R Notebook

Parametros:

Measure = G-mean
Columns = sampling, weight_space, underbagging, learner
Performance = tuning_measure
Filter keys = imba.rate
Filter values = 0.03

```
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

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

Filtrando o data set

```
if(params$filter_keys != 'NULL' && !is.null(params$filter_keys)){
  dots = paste0(params$filter_keys," == '",params$filter_values,"'")
  ds = filter_(ds, .dots = dots)
}

summary(ds)
```

```
##                  learner    weight_space
##  classif.ksvm        :990   Mode :logical
##  classif.randomForest:990   FALSE:2376
##  classif.rusboost    :  0   TRUE :594
##  classif.xgboost     :990   NA's :0
##
##
##
##                             measure       sampling    underbagging
##  Accuracy                       :   0   ADASYN: 594   Mode :logical
##  Area under the curve           :   0   FALSE :1782   FALSE:2376
##  F1 measure                     :   0   SMOTE : 594   TRUE :594
##  G-mean                         :2970                 NA's :0
##  Matthews correlation coefficient:   0
##
##
##  tuning_measure   holdout_measure  holdout_measure_residual
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.6338   1st Qu.:0.2132   1st Qu.:0.1828
##  Median :0.9453   Median :0.7348   Median :0.4920
##  Mean   :0.7583   Mean   :0.6032   Mean   :0.4882
##  3rd Qu.:0.9933   3rd Qu.:0.9533   3rd Qu.:0.8073
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  NA's   :48       NA's   :48       NA's   :48
##  iteration_count          dataset       imba.rate
##  Min.   :1       abalone      :  45   Min.   :0.03
##  1st Qu.:1       adult        :  45   1st Qu.:0.03
##  Median :2       annealing    :  45   Median :0.03
##  Mean   :2       arrhythmia   :  45   Mean   :0.03
##  3rd Qu.:3       balance-scale:  45   3rd Qu.:0.03
##  Max.   :3       bank         :  45   Max.   :0.03
##  NA's   :48      (Other)      :2700
```

Computando as médias das iteracoes

```
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$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] 66 15
```

```
# Renomeando a variavel
df = df_tec_wide_residual

head(df)
```

```
##   ADASYN, FALSE, FALSE, classif.ksvm
## 1                          0.9592389
## 2                          0.9748817
## 3                          0.9830574
## 4                          0.9190492
## 5                          1.0000000
## 6                          0.9849672
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.9565473
## 2                                         NA
## 3                                  0.9926273
## 4                                  0.9970436
## 5                                  1.0000000
## 6                                  0.9862039
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.9671729                          0.23019478
## 2                            0.9842448                          0.34575254
## 3                            0.9883318                          0.26604411
## 4                            0.9865916                          0.00000000
## 5                            1.0000000                          0.95922146
## 6                            0.9842647                          0.08803044
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                 0.0000000
## 2                                        NA
```

```
## 3                                0.6540313
## 4                                0.5037406
## 5                                1.0000000
## 6                                0.2017186
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                            0.0000000                         0.6590811
## 2                            0.5292628                         0.7556106
## 3                            0.5043107                         0.7592993
## 4                            0.5408550                         0.6059312
## 5                            1.0000000                         0.9674543
## 6                            0.2888779                         0.5458562
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                               0.6679185
## 2                               0.8315968
## 3                               0.8831278
## 4                               0.8911189
## 5                               0.9571436
## 6                               0.8060986
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.6628393                        0.18890575
## 2                           0.8368267                        0.34514464
## 3                           0.8440532                        0.26604411
## 4                           0.9006323                        0.00000000
## 5                           0.9620748                        0.95922146
## 6                           0.8018486                        0.08803044
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                               0.0000000
## 2                                      NA
## 3                               0.6204237
## 4                               0.6680696
## 5                               1.0000000
## 6                               0.2183992
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                          0.04583346                         0.9608493
## 2                          0.52669600                         0.9762996
## 3                          0.44017800                         0.9821256
## 4                          0.61329550                         0.8568056
## 5                          1.00000000                         1.0000000
## 6                          0.28985765                         0.9838950
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                               0.9548872
## 2                                      NA
## 3                               0.9929398
## 4                               0.9950259
## 5                               1.0000000
## 6                               0.9848749
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                           0.9659116
## 2                           0.9851010
## 3                           0.9880372
## 4                           0.9890509
## 5                           1.0000000
## 6                           0.9849288
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.7437
##  1st Qu.:0.9785
##  Median :0.9942
##  Mean   :0.9799
##  3rd Qu.:0.9977
##  Max.   :1.0000
##  NA's   :2
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.7372
##  1st Qu.:0.9829
##  Median :0.9964
##  Mean   :0.9857
##  3rd Qu.:0.9992
##  Max.   :1.0000
##  NA's   :7
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.7368                        Min.   :0.00000
##  1st Qu.:0.9762                        1st Qu.:0.01121
##  Median :0.9927                        Median :0.27026
##  Mean   :0.9807                        Mean   :0.33794
##  3rd Qu.:0.9982                        3rd Qu.:0.64915
##  Max.   :1.0000                        Max.   :0.96114
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.2017
##  Median :0.6758
##  Mean   :0.5755
##  3rd Qu.:0.8985
##  Max.   :1.0000
##  NA's   :1
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.0000                       Min.   :0.3521
##  1st Qu.:0.4237                       1st Qu.:0.5646
##  Median :0.6896                       Median :0.7481
##  Mean   :0.6295                       Mean   :0.7193
##  3rd Qu.:0.8910                       3rd Qu.:0.8388
##  Max.   :1.0000                       Max.   :0.9838
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.4493
##  1st Qu.:0.8140
##  Median :0.9163
##  Mean   :0.8706
##  3rd Qu.:0.9785
##  Max.   :1.0000
##  NA's   :1
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.5522                      Min.   :0.0000
##  1st Qu.:0.8058                      1st Qu.:0.0000
##  Median :0.9021                      Median :0.2380
```

```
##  Mean   :0.8565                     Mean   :0.3170
##  3rd Qu.:0.9660                     3rd Qu.:0.6230
##  Max.   :1.0000                     Max.   :0.9611
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.2055
##  Median :0.6768
##  Mean   :0.5717
##  3rd Qu.:0.9093
##  Max.   :1.0000
##  NA's   :2
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000                     Min.   :0.7440
##  1st Qu.:0.4604                     1st Qu.:0.9767
##  Median :0.7108                     Median :0.9923
##  Mean   :0.6323                     Mean   :0.9798
##  3rd Qu.:0.8997                     3rd Qu.:0.9976
##  Max.   :1.0000                     Max.   :1.0000
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.7425
##  1st Qu.:0.9831
##  Median :0.9960
##  Mean   :0.9866
##  3rd Qu.:0.9994
##  Max.   :1.0000
##  NA's   :3
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.7425
##  1st Qu.:0.9821
##  Median :0.9943
##  Mean   :0.9855
##  3rd Qu.:0.9981
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

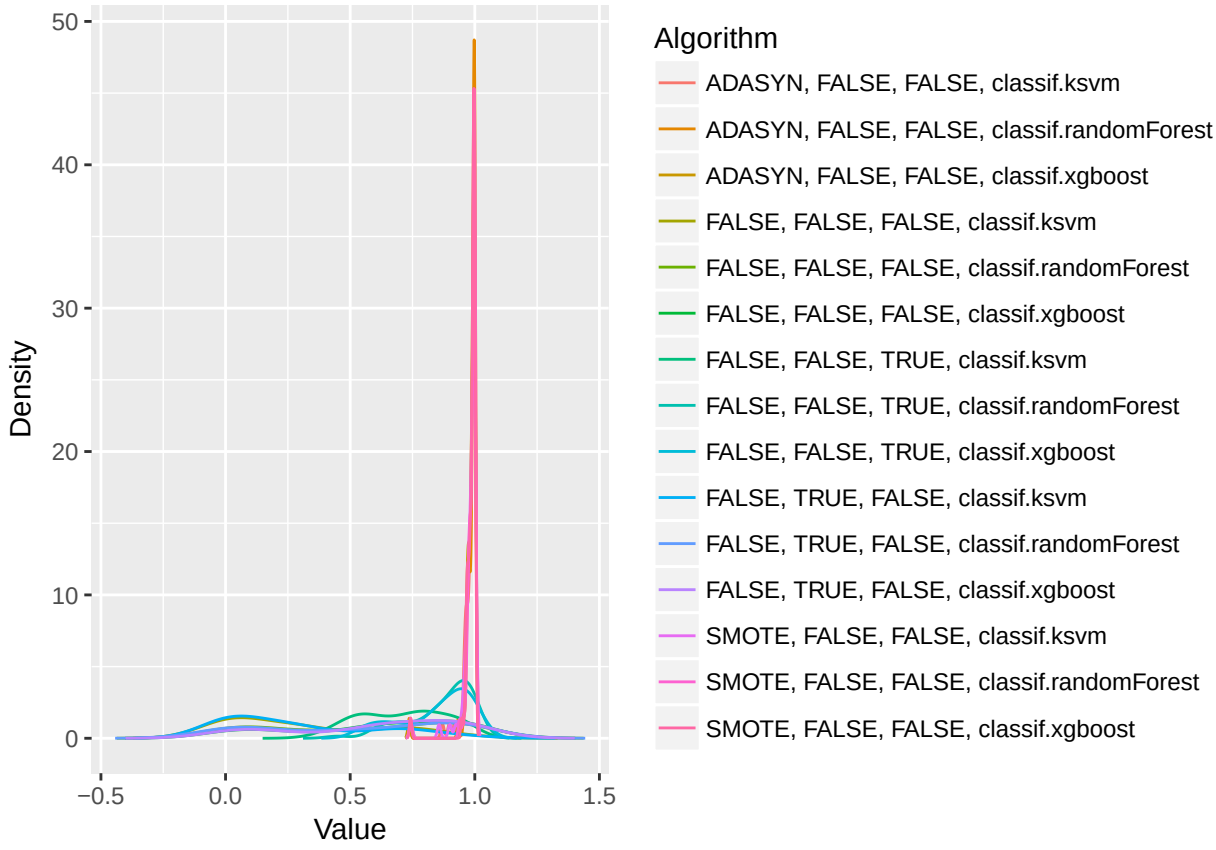
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.979917890040256"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.985713715356678"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.980725954597841"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.337943434843279"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.57549035347215"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.629513573662504"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.719299497389786"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.87060596098958"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.856470435913349"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.317003888490955"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.571678603997497"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.632332489120912"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.979810993858305"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.986599434491815"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.985471051278571"
```

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 = 624.35, 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,]                               FALSE
##  [3,]                               FALSE
##  [4,]                                TRUE
##  [5,]                                TRUE
##  [6,]                                TRUE
##  [7,]                                TRUE
##  [8,]                                TRUE
##  [9,]                                TRUE
## [10,]                                TRUE
## [11,]                                TRUE
## [12,]                                TRUE
## [13,]                               FALSE
## [14,]                               FALSE
## [15,]                               FALSE
##       ADASYN, FALSE, FALSE, classif.randomForest
##  [1,]                                      FALSE
##  [2,]                                      FALSE
##  [3,]                                      FALSE
##  [4,]                                       TRUE
##  [5,]                                       TRUE
##  [6,]                                       TRUE
##  [7,]                                       TRUE
##  [8,]                                       TRUE
##  [9,]                                       TRUE
## [10,]                                       TRUE
## [11,]                                       TRUE
## [12,]                                       TRUE
## [13,]                                      FALSE
## [14,]                                      FALSE
## [15,]                                      FALSE
##       ADASYN, FALSE, FALSE, classif.xgboost
##  [1,]                                 FALSE
##  [2,]                                 FALSE
##  [3,]                                 FALSE
##  [4,]                                  TRUE
##  [5,]                                  TRUE
##  [6,]                                  TRUE
##  [7,]                                  TRUE
##  [8,]                                  TRUE
##  [9,]                                  TRUE
## [10,]                                  TRUE
## [11,]                                  TRUE
## [12,]                                  TRUE
## [13,]                                 FALSE
## [14,]                                 FALSE
## [15,]                                 FALSE
```

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

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

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

```
##  [2,]                                FALSE
##  [3,]                                FALSE
##  [4,]                                 TRUE
##  [5,]                                 TRUE
##  [6,]                                 TRUE
##  [7,]                                 TRUE
##  [8,]                                 TRUE
##  [9,]                                 TRUE
## [10,]                                 TRUE
## [11,]                                 TRUE
## [12,]                                 TRUE
## [13,]                                FALSE
## [14,]                                FALSE
## [15,]                                FALSE
```

Plotando os ranks

```
print(colMeans(rankMatrix(df)))
```

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                   4.280303
## ADASYN, FALSE, FALSE, classif.randomForest
##                                   4.371212
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                   4.166667
##          FALSE, FALSE, FALSE, classif.ksvm
##                                  13.287879
##  FALSE, FALSE, FALSE, classif.randomForest
##                                  11.295455
##       FALSE, FALSE, FALSE, classif.xgboost
##                                  10.325758
##           FALSE, FALSE, TRUE, classif.ksvm
##                                  10.181818
##   FALSE, FALSE, TRUE, classif.randomForest
##                                   7.590909
##        FALSE, FALSE, TRUE, classif.xgboost
##                                   7.916667
##           FALSE, TRUE, FALSE, classif.ksvm
##                                  13.409091
##   FALSE, TRUE, FALSE, classif.randomForest
##                                  11.446970
##        FALSE, TRUE, FALSE, classif.xgboost
##                                  10.234848
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   3.674242
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                   3.818182
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                   4.000000
```

Plotando grafico de Critical Diference

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