Lookalike Model

Building a lookalike Model:

```
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np

# Merge datasets
merged = transactions.merge(customers, on="CustomerID").merge(products, on="ProductID")

# Create customer profiles
customer_profiles = merged.groupby("CustomerID").agg({
    "TotalValue": "sum",  # Total spending
    "Quantity": "sum",    # Total quantity purchased
    "Category": lambda x: x.mode()[0],  # Most purchased category
    "Region": "first"     # Customer's region
}).reset_index()

customer_profiles_encoded = pd.get_dummies(customer_profiles, columns=["Category", "Region"])

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
numerical_cols = ["TotalValue", "Quantity"]
customer_profiles_encoded[numerical_cols] =
scaler.fit_transform(customer_profiles_encoded[numerical_cols])

# Calculate similarity scores for the first 20 customers
subset_customers = customer_profiles_encoded.iloc[:20]
similarity_matrix = cosine_similarity(subset_customers.iloc[:, 1:], customer_profiles_encoded.iloc[:, 1:])

# Find top 3 lookalikes for each customer
lookalike_dict = {}
for idx, customer in enumerate(subset_customers["CustomerID"]):
    # Sort scores and get top 3 excluding self (index 0)
    top_indices = np.argsort(-similarity_matrix[idx, :])[1:4]
    lookalikes = [
        (customer_profiles_encoded.iloc[i]["CustomerID"], similarity_matrix[idx, i])
        for i in top_indices
    ]
    lookalike_dict[customer] = lookalikes

# Prepare Lookalike.csv
lookalike_output = pd.DataFrame([
    {"cust_id": customer, "lookalikes": lookalikes}
    for customer, lookalikes in lookalike_dict.items()
])
lookalike_path = "Lookalike.csv"
lookalike_output.to_csv(lookalike_path, index=False)

lookalike_path
```

This file contains:

- **cust_id:** The customer for whom lookalikes are generated.
- **lookalikes:** A list of top 3 similar customers with their similarity scores.