Short HW #3

Bennett Hellman
15.072 - Advanced Analytics Edge
MIT

October 18, 2021

1. (a) i. Cost of (TN, FP, FN, TP) = (0, 1200, 35000, 27450)
Confusion Matrix:

	Predicted 0	Predicted 1
Actual 0	0	1200
Actual 1	7550	0

Figure 1: Tree 1 (cp = default)

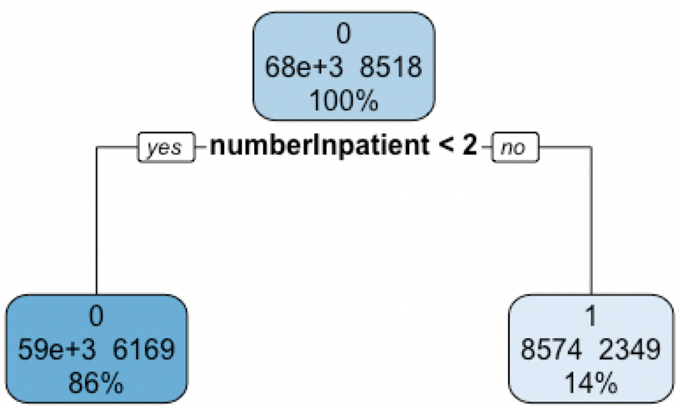


Figure 2: Tree 2 (cp = 0.0016)

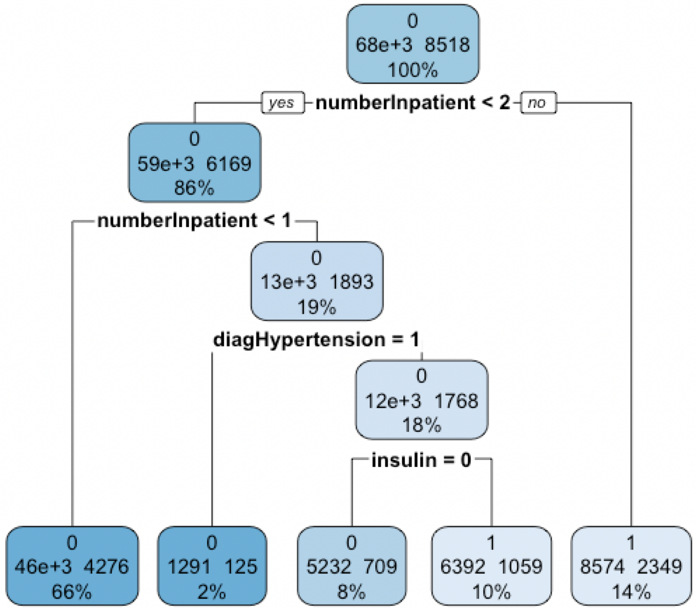
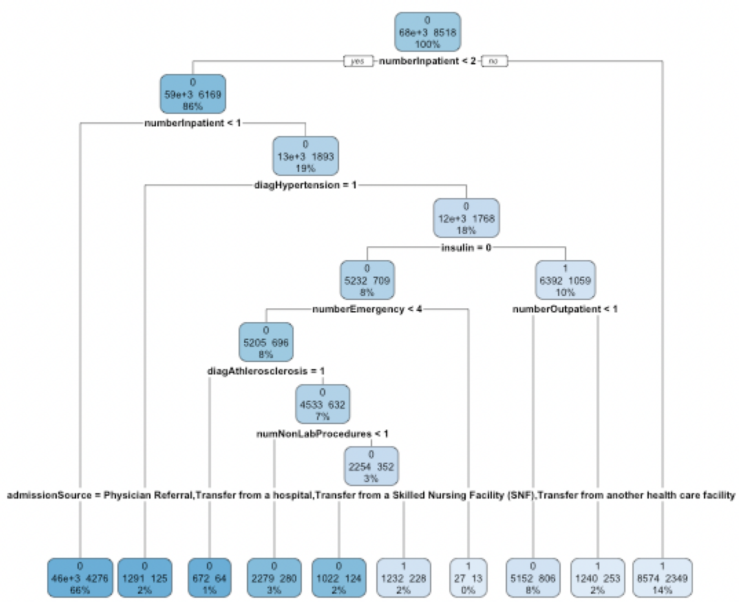


Figure 3: Tree 3 (cp = 0.0012)



(b) i.

ii. The main driver of those receiving telehealth intervention are those with a large number of impatient and outpatient days. This is intuitive because they are likely ones to be more seriously injured or have preexisting conditions that require more medical attention.

(c) i. Number of patients subjected to telehealth intervention in test set:
$(Tree_1, Tree_2, Tree_3)$ = (3692, 6106, 4604)

ii. Expected number of prevented re-admissions:
$(Tree_1, Tree_2, Tree_3)$ = (584.25, 844.5, 678.75)

iii. Accuracy on test set:
$(Tree_1, Tree_2, Tree_3)$ = (80%, 74%, 78%)

iv. True positive rate:
$(Tree_1, Tree_2, Tree_3)$ = (26%, 40%, 32%)

v. False positive rate:
$(Tree_1, Tree_2, Tree_3)$ = (13%, 22%, 16%)

vi. Total Cost:
$(Tree_1, Tree_2, Tree_3)$ = ($96,979,150, $96,839,700, $96,971,050)

(d) i. Absolute Cost Savings:
$(Tree_1, Tree_2, Tree_3)$ = ($2,385,850, $2,525,300, $2,393,950)
Relative Cost Savings:
$(Tree_1, Tree_2, Tree_3)$ = (2.40%, 2.54%, 2.41%)

(e) i. I artificially increased the associated value for false positive treatment in the lost function. I "guess and checked" this value until I was below their desired number of patients treated.

ii. The only parameter I changed was the loss matrix, which I changed from (0,1200,7550,0) to (0,1200×1.75,7550,0).

iii. Patients receiving telehealth intervention = 689
Expected # of re-admissions = 54.25
Value of expected re-admission = $1,833,650

iv. Absolute and relative costs vs. current practice: $1,071,950, 1.08%
Absolute and relative costs vs. model with no "budget" constraint: -$1,453,350, -1.5%

v. This 5% telehealth prescription constraint, in an effort to minimize cost, actually leads to a worse strain on the budget. Hospital management should obviously select the model that minimizes cost in the end, rather than restrict the number of interventions.

```
library(caTools)
library(tidyverse)
library(miscTools)
library(Metrics)
library(plotly)
library(glmnet)
library(PRROC)
library(ROCit)
library(rpart)
library(rpart.plot)
library(randomForest)
library(caret)
library(caTools)
library(dplyr)

data = read.csv("/Users/bennetthellman/Desktop/OneDrive - Massachusetts
Institute of Technology/AE/HWs/HW3/readmission.csv")
data<-data%>%mutate(readmission = as.factor(readmission))
set.seed(144)
split = createDataPartition(data$readmission, p = 0.75, list = FALSE)
readm.train <- data[split,]
readm.test <- data[-split,]

#a
#this is actually the lost matrix, I will redefine the cost matrix later
cm <- matrix(c(0,1200,7550,0), nrow=2, ncol=2, byrow=TRUE)

#b
tree1 = rpart(readmission ~ .,
                    data=readm.train, parms=list(loss=cm), method =
'class')
tree2 = rpart(readmission ~ .,
             data=readm.train, parms=list(loss=cm),
             cp=0.0016)
tree3 = rpart(readmission ~ .,
             data=readm.train, parms=list(loss=cm),
             cp=0.0012)
rpart.plot(tree1, extra = 101)
rpart.plot(tree2, extra =101)
rpart.plot(tree3, extra =101)

#c
pred_t1 = predict(tree1, newdata = readm.test, type = 'class')
pred_t2 = predict(tree2, newdata = readm.test, type = 'class')
pred_t3 = predict(tree3, newdata = readm.test, type = 'class')

#ci
table_mat1 <- table(readm.test$readmission, pred_t1)
table_mat1[1,2]+table_mat1[2,2]
table_mat2 <- table(readm.test$readmission, pred_t2)
table_mat2[1,2]+table_mat2[2,2]
table_mat3 <- table(readm.test$readmission, pred_t3)
table_mat3[1,2]+table_mat3[2,2]
```

```
#cii
#simply .75 * bottom true positives
.75*table_mat1[2,2]
.75*table_mat2[2,2]
.75*table_mat3[2,2]
#ciii
accuracy_Test1 <- sum(diag(table_mat1)) / sum(table_mat1)
accuracy_Test1
accuracy_Test2 <- sum(diag(table_mat2)) / sum(table_mat2)
accuracy_Test2
accuracy_Test3 <- sum(diag(table_mat3)) / sum(table_mat3)
accuracy_Test3

#civ
tpr1 = table_mat1[2,2]/(table_mat1[2,1]+table_mat1[2,2])
tpr1
tpr2 = table_mat2[2,2]/(table_mat2[2,1]+table_mat2[2,2])
tpr2
tpr3 = table_mat3[2,2]/(table_mat3[2,1]+table_mat3[2,2])
tpr3

#cv
fpr1 = table_mat1[1,2]/(table_mat1[1,1]+table_mat1[1,2])
fpr1
fpr2 = table_mat2[1,2]/(table_mat2[1,1]+table_mat2[1,2])
fpr2
fpr3 = table_mat3[1,2]/(table_mat3[1,1]+table_mat3[1,2])
fpr3

#cvi
cm <- matrix(c(0,1200,35000,27450), nrow=2, ncol=2, byrow=TRUE)
sum(cm*table_mat1)
sum(cm*table_mat2)
sum(cm*table_mat3)

#di
bl_cost = sum(readm.test$readmission==1)*35000 #absolute cost reduction
bl_cost-sum(cm*table_mat1) #relative cost reduction
(bl_cost-sum(cm*table_mat1))/bl_cost*100
bl_cost-sum(cm*table_mat2)
(bl_cost-sum(cm*table_mat2))/bl_cost*100
bl_cost-sum(cm*table_mat3)
(bl_cost-sum(cm*table_mat3))/bl_cost*100

#eii
cm <- matrix(c(0,1200,7550,0), nrow=2, ncol=2, byrow=TRUE)

bud <- matrix(c(0,1200*1.25,7550,0), nrow=2, ncol=2, byrow=TRUE)
treebud = rpart(readmission ~ ., data=readm.train,
parms=list(loss=bud),cp=0.001)
pred_bud = predict(treebud, newdata = readm.test, type = 'class')
table_bud <- table(readm.test$readmission, pred_bud)
((table_bud[1,2]+table_bud[2,2])/sum(table_bud))
```

```
bud <- matrix(c(0,1200*1.5,7550,0), nrow=2, ncol=2, byrow=TRUE)
treebud = rpart(readmission ~ ., data=readm.train,
parms=list(loss=bud),cp=0.001)
pred_bud = predict(treebud, newdata = readm.test, type = 'class')
table_bud <- table(readm.test$readmission, pred_bud)
((table_bud[1,2]+table_bud[2,2])/sum(table_bud))

bud <- matrix(c(0,1200*1.75,7550,0), nrow=2, ncol=2, byrow=TRUE)
treebud = rpart(readmission ~ ., data=readm.train,
parms=list(loss=bud),cp=0.001)
pred_bud = predict(treebud, newdata = readm.test, type = 'class')
table_bud <- table(readm.test$readmission, pred_bud)
((table_bud[1,2]+table_bud[2,2])/sum(table_bud))

bud <- matrix(c(0,1200*1.6,7550,0), nrow=2, ncol=2, byrow=TRUE)
treebud = rpart(readmission ~ ., data=readm.train,
parms=list(loss=bud),cp=0.001)
pred_bud = predict(treebud, newdata = readm.test, type = 'class')
table_bud <- table(readm.test$readmission, pred_bud)
((table_bud[1,2]+table_bud[2,2])/sum(table_bud))

bud <- matrix(c(0,1200*1.5,7550,0), nrow=2, ncol=2, byrow=TRUE)
treebud = rpart(readmission ~ ., data=readm.train,
parms=list(loss=bud),cp=0.001)
pred_bud = predict(treebud, newdata = readm.test, type = 'class')
table_bud <- table(readm.test$readmission, pred_bud)
((table_bud[1,2]+table_bud[2,2])/sum(table_bud))

bud <- matrix(c(0,1200*1.75,7550,0), nrow=2, ncol=2, byrow=TRUE)
treebud = rpart(readmission ~ ., data=readm.train,
parms=list(loss=bud),cp=0.001)
pred_bud = predict(treebud, newdata = readm.test, type = 'class')
table_bud <- table(readm.test$readmission, pred_bud)
((table_bud[1,2]+table_bud[2,2])/sum(table_bud))

#eiii
(table_bud[1,2]+table_bud[2,2])
.25*table_bud[2,2]
.25*table_bud[2,2]*(35000-1200)

#eiv
bl_cost-sum(cm*table_bud)
(bl_cost-sum(cm*table_bud))/bl_cost*100
sum(cm*table_mat2)-sum(cm*table_bud)
(sum(cm*table_mat2)-sum(cm*table_bud))/sum(cm*table_mat2)*100
```