Assignment No. 4

Aim : Implement K-Nearest Neighbors algorithm on diabetes.csv dataset. Compute confusion matrix, accuracy, error rate, precision and recall on the given dataset.

Code :

In [3]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings ('ignore')
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn import metrics
```

In [4]:
```
df =pd.read_csv("diabetes.csv")
```

In [5]:
```
df.columns
```

Out[5]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [17]:
```
X = df.drop('Outcome', axis = 1 )
y = df['Outcome']
```

In [20]:
```
from sklearn.preprocessing import scale
X= scale(X)
#split into train and test
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,random_
```

In [21]:
```
from sklearn.neighbors import KNeighborsClassifier
knn =KNeighborsClassifier(n_neighbors= 7)
knn.fit(X_train,y_train)
```

```
y_pred=knn.predict(X_test)
```

In [23]:
```
print("Confusion Matrix:")
cs = metrics.confusion_matrix(y_test,y_pred)
print(cs)
```

```
Confusion Matrix:
[[123  28]
 [ 37  43]]
```

In [24]:
```
print('Accuracy:',metrics.accuracy_score(y_test,y_pred))
```

```
Accuracy: 0.7186147186147186
```

In [27]:
```
total_misclassified = cs[0, 1] + cs[1, 0]
print(total_misclassified)
total_examples = cs[0, 0] + cs[0, 1] + cs[1, 0] + cs[1, 1]
print(total_examples)
print("Error_rate", total_misclassified/total_examples)
print("Error_rate", 1-metrics.accuracy_score(y_test,y_pred))
```

```
65
231
Error_rate 0.2813852813852814
Error_rate 0.2813852813852814
```

In [28]:
```
print("Precision Score", metrics.precision_score(y_test,y_pred))
```

```
Precision Score 0.6056338028169014
```

In [29]:
```
print("Recall Score", metrics.recall_score (y_test,y_pred))
```

```
Recall Score 0.5375
```

In [30]:
```
print("Classificatio Report", metrics.classification_report(y_test,y_pred))
```

```
Classificatio Report               precision    recall  f1-score   support

           0       0.77      0.81      0.79       151
           1       0.61      0.54      0.57        80

    accuracy                           0.72       231
   macro avg       0.69      0.68      0.68       231
weighted avg       0.71      0.72      0.71       231
```