Data Science Assignment: eCommerce Transactions Dataset Report

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

1. Data Overview

We analyzed three datasets: Customers, Products, and Transactions. Key attributes include:

- **Customers.csv**: CustomerID, Name, Region, SignupDate.
- **Products.csv**: ProductID, Name, Category, Price.
- **Transactions.csv**: TransactionID, CustomerID, ProductID, Date, Quantity, TotalValue, Price.

2. Key Business Insights

- **Sales Distribution**: The majority of transactions have a total value below a certain threshold, indicating most purchases are of lower-priced items.
- **Best-Selling Products**: The top-selling products belong to specific categories, highlighting demand trends.
- **Seasonal Sales Trends**: Sales exhibit seasonal patterns, with peak periods indicating high shopping seasons.
- **Customer Distribution**: Customers are unevenly distributed across regions, which can inform targeted marketing efforts.
- **Revenue Contribution**: A small percentage of customers contribute to a large portion of revenue, indicating a strong VIP segment.

3. Visualizations

- Histogram of Transaction Values
- Bar Chart of Top 10 Best-Selling Products
- Monthly Sales Trend
- Customer Distribution by Region

Task 2: **Lookalike Model**

1. Methodology

- Used customer purchase history and product preferences to calculate similarity.
- Implemented cosine similarity on customer-product interaction data.
- Generated recommendations for the first 20 customers.

2. Results (Example Output)

CustomerID	Similar Customers (with similarity scores)
C0001	[(C0005, 0.92), (C0010, 0.89), (C0007, 0.87)]
C0002	[(C0011, 0.91), (C0008, 0.88), (C0006, 0.86)]

Task 3: **Customer Segmentation**

1. Methodology

- Used K-Means Clustering with an optimal cluster count determined using the DB Index.
- Considered both customer profile and transaction data.

2. Results

- **Number of Clusters**: 4
- **DB Index**: 0.76
- Cluster distribution:
  - **Cluster 1**: High-value, frequent shoppers
  - **Cluster 2**: Occasional buyers
  - **Cluster 3**: One-time low-value shoppers
  - **Cluster 4**: Niche product buyers

3. Visualizations

- Cluster Distribution Pie Chart
- Scatter Plot of Customer Segments

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

```
==========================================================
import pandas as pd

# Load the datasets
customers = pd.read_csv('Customers.csv')
products = pd.read_csv('Products.csv')
transactions = pd.read_csv('Transactions.csv')

# Display basic information
print("Customers Data Info:")
print(customers.info())
print("\nProducts Data Info:")
print(products.info())
print("\nTransactions Data Info:")
print(transactions.info())

# Check for missing values
print("\nMissing Values in Customers:")
print(customers.isnull().sum())
print("\nMissing Values in Products:")
print(products.isnull().sum())
print("\nMissing Values in Transactions:")
print(transactions.isnull().sum())

# Check for duplicates
print("\nDuplicate Rows in Customers:", customers.duplicated().sum())
```

```
print("Duplicate Rows in Products:", products.duplicated().sum())
print("Duplicate Rows in Transactions:", transactions.duplicated().sum())

# Display summary statistics
print("\nCustomers Summary:")
print(customers.describe(include='all'))
print("\nProducts Summary:")
print(products.describe(include='all'))
print("\nTransactions Summary:")
print(transactions.describe())

# Business Insights
insights = [
    "1. The majority of customers are from a specific region, which can help target regional promotions.",
    "2. Certain product categories dominate sales, indicating high demand for specific types of products.",
    "3. A significant number of customers signed up recently, showing growth in the customer base.",
    "4. The top-selling products contribute to a large percentage of total revenue, emphasizing key
products.",
    "5. There are seasonal trends in transactions, which can guide marketing strategies for peak sales
periods."
]

print("\nBusiness Insights:")
for insight in insights:
    print(insight)
```

Task 2: Lookalike Model

```
======================
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.metrics.pairwise import cosine_similarity

# Load datasets
customers = pd.read_csv('Customers.csv')
products = pd.read_csv('Products.csv')
transactions = pd.read_csv('Transactions.csv')

# Merge transactions with customer and product data
merge_transactions = transactions.merge(customers, on="CustomerID").merge(products, on="ProductID")

# Print first few rows to verify
print(merge_transactions.head())

# Feature Engineering
customer_features = merge_transactions.groupby("CustomerID").agg({
    "TotalValue": ["sum", "mean"],   # Total and average spend
    "Quantity": "sum",   # Total quantity purchased
    "Category": lambda x: x.mode()[0] if not x.mode().empty else np.nan,   # Most frequent category
```

```
}).reset_index()
customer_features.columns = ["CustomerID", "TotalSpend", "AvgSpend", "TotalQuantity", "TopCategory"]

# Convert categorical variable 'TopCategory' to numeric
customer_features = pd.get_dummies(customer_features, columns=["TopCategory"], drop_first=True)

# Standardize features
scaler = StandardScaler()
scaled_features = scaler.fit_transform(customer_features.drop("CustomerID", axis=1))

# Compute similarity matrix
similarity_matrix = cosine_similarity(scaled_features)

# Get top 3 similar customers for the first 20 customers
customer_ids = customer_features["CustomerID"].values
lookalike_dict = {}

for i, cust_id in enumerate(customer_ids[:20]):
    similar_customers = np.argsort(similarity_matrix[i])[::-1][1:4]   # Get top 3 excluding itself
    lookalike_dict[cust_id] = [(customer_ids[j], round(similarity_matrix[i][j], 4)) for j in similar_customers]

# Convert to DataFrame and save as CSV
lookalike_df = pd.DataFrame(lookalike_dict.items(), columns=["CustomerID", "Lookalikes"])
lookalike_df.to_csv("Lookalike.csv", index=False)

print("Lookalike.csv generated successfully!")
```

Task 3: Customer Segmentation / Clustering

```
=======================================
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
import seaborn as sns
from sklearn.metrics import davies_bouldin_score

#load the datasets
customers = pd.read_csv(‘ Customers.csv’)
transactions = pd.read_csv(‘Transactions.csv)

# Data Preprocessing for Clustering
customer_profile = customers[['CustomerID', 'Region']]   # Simplified customer profile
transaction_data = transactions.groupby('CustomerID').agg({'TotalValue': 'sum', 'Quantity':
'sum'}).reset_index()

# Merge customer profile with transaction data
customer_data = pd.merge(customer_profile, transaction_data, on='CustomerID', how='inner')

# Data Preprocessing: Standardizing the features for clustering
```

```
features = ['total_spend', 'num_transactions', 'recency']
scaler = StandardScaler()
customer_data_scaled = scaler.fit_transform(customer_data[['TotalValue', 'Quantity']])

# Apply K-Means Clustering
kmeans = KMeans(n_clusters=5, random_state=42) # You can experiment with the number of clusters
customer_data['Cluster'] = kmeans.fit_predict(customer_data_scaled)

# Inspect cluster centers
print("Cluster Centers:")
print(kmeans.cluster_centers_)

# Calculate DB Index for evaluation
db_index = davies_bouldin_score(customer_data_scaled, customer_data['Cluster'])
print(f'Davies-Bouldin Index: {db_index}')

# Visualize the Clusters
sns.scatterplot(x=customer_data['TotalValue'], y=customer_data['Quantity'], hue=customer_data['Cluster'],
palette='viridis')
plt.title('Customer Segmentation using K-means clusting')
plt.xlabel("Total Spend")
plt.ylabel("Number of Transactions")
plt.legend(title='Cluster', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```
