

(原创) sklearn中 F1-micro 与 F1-macro区别和计算原理

最近在使用sklearn做分类时候，用到metrics中的评价函数，其中有一个非常重要的评价函数是F1值。（关于这个值的原理自行google或者百度）

在sklearn中的计算F1的函数为 f1_score ， 其中有一个参数average用来控制F1的计算方式，今天我们就说说当参数取micro和macro时候的区别

1、F1公式描述:

F1-score: $\frac{2*(P*R)}{(P+R)}$

准确率 (P) : TP/ (TP+FP)

召回率(R): TP/(TP + FN)

对于数据测试结果有下面4种情况:

真阳性 (TP) : 预测为正, 实际也为正

假阳性 (FP) : 预测为正, 实际为负

假阴性 (FN) : 预测为负, 实际为正

真阴性 (TN) : 预测为负, 实际也为负

2、f1_score中关于参数average的用法描述:

'micro': Calculate metrics globally by counting the total true positives, false negatives and false positives.

'micro':通过先计算总体的TP, FN和FP的数量, 再计算F1

'macro': Calculate metrics for each label, and find their unweighted mean. This does not take label imbalance into account.

'macro':分布计算每个类别的F1, 然后做平均 (各类别F1的权重相同)

3、初步理解

通过参数用法描述, 想必大家从字面层次也能理解他是什么意思, micro就是先计算所有的TP, FN , FP的个数后, 然后再利上文提到公式计算出F1

macro其实就是先计算出每个类别的F1值, 然后去平均, 比如下面多分类问题, 总共有1,2,3,4这4个类别, 我们可以先算出1的F1, 2的F1, 3的F1, 4的F1,然后再取平均 (F1+F2+F3+4) /4

y_true = [1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4]

y_pred = [1, 1, 1, 0, 0, 2, 2, 3, 3, 3, 3, 4, 3, 4, 3]

4、进一步理解

我们还是上面的例子为例说明sklearn中是如何计算micro 和 macro的:

micro计算原理

首先计算总TP值, 这个很好就算, 就是数一数上面有多少个类别被正确分类, 比如1这个类别有3个分正确, 2有2个, 3有2个, 4有1个, 那TP=3+2+2+1=8

其次计算总FP值, 简单的说就是不属于某一个类别的元数被分到这个类别的数量, 比如上面不属于4类的元素被分到4的有1个

y_true = [1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4]

y_pred = [1, 1, 1, 0, 0, 2, 2, 3, 3, 3, 3, 4, 3, 4, 3]

如果还比较迷糊, 我们在计算时候可以把4保留, 其他全改成0, 就可以更加清楚地看出4类别下面的FP数量了, 其实这个原理就是 One-vs-All (OvA), 把4看成正类, 其他看出负类

公告

昵称: tech-engin

园龄: 3年11个月

粉丝: 1

关注: 1

+加关注

2022年3月						
日	一	二	三	四	五	六
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

搜索

找找看

谷歌搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

我的标签

leetcode(3)

机器学习(3)

字符匹配(1)

KMP(1)

lightgbm(1)

macro(1)

micro(1)

f1_score(1)

sklearn(1)

xgboost(1)

更多

随笔档案

2019年1月(3)

2018年5月(1)

2018年4月(3)

阅读排行榜

- (原创) sklearn中 F1-micro 与 F1-macro区别和计算原理(36985)
- (原创) lightgbm 一些错误情况处理(7462)
- (原创) linux安装xgboost快速高效方法(4856)

4. 5. 最长回文子串（动态规划算法）(893)
5. 字符串匹配算法系列一：KMP算法原理(277)

y_true = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4]

y_pred = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 4, 0]

同理我们可以再计算FN的数量

	1类	2类	3类	4类	总数
TP	3	2	2	1	8
FP	0	0	3	1	4
FN	2	2	1	1	6

所以micro的 精确度P 为 TP/(TP+FP)=8/(8+4)=0.666 召回率R TP/(TP+FN)=8/(8+6)=0.571 所以F1-micro的值为： 0.6153

可以用sklearn来核对,把average设置成micro

```
y_true = [1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4]
y_pred = [1, 1, 1, 0, 0, 2, 2, 3, 3, 3, 4, 3, 4, 3]
print(f1_score(y_true,y_pred,labels=[1,2,3,4],average='micro'))
#>>> 0.615384615385
```

计算macro

macro先要计算每一个类的F1, 有了上面那个表, 计算各个类的F1就很容易了, 比如1类, 它的精确率P=3/(3+0)=1 召回率R=3/(3+2)=0.6 F1=2*(1*0.5)/1.5=0.75

可以sklearn, 来计算核对,把average设置成macro

```
#average=None,取出每一类的P,R,F1值
p_class, r_class, f_class, support_micro=precision_recall_fscore_support(y_true=y_true, y_pred=y_pred, labels=[1, 2, 3, 4], average=None)
print('各类单独F1:', f_class)
print('各类F1取平均: ', f_class.mean())
print(f1_score(y_true,y_pred,labels=[1,2,3,4],average='macro'))
#>>>各类单独F1: [ 0.75          0.66666667  0.5          0.5          ]
#>>>各类F1取平均: 0.604166666667
#>>>0.604166666667
```

如有转载, 请注明出处, 谢谢

标签: sklearn, f1_score, micro, macro

好文要顶

关注我

收藏该文

tech-engin

关注 + 1

粉丝 + 1

±加关注

« 上一篇: [\(原创\) linux安装xgboost快速高效方法](#)

» 下一篇: [\(原创\) lightgbm 一些错误情况处理](#)

评论排行榜

1. (原创) sklearn中 F1-micro 与 F1-macro区别和计算原理(2)

推荐排行榜

1. (原创) sklearn中 F1-micro 与 F1-macro区别和计算原理(3)

最新评论

1. Re: (原创) sklearn中 F1-micro 与 F1-macro区别和计算原理

@ Songyulong貌似只有错误分类的情况下才会出现这几个不等, 其他情况, micro *** 和acc全部是相等的。可以看看我的总结: ...

--gongel

2. Re: (原创) sklearn中 F1-micro 与 F1-macro区别和计算原理

博主demo里通过四分类型计算micro怎么出现了0类呢?假如0换成任意一类P和R的值就相同了,microp,micror,micorF1就都相同了

--Songyulong

登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) 博客园首页

【推荐】 阿里云采购季-热卖返场优惠享不停, 云产品新老同享1元起

【推荐】 百度智能云2022开年大促0.4折起, 企业新用户享高配优惠

【推荐】 华为开发者专区, 与开发者一起构建万物互联的智能世界

编辑推荐:

- [\[ASP.NET Core\] MVC 模型绑定: 非规范正文内容的处理](#)
- [全面认识数据指标体系](#)
- [项目开发中对成长的一些思考](#)
- [专车架构进化往事: 好的架构是进化来的, 不是设计来的](#)
- [一个故事看懂 CPU 的 SIMD 技术](#)

最新新闻:

- [14人开发, 采用自研引擎, 西山居国风城建游戏惊艳亮相!](#)
 - [蚂蚁集团推出可信密态计算技术, 可支撑隐私计算落地大规模数据场景](#)
 - [Debian 允许不记名投票](#)
 - [科学家模拟 140 亿年前的宇宙](#)
 - [GCC 编译器接受了龙芯移植版本](#)
- » [更多新闻...](#)