消费金融场景下的用户购买预测

南开大学数据智能分析与处理实验室考核

王丽娟 | 2018.9.12

目录

[问题分析 1](#_Toc524651394)

[数据处理 1](#_Toc524651395)

[验证指标 1](#_Toc524651396)

[模型构建 2](#_Toc524651397)

[svm算法 2](#_Toc524651398)

[原始模型： 2](#_Toc524651399)

[权重平衡svm 2](#_Toc524651400)

[融合svm 3](#_Toc524651401)

[决策树 3](#_Toc524651402)

[原始模型 3](#_Toc524651403)

[融合decision tree 3](#_Toc524651404)

[逻辑回归 3](#_Toc524651405)

[原始模型 3](#_Toc524651406)

[Smote合成数据LR 4](#_Toc524651407)

[融合LR 4](#_Toc524651408)

[神经网络 4](#_Toc524651409)

# 问题分析

由于数据经过简化，只有每个用户的30维特征和对应的行为label，用户的行为是不买或者买，对应0和1，是一个二分类问题。没有其他的信息，我们的任务就是利用这三十维信息构建模型进行预测分类

# 数据处理

代码：github.com/wlj961012/data.py

给定数据格式为：data\_agg.csv（features，useid），data\_.csv（useid，behavior）。按以下步骤处理:

1. 根据useid来连接特征与标注，useid丢弃，将数据保存为data.npy
2. 随机划分数据集6：2：2训练集、测试集和验证集。并存成文件。

# 验证指标

代码：github.com/measure.py

Precision：查准率指标，即预测结果为正例时，确实为正例的占比。公式为：TP/（TP+FP）

Recall：查全率指标,即有多少正例被成功的识别。公式为：TP/（TP+FN），在用户购买预测上，尽量挖掘所有潜在客户（recall）比识别用户的精确度（precision）更重要，所以recall指标在这个实例中更为关键。

F1：兼顾Precision和recall的指标,f1值中precision和recall视为一样重要。模型预测结果概率分布，通过选择阈值计算Precision，recall，其中PR曲线中最外凸的点即该问题最大的f1。公式为：2\*P\*R/（P+R）

Accuarcy：正确率指标，即预测结果与标签一致的数量占总量的比值。公式为：

（TP+TN）/（TP+TN+FP+FN）， 其中，若二分类中，某一类的数量远大于另一类（比如：负类远大于正类），模型将所有样本分为负类可以获得接近100%的Acc。不适合用于评价该任务。

# 模型构建

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验评价  实验模型 | Precision | Recall | F1 | Accuracy | AUC |
| Svm | 0.12 | 0.19 | 0.14 | 0.91 | 0.56 |
| Weight-balanced SVM | 0.13 | 0.29 | 0.18 | 0.89 | 0.69 |
| Aggregate SVM | 0.12 | 0.46 | 0.19 | 0.84 | 0.73 |
| Decision Tree | 0.16 | 0.33 | 0.22 | 0.91 | 0.72 |
| Aggregate Decision Tree | 0.16 | 0.43 | 0.24 | 0.89 | 0.74 |
| Logistic regression | 0.14 | 0.39 | 0.21 | 0.88 | 0.73 |
| Upsampling LR | 0.14 | 0.39 | 0.21 | 0.88 | 0.72 |
| Aggregate LR | 0.15 | 0.34 | 0.21 | 0.90 | 0.73 |
| Two-layer NN | 0.14 | 0.45 | 0.21 | 0.87 | 0.74 |

表1：实验结果-f1

**实验评价计算说明，训练集：验证集：测试集=6：2：2，在训练集训练完毕后，在验证集寻找最佳f1值的threshold，在测试集用此threshold进行对输出的概率进行二值化。得到结果计算评价标准。**

**由于该问题给出的评价标准是f1，即precision，recall是同等重要。实验求取最大的f1值就兼顾二者，所以recall不是很高。但实际上，行为预测更关心recall，即不让潜在用户流失，所以可以使用度量，beta=2，即recall为precision两倍的重要性，更符合这个问题。**

## svm算法

代码：github.com/wlj961012/SVM.py

### 原始模型：

使用参数：核函数：高斯核

实验结果：模型偏向于将结果分为负类，有91%的正确率，但是因为没有正确识别正类，所以recall和precision很低。AUC值表明该分类器接近随机分类器。

实验改进：由于给定数据中正类样本大约仅占4%左右，属于样本不平衡问题。所以添加weight\_loss项来增加正类权重。

### 权重平衡svm

使用参数：class\_weight：balanced

实验结果：模型由于设置了balanced参数，模型会自动根据训练的样本数进行权重平衡，给予正类更大的权重。recall比之前大幅提升，说明模型对识别正类有一定的效果，accuarcy有所下降。

实验改进：希望在增加recall的同时不要大幅损害其识别精度，并且svm方法不太适合大规模数据集，同时考虑均衡正负样本。我们使用下采样方法。

### 融合svm

使用参数：subset=10有十个子svm模型用于投票。

首先对数据的负类进行随机抽样，抽出与正类相同数量的样本，构成十份1：1的子数据集，分别对用每一个小数据集训练一个svm分类器，最后将每一个svm预测的结果进行投票，得出最后结果。

实验结果：模型由于分散为十个子数据集，模型容错性增强，并且训练变得相对快了不少。Recall大幅增加，precision没有明显下降。

## 决策树

### 原始模型

使用参数:max-depth=6 最大设置六层，防止过拟合

实验结果：决策树在类别不平衡问题的初始表现较好，在验证集上找到最优的参数depth=6，层数太少会欠拟合，太多会过拟合。

实验改进：仿照之前思路，仅使用原数据集，在多数的负类样本进行采样，构成十份1：1的均衡数据集，进行模型融合。

### 融合decision tree

实验结果：在目前的f1评价标准上，取得了最优的效果，与之前的decision tree在精度上持平，但是的recall上有大幅提升。

## 逻辑回归

### 原始模型

使用参数：class\_weight

参考前面的svm算法平衡数据集的方式，我们在原始模型中使用了class\_weight参数。

实验结果：Logistic 回归使用最优化方法求解参数，速度较快，由于使用了类平衡，模型初期效果良好。

实验改进：考虑使用上采样数据的方式，根据正类的分布合成部分数据来进行模型训练。Svm由于使用大量的数据会增加其运算负担，不适合使用该方法。

### Smote合成数据LR

使用参数：合成数据数量：10000

考虑到对正类进行大量数据合成会导致过拟合该分布，并且会出现大量重复，所以仅拟合至一定数量并且配合class\_weight=”balanced”来进行权重的调整。

实验结果：没有显著提升，合成的数据质量可能不高

实验改进：考虑使用类似融合svm多个子模型的方法，减轻一个模型带来的评估不准确问题。

### 融合LR

类似融合svm，我们将数据集划分为十个subset，为每一个subset训练一个LR模型，最后对测试集，综合十个模型的评估进行分类。

实验结果：实验效果比之前有下滑，LR模型融合没有带来性能的提升。

## 神经网络

代码：github.com/wlj961012/torchmodel.py

模型构建：pytorch构建简单的两层神经网络，选用cross-entropy loss，sgd优化器，learning rate=1e-2，学习率指一次参数更新的步长，epoches=2000，总迭代次数2000。Weight={0：0.1，1：0.9}，给予正类更大的权重。

实验结果：实验结果与前面融合的decision tree很接近。