R Notebook

Parametros:

Measure = Matthews correlation coefficient
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure
Filter keys = NULL
Filter values = NULL

```r
library("scmamp")
library(dplyr)
```

Tratamento dos dados

Carregando data set compilado

```r
ds = read.csv("/home/rodrigo/Dropbox/UNICAMP/IC/estudo_cost_learning/SummaryResults/summary_compilation

ds = filter(ds, learner != "classif.rusboost")
summary(ds)
```

```
##                 learner      weight_space
##  classif.ksvm        :17100   Mode :logical
##  classif.randomForest:17100   FALSE:41040
##  classif.rusboost    :    0   TRUE :10260
##  classif.xgboost     :17100   NA's :0
##
##
##
##                              measure        sampling     underbagging
##  Accuracy                        :10260   ADASYN:10260   Mode :logical
##  Area under the curve            :10260   FALSE :30780   FALSE:41040
##  F1 measure                      :10260   SMOTE :10260   TRUE :10260
##  G-mean                          :10260                  NA's :0
##  Matthews correlation coefficient:10260
##
##
##  tuning_measure    holdout_measure   holdout_measure_residual
##  Min.   :-0.1277   Min.   :-0.2120   Min.   :-0.4658
##  1st Qu.: 0.6911   1st Qu.: 0.4001   1st Qu.: 0.1994
##  Median : 0.9700   Median : 0.8571   Median : 0.5581
##  Mean   : 0.7903   Mean   : 0.6718   Mean   : 0.5298
##  3rd Qu.: 0.9975   3rd Qu.: 0.9900   3rd Qu.: 0.8755
##  Max.   : 1.0000   Max.   : 1.0000   Max.   : 1.0000
##  NA's   :1077      NA's   :1077      NA's   :1077
##  iteration_count                      dataset        imba.rate
##  Min.   :1       abalone                  :  900   Min.   :0.0010
##  1st Qu.:1       adult                    :  900   1st Qu.:0.0100
##  Median :2       bank                     :  900   Median :0.0300
##  Mean   :2       car                      :  900   Mean   :0.0286
##  3rd Qu.:3       cardiotocography-10clases:  900   3rd Qu.:0.0500
##  Max.   :3       cardiotocography-3clases :  900   Max.   :0.0500
##  NA's   :1077    (Other)                  :45900
```

Filtrando pela metrica

```r
ds = filter(ds, measure == params$measure)
```

Filtrando o data set

```r
if(params$filter_keys != 'NULL' && !is.null(params$filter_keys)){
  ds = filter_at(ds, .vars = params$filter_keys, .vars_predicate = any_vars(. == params$filter_values))
}

summary(ds)
```

```
##                 learner     weight_space
##  classif.ksvm        :3420   Mode :logical
##  classif.randomForest:3420   FALSE:8208
##  classif.rusboost    :   0   TRUE :2052
##  classif.xgboost     :3420   NA's :0
##
##
##
##                              measure        sampling    underbagging
##  Accuracy                        :    0   ADASYN:2052   Mode :logical
##  Area under the curve            :    0   FALSE :6156   FALSE:8208
##  F1 measure                      :    0   SMOTE :2052   TRUE :2052
##  G-mean                          :    0                 NA's :0
##  Matthews correlation coefficient:10260
##
##
##  tuning_measure    holdout_measure   holdout_measure_residual
##  Min.   :-0.1277   Min.   :-0.2120   Min.   :-0.46576
##  1st Qu.: 0.3307   1st Qu.: 0.0000   1st Qu.: 0.03886
##  Median : 0.8174   Median : 0.4907   Median : 0.21377
##  Mean   : 0.6548   Mean   : 0.4657   Mean   : 0.30966
##  3rd Qu.: 0.9890   3rd Qu.: 0.8152   3rd Qu.: 0.53139
##  Max.   : 1.0000   Max.   : 1.0000   Max.   : 1.00000
##  NA's   :225       NA's   :225       NA's   :225
##  iteration_count                      dataset       imba.rate
##  Min.   :1       abalone                   : 180   Min.   :0.0010
##  1st Qu.:1       adult                     : 180   1st Qu.:0.0100
##  Median :2       bank                      : 180   Median :0.0300
##  Mean   :2       car                       : 180   Mean   :0.0286
##  3rd Qu.:3       cardiotocography-10clases: 180   3rd Qu.:0.0500
##  Max.   :3       cardiotocography-3clases : 180   Max.   :0.0500
##  NA's   :225     (Other)                  :9180
```

Computando as médias das iteracoes

```r
ds = group_by(ds, learner , weight_space , measure , sampling , underbagging , dataset , imba.rate)
ds = summarise(ds, tuning_measure = mean(tuning_measure), holdout_measure = mean(holdout_measure),
          holdout_measure_residual = mean(holdout_measure_residual))

ds = as.data.frame(ds)
```

Criando dataframe

```
# Dividindo o ds em n, um para cada técnica
splited_df = ds %>% group_by_at(.vars = params$columns) %>% do(vals = as.data.frame(.)) %>% select(vals

# Juntando cada uma das partes horizontalmente em um data set
df_tec_wide = do.call("cbind", splited_df)

# Renomeando duplicacao de nomes
colnames(df_tec_wide) = make.unique(colnames(df_tec_wide))

# Selecionando apenas as medidas da performance escolhida
df_tec_wide_residual = select(df_tec_wide, matches(paste("^", params$performance, "$|", params$performan

# Renomeando colunas
new_names = NULL
for(i in (1:length(splited_df))){
  id = toString(sapply(splited_df[[i]][1, params$columns], as.character))
  new_names = c(new_names, id)
}
colnames(df_tec_wide_residual) = new_names

# Verificando a dimensao do df
dim(df_tec_wide_residual)
```

```
## [1] 228  15
```

```
# Renomeando a variavel
df = df_tec_wide_residual

summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :-0.06657
##  1st Qu.: 0.00000
##  Median : 0.16917
##  Mean   : 0.27983
##  3rd Qu.: 0.48522
##  Max.   : 1.00000
##  NA's   :7
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :-0.05198
##  1st Qu.: 0.23129
##  Median : 0.60454
##  Mean   : 0.55423
##  3rd Qu.: 0.89244
##  Max.   : 1.00000
##  NA's   :25
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :-0.06053                      Min.   :-0.04044
##  1st Qu.: 0.32126                      1st Qu.: 0.00000
##  Median : 0.69693                      Median : 0.19993
##  Mean   : 0.60211                      Mean   : 0.32962
##  3rd Qu.: 0.91772                      3rd Qu.: 0.63626
##  Max.   : 1.00000                      Max.   : 1.00000
##
```

```
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :-0.02449
##  1st Qu.: 0.08717
##  Median : 0.64741
##  Mean   : 0.53999
##  3rd Qu.: 0.87917
##  Max.   : 1.00000
##  NA's   :6
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :-0.03192                     Min.   :-0.0266
##  1st Qu.: 0.18970                     1st Qu.: 0.1289
##  Median : 0.66518                     Median : 0.3341
##  Mean   : 0.56276                     Mean   : 0.3896
##  3rd Qu.: 0.88876                     3rd Qu.: 0.6294
##  Max.   : 1.00000                     Max.   : 1.0000
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :-0.06927
##  1st Qu.: 0.20180
##  Median : 0.40785
##  Mean   : 0.44846
##  3rd Qu.: 0.66970
##  Max.   : 1.00000
##  NA's   :6
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :-0.06218                    Min.   :-0.03836
##  1st Qu.: 0.21955                    1st Qu.: 0.00000
##  Median : 0.37522                    Median : 0.19061
##  Mean   : 0.42007                    Mean   : 0.31993
##  3rd Qu.: 0.59865                    3rd Qu.: 0.63475
##  Max.   : 1.00000                    Max.   : 1.00000
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :-0.02177
##  1st Qu.: 0.12704
##  Median : 0.62299
##  Mean   : 0.54024
##  3rd Qu.: 0.89006
##  Max.   : 1.00000
##  NA's   :11
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :-0.01983                    Min.   :-0.05605
##  1st Qu.: 0.19181                    1st Qu.: 0.00000
##  Median : 0.66660                    Median : 0.17796
##  Mean   : 0.55830                    Mean   : 0.28889
##  3rd Qu.: 0.88419                    3rd Qu.: 0.55644
##  Max.   : 1.00000                    Max.   : 1.00000
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :-0.0748
##  1st Qu.: 0.1891
##  Median : 0.6299
##  Mean   : 0.5582
##  3rd Qu.: 0.9360
##  Max.   : 1.0000
##  NA's   :20
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :-0.03402
##  1st Qu.: 0.31873
##  Median : 0.70793
##  Mean   : 0.61125
##  3rd Qu.: 0.92233
##  Max.   : 1.00000
##
```

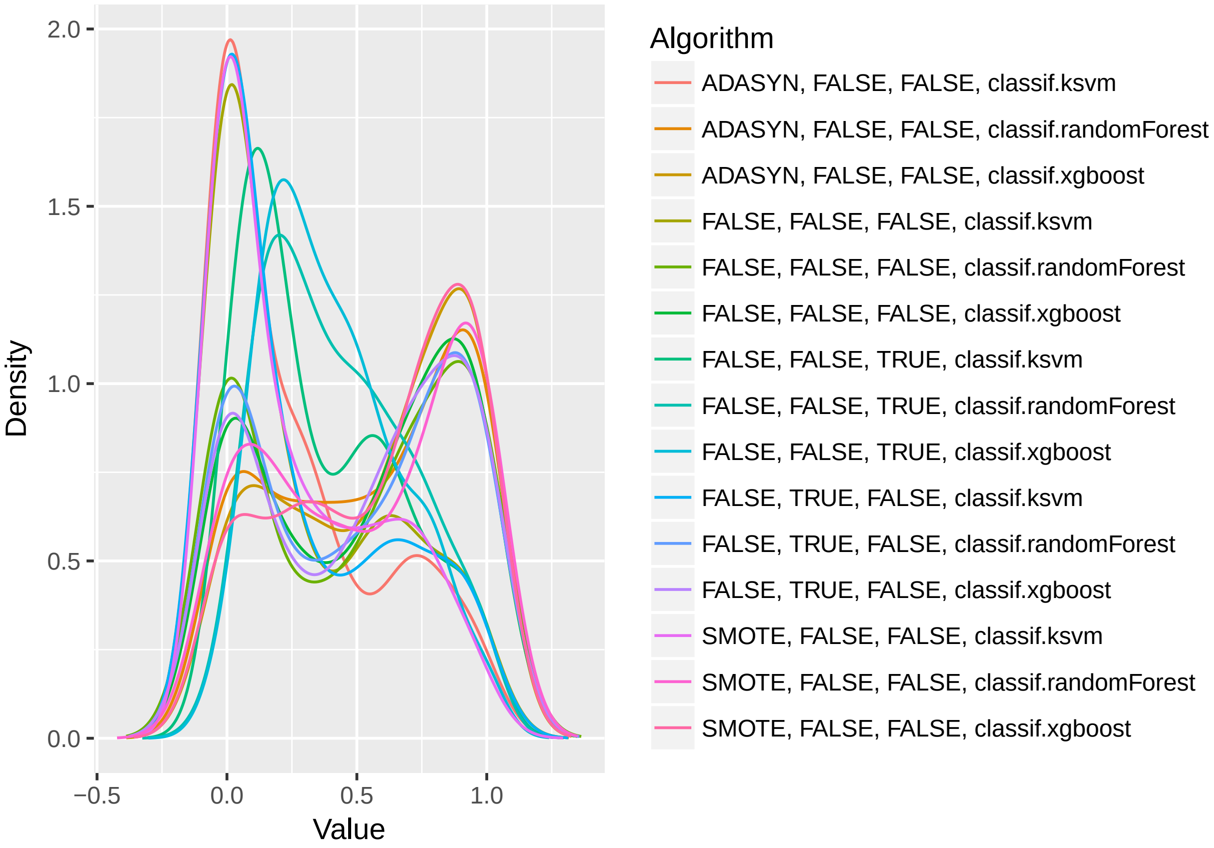
Verificando a média de cada coluna selecionada

```
for(i in (1:dim(df)[2])){
  #print(df[,i])
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.279829884673111"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.554228194573526"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.602108220680284"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.329621774060411"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.539987503010754"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.562761901495153"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.389559127733891"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.448457719824821"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.420065766646077"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.319931271475548"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.540235884982955"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.55829817503019"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.288890409106331"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.558193276234751"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.611248577729462"
```

Fazendo teste de normalidade

```
plotDensities(data = na.omit(df))
```



Testando as diferencas

```
friedmanTest(df)
```

```
##
##  Friedman's rank sum test
##
## data:  df
## Friedman's chi-squared = 705.58, df = 14, p-value < 2.2e-16
```

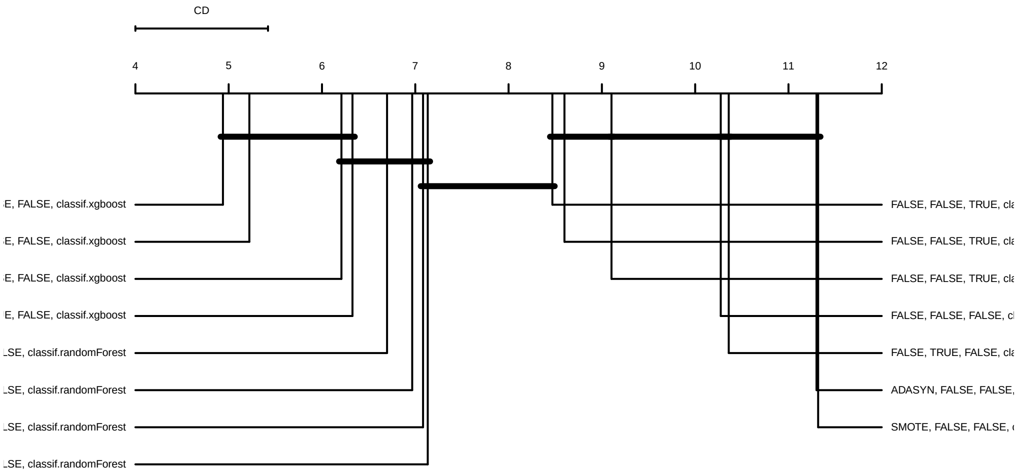
Testando as diferencas par a par

```
test <- nemenyiTest (df, alpha=0.05)
abs(test$diff.matrix) > test$statistic
```

```
##       ADASYN, FALSE, FALSE, classif.ksvm
##  [1,]                              FALSE
##  [2,]                               TRUE
##  [3,]                               TRUE
##  [4,]                              FALSE
##  [5,]                               TRUE
##  [6,]                               TRUE
##  [7,]                               TRUE
##  [8,]                               TRUE
##  [9,]                               TRUE
## [10,]                              FALSE
## [11,]                               TRUE
## [12,]                               TRUE
## [13,]                              FALSE
## [14,]                               TRUE
## [15,]                               TRUE
##       ADASYN, FALSE, FALSE, classif.randomForest
##  [1,]                                       TRUE
##  [2,]                                      FALSE
##  [3,]                                       TRUE
##  [4,]                                       TRUE
##  [5,]                                      FALSE
##  [6,]                                      FALSE
##  [7,]                                       TRUE
##  [8,]                                       TRUE
##  [9,]                                       TRUE
## [10,]                                       TRUE
## [11,]                                      FALSE
## [12,]                                      FALSE
## [13,]                                       TRUE
## [14,]                                      FALSE
## [15,]                                       TRUE
##       ADASYN, FALSE, FALSE, classif.xgboost
##  [1,]                                  TRUE
##  [2,]                                  TRUE
##  [3,]                                 FALSE
##  [4,]                                  TRUE
##  [5,]                                  TRUE
##  [6,]                                 FALSE
##  [7,]                                  TRUE
##  [8,]                                  TRUE
##  [9,]                                  TRUE
## [10,]                                  TRUE
## [11,]                                  TRUE
## [12,]                                 FALSE
## [13,]                                  TRUE
## [14,]                                  TRUE
## [15,]                                 FALSE
##       FALSE, FALSE, FALSE, classif.ksvm
##  [1,]                             FALSE
##  [2,]                              TRUE
##  [3,]                              TRUE
##  [4,]                             FALSE
##  [5,]                              TRUE
##  [6,]                              TRUE
##  [7,]                             FALSE
##  [8,]                              TRUE
##  [9,]                              TRUE
## [10,]                             FALSE
## [11,]                              TRUE
## [12,]                              TRUE
## [13,]                             FALSE
## [14,]                              TRUE
## [15,]                              TRUE
##       FALSE, FALSE, FALSE, classif.randomForest
##  [1,]                                      TRUE
##  [2,]                                     FALSE
##  [3,]                                      TRUE
##  [4,]                                      TRUE
##  [5,]                                     FALSE
##  [6,]                                     FALSE
##  [7,]                                      TRUE
##  [8,]                                     FALSE
##  [9,]                                      TRUE
## [10,]                                      TRUE
## [11,]                                     FALSE
## [12,]                                     FALSE
## [13,]                                      TRUE
## [14,]                                     FALSE
## [15,]                                      TRUE
##       FALSE, FALSE, FALSE, classif.xgboost
##  [1,]                                 TRUE
##  [2,]                                FALSE
##  [3,]                                FALSE
##  [4,]                                 TRUE
##  [5,]                                FALSE
##  [6,]                                FALSE
##  [7,]                                 TRUE
##  [8,]                                 TRUE
##  [9,]                                 TRUE
## [10,]                                 TRUE
## [11,]                                FALSE
## [12,]                                FALSE
## [13,]                                 TRUE
## [14,]                                FALSE
## [15,]                                FALSE
##       FALSE, FALSE, TRUE, classif.ksvm
##  [1,]                             TRUE
##  [2,]                             TRUE
##  [3,]                             TRUE
##  [4,]                            FALSE
##  [5,]                             TRUE
##  [6,]                             TRUE
##  [7,]                            FALSE
##  [8,]                            FALSE
##  [9,]                            FALSE
## [10,]                            FALSE
## [11,]                             TRUE
## [12,]                             TRUE
## [13,]                             TRUE
## [14,]                             TRUE
## [15,]                             TRUE
##       FALSE, FALSE, TRUE, classif.randomForest
##  [1,]                                     TRUE
##  [2,]                                     TRUE
##  [3,]                                     TRUE
##  [4,]                                     TRUE
##  [5,]                                    FALSE
##  [6,]                                     TRUE
##  [7,]                                    FALSE
##  [8,]                                    FALSE
##  [9,]                                    FALSE
## [10,]                                     TRUE
## [11,]                                    FALSE
## [12,]                                     TRUE
## [13,]                                     TRUE
## [14,]                                     TRUE
## [15,]                                     TRUE
##       FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  [1,]                                TRUE                            FALSE
##  [2,]                                TRUE                             TRUE
##  [3,]                                TRUE                             TRUE
##  [4,]                                TRUE                            FALSE
##  [5,]                                TRUE                             TRUE
##  [6,]                                TRUE                             TRUE
##  [7,]                               FALSE                            FALSE
##  [8,]                               FALSE                             TRUE
##  [9,]                               FALSE                             TRUE
## [10,]                                TRUE                            FALSE
## [11,]                                TRUE                             TRUE
## [12,]                                TRUE                             TRUE
## [13,]                                TRUE                            FALSE
## [14,]                                TRUE                             TRUE
## [15,]                                TRUE                             TRUE
##       FALSE, TRUE, FALSE, classif.randomForest
##  [1,]                                     TRUE
##  [2,]                                    FALSE
##  [3,]                                     TRUE
##  [4,]                                     TRUE
##  [5,]                                    FALSE
##  [6,]                                    FALSE
##  [7,]                                     TRUE
##  [8,]                                    FALSE
##  [9,]                                     TRUE
## [10,]                                     TRUE
## [11,]                                    FALSE
## [12,]                                    FALSE
## [13,]                                     TRUE
## [14,]                                    FALSE
## [15,]                                     TRUE
##       FALSE, TRUE, FALSE, classif.xgboost
##  [1,]                                TRUE
##  [2,]                               FALSE
##  [3,]                               FALSE
##  [4,]                                TRUE
##  [5,]                               FALSE
##  [6,]                               FALSE
##  [7,]                                TRUE
##  [8,]                                TRUE
##  [9,]                                TRUE
## [10,]                                  TRUE
## [11,]                                 FALSE
## [12,]                                 FALSE
## [13,]                                  TRUE
## [14,]                                 FALSE
## [15,]                                 FALSE
##       SMOTE, FALSE, FALSE, classif.ksvm
##  [1,]                               FALSE
##  [2,]                                TRUE
##  [3,]                                TRUE
##  [4,]                               FALSE
##  [5,]                                TRUE
##  [6,]                                TRUE
##  [7,]                                TRUE
##  [8,]                                TRUE
##  [9,]                                TRUE
## [10,]                               FALSE
## [11,]                                TRUE
## [12,]                                TRUE
## [13,]                               FALSE
## [14,]                                TRUE
## [15,]                                TRUE
##       SMOTE, FALSE, FALSE, classif.randomForest
##  [1,]                                        TRUE
##  [2,]                                       FALSE
##  [3,]                                        TRUE
##  [4,]                                        TRUE
##  [5,]                                       FALSE
##  [6,]                                       FALSE
##  [7,]                                        TRUE
##  [8,]                                        TRUE
##  [9,]                                        TRUE
## [10,]                                        TRUE
## [11,]                                       FALSE
## [12,]                                       FALSE
## [13,]                                        TRUE
## [14,]                                       FALSE
## [15,]                                        TRUE
##       SMOTE, FALSE, FALSE, classif.xgboost
##  [1,]                                   TRUE
##  [2,]                                   TRUE
##  [3,]                                  FALSE
##  [4,]                                   TRUE
##  [5,]                                   TRUE
##  [6,]                                  FALSE
##  [7,]                                   TRUE
##  [8,]                                   TRUE
##  [9,]                                   TRUE
## [10,]                                   TRUE
## [11,]                                   TRUE
## [12,]                                  FALSE
## [13,]                                   TRUE
## [14,]                                   TRUE
## [15,]                                  FALSE
```

Plotando grafico de Critical Diference

```r
result = tryCatch({
    plotCD(df, alpha=0.05, cex = 0.35)
}, error = function(e) {})
```


CD
E, FALSE, classif.xgboost
E, FALSE, classif.xgboost
E, FALSE, classif.xgboost
E, FALSE, classif.xgboost
LSE, classif.randomForest
LSE, classif.randomForest
LSE, classif.randomForest
LSE, classif.randomForest
FALSE, FALSE, TRUE, cla
FALSE, FALSE, TRUE, cla
FALSE, FALSE, TRUE, cla
FALSE, FALSE, FALSE, cl
FALSE, TRUE, FALSE, cla
ADASYN, FALSE, FALSE,
SMOTE, FALSE, FALSE,