Assignement NO 4 Implement K-Nearest Neighbors algorithm on diabetes.csv dataset. Compute confusion matrix, accuracy, error rate, precision and recall on the given dataset.

In [1]:
```
import pandas as pd
import numpy as np
from sklearn import metrics
```

In [2]:
```
df =pd.read_csv("diabetes.csv")
df
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Pedigree	Age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33
...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63
764	2	122	70	27	0	36.8	0.340	27
765	5	121	72	23	112	26.2	0.245	30
766	1	126	60	0	0	30.1	0.349	47
767	1	93	70	31	0	30.4	0.315	23

768 rows × 9 columns

In [3]:
```
df.shape
```

Out[3]:
```
(768, 9)
```

In [4]:
```
#checking for null values
df.isnull().any().value_counts()
```

Out[4]:
```
False    9
Name: count, dtype: int64
```

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

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

In [6]:
```
df_x =df.drop(columns='Outcome',axis=1)
df_y =df['Outcome']
```

In [7]:
```
from sklearn.preprocessing import StandardScaler
scale =StandardScaler()
```

```
scaledx =scale.fit_transform(df_x)
```

In [8]:
```
#split into train and test
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test =train_test_split(scaledx, df_y,test_size=0.2,rand
```

In [9]:
```
#KNN
from sklearn.neighbors import KNeighborsClassifier
knn =KNeighborsClassifier(n_neighbors=7)

knn.fit(x_train,y_train)
y_pred=knn.predict(x_test)
```

In [10]:
```
#confusion matrix
cs =metrics.confusion_matrix(y_test,y_pred)
print("confusion matrix\n",cs)
```

```
confusion matrix
 [[89 10]
 [21 34]]
```

In [11]:
```
#Accuracy score
ac=metrics.accuracy_score(y_test,y_pred)
print("Accuracy score:",ac)
```

```
Accuracy score: 0.7987012987012987
```

In [13]:
```
#Error Rate  (error_rate=1-accuracy)
er=1-ac
print("Error Rate",er)
```

```
Error Rate 0.2012987012987013
```

In [14]:
```
#Precision
p=metrics.precision_score(y_test,y_pred)
print("Precision",p)
```

```
Precision 0.7727272727272727
```

In [15]:
```
#Recall
r=metrics.recall_score(y_test,y_pred)
print("Recall",r)
```

```
Recall 0.6181818181818182
```

In [16]:
```
#classification report
cr=metrics.classification_report(y_test,y_pred)
print("classification report:\n\n",cr)
```

```
classification report:

              precision    recall  f1-score   support

           0       0.81      0.90      0.85        99
           1       0.77      0.62      0.69        55

    accuracy                           0.80       154
   macro avg       0.79      0.76      0.77       154
weighted avg       0.80      0.80      0.79       154
```

In [ ]: