

```

import numpy as np
import matplotlib.pyplot as plt
from sklearn import metrics
from sklearn.metrics import 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]])

n_classes = len(y_true[0])
fpr = dict()
tpr = dict()
roc_auc = dict()

# 计算每个类别的 ROC 曲线
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])

# 计算 micro-average ROC
micro_fpr,micro_tpr,_ = roc_curve(y_true.ravel(),y_pred.ravel())
micro_auc = auc(micro_fpr,micro_tpr)

# 计算 macro-average ROC
fpr_grid = np.linspace(0.0,1.0,100)
macro_tpr = np.zeros_like(fpr_grid)
for i in range(n_classes):
    macro_tpr += np.interp(fpr_grid,fpr[i],tpr[i])
macro_tpr /=n_classes
macro_auc = auc(fpr_grid,macro_tpr)

# 计算 weighted-average ROC
weights = np.array([np.sum(y_true[:,i])for i in range(n_classes)])
weighted_tpr = np.zeros_like(fpr_grid)
for i in range(n_classes):
    weighted_tpr += weights[i]*np.interp(fpr_grid,fpr[i],tpr[i])
weighted_tpr/=np.sum(weights)
weighted_auc = auc(fpr_grid,weighted_tpr)

# 画出每个类别的曲线

```

```
colors = ['red', 'orange', 'green']
for i in range(n_classes):
    plt.plot(fpr[i], tpr[i], color=colors[i], label='ROC curve of
class{0}(area={1:0.2f})' ''.format(i, roc_auc[i]))

# micro-average ROC 曲线
plt.plot(micro_fpr, micro_tpr,
         color='deeppink', linestyle=':', label='Micro-average ROC curve
(area = {0:0.2f})'
         ''.format(micro_auc))

# macro-average ROC 曲线
plt.plot(fpr_grid, macro_tpr, color='blue',
         linestyle=':', label='Macro-average ROC curve (area =
{0:0.2f})'
         ''.format(macro_auc), linewidth=2)

# weighted-average ROC 曲线
plt.plot(fpr_grid, weighted_tpr, color='black',
         linestyle='--', label='Weighted-average ROC curve (area =
{0:0.2f})'
         ''.format(weighted_auc), linewidth=2)

plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('six curves')
plt.legend(loc="lower right")
plt.show()
```

