Name: Kalhar Patel

Mail Id: kalharpatel9@gmail.com

Documentation for Task 2: Lookalike Model

1. Objective:

The purpose of this task was to develop a Lookalike Model that recommends the top 3 customers similar to a given customer. The recommendations are based on customer profiles and transaction histories, combining features like purchasing behavior and product category preferences. The model aims to aid targeted marketing and personalized recommendations.

2. Dataset Overview:

Three datasets were utilized:

1. Customers.csv: Contains customer details like CustomerID, CustomerName, Region, and SignupDate.
2. Products.csv: Provides product details, including ProductID, ProductName, Category, and Price.
3. Transactions.csv: Includes transactional details, such as TransactionID, CustomerID, ProductID, TransactionDate, Quantity, and TotalValue.

3. Methodology:

A. Data Loading and Preprocessing

The three datasets were loaded into pandas DataFrames. Product and transaction data were joined to associate products with their categories for further analysis.

B. Feature Engineering

Customer Purchase Behavior: Aggregated transactional data for each customer to calculate metrics like total transactions, spending, and quantities purchased.
Product Category Preferences: Created a crosstab showing the count of purchases in each category for each customer.

C. Feature Integration

Combined all the engineered features into a single DataFrame representing customer profiles. Scaled the features using the StandardScaler to ensure uniformity across different feature ranges.

D. Similarity Calculation

Used the cosine_similarity function from scikit-learn to compute pairwise similarities between customer profiles.

E. Generating Recommendations

For each customer in the first 20 entries, extracted the top 3 most similar customers and stored their CustomerIDs and similarity scores.

Name: Kalhar Patel

Mail Id: kalharpatel9@gmail.com

4. Code Highlights

Data Aggregation

```
customer_features = transactions_df.groupby('CustomerID').agg({
'TransactionID': 'count',
'TotalValue': ['sum', 'mean'],
'Quantity': ['sum', 'mean']
}).reset_index()
```

Category Preferences

```
category_pivot = pd.crosstab(
category_data['CustomerID'],
category_data['Category']
)
```

Similarity Calculation

```
similarity_matrix = cosine_similarity(scaled_features)
```

Top-3 Similar Customers

```
def get_top_3_similar(customer_idx, similarity_scores):
    similar_indices = np.argsort(similarity_scores)[-4:][:-1][::-1]
    return [
        (customer_profiles.iloc[idx]['CustomerID'], similarity_scores[idx])
        for idx in similar_indices
    ]
```

5. Results

The generated 'Lookalike.csv' contains the top 3 similar customers for each of the first 20 customers, along with their similarity scores. For example:
CustomerID: C0001
- SimilarCustomerID: C0069 (SimilarityScore: 0.924680)
- SimilarCustomerID: C0035 (SimilarityScore: 0.799590)
- SimilarCustomerID: C0127 (SimilarityScore: 0.788884)

6. Insights and Usage:

This model provides actionable insights for targeted marketing by identifying customers with similar purchasing behaviors and preferences. Businesses can leverage these recommendations to:
- Promote products to customers with similar interests.
- Design personalized offers for better customer retention.

The implementation's scalability allows extending the recommendations beyond the initial 20 customers, enhancing its utility for large-scale customer bases.