**Project: Customer Purchase Analysis for a Retail Store**

Objective: Analyze customer purchase data to understand buying patterns, identify best-selling products, and provide insights into how customer segments behave, using Pandas.

**Dataset:**

Use a fictional or publicly available dataset (e.g., Kaggle's "Retail Sales Data") containing information such as:

Customer ID: Unique identifier for each customer

Product ID: Unique identifier for each product

Product Name: Name of the product

Category: Category of the product (e.g., electronics, apparel)

Purchase Date: Date when the product was purchased

Quantity: Number of units purchased

Price: Price per unit

Total Amount: Quantity \* Price

**Steps and Solution:**

Loading the Dataset:

Load the CSV file into a Pandas DataFrame.

import pandas as pd

**df = pd.read\_csv('retail\_data.csv')**

Basic Data Inspection:

Inspect the data for missing values, data types, and overall structure.

print(df.info())

print(df.head())

Data Cleaning:

Handle missing data (e.g., drop rows or fill missing values).

Remove duplicates, if any.

df.dropna(inplace=True)

df.drop\_duplicates(inplace=True)

Manipulating Data with Series and DataFrames:

Extract useful columns like Customer ID, Product Name, Total Amount into a new DataFrame.

df\_filtered = df[['Customer ID', 'Product Name', 'Total Amount']]

**Group Data and Aggregation:**

Group data by Product Name to calculate the total quantity sold and revenue generated for each product.

product\_sales = df.groupby('Product Name').agg({'Quantity': 'sum', 'Total Amount': 'sum'})

**Merge DataFrames:**

Merge customer details with the purchase data to create a unified view.

customer\_data = pd.read\_csv('customer\_details.csv')

merged\_data = pd.merge(df, customer\_data, on='Customer ID')

**Generate Summary Tables:**

Create a summary table showing the top 10 products by total revenue.

top\_products = product\_sales.sort\_values(by='Total Amount', ascending=False).head(10)

**Grouping Data into Logical Pieces:**

Group customers by their purchasing behavior (e.g., high-spending customers).

customer\_spending = df.groupby('Customer ID').agg({'Total Amount': 'sum'})

high\_spenders = customer\_spending[customer\_spending['Total Amount'] > 1000]

**Data Visualization:**

Visualize sales trends over time, using matplotlib or seaborn.

import matplotlib.pyplot as plt

df['Purchase Date'] = pd.to\_datetime(df['Purchase Date'])

df.groupby(df['Purchase Date'].dt.month)['Total Amount'].sum().plot(kind='line')

plt.show()

**Insights:**

Top Products: Identify which products generate the most revenue.

Customer Segments: Classify customers into segments (e.g., high spenders, low spenders).

Sales Trends: Analyze monthly sales trends to inform marketing and inventory decisions.