import pandas as pd

import numpy as np

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

from sklearn.preprocessing import MinMaxScaler

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import plot\_confusion\_matrix, classification\_report

import matplotlib.pyplot as plt

# load data

data = pd.read\_csv("diabetes\_sep2022.csv")

print(data.head(10))

# creating null data

data[["Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI"]] = data[["Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI"]].replace(0, np.NAN)

#checking null data

res = data.isnull().sum()

print(res)

# handle null data

ndata = data.fillna({

"BloodPressure" : data["BloodPressure"].mean(),

"Glucose" : data["Glucose"].mean(),

"SkinThickness" : data["SkinThickness"].mean(),

"Insulin" : data["Insulin"].mean(),

"BMI" : data["BMI"].mean(),

})

print(ndata)

#checking null data

res1 = data.isnull().sum()

print(res1)

#feature and target

features = ndata.drop("Outcome" , axis="columns")

target = ndata["Outcome"]

print(features)

print(target)

# feature scaling

mms = MinMaxScaler()

nfeatures = mms.fit\_transform(features)

print(features)

#print(nfeatures)

# value of k

N = int(len(data) \*\* .5 )

if N % 2 == 0:

N = N + 1

#model and fit

model = KNeighborsClassifier(n\_neighbors=N, metric="euclidean")

model.fit(nfeatures, target)

pr = float(input("Enter Pregnancies "))

gl = float(input("Enter Glucose "))

bp = float(input("Enter Blood Pressure "))

st = float(input("Enter Skin Thickness "))

ins = float(input("Enter Insulin "))

bmi = float(input("Enter BMI "))

pf = float(input("Enter Diabetes Pedigree Function "))

age = float(input("Enter Age "))

data = [[pr, gl, bp, st, ins, bmi, pf, age]]

ndata = mms.transform(data)

ans = model.predict(ndata)

print(ans)

ne = model.kneighbors(ndata, n\_neighbors=N)

#print(ne)