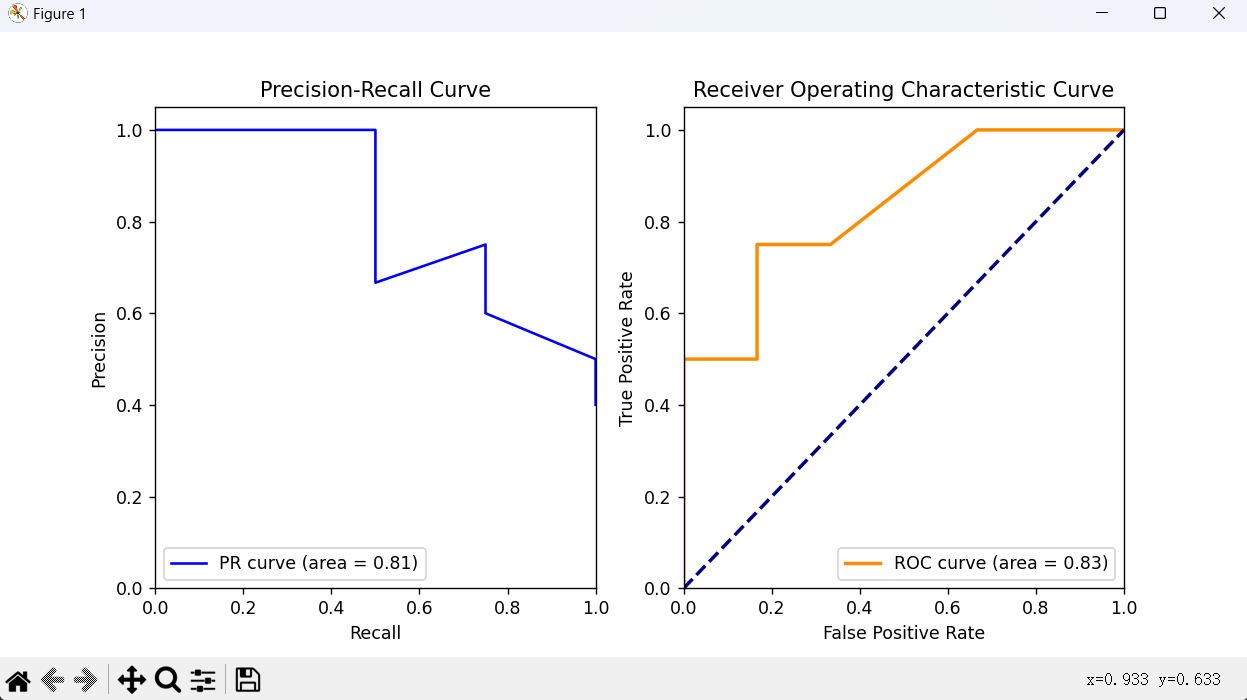
import numpy as np  
import matplotlib.pyplot as plt  
from sklearn.metrics import precision\_recall\_curve, roc\_curve, auc  
  
# 给定的数据  
Y\_true = np.array([1, 0, 0, 1, 0, 0, 1, 1, 0, 0])  
Y\_score = np.array([0.90, 0.40, 0.20, 0.60, 0.50, 0.40, 0.70, 0.40, 0.65, 0.35])  
  
# 计算Precision-Recall曲线的相关指标  
precision, recall, thresholds = precision\_recall\_curve(Y\_true, Y\_score)  
pr\_auc = auc(recall, precision)  
  
# 计算ROC曲线的相关指标  
fpr, tpr, roc\_thresholds = roc\_curve(Y\_true, Y\_score)  
roc\_auc = auc(fpr, tpr)  
  
# 绘制PR曲线  
plt.figure(figsize=(10, 5))  
plt.subplot(1, 2, 1)  
plt.plot(recall, precision, color='b', label='PR curve (area = %0.2f)' % pr\_auc)  
plt.xlabel('Recall')  
plt.ylabel('Precision')  
plt.ylim([0.0, 1.05])  
plt.xlim([0.0, 1.0])  
plt.title('Precision-Recall Curve')  
plt.legend(loc="lower left")  
  
# 绘制ROC曲线  
plt.subplot(1, 2, 2)  
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc\_auc)  
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')  
plt.xlim([0.0, 1.0])  
plt.ylim([0.0, 1.05])  
plt.xlabel('False Positive Rate')  
plt.ylabel('True Positive Rate')  
plt.title('Receiver Operating Characteristic Curve')  
plt.legend(loc="lower right")  
  
plt.show()

输出图像：



import numpy as np  
import matplotlib.pyplot as plt  
from sklearn.metrics import precision\_recall\_curve, roc\_curve, auc  
  
y\_true=np.asarray([[0,0,1],[0,1,0],[1,0,0],[0,0,1],[1,0,0],[0,1,0],[0,1,0],[0,1,0],[0,0,1],[0,1,0]])  
y\_pred=np.asarray([[0.1,0.2,0.7],[0.1,0.6,0.3],[0.5,0.2,0.3],[0.1,0.1,0.8],[0.4,0.2,0.4],[0.6,0.3,0.1],[0.4,0.2,0.4],[0.4,0.1,0.5],[0.1,0.1,0.8],[0.1,0.8,0.1],])  
print(y\_true.shape,y\_pred.shape)  
  
n\_classes=len(y\_true[1,:1])  
fpr =dict()  
tpr =dict()  
roc\_auc=dict()  
  
for i in range(n\_classes):  
 fpr[i],tpr[i],\_=roc\_curve(y\_true[:,i],y\_pred[:,i])  
 roc\_auc[i]=auc(fpr[i],tpr[i])  
  
fpr["micro"], tpr["micro"], \_ = roc\_curve(y\_true.ravel(), y\_pred.ravel())  
  
fpr\_grid=np.linspace(0.0,1.0,100)  
mean\_tpr=np.zeros\_like(fpr\_grid)  
for i in range(n\_classes):  
 mean\_tpr+=np.interp(fpr\_grid,fpr[i],tpr[i])  
mean\_tpr/=n\_classes  
fpr["macro"]=fpr\_grid  
tpr["macro"]=mean\_tpr  
roc\_auc["macro"]=auc(fpr["macro"],tpr["macro"])  
  
y\_true\_list=list([tuple(t)for t in y\_true])  
classNum=dict((a,y\_true\_list.count(a)) for a in y\_true\_list)  
n1=classNum[(1,0,0)]  
n2=classNum[(0,1,0)]  
n3=classNum[(0,0,1)]  
ratio=[n1/(n1+n2+n3),n2/(n1+n2+n3),n3/(n1+n2+n3)]  
avg\_tpr=np.zeros\_like(fpr\_grid)  
for i in range(n\_classes):  
 avg\_tpr +=ratio[i]\*np.interp(fpr\_grid,fpr[i],tpr[i])  
fpr["weighted"]=fpr\_grid  
tpr["weighted"]=avg\_tpr  
roc\_auc["weighted"]=auc(fpr["weighted"],tpr["weighted"])  
  
# 绘制ROC曲线  
plt.figure(figsize=(10, 8))  
  
# 绘制每个类别的ROC曲线  
colors = ['aqua', 'darkorange', 'cornflowerblue']  
for i, color in zip(range(n\_classes), colors):  
 plt.plot(fpr[i], tpr[i], color=color, lw=2, label='ROC curve of class {0} (area = {1:0.2f})'  
 ''.format(i, roc\_auc[i]))  
  
# 绘制微平均ROC曲线  
plt.plot(fpr["micro"], tpr["micro"], color='gold', lw=2, linestyle=':',  
 label='micro-average ROC curve (area = {0:0.2f})'  
 ''.format(roc\_auc["macro"]))  
  
# 绘制宏平均ROC曲线  
plt.plot(fpr["macro"], tpr["macro"], color='deeppink', lw=2, linestyle='--',  
 label='macro-average ROC curve (area = {0:0.2f})'  
 ''.format(roc\_auc["macro"]))  
  
# 绘制加权平均ROC曲线  
plt.plot(fpr["weighted"], tpr["weighted"], color='cyan', lw=2, linestyle='-.',  
 label='weighted-average ROC curve (area = {0:0.2f})'  
 ''.format(roc\_auc["weighted"]))  
  
# 绘制对角线  
plt.plot([0, 1], [0, 1], 'k--', lw=2)  
plt.xlim([0.0, 1.0])  
plt.ylim([0.0, 1.05])  
plt.xlabel('False Positive Rate')  
plt.ylabel('True Positive Rate')  
plt.title('Receiver Operating Characteristic Example')  
plt.legend(loc="lower right")  
plt.show()

输出图像：

