dt_lab.R

alexaubrey

Wed May 23 18:29:49 2018

```
library(rpart)
#library(rattle)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.4
```

```
## Loading required package: lattice
```

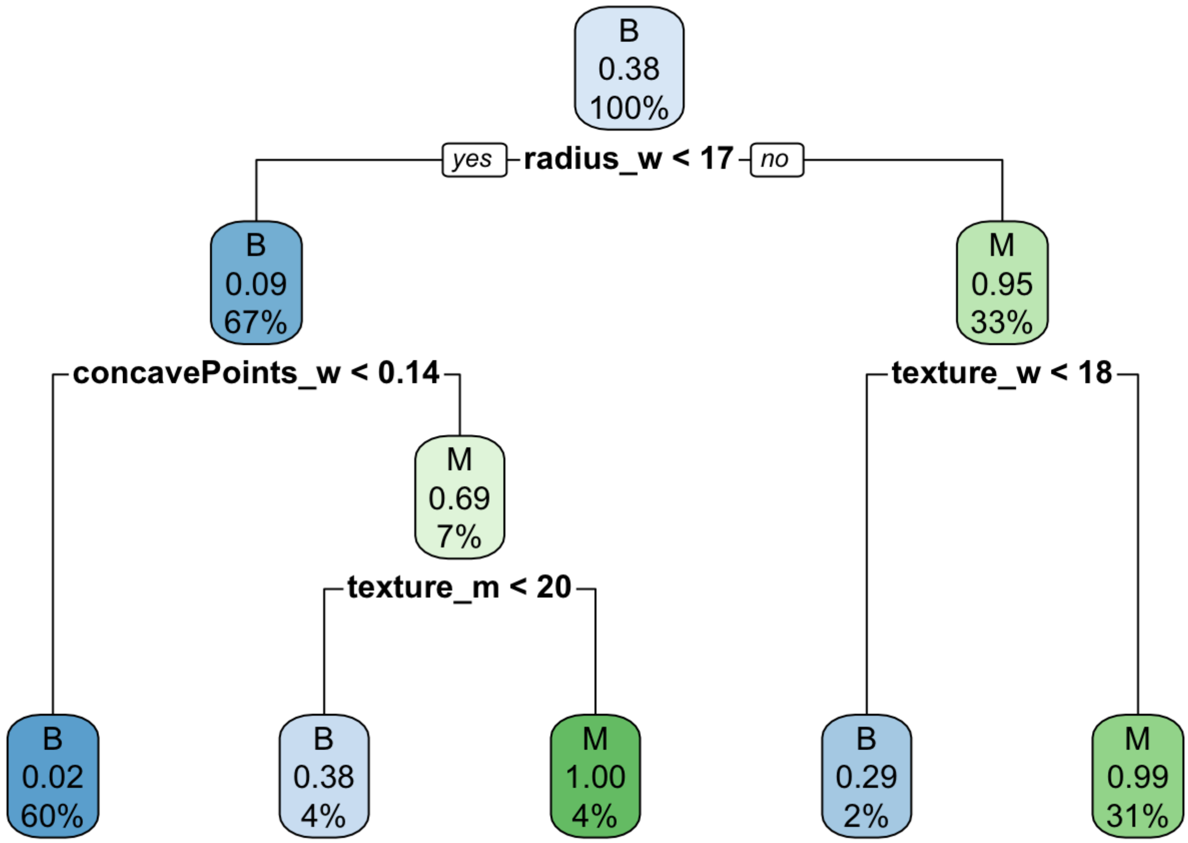
```
## Loading required package: ggplot2
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'zone/tz/2018c.
## 1.0/zoneinfo/America/Chicago'
```

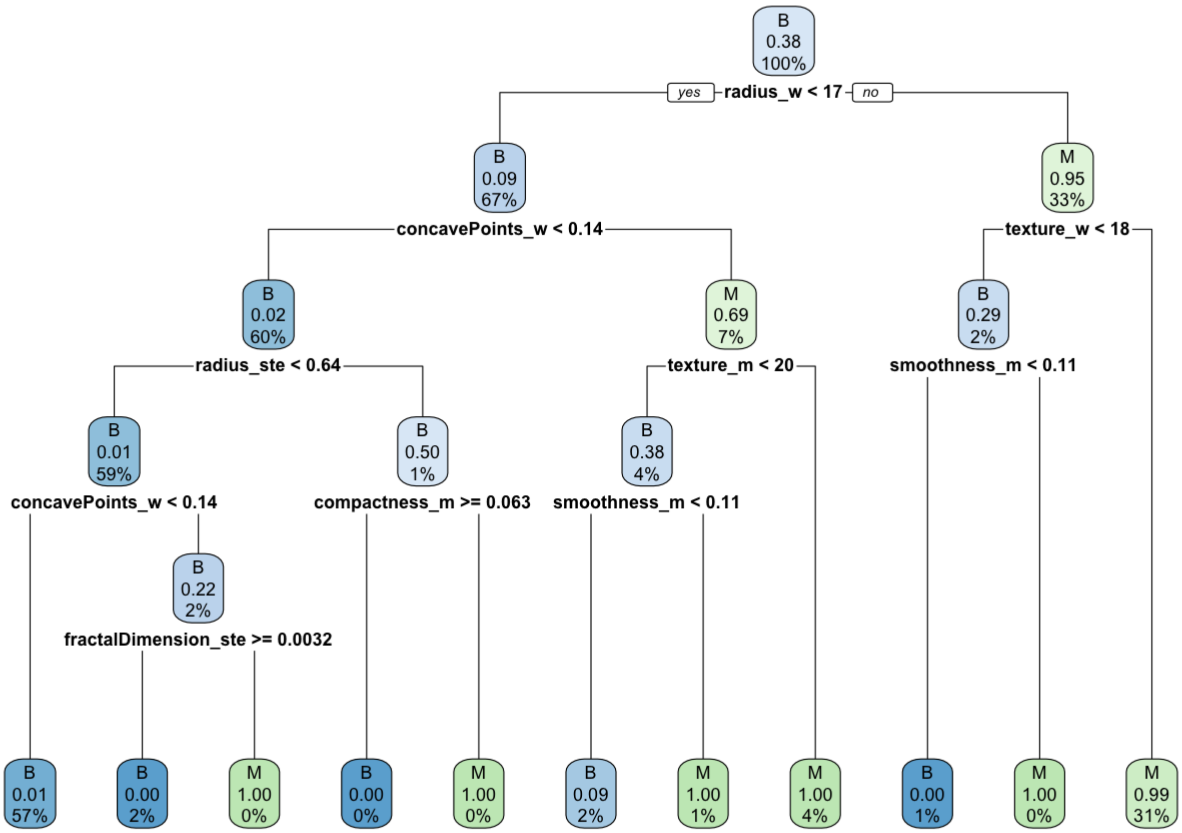
```
data <- read.csv("http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer
-wisconsin/wdbc.data",
                 col.names = c("ID", "Diagnosis","radius_m","texture_m","perimeter_m","a
rea_m",
                               "smoothness_m","compactness_m", "concavity_m",
                               "concavePoints_m","symmetry_m","fractalDimension_m",
                               "radius_ste","texture_ste","perimeter_ste","area_ste",
                               "smoothness_ste","compactness_ste", "concavity_ste",
                               "concavePoints_ste","symmetry_ste","fractalDimension_ste"
,
                               "radius_w","texture_w","perimeter_w","area_w",
                               "smoothness_w","compactness_w", "concavity_w",
                               "concavePoints_w","symmetry_w","fractalDimension_w"),
                 header=FALSE)


train <- sample(1:569, 455)
test <- setdiff(1:569, train)

data_train = data[train,]
data_test = subset(data[test,], select =-Diagnosis)

rpartTree <- rpart(Diagnosis ~ ., data=data_train)

out = predict(rpartTree, data_test, type="class")

confusionMatrix(out, data[test,]$Diagnosis)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 70  6
##          M  3 35
##
##                Accuracy : 0.9211
##                  95% CI : (0.8554, 0.9633)
##     No Information Rate : 0.6404
##     P-Value [Acc > NIR] : 3.615e-12
##
##                   Kappa : 0.8258
##  Mcnemar's Test P-Value : 0.505
##
##             Sensitivity : 0.9589
##             Specificity : 0.8537
##          Pos Pred Value : 0.9211
##          Neg Pred Value : 0.9211
##              Prevalence : 0.6404
##          Detection Rate : 0.6140
##    Detection Prevalence : 0.6667
##       Balanced Accuracy : 0.9063
##
##        'Positive' Class : B
##
```

```
## Could not get Rattle Installed
#fancyRpartPlot(rpartTree)
rpart.plot::rpart.plot(rpartTree)
```




```
temp <- rpart.control(xval=10, minbucket = 2, minsplit = 4, cp = 0)
dfit <- rpart(Diagnosis ~ ., data=data_train, control=temp)

## Could not get Rattle Installed
#fancyRpartPlot(dfit)
rpart.plot::rpart.plot(dfit)
```




```
library(caret)
#10-folds cross validation
fitControl <- trainControl(method='cv', number=10)

Grid <- expand.grid(cp=seq(0,0.05, 0.005))
#Run the training, needed to remove NAs
trained_tree <- train(Diagnosis ~ ., data=data_train, method='rpart',
                      trControl=fitControl, metric='Accuracy', maximize=TRUE, tuneGrid=G
rid)
trained_tree
```

```
## CART
##
## 455 samples
##  31 predictor
##   2 classes: 'B', 'M'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 409, 410, 410, 410, 409, 409, ...
## Resampling results across tuning parameters:
##
##   cp     Accuracy   Kappa
##   0.000  0.9362761  0.8625088
##   0.005  0.9340539  0.8577303
##   0.010  0.9230373  0.8363293
##   0.015  0.9230373  0.8365792
##   0.020  0.9252595  0.8407133
##   0.025  0.9252595  0.8407133
##   0.030  0.9230373  0.8355891
##   0.035  0.9230373  0.8360161
##   0.040  0.9141484  0.8193825
##   0.045  0.9141484  0.8193825
##   0.050  0.8987820  0.7863548
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.
```

```
out2 = predict(trained_tree, data_test, type="raw")
confusionMatrix(out2, data[test, ]$Diagnosis)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 70  6
##          M  3 35
##
##                Accuracy : 0.9211
##                  95% CI : (0.8554, 0.9633)
##     No Information Rate : 0.6404
##     P-Value [Acc > NIR] : 3.615e-12
##
##                   Kappa : 0.8258
##  Mcnemar's Test P-Value : 0.505
##
##             Sensitivity : 0.9589
##             Specificity : 0.8537
##          Pos Pred Value : 0.9211
##          Neg Pred Value : 0.9211
##              Prevalence : 0.6404
##          Detection Rate : 0.6140
##    Detection Prevalence : 0.6667
##       Balanced Accuracy : 0.9063
##
##        'Positive' Class : B
##
```

```
### BAGGING ###
library(ipred)
baggedTree <- bagging(Diagnosis ~ ., data=data_train)
out3 = predict(baggedTree,data_test)
confusionMatrix(out3, data[test,]$Diagnosis)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 70  3
##          M  3 38
##
##                Accuracy : 0.9474
##                  95% CI : (0.889, 0.9804)
##     No Information Rate : 0.6404
##     P-Value [Acc > NIR] : 7.914e-15
##
##                   Kappa : 0.8857
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9589
##             Specificity : 0.9268
##          Pos Pred Value : 0.9589
##          Neg Pred Value : 0.9268
##              Prevalence : 0.6404
##          Detection Rate : 0.6140
##    Detection Prevalence : 0.6404
##       Balanced Accuracy : 0.9429
##
##        'Positive' Class : B
##
```

```
baggedTree <- bagging(Diagnosis ~ ., data=data_train, nbagg = 4)

#needed to remove NAs in training set
mod <- train(Diagnosis ~ ., data=data_train,
             method="treebag",
             trControl=fitControl,
             metric='Accuracy',
             maximize=TRUE)

print(mod)
```

```
## Bagged CART
##
## 455 samples
##  31 predictor
##   2 classes: 'B', 'M'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 409, 410, 410, 410, 408, 409, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9491746  0.8929245
```

```
## Random Forest ##
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.4.4
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
rfModel <- randomForest(Diagnosis ~ ., data = data_train)
rfModel
```

```
##
## Call:
##  randomForest(formula = Diagnosis ~ ., data = data_train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 3.74%
## Confusion matrix:
##     B   M class.error
## B 277   7  0.02464789
## M  10 161  0.05847953
```

```
out5 = predict(rfModel, data_test)
confusionMatrix(out5, data[test,]$Diagnosis)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 72  3
##          M  1 38
##
##                Accuracy : 0.9649
##                  95% CI : (0.9126, 0.9904)
##     No Information Rate : 0.6404
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.923
##  Mcnemar's Test P-Value : 0.6171
##
##             Sensitivity : 0.9863
##             Specificity : 0.9268
##          Pos Pred Value : 0.9600
##          Neg Pred Value : 0.9744
##              Prevalence : 0.6404
##          Detection Rate : 0.6316
##    Detection Prevalence : 0.6579
##       Balanced Accuracy : 0.9566
##
##        'Positive' Class : B
##
```

```
rfModel <- randomForest(Diagnosis ~ ., data = data_train, ntree=10, mtry=4)
rfModel
```

```
##
## Call:
##  randomForest(formula = Diagnosis ~ ., data = data_train, ntree = 10,      mtry = 4)
##                Type of random forest: classification
##                      Number of trees: 10
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 5.57%
## Confusion matrix:
##     B   M class.error
## B 275   4  0.01433692
## M  21 149  0.12352941
```

```
control <- trainControl(method="repeatedcv", number = 10, repeats=3)
metric <- "Accuracy"
n <- round(sqrt(ncol(data_train)))
tunegrid <- expand.grid(.mtry=seq(4,n,1))
rf_default <- train(Diagnosis ~ ., data=data_train, method="rf", metric=metric, tuneGrid
=tunegrid,
                    trainControl = control)
print(rf_default)
```

```
## Random Forest
##
## 455 samples
##  31 predictor
##   2 classes: 'B', 'M'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 455, 455, 455, 455, 455, 455, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   4     0.9572637  0.9081601
##   5     0.9577480  0.9092482
##   6     0.9573065  0.9081881
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 5.
```

```
#variable importance
Imp <- varImp(rpartTree)
Imp
```

```
##                          Overall
## area_w                148.012959
## compactness_w          17.928482
## concavePoints_m       173.137046
## concavePoints_w       170.913906
## concavity_m            22.051825
## concavity_w            20.263401
## perimeter_w           145.541153
## radius_w              149.262449
## smoothness_m            4.276316
## symmetry_w              5.315714
## texture_m              12.156510
## texture_ste             4.043969
## texture_w              12.795468
## ID                      0.000000
## radius_m                0.000000
## perimeter_m             0.000000
## area_m                  0.000000
## compactness_m           0.000000
## symmetry_m              0.000000
## fractalDimension_m      0.000000
## radius_ste              0.000000
## perimeter_ste           0.000000
## area_ste                0.000000
## smoothness_ste          0.000000
## compactness_ste         0.000000
## concavity_ste           0.000000
## concavePoints_ste       0.000000
## symmetry_ste            0.000000
## fractalDimension_ste    0.000000
## smoothness_w            0.000000
## fractalDimension_w      0.000000
```

```
Imp <- varImp(mod)
Imp
```

```
## treebag variable importance
##
##   only 20 most important variables shown (out of 31)
##
##                  Overall
## area_w           100.000
## radius_w          99.690
## perimeter_w       97.265
## concavePoints_w   94.557
## concavePoints_m   90.197
## area_ste          11.609
## area_m             7.478
## perimeter_m        7.431
## radius_m           6.811
## texture_m          6.473
## texture_w          6.272
## concavity_w        4.443
## concavity_m        4.070
## smoothness_w       3.936
## ID                 3.659
## radius_ste         2.976
## compactness_w      2.722
## symmetry_w         2.051
## compactness_m      1.941
## smoothness_m       1.779
```