评估机器学习算法的指标

原创 Sophie Li 大数据应用 5月30日

今日份知识你摄入了么?

```
48000/48000 [==============================] - 55s 1ms/step - loss: 0.0375 - acc: 0.8041 - val_loss: 0.0241 - val_acc: 0.8078
Epoch 2/50
48000/48000 [==============================] - 49s 1ms/step - loss: 0.0219 - acc: 0.8085 - val_loss: 0.0207 - val_acc: 0.8088
Epoch 3/50
48000/48000 [==============================] - 59s 1ms/step - loss: 0.0196 - acc: 0.8096 - val_loss: 0.0190 - val_acc: 0.8113
Epoch 4/50
48000/48000 [==============================] - 65s 1ms/step - loss: 0.0185 - acc: 0.8101 - val_loss: 0.0181 - val_acc: 0.8106
Epoch 5/50
48000/48000 [==============================] - 71s 1ms/step - loss: 0.0177 - acc: 0.8105 - val_loss: 0.0174 - val_acc: 0.8113
Epoch 6/50
48000/48000 [==============================] - 73s 2ms/step - loss: 0.0171 - acc: 0.8108 - val_loss: 0.0170 - val_acc: 0.8122
Epoch 7/50
48000/48000 [==============================] - 73s 2ms/step - loss: 0.0167 - acc: 0.8110 - val_loss: 0.0166 - val_acc: 0.8120
Epoch 8/50
48000/48000 [==============================] - 74s 2ms/step - loss: 0.0163 - acc: 0.8111 - val_loss: 0.0162 - val_acc: 0.8121
Epoch 9/50
48000/48000 [==============================] - 80s 2ms/step - loss: 0.0161 - acc: 0.8113 - val_loss: 0.0160 - val_acc: 0.8119
Epoch 10/50
48000/48000 [==============================] - 76s 2ms/step - loss: 0.0158 - acc: 0.8114 - val_loss: 0.0158 - val_acc: 0.8128
Epoch 11/50
48000/48000 [==============================] - 80s 2ms/step - loss: 0.0156 - acc: 0.8115 - val_loss: 0.0156 - val_acc: 0.8129
Epoch 12/50
48000/48000 [==============================] - 79s 2ms/step - loss: 0.0153 - acc: 0.8116 - val_loss: 0.0153 - val_acc: 0.8127
Epoch 13/50
48000/48000 [==============================] - 86s 2ms/step - loss: 0.0151 - acc: 0.8117 - val_loss: 0.0152 - val_acc: 0.8120
Epoch 14/50
48000/48000 [==============================] - 89s 2ms/step - loss: 0.0150 - acc: 0.8118 - val_loss: 0.0151 - val_acc: 0.8120
Epoch 15/50
48000/48000 [==============================] - 86s 2ms/step - loss: 0.0148 - acc: 0.8118 - val_loss: 0.0148 - val_acc: 0.8130
Epoch 16/50
48000/48000 [==============================] - 94s 2ms/step - loss: 0.0147 - acc: 0.8119 - val_loss: 0.0147 - val_acc: 0.8130
Epoch 17/50
48000/48000 [==============================] - 98s 2ms/step - loss: 0.0145 - acc: 0.8120 - val_loss: 0.0145 - val_acc: 0.8129
```

评估你的机器学习算法是任何项目的重要部分。当使用指标(如accuracy_score)进行评估时，您的模型可能会给出令人满意的结果，但是当与其他指标，如logarithmic_loss或任何其他此类度量进行评估时，模型可能会给出糟糕的结果。大多数情况下，我们使用分类精度来衡量我们的模型的性能，但这并不足以真正判断我们的模型。**在这篇文章中，我们将介绍不同类型的可用评估指标。**

分类精度

当我们使用准确性这个词的时候，我们通常指的是分类的准确性。它是正确预测的数量与输入样本的总数之比。

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

只有当每个类的样本数量相等时，这种方法才有效。
例如，假设我们的训练集中有98%的A类样本和2%的B类样本，那么我们的模型通过简单地预测每个属于A类的训练样本就可以很容易地获得98%的训练精度。
当同一模型在a类样本占60%，B类样本占40%的测试集上进行测试时，测试精度会下降到60%。分类的准确度是很高的，但给我们带来了获得高准确度的错觉。
当小类样本的误分类代价很高时，真正的问题就出现了。如果我们面对的是一种罕见但致命的疾病，诊断一个病人的疾病失败的代价要远远高于让一个健康的人去做更多检查的代价。

对数损失

对数损失，通过处理错误的分类工作。它对于多类分类很有效。当处理对数损失时，分类器必须为所有样本的每个类分配概率。假设有N个样本，属于M个类，那么Log Loss计算如下:

$$LogarithmicLoss = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} * \log(p_{ij})$$

当y_ij表示样本i是否属于类j，p_ij表示样本i属于类j的概率。
对数损失没有上界，它存在于[0，∞）范围内。接近于0的对数损失表示更高的精度，而如果日志损失远离0，则表示更低的精度。一般来说，最小化对数损失可以提高分类器的准确性。

混淆矩阵

顾名思义，混淆矩阵给出了一个矩阵作为输出，并描述了模型的完整性能。

假设我们有一个二元分类问题。我们有一些样本属于两个类别：是或否。此外，我们有自己的分类

器，它可以为给定的输入样本预测一个类。在165个样本上测试我们的模型，我们得到以下结果。

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

混淆矩阵有4个重要术语：

- 真阳性：在我们预测为“是”的情况下，实际输出也是“是”。
- 真阴性：在我们预测结果为“否”而实际输出为“否”的情况下。
- 假阳性：在我们预测为“是”而实际输出为“否”的情况下。
- 假阴性：在我们预测为“否”而实际输出为“是”的情况下。

矩阵的精度可以通过计算位于“主对角线”上的值的平均值来计算。

$$Accuracy = \frac{True Positives + False Negatives}{Total Number of Samples}$$

$$\therefore Accuracy = \frac{100 + 50}{165} = 0.91$$

混淆矩阵构成了其他类型度量的基础。

/// 曲线下的面积 ///

曲线下面积(AUC)是应用最广泛的评价指标之一。它是用来解决二元分类问题的。分类器的AUC等于分类器对随机选择的正样本的排序高于随机选择的反样本的排序的概率。在定义AUC之前，让我们先了解两个基本术语：

- 真阳性率(灵敏度): 真阳性率定义为TP/ (FN+TP)。真实阳性率对应于所有正数据点被正确认为为正的正数据点的比例。

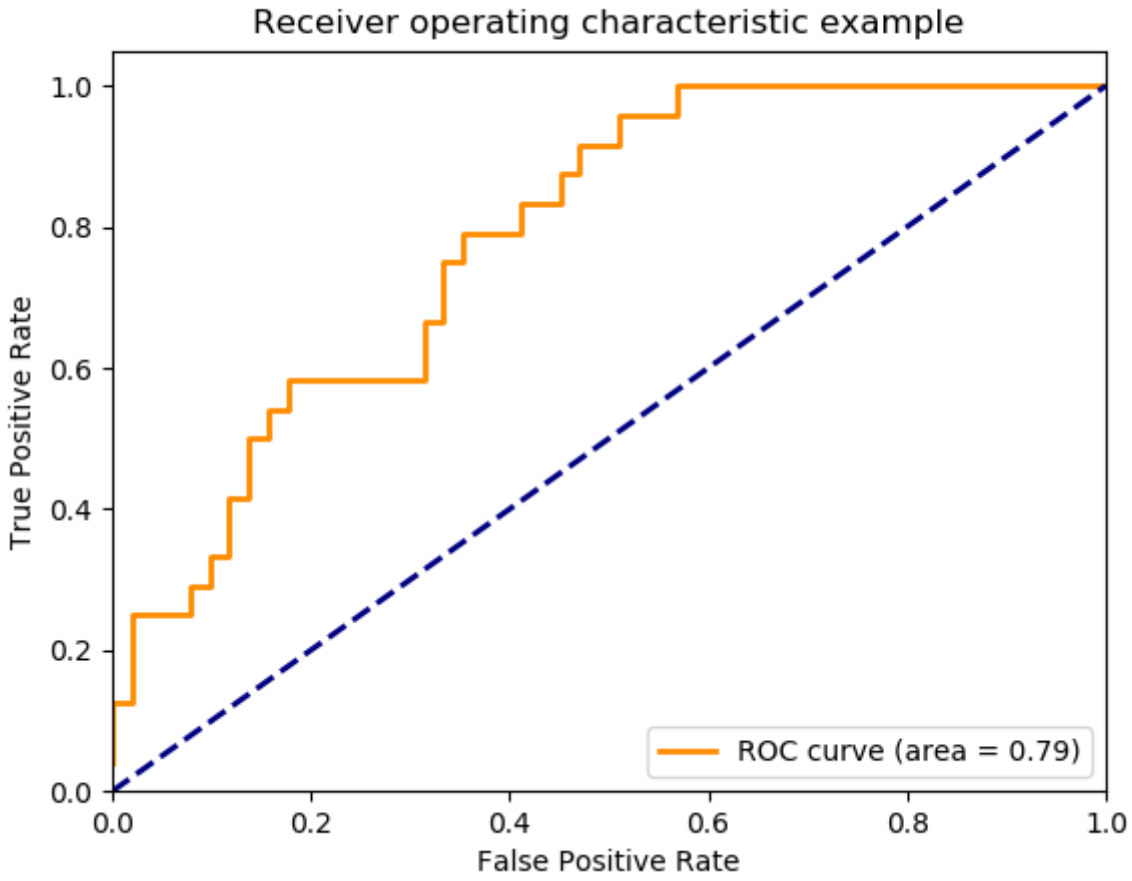
$$True Positive Rate = \frac{True Positive}{False Negative + True Positive}$$

- 假阳性率(特异度): 假阳性率定义为FP / (FP+TN)。假阳性率对应误认为为阳性的负数据点占所有负数据点的比例。

$$False Positive Rate = \frac{False Positive}{False Positive + True Negative}$$

假阳性率和真阳性率的值都在[0,1]范围内。FPR和TPR 都是在计算阈值(0.00,0.02,0.04，...

(1.00)，然后画一个图。AUC是图中假阳性率与真阳性率在[0,1]不同点曲线下的面积。



AUC的取值范围为[0,1]。值越大，我们的模型的性能就越好。

/// F1分数 ///

F1分数用来衡量测试的准确性。

F1得分是精确率和召回率之间的调和平均值。F1分数的范围是[0,1]。它告诉您您的分类器有多精确(它正确地分类了多少实例)，以及它有多健壮(它不会遗漏大量实例)。

高精确率但较低的召回率，给您一个非常精确的结果，但是它会遗漏大量难以分类的实例。F1分数越大，我们的模型的性能就越好。数学上可以表示为：

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

F1 分数试图找到精确度和召回率之间的平衡。

- 精度：正确的正面结果数除以分类器预测的正面结果数。

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

- 召回率：它是正确阳性结果的数量除以所有相关样本的数量(所有应该被确定为阳性的样本)。

$$Precision = \frac{TruePositives}{TruePositives + FalseNegatives}$$

$$TruePositives + FalseNegatives$$

平均绝对误差 （MAE）

平均绝对误差是原始值与预测值之差的平均值。它给出了预测与实际产出之间的距离。然而，它们并没有给我们任何关于误差方向的概念，例如，我们是在预测数据不足还是在预测数据过量。数学上表示为：

$$MeanAbsoluteError = \frac{1}{N}\sum_{j=1}^{N}|y_j - \hat{y}_j|$$

均方误差 (MSE)

均方误差(MSE)与平均绝对误差非常相似，唯一的区别是MSE取原始值与预测值之差的平方的平均值。MSE的优点是它更容易计算梯度，而平均绝对误差需要复杂的线性规划工具来计算梯度。当我们取误差的平方时，较大误差的影响比较小误差更明显，因此模型现在可以更多地关注较大误差。

$$MeanSquaredError = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2$$

就是这样。

原文作者：Aditya Mishra
翻译作者：Sophie Li
美工编辑：过儿
校对审稿：Dongdong
原文链接：https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

往期精彩回顾

用无监督学习生成吊炸天Spotify播放列表
6种常见处理Missing Value的方法
北美疫情家庭办公物品采购大数据
重塑世界秩序：冠状病毒后的7种预测
用Python (scikit-learn) 做PCA分析