MetaDatasetClassification.R

dieudonneouedraogo

Sun Nov 11 22:50:23 2018

```
#install.packages("caret")
#install.packages("caret", dependencies=c("Depends", "Suggests"))
library(caret)
```

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

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

```
DummyChar <- read.csv("~/DummyChar.csv")
#My Meta Dataset
dataset <- DummyChar
dataset
```

```
##        X  Al     Density     ClsCoef        Hubs
## 1      1 Knn 0.434580803 0.505828630 0.834303992
## 2      2 ANN 0.090080537 0.224253573 0.436768870
## 3      3 Knn 0.419699779 0.155886438 0.273755538
## 4      4 ANN 0.052599533 0.369392853 0.007740361
## 5      5 ANN 0.256935979 0.969474012 0.294008983
## 6      6 Knn 0.368179576 0.110480352 0.825518414
## 7      7 ANN 0.670263665 0.134672191 0.188341918
## 8      8 SVM 0.327016433 0.848711706 0.972758641
## 9      9 SVM 0.934182715 0.660916655 0.727416854
## 10    10  DT 0.793256408 0.397110511 0.998485462
## 11    11 ANN 0.832247808 0.139800921 0.567296213
## 12    12 ANN 0.955299830 0.582108269 0.027239352
## 13    13  DT 0.751189198 0.441838027 0.636733853
## 14    14  DT 0.247397340 0.274850797 0.467445820
## 15    15 ANN 0.635009364 0.096827013 0.212732401
## 16    16 GLM 0.477511239 0.127507338 0.912666553
## 17    17 GLM 0.075253368 0.555477221 0.131263494
## 18    18 Knn 0.714617108 0.749057634 0.292804381
## 19    19 GLM 0.279828392 0.433155332 0.310376519
## 20    20  DT 0.083322857 0.614668286 0.229193623
## 21    21 SVM 0.774320417 0.394164665 0.459603766
## 22    22 SVM 0.742131842 0.296841853 0.609014487
## 23    23  DT 0.917339455 0.066138336 0.830549632
## 24    24  DT 0.706701624 0.557290955 0.594914233
## 25    25 ANN 0.531069254 0.627113671 0.246578350
## 26    26 GLM 0.660572416 0.348658987 0.635287225
## 27    27 GLM 0.505592469 0.325275371 0.344468823
## 28    28  DT 0.067123532 0.771095380 0.061027562
## 29    29 ANN 0.839457619 0.998739472 0.378727464
## 30    30  DT 0.534837588 0.838920244 0.377442583
## 31    31 SVM 0.294662886 0.328155127 0.282598115
## 32    32 Knn 0.647637736 0.346642039 0.279111577
## 33    33 Knn 0.892987225 0.268089701 0.303393231
## 34    34 Knn 0.789889335 0.372382391 0.374572926
## 35    35  DT 0.327302751 0.923850395 0.554780233
```

```
## 36   36  DT 0.385755476 0.549616409 0.649107078
## 37   37 ANN 0.844058162 0.778503740 0.161517241
## 38   38 SVM 0.149129537 0.400834924 0.434228894
## 39   39 Knn 0.905944139 0.488055433 0.931709651
## 40   40 Knn 0.490494096 0.846814046 0.920801261
## 41   41 SVM 0.849580212 0.565812823 0.076611118
## 42   42 ANN 0.524455148 0.608506304 0.829641144
## 43   43 ANN 0.435558229 0.686716855 0.890104957
## 44   44  DT 0.377797641 0.717299161 0.678433914
## 45   45 ANN 0.842881087 0.309635599 0.120701248
## 46   46 SVM 0.396095385 0.717672178 0.521147619
## 47   47 ANN 0.433861117 0.652951320 0.783085904
## 48   48 ANN 0.619266049 0.241513318 0.112547658
## 49   49 GLM 0.747957997 0.822065714 0.203100086
## 50   50 SVM 0.635721548 0.010247614 0.981186080
## 51   51  DT 0.915156104 0.672113417 0.188252902
## 52   52 ANN 0.381690593 0.063075250 0.492589360
## 53   53  DT 0.136195937 0.315327777 0.499114726
## 54   54 Knn 0.394717490 0.947672809 0.518582083
## 55   55 GLM 0.452000631 0.925481438 0.121044706
## 56   56 GLM 0.955698144 0.278794164 0.629601354
## 57   57 Knn 0.809108085 0.160447626 0.367712016
## 58   58 SVM 0.677199753 0.828859241 0.855054024
## 59   59 ANN 0.942715781 0.297373000 0.386013612
## 60   60 ANN 0.321297069 0.194484663 0.540044380
## 61   61 GLM 0.840824933 0.358876864 0.053578788
## 62   62 GLM 0.761308599 0.025766444 0.963520196
## 63   63 ANN 0.333991461 0.536774381 0.217451517
## 64   64  DT 0.383448350 0.727733783 0.347446445
## 65   65 ANN 0.520947080 0.470624811 0.232816439
## 66   66 GLM 0.932378297 0.989035743 0.763457848
## 67   67 ANN 0.908609169 0.689000950 0.336789932
## 68   68 SVM 0.237595514 0.947060585 0.101169857
## 69   69 GLM 0.568510177 0.803745979 0.852907582
## 70   70 Knn 0.121289274 0.369490425 0.735202145
## 71   71 Knn 0.464104182 0.054309928 0.114017973
## 72   72 GLM 0.001329487 0.497923074 0.853308586
## 73   73 Knn 0.785753832 0.006241411 0.089414136
## 74   74 ANN 0.271361142 0.791904239 0.340363621
## 75   75 ANN 0.395177545 0.179661431 0.305195317
## 76   76 Knn 0.571995028 0.368167469 0.155040714
## 77   77 GLM 0.396877033 0.142592615 0.640644673
## 78   78  DT 0.214104882 0.383716457 0.493126134
## 79   79 SVM 0.266732447 0.327382949 0.722287842
## 80   80 ANN 0.921491834 0.343418207 0.183504572
## 81   81 Knn 0.257919340 0.544612807 0.475219886
## 82   82 Knn 0.818323778 0.673514441 0.496796929
## 83   83 ANN 0.011563461 0.742580002 0.406963136
## 84   84 SVM 0.431548402 0.441676070 0.837973147
## 85   85 GLM 0.961294862 0.308448579 0.166292246
## 86   86 ANN 0.173327772 0.763190150 0.343508566
## 87   87 Knn 0.236495495 0.920607767 0.386814666
## 88   88 Knn 0.150972271 0.709880081 0.668038144
## 89   89 SVM 0.118577880 0.908701573 0.011638352
```

```
## 90   90 GLM 0.326196855 0.013256095 0.751364159
## 91   91 SVM 0.041436470 0.825745783 0.860852013
## 92   92  DT 0.799129178 0.355889887 0.582804612
## 93   93 SVM 0.199266033 0.251417209 0.375890657
## 94   94  DT 0.321273886 0.342287786 0.845589792
## 95   95 ANN 0.636599988 0.398546886 0.332593503
## 96   96 Knn 0.659727328 0.808344178 0.669480772
## 97   97 GLM 0.259194372 0.473933065 0.777932419
## 98   98 GLM 0.722354265 0.921819190 0.944273145
## 99   99 SVM 0.813069060 0.880828673 0.986080467
## 100 100 GLM 0.603607563 0.028597273 0.966878956
```

```
validation_index <- createDataPartition(dataset$Al, p=0.80, list=FALSE)
# select 20% of the data for validation
validation <- dataset[-validation_index,]
# use the remaining 80% of data to training and testing the models
dataset <- dataset[validation_index,]

# dimensions of dataset
dim(dataset)
```

```
## [1] 82  5
```

```
# list types for each attribute
sapply(dataset, class)
```

```
##         X        Al   Density   ClsCoef      Hubs
## "integer"  "factor" "numeric" "numeric" "numeric"
```

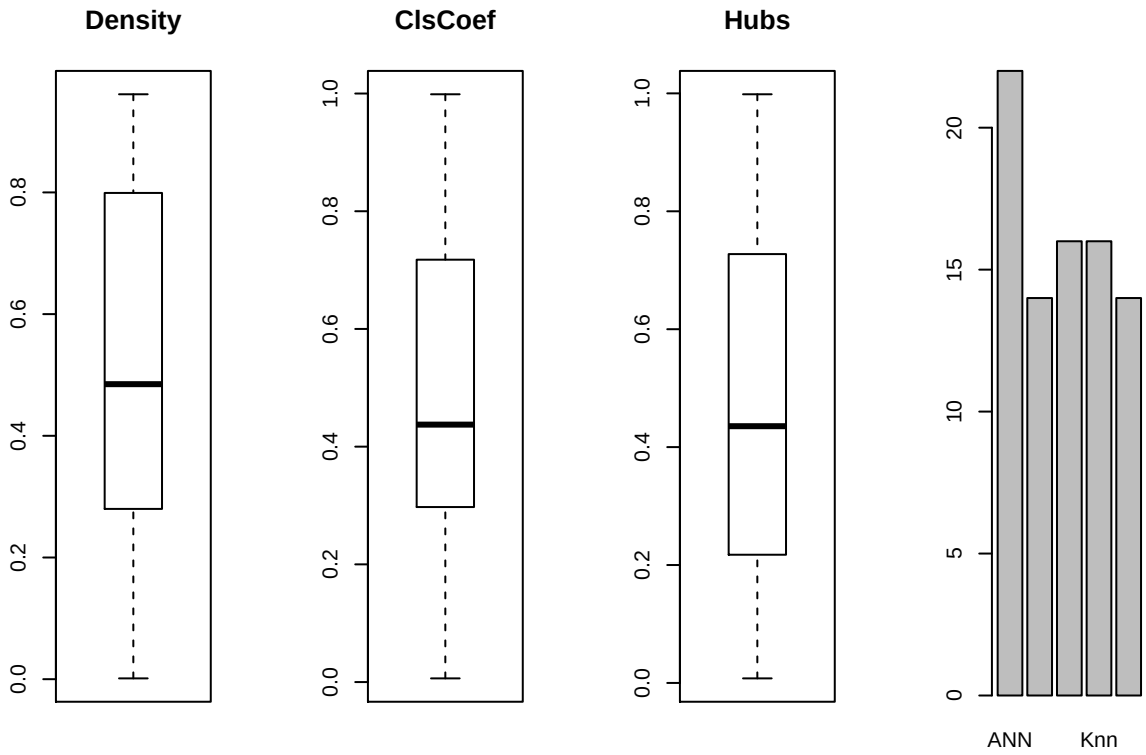
```
# list the levels for the class
levels(dataset$Al)
```

```
## [1] "ANN" "DT"  "GLM" "Knn" "SVM"
```

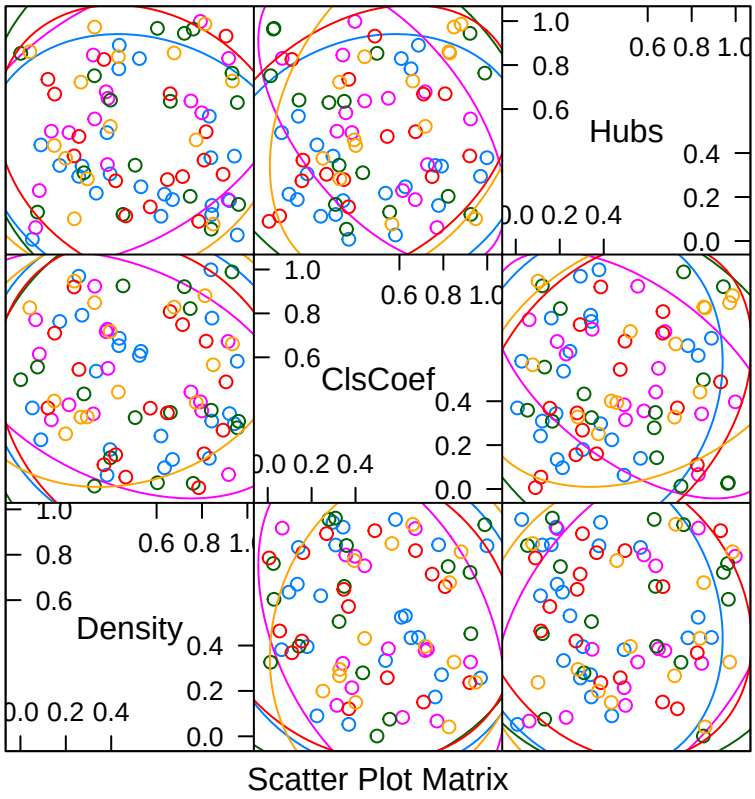
```
# summarize the class distribution
percentage <- prop.table(table(dataset$Al)) * 100
cbind(freq=table(dataset$Al), percentage=percentage)
```

```
##     freq percentage
## ANN   22   26.82927
## DT    14   17.07317
## GLM   16   19.51220
## Knn   16   19.51220
## SVM   14   17.07317
```

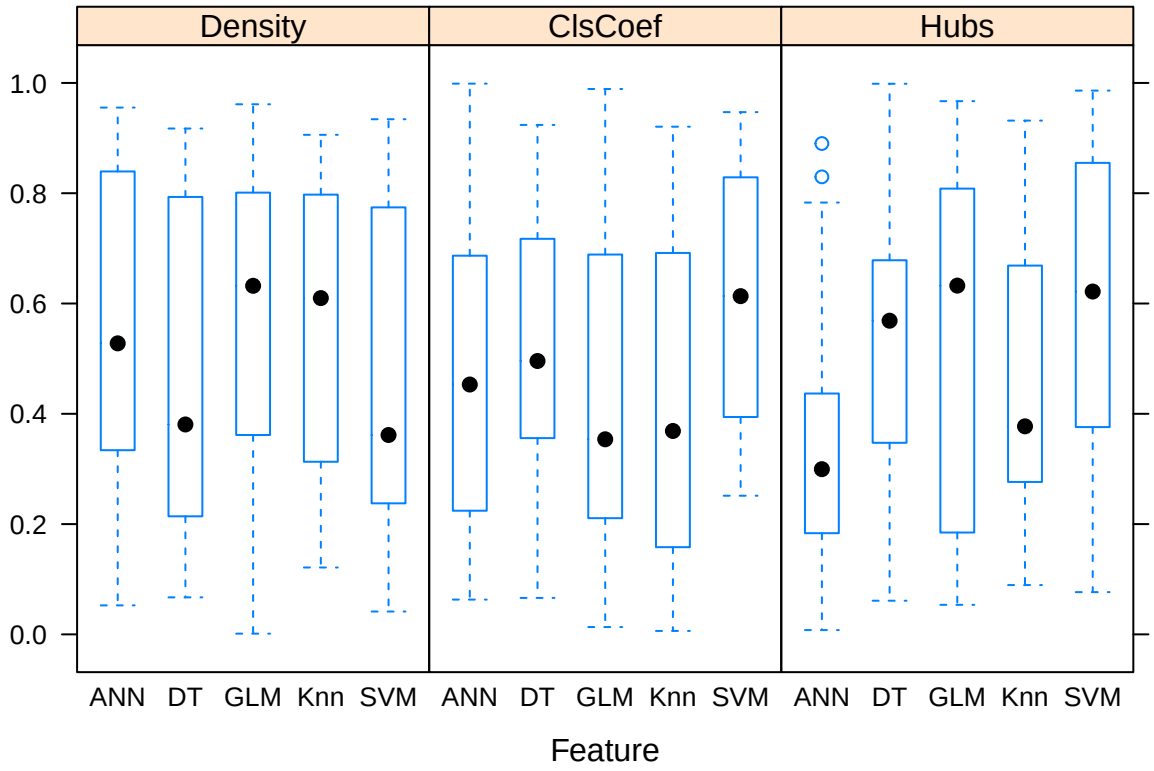
```
# split input and output
x <- dataset[,3:5]
y <- dataset[,2]
# boxplot for each attribute on one image
par(mfrow=c(1,4))
for(i in 1:3) {
  boxplot(x[i], main=names(x)[i])
}

# barplot for class breakdown
plot(y)
```



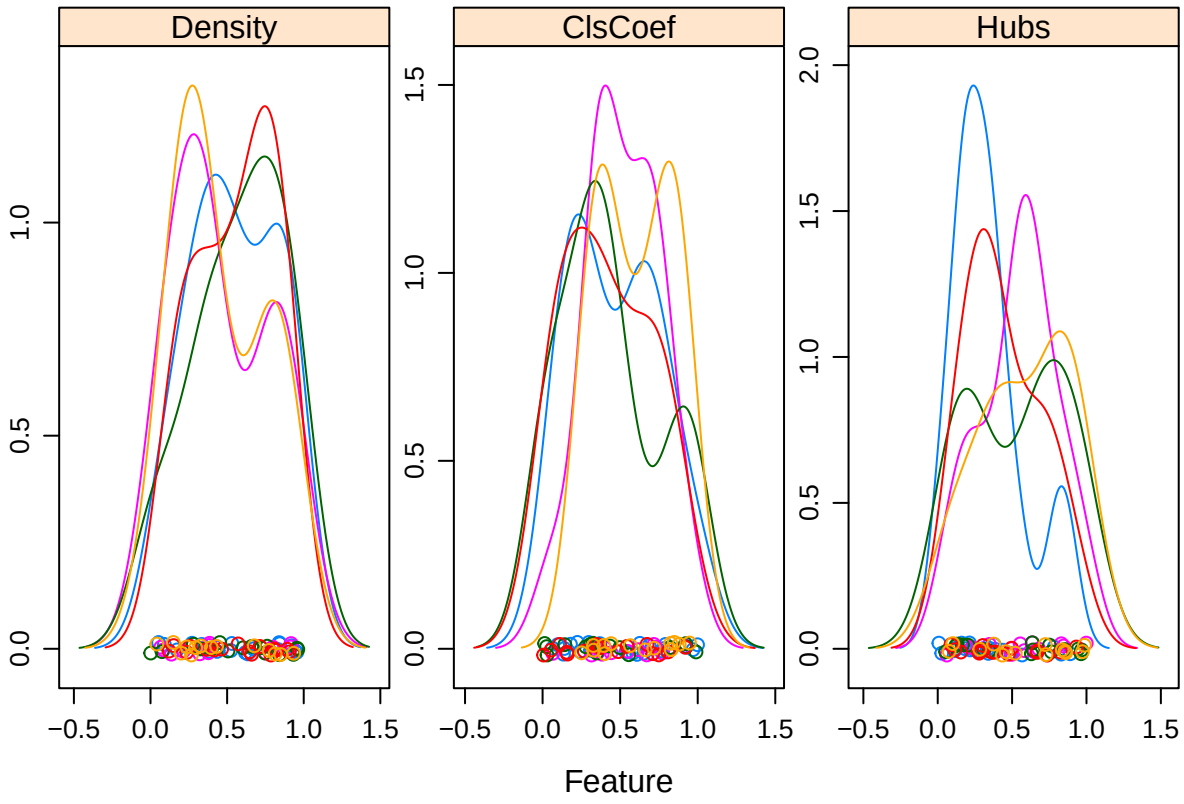

```
# scatterplot matrix
featurePlot(x=x, y=y, plot="ellipse")
```




```
# box and whisker plots for each attribute
featurePlot(x=x, y=y, plot="box")
```




```
# density plots for each attribute by class value
scales <- list(x=list(relation="free"), y=list(relation="free"))
featurePlot(x=x, y=y, plot="density", scales=scales)
```

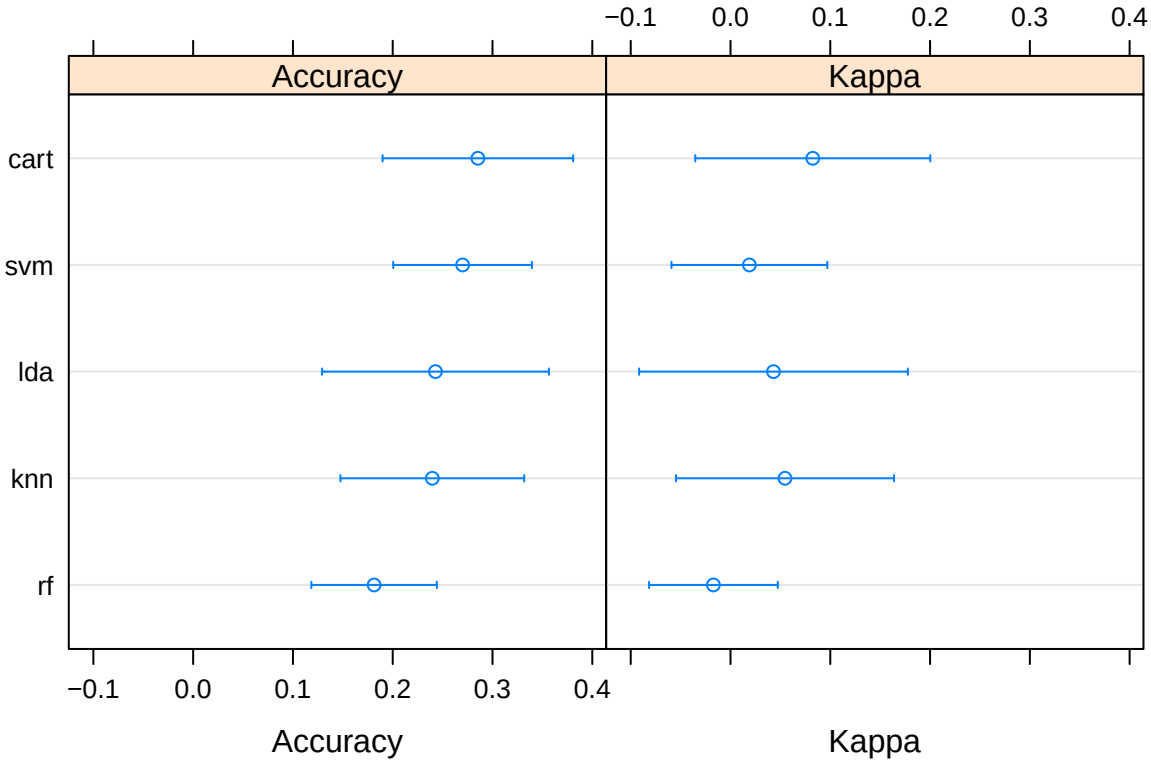



```
# Run algorithms using 10-fold cross validation
control <- trainControl(method="cv", number=10)
metric <- "Accuracy"

# a) linear algorithms
set.seed(7)
fit.lda <- train(Al~., data=dataset, method="lda", metric=metric, trControl=control)
# b) nonlinear algorithms
# CART
set.seed(7)
fit.cart <- train(Al~., data=dataset, method="rpart", metric=metric, trControl=control)
# kNN
set.seed(7)
fit.knn <- train(Al~., data=dataset, method="knn", metric=metric, trControl=control)
# c) advanced algorithms
# SVM
set.seed(7)
fit.svm <- train(Al~., data=dataset, method="svmRadial", metric=metric, trControl=control)
# Random Forest
set.seed(7)
fit.rf <- train(Al~., data=dataset, method="rf", metric=metric, trControl=control)

# summarize accuracy of models
results <- resamples(list(lda=fit.lda, cart=fit.cart, knn=fit.knn, svm=fit.svm, rf=fit.rf))
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
```

```
##
## Models: lda, cart, knn, svm, rf
## Number of resamples: 10
##
## Accuracy
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lda  0.0000000 0.1294643 0.2222222 0.2428571 0.3526786 0.5000000    0
## cart 0.1111111 0.1493056 0.3303571 0.2853175 0.3750000 0.4444444    0
## knn  0.1111111 0.1294643 0.1825397 0.2396825 0.3645833 0.4444444    0
## svm  0.1111111 0.2500000 0.2678571 0.2700397 0.3333333 0.4285714    0
## rf   0.0000000 0.1250000 0.2222222 0.1813492 0.2430556 0.2857143    0
##
## Kappa
##            Min.     1st Qu.      Median        Mean    3rd Qu.      Max.
## lda  -0.1851852 -0.10000000 -0.01851369  0.04308563 0.20360713 0.3725490
## cart -0.2000000 -0.04245283  0.13461538  0.08246391 0.20343137 0.2741935
## knn  -0.1428571 -0.07019231  0.01176471  0.05462139 0.18006279 0.2857143
## svm  -0.1250000  0.00000000  0.00000000  0.01892863 0.02593537 0.2432432
## rf   -0.1951220 -0.08233276  0.01801658 -0.01711677 0.04409091 0.1025641
##      NA's
## lda     0
## cart    0
## knn     0
## svm     0
## rf      0
```

```
# compare accuracy of models
dotplot(results)
```



```r
# summarize Best Model
print(fit.rf)
```

```
## Random Forest
##
## 82 samples
##  4 predictor
##  5 classes: 'ANN', 'DT', 'GLM', 'Knn', 'SVM'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 73, 73, 75, 74, 74, 74, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.1813492  -0.01711677
##   3     0.1720238  -0.02874688
##   4     0.1799603  -0.01127892
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

```r
# estimate skill of LDA on the validation dataset
predictions <- predict(fit.rf, validation)
confusionMatrix(predictions, validation$Al)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction ANN DT GLM Knn SVM
##        ANN   3  2   0   2   0
##        DT    0  1   0   0   0
##        GLM   0  0   0   0   2
##        Knn   2  0   1   1   0
##        SVM   0  0   2   1   1
##
## Overall Statistics
##
##                Accuracy : 0.3333
##                  95% CI : (0.1334, 0.5901)
##     No Information Rate : 0.2778
##     P-Value [Acc > NIR] : 0.3825
##
##                   Kappa : 0.1429
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: ANN Class: DT Class: GLM Class: Knn Class: SVM
## Sensitivity              0.6000   0.33333     0.0000    0.25000    0.33333
## Specificity              0.6923   1.00000     0.8667    0.78571    0.80000
## Pos Pred Value           0.4286   1.00000     0.0000    0.25000    0.25000
## Neg Pred Value           0.8182   0.88235     0.8125    0.78571    0.85714
## Prevalence               0.2778   0.16667     0.1667    0.22222    0.16667
## Detection Rate           0.1667   0.05556     0.0000    0.05556    0.05556
```

```
## Detection Prevalence     0.3889   0.05556     0.1111    0.22222    0.22222
## Balanced Accuracy        0.6462   0.66667     0.4333    0.51786    0.56667
```

```
#Al=sample(c("SVM","ANN","Knn","GLM","DT"),100,replace=T)
```