Statistics in R: Task 22

Liuaza Etezova

```
library(mlbench)
library(dplyr)
library(randomForest)
```

Random Forest

```
data(Vehicle)
df <- Vehicle
str(df)
```

```
## 'data.frame':    846 obs. of  19 variables:
##  $ Comp        : num  95 91 104 93 85 107 97 90 86 93 ...
##  $ Circ        : num  48 41 50 41 44 57 43 43 34 44 ...
##  $ D.Circ      : num  83 84 106 82 70 106 73 66 62 98 ...
##  $ Rad.Ra      : num  178 141 209 159 205 172 173 157 140 197 ...
##  $ Pr.Axis.Ra  : num  72 57 66 63 103 50 65 65 61 62 ...
##  $ Max.L.Ra    : num  10 9 10 9 52 6 6 9 7 11 ...
##  $ Scat.Ra     : num  162 149 207 144 149 255 153 137 122 183 ...
##  $ Elong       : num  42 45 32 46 45 26 42 48 54 36 ...
##  $ Pr.Axis.Rect: num  20 19 23 19 19 28 19 18 17 22 ...
##  $ Max.L.Rect  : num  159 143 158 143 144 169 143 146 127 146 ...
##  $ Sc.Var.Maxis: num  176 170 223 160 241 280 176 162 141 202 ...
##  $ Sc.Var.maxis: num  379 330 635 309 325 957 361 281 223 505 ...
##  $ Ra.Gyr      : num  184 158 220 127 188 264 172 164 112 152 ...
##  $ Skew.Maxis  : num  70 72 73 63 127 85 66 67 64 64 ...
##  $ Skew.maxis  : num  6 9 14 6 9 5 13 3 2 4 ...
##  $ Kurt.maxis  : num  16 14 9 10 11 9 1 3 14 14 ...
##  $ Kurt.Maxis  : num  187 189 188 199 180 181 200 193 200 195 ...
##  $ Holl.Ra     : num  197 199 196 207 183 183 204 202 208 204 ...
##  $ Class       : Factor w/ 4 levels "bus","opel","saab",..: 4 4 3 4 1 1 1 4 4 3 ...
```

```
summary(df$Class)
```

```
##  bus opel saab  van
##  218  212  217  199
```

```
df <- df %>%
    filter(Class != 'opel')
df$Class <- factor(df$Class)
str(df)
```

```
## 'data.frame':    634 obs. of  19 variables:
##  $ Comp         : num  95 91 104 93 85 107 97 90 86 93 ...
##  $ Circ         : num  48 41 50 41 44 57 43 43 34 44 ...
##  $ D.Circ       : num  83 84 106 82 70 106 73 66 62 98 ...
##  $ Rad.Ra       : num  178 141 209 159 205 172 173 157 140 197 ...
##  $ Pr.Axis.Ra   : num  72 57 66 63 103 50 65 65 61 62 ...
##  $ Max.L.Ra     : num  10 9 10 9 52 6 6 9 7 11 ...
##  $ Scat.Ra      : num  162 149 207 144 149 255 153 137 122 183 ...
##  $ Elong        : num  42 45 32 46 45 26 42 48 54 36 ...
##  $ Pr.Axis.Rect: num  20 19 23 19 19 28 19 18 17 22 ...
##  $ Max.L.Rect   : num  159 143 158 143 144 169 143 146 127 146 ...
##  $ Sc.Var.Maxis: num  176 170 223 160 241 280 176 162 141 202 ...
##  $ Sc.Var.maxis: num  379 330 635 309 325 957 361 281 223 505 ...
##  $ Ra.Gyr       : num  184 158 220 127 188 264 172 164 112 152 ...
##  $ Skew.Maxis   : num  70 72 73 63 127 85 66 67 64 64 ...
##  $ Skew.maxis   : num  6 9 14 6 9 5 13 3 2 4 ...
##  $ Kurt.maxis   : num  16 14 9 10 11 9 1 3 14 14 ...
##  $ Kurt.Maxis   : num  187 189 188 199 180 181 200 193 200 195 ...
##  $ Holl.Ra      : num  197 199 196 207 183 183 204 202 208 204 ...
##  $ Class        : Factor w/ 3 levels "bus","saab","van": 3 3 2 3 1 1 1 3 3 2 ...
```

```
summary(df$Class)
```

```
##  bus saab  van
##  218  217  199
```

Bagging Trees

```
set.seed(1)

train <- sample(1:nrow(df), nrow(df)/2)
bag <- randomForest(Class ~ ., data = df,
                    subset = train,
                    mtry = 18,
                    importance = T)
bag
```

```
##
## Call:
##  randomForest(formula = Class ~ ., data = df, mtry = 18, importance = T,      subset = train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 18
##
##         OOB estimate of  error rate: 6.62%
## Confusion matrix:
##      bus saab van class.error
## bus  107    2   1  0.02727273
## saab   7   95  10  0.15178571
## van    0    1  94  0.01052632
```

```r
set.seed(1)

bag <- randomForest(Class ~ ., data = df,
                    subset = train,
                    ntree = 25,
                    mtry = 18,
                    importance = T)
bag
```

```
##
## Call:
##  randomForest(formula = Class ~ ., data = df, ntree = 25, mtry = 18,      importance = T, subset = t
##                Type of random forest: classification
##                      Number of trees: 25
## No. of variables tried at each split: 18
##
##         OOB estimate of  error rate: 8.2%
## Confusion matrix:
##      bus saab van class.error
## bus  106    2   2  0.03636364
## saab   9   94   9  0.16071429
## van    2    2  91  0.04210526
```

Random Forest

```r
set.seed(1)

rf <- randomForest(Class ~ .,
                   data = df,
                   subset = train,
                   mtry = 9,
                   importance = T)
rf
```

```
##
## Call:
##  randomForest(formula = Class ~ ., data = df, mtry = 9, importance = T,      subset = train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 9
##
##         OOB estimate of  error rate: 6.62%
## Confusion matrix:
##      bus saab van class.error
## bus  108    1   1  0.01818182
## saab   7   95  10  0.15178571
## van    1    1  93  0.02105263
```

```r
sample_size <- ceiling(.632*nrow(df[-train,]))
vars <- floor(sqrt(ncol(df)))
sample_size
```

```
## [1] 201
```

```
vars
```

```
## [1] 4
```

```
set.seed(1)
rf <- randomForest(Class ~ .,
                   data = df,
                   subset = train,
                   mtry = vars,
                   sampsize = sample_size,
                   importance = T)
rf
```

```
##
## Call:
##  randomForest(formula = Class ~ ., data = df, mtry = vars, sampsize = sample_size,      importance =
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 6.31%
## Confusion matrix:
##      bus saab van class.error
## bus  108    0   2  0.01818182
## saab   8   95   9  0.15178571
## van    1    0  94  0.01052632
```

```
# the smallest error rate for now
```

```
# test
est_class <- predict(rf, newdata = df[-train,])
mean(est_class != df$Class[-train])
```

```
## [1] 0.1041009
```

```
# error rate for the test data is larger maybe because of overfitting,
#                                        maybe because of unforfunate choice of test/train data
```

```
set.seed(1)

ntrees <- 500
rf <- randomForest(Class ~ .,
                   data = df,
                   subset = train,
                   ntree = ntrees,
                   mtry = vars,
                   sampsize = sample_size,
                   importance = T,
                   do.trace = ntrees/10)
```

```
## ntree      OOB      1      2      3
##    50:   7.57%  2.73% 17.86%  1.05%
##   100:   6.94%  1.82% 16.07%  2.11%
##   150:   6.31%  1.82% 15.18%  1.05%
##   200:   6.62%  1.82% 16.07%  1.05%
##   250:   5.99%  1.82% 14.29%  1.05%
##   300:   6.31%  1.82% 15.18%  1.05%
##   350:   6.62%  1.82% 16.07%  1.05%
##   400:   6.31%  1.82% 15.18%  1.05%
##   450:   6.31%  1.82% 15.18%  1.05%
##   500:   6.31%  1.82% 15.18%  1.05%
```

```
rf
```

```
## 
## Call:
##  randomForest(formula = Class ~ ., data = df, ntree = ntrees,      mtry = vars, sampsize = sample_si
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
## 
##         OOB estimate of  error rate: 6.31%
## Confusion matrix:
##      bus saab van class.error
## bus  108    0   2  0.01818182
## saab   8   95   9  0.15178571
## van    1    0  94  0.01052632
```

```
set.seed(1)

ntrees <- 300
rf <- randomForest(Class ~ .,
                   data = df,
                   subset = train,
                   ntree = ntrees,
                   mtry = vars,
                   sampsize = sample_size,
                   importance = T,
                   do.trace = ntrees/30)
```

```
## ntree      OOB      1      2      3
##    10:   9.78%  6.36% 19.64%  2.11%
##    20:   7.57%  4.55% 15.18%  2.11%
##    30:   7.57%  3.64% 16.96%  1.05%
##    40:   6.62%  2.73% 15.18%  1.05%
##    50:   7.57%  2.73% 17.86%  1.05%
##    60:   6.31%  1.82% 15.18%  1.05%
##    70:   6.62%  1.82% 16.07%  1.05%
##    80:   6.94%  1.82% 16.07%  2.11%
##    90:   6.94%  1.82% 16.07%  2.11%
##   100:   6.94%  1.82% 16.07%  2.11%
##   110:   6.62%  1.82% 16.07%  1.05%
##   120:   6.62%  1.82% 16.07%  1.05%
##   130:   6.62%  1.82% 16.07%  1.05%
##   140:   6.62%  1.82% 16.07%  1.05%
##   150:   6.31%  1.82% 15.18%  1.05%
##   160:   6.31%  1.82% 15.18%  1.05%
##   170:   6.31%  1.82% 15.18%  1.05%
##   180:   6.31%  1.82% 15.18%  1.05%
##   190:   6.62%  1.82% 16.07%  1.05%
##   200:   6.62%  1.82% 16.07%  1.05%
##   210:   6.31%  1.82% 15.18%  1.05%
##   220:   6.62%  1.82% 15.18%  2.11%
##   230:   5.99%  1.82% 14.29%  1.05%
##   240:   5.99%  1.82% 14.29%  1.05%
##   250:   5.99%  1.82% 14.29%  1.05%
##   260:   6.31%  1.82% 15.18%  1.05%
##   270:   6.31%  1.82% 15.18%  1.05%
##   280:   6.31%  1.82% 15.18%  1.05%
##   290:   6.31%  1.82% 15.18%  1.05%
##   300:   6.31%  1.82% 15.18%  1.05%
```

```
rf
```

```
##
## Call:
##  randomForest(formula = Class ~ ., data = df, ntree = ntrees,      mtry = vars, sampsize = sample_si
##                Type of random forest: classification
##                      Number of trees: 300
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 6.31%
## Confusion matrix:
##      bus saab van class.error
## bus  108    0   2  0.01818182
## saab   8   95   9  0.15178571
## van    1    0  94  0.01052632
```

```
set.seed(1)

ntrees <- 230
rf <- randomForest(Class ~ .,
                   data = df,
                   subset = train,
                   ntree = ntrees,
                   mtry = vars,
                   sampsize = sample_size,
                   importance = T)
rf
```

```
##
## Call:
##  randomForest(formula = Class ~ ., data = df, ntree = ntrees,      mtry = vars, sampsize = sample_si
##                Type of random forest: classification
##                      Number of trees: 230
## No. of variables tried at each split: 4
```

```
##
##         OOB estimate of  error rate: 5.99%
## Confusion matrix:
##      bus saab van class.error
## bus  108    0   2  0.01818182
## saab   7   96   9  0.14285714
## van    1    0  94  0.01052632
```

```
est_class <- predict(rf, newdata = df[-train,])
mean(est_class != df$Class[-train])
```

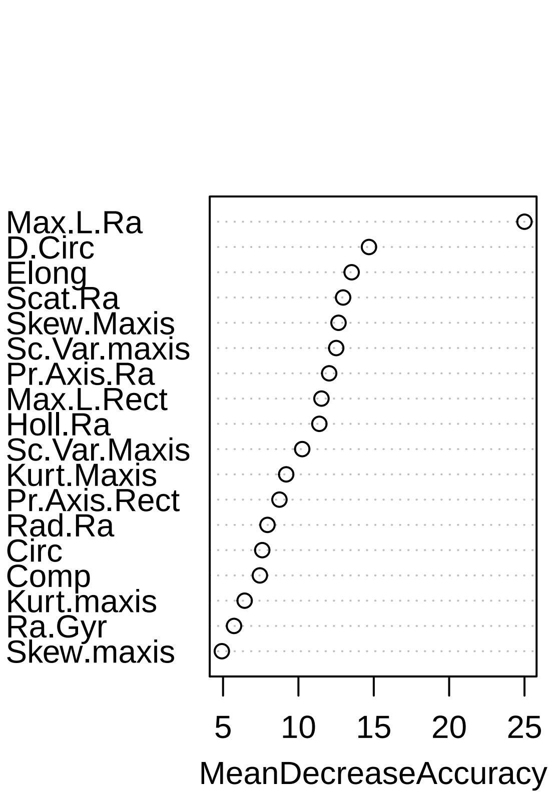
```
## [1] 0.09463722
```

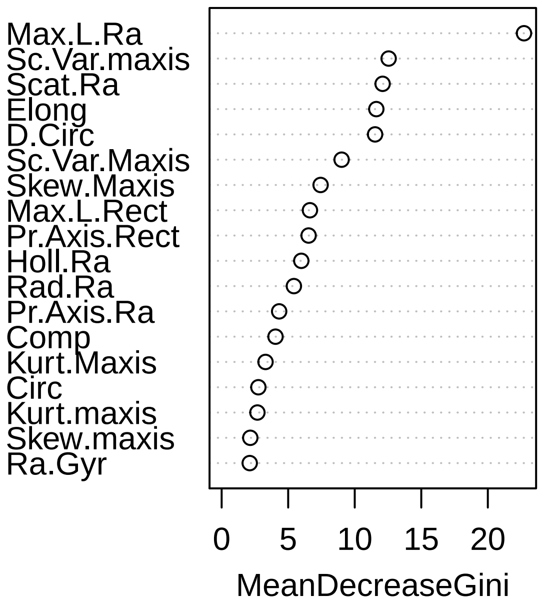
```
importance(rf)
```

```
##                    bus       saab         van MeanDecreaseAccuracy
## Comp          6.059377  4.7009247  4.85796831             7.442709
## Circ          6.034482  5.6443849  2.27603397             7.600410
## D.Circ       12.476425  8.8535369  8.96415111            14.680438
## Rad.Ra        6.169519  5.8866048  6.68848695             7.947228
## Pr.Axis.Ra   10.621176  7.1856596  5.67786171            12.040802
## Max.L.Ra     21.878787 17.7790224 18.23877766            24.998139
## Scat.Ra      10.756709  7.1122217 13.16633136            12.964160
## Elong        10.916012  7.9774177 14.64730755            13.531771
## Pr.Axis.Rect  6.538649  3.9292392  8.81201743             8.740163
## Max.L.Rect    9.088398  9.9761215  5.57174737            11.528983
## Sc.Var.Maxis  9.979156  6.6408687  8.67784453            10.251777
## Sc.Var.maxis  9.847864  7.7914692 12.58708627            12.508193
## Ra.Gyr        3.728664  3.7132889  2.62058044             5.727017
## Skew.Maxis    9.573700 10.5193031  5.10122319            12.660068
## Skew.maxis    5.870879 -0.6134069  0.02370082             4.918691
## Kurt.maxis    3.675142  5.0230821  3.14545219             6.431313
## Kurt.Maxis    8.566836  4.8730573  4.60439629             9.189796
## Holl.Ra       8.172886  8.7092794  6.33963654            11.388951
##              MeanDecreaseGini
## Comp                 4.047486
## Circ                 2.760725
## D.Circ              11.529290
## Rad.Ra               5.428790
## Pr.Axis.Ra           4.322950
## Max.L.Ra            22.718260
## Scat.Ra             12.094844
## Elong               11.622053
## Pr.Axis.Rect         6.534919
## Max.L.Rect           6.633753
## Sc.Var.Maxis         9.015377
## Sc.Var.maxis        12.547583
## Ra.Gyr               2.111020
## Skew.Maxis           7.435177
## Skew.maxis           2.152614
## Kurt.maxis           2.687833
## Kurt.Maxis           3.296486
## Holl.Ra              5.984613
```

```
# Max.L.Ra - max length rectangularity
```

```
varImpPlot(rf)
```


rf
Max.L.Ra
D.Circ
Elong
Scat.Ra
Skew.Maxis
Sc.Var.maxis
Pr.Axis.Ra
Max.L.Rect
Holl.Ra
Sc.Var.Maxis
Kurt.Maxis
Pr.Axis.Rect
Rad.Ra
Circ
Comp
Kurt.maxis
Ra.Gyr
Skew.maxis
MeanDecreaseAccuracy


Max.L.Ra
Sc.Var.maxis
Scat.Ra
Elong
D.Circ
Sc.Var.Maxis
Skew.Maxis
Max.L.Rect
Pr.Axis.Rect
Holl.Ra
Rad.Ra
Pr.Axis.Ra
Comp
Kurt.Maxis
Circ
Kurt.maxis
Skew.maxis
Ra.Gyr
MeanDecreaseGini