数据分析与数据挖掘第三次作业第二大题 – c问分析文档

数据分析与数据挖掘

DAM COURSE, SPRING 2018

BY

1552674 李　源

同濟大學
TONGJI UNIVERSITY

Tongji University
School of Software Engineering

1 代码运行结果

1.1 ci问运行结果

这里我参照了文档要求，选择了如下指标作为评判分类器预测结果，分别为：总体的 precision 值，recall 值以及 auc值。同时为了弥补数据集数据集数量不足的问题，我分别采用了基于 SMOTE 算法的过采样和基于 SMOTEENN 的过采样，与不做正负样本平衡进行比较。

三种针对数据集处理后的评价结果和对应的ROC曲线如下：

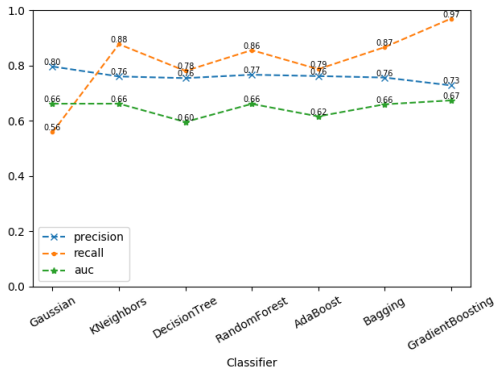

Figure 1.1: 不做正负样本平衡

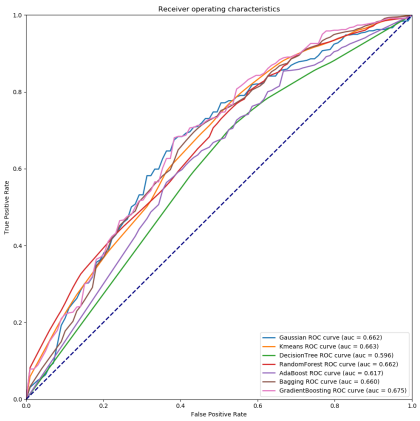

Figure 1.2: 不做正负样本平衡

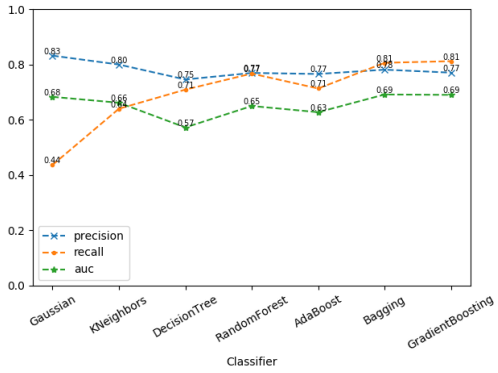

Figure 1.3: 基于 SMOTE 算法的过采样

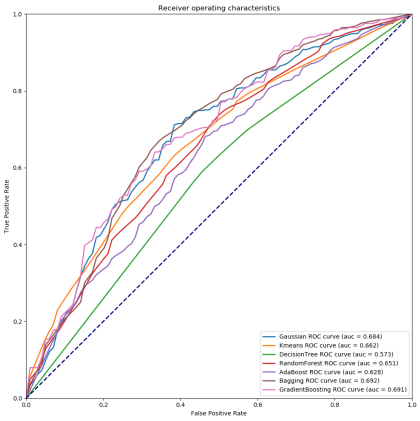

Figure 1.4: 基于 SMOTE 算法的过采样

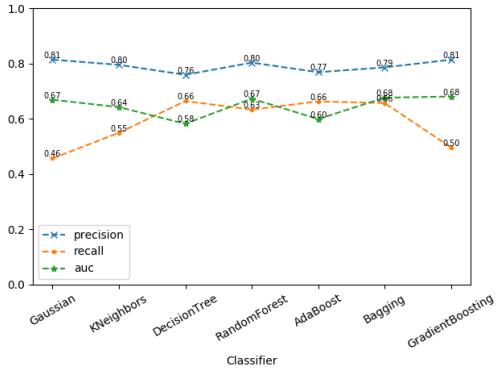

Figure 1.5: 基于 SMOTEENN 的过采样

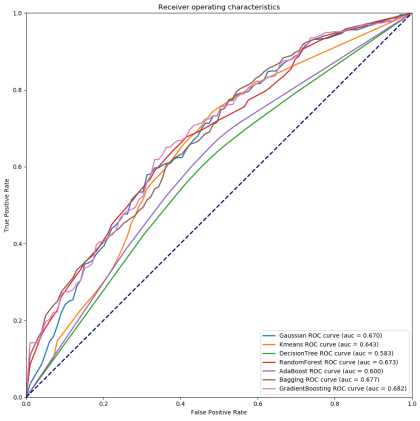

Figure 1.6: 基于 SMOTEENN 的过采样

七种分类器的 precision、recall、auc 结果汇总如下：

(a) 不做正负样本平衡

分类器	评价标准		
	precision	recall	auc
Gaussian	0.80	0.56	0.66
KNeighbors	0.76	0.88	0.66
DecisionTree	0.76	0.78	0.60
RandomForest	0.77	0.86	0.66
AdaBoost	0.76	0.79	0.62
Bagging	0.76	0.79	0.62
GradientBoosting	0.73	0.97	0.67

(b) 基于 SMOTE 算法的过采样

分类器	评价标准		
	precision	recall	auc
Gaussian	0.83	0.44	0.68
KNeighbors	0.80	0.64	0.66
DecisionTree	0.75	0.71	0.57
RandomForest	0.77	0.77	0.65
AdaBoost	0.77	0.71	0.63
Bagging	0.78	0.81	0.69
GradientBoosting	0.77	0.81	0.69

(c) 基于 SMOTEENN 的过采样

分类器	评价标准		
	precision	recall	auc
Gaussian	0.81	0.46	0.67
KNeighbors	0.80	0.55	0.64
DecisionTree	0.76	0.66	0.58
RandomForest	0.80	0.63	0.67
AdaBoost	0.77	0.66	0.60
Bagging	0.79	0.66	0.68
GradientBoosting	0.81	0.50	0.68

Table 1: precision、recall、auc 结果汇总

1.2 cii、ciii 问运行结果

这里我参照了文档要求，选择了如下指标作为评判分类器预测结果，分别为：总体的 precision 值，recall 值以及 auc值。同时为了弥补数据集正负样本不均匀的问题，我分别采用了朴素随机欠采样和基于 SMOTEENN 算法的过采样，与不做正负样本平衡进行比较。

1.2.1 cii 问结果

七种分类器的 precision、recall、auc 结果汇总如下：

(a) 不做正负样本平衡

分类器	评价标准		
	precision	recall	auc
Gaussian	0.28	0.36	0.73
KNeighbors	0.36	0.10	0.66
DecisionTree	0.00	0.00	0.68
RandomForest	0.44	0.12	0.71
AdaBoost	0.25	0.25	0.63
Bagging	0.41	0.13	0.74
GradientBoosting	0.00	0.00	0.76

(b) 朴素随机欠采样

分类器	评价标准		
	precision	recall	auc
Gaussian	0.23	0.53	0.76
KNeighbors	0.16	0.65	0.70
DecisionTree	0.17	0.75	0.72
RandomForest	0.17	0.62	0.73
AdaBoost	0.13	0.60	0.61
Bagging	0.17	0.63	0.73
GradientBoosting	0.17	0.67	0.75

(c) 基于 SMOTEENN 算法的过采样

分类器	评价标准		
	precision	recall	auc
Gaussian	0.23	0.58	0.76
KNeighbors	0.14	0.73	0.69
DecisionTree	0.15	0.79	0.71
RandomForest	0.25	0.50	0.76
AdaBoost	0.18	0.49	0.64
Bagging	0.24	0.51	0.76
GradientBoosting	0.20	0.62	0.77

Table 2: precision、recall、auc 结果汇总

三种针对数据集处理后的评价结果和对应的ROC曲线如下：

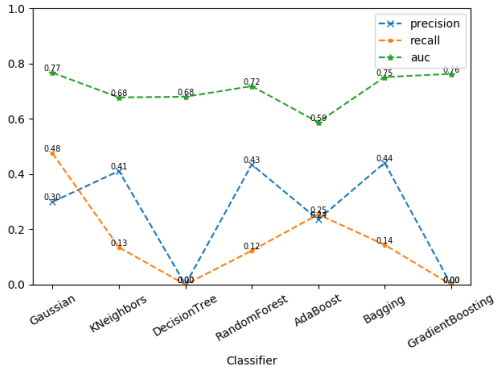

Figure 1.7: 不做正负样本平衡

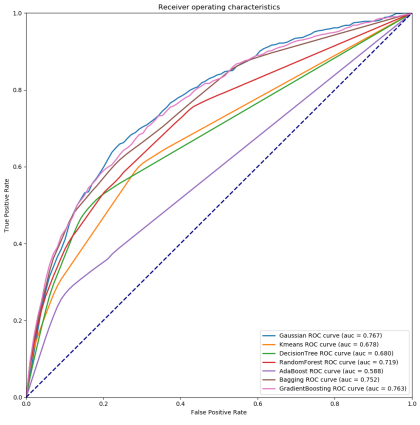

Figure 1.8: 不做正负样本平衡

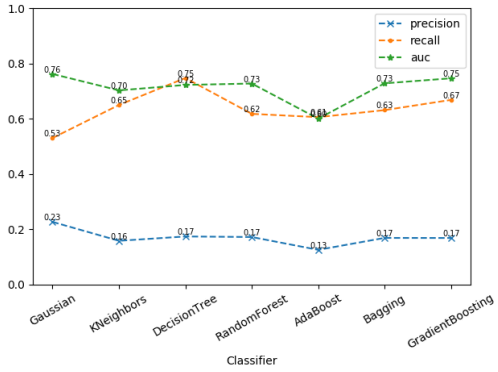

Figure 1.9: 朴素随机欠采样

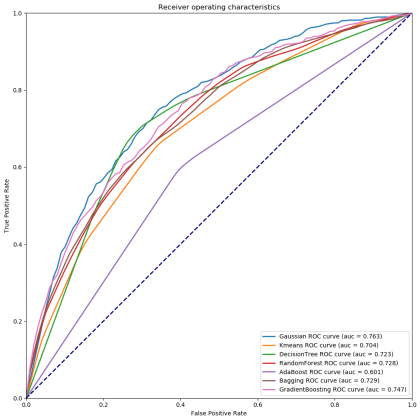

Figure 1.10: 朴素随机欠采样

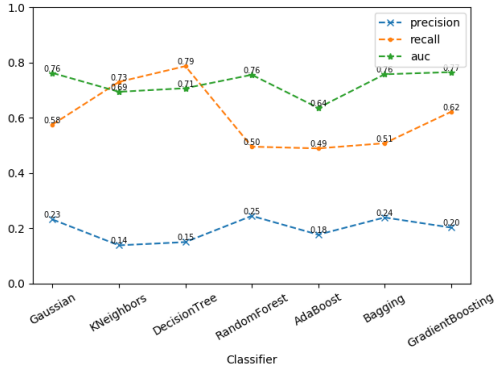

Figure 1.11: 基于 SMOTEENN 的过采样

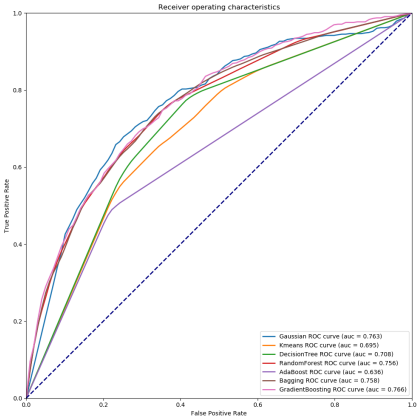

Figure 1.12: 基于 SMOTEENN 的过采样

1.2.2 ciii 问结果

三种针对数据集处理后的评价结果和对应的ROC曲线如下：

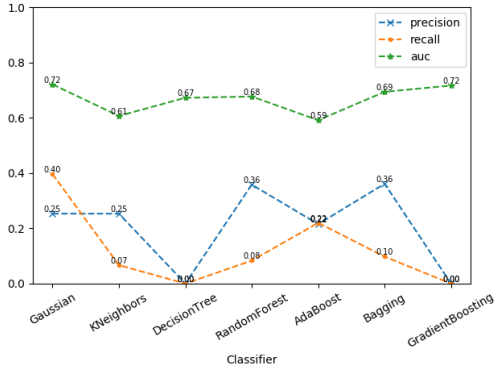

Figure 1.13: 不做正负样本平衡

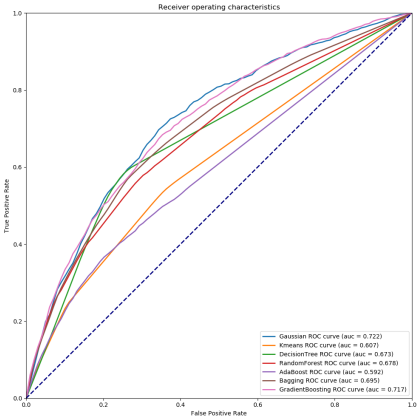

Figure 1.14: 不做正负样本平衡

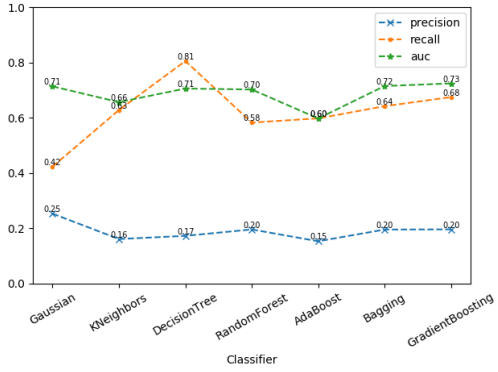

Figure 1.15: 朴素随机欠采样

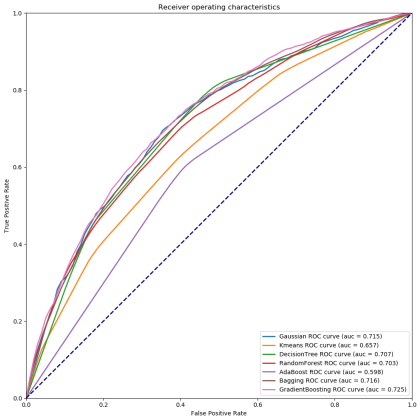

Figure 1.16: 朴素随机欠采样

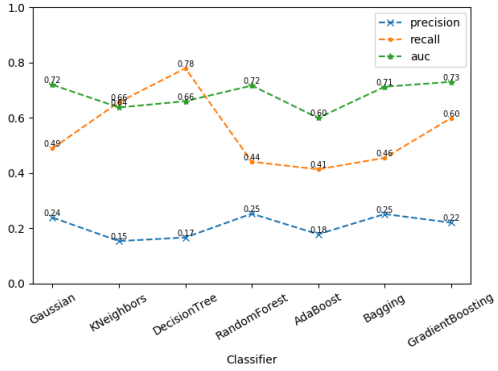

Figure 1.17: 基于 SMOTEENN 的过采样

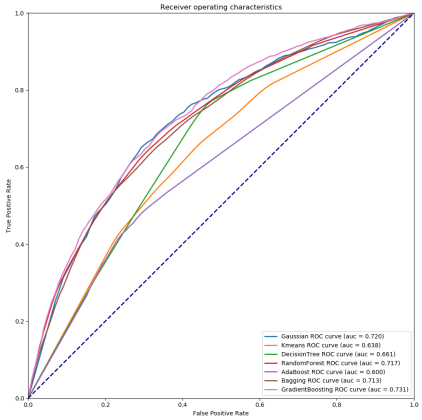

Figure 1.18: 基于 SMOTEENN 的过采样

七种分类器的 precision、recall、auc 结果汇总如下：

(a) 不做正负样本平衡

分类器	评价标准		
	precision	recall	auc
Gaussian	0.25	0.40	0.72
KNeighbors	0.25	0.07	0.61
DecisionTree	0.00	0.00	0.67
RandomForest	0.36	0.08	0.68
AdaBoost	0.22	0.22	0.59
Bagging	0.36	0.10	0.69
GradientBoosting	0.00	0.00	0.72

(b) 朴素随机欠采样

分类器	评价标准		
	precision	recall	auc
Gaussian	0.25	0.42	0.71
KNeighbors	0.16	0.63	0.66
DecisionTree	0.17	0.81	0.71
RandomForest	0.20	0.58	0.70
AdaBoost	0.15	0.60	0.60
Bagging	0.20	0.64	0.72
GradientBoosting	0.20	0.68	0.73

(c) 基于 SMOTEENN 的过采样

分类器	评价标准		
	precision	recall	auc
Gaussian	0.24	0.48	0.72
KNeighbors	0.15	0.66	0.66
DecisionTree	0.17	0.78	0.66
RandomForest	0.25	0.44	0.72
AdaBoost	0.18	0.41	0.60
Bagging	0.25	0.46	0.71
GradientBoosting	0.22	0.60	0.73

Table 3: precision、recall、auc 结果汇总

可以发现，cii 问和 ciii 问的预测结果在三种评测指标上取得了相似的成绩。

1.3 civ 问运行结果

针对 civ 问这种回归问题，我使用了两种指标作为评价标准。指标一是对于所有的误差，都求绝对值，然后考虑类似第一大题的方法，做出其 CDF 曲线，求得中位误差；指标二则是所有误差均求相对值，即可能为正可能为负，然后计算 60% 的数据都是在什么误差范围内。其结果如下：

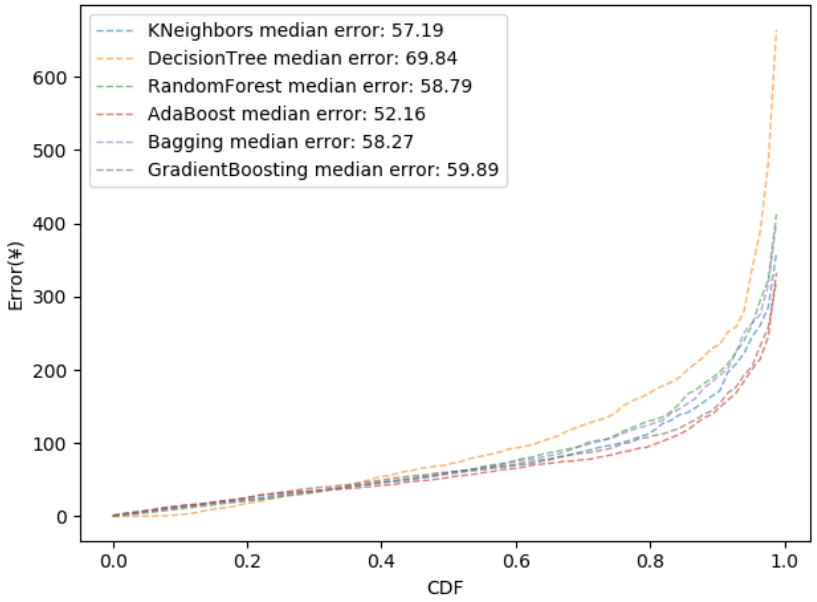

Figure 1.19: 绝对误差的 CDF 曲线

分类器	评价标准	
	中位误差(元)	60% 数据的误差范围(元)
KNeighbors	57.19	$-76.51 \sim 50.86$
DecisionTree	69.84	$-114.34 \sim 75.47$
RandomForest	58.79	$-93.38 \sim 43.49$
AdaBoost	52.16	$-69.40 \sim 45.05$
Bagging	58.27	$-92.299 \sim 37.19$
GradientBoosting	59.89	$-83.38 \sim 42.12$

2 分析讨论

这里我主要针对 ci 问、civ 问特征的选择，ci 问的预测结果，SMOTENN 过采样方法以及 cv 问我的方法进行讨论。

2.1 ci、civ 问特征选择

ci、civ 两问与其他问题不同的点在于，其分类的主体是 U 即 user 这一个单一的主体，而不像其他问，会加入 B、C、I 而构成一个二类的主体。那么在选择特征时，我加入了一些只与 U 相关的特征，具体来说如下：

特征种类	特征含义	分组方法	特征个数
TYPE.4 complex feature - repeat feature	重复购买者	I 分组	1
		B 分组	1
		C 分组	1

除此之外，我也加入了相关的按照 monthly 分组后统计的 AGG 相关的特征，以及按照 UI、UB、UC 进行分组后得到的部分特征。

我这样做的目的是，这两问是为了求得和 U 相关的一些预测结果，那么我觉得需要在一定程度上加强与 U 相关的特征的数量和重要性，而弱化只与 I、B、C 相关的特征的重要性。

2.2 ci 问的预测结果

ci 问的结果与 b 问 和 cii、ciii 问的一个很大的不同点是，其预测结果的 precision 评价标准要明显地高于其他问。因为我在分析 b 问时候，发现了如果数据集的正负样本差异较大，可能会造成 precision 过低的情况，所以对于 ci 问，我猜测可能是因为其数据集正负样本不像其他问那么大。于是我统计了其正负样本的个数：

样本种类	样本个数
负样本	109
正样本	223

很显然，与其他问相比，ci 问的正负样本分布更加均匀，所以其结果在 precision 上取得了更好的结果。

2.3 SMOTENN 过采样方法

在做 c 问的时候，我查到了一种新的过采样方法，叫 SMOTENN 过采样。之前我在 b 问所使用的过采样方法为 SMOTE 过采样，这种方法可能会存在一个问题，即可能会在边缘离群点和内点之间插入一些噪音点来生成噪声样本。

而 SMOTENN 方法则在一定程度上解决了这个问题，其在 SMOTE 的基础上，又添加了 edited nearest-neighbours 方法，来通过将欠采样和过采样结合的方法，补充数据集。这么做的一个好处是，能够让生成的新数据集分布在原始的对应正负的数据集更近的地方。下面图是一个示例：

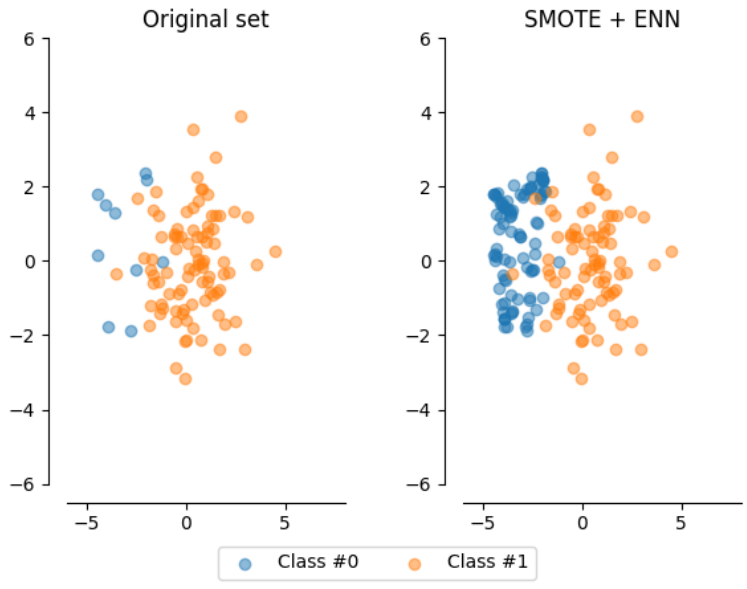

Figure 2.1: SMOTENN 结果

2.4 cv 问的方法

2.4.1 步骤说明

这里我想说明一下我的 cv 问的方法，其大致的步骤如下：

- 抽取 b 问运用到的特征，加上与购买商品数量相关的特征，然后利用在 Adaboost 进行回归，获得对于一个 vipno-pluno 的对，其 pluno 被购买了的数量。
- 针对 获取到的数量，将其前三个月对应的数量，合并到一起，然后乘以购买系数，再除以 4。
- 针对 U 即 vipno 进行分组，得到 vipno 其总共购买了多少的 pluno 及其对应的数量。
- 根据其单位，如果单位不是“千克”，而是类似于“袋”、“盒”之类的，则将数量转变为整数。
- 利用 civ 问预测得到的总金额，进一步修正，如果计算到当前的总金额大于预测得到的金额，则按照购买系数的由小到大删去部分 pluno。
- 重复上述步骤，直到总金额小于预测得到的金额。

首先我想说明的是，为什么我想到将其前三个月对应的数量合并到一起进行处理。我检查了第一步预测的结果，发现其结果存在一个问题，大量的结果都是一样的，且为小数。

查看原始数据可以发现，实际上大部分的商品都应该按照整数出售，而且大部分的用户都只购买了 1 个。由于这两个原因，导致了我第一步的预测结果并不是很好，因此我尝试加入了前三个月的数量，进行修正。

在第二步用到的购买系数的定义是：

针对某一个 pluno，我考虑了它是否被对应的 vipno 所购买，以及其所属的 bndno 和 dptno 是否被 vipno 所购买，然后在此基础上，得到了如下的购买系数公式：

$$cof = v * 0.5 + b * 0.2 + c * 0.3$$

其中 v、b、c 分别代表 pluno 是否被 vipno购买，以及其所属的 bndno 和 dptno 是否被 vipno 所购买，如果被购买，值为 1；否则为 0。 我将 bndno 的系数设置得比 dptno 较小是因为，在原始数据集中，有许多 bndno 为空值，如果将其系数设置过大，可能会导致缺省值的影响更大。

2.4.2 优缺点分析

cv 问我这样做的优点，我认为是比 b 问考虑了更多的隐含特征，这些特征是在原始数据中无法提取出来的，即总金额和购买系数。这在一定程度上优化了预测的结果。

同时，cv 问的预测结果是 pluno 被购买的数量，这要比 b 问更近一步。

但是 cv 问的一个很大的不足，或者说是易受到影响的因素，就是用到的前面几问预测的结果的准确度。在这里我选择的是 b 问中 RandomForest 分类器的结果和 civ 问中 Adaboost 回归得到的结果，其在之前的评价结果都是七种分类器中较好的。

虽然是最好的，其结果仍然没有达到特别高的准确率，那么将其应用到 cv 问中时，势必会带来前问的误差，这是一个 cv 问的很受影响的地方。

3 性能分析

比较 b 问、ci 问、cii 问和 ciii 问的耗时可以发现，特征使用的数量和数据集的数量是耗时的主要影响因素。除此之外，对于 type1 - type2 这种特征的生成和使用的耗时要明显高于单一分组的特征的耗时。