R Notebook

Parametros:

Measure = G-mean
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_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                          0.1570779
## 2                          0.3325840
## 3                          0.3290109
## 4                          0.0000000
## 5                          0.6666667
## 6                          0.2665082
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.2696244
## 2                                         NA
## 3                                  0.8495315
## 4                                  0.0000000
## 5                                  1.0000000
## 6                                  0.3200615
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.1604060                          0.19012012
## 2                            0.5650640                          0.37952513
## 3                            0.6169629                          0.40088690
## 4                            0.7999249                          0.00000000
## 5                            1.0000000                          1.00000000
## 6                            0.3968667                          0.06782708
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                0.00000000
## 2                                        NA
```

```
## 3                                0.80054149
## 4                                0.66666667
## 5                                1.00000000
## 6                                0.06791288
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                           0.0000000                         0.6183473
## 2                           0.5331654                         0.7785217
## 3                           0.2348341                         0.8179164
## 4                           0.3313433                         0.2909572
## 5                           1.0000000                         0.9842296
## 6                           0.3093709                         0.5698472
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                                0.6228767
## 2                                0.8385555
## 3                                0.9432048
## 4                                0.9468065
## 5                                1.0000000
## 6                                0.8226052
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.6800788                        0.23641729
## 2                           0.8339146                        0.37433157
## 3                           0.9234272                        0.40088690
## 4                           0.9573321                        0.00000000
## 5                           0.9492242                        1.00000000
## 6                           0.8061809                        0.06782708
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                                0.0000000
## 2                                       NA
## 3                                0.6629766
## 4                                0.3293412
## 5                                1.0000000
## 6                                0.1354393
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                           0.0000000                         0.2337788
## 2                           0.5406144                         0.3582437
## 3                           0.1666667                         0.3314894
## 4                           0.2357023                         0.0000000
## 5                           1.0000000                         0.9023689
## 6                           0.1639916                         0.0000000
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                                 0.1919372
## 2                                        NA
## 3                                 0.6855182
## 4                                 0.0000000
## 5                                 1.0000000
## 6                                 0.3354144
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                           0.1601916
## 2                           0.5595411
## 3                           0.6853256
## 4                           0.2342951
## 5                           1.0000000
## 6                           0.1912967
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.3249
##  Mean   :0.3659
##  3rd Qu.:0.6939
##  Max.   :1.0000
##  NA's   :2
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.4646
##  Median :0.7662
##  Mean   :0.6638
##  3rd Qu.:0.9399
##  Max.   :1.0000
##  NA's   :7
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000                        Min.   :0.0000
##  1st Qu.:0.6395                        1st Qu.:0.0000
##  Median :0.8629                        Median :0.2865
##  Mean   :0.7453                        Mean   :0.3769
##  3rd Qu.:0.9701                        3rd Qu.:0.6916
##  Max.   :1.0000                        Max.   :1.0000
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.2357
##  Median :0.7071
##  Mean   :0.5844
##  3rd Qu.:0.9277
##  Max.   :1.0000
##  NA's   :1
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.0000                       Min.   :0.1968
##  1st Qu.:0.2815                       1st Qu.:0.5600
##  Median :0.7127                       Median :0.7410
##  Mean   :0.6047                       Mean   :0.7001
##  3rd Qu.:0.9082                       3rd Qu.:0.8724
##  Max.   :1.0000                       Max.   :1.0000
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.4979
##  1st Qu.:0.8176
##  Median :0.9168
##  Mean   :0.8771
##  3rd Qu.:0.9781
##  Max.   :1.0000
##  NA's   :1
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.4333                      Min.   :0.0000
##  1st Qu.:0.7861                      1st Qu.:0.0000
##  Median :0.9110                      Median :0.2361
```

```
##  Mean   :0.8532                     Mean   :0.3518
##  3rd Qu.:0.9688                     3rd Qu.:0.6834
##  Max.   :1.0000                     Max.   :1.0000
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.1901
##  Median :0.6596
##  Mean   :0.5669
##  3rd Qu.:0.9351
##  Max.   :1.0000
##  NA's   :2
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000                     Min.   :0.0000
##  1st Qu.:0.3421                     1st Qu.:0.0000
##  Median :0.7199                     Median :0.2997
##  Mean   :0.6100                     Mean   :0.3583
##  3rd Qu.:0.9292                     3rd Qu.:0.7080
##  Max.   :1.0000                     Max.   :0.9835
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.4589
##  Median :0.8047
##  Mean   :0.6630
##  3rd Qu.:0.9611
##  Max.   :1.0000
##  NA's   :3
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.0000
##  1st Qu.:0.5866
##  Median :0.8530
##  Mean   :0.7318
##  3rd Qu.:0.9505
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

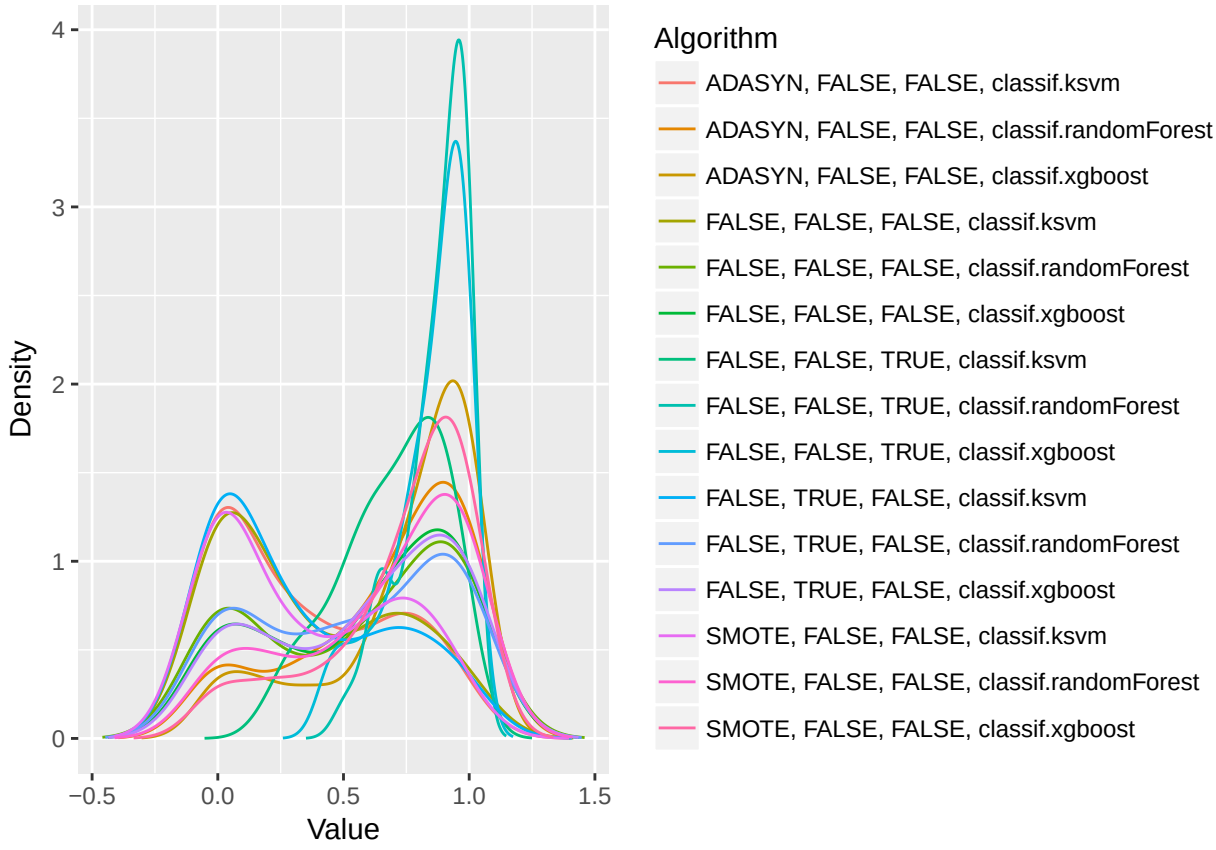
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.365902209629931"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.663843376110687"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.745340678566945"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.376889932514293"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.584378314900967"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.604699207162843"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.700107124176518"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.877095686653652"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.853152384473385"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.351818557486473"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.56692509017948"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.610049715923186"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.358349594822169"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.662962842206008"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.731777137568974"
```

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 = 400.22, 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,]                                      FALSE
##  [4,]                                       TRUE
##  [5,]                                      FALSE
##  [6,]                                      FALSE
##  [7,]                                      FALSE
##  [8,]                                       TRUE
##  [9,]                                       TRUE
## [10,]                                       TRUE
## [11,]                                      FALSE
## [12,]                                      FALSE
## [13,]                                       TRUE
## [14,]                                      FALSE
## [15,]                                      FALSE
##       ADASYN, FALSE, FALSE, classif.xgboost
##  [1,]                                  TRUE
##  [2,]                                 FALSE
##  [3,]                                 FALSE
##  [4,]                                  TRUE
##  [5,]                                  TRUE
##  [6,]                                  TRUE
##  [7,]                                 FALSE
##  [8,]                                 FALSE
##  [9,]                                 FALSE
## [10,]                                  TRUE
## [11,]                                  TRUE
## [12,]                                  TRUE
## [13,]                                  TRUE
## [14,]                                 FALSE
## [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,]                                      TRUE
## [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,]                            FALSE
##  [4,]                             TRUE
##  [5,]                            FALSE
```

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

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

Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                   12.060606
## ADASYN, FALSE, FALSE, classif.randomForest
##                                    7.492424
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                    5.143939
##          FALSE, FALSE, FALSE, classif.ksvm
##                                   11.325758
##  FALSE, FALSE, FALSE, classif.randomForest
##                                    8.954545
##       FALSE, FALSE, FALSE, classif.xgboost
##                                    8.196970
##           FALSE, FALSE, TRUE, classif.ksvm
##                                    7.159091
##   FALSE, FALSE, TRUE, classif.randomForest
##                                    2.863636
##        FALSE, FALSE, TRUE, classif.xgboost
##                                    3.643939
##           FALSE, TRUE, FALSE, classif.ksvm
##                                   11.568182
##   FALSE, TRUE, FALSE, classif.randomForest
##                                    9.196970
##        FALSE, TRUE, FALSE, classif.xgboost
##                                    7.878788
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   12.037879
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                    7.136364
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                    5.340909
```

Plotando grafico de Critical Diference

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