R Notebook

Parametros:

Measure = G-mean
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure
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                         0.36347460
## 2                         0.43290439
## 3                         0.57108256
## 4                         0.00000000
## 5                         1.00000000
## 6                         0.08824582
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.3308746
## 2                                  0.6141655
## 3                                  0.8640670
## 4                                  0.2357023
## 5                                  1.0000000
## 6                                  0.5243201
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.2664758                          0.4188068
## 2                            0.6451580                          0.4940109
## 3                            0.9179553                          0.5474844
## 4                            0.5690356                          0.0000000
## 5                            1.0000000                          1.0000000
## 6                            0.4744890                          0.3978845
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                 0.0000000
## 2                                 0.5800935
```

```
## 3                                0.9415891
## 4                                0.7022464
## 5                                1.0000000
## 6                                0.3617949
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                            0.0606977                         0.6449588
## 2                            0.5972897                         0.7810530
## 3                            0.7514148                         0.8696251
## 4                            0.5690356                         0.5826260
## 5                            1.0000000                         0.9973856
## 6                            0.4449251                         0.6812398
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                                0.6496529
## 2                                       NA
## 3                                0.9334878
## 4                                0.8621903
## 5                                0.9947712
## 6                                0.8102347
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                            0.6590354                         0.3765112
## 2                            0.8372833                         0.4757633
## 3                            0.9231073                         0.5487756
## 4                            0.9723959                         0.0000000
## 5                            0.9732218                         1.0000000
## 6                            0.8191576                         0.3094874
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                                0.0000000
## 2                                0.5821508
## 3                                0.9124818
## 4                                0.7989014
## 5                                1.0000000
## 6                                0.3617949
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                            0.0000000                         0.4067937
## 2                            0.5951135                         0.4771356
## 3                            0.6239091                         0.5190879
## 4                            0.5690356                         0.0000000
## 5                            1.0000000                         0.7979490
## 6                            0.4225366                         0.2027097
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                                0.3554756
## 2                                       NA
## 3                                0.9670808
## 4                                0.4689870
## 5                                1.0000000
## 6                                0.4566405
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                            0.3687932
## 2                            0.6294483
## 3                            0.9436014
## 4                            0.7954467
## 5                            1.0000000
## 6                            0.4877424
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.3333
##  Mean   :0.3965
##  3rd Qu.:0.6691
##  Max.   :1.0000
##  NA's   :1
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.4758
##  Median :0.8012
##  Mean   :0.6881
##  3rd Qu.:0.9305
##  Max.   :1.0000
##  NA's   :3
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000                        Min.   :0.0000
##  1st Qu.:0.5246                        1st Qu.:0.0000
##  Median :0.8307                        Median :0.2487
##  Mean   :0.7286                        Mean   :0.3448
##  3rd Qu.:0.9626                        3rd Qu.:0.6026
##  Max.   :1.0000                        Max.   :1.0000
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.3311
##  Median :0.6748
##  Mean   :0.5991
##  3rd Qu.:0.9213
##  Max.   :1.0000
##  NA's   :1
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.0000                       Min.   :0.02351
##  1st Qu.:0.3853                       1st Qu.:0.59441
##  Median :0.7548                       Median :0.78098
##  Mean   :0.6339                       Mean   :0.70721
##  3rd Qu.:0.9295                       3rd Qu.:0.88916
##  Max.   :1.0000                       Max.   :0.99893
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.2962
##  1st Qu.:0.8063
##  Median :0.9053
##  Mean   :0.8524
##  3rd Qu.:0.9732
##  Max.   :1.0000
##  NA's   :3
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.2897                      Min.   :0.0000
##  1st Qu.:0.7985                      1st Qu.:0.0000
##  Median :0.9033                      Median :0.2487
```

```
##  Mean   :0.8533                     Mean   :0.3369
##  3rd Qu.:0.9662                     3rd Qu.:0.5876
##  Max.   :1.0000                     Max.   :1.0000
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.3285
##  Median :0.7368
##  Mean   :0.6109
##  3rd Qu.:0.8995
##  Max.   :1.0000
##  NA's   :1
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000                     Min.   :0.00000
##  1st Qu.:0.2970                     1st Qu.:0.03727
##  Median :0.7530                     Median :0.42218
##  Mean   :0.6097                     Mean   :0.42015
##  3rd Qu.:0.9236                     3rd Qu.:0.74307
##  Max.   :1.0000                     Max.   :1.00000
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.4641
##  Median :0.7933
##  Mean   :0.6892
##  3rd Qu.:0.9627
##  Max.   :1.0000
##  NA's   :4
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.0000
##  1st Qu.:0.5619
##  Median :0.8524
##  Mean   :0.7367
##  3rd Qu.:0.9590
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

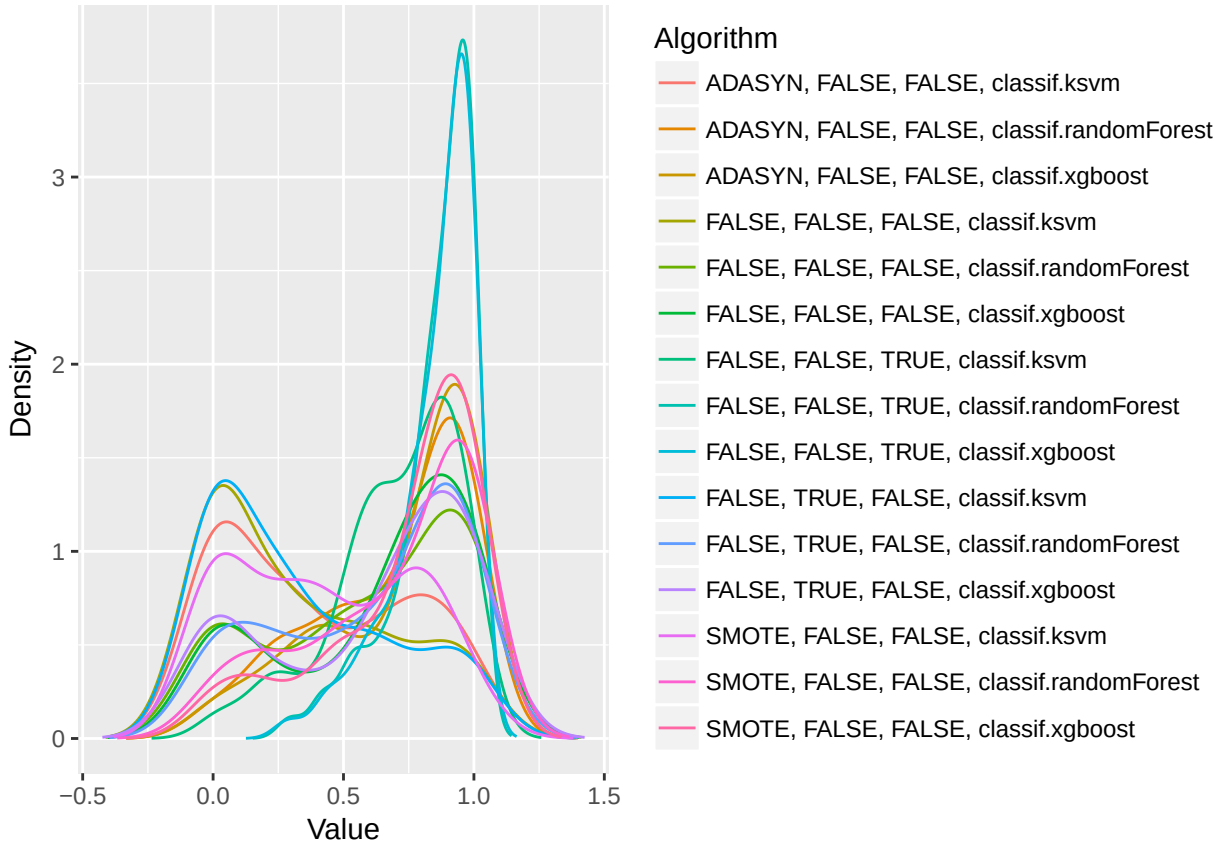
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.396525045613745"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.688077638325015"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.728579427465942"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.344803814251011"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.59905287753742"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.633916791482359"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.707214101221939"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.852379156362427"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.853321106649564"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.336874106445352"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.6108875245828"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.609656419949039"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.420146412846803"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.689162315735826"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.736672739062473"
```

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 = 450.33, 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,]                               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,]                               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,]                                     FALSE
##  [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,]                            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                               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,]                               FALSE                               TRUE
##       FALSE, TRUE, 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,]                                    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,]                                     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,]                                 TRUE
## [13,]                                 TRUE
## [14,]                                FALSE
## [15,]                                FALSE
```

Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                  11.274390
## ADASYN, FALSE, FALSE, classif.randomForest
##                                   7.079268
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                   5.548780
##          FALSE, FALSE, FALSE, classif.ksvm
##                                  12.109756
##  FALSE, FALSE, FALSE, classif.randomForest
##                                   8.987805
##       FALSE, FALSE, FALSE, classif.xgboost
##                                   8.067073
##           FALSE, FALSE, TRUE, classif.ksvm
##                                   6.621951
##   FALSE, FALSE, TRUE, classif.randomForest
##                                   3.670732
##        FALSE, FALSE, TRUE, classif.xgboost
##                                   3.628049
##           FALSE, TRUE, FALSE, classif.ksvm
##                                  12.134146
##   FALSE, TRUE, FALSE, classif.randomForest
##                                   8.823171
##        FALSE, TRUE, FALSE, classif.xgboost
##                                   8.548780
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                  11.048780
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                   6.932927
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                   5.524390
```

Plotando grafico de Critical Diference

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