评分卡专题算法

1. 应用场景

信用评分分为个人信用评分和企业信用评分，本文主要研究个人信用评分，最早追溯到1801年英国征信局在布鲁克林成立，其对授信人、受信人以及社会产生了巨大的价值。具体而言，对于授信人，信用评分系统可以用来确定市场和开拓新顾客，使授信人更加懂得顾客的消费行为和购买力，有利于与顾客建立良好的信誉关系，促进内部金融管理质量的提高。对于受信人，信用评分系统可以使得受信人得到公正客观的信用评价，从而有利于充分利用各种信用产品。对社会而言，信用评分系统可以降低信用评估成本，提高信用评分效率，推动信用产品的应用，从而使整个社会的经济效率提高。

随着大数据时代的到来，大数据技术为信用评分带来了新的机遇。在传统信用评分系统中，所利用的数据主要包括：客户向金融机构提交的个人申请信息、金融机构内部积累的客户历史数据、人民银行征信中心等机构提供的数据等。这些数据价值密度高、维度相对较低、质量相对较高。而基于大数据的信用评分会考察更加多元化的客户信息数据，在传统数据基础上加入来自各互联网金融生态体系内的数据。互联网数据往往较为稀疏、价值密度较低、单变量区分能力较弱，但覆盖范围广泛。在基于大数据的信用评分系统中，采用先进的数据挖掘工具进行更加复杂的信用评分，将产生更加精确的个人信用评分。

1. 问题定义

信用评分卡一般分为三种：A卡、B卡、C卡、反欺诈评分卡：

A卡：申请评分卡，对客户申请贷款进行筛选并判断其违约风险。

B卡：行为评分卡，对通过审批的客户进行整个生命周期的管理。

C卡：催收评分卡，对逾期账户预测催收策略反应的概率，从而采取相应的催收措施。

其中申请评分卡、反欺诈评分卡使用在申请环节，行为评分卡使用在监控环节，催收评分卡使用在逾期管理环节。

在此以申请评分卡为例，其评分结果将应用于以下几个方面：

1. 信用状况，即正常和违约，并据此决定批准还是拒绝该笔贷款申请。
2. 为了获得审批通过需要提供的抵押物。
3. 贷款金额（信用额度）。
4. 贷款定价（利率水平）。

而行为评分卡是已经进行了一定交易的账户，进行信用评分。该评分过程反复进行，以监测和管理业务账户。其评分结果将用于：

1. 审查信用重建。
2. 审查信用额度。
3. 制定清收策略。（如果违约或逾期付款）
4. 审查贷款定价和贷款条件。
5. 开发流程

标准评分卡格式：

|  |  |  |
| --- | --- | --- |
| 标准评分卡1 | | |
| 变量 | 条件 | 分值 |
| 基础分值 |  | 485 |
| Age | 如果Age<25 | 19 |
| 如果Age>=25且Age<33 | 28 |
| 如果Age>=33且Age<48 | 39 |
| 如果Age>=48且Age<56 | 24 |
| 如果Age>=56 | 20 |
| 如果Age=缺失值 | 19 |
| TmAtAddress | 如果TmAtAddress<1 | 12 |
| 如果TmAtAddress>=1且TmAtAddress<3 | 24 |
| 如果TmAtAddress>=3且TmAtAddress<5 | 36 |
| 如果TmAtAddress>=5 | 41 |
| 如果TmAtAddress=缺失值 | 17 |
| EmpStatus | 如果EmpStatus=全职 | 38 |
| 如果EmpStatus=兼职 | 19 |
| 如果EmpStatus=自由职业 | 25 |
| 如果EmpStatus=失业 | 7 |
| 如果EmpStatus=缺失值 | 3 |

上表是一个只有三个变量的评分卡，若一个人Age=37，TmAtAddress=3.5，EmpStatus=全职，则他的信用评分=485+39+36+38=598

|  |  |  |
| --- | --- | --- |
| 标准评分卡 2 | | |
| 变量 | 条件 | 分值 |
| Age | 如果Age<25 | 181 |
| 如果Age>=25且Age<33 | 190 |
| 如果Age>=33且Age<48 | 201 |
| 如果Age>=48且Age<56 | 186 |
| 如果Age>=56 | 182 |
| 如果Age=缺失值 | 181 |
| TmAtAddress | 如果TmAtAddress<1 | 174 |
| 如果TmAtAddress>=1且TmAtAddress<3 | 186 |
| 如果TmAtAddress>=3且TmAtAddress<5 | 198 |
| 如果TmAtAddress>=5 | 203 |
| 如果TmAtAddress=缺失值 | 179 |
| EmpStatus | 如果EmpStatus=全职 | 200 |
| 如果EmpStatus=兼职 | 181 |
| 如果EmpStatus=自由职业 | 187 |
| 如果EmpStatus=失业 | 169 |
| 如果EmpStatus=缺失值 | 165 |

上表是一个只有三个变量的评分卡，

若一个人Age=37，TmAtAddress=3.5，EmpStatus=全职，

则他的信用评分=201+198+200=599（四舍五入有误差）分数一般是整数表示。本人更倾向于第一种表示方式。第二种是把第一种表示中的基础分数平均分配到三个变量计算得到的。

评分卡开发流程图

1）数据准备

申请评分卡用在申请环节，以申请者在申请当日及过去的信息为基础，预测未来放款后的逾期或者违约的概率。主要目的是筛选优质客户。现在各大互联网借贷平台雨后春笋般涌现，客户只需在APP上注册姓名、身份证号、手机号三要素即可给定一个信用额度，其背后的风控是借用第三方公司提供的信息，如央行征信、银行流水、网上消费记录等。

* 搜集数据和整合

因此准备如下数据，建立宽表：

1. 个人信息：性别、年龄、学历等
2. 负债能力：在申请的金融机构或其他金融机构的负债情况
3. 消费能力：商品购买记录、出境游、奢侈品消费
4. 历史信用记录：历史逾期行为
5. 其他数据：个人交际、网络轨迹、个人财务等

示例如下：

|  |  |
| --- | --- |
| 字段 | 名称 |
| member\_id | ID |
| loan\_amnt | 申请额度 |
| term | 产品期限 |
| int\_rate | 利率 |
| emp\_length | 工作期限 |
| home\_ownership | 是否有自有住宅 |
| annual\_inc | 年收入 |
| verification\_status | 收入核验状态 |
| desc | 描述 |
| purpose | 贷款目的 |
| title | 贷款目的描述 |
| zip\_code | 联系地址邮政编码 |
| addr\_state | 联系地址所属州 |
| delinq\_2yrs | 申贷日期前2年逾期次数 |
| inq\_last\_6mths | 申请日前6个月咨询次数 |
| mths\_since\_last\_delinq | 上次逾期距今月份数 |
| mths\_since\_last\_record | 上次登记公众记录距今的月份数 |
| open\_acc | 征信局中记录的信用产品数 |
| pub\_rec | 公众不良记录数 |
| total\_acc | 正在使用的信用产品数 |
| pub\_rec\_bankruptcies | 公众破产记录数 |
| earliest\_cr\_line | 第一次借贷时间 |
| loan\_status | 贷款状态—目标变量 |

* 数据EDA描述

样本标签分布，图表展示（略）

但是同时这些数据也会带来一些问题，如正负样本不平衡问题，在一个样本中好的样本很少或坏的样本很少，解决方案如下：

1. 过采样：对少的样本进行重复采样，操作简单，对数据的质量要求也低；缺点是容易造成模型的过拟合。
2. 欠采样：方法和过采样类似，缺点是容易造成模型的欠拟合。
3. SMOTE：采用最近邻算法计算每个少数类样本的K个同类近邻；从K个同类近邻中随机挑选N个样本进行随机线性插值；构造新的少数类样本，这些样本和原数据合成，生成新的训练集。优点是不易过拟合，能够保留大量信息，缺点是不能对缺失值和类别变量做处理。

2）数据预处理

* 时间格式

1. 利率方面的处理办法：带%的百分比，需要转化为浮点数
2. 工作年限“<1 year”转化为0，“>10 year”的转化为1
3. 日期方面：直接转化为标准日期

* 缺失值

1. 直接删除，行级、列级
2. 填充数值，中位数、众数
3. 分箱时单独一箱

* 衍生特征

1. 计数：过去1年内申请贷款的总次数
2. 求和：过去1年内的在线上的消费金额
3. 比例：贷款申请额度和年收入的占比
4. 时间差：第一次开户距离今天的时间长度
5. 波动率：过去3年内每份工作的时间的标准差

* 分箱

1. best\_ks，有监督的分箱算法，对单变量建立决策树。对连续变量进行排序、拆分具体实现细节要考究？为了方便评分卡的解释，建议分箱字段是单调的，并计算woe值，woe值相近的箱应该合并，同时每箱的数据占比>=5%，因为太少的数据为一箱没有什么意义。建议单变量分箱个数5—10之间。

解释woe：一般定义违约或逾期用户=1（即坏用户），正常用户=0（好用户），某变量第i箱的woe(i)=ln(bad\_rate(i)/good\_rate(i))，bad\_rate(i)=bad\_num(i)/bad\_num\_total

good\_rate(i)=good\_num(i)/good\_num\_total

1. 等频，无监督算法，
2. 等距，无监督算法。

分箱和建模类似，都是根据预测效果好坏不断迭代优化，进行微调。

3）特征选择

* 相关性

皮尔逊相关系数，计算两两变量相关系数，如何做筛选呢？

* 显著性

通过计算某变量的IV值得到该变量是否有预测力，如果IV值较大，则该变量预测力较强。

IV(i)=bad\_rate(i)/good\_rate(i)\*woe(i)

某变量的IV=sum(IV(i))

4）模型开发

截止目前，看到的评分卡模型基本都是用逻辑回归开发的，优点比较多，稳定和解释性强，解释性强因为在对比其他分数期间，逻辑回归相对是有多个可加项，可具体比较，SVM就基本做不到，缺点是对数据质量的要求非常高，需要在数据预处理方面花很多的时间，模型的简单但是开发成本并不低；决策树模型方面，对数据质量要求低，也比较容易解释，但是模型的准确度不高；组合模型方面，部署比较麻烦，在评分卡方面应用不是很多。

其中理想模型是“逐步逻辑回归”。

1. 应用部署
2. 上线监控