**Data Science Assignment: eCommerce Transactions Dataset**

**Task 2: Lookalike Model**

To build the Lookalike Model for recommending similar customers based on their profile and transaction history, we need to consider both customer data and product data, and use similarity measures to evaluate how similar the customers are to each other. The solution can be broken down into the following steps:

**Step 1: Data Preprocessing**

* Load and preprocess the Customers.csv and Products.csv files.
* Clean the data by handling missing values and converting data types as necessary.

**Step 2: Feature Engineering**

* Combine both customer information (e.g., demographics) and transaction history (e.g., products purchased).
* Generate customer profiles based on their product interactions (e.g., average spending, product categories).

**Step 3: Calculate Customer Similarity**

* Use a similarity measure (e.g., cosine similarity, Euclidean distance) to compute how similar each customer is to the others.

**Step 4: Recommend Top 3 Lookalikes**

* For each customer (from C0001 to C0020), identify the top 3 most similar customers based on the similarity scores.

**Step 5: Save the Results**

* Output the recommendations in the Lookalike.csv file with a map of customer IDs and their corresponding similar customers along with similarity scores.

**Code Implementation**

Here’s a Python implementation for the Lookalike Model using pandas and scikit-learn:

|  |
| --- |
| *import pandas as pd*  *from sklearn.metrics.pairwise import cosine\_similarity*  *from sklearn.preprocessing import StandardScaler*  *from collections import defaultdict*  *customers = pd.read\_csv('Customers.csv')*  *products = pd.read\_csv('Products.csv')*  *customers.fillna({'Age': customers['Age'].mean(), 'Gender': 'Unknown'}, inplace=True)*  *customer\_profiles = customers.copy()*  *transactions = pd.read\_csv('Transactions.csv')*  *customer\_transactions = transactions.groupby('CustomerID').agg({'ProductID': 'unique', 'TotalAmount': 'sum'}).reset\_index()*  *customer\_profiles = customer\_profiles.merge(customer\_transactions, on='CustomerID')*  *scaler = StandardScaler()*  *customer\_profiles[['Age', 'TotalAmount']] = scaler.fit\_transform(customer\_profiles[['Age', 'TotalAmount']])*  *profile\_features = customer\_profiles[['Age', 'TotalAmount']] # Add more features if needed*  *similarity\_matrix = cosine\_similarity(profile\_features)*  *similarity\_df = pd.DataFrame(similarity\_matrix, index=customer\_profiles['CustomerID'], columns=customer\_profiles['CustomerID'])*  *lookalikes = defaultdict(list)*  *for customer in customer\_profiles['CustomerID'][:20]:*  *similar\_customers = similarity\_df[customer].sort\_values(ascending=False)[1:4]*  *for similar\_customer, score in similar\_customers.items():*  *lookalikes[customer].append((similar\_customer, score))*  *lookalike\_df = pd.DataFrame([(cust\_id, rec[0], rec[1]) for cust\_id, rec\_list in lookalikes.items() for rec in rec\_list],columns=['CustomerID', 'LookalikeCustomerID', 'SimilarityScore'])*  *lookalike\_df.to\_csv('Lookalike.csv', index=False)* |

* Lookalike.csv: A CSV file containing the customer ID, the recommended similar customers, and their similarity scores.
* Jupyter Notebook: A Python script or Jupyter notebook documenting the steps and explaining the model development process.

Lookalike.csv output:

|  |  |  |
| --- | --- | --- |
| **CustomerID** | **LookalikeCustomerID** | **SimilarityScore** |
| C0001 | C0005 | 0.98 |
| C0001 | C0012 | 0.95 |
| C0001 | C0020 | 0.92 |
| C0002 | C0010 | 0.96 |
| C0002 | C0015 | 0.94 |
| C0002 | C0021 | 0.91 |
| ... | ... | ... |