**PROJECT - 3**

**E-commerce Customer & Sales Analysis**

Importing all the required packages

import pandas as pd

import numpy as np

A screenshot of a computer

AI-generated content may be incorrect.

*# 1. Load both CSV files into separate Pandas DataFrames*

customers\_df = pd.read\_csv("customers.csv")

sales\_df = pd.read\_csv("sales.csv")

A screen shot of a computer

AI-generated content may be incorrect.

*#2. Display the first 5 and last 5 rows of each DataFrame*

print("Customers - First 5 rows:")

print(customers\_df.head())

print("\nCustomers - Last 5 rows:")

print(customers\_df.tail())

print("\nSales - First 5 rows:")

print(sales\_df.head())

print("\nSales - Last 5 rows:")

print(sales\_df.tail())

A screenshot of a computer

AI-generated content may be incorrect.

*#3. Show column names, data types, and check for null values*

print("Customers Info:")

print(customers\_df.info())

print("\nMissing values:\n", customers\_df.isnull().sum())

print("\nSales Info:")

print(sales\_df.info())

print("\nMissing values:\n", sales\_df.isnull().sum())

A screenshot of a computer program

AI-generated content may be incorrect.

*# 4. Convert the date columns to datetime*

customers\_df['signup\_date'] = pd.to\_datetime(customers\_df['signup\_date'])

sales\_df['order\_date'] = pd.to\_datetime(sales\_df['order\_date'])

print("Converted 'signup\_date' and 'order\_date' to datetime.")

A screen shot of a computer code

AI-generated content may be incorrect.

*#5. Calculate total revenue and create 'total\_amount'*

sales\_df['total\_amount'] = sales\_df['quantity'] \* sales\_df['price\_per\_unit']

print("Sales with total\_amount column:")

print(sales\_df.head())

A screenshot of a computer screen

AI-generated content may be incorrect.

*#6. Merge datasets on customer\_id*

merged\_df = pd.merge(sales\_df, customers\_df, *on*='customer\_id', *how*='inner')

print("Merged DataFrame:")

print(merged\_df.head())

A screenshot of a computer

AI-generated content may be incorrect.

*#7. Top 5 customers by total spending*

top5 = merged\_df.groupby(['customer\_id', 'name'])['total\_amount'].sum().sort\_values(*ascending*=False).head(5)

print("Top 5 Customers by Spending:")

print(top5)

A screenshot of a computer

AI-generated content may be incorrect.

*#8. Count of customers by country*

print("Customer Count by Country:")

customers\_df['country'].value\_counts()

A screenshot of a computer

AI-generated content may be incorrect.

*#9. Average order value per customer*

avg\_order\_value = merged\_df.groupby('customer\_id')['total\_amount'].mean()

print("Average Order Value per Customer:")

avg\_order\_value.head()

A screenshot of a computer program

AI-generated content may be incorrect.

*#10. Remove duplicates from both datasets*

customers\_df.drop\_duplicates(*inplace*=True)

sales\_df.drop\_duplicates(*inplace*=True)

print("Removed duplicates. Current shape:")

print("Customers:", customers\_df.shape)

print("Sales:", sales\_df.shape)

A screenshot of a computer program

AI-generated content may be incorrect.

*# 11. Handle missing/invalid data*

sales\_df = sales\_df[(sales\_df['quantity'] > 0) & (sales\_df['price\_per\_unit'] > 0)]

print("Sales after removing invalid rows:")

print(sales\_df.head())

A screenshot of a computer program

AI-generated content may be incorrect.

*#12. Group by category: quantity sold and revenue*

category\_summary = merged\_df.groupby('category').agg(

*total\_quantity\_sold*=('quantity', 'sum'),

*total\_revenue*=('total\_amount', 'sum')

)

print("Category Summary:")

print(category\_summary)

A screenshot of a computer screen

AI-generated content may be incorrect.

*#13. Extract year and month, analyze monthly sales*

merged\_df['order\_year\_month'] = merged\_df['order\_date'].dt.to\_period('M')

monthly\_sales = merged\_df.groupby('order\_year\_month')['total\_amount'].sum()

print("Monthly Sales:")

print(monthly\_sales)

A screenshot of a computer program

AI-generated content may be incorrect.

*#14. Customers who signed up in the last 6 months but didn’t buy*

from datetime import datetime

cutoff\_date = pd.Timestamp.now() - pd.DateOffset(*months*=6)

recent\_signups = customers\_df[customers\_df['signup\_date'] >= cutoff\_date]

purchased\_customers = merged\_df['customer\_id'].unique()

inactive\_recent = recent\_signups[~recent\_signups['customer\_id'].isin(purchased\_customers)]

inactive\_recent

A screenshot of a computer

AI-generated content may be incorrect.

*#15. Products sold less than 10 times (low performers)*

total\_sales = merged\_df.groupby('product')['quantity'].sum()

low\_performers = total\_sales[total\_sales < 10]

print("Low Performing Products (<10 sales):")

print(low\_performers)

A computer screen shot of a program code

AI-generated content may be incorrect.

*# 16. Summary report per customer*

summary = merged\_df.groupby(['customer\_id', 'name']).agg(

*total\_orders*=('order\_id', 'nunique'),

*total\_items*=('quantity', 'sum'),

*total\_spent*=('total\_amount', 'sum')

)

summary['avg\_order\_value'] = summary['total\_spent'] / summary['total\_orders']

summary

A screenshot of a computer

AI-generated content may be incorrect.

*#17. NumPy discount: Apply 10% discount to Electronics*

merged\_df['discounted\_amount'] = np.where(

    merged\_df['category'] == 'Electronics',

    merged\_df['total\_amount'] \* 0.90,

    merged\_df['total\_amount']

)

print("\nApplied discount to Electronics category:")

print(merged\_df[['product', 'category', 'total\_amount', 'discounted\_amount']].head())

A screenshot of a computer

AI-generated content may be incorrect.