***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 : T.K.Sabarish, Keerthi Varman.M ,Sabith Ahamed.S,***

***L.Suresh***

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

**Data Collection:**

The first step is to collect the relevant data. This data could come from various sources, including your company's internal databases, external market data, or customer surveys. The key is to gather comprehensive and clean data, including historical sales figures and any other variables that may affect sales, such as marketing spending, economic indicators, or seasonality.

**Data Preprocessing:**

Raw data often requires preprocessing to make it usable for analysis. This may involve dealing with missing data, handling outliers, and converting data types. For sales data, it might include date-time conversions and currency adjustments if you're dealing with international sales.

**Exploratory Data Analysis (EDA):**

EDA involves visualizing and summarizing the data to understand its characteristics. You might create line plots to visualize the historical sales trends, histograms to understand the distribution of sales, and scatter plots to explore relationships between sales and other variables.

**Feature Engineering:**

Depending on the data, you may need to engineer new features. For sales analysis, this might involve creating lag features to capture trends or rolling averages to smooth out noise in the data.

**Model Building:**

Machine learning models can be used to predict future sales trends. Popular models include time series forecasting models like ARIMA or machine learning models like Random Forest and XGBoost. These models take historical data and other features to make predictions. The choice of the model depends on the nature of your data.

**Model Evaluation:**

Models need to be evaluated using appropriate metrics. For sales predictions, metrics like Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE) are common. The goal is to assess how well the model performs on unseen data.

**Prediction:**

Once the model is trained and evaluated, you can use it to make future sales predictions. This provides insights into what sales might look like in the future based on the historical and contextual data used for training.

**Visualization:**

Now, we come to the pie chart. While pie charts are often used for showing the distribution of a whole, they may not be the ideal choice for sales trend analysis. Pie charts are more suited for showing parts of a whole. A line plot or bar chart might be better for visualizing sales trends over time, as they can show the changes and variations more clearly. However, if you wish to show the proportion of different product categories in your sales, a pie chart can be relevant.

**Interpretation:**

The visualization provides insights that stakeholders can use to make informed decisions. For example, if you observe that a certain product category is growing in proportion to the total sales, you might consider focusing more on marketing or production for that category.

**Reporting and Communication:**

The results and insights should be communicated effectively to stakeholders. This includes providing a narrative of the analysis process, the models used, their performance, and the implications of the sales trends on the business.

In conclusion, analyzing future sales trends using Python involves a comprehensive data analysis process, from data collection to visualization. The selection of appropriate models and metrics is crucial for accurate predictions, while data visualization tools like pie charts can enhance the communication of findings. The insights gained from this analysis can guide businesses in making informed decisions and strategies for the future

# Algorithm:

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**](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**

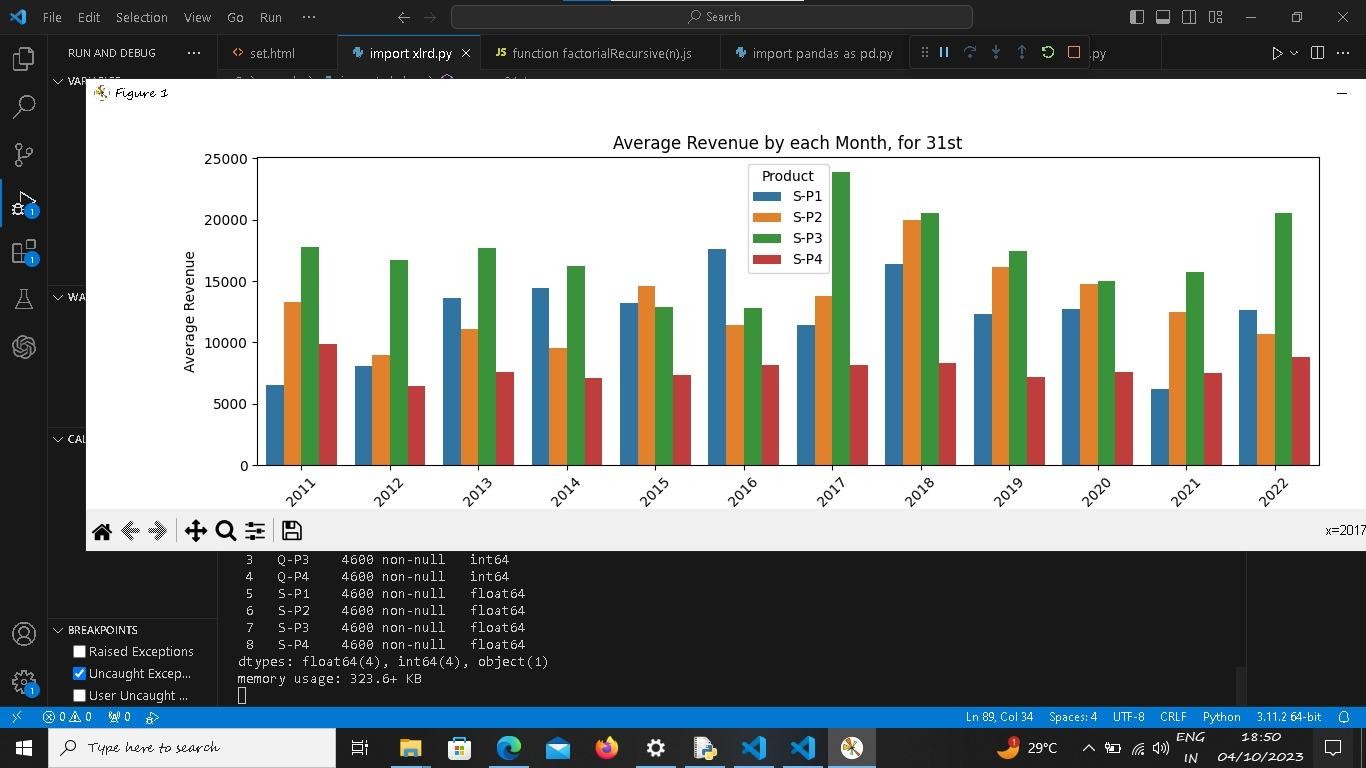
while True:

update\_pie\_chart()

plt.pause(1)

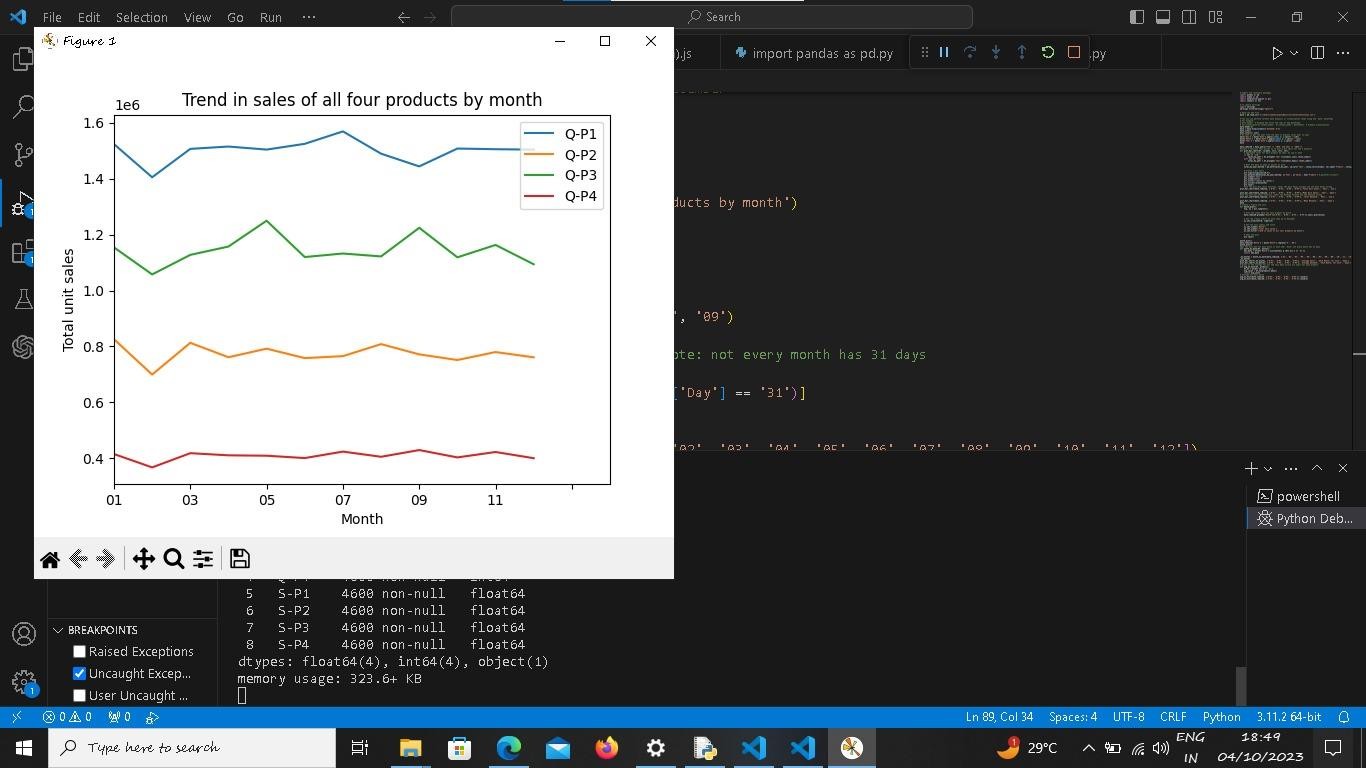
**# Update and display the pie chart every second**

output:

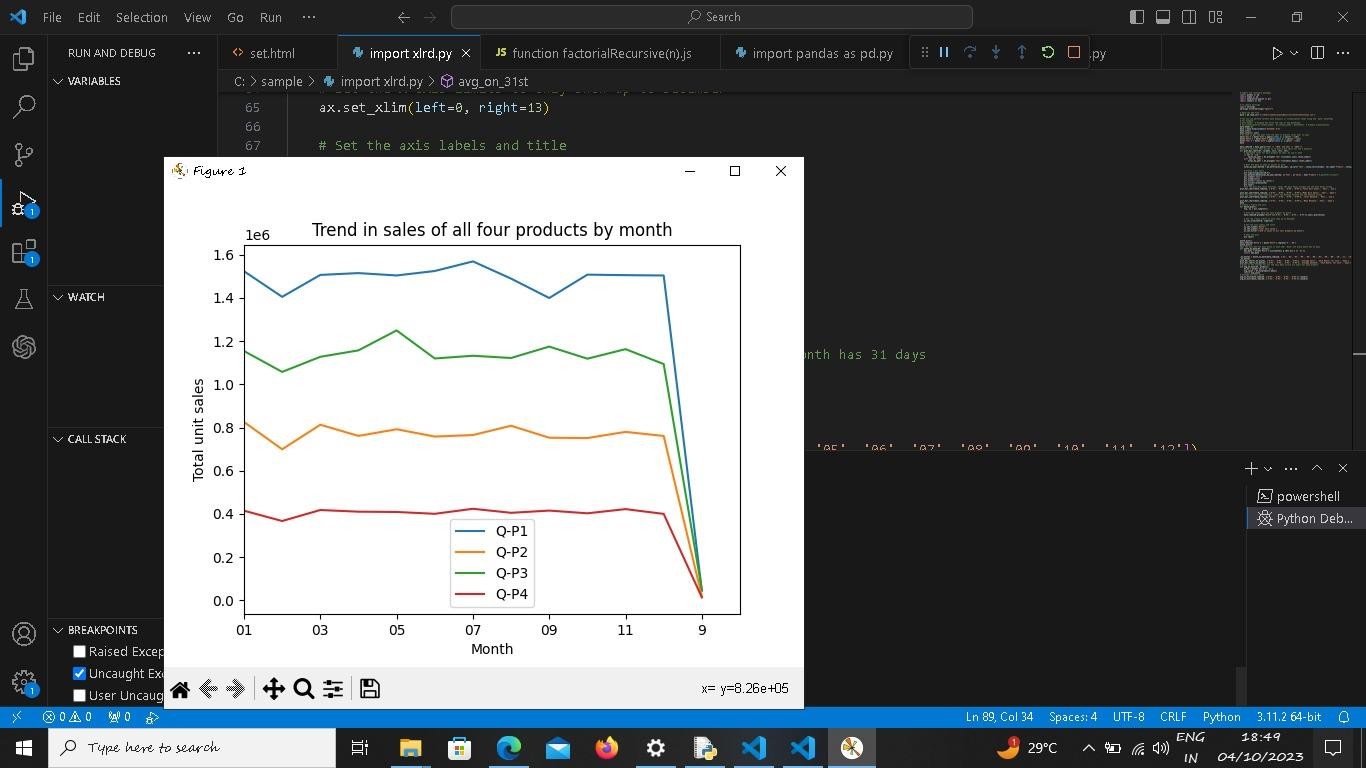
 Output 1:

Output 2:

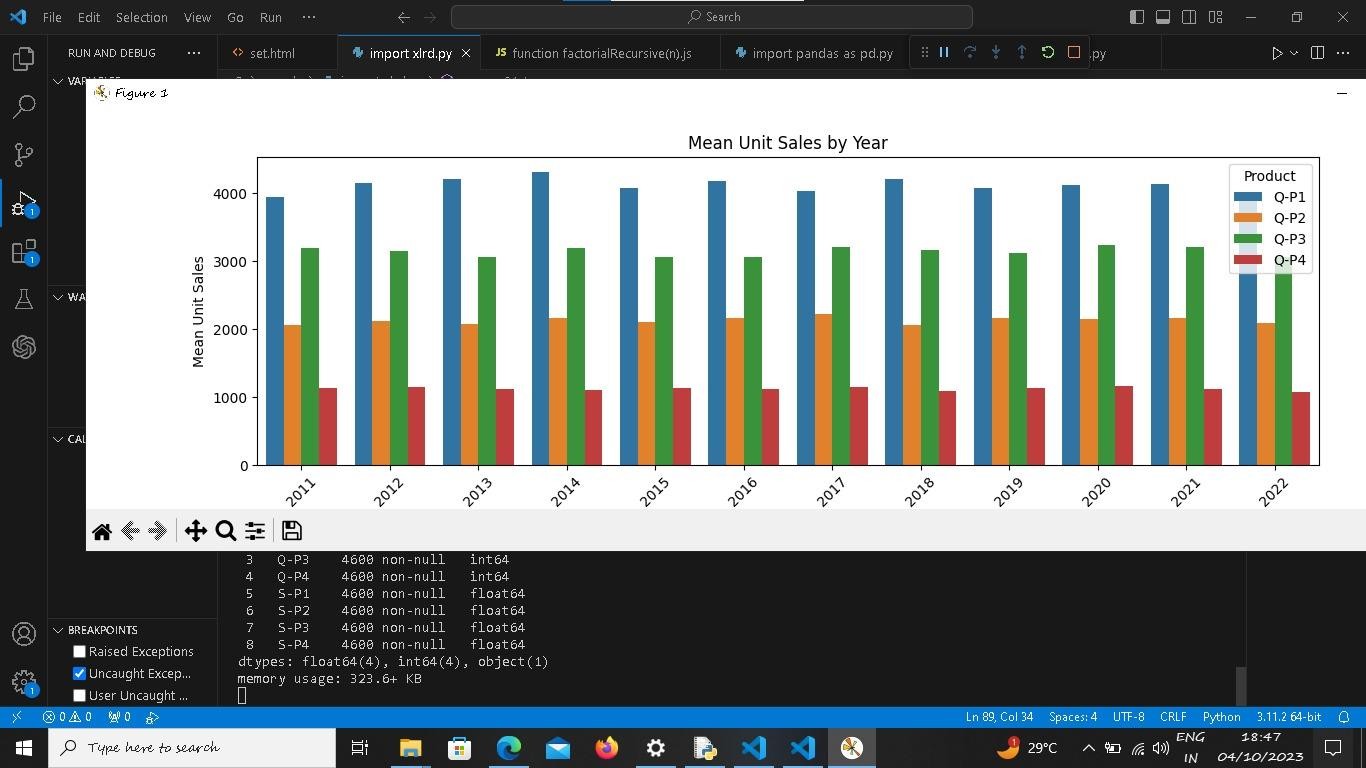


Output 3:

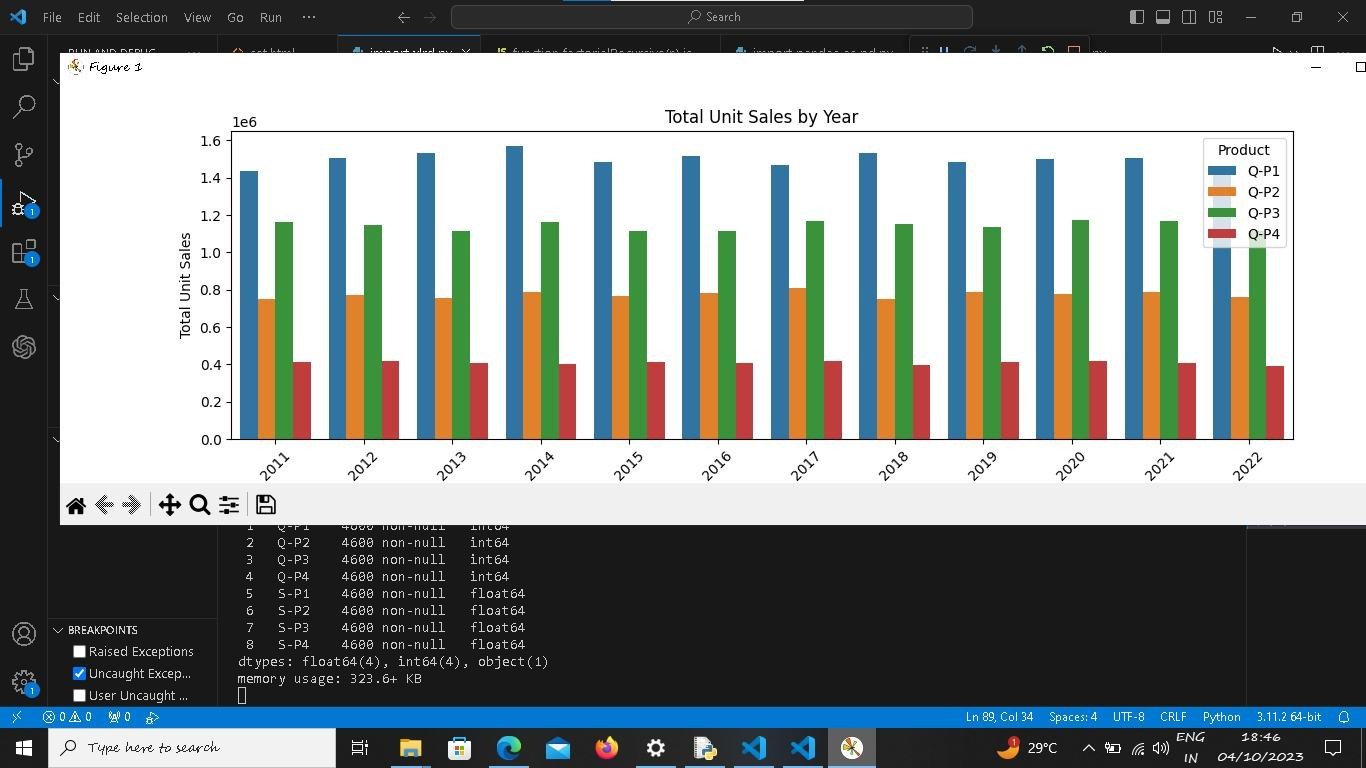
Output 4:



Output 5:



Output 6:



# CONCLUSION:

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