过采样与极端梯度提升在信用评估领域的应用

衡强 刘宇扬 孙爽

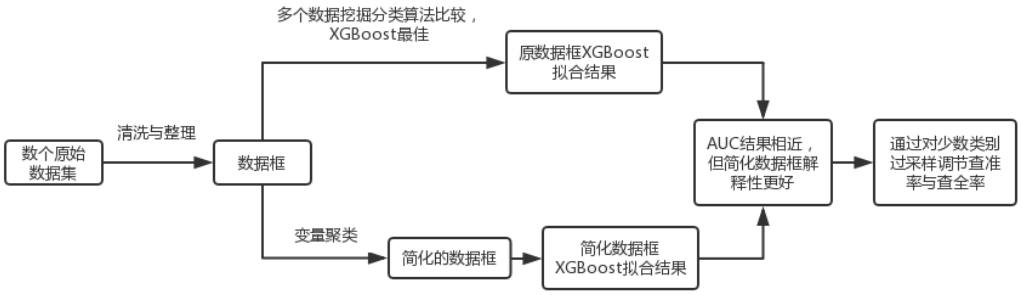
指导老师：吴纯杰

统计与管理学院

上海财经大学

参赛编号：1137

摘要:在信用评估领域，数据挖掘算法的应用成为了新趋势。本次大赛的选题二给出了一个典型的有监督二分类问题。经过对大赛提供的数个文件的清洗和特征提取，我们构造了一个具有 40000 条观测，121 个解释变量以及一个因变量的数据集。我们首先通过过采样解决原数据集因变量不平衡的问题，并使得相关算法对潜在违约风险的客户样本更加敏感。接着我们在扩展后的数据集上尝试了多个数据挖掘算法并进行了相互比较，极端梯度提升(XGBoost) 在运算速度，AUC 方面均取得了最佳表现。我们同时尝试了将多模型进行集成，但结果相较 XGBoost 单一模型效果提升并不显著。我们提出可以通过过采样比例的调节调整模型预测的各项指标，如查准与查全率等。为了提升模型的解释性，我们尝试了进行变量聚类，并发现聚类后的数据集取得了更佳的 AUC 表现。
关键词 不平衡分类 过采样 极端梯度提升 变量聚类



1 文献综述

信用评分是金融机构进行风险管理的基础。因此，对信用风险进行有效的评价，进而合理地控制风险对金融机构来说至关重要，甚至对有效识别信用风险、规避金融危机及保持信贷金融市场的正常运转都有重要意义。个人信用评分是指金融机构根据客户的各种历史信用资料，利用一定的信用评分模型，对客户的信用等级进行评价，并据此分析客户按时还款的可能性，决定对该客户是否授信以及授信的额度和利率。金融机构的传统做法是由专家基于个人经验对个人信用进行判断。自上世纪 40 年代开始，部分美国银行陆续开始进行各类信用评分方法的实验，致力于提供一种可以大量处理信贷申请的工具。

近年来，随着金融机构获取数据的规模日益扩大与相关算法的日益成熟，数据挖掘相关算法在信用评估领域得到了越来越广泛的应用。线性判别式分析 (Linear Discriminant Analysis, LDA) 最早由 Fisher (1936) 提出 [1]。Durand (1941) 将此方法用于信用评分[2]，将贷款分为“好”贷款和“坏”贷款，首次正式地提出使用统计方法辅助授信决策的观念，这是个人信用评估由定性分析逐步过渡到定量分析的开端。1956 年，工程师 Bill Fair 和数学家 Earl Isaac 利用判别分析法共同发明了著名的 FICO 评分方法，并成立了 Fair Isaac 公司，成为世界上第一家提供信用评分数学模型的公司，该公司于 1958 年发布了第一套信用评分系统。经过不断的发展与完善，此 FICO 评分体系已被各金融机构广泛使用。Wiginton(1980) 对 Logistic 回归模型应用于信用评分的效果进行了分析，认为该方法优于判别分析 [3]。Chatterjee 和 Barcun(1970) 最先将最近邻法引用于建立个人信用评分模型 [4]。AdaBoost 是一种通过集成一系列弱分类器的分类算法，Lessmann 等 (2015) 研究表明 AdaBoost 方法的精确度与泛化能力显著优于单一的信用评价方法 [5]。XGBoost 是陈天奇 (2016) 提出并实现的序列集成算法 [6]，近年来备受关注，在 Kaggle 平台上的各大比赛表现引人注目。XGBoost 在梯度提升的目标函数中引入的新的惩罚项，模型拟合能力与泛化能力均十分出色。本文将尝试将这一算法运用在存在潜在违约风险的贷款识别上。

2 数据清洗与描述分析

2.1 特征提取

大赛提供了数个文件，每个文件中都有报告编号这一变量。为了使问题更符合传统视角，方便我们的统计分析，我们希望将这数个文件整理成一个数据框，其中每个报告编号对应一个观测，唯一标识某个客户。我们发现文件中共有 40000 个报告编号，其中 30000 个给予了是否存在违约行为的信息。我们的目标是利用 30000 个有标注的观测拟合模 型并在 10000 个未标注的观测上做出预测。对于某些文件中存在的某个报告编号对应多条观测的状况，我们采用了求和，平均等方法构造出新的特征，并对所有的特征依据以下原则进行了筛选。

- 贷款、贷记卡两张表与信用评级的相关数据在之后的未销户贷记卡或者未结清贷款、逾期（透支）信息汇总、贷记卡逾期/透支记录等几张表中均有所体现，所以我们选择从后几张表中提取我们所关心的解释变量。
- 我们发现某些变量方差过小，甚至在所有观测中都是一样的，即不能提供任何有益的信息，例如中信用提示表中的本人声明数目、异议标注数目，查询记录汇总、信贷审批查询记录中的查询次数。引入这些变量并不能显著提升模型的性能，并可能降低某些不适用于高维数据的算法的精度，因此我们对这些变量予以舍弃。
- 我们希望我们所选取的特征能尽可能与贷款申请人或担保人的信用行为特征，身份背景，经济状况相关，以提高模型的解释性并便于后续的实证分析。比如某个贷款申请人或担保人的薪水，曾经的违约记录是我们尤其关心的变量，而贷款发放日期，查询操作员等变量则无关紧要，应予以舍弃。

我们最终提取了 44 个变量来拟合我们的模型，其中有 8 个类别变量，3 个连续变量。python 的 scikit-learn 模块的大多数算法都默认变量都是数值变量，为了兼容这一特征，我们对类别变量采用了独热编码的量化方法。变量名，量化方法，及其现实意义将以表格的形式在附录中呈现。

2.2 缺失特征与填补方法

在我们构造的数据集中，部分分类变量存在一定程度的缺失，而除了薪水以外的数值变量只存在极少的缺失。对于其余数值变量存在缺失的观测，我们直接采用舍弃的策略。舍弃数值变量存在缺失的变量后，训练集还剩 29951 个观测，测试集还剩 9997 个观测。我们所关心的就只剩如何处理分类变量以及薪水变量中存在的大面积缺失问题。

方法	操作	潜在风险
个案剔除法	整体缺失值小于 5% 时，剔除所有存在缺失的变量	信息的丢失
均值替换法	数值变量：平均值；分类变量：众数	假设缺失特征完全随机，引入偏差，并减小了数据集的方差
回归替换法	将缺失变量作为因变量使用回归模型拟	减小了数据集的方差
特殊值填充	在分类变量中将缺失作为一个新的水平	引入偏差
不处理		大多数算法不能容忍缺失值的存在

表 1: 常用数据缺失处理方法及潜在风险

考虑到在收集相关信息时，某些客户不愿意透露相关信息通常而言意味着该客户在这个变量上的指征异于常人，包含了独特的信息。我们在所有存在缺失的分类变量上应用了特殊值填充，即将“缺失”作为一个独立的水平参与独热编码。而在后续的分析中，我们发现薪水变量有很重要的角色，为了尽可能减小填补的偏差，我们将其余变量作为解释变量使用线性回归填补了薪水这一变量。

2.3 数据可视化与描述分析

我们在后续的分析中发现，贷款申请人或担保人所在的地域对预测贷款人的违约情况非常有帮助。我们列出违约比例最低的几个省份。

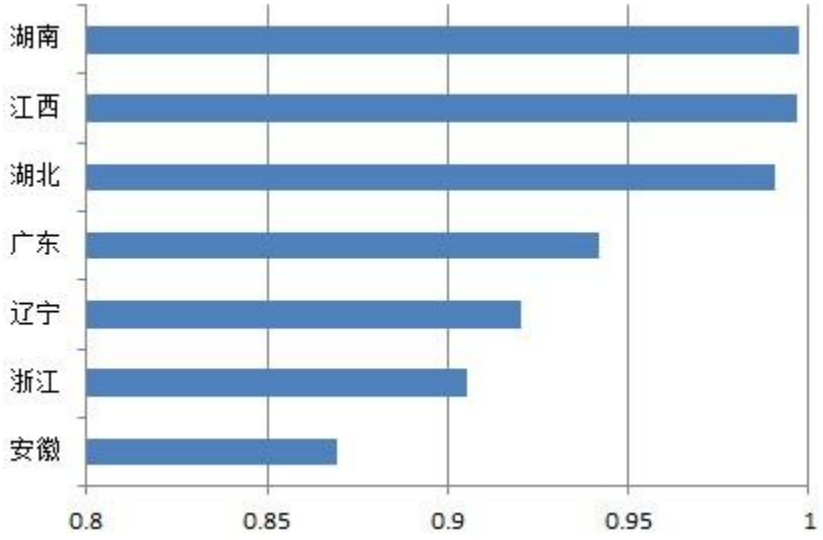


图 1: 未违约比例最高的 7 个省份

我们同时也对不同学历的贷款申请人或担保人的违约比例感兴趣。

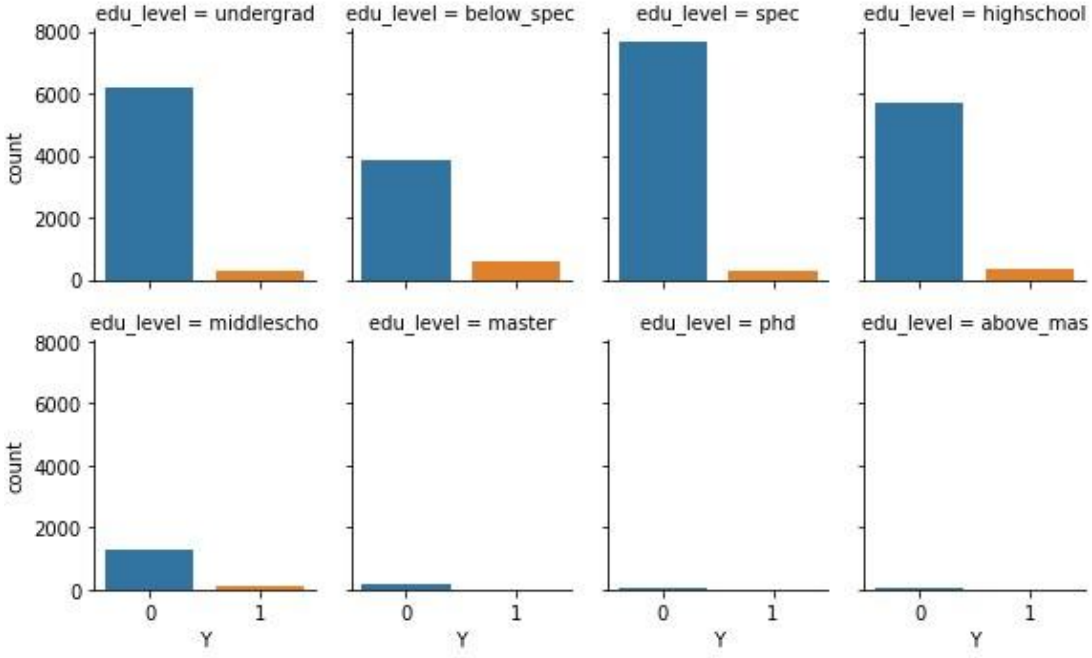


图 2: 各学历水平的违约情况

但违约行为的发生比例并未在不同的学历水平的贷款申请人里展示出明显差距。我们面将关注几个关键的连续变量的核密度估计与箱线图。

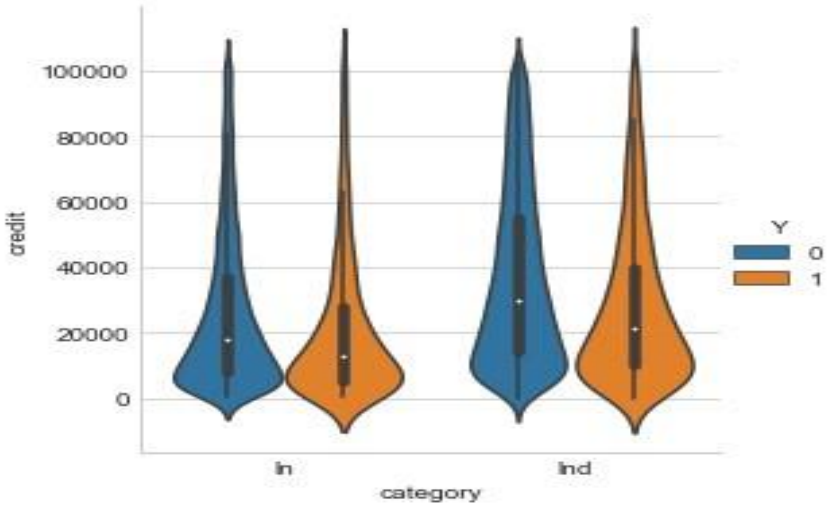


图 3: 违约与未违约客户的贷款与贷记卡合同金额核密度估计与箱线图

我们发现未违约客户的合同金额的分布整体要高于存在违约的行为的客户。同样的，未违约客户的最近六个月平均使用额度要高于存在违约的行为的客户。

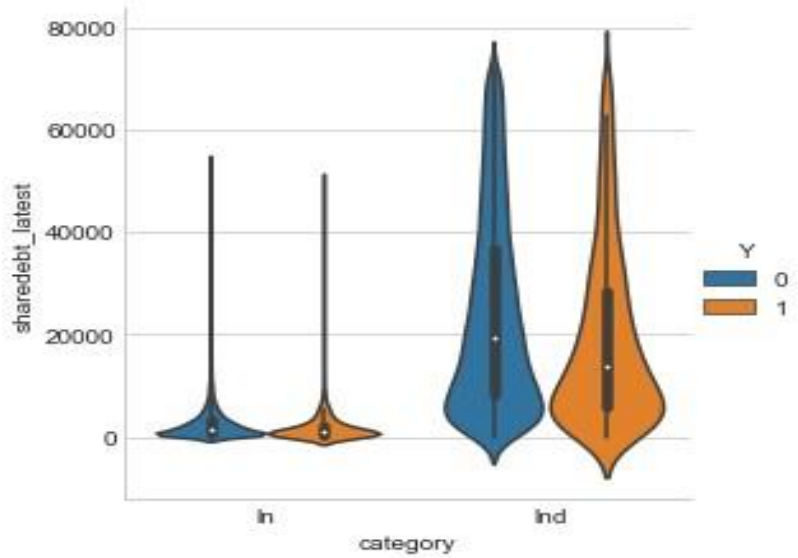


图 4: 违约与未违约客户的最近六个月使用信用额度核密度估计与箱线图

这证明了更活跃的借贷行为展示了更好的信用状况。

2.4 解决因变量分布不平衡的问题

2.4.1 为何要解决因变量不平衡的问题

大多数机器学习分类算法使用了两个假设：各个观测之间是独立的，因变量的分布是平衡的。在本问题中，各个观测之间的独立性并不令人怀疑，但显然因变量的分布极度不平衡。如果这些被错误率驱动的分类算法如果直接运用到不平衡的训练集上，它们会倾向于将所有的观测分类为多数观测 [7]。但显然将所有观测分类为多数观测的预测在这个问题的框架下是毫无意义的，因为我们尤其关心找出那些存在违约风险的客户。

	未违约	违约
比例	93.75%	6.25

表 2: 因变量的分布

事实上，如果我们将训练集按照 8:2 的比例进一步拆分成训练集与验证集（具有 5991 个观测），我们发现在以 0.5 为概率阈值的条件下，我们所应用的算法几乎全部将验证集上的观测预测为多数分类（未违约）。

算法	验证集预测结果（为 1 的个数）
逻辑斯蒂克回归	0/5991
线性判别分析	94/5991
随机森林	0/5991
极端随机森林	0/5991
自适应提升	4/5991
极端梯度提升	7/5991

表 3: 使用原数据集，各算法在验证集上的预测结果

我们期望的理想的结果是各个算法将验证集 6%左右的观测预测为 1，但这里的预测为 1 的数目显然远远小于这一期望值，原因正是前面提到的这些算法以错误率为驱动，而倾向于将所有观测分类为 1。因此我们需要一些方法来解决因变量不平衡的问题。

2.4.2 如何解决因变量不平衡的问题

当前学术界提出的解决不平衡因变量分类的问题的方法可以总结为以下几类:

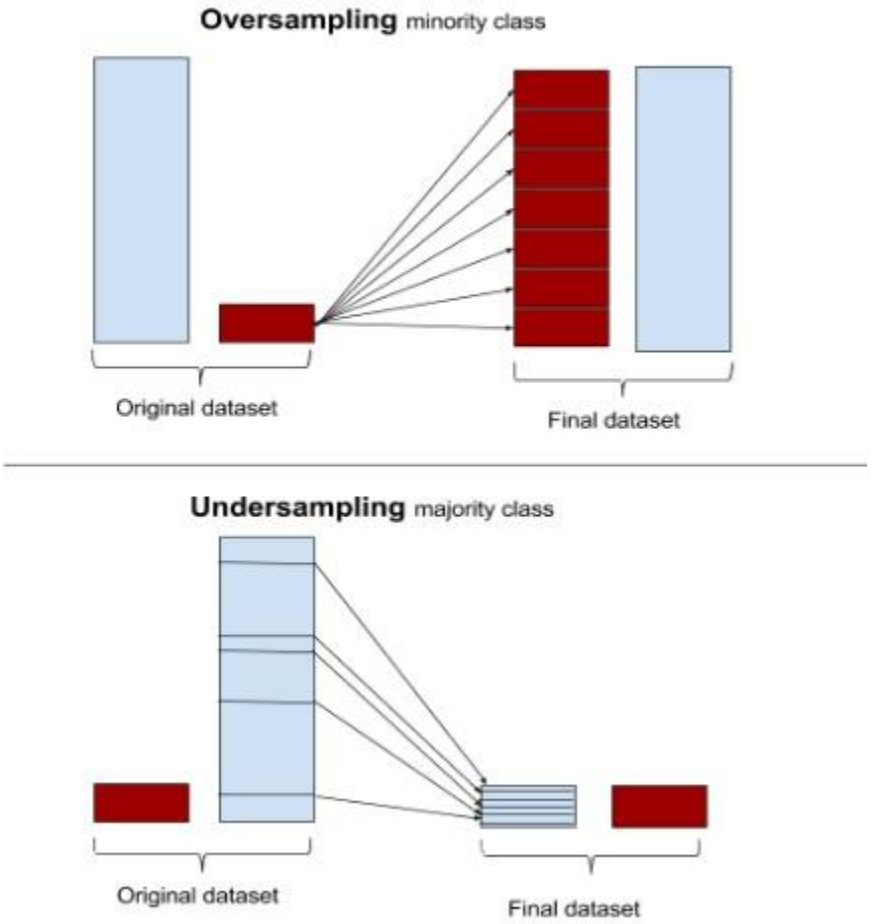


图 5: 过采样与降采样示意图 [9]

- **降采样方法**：通过舍弃多数类别中的观测达到使各个类别平衡的目的，但该方法的缺陷明

显，即丢失了大量的训练样本的信息。

• **过采样方法**：重复采样少数类别中的观测达到使各个类别平衡的目的，但如果直接使用原始的少数类别样本（例如简单随机过采样），模型存在过拟合的风险。早在 2002 年，NV Chawla 等人提出了 SMOTE 算法 [8]，即向从少数类别提取出的样本添加噪音以避免过拟合。

• **代价敏感学习**：一些学者提出通过调整相关目标函数里各个类别的权重达到使模型对少数类别更加敏感的目的。学者们提出了神经网络，支持向量机等的代价敏感学习版本。但权重的确定存在一定的主观性。

我们使用了过采样的方法使训练集中的两个类别观测数目一致。我们尝试了简单过采样与 SMOTE 算法。我们发现在这个问题中，由于随机森林等集成算法具有良好的泛化性能，并不存在过拟合状况，而 SMOTE 算法反而加入了额外的噪音，降低了各个算法的 AUC-ROC，因此我们最终选用了简单过采样。另一个值得注意的问题是，应该将过采样运用在训练集上，而验证集保持因变量分布不平衡的情况。我们衡量算法在验证集上的表现是为了估计模型面对新的数据时的表现，因此我们应该保证验证集里的因变量分布是不平衡的，与现实一致。

3 为何使用 XGBoost

XGBoost 作为当前在大多数机器学习比赛中实现了最先进 (state-of-the-art) 的算法，具有强大的拟合能力与泛化能力。在 Kaggle 举办的上百个机器学习比赛中，超过半数军使用了 XGBoost 或者将 XGBoost 作为集成模型中的一个环节，并且大多数集成模型只能小幅度地提升单一 XGBoost 模型的结果。XGBoost 项目由华盛顿大学的陈天奇博士主持，强大的并行计算实现使它比 scikit-learn 平均而言快 3 倍。XGBoost 同时是一个灵活的框架，允许使用者修改模型底层细节以更好的针对实际问题。

3.1 算法表现的度量指标

大多数问题中，算法的精度是由准确率 (accuracy)，即预测结果与实际结果相符的比例衡量的。但本问题中，仅仅使用准确率是不合适的。如果我们将所有的观测预测 0，我们将会获得 93.75% 的准确率，看似出色但其实毫无意义。为了衡量在不平衡分类问题中算法的表现，我们需要其他指标进行补充。

3.1.1 混淆矩阵

二分类问题分类精度的度量指标有很多，但其中大多数基于如下的混淆矩阵 [10]。

	Predicted Postives	Predicted Negatives
Real Postives	True Positive(TP)	False Negative(FN)
Real Negatives	False Positive(FP)	True Negative(TN)

表 4: 混淆矩阵

3.1.2 AUC-ROC

受试者工作特征曲线 (receiver operating characteristic curve)，又称感受性曲线，是常用的衡量二分类问题预测表现的指标。通过设定一系列阈值，并返回一系列的真阳性率-假阳性率点描述预测的概率的质量。ROC 曲线描绘了任意阈值时，对 1 类别的识别能力。ROC 曲线下方的面积 AUC（Area Under Curve）常被用来衡量预测概率的识别能力。 AUC 越大，意味着所预测概率的识别能力越强。值得注意的是，AUC 只与各个观测概率的排序有关，任何不会改变概率排序的变动都不会影响 AUC。

3.2 所用模型及参数

由于我们使用了 AUC 作为指标，因此我们希望所使用的算法能够产生属于两个类别的

概率。我们尝试了邻近算法（K Nearest Neighbors), 线性判别分析（Linear Discriminant Analysis)，逻辑斯蒂克回归（Logistic Regression), 随机森林（Random Forest)，极端森 林（Extra Trees)，自适应提升(Adaptive Boosting), 极端梯度提升(Extreme Gradient Boosting)。K 邻近算法是找出验证集的观测在训练集中欧几里得距离最小的 K 个观测，预测类别为这 K 个观测类别的众数。线性判别分析在多远正态的假设下通过计算给予解释变量 后两个类别的贝叶斯后验概率进行判别。逻辑斯蒂克回归是业界广泛使用的广义线性模型，通过极大似然估计获得参数与预测概率。随机森林是最直观的集成算法，通过建立多棵决策树并将某一特定观测预测为所有决策树的预测的众数来实现对决策树的集成。如果每棵决策树都只使用从原数据集中自助采样出的一个小样本，并且所能使用的变量限定在所有变量的一个随机子集中，随机森林算法就成为了装袋算法。极端森林算法比装袋算法具有更强的随机性，它在形成决策树时每个分支使用的变量并不是随机子集中最优的变量而是随机选择的一个变量。这种机制更大程度上减小了方差但引入了偏差。自适应提升也是一种集成算法，相比随机森林，自适应提升会随着模型的演变调整各个观测的权重，使得集成模型更加关注那些容易被分类错误的观测。梯度提升一般用于解决回归问题，如果我们将回归的结果解读为概率也可以用于解决分类问题。梯度提升通过建立新的弱分类器拟合残差来实现模型的集成。而极端梯度提升则在梯度提升的目标函数中引入了新的惩罚项，并且 python 和 r 中具有非常强大的并行计算实现。我们在这里呈现模型参数经过调试后的选择。

算法	预处理	弱分类器	参数
KNN	标准化		K=10
LR	去掉了某些存在复共线性的变量		
LDA	去掉了某些存在复共线性的变量		solver='SVD'
RF		决策树	n_estimators=1000,max_depth=10
ET		决策树	n_estimators=1000,max_depth=10
AdaBoost		决策树	n_estimators=1000,max_depth=1 learning_rate=0.1
XGBoost		回归树	n_estimators=1000,max_depth=1 learning_rate=0.1

表 5: 单一模型细节与参数

3.3 过采样 vs 原始数据

我们分别在过采样后的训练集以及未经过过采样的原始训练集上拟合了上述算法。未经过过采样的原始训练集通常会将所有验证集样本预测为 0，因此准确率在 93.7%左右。过采样后的训练集拟合的模型对少数类别（违约的客户）更加敏感，通常会比未经过过采样的原始训练集拟合的模型具有更高的 AUC。但如果以 0.5 为阈值，过采样后的训练集拟合模型会将 20%-30%的验证集观测预测为 1，所以准确率会下降到 70%左右。0.5 的阈值是一个直观的选择，因为过采样已经使训练集中 0 类别与 1 类别数目的观测相等。

算法	过采样前验证集表现		过采样后验证集表现	
	准确率（0.5 为阈值）	AUC	准确率（0.5 为阈值）	AUC
KNN	93.7%	0.682	76.8%	0.729
LR	93.8%	0.586	64.5%	0.609
LDA	93.1%	0.823	70.2%	0.836
RF	93.8%	0.793	79.7%	0.801
ET	93.8%	0.793	78.7%	0.792
AdaBoost	93.8%	0.834	73.3%	0.835

XGBoost	93.8%	0.842	73.2%	0.841

表 6: 过采样前后各算法表现

3.4 XGBoost vs 其他算法

综合查看各算法的表现，XGBoost 展现出了最强大的识别能力，具有最高的 AUC-ROC。

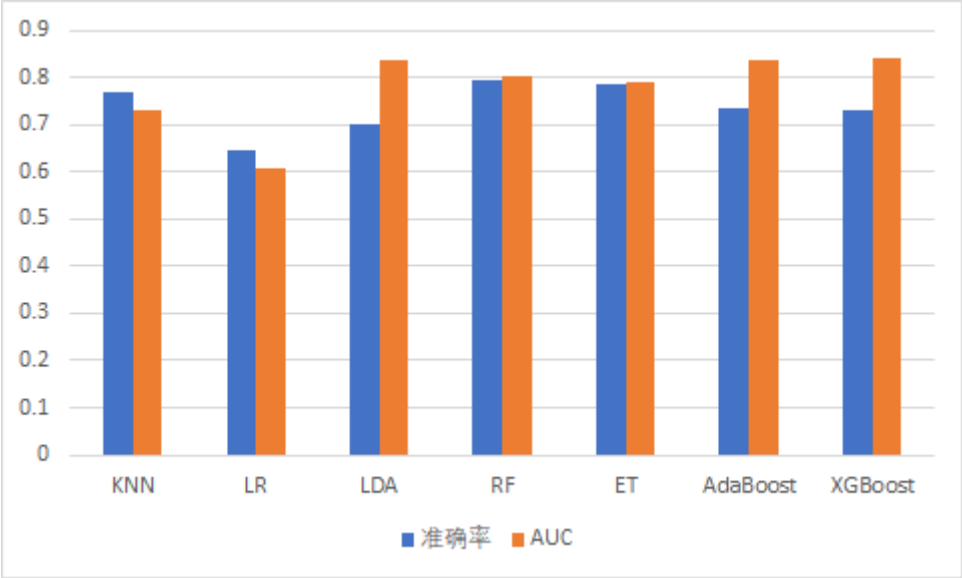


图 6: 各算法在验证集的表现

我们将各个算法的 ROC 曲线画在一张图上以更直观查看各个算法对潜在违约客户的识别能力。

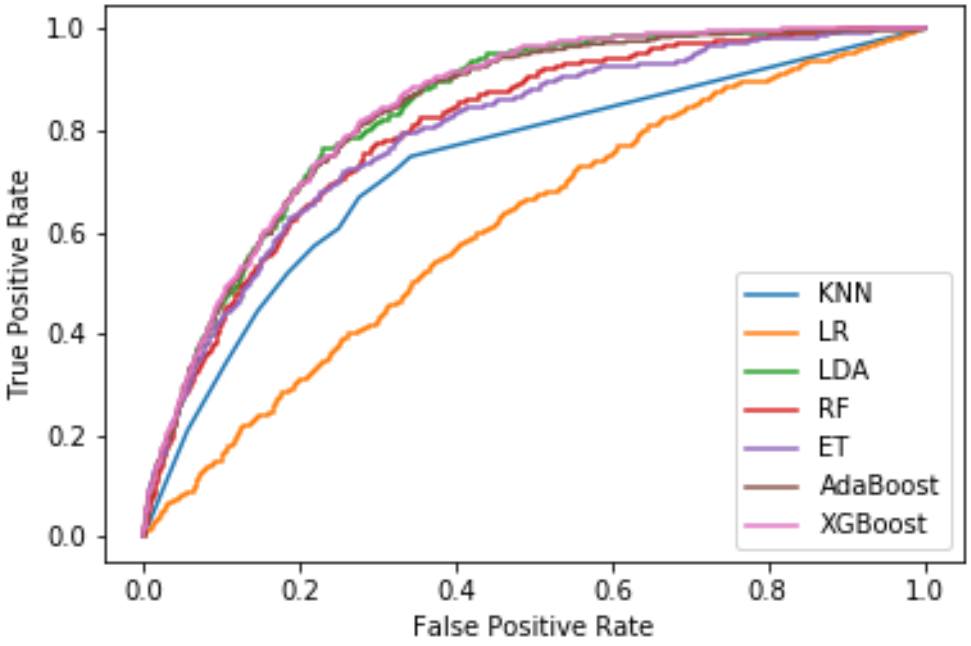


图 7: 各算法预测概率的 ROC 曲线

4 XGBoost 在本问题中的运用

4.1 XGBoost 算法原理

GBDT(Gradient Boosting Decision Trees)是 Jerome Friedman 提出的序列集成算法,它拟

合一系列的弱分类器（通常是决策树），每一个弱分类器都是在前面已经拟合的模型的残差上进行拟合。XGBoost 向 GBDT 的目标函数中加入了关于决策树的叶子结点数目以及各科树权重的惩罚项提升了泛化性能。XGBoost 的每一次迭代解决了这样一个优化问题。

$$\min L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$

XGBoost 对各个参数的更新都是通过梯度实现的，这意味着 XGBoost 本质上只能解决回归问题。但如果我们使用二分类的因变量，则 XGBoost 返回的每个观测的结果可以解为观测属于不同类别的概率，因此 XGBoost 也可以用于解决分类问题。

4.2 选择最优的过采样比例

在前面应用各个算法时，我们选择了使过采样后的训练样本两个类别观测数相等，并将阈值设定为 0.来进行预测。针对不平衡分类问题倾向于将所有观测预测为少数类别的问题，通常有两种解决策略。一是通过操纵阈值使分类器将更多的样本预测为少数分类，而是通过过采样使分类器更关注少数样本。我们在应用 XGBoost 算法时，选择使用不同的过采样比例，仍以 0.5 为阈值并查看各个比例下我们所关心的依赖于混淆矩阵的指标，比如精确率(accuracy)，查准率（precison)，查全率（recall),F-Measure 等。我们以图的形式呈现这些指标。决策者可以根据实际情况决定哪个指标更值得关注并选择相应的最优的过采样比。

在原始训练集中多数类别样本约是少数类别样本的 15 倍，因此我们选择将少数样本分别扩展 1-15 倍来画出上述评价指标的类别图。

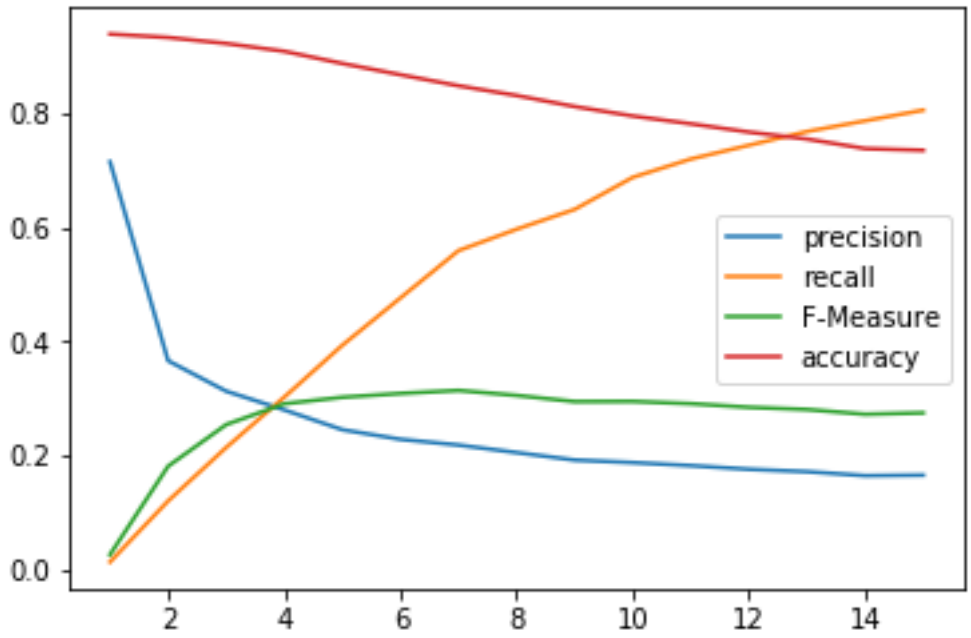


图 8: 不同过采样比例下的指标

4.3 变量聚类重构数据集

在我们构造的数据集中，共有 44 个解释变量，其中 8 个分类变量，并且部分分类变量具有极多的水平数，例如省份这一变量具有多达数十个水平。在进行独热编码后，数据框达到了 121 列。这十分不利于模型的解释性，也不利于业界在实际问题中的操作。我们注意到这些变量之间部分存在复共线性，因此我们希望通过变量聚类的方法减少连续变量并为分类变量构建新的水平。新的水平是原水平的集合。在聚类后，我们将 36 个连续变量聚类为

16 个类，在每个类中选择了解释能力最强的变量，为分类变量构造了新的水平取代了原水平。新的独热编码后的数据框只有 34 个变量。聚类后剩下的连续变量与分类变量构造的新的水平请参考附录 B。我们发现在重构数据集后，识别 AUC 不但没有下降反而略有提升!

	聚类前	聚类后
独热编码后数据框列数	121	34
XGBoost 在验证集上的 AUC	0.841	0.846

表 8: 聚类前后 XGBoost 的表现

4.4 变量重要性解读

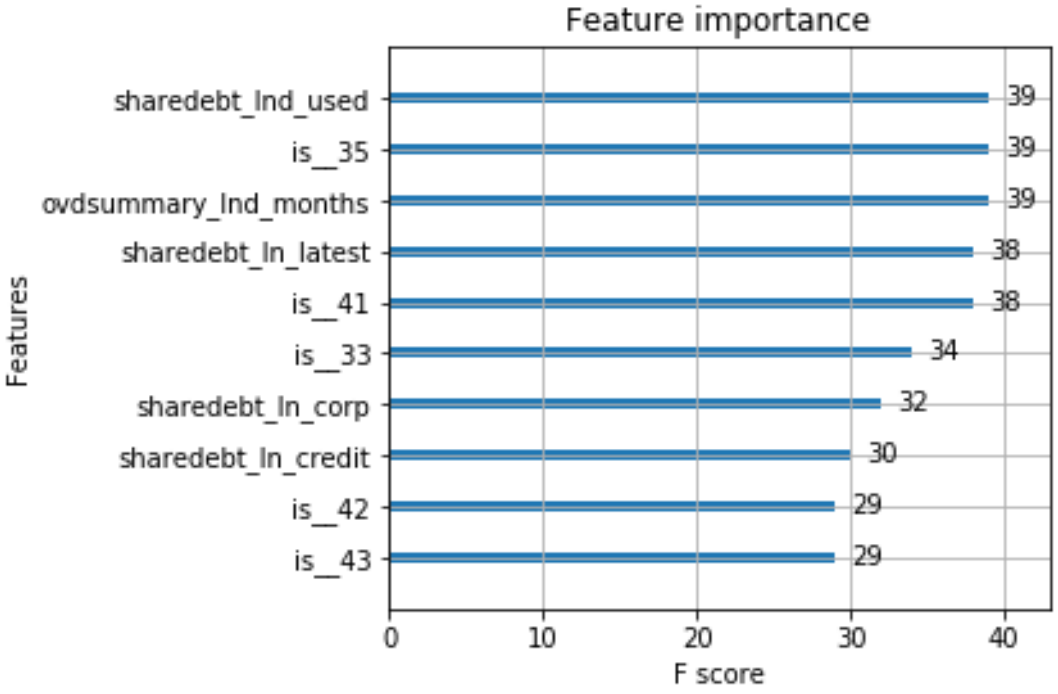


图 9: XGBoost 训练过程中变量重要性

最重要的变量是薪水，说明贷款申请人的经济状况对于判断违约风险是至关重要的。另外地域与学历分类变量中的哑变量在前 15 最重要的的变量中频繁出现，意味着这两个因素是关于申请人背景的最重要的两个变量。关于贷款申请人信用记录的最重要的变量是贷记卡的信用额度，这一现象也是可以直观理解的。获得贷记卡较高的信用额度本就说明了客户良好的信用状况。除此以外住房贷款数量以及信用卡数量则成为了高违约风险的典型特征。

5 总结

大赛的选题而给出了一个因变量分布极度不平衡的二分类问题。经过清洗与整理我们得到了一个 30000*46 的训练集。我们 8:2 的比例进行训练集验证集拆分以衡量相关算法的表现。传统数据挖掘算法会倾向于将所有的观测预测为多数类别。在这个问题的情境下，即我们希望识别潜在违约的客户，这一问题亟待解决。为了使数据挖掘算法将部观测预测为多数类别，我们通常有两种思路，一是通过操纵所预测概率的阈值（cutoff），而是通过过采样改变训练样本的类别分布。我们发现过采样相比阈值操纵还具有能使模 型对少数类别更加敏感，产生更高 AUC 的特点，因此选择使用过采样解决分布不平衡的问题。经典的过采样算法主要有两种，一是简单随机过采样，而是引入人造观测的 SMOTE 算法。我们后续使用的 XGBoost 算法属于集成模型，能够极大程度地避免过拟合，而简单随机过采样相比 SMOTE 算法不会引入噪音，因此我们选择简单随机过采样。通过在数据集上拟合了邻近算法，逻辑

斯蒂克回归，线性判别分析，随机森林，极端随机森林，自适应提升，极端梯度提升，并相互比较后，我们发现极端梯度提升展现了优良的识别性能（(AUC) 与运行速度。为了减少变量的个数并提升模型的解释性，我们尝试了变量聚类，将连续变量减少了一半并大幅减少了分类变量的水平数。我们发现 变量聚类并未缩减模型的识别能力，模型的 AUC 依旧十分出色，达到 0.846。 在模型的解释性方面，XGBoost 的拟合过程中每个变量的累计贡献显示了薪水、地域、 学历这三个反映申请人背景的变量对违约的现象最具解释能力。其余相较其他变量更加重要的有贷款渠道，信用卡信用额度，信用卡数目，房屋贷款数目等。通过在不同过采样比例下 XGBoost 的拟合，我们发现查全率的提升总是以查准率与准确率的下降为代价。即对风险越厌恶，就能更多的排除存在违约风险的贷款申请人，但同时也可能拒绝一些并不会违约的客户。通过对过采样比例的调整，银行从业人员可以自行进行查准率与查全率的取舍。

A 选取变量的解释

变量名	属性	解释
report_ID		信用报告编号
work_province	类别	工作省份
is_local	类别	是否本地
edu_level	类别	学历
marry_status	类别	婚姻状况
has_fund	类别	有无公积金
agent	类别	从何种渠道贷款
query_reason	类别	查询原因
query_org	类别	查询机构
salary	数值	薪水
house_loan_count	数值	个人住房贷款笔数
commercial_loan_count	数值	商业贷款笔数
other_loan_count	数值	其他贷款笔数
loancard_account	数值	贷记卡账户数
standard_loancard_account	数值	准贷记卡账户数
sharedebt_ln_account	数值	贷款账户数 (贷款)
sharedebt_lnd_account	数值	贷款账户数 (贷记卡)
sharedebt_slnd_account	数值	贷款账户数 (准贷记卡)
sharedebt_ln_credit	数值	总合同金额（贷款）
sharedebt_lnd_credit	数值	总合同金额（贷记卡）
sharedebt_slnd_credit	数值	总合同金额（准贷记卡）
sharedebt_lnd_max	数值	最大合同金额（贷记卡）
sharedebt_slnd_max	数值	最大合同金额（准贷记卡）
sharedebt_lnd_min	数值	最小合同金额（贷记卡）
sharedebt_slnd_min	数值	最小合同金额（准贷记卡）
sharedebt_lnd_used	数值	前六个月平均使用额度（贷记卡）
sharedebt_slnd_used	数值	前六个月平均使用额度（准贷记卡）
sharedebt_ln_balance	数值	总贷款余额（贷款）
ovdsummary_ln_count_dw	数值	逾期笔数（贷款）
ovdsummary_lnd_count_dw	数值	逾期笔数（贷记卡）
ovdsummary_slnd_count_dw	数值	逾期笔数（准贷记卡）

ovdsummary_ln_months	数值	累计逾期月数（贷款）
ovdsummary_lnd_months	数值	累计逾期月数（贷记卡）
ovdsummary_slnd_months	数值	累计逾期月数（准贷记卡）
ovdsummary_ln_highest	数值	单月最高逾期总额（贷款）
ovdsummary_lnd_highest	数值	单月最高逾期总额（贷记卡）
ovdsummary_slnd_highest	数值	单月最高逾期总额（准贷记卡）
ovdsummary_ln_max	数值	最大贷款时长
sharedebt_ln_corp	数值	总法人数（贷款）
sharedebt_lnd_corp	数值	总法人数（贷记卡）
sharedebt_slnd_corp	数值	总法人数（准贷记卡）
sharedebt_ln_org	数值	总机构数（贷款）
sharedebt_lnd_org	数值	总机构数（贷记卡）
sharedebt_slnd_org	数值	总法人数（准贷记卡）
Y	分类	是否存在违约情况

B 变量聚类后所选择的连续变量

loancard_count	贷记卡数量
sharedebt_slnd_credit	准贷记卡信用额度
sharedebt_ln_corp	贷款总法人数
sharedebt_slnd_account	准贷记卡数目
house_loan_count	房屋贷款数
ovdsummary_ln_count_dw	贷款累计逾期次数
sharedebt_lnd_credit	贷记卡信用额度
ovdsummary_lnd_count_dw	贷记卡累计逾期次数
ovdsummary_slnd_highest	单月最高逾期总额（准贷记卡）
ovdsummary_slnd_months	累计逾期月数（准贷记卡）
salary	月薪水
ovdsummary_lnd_months	累计逾期月数（贷记卡）
other_loan_count	其他贷款数目
sharedebt_slnd_used	前六个月平均使用额度(准贷记卡)
sharedebt_lnd_min	最小合同金额（贷记卡）

参考文献

[1] L. C. Thomas, “A survey of credit and behavioral scoring: forecasting financial risk of lending to consumers”, *NBER*, 1941

[2] D. Durhand, “Application of the method of discriminant functions to the good and bad loan samples”, *International Journal of Forecasting*, 2000.

[3] J. C. Wiginton, ”A note on the comparison of logit and discriminant models of consumer credit behavior”, *Journal of Financial and Quantitative Analysis*, 1980. \

[4] S. Chatterjee and S. Barcun, ”A nonparametric analysis of empirical versus judgmental credit evaluation”, *The Journal of Retail Banking*, 1970.

[5] S. Lessmann, ”Benchmarking state-of-art classification algorithms for credit scoring: an update of research”, European Journal of Operations Research, 2015

[6] T. Q. Chen and C. Guestrin, ”XGBoost: a scalable tree boosting system”, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016

[7] A. Fernandez, S. Garcia, and F. Herrera, “Addressing the classification with imbalanced data: Open problems and new challenges on class distribution”, *HAIS*, 2011.

[8] N. V. Chawla, K. W. Bowyer, and L. O. Hall, “SMOTE: synthetic minority over-sampling technique”, *Journal of Artificial Intelligence Research*, 2002

[9] Intel, “How to handle imbalanced classification problems in machine learning?”, https:// www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/, Mar 2017.

[10] Y. Tang, Y. Zhang, and Chawla, “SVM modeling for highly imbalanced classification”, *IEEE*, 2009