上海财经大学

SHANGHAI UNIVERSITY OF FINANCE AND ECONOMICS

《机器学习》期末报告

题目：基于机器学习的电信客户流失预测

姓名：谢嘉薪

学号：2020111142

班级：20 数据科学

目录

摘要

随着客户流失问题关注度日益攀升，如何预测客户流失成为热点问题。基于此背景，本文选取 Kaggle 上的公开电信客户流失数据集的 7043 条电信客户数据及 20 个相关变量，通过建立 Logistic 模型并引入交互项和评分卡模型深入探究影响客户流失的主要特征、深入理解决策树模型并自行编码建立 Adaboost 模型进行预测、利用 R 包建立随机森林和朴素贝叶斯模型进行补充研究，各模型预测效果良好，同时发现各变量中合同期限和客户留存月份对流失情况影响最大；此外，基于上述研究，本文着重对 Logistic 模型给出应用结论，并提出本次研究不足之处，如尝试集成方法、处理类别不均等。

关键词：客户流失 逻辑回归 评分卡 Adaboost 随机森林 朴素贝叶斯

一、引言

（一）研究背景与研究问题

当今社会市场竞争激烈，同行业间同质化严重，客户可选择性越来越多，从而使得新增客户成本增加。研究表明发展一个新顾客成本远远超过维护老客户的成本，因此分析客户流失是各行业不可避免的话题。本文聚焦于电信客户流失预测，以帮助电信公司查寻客户流失原因，以及对即将流失的客户采取相应补救措施，从而降低客户流失率，大大提高企业利润。

电信客户流失情况有很多影响因素，如消费者个体因素、注册服务类型、账户情况等，探究客户流失的主要因素，需要建立模型与实证研究。余路（2016）运用逻辑回归、决策树、神经网络及组合模型进行客户流失预测；杨婷等（2015）提出改进贝叶斯算法来研究电信客户流失问题等，均为本文进行电信客户流失预测提供了关键思路。

在此背景下，本文选取 Kaggle 上的电信客户流失数据，通过建立 Logistic 模型、Adaboost 模型、朴素贝叶斯模型，进行客户流失预测。其中 Logistic 模型中引入评分卡模型，通过 IV 值探究各特征重要性并建立评分卡用于预测，随机森林则利用 R 包直接输出特征重要性排名并进行可视化。此外，为深入理解决策树模型，自行编码实现决策树和 Adaboost 进行建模和预测。根据机器学习的方法，基于电信客户大数据，运营商便能预测客户流失的可能性大小，及时获取“待流失”的客户名单，可尽早通过提供优惠等途径提高用户粘性，挽回用户，降低客户流失率。

（二）数据获取与变量说明

本文数据集来自 Kaggle 平台的公开数据——Telco Customer Churn[1]，它包括 7043 条电信客户的数据、20 个特征、1 个分类标签，其中特征有一列为样本编号，无意义，予以剔除，剩余 16 列特征为分类型变量，3 列为数值型变量；分类标签“客户流失（Churn）”为本文的目标变量，分为“流失客户”和“现有客户（非流失）”两类，代表上个月的客户流失情况。

该数据集中含 11 个缺失值，相对样本可忽略不计，因此本文选择对缺失值所在的样本进行剔除。此外，客户注册服务类型变量分电话服务和互联网服务两方面，其中细分变量如“是否有多条线路”建立在有电话服务之上，“在线安全服务”、“在线备份”、“设备保护”、“技术支持”、“媒体电视服务”、“媒体电影服务”建立在互联网服务之上，以上变量均有三个水平：“是”、“否”、“无电话/互联网服务”，其中“无电话/互联网服务”直接导致无对应细分服务内容，为便于后续建模和分析，将其与“否”合并为同一水平。综上，变量说明如表 1 所示。

表 1 变量说明

指标类型	变量名称	变量含义	取值情况
自变量	Gender	性别	男，女
客户个人信息	Senior Citizen	是否是老人	是，否

[1] 数据来源：https://www.kaggle.com/datasets/blastchar/telco-customer-churn

	Partner	是否有朋友	是，否
	Dependents	是否有家属	是，否
自变量 客户注册服务	Phone Service	是否有电话服务	是，否
	Multiple Lines	是否有多条线路	是，否
	Internet Service	互联网服务类型	无，拨号上网，光纤
	Online Security	是否有在线安全服务	是，否
	Online Backup	是否有在线备份服务	是，否
	Device Protection	是否有设备保护服务	是，否
	Tech Support	是否有技术支持服务	是，否
	Streaming TV	是否有媒体电视服务	是，否
	Streaming Movies	是否有媒体电影服务	是，否
自变量 客户账户信息	Tenure	客户留存月数	1-72
	Contract	合同期限类型	逐月，一年期，两年期
	Paperless Billing	是否有纸质账单	是，否
	Payment Method	付款方式	银行转账（自动） 信用卡转账（自动） 电子支票 纸质支票
	Monthly Charges	每月收取金额	18.25-118.75
	Total Charges	向客户收取的总金额	18.8-8684.8
因变量	Churn	是否流失	是，否

二、探索性数据分析

本文研究的自变量包括 16 个分类变量和 3 个连续变量，为判断自变量对因变量是否有影响，绘制分类变量的堆积图并对其依次做卡方检验，若拒绝原假设，表明自变量与因变量有显著的相关性，即自变量会对因变量有影响。连续变量则绘制箱线图并进行单因素方差分析。

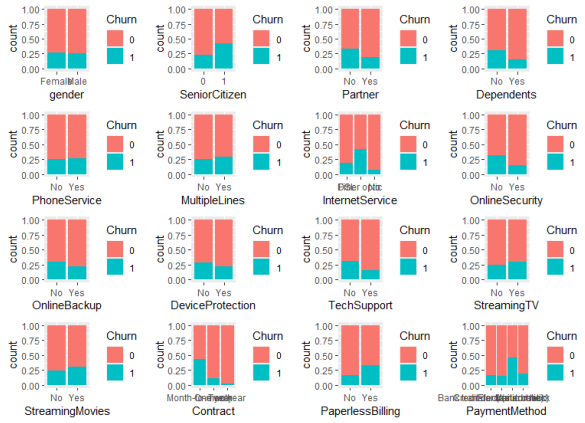

图 1 分类变量与因变量的条形图

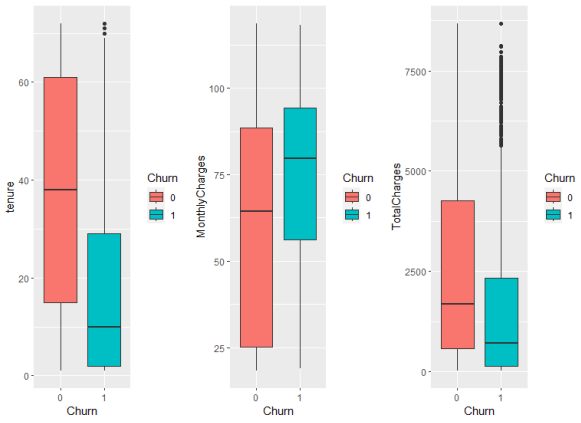

图 2 连续变量与因变量的箱线图

对于分类变量，观察图 1 看出，不同性别的流失客户占比、是否有电话服务的流失客户占比非常相近，即这两个因素几乎不会对客户流失产生影响，推测是因为电信服务属于基础服务，不存在审美等存在男女差异的属性，而电话服务附加价值较低，且在互联网时代逐渐可有可无。相反，合同期限类型为两年付的客户流失占比远低于月付的，是因为年费客户一方面出于违约成本不更换运营商，形成绑定关系，另一方面越忠诚的客户越倾向于订购长期合同，符合基础认知；无网络服务的用户流失概率远低于有光纤服务的用户，这是因为对于有网络需求的客户来说，不同运营商的网络服务质量和优惠力度等众多因素都会造成流动，而不需要网络的客户则相对流动性较弱。

对于连续变量，观察图 2 可以看出，在因变量的两个水平上，连续变量的平均水平差异均较大，即都会对客户是否流失产生一定影响，其中流失客户的留存月份平均水平较低，考虑到观测流失情况的时点均为上个月，初步判断新用户更容易流失，可能是行业同质化所致。

为使后续模型建立和分析更有效，对卡方检验和单因素方差分析中 P-value>0.001 的变量予以剔除。各变量的 P-value 如图 3 所示，基于此剔除性别、是否有电话服务两个变量，其次是否有多条线路是基于电话服务的具体内容，若不研究电话服务则该变量也无研究意义，予以剔除。

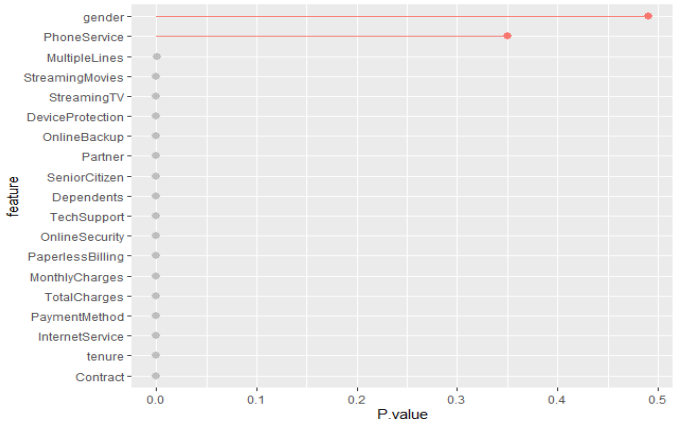

图 3 分类变量与因变量的卡方检验/方差分析

三、机器学习模型

（一）Logistic 回归模型

1、模型原理简述

假设有数据集$\{(y_i, x_i)\}_{i=1}^n$，其中$y_i \in \{0,1\}$，Logistic 回归尝试估计$p(y_i = 1|x_i) \triangleq p(x_i)$。考虑线性组合$w^T x_i + b$，一个实际的问题是：$p(x_i)$的取值范围为(0,1)，而$w^T x_i + b$可能取到一切实数，即它们的值域不同，考虑 Logit 变换：

$$logit\big(p(x_i)\big) = \log\frac{p(x_i)}{1-p(x_i)}, i = 1, \dots, n$$

直接计算可得：

$$p(x_i) = \frac{1}{1+e^{-x_i^T\beta}}$$

通常将$p(x_i) > 0.5$的实例划分为正例，反之为反例，随阈值变化，预测的各项指标会随之变化，为充分考虑 AUC、敏感性、特异性，本文以 ROC 曲线输出最优分类阈值为准，后续模型同理。

2、数据分箱与 WOE 编码

对于连续变量，Logistic 回归模型返回的结果是概率值，但连续变量增加时概率值未必随之线性变动。分箱后的特征对异常数据更具有鲁棒性，不会因数据错误而对预测结果造成很大的干扰；分箱后的特征在一定程度上解决了连续变量的变化对事故严重率变动非线性的问题，且离散特征的可操作性和实际意义更强；另外，特征离散化后简化了 Logistic 回归模型，降低了过拟合风险。对于离散变量（包括分箱后的连续变量），直接进行独热编码容易造成维数灾难，降低模型效果，因此本部分引入评分卡模型中的 WOE 编码。

（1）一维高斯混合模型原理与求解算法

设有一维样本集$\{x_i\}_{i=1}^n$，，高斯混合模型采用混合高斯分布来刻画数据原型，使用原型对应的后验概率确定样本点所属的簇。具体来说，假设$\{x_i\}_{i=1}^n$满足

$$x_i \sim \sum_{k=1}^{M} \pi_k N(\mu_k, \sigma_k^2)$$

其中 M 是一个事先确定的数。设参数$\boldsymbol{\theta}$=($\boldsymbol{\pi}$,$\boldsymbol{\mu}$,$\boldsymbol{\sigma}^2$)，z_{ik}为x_i是否来自第 k 类的示性变量，则π_k是x_i属于第 k 类的先验概率，即$p(z_{ik} = 1|\theta) = |\pi_k$。由贝叶斯公式可算出后验概率：

$$p(z_{ik} = 1|\theta) = \frac{p(x_i|z_{ik} = 1, \theta)p(z_{ik} = 1|\theta)}{p(x_i|\theta)} = \frac{N(x_i|\mu_k, \sigma_k^2)\pi_k}{\sum_{k=1}^{M}\pi_k N(\mu_k, \sigma_k^2)} \triangleq \gamma_{ik}$$

则得到参数θ的估计后，就可以得到样本x_i所属的类别k'：$k'=argmax\gamma_{ik}$。

为估计θ，使用 EM 算法，算法步骤如下：

Step1：设置初值$\theta^0=(\pi^0,\mu^0,\sigma^{2^0})$，flag=1，t=0

Step2：检查收敛准则是否满足，即$flag < 10^{-6}$，若满足则结束

Step3：计算γ_{ik}^t，计算$\mu_k^{t+1}=\frac{\sum_{i=1}^n \gamma_{ik}^t x_i}{\sum_{i=1}^n \gamma_{ik}^t}$，$\sigma_k^{2^{t+1}}=\frac{\sum_{i=1}^n \gamma_{ik}^t (x_i-\mu_k^{t+1})^2}{\sum_{i=1}^n \gamma_{ik}^t}$，$\pi_k^{t+1}=\frac{1}{n}\sum_{i=1}^n \gamma_{ik}^t$

Step4：$flag=\|\theta^{t+1}-\theta^t\|^2$，t=t+1，回到 Step2

得到参数估计后即可确定每个样本点所属类别。一维高斯混合模型的 R 语言代码实现见附录 C

（2）WOE 编码介绍

记因变量为$y_i \in \{0,1\}$，$y=0$ 的频数为$N0$，$y=1$ 的频数为$N1$，对于一个分类变量X_i，设它有n_i个水平，在第 K 个水平上$y=0$ 的频数为$N_{K_0}^{X_i}$，$y=1$ 的频数为$N_{K_1}^{X_i}$，可定义该变量在该水平上的 WOE 值，并算出对应 IV 值：

$$WOE(X_i)=ln\left(\frac{N_{K_1}^{X_i}/N_{K_0}^{X_i}}{N_1/N_0}\right)$$

$$IV=\sum_i^n\left(\frac{N_{K_1}^{X_i}}{N_{K_0}^{X_i}}-\frac{N_1}{N_0}\right)WOE(X_i)$$

WOE 值刻画了该变量的当前水平对因变量起到的影响的方向和大小，WOE 值大于 0 表明当前水平$y=1$ 的比例超过整体比例，而 WOE 值越大，表明在当前水平中$y=1$ 的比例越大。

计算得到 WOE 值后，用该变量该水平下的 WOE 值替代原值，即用 WOE 值编码，将该变量转换成一个数值型变量。WOE 编码最大的作用在于：它巧妙地给自变量的每个水平赋予一个数值，能够使得该变量每增加一个单位，因变量的优势增加相同的量，解决了线性性问题。因此 WOE 编码后可以不做哑变量编码，避免了维数灾难，且与研究表明 WOE 编码可能会提高模型的预测效果（与因变量和自变量的线性性有关）。另外，WOE 编码后的值可以直接比较，无需标准化，也有利于变量间相关性的分析。IV 值则是一种衡量自变量预测能力的指标，我们可以根据 IV 值对变量进行排序，观察每个变量的预测能力大小。WOE 编码的 R 语言代码实现见附录 C

（3）具体实现

高斯混合模型要求事先给定聚类簇数，一般而言，最优的簇数需要通过轮廓系数等指标来加以选取，但轮廓系数需要计算样本点两两之间的距离，这对于本问题而言是一个巨大的开销，因此本问题对常见的簇数进行了尝试，选择簇数为 4，部分聚类结果如图 4 所示。用样本点所在的类别作为样本点的值，将这三个连续变量变为分类变量，依次与因变量做卡方检验，结果显示全部拒绝原假设（P<0.001），即分箱后的变量与因变量不独立，效果理想。其次是对目前剩余的 16 个分类变量（连续变量已分箱）进行 WOE 编码，用 WOE 值作为自变量相应水平的替代值。基于此绘制相关性矩阵，以初步筛选变量，简洁起见，图 5 中仅展示相关性高的变量。

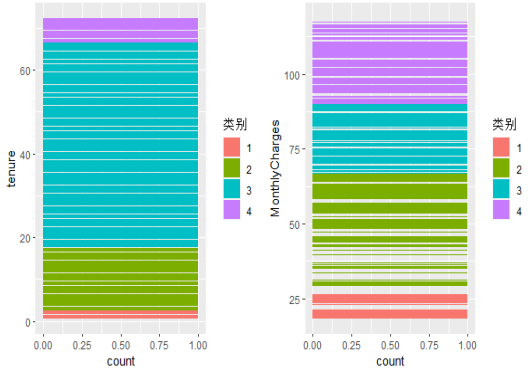

图 4 聚类结果可视化

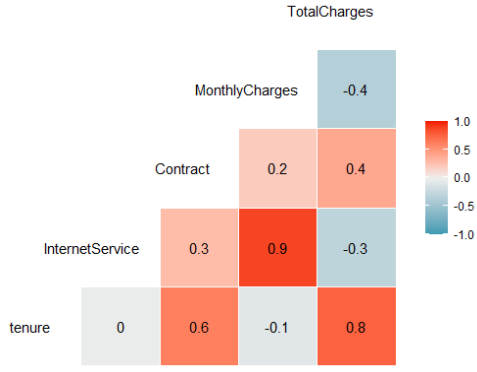

图 5 相关性矩阵（相关性较高变量）

可以发现，互联网服务类型与月缴费高度相关，不难推测是因为网络费用较高导致，符合实际，因此可删去其一，此处保留前者；客户留存时间与总缴费高度相关，但考虑到存在长期留存但服务较少而总缴费较少的特殊情况，本文忽略此共线性，保留二者。至此已对特征进行初步筛选，后续通过建立模型继续充分考虑处理自变量的相关关系。

3、模型建立与评估

（1）模型建立

对目前剩余的 15 个变量，采用全部数据建立 Logistic 回归模型，回归系数见表 2。

表 2 回归系数表

变量	回归系数	变量	回归系数
截距项	-1.04662	DeviceProtection	-0.10956
SeniorCitizen	0.28515	TechSupport	0.28624
Partner	-0.02541	StreamingTV	1.19883
Dependents	0.14530	StreamingMovies	1.27282
tenure	0.44965	Contract	0.49530
InternetService	0.93545	PaperlessBilling	0.39320
OnlineSecurity	0.31940	PaymentMethod	0.24266
OnlineBackup	0.31699	TotalCharges	0.81519

其中较为反常的是，WOE 编码后，WOE 的值越大，$P(y=1)$越大，因此回归系数应该都是非负数，但其中是否有朋友、是否有设备保护服务系数均小于 0，且都很小，推测是自变量列的复共线性导致的，因此将它们删除，再拟合 Logistic 回归模型，调整后的回归系数见附录 B。同时将 IV 值排序并绘制条形图，如图 6 所示。

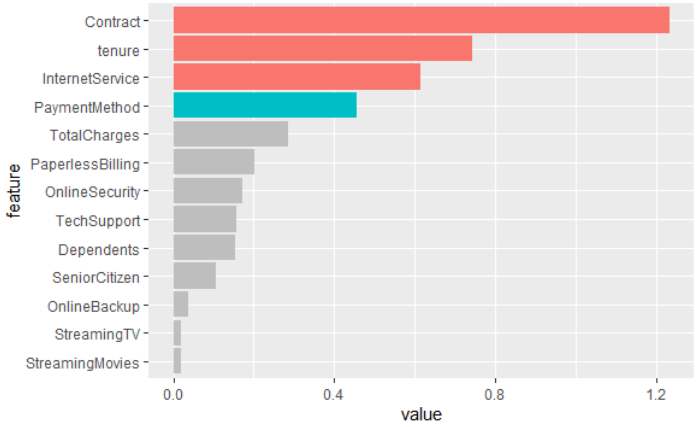

图 6　IV值排序

模型结果可从两方面应用，一是基于评分卡思想，将自变量的 WOE 值与回归系数相乘再扩大 100 倍作为评分卡，见附录 A，将电信客户特征代入表中并把评分相加，总分越高表面其流失的概率越大；二是观察IV值，IV值介于 0.3~0.49 之间说明具有高预测能力，≥0.5 说明具有极高的预测能力。根据上图结果，合同期限、留存月份、互联网服务类型具有极高的预测能力，支付方式具有高预测能力。因此这 4 个指标可以作为预测客户流失的重要因素，需重点关注。

（2）模型评估

为对模型的分类预测能力做出评估，将数据集按 7:3 的比例划分为训练集和测试集 [2]，对训练集进行数据分箱、WOE 编码、拟合 Logistic 模型，再用训练集的分享结果对测试集分箱，用训练集的 WOE 映射规则给测试集编码，完成后代入模型评估预测效果，其核心是保存训练集与测试集的独立性。为充分考虑模型的敏感性和特异性，采取绘制 ROC 曲线确定分类阈值，Logistic 模型的测试 ROC 曲线如图 7 所示，AUC 为 0.849，阈值选取 0.289 时最优，从阈值大小可以看出，Logistic 回归模型输出的是概率值，通过调整分类阈值能够在一定程度上缓解样本类不平衡问题。最优阈值下的测试混淆

[2] 本文所有随机步骤均设定种子 2020111142

矩阵如表 3 所示，此时准确率为 0.772，召回率为 0.803，特异性为 0.760，分类效果良好。

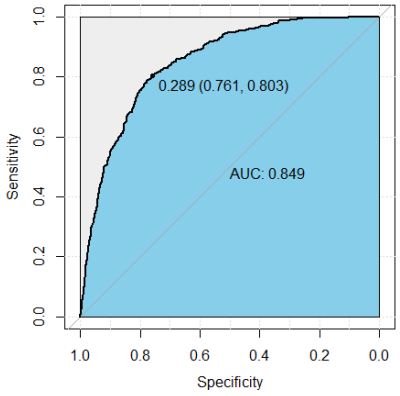

图 7　Logistic 测试 ROC 曲线

表 3 Logistic 测试混淆矩阵

混淆矩阵		真实值	
		1	0
预测值	1	457	370
	0	112	1171

（二）Adaboost 模型

1、决策树算法

（1）决策树原理

建立决策树的主要是作为 Adaboost 算法的基学习器，由于 Adaboost 中的树深度很浅，因此本文中的树的深度很浅，因此本文构建的树不考虑剪枝。另外，由于算力限制，本文建立的树是二叉的。

考虑数据集$\{(y_i, x_i)\}_{i=1}^{n}$，其中$y_i \in \{0,1\}$，每个样本点的权重为w_i，决策树是一种非线性模型，它通过对训练集的学习将样本空间划分为多个不相交的区域，建立树形结构用于分类，其中每个内部节点表示一次对属性值的判断，每个叶子节点表示一个结果输出。每个样本从根节点开始，根据节点的判断结果选择分支，直到到达叶子节点，将叶子节点的值作为样本的预测值。构造决策树主要包括如下三个方面：分裂节点的准则、停止分裂的准则、叶子节点所预测的类，而关键在于分裂节点的准则。

由于本文的树无需剪枝，因此所选取的停止分裂的准则较为简单，即当节点上的样本都属于同一个类别，或分裂的信息增益率为零，或达到最大的深度限制时停止分裂。本文选取带样本权重的多数投票方法确定叶子节点所预测的类，具体而言，设节点 t 是一个叶子节点，按照如下准则确定该节点的类：

$$\underset{k}{\operatorname{argmax}} \sum_{i \in t} I(y_i = k) w_i \ , k \in \{-1,1\}$$

就最关键的分裂节点的准则而言，本文采用交叉熵作为不纯度的度量函数，信息增益率作为分裂准则，具体而言，信息增益率作为分裂准则，具体而言，设p_{tk}，$k \in \{-1,1\}$表示节点 t 上$\{y=k\}$所占的比例，即

$$p_{tk} = P(y = k|t) = \frac{\sum_{i \in t} I(y_i = k) w_i}{\sum_{i \in t} w_i}$$

节点 t 的不纯度为：

$$H(t) = -\sum_{k \in \{-1,1\}} p_{tk} log p_{tk}$$

设由节点 t 分裂出的两个子节点为t_L，t_R，t_L和t_R上的样本点占 t 上样本点的加权比例为p_L，p_R，易知：

$$p_L + p_R = 1$$

$$\sum_{k \in \{-1,1\}} p_{t_L k} = 1$$

$$\sum_{k \in \{-1,1\}} p_{t_R k} = 1$$

则节点 t 分裂后的不纯度为：

$$E = p_L H(t_L) + p_R H(t_R)$$

信息增益为：

$$gain = H(t) - E$$

信息增益率为：

$$ratio = -\frac{gain}{p_L \log(p_L) + p_R \log(p_R)}$$

对于分类属性，在属性的水平构成的集合中随机抽取 30 个子集计算信息增益率，即若属性的水平数不超过 4，则对每个组合都计算信息增益率；若属性水平数大于 4，则在所有可能的组合中随机抽取 30 个组合计算信息增益率。对于连续属性，先找出所有可能取值的中点，然后随机抽取 30 个点计算以该点作为分裂节点的信息增益率。这种做法兼顾了节点分裂的最优性与计算成本，提高了树的构建速度，每次选取找到的信息增益率最大的属性位置进行二叉划分，通过递归可快速建立决策树。

（2）决策树实现算法与步骤

本文采用递归的方法，先判断停止分裂的准则是否被满足，再查找最优的分裂节点并进行分裂，之后递归地得到左子树与右子树。本文使用 R 语言实现树模型，树的存储结构为一个列表，其中第一个元素为节点上的条件，第二个元素为左子树，第三个元素为右子树，若条件满足则选择左子树，否则选择右子树。子树也是一个列表，但当它是叶子节点时退化为一个数，表示该叶子节点所预测的类。树的构建算法的伪代码如下：

Tree=DTML(x,y,depth=0,weight)
BEGIN
更新当前层数：depth= depth+1
检查停止分裂准则：Flag= (y 都属于同一个类别或 depth 达到最大深度)
IF Flag Then return(该叶子节点所预测的类)
查找最优分裂点：BestVar=FindPoint(x,y,weight)
判断停止分裂准则：IF BestVar≈0 Then return(该叶子节点所预测的类)
分裂得到 x1,x2,y1,y2,weight1,weight2，分别为自变量，因变量，样本权重
递归得到左右子树：LeftTree= DTML(x1,y1,depth,weight1)
RightTree= DTML(x2,y2,depth,weight2)
输出结果：Result=list(condition= BestVar,LeftTree,RightTree)
Return(Result)
END

最后，在使用决策树预测时，同样使用类似的递归方法实现。R 语言的构建决策树和预测代码实现详见附录 C

2、Adaboost 算法步骤

1988 年，Kearns 等提出了 Boosting 问题，即“弱可学习是否等价于强可学习”，随后 Schapire 通过构造性方法给出了肯定的回答。Adaboost 算法就是一种成熟的 Boosting 算法，它通过依次训练一系列的弱分类器，按照不同的权重将它们结合起来，最终构造出一个强分类器。具体而言，对于二分类问题，每次训练时要求弱分类器的错误率不超过 1/2，训练完成后根据错分率确定该弱分类器的权重，并调整样本权重：给分类错误的样本加大权重，分类正确的样本降低权重。在新的分布上训练下一个弱分类器，如此直到弱分类器个数达到事先确定的上限为止。

就具体实现流程而言，考虑数据集$\{(y_i, x_i)\}_{i=1}^n$，其中$y_i \in \{0,1\}$，第一次学习时将样本权重$w_i^{(1)}$都设为 1/n。在第 k 次学习时，在带权重为$\{w_i^{(k)}\}$的样本上训练弱分类器$G_k(x)$，计算训练误差：

$$err^{(k)} = \sum_{i=1}^{n} w_i^{(k)} I(y_i \neq G_k(x_i))$$

前述的“错误率不超过 1/2”即是要求$err^{(k)}$<0.5。设定弱分类器的权重：

$$\alpha_k = 0.5\log(\frac{1-err^{(k)}}{err^{(k)}})$$

从上式中可以看到$err^{(k)}$越小时，α_k越大，即分类错误率越小的弱分类器在最终的强分类器中的“话语权”越大。但需要注意的是，$err^{(k)}$是在调整过权重后的样本上的错分率，它未必与此弱分类器在原始数据集上的错分率相等，付忠良（2008）对这个问题给出了解答，指出 Adaboost 调整样本权重的目的是确保正确分类样本分布的均匀性，证明了用$err^{(k)}$确定α_k的有效性。确定了弱分类器的权重之后，按如下式子调整样本权重以供下一次学习使用：

$$w_i^{(k+1)} = w_i^{(k)} e^{-y_i \alpha_i G_k(x_i)} \ , i = 1,2,\dots,n$$

$$w_i^{(k+1)} = w_i^{(k+1)} / \sum_{i=1}^{n} w_i^{(k+1)}$$

训练完指定个数（记为 M）的弱分类器后，使用线性加权的方法组合得到最终的集成分类器：

$$f(x) = sign\left(\sum_{i=1}^{M} \alpha_i G_i(x)\right)$$

对于本问题而言，使用深度很浅的决策树作为弱学习器，因为虽然单棵决策树的泛化能力较弱，但一棵很浅的树就很容易满足错分率小于 1/2 的条件。R 语言的 Adaboost 算法实现见附录 C

3、模型建立与评估

按照 Logistic 模型方法划分数据集进行拟合与预测。Adaboost 的参数包括：决策树的最大深度，树的棵树。Adaboost 是一种 Boosting 算法，因此树的深度不需要很大，本文选取最大深度为 3，即分裂 2 次；有研究表明，当使用 Adaboost 的拟合误差达到 0 时，继续增加树的个数，泛化误差还在减小，故树的个数较大时模型效果会较好。受限于算力和时间，本文选取树的个数为 50 棵。

Adaboost 模型的测试 ROC 曲线如图 8 所示，AUC 为 0.845，测试混淆矩阵如表 4 所示，此时准确率为 0.728，召回率为 0.854，特异性为 0.681，分类效果良好。

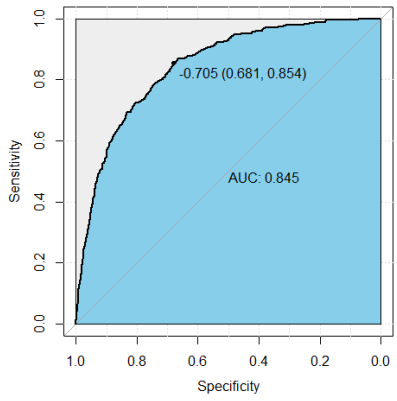

图 8　Adaboost 测试 ROC 曲线

表 4 Adaboost 测试混淆矩阵

混淆矩阵		真实值	
		1	0
预测值	1	486	491
	0	83	1050

（三）随机森林模型

除 Adaboost 外，随机森林也是基于树模型的机器学习算法，属于 Bagging 类集成算法。R 语言中可通过 randomForest 函数建立随机森林模型，并通过 varImpPlot 函数，得到通过拆分特定特征的情况下节点非纯度平均减少量度量的特征重要性排序。

建立模型时，首先寻找最优参数 mtry，即指定节点中用于二叉树的最佳变量个数，通过循环分别构造 mtry 为 1-16 时的随机森林，通过图 9 选取在验证集上预测准确率最高的 mtry=2；其次对于 ntree 的选取，由于不建议过小，设定为 1000，根据图 10 发现随着树个数的增加误差逐渐减小，在 1000 处几乎无变化，故 ntree=1000 合理。

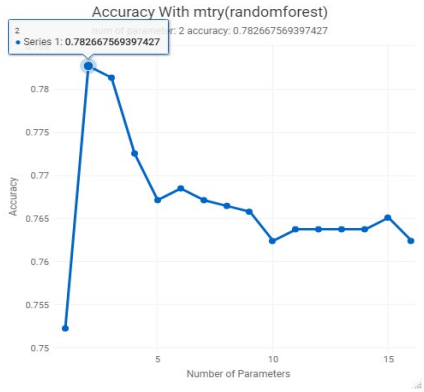

图 9 准确率随 mtry 变化情况

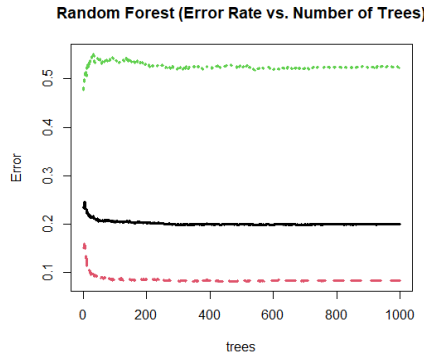

图 10 误差随 ntree 变化情况

通过 varImpPlot 函数，得到如图 11 所示通过拆分特定特征的情况下节点非纯度平均减少量度量的特征重要性排序，可以看出客户留存月份、缴费情况、合同期限重要性均超过 100，说明其对客户流失的影响程度较大。

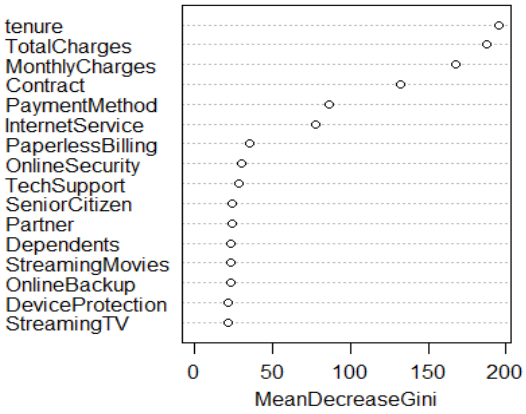

图 11 特征重要性排序（RF）

模型建立完毕后对测试集进行预测，得到的测试 ROC 曲线如图 12 所示，AUC 为 0.842，测试混淆矩阵如表 5 所示，此时准确率为 0.792，召回率为 0.724，特异性为 0.816，分类效果良好。

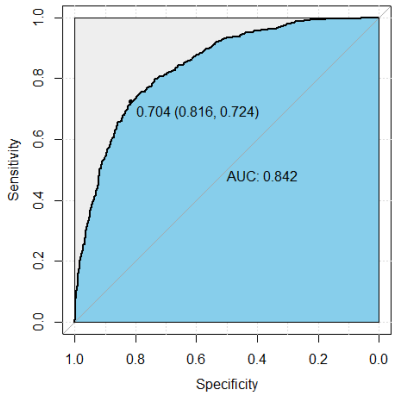

图 12 随机森林测试 ROC 曲线

表 5 随机森林测试混淆矩阵

混淆矩阵		真实值	
		1	0
预测值	1	412	283
	0	157	1258

（四）朴素贝叶斯模型

最为广泛的两种分类模型是决策树模型和朴素贝叶斯模型。和决策树模型相比，朴素贝叶斯分类器发源于古典数学理论，有着坚实的数学基础以及稳定的分类效率，在 R 中可通过 e1071 包中的 naiveBayes 函数拟合朴素贝叶斯模型，其基本语法结构为

$$naiveBayes(formula, data, laplace = 0, \dots, subset, na.action = na.pass)$$

其中$laplace$为 Laplace 平滑处理时给定的值，通过迭代发现 naiveBayes 对参数$laplace$的值不敏感，故不进行调参。模型建立完毕后对测试集进行预测，得到的测试 ROC 曲线如图 13 所示，AUC 为 0.829，测试混淆矩阵如表 6 所示，此时准确率为 0.782，召回率为 0.729，特异性为 0.801，分类效果良好。

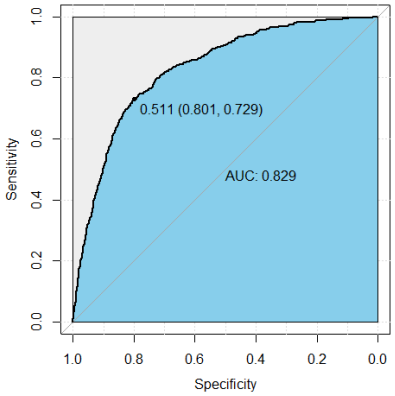

图 13 朴素贝叶斯测试 ROC 曲线

表 6 naiveBayes 测试混淆矩阵

混淆矩阵		真实值	
		1	0
预测值	1	415	307
	0	154	1234

四、结论

（一）模型比较

将本文所用的四个模型在测试集上的 AUC、准确率、召回率、特异性列于表 7，其中所有模型的测试集均相同，类别不均的问题依赖分类阈值改善。

表 7 模型评价指标表

模型	AUC	准确率	召回率	特异性
Logistic	**0.849**	0.772	0.803	0.760
Adaboost	0.845	0.728	**0.854**	0.681
随机森林	0.842	**0.792**	0.724	**0.816**
naiveBayes	0.829	0.782	0.729	0.801

（1）由于研究问题是二分类问题，AUC 往往作为关键评价指标，从上表中可知前三种分类器效果接近，其中 Logistic 效果最佳，且具有最强的解释性，可作为最优分类器。树模型受类别不均影响较大，但其准确率、召回率、特异性表演都较为优异，这一方面与模型本身的优势有关，也与数据集有关。

（2）在特征重要性探索方面，Logistic 回归引入了评分卡模型，具有最大的应用价值，由 Logistic 回归方程和 WOE 的映射规则，不难算出：总分每增加 1 分，客户流失的对数“优势”增加 1%；当总分超过 429 分时，客户流失的预测概率达到 90%以上。随机森林也给出了特征重要性，结论可与评分卡模型相呼应，同时两者均可验证探索性数据分析时的猜想。

（二）研究结论

基于上述对电信客户流失预测模型的研究，发现合同期限、客户留存月份对客户流失影响最大，这提醒运营商可通过对长期合同采取较大优惠力度与客户形成绑定关系，重点关注新用户。其次互联网服务类型也较为重要，对于光纤入户的新客户应给予重点关注。最后，通过评分卡和客户大数据可以预测出客户流失的概率（分数），对于此类客户应及时回馈，采取优惠策略留住该类人群。

（三）不足之处

（1）虽然本文研究了多种分类模型的效果，但采用的是单一模型，可以尝试借助 Stacking 集成方法融合几种模型，探究组合模型的效果是否有显著提升。

（2）受算力限制，Adaboost 的参数非最优，交叉验证的成本也较大，可尝试用 R 自带包进行探索或优化。

（2）由于流失客户占少数，导致存在样本类别不均的情况，对于 Logistic 模型影响较小，但对随机森林的效果有一定限制，可考虑用 SMOTE 采样法进行改进。

参考文献

[1]余路.电信客户流失的组合预测模型[J].华侨大学学报(自然科学版),2016,37(05):637-640.
[2]夏国恩,金炜东.基于支持向量机的客户流失预测模型[J].系统工程理论与实践,2008(01):71-77.
[3]于小兵,曹杰,巩在武.客户流失问题研究综述[J].计算机集成制造系统,2012,18(10):2253-2263.DOI:10.13196/j.cims.2012.10.125.yuxb.017.
[4]陈可.基于B-SMOTE1-XGBoost预测电信客户流失[J].郑州师范教育,2022,11(04):21-26.
[5]冀慧杰,倪枫,刘姜,陆祺灵,张旭阳,阙中力.基于XGB-BFS特征选择算法的电信客户流失预测[J].计算机技术与发展,2021,31(05):21-25.
[6]王雷,陈松林,顾学道.客户流失预警模型及其在电信企业的应用[J].电信科学,2006(09):47-51.
[7]杨婷,滕少华.改进的贝叶斯分类方法在电信客户流失中的研究与应用[J].广东工业大学学报,2015,32(03):67-72.
[8]曹莹,苗启广,刘家辰,高琳.AdaBoost算法研究进展与展望[J].自动化学报,2013,39(06):745-758.

附录

附录 A 评分卡

变量	取值水平	评分
是否是老人	是	19.61
	否	-4.49
是否有家属	是	-9.22
	否	3.12
互联网服务类型	拨号上网	-40.69
	光纤	64.59
	无服务	-141.17
是否有在线安全服务	是	-23.84
	否	7.45
是否有在线备份服务	是	-8.61
	否	4.11
是否有技术支持服务	是	-19.95
	否	6.43
是否有媒体电视服务	是	21.15
	否	-14.12
是否有媒体电影服务	是	21.45
	否	-14.53
客户留存月数	第一类型	61.86
	第二类型	24.86
	第三类型	-21.84
	第四类型	-83.45
合同期限类型	逐月	35.58
	一年期	-51.55
	两年期	-123.78
是否有纸质账单	是	13.13
	否	-24.13
付款方式	银行转账（自动）	-12.26
	信用卡转账（自动）	-16.92
	电子支票	20.03
	纸质支票	-10.19
向客户收取的总金额	第一类型	89.06
	第二类型	23.32
	第三类型	-14.46
	第四类型	-39.44

附录 B 回归系数

变量	回归系数
截距项	-1.04611
SeniorCitizen	0.28835
Dependents	0.13610
tenure	0.44683
InternetService	0.93753
OnlineSecurity	0.31911
OnlineBackup	0.31315
TechSupport	0.28372
StreamingTV	1.21375
StreamingMovies	1.28821
Contract	0.49251
PaperlessBilling	0.39269
PaymentMethod	0.24219
TotalCharges	0.81047

附录 C 代码

```
# 载入相关包

```{r}
library(tidyverse)
library(patchwork)
library(purrr)
```

导入数据

```{r}
rm(list = ls())
data <- read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
```

数据预处理

```{r}
data <- data[,-1] #删除 ID 列
data$Churn <- factor(data$Churn,labels = c("0","1")) #响应变量因子化
sum(is.na(data))
data <- na.omit(data)
summary(data)

# 类别合并
data$MultipleLines[data$MultipleLines == 'No phone service'] <- 'No'
data$OnlineSecurity[data$OnlineSecurity == 'No internet service'] <- 'No'
data$OnlineBackup[data$OnlineBackup == 'No internet service'] <- 'No'
data$DeviceProtection[data$DeviceProtection == 'No internet service'] <- 'No'
data$TechSupport[data$TechSupport == 'No internet service'] <- 'No'
data$StreamingTV[data$StreamingTV == 'No internet service'] <- 'No'
data$StreamingMovies[data$StreamingMovies == 'No internet service'] <- 'No'

# 数据类型处理
data <- data %>%
  mutate(SeniorCitizen = as.character(SeniorCitizen)) %>%
  mutate_if(is.character,factor,ordered=F)

sapply(data,class)
```

探索性数据分析

```{r}
### 卡方检验 分类变量
# gender 不显著
pv1 <- chisq.test(table(data$gender,data$Churn))$p.value
p1 <- ggplot(data = data) +
  geom_bar(mapping = aes(x = gender, fill = Churn), position = "fill")

# SeniorCitizen
pv2 <- chisq.test(table(data$SeniorCitizen,data$Churn))$p.value
p2 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = SeniorCitizen, fill = Churn), position = "fill")

# Partner
pv3 <- chisq.test(table(data$Partner,data$Churn))$p.value
p3 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = Partner, fill = Churn), position = "fill")

# Dependents
pv4 <- chisq.test(table(data$Dependents,data$Churn))$p.value
p4 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = Dependents, fill = Churn), position = "fill")

# PhoneService 不显著
pv5<- chisq.test(table(data$PhoneService,data$Churn))$p.value
p5 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = PhoneService, fill = Churn), position = "fill")

# MultipleLines 无意义
pv6<- chisq.test(table(data$MultipleLines,data$Churn))$p.value
```
```

```
p6 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = MultipleLines, fill = Churn),
position = "fill")

# InternetService
pv7<-
chisq.test(table(data$InternetService,data$Churn))$p.value
p7 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = InternetService, fill = Churn),
position = "fill")

# OnlineSecurity
pv8<-
chisq.test(table(data$OnlineSecurity,data$Churn))$p.value
p8 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = OnlineSecurity, fill = Churn),
position = "fill")

# OnlineBackup
pv9<-
chisq.test(table(data$OnlineBackup,data$Churn))$p.value
p9 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = OnlineBackup, fill = Churn),
position = "fill")

# DeviceProtection
pv10<-
chisq.test(table(data$DeviceProtection,data$Churn))$p.valu
e
p10 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = DeviceProtection, fill =
Churn), position = "fill")

# TechSupport
pv11<-
chisq.test(table(data$TechSupport,data$Churn))$p.value
p11 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = TechSupport, fill = Churn),
position = "fill")

# StreamingTV
pv12<-
chisq.test(table(data$StreamingTV,data$Churn))$p.value
p12 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = StreamingTV, fill = Churn),
position = "fill")

# StreamingMovies
pv13<-
chisq.test(table(data$StreamingMovies,data$Churn))$p.valu
e
p13 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = StreamingMovies, fill =
Churn), position = "fill")

# Contract
pv14<-chisq.test(table(data$Contract,data$Churn))$p.value
p14 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = Contract, fill = Churn),
position = "fill")

# PaperlessBilling
pv15<-
chisq.test(table(data$PaperlessBilling,data$Churn))$p.value
p15 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = PaperlessBilling, fill =
Churn), position = "fill")

# PaymentMethod
pv16<-
chisq.test(table(data$PaymentMethod,data$Churn))$p.value
p16 <- ggplot(data=data) +
  geom_bar(mapping = aes(x = PaymentMethod, fill =
Churn), position = "fill")

p1+p2+p3+p4+p5+p6+p7+p8+p9+p10+p11+p12+p13+p14
+p15+p16

### 单因素方差分析
# tenure
tenure.aov <- aov(data$tenure~data$Churn)
pv17<-summary(tenure.aov)[[1]][["Pr(>F)"]][1]

# MonthlyCharges
MonthlyCharges.aov                                    <-
aov(data$MonthlyCharges~data$Churn)
pv18<-summary(MonthlyCharges.aov)[[1]][["Pr(>F)"]][1]

# TotalCharges
TotalCharges.aov <- aov(data$TotalCharges~data$Churn)
pv19<-summary(TotalCharges.aov)[[1]][["Pr(>F)"]][1]

p17 <- ggplot(data=data) +
  geom_boxplot(mapping = aes(x = Churn, y = tenure,
```

```
fill=Churn))
p18 <- ggplot(data=data) +
  geom_boxplot(mapping = aes(x = Churn, y = MonthlyCharges, fill=Churn))
p19 <- ggplot(data=data) +
  geom_boxplot(mapping = aes(x = Churn, y = TotalCharges, fill=Churn))

p17+p18+p19

# p-value plot
pv<-c(pv1,pv2,pv3,pv4,pv17,pv5,pv6,pv7,pv8,pv9,pv10,pv11,pv12,pv13,pv14,pv15,pv16,pv18,pv19)
p.frame <- data.frame(feature = names(data[-20]),P.value = pv)
p.frame <- p.frame[order(p.frame$P.value),]
p.frame$feature <- factor(p.frame$feature,levels = p.frame$feature)
ggplot(data = p.frame,aes(x = feature, y = P.value)) +
  geom_pointrange(aes(ymin=0,ymax=P.value),
                  size = 0.5,
                  color = ifelse(p.frame$P.value>0.001,'#F8766D','grey'))+
  coord_flip()
```

剔除不显著

```{r}
data <- data %>%
  dplyr::select(-gender,-PhoneService,-MultipleLines)
```

Logistic 模型

导入相关包

```{r}
library(tidyverse)
library(patchwork)
library(GGally)
library(pROC)
library(highcharter)
library(caret)
data.all <- data
```

数据分箱

一维高斯混合模型

```{r}
#高斯混合模型的一维情形，用来对连续变量分箱
#输入：数据向量，聚类簇 数
#输出：聚类结果和高斯分布的参数
#方法： EM 算法
GMMML <- function(x,k=4,eps=1e-8){
  x <- scale(x)%>%as.vector
  #初值
  mu <- quantile(x,seq(0,1,length=k+3))[2:(k+1)]
  pai <- rep(1/k,length=k)
  sigma <-rep(2,k)
  #计算 rik
  calculate_r <- function(x,mu,sigma,pai,k){
    #返回列表，个数为 k，每个长度为样本数
    a1 <- map(1:k,~exp(-(x-mu[.])^2/(2*sigma[.]))*pai[.]/sqrt(sigma[.]))
    p <- rep(0,length(a1[[1]]))
    for(m in 1:k){p <- p+a1[[m]]}
    for(m in 1:k){a1[[m]] <- a1[[m]]/p}
    return(a1)
  }
  #准备迭代
  flag <- 1
  while(flag>eps){
    #r 为列表，含有 k 个元素，为 rik
    r <- calculate_r(x,mu,sigma,pai,k)
    R <- r%>%map_dbl(sum)
    mu.new <-(1:k)%>%
      map_dbl(~sum(r[[.]]*x)/R[.])
    sigma.new <- (1:k)%>%
      map_dbl(~sum(r[[.]]*(x-mu.new[.])^2)/R[.])
    pai.new <- R/length(r[[1]])
    #退出循环条件退出循环条件
    flag <- (mu.new-mu)%>%'^'(2)%>%sum+
      (sigma.new-sigma)%>%'^'(2)%>%sum+
      (pai.new-pai)%>%'^'(2)%>%sum
    #更新参数更新参数
    mu <- mu.new
    sigma <- sigma.new
    pai <- pai.new
  }
  #确定聚类结果确定聚类结果
  cluster_result <- r%>%as.data.frame%>%
    apply(1,which.max)
  result <- list(cluster=cluster_result,
```

```
parm=list(pai=pai,mu=mu,sigma=sigma))
  return(result)
}
```

对连续变量分箱

```{r}
dataCON <- data.all %>%
  dplyr::select(where(is.numeric),Churn)%>%
  as.data.frame
class(dataCON)
# tenure
tenure.clust<-GMMML(dataCON[,1],k=4,eps=1e-6)[[1]]
table(tenure.clust)#查看频数
p20 <- ggplot(cbind(data.all, 类别=factor(tenure.clust)),aes(y=tenure,fill=类别 ))+
  geom_bar(position = 'fill')
data.all$tenure <- factor(tenure.clust)
table(data.all$tenure,data.all$Churn)%>%chisq.test()

# MonthlyCharges
MonthlyCharges.clust<-GMMML(dataCON[,2],k=4,eps=1e-6)[[1]]
table(MonthlyCharges.clust)#查看频数
p21 <- ggplot(cbind(data.all, 类别=factor(MonthlyCharges.clust)),aes(y=MonthlyCharges,fill=类别 ))+
  geom_bar(position = 'fill')
data.all$MonthlyCharges <- factor(MonthlyCharges.clust)
table(data.all$MonthlyCharges,data.all$Churn)%>%chisq.test()

#
TotalCharges.clust<-GMMML(dataCON[,3],k=4,eps=1e-6)[[1]]
table(TotalCharges.clust)#查看频数
p22 <- ggplot(cbind(data.all, 类别=factor(TotalCharges.clust)),aes(y=TotalCharges,fill= 类别 ))+
  geom_bar(position = 'fill')
data.all$TotalCharges <- factor(TotalCharges.clust)
table(data.all$TotalCharges,data.all$Churn)%>%chisq.test()

p20+p21
```

评分卡模型

WOE 编码

```{r}
#给数据框 WOE 编码
#输入：
#data：数据框
#y:因变量列名
#test：测试集数据框
#输出： WOE 编码后的数据框和 IV 值
#若有 test 则再返回 test 的编码结果
#注： test 和 data 的列名需要完全相同
WOEML <- function(data,y,test=NULL){
  flag <- 1
  if(is.null(test)){
    flag<- 0
    test <- data
  }
  result <- data%>%dplyr::select(eval(y))%>%unlist
  data <- data%>%dplyr::select(-eval(y))
  retest <- test%>%dplyr::select(eval(y))%>%unlist
  test <- test%>%dplyr::select(-eval(y))
  name <- names(data)
  IV <- rep(0,length(name))
  names(IV) <- name
  N0 <- table(result)["0"]
  N1 <- table(result)["1"]
  for(ii in 1:length(name)){
    M0 <- tapply(unlist(result),list(data[,ii],unlist(result)),length)%>%as.matrix
    for(jj in 1:nrow(M0)){
      WOE <- log((M0[jj,"1"]/M0[jj,"0"])/(N1/N0))
      IV[ii] <- IV[ii]+(M0[jj,"1"]/N1-M0[jj,"0"]/N0)*WOE

levels(data[,ii])[which(levels(data[,ii])==rownames(M0)[jj])] <- WOE

levels(test[,ii])[which(levels(test[,ii])==rownames(M0)[jj])] <- WOE
    }
    data[,ii] <- as.numeric(as.character(data[,ii]))
    test[,ii] <- as.numeric(as.character(test[,ii]))
  }
  data <- cbind(data,y=result)
  test <- cbind(test,y=retest)
  IV <- sort(IV)
  if(flag){
```

```
    return(list(data,IV,test))
  }else{
    return(list(data,IV))
  }
}
```

给所有变量编码
```{r}
data.all <- as.data.frame(data.all)
WOE <- WOEML(data.all,y="Churn")
data.WOE <- WOE[[1]]
# 相关性（all）
ggcorr(data.WOE[-17],label = T,digits = 2,hjust=0.8)
# 相关性（>0.5）
data.WOE %>%

dplyr::select(tenure,InternetService,Contract,MonthlyCharges,TotalCharges)%>%
  ggcorr(label = T,digits = 2,hjust=0.8)
```

建模
特征筛选
```{r}
data.WOE.pro <- data.WOE %>%
  select(-MonthlyCharges) # 高度相关 保留其一
fit.logit <- glm(y~.,data.WOE.pro,family = binomial(link = "logit"))
summary(fit.logit)
# 删除系数小于 0 的
data.WOE.pro <- data.WOE.pro %>%
  select(-Partner,-DeviceProtection)
fit.logit <- glm(y~.,data.WOE.pro,family = binomial(link = "logit"))
summary(fit.logit)
```

特征重要性
```{r}
data.all.select <- data.all[,-c(2,8,15)]
WOE <- WOEML(data.all.select,y="Churn")
WOE[[2]]
IV.frame <- data.frame(feature = names(WOE[[2]]),value = WOE[[2]])
IV.frame <- IV.frame[order(IV.frame$value),]
IV.frame$feature <- factor(IV.frame$feature,levels = IV.frame$feature)
ggplot(data = IV.frame,aes(x = value, y = feature)) +
  geom_bar(stat = 'identity',
           fill = ifelse(IV.frame$value>0.3,ifelse(IV.frame$value>=0.5,'#F8766D','#00BFC4'),'grey'),
           width = 0.9)

```

模型评估
```{r}
### 划分数据集
set.seed(2020111142)
ind <- sample(1:nrow(data),size = 0.7*nrow(data))
train <- data[ind,-c(2,8,15)]
test <- data[-ind,-c(2,8,15)]

### 训练集分箱
trainCON <- train %>%
  dplyr::select(where(is.numeric),Churn)%>%
  as.data.frame
avg <- map_dbl(trainCON[1:2],mean)
std <- map_dbl(trainCON[1:2],sd)
cl1<-GMMML(trainCON[,1],k=4,eps=1e-6)
train$tenure <- factor(cl1[[1]])
cl2<-GMMML(trainCON[,2],k=4,eps=1e-6)
train$TotalCharges <- factor(cl2[[1]])

### 测试集分箱
testCON <- test %>%
  dplyr::select(where(is.numeric),Churn)%>%
  as.data.frame
#分箱 1
p <- rep(0,nrow(testCON))
t <- testCON[,1]
cl1 <- cl1[[2]]
for(j in 1:length(t)){
  q <- rep(0,4)
  for(k in 1:4){
    q[k] <- cl1$pai[k]*dnorm((t[j]-avg[1])/std[1],cl1$mu[k],sqrt(cl1$sigma[k]))
  }
  p[j] <- which.max(q)
}
test$tenure <- factor(p)
#分箱 2
```

```
p <- rep(0,nrow(testCON))
t <- testCON[,2]
cl2 <- cl2[[2]]
for(j in 1:length(t)){
  q <- rep(0,4)
  for(k in 1:4){
    q[k] <- cl1$pai[k]*dnorm((t[j]-avg[2])/std[2],cl1$mu[k],sqrt(cl1$sigma[k]))
  }
  p[j] <- which.max(q)
}
test$TotalCharges<- factor(p)

### WOE 编码
train <- as.data.frame(train)
test <- as.data.frame(test)
WOE <- WOEML(train,y="Churn",test=test)
train.WOE <- WOE[[1]]
test.WOE <- WOE[[3]]

### 建模
train.WOE.fac <- train.WOE%>%
  mutate_if(is.numeric,factor,ordered=F)
test.WOE.fac <- test.WOE%>%
  mutate_if(is.numeric,factor,ordered=F)
fit.logit <- glm(y~.,train.WOE.fac,family = binomial(link = "logit"))
summary(fit.logit)

pred.logit.p <- predict(fit.logit,newdata = test.WOE.fac,type = 'response')

roc.logit <- roc(test.WOE.fac$y,pred.logit.p,quiet=T)
plot(roc.logit,print.auc=T,auc.polygon=T,grid=c(0.1,0.2),max.auc.polygon=T,auc.polygon.col='skyblue',print.thres=T)

pred.logit <- ifelse(pred.logit.p < 0.289,'0','1')
confusionMatrix(data = factor(pred.logit,levels = c('0','1')),reference = test.WOE.fac$y)

```

Adaboost
导入相关包
```{r}
library(tidyverse)
```

决策树
```{r}
#决策树实现
#输入：
#x：自变量数据框
#y：因变量向量
#i：当前深度，调用取 0 即可
#subset：该树可以试用的变量列
#weight：样本权重向量
#输出：列表嵌套结 构的决策树
#方法：递归
DTML <- function(x,y,i,subset,weight=rep(1,length(y))){
  w <- weight
  i <- i+1
  #停止准则
  if( (max(y)==min(y)) || i>max_iter || (length(subset)==0) )
{
    return(round(sum(y*w)/sum(w)))}
  #调用子函数找分裂点
  tvar <- FindPoint(x,y,subset,w)
  #判断信息增益率
  if(tvar[[2]]< 1e-6){
    return(round(sum(y*w)/sum(w)))}
  #分裂分裂
  data <- cbind(w,y,x)
  index <- which(eval(parse(text=paste0("data$",tvar[[1]]))))
  x1 <- data[index,]
  x2 <- data[-index,]
  y1 <- x1[,2]
  y2 <- x2[,2]
  w1 <- x1[,1]
  w2 <- x2[,1]
  x1 <- x1[,-c(1,2)]
  x2 <- x2[,-c(1,2)]
  #递归拿树递归拿树
  left.tree <- DTML(x1,y1,i,subset,w1)
  right.tree <- DTML(x2,y2,i,subset,w2)
  return(list(condition=tvar[[1]],
              Left =left.tree,
              Right=right.tree))
}
#查找最佳分裂点的子函数查找最佳分裂点的子函数
FindPoint <- function(x,y,subset,weight=8){
  w <- weight
  #计算交叉熵计算交叉熵 H(t)
  n <- length(y)
```

```
  H <- cross.entropy(y,weight=w)
  #寻找分裂结点寻找分裂结点
  x <- x%>%select(subset)
  x1 <- x%>%select(where(is.numeric))#连连续型续型
  x2 <- x%>%select(where(is.factor))#因子因子型型
  #对变量循环对变量循环
  tvar <- list(con="",fai=0,varname="")
  p <- ncol(x1)
  if(p>0){
    for(ii in 1:p){
      a <- sort(unique(x1[,ii]))
      a <- (a[1:(length(a)-1)]+a[-1])/2
      a <- a[sample(length(a),min(30,length(a)))]
      #对取值循环对取值循环
      for(jj in a){
        p0 <- y[x1[,ii]<=jj]#左节左节点点
        Htl <- cross.entropy(p0,weight=w[x1[,ii]<=jj])
        p1 <- y[x1[,ii]>jj]
        Htr<-  cross.entropy(p1,weight=w[x1[,ii]>jj])# 右节右节点点
        #计算信息增益计算信息增益
        fai <- H-
          sum(w[x1[,ii]<=jj])/sum(w)*Htl-
          sum(w[x1[,ii]>jj])/sum(w)*Htr
        if(fai>tvar[[2]]){
          tvar[[2]] <- fai
          tvar[[3]] <- names(x1)[ii]
          tvar[[1]] <- paste0(names(x1)[ii],"<=",jj)
        }
      }
    }
  }
  p <- ncol(x2)
  if(p>0){
    for(ii in 1:p){
      a <- which(table(x2[,ii])>0)%>%names
      xf <- as.character(x2[,ii])
      #对取值循环对取值循环
      for(jj in 1:20){
        b <- 0
        while(sum(b)==0){b <- sample(2,length(a),T)-1}
        p0 <- y[xf %in% a[b]]#左左节点节点
        Htl <- cross.entropy(p0,weight=w[xf %in% a[b]])
        p1 <- y[!(xf %in% a[b])]
        Htr<-  cross.entropy(p1,weight=w[!(xf %in% a[b])])#右节点右节点
        #计算信息增益计算信息增益
        fai <- H-
          sum(w[xf %in% a[b]])/sum(w)*Htl-
          sum(w[!(xf %in% a[b])])/sum(w)*Htr
        if(fai>tvar[[2]]){
          tvar[[2]] <- fai
          tvar[[3]] <- names(x2)[ii]
          d <- "c("
          for(m in unique(a[b])){d <- paste0(d,"\"",m,"\"",",")}
          d <- substr(d,1,nchar(d)-1)
          d <- paste0(d,")")
          tvar[[1]] <- paste0(names(x2)[ii]," %in% ",d)
        }
      }
    }
  }
  return(tvar)
}
#决策树的子函数：计算交叉熵决策树的子函数：计算交叉熵
cross.entropy <- function(y,weight=1){
  if(length(unique(y))<=1){return(0)}
  P1 <- sum(y*weight)/sum(weight)
  P0 <- 1-P1
  H <- -P0*log(P0)-P1*log(P1)
  return(H)
}
#决策树预测函数决策树预测函数
#输入：输入：
#modeltree：模型；：模型；x：自变量数据框：自变量数据框
#输出：分类结果向量输出：分类结果向量
#方法：递归方法：递归
pred.tree <- function(modeltree,x){
  if(nrow(x)<1){return(NULL)}
  re <- rep(-1,nrow(x))
  a <- modeltree
  #停止准则停止准则
  if(!is.list(a)){return(rep(a,nrow(x)))}
  #分裂分裂
  p <- which(eval(parse(text=paste0("x$",a[[1]]))))#左孩子行号左孩子行号
  if(length(p)<1){#全进右边全进右边
    x2 <- x
    b<- pred.tree(modeltree = a[[3]],x=x2)
    if(!is.null(b)){re <- b}
  }else{
```

```
    x1 <- x[p,] #进左进左
    x2 <- x[-p,]
    b <- pred.tree(modeltree = a[[2]],x=x1)
    if(!is.null(b)){re[p] <- b}
    b<- pred.tree(modeltree = a[[3]],x=x2)
    if(!is.null(b)){re[-p] <- b}
  }
  return(re)
}
```

Adaboost

```{r}
#adaboost 实现代码
#输入：
#x:自变量数据框
#y:因变量向量 {0,1}
#M：弱学习器个数
#输出：每个弱学习器和各自的权重
AdaboostML <- function(x,y,M=5,subset=names(x),max_iter=5){
  n <- nrow(x)
  w <- rep(1/n,n)
  L <- list()
  alpha <- c()
  for(iii in 1:M){
    #拟合弱分类器
    L1 <- DTML(x=x,y=y,i=0,subset=subset,weight = w)
    pred <- pred.tree(L1,x=x) *2-1
    err <- sum(w*(pred!=(y*2-1)))
    if(err==0 || err>=0.5 ){break}
    alpha1 <- 0.5*log((1-err)/err)
    if(is.na(alpha1) || alpha1<0){next}
    w <- w*exp(-(y*2-1)*alpha1*pred)
    w <- w/sum(w)
    L[[iii]] <- L1
    alpha[iii] <- alpha1
  }
  return(list(L,alpha))
}
#adaboost 模型预测代码
#输入：
#L：模型 test：测试自变量
#输出：分类结果
pred.adaboost <- function(L,test){
  M <- length(L[[1]])
  re <- map(1:M,~( pred.tree( L[[1]][[.]], test )*2-1 )*L[[2]][[.]])%>%
    as.data.frame%>%apply(1,sum)
  return(re)
}
```

模型建立

```{r}
### 划分数据集
set.seed(2020111142)
ind <- sample(1:nrow(data),size = 0.7*nrow(data))
train <- data[ind,]
test <- data[-ind,]

#训练预备
xt <- test%>%select(-Churn)%>%as.data.frame
yt <- test%>%select(Churn)%>%unlist%>%as.character%>%as.numeric
x <- train%>%select(-Churn)%>%as.data.frame
y <- train%>%select(Churn)%>%unlist%>%as.character%>%as.numeric
#训练模型
max_iter=2
L <- AdaboostML(x,y,M=50,subset=names(x),max_iter = 2)
#预测效果
pred.ada.fx <- pred.adaboost(L,xt)

roc.ada <- roc(as.factor(yt*2-1),pred.ada.fx,quiet=T)
plot(roc.ada,print.auc=T,auc.polygon=T,grid=c(0.1,0.2),max.auc.polygon=T,auc.polygon.col='skyblue',print.thres=T)

# 按最优阈值分类
pred.ada <- ifelse(pred.ada.fx<(-0.705),'0','1')
confusionMatrix(data = factor(pred.ada,levels = c('0','1')),reference = as.factor(yt))
```

随机森林

导入相关包

```{r}
library(randomForest)
```

建模

```{r}
```

```
# 划分数据集
set.seed(2020111142)
ind <- sample(1:nrow(data),size = 0.7*nrow(data))
train <- data[ind,]
test <- data[-ind,]

train <- as.data.frame(train)

# 验证集
ind.val <- sample(1:nrow(train),size = 0.7*nrow(train))
val.train <- train[ind.val,]
val <- train[-ind.val,]

# 调参
acc.test <- numeric()
accuracy1 <- NULL
accuracy2 <- NULL
for(i in 1:16){
  set.seed(2020111142)
  rf.train<-
randomForest(Churn~.,data=val.train,mtry=i,ntree=1000)
  rf.pred <- predict(rf.train, val[,-17])
  accuracy1 <- confusionMatrix(rf.pred,val$Churn)
  accuracy2[i] <- accuracy1$overall[1]
}
acc.test <- data.frame(p=1:16,cnt=accuracy2)
opt.p <- subset(acc.test,cnt==max(cnt))[1,]
sub <- paste("num of parameter:",opt.p$p,"  accuracy:",
opt.p$cnt)
sub

hchart(acc.test,'line',hcaes(p,cnt))%>%
  hc_title(text='Accuracy With mtry(randomforest)')%>%
  hc_subtitle(text=sub)%>%
  hc_add_theme(hc_theme_google())%>%
  hc_xAxis(title=list(text = 'Number of Parameters'))%>%
  hc_yAxis(title=list(text = 'Accuracy'))

fit.rf <- randomForest(Churn~.,data =
train,mtry=2,ntree=1000,proximity=T,importance=T)
plot(fit.rf,lwd=3,main = "Random Forest (Error Rate vs.
Number of Trees)")
varImpPlot(fit.rf)

pre.rf.p <- predict(fit.rf,test[,-17],type="prob")[,1]
roc.rf <- roc(test$Churn,pre.rf.p,quiet=T)
plot(roc.rf,print.auc=T,auc.polygon=T,grid=c(0.1,0.2),max.a
uc.polygon=T,auc.polygon.col='skyblue',print.thres=T)

# 按最优阈值分类
pred.rf <- ifelse(pre.rf.p<0.704,'1','0')
confusionMatrix(data = factor(pred.rf,levels =
c('0','1')),reference = test$Churn)

```

朴素贝叶斯
导入相关包
```{r}
library(e1071)
```

建模
```{r}
### 划分数据集
set.seed(2020111142)
ind <- sample(1:nrow(data),size = 0.7*nrow(data))
train <- data[ind,]
test <- data[-ind,]

fit.nb <- naiveBayes(train[,-17],train$Churn)
pred.nb.p <- predict(fit.nb,test[,-17],type='raw')[,1]
roc.nb <- roc(test$Churn,pred.nb.p,quiet=T)
plot(roc.nb,print.auc=T,auc.polygon=T,grid=c(0.1,0.2),max.
auc.polygon=T,auc.polygon.col='skyblue',print.thres=T)

# 按最优阈值分类
pred.nb <- ifelse(pred.nb.p<0.511,'1','0')
confusionMatrix(data = factor(pred.nb,levels =
c('0','1')),reference = test$Churn)
```