# Lookalikemodel

In [16]:

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

# Files
customers_file = "Customers.csv.csv"
products_file = "Products.csv.csv"
transactions_file = "Transactions.csv.csv"

# Read data
customers_df = pd.read_csv(customers_file)
products_df = pd.read_csv(products_file)
transactions_df = pd.read_csv(transactions_file)

# Convert date columns to datetime
customers_df['SignupDate'] =
pd.to_datetime(customers_df['SignupDate'])
transactions_df['TransactionDate'] =
pd.to_datetime(transactions_df['TransactionDate'])

# Merge datasets
merged_df = transactions_df.merge(customers_df,
on='CustomerID').merge(products_df, on='ProductID')

# Rename columns
merged_df.rename(columns={'Price_x': 'TransactionPrice', 'Price_y':
'ProductPrice'}, inplace=True)

# Feature Engineering
customer_features = merged_df.groupby('CustomerID').agg({
    'TotalValue': 'sum',  # Total spending
    'Quantity': 'sum',  # Total quantity purchased
    'ProductPrice': 'mean',  # Average price of products purchased
    'Region': lambda x: x.mode()[0],  # Most frequent region
    'Category': lambda x: ','.join(x.unique())  # Unique categories
purchased
}).reset_index()

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

# Normalize numerical features
```

```
scaler = StandardScaler()
numerical_cols = ['TotalValue', 'Quantity', 'ProductPrice']
customer_features[numerical_cols] =
scaler.fit_transform(customer_features[numerical_cols])

# Compute Similarity Matrix
similarity_matrix =
cosine_similarity(customer_features[numerical_cols])
similarity_df = pd.DataFrame(similarity_matrix,
index=customer_features['CustomerID'],
columns=customer_features['CustomerID'])

# Generate Lookalike Recommendations
lookalike_map = {}
for customer_id in customers_df['CustomerID'][:20]:  # First 20
customers (C0001 - C0020)
    similar_customers =
similarity_df[customer_id].sort_values(ascending=False)[1:4]  # Top 3
similar customers
    lookalike_map[customer_id] = list(zip(similar_customers.index,
similar_customers.values))

# Create Lookalike.csv
lookalike_data = []
for customer_id, lookalikes in lookalike_map.items():
    for similar_id, score in lookalikes:
        lookalike_data.append({'CustomerID': customer_id,
'LookalikeID': similar_id, 'Score': score})

lookalike_df = pd.DataFrame(lookalike_data)

# Save to CSV in the current working directory
lookalike_df.to_csv('Lookalike.csv', index=False)

print("Lookalike Model completed. Results saved to Lookalike.csv.")
```

```
Lookalike Model completed. Results saved to Lookalike.csv.
```

In [ ]:

In [ ]: