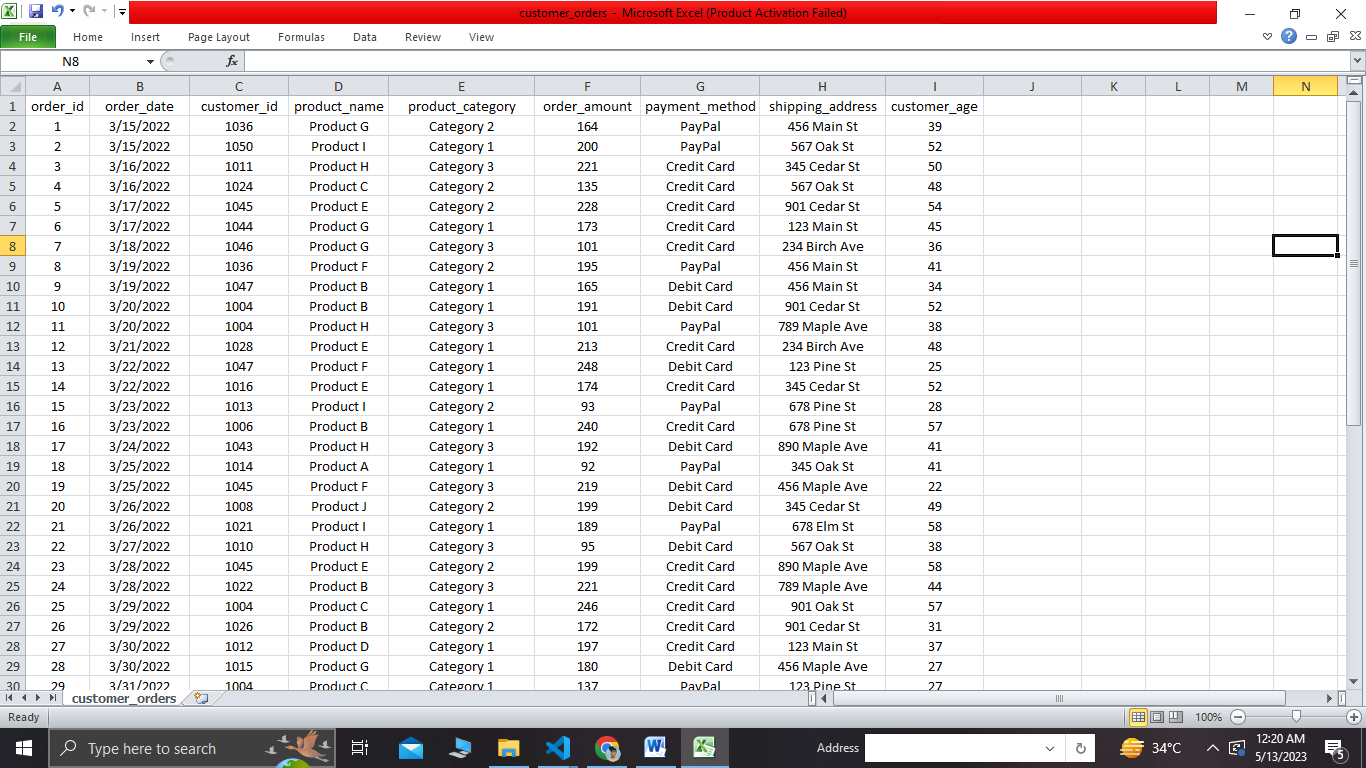
**PROJECT**

**Title : Make Simple Dashboard Using Plotly.**

**DataSet :** Customer\_order.csv

****

**Code :**

# Import required libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

import dash

from dash import Dash,html,dcc

import dash\_bootstrap\_components as dbc

# Load data into a pandas dataframe

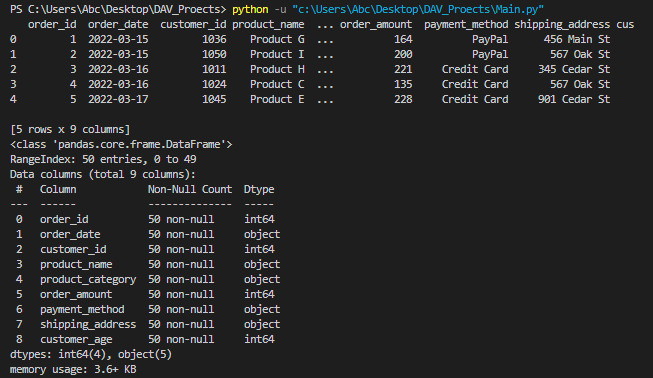
df = pd.read\_csv('customer\_orders.csv')

# Data Exploration

print(df.head()) # Print first 5 rows of data

print(df.info()) # Print column names, data types, and non-null counts

**Output :**

****

# Data Cleaning and Preprocessing

df.dropna(inplace=True) # Remove rows with missing values

df.drop\_duplicates(inplace=True) # Remove duplicate rows

df['order\_date'] = pd.to\_datetime(df['order\_date']) # Convert order\_date to datetime format

# Data Analysis

# Create a scatter plot of order amount vs. customer age

plt.figure(figsize=(8,6))

sns.scatterplot(x='customer\_age', y='order\_amount', data=df)

plt.title('Order Amount vs. Customer Age')

plt.show()

**Output :**

****

# Create a bar plot of order count by shipping address

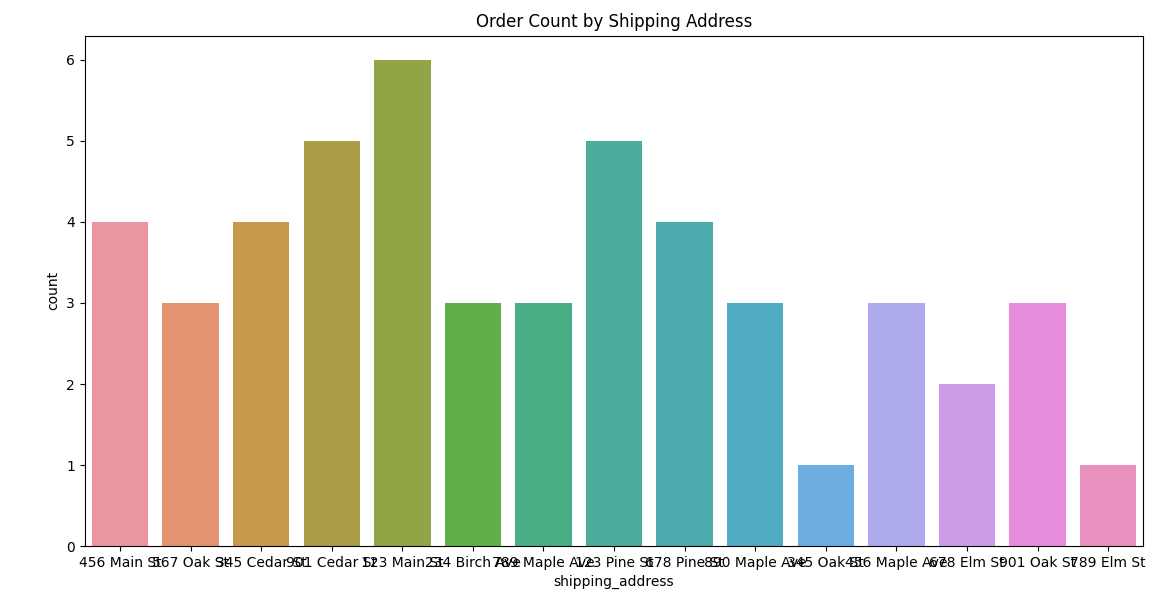
plt.figure(figsize=(8,6))

sns.countplot(x='shipping\_address', data=df)

plt.title('Order Count by Shipping Address')

plt.show()

**Output :**

****

# Create a pie chart of order amount by payment method

order\_amt\_by\_pay\_method = df.groupby('payment\_method')['order\_amount'].sum()

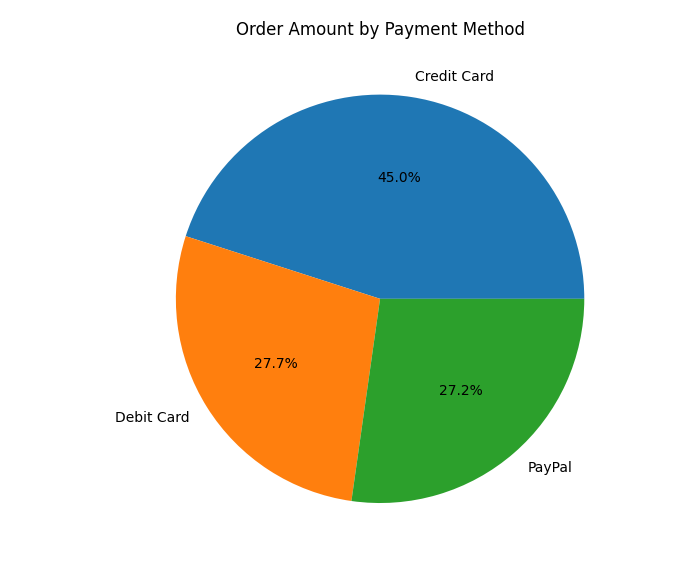
plt.figure(figsize=(8,6))

plt.pie(order\_amt\_by\_pay\_method, labels=order\_amt\_by\_pay\_method.index, autopct='%1.1f%%')

plt.title('Order Amount by Payment Method')

plt.show()

**Output :**

****

# Dashboard Implementation

# Create the app

app = dash.Dash(\_\_name\_\_, external\_stylesheets=[dbc.themes.BOOTSTRAP])

# Define the layout

app.layout = dbc.Container([

dbc.Row([ dbc.Col(html.H1('Customer Orders Dashboard'), width={'size': 12})]),

dbc.Row([dbc.Col(html.Div('Visualizing insights from customer orders data.'), width={'size': 12}) ]),

dbc.Row([

dbc.Col(dcc.Graph(

id='order-amount-vs-age',

figure=px.scatter(df, x='customer\_age', y='order\_amount', color='payment\_method')

), width={'size': 6}),

dbc.Col(dcc.Graph(

id='order-count-by-shipping-address',

figure=px.bar(df, x='shipping\_address', title='Order Count by Shipping Address')

), width={'size': 6})

]),

dbc.Row([

dbc.Col(dcc.Graph(

id='order-amount-by-payment-method',

figure=px.pie(df, values='order\_amount', names='payment\_method', title='Order Amount by Payment Method')

), width={'size': 6}),

dbc.Col(dcc.Graph(

id='order-amount-over-time',

figure=px.line(df, x='order\_date', y='order\_amount', title='Order Amount Over Time')

), width={'size': 6})

])]

, fluid=True)

# Run the app

if \_\_name\_\_ == '\_\_main\_\_':

app.run\_server(debug=True)

**Output :**

****