```
from google.colab import drive
drive.mount('/content/drive')
```

```
     Mounted at /content/drive
```

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
import matplotlib.pyplot as plt
from matplotlib import pyplot
from sklearn import preprocessing
from sklearn.metrics import accuracy_score, classification_report,roc_curve
```

```
df=pd.read_csv("/content/drive/MyDrive/covid.csv")
df.head()
```

	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

```
lb=preprocessing.LabelEncoder()
pc=lb.fit_transform(df["pc"].values)
wbc=lb.fit_transform(df["wbc"].values)
mc=lb.fit_transform(df["mc"].values)
ast=lb.fit_transform(df["ast"].values)
bc=lb.fit_transform(df["bc"].values)
ldh=lb.fit_transform(df["ldh"].values)
y=lb.fit_transform(df["diagnosis"].values)
```

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

```
     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]])
```

```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.25)
naivee = MultinomialNB()
naivee.fit(xtrain,ytrain)
ypred=naivee.predict(xtest)
```

```
print("accuracy:",accuracy_score(ytest,ypred))
print("classification",classification_report(ytest,ypred))
```

```
    :curacy: 0.42857142857142855
    .assification               precision    recall  f1-score   support

              0       0.00      0.00      0.00         4
```

```
           1       0.43      1.00      0.60         3

    accuracy                           0.43         7
   macro avg       0.21      0.50      0.30         7
ighted avg       0.18      0.43      0.26         7

ısr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill
 _warn_prf(average, modifier, msg_start, len(result))
ısr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill
 _warn_prf(average, modifier, msg_start, len(result))
ısr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill
 _warn_prf(average, modifier, msg_start, len(result))
```

```
lb_prob=naivee.predict_proba(xtest)
lb_prob
```

```
array([[0.49758402, 0.50241598],
       [0.08277218, 0.91722782],
       [0.05111746, 0.94888254],
       [0.10336167, 0.89663833],
       [0.09183094, 0.90816906],
       [0.08101151, 0.91898849],
       [0.14573154, 0.85426846]])
```

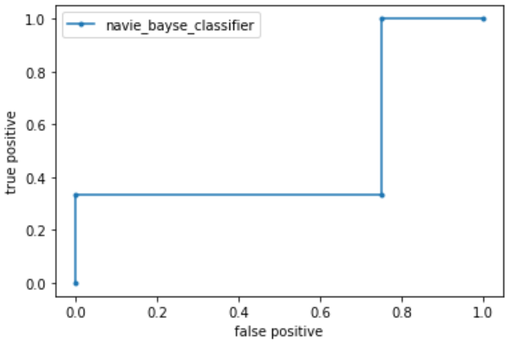
```
lb_prob=lb_prob[:,1]
```

```
lf_prob,lt_prob, _=roc_curve(ytest,lb_prob)
```

```
print(lf_prob)
print(lt_prob)
```

```
[0.   0.   0.75 0.75 1.  ]
[0.         0.33333333 0.33333333 1.         1.        ]
```

```
plt.plot(lf_prob,lt_prob,marker=".", label="navie_bayse_classifier")
plt.xlabel("false positive")
plt.ylabel("true positive")
plt.legend()
plt.show()
```


navie_bayse_classifier
true positive
false positive