

Introduction to Machine Learning in R

NPFL 054

Easy homework assigned on March 22, 2023

Contact teacher: Martin Holub
holub@ufal.mff.cuni.cz

In all tasks specified below you will work with a part of *Forbes2000* data set, which comes from *HSAUR* package. You should prepare your data using the following function:

```
prepare_data = function(){
  library(HSAUR)                # load the library with Forbes data set
  F = Forbes2000                # just to make a copy

  F = F[!is.na(F$profits), ]    # rows with NA values are removed

  # now we select only countries with at least 25 companies in the data
  selected.countries = names(table(F$country)[table(F$country) >= 25])
  F = F[F$country %in% selected.countries, ]
  F$country = droplevels(F$country)

  cat(nrow(F), "observations selected from Forbes2000 data set.\n")
  cat("Selected countries: ",
      paste(selected.countries, collapse=", "), ".\n", sep="")

  # to randomly split the data into two disjoint subsets
  set.seed(123); s = sample(1710)
  forbes.train <- F[s[1:1200], ]    # training examples
  forbes.test  <- F[s[1201:1710], ] # test examples
}
```

When you run `prepare_data()`, you will get two data frames, namely `forbes.train` and `forbes.test` with the same structure:

```
> str(forbes.train)
'data.frame':   1200 obs. of  8 variables:
 $ rank       : int  555 1568 795 1762 1873 81 1026 1774 1076 882 ...
 $ name       : chr   "KeySpan" "M6-Metropole Television" "Zions Bancorp" "Buderus" ...
 $ country    : Factor w/ 16 levels "Australia","Canada",...: 16 4 16 5 5 3 8 15 16 12 ...
 $ category   : Factor w/ 27 levels "Aerospace & defense",...: 27 18 2 7 9 19 18 27 2 3 ...
 $ sales      : num  6.85 1.48 1.89 1.95 0.42 ...
 $ profits    : num  0.4 0.17 0.34 0.25 0.16 1.94 0.16 -0.1 0.23 0.17 ...
 $ assets     : num  13 1.2 28.56 1.51 2.14 ...
 $ marketvalue: num  5.79 4.37 5.25 2.46 3.01 ...
```

Variable `profits` will be considered as an output attribute. Look at its distribution. Is it similarly distributed in the training and the test set?

Task 1 – Evaluation and analysis of Decision Trees

Load Forbes2000 data set and build a decision tree (DT) as indicated in *Tutorial on Decision Trees*, Parts I and II. Transform output attribute profits to a binary variable

```
> forbes.train$profits = factor(forbes.train$profits > 0.2)
```

Use the DT model to predict profits on the training data and determine the training error. Then compare it to the test error.

2a)

Do some experiments with the proportion of the training and test data size. Does the model improve when you use bigger data set for training? Would it be enough to use smaller portion of the data to train the model? Try different sizes of training and test data sets and for each size make 10 random samples. Report on your results and its variance.

2b)

Which of the used features are most important? Experiment with different subsets of the given features. Report on differences in the performance of resulting DT models.

2c)

Compare discrete features country and category in training data set `forbes.train`. Use conditional entropy and compare how the two features can contribute to the prediction of binary profits.

Task 2 – Evaluation of classification Random Forests (RF)

Work with the above mentioned data frames `forbes.train` and `forbes.test`. First transform output attribute profits to a binary variable in both training and test data sets

```
> forbes.train$profits = factor(forbes.train$profits > 0.2)
> forbes.test$profits = factor(forbes.test$profits > 0.2)
```

Build and evaluate RF models to predict binary profits using R package `randomForest`. There are 5 features that you can use for prediction: `category`, `sales`, `assets`, `marketvalue`, `country`.

- Learn 20 random forests with different number of trees using `n tree` in `seq(100, 2000, 100)`.
- For each random forest compute error rate estimate using 6-fold cross validation. In each cross validation run you will have 1000 examples for training and 200 examples for test. Compute mean and standard deviation of the error rate.
- Then compare the cross validation results with the OOB error estimates, and also with test error rates measured using `forbes.test`.
- Arrange all your results in a nice table and plot a chart.

Task 3 – Regression Random Forest (RF)

Work with the above mentioned data frames `forbes.train` and `forbes.test`. Build and evaluate a regression RF model to predict profits using R package `randomForest`. There are 5 features that you can use for prediction: `category`, `sales`, `assets`, `marketvalue`, `country`. For differences between classification and regression RF, see `help(randomForest)`.

- During the development process use only you development data in `forbes.train`. Choose a good value of `n tree` parameter and estimate the generalization error using cross validation. Report on your work.
- Only when you finish whole development, take the test set and evaluate your model. Compare the result with the error estimated during the development.
- Develop also a single decision tree and compare its performance to your random forest.