


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

```
from google.colab import files
uploaded = files.upload()
```

 No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
 Saving Transactions.csv to Transactions.csv
 Saving Products.csv to Products.csv
 Saving Customers.csv to Customers.csv

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

```
# Merge datasets
data = transactions.merge(customers, on="CustomerID", how="left")
data = data.merge(products, on="ProductID", how="left")
```

```
# Feature Engineering: Aggregate features for each customer
customer_features = data.groupby("CustomerID").agg(
    total_purchases=("Quantity", "sum"),
    total_spent=("TotalValue", "sum"),
    avg_transaction_value=("TotalValue", "mean"),
    preferred_category=("Category", lambda x: x.mode()[0]),
    region=("Region", "first")
).reset_index()
```

```
# Scale numerical features
scaler = StandardScaler()
numeric_features = ["total_purchases", "total_spent", "avg_transaction_value"]
customer_features[numeric_features] = scaler.fit_transform(customer_features[numeric_features])
```

```
# Compute pairwise similarity
customer_matrix = customer_features.drop(["CustomerID", "preferred_category", "region"], axis=1) # Drop categorical columns
similarity_matrix = cosine_similarity(customer_matrix)
```


```
# Create a DataFrame for similarity scores
similarity_df = pd.DataFrame(similarity_matrix, index=customer_features["CustomerID"], columns=customer_features["CustomerID"])
```

```
# Generate recommendations for customers C0001 to C0020
recommendations = {}
for customer_id in customer_features["CustomerID"][:20]: # First 20 customers
    similar_customers = similarity_df[customer_id].sort_values(ascending=False)[1:4] # Top 3 similar customers
    recommendations[customer_id] = list(zip(similar_customers.index, similar_customers.values))
```

```
# Convert recommendations to the desired output format
lookalike_data = []
for cust_id, similar_list in recommendations.items():
    lookalike_data.append({
        "cust_id": cust_id,
        "lookalikes": similar_list
    })
```

```
# Save to Lookalike.csv
lookalike_df = pd.DataFrame({
    "cust_id": [entry["cust_id"] for entry in lookalike_data],
    "lookalikes": [entry["lookalikes"] for entry in lookalike_data]
})
lookalike_df.to_csv("Lookalike.csv", index=False)
```

```
print("Lookalike recommendations saved to Lookalike.csv")
```

 Lookalike recommendations saved to Lookalike.csv