# Task 1: Data Cleaning & Preprocessing - Customers Dataset

# ---------------------------------------------------------

# Run this code in Google Colab

import pandas as pd

import numpy as np

# ===============================

# 1. Data Ingestion

# ===============================

# Customers dataset (from PDF source)

url = "https://people.sc.fsu.edu/~jburkardt/data/csv/hw\_200.csv" # replaceable with customers-100.csv if available

df = pd.read\_csv(url)

print("Initial Dataset Shape:", df.shape)

print("\nBasic Info:")

print(df.info())

print("\nFirst 5 rows:")

print(df.head())

# ===============================

# 2. Deduplication

# ===============================

print("\nDuplicate rows before removal:", df.duplicated().sum())

df = df.drop\_duplicates()

print("Duplicate rows after removal:", df.duplicated().sum())

# ===============================

# 3. Column Management

# ===============================

# Drop irrelevant columns if they exist

drop\_cols = [col for col in ["id", "temp\_id", "notes", "index"] if col in df.columns]

df = df.drop(columns=drop\_cols, errors="ignore")

# Rename columns for clarity

df = df.rename(columns=lambda x: x.strip().lower().replace(" ", "\_"))

# ===============================

# 4. Missing Value Handling

# ===============================

print("\nMissing values before handling:\n", df.isna().sum())

# Drop columns if more than 70% missing

threshold = 0.7 \* len(df)

df = df.dropna(axis=1, thresh=threshold)

# Impute numerical missing with median

for col in df.select\_dtypes(include=np.number).columns:

df[col].fillna(df[col].median(), inplace=True)

# Impute categorical missing with mode

for col in df.select\_dtypes(exclude=np.number).columns:

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

print("\nMissing values after handling:\n", df.isna().sum())

# ===============================

# 5. Data Type Correction

# ===============================

# Convert numerical columns trapped as strings

for col in df.columns:

if df[col].dtype == "object":

# Try numeric conversion

df[col] = pd.to\_numeric(df[col].str.replace(",", "").str.replace("$", ""), errors="ignore")

# Convert any date columns

for col in df.columns:

if "date" in col:

df[col] = pd.to\_datetime(df[col], errors="coerce")

# ===============================

# 6. Format Standardization

# ===============================

for col in df.select\_dtypes(include="object").columns:

df[col] = df[col].astype(str).str.lower().str.strip()

# Example: mapping gender values

if "gender" in df.columns:

df["gender"] = df["gender"].replace({"m": "male", "f": "female"})

# ===============================

# Final Cleaned Dataset

# ===============================

print("\nFinal Dataset Shape:", df.shape)

print("\nSample of cleaned data:")

print(df.head())

# Save cleaned dataset

df.to\_csv("cleaned\_customers\_dataset.csv", index=False)

print("\n✅ Cleaned dataset saved as cleaned\_customers\_dataset.csv")

Output:

