拼多多优惠券使用行为预测

一、问题界定

1. 背景介绍

拼多多是国内主流的手机购物 APP，成立于 2015 年 9 月，用户通过发起和朋友、家人、邻居等的拼团，以更低的价格，拼团购买商品。拼多多作为新电商开创者,致力于将娱乐社交的元素融入电商运营中，通过“社交+电商”的模式，让更多的用户带着乐趣分享实惠，享受全新的共享式购物体验。对于各大电商平台，在“双十一”这种大促时间段，优惠券会起到非常大的促销作用。

2. 目标问题

（1）使用 Python 建立逻辑回归模型

（2）预测用户是否会在活动中使用优惠券

（3）找到对用户使用优惠券影响较大而因素

二、数据概览

1. 数据维度分析

维度	字段	含义	变量类型
用户	ID	记录编码	类别型
	age	年龄	数值型
	job	职业	类别型
	marital	婚姻状态	类别型
消费行为	default	信用卡是否违约	类别型
	returned	是否有过退货	类别型
	loan	是否使用信用卡付款	类别型
	coupon_used_in_last6_month	过去 6 个月使用的优惠券	数值型
	coupon_used_in_last_month	过去 1 个月使用的优惠券	数值型

是否使用优惠券（预测目标）	coupon_ind	在本次活动中是否使用优惠券	类别型

总计 25317 行数据，10 个字段，3 大维度。

2. 缺失值分析

	字段名	缺失值数量
0	ID	0
1	age	0
2	job	0
3	marital	0
4	default	0
5	returned	0
6	loan	0
7	coupon_used_in_last6_month	0
8	coupon_used_in_last_month	0
9	coupon_ind	0

数据中各字段无缺失值。

3. 重复值和离群值分析

	age	coupon_used_in_last6_month	coupon_used_in_last_month
count	25317.000000	25317.000000	25317.000000
mean	40.935379	2.772050	0.292847
std	10.634289	3.136097	0.765498
min	18.000000	1.000000	0.000000
25%	33.000000	1.000000	0.000000
50%	39.000000	2.000000	0.000000
75%	48.000000	3.000000	0.000000
max	95.000000	55.000000	15.000000

可以看出，用户年龄是从 18 岁到 95 岁变化，且 75%的用户年龄在 50 岁以下，可能存在离群值；近 6 个月使用优惠券数量从 1-55 内变化，且 75%的数据在 3 以内，也可能存在离群值；用户近 1 个月内使用优惠券数量在 0-15 内变化，看似正常。

```
sum(pdd.duplicated())
0
```

该数据集中没有重复值。

三、单变量分析

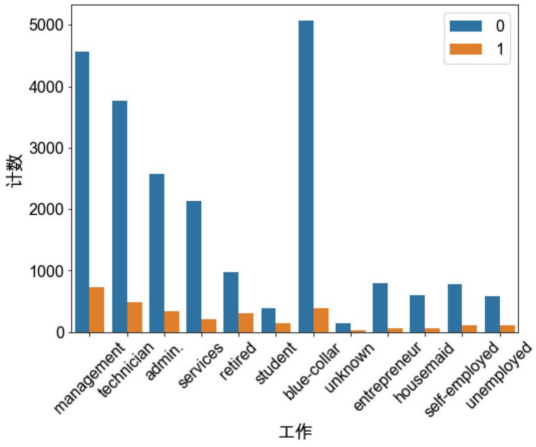
1. 预测目标中 0 和 1 的占比

类别	占比
0	0.883043
1	0.116957

数据集中 0 和 1 的占比严重失衡，建模时需要考虑如何优化。

2. 类别型变量

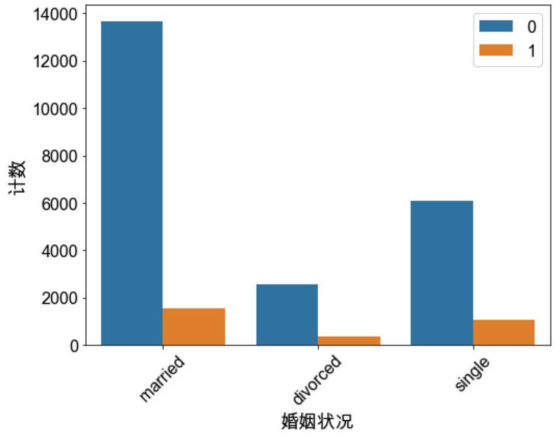
2.1 用户工作情况



coupon_ind	job	
0	blue-collar	0.226740
	management	0.203972
	technician	0.168188
	admin.	0.114868
	services	0.095321
	retired	0.043612
	entrepreneur	0.035293
	self-employed	0.034845
	housemaid	0.027062
	unemployed	0.026257
	student	0.017445
	unknown	0.006396
1	management	0.248565
	technician	0.162445
	blue-collar	0.130699
	admin.	0.115164
	retired	0.100642
	services	0.071260
	student	0.048294
	unemployed	0.038501
	self-employed	0.035461
	entrepreneur	0.022627
	housemaid	0.019588
	unknown	0.006754

使用和不使用优惠券人群中职业分布不同，选择使用优惠券的用户群体中从事管理行业的人居多，而不适用优惠券的人群中蓝领居多。

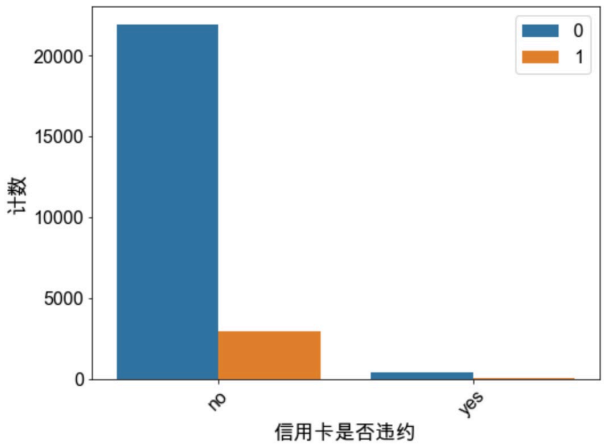
2.2 用户婚姻状况



coupon_ind	marital	
0	married	0.611916
	single	0.273260
	divorced	0.114824
1	married	0.528538
	single	0.353934
	divorced	0.117528

总体上看，使用和不使用优惠券的人群中结婚、单身、离婚人群的占比略有差别，但不显著。

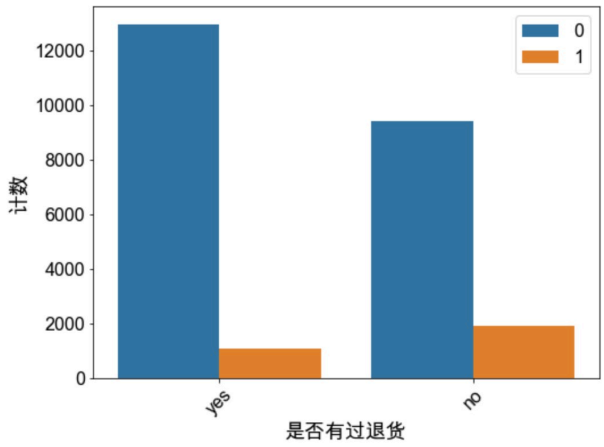
2.3 用户信用卡违约情况



coupon_ind	default	
0	no	0.981124
	yes	0.018876
1	no	0.991219
	yes	0.008781

可以看出，两个群体中均是信用卡不违约的人群占绝大多数，几乎无差异。

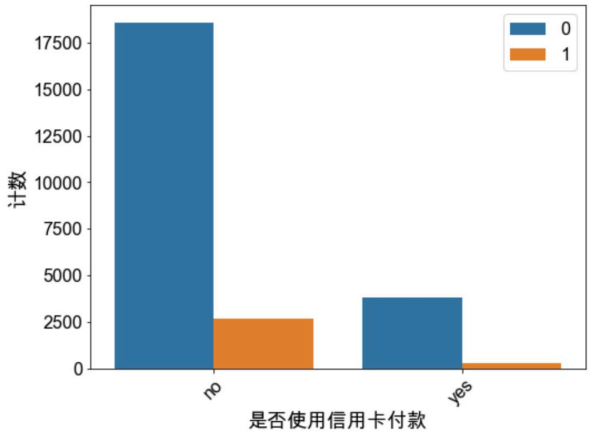
2.4 用户退货情况



coupon_ind	returned	
0	yes	0.579755
	no	0.420245
1	no	0.642351
	yes	0.357649

不会使用优惠券的人群中用户多有过退货行为，而选择使用优惠券的人群中多数用户不会退货。

2.5 用户使用信用卡付费情况



coupon_ind	loan	
0	no	0.830963
	yes	0.169037
1	no	0.905437
	yes	0.094563

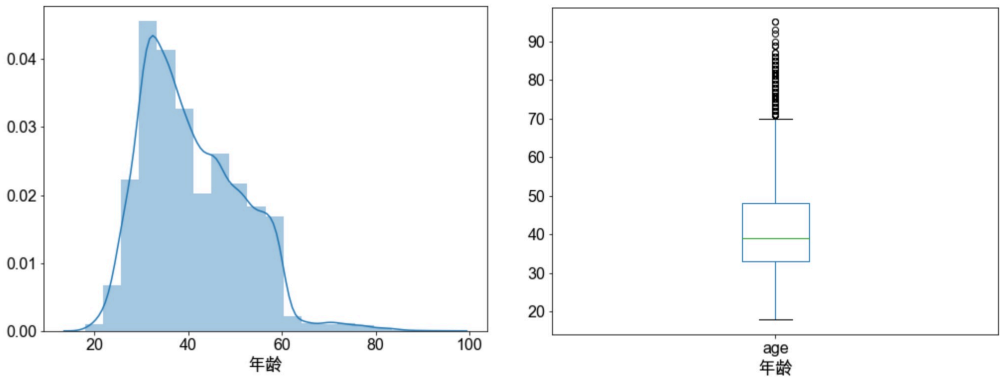
两个类别中均是使用过信用卡付款的用户占绝大多数，但选择使用优惠券的人群中的比例更高。

3. 数值型变量

coupon_ind	age	coupon_used_in_last6_month	coupon_used_in_last_month
0	40.819601	2.857846	0.260378
1	41.809524	2.124282	0.537994

可以看出，近 6 个月和近 1 个月用户使用优惠券的数量在 0 和 1 两个组别间均值存在差异，而年龄则差异不大。

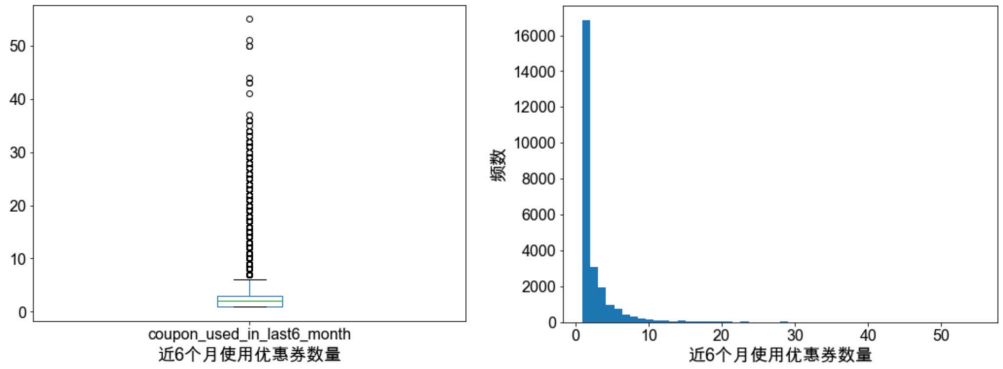
3.1 年龄



用户年龄主要集中在 50 岁以下，因此超过 70 岁的部分存在离群值，建模时

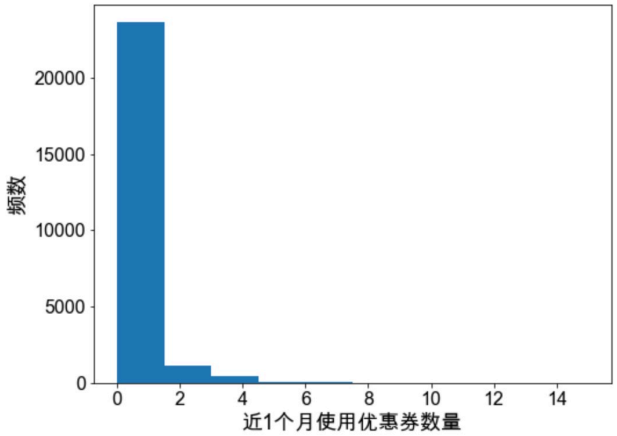
可以考虑分箱操作或删除离群值。

3.2 近 6 个月使用优惠券数量



用户近 6 个月使用优惠券适量主要集中在 10 以下且分布较为分散，后期数据处理考虑分箱操作或剔除离群值。

3.3 近 1 个月使用优惠券数量



用户近 1 个月使用优惠券数量拖尾情况正常，无需特殊处理。

四、数据处理

1. 去掉无用字段 ID

2. 将用户年龄分段，分为小于 25 岁（学生），25-45 岁（青年工作者），45 岁-65 岁（中年工作者）和大于 65 岁（老年），命名为 age_range

3. 将用户近 6 个月使用优惠券的数量分段，分为小于 10 和大于等于 10，命名为 coupon_last6_range

4. 将 coupon_ind 字段（预测目标）更名为 flag

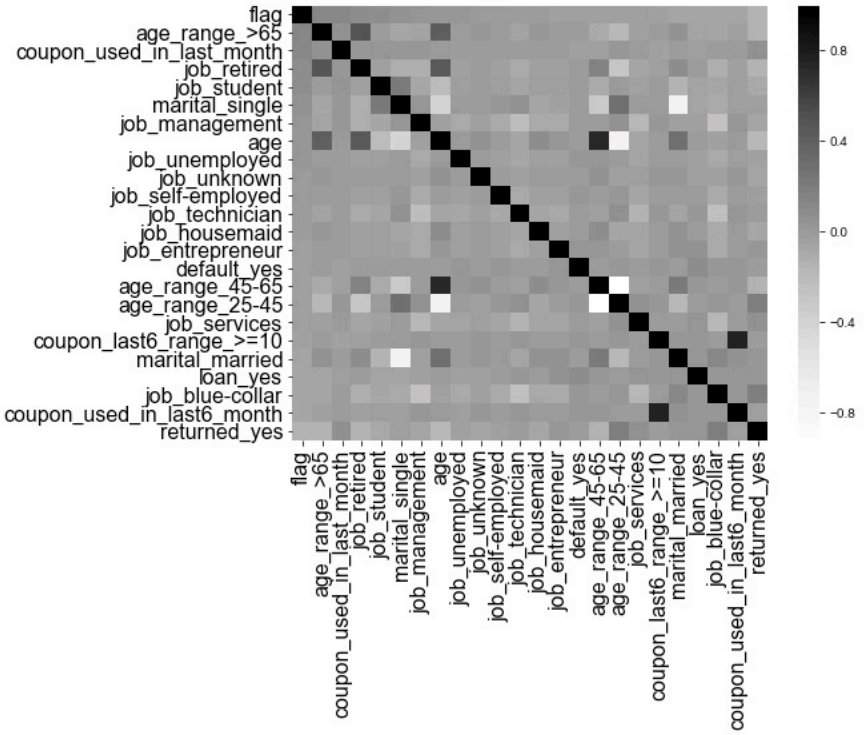
5. 将类别变量转换成哑变量，并删除冗余字段

五、多变量分析

1. 计算其余变量与预测目标 flag 的相关性系数

	flag
flag	1.000000
age_range_>65	0.130724
coupon_used_in_last_month	0.116550
job_retired	0.083868
job_student	0.069058
marital_single	0.057574
job_management	0.035234
age	0.029916
job_unemployed	0.023980
job_unknown	0.001438
job_self-employed	0.001078
job_technician	-0.004942
job_housemaid	-0.015041
job_entrepreneur	-0.022519
default_yes	-0.024608
age_range_45-65	-0.024717
age_range_25-45	-0.025546
job_services	-0.026688
coupon_last6_range_>=10	-0.046027
marital_married	-0.054746
loan_yes	-0.065231
job_blue-collar	-0.075065
coupon_used_in_last6_month	-0.075173
returned_yes	-0.143589

2. 热力图可视化



可以看出，颜色越深或越浅，两个变量的相关性越强。如变量 age_range_>65、coupon_used_in_last_month、job_retired、job_student、matital_single 与 flag 较强烈正相关，而变量 marital_married、loan_yes、job_blue-collar、coupon_used_in_last6_month、returned_yes 与 flag 呈现较强负相关性。

六、逻辑回归模型建立

1. 特征选择

用于构建模型的自变量应选择与因变量相关性较强，同时自变量之间尽量保证独立性（即相互无相关性，避免信息冗余），此处先人工设置一个 threshold=0.05，用于筛选出与 flag 相关性系数大于 0.05 或小于-0.05 的变量，即 age_range_>65、coupon_used_in_last_month、job_retired、job_student、matital_single、marital_married、loan_yes、job_blue-collar、coupon_used_in_last6_month、returned_yes 共 10 个。

2. 分割数据集

按训练集和测试集为 7:3 的初始比例分割数据集，先暂时不考虑数据不均衡的问题。

3. 训练模型并评估模型

对模型中的调试参数不作修改，以默认值训练模型。

（1）模型参数

参数名称	系数	概率比值
age_range_>65	1.069	2.912
coupon_used_in_last_month	0.409	1.505
job_retired	0.221	1.247
job_student	0.461	1.585
matital_single	0.262	1.300
marital_married	-0.001	0.999
loan_yes	-0.471	0.624
job_blue-collar	-0.300	0.741
coupon_used_in_last6_month	-0.154	0.858
returned_yes	-0.835	0.434
intercept	-1.437	0.238

对于参数的解读，以 age_range_>65 为例，假设 age_range_>65=0 时用户使

用优惠券的概率是 1-p，而 age_range_>65=1 时用户使用优惠券的概率是 p，因此该变量在模型中的系数为 1.069，表示老年用户（大于 65 岁）使用优惠券的概率是非老年用户（年龄小于等于 65 岁）的概率的 2.912 倍。

（2）评估模型

参数	结果
训练集准确率打分	0.8813
测试集准确率打分	0.8859
AUC	0.5098

若只看测试集中的 precision、recall 和 f1 评分，

参数	precision	recall	f1-score	数据量
flag=0	0.89	1.00	0.94	6733
flag=1	0.45	0.02	0.04	863

从上表可以看出，由于前面分析提到，本数据集较为不均衡（0 和 1 的比例为 8:1），因此造成 AUC 分数接近 0.5（模型随机性太强）且 flag=1 精确率、召回率极低，遇到新数据模型会优先判断其为 flag=0，需要进行优化。

4. 模型优化

以下参数均与原始模型作对比，打分参数数值若降低用蓝色，升高用红色标注。

4.1 提高测试集的比例

将训练集和测试集的初始比例 7:3 改为 5:5，重新训练模型并评估。

参数	结果
训练集准确率打分	0.8845
测试集准确率打分	0.8806
AUC	0.5092

若只看测试集中的 precision、recall 和 f1 评分，

参数	precision	recall	f1-score	数据量
flag=0	0.88	1.00	0.94	11162
flag=1	0.41	0.02	0.04	1497

可以看出，增加测试集的比例，准确率、AUC、精确率略有下降，因此还维持 7:3 不变。

4.2 修改自变量

（1）将 threshold 改为 0.01，筛选出与 flag 相关性系数大于 0.01 或小于-0.01 的变量并剔除冗余信息，即 age_range_>65、coupon_used_in_last_month、job_retired、job_student 、 matital_single 、 marital_married 、 loan_yes 、 job_blue-collar 、coupon_used_in_last6_month、returned_yes、job_management、job_unemployed、job_housemaid、job_enterpreneur、default_yes、age_range_45-65、age_range_25-45、job_services 共 18 个。

参数	结果
训练集准确率打分	0.8808
测试集准确率打分	0.8855
AUC	0.5086

若只看测试集中的 precision、recall 和 f1 评分，

参数	precision	recall	f1-score	数据量
flag=0	0.89	1.00	0.94	6733
flag=1	0.42	0.02	0.04	863

扩大自变量范围后，个别分数略有降低。

（2）将全部变量放进模型，并删除冗余信息，共 21 个变量。

参数	结果
训练集准确率打分	0.8809
测试集准确率打分	0.8855
AUC	0.5086

若只看测试集中的 precision、recall 和 f1 评分，

参数	precision	recall	f1-score	数据量
flag=0	0.89	1.00	0.94	6733
flag=1	0.42	0.02	0.04	863

将全部变量放入后，评估分数依然没有提高，因此自变量还维持 threshold=0.05 的情况。

4.3 对数据标准化处理

参数	结果
训练集准确率打分	0.8813
测试集准确率打分	0.8859
AUC	0.5098

若只看测试集中的 precision、recall 和 f1 评分，

参数	precision	recall	f1-score	数据量
flag=0	0.89	1.00	0.94	6733
flag=1	0.45	0.02	0.04	863

没有变化，因此也不采纳。

4.4 处理数据不平衡问题

算法主要分为上采样和下采样，上采样即通过算法增加数据集中少数类别的数量，而下采样则是通过算法剔除多数类别中的部分数据，最终的目的都是使得数据集中 0 和 1 的比例为 1:1。

（1）SMOTE 算法（上采样）

（2）RandomOverSampler 算法（上采样）

（3）RandomUnderSampler 算法（下采样）

（4）RENN 算法（下采样）

（5）NearMiss 算法（下采样）

（6）SMOTEENN（上采样+下采样）

（7）SMOTETomek 算法（上采样+下采样）

结果对比见下表。

算法	Initial	SMOTE	RandomOverSampler	RandomUnderSampler	RENN	NearMiss(v2)	SMOTEENN	SMOTETomek
训练集准确率打分	0.8813	0.6471	0.6487	0.6495	0.8475	0.9575	0.9306	0.6503
测试集准确率打分	0.8859	0.6583	0.6501	0.6320	0.8418	0.9651	0.9237	0.6519
AUC	0.5098	0.6583	0.6501	0.6320	0.5480	0.9645	0.8482	0.6518

测试集类别	算法	precision	recall	f1-score	数据量
flag=0	Initial	0.89	1.00	0.94	6733
	SMOTE	0.65	0.69	0.67	6689
	RandomOverSampler	0.64	0.68	0.66	6689
	RandomUnderSampler	0.64	0.63	0.64	906
	RENN	0.85	0.98	0.91	4768
	NearMiss(v2)	0.94	1.00	0.97	906
	SMOTEENN	0.93	0.97	0.95	4564
	SMOTETomek	0.65	0.68	0.66	6719
flag=1	Initial	0.45	0.02	0.04	863
	SMOTE	0.67	0.63	0.65	6725
	RandomOverSampler	0.66	0.62	0.64	6725
	RandomUnderSampler	0.62	0.63	0.63	871
	RENN	0.54	0.11	0.19	914
	NearMiss(v2)	1.00	0.93	0.96	871
	SMOTEENN	0.88	0.72	0.79	1148
	SMOTETomek	0.66	0.63	0.64	6693

对比后发现，在 flag=1 的类别下，所有算法都能在一定程度上优化测试集的精确率、召回率和 f1 评分，而综合选择 NearMiss(v2)算法。

4.5 在 NearMiss 算法的基础上改变损失函数权重

4.6 在 NearMiss 算法的基础上改变正则化规则（L1）

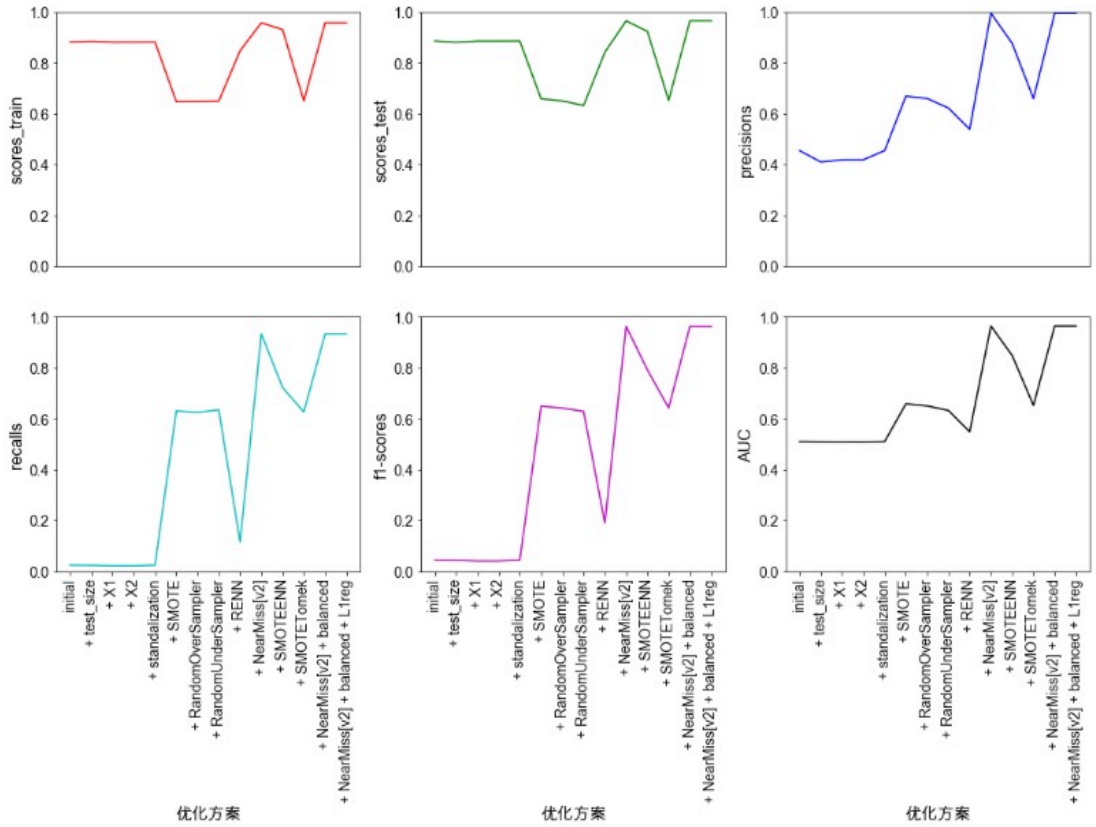
算法	Initial	NearMiss(v2)	NearMiss(v2) +balanced	NearMiss(v2) +balanced +L1reg
训练集准确率打分	0.8813	0.9575	0.9575	0.9575
测试集准确率打分	0.8859	0.9651	0.9651	0.9651
AUC	0.5098	0.9645	0.9645	0.9645

测试集类别	算法	precision	recall	f1-score	数据量
flag=0	Initial	0.89	1.00	0.94	6733
	NearMiss(v2)	0.94	1.00	0.97	906
	NearMiss(v2)+balanced	0.94	1.00	0.97	906
	NearMiss(v2)+balanced+L1reg	0.94	1.00	0.97	906
flag=1	Initial	0.45	0.02	0.04	863
	NearMiss(v2)	1.00	0.93	0.96	871
	NearMiss(v2)+balanced	1.00	0.93	0.96	871
	NearMiss(v2)+balanced+L1reg	1.00	0.93	0.96	871

综合来看，在 NearMiss 算法基础上，对模型中损失函数权重的改变和正则化规则的改变模型的评分均没有明显的提高，因此二者均不采纳。

4.7 可视化模型优化的各参数结果

将以上结果通过可视化的方式呈现，此处对于 precision、recall、f1-score 基于数据集的不均衡性，只选择呈现测试集中 flag=1 的类别情况。



可以看出，NearMiss 算法在几个方案中的显著优势。

5. 优化后模型参数解读

参数名称	系数	概率比值
age_range_>65	1.464	4.325
coupon_used_in_last_month	0.314	1.369
job_retired	-0.061	0.941
job_student	0.269	1.309
matital_single	0.042	1.043
marital_married	-0.308	0.735
loan_yes	-0.340	0.712
job_blue-collar	-0.132	0.877
coupon_used_in_last6_month	-1.242	0.289
returned_yes	-0.547	0.578
intercept	5.802	330.947

对于参数的解读，以 age_range_>65 和 returned_yes 为例，假设 age_range_>65=0 时用户使用优惠券的概率是 1-p，而 age_range_>65=1 时用户使用优惠券的概率是 p，因此该变量在模型中的系数为 1.464，表示老年用户（大

于 65 岁）使用优惠券的概率是非老年用户（年龄小于等于 65 岁）的概率的 4.325 倍；而假设 returned_yes=0 时用户使用优惠券的概率是 1-p，而 returned_yes=1 时用户使用优惠券的概率是 p，因此该变量在模型中的系数为-0.547，表示有过退货行为的用户使用优惠券的概率是没有过退货行为的用户使用优惠券概率的 0.578 倍，即没有过退货行为的用户更有可能在本次活动中使用优惠券。