****

**《金融数据挖掘案例分析》**

**课程设计报告**

题 目： 根据声音数据预测性别

学生姓名： 王友情 学 号： 20161209032032

院 系： 信息管理与工程系

班 级： 16信息与计算科学二班

指导教师：  许美玲

信息管理与工程系

2019年06月14日

**根据声音数据预测性别**

**摘 要：**本文从kaggle平台下载数据集，该数据集源于扬声器采集的真实录音样本，共3168条数据，21列属性。对于判定性别，声音特性是基本指标，而某些特性却是判定性别的关键因素。本文为完成预测性别这个实验目标，分三步；第一步，对数据进行预处理，先处理缺失值；再用分层抽样划分测试集和训练集；最后对样本数据进行标准化，应用到KNN模型中；第二步，分别采用C50决策树和KNN算法对声音是男性还是女性进行建模和预测；第三步，分别对两种模型完成优化。

**关键词：**C50，KNN，分层抽样，建模，预测

**Predicting Gender Based on Sound Data**

**Abstract:** This paper downloads the data set from the kaggle platform. The data set is derived from the real recording samples collected by speakers. There are 3168 pieces of data and 21 columns of attributes. Sound characteristics are the basic indicators for determining gender, while some characteristics are the key factors for determining gender. In order to achieve the experimental goal of gender prediction, this paper divides into three steps: the first step is to pre-process the data, first deal with missing values; then divide the test set and training set by stratified sampling; finally standardize the sample data and apply it to KNN model; the second step is to use C50 decision tree and KNN algorithm to model and predict whether the voice is male or female; and the third step is to use C50 decision tree and KNN algorithm to predict whether the voice is male or female. The above two models are optimized respectively.

**Key words:** C50，KNN，stratified sampling，modeling，forecast

**目 录**

[一、数据挖掘目标 1](#_Toc12292868)

[二、数据理解 1](#_Toc12292869)

[三、数据预处理 2](#_Toc12292870)

[1、 缺失值处理 2](#_Toc12292871)

[2、 分层抽样 2](#_Toc12292872)

[3、 标准化（KNN算法需要） 3](#_Toc12292873)

[四、描述性分析 4](#_Toc12292874)

[1、 meanfreq、sd、median、Q25的密度曲线 4](#_Toc12292875)

[2、 Q75、IQR、skew、kurt的密度曲线 5](#_Toc12292876)

[3、 sp.ent、sfm、mode、centroid的密度曲线 5](#_Toc12292877)

[4、 meanfun、minfun、maxfun、meandom的密度曲线 6](#_Toc12292878)

[5、 mindom、maxdom、dfrange、modindx的密度曲线 7](#_Toc12292879)

[五、模型的建立 8](#_Toc12292880)

[1、 决策树C50 8](#_Toc12292881)

[2、 KNN模型 9](#_Toc12292882)

[六、模型评估及改进 9](#_Toc12292883)

[1、 决策树模型的评估及改进 9](#_Toc12292884)

[2、 KNN模型的评估及改进 11](#_Toc12292885)

[七、结论 11](#_Toc12292886)

[八、系统收获 11](#_Toc12292887)

[九、参考文献 12](#_Toc12292888)

[附录 13](#_Toc12292889)

# 一、数据挖掘目标

本次建模目标是根据声音和语音的声学特性来识别声音是男性还是女性，采用数据挖掘技术，通过声学对语音样本进行分析，构建根据声音识别性别的评价指标体系以及评价模型，实现对性别的预测，从而得出哪些属性是辨别性别的重要指标。

# 二、数据理解

该数据集由3168个录音语音样本组成，样本从扬声器中采集，其中男性声音和女性声音各有1584条，含有20个声学特性用来判定性别，数据集信息如下表：

表2-1数据集信息

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 属性 | 属性含义 | 属性类型 |
| 1 | meanfreq | 平均频率（千赫兹） | Number |
| 2 | sd | 标准偏差的频率 | Number |
| 3 | median | 频率中位数 | Number |
| 4 | Q25 | 频率第一分位数 | Number |
| 5 | Q75 | 频率第三分位数 | Number |
| 6 | IQR | 四分间距 | Number |
| 7 | skew | 斜偏度 | Number |
| 8 | kurt | 峰度 | Number |
| 9 | sp.ent | 谱熵 | Number |
| 10 | sfm | 频谱平坦度 | Number |
| 11 | mode | 模态频率 | Number |
| 12 | centroid | 频率中心 | Number |
| 13 | meanfun | 基本频率均值 | Number |
| 14 | minfun | 基本频率最小值 | Number |
| 15 | maxfun | 基本频率最大值 | Number |
| 16 | meandom | 主导频率均值 | Number |
| 17 | mindom | 主导频率最小值 | Number |
| 18 | maxdom | 主导频率最大值 | Number |
| 19 | dfrange | 主导频率范围 | Number |
| 20 | modindx | 调制指数 | Number |
| 21 | label | 标签（男性or女性） | Factor |

# 三、数据预处理

数据挖掘分析的数据往往是海量的，且常常含有一些干扰数据，不适合直接分析，所以需要对数据进行预处理。

## 1、 缺失值处理



图3-1 缺失值

由图3-1可知，本数据集中不含缺失值，可直接进行下一步。

## 2、 分层抽样

由于最后一列目标变量本身就是因子型，所以不需要转化为因子型，如图3-2：

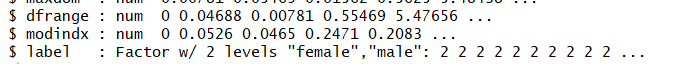


图3-2 目标变量为因子型变量

然后开始划分测试集和训练集，选用分层抽样是为了让训练集和测试集拥有和原始数据相同的分布，这样抽样获得的训练集和测试集均可以代表总体，符合抽样的有效性，本项目采用7：3分层抽样划分训练集和测试集：

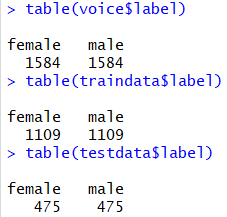


图3-3 测试集和训练集数据

由图3-3可知，总数据集中含有1584名女性，1584名男性，经过分层抽样，按照7：3的比例，训练集均具有1109名女性和1109名男性，测试集均具有475名女性和475名男性。

## 3、 标准化（KNN算法需要）

用KNN算法进行建模时，需要对数据进行标准化，如果没有这个步骤，那么小数量级的变量对最后测试出来的结果影响很小，会导致预测结果不准确。

对本数据集用preProcess()函数进行标准化后得到样本voice，同时也对测试集和训练集进行标准化得到train1和train2。

# 四、描述性分析

通过绘制密度曲线图查看自变量的重要性，

首先把自变量的列名赋值给cn（不包括最后一列）；因为图片的内容比较多，所以需要对这样的图像进行函数的封装，绘制密度函数首先需要了解两个函数：prase()函数将字符串转化为表达式，eval()函数对表达式求解。函数如图4-1：

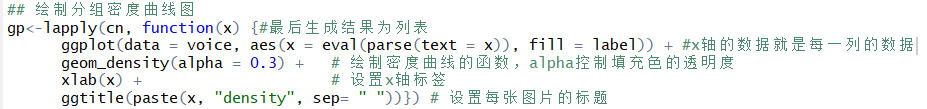


图4-1 绘制密度曲线的函数

最后通过multiplot()函数进行展示，蓝色代表男性，粉红代表女性：

## 1、 meanfreq、sd、median、Q25的密度曲线

Meanfreq代表属性：平均频率、sd代表属性：标准偏差的频率、

Median代表属性：频率中位数、 Q25代表属性：频率第一分位数。

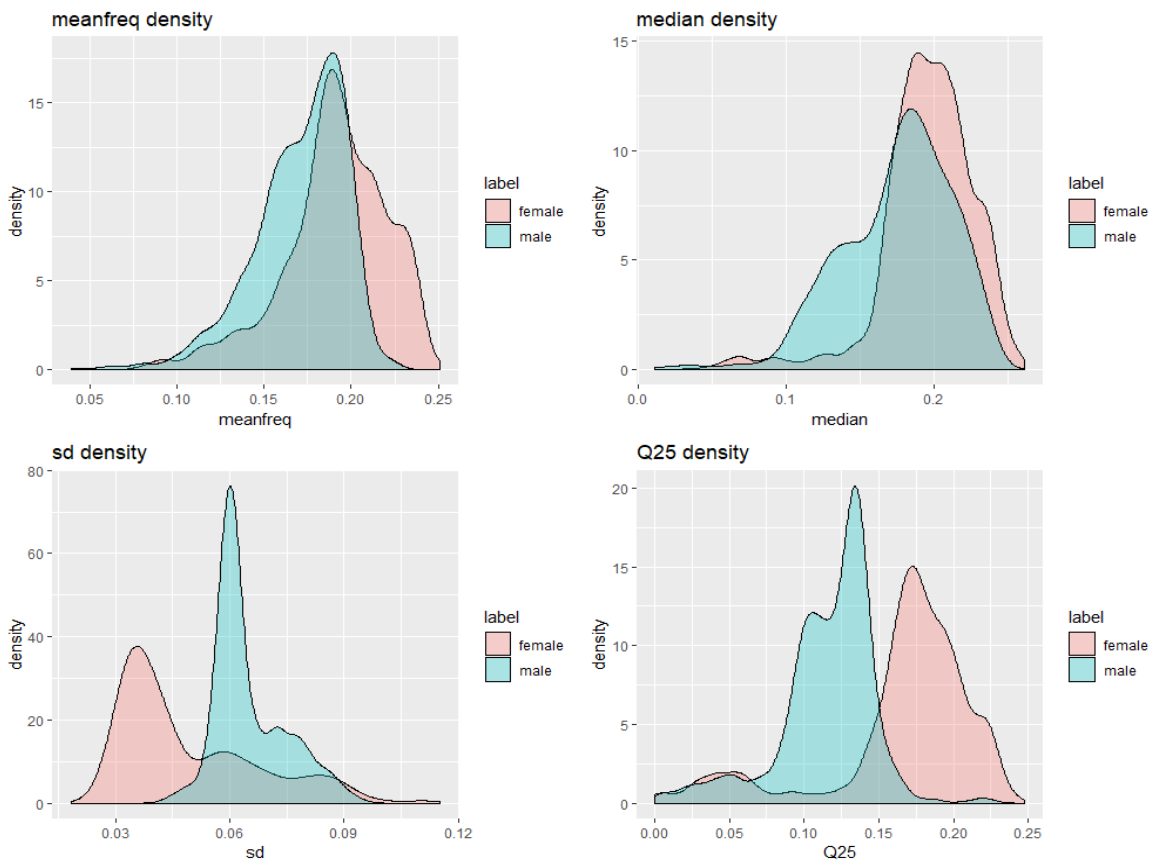


图4-2 绘制密度曲线的函数

由图4-2可以看出，标准偏差的频率（sd）和频率第一分位数（Q25）这两个变量对区分类别还是比较重要的。

## 2、 Q75、IQR、skew、kurt的密度曲线

Q75代表属性：频率第三分位数、IQR代表属性：四分间距、

skew代表属性：斜偏度、 kurt代表属性：峰度。

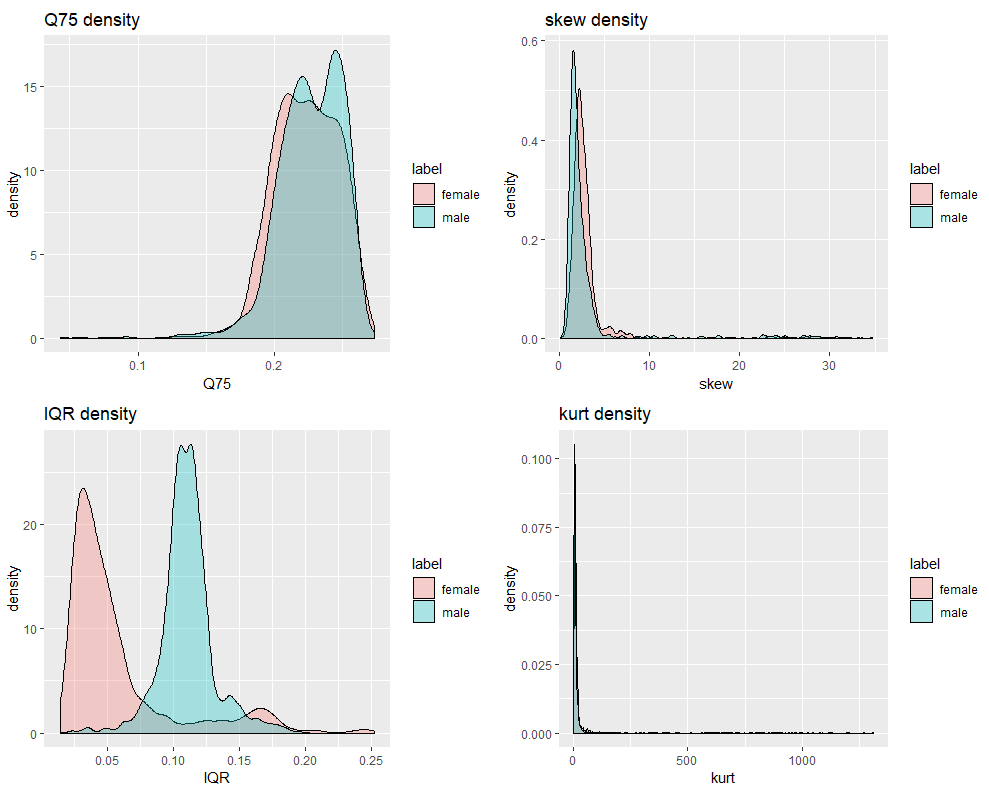


图4-3 绘制密度曲线的函数

由图4-3可看出，这四幅图里，四分间距（IQR）这个变量对区分类别还是比较重要的。

## 3、 sp.ent、sfm、mode、centroid的密度曲线

sp.ent代表属性：谱熵、sfm代表属性：频谱平坦度、

mode代表属性：模态频率、 centroid代表属性：频率中心。

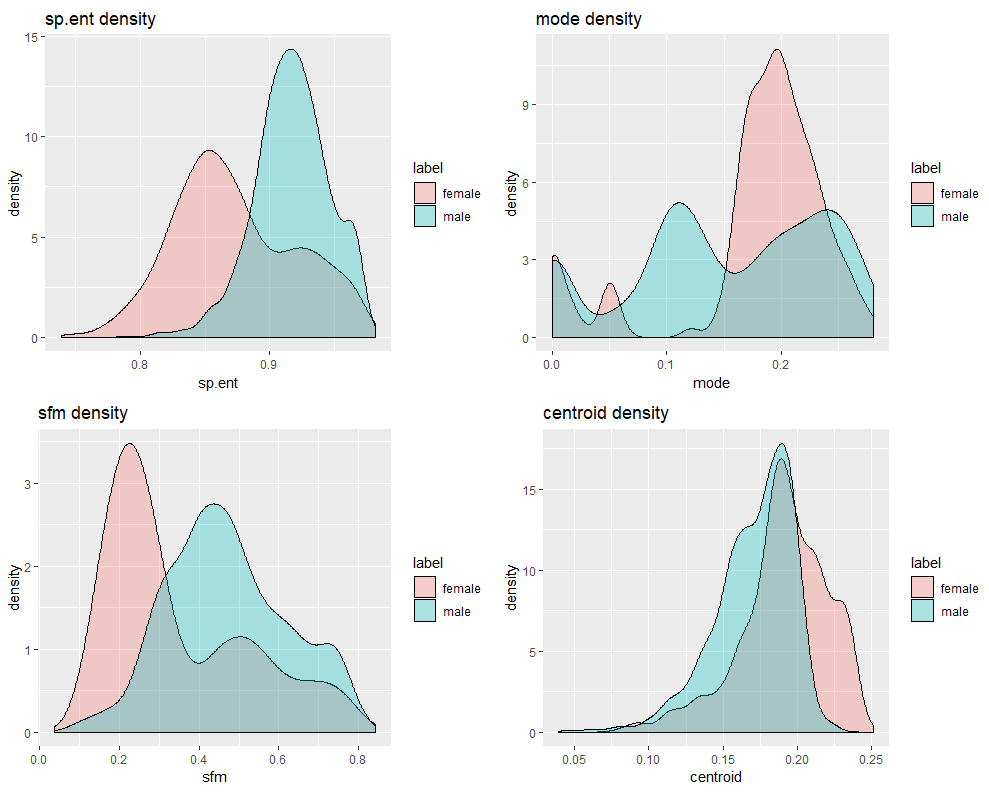


图4-4 绘制密度曲线的函数

由图4-4可看出，这四幅图里，谱熵（sp.ent）和频谱平坦度（sfm）这两个变量对区分类别还是比较重要的。

## 4、 meanfun、minfun、maxfun、meandom的密度曲线

meanfun代表属性：基本频率均值、minfun代表属性：基本频率最小值、

maxfun代表属性：基本频率最大值、meandom代表属性：主导频率均值。

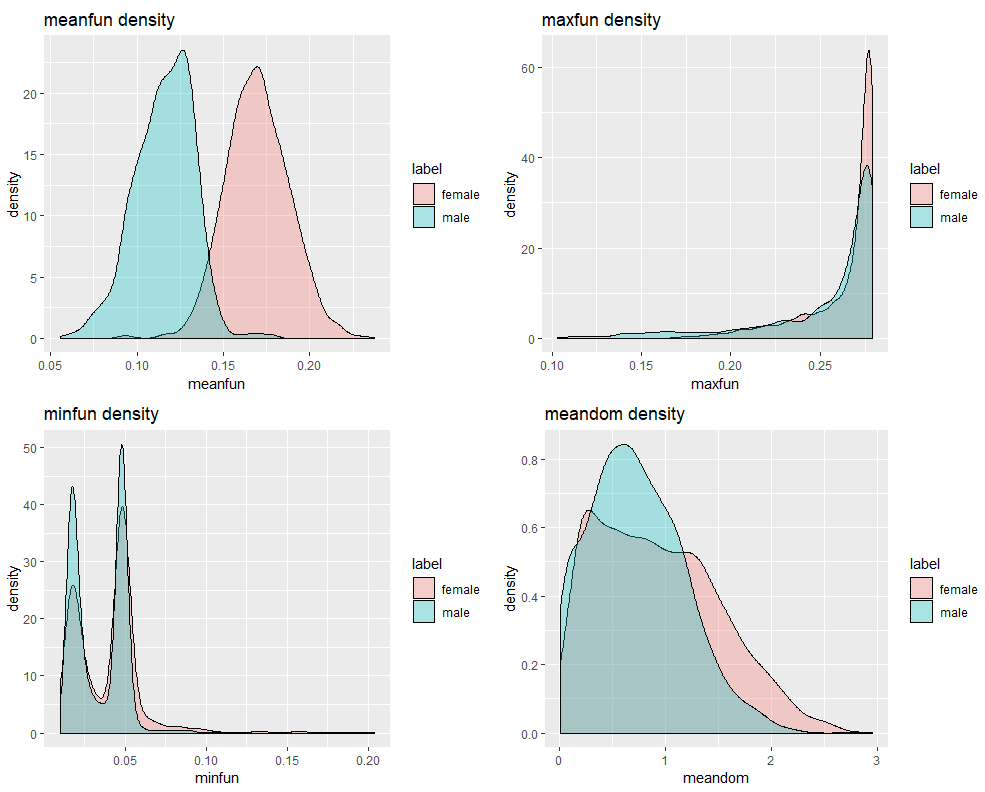


图4-5 绘制密度曲线的函数

由图4-5可看出，这四幅图里，基本频率均值（meanfun）这个变量对区分类别是比较重要的。

## 5、 mindom、maxdom、dfrange、modindx的密度曲线

mindom代表属性：主导频率最小值、maxdom代表属性：主导频率最大值、

dfrange代表属性：主导频率范围、modindx代表属性：调制指数。

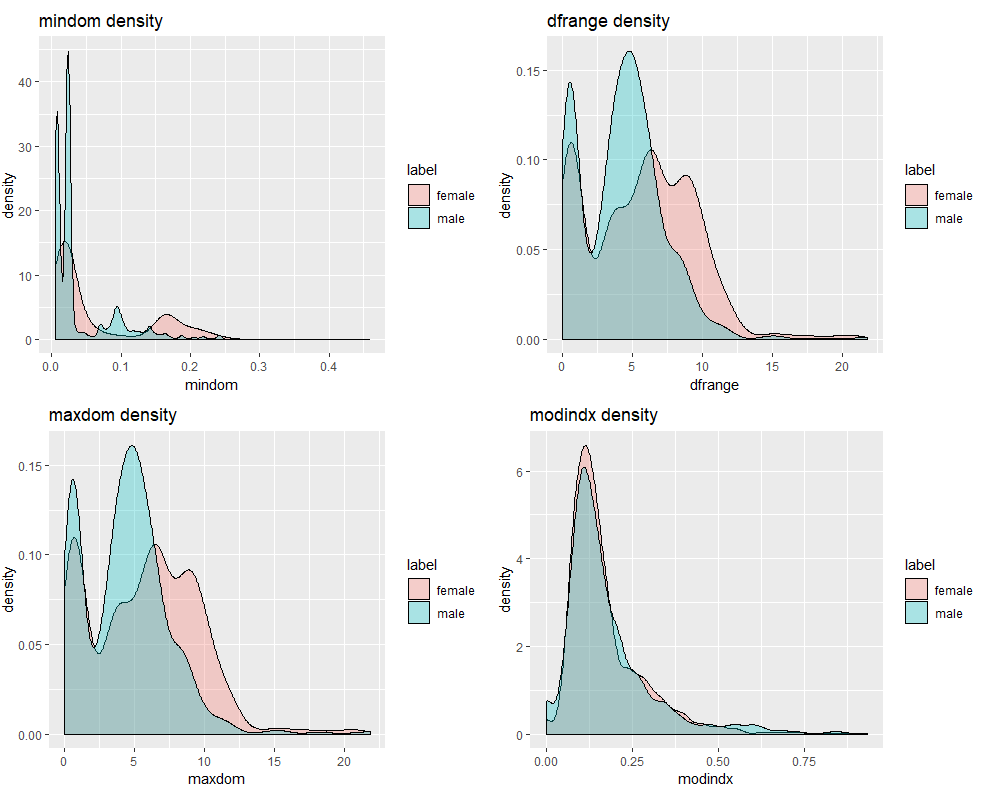


图4-6 绘制密度曲线的函数

# 五、模型的建立

## 1、 决策树C50

在数据预处理阶段进行训练集和测试集划分之后，首先把因变量转化为因子型，方便预测模型中进行预测；然后基于C50模型，通过训练集构建决策树，如图5-1：

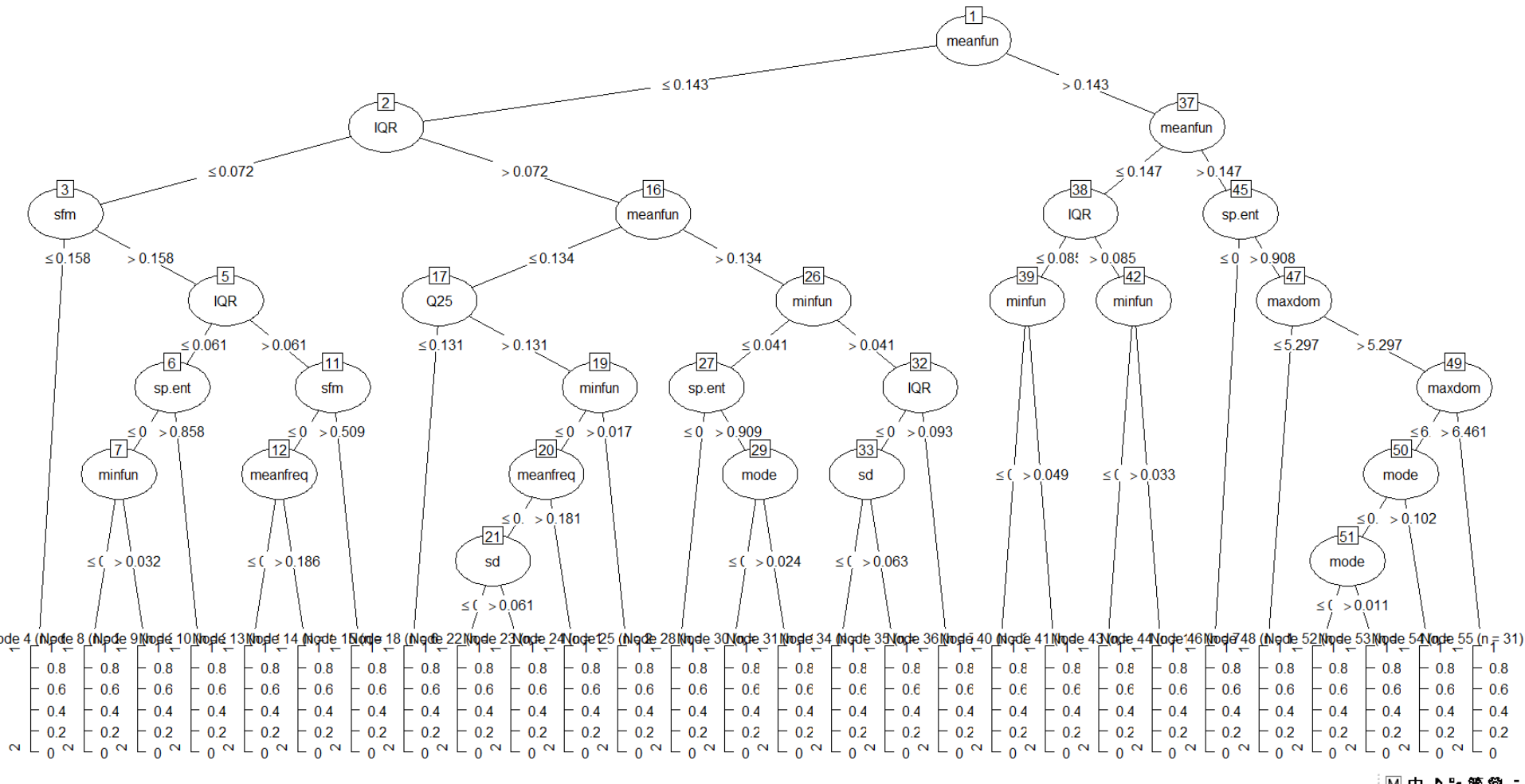


图5-1 未经处理的C50决策树

最后在测试集上预测，将测试集计算出来的预测的值和实际的值整合到一起，最后输出混淆矩阵，并算出预测精度，如图5-2：

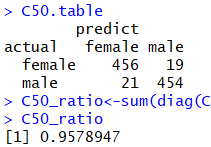


图5-2 未经处理C50模型的混淆矩阵和准确率

## 2、 KNN模型

在KNN建模过程中，最重要的就是k值的确定，一般K值在3-10之间，先随便取一个K值5，通过knn()函数建立模型，查看准确率，如图5-3：

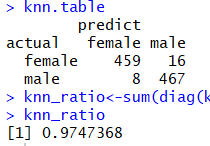


图5-3 未经处理KNN模型的混淆矩阵和准确率

# 六、模型评估及改进

## 1、 决策树模型的评估及改进

在C5.0Control()这个函数里设置相关参数，通过不断调整mincases的值对决策树进行剪枝；在C5.0函数中通过设置迭代次数trails的值，控制自适应增强循环的次数，可以降低关于测试集大约25%的概率。

通过以上两个步骤完成对决策树模型的优化。

模型中参数说明如表6-1：

表6-1决策树算法参数说明

|  |  |
| --- | --- |
| 参数名称 | 含义 |
| trials | 迭代次数 |
| winnow | 建模之前是否对变量进行特征选择 |
| CF | 剪枝时的置信度 |
| minCases | 最小分支 |
| noGlobalPruning | 若等于F，则为对树进行剪枝 |

对树进行剪枝和迭代的主要代码如下：

x<-C5.0Control(subset =F,CF=0.25,winnow=T,noGlobalPruning=F,minCases =10)

c50\_tree1=C5.0(label~.,traindata,trials=5,control =x)

summary(c50\_tree1)

查看模型结果，有14个属性被剪掉，还剩6个属性决定性别，如图6-1：

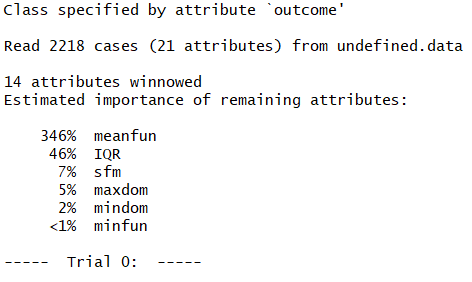


图6-1 优化后的C50决策树模型结果

优化后的C50决策树以及准确率，如图6-2、图6-3：

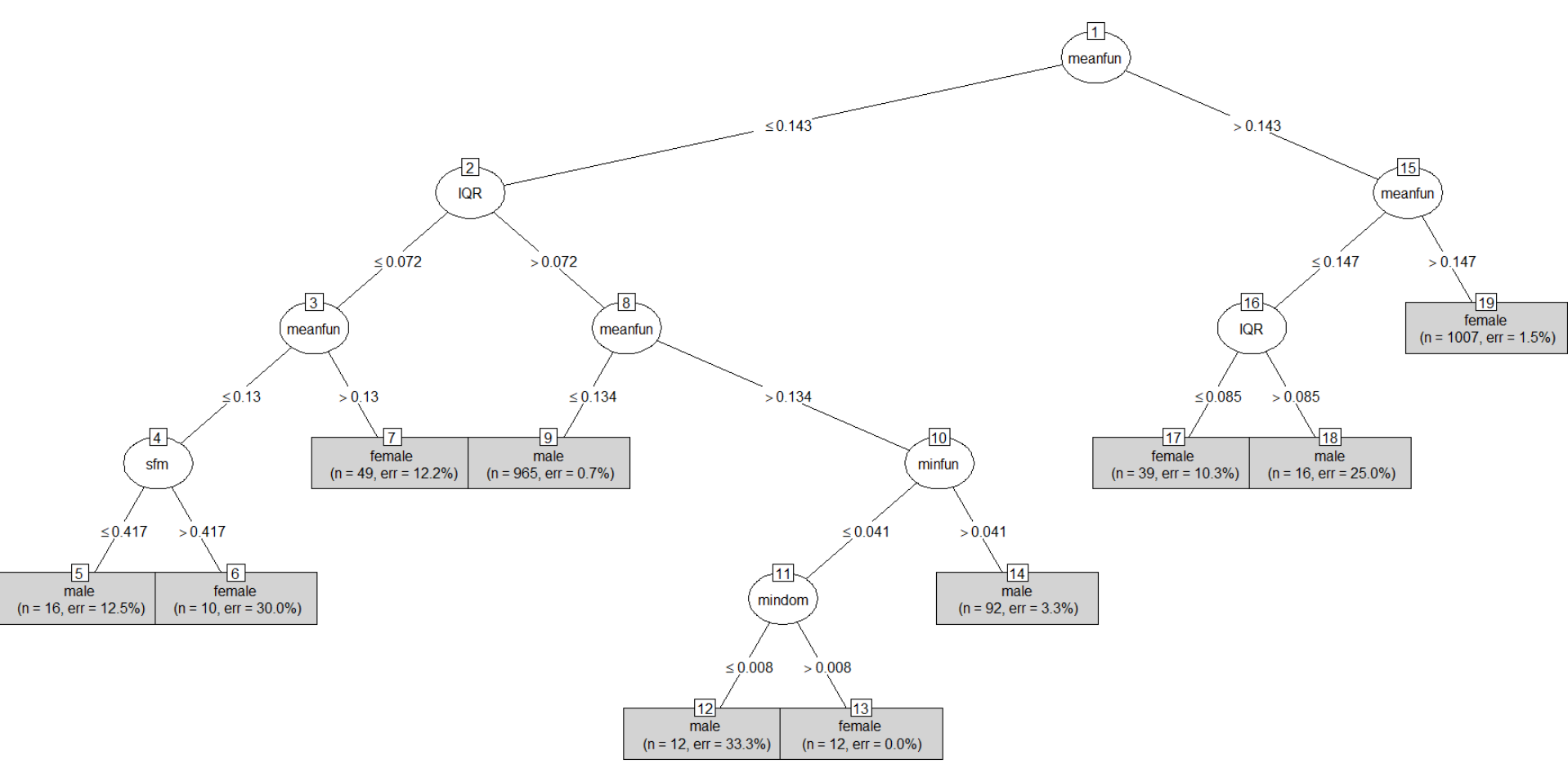


图6-2 优化后的C50决策树

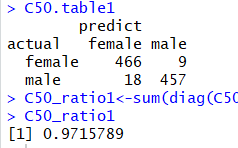


图6-3 优化后C50决策树的混淆矩阵和准确率

## 2、 KNN模型的评估及改进

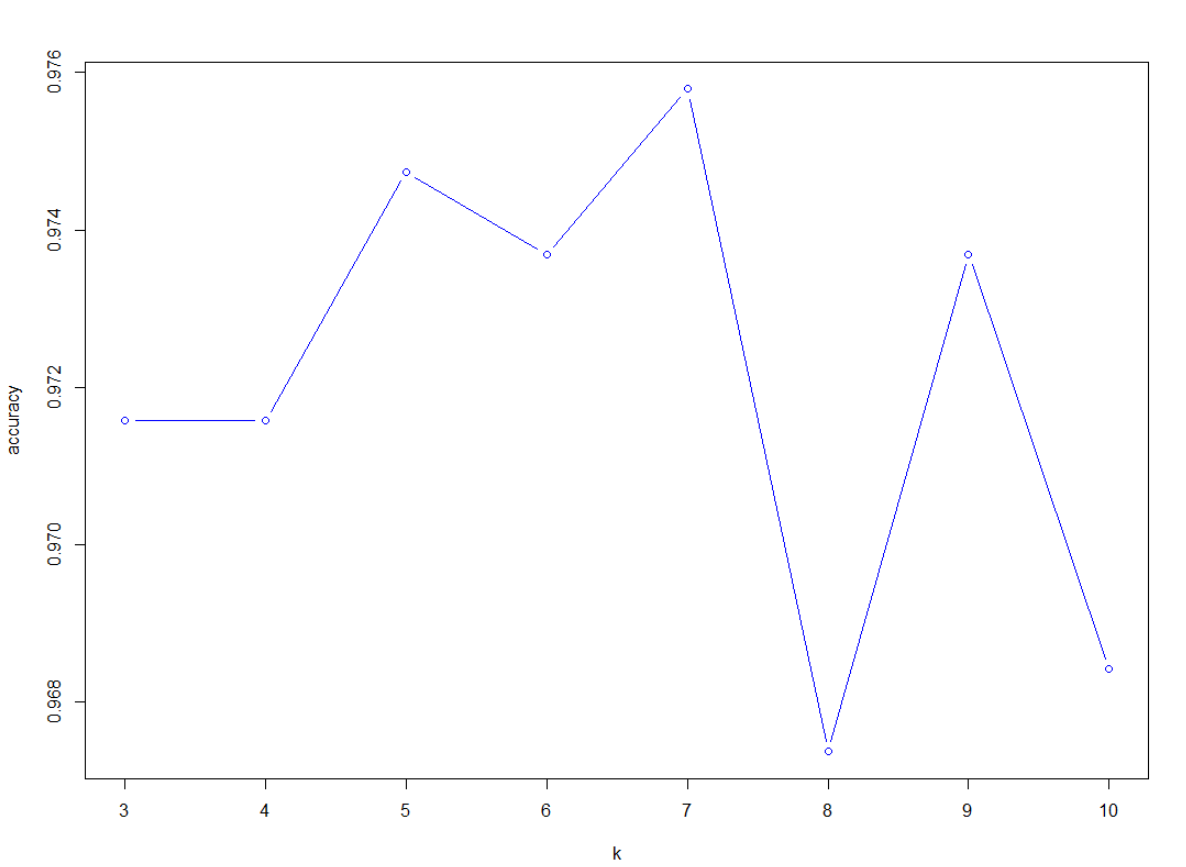
通过选取使模型准确率最高的k值来提高模型的准确率，最优k值如图6-4： 

图6-4 选取最优k值

由图6-4可以看出，k=7时，模型准确率是最高的。

优化后模型准确率如图6-5：

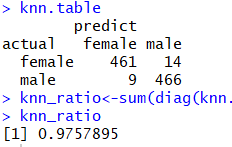


图6-5 改进后KNN模型的混淆矩阵和准确率

展示C50和KNN有哪些不同的预测值，如图6-6：

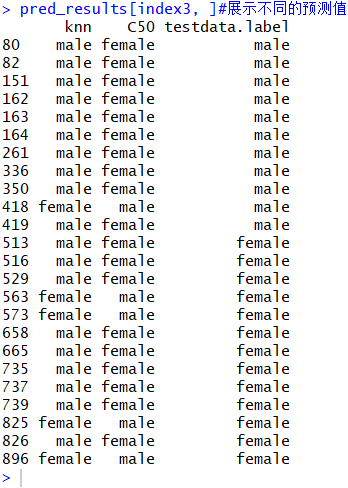


图6-6 不同的预测值

# 七、结论

由决策树可以看出，根据基本频率均值meanfun这个属性，最能据声音判断出是男性还是女性，然后依次是四分间距IQR、频谱平坦度sfm、主导频率最大值maxdom、主导频率最小值mindom、基本频率最小值minfun,，决策树预测出来的结果与真实数据画出的密度曲线十分相似。

比较决策树C50和KNN两种模型，KNN准确率略高。但我们选择C50决策树来预测这个数据集中的性别更为合适，因为决策树的目标是通过学习从数据特征推断得到的决策规则，构建一个可以预测目标变量的决策模型，使得本实验可以根据决策树判断出某条声音数据是男性还是女性，但是其他模型只能给出一个准确率，并不能达到预测的功能。

# 八、系统收获

本次实验遇到了很多问题，主要有以下几个方面：

1、当划分测试集和训练集时，刚开始是随机抽样进行划分，还是存在误差的，且不适用于样本数较多的情况，后来发现可以使用分层抽样，不仅适用于数据较多的情况，且能充分保证样本结构与总体的一致。

2、用C50算法构建决策树时，由于变量比较多，没剪枝之前树太大，剪枝后，留下了比重较大的六个属性，且增加迭代次数可以增加准确率。

3、KNN算法时，计算K的值也是很大的一个问题，一般K值都在3-10之间，随便选一个就好，但是作为模型优化，我最后选择用精确度最高时的K值。

一个项目下来，总的来说，收获还是很大的，对R语言的理解更升了一层。

# 九、参考文献

[1] 汤银才. R语言与统计分析[J]. 北京：高等教育出版社,2008.11.

[2] 薛毅，陈丽萍. 统计建模与R语言[D].北京：清华大学出版社,2007.04.

# 附录

library(ggplot2) # ggplot()

library(caret) # createDataPartition() / train()

# 读取数据

voice <- read.csv('C://Users//friendship-W//Desktop//voice.csv')

str(voice)# 查看数据集的基本数据结构

sum(is.na(voice))# 查看数据的缺失状况

cn <- colnames(voice[-21]) # 统计数据的列名(去除最后一列)

colnames(voice)

## 绘制分组密度曲线图

gp<-lapply(cn, function(x) {#最后生成结果为列表

ggplot(data = voice, aes(x = eval(parse(text = x)), fill = label)) + #x轴的数据就是每一列的数据,填充颜色根据男女的性别来填充

geom\_density(alpha = 0.3) + # 绘制密度曲线的函数，alpha控制填充色的透明度

xlab(x) + # 设置x轴标签

ggtitle(paste(x, "density", sep= " "))}) # 设置每张图片的标题（列名+density）,paste连接字符串的函数，

library(Rmisc) # multiplot()

multiplot(gp[[1]], gp[[2]], gp[[3]], gp[[4]], cols = 2) # 展示前4张图片

multiplot(gp[[5]], gp[[6]], gp[[7]], gp[[8]], cols = 2)

multiplot(gp[[9]], gp[[10]], gp[[11]], gp[[12]], cols = 2)

multiplot(gp[[13]], gp[[14]], gp[[15]], gp[[16]], cols = 2)

multiplot(gp[[17]], gp[[18]], gp[[19]], gp[[20]], cols = 2)

set.seed(1023) # 设定随机种子

# 按照性别以7:3的比例进行分层抽样，并且以矩阵的形式返回索引

index <- createDataPartition(voice$label, p = 0.7, list = FALSE)

traindata <- voice[index, ] # 创建训练集

testdata <- voice[-index, ] # 创建测试集

#决策树C50

library(C50)

C50\_tree<-C5.0(traindata[-21],traindata$label)

plot(C50\_tree)

C50\_predict<-predict(C50\_tree,testdata)#在测试集上预测

C50.table<-table(actual=testdata$label,predict=C50\_predict)

C50.table

C50\_ratio<-sum(diag(C50.table))/sum(C50.table)

C50\_ratio

#优化

x<-C5.0Control(subset =F,CF=0.25,winnow=T,noGlobalPruning=F,minCases =10)

c50\_tree1=C5.0(label~.,traindata,trials=5,control =x)

summary(c50\_tree1)

plot(c50\_tree1,uniform=T,branch=0.1,margin=0.2,main= "Classification",compress=T,type="simple")

C50\_predict<-predict(c50\_tree1,testdata)#在测试集上预测

C50.table1<-table(actual=testdata$label,predict=C50\_predict)

C50.table1

C50\_ratio1<-sum(diag(C50.table1))/sum(C50.table1)

C50\_ratio1

#标准化

library(caret)

standard<-preProcess(voice,method='range')

voice1<-predict(standard,voice)

train1<-voice1[index,]

test1<-voice1[-index,]

#KNN预测，确定k的值

library(class)

results=c()

for(i in 3:10) { #一般而言,k在3到10之间取值

set.seed(1500)

pred\_knn<-knn(train1[-21],test1[-21],train1$label,i)

Table<-table(pred\_knn,testdata$label)

accuracy<-sum(diag(Table))/sum(Table)

results<-c(results,accuracy)

}

plot(x=3:10,y=results,type='b',col='blue',xlab='k',ylab='accuracy')

#KNN建模

set.seed(1500)

pred\_knn<-knn(train=train1[-21],test= test1[-21],cl=train1$label, k=7)

knn.table<-table(actual=test1$label,predict=pred\_knn)

knn.table

knn\_ratio<-sum(diag(knn.table))/sum(knn.table)

knn\_ratio

##比较几种算法

#KNN和朴素贝叶斯比较

pred\_results<-data.frame(knn=pred\_knn,nb=nb\_predict,testdata$label)

index2<-which(pred\_results$knn!=nb\_predict)

pred\_results[index2, ]#展示不同的预测值

#KNN和C50比较

pred\_results<-data.frame(knn=pred\_knn,C50=C50\_predict,testdata$label)

index3<-which(pred\_results$knn!=C50\_predict)

pred\_results[index3, ]#展示不同的预测值