学号：XXXXXXXXXX

姓名：XXX

切换到工作目录，执行以下脚本，训练六种分类算法的对应模型

#install.packages("caret")  
#install.packages("kernlab")  
#install.packages("ellipse")  
#install.packages("randomForest")  
​  
# 导入必要的库  
library(caret)  
library(kernlab)  
library(randomForest)  
library(ellipse)  
​  
​  
# 读取数据集  
data <- read.csv("homework3\_data.csv")  
​  
data$Weekend <- as.factor(data$Weekend)  
data$Revenue <- as.factor(data$Revenue)  
​  
# 将数据集划分为训练集和测试集  
set.seed(123)  # 设置随机种子以确保结果可复现  
trainIndex <- createDataPartition(data$Revenue, p = 0.7, list = FALSE)  
trainData <- data[trainIndex, ]  
testData <- data[-trainIndex, ]  
​  
# 使用5折交叉验证  
ctrl <- trainControl(method = "cv", number = 5)  
​  
​  
# KNN  
model\_knn <- train(Revenue ~ ., data = trainData, method = "knn", trControl = ctrl)  
knn\_results <- predict(model\_knn, testData)  
​  
# LDA  
model\_lda <- train(Revenue ~ ., data = trainData, method = "lda", trControl = ctrl)  
lda\_results <- predict(model\_lda, testData)  
​  
# GLM  
model\_glm <- train(Revenue ~ ., data = trainData, method = "glm", trControl = ctrl)  
glm\_results <- predict(model\_glm, testData)  
​  
# CART  
model\_tree <- train(Revenue ~ ., data = trainData, method = "rpart", trControl = ctrl)  
tree\_results <- predict(model\_tree, testData)  
​  
# Random Forest  
model\_rf <- train(Revenue ~ ., data = trainData, method = "rf", trControl = ctrl)  
rf\_results <- predict(model\_rf, testData)  
​  
# SVM  
model\_svm <- train(Revenue ~ ., data = trainData, method = "svmRadial", trControl = ctrl)  
svm\_results <- predict(model\_svm, testData)

完成后依次执行以下指令，比较不同模型的性能

confusionMatrix(knn\_results, testData$Revenue)  
confusionMatrix(lda\_results, testData$Revenue)  
confusionMatrix(glm\_results, testData$Revenue)  
confusionMatrix(tree\_results, testData$Revenue)  
confusionMatrix(rf\_results, testData$Revenue)  
confusionMatrix(svm\_results, testData$Revenue)

K 近邻算法（KNN）



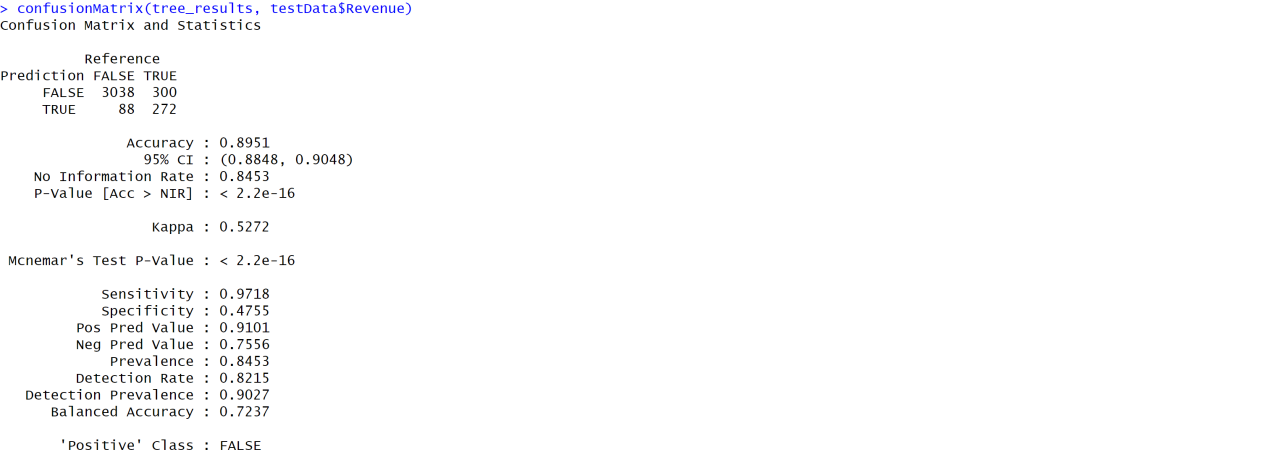
线性判别分析（LDA）



广义线性回归（GLM）



决策树（CART）



随机森林（Random Forest）



支持向量机（SVM）



综合分析使用的六种算法准确率，随机森林算法的性能最佳。