Data Science Project Implementation

Project Overview

This document provides a step-by-step guide for implementing the Data Science assignment in Google Colab and submitting it using GitHub.

Step 1: Setting Up the Environment

1. Open **Google Colab** (Google Colab).
2. Mount your Google Drive (if required) by running:
3. from google.colab import drive
4. drive.mount('/content/drive')
5. Upload the required dataset files (Customers.csv, Products.csv, Transactions.csv) to Colab.

Step 2: Loading and Exploring the Data

```
import pandas as pd


# Load the datasets
customers = pd.read_csv("Customers.csv")
products = pd.read_csv("Products.csv")
transactions = pd.read_csv("Transactions.csv")


# Display the first few rows
print(customers.head())
print(products.head())
print(transactions.head())
```

Step 3: Data Preprocessing

```
# Checking for missing values
```

```
print(customers.isnull().sum())
print(products.isnull().sum())
print(transactions.isnull().sum())

# Fill missing values or drop rows/columns if necessary
customers.fillna(method='ffill', inplace=True)
products.fillna(method='bfill', inplace=True)
transactions.fillna(0, inplace=True)
```

Step 4: Implementing Customer Clustering

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Selecting features for clustering
features = customers[['Annual_Spend', 'Age']]

# Applying KMeans Clustering
kmeans = KMeans(n_clusters=3, random_state=42)
customers['Cluster'] = kmeans.fit_predict(features)

# Plot the clusters
plt.scatter(customers['Annual_Spend'], customers['Age'], c=customers['Cluster'],
cmap='viridis')
plt.xlabel('Annual Spend')
plt.ylabel('Age')
plt.title('Customer Clusters')
plt.show()
```

Step 5: Implementing Lookalike Modeling

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

# Create similarity matrix
customer_features = customers[['Annual_Spend', 'Age']]
similarity_matrix = cosine_similarity(customer_features)

# Function to recommend similar customers
def recommend(customers, similarity_matrix, top_n=5):
    recommendations = {}
    for idx, customer_id in enumerate(customers['CustomerID']):
        similar_indices = similarity_matrix[idx].argsort()[-top_n-1:-1][::-1]
        recommendations[customer_id] = [customers.iloc[i]['CustomerID'] for i in
similar_indices]
    return recommendations

# Generate recommendations
recommendations = recommend(customers, similarity_matrix)
print(recommendations)
```

Step 6: Saving the Output Files

```
# Save clustered customers and lookalike recommendations
customers.to_csv("Anusha_Paladugu_Clustering.csv", index=False)
recommend_df = pd.DataFrame(list(recommendations.items()), columns=['CustomerID',
'Similar_Customers'])
recommend_df.to_csv("Anusha_Paladugu_Lookalike.csv", index=False)
```

Step 7: Uploading to GitHub

1. Clone GitHub Repository

```
!git clone https://github.com/YourGitHubUsername/DataScienceAssignment.git
```

2. Move Files to Repository Folder

```
import shutil


files = ["Anusha_Paladugu_Clustering.csv", "Anusha_Paladugu_Lookalike.csv",
"Customers.csv", "Products.csv", "Transactions.csv"]

target_folder = "DataScienceAssignment/"


for file in files:
    shutil.move(file, target_folder)
```

3. Commit and Push to GitHub

```
%cd DataScienceAssignment

!git config --global user.email "your_email@example.com"

!git config --global user.name "Your Name"

!git add .

!git commit -m "Added CSV and project files"

!git push origin main
```

Step 8: Verifying Submission

1. Go to your GitHub repository.
2. Confirm that the CSV files and project files have been uploaded.
3. Share the GitHub repository link for submission.

Conclusion

This document provides a complete step-by-step implementation of the Data Science project, including data preprocessing, clustering, lookalike modelling, and submission via GitHub. If you encounter any issues, recheck the commands and ensure your GitHub authentication is correctly set up.