

  
程序员外包平台

  
logo设计


  
永久虚拟主机

  
进销存erp软件


  
logo设计制作

  
人才招聘系统

# 多分类算法的评估指标

 taon 

关注

 0.78 2020.05.06 22:10:04 字数 1,055 阅读 2,248

在以往的分类问题求解当中，我们遇到的问题多为二分类问题，我们常用的评估指标有 accuracy, precision, recall\_score, f1-score, roc\_auc\_score等。但是在实际生活中，多分类问题也是大量存在的。这一小节，我们就详细的说明一下多分类问题的评估指标。



iris.png

我们先来看一下sklearn库中的二分类的评估指标，以recall\_score为例。在recall\_score方法中，有一个很重要的参数'average'，它的默认值为'binary'。当在默认参数的情况，该评估方法只能求解二分类问题，如果将该评估方法用于多分类问题，则系统会报错。但'average'同时也向我们提供了其他四个用于解决多分类的问题的参数'micro','macro','weighted','samples'。下面我们以鸢尾花数据集为例来对这四个参数进行逐一说明。

```
[sklearn.metrics.recall_score]('https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html')
```


```
1 | sklearn.metrics.recall_score(y_true, y_pred, labels=None,
2 | pos_label=1, average='binary', sample_weight=None, zero_division='warn')
```

我们以recall\_score的计算为例，recall\_score的计算公式如下：

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



著名律师排行榜

 taon 

关注

总资产13

图像数据处理  
阅读 256

词向量构造 - Tf-idf模型  
阅读 1,017

## 热门故事

- 凤凰男的醒悟：花了半辈子孝顺父母，换来妻离子散
- 扎心虐文：不是所有的女主最后都会选择原谅
- 闺蜜的一条朋友圈，结束了我和老公5年的婚姻
- 被戳穿的爱情是骗局，没被戳穿的变成了信仰

## 推荐阅读

- 十一、timeROC  
阅读 249
- 基于随机森林算法的心脏病人预测分类  
阅读 297
- 七、lasso回归  
阅读 299
- 【实战篇】集成算法建模（三）  
阅读 126
- 随机森林与神经网络联合诊断心衰模型的构建与分析  
阅读 1,036



值。

```
1 #导入数据分析的常用工具包
2 import numpy as np
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6 %matplotlib inline
7
8 #导入iris数据集
9 from sklearn import datasets
10 iris = datasets.load_iris()
11
12 #对iris数据集进行切分
13 from sklearn.model_selection import train_test_split
14 X = iris.data
15 Y = iris.target
16 X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size = 0.3,random_state = 0)
17
18 #建立一个基础的逻辑回归模型
19 from sklearn.linear_model import LogisticRegression
20 LR = LogisticRegression()
21 LR.fit(X_train,Y_train)
22 prediction = LR.predict(X_test)
23
24 #计算混淆矩阵
25 from sklearn.metrics import confusion_matrix
26 cnf = confusion_matrix(Y_test,prediction,labels = [0,1,2])
27 print(cnf)
28 [out]:
29 array([[16,  0,  0],
30        [ 0, 13,  5],
31        [ 0,  0, 11]], dtype=int64)
```



人脸识别算法

该分类问题的混淆矩阵如下，列为真实类别，行为预测类别：

class	0	1	2
0	16	0	0
1	0	13	5
2	0	0	11

混淆矩阵（confusion matrix）说明：

TP(True positive)：把正例正确地预测为了正例，如把类别0预测为0的个数有16个。  
FN(False negative)：把正例错误地预测为了负例，如把类别1预测为2的个数有5个。  
FP(False positive)：把负例错误地预测为了正例，假设0为正例，错误地把1,2预测为0就是FP。  
TN(True negative)：把负例正确地预测为了负例，假设0为正例，1,2为负例，正确地把1,2预测为1,2就是TN。  
对于混淆矩阵，可以这样理解。第一个字母T/F，表示预测的正确与否；第二个字母P/N，表示预测的结果为正例或者负例。如TP就表示预测对了，预测的结果是正例，那它的意思就是把正例预测为了正例。

1.Micro

Micro：把所有类汇总在一起计算出最终recall值，其计算公式如下：

$$Recall = \frac{TP_0 + TP_1 + TP_2}{TP_0 + FN_0 + TP_1 + FN_1 + TP_2 + FN_2}$$



在使用Micro参数时，其recall\_score = (16+13+11)/(16+13+11+5) = 0.89。

从上述计算结果可以看出，两者的计算结果是一致的，均为0.89。

## 2.Macro

Macro：分别计算出每一类的recall值，再取算数平均值，其计算公式如下：

$$Recall = \left( \frac{TP_0}{TP_0 + FN_0} + \frac{TP_1}{TP_1 + FN_1} + \frac{TP_2}{TP_2 + FN_2} \right) * \frac{1}{3}$$

在使用Macro参数时，其recall\_score = [16/(16+0+0) + 13/(0+13+5) + 11/(11+0+0)] \* 1/3 = 0.91

使用sklearn.metrics方法计算：

```
1 from sklearn.metrics import recall_score
2 recall_score(Y_test,prediction,average = 'macro')
3 [out]:0.9074074074074074
```

从上述计算结果可以看出，macro参数下的recall值为0.91。

## 3.Weighted

Weighted：分别计算每一类的recall值，再乘以各自的权重，然后求和，其计算公式如下：

$$Recall = \frac{TP_0}{TP_0 + FN_0} * W_0 + \frac{TP_1}{TP_1 + FN_1} * W_1 + \frac{TP_2}{TP_2 + FN_2} * W_2$$

计算各类的权重：

```
1 W0,W1,W2 = np.bincount(Y_test) / len(Y_test)
2 print(W0,W1,W2)
3 [out]: 0.3555555555555557 0.4 0.24444444444444444
```

在使用weighted参数时，其recall\_score = 16/(16+0+0) \* 0.356 + 13/(0+13+5) \* 0.4 + 11/(11+0+0) \* 0.244 = 0.89

使用sklearn.metrics方法计算：

```
1 from sklearn.metrics import recall_score
2 recall_score(Y_test,prediction,average = 'weighted')
3 [out]:0.8888888888888888
```

从上述计算结果可以看出，weighted参数下的recall值为0.89。

## 4.Samples

samples应用于多标签的分类问题，每一个样本拥有一个以上的标签。如一个感染病毒性肺炎的患者，就可以说他既属于病毒性肺炎患者这一类，也可以说他属于肺炎患者类。

小结：

1.对于多分类算法的评估，我们需要将sklearn.metrics.recall\_score中的'average'参数修改为'micro'或'macro'或者'weighted'。



写下你的评论...

评论0


赞7


文章参考：

[1].<https://zhuanlan.zhihu.com/p/59862986>

 7人点赞 >



 机器学习算法



更多精彩内容，就在简书APP



"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



taon

总资产13 共写了7.3W字 获得183个赞 共178个粉丝

关注

如何给自己的名字签名，点击查看详情！



写下你的评论...

全部评论 0 只看作者

按时间倒序 按时间正序

