Exercise 11

Business Analytics and Data Science WS16/17

Introduction

You often hear the complaint that advanced machine learning models are "black box" models, because it is supposedly not possible for us to look into and interpret the process by which the model comes to its results. While established or simple models like regression coefficients or decision trees are easier to interpret even without additional tools, both model-dependent and -independent tools have been developed that allow us to peek into complex models.
We will use some of these to answer the important questions which variables are important and what is the size and direction of the influence of the variables.

Variable importance

For both the random forest and the gradient boosting model, we can calculate which variables have the largest influence on the prediction. This *variable importance* is often model-based, i.e. caluclated in a specific way for a certain model. Two measures of variable importance, one for all tree-based models and one specific to random forests, will be discussed in detail in the lecture. For other models or approaches that are not model dependent, see the caret page on variable importance{http://topepo.github.io/caret/variable-importance.html} or the recommended literature.
- *Tree-based Gini importance*: The mean squared relative importance of each variable is the sum of squared improvement in the error risk over all internal nodes for which it was chosen as the splitting variable, averaged over all trees.
- *Random forest OOB importance*: The decrease in accuracy when randomly permuting the values of each variable in turn for each tree, averaged over all trees. The test sample for each tree are the observations not contained in the bootstrap training set for that tree a.k.a. out-of-bag observations.

Note: These measures do not capture the effect on prediction in case a variable were not available, because other variables could be used as surrogates.
Note: The importance of highly correlated variables will not be accurate. Expect RF to split the importance between correlated features and boosting to focus on one of them.

1. Train a random forest model and gradient boosted trees on the training data for the optimal parameters determined in the previous exercises. For random forest, set **importance = TRUE** to calculate the performance on the out-of-bag samples.
2. Calculate the variable importance of the random forest and the gradient boosting model using the package specific importance functions or mlr's **getFeatureImportance()**. How are the respective importance values calculated for the random forest and gradient boosting model?
3. Sort the variable importance for both models and remember the most important variables. Does the importance order of the variables fit your expectation?

```
## FeatureImportance:
## Task: tr
##
## Learner: classif.randomForest
## Measure: NA
## Contrast: NA
## Aggregation: function (x)  x
## Replace: NA
## Number of Monte-Carlo iterations: NA
## Local: FALSE
```

```
##        YOB    nKIDS      nDEP     PHON  dINC_SP   dINC_A    dHVAL     dMBO
## 1 8.488066 2.404636 0.5862555 1.085736 2.967138 9.256003 4.447245 3.960281
##      dOUTM    dOUTL   dOUTHP  dOUTCC YOB_missing  EMPS_A.B  EMPS_A.E
## 1 5.393078 3.373984 1.528043 1.92953    0.181963 0.3963593 0.9731003
##    EMPS_A.M EMPS_A.N EMPS_A.P EMPS_A.R  EMPS_A.T  EMPS_A.U EMPS_A.V
## 1 0.2523003 0.102761 1.338444 1.155427 0.6515614 0.1486272 1.049642
##    EMPS_A.W  EMPS_A.Z     RES.F    RES.N    RES.O    RES.P     RES.U
## 1 0.5816859 0.2585872 0.6927927 1.359245 0.926943 0.791234 0.8105294

## [1] "randomForest.formula" "randomForest"

##                   good        bad MeanDecreaseAccuracy MeanDecreaseGini
## YOB         14.4327083  0.6427236           17.9522084        8.4880656
## nKIDS       10.3046081 -5.9197254            9.1586564        2.4046359
## nDEP         0.7588072 -2.0576349           -0.3869245        0.5862555
## PHON         3.8211491  3.5612856            5.3964344        1.0857360
## dINC_SP     11.6649091 -5.6041456           10.0187850        2.9671380
## dINC_A      11.2346310 18.6840209           23.8842844        9.2560027
## dHVAL        4.6805995  0.8854696            6.3952155        4.4472450
## dMBO         9.0677367  0.7710914           11.7429837        3.9602813
## dOUTM        8.0225889 -2.3761775            8.2225744        5.3930781
## dOUTL       -0.1907716  7.0169240            4.3970178        3.3739838
## dOUTHP      -3.9638231  4.1734266           -1.6305591        1.5280432
## dOUTCC      -3.2505239  6.6469766            1.0195760        1.9295296
## YOB_missing  0.5592806 -1.9797422           -1.3178060        0.1819630
## EMPS_A.B    -4.3307627  0.1940057           -4.0385802        0.3963593
## EMPS_A.E    -0.2077683  1.3549396            0.6365132        0.9731003
## EMPS_A.M    -0.3134993 -3.1975211           -1.8574979        0.2523003
## EMPS_A.N    -2.6799376 -1.1210346           -2.7242338        0.1027610
## EMPS_A.P     0.6600805  1.0240267            1.4957432        1.3384438
## EMPS_A.R     2.5795869  5.2183566            8.1954443        1.1554268
## EMPS_A.T     9.7879753 -8.2119864            9.4697946        0.6515614
## EMPS_A.U    -2.3965313 -2.3155409           -3.1527420        0.1486272
## EMPS_A.V     3.0335064  1.4985552            3.9321287        1.0496424
## EMPS_A.W     6.8936937  1.0937984            7.2868829        0.5816859
## EMPS_A.Z     2.0341801 -2.8219218           -0.4918423        0.2585872
## RES.F       -1.4682379 -2.6254981           -3.0351310        0.6927927
## RES.N        2.7111085  9.7110807            9.5012680        1.3592446
## RES.O        6.0290270 -3.5035361            6.3636564        0.9269430
## RES.P        6.3900997 -5.3802893            5.8887789        0.7912340
## RES.U        0.7777034 -2.0188804           -0.2339993        0.8105294

##                   good        bad MeanDecreaseAccuracy MeanDecreaseGini
## YOB         17.7258718 -1.4941669           21.1069374        8.6967704
## nKIDS       11.3987576 -6.7656879            9.7170702        2.4507753
## nDEP        -2.1206532 -2.3987676           -3.1666183        0.5782920
## PHON         3.8684249  4.4854200            6.0452046        1.1453039
## dINC_SP     10.9020412 -6.5093898            8.8274307        2.9735335
## dINC_A       8.8001313 19.0608847           20.3301621        9.0805768
## dHVAL        6.1186499 -2.5866815            6.5180634        4.4636854
## dMBO        10.0638695 -0.7194739           12.3125984        3.9637976
## dOUTM        8.7876811 -2.9791368            8.5150795        5.5323451
## dOUTL        0.6054358  5.6490191            3.9914026        3.5191151
## dOUTHP      -2.2794873  3.5238952           -0.2985438        1.4809514
## dOUTCC      -2.4311607  5.0116142            0.5633846        1.8591911
```

```
## YOB_missing  0.3182870 -1.1306504          -0.6786636        0.1833497
## EMPS_A.B    -4.7213719  1.3740397          -3.8805780        0.4307070
## EMPS_A.E    -2.0870121  2.3264081          -0.6253402        1.0133077
## EMPS_A.M    -1.3617297 -2.4529821          -2.4769077        0.2489064
## EMPS_A.N    -2.3104722 -2.0913907          -2.7538880        0.1024786
## EMPS_A.P    -1.4310729  1.6254098          -0.6704039        1.3520756
## EMPS_A.R     3.3948515  4.3787463           8.8616120        1.1273389
## EMPS_A.T     8.2046040 -6.8506765           7.8359190        0.6759480
## EMPS_A.U    -3.0165283 -3.2329889          -4.3976916        0.1433446
## EMPS_A.V     3.1064790  1.2959541           3.9496290        1.0375792
## EMPS_A.W     6.6154166  4.7869367           9.4385551        0.5747231
## EMPS_A.Z     1.5246462 -0.9656538           0.5667488        0.2930259
## RES.F       -1.5205436 -3.4557026          -3.7306668        0.6644564
## RES.N        3.4381707 11.5845247          10.9506406        1.2959994
## RES.O        8.5137511 -6.0826541           8.4894867        1.0404929
## RES.P        5.9919631 -5.1913801           5.3258041        0.7861295
## RES.U        3.0015985 -0.6316972           3.0931129        0.7659305
```

```
##         Feature         Gain       Cover    Frequency
##  1:      dINC_A 0.2299245904 0.205618561 0.2037914692
##  2:         YOB 0.1659004170 0.166629729 0.1744075829
##  3:       dOUTM 0.1175878011 0.104295897 0.1336492891
##  4:       dHVAL 0.0944318171 0.079595670 0.0947867299
##  5:        dMBO 0.0879245751 0.081750083 0.0853080569
##  6:       dOUTL 0.0780987240 0.069021041 0.0748815166
##  7:     dINC_SP 0.0617081038 0.070625947 0.0616113744
##  8:       nKIDS 0.0375661316 0.038721694 0.0350710900
##  9:      dOUTHP 0.0332408160 0.057053480 0.0322274882
## 10:      dOUTCC 0.0324765877 0.057969687 0.0369668246
## 11:        PHON 0.0143042729 0.015077932 0.0151658768
## 12:    EMPS_A.N 0.0113034954 0.004231920 0.0132701422
## 13:    EMPS_A.R 0.0093763583 0.010086586 0.0104265403
## 14:    EMPS_A.P 0.0062183415 0.006108448 0.0047393365
## 15:    EMPS_A.U 0.0041640287 0.005849441 0.0056872038
## 16:    EMPS_A.B 0.0039621507 0.005907199 0.0047393365
## 17:        nDEP 0.0038603924 0.004426010 0.0037914692
## 18:    EMPS_A.V 0.0037884493 0.008535142 0.0047393365
## 19: YOB_missing 0.0036235134 0.006620577 0.0037914692
## 20:    EMPS_A.E 0.0005394335 0.001874957 0.0009478673
```

```
## FeatureImportance:
## Task: tr
##
## Learner: classif.xgboost
## Measure: NA
## Contrast: NA
## Aggregation: function (x)  x
## Replace: NA
## Number of Monte-Carlo iterations: NA
## Local: FALSE
##         YOB      nKIDS        nDEP       PHON   dINC_SP    dINC_A
## 1 0.1659004 0.03756613 0.003860392 0.01430427 0.0617081 0.2299246
##        dHVAL       dMBO     dOUTM      dOUTL     dOUTHP     dOUTCC
## 1 0.09443182 0.08792458 0.1175878 0.07809872 0.03324082 0.03247659
##   YOB_missing    EMPS_A.B    EMPS_A.E     EMPS_A.M EMPS_A.N  EMPS_A.P
```

```
## 1           0 0.003623513 0.003962151 0.0005394335        0 0.0113035
##      EMPS_A.R    EMPS_A.T EMPS_A.U    EMPS_A.V    EMPS_A.W EMPS_A.Z RES.F
## 1 0.006218342 0.009376358        0 0.004164029 0.003788449        0     0
##   RES.N RES.O RES.P RES.U
## 1     0     0     0     0
```

```
##                      rf         xgb
## dINC_A      100.0000000 100.0000000
## YOB          91.6102172  72.1542732
## dOUTM        57.7971962  51.1418987
## dHVAL        47.4638834  41.0707775
## dMBO         42.1437614  38.2406140
## dOUTL        35.7384074  33.9671037
## dINC_SP      31.2935802  26.8384098
## nKIDS        25.1481932  16.3384575
## dOUTCC       19.9576138  14.1248866
## dOUTHP       15.5713378  14.4572688
## EMPS_A.P     13.4999479   4.9161751
## PHON         10.7390918   6.2212888
## EMPS_A.R     11.5004701   2.7045135
## RES.N        13.7271983   0.0000000
## EMPS_A.V     10.3447658   1.8110410
## EMPS_A.E      9.5085359   1.7232392
## EMPS_A.T      5.9956949   4.0780146
## RES.O         9.0042639   0.0000000
## RES.U         7.7324345   0.0000000
## RES.P         7.5216308   0.0000000
## nDEP          5.2822217   1.6789819
## EMPS_A.W      5.2322983   1.6476921
## RES.F         6.4461498   0.0000000
## EMPS_A.B      3.2075879   1.5759573
## EMPS_A.M      1.6337308   0.2346132
## EMPS_A.Z      1.7024153   0.0000000
## YOB_missing   0.8652888   0.0000000
## EMPS_A.U      0.5010924   0.0000000
## EMPS_A.N      0.0000000   0.0000000
```

Partial dependence plots

While variable importance answers our question as to which variables are relevant to the classification problem, they do not provide information on the direction of the effect. Since our predictor function, i.e. the random forest or boost model, depends on so many features, we cannot visualize its form in 2D or even 3D space, because we would need one axis for each feature and one axis for the target variable.
However, we can plot a collection of plots, each of which shows the relation of the target variable to *one* variable. This idea translates into *partial dependence plots* of the average prediction output for a value of variable S, averaged over the values for all other variables that occur in the data. Simply speaking, for each value of variable S that we would like to analyze (e.g. variable nKids = 1, 2,. . .), we set this value for the n observations in the data, and calculate the average prediction over all N. This computationally intensive procedure works for any (blackbox) model, but there's a trick to compute it from trees directly (see *weighted tree traversal*).

1. Use function **partial()** from {pdp} to calculate partial dependence plots for the most important variables.

2. Plot the partial dependence in terms of probability for the most important variables on a common scale [0;1].
3. Interpret the plots. Are they in line with your intuition on how the variables relate to the credit risk of the applicant?

```
## List of 30
##  $ coefficients     : Named num [1:30] -0.1267 -0.0137 -0.0375 0.0968 -0.1711 ...
##   ..- attr(*, "names")= chr [1:30] "(Intercept)" "YOB" "nKIDS" "nDEP" ...
##  $ residuals        : Named num [1:979] -2.2 -1.23 -1.38 -1.62 -1.29 ...
##   ..- attr(*, "names")= chr [1:979] "1" "2" "3" "4" ...
##  $ fitted.values    : Named num [1:979] 0.545 0.188 0.275 0.381 0.227 ...
##   ..- attr(*, "names")= chr [1:979] "1" "2" "3" "4" ...
##  $ effects          : Named num [1:979] 12.39 -3.56 -1.28 -0.25 1.44 ...
##   ..- attr(*, "names")= chr [1:979] "(Intercept)" "YOB" "nKIDS" "nDEP" ...
##  $ R                : num [1:30, 1:30] -13.1 0 0 0 0 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:30] "(Intercept)" "YOB" "nKIDS" "nDEP" ...
##   .. ..$ : chr [1:30] "(Intercept)" "YOB" "nKIDS" "nDEP" ...
##  $ rank             : int 28
##  $ qr               :List of 5
##   ..$ qr   : num [1:979, 1:30] -13.118 0.0298 0.0341 0.037 0.0319 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:979] "1" "2" "3" "4" ...
##   .. .. ..$ : chr [1:30] "(Intercept)" "YOB" "nKIDS" "nDEP" ...
##   ..$ rank : int 28
##   ..$ qraux: num [1:30] 1.04 1.01 1.02 1.01 1.02 ...
##   ..$ pivot: int [1:30] 1 2 3 4 5 6 7 8 9 10 ...
##   ..$ tol  : num 1e-11
##   ..- attr(*, "class")= chr "qr"
##  $ family           :List of 12
##   ..$ family    : chr "binomial"
##   ..$ link      : chr "logit"
##   ..$ linkfun   :function (mu)
##   ..$ linkinv   :function (eta)
##   ..$ variance  :function (mu)
##   ..$ dev.resids:function (y, mu, wt)
##   ..$ aic       :function (y, n, mu, wt, dev)
##   ..$ mu.eta    :function (eta)
##   ..$ initialize:  expression({  if (NCOL(y) == 1) {  if (is.factor(y))  y <- y != levels(y)[1L]  n
##   ..$ validmu   :function (mu)
##   ..$ valideta  :function (eta)
##   ..$ simulate  :function (object, nsim)
##   ..- attr(*, "class")= chr "family"
##  $ linear.predictors: Named num [1:979] 0.178 -1.463 -0.967 -0.486 -1.224 ...
##   ..- attr(*, "names")= chr [1:979] "1" "2" "3" "4" ...
##  $ deviance         : num 1038
##  $ aic              : num 1094
##  $ null.deviance    : num 1129
##  $ iter             : int 5
##  $ weights          : Named num [1:979] 0.248 0.153 0.2 0.236 0.176 ...
##   ..- attr(*, "names")= chr [1:979] "1" "2" "3" "4" ...
##  $ prior.weights    : Named num [1:979] 1 1 1 1 1 1 1 1 1 1 ...
##   ..- attr(*, "names")= chr [1:979] "1" "2" "3" "4" ...
##  $ df.residual      : int 951
```

```
##  $ df.null          : int 978
##  $ y                : Named num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..- attr(*, "names")= chr [1:979] "1" "2" "3" "4" ...
##  $ converged        : logi TRUE
##  $ boundary         : logi FALSE
##  $ model            :'data.frame':   979 obs. of  30 variables:
##   ..$ BAD        : Factor w/ 2 levels "good","bad": 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ YOB        : num [1:979] 19 41 66 51 65 42 59 43 52 65 ...
##   ..$ nKIDS      : num [1:979] 4 2 0 2 0 2 0 1 0 0 ...
##   ..$ nDEP       : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ PHON       : int [1:979] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ dINC_SP    : num [1:979] 0 0 0 0 0 10500 6500 13500 0 0 ...
##   ..$ dINC_A     : num [1:979] 0 36000 30000 464 15000 48000 30000 9000 22500 19500 ...
##   ..$ dHVAL      : num [1:979] 14464 0 0 24928 0 ...
##   ..$ dMBO       : num [1:979] 4 0 0 8464 0 ...
##   ..$ dOUTM      : num [1:979] 0 280 0 584 0 1120 520 0 0 540 ...
##   ..$ dOUTL      : num [1:979] 0 664 0 320 0 0 0 200 200 0 ...
##   ..$ dOUTHP     : num [1:979] 0 0 0 0 0 0 96 0 0 0 ...
##   ..$ dOUTCC     : num [1:979] 0 80 0 60 0 0 0 0 80 0 ...
##   ..$ YOB_missing: num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.B   : num [1:979] 0 0 0 0 0 0 1 0 0 0 ...
##   ..$ EMPS_A.E   : num [1:979] 0 0 0 0 0 1 0 1 1 0 ...
##   ..$ EMPS_A.M   : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.N   : num [1:979] 0 0 1 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.P   : num [1:979] 0 1 0 1 1 0 0 0 0 1 ...
##   ..$ EMPS_A.R   : num [1:979] 1 0 0 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.T   : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.U   : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.V   : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.W   : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.Z   : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ RES.F      : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ RES.N      : num [1:979] 0 0 1 0 0 0 0 0 0 0 ...
##   ..$ RES.O      : num [1:979] 1 1 0 1 0 1 1 1 0 1 ...
##   ..$ RES.P      : num [1:979] 0 0 0 0 1 0 0 0 1 0 ...
##   ..$ RES.U      : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..- attr(*, "terms")=Classes 'terms', 'formula'  language BAD ~ YOB + nKIDS + nDEP + PHON + dINC_S
##   .. .. ..- attr(*, "variables")= language list(BAD, YOB, nKIDS, nDEP, PHON, dINC_SP, dINC_A, dHVAL,
##   .. .. ..- attr(*, "factors")= int [1:30, 1:29] 0 1 0 0 0 0 0 0 0 0 ...
##   .. .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. .. ..$ : chr [1:30] "BAD" "YOB" "nKIDS" "nDEP" ...
##   .. .. .. .. ..$ : chr [1:29] "YOB" "nKIDS" "nDEP" "PHON" ...
##   .. .. ..- attr(*, "term.labels")= chr [1:29] "YOB" "nKIDS" "nDEP" "PHON" ...
##   .. .. ..- attr(*, "order")= int [1:29] 1 1 1 1 1 1 1 1 1 1 ...
##   .. .. ..- attr(*, "intercept")= int 1
##   .. .. ..- attr(*, "response")= int 1
##   .. .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. .. ..- attr(*, "predvars")= language list(BAD, YOB, nKIDS, nDEP, PHON, dINC_SP, dINC_A, dHVAL,
##   .. .. ..- attr(*, "dataClasses")= Named chr [1:30] "factor" "numeric" "numeric" "numeric" ...
##   .. .. .. ..- attr(*, "names")= chr [1:30] "BAD" "YOB" "nKIDS" "nDEP" ...
##  $ call             : language glm(formula = BAD ~ ., family = binomial(link = "logit"), data = tr)
##  $ formula          :Class 'formula'  language BAD ~ .
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##  $ terms            :Classes 'terms', 'formula'  language BAD ~ YOB + nKIDS + nDEP + PHON + dINC_SP
```

```
##   .. ..- attr(*, "variables")= language list(BAD, YOB, nKIDS, nDEP, PHON, dINC_SP, dINC_A, dHVAL, dMB
##   .. ..- attr(*, "factors")= int [1:30, 1:29] 0 1 0 0 0 0 0 0 0 0 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:30] "BAD" "YOB" "nKIDS" "nDEP" ...
##   .. .. .. ..$ : chr [1:29] "YOB" "nKIDS" "nDEP" "PHON" ...
##   .. ..- attr(*, "term.labels")= chr [1:29] "YOB" "nKIDS" "nDEP" "PHON" ...
##   .. ..- attr(*, "order")= int [1:29] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(BAD, YOB, nKIDS, nDEP, PHON, dINC_SP, dINC_A, dHVAL, dMB
##   .. ..- attr(*, "dataClasses")= Named chr [1:30] "factor" "numeric" "numeric" "numeric" ...
##   .. .. ..- attr(*, "names")= chr [1:30] "BAD" "YOB" "nKIDS" "nDEP" ...
##  $ data             :'data.frame':   979 obs. of  30 variables:
##   ..$ YOB       : num [1:979] 19 41 66 51 65 42 59 43 52 65 ...
##   ..$ nKIDS     : num [1:979] 4 2 0 2 0 2 0 1 0 0 ...
##   ..$ nDEP      : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ PHON      : int [1:979] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ dINC_SP   : num [1:979] 0 0 0 0 0 10500 6500 13500 0 0 ...
##   ..$ dINC_A    : num [1:979] 0 36000 30000 464 15000 48000 30000 9000 22500 19500 ...
##   ..$ dHVAL     : num [1:979] 14464 0 0 24928 0 ...
##   ..$ dMBO      : num [1:979] 4 0 0 8464 0 ...
##   ..$ dOUTM     : num [1:979] 0 280 0 584 0 1120 520 0 0 540 ...
##   ..$ dOUTL     : num [1:979] 0 664 0 320 0 0 0 200 200 0 ...
##   ..$ dOUTHP    : num [1:979] 0 0 0 0 0 0 96 0 0 0 ...
##   ..$ dOUTCC    : num [1:979] 0 80 0 60 0 0 0 0 80 0 ...
##   ..$ BAD       : Factor w/ 2 levels "good","bad": 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ YOB_missing: num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.B  : num [1:979] 0 0 0 0 0 0 1 0 0 0 ...
##   ..$ EMPS_A.E  : num [1:979] 0 0 0 0 0 1 0 1 1 0 ...
##   ..$ EMPS_A.M  : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.N  : num [1:979] 0 0 1 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.P  : num [1:979] 0 1 0 1 1 0 0 0 0 1 ...
##   ..$ EMPS_A.R  : num [1:979] 1 0 0 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.T  : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.U  : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.V  : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.W  : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ EMPS_A.Z  : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ RES.F     : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ RES.N     : num [1:979] 0 0 1 0 0 0 0 0 0 0 ...
##   ..$ RES.O     : num [1:979] 1 1 0 1 0 1 1 1 0 1 ...
##   ..$ RES.P     : num [1:979] 0 0 0 0 1 0 0 0 1 0 ...
##   ..$ RES.U     : num [1:979] 0 0 0 0 0 0 0 0 0 0 ...
##  $ offset           : NULL
##  $ control          :List of 3
##   ..$ epsilon: num 1e-08
##   ..$ maxit  : num 25
##   ..$ trace  : logi FALSE
##  $ method           : chr "glm.fit"
##  $ contrasts        : NULL
##  $ xlevels          : Named list()
##  - attr(*, "class")= chr [1:2] "glm" "lm"
```

```
## Classes 'partial' and 'data.frame':  51 obs. of  2 variables:
##  $ dINC_A: num  0 1296 2592 3888 5184 ...
##  $ yhat  : num  0.381 0.373 0.365 0.356 0.348 ...
```

```
## Model for learner.id=classif.randomForest; learner.class=classif.randomForest
## Trained on: task.id = tr; obs = 979; features = 29
## Hyperparameters: replace=TRUE,importance=TRUE,mtry=4,sampsize=200,ntree=1e+03
```

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