In [3]:

```
import pandas as pd
import numpy as np
from sklearn import preprocessing
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report, roc_curv
from sklearn.model_selection import train_test_split

data=pd.read_csv('covid.csv')
data.head()
```

Out[3]:

	no	pc	wbc	mc	ast	bc	ldh	diagnosis
0	1	Low	Low	Low	High	Normal	Normal	True
1	2	Low	Low	Normal	High	Normal	High	True
2	3	Low	High	Normal	High	Normal	Normal	False
3	4	Low	High	Normal	High	High	Normal	True
4	5	Low	Normal	High	High	Normal	Normal	False

In [4]:

```
le=preprocessing.LabelEncoder()
no=le.fit_transform(data['no'].values)
pc=le.fit_transform(data['pc'].values)
wbc=le.fit_transform(data['wbc'].values)
mc=le.fit_transform(data['mc'].values)
ast=le.fit_transform(data['ast'].values)
bc=le.fit_transform(data['bc'].values)
ldh=le.fit_transform(data['ldh'].values)
y=le.fit_transform(data['diagnosis'].values)
```

In [7]:

```
X=np.array(list(zip(pc,wbc,mc,ast,bc,ldh)))
X
```

Out[7]:

```
array([[1, 1, 1, 0, 1, 1],
       [1, 1, 2, 0, 1, 0],
       [1, 0, 2, 0, 1, 1],
       [1, 0, 2, 0, 0, 1],
       [1, 2, 0, 0, 1, 1],
       [1, 2, 2, 0, 1, 0],
       [2, 1, 1, 0, 1, 1],
       [2, 0, 2, 0, 1, 1],
       [2, 0, 2, 0, 0, 0],
       [2, 2, 0, 0, 1, 1],
       [2, 2, 0, 0, 1, 0],
       [0, 1, 1, 1, 1, 1],
       [0, 2, 0, 1, 1, 1],
       [0, 2, 0, 1, 0, 0],
       [0, 0, 2, 1, 1, 0],
       [1, 2, 0, 0, 0, 1],
       [2, 2, 0, 0, 0, 1],
       [0, 1, 1, 1, 1, 0],
       [2, 2, 2, 0, 1, 1],
       [2, 0, 2, 0, 1, 0],
       [2, 1, 2, 0, 1, 0],
       [1, 0, 2, 0, 0, 0],
       [1, 1, 1, 0, 0, 0],
       [0, 0, 2, 1, 1, 1],
       [0, 2, 2, 1, 1, 1]])
```

In [16]:

```
X_train ,  X_test,Y_train, Y_test=train_test_split(X,y, test_size=0.25)
Y_train
```

Out[16]:

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

In [18]:

```
naivee=MultinomialNB()
naivee.fit(X_train,Y_train)
Y_pred=naivee.predict(X_test)
```

In [19]:

```
print("Accuracy Score:",accuracy_score(Y_test,Y_pred))
print("\nClassification report:\n",classification_report(Y_test,Y_pred))
```

```
Accuracy Score: 0.5714285714285714

Classification report:
               precision    recall  f1-score   support

           0       1.00      0.25      0.40         4
           1       0.50      1.00      0.67         3

    accuracy                           0.57         7
   macro avg       0.75      0.62      0.53         7
weighted avg       0.79      0.57      0.51         7
```

In [34]:
```
lr_probs=naivee.predict_proba(X_test)
lr_probs
```

Out[34]:
```
array([[0.14512894, 0.85487106],
       [0.10617972, 0.89382028],
       [0.62055948, 0.37944052],
       [0.05117072, 0.94882928],
       [0.18654376, 0.81345624],
       [0.08083368, 0.91916632],
       [0.05507014, 0.94492986]])
```

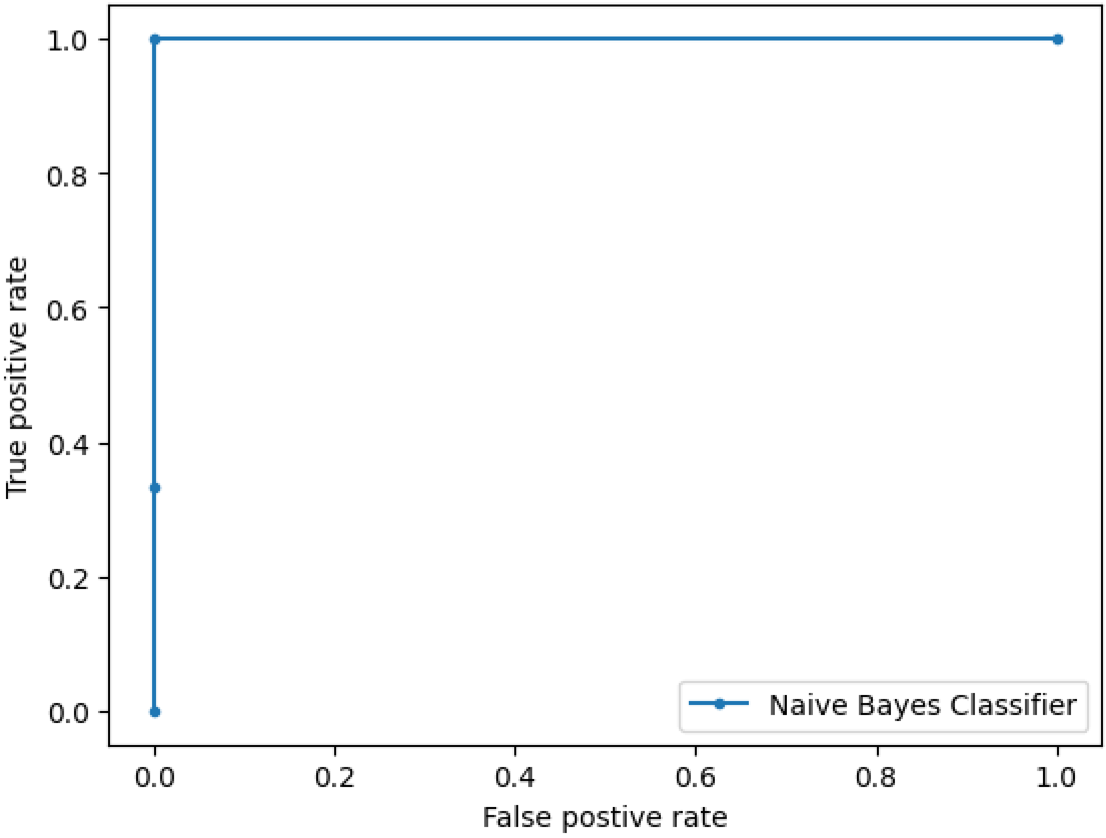
In [35]:
```
lr_probs=lr_probs[:,1]
lr_probs
```

Out[35]:
```
array([0.85487106, 0.89382028, 0.37944052, 0.94882928, 0.81345624,
       0.91916632, 0.94492986])
```

In [36]:
```
lr_fpr, lr_tpr, _ =roc_curve(Y_test,lr_probs)
lr_fpr
```

Out[36]:
```
array([0., 0., 0., 1.])
```

In [37]:
```
import matplotlib.pyplot as plt
plt.plot(lr_fpr, lr_tpr, marker='.', label="Naive Bayes Classifier")
plt.xlabel('False postive rate')
plt.ylabel('True positive rate')
plt.legend()
plt.show()
```



In [ ]: