Ensemble Learning

Business Analytics
Stefan Feuerriegel

Outline

Setup

- Accessing credit scores

```
library(caret)
data(GermanCredit)
```

- Split data into index subset for training (20 %) and testing (80 %) instances

```
inTrain <- runif(nrow(GermanCredit)) < 0.2
```

Outline

Decision Trees in R

- Loading required libraries `rpart`, `party` and `partykit`

```r
library(rpart)
library(party)
library(partykit)
```

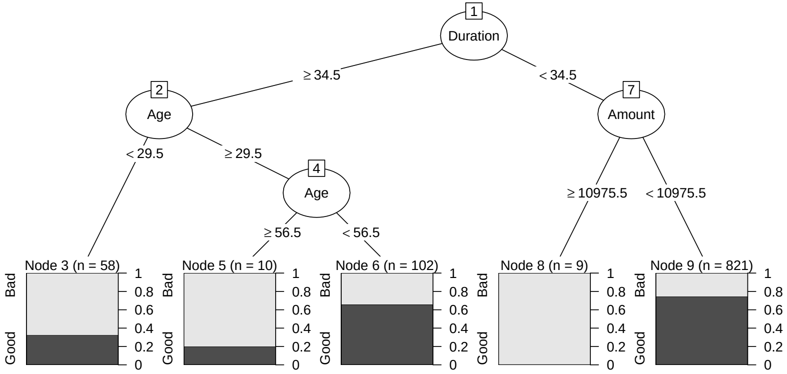
- Building a decision tree with
  `rpart(formula, method="class", data=d)`

```r
dt <- rpart(Class ~ Duration + Amount + Age,
            method="class", data=GermanCredit)
```

Decision Trees in R

- Plot decision tree using `plot(dt)`

```
plot(as.party(dt))
```



Prediction with Decision Trees

- `predict(dt, test, type="class")` predicts classes on new data `test`

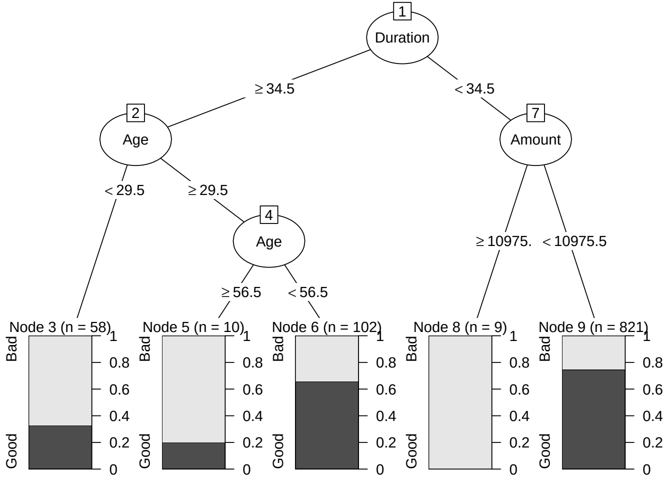
```
pred <- predict(dt, GermanCredit, type="class")
pred[1:5]
```

```
##    1    2    3    4    5
## Good  Bad Good Good Good
## Levels: Bad Good
```

- Output: predicted label in 1st row out of all possible labels (2nd row)
- Pruning occurs through `prune(dt, cp = ...)` with a given complexity parameter
  - Usual heuristic: `dt$cptable[which.min(dt$cptable[, "xerror"]), "CP"]`

Pruning Decision Trees

```r
p <- prune(dt, cp = dt$cptable[which.min(dt$cptable[, "xerror"]), "CP"])
plot(as.party(p))
```



Outline

Random Forests in R

- Load required library `randomForest`

```r
library(randomForest)
```

- Learn random forest on training data with `randomForest(...)`

```r
rf <- randomForest(Class ~ .,
                   data=GermanCredit,
                   ntree=100)
```

- Options to control behavior
  - `ntree` controls the number of trees (default: 500)
  - `mtry` gives number of variables to choose from at each node
  - `na.action` specifies how to handle missing values
  - `importance=TRUE` calculates variable importance metric
- Predict credit scores for testing instances

```r
pred <- predict(rf, newdata=GermanCredit)
```

Variable Importance in R

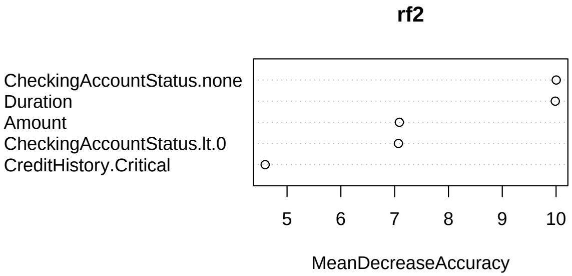
- Learn random forest and enable the calculation of variable importance metrics via `importance=TRUE`

```r
rf2 <- randomForest(Class ~ .,
                    data=GermanCredit, #with full dataset
                    ntree=100,
                    importance=TRUE)
```

Variable Importance in R

- Plot variable importance via `varImpPlot(rf, ...)`

```
varImpPlot(rf2, type=1, n.var=5)
```



- `type` choose the importance metric ($=1$ is the mean decrease in accuracy if the variable would be randomly permuted)
- `n.var` denotes number of variables

Outline

Boosting in R

- Load the required packages `mboost`

```
library(mboost)
```

- Fit a generalized linear model via `glmboost(...)`

```
m.boost <- glmboost(Class ~ Amount + Duration
                        + Personal.Female.Single,
                    family=Binomial(), # needed for classification
                    data=GermanCredit)
coef(m.boost)

##   (Intercept)        Amount      Duration
##  4.104949e-01 -1.144369e-05 -1.703911e-02
## attr(,"offset")
## [1] 0.4236489
```

- Different from the normal `glm(...)` routine, the boosted version inherently performs variable selection

Outline

AdaBoost in R

- Load required package `ada`

```
library(ada)
```

- Fit AdaBoost model on training data with `ada(..., iter)` given a fixed number `iter` of iterations

```
m.ada <- ada(Class ~ .,
             data=GermanCredit,
             iter=50)
```

- Evaluate on test data

```
pred <- predict(m.ada, newdata=GermanCredit)
```