****



**基于Bank Marketing数据集的分类算法应用**

作者姓名 张开顺

作者学号 21851187

指导教师 贝毅君

学科专业 软件工程

所在学院 软件学院

提交日期 二〇一九年〇一月

目录

[1 实验目的 3](#_Toc535487745)

[2 数据描述 3](#_Toc535487746)

[3 数据预览 4](#_Toc535487747)

[4 预处理方法 6](#_Toc535487748)

[5 模型建立 7](#_Toc535487749)

[5.1 对数几率回归模型 7](#_Toc535487750)

[5.2 决策树模型 7](#_Toc535487751)

[5.3 AdaBoost算法 9](#_Toc535487752)

[5.4 其他算法 10](#_Toc535487753)

[6 模型评估 10](#_Toc535487754)

[7 结果可视化 11](#_Toc535487755)

## 1 实验目的

对数据进行分类是机器学习中常见的任务，分类的方式有有监督和无监督的。本实验将使用机器学习中常用的有监督的分类方法对Bank Marketing数据集进行分类。学习分类算法的基本原理，使用Python编写算法实现，掌握sklearn等库的使用方法。

## 2 数据描述

该数据集一共包含4个文件，如表1-1。

**表1-1**

|  |  |
| --- | --- |
| **文件名** | **文件说明** |
| bank-additional-full.csv | 包含41188个样例，每个样例20个特征 |
| bank-additional.csv | 10% 的 bank-additional-full.csv |
| bank-full.csv | bank-addition-full.csv的旧版本 |
| bank.csv | bank-additional.csv的旧版本 |

本实验主要使用bank-additional-full.csv文件来作为数据集。该文件中，每一个样例都有20个特征，特征列表如表1-2。

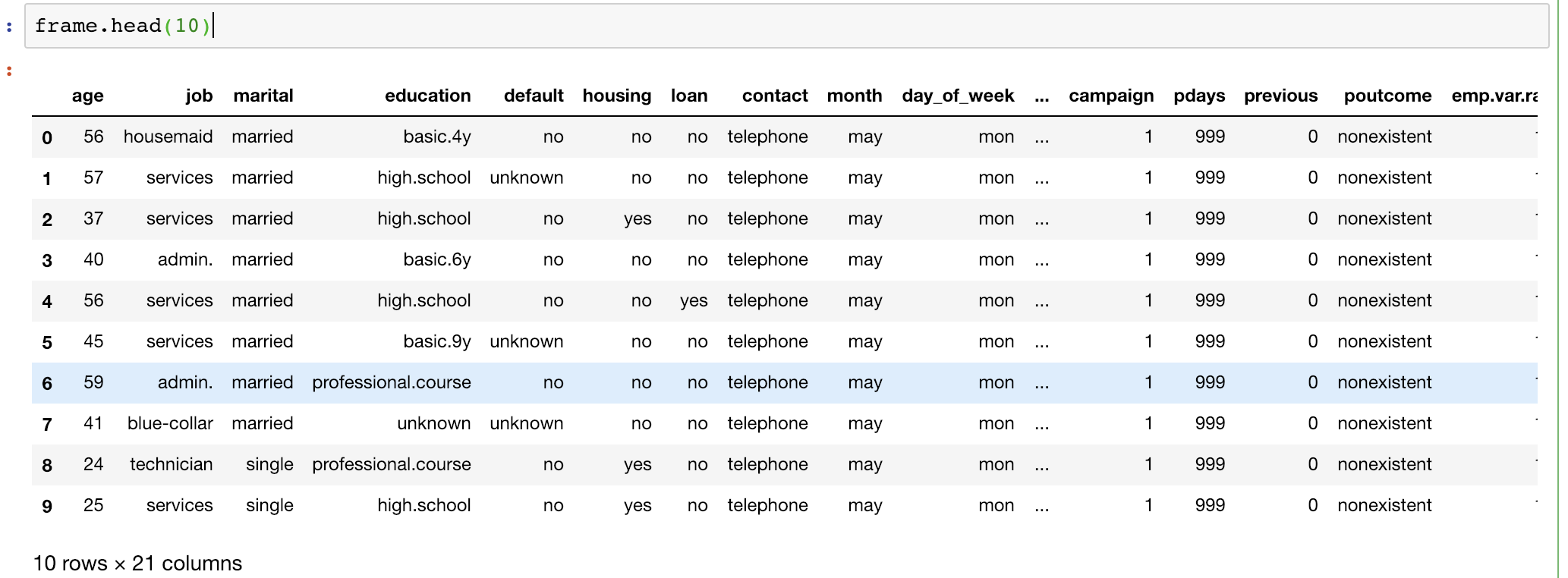
**表1-2**

|  |  |
| --- | --- |
| **特征** | **特征描述** |
| age | 年龄，数值型 |
| job | 工作，类别型，取值有：  'admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown' |
| marital | 婚姻状况，类别型，取值有:  'divorced', 'married', 'single', 'unknown' |
| education | 教育程度，类别型，取值有：  'basic.4y','basic.6y','basic.9y','high.school','illiterate'，  'professional.course', 'university.degree', 'unknown' |
| default | 信用状况，类别型，取值有：'no', 'yes', 'unknown' |
| housing | 是否有房贷，类别型，取值有：'no', 'yes', 'unknown' |
| loan | 是否有个人贷款，类别型，取值有：'no', 'yes', 'unknown' |
| contact | 交流类型，类别型，取值有：'cellular', 'telephone' |
| month | 月份，类别型，取值有：'jan', 'feb', 'mar', ..., 'nov', 'dec' |
| day\_of\_week | 星期中的哪一天，类别型，取值有：'mon', 'tue', 'wed', 'thu', 'fri' |
| duration | 时间间隔(s)，数值型 |
| campaign | 这次活动联系的次数，数值型 |
| pdays | 距上次活动联系后持续的天数，数值型 |
| previous | 活动之前联系的次数，数值型 |
| poutcome | 上次活动的结果，类别型，取值有：'failure', 'nonexistent', 'success' |
| emp.var.rate | 就业变动率，数值型 |
| cons.price.idx | 消费者物价指数，数值型 |
| cons.conf.idx | 消费者自信指数，数值型 |
| euribor3m | euribor 3个月的费率，数值型 |
| nr.employed | 雇员的数量，数值型 |
| y | 客户是否订购，类别型，取值有1，0 |

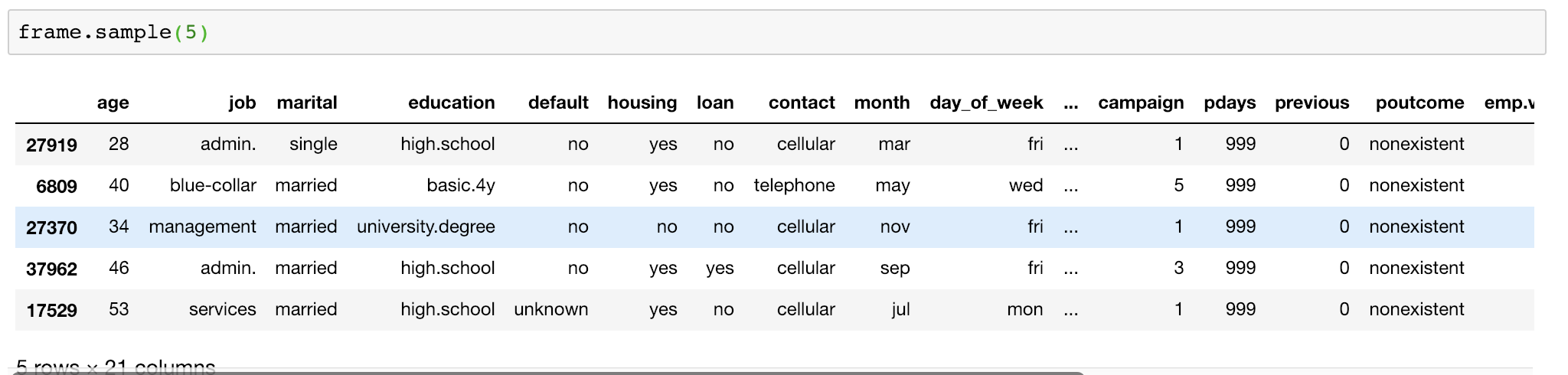
## 3 数据预览

当拿到一份数据的时候，我们可以通过一些方式简单地看一下数据，比如：

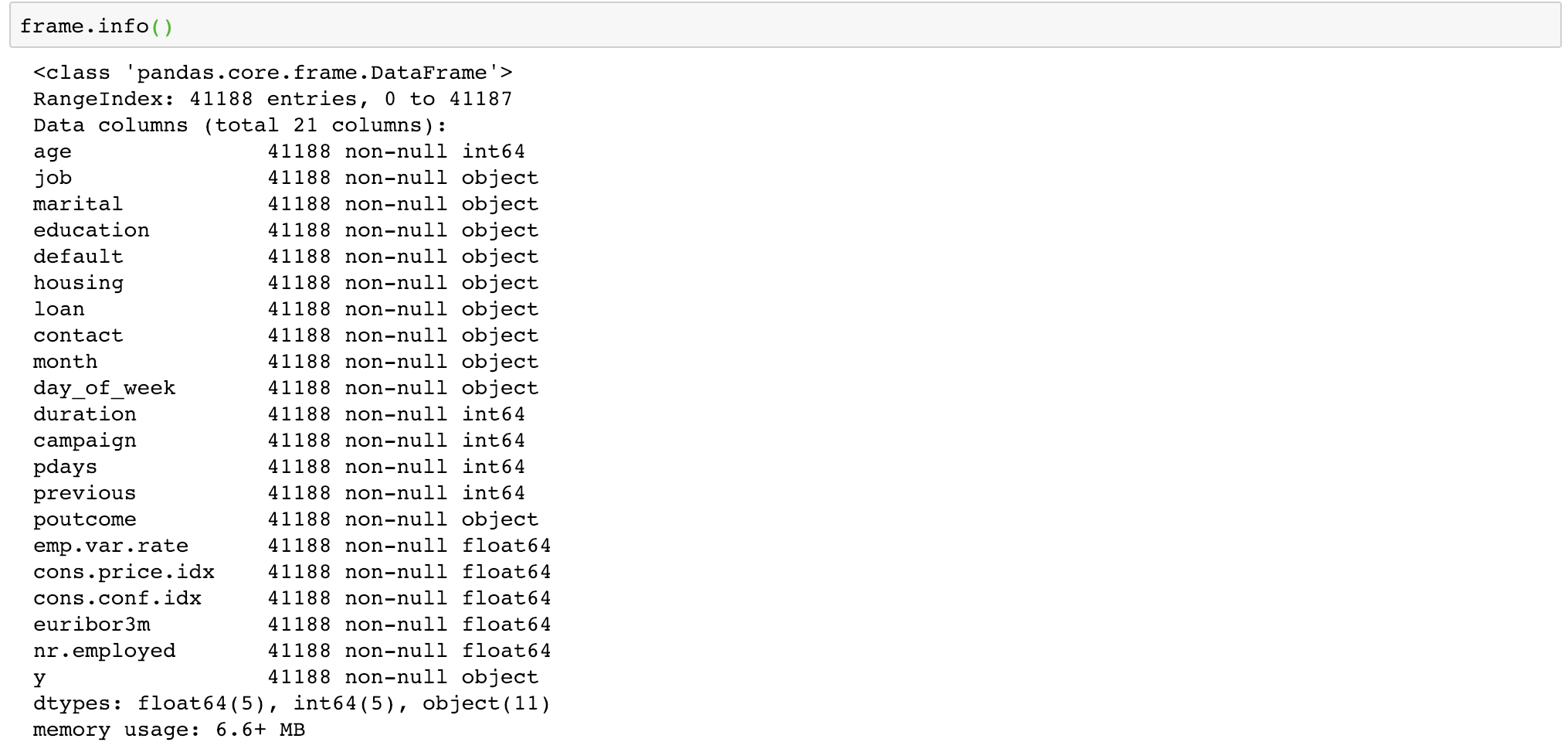
1. 查看数据的头几个数据，使用head(n)方法。



1. 随机查看几条数据，使用sample(n)方法。

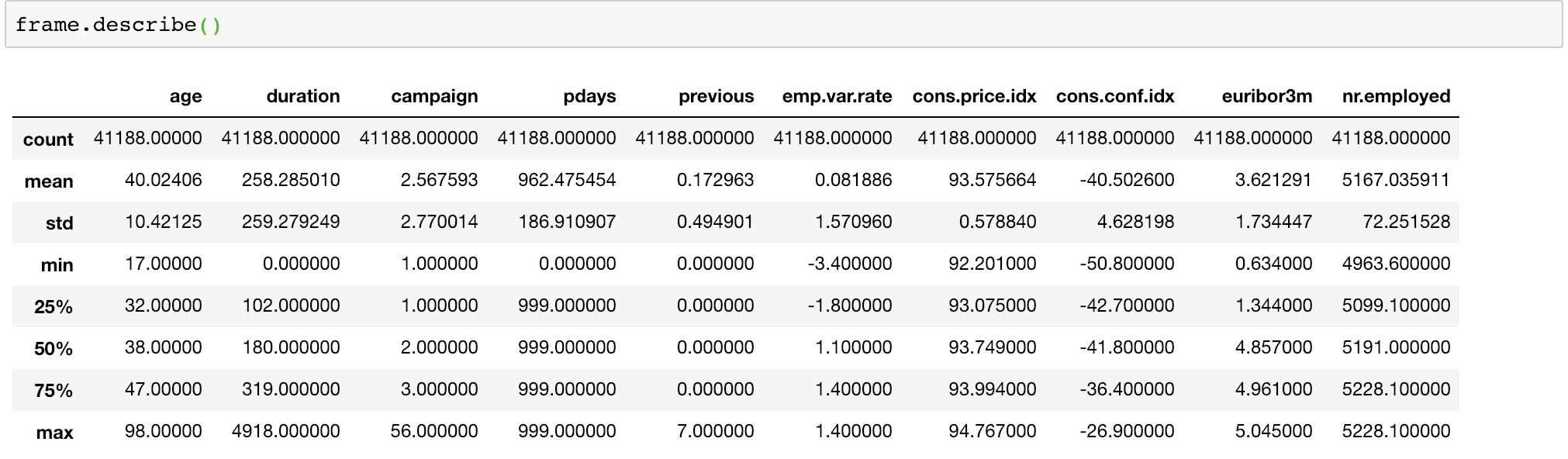


1. 查看有关数据的一些信息,使用info()方法。



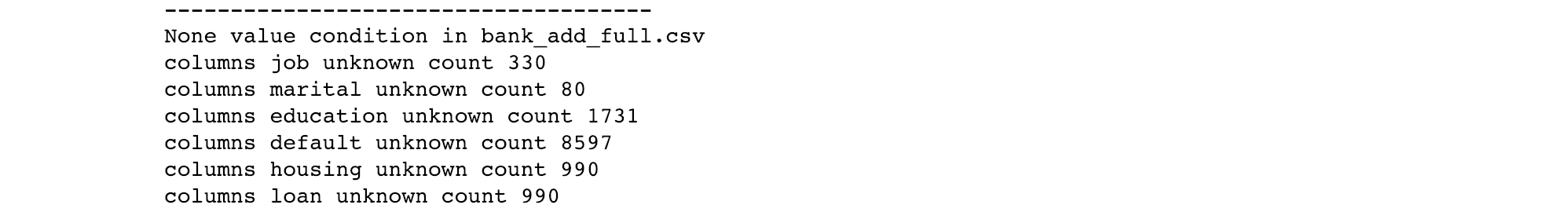
这里显示了共有多少行，多少列的数据。以及每一列的数据类型，还有使用多少内存。

1. 查看数值特征的相关信息，使用describe()方法。

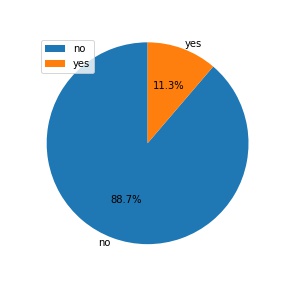


这里面包含了数值类特征的相关信息，主要是进行一些统计，包括个数，均值，标准差，最大值，最小值，分位数。

1. 根据数据缺失值的情况，查看数据缺失值的相关统计。



1. 查看预测y值的分布状况：



## 4 预处理方法

对数据简单查看之后，对于该数据集，我们知道其特征即有数值型的，也有类别型的。

对于数值型的特征，我们采取对列归一化的做法，使得数据的均值为0，标准差为1。

而对类别属性，有两种情况。

1. 无缺失值。使用one-hot Encoding的方式。
2. 有缺失值。对于缺失值少于500的数据直接删除；缺失值数量大于500的特征，将该特征作为目标特征用随机森林算法预测其类别。

其中有一个年龄的特征，预处理的方法是先将其划分分组，然后再归一化。

对于正负样本不均衡的的情况，我们采用smote算法对正样本进行过采样，使得正负样本比较均衡。具体方法为：对于每一个正例样本，在正例样本空间中用欧几里德距离算k个离其最短的正例样本，然后使用下列公式生成一个新的样本。

## 5 模型建立

### 5.1 对数几率回归模型

对于一个样例，我们要算的是其为正样本的概率，为负样本的概率。计算公式如下：

我们可以使用极大似然估计来求解值

似然函数为：

对数似然函数为:

将公式1-2代入1-3。用梯度下降法或者拟牛顿法，求解w，对于未知数据集，只要代入式1-2或者式1-3就可以算出对应的概率。

### 5.2 决策树模型

决策树是一种树型结构的判别树，其根据信息增益，信息增益比或者基尼指数来选择划分属性，以属性的值来作为边(离散值)或者满足某种判别条件(连续值)以此来建立决策树，对于预测数据只需要按照对应属性和属性的值来遍历这棵树，直到达到叶子节点，然后将叶子节点的类别做预测类别。

本实验使用信息增益的方法来选择属性，信息增益的算法如下：

|  |
| --- |
| 输入：训练数据集D和特征A  输出：特征A对训练数据集的信息增益g(D,A)   1. 计算数据集D的经验熵H(D) 2. 根据特征A对数据集D的经验条件熵H(D|A) 3. 计算信息增益 |

决策树的生成方法使用ID3算法，算法的流程如下：

|  |
| --- |
| 输入：训练数据集D，特征集A，阀值epsilon  输出：决策树T   1. 若D中所有的实例属于同一类,则将类作为该节点的类标记 2. 计算A中的各个特征对D的信息增益，选择信息增益最大的特征和划分的属性值value 3. 如果的信息增益小于阈值epsilon。则置T为单节点树，并将D中的实例数最大的类,作为该节点的类标记，返回T 4. 否则对根据value的值(分别为小于等于和大于value)划分为两个非空的集合 ，将中实例数最大的类作为标记，构建子节点，由该结点及其子节点构成树T，返回T 5. 对于第个子节点，以为训练集，以为特征集，递归调用(1) - (4),得到子树,返回T |

对于算法中value的选择采用二分的方法，即对于属性A,先对其值进行排序，然后计算候选值，候选值为,候选值即为value。

### 5.3 AdaBoost算法

输入：训练数据集 其中,；弱学习算法；

输出：最终的分类器

1. 初始化训练数据的权值分布
2. 对于m = 1, 2 ,…, M
3. 使用具有权值分布的训练数据集学习，得到基本分类器
4. 计算在训练数据集上的分类误差率
5. 计算的系数
6. 更新训练数据集的权值分布

其中是一个规范化因子

它使得成为一个概率分布。

1. 构建基本分类的线性组合

最终的分类器为

### 5.4 其他算法

sklearn，keras和xgboost中有大量的算法可以用于分类，在本实验中，也使用了这些库中的一些算法。具体如下表：

|  |  |
| --- | --- |
| 库名称 | 算法名称 |
| sklearn | 随机森林 |
| sklearn | 最近邻 |
| sklearn | 支持向量机 |
| keras | 神经网络 |
| xgboost | xgboost |

## 6 模型评估

在本实验中使用的评估方法是留出法，留出法直接将数据集D划分为两个互斥的集合，一个作为训练集S，一个作为测试集T，即,在S上训练出模型之后，用T来评估其测试误差，作为其泛化误差的估计。

在本实验中将数据集按照正反样本的比例从正反样例中各抽取5/6来作为训练集，抽取1/6来作为测试集。对于训练集，使用smote算法对正样本进行了过采样，对负样本进行了欠采样。使得训练集中正负样本的比例接近1:1。

因为在测试集中，正反样本的比例不均衡。所以用错误率与精度的性能度量方法，不能准确地反应分类器的泛化性能。在本实验中，我使用查准率，查全率与F1来作为分类器泛化性能好坏的标准。

对于二分类问题，可以根据测试集真实类别和学习器的预测类别的组合将样例分为：真正例(TP)，假正例(FP)，真反例(TN)，假反例(FN)。显然TP + FP + TN + FN = 样例总数。分类结果的混淆矩阵如下：

|  |  |  |
| --- | --- | --- |
| 真实情况 | 预测结果 | |
| 正例 | 反例 |
| 正例 | TP(真正例) | FN(假反例) |
| 反例 | FP(假正例) | TN(真反例) |

查准率P和查全率R分别定义为

查准率和查全率对于不同的应用侧重点不同。而F1是基于查准率和查全率的调和平均。定义如下:

## 7 结果可视化

因为本实验中使用查准率和查全率，F1值作为分类器性能的比较。自己手写的逻辑回归，决策树，Adaboost，并让其在训练上进行训练，然后在测试集上进行预测。得到的结果比较如下表1-3.

**表1-3**

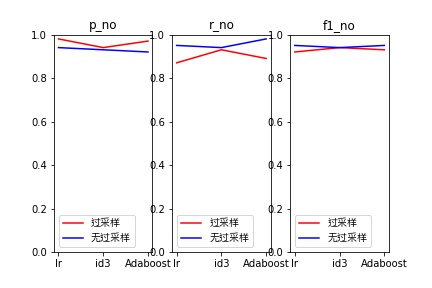
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 算法名称 | no | | | yes | | |
| P | R | F1 | P | R | F1 |
| 逻辑回归 | 0.98 | 0.87 | 0.92 | 0.45 | 0.85 | 0.59 |
| 决策树 | 0.94 | 0.93 | 0.94 | 0.50 | 0.55 | 0.53 |
| Adaboost | 0.97 | 0.89 | 0.93 | 0.47 | 0.75 | 0.58 |

若是没有对训练集进行过采样和欠采样处理，得到的结果如1-4.

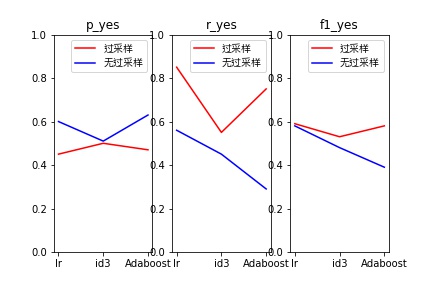
**表1-4**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 算法名称 | no | | | yes | | |
| P | R | F1 | P | R | F1 |
| 逻辑回归 | 0.94 | 0.95 | 0.95 | 0.60 | 0.56 | 0.58 |
| 决策树 | 0.93 | 0.94 | 0.94 | 0.51 | 0.45 | 0.48 |
| Adaboost | 0.92 | 0.98 | 0.95 | 0.63 | 0.29 | 0.39 |

对负样本结果可视化如下：



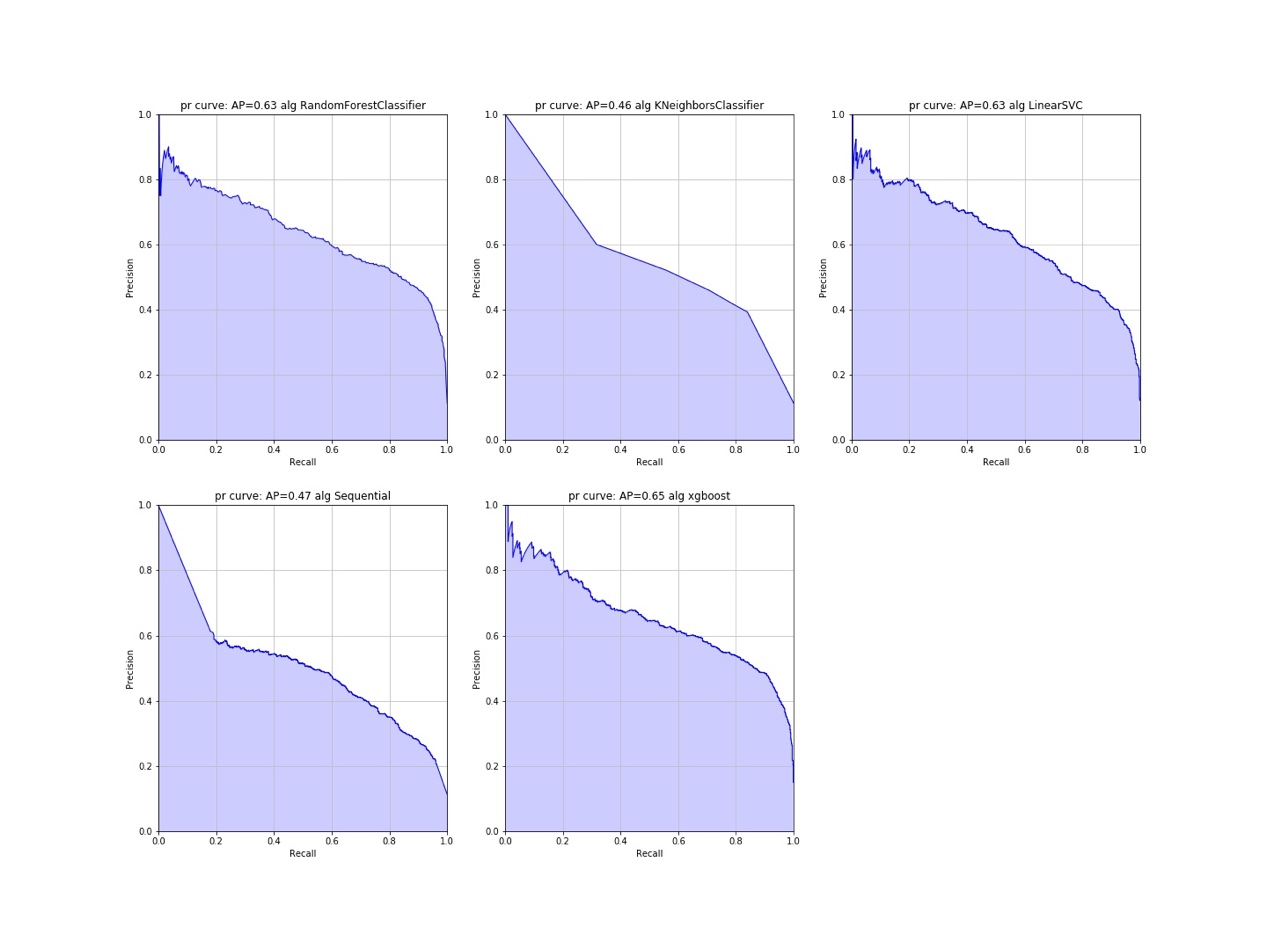
正样本结果可视化如下:



在本实验中，从上面的结果可以看出：

1. 正负样本不均衡确实会影响实验结果。
2. 过采样能够明显提高正样本的查全率，但降低了正样本的查准率。
3. 过采样提高了负样本的查准率，但降低了负样本的查全率。
4. 总的来说，过采样使得样本均衡后确实能够提高分类器的泛化性能。

使用sklearn，xgboost，和keras 的库中的算法得到的PR曲线如下：



从上图中也可以看到，xgboost表现是最好的，随后是随机森林，支持向量机。最后是最近邻和神经网络。