Lookalike Model;

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

```
customers_file = 'Customers.csv'
products_file = 'Products.csv'
transactions_file = 'Transactions.csv'
```

```
customers_df = pd.read_csv(customers_file)
products_df = pd.read_csv(products_file)
transactions_df = pd.read_csv(transactions_file)
#reading the files
```

```
transactions_customers = pd.merge(transactions_df, customers_df, on="CustomerID", how="left")
merged_df = pd.merge(transactions_customers, products_df, on="ProductID", how="left")
#merging datasets
```

```
#Create customer profiles
def create_customer_profiles(data):
    profiles = (
        data.groupby("CustomerID")
        .agg(
            {
                "TotalValue": "sum",  # Total transaction value
                "Quantity": "sum",  # Total quantity purchased
                "ProductID": "nunique",  # Number of unique products purchased
                "Category": lambda x: x.mode()[0] if len(x) > 0 else None,  # Most common category
                "Region": "first",  # Customer's region
            }
        )
        .reset_index()
    )
    return profiles

customer_profiles = create_customer_profiles(merged_df)
```

```
# Encoding
customer_profiles_encoded = pd.get_dummies(customer_profiles, columns=["Category", "Region"], drop_first=True)
```

```
# Normalize numerical columns
scaler = StandardScaler()
numerical_cols = ["TotalValue", "Quantity", "ProductID"]
customer_profiles_encoded[numerical_cols] = scaler.fit_transform(customer_profiles_encoded[numerical_cols])
```

```
# Calculate similarity matrix
def calculate_similarity_matrix(data):
    return cosine_similarity(data)

customer_ids = customer_profiles_encoded["CustomerID"]
similarity_matrix = calculate_similarity_matrix(customer_profiles_encoded.drop(columns=["CustomerID"]))
```

```
# Recommend top 3 similar customers
def get_top_3_similar(customers, sim_matrix, num_customers=20):
    lookalike_map = {}
    for i in range(num_customers):  # First 20 customers
        customer_id = customers.iloc[i]
        similarities = list(enumerate(sim_matrix[i]))
        similarities = sorted(similarities, key=lambda x: x[1], reverse=True)  # Sort by similarity score
```

```
        top_3 = [(customers.iloc[j], score) for j, score in similarities[1:4]]  # Exclude self
        lookalike_map[customer_id] = top_3
    return lookalike_map
```

```
# Get lookalikes for the first 20 customers
lookalike_map = get_top_3_similar(customer_ids, similarity_matrix)
```

```
# Generate Lookalike.csv
def generate_lookalike_csv(lookalike_map, output_file="Lookalike.csv"):
    lookalike_data = []
    for cust_id, lookalikes in lookalike_map.items():
        for lookalike_id, score in lookalikes:
            lookalike_data.append({"CustomerID": cust_id, "LookalikeID": lookalike_id, "Score": score})
    lookalike_df = pd.DataFrame(lookalike_data)
    lookalike_df.to_csv(output_file, index=False)
    print(f"{output_file} generated successfully!")
```

```
# Generate the CSV and display the first 20 results
generate_lookalike_csv(lookalike_map)
```

```
Lookalike.csv generated successfully!
```

```
# Print lookalike recommendations for the first 20 customers
for customer_id, recommendations in lookalike_map.items():
    print(f"Customer {customer_id}:")
    for lookalike_id, score in recommendations:
        print(f"  Lookalike: {lookalike_id}, Score: {score:.4f}")
```

```
  Lookalike: C0187, Score: 0.8966
  Lookalike: C0011, Score: 0.8729
Customer C0007:
  Lookalike: C0005, Score: 0.9866
  Lookalike: C0115, Score: 0.9825
  Lookalike: C0140, Score: 0.9770
Customer C0008:
  Lookalike: C0065, Score: 0.8779
  Lookalike: C0090, Score: 0.8772
  Lookalike: C0139, Score: 0.8510
Customer C0009:
  Lookalike: C0198, Score: 0.9877
  Lookalike: C0061, Score: 0.9676
  Lookalike: C0062, Score: 0.9271
Customer C0010:
  Lookalike: C0111, Score: 0.8899
  Lookalike: C0062, Score: 0.8798
  Lookalike: C0061, Score: 0.8539
Customer C0011:
  Lookalike: C0137, Score: 0.9278
  Lookalike: C0174, Score: 0.8903
  Lookalike: C0191, Score: 0.8788
```

```
    Lookalike: C0041, Score: 0.9329
    Lookalike: C0175, Score: 0.9000
  Customer C0018:
    Lookalike: C0046, Score: 0.8774
    Lookalike: C0122, Score: 0.8409
    Lookalike: C0068, Score: 0.7733
  Customer C0019:
    Lookalike: C0172, Score: 0.8080
    Lookalike: C0081, Score: 0.7612
    Lookalike: C0121, Score: 0.7537
  Customer C0020:
    Lookalike: C0015, Score: 0.9426
    Lookalike: C0140, Score: 0.9390
    Lookalike: C0186, Score: 0.9193
```

Start coding or generate with AI.