```
import pandas as pd
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.compose import ColumnTransformer



# Load datasets
customers = pd.read_csv("/content/drive/MyDrive/Customers.csv")
products = pd.read_csv("/content/drive/MyDrive/Products.csv")
transactions = pd.read_csv("/content/drive/MyDrive/Transactions.csv")




# Merge datasets
data = transactions.merge(products, on="ProductID", how="left").merge(customers, on="Cust




# Aggregate customer profiles
profiles = data.groupby("CustomerID").agg(
    total_spending=("TotalValue", "sum"),
    purchase_frequency=("TransactionID", "count"),
    unique_products=("ProductID", "nunique"),
    regions=("Region", "first"),
    categories=("Category", lambda x: x.mode()[0] if not x.empty else None)
).reset_index()




# Preprocess features
numerical_cols = ["total_spending", "purchase_frequency", "unique_products"]
categorical_cols = ["regions", "categories"]

preprocessor = ColumnTransformer([
    ("num", StandardScaler(), numerical_cols),
    ("cat", OneHotEncoder(), categorical_cols)
])

features = preprocessor.fit_transform(profiles)
features_df = pd.DataFrame(features.toarray() if hasattr(features, "toarray") else featur




# Compute cosine similarity
similarities = cosine_similarity(features_df)




# Find top 3 lookalikes for the first 20 customers
```

```
lookalike_results = {}
for idx, customer_id in enumerate(profiles["CustomerID"][:20]):
    sim_scores = list(enumerate(similarities[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: -x[1])[1:4]  # Exclude self-similarity
    lookalike_results[customer_id] = [(profiles["CustomerID"][i], round(score, 4)) for i,



# Save to CSV
lookalike_df = pd.DataFrame({
    "CustomerID": lookalike_results.keys(),
    "Lookalikes": [str(v) for v in lookalike_results.values()]
})
lookalike_df.to_csv("/content/drive/MyDrive/Lookalike.csv", index=False)

print("Lookalike.csv generated successfully!")
```

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