1

1.1 见代码

1.2 P(MAX_SEV_IR=1 | (WEATHER_R=2,TRAF_CON_R=0))=1/(5+1)=1/6

1.3 P(MAX_SEV_IR=1 | (WEATHER_R=2,TRAF_CON_R=0))

=(1/4 * 1 * 1/3)/((1/4 * 1 * 1/3)+(3/4 * 2/3 * 2/3))

=1/5

1.4 见代码

1.5 AccidentModel<-

Accidents[,c('HOUR_I_R','ALIGN_I','WRK_ZONE','WKDY_I_R','INT_HWY','RELJCT_I_R','REL_RWY_R','TRAF_CON_R','TRAF_WAY','MAX_SEV_IR')]

#以 HOUR_I_R,ALIGN_I...'MAX_SEV_IR'建立模型

set.seed(1000)

#固定随机数种子为 1000

RowNum <- nrow(AccidentModel)

#取模型中行数

SampleIndex <- sample(1:RowNum,round(RowNum*0.8),replace = FALSE)

#前 80%的行索引不放回取出作为样本

TrainData <- AccidentModel[SampleIndex,]

#样本定义为训练集

ValidationData <- AccidentModel[-SampleIndex,]

#后 20%定义为验证集

TargetIndex <- which(colnames(AccidentModel)=='MAX_SEV_IR')

#预测目标为名为 MAX_SEV_IR 的列

Predictors <- TrainData[,-TargetIndex]

#其余为预测因子

1.6 见代码

1.7 一律预测为占比较多的那个值，见代码

1.8 不使用任何因子错误率 4176/(4176+4261)= 0.4949627

朴素贝叶斯分类错误率 (1683+2265)/(1683+2265+1911+2578)= 0.4679388

1.9 见代码 截值 0.55 错误率 47.86%

2

2.1 见代码 k=8

2.2 将变量分段后转化为类别型变量，比如 k-means 算法

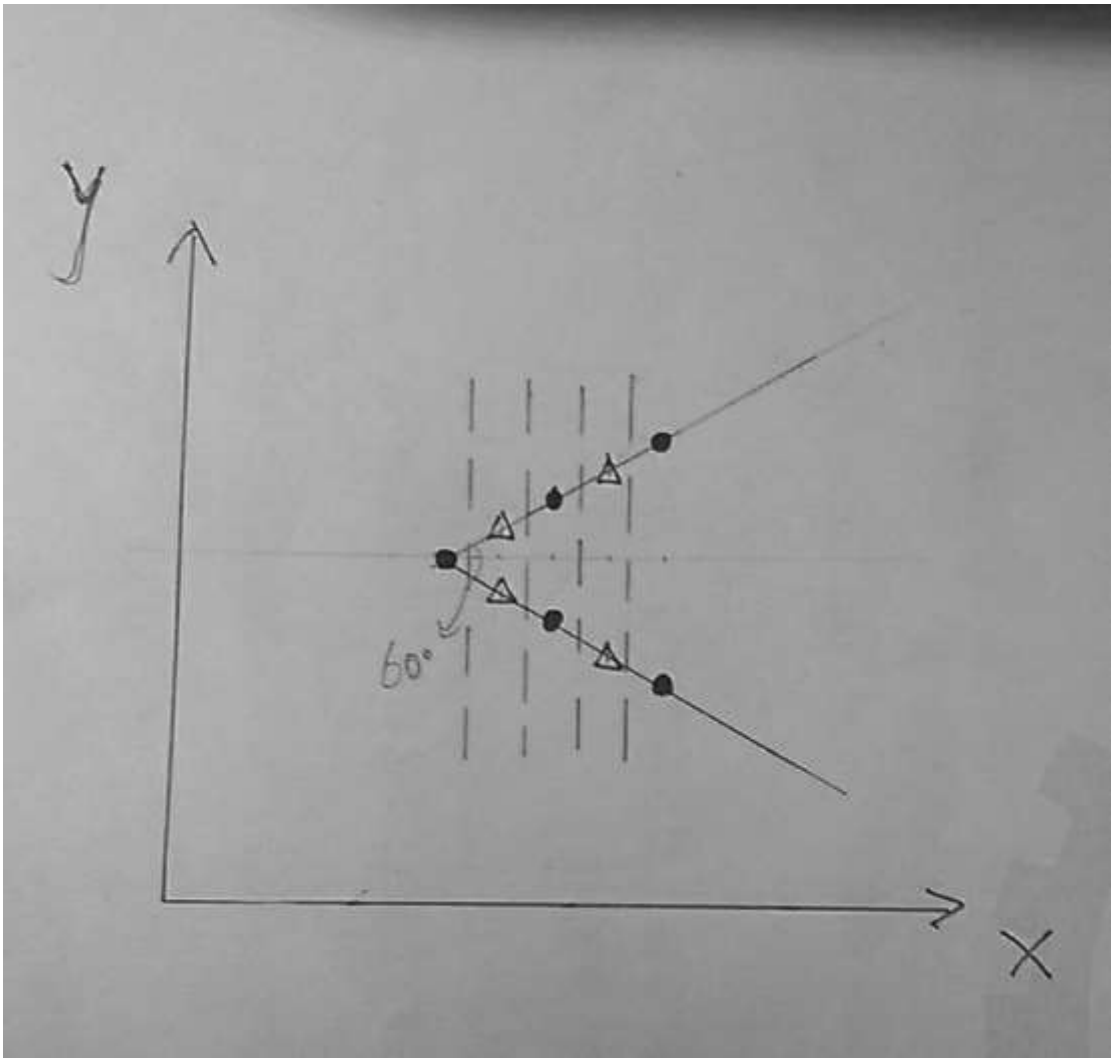
3

3.1 逻辑回归

3.2 如果用于预测的因子间存在多重共线性，模型会失败

3.3 进行主成分分析

4



适合决策树：虚线代表的规则可以完全划分

不适合 KNN:每一个点的周围都是另一种点更多