BTech/III Year CSE/VI Semester

19CSE352/Business Analytics

Case Study Report

Roll.No	Name
Jaswanth Pasumarthy	CB.EN.U4CSE21227
Ajay Kumar	CB.EN.U4CSE21236
N Sandeep	CB.EN.U4CSE21244
Samarth P	CB.EN.U4CSE21253

Abstract

In today's dynamic business environment, efficient inventory management is crucial for organizations to meet customer demands while minimizing costs and maximizing profitability. However, many businesses struggle with challenges such as overstocking, stockouts, and inefficient replenishment processes, leading to decreased customer satisfaction and increased operational expenses.

This report presents a comprehensive analysis of inventory management using advanced data analytics techniques applied to the Sample Superstore dataset. The objective is to derive actionable insights and recommendations that can help businesses optimize their inventory management strategies.

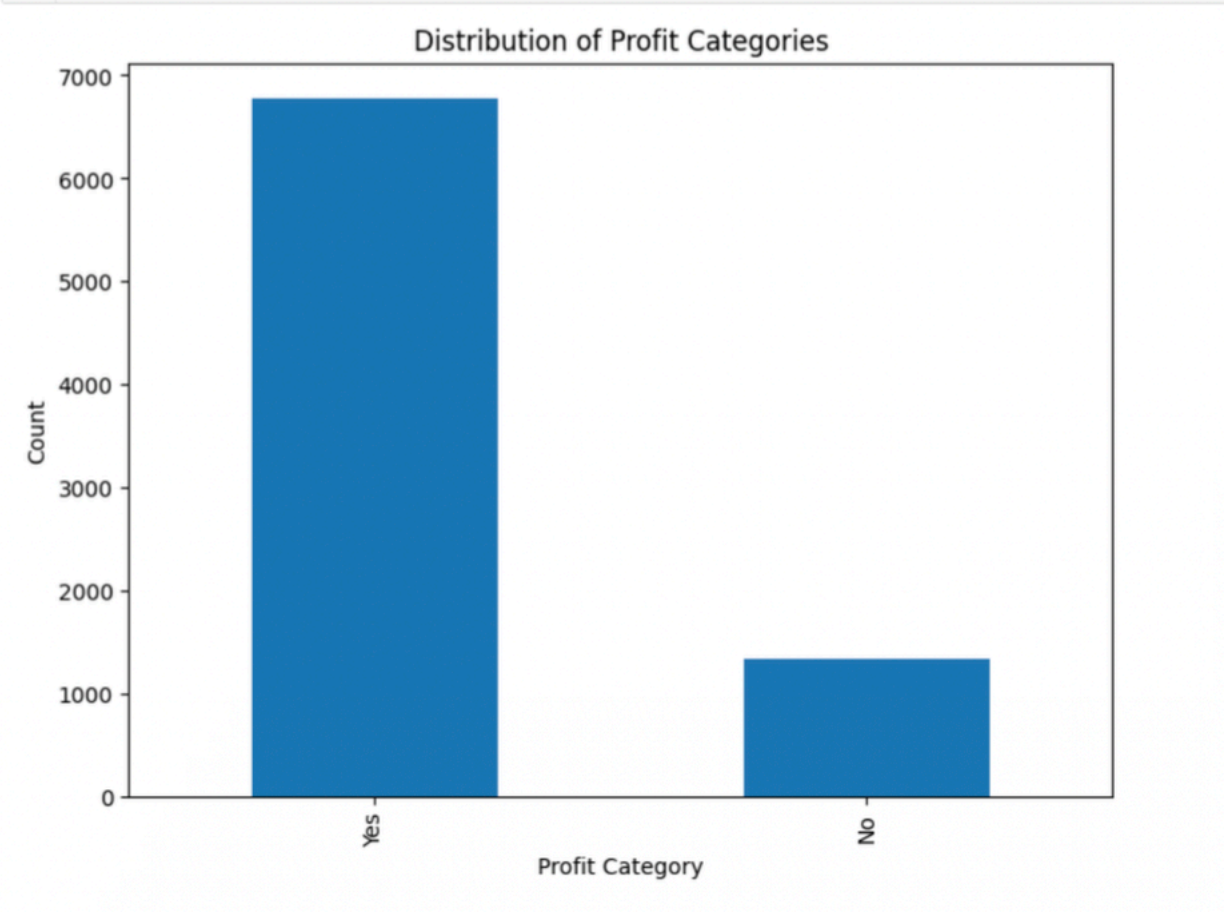
Data Preprocessing

In the preprocessing phase, several key steps were undertaken to prepare the data for analysis. The process began with calculating the first quartile (Q1) and third quartile (Q3) for the 'Profit' column, which helped in understanding the distribution and spread of profit values.

Using these quartiles, the Interquartile Range (IQR) was computed, providing a measure of the data's variability. By establishing lower and upper bounds based on 1.5 times the IQR below Q1 and above Q3, potential outliers were identified within the 'Profit' data.

Visualizing the data distribution, a boxplot was created, offering a graphical representation of the profit values and highlighting any outliers that fell beyond the computed bounds. Additionally, a five-number summary including the minimum, Q1, median, Q3, and maximum values of the 'Profit' column was generated, providing a comprehensive overview of the data's central tendency and extreme values.

Furthermore, a new column named 'Profit_Category' was introduced to categorize profits as either 'Yes' (indicating positive profits) or 'No' (indicating negative profits or losses). This categorization facilitated a deeper analysis of profit trends and patterns within the dataset.



Overall, these preprocessing steps laid the groundwork for subsequent analyses and insights into inventory management dvnamics.

Apriori

The Apriori algorithm was applied twice to analyze the relationships between different factors and their impact on profitability. In the first analysis, transactions were created based on 'State', 'Category', and
'Profit_Category', extracting association rules that reveal the connection between states, product categories, and profitability levels.

Similarly, in the second analysis, transactions were formed with 'State', 'Sub-Category', and 'Profit_Category', uncovering insights into the relationships between states, product sub-categories, and profitability categories. These association rules provide valuable insights into which product categories or sub-categories sold in each state are likely to result in either profit or loss, aiding in strategic decision-making for inventory management.

Reduction Techniques

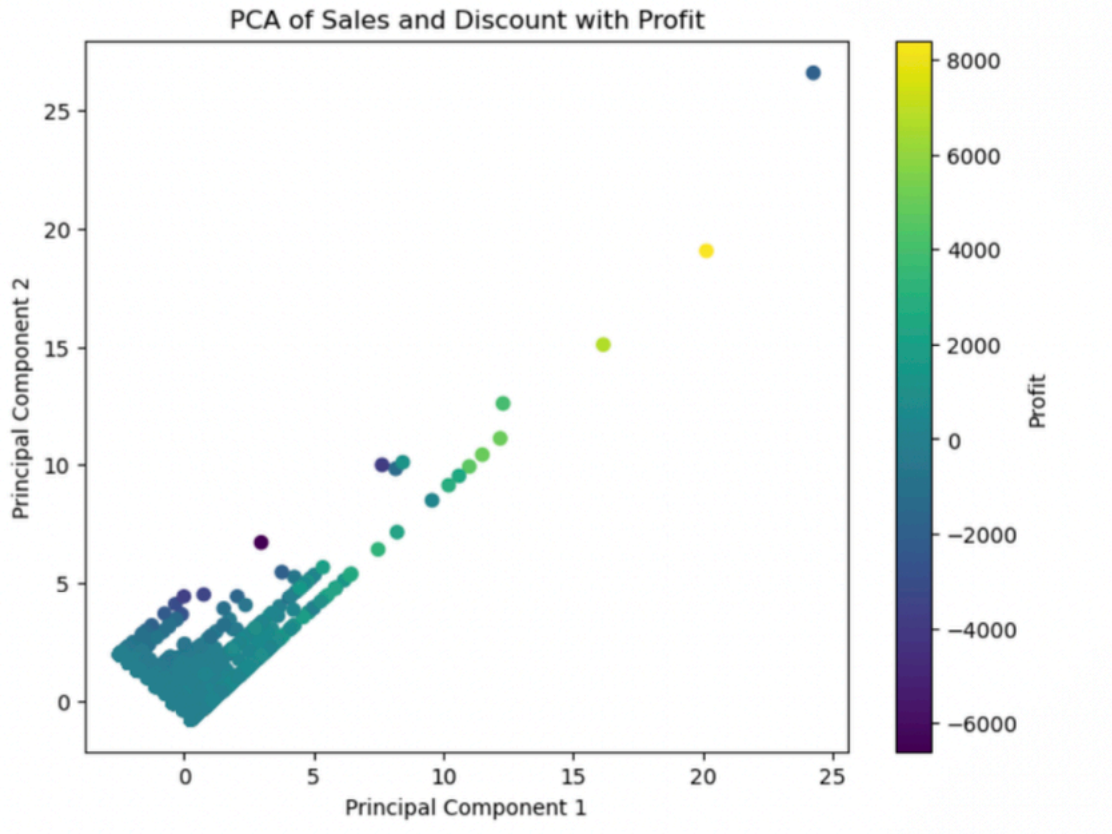
PCA

These features were standardized using the StandardScaler to ensure that they were on the same scale, which is a common preprocessing step in PCA to prevent features with larger scales from dominating the analysis.

After standardization, PCA was applied with the aim of reducing the dimensionality of the data while preserving its variance. In this case, PCA was set to reduce the features to two principal components ('Principal Component 1' and 'Principal Component 2').

The PCA result, represented by the transformed data points in the reduced-dimensional space, was then plotted against the target variable 'Profit'. Each data point in the scatter plot corresponds to a sample in the dataset, with its position determined by the values of the two principal components.

The color of each point in the scatter plot is determined by the 'Profit' variable, represented by the color bar alongside the plot. This visualization helps to understand if there's any discernible pattern or clustering in the data points
based on the sales, discounts, and resulting profits, offering insights into potential relationships or trends in the dataset.



LDA

In the Linear Discriminant Analysis (LDA) implementation for the dataset, the objective was to uncover meaningful relationships between features such as 'Sales' and 'Discount' and the target variable ‘Profit_Category.'
By transforming the data into a lower-dimensional space using LDA, the aim was to maximize class separability and identify linear combinations of features that distinguish between positive and negative profit categories.
The resulting analysis provided insights into which combinations of sales and discounts are likely to lead to profitable outcomes, contributing to strategic decision-making in inventory management and business operations. The accuracy was
Accuracy: 0.9155884165126309

Collaborative Filtering

The function collaborative filtering was designed to provide personalized product recommendations for a given customer ID based on collaborative filtering techniques. Specifically, the function retrieves orders associated with the customer, sorts them by sales volume in descending order, and selects the top 5 products as recommended items. For example, using the customer ID 'CG-12520', the function generates a list of recommended products tailored to that specific customer.

The recommended products for Customer CG-12520 based on Collaborative Filtering, listed in order:

Hon Deluxe Fabric Upholstered Stacking Chairs, Rounded Back
Bush Somerset Collection Bookcase
C-Line Cubicle Keepers Polyproplyene Holder w/Velcro Back, 8-1/2x11, 25/Bx
SimpliFile Personal File, Black Granite, 15w x 6-15/16d x 11-1/4h
Xerox 1986

Content Filtering

The function 'content based filtering was designed to generate product recommendations based on content-based filtering techniques, specifically focusing on a given product category. It employs TF-IDF vectorization to transform the textual data of product categories into numerical vectors and computes the cosine similarity between products' category vectors to identify similar products.

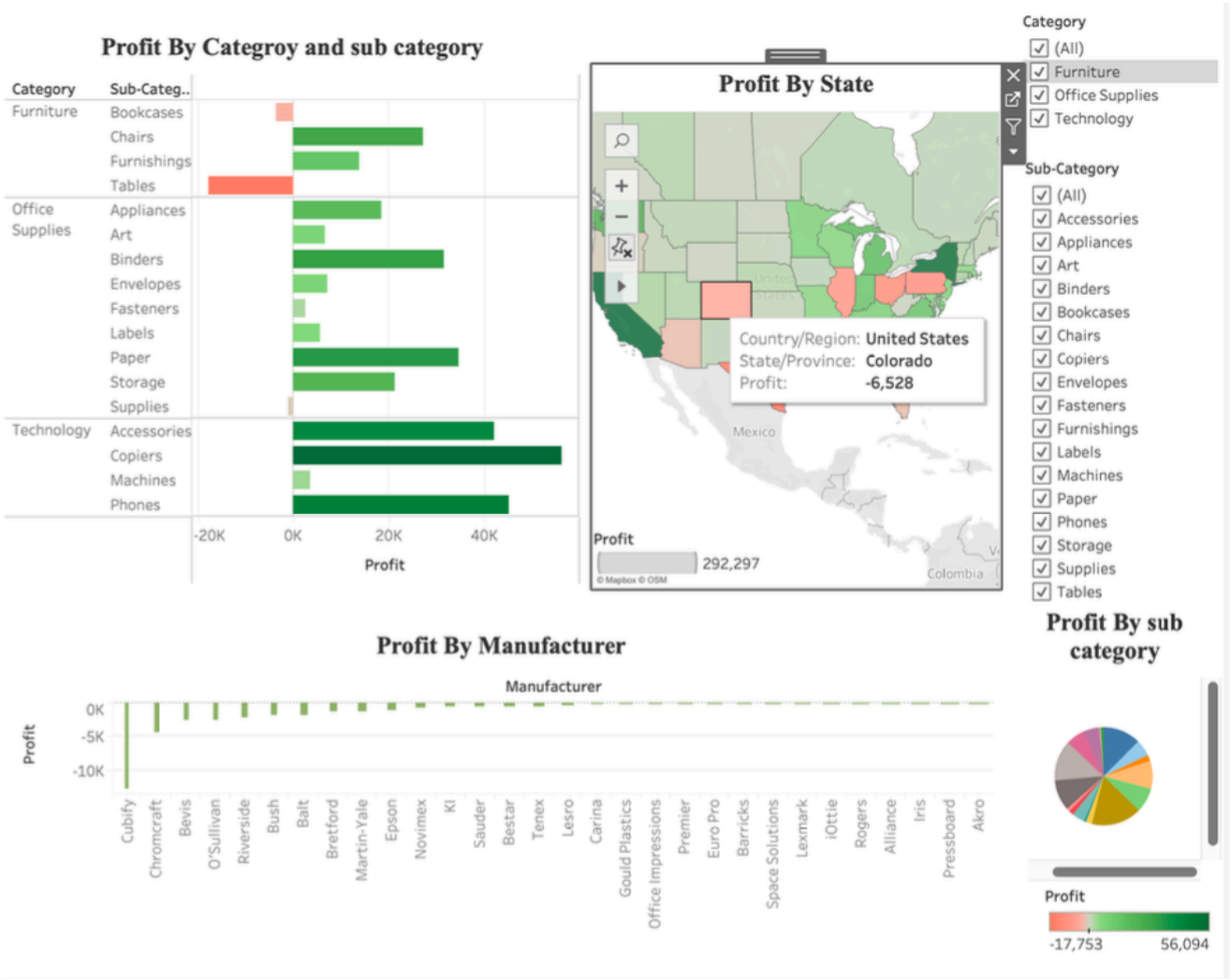
The recommended products are :-

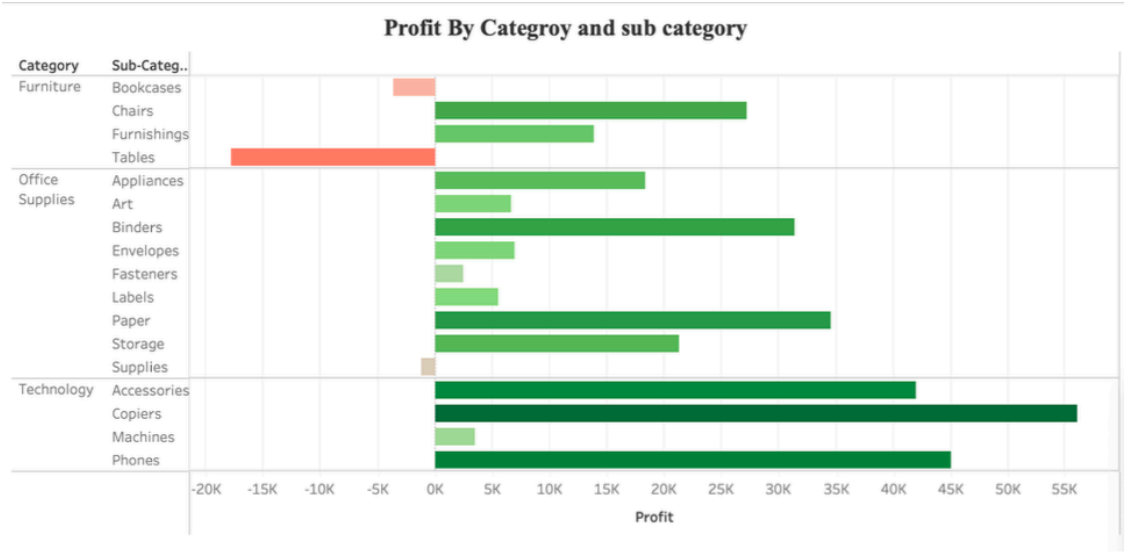
Hon Deluxe Fabric Upholstered Stacking Chairs, Rounded Back
Chromcraft Rectangular Conference Tables
Bretford CR4500 Series Slim Rectangular Table
Eldon Expressions Wood and Plastic Desk Accessories, Cherry Wood
Global Deluxe Stacking Chair, Gray

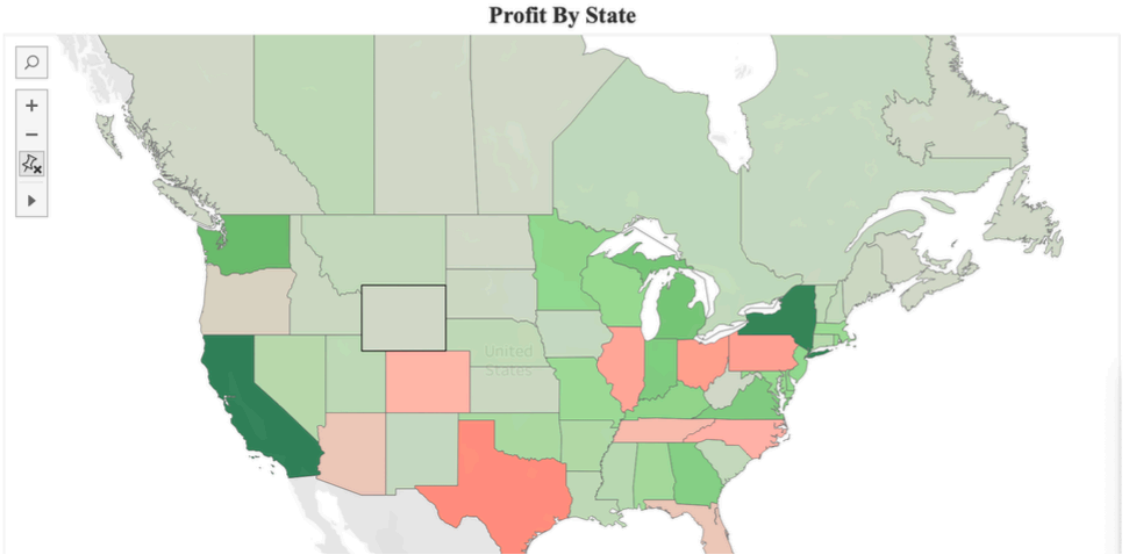
These products are recommended based on content-based filtering techniques focused on the 'Furniture' category.

Tableau Plots:

Dashboard




Profit By Categroy and sub category
Category
Sub-Categ..
Furniture
Bookcases
Chairs
Furnishings
Tables
Office Supplies
Appliances
Art
Binders
Envelopes
Fasteners
Labels
Paper
Storage
Supplies
Technology
Accessories
Copiers
Machines
Phones
-20K
-15K
-10K
-5K
0K
5K
10K
15K
20K
25K
30K
35K
40K
45K
50K
55K
Profit

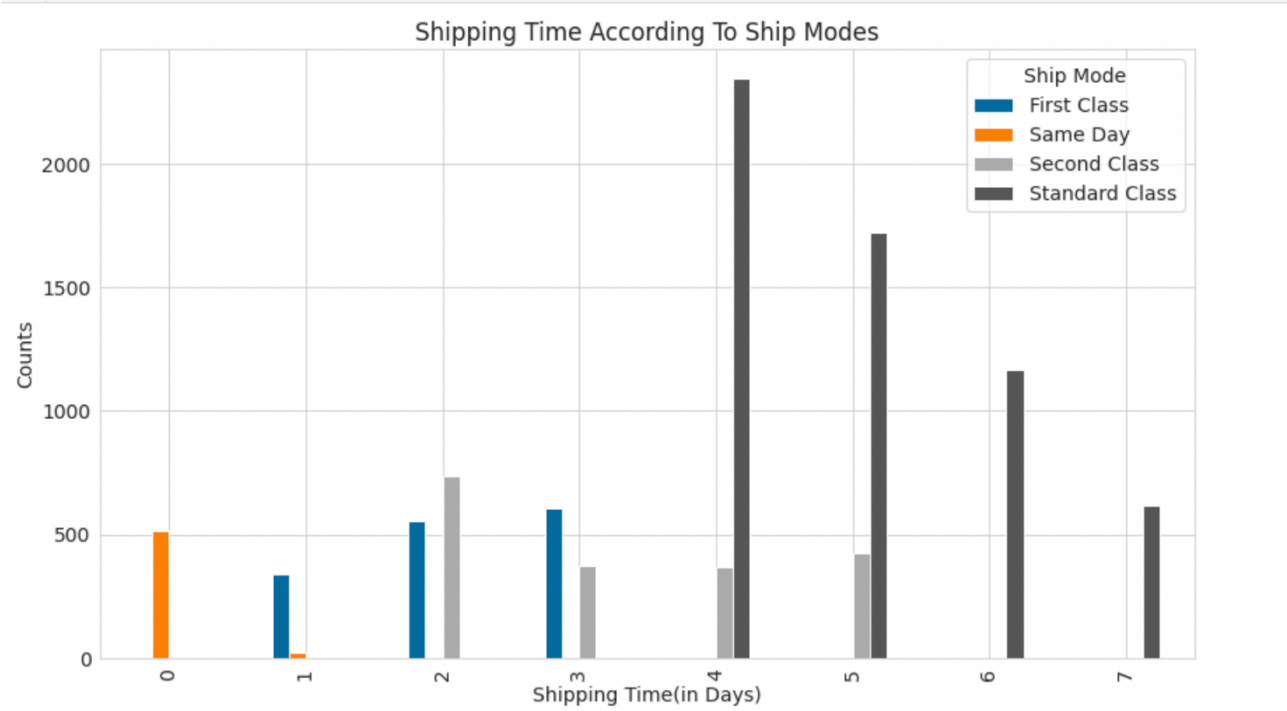

Profit By State
United States

TIME SERIES ANALYSIS

A time series is a sequence of data points or observations recorded at specific time intervals, typically equidistant. It's used to analyze trends, patterns, and behavior over time, revealing insights into how a variable changes over a period. Time series data often includes measurements like stock prices, weather conditions, sales figures, or sensor readings, with each data point timestamped. Statistical methods such as forecasting, trend analysis, and seasonal adjustment are applied to time series to extract

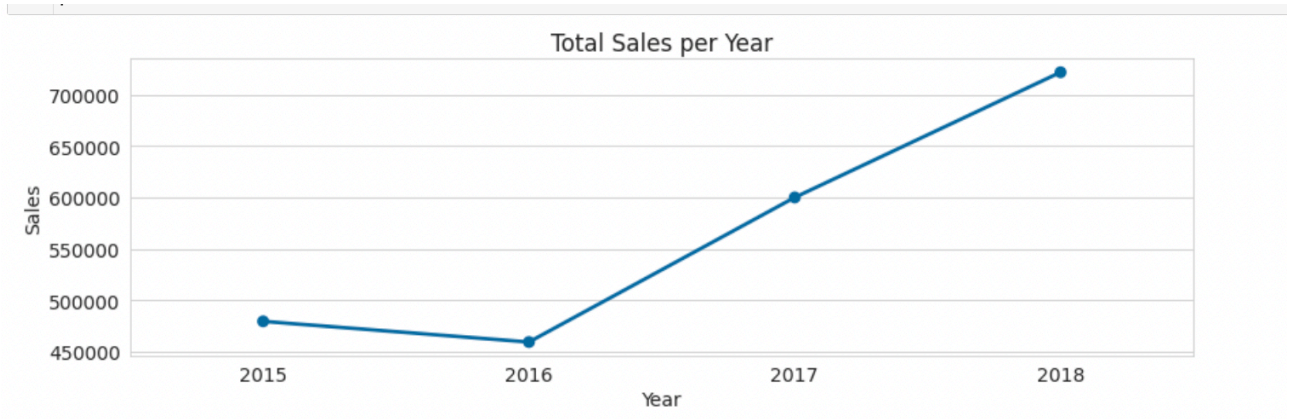
1. Visualisation

Shipping Time observation according to ShipModes.



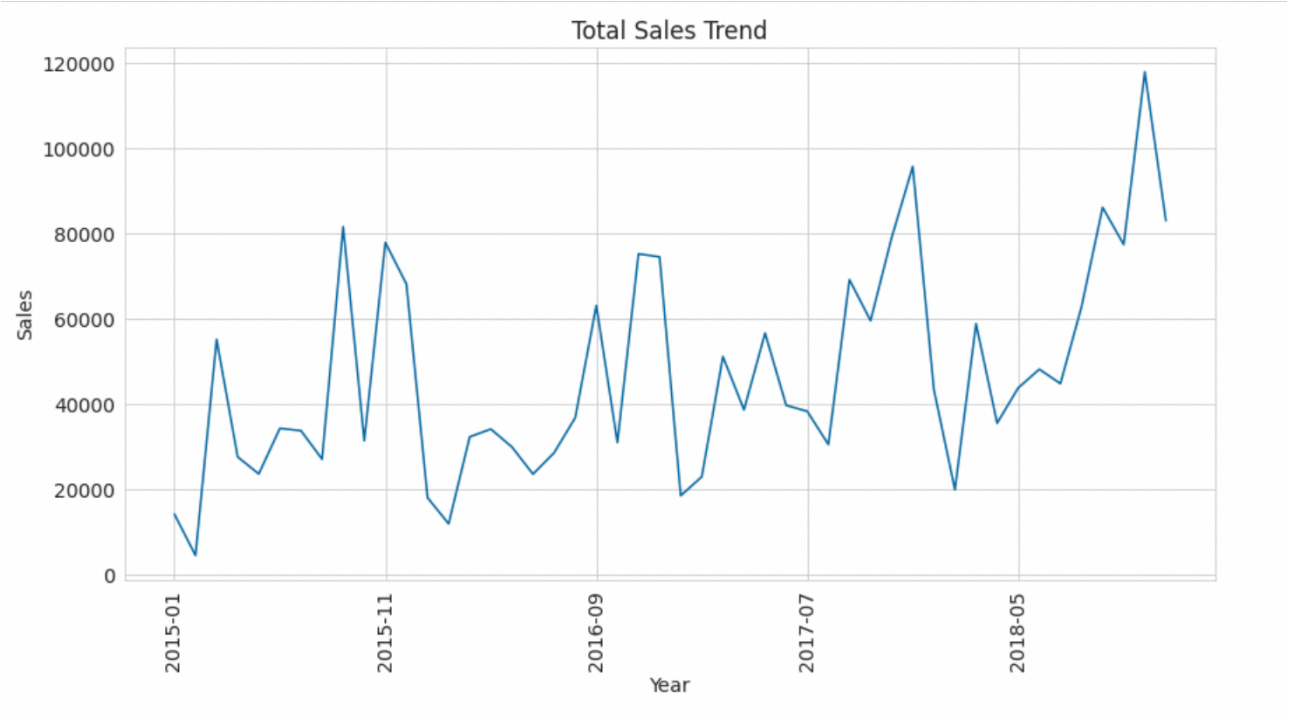
shipping data by ship modes and shipping times, counts the number of orders, unstacks the data for plotting, creates a bar chart showing order counts per mode and time, and labels the axes and title for clarity.

Trends and Seasonality



Line Plot

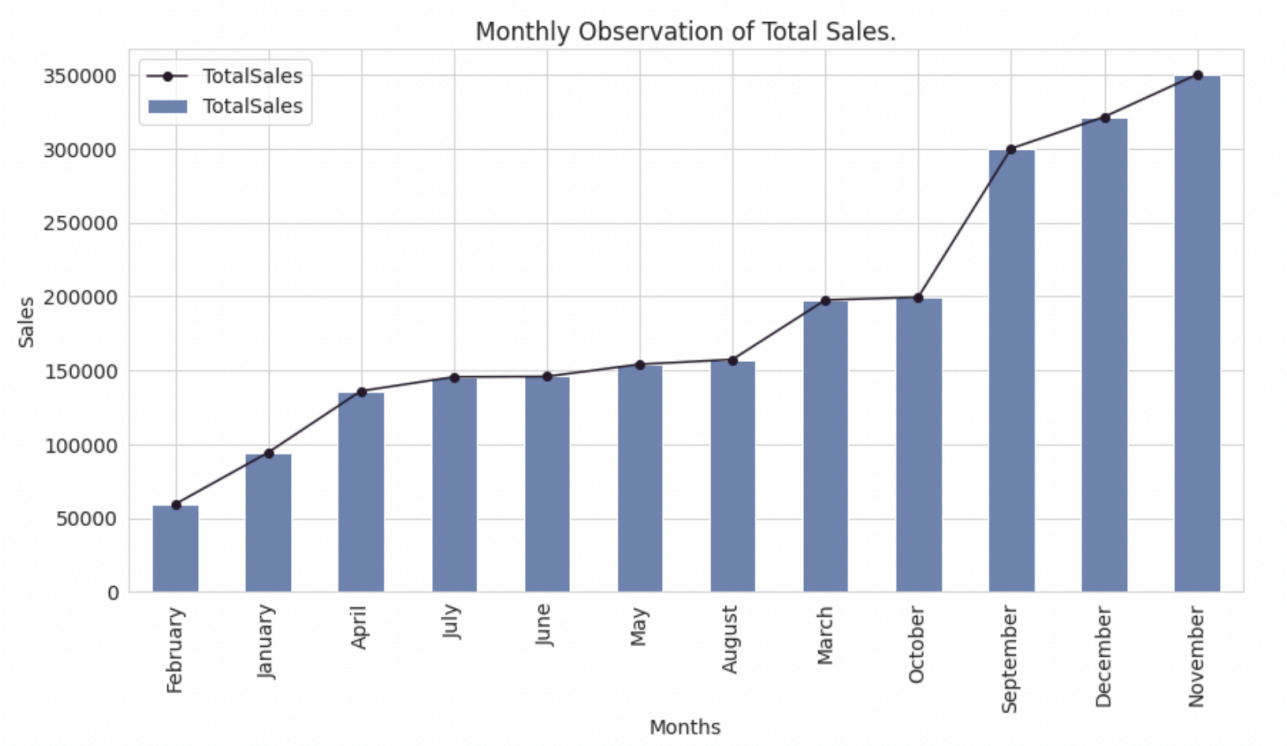
Year-to-Year observation of TotalSales



There is increasing trends or growth in sales over time. There may be seasonality to the sales for each year.

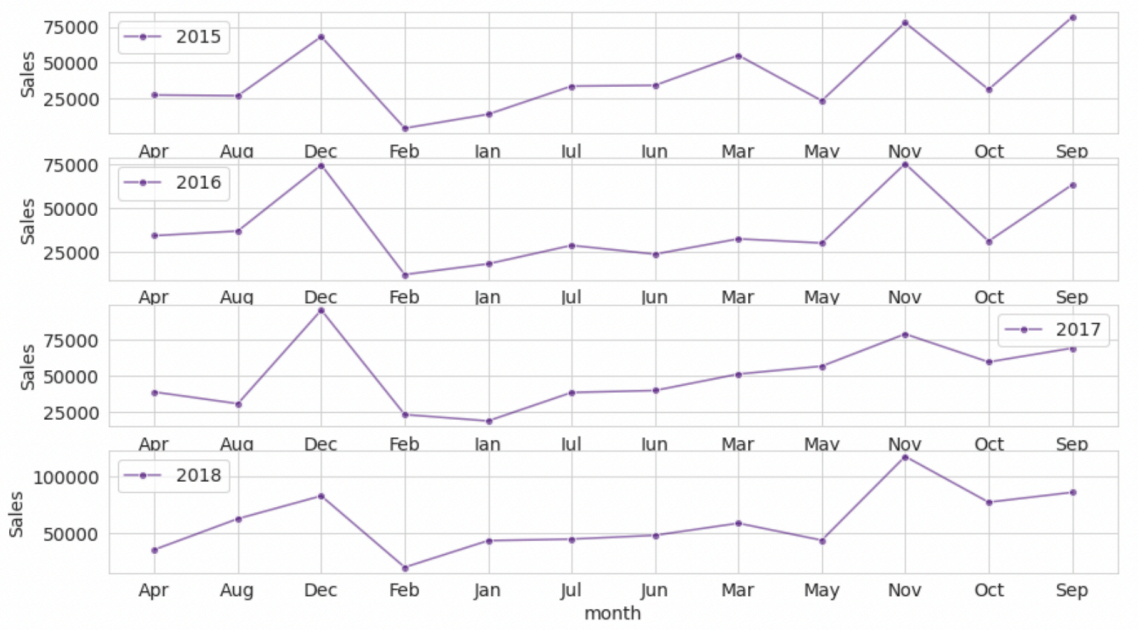
Bar Plot

Monthly observation of Sales Pattern



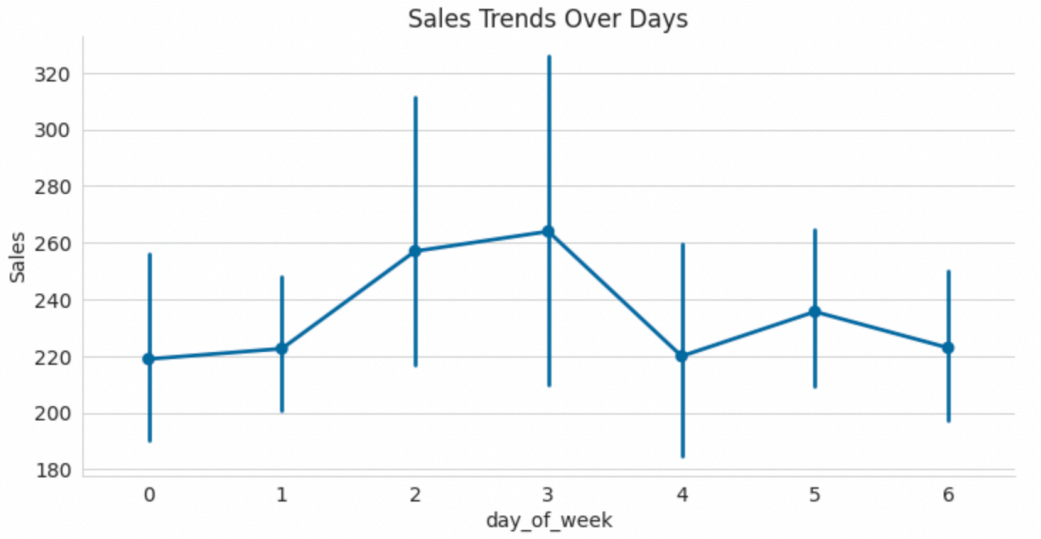
The bar plot indicates higher sales in September, December, and November. To verify this pattern across years, we can analyze sales data yearly. By comparing sales trends for these months across different years, we can determine if the observed growth is consistent annually.

monthly Year-to-Year observation of Sales Pattern



We can see that, There is rise in months of December, November, and September. The same pattern observed in each year, however it appears at the different levels.

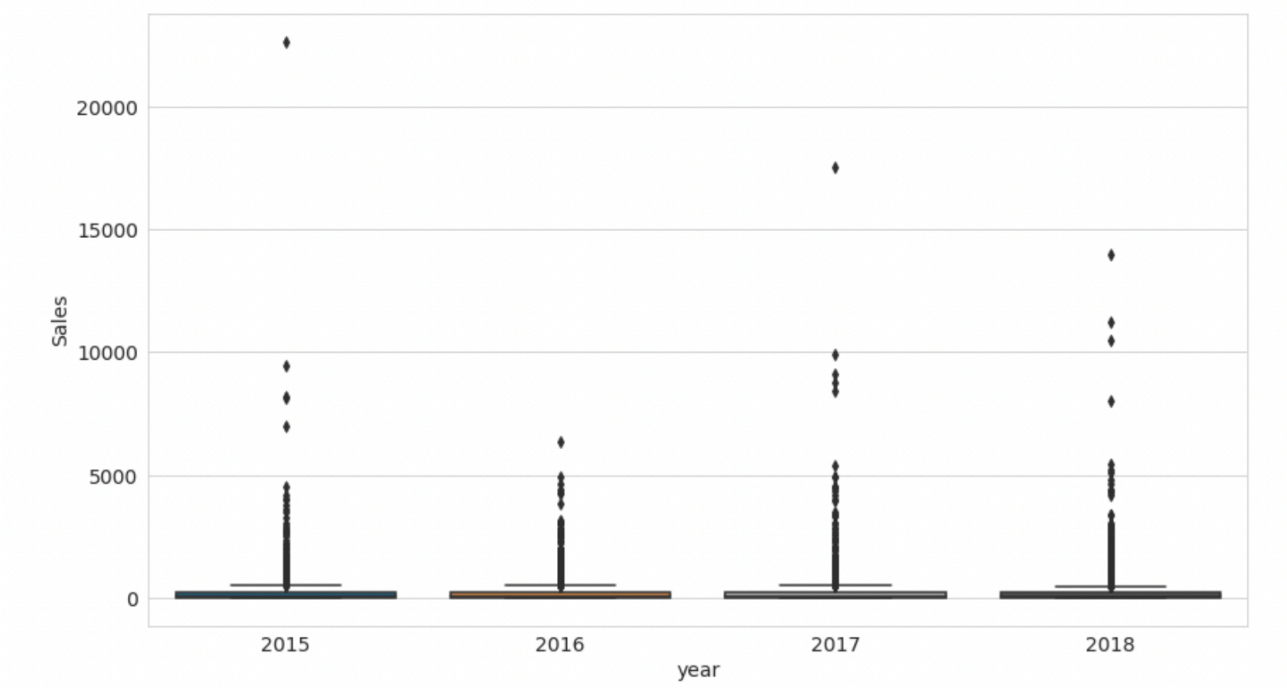
Sales trends over days



We can see that, there is maximum sales on Wednesday and Thursday.

Box and Whisker plots for Distribution.

Yearly observation of Distribution of Data. This will gives us an idea of spread of observation for each year



We can see that there are outliers in Sales values for each year.

2. Stationarity of Time Series and Identifying Patterns

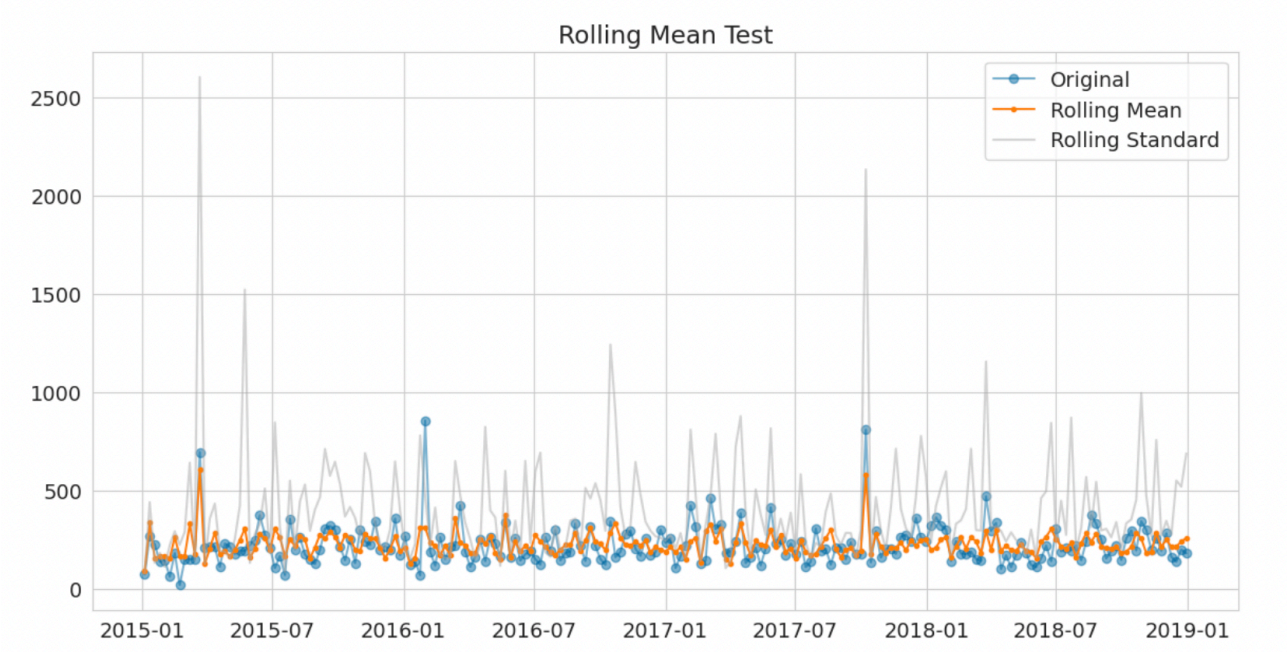
To use the time series forecasting models, we need to ensure that the our data is stationary

The time series is stationary when data has constant mean, constant variance, and constant covariance with respect to time.

There are two ways to check Stationarity of Time Series.

i. Rolling Mean:

A rolling analysis of a time series model is often used to assess the model's stability over time. The window is rolled (slid across the data) on a weekly basis, in which the average is taken on a weekly basis. Rolling Statistics is a visualization test, where we can compare the original data with the rolled data and check if the data is stationary or not.



Mean and Standard Deviation Stability:

The observation that the mean and standard deviation remain relatively constant over time indicates a stationary time series. This stability is essential for time series forecasting models.

ii. Augmented Dickey-Fuller test:

The Dickey Fuller test is one of the most popular statistical tests. It can be used to determine the presence of unit root in the series, and hence help us to understand if the series is stationary or not. The null hypothesis of the Augmented Dickey-Fuller is that there is a unit root data is non-stationary), with the alternative that there is no unit root data is stationary). if the p-value is less than critical value (i.e 0.05) we reject the null hypothesis which means that data is Stationary.

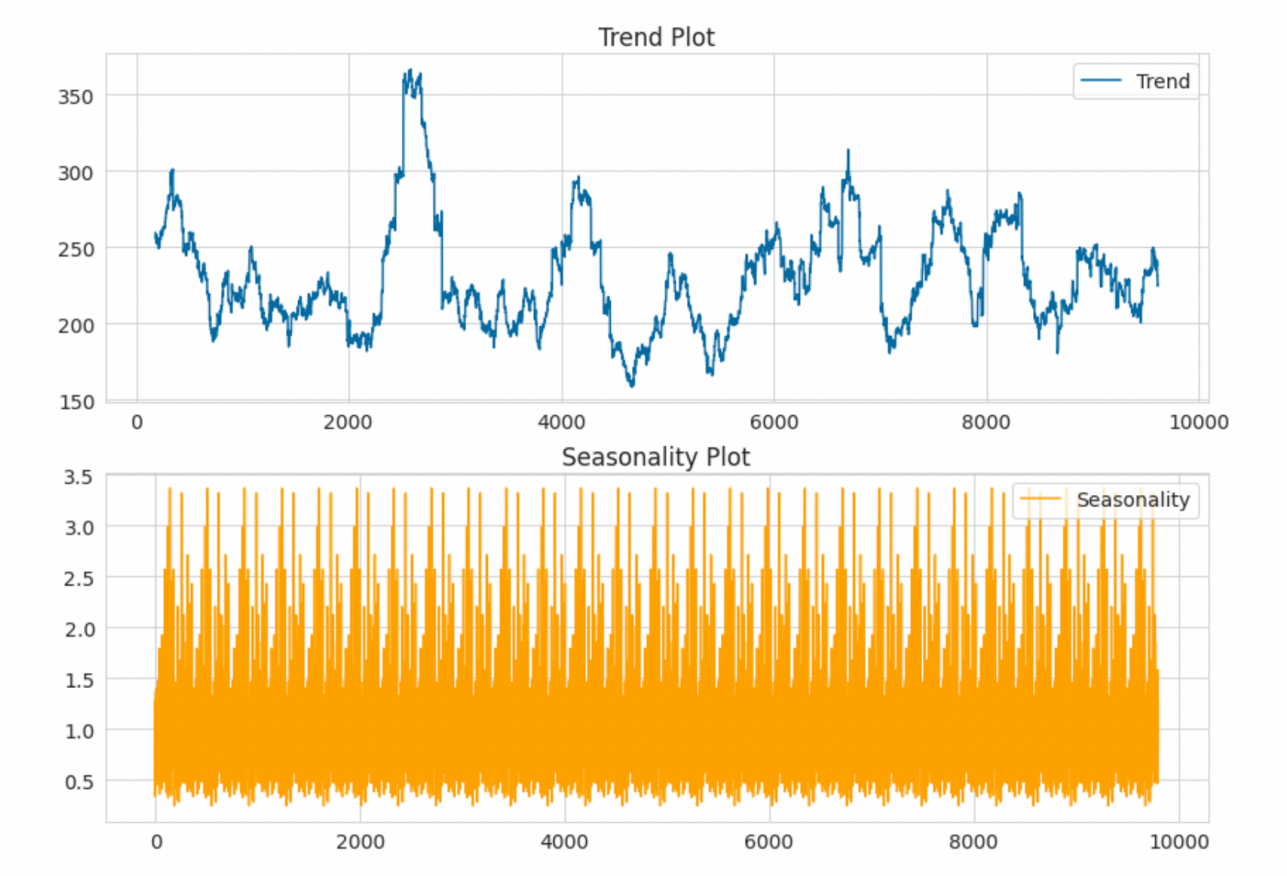
```
Augmented Dickey-fuller test result:
ADF test statistic:  -98.33059943935697
p-value: 0.0
Critical Values:
        1% : -3.431018
        5% : -2.861835
        10% : -2.566927
```

Augmented Dickey-Fuller (ADF) Test Result:

The ADF test statistic of -98.33 is substantially lower than the critical value at a 1% significance level (-3.43 in your example).

Since the test statistic is below the critical value and the p-value is less than 0.01 (1% significance level), we reject the null hypothesis of having a unit root.

Rejecting the null hypothesis indicates that the time series data is indeed stationary and does not exhibit a time-dependent structure.



The above line plot does not show any trends in data. So, There no differencing is required.

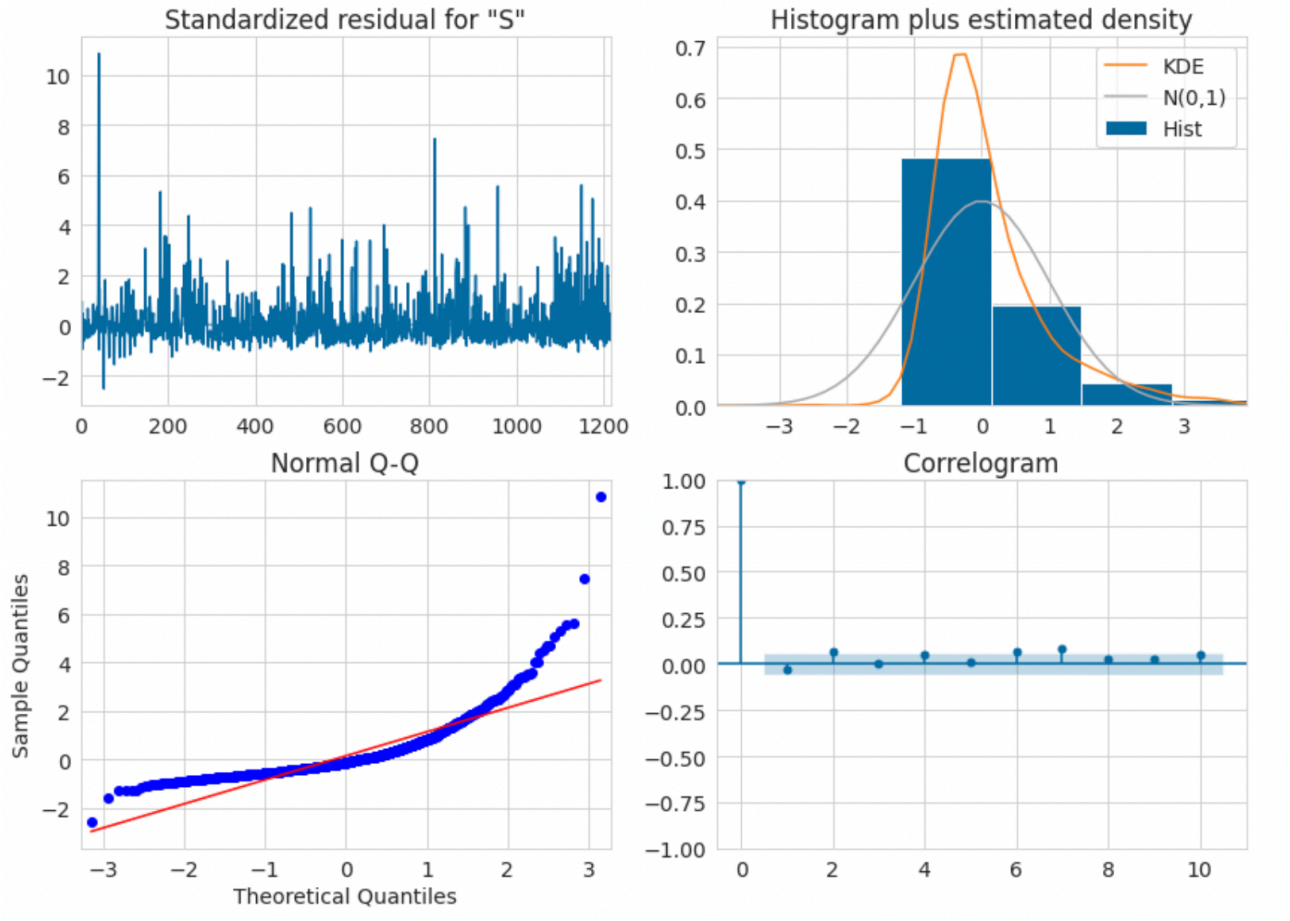
3. ARIMA

Fitting Seasonal Autoregressive Integrated Moving Average (SARIMA) model to aggregated sales data using statsmodels. The model's parameters are set for both non-seasonal (p=1, d=0, q=0) and seasonal (P=1, D=1, Q=1, m=12) components. After fitting the model to the data, it prints a summary table containing coefficients, standard errors, t-values, and p-values, providing insights into the model's performance and significance of its parameters.

SARIMAX Summary

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.1067	0.031	3.466	0.001	0.046	0.167
ar.S.L12	0.0394	0.023	1.689	0.091	-0.006	0.085
ma.S.L12	-0.9993	0.014	-69.130	0.000	-1.028	-0.971
sigma2	5.199e+06	2.82e-09	1.84e+15	0.000	5.2e+06	5.2e+06

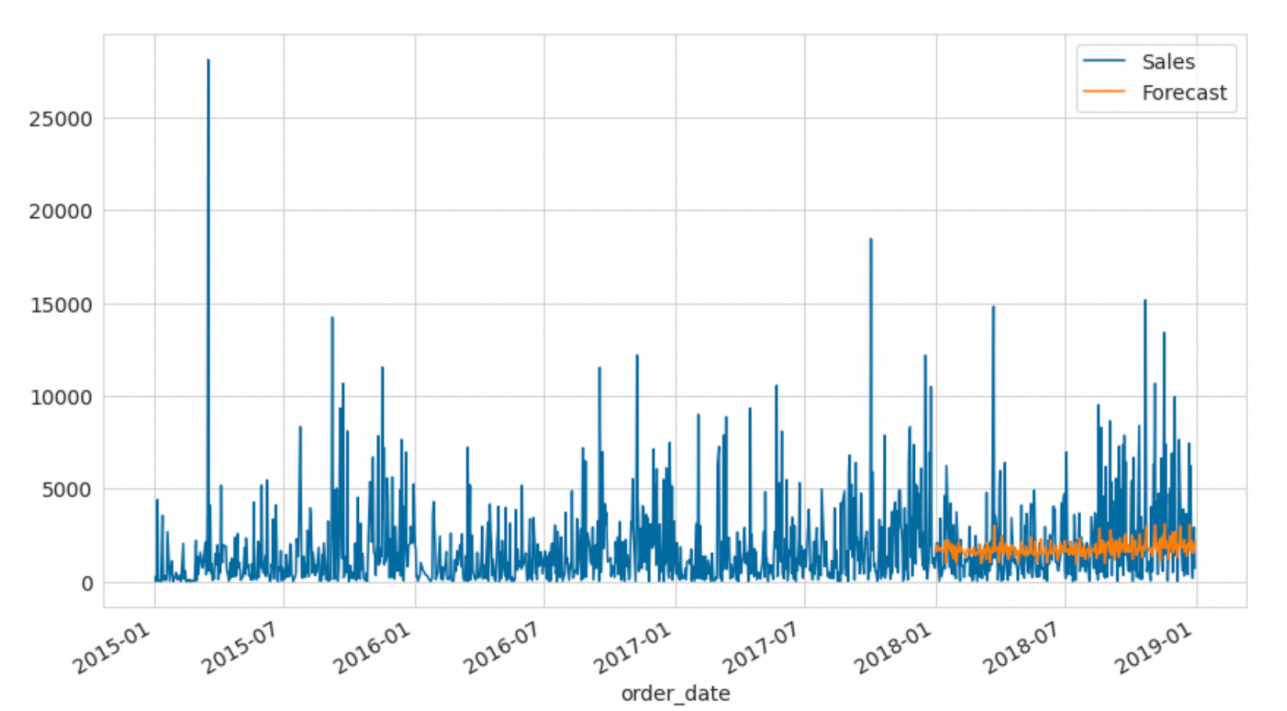
Visualization of the performance of the model :



A Forecast column to the sales DataFrame, containing predictions from the fitted SARIMA model using the predict method. The predictions are generated for the time period from '2018-01-01' to '2018-12-30' with dynamic=False (no dynamic forecasting).

After adding the forecasted values, the code creates a visualization by plotting the 'Sales' and 'Forecast' columns of the sales DataFrame. The resulting plot is displayed with a figure size of 14x8 using matplotlib's plot function and plt.show().

This visualization helps compare the actual sales data with the forecasted values from the SARIMA model, providing insights into the model's performance in predicting future sales trends.



Calculating the Root Mean Squared Error (RMSE) for the SARIMA model's forecasted values compared to the actual sales data for the time period from '2018-01-01' to '2018-12-30'. The RMSE value obtained is approximately 2404.88, indicating the average difference between the forecasted and actual values. Lower RMSE values suggest better model accuracy in predicting future sales.

```
Root Mean Squared Error for SARIMAX: 2404.877288233176
```

4. Prediction

XGBoost to build a regressor model for sales prediction. It aggregates sales data by date, splits it into features (excluding sales) and targets (sales). The data is then split into training and testing sets. An XGBoost regressor is trained with specified hyperparameters (learning rate, max depth). The model predicts sales on the test set, and the Root Mean Squared Error (RMSE) is calculated to evaluate model performance. Lower RMSE indicates better predictive accuracy. Overall, the code demonstrates training an XGBoost model for sales forecasting and evaluating its performance using RMSE.

```
Root Mean Squared Error for XGBoost: 1714.6441622267337
```

	RMSE
SARIMAX	2404.877288
XGBRegressor	1714.644162

The Root mean squared error of XGBRegressor model is less than SARIMAX. We can use XGBRegressor for forecasting Sales.