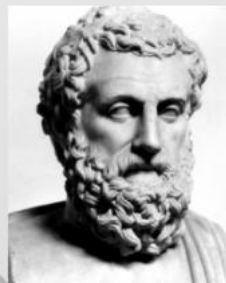


AI

评分卡模型 在银行风控中的应用

门徒计划——开班预热课

授课老师：陈旻



Self-introduction

● 陈旻

● 和计算机、算法相关

(10岁, 清华计算机博士, NOI, ACM比赛, 阿里云MVP, 腾讯云TVP, 百度比赛教练, ACM, IEEE, 中国人工智能协会, CCF专委)

● 和企业培训、企业服务相关

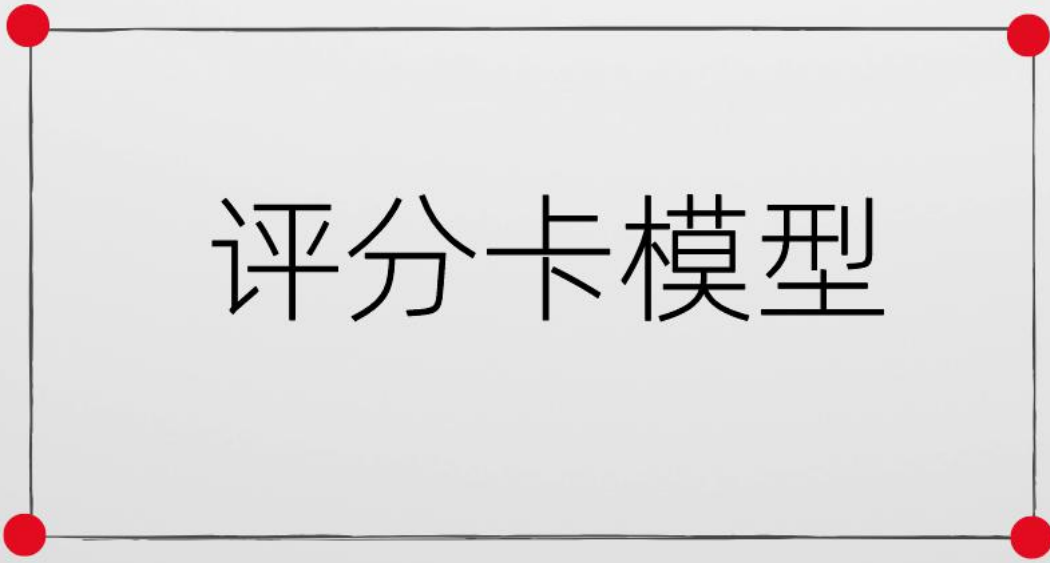
(专栏付费人数超过4.1万, 企业客户包括: 腾讯视频, 易车, 汽车之家, 京东, 蚂蚁金服, 美的, 中国银联, 上汽大众, 中原银行等)



>>今天的学习目标

评分卡模型

- Logistics Regression
- 评分卡模型
- WOE, IV
- 变量分箱
- 缺失值处理
- 区分度评估指标KS
- 平稳性评估指标PSI



评分卡模型

逻辑回归模型

逻辑回归的假设：

- 任何的模型都有自己的假设，在假设条件下才是适用的
- 假设1：数据服从伯努利分布

典型例子：连续的掷n次硬币（每次实验结果不受其他实验结果的影响，即n次实验是相互独立的）

贝努力分布为离散型概率分布，如果成功，随机变量取值为1；如果失败，随机变量取值为0。成功概率记为p，失败概率为 $q=1-p$

$$f_X(x) = p^x(1-p)^{1-x} = \begin{cases} p & \text{if } x = 1 \\ q & \text{if } x = 0 \end{cases}$$

对应二分类问题，样本为正类的概率p，和样本为负类的概率 $q=1-p$

$$p = h_{\theta}(x; \theta)$$

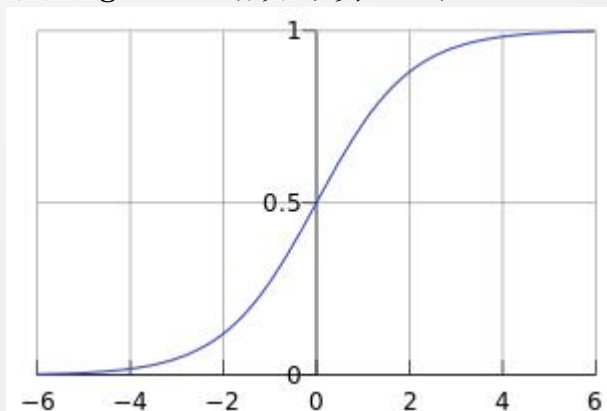
$$q = 1 - h_{\theta}(x; \theta)$$

- 假设2：正类的概率由sigmoid函数计算，即

$$g(z) = \frac{1}{1 + e^{-z}}$$



$$p = \frac{1}{1 + e^{-\theta^T x}}$$



预测样本为正的

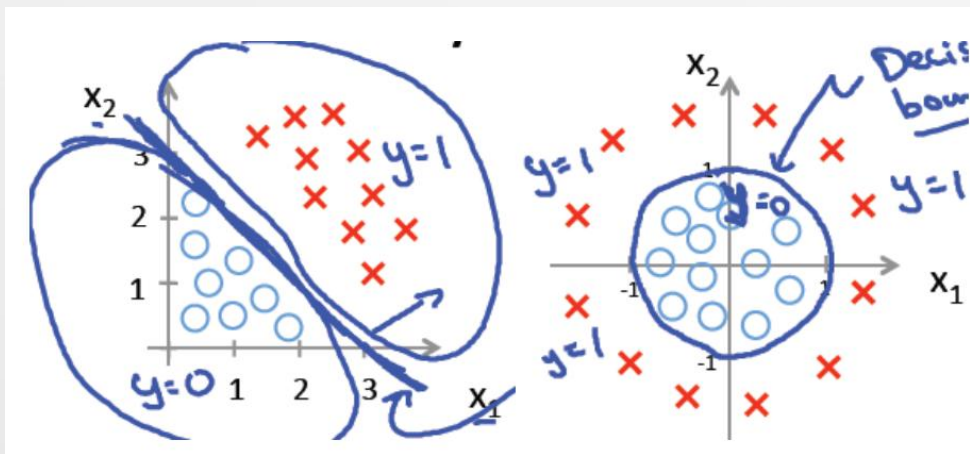
预测样本为负的

$$p(y = 1|x; \theta) = h_{\theta}(x; \theta) = \frac{1}{1 + e^{-\theta^T x}}$$

$$p(y = 0|x; \theta) = 1 - h_{\theta}(x; \theta) = \frac{1}{1 + e^{\theta^T x}}$$

逻辑回归模型

Thinking: 决策边界是线性 or 非线性?



对于线性的决策边界: $\theta_0 + \theta_1 x_1 + \dots + \theta_n x_n = \sum_{i=1}^n \theta_i x_i = \theta^T x$

构造预测函数 $h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$

$h_\theta(x)$ 表示结果为1的概率, 对于输入x, 分类结果为1和0的概率为

$$P(y=1 | x; \theta) = h_\theta(x)$$

$$P(y=0 | x; \theta) = 1 - h_\theta(x)$$

公式(1)

- 构造损失函数

$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

$$J(\theta) = \frac{1}{m} \sum_{i=1}^n Cost(h_\theta(x^{(i)}), y^{(i)}) = -\frac{1}{m} \sum_{i=1}^n (y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})))$$

Cost函数与J函数是基于最大似然估计推导得到的

将公式1综合起来: $P(y | x; \theta) = (h_\theta(x))^y (1 - h_\theta(x))^{1-y}$

取似然函数为 $L(\theta) = \prod_{i=1}^m P(y^{(i)} | x^{(i)}; \theta) = \prod_{i=1}^m (h_\theta(x^{(i)}))^{y^{(i)}} (1 - h_\theta(x^{(i)}))^{1-y^{(i)}}$

对数似然函数为

$$l(\theta) = \log L(\theta) = \sum_{i=1}^m (y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})))$$

最大似然估计就是求使 $l(\theta)$ 取最大值时的 θ , 这里可以使用梯度上升法求解, 也即对 $J(\theta)$ 使用梯度下降法求最小值

$$J(\theta) = -\frac{1}{m} l(\theta)$$

逻辑回归模型

似然函数:

- 关于统计模型中的参数的函数，表示模型参数中的似然性
- 给定输出 x 时，关于参数 θ 的似然函数 $L(\theta | x)$ 等于给定参数 θ 后变量 X 的概率

$$L(\theta|x) = P(X = x|\theta)$$

逻辑回归:

- 如何进行分类：设定一个阈值，判断正类概率是否大于该阈值，一般阈值是0.5，所以只用判断正类概率是否大于0.5即可
- **Thinking:** 为什么会在训练中将高度相关的特征去掉？
 - 1) 可解释性更好
 - 2) 提高训练的速度，特征多了，会增大训练的时间

逻辑回归模型

优点:

- 形式简单, 模型的可解释性非常好
- 根据特征的权重可以得到不同的特征对最后结果的影响 (某个特征的权重值高 => 这个特征对结果的影响大)
- 工程上的**baseline**, 如果特征工程做的好, 效果不会差
- 训练速度较快, 计算量只和特征的数目相关
- 模型资源占用小, 只需要存储各个维度的特征值
- 方便输出结果调整。逻辑回归可以很方便的得到最后的分类结果, 因为输出的是每个样本的概率分数, 我们可以很容易的对这些概率分数进行**cut off**, 也就是划分阈值(大于某个阈值的是一类, 小于某个阈值的是一类)

缺点:

- 准确率不是很高, 形式简单, 很难拟合数据的真实分布
- 很难处理样本不均衡问题
- 很难处理非线性数据, 在不引入其他方法的情况下, 只能处理线性可分的数据
- 逻辑回归本身无法筛选特征, 可以采用**gbd**t来筛选特征, 然后再用逻辑回归

逻辑回归模型

逻辑回归：

- 假设数据服从伯努利分布
- 通过极大化似然函数的方法
- 运用梯度下降来求解参数
- 将数据进行二分类

评分卡模型

评分卡模型：

评分卡模型是常用的金融风控手段之一

风控，就是风险控制，我们采取各种措施和方法，减少风险发生的可能性，或风险发生时造成的损失

根据客户的各种属性和行为数据，利用信用评分模型，对客户的信用进行评分，从而决定是否给予授信，授信的额度和利率，减少在金融交易中存在的交易风险

按照不同的业务阶段，可以划分为三种：

贷前：申请评分卡（Application score card），称为A卡

贷中：行为评分卡（Behavior score card），称为B卡

贷后：催收评分卡（Collection score card），称为C卡

（借款人当前还款状态为逾期的情况下，未来坏账的概率）

评分卡模型



评分卡模型：

- 客户评分 = 基准分 + 年龄评分 + 性别评分 + 婚姻状况评分 + 学历评分 + 月收入评分

Thinking: 某客户年龄为27岁，性别为男，婚姻状况为已婚，学历为本科，月收入为16000，那么他的评分=？

$223(\text{基准分}) + 8(\text{年龄评分}) + 4(\text{性别评分}) + 8(\text{婚姻评分}) + 8(\text{学历评分}) + 13(\text{收入评分}) = 264$

Thinking: 评分卡的最高分和最低分是多少？

最低分: $223-2+2-2-8-8=205$

最高分: $223+10+4+8+12+20=277$

变量名称	变量范围	得分
基准分	-	223
年龄	$18 \leq \text{age} < 25$	-2
	$25 \leq \text{age} < 35$	8
	$35 \leq \text{age} < 55$	10
	$55 \leq \text{age}$	5
性别	男	4
	女	2
婚姻状况	已婚	8
	未婚	-2
学历	硕士、博士	12
	本科	8
	大专	1
	中专，技校，高中	-3
	初中，小学	-8
月收入	月收入 < 6000	-8
	$6000 \leq \text{月收入} < 10000$	0
	$10000 \leq \text{月收入} < 15000$	5
	$15000 \leq \text{月收入} \leq 30000$	13
	$30000 \leq \text{月收入}$	20

评分卡模型

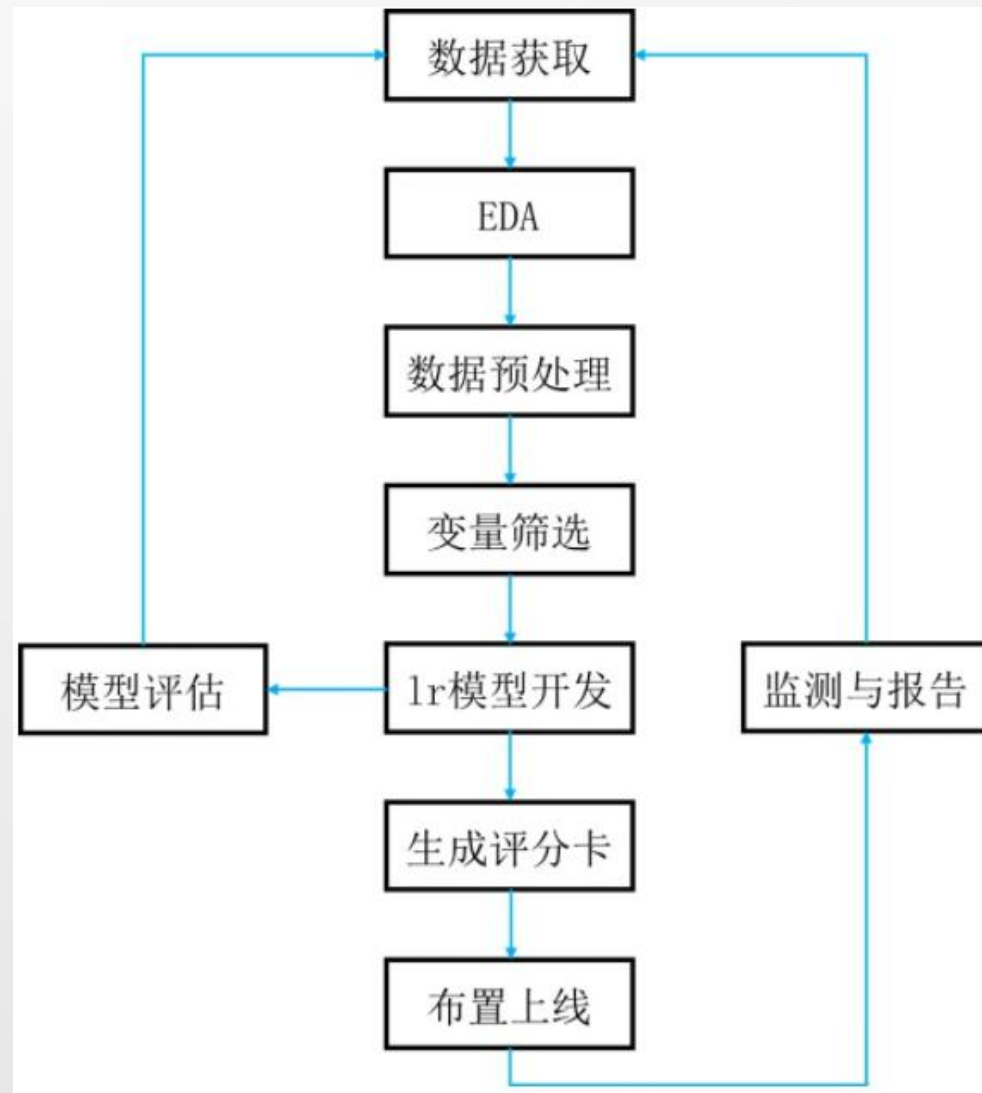
评分卡模型：

- 评分卡模型使用的字段属性通常不超过30个，但是可以使用的属性有很多，如何挑选这些字段？
- 评分卡模型是基于每个字段的分段进行的评分，那么该如何对这些字段进行有效的分段及评分？

评分卡模型

评分卡模型开发步骤：

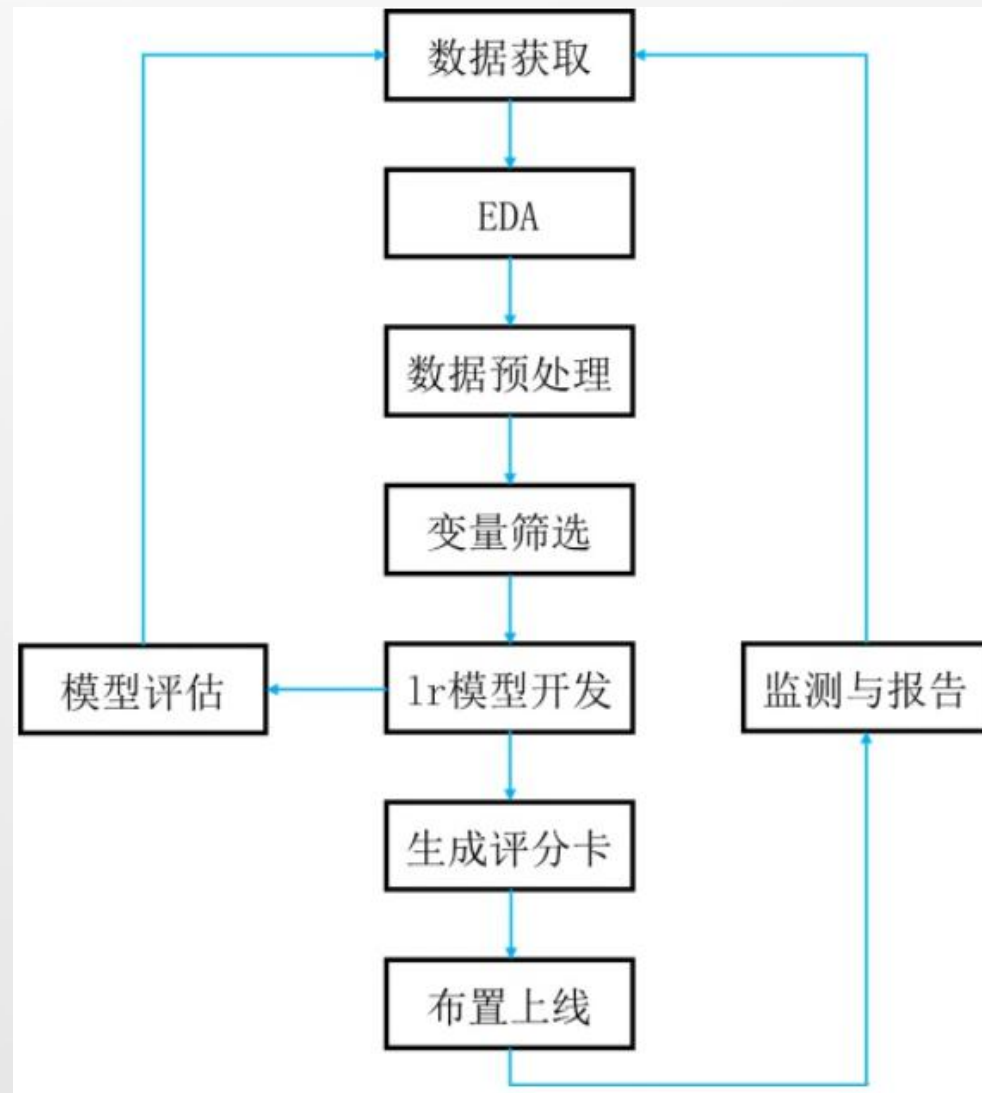
- Step1, 数据获取, 包括获取存量客户及潜在客户的数据
存量客户, 已开展融资业务的客户, 包括个人客户和机构客户;
潜在客户, 将要开展业务的客户
- Step2, EDA, 获取样本整体情况, 进行直方图、箱形图可视化
- Step3, 数据预处理, 包括数据清洗、缺失值处理、异常值处理
- Step4, 变量筛选, 通过统计学的方法, 筛选出对违约状态影响最显著的指标。主要有单变量特征选择和基于机器学习的方法



评分卡模型

评分卡模型开发步骤：

- Step5, 模型开发, 包括变量分段、变量的WOE（证据权重）变换和逻辑回归估算三个部分
- Step6, 模型评估, 评估模型的区分能力、预测能力、稳定性, 并形成模型评估报告, 得出模型是否可以使用的结论
- Step7, 生成评分卡（信用评分）, 根据逻辑回归的系数和WOE等确定信用评分的方法, 将Logistic模型转换为标准评分的形式
- Step8, 建立评分系统（布置上线）, 根据生成的评分卡, 建立自动信用评分系统



评分卡模型

WOE编码:

- Weight of Evidence, 证据权重
- 是自变量的一种编码, 常用于特征变换用来衡量自变量与因变量的相关性

$$woe_i = \ln \left(\frac{\text{Event}\%}{\text{Not Event}\%} \right) = \ln \left(\frac{\#B_i / \#B_T}{\#G_i / \#G_T} \right)$$

B代表风险客户, G代表正常客户

对于某一变量某一分组的WOE, 衡量了这组里面的好坏客户的占比与整体样本好坏样本占比的差异

Thinking: 对于二分类问题共100条记录, 一个自变量只有两个值 value1, value2, 如何计算value1, value2对应的woe1, woe2?

value1有50条记录, 其中40条对应label 1, 另外10条对应label 0

value2有50条记录, 其中25条对应label 1, 另外25条对应label 0

记录value	Label=1 个数	Label=0 个数	Label=1 的比率	Label=0 的比率	Woe
Value1	40	10	40/(40+25)= 62%	10/(10+25) =28%	$\ln(62\%/28\%) = \ln(2.2) = 0.79$
Value2	25	25	25/(40+25)= 38%	25/(10+25) =72%	$\ln(38\%/72\%) = \ln(0.52) = -0.64$

Thinking: WOE差异越大, 对风险区分能力=?

差异越大, 对风险区分越明显

评分卡模型

WOE计算:

- 对于连续型变量, 分成N个bins
- 对于分类型变量保持类别group不变
- 计算每个bin or group中event和non-event的百分比

$$woe_i = \ln \left(\frac{\text{Event}\%}{\text{Not Event}\%} \right) = \ln \left(\frac{\#B_i / \#B_T}{\#G_i / \#G_T} \right)$$

WOE的作用:

- 可以将连续型变量转化为woe的分类变量
- 可以对相似的bin或group进行合并 (woe相似)

计算woe需要注意:

- 每个bin or group记录不能过少, 至少有5%的记录
- 不要用过多的bin or group, 会导致不稳定性
- 对bin or group中全为0或者1的特列, 用修正的woe

$$WOE_i = \ln \left(\left(\frac{Bad_i + 0.5}{Good_i + 0.5} \right) / \left(\frac{Bad_T}{Good_T} \right) \right)$$

防止分母为0的情况

评分卡模型

IV (Information Value) :

- woe只考虑了风险区分的能力，没有考虑能区分的用户有多少
- IV衡量一个变量的风险区分能力, 即衡量各变量对y的预测能力，用于筛选变量

$$IV_i = (\text{Event\%} - \text{Nbt Event\%}) * \ln\left(\frac{\text{Event\%}}{\text{Nbt Event\%}}\right)$$
$$= \left(\frac{\#B_i}{\#B_T} - \frac{\#G_i}{\#G_T}\right) * \ln\left(\frac{\#B_i / \#B_T}{\#G_i / \#G_T}\right)$$

$$IV = \sum_{k=0}^n IV_i$$



IV的计算，可以认为是WOE的加权和

IV是与WOE密切相关的一个指标，在应用实践中，评价标准可参考如下：

IV范围	变量评估（预测效果）
小于0.02	几乎没有
0.02~0.1	弱
0.1~0.3	中等
0.3~0.5	强
大于0.5	难以置信，需要确认

Thinking: 如何使用IV值进行特征变量的筛选？

比如筛选掉IV < 0.1的变量，因为该特征对于y的预测能力很弱

评分卡模型

WOE和IV计算步骤:

- Step1, 对于连续型变量, 进行分箱 (binning), 可以选择等频、等距, 或者自定义间隔, 对于离散型变量, 如果分箱太多, 则进行分箱合并
- Step2, 统计每个分箱里的好人数(bin_goods)和坏人数(bin_bads)
- Step3, 分别除以总的好人数(total_goods)和坏人数(total_bads), 得到每个分箱内的边际好人占比margin_good_rate和边际坏人比margin_bad_rate
- Step4, 计算每个分箱的WOE

$$WOE = \ln \left(\frac{\text{margin_badrate}}{\text{margin_goodrate}} \right)$$

- Step5, 检查每个分箱 (除null分箱外) 里WOE值是否满足单调性, 若不满足, 返回step1

说明: null分箱由于有明确的业务解释, 因此不需要考虑满足单调性

- Step6, 计算每个分箱里的IV, 最终求和, 即得到最终的IV

评分卡模型



Thinking: 如何计算每个bucket中的WOE和IV?

margin_bad_rate = bad/total_bads

WOE=ln (margin_bad_rate/margin_good_rate)

margin_good_rate = good/total_goods

IV=(bad/total_bads - good/total_goods)*WOE

bucket	min_score	max_score	obs	bad	good	bad_rate	good_rate	margin_bad_rate	margin_good_rate	odds (bad/good)	woe	IV
1	0	18	1390	70	1320							
2	18	23	1070	33	1037							
3	23	28	1162	20	1142							
4	28	34	1162	15	1147							
5	34	44	1212	12	1200							
6	44	100	1153	9	1144							
7	null	null	1775	17	1758							
总计	0	100	8924	176	8748							

评分卡模型



计算每个分箱里的WOE和IV

bucket	min_score	max_score	obs	bad	good	bad_rate	good_rate	margin_bad_rate	margin_good_rate	odds (bad/good)	woe	IV
1	0	18	1390	70	1320	5.04%	94.96%	39.77%	15.09%	0.053030303	0.969204613	0.239234241
2	18	23	1070	33	1037	3.08%	96.92%	18.75%	11.85%	0.031822565	0.45851674	0.031618681
3	23	28	1162	20	1142	1.72%	98.28%	11.36%	13.05%	0.017513135	-0.13870773	0.002345237
4	28	34	1162	15	1147	1.29%	98.71%	8.52%	13.11%	0.013077594	-0.430758529	0.019766824
5	34	44	1212	12	1200	0.99%	99.01%	6.82%	13.72%	0.01	-0.699073799	0.048230774
6	44	100	1153	9	1144	0.78%	99.22%	5.11%	13.08%	0.007867133	-0.938965208	0.074775794
7	null	null	1775	17	1758	0.96%	99.04%	9.66%	20.10%	0.00967008	-0.732622347	0.076463289
总计	0	100	8924	176	8748	1.97%	98.03%	100.00%	100.00%	0.020118884	0	0.492434842

评分卡模型



WOE编码计算:

- 假设，我们对Age字段，计算相关的woe

Step1，首先对每个level进行分层统计

Step2，计算每层的好坏占比

Step3，通过好坏占比 => 计算WOE

Age	bad (Y=1)	good (Y=0)
Age1 (0-10)	50	200
Age2 (10-18)	20	200
Age3 (18-35)	5	200
Age4 (35-50)	15	200
Age5 (>50)	10	200
Total	100	1000



Age	bad (Y=1)	good (Y=0)	bad%	good%	woe= $\ln(\text{bad\%}/\text{good\%})$
Age1 (0-10)	50	200	50%	20%	$\ln(50\%/20\%)$
Age2 (10-18)	20	200	20%	20%	$\ln(20\%/20\%)$
Age3 (18-35)	5	200	5%	20%	$\ln(5\%/20\%)$
Age4 (35-50)	15	200	15%	20%	$\ln(15\%/20\%)$
Age5 (>50)	10	200	10%	20%	$\ln(10\%/20\%)$
Total	100	1000	1	1	

Project: 基于评分卡的风控模型开发



Project: 基于评分卡的风控模型开发

- 数据集GiveMeSomeCredit, 15万样本数据

<https://www.kaggle.com/c/GiveMeSomeCredit/data>

- 基本属性: 包括了借款人当时的年龄
- 偿债能力: 包括了借款人的月收入、负债比率
- 信用往来: 两年内35-59天逾期次数、两年内60-89天逾期次数、两年内90天或高于90天逾期的次数
- 财产状况: 包括了开放式信贷和贷款数量、不动产贷款或额度数量。
- 其他因素: 包括了借款人的家属数量
- 时间窗口: 自变量的观察窗口为过去两年, 因变量表现窗口为未来两年

字段	说明	类型
SeriousDlqin2yrs	90天以上逾期或更差	Y/N
Age	年龄	整数
RevolvingUtilizationOfUnsecuredLines	除房地产和汽车贷款等无分期付款债务外, 信用卡和个人信用额度的总余额除以信贷限额	百分比
DebtRatio	债务比 (每月偿还的债务, 赡养费, 生活费除以每月的总收入)	百分比
MonthlyIncome	每月收入	实数
NumberOfOpenCreditLinesAndLoans	公开贷款(如汽车贷款或抵押贷款)和信用额度(如信用卡)的数量	整数
NumberRealEstateLoansOrLines	抵押贷款和房地产贷款的额度 (包括房屋净值信贷)	整数
NumberOfTime30-59DaysPastDueNotWorse	借款人逾期30-59天的次数, 但在过去两年没有更糟	整数
NumberOfTime60-89DaysPastDueNotWorse	借款人逾期60-89天的次数, 但在过去两年没有更糟	整数
NumberOfTimes90DaysLate	借款人逾期90天 (或以上) 的次数	整数
NumberOfDependents	除自己(配偶、子女等)以外的家庭受养人人数	整数

Project: 基于评分卡的风控模型开发



Project 基于评分卡的风控模型开发:

- Step1, 数据探索性分析

违约率分析

缺失值分析

对于某个字段的统计分析 (比如

RevolvingUtilizationOfUnsecuredLines)

- Step2, 数据缺失值填充, 采用简单规则, 如使用中位数进行填充
- Step3, 变量分箱

1) 对于age字段, 分成6段 $[-\text{math.inf}, 25, 40, 50, 60, 70, \text{math.inf}]$

2) 对于NumberOfDependents (家属人数) 字段, 分成6段

$[-\text{math.inf}, 2, 4, 6, 8, 10, \text{math.inf}]$

3) 对于3种逾期次数, 即NumberOfTime30-

59DaysPastDueNotWorse, NumberOfTime60-

89DaysPastDueNotWorse, NumberOfTimes90DaysLate, 分成10段

$[-\text{math.inf}, 1, 2, 3, 4, 5, 6, 7, 8, 9, \text{math.inf}]$

4) 对于其余字段, 即

RevolvingUtilizationOfUnsecuredLines, DebtRatio,

MonthlyIncome, NumberOfOpenCreditLinesAndLoans,

NumberRealEstateLoansOrLines 分成5段

Project: 基于评分卡的风控模型开发



Project 基于评分卡的风控模型开发:

- Step4, 特征筛选

使用IV值衡量自变量的预测能力, 筛选IV值>0.1的特征字段

- Step5, 对于筛选出来的特征, 计算每个bin的WOE值

- Step6, 使用逻辑回归进行建模

训练集、测试集切分

计算LR的准确率

features	bin	woe
RevolvingUtilizationOfUnsecuredLines	(0.699, 50708.0]	3.463412
RevolvingUtilizationOfUnsecuredLines	(0.271, 0.699]	1.054603
RevolvingUtilizationOfUnsecuredLines	(0.0832, 0.271]	0.420420
RevolvingUtilizationOfUnsecuredLines	(-0.001, 0.0192]	0.276204
RevolvingUtilizationOfUnsecuredLines	(0.0192, 0.0832]	0.235185
NumberOfTime30-59DaysPastDueNotWorse	(1.0, 2.0]	5.036574
NumberOfTime30-59DaysPastDueNotWorse	(-inf, 1.0]	0.772730
NumberOfTime30-59DaysPastDueNotWorse	(2.0, 3.0]	7.595036
.....

Project: 基于评分卡的风控模型开发



- Step7, 评分卡模型转换

设 p 为客户违约的概率, 那么正常的概率为 $1-p$

$$Odds = \frac{p}{1-p}$$

客户违约概率 p 可以表示

$$p = \frac{Odds}{1 + Odds}$$

评分卡的分值计算, 可以通过 分值表示为比率对数的 线性表达式来定义, 即

$$Score = A - B * \ln(Odds)$$

Score计算公式类似 $y=kx+b$, A 和 B 是常数, A 称为“补偿”, B 称为“刻度”, 公式中的负号可以使得违约概率越低, 得分越高

常数 A 、 B 可以通过将两个假设的分值带入计算得到:

1) 基准分, 即给某个特定的比率 时, 预期的分值为

通常, 业内的基准分为500/600/650

2) PDO (point of double odds), 即比率翻倍的分数

比如, odds翻倍时, 分值减少50

比率为 的点的分值应该为

代入式中, 可以得到:

求解得:

Project: 基于评分卡的风控模型开发



- Step7, 评分卡模型转换

假设odds=1的时候, 特定的分数为650分

Thinking A 和 B=?

- Step7, 评分卡模型转换

逻辑回归:
$$p = \frac{1}{1 + e^{-\theta^T x}}$$

将公式变化下, 可得

$$\ln\left(\frac{p}{1-p}\right) = \theta^T x, \text{ 即 } \ln(odds) = \theta^T x$$

所以, Odds可以和逻辑回归无缝结合

评分卡的逻辑是Odds的变动与评分变动的映射, 即把Odds映射为评分

因为

所以
$$Score = A - B\{\theta_0 + \theta_1 x_1 + \dots + \theta_n x_n\}$$

Project: 基于评分卡的风控模型开发



- qcut使用

使用qcut可以对一组数据分成几个区间

比如，我们有11家公司，他们的年销售额分别为：

[1000,856,123,523,33,71,223,699,103,456,923]

请你对这11家公司的年销售额进行分箱

1) 按照 高/低，两个等级

2) 按照 first 10%, second 10%, third 10% 以及 last 70% 四个等级

随机销售额

sales =

```
pd.Series([1000,856,123,523,33,71,223,699,103,456,923])
```

```
print(len(sales))
```

将销售额分成 低/高 两个等级

```
print(pd.qcut(sales,[0,0.5,1],labels=['small sales','large sales']))
```

将销售额分成 first 10%, second 10%, third 10% 以及 后 70% 四种等级

```
print(pd.qcut(sales,[0, 0.7, 0.8, 0.9, 1],labels=['last 70%','third 10%','second 10%','first 10%']))
```



Project: 基于评分卡的风控模型开发



- qcut使用

比如，我们有11家公司，他们的年销售额分别为：

[1000,856,123,523,33,71,223,699,103,456,923]

Thinking: 自动将这11家公司的销售额按照5组进行划分

```
print(pd.qcut(sales, q=5))
```

这里q为参数，表示要分组的个数

- qcut与cut的区别

根据数值的频率来选择分箱，使得区间内的频率是均匀的

```
print(pd.qcut(sales, q=5))
```

根据数值本身来选择分箱，使得区间是均匀的间隔

```
print(pd.cut(sales, 5))
```

0	(856.0, 1000.0]	0	(806.6, 1000.0]
1	(523.0, 856.0]	1	(806.6, 1000.0]
2	(103.0, 223.0]	2	(32.033, 226.4]
3	(223.0, 523.0]	3	(419.8, 613.2]
4	(32.999, 103.0]	4	(32.033, 226.4]
5	(32.999, 103.0]	5	(32.033, 226.4]
6	(103.0, 223.0]	6	(32.033, 226.4]
7	(523.0, 856.0]	7	(613.2, 806.6]
8	(32.999, 103.0]	8	(32.033, 226.4]
9	(223.0, 523.0]	9	(419.8, 613.2]
10	(856.0, 1000.0]	10	(806.6, 1000.0]

Project: 基于评分卡的风控模型开发

- `'delimiter'.join(seq)`

通过指定字符连接序列中元素，生成新字符串

```
a = 'abcd'
```

```
print(','.join(a))
```

```
# 结果a,b,c,d
```

```
a = 'abcd'
```

```
print(' '.join(a))
```

```
# 结果a b c d
```

Project: 基于评分卡的风控模型开发



Thinking: 特征分箱（离散）后的优势？

- 变量分箱是对连续变量进行离散化，分箱后的特征对异常数据有很强的鲁棒性

比如 $\text{age} > 30$ 为1，否则0，如果特征没有离散化，杜宇异常数据“年龄300岁”会给模型造成很大的干扰

- 逻辑回归属于广义线性模型，表达能力受限，单变量离散化为N个后，相当于为模型引入了非线性，能够提升模型表达能力
- 离散化后可以进行特征交叉，由M+N个变量变为M*N个变量，进一步引入非线性，提升表达能力
- 可以将缺失作为独立的一类带入模型
- 将所有变量变换到相似的尺度上

缺失值处理

针对字段X，存在缺失值的处理：

- 直接删除含有缺失值的样本
- 如果缺失的样本占总数很大，可以直接舍弃字段X（如果将X作为特征加入，噪音会很大）
- 采用简单规则进行补全

删除：删除数据缺失的记录；

均值：使用当前列的均值；

高频：使用当前列出现频率最高的数据。

- 采用预测进行补全：

根据样本之间的相似性填补缺失值

根据变量之间的相关关系填补缺失值

To Do：采用随机森林对Titanic乘客生存预测中的Embarked, Age进行补全

1) 通过Survived, Pclass, Sex, SibSp, Parch, Fare字段预测Embarked字段中的缺失值

2) 通过Survived, Pclass, Sex, SibSp, Parch, Fare, Embarked字段，预测Age字段中的缺失值

评估指标KS

评估指标KS（Kolmogorov-Smirnov）：

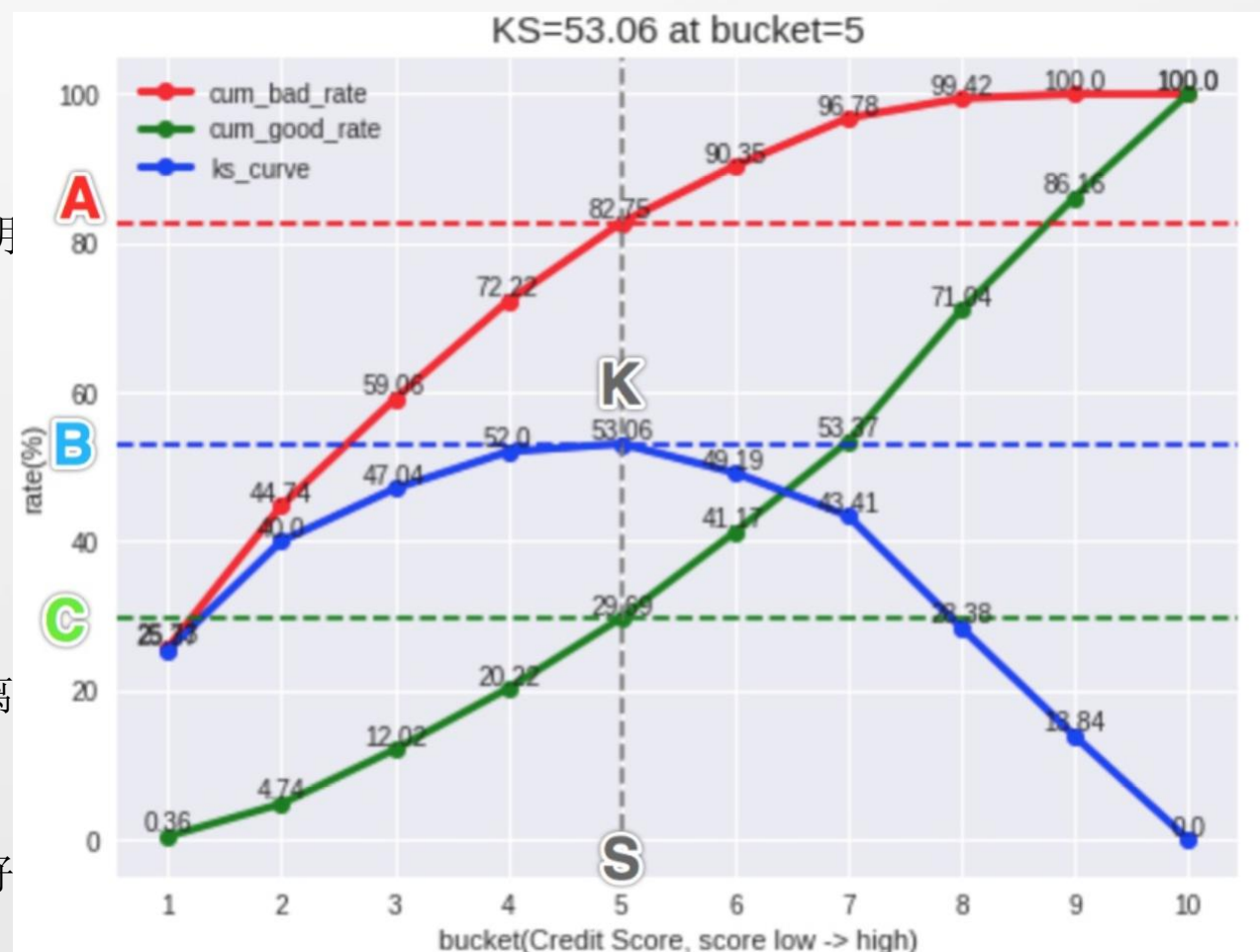
- 由两位苏联数学家A.N. Kolmogorov和N.V. Smirnov提出
- 在风控中，KS常用于评估模型区分度。区分度越大，说明模型的风险排序能力（ranking ability）越强

$$ks = \max\{|cum(bad_rate) - cum(good_rate)|\}$$

KS曲线：计算每个Score分箱区间累计坏账户占比与累计好账户占比差的绝对值

KS值：在这些绝对值中取**最大值**，是衡量好坏客户分数距离的上限值

KS含义：如果排除掉一定比例的坏用户，会有多少比例的好用户会被误杀掉



评估指标KS

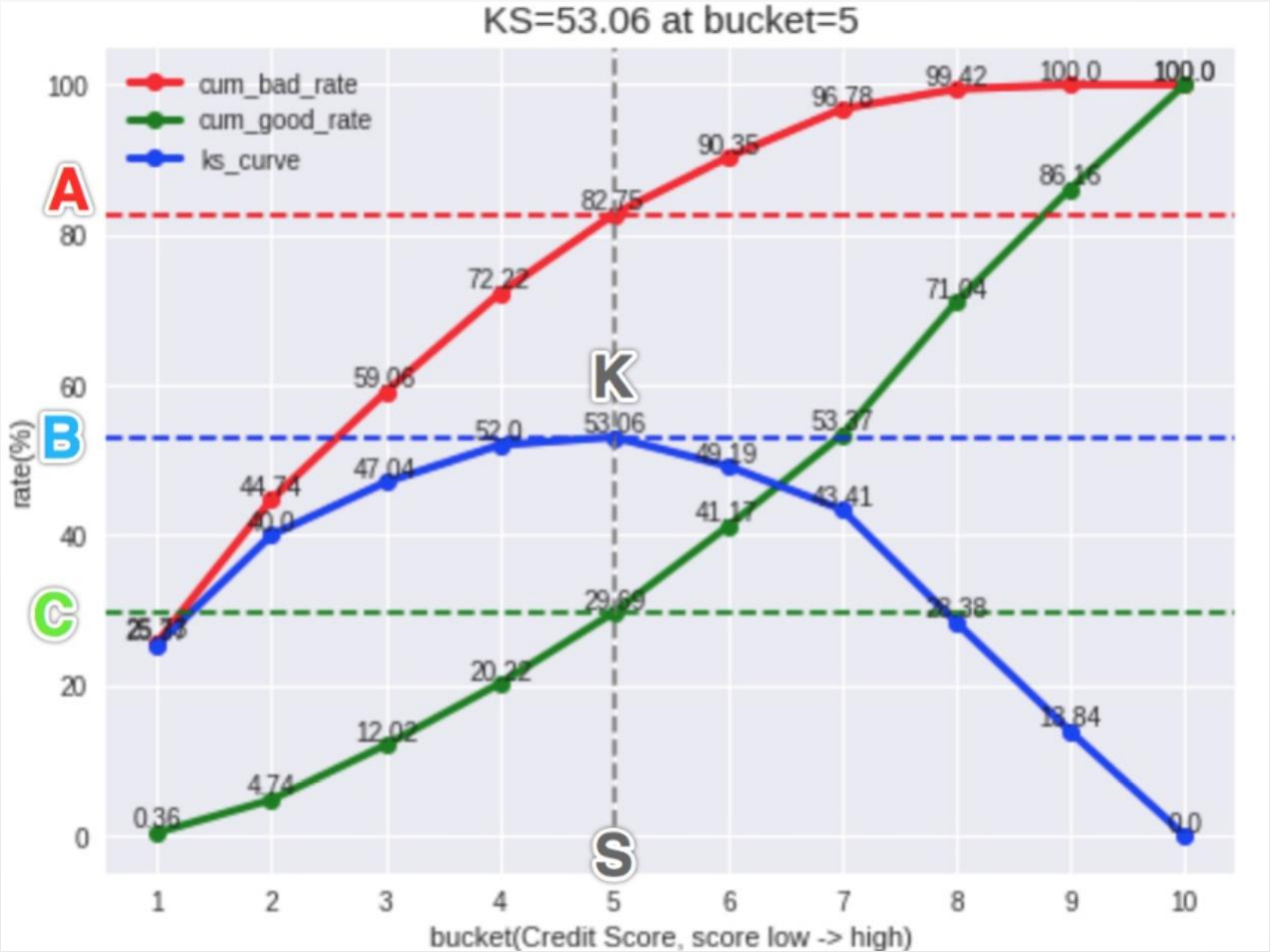


评估指标KS（Kolmogorov-Smirnov）：

- KS统计量是好坏距离或区分度的上限
- KS越大，表明正负样本区分程度越好

KS (%)	好坏区分能力
20以下	不建议采用
20-40	较好
41-50	良好
51-60	很强
61-75	非常强
75以上	不可思议，需要check

KS评价标准



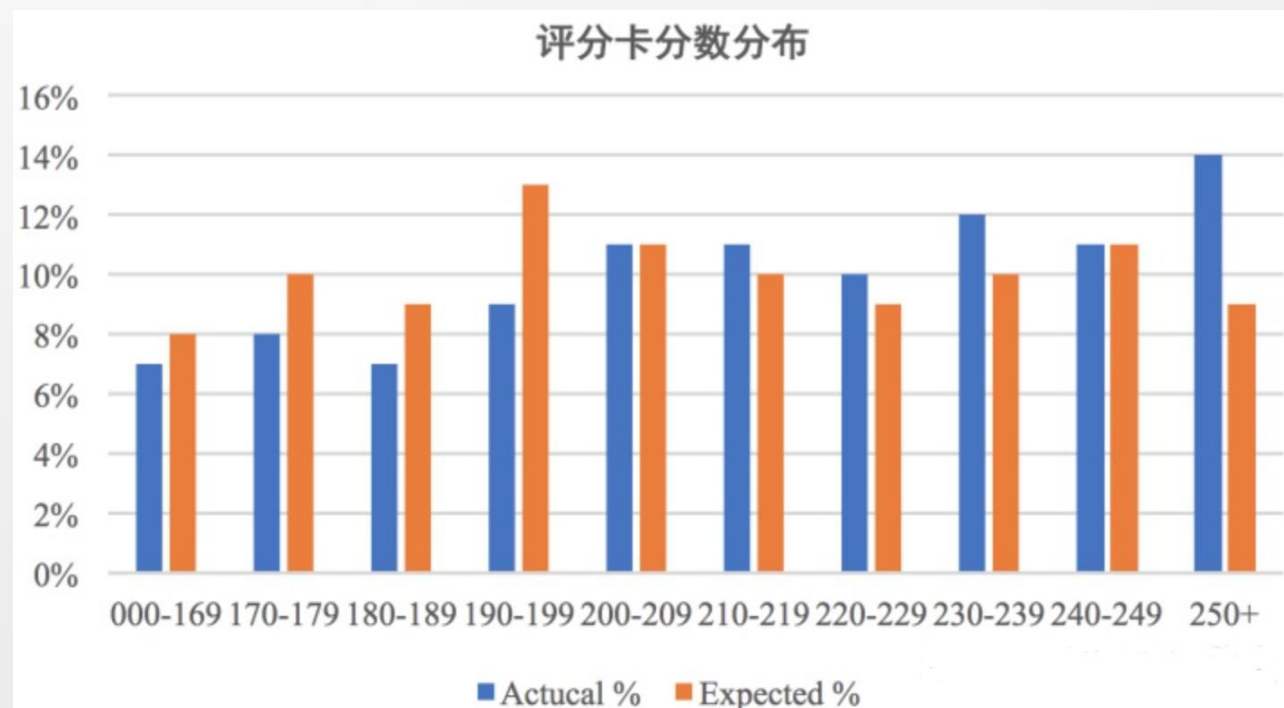
$$ks = \max\{|cum(bad_rate) - cum(good_rate)|\}$$

评估指标PSI

评估指标PSI:

- 群体稳定性指标, Population Stability Index
- 反映了验证样本在各分数段的分布与建模样本分布的稳定性。在建模中, 我们常用来筛选特征变量、评估模型稳定性
- 稳定性是有参照的, 需要有两个分布, 即实际分布 (actual) 与预期分布 (expected)
- 其中, 建模时以训练样本 (In the Sample, INS) 作为预期分布, 而验证样本作为实际分布
- $PSI = \text{SUM}((\text{实际占比} - \text{预期占比}) * \ln(\text{实际占比} / \text{预期占比}))$

$$psi = \sum_{i=1}^n (A_i - E_i) * \ln(A_i / E_i)$$



评估指标PSI



评估指标PSI:

- $PSI = \text{SUM}((\text{实际占比} - \text{预期占比}) * \ln(\text{实际占比} / \text{预期占比}))$

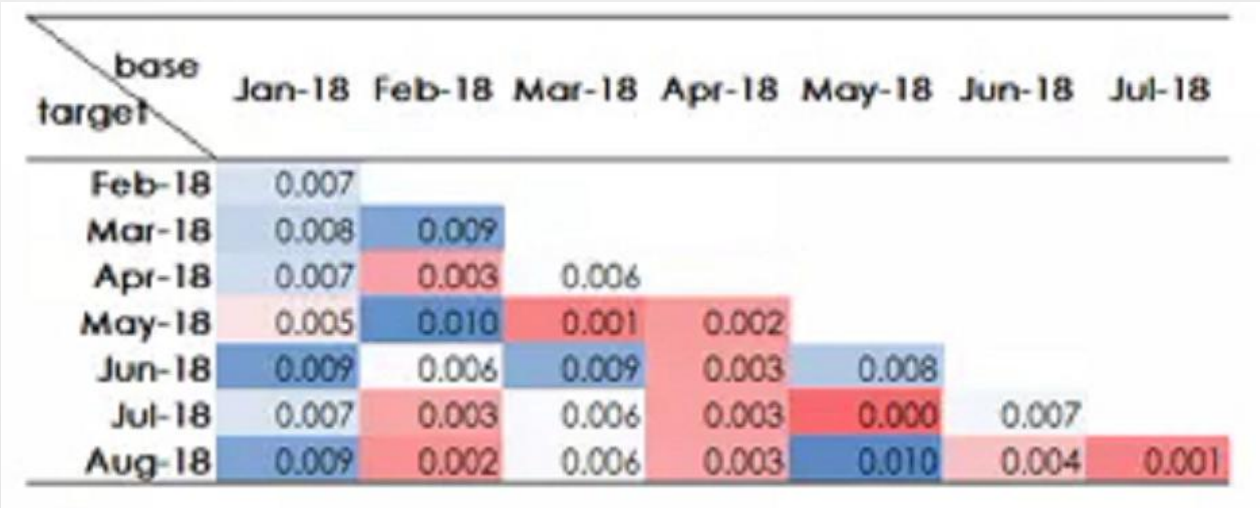
$$psi = \sum_{i=1}^n (A_i - E_i) * \ln(A_i / E_i)$$

- PSI数值越小，两个分布之间的差异就越小 => 越稳定

PSI矩阵:

- 衡量base月份与target月份之间的模型稳定性
- 一般认为 $PSI < 0.1$ 模型是优秀的, $PSI > 0.1$ 不一定有问题, 需要具体分析

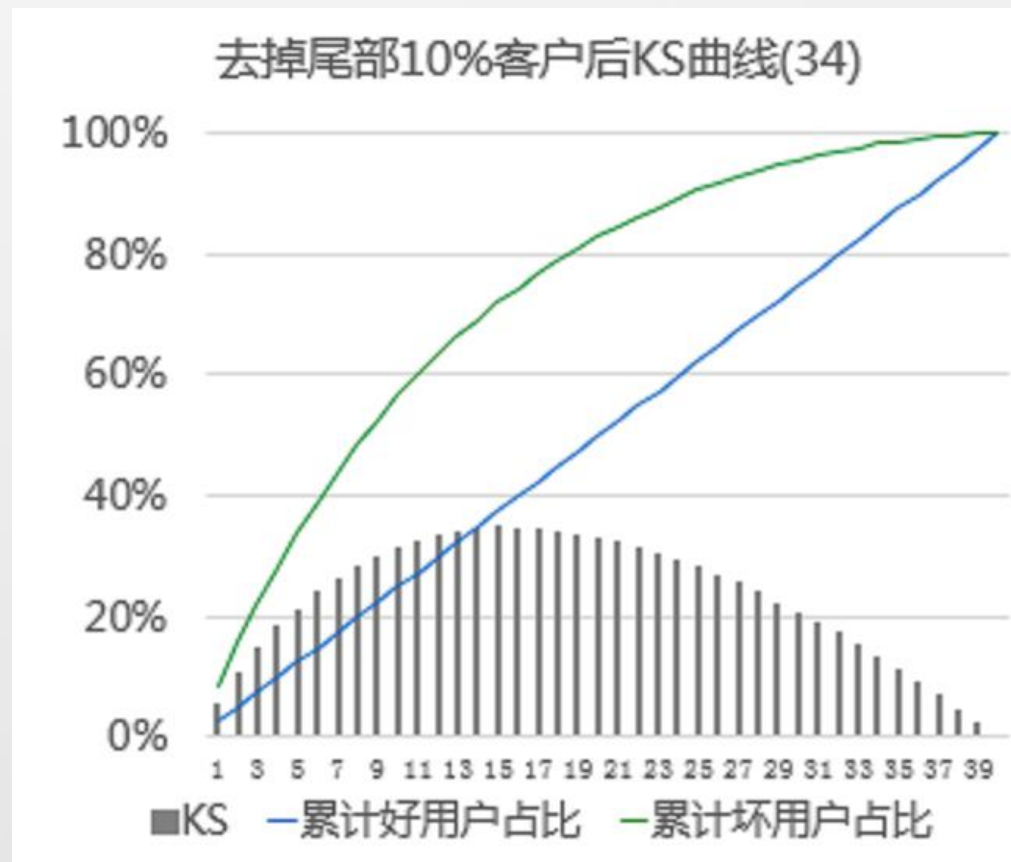
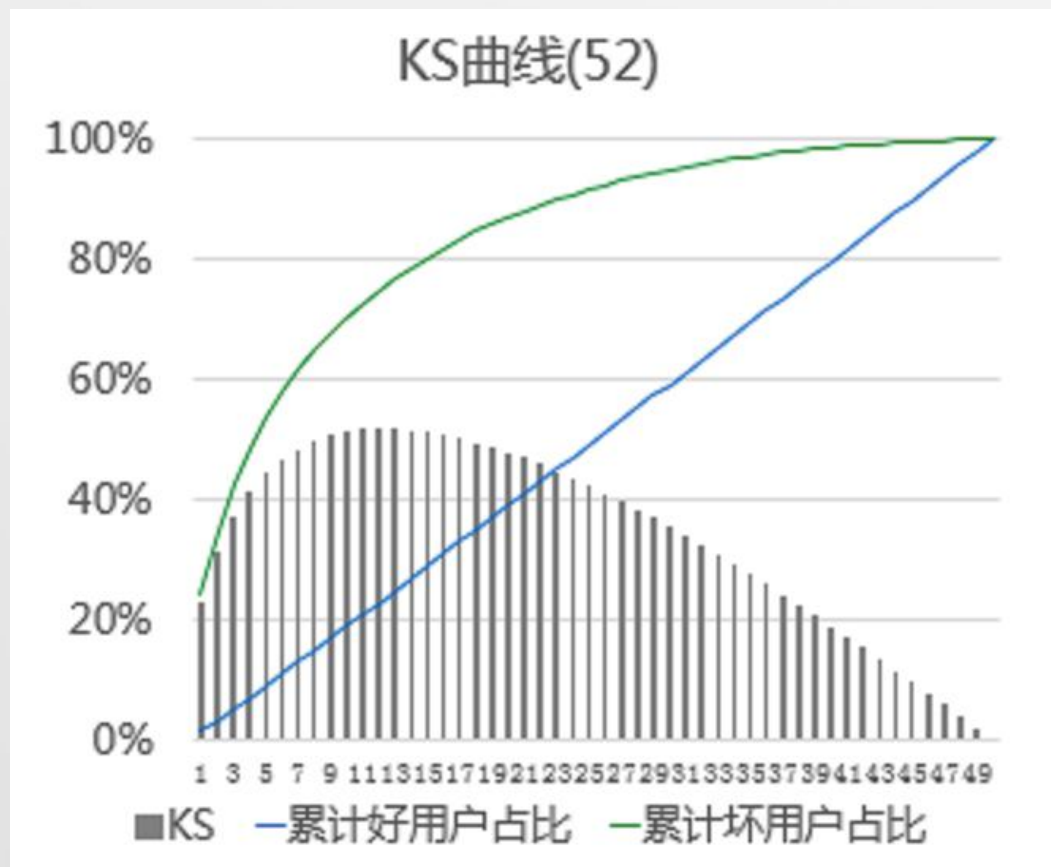
PSI范围	稳定性	建议
0~0.1	好	没有变化或很少变化
0.1~0.25	略不稳定	有变化，继续监控后续变化
大于0.25	不稳定	发生大变化，进行特征项分析



Covariate Shift场景下的模型监控

Thinking: Offline KS=52, Online KS=34的原因?

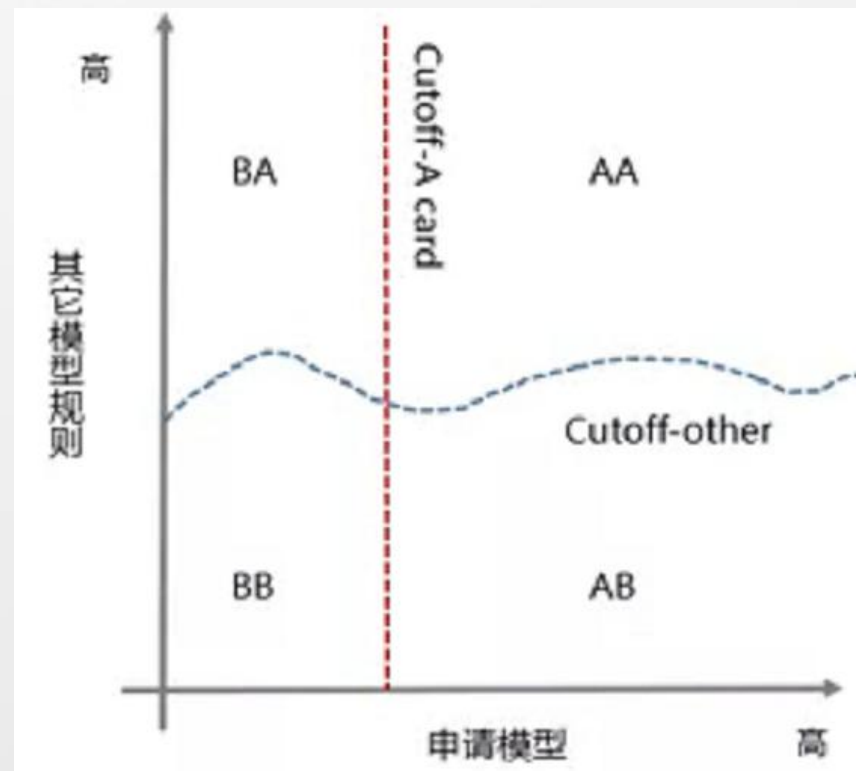
模型上线后, 最坏的客户会被模型拒绝掉, 不会进入到模型评估的样本中 => 导致 Covariate Shift (客群本身发生迁移)



Covariate Shift场景下的模型监控

Thinking: Offline KS=52, Online KS=34的原因?

模型上线后, 最坏的客户会被模型拒绝掉, 不会进入到模型评估的样本中 => 导致 Covariate Shift (客群本身发生迁移)



Summary

评分卡模型的流程：

数据获取

金融机构自身，第三方机构

EDA（探索性数据分析）

统计每个字段的缺失值情况、异常值情况、平均值、中位数、最大值、最小值、分布情况等

为后续的数据处理制定方案

数据清洗

对数据中脏数据，缺失值，异常值进行处理

异常点检测，可以通过聚类检测异常值，先把数据聚成不同的类，选择不属于任何类的数据作为异常值

1) DBSCAN算法，将与数据稠密区域紧密相连的数据对象划分为一个类，因此分离的数据就会作为异常值

2) KMeans算法，把数据聚成k类，计算每个样本和对应的簇中心的距离，找到距离最大的点作为异常值

- 变量分箱

等频分箱，把自变量从小到大排序，根据自变量的个数等分为k部分，每部分作为一个分箱

等距分箱，把自变量从小到大排序，将自变量的取值范围分为k个等距的区间，每个区间作为一个分箱

聚类分箱，用k-means聚类法将自变量聚为k类，但在聚类过程中需要保证分箱的有序性

- WOE编码

特征离散化，是将数值型特征（一般是连续型的）转变为离散特征，比如woe转化，将特征进行分箱，再将每个分箱映射到woe值上，即转换为离散特征

采用woe编码的好处：

- 1) 简化模型，使模型变得更稳定，降低了过拟合的风险
- 2) 对异常数据有很强的鲁棒性，实际工作中的那些很难解释的异常数据一般不会做删除处理，如果特征不做离散化，这个异常数据带入模型，会给模型带来很大的干扰

- 逻辑回归是一种广义线性模型，虽然它引入了Sigmoid函数，是非线性模型，但本质上还是一个线性回归模型（除去Sigmoid函数映射，是线性回归的）
- 如果逻辑回归发过拟合，如何解决？

- 1) 减少特征数量，比如基于IV值的大小进行筛选
- 2) 正则化，L1正则或L2正则

Summary

多模态的数据建模:

- 模型的预测目标与近期的行为关系大
- 信用风险一般比较稳定, 和客户的长期行为相关性强=> 会取比较长时间的数据, 数据维度会很多

1) 高维稀疏与低维稠密

比如客户购买商品ID, 浏览ID => 有上亿的维度

对于高维稀疏特征, 可以采用

人工经验, Embedding

AutoEncoder, PCA (算法生成特征, 姜维)

2) 结构化与非结构化:

Topic Model

Word2Vec

Graph Embedding

3) 线上和线下数据:

线上, 有些数据直接存储在数据库中 => 建立尽可能稳定的模型

线下, 有些数据是客户申请的时候才会产生的 => 建立相对复杂模型, 尽可能挖掘特征的信息

对模型进行融合 => 最终模型

Thinking&Action

Thinking1: 逻辑回归的假设条件是怎样的?

Thinking2: 逻辑回归的损失函数是怎样的?

Thinking3: 逻辑回归如何进行分类?

Thinking&Action

Action1: 基于评分卡的风控模型开发

数据集GiveMeSomeCredit, 15万样本数据

<https://www.kaggle.com/c/GiveMeSomeCredit/data>

使用WOE进行特征变换, IV进行特征筛选, LR构建风控模型,
并对模型评分规则进行可解释性说明

- 基本属性: 包括了借款人当时的年龄
- 偿债能力: 包括了借款人的月收入、负债比率
- 信用往来: 两年内35-59天逾期次数、两年内60-89天逾期次数、两年内90天或高于90天逾期的次数
- 财产状况: 包括了开放式信贷和贷款数量、不动产贷款或额度数量。
- 其他因素: 包括了借款人的家属数量

字段	说明	类型
SeriousDlqin2yrs	90天以上逾期或更差	Y/N
Age	年龄	整数
RevolvingUtilizationOfUnsecuredLines	除房地产和汽车贷款等无分期付款债务外, 信用卡和个人信用额度的总余额除以信贷限额	百分比
DebtRatio	债务比(每月偿还的债务, 赡养费, 生活费除以每月的总收入)	百分比
MonthlyIncome	每月收入	实数
NumberOfOpenCreditLinesAndLoans	公开贷款(如汽车贷款或抵押贷款)和信用额度(如信用卡)的数量	整数
NumberRealEstateLoansOrLines	抵押贷款和房地产贷款的额度(包括房屋净值信贷)	整数
NumberOfTime30-59DaysPastDueNotWorse	借款人逾期30-59天的次数, 但在过去两年没有更糟	整数
NumberOfTime60-89DaysPastDueNotWorse	借款人逾期60-89天的次数, 但在过去两年没有更糟	整数
NumberOfTimes90DaysLate	借款人逾期90天(或以上)的次数	整数
NumberOfDependents	除自己(配偶、子女等)以外的家庭受养人人数	整数

END

THANK YOU

Using data to solve problems

AI专业方向

门徒计划——开班预热课

