**Task 1: Exploratory Data Analysis (EDA) and Business Insights**

1. **Perform EDA:**
   * Load the provided data files.
   * Check for missing values and duplicate entries.
   * Generate summary statistics for each dataset.
   * Analyse relationships between datasets.
2. **Derive Business Insights:**
   * Insights could include:
     + Popular product categories.
     + High-value customers by region.
     + Trends in purchasing behaviour over time.
     + Correlation between product price and quantity purchased.
     + Seasonality in transactions.

**Task 2: Lookalike Model**

1. **Model Development:**
   * Define customer similarity based on features.
   * Use techniques such as cosine similarity, clustering, or machine learning models like KNN.
2. **Output:**
   * Recommend 3 similar customers for the first 20 customers (C0001 - C0020) based on their profile and transaction history.
   * Save results in a CSV file.

**Task 3: Customer Segmentation / Clustering**

1. **Clustering:**
   * Combine customer profiles with their transaction summaries.
   * Use a clustering algorithm to segment customers into meaningful groups.
   * Evaluate clustering performance using the DB Index and visualize the clusters.
2. **Output:**
   * Report the number of clusters, DB Index value, and relevant metrics.
   * Visualize the clusters using appropriate plots.

# Import necessary libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

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

products = pd.read\_csv("Products.csv")

transactions = pd.read\_csv("Transactions.csv")

print("Customers Dataset Summary:")

print(customers.info(), "\n")

print("Products Dataset Summary:")

print(products.info(), "\n")

print("Transactions Dataset Summary:")

print(transactions.info(), "\n")

print("Missing Values in Customers Dataset:")

print(customers.isnull().sum(), "\n")

print("Missing Values in Products Dataset:")

print(products.isnull().sum(), "\n")

print("Missing Values in Transactions Dataset:")

print(transactions.isnull().sum(), "\n")

merged\_data = transactions.merge(customers, on="CustomerID", how="inner").merge(products, on="ProductID", how="inner")

print("Merged Dataset:")

print(merged\_data.head(), "\n")

print("Top 5 Regions by Total Spending:")

print(merged\_data.groupby("Region")["TotalValue"].sum().sort\_values(ascending=False), "\n")

print("Top 5 Categories by Quantity Sold:")

print(merged\_data.groupby("Category")["Quantity"].sum().sort\_values(ascending=False), "\n")

print("Top 5 Products by Revenue:")

print(merged\_data.groupby("ProductName")["TotalValue"].sum().sort\_values(ascending=False).head(), "\n")

# Visualizations

# 1. Spending by Region

plt.figure(figsize=(10, 6))

sns.barplot(data=merged\_data.groupby("Region")["TotalValue"].sum().reset\_index(), x="Region", y="TotalValue", palette="viridis")

plt.title("Total Spending by Region")

plt.xlabel("Region")

plt.ylabel("Total Spending")

plt.show()

# 2. Top Product Categories by Quantity Sold

plt.figure(figsize=(10, 6))

sns.barplot(data=merged\_data.groupby("Category")["Quantity"].sum().reset\_index(), x="Category", y="Quantity", palette="coolwarm")

plt.title("Top Product Categories by Quantity Sold")

plt.xlabel("Category")

plt.ylabel("Quantity Sold")

plt.show()

# 3. Monthly Sales Trend

merged\_data["TransactionDate"] = pd.to\_datetime(merged\_data["TransactionDate"])

merged\_data["Month"] = merged\_data["TransactionDate"].dt.to\_period("M")

monthly\_sales = merged\_data.groupby("Month")["TotalValue"].sum().reset\_index()

plt.figure(figsize=(12, 6))

sns.lineplot(data=monthly\_sales, x="Month", y="TotalValue", marker="o")

plt.title("Monthly Sales Trend")

plt.xlabel("Month")

plt.ylabel("Total Sales")

plt.xticks(rotation=45)

plt.show()