import pandas as pd diabetes

import numpy as np

from sklearn. model selection import train\_test\_split from sklearn.neighbors import KNeighborsClassifier from sklearn.preprocessing import StandardScaler

from sklearn.metrics import confusion\_matrix, accuracy\_score, precision\_score, recall\_score

**--------------------------------------------------------------------------------------------------------------------------------------------------**

df = pd.read\_csv('diabetes.csv')

**--------------------------------------------------------------------------------------------------------------------------------------------------**

df.head()

**--------------------------------------------------------------------------------------------------------------------------------------------------**

df.isnull().sum()

**--------------------------------------------------------------------------------------------------------------------------------------------------**

X = df.drop('Outcome', axis=1) # Features excluding 'Outcome'

y = df['Outcome'] # Target variable 'Outcome' (0: No diabetes, 1: Diabetes)

**--------------------------------------------------------------------------------------------------------------------------------------------------**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

**--------------------------------------------------------------------------------------------------------------------------------------------------**

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train) X\_test\_scaled = scaler.transform(X\_test)

**--------------------------------------------------------------------------------------------------------------------------------------------------**

knn = KNeighborsClassifier(n\_neighbors=5)

**--------------------------------------------------------------------------------------------------------------------------------------------------**

knn.fit(X\_train\_scaled, y\_train)

**--------------------------------------------------------------------------------------------------------------------------------------------------**

y\_pred = knn.predict(X\_test\_scaled)

**--------------------------------------------------------------------------------------------------------------------------------------------------**

conf\_matrix = confusion\_matrix(y\_test, y\_pred) print("Confusion Matrix:")

print(conf\_matrix)

**------------------------------------------------------------------------------------------------------------------------------------------------**

# Accuracy

accuracy = accuracy\_score(y\_test, y\_pred) print(f"Accuracy: {accuracy:.2f}")

**------------------------------------------------------------------------------------------------------------------------------------------------**

error\_rate = 1 - accuracy

print(f"Error Rate: {error\_rate:.2f}")

**--------------------------------------------------------------------------------------------------------------------------------------------------**

# Precision

precision = precision\_score(y\_test, y\_pred) print(f"Precision: {precision:.2f}")

**------------------------------------------------------------------------------------------------------------------------------------------------**

recall = recall\_score(y\_test, y\_pred) print(f"Recall: {recall:.2f}")