Task 2 Lookalike Model:

Deliverables:

**Deliverables Summary**

1. **Jupyter Notebook**:
   * File: FirstName\_LastName\_Lookalike.ipynb
   * Purpose: Explain the model development process, including data preparation, feature engineering, similarity calculation, and results.
2. **CSV File**:
   * File: FirstName\_LastName\_Lookalike.csv
   * Purpose: List top 3 lookalikes for the first 20 customers along with similarity scores.
3. **GitHub Repository**:
   * Repository structure:

lua

CopyEdit

|-- Notebooks/

|-- FirstName\_LastName\_Lookalike.ipynb

|-- Outputs/

|-- FirstName\_LastName\_Lookalike.csv

**Jupyter Notebook Content**

Here’s a breakdown of what should be included in the Jupyter Notebook:

**1. Data Loading**

python

CopyEdit

import pandas as pd

# Load datasets

customers = pd.read\_csv("Customers.csv")

transactions = pd.read\_csv("Transactions.csv")

# Merge datasets

customer\_transactions = transactions.merge(customers, on='CustomerID', how='left')

**2. Feature Engineering**

python

CopyEdit

from sklearn.preprocessing import StandardScaler

# Aggregate transaction data

customer\_profiles = customer\_transactions.groupby('CustomerID').agg({

'TotalValue': 'sum',

'Quantity': 'sum'

}).reset\_index()

# Encode categorical features

customer\_profiles = pd.get\_dummies(customer\_profiles.merge(

customers[['CustomerID', 'Region']], on='CustomerID', how='left'

), columns=['Region'], drop\_first=True)

# Standardize features

scaler = StandardScaler()

features = scaler.fit\_transform(customer\_profiles.drop('CustomerID', axis=1))

**3. Similarity Calculation**

python

CopyEdit

from sklearn.metrics.pairwise import cosine\_similarity

# Compute similarity matrix

similarity\_matrix = cosine\_similarity(features)

# Get top 3 similar customers

lookalike\_results = {}

customer\_ids = customer\_profiles['CustomerID'].tolist()

for i, cust\_id in enumerate(customer\_ids[:20]): # First 20 customers

similar\_indices = similarity\_matrix[i].argsort()[-4:-1][::-1] # Top 3 excluding self

similar\_customers = [(customer\_ids[j], similarity\_matrix[i][j]) for j in similar\_indices]

lookalike\_results[cust\_id] = similar\_customers

**4. Save Results**

python

CopyEdit

# Save results as CSV

results\_df = pd.DataFrame({

'CustomerID': lookalike\_results.keys(),

'Lookalikes': [str(lst) for lst in lookalike\_results.values()]

})

results\_df.to\_csv('FirstName\_LastName\_Lookalike.csv', index=False)

**CSV File Sample**

FirstName\_LastName\_Lookalike.csv:

arduino

CopyEdit

CustomerID,Lookalikes

C0001,"[(C0002, 0.95), (C0005, 0.92), (C0010, 0.89)]"

C0002,"[(C0001, 0.96), (C0003, 0.91), (C0006, 0.89)]"

...

**GitHub Repository Description**

Include a README.md file in the repository with the following content:

**Sample README.md**

markdown

CopyEdit

# Lookalike Model for eCommerce Dataset

## Overview

This project implements a lookalike model that recommends 3 similar customers for the first 20 customers based on their profiles and transaction history.

## Contents

- \*\*Jupyter Notebook\*\*: Explains data preprocessing, feature engineering, similarity computation, and results.

- \*\*CSV File\*\*: Contains the lookalike recommendations.

## Usage

1. Place the datasets (`Customers.csv` and `Transactions.csv`) in the working directory.

2. Run the notebook to generate recommendations.

3. Review the results in `FirstName\_LastName\_Lookalike.csv`.

## Example Result

For `C0001`:

- Lookalikes: `C0002, C0005, C0010`

- Similarity scores: `0.95, 0.92, 0.89`

**Actionable Steps**

1. Populate the Customers.csv and Transactions.csv files with mock or real data if not already available.
2. Execute the Python code to generate results.
3. Organize outputs in the repository structure as outlined.
4. Share the repository link for evaluation.