# 交通信息挖掘与融合-作业5

1752221 潘帅鑫

## 一，关联

列举强关联原则：

通过Aprior算法得出频繁项集为{‘O’,’K’,’E’};

则根据0.6和0.8的最小支持度和置信度，可以得到强关联规则为：方括号内依次为支持度和置信度

OKE [0.6,1]

EOK [0.6,1]

## 二，分类

#### 1.决策树算法：

数据导入，导入后，利用for循环，将交通违章情况按要求分为3类：0、1、2。利用sample函数，将数据集以8：2的形式随机分为训练集和测试集。

运用R语言，使用rpart包，用rpart函数来生成决策树。

1）.rpart（）函数参数设置如下：

rpart(Q2 ~ Q4+Q5+Q14\_2+Q14\_3+Q14\_6+Q14\_10+Q14\_12+Q15\_2

+Q15\_5+Q15\_7+Q15\_8+Q15\_11+Q15\_13,

data = train\_data,method="class",parms=list(split="information"))

其中Q2~Q4+Q5…表达式是表示因变量为Q2，自变量为Q4+Q5…，设置理由：

Q2表示记录的交通违章次数，划分为3档后作为因变量；

Q4是驾驶车的频率。驾驶车越频繁，会随着车技的熟练而减小发生违章的概率，同时驾驶的频率大意味着基数大，违章情况也会有一定增加。

Q5是驾驶车的用途。交通的需求有刚性和弹性之分，常理来说，刚性需求例如上班上学，由于时间的限制，因此发生闯红灯和超速行驶的可能性会增加，违章次数也会相应增多。

Q14\_2、3、6、10、12分别是对Q14数据进行主成分分析，解释度最高的几个成分中系数最大的，能够较好的反映出Q14问题的答案。

Q15\_2、5、7、8、11、13同上，也是主成分分析中解释度较高的几个问题。

data = train\_data,这里设置的是训练的数据集；

method="class"，因为因变量我们设置的是Q2这一列，所以采用class的方法；

parms=list(split="information")，拆分功能的可选参数，用split指的是拆分索引，分割标准选用的是information gain的方式，公式是Entropy=-p\*log2p-q\*log2q，

当所有的资料都是相同一致，entropy=0，不用分类，当各有一半不同，则entropy=1，就是不纯净的资料需要分类。最常用的是gini和information，二者的区别在于information一次可以产生多个节点，gini只能产生两个，因此选用information。

决策树剪枝及剪枝判断，输出结果，选择xerror最小的cp值，绘制树图，

2）prune(data\_decisiontree, cp=0.010000)

data\_decisiontree是1）产生的结果决策树;cp=0.010000是指的修剪对象的复杂度参数，此时对决策树进行终止，a得到经修剪后的决策树。

利用rpart.plot绘制树图，

3）rpart.plot(data\_decisiontree\_prune,branch=1,fallen.leaves=T,cex=0.6)

第一个参数是经修剪后的决策树，是rpart对象;

type=1,指的是标记所有节点，不仅仅是叶子；

fallen.leaves=T,指的是将叶节点放置再图的底部；

cex=0.6，表示文本大小。

然后根据分类的决策树对测试集进行测试，

4）pre\_decisiontree <- predict(data\_decisiontree\_prune,

newdata = test\_data,

type="class")

参数1表示建立的分类模型，就是上述得到的rpart对象；

newdata=test\_data，指的是用于测试的数据集；

type=”class”，class表示返回预测结果；

输出混淆矩阵，及测试值与真实值的结果

5）obs\_p\_decision\_cart = data.frame(prob=pre\_decisiontree,

obs=test\_data$Q2)

6）table(test\_data$Q2,pre\_decisiontree,dnn=c("真实值","预测值"))

以上是分别进行真实值和预测值的融合和输出。

然后利用pROC，绘制出ROC图像。

#### 2.朴素贝叶斯分类法

导入数据与选择变量与上述大致相同，然后利用klaR包里的NaiveBayes函数进行朴素贝叶斯分类，

1）data\_model <- naiveBayes(Q2~.,data = train\_data,laplace = 1)

Q2~.是和上述类似的公式，但是不允许互动，因此我们在选择数据时，将所需的列选出成立一个新的数据集。这表示Q2为因变量，其他为自变量；

data = train\_data，是选出所用的列合成的新的测试数据集；

在处理时，需要将字符类型的变量转化为factor，因此采用下述函数：

2）train\_data$Q2 <- as.factor(train\_data$Q2)

laplace = 1;进行拉普拉斯校准，对每组+1，避免出现概率为0的情况

得到朴素贝叶斯分类的模型，用以预测测试集：

3）data\_predict <- predict(data\_model,newdata = test\_data)

参数1为模型方法，即是上述得到的朴素贝叶斯算法模型；

参数2是用以测试的测试集test\_data。

将得到的测试结果的测试值和真实值输出，

4）table(test\_data$Q2,data\_predict,dnn=c("真实值","预测值"))

参数1为真实值，即是测试数据的Q2列；

参数2为预测结果，默认返回概率最大代表的值。

然后同决策树算法，绘制出ROC图。

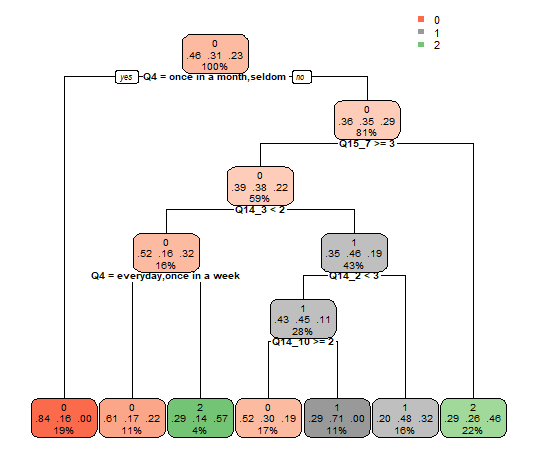
#### 3.测试结果：

由于每次都是在总的数据集中随机选取测试集和训练集，因此每次结果都有些许差别，控制两种算法使用相同的训练集和测试集，两次随机分测试集和训练集，得到三组结果如下所示：

测试结果1：

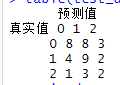
朴素贝叶斯：

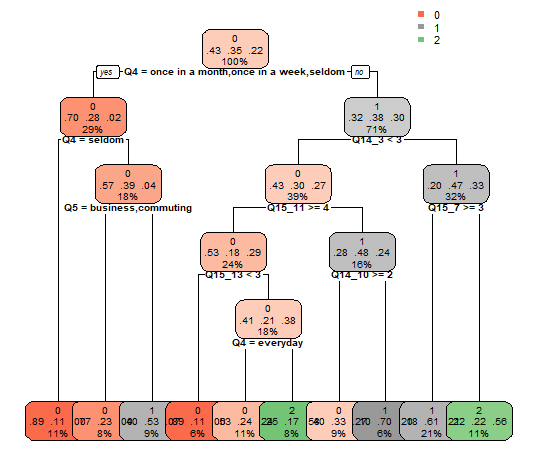
决策树： 



测试结果2：

朴素贝叶斯：

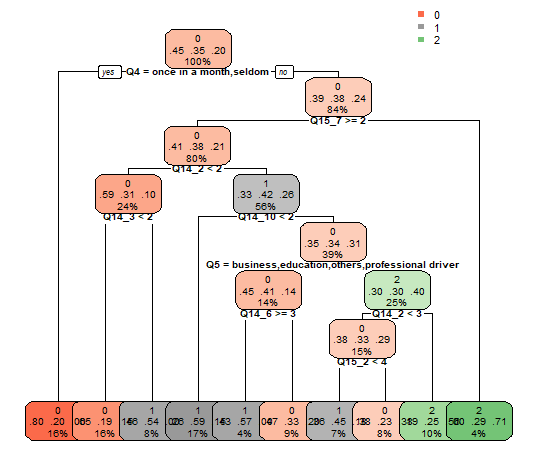
决策树： 



测试结果3：

朴素贝叶斯：

决策树： 



从上述结果来看，两种结果都对0的判断准确率比较高，对1和2的判断准确率相对较低。驾驶的频率很大程度上能够判断出违章情况，Q14\_3和Q15\_7对整个的判断也有着很重要的影响。

## 4.模型优劣对比

准确度：

朴素贝叶斯的准确率为：22/40=55%；17/40=42.5%，19/40=47.5%平均值为：48.3%

决策树的准确率为：18/40=45%；19/40=47.5%；21/40=52.5%平均值为：48.3%

从准确率来看，两者的判断准确率大致相同，相差不多。

朴素贝叶斯的kappa系数为0.235；0.06；0.185；平均值：0.16

决策树的kappa系数为0.143；0.17；0.236；平均值：0.183

从一致性系数来看，决策树的kappa系数平均值相对较高，相对来说，决策树的判断准确率略高于朴素贝叶斯。

运算效率：

两者均采用R包里的函数，实现过程相对来说，决策树由于需要绘制决策树图，因此更复杂一些。但同时决策树的自变量表达式可以互动，更加灵活。而朴素贝叶斯的函数表达式不允许互动，因此需要在写表达式前先将数据集进行剔除，剩余需要用到的数据，而且数值型和字符型需要有所区分，将字符型的自变量先转化为factor，才能进行区分。

鲁棒性：

由于我们三次实验采用的测试集和训练集是不同的，得到的准确性如上所示，由准确性和kappa系数来看，朴素贝叶斯的算法波动更大，而决策树算法的波动相对较小，相对稳定，因此可以得出结论，决策树的鲁棒性相对更好。