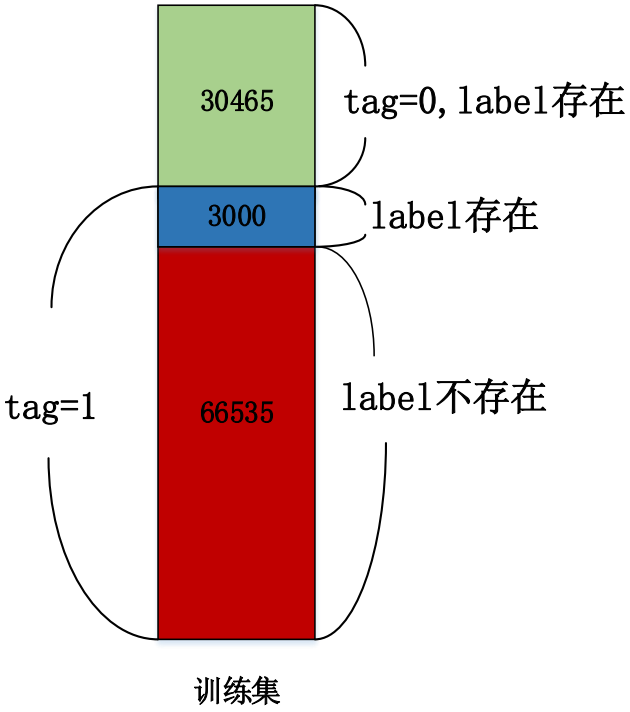
推断拒绝答辩

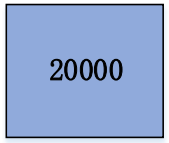
关键字：特征工程、半监督学习、模型融合

队伍：irootech
队长：申将伟
单位：树根互联 AI lab

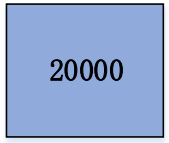
目录

1.数据集介绍





验证集



测试集

数据特点:

1. 数据的特征没有明确含义，不利于做特征的构建有字段含义的特征不多：id、loan_dt、tag（只有训练集有tag特征）

2.维度高，除了有字段含义的特征外，剩余的特征为f1~f6745

2.数据分析和处理

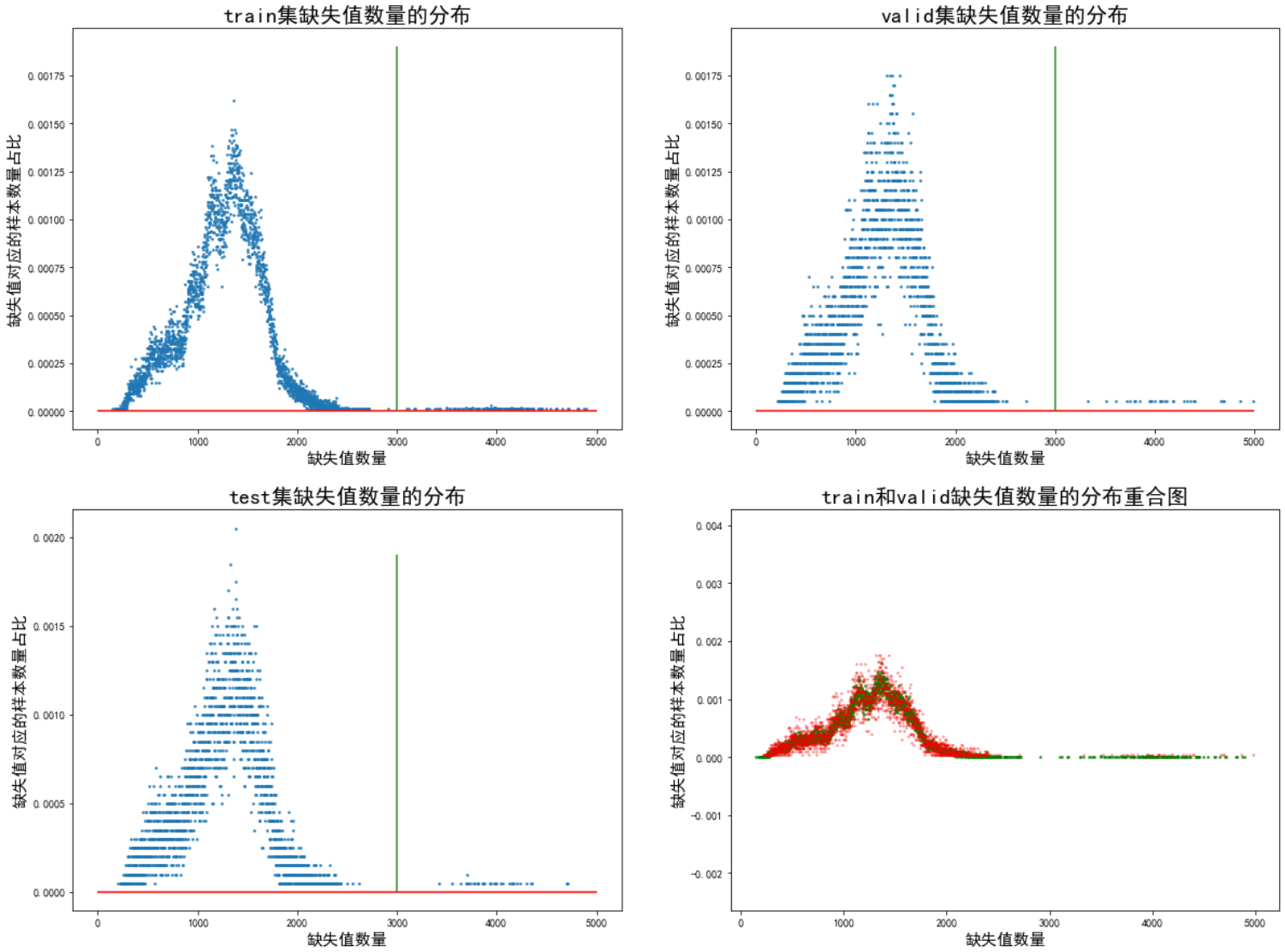
2.1数据探索
2.2特征预处理
2.3特征提取
2.4新特征构建

2.1.1 数据类型

数据类型	int64	float64	string	总计
数量	494	6253	1（loan_dt）	6748

数据集中的数据类型有三种，分别为int64、float64、string，其中int64类型的特征数量为494，float64类型的特征数量为6253，string类型特征为loan_dt，总计6748个特征。

2.1.2 查看样本特征缺失数量的分布



三个数据集的n_null特征分布图

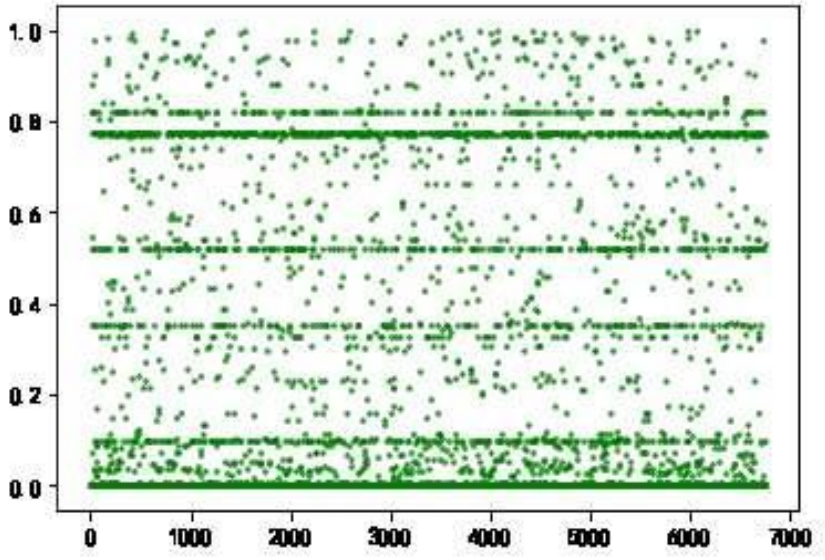
分析:
1.n_null统计每个样本缺失的特征数

2.根据图像，我们猜想三个数据集源于同一个分布。

3.但是通过K-S检验，
发现测试集和验证集的P值高达0.6，
而训练集与验证集的P值远远小于0.05.

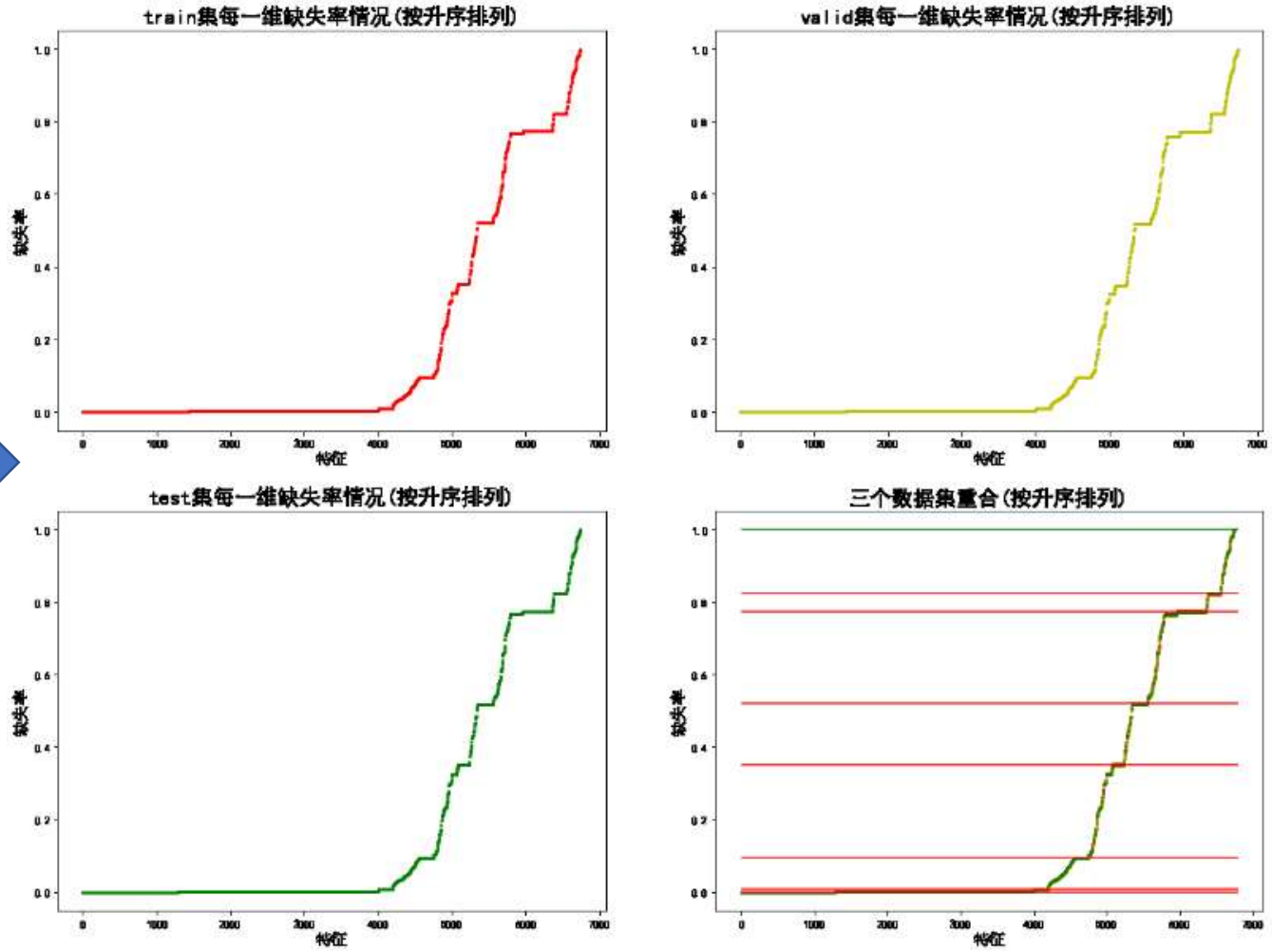
4. 因此我们需要从其他角度来提高我们猜想的可信度。

2.1.3查看特征的缺失率分布



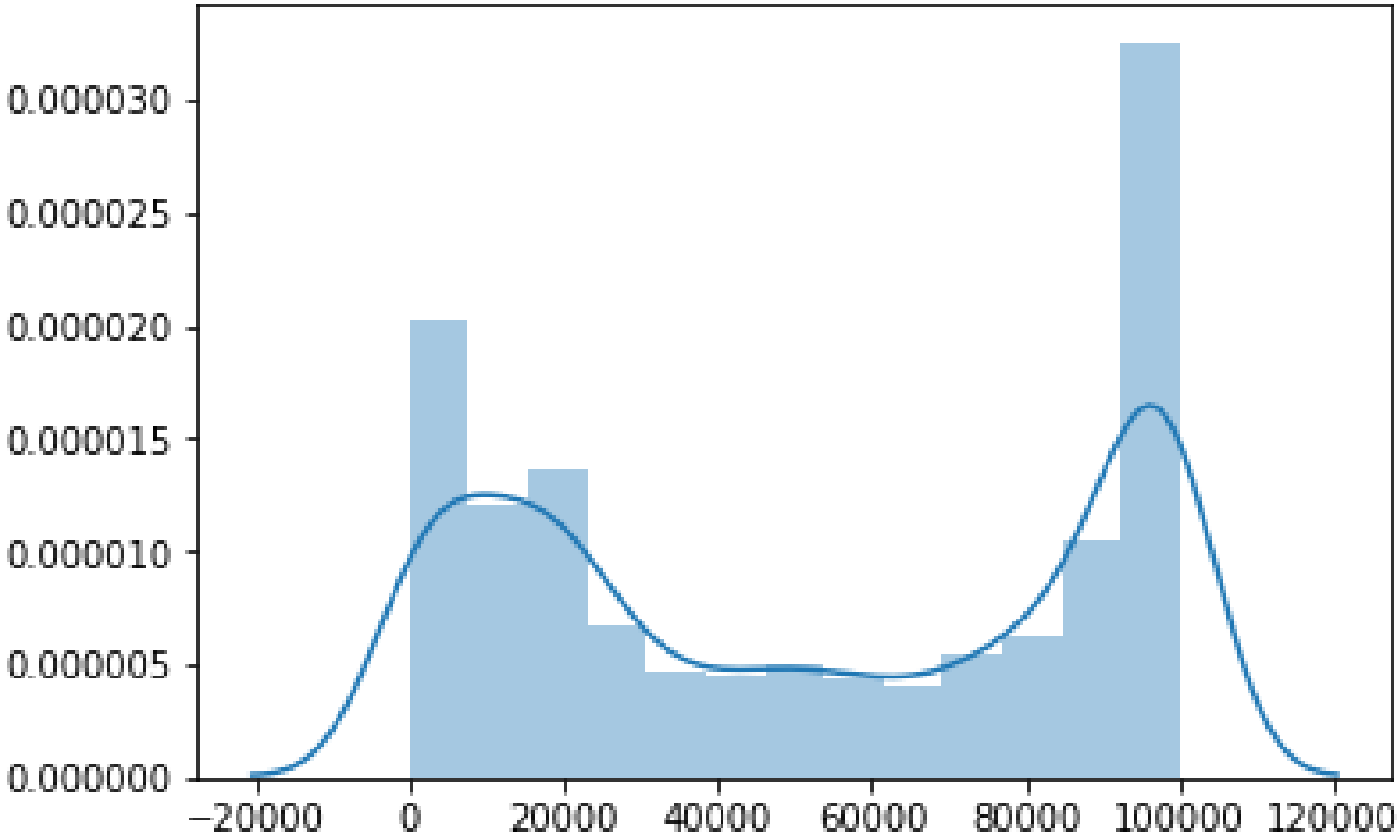
每一维数据缺失率图

按缺失率排序



每一维数据缺失率的分布图（按升序排列）

2.1.4 查看每一维众数的分布



◄众数数量减少—　　—众数数量增加►

1. 提取每一维是否为众数作为指示特征特征maj

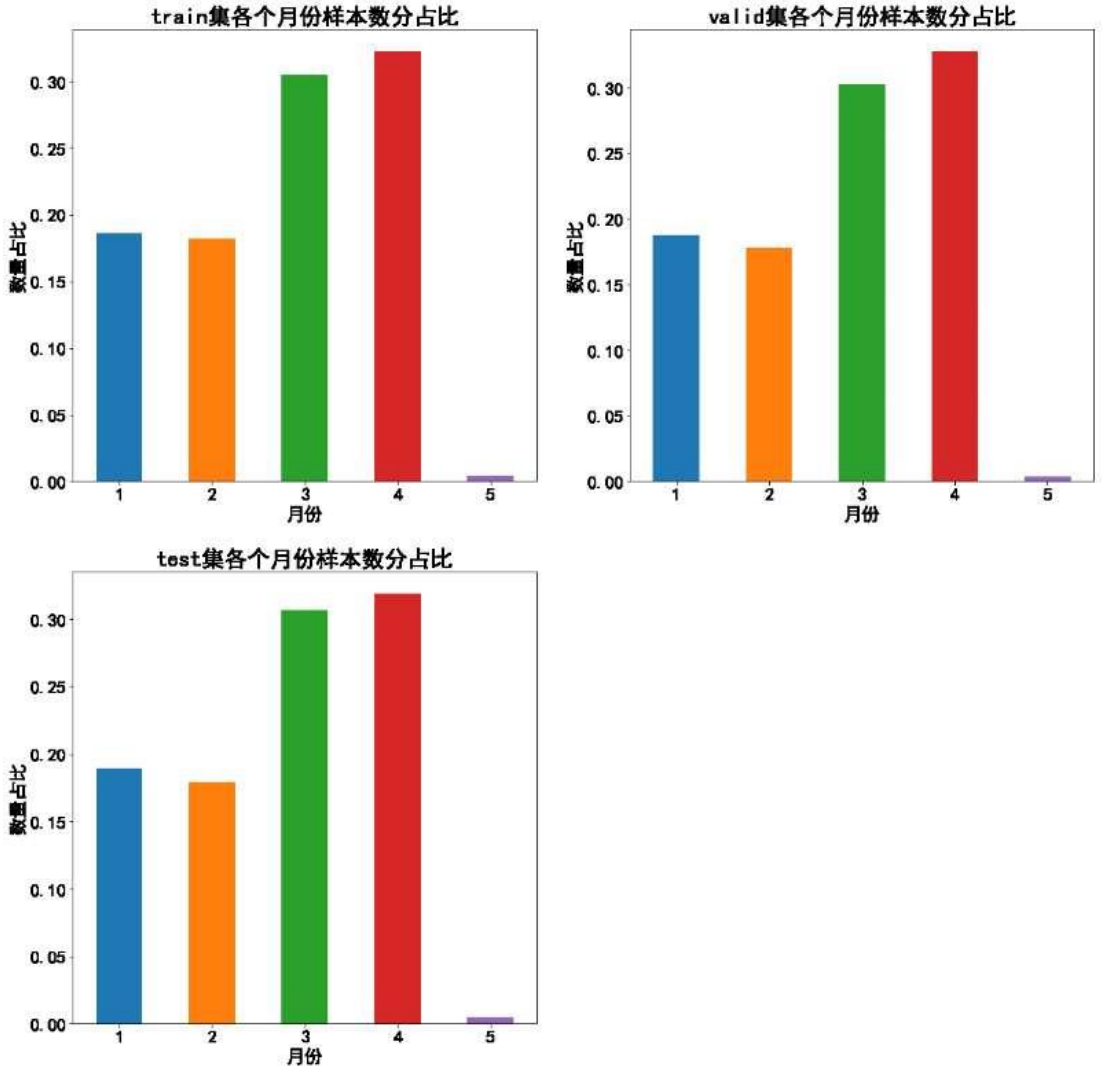
2. 统计所有维maj=1的次数为maj_cnt

2.1.5 深度挖掘时间特征的信息

为什么要针对时间这个特征做文章?

1. loan_dt是为数不多有字段含义的特征之一
2. 对于金融风控来说，数据的时间越新，包含的信息越有用
3. 时间特征可能隐含了一些规律
4. 方便我们从时间维度上去验证训练集与测试集分布的一致性

2.1.6 loan_month

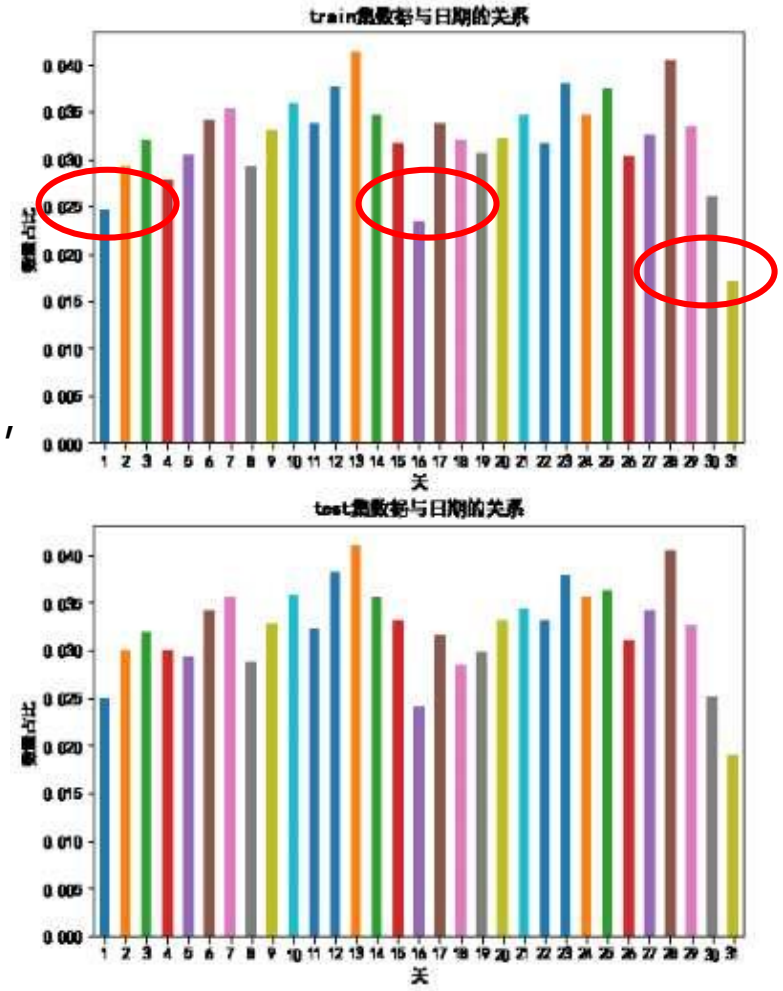


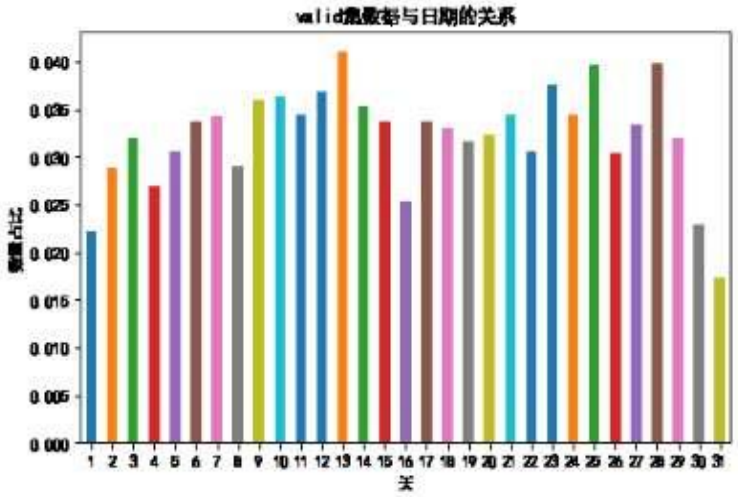
数据集各个月份样本数量占比图

分析：
1.数据都分布在1~5月
2.三个数据集数量占比分布一致
3.loan_month提升学习器区分不同月份数据的能力

2.1.7 loan_date

分析：
1.三个数据集分布是相似的，
我们更加确信两个数据集分布一致。
2. 提取每个月的第几天这个特征为loan_day，
loan_date将进一步细粒化数据
3. 发现规律：
在月初、月中、月末数据量的分布偏少



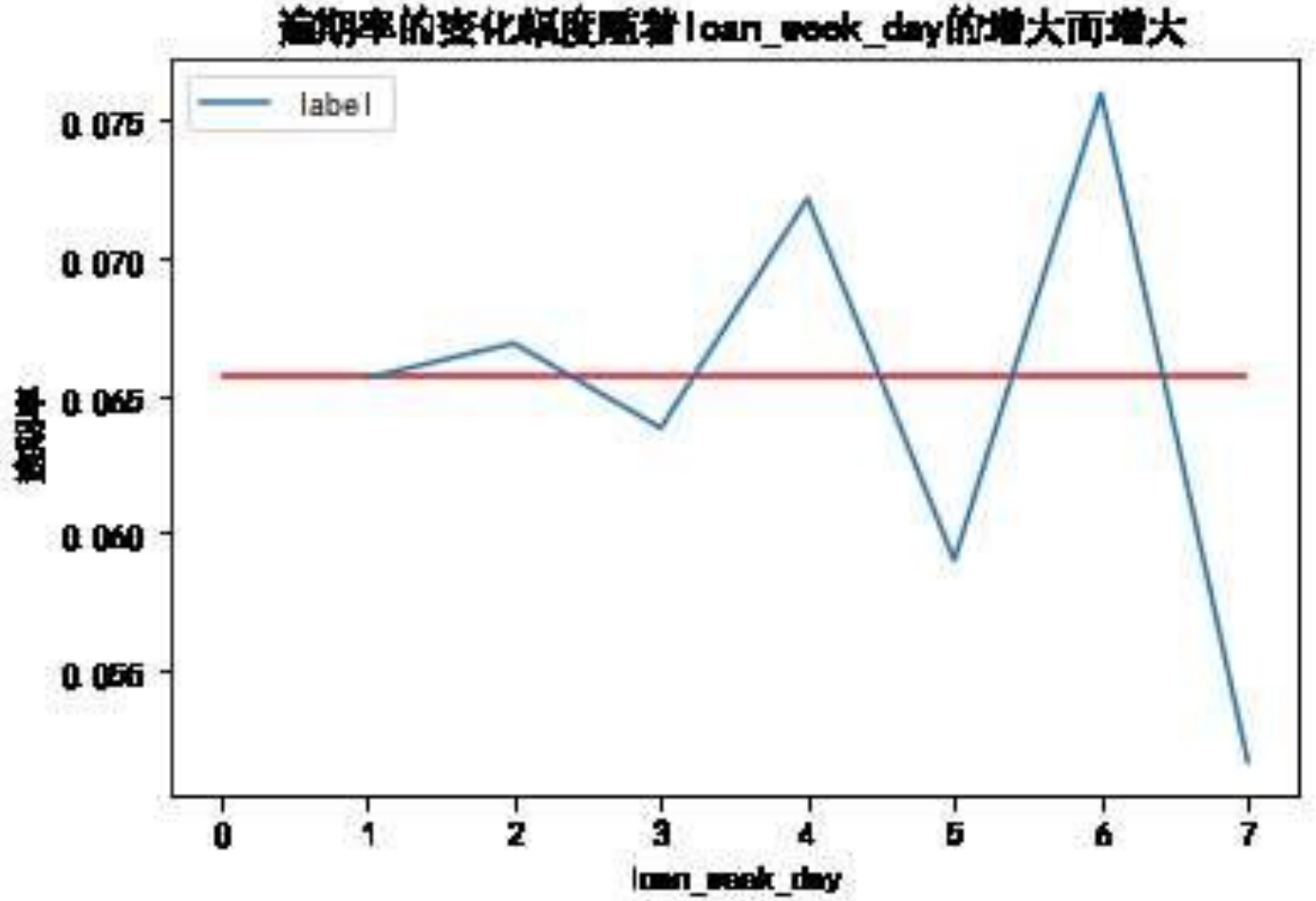


2.1.8 loan_week_day

分析:
1.与前面不同的地方在于:
因为计算的是逾期率，所以只用了有标签数据。

2.逾期率的变化幅度随loan_week_day增大而增大。

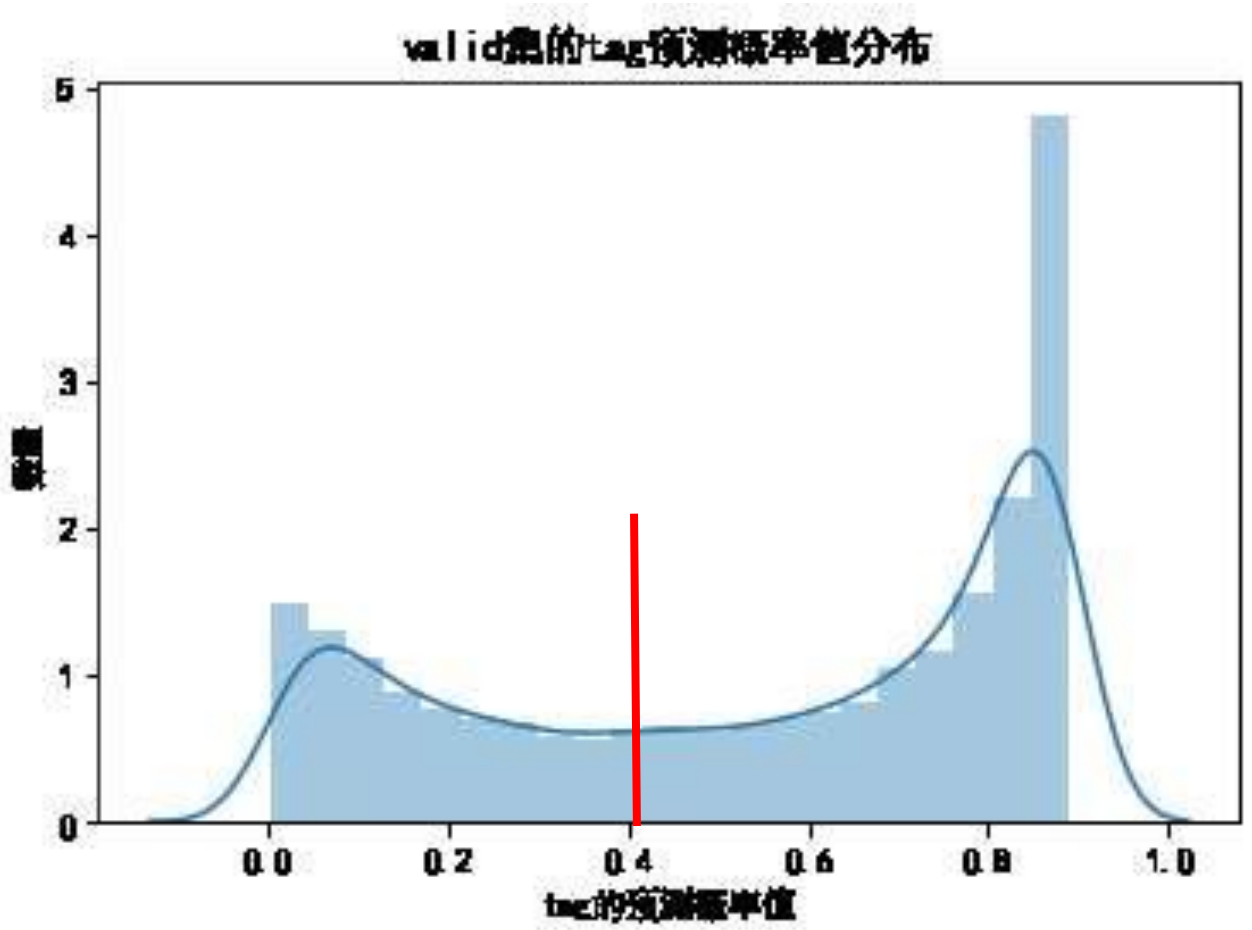
注：红线代表平均逾期率



逾期率随loan_week_day的变化趋势图

2.1.9 tag特征的使用

问题：
只有训练集tag特征，
需要对测试集的tag进行预测，
而测试集上tag分布未知。



tag的预测概率值分布图

2.2 特征预处理

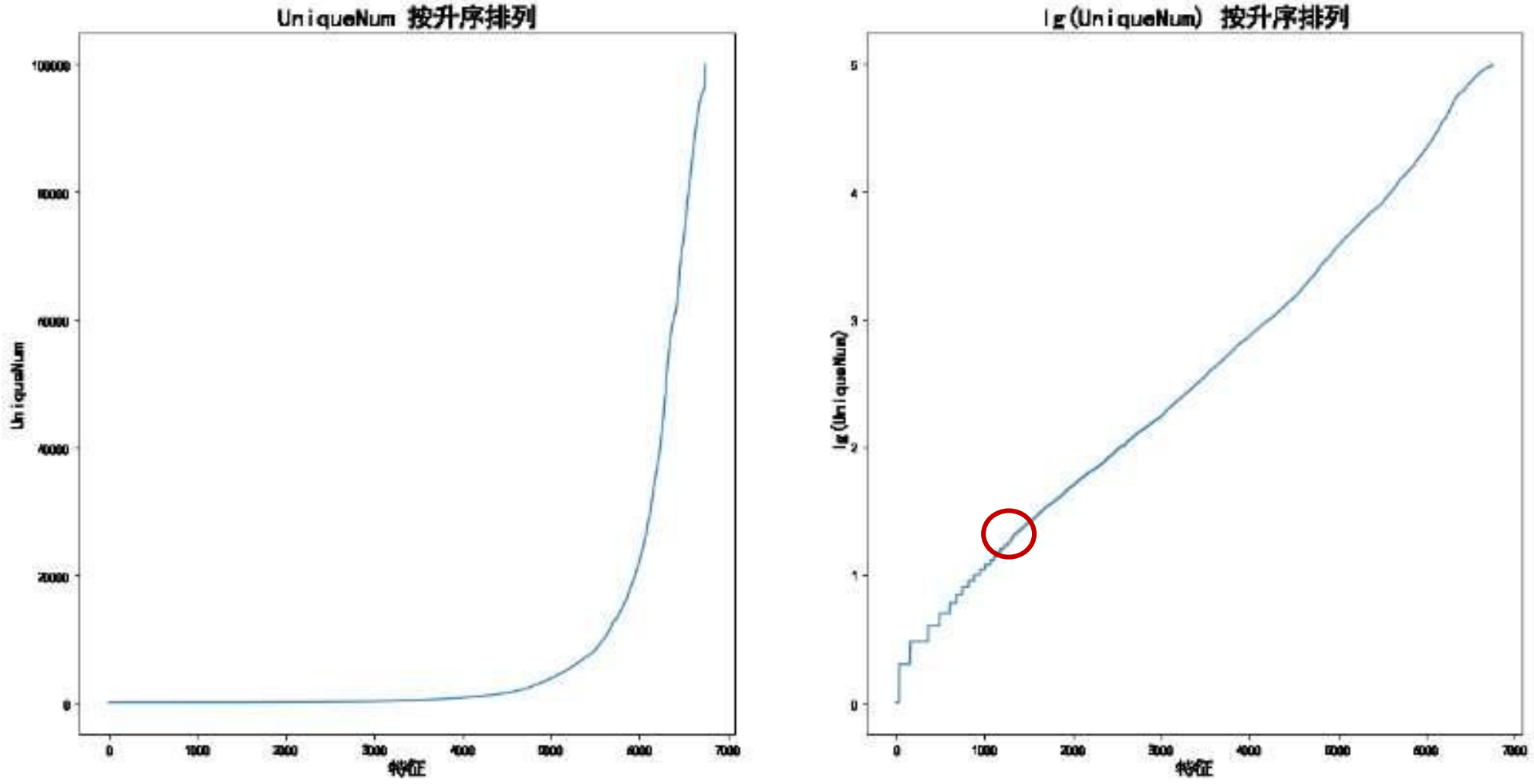
1. 将loan_dt转化为loan_month, loan_date和loan_week_day。

2. 缺失值处理：填-1，因为数据绝大部分都是正数，所以填一个负数，就能很好把缺失值和正常数据给区分开。

3. 将数据划分为离散值和连续值，再分别对离散值做label encoding 和 对连续值做分桶化。

4. 对连续值做分桶化，可以利用IV-WOE来选择区分能力最强的分段区间。

2.2.1 划分离散值和连续值



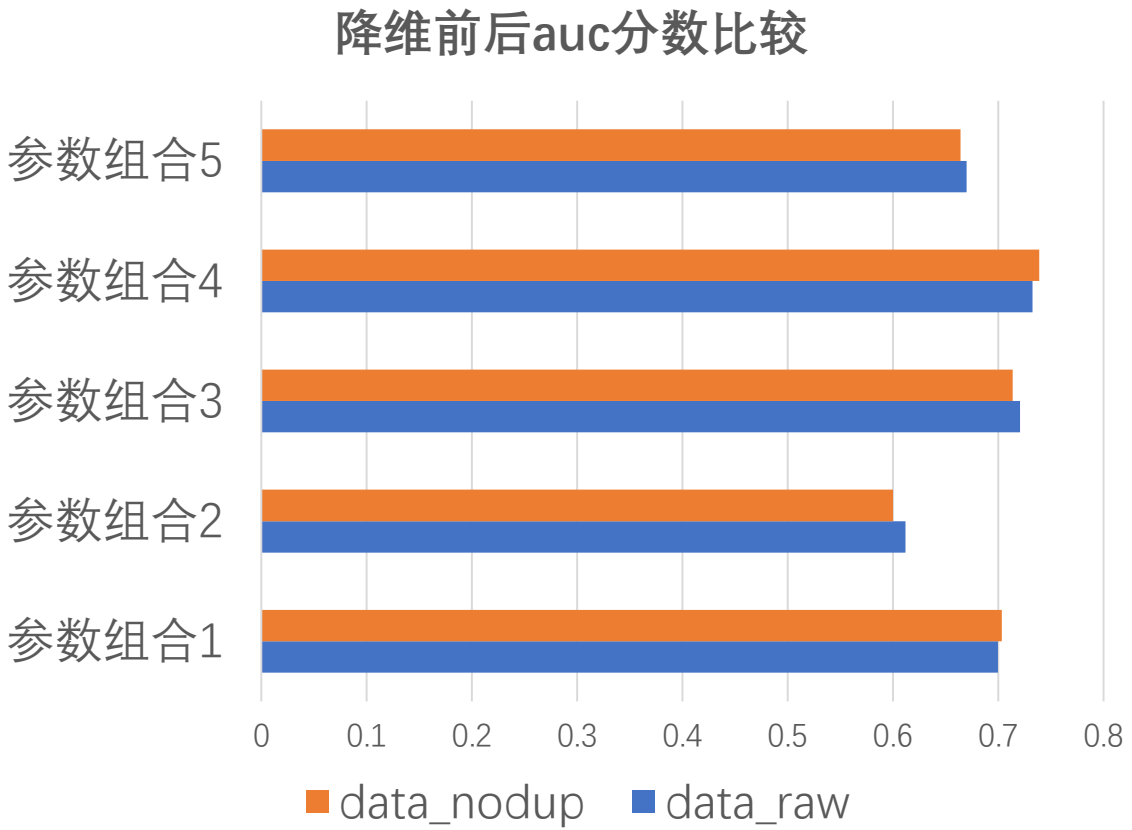
特征的UniqueNum按升序排列

2.3 特征降维

把无字段含义特征中重复率超过95%的特征剔除掉，特征数量从6745维(称为data_raw)下降到4807维，并称之为data_nodup。
接下来我们使用lightgbm学习器，设置五组不同的参数，来对比降维前后auc分数的表现。

本地验证集auc对比	参数组合1	参数组合2	参数组合3	参数组合4	参数组合5
data_raw	0.6998	0.6119	0.7207	0.7325	0.6699
data_nodup	0.7035	0.6002	0.7137	0.7391	0.6643

注：通过设置多组参数，我们保证了对比的客观性。
降维前后auc分数基本相同



2.4 新特征构建

1.指示特征

null_sign : 给每一个特征生成一个是否为缺失值的指示变量。

maj：给每一个特征生成一个是否为众数的指示变量。

2.统计特征

n_null ： 统计每个样本null_sign为1的次数

majcnt ： 统计maj为1的次数

3.时间特征

loan_month、loan_day、loan_week_day

4.特征交叉

比如f1和f2相乘，得到新特征f1*f2

5.tag特征

利用“历史学习器”的信息

	f1	f2	f3
f1	f1*f1	f1*f2	f1*f3
f2		f2*f2	f3*f3
f3			f3*f3

例子：3个特征做乘法特征交叉

特征汇总

指示特征	描述
null_sign	特征是否缺失
maj	特征是否为众数

时间特征	描述
loan_month	月份
loan_day	一个月内第几天
loan_week_day	一周内第几天

统计特征	描述
n_null	一个样本缺失的特征总数
nd_null	n_null离散化
maj_cnt	一个样本属于众数的特征总数
maj_cntd	maj_cnt离散化

降维特征	描述
data_nodup	4807维无字段含义特征

交叉特征	描述
cross feature	前100维特征两两相乘

tag特征	描述
tag	指示样本是否通过

3.本道题的难点

1.由于只有部分数据有标签，在本地做交叉验证时，
生成的验证集合线上的验证集分布不一致。

2. 数据字段含义特征太少，不利于特征工程构建。

3.想要利用训练集中有60000多条unlabel数据，
需要使用半监督学习方法。

4.预测验证集和测试集上的tag特征。

4. 模型构建

4.1 模型选择
4.2 主要基础模型（用于模型融合）
4.3 模型融合

4.1.1 线性模型的表现巨差

1. 尝试逻辑回归和SVM，其中线性核的SVM的线上AUC成绩不超过0.6。

2. 我认为原因有两点：

一是“喂”给线性学习的数据存在离散值，没有对离散值做one-hot编码；

二是 对非线性数据，没法直接用线性分类器来划分，除非对特征做一些非线性的转化，比如特征做平方。

4.1.2 树相关的模型表现最好

学习器	Random Forest	GDBT	XGBoost	LightGBM
参数	n_estimators: 200, max_depth: -1, n_jobs: 8	n_estimators: 200, max_depth: 3, learning_rate: 0.1	num_boost_round: 200, max_depth: 5, learning_rate: 0.05, n_jobs: 8	num_boost_round: 200, max_depth: 5, learning_rate: 0.05, max_bin: 200, n_jobs: 8
训练速度（训练33465条有标签数据，使用所有原始特征+少量构建的特征）	20.64 s	1509.91 s	62.89 s	31.93 s
训练效果（线上最好AUC分数）	0.5082	0.5288	0.7181	0.7035
原理	Bagging	Boosting	Boosting	Boosting

主要树学习器性能比较

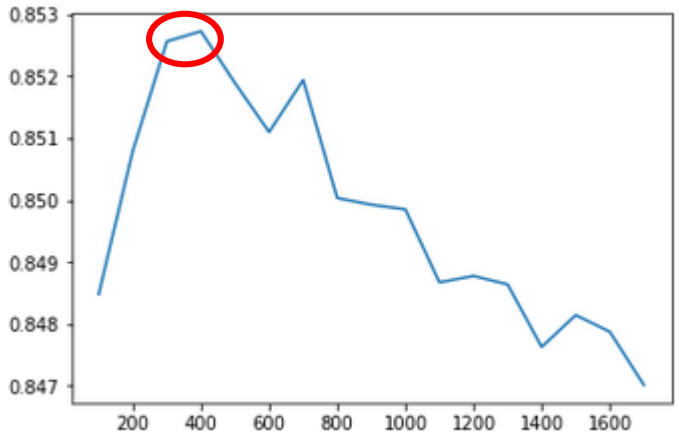
4.2.1 单模型（蓝色表格表示）

gbc_pred0819	
数据集	loan_month, loan_day, nodup , n_null, tag
学习器	单个GDBT
线上验证集AUC	0.819

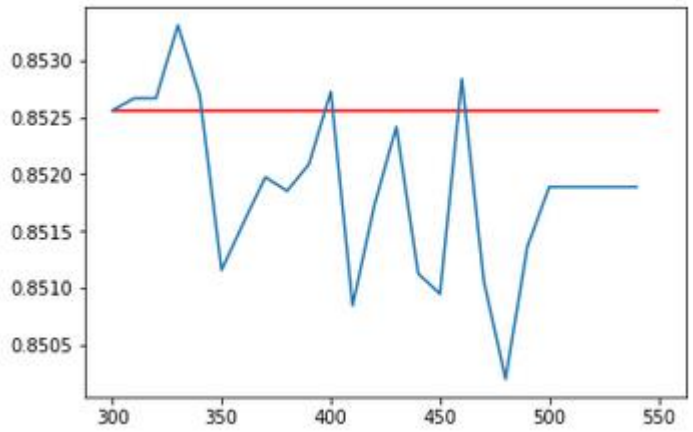
lgb_pred0819	
特征集	loan_month, loan_day, loan_week_day, nodup, n_null , tag, cross feature
学习器	单个lightGBM
线上验证AUC	0.819

xgb_pred_330_08242	
数据集	取“特征大集合”中特征重要性分数**前330**的特征
学习器	单个xgboost
线上验证集AUC	0.8242

注：nodup, tag, loan_month, loan_day, loan_week_day, null_sign, majcnt, cross_feature 为“特征大集合”



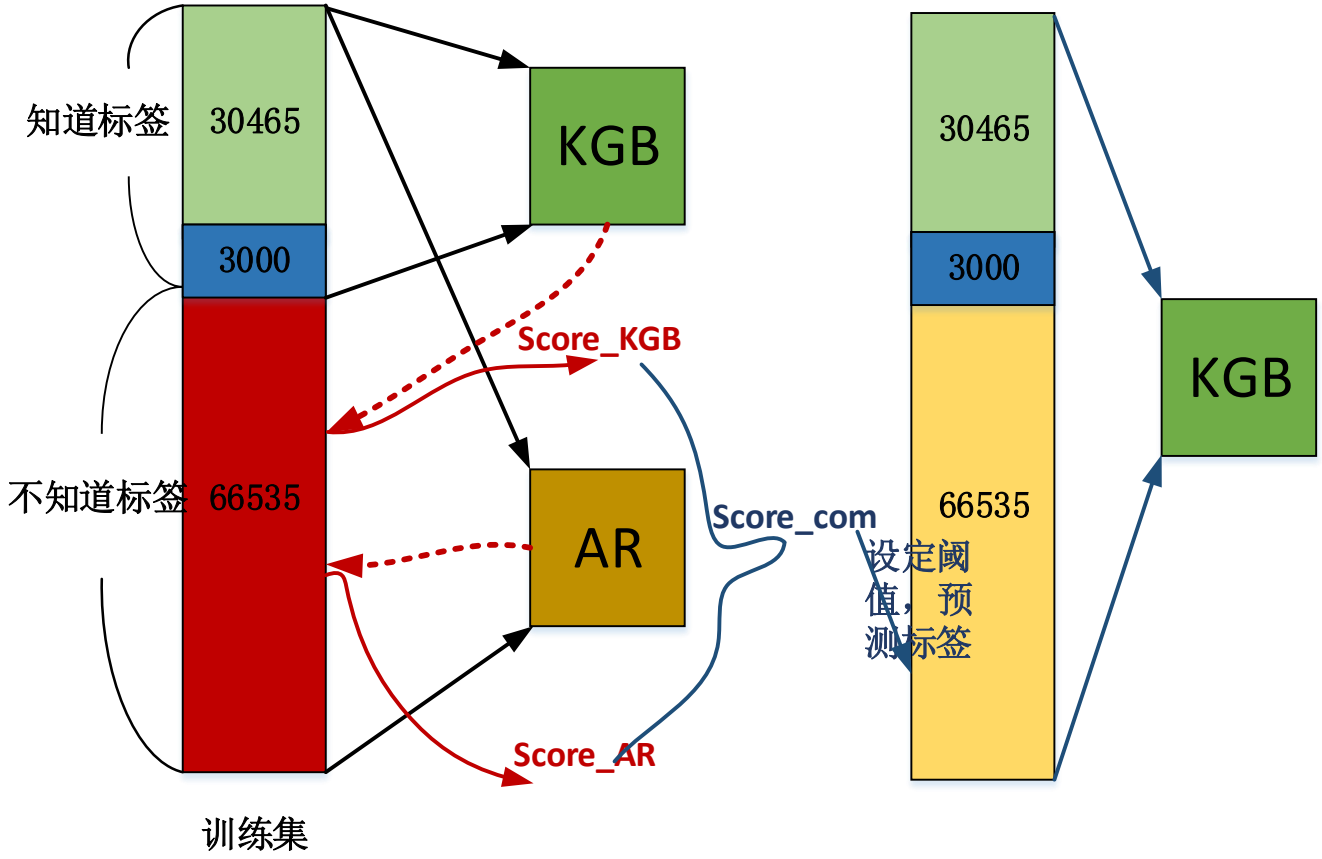
粗搜索



精搜索

4.2.2 半监督模型

semi_pred1	
数据集	loan_month, loan_day, nodup, n_null, tag（**33465条有标签数据+6653个负例**（unlabel数据）+**1997个正例**（unlabel数据））
学习器	单个xgboost
线上测试集AUC	0.8200左右



结合了“历史模型”的区分能力与现有数据的信息量

AR　　KGB

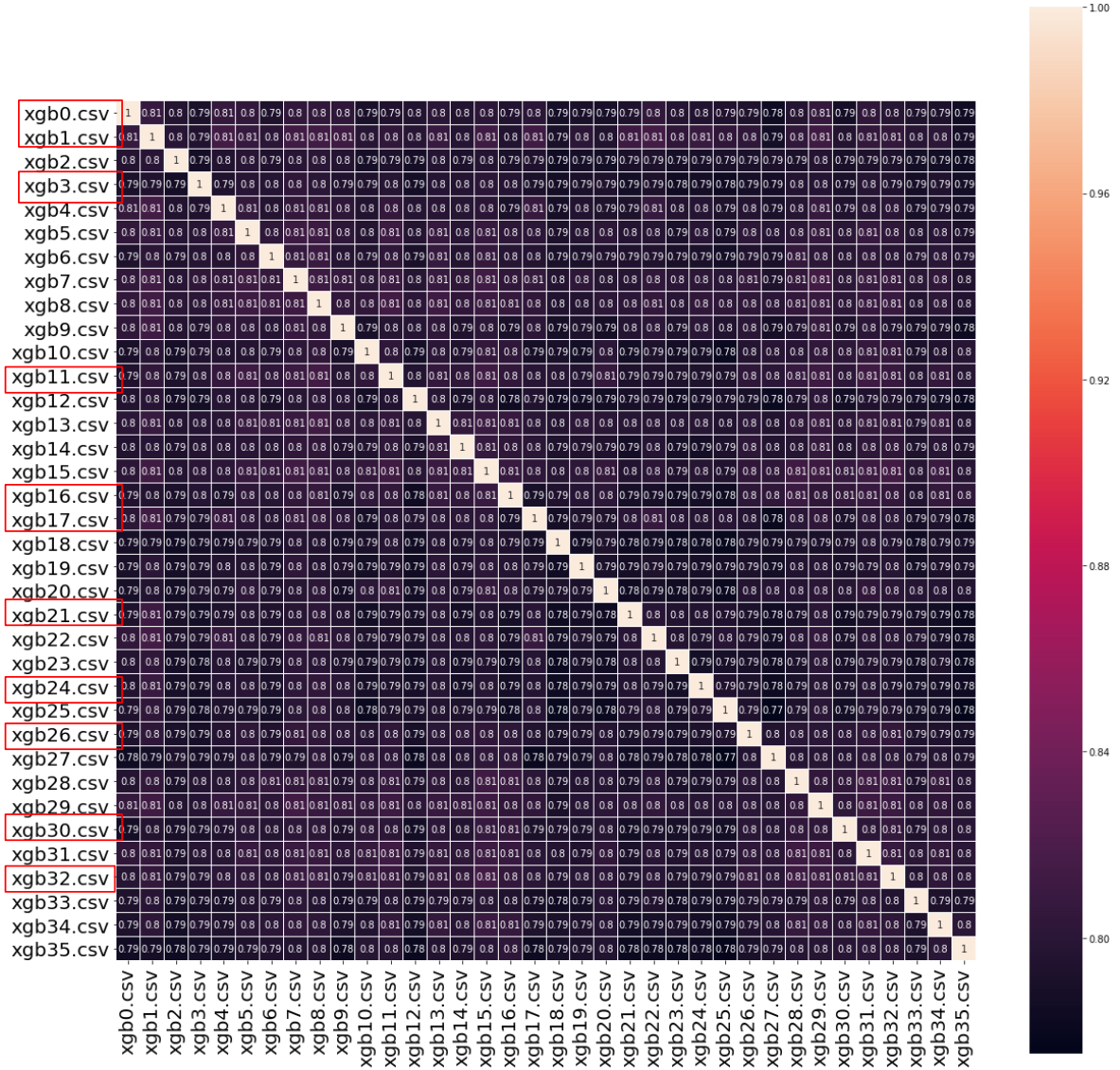
$$Score\_com = (1-a)*Score\_AR + a*Score\_KGB$$

4.2.3 多个同种类学习器融合(绿色表格表示)

lgb_pred_en_8079	
数据集	loan_month, loan_day, nodup, n_null, tag
学习器	20个lightGBM
线上验证集AUC	0.8079

xgb_pred0829	
特征集	loan_month, loan_day, nodup, n_null
学习器	36个xgboost 做参数扰动，再做blending
线上验证集AUC	0.8296

xgbtest1_pred_083046	
特征集	loan_month, loan_day, nodup, n_null
学习器	36个xgboost 做参数扰动，根据mic matrix绘制热力图，挑选相关度较低的模型融合，如图7所示。
线上验证集AUC	0.8305

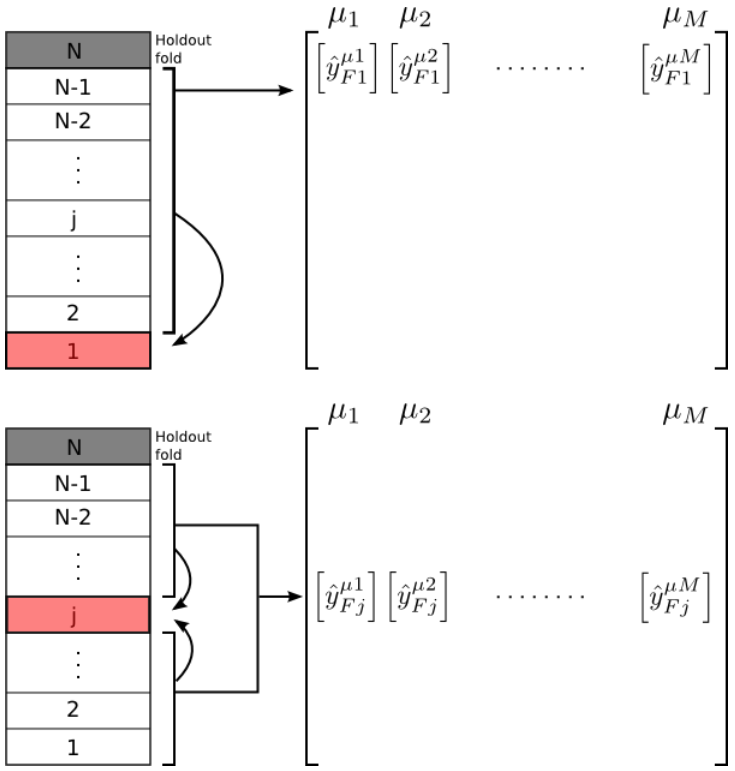


4.3 模型融合

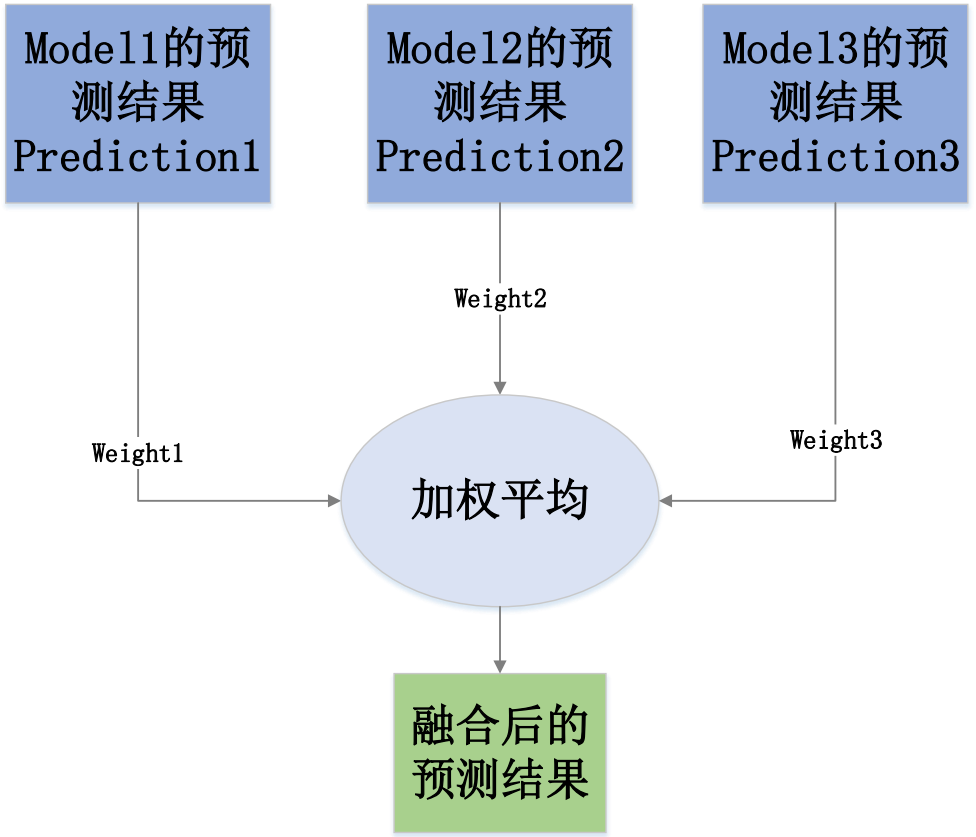
模型的结合策略：

4.3.1 学习法：Stacking（模型的多层叠加）

4.3.2 平均法：多个模型预测结果的线性组合



stacking-k folds

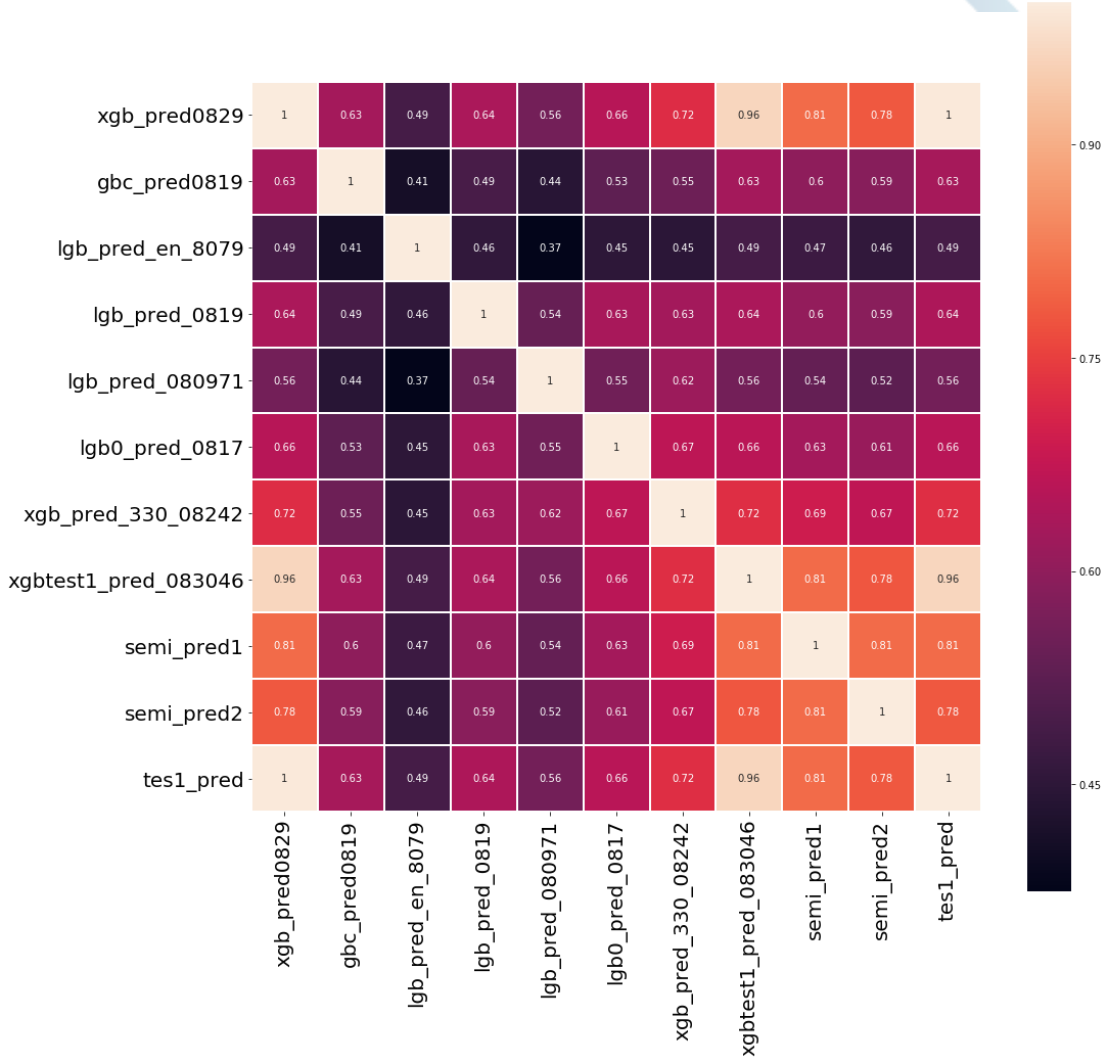


平均法

4.3.2 平均法

一般注意两个点：
1)模型本身的素质要尽可能好，即AUC分数应该尽可能高；
2)模型之间的相关度不宜过大

注：MIC是一个用于衡量两个序列相关程度的指标



4.3.2 平均法

融合步骤	计算公式	说明	测试集AUC
Step1（三个臭皮匠顶个诸葛亮）	Test1 = (3*xgb_pred0829.prob + gbc_pred0819.prob + lgb_pred_en_8079.prob + lgb_pred_0819.prob + lgb_pred_080971.prob)	使用上述的预测结果，做线性组合，给予auc高的模型更多的权重。Test1是在验证集上得分最高的组合。	0.82399
Step2（同一个“平台”）	Test1 = (3*xgb_pred0829.prob.rank() + gbc_pred0819.prob.rank()+ lgb_pred_en_8079.prob.rank()+ lgb_pred_0819.prob.rank() + lgb_pred_080971.prob.rank())	在Step1的基础上，使用排序融合，使得所有的预测结果都能在同一个“平台”上进行融合。	0.8253
Step3(“优中选优”)	Test2 = (test1_pred.prob.rank() + gbc_pred0819.prob.rank() + lgb_pred_0819.prob.rank() + xgb_pred_330_08242.prob.rank() + semi_pred2.prob.rank())	Test2把Test1与其他得分比较高的模型做融合，融合的方式有别于Test1，使用的融合Rank average。	0.8260

致谢

感谢各位评委和选手的聆听！