R Notebook

Parametros:

Measure = Accuracy
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure_residual
Filter keys = NULL
Filter values = NULL

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

Tratamento dos dados

Carregando data set compilado

```
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                        :10260   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:    0
##
##
##  tuning_measure    holdout_measure   holdout_measure_residual
##  Min.   :0.09041   Min.   :0.01517   Min.   :0.0346
##  1st Qu.:0.96185   1st Qu.:0.95349   1st Qu.:0.3809
##  Median :0.98796   Median :0.98113   Median :0.7239
##  Mean   :0.95509   Mean   :0.94933   Mean   :0.6600
##  3rd Qu.:0.99669   3rd Qu.:0.99347   3rd Qu.:0.9428
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
##  NA's   :204       NA's   :204       NA's   :204
##  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   :204     (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

```r
# 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$performa

# 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
```

```r
# Renomeando a variavel
df = df_tec_wide_residual

summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.03682
##  1st Qu.:0.33545
##  Median :0.56831
##  Mean   :0.60784
##  3rd Qu.:0.93507
##  Max.   :0.99991
##  NA's   :7
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.03934
##  1st Qu.:0.40652
##  Median :0.73128
##  Mean   :0.67447
##  3rd Qu.:0.94400
##  Max.   :0.99987
##  NA's   :26
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.04525                      Min.   :0.0367
##  1st Qu.:0.44305                      1st Qu.:0.3107
##  Median :0.76046                      Median :0.5642
##  Mean   :0.69548                      Mean   :0.6038
##  3rd Qu.:0.95421                      3rd Qu.:0.9332
##  Max.   :0.99992                      Max.   :0.9999
##
```

```
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.06542
##  1st Qu.:0.33855
##  Median :0.69381
##  Mean   :0.64052
##  3rd Qu.:0.94948
##  Max.   :1.00000
##  NA's   :6
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.03977                      Min.   :0.04134
##  1st Qu.:0.36846                      1st Qu.:0.44847
##  Median :0.70059                      Median :0.67057
##  Mean   :0.65338                      Mean   :0.65147
##  3rd Qu.:0.96432                      3rd Qu.:0.86285
##  Max.   :0.99992                      Max.   :0.99926
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.2038
##  1st Qu.:0.6526
##  Median :0.8291
##  Mean   :0.7617
##  3rd Qu.:0.9300
##  Max.   :0.9998
##  NA's   :5
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.1649                      Min.   :0.0367
##  1st Qu.:0.6359                      1st Qu.:0.3107
##  Median :0.8215                      Median :0.5642
##  Mean   :0.7527                      Mean   :0.6013
##  3rd Qu.:0.9269                      3rd Qu.:0.9332
##  Max.   :0.9998                      Max.   :0.9999
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.06468
##  1st Qu.:0.34493
##  Median :0.68588
##  Mean   :0.63852
##  3rd Qu.:0.94766
##  Max.   :1.00000
##  NA's   :6
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.04244                     Min.   :0.03682
##  1st Qu.:0.36039                     1st Qu.:0.32220
##  Median :0.69388                     Median :0.54401
##  Mean   :0.65270                     Mean   :0.60250
##  3rd Qu.:0.96490                     3rd Qu.:0.93586
##  Max.   :1.00000                     Max.   :0.99992
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.04019
##  1st Qu.:0.39354
##  Median :0.73242
##  Mean   :0.66903
##  3rd Qu.:0.95289
##  Max.   :0.99992
##  NA's   :18
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.04523
##  1st Qu.:0.44211
##  Median :0.76059
##  Mean   :0.69633
##  3rd Qu.:0.94943
##  Max.   :1.00000
##
```

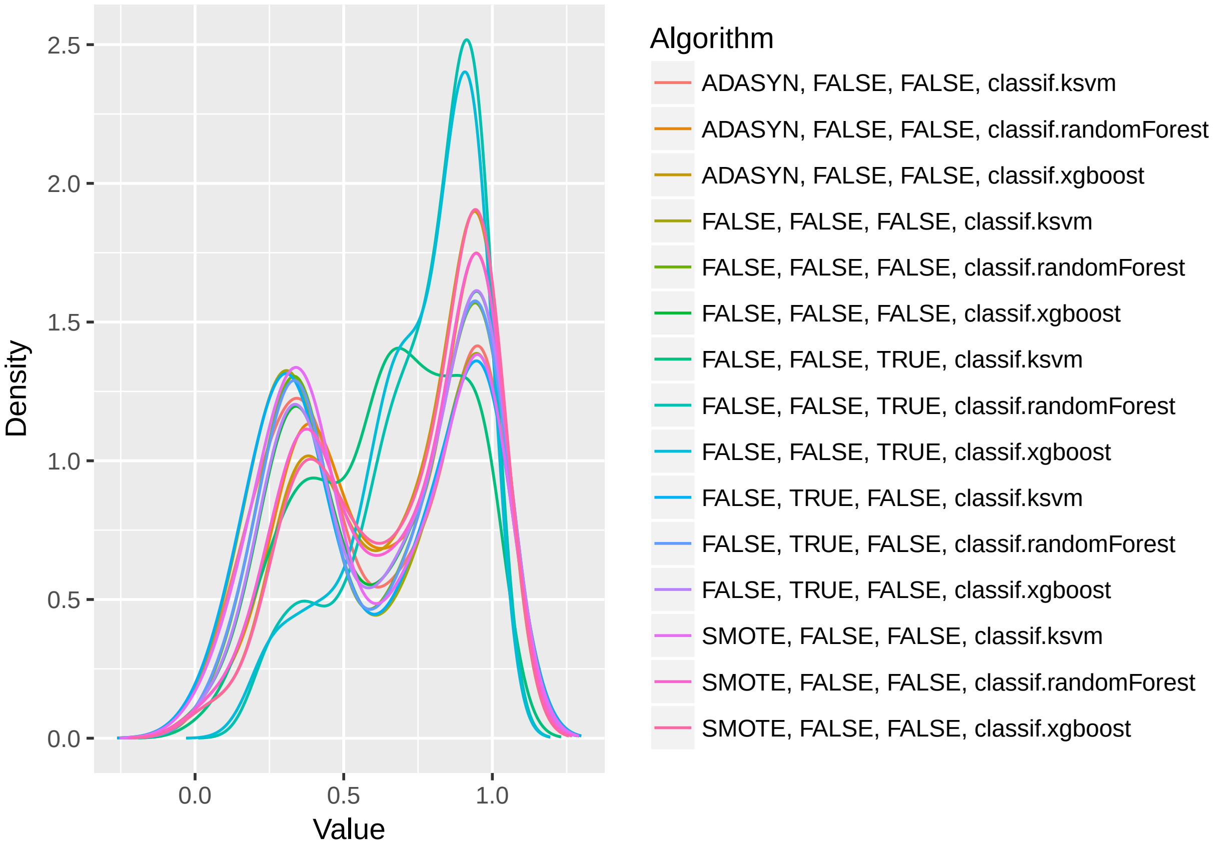
Verificando a média de cada coluna selecionada

```
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 = 0.607844620141704"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.674469188778052"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.695483910997827"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.603779074353433"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.640517319715074"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.653375584370522"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.651467926279361"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.76172069748065"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.752733911218448"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.60127866262381"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.638517379607029"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.652700570805191"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.602499475417883"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.669030316416214"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.6963314587541"
```

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 = 386.9, df = 14, p-value < 2.2e-16
```

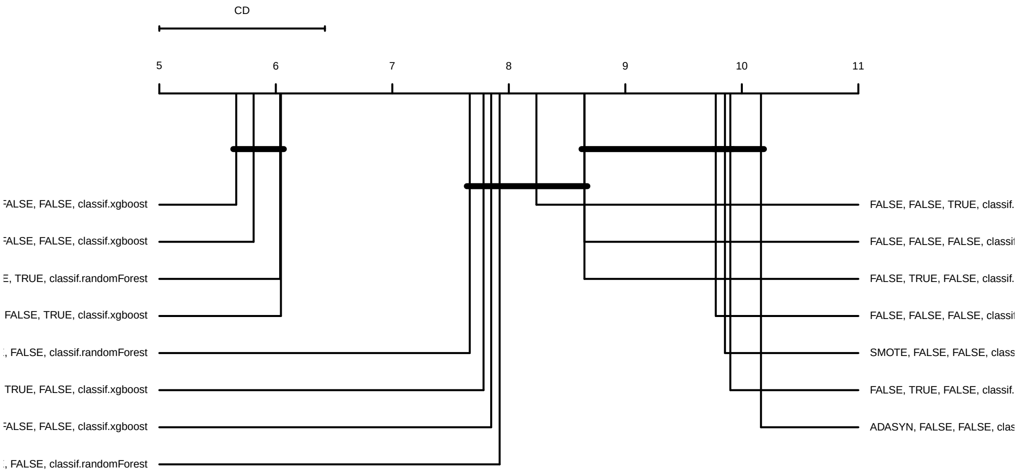
Testando as diferencas par a par

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

Plotando grafico de Critical Diference

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


CD
FALSE, FALSE, classif.xgboost
FALSE, FALSE, classif.xgboost
E, TRUE, classif.randomForest
FALSE, TRUE, classif.xgboost
, FALSE, classif.randomForest
TRUE, FALSE, classif.xgboost
FALSE, FALSE, classif.xgboost
, FALSE, classif.randomForest
FALSE, FALSE, TRUE, classif.
FALSE, FALSE, FALSE, classif
FALSE, TRUE, FALSE, classif.
FALSE, FALSE, FALSE, classif
SMOTE, FALSE, FALSE, class
FALSE, TRUE, FALSE, classif.
ADASYN, FALSE, FALSE, clas