*#Name : Anuprita Girme #Roll No: COBA56*

import numpy as np

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

from sklearn.metrics import confusion\_matrix, accuracy\_score, precision\_score, recall\_score, f1\_score

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI |
| \ |  |  |  |  |  |  |
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 |
|  |  |  |  |  |  |  |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 |
|  |  |  |  |  |  |  |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 |
|  |  |  |  |  |  |  |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 |
|  |  |  |  |  |  |  |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 |
|  |  |  |  |  |  |  |
| .. | ... | ... | ... | ... | ... | ... |
|  |  |  |  |  |  |  |
| 763 | 10 | 101 | 76 | 48 | 180 | 32.9 |
|  |  |  |  |  |  |  |
| 764 | 2 | 122 | 70 | 27 | 0 | 36.8 |
|  |  |  |  |  |  |  |
| 765 | 5 | 121 | 72 | 23 | 112 | 26.2 |
|  |  |  |  |  |  |  |
| 766 | 1 | 126 | 60 | 0 | 0 | 30.1 |
|  |  |  |  |  |  |  |
| 767 | 1 | 93 | 70 | 31 | 0 | 30.4 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | DiabetesPedigreeFunction | Age | Outcome |
| 0 | 0.627 | 50 | 1 |
| 1 | 0.351 | 31 | 0 |
| 2 | 0.672 | 32 | 1 |
| 3 | 0.167 | 21 | 0 |
| 4 | 2.288 | 33 | 1 |
| .. | ... | ... | ... |
| 763 | 0.171 | 63 | 0 |
| 764 | 0.340 | 27 | 0 |
| 765 | 0.245 | 30 | 0 |
| 766 | 0.349 | 47 | 1 |
| 767 | 0.315 | 23 | 0 |

[768 rows x 9 columns]

X = data.drop("Outcome", axis=1) y = data["Outcome"]

X

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI |
| \ |  |  |  |  |  |  |
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 |
|  |  |  |  |  |  |  |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 |
|  |  |  |  |  |  |  |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 |
|  |  |  |  |  |  |  |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 |
|  |  |  |  |  |  |  |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 |
|  |  |  |  |  |  |  |
| .. | ... | ... | ... | ... | ... | ... |
|  |  |  |  |  |  |  |
| 763 | 10 | 101 | 76 | 48 | 180 | 32.9 |
|  |  |  |  |  |  |  |
| 764 | 2 | 122 | 70 | 27 | 0 | 36.8 |
|  |  |  |  |  |  |  |
| 765 | 5 | 121 | 72 | 23 | 112 | 26.2 |
|  |  |  |  |  |  |  |
| 766 | 1 | 126 | 60 | 0 | 0 | 30.1 |
|  |  |  |  |  |  |  |
| 767 | 1 | 93 | 70 | 31 | 0 | 30.4 |

|  |  |  |
| --- | --- | --- |
|  | DiabetesPedigreeFunction | Age |
| 0 | 0.627 | 50 |
| 1 | 0.351 | 31 |
| 2 | 0.672 | 32 |
| 3 | 0.167 | 21 |
| 4 | 2.288 | 33 |
| .. | ... | ... |
| 763 | 0.171 | 63 |
| 764 | 0.340 | 27 |
| 765 | 0.245 | 30 |
| 766 | 0.349 | 47 |
| 767 | 0.315 | 23 |

[768 rows x 8 columns]

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

scaler = StandardScaler()

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

X\_train

array([[-0.52639686, -1.15139792, -3.75268255, ..., -4.13525578,

-0.49073479, -1.03594038],

[ 1.58804586, -0.27664283, 0.68034485, ..., -0.48916881,

2.41502991, 1.48710085],

[-0.82846011, 0.56687102, -1.2658623 , ..., -0.42452187,

0.54916055, -0.94893896],

...,

[ 1.8901091 , -0.62029661, 0.89659009, ..., 1.76054443,

1.981245 , 0.44308379],

[-1.13052335, 0.62935353, -3.75268255, ..., 1.34680407,

-0.78487662, -0.33992901],

[-1.13052335, 0.12949347, 1.43720319, ..., -1.22614383,

-0.61552223, -1.03594038]])

k = 3

knn = KNeighborsClassifier(n\_neighbors=k) knn.fit(X\_train, y\_train)

KNeighborsClassifier(n\_neighbors=3)

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

C:\Users\rohit\anaconda3\lib\site-packages\sklearn\neighbors\

\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statistic is taken will be eliminated, and the value None will no longer be accepted. Set

`keepdims` to True or False to avoid this warning.

mode, \_ = stats.mode(\_y[neigh\_ind, k], axis=1)

array([0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1,

0,

0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1,

0,

0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0,

1,

0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,

0,

0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,

1,

0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1,

0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,

0],

dtype=int64)

conf\_matrix = confusion\_matrix(y\_test, y\_pred) accuracy = accuracy\_score(y\_test, y\_pred) error\_rate = 1 - accuracy

precision = precision\_score(y\_test, y\_pred) recall = recall\_score(y\_test, y\_pred)

print("Confusion Matrix:") print(conf\_matrix) print("Accuracy:", accuracy) print("Error Rate:", error\_rate) print("Precision:", precision) print("Recall:", recall)

Confusion Matrix:

[[81 18]

[27 28]]

Accuracy: 0.7077922077922078

Error Rate: 0.29220779220779225

Precision: 0.6086956521739131

Recall: 0.509090909090909