玩转逻辑回归算法之金融评分卡模型

----------网易云课堂机器学习微专业：August 助教----------

虽然现在出现了很多性能优秀的分类算法，包括svm，RF，GBDT，DNN等，作为最简单的分类算法，lr 依然是工业界主流的分类算法之一。那么 lr 到底有什么魔力，即使面对如此众多的 “高手” 面前，依然屹立不倒呢?

市面上关于 lr 的书籍和文章大部分的讲解都是针对 lr 一些基本理论或者一些推导公式。掌握这些还远远不够，要想让 lr 发挥其最大效果，必须要有一套科学的、严密的数据预处理流程。

和市面上对 lr 算法的讲解不同，本文将以金融评分卡模型为例，讲解一整套 lr 配套的数据处理流程，包括数据获取，EDA (探索性数据分析)，数据预处理，到变量筛选，lr 模型的开发和评估，生成评分卡模型。希望大家在阅读本篇文章之后能够轻松驾驭 lr 算法。

1. 评分卡模型的背景知识

风控顾名思义就是风险控制，指风险管理者采取各种措施和方法，消灭或减少风险事件发生的各种可能性，或风险事件发生时造成的损失。

信用评分卡模型是最常见的金融风控手段之一，它是指根据客户的各种属性和行为数据，利用一定的信用评分模型，对客户进行信用评分，据此决定是否给予授信以及授信的额度和利率，从而识别和减少在金融交易中存在的交易风险。

评分卡模型在不同的业务阶段体现的方式和功能也不一样。按照借贷用户的借贷时间，评分卡模型可以划分为以下三种：

- 贷前：申请评分卡（Application score card），又称为A卡
- 贷中：行为评分卡（Behavior score card），又称为B卡
- 贷后：催收评分卡（Collection score card），又称为C卡

以下为评分卡模型的示意图：

变量名称	变量范围	得分
基准分	-	223
年龄	$18 \leqslant$ 年龄 < 25	-2
	$25 \leqslant$ 年龄 < 35	8
	$35 \leqslant$ 年龄 < 55	10
	$55 \leqslant$ 年龄	5
性别	男	4
	女	2
婚姻状况	已婚	8
	未婚	-2
学历	硕士，博士	10
	本科	8
	大专	5
	中专，技校，高中	1
	初中，小学	-2
月收入	月收入 $<$ 3000	-8
	$3000 \leqslant$ 月收入 < 5000	0
	$5000 \leqslant$ 月收入 < 8000	5
	$8000 \leqslant$ 月收入 < 12000	13
	$12000 \leqslant$ 月收入	20

那么怎么利用评分卡对用户进行评分呢？一个用户总的评分等于基准分加上对客户各个属性的评分。以上面的评分卡为例：

$$\mathrm{客户评分} = \mathrm{基准分} + \mathrm{年龄评分} + \mathrm{性别评分} + \mathrm{婚姻状况评分} + \mathrm{学历评分} + \mathrm{月收入评分}$$

举个例子某客户年龄为27岁，性别为男，婚姻状况为已婚，学历为本科，月收入为10000，那么他的评分为：

$$223(\mathrm{基准分}) + 8(\mathrm{年龄评分}) + 4(\mathrm{性别评分}) + 8(\mathrm{婚姻状况评分}) + 8(\mathrm{学历评分}) + 13(\mathrm{月收入评分}) = 264$$

Q1: 请计算以上评分卡模型的最低分和最高分

最低分为基准分与每个字段最低分相加：

$$223 - 2 + 2 - 2 - 2 - 8 = 211$$

最高分为基准分与每个字段最高分相加：

$$223 + 10 + 4 + 8 + 10 + 20 = 275$$

以上我们基本了解了评分卡模型的具体用法，看到以上评分卡案例之后，相信很多人肯定会有以下三个疑问：

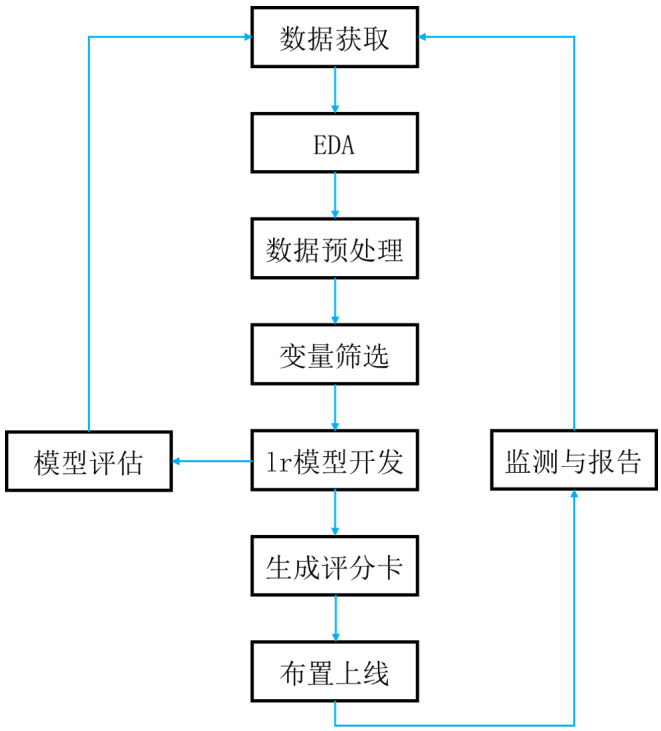
- 用户的属性有千千万万个维度，而评分卡模型所选用的字段在30个以下，那么怎样挑选这些字段呢？
- 评分法卡模型采用的是对每个字段的分段进行评分，那么怎样对评分卡进行有效分段呢？
- 最关键的，也是大家最关心的问题是怎样对字段的每个分段进行评分呢？这个评分是怎么来的？

下面我们一一来解答。

2. 评分卡模型的开发

2.0 总体流程介绍

信用评分卡的开发有一套科学的、严密的流程，包括数据获取，EDA，数据预处理，到变量筛选，lr 模型的开发和评估，生成评分卡模型以及布置上线和模型监测。典型的开发流程如下图所示：



本文仅介绍线下评分卡模型的开发，即数据获取，EDA, 数据预处理，变量筛选，lr 模型开发，模型评估和生成评分卡。

2.1 数据获取

数据的获取途径主要有两个：

- 金融机构自身字段：例用户的年龄，户籍，性别，收入，负债比，在本机构的借款和还款行为等；
- 第三方机构的数据：如用户在其他机构的借贷行为，用户的消费行为数据等。

2.2 EDA（探索性数据分析）

该步骤主要是获取数据的大概情况，例如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。以便制定合理的数据预处理方案。

2.3 数据预处理

数据预处理主要包括数据清洗，变量分箱和WOE编码三个步骤。

2.3.1 数据清洗

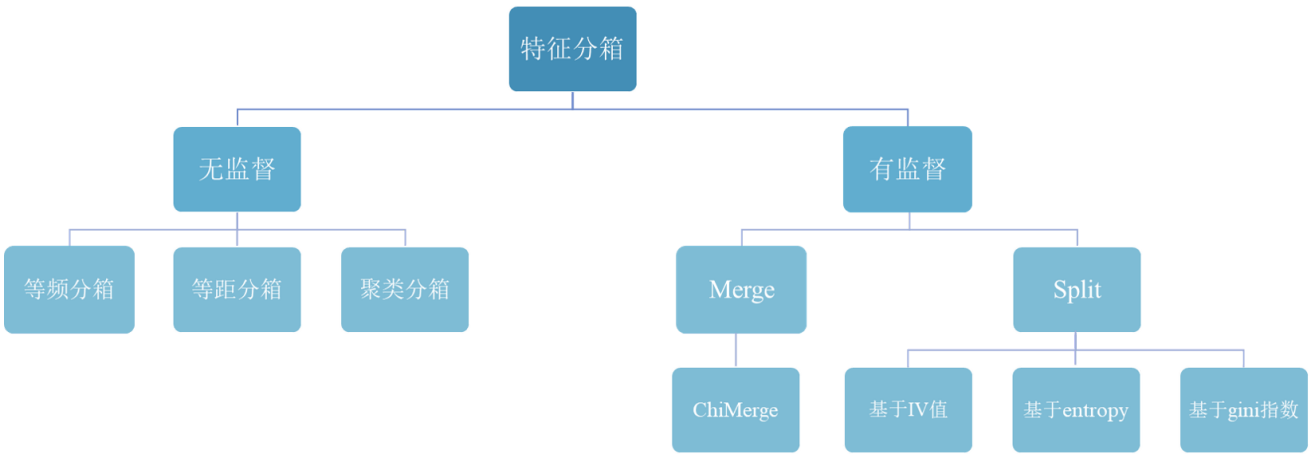
数据清洗主要是对原始数据中脏数据，缺失值，异常值进行处理。关于对缺失值和异常值的处理，我们采用的方法非常简单粗暴，即删除缺失率超过某一阈值（阈值自行设定，可以为30%，50%，90%等）的变量，将剩余变量中的缺失值和异常值作为一种状态 。

2.3.2 变量分箱

在这里我们回答第二个问题评分卡是怎样对变量进行分段的，评分卡模型通过对变量进行分箱来实现变量的分段。那么什么是分箱呢？以下为分箱的定义：

- 对连续变量进行分段离散化；
- 将多状态的离散变量进行合并，减少离散变量的状态数。

常见的分箱类型有以下几种，下面将一一讲解：



1. 无监督分箱

无监督的分箱主要包括以下几类：

(1) 等频分箱：把自变量按从小到大的顺序排列，根据自变量的个数等分为k部分，每部分作为一个分箱。

(2) 等距分箱：把自变量按从小到大的顺序排列，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。

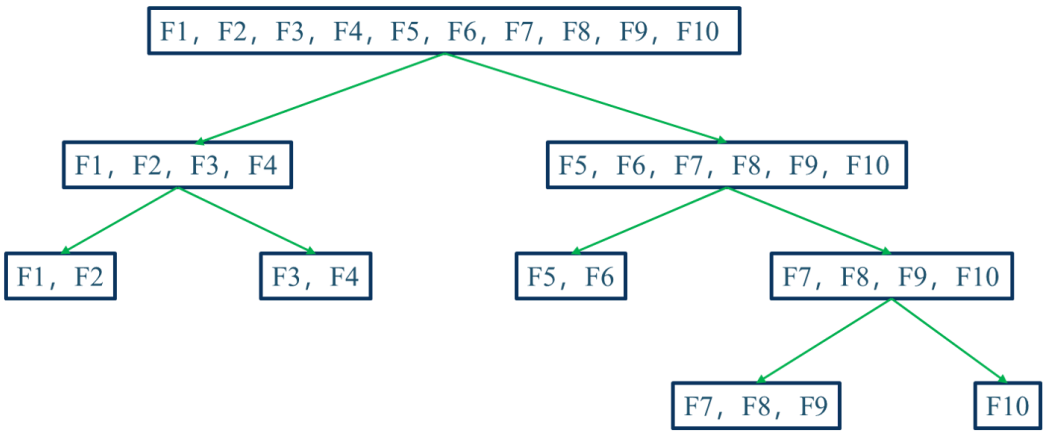
(3) 聚类分箱：用k-means聚类法将自变量聚为k类，但在聚类过程中需要保证分箱的有序性。

由于无监督分箱仅仅考虑了各个变量自身的数据结构，并没有考虑自变量与目标变量之间的关系，因此无监督分箱不一定会带来模型性能的提升。

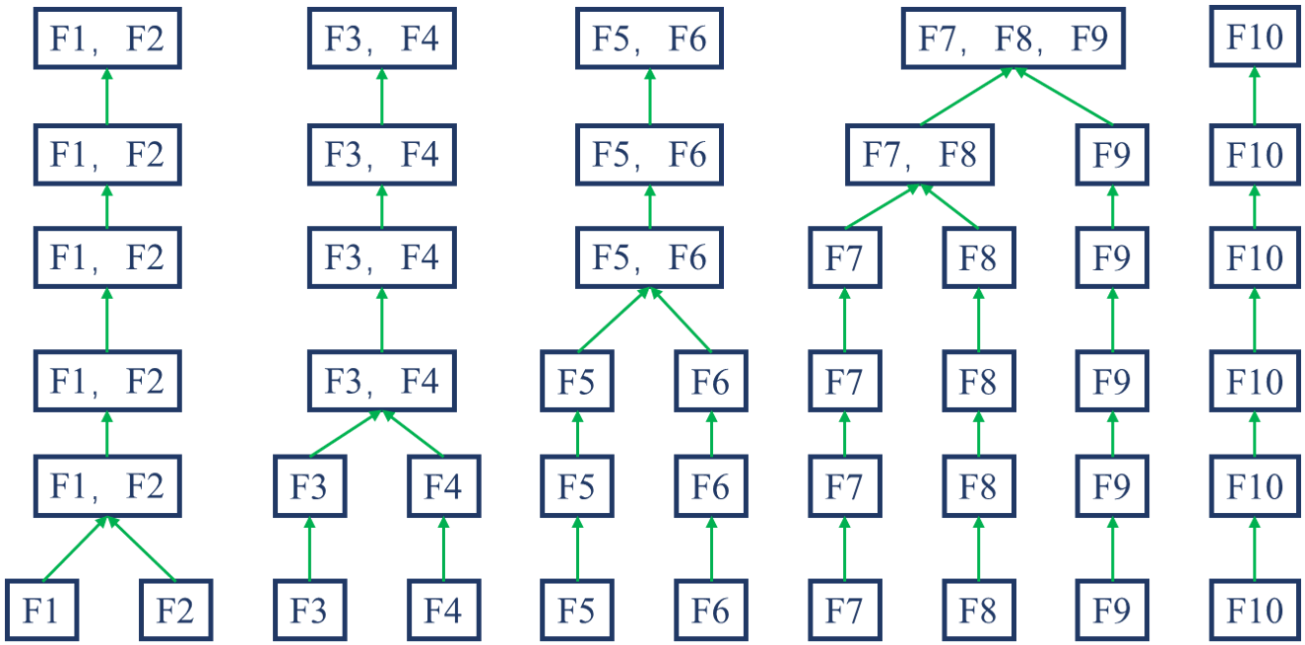
2. 有监督分箱

包括 Split 分箱和 Merge 分箱

1. Split 分箱是一种自上而下(即基于分裂)的数据分段方法。如下图所示，Split 分箱和决策树比较相似，切分点的选择指标主要有 entropy，gini 指数和 IV 值等。



2. Merge 分箱，是一种自底向上(即基于合并)的数据离散化方法。如下图所示为Merge 分箱的示意图，Merge 分箱常见的类型为Chimerge分箱。



Chimerge 分箱是目前最流行的分箱方式之一，其基本思想是如果两个相邻的区间具有类似的类分布，则这两个区间合并；否则，它们应保持分开。Chimerge通常采用卡方值来衡量两相邻区间的类分布情况。

Chimerge的具体算法如下：

- 输入：分箱的最大区间数 n
- 初始化
  - 连续值按升序排列，离散值先转化为坏客户的比率，然后再按升序排列；
  - 对于变量状态数量大于某一阈值 (阈值可以自定义，建议为100) 的变量，为了减少计算量利用等频分箱进行粗分箱。
  - 若有缺失值，则缺失值单独作为一个分箱。
- 合并区间
  - 计算每一对相邻区间的卡方值；
  - 将卡方值最小的一对区间合并

$$X^2 = \sum_{i=1}^{2}\sum_{j=1}^{2}\frac{(A_{ij}-E_{ij})^2}{E_{ij}}$$

A_{ij}：第 i 区间第 j 类的实例数量

E_{ij}：$E_{ij} = \frac{N_i}{N} \times C_j$，$N$ 是合并区间的样本数，N_i 是第 i 组的样本数，C_j 是第 j 类样本在合并区间的样本数。

- 重复以上两个步骤，直到分箱数量不大于 n

- 分箱后处理
  - 对于坏客户比例为 0 或 1 的分箱进行合并 (一个分箱内不能全为好客户或者全为坏客户)。
  - 对于分箱后某一箱样本占比超过 95% 的箱子进行删除。
  - 检查缺失分箱的坏客户比例是否和非缺失分箱相等，如果相等，进行合并。
- 输出：分箱后的数据和分箱区间。

Q2: 一般一个评分卡模型的有效持续时间是 1 个月左右甚至更长时间，中间也许会有一些客户的数据发生变化，比如一个月之内突然换工作，工资上涨等等，针对这种情况，我们该怎样处理呢?

这里我们需要假设客户在短期内属性变化不会太大，即使客户的属性变化，只要在同一分箱中，依然会给这个客户相同的分数。举例来说：对于工资我们可以划分为5箱，即<3000, 3000-5000, 5000-8000, 8000-12000, >12000，假设一个客户的工资为9000，在一个月内工资上涨，那我们就假设这个客户的工资上涨之后不会超过12000，也就是说依然在8000-12000分箱中。这样在考虑客户工资变化的前提下，不会因为客户工资的发生变化而变成了另外一个人，保证了模型的稳定性。

Q3：上文说到将变量中的缺失值作为一种状态是什么意思?

这里的意思是说让缺失值单独分为一箱。

Q4：比如年龄变量中出现'500岁'这种异常字段该怎样处理?

对于年龄特征我们划分为4段，即18-25, 25-35, 35-55, > 55，我们可以直接把500划分到>55这一个分箱中。另外我们也可以通过一些手段检测出异常值，将异常值单独分为一箱。

总结一下特征分箱的优势：

1. 特征分箱可以有效处理特征中的缺失值和异常值。
2. 特征分箱后，数据和模型会更稳定。
3. 特征分箱可以简化逻辑回归模型，降低模型过拟合的风险，提高模型的泛化能力。
4. 将所有特征统一变换为类别型变量。
5. 分箱后变量才可以使用标准的评分卡格式，即对不同的分段进行评分。

2.3.3 WOE编码

分箱之后我们便得到了一系列的离散变量，下面需要对变量进行编码，将离散变量转化为连续变量。WOE编码是评分卡模型常用的编码方式。

WOE 称为证据权重(weight of evidence)，是一种有监督的编码方式，将预测类别的集中度的属性作为编码的数值。对于自变量第 i 箱的WOE值为：

$$WOE_i = \log(\frac{p_{i_1}}{p_{i_0}}) = \log(\frac{\#B_i/\#B_T}{\#G_i/\#G_T})$$

p_{i_1} 是第 i 箱中坏客户占所有坏客户比例

p_{i_0} 是第 i 箱中好客户占所有好客户比例

$\#B_i$ 是第 i 箱中坏客户人数

$\#G_i$ 是第 i 箱中好客户人数

$\#B_T$是所有坏客户人数

$\#G_T$ 是所有好客户人数

公式中的 $\log$ 函数的底一般取为e，即为 $\ln$

从以上公式中我们可以发现，WOE表示的实际上是"当前分箱中坏客户占所有坏客户的比例"和"当前分箱中好客户占所有好客户的比例"的差异。

对以上公式做一个简单变换，可以得到：

$$WOE_i = \log(\frac{\#B_i/\#B_T}{\#G_i/\#G_T}) = \log(\frac{\#B_i/\#G_i}{\#B_T/\#G_T})$$

变换以后可以看出，WOE也可以这么理解，当前分箱中坏客户和好客户的比值，和所有样本中这个比值的差异 (**也就是我们随机的坏客户和好客户的比例**)。WOE越大，这种差异越大，当前分组里的坏客户的可能性就越大，WOE越小，差异越小，这个分组里的样本响应的可能性就越小。当分箱中坏客户和好客户的比例等于随机坏客户和好客户的比值时，说明这个分箱没有预测能力，即WOE=0。

WOE具体计算过程如下表所示：

Bins	Good	Bad	Good%	Bad%	WOE
Bin_1	G_1	B_1	$\frac{G_1}{G_t}$	$\frac{B_1}{B_t}$	$\log\frac{B_1/B_t}{G_1/G_t}$
Bin_2	G_2	B_2	$\frac{G_2}{G_t}$	$\frac{B_2}{B_t}$	$\log\frac{B_2/B_t}{G_2/G_t}$
...	...	...	...	...	...
Bin_n	G_n	B_n	$\frac{G_n}{G_t}$	$\frac{B_n}{B_t}$	$\log\frac{B_n/B_t}{G_n/G_t}$
$Total$	$G_t = \sum G_i$	$B_t = \sum B_i$			

Q5：我们还有没有学过其他编码方式？这里为什么选择采用WOE编码？

我们还学过one-hot编码。one-hot 编码会将原始变量中的每个状态都做为作为一个新的特征，当原始特征状态较多时，数据经过one-hot编码之后特征数量会成倍的增加，同时新特征也会变得过于稀疏。在进行变量筛选的过程中，也会出现原始特征的一部分状态被筛选出来，另一部分状态未被筛选出来，造成特征的不完整。

而WOE编码不仅可以解决以上这些问题，同时还可以将特征转化为线性。

我们知道，逻辑回归的假设函数为:

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

其中p为样本为坏客户的概率，1-p为样本为好客户的概率，整理可得：

$$\log(\frac{p}{1-p}) = \theta^T x$$

其中 $\text{odds} = \frac{p}{1-p}$。

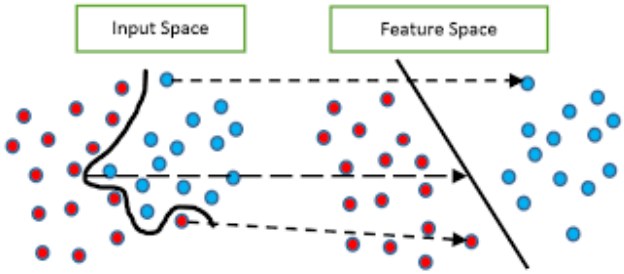
我们再来看来看WOE编码的定义：

$$
\begin{aligned}
WOE_i &= \log(\frac{\#B_i/\#G_i}{\#B_T/\#G_T}) \\
&= \log(\frac{\#B_i}{\#G_i}) - \log(\frac{\#B_T}{\#G_T}) \\
&= \log(\frac{\#B_i/(\#B_i+\#G_i)}{\#G_i/(\#B_i+\#G_i)}) - \text{Const} \\
&= \log(\frac{p_i}{1-p_i}) - \text{Const}
\end{aligned}
$$

其中 p_i 表示第 i 个分箱的坏客户比率，类比$\frac{p_i}{1-p_i}$ 表示的是第 i 个分箱的odds。我们设 $\log(\frac{p}{1-p}) = C$，那么 $\log(\frac{p_i}{1-p_i}) = C_i$，逻辑回归假设函数可以化简为：

$$C = \theta^T(C_i - Const)$$

实际上WOE编码相当于把分箱后的特征从非线性可分映射到近似线性可分的空间内。如下图所示：



Q6：WOE编码为什么不直接表示为该分箱好坏客户数量之比，即$WOE_i = \log(\frac{\#B_i}{\#G_i})$？

如果直接表示为表示为某个分箱好坏客户数量之比，WOE的值在很大程度上受到好坏客户的影响，在严重非均衡的问题中，该比值会非常小，严重影响woe的值。

这里我们举个例子，假设数据中共有5000个好客户和50个坏客户，共有三个分箱，箱1的好客户和坏客户分别有1000个和20个，箱2好客户和坏客户分别有1000个和10个，箱3好客户和坏客户分别有1000个和5个。

显然，箱1和箱3都具有较好的预测能力，而箱2因为坏客户比例和随机预测类似，因此不具有预测能力。

用法1 原公式来计算：

$$
\begin{aligned}
WOE_1 &= \log(\frac{\#B_1/\#B_T}{\#G_1/\#G_T}) = \log(\frac{20/50}{1000/5000}) = \log 2 \\
WOE_2 &= \log(\frac{\#B_2/\#B_T}{\#G_2/\#G_T}) = \log(\frac{10/50}{1000/5000}) = 0 \\
WOE_3 &= \log(\frac{\#B_3/\#B_T}{\#G_3/\#G_T}) = \log(\frac{5/50}{1000/5000}) = -\log 2
\end{aligned}
$$

从以上结果我们发现，箱2对应的WOE=0，说明不具有预测能力，而箱1和箱3的WOE分别为log2和-log2，均远离0点，具有预测能力。

用法2 利用分箱中好坏客户数量来计算：

$$WOE_1 = \log(\frac{\#B_1}{\#G_1}) = \log(\frac{20}{1000}) = -\log 50$$
$$WOE_2 = \log(\frac{\#B_2}{\#G_2}) = \log(\frac{10}{1000}) = -\log 100$$
$$WOE_3 = \log(\frac{\#B_3}{\#G_3}) = \log(\frac{5}{1000}) = -\log 200$$

然而从法2得到的结果中只能判断三个分箱的坏客户的比例大小情况，无法判断箱1，箱2和箱3的预测能力。

总结一下WOE编码的优势：

- 可提升模型的预测效果
- 将自变量规范到同一尺度上
- WOE能反映自变量取值的贡献情况
- 有利于对变量的每个分箱进行评分
- 转化为连续变量之后，便于分析变量与变量之间的相关性
- 与独热向量编码相比，可以保证变量的完整性，同时避免稀疏矩阵和维度灾难

2.4 变量筛选

之前我们说到过用户的属性有千千万万个维度，而评分卡模型所选用的字段在30个以下，那么怎样挑选这些字段呢？

挑选入模变量需要考虑很多因素，比如：变量的预测能力，变量之间的线性相关性，变量的简单性（容易生成和使用），变量的强壮性（不容易被绕过），变量在业务上的可解释性（被挑战时可以解释的通）等等。其中最主要和最直接的衡量标准是变量的预测能力和变量的线性相关性。本文主要探讨基于变量预测能力的单变量筛选，变量两两相关性分析，变量的多重共线性分析。

2.4.1 单变量筛选

单变量的筛选基于变量预测能力，常用方法：

- **基于IV值的变量筛选**
- **基于stepwise的变量筛选**
- **基于特征重要度的变量筛选：RF, GBDT...**
- **基于LASSO正则化的变量筛选**

1. 基于IV值的变量筛选

IV称为信息价值(information value)，是目前评分卡模型中筛选变量最常用的指标之一，自变量的IV值越大，表示自变量的预测能力越强。类似的指标还有信息增益、基尼(gini)系数等。常用判断标准如下：

IV范围	预测能力
< 0.02	无效
0.02-0.10	弱预测力
0.10-0.20	中预测力
> 0.20	强预测力

那么怎么计算变量中第 i 个分箱对应的 IV 值的计算公式为：

$$IV_i = (\frac{\#B_i}{\#B_T} - \frac{\#G_i}{\#G_T}) * log(\frac{\#B_i/\#B_T}{\#G_i/\#G_T}) = (\frac{\#B_i}{\#B_T} - \frac{\#G_i}{\#G_T}) * WOE_i$$

变量对应的IV值为所有分箱对应的 IV 值之和：

$$IV = \sum_i^n IV_i$$

从上式我们可以看出变量的 IV 值实际上式变量各个分箱的加权求和。且和决策树中的交叉熵有异曲同工之妙。以下为交叉熵公式：

$$Ent(D) = -\sum_{i=1}^{n} p_i \ log \ p_i$$

IV值的具体的计算流程如下：

Bins	Good	Bad	Good%	Bad%	WOE	IV
Bin_1	G_1	B_1	$\frac{G_1}{G_t}$	$\frac{B_1}{B_t}$	$\log \frac{B_1/B_t}{G_1/G_t}$	$(\frac{G_1}{G_t} - \frac{B_1}{B_t}) \log \frac{B_1/B_t}{G_1/G_t}$
Bin_2	G_2	B_2	$\frac{G_2}{G_t}$	$\frac{B_1}{B_t}$	$\log \frac{B_2/B_t}{G_2/G_t}$	$(\frac{G_2}{G_t} - \frac{B_2}{B_t}) \log \frac{B_2/B_t}{G_2/G_t}$
...	...	...	...	...	...	...
Bin_n	G_n	B_n	$\frac{G_n}{G_t}$	$\frac{B_1}{B_t}$	$\log \frac{B_n/B_t}{G_n/G_t}$	$(\frac{G_n}{G_t} - \frac{B_n}{B_t}) \log \frac{B_n/B_t}{G_n/G_t}$
$Total$	$G_t = \sum G_i$	$B_t = \sum B_i$				$\sum (\frac{G_i}{G_t} - \frac{B_i}{B_t}) \log \frac{B_i/B_t}{G_i/G_t}$

Q7：请补全以下表格

自变量为age，Y表示目标变量，其中bad代表坏客户，good代表好客户。我们希望能用自变量age来预测好坏客户的概率，以此来决定是否放贷。

age	Bad	Good	Bad/(Bad+Good)	Bad%	Good%	WOE	IV
18-30	250	4750	0.0500	0.25	0.5278	-0.7472	0.2076
30-45	300	2700	0.1000	0.3	0.3000	0.0000	0.0000
45-55	250	1200	0.1724	0.25	0.1333	0.6286	0.0733
>55	200	350	0.3636	0.2	0.0389	1.6376	0.2638
Total	1000	9000		1	1		0.5447

从以上案例中我们可以分析出：

- 当前分箱中，坏客户占比越大，WOE值越大；
- 当前分箱中WOE的正负，由当前分箱中好坏客户比例，与样本整体好坏客户比例的大小关系决定。
  - 当分箱的比例小于整体比例时，WOE为负 。例如年龄18-30分箱中：$250/4750 < 1000/9000$ ，该分箱对应的WOE为负值；
  - 当分箱的比例大于整体比例时，WOE为正。例如年龄45-55分箱中：$250/1200 > 1000/9000$ ，该分箱对应的WOE为正值；
  - 当分箱的比例等于整体比例时，WOE为0。例如年龄30-45分箱中：$300/2700 = 1000/9000$ ，该分箱对应的WOE为0。

- WOE的取值范围是[-∞,+∞]，当分箱中好坏客户比例等于整体好坏客户比例时，WOE为0。
- 对于变量的一个分箱，这个分组的好坏客户比例与整体好坏客户比例相差越大，IV值越大，否则，IV值越小。
- IV值的取值范围是[0,+∞)，当分箱中只包含好客户或坏客户时，IV = +∞，当分箱中好坏客户比例等于整体好坏客户比例时，IV为0。

2. 基于stepwise的变量筛选

基于基于stepwise的变量筛选方法也是评分卡中变量筛选最常用的方法之一。具体包括三种筛选变量的方式：

- 前向选择forward：逐步将变量一个一个放入模型，并计算相应的指标，如果指标值符合条件，则保留，然后再放入下一个变量，直到没有符合条件的变量纳入或者所有的变量都可纳入模型。
- 后向选择backward：一开始将所有变量纳入模型，然后挨个移除不符合条件的变量，持续此过程，直到留下所有最优的变量为止。
- 逐步选择stepwise：该算法是向前选择和向后选择的结合，逐步放入最优的变量、移除最差的变量。

3. 基于特征重要度的变量筛选

基于特征重要度的变量筛选方法是目前机器学习最热门的方法之一，其原理主要是通过随机森林和GBDT等集成模型选取特征的重要度。

- 随机森林计算特征重要度的步骤：
  - 对每一颗决策树，选择相应的袋外数据（OOB）计算袋外数据误差，记为e rrOOB1；
  - 随机对袋外数据OOB所有样本的特征加入噪声干扰(随机的改变样本在该特征的值)，再次计算袋外数据误差，记为errOOB2；
  - 特征的重要度= $\sum(err_{OOB2}-err_{OOB1})/N$，N 表示随机森林中决策树的个数。

  当改变样本在该特征的值，若袋外数据准确率大幅度下降，则该特征对于样本的预测结果有很大影响，说明特征的重要度比较高。
- GBDT计算特征重要度原理：

  特征 j 在单颗树中的重要度的如下：

$$\hat{J}_j^2(T)=\sum_{t=1}^{L-1}\hat{i}_t^2 1(v_t=j)$$

  其中，L 为树的叶子节点数量，L-1 为树的非叶子节点数量，v_t 是和节点 t 相关联的特征，i_t^2 是节点 t 分裂之后平方误差的减少值。

  特征 j 的全局重要度为特征j在单颗树中的重要度的平均值：

$$\hat{J_j^2}=\frac{1}{M}\sum_{m=1}^{M}\hat{J}_j^2(T_m)$$

  其中，M 是树的数量。

4. 基于LASSO正则化的变量筛选

L1正则化通常称为Lasso正则化，它是在代价函数上增加了一个L1范数：

$$J(\theta)=-\sum_{i=1}^{m}(y^{(i)}log(h_\theta(x^{(i)}))+(1-y^{(i)})log(1-h_\theta(x^{(i)})))+\frac{\lambda}{m}\sum_{j=1}^{n}|\theta_j|$$

具体原理参照本人博客 直观理解正则化。

2.4.2 变量相关性分析

1 变量两两相关性分析

对于自变量 X_1, X_2，如果存在常数 c_0, c_1, c_2 使得以下线性等式近似成立:

$$c_1 X_1 + c_2 X_2 \approx c_0$$

称自变量 X_1, X_2 具有较强的线性相关性。

两变量间的线性相关性可以利用皮尔森相关系数来衡量。系数的取值为 $[-1.0, 1.0]$，相关系数越接近0的说明两变量线性相关性越弱，越接近1或-1两变量线性相关性越强。

$$r_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\bar{X})(Y-\bar{Y}))}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}} = \frac{E(XY)-E(X)(Y)}{\sqrt{E(X^2)-E^2(X)}\sqrt{E(Y^2)-E^2(Y)}}$$

当两变量间的相关系数大于阈值时（一般阈值设为 0.7 或 0.4），剔除IV值较低的变量，或分箱严重不均衡的变量。

2 变量的多重共线性分析

对于自变量 $X_1, X_2, \cdots, X_n$，如果存在常数 $c_0, c_1, c_2, \cdots, c_n$ 使得以下线性等式近似成立:

$$c_1 X_1 + c_2 X_2 + \cdots + c_n X_n \approx c_0$$

称自变量 $X_1, X_2, \cdots, X_n$具有较强的多重共线性。

通常用 VIF 值来衡量一个变量和其他变量的多重共线性：

$$VIF_i = \frac{1}{1-R_i^2}$$

其中 R_i 为 X_i 与其他自变量的复相关系数。

$$R_i = \frac{\sum(X_i-\bar{X}_i)(\hat{X}_i-\bar{X}_i)}{\sqrt{\sum(X_i-\bar{X}_i)^2\sum(\hat{X}_i-\bar{X}_i)^2}}$$

其中：$\hat{X}_i$为其他变量的线性表示 $\hat{X}_i = \theta_0 + \theta_1 X_1 + \cdots + \theta_{i-1}X_{i-1} + \theta_{i+1}X_{i+1} + \cdots$

$\bar{X}_i$ 为变量 X_i 的均值

当某个变量的 VIF 大于阈值时（一般阈值设为10 或 7），需要逐一剔除解释变量。当剔除掉 X_k 时发现VIF低于阈值，从 $\{X_k\ ,\ X_i\}$ 中剔除IV值较低的一个。

Q8：为什么要进行相关性分析?

设想建立一个具有两变量 X_1 和 X_2 的线性模型，真实模型是 $Y = X_1 + X_2$。如果 X_1 和 X_2 线性相关（比如说 $X_1 \approx 2X_2$），那么拟合模型 $Y = 3X_2, Y = 2X_1 - X_2$ 或 $Y = 51X_1 - 99X_2$ 的效果都一样好，理想状态下，系数权重会有无数种取法，使系数权重变得无法解释，导致变量的每个分段的得分也有无数种取法（后面我们会发现变量中不同分段的评分会用到变量的系数）

即使不进行线性相关性分析也不会影响模型的整体性能，进行相关性分析只是为了让我们的模型更易于解释，保证不同的分箱的得分正确。

总结一下变量筛选的意义：

- 剔除跟目标变量不太相关的特征
- 消除由于线性相关的变量，避免特征冗余
- 减轻后期验证、部署、监控的负担
- 保证变量的可解释性

2.5 构建逻辑回归模型

主要包括构建初步的逻辑回归模型，根据p-value进行变量筛选，根据各个变量的系数符号进行筛选，得到最终的逻辑回归模型。

采用逻辑回归模型的优点：

- 简单，稳定
- 可解释性强
- 技术成熟
- 易于检测和部署

2.5.1 根据p-value进行筛选

p-value是假设检验的里面的概念。模型假设某自变量与因变量线性无关，p-value可以理解为该假设成立的可能性(便于理解，不太准确)。一般，当p-value大于阈值时，表示假设显著，即自变量与因变量线性无关；当p-value小于阈值时，表示假设不显著，即自变量与因变量线性相关。阈值又称为显著性水平，通常取0.05。

因此当某个字段的 p-value 大于0.05时，应该删除此变量。

2.5.2 根据系数符号进行筛选

检查逻辑回归模型中各个变量的系数，如果所有变量的系数均为正数，模型有效。假如有一些变量的系数出现了负数，说明有一些自变量的线性相关性较强，需要进一步进行变量筛选。通常的做法是：

- 综合考虑变量的IV值和业务的建议，按照变量的优先级进行降序排列；
- 选择优先级最高的4-5个基本变量；
- 按优先级从高到低逐渐添加变量，当新添加的变量之后，出现系数为负的情况，舍弃该变量；
- 直到添加最后一个变量。

Q9：为什么回归模型中各个变量的系数均为正数?

由以上分析我们知道对于分箱的WOE编码，分箱中坏客户占比越大，WOE值越大； 也就是说WOE值越大，表示该分箱中客户为坏客户的概率就越大，即WOE与逻辑回归的预测结果成正比。

Q10：为什么说假如有一些变量的系数出现了负数，说明有一些自变量的线性相关性较强?

我们知道，正常情况下，WOE编码后的变量系数一定为正值。由上面为什么进行线性相关性分析的问题可知，由于一些自变量线性相关，导致系数权重会有无数种取法，使得可以为正数，也可以为负数。

2.6 模型评价

(具体参见我的博客 机器学习评估指标的前世今生)

2.6.1 混淆矩阵，TPR (Recal)，FPR

TPR (或Recall) 为坏客户的查全率，表示被模型抓到的坏客户占总的坏客户的比例，表达式为：

$$TPR = \frac{TP}{TP + FN}$$

FPR 为好客户误判率，表示好客户中倍模型误误判的比例，表达式为：

$$FPR = \frac{FP}{FP + TN}$$

可以把TPR看做模型的收益，FPR看做模型付出的代价。如果一个模型 TPR越大，表示模型能够抓到的坏客户比例越大，即收益越大；FPR越大，表示模型能够将好客户误抓的比例越大，即代价越大。

2.6.2 AUC

AUC 表示模型对任意坏客户的输出结果为大于模型对任意好客户的输出结果的概率。AUC的取值范围在0.5和1之间，AUC 越大，表示模型预测性能越好。

2.6.3 KS值

KS 值表示了模型区分好坏客户的能力。其实质是 $TPR - FPR$ 随好坏客户阈值变化的最大值。KS 的取值范围在0.5和1之间，值越大，模型的预测准确性越好。一般，KS > 0.4 即认为模型有比较好的预测性能。

2.7 转化为评分卡

我们将客户违约的概率表示为p，则正常的概率为1-p。由逻辑回归的基本原理可得：

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

整理以上公式：

$$\log(\frac{p}{1-p}) = \theta^T x$$

我们可以定义比率来表示客户违约的相对概率：

$$\text{odds} = \frac{p}{1-p}$$

将 odds 带入可得：

$$\log(\text{odds}) = \theta^T x$$

评分卡的分值可以定义为比率对数的线性表达来，即：

$$Score = A - B \times \log(\text{odds})$$

其中A与B是常数，B前面的负号可以使得违约概率越低，得分越高。通常情况下，即高分值代表低风险，低分值代表高风险。

A、B的值可以通过将两个已知或假设的分值带入计算得到。通常情况下，需要设定两个假设：

- 某个特定的违约概率下的预期评分，即比率 odds 为 θ_0 时的分数为 P_0

- 该违约概率翻倍的评分（PDO）

根据以上的分析，则 odds 为 $2\theta_0$ 时的分数为 $P_0 - PDO$，代入以上线性表达式，可得：

$$P_0 = A - B \times \log(\theta_0)$$
$$P_0 - PDO = A - B \times \log(2\theta_0)$$

解该方程组，可得：

$$B = \frac{PDO}{\log 2}$$
$$A = P_0 + B * \log(\theta_0)$$

在实际的应用中，我们会计算出每个变量的各分箱对应的分值。新用户产生时，对应到每个分箱的值，将这些值相加，最后加上初始基础分，得到最终的结果。

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

式中：变量 $x_1 \ldots x_n$ 是出现在最终模型的入模变量。由于所有的入模变量都进行了WOE编码，可以将这些自变量中的每一个都写 $(\theta_i w_{ij})\delta_{ij}$的形式：

$$Score = A - B \left\{ \begin{array}{c} \theta_0 \\ +(\theta_1 w_{11})\delta_{11} + (\theta_1 w_{12})\delta_{12} + \cdots \\ \cdots \\ +(\theta_n w_{n1})\delta_{n1} + (\theta_n w_{n2})\delta_{n2} + \cdots \end{array} \right\}$$

其中，$A - B\theta_0$为基础分数，θ_i为逻辑回归中第 i 个自变量的系数，w_{ij} 为第 i个变量的第 j 个分箱的WOE值，δ_{ij} 是0，1逻辑变量，当 $\delta_{ij} = 1$ 代表自变量 i 取第 j 个分箱，当 $\delta_{ij} = 0$ 代表自变量 i 不取第 j 个分箱。最终得到评分卡模型：

变量	分箱类别	分值
基准分	--	$A - B\theta_0$
x_1	1	$-B\theta_1 w_{11}$
	2	$-B\theta_1 w_{12}$
	...	...
	k_1	$-B\theta_1 w_{1k_1}$
x_2	1	$-B\theta_2 w_{21}$
	2	$-B\theta_2 w_{22}$
	...	...
	k_2	$-B\theta_2 w_{2k_2}$
...	...	...
x_n	1	$-B\theta_n w_{n1}$
	2	$-B\theta_n w_{n2}$
	...	...
	k_n	$-B\theta_n w_{nk_n}$

从以上公式中，我们发现每个分箱的评分都可以表示为$-B(\theta_i w_{ij})$，也就是说影响每个分箱的因素包括三部分，分别为参数B，变量系数 θ_i，和对应分箱的WOE编码 w_{ij}。

2.8 小结

最后我们再来回答最初的三个问题作为本文的小结：

1. 用户的属性有千千万万个维度，而评分卡模型所选用的字段在30个以下，那么怎样挑选这些字段呢？
    - 变量预测能力筛选，
    - 变量相关性分析（包括两两相关性分析，多重共线性分析），
    - 根据p-value筛选，
    - 根据变量的系数符号进行筛选。
2. 评分法卡模型采用的是对每个字段的分段进行评分，那么怎样对评分卡进行分段呢？
    - 变量分箱。
3. 怎样对字段的每个分段进行评分呢？这个评分是怎么来的？
    - WOE编码，
    - 将预测概率值转化为评分：$Score = A - B \times \log(\text{odds})$，
    - 利用变量相关性分析和变量的系数符号保证每个分箱评分的合理性。