nodedataNAGini.R

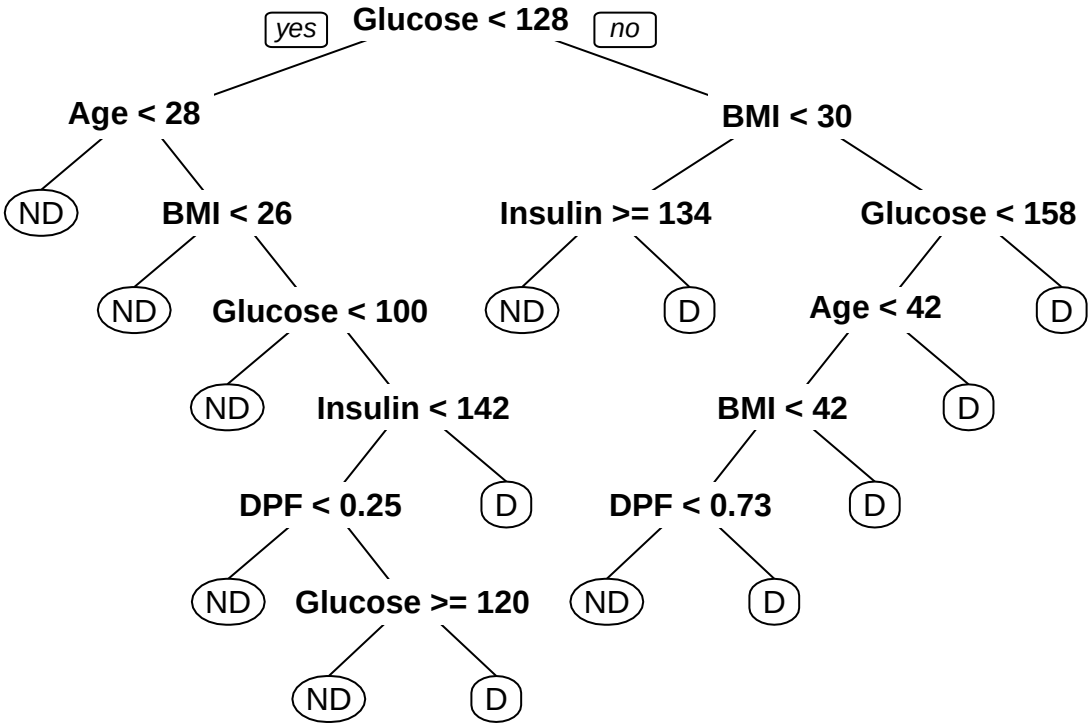
saurav

Mon Jan 19 21:41:21 2015

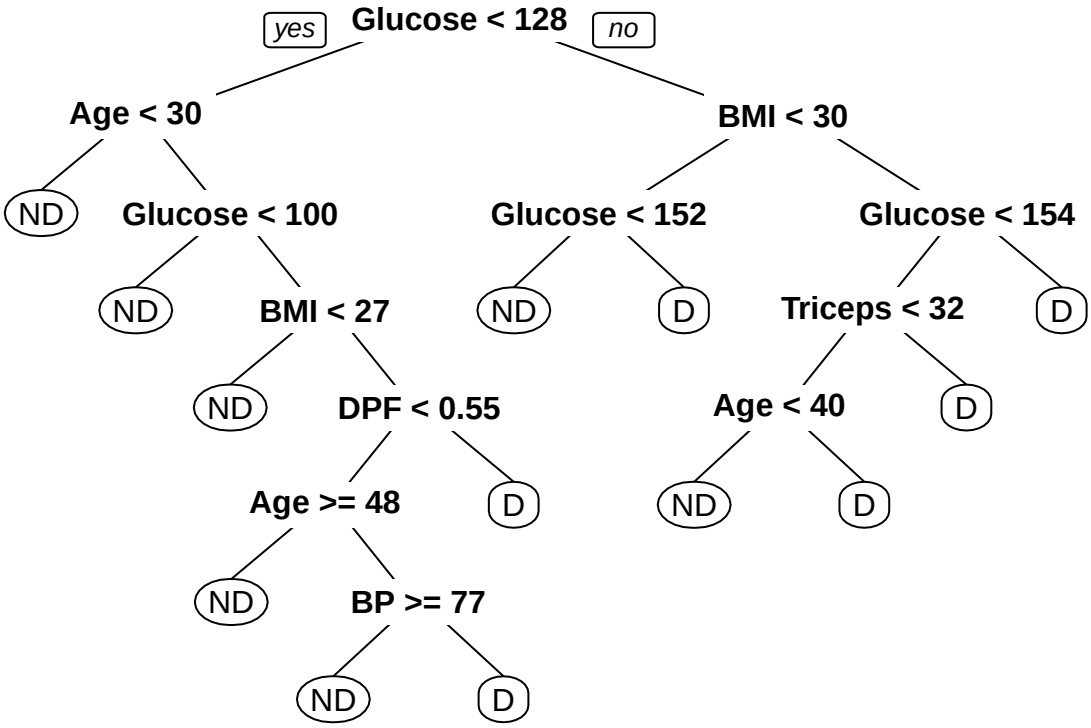
```
library(rpart)
library(rpart.plot)
set.seed(10)
rawData = read.csv(file="data", header=F, sep=",")
originalData = rawData[sample(nrow(rawData)),]
colnames(originalData) = c("PregnantCount","Glucose","BP","Triceps",
                        "Insulin","BMI","DPF","Age","Class")
N = nrow(originalData)
K = 5
foldWidth = floor(N/K)
Accuracy = 0
for (i in (1:K))
{
    data = originalData
    data$Glucose[data$Glucose==0] = NA          # Missing Data # c
    data$BP[data$BP==0] = NA
    data$Triceps[data$Triceps==0] = NA
    data$Insulin[data$Insulin==0] = NA
    data$BMI[data$BMI==0] = NA
    start = as.integer((i-1)*foldWidth)+1
    end = as.integer(i*foldWidth)
    if(i==K)
    {
        end = N
    }
    testData = data[c(start:end),]
    learnData = data[c(-start:-end),]
    diabStat = factor(learnData$Class, levels=0:1, labels=c('ND','D'))
    cfit = rpart(
                    diabStat ~ PregnantCount+Glucose+BP+Triceps+Insulin+BMI+DPF+Age,
                    data = learnData,
                    na.action = na.rpart,
                    method ='class',
                    parms = list(split = "gini"),
                    control = rpart.control(
                                                minsplit = 20,  # Min no. of obs. for which the routine
                                                minbucket = 9,  # Min no. of obs in leaf. Default = min
                                            )
                )
    predictedFactor = predict(cfit, testData, type="class")
    predictedFrame = as.data.frame.factor(predictedFactor)
    predicted = c(predictedFrame[ ,1]) - 1  # gives 1 for ND, 2 for D
    actual = testData$Class
    TP = sum(predicted & actual)
    TN = nrow(testData) - sum(predicted | actual)
    # Accuracy
    print((TP+TN)/nrow(testData))
```

```
    Accuracy = Accuracy + (TP+TN)/nrow(testData)
    rpart.plot(cfit)
}
```

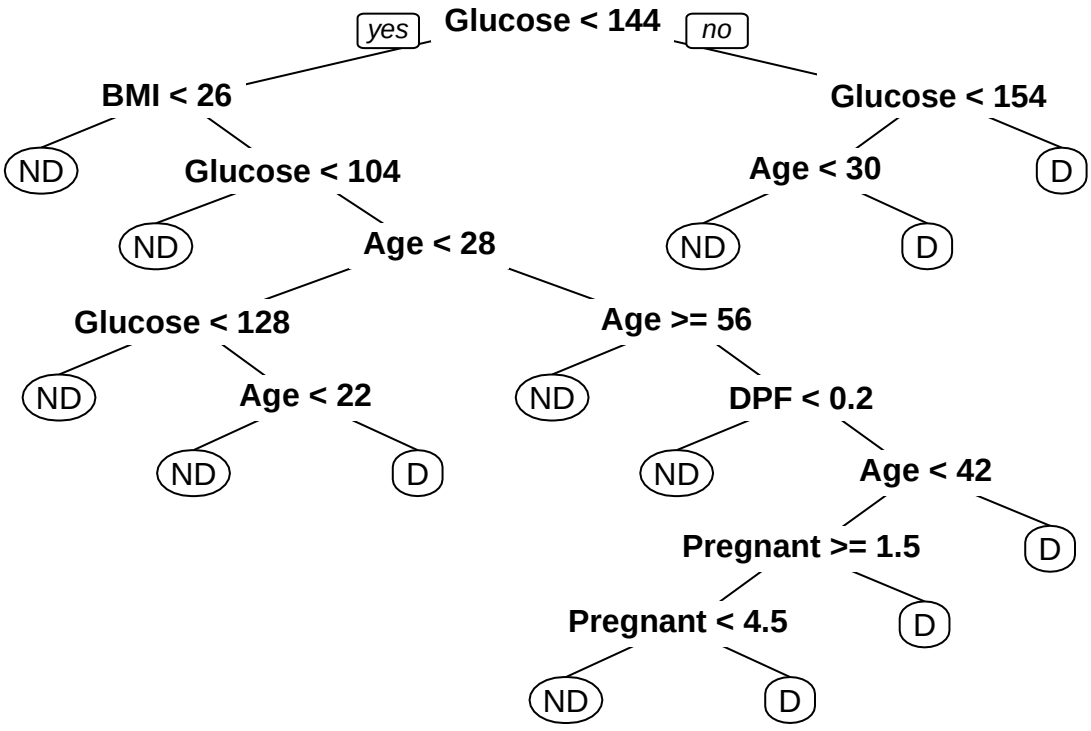
```
## [1] 0.6862745
```



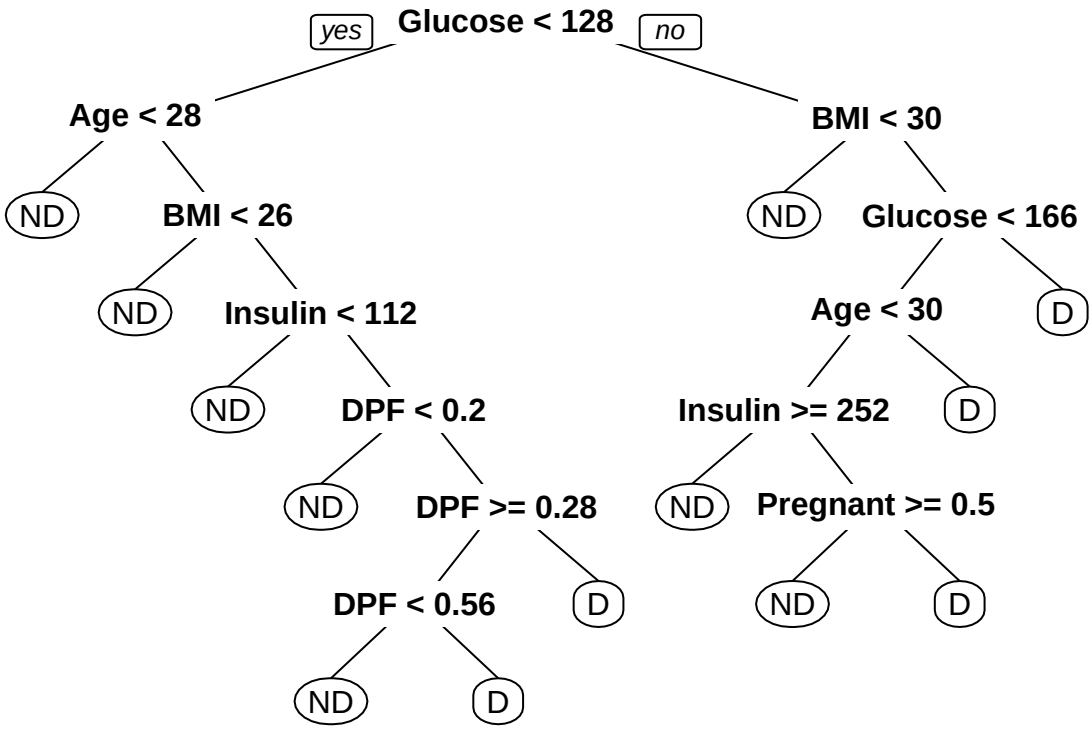

```
## [1] 0.751634
```



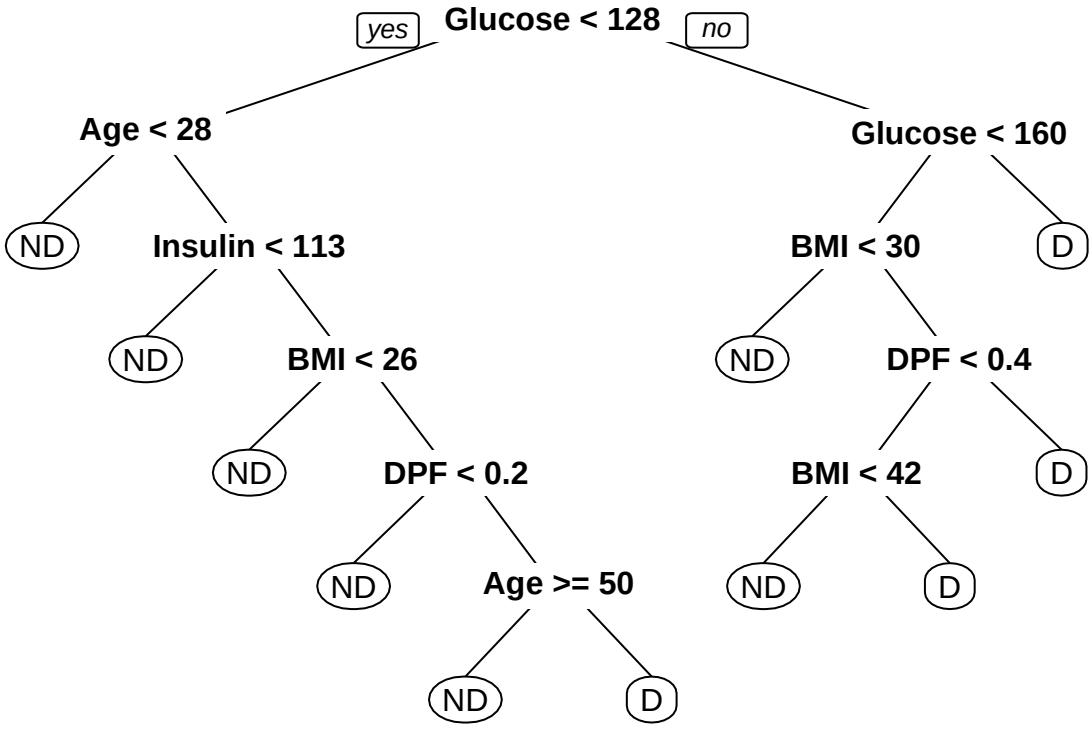

```
## [1] 0.7189542
```




```
## [1] 0.7777778
```




```
## [1] 0.8397436
```




```
# Mean Accuracy
print(Accuracy/K)
```

```
## [1] 0.7548768
```