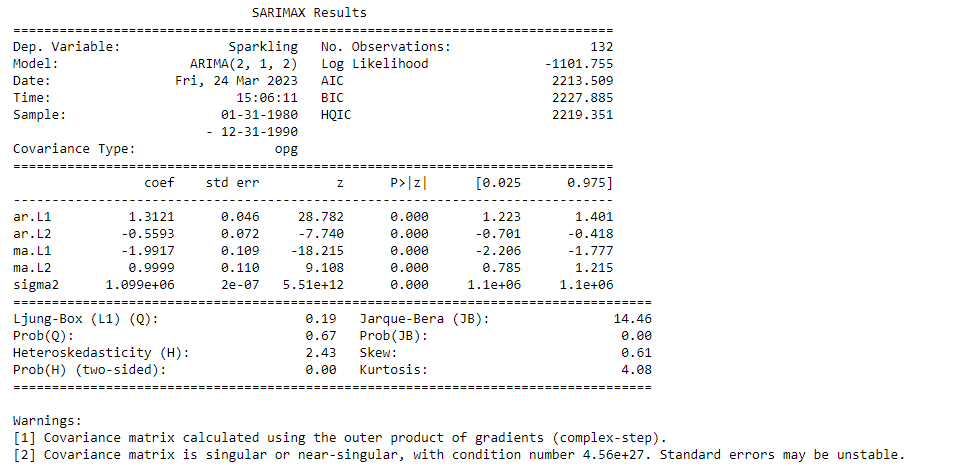
A picture containing graphical user interface

Description automatically generated  
1.39 Training dataset stationarity using difference order 1

We see that at 𝛼 = 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 ARIMA:  
  
Text

Description automatically generated  
1.40 Various ARIMA combination  
  
Auto ARIMA is a class of statistical algorithms that captures the standard temporal dependencies that is unique to a time series data. We use different parameter combinations to find the value of p,d,q to build the model. In this model, Python does the permutation and combination to arrive at the best values for p,d,q.   
  
  
  
Sorting the ARIMA values based on lowest AIC  
  
Table

Description automatically generated with medium confidence  
  
1.41 Sorting the ARIMA values based on lowest AIC  
  
We can see that the auto ARIMA model can be built using 2,1,2 since it has the lowest AIC.  
  
  
1.42 Automated ARIMA result   
  
  
RMSE value:  
  
  
  
Table

Description automatically generated  
1.43 RMSE values for Automated ARIMA and all the other models built

Automated SARIMA:  
Timeline

Description automatically generated  
1.44 ACF Plot showcasing seasonality

SARIMA is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component. From the above plot, we can see that there can be a seasonality of 6 as well as 12.   
  
Text

Description automatically generated  
  
Text

Description automatically generated

SARIMA at 6:  
Text

Description automatically generated  
  
1.45 Automated SARIMA AIC values in ascending order  
  
Table

Description automatically generated  
1.46 SARIMAX Result

Plot for SARIMA:  
Chart

Description automatically generated  
1.47 Plot of SARIMA  
  
From the above graph, we see that all the individual diagnostics plots almost followed the theoretical numbers and thus we cannot develop any pattern from these plots.   
  
  
SARIMA predicted summary frame:   
  
Table

Description automatically generated  
1.48 SARIMA predicted summary frame  
  
Test Data – RMSE:  
  
SARIMA predicted summary frame  


RMSE value:   
Table

Description automatically generated  
1.49 RMSE values for Automated SARIMA and all the other models built   
  
  
SARIMA at 12:  
Text

Description automatically generated  
  
Text

Description automatically generated  
  
  
SARIMA values based on lowest AIC values:  
Text, letter

Description automatically generated  
1.50 SARIMA values based on lowest AIC values

SARIMA at 12 result:  
Table

Description automatically generated  
1.51 SARIMA at 12 result  
  
Plot for SARIMA at 12:  
Chart

Description automatically generated  
1.52 Plot diagnostic for SARIMA at 12

Like the last iteration of the model where the seasonality parameter was taken as 6, here also we see that the model diagnostics plot does not indicate any remaining information that we can get.

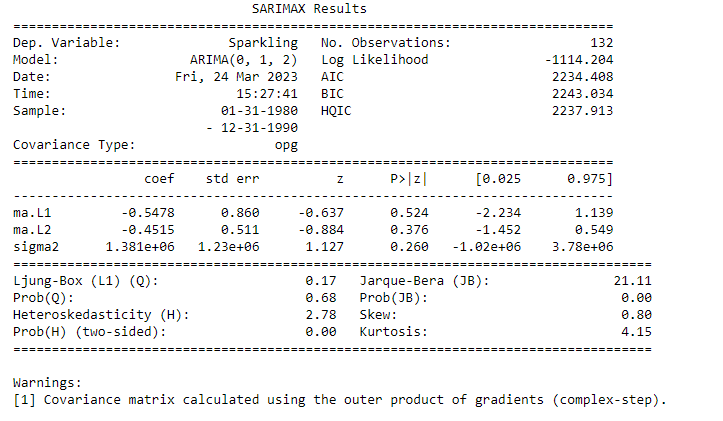
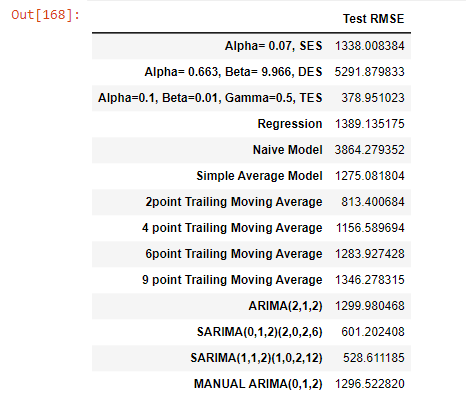
RMSE:   
Table

Description automatically generated  
  
Table

Description automatically generated  
1.53 RMSE values for automated SARIMA at 12 and all the other models built  
  
We can see that the RMSE value has only reduced further by a margin when the seasonality parameter was changed to 12, and there is not much difference in the model diagnostics of the two models.

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:  
Chart, histogram

Description automatically generated  
1.54 ACF and PACF plot  
  
So, we have taken alpha=0.05. In ARIMA model, p which comes from the significant lag before which the PACF plot cuts-off to 0. The MA parameter in an ARIMA model is q which comes from the significant lag before the ACF plot cuts-off to 0. By looking at the above plots, we will take the value of p and q to be 0 and 0 respectively.   
  
  
  
1.55 ARIMA Model result   
  
We get a comparatively simpler model by looking at the ACF and the PACF plots.   
  
  
RMSE value:   
  
  
  
1.56 RMSE values for Manual ARIMA and all the other models built  
  
SARIMA manual:   
Chart

Description automatically generated  
1.57 Graph for SARIMA at 6 dataframe

We see that our ACF plot at the seasonal interval 6 does not taper off. So, we go ahead and take a seasonal differencing of the original series.   
  
Chart

Description automatically generated  
1.58 Graph for dataset SARIMA at 6 1st order difference on seasonally differenced series  
We see that there is a trend and a seasonality. So, now we take a seasonal differencing and check the series.

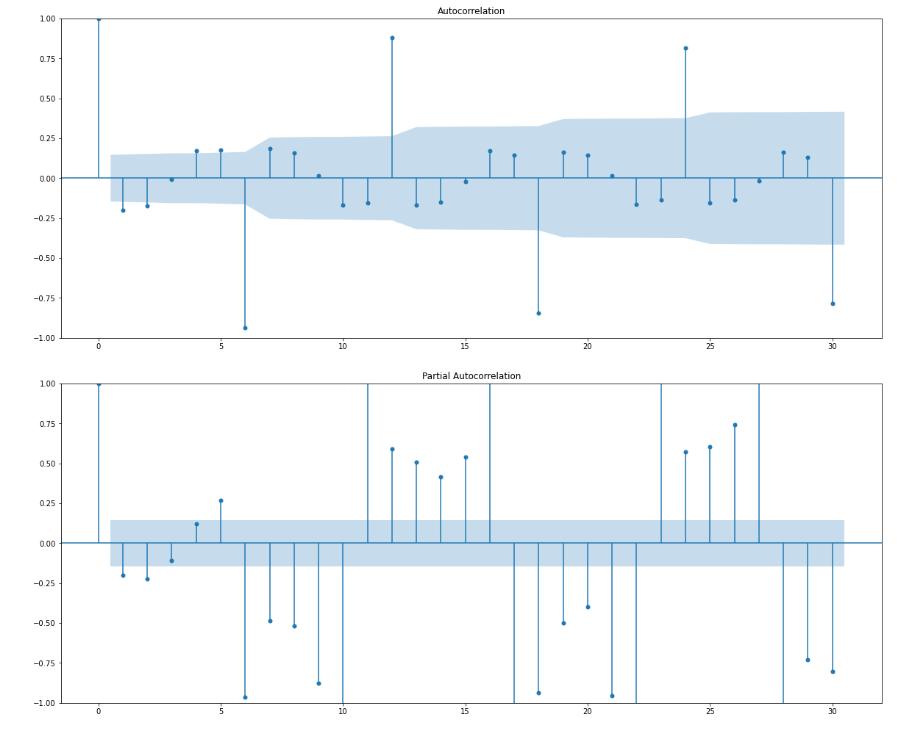
A picture containing chart

Description automatically generated  
  
We see that there might be a slight trend which can be noticed in the data. So, we take a differencing of first order on the seasonally differenced series.

Graphical user interface, chart

Description automatically generated  
  
Now we see that there is almost no trend present in the data. Seasonality is only present in the data.   
  
Checking the stationarity:   
Graphical user interface

Description automatically generated  
1.59 Stationarity of dataset for building SARIMA model

Checking the ACF and the PACF plots:  


1.60 ACF and PACF plots

Here, we have taken alpha=0.05. We are going to take the seasonal period as 6. We will keep the p(1) and q(1) parameters same as the ARIMA model. The Auto-Regressive parameter in a SARIMA model is P which comes from the significant lag after which the PACF plot cuts-off to 0. The Moving-Average parameter in a SARIMA model is q which comes from the significant lag after which the ACF plot cuts-off to 0. Remember to check the ACF and the PACF plots only at multiples of 6 (since 6 is the seasonal period). By looking at the plots we see that the ACF and the PACF do not directly cut-off to 0.   
  
Manual SARIMA result:  
Table

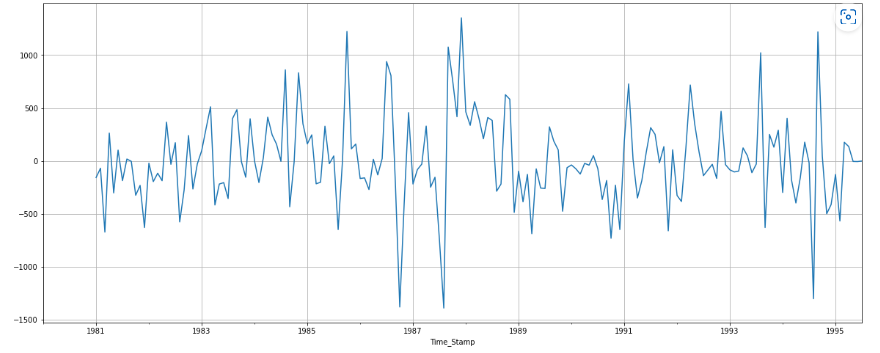
Description automatically generated  
1.61 Manual SARIMA Result

Plot for SARIMA at 6:  
Chart

Description automatically generated  
1.62 Plot diagnostic for SARIMA at 6  
  
The model diagnostics plot looks okay for SARIMA at 6.

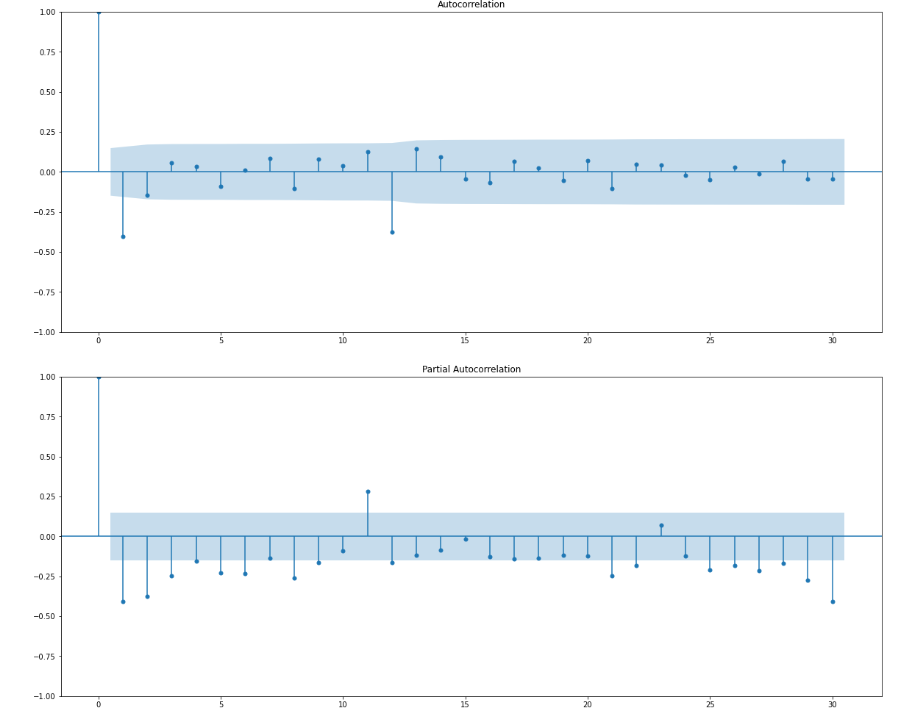
RMSE value:   
Table

Description automatically generated  
  
  
  
Graphical user interface, table

Description automatically generated  
1.63 RMSE values for Manual SARIMA at 6 and all the other models built  
  
SARIMA at 12:  
  
1.64 Graph for SARIMA at 12

We can observe that there might be a slight trend, so we take difference of first order on the seasonally differenced series.

  
1.65 Graph for dataset SARIMA at 12 1st order difference on seasonally differenced series   
  
From the above plot, we see that there is almost no trend present in the data. Seasonality is only present in the data.   
  
Graphical user interface

Description automatically generated  
  
  
1.66 ACF and PACF plots for newly modifies time series SARIMA at 12  
SARIMA at 12 result:   
Table

Description automatically generated  
1.67 SARIMA at 12 result

Plot of SARIMA at 12:   
Chart, line chart

Description automatically generated  
1.68 Plot diagnostic SARIMA at 12   
  
RMSE value:   
Table

Description automatically generated

  
  
Table

Description automatically generated

1.69 RMSE values for Manual SARIMA at 12 and all the other models built

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

Description automatically generated  
1.70 Dataframe models built along with their RMSE

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.   
  
  
Text

Description automatically generated  
1.71 List of Prediction for next 12 months

Prediction for full dataset:   
Chart

Description automatically generated  
1.72 Prediction for full dataset   
  
  
Prediction values:   
Text

Description automatically generated  
  
1.73 Class Intervals

Class intervals:   
Chart

Description automatically generated with medium confidence  
1.74 Predicted values for Sparkling wine sale

Chart, histogram

Description automatically generated  
1.75 Graph for class intervals and next 12 months prediction

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. We see that the best model is the Triple Exponential Smoothing with multiplicative seasonality with the parameters α = 0.25, β = 0.0 and γ = 0.74.

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

From the given data, we can see that in the month of January, February, June and May it has recorded the lowest number of Sparkling wine sales across the years. It can be seen here that. Highest sale was recorded in the month of December in the year 1987. Even in the months of November, October and September, the sales were more when compared to the other months. The highest sale had its mean around 5800 in the month of December and lowest mean around 600 in the month of June. Decomposition was done using additive decomposition and it resulted up in residuals to have pattern around 0 but while using multiplicative method, the data did not show any pattern for residuals and the components were located around 1. The models were built on training dataset and the performance was checked on test data using RMSE. The TES had the lowest AIC value at 379. The lowest RMSE value is the optimal for creating the best model for overall predictions.

Since TES is the best fit model so future sales predictions for next 12 months can be done. Using the predictions, the company will have an insight on the profit, loss, production quantity, materials needed to produce Sparking wine. The company can even forecast. Like which month will have higher or lower production with respect to the demand and supply for the month. The company can use various strategies to make their business more profitable such as advt in Radio, T.V , newspaper, etc to increase their sale , consumer base and production for the time period where the sale is low. Various offers and discounts can give to both the sale team as well as people who purchase the products to boost sales. Special offers at certain events can be discussed to uplift the production at low seasons. The company can have low production for months with lowest sale to avoid wastage which in turn helps with profit in the future.