In [2]:
```python
import numpy as np
import pandas as pd
```

In [3]:
```python
dataset=pd.read_csv(r"D:\ML_Course\Works_on_python\Naive Bayes\diabetes.csv")
```

In [4]:
```python
dataset.head(5)
```

Out[4]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Diabetes
0	6	148	72	35	0	33.6	0.627	50	pos
1	1	85	66	29	0	26.6	0.351	31	neg
2	8	183	64	0	0	23.3	0.672	32	pos
3	1	89	66	23	94	28.1	0.167	21	neg
4	0	137	40	35	168	43.1	2.288	33	pos

In [5]:
```python
dataset.isnull().any()
```

Out[5]:
```
Pregnancies                 False
Glucose                     False
BloodPressure               False
SkinThickness               False
Insulin                     False
BMI                         False
DiabetesPedigreeFunction    False
Age                         False
Diabetes                    False
dtype: bool
```

In [6]:
```python
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
dataset["Diabetes"]=le.fit_transform(dataset["Diabetes"])
```

In [7]:
```python
x=dataset.iloc[:,0:8].values
y=dataset.iloc[:,8:9].values
```

```
In [80]: from sklearn.model_selection import train_test_split
         x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

```
In [81]: from sklearn.preprocessing import StandardScaler
         sc=StandardScaler()
         x_train=sc.fit_transform(x_train)
         x_test=sc.fit_transform(x_test)
```

Naive Bayes

```
In [82]: from sklearn.naive_bayes import GaussianNB
         naive=GaussianNB()
```

```
In [83]: naive.fit(x_train,y_train)
```

C:\Users\anikp\Anaconda3\lib\site-packages\sklearn\utils\validation.py:761: DataConversionWarning: A column-vector y was passe
d when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  y = column_or_1d(y, warn=True)

```
Out[83]: GaussianNB(priors=None, var_smoothing=1e-09)
```

```
In [84]: y_pred=naive.predict(x_test)
```

```
In [85]: y_pred
```

```
Out[85]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
                1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0,
                0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0,
                0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1,
                0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1,
                0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0])
```

In [86]: `y_test`

Out[86]:
```
array([[0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [1],
       [0],
       [1],
       [0],
       [0],
       [1],
       [0],
       [0],
       [1],
```

In [87]:
```python
from sklearn.metrics import accuracy_score
acc=accuracy_score(y_test,y_pred)
```

In [88]: `acc`

Out[88]: `0.7922077922077922`

In [89]:
```python
import sklearn.metrics as metrics
fpr,tpr , threshold = metrics.roc_curve(y_test,y_pred)
roc_auc = metrics.auc(fpr,tpr)
```

In [90]:
```python
import matplotlib.pyplot as plt
plt.title("roc naive")
plt.plot(fpr,tpr,'b',label = 'auc = %0.2f'%roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0,1],[0,1],'r--')
plt.xlim([0,1])
plt.ylim([0,1])
plt.ylabel('tpr')
plt.xlabel('fpr')
```

Out[90]: Text(0.5, 0, 'fpr')
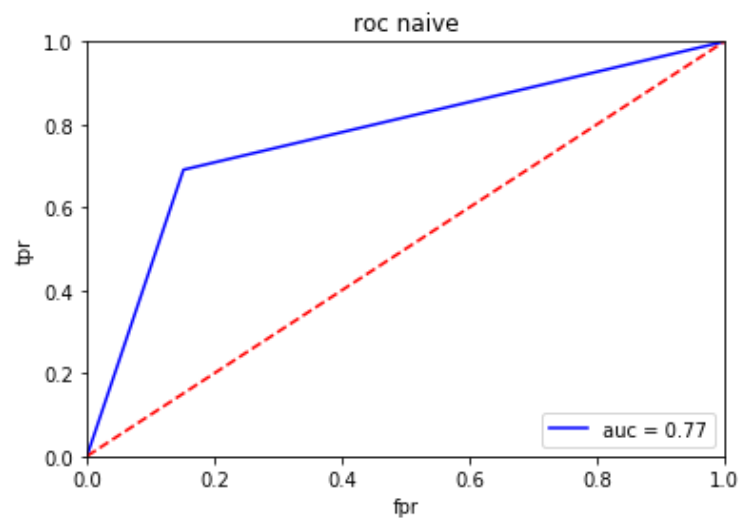
SVM

```
In [91]: from sklearn.svm import SVC
         svm=SVC(kernel="linear")# if data not linear then kernel='rbf'
         svm.fit(x_train,y_train)
```

C:\Users\anikp\Anaconda3\lib\site-packages\sklearn\utils\validation.py:761: DataConversionWarning: A column-vector y was passe
d when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  y = column_or_1d(y, warn=True)

```
Out[91]: SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
             decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
             kernel='linear', max_iter=-1, probability=False, random_state=None,
             shrinking=True, tol=0.001, verbose=False)
```

```
In [92]: y_pred1=svm.predict(x_test)
```

```
In [93]: y_pred1
```

```
Out[93]: array([0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
                0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0,
                0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1,
                0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0,
                0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0,
                0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
                0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0])
```

```
In [94]: acc1=accuracy_score(y_test,y_pred)
```
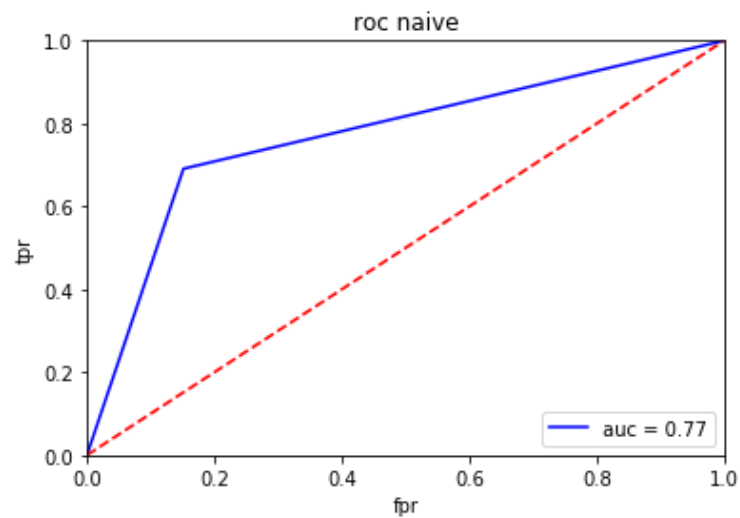
```
In [95]: acc1
```

```
Out[95]: 0.7922077922077922
```

```
In [96]: fpr1,tpr1 , threshold1 = metrics.roc_curve(y_test,y_pred)
         roc_auc = metrics.auc(fpr1,tpr1)
```

In [68]:
```python
import matplotlib.pyplot as plt
plt.title("roc naive")
plt.plot(fpr1,tpr1,'b',label = 'auc = %0.2f'%roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0,1],[0,1],'r--')
plt.xlim([0,1])
plt.ylim([0,1])
plt.ylabel('tpr')
plt.xlabel('fpr')
```

Out[68]: Text(0.5, 0, 'fpr')

In [ ]: