Project: Cloud Data Engineering – Cleaning & Transforming Customer Dataset using Google Colab + Google Drive

Overview

This project demonstrates how to use Google Colab to:

- Mount and read a CSV file from Google Drive
- Perform data cleaning and transformations on a customer dataset
- Save the final dataset back to Google Drive

Dataset used: customers-100000.csv

Step 1: Mount Google Drive

Mount Google Drive to access and save files:

```
from google.colab import drive
drive.mount('/content/drive')
```

Step 2: Load the Dataset

Load the customers-100000.csv file from the specified path in Google Drive:

```
import pandas as pd

file_path = '/content/drive/My Drive/customers-100000.csv'
df = pd.read_csv(file_path)

# View the first few rows
df.head()
```

Step 3: Clean the Data

Remove Duplicates:

df.drop_duplicates(inplace=True)

Handle Missing Values:

- Fill numeric columns with the median
- Fill categorical columns with the mode

```
# Check for missing values
print("\nMissing values per column:")
print(df.isnull().sum())

# Fill missing numeric columns
for col in df.select_dtypes(include='number').columns:
    df[col].fillna(df[col].median(), inplace=True)

# Fill missing categorical columns
for col in df.select_dtypes(include='object').columns:
    df[col].fillna(df[col].mode()[0], inplace=True)

# Fill numeric columns with median
numeric_cols = df.select_dtypes(include='number').columns
for col in numeric_cols:
    median_value = df[col].median()
    df[col].fillna(median_value, inplace=True)

# Fill object/categorical columns with mode
categorical_cols = df.select_dtypes(include='object').columns
for col in categorical_cols:
```

```
    mode_value = df[col].mode()[0]
    df[col].fillna(mode_value, inplace=True)

# Confirm all missing values are handled
print("\nMissing values after cleaning:")
print(df.isnull().sum())

# Fill missing numeric values with median
for col in df.select_dtypes(include='number').columns:
    df[col].fillna(df[col].median(), inplace=True)

# Fill missing categorical with mode
for col in df.select_dtypes(include='object').columns:
    df[col].fillna(df[col].mode()[0], inplace=True)
```

Step 4: Transform the Data

```
1. Create Age Groups
if 'age' in df.columns:
    bins = [0, 18, 30, 45, 60, 120]
    labels = ['Teen', 'Young Adult', 'Adult', 'Middle Aged', 'Senior']
    df['age_group'] = pd.cut(df['age'], bins=bins, labels=labels)

2. Cap Outliers for annual_income
if 'annual_income' in df.columns:
    q1 = df['annual_income'].quantile(0.25)
    q3 = df['annual_income'].quantile(0.75)
    iqr = q3 - q1
    upper_limit = q3 + 1.5 * iqr
```

```
    df['annual_income_capped'] = df['annual_income'].apply(lambda x: min(x,
upper_limit))
```

3. Binary Encode gender

python

Copy code

```
if 'gender' in df.columns:
    df['gender_binary'] = df['gender'].map({'Male': 1, 'Female': 0})
```

4. Combine First and Last Name

python

Copy code

```
if {'first_name', 'last_name'}.issubset(df.columns):
    df['full_name'] = df['first_name'] + ' ' + df['last_name']
```

5. Create Spending Score Ratio

python

Copy code

```
if {'spending_score', 'annual_income'}.issubset(df.columns):
    df['score_per_income'] = df['spending_score'] / df['annual_income'].replace(0, 1)
```

Step 5: Save the Final Dataset

Save the transformed data back to Google Drive:

```
output_path = '/content/drive/My Drive/transformed_customers.csv'
df.to_csv(output_path, index=False)
print("Cleaned and transformed dataset saved to:", output_path)
```

Recommendations for Next Steps

- Perform data visualization using matplotlib or seaborn
- Apply clustering or segmentation on customer data
- Upload the transformed dataset into BigQuery or other cloud platforms for further analytics