

In [1]:

```
import numpy as np
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB

iris = datasets.load_iris()
X = iris.data
y = iris.target
random_state = np.random.RandomState(0)
n_samples, n_features = X.shape
X = np.c_[X, random_state.randn(n_samples, 100 * n_features)]
xTrain, xTest, yTrain, yTest = train_test_split(X[y < 2], y [y < 2], test_size = 0.15)
```

Build linear SVM model and train. Predict for test values.

In [2]:

```
from sklearn import svm
model=svm.SVC(kernel='linear')
model.fit(xTrain,yTrain)
yPred=model.predict(xTest)
```

find accuracy, confusion matrix, precision recall curve and classification report

In [7]:

```

from sklearn import metrics
lAcc = metrics.accuracy_score(yTest, yPred)
lConf = metrics.confusion_matrix(yTest, yPred)
lCReport = metrics.classification_report(yTest,yPred)
print("ACCURACY" + str(lAcc))
print("CONFUSION MATRIX" + str(lConf))
print("CLASSIFICATION REPORT" + str(lCReport))
ls = model.decision_function(xTest)
prec, recall, _ = metrics.precision_recall_curve(yTest, ls)
import matplotlib.pyplot as plt
plt.step(recall, prec, color='r')
plt.xlabel('Recall')
plt.ylabel('Precision')

```

ACCURACY0.733333333333

CONFUSION MATRIX[[6 1]

[3 5]]

CLASSIFICATION REPORT

precision

recall

f1-score

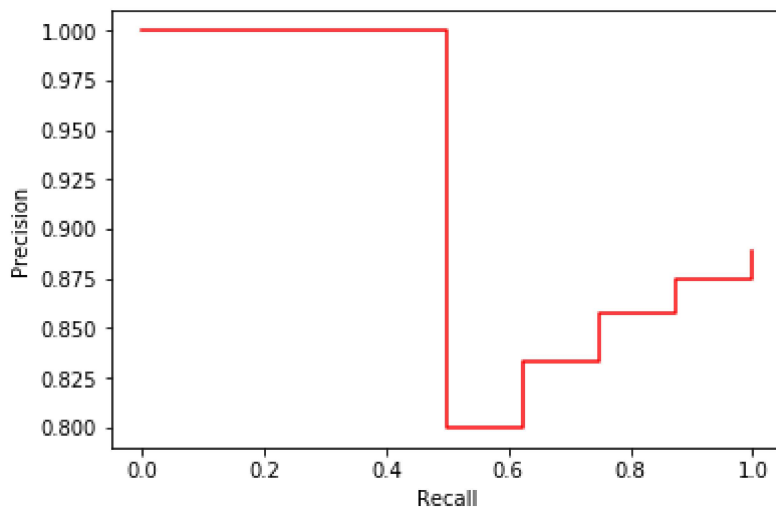
support

0	0.67	0.86	0.75	7
1	0.83	0.62	0.71	8

avg / total	0.76	0.73	0.73	15
-------------	------	------	------	----

Out[7]:

Text(0,0.5,'Precision')



poly svm

In [13]:

```

pModel=svm.SVC(kernel='poly',random_state=20)
pModel.fit(xTrain,yTrain)
polyYPred=pModel.predict(xTest)

pAcc = metrics.accuracy_score(yTest, polyYPred)
pConf = metrics.confusion_matrix(yTest, polyYPred)
pCReport = metrics.classification_report(yTest,polyYPred)
print("ACCURACY" + str(pAcc))
print("CONFUSION MATRIX" + str(pConf))
print("CLASSIFICATION REPORT" + str(pCReport))
ps = model.decision_function(xTest)
prec, recall, _ = metrics.precision_recall_curve(yTest, ps)
plt.step(recall, prec, color='r')
plt.xlabel('Recall')
plt.ylabel('Precision')

```

ACCURACY0.466666666667

CONFUSION MATRIX[[7 0]

[8 0]]

CLASSIFICATION REPORT

precision

recall

f1-score

support

0 0.47 1.00 0.64 7

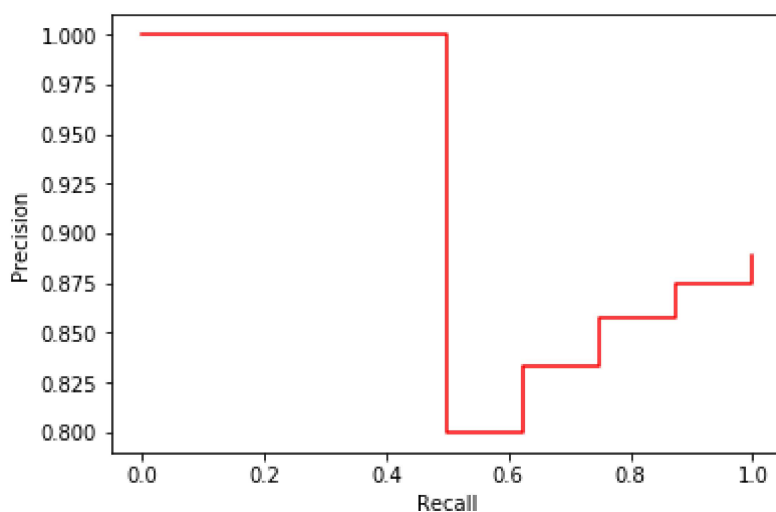
1 0.00 0.00 0.00 8

avg / total 0.22 0.47 0.30 15

C:\Users\User\Anaconda3\lib\site-packages\sklearn\metrics\classification.p  
y:1135: UndefinedMetricWarning: Precision and F-score are ill-defined and  
being set to 0.0 in labels with no predicted samples.  
'precision', 'predicted', average, warn\_for)

Out[13]:

Text(0,0.5,'Precision')



RBF

In [15]:

```

rmodel=svm.SVC(kernel='rbf')
rmodel.fit(xTrain,yTrain)
rYPred=rmodel.predict(xTest)

rAcc = metrics.accuracy_score(yTest, rYPred)
rConf = metrics.confusion_matrix(yTest, rYPred)
rCReport = metrics.classification_report(yTest,rYPred)
print("ACCURACY" + str(rAcc))
print("CONFUSION MATRIX" + str(rConf))
print("CLASSIFICATION REPORT" + str(rCReport))
rs = model.decision_function(xTest)
prec, recall, _ = metrics.precision_recall_curve(yTest, rs)
plt.step(recall, prec, color='r')
plt.xlabel('Recall')
plt.ylabel('Precision')

```

ACCURACY0.733333333333

CONFUSION MATRIX[[6 1]

[3 5]]

CLASSIFICATION REPORT

			precision	recall	f1-score	support
0	0.67	0.86	0.75	7		
1	0.83	0.62	0.71	8		
avg / total	0.76	0.73	0.73	15		

Out[15]:

Text(0,0.5,'Precision')

