Institute of Computer Technology
B. Tech Computer Science and Engineering
Sub: Data Mining and Warehousing (2CSE60E27)

PRACTICAL 10: K-NEAREST NEIGHBOUR CLASSIFICATION

The attached dataset contains information related to Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction and Age. With respect to these features, a particular person's diabetic status is given in the attribute Outcome. Your task is to divide the dataset into 2 parts and predict whether the particular person with the given features Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction and Age can have diabetes or not! Also derive confusion matrix and accuracy of your result.

```
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import matplotlib as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.preprocessing import StandardScaler

df = pd.read_csv(r'C:\Users\admin\Desktop\dishwa\dmw\Practical 10\KNN.csv')
df.head()

X = df.drop('Outcome',axis=1)
X.head()

y = df[['Outcome']]
y.head()

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
model = KNeighborsClassifier(n_neighbors=5)
model.fit(X_train,y_train) #Train the model using the training sets
predicted= model.predict(X_test) #Predict Output
predicted[:10]

y_test[:10]

print("Accuracy:",metrics.accuracy_score(y_test, predicted))

print("Confusion matrix")
```

print(" ")
print(metrics.confusion_matrix(y_test, predicted))

In [3]:
```
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import matplotlib as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.preprocessing import StandardScaler
```

In [4]:
```
df = pd.read_csv(r'C:\Users\admin\Desktop\dishwa\dmw\Practical 10\KNN.csv')
df.head()
```

Out[4]:

	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
2	183	64	0	0	23.3	0.672	32	1
3	89	66	23	94	28.1	0.167	21	0
4	137	40	35	168	43.1	2.288	33	1

In [5]:
```
X = df.drop('Outcome',axis=1)
X.head()
```

Out[5]:

	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	148	72	35	0	33.6	0.627	50
1	85	66	29	0	26.6	0.351	31
2	183	64	0	0	23.3	0.672	32
3	89	66	23	94	28.1	0.167	21
4	137	40	35	168	43.1	2.288	33

In [6]:
```
y = df[['Outcome']]
y.head()
```

Out[6]:

	Outcome
0	1
1	0
2	1
3	0
4	1

In [7]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
model = KNeighborsClassifier(n_neighbors=5)
model.fit(X_train,y_train) #Train the model using the training sets
predicted= model.predict(X_test) #Predict Output
predicted[:10]
```

```
C:\Users\admin\anaconda3\envs\dmw5\lib\site-packages\sklearn\neighbors\_cla
vector y was passed when a 1d array was expected. Please change the shape o
  return self._fit(X, y)
```

Out[7]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 1, 0], dtype=int64)
```

In [8]:
```
y_test[:10]
```

Out[8]:

	Outcome
547	0
447	0
403	0
271	0
149	0
432	0
464	0
77	0
739	1
752	0

In [9]:
```
print("Accuracy:",metrics.accuracy_score(y_test, predicted))
```

```
Accuracy: 0.7857142857142857
```

In [10]:
```
print("Confusion matrix")
print(" ")
print(metrics.confusion_matrix(y_test, predicted))
```

```
Confusion matrix

[[84 16]
 [17 37]]
```