

二分类模型评价

混淆矩阵 (confusion matrix)

反类	TN	FP
正类	FN	TP
	预测为反类	预测为正类

`((y==1) & (y_pred==1)).sum()`

使用 sklearn 计算混淆矩阵

`sklearn.metrics.confusion_matrix`

主要参数:

- `y_true`: 真实的 y 值
- `y_pred`: 预测的 y 值

精度 (Accuracy)

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

反类	TN	FP
正类	FN	TP
	预测为反类	预测为正类

准确率 (Precision)

$$\text{Precision} = \frac{TP}{TP + FP}$$

反类	TN	FP
正类	FN	TP
	预测为反类	预测为正类

在所有正类的预测当中，
有多少是正确的？

召回率 (Recall)

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

反类	TN	FP
正类	FN	TP
	预测为反类	预测为正类

有多少正类被预测出来了?

FPR (False Positive Rate)

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

反类	TN	FP
正类	FN	TP
	预测为反类	预测为正类

有多少负类被预测出来了?

F1-Score

$$F = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

优点：结合了 precision 和 recall 两个指标

缺点：难以解释其数值的意义

使用 sklearn 计算评价指标

`sklearn.metrics.classification_report`

主要参数:

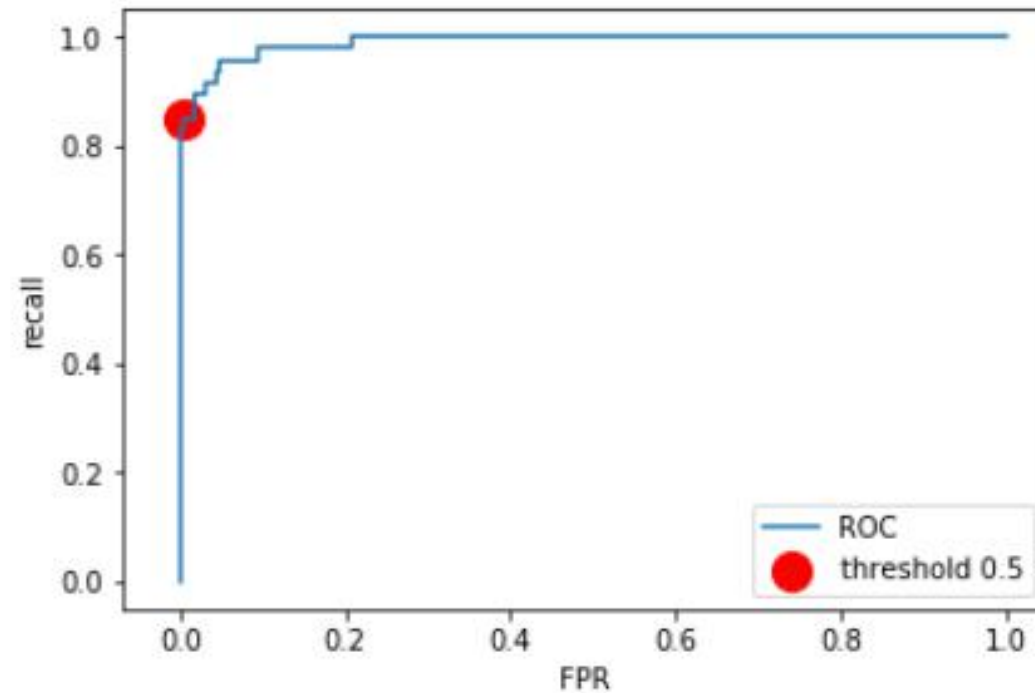
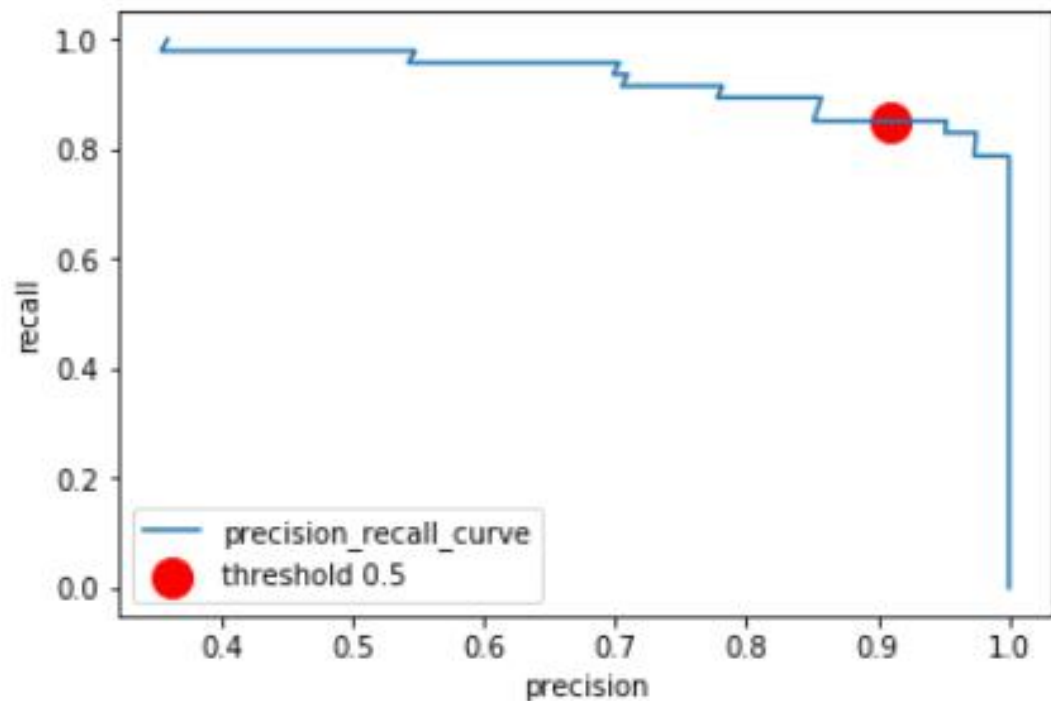
- `y_true`: 真实的 y 值
- `y_pred`: 预测的 y 值

分类阈值

默认情况下, 将 $y_prob > 0.5$ 或者 $y_decision > 0$ 的样本分类为**正类**

即 $y_pred = y_prob > 0.5$ or $y_pred = y_decision > 0$

准确率 - 召回率曲线 和 ROC 曲线



如何在两条曲线之间进行比较?



average precision score



AUC

使用 sklearn 生成曲线

```
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import average_precision_score, roc_curve, roc_auc_score

precision, recall, thresholds = precision_recall_curve(y,y_prob)
ap = average_precision_score(y,y_prob)
fpr, recall, thresholds = roc_curve(y,y_prob)
auc = roc_auc_score(y,y_prob)
```