19CSE32 Business Analytics Case Study Report

Topic: Sales Forecasting

Team Details:

Roll Number	Name
CB.EN.U4CSE21213	B J Vasanth
CB.EN.U4CSE21234	B Lakshman Sai
CB.EN.U4CSE21256	M Sasidhar
CB.EN.U4CSE21269	Y G Ram Darshan Reddy

Introduction:

Sales forecasting using the sales data.csv dataset involves a multi-faceted approach integrating preprocessing, exploratory data analysis (EDA), and advanced analytics techniques to derive actionable insights and predict future sales trends. Initially, preprocessing steps such as data cleaning, handling missing values, and encoding categorical variables are performed to ensure data quality and compatibility for analysis. EDA techniques, including descriptive statistics, data visualization using libraries like Matplotlib and Seaborn, and correlation analysis, provide a comprehensive understanding of the dataset's characteristics, relationships, and potential trends. Visualizations such as histograms, scatter plots, and heatmaps offer intuitive representations of sales patterns, seasonal variations, and relationships between different sales attributes.

The analysis extends to sophisticated techniques like the Apriori algorithm for market basket analysis, identifying frequent itemsets and association rules among products purchased together. This facilitates targeted marketing strategies, product bundling, and inventory management optimization based on customer preferences and purchasing behavior. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are utilized for dimensionality reduction and feature extraction, enabling visualization of high-dimensional sales data and uncovering underlying structures or patterns.

Content-based and Collaborative Filtering methods are employed for personalized recommendation systems, leveraging product attributes, customer profiles, and historical transaction data to suggest relevant products to customers. These approaches enhance customer engagement, satisfaction, and retention by delivering tailored shopping experiences.

Moreover, interactive visualization libraries such as Bokeh and Plotly are utilized to create dynamic and interactive dashboards, facilitating intuitive exploration of sales data trends, forecasts, and performance metrics. These visualizations enable stakeholders to gain actionable insights, make informed decisions, and adapt strategies in real-time to market dynamics.

In conclusion, the integration of preprocessing, EDA, and advanced analytics techniques like Apriori, PCA, LDA, content filtering, and collaborative filtering, supported by interactive visualization tools like Bokeh and Plotly, empowers businesses to achieve accurate sales forecasting, optimize marketing strategies, and drive revenue growth in today's competitive market landscape.

Pre Processing:

Dataset columns:

```
df.columns
```

```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country/Region', 'City',
       'State/Province', 'Postal Code', 'Region', 'Product ID', 'Category',
       'Sub-Category', 'Product Name', 'Sales', 'Quantity', 'Discount',
       'Profit'],
      dtype='object')
```

Missing Values:

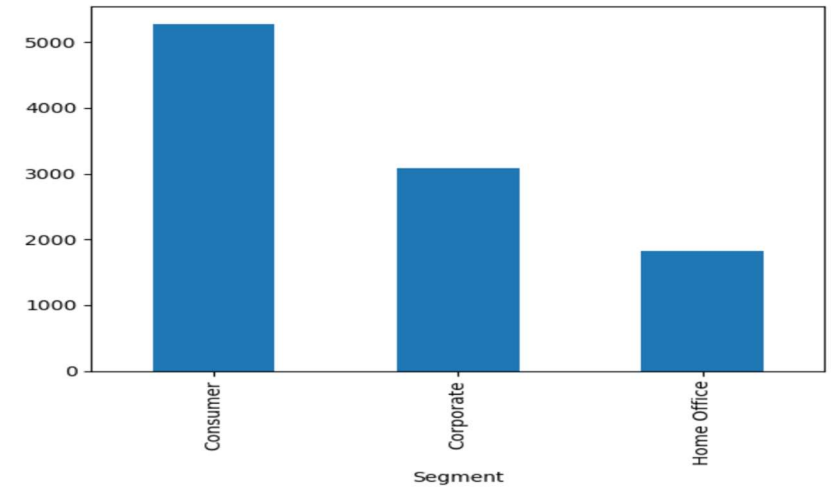
```
df.isnull().sum()
```

```
Row ID           0
Order ID         0
Order Date       0
Ship Date        0
Ship Mode        0
Customer ID      0
Customer Name    0
Segment          0
Country/Region   0
City             0
State/Province   0
Postal Code      0
Region           0
Product ID       0
Category         0
Sub-Category     0
Product Name     0
Sales            0
Quantity         0
Discount         0
Profit           0
dtype: int64
```

Segement Values:

```
df["Segment"].value_counts().plot(kind="bar")
```

```
<Axes: xlabel='Segment'>
```



Heatmap:

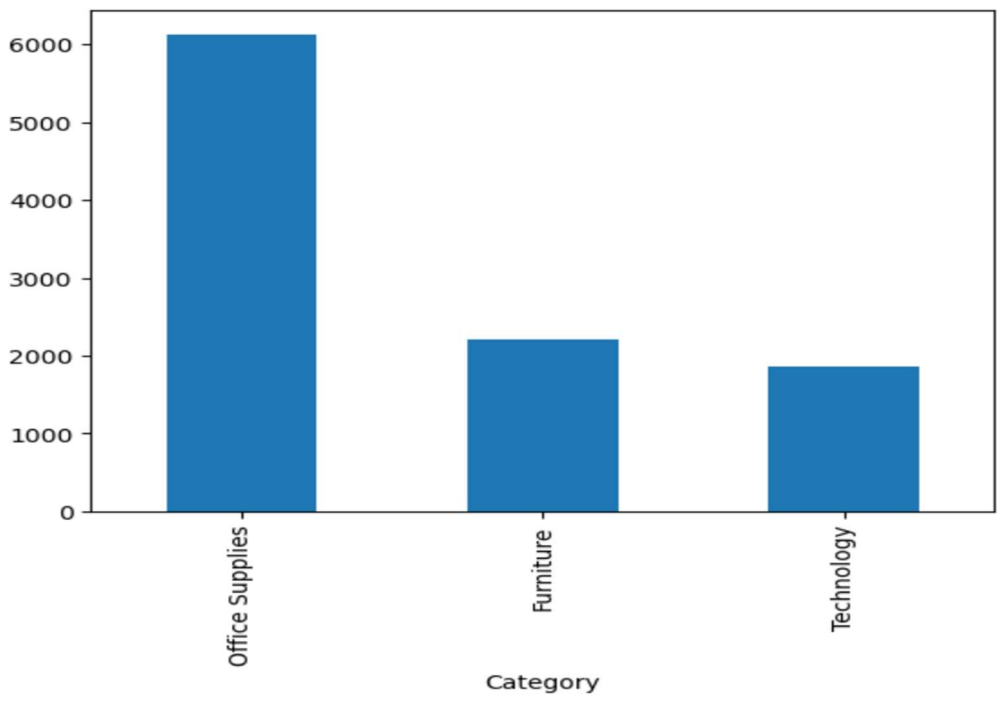
```
data=df[['Sales','Quantity','Discount','Profit']]
sns.heatmap(data.corr(),annot=True)
```

```
<Axes: >
```

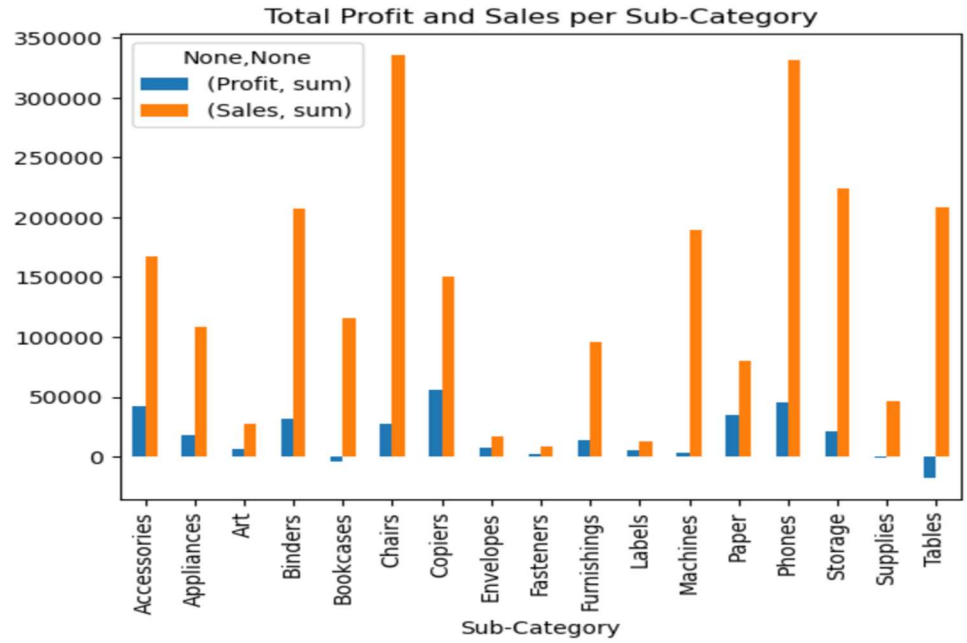
Category Values:

```
df['Category'].value_counts().plot(kind="bar")
```

```
<Axes: xlabel='Category'>
```



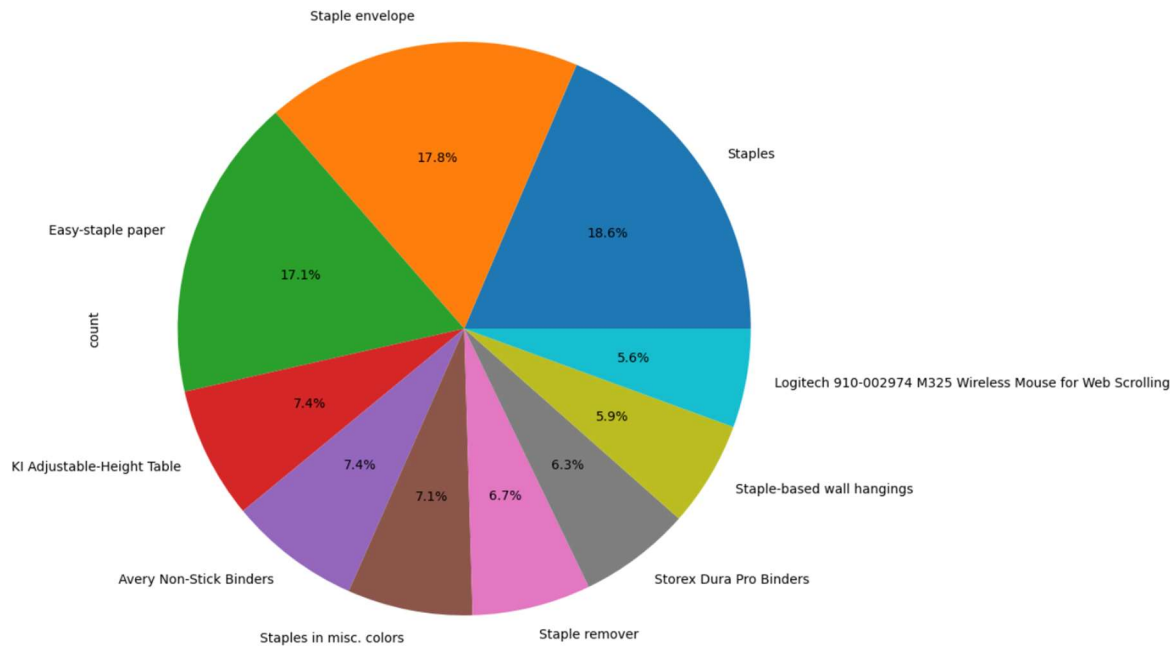
```
# Assuming df is your DataFrame containing 'Sub-Category', 'Profit', and 'Sales' columns
df.groupby('Sub-Category')[['Profit', 'Sales']].agg(['sum']).plot.bar()
plt.title('Total Profit and Sales per Sub-Category')
plt.show()
```



Top 10 Products:

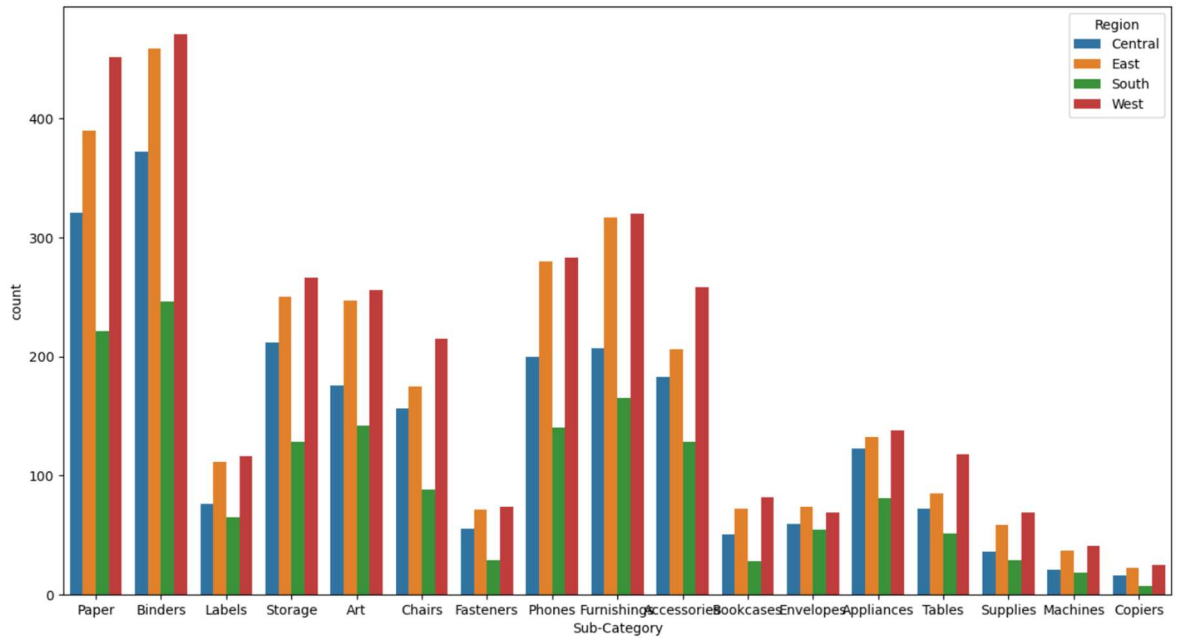
```
#Distribution of Top 10 Products
plt.figure(figsize=(12,10))
df['Product Name'].value_counts().head(10).plot.pie(autopct="%1.1f%%")
```

```
<Axes: ylabel='count'>
```

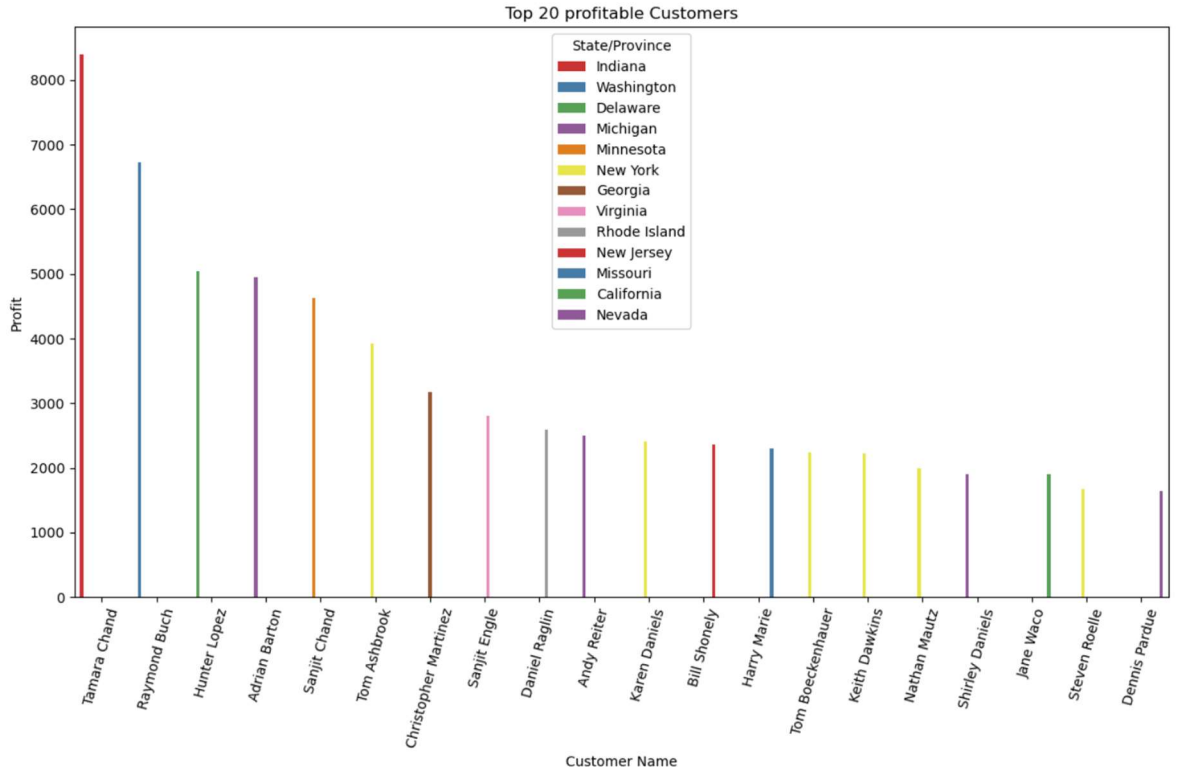


Sub category Region Wise:

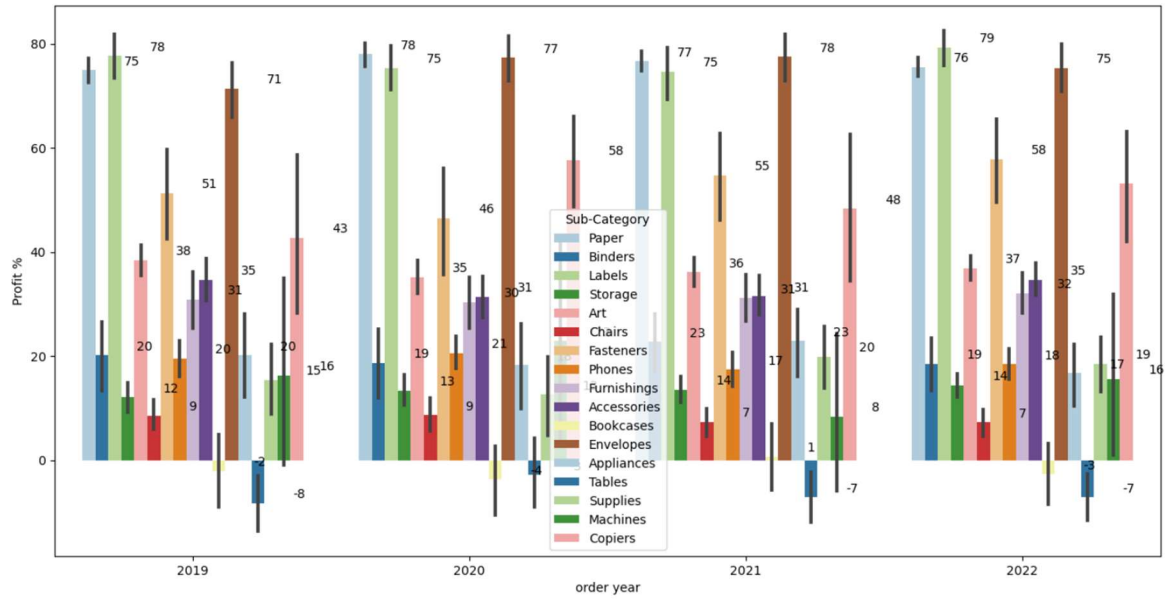
```
#Count of Sub-Category region wise
plt.figure(figsize=(15,8))
sns.countplot(x="Sub-Category", hue="Region", data=df)
plt.show()
```



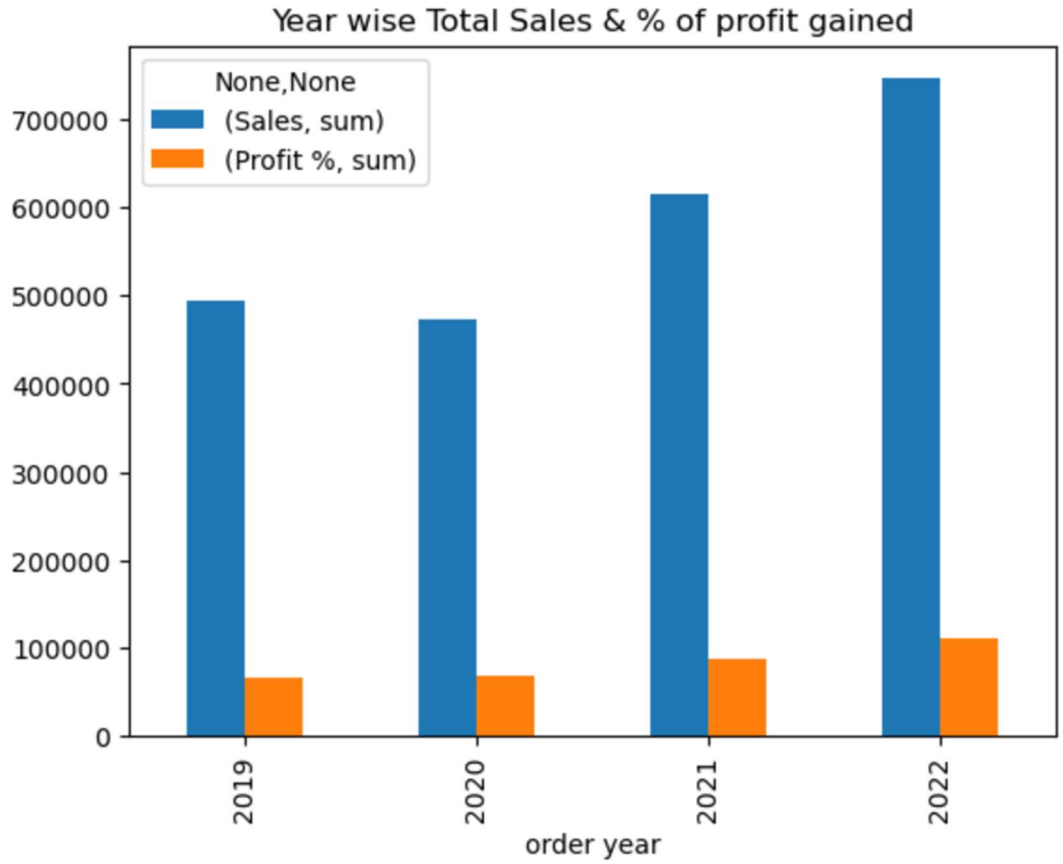
Top 20 Profitable Customers:



Sub Category : Profits Vs Order year

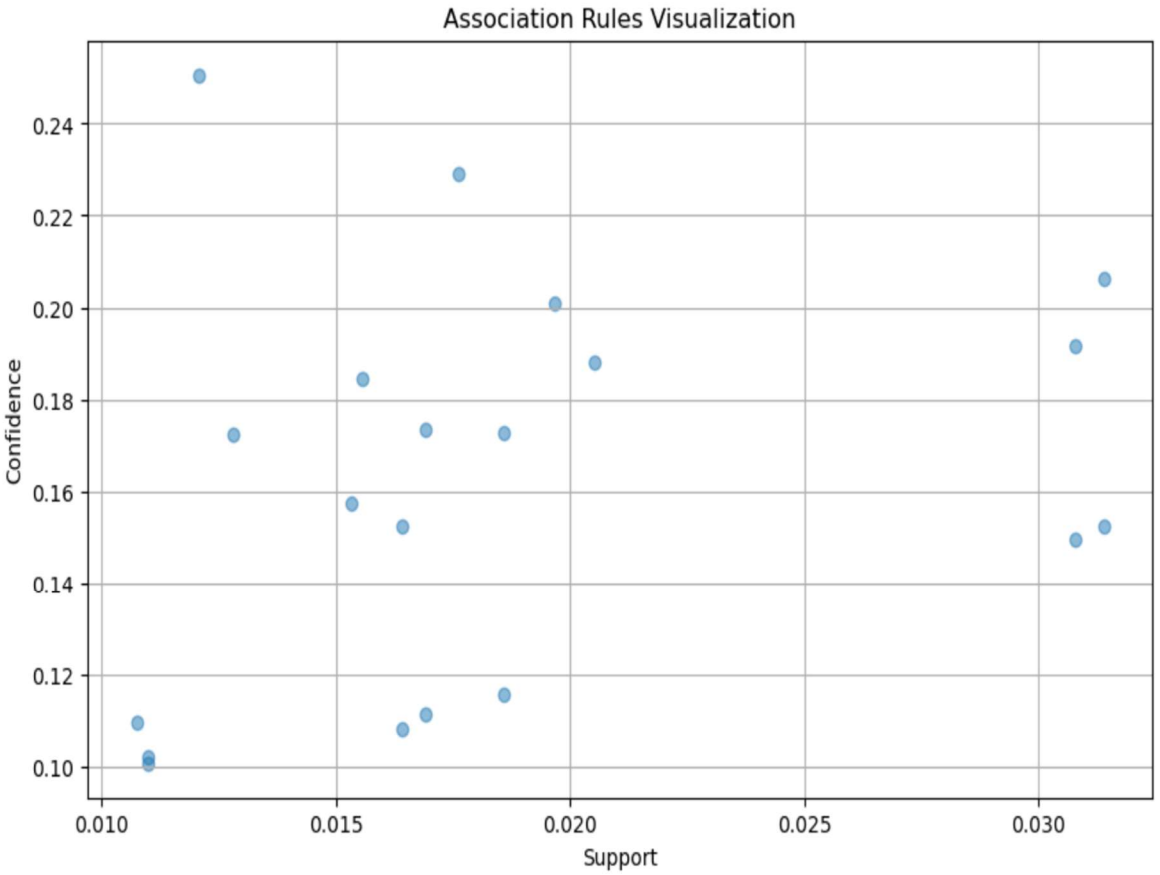


Yeats sales Vs Profits



Apriori:

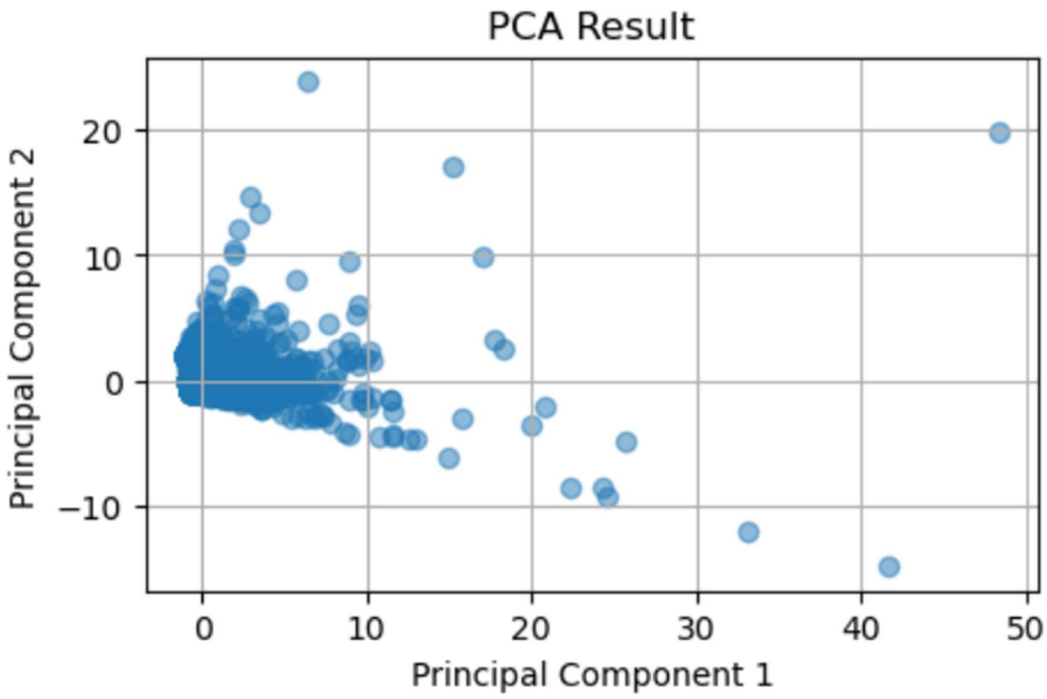
The Apriori algorithm was applied twice to analyze the relationships between different factors and their impact on profitability. In the first analysis, transactions were created based on 'State', 'Category', and 'Profit_Category', extracting association rules that reveal the connection between states,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

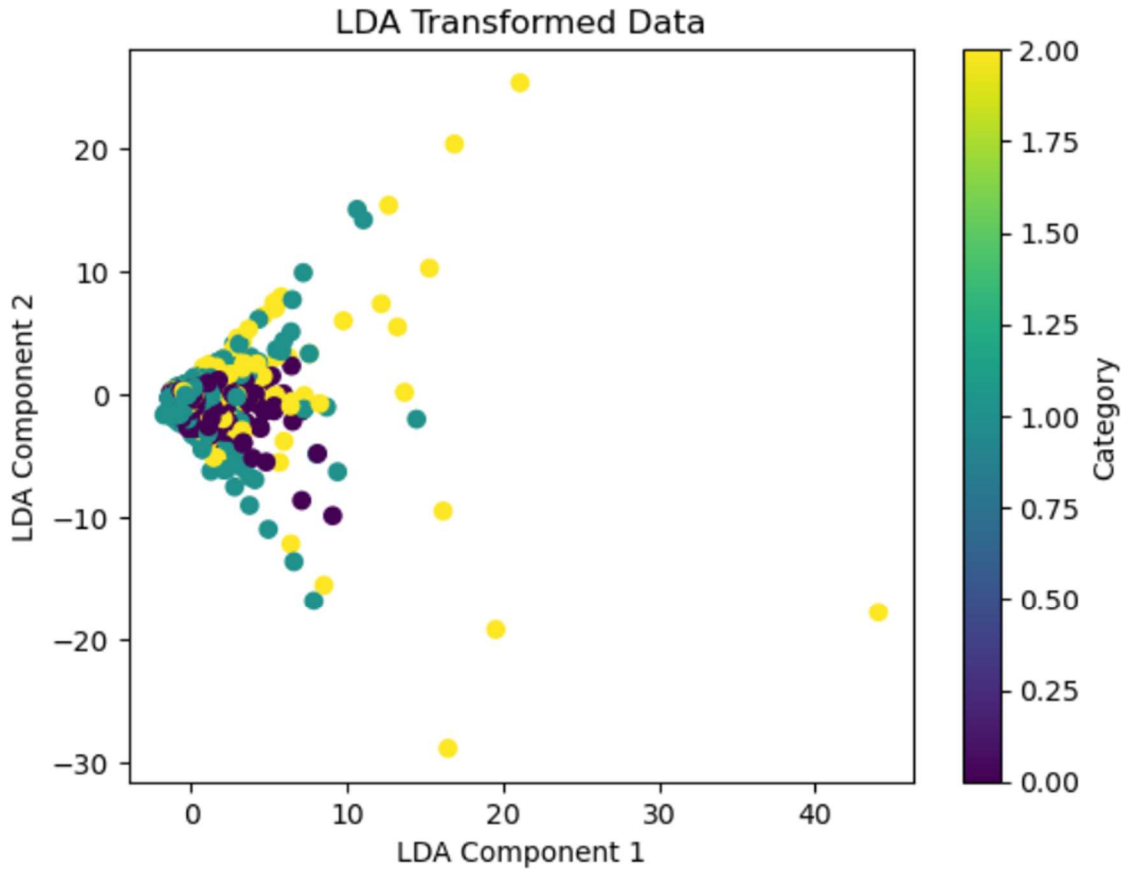
PCA:

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 reduceddimensional 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.' 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.



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:

1. Hon Deluxe Fabric Upholstered Stacking Chairs, Rounded Back

2. Bush Somerset Collection Bookcase

3. C-Line Cubicle Keepers Polyproplyene Holder w/Velcro Back, 8-1/2x11, 25/Bx

4. SimpliFile Personal File, Black Granite, 15w x 6-15/16d x 11-1/4h

5. 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.

Recommended products based on 'Furniture' category (Content-Based Filtering):

1)Howard Miller 14-1/2" Diameter Chrome Round Wall Clock.

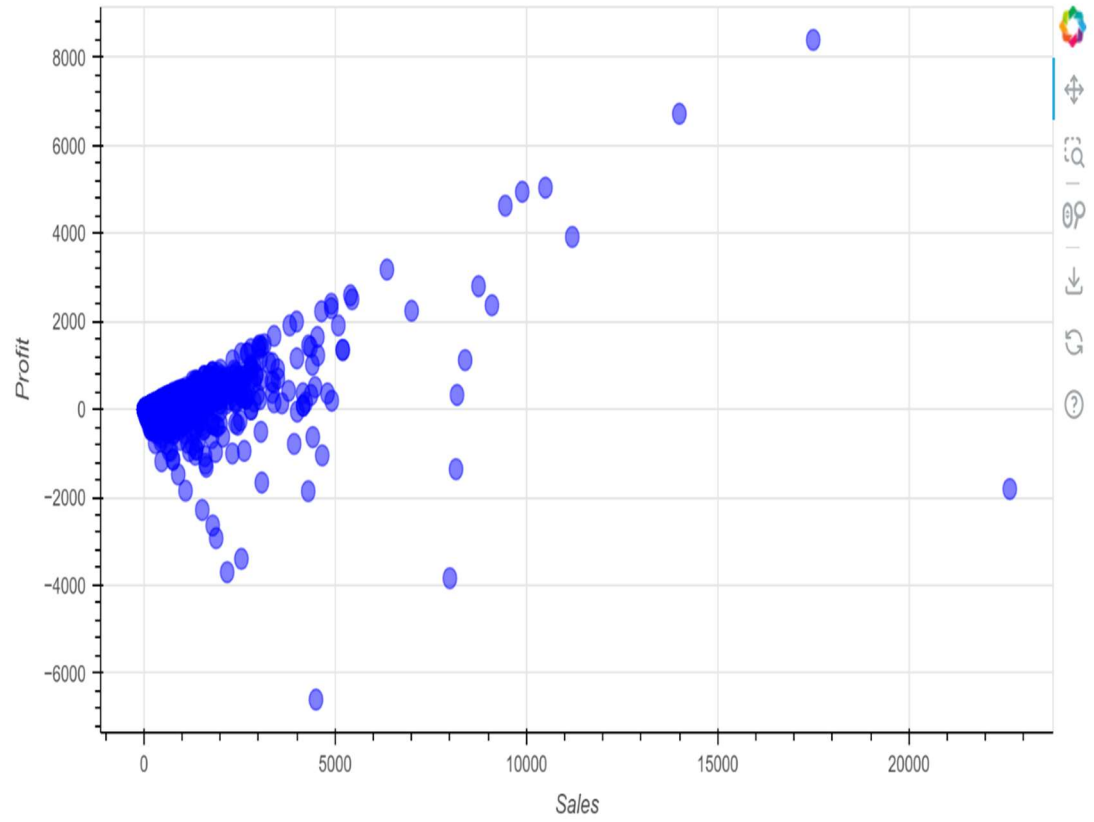
2)DAX Value U-Channel Document Frames, Easel Back.

3)Global Highback Leather Tilter in Burgundy.

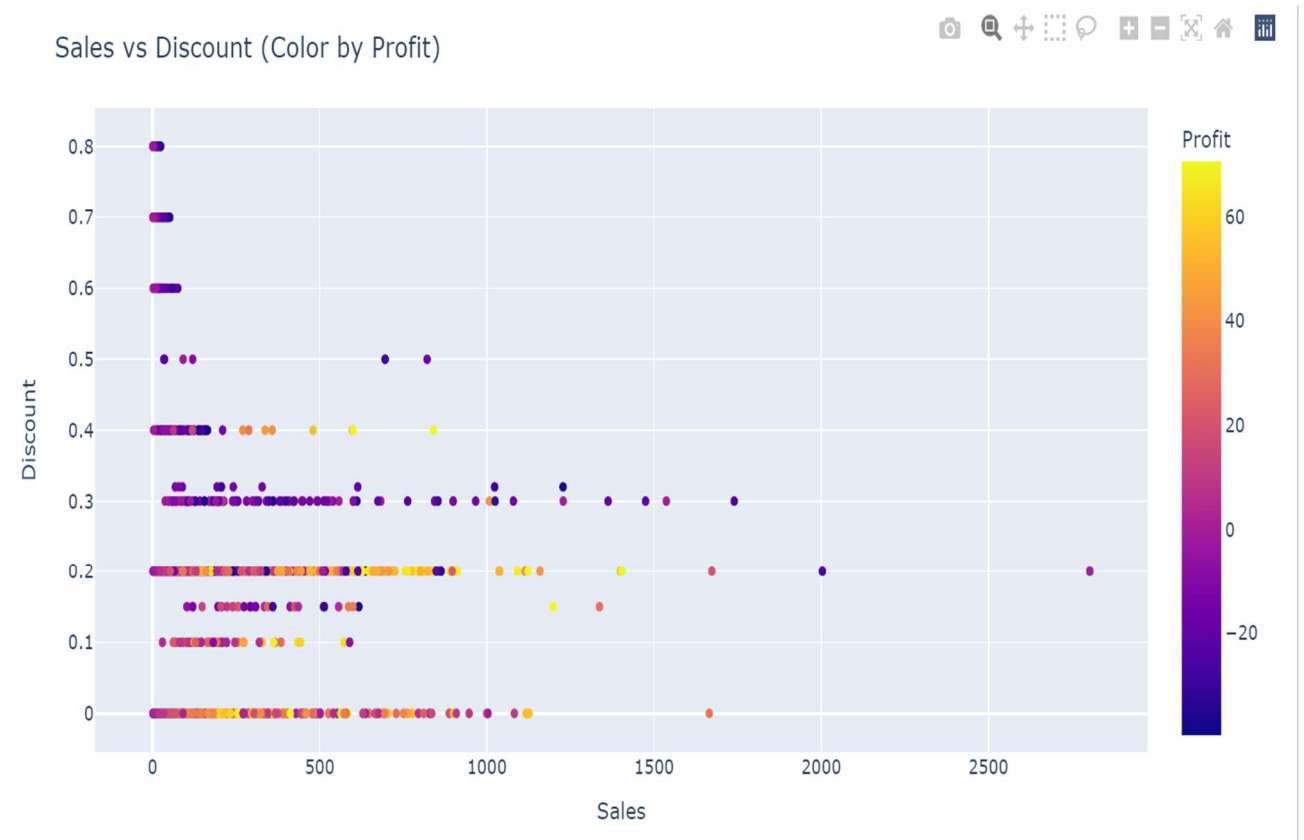
4)O'Sullivan Elevations Bookcase, Cherry Finish.

5)Howard Miller 11-1/2" Diameter Ridgewood Wall Clock.

Bokeh:



Plotly:

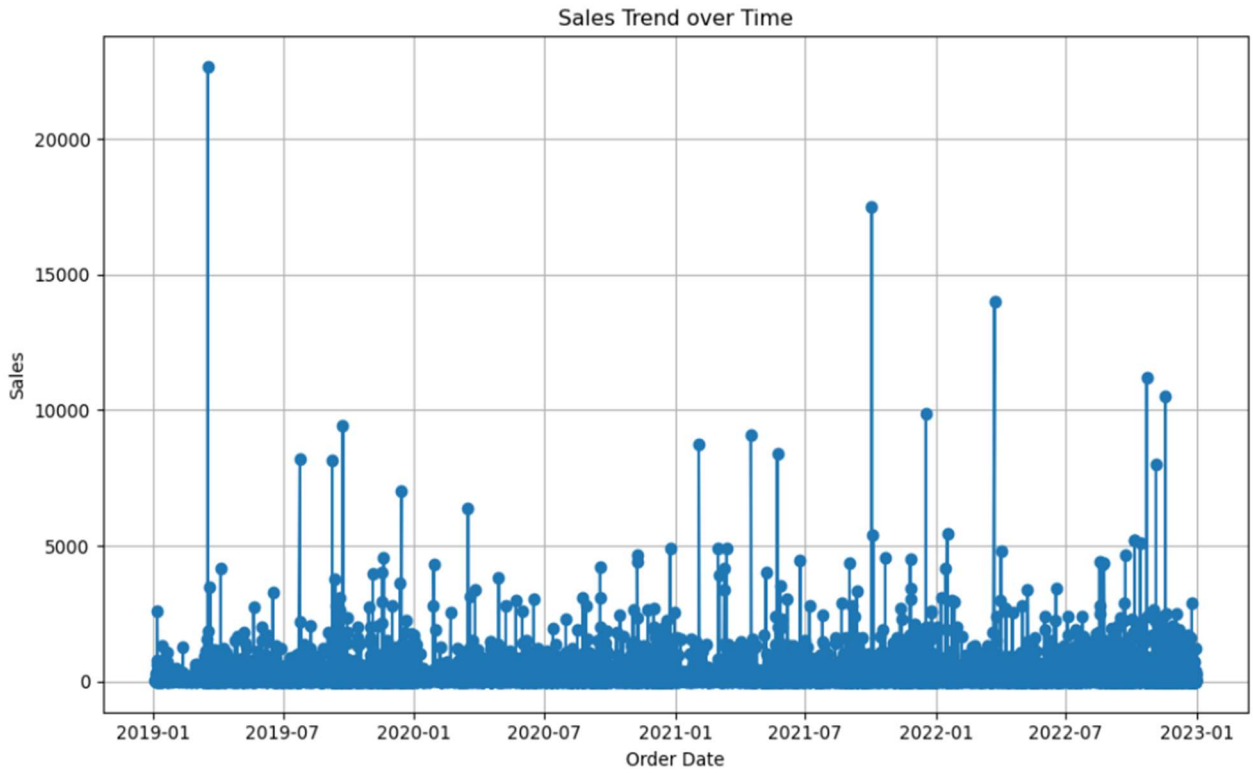


TIME SERIES ANALYSIS

Time series analysis is a statistical method focused on studying patterns, trends, and relationships within data collected over time. It involves exploring components like trend, seasonality, and randomness to gain insights and make predictions. With applications in economics, finance, healthcare, and more, time series analysis aids in forecasting future trends, identifying patterns, and informing decision-making processes. Using specialized software tools and libraries, analysts can effectively analyze time-dependent data to extract valuable insights for various domains and industries.

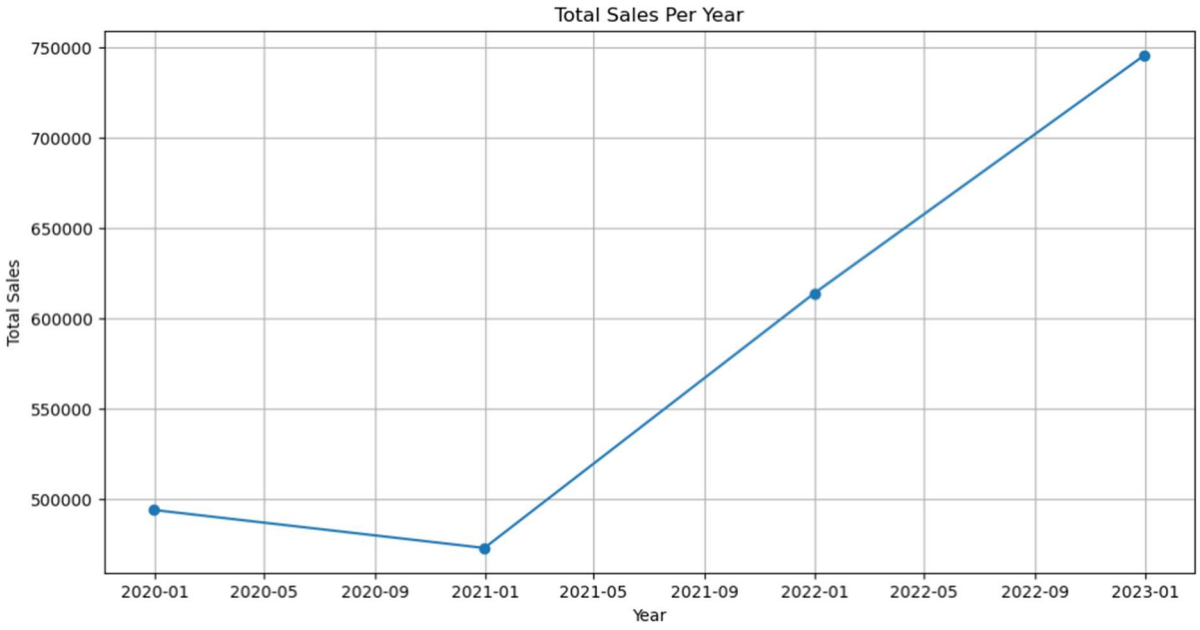
1. Visualisation

Visualizations in time series analysis provide a concise way to understand data patterns, trends, and outliers. Techniques like line plots and histograms offer quick insights, while interactive tools aid exploration of large datasets, facilitating informed decision.



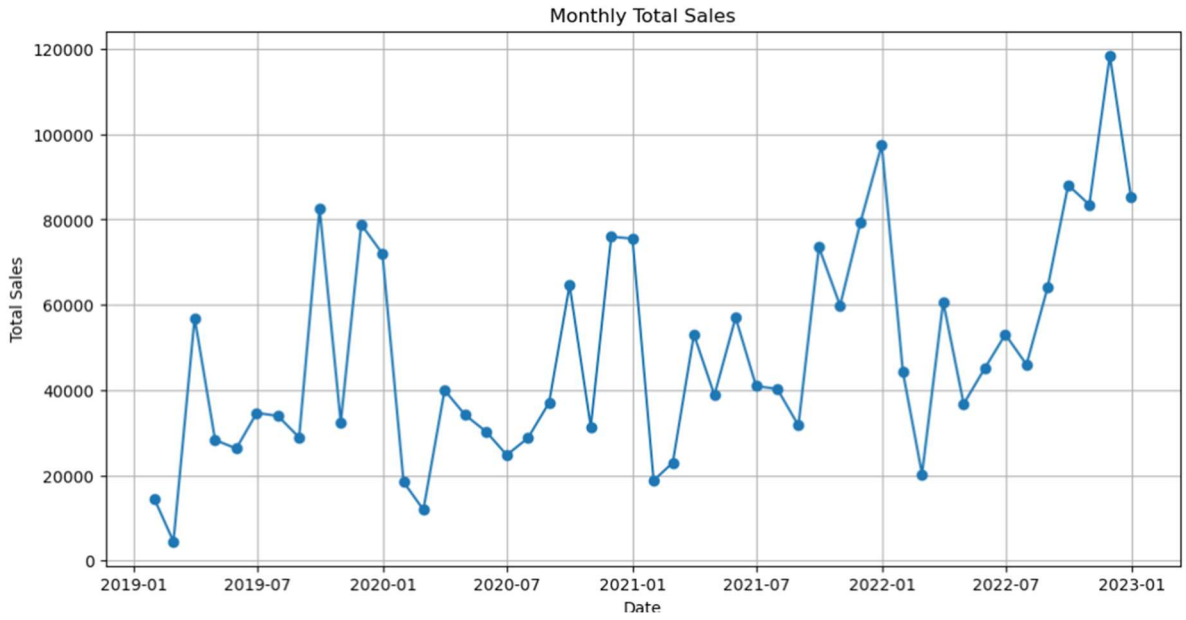
Total Sales Per Year Plot:

This plot presents the total sales for each year, providing insights into the annual performance of sales over the entire duration of the dataset.



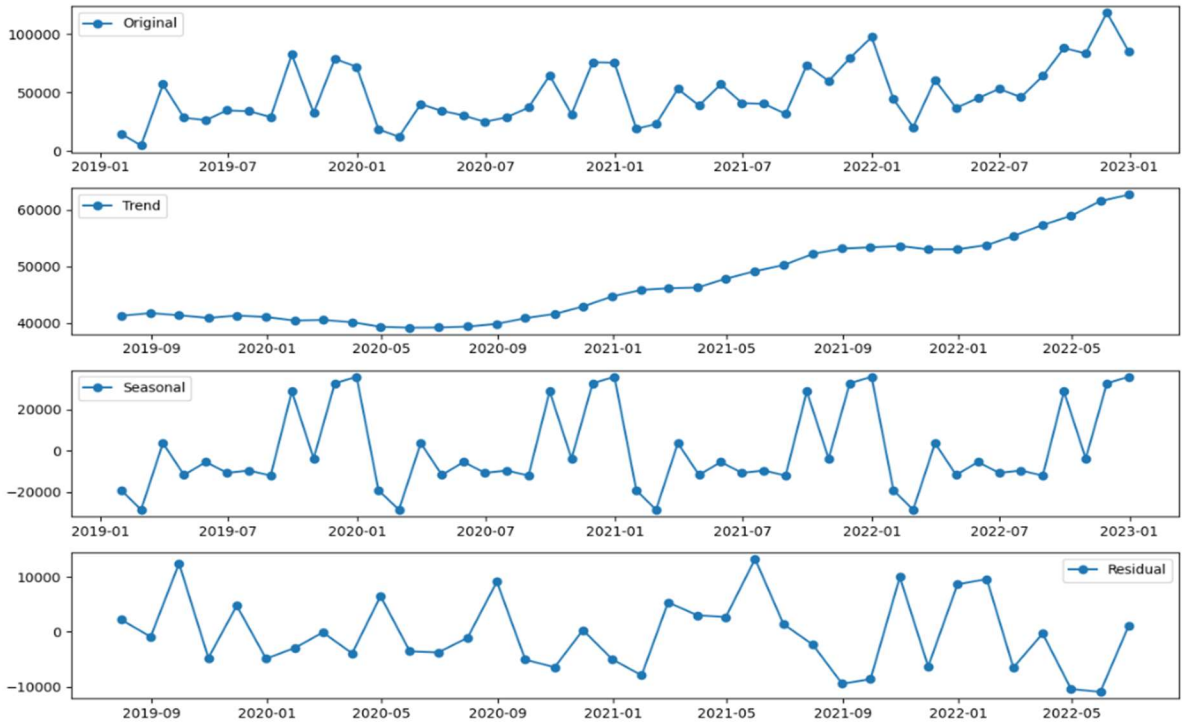
Monthly Total Sales Plot:

This plot illustrates the total sales over time, with each data point representing the sum of sales for a particular month. The plot shows the overall trend in sales over the entire period covered by the dataset.



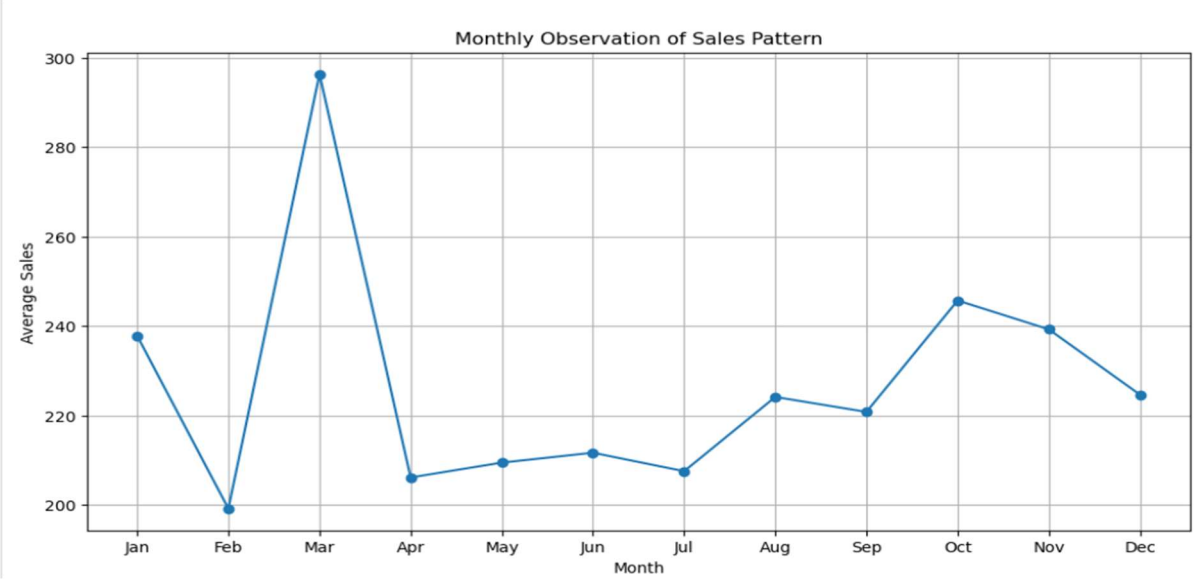
Decomposed Components Plot:

This plot decomposes the monthly total sales into its constituent components: trend, seasonal, and residual. The trend component represents the long-term movement or direction of sales, while the seasonal component shows the periodic patterns that repeat over each year. The residual component captures the irregular fluctuations or noise in the data.



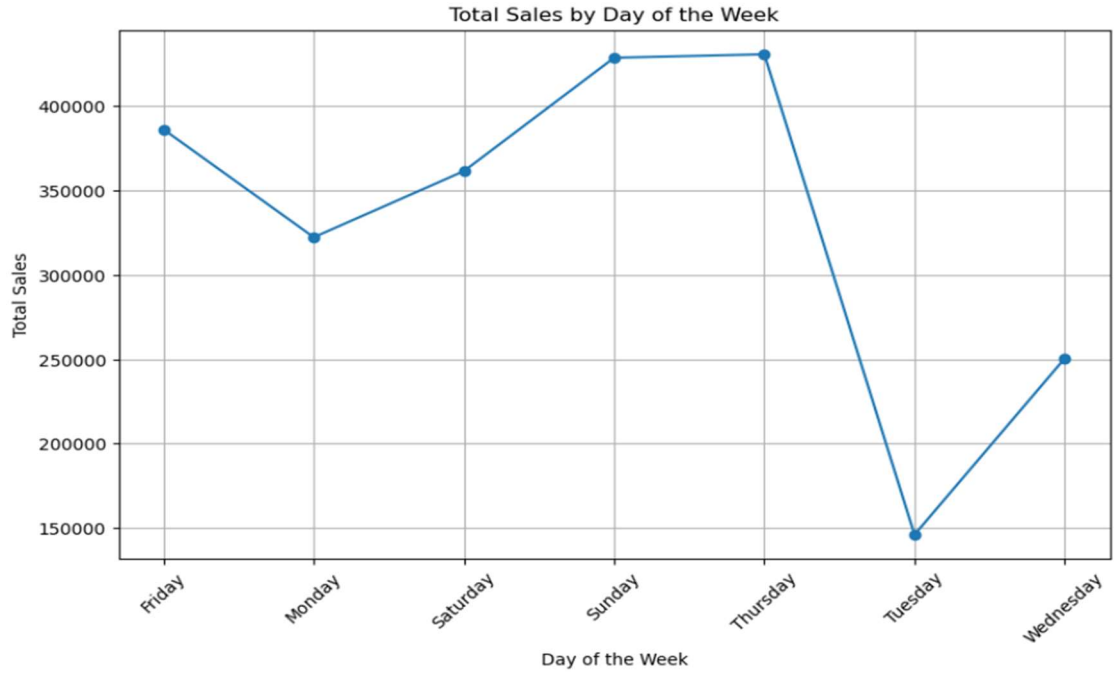
Monthly Observation of Sales Pattern Plot:

This plot displays the average sales for each month across all years, highlighting any recurring patterns or seasonality in sales behavior on a monthly basis.



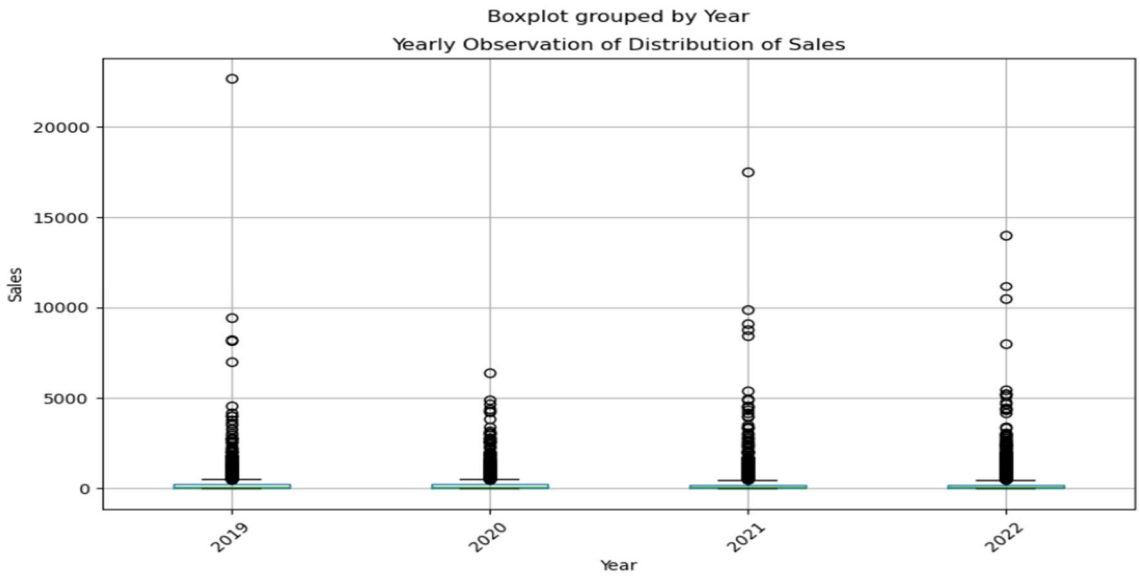
Total Sales by Day of the Week Plot:

It aggregates the sales data by day of the week and creates a plot illustrating the total sales for each day. The resulting visualization, with markers and lines indicating sales trends, offers insights into variations in sales activity throughout the week.



Yearly Observation of Distribution of Sales - Box and Whisker Plot:

The plot displays yearly observations of the distribution of sales data. It presents box and whisker plots for each year separately, allowing for a comparative analysis of sales distributions across different years. This visualization helps in identifying any changes or patterns in sales distribution over time, providing valuable insights for strategic decision-making.



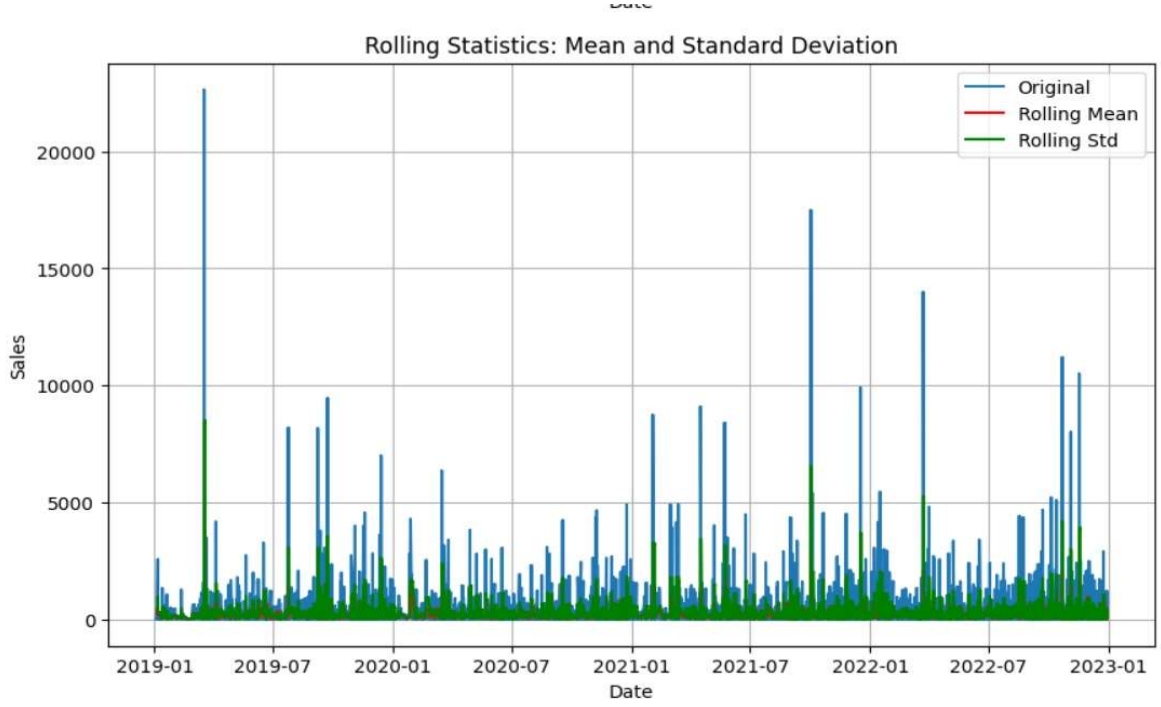
2.Stationarity of Time Series and Identifying Patterns

Stationarity in time series data ensures that statistical properties remain constant over time, like mean, variance, and covariance. Methods such as rolling mean analysis help identify trends and variations, crucial for accurate modeling and forecasting.

There are two ways to check Stationarity of Time Series.

i. Rolling Mean:

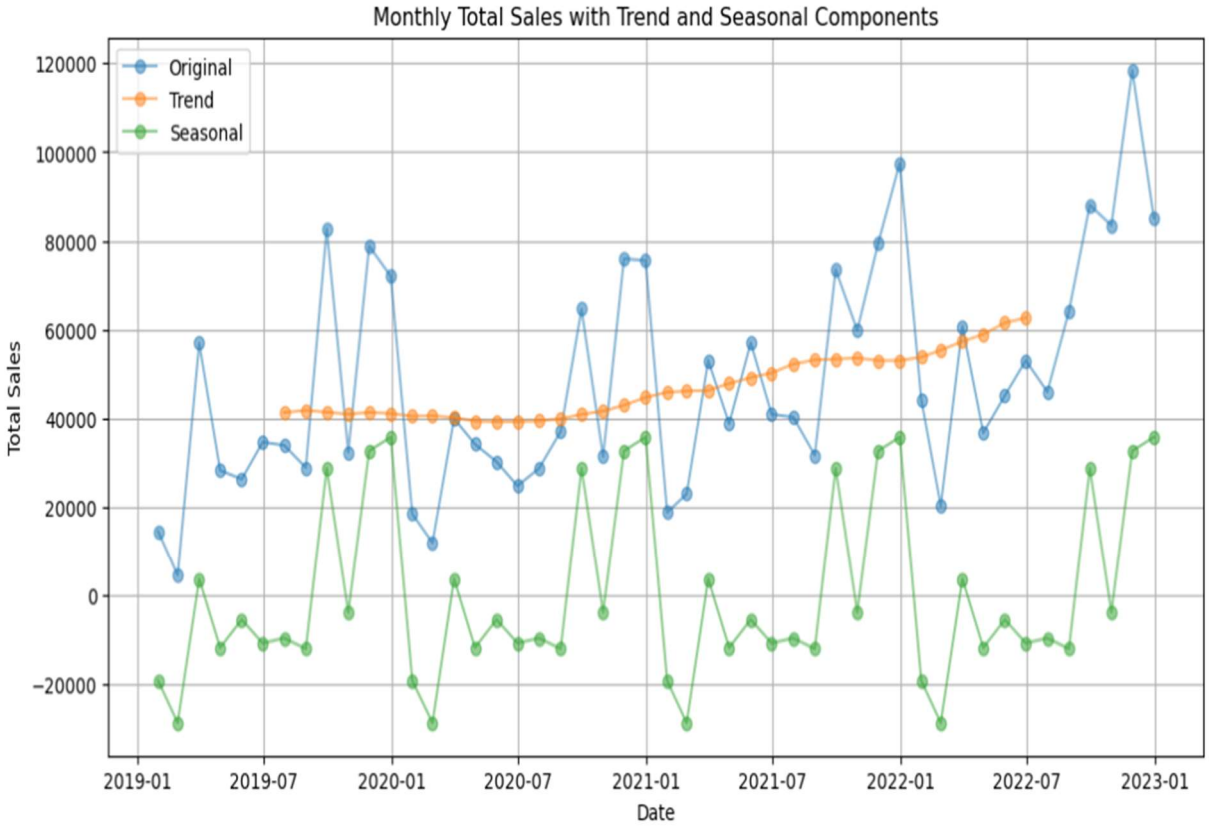
A rolling analysis of a time series model is often used to assess the model's stability over time. The window is rolled (slid across the data) on a weekly basis, in which the average is taken on a weekly basis. Rolling Statistics is a visualization test, where we can compare the original data with the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



ii. Augmented Dickey-Fuller test:

The Augmented Dickey-Fuller (ADF) test is a statistical method used to determine the stationarity of time series data. It assesses whether a unit root is present in the data, indicating non-stationarity. By comparing a test statistic to critical values, the ADF test provides a clear indication of stationarity, crucial for accurate modeling and forecasting.

```
ADF Statistic: -4.437507341232084
p-value: 0.00025460649254061 47
Critical Values:
   1%: -3.577848037043814 6
   5%: -2.925338105429433
   10%: -2.600773531009506 4
```



3. ARIMA

ARIMA models are useful for analyzing and forecasting time series data that exhibit trends, seasonality, or both. They are widely used in various domains such as finance, economics, weather forecasting, and more. However, it's essential to choose appropriate values for the ARIMA parameters based on the characteristics of the data and perform model diagnostics to ensure the model's reliability and accuracy

SARIMAX Summary:

SARIMAX Results

Dep. Variable:	Sales	No. Observations:	1458
Model:	SARIMAX(1, 1, 1)x(1, 1, 1, 12)	Log Likelihood	-13189.689
Date:	Sat, 11 May 2024	AIC	26389.378
Time:	15:46:22	BIC	26415.757
Sample:	01-03-2019	HQIC	26399.224
	- 12-30-2022		
Covariance Type:	opg		

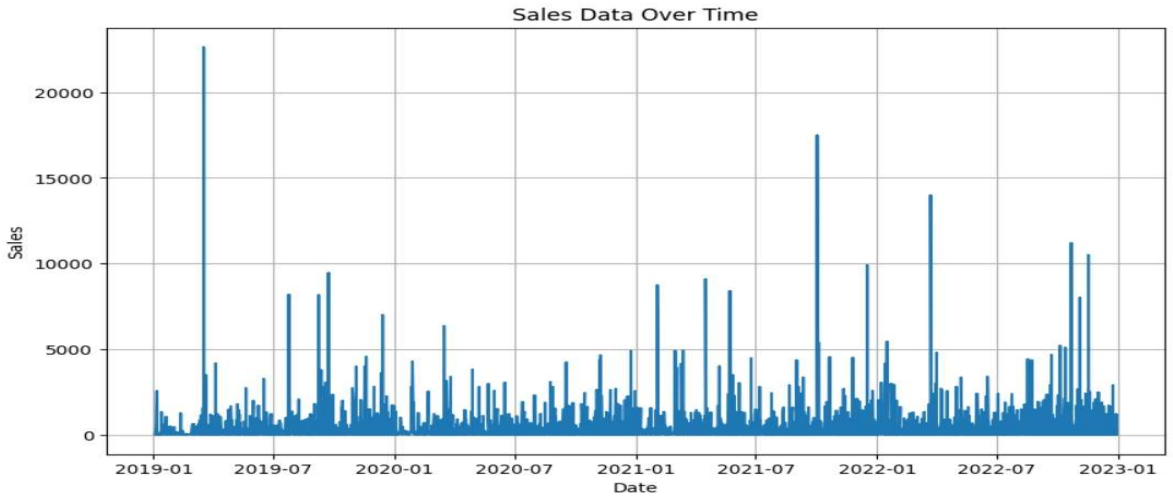
	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.0393	0.025	1.585	0.113	-0.009	0.088
ma.L1	-0.9582	0.009	-103.825	0.000	-0.976	-0.940
ar.S.L12	-0.0447	0.027	-1.636	0.102	-0.098	0.009
ma.S.L12	-0.9988	0.012	-82.088	0.000	-1.023	-0.975
sigma2	4.794e+06	2.62e-09	1.83e+15	0.000	4.79e+06	4.79e+06

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	25777.97
Prob(Q):	0.98	Prob(JB):	0.00
Heteroskedasticity (H):	1.34	Skew:	3.11
Prob(H) (two-sided):	0.00	Kurtosis:	22.74

Visualization of the performance of the model:

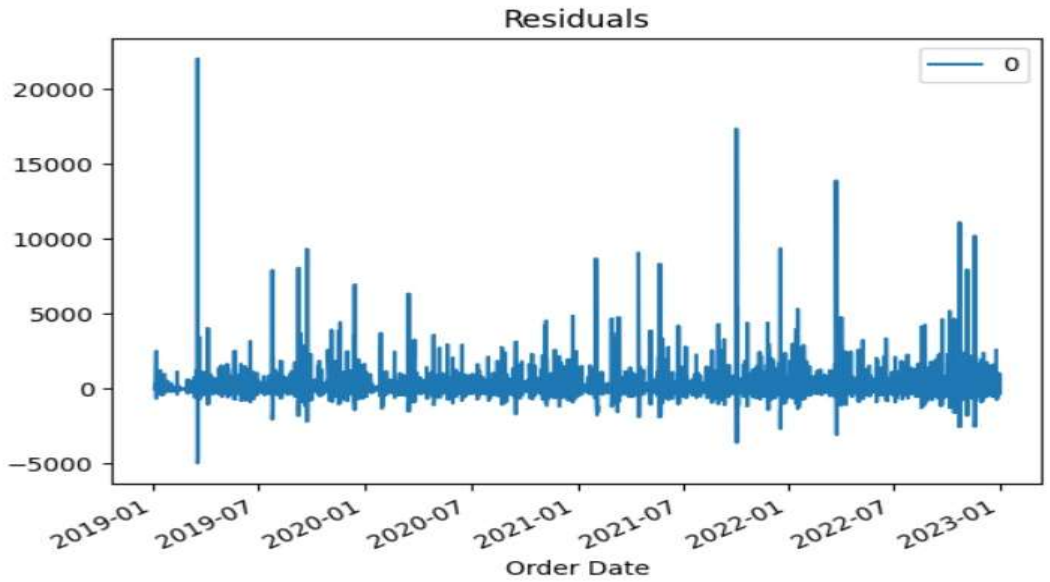
Sales Data Over Time Plot:

This plot illustrates the trend of sales over time. The x-axis represents the dates, while the y-axis represents the corresponding sales amounts. It provides a visual overview of the sales pattern throughout the observed period.



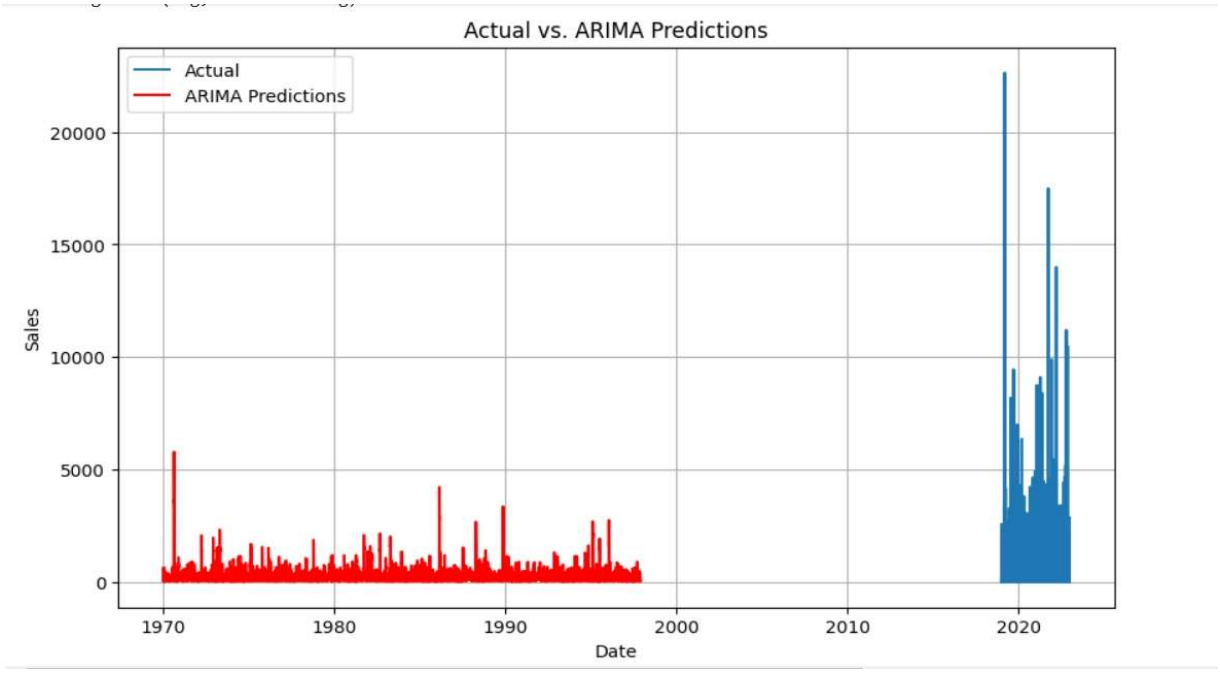
Residuals Plot:

This plot displays the residuals (the differences between the actual sales data and the ARIMA model's predictions) over time. Residuals represent the model's errors, and ideally, they should be random and centered around zero. This plot helps assess the model's assumption of homoscedasticity and detect any patterns or outliers in the residuals.



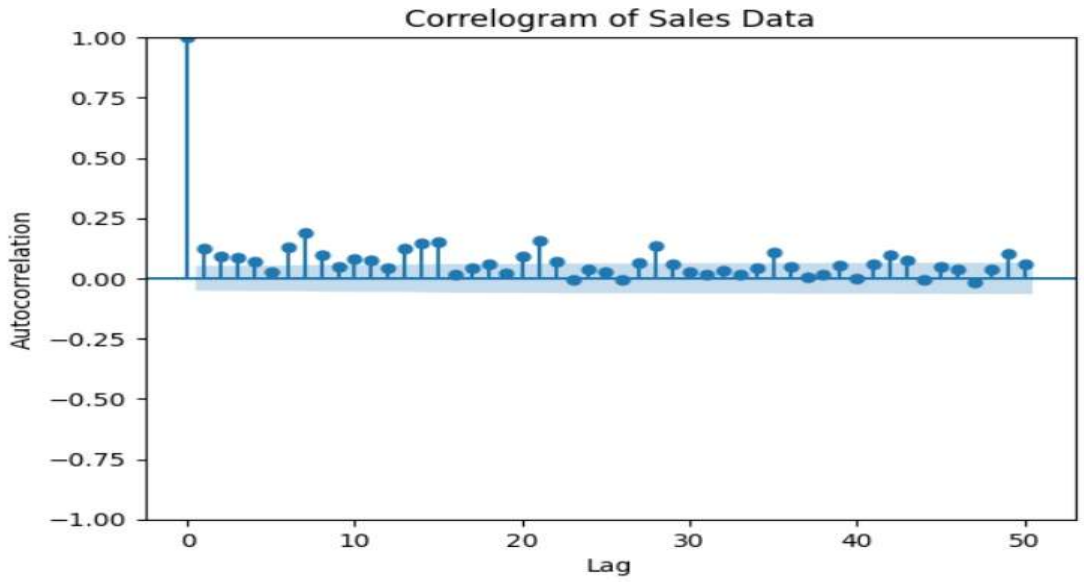
Actual vs. ARIMA Predictions Plot:

This plot compares the actual sales data with the predictions generated by the ARIMA model. The actual sales data are represented by the blue line, while the red line depicts the ARIMA model's predictions. By visually comparing the two series, we can evaluate the model's ability to capture the underlying patterns and dynamics of the sales data. It helps assess the accuracy and effectiveness of the ARIMA model in forecasting future sales.



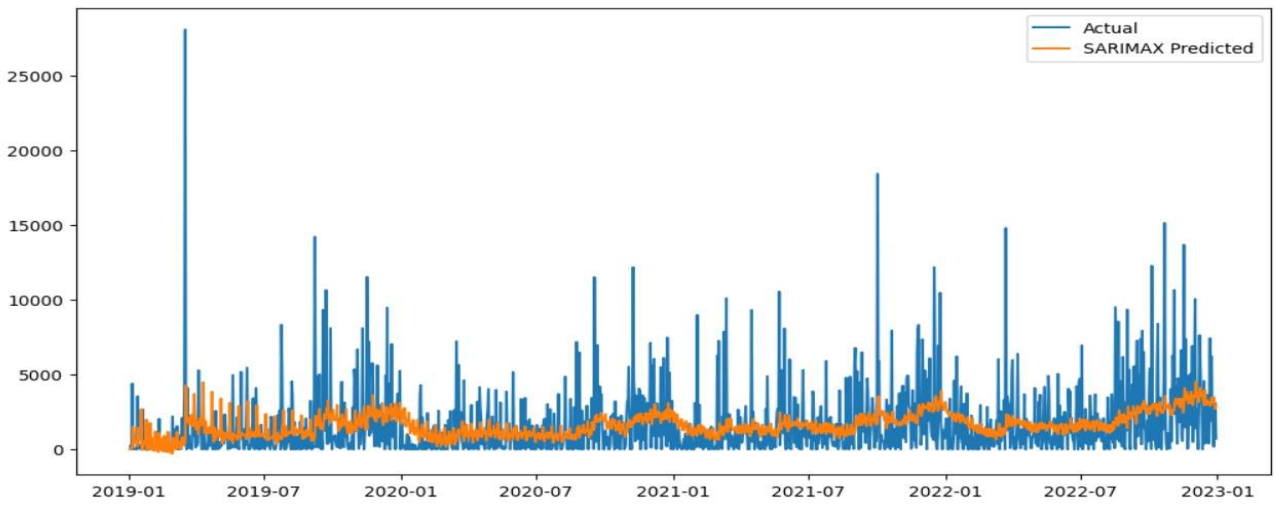
Correlogram of Sales Data:

The correlogram, also known as the autocorrelation plot, visualizes the autocorrelation function (ACF) of the sales data over different lag values. Each point on the plot represents the correlation between the sales data at time t and the sales data at a lagged time t-k, where k is the lag value on the x-axis. The correlogram helps identify any significant autocorrelation in the sales data, indicating whether there are repeating patterns or seasonality present in the time series.



Actual vs. SARIMAX Predicted Values Plot:

This plot compares the actual sales data with the predictions generated by the SARIMAX model. The actual sales data are represented by the blue line, while the orange line depicts the SARIMAX model's predictions. By visually comparing the two series, we can evaluate the model's ability to capture the underlying patterns and dynamics of the sales data. It helps assess the accuracy and effectiveness of the SARIMAX model in forecasting future sales.



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 Square Error (RMSE): 232.06058351718383
```

```
RMSE for SARIMAX model: 2743.232009487831
RMSE for XGBRegressor model: 2614.17732100618
```

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

Document Contribution:

B J Vasanth

CB.EN.U4CSE21213

- Handling Missing Values
- Creating Dummy Variables
- Stationarity Checks

B Lakshman Sai

CB.EN.U4CSE21234

- EDA with Static Visualization using Bokeh and Tableau
- Identifying Patterns

M Sasidhar

CB.EN.U4CSE21256

- Data Scaling/Transformation
- Dimensionality Reduction: Demonstrate at least 2 methods
- Content-Based Recommendation

ARIMA Modeling

Y G Ram Darshan Reddy

CB.EN.U4CSE21269

- Visualization
- Collaborative Filtering
- Prediction