R Notebook

Parametros:

Measure = Area under the curve
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure
Filter keys = NULL
Filter values = NULL

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

Tratamento dos dados

Carregando data set compilado

```
ds = read.csv("/home/rodrigo/Dropbox/UNICAMP/IC/estudo_cost_learning/SummaryResults/summary_compilation_

ds = filter(ds, learner != "classif.rusboost")
summary(ds)
```

```
##                  learner      weight_space
##  classif.ksvm        :17100   Mode :logical
##  classif.randomForest:17100   FALSE:41040
##  classif.rusboost    :    0   TRUE :10260
##  classif.xgboost     :17100   NA's :0
##
##
##
##                             measure        sampling     underbagging
##  Accuracy                       :10260   ADASYN:10260   Mode :logical
##  Area under the curve           :10260   FALSE :30780   FALSE:41040
##  F1 measure                     :10260   SMOTE :10260   TRUE :10260
##  G-mean                         :10260                  NA's :0
##  Matthews correlation coefficient:10260
##
##
##  tuning_measure    holdout_measure   holdout_measure_residual
##  Min.   :-0.1277   Min.   :-0.2120   Min.   :-0.4658
##  1st Qu.: 0.6911   1st Qu.: 0.4001   1st Qu.: 0.1994
##  Median : 0.9700   Median : 0.8571   Median : 0.5581
##  Mean   : 0.7903   Mean   : 0.6718   Mean   : 0.5298
##  3rd Qu.: 0.9975   3rd Qu.: 0.9900   3rd Qu.: 0.8755
##  Max.   : 1.0000   Max.   : 1.0000   Max.   : 1.0000
##  NA's   :1077      NA's   :1077      NA's   :1077
##  iteration_count                      dataset        imba.rate
##  Min.   :1       abalone                 :  900   Min.   :0.0010
##  1st Qu.:1       adult                   :  900   1st Qu.:0.0100
##  Median :2       bank                    :  900   Median :0.0300
##  Mean   :2       car                     :  900   Mean   :0.0286
```

```
##  3rd Qu.:3       cardiotocography-10clases:  900   3rd Qu.:0.0500
##  Max.   :3       cardiotocography-3clases :  900   Max.   :0.0500
##  NA's   :1077    (Other)                  :45900
```

Filtrando pela metrica

```
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        :3420   Mode :logical
##  classif.randomForest:3420   FALSE:8208
##  classif.rusboost    :   0   TRUE :2052
##  classif.xgboost     :3420   NA's :0
##
##
##
##                             measure        sampling    underbagging
##  Accuracy                       :    0   ADASYN:2052   Mode :logical
##  Area under the curve           :10260   FALSE :6156   FALSE:8208
##  F1 measure                     :    0   SMOTE :2052   TRUE :2052
##  G-mean                         :    0                 NA's :0
##  Matthews correlation coefficient:    0
##
##
##  tuning_measure   holdout_measure  holdout_measure_residual
##  Min.   :0.3023   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.9325   1st Qu.:0.8620   1st Qu.:0.7067
##  Median :0.9967   Median :0.9831   Median :0.8932
##  Mean   :0.9380   Mean   :0.8972   Mean   :0.8310
##  3rd Qu.:1.0000   3rd Qu.:0.9999   3rd Qu.:0.9819
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  NA's   :243      NA's   :243      NA's   :243
##  iteration_count                      dataset       imba.rate
##  Min.   :1       abalone                  : 180   Min.   :0.0010
##  1st Qu.:1       adult                    : 180   1st Qu.:0.0100
##  Median :2       bank                     : 180   Median :0.0300
##  Mean   :2       car                      : 180   Mean   :0.0286
##  3rd Qu.:3       cardiotocography-10clases: 180   3rd Qu.:0.0500
##  Max.   :3       cardiotocography-3clases : 180   Max.   :0.0500
##  NA's   :243     (Other)                  :9180
```

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] 228  15
```

```
# Renomeando a variavel
df = df_tec_wide_residual

head(df)
```

```
##   ADASYN, FALSE, FALSE, classif.ksvm
## 1                          0.5908544
## 2                          0.5908544
## 3                          0.6127365
## 4                          0.6462768
## 5                                 NA
## 6                                 NA
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.6115523
## 2                                  0.6115523
## 3                                  0.6578104
## 4                                  0.7047856
## 5                                         NA
## 6                                  0.8760552
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                             0.5329723                          0.5048135
## 2                             0.5329723                          0.5048135
## 3                             0.6776939                          0.6855853
## 4                             0.6442008                          0.6674464
## 5                             0.8532475                          0.4916767
## 6                             0.8532475                          0.4916767
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                 0.7033093
## 2                                 0.7033093
```

```
## 3                                0.6779007
## 4                                0.6383431
## 5                                0.8886010
## 6                                0.8886010
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                            0.6906137                         0.5738869
## 2                            0.6906137                         0.5738869
## 3                            0.7682208                         0.6796409
## 4                            0.6924366                         0.6777485
## 5                            0.9002132                         0.8178913
## 6                            0.9002132                         0.8178913
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                               0.6041516
## 2                               0.6041516
## 3                               0.7256108
## 4                               0.6600585
## 5                               0.9036622
## 6                               0.9036622
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.6801444                        0.5037304
## 2                           0.6801444                        0.5037304
## 3                           0.7254730                        0.6863779
## 4                           0.7141033                               NA
## 5                           0.9019359                        0.5547681
## 6                           0.9019359                        0.5547681
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                               0.6951264
## 2                               0.6951264
## 3                               0.6210758
## 4                               0.6383431
## 5                                      NA
## 6                                      NA
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                           0.6904934                        0.5335740
## 2                           0.6904934                        0.5335740
## 3                           0.7695303                        0.5259313
## 4                           0.6997758                        0.6413060
## 5                           0.8966787                        0.6074513
## 6                           0.8966787                        0.6074513
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                               0.6535499
## 2                               0.6535499
## 3                               0.7177539
## 4                               0.6820760
## 5                               0.8881366
## 6                                      NA
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                           0.5206980
## 2                           0.5206980
## 3                           0.6558117
## 4                           0.6484016
## 5                           0.8799975
## 6                           0.8799975
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.3593
##  1st Qu.:0.7152
##  Median :0.8995
##  Mean   :0.8476
##  3rd Qu.:0.9922
##  Max.   :1.0000
##  NA's   :14
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.3435
##  1st Qu.:0.8856
##  Median :0.9818
##  Mean   :0.9241
##  3rd Qu.:0.9993
##  Max.   :1.0000
##  NA's   :20
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.4176                        Min.   :0.3333
##  1st Qu.:0.8854                        1st Qu.:0.7141
##  Median :0.9783                        Median :0.9436
##  Mean   :0.9158                        Mean   :0.8470
##  3rd Qu.:0.9990                        3rd Qu.:0.9983
##  Max.   :1.0000                        Max.   :1.0000
##                                        NA's   :5
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.2924
##  1st Qu.:0.9067
##  Median :0.9872
##  Mean   :0.9220
##  3rd Qu.:0.9998
##  Max.   :1.0000
##  NA's   :4
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.4439                       Min.   :0.4413
##  1st Qu.:0.9049                       1st Qu.:0.7752
##  Median :0.9834                       Median :0.8783
##  Mean   :0.9298                       Mean   :0.8478
##  3rd Qu.:0.9995                       3rd Qu.:0.9648
##  Max.   :1.0000                       Max.   :1.0000
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.5018
##  1st Qu.:0.8917
##  Median :0.9808
##  Mean   :0.9194
##  3rd Qu.:0.9977
##  Max.   :1.0000
##  NA's   :6
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.4469                      Min.   :0.3333
##  1st Qu.:0.8885                      1st Qu.:0.7141
##  Median :0.9743                      Median :0.9427
```

```
##  Mean   :0.9170                     Mean   :0.8447
##  3rd Qu.:0.9968                     3rd Qu.:0.9983
##  Max.   :1.0000                     Max.   :1.0000
##                                     NA's   :5
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.3369
##  1st Qu.:0.9096
##  Median :0.9870
##  Mean   :0.9242
##  3rd Qu.:0.9997
##  Max.   :1.0000
##  NA's   :9
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.3793                     Min.   :0.2679
##  1st Qu.:0.9013                     1st Qu.:0.7202
##  Median :0.9831                     Median :0.9052
##  Mean   :0.9274                     Mean   :0.8402
##  3rd Qu.:0.9996                     3rd Qu.:0.9920
##  Max.   :1.0000                     Max.   :1.0000
##                                     NA's   :5
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.4685
##  1st Qu.:0.9052
##  Median :0.9896
##  Mean   :0.9298
##  3rd Qu.:0.9997
##  Max.   :1.0000
##  NA's   :13
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.3896
##  1st Qu.:0.8983
##  Median :0.9860
##  Mean   :0.9213
##  3rd Qu.:0.9994
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

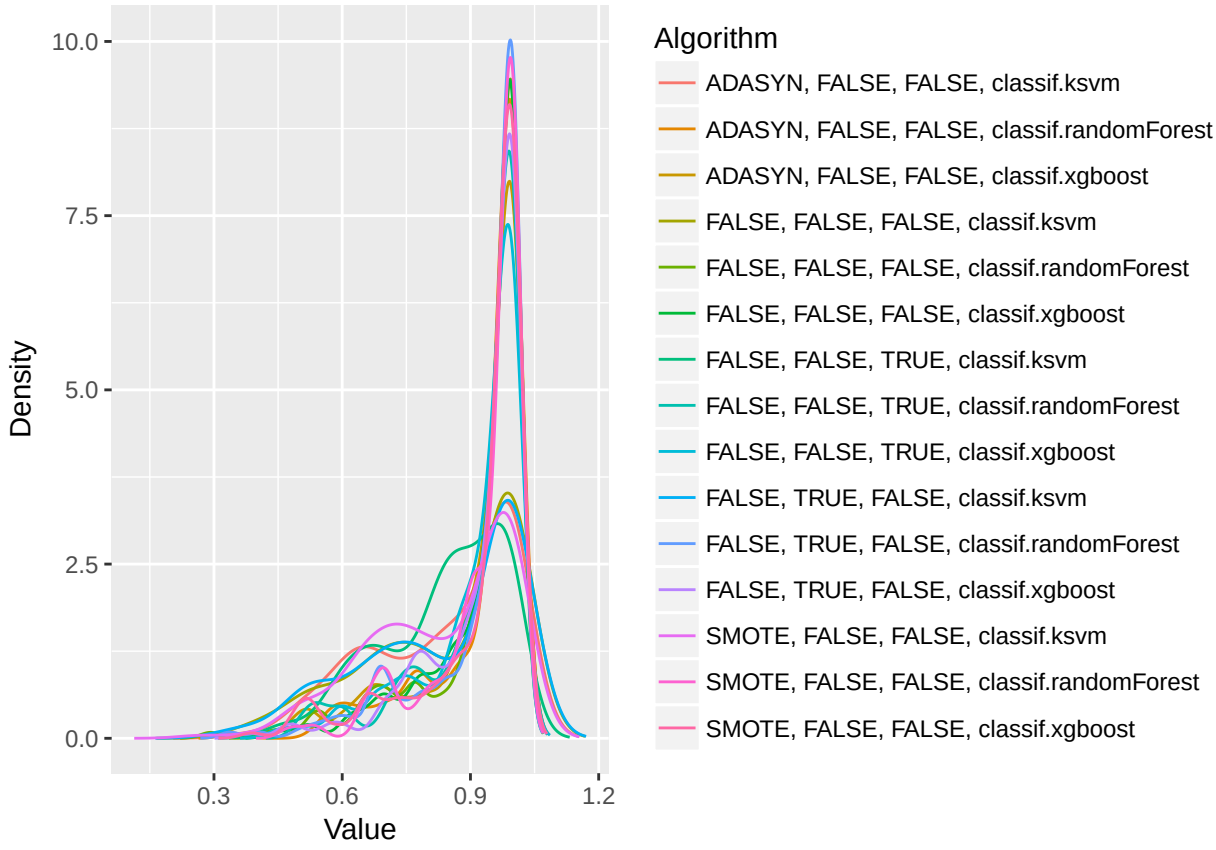
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.847600975288232"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.924075196327091"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.915784354960603"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.846954376991967"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.922012773476355"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.929783475298908"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.847805556989247"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.919399133478833"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.917034287580282"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.84467330893257"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.924221926965532"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.927399939339536"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.840218688802973"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.92982450763708"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.921273412559642"
```

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 = 588.22, 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,]                              FALSE
##  [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,]                                       TRUE
##  [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,]                                 FALSE
##  [6,]                                 FALSE
##  [7,]                                  TRUE
##  [8,]                                 FALSE
##  [9,]                                 FALSE
## [10,]                                  TRUE
## [11,]                                 FALSE
## [12,]                                 FALSE
## [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,]                              FALSE
## [10,]                              FALSE
## [11,]                               TRUE
## [12,]                               TRUE
## [13,]                              FALSE
## [14,]                               TRUE
## [15,]                               TRUE
##       FALSE, FALSE, FALSE, classif.randomForest
##  [1,]                                      TRUE
##  [2,]                                     FALSE
##  [3,]                                     FALSE
##  [4,]                                      TRUE
##  [5,]                                     FALSE
##  [6,]                                     FALSE
##  [7,]                                      TRUE
##  [8,]                                      TRUE
##  [9,]                                      TRUE
## [10,]                                      TRUE
## [11,]                                     FALSE
## [12,]                                     FALSE
## [13,]                                      TRUE
## [14,]                                     FALSE
## [15,]                                     FALSE
##       FALSE, FALSE, FALSE, classif.xgboost
##  [1,]                                 TRUE
##  [2,]                                FALSE
##  [3,]                                FALSE
##  [4,]                                 TRUE
##  [5,]                                FALSE
##  [6,]                                FALSE
##  [7,]                                 TRUE
##  [8,]                                 TRUE
##  [9,]                                 TRUE
## [10,]                                 TRUE
## [11,]                                FALSE
## [12,]                                FALSE
## [13,]                                 TRUE
## [14,]                                FALSE
## [15,]                                FALSE
##       FALSE, FALSE, TRUE, classif.ksvm
##  [1,]                             FALSE
##  [2,]                              TRUE
##  [3,]                              TRUE
##  [4,]                              TRUE
##  [5,]                              TRUE
```

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

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

Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                   10.473684
## ADASYN, FALSE, FALSE, classif.randomForest
##                                    6.508772
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                    7.322368
##          FALSE, FALSE, FALSE, classif.ksvm
##                                    9.493421
##  FALSE, FALSE, FALSE, classif.randomForest
##                                    6.094298
##       FALSE, FALSE, FALSE, classif.xgboost
##                                    6.195175
##           FALSE, FALSE, TRUE, classif.ksvm
##                                   11.458333
##   FALSE, FALSE, TRUE, classif.randomForest
##                                    7.997807
##        FALSE, FALSE, TRUE, classif.xgboost
##                                    8.497807
##           FALSE, TRUE, FALSE, classif.ksvm
##                                    9.655702
##   FALSE, TRUE, FALSE, classif.randomForest
##                                    6.366228
##        FALSE, TRUE, FALSE, classif.xgboost
##                                    6.258772
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   10.835526
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                    6.407895
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                    6.434211
```

Plotando grafico de Critical Diference

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