计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(6):28−36 [doi: 10.15888/j.cnki.csa.007957]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

融合深度学习与集成学习的用户离网预测①

梁　晓[1], 洪　榛[2]

[1](中国电信股份有限公司 浙江分公司 企业信息化事业部, 杭州 310001)
[2](浙江工业大学 信息工程学院, 杭州 310023)
通讯作者: 梁　晓, E-mail: liangx.zj@chinatelecom.cn

摘　要: 随着国内通信市场逐渐饱和, 电信运营商之间的竞争日趋激烈. 用户流失预测已成为电信运营商最关注的问题之一. 本文提出一种基于多模型融合的方法创建用户离网预测模型. 首先, 将原始训练数据经过有放回采样和正负样本平衡得到多份不同的训练数据; 然后, 利用多份不同的训练数据使用集成学习与深度学习算法训练得到多个基础模型; 最终, 将多个基础模型进行融合形成高层模型. 实验结果表明, 融合模型在各类用户测试集上的表现均优于基础模型, 具有实际生产应用价值.

关键词: 用户离网预测; 深度学习; 集成学习; 融合模型

引用格式: 梁晓,洪榛.融合深度学习与集成学习的用户离网预测.计算机系统应用,2021,30(6):28–36. http://www.c-s-a.org.cn/1003-3254/7957.html

Churn Prediction Based on Fusion of Deep Learning and Ensemble Learning

LIANG Xiao[1], HONG Zhen[2]

[1](Enterprise Information Division, Zhejiang Branch, China Telecom, Hangzhou 310001, China)
[2](College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: As the China's communication market has been saturated over time, the competition among telecom operators is becoming increasingly fierce. Churn prediction of customers has turned into one of the most concerns for telecom operators. This study proposes a method based on multi-model fusion to create a churn prediction model of customers. First, through bootstrap sampling and positive-negative sample balancing, multiple training datasets are obtained from the original training data. Then, base models are trained by these datasets with ensemble learning and deep learning algorithms. Finally, the base models are merged into a high-level model. The experimental results prove that the fusion model performs better than all base models in the test datasets, with a practical value for production.

Key words: churn prediction of customers; deep learning; ensemble learning; fusion model

随着互联网与通信技术的快速发展, 国内通信市场已经趋于饱和, 通信运营商之间的竞争异常激烈. 用户离网已成为运营商重点关注的问题之一. 因此, 创建一个性能优异的用户离网预测模型预测用户离网, 及时发现具有较高离网概率的用户, 并制定有效的挽留策略, 这对通信运营商来说具有重要意义.

在众多分类算法中, 决策树算法效率高、简单易实现, 能够可视化决策规则, 业务解释性较强, 因此, 非常适合用户离网预测, 被广泛应用于各类用户离网预测场景中[1]. Logistic 回归、Bayesian 网络、人工神经网络、支持向量机等算法也被学者用于用户离网预测[2–6], 都获得了一定的成果, 创建了具有优秀预测性能的单一模型. 然而, 由于分类算法通常都具有不稳定性的问题, 在实际生产应用过程中, 训练数据集轻微的变化就

① 基金项目: 浙江省自然科学基金 (LY20F020030)
Foundation item: Natural Science Foundation of Zhejiang Province (LY20F020030)
收稿时间: 2020-10-09; 修改时间: 2020-11-16; 采用时间: 2020-11-24; csa 在线出版时间: 2021-06-01

能够造成模型性能的显著差异, 模型预测鲁棒性和泛化能力均较差, 不能满足实际生产应用的要求. 因此, 有学者将集成学习算法 (如: Random Forest、GBDT 等) 应用于用户离网预测, 在提升模型稳定性和预测准确率方面都取得了较大的进步[7,8]. 近年来, 各类新型梯度提升树算法层出不穷, 最具有代表性的为: XGBoost、LighGBM 和 CatBoost, 它们在各类机器学习竞赛中表现优异. 本文提出一种基于多分类器融合的方法创建用户离网预测模型, 该方法将应用 XGBoost、LighGBM、CatBoost、深度神经网络以及随机森林分别创建多个分类器, 并将多个分类器进行融合, 以利用多个分类器之间的互补性有效提升用户离网预测效果.

1 模型算法介绍

1.1 基于批规范化的深度神经网络 DNN-BN

深度学习通过建立具有层次结构的神经网络对输入信息进行逐层提取和筛选, 从而自动获得数据的特征表示, 最终实现端到端学习. 深度神经网络 (Deep Neural Network, DNN) 是由一组受限玻尔兹曼机构成的层次神经网络[9], 其网络结构如图 1 所示, 包括: Input 层、Hidden 层和 Output 层. Input 层负责接收样本数据, Output 层负责生成预测结果, 相邻层神经元之间采用全连接, 位于同一层中的神经元之间不存在连接. DNN 被广泛应用图像识别、语音识别等领域中, 并拥有良好的表现.

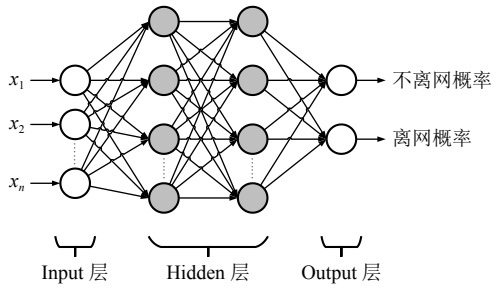


图 1 DNN 网络模型

批规范化 (BN) 本质上解决了深层网络难以训练的弊端[10]. 随着层次的增多, 信号的正向传播和梯度的反向计算会越来越大或越来越小, 导致梯度消失或梯度爆炸等问题. 为了解决上述问题, BN 将过小或过大的信号进行归一化. 即首先对输入进行白化预处理:

$$\hat{x}=\frac{x-E(x)}{\sqrt{Var(x)+\varepsilon}} \tag{1}$$

式 (1) 中, $E(x)$ 指其中一批输入 x 的平均值; Var(x) 为该批次数据的方差; ε是极小的正数, 为了确保分母不等于零. 对于深层网络, 在每个单元输出后都可加上一层 BN, 使得单元输出信号的每一维特征均值为 0, 标准差为1+ε, 但这样做会降低每个单元的表达能力. 为了提升模型的表达能力, 加入“比例和平移 (scale and shift)”操作, 即:

$$y=\alpha\hat{x}+\beta\equiv BN_{\alpha,\beta}(x) \tag{2}$$

式 (2) 中, 参数α,β随着网络中每层的迭代训练而得到更新学习, 当$\alpha=\sqrt{Var(x)+\varepsilon}$,$\beta=E(x)$时, BN 也就能够还原最初的输入, 如此可使 BN 层智能地更新参数, 在改变信号的同时也可以保持原输人, 不仅提升了模型的表达能力, 而且使信号在深层网络里更好地传递, 加速网络收敛. 在训练阶段, 每个批次数据的均值和方差都不同, 采用滑动平均的方式记录并更新均值和方差. 在测试阶段, 可直接调用最后一次修改的均值方差进行测试.

1.2 LightGBM 算法

Microsoft 在 2016 年开源 LightGBM, 它是基于决策树的梯度提升集成学习框架. 与基于决策树的传统集成学习方法相比, LightGBM 的训练速度更快、效率更高、内存使用率更低、模型效果更好、支持并行学习[11,12].

LightGBM 主要有以下特点:

(1) Histogram 优化

LightGBM 将连续型数值特征的每一个特征值划分到一系列离散的域中 (bins), 摒弃了传统的预排序思路, 如图 2 所示. 以浮点型特征为例, 一个区间的值会被作为一个桶, 然后用这些以桶为精度单位的直方图来做. 通过这样的方法, 有效简化了数据的表达, 而且降低了内存使用率. 此外, 直方图还带来了一定的正则化的效果, 可以平滑异常数据, 避免模型过拟合, 使其具有良好的泛化性.

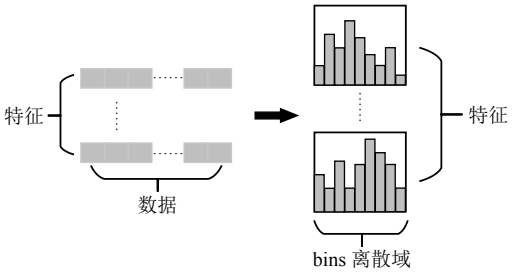


图 2 Histogram 优化流程

(2) Leaf-wise 生长方法

为了提升模型的预测效果, LightGBM 采用 Leaf-wise (按叶子) 生长方法, 如图 3 所示. 相比较于 XGBoost 中的 Level-wise (按层) 生长方法, Leaf-wise 生长方法效率更高. 使用 Leaf-wise 生长方法, 能够有效降低训练误差. 然而, 仅使用 Leaf-wise 生长方法很容易得到深度较大的树, 从而出现过拟合现象. 因此, 深度限制被添加到 Leaf-wise 生长方法中.

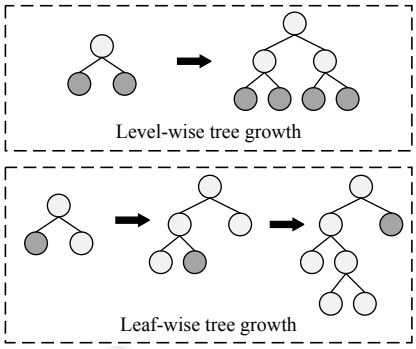


图 3 Leaf-wise 和 Level-wise 节点展开方式比较

(3) 类别型变量支持

传统的集成学习算法一般不能直接处理类别型变量, 需要先将类别型变量通过 One-Hot 编码等方式转化为数值型变量, 这种处理类别型变量的方法往往效率较低. LightGBM 通过在决策树算法上添加了对类别型变量的决策规则, 能够实现类别型变量的直接输入, 无需再对类别型变量进行转化.

1.3 CatBoost 算法

CatBoost 是由俄罗斯公司 Yandex 开发的一种新型梯度提升树算法, 于 2017 年 7 月宣布开源, CatBoost 算法的设计初衷是为了更好的处理类别特征[13]. 在传统 GBDT 中, 处理类别特征的方法是用类别特征对应标签的平均值来替换. 在决策树中, 标签平均值将作为节点分裂的标准, 此方法被称为 Greedy Target-Based Statistics (TBS), 用公式表示为:

$$\begin{cases} x_{i,k}=\dfrac{\sum\limits_{j=1}^{n}\left[x_{j,k}=x_{i,k}\right]\cdot Y_j}{\sum\limits_{j=1}^{n}\left[x_{j,k}=x_{i,k}\right]} \\ \left[x_{j,k}=x_{i,k}\right]=\begin{cases}1, & \text{if } x_{j,k}=x_{i,k}\\ 0, & \text{otherwise}\end{cases} \end{cases} \tag{3}$$

式 (3) 中, $x_{i,k}$ 表示第 k 个特征的第 i 类值, 分子表示第 k 个特征的第 i 类值对应的标签值的和, 分母表示第 k 个特征的第 i 类值的数量.

该方法有一个明显的缺陷, 即: 通常特征比标签包含更多的信息, 如果强行用标签平均值来表示特征的话, 当训练数据集和测试数据集数据结构和分布不一样的时候会出现条件偏移问题. CatBoost 改进 Greedy TBS 的方式是添加先验分布项, 从而有效减少噪声和低频率数据对于数据分布的影响[14]. 用公式表示为:

$$x_{i,k}=\frac{\sum\limits_{j=1}^{n}\left[x_{j,k}=x_{i,k}\right]\cdot Y_j+\alpha P}{\sum\limits_{j=1}^{n}\left[x_{j,k}=x_{i,k}\right]+\alpha} \tag{4}$$

式 (4) 中, P 是添加的先验项, α通常是大于 0 的权重系数, 其余参数与式 (3) 一致.

2 融合深度学习与集成学习的用户离网预测模型

本文采用一种融合多个模型的方法创建最终的离网用户预测模型, 方法过程如图 4 所示. 首先, 由原始训练数据经过有放回随机抽样和正负样本平衡后得到 5 份不同的训练数据, 并将每份训练数据随机切分成数量相等的两份 (如: 将训练数据集 i 切分为训练数据集 i_a 和训练数据集 i_b, 其中, i=1,2,3,4,5); 然后, 通过训练数据集 i_a 分别使用 CatBoost、LightGBM、XGBoost、随机森林、DNN-BN 训练得到不同的基础模型; 最后, 将训练数据集 i_b 分别输入基础模型, 得出输出结果, 并将该结果作为输入, 把训练数据集 i_b 的标签作为训练目标, 使用逻辑回归算法训练得到高层模型. 评估模型时, 将测试数据输入融合模型得到用户离网概率, 离网概率值大于 0.5 的用户判为离网用户, 反之, 离网概率值小于 0.5 的用户判为非离网用户.

DNN-BN 需要搭建如图 5 所示网络结构, 具体包括: 1 个 Input 层、3 个 Hidden 层以及 1 个 Output 层. DNN-BN 使用全连接层提取特征, 并结合 BN 层来帮助深层网络更好地传递信息. 本文使用 Keras 实现深度神经网络. 预测用户是否离网是典型的二分类问题, 因此, Loss 函数选用 Cross Entropy. Adam 是一种能自适应选择学习率的优化算法, 在计算学习率用于更新参数时, 综合考虑当前梯度和历史梯度, Adam 计

算效率高, 对内存需求少, 对超参数不敏感, 应用于大规模数据及参数的场景中能取得较好的效果, 因此本文选择 Adam 作为优化算法. 为避免神经元权重无法更新, 出现梯度为 0 的情况, 激活函数使用 LeakyReLU (alpha=0.05); 最终, 通过 Sigmoid 函数输出用户离网概率.

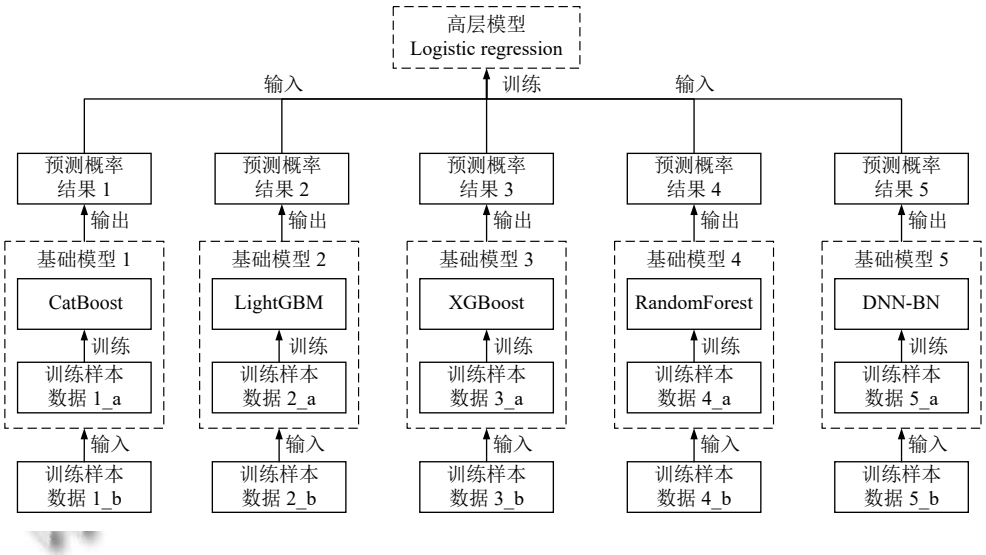


图 4 融合模型创建流程

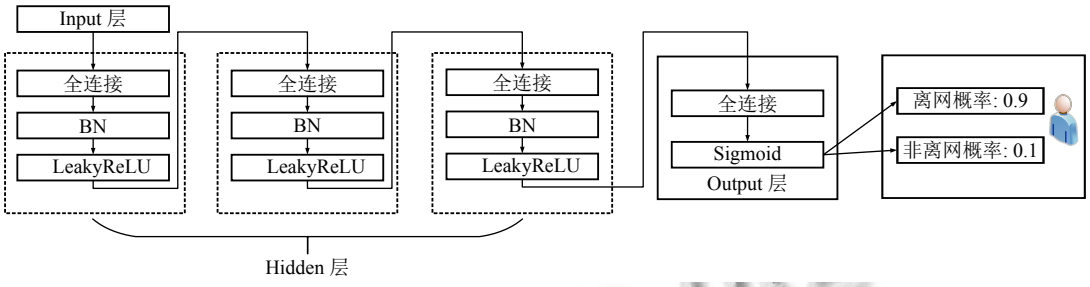


图 5 DNN-BN 网络结构图

3 模型实验与评估

本文选取浙江省某大型城市电信公司的手机用户为研究对象, 手机用户指该用户使用电信公司的手机业务而没有使用电信公司的其他业务, 如: 宽带、网络电视等. 由于用户数量较为庞大, 需要将研究目标聚焦在质量较高且相对较为活跃的公众市场用户上, 因此, 需要剔除行业客户、商业客户、校园客户等非公众市场客户和入网时长低于 6 个月的用户.

为了预测用户在未来一段时间内的离网倾向, 在定义数据的时间范围时, 需要在模型输入训练数据的时间和模型输出预测结果的时间之间增加一段间隔, 因此, 用户数据的时间范围包括: 观察分析期、维系期和预测期, 如图 6 所示. 观察分析期是指用户产生通信行为信息、消费信息等数据的时间范围, 即模型训练所需输入数据的时间窗口; 预测期是模型输出用户离网标识的时间; 维系期位于预测期与观察分析期之间, 当模型预测到某个用户在预测期有很大离网倾向时, 公司营销人员可以充分利用维系期去对潜在离网用户采取维系和挽留措施. 本文将训练集数据时间跨度界定为 2019 年 5 月至 2019 年 9 月, 其中, 观察期为: 2019 年 5 月至 2019 年 7 月, 该时间范围内, 用户数据如: 近 3 个月通话总次数、通话总时间、近 3 月新增积分等经过计算后作为建模需要的用户属性; 2019 年 9 月用户离网数据作为模型预测的目标数据, 即作为预测期数据输入模型. 另外, 本文将测试集数据时间跨度界定为 2019 年 7 月至 2019 年 11 月, 其中, 观察期为: 2019

年 7 月至 2019 年 9 月, 预测期为 2019 年 11 月.

对于用户离网的定义, 根据业务经验, 以下 3 类用户基本可以判定为离网, 分别为: 缓冲期和预测期内, 主动拆机的用户; 缓冲期和预测期内, 连续两个月出账金额为 0 元且通话时长为 0 秒的用户; 截止到预测期, 欠费双向停机超过 30 天的用户. 本文根据以上口径标准为样本数据打上“是否离网”标签, “离网”为 1, “非离网”为 0.

最终, 经过数据筛选和数据处理, 选取 2019 年 5 月浙江省某城市电信公司的 1812 311 户手机用户为训练样本用于训练模型; 选取 2019 年 7 月该市电信公司的 1841 950 户手机用户为测试样本, 用于评估模型.

另外, 根据业务经验, 用户的付费类型和套餐类型不同, 通信行为和消费习惯会存在较大差异, 因此, 本文将用户分为后付费畅享、后付费非畅享、预付费畅享、预付费非畅享 4 类, 并针对每类用户分别进行建模. 表 1 为 4 类用户的离网人数和离网率统计情况.

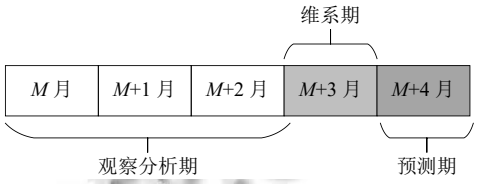


图 6 用户数据时间范围选择

表 1 各类用户的离网情况统计

类别	训练集			测试集		
	总人数	离网人数	离网率	总人数	离网人数	离网率
后付费畅享	170 740	1669	0.009 775	184 206	1885	0.010 233
后付费非畅享	240 953	2455	0.010 189	228 408	2648	0.011 593
预付费畅享	752 253	13 056	0.017 356	808 640	13 160	0.016 274
预付费非畅享	648 365	15 480	0.023 875	620 696	13 949	0.022 473
用户总数	1812 311	32 660	0.018 021	1841 950	31 642	0.017 179

3.1 特征选择

电信公司经过多年的数据积累已经获取了较为全面的用户特征信息, 这些特征信息可以归纳为 8 大类, 分别为: 用户基本信息, 如: 用户年龄、性别、等级、在网时长等; 用户消费信息, 如: ARPU 值、近三月 ARPU 均值等; 用户产品信息, 如: 套餐名称、套餐大类、套餐协议到期时间等; 用户服务信息, 如: 近三月投诉次数、投诉类型等; 用户通信行为信息, 如: 通话次数、通话时长、主被叫比例、上网流量等; 用户互联网应用信息, 如: APP 使用情况等; 用户社交圈信息, 如: 社交圈大小, 社交圈本网用户占比等; 用户终端信息, 如: 终端品牌、终端型号、终端价格等. 根据业务经验, 初步选取 8 大类特征信息中 134 个特征制作成用户宽表. 用户宽表制作完成后, 数据量依然较为庞大, 并且存在部分与用户离网相关性不大特征, 需要通过一定的方法将这些特征进行过滤, 保留与用户离网相关性较大的特征, 从而保证后续建模的效果.

(1) 基于 Pearson 相关系数的特征选择

使用 Pearson 相关系数可以衡量每个特征与标签变量的线性相关性, 其计算方法如下:

$$r(x,y)=\frac{Cov(x,y)}{\sigma_x\sigma_y} \tag{5}$$

式 (5) 中, x 和 y 表示两个特征变量, $Cov(x,y)$ 表示协方差, σ_x,σ_y表示标准差; $r(x,y)$ 表示 Pearson 相关系数, 其取值范围为 [−1,1], 其中, 0 表示线性不相关, 其值越接近 1 则是正线性相关性越大, 越接近−1 则是负线性相关性越大.

通过上述方式计算特征变量与离网标签之间的线性相关性, 对与目标变量不相关的特征或相关性弱的特征予以排除. 使用 scipy.stats 包中的 Pearson 或者 sklearn.feature_selection 包中的 f_regrssion, 均可以实现 Pearson 相关系数的计算. 本文计算得到每个特征与离网标签特征的相关性系数和 P 值后, 将与离网标签特征相关性小于 0.001 的特征进行过滤, 过滤特征的名称、相关性系数以及 P 值如表 2 所示.

表 2 基于 Pearson 相关性系数的过滤特征列表

字段名称	相关性系数	P值
性别	0.0001	0.889
本地流量使用	0.000 19	0.868
短信超出量	0.0005	0.623
套餐名称	0.0007	0.519

(2) 基于卡方检验的特征选择

卡方检验是一种基于卡方分布的假设检验方法[15]. 本文采用卡方检验来确定特征变量是否与离网标签目

标变量相关联, 基本假设为: H0 (特征变量与离网标签变量无关联); H1 (特征变量与离网标签变量有关联). 使用 sklearn.feature_selection 的 SelectKBest 和 chi2 可以实现卡方检验, 得到每个特征的卡方值和 P 值, 如果某特征的 P 值小于显著性水平或者其卡方值大于在显著性水平下的卡方值, 则拒绝 H0 假设, 即该特征与离网标签变量存在关联. 通过设定参数 k, 可以得到 k 个与标签值相关性最大的特征. 本文通过程序计算得到所有特征的卡方值和 P 值, 并将所有特征按照 P 值从大到小排列 (或者卡方值从小到大排列), 最后结合业务经验知识过滤一部分特征, 过滤特征信息如表 3 所示.

表 3 基于卡方检验的过滤特征列表

字段名称	卡方值	P值
其他流量包订购金额	0.00002	0.996
通话包订购金额	0.0008	0.977
增值包订购金额	0.0027	0.958
本月短信条数	0.0043	0.947
客户名下ITV数目	0.0051	0.9413
终端品牌	0.0073	0.9331
性别	0.0113	0.915
短信超出量	0.0117	0.913

3.2 数据转换

由于样本数据中绝大部分数值型特征的值均为非负值, 并且特征之间数值量纲差距较大, 故需要对这些数据进行 Min-Max 标准化, 将每个特征的数值缩放到同一尺度, 数值在 0 到 1 之间, 计算公式如下:

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{6}$$

式 (6) 中, x*表示标准化后的各个数据点取值, x 表示各个数据点的原始取值; $x_{\min}$ 和 $x_{\max}$ 分别表示每个特征数值系列中的最小值和最大值. Min-Max 标准化能够完整的保留原始数据之间的关系.

3.3 数据采样

本文使用有放回抽样和基于 K-means 聚类算法的分层抽样来得到多份训练样本, 用于训练多个基础模型, 其基本流程如图 7 所示.

有放回抽样借鉴了 Bagging 集成算法的思想, 抽样数据 D_t 的大小和原始训练数据集大小虽然一致, 但是理论上原始训练数据集中有 36.8% 的数据不会出现在 D_t 中, 这样就可以获得多个具有一定差异性的训练集, 从而保证了每个基础模型的差异性.

本文使用基于 K-means 聚类算法的分层抽样来平衡训练数据集中的正样本和负样本数量. 首先, 使用 K-means 算法分别将每个训练数据集 D_t 中的负样本进行聚类; 然后在每个类簇中分别抽取一定数量的负样本组成新的负样本集合; 最后, 将该负样本集合与正样本集合进行合并组成新的训练集合 $D_{t\text{-balanced}}$, 使得 $D_{t\text{-balanced}}$ 中的负样本数量与正样本数量的比例为 10:1. 这种抽样方法最大限度保证了负样本的特征多样性.

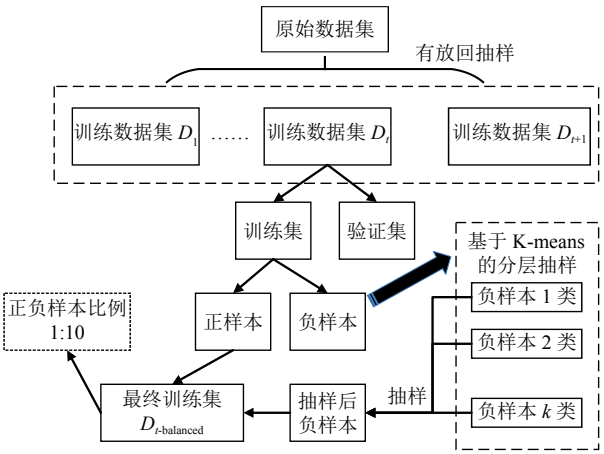


图 7 数据采样流程

3.4 模型训练

通过前述数据预处理工作, 四类用户数据集都可获得 5 份不同训练数据用于训练基础模型并融合成高层型. 根据第 2 章提出的模型训练方法, 针对每类用户数据分别训练一个融合模型. 其中, 基于批规范化的深度神经网络使用 TensorFlow2.1.0 版本中的 Keras 模块进行实现, Keras 支持快速构建神经网络模型, 并且代码同时支持在 CPU 和 GPU 上运行, 本文采用 keras.models 模块中的 Sequential 实现顺序模型构建深度神经网络, 全连接神经网络层采用 keras.layers 模块中的 Dense 实现, 批规范化层采用 keras.layers.normalization 模块中的 BatchNormalization 实现. CatBoost、LightGBM、XGBoost 均需要通过 pip 工具安装对应程序包, 随机森林算法则通过 sklearn.ensemble 模块中的 RandomForestClassifier 进行实现, CatBoost、LightGBM、XGBoost 的基础模型均为决策树, 因此 3 种算法都含有两类主要参数, 一类为控制树生长的超参数, 如: LightGBM 和 XGBoost 中的 max_depath, CatBoost 中的 depth、min_child_weight 等; 另一类为控制集成的超参数, 如: LightGBM 和 XGBoost 中的 n_estimators, CatBoost 中的 iteration 以及 learning_rate 等; 另外, CatBoost 和

LightGBM 均支持类别字段的直接输入, 将类别字段索引列表和类别字段名称列表分别赋给 CatBoost 中 cat_features 参数和 LightGBM 中的 categorical_feature 即可. 对基础模型的整合使用 mlxtend.classifier 中的 StackingClassifier 模块进行实现, StackingClassifier 中的 classifiers 参数用于设定基础模型列表, meta_classifier 参数用于设定高层模型, 训练高层模型的算法采用逻辑回归, 使用 sklearn.linear_mode 模块中的 LogisticRegression 进行实现, 主要通过设定参数 C 来控制正则化强度, 同时通过设定 class_weight 参数来进一步提升对数量较少的正样本判错惩罚. 算法的参数设定较为复杂, 需要使用 sklearn.model_selection 模块中的 GridSearchCV 实现带交叉验证的网格搜索来获取最佳参数, 参数最终设定如表 4 所示.

表 4 CatBoost、LightGBM、XGBoost、LogisticRegression 参数设定

CatBoost参数设定	LightGBM参数设定	XGBoost参数设定	LogisticRegression参数设定
eval_metric:F1,AUC depth:9 learning_rate:0.1 l2_leaf_reg:3 iteration:800 one_hot_max_size:2	objective:binary num_leaves:300 max_depath:7 learning_rate:0.05 n_estimators:600 metric:binary_logloss bagging_fraction:0.7 feature_fraction:0.7 bagging_frequency:7 bagging_seed:3	objective :binary:logistic eval_metric:logloss n_estimators:300 min_child_weight:6 max_depth:9 learning_rate:0.05 subsample:0.7 colsample_bytree:0.7 early_stopping_rounds:100	penalty:l2 dual:False tol:0.0001 C:0.9 fit_intercept:True intercept_scaling:1 class_weight:balanced solver:lbfgs max_iter:100

3.5 模型评估

本文使用精度、召回率、$F1$ 值、ROC 曲线以及 AUC 值对模型进行全面评估. 由于样本数据中, 正样本与负样本数量相差悬殊, 负样本数量远大于正样本, 因此, 本文重点关注正样本 (离网用户) 的精度、召回率和 $F1$ 值. 从表 5 中可知, 融合模型在 4 类用户数据集上均表现较好, $F1$ 值全部高于其他 5 类算法所创建的模型. 表 6 为融合模型综合指标.

表 5 融合模型与其他模型 $F1$、精度、召回率比较

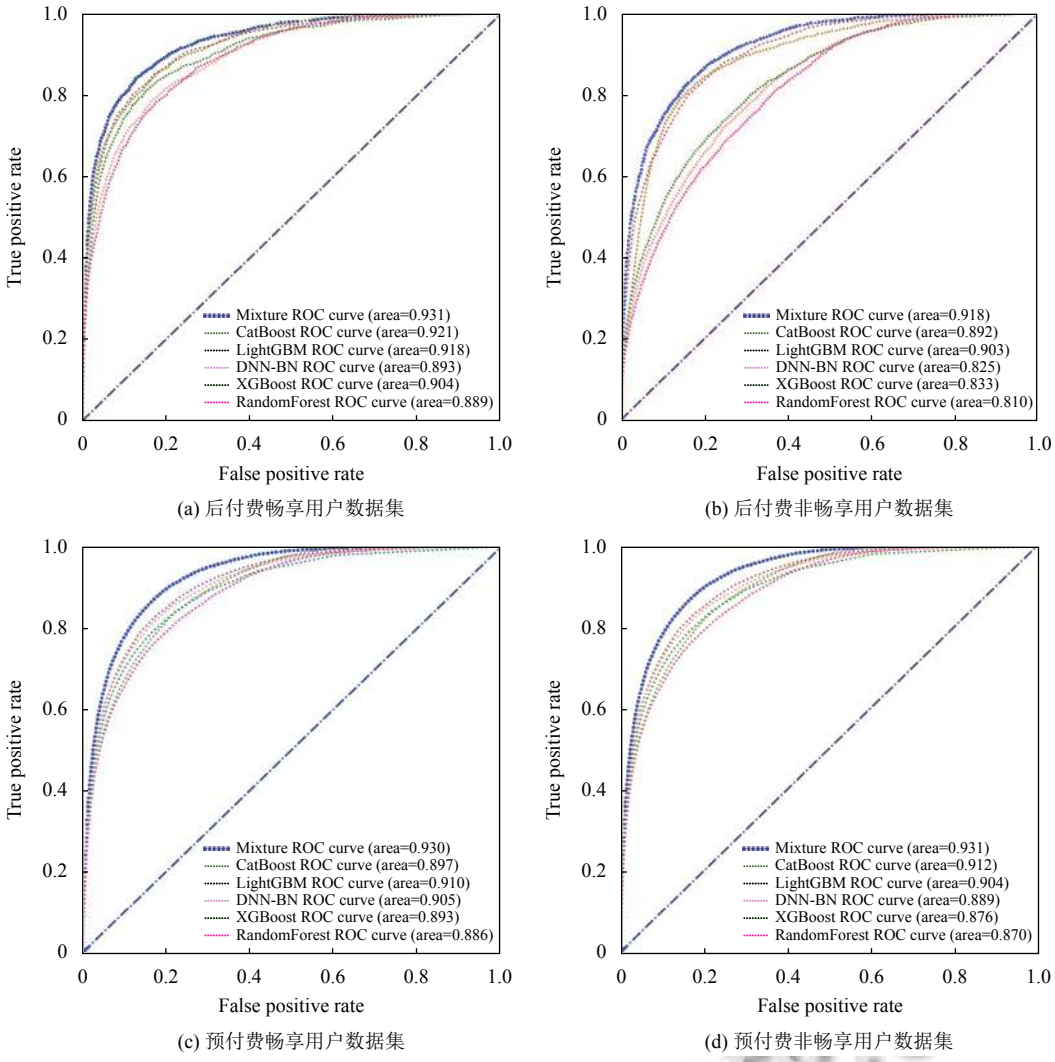
用户类型	CatBoost			LightGBM			XGBoost			Random Forest			DNN-BN			融合模型		
	$F1$	召回率	精度	$F1$	召回率	精度	$F1$	召回率	精度	$F1$	召回率	精度	$F1$	召回率	精度	$F1$	召回率	精度
后付费畅享	0.30	0.46	0.22	0.31	0.42	0.24	0.25	0.43	0.18	0.26	0.24	0.28	0.25	0.36	0.19	0.33	0.49	0.25
后付费非畅享	0.23	0.41	0.16	0.24	0.42	0.16	0.18	0.19	0.18	0.16	0.26	0.12	0.19	0.20	0.18	0.27	0.43	0.20
预付费畅享	0.31	0.35	0.28	0.32	0.34	0.29	0.29	0.41	0.21	0.28	0.32	0.25	0.31	0.34	0.29	0.35	0.42	0.30
预付费非畅享	0.35	0.27	0.49	0.33	0.26	0.45	0.28	0.26	0.31	0.28	0.24	0.32	0.35	0.28	0.46	0.42	0.45	0.39

表 6 融合模型综合指标

用户群	精确度	召回率	$F1$	预测离网人数	实际离网人数	正确预测离网人数
后付费畅享	0.25	0.48	0.33	3584	1885	899
后付费非畅享	0.20	0.43	0.27	5695	2648	1139
预付费畅享	0.30	0.42	0.35	18548	13160	5474
预付费非畅享	0.39	0.45	0.42	16101	13949	6249
综合指标	0.31	0.43	0.36	43928	31642	13761

从表 6 中可知, 融合模型在 1841950 个测试样本数据中找出了 43928 个疑似离网用户, 其中真实离网用户 13761 个, 精确度为 0.31; 实际离网用户总数为 31642, 模型找到的真实离网用户数占实际离网用户总数的 43%, 即模型召回率为 0.43, 模型综合 $F1$ 值为 0.36, 符合业务要求, 尤其是模型在占比最大的预付费用户数据集上表现较好.

为了更加直观地将融合模型的性能展现出来, 本文绘制与计算了融合模型与其他基础模型在四类用户数据集上的 ROC 曲线和 AUC 值, 如图 8 所示. 从图中的 ROC 曲线以及对应的 AUC 值可以看出, 融合模型的综合性能相比于其他 5 类模型有较为明显的优势, 并且在 4 类用户数据集上, AUC 值均稳定地保持在 0.9 以上.



(a) 后付费畅享用户数据集 (b) 后付费非畅享用户数据集

(c) 预付费畅享用户数据集 (d) 预付费非畅享用户数据集

图 8 融合模型与其他模型 ROC 曲线与 AUC 值的比较

4 结论与展望

本文针对电信用户离网预测问题, 提出一种基于多分类器融合的方法创建用户离网预测模型. 该方法使用深度学习算法和集成学习算法分别训练多个基础分类器, 并将这些基础分类器进行融合形成一个高层模型, 从而充分利用了多个分类器之间的互补性. 实验结果表明, 该方法创建的模型在各类用户数据集上的预测效果相比较于基础分类器均有一定程度的提升, 模型预测效果也符合业务要求, 因此, 具有实际生产应用价值.

本文所提出的方法虽然是应用于用户离网预测问题, 但是该方法中的思想, 可以用于解决运营商客户经营领域中遇到的各类问题. 近期, 循环神经网络在各领域中被广泛应用, 今后将应用循环神经网络建立具有时序特性的电信用户离网预测模型, 以捕捉用户从入网到离网的全生命周期时序特征, 从而提高模型预测效果.

参考文献

1 De Caigny A, Coussement K, De Bock KW. A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. European Journal of Operational Research, 2018, 269(2): 760–772. [doi: 10.1016/j.ejor.2018.02.009]

2 Su Q, Shao PJ, Ye QF. The analysis on the determinants of mobile VIP customer churn: A logistic regression approach. International Journal of Services Technology and Management, 2012, 18(1–2): 61–74.

3 Kisioglu P, Topcu YI. Applying Bayesian belief network

approach to customer churn analysis: A case study on the telecom industry of Turkey. Expert Systems with Applications, 2011, 38(6): 7151–7157. [doi: 10.1016/j.eswa.2010.12.045]

4 Verbraken T, Verbeke W, Baesens B. Profit optimizing customer churn prediction with Bayesian network classifiers. Intelligent Data Analysis, 2014, 18(1): 3–24. [doi: 10.3233/IDA-130625]

5 Sharma A, Panigrahi PK. A neural network based approach for predicting customer churn in cellular network services. International Journal of Computer Applications, 2011, 27(11): 26–31. [doi: 10.5120/3344-4605]

6 卢光跃, 王航龙, 李创创, 等. 基于改进的 K 近邻和支持向量机客户流失预测. 西安邮电大学学报, 2018, 23(2): 1–6.

7 Idris A, Rizwan M, Khan A. Churn prediction in telecom using random forest and PSO based data balancing in combination with various feature selection strategies. Computers & Electrical Engineering, 2012, 38(6): 1808–1819.

8 Ahmed AAQ, Maheswari D. An enhanced ensemble classifier for telecom churn prediction using cost based uplift modelling. International Journal of Information Technology, 2019, 11(2): 381–391. [doi: 10.1007/s41870-018-0248-3]

9 黄立威, 江碧涛, 吕守业, 等. 基于深度学习的推荐系统研究综述. 计算机学报, 2018, 41(7): 1619–1647. [doi: 10.11897/SP.J.1016.2018.01619]

10 Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on International Conference on Machine Learning. Lille, France. 2015. 448–456.

11 Chen TQ, Guestrin C. XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, CA, USA. 2016. 785–794.

12 Ke GL, Meng Q, Finley T, *et al*. LightGBM: A highly efficient gradient boosting decision tree. Proceedings of the 31st International Conference on Neural Information Processing Systems. Los Angeles, CA, USA. 2017. 3149–3157.

13 Dorogush AV, Ershov V, Gulin A. CatBoost: Gradient boosting with categorical features support. arXiv: 1810.11363v1, 2018.

14 Prokhorenkova L, Gusev G, Vorobev A, *et al*. CatBoost: Unbiased boosting with categorical features. Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal, QC, Canada. 2018. 6639–6649.

15 McHugh ML. The Chi-square test of independence. Biochemia Medica, 2013, 23(2): 143–149.