import os

import pandas as pd

import numpy as np

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

# Load the dataset

file\_path = r"C:\Users\samia\OneDrive\Documents\GitHub\springboard\Project Proposal\bank+marketing\bank-additional\bank-additional\bank-additional-full.csv"

# Check if the file exists and load the data

if os.path.exists(file\_path):

bank\_additional\_full = pd.read\_csv(file\_path, sep=';') # Ensure correct separator

else:

print(f"File not found: {file\_path}")

exit()

# Step 1: Check if there are any missing values in the dataset

print(bank\_additional\_full.isnull().sum())

# Step 2: Handle missing values

bank\_additional\_full.fillna(bank\_additional\_full.median(numeric\_only=True), inplace=True)

# Step 3: Identify feature columns

target\_column = "y" # Update if needed

if target\_column not in bank\_additional\_full.columns:

print("Error: Target column not found.")

exit()

X = bank\_additional\_full.drop(columns=[target\_column])

y = bank\_additional\_full[target\_column].map({'yes': 1, 'no': 0}) # Convert target to binary

# Step 4: Train-Test Split BEFORE transformations

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42, stratify=y)

# Step 5: Convert categorical columns to dummies separately

categorical\_columns = X\_train.select\_dtypes(include=['object']).columns

X\_train = pd.get\_dummies(X\_train, columns=categorical\_columns, drop\_first=True)

X\_test = pd.get\_dummies(X\_test, columns=categorical\_columns, drop\_first=True)

# Ensure both train & test have the same dummy columns

missing\_cols = set(X\_train.columns) - set(X\_test.columns)

for col in missing\_cols:

X\_test[col] = 0 # Add missing columns in test set

X\_test = X\_test[X\_train.columns] # Reorder columns to match training set

# Step 6: Standardize Numeric Features (Only on Train, then Apply to Test)

numeric\_columns = X\_train.select\_dtypes(include=['int64', 'float64']).columns

if numeric\_columns.empty:

print("Warning: No numeric columns found.")

else:

scaler = StandardScaler()

X\_train.loc[:, numeric\_columns] = scaler.fit\_transform(X\_train[numeric\_columns])

X\_test.loc[:, numeric\_columns] = scaler.transform(X\_test[numeric\_columns]) # Apply same scaling

# Step 7: Verify Data

print("First 5 rows after preprocessing:\n", X\_train.head())

# Step 8: Save Preprocessed Data (Optional)

X\_train.to\_csv("X\_train\_preprocessed.csv", index=False)

X\_test.to\_csv("X\_test\_preprocessed.csv", index=False)

print("Preprocessed data saved successfully.")