TASK 2 :

```
import pandas as pd

from sklearn.metrics.pairwise import cosine_similarity

from sklearn.preprocessing import StandardScaler


# Step 1: Load the Data

customers = pd.read_csv(r'D:/Downloads/Customers.csv')  # Replace with actual path if needed

transactions = pd.read_csv(r'D:/Downloads/Transactions.csv')

# Step 2: Data Preprocessing

# Merge datasets on CustomerID

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

# Check for missing values and clean if necessary

data.dropna(inplace=True)

# Step 3: Feature Engineering

# Create features for each customer

customer_features = data.groupby('CustomerID').agg({

   'TotalValue': 'sum',

   'TransactionID': 'count',

   'ProductID': lambda x: x.nunique()  # Number of unique products purchased

}).reset_index()

customer_features.columns = ['CustomerID', 'TotalSpend', 'PurchaseFrequency', 'UniqueProducts']

# Step 4: Calculate Similarity Scores

# Scale the features for better similarity calculation

scaler = StandardScaler()

scaled_features = scaler.fit_transform(customer_features[['TotalSpend', 'PurchaseFrequency', 'UniqueProducts']])


# Calculate cosine similarity matrix

similarity_matrix = cosine_similarity(scaled_features)
```

```
# Create a DataFrame for similarity scores
similarity_df = pd.DataFrame(similarity_matrix, index=customer_features['CustomerID'], columns=customer_features['CustomerID'])
# Step 5: Recommend Lookalikes
lookalike_recommendations = {}
for cust_id in customer_features['CustomerID'][:20]:  # For first 20 customers (C0001 to C0020)
    # Get the similarity scores for the current customer and sort them in descending order
    similar_customers = similarity_df[cust_id].sort_values(ascending=False).head(4)  # Top 4 (including self)
    # Exclude self from recommendations and get top 3 lookalikes
    top_lookalikes = similar_customers.index[1:4]  # Get top 3 excluding self
    scores = similar_customers.values[1:4]        # Corresponding scores
    lookalike_recommendations[cust_id] = list(zip(top_lookalikes, scores))
# Step 6: Create DataFrame for Recommendations
data_for_df = []
for cust_id, lookalikes in lookalike_recommendations.items():
    for lookalike_id, score in lookalikes:
        data_for_df.append({'CustomerID': cust_id, 'LookalikeID': lookalike_id, 'Score': score})

lookalike_df = pd.DataFrame(data_for_df)
# Step 7: Save to CSV file
lookalike_df.to_csv(r'D:/Downloads/Lookalike.csv', index=False)
print("Lookalike recommendations have been saved to Lookalike.csv.")
```

Explanation of Each Step:

1. **Load the Data**: The code reads customer and transaction data from CSV files.
2. **Data Preprocessing**: Merges the two datasets on CustomerID and cleans any missing values.
3. **Feature Engineering**: Aggregates transaction data to create features such as total spend, purchase frequency, and unique products purchased per customer.

4. **Calculate Similarity Scores**: Uses StandardScaler to normalize the features and computes the cosine similarity matrix.
5. **Recommend Lookalikes**: For each of the first 20 customers, it finds their top three similar customers based on similarity scores.
6. **Create DataFrame for Recommendations**: Structures the recommendations into a DataFrame suitable for export.
7. **Save to CSV file**: Exports the recommendations to a CSV file named Lookalike.csv.