```python
import pandas as pd

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

# Preview datasets
print(customers.head())
print(products.head())
print(transactions.head())
```

```
  CustomerID        CustomerName         Region  SignupDate
0      C0001    Lawrence Carroll  South America  2022-07-10
1      C0002      Elizabeth Lutz           Asia  2022-02-13
2      C0003      Michael Rivera  South America  2024-03-07
3      C0004  Kathleen Rodriguez  South America  2022-10-09
4      C0005         Laura Weber           Asia  2022-08-15
  ProductID              ProductName     Category   Price
0      P001     ActiveWear Biography        Books  169.30
1      P002    ActiveWear Smartwatch  Electronics  346.30
2      P003  ComfortLiving Biography        Books   44.12
3      P004            BookWorld Rug   Home Decor   95.69
4      P005           TechPro T-Shirt     Clothing  429.31
  TransactionID CustomerID ProductID      TransactionDate  Quantity  \
0        T00001      C0199      P067  2024-08-25 12:38:23         1
1        T00112      C0146      P067  2024-05-27 22:23:54         1
2        T00166      C0127      P067  2024-04-25 07:38:55         1
3        T00272      C0087      P067  2024-03-26 22:55:37         2
4        T00363      C0070      P067  2024-03-21 15:10:10         3

   TotalValue   Price
0      300.68  300.68
1      300.68  300.68
2      300.68  300.68
3      601.36  300.68
4      902.04  300.68
```

```python
# Merge transactions with products
transactions = transactions.merge(products, on='ProductID')

# Merge transactions with customers
data = transactions.merge(customers, on='CustomerID')

print(data.head())
```

```
  TransactionID CustomerID ProductID      TransactionDate  Quantity  \
0        T00001      C0199      P067  2024-08-25 12:38:23         1
1        T00112      C0146      P067  2024-05-27 22:23:54         1
2        T00166      C0127      P067  2024-04-25 07:38:55         1
3        T00272      C0087      P067  2024-03-26 22:55:37         2
4        T00363      C0070      P067  2024-03-21 15:10:10         3

   TotalValue  Price_x                      ProductName     Category  Price_y  \
0      300.68   300.68  ComfortLiving Bluetooth Speaker  Electronics   300.68
1      300.68   300.68  ComfortLiving Bluetooth Speaker  Electronics   300.68
2      300.68   300.68  ComfortLiving Bluetooth Speaker  Electronics   300.68
3      601.36   300.68  ComfortLiving Bluetooth Speaker  Electronics   300.68
4      902.04   300.68  ComfortLiving Bluetooth Speaker  Electronics   300.68

      CustomerName         Region  SignupDate
0   Andrea Jenkins         Europe  2022-12-03
1  Brittany Harvey           Asia  2024-09-04
2  Kathryn Stevens         Europe  2024-04-04
3  Travis Campbell  South America  2024-04-11
4    Timothy Perez         Europe  2022-03-15
```

```python
# Calculate total spending per customer
customer_spending = data.groupby('CustomerID')['TotalValue'].sum().reset_index()
customer_spending.rename(columns={'TotalValue': 'TotalSpending'}, inplace=True)
```

```
# Calculate average transaction value per customer
customer_avg_transaction = data.groupby('CustomerID')['TotalValue'].mean().reset_index()
customer_avg_transaction.rename(columns={'TotalValue': 'AvgTransactionValue'}, inplace=True)

# Count product category preferences
category_preferences = pd.crosstab(data['CustomerID'], data['Category'])

# Combine all features into a single DataFrame
customer_features = customers.merge(customer_spending, on='CustomerID', how='left')
customer_features = customer_features.merge(customer_avg_transaction, on='CustomerID', how='left')
customer_features = customer_features.merge(category_preferences, on='CustomerID', how='left')

# Fill missing values with 0
customer_features.fillna(0, inplace=True)

print(customer_features.head())
```

```
  CustomerID        CustomerName         Region  SignupDate  TotalSpending  \
0      C0001    Lawrence Carroll  South America  2022-07-10        3354.52
1      C0002      Elizabeth Lutz           Asia  2022-02-13        1862.74
2      C0003      Michael Rivera  South America  2024-03-07        2725.38
3      C0004  Kathleen Rodriguez  South America  2022-10-09        5354.88
4      C0005         Laura Weber           Asia  2022-08-15        2034.24

   AvgTransactionValue  Books  Clothing  Electronics  Home Decor
0              670.904    1.0       0.0          3.0         1.0
1              465.685    0.0       2.0          0.0         2.0
2              681.345    0.0       1.0          1.0         2.0
3              669.360    3.0       0.0          2.0         3.0
4              678.080    0.0       0.0          2.0         1.0
```

```
from sklearn.preprocessing import MinMaxScaler

# Select numerical columns for normalization
numerical_cols = customer_features.select_dtypes(include=['float64', 'int64']).columns

# Normalize numerical columns
scaler = MinMaxScaler()
customer_features[numerical_cols] = scaler.fit_transform(customer_features[numerical_cols])

print(customer_features.head())
```

```
  CustomerID        CustomerName         Region  SignupDate  TotalSpending  \
0      C0001    Lawrence Carroll  South America  2022-07-10       0.314274
1      C0002      Elizabeth Lutz           Asia  2022-02-13       0.174514
2      C0003      Michael Rivera  South America  2024-03-07       0.255332
3      C0004  Kathleen Rodriguez  South America  2022-10-09       0.501681
4      C0005         Laura Weber           Asia  2022-08-15       0.190581

   AvgTransactionValue  Books  Clothing  Electronics  Home Decor
0             0.507057    0.2       0.0          0.6    0.166667
1             0.351956    0.0       0.4          0.0    0.333333
2             0.514948    0.0       0.2          0.2    0.333333
3             0.505890    0.6       0.0          0.4    0.500000
4             0.512480    0.0       0.0          0.4    0.166667
```

```
from sklearn.metrics.pairwise import cosine_similarity

# Compute cosine similarity matrix
similarity_matrix = cosine_similarity(customer_features[numerical_cols])

# Convert to a DataFrame for easy manipulation
similarity_df = pd.DataFrame(similarity_matrix, index=customer_features['CustomerID'], columns=customer_features['Custo

print(similarity_df.head())
```

```
CustomerID     C0001     C0002     C0003     C0004     C0005     C0006  \
CustomerID
```

```
C0001       1.000000  0.500277  0.808600  0.856999  0.951308  0.711859
C0002       0.500277  1.000000  0.885126  0.586321  0.591543  0.736503
C0003       0.808600  0.885126  1.000000  0.778289  0.889464  0.832574
C0004       0.856999  0.586321  0.778289  1.000000  0.758583  0.810212
C0005       0.951308  0.591543  0.889464  0.758583  1.000000  0.726567

CustomerID     C0007     C0008     C0009     C0010  ...     C0191     C0192  \
CustomerID                                          ...
C0001       0.930154  0.802864  0.580197  0.395416  ...  0.854094  0.928400
C0002       0.612147  0.769940  0.767015  0.804178  ...  0.356461  0.587797
C0003       0.900881  0.849063  0.702609  0.623572  ...  0.594605  0.783015
C0004       0.750220  0.830519  0.411692  0.442937  ...  0.878161  0.775528
C0005       0.993777  0.754367  0.597472  0.387383  ...  0.712469  0.868243

CustomerID     C0193     C0194     C0195     C0196     C0197     C0198  \
CustomerID
C0001       0.593943  0.898720  0.709999  0.714257  0.954553  0.800502
C0002       0.374319  0.698138  0.949697  0.813518  0.585653  0.730442
C0003       0.494673  0.846411  0.967996  0.918740  0.885190  0.853774
C0004       0.787605  0.963363  0.744878  0.775095  0.759178  0.580685
C0005       0.484430  0.810961  0.770387  0.789633  0.999660  0.863398

CustomerID     C0199     C0200
CustomerID
C0001       0.919925  0.788766
C0002       0.631991  0.844707
C0003       0.894424  0.921100
C0004       0.807205  0.776745
C0005       0.950526  0.811187

[5 rows x 200 columns]
```

```
# Function to get top 3 similar customers
def get_top_3_similar(customers_df, customer_id):
    similar_customers = customers_df[customer_id].sort_values(ascending=False).iloc[1:4]
    return list(zip(similar_customers.index, similar_customers.values))

# Generate recommendations for customers C0001 to C0020
lookalike_data = {}
for customer_id in customer_features['CustomerID'][:20]:
    lookalike_data[customer_id] = get_top_3_similar(similarity_df, customer_id)

# Convert to a DataFrame
lookalike_df = pd.DataFrame({
    'CustomerID': lookalike_data.keys(),
    'Lookalikes': [value for value in lookalike_data.values()]
})

print(lookalike_df)
```

```
   CustomerID                                         Lookalikes
0       C0001  [(C0146, 0.9849093056855885), (C0035, 0.984569...
1       C0002  [(C0134, 0.9861738653029506), (C0133, 0.976658...
2       C0003  [(C0166, 0.9981368903264345), (C0158, 0.990010...
3       C0004  [(C0017, 0.9721830996350582), (C0113, 0.969024...
4       C0005  [(C0197, 0.9996595915950468), (C0007, 0.993776...
5       C0006  [(C0135, 0.9910796141297742), (C0167, 0.983186...
6       C0007  [(C0005, 0.9937765536360452), (C0197, 0.990532...
7       C0008  [(C0162, 0.9656006917907872), (C0181, 0.945891...
8       C0009  [(C0034, 0.951112236091614), (C0092, 0.9490201...
9       C0010  [(C0077, 0.9895437462584057), (C0083, 0.979444...
10      C0011  [(C0126, 0.9921044763819368), (C0027, 0.989205...
11      C0012  [(C0065, 0.9876468679598578), (C0152, 0.983406...
12      C0013  [(C0107, 0.9896481906759469), (C0105, 0.988992...
13      C0014  [(C0151, 0.9852344025310596), (C0128, 0.968680...
14      C0015  [(C0123, 0.9969369053054044), (C0073, 0.972094...
15      C0016  [(C0183, 0.9999951812780966), (C0107, 0.999447...
16      C0017  [(C0075, 0.9883295070302321), (C0194, 0.986684...
17      C0018  [(C0168, 0.9788664595481981), (C0125, 0.973593...
18      C0019  [(C0191, 0.9571261980012316), (C0121, 0.942932...
19      C0020  [(C0130, 0.9909853390997839), (C0140, 0.970342...
```

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

Start coding or generate with AI.