Seeds

Marat Habibullin

26.12.2014

Load packages:

```
## Loading required package: RColorBrewer
## Loading required package: gplots
##
## Attaching package: 'gplots'
##
## The following object is masked from 'package:stats':
##
##     lowess
##
## Loading required package: ggplot2
##
## Attaching package: 'ggplot2'
##
## The following object is masked from 'package:latticeExtra':
##
##     layer
```

Read data:

```
seeds = read.table("data/seeds_dataset.txt", comment.char = "#")
names(seeds) = c("area", "perimeter", "compactness", "length", "width", "assymetry", "groove", "stype")
seeds$stype = factor(seeds$stype)
```

Let's use multinomial regression:

```
mln <- multinom(stype ~ ., data = seeds, trace = FALSE)

tn.mln <- tune(multinom, stype ~ .,
               data = seeds,
               trace = FALSE,
               tunecontrol = tune.control(sampling = "cross"))
tn.mln$performances
```

```
##   dummyparameter      error dispersion
## 1              0 0.02857143 0.04015591
```

Pretty good. Let's try to simplify the model using stepAIC:

```
mln_aic = stepAIC(mln)
```

```
## Start:  AIC=49.87
## stype ~ area + perimeter + compactness + length + width + assymetry +
```

```
##     groove
##
##               Df    AIC
## - compactness  2 46.790
## - perimeter    2 47.376
## - width        2 48.313
## <none>           49.875
## - area         2 49.908
## - assymetry    2 61.847
## - length       2 72.274
## - groove       2 97.356
##
## Step:  AIC=46.79
## stype ~ area + perimeter + length + width + assymetry + groove
##
##             Df    AIC
## - perimeter  2 43.738
## - width      2 44.104
## - area       2 46.656
## <none>         46.790
## - assymetry  2 59.129
## - length     2 70.016
## - groove     2 94.284
##
## Step:  AIC=43.74
## stype ~ area + length + width + assymetry + groove
##
##             Df    AIC
## - width      2 41.871
## <none>         43.738
## - area       2 45.876
## - assymetry  2 57.362
## - length     2 64.133
## - groove     2 92.839
##
## Step:  AIC=41.87
## stype ~ area + length + assymetry + groove
##
##             Df    AIC
## <none>         41.871
## - assymetry  2 55.804
## - area       2 66.380
## - length     2 70.802
## - groove     2 95.127
```

```
tn.mln.aic <- tune(multinom,
                   mln_aic$call$formula,
                   data = seeds,
                   trace = FALSE,
                   tunecontrol = tune.control(sampling = "cross"))
tn.mln.aic$performances
```

```
##   dummyparameter      error dispersion
## 1              0 0.02857143 0.02459037
```

The error is the same, so it's a good idea to take the second one. But first let's leave one out approach:

```
tn.full.loo <- tune(multinom, stype ~ .,
     data = seeds,
     trace = FALSE,
     tunecontrol = tune.control(sampling = "cross", cross = nrow(seeds)))
tn.full.loo$performances
```

```
##   dummyparameter      error dispersion
## 1              0 0.04285714  0.2030189
```

```
tn.aic.loo <- tune(multinom,
     mln_aic$call$formula,
     data = seeds,
     trace = FALSE,
     tunecontrol = tune.control(sampling = "cross", cross = nrow(seeds)))
tn.aic.loo$performances
```

```
##   dummyparameter      error dispersion
## 1              0 0.02857143  0.1669967
```

Definitely should take the second model. Let's check the results with test-train:

```
train = sample(1 : nrow(seeds), 0.7 * nrow(seeds))

mln = multinom(mln_aic$call$formula, data = seeds[train, ], trace = FALSE)

tbl = table(predicted = predict(mln, seeds[-train, ]), actual = seeds[-train,]$stype)
tbl
```

```
##          actual
## predicted  1  2  3
##         1 14  1  0
##         2  1 24  0
##         3  4  0 19
```

```
chisq.test(tbl)
```

```
## Warning in chisq.test(tbl): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 93.0379, df = 4, p-value < 2.2e-16
```

Let's also check what lda and qda say:

```
tune(lda, stype ~ ., data = seeds, predict.func = simple.predict.da)
```

```
##
## Error estimation of 'lda' using 10-fold cross validation: 0.03809524
```

```
tune(lda, mln_aic$call$formula, data = seeds, predict.func = simple.predict.da)
```

```
##
## Error estimation of 'lda' using 10-fold cross validation: 0.02857143
```

```
tune(qda, stype ~ ., data = seeds, predict.func = simple.predict.da)
```

```
##
## Error estimation of 'qda' using 10-fold cross validation: 0.05714286
```

```
tune(qda, mln_aic$call$formula, data = seeds, predict.func = simple.predict.da)
```

```
##
## Error estimation of 'qda' using 10-fold cross validation: 0.02857143
```