Classification Trees: Preventing Hospital Readmissions

Suzana Iacob

13/10/2019

Data Description

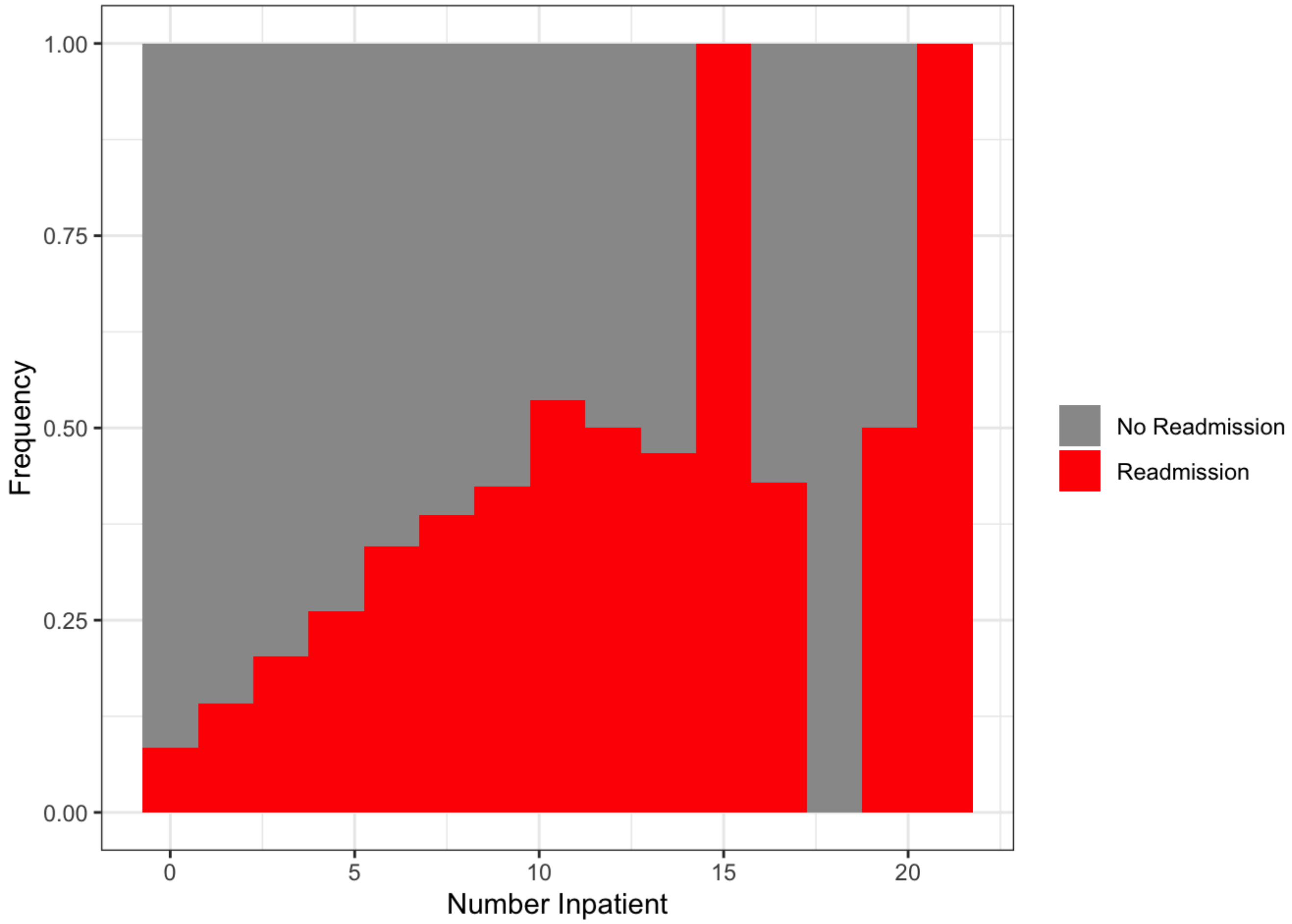
This project uses patient information to predict whether a patient will be re-admitted to the hospital within 30 days.

It contains the following variables: • readmission: 1 if the patient had a readmission within 30 days of discharge, 0 otherwise. • Patient features: race, gender, and age • Medical system use: numberOutpatient, numberEmergency, and numberInpatient count the number of times the patient used medical services in the past year. • Diabetic treatments: acarbose, chlorpropamide, glimepiride, glipizide, glyburide, glyburide.metformin, insulin, metformin, nateglinide, pioglitazone, repaglinide, and rosiglitazone. • Admission information: admissionType, admissionSource, numberDiagnoses, and several other variables representing the diagnosed conditions. • Treatment information: timeInHospital, numLabProcedures, numNonLabProcedures, numMedications.

```
library(caret)
library(rpart)
library(rpart.plot)
library(caTools)
library(dplyr)
readmission = read.csv("readmission.csv")
```

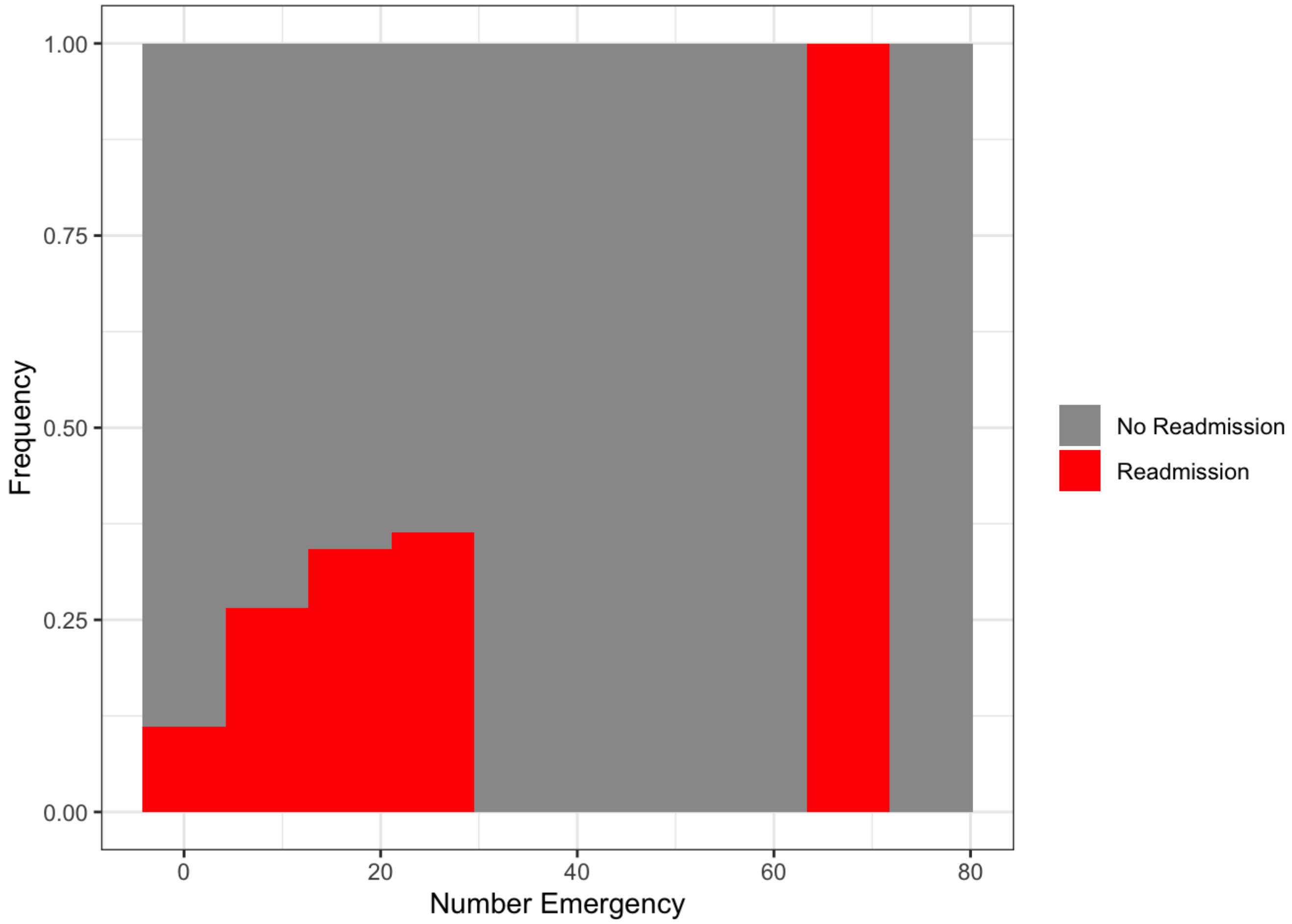
The dataset contains information regarding patient characteristics. We expect recent medical system use to be an important predictor of readmissions. Inpatient refers to hospital admission, whereas Outpatient means a person was treated and discharged immediately. Number Emergency refers to ER visits. Since readmission rate refers to hopital admissions, we expect numberInpatient to be significant, and it appears to be, according to the plot.

```
library(ggplot2)
ggplot(data=readmission) +
  geom_histogram(aes(x=numberInpatient,fill=(readmission==1)),position='fill', bins =
15) +
  theme_bw() +
  xlab('Number Inpatient') +
  ylab('Frequency') +
  scale_fill_manual (name='', labels=c('No Readmission','Readmission'), values=c('gre
y60','red'))
```



Number Emergency could also be a factor since an ER patient may suffer from a more critical condition and may be at a higher readmission rate.

```
library(ggplot2)
ggplot(data=readmission) +
  geom_histogram(aes(x=numberEmergency,fill=(readmission==1)),position='fill', bins =
10) +
  theme_bw() +
  xlab('Number Emergency') +
  ylab('Frequency') +
  scale_fill_manual (name='', labels=c('No Readmission','Readmission'), values=c('gre
y60','red'))
```



We would also expect age to be correlated with higher readmission rates.

We also have a high number of treatments and admission diagnostics. It is difficult to examine pairwise relationships, especially since many of these variables are correlated to one another. We would also need medical expertise to assess exactly which veriables may be relevant and which not in case of multicollinearity. Hence we are motivated to use predictive analytics to model these relationships. The model will be able to distinguish the important variables.

Cost of readmission

Let's assume the cost of a 30-day unplanned readmission is $35,000. We want to explore the alternative of offering **telehealth interivientions** which will hopefully reduce readmission. Say the cost of telehealth is $1,200 per intervention.

We investigate the costs of true negatives, false positives, false negatives, and true positives and define a loss matrix for a CART model based on this.

```
set.seed(144)
split = createDataPartition(readmission$readmission, p = 0.75, list = FALSE)
readm.train <- readmission[split,]
readm.test <- readmission[-split,]
```

The cost of True Negatives is $0.

The cost of True Positives is $1,200 + $35,000* 0.75 = $27,450 (telehealth cost plus the expected value of readmission cost reduced by 25%).

The cost of False Positives is $1,200 (telehealth cost).

The cost of False Negatives is $35,000 (readmission cost).

We would like 0 on the diagonal of the loss matrix, since we incur the cost of the patients that we correctly predict but get readmitted anyway. So we subtract $27,450 from both the True Positive cost and the False Negative cost. Hence the True Positives will have 0 in the loss matrix and the False Negatives will have

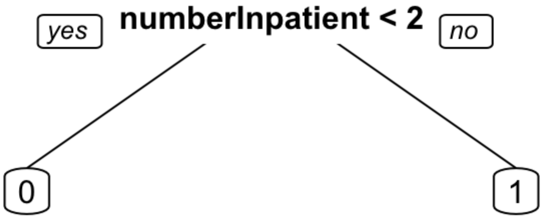
$35,000 - $27,450 = $7,550.

```
PenaltyMatrix = matrix(c(0,1200,7550,0), byrow=TRUE, nrow=2)
```

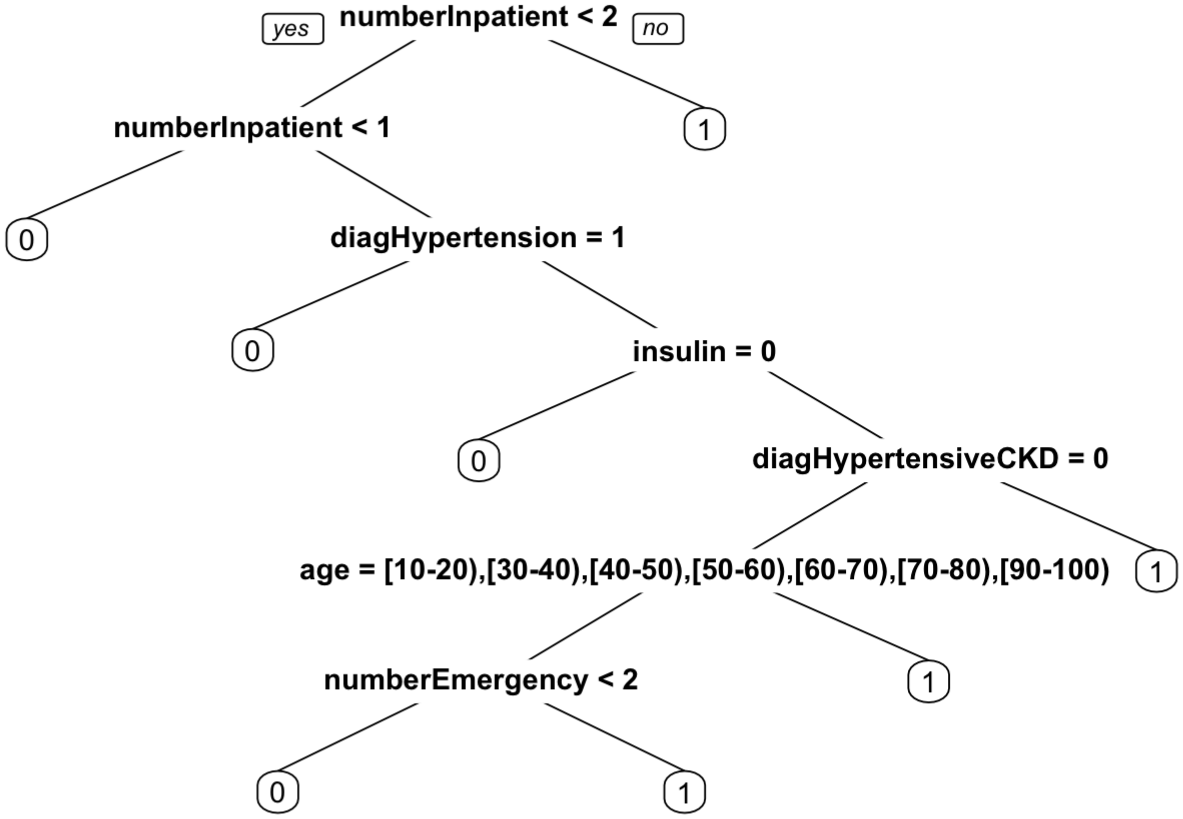
Classification Tree

```
readmissionsTree = rpart(readmission ~ ., data=readm.train,cp=0.01, minbucket=50, met
hod="class", parms=list(loss=PenaltyMatrix))
readmissionsTree2 = rpart(readmission ~ .,data=readm.train, cp=0.0018,minbucket=50, m
ethod="class", parms=list(loss=PenaltyMatrix))
readmissionsTree3 = rpart(readmission ~ .,data=readm.train, cp=0.001, minbucket=50, m
ethod="class", parms=list(loss=PenaltyMatrix))
```

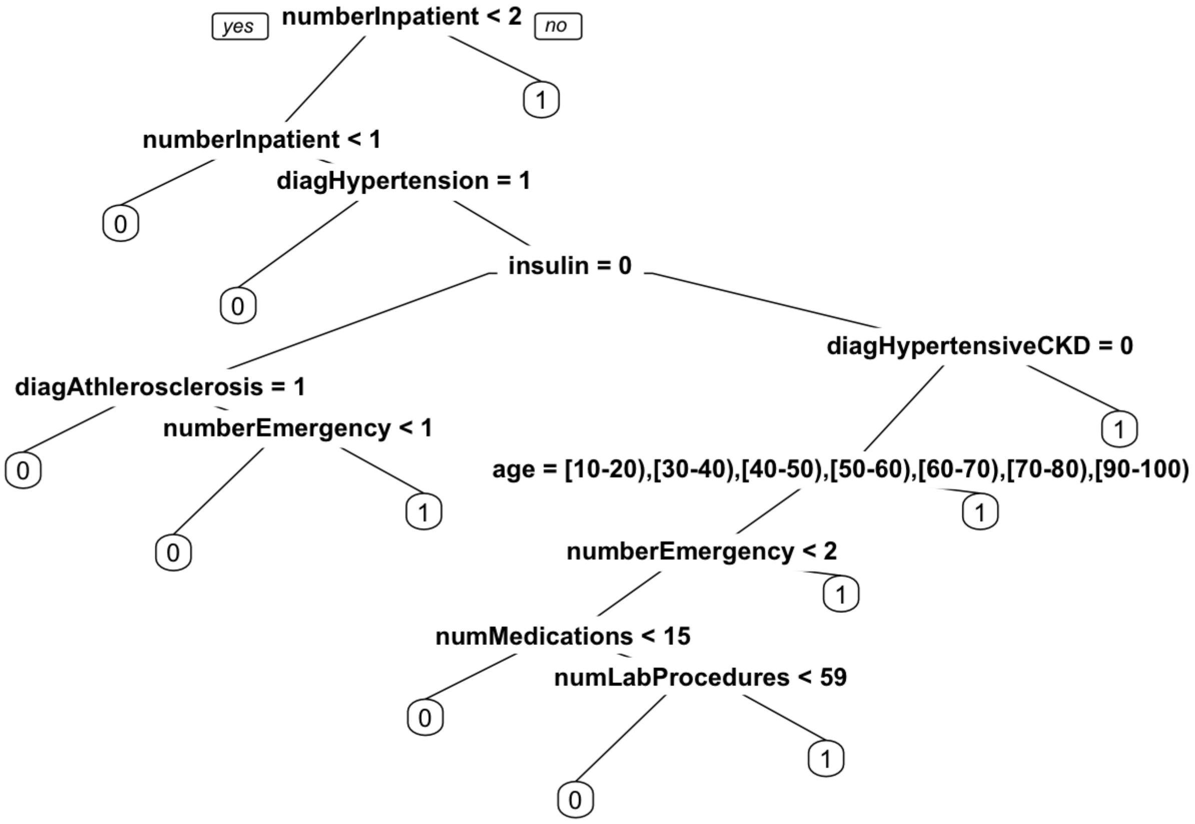
```
prp(readmissionsTree, digits = 0, varlen = 0, faclen = 0)
```



```
prp(readmissionsTree2, digits = 0, varlen = 0, faclen = 0)
```



```
prp(readmissionsTree3, digits = 0, varlen = 0, faclen = 0)
```



Interpretation

Important variables: numberInpatient, diagHypertention, insulin, diagHypertensiveCKD and diagAthlerosclerosis (diagnostics the patient received), numberEmergency, age, numMedications, numberOutpatient, and numLabProcedures.

The types of patients we predict will be readmitted are patients who recently used the medical system (numberInpatient>2), have been also diagnosed with various complications such as Hypertention, Athlerosclerosis, and have other forms of treatment (such as requiring insulin). In general, the longer the stay in the hospital, the larger the number of diagnostics and other complications, the higher the chance we predict readmission. This matches intuition and our analysis in part a.

Making prediction

```
PredictTest = predict(readmissionsTree, newdata = readm.test, type="class")
PredictTest2 = predict(readmissionsTree2, newdata = readm.test, type="class")
PredictTest3 = predict(readmissionsTree3, newdata = readm.test, type="class")
```

```
matrix1 = table(readm.test$readmission, PredictTest)
print(matrix1)
```

```
##    PredictTest
##         0     1
##   0 19752  2948
##   1  1994   747
```

```
matrix2 = table(readm.test$readmission, PredictTest2)
print(matrix2)
```

```
##    PredictTest2
##         0     1
##   0 19097  3603
##   1  1881   860
```

```
matrix3 = table(readm.test$readmission, PredictTest3)
print(matrix3)
```

```
##    PredictTest3
##         0     1
##   0 18644  4056
##   1  1796   945
```

No. of patients subject to telehealth intervention:

```
# First tree
print(matrix1[1,2]+matrix1[2,2])
```

```
## [1] 3695
```

```
# Second tree
print(matrix2[1,2]+matrix2[2,2])
```

```
## [1] 4463
```

```
# Third tree
print(matrix3[1,2]+matrix3[2,2])
```

```
## [1] 5001
```

Expected no. of readmissions prevented:

```
# First tree
print(matrix1[2,2]*0.25)
```

```
## [1] 186.75
```

```
# Second tree
print(matrix2[2,2]*0.25)
```

```
## [1] 215
```

```
# Third tree
print(matrix3[2,2]*0.25)
```

```
## [1] 236.25
```

Prediction accuracy

```
# First tree
print((matrix1[1,1]+matrix1[2,2])/nrow(readm.test))
```

```
## [1] 0.8057466
```

```
# Second tree
print((matrix2[1,1]+matrix2[2,2])/nrow(readm.test))
```

```
## [1] 0.7844424
```

```
# Third tree
print((matrix3[1,1]+matrix3[2,2])/nrow(readm.test))
```

```
## [1] 0.7699776
```

True positive rate:

```
# First tree
print(matrix1[2,2]/(matrix1[2,1]+matrix1[2,2]))
```

```
## [1] 0.2725283
```

```
# Second tree
print(matrix2[2,2]/(matrix2[2,1]+matrix2[2,2]))
```

```
## [1] 0.3137541
```

```
# Third tree
print(matrix3[2,2]/(matrix3[2,1]+matrix3[2,2]))
```

```
## [1] 0.3447647
```

False positive rate:

```
# First tree
print(matrix1[1,2]/(matrix1[1,1]+matrix1[1,2]))
```

```
## [1] 0.1298678
```

```
# Second tree
print(matrix2[1,2]/(matrix2[1,1]+matrix2[1,2]))
```

```
## [1] 0.1587225
```

```
# Third tree
print(matrix3[1,2]/(matrix3[1,1]+matrix3[1,2]))
```

```
## [1] 0.1786784
```

Total cost:

```
# First tree
print(matrix1[1,2] * 1200 + matrix1[2,1] * 35000 + matrix1[2,2] * (1200 + 35000* 0.75
) )
```

```
## [1] 93832750
```

```
# Second tree
print(matrix2[1,2] * 1200 + matrix2[2,1] * 35000 + matrix2[2,2] * (1200 + 35000* 0.75
) )
```

```
## [1] 93765600
```

```
# Third tree
print(matrix3[1,2] * 1200 + matrix3[2,1] * 35000 + matrix3[2,2] * (1200 + 35000* 0.75
) )
```

```
## [1] 93667450
```

In current practice the expected cost is:

```
baselineCost = sum(readm.test$readmission==1)*35000
baselineCost - matrix3[1,2] * 1200 + matrix3[2,1] * 35000 + matrix3[2,2] * (1200 + 35
000* 0.75)
```

```
## [1] 179868050
```

We consider the final tree.

Currently we use no telehealth and incur a total cost of $95,935,000 from the hospital readmissions of 2,741 patients (in the test set).

The model predicts that 5,001 patients are at high risk of readmission and we suggest that telehealth is used. The False Positive rate is 18% and so we incorrectly prescribe telehealth to 4,056 patients who do not require it as they would not have been readmitted. We also miss 1,796 who would have needed telehealth but we did not prescribe it to them. We correctly identify 945 patients who were going to get readmitted and we expect their readmission rate to drop by 25%.

We clearly see that using a predictive model and telehealth results in a lower cost since we decrease the readmission rate by $2,267,550. This is a **significant cost reduction** and improves patients health and quality of life.

Sensitivity analysis

Since the numbers we used for the cost are estimates, we run a sensitivity analysis where we vary the rate of telehealth efficiency (i.e. the rate of reduction in readmissions), and re-examine the final cost:

```
rate = 0.25
seq = seq(0.05, 0.30, 0.01)

for (i in seq(0.08, 0.28, 0.01)){
  rate = i
  fnCost = 35000 - (1200 + 35000* (1-rate))
  PenaltyMatrixRange = matrix(c(0,1200,fnCost,0), byrow=TRUE, nrow=2)

  readmissionsTreeRange = rpart(readmission ~ ., data=readm.train,cp=0.01, minbucket=
50, method="class", parms=list(loss=PenaltyMatrixRange))
  PredictTestRange = predict(readmissionsTreeRange, newdata = readm.test, type="clas
s")
  matrixRange = table(readm.test$readmission, PredictTestRange)
  print(matrixRange[1,2] * 1200 + matrixRange[2,1] * 35000 + matrixRange[2,2] * (1200
+ 35000* (1-rate)))
}
```

```
## [1] 95935000
## [1] 95935000
## [1] 95935000
## [1] 95935000
## [1] 95821600
## [1] 95759650
## [1] 95697700
## [1] 95635750
## [1] 95573800
## [1] 95511850
## [1] 95662900
## [1] 95401450
## [1] 95140000
## [1] 94878550
## [1] 94617100
## [1] 94355650
## [1] 94094200
## [1] 93832750
## [1] 93571300
## [1] 93309850
## [1] 92844600
```

We note that for very small values of the rate (corresponding to telehealth efficiency) the cost is equal to the cost without telehealth, hence it is not profitable. We note that from the value 0.13 onwards the telehealth becomes profitable. Naturally as the rate increases beyond 25% we get an even larger cost reduction.

We currentley treat the following percentage of patients (on the test set).

```
print((matrix3[1,2]+matrix3[2,2])/nrow(readm.test))
```

```
## [1] 0.1965725
```

Suppose we can only offer telehealth to **5% of patients**. We can modify the loss matrix to decrease the ratio of FN loss to FP loss. Since we are overpenalizing false negatives, we can adjust this ratio so that we treat fewer patients with telehealth. The smaller the FN penalty compared to the FP penalty, the smaller the number of patients we treat. We want to penalize false positives more, so that we only treat the patients with a very high risk (the “top” 5%).

```
  PenaltyMatrix4 = matrix(c(0,2000,6750,0), byrow=TRUE, nrow=2)

  readmissionsTree4 = rpart(readmission ~ ., data=readm.train,cp=0.001, minbucket=50,
method="class", parms=list(loss=PenaltyMatrix4))
  PredictTrain4= predict(readmissionsTree4, type="class")
  matrix4 = table(readm.train$readmission, PredictTrain4)
 print((matrix4[1,2]+matrix4[2,2])/nrow(readm.train))
```

```
## [1] 0.041415
```

```
  PenaltyMatrix4 = matrix(c(0,1500,7250,0), byrow=TRUE, nrow=2)

  readmissionsTree4 = rpart(readmission ~ ., data=readm.train,cp=0.001, minbucket=50,
method="class", parms=list(loss=PenaltyMatrix4))
  PredictTrain4= predict(readmissionsTree4, type="class")
  matrix4 = table(readm.train$readmission, PredictTrain4)
 print((matrix4[1,2]+matrix4[2,2])/nrow(readm.train))
```

```
## [1] 0.1430724
```

```
  PenaltyMatrix4 = matrix(c(0,1900,6850,0), byrow=TRUE, nrow=2)

  readmissionsTree4 = rpart(readmission ~ ., data=readm.train,cp=0.001, minbucket=50,
method="class", parms=list(loss=PenaltyMatrix4))
  PredictTrain4= predict(readmissionsTree4, type="class")
  matrix4 = table(readm.train$readmission, PredictTrain4)
 print((matrix4[1,2]+matrix4[2,2])/nrow(readm.train))
```

```
## [1] 0.05076973
```

We use the following penalty matrix with the cost of FP of $1,900 and cost of FN of $6,850 which brings us to 5.1% of patients.

```
  FN = i
  PenaltyMatrixFinal = matrix(c(0,1900,6850,0), byrow=TRUE, nrow=2)

  readmissionsTreeFinal = rpart(readmission ~ ., data=readm.train,cp=0.001, minbucket
=50, method="class", parms=list(loss=PenaltyMatrixFinal))
  PredictTrainFinal= predict(readmissionsTreeFinal, type="class")
  matrixFinal = table(readm.train$readmission, PredictTrainFinal)
 print((matrixFinal[1,2]+matrixFinal[2,2])/nrow(readm.train))
```

```
## [1] 0.05076973
```

Evaluating models

```
 PredictTestFinal= predict(readmissionsTreeFinal, newdata = readm.test, type="class")
 matrixFinalTest = table(readm.test$readmission, PredictTestFinal)
 print(matrixFinalTest)
```

```
##    PredictTestFinal
##         0     1
##   0 21702   998
##   1  2388   353
```

Number of patients who receive telehealth interventions:

```
 print((matrixFinalTest[1,2]+matrixFinalTest[2,2]))
```

```
## [1] 1351
```

Expected no. of readmissions prevented:

```
print(matrixFinalTest[2,2]*0.25)
```

```
## [1] 88.25
```

The net value is the total cost under no telehealth - the cost with this model.

```
costModel = matrixFinalTest[1,2] * 1200 + matrixFinalTest[2,1] * 35000 + matrixFinalT
est[2,2] * (1200 + 35000* (1-rate))
baselineCost = sum(readm.test$readmission==1)*35000

print(baselineCost - costModel)
```

```
## [1] 1838200
```

Compared to the model under no buget constraint the value is:

```
costNoConstraint = matrix3[1,2] * 1200 + matrix3[2,1] * 35000 + matrix3[2,2] * (1200
 + 35000* 0.75)

print(costNoConstraint - costModel)
```

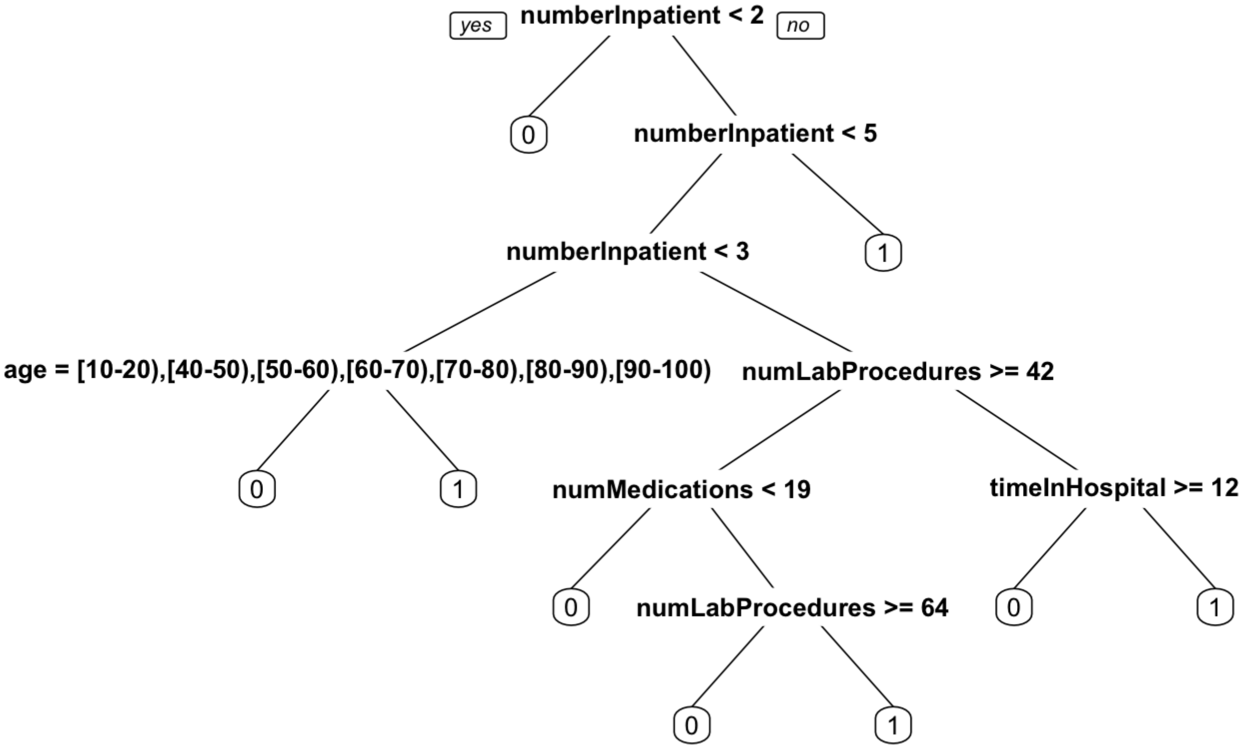
```
## [1] -429350
```

The value is negative since we naturally treat more patients and prevent more readmission under no buget constraint.

Compared to the baseline we have seen that the model brings a net value of $2,085,300.

We built a predictive model to assess which patients have high readmission risk. The current estimated cost of readmissions is $95,935,000 to the hospital.

The model represents a decision tree using only a few variables: numberImpatient, age, numberLabProcedures, numberMedications and timeInHospital. It is highly interpretable, the doctors can easily follow the splits in the tree to assess whether a new patient will be readmitted within 30 days (E.g. If the patient was admitted for more than 5 days, we predict they will get readmitted within 30 days).



We recommend using a predictive model to select the high risk patients and prescribe telehealth, as it brings clear financial benefits of $2,085,300, under a budget constraint of treating only 5% of patients. If there was no budger constraint, we could save up to $2,267,550.

Naturally, a prescriptive model should not replace medical expertise, and doctors should assess patients when prescribing telehealth. Perhaps some patients have known history and could benefit from the extra care, despite a 0 prediction from the model.