基于网络行为测量的云服务用户流失预测

摘　要

云服务为个体用户和企业级用户提供了大容量的虚拟存储空间和计算资源，能够以一种廉价的方式承载其计算方面的业务及服务。但近年来，随着公有云业务的广泛普及与互联网市场占比的日趋饱和，有部分用户在使用了一段时间云服务后不再继续使用，将这类客户定义为“流失用户”，对流失用户进行问询或挽留将有助于提高云服务质量，规避用户流失风险和提高云服务商发展潜力。

对于问题一，我们对比了基于 **XGBoost** 的自动特征工程和**层次分析法**模型，认为自动特征工程得到的特征重要性过于平均不利于筛选。而基于**网络行为测量理论**，将用户各项指标的统计测度例如偏差量、波动率等构造为评价准则，基于**层次分析法**构造了层次评价模型。所得到的最终结果均通过检验并获得每个特征的特征排序，最终结合 **XGBoost** 分类器性能筛选出 20 项指标。

对于问题二，基于**用户画像理论**和**网络行为测量理论**，将用户按流失预警次数分为三个风险等级，再在每个风险等级内按照计算、存储、网络通信三个层面进行用户分群。而分群的方法采用了 **KMeans** 聚类法，通过分析最优轮廓系数下的聚类结果为用户制定个性化标签。

对于问题三，基于问题一中筛选出的指标，将流失用户数据和非流失用户数据进行整合统计。由于样本相对普通数据非常稀疏这里选取 **XGBoost** 模型进行测试，并通过**遗传算法**进行参数调优。最终得到结果相对较好，F1 分数达到了 0.82，预测出 123 名有流失风险的用户。

对于问题四，从问题三的二分类出发进行改进再学习，抽象为一个多分类问题。基于流失用户的流失预警时间进行预测，对待预测数据进行测试得到流失时间预测，发现超过一半的有流失风险的用户都会在 6 个月以后流失，1 个月以内流失的高危用户仅占 2%。

模型结合了统计学与机器学习方法的优点，具有可靠性，并且结果具有良好的可解释性。从机器学习常用的衡量指标来看，模型的表现无疑是非常优秀的，能够高效率地进行客户流失行为的预测与背后特征原因探究，并生成最优策略，在实际商业应用中有一定价值。

关键词：网络行为测量，用户画像，XGBoost，KMeans，层次分析法，遗传算法

一、 问题重述

1.1 问题背景

公有云服务指为用户提供可通过互联网访问的虚拟服务器空间及其配套资源。在公有云中，所有硬件、软件和其他支持性基础结构均为云服务商拥有和管理，用户可以按需购买云服务器、数据存储和其他云相关服务并通过互联网访问这些服务器。由于公有云服务具有较好的弹性伸缩能力，因此微博的可伸缩业务如果利用公有云进行弹性部署，将完美解决突发热点事件导致流量激增的需求。目前，公有云通常用于提供网上办公应用，机器学习的训练、下载及存储，游戏开发和环境测试等。

近年来，随着公有云业务的广泛普及与互联网市场占比的日趋饱和，有部分用户在使用了一段时间云服务后不再继续使用，将这类客户定义为“流失用户”，对流失用户进行问询或挽留将有助于提高云服务质量，规避用户流失风险和提高云服务商发展潜力。

1.2 问题提出

我们团队需要解决如下问题：

（1）根据附件 1 中的流失用户监控指标的监控值，建立筛选指标模型，选出与用户流失相关的重要指标，请说明选取的指标数量以及原因。

（2）根据附件 1 和附件 3 中的用户资源利用情况，建模刻画用户画像，对用户的流失风险进行分级，给出每一流失风险等级用户特征的数学描述。

（3）基于问题（1）筛选出的重要监控指标，根据附件 1 与附件 3 中的用户监控指标的监控值，构建用户流失预测模型，说明流失用户的具体判别标准及流失用户的监控指标的长期变化趋势特征。

（4）根据附件 4 中用户监控指标的监控值，结合问题（2）构建的模型，预测用户的最终流失时间点所在范围。

二、 问题分析

2.1 问题一的分析

对于问题一，这是一个特征筛选问题。但数据所包含的特征虽然数量多但是平均一个用户所观测的指标却并不多，故造成了面板数据大量缺失的现象。属于典型的高维稀疏数据。而对于高维度稀疏数据的特征选择，我们采用对比自动特征工程和多因素评价类模型两种方法。而在多因素评价类模型当中，我们选择使用层次分析法进行综合评价。

2.2 问题二的分析

问题二需要给流失用户进行等级分级，再为每一级用户进行画像。用户画像（persona）的概念最早由交互设计之父 Alan Cooper 提出：

“Personas are a concrete representation of target users.” 是指真实用户的虚拟代表，是建立在一系列属性数据之上的目标用户模型。随着互联网的发展，现在我们说的用户画像又包含了新的内涵——通常用户画像是根据用户人口学特征、网络浏览内容、网络社交活动和消费行为等信息而抽象出的一个标签化的用户模型。构建用户画像的

核心工作，主要是利用存储在服务器上的海量日志和数据库里的大量数据进行分析和挖掘，给用户贴“标签”，而“标签”是能表示用户某一维度特征的标识。具体的标签形式可以参考下图某网站给其中一个用户打的标签。

我们通过多种对比分析，确定建立自顶而下的层级标注模型对用户进行标签画像。首先，按照流失用户的预警次数，将预警次数为 1 次的用户标注为低风险用户，预警次数为 2 或 3 的用户为中风险用户，高于 4 次的用户为高风险用户进行等级分类。然后，根据用户不同的观测指标，尽管观测指标共计高达 132 项，每个用户的观测指标却只有其中几项，我们仍然可以将其按照计算、存储、网络通信三大类进行区分，对不同方面的网络行为进行测量。最后，在同一风险等级下同一方面的用户网络行为，可以通过 KMeans 聚类为用户分群，通过聚类分群的方式为用户实现标注。大体流程如图 1 所示：

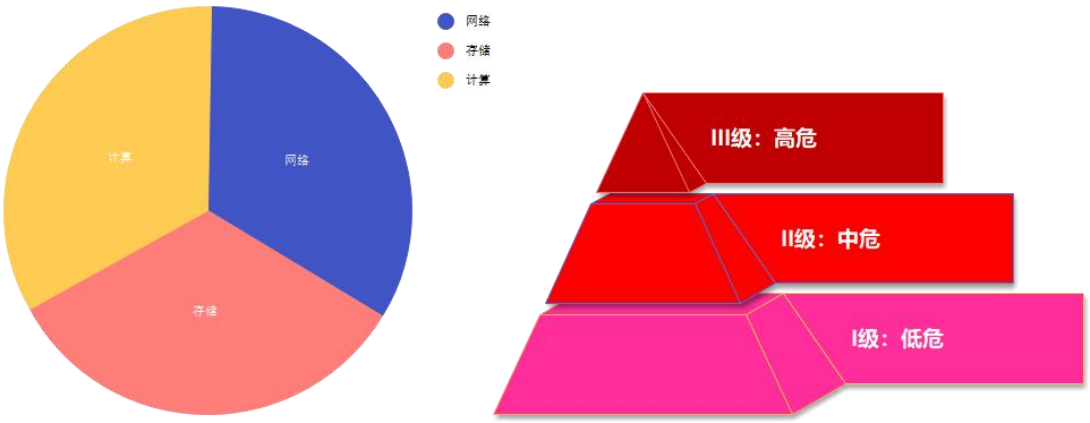

图 1　用户画像的标准

用户在某一方面的网络行为需要构造特征用于聚类。例如，对于用户的计算型网络行为而言，为了消除不同计算型特征的数值水平影响和量纲影响，采取与问题一类似的策略使用波动率、平均变化率等四项指标对同一用户的整体网络行为进行整体测量。而为了探索每一层的最优 K 值以及评价聚类效果的好坏，可以采取轮廓系数法。轮廓系数法结合了聚类的凝聚度(Cohesion)和分离度(Separation)，用于评估聚类的效果。指标:内部距离最小化，外部距离最大化。平均轮廓系数的取值范围为[-1,1]，系数越大，聚类效果越好。每次聚类后，每个样本都会得到一个轮廓系数，当它为 1 时，说明这个点与周围簇距离较远，结果非常好，当它为 0，说明这个点可能处在两个簇的边界上，当值为负时，暗含该点可能被误分了。

2.3 问题三的分析

将流失用户与非流失用户进行汇总，根据我们在问题一中筛选的 20 项指标对数据进行整合，根据每个用户在该指标中体现的面板数据的统计特性构造分类器进行分类。由于数据的缺失量比较高，属于典型的缺失数据，我们选择使用集成学习方法中的 XGBoost 对问题进行建模。另外，为了与 XGBoost 形成对比，我们也对比了常见的随机森林等算法。

2.4 问题四的分析

问题四的求解采用二阶段分类策略。首先，基于问题三中对用户网络行为测量将

用户进行流失用户和非流失用户的二分类，这一过程可以作为流失时间预测的baseline，即初步判定哪些用户会流失再进行流失时间的预测会更准确。

三、 模型假设

针对这些问题，我们的模型假设如下：

1.观察到购买和非购买人群比例悬殊较大，我们假设类别的失衡不会造成严重影响，或者说造成的影响在我们模型的误差范围之内。

2.假设 a1-a8 内超过 100 的数值是由于小数点标定错误导致的，且部分客户没有子女的情况。

3.实施精准营销策略时认为并非提升越多效果越好，只要提升量超过一定阈值使客户产生购买行为即为成功，且暂不考虑收益与营销成本的量化关系。

四、 符号说明

符号	说明
$\overline{x}$	平均值
S^2	方差
n	数值个数
μ	数学期望
t	t 检验统计量
χ^2	卡方统计量
A	离散数据的实际列联表
T	卡方公式中指理想列联表，XGBoost 中指树深度
$p(x)$	x 的概率
$H(X)$	X 的信息熵
$H(X\|Y)$	按照 Y 划分的条件信息熵
$ID(X,Y)$	信息增益
$L(\Phi)$	XGBoost 的损失函数
$f_t(x)$	学习的基学习器
$\Omega(f_t(x))$	正则化项，与基学习器的深度与权值范数有关
λ	常数项
w	XGBoost 中的各项权值系数

五、 模型的建立与求解

5.1 问题一模型的建立与求解

5.1.1 模型的建立

我们首先对比了自动特征工程。为实现特征工程的自动化并能够使模型适应大批量的高维稀疏数据，选择 xgboost 算法进行测试。特征工程分为数据预处理、特征抽取、特征构造和特征选择等几个步骤[3]，使用机器学习算法自动进行特征筛选的一类重要模型是树模型，这里选用 XGBoost 算法进行测试。XGBoost 算法是一类基于树结构的算法，而树结构为每个特征分配重要性分数的核心在于信息增益的计算。信息增益是指在按照某一条件划分以后的信息熵相较划分前信息熵的增量，能够衡量划分的效果，此外，信息增益率和基尼指数也可以衡量划分能力[4]。

$$IG(X,Y)=H(Y)-H(Y|X)=\sum_{x,y}p(x)\cdot p(y|x)\cdot \log p(y|x)-\sum_{y}p(y)\cdot \log p(y) \quad (1)$$

使用信息熵我们就能够计算出，根据哪一特征划分信息增益最大，然后按照信息增益从大到小排序即可得到最重要的特征。

决策树是一类基于树结构的监督学习算法，可以进行回归也可以用来分类。在决策树的算法中引入了信息论的方法，用熵来衡量非叶节点的信息量的大小，决策树中的非叶节点表示属性，叶子节点表示样本实例所属类别[6]。通过输入一组带有类别标记的数据，输出一棵二叉或多叉的树，优点是能够直观展示其结构。

随机森林(RF)是一种统计学习理论,它是利用 bootsrap 采样方法从原始样本中抽取多个样本, 对每个样本进行决策树建模, 然后综合多棵决策树的结果,通过投票得出最终预测结果，具有很高的预测准确率,对异常值和噪声具有很好的容忍度,且不容易出现过拟合[7]。

由 Chen 等人提出的 XGBoost 是对梯度提升树框架(Gradient Boost Decision Tree Framework)的一种具体实现，以 CART 决策树作为基学习器，既可以用于分类问题也可以用于回归问题[8]。XGBoost 的基本形式为

$$\begin{cases} L(\phi)=\sum_{i} l(\hat{y}_i,y_i)+\sum_{k}\Omega(f_k) \\ \Omega(f)=\gamma T+\frac{1}{2}\lambda\ ||\omega||^2 \end{cases} \quad (2)$$

在迭代过程中的损失函数表达式如下

$$L^{(t)}=\sum_{i=1}^{n} l\left(y_i,\hat{y}_i^{(t-1)}+f_t(x_i)\right)+\Omega(f_t)\simeq\sum_{i=1}^{n}\left[l\left(y_i,\hat{y}^{(t-1)}\right)+g_i f_t(x_i)+\frac{1}{2}h_i f_t^2(x_i)\right]+\Omega(f_t) \quad (3)$$

不同于普通的 GBDT，XGBoost 在 L 后加入了与决策树深度 T 和分类器分数 w 有关的正则化项，使得精度本就出众的模型更具有鲁棒性。此外从某种意义上讲，它本身还具有降维的效果，并且对类别失衡的数据效果较好。

所筛选特征的基本要求是需要能够被附件 1、附件 3 和附件 4 同时包含。经过测试，我们将特征的重要性分数条形图绘制如图 2 所示。从图 2 中我们可以发现，每个特征的重要性分数都比较平均，区分度不够，难以对比特征的重要性程度。故我们认

为自动化特征工程模型并不适用于这一问题。下面我们使用多因素评价类模型当中常用的主成分分析法进行测试。

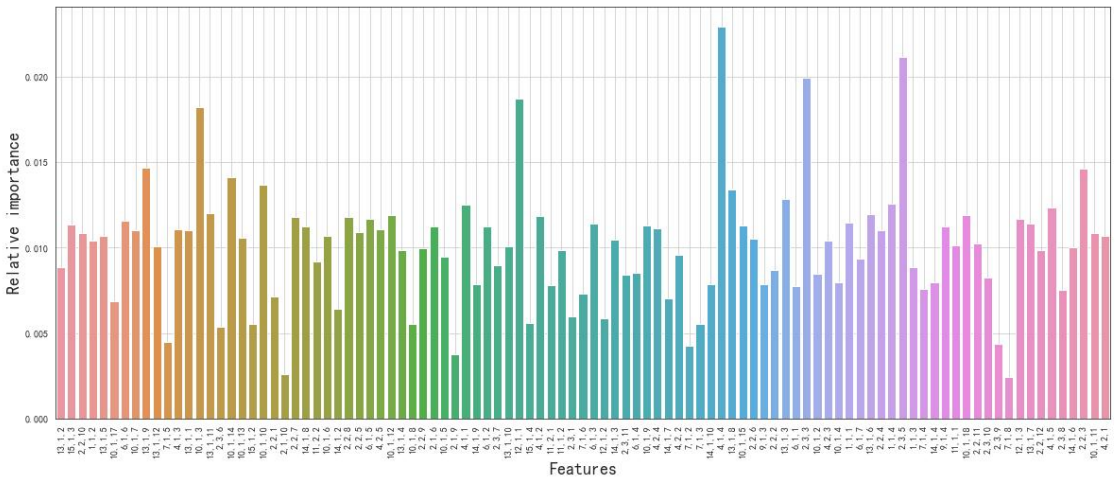

图 2　特征重要性分数

5.1.2 模型的求解

在进行主成分分析之前，我们需要合理根据观测特征的统计特性构造指标。注意到同一用户的某一指标呈现出时序性、面板性特征，将用户对应的预警时间在原始数据中进行标记后，我们基于序列的统计特性将评价指标分为三大板块：静态特征、动态特征和关联特征。

对于静态特征，我们主要用某一指标包含的数据量和用户量来进行评价。对于某一数据，有记录的数据量越多模型越能够学到更多消息，所以数据量指标对筛选过程有重要作用；而某一个特征所覆盖的用户越多，说明数据丰富度更高，能够对样本的特殊性达成一定约束。

对于动态特征，我们将其分解为四项小指标如下：

第一是偏差量。我们将其定义为最大值与最小值差值的平均值与平均水平的平均值之比。以最大值减最小值的目的是为了描述用户在该特征上的网络行为稳定性。如果在某一天内的特征观测量偏差较大说明该用户在当日网络行为不稳定，使用量在一日之内有较大变化。但为了消除单位和基准水平的影响，选择以平均水平的平均值作为除数。

$$\Delta = \frac{E(x_{\max} - x_{\min})}{E(x)} \tag{1}$$

第二是波动率。波动率我们以该特征观测值序列的标准差除以平均值来描述。这一指标的意义在于描述特征是否构成足够的信息丰富度，若没有足够的标准差则这一特征对用户是否流失的预测不能提供足够的信息。另外，若特征没有足够高的波动率也不容易观测到在何种水平下能够被预判为是否流失。但同样为了消除单位和基准水平的影响，以平均水平的平均值作为除数。

$$e = \frac{\sqrt{D(x)}}{E(x)} \tag{2}$$

第三是平均变化率。该指标定义为某一用户当日与下一日特征观测值之差的绝对值与当日观测值的平均值，再将所有用户进行平均。这一特征的意义在于描述特征的变化量是否容易发生突变，容易发生突变的特征其观测值更容易被注意到并作为判定

是否流失的标准。

$$r = E(r_i)$$
$$r_i = \underset{t}{E}\left(\frac{|x_{i,t+1} - x_{i,t}|}{x_{i,t}}\right) \tag{3}$$

第四是平均最大变化率。该指标定义为某一用户当日与下一日特征观测值之差的绝对值与当日观测值的最大值，再将所有用户进行平均。这一特征的意义与上面类似，但不同之处在于，平均变化率描述的是指标与用户的平均水平，平均最大变化率考察的是最大水平，削弱了时间周期长带来的影响，更关注特异值。

$$r = E(rm_i)$$
$$rm_i = \max_{t}\left(\frac{|x_{i,t+1} - x_{i,t}|}{x_{i,t}}\right) \tag{4}$$

对于关联特征，我们将其分解为关联系数和 T 检验概率两个方面。我们将用户是否流失记为 0/1，在对应的日期和用户进行标记以后可以计算出特征与标记之间的皮尔逊关联系数：

$$\rho = \frac{Cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{5}$$

将标记为流失的样本和没有标记为流失的样本分为两个样本可以进行双样本 T 检验。t 检验也用来判断样本均值和总体均值的显著性差异。很多地方 t 检验和 Z 检验类似，但是最大的区别在于总体的理论方差是未知的，t 分布只能用样本数据估计。独立样本 t 检验分析定类数据与定量数据之间的差异，配对样本 t 检验用来揭示定量数据的对比关系，样本先后的顺序要一一对应[1]。配对 t 检验的统计量定义为：

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \tag{6}$$

t 检验的原假设为统计量在两样本中无显著差异，备择假设为两样本中这一统计量有显著性差异。通常若概率小于 0.05，则接受备择假设，显著度 0.05。选择以 t 检验概率作为指标的意义在于若特征在两类样本中区分度不明显则难以用于预测是否流失。

指标之间的层级关系如图 3 所示。

图 3　指标构建的层次体系

运用层次分析法建模，大体上可按下面四个步骤进行：

①建立递阶层次结构模型；

②构造出各层次中的所有判断矩阵；

③层次单排序及一致性检验；

④层次总排序及一致性检验。

应用 AHP 分析决策问题时，首先要把问题条理化、层次化，构造出一个有层次的结构模型。在这个模型下，复杂问题被分解为元素的组成部分。这些元素又按其属性及关系形成若干层次。上一层次的元素作为准则对下一层次有关元素起支配作用。这些层次可以分为三类：

（1）最高层：这一层次中只有一个元素，一般它是分析问题的预定目标或理想结果，因此也称为目标层。

（2）中间层：这一层次中包含了为实现目标所涉及的中间环节，它可以由若干个层次组成，包括所需考虑的准则、子准则，因此也称为准则层。

（3）最底层：这一层次包括了为实现目标可供选择的各种措施、决策方案等， 因此也称为措施层或方案层。

对于静态特征而言，我们认为用户数相比数据量更重要，能够包容更丰富的数据分布，减弱有偏性。因此它的成对比较矩阵为：

$$A_1=\begin{pmatrix}1 & \frac{1}{3}\\ 3 & 1\end{pmatrix} \tag{7}$$

对于动态特征而言，我们认为最重要的是平均变化率和平均最大变化率，而波动率作为数据最基本的统计特性其重要性是最弱的。

$$A_2=\begin{pmatrix}1 & \frac{1}{5} & 1 & \frac{1}{5}\\ 5 & 1 & 3 & 3\\ 1 & \frac{1}{3} & 1 & \frac{1}{3}\\ 5 & \frac{1}{3} & 3 & 1\end{pmatrix} \tag{8}$$

对于关联特性而言，由于数据中流失样本也就是 1 样本不多所以关联系数基本都不会太大，真正更有影响的是 T 检验的概率值。因此它的成对比较矩阵为：

$$A_3=\begin{pmatrix}1 & \frac{1}{5}\\ 5 & 1\end{pmatrix} \tag{9}$$

上层的静态特征、动态特征和关联特征的综合比较矩阵，我们认为，动态变化的特征相比于有偏数据下统计得到的关联特征更能反映特征的变化情况，而数据的静态特征只是最基本的参考。因此，我们得到顶层的成对比较矩阵为：

$$A_4=\begin{pmatrix}1 & \frac{1}{3} & \frac{1}{3}\\ 3 & 1 & 3\\ 3 & \frac{1}{3} & 1\end{pmatrix} \tag{10}$$

接下来，我们将基于层次分析法求解每一层的权重

表 1　权重分解的层次图

顶层设计	一级变量	顶层权重	二级变量	二级权重	最终权重
评价体系	静态特性	0.135	数据量	0.25	0.03375
			用户量	0.75	0.10125
	动态特性	0.584	偏差量	0.102	0.059568
			平均最大变化率	0.479	0.279736
			波动率	0.112	0.065408
			平均变化率	0.308	0.179872
	关联特性	0.287	相关系数	0.167	0.047929
			T 检验概率	0.833	0.239071

获得权重后我们对每个特征进行评分排序。对于 T 检验概率，由于其他的量都越大越好而概率是越小越好，我们以 0.05 的显著度为标准，用 0.05 减去原始概率作为变换后特征，这样就完成了数据的正向化。为使数据归一化，我们对以上的八个二级指标进行 min-max 规约化：

$$x=\frac{x-\min(x)}{\max(x)-\min(x)} \tag{14}$$

这一操作的目的是为了消除八项指标水平偏差太大造成的影响，将所有数据的范围全部规约到 0-1 之间。例如，偏差量的取值可能为几百甚至几千，但 T 检验概率无论如何不会超过 1，这一操作就可以削弱这一影响。

经计算，我们可以获得 132 项特征的评分排序。而为了筛选出最有用的前若干项特征，我们应用 XGBoost 分类器进行逐一训练。在流失预警记录中仅有 472 项数据，但每个用户每天的指标记录存在 79086 条，属于典型的样本失衡问题。对于样本失衡问题，一个重要的指标就是失衡样本的 F1 分数。因为对于流失用户的预测而言，其精准率和召回率往往是一对相冲突的指标，精准率过高的同时召回率就难以取得较高水平，故用它的 F1 分数作为评判指标。我们的方案是按照特征得分的降序排序逐一向模型内添加特征，当继续添加特征模型的 F1 分数没有得到显著提升时证明选择完毕。

经调参并测试，模型 F1 分数随着特征数量的变化曲线如图 4 所示：

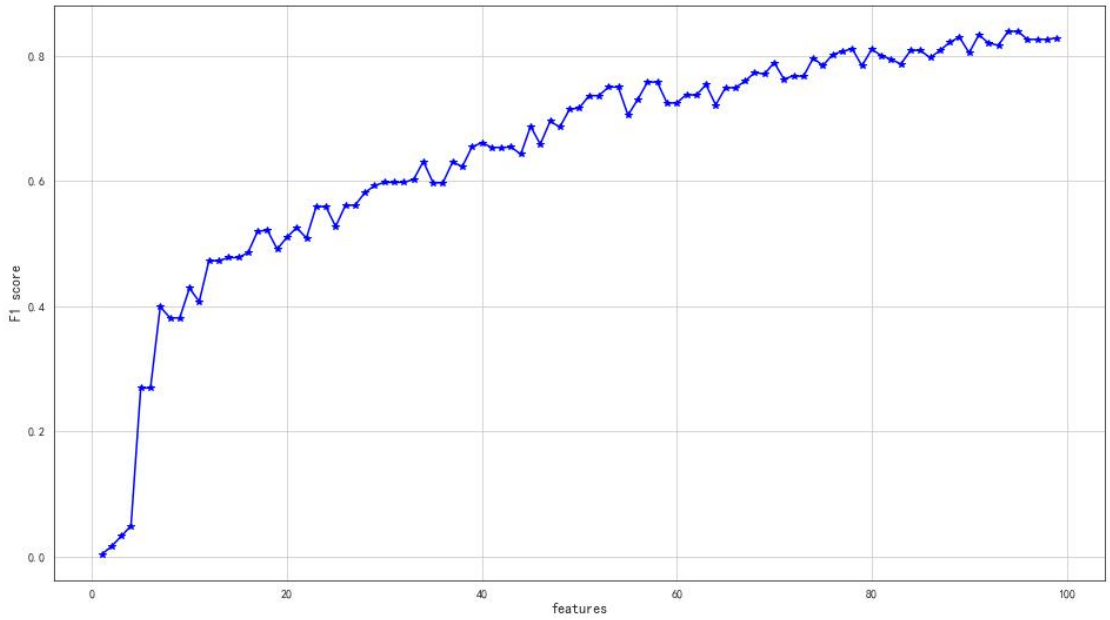

图 4　F1 分数随特征数量变化曲线

从图 4 中可以看出，曲线的拐点大约在 20 个特征。故我们筛选排名前 20 的指标。指标排名情况见附录。

5.2 问题二模型的建立与求解

5.2.1 模型的建立

经过测试，我们分别对无风险、低风险、中风险和高风险用户的轮廓系数随 K 变化的关系进行了探究，并将曲线图绘制如图 5 所示：

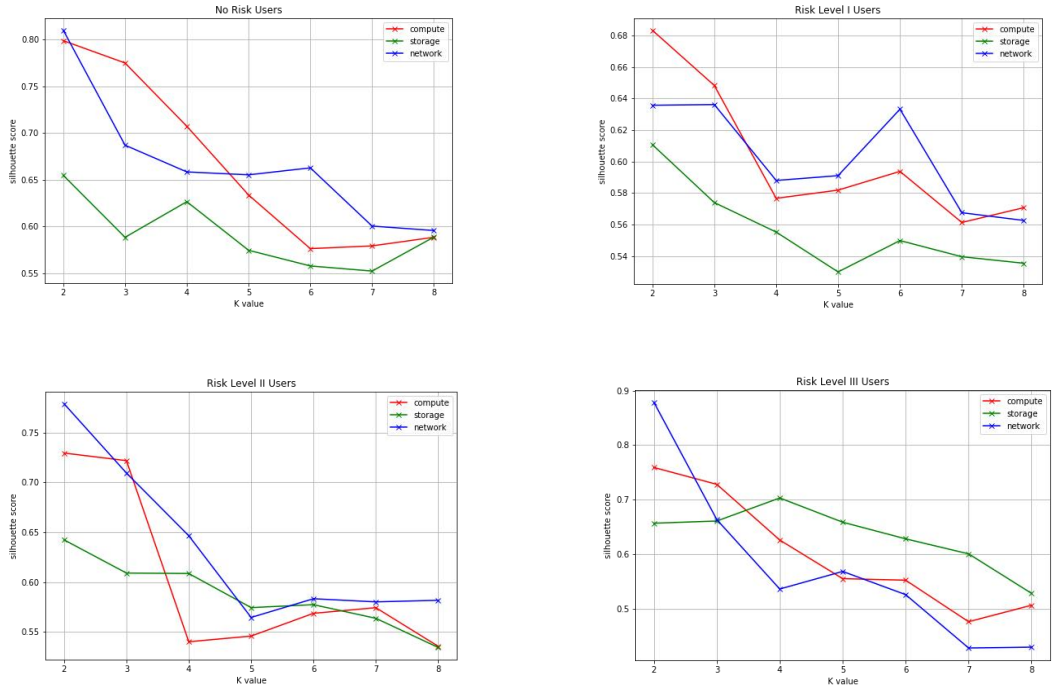

图 5　聚类的轮廓系数与 K 值变化曲线

5.2.2 模型的求解

对于低风险用户，从计算、网络、存储三个层面来看都应该划分为两大类比较合适。从计算层面来看，两类用户的明显差异主要表现在当日指标极大值与极小值的极差与平均水平的比值。一群用户的计算特征极大值与极小值的偏差幅度是较大的，而另一群用户的每日监测指标极差较小，更为稳定，故可以区分为“计算稳定型用户”和“计算不稳定用户”。这类用户的区别主要是由于计算算力的使用需求和使用用途决定的，对于计算稳定型用户，他们对算力的需求更多集中在为前后端提供更稳定的服务。而从存储层面来看，两类人群的表现主要集中表现在存储水平的区分，分为“高存储量用户”和“低存储量用户”。两类用户的主要区别在于对数据存储量的需求，高存储量用户对云服务的需求更多的是存储海量数据，例如爬虫数据、大型的深度学习开源数据集等。从网络层面来看，K 取值为 2 时最优，而两类人群的表现差异同样主要集中在偏差量。类比计算层面的画像，网络通信层面的画像同样可以区分为“网络稳定型用户”和“网络不稳定用户”，两类用户的区别很可能是由于自身网络流量限制和对通信需求高低造成的。

对于中风险用户，从计算、网络、存储三个层面来看同样应该划分为两大类比较合适。从计算层面来看，两类用户的明显差异主要表现在指标在一段时间的平均变化率，一群用户的计算特征变化较快而另一类变化较慢，可以区分为“计算变化快用户”和“计算变化慢用户”，这两类用户的区别在于计算量的稳定性需求不同，计算变化快用户并不需要长期稳定的计算力服务。而从存储层面来看，两类用户的主要差异表现在存储的偏差量上，可以按照偏差量大小分为“存储稳定型用户”和“存储不稳定用户”，而二者主要的差异在于存储数据的临时性与交换，频繁的数据读写和修改容易造成存量和存储内容的不稳定。少量存储不稳定用户也是计算变化快的用户，说明此类用户对算存一体的需求较大，计算算力和存储存量之间的频繁读写交换造成此类用户的不稳定。从网络层面来看，两类用户之间的主要差别同样是偏差量，区分为“网络稳定性用户”和“网络不稳定用户”，两类用户的区别很可能是由于自身网络流量限制和对通信需求高低造成的。

对于高风险用户，标签的划分则有所不同。计算层面的用户划分同样是划分为两大类，按照偏差量分为“计算稳定型用户”和“计算不稳定用户”。但存储层面需要划分为四个用户群才是最优。经分析，这四类用户群在存储变化率、和存储最大变化率二者上表现存在较为明显的差异，可以区分为“存储量稳定用户”、“存储稳步变化用户”、“存储强波动用户”和“存储快波动用户”四大类，造成这四大类用户区别的原因在于存储量需求水平不同，对不同时间的存量变化趋势不同，可能是稳步上涨或下降，也可能是其他时间较为平稳但某几日获得存量突变，还可能是存量在一段时间内表现极不稳定。网络层面上按照网络特征的平均变化率划分为“网络变化快用户”和“网络变化慢用户”，这是由于不同用户对通信能力的需求随着时间变化形成的。但从高风险用户的画像特征来看，存储行为的指标变化与波动能够最有效判定用户流失风险。

5.3 问题三模型的建立与求解

5.3.1 模型的建立

随机森林是 Breiman[1] 提出的基于树的集成学习算法, 根据特征数对每个样本选取分裂指标进而构建单棵子树去完成分类或回归任务。随机森林通过集成多性能较弱的多个基学习器来构建一个强学习器, 各个基学习器之间相互互补, 降低了方差以及过拟合的风险, 从而提高模型的性能。由于随机森林是基于 Bagging 策略[2]的的算法，利用 bootsrap 采样方法从原始样本中抽取多个样本, 对每个样本进行决策树建模, 然后综合多棵决策树的结果,通过投票得出最终预测结果，具有很高的预测准确率,对异常值和噪声具有很好的容忍度,其基学习器之间并无直接关联，于是可以很方便地进行并行化计算，且不容易出现过拟合[3]。

5.3.2 模型的求解

由于随机森林基于 Bagging 策略，其基学习器只会抽样式从服从同一分布的随机向量中对数据进行采样并学习基学习器，最终应用投票法将多个基学习器的结果投票进行输出。其学习算法如下：

Step 1. 采用 Bootstrap 策略从原始数据集中采样生成 N 个子数据集，并在每个子数据集上训练 CART 决策树[4]。

Step 2. 随机森林由 i 棵分类树构成, 每棵分类树的子节点在进行分裂时随机选择分裂指标数, 根据衡量指标大小选择最优分割指标进行划分。

Step 3. 重复第二步，使每棵子树都构建完成。

Step 4. 将生成子树进行投票获得最终学习器：

$$P_{rf}(X)=arg\max_{Y}\sum_{i=1}^{n}I(w(X,\theta_i)=Y)NY_c \tag{15}$$

随机森林的决策结果取决于每一棵子树的训练结果, 分裂指标的选取决定了分裂标准, 随机森林一般采用基尼指数 (Gini)[5], 其大小衡量了各节点混乱程度, 其计算方法如下:

$$Gini(c)=1-\sum_{y=1}^{u}p(y|c)^2 \tag{16}$$

我们将模型的思想流程图绘制在图 6 中：

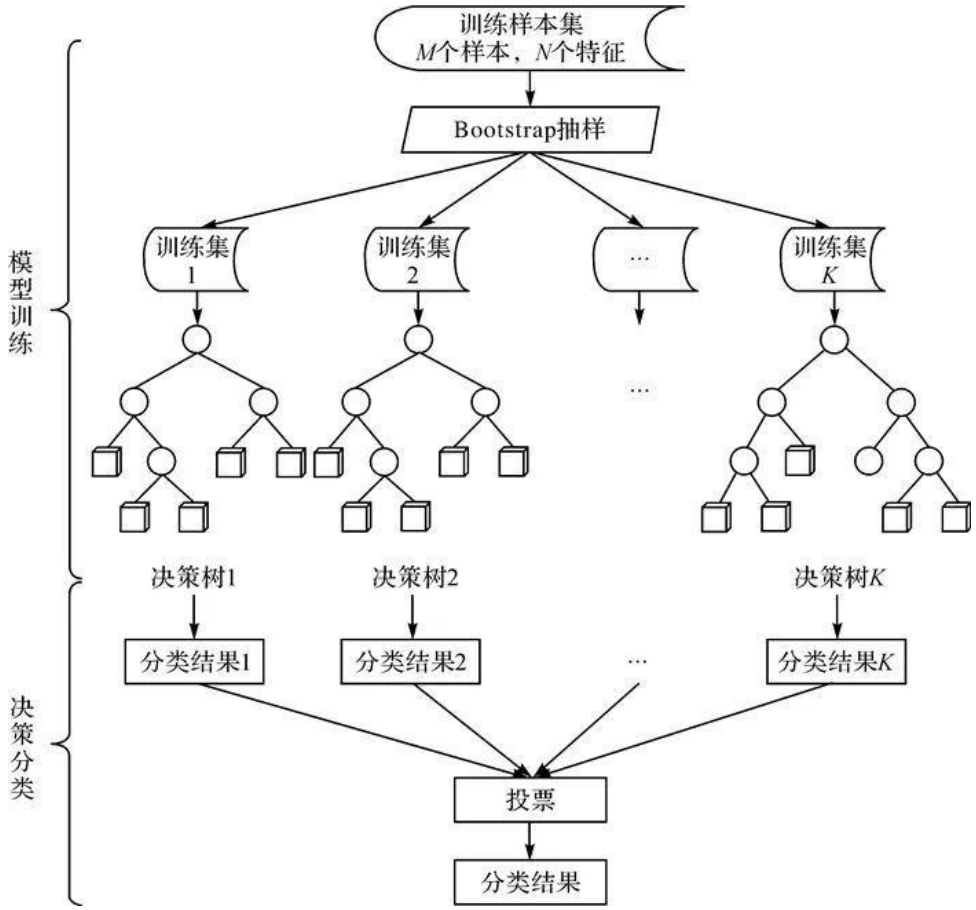

图 6　随机森林训练流程

由 Chen 等人提出的 XGBoost 是对梯度提升树框架(Gradient Boost Decision Tree Framework)的一种具体实现，以 CART 决策树作为基学习器，既可以用于分类问题也可以用于回归问题[6]。XGBoost 的基本形式为

$$\begin{cases} L(\phi)=\sum_i l(\hat{y}_i,y_i)+\sum_k \Omega(f_k) \\ \Omega(f)=\gamma T+\frac{1}{2}\lambda\,||\omega||^2 \end{cases} \tag{17}$$

在迭代过程中的损失函数表达式如下

$$\begin{aligned} L^{(t)} &= \sum_{i=1}^{n} l\left(y_i,\hat{y}_i^{(t-1)}+f_t(x_i)\right)+\Omega(f_t) \\ &\simeq \sum_{i=1}^{n}\left[l\left(y_i,\hat{y}^{(t-1)}\right)+g_i f_t(x_i)+\frac{1}{2}h_i f_t^2(x_i)\right]+\Omega(f_t) \end{aligned} \tag{18}$$

不同于普通的 GBDT，XGBoost 在 L 后加入了与决策树深度 T 和分类器分数 w 有关的正则化项，使得精度本就出众的模型更具有鲁棒性。此外从某种意义上讲，它本身还具有降维的效果，并且对类别失衡的数据效果较好。

XGBoost 由于是基于 Boosting 原理训练基学习器，每一步的学习器是在上一步学习器的基础上改进得到。其迭代训练算法如下所示：

Algorithm 1: Exact Greedy Algorithm for Split Finding

Input: I, instance set of current node
Input: d, feature dimension
$gain \leftarrow 0$
$G \leftarrow \sum_{i \in I} g_i$, $H \leftarrow \sum_{i \in I} h_i$
for $k = 1$ ***to*** m **do**
  $G_L \leftarrow 0$, $H_L \leftarrow 0$
  for j *in sorted*$(I,$ *by* $\mathbf{x}_{jk})$ **do**
    $G_L \leftarrow G_L + g_j$, $H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L$, $H_R \leftarrow H - H_L$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
end
Output: Split with max score

图 7　XGBoost 算法训练过程

XGBoost 使用了和 CART 回归树一样的想法，利用贪婪算法，将所有特征升序排序，然后遍历所有特征的所有特征划分点，不同的是使用的目标函数不一样。具体做法就是求分裂后的目标函数值比分裂前的目标函数的增益。而 XGBoost 虽然采用了 Boosting 系列[7]方法，但同样可以在特征级上做到并行训练，对不同特征的基尼指数进行并行化计算。

而为了调节出最优参数，我们选择使用遗传算法进行调参。这一启发式搜索的方式相比暴力的网格搜索能够节省更多时间空间资源。遗传算法是一类借鉴生物界自然选择和自然遗传机制的随机搜索算法，通过模拟生物的遗传、变异等自然现象搜索函数极值[7]。其主要特点是直接对结构对象进行操作，不存在求导和函数连续性的限定；具有更好的全局寻优能力；不需要确定的规则就能自动获取和指导优化的搜索空间，自适应地调整搜索方向[8,9]。

遗传算法借鉴了生物学的概念，首先需要对问题进行编码，通常是将函数编码为二进制代码以后，随机产生初始种群作为初始解。随后是遗传算法的核心操作之一——“选择”，通常选择首先要计算出个体的适应度，根据适应度不同来采取不同选择方法进行选择，常用方法有适应度比例法、期望值法、排位次法、轮盘赌法等[10]。

在自然界中，基因的突变与染色体的交叉组合是常见现象，这里也需要在选择以后按照一定的概率发生突变和组合。不断重复上述操作直到收敛，得到的解即最优。遗传算法基本思想如图 8 所示。

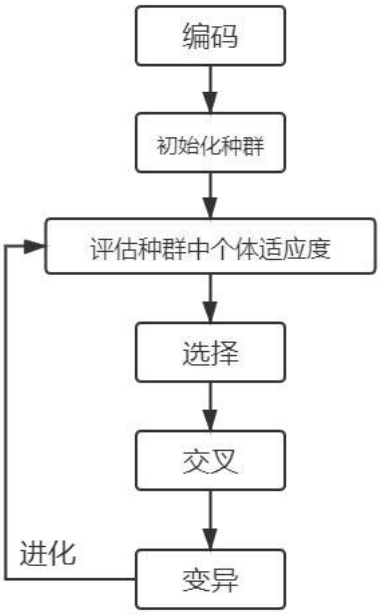

图 8　遗传算法基本思想示意图

经对比测试，我们得到 XGBoost 的分类结果表格如下：

表 2　分类结果

类别	precision	recall	F1-score	support
0	0.76	0.89	0.82	58
1	0.89	0.76	0.82	70
accuracy			0.82	118
Macro-avg	0.83	0.82	0.82	118
Micro-avg	0.83	0.83	0.83	118

从表中可以看到，模型的准确率和 F1 分数都达到了 0.82，在测试集上的 AUC 也可以达到 0.8909，属于一个相当好的结果，说明模型在一定程度上确实可以达到想要的效果。但之所以 SOTA 方法的效果没有突破 0.9 的原因主要集中于流失用户和非流失用户的宏观网络行为测度区分度并不够显著。

我们也将不同模型对比的 AUC 曲线绘制在图中。我们将缺失值按-1 填充，对比了 XGBoost、LightGBM、随机森林、逻辑回归几种方法的 AUC 曲线和准确值。

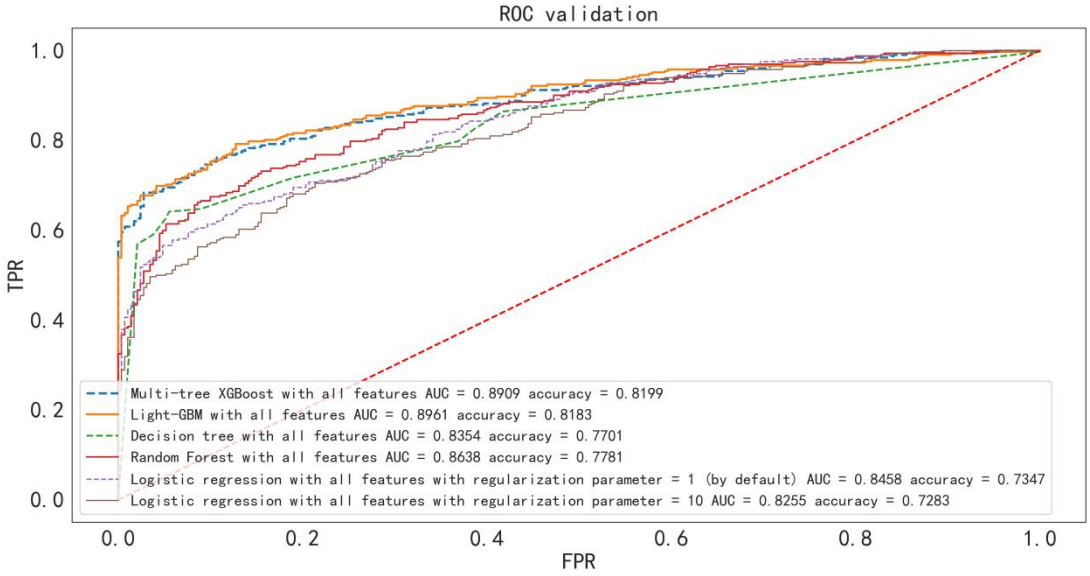

图 9　AUC 曲线

从图中可以看到，XGBoost 不仅能自动处理缺失数据，在 AUC 曲线和准确率值的分类综合评价上也是最优的。其次就是 Light—GBM 的 AUC 曲线和准确度在总体的情况下处于第二位。

而我们将 XGBoost 基学习器的树状结构绘制于下图:

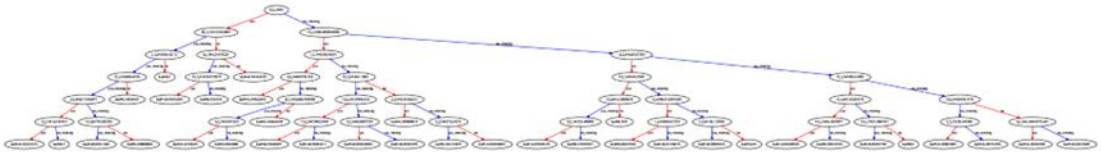

图 10　树状结构图

处于核心地位的根节点和第一层节点、第二层节点基本开头以 1、10、11、13 为主，说明真正影响用户流失的并非网络流量而是计算和存储。其中根节点为 13,1,9，判定阈值为 94；左子节点 10,1,14 的阈值为 12.63；右子节点 13,1,10 阈值 0.076，是相对比较重要的节点。这说明网络用户流失的根源在于计算行为，若计算性能出现阈值过低则开始注意存储行为。

最终，在附件 4 当中我们根据用户行为筛选出 124 名可能流失的用户。

5.4 问题四模型的建立与求解

5.4.1 模型的建立

问题四的求解采用二阶段分类策略。首先，基于问题三中对用户网络行为测量将用户进行流失用户和非流失用户的二分类，这一过程可以作为流失时间预测的 baseline，即初步判定哪些用户会流失再进行流失时间的预测会更准确。

第二阶段为了准确预测流失时间，按照问题所给的时间范围再将其抽象为一个多分类问题。从流失用户中重新训练分类器。

因此进一步得出的多阶段分类的结构的准确度更高，输出的结果更加符合预期。

多阶段分类的结果如表所示：

表 3　多阶段分类结果

类别	precision	recall	F1-score	support
A	1.00	1.00	1.00	3

B	0.87	0.88	0.87	11
C	0.94	0.98	0.96	28
D	1.00	1.00	1.00	33
accuracy			0.96	75
Macro-avg	0.95	0.97	0.96	75
Micro-avg	0.96	0.96	0.96	75

5.4.2 模型的求解

时间分类的 f1 分数达到了 0.96，说明这是一个效果相对良好的分类器。而对附件 5 中预测出的流失用户进行时间预测，得到流失时间的饼状图如图所示：

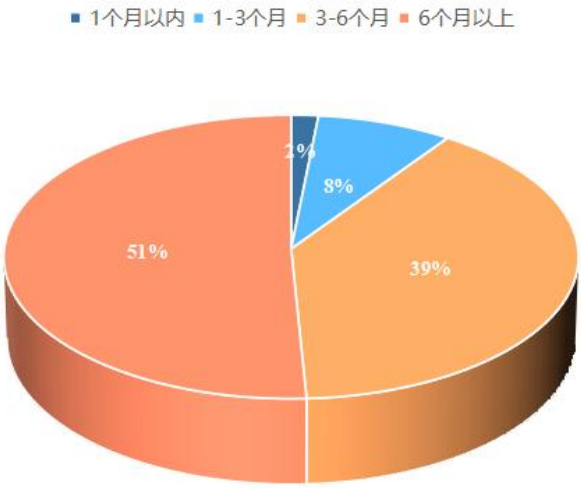

图 11　流失用户时间预测图

从图 11 中可以看到，大部分被预测为流失用户的流失时间其实并不短。超过 50%的用户需要六个月以上才有流失的可能。但对于流失时间在一个月以内的用户，我们经过分析，认为此类用户的计算行为和存储行为存在高度的不稳定性。除了降幅波动大以外，还存在波动速度快的特征。这是由于此类用户需要频繁进行数据的写入读出和修改所导致。需要根据不同等级增强用户黏性。

六、 模型的评价、改进与推广

6.1 模型的优点

该模型从统计学角度出发，结合机器学习算法，保证了结果的可靠性，随后我们根据一些模型评价指标评估，认为模型在本数据集上取得了较好的效果。我们认为，我们在这一系列问题中提出的模型有以下优点：

1.使用统计学与机器学习算法，结果准确可靠，模型表现好。

2.利用 XGBoost 进行了特征工程与特征筛选，有力降低维度选择最核心的指标。

3.通过聚类算法和因子分析对客户特征进行区分与画像，能够描述不同群体的独立特征，有助于精准营销。

4.在问题四的营销策略中根据 XGBoost 的生成树提出了营销策略生成算法，自底而上对可行策略进行递归式的变化分析，能高效准确地分析营销策略。

6.2 模型的缺点

尽管模型整体表现良好，但我们认为还存在一些不足之处，例如：

1.对于客户数据的画像，未能将两类客户在画像的同时进行区分，可以尝试在降维算法中加入聚类的效果进行测试。

2.营销策略的生成仍然存在一定的问题，算法时间复杂度较高并且严重依赖生成树，而且没能充分利用到营销成本与提升百分点的关系。可以考虑从结构方程的角度入手进行进一步分析。

6.3 模型的改进

该模型除了可以使用 XGBoost 作为分类器以外，同样属于 GBDT 框架下的 LightGBM 作为 XGBoost 的一种改进算法也可以用于尝试。另外，对于末尾提出的多目标路径寻优算法中出现有多次循环，加之本身就存在递归寻路的操作使得算法的时间复杂度较大，实际上这一系列循环是可以通过更改算法结构简省的。通过将循环合并或利用 numpy 等加速工具能够提升算法的工作效率。

6.4 模型的推广

该问题的求解方法不仅可以用于电动汽车的销售领域，结合问题实际的简单数值处理原理纯粹但有高度的可解释性和效率;统计学与机器学习结合的特征筛选与特征工程能高效筛选出强影响特征;聚类和以因子分析为代表的降维算法结合揭示了不同用户的特征而 XGBoost 也是一种很好的分类模型;最后，我们提出了基于寻路的最优销售策略生成算法。这一系列模型具有高度的可移植性。

七、 参考文献

[1] 李龙.配对样本t检验在实验室分析质量控制中的应用[J].上海计量测试,2020,47(05):32-34+37.

[2] 吕世杰,许茂发,任佳,姚荣,卫智军.卡方独立性检验的实践与可操作性研究[J].统计与管理,2015(05):41-44.

[3] Dong G, Liu H. Feature Engineering for Machine Learning and Data Analytics[M],2018

[4] Bicici Ufuk Can,Akarun Lale. Conditional information gain networks as sparse mixture of experts[J]. Pattern Recognition,2021,120:

[5] Mitchell T M . Machine Learning[M]. McGraw-Hill, 2003.

[6] Wu Cong,Li Hongxin,Ren Jiajia. Research on hierarchical clustering method based on partially-ordered Hasse graph[J]. Future Generation Computer Systems,2021(prepublish):

[7] Godwin Ogbuabor,Ugwoke F. N. Clustering Algorithm for a Healthcare Dataset Using Silhouette Score Value[J]. International Journal of Computer Science and Information Technology,2018,10(2):

[8] 杜丽英.基于数据挖掘的决策树算法分析[J].吉林建筑工程学院学报,2014,31(05):48-50.

[9] 方匡南,吴见彬,朱建平,谢邦昌.随机森林方法研究综述[J].统计与信息论坛,2011,26(03):32-38.

[10] Chen T , Tong H , Benesty M . xgboost: Extreme Gradient Boosting[J]. 2016.

[11] 王照国,张红云,苗夺谦.基于F1值的非极大值抑制阈值自动选取方法[J].智能系统学报,2020,15(05):1006-1012.

[12] 王彦光,朱鸿斌,徐维超.AUC统计特性概述[J].电子世界,2021(13):107-109.

[13]. 余孟杰.产品研发中用户画像的数据模建——从具象到抽象[J].设计艺术研究,2014,4(06):60-64.

[14]. 马永杰,云文霞.遗传算法研究进展[J].计算机应用研究,2012,29(04):1201-1206+1210.

[15]. MATLAB 中文论坛. MATLAB 智能算法 30 个案例分析[M]. 北京航空航天大学出 版社, 2010. P17-P22

[16]. 吴渝, 唐红, 刘洪涛. 网络群体智能与突现计算[M]. 科学出版社, 2012.

[17]. 李士勇，李研，林永茂. 智能优化算法与涌现计算[M]. 清华大学出版社，2020，P115-118.

附录

编程软件：python 3.7.6

附录 1：层次分析法得到的排序

特征	datashape	user_num	delta	var	return	maxdel	correlation	pvalue	score
2, 1, 4	0.011	0.016	0.018	0.636	0.000	0.000	0.009	1.000	0.593
14, 1, 8	0.021	0.048	0.013	0.795	0.000	0.000	0.076	0.527	0.394
2, 3, 9	0.022	0.044	0.265	0.527	0.000	0.000	0.064	0.461	0.385
14, 1, 1	0.021	0.048	0.010	0.466	0.000	0.000	0.077	0.531	0.335
2, 2, 5	0.292	0.522	0.035	0.581	0.000	0.000	0.051	0.927	0.317
2, 3, 3	0.022	0.044	0.010	0.473	0.000	0.000	0.051	0.372	0.312
10, 1, 9	0.344	0.614	0.025	0.645	0.285	0.266	0.009	0.251	0.307
2, 1, 3	0.011	0.016	0.010	0.476	0.000	0.000	0.205	0.845	0.295
7, 2, 1	0.000	0.000	0.011	0.542	0.000	0.000	0.000	1.000	0.276
10, 1, 8	0.059	0.092	0.010	0.479	0.000	0.000	0.070	0.728	0.274
2, 3, 4	0.022	0.044	0.013	0.508	0.000	0.000	0.055	0.398	0.271
4, 2, 4	0.835	0.988	0.040	0.734	0.000	0.000	0.011	0.483	0.269
7, 1, 5	0.006	0.008	0.016	0.440	0.000	0.000	0.029	0.114	0.269
14, 1, 6	0.021	0.048	0.011	0.474	0.000	0.000	0.020	0.145	0.266
7, 2, 7	0.000	0.000	0.011	0.530	0.000	0.000	0.000	1.000	0.264
6, 1, 3	0.322	0.438	0.065	0.647	0.000	0.000	0.024	0.622	0.262
2, 3, 8	0.022	0.044	0.010	0.464	0.000	0.000	0.064	0.460	0.260
2, 2, 2	0.292	0.522	0.012	0.472	0.000	0.000	0.017	0.439	0.255
9, 1, 1	0.001	0.004	0.012	0.440	0.000	0.000	0.255	1.000	0.255
7, 1, 3	0.006	0.008	0.013	0.440	0.000	0.000	0.146	1.000	0.254
14, 1, 9	0.021	0.048	0.015	0.499	0.000	0.000	0.022	0.155	0.254
10, 1, 7	0.344	0.622	0.017	0.542	0.000	0.000	0.001	0.035	0.252
2, 2, 3	0.291	0.522	0.010	0.466	0.000	0.000	0.002	0.036	0.252
7, 2, 3	0.000	0.008	0.010	0.550	0.000	0.000	0.000	1.000	0.250
2, 1, 2	0.011	0.016	0.013	0.504	0.000	0.000	0.021	0.109	0.250
2, 3, 1	0.022	0.044	0.010	0.466	0.000	0.000	0.274	0.750	0.248
11, 2, 2	0.238	0.562	0.019	0.742	0.005	0.002	0.020	0.476	0.248
1, 1, 2	0.680	0.964	0.043	0.704	0.000	0.000	0.010	0.414	0.248
2, 1, 6	0.011	0.016	0.020	0.477	0.000	0.000	0.152	0.707	0.248
14, 1, 4	0.021	0.048	0.010	0.446	0.000	0.000	0.035	0.253	0.248
3, 1, 1	0.002	0.012	0.011	0.441	0.000	0.000	0.000	1.000	0.247
3, 1, 4	0.002	0.012	0.046	0.440	0.000	0.000	0.000	1.000	0.246
10, 1, 6	0.062	0.096	0.066	0.500	0.000	0.000	0.058	0.650	0.246
13, 1, 9	1.000	1.000	0.010	0.532	0.000	0.000	0.056	0.799	0.246
8, 1, 3	0.002	0.008	0.010	0.452	0.000	0.000	0.057	1.000	0.244
4, 2, 1	0.835	0.988	0.040	0.734	0.000	0.000	0.011	0.483	0.244
12, 1, 1	0.043	0.096	0.010	0.476	0.000	0.000	0.001	0.000	0.243

	6, 1, 7	0. 322	0. 438	0. 069	0. 623	0. 000	0. 000	0. 023	0. 609	0. 243
	12, 1, 3	0. 043	0. 096	0. 022	0. 499	0. 000	0. 000	0. 101	0. 827	0. 242
	11, 1, 2	0. 238	0. 562	0. 023	0. 696	0. 005	0. 001	0. 008	0. 191	0. 242
4	10, 1, 1	0. 062	0. 096	0. 022	0. 551	0. 000	0. 000	0. 009	0. 112	0. 242
	7, 2, 8	0. 000	0. 000	0. 011	0. 537	0. 000	0. 000	0. 000	1. 000	0. 241
	7, 1, 1	0. 006	0. 008	0. 013	0. 440	0. 000	0. 000	0. 133	0. 513	0. 241
	2, 2, 8	0. 292	0. 522	0. 010	0. 462	0. 000	0. 000	0. 031	0. 724	0. 241
	1, 1, 1	0. 679	0. 964	0. 011	0. 537	0. 000	0. 000	0. 022	0. 767	0. 241
	11, 2, 1	0. 186	0. 486	0. 012	0. 672	0. 001	0. 000	0. 007	0. 142	0. 240
5	10, 1, 1	0. 062	0. 096	0. 010	0. 502	0. 000	0. 000	0. 020	0. 252	0. 240
	1, 1, 4	0. 679	0. 964	0. 047	0. 725	0. 000	0. 000	0. 009	0. 347	0. 240
	4, 2, 2	0. 835	0. 988	0. 046	0. 735	0. 012	0. 002	0. 017	0. 688	0. 240
	10, 1, 2	0. 344	0. 622	0. 011	0. 467	0. 000	0. 000	0. 011	0. 313	0. 240
	2, 1, 9	0. 011	0. 016	0. 317	0. 519	0. 000	0. 000	1. 000	0. 476	0. 240
	3, 1, 2	0. 002	0. 016	0. 010	0. 440	0. 000	0. 000	0. 000	1. 000	0. 240
	13, 1, 5	0. 248	0. 410	0. 010	0. 469	0. 000	0. 000	0. 025	0. 582	0. 239
	8, 1, 5	0. 002	0. 012	0. 011	0. 599	0. 000	0. 000	0. 088	1. 000	0. 239
3	10, 1, 1	0. 062	0. 096	0. 018	0. 496	0. 000	0. 000	0. 073	0. 765	0. 237
	6, 1, 1	0. 322	0. 438	0. 106	0. 636	0. 000	0. 000	0. 030	0. 729	0. 237
	14, 1, 2	0. 021	0. 048	0. 011	0. 462	0. 000	0. 000	0. 084	0. 571	0. 234
	3, 2, 1	0. 004	0. 016	0. 048	0. 440	0. 000	0. 000	0. 034	1. 000	0. 233
1	10, 1, 1	0. 344	0. 622	0. 020	0. 605	1. 000	1. 000	0. 009	0. 270	0. 226
	2, 1, 5	0. 011	0. 016	0. 104	0. 503	0. 000	0. 000	0. 070	0. 370	0. 225
	7, 1, 6	0. 006	0. 008	0. 015	0. 440	0. 000	0. 000	0. 076	1. 000	0. 223
	7, 1, 4	0. 006	0. 008	0. 016	0. 440	0. 000	0. 000	0. 128	0. 495	0. 223
	4, 2, 3	0. 835	0. 988	0. 046	0. 718	0. 008	0. 001	0. 006	0. 265	0. 222
	12, 1, 2	0. 043	0. 096	0. 010	0. 472	0. 000	0. 000	0. 105	0. 843	0. 222
	4, 1, 2	0. 832	0. 988	0. 046	0. 716	0. 008	0. 001	0. 006	0. 272	0. 221
2	10, 1, 1	0. 344	0. 622	0. 022	0. 605	0. 003	0. 003	0. 008	0. 235	0. 221
	2, 2, 1	0. 292	0. 518	0. 010	0. 463	0. 000	0. 000	0. 038	0. 823	0. 220
	13, 1, 7	0. 680	0. 964	0. 011	0. 544	0. 000	0. 000	0. 004	0. 170	0. 217
	4, 1, 3	0. 832	0. 988	0. 046	0. 721	0. 028	0. 004	0. 010	0. 449	0. 213
	14, 1, 7	0. 021	0. 048	0. 010	0. 467	0. 000	0. 000	0. 107	0. 689	0. 210
	2, 2, 7	0. 292	0. 518	0. 018	0. 569	0. 000	0. 000	0. 033	0. 755	0. 209
0	14, 1, 1	0. 021	0. 048	0. 013	0. 504	0. 000	0. 000	0. 071	0. 495	0. 208
	10, 1, 4	0. 062	0. 096	0. 010	0. 477	0. 000	0. 000	0. 108	0. 921	0. 205
	2, 2, 11	0. 292	0. 522	0. 010	0. 455	0. 000	0. 000	0. 011	0. 293	0. 204

	2, 2, 12	0. 291	0. 522	0. 012	0. 510	0. 000	0. 000	0. 025	0. 611	0. 199
1	13, 1, 1	0. 680	0. 964	0. 047	0. 740	0. 000	0. 000	0. 012	0. 468	0. 198
	2, 2, 9	0. 291	0. 522	0. 252	0. 740	0. 000	0. 000	0. 011	0. 297	0. 198
	8, 1, 12	0. 002	0. 008	0. 012	0. 502	0. 000	0. 000	0. 038	1. 000	0. 195
	2, 3, 6	0. 022	0. 044	0. 020	0. 513	0. 000	0. 000	0. 063	0. 454	0. 188
	6, 1, 6	0. 322	0. 438	0. 116	0. 672	0. 000	0. 000	0. 022	0. 572	0. 187
	10, 1, 3	0. 059	0. 092	1. 000	0. 564	0. 000	0. 000	0. 037	0. 442	0. 186
	2, 3, 10	0. 022	0. 044	0. 010	0. 465	0. 000	0. 000	0. 103	0. 680	0. 185
	2, 1, 8	0. 011	0. 016	0. 010	0. 473	0. 000	0. 000	0. 114	0. 570	0. 183
	13, 1, 6	0. 679	0. 964	0. 015	0. 584	0. 000	0. 000	0. 012	0. 463	0. 183
	8, 1, 7	0. 002	0. 008	0. 012	0. 508	0. 000	0. 000	0. 145	1. 000	0. 183
	7, 1, 2	0. 006	0. 008	0. 015	0. 440	0. 000	0. 000	0. 080	0. 319	0. 180
	4, 1, 5	0. 832	0. 988	0. 366	0. 665	0. 000	0. 000	0. 025	0. 865	0. 180
	13, 1, 4	0. 680	0. 964	0. 014	0. 543	0. 000	0. 000	0. 001	0. 024	0. 174
	6, 1, 4	0. 322	0. 438	0. 141	0. 679	0. 001	0. 000	0. 014	0. 380	0. 170
8	10, 1, 1	0. 344	0. 614	0. 022	0. 000	0. 419	0. 391	0. 009	0. 274	0. 164
	15, 1, 2	0. 009	0. 036	0. 010	0. 464	0. 000	0. 000	0. 081	0. 386	0. 161
	15, 1, 1	0. 009	0. 032	0. 034	0. 623	0. 000	0. 000	0. 029	1. 000	0. 161
7	10, 1, 1	0. 059	0. 092	0. 062	0. 511	0. 000	0. 000	0. 077	0. 775	0. 157
	15, 1, 4	0. 009	0. 032	0. 011	0. 488	0. 000	0. 000	0. 417	1. 000	0. 156
	7, 2, 6	0. 000	0. 008	0. 011	0. 538	0. 000	0. 000	0. 000	1. 000	0. 156
	11, 1, 1	0. 187	0. 494	0. 016	0. 670	0. 002	0. 001	0. 003	0. 057	0. 155
	8, 1, 6	0. 002	0. 008	0. 013	0. 518	0. 000	0. 000	0. 080	1. 000	0. 151
	2, 3, 7	0. 022	0. 044	0. 021	0. 498	0. 000	0. 000	0. 037	0. 273	0. 148
	4, 1, 1	0. 833	0. 988	0. 041	0. 728	0. 000	0. 000	0. 003	0. 156	0. 147
	6, 1, 5	0. 322	0. 438	0. 061	0. 671	0. 000	0. 000	0. 029	0. 719	0. 142
	2, 3, 5	0. 022	0. 044	0. 092	0. 592	0. 000	0. 000	0. 048	0. 355	0. 142
	2, 1, 7	0. 011	0. 016	0. 021	0. 625	0. 000	0. 000	0. 009	1. 000	0. 141
	14, 1, 3	0. 021	0. 048	0. 010	0. 461	0. 000	0. 000	0. 096	0. 634	0. 140
	2, 1, 1	0. 011	0. 016	0. 010	0. 479	0. 000	0. 000	0. 012	1. 000	0. 138
	7, 2, 5	0. 000	0. 000	0. 011	0. 534	0. 000	0. 000	0. 000	1. 000	0. 138
	4, 1, 4	0. 833	0. 988	0. 041	0. 728	0. 000	0. 000	0. 004	0. 157	0. 132
	7, 2, 4	0. 000	0. 000	0. 010	0. 531	0. 000	0. 000	0. 000	1. 000	0. 130
	2, 1, 10	0. 011	0. 016	0. 010	0. 475	0. 000	0. 000	0. 227	1. 000	0. 121
	2, 2, 10	0. 292	0. 518	0. 020	0. 496	0. 000	0. 000	0. 003	0. 062	0. 120
0	13, 1, 1	0. 296	0. 486	0. 011	0. 525	0. 000	0. 000	0. 025	0. 614	0. 118
	7, 2, 2	0. 000	0. 000	0. 011	0. 526	0. 000	0. 000	0. 000	1. 000	0. 118
	3, 1, 3	0. 002	0. 008	0. 036	0. 440	0. 000	0. 000	0. 000	1. 000	0. 114
	14, 1, 5	0. 021	0. 048	0. 010	0. 466	0. 000	0. 000	0. 110	0. 700	0. 111

8, 1, 9	0.002	0.012	0.012	0.503	0.000	0.000	0.164	1.000	0.108
10, 1, 10	0.059	0.092	0.010	0.475	0.000	0.000	0.115	0.931	0.104
13, 1, 1	0.240	0.398	0.010	0.471	0.000	0.000	0.002	0.033	0.100
7, 1, 8	0.006	0.008	0.014	0.440	0.000	0.000	0.077	0.309	0.100
13, 1, 8	0.679	0.964	0.044	0.715	0.000	0.000	0.010	0.414	0.096
9, 1, 4	0.001	0.004	0.010	0.441	0.000	0.000	0.106	1.000	0.094
13, 1, 2	0.296	0.486	0.010	0.483	0.000	0.000	0.033	0.764	0.092
2, 3, 11	0.022	0.044	0.010	0.448	0.000	0.000	0.070	0.498	0.091
10, 1, 5	0.062	0.096	0.000	0.492	0.000	0.000	0.015	0.182	0.091
13, 1, 3	0.296	0.490	0.011	1.000	0.000	0.000	0.006	0.170	0.083
1, 1, 3	0.680	0.964	0.015	0.583	0.000	0.000	0.001	0.045	0.077
15, 1, 3	0.009	0.036	0.027	0.635	0.000	0.000	0.063	0.302	0.076
2, 2, 6	0.291	0.522	0.020	0.569	0.000	0.000	0.040	0.842	0.074
4, 2, 5	0.835	0.988	0.328	0.660	0.000	0.000	0.030	0.923	0.064
2, 2, 4	0.292	0.518	0.021	0.523	0.000	0.000	0.022	0.562	0.063
13, 1, 12	0.296	0.486	0.010	0.511	0.000	0.000	0.038	0.818	0.060
7, 1, 7	0.006	0.008	0.013	0.440	0.000	0.000	0.178	0.647	0.050
6, 1, 2	0.322	0.438	0.072	0.716	0.000	0.000	0.019	0.525	0.049
2, 3, 2	0.022	0.044	0.013	0.488	0.000	0.000	0.029	0.214	0.033

附表 2： 用户画像

用户分级	计算	存储	网络	风险等级
User 9	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 24	计算稳定型用户	高存储量用户	网络不稳定用户	1
User 28	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 33	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 34	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 43	计算变化快	存储不稳定用户	网络稳定型用户	2
User 61	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 66	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 72	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 75	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 76	计算变化慢	存储稳定型用户	网络不稳定用户	2
User 92	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 98	计算稳定型用户	低存储量用户	网络不稳定用户	1
User 105	计算不稳定用户	低存储量用户	网络稳定型用户	1
User 114	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 124	计算稳定型用户	存储量稳定用户	网络变化慢用户	3
User 128	计算不稳定用户	高存储量用户	网络稳定型用户	1

User 144	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 151	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 154	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 157	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 162	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 177	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 181	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 183	计算稳定型用户	存储稳步变化用户	网络变化慢用户	3
User 191	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 206	计算变化慢	存储不稳定用户	网络不稳定用户	2
User 213	计算变化慢	存储不稳定用户	网络不稳定用户	2
User 215	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 224	计算变化快	存储稳定型用户	网络稳定型用户	2
User 226	计算变化快	存储不稳定用户	网络稳定型用户	2
User 233	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 242	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 246	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 249	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 252	计算稳定型用户	存储量稳定用户	网络变化快用户	3
User 261	计算稳定型用户	存储强波动用户	网络变化慢用户	3
User 274	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 278	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 283	计算变化快	存储稳定型用户	网络稳定型用户	2
User 295	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 297	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 299	计算变化快	存储不稳定用户	网络稳定型用户	2
User 302	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 305	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 319	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 320	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 330	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 332	计算不稳定用户	存储快波动用户	网络变化慢用户	3
User 335	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 337	计算不稳定用户	低存储量用户	网络稳定型用户	1
User 347	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 356	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 360	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 362	计算变化慢	存储不稳定用户	网络稳定型用户	2

User 371	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 382	计算稳定型用户	存储变化快用户	网络变化慢用户	3
User 400	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 405	计算变化慢	存储不稳定用户	网络不稳定用户	2
User 417	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 429	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 449	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 452	计算变化快	存储稳定型用户	网络稳定型用户	2
User 454	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 455	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 472	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 473	计算变化慢	存储稳定型用户	网络不稳定用户	2
User 481	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 483	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 486	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 487	计算变化慢	存储稳定型用户	网络不稳定用户	2
User 488	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 489	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 492	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 499	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 508	计算稳定型用户	存储稳步变化用户	网络变化慢用户	3
User 509	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 516	计算稳定型用户	存储量稳定用户	网络变化慢用户	3
User 517	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 522	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 531	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 547	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 552	计算不稳定用户	高存储量用户	网络稳定型用户	1
User 555	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 557	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 558	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 559	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 560	计算变化快	存储稳定型用户	网络稳定型用户	2
User 577	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 579	计算变化慢	存储稳定型用户	网络不稳定用户	2
User 609	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 613	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 617	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 618	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 621	计算变化慢	存储稳定型用户	网络稳定型用户	2

User 625	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 633	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 634	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 643	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 646	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 648	计算不稳定用户	存储快波动用户	网络变化慢用户	3
User 659	计算稳定型用户	高存储量用户	网络不稳定用户	1
User 661	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 663	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 668	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 673	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 680	计算不稳定用户	高存储量用户	网络稳定型用户	1
User 691	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 696	计算不稳定用户	低存储量用户	网络稳定型用户	1
User 700	计算不稳定用户	低存储量用户	网络稳定型用户	1
User 711	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 714	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 719	计算稳定型用户	存储快波动用户	网络变化慢用户	3
User 729	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 732	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 734	计算变化快	存储稳定型用户	网络稳定型用户	2
User 748	计算变化慢	存储稳定型用户	网络不稳定用户	2
User 749	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 750	计算变化快	存储不稳定用户	网络稳定型用户	2
User 754	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 759	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 768	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 770	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 779	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 783	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 788	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 790	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 791	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 794	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 804	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 805	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 813	计算变化快	存储稳定型用户	网络稳定型用户	2
User 816	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 818	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 820	计算稳定型用户	低存储量用户	网络稳定型用户	1

User 824	计算不稳定用户	高存储量用户	网络稳定型用户	1
User 832	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 835	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 837	计算变化快	存储稳定型用户	网络稳定型用户	2
User 839	计算稳定型用户	存储强波动用户	网络变化快用户	3
User 843	计算变化快	存储稳定型用户	网络稳定型用户	2
User 854	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 855	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 858	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 859	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 860	计算不稳定用户	低存储量用户	网络不稳定用户	1
User 867	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 868	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 872	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 882	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 883	计算变化快	存储不稳定用户	网络稳定型用户	2
User 888	计算稳定型用户	存储快波动用户	网络变化慢用户	3
User 898	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 899	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 903	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 905	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 915	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 926	计算变化快	存储不稳定用户	网络稳定型用户	2
User 927	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 929	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 931	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 942	计算变化快	存储稳定型用户	网络稳定型用户	2
User 943	计算稳定型用户	存储强波动用户	网络变化慢用户	3
User 948	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 949	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 950	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 958	计算变化慢	存储稳定型用户	网络不稳定用户	2
User 961	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 964	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 969	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 971	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 973	计算不稳定用户	低存储量用户	网络稳定型用户	1
User 985	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 987	计算变化慢	存储稳定型用户	网络稳定型用户	2

User 994	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1006	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1009	计算变化慢	存储稳定型用户	网络不稳定用户	2
User 1010	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 1020	计算变化快	存储稳定型用户	网络稳定型用户	2
User 1027	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 1036	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1042	计算变化快	存储不稳定用户	网络稳定型用户	2
User 1047	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 1049	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 1056	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 1064	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 1066	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1070	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 1075	计算稳定型用户	存储稳步变化用户	网络变化慢用户	3
User 1076	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1080	计算不稳定用户	低存储量用户	网络稳定型用户	1
User 1085	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1092	计算不稳定用户	低存储量用户	网络稳定型用户	1
User 1097	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 1115	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 1120	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 1124	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 1131	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1132	计算不稳定用户	低存储量用户	网络稳定型用户	1
User 1144	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1148	计算稳定型用户	低存储量用户	网络不稳定用户	1
User 1153	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 1162	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1163	计算稳定型用户	存储快波动用户	网络变化慢用户	3
User 1176	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1180	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 1182	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 1188	计算不稳定用户	高存储量用户	网络稳定型用户	1
User 1199	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 1202	计算变化快	存储稳定型用户	网络稳定型用户	2
User 1205	计算稳定型用户	存储快波动用户	网络变化慢用户	3
User 1209	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 1226	计算不稳定用户	低存储量用户	网络稳定型用户	1

User 1228	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 1230	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 1234	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 1236	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 1240	计算稳定型用户	存储强波动用户	网络变化慢用户	3
User 1244	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 1254	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1275	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 1282	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 1305	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1306	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 1307	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 1312	计算稳定型用户	高存储量用户	网络稳定型用户	1
User 1318	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 1332	计算变化慢	存储不稳定用户	网络稳定型用户	2
User 1334	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1337	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1342	计算不稳定用户	存储强波动用户	网络变化慢用户	3
User 1352	计算不稳定用户	低存储量用户	网络稳定型用户	1
User 1353	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 1360	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1366	计算变化快	存储不稳定用户	网络稳定型用户	2
User 1370	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1372	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1373	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1382	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 1385	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1401	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 1408	计算不稳定用户	高存储量用户	网络稳定型用户	1
User 1410	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1411	计算不稳定用户	高存储量用户	网络稳定型用户	1
User 1422	计算变化慢	存储稳定型用户	网络稳定型用户	2
User 1423	计算稳定型用户	高存储量用户	网络不稳定用户	1
User 1433	计算稳定型用户	低存储量用户	网络稳定型用户	1
User 1435	计算稳定型用户	存储强波动用户	网络变化慢用户	3
User 1438	计算变化快	存储不稳定用户	网络稳定型用户	2
User 1454	计算变化慢	存储不稳定用户	网络稳定型用户	2
