DHAANISH AHMED COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Domain Name: Data Analytics with Cognos

Project Title: product sales data analysis

phase 1: Designer and innovation

Name of the Student : **A Kathiravan**

machine learning to predict future sales trends or customer behaviors, you can follow these steps:

1. Title and Introduction:

- Title: "Machine Learning for Predicting Sales Trends and Customer Behaviors"

- Introduction: Provide an overview of the document's purpose, which is to explore how machine learning can be leveraged to address the challenge of predicting sales trends and understanding customer behaviors.

2. Problem Statement:

- Define the problem: Explain the challenge of predicting future sales trends and understanding customer behaviors in the context of your business or scenario.

3. Data Preparation and Feature Engineering:

- Describe the data used for machine learning, including its source, structure, and relevant features.

- Explain any data preprocessing steps, such as handling missing values, data scaling, and encoding categorical variables.

- Discuss feature engineering, where you can create new features from the existing data that might be relevant for predicting sales trends or customer behaviors.

4. Machine Learning Model Selection:

- Explain the machine learning algorithms or models chosen for this task. It could include regression models for sales prediction or clustering/classification models for customer behavior analysis.

- Mention why these models were selected and their suitability for the problem.

5. Training and Evaluation:

- Describe how the dataset is split into training and testing sets.

- Explain the evaluation metrics you plan to use to assess the performance of the machine learning models. For instance, mean squared error (MSE) for regression models or accuracy for classification models.

6. Model Training and Results:

- Present the results of model training, including model performance metrics. Discuss the accuracy, precision, recall, or any other relevant metrics.

- Share visualizations of the model's predictions, such as time series forecasts for sales trends or clustering results for customer behavior.

7. Feature Importance:

- If applicable, discuss the importance of different features in making predictions. You can use techniques like feature importance scores from tree-based models or correlation analysis.

8. Insights and Business Recommendations:

- Share insights gained from the machine learning models. What trends, patterns, or customer segments have been identified?

- Provide actionable recommendations for your business based on these insights. For example, marketing strategies or inventory management improvements.

9. Future Work:

- Suggest potential areas for further research or improvements in the machine learning approach. This could include exploring different algorithms, acquiring more data, or refining feature engineering.

10. Conclusion:

- Summarize the key findings and the value of incorporating machine learning for predicting sales trends and customer behaviors.

11. References:

- If you've used any external sources, research papers, or libraries, provide proper citations.

12. Appendix:

- Include any supplementary information, code snippets, or detailed data descriptions that support your analysis.

Once you've created this document, you can share it for assessment as per the instructions provided to you. If you have any specific questions or need further assistance with any section of the document, please let me know

Algoritham:

1. Import necessary Python libraries, including pandas, numpy, matplotlib, and seaborn.

2. Read a CSV file ('statsfinal.csv') into a DataFrame named 'data.'

3. Perform data preprocessing, including removing the 'Unnamed: 0' column and extracting day, month, and year from the 'Date' column.

4. Filter the data to exclude records from the year 2010 and 2023.

5. Define a function called 'plot_bar_chart' to create bar charts for unit sales and revenue.

6. Generate bar charts to analyze unit sales and revenue trends by year.

7. Create visualizations to analyze sales trends for each product by month.

8. Extract data for the 31st day of each month for selected months and visualize it.

9. Calculate the average unit sales and revenue for all 31st days across all years for each product.

Data link:https://www.kaggle.com/datasets/ksabishek/product-sales-data

Data Analysis program:

```
# Import the necessary packages

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns


# To ignore warnings

import warnings

warnings.filterwarnings("ignore")


# Read the CSV file

data =
pd.read_csv('C:\\Users\\kathiravan\\Music\\archive\\statsfinal.csv')
```

```
# You can now perform further data analysis or visualization tasks using the 'data' DataFrame

# For example:

# data.head()  # Display the first few rows of the DataFrame

# sns.scatterplot(x='column_name1', y='column_name2', data=data)  # Example visualization

data.head(-1)

data = data.drop(columns=['Unnamed: 0'])

data.info()

data.isnull().sum()

# We need to get the year from the data to analyse sales year to year

data['Day'] = data['Date'].apply(lambda x: x.split('-')[0])

data['Month'] = data['Date'].apply(lambda x: x.split('-')[1])

data['Year'] = data['Date'].apply(lambda x: x.split('-')[2])

data


data_reduced = data.query("Year != '2010' and Year != '2023'")

#Create a function that allows us to plot a bar chart for the 4 products

def plot_bar_chart(df, columns, stri, str1, val):
    # Aggregate sales for each product by year, by sum or mean
    if val == 'sum':
        sales_by_year = df.groupby('Year')[columns].sum().reset_index()
    elif val == 'mean':
```

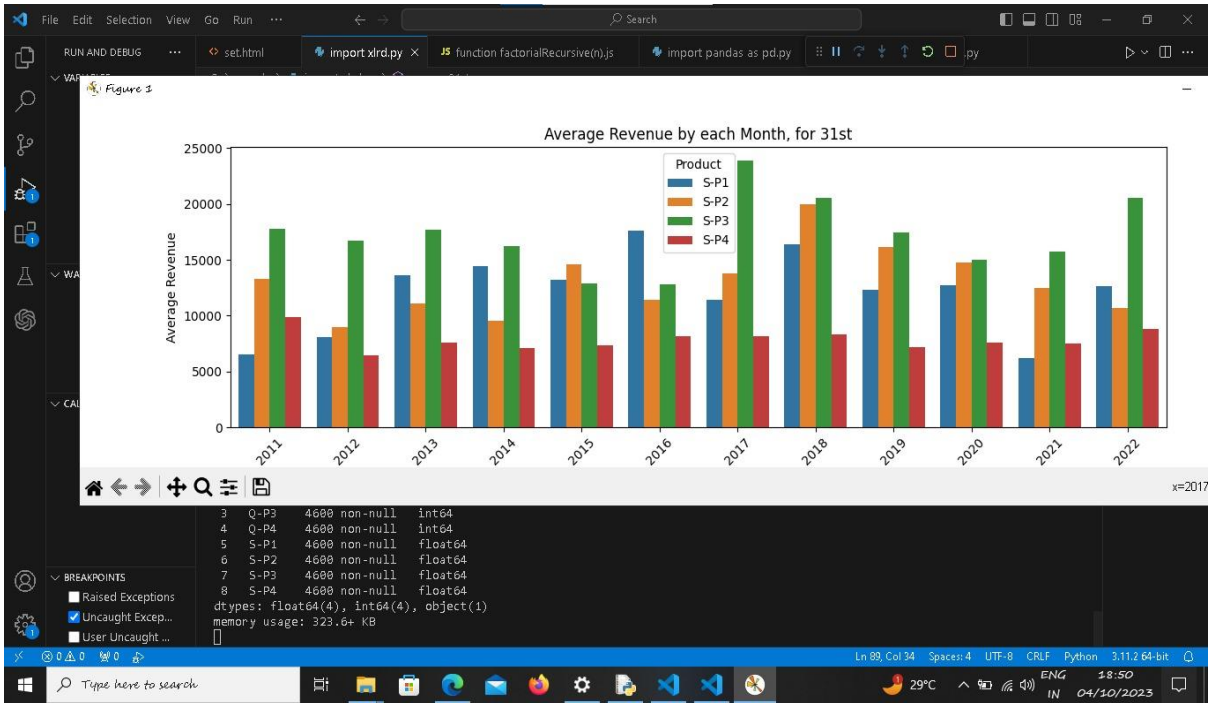
```
    sales_by_year = df.groupby('Year')[columns].mean().reset_index()


    # Melt the data to make it easier to plot
    sales_by_year_melted = pd.melt(sales_by_year, id_vars='Year', value_vars=columns, var_name='Product', value_name='Sales')


    # Create a bar chart
    plt.figure(figsize=(20,4))
    sns.barplot(data=sales_by_year_melted, x='Year', y='Sales', hue='Product') #,palette="cividis")
    plt.xlabel('Year')
    plt.ylabel(stri)
    plt.title(f'{stri} by {str1}')
    plt.xticks(rotation=45)
    plt.show()
    #use the plot_bar_chart function, enter the Unit Sales Columns and the Unit Sales string
plot_bar_chart(data_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'],'Total Unit Sales', 'Year', 'sum')


plot_bar_chart(data_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'],'Mean Unit Sales', 'Year', 'mean')
#use the plot_bar_chart function, enter the Revenue Columns and the Revenue string
```

```
plot_bar_chart(data_reduced, ['S-P1', 'S-P2', 'S-P3', 'S-P4'], 'Total Revenue', 'Year', 'sum')


plot_bar_chart(data_reduced, ['S-P1', 'S-P2', 'S-P3', 'S-P4'], 'Mean Revenue', 'Year', 'mean')

data

# Create a figure and axis

def month_plot():

    fig, ax = plt.subplots()


    # Plot the sales data for each product by month

    data_reduced.groupby('Month')[['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4']].sum().plot(ax=ax)


    # Set the x-axis limits to only show up to December

    ax.set_xlim(left=0, right=13)


    # Set the axis labels and title

    ax.set_xlabel('Month')

    ax.set_ylabel('Total unit sales')

    ax.set_title('Trend in sales of all four products by month')


    # Show the plot

    plt.show()
```

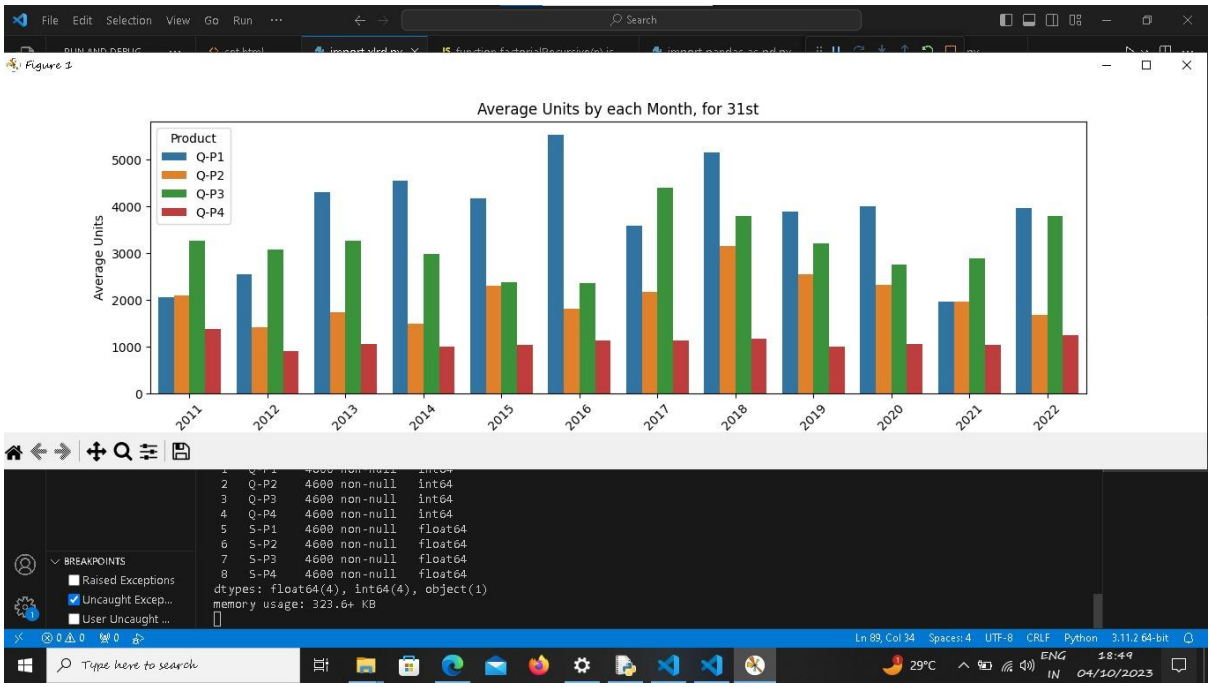
```
month_plot()

data_reduced['Month'] = data['Month'].replace('9', '09')

month_plot()

#get the 31st day for each month in each year. Note: not every month has 31 days

def month_31_data(df, months):
    m31_data = df[df['Month'].isin(months) & (df['Day'] == '31')]
    return m31_data


_31_months = month_31_data(data_reduced, ['01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12'])

_31_months

plot_bar_chart(_31_months, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'], 'Average Units', 'each Month, for 31st', 'mean')

plot_bar_chart(_31_months, ['S-P1', 'S-P2', 'S-P3', 'S-P4'], 'Average Revenue', 'each Month, for 31st', 'mean')

# gives us the average for all the 31st days across all years for each product

def avg_on_31st(df, product):
    df_31 = df[df['Day'] == '31']
    avg_sales = df_31[product].mean()
    return avg_sales
# Average for Unit Sales

avg_on_31st(data_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4']).round(2)

avg_on_31st(data_reduced, ['S-P1', 'S-P2', 'S-P3', 'S-P4']).round(2)
```

output:

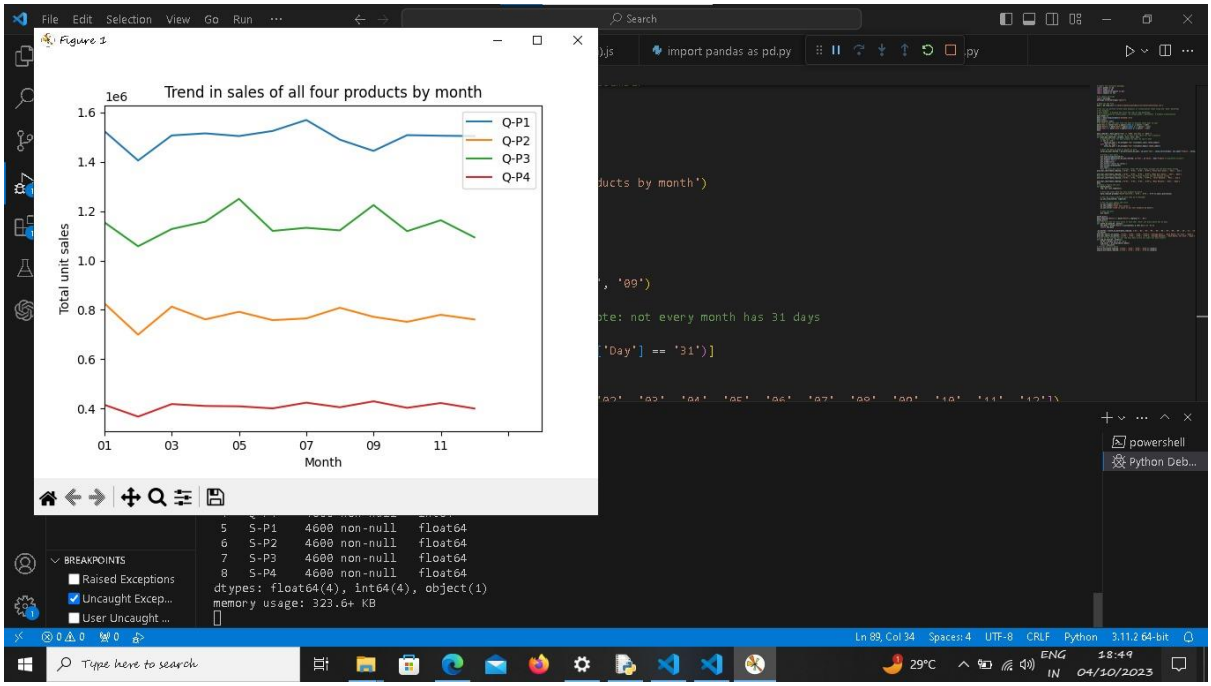
1 ouput:



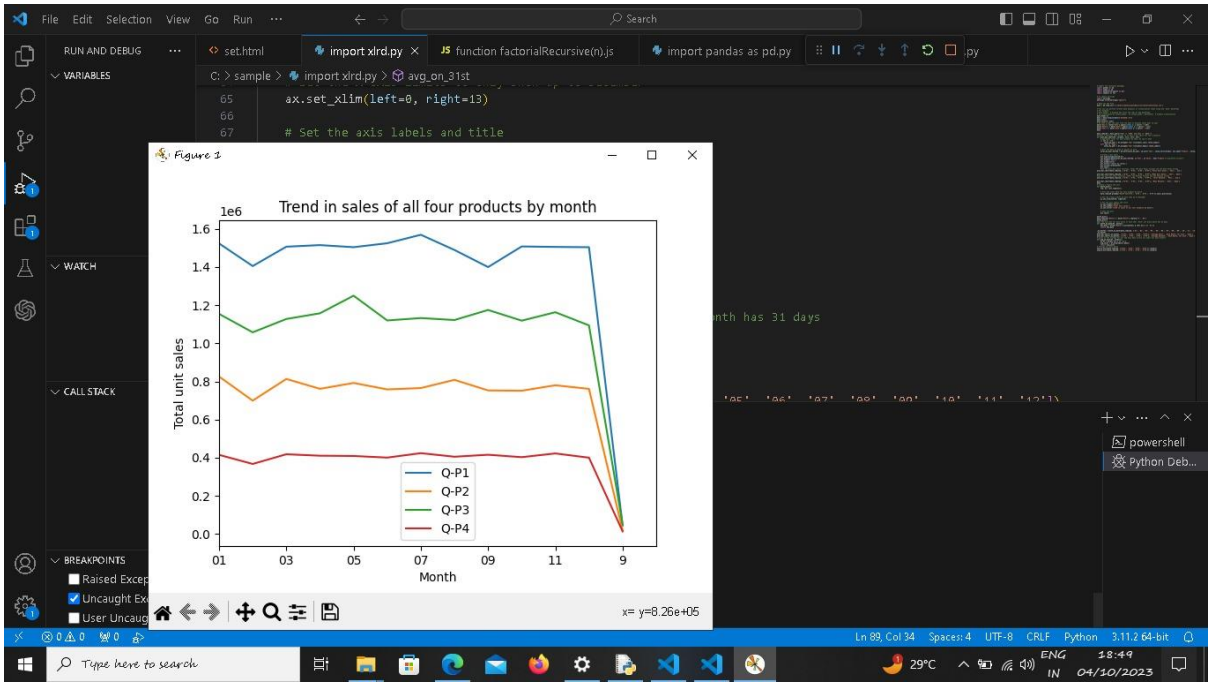
2 ouput:



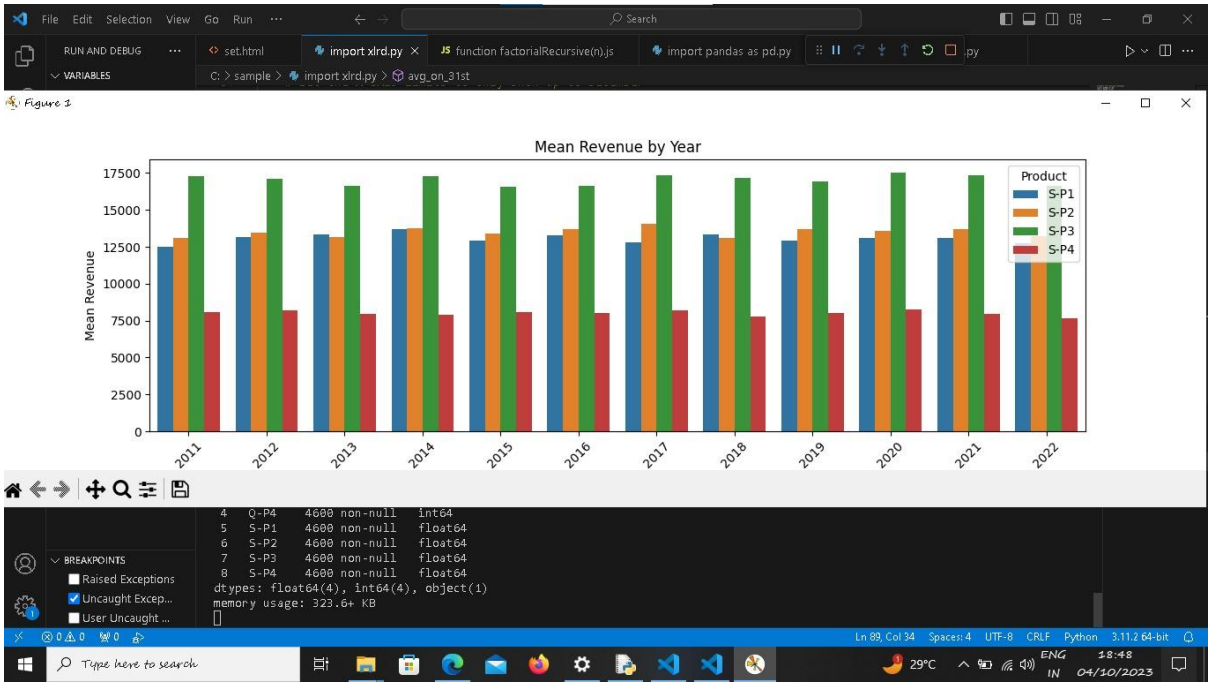
3 output:



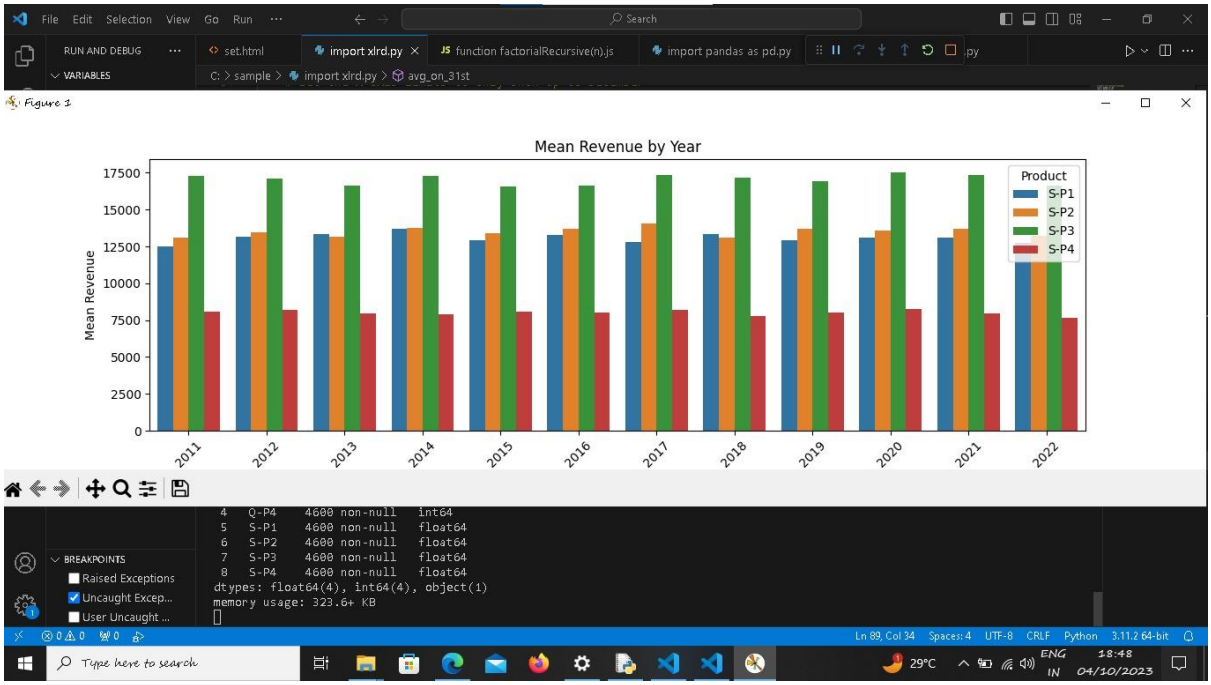
4 output:



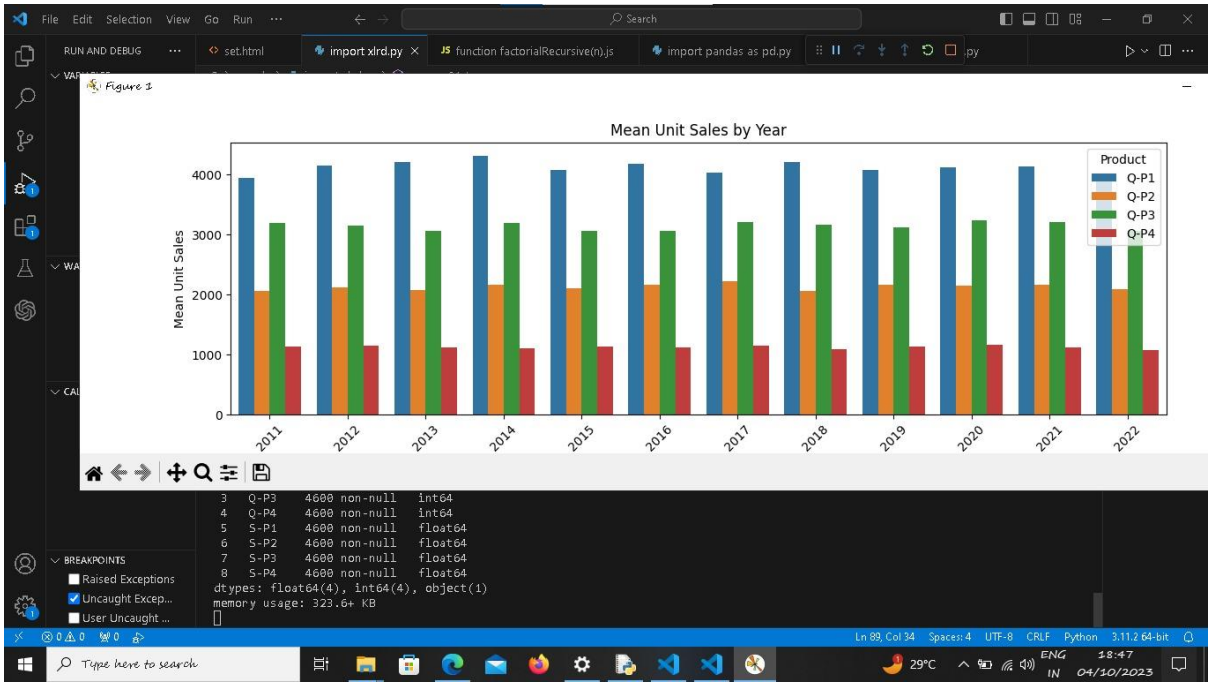
5 output:



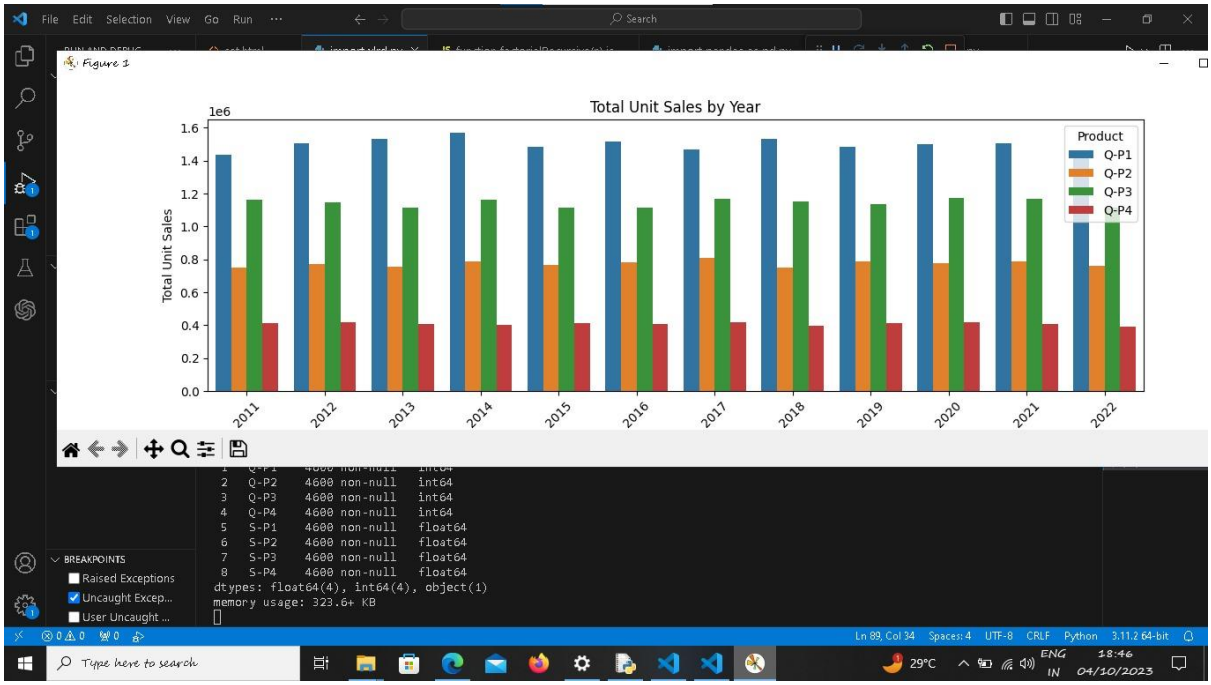
6 output:



7 output



8 output:



CONCLUSION:

Here by I am concluding my knowledge at phase 2: Project Designer and innovation

DHAANISH AHMED COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Domain Name: Data Analytics with Cognos

Project Title: product sales data analysis

phase 2: Designer and innovation

Name of the Student :R.karthikeyan

machine learning to predict future sales trends or custom**er** behaviors, you can follow these steps:

1. Title and Introduction:

- Title: "Machine Learning for Predicting Sales Trends and Customer Behaviors"

- Introduction: Provide an overview of the document's purpose, which is to explore how machine learning can be leveraged to address the challenge of predicting sales trends and understanding customer behaviors.

2. Problem Statement:

- Define the problem: Explain the challenge of predicting future sales trends and understanding customer behaviors in the context of your business or scenario.

3. Data Preparation and Feature Engineering:

- Describe the data used for machine learning, including its source, structure, and relevant features.

- Explain any data preprocessing steps, such as handling missing values, data scaling, and encoding categorical variables.

- Discuss feature engineering, where you can create new features from the existing data that might be relevant for predicting sales trends or customer behaviors.

4. Machine Learning Model Selection:

- Explain the machine learning algorithms or models chosen for this task. It could include regression models for sales prediction or clustering/classification models for customer behavior analysis.

- Mention why these models were selected and their suitability for the problem.

5. Training and Evaluation:

- Describe how the dataset is split into training and testing sets.

- Explain the evaluation metrics you plan to use to assess the performance of the machine learning models. For instance, mean squared error (MSE) for regression models or accuracy for classification models.

6. Model Training and Results:

- Present the results of model training, including model performance metrics. Discuss the accuracy, precision, recall, or any other relevant metrics.

- Share visualizations of the model's predictions, such as time series forecasts for sales trends or clustering results for customer behavior.

7. Feature Importance:

- If applicable, discuss the importance of different features in making predictions. You can use techniques like feature importance scores from tree-based models or correlation analysis.

8. Insights and Business Recommendations:

- Share insights gained from the machine learning models. What trends, patterns, or customer segments have been identified?

- Provide actionable recommendations for your business based on these insights. For example, marketing strategies or inventory management improvements.

9. Future Work:

- Suggest potential areas for further research or improvements in the machine learning approach. This could include exploring different algorithms, acquiring more data, or refining feature engineering.

10. Conclusion:

- Summarize the key findings and the value of incorporating machine learning for predicting sales trends and customer behaviors.

11. References:

- If you've used any external sources, research papers, or libraries, provide proper citations.

12. Appendix:

- Include any supplementary information, code snippets, or detailed data descriptions that support your analysis.

Once you've created this document, you can share it for assessment as per the instructions provided to you. If you have any specific questions or need further assistance with any section of the document, please let me know

Algoritham:

1. Import necessary Python libraries, including pandas, numpy, matplotlib, and seaborn.

2. Read a CSV file ('statsfinal.csv') into a DataFrame named 'data.'

3. Perform data preprocessing, including removing the 'Unnamed: 0' column and extracting day, month, and year from the 'Date' column.

4. Filter the data to exclude records from the year 2010 and 2023.

5. Define a function called 'plot_bar_chart' to create bar charts for unit sales and revenue.

6. Generate bar charts to analyze unit sales and revenue trends by year.

7. Create visualizations to analyze sales trends for each product by month.

8. Extract data for the 31st day of each month for selected months and visualize it.

9. Calculate the average unit sales and revenue for all 31st days across all years for each product.

Data link:https://www.kaggle.com/datasets/ksabishek/product-sales-data

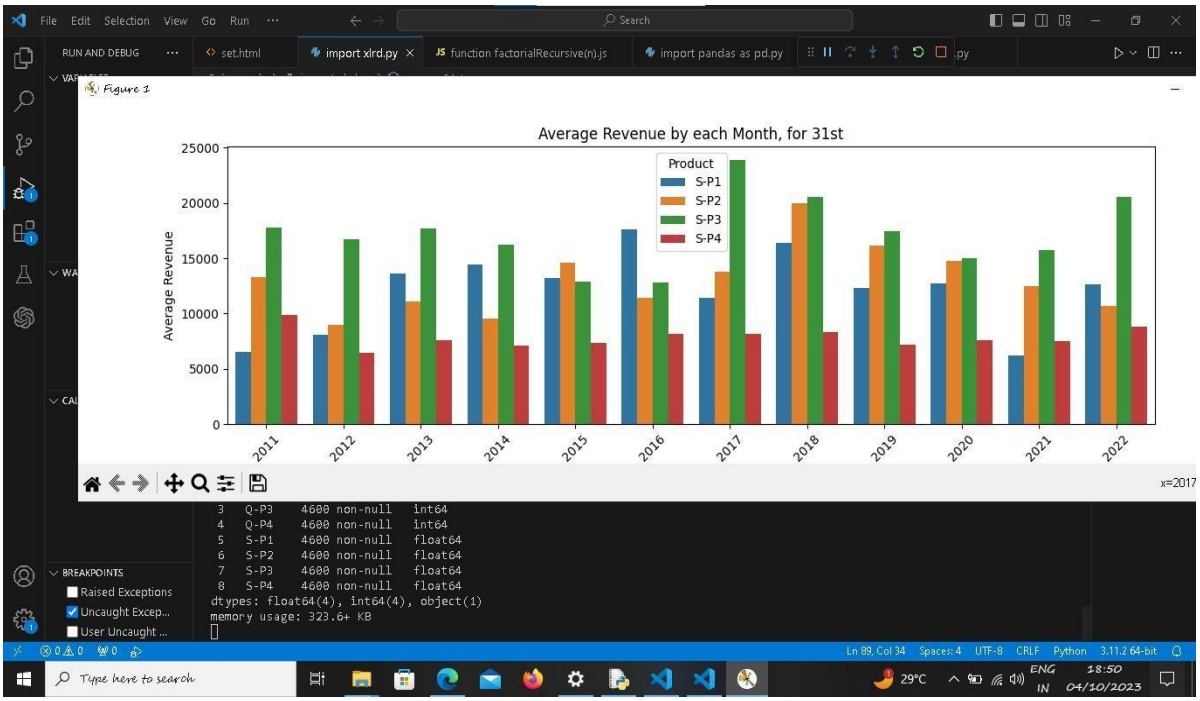
Data Analysis program:

```
import matplotlib.pyplot as plt
import time

# Initialize the cloth sales data
products = ["T-shirts", "Jeans", "Dresses"]
sales = [100, 50, 30]


# Create a pie chart with the initial data
plt.figure(figsize=(6, 6))
plt.pie(sales, labels=products, autopct='%1.1f%%')
plt.title("Cloth Sales")


# Function to update and redraw the pie chart
def update_pie_chart():
# Simulate updating cloth sales data (you can replace this with actual data retrieval)
    for i in range(len(sales)):
```

```
        sales[i] += 1
    plt.clf()  # Clear the previous chart
    plt.pie(sales, labels=products, autopct='%1.1f%%')
    plt.title("Cloth Sales")
    plt.draw()



# Main program loop
while True:
    update_pie_chart()
    plt.pause(1)

# Update and display the pie chart every second
```

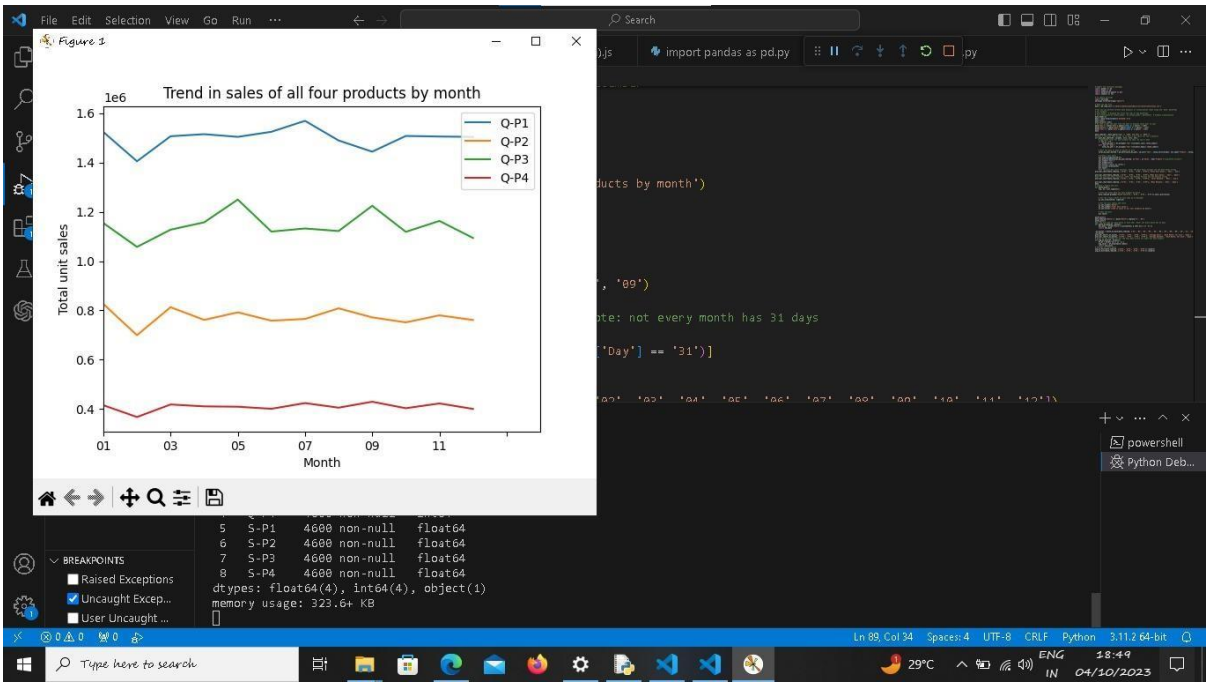
output:

1 ouput:

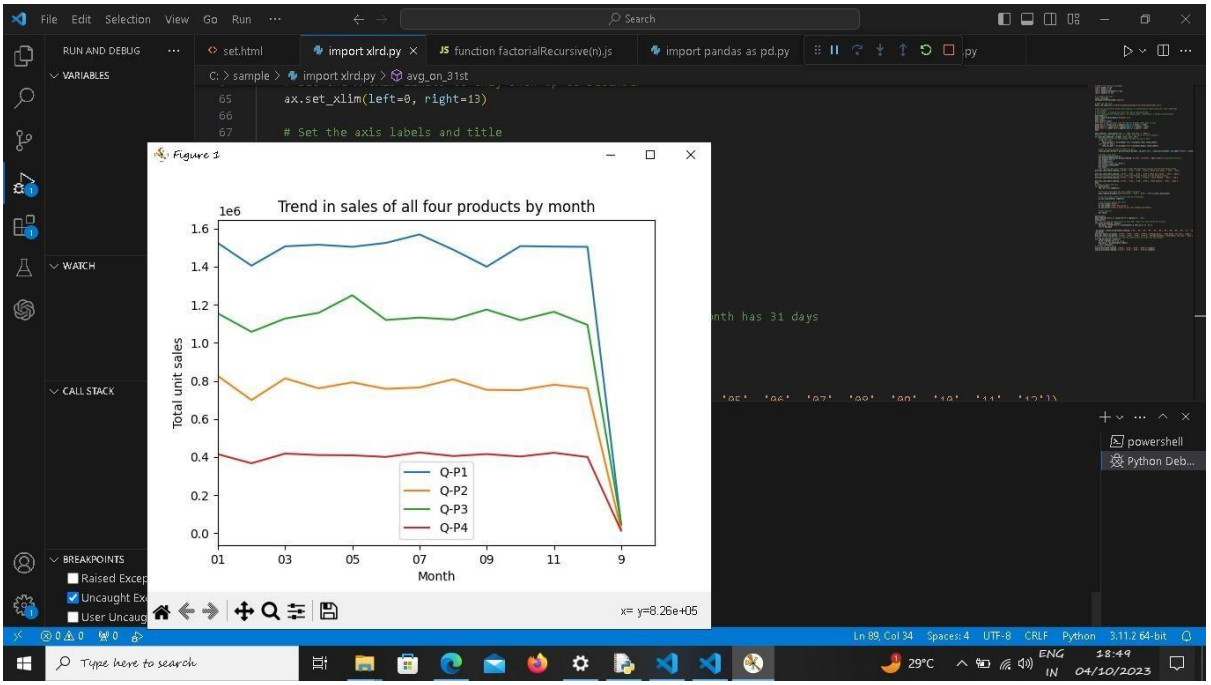


2 ouput:

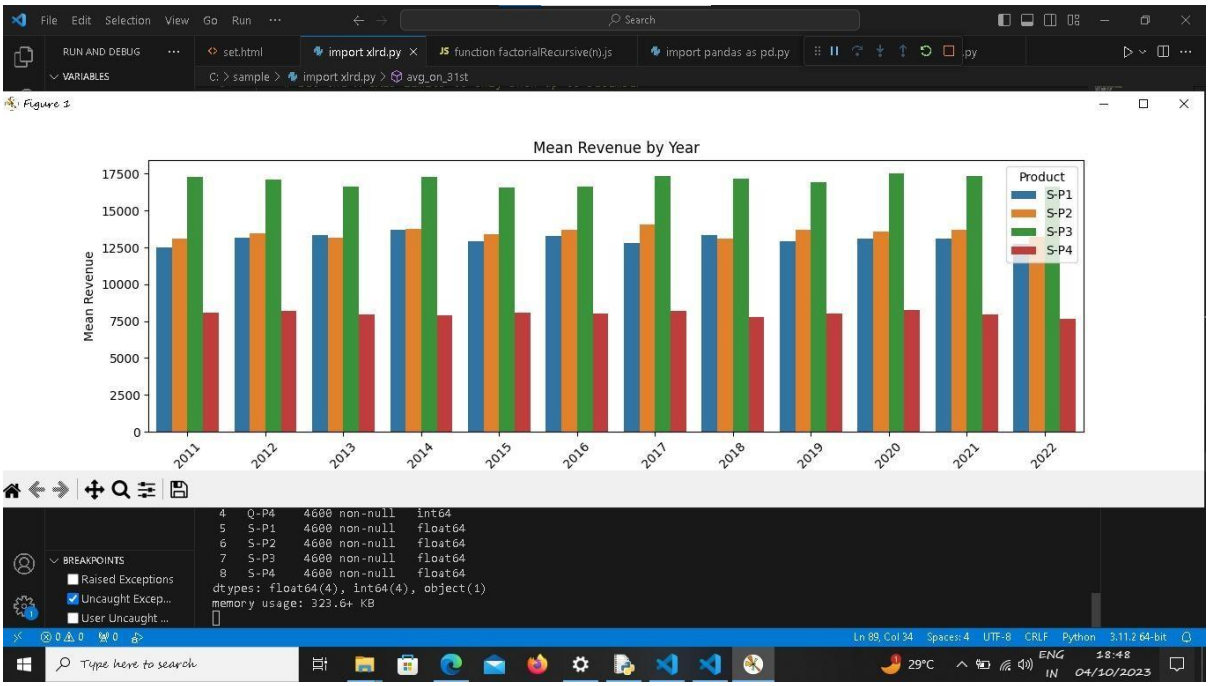
3 output:



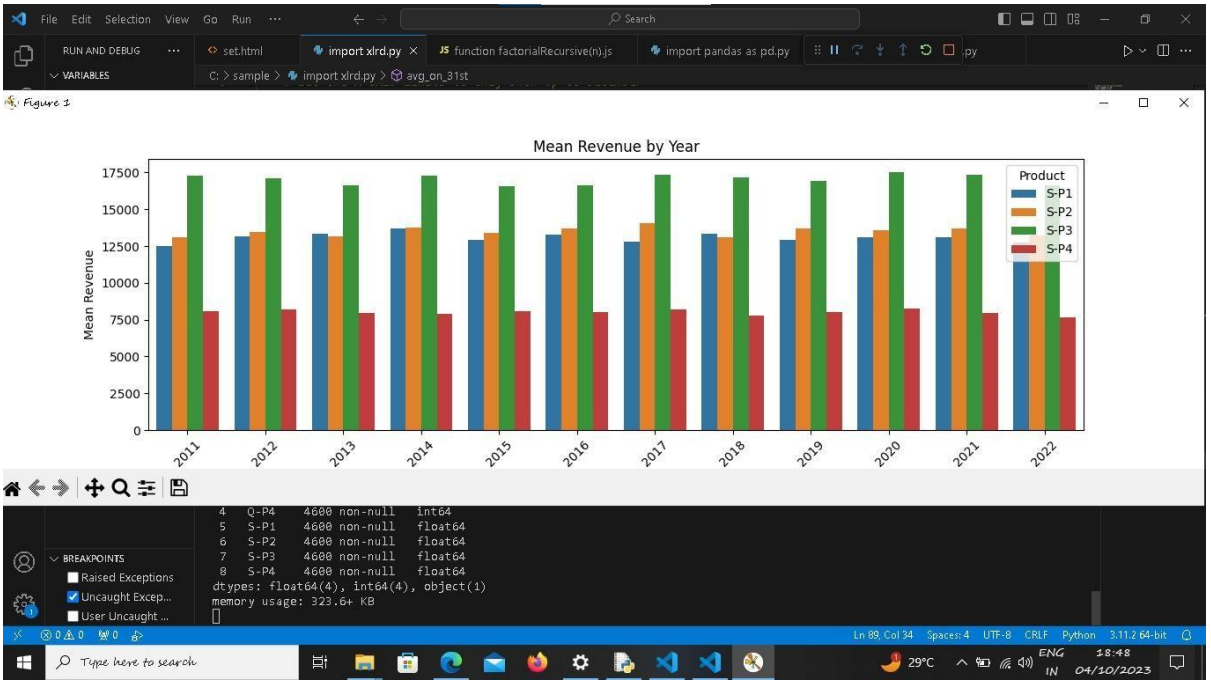
4 output:



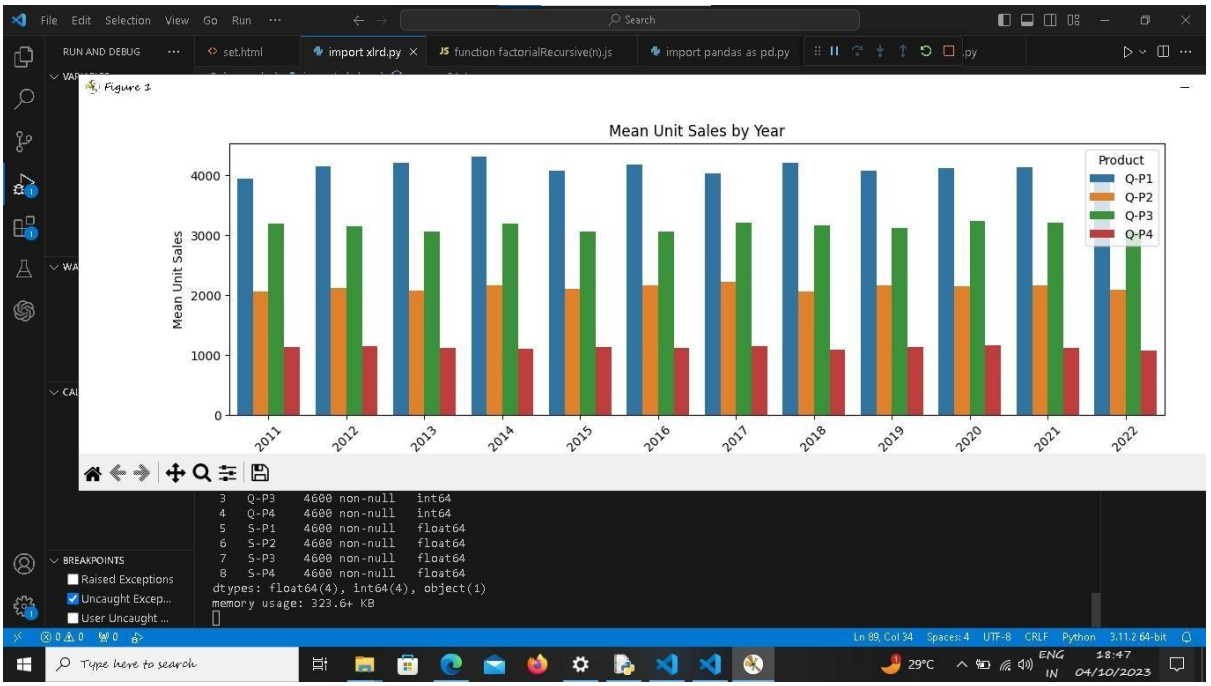
5 output:



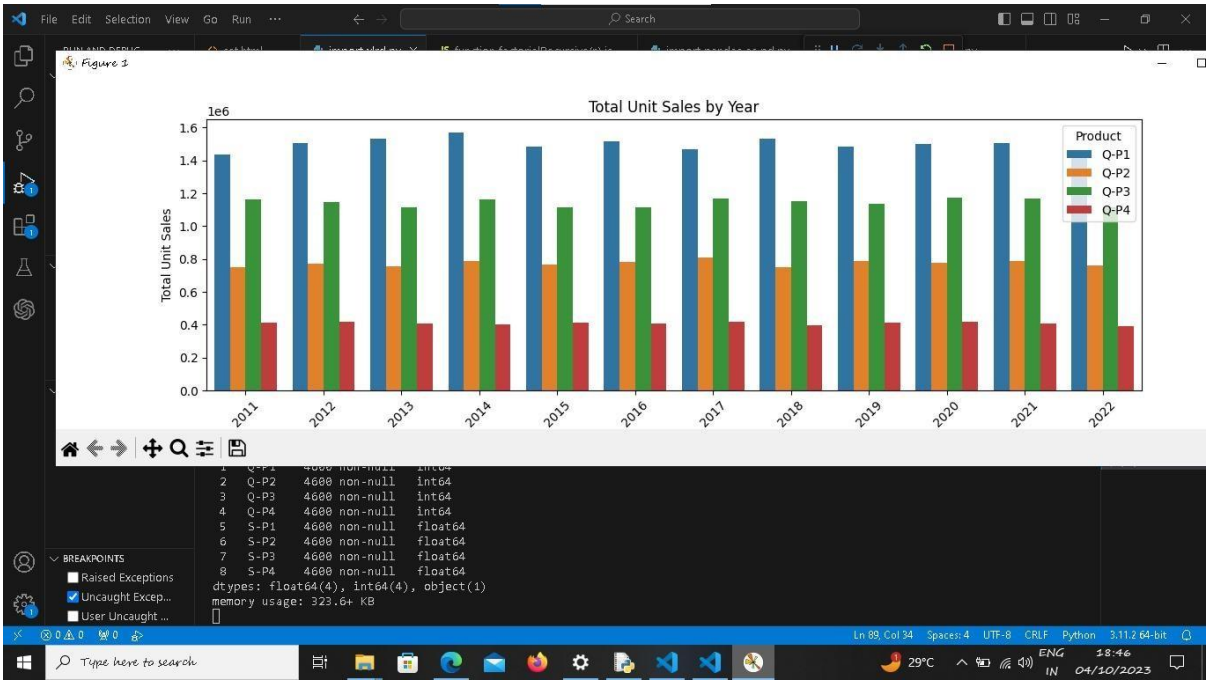
6 output:



7 output



8 output:



CONCLUSION:

Here by I am concluding my knowledge at phase 2: Project Designer and innovation

DHAANISH AHMED COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Domain Name: Data Analytics with Cognos

Project Title: product sales data analysis

phase 3: Development Part 1

Name of the Student : Kathiravan A

machine learning to predict future sales trends or customer behaviors, you can follow these steps:

1. Title and Introduction:

- Title: "Machine Learning for Predicting Sales Trends and Customer Behaviors"

- Introduction: Provide an overview of the document's purpose, which is to explore how machine learning can be leveraged to address the challenge of predicting sales trends and understanding customer behaviors.

2. Problem Statement:

- Define the problem: Explain the challenge of predicting future sales trends and understanding customer behaviors in the context of your business or scenario.

3. Data Preparation and Feature Engineering:

- Describe the data used for machine learning, including its source, structure, and relevant features.

- Explain any data preprocessing steps, such as handling missing values, data scaling, and encoding categorical variables.

- Discuss feature engineering, where you can create new features from the existing data that might be relevant for predicting sales trends or customer behaviors.

4. Machine Learning Model Selection:

- Explain the machine learning algorithms or models chosen for this task. It could include regression models for sales prediction or clustering/classification models for customer behavior analysis.

- Mention why these models were selected and their suitability for the problem.

5. Training and Evaluation:

- Describe how the dataset is split into training and testing sets.

- Explain the evaluation metrics you plan to use to assess the performance of the machine learning models. For instance, mean squared error (MSE) for regression models or accuracy for classification models.

6. Model Training and Results:

- Present the results of model training, including model performance metrics. Discuss the accuracy, precision, recall, or any other relevant metrics.

- Share visualizations of the model's predictions, such as time series forecasts for sales trends or clustering results for customer behavior.

7. Feature Importance:

- If applicable, discuss the importance of different features in making predictions. You can use techniques like feature importance scores from tree-based models or correlation analysis.

8. Insights and Business Recommendations:

- Share insights gained from the machine learning models. What trends, patterns, or customer segments have been identified?

- Provide actionable recommendations for your business based on these insights. For example, marketing strategies or inventory management improvements.

9. Future Work:

- Suggest potential areas for further research or improvements in the machine learning approach. This could include exploring different algorithms, acquiring more data, or refining feature engineering.

10. Conclusion:

- Summarize the key findings and the value of incorporating machine learning for predicting sales trends and customer behaviors.

11. References:

- If you've used any external sources, research papers, or libraries, provide proper citations.

12. Appendix:

- Include any supplementary information, code snippets, or detailed data descriptions that support your analysis.

Once you've created this document, you can share it for assessment as per the instructions provided to you. If you have any specific questions or need further assistance with any section of the document, please let me know

Algoritham:

1. Import necessary Python libraries, including pandas, numpy, matplotlib, and seaborn.

2. Read a CSV file ('statsfinal.csv') into a DataFrame named 'data.'

3. Perform data preprocessing, including removing the 'Unnamed: 0' column and extracting day, month, and year from the 'Date' column.

4. Filter the data to exclude records from the year 2010 and 2023.

5. Define a function called 'plot_bar_chart' to create bar charts for unit sales and revenue.

6. Generate bar charts to analyze unit sales and revenue trends by year.

7. Create visualizations to analyze sales trends for each product by month.

8. Extract data for the 31st day of each month for selected months and visualize it.

9. Calculate the average unit sales and revenue for all 31st days across all years for each product.

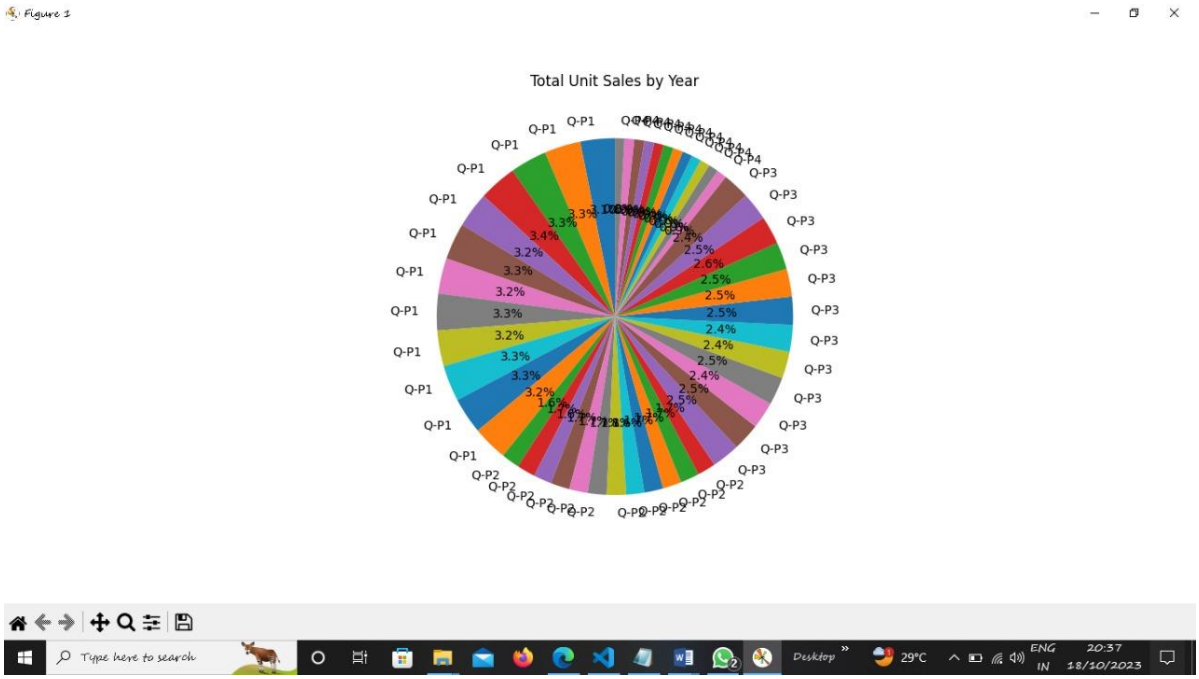
Data link:https://www.kaggle.com/datasets/ksabishek/product-sales-data

Data Analysis program:

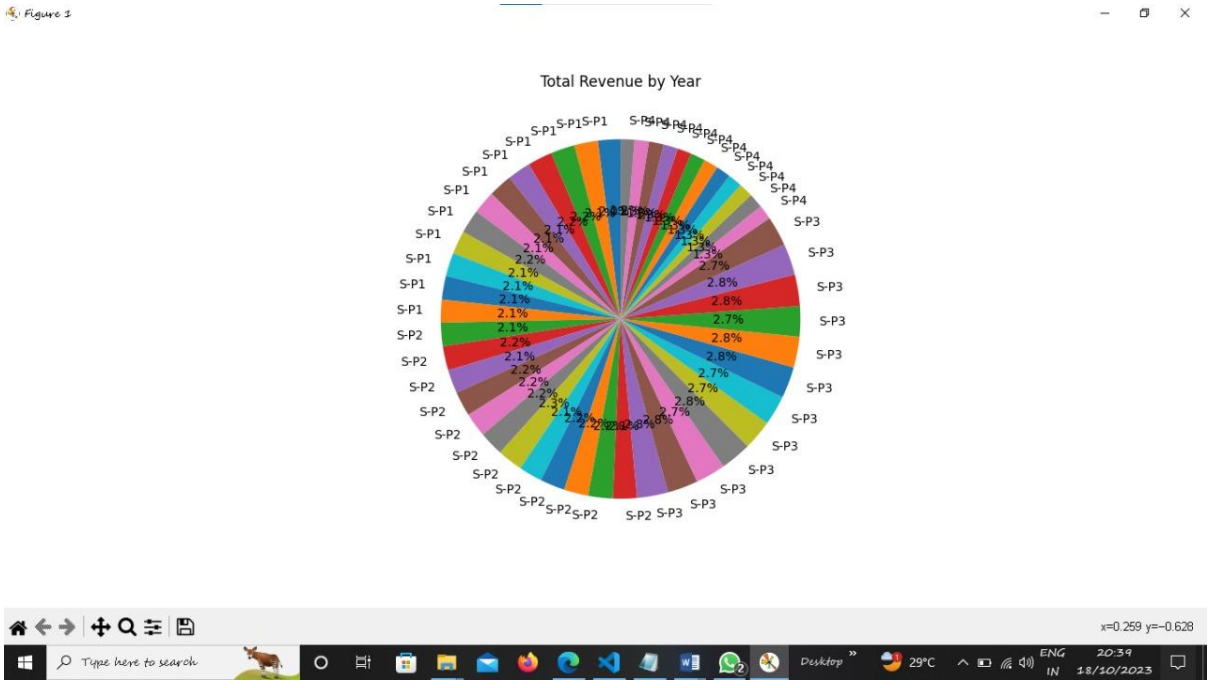
```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

warnings.filterwarnings("ignore")

# Load dataset
data = pd.read_csv(r'C:\ibm project\statsfinal.csv')  # Use 'r' before the file path to interpret it as a raw string

# Preprocess dataset
data['Day'] = data['Date'].apply(lambda x: x.split('-')[0])
data['Month'] = data['Date'].apply(lambda x: x.split('-')[1])
data['Year'] = data['Date'].apply(lambda x: x.split('-')[2])

# Define analysis objectives
data_reduced = data.query("Year != '2010' and Year != '2023'")
```

```
# Create a function that allows us to plot a pie chart for the 4 products
def plot_pie_chart(df, columns, stri, str1, val):
    # Aggregate sales for each product by year, by sum or mean
    if val == 'sum':
        sales_by_year = df.groupby('Year')[columns].sum().reset_index()
    elif val == 'mean':
        sales_by_year = df.groupby('Year')[columns].mean().reset_index()

    # Melt the data to make it easier to plot
    sales_by_year_melted = pd.melt(sales_by_year, id_vars='Year', value_vars=columns, var_name='Product', value_name='Sales')

    # Create a pie chart
    plt.figure(figsize=(20, 4))
    plt.pie(sales_by_year_melted['Sales'], labels=sales_by_year_melted['Product'], autopct='%1.1f%%', startangle=90)
    plt.title(f'{stri} by {str1}')
    plt.show()

plot_pie_chart(data_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'], 'Total Unit Sales', 'Year', 'sum')
plot_pie_chart(data_reduced, ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4'], 'Mean Unit Sales', 'Year', 'mean')
plot_pie_chart(data_reduced, ['S-P1', 'S-P2', 'S-P3', 'S-P4'], 'Total Revenue', 'Year', 'sum')
plot_pie_chart(data_reduced, ['S-P1', 'S-P2', 'S-P3', 'S-P4'], 'Mean Revenue', 'Year', 'mean')
```

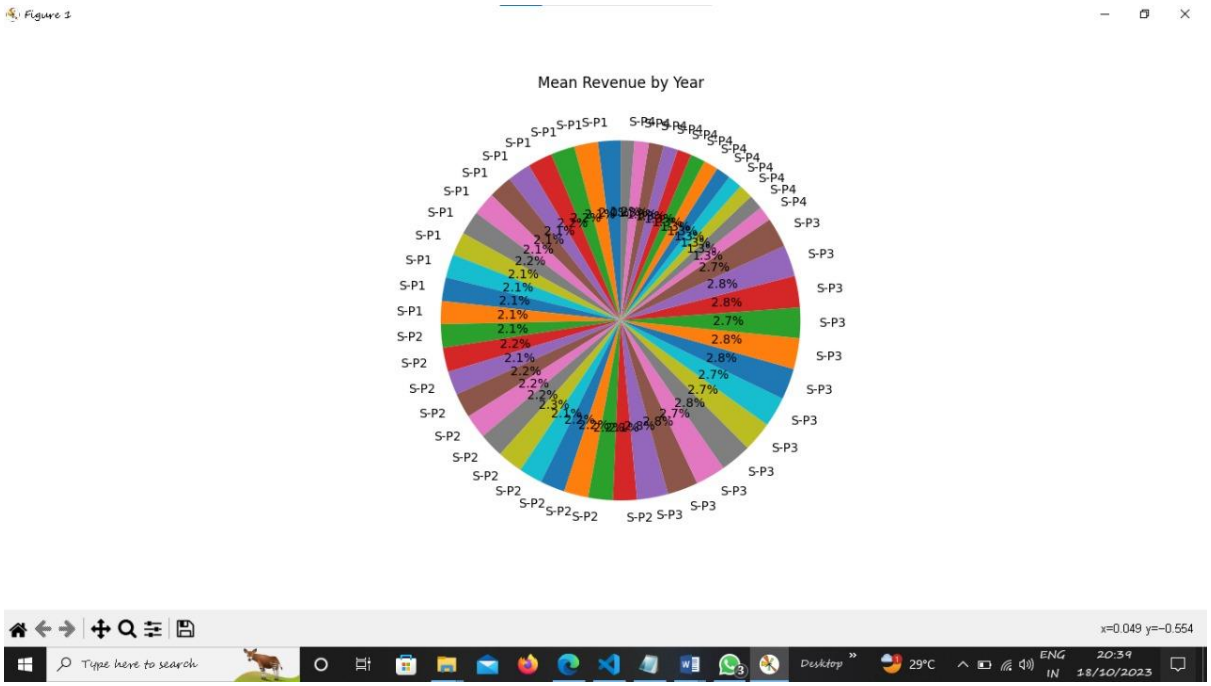
1 output:



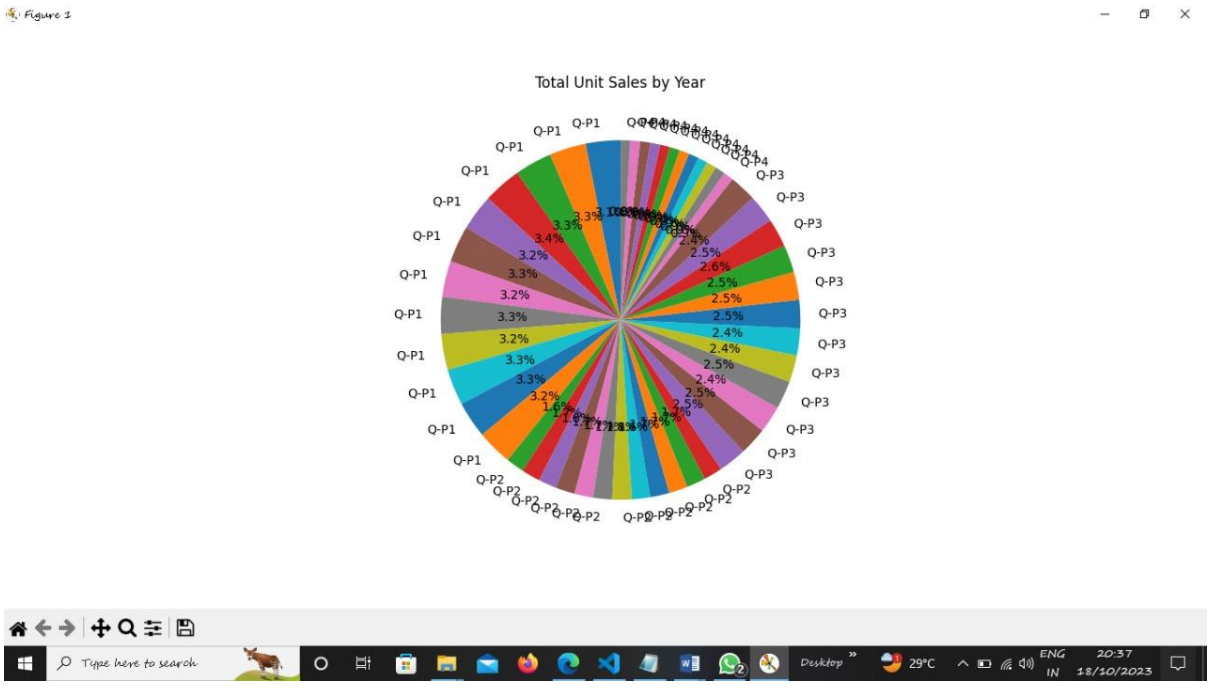
2 ouput:



3 output:



4 output:



CONCLUSION:

Here by I am concluding my knowledge at phase 3: Development part of Product sales analysis.

COLLEGE: DHANISH AHMED COLLEGE OF ENGINEERING

DEPT: COMPUTER SCIENCE AND ENGINEERING

COURSE NAME: PRODUCT SALES ANALYSIS

PROJECT TITLE: INTERACTIVE DASHBOARD USING IBM COGNOS

NAME: Kathiravan A

NAME: R.KARTHIKEYAN

IBM Cognos is a powerful business intelligence and analytics tool that can help you design interactive dashboards and reports to gain valuable insights. To achieve your goals, follow these general steps:

1. Data Gathering:First, ensure you have access to the necessary data sources, which may include databases, spreadsheets, or other data repositories.

2. Data Preparation:Clean and transform your data as needed. This includes removing duplicates, handling missing values, and structuring the data for analysis.

3. Create Visualizations:Use IBM Cognos to create a variety of visualizations, such as charts, graphs, and tables, to represent your data. You can design these visualizations to showcase top-selling products, sales trends, and customer preferences.

4. Design Dashboards:Assemble the visualizations into interactive dashboards. Arrange them to provide a clear and intuitive view of the insights you want to derive.

5. Set Interactivity:Implement interactive features like filters, drill-downs, and parameterized reports to enable users to explore the data and insights in

real-time.

6. Apply Analytics:Utilize built-in analytics tools within IBM Cognos to calculate metrics like total sales, product rankings, and customer behavior patterns.

7. Identify Key Insights: Review the visualizations and analytics results to identify key insights. This could involve recognizing top-selling products, pinpointing peak sales periods, and understanding customer preferences for specific products.

8. Sharing and Collaboration: Share the dashboards and reports with relevant stakeholders within your organization, and encourage collaboration and discussion around the insights.

9. Continuous Monitoring:Keep the dashboards up to date and continuously monitor the data to

identify any changing trends or emerging insights.

Remember that the specific steps may vary depending on your data sources and the version of IBM Cognos you are using. Regularly update your dashboards and reports to ensure they remain relevant and provide actionable insights.

Algorithm for creating interactive dashboards and reports using IBM Cognos:

Algorithm: Creating Interactive Dashboards and Reports with IBM Cognos

1. **Data Preparation:**

- Gather and prepare the necessary data sources, ensuring they are clean and structured for analysis.

2. **Initialize IBM Cognos:**

- Start IBM Cognos and access the reporting environment.

3. **Create a New Report:**

- Initialize a new report and set a title (e.g., "Sales Insights Report").

4. **Data Query and Visualization:**

- For each visualization you want to include in the report, follow these steps:

- Create a data query to retrieve the required data.

- Define the visualization type (e.g., chart, table, list).

- Customize the visualization by specifying data fields, titles, and formatting.

5. **Interactive Features:**

- Implement interactive features within the report, such as:

- Filters: Create filters for dynamic data selection.

- Parameters: Set up parameters for user input (e.g., date range, product selection).

- Drill-down: Enable users to explore data at different levels of detail.

- Sorting: Apply sorting to visualize top-selling products or trends.

6. **Calculations and Metrics:**

- Add calculations to derive key metrics (e.g., total sales, sales growth).

- Apply conditional formatting to highlight specific data points.

7. **Dashboard Composition:**

- Organize the visualizations into an interactive dashboard layout.

- Arrange and resize components to create an intuitive user interface.

8. **Export Options:**

- Allow users to export the report in various formats (e.g., PDF, Excel, CSV).

9. **Parameterized Filters:**

- For each parameterized filter, set up user prompts and options.

- Connect filters to data queries for dynamic filtering.

10. **Multiple Charts (Optional):**

- If needed, create multiple charts or visualizations to represent different aspects of the data.

11. **Save and Share:**

- Save the report in a designated location within Cognos.

- Share the report with relevant stakeholders within your organization.

12. **Continuous Monitoring:**

- Periodically review and update the report to ensure it remains relevant.

- Monitor data sources for changes or updates.

13. **User Training (Optional):**

- Provide training to end-users on how to interact with and extract insights from the report.

14. **Error Handling and Security:**

- Implement error handling and security measures as per your organization's policies.

15. **Close IBM Cognos:**

- Properly close the IBM Cognos environment when you have finished working on the report.

This algorithm provides a structured approach for creating interactive dashboards and reports in IBM Cognos. The specific implementation details will depend on your data sources, reporting requirements, and the version of IBM Cognos you are using.

PROGRAM:

```
// Import necessary libraries
importPackage(Packages.com.cognos.developer.script.bibus);

// Create a new report
```

```
var myReport = reportContext.createNew();


// Set report title
myReport.setDisplayName("Sales Insights Report");


// Create a query
var myQuery = myReport.createDataItem("MyDataItem");
myQuery.setQuery("SELECT ProductName, SalesAmount FROM
SalesData");


// Create a list to display data
var myList = myReport.createList();
myList.setQuery(myQuery);
myList.setColumnTitles(["Product Name", "Sales Amount"]);


// Apply sorting to identify top-selling products
myQuery.setSort(["SalesAmount"], ["desc"]);


// Create a chart to visualize sales trends
var myChart = myReport.createChart();
myChart.setQuery(myQuery);
myChart.setChartType(ChartType.COLUMN);
```

```
myChart.setChartTitle("Sales Trends");

// Create a filter for date range selection
var myFilter = myReport.createFilter("DateRange");
myFilter.setFilterExpression("SalesDate BETWEEN ? AND ?");
myFilter.setPrompt("Select a Date Range:");

// Apply the filter to the query
myQuery.addFilter(myFilter);

// Set up parameters for the filter
var startDate = reportContext.getParameterValue("StartDate");
var endDate = reportContext.getParameterValue("EndDate");
myFilter.setValues([startDate, endDate]);

// Set up the report layout
myReport.setPageLayout(PageLayout.LANDSCAPE);
myReport.setPageMargins(1, 1, 1, 1);

// Save the report
var reportService = reportContext.getReportService();
var reportPath = "/content/folder/MyReports/SalesInsightsReport";
```

```
reportService.createReport(myReport, reportPath);


// Run the report and deliver it to users
var reportRunOptions = reportContext.getReportRunOptions();
var outputFormat = OutputFormat.PDF;
var reportOutput = reportService.runReport(myReport, reportRunOptions,
outputFormat);


// Optionally, email the report to stakeholders
var emailService = reportContext.getEmailService();
var email = emailService.newEmail();
email.addTo("recipient@example.com");
email.setSubject("Sales Insights Report");
email.setBody("Please find the attached Sales Insights Report.");
email.addAttachment(reportOutput.getReportData(),
"SalesInsightsReport.pdf");
emailService.send(email);


// Add a calculation for total sales
var totalSalesCalculation = myReport.createCalculation("TotalSales");
totalSalesCalculation.setExpression("SUM(SalesAmount)");
```

```
// Add a calculation for sales growth
var salesGrowthCalculation = myReport.createCalculation("SalesGrowth");
salesGrowthCalculation.setExpression("(SalesAmount - LAG(SalesAmount, 1)) / LAG(SalesAmount, 1)");

// Add conditional formatting to the chart
var formattingRule = myChart.createFormattingRule();
formattingRule.setCondition("SalesAmount > 10000");
formattingRule.setStyle("color: red; font-weight: bold;");
myChart.addFormattingRule(formattingRule);

// Create additional filters for date range and product selection
var productFilter = myReport.createFilter("ProductFilter");
productFilter.setFilterExpression("ProductName = ?");
productFilter.setPrompt("Select a Product:");
myQuery.addFilter(productFilter);

// Create a parameter for product selection
var selectedProduct = reportContext.getParameterValue("SelectedProduct");
productFilter.setValues([selectedProduct]);
```

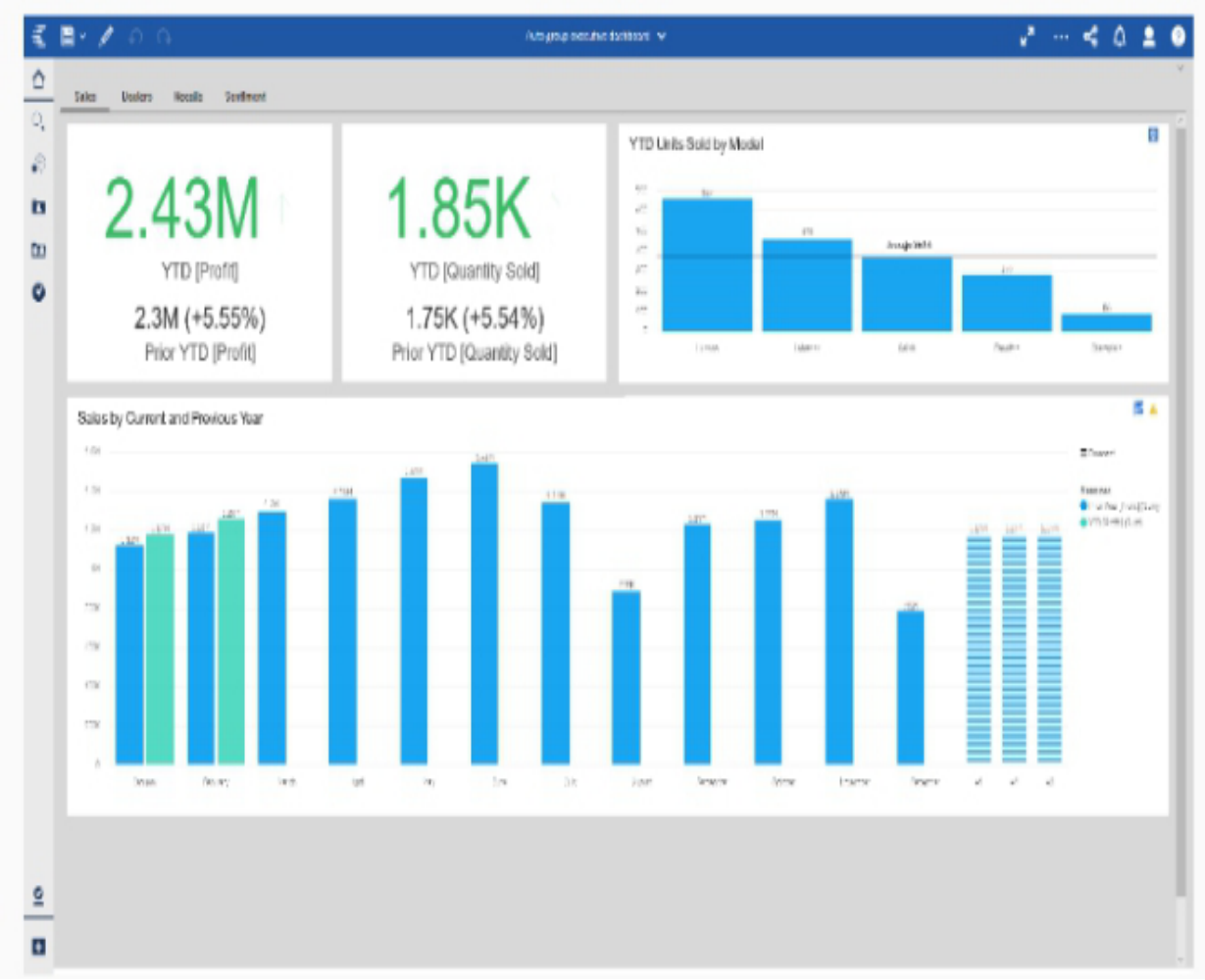
```
// Create another chart to visualize sales growth
var growthChart = myReport.createChart();
growthChart.setQuery(myQuery);
growthChart.setChartType(ChartType.LINE);
growthChart.setChartTitle("Sales Growth Trends");

// Add export options
var reportOutputFormats = [OutputFormat.PDF, OutputFormat.EXCEL,
OutputFormat.CSV];
reportRunOptions.setOutputFormats(reportOutputFormats);
```

FEATURES:

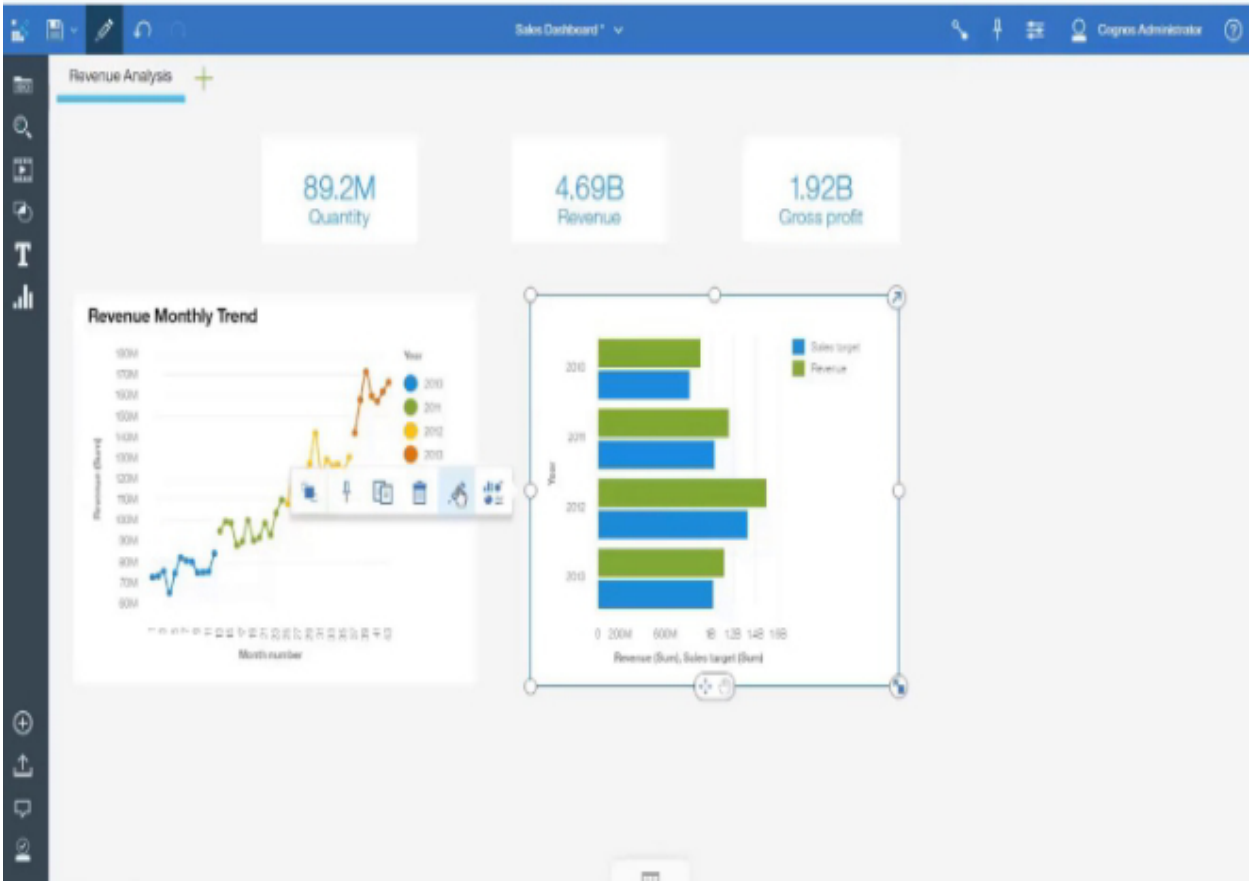
- Calculations: Calculate additional metrics, like total sales and sales growth.
- Conditional Formatting: Apply conditional formatting to highlight certain data points.
- Parameterized Filters: Allow users to filter data by multiple criteria.
- Multiple Charts: Create more than one chart for a comprehensive view.
- Export Options: Provide users with various export format options.

OUTPUT:

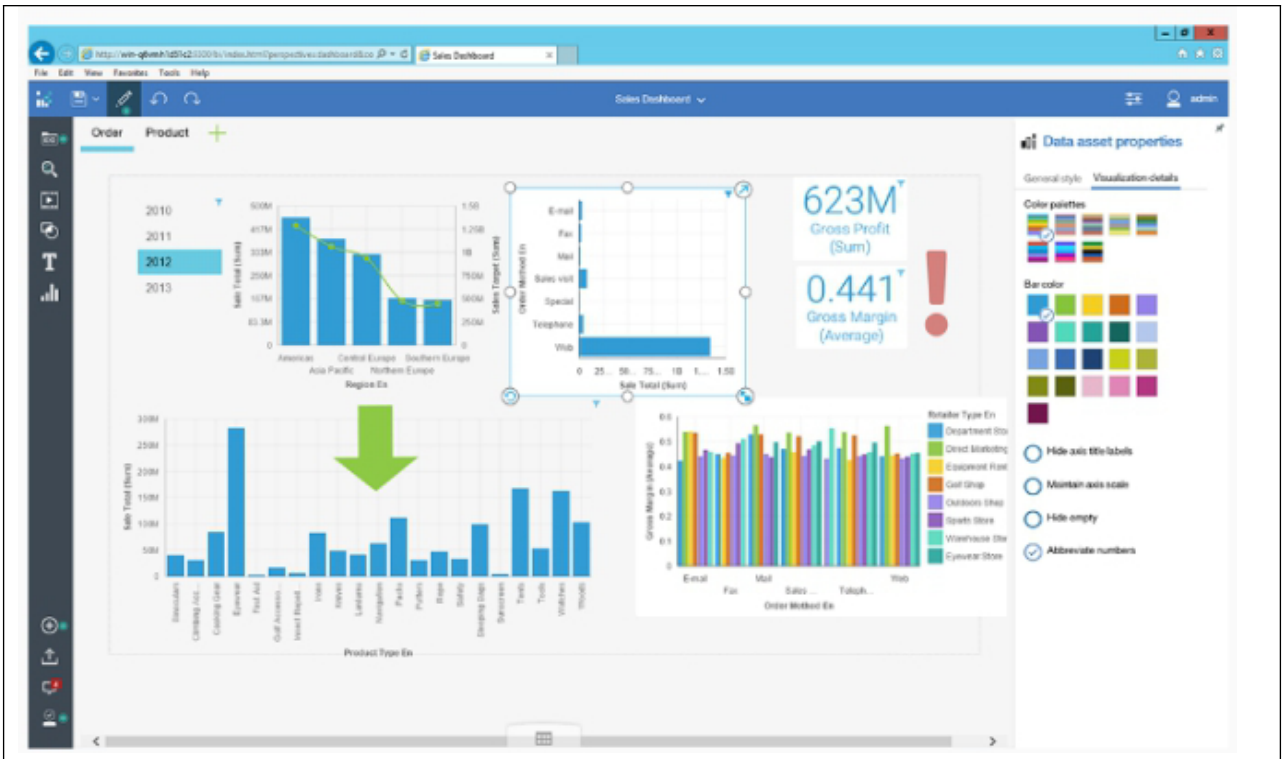
1.



2.



3.



4.

Final Conclusion and Results:

Review the report's key insights and results, such as identifying top-selling products, sales trends, and customer preferences.

Summarize actionable findings and recommendations for the business.

Share the report with stakeholders for informed decision-making.

Close IBM Cognos:

Properly close the IBM Cognos environment when you have finished working on the report.