Business Report on Time Series Analysis of sales of Sparkling Wine

Objective:- To forecast the approximate sales number for 12 months into the future basis the past data provided for 187 months for Sparkling wine.

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

Solution: - We have started the time series analysis of the given dataset by importing the usual libraries with an additional library for the decomposition of the time series data.

- The data talks about the sales number for a particular given month in a year.
- We checked the data types of the columns of the dataset and found that "YearMonth" column is of Object data type and column "Sparkling" is of integer data type. Hereby we need to instruct python that we are reading a time series data.

```
YearMonth    object
Sparkling     int64
dtype: object
```

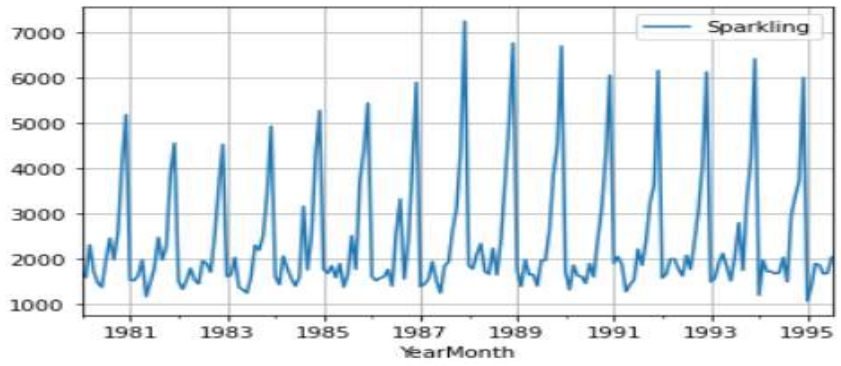
- We need to parse the data to make python understand that we are working in time series data and now we can observe that the "YearMonth" column has been identified as datetime data type. (Note: ns stands for nano seconds)

```
YearMonth    datetime64[ns]
Sparkling             int64
dtype: object
```

- It is also recommended that for all time-series analysis we should mostly put the time series reference column as the index. It makes it easy while slicing and dicing the data. Which can be achieved by passing an additional function known as "index_col". And now we see that the "YearMonth" variable as set as index now.

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

- Checking the shape of data and we can find that there are 187 observations and 1 target variable.
- Plotting data: -



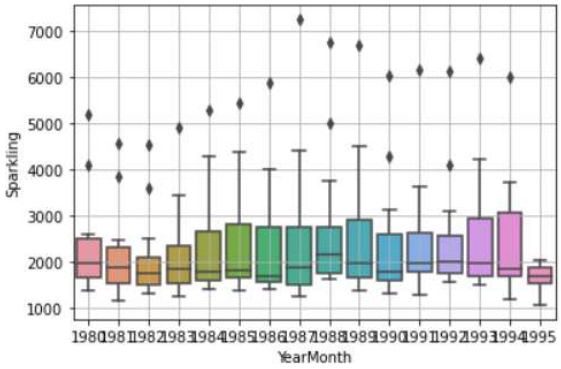
- From the above plot we cannot visually conclude that the data have presence of both trend and seasonality.
- Checking for null values we can find zero missing data in the dataset.
- Describing data: -

	Sparkling
count	187.000000
mean	2402.417112
std	1295.111540
min	1070.000000
25%	1605.000000
50%	1874.000000
75%	2549.000000
max	7242.000000

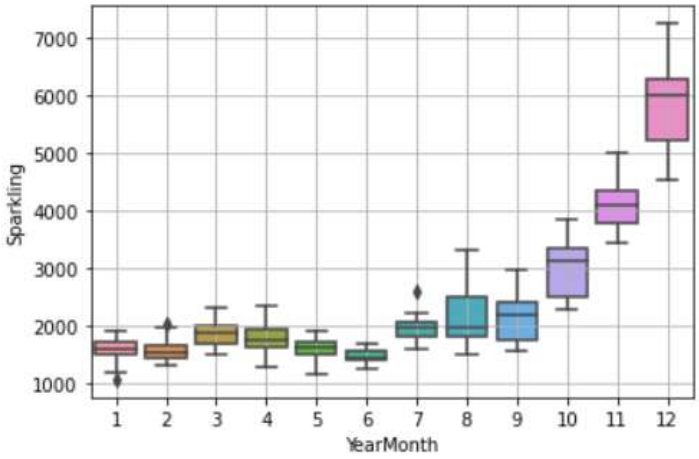
2. Perform appropriate Exploratory Data Analysis to understand the data and also perform decomposition.

Solution: - Performing exploratory data analysis: -

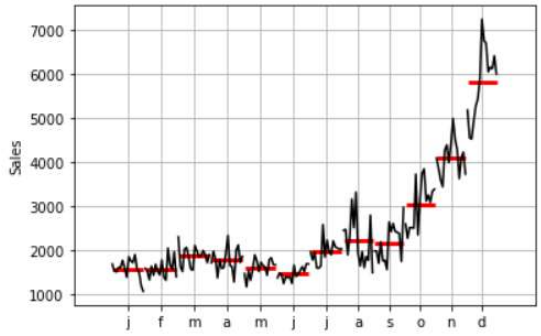
- Plotting yearly box plot to check on the sales distribution and trends across years and it also gives a glimpse of trend and outliers in the dataset.



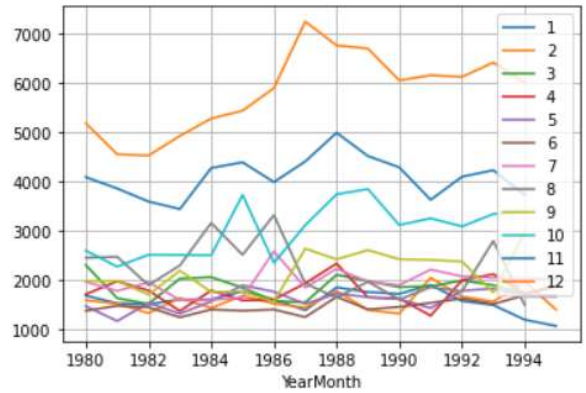
- Plotting monthly box plot for all subsequent years.



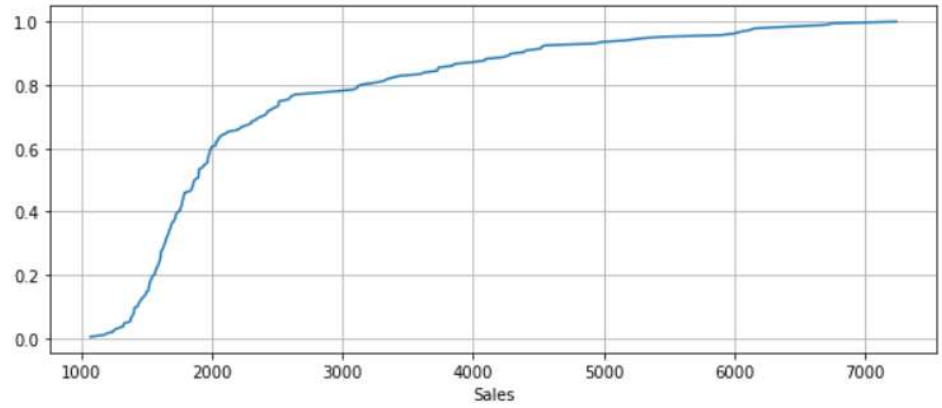
- From above we can conclude that for every year the sales are low at starting od the year and we can notice significant improvement in sales numbers in last 3 months for every year.
- Plotting a time series month plot to understand the spread of accidents across different years and within different months across years.



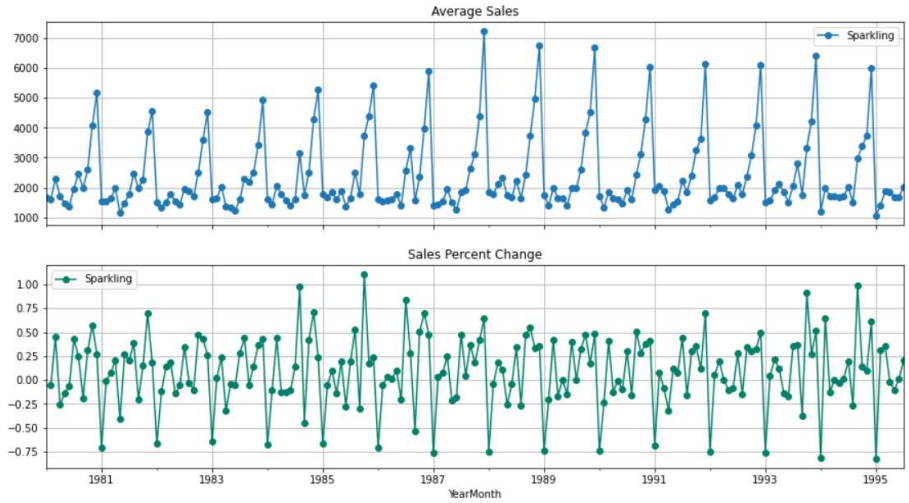
- Plot a graph of monthly Sales across years.



- We can notice that the sales are at decent numbers during mid of every year however it shows decreasing pattern at the end of the year just like starting of the year.
- Plotting the Empirical Cumulative Distribution



- Plot the average Sales per month and the month-on-month percentage change of Sales.

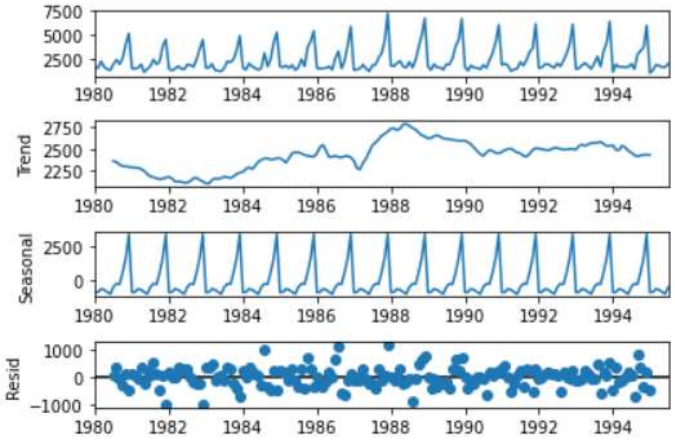


Performing Decomposition of Data: -

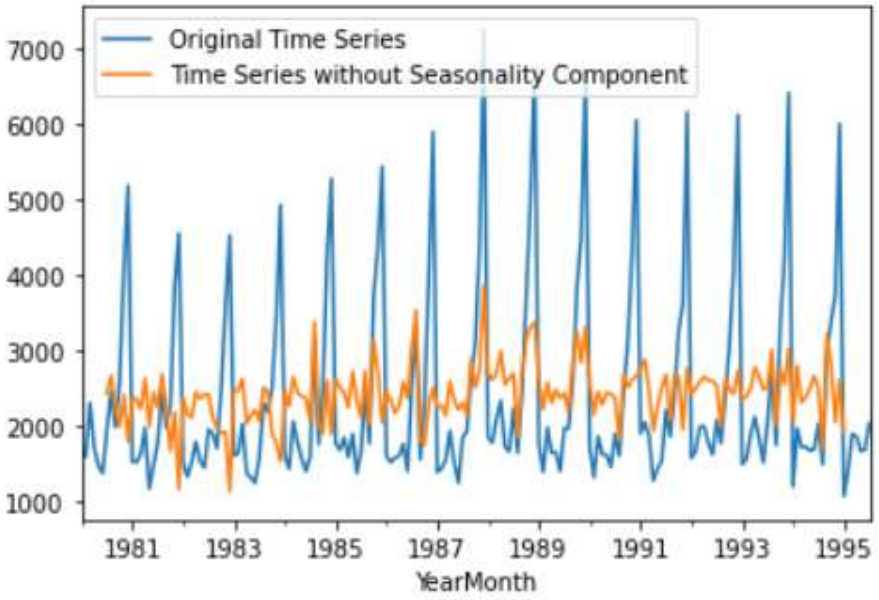
From the above plot we can see that we have a time series data which is not a constant time series. It has a fluctuating trend.

It also seems to have a repetitive nature which is a repeatable pattern every year. Which is known as seasonality. But it does not seem to be a constant seasonality. The peaks are repeated but peaks are decreasing as we move along the years. And we can ascertain it even better when we decompose this time series data.

1. **Additive Decomposition**
   - Below is the plot of the decomposed time series using additive method



   - The above shows that the plot of original data and then the plot of trend. It has NO clear trend present in the data but it has a seasonality component.
   - We also have some residual/ error component available in data the distribution is random. Its not showing a specific pattern. Which indicates that it can be good with additive decomposition still we can compare that with multiplicative decomposition to have better understanding.
   - Plotting graph with and without seasonality



- Inspecting the trend, seasonal and residual components: -

Trend

YearMonth	
1980-01-01	NaN
1980-02-01	NaN
1980-03-01	NaN
1980-04-01	NaN
1980-05-01	NaN
1980-06-01	NaN
1980-07-01	2360.666667
1980-08-01	2351.333333
1980-09-01	2320.541667
1980-10-01	2303.583333
1980-11-01	2302.041667
1980-12-01	2293.791667

Name: trend, dtype: float64

Seasonality

YearMonth	
1980-01-01	-854.260599
1980-02-01	-830.350678
1980-03-01	-592.356630
1980-04-01	-658.490559
1980-05-01	-824.416154
1980-06-01	-967.434011
1980-07-01	-465.502265
1980-08-01	-214.332821
1980-09-01	-254.677265
1980-10-01	599.769957
1980-11-01	1675.067179
1980-12-01	3386.983846

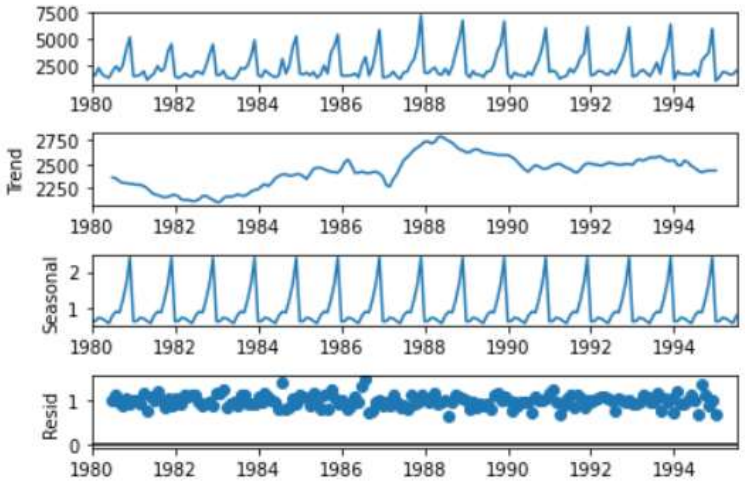
Name: seasonal, dtype: float64

Residual

YearMonth	
1980-01-01	NaN
1980-02-01	NaN
1980-03-01	NaN
1980-04-01	NaN
1980-05-01	NaN
1980-06-01	NaN
1980-07-01	70.835599
1980-08-01	315.999487
1980-09-01	-81.864401
1980-10-01	-307.353290
1980-11-01	109.891154
1980-12-01	-501.775513

Name: resid, dtype: float64

2. **Multiplicative Decomposition**
   - Below is the plot of the decomposed time series using multiplicative method



   - We can notice that between the additive and multiplicative plot the scales have changed.
   - We can see the error component is also flat and near to “1”.
   - Seasonal component is very clear however we cannot see a clear trend in the data.

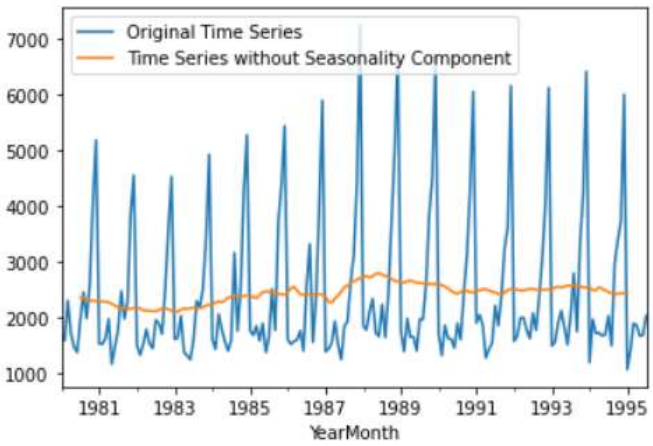
- Distribution of error component is still random.
- Inspecting the trend, seasonal and residual components of data

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

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

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

- Plotting graph with and without seasonality

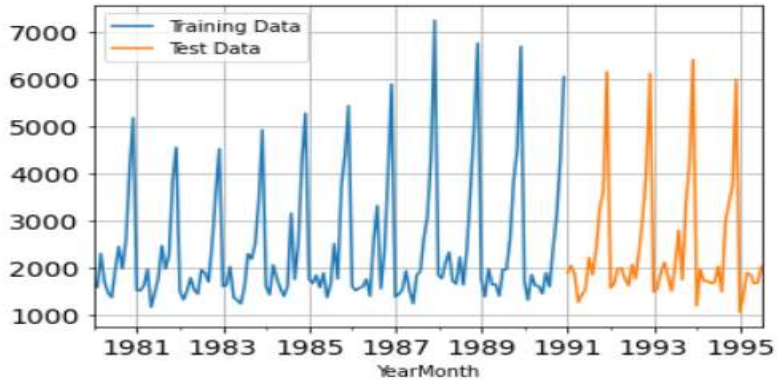


3. Split the data into training and test. The test data should start in 1991.

Solution: - Splitting data into test and train dataset in such a way that the train data has all the observations before 1991 and in test data observations start from 1991 and after.

After splitting the data and checking the shape we find that the train data have 132 observations and test dataset have 55 observations.

Below is the plot of train and test dataset: -



We understand that the train-test split cannot be done randomly as we are dealing with continuous data and time series has to be in continuous manner.

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.

Solution: - Building various exponential smoothing models: -

To build various exponential smoothing model we have to start with importing ExponentialSmoothing library from statsmodels and also mean_sqaured_error library to compare the various models built which in-turn helps in selecting the best optimal model. The model with least Mean squared error will be considered the best optimal model as that model denoted less error components and less loss of data.

Here, we are about to build 4 models, 1st will be only with level (alpha), 2nd with level and trend component (alpha and beta), 3rd with Level, trend and additive seasonality component and 4th with level, trend and multiplicative seasonality component(alpha, beta and gamma).

- **Building Simple Exponential smoothing model: -**

  This method involves only the "Level" component of data (alpha) with no trend and no seasonality. which mean it is best suitable for data with no clear trend and seasonality. The value of alpha lied between 0 and 1. This model uses only single exponential component so also called as "Single Exponential Smoothing".

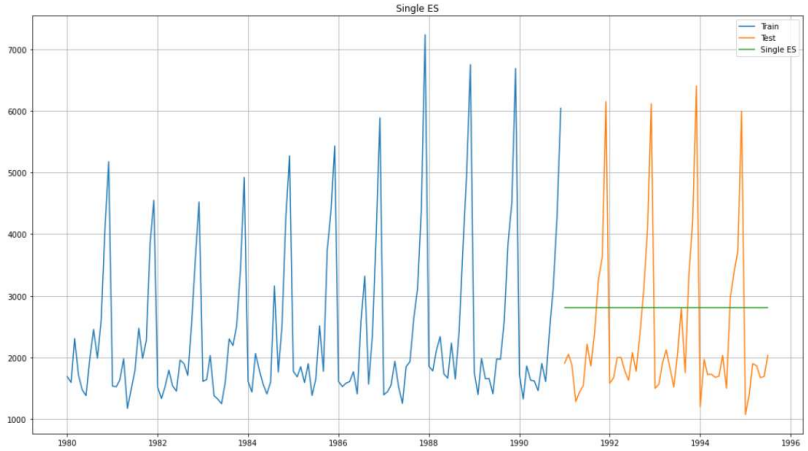
  Here we are building model on train data and predicting on testing data and to check the accuracy of the built model we are using RMSE as the parameter.

  Below are the parameter details for the model: -

```
{'smoothing_level': 0.07029459943040381,
 'smoothing_trend': nan,
 'smoothing_seasonal': nan,
 'damping_trend': nan,
 'initial_level': 1764.1004162520212,
 'initial_trend': nan,
 'initial_seasons': array([], dtype=float64),
 'use_boxcox': False,
 'lamda': None,
 'remove_bias': False}
```

  We can see that the value is close to "0" (zero), which means that the previous time series data not that accurately related to the forecast for the next period.

  It's a flat forecast and gives a constant value. (as shown in graph below in green line).



  By looking at the graph we can conclude that this is not the right model as the forecast is flat and doesn't acknowledge the element of trend and seasonality in the data.

Below is the RMSE value of this model: -

```
SES RMSE: 1338.0121443910186
SES RMSE (calculated using statsmodels): 1338.0121443910189
```

❖ **Building Double Exponential smoothing model: -**

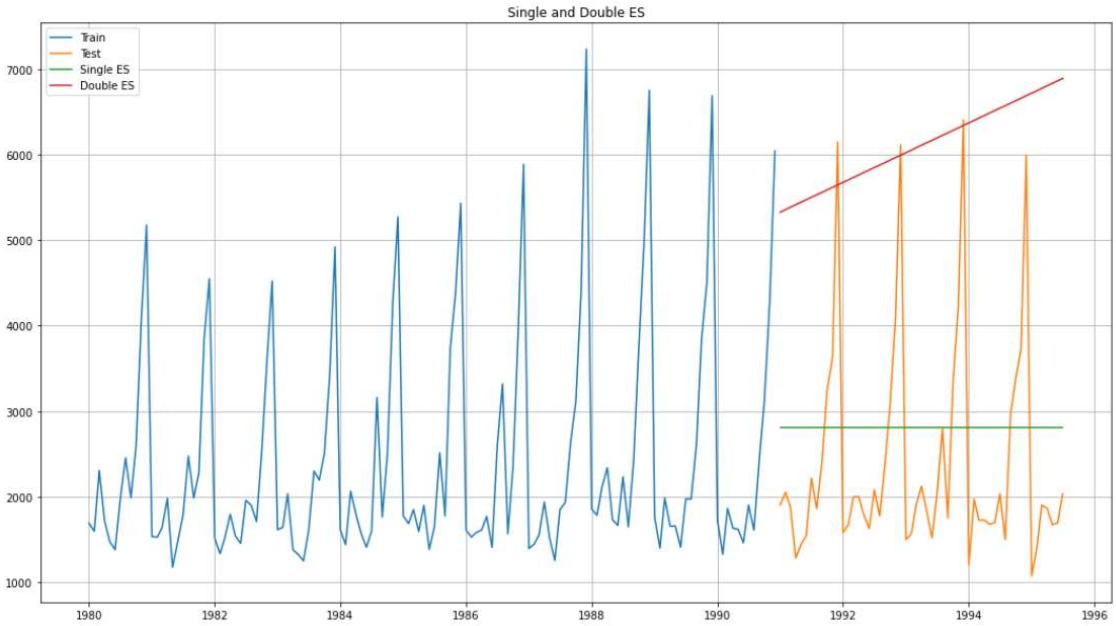
This method involves the "Level" component of data (alpha) as well as the "Trend" component (beta) with no seasonality. which mean it is best suitable for data with clear trend but no clear seasonality. This is also called as "Holts Linear method".

Below are the parameter details for the model: -

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

{'smoothing_level': 0.6638769092832238, 'smoothing_trend': 9.966251357628782e-05, 'smoothing_seasonal': nan, 'damping_trend': n
an, 'initial_level': 1502.5681711003654, 'initial_trend': 29.020225552837097, 'initial_seasons': array([], dtype=float64), 'use
_boxcox': False, 'lamda': None, 'remove_bias': False}
```

Below is the forecast plot using double exponential method: -



By looking at the graph we can see that this time the forecast values are not flat, but looks linear in nature.

Below is the RMSE value of this model: -

```
DES RMSE: 3949.993290409098
```

❖ **Building Triple Exponential smoothing model with additive seasonality: -**

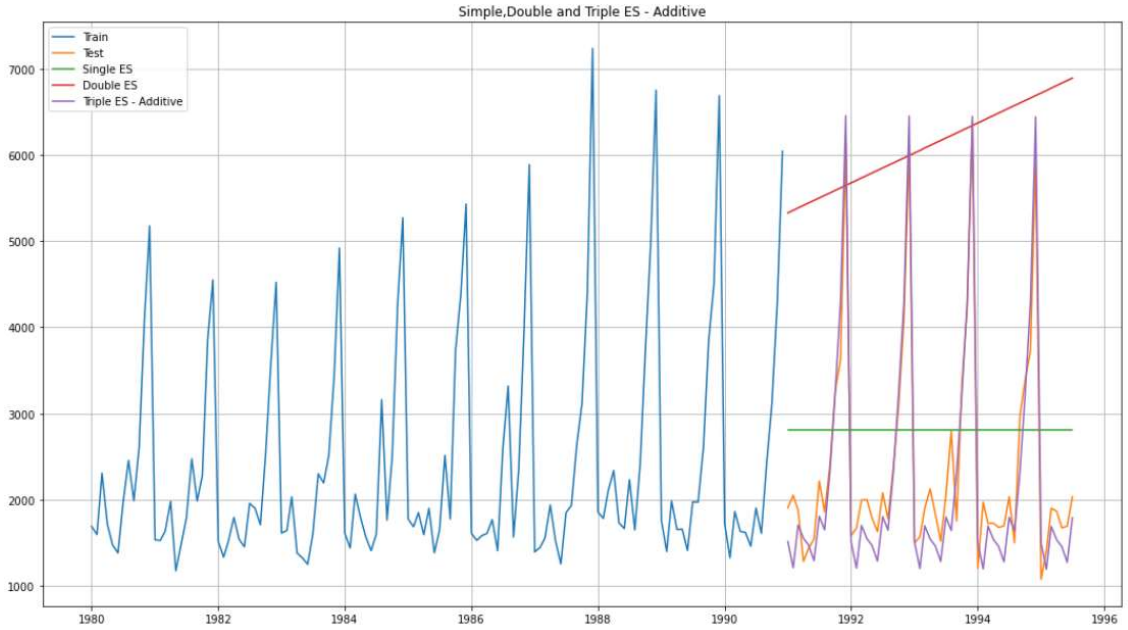
This method involves the "Level" component of data (alpha) as well as the "Trend" component (beta) along with "seasonality" component (Gamma) with additive nature. which mean it is best suitable for data with clear trend and clear seasonality. This is also called as "Holt winter's Linear method".

Below are the parameter details for the model: -

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

{'smoothing_level': 0.10005373820823961, 'smoothing_trend': 0.010034490652580457, 'smoothing_seasonal': 0.5095957543425532, 'da
mping_trend': nan, 'initial_level': 2364.584774604334, 'initial_trend': -0.016752880078245408, 'initial_seasons': array([-653.8
2559323, -736.67734144, -368.25456128, -483.63906084,
       -826.15467946, -832.96819741, -386.3751117 ,   91.82676187,
       -261.32455153,  265.38968222, 1580.26233564, 2619.56221896]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

Below is the forecast plot using triple exponential method: -



By looking at the graph we can see that the forecast values are getting better as it acknowledge both trend and seasonality component of data.

Below is the RMSE value of this model: -

```
TES RMSE: 379.6956857387101
```

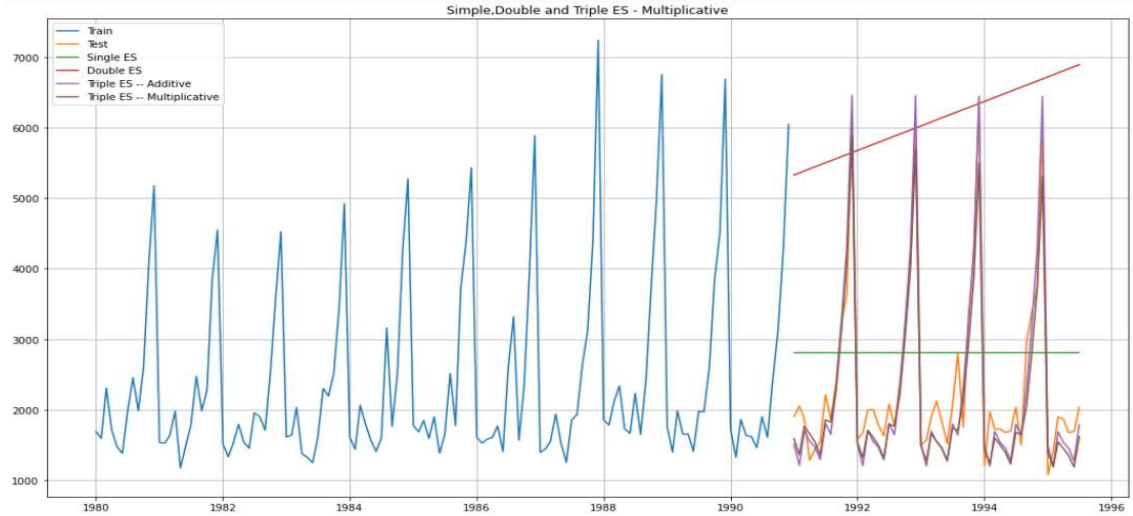
❖ **Building Triple Exponential smoothing model with multiplicative seasonality: -**

Below are the parameter details for the model: -

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

{'smoothing_level': 0.11194572287706502, 'smoothing_trend': 0.04979454913988668, 'smoothing_seasonal': 0.3616765678435302, 'dam
ping_trend': nan, 'initial_level': 2356.340229937152, 'initial_trend': -10.519480221963526, 'initial_seasons': array([0.7146511
8, 0.68302129, 0.90263858, 0.80589958, 0.65660325,
       0.65654363, 0.88525948, 1.132562  , 0.92225104, 1.21110112,
       1.8820382 , 2.38194187]), 'use_boxcox': False, 'lamda': None, 'remove_bias': False}
```

Below is the forecast plot using triple exponential method: -



Below is the RMSE value of this model: -

```
TES_am RMSE: 406.51016963157673
```

BELOW IS THE CONSOLIDATE RMSE VALUES OF ALL THE EXPONENTIAL MODEL BUILT: -

	Test RMSE
Single ES	1338.012144
Double ES	3949.993290
Triple ES - Additive	379.695686
Triple ES - Multiplicative	406.510170

Inferences: - From the above all the exponential models built we can conclude that the **triple exponential model with additive seasonality** out performs all the other model based on respective RMSE scores.

Building Linear Regression Model: -

In this particular linear regression, we are going to regress the "Sparkling" variable against the order of the occurrence. For this we need to modify our training data before fitting into linear regression model.

We see that the total observation in data is 187 out of which 132 are in training and 55 are in testing dataset.

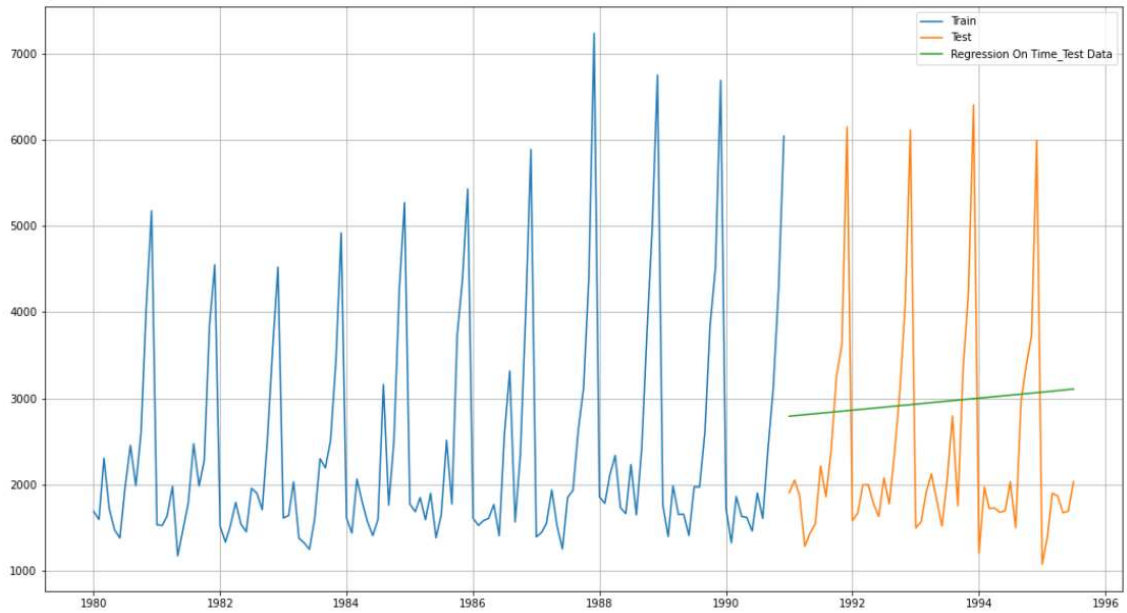
As linear regression requires at least one "X" variable to get the outcome of our interest but in our data, we have the time element. So, we will take the time as our independent variable and the sales number as the dependent variable. However, we have set the time variable as index. Now, we can create a dummy variable in sequence which will represent time variable as X but not as an index.

```
Training Time instance
 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 3
4, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65,
66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97,
98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123,
124, 125, 126, 127, 128, 129, 130, 131, 132]
Test Time instance
 [133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157,
158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 18
3, 184, 185, 186, 187]
```

We can see above that as we define the train data it contains 132 observations and test data contains observations from 133 till 187 and this is the independent variable.

Now we need to add the above created dummy variable to the original dataset and to do that we have taken a copy of the original dataset so that we do not mess up with the original data.

Now we will build the linear regression model in a usual way and below is the graphical representation of the forecast.

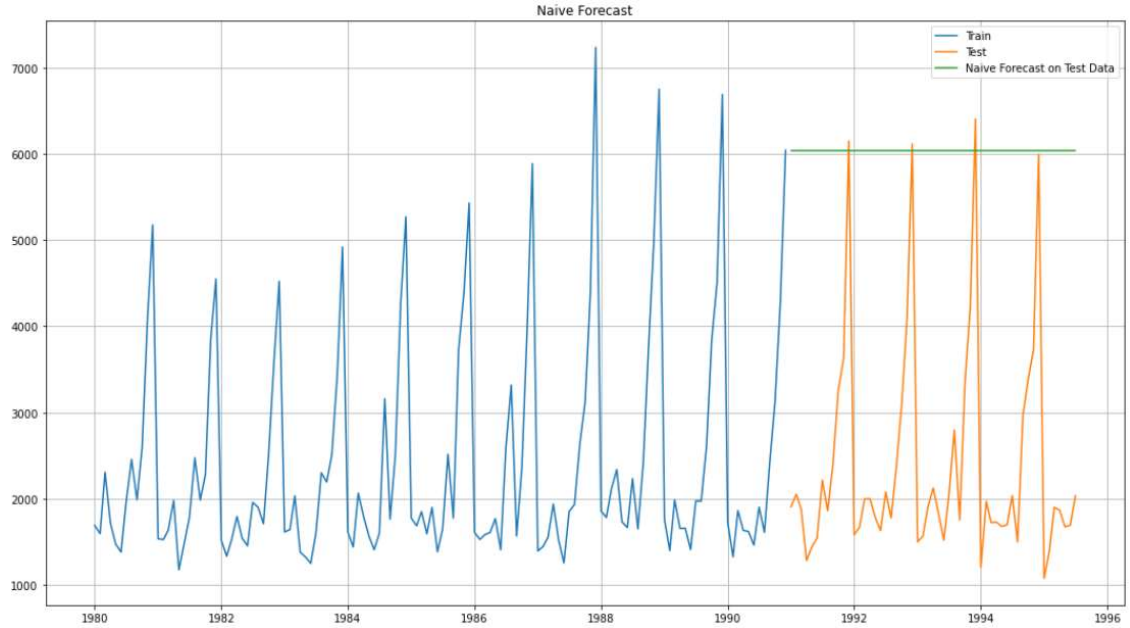


The green line shown above is the linear regression model.
Below is the RMSE score of linear regression model: -

```
For RegressionOnTime forecast on the Test Data,  RMSE is 1389.135
```

Building Naive Model: -

The naïve model works on the basis of the last value of the data and the same value repeats for rest of the data. Which mean that the forecast is going to be flat in nature. In this case the last value of the train set is 6047. Now let's see how the prediction plot looks like: -

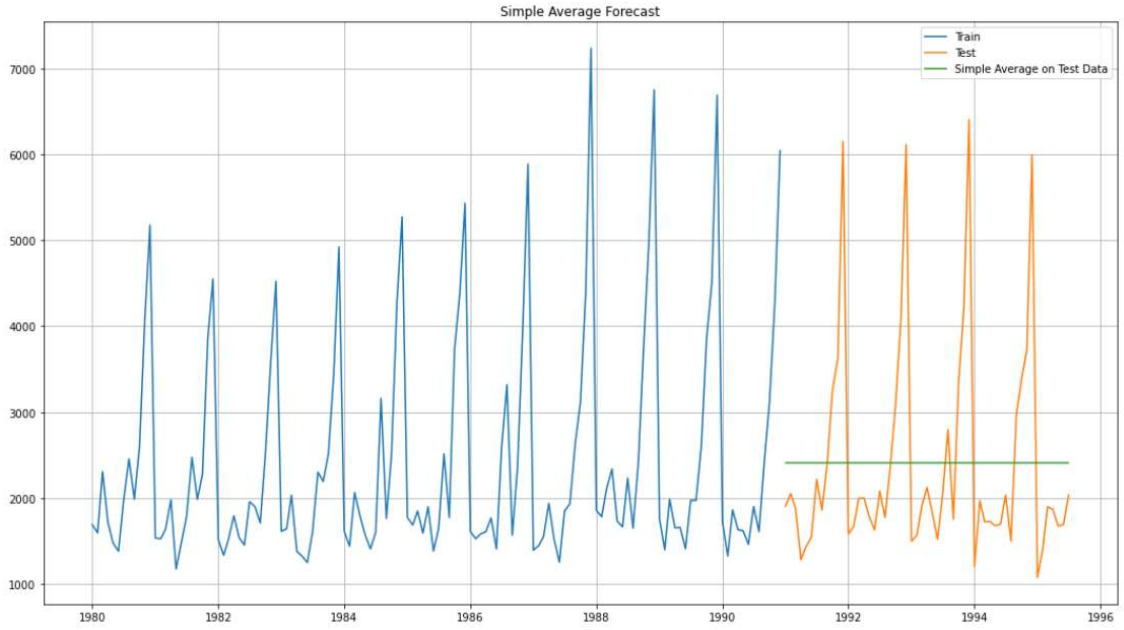


Let's check the model evaluation using RMSE: -

```
For RegressionOnTime forecast on the Test Data,  RMSE is 3864.279
```

Building Simple Average Model: -

In this model the average of the train data becomes forecast for the test data and we will get a flat forecast. Below is the graphical representation of the forecast using this model: -



Below is the model evaluation value using RMSE: -

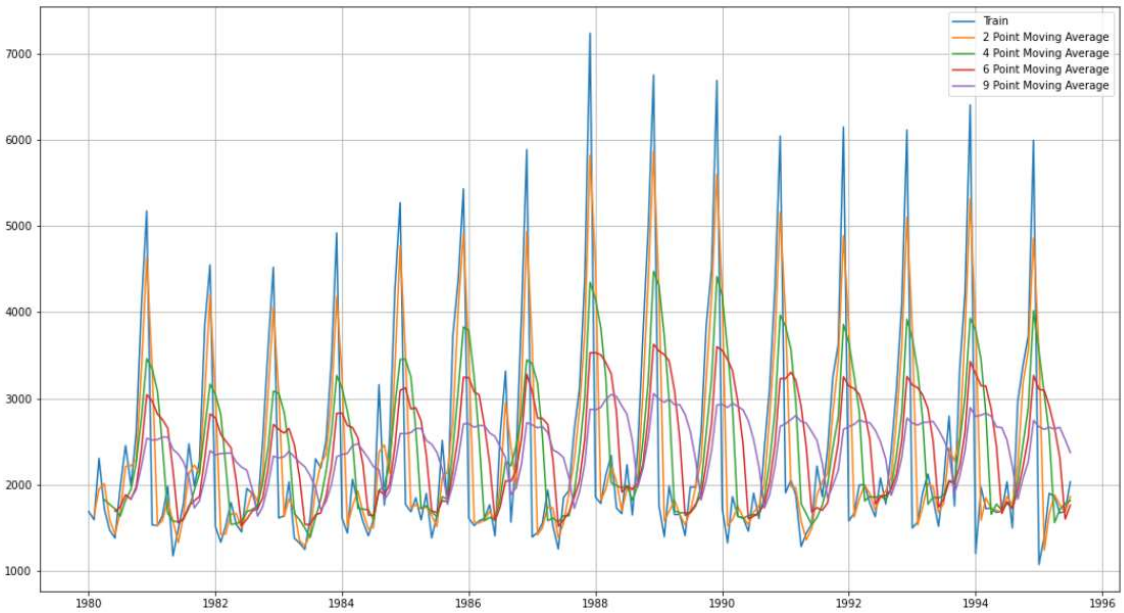
```
For Simple Average forecast on the Test Data,  RMSE is 1275.082
```

Building Moving Average Model: -

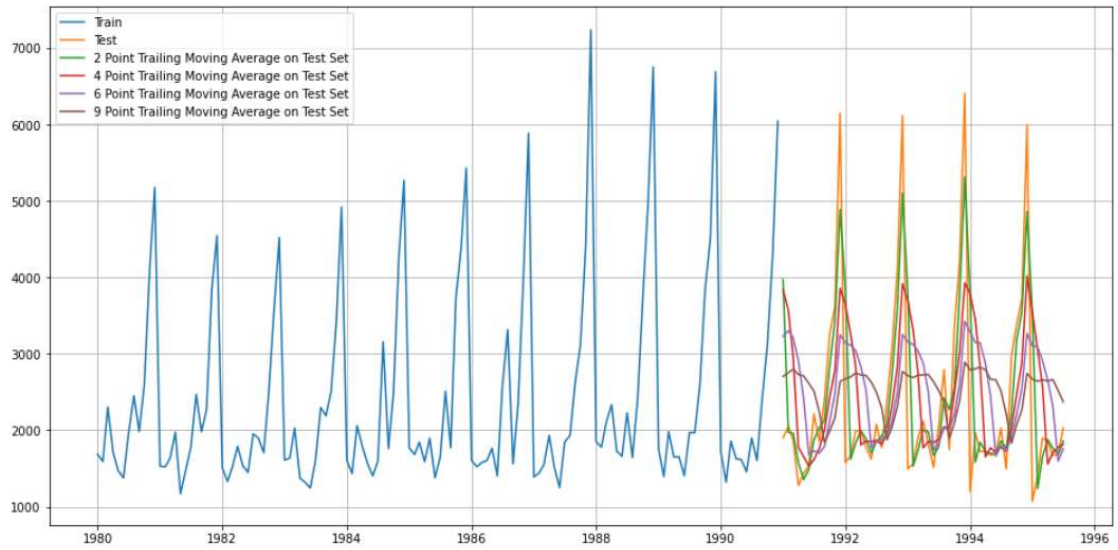
This model creates a cascading window which helps in calculating a rolling mean for different intervals. In this model we cannot use the train or test dataset as the components changes in moving average so we need to consider the complete data for this model. And once the model is built, we can then divide them into training and testing data.

Here are building model with different window size of 2, 4, 6 and 9.

Below plot shows the graphical representation of model built on the whole data.



Now let's divide the data into train and test dataset. Below is the visualization of forecast onto testing dataset.

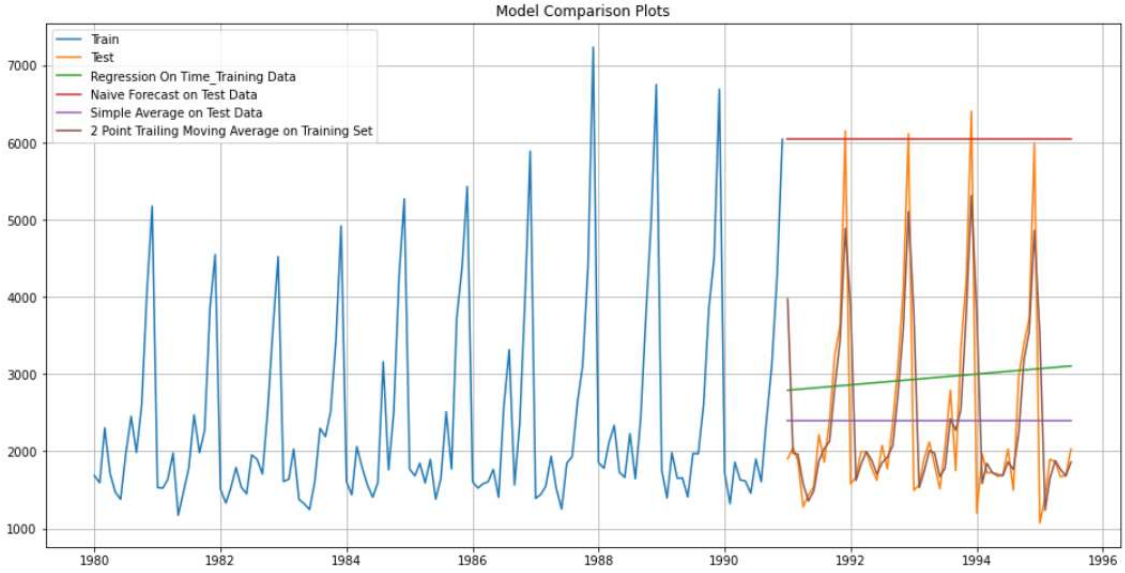


Below are the RMSE scores of the model built with different rolling window.

```
For 2 point Moving Average Model forecast on the Training Data,  RMSE is 813.401
For 4 point Moving Average Model forecast on the Training Data,  RMSE is 1156.590
For 6 point Moving Average Model forecast on the Training Data,  RMSE is 1283.927
For 9 point Moving Average Model forecast on the Training Data,  RMSE is 1346.278
```

From above we can conclude that the moving average with rolling window "2" gives the best forecast with least RMSE value.

Now let's visualize the forecast with moving average and rolling window as "2".



Below is the consolidated RMSE values of all the model built above.

	Test RMSE
RegressionOnTime	1389.135175
NaiveModel	3864.279352
SimpleAverageModel	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315

From above we can conclude that the model built with moving average with rolling window "2" out performs all other models in terms of accuracy with minimal error among other models.

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.

Solution: -

Stationarity is a process of making a time series data stationary throughout the time. That means the data won't be having upward or downward trend. The stationarity of data can be tested using dickey fuller test alongside the required level of significance. If we find that the data is non-stationary, we can have various level of differencing done within the data and stationarity can be achieved. Overall, this process will eliminate the trend part.

We could see visually that there are some elements of trend present in the data. Now let's perform a statistical test to confirm if the series is stationary or not.

The null hypothesis of this test says "the time series is not stationary" and alternate hypothesis would be "time series is stationary".

Ho == Time series is not stationary

Ha == Time series is stationary

We see ta 5% significance as instructed.

Below is the outcome of the test performed: -

```
DF test statistic is -1.798
DF test p-value is 0.7055958459932035
Number of lags used 12
```

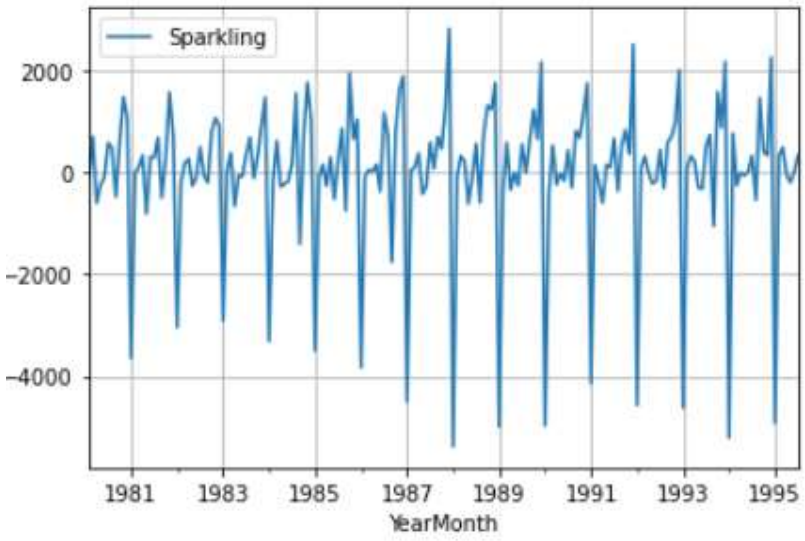
We see that the p value is more than 0.05 and we fail to reject the null hypothesis and conclude that the given time series is not stationary.

In order to make the time series stationary we start with taking a 1st order difference and test again if stationarity is achieved or not.

While performing 1st order differencing we understand that the 1st value will be a NULL value or missing data. Hence, we are dropping the missing value.
Below is the outcome of dickey-fuller test performed after 1st level differencing.

```
DF test statistic is -44.912
DF test p-value is 0.0
Number of lags used 10
```

We notice that this time the P-value is very small and less than 0.05% of significance level. Therefore, the stationarity in data is now achieved.
Now let's visually inspect the differenced time series.



We can see that there is no trend element present in the data and its stationary now.

YearMonth	Sparkling
1980-02-01	-95.0
1980-03-01	713.0
1980-04-01	-592.0
1980-05-01	-241.0
1980-06-01	-94.0
...	...
1995-03-01	495.0
1995-04-01	-35.0
1995-05-01	-192.0
1995-06-01	18.0
1995-07-01	343.0

This is how the difference data looks like and same can used further for analysis and forecasting. The differenced data should be used for building ARIMA or SARIMA models.

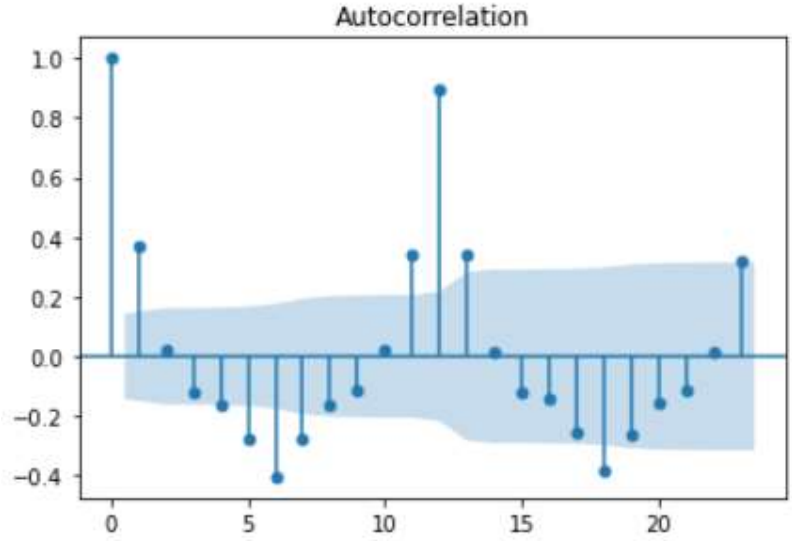
6. Build an automated version of the ARIMA/SARIMA model in which the parameters are selected using the lowest Akaike Information Criteria (AIC) on the training data and evaluate this model on the test data using RMSE.

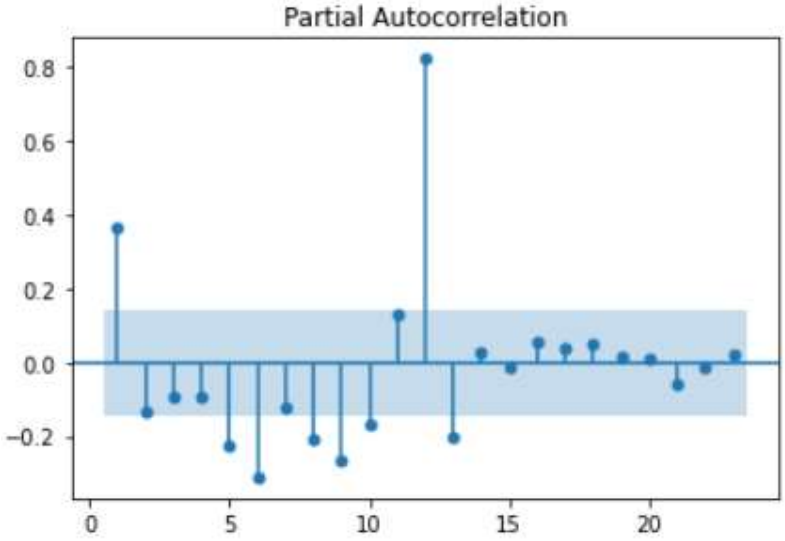
Solution: -

While building ARIMA/SARIMA model's it is recommended at we try and figure out the order of ARIMA (p, d and q) and that can be done by plotting ACF and PACF plot.

ACF does account for the intermediary series where as in PACF the influence of intermediary series is completely discounted.

Below is the ACF and PACF plot for the complete data: -





Now let's discuss a little about the above plots and see how to read and get the cut off points from these plots.

- The ACF plot gives the value of the "q" term, which is the moving average.
- The PACF plot gives us the value of the "p" term.
- The term "d" is the level of differencing considered while performing stationarity in data.
- There are two possibilities of these plots, 1st we may have a cut-off or we may not have a cut-off.

- Lag “0” (zero) is never counted because a series will always have 100% correlation with itself.
- Now let’s count the number of significant lags. The significant lags are the once whose tip points are outside the shaded region. Shaded region is our 95% confidence band.
- The cut-off starts when the tip of lag lies within the shaded region. From above ACF plot we can see that the cut-off will be “1” as 2nd lag is within the shaded region. Value of q = 1.
- From above we know the value of d = 1
- There may be chances at we will not get a proper cut off till the higher order lag. We see a cut off at much later stage. By market practice we account up to 12 lags and if we did not get any cut-off, we conder the value as “0” (zero).
- From the above PACF plot we can see that value of p will be “1”.
- So manually we got all the values which is **p = 1, d = 1 and q = 1**.

Earlier we have divided into train and test split. Let check if we have stationarity in training data set using dickey-fuller test and below are the results: -

```
sparkling test statistic is -2.062
sparkling test p-value is 0.5674110388593696
Number of lags used 12
```

From above output we can conclude that the data is not stationary as P-value is greater than 0.05% level of significance and we fail to reject the null hypothesis of stationarity. we need to make is stationary before building the model. We are trying to get stationarity using 1s order differencing and check for stationarity. Below is the outcome: -

```
sparkling test statistic is -7.968
sparkling test p-value is 8.479210655516242e-11
Number of lags used 11
```

Now we see a change in P-value and its very much less than level of significance of 0.05% and our train data is now stationary and ready for model building.

Building Automated ARIMA Model: -

In this we will try with various combinations of “p”, “d” and “q” terms. which is more like a grid search approach and will pick the best combination based on AIC value. “d” value is fixed at “1” as we see that the data attains stationarity with 1st order differencing and this grid approach we will try with values of “p” and “q” ranging from 1 to 3.

Below is the combination generated over which we will try and fit the ARIMA model: -

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

Now let's fit ARIMA model to all the above combination and get their respective AIC value's to choose the best. Please note that the model with least AIC Value will be considered as the best. Below is the top 5 AIC model sorted in ascending order: -

	param	AIC
10	(2, 1, 2)	2213.509217
15	(3, 1, 3)	2221.451977
14	(3, 1, 2)	2230.757294
11	(2, 1, 3)	2232.983058
9	(2, 1, 1)	2233.777626

From above we can conclude that the model with p, d and q values of 2,1 and 2 respectively gives the least AIC value.

Now let's fit this ARIMA model into train set, below is the output of ARIMA model for train set:

SARIMAX Results

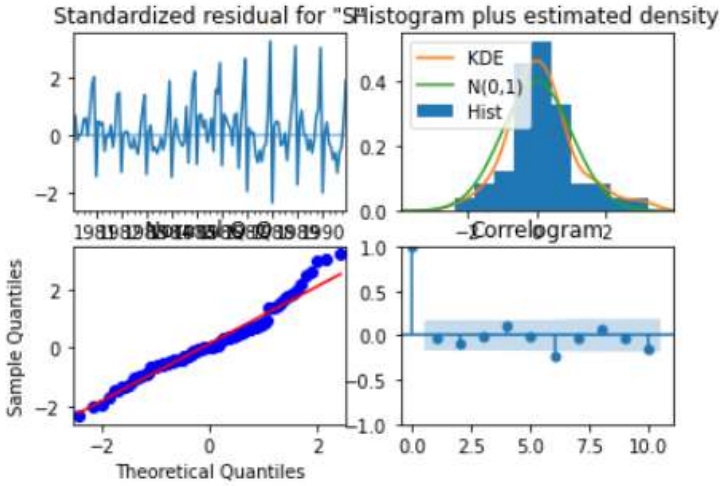
Dep. Variable:	Sparkling	No. Observations:	132
Model:	ARIMA(2, 1, 2)	Log Likelihood	-1101.755
Date:	Tue, 20 Jul 2021	AIC	2213.509
Time:	22:40:03	BIC	2227.885
Sample:	01-01-1980	HQIC	2219.351
	- 12-01-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	1.3121	0.046	28.786	0.000	1.223	1.401
ar.L2	-0.5593	0.072	-7.731	0.000	-0.701	-0.417
ma.L1	-1.9916	0.110	-18.184	0.000	-2.206	-1.777
ma.L2	0.9999	0.110	9.093	0.000	0.784	1.215
sigma2	1.099e+06	2e-07	5.49e+12	0.000	1.1e+06	1.1e+06

Ljung-Box (L1) (Q):	0.19	Jarque-Bera (JB):	14.46
Prob(Q):	0.67	Prob(JB):	0.00
Heteroskedasticity (H):	2.43	Skew:	0.61
Prob(H) (two-sided):	0.00	Kurtosis:	4.08

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 8.97e+27. Standard errors may be unstable.

Below is the diagnostic graph for the same: -



Forecasting on the test data set using the same parameter as train dataset and below is the RMSE score for the same: -

```
RMSE: 1299.980372953183
```

Building Automated SARIMA Model: -

While building SARIMA model there are extra components attached along with the above components of p, d and q. The additional components are below: -

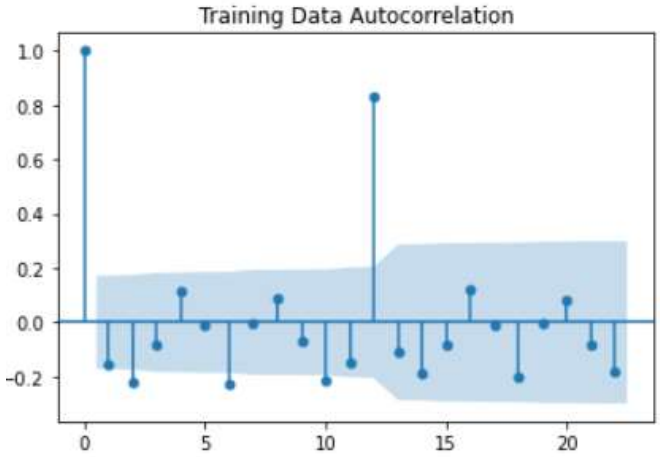
P = AR component with seasonality

D = Differencing for data with seasonality

Q = MA component with seasonality

S = number of lags between the two seasonality components.

to build a model we need to set a mark for the "S" component and same can be decided using ACF plot of training dataset. Below is the ACF plot: -



From above ACF plot we can see that every 4th lag is significant and we can consider that as the value of "S".

To build an automated SARIMA model we will use a grid search approach with range values of p, q, P, Q and S. The values of d = 1 and D = 0 as obtained from level of differencing while performing stationarity.

Below are the combinations obtained from all the above values of parameters: -

```
Model: (0, 1, 1)(0, 0, 1, 4)
Model: (0, 1, 2)(0, 0, 2, 4)
Model: (0, 1, 3)(0, 0, 3, 4)
Model: (1, 1, 0)(1, 0, 0, 4)
Model: (1, 1, 1)(1, 0, 1, 4)
Model: (1, 1, 2)(1, 0, 2, 4)
Model: (1, 1, 3)(1, 0, 3, 4)
Model: (2, 1, 0)(2, 0, 0, 4)
Model: (2, 1, 1)(2, 0, 1, 4)
Model: (2, 1, 2)(2, 0, 2, 4)
Model: (2, 1, 3)(2, 0, 3, 4)
Model: (3, 1, 0)(3, 0, 0, 4)
Model: (3, 1, 1)(3, 0, 1, 4)
Model: (3, 1, 2)(3, 0, 2, 4)
Model: (3, 1, 3)(3, 0, 3, 4)
```

Let's fit these combinations into SARIMAX and obtain their AIC values. The combination with least AIC value will be considered as the best and can be used for prediction on training dataset.

Below are the AIC scores obtained for top 5 combinations: -

	param	seasonal	AIC
63	(0, 1, 3)	(3, 0, 3, 4)	1710.552848
127	(1, 1, 3)	(3, 0, 3, 4)	1711.542457
191	(2, 1, 3)	(3, 0, 3, 4)	1714.121986
255	(3, 1, 3)	(3, 0, 3, 4)	1714.727577
251	(3, 1, 3)	(2, 0, 3, 4)	1714.874679

Form above we can conclude that the parameter values with p = 0, d = 1, q = 3, P = 3, D = 0, Q = 4 and S = 4 outperforms will less AIC scores.

Now we can build a SARIMA model after getting the right values of the parameters and below is the result of building the model: -

SARIMAX Results

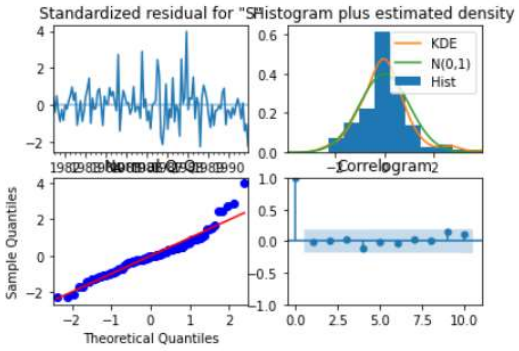
Dep. Variable:	Sparkling	No. Observations:	132
Model:	SARIMAX(0, 1, 3)x(3, 0, 3, 4)	Log Likelihood	-845.276
Date:	Wed, 21 Jul 2021	AIC	1710.553
Time:	17:58:55	BIC	1738.002
Sample:	01-01-1980	HQIC	1721.694
	- 12-01-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.7687	0.100	-7.685	0.000	-0.965	-0.573
ma.L2	-0.1863	0.161	-1.154	0.249	-0.503	0.130
ma.L3	0.1081	0.127	0.854	0.393	-0.140	0.356
ar.S.L4	-0.0034	0.012	-0.295	0.768	-0.026	0.019
ar.S.L8	-0.0236	0.010	-2.439	0.015	-0.043	-0.005
ar.S.L12	1.0406	0.009	117.682	0.000	1.023	1.058
ma.S.L4	-0.1269	0.139	-0.915	0.360	-0.399	0.145
ma.S.L8	-0.1061	0.124	-0.857	0.391	-0.349	0.136
ma.S.L12	-0.7675	0.098	-7.817	0.000	-0.960	-0.575
sigma2	1.271e+05	1.9e-06	6.7e+10	0.000	1.27e+05	1.27e+05

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	40.06
Prob(Q):	0.95	Prob(JB):	0.00
Heteroskedasticity (H):	2.75	Skew:	0.82
Prob(H) (two-sided):	0.00	Kurtosis:	5.37

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 3.61e+26. Standard errors may be unstable.

Below is the diagnostic plot for the same: -



With the same model build above we can now perform forecast in test dataset and obtain its RMSE value: -

```
RMSE: 564.9245400909168
```

The top 5 forecasts are shown below with 95% confidence interval: -

Sparkling	mean	mean_se	mean_ci_lower	mean_ci_upper
1991-01-01	1435.653654	362.183892	725.786269	2145.521039
1991-02-01	1311.450951	371.760128	582.814488	2040.087413
1991-03-01	1657.997337	372.109091	928.676922	2387.317753
1991-04-01	1571.799568	376.241363	834.380047	2309.219088
1991-05-01	1382.238793	376.671598	643.976027	2120.501559

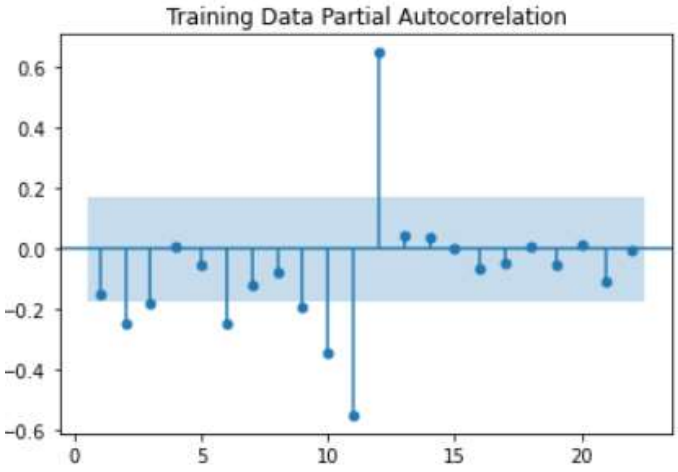
7. Build ARIMA/SARIMA models based on the cut-off points of ACF and PACF on the training data and evaluate this model on the test data using RMSE.

Solution: -

Building ARIMA model basis cut-off obtained from ACF and PACF plot: -

Earlier we have discussed about how to read the ACF and PACF plot for the overall data. Now we will plot the same ACF and PACF plot for training data and determine the cut-offs for p, d and q.

Below are the ACF and PACF plots for training data: -



From plots as shown above indicates that the value of p = 0 and q = 0 and we know the value of d = 1 using which we have made the training data stationary. Now lets fit the training data with these values and below is the output: -

SARIMAX Results

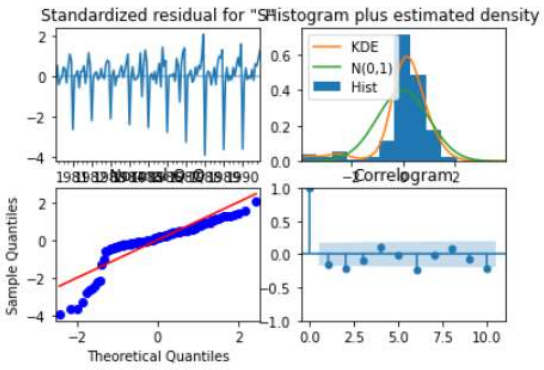
Dep. Variable:	Sparkling	No. Observations:	132
Model:	ARIMA(0, 1, 0)	Log Likelihood	-1132.832
Date:	Wed, 21 Jul 2021	AIC	2267.663
Time:	00:36:32	BIC	2270.538
Sample:	01-01-1980 - 12-01-1990	HQIC	2268.831
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
sigma2	1.885e+06	1.29e+05	14.658	0.000	1.63e+06	2.14e+06

Ljung-Box (L1) (Q):	3.07	Jarque-Bera (JB):	198.83
Prob(Q):	0.08	Prob(JB):	0.00
Heteroskedasticity (H):	2.46	Skew:	-1.92
Prob(H) (two-sided):	0.00	Kurtosis:	7.65

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

Below is the diagnostic plot of the model built: -

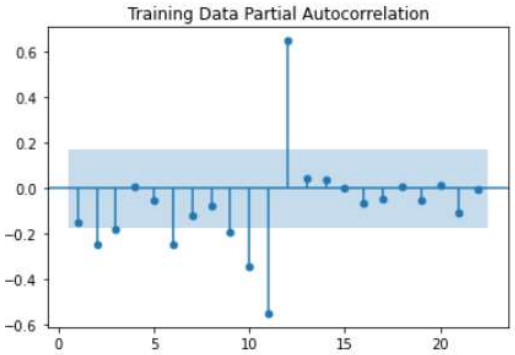


Now as the model is built, we can use the same for prediction on to testing dataset and check on its RMSE value. Below is the RMSE value obtained out of this model: -

```
RMSE: 3864.2793518443914
```

Building SARIMA model basis cut-off obtained from ACF and PACF plot: -

To build a SARIMA model we need to obtain the value of "P" and "Q" in a seasonal way from the below plots: -



- From above manually built ARIMA model we know that the values of p, d and q as 0,1 and 0 respectively.
- From the above ACF plot we can see that every 4th lag is significant and we can conclude the value of Q as 4.
- From the above PACF plot we can see that every 3rd lag is significant can we can conclude the value of P as 3.
- Value of D will be "0" as we are not performing and differencing and value of as will be 4 as shown above.
  Now will fit the above obtained parameter from plots to build SARIMA model on train dataset and below is the output: -

SARIMAX Results

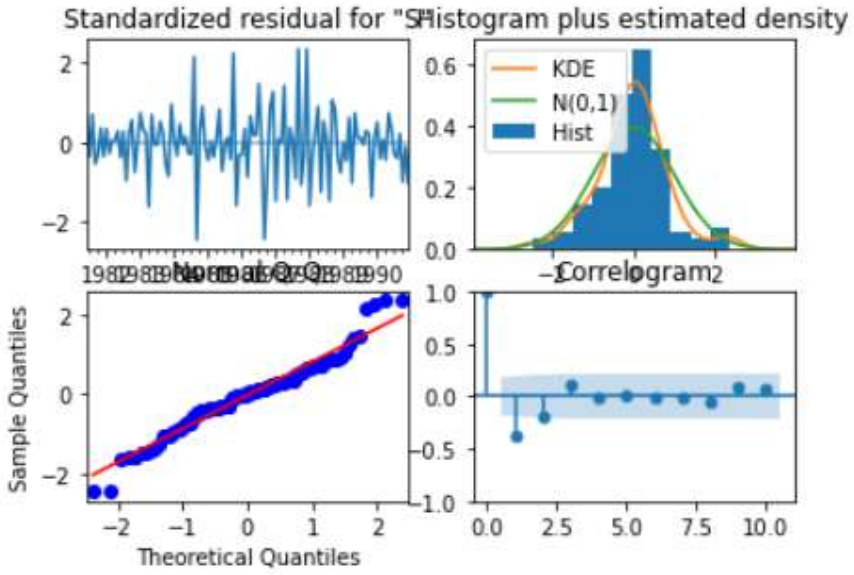
Dep. Variable:	Sparkling	No. Observations:	132
Model:	SARIMAX(0, 1, 0)x(4, 0, [1, 2, 3], 4)	Log Likelihood	-873.204
Date:	Wed, 21 Jul 2021	AIC	1762.408
Time:	19:08:11	BIC	1784.368
Sample:	01-01-1980	HQIC	1771.321
	- 12-01-1990		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.S.L4	-0.1454	0.197	-0.739	0.460	-0.531	0.240
ar.S.L8	-0.0377	0.020	-1.898	0.058	-0.077	0.001
ar.S.L12	1.0470	0.028	37.073	0.000	0.992	1.102
ar.S.L16	0.1609	0.204	0.788	0.430	-0.239	0.561
ma.S.L4	-0.0619	0.222	-0.279	0.781	-0.497	0.374
ma.S.L8	-0.0993	0.135	-0.736	0.462	-0.364	0.165
ma.S.L12	-0.7004	0.156	-4.477	0.000	-1.007	-0.394
sigma2	2.883e+05	5.81e+04	4.960	0.000	1.74e+05	4.02e+05

Ljung-Box (L1) (Q):	16.65	Jarque-Bera (JB):	7.59
Prob(Q):	0.00	Prob(JB):	0.02
Heteroskedasticity (H):	2.21	Skew:	0.07
Prob(H) (two-sided):	0.02	Kurtosis:	4.25

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

Below is the diagnostic plot of the same: -



We can perform prediction on test dataset using the model built above and check on its RMSE score: -

RMSE: 1286.3607474950072

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.

Solution: -

	Test RMSE
Single ES	1338.012144
Double ES	3949.993290
Triple ES - Additive	379.695686
Triple ES - Multiplicative	406.510170
RegressionOnTime	1389.135175
NaiveModel	3864.279352
SimpleAverageModel	1275.081804
2pointTrailingMovingAverage	813.400684
4pointTrailingMovingAverage	1156.589694
6pointTrailingMovingAverage	1283.927428
9pointTrailingMovingAverage	1346.278315
ARIMA(2,1,2) - Auto	1299.980373
SARIMA(0,1,3)(3,0,3,4) - Auto	564.924540
ARIMA(0,1,0)- Manual	3864.279352
SARIMA(0,1,0)(4,0,3,4) - Manual	1286.360747

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.

Solution: -

From above data-frame we can conclude that the best optimized model has the values of 0, 1, 3, 3, 0, 3, 4 for p, d, q, P, D, Q and S respectively as it gives the least RMSE score among all the models. We can use the same values of parameters for the complete dataset and output is shown below: -

```
                                SARIMAX Results
==========================================================================================
Dep. Variable:                           Sparkling   No. Observations:                  187
Model:             SARIMAX(0, 1, 3)x(3, 0, 3, 4)   Log Likelihood               -1247.533
Date:                           Wed, 21 Jul 2021   AIC                           2515.066
Time:                                   21:12:40   BIC                           2546.424
Sample:                             01-01-1980   HQIC                          2527.790
                                  - 07-01-1995
Covariance Type:                            opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -0.8199      0.076    -10.849      0.000      -0.968      -0.672
ma.L2         -0.1512      0.117     -1.293      0.196      -0.380       0.078
ma.L3          0.0668      0.094      0.711      0.477      -0.117       0.251
ar.S.L4        0.0012      0.011      0.108      0.914      -0.020       0.022
ar.S.L8       -0.0091      0.008     -1.202      0.229      -0.024       0.006
ar.S.L12       1.0116      0.009    108.320      0.000       0.993       1.030
ma.S.L4       -0.1920      0.103     -1.863      0.062      -0.394       0.010
ma.S.L8       -0.1834      0.100     -1.834      0.067      -0.379       0.013
ma.S.L12      -0.7012      0.087     -8.074      0.000      -0.871      -0.531
sigma2      1.226e+05   1.64e+04      7.486      0.000    9.05e+04    1.55e+05
==========================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                64.15
Prob(Q):                              0.96   Prob(JB):                         0.00
Heteroskedasticity (H):               1.39   Skew:                             0.81
Prob(H) (two-sided):                  0.22   Kurtosis:                         5.54
==========================================================================================

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

Predicting for 12 months into the future: -

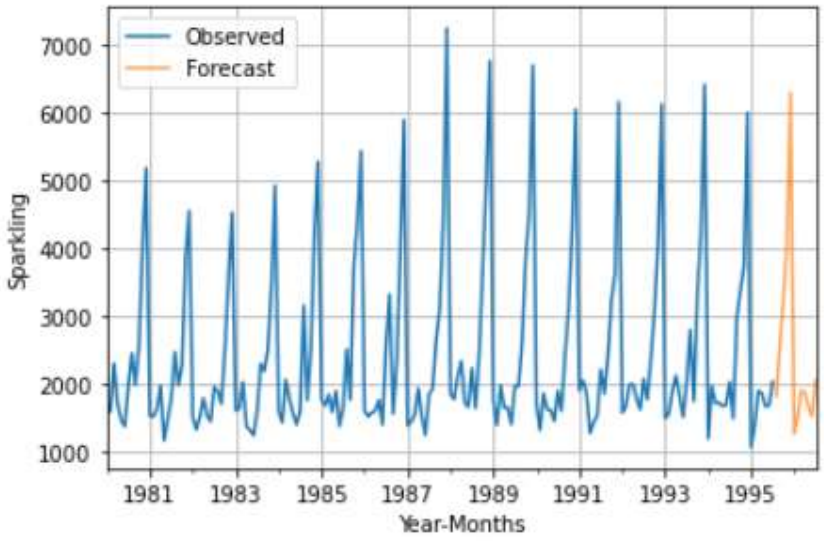
The forecast for 12 months into the future is as below: -

Sparkling	mean	mean_se	mean_ci_lower	mean_ci_upper
1995-08-01	1823.680700	361.616287	1114.925800	2532.435599
1995-09-01	2533.134169	367.439083	1812.966799	3253.301539
1995-10-01	3236.379330	367.573919	2515.947686	3956.810973
1995-11-01	4079.101814	369.204629	3355.474039	4802.729589
1995-12-01	6291.856778	369.371093	5567.902738	7015.810818
1996-01-01	1269.892214	370.302726	544.112208	1995.672220
1996-02-01	1506.922059	371.792373	778.222399	2235.621719
1996-03-01	1908.648694	373.021956	1177.539096	2639.758293
1996-04-01	1868.479086	373.553288	1136.328095	2600.630076
1996-05-01	1670.792491	374.144943	937.481877	2404.103105
1996-06-01	1520.028361	375.245791	784.560125	2255.496597
1996-07-01	2051.529824	376.106368	1314.374888	2788.684759

The RMSE score of this forecast is below: -

```
RMSE of the Full Model 530.5186668518359
```

The graphical representation of forecast is as below: -



Apart from SARIMA model we have also noticed that the that triple exponential smoothing also performed well in terms of RMSE score and same can be used for prediction.

Below is the forecast for 12 months in future: -

```
1995-08-01    1877.418973
1995-09-01    2405.272289
1995-10-01    3242.091582
1995-11-01    3922.174721
1995-12-01    6118.486885
1996-01-01    1262.602775
1996-02-01    1592.120997
1996-03-01    1831.635313
1996-04-01    1806.451718
1996-05-01    1651.704099
1996-06-01    1586.487882
1996-07-01    1976.989421
Freq: MS, dtype: float64
```

We do understand that modelling is an iterative process and trying various other combinations of values may lead to better RMSE score and better model.

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

Please explain and summarise the various steps performed in this project. There should be proper business interpretation and actionable insights present.

Solution: -

From all the above model built we have concluded the SARIMA model with p,d,q,P,D,Q and S values as 1,0,2,3,0,3 and 4 respectively as it gives less RMSE score when compared with other models which indicates that this model looses the least data ana captures the maximum for modelling.

Also, triple exponential model works well will minimal difference in RMSE score and can be used for future forecasting.

Various Steps performed in this project: -

- Plotting data and understand about trend and seasonality about the data
- Multiplicative and additive decomposition of data.
- Splitting data into training and test dataset.
- Performing exponential smoothing on train dataset and forecasting on test dataset and evaluating the model using RMSE scores.
- Building Linear regression, Naïve base, simple average model and moving average model on training dataset and forecasting on test dataset, preforming model evaluation using RMSE score.
- Checking on stationarity using dickey-fuller test and performing stationarity using level of differencing.
- Building automated ARIMA and SARIMA model over the stationary data, using grid search approach and selecting the best using AIC score. The model with least AIC score considered the best. Forecasting on test data using the model with best p,d and q values with least AIC score and perform model evaluation using RMSE score.
- Manually building ARIMA and SARIMA model but deciding on p,d and q values basis cut offs from ACF and PACF plot. Fitting these cut-offs on training dataset and forecasting on test dataset and evaluating performance using RMSE score.
- From all the model built above selecting the best model with least RMSE score and which accounts best for the trend and seasonality component in the data.
- Building the most optimal model on the complete dataset and forecasting into the future for 12 months and performing model evaluation using RMSE score.
- Finally plotting the data provided along with the forecast.

Insight from data: -

From the above exploratory data analysis we can conclude that the data definitely have a seasonality component but its very difficult to comment on the trend of it. There is no positive or negative trend observed. Which implies that the sales across the years remains almost the same and indicates that the sales growth is stagnant.

Business Recommendations: -

- Business can improve on its visibility across the wine stores.
- A drive towards digital marketing should help in increasing the sales.
- A separate desk for in-store testing for customers before they make the purchase.

- Enhancing the knowledge of the sales staff and updating them with the latest knowledge and also about the competitors.
- Pricing discount during festive seasons.

----------END----------