基于系统决策树分类的移动用户体验影响因素研究

摘要

本题主要是一个研究预测模型，以中国移动通信集团北京公司为背景，让客户根据自身在网络覆盖与信号强度方面的体验和语音通话过程中的整体体验来进行语音通话整体满意度的打分，统计出客户语音业务体验中的影响因素，从而提升客户语音业务满意度。通过分析影响满意度的各项因素，得出量化分析结果，进而进行预测研究。本文主要建立了决策树分类模型，随机森林和梯度提升树分类模型，基于这三个模型，进行附加 1 和附件 2 的满意度评估，附件 3 和附件 4 的打分预测。

针对问题一，主要有三个小问题，首先分析影响客户语音业务的主要因素，然后分析影响客户上网业务的主要因素，最后给出各因素对客户打分影响程度的量化分析和结果，也就是说给出个影响因素的影响权重。在解决问题时，首先进行了数据预处理的操作，对附件 1 和附件 2 中的数据进行了数据的删减与补充、数据编码和整体性分析等操作，同时进行了异常值的处理。然后，我们建立决策树分类模型，随机森林分类模型来解决该问题，训练数据的过程中，对数据的预测精度进行了较好的把控，根据预测精度选择合适的模型，合适的参数值。主要使用了决策树的分类模型，得到了语音业务和手机上网业务是相关影响因素的权重，在文中也给出了相应的表格。之后得出结果：影响客户语音业务满意度的主要因素有 4 个，分别是当月 ARPU、是否遇到网络问题、前三月 MOU 和 GPRS 国际漫游流量；影响客户手机上网业务满意度的主要因素有 4 个，分别是上网过程中网络时断时续或时快时慢、重定向次数、优酷视频使用流量和当月 MOU。最后计算数据得出了各因素对客户打分项目的影响程度，得出了量化分析结果，在文中主要以权重表格的方式呈现。

针对问题二，根据题目要求，需要建立客户打分基于相关影响因素的相关数学模型，该题仍然可以使用第一问的模型，在这里补充使用了梯度提升树模型，根据预测精度选择合适的模型。首先我们根据问题一的预测结果，对相关训练集进行预测精度的检查，发现预测精度高于 97%，证明该模型具备较高的合理性，接着开始进行实际性预测，先对附件 3 和附件 4 的表格数据进行预处理，预处理之后使数据成为了与问题一的训练数据完全一样的格式，接着代入模型，得出了所有客户打分的分类预测结果，根据题目要求，将预测结果填入到 result 表格中，完成预测任务，最终我们对结果的合理性进行了分析和解释说明。

关键词：决策树分类 数据预处理 随机森林分类 影响权重 梯度提升树分类

目录

一、问题的重述 1
1.1 问题背景 1
1.2 问题要求 1
二、问题的分析 1
2.1 问题一的分析 1
2.2 问题二的分析 1
三、模型的假设 2
四、符号说明 2
五、模型的建立与求解 2
5.1 问题一的初步分析求解 2
5.1.1 数据预处理 3
5.1.2 数据整体性分析 4
5.2 问题一模型建立与求解 5
5.2.1 基于决策树模型的建立 5
5.2.2 影响客户满意度主要因素 7
5.2.3 各因素对客户打分的影响程度 11
5.3 问题二模型建立与求解 14
5.3.1 决策树分类模型的训练 14
5.3.2 基于决策树分类的预测 15
六、结果检验和误差分析 17
6.1 结果检验 17
6.2 误差分析 17
七、模型的评价与推广 18
7.1 模型的优点 18
7.2 模型的缺点 18
7.3 模型的推广 18
八、参考文献 18
九、附录 19

一、问题的重述

1.1 问题背景

客户对运营商开发产品的服务满意程度即客户满意度，反映出了客户对产品的期望与实际使用体验之间的差异。随着信息越来越透明，产品同质化，客户满意度愈来愈能体现各大运营商市场运营状况。在我们熟知的数字信息时代，客户体验是很重要的一项指标，客户反馈是有利于商业决策的，商业决策有时候可以为公司带来丰厚的回报，各大公司运用合适的经营手段，建立起客户相关的改善体系，进一步实现客户满意度评价的数字化方向转型，从而来推动移动网络这一领域的的高质量和 可持续发展。传统提升客户满意度的方法是根据客户投诉，逐点解决影响用户体验的问题。但是用户的数量，产品的种类，客户的需求促使运营商们需要寻求更加有效的方法，从而实现更早、更全面的提升客户满意度。

中国移动通信集团北京公司，让客户根据自身在网络覆盖与信号强度方面的体验和语音通话过程中的整体体验来进行语音通话整体满意度的打分，统计出客户语音业务体验中的影响因素，从而提升客户语音业务满意度。同时，中国移动北京公司还让客户根据自身在手机上网中的整体体验，以及在网络覆盖与信号强度方面的体验来进行手机上网整体满意度的打分，并统计整理出影响客户上网体验中的影响因素，从而提升客户的上网体验。

1.2 问题要求

问题一：首先第一小问是根据附件一和附件二，分别研究影响客户语音业务和上网业务满意度的主要因素，然后第二小问是在第一小问的基础上给出各因素对客户打分影响程度的量化分析和结果。

问题二：结合问题一中的分析，由此分别建立基于客户语音业务和上网业务中影响客户打分的相关因素的数学模型；据此对附件三和附件四中的客户打分进行预测研究分析，并将预测结果分别填写在 result.xlsx 的 sheet1“语音”和 sheet2“上网”两个工作表中，上传到竞赛平台说明预测的合理性。

二、问题的分析

2.1 问题一的分析

在解决问题一时，根据题目要求，将题目分为三小问。首先分析影响客户语音业务的主要因素，然后分析影响客户上网业务的主要因素，最后给出各因素对客户打分影响程度的量化分析和结果，也就是说给出个影响因素的影响权重。解决问题时，需要对数据进行数据预处理操作，通过观察表格发现，表格中有一些数据缺失，而且为了方便数据处理，需要删减一些数据，同时在经过异常值检测后，完成数据预处理的工作。接下来为了解决问题，引入决策树分类模型，我们又根据预测精度合理的进行了模型的选用和参数选取，最终，我们给出了影响相关满意度的权重，得到了主要因素，同时给出了相关权重表格和统计图。

2.2 问题二的分析

在解决问题二时，根据题目要求，我们发现我们已经建立了客户打分基于相关影响因素的相关数学模型，也就是决策树模型，首先我们根据问题一的预测结果，对相关训练集进行预测精度的检查，发现预测精度高于 95%，证明该模型具备较高的合理性，同时，我们又对该模型的合理性进行了分析和解释说明。然后，解决问题时，首

先对附件 3 和附件 4 的表格数据进行预处理，预处理之后使数据成为了与问题一的训练数据完全一样的格式，接着在模型进行引入后，得出了所有客户打分的分类预测结果，最终，我们对结果的合理性进行了分析和解释说明。

三、模型的假设

为了便于模型求解，现做如下假设：

- 忽略影响因素之间彼此的影响，便于量化分析的计算；
- 检测分析时，不考虑可能产生的极端因素的影响；
- 假定实验数据在进行处理的过程中没有人为的操作误差；
- 假定在试验的过程中除因素自身外其他影响指标的因素都保持不变；
- 预测研究时，不考虑极端异常值对实验预测结果的影响。

四、符号说明

符号	含义
A_i	因素所处的水平
c_i	不同类型的数据值
n_i	样本总容量
p_i	概率
s_{ij}	样本数量
$\bar{x}$	平均值
Md	中位数
S^2	方差
v	输入值的个数
α_i	A_i的效应
$f(x)$	梯度提升树的虚拟函数
p	预测研究分析概率值
$f_M(x)$	梯度提升树的代表性函数
F_α	定义在F分布上的α分位数

五、模型的建立与求解

5.1 问题一的初步分析求解

首先根据题目的要求将题目分为三个小问。首先分析两个主要因素，也就是影响客户语音业务的主要因素和影响客户上网业务的主要因素，其次分析影响权重，也就是给出各因素对客户打分影响程度的量化分析和结果。在解决问题时，需要对数据进

行数据预处理操作，通过观察表格发现，表格中有一些数据缺失，因此需要添加一些数据，而且为了方便数据处理，需要删减一些数据，同时在经过异常值检测后，完成了数据预处理的工作。接下来引入决策树分类模型，我们又根据预测精度合理的进行了模型的选用和参数选取，也就是引入了随机森林等模型。最终，我们给出了影响相关客户打分满意度的权重，得到了主要因素，同时给出了相关权重表格和统计图，并进行了分析说明。

5.1.1 数据预处理

为了方便数据的分析与计算，首先对数据进行预处理，从而更加简便而且更加直观的分析出影响满意度的主要因素，也更加便于通过结合决策树分类的类型，分析出各个主要因素之间的统计规律。

Step1.数据的删减和补充添加

在进行数据的处理之前，需要对表格进行比较深度的理解，通过对表格 5 的一定行分析理解，对表格数据尤其是表格中空白的数据进行理解，接下来去进行数据预处理操作，我们重点关注表格中一些空白的数据。

对于附件一中的用户描述数据是空白的，根据附件 5 中的文字说明，即除了前面几个场景之外的情况，-1 表示没有，98 就表示有，用户描述是具体的情况，没有的时候是空的，可以分析得到这一行数据可以直接删去。

对于附件一中的重定向次数和重定向驻留时长，虽然其数据不是完全缺失，但数据值明显缺失较多，根据附件 5 的相关文字说明，发现这两个数据和上网满意度有较大的关系，和语音满意度几乎没有关系，因此说这一行数据可以直接删去。

对于附件一中的是否关怀用户和是否去过营业厅数据，根据附件 5 的相关信息，发现空白的即为“否”，所以说将空白表格填上“否”即可。下表展示了附件一的数据处理方法。

表 1 附件一数据预处理

用户描述	重定向次数	重定向 驻留时长	是否关怀用户	是否去 过营业厅
直接删去	直接删去	直接删去	填“否”	填“否”

对于附件二中的场景备注数据和现象备注数据虽然有部分表格不是空白，但根据附件 5 的题目说明，也就是除了前面几个场景之外的情况，-1 表示没有，98 就表示有，用户描述是具体的情况，没有的时候是空的，可以分析得到这一行数据可以直接予以删去。

对于附件二中的 APP 小类游戏备注，小类上网备注与上述同理，可以直接删去。对于上网质差次数、脱网次数、重定向次数和 2G 驻留时长等数据，根据题目相关描述，可以将空白值全部填为 0。同样，对于码号资源的激活时间和发卡时间等，也可以直接予以删去数据。下表展示了附件二的数据处理方法。

表 2 附件二数据预处理

场景备注数据	APP 小类游戏 备注	码号资源激 活和发卡时间	上网质差次数、 脱网次数	重定向次数 2G 驻留时长
直接删去	直接删去	直接删去	填“0”	填“0”

Step2.数据编码

已经对数据进行完补充和删去处理后，需要对数据进行编码，也就是对每一个因素用数据表示，这样可以更方便的计算，通过合理分析，编码结果如下表所示：

表 3 数据编码化

类别	选项	编码
2\5 用户	2G	1
	4G	2
	5G	3
语音方式	EPSFB	1
	CSFB	2
	VOLTE	3
是否关怀用户	是	1
	否	2
是否去过营业厅	是	1
	否	2
是否 4G 网络用户	是	1
	否	2
性别	男	1
	不详	2
	女	3
是否全月漫游用户	是	1
	否	2
终端制式	4G 终端	1
	5G 终端	2
	2G 终端	3
终端品牌	华为	1
	小米	2
	苹果	3
	步步高	4
	三星	5
	万普	6
	奇酷	7

5.1.2 数据整体性分析

在完成了数据删减补充后，因为剩下数据仍然剩下很多空值和异常值，为了方便处理数据，我们将空值均填成众数处理。在完成了数据的初步预处理之后，我们仍然需要对数据进行一个总体性描述，也就是一个整体性的分析，下表是整体描述的结果，下表是部分表格，完整表格请看附录部分：

表 4 数据整体性分析计算

变量	最大值	最小值	平均数	标准差	中位数	方差	变异系数
信号强度	10	1	8.34	2.42	9	5.85	0.290
语音通话	10	1	8.64	2.20	10	4.85	0.255

清晰度							
语音通话稳定性	10	1	8.43	2.37	10	5.63	0.339
是否有网络问题	2	1	1.47	0.50	1	0.25	-2.945
居民小区	1	-1	-0.32	0.95	-1	0.90	-3.138
办公室	2	-1	-0.39	1.21	-1	1.47	-0.633
高校	3	-1	-0.91	0.58	-1	0.34	-2.290
商业街	4	-1	-0.60	1.36	-1	1.86	7.340
地铁	5	-1	-0.34	2.50	-1	6.24	-5.680
农村	6	-1	-0.36	2.02	-1	4.10	267.02
高铁	7	-1	0.01	2.66	-1	7.07	5.423
手机没有信号	1	-1	-0.47	0.88	-1	0.78	-1.882
有信号无法拨通	2	-1	-0.47	1.14	-1	1.46	2.423
通话有杂音	3	-1	0.50	2.29	-1	5.25	4.578
串线	4	-1	-0.76	1.17	-1	1.36	-1.530
脱网次数	1275	0	7.74	33.91	0	1149.45	4.381

5.2 问题一模型建立与求解

决策树模型（Decision Tree）就是所有样本情况均发生的情形下，通过计算构成决策树一步一步分析求取净现值的方法，在这里要求净现值是大于等于零的，接着需要进行项目风险的评估，通过评价方法进行决策树评估，此方法是用到了概率分析基础的方法。因为这种方法画出的图形非常像树干，所以这种方法称之为决策树。这种方法多用于机器学习，在该领域，决策树被视为预测模型，这种模型反映了对象的属性和对象值的一种映射性关系。

决策树从其使用结构上来看，它是一种典型的树形结构，树形结构的每一个节点就是一种测试，测试结构就是一种输出，最终形成一种测试集合。

决策树的分类模型是常见的分类方法，它更像是一种监督性的学习，我们认识的监督学习就是假设给定一堆样本，通过分类进行学习，最终进行分类性的预测研究。

5.2.1 基于决策树模型的建立

a. 决策树分类模型

决策树模型在使用方法上，是训练样本这个集和在不断分组，进行这个不断分组的过程中进行计算，决策树是有分支的，决策树计算方法的核心是测试属性的互相选择问题。

ID3 算法是决策树模型的算法之一，该算法基于其他理论学的概念而提炼出来的。假设N是n个样本的集合，假设类标号这个有m个不同的数据值，如果定义m个不同类型的数据$c_i(i = 1,2\ldots,m)$。需要假设n_i是c_i中的样本数量，这样的话如果给定一个样本信息，对其分类，其期望信息由下式表示：

$$I(n_1, n_2, \cdots, n_m) = -\sum_{i=1}^{m} p_i \log_2(p_i) \tag{1}$$

注意该公式中$p_i = n_i/n$是任意样本数值中属于c_i的概率，并且注意该公式中对数函数是以 2 为底的。

假设属性 A 有v个不相同的输值$a_1, a_2, ..., a_v$,这样可以将属性 A 分成v个子集$\{s_1, s_2, ..., s_v\}$,再设其中s_j中的样本在该样本属性 A 中具有相同的值$a_j(1,2, ..., v)$。接着假设s_{ij}是其样本数量，那么由 A 属性分成子集的熵或期望就由下式给出：

$$E(A) = \sum_{j=1}^{n} ((s_{1j} + s_{2j} + \cdots + s_{mj})/s) * I(s_{1j} + s_{2j} + \cdots + s_{mj}) \tag{2}$$

由模型原理可以知道，得到的熵的数量值越小，那么对应的子集划分的精度就会变得越高。假定对于给定的任何子集，由下式可以得到它们的信息期望

$$I(s_{1j} + s_{2j} + \cdots + s_{mj}) = -\sum_{i=1}^{m} p_{ij} \log_2(p_{ij}) \tag{3}$$

该公式中$p_i = n_i/n$同公式(1)原理类似，它是任意样本中属于c_i的概率，在属性 A 中分支能够得到的信息增益是:

$$Gain(A) = I(s_1, s_2, \cdots, s_m) - E(A) \tag{4}$$

另一个用到的算法 C4 5 算法，这个算法的特点就是各级结点的选择性，它可以用增益比率的值的大小作为属性的选择标准，当然这个选择标准不是唯一的。公式如下：

$$\begin{cases} Split(A, s) = -\sum_{i=1}^{c} \frac{|s_i|}{|s|} \log_2 \frac{|s_i|}{|s|} \\ Gain(A, s) = \frac{Gain(s, A)}{Split(s, A)} \end{cases} \tag{5}$$

接下来我们将采用两种分类方法，决策树分类模型，随机森林分类，后面那个分类方法是基于决策树模型进行的，我们将比较这些分类方法，选择预测准确度最高的方法进行。

b. 随机森林分类模型

随机森林模型实质上就是包含决策树分类器的一种分类方法，随机森林模型一个重要的指标就是众数，参数数值的选取离不开众数。Leo Breiman 和 Adele Cutler 两个人共同发现了随机森林这个方法。随机森林术语的来历与贝尔实验室有较大的关系。随机森林方法融合了 Breimans 的"Bootstrap aggregating"想法和 Ho 的"random subspace method"想法，最终建造决策树集合。

随机森林模型将如何构建，主要从以下两个方面进行：

1.数据的随机选取：

随机选取时避免不了随机抽样调查，从原始数据开始进行抽样，这种抽样要注意是需要放回的，最终构造成一个子数据集，这个数据集合的数量很明显与原来是相同的。有利的一点是子数据集合是可以重复的。接下来，需要来构建子决策树，数据集合完成后，将数据放到子决策树中，输出结果从子决策树中来输出。最后，在进行新

数据处理时将其加入到子决策树范围里面，可以通过决策树输出投票情况，得到该模型的各个输出结果。

2.待选特征的随机选取

这个步骤和步骤一具有一定的相似性，随机森林中的子决策树需要用到待选特征，但没有用到所有的待选性特征，应该是从这些所有的特征中随机选择特征，之后再在随机选取的特征中选取最优的特征。这样能够使得随机森林中的决策树都能够彼此不同，这样可以提升系统方法的多样性，从而提升分类的性能。

5.2.2 影响客户满意度主要因素

我们已经引入了以上两个分类模型，接下来需要对附件 1 和附件 2 进行数据的训练，这对于第二问模型的建立有很大作用，实际上解决了第二问的部分内容，所以我们希望用更高的精度进行训练，而且在求解过程中也得到了问题一中影响因素的权重大小，在这一部分中，我们将展示求解过程中的参数值和影响因素的大小等内容。

Step1.语音通话整体满意度

这里先进行数据的训练，数据训练方式一般均按照以下步骤来进行，之后不再赘述训练步骤，训练过程如下图所示：

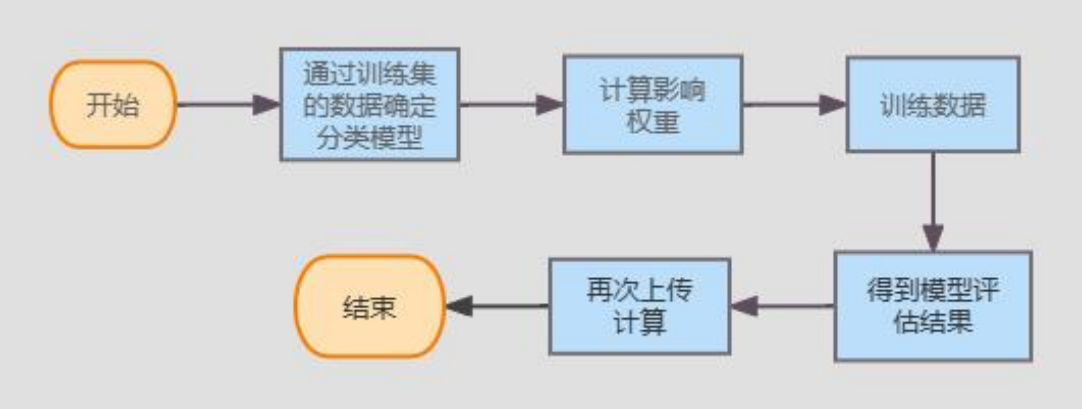

图 1 模型训练流程图

参数输出如下表所示，该表显示了模型的各项参数：

表 5 语音通话整体满意度参数取值

参数名称	参数数值
训练时长	22.175s
数据切分	0.7
数据洗牌	否
交叉验证	否
损失函数	deviance
节点分裂评价准则	friedman_mse
基学习器数量	100
学习率	0.112
无放回采样比例	1
内部节点分裂最小样本数	2
叶子节点最小样本数	1
叶子节点样本最小权重	0

树的最大深度	10
节点划分不纯度阈值	0

接下来显示第二项输出的结果，下表展示了各影响因素的所占权重：

表 6 语音通话整体满意度影响因素权重

影响因素	权重占比
居民小区	0.90%
是否遇到网络问题	11.10%
办公室	1.20%
高校	0.40%
商业街	0.50%
地铁	0.70%
农村	0.60%
高铁	0.60%
其他	0.30%
手机没有信号	0.90%
有信号无法拨通	1.30%
通话过程中突然中断	1.00%
通话有杂音、听不清	1.30%
串线	0.30%
通话过程中另一方听不见	1.00%
脱网次数	3.30%
mos 质差次数	4.40%
未接通电话次数	2.60%
家宽投诉	0.40%
资费投诉	0.00%
4\5 用户	0.40%
语音方式	1.00%
是否关怀用户	0.50%
是否去过营业厅	0.80%
ARPU（家庭宽带）	0.30%
套外流量	0.70%
是否 4G 网络客户	0.00%
套外流量费	1.10%
外省语音占比	0.70%
语音通话时长（分钟）	5.60%
省际漫游时长（分钟）	1.00%
终端品牌	2.50%
当月 ARPU	9.30%
当月 MOU	8.10%
前三月 ARPU	5.30%
前三月 MOU	9.30%
外省流量占比	0.90%
GPRS 总流量	1.40%

GPRS 国际漫游流量	11.60%
是否 5G 网络客户	1.90%
是否实名登记客户	0.50%
客户星级标识	0.60%
当月欠费金额	1.90%
前三个月欠费金额	0.80%

表格中权重分别就代表了每一个因素的重要性程度大小，显然权重越大，该因素的影响程度就越大，这样我们就比较轻松的找到了主要因素。为了更加清楚直观的显示，绘出了以下统计图，如下图所示：

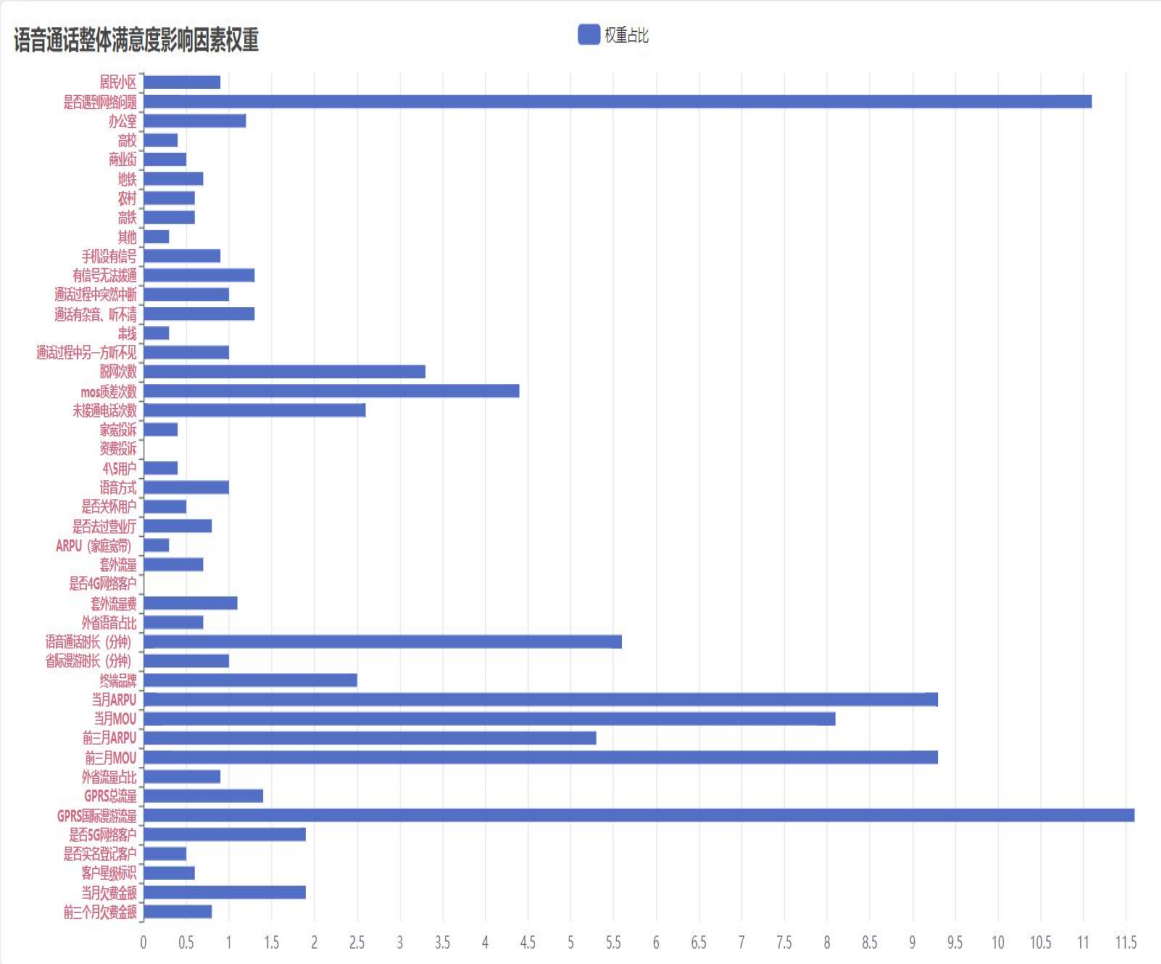

图 2 语音通话满意度影响因素权重图

从以上图表中可以看到，影响语音通话稳定性的主要因素大致有 4 个：是否遇到网络问题，当月 ARPU，前三月 MOU，GPRS 国际漫游流量。

Step2. 手机上网整体满意度

参数输出如下表所示，该表显示了模型的各项参数：

表 7 手机上网整体满意度参数取值

参数名称	参数数值
训练时长	51.175s
数据切分	0.7

数据洗牌	是
交叉验证	否
损失函数	deviance
节点分裂评价准则	friedman_mse
基学习器数量	100
学习率	0.0991
无放回采样比例	0.82
内部节点分裂最小样本数	4
叶子节点最小样本数	2
叶子节点样本最小权重	0.002
树的最大深度	15
节点划分不纯度阈值	0

接下来显示第二项输出的结果，分析与语音通话稳定性相似，这里不再赘述，下表展示了各影响因素的所占权重：

表 8 手机上网满意度影响因素权重

影响因素	权重占比
居民小区	0.60%
办公室	0.40%
高校	0.10%
商业街	0.50%
地铁	0.60%
农村	0.30%
高铁	0.10%
其他	0.10%
网络信号差	2.20%
显示有信号无法上网	0.40%
上网过程中网络时断时续	5.50%
手机上网速度慢	0.60%
看视频卡顿	0.20%
打游戏延时长	0.10%
打开网页慢	0.30%
下载速度慢	0.10%
手机支付较慢	0.20%
爱奇艺	0.20%
优酷	0.20%
腾讯视频	0.20%
芒果 TV	0.30%
搜狐视频	0.20%
抖音	0.40%
快手	0.30%
火山	0.10%
和平精英	0.00%
王者荣耀	0.10%

穿越火线	0.20%
微信	0.10%
手机 QQ	0.10%
拼多多	0.10%
优酷视频使用流量	3.30%
当月 MOU	3.80%
重定向次数	5.20%

同样为了更加直观的显示权重结果，下面是手机上网满意图影响因素权重图，如下图所示：

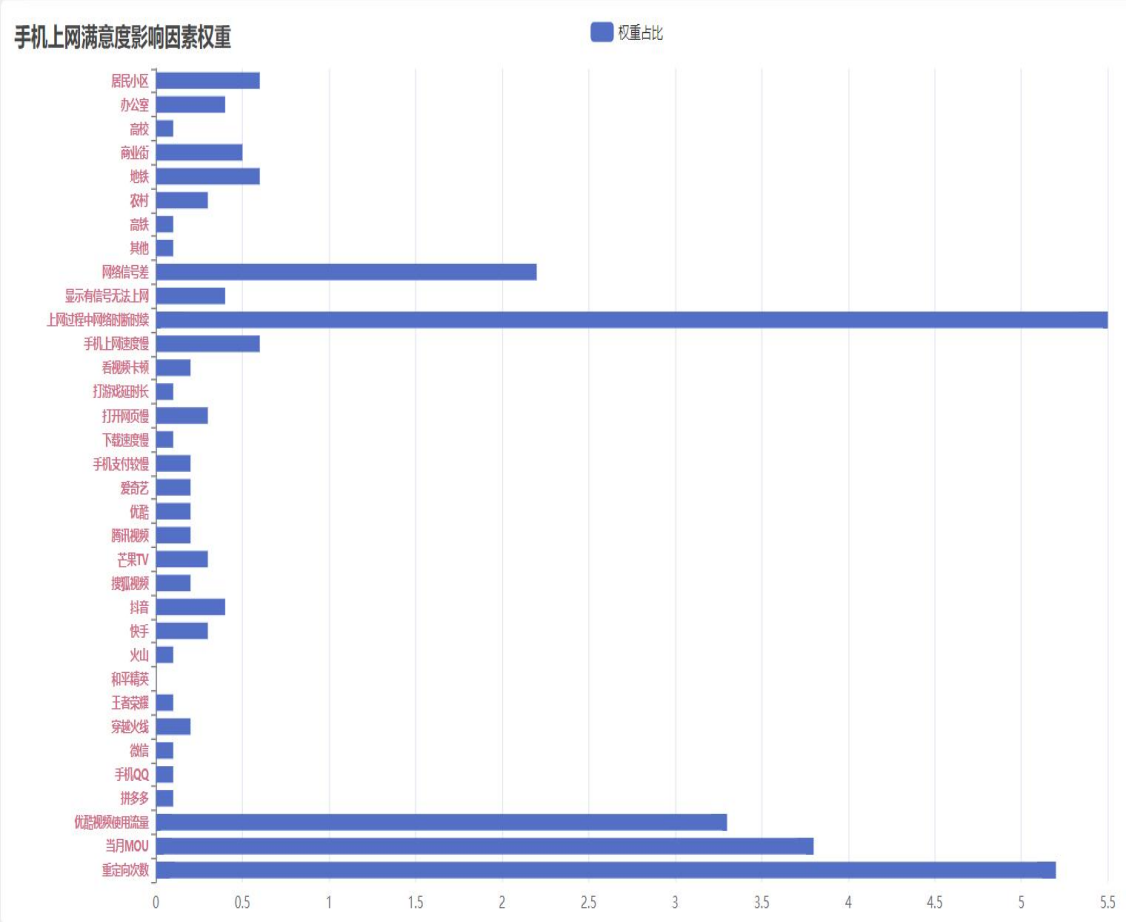

图 3 手机上网满意度影响因素权重图

从以上图表中可以看到，影响手机上网满意度的主要因素大致有 4 个：上网过程中网络时断时续或时快时慢，重定向次数，优酷视频使用流量，当月 MOU。

5.2.3 各因素对客户打分的影响程度

从题干中可知，存在着语音通话稳定性和手机上网满意度两个研究对象，他们分别有 3 个影响客户打分的项目，所以说需要计算 6 个因素对打分项目的影响程度，我们仍然采用之前建立的模型进行计算，以下将展示部分计算结果：

Step1. 附件一

下表仅展示网络覆盖与信号强度，由于模型参数展示方面与前面的方式类似，所以这部分在这里不再展示，全部内容将在附录中展示。

表 9 附件一网络覆盖部分量化分析结果

影响因素	权重占比
居民小区	1.10%
是否遇到网络问题	12.10%
办公室	1.20%
高校	0.30%
商业街	0.50%
地铁	0.60%
农村	0.60%
高铁	0.60%
其他	0.30%
手机没有信号	0.90%
有信号无法拨通	1.30%
通话过程中突然中断	1.00%
通话有杂音、听不清	1.30%
串线	0.30%
通话过程中另一方听不见	1.00%
脱网次数	3.30%
mos 质差次数	4.70%
未接通电话次数	1.60%
家宽投诉	0.40%
资费投诉	0.00%
4\5 用户	0.40%
语音方式	1.00%
是否关怀用户	0.50%
是否去过营业厅	0.80%
ARPU（家庭宽带）	0.30%
套外流量	0.70%
是否 4G 网络客户	0.00%
套外流量费	1.10%
外省语音占比	0.70%
语音通话时长（分钟）	4.60%
省际漫游时长（分钟）	1.10%
终端品牌	2.50%
当月 ARPU	9.50%
当月 MOU	9.10%
前三月 ARPU	4.30%
前三月 MOU	9.30%
外省流量占比	0.90%
GPRS 总流量	12.40%
GPRS 国际漫游流量	1.60%
是否 5G 网络客户	1.90%
是否实名登记客户	0.70%
客户星级标识	0.50%

当月欠费金额	1.00%
前三个月欠费金额	0.90%

Step2.附件二

下表是附件二中网络覆盖与信号强度的量化分析结果，由于模型参数展示方面与前面的方式类似，所以不再显示参数取值方面的内容，网络覆盖与信号强度的量化分析结果如下表所示：

表 10 附件二网络覆盖部分量化分析结果

影响因素	权重占比
居民小区	1.20%
办公室	0.30%
高校	0.20%
商业街	0.30%
地铁	0.90%
农村	0.20%
高铁	0.20%
其他	0.30%
网络信号差	4.20%
显示有信号无法上网	1.40%
上网过程中网络时断时续	1.50%
手机上网速度慢	1.00%
看视频卡顿	0.30%
打游戏延时长	0.90%
打开网页慢	0.20%
下载速度慢	0.20%
手机支付较慢	0.10%
爱奇艺	0.10%
优酷	0.10%
腾讯视频	0.10%
芒果 TV	0.00%
搜狐视频	0.10%
抖音	0.10%
快手	0.10%
火山	0.10%
和平精英	0.00%
王者荣耀	0.10%
穿越火线	0.20%
微信	0.10%
手机 QQ	0.10%
拼多多	0.10%
优酷视频使用流量	3.20%
当月 MOU	3.70%
重定向次数	4.80%

通过得到上述结果，我们得到了所有的因素对客户打分的影响权重，也就是量化分析结果。

5.3 问题二模型建立与求解

由于在第一问的模型已建立一部分，这一部分仍然可以进行部分第二问的操作，所以说这里仅补充一个分类模型，实际上这一问的求解用到了三个小的分类模型，当然前面已经说过的内容不再赘述，在这部分呈现的是模型的精度和预测结果等内容。

对于这个问题，首先我们要进行模型的建立与补充，接着进行模型的精度检查，发现预测精度高于 95%，精度的提高也为后面的模型合理性说明奠定了良好的基础。解决问题时，需要对附件 3 和附件 4 的表格数据先进行预处理，预处理之后使数据成为了与问题一的训练数据相同的格式，载入模型后，得到最终的预测结果。

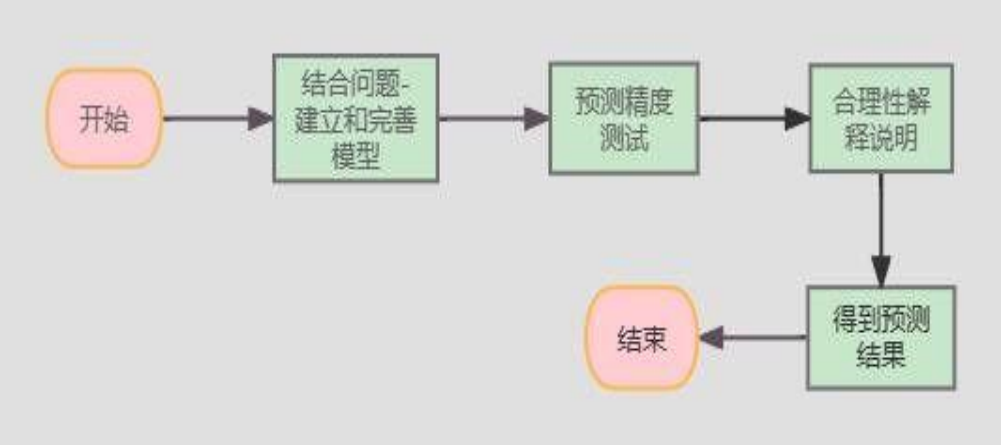

图 4 问题二分析流程图

梯度提升树（GBDT）模型

梯度提升树实际上是一种加法模型，它从始至终的串联一组回归树，然后它对所有的回归树结果进行加和，最终得到一个强学习器。梯度提升树的特点之一是用更多的基学习器来弥补前面学习器的不足。在计算残差过程中，平方损失函数的残差比较容易计算,但是一般形式的损失函数不是很好计算。不过我们可以可以把残差当做成一个函数看待,也就是说我们可以使用梯度的方式来找到最小的残差值。

假设令一颗树为$f(x)$,那么损失函数就相当于是关于$f(x)$的函数,如果让损失函数减小的话,可以用下面的方法：

$$f_M(x) = f_1(x) - \frac{\partial L(y, f_1(x))}{\partial f_1(x)} \tag{6}$$

如果这时增加第一棵树,可以表示为：

$$f_M(x) = f_1(x) + f_2(x) \tag{7}$$

对照公式(6)可以发现新增加的树可以用下列公式表示，相当于负梯度：

$$f_2(x) = -\frac{\partial L(y, f_1(x))}{\partial f_1(x)} \tag{8}$$

该公式表明了负梯度是残差的近似值的说法。

5.3.1 决策树分类模型的训练

在进行这个参数训练的时候，首先要进行参数取值，实际上第一问已经进行了相

关参数取值这里的取值过程与问题一相似，所以说合理不再展示其步骤，这里仅展示模型训练结果的精度，如下表所示：

表 11 模型训练测试结果

	准确率	召回率	精确率	F1
训练集	0.997	0.998	0.998	0.998
测试集	0.623	0.675	0.745	0.720

从以上图表中可以明显的看出，训练集的精度是非常高的，所以说明该模型得到的数据预测是比较准确的。

5.3.2 基于决策树分类的预测

Step1.数据预处理

在对客户打分进行预测研究之前，我们要对附件三和附件四进行数据预处理，同样需要进行数据删减补充，异常值处理等操作，这些操作会帮助后面较好的计算，应该将数据处理成与训练数据相仿的格式，这样才能预测。我们注意到，前两个附件的数据与附件三和附件四的影响因素有一些不同，比如，附件 2 是没有学习强国等这些数据的，所以说这里需要进行数据的删减。与此同时，在附件三和附件四中一些数据相比前两个附件是没有的，比如说是否去过营业厅，是否实名登记用户等内容也是没有的，对于这些数据需要进行补充，可以采用附件 1 或附件 2 的数据进行补充，也可以采用生成随机数的方式，对最终预测结果影响不是很大。对于附件三和附件四出现的空值，可以选择使用众数的方法进行补充，这些数据补充之后可以较好的进行计算处理。

Step2.附件 3 预测

在进行完上述训练模型后，可以进行附件三的预测，首先得到的是满意度预测结果，满意度预测结果在这里不再展示，在附录中会进行部分展示，得到最终结果后，将数据填到题目中要求的 result 表格中，result 表数据部分展示数据如下：

表 12 附件三预测分析结果

用户 id	语音通话 整体满意度	网络覆盖 与信号强度	语音通话 清晰度	语音通话 稳定性
1	10	9	10	10
2	9	6	9	10
3	8	9	10	10
4	8	10	9	9
5	9	5	9	8
6	9	5	8	7
7	10	10	10	10
8	10	10	10	10
9	10	10	10	10
10	7	10	8	10
11	10	10	10	10
12	10	8	10	9
13	9	9	9	10

14	10	10	10	9
16	10	9	10	10
17	9	8	9	10
18	8	9	10	10
19	5	9	5	9
20	10	10	10	4
21	10	10	10	10

Step3.附件 4 预测

采用与附件三相似的方法进行预测，result 表格结果如下：

表 13 附件四预测分析结果

用户 id	手机上网整体满意度	网络覆盖与信号强度	手机上网速度	手机上网稳定性
1	8	9	7	6
2	9	9	8	10
3	8	9	10	10
4	10	8	8	10
5	10	10	8	10
6	10	10	9	10
7	10	10	10	10
8	4	8	10	9
9	8	8	9	8
10	7	6	8	10
11	6	10	10	10
12	6	8	4	3
13	7	10	7	10
14	9	9	8	10
16	2	7	8	6
17	8	9	7	10
18	10	10	10	10
19	10	10	9	9
20	9	8	7	10
21	10	10	10	10

Step4.预测结果合理性说明

在做这个预测精度的测试中，我发现训练集的精度是相对较高的，达到了 95%以上，但是测试集的精度没有达到 80%，实际上这已经比较好的结果了。测试集精度不是太高的原因主要有三个，第一个原因就是空值和异常数值的原因，因为有些空值是用 0 替代了，所以可能产生了偏差；第二个原因主要是这个题的影响因素确实是太多了，一些影响因素很容易出现一些无关特征，可能这也导致了偏差；第三就是人们的主关打分，对于同一个问题，部分人打分偏高，部分人打分偏低，而且说占的比例还不小，所以说这些数据无法去除的，如果直接删去这部分数据而使精度提升至 90%以

上，这是毫无意义的做法。重要的是，这道题的训练精度已经达到了 95%以上，已经足够说明预测的准确性。综上以上观点，这道题进行这样的预测的研究是比较合理的。

六、结果检验和误差分析

6.1 结果检验

对于问题二，在测试训练集的过程中，我们增加了预测精度的检查，精度检查结果如下图所示：

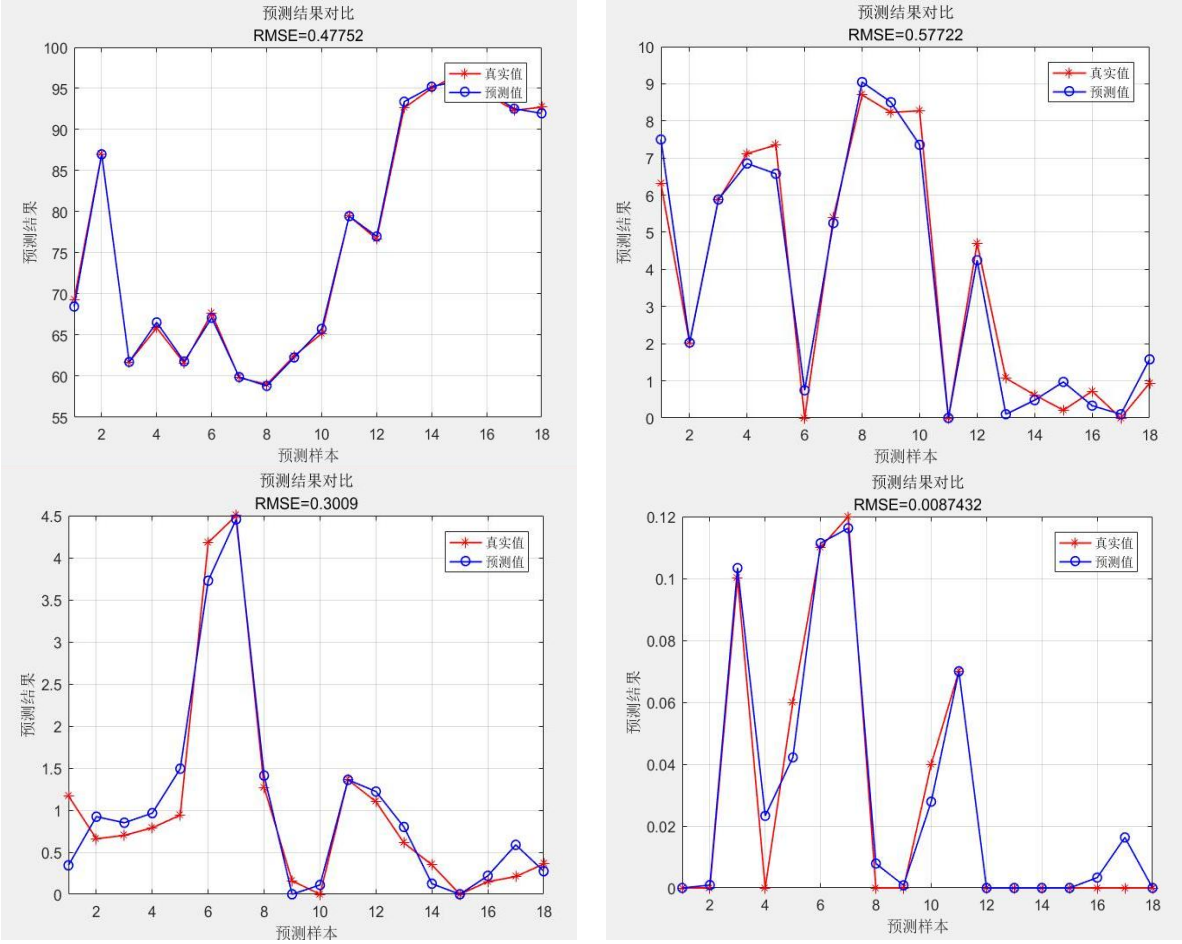

图 5 问题二预测精度检验图

根据问题二的分析，训练集精度是很高的，从以上图中也清晰的展现了出来，说明预测结果符合预期。

6.2 误差分析

对于这个问题，误差主要是出在了测试集精度这个地方，实际上前文在合理性分析说明上已经分析了其出现误差的原因，分别是异常值空值，人为主观打分等原因，在这里不再过多赘述其误差原因。接下来主要说明因素过多可能导致的误差分析，主要有两点：

(1)因素过多可能导致的人为误差，比如数据输入错误等；

(2)因素过多可能对于主要因素的评估出现一定量化分析偏差等，因为特征因素较多，难免会增加分析的复杂性。

七、模型的评价与推广

7.1 模型的优点

（1）问题中所建立的模型很简便而且巧妙的解决问题，并且对于问题做出来一些简化，易于问题的理解和解答；

（2）问题一对大量的数据进行定量分组，接着对数据进行了可视化分析，便于直观的对多组数据进行分析和处理；

（3）问题三中的预测值和真实值很接近，说明拟合的效果很好；

（4）在对于解决该问题中用到了三个分类模型，而且都是基于决策树模型而建立的，模型更容易进行数据的填充分析。

（5）梯度提升树的特点之一是用更多的基学习器来弥前面学习器的不足。在计算残差过程中，我们知道平方损失函数的残差比较容易计算,但是一般形式的损失函数不是很好计算。不过我们可以可以把残差当做成一个函数看待,也就是说我们可以使用梯度的方式来找到最小的残差计算值。

7.2 模型的缺点

（1）模型在进行预测研究时，测试的精度应该有提升的空间；

（2）决策树分类模型在进行测试之前，需要进行比较严格的数据预处理。

7.3 模型的推广

（1）文中所涉及的分类模型可以用于其他方面的研究；

（2）随机森林模型的应用可以为其他预测性研究提供一定的借鉴性；

（3）决策树模型可以应用于导航研究，汽车检测，报文数据研究等领域。

八、参考文献

[1]司守奎，孙玺菁.数学建模算法与应用[M].北京:国防工业出版社，2016-2.

[2]周志华.机器学习[M].第一版.北京：清华大学出版社，2016-1.

[3]李晓辉,杨勇,杨洪伟.基于 BP 神经网络与灰色模型的干旱预测方法研究[J].沈阳农业大学学报,2014,45(02):253-256.

[4]史达伟,沈阳,马晨晨,董京铭,颜佳任.基于决策树算法的江苏省不同区域短时强降水预报研究[J].气象科学,2022,42(05):631-637.

[5]夏安林,杜董生,盛远杰,刘贝.基于决策树的银行目标客户预测算法[J].电脑知识与技术,2022,18(24):8-11+28.DOI:10.14004/j.cnki.ckt.2022.1479.

[6]车维崧,祁静,白文其.基于随机森林及地理围栏的千兆宽带用户规模预测[J].现代信息科技,2023,7(01):61-63.DOI:10.19850/j.cnki.2096-4706.2023.01.016.

[7]林芸舟.基于梯度提升树模型的纯电动二手车价值评估研究[D].重庆理工大学,2022.DOI:10.27753/d.cnki.gcqgx.2022.000673.

九、附录

附录 1

语音通话和上网业务满意度数据处理表格

表 14 语音通话满意度权重表格

影响因素	权重占比
居民小区	0.90%
是否遇到网络问题	11.10%
办公室	1.20%
高校	0.40%
商业街	0.50%
地铁	0.70%
农村	0.60%
高铁	0.60%
其他	0.30%
手机没有信号	0.90%
有信号无法拨通	1.30%
通话过程中突然中断	1.00%
通话有杂音、听不清	1.30%
串线	0.30%
通话过程中另一方听不见	1.00%
脱网次数	3.30%
mos 质差次数	4.40%
未接通电话次数	2.60%
家宽投诉	0.40%
资费投诉	0.00%
4\5 用户	0.40%
语音方式	1.00%
是否关怀用户	0.50%
是否去过营业厅	0.80%
ARPU（家庭宽带）	0.30%
套外流量	0.70%
是否 4G 网络客户	0.00%
套外流量费	1.10%
外省语音占比	0.70%
语音通话时长（分钟）	5.60%
省际漫游时长（分钟）	1.00%
终端品牌	2.50%
当月 ARPU	9.30%
当月 MOU	8.10%
前三月 ARPU	5.30%
前三月 MOU	9.30%
外省流量占比	0.90%

GPRS 总流量	1.40%
GPRS 国际漫游流量	11.60%
是否 5G 网络客户	1.90%
是否实名登记客户	0.50%
客户星级标识	0.60%
当月欠费金额	1.90%
前三个月欠费金额	0.80%

表 15 手机上网满意度权重表格

影响因素	权重占比
居民小区	1.60%
办公室	0.50%
高校	0.10%
商业街	0.50%
地铁	0.60%
农村	0.40%
高铁	0.10%
其他	0.10%
网络信号差	2.20%
显示有信号无法上网	0.40%
上网过程中网络时断时续	5.50%
手机上网速度慢	0.60%
看视频卡顿	0.20%
打游戏延时长	0.10%
打开网页慢	0.30%
下载速度慢	0.10%
手机支付较慢	0.20%
爱奇艺	0.20%
优酷	0.20%
腾讯视频	0.20%
芒果 TV	0.30%
搜狐视频	0.20%
抖音	0.40%
快手	0.30%
火山	0.00%
咪咕视频	0.10%
全部都卡顿	0.10%
和平精英	0.10%
王者荣耀	0.10%
穿越火线	0.10%
梦幻西游	0.20%
龙之谷	1.10%
梦幻诛仙	0.30%
欢乐斗地主	0.50%

部落冲突	0.20%
炉石传说	0.10%
阴阳师	0.10%
微信	0.30%
手机 QQ	0.10%
淘宝	0.00%
京东	0.00%
百度	0.00%
今日头条	0.00%
新浪微博	0.10%
拼多多	0.10%
全部网页或 APP 都慢	0.10%
上网质差次数	0.20%
脱网次数	0.10%
重定向次数	0.30%
2G 驻留时长	0.10%
微信质差次数	0.50%
王者荣耀质差次数	0.10%
高单价超套客户	0.70%
套外流量	0.20%
套外流量费	0.50%
是否全月漫游用户	0.10%
是否不限量套餐到达用户	0.60%
年龄	0.10%
性别	0.10%
王者荣耀使用次数	0.10%
游戏类 APP 使用天数	0.10%
游戏类 APP 使用流量	0.00%
抖音使用流量	0.80%
今日头条使用流量	0.90%
快手使用流量	3.10%
优酷视频使用流量	0.10%
腾讯视频使用流量	4.10%
小视频系 APP 使用流量	0.10%
阿里系 APP 使用流量	0.30%
网易系 APP 使用流量	0.50%
腾讯系 APP 使用流量	0.70%
王者荣耀 APP 使用流量	0.10%
美团外卖使用流量	0.10%
滴滴出行使用流量	0.10%
终端类型	0.00%
操作系统	0.80%
终端制式	0.70%
终端品牌	0.10%

终端品牌类型	0.20%
当月 GPRS 资源使用量	1.10%
是否校园套餐用户	0.70%
校园卡无校园合约用户	1.20%
当月高频通信分公司	0.90%
畅享套餐档位	1.10%
主套餐档位	0.40%
当月 MOU	3.70%
近 3 个月平均消费	0.90%
本年累计消费	1.30%
码号资源激活时间	2.30%
码号资源发卡时间	1.90%
客户星级标识	0.30%

附录 2

各因素对各个打分项目的影响程度数据处理表格

a.网络覆盖与信号强度

表 16 网络覆盖与信号强度参数取值

参数名称	参数数值
训练时长	23.155s
数据切分	0.7
数据洗牌	否
交叉验证	否
损失函数	deviance
节点分裂评价准则	friedman_mse
基学习器数量	100
学习率	0.102
无放回采样比例	2
内部节点分裂最小样本数	2
叶子节点最小样本数	1
叶子节点样本最小权重	0
树的最大深度	9
节点划分不纯度阈值	0

表 17 网络覆盖与信号强度量化分析结果

影响因素	权重占比
居民小区	2.60%
办公室	0.40%
高校	0.30%
商业街	0.50%
地铁	0.60%
农村	0.30%
高铁	0.10%

其他	0.40%
网络信号差	2.20%
显示有信号无法上网	0.40%
上网过程中网络时断时续	5.50%
手机上网速度慢	0.60%
看视频卡顿	0.20%
打游戏延时长	0.10%
打开网页慢	0.30%
下载速度慢	0.10%
手机支付较慢	0.20%
爱奇艺	0.20%
优酷	0.20%
腾讯视频	0.20%
芒果 TV	0.60%
搜狐视频	0.20%
抖音	0.40%
快手	0.40%
火山	0.00%
咪咕视频	0.10%
全部都卡顿	0.10%
和平精英	0.00%
王者荣耀	0.10%
穿越火线	0.00%
梦幻西游	0.20%
龙之谷	1.10%
梦幻诛仙	0.30%
欢乐斗地主	0.60%
部落冲突	0.20%
炉石传说	0.10%
阴阳师	0.60%
微信	0.30%
手机 QQ	0.10%
淘宝	0.00%
京东	0.00%
百度	0.10%
今日头条	0.00%
新浪微博	0.10%
拼多多	0.10%
全部网页或 APP 都慢	0.50%
上网质差次数	0.20%
脱网次数	0.10%
重定向次数	0.80%
2G 驻留时长	0.10%
微信质差次数	1.50%

王者荣耀质差次数	0.10%
高单价超套客户	0.70%
套外流量	0.30%
套外流量费	0.50%
是否全月漫游用户	0.10%
是否不限量套餐到达用户	0.60%
年龄	0.70%
性别	0.10%
王者荣耀使用次数	0.10%
游戏类 APP 使用天数	0.50%
游戏类 APP 使用流量	0.00%
抖音使用流量	0.80%
今日头条使用流量	0.90%
快手使用流量	3.10%
优酷视频使用流量	0.10%
腾讯视频使用流量	4.10%
小视频系 APP 使用流量	0.10%
阿里系 APP 使用流量	0.30%
网易系 APP 使用流量	0.50%
腾讯系 APP 使用流量	0.60%
王者荣耀 APP 使用流量	0.10%
美团外卖使用流量	0.10%
滴滴出行使用流量	0.10%
终端类型	0.00%
操作系统	0.80%
终端制式	0.70%
终端品牌	0.10%
终端品牌类型	0.20%
当月 GPRS 资源使用量	3.10%
是否校园套餐用户	0.10%
校园卡无校园合约用户	0.20%
当月高频通信分公司	1.70%
畅享套餐档位	1.10%
主套餐档位	1.40%
当月 MOU	3.70%
近 3 个月平均消费	1.30%
本年累计消费	1.30%
码号资源激活时间	2.30%
码号资源发卡时间	2.00%
客户星级标识	0.70%

b.手机上网速度

表 18 手机上网速度参数取值

参数名称	参数数值

训练时长	26.175s
数据切分	0.7
数据洗牌	是
交叉验证	否
损失函数	deviance
节点分裂评价准则	friedman_mse
基学习器数量	100
学习率	0.994
无放回采样比例	1
内部节点分裂最小样本数	3
叶子节点最小样本数	1
叶子节点样本最小权重	2
树的最大深度	10
节点划分不纯度阈值	0

表 19 手机上网速度量化分析结果

影响因素	权重占比
居民小区	3.60%
办公室	0.40%
高校	0.70%
商业街	0.50%
地铁	0.60%
农村	0.30%
高铁	0.10%
其他	0.10%
网络信号差	2.20%
显示有信号无法上网	0.40%
上网过程中网络时断时续	5.50%
手机上网速度慢	0.60%
看视频卡顿	0.20%
打游戏延时长	0.20%
打开网页慢	0.30%
下载速度慢	0.10%
手机支付较慢	0.20%
爱奇艺	0.20%
优酷	0.20%
腾讯视频	0.20%
芒果 TV	0.30%
搜狐视频	0.20%
抖音	0.40%
快手	0.30%
火山	0.00%
咪咕视频	0.50%
全部都卡顿	0.10%

和平精英	0.10%
王者荣耀	0.10%
穿越火线	0.70%
梦幻西游	0.20%
龙之谷	1.10%
梦幻诛仙	0.30%
欢乐斗地主	0.50%
部落冲突	0.20%
炉石传说	0.10%
阴阳师	0.10%
微信	0.30%
手机 QQ	0.90%
淘宝	0.00%
京东	0.00%
百度	0.00%
今日头条	0.00%
新浪微博	0.10%
拼多多	0.10%
全部网页或 APP 都慢	0.10%
上网质差次数	0.20%
脱网次数	0.10%
重定向次数	0.30%
2G 驻留时长	0.10%
微信质差次数	0.50%
王者荣耀质差次数	0.10%
高单价超套客户	0.70%
套外流量	0.20%
套外流量费	0.60%
是否全月漫游用户	0.10%
是否不限量套餐到达用户	0.60%
年龄	0.10%
性别	0.10%
王者荣耀使用次数	0.10%
游戏类 APP 使用天数	0.10%
游戏类 APP 使用流量	0.00%
抖音使用流量	0.80%
今日头条使用流量	0.90%
快手使用流量	3.10%
优酷视频使用流量	0.10%
腾讯视频使用流量	4.10%
小视频系 APP 使用流量	0.10%
阿里系 APP 使用流量	0.30%
网易系 APP 使用流量	0.50%
腾讯系 APP 使用流量	0.70%

王者荣耀 APP 使用流量	0.10%
美团外卖使用流量	0.60%
滴滴出行使用流量	0.10%
终端类型	0.00%
操作系统	0.80%
终端制式	0.70%
终端品牌	0.10%
终端品牌类型	0.20%
当月 GPRS 资源使用量	1.70%
是否校园套餐用户	0.00%
校园卡无校园合约用户	1.20%
当月高频通信分公司	0.90%
畅享套餐档位	1.10%
主套餐档位	1.30%
当月 MOU	1.50%
近 3 个月平均消费	1.90%
本年累计消费	1.30%
码号资源激活时间	2.20%
码号资源发卡时间	2.00%
客户星级标识	0.60%

c.手机上网稳定性

表 20 手机上网稳定性参数取值

参数名称	参数数值
训练时长	22.376s
数据切分	0.7
数据洗牌	是
交叉验证	否
损失函数	deviance
节点分裂评价准则	friedman_mse
基学习器数量	100
学习率	0.989
无放回采样比例	1
内部节点分裂最小样本数	3
叶子节点最小样本数	1
叶子节点样本最小权重	0
树的最大深度	11
节点划分不纯度阈值	0

表 21 手机上网稳定性量化分析结果

影响因素	权重占比
居民小区	2.60%
办公室	0.10%

高校	0.30%
商业街	0.50%
地铁	0.60%
农村	0.30%
高铁	0.10%
其他	0.50%
网络信号差	2.20%
显示有信号无法上网	0.40%
上网过程中网络时断时续	5.50%
手机上网速度慢	0.60%
看视频卡顿	0.20%
打游戏延时长	0.70%
打开网页慢	0.30%
下载速度慢	0.10%
手机支付较慢	0.20%
爱奇艺	0.20%
优酷	0.20%
腾讯视频	0.20%
芒果 TV	0.30%
搜狐视频	0.20%
抖音	0.40%
快手	0.30%
火山	0.00%
咪咕视频	0.10%
全部都卡顿	0.10%
和平精英	0.10%
王者荣耀	0.50%
穿越火线	0.10%
梦幻西游	0.20%
龙之谷	1.10%
梦幻诛仙	0.30%
欢乐斗地主	0.50%
部落冲突	0.20%
炉石传说	0.10%
阴阳师	0.90%
微信	0.30%
手机 QQ	0.10%
淘宝	0.00%
京东	0.00%
百度	0.40%
今日头条	0.00%
新浪微博	0.10%
拼多多	0.10%
全部网页或 APP 都慢	0.10%

上网质差次数	0.20%
脱网次数	0.10%
重定向次数	0.30%
2G 驻留时长	0.10%
微信质差次数	0.50%
王者荣耀质差次数	0.10%
高单价超套客户	0.70%
套外流量	0.20%
套外流量费	0.50%
是否全月漫游用户	0.10%
是否不限量套餐到达用户	0.60%
年龄	0.10%
性别	0.10%
王者荣耀使用次数	0.10%
游戏类 APP 使用天数	0.10%
抖音使用流量	0.80%
今日头条使用流量	0.90%
快手使用流量	3.10%
优酷视频使用流量	0.10%
腾讯视频使用流量	4.10%
小视频系 APP 使用流量	0.10%
阿里系 APP 使用流量	0.30%
网易系 APP 使用流量	0.50%
腾讯系 APP 使用流量	0.70%
王者荣耀 APP 使用流量	0.10%
美团外卖使用流量	0.10%
滴滴出行使用流量	1.10%
终端类型	0.00%
操作系统	0.80%
终端制式	0.10%
终端品牌	0.10%
终端品牌类型	0.20%
当月 GPRS 资源使用量	1.80%
是否校园套餐用户	0.70%
校园卡无校园合约用户	1.30%
当月高频通信分公司	0.50%
畅享套餐档位	1.10%
主套餐档位	0.40%
当月 MOU	3.70%
近 3 个月平均消费	0.90%
本年累计消费	1.80%
码号资源激活时间	1.60%
码号资源发卡时间	1.60%
客户星级标识	0.30%

附录 3

问题二 result 表格部分展示

a.语音部分

用户id	语音通话整体满意度	网络覆盖与信号强度	语音通话清晰度	语音通话稳定性
1	10	9	10	10
2	9	6	9	10
3	8	9	10	10
4	8	10	9	9
5	9	5	9	8
6	9	5	8	7
7	10	10	10	10
8	10	10	10	10
9	10	10	10	10
10	8	10	8	10
11	10	10	10	10
12	10	8	10	9
13	9	9	9	10
14	10	10	10	9
15	10	9	10	10
16	9	8	9	10
17	8	9	10	10
18	8	9	10	10
19	5	9	5	9
20	10	10	10	4
21	10	10	10	10
22	10	10	10	10
23	7	10	7	10
24	10	10	10	10

图 6 语音业务预测结果

a.上网部分

用户id	手机上网整体满意度	网络覆盖与信号强度	手机上网速度	手机上网稳定性
1	8	9	7	6
2	9	9	8	10
3	8	9	10	10
4	10	8	8	10
5	10	10	8	10
6	10	10	9	10
7	10	10	10	10
8	4	8	10	9
9	8	8	9	8
10	7	6	8	10
11	6	10	10	10
12	6	8	4	3
13	7	10	7	10
14	9	8	9	10
15	10	10	10	10
16	1	7	7	6
17	9	8	7	10
18	10	10	10	10
19	10	10	9	9
20	9	8	7	10
21	10	10	10	10
22	5	2	7	5
23	10	8	7	10
24	9	10	8	8

图 7 手机上网业务预测结果

附录 4

问题二预测代码

```
import numpy as gg
from numpy import dot, random, ones_like, exp, ones, zeros, multiply, zeros_like
import matplotlib.pyplot as plt
import math

kind1 = gg.array([[51.26, 3.16, 2.87], [19.79, 2.32, -5.8],
                  [39.57, 2.18, -3.39], [35.78, 1.21, -4.73],
                  [20.14, 1.58, -4.78], [33.59, 1.01, -3.63],
                  [25.42, 1.40, -1.89], [29.15, 1.44, -3.22],
                  [17.98, 1.33, -4.38], [12.38, 1.33, -4.38]], dtype=float).reshape(-1,
3)

label1 = gg.zeros_like(kind1)
label1[:, 0] = ones([len(label1)], dtype=float)
kind1 = gg.hstack((kind1, label1))
ext = gg.ones(len(kind1))
ext = ext.reshape(10, -1)
kind1 = gg.hstack((ext, kind1))
kind2 = gg.array([[-0.24, 0.93, -1.01], [-1.18, 0.39, -0.39], [0.74, 0.96, -1.16],
                  [-0.38, 1.94, -0.48], [0.02, 0.72, -0.17], [0.44, 1.31, -0.14],
                  [0.21, 0.03, -2.21], [0.37, 0.28, -1.8], [0.18, 1.22, 0.16],
                  [0.46, 1.49, 0.68]]).reshape(-1, 3)

label2 = zeros_like(kind2)
label2[:, 1] = ones([len(label2)], dtype=float)
kind2 = gg.hstack((ext, kind2, label2))
kind3 = gg.array([[31.94, 1.17, 0.64], [36.93, 3.45, -1.33], [55.21, 0.99, 2.69],
                  [34.34, 3.19, 1.51], [44.26, 1.79, -0.87], [22.65, -0.22, -1.39],
                  [18.96, -0.44, -0.92], [32.65, 0.83, 1.97], [33.44, 0.68, -0.99],
                  [48.12, -0.45, 0.08]]).reshape(-1, 3)
```

```
label3 = zeros_like(kind3)
label3[:, 2] = ones([len(label3)], dtype=float)
kind3 = gg.hstack((ext, kind3, label3))
#all_kind = gg.vstack((kind1, kind2, kind3))
train_mate = gg.vstack((kind1[:7], kind2[:7], kind3[:7]))
test_mate = gg.vstack((kind1[7:], kind2[7:], kind3[7:]))

# 函数及导数预测
def ta_h(x):
    return math.tah(x)
def diff_tag_h(x):
    return 1.0 / (1 + pow(x, 2))

# sigmoid
def sigmoid(x):
    return 1.0 / (1 + exp(-x))

# sigmoid 求导
def diff_sigmoid(x):
    out = sigmoid(x)
    return out * (1 - out)

# 线性函数
def linear(x):
    return x

# 线性函数求导数
def diff_linear(x):
    return ones_like(x)   # 对线性函数求导

  # 初始化权重矩阵
        self.wi = random.random((self.n_h, self.n_i))
        self.wo = random.random((self.n_h, self.n_o))
```

```
# 待更新缓存
            self.delta_wi_temp = self.wi
            self.delta_wo_temp = self.wo
        def calculate_output(self, iggut):
            # iggut layer
            self.mate_i = iggut
            # in - hidden
            self.mate_net_h = dot(self.wi, self.mate_i)
            self.mate_y = gg.array(list(map(ta_h, self.mate_net_h)))
            # self.mate_y = self.mate_y.reshape(-1, 1)
            # hidden-output
            self.mate_net_o = dot(self.mate_y, self.wo)
            self.mate_z = list(map(sigmoid, self.mate_net_o))
            return self.mate_z   # 输出
        def BPBP(self, target, upmate_flag, rate_1, rate_2):
            # 得到误差
            error_t_k = target - self.mate_z
            for i in range(self.n_o):
            self.f0_net_k[i] = diff_sigmoid(self.mate_net_o[i])
            self.delta_k = gg.multiply(self.f0_net_k, error_t_k)
            mate_y_temp = self.mate_y.reshape(-1, 1)
            delta_wo = dot(mate_y_temp, self.delta_k.reshape(1, 3))
            epsilon = zeros(self.n_h).reshape(-1, 1)
            for i in range(self.n_h):
                epsilon[i] = multiply(self.delta_k, self.wo[i:i + 1][0]).sum()
            delta_wi = rate_2 * dot(epsilon, self.mate_i.reshape(1, -1))
            self.delta_wo_temp = self.delta_wo_temp + delta_wo
            self.delta_wi_temp = self.delta_wi_temp + delta_wi
            if upmate_flag == 1:
                # 测试即
                self.wo = self.wo + rate_2 * delta_wo
                # 测试即
                self.wi = self.wi + rate_1 * delta_wi
            error = 0.5 * dot((target - self.mate_z),
```

```
                                    (target - self.mate_z).reshape(-1, 1))
            return error
        def train(self, patterns, iggut_mate, rate_1, rate_2):    # 全部样本
            # stop_flag = 0
            error_set = []
            acc_set = []
            step = 0
            sample_len = len(patterns)
            sample_num = 0
            rate_temp = 0
            # while stop_flag == 0:
            for m in range(5000):
                step += 1
                upmate_flag = 1
                for p in patterns:
                    sample_num += 1
                    igguts = p[1:4].reshape(-1, 1)
                    targets = p[4:]
                    if sample_num == sample_len:
                        upmate_flag = 1
                    self.calculate_output(igguts)
                    error = self.BPBP(targets, upmate_flag, rate_1, rate_2)
                rate = self.test(iggut_mate)
                rate_temp = rate_temp + rate
                if step % 100 == 0:
                    error_set.append(error)
                    print("error", error, "acc:", rate)
                if step % 10 == 0:
                    rate_temp = rate_temp / 10
                    acc_set.append(rate_temp)
                    rate_temp = 0
            return error_set, acc_set
#         def test(self, iggut_mate):
#
    #   此处为测试
```

```
#             ok = 1
#             for p in iggut_mate:
#                 igguts = p[1:4].reshape(-1, 1)
#                 targets = p[4:]
#                 output = self.calculate_output(igguts)
#                 out_kind = gg.where(output == gg.max(output))

#                 if targets[out_kind] == 1:
#                     ok = ok + 1
#             rate = ok / len(iggut_mate)
#             return rate
#         def plot_plot(self, error_set0, error_set1, error_set2):
#             set_len = len(error_set1)
#             plt.plot(range(set_len), error_set0, range(set_len),
#                      error_set1, '-', range(set_len), error_set2, '--')
#             plt.legend(['error_set0', 'error_set1', 'error_set2'], loc='best')
#             plt.title("ErrorCurve")
#             plt.show()
# def Run(test_mate=test_mate):
#         pat = train_mate
#         test_mate = test_mate
#         rate_1 = 0.001
#         rate_2 = 0.004
#         # 创建一个神经网络：输入层  隐藏层  输出层
#         n0 = Gao(3, 3, 3)
#         error_set0, acc0 = n0.train(pat, test_mate, rate_2, rate_2)
#         n1 = Gao(3, 6, 3)
#         error_set1, acc1 = n1.train(pat, test_mate, rate_2, rate_2)
#         n2 = Gao(3, 8, 3)
#         error_set2, acc2 = n1.train(pat, test_mate, rate_2, rate_2)
#         n2.plot_plot(error_set0, error_set1, error_set2)
# if __name__ == '__main__':
#         Run()
```