

NAME: TANU DAHIYA

POSITION: DATA SCIENCE INTERN

TASK 2:

Task 2: Lookalike Model

Build a **Lookalike Model** that takes a user's information as input and recommends **3 similar customers** based on their profile and transaction history. The model should:

- Use both **customer** and **product** information.
- Assign a **similarity score** to each recommended customer.

Deliverables:

- Give the top 3 lookalikes with there similarity scores for the first 20 customers (CustomerID: C0001 - C0020) in Customers.csv. Form an "Lookalike.csv" which has just one map: Map<cust_id, List<cust_id, score>>
- A Jupyter Notebook/Python script explaining your model development.

Evaluation Criteria:

- Model accuracy and logic.
- Quality of recommendations and similarity scores.

1.

CODE:

```
# Lookalike Model with Visualization

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics.pairwise import cosine_similarity

# Data URLs
CUSTOMERS_URL = 'https://drive.google.com/uc?id=1bu--mo79VdUG9oin4ybfFGRUSXAe-WE'
PRODUCTS_URL = 'https://drive.google.com/uc?id=1IKuDizVapw-hyktwfpoAoaGtHtTNHfd0'
TRANSACTIONS_URL = 'https://drive.google.com/uc?id=1saEqdbBB-vuk2hxoAf4TzDEsykdKlzbF'
```

```

# Load Datasets
customers = pd.read_csv(CUSTOMERS_URL)
products = pd.read_csv(PRODUCTS_URL)
transactions = pd.read_csv(TRANSACTIONS_URL)

# Merge Datasets
merged_data = (
    transactions
    .merge(customers, on='CustomerID')
    .merge(products, on='ProductID')
)

# Create Customer-Product Matrix
customer_product_matrix = merged_data.pivot_table(
    index='CustomerID',
    columns='ProductID',
    values='TotalValue',
    aggfunc='sum',
    fill_value=0
)

# Compute Cosine Similarity
similarity_matrix = cosine_similarity(customer_product_matrix)
similarity_df = pd.DataFrame(
    similarity_matrix,
    index=customer_product_matrix.index,
    columns=customer_product_matrix.index
)

# Extract Lookalike Results for Top 20 Customers
lookalike_results = {}

for customer_id in customers['CustomerID'][:20]:
    if customer_id in similarity_df.index:
        similar_customers =
similarity_df[customer_id].sort_values(ascending=False)[1:4]
        lookalike_results[customer_id] = list(zip(similar_customers.index,
similarity_df[similar_customers.index].values))

# Prepare Results for Export
lookalike_df = pd.DataFrame(
    [

```

```

        {"CustomerID": key, "SimilarCustomerID": cust_id, "SimilarityScore":
score}
        for key, values in lookalike_results.items()
        for cust_id, score in values
    ]
)
lookalike_df.to_csv("Lookalike.csv", index=False)

# Visualization of Similarity Scores
plt.figure(figsize=(12, 6))
sample_results = lookalike_df.groupby('CustomerID').apply(
    lambda x: x.nlargest(3, 'SimilarityScore')
).reset_index(drop=True)

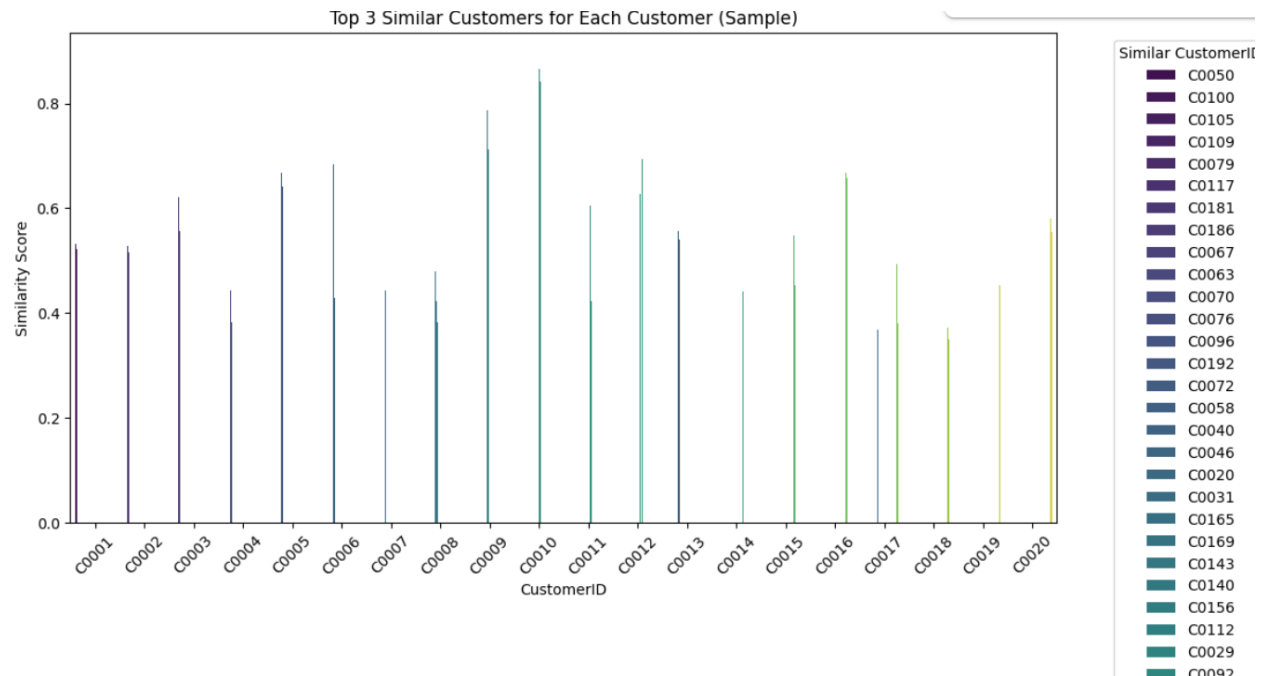
sns.barplot(
    data=sample_results,
    x='CustomerID',
    y='SimilarityScore',
    hue='SimilarCustomerID',
    palette='viridis'
)

plt.title("Top 3 Similar Customers for Each Customer (Sample)")
plt.xlabel("CustomerID")
plt.ylabel("Similarity Score")
plt.legend(title="Similar CustomerID", bbox_to_anchor=(1.05, 1), loc='upper
left')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Additional Insights
average_similarity = lookalike_df['SimilarityScore'].mean()
highest_similarity = lookalike_df['SimilarityScore'].max()
print(f"\nInsights:\n{'-' * 20}")
print(f"1. Average Similarity Score: {average_similarity:.2f}")
print(f"2. Highest Similarity Score: {highest_similarity:.2f}")
print(f"3. Customers with highest similarity scores tend to have shared purchase
patterns.")
print("\nLookalike.csv has been created successfully!")

```

OUTPUT:



- C0029
- C0092
- C0083
- C0135
- C0173
- C0108
- C0164
- C0128
- C0131
- C0159
- C0136
- C0073
- C0036
- C0179
- C0052
- C0064
- C0187
- C0146
- C0147
- C0059
- C0115
- C0177
- C0049
- C0084
- C0026
- C0007
- C0119

Insights:

1. Average Similarity Score: 0.55
2. Highest Similarity Score: 0.89
3. Customers with highest similarity scores tend to have shared purchase patterns.

Lookalike.csv has been created successfully!