HOMEWORK 1 SİNAN DEMİRHAN 2016402330

**1. Consider the dataset called “heart.csv”. There are 14 attributes that are described below, where the last one is the output attribute.**

**age: age in years**

**sex: (1 = male; 0 = female)**

**cp: chest pain type**

**trestbps: testing blood pressure (in mm Hg on admission to the hospital)**

**chol: serum cholestoral in mg/dl**

**fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)**

**restecg: resting electrocardiographic results**

**thalach: maximum heart rate achieved**

**exang: exercise induced angina (1 = yes; 0 = no)**

**oldpeak: ST depression induced by exercise relative to rest**

**slope: the slope of the peak exercise ST segment**

**ca: number of major vessels (0-3) colored by flourosopy**

**thal: 3 = normal; 6 = fixed defect; 7 = reversable defect**

**target: 1 or 0**

**library("caTools")**

**getwd()**

**setwd("C:/Users/Sinan/Desktop/dersler/ie 425/HW1")**

**heartdata<- read.csv(file = "heart.csv",header=TRUE)**

**library(tree)**

**require(caTools)**

**a. Partition the dataset heart.csv into training and test sets with 70% going into the training set by using CaTools package and a seed value of 425.**

**data1= sample.split(heartdata,SplitRatio = 0.70)**

**train\_heart=subset(heartdata,data1==TRUE)**

**test\_heart=subset(heartdata,data1==FALSE)**

**b. By playing with the parameter values of the “tree” function of the “tree” package, find the best tree in terms of the performance on the test set. Give the number of leaf nodes of the best tree and provide the summary of the tree using the summary(tree\_name) function with its corresponding performance (accuracy, error rate) on the training as well as test set.**

**acc<-0**

**c<-0**

**s<-0**

**u<-0**

**for(i in 2:60) {**

**for(j in 1:20){**

**if(j>=(i/2)){break}**

**pbest<-tree(target~.,data=train\_heart,mincut = j, minsize = i)**

**ptest<-predict(pbest,newdata=test\_heart,type="class")**

**t11<-table(test\_heart$target,ptest)**

**acl<-100\*(1-sum(diag(t11))/sum(t11))**

**if(acl>acc){**

**acc<-acl**

**c<-j**

**s<-i**

**}**

**}**

**}**

**bestmincut<-c #1**

**bestminsize<-s #3**

**#mindev is default =0.001**

**I did not add it to the for loop because of the reason that the time is really long.**

**pstree<-tree(target~. , data = train\_heart,mincut=bestmincut,minsize=bestminsize,mindev=bestmindev)**

**pstree**

1) root 195 268.500 1 ( 0.45128 0.54872 )

2) cp < 0.5 89 99.540 0 ( 0.75281 0.24719 )

4) ca < 0.5 37 51.270 1 ( 0.48649 0.51351 )

8) thalach < 146 17 15.840 0 ( 0.82353 0.17647 )

16) exang < 0.5 7 9.561 0 ( 0.57143 0.42857 )

32) thal < 2.5 4 4.499 1 ( 0.25000 0.75000 )

64) ï..age < 66.5 3 0.000 1 ( 0.00000 1.00000 ) \*

65) ï..age > 66.5 1 0.000 0 ( 1.00000 0.00000 ) \*

33) thal > 2.5 3 0.000 0 ( 1.00000 0.00000 ) \*

17) exang > 0.5 10 0.000 0 ( 1.00000 0.00000 ) \*

9) thalach > 146 20 20.020 1 ( 0.20000 0.80000 )

18) chol < 177.5 1 0.000 0 ( 1.00000 0.00000 ) \*

19) chol > 177.5 19 16.570 1 ( 0.15789 0.84211 )

38) chol < 243.5 11 0.000 1 ( 0.00000 1.00000 ) \*

39) chol > 243.5 8 10.590 1 ( 0.37500 0.62500 ) \*

5) ca > 0.5 52 22.940 0 ( 0.94231 0.05769 )

10) restecg < 0.5 30 0.000 0 ( 1.00000 0.00000 ) \*

11) restecg > 0.5 22 17.530 0 ( 0.86364 0.13636 )

22) chol < 231.5 11 0.000 0 ( 1.00000 0.00000 ) \*

23) chol > 231.5 11 12.890 0 ( 0.72727 0.27273 )

46) ï..age < 63 9 6.279 0 ( 0.88889 0.11111 )

92) chol < 233.5 1 0.000 1 ( 0.00000 1.00000 ) \*

93) chol > 233.5 8 0.000 0 ( 1.00000 0.00000 ) \*

47) ï..age > 63 2 0.000 1 ( 0.00000 1.00000 ) \*

3) cp > 0.5 106 105.500 1 ( 0.19811 0.80189 )

6) ï..age < 55.5 64 35.090 1 ( 0.07812 0.92188 )

12) oldpeak < 3 62 24.020 1 ( 0.04839 0.95161 )

24) trestbps < 182 61 17.600 1 ( 0.03279 0.96721 )

48) chol < 232.5 34 0.000 1 ( 0.00000 1.00000 ) \*

49) chol > 232.5 27 14.260 1 ( 0.07407 0.92593 )

98) chol < 243.5 6 7.638 1 ( 0.33333 0.66667 )

196) ï..age < 45.5 4 0.000 1 ( 0.00000 1.00000 ) \*

197) ï..age > 45.5 2 0.000 0 ( 1.00000 0.00000 ) \*

99) chol > 243.5 21 0.000 1 ( 0.00000 1.00000 ) \*

25) trestbps > 182 1 0.000 0 ( 1.00000 0.00000 ) \*

13) oldpeak > 3 2 0.000 0 ( 1.00000 0.00000 ) \*

7) ï..age > 55.5 42 55.820 1 ( 0.38095 0.61905 )

14) trestbps < 151 33 38.670 1 ( 0.27273 0.72727 )

28) trestbps < 137 18 24.730 1 ( 0.44444 0.55556 )

56) trestbps < 125.5 11 10.430 1 ( 0.18182 0.81818 )

112) chol < 279 8 0.000 1 ( 0.00000 1.00000 ) \*

113) chol > 279 3 3.819 0 ( 0.66667 0.33333 )

226) sex < 0.5 1 0.000 1 ( 0.00000 1.00000 ) \*

227) sex > 0.5 2 0.000 0 ( 1.00000 0.00000 ) \*

57) trestbps > 125.5 7 5.742 0 ( 0.85714 0.14286 )

114) ca < 2.5 6 0.000 0 ( 1.00000 0.00000 ) \*

115) ca > 2.5 1 0.000 1 ( 0.00000 1.00000 ) \*

29) trestbps > 137 15 7.348 1 ( 0.06667 0.93333 )

58) chol < 324 13 0.000 1 ( 0.00000 1.00000 ) \*

59) chol > 324 2 2.773 1 ( 0.50000 0.50000 ) \*

15) trestbps > 151 9 9.535 0 ( 0.77778 0.22222 )

30) sex < 0.5 1 0.000 1 ( 0.00000 1.00000 ) \*

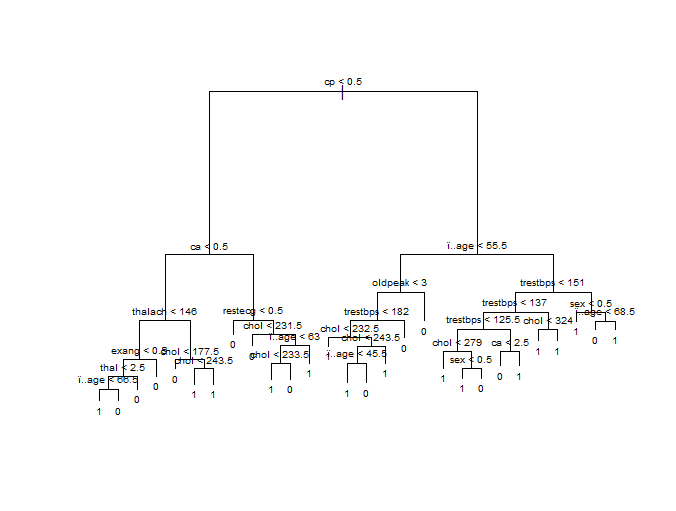
31) sex > 0.5 8 6.028 0 ( 0.87500 0.12500 )

62) ï..age < 68.5 6 0.000 0 ( 1.00000 0.00000 ) \*

63) ï..age > 68.5 2 2.773 1 ( 0.50000 0.50000 ) \*

**plot(pstree)**

**text(pstree,cex=0.7)**



**summary(pstree)**

Classification tree:

tree(formula = target ~ ., data = train\_heart, mincut = bestmincut,

minsize = bestminsize)

Variables actually used in tree construction:

[1] "cp" "ca" "thalach" "exang" "thal" "ï..age" "chol" "restecg" "oldpeak"

[10] "trestbps" "sex"

Number of terminal nodes: 28

Residual mean deviance: 0.09659 = 16.13 / 167

Misclassification error rate: 0.02564 = 5 / 195

**pred\_train<-predict(pstree,type = "class")**

**pred\_train**

**table(train\_heart$target,pred\_train)**

**#CONFUSİON MATRİX OF TRAİN DATA**

pred\_train

0 1

0 83 5

1. 1 106

Misclassification error rate=0.02564 = 5 / 195

ACCURACY=83+106/86+106+6=0.96

**pred\_test=predict(pstree,newdata=test\_heart,type="class")**

**table(test\_heart$target,pred\_test)**

**#CONFUSİON MATRİX OF TRAİN DATA**

pred\_test

0 1

0 33 17

1 13 45

ACCURACY=33+45/33+45+13+17=0.72

**c. Provide the Confusion Matrix along with sensitivity, specificity, precision and recall. Comment on which input attributes are important in making predictions.**

**#CONFUSİON MATRİX OF TEST DATA**

**** pred\_test

0 1

0 33 17

1 13 45

SENSİTİVİTY =RECALL = TPR= TP/TP+FN

SPECİFİCİTY=TNR= TN/TN+FP

PRECİSİON=TP/TP+FP

ACCURACY=(TP+TN)/(P+N)

SENSİTİVİTY =RECALL = TPR= 33/33+17=0.66

SPECİFİCİTY=TNR=45/58=0.75

PRECİSİON= 33/46=0.71

ACCURACY=33+45/33+45+13+17=0.72

The attribute CP is the most important attribute and also CA and AGE attributes. We can see this in the classification tree and also from the anova table. They are on top of the tree

**d. Repeat (a)-(c) with rpart package**

PART A

**library(rpart)**

**set.seed(425)**

**train=sample(1:303,195)**

**rptrain=heartdata[train,]**

**rptest=heartdata[-train,]**

PART B

**rp = rpart(target~.,data=rptrain,control=rpart.control(minsplit=3,minbucket=1,cp=0.001))**

**#CONFUSİON MATRİX OF TEST DATA**

prediction

0 1

0 36 12

1 15 45

SENSİTİVİTY =RECALL = TPR= 36/36+12=0.75

SPECİFİCİTY=TNR=45/60=0.75

PRECİSİON= 36/51=0.71

The results are very similar with the results that are found in former part.

**2. Consider the dataset called “suicide-rates.csv”. There are 11 attributes that are given below, where the last one is the output attribute (suicide rate).**

**country**

**year**

**sex**

**age**

**suicides\_no**

**population**

**country-year**

**gdp\_for\_year ($)**

**gdp\_per\_capita ($)**

**generation**

**suicides/100k pop**

**getwd()**

**setwd("C:/Users/Sinan/Desktop/dersler/ie 425/HW1")**

**suicide<- read.csv(file = "suicide-rate.csv",header=TRUE)**

**library(tree)**

**str(suicide)**

**a. Partition the data set into training and test sets with 70% going into the training set by using a seed value of 492.**

**set.seed(492)**

**train=sample(1:27820,19474)**

**suitrain=suicide[train,]**

**suitest=suicide[-train,]**

**suitrain$country<- NULL**

**suitrain$country.year<- NULL**

**suitrain$gdp\_for\_year.... <- NULL**

**suitest$country<- NULL**

**suitest$country.year<- NULL**

**suitest$gdp\_for\_year.... <- NULL**

**suicide$country<- NULL**

**suicide$country.year<- NULL**

**suicide$gdp\_for\_year.... <- NULL**

**b. By playing with the parameter values of the “tree” function of the “tree” package and using the training set, find the best tree in terms of the performance on the test set given in RMSE. Give the number of leaf nodes of the best tree and provide the summary of the tree using the summary(tree\_name) function with its corresponding performance (RMSE) on the training as well as test set.**

**error<-0**

**thebesterror<-10^10**

**for (i in 1:150) {**

**traintree<-tree (suicides.100k.pop~ .,data=suitrain,mincut=i)**

**yhat = predict(traintree, newdata=suitest)**

**sui\_test = suicide[-train,"suicides.100k.pop"]**

**error<-(mean((yhat-sui\_test)^2))^0.5**

**if(error<thebesterror){**

**thebesterror<-error**

**the\_best<-error**

**the\_best\_mincut<-i**

**}**

**}**

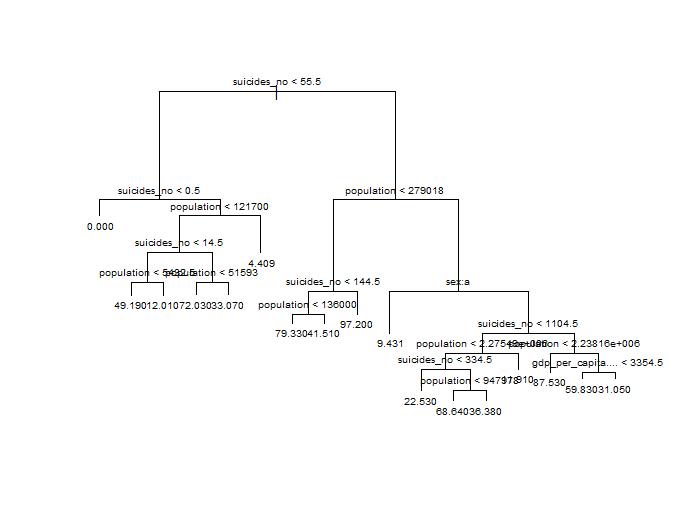
I have done this with only mincut because the data is very long and it lasts in many time to finish 3 for loop.

**the\_best #10.10881 The best RMSE in the test data according to mincut =88**

**the\_best\_mincut #88**

**the\_last\_tree<-tree (suicides.100k.pop~ .,data=suitrain, mincut=the\_best\_mincut)**

**plot(the\_last\_tree)**

**text(the\_last\_tree,cex=0.6)** 

**summary(the\_last\_tree)**

**c. Comment on which input attributes are important in making predictions**

The attribute SUİCİDES\_NO is the most important attribute and also POPULATİON attribute. We can see this in the classification tree and also from the anova table.They are on top of the tree