基于机器学习的贷款违约预测

Datawhale September

StarLEE

1. 背景

个人资产业务，即个人信贷业务。主要指运用从负债业务筹集的资金，将资金的使用权在一定期限内有偿让渡给个人，并在贷款到期时收回资金本息以取得收益的业务。个人资产业务作为商业银行主要的收益来源，对商业银行的经营成果起着重要的作用。互联网技术和移动通信技术的不断创新以及与金融行业的快速融合，使得互联网金融中的重要发展模式——网络借贷得到了飞速的发展。网络借贷是指互联网公司利用互联网技术和移动通信技术搭建一个线上平台，为资金提供者和资金需求者提供直接的资金融通服务。这种基于网络平台的借贷方式相较于银行借贷，不仅手续简便、效率高，且允许信用贷款，为自有资本不足、缺乏担保的小额借款人提供了新的出路。它的出现满足了人们多元化的投融资需求，提高了社会闲散资金的利用率。

在以往，商业银行对贷款用户进行信用风险评估时，往往依靠风控人员依赖5C分类法来主观判断，从其个人品格、信用额度、偿付能力、市场经济状况这5个因素对贷款用户进行判断和权衡，以此作为是否贷款给该用户的参考，决定是否发放贷款，这种依赖主观判断的方法显然效率低下，而且评估的时候十分依赖风控人员的主观判断能力，从公司内部管控的角度来看，甚至存在风控人员内部作弊的可能性，不能适应市场经济的快速发展，满足贷款用户的需求，也不能满足网贷平台风险管理的需求。面对数以万计甚至是数以十万计的申请借款的用户时，网贷平台则需要采用各种机器学习的方法来减少监控与检测过程中的人工参与部分，利用自动化的方法提高放款审核的准确率和效率。

目前，网贷行业运用大数据技术进行风险控制管理已经取得了一定的成效，比较成熟的产品有Zest Finance公司所开发的基于数据挖掘和机器学习理论的分析模型，以及美国使用最广泛的个人信用评分系统——FICO信用评分，都是美国借贷行业贷款决策的重要参考标准。而在国内，2014年10月，宜人贷将采用了大数据技术的“极速模式”添加进“宜人贷借款”APP中，积木盒子根据建立的“读秒”标准判断用户的信用等级，拍拍贷于2015年推出的魔镜风控系统被认为是行业内首个基于大数据的风控模型，首个能准确预测借款标的风险概率的风控系统，爱钱进在其两周年发布会上，推出了基于机器学习、深度学习等技术的全新风控体系——“云图动态风控系统”。由此可见，机器学习的蓬勃兴起使网络借贷平台利用多维大数据构建智能风控模型，更加准确的评估个人信用状况，有效地降低违约风险。

2. 数据理解

2.1 变量信息

训练集数据包含了800000个观测值以及47个变量（包括15列匿名变量），变量字段表如下：

**表2.1 变量字段表**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Field** | **Description** |
| id | 为贷款清单分配的唯一信用证标识 | dti | 债务收入比 |
| loanAmnt | 贷款金额 | delinquency\_2years | 借款人过去2年信用档案中逾期30天以上的违约事件数 |
| term | 贷款期限（year） | ficoRangeLow | 借款人在贷款发放时的fico所属的下限范围 |
| interestRate | 贷款利率 | ficoRangeHigh | 借款人在贷款发放时的fico所属的上限范围 |
| installment | 分期付款金额 | openAcc | 借款人信用档案中未结信用额度的数量 |
| grade | 贷款等级 | pubRec | 贬损公共记录的数量 |
| subgrade | 贷款等级之子级 | pubRecBankruptcies | 公开记录清除的数量 |
| employmentTitle | 就业职称 | revolBal | 信贷周转余额合计 |
| employmentLength | 就业年限（年） | revolUtil | 循环额度利用率，或借款人使用的相对于所有可用循环信贷的信贷金额 |
| homeownership | 借款人在登记时提供的房屋所有权状况 | totalAcc | 借款人信用档案中当前的信用额度总数 |
| annuallncome | 年收入 | initialListStatus | 贷款的初始列表状态 |
| verificationStatus | 验证状态 | applicationType | 表明贷款是个人申请还是与两个共同借款人的联合申请 |
| issueDate | 贷款发放月份 | earliesCreditLine | 借款人最早报告的信用额度开立的月份 |
| purpose | 借款人在贷款申请时的贷款用途类别 | title | 借款人提供的贷款名称 |
| postCode | 借款人在贷款申请中提供的邮政编码的前3位数字 | policyCode | 公开可用的策略\_代码=1新产品不公开可用的策略\_代码=2 |
| regionCode | 地区编码 | n系列匿名特征 | 匿名特征n0-n14，为一些贷款人行为计数特征的处理 |

2.2 评价指标

**2.2.1 混淆矩阵 Confuse Matrix**

* True Positive：实例为正，预测为正；
* False Negative：实例为正，预测为负；
* False Positive：实例为负，预测为正；
* True Negative：实例为负，预测为负。

**2.2.2 准确率 Accuracy**

Accuracy不适用于样本分布不均衡的情况（正样本过多或负样本过多）。

**2.2.3 精确率 Precision**

Precision为所有预测为正样本中真实为正的占比。

**2.2.4 召回率 Recall**

Recall为所有正样本中被预测为正的占比。

**2.2.5 F1 Score**

Precision和recall是相互影响的，精确率提高则召回率下降。F1 Score作为精确率和召回率的调和平均数，很好地综合了两者的信息。

**2.2.6 P-R曲线（Precision-Recall Curve）**

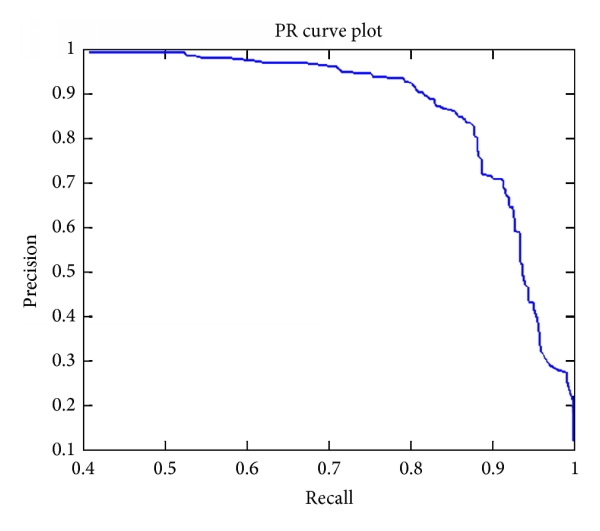


图2.1 P-R曲线示意图

Precision和recall是相互影响的，精确率提高则召回率下降。F1 Score作为精确率和召回率的调和平均数，很好地综合了两者的信息。图2.1展示了P-R曲线样例，可见Precision和Recall存在反相关关系。

**2.2.7 ROC（Receiver Operating Characteristic）**

ROC（receiver operating characteristic）曲线说明了二元分类器系统的鉴别阈值变化时的诊断能力，它是根据一系列不同的二分类方式（分界值或决定阈），以真阳性率（TPR，正样本中预测为正占比）为y轴，假阳性率（FPR，负样本中预测为正占比）为x轴绘制而成，样例如图2.2所示。

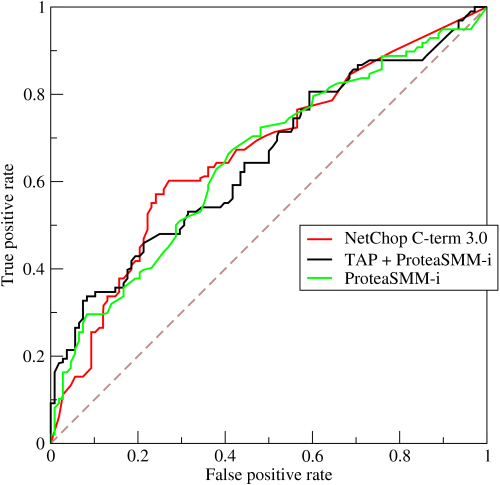


图2.2 ROC示意图

**2.2.8 AUC（Area Under Curve）**

AUC（Area Under Curve）被定义为ROC曲线与坐标轴围成的面积。使用AUC值作为模型的评价标准是因为很多时候ROC曲线并不能清晰地说明哪个分类器效果更好，而作为一个数值，对应AUC更大的分类器效果更好。AUC就是衡量学习器优劣的一种性能指标。

在金融风控领域中，K-S（Kolmogorov-Smirnov）曲线与ROC曲线类似但稍有区别：K-S曲线将TPR和FPR作为y轴，以选定的阈值作为x轴，公式为：；KS值小于0.2认为模型没有区分能力；KS值处于[0.2, 0.3]认为模型有一定区分能力；KS值处于[0.3, 0.5]认为模型有较强区分能力；KS值大于0.75认为模型异常。

2.3 数据读取及评价指标实现

CSV文件数据的读取可由Pandas轻松实现，而评价指标都可以在sklearn中找到已有的api。数据读取结果如图2.3所示，调用api实现评价指标样例结果如图2.4~2.6所示：

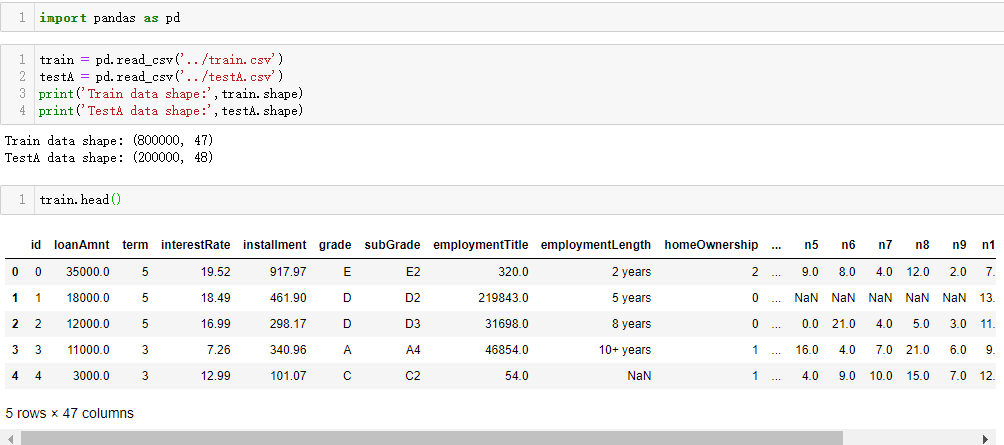


图2.3 训练集及测试集数据读取

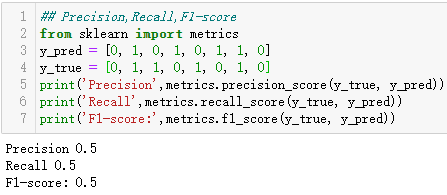
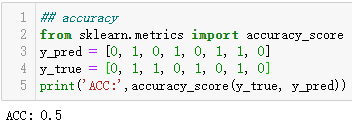
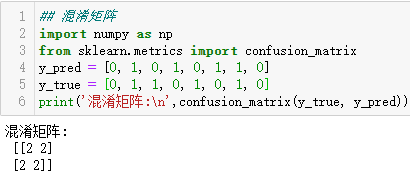


图2.4 混淆矩阵，准确率，精确率，召回率及F1-score调用样例

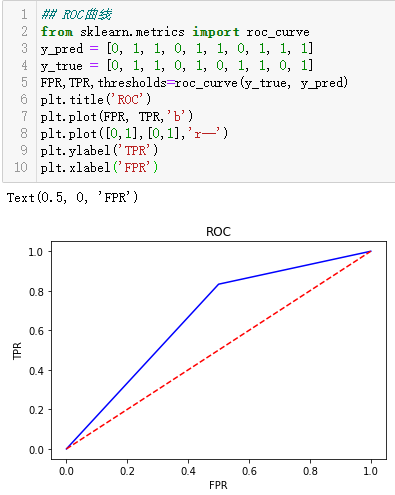
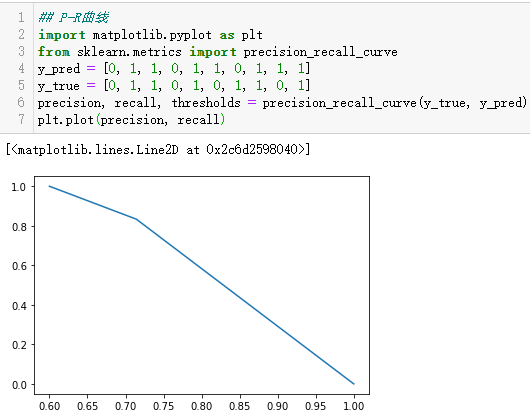


图2.5 P-R，ROC曲线调用样例

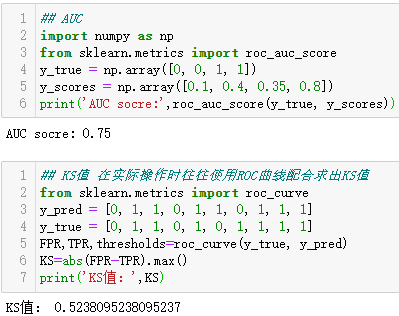


图2.6 AUC，KS值调用样例

3. 数据分析

数据分析是十分重要的，可以根据数据分析得到一定的结论，然后根据这些结论来制定出相应的方案，有的时候根据数据分析可以得到出现错误的原因，可以及时的修正，但是这些前提是，得到的数据必须是真实的，数据分析可以帮助人们做出判断。

3.1 数据总体信息

pandas dataframe 拥有许多attributes和functions，借助 shape, columns, info(), describe() 等可以对数据整体有很好的了解：



图2.7 Training & testing datasets shape

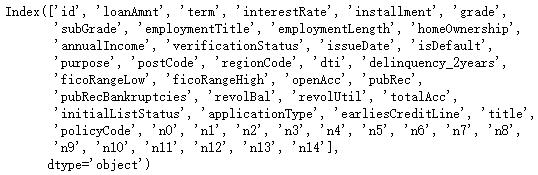


图3.1 Features in dataset

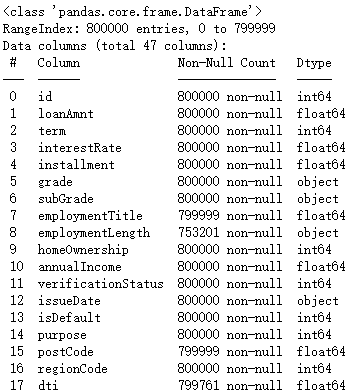


图3.2 Information of dataset

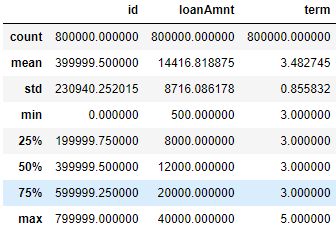


图3.3 Description of features

3.2 缺失值与唯一值

借助 isnull() 函数可得到关于数据是否为空的bool matrix：

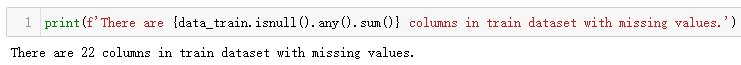


图3.4 包含缺失值features个数统计

数据中尚不存在缺失率大于50%的特征。进一步，还可以对 features 的缺失值情况进行可视化：

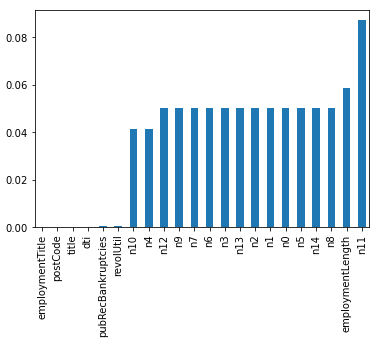


图3.5 Features缺失值统计柱状图

对于缺失值过高的feature，可以认定它对于最终分类的影响是微小的，可以直接删去该feature。其次，通过唯一值分析，发现training和testing数据中存在名为 ‘policyCode’ 的feature具有唯一值，也可直接删去。

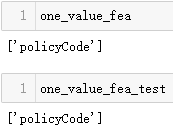


图3.6 Features唯一值分析

通过分析feature的数据类型也可以对数据整体有更好地了解：

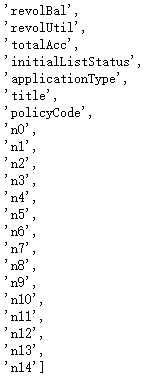
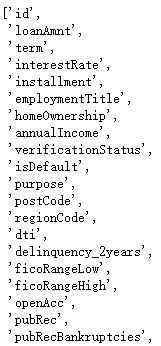


图3.7 数值类型Features



图3.8 类别类型Features

更进一步，对于数值类型feature还可分成连续型与离散型：

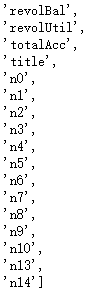
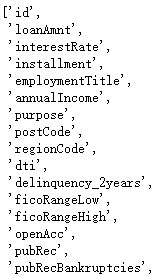


图3.9 连续型Features

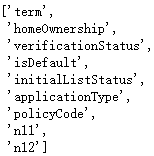


图3.10 离散型Features

借助value\_counts()可以对离散型feature的取值进行分析。对于具有单一值和极差较大的feature可以直接抛去。

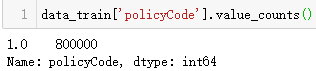


图3.11 具有唯一值的policyCode

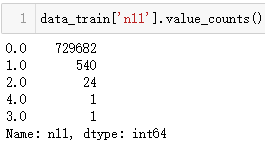


图3.12 具有较大极差的n11

对于连续型数值变量可以对其分布进行可视化

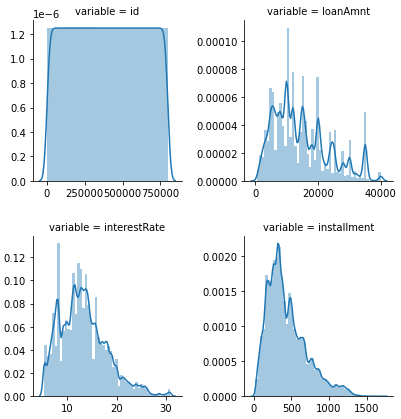


图3.13 连续型features分布

对于不符合正态分布的变量可通过对数log变换，之后再进行观察。一些情形中正态分布可以使得算法收敛速度加快，一些模型更是要求数据具有正太分布，有些只需要数据不要过于具有偏态（可能会影响预测结果）。

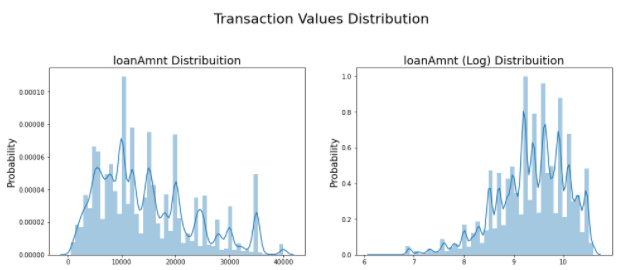


图3.14 对loanAmnt进行log变换

对于非数值型变量，则可直接用value\_counts()进行分析并使用柱状图可视化：

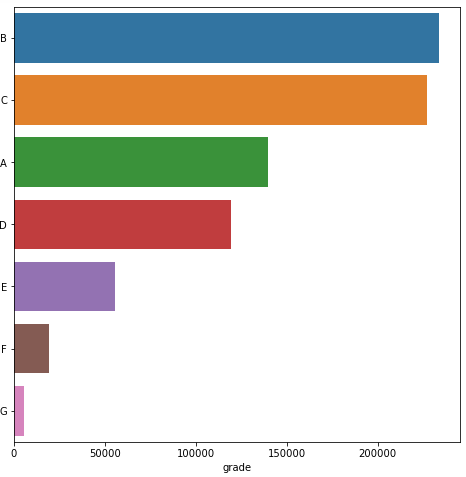
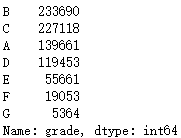


图3.15 对非数值grade变量分析

此外，在training set中还可以根据label的不同对数据进行筛选，分析以及可视化：

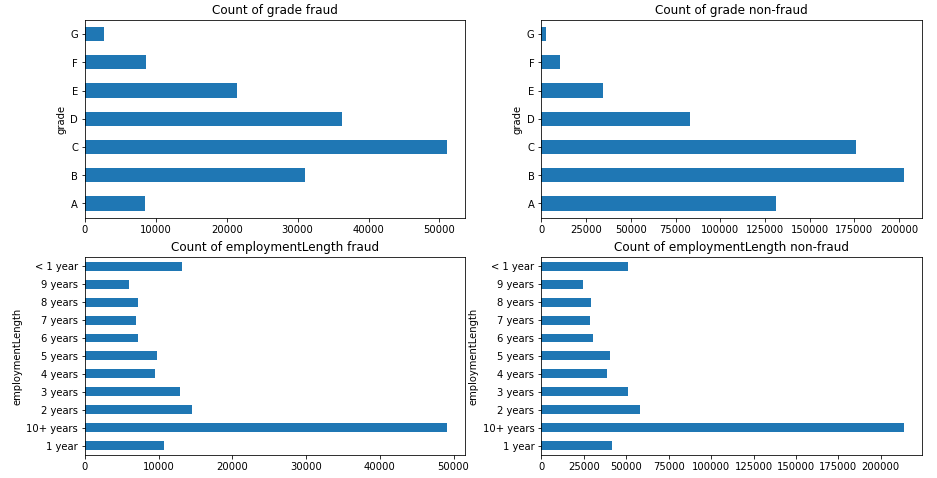


图3.16 基于不同分类的非数值feature数据分析

对于连续变量也可查看对于不同分类，其分布情况

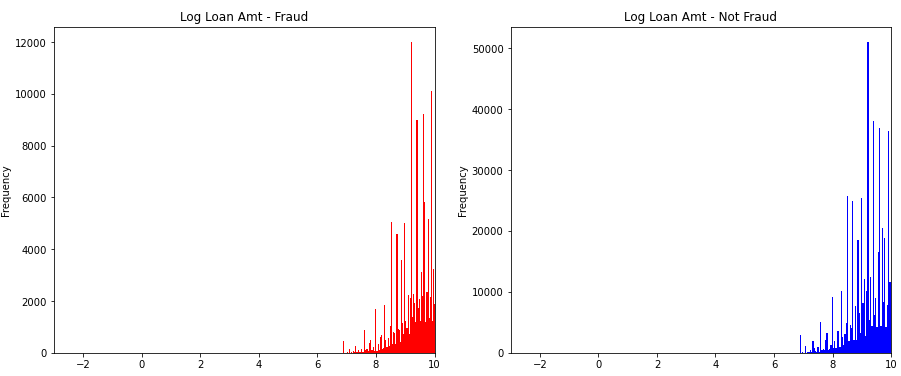


图3.16 基于不同分类的连续型数值feature分布图

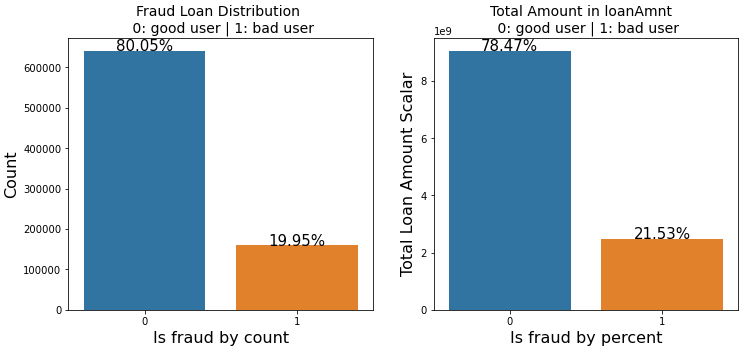


图3.17 基于不同分类的连续型数值feature柱状图

对于时间类型的数据也可以进行可视化和分析：

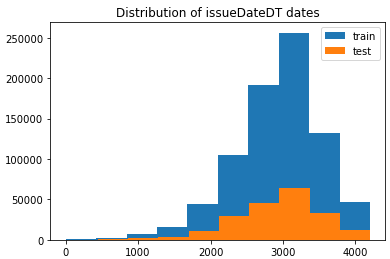


图3.18 时间类型变量issueDateDT可视化

可以看到training和testing data的时间变量是有很大交集的，则该feature很有可能对于最终的分类没有太多贡献。

数据透视表可以动态地改变它们的版面布置，以便按照不同方式分析数据，也可以重新安排行号、列标和页字段。每一次改变版面布置时，数据透视表会立即按照新的布置重新计算数据。另外，如果原始数据发生更改，则可以更新数据透视表。

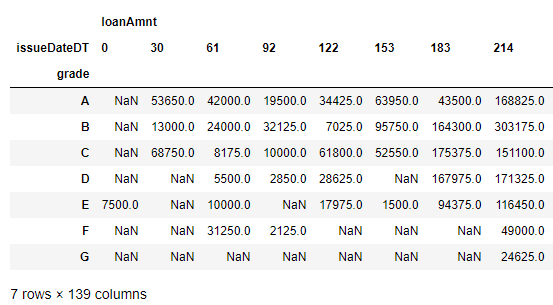


图3.19 数据透视表示例

此外，还可以借助pandas\_profiling来生成dataframe的summary，其中包括了各个变量信息，以及变量间的统计信息。

4. 特征工程

特征工程指的是把原始数据转变为模型的训练数据的过程，它的目的就是获取更好的训练数据特征，使得模型的性能得到提升，有时甚至在简单的模型上也能取得不错的效果。特征工程在机器学习中占有非常重要的作用，一般认为包括特征构建、特征提取、特征选择三个部分。特征构建比较麻烦，需要一定的经验。 特征提取与特征选择都是为了从原始特征中找出最有效的特征。它们之间的区别是特征提取强调通过特征转换的方式得到一组具有明显物理或统计意义的特征；而特征选择是从特征集合中挑选一组具有明显物理或统计意义的特征子集。两者都能帮助减少特征的维度、数据冗余，特征提取有时能发现更有意义的特征属性，特征选择的过程经常能表示出每个特征的重要性对于模型构建的重要性。

4.1 缺失值填充

首先可以查询数据中的对象特征和数值特征：

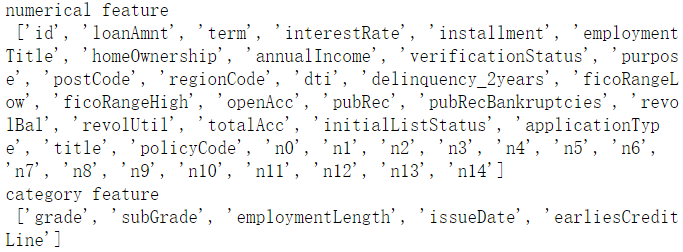


图4.1 对象特征与数值特征

* 缺失值替换为0：data\_train = data\_train.fillna(0)；
* 缺失值上方值复制：data\_train = data\_train.fillna(axis=0,method='ffill')
* 缺失值下方值复制：data\_train = data\_train.fillna(axis=0,method='bfill',limit=2)

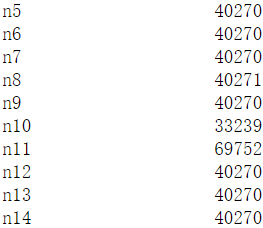
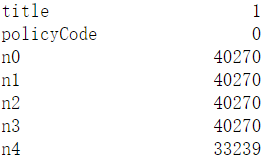
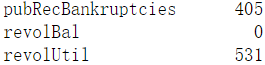


图4.2 存在缺失值的特征

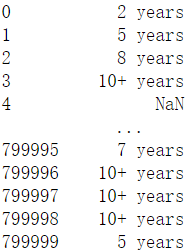


图4.3 ‘employmentLength’未被填充

对数值特征使用中位数进行填充，对对象特征使用众数进行填充后，发现对象特征‘employmentLength’未完成填充，且需要对时间特征‘issueDate’和‘earliestCreditLine’进行数据类型转换操作：将‘issueDate’转换成时间间隔天数，将‘earliestCreditLine’转换成后四位代表的年份。

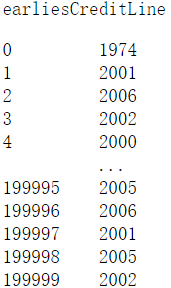
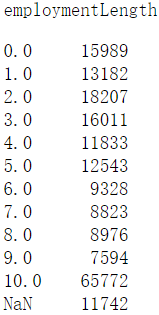
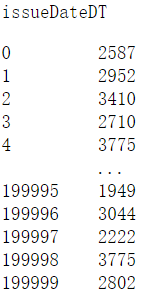


图4.4 特殊特征处理结果

对于其余的类别特征也可进一步处理，对于等级类特征，可以进行label encode或自映射：

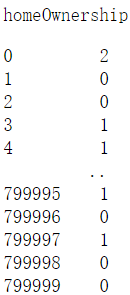
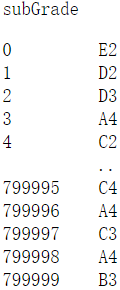
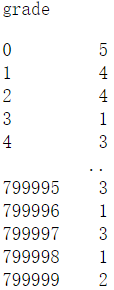


图4.5 类别特征处理结果

4.2 异常值处理

检测异常的方法有：均方差及箱型图：

1. 均方差

依据3-sigma原则，对于近似正态分布的数据，数据约68% 在内，约95% 在内，约99.7% 在内。

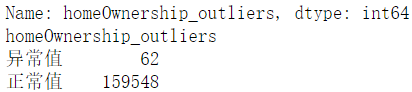


图4.6 均方差检验样例

4.3 数据分桶

数据存在极值过大的特征，对于KNN之类使用欧氏距离作为相似度衡量的算法来说，这些特征都会造成大值覆盖小值的结果。而特征分桶的目的就是为了降低变量复杂度，减少噪音对算法和模型的影响，提高自变量和因变量之间的相关度，提高模型鲁棒性。数据分桶的内容包括连续变量离散化和多状态离散变量合并为少状态。数据分箱具有很多优势：

* 处理缺失值：若特征存在缺失值，则可以把null作为单独一个分箱。
* 处理异常值：对于数据中的outlier，可以把他通过分箱进行离散化处理，从而提高鲁棒性。
* 可解释性：x，y之间常常存在非线性关系，此时可利用WOE（Weight of Evidence）变换。

分箱有如下基本原则：

* 最小分箱占比不低于5%。
* 箱内不能全是y = 0。
* 连续箱单调。

1. 固定宽度分箱

对于跨度较大（横跨多个量级）的特征，可以按照幂级数来分组。如果计数值中有比较大的缺口时，会出现很多空箱子。

2. 分位数分箱

按照数据的分位数对特征数据进行分箱，更符合统计分布。

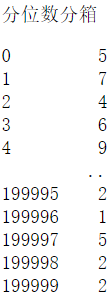
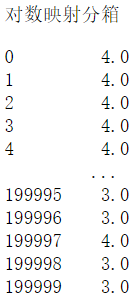
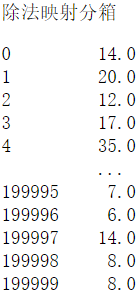


图4.7 数据分箱结果

4.4 特征交互

想要丰富特征，特别是对于线性模型而言，除了分箱外，另一种方法是添加原始数据的交互特征和多项式特征。例如对于每一个grade都对应着default的个数，由此可以生成新的特征每个grade中default的均值：

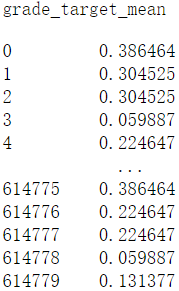


图4.8 生成交互特征‘grade\_target\_mean’

或是对于匿名特征n\*，可以生成每个值对应的grade水平：

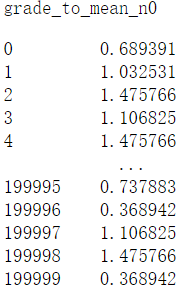


图4.9 生成衍生特征‘grade\_to\_mean\_n0’

4.5 特征编码

对于离散型数据，可以借助LabelEncoder将其转换成0至n-1之间的数。

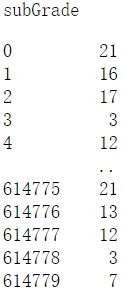


图4.10 ‘subGrade’ labelencode 编码

而对于Logistic Regression等模型还需要额外操作：特征归一化，去除相关性高的特征。归一化使得训练过程收敛速度更快，避免大吃小问题；去除相关性使得模型可解释性提高，并加快了预测过程。

4.6 特征选择

特征选择是特征工程里的一个重要问题，其目标是寻找最优特征子集。特征选择能剔除不相关(irrelevant)或冗余(redundant )的特征，从而达到减少特征个数，提高模型精确度，减少运行时间的目的。另一方面，选取出真正相关的特征简化模型，可以协助理解数据产生的过程。

特征选择方法有：

* Filter
* 方差选择法
* 相关系数（Pearson相关系数）法
* 卡方检验
* 互信息法
* Wrapper（RFE）
* 递归特征消除法
* Embedded
* 基于惩罚项的特征选择法
* 基于树模型的特征选择

**4.6.1 Filter**

Filter是基于特征间的关系进行筛选：

1. 方差选择法：计算特征方差，然后根据阈值选择方差大于阈值的特征

2. 相关系数法：皮尔森相关系数是一种最简单的可以帮助理解特征和响应变量之间关系的方法，衡量的是变量之间的线性相关性。值域为-1到1，+1表示完全正相关，-1表示完全负相关，0变时无线性相关。

3. 卡方检验：用于检验自变量对因变量的相关性。自变量有N种取值，因变量有M种取值，考虑自变量等于i且因变量等于j的样本频数的观察值与期望的差距。统计量为：，其中A为实际值，T为理论值。

4. 互信息法：用于评价自变量对因变量的相关性。在feature\_selection库的SelectBest类结合最大信息系数法可以用于选择特征。

**4.6.2 Wrapper**

递归特征消除法：使用一个基模型来进行多轮训练，每轮训练之后消除若干权值系数的特征，再基于新特征进行下一轮训练。在feature\_selection库的RFE类可以用于选择特征。

**4.6.3 Embedded**

1. 基于惩罚项的特征选择法：使用带惩罚项的基模型，除了筛选特征外，同时进行降维。在feature\_selection库的SelectFromModel类结合逻辑回归模型可以用于选择特征。

2. 基于树模型的特征选择法：树模型中的GBDT也可以用来作为基模型进行特征选择。在feature\_selection库的SelectFromModel类结合GBDT模型可以用于选择特征。

本数据经过非入模特征剔除和缺失值填充后，可以通过计算协方差矩阵来观察特征间的相关性：

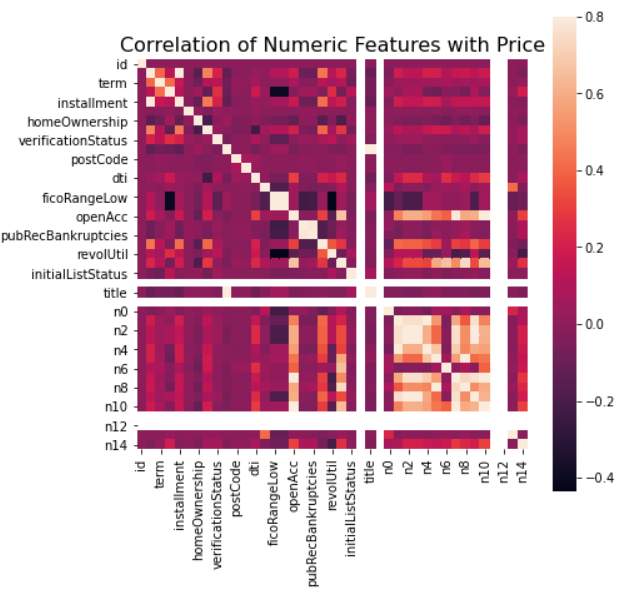


图4.10 ‘subGrade’ labelencode 编码

5. 建模与调参

5.1 常用模型

在金融风控领域常用模型包括逻辑回归，树以及集成模型。它们各具优缺点：

**5.1.1 逻辑回归**

优点：

* 训练速度快，计算量仅和特征数目相关；
* 模型可解释性好，从权重可看出特征对结果的影响；
* 适合binary classification；
* 占用内存小，只需储存各个维度特征值。

缺点：

* 需要提前进行数据处理；
* 无法解决非线性问题；
* 对多重共线性敏感，难以处理数据不平衡问题；
* 准确率低，难以拟合数据真实分布。

**5.1.2 决策树**

优点：

* 模型直观，方便可视化；
* 数据无需预处理，不需要归一化，不需要处理缺失值；
* 离散值和连续值都能处理；

缺点：

* 易过拟合，泛化能力不强（可通过剪枝改善）；
* 使用贪心算法，易得到局部最优解。

**5.1.3 Ensemble Method**

集成学习中，会训练多个模型（弱学习器）解决相同问题，并将它们结合起来以获得更好结果。其中的重要假设为：当弱模型被正确组合时，我们可以得到更精确和/或更具鲁棒性的模型。大多数情况中，这些弱学习器本身性能并不好（high bias / high variance）。集成方法的思想是通过将这些弱学习器的bias / variance结合起来，从而创建一个“强学习器”以获得更好的性能。

集成方法主要包括Bagging和Boosting，常见基于Bagging思想的集成模型有：随机森林；基于Boosting思想的集成模型有：Adaboost、GBDT、XgBoost以及LightGBM等。Baggin和Boosting的区别总结如下：

* 样本选择上：Bagging方法的训练集是从原数据集有放回的选取，所以从原始集中选出的各轮训练集之间是相互独立的；Boosting方法需要每轮训练集不变，而是每个样本在分类器中的权重发生变化。而权值是根据上一轮的分类结果进行调整的；
* 样例权重上：Bagging方法使用均匀取样，每个样本权重相等；Boosting方法根据错误率不断调整样本权重，错误率越大权重越大；
* 预测函数上：Bagging方法中所有预测函数的权重相等；Boosting方法中每个弱分类器都有相应权重，对于分类误差小的分类器有更大的权重；
* 并行计算上：Bagging方法中各个预测函数可以并行生成；Boosting方法各个预测函数只能顺序生成，因为后一个模型参数需要前一轮模型的结果。

5.2 数据集划分

训练集上的误差称之为训练误差或经验误差；测试集上的误差称之为测试误差。对于数据集的划分通常需要满足两个条件：

* Training和testing分布要与样本真实分布一致，从样本真实分布中独立同分布采样而得；
* Training和testing互斥。

数据划分方法有：留出法、交叉验证法以及自助法：

* 留出法：直接将数据集划分为两个互斥的集合。需要注意的是在划分的时候尽可能保证数据分布的一致性，常采用分层采样的方式。
* 交叉验证法：k折交叉验证通常将数据集分为k份，k-1份作为training，剩余一份为testing，最终返回k个测试结果的均值（k=1时为留一法）。
* 自助法：每次从数据集中去一个样本作为训练集中的元素，然后放回，重复该操作m次，把没有出现过的样本作为测试集。这样的采样方式会使得数据集中约有36.8%的数据没有在训练集中出现过。留出法和交叉验证法都是分层采样，自助法是有放回的重复采样。

当数据量充足时，常采用留出法和k折交叉验证法；当数据量小且难有效划分时，使用自助法；当数据量小且可以有效划分时，使用留一法。

5.3 LightGBM Model

GBDT（Gradient Boosting Decision Tree）常用于多分类、点击率预测、搜索排序等任务，据统计Kaggle上比赛有一半以上的冠军方案都是基于GBDT的。LightGBM是实现GBDT算法的框架，支持高效率的并行训练，且具有更快的训练速度，更低的内存消耗，更好的准确率，支持分布式可以快速处理海量数据等优点。

常见机器学习算法（NN等）都可以使用mini-batch方式训练，不受内存限制。而GBDT在每一次迭代的时候都需要遍历整个训练数据多次。LightGBM提出的主要原因就是为了解决GBDT在海量数据遇到的问题，让GBDT可以更好更快地用于工业实践。

LightGBM具有以下优缺点：

优点：

* 采用直方图算法将遍历样本转变为遍历直方图，极大降低了时间复杂度；
* 训练过程中采用单边梯度算法过滤掉梯度小的样本，减少计算量；
* 采用了基于Leaf-wise算法的增长策略构建树，减少了不必要的计算量；
* 采用优化后的特征并行、数据并行方法加速计算，当数据量非常大的时候还可以采用投票并行。
* 对缓存也进行了优化，增加了缓存命中率；
* XGBoost使用与排序后需要记录特征值及其对应样本的统计值的索引，LightGBM使用了直方图算法将特征值转变为bin值，且不需要记录特征到样本的索引，将空间复杂度从O(2#data)降到O(bin)；
* 在训练过程中采用互斥特征捆绑算法减少了特征数量，降低了内存消耗。

缺点：

* 可能会训练出过深的决策树，产生过拟合。因此LightGBM在Leaf-wise上增加了最大深度限制，保证高效的同时防止过拟合；
* Boosting族是迭代算法，每次迭代都根据上一次迭代的预测结果对样本进行权重调整，随着迭代不断增加，误差会越来越小，bias降低。LightGBM是基于偏差的算法，所以会对噪点敏感；
* 寻找最优解时，依据的是最优切分变量，没有将最优解是全部特征的综合考虑进去。

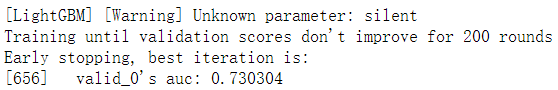


图5.1 LightGBM训练及验证结果

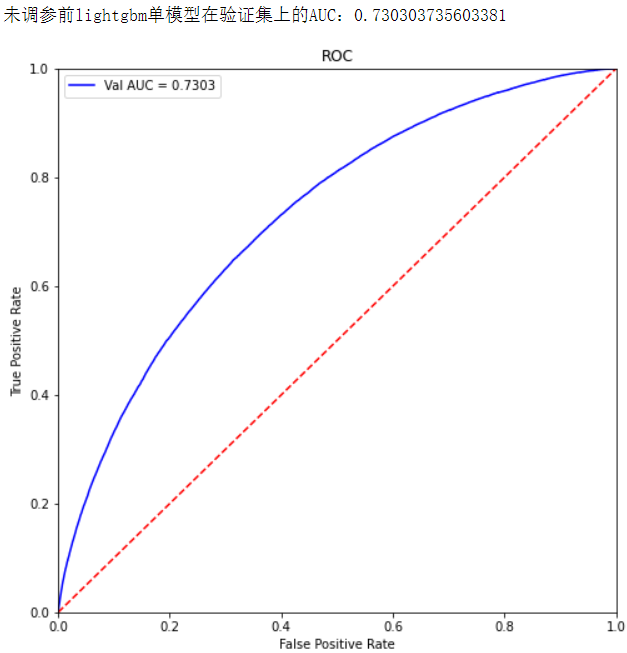


图5.2 LightGBM单模型ROC、AUC结果

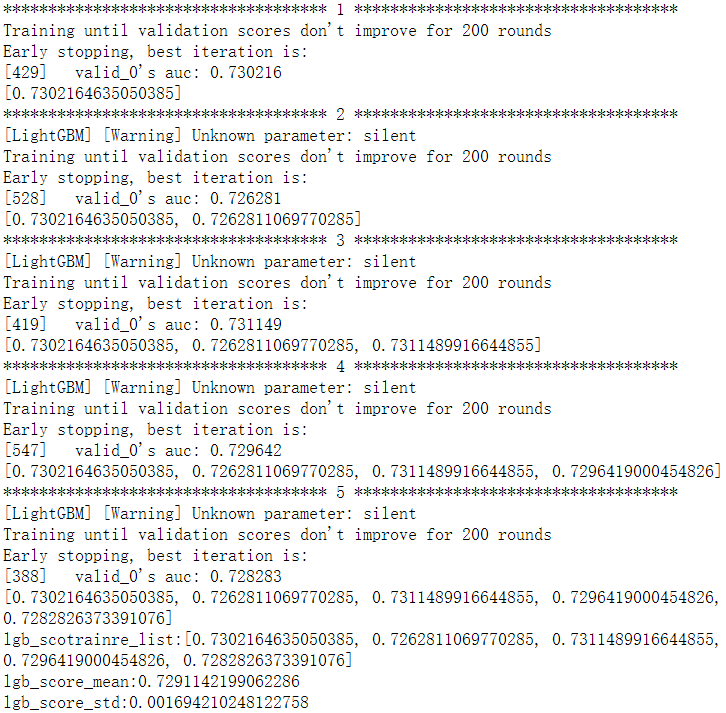


图5.3 5折LightGBM模型训练及验证结果

成功构建LightGBM后，使用贝叶斯调参得到最优参数：

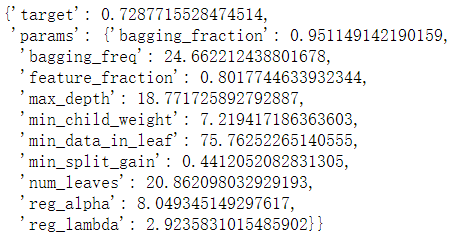


图5.4 贝叶斯调参最优参数结果

带入最优参数再次训练，得到最优迭代次数为13176，最终模型AUC为0.732。将模型带入到验证集：

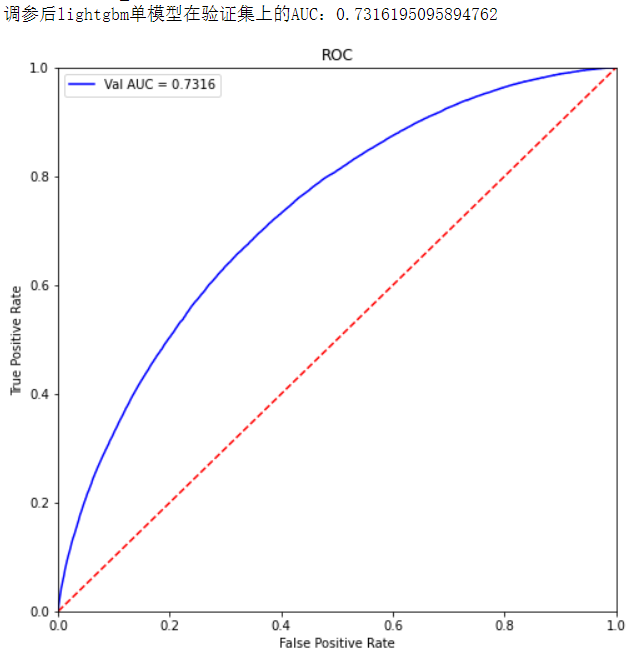


图5.5 调参后模型于验证集上结果

6. 模型融合

6.1 常见方法

* 平均
* 简单平均法
* 加权平均法
* 投票
* 简单投票法
* 加权投票法
* 综合
* 排序融合
* Log融合
* stacking：构建多层模型，利用预测结果再拟合预测
* blending：选取部分数据预测得到结果作为新特征带入剩下的数据中预测
* boosting/bagging

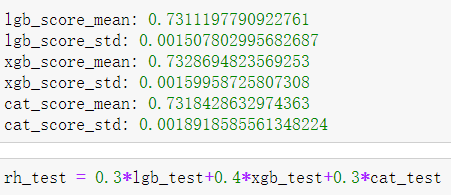


图6.1 使用加权平均进行模型融合调参后模型于验证集上结果

6.2 stacking & blending

Stacking是把若干基学习器的预测结果作为新的训练集来训练一个学习器。Blending是将预测值作为新的特征与原特征合并，构成新的特征用于预测。两者的差异如下：

* stacking
* stacking的两层使用数据不同，可以避免信息泄露；
* 无需共享seed；
* blending
* blending将数据分为两部分，在最后预测时部分信息将被忽略；
* 第二层数据较少，易产生过拟合。