信用卡评分建模分析报告

刘华秋, 顾逸鸥, 罗文

June 3, 2021

1 问题定义

1.1 已知

1、15000 条拥有如下属性的用户数据

属性名	描述	数据类型
SeriousDlqin2yrs (label)	是否违约（两年内出现逾期超过 90 天的情况）	**Y/N**
RevolvingUtilizationOfUnsecuredLines	信用卡总余额和个人信用额度除以总信用限制。	percentage
age	年龄	integer
NumberOfTime30-59DaysPastDueNotWorse	30-59 天欠款逾期次数	integer
DebtRatio	月债务支出、赡养费、生活费除以总收入（毛收入）	percentage
MonthlyIncome	月收入	real
NumberOfOpenCreditLinesAndLoans	公开贷款（如汽车和抵押分期）和在线信用（如信用卡）数量	integer
NumberOfTimes90DaysLate	90 天或以上贷款逾期未还的次数。	integer
NumberRealEstateLoansOrLines	抵押和房地产数量（包括房屋净值信用额度）	integer
NumberOfTime60-89DaysPastDueNotWorse	60-89 天欠款逾期次数	integer
NumberOfDependents	家庭成员数目（比如说配偶子女，但不包括他自己）	integer

2、101503 条缺失“SeriousDlqin2yrs”属性的用户

1.2 目标

1、分析并量化不同特征对用户信用评分的重要程度
2、不同特征之间的相关关系
3、建立相关模型对于缺失“是否违约”属性的用户进行信用评分

2 数据集初步分析及预处理

首先使用 pandas 中的 describe() 函数对于 cs-training 与 cs-test 中的数据集进行粗略的分析。由于涉及到的属性过多，篇幅无法一次性容下，因此只展示需要进行分析处理的数据。

2.1 数据重复

使用 $data.duplicated().value\_counts()$ 函数发现两数据集并没有重复元素。

2.2 数据缺失

2.2.1 MonthlyIncome

	Training Data MonthlyIncome(共 150000)	Test Data MonthlyIncome(共 101503)
count	120269.0	81400.0

可以看出，在 traindata 与 testdata 中，MonthlyIncome 都有着不同程度的缺失。由于这二者数据都缺失了 1/5，因此希望通过随机森林算法对其进行填充。然而，test 数据集缺失了 SeriousDlqin2yrs 这一项，导致其 MonthlyIncome 预测出的结果并不很好，因此最后使用了均值填充。下面我们对 train 的 MonthlyIncome 进行随机森林回归：

我们发现，在初始设置传入的参数时，如果将'DebtRatio' 这一列数据加入，则该回归的误差（OOB）相当小 (小于 7%)，然而当我们去掉它时，OOB 回到了一个较为正常、符合我们预期的数值 (17.9%) 我们认为，这其中可能是由于 DebtRatio 与 MonthyIncome 有强烈的相关关系有关，在 DebtRatio 的计算公式中就有 $DebtRatio = \frac{Spendings}{MonthlyIncome}$。这导致了在我们的预测模型中 DebtRatio 占比过大使其它因素的影响几乎无法表现。因此我们为了不让 MonthlyIncome 变成第二个 DebtRatio 而将 DebtRatio 去掉，使补进去的数据成为一个有价值的、与其它多个因素相关联的数据。

2.2.2 NumberOfDependents

	Training Data NumberOfDependents(共 150000)	Test Data NumberOfDependents(共 101503)
count	146076.0	98877.0

可以看出，在 traindata 与 testdata 中，NumberOfDependents 都有着不同程度的缺失。而 traindata 的缺失极少，因此直接将缺失的用户数据舍弃。出于与 MonthlyIncome 同样的原因，使用均值对其进行填充。

2.3 数据的异常值处理

对所有数据做直方图 1并逐个考虑：

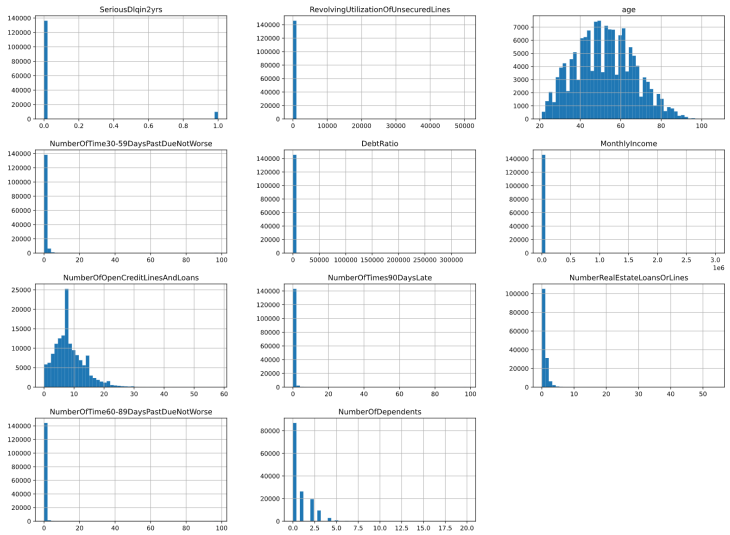

Figure 1: 用于数据清洗的数据直方图

2.3.1 RevolvingUtilizationOfUnsecuredLines 以及 DebtRatio

这两个属性单位为百分比，因此不应大于一，可将所有大于一的数据清除。

2.3.2 Age

该属性不能为 0，且一般不大于 150，因此我们只取 (0,150) 这一区间，其它的清除。

2.3.3 NumberOfTime...

NumberOfTime30-59DaysPastDueNotWorse, NumberOfTimes90DaysLate, NumberOfTime60-89DaysPastDueNotWorse 这三项在直方图中不太清楚，画为箱线图 2后能够明显的看到它们的异常值均在 80 以上，因此我们只保留用户三项属性的值小于等于 80 的部分。

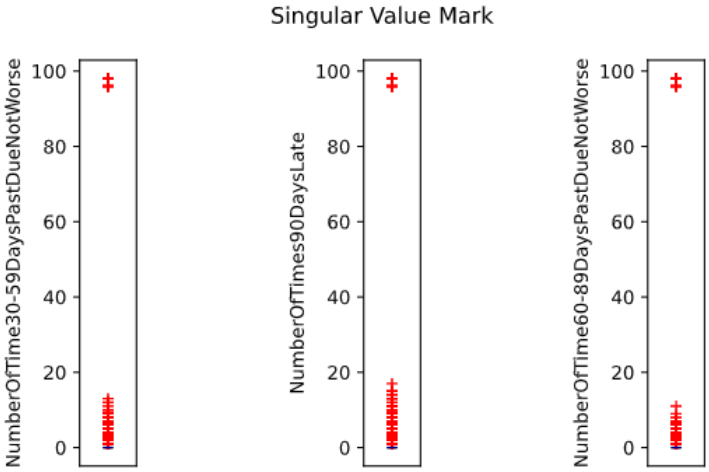

Figure 2: NumberOfTime... 箱线图

2.3.4 NumberRealEstateLoansOrLines,NumberOfDependents

通过从大到小试探出异常值的下确界的方式 (NumberRealEstateLoansOrLines:>50 ; NumberOfDependents:>15) 将其清除

2.3.5 MonthlyIncome

剔除月薪大于 50000 以上的异常值。

2.4 数据集分析

2.4.1 部分属性的分布特征

下面 3是经过数据清洗后的数据直方图

从 4可以发现 age, MonthlyIncome, DebtRatio, NumberOfOpenCreditLinesAndLoans 四个属性的的分布情况近似于正态分布，符合一般的统计假设。

如果只关注需要评估的 SeriousDlqin2yrs 的图像 5, 会发现 data 中违约率的分布是相当不均衡, 属于 imbalanced classification 问题

2.4.2 属性间的联系与特征

通过计算属性间的相关性系数矩阵 R，我们得到了它的相关性系数热力图 6

其相关系数矩阵为：

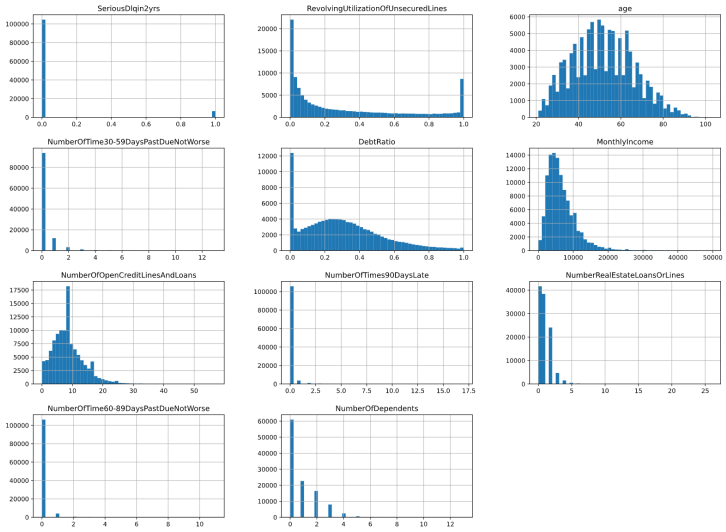

Figure 3: 清洗后的数据直方图

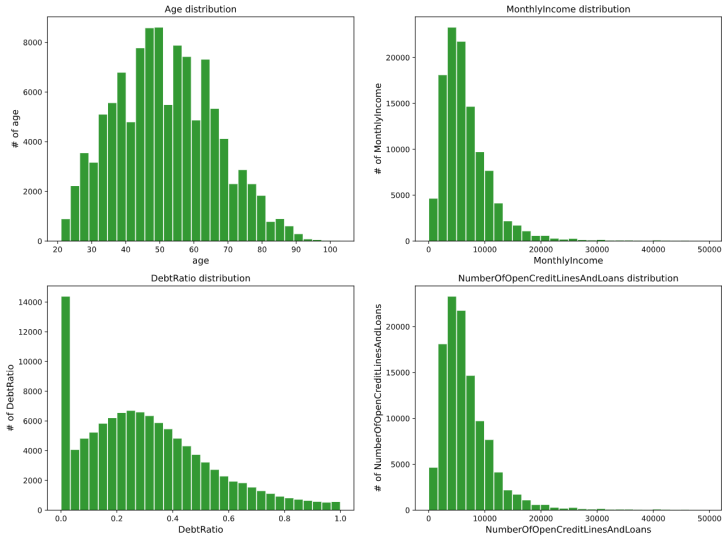

Figure 4: 部分属性的分布特征

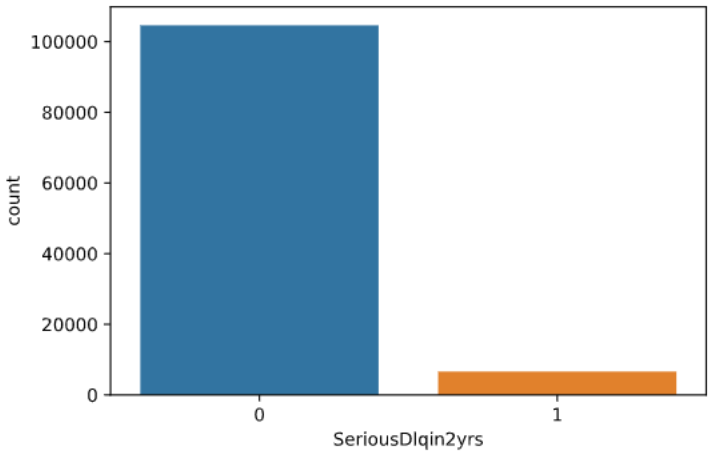

Figure 5: SeriousDlqin2yrs Plot

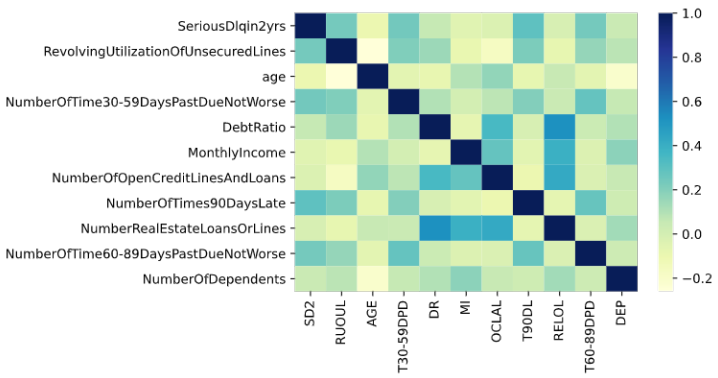

Figure 6: Correspondence Plot

	SD2	RUOUL	AGE	T30-59DPD	DR	MI	OCLAL	T90DL	RELOL	T60-89DPD	DEP
SD2	1.0	0.2375	-0.096	0.2430	0.058	-0.04	-0.026	0.2961	-0.013	0.2379	0.0440
RUOUL	0.2375	1.0	-0.259	0.2093	0.149	-0.08	-0.166	0.2195	-0.075	0.1654	0.0815
AGE	-0.096	-0.259	1.0	-0.058	-0.07	0.095	0.1715	-0.075	0.0535	-0.059	-0.209
T30-59DPD	0.2430	0.2093	-0.058	1.0	0.102	0.006	0.0779	0.2031	0.0405	0.2812	0.0579
DR	0.0584	0.1490	-0.079	0.1027	1.0	-0.07	0.3540	-0.012	0.5228	0.0374	0.1034
MI	-0.046	-0.085	0.0959	0.0069	-0.07	1.0	0.2793	-0.057	0.3962	-0.029	0.1861
OCLAL	-0.026	-0.166	0.1715	0.0779	0.354	0.279	1.0	-0.095	0.4248	-0.022	0.0506
T90DL	0.2961	0.2195	-0.075	0.2031	-0.01	-0.05	-0.095	1.0	-0.066	0.2716	0.0284
RELOL	-0.013	-0.075	0.0535	0.0405	0.522	0.396	0.4248	-0.066	1.0	-0.022	0.1384
T60-89DPD	0.2379	0.1654	-0.059	0.2812	0.037	-0.02	-0.022	0.2716	-0.022	1.0	0.0332
DEP	0.0440	0.0815	-0.209	0.0579	0.103	0.186	0.0506	0.0284	0.1384	0.0332	1.0

从系数矩阵中我们可以看出，和 SeriousDlqin2yrs 相关性较强的变量为 RevolvingUtilizationOfUnsecuredLines、NumberOfTime30-59DaysPastDueNotWorse、NumberOfTimes90DaysLate、NumberOfTime60-89DaysPastDueNotWorse

2.5 具体实现

2.5.1 准备工作

1、由于原始数据中违约率的分布是十分不均衡的，对于这样的不平衡分类问题，我们引入 imlearn 库，使用降采样 RandomUnderSampler 的手段使二者达到 1:1 平衡状态。

2、使用 *train_test_split* 函数对降采样后的数据集进行训练集与测试集的划分。

2.5.2 训练拟合

我们使用梯度下降树 (GBDT) 的方式进行训练拟合，并使用 GridSearchCV 进行参数上的优化调整。具体代码如下：

```
param_search_1={'n_estimators':range(1,1002,100)}
sample=GridSearchCV(estimator=GradientBoostingClassifier(learning_rate=0.1,random_state=10),
    param_grid=param_search_1,scoring='roc_auc',n_jobs=-1,return_train_score=True,refit=True)
sample.fit(X_resampled,y_resampled)
sample.cv_results_,sample.best_params_,sample.best_score_
```

该段代码得到：

```
  ......(前略)
{'n_estimators': 101},0.8401951053177882
```

前面的为 GBDT 中的参数在 GridSearchCV 中获得最大‘roc auc’值，也即后面的数字的参数。该参数调整时范围较大，缩小范围继续调整

```
param_search_1={'n_estimators':range(1,202,20)}
sample=GridSearchCV(estimator=GradientBoostingClassifier(learning_rate=0.1,random_state=10),
    param_grid=param_search_1,scoring='roc_auc',n_jobs=-1,return_train_score=True,refit=True)
sample.fit(X_resampled,y_resampled)
sample.cv_results_,sample.best_params_,sample.best_score_
```

得到

```
  ......(前略)
{'n_estimators': 121},0.840527237620235
```

将该参数带入，继续调整其他参数，得到：

```
  {'max_depth': 5, 'min_samples_split': 401},0.8405995893974124
{'max_features': 5}, 0.8410558371623263
```

最后同步减小学习率提高决策树的数量：

```
param_search_5={'n_estimators':[120,240,480,960,960*2],'learning_rate'
    :[0.1,0.05,0.025,0.0125,0.00625]}
search_5=GridSearchCV(estimator=GradientBoostingClassifier(random_state=10,min_samples_split
    =401,max_depth=5,max_features=3), param_grid = param_search_5, scoring='roc_auc',n_jobs
    =-1)
search_5.fit(X_resampled,y_resampled)
search_5.cv_results_,search_5.best_params_,search_5.best_score_
```

在该参数上进行微调，得到最终的参数为：

$$min\_samples\_split = 400$$

$$max\_depth = 4$$

$$max\_features = 8$$

$$learning\_rate = 0.025$$

$$n\_estimators = 480$$

此时

$$trainAUCScore : 0.8677278645422821$$

$$testAUCScore : 0.8354992323979017$$

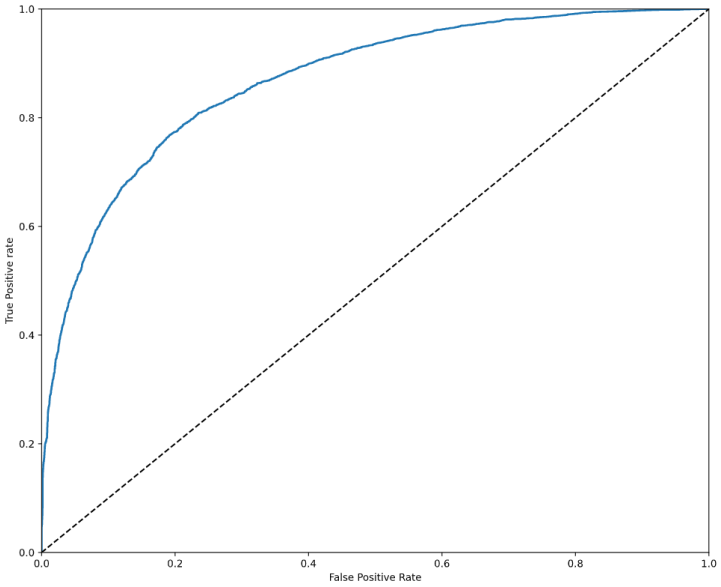

Figure 7: TRAIN TPR-FPR 曲线

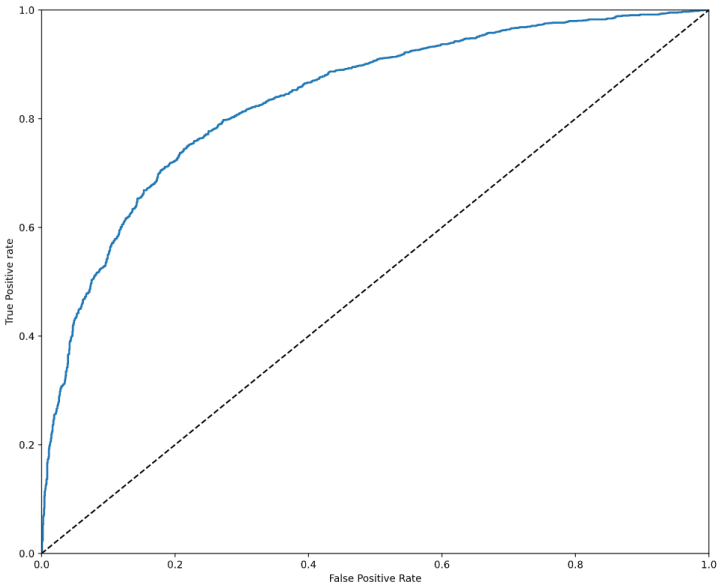

Figure 8: TEST TPR-FPR 曲线

2.6 可视化结果

下面是我们做出的关于信用卡评分模型的可视化结果。

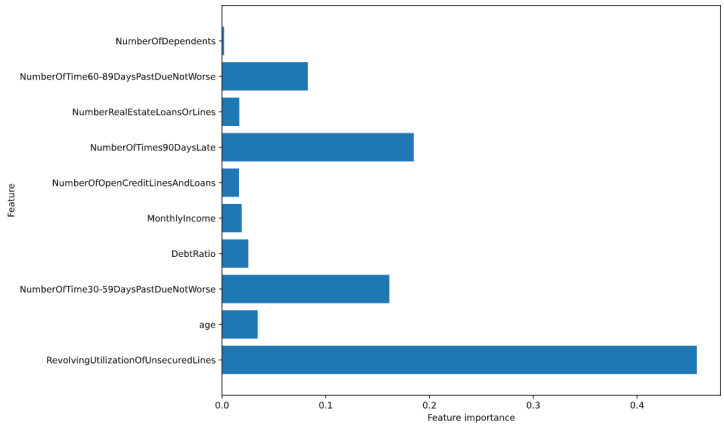

Figure 9: 不同特征对用户信用评分的重要程度

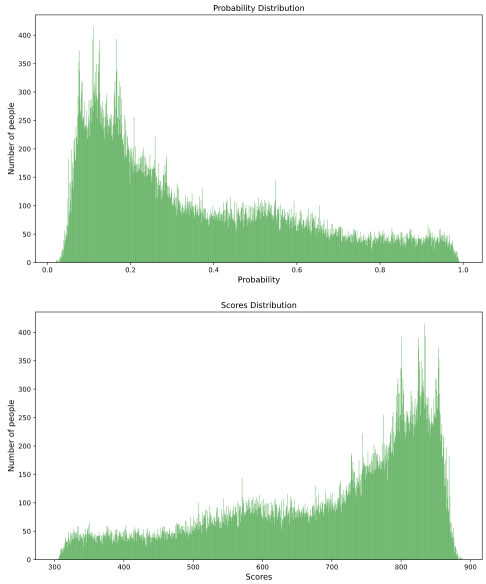

Figure 10: 可能违约的概率密度以及对每个个体的评分

3 进行评估与预测

导入'cs-test.csv'，使用之前训练好参数的函数进行评估预测，并将函数返回的概率以线性的方式折合到个人评分上，形成最终的结论。

4 结论

通过上述的分析与计算，我们发现影响一个人是否违约的主要因素是其延期 30-59 天、60-89 天、90 天以上未偿付的次数，还有他的信用卡总余额和个人信用额度除以总信用限度的多少。反而个人的收入、背负债务的比重似乎在我们的分析下影响力显得并没有那么大。

5 一些问题与讨论

在进行数据处理的过程中，我们小组曾经想把对于 MonthlyIncome 缺省值的补全放在异常值剔除之后，因为这样做能够使随机森林算法在构建决策树时不会受到过多异常值的干扰，确实有道理。然而当我们动手实践之后，我们发现在中间过程完全一样的情况下，将补全缺省值放在最后做将会使训练集上的 AUC 下降 2%，测试集上的 AUC 下降近 3%。在进行了一番分析后我们发现，如果我们将异常值提前剔除，那么随机森林的 OOB 指标最小也要大于-0.225，不然 OOB 只有-0.170。（二者均为 $n\_estimators = 600$ 时达到）因此也许正是由于拥有异常值的数据所贡献的“非异常值”对于模型的正向作用要大于其携带的异常值对于模型的负向作用要更强，导致了这种现象的发生。确实随机森林算法本质上是由许多个弱分类器“投票”组成的强分类器，因此可能对于奇异值、缺省值有较强的鲁棒性。

6 分工

罗文

主要代码架构实现

刘华秋

调参、对比不同方法，完善代码、报告写作、做 PPT 准备展示

顾逸鸥

完善重要代码、进行项目规划、进度监督