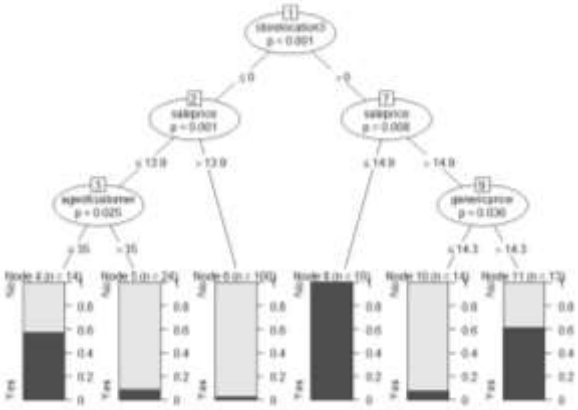
AE10 SOLUTION

The dataset containing store sales (units) of a brand's flagship six blade razor was analyzed using decision trees where stores that sold more than 1000 units were classified as high unit sale stores. The base model, shows that the small stores and high prices with generic price greater than $14.30 are indicative of a high sale possibility (>1000 units) for the branded razor. The lower cut off sale price of <$14.90 is the biggest indicator of >1000 purchases. The average age of the shopper and a sale price under $13.90 is also indicative of the possibility for >1000 unit sales.

The Bagg model confirms that sale price is the most important feature followed by age of the customer and store traffic. The Bagg model appears to be slightly more accurate. Random forest models reveal similar results as does Boosting. There is significant likelihood of overfitting in this case with just a small number of cases. More details are provided in the code for comparison of ROC curves and accuracy as well as AUC values.



```
# Install and load the required packages.

pacman::p_load(psych, ggplot2, devtools, caret, rpart, rpart.plot,
RColorBrewer, party, partykit, pROC, e1071, randomForest, gbm)


storedata <- read.csv(file.choose()) #choose the RazorDataset.csv


str(storedata)

storedata$hipster <- as.factor(storedata$hipster)

storedata$storeparking <- as.factor(storedata$storeparking)
```

```
str(storedata)

ggplot(data=storedata, aes(x=numbertotalsales)) +
geom_histogram(binwidth=1, boundary=.5, fill="white", color="black") +
geom_vline(xintercept = 1000, color="red", size=2) +
labs(x = "Sales in Thousands of Dollars")



storedata$highsale=ifelse(storedata$numbertotalsales<=1000,"No","Yes")

storedata$highsale <- as.factor(storedata$highsale)
storedata$x <- NULL
storedata$numbertotalsales <- NULL

str(storedata)

halfsample = sample(dim(storedata)[1], dim(storedata)[1]/2)
storedata.train = storedata[halfsample, ]
storedata.test = storedata[halfsample, ]

set.seed(123456)

cvcontrol <- trainControl(method="repeatedcv", number = 10,
allowParallel=TRUE)

train.tree <- train(as.factor(highsale) ~ .,
                    data=storedata.train,
                    method="ctree",
                    trControl=cvcontrol,
                    tuneLength = 10)

 train.tree
```

```
plot(train.tree$finalModel, main="Regression Tree for Product Sales over 1000
Units")

tree.classTrain <- predict(train.tree, type="raw")
head(tree.classTrain)

confusionMatrix(storedata.train$highsale,tree.classTrain)

tree.classTest <- predict(train.tree, newdata = storedata.test, type="raw")
head(tree.classTest)

confusionMatrix(storedata.test$highsale,tree.classTest)

tree.probs=predict(train.tree, newdata=storedata.test, type="prob")
head(tree.probs) #examine the data


#calcualte the ROC curve
rocCurve.tree <- roc(storedata.test$highsale,tree.probs[,"Yes"])


#plot the ROC curve, changing the c value will change the color
plot(rocCurve.tree,col=c(4))

auc(rocCurve.tree)



train.bagg <- train(as.factor(highsale) ~ .,
                    data=storedata.train,
                    method="treebag",
                    trControl=cvcontrol,
                    importance=TRUE)

train.bagg
```

```
plot(varImp(train.bagg)) #understandably, price is a big deal)


bagg.classTrain <-  predict(train.bagg, type="raw")
head(bagg.classTrain)
confusionMatrix(storedata.train$highsale,bagg.classTrain)


bagg.classTest <- predict(train.bagg, newdata = storedata.test, type="raw")
head(bagg.classTest)
confusionMatrix(storedata.test$highsale,bagg.classTest)


# procedure to calculate the yes probabilities first.
bagg.probs=predict(train.bagg, newdata=storedata.test, type="prob")
head(bagg.probs)

#calcualte the ROC curve
rocCurve.bagg <- roc(storedata.test$highsale,bagg.probs[,"Yes"])

#plot the ROC curve, changing the c value will change the color
plot(rocCurve.bagg,col=c(6)) #note how the curve doesn't look like a curve


auc(rocCurve.bagg)


train.rf <- train(as.factor(highsale) ~ .,
                  data=storedata.train,
                  method="rf",
                  trControl=cvcontrol,
                  importance=TRUE)
```

```
train.rf


rf.classTrain <-  predict(train.rf, type="raw")
head(rf.classTrain)

confusionMatrix(storedata.train$highsale,rf.classTrain)

rf.probs=predict(train.rf, newdata=storedata.test, type="prob")
head(rf.probs)

rocCurve.rf <- roc(storedata.test$highsale,rf.probs[,"Yes"])
plot(rocCurve.rf,col=c(1))


auc(rocCurve.rf)



train.cf <- train(highsale ~ .,
                  data=storedata.train,
                  method="cforest",
                  trControl=cvcontrol)
train.cf

cf.classTrain <-  predict(train.cf, type="raw")
head(cf.classTrain)

confusionMatrix(storedata.train$highsale,cf.classTrain)

cf.probs=predict(train.cf, newdata=storedata.test, type="prob")
head(cf.probs)

rocCurve.cf <- roc(storedata.test$highsale,cf.probs[,"Yes"])
```

```
plot(rocCurve.cf,col=c(2))

auc(rocCurve.cf)



train.gbm <- train(as.factor(highsale) ~ .,
                   data=storedata.train,
                   method="gbm",
                   verbose=F,
                   trControl=cvcontrol)
train.gbm

gbm.classTrain <-  predict(train.gbm, type="raw")
head(gbm.classTrain)

confusionMatrix(storedata.train$highsale,gbm.classTrain)

gbm.classTest <-  predict(train.gbm, newdata = storedata.test, type="raw")
head(gbm.classTest)

confusionMatrix(storedata.test$highsale,gbm.classTest)

gbm.probs=predict(train.gbm, newdata=storedata.test, type="prob")
head(gbm.probs)

rocCurve.gbm <- roc(storedata.test$highsale,gbm.probs[,"Yes"])
plot(rocCurve.gbm, col=c(3))

auc(rocCurve.gbm)



plot(rocCurve.tree,col=c(4)) # this is our base model
```

```
plot(rocCurve.bagg,add=TRUE,col=c(6)) # color magenta is bagg
plot(rocCurve.rf,add=TRUE,col=c(1)) # color black is rf
plot(rocCurve.cf,add=TRUE,col=c(2)) # color red is cforest
plot(rocCurve.gbm,add=TRUE,col=c(3)) # color green is gbm


# if you wish to compare all AUC's in one shot

auc(rocCurve.tree) # this is our base model AUC
auc(rocCurve.bagg) # this is our bagged model AUC
auc(rocCurve.rf)   # this is our random forest AUC
auc(rocCurve.cf)   # this is our conditional inference tree AUC
auc(rocCurve.gbm)  # this is gradient boosted model AUC
```