

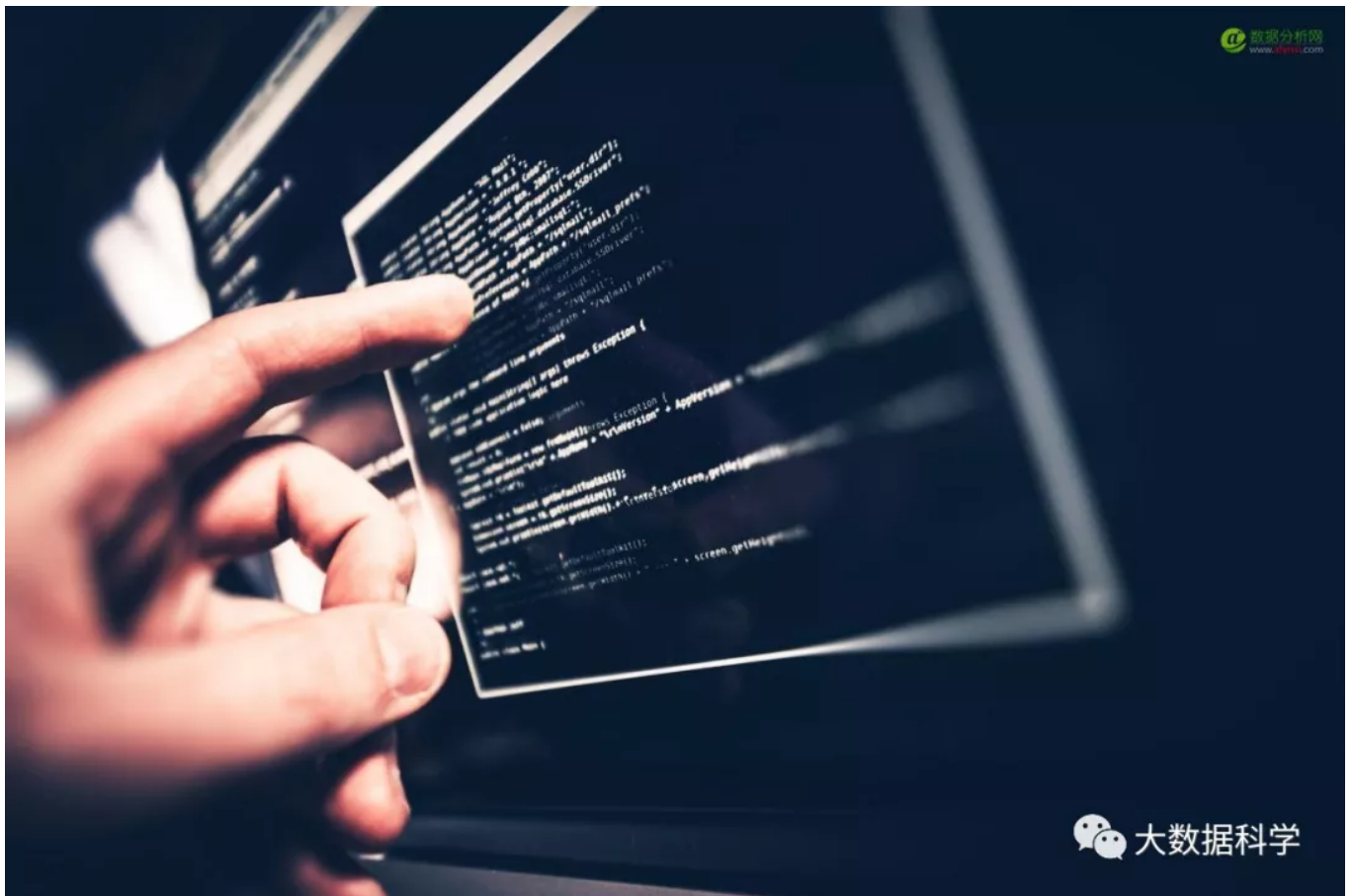
11个机器学习的模型评估指标

大数据科学 2019-08-16



概观

- 评估模型是构建有效机器学习模型的核心部分
- 有几种评估指标，如混淆矩阵，交叉验证，AUC-ROC曲线等。
- 不同的评估指标用于不同类型的问题



介绍

构建机器学习模型的想法是建设性的反馈原则。您可以构建模型，从指标获取反馈，进行改进并继续，直到达到理想的准确度。评估指标解释了模型的性能。评估指标的一个重要方面

是它们区分模型结果的能力。

我见过很多分析师和有抱负的数据科学家甚至都不愿意去检查他们模型的稳健性。一旦他们完成了模型的建立，他们就会急忙将预测值映射到看不见的数据上。这是一种不正确的方法。

简单地构建预测模型不是您的动机。它是关于创建和选择一个模型，该模型可以提供样本数据的高精度。因此，在计算预测值之前检查模型的准确性至关重要。

在我们的行业中，我们会考虑不同类型的指标来评估我们的模型。度量的选择完全取决于模型的类型和模型的实现计划。

完成模型构建后，这11个指标将帮助您评估模型的准确性。考虑到交叉验证的日益普及和重要性，我在本文中也提到了它的原理。

目录

1. 混乱矩阵
2. F1得分
3. 增益和提升图表
4. Kolmogorov Smirnov图表
5. AUC – ROC
6. 记录丢失
7. 基尼系数
8. 协调 – 不和谐比率
9. 均方根误差
10. 交叉验证（虽然不是指标！）

预热：预测模型的类型

当我们谈论预测模型时，我们谈论的是回归模型（连续输出）或分类模型（标称或二进制输出）。每种模型中使用的评估指标都不同。

在分类问题中，我们使用两种类型的算法（取决于它创建的输出类型）：

1. **类输出**：SVM和KNN等算法创建类输出。例如，在二进制分类问题中，输出将是0或1。但是，今天我们有可以将这些类输出转换为概率的算法。但是统计界并没有很好地接受这些算法。
2. **概率输出**：Logistic回归，随机森林，梯度提升，Adaboost等算法给出概率输出。将概率输出转换为类输出只是创建阈值概率的问题。

在回归问题中，我们在输出中没有这种不一致性。输出本质上是连续的，不需要进一步处理。

说明性示例

对于分类模型评估度量讨论，我使用了我对Kaggle问题BCI挑战的预测。问题的解决方案超出了我们在此讨论的范围。但是，本文使用了训练集的最终预测。对此问题的预测是概率输出，假设阈值为0.5，已将其转换为类输出。

1.混淆矩阵

混淆矩阵是 $N \times N$ 矩阵，其中 N 是预测的类的数量。对于手头的问题，我们有 $N = 2$ ，因此我们得到一个 2×2 矩阵。以下是一些定义，您需要记住混淆矩阵：

- **准确性**：正确的预测总数的比例。
- **阳性预测值或精确度**：正确识别的阳性病例的比例。
- **负面预测值**：正确识别的负面案例的比例。
- **敏感度或召回率**：正确识别的实际阳性病例的比例。
- **特异性**：正确识别的实际阴性病例的比例。

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	Positive Predictive Value	$a/(a+b)$
	Negative	c	d	Negative Predictive Value	$d/(c+d)$
		Sensitivity	Specificity	Accuracy = $(a+d)/(a+b+c+d)$	
		$a/(a+c)$	$d/(b+d)$		

Count of ID	Target			
Model	1	0	Grand Total	
1	3,834	639	4,473	85.7%
0	16	951	967	1.7%
Grand Total	3,850	1,590	5,440	
	99.6%	40.19%		

手头问题的准确率达到88%。从上面两个表中可以看出，阳性预测值很高，但阴性预测值很低。同样适用于灵敏度和特异性。这主要是由我们选择的阈值驱动的。如果我们降低我们的阈值，两对截然不同的数字将更接近。

通常，我们关注上面定义的度量标准之一。例如，在一家制药公司，他们会更关心最小的错误阳性诊断。因此，他们会更关注高特异性。另一方面，消耗模型将更关注灵敏度。混淆矩阵通常仅用于类输出模型。

2. F1得分

在上一节中，我们讨论了分类问题的精确度和召回率，并强调了在我们的用例中选择精确度/召回率的重要性。如果对于一个用例，我们试图获得最佳精度并同时召回？F1-Score是分类问题的精度和召回值的调和平均值。

现在，一个显而易见的问题是为什么采用调和均值而不是算术平均值。这是因为HM更多地惩罚极端值。让我们通过一个例子来理解这一点。我们有一个二元分类模型，结果如下：

精度：0，召回：1

在这里，如果我们采用算术平均值，我们得到0.5。很明显，上面的结果来自一个哑巴分类器，它只是忽略输入而只是预测其中一个类作为输出。现在，如果我们要使用HM，我们将得到0这是准确的，因为这个模型对于所有目的都是无用的。

这看起来很简单。然而，有些情况下，数据科学家希望给予精确度或召回率更高的重要性/权重百分比。为了这个目的，改变上面的表达式，我们可以包含一个可调参数beta。

Fbeta测量模型的有效性，该模型相对于精确度为 β 次的重要性的用户。

3.增益和提升图表

增益和提升图主要用于检查概率的等级排序。以下是构建提升/增益图表的步骤：

第1步：计算每次观察的概率

第2步：按降序排列这些概率。

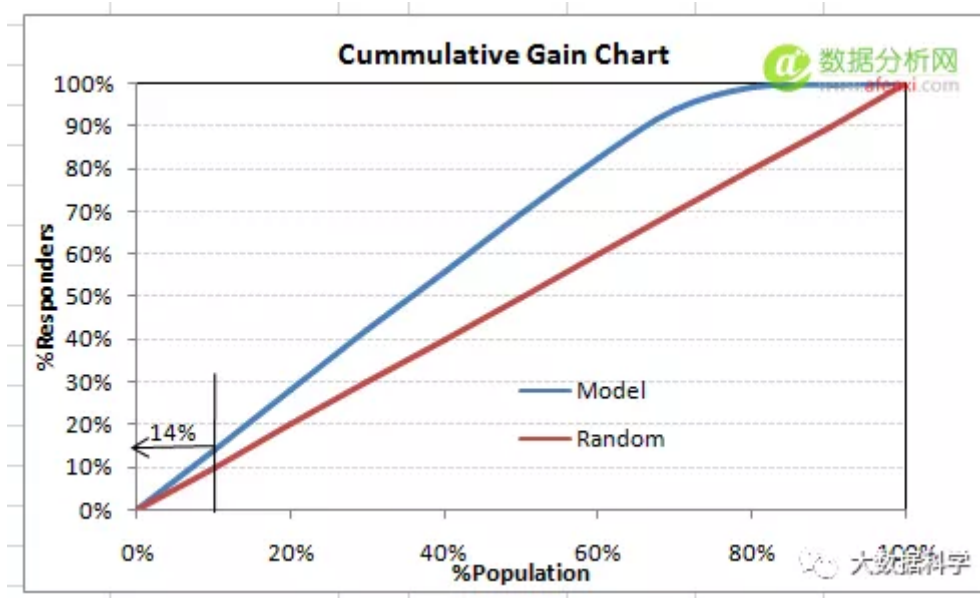
步骤3：构建十分位数，每组具有几乎10%的观察结果。

步骤4：计算Good（响应者），Bad（非响应者）和总数的每个十分位数的响应率。

您将获得下表，您需要从中绘制增益/提升图表：

Lift/Gain	Column Labels			%Rights	%Wrongs	%Population	Cum %Right	Cum %Pop	Lift @decile	Total Lift
Row Labels	0	1	Grand Total							
1	543	543	543	14%	0%	10%	14%	10%	141%	141%
2	2	542	544	14%	0%	10%	28%	20%	141%	141%
3	7	537	544	14%	0%	10%	42%	30%	139%	141%
4	15	529	544	14%	1%	10%	56%	40%	137%	140%
5	20	524	544	14%	1%	10%	69%	50%	136%	139%
6	42	502	544	13%	3%	10%	83%	60%	130%	138%
7	104	440	544	11%	7%	10%	94%	70%	114%	134%
8	345	199	544	5%	22%	10%	99%	80%	52%	124%
9	515	29	544	1%	32%	10%	100%	90%	8%	111%
10	540	5	545	0%	34%	10%	100%	100%	10%	10%
Grand Total	1590	3850	5440							

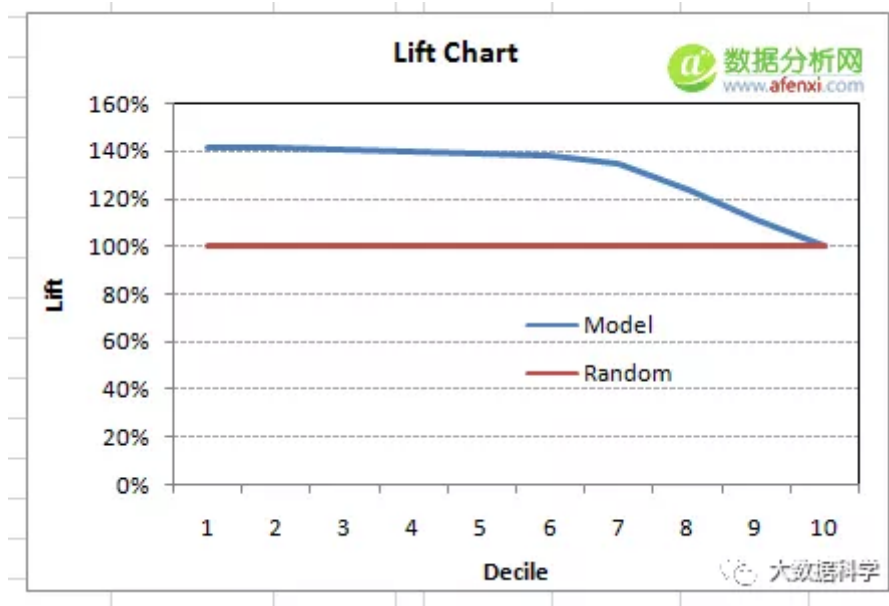
这是一个非常有益的表。累积增益图表是累积%权利和累计%人口之间的图表。对于手头的情况，这里是图：



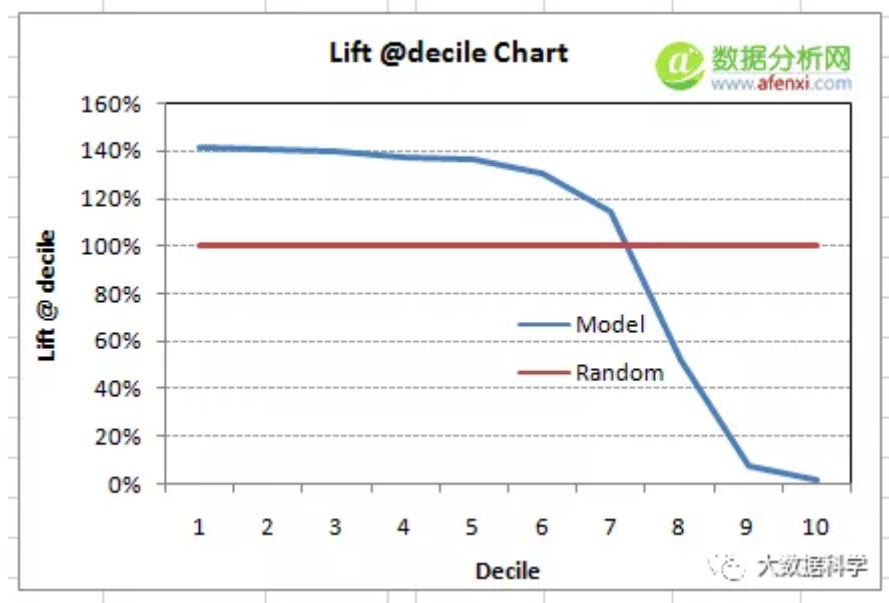
该图表告诉您模型将响应者与非响应者隔离的程度。例如，第一个十分位数有10%的人口，有14%的响应者。这意味着我们在第一个十分位数时有140%的升力。

在第一个十分位数中我们可以达到的最大升力是多少？从本文的第一个表中，我们知道响应者的总数是3850.第一个十分位数将包含543个观察值。因此，第一个十分位数的最大升力可能是 $543 / 3850 \sim 14.1\%$ 。因此，我们对此模型非常接近完美。

现在让我们绘制升力曲线。提升曲线是总升力与%人口之间的关系曲线。请注意，对于随机模型，它始终保持100%不变。以下是手头案例的情节：



您还可以使用十分位数绘制十分位明智的提升：



这个图告诉你什么？它告诉你我们的模型一直运行到第7个十分位数。每个十分位数都会向非响应者倾斜。任何具有提升@十分位数超过100%直到最小第三十分位数和最大七分位数的模型都是一个很好的模型。否则，您可能会先考虑过采样。

提升/增益图表广泛用于广告系列定位问题。这告诉我们，直到哪个十分位数可以针对特定广告系列的客户。此外，它会告诉您对新目标库的期望响应量。

4. Kolomogorov Smirnov图表

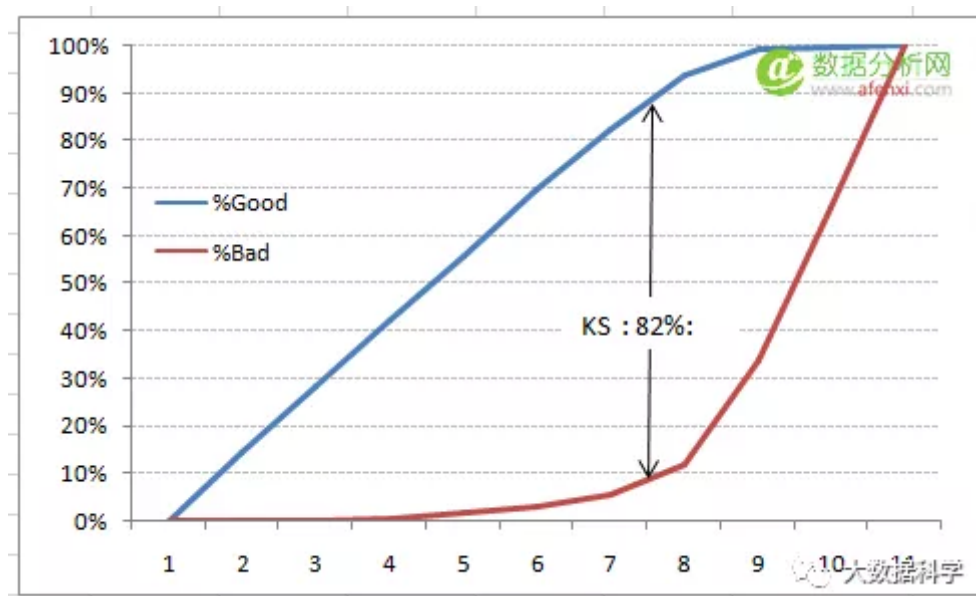
KS或Kolmogorov-Smirnov图表测量分类模型的性能。更准确地说，KS是衡量正负分布之间分离程度的指标。如果分数将人口分成两个独立的组，其中一组包含所有正数而另一组包含所有负数，则KS为100。

另一方面，如果模型不能区分正面和负面，那么就好像模型从总体中随机选择案例。KS将为0。在大多数分类模型中，KS将介于0和100之间，并且值越高，模型在分离正面和负面情况时越好。

对于手头的情况，以下是表格：

				Cummulative		K-S
Lift/Gain	Column			%Rights	%Wrongs	
Row La	0	1	Grand Tot			
1		543	543	0%	0%	0%
2	2	542	544	14%	0%	14%
3	7	537	544	14%	0%	28%
4	15	529	544	14%	1%	42%
5	20	524	544	14%	1%	54%
6	42	502	544	13%	3%	67%
7	104	440	544	11%	7%	77%
8	345	199	544	5%	22%	82% K-S
9	515	29	544	1%	32%	65%
10	540	5	545	0%	34%	34%
Grand Tot	1590	3850	5440			0%

我们还可以绘制%Cumulative Good和Bad来查看最大分离。以下是一个示例图：



到目前为止所涵盖的指标主要用于分类问题。直到这里，我们了解了混淆矩阵，升力和增益图表以及kolmogorov-smirnov图表。让我们继续学习一些更重要的指标。

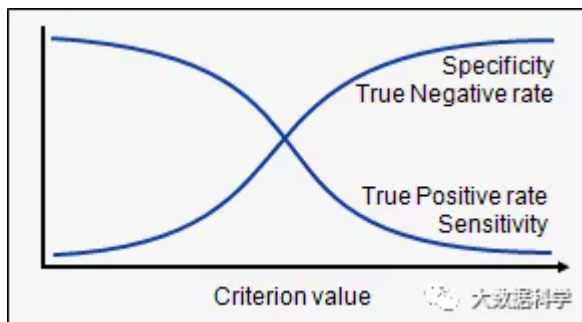
5. ROC曲线下面积 (AUC – ROC)

这又是业界常用的指标之一。使用ROC曲线的最大优点是它独立于响应者比例的变化。本声明将在以下部分中更加清晰。

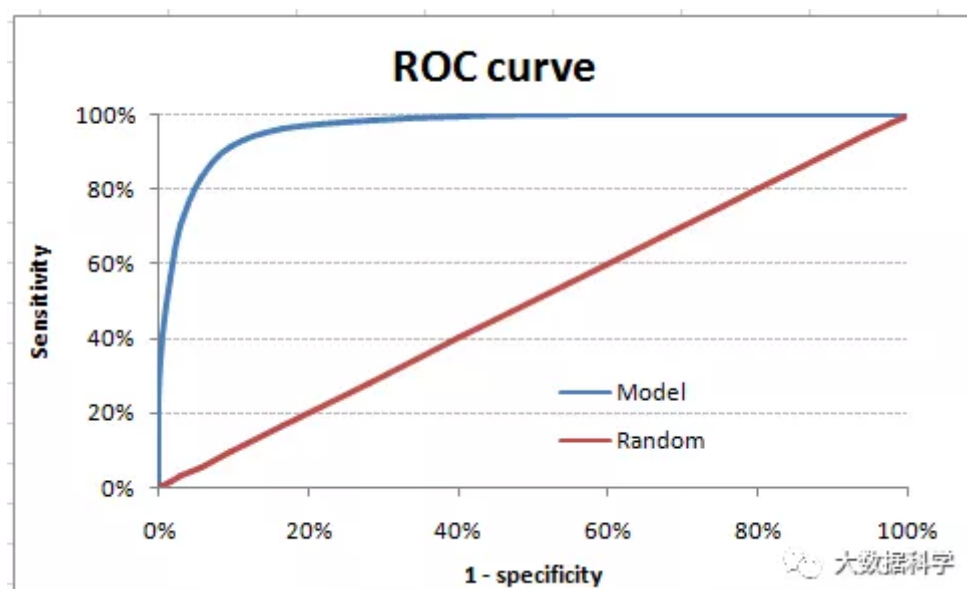
让我们首先尝试了解什么是ROC（接收器工作特性）曲线。如果我们看下面的混淆矩阵，我们观察到对于概率模型，我们得到每个度量的不同值。

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	Positive Predictive Value	$a/(a+b)$
	Negative	c	d	Negative Predictive Value	$d/(c+d)$
		Sensitivity	Specificity	Accuracy = $(a+d)/(a+b+c+d)$	
		$a/(a+c)$	$d/(b+d)$		

因此，对于每种敏感性，我们得到不同的特异性。两者的变化如下：



ROC曲线是灵敏度和（1-特异性）之间的曲线。（1-特异性）也称为假阳性率，灵敏度也称为真阳性率。以下是手头案例的ROC曲线。



让我们以阈值 = 0.5为例（参考混淆矩阵）。这是混淆矩阵：

Count of ID Target				
Model		1	0	Grand Total
1		3,834	639	4,473 85.7%
0		16	951	967 1.7%
Grand Total		3,850	1,590	5,440
		99.6%	40.19%	96.4%

如您所见，此阈值的灵敏度为99.6%，（1-特异性）为~60%。该坐标在我们的ROC曲线中成为点。为了将该曲线降低到单个数字，我们找到该曲线下的面积（AUC）。

注意，整个正方形的面积是 $1 * 1 = 1$ 。因此AUC本身是曲线下面的比率和总面积。对于手头的情况，我们将AUC ROC定为96.4%。以下是一些拇指规则：

- .90-.1 = 优秀 (A)
- .80-.90 = 好 (B)
- .70-.80 = 一般 (C)
- .60-.70 = 差 (D)
- .50-.60 = 失败 (F)

我们看到我们属于当前模型的优秀乐队。但这可能只是过于贴合。在这种情况下，及时和不合时宜的验证变得非常重要。

要记住的要点：

- 1.对于给出类作为输出的模型，将在ROC图中表示为单个点。
- 2.这些模型无法相互比较，因为需要对单个指标进行判断而不使用多个指标。例如，具有参数（0.2,0.8）的模型和具有参数（0.8,0.2）的模型可以来自相同的模型，因此不应直接比较这些度量。
- 3.在概率模型的情况下，我们有幸得到一个AUC-ROC的数字。但是，我们仍然需要查看整个曲线以做出决定性的决定。一个模型也可能在某些区域表现更好，而其他模型在其他区域表现更好。

使用ROC的优点

为什么要使用ROC而不是升力曲线等指标？

提升取决于人口的总回应率。因此，如果人口的响应率发生变化，同一模型将给出不同的升力图。这种关注的解决方案可以是真正的升力图（在每个十分位数处找到升力和完美模型升力的比率）。但这种比例很少对企业有意义。

另一方面，ROC曲线几乎与响应速率无关。这是因为它具有从混淆矩阵的柱状计算中出来的两个轴。在响应率变化的情况下，x轴和y轴的分子和分母将以类似的比例改变。

6.记录丢失

AUC ROC考虑用于确定模型性能的预测概率。然而，AUC ROC存在问题，它只考虑概率的顺序，因此没有考虑模型预测更可能为正的样本的更高概率的能力。在这种情况下，我们可以记录每个实例的校正预测概率的对数的负平均值。

- $p(y_i)$ 是预测的正类概率
- $1-p(y_i)$ 是预测的负类概率
- $y_i = 1$ 表示正类，0表示负类（实际值）

让我们计算几个随机值的对数损失，以得到上述数学函数的要点：

$$\text{Logloss}(1, 0.1) = 2.303$$

$$\text{Logloss}(1, 0.5) = 0.693$$

$$\text{Logloss}(1, 0.9) = 0.105$$

从向右平缓的向下斜率可以看出，随着预测概率的改善，对数损失逐渐下降。然而，在相反方向上移动时，当预测概率接近0时，对数损失会非常快速地增加。

因此，降低日志损失，更好的模型。但是，对于良好的日志丢失没有绝对的衡量标准，并且它取决于用例/应用程序。

尽管AUC是根据具有变化的判定阈值的二元分类计算的，但是对数损失实际上考虑了分类的“确定性”。

7.基尼系数

基尼系数有时用于分类问题。基尼系数可以从AUC ROC数得出。基尼系数只是ROC曲线与诊断线之间的面积与上述三角形的面积之比。以下是使用的公式：

$$\text{基尼} = 2 * \text{AUC} - 1$$

基尼系数高于60%是一个很好的模型。对于手头的案例，我们将基尼计为92.7%。

8.一致 – 不和谐的比例

对于任何分类预测问题，这也是最重要的指标之一。要理解这一点，我们假设我们有3名学生今年有可能通过。以下是我们的预测：

A – 0.9

B – 0.5

C – 0.3

现在想象一下。如果我们要从这三个学生那里取两对，我们会有多少对？我们将有3对：AB，BC，CA。现在，在年底结束后，我们看到A和C今年通过而B失败了。不，我们选择所有配对，我们将找到一个响应者和其他非响应者。我们有多少这样的配对？

我们两对AB和BC。现在对于2对中的每一对，一致对是响应者的概率高于非响应者的概率。而不和谐的对是反之亦然。如果两个概率相等，我们说它是平局。让我们看看在我们的案例中发生了什么：

AB – 一致

BC – 不和谐

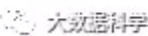
因此，在这个例子中我们有50%的一致案例。超过60%的一致比率被认为是一个很好的模型。在决定要定位的客户数量等时，通常不使用此度量标准。它主要用于访问模型的预测能力。对于像KS / Lift图表再次采用多少目标的决定。

9.均方根误差（RMSE）

RMSE是回归问题中最常用的评估指标。它遵循一个假设，即误差是无偏的并遵循正态分布。以下是RMSE需要考虑的要点：

1. “平方根”的功效使该指标能够显示大量偏差。
2. 此度量标准的“平方”特性有助于提供更强大的结果，从而防止取消正负误差值。换句话说，该度量恰当地显示了错误术语的合理幅度。
3. 它避免使用绝对误差值，这在数学计算中是非常不希望的。
4. 当我们有更多样本时，使用RMSE重建误差分布被认为更可靠。
5. RMSE受到异常值的影响很大。因此，请确保在使用此指标之前已从数据集中删除了异常值。
6. 与平均绝对误差相比，RMSE提供更高的权重并惩罚大的错误。

RMSE指标由下式给出：

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (Predicted_i - Actual_i)^2}{N}}$$


其中，N是观察总数。

10.均方根对数误差

在均方根对数误差的情况下，我们采用预测和实际值的对数。基本上，我们正在测量的方差有哪些变化。当我们不希望在预测值和真值都是巨大数字时惩罚预测值和实际值的巨大差异时，通常使用RMSLE。

1. 如果预测值和实际值都很小：RMSE和RMSLE相同。
2. 如果预测或实际值很大：RMSE > RMSLE
3. 如果预测值和实际值都很大：RMSE > RMSLE（RMSLE几乎可以忽略不计）

11. R平方/调整R平方

我们了解到，当RMSE降低时，模型的性能将会提高。但仅凭这些价值观并不直观。

在分类问题的情况下，如果模型的精度为0.8，我们可以衡量我们的模型对随机模型的有效性，随机模型的精度为0.5。因此随机模型可以作为基准。但是，当我们谈论的RMSE指标，我们不要有一个基准来比较。

换句话说，与一个非常简单的模型相比，我们的回归模型有多好，这个模型只是预测火车的目标平均值作为预测。

调整后的R-Squared

执行等于基线的模型将R-Squared设为0.更好的模型，更高的 r^2 值。具有所有正确预测的最佳模型将使R-Squared为1.然而，在向模型添加新特征时，R-Squared值增加或保持不变。R-Squared不会因添加对模型没有任何价值的功能而受到惩罚。因此，R-Squared的改进版本是经过调整的R-Squared。

当我们添加更多特征时，分母 $n - (k + 1)$ 中的项减少，因此整个表达式增加。

如果R-Squared没有增加，那意味着添加的功能对我们的模型没有价值。因此总的来说，我们从1减去一个更大的值，而调整后的 r^2 反过来会减少。

除了这11个指标之外，还有另一种检查模型性能的方法。这7种方法在数据科学中具有统计学意义。但是，随着机器学习的到来，我们现在拥有更强大的模型选择方法。**是！我在谈论交叉验证。**

但是，交叉验证并不是一个真正的评估指标，它可以公开用于传达模型的准确性。但是，交叉验证的结果提供了足够直观的结果来概括模型的性能。

现在让我们详细了解交叉验证。

12.交叉验证

让我们首先了解交叉验证的重要性。由于时间紧迫，这些天我没有太多时间参加数据科学竞赛。很久以前，我参加了Kaggle的TFI比赛。在不深入了解我的竞争表现的情况下，我想向您展示我的公共和私人排行榜得分之间的差异。

以下是Kaggle得分的一个例子！

对于TFI比赛，以下是我的三个解决方案和分数（越小越好）：

Submission	Files	Public Score	Private Score	Selected?
Mon, 04 May 2015 12:59:31 Edit description	submission_all_with_sai3.csv	1649776.86428	1809956.02878	<input checked="" type="checkbox"/>
Mon, 04 May 2015 11:48:54 Edit description	submission_all_wit	1651071.47287	1802503.24607	<input type="checkbox"/>
Mon, 13 Apr 2015 13:28:08 Edit description	submission_all.csv	1677138.71291	1795007.23155	<input checked="" type="checkbox"/>

您会注意到，公共分数最差的第三个条目变成了私人排名的最佳模型。在

“submission_all.csv”之上有超过20个模型，但我仍然选择“submission_all.csv”作为我的最终条目（这确实很有效）。是什么导致了这种现象 我的公共和私人排行榜的不同之处是过度拟合造成的。

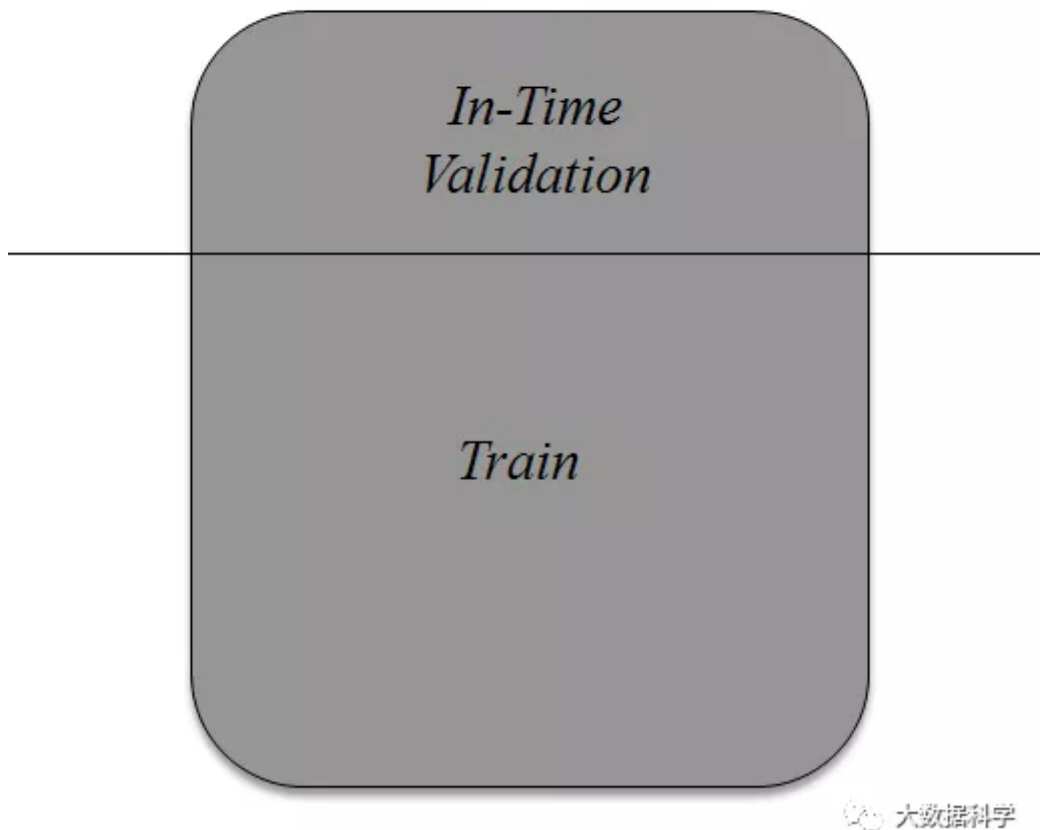
过度拟合只不过是当你的模型变得非常复杂时它也会开始捕捉噪音。这种“噪音”对模型没有任何价值，但只是不准确。

在下一节中，我将讨论在我们真正了解测试结果之前如何知道解决方案是否过度适应。

概念：交叉验证

交叉验证是任何类型的数据建模中最重要的概念之一。它只是说，尝试留下一个样本，在这个样本上你不训练模型，并在最终确定模型之前测试该样本上的模型。

Training Population



上图显示了如何使用及时样本验证模型。我们简单地将人口分成2个样本，并在一个样本上建立模型。其余人口用于及时验证。

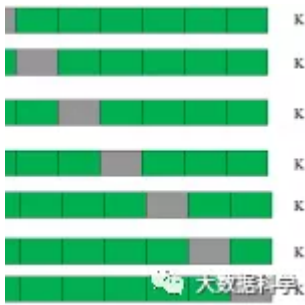
上述方法会有消极的一面吗？

我认为，这种方法的一个消极方面是我们从训练模型中丢失了大量数据。因此，该模型具有很高的偏差。这不会给出系数的最佳估计。那么下一个最佳选择是什么？

如果，我们将50:50的训练人口和前50的火车分开并在休息时进行50次验证。然后，我们在另外50次训练，在前50次训练。这样我们在整个人群中训练模型，然而，一次性使用50%。这样可以减少偏差，因为样品选择在一定程度上可以提供较小的样本来训练模型。这种方法称为2倍交叉验证。

k-fold交叉验证

让我们将最后一个例子从2倍交叉验证推断为k-fold。现在，我们将尝试可视化k-fold验证的工作原理。



这是一个7倍的交叉验证。

以下是幕后发生的事情：我们将整个人口划分为7个相同的样本。现在我们在6个样本（绿色框）上训练模型并在1个样本（灰色框）上进行验证。然后，在第二次迭代中，我们使用不同的样本训练模型作为验证。在7次迭代中，我们基本上在每个样本上构建了模型，并将每个样本作为验证。这是一种减少选择偏差并减少预测功率变化的方法。一旦我们拥有所有7个模型，我们将平均误差项找到哪个模型是最好的。

这有助于找到最佳（非过度拟合）模型？

k折交叉验证广泛用于检查模型是否是过度拟合。如果k次建模中的每一次的性能度量彼此接近并且度量的均值最高。在Kaggle比赛中，您可能更多地依赖交叉验证分数而不是Kaggle公共分数。通过这种方式，您将确保公共分数不仅仅是偶然的。

我们如何使用任何型号实现k-fold？

R和Python中的k-fold编码非常相似。以下是在Python中编码k-fold的方法：

```
from sklearn import cross_validation model = RandomForestClassifier(n_estimator:
```

但是我们如何选择k呢？

这是棘手的部分。我们需要权衡选择k。

对于小k，我们有更高的选择偏差但性能差异很小。

对于大k，我们有一个小的选择偏差但性能差异很大。

想想极端情况：

$k = 2$ ：我们只有2个样本类似于我们的50-50个例子。在这里，我们每次仅在50 %的人口中立模型。但由于验证是一个重要的人口，验证性能的差异是最小的。

$k = \text{观察次数 } (n)$ ：这也称为“留一个”。我们有 n 次样本和建模重复 n 次，只留下一个观察结果进行交叉验证。因此，选择偏差很小，但验证性能的差异非常大。

通常，对于大多数目的，建议使用 $k = 10$ 的值。

结束笔记

测量训练样本的表现更少。暂时搁置验证批次是浪费数据。K-Fold为我们提供了一种使用每个单数据点的方法，可以在很大程度上减少这种选择偏差。此外，K折交叉验证可以与任何建模技术一起使用。

此外，本文中介绍的度量标准是分类和回归问题中评估最常用的度量标准。

您经常在分类和回归问题中使用哪个指标？您之前是否使用过k-fold交叉验证进行任何分析？您是否看到使用批量验证的任何重大好处？请在下面的评论部分告诉我们您对本指南的看法。

本文由数据分析网 - 翻译小组 编译发布，出处：

<https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/>，转载或内容合作请联系授权，未经允许谢绝转载，本文链接：

<https://www.afenxi.com/71704.html>