编 程 学 习 笔 记

机器学习模型评估指标

Code

作 者 姓 名： 颜佳

目 录

[1 回归算法指标 1](#_Toc104038615)

[1.1 平均绝对误差（Mean Absolute Error） 1](#_Toc104038616)

[1.2 均方误差（Mean Squared Error） 1](#_Toc104038617)

[1.3 均方根误差（Root Mean Squared Error） 1](#_Toc104038618)

[1.4 决定系数（Coefficient of determination） 1](#_Toc104038619)

[2 分类算法指标 1](#_Toc104038620)

[2.1 混淆矩阵（Confusion Matrix） 1](#_Toc104038621)

[2.2 准确率（Accuracy） 1](#_Toc104038622)

[2.3 错误率（Error Rate） 1](#_Toc104038623)

[2.4 精确率（Precision，查准率） 1](#_Toc104038624)

[2.5 召回率（Recall，查全率） 2](#_Toc104038625)

[2.6 F Score 2](#_Toc104038626)

[2.7 AUC（Area Under Curve） 2](#_Toc104038627)

[2.8 KS Kolmogorov-Smirnov 2](#_Toc104038628)

# 分类算法指标

## 混淆矩阵（Confusion Matrix）

True Positive（真正，TP）：将正类预测为正类数;

True Negative（真负，TN）：将负类预测为负类数;

False Positive（假正，FP）：将负类预测为正类数 (Type I error);

False Negative（假负，FN）：将正类预测为负类数 (Type II error)

|  |  |  |  |
| --- | --- | --- | --- |
| 混淆矩阵 | | 实际值 | |
| P | N |
| 预测值 | P | TP | FP |
| N | FN | TN |

## 准确率（Accuracy）

被分对的样本数除以所有的样本数



## 错误率（Error Rate）



## 精确率（Precision，查准率）

被预测为正例的正样本占所有预测为正例的比例。



## 召回率（Recall，查全率）

被预测为正例的样本占实际为正例的比例。



## F Score

P和R指标有时候会出现的矛盾的情况，这样就需要综合考虑他们，最常见的方法就是F-Measure（又称为F-Score）。



当时，又称为F1-Score

## AUC（Area Under Curve of ROC）

ROC（Receiver Operating Characteristic，接收者操作特征曲线）曲线和AUC（Area Under Curve）常被用来评价一个二值分类器（binary classifier）的优劣。

ROC曲线横坐标是FPR（False Positive Rate），纵坐标是TPR（True Positive Rate）





### 四个特殊点

（0, 1）表示且，表示所有的样本都被正确分类；

（1, 0）表示且，表示所有的样本都被错误分类；

（0, 0）表示且，表示所有的样本都被预测为负样本；

（1, 1）表示且，表示所有的样本都被预测为正样本；

### ROC曲线的画法

ROC曲线是由一系列因区分阈值变化产生的点，用于描述二分类模型的判断能力。对于一个特定的数据集，一个二分分类器的输出结果是样本属于正样本的概率。一般而言，把预测概率大于0.5的样本判定为正样本，把预测概率小于0.5的样本判定为负样本。如果判定为正样本的阈值取不同的值，则会产生一系列的（FPR，TPR）点，用平滑的曲线连接起来，就成为ROC曲线。

按照每个测试样本预测属于正样本的概率值从大到小排序，再把每一个概率值作为阈值。当样本的预测概率值大于阈值时，判定为正样本，当样本的预测概率值小于阈值时，判定为负样本，计算出（FPR，TPR）点。

## KS Kolmogorov-Smirnov

# 回归算法指标

## 平均绝对误差（Mean Absolute Error）

## 均方误差（Mean Squared Error）

## 均方根误差（Root Mean Squared Error）

## 决定系数（Coefficient of determination）

# TensorFlow中的损失函数

## 分类问题

### tf.nn.sigmoid\_cross\_entropy\_with\_logits(labels=None, logits=None, name=None)

先计算sigmoid输出，再求交叉熵，适合二分类问题。

令x = logits，z = labels，



为了防止溢出，实际计算公式为：



### tf.nn.softmax\_cross\_entropy\_with\_logits(labels, logits, axis=-1, name=None)

计算logits和labels之间的softmax交叉熵，适合多分类问题。

labels的形状为[batch\_size, num\_classes]，且每一行都应该符合概率分布（即每一行的每个元素都应该在0和1之间，且每一行的和为1）

### tf.nn.sparse\_softmax\_cross\_entropy\_with\_logits(labels, logits, name=None)

计算logits和labels之间的稀疏softmax交叉熵。

softmax\_cross\_entropy\_with\_logits传入的labels为稀疏标签，如one-hot标签[[0,0,1], [0,1,0]]；

sparse\_softmax\_cross\_entropy\_with\_logits传入的labels为非稀疏标签，如三分类问题传入labels值为[2,1]，其中[2,1]中的2表示属于第3类，对应one-hot标签里的[0,0,1]，[2,1]中的1表示属于第2类，对应one-hot标签里的[0,1,0]。

### tf.nn.weighted\_cross\_entropy\_with\_logits(labels, logits, pos\_weight, name=None)

带权重的sigmoid交叉熵，适用于正、负样本数量差距过大的情形。通过增加一个权重的系数，来平衡正、负样本差距，可在一定程度上解决差距过大时训练结果严重偏向大样本的情况。

当pos\_weight > 1时，会减少FN计数，从而增加召回率。当pos\_weight < 1时，会减少FP计数并提高精度。

令x = logits, z = labels, q = pos\_weight



为了防止溢出，实际计算公式为：



## 回归问题

### tf.losses.mean\_squared\_error：均方根误差（MSE） —— 回归问题中最常用的损失函数

优点：便于梯度下降，误差大时下降快，误差小时下降慢，有利于函数收敛。

缺点：受明显偏离正常范围的离群样本的影响较大

### tf.losses.absolute\_difference：平均绝对误差（MAE） —— 想格外增强对离群样本的健壮性时使用

优点是其克服了 MSE 的缺点，受偏离正常范围的离群样本影响较小。

缺点是收敛速度比 MSE 慢，因为当误差大或小时其都保持同等速度下降，而且在某一点处还不可导，计算机求导比较困难。

### tf.losses.huber\_loss：Huber loss —— 集合 MSE 和 MAE 的优点，但是需要手动调超参数

核心思想是，检测真实值（y\_true）和预测值（y\_pred）之差的绝对值在超参数 δ 内时，使用 MSE 来计算 loss, 在 δ 外时使用类 MAE 计算 loss。sklearn 关于 huber 回归的文档中建议将 δ=1.35 以达到 95% 的有效性。

# logits

在深度学习中，logits是指最终的全连接层的输出。通常神经网络中都是先有logits，而后通过sigmoid函数或者softmax函数得到概率p。