评分卡模型的评估方法论

author: 俞鑫鑫
date: 2018-11-14

> 参考资料：
> 单良：《数据化风控》
> 周志华：《机器学习》
> [德]安德里亚斯·穆勒 / [美]莎拉·吉多 ：《python机器学习基础教程》

目录

1.区隔能力
2.准确性
3.泛化能力
4.稳定性
5.可解释性
6.评估流程

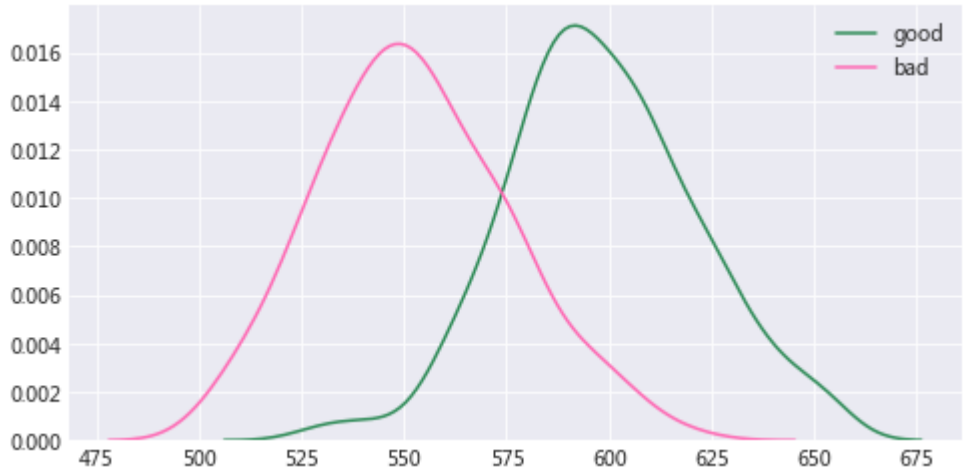
一. 区隔能力

1.AUC，KS，Gini系数
这三个作为最常用的评估指标，网上的资料很多，这里就不详细赘述了，贴一下三者综合的判断标准：

Gini系数	KS值	AUC值	模型优劣程度
0	<0.2	0.5	没有区隔能力
0-0.4	0.2-0.3	0.5-0.7	区隔能力较弱
0.4-0.6	0.3-0.5	0.7-0.8	可以接受的区隔能力
0.6-0.8	0.5-0.75	0.8-0.9	区隔能力较强
0.8-1	>=0.75	0.9-1	模型可能过度匹配

2.评分分布图的区分度
如果通过评分能将好坏用户完全区隔开来，那是理想中最好的评分卡模型，但实际情况中好坏用户的评分会有一定程度的重叠，我们要做的就是尽量减

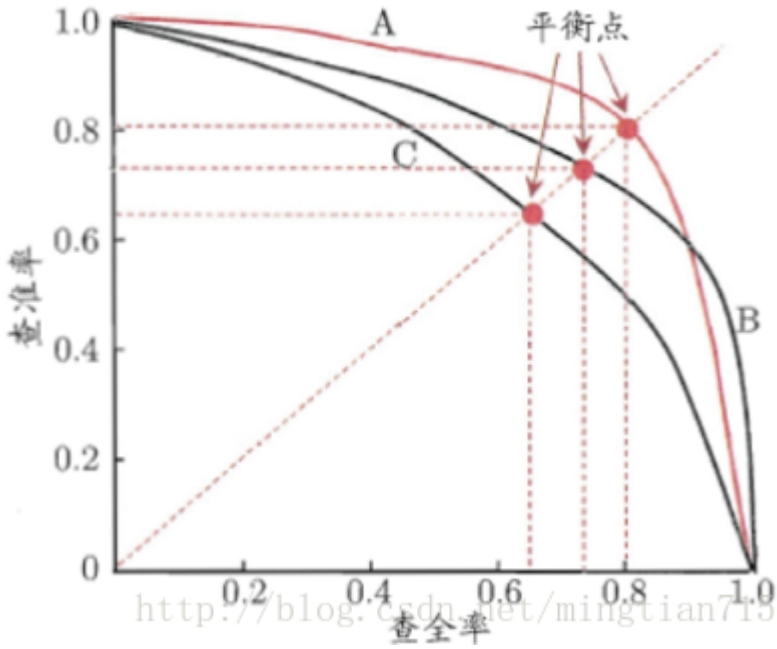
小重叠程度。另外好坏用户的得分分布最好都是正态分布，如果呈双峰或多峰分布，那么很有可能是某个变量的得分过高导致，这样对评分卡的稳定性会有影响。



二. 准确性

1.混淆矩阵（精确率，查全率，误伤率）
这三个指标的值取决于评分卡的cutoff点怎么设置。评分卡最后会输出一个评分分布表，根据评分的等级和业务目标来选择适当的cutoff点。如何通过这三个指标来评价模型可参考之前写的“策略评估方法论”。

2.PR曲线
ROC曲线表现的是查全率和假正率之间的关系，而PR曲线则是查全率和精确率之间的关系。以精确率作为y轴，查全率作为x轴作图，就得到了PR曲线，曲线上的点对应根据设定的cutoff点得出的精确率和查全率，在实际情况中，精确率和查全率不可能同时高，这两者是一对矛盾的度量。如果业务目标要同时兼顾精确率和查全率，可以绘制PR曲线来找出两者的平衡点（如下图所示）：



同时PR曲线也可以用来评价模型的效果，这里引入两个度量：
1）BEP(平衡点)：当查全率=精确率时的对应的cutoff点，假如在相同的评分刻度下，A评分卡的BEP为620，B评分卡的BEP为600，则A评分卡优于B评分卡。
2）Fp度量：Fp考虑了查全率和精确率不同的重要程度，能够表达出对精确率/查全率的不同偏好。β=1表示两者重要性程度相同，β>1时更看重查全率，β<1更看重精确率。

PS: P--精确率 R--查全率 β--重要程度

$$F_\beta = \frac{(1+\beta^2) \times P \times R}{(\beta^2 \times P) + R}$$

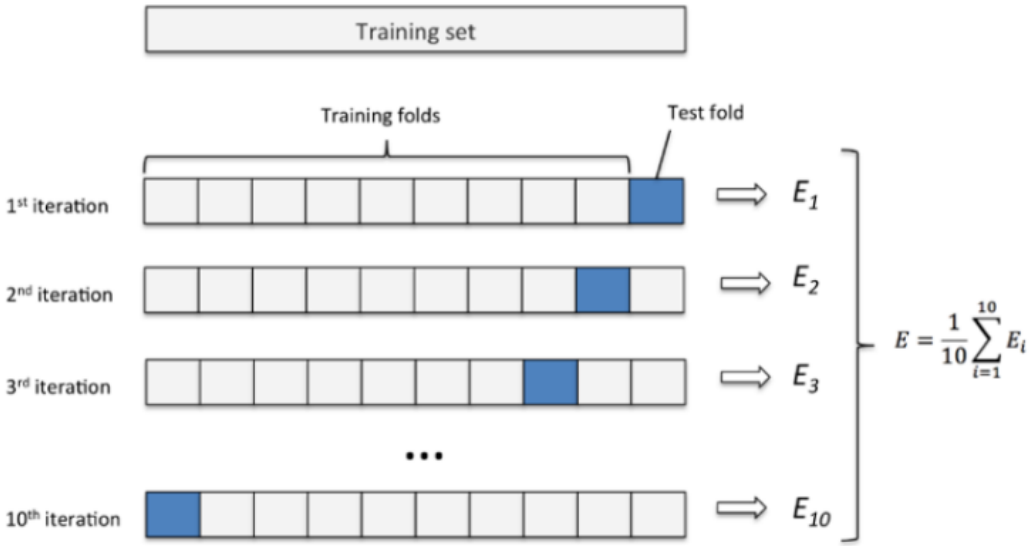
三. 泛化能力

1.**交叉验证---模型泛化能力的评估**
泛化能力指的是模型对于未知数据的预测能力。实际建模中，一般用模型在测试集上的表现来近似泛化能力。如果测试集和训练集的表现差不多，则模型的泛化能力较好。但这种做法过于简单，因为我们只是单纯的把数据集随机划分为训练集和测试集，假如很幸运地将难以分类的样本划分进训练集中，则在测试集会得出一个很高的分数，但如果不够幸运地将难以分类的样本划分进测试集中，则会得到一个很低的分数。所以得出的结果随机性太大，不够具有代表性。
而交叉验证对于泛化能力的评估，比单次划分训练集和测试集的方法更加稳

定，全面。交叉验证中每个样本都会出现在训练集和测试集中各一次，因此，模型需要对所有样本的泛化能力都很好，才能使其最后交叉验证得分，及其平均值都很高，这样的结果更加稳定，全面，具有说服力。另外通过对数据集多次划分后，还可以通过每个样本的得分比较，来反映数据扰动对于模型结果的影响。

1）**K折交叉验证**

交叉验证的策略有很多，最常见的就是K折交叉验证，下面这张图可以很清晰的解释K折验证的策略：



第一步，不重复抽样将原始数据随机分为 k 份。

第二步，每一次挑选其中 1 份作为测试集，剩余 k-1 份作为训练集用于模型训练。

第三步，重复第二步 k 次，这样每个子集都有一次机会作为测试集，其余子集作为训练集。在每个训练集上训练后得到一个模型，用这个模型在相应的测试集上测试，计算并保存模型的评估指标。

第四步，计算 k 组测试结果的平均值作为模型精度的估计，并作为当前 k 折交叉验证下模型的性能指标。

PS：在实际应用中一般设K值为5-10，建议先将数据集随机打乱，再进行K折划分，k组测试结果的平均值作为模型的最终评估结果，k组测试结果的标准差可以反映数据扰动对于模型结果的影响。在python里利用sklearn的model_selection模块中的cross_val_score可实现交叉验证。对于分类问题，默认使用分层K折策略（不随机打乱数据）,对于回归问题，默认使用标准K折策略。

2）**时间序列交叉验证**

金融数据具有时间周期性的特点，不同时间段的样本分布和变量分布会有一

定差异，首先在选取建模的样本时就要考虑是否能代表总体的样本分布或者近段时间用户的状态。在做交叉验证时也需要考虑到时间周期这一点，例如我们选取的是1月份至10月份的数据，可以借鉴K折验证的思想，将数据集按照月份分为 10份，每次挑选其中一份作为测试集，其他作为训练集，得到10组的验证结果，观察随着月份的推移，模型的结果是否有比较大的趋势变化，这个也可以反映出样本是否稳定。如果变化较明显，则需要分析是什么原因导致的，是内部大的业务政策，还是外部的经济环境。

2.**学习曲线--判定模型的拟合情况**

模型的拟合情况有三种：

1）欠拟合：模型在训练集和测试集的表现结果都很差

2）过拟合：模型在训练集上表现结果很好，但在测试集上的表现结果很差，也就是泛化能力不好。

3）拟合良好：模型在训练集和测试集的表现结果都较好。

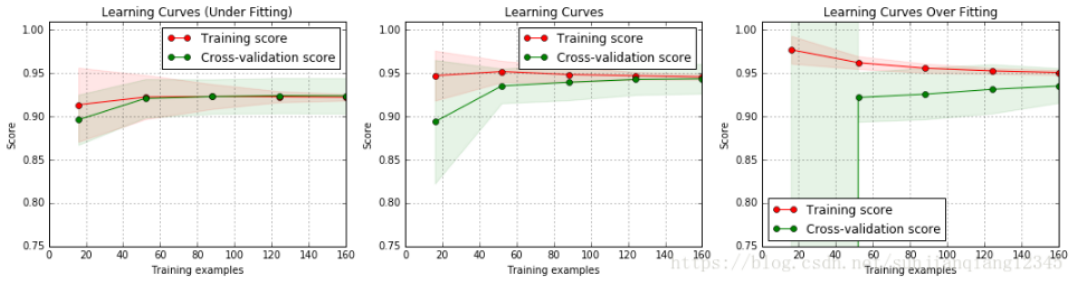
学习曲线：纵轴是训练集的大小，横轴是模型在训练集上和交叉验证集上的平均得分（准确率）。来反映随着训练集大小的改变，模型在训练集和验证集上的误差得分情况。进而判定模型的拟合情况。

模型的三种拟合情况：

1）第一张图中，随着训练集不断增大，模型在训练集和验证集上的得分不断靠近，但两者的得分都比较低，存在欠拟合的问题。

2）第二张图中，随着训练集增大，模型在训练集和验证集上的得分不断靠近，且两者的得分都比较高，说明模型的拟合比较良好。

3）第三张图中，虽然随着训练集增大，训练集和验证集上的得分在不断靠近，但是两者的间隙仍然比较大，说明模型在训练集和验证集的结果差异较大，模型存在“过拟合”问题。



PS：金融模型很容易出现过拟合的问题，解决过拟合的方法有很多，例如增加建模样本，正则化等，例如逻辑回归里可以选择L1正则化或L2正则化，且可设置正则化的强度，另外做评分卡时，入模的变量不宜太多，太多的变量会使模型过于复杂，容易出现过拟合，一般应限制在15个以下。

四. 稳定性

稳定性指标：PSI
PSI是用来衡量两个样本的分布差异，一般是根据某个指标（例如分数）进行分档后，针对不同样本，或者不同时间的样本，计算PSI指标，观察分布是否有差异。衡量标准如下图：

PSI值	稳定性程度
<=0.1	分布比较稳定
0.1-0.25	两个样本分布有部分差异
>0.25	两个样本分布有显著差异

1.**建模样本的稳定性**
1）比较对象：建模样本 VS 总体
我们取的建模样本只是总体的一部分，并且在金融模型中很少用到随机抽样，一般是选取一段时间段内的样本，所以样本能不能代表总体显得非常重要。这里就可以用PSI来衡量样本和总体在分布上的差异程度，目前我想到的可以从以下几点来衡量：
① 变量的缺失率和数据分布：
因为取数的变量一般比较多，如果对每个变量分析稳定性非常耗时间，我的想法是可以先对变量基于IV值进行一下粗筛选，再对筛选后的变量进行PSI计算。一个是从缺失率的角度，另一个是变量的数据分布（数值型变量需先进行等频分箱，再计算PSI）
② 好坏样本的分布：
计算总体的坏用户占比和样本的坏用户占比是否有显著差异（可不计算PSI）。

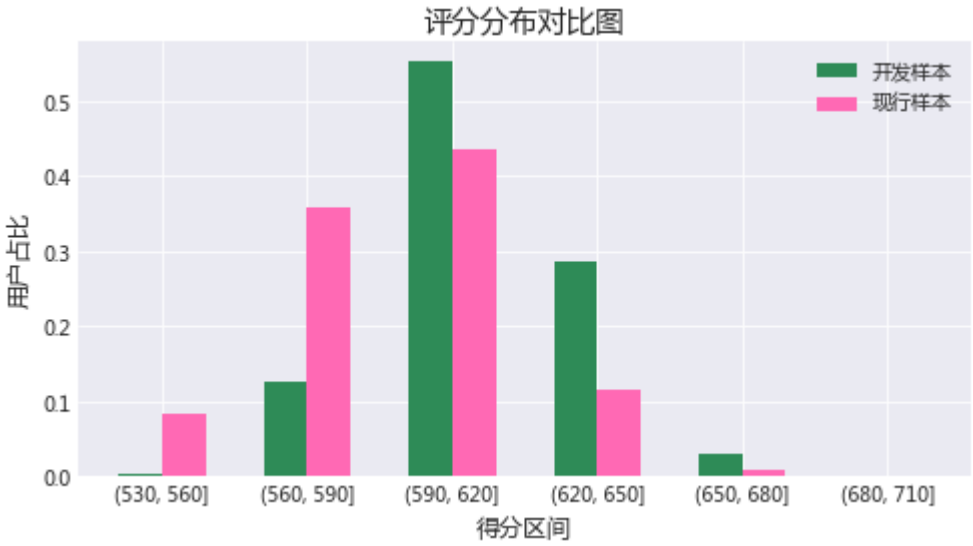
2）比较对象：不同时间段分布的差异
有时候我们不要求样本能代表总体，而是能否代表最近一段时间的样本，例如我们取的是2月--8月的样本，现在的月份为11月份，则需要比较2月-8月的样本与9月--11月的样本是否有显著差异（从缺失率，数据分布，和好坏样本分布的角度），如果差异不大，那么做的评分卡上线后，线上的拒绝率和样本上的拒绝率不会有很大的差异。

2.**评分卡上线后的稳定性**
目的是观察评分卡上线之后的申请用户与建模样本的用户是否一致，并呈现稳定的状态。可在评分卡上线一周或一个月后进行稳定性的分析。
比较对象：上线样本 VS 模型开发样本

衡量角度:

1）评分分布表和评分分布图

评分表是统计开发样本和上线样本在各个评分组别的用户数和用户占比，来比较开发样本和上线样本在评分上的差异。可用评分分布图进行可视化展示（如下图）



2）评分的稳定性PSI

在评分分布表的基础上可以计算评分的稳定度指标（PSI），用来衡量开发样本和上线样本在各评分组别的差异程度，PSI越小，评分卡模型越稳定。但这里要注意的是，用户可能受内部业务活动或者外部经济环境的影响而发生改变，评分卡的稳定性不好，不代表其区隔能力和准确性也不好，如果稳定性不好，需要分析背后的原因，再进行评分卡的调整。另外需要每个月或者每个星期观察PSI的变化，以便了解用户群体的趋势变化情况。稳定性分析表如下图所示：

	得分区间	建模样本户数	建模户数占比	上线样本户数	上线户数占比	占比差异	占比权重	Index	PSI
0	(530, 560]	109	0.004	450	0.083	0.079	3.032	0.240	0.698
1	(560, 590]	3410	0.125	1943	0.358	0.233	1.051	0.245	0.698
2	(590, 620]	15020	0.552	2356	0.435	-0.117	-0.239	0.028	0.698
3	(620, 650]	7813	0.287	621	0.115	-0.172	-0.919	0.158	0.698
4	(650, 680]	850	0.031	50	0.009	-0.022	-1.220	0.027	0.698
5	(680, 710]	17	0.001	1	0.000	-0.000	-1.220	0.001	0.698

3）变量稳定度分析

做这个的目的是分析是哪些变量导致了评分卡的不稳定，变量的稳定性用变量的分布差距来衡量。变量的分布差距的绝对值越大，说明变量越不稳定，反之绝对值越小，说明变量越稳定。对于不稳定的变量，需要深入分析背后的原因，并考虑是否弃用该变量。

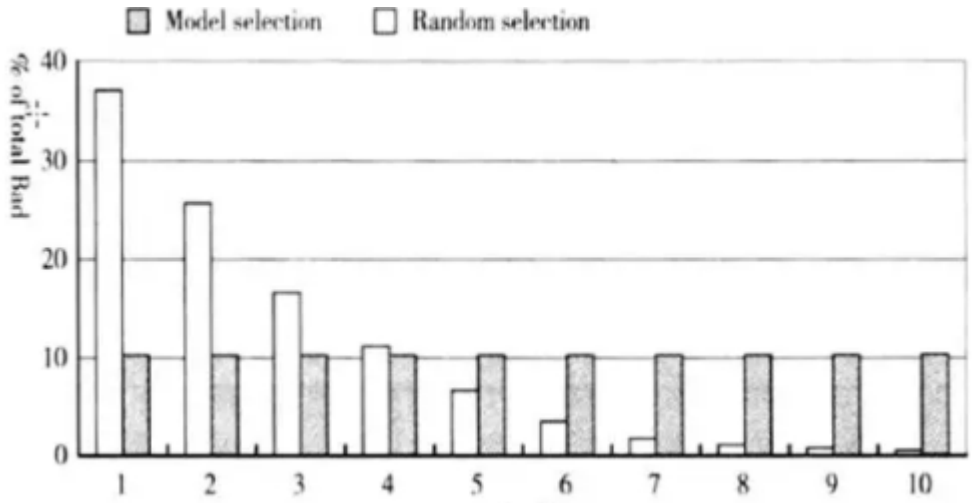
PS：变量的稳定性其实在建模之前就需要分析，如果在建模之前这个变量比较稳定，那么上线之后如果没有什么大的外界影响，变量的稳定程度其实不会发生大的改变。
变量的稳定度分析表如下图所示：

	区间	得分	建模用户数	建模用户占比	上线用户数	上线用户占比	建模样本权重	上线样本权重	权重差距
0	(-inf, 40.0]	13	4324	0.159	820	0.151	2.065	1.961	-0.104
1	(40.0, 110.0]	3	17413	0.640	2669	0.491	1.919	1.473	-0.446
2	(110.0, inf]	-18	5482	0.201	1946	0.358	3.625	6.445	2.820

五.可解释性

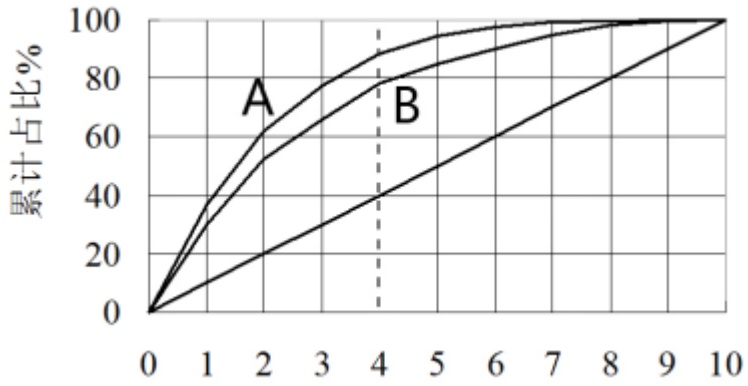
1.**提升图&洛伦兹曲线**
假设目前有10000个样本，坏用户占比为30%，我们做了一个评分卡（分数越低，用户坏的概率越高），按照评分从低到高划分成10等份（每个等份用户数为1000），计算每等份的坏用户占比，如果评分卡效果很好，那么越靠前的等份里，包含的坏用户应该越多，越靠后的等份里，包含的坏用户应该要更少。作为对比，如果不对用户评分，按照总体坏用户占比30%来算，每个等份中坏用户占比也是30%。将这两种方法的每等份坏用户占比放在一张柱状图上进行对比，就是提升图（如下图）。

依据业务上的可解释性，随着分数增大，每等份的坏用户占比应该是逐渐递减的。实际情况中可能不会像这样严格递减变化，但只要没有大的偏差，还是可以接受的。



将这两种方法的累计坏用户占比放在一张曲线图上，就是洛伦兹曲线图（如下图）。
洛伦兹曲线可以比较两个评分卡的优劣，例如下图中虚线对应的分数假设是

600分，那么在600分这cutoff点下，A和B的拒绝率都是40%，但A可以拒绝掉88%的坏用户，B只能拒掉78%的坏用户，说明A评分卡的效果更好。



2.**WOE分析**

1）WOE变化趋势

这个趋势变化主要针对连续数值型变量，假设WOE的计算方式是bad/good，某个变量按照业务理解是值越大，坏用户概率越大。那么变量分箱后，WOE的变化趋势应该与实际的业务经验一致，也就是变量的值越大，WOE越大，且呈单调性变化。对于WOE没有呈单调性变化的变量（例如U型或倒U型），如果业务上能解释的通，那也可以采用该变量。WOE呈波浪形变化的变量建议不采用。

PS：在做WOE趋势分析时不考虑缺失的箱体，并且最好将变量分为4-8箱。

2）箱体之间WOE的差异分析

分箱的原则是组内差异小，组间差异大，所以箱体之间的WOE要有显著差异，个人认为WOE的差值至少要在0.1以上，这样每个箱体的好坏比才有区别。另外我觉得WOE最好不要出现跃阶式变化，例如第一个箱体的woe是0.1，下一个箱体WOE直接增大到0.9，这样会导致最后转化的分数也会呈跃阶式变化，这个对总体评分的分布及稳定性会有很大影响。箱体的WOE最好是单调线性变化的。

3）箱体的WOE绝对值大小

箱体的WOE值最好是在-1至1之间，如果WOE的绝对值大于1，说明这个箱体的坏用户占比或者好用户占比在65%以上，这种变量适合做单条策略，如果放到模型中，这个变量的权重可能会很大，会增加模型过拟合的危险，并会影响评分卡的稳定性。

3.**SCORE分析**

score就是每个变量各个区间对应的得分，根据score的计算公式（score=系数xWOEx刻度B），score的值和WOE的关系是很大的，所以和WOE分析类似，score也要分析它的趋势变化，箱体间score的差异，以及score的绝对值

大小。分析逻辑与WOE大致相同。这里就不再说明。一般来说WOE的可解释性较好，score应该不会有大的问题。

六. 评估流程

目前我整理的这些评估方法是贯穿整个评分卡建模流程的，这里我按顺序分为5个部分：

1.**评分卡建模之前的评估**：

主要评估建模样本的稳定性，根据评分卡的目的不同，比较对象为总体或者近段时间的样本。

2.**分箱过程的评估**

变量分箱的同时会计算WOE，这里是对WOE进行可解释性上的评估，包括变化趋势，箱体之间WOE差异，WOE绝对值大小等。

3.**对逻辑回归模型的评估**

1）将数据集随机划分为训练集和测试集，计算AUC, KS及Gini系数

2）通过交叉验证的方法，评估模型的泛化能力，评判指标选择AUC。

3）绘制学习曲线，评估模型是否有过拟合的风险，评判指标为准确率（Accuracy）。

4.**转化评分之后的评估**

1）对score进行可解释上的评估，评估原则与WOE评估大致相同。

2）绘制评分分布图，观察分布的形状及好坏用户分布的重叠程度。

3）绘制提升图和洛伦兹曲线，评估评分卡的可解释性和好坏用户区分效果。

4）评估准确性，根据对精确率和查全率的重视程度绘制PR曲线，并根据业务目标设定cutoff点。

5.**评分卡上线后的评估**

1）绘制评分分布表和评分分布图，计算评分的PSI，评估其稳定性。

2）评估每个入模变量的稳定性。