R Notebook

Parametros:

Measure = G-mean
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure_residual
Filter keys = imba.rate
Filter values = 0.05

```
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        :1230   Mode :logical
##  classif.randomForest:1230   FALSE:2952
##  classif.rusboost    :   0   TRUE :738
##  classif.xgboost     :1230   NA's :0
##
##
##
##                              measure       sampling    underbagging
##  Accuracy                        :   0   ADASYN: 738   Mode :logical
##  Area under the curve            :   0   FALSE :2214   FALSE:2952
##  F1 measure                      :   0   SMOTE : 738   TRUE :738
##  G-mean                          :3690                 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.:0.6329   1st Qu.:0.3162   1st Qu.:0.2321
##  Median :0.9254   Median :0.7412   Median :0.5564
##  Mean   :0.7606   Mean   :0.6130   Mean   :0.5202
##  3rd Qu.:0.9872   3rd Qu.:0.9487   3rd Qu.:0.8165
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  NA's   :39       NA's   :39       NA's   :39
##  iteration_count          dataset       imba.rate
##  Min.   :1       abalone      :  45   Min.   :0.05
##  1st Qu.:1       adult        :  45   1st Qu.:0.05
##  Median :2       annealing    :  45   Median :0.05
##  Mean   :2       arrhythmia   :  45   Mean   :0.05
##  3rd Qu.:3       balance-scale:  45   3rd Qu.:0.05
##  Max.   :3       bank         :  45   Max.   :0.05
##  NA's   :39      (Other)      :3420
```

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] 82 15
```

```
# Renomeando a variavel
df = df_tec_wide_residual

head(df)
```

```
##   ADASYN, FALSE, FALSE, classif.ksvm
## 1                          0.3591805
## 2                          0.4457936
## 3                          0.4942148
## 4                          0.0000000
## 5                          0.3703755
## 6                          0.1432253
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.3628625
## 2                                  0.6063279
## 3                                  0.8494581
## 4                                  0.2357023
## 5                                  0.2739035
## 6                                  0.5039727
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.2834615                          0.3217932
## 2                            0.6169366                          0.4872910
## 3                            0.8491008                          0.4519600
## 4                            0.5607107                          0.0000000
## 5                            0.3328651                          0.3978394
## 6                            0.4481851                          0.3240835
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                0.04668654
## 2                                0.55285818
```

```
## 3                                0.86954923
## 4                                0.69675874
## 5                                0.33185308
## 6                                0.32967244
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                           0.09187478                         0.6140664
## 2                           0.56575933                         0.7758681
## 3                           0.77940184                         0.7074974
## 4                           0.56330522                         0.5582570
## 5                           0.33286510                         0.4467726
## 6                           0.38747884                         0.6727911
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                                0.6408856
## 2                                       NA
## 3                                0.9144812
## 4                                0.8556430
## 5                                0.4304501
## 6                                0.8094284
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.6484918                         0.3179478
## 2                           0.8294728                         0.4669393
## 3                           0.8611455                         0.4865114
## 4                           0.9518122                         0.0000000
## 5                           0.5022607                         0.3978394
## 6                           0.8051700                         0.2500764
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                               0.04668654
## 2                               0.55652677
## 3                               0.85840972
## 4                               0.79226103
## 5                               0.31752864
## 6                               0.32967244
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                          0.07204807                         0.3728492
## 2                          0.55757016                         0.4536133
## 3                          0.72844276                         0.5098786
## 4                          0.56330522                         0.0000000
## 5                          0.33286510                         0.2378804
## 6                          0.37145103                         0.2171279
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                                 0.3630242
## 2                                        NA
## 3                                 0.8822864
## 4                                 0.4684571
## 5                                 0.2949278
## 6                                 0.4909869
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                            0.2731270
## 2                            0.6109004
## 3                            0.8247874
## 4                            0.7901410
## 5                            0.2851806
## 6                            0.4683657
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.00000
##  1st Qu.:0.07978
##  Median :0.27601
##  Mean   :0.33562
##  3rd Qu.:0.54361
##  Max.   :0.98958
##  NA's   :1
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.3444
##  Median :0.6163
##  Mean   :0.5797
##  3rd Qu.:0.8444
##  Max.   :0.9999
##  NA's   :3
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000                        Min.   :0.0000
##  1st Qu.:0.3473                        1st Qu.:0.0000
##  Median :0.6219                        Median :0.2390
##  Mean   :0.6101                        Mean   :0.2878
##  3rd Qu.:0.8630                        3rd Qu.:0.4438
##  Max.   :0.9999                        Max.   :1.0000
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.2255
##  Median :0.4749
##  Mean   :0.5081
##  3rd Qu.:0.7899
##  Max.   :0.9999
##  NA's   :1
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.0000                       Min.   :0.02295
##  1st Qu.:0.2368                       1st Qu.:0.44758
##  Median :0.5645                       Median :0.61845
##  Mean   :0.5301                       Mean   :0.60073
##  3rd Qu.:0.8117                       3rd Qu.:0.77313
##  Max.   :0.9999                       Max.   :0.99115
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.1800
##  1st Qu.:0.6256
##  Median :0.7864
##  Mean   :0.7498
##  3rd Qu.:0.9274
##  Max.   :0.9999
##  NA's   :3
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.1744                      Min.   :0.0000
##  1st Qu.:0.6174                      1st Qu.:0.0000
##  Median :0.7769                      Median :0.2379
```

```
##  Mean   :0.7375                     Mean   :0.2785
##  3rd Qu.:0.9241                     3rd Qu.:0.4203
##  Max.   :0.9999                     Max.   :1.0000
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.2335
##  Median :0.5441
##  Mean   :0.5258
##  3rd Qu.:0.8035
##  Max.   :1.0000
##  NA's   :1
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000                     Min.   :0.00000
##  1st Qu.:0.2404                     1st Qu.:0.09306
##  Median :0.5446                     Median :0.25092
##  Mean   :0.5251                     Mean   :0.33190
##  3rd Qu.:0.8135                     3rd Qu.:0.51760
##  Max.   :1.0000                     Max.   :0.97432
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.3426
##  Median :0.6210
##  Mean   :0.5948
##  3rd Qu.:0.8577
##  Max.   :0.9999
##  NA's   :4
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.0000
##  1st Qu.:0.3987
##  Median :0.6347
##  Mean   :0.6188
##  3rd Qu.:0.8517
##  Max.   :0.9999
##
```

Verificando a média de cada coluna selecionada

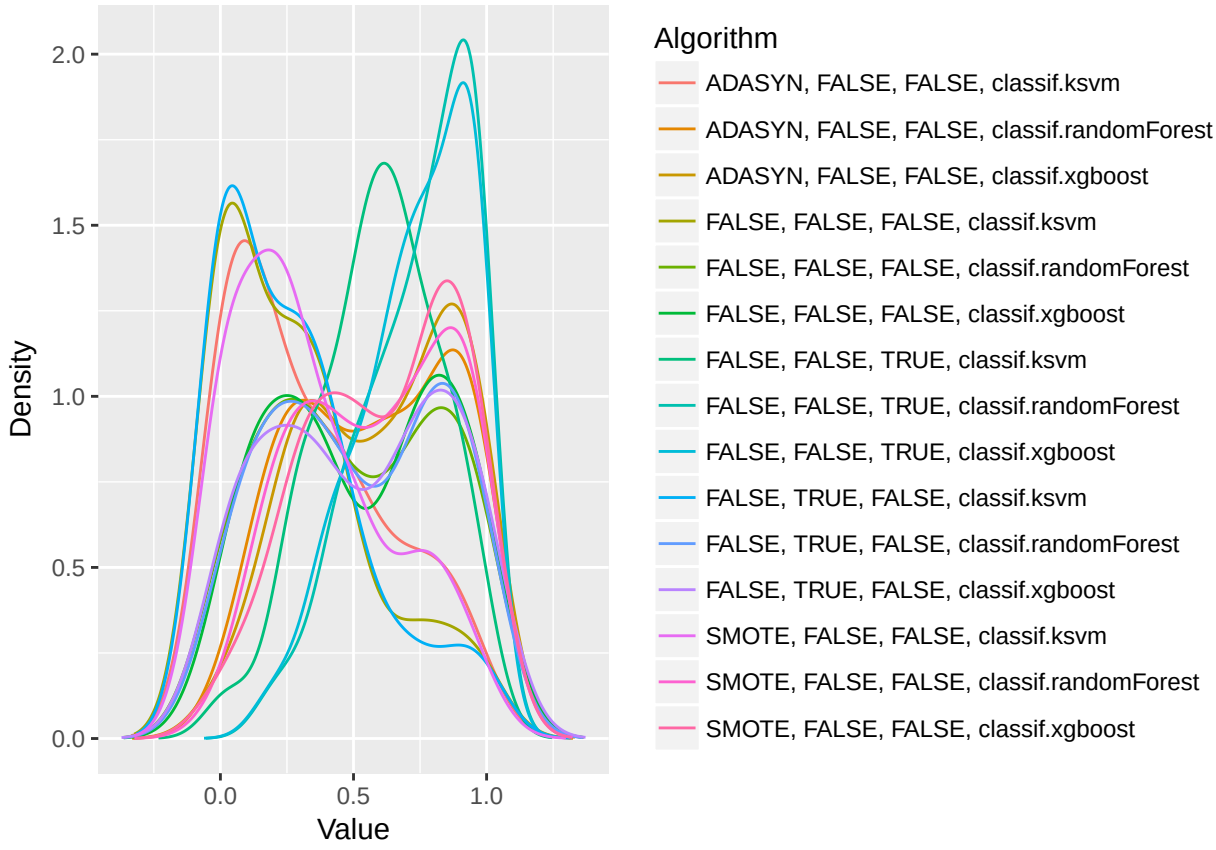
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.335617285973206"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.579683857721563"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.610139689852958"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.287789278569379"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.508099322739096"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.530100482332132"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.600727325466884"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.749830081812749"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.737450202566195"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.278469691750086"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.525800241493586"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.525146253916715"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.331898396128473"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.594775374330153"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.618804323312565"
```

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 = 486.98, 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,]                              FALSE
##  [6,]                               TRUE
##  [7,]                               TRUE
##  [8,]                               TRUE
##  [9,]                               TRUE
## [10,]                              FALSE
## [11,]                              FALSE
## [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,]                                  TRUE
##  [9,]                                  TRUE
## [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,]                               TRUE
##  [6,]                               TRUE
##  [7,]                               TRUE
##  [8,]                               TRUE
##  [9,]                               TRUE
## [10,]                              FALSE
## [11,]                               TRUE
## [12,]                               TRUE
## [13,]                              FALSE
## [14,]                               TRUE
## [15,]                               TRUE
##       FALSE, FALSE, FALSE, classif.randomForest
##  [1,]                                     FALSE
##  [2,]                                     FALSE
##  [3,]                                      TRUE
##  [4,]                                      TRUE
##  [5,]                                     FALSE
##  [6,]                                     FALSE
##  [7,]                                      TRUE
##  [8,]                                      TRUE
##  [9,]                                      TRUE
## [10,]                                      TRUE
## [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,]                            FALSE
##  [4,]                             TRUE
##  [5,]                             TRUE
```

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

Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                   11.268293
## ADASYN, FALSE, FALSE, classif.randomForest
##                                    7.310976
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                    5.725610
##          FALSE, FALSE, FALSE, classif.ksvm
##                                   11.902439
##  FALSE, FALSE, FALSE, classif.randomForest
##                                    9.182927
##       FALSE, FALSE, FALSE, classif.xgboost
##                                    8.420732
##           FALSE, FALSE, TRUE, classif.ksvm
##                                    6.195122
##   FALSE, FALSE, TRUE, classif.randomForest
##                                    3.286585
##        FALSE, FALSE, TRUE, classif.xgboost
##                                    3.091463
##           FALSE, TRUE, FALSE, classif.ksvm
##                                   12.048780
##   FALSE, TRUE, FALSE, classif.randomForest
##                                    8.969512
##        FALSE, TRUE, FALSE, classif.xgboost
##                                    8.670732
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   11.341463
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                    7.097561
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                    5.487805
```

Plotando grafico de Critical Diference

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