你知道这11个重要的机器学习模型评估指标吗?

原创 Arno 磐创AI 2019-08-21

译者 | Arno
来源 | Analytics Vidhya

【磐创AI导读】：评估一个模型是建立一个有效的机器学习模型的核心部分，本文为大家介绍了一些机器学习模型评估指标，希望对大家有所帮助。想要获取更多的机器学习、深度学习资源，欢迎大家点击上方蓝字关注我们的公众号：磐创AI。

概览

- 评估一个模型是建立一个有效的机器学习模型的核心部分
- 评价指标有混淆矩阵、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

建立机器学习模型的想法是基于一个建设性的反馈原则。你构建一个模型，从指标中获得反馈，进行改进，直到达到理想的精度为止。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数据科学家不费心检查他们的模型的鲁棒性。一旦他们完成了模型的构建，他们就会匆忙地将其应用到不可见的数据上。这是一种错误的方法。

你的动机不是简单地建立一个预测模型。它是关于创建和选择一个模型，使其对样本外的数据具有高精度。因此，在计算预测值之前，检查模型的准确性是至关重要的。

在我们的行业中，我们考虑不同种类的指标来评估我们的模型。指标的选择完全取决于模型的类型和模型的实现计划。

在你构建完模型之后，这11个指标将帮助你评估模型的准确性。考虑到交叉验证的日益流行和重要性，我还将在本文中讨论它。

热身:预测模型的类型

当我们谈论预测模型时，我们谈论的要么是回归模型(连续输出)，要么是分类模型(离散输出)。这些模型中使用的评估指标是不同的。

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**: 像SVM和KNN这样的算法创建一个类输出。例如，在一个二分类问题中，输出将是0或1。然而，今天我们有算法可以将这些类输出转换为概率。但是这些算法并没有被统计学界很好地接受。
2. **概率输出**: 逻辑回归、随机森林、梯度增强、Adaboost等算法给出概率输出。将概率输出转换为类输出只需要创建一个阈值。

在回归问题中，我们的输出没有这样的不一致性。输出在本质上总是连续的，不需要进一步处理。

例证

分类模型评估指标的讨论中，我使用了我在Kaggle上的BCI挑战的预测。这个问题的解决超出了我们在这里讨论的范围。然而，本文使用了在此训练集上的最终预测。对这个问题的预测结果是概率输出，假设阈值为0.5，将这些概率输出转换为类输出。

1. 混淆矩阵(Confusion Matrix)

混淆矩阵是一个NxN矩阵，其中N是预测的类数。对于我们的案例，我们有N=2，因此我们得到一个2x2矩阵。你需要记住一个混淆矩阵一些定义:

- **准确率(Accuracy)**: 分类模型中所有判断正确的结果占总观测值得比重。
- **精确率、查准率(Precision)**: 在模型预测是正例的所有结果中，模型预测对的比重

- **真负率**: 在模型预测是负例的所有结果中，模型预测对的比重
- **召回率、查全率(Recall)、灵敏度(Sensitivity)**: 在真实值是正例的所有结果中，模型预测对的比重
- **特异度(Specificity)**: 在真实值是负例的所有结果中，模型预测对的比重

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%		88.0%

我们的案例的准确率达到88%。从以上两个表中可以看出，精确率较高，而真负率较低。灵敏度和特异度也一样。这主要是由我们选择的阈值驱动的。如果我们降低阈值，这两对完全不同的数值会更接近。

一般来说，我们关心的是上面定义的指标其中之一。例如，在一家制药公司，他们会更关注最小的错误正类诊断。因此，他们将更加关注高特异度。另一方面，损耗模型更关注灵敏度。混淆矩阵通常只用于类输出模型。

2. F1 Score

在上一节中，我们讨论了分类问题的精确率和召回率，并强调了我们的选择案例的精确率/召回率基础的重要性。如果对于一个案例，我们试图同时获得最佳的精确率和召回率，会发生什么呢? F1-Score是一个分类问题的精确率和召回率的调和平均值。f1分的计算公式如下:

$$F_1 = \left(\frac{\text{recall}^{-1} + \text{precision}^{-1}}{2} \right)^{-1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

一个显而易见的问题是为什么取调和均值而不是算术均值。这是因为调和均值对极值的惩罚更多。让我们用一个例子来理解这一点。我们有一个二分类模型，结果如下:

精确率:0，召回率:1

这里取算术平均值，得到0。5。很明显，上面的结果来自于一个“傻瓜”的分类器，它忽略了输入，只选择其中一个类作为输出。现在，如果我们取调和均值，我们会得到0，这是准确的，因为这个模型对所有的目的都没用。

这似乎是简单的。然而，在某些情况下，对精确率和召回率的重视程度有所不同。将上面的表达式稍微修改一下，我们可以为此包含一个可调参数 β ，我们得到:

$$F_\beta = (1+\beta^2)\cdot\frac{\text{precision}\cdot\text{recall}}{(\beta^2\cdot\text{precision})+\text{recall}}.$$

Fbeta 测量用户认为召回率比精确率重要 β 倍模型的有效性。

3. 增益图和提升图(Gain and Lift charts)

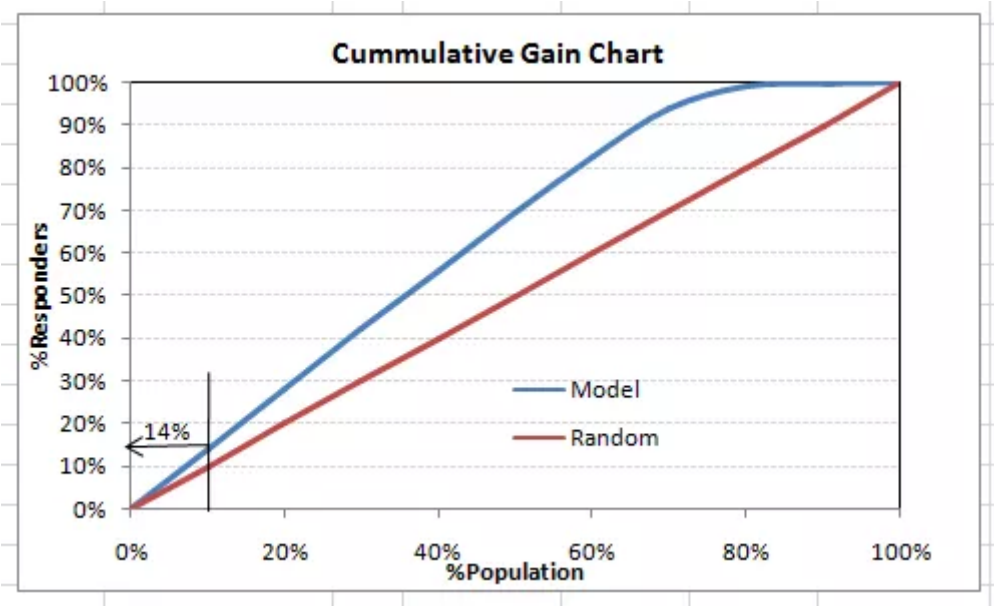
增益图和提升图主要用于检验概率的排序。以下是制作增益图和提升图的步骤:

步骤1:计算每个样本的概率。
步骤2:按递减顺序排列这些概率。
步骤3:构建十分位数，每个组有近10%的样本数。
步骤4:计算每个十分位处好(Responders)、坏(Non-responders)和总的响应率。

你会得到以下表格，你需要从中绘制增益图和提升图:

Lift/Gain	Column Labels			%Rights	%Wrongs	%Population	Cum %Right	Cum %Pop	Lift @decile	Total Lift
Row Labels	0	1	Grand Total	0%	0%	0%	0%	0%		
1		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%	1%	100%
Grand Total	1590	3850	5440							

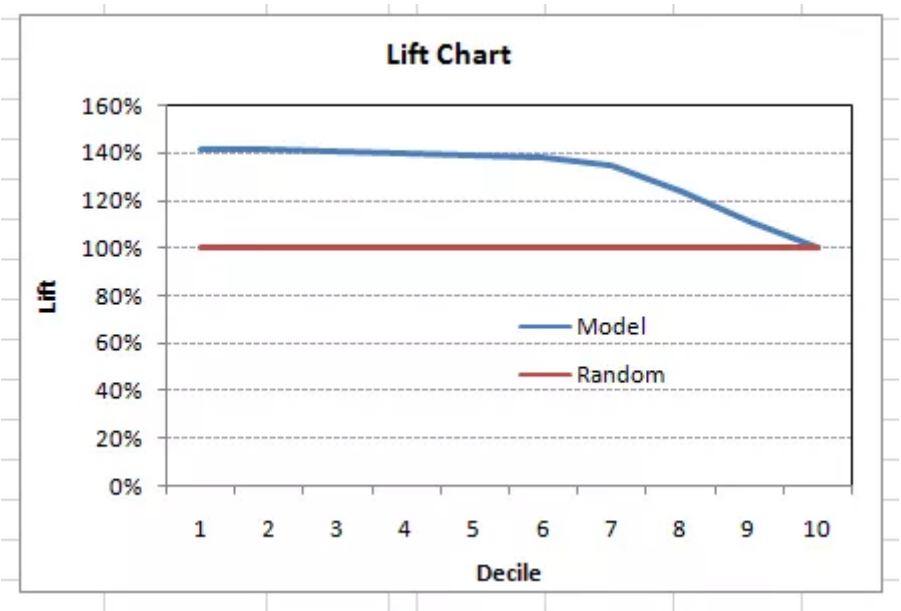
这是一个包含很多信息的表。累积增益图(Cumulative Gain chart)是累计 %Right和累计%Population之间的图。下面是对应我们的案例的图:



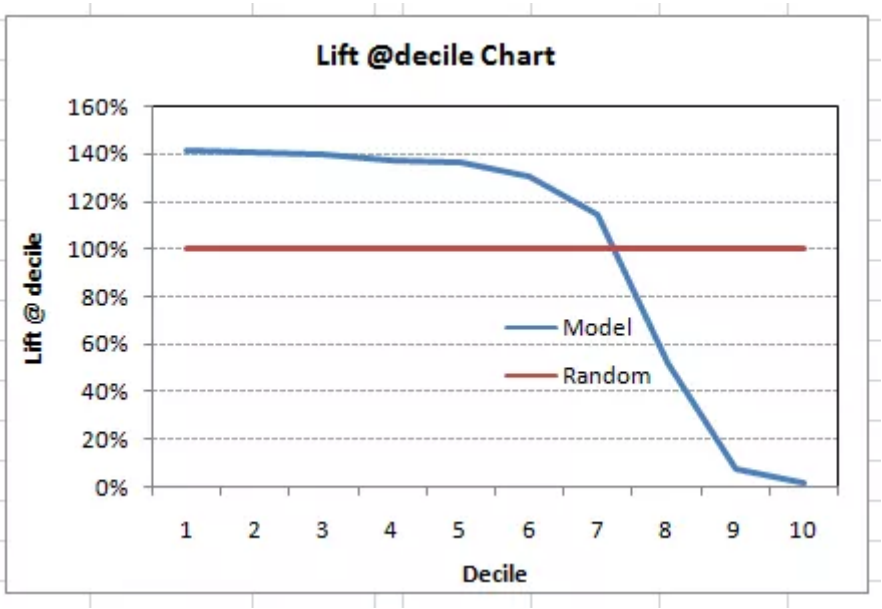
该图告诉你模型将responders与non-responders分离的程度。

在第一个十分位处我们可以达到的最大提升是多少？从本文的第一个表中，我们知道responders的总数是3850.第一个十分位处将包含543个观察值。因此，第一个十分位数的最大提升可能是543 / 3850约为14.1%。因此，此模型非常接近完美。

现在让我们绘制提升曲线。提升曲线是总提升(total lift)与%population之间的关系曲线。请注意，对于随机模型，它始终保持100%不变。以下是我们的案例对应的提升图:



你还可以用十分位数绘制十分位处对应的提升:



这个图告诉你什么？它告诉你我们的模型运行直到第7个十分位处都表现得不错。在第三十分位处和第七十分位处之间提升超过100%的模型都是一个很好的模型。否则，你可能首先考虑过采样。

提升/增益图广泛用于活动目标问题。这告诉我们，对于特定活动，可以在哪个十分位处可以定位客户。此外，它会告诉你对新的目标基础的期望响应量。

4. KS图(Kolomogorov Smirnov chart)

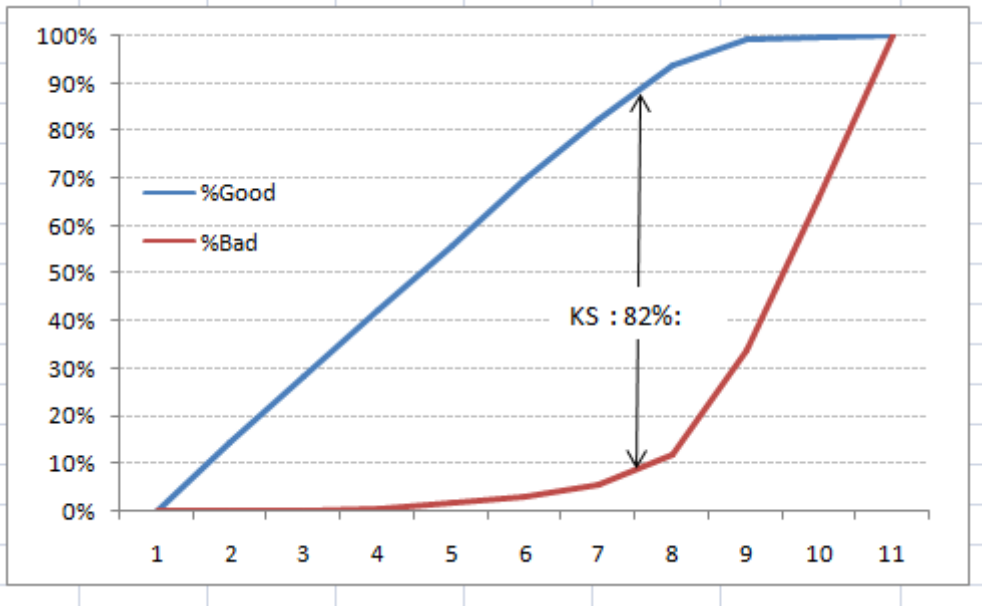
K-S或Kolmogorov Smirnov图测量分类模型的性能。更准确地说，K-S是衡量正负例分布之间分离程度的指标。如果将人口划分为两个独立的组，其中一组包含所有正例而另一组包含所有负例，则K-S值为100。

另一方面，如果模型不能区分正负例，那么模型从总体中随机选择案例。K-S值将为0.在大多数分类模型中，K-S将介于0和100之间，并且值越高，模型在区分正负例情况时越好。

对于我们的案例，下面是对应的表格:

					Cummulative			
Lift/Gain	Columr			%Rights	%Wrongs	Cum %Rig	Cum %Wrong	K-S	
Row La	0	1	Grand Tot	0%	0%	0%	0%	0%	
1		543	543	14%	0%	14%	0%	14%	
2	2	542	544	14%	0%	28%	0%	28%	
3	7	537	544	14%	0%	42%	1%	42%	
4	15	529	544	14%	1%	56%	2%	54%	
5	20	524	544	14%	1%	69%	3%	67%	
6	42	502	544	13%	3%	83%	5%	77%	
7	104	440	544	11%	7%	94%	12%	82%	K-S
8	345	199	544	5%	22%	99%	34%	65%	
9	515	29	544	1%	32%	100%	66%	34%	
10	540	5	545	0%	34%	100%	100%	0%	
Grand Tot	1590	3850	5440						

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



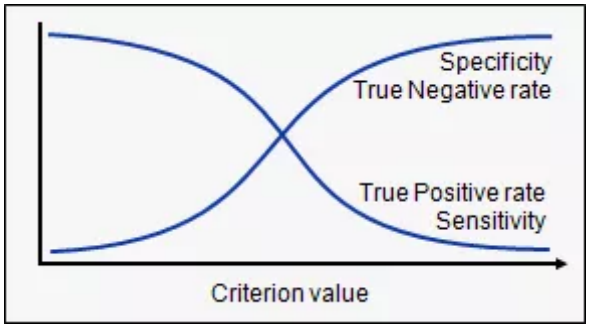
到目前为止所涵盖的指标主要用于分类问题。我们了解了混淆矩阵，提升和增益图以及kolmogorov-smirnov图。让我们继续学习一些更重要的指标。

5. AUC曲线(AUC-ROC)

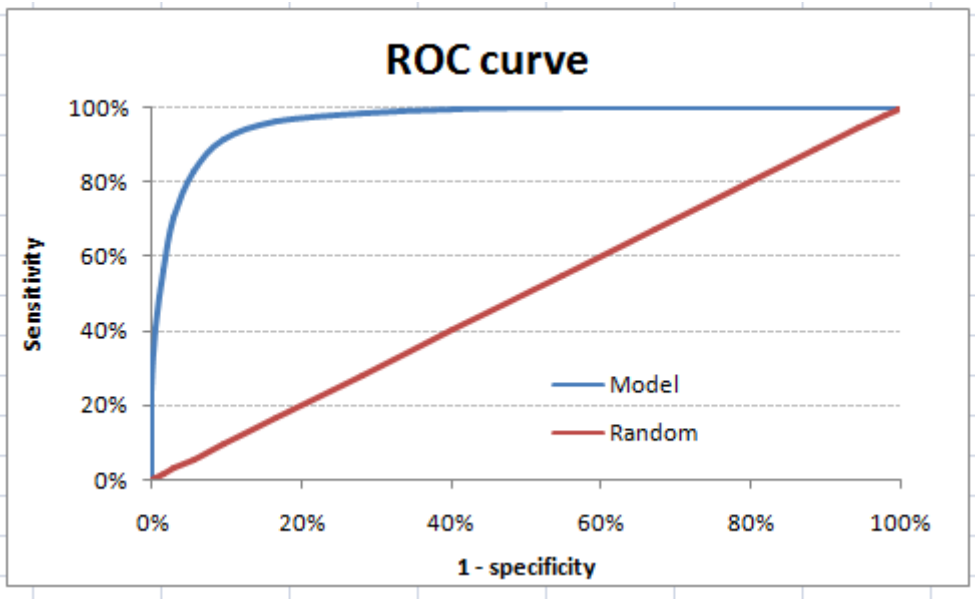
这是业界流行的指标之一。使用ROC曲线的最大优点是它独立于responders比例的变化。让我们首先尝试了解什么是ROC(接收者操作特征)曲线。如果我们看下面的混淆矩阵，我们观察到对于概率模型，我们得到每个度量的不同值。

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，对于每个灵敏度，我们得到不同的特异度。两者的变化如下：



ROC曲线是灵敏度和(1-特异度)之间的曲线。(1-特异性)也称为假正率，灵敏度也称为真正率。以下我们案例对应的ROC曲线。



以阈值为0.5为例，下面是对应的混淆矩阵：

Target		
	1	0 Grand
	3,834	639
	16	951
	3,850	**1,590**
	99.6%	40.19%

你可以看到，这个阈值的灵敏度是99.6%，(1-特异性)约为60%。这一对值在我们的ROC曲线中成为一个点。为了将该曲线映射为数值，我们计算该曲线下的面积(AUC)。

注意到，整个正方形的面积是1 * 1 = 1。因此AUC本身是曲线下的面积与总面积的比率，对于我们的实验，我们的AUC ROC值为96.4%。以下是一些经验法则(thumb rules)：

- .90-1 = excellent (A)
- 0.80-.90 = good (B)
- 0.70-.80 = fair (C)
- 0.60-.70 = poor (D)
- 0.50-.60 = fail (F)

我们看到我们当前模型属于excellent范围。但这可能只是过拟合，在这种情况下，验证变得非常重要。

需要记住几点

1. 对于将类作为输出的模型，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。

在概率模型的情况下，我们很幸运可以得到一个AUC-ROC的数值。但是，我们仍然需要查看整个曲线以做出最终的决定。一个模型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的好处

为什么要使用ROC而不是提升曲线等指标?

提升取决于人口的总响应率(total response rate)。因此，如果人口的响应率发生变化，同一模型将给出不同的提升图，这种的情况的解决方案可以用真正提升图(true lift chart)(在每个十分位处找到提升和模型最大提升的比率)。但这种比率没有什么意义。

另一方面，ROC曲线几乎与响应率无关。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. 对数损失(Log Loss)

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样本的更高概率的能力。在这种情况下，我们可以使用对数损失，即每个实例的正例预测概率的对数的负平均值。

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

- p(yi)是预测为正类的概率
- 1-p(yi)是预测为负类的概率
- yi = 1表示正类，0表示负类(实际值)

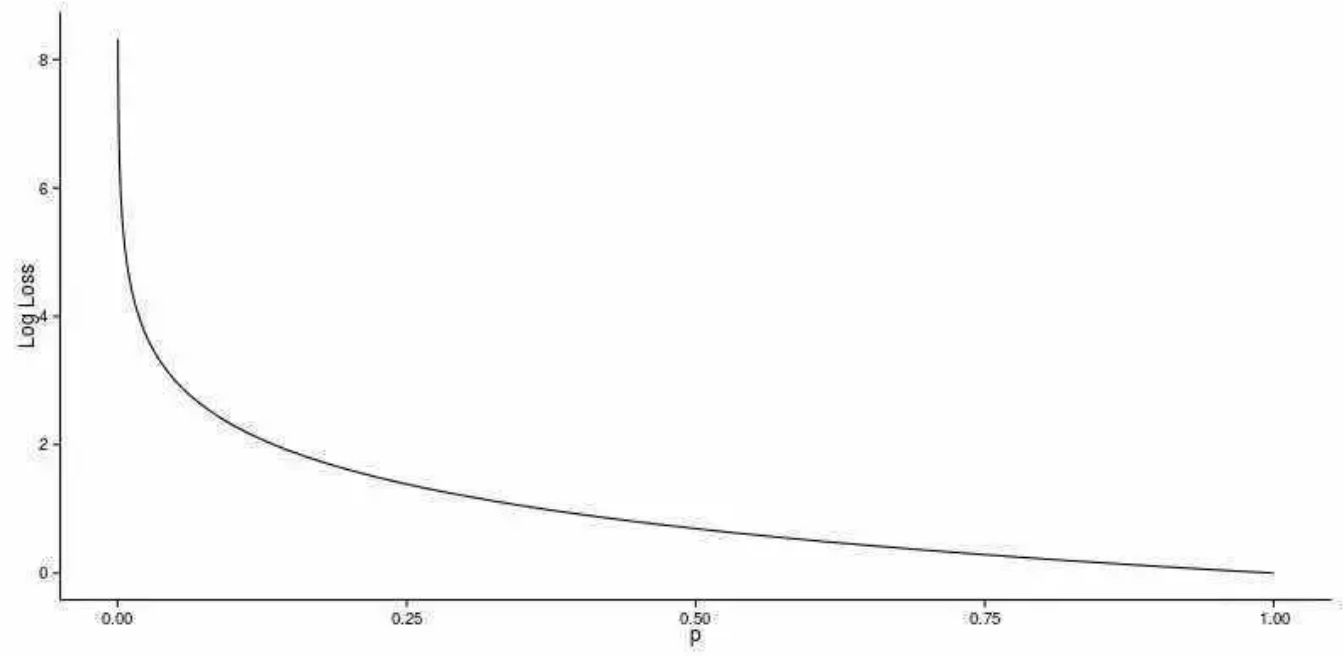
让我们计算几个随机值的对数损失，以得到上述数学函数的要点：

Logloss(1, 0.1) = 2.303

Logloss(1, 0.5) = 0.693

Logloss(1, 0.9) = 0.105

如果我们绘制这种关系，我们将获得如下曲线：



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，降低对数损失，对模型更好。但是，对于好的对数损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(Gini Coefficient)

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曲线与对角线之间的面积与对角线上三角形的面积之比。

以下是使用的公式：

Gini = 2*AUC – 1

基尼系数高于60％是一个很好的模型。对于我们的案例，我们的基尼系数为92.7％。

8. Concordant – Discordant ratio

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responder和其他non-responder。我们有多少这样的配对？

我们有两对AB和BC。现在对于2对中的每一对，一致对(concordant pair)是responder的概率高于non-responder的概率。而不一致的对(discordant pair)是反之亦然的。如果两个概率相等，我们说它是平等的。让我们看看在我们的案例中发生了什么：

AB – Concordant

BC – Discordant

因此，在这个例子中我们有50％的一致案例。超过60％的一致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。对于定位的客户数量则再次采用KS / Lift图。

9. 均方根误差(Root Mean Squared Error,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”使该指标能够显示大的偏差。
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(mean absolute error)相比，RMSE提供更高的权重并惩罚大的错误。

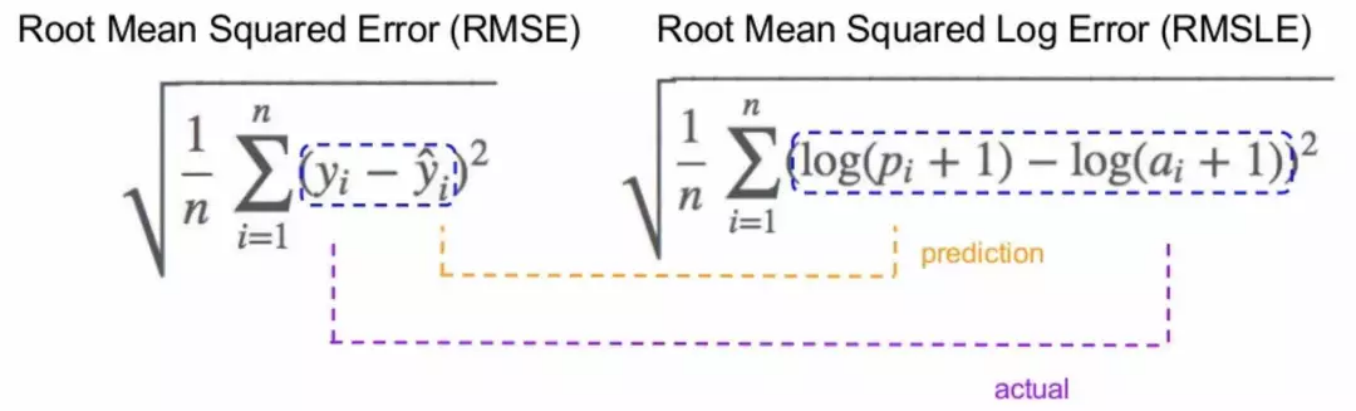
RMSE指标由下式给出：

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

其中，N是样本总数。

10. 均方根对数误差(Root Mean Squared Logarithmic Error)

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。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. 如果预测或是实际值很大：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-squared/Adjusted R-squared

我们了解到，当RMSE降低时，模型的性能将会提高。但仅凭这些值并不直观。

在分类问题的情况下，如果模型的准确率为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RMSE指标时，我们没有比较基准。

这是我们可以使用R-Squared指标的地方。R-Squared的公式如下：

$$R^2 = 1 - \frac{\text{MSE(model)}}{\text{MSE(baseline)}}$$

$$\frac{\text{MSE(model)}}{\text{MSE(baseline)}} \qquad \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(\overline{y}_i - \hat{y}_i)^2}$$

MSE(model)：预测与实际值的均方差。
MSE(baseline)：平均预测值与实际值的均方差

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，这个模型只是预测训练集中目标的平均值作为预测。

Adjusted R-Squared

当模型表现与baseline相同时R-Squared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。调整后的R-Squared的公式由下式给出：

$$\bar{R}^2 = 1 - \left(1 - R^2\right)\left[\frac{n-1}{n-\left(k+1\right)}\right]$$

k：特征数量
n：样本数量

如你所见，此指标会考虑特征的数量。当我们添加更多特征时，分母中项n-(k +1)减小，因此整个表达式增加。

如果R-Squared没有增加，那意味着添加的特征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r2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。没错!就是交叉验证。

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说明模型的性能。

现在让我们详细了解交叉验证。

12. 交叉验证(Cross Validation)

让我们首先了解交叉验证的重要性。很久以前，我参加了Kaggle的TFI比赛。我想向你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!

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	☑
Mon, 04 May 2015 11:48:54 Edit description	submission_all_wit	1651071.47287	1802503.24607	☐
Mon, 13 Apr 2015 13:28:08 Edit description	submission_all.csv	1677138.71291	1795007.23155	☑

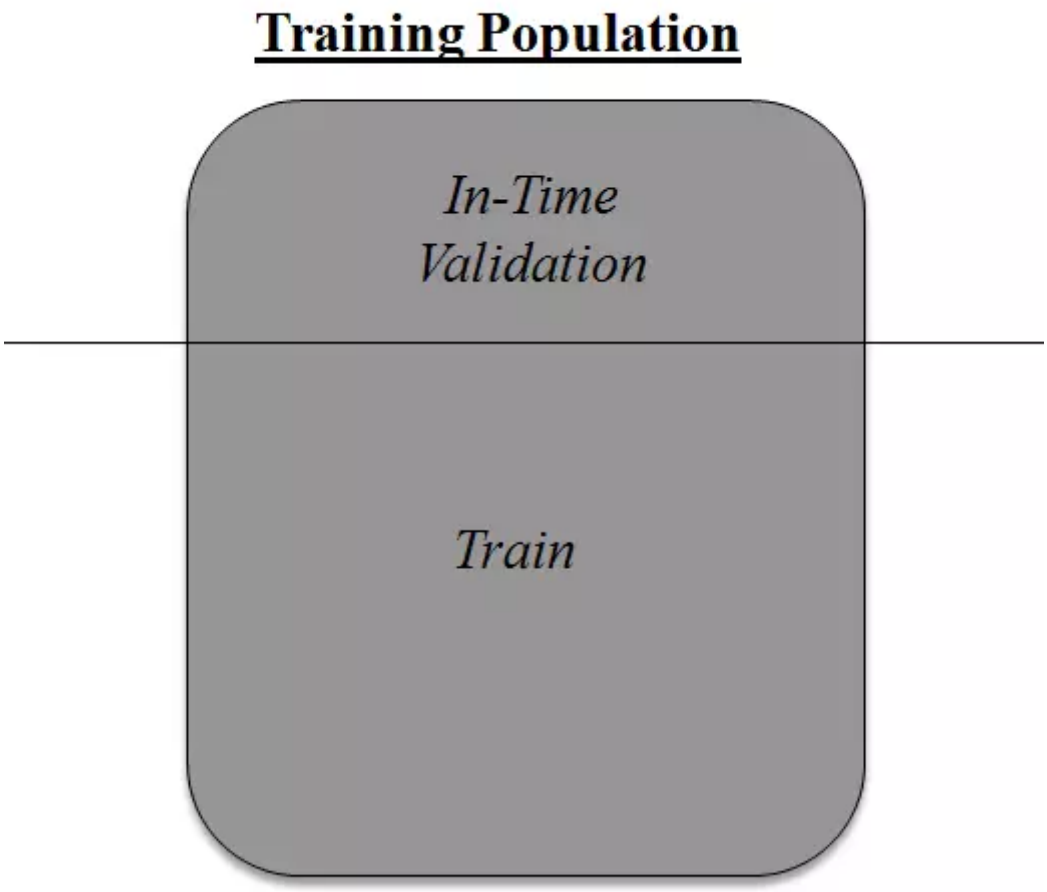
你会注意到，公共分数最差的第三个条目变成了私人排名的最佳模型。在“submission_all.csv”之上有超过20个模型，但我仍然选择“submission_all.csv”作为我的最终条目(这确实很有效)。是什么导致了这种现象?我的公共和私人排行榜的不同之处是过度拟合造成的。

过度拟合只不过是当你的模型变得非常复杂时它会捕捉噪音。这种“噪音”对模型没有任何价值除了造成模型不准确。

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拟合。

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，尝试留下一个样本集，不在这个样本集上训练模型，并在最终确定模型之前在该样本集上测试模型。



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

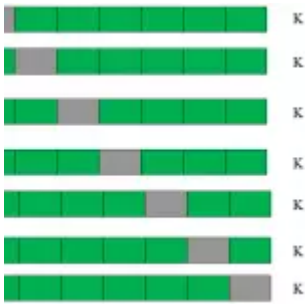
上述方法会有消极的一面吗?

这种方法的一个消极方面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，我们将训练人口以50:50的划分，前50用于训练，后50用于验证。然后，我们在后50进行训练，在前50进行测试。这样我们在整个人口中训练模型，即使是一次性使用50％。这样可以减少偏差，因为样本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折交叉验证。

k折交叉验证

让我们最后演示一个从2折交叉验证到k折交叉验证的例子。现在，我们将尝试可视化k折交叉验证的工作原理。



这是一个7折交叉验证。我们将整个人口划分为7个相同的样本集。现在我们在6个样本集(绿色框)上训练模型并在1个样本集(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集训练模型剩余的一个样本集作为验证。在7次迭代中，我们基本上在每个样本集上构建了模型，并将每个样本集作为验证。这是一种减少选择偏差并减少预测方差的方法。一旦我们拥有所有7个模型，我们使用平均误差决定那个模型是最好的。

这怎样找到最佳(非过拟合)模型?

k折交叉验证广泛用于检查模型是否过拟合。如果k次建模中的每一次的性能度量彼此接近，则度量的均值最高。在Kaggle比赛中，你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，你将确保公共分数不仅仅是偶然的。

我们如何使用任意模型上实现k折?

R和Python中的k折编码非常相似。以下是在Python中k折编码的方法：

```
from sklearn import cross_validation
model = RandomForestClassifier(n_estimators=100)

#简单的K-fold交叉验证。5折。(注意：在较旧的scikit-learn版本中，“n_folds”参数名为“k”。
cv = cross_validation.KFold(len(train), n_folds=5, indices=False)
results = []
 # "model" 可以替换成你的模型对象
 # "“error_function”可以替换为cv中traincv，testcv的分析错误函数：
probas = model.fit(train[traincv], target[traincv]).predict_proba(train[testcv])
results.append( Error_function )
#打印出交叉验证结果的平均值
print "Results: " + str( np.array(results).mean() )
```

怎样选择k?

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方差很小。

对于大k，我们有一个小的选择偏差但方差很大。

k =样本数(n)：这也称为“留一法”。我们有n个样本集合和并重复建模n次，只留下一个样本集进行交叉验证。
通常，对于大多数目的，建议使用k = 10的值。

总结

在训练样本上评估模型没有意义，但留出大量的样本以验证模型则比较浪费数据。k折交叉验证为我们提供了一种使用每个数据点的方法，可以在很大程度上减少这种选择偏差。

另外，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。

你在分类和回归问题中经常使用哪个指标？你之前是否使用过k折交叉验证进行分析？你是否看到使用各种验证的好处？请在下面的评论部分告诉我们你的看法。

留言送书福利

为了鼓励大家踊跃在文章**留言区**分享自己的看法，磐创AI推出了“留言送书”活动~在本文文末**留言**即可参与活动，留言内容需为主题相关。欢迎大家在日常推文中留言，以后将不定期推出“留言送书”活动。

这次磐小仙精心挑选了本《机器学习线性代数基础》送给大家。本书以机器学习涉及的线性代数核心知识为重点，进行新的尝试和突破，环环相扣地展开线性代数与机器学习算法紧密结合的核心内容，并分析推荐系统和图像压缩两个实践案例。极力避免数学的晦涩枯燥，充分挖掘线性代数的几何内涵，并以Python语言为工具进行数学思想和解决方案的有效实践。书籍详细介绍可以点击文末**阅读原文**查看。

/ 今日赠送书籍 /