

✓ TASK - 2 LOOK A LIKE MODEL

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

```
#Load Datasets
Customers = pd.read_csv('/content/Customers.csv')
Products = pd.read_csv('/content/Products.csv')
Transactions = pd.read_csv('/content/Transactions.csv')
```

```
# Parse date columns
Customers['SignupDate'] = pd.to_datetime(Customers['SignupDate'])
Transactions['TransactionDate'] = pd.to_datetime(Transactions['TransactionDate'])
```

```
# Merge datasets
data = pd.merge(Transactions, Customers, on='CustomerID')
data = pd.merge(data, Products, on='ProductID')
```

```
# Aggregate transaction data for customers
# Merge 'Price' column from Products DataFrame to data DataFrame
data = pd.merge(data, Products[['ProductID', 'Price']], on='ProductID', how='left')

Customer_features = data.groupby('CustomerID').agg({
    'TotalValue': 'sum',
    'Quantity': 'sum',
    'Price': 'mean' # Now 'Price' column is available
}).reset_index()
```

```
# Merge with customer profile
Customer_profiles = pd.merge(Customers, Customer_features, on='CustomerID')
```

```
# Encode categorical data (e.g., Region)
Customer_profiles = pd.get_dummies(Customer_profiles, columns=['Region'], drop_first=True)
```

```
# Standardize features, excluding the 'SignupDate' column
features_to_scale = Customer_profiles.select_dtypes(include=np.number) # Select only numeric columns

# Check if 'CustomerID' is in the columns before dropping it
if 'CustomerID' in features_to_scale.columns:
    features_to_scale = features_to_scale.drop(columns=['CustomerID']) # Exclude 'CustomerID' if it exists
```

```
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features_to_scale)
```

```
# Create a new DataFrame with scaled features and original 'CustomerID' and 'SignupDate'
Customer_profiles_scaled = pd.DataFrame(features_scaled, columns=features_to_scale.columns, index=Customer_profiles.index)
Customer_profiles_scaled[['CustomerID', 'SignupDate']] = Customer_profiles[['CustomerID', 'SignupDate']]
```

```
# Compute similarity matrix
similarity_matrix = cosine_similarity(features_scaled)
```

```
# Map each customer ID to their top 3 similar customers
lookalike_dict = {}
Customer_ids = Customer_profiles['CustomerID']

for i, customer_id in enumerate(Customer_ids[:20]): # First 20 customers
    # Get similarity scores for the current customer
    similarity_scores = list(enumerate(similarity_matrix[i]))
    # Sort by similarity score in descending order (excluding the customer itself)
    similarity_scores = sorted(similarity_scores, key=lambda x: x[1], reverse=True)[1:4]
    # Map customer ID to the top 3 similar customers
    lookalike_dict[customer_id] = [(Customer_ids[j], round(score, 2)) for j, score in similarity_scores]
```

```
# Convert lookalike dictionary to DataFrame and save to CSV
lookalike_df = pd.DataFrame([
    {'CustomerID': cust_id, 'Lookalikes': str(lookalikes)}
    for cust_id, lookalikes in lookalike_dict.items()
])

lookalike_df.to_csv('Lookalike.csv', index=False)

# Display the generated lookalike file for verification
print(lookalike_df)
```

```
↩ CustomerID Lookalikes
0      C0001 [('C0103', 1.0), ('C0092', 1.0), ('C0135', 0.99)]
1      C0002 [('C0029', 1.0), ('C0077', 1.0), ('C0157', 1.0)]
2      C0003 [('C0111', 1.0), ('C0190', 1.0), ('C0038', 0.99)]
3      C0004 [('C0165', 1.0), ('C0162', 1.0), ('C0075', 1.0)]
4      C0005 [('C0167', 1.0), ('C0020', 1.0), ('C0128', 1.0)]
5      C0006 [('C0168', 1.0), ('C0196', 1.0), ('C0187', 0.99)]
6      C0007 [('C0125', 1.0), ('C0089', 1.0), ('C0085', 1.0)]
7      C0008 [('C0084', 1.0), ('C0113', 1.0), ('C0017', 0.99)]
8      C0009 [('C0130', 1.0), ('C0128', 1.0), ('C0192', 1.0)]
9      C0010 [('C0176', 1.0), ('C0055', 0.99), ('C0174', 0.99)]
10     C0011 [('C0023', 1.0), ('C0139', 0.99), ('C0100', 0.99)]
11     C0012 [('C0101', 1.0), ('C0093', 1.0), ('C0153', 1.0)]
12     C0013 [('C0021', 1.0), ('C0141', 1.0), ('C0059', 1.0)]
13     C0014 [('C0097', 1.0), ('C0043', 1.0), ('C0032', 1.0)]
14     C0015 [('C0058', 1.0), ('C0186', 0.99), ('C0131', 0.99)]
15     C0016 [('C0040', 1.0), ('C0107', 1.0), ('C0066', 1.0)]
16     C0017 [('C0113', 1.0), ('C0084', 0.99), ('C0008', 0.99)]
17     C0018 [('C0041', 0.99), ('C0068', 0.99), ('C0004', 0.99)]
18     C0019 [('C0166', 1.0), ('C0031', 0.99), ('C0088', 0.99)]
19     C0020 [('C0005', 1.0), ('C0128', 1.0), ('C0167', 1.0)]
```