目录

[机器学习课程设计报告 0](#_Toc121357348)

[实验一 运营商流失客户分析 2](#_Toc121357349)

[一、实验目的 2](#_Toc121357350)

[二、实验原理及说明 2](#_Toc121357351)

[三、实验内容 3](#_Toc121357352)

[1. 数据预处理 5](#_Toc121357353)

[2. 特征分析 8](#_Toc121357354)

[3. 模型建立与求解 10](#_Toc121357355)

[4. 最优模型选择和预测 11](#_Toc121357356)

[四、实验安全事项 11](#_Toc121357357)

[五、实验提交方式 11](#_Toc121357358)

# 实验一 运营商流失客户分析

## 一、实验目的

1. 了解数据的预处理。
2. 掌握机器学习算法在该案例中的应用（重点）。
3. 了解最优模型的选择（难点）。

## 二、实验原理及说明

随着业务的快速发展、移动业务市场的竞争愈演愈烈。如何最大程度地挽留在网用户、吸取新客户，是电信企业最关注的问题之一。竞争对手的促销、公司资费软着陆措施的出台和政策法规的不断变化，影响了客户消费心理和消费行为，导致客户的流失特征不断变化。对于电信运营商而言，流失会给电信企业带来市场占有率下降、营销成本增加、利润下降等一系列问题。在发展用户每月增加的同时，如何挽留和争取更多的用户，是一项非常重要的工作。

随着大数据挖掘技术的不断发展和应用，移动运营商希望能借助数据挖掘技术识别哪些用户可能流失，什么时候会发生流失。而通过建立流失预测模型，分析用户的历史数据和当前数据，提取辅助决策的关键性数据，并从中发现隐藏关系和模式，进而预测未来可能发生的行为，就可以帮助移动运营商实现这些要求。

利用用户的短信、流量、通话、消费的使用情况以及客户基本信息的数据，采用数据挖掘技术对用户进行分群。

分析不同群体用户的使用规律，识别各群体客户流失的重要特征。

建立不同群体用户流失模型，结合结果为运营商提供差异化的意见和建议。

## 三、实验内容

1. 数据预处理
2. 重复数据处理

由于使用重复数据会对下面的分析和建模产生不利影响，因此，将所有重复数据视

为异常值剔除。

1. 降维

降维处理原始数据的变量。

1. 数据的提取和整合

 在原始的建模数据中，每个用户有三行的数据，对应着同一个 ID 三个月的信息。这种数据格式不便于进行建模和分析。因此，本文对数据进行提取和整合，将每个用户信息处理为一行数据。具体的处理方式如下:

a 移动客户基本信息：由于同一个用户的基本信息在三个月内是相同的。因此，用户 ID、性别、星座、年龄以及终端硬件类型只取 1 月的数据。

b 在网时长：由于每个月的在网时长等于上个月的在网时长加一（为零的除外），故第三个月的在网时长的数据就包含了这三个月在网时长的全部信息。因此，每个 ID 只提取第三个月的数据。

c 是否合约有效：将这三个月的合约有效的情况规整为一个数据，处理如下：当三个月不全为 1 时，用第三个月的值减去前两个的均值；当三个月的值都是为 1 时，取值为 1.5。所有取值情况为-1、-0.5、0、0.5、1、1.5。

d 合约计划到期时间：取第三个月的数据作为合约计划到期时长，将空值赋值为-1， 不是空值的到期时间以 201603 为基准 0，每增加一个月，数据为 1，例如日期为 201607，那么该月份的到期时间为 4，以此类推。

e 信用等级：取 3 个月的平均值。

f VIP 等级：将空值赋值为 0 ，同一个 ID 值，三个月的数值相等，则取第三个月的数值；三个月的数值都不相等，则第三个月的数据减去前两个月的均值。这样的处理可以突出用户在第三个月的变化情况。

g 本月费用：取三个月的平均值。

h 平均每次通话时长：将各种通话时长除以通话次数，得到各类的平均通话时长。

i 其余的变量处理方式相同：同一个 ID 下，取三个月的平均值。经上述规则提取和整合后，得到的新数据集。

1. 缺失值的处理

用 python 对新数据集进行缺失值查找，缺失值集中在用户的基本信息——星座、年龄和性别上。对缺失的数据进行赋值，性别缺失的赋值为 3，年龄缺失的赋值为 0，星座缺失的赋值为 0。

1. 异常值的处理

对新数据集进行异常值的查找，查找出在网时长小于 0 的数据，费用大于 4 万元的数据。这些异常数据有很大的可能是因为人为录入时出现失误，直接利用异常数据建模会导致模型结果与实际情况出现严重的偏差。

1. 流失用户的特征分析

对性别、年龄、星座、在网时长、合约是否有效、合约到期时间、信用等级、VIP 等级等进行分析吗，进一步对数据进行降维。接着，使用 K-Means 进行聚类分析，对人群进行分类。

1. 特征值的提取

特征抽取是数据挖掘任务最为重要的一个环节，一般而言，它对最终结果的影响要高过数据挖掘算法本身。只有先把现实用特征表示出来，才能借助数据挖掘的力量找到问题的答案。特征选择的另一个优点在于：降低真实世界的复杂度，模型比现实更容易操纵。

特征选择的原因有：(1)降低复杂度；(2)降低噪音；(3)增加模型可读性。

单个特征和某一类别之间相关性的计算方法有很多，比较有效的有卡方检验（chi2） 以及互信息和信息熵，本文我们选择基于信息熵的方法来选取特征变量。

信息熵是在决策树中广泛使用的一个变量，用以获取最优划分的节点。基于树的预测模型能够用来计算特征的重要程度，因此能用来去除不相关的特征。

因此选择基于树的特征选择(Tree-based feature selection)来获取特征变量。

1. 模型的建立与求解

分别对分类后的每个数据集使用神经网络模型、决策树模型、朴素贝叶斯模型和支持向量机模型进行建模，使用测试数据集计算出模型的错误率，选择错误率最低的模型作为最优模型，并使用最优的模型对该类别的预测数据集进行预测。

分类后的五个数据集作为建模数据集，将各个建模数据集按照 8:2 的比例随机抽取80%的数据作为待处理训练数据集，20%的建模数据作为测试数据集，待处理训练测试集和测试数据集中均包含流失和非流失用户的信息。

1. 最优模型的选择和预测

对整合数据的本月消费进行 K-Means 聚类，将数据分为 5 类，分别为高费用、中高费用、一般费用、中低费用、低费用，结合选出的关于每类的最优模型，对预测数据的每一类进行流失预测。

### 数据预处理

1. 删除重复数据

由于使用重复数据会对下面的分析和建模产生不利影响，因此，将所有重复数据视为异常值剔除。找到后直接删除即可

|  |
| --- |
| print("2 duplicating data:", data.duplicated().sum()) # 查看重复数据  data.drop\_duplicates(inplace=True) # 数据去重  print("3 result", data.duplicated().sum()) # 去重后的数据 |

1. 降维

删除一些无关数据，咱们相信科学，肯定不会认为客户离开和星座有关，星座只是ta想离开你的借口罢了。此外，手机品牌、手机型号、操作系统 同样无关，一并删除即可，如此一来就减少了变量的数量，从而达到降维的效果。

|  |
| --- |
| data.drop(['MANU\_NAME', 'MODEL\_NAME', 'OS\_DESC', 'CONSTELLATION\_DESC'], axis=1, inplace=True)  cleardata = data;  print(cleardata) # 降维后的数据 |

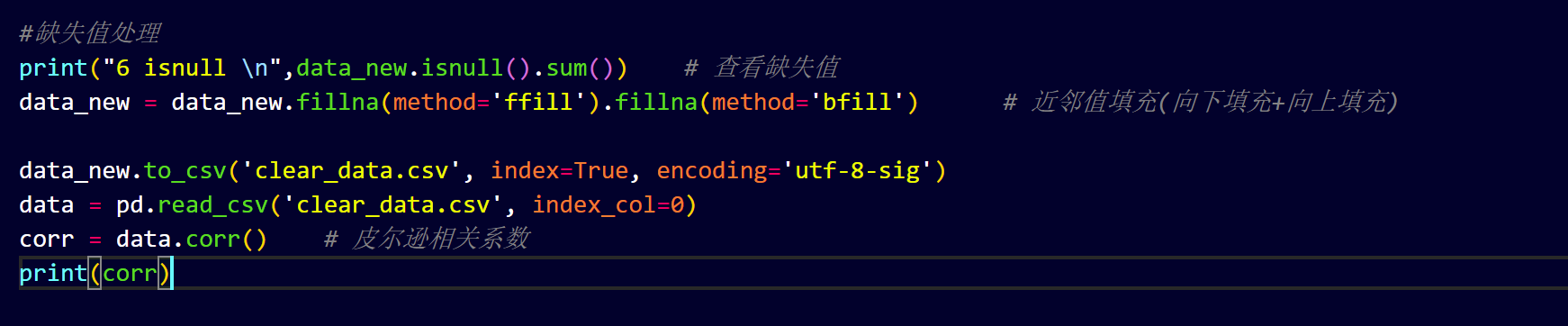
1. 数据提取与整合

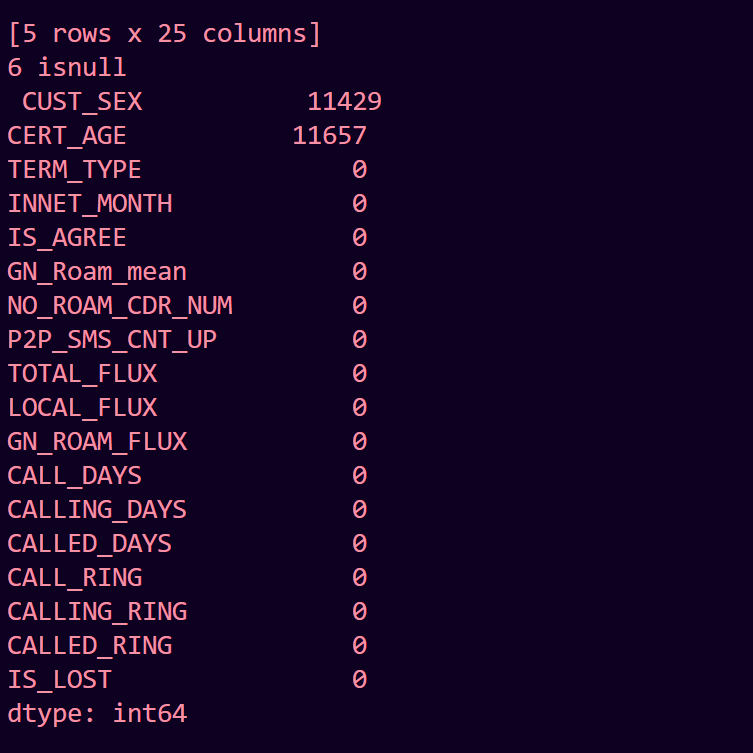
在原始的建模数据中，每个用户有三行的数据，对应着同一个ID三个月的信息。这种数据格式不便于进行 建模和分析。因此，本文对数据进行提取和整合，将每个用户信息处理为一行数据。

|  |
| --- |
| data\_1 = data\_group[['CUST\_SEX', 'CERT\_AGE', 'TERM\_TYPE']].first()  data\_2 = data\_group['INNET\_MONTH'].last()  data\_3 = pd.DataFrame(data\_group['IS\_AGREE'].agg(cal\_is\_agree))#agg是一个聚合函数，聚合函数操作始终是在轴（默认是列轴，也可设置行轴）上执行，  date = data\_group['AGREE\_EXP\_DATE'].last() # 取第3个月的"合约计划到期时长"  num\_mon = (pd.to\_datetime(date, format='%Y%m') - pd.to\_datetime('2016-03')).dt.days/30 # 时长以“月”为单位  data\_4 = pd.DataFrame(num\_mon).fillna(-1) #用-1填充缺失值  data\_5 = pd.DataFrame(data\_group['CREDIT\_LEVEL'].agg('mean')) # 信用等级  # 3.7 VIP等级  data\_6 = data\_group['VIP\_LVL'].last().fillna(0) # 取最后一个值  # 3.8 本月费用(取三个月的平均值)特征构建  data\_7 = pd.DataFrame(data\_group['ACCT\_FEE'].mean())  # 3.9 平均每次通话时长  # 总通话  data\_8\_1 = pd.DataFrame(data\_group['CALL\_DURA'].sum()/data\_group['CDR\_NUM'].sum(),  columns=['Total\_mean'])  # 本地通话  data\_8\_2 = pd.DataFrame(data\_group['NO\_ROAM\_LOCAL\_CALL\_DURA'].sum()/data\_group['NO\_ROAM\_LOCAL\_CDR\_NUM'].sum(),  columns=['Local\_mean'])  # 国内长途通话  data\_8\_3 = pd.DataFrame(data\_group['NO\_ROAM\_GN\_LONG\_CALL\_DURA'].sum() / data\_group['NO\_ROAM\_GN\_LONG\_CDR\_NUM'].sum(),  columns=['GN\_Long\_mean'])  # 国内漫游通话  data\_8\_4 = pd.DataFrame(data\_group['GN\_ROAM\_CALL\_DURA'].sum() / data\_group['GN\_ROAM\_CDR\_NUM'].sum(),  columns=['GN\_Roam\_mean'])  # 数据拼接  data\_8 = pd.concat([data\_8\_1, data\_8\_2, data\_8\_3, data\_8\_4], axis=1).fillna(0)  # 3.10 其他变量  # 非漫游通话次数（次）、短信发送数（条）、上网流量(MB)、本地非漫游上网流量(MB)、国内漫游上网流量(MB)、  # 有通话天数、有主叫天数、有被叫天数 （主叫 + 被叫 ≠ 总通话）  # 语音呼叫圈、主叫呼叫圈、被叫呼叫圈  data\_9 = data\_group[['NO\_ROAM\_CDR\_NUM', 'P2P\_SMS\_CNT\_UP', 'TOTAL\_FLUX', 'LOCAL\_FLUX','GN\_ROAM\_FLUX',  'CALL\_DAYS', 'CALLING\_DAYS', 'CALLED\_DAYS',  'CALL\_RING','CALLING\_RING', 'CALLED\_RING']].agg('mean') |

1. 缺失值与异常值

用python对新数据集进行缺失值查找，共发现存在11429条性别缺失值，还有11657条年龄缺失值。对缺失的数据进行赋值，这里就不按照指导书上来写了，指导书上写的很模糊，这里将缺失值用近邻值来填充。最后就把所有处理好的数据保存在新的文件中。

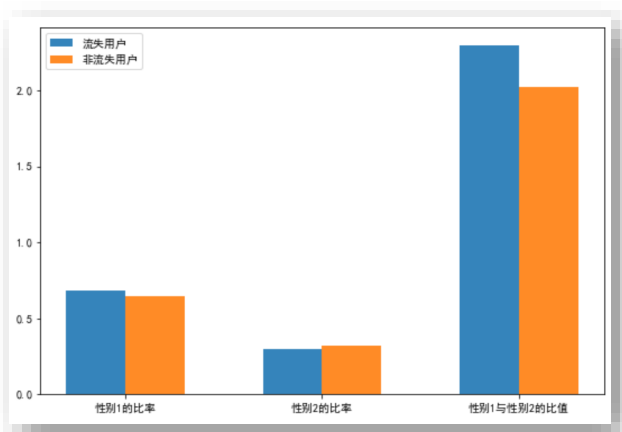




### 特征分析

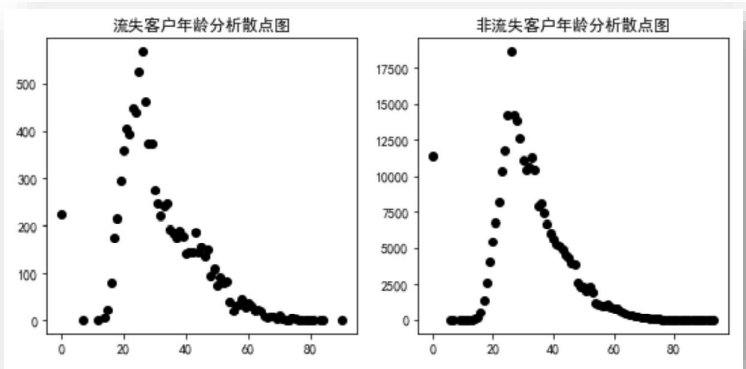
性格分析：

性别为1和性别为2占所在群体中比例的差别不大， 但从性别1与性别的比例中可以看到，流失用户群 体中性别的比值要比非流失用户群体中性别的比值 相对更大，即虽然流失用户与非流失用户中的性别 比例相当，但是在流失用户中，性别1比性别2的 用户更易流失。



年龄分析：

流失用户群体与非流失用户群体各个年龄使用的人数呈偏态分布，其中年龄为26岁的用户是最多的。流失 用户群体与非流失用户群体在各个年龄上均有用户流失，由于两个群体关于年龄的特征相似，因此，认为 年龄与用户是否在3月流失无重要关系。

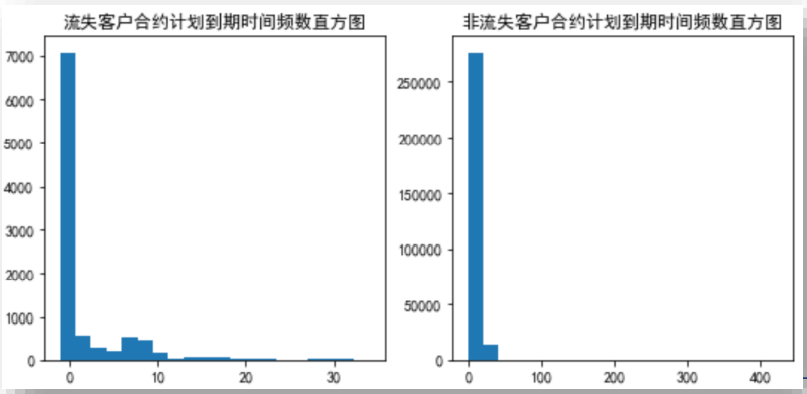


星座分析：

星座不用分析，星座不可信。

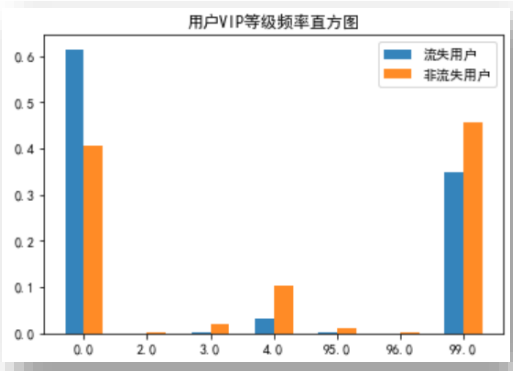
在网时长分析：

在流失客户中，在网时长高于20个月的用户数都不超过100，而大多数流失用户的在网时长都低于20个月， 数量占了流失客户的85.19%；在非流失客户中，有44.95%的用户在网时长低于20个月；流失用户在网时 长低于20个月的比例约为非流失用户的2倍，因此可以认为在网时长对用户是否在3月流失有重要的影响， 同时，在网时长低于20个月的用户易于流失。

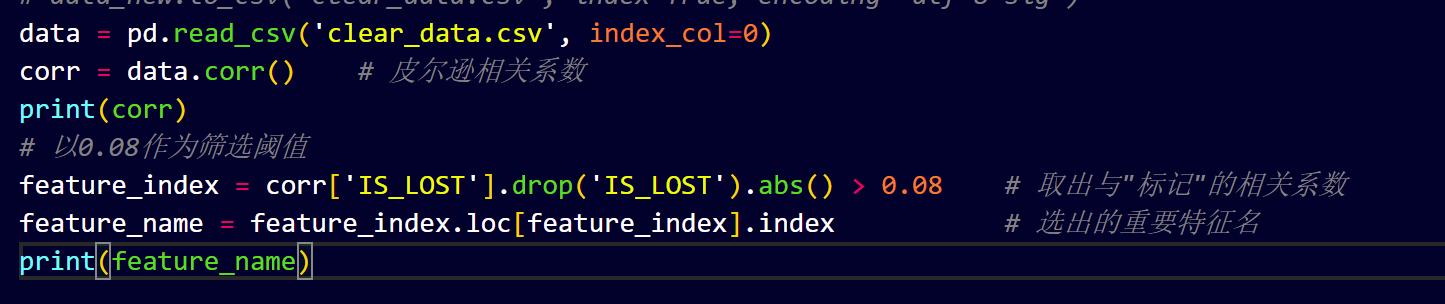


Vip等级分析：

流失用户中VIP等级缺失的情况比非流失的更多，且VIP等级为4的用户更不容易流失。因此，可认为 VIP等级是用户流失的一个特征变量。



使用皮尔逊相关系数取出相关系数高的特征：



输出：

|  |
| --- |
| Index(['INNET\_MONTH', 'CREDIT\_LEVEL', 'NO\_ROAM\_CDR\_NUM', 'CALL\_DAYS',  'CALLING\_DAYS', 'CALLED\_DAYS', 'CALL\_RING', 'CALLED\_RING'], dtype='object') |

### 模型建立与求解

1. **随机森林**

|  |
| --- |
| rfc = RandomForestClassifier() # 初始化随机森林模型  rfc.fit(X\_train, y\_train) # 模型训练  y\_pre = rfc.predict(X\_test) # 调用模型对测试样本进行预测  print(classification\_report(y\_test, y\_pre)) # 打印分类报告（包含了各模型性能评价指标）  lr\_acc = round(accuracy\_score(y\_pre,y\_test)\*100,2)  print(f"logistic accuracy is: {lr\_acc}%") |

准确度为：78.85%

1. **决策树**

|  |
| --- |
| # 创建决策树模型  dtc = DecisionTreeClassifier()  # 训练模型  dtc.fit(X\_train,y\_train)  # 预测训练集和测试集结果  dtc\_pred = dtc.predict(X\_test)  # 计算精确度  dtc\_acc = round(accuracy\_score(dtc\_pred,y\_test)\*100,2)  print(f"decision tree accuracy is: {dtc\_acc}%") |

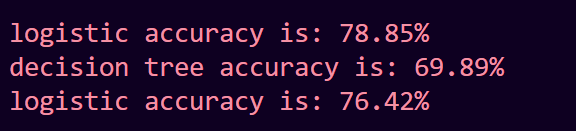
准确度为：69.89%

1. **逻辑回归**

|  |
| --- |
| # 创建逻辑回归模型  lr = LogisticRegression()  # 训练模型  lr.fit(X\_train,y\_train)  # 预测训练集和测试集结果  lr\_pred = lr.predict(X\_test)  # 计算精确度  lr\_acc = round(accuracy\_score(lr\_pred,y\_test)\*100,2)  print(f"logistic accuracy is: {lr\_acc}%") |

准确度为：76.42%

### 最优模型选择和预测



从上可见，准确度最好的是决策树模型。

## 四、实验安全事项

实验过程中注意用电安全。

## 五、实验提交方式

√ 实验报告 □ 现场打分 □ 线上平台提交