Personalized Healthcare Laboratory

Personalized Healthcare Laboratory

Load Packages

```
pacman::p_load(mlr3, mlr3proba, mlr3pipelines,
               mlr3learners, mlr3viz,mlr3tuning,
               mlr3benchmark, mlr3misc, mlr3extralearners,
               R6, data.table, lgr, uuid, mlbench, digest,
               backports, checkmate, paradox, reticulate,
               keras, devtools, survival, rms, summarytools, knitr)
```

This function is a wrapper for library and require. It checks to see if a package is installed, if not it attempts to install the package from CRAN and/or any other repository in the pacman repository list.

Then we install the required packages and import into environvment.

```
library(remotes)
```

```
##
## Attaching package: 'remotes'
```

```
## The following objects are masked from 'package:devtools':
##
##     dev_package_deps, install_bioc, install_bitbucket, install_cran,
##     install_deps, install_dev, install_git, install_github,
##     install_gitlab, install_local, install_svn, install_url,
##     install_version, update_packages
```

```
## The following object is masked from 'package:usethis':
##
##     git_credentials
```

```
install_github("binderh/CoxBoost")
```

```
## Skipping install of 'CoxBoost' from a github remote, the SHA1 (1dc47d70) has not changed since last i
##   Use `force = TRUE` to force installation
```

```
install.packages("mlr3verse")
```

```
##
## The downloaded binary packages are in
##  /var/folders/tx/6bn9p3z538j26cxyj373sb2r0000gn/T//RtmpntAjMA/downloaded_packages
```

```
remotes::install_github("mlr-org/mlr3extralearners")
```

```
## Skipping install of 'mlr3extralearners' from a github remote, the SHA1 (38759e6c) has not changed sin
##   Use `force = TRUE` to force installation
```

```
library(mlr3extralearners)
install_learners('surv.coxboost')
```

```
library(mlr3learners) #ranger
library(mlr3proba) #coxph
```

In particular we install: * **CoxBoost** : This package provides routines for fitting Cox models by likelihood based boosting for a single endpoint or in presence of competing risks. * **mlr3verse** : This package is intended to simplify both installation and loading of packages from the mlr3 ecosystem. Instead of depending on the extension packages, functions required for data analysis are re-exported, providing a thin view on the most important functionality of the mlr3 ecosystem. * **mlr3extralearners** : mlr3extralearners contains all learners from mlr3 that are not in mlr3learners or the core packages. mlr3extralearners contains helper functions to find where all the learners, across the mlr3verse, live and to install required packages to run these learners. See the interactive learner list for the full list of learners in the mlr3verse and the learner status page for a live build status. * **mlr3proba** : mlr3proba is a machine learning toolkit for making probabilistic predictions within the mlr3 ecosystem. It currently supports the tasks of Probabilistic supervised regression, Predictive survival analysis, Unconditional distribution estimation.

Load Dataset

```
gbcs <- mlr3proba::gbcs

gbcs2 <- gbcs[,c(5:12,15:16)]
summarytools::dfSummary(gbcs2,
                        graph.col = F,
                        valid.col = F
)
```

```
## Data Frame Summary
## gbcs2
## Dimensions: 686 x 10
## Duplicates: 0
##
## ------------------------------------------------------------------------------
## No   Variable      Stats / Values                Freqs (% of Valid)    Missing
## ---- ------------- ----------------------------- --------------------- ---------
## 1    age           Mean (sd) : 53.1 (10.1)       54 distinct values    0
##      [numeric]     min < med < max:                                    (0.0%)
##                    21 < 53 < 80
##                    IQR (CV) : 15 (0.2)
##
## 2    menopause     Min  : 1                     1 : 290 (42.3%)       0
##      [integer]     Mean : 1.6                   2 : 396 (57.7%)       (0.0%)
##                    Max  : 2
##
## 3    hormone       Min  : 1                     1 : 440 (64.1%)       0
##      [integer]     Mean : 1.4                   2 : 246 (35.9%)       (0.0%)
##                    Max  : 2
##
## 4    size          Mean (sd) : 29.3 (14.3)       58 distinct values    0
##      [numeric]     min < med < max:                                    (0.0%)
##                    3 < 25 < 120
##                    IQR (CV) : 15 (0.5)
##
## 5    grade         Mean (sd) : 2.1 (0.6)         1 :  81 (11.8%)       0
##      [integer]     min < med < max:              2 : 444 (64.7%)       (0.0%)
##                    1 < 2 < 3                     3 : 161 (23.5%)
```

```
##                   IQR (CV) : 0 (0.3)
##
## 6    nodes        Mean (sd) : 5 (5.5)          30 distinct values    0
##      [integer]    min < med < max:                                   (0.0%)
##                   1 < 3 < 51
##                   IQR (CV) : 6 (1.1)
##
## 7    prog_recp    Mean (sd) : 110 (202.3)      242 distinct values   0
##      [numeric]    min < med < max:                                   (0.0%)
##                   0 < 32.5 < 2380
##                   IQR (CV) : 124.8 (1.8)
##
## 8    estrg_recp   Mean (sd) : 96.3 (153.1)     244 distinct values   0
##      [numeric]    min < med < max:                                   (0.0%)
##                   0 < 36 < 1144
##                   IQR (CV) : 106 (1.6)
##
## 9    survtime     Mean (sd) : 1320.6 (619.2)   574 distinct values   0
##      [numeric]    min < med < max:                                   (0.0%)
##                   8 < 1338 < 2668
##                   IQR (CV) : 1026 (0.5)
##
## 10   censdead     Min  : 0                     0 : 515 (75.1%)       0
##      [integer]    Mean : 0.2                   1 : 171 (24.9%)       (0.0%)
##                   Max  : 1
## ----------------------------------------------------------------------------
```

```
head(gbcs) # print example
```

```
##   id   diagdate    recdate  deathdate age menopause hormone size grade nodes
## 1  1 1984-08-17 1988-04-15 1990-11-16  38         1       1   18     3     5
## 2  2 1985-04-25 1989-03-15 1990-10-22  52         1       1   20     1     1
## 3  3 1984-10-11 1988-04-12 1988-10-06  47         1       1   30     2     1
## 4  4 1984-06-29 1984-11-24 1984-11-24  40         1       1   24     1     3
## 5  5 1984-07-03 1989-08-09 1989-08-09  64         2       2   19     2     1
## 6  6 1984-07-24 1989-11-08 1989-11-08  49         2       2   56     1     3
##   prog_recp estrg_recp rectime censrec survtime censdead
## 1       141        105    1337       1     2282        0
## 2        78         14    1420       1     2006        0
## 3       422         89    1279       1     1456        1
## 4        25         11     148       0      148        0
## 5        19          9    1863       0     1863        0
## 6       356         64    1933       0     1933        0
```

The dataset has the followuing attributes: * **id** : Identification Code * **diagdate** : Date of diagnosis. * **recdate** : Date of recurrence free survival. * **deathdate** : Date of death. * **age** : Age at diagnosis (years). * **menopause** : Menopausal status. 1 = Yes, 0 = No. * **hormone** : Hormone therapy. 1 = Yes. 0 = No. * **size** : Tumor size (mm). * **grade** : Tumor grade (1-3). * **nodes** : Number of nodes. * **prog_recp** : Number of progesterone receptors. * **estrg_recp** : Number of estrogen receptors. * **rectime** : Time to recurrence (days). * **censrec** : Recurrence status. 1 = Recurrence. 0 = Censored. * **survtime** : Time to death (days). * **censdead** : Censoring status. 1 = Death. 0 = Censored.

Data Cleaning

```
gbcs2$age <- scale(gbcs2$age)
gbcs2$menopause <- gbcs2$menopause-1
gbcs2$hormone <- gbcs2$hormone-1
gbcs2$size <- scale(gbcs2$size)
gbcs2$grade1 <- ifelse(gbcs2$grade==1, 1,0)
gbcs2$grade2 <- ifelse(gbcs2$grade==2, 1,0)
gbcs2$grade3 <- ifelse(gbcs2$grade==3, 1,0)
gbcs2$grade <- NULL
gbcs2$nodes <- scale(gbcs2$nodes)
gbcs2$prog_recp <- scale(gbcs2$prog_recp)
gbcs2$estrg_recp <- scale(gbcs2$estrg_recp)
```

We perform some preliminary operations of data cleaning on the dataset: * scale is generic function whose default method centers and/or scales the columns of a numeric matrix. We apply that to age, size, nodes, prog_recp and estrg_recp attributes. * create a boolean attribute for the grade of tumor (1, 2, 3) then set to null the original attribute * change values for boolean attributes menopause, hormone (just for convenience purposes).

```
set.seed(123)
train_set = sample(nrow(gbcs2), 0.8 * nrow(gbcs2))
#str(train_set)
test_set = setdiff(seq_len(nrow(gbcs2)), train_set)

## train/test set initialization and summary

train_gbcs <- gbcs2[train_set, ]
summarytools::dfSummary(train_gbcs,
                        graph.col = F,
                        valid.col = F
)
```

```
## Data Frame Summary
## train_gbcs
## Dimensions: 548 x 12
## Duplicates: 0
##
## ------------------------------------------------------------------------------------
## No   Variable          Stats / Values                Freqs (% of Valid)    Missing
## ---- ----------------- ----------------------------- --------------------- ---------
## 1    age               Mean (sd) : 0 (1)             50 distinct values    0
##      [matrix, array]   min < med < max:                                     (0.0%)
##                        -2.8 < -0.1 < 2.7
##                        IQR (CV) : 1.5 (-43)
##
## 2    menopause         Min  : 0                     0 : 238 (43.4%)       0
##      [numeric]         Mean : 0.6                   1 : 310 (56.6%)       (0.0%)
##                        Max  : 1
##
## 3    hormone           Min  : 0                     0 : 352 (64.2%)       0
##      [numeric]         Mean : 0.4                   1 : 196 (35.8%)       (0.0%)
##                        Max  : 1
##
## 4    size              Mean (sd) : 0 (1)             54 distinct values    0
```

```
##      [matrix, array]   min < med < max:                                    (0.0%)
##                        -1.8 < -0.3 < 6.3
##                        IQR (CV) : 1 (-64.2)
##
## 5    nodes             Mean (sd) : 0 (1)            29 distinct values    0
##      [matrix, array]   min < med < max:                                    (0.0%)
##                        -0.7 < -0.4 < 8.4
##                        IQR (CV) : 1.1 (-124.3)
##
## 6    prog_recp         Mean (sd) : 0 (1.1)          214 distinct values   0
##      [matrix, array]   min < med < max:                                    (0.0%)
##                        -0.5 < -0.4 < 11.2
##                        IQR (CV) : 0.6 (56.3)
##
## 7    estrg_recp        Mean (sd) : 0 (1)            217 distinct values   0
##      [matrix, array]   min < med < max:                                    (0.0%)
##                        -0.6 < -0.4 < 6.8
##                        IQR (CV) : 0.7 (87.2)
##
## 8    survtime          Mean (sd) : 1321.2 (613.4)   467 distinct values   0
##      [numeric]         min < med < max:                                    (0.0%)
##                        8 < 1338 < 2668
##                        IQR (CV) : 994.8 (0.5)
##
## 9    censdead          Min  : 0                     0 : 416 (75.9%)       0
##      [integer]         Mean : 0.2                   1 : 132 (24.1%)       (0.0%)
##                        Max  : 1
##
## 10   grade1            Min  : 0                     0 : 481 (87.8%)       0
##      [numeric]         Mean : 0.1                   1 :  67 (12.2%)       (0.0%)
##                        Max  : 1
##
## 11   grade2            Min  : 0                     0 : 197 (35.9%)       0
##      [numeric]         Mean : 0.6                   1 : 351 (64.1%)       (0.0%)
##                        Max  : 1
##
## 12   grade3            Min  : 0                     0 : 418 (76.3%)       0
##      [numeric]         Mean : 0.2                   1 : 130 (23.7%)       (0.0%)
##                        Max  : 1
## ----------------------------------------------------------------------------
```

```
test_gbcs <- gbcs2[test_set, ]
summarytools::dfSummary(test_gbcs,
                        graph.col = F,
                        valid.col = F
)
```

```
## Data Frame Summary
## test_gbcs
## Dimensions: 138 x 12
## Duplicates: 0
##
## ----------------------------------------------------------------------------
## No   Variable          Stats / Values               Freqs (% of Valid)    Missing
## ---- ----------------- ---------------------------- --------------------- ---------
```

```
## 1    age               Mean (sd) : 0.1 (1)          40 distinct values    0
##      [matrix, array]   min < med < max:                                    (0.0%)
##                        -3.2 < 0 < 2.4
##                        IQR (CV) : 1.4 (10.5)
##
## 2    menopause         Min  : 0                     0 : 52 (37.7%)        0
##      [numeric]         Mean : 0.6                   1 : 86 (62.3%)        (0.0%)
##                        Max  : 1
##
## 3    hormone           Min  : 0                     0 : 88 (63.8%)        0
##      [numeric]         Mean : 0.4                   1 : 50 (36.2%)        (0.0%)
##                        Max  : 1
##
## 4    size              Mean (sd) : 0.1 (1)          44 distinct values    0
##      [matrix, array]   min < med < max:                                    (0.0%)
##                        -1.8 < -0.1 < 3.4
##                        IQR (CV) : 1.2 (16.5)
##
## 5    nodes             Mean (sd) : 0 (0.9)          20 distinct values    0
##      [matrix, array]   min < med < max:                                    (0.0%)
##                        -0.7 < -0.3 < 5.7
##                        IQR (CV) : 0.9 (28.4)
##
## 6    prog_recp         Mean (sd) : -0.1 (0.8)       89 distinct values    0
##      [matrix, array]   min < med < max:                                    (0.0%)
##                        -0.5 < -0.4 < 3.7
##                        IQR (CV) : 0.6 (-10.3)
##
## 7    estrg_recp        Mean (sd) : 0 (0.9)          85 distinct values    0
##      [matrix, array]   min < med < max:                                    (0.0%)
##                        -0.6 < -0.4 < 6.5
##                        IQR (CV) : 0.6 (-20.3)
##
## 8    survtime          Mean (sd) : 1318.4 (644.2)   133 distinct values   0
##      [numeric]         min < med < max:                                    (0.0%)
##                        16 < 1331.5 < 2612
##                        IQR (CV) : 1097.8 (0.5)
##
## 9    censdead          Min  : 0                     0 : 99 (71.7%)        0
##      [integer]         Mean : 0.3                   1 : 39 (28.3%)        (0.0%)
##                        Max  : 1
##
## 10   grade1            Min  : 0                     0 : 124 (89.9%)       0
##      [numeric]         Mean : 0.1                   1 :  14 (10.1%)       (0.0%)
##                        Max  : 1
##
## 11   grade2            Min  : 0                     0 : 45 (32.6%)        0
##      [numeric]         Mean : 0.7                   1 : 93 (67.4%)        (0.0%)
##                        Max  : 1
##
## 12   grade3            Min  : 0                     0 : 107 (77.5%)       0
##      [numeric]         Mean : 0.2                   1 :  31 (22.5%)       (0.0%)
##                        Max  : 1
## ----------------------------------------------------------------------------
```

set.seed is a function that permits to probabilistic/random processes such as estimation to be reproduced in a deterministic way. So for reproducibilty of the experiment and of our result we set it to a fixed integer value (123). **sample** function takes a sample of the specified size from the elements of gbcs2 without replacement (default).

In the end we print 2 summaries of the characteristics of the train and test dataset.

Analysis

Cox Model

```
fit <- coxph(Surv(survtime, censdead) ~ age + menopause + hormone + size + grade1 + grade2 + nodes + pr
summary(fit)
```

```
## Call:
## coxph(formula = Surv(survtime, censdead) ~ age + menopause +
##     hormone + size + grade1 + grade2 + nodes + prog_recp + estrg_recp,
##     data = train_gbcs)
##
##   n= 548, number of events= 132
##
##                coef exp(coef) se(coef)      z Pr(>|z|)
## age         0.15550   1.16825  0.14199  1.095  0.27345
## menopause  -0.17756   0.83731  0.29747 -0.597  0.55058
## hormone    -0.22560   0.79804  0.19050 -1.184  0.23631
## size        0.22094   1.24725  0.07416  2.979  0.00289 **
## grade1     -1.28643   0.27626  0.48423 -2.657  0.00789 **
## grade2     -0.48236   0.61732  0.19292 -2.500  0.01241 *
## nodes       0.29402   1.34182  0.05827  5.046 4.51e-07 ***
## prog_recp  -1.01213   0.36345  0.25477 -3.973 7.11e-05 ***
## estrg_recp  0.05275   1.05416  0.11275  0.468  0.63990
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##            exp(coef) exp(-coef) lower .95 upper .95
## age           1.1682     0.8560    0.8844    1.5431
## menopause     0.8373     1.1943    0.4674    1.5000
## hormone       0.7980     1.2531    0.5494    1.1592
## size          1.2472     0.8018    1.0785    1.4424
## grade1        0.2763     3.6198    0.1069    0.7137
## grade2        0.6173     1.6199    0.4230    0.9010
## nodes         1.3418     0.7453    1.1970    1.5042
## prog_recp     0.3634     2.7514    0.2206    0.5988
## estrg_recp    1.0542     0.9486    0.8452    1.3149
##
## Concordance= 0.758  (se = 0.021 )
## Likelihood ratio test= 96.67  on 9 df,   p=<2e-16
## Wald test            = 94.78  on 9 df,   p=<2e-16
## Score (logrank) test = 106.9  on 9 df,   p=<2e-16
```

```
check_PH <- cox.zph(fit, transform = "km")
check_PH
```

```
##              chisq df     p
## age         0.0315  1 0.859
## menopause   0.0797  1 0.778
```

```
## hormone     0.1963  1 0.658
## size        0.0316  1 0.859
## grade1      0.1226  1 0.726
## grade2      3.6356  1 0.057
## nodes       0.7930  1 0.373
## prog_recp   5.7383  1 0.017
## estrg_recp  5.1836  1 0.023
## GLOBAL     13.2802  9 0.150
```

```
ND <- data.frame(age = 0, menopause = 1, hormone = 2,
                 size = 0, grade1 = c(1,0,0), grade2=c(0,1,0), grade3=c(0,0,1), nodes = 0, prog_recp=0,
```

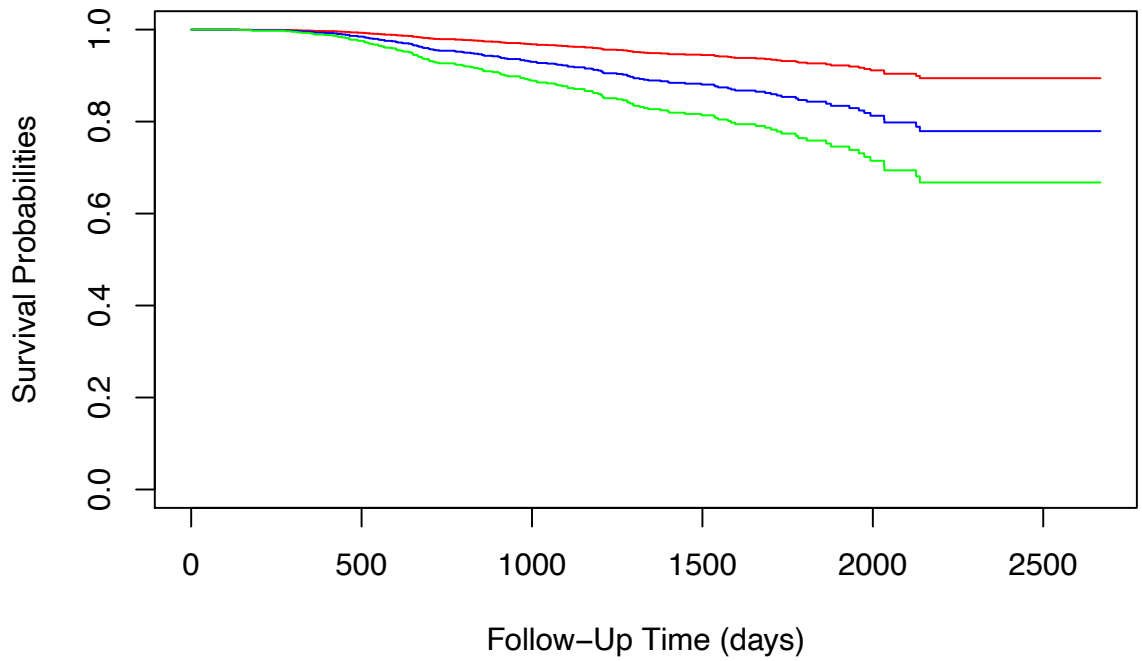
```
surv_probs_Cox <- survfit(fit, newdata = ND)
surv_probs_Cox
```

```
## Call: survfit(formula = fit, newdata = ND)
##
##     n events median 0.95LCL 0.95UCL
## 1 548    132     NA      NA      NA
## 2 548    132     NA      NA      NA
## 3 548    132     NA    2138      NA
```

```
summary(surv_probs_Cox, times = 500)
```

```
## Call: survfit(formula = fit, newdata = ND)
##
##  time n.risk n.event survival1 survival2 survival3
##   500    501      20     0.993     0.985     0.975
```

```
plot(surv_probs_Cox, col = c("red", "blue", "green"),
     xlab = "Follow-Up Time (days)", ylab = "Survival Probabilities")
```




```
task_gbcs = TaskSurv$new(id = "train_gbcs", backend = train_gbcs, time = "survtime", event = "censdead"
test_gbcs = TaskSurv$new(id = "test_gbcs", backend = test_gbcs, time = "survtime", event = "censdead")
```

```
learner.cox = lrn("surv.coxph")
learner.cox$train(task_gbcs)
learner.cox$model
```

```
## Call:
## survival::coxph(formula = task$formula(), data = task$data(),
##     x = TRUE)
##
##                coef exp(coef) se(coef)      z        p
## age         0.15550   1.16825  0.14199  1.095  0.27345
## estrg_recp  0.05275   1.05416  0.11275  0.468  0.63990
## grade1     -1.28643   0.27626  0.48423 -2.657  0.00789
## grade2     -0.48236   0.61732  0.19292 -2.500  0.01241
## grade3           NA        NA  0.00000     NA       NA
## hormone    -0.22560   0.79804  0.19050 -1.184  0.23631
## menopause  -0.17756   0.83731  0.29747 -0.597  0.55058
## nodes       0.29402   1.34182  0.05827  5.046 4.51e-07
## prog_recp  -1.01213   0.36345  0.25477 -3.973 7.11e-05
## size        0.22094   1.24725  0.07416  2.979  0.00289
##
## Likelihood ratio test=96.67  on 9 df, p=< 2.2e-16
## n= 548, number of events= 132
```

```
prediction.cox = learner.cox$predict(test_gbcs)
prediction.cox
```

```
## <PredictionSurv> for 138 observations:
##     row_ids time status      crank         lp                      distr
##           1 2282  FALSE -0.5312450 -0.5312450 <VectorDistribution[138]>
##           2 1456   TRUE -2.3158269 -2.3158269 <VectorDistribution[138]>
##           3 2563  FALSE -0.4070265 -0.4070265 <VectorDistribution[138]>
## ---
##         136  841  FALSE -0.2694111 -0.2694111 <VectorDistribution[138]>
##         137   16  FALSE  1.1737511  1.1737511 <VectorDistribution[138]>
##         138  857  FALSE -0.4004860 -0.4004860 <VectorDistribution[138]>
```

```
prediction.cox$score()
```

```
## surv.harrell_c
##      0.6897718
```

```
measure = lapply(c("surv.graf"), msr)
```

```
## Warning in .__MeasureSurvGraf__initialize(self = self, private = private, : The
## default of 'proper' will be changed to 'TRUE' in v0.6.0.
```

```
prediction.cox$score(measure)
```

```
## surv.graf
## 0.1326272
```

Support-Vector Machine

```
library("bbotk")
library("mlr3tuning")
```

```
install_learners('surv.svm')
svm <- lrn('surv.svm')

svm$param_set$values = list(gamma.mu = 1, kernel = "lin_kernel", opt.meth = "ipop")
svm$train(task_gbcs)
svm$model
```

```
##
## survivalsvm result
##
## Call:
##
##  survivalsvm::survivalsvm(formula = task$formula(), data = task$data(), gamma.mu = 1, kernel = "lin_
##
## Survival svm approach              : regression
## Type of Kernel                     : lin_kernel
## Optimization solver used           : ipop
## Number of support vectors retained : 548
## survivalsvm version                : 0.0.5
```

At the heart of mlr3tuning are the R6 classes: * **TuningInstanceSingleCrit, TuningInstanceMultiCrit** : These two classes describe the tuning problem and store the results. * **Tuner**: This class is the base class for implementations of tuning algorithms.

```
svm.pred <- svm$predict(test_gbcs)
svm.pred$score()
```

```
## surv.harrell_c
##      0.6534235
```

We use the SVM algorithm from rpart and choose a subset of the hyperparameters we want to tune. This is often referred to as the **"tuning space."**

```
svm$param_set
```

```
## <ParamSet>
##              id    class lower upper nlevels        default parents      value
##  1:       bound ParamDbl     0   Inf     Inf             10
##  2:    conv.tol ParamDbl     0   Inf     Inf          1e-07
##  3:   diff.meth ParamFct    NA    NA       3 <NoDefault[3]>    type
##  4:     eig.tol ParamDbl     0   Inf     Inf          1e-06
##  5:    gamma.mu ParamUty    NA    NA     Inf <NoDefault[3]>       1
##  6:      kernel ParamFct    NA    NA       4     lin_kernel lin_kernel
##  7: kernel.pars ParamUty    NA    NA     Inf <NoDefault[3]>
##  8:      margin ParamDbl     0   Inf     Inf           0.05
##  9:     maxiter ParamInt     0   Inf     Inf             20
## 10:    opt.meth ParamFct    NA    NA       2       quadprog       ipop
## 11:    posd.tol ParamDbl     0   Inf     Inf          1e-08
## 12:      sgf.sv ParamInt     0   Inf     Inf              5
## 13:        sigf ParamInt     0   Inf     Inf              7
## 14:        type ParamFct    NA    NA       4     regression
```

Here, we opt to tune parameter **gamma.mu** as a double value (x.xx) in the set 0.01 and 1.

```
search_space = ps(gamma.mu = p_dbl(lower = 0.01, upper = 1))
search_space
```

```
## <ParamSet>
##          id    class lower upper nlevels        default value
## 1: gamma.mu ParamDbl  0.01     1     Inf <NoDefault[3]>
```

Next, we need to specify how to evaluate the performance. For this, we need to choose a resampling strategy

```
hout = rsmp("holdout")
```

and the performance measure

```
measure = msr("surv.cindex")
```

Finally, one has to select the budget available, to solve this tuning instance. This is done by selecting one of the available Terminators. We decided to set : terminate after 8 iteration to same computational costs with respect to a potential gain in performance.

```
evals8 = trm("evals", n_evals = 8)

instance = TuningInstanceSingleCrit$new(
  task = task_gbcs,
  learner = svm,
  resampling = hout,
  measure = measure,
  search_space = search_space,
  terminator = evals8
)
#Type of optimization
tuner = tnr("grid_search", resolution = 10)
```

Through the Tuner R6 class we trigger the tuner. To start the tuning, we simply pass the TuningInstanceSingleCrit to the $optimize() method of the initialized Tuner. The tuner proceeds as follows

```
tuner$optimize(instance)
```

```
## INFO  [09:47:53.468] [bbotk] Starting to optimize 1 parameter(s) with '<OptimizerGridSearch>' and '<
## INFO  [09:47:53.531] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:47:53.617] [mlr3]  Running benchmark with 1 resampling iterations
## INFO  [09:47:53.892] [mlr3]  Applying learner 'surv.svm' on task 'train_gbcs' (iter 1/1)
## INFO  [09:47:58.434] [mlr3]  Finished benchmark
## INFO  [09:47:58.544] [bbotk] Result of batch 1:
## INFO  [09:47:58.546] [bbotk]  gamma.mu surv.harrell_c                                uhash
## INFO  [09:47:58.546] [bbotk]      0.12      0.7162162 2869ae55-315d-46c8-b333-c7d87c97d10a
## INFO  [09:47:58.548] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:47:58.632] [mlr3]  Running benchmark with 1 resampling iterations
## INFO  [09:47:58.649] [mlr3]  Applying learner 'surv.svm' on task 'train_gbcs' (iter 1/1)
## INFO  [09:48:02.916] [mlr3]  Finished benchmark
## INFO  [09:48:02.980] [bbotk] Result of batch 2:
## INFO  [09:48:02.982] [bbotk]  gamma.mu surv.harrell_c                                uhash
## INFO  [09:48:02.982] [bbotk]         1      0.7225225 bf7e457e-fa4e-4f46-9c5c-d051d4372776
## INFO  [09:48:02.984] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:48:03.030] [mlr3]  Running benchmark with 1 resampling iterations
## INFO  [09:48:03.038] [mlr3]  Applying learner 'surv.svm' on task 'train_gbcs' (iter 1/1)
## INFO  [09:48:07.632] [mlr3]  Finished benchmark
## INFO  [09:48:07.704] [bbotk] Result of batch 3:
## INFO  [09:48:07.706] [bbotk]  gamma.mu surv.harrell_c                                uhash
## INFO  [09:48:07.706] [bbotk]      0.67      0.7210811 be16e297-caa4-42ab-b978-a48ffa6fd2d6
## INFO  [09:48:07.708] [bbotk] Evaluating 1 configuration(s)
```

```
## INFO  [09:48:07.755] [mlr3]  Running benchmark with 1 resampling iterations
## INFO  [09:48:07.763] [mlr3]  Applying learner 'surv.svm' on task 'train_gbcs' (iter 1/1)
## INFO  [09:48:11.803] [mlr3]  Finished benchmark
## INFO  [09:48:11.871] [bbotk] Result of batch 4:
## INFO  [09:48:11.873] [bbotk]  gamma.mu surv.harrell_c                                uhash
## INFO  [09:48:11.873] [bbotk]      0.89       0.721982 fc0b2e09-193e-453e-ba80-440ee10925ee
## INFO  [09:48:11.874] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:48:11.923] [mlr3]  Running benchmark with 1 resampling iterations
## INFO  [09:48:11.931] [mlr3]  Applying learner 'surv.svm' on task 'train_gbcs' (iter 1/1)
## INFO  [09:48:16.142] [mlr3]  Finished benchmark
## INFO  [09:48:16.207] [bbotk] Result of batch 5:
## INFO  [09:48:16.209] [bbotk]  gamma.mu surv.harrell_c                                uhash
## INFO  [09:48:16.209] [bbotk]      0.34      0.7163964 9029cef0-0867-48e5-8753-00e2155ab7ef
## INFO  [09:48:16.210] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:48:16.258] [mlr3]  Running benchmark with 1 resampling iterations
## INFO  [09:48:16.273] [mlr3]  Applying learner 'surv.svm' on task 'train_gbcs' (iter 1/1)
## INFO  [09:48:20.637] [mlr3]  Finished benchmark
## INFO  [09:48:20.736] [bbotk] Result of batch 6:
## INFO  [09:48:20.738] [bbotk]  gamma.mu surv.harrell_c                                uhash
## INFO  [09:48:20.738] [bbotk]      0.23       0.716036 680d3dd7-9939-4eed-9de4-be92ad6403ac
## INFO  [09:48:20.740] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:48:20.835] [mlr3]  Running benchmark with 1 resampling iterations
## INFO  [09:48:20.860] [mlr3]  Applying learner 'surv.svm' on task 'train_gbcs' (iter 1/1)
## INFO  [09:48:25.469] [mlr3]  Finished benchmark
## INFO  [09:48:25.534] [bbotk] Result of batch 7:
## INFO  [09:48:25.536] [bbotk]  gamma.mu surv.harrell_c                                uhash
## INFO  [09:48:25.536] [bbotk]      0.78      0.7181982 d2db9193-42d6-4ec7-a241-a96ca9a7ce70
## INFO  [09:48:25.537] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:48:25.584] [mlr3]  Running benchmark with 1 resampling iterations
## INFO  [09:48:25.592] [mlr3]  Applying learner 'surv.svm' on task 'train_gbcs' (iter 1/1)
## INFO  [09:48:29.526] [mlr3]  Finished benchmark
## INFO  [09:48:29.597] [bbotk] Result of batch 8:
## INFO  [09:48:29.598] [bbotk]  gamma.mu surv.harrell_c                                uhash
## INFO  [09:48:29.598] [bbotk]      0.45      0.7192793 7db8145a-791b-448a-9e7f-ae8684d4fb08
## INFO  [09:48:29.608] [bbotk] Finished optimizing after 8 evaluation(s)
## INFO  [09:48:29.610] [bbotk] Result:
## INFO  [09:48:29.613] [bbotk]  gamma.mu learner_param_vals  x_domain surv.harrell_c
## INFO  [09:48:29.613] [bbotk]         1          <list[3]> <list[1]>      0.7225225
```

```
##    gamma.mu learner_param_vals  x_domain surv.harrell_c
## 1:        1          <list[3]> <list[1]>      0.7225225
```

```
print(instance$result_learner_param_vals)
```

```
## $gamma.mu
## [1] 1
##
## $kernel
## [1] "lin_kernel"
##
## $opt.meth
## [1] "ipop"
```

We obtain the best performanc e result

```
print(instance$result_y)
```

```
## surv.harrell_c
##      0.7225225
```

However, one can investigate all resamplings which were undertaken, as they are stored in the archive of the TuningInstanceSingleCrit and can be accessed by using as.data.table()

```
as.data.table(instance$archive)
```

```
##    gamma.mu surv.harrell_c                                uhash
## 1:     0.12      0.7162162 2869ae55-315d-46c8-b333-c7d87c97d10a
## 2:     1.00      0.7225225 bf7e457e-fa4e-4f46-9c5c-d051d4372776
## 3:     0.67      0.7210811 be16e297-caa4-42ab-b978-a48ffa6fd2d6
## 4:     0.89      0.7219820 fc0b2e09-193e-453e-ba80-440ee10925ee
## 5:     0.34      0.7163964 9029cef0-0867-48e5-8753-00e2155ab7ef
## 6:     0.23      0.7160360 680d3dd7-9939-4eed-9de4-be92ad6403ac
## 7:     0.78      0.7181982 d2db9193-42d6-4ec7-a241-a96ca9a7ce70
## 8:     0.45      0.7192793 7db8145a-791b-448a-9e7f-ae8684d4fb08
##              timestamp batch_nr x_domain_gamma.mu
## 1: 2021-06-01 09:47:58        1              0.12
## 2: 2021-06-01 09:48:02        2              1.00
## 3: 2021-06-01 09:48:07        3              0.67
## 4: 2021-06-01 09:48:11        4              0.89
## 5: 2021-06-01 09:48:16        5              0.34
## 6: 2021-06-01 09:48:20        6              0.23
## 7: 2021-06-01 09:48:25        7              0.78
## 8: 2021-06-01 09:48:29        8              0.45
```

Now the optimized hyperparameters can take the previously created Learner, set the returned hyperparameters and train it on the full dataset (follows).

```
svm$param_set$values = instance$result_learner_param_vals
```

```
svm$train(task_gbcs)
```

The trained model can now be used to make a prediction on external data. Note that predicting on observations present in the task, should be avoided. The model has seen these observations already during tuning and therefore results would be statistically biased. Hence, the resulting performance measure would be over-optimistic. Instead, to get statistically unbiased performance estimates for the current task, nested resampling is required.

Then we train the model so we can use the learner like any other learner, calling the $train()

```
svm$model
```

```
##
## survivalsvm result
##
## Call:
##
##  survivalsvm::survivalsvm(formula = task$formula(), data = task$data(), gamma.mu = 1, kernel = "lin_
##
## Survival svm approach              : regression
## Type of Kernel                     : lin_kernel
## Optimization solver used           : ipop
## Number of support vectors retained : 548
## survivalsvm version                : 0.0.5
```

and the $predict() method.

```
svm.pred <- svm$predict(test_gbcs)
svm.pred$score()
```

```
## surv.harrell_c
##      0.6534235
```

Random Forest

```
install.packages('ranger')
```

```
##
## The downloaded binary packages are in
##  /var/folders/tx/6bn9p3z538j26cxyj373sb2r0000gn/T//RtmpntAjMA/downloaded_packages
```

```
library(ranger)
library("mlr3verse")
```

```
##
## Attaching package: 'mlr3verse'
```

```
## The following objects are masked from 'package:mlr3extralearners':
##
##     lrn, lrns
```

```
rf <-lrn("surv.ranger")
rf$train(task_gbcs)
rf$oob_error()
```

```
## [1] 0.2967632
```

```
rf$model
```

```
## Ranger result
##
## Call:
##  ranger::ranger(formula = NULL, dependent.variable.name = targets[1L],      status.variable.name = ta
##
## Type:                             Survival
## Number of trees:                  500
## Sample size:                      548
## Number of independent variables:  10
## Mtry:                             4
## Target node size:                 3
## Variable importance mode:         none
## Splitrule:                        logrank
## Number of unique death times:     467
## OOB prediction error (1-C):       0.2967632
```

```
rf.pred <- rf$predict(test_gbcs)
rf.pred$score()
```

```
## surv.harrell_c
##      0.6979431
```

```
rf$param_set
```

```
## <ParamSet>
```

```
##                              id    class lower upper nlevels        default
##  1:                   num.trees ParamInt     1   Inf     Inf            500
##  2:                        mtry ParamInt     1   Inf     Inf <NoDefault[3]>
##  3:                  importance ParamFct    NA    NA       4 <NoDefault[3]>
##  4:                write.forest ParamLgl    NA    NA       2           TRUE
##  5:               min.node.size ParamInt     1   Inf     Inf              5
##  6:                     replace ParamLgl    NA    NA       2           TRUE
##  7:             sample.fraction ParamDbl     0     1     Inf <NoDefault[3]>
##  8:                    splitrule ParamFct    NA    NA       4        logrank
##  9:           num.random.splits ParamInt     1   Inf     Inf              1
## 10:                   max.depth ParamInt  -Inf   Inf     Inf
## 11:                       alpha ParamDbl  -Inf   Inf     Inf            0.5
## 12:                     minprop ParamDbl  -Inf   Inf     Inf            0.1
## 13:       regularization.factor ParamUty    NA    NA     Inf              1
## 14:     regularization.usedepth ParamLgl    NA    NA       2          FALSE
## 15:                        seed ParamInt  -Inf   Inf     Inf
## 16:        split.select.weights ParamDbl     0     1     Inf <NoDefault[3]>
## 17:      always.split.variables ParamUty    NA    NA     Inf <NoDefault[3]>
## 18:   respect.unordered.factors ParamFct    NA    NA       3         ignore
## 19: scale.permutation.importance ParamLgl    NA    NA       2          FALSE
## 20:                   keep.inbag ParamLgl    NA    NA       2          FALSE
## 21:                      holdout ParamLgl    NA    NA       2          FALSE
## 22:                  num.threads ParamInt     1   Inf     Inf              1
## 23:                  save.memory ParamLgl    NA    NA       2          FALSE
## 24:                      verbose ParamLgl    NA    NA       2           TRUE
## 25:                    oob.error ParamLgl    NA    NA       2           TRUE
##                              id    class lower upper nlevels        default
##     value
##  1:
##  2:
##  3:
##  4:
##  5:
##  6:
##  7:
##  8:
##  9:
## 10:
## 11:
## 12:
## 13:
## 14:
## 15:
## 16:
## 17:
## 18:
## 19:
## 20:
## 21:
## 22:     1
## 23:
## 24:
## 25:
##     value
```

```
search_space = ps(
  min.node.size = p_int(lower = 1, upper = 6),
  mtry = p_int(lower = 2, upper = 10),
  sample.fraction = p_dbl(lower = 0.5, upper = 0.7)
  )

search_space
```

```
## <ParamSet>
##                 id    class lower upper nlevels        default value
## 1:   min.node.size ParamInt   1.0   6.0       6 <NoDefault[3]>
## 2:            mtry ParamInt   2.0  10.0       9 <NoDefault[3]>
## 3: sample.fraction ParamDbl   0.5   0.7     Inf <NoDefault[3]>
```

We use as ressampling strategy a cross validation with 5 folds

```
hout = rsmp("cv", folds = 5)
```

```
measure = msr("surv.cindex")
```

```
evalsTerm = trm("stagnation")
```

```
instance = TuningInstanceSingleCrit$new(
  task = task_gbcs,
  learner = rf,
  resampling = hout,
  measure = measure,
  search_space = search_space,
  terminator = evalsTerm
)
```

```
#Type of optimization
tuner = tnr("grid_search", resolution = 5)
#Start the tuning
tuner$optimize(instance)
```

```
## INFO  [09:48:56.491] [bbotk] Starting to optimize 3 parameter(s) with '<OptimizerGridSearch>' and '<
## INFO  [09:48:56.495] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:48:56.581] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:48:56.594] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:49:01.770] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:49:07.083] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:49:12.911] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:49:18.137] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:49:22.952] [mlr3]  Finished benchmark
## INFO  [09:49:23.064] [bbotk] Result of batch 1:
## INFO  [09:49:23.066] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:49:23.066] [bbotk]              6    8             0.7      0.7186124
## INFO  [09:49:23.066] [bbotk]                                    uhash
## INFO  [09:49:23.066] [bbotk]  5ea0e6f1-5a83-4ca8-a3e4-5b363945a5b2
## INFO  [09:49:23.067] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:49:23.128] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:49:23.137] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:49:26.818] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:49:30.859] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:49:34.798] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
```

```
## INFO  [09:49:38.775] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:49:42.455] [mlr3]  Finished benchmark
## INFO  [09:49:42.535] [bbotk] Result of batch 2:
## INFO  [09:49:42.537] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:49:42.537] [bbotk]              5    4             0.5      0.7319337
## INFO  [09:49:42.537] [bbotk]                                    uhash
## INFO  [09:49:42.537] [bbotk]  5e6d6eb1-600c-4b80-82ff-93a029807f1d
## INFO  [09:49:42.539] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:49:42.600] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:49:42.608] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:49:46.863] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:49:50.692] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:49:55.131] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:49:59.125] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:50:03.389] [mlr3]  Finished benchmark
## INFO  [09:50:03.478] [bbotk] Result of batch 3:
## INFO  [09:50:03.479] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:50:03.479] [bbotk]              2    4             0.5      0.7221642
## INFO  [09:50:03.479] [bbotk]                                    uhash
## INFO  [09:50:03.479] [bbotk]  43459bb1-8911-4b05-ab05-dbd807041aa0
## INFO  [09:50:03.481] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:50:03.564] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:50:03.573] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:50:09.838] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:50:15.554] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:50:21.256] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:50:27.451] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:50:33.927] [mlr3]  Finished benchmark
## INFO  [09:50:34.031] [bbotk] Result of batch 4:
## INFO  [09:50:34.034] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:50:34.034] [bbotk]              1    8             0.7       0.703284
## INFO  [09:50:34.034] [bbotk]                                    uhash
## INFO  [09:50:34.034] [bbotk]  51619f50-47eb-426e-b1ab-7e633362cf46
## INFO  [09:50:34.037] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:50:34.120] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:50:34.131] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:50:39.237] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:50:44.157] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:50:49.720] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:50:54.931] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:51:02.296] [mlr3]  Finished benchmark
## INFO  [09:51:02.485] [bbotk] Result of batch 5:
## INFO  [09:51:02.490] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:51:02.490] [bbotk]              5   10             0.6      0.7163996
## INFO  [09:51:02.490] [bbotk]                                    uhash
## INFO  [09:51:02.490] [bbotk]  32b26472-bc11-4529-8b9c-41cdc1b77c51
## INFO  [09:51:02.493] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:51:02.612] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:51:02.634] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:51:08.899] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:51:14.314] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:51:20.442] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:51:25.863] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:51:30.311] [mlr3]  Finished benchmark
```

```
## INFO  [09:51:30.396] [bbotk] Result of batch 6:
## INFO  [09:51:30.398] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:51:30.398] [bbotk]             1    4            0.55      0.7086077
## INFO  [09:51:30.398] [bbotk]                                uhash
## INFO  [09:51:30.398] [bbotk]  a551742a-85b8-4af1-ac94-5b60e2b20677
## INFO  [09:51:30.400] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:51:30.462] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:51:30.470] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:51:34.610] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:51:39.298] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:51:43.522] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:51:47.777] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:51:52.076] [mlr3]  Finished benchmark
## INFO  [09:51:52.160] [bbotk] Result of batch 7:
## INFO  [09:51:52.161] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:51:52.161] [bbotk]             3    8             0.5      0.7107238
## INFO  [09:51:52.161] [bbotk]                                uhash
## INFO  [09:51:52.161] [bbotk]  fb7061ba-5745-46b1-8671-8a1caa2163b0
## INFO  [09:51:52.163] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:51:52.226] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:51:52.234] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:51:56.509] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:52:00.387] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:52:04.283] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:52:08.717] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:52:14.448] [mlr3]  Finished benchmark
## INFO  [09:52:14.663] [bbotk] Result of batch 8:
## INFO  [09:52:14.665] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:52:14.665] [bbotk]             6    4             0.7      0.7320381
## INFO  [09:52:14.665] [bbotk]                                uhash
## INFO  [09:52:14.665] [bbotk]  a5d93155-5a0a-4603-b6ee-b1411aa7c8a7
## INFO  [09:52:14.668] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:52:14.779] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:52:14.794] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:52:19.436] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:52:24.539] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:52:29.078] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:52:34.665] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:52:39.422] [mlr3]  Finished benchmark
## INFO  [09:52:39.514] [bbotk] Result of batch 9:
## INFO  [09:52:39.516] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:52:39.516] [bbotk]             6    6            0.65      0.7264181
## INFO  [09:52:39.516] [bbotk]                                uhash
## INFO  [09:52:39.516] [bbotk]  b6713b29-ae89-45a7-b323-757c9d7c2f75
## INFO  [09:52:39.517] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:52:39.580] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:52:39.592] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:52:44.408] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:52:50.527] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:52:55.277] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:53:00.116] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:53:04.312] [mlr3]  Finished benchmark
## INFO  [09:53:04.422] [bbotk] Result of batch 10:
## INFO  [09:53:04.424] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
```

```
## INFO  [09:53:04.424] [bbotk]              3    6            0.65      0.7093923
## INFO  [09:53:04.424] [bbotk]                                  uhash
## INFO  [09:53:04.424] [bbotk]  cce6a136-ed0f-4b87-b7f4-e5f4e54fb216
## INFO  [09:53:04.426] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:53:04.491] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:53:04.499] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:53:09.290] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:53:14.035] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:53:19.367] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:53:26.074] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:53:31.767] [mlr3]  Finished benchmark
## INFO  [09:53:31.883] [bbotk] Result of batch 11:
## INFO  [09:53:31.886] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:53:31.886] [bbotk]              6   10             0.7      0.7165585
## INFO  [09:53:31.886] [bbotk]                                  uhash
## INFO  [09:53:31.886] [bbotk]  caa08210-8d97-412a-bbd1-f4796bd72ddd
## INFO  [09:53:31.889] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:53:31.976] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:53:31.986] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:53:36.420] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:53:41.526] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:53:51.281] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:54:01.228] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:54:06.473] [mlr3]  Finished benchmark
## INFO  [09:54:06.593] [bbotk] Result of batch 12:
## INFO  [09:54:06.596] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:54:06.596] [bbotk]              5    4            0.55      0.7291968
## INFO  [09:54:06.596] [bbotk]                                  uhash
## INFO  [09:54:06.596] [bbotk]  4e48e6bc-fb30-4874-a48b-b2e4c7977022
## INFO  [09:54:06.598] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:54:06.703] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:54:06.716] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:54:10.586] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:54:14.027] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:54:17.485] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:54:22.535] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:54:27.029] [mlr3]  Finished benchmark
## INFO  [09:54:27.151] [bbotk] Result of batch 13:
## INFO  [09:54:27.153] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:54:27.153] [bbotk]              6    2            0.55      0.7413947
## INFO  [09:54:27.153] [bbotk]                                  uhash
## INFO  [09:54:27.153] [bbotk]  7e723da7-0126-4a52-927a-e8677f308c2d
## INFO  [09:54:27.155] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:54:27.216] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:54:27.224] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:54:31.761] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:54:36.499] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:54:41.484] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:54:46.064] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:54:51.475] [mlr3]  Finished benchmark
## INFO  [09:54:51.555] [bbotk] Result of batch 14:
## INFO  [09:54:51.557] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:54:51.557] [bbotk]              5    8             0.5      0.7225045
## INFO  [09:54:51.557] [bbotk]                                  uhash
```

```
## INFO  [09:54:51.557] [bbotk]  ea7d6eb0-d1ad-41fa-8d1b-a3d633fe024b
## INFO  [09:54:51.559] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:54:51.623] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:54:51.632] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:54:58.706] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:55:03.247] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:55:08.165] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:55:14.875] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:55:20.467] [mlr3]  Finished benchmark
## INFO  [09:55:20.558] [bbotk] Result of batch 15:
## INFO  [09:55:20.560] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:55:20.560] [bbotk]              3   10            0.55      0.7081625
## INFO  [09:55:20.560] [bbotk]                                  uhash
## INFO  [09:55:20.560] [bbotk]  025283da-6452-4f4c-b2be-d4fe50731ba0
## INFO  [09:55:20.563] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:55:20.631] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:55:20.644] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:55:24.911] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:55:29.119] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:55:34.237] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:55:39.705] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:55:44.596] [mlr3]  Finished benchmark
## INFO  [09:55:44.683] [bbotk] Result of batch 16:
## INFO  [09:55:44.686] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:55:44.686] [bbotk]              6    2            0.65      0.7385701
## INFO  [09:55:44.686] [bbotk]                                  uhash
## INFO  [09:55:44.686] [bbotk]  9891f24b-d108-469c-a20d-8b8e17e6dfeb
## INFO  [09:55:44.689] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:55:44.771] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:55:44.784] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:55:50.485] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:55:56.091] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:56:00.401] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:56:05.773] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:56:11.810] [mlr3]  Finished benchmark
## INFO  [09:56:11.927] [bbotk] Result of batch 17:
## INFO  [09:56:11.934] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:56:11.934] [bbotk]              3    8             0.6      0.7103653
## INFO  [09:56:11.934] [bbotk]                                  uhash
## INFO  [09:56:11.934] [bbotk]  da3406e2-59e5-4dcc-acd7-d794ea1e7e7a
## INFO  [09:56:11.954] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:56:12.038] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:56:12.048] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:56:16.822] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:56:24.204] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:56:28.330] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:56:32.587] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:56:38.016] [mlr3]  Finished benchmark
## INFO  [09:56:38.138] [bbotk] Result of batch 18:
## INFO  [09:56:38.141] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:56:38.141] [bbotk]              2    2            0.55      0.7309817
## INFO  [09:56:38.141] [bbotk]                                  uhash
## INFO  [09:56:38.141] [bbotk]  e041253d-9b05-4b57-8079-e503f634eab4
## INFO  [09:56:38.144] [bbotk] Evaluating 1 configuration(s)
```

```
## INFO  [09:56:38.261] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:56:38.277] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:56:43.217] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:56:48.089] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:56:53.238] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:57:00.323] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:57:06.478] [mlr3]  Finished benchmark
## INFO  [09:57:06.579] [bbotk] Result of batch 19:
## INFO  [09:57:06.582] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:57:06.582] [bbotk]              2    6             0.6      0.7072376
## INFO  [09:57:06.582] [bbotk]                                 uhash
## INFO  [09:57:06.582] [bbotk]  a58fa124-72c1-4d9d-a7ae-80167465ee5c
## INFO  [09:57:06.585] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:57:06.661] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:57:06.670] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:57:10.988] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:57:15.501] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:57:19.952] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:57:24.948] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:57:29.219] [mlr3]  Finished benchmark
## INFO  [09:57:29.332] [bbotk] Result of batch 20:
## INFO  [09:57:29.334] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:57:29.334] [bbotk]              3    4             0.5      0.7203543
## INFO  [09:57:29.334] [bbotk]                                 uhash
## INFO  [09:57:29.334] [bbotk]  0cd276c6-67f7-4244-8746-6c1a698a2008
## INFO  [09:57:29.337] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:57:29.412] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:57:29.421] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:57:34.079] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:57:38.150] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:57:42.291] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:57:46.273] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:57:50.339] [mlr3]  Finished benchmark
## INFO  [09:57:50.434] [bbotk] Result of batch 21:
## INFO  [09:57:50.436] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:57:50.436] [bbotk]              5    2             0.6      0.7371803
## INFO  [09:57:50.436] [bbotk]                                 uhash
## INFO  [09:57:50.436] [bbotk]  6f11fa44-0b6f-47d4-83cf-474554215f19
## INFO  [09:57:50.438] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:57:50.512] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:57:50.524] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:57:55.839] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
## INFO  [09:58:00.787] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:58:06.237] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:58:12.371] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:58:17.000] [mlr3]  Finished benchmark
## INFO  [09:58:17.096] [bbotk] Result of batch 22:
## INFO  [09:58:17.098] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:58:17.098] [bbotk]              5    8             0.6      0.7167324
## INFO  [09:58:17.098] [bbotk]                                 uhash
## INFO  [09:58:17.098] [bbotk]  cab8e2b1-127e-4827-87e1-594defa314aa
## INFO  [09:58:17.100] [bbotk] Evaluating 1 configuration(s)
## INFO  [09:58:17.168] [mlr3]  Running benchmark with 5 resampling iterations
## INFO  [09:58:17.178] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 5/5)
```

```
## INFO  [09:58:22.006] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 1/5)
## INFO  [09:58:26.689] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 2/5)
## INFO  [09:58:31.414] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 3/5)
## INFO  [09:58:36.307] [mlr3]  Applying learner 'surv.ranger' on task 'train_gbcs' (iter 4/5)
## INFO  [09:58:41.491] [mlr3]  Finished benchmark
## INFO  [09:58:41.593] [bbotk] Result of batch 23:
## INFO  [09:58:41.595] [bbotk]  min.node.size mtry sample.fraction surv.harrell_c
## INFO  [09:58:41.595] [bbotk]              5   10             0.5      0.7204578
## INFO  [09:58:41.595] [bbotk]                                 uhash
## INFO  [09:58:41.595] [bbotk]  fe83ec66-7290-41e9-8091-2d3e88201b7e
## INFO  [09:58:41.603] [bbotk] Finished optimizing after 23 evaluation(s)
## INFO  [09:58:41.604] [bbotk] Result:
## INFO  [09:58:41.606] [bbotk]  min.node.size mtry sample.fraction learner_param_vals  x_domain surv.ha
## INFO  [09:58:41.606] [bbotk]              6    2            0.55          <list[4]> <list[3]>      0
```

```
##    min.node.size mtry sample.fraction learner_param_vals  x_domain
## 1:             6    2            0.55          <list[4]> <list[3]>
##    surv.harrell_c
## 1:      0.7413947
```

```
instance$result_learner_param_vals
```

```
## $num.threads
## [1] 1
##
## $min.node.size
## [1] 6
##
## $mtry
## [1] 2
##
## $sample.fraction
## [1] 0.55
```

```
instance$result_y
```

```
## surv.harrell_c
##      0.7413947
```

```
as.data.table(instance$archive)
```

```
##     min.node.size mtry sample.fraction surv.harrell_c
##  1:             6    8            0.70      0.7186124
##  2:             5    4            0.50      0.7319337
##  3:             2    4            0.50      0.7221642
##  4:             1    8            0.70      0.7032840
##  5:             5   10            0.60      0.7163996
##  6:             1    4            0.55      0.7086077
##  7:             3    8            0.50      0.7107238
##  8:             6    4            0.70      0.7320381
##  9:             6    6            0.65      0.7264181
## 10:             3    6            0.65      0.7093923
## 11:             6   10            0.70      0.7165585
## 12:             5    4            0.55      0.7291968
## 13:             6    2            0.55      0.7413947
## 14:             5    8            0.50      0.7225045
## 15:             3   10            0.55      0.7081625
```

```
## 16:              6    2            0.65      0.7385701
## 17:              3    8            0.60      0.7103653
## 18:              2    2            0.55      0.7309817
## 19:              2    6            0.60      0.7072376
## 20:              3    4            0.50      0.7203543
## 21:              5    2            0.60      0.7371803
## 22:              5    8            0.60      0.7167324
## 23:              5   10            0.50      0.7204578
##     min.node.size mtry sample.fraction surv.harrell_c
##                                    uhash           timestamp batch_nr
##  1: 5ea0e6f1-5a83-4ca8-a3e4-5b363945a5b2 2021-06-01 09:49:23        1
##  2: 5e6d6eb1-600c-4b80-82ff-93a029807f1d 2021-06-01 09:49:42        2
##  3: 43459bb1-8911-4b05-ab05-dbd807041aa0 2021-06-01 09:50:03        3
##  4: 51619f50-47eb-426e-b1ab-7e633362cf46 2021-06-01 09:50:34        4
##  5: 32b26472-bc11-4529-8b9c-41cdc1b77c51 2021-06-01 09:51:02        5
##  6: a551742a-85b8-4af1-ac94-5b60e2b20677 2021-06-01 09:51:30        6
##  7: fb7061ba-5745-46b1-8671-8a1caa2163b0 2021-06-01 09:51:52        7
##  8: a5d93155-5a0a-4603-b6ee-b1411aa7c8a7 2021-06-01 09:52:14        8
##  9: b6713b29-ae89-45a7-b323-757c9d7c2f75 2021-06-01 09:52:39        9
## 10: cce6a136-ed0f-4b87-b7f4-e5f4e54fb216 2021-06-01 09:53:04       10
## 11: caa08210-8d97-412a-bbd1-f4796bd72ddd 2021-06-01 09:53:31       11
## 12: 4e48e6bc-fb30-4874-a48b-b2e4c7977022 2021-06-01 09:54:06       12
## 13: 7e723da7-0126-4a52-927a-e8677f308c2d 2021-06-01 09:54:27       13
## 14: ea7d6eb0-d1ad-41fa-8d1b-a3d633fe024b 2021-06-01 09:54:51       14
## 15: 025283da-6452-4f4c-b2be-d4fe50731ba0 2021-06-01 09:55:20       15
## 16: 9891f24b-d108-469c-a20d-8b8e17e6dfeb 2021-06-01 09:55:44       16
## 17: da3406e2-59e5-4dcc-acd7-d794ea1e7e7a 2021-06-01 09:56:11       17
## 18: e041253d-9b05-4b57-8079-e503f634eab4 2021-06-01 09:56:38       18
## 19: a58fa124-72c1-4d9d-a7ae-80167465ee5c 2021-06-01 09:57:06       19
## 20: 0cd276c6-67f7-4244-8746-6c1a698a2008 2021-06-01 09:57:29       20
## 21: 6f11fa44-0b6f-47d4-83cf-474554215f19 2021-06-01 09:57:50       21
## 22: cab8e2b1-127e-4827-87e1-594defa314aa 2021-06-01 09:58:17       22
## 23: fe83ec66-7290-41e9-8091-2d3e88201b7e 2021-06-01 09:58:41       23
##                                    uhash           timestamp batch_nr
##     x_domain_min.node.size x_domain_mtry x_domain_sample.fraction
##  1:                      6            8                      0.70
##  2:                      5            4                      0.50
##  3:                      2            4                      0.50
##  4:                      1            8                      0.70
##  5:                      5           10                      0.60
##  6:                      1            4                      0.55
##  7:                      3            8                      0.50
##  8:                      6            4                      0.70
##  9:                      6            6                      0.65
## 10:                      3            6                      0.65
## 11:                      6           10                      0.70
## 12:                      5            4                      0.55
## 13:                      6            2                      0.55
## 14:                      5            8                      0.50
## 15:                      3           10                      0.55
## 16:                      6            2                      0.65
## 17:                      3            8                      0.60
## 18:                      2            2                      0.55
## 19:                      2            6                      0.60
```

```
## 20:                       3             4                       0.50
## 21:                       5             2                       0.60
## 22:                       5             8                       0.60
## 23:                       5            10                       0.50
##     x_domain_min.node.size x_domain_mtry x_domain_sample.fraction
```

```
#Setting the best parameters to the learner
rf$param_set$values = instance$result_learner_param_vals
#Retraining the learner
rf$train(task_gbcs)
```

```
rf$oob_error()
```

```
## [1] 0.2701964
```

```
rf$model # (0.7244294)
```

```
## Ranger result
##
## Call:
##  ranger::ranger(formula = NULL, dependent.variable.name = targets[1L],      status.variable.name = ta
##
## Type:                             Survival
## Number of trees:                  500
## Sample size:                      548
## Number of independent variables:  10
## Mtry:                             2
## Target node size:                 6
## Variable importance mode:         none
## Splitrule:                        logrank
## Number of unique death times:     467
## OOB prediction error (1-C):       0.2701964
```

```
rf.pred <- rf$predict(test_gbcs)
rf.pred$score()
```

```
## surv.harrell_c
##      0.7199211
```