基于数据挖掘方法的公司客户流失预测

梅佳奕
数据科学与工程学院，华东师范大学
10165300206@stu.ecnu.edu.cn

Abstract

根据调查，企业平均 5 年就会失去一半的客户，而失去一个客户带来的巨大损失，可能需要拉进来 10 个新客户才可以弥补。在企业发展不断吸纳新客户的同时，及时检测具有流失倾向的老客户、将其挽回，可以更大程度地缓减、遏止损失，对企业来说具有重大意义。对于“收钱吧”这样为商户店铺提供服务的支付平台，店铺的流失倾向与此前该店铺的数据有着密不可分的联系。运用数据挖掘的一些模型方法，通过对每个店铺的交易流水、所属商户信息等数据的分析，提出对该店铺流失倾向的预测判断，从而帮助企业精准地挽回将要流失的老客户，在不增大工作负荷的同时，遏制住损失。

1 Introduction

在“收钱吧”公司平台中，记录着店铺营业的往来流水、店铺注册所属的商户信息等，从这些数据中可以提取出与流失倾向有关的特征信息。本项目使用了决策树、随机森林的算法分类模型，对店铺的流失进行预测。为了更好地预测流失率，定义：自统计日向前推 5 天，前 3 天有 10 笔以上大于 2 元的交易，后 2 天无交易的商户记为“抛出”；抛出商户在后续 28 天中没有交易，类别划为流失。项目的主要目的是从抛出的店铺中预测哪些是未来 28 天内不再有交易（流失）的门店。在公司原有的工作规则下，由于无法进行精准预测，需要人工对所有抛出的店铺电话回访进行挽回，工作量与人工成本较大，而此过程中 recall=1，precision=0.4，f1 score=0.571，命中率并不高，说明很大一部分的人工成本是无效的。在此项目的进展过程中，以 f1 score 为衡量预测结果好坏的标准，追求在提高 f1 score 的同时，削减企业的人工开销。

2 Methodology

此项目中，使用的数据集来此“收钱吧”公司，由交易流水 table_transaction、商户信息 table_datamining_merchant、抛出点信息 table_throwpoint 构成。各个实体间的关系可以用图 *1-ER* 图表示：

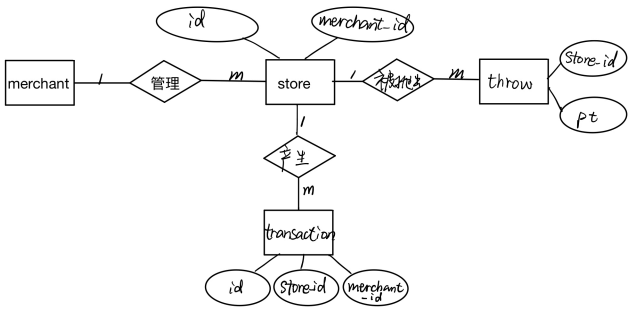


图 1: ER 图

2.1 特征提取

在提取有效的数据特征之前，先筛选出可用的数据：

1. 由于抛出的店铺最终不一定流失，一个店铺可能对应多个抛出点；最后一次抛出前的每一次”抛出“结果都是已知的（不流失），并非我们要做预测的抛出点，故仅保留每个店铺最后一次抛出点时间。

2. 由于做预测时并不能知道店铺在抛出点以后交易（未来），所以仅根据抛出点以前的信息做预测。删去训练集上最后一次抛出点以后的所有交易记录。

3. 在流水、商户信息表的多个维度中，有部分属性分布稀疏，呈明显的长尾分布。为了检验稀疏性是否可用，定义 $K_{col} = \frac{max\{c_i:\Sigma_i c_i=C\}}{C}$，其中 $i \in C, c_i$ 是 i 在 C 中的计数。K 值代表属性中单个值的集中程度，将 K 值过高的属性先行筛去。

在剩下的数据集上进行特征提取，共组合得到 12 个特征：

1. **行业等级 (industry_level)** $\in \{1,2,3\}$
2. **省份 province** $\in \{1,2,3\}$
3. **支付优惠率 price_reduced_rate** $\in (0,1)$
   $price\_reduced\_rate_{store\ x} = \frac{Count_{store\ x}(original\_amount - effective\_amount > 0)}{Count_{store\ x}(allrecords)}$
   通过**实付金额 effective_amount** 与**发起交易请求的金额 original_amount** 之间的差异，可知每一条交易**是否在支付通道中得到优惠**，再进行计数，能得到 *store_x* 所有交易纪录中发生优惠情况的比率。
4. **支付折扣率 reduce_range** $\in (0,1)$
   $reduce\_range_{store\ x} = \frac{Sum_{store\ x}(original\_amount - effective\_amount)}{Sum_{store\ x}(original\_amount)}$
   相当于是：将在支付通道中发生的优惠，转换成折扣力度。
   需要注意的是，支付通道中产生的优惠是不以店铺折扣、活动形式张贴宣传写出的，有可能以”用户的积分抵扣“的形式存在，属于通道方（收钱吧）给出的优惠，店铺最终到账依然是 original_amount。在这个过程中，惠及的是用户。考虑用户心理：趋于再次以受惠形式购买，影响作用于用户、当前交易发生后。
5. **（店铺）折扣率 discount** $\in N$
   除了在支付通道中会产生优惠，店铺提供的满减、折扣、会员等活动对用户刺激更大，且通常直接作用于当前交易。与该特征有关的字段为：**discount_channel**、**discount_channel_mch**、**discount_channel_mch_top_up**、**hongbao_channel**。由于这些信息都极其稀疏（$K_{col} > 0.9$），且金额较小，所以将它们弱化成 bool 类型以表示：店铺是否启用该优惠方式。
   计算公式：
   $discount_{store\ s} = Count_{store\ x}(bool(discount\_channel) + bool(discount\_channel\_mch) + bool(discount\_channel\_mch\_top)\_up) + bool(hongbao\_channel)$
   该特征的含义是店铺提供优惠活动的种类，即活动丰富程度。活动越丰富越容易吸引消费。
6. **提供支付方式种类 payw_offer** $\in$1,2,3,4
   店铺提供支付方式的多寡将影响消费的便利性，降低/抬高消费门槛，从而影响营业额。经过筛选，和此维度有关的字段为：bankcard_debit、wallet_weixin、wallet_alipay、alipay_huabei。
   $pay\_offer_{store\ x} = Count_{store\ x}[bool(bankcard\_debit) + bool(wallet\_weixin) + bool(wallet\_alipay) + bool(alipay\_huabei)]$
7. **日均交易笔数 daily_count** $\in [0, +\infty)$
   以 **original_amount** 作为标准的交易额数据，计算店铺自有纪录营业的起始日期（训练集 2018-04-01，测试集 2018-06-01），至抛出点的日均交易笔数。
   $daily\_count_{store\ x} = \frac{Count_{store\ x}(original\_amount > 0)}{Count_{store\ x}(all)}$
8. **单笔交易均额 original_amount_mean** $\in [0, +\infty)$
   鉴于店铺经营种类不同，用单笔交易均额的方式，可以区分出大宗买卖与小额零售的交易类型；即便对于同一类型的交易，我们常用“人均消费”的方法刻画它的受众群体。在这里，用**单笔交易金额**的信息来刻画这一维度，有助于对店铺类型、层级加以区分。
   $original\_amount\_mean_{store\ x} = Mean_{store\ x}(original\_amount)$
9. **经营状况 operate_state** $\in [0,1]$

要考虑行业经营状况，需将店铺营业信息与行业信息联合考虑。店铺发展除了受到行业本身的影响（**industry_level** 行业等级），还受到行业内竞争的影响；行业内竞争反应在经营状况（交易额）上，就是同等时间内，该行业市场中每家店铺分得”蛋糕“大小。

$operate\_state_{store\ x} = \frac{Sum_{store\ x}(original\_amount)}{Sum_{merchant\ \chi}(original\_amount)}, store \quad x \quad \in merchant\ \chi$

在时间对齐的问题上，要注意对于不同的店铺，分母”行业交易总额“总是统计到与分子”该店铺抛出日期“一样的时刻。

10. **付款率**

在总的交易流水中，大部分的交易类型是“付款”，但也存在有“退款”和“取消”情况，退款率/付款率可以反映一个店铺售出商品/服务质量的好坏。

11. **交易成功率**

对于不同类型的流水记录，存在因网络、机器故障等原因交易失败的情况。正常情况下成功率应该非接近 1（因为分母很大），那么如果某个店铺交易成功率显著地区别于 1，很有可能它的设备/网络外部条件存在故障，需回访更换。

2.2 模型 1：决策树

基于提取得到得特征值，使用 CART 算法构建回归决策树。

1.CAR Tree 的分支：二分支

CART 由算法决定，每个非叶节点只能产生两个分支。

2. 分类树确定最佳划分的度量：Gini 系数。

在信息熵、Gini 系数、分类误差这三种度量之间，选择 Gini 系数作为确定最佳划分的度量标准。这是因为基于信息论的熵模型常会涉及大量的对数运算，CART 分类树算法使用基尼系数来代替信息增益比，在简化模型同时保持了与熵模型相似的优良特性。

Gini 系数的计算式为：

$Gini(t) = 1 - \sum_{i=0}^{c-1}[p(i|t)]^2$

其中 $p(i|t)$ 表示给定节点 t 中属于 i 类的记录所占的比例。c 为类别数。Gini 系数代表了模型的不纯度，基尼系数越小，则不纯度越低，特征越好。

以 Gini 系数作为最佳划分度量标准，则在分类问题上选择最佳划分的方法为：按某一特征划分前后度量的差值，每次选取使数据差值最大的那个特征做最佳分支特征。这样的差值叫做 Gini 增益：

$$\Delta = I(parent) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j)$$

其中，$I(\cdot)$ 是给定节点的不纯性度量，N 是父节点上的记录总数，k 是属性值的个数，$N(v_j)$ 是与子女节点相关联的记录个数。因为对所有的测试条件来说，$I(parent)$ 是一个不变的值，所以最大化增益等同于最大化最小子女节点的不纯性度量。

那么，每次最佳分支特征的选取过程为：

(1) 先令 Gini 增益为 0：$\Delta = 0$, bestGini = I(parent)；

(2) 依次计算根据某特征（FeatureCount 次迭代）划分数据后的 Gini（计算方法为：划分后左右子数据集的总方差之和），如果 currentGini>bestGini，则 bestGini=currentGini, Δ =bestGini-I(parent)。

(3) 返回最佳分支特征、分支特征值（离散特征则为二分序列、连续特征则为分裂点的值），左右分支子数据集。

4.CART 的剪枝

剪枝是为了防止决策树算法过拟合，剪枝的基本策略分为“预剪枝”和“后剪枝”。预剪枝是指在构建决策树的过程中，先对每个结点进行估计，若当前节点的划分不能带来决策树表现能力的提高，则停止划分使当前节点成为叶节点；后剪枝则是先完整一颗完

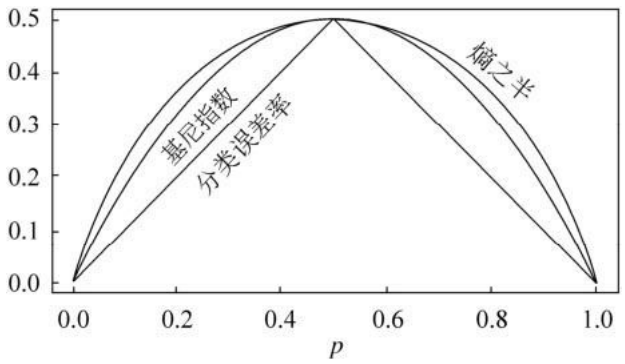


图 2: Entrophy-Gini-Classification error

整决策树的生成，然后自底向上地考察非叶节点，若将该节点对应的子树替换为叶节点能带来决策树泛化能力提升，则将该节点替换为叶节点。

为了获得最佳模型，树剪枝常采用预剪枝和后剪枝结合的方法进行。CART 的剪枝通常采用的是“后剪枝”策略。

2.3 模型 2：随机森林

随机森林属于集成学习中的 bagging 算法，bagging 的算法过程如下：

（1）从原始样本集中使用 Bootstraping 方法随机抽取 n 个训练样本，共进行 k 轮抽取，得到 k 个训练集。（k 个训练集之间相互独立，元素可以有重复）

（2）对于 k 个训练集，我们训练 k 个模型

（3）对于分类问题：由投票表决产生分类结果；对于回归问题：由 k 个模型预测结果的均值作为最后预测结果。（所有模型的重要性相同）

它的特点是：每构建一棵树，都通过随机选择样本数据来构建（有放回的），且对使用的输入指标也随机选择，目的是均匀取样，使每个样本权重相等。最终预测时各个预测函数可以并行执行，训练速度更快。

1. 模型的构建：

基于模型 1：CART 决策树构建随机森林的过程大致如下：

（1）从训练集中使用 Bootstraping 方法随机有放回采样选出 m 个样本，共进行 n_tree 次采样，生成 n_tree 个训练集；

（2）对于 n_tree 个训练集，我们分别训练 n_tree 个 CART 决策树模型；

（3）对于单个决策树模型，假设训练样本特征的个数为 n，那么每次分裂时根据信息增益/信息增益比/基尼指数选择最好的特征进行分裂；

（4）每棵树都一直这样分裂下去，直到该节点的所有训练样例都属于同一类。在决策树的分裂过程中不需要剪枝将生成的多棵决策树组成随机森林。对于分类问题，按多棵树分类器投票决定最终分类结果；对于回归问题，由多棵树预测值的均值决定最终预测结果。

其中 n_tree 是构建的模型数量：即构建多少棵树；

2. 模型的一些参数：

1. 样本大小：

是每次随机选择的样本占原来的百分比，如果是 1 的话，代表每次选择的样本数据与原来的数据量一样，如果是 0.9，则选择原来的数据量的 90% 作为的样本数据，在处理大数据集时，减少样本大小可以提高性能。

2. 是否需要处理不平衡数据：

模型的目标是标志结果（例如，流失或不流失）比率很小，那么数据是不平衡数据并且模型所执行的 Bootstrap 采样可能会影响模型精确性。要提高准确性，需选择该项；模型随后会捕获所需结果中的更大比例部分并生成更好的模型。使用加权采样选择变量：缺省情况下，每个叶节点的变量是使用同一概率随机选择的。要将加权用于变量并改进选择过程，请选中此复选框。

3. 指定要用于拆分的最小预测变量数：

如果是构建拆分模型，请设置要用于构建每个拆分的最小预测变量数。这防止拆分创建过小的子组。

4. 当准确性无法再提高时停止构建：

要改进模型构建时间，请选择此选项，以在结果的准确性无法提高时停止模型构建过程。

5. 缺失值最大百分比指定允许任何输入中存在的缺失值的最大百分比：如果该百分比超过了此数字，那么将从模型构建中排除此输出。

6. 排除单个类别多数超过以下值的字段指定单个类别可以在某个字段中具有的最大记录百分比：如果任何类别值表示的记录百分比高于指定值，那么将从模型构建中排除整个字段。

7. 最大字段类别数：

指定字段中可以包含的最大类别数。如果类别数超过了此数字，那么将从模型构建中排除此字段。

8. 最小字段变化：

如果连续字段的变异系数小于您在此处指定的值，那么将从模型构建中排除此字段。

9. 分箱数：

指定要用于连续输入的均等频率分箱数。可用选项包括：2、4、5、10、20、25、50 或 100。

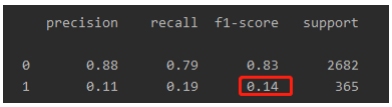
3 Result and Discussion

在实现过程中，决策树模型和随机森林模型实现使用了 scikit-learn 提供的接口来创建，在 11 个维度的特征集上，使用网格搜索的方法进行调参。

由于测试集的 lable 未知（待测），在训练集上训练模型时，按 8：2 分割数据，作为模型的验证集 x_val、y_val。

3.1 模型 1：决策树

首先，使用默认的参数创建初始的决策树模型，记为 t_0，此时在 y_test 测试集上的表现为：$f_1\ score = 0.16$，而根据助教返回在真实测试集集上的表现为：对



	precision	recall	f1-score	support
0	0.88	0.79	0.83	2682
1	0.11	0.19	0.14	365

图 3: f1 result

比两个结果，比较接近，但模型 t_0 的表现仍受到了泛化能力的限制。

由于模型泛化能力比较弱，对用于训练模型的特征值产生了思考；经思考调整后增添了两个维度：即特征集中的（10）**付款率**（11）**交易成功率**。经修改后的模型 t_1 表现为：$f_1\ score = 0.286$ 其混淆矩阵如图 4。

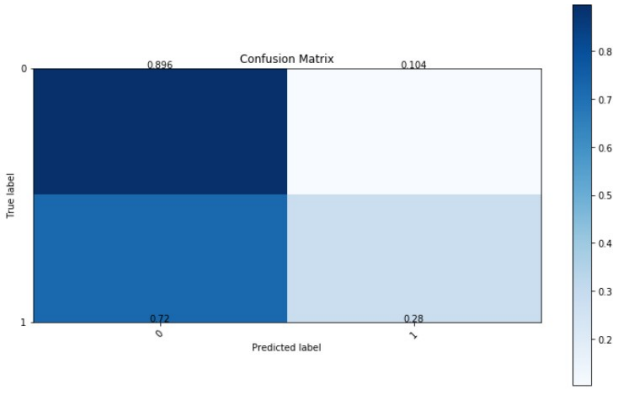


图 4: 混淆矩阵

接着使用网格搜索的方法自动调参，经调参后的模型 t_2 在验证集上交叉验证的表现为：$f_1\ score = 0.292$；在 x_test，y_test 上表现为：$f_1\ score = 0.242$。作出混淆矩阵如图 5。

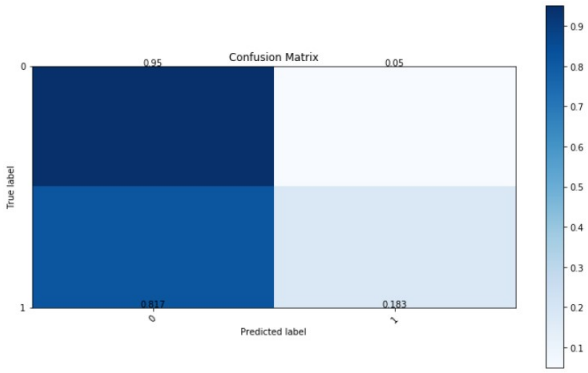


图 5: 混淆矩阵（调参后）

参数	值
max_depth	11
max_features	6
min_sample_leaf	5
min_samples_split	2

表 1: 决策树 t_3 参数

此时的参数如表 1 所示。

模型的泛化能力反而在调参之后降低了，产生了更严重的过拟合现象。

3.2 模型 2：随机森林

首先，使用默认的参数创建初始的随机森林模型，记为 F_0，此时在 y_test 测试集上的表现为 $f_1\ score = 0.16$。

为提高其表现能力，同样地采用网格方法调参。首先对参数 *n_estimators* 网格搜索，得到最佳 n_estimators = 13，此时在验证集上 $f_1\ score = 0.252$，在测试集上 $f_1\ score = 0.178$。

接着，对决策树最大深度 max_depth 和内部节点再划分所需最小样本数 min_samples_split 进行网格搜索。得到最佳 $max\_depth = 9$，最佳 $min\_sample\_split = 2$。此时在验证集上 $f_1\ score = 0.233$，在测试集上 $f_1\ score = 0.204$。

依次逐个对主要参数进行调参，其中注意部分参数间有联系，如所需最小样本数 min_samples_split 和叶子节点最少样本数 min_samples_leaf，不可单独固定一个搜索另一个，需要一起调参。（以上仅呈现对

n_estimators	13
max_depth	9
min_samples_split	2
f_1 score(val)	0.233
f_1 score(test)	0.204

表 2: 随机森林 F_1 参数与表现

f_1 *score* 有改进的参数）经调整后，模型最终参数与表现如表 2。

3.3 分析

在特征方面，上文 12 个特征的提取是由主观判断、构造的，其中不乏特征之间存在耦合、冗余的现象。将其看作重复信息的不同组成方式。由于不清楚不同特征在不同模型的表现如何，所以不预先进行人为筛选，在优化模型的过程中通过参数 $max\_feature$ 使模型自动选择。

特征总体分布情况在图 6 中以聚类热图的形式呈现，由于各特征对应离散/连续类型不同，且值域不一，所以对每一个特征维度进行规范化处理，即，减去均值后除以标准差。处理后的数据是零均值化的，用红色标记正值，蓝色标记负值，白色及浅色代表在均值附近。从颜色分布和上方的聚类索引都可以看出特征间相似程度、分布是否稀疏、是否对称。非稀疏的特征中，除了 **operate state 经营状况**和 **daily_count 日均交易笔数**外，其他特征的分布都较对称，可用性较高。由两个非对称特征推测，店铺中存在部分经营状态完全不活跃，存在部分低价走量的经营模式。

由于在构造特征时，使用都为一阶运算，所以相关系数 (Correlation coefficient) 可以较好地反映特征之间的信息耦合关系，如图 7，可以看出存在一定信息冗余的特征维度有约 4 6 个，那么用总特征数减去冗余维度，推测模型会在选择 6 8 个特征进行训练时达到最优。这一点也在网格搜索调参给出的结果中（参数 max_feature）得到了一定印证。

在模型方面，决策树属于白盒模型，其分支节点可以由代码绘制（因为图片过大不作展示），而决策树属于黑盒模型，如果用决策树分类模型 1 中同样参数的树来直接构成模型 2 随机森林，一定会过拟合。实则在使用过程中，由于使用了 bagging 的策略，随机森林模型在小数据集上的表现也容易过拟合。

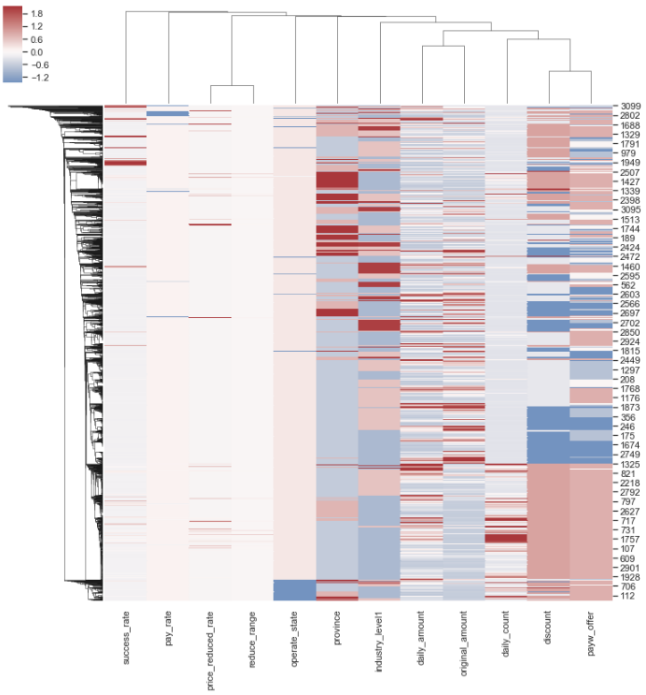


图 6: 12 个维度的特征 cluster-map 可视化

参考文献

[1] (美) 陈封能,(美)Steinbach,M.,(美)Kumar,V. 数据挖掘导论: 完整版 [M].2 版 (范明等译), 北京: 人民邮电出版社,2011.1:

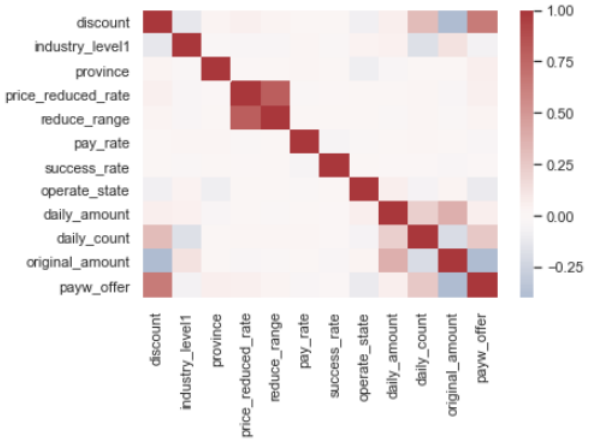


图 7: 12 个维度的特征相关性 heat-map 分析