

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

```
#Loading the datasets
customers = pd.read_csv('/content/sample_data/Customers.csv')
products = pd.read_csv('/content/sample_data/Products.csv')
transactions = pd.read_csv('/content/sample_data/Transactions.csv')
```

```
#Merging data
transactions = transactions.merge(customers, on='CustomerID', how='left')
transactions = transactions.merge(products, on='ProductID', how='left')
```

```
#Check the columns of the merged DataFrame
print(transactions.columns)
```

```
Index(['TransactionID', 'CustomerID', 'ProductID', 'TransactionDate',
      'Quantity', 'TotalValue', 'Price_x', 'CustomerName', 'Region',
      'SignupDate', 'ProductName', 'Category', 'Price_y'],
      dtype='object')
```

✓ Performing Feature Engineering

```
# Calculate TotalValue using the price_x column
transactions['TotalValue'] = transactions['Quantity'] * transactions['Price_x']

#Creating customer features
customer_features = transactions.groupby('CustomerID').agg({
    'TotalValue': 'sum',
    'TransactionID': 'count',
}).reset_index()

customer_features.columns = ['CustomerID', 'TotalSpent', 'NumTransactions']

#Adding demographic features
customer_features = customer_features.merge(customers[['CustomerID', 'Region']], on='CustomerID', how='left')

#One-hot encode categorical features
customer_features = pd.get_dummies(customer_features, columns=['Region'], drop_first=True)

#Standardize the features
scaler = StandardScaler()
features_scaled = scaler.fit_transform(customer_features.drop('CustomerID', axis=1))

#Calculate similarity scores
similarity_matrix = cosine_similarity(features_scaled)

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

✓ Lookalike Model

Start coding or [generate](#) with AI.

