R Notebook

Parametros:

Measure = Area under the curve
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure_residual
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                          0.5136216
## 2                          0.5136216
## 3                          0.5814637
## 4                          0.5948389
## 5                                 NA
## 6                                 NA
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.6617140
## 2                                  0.6617140
## 3                                  0.6998561
## 4                                  0.6927146
## 5                                         NA
## 6                                  0.8661754
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                             0.6206414                          0.5133720
## 2                             0.6206414                          0.5133720
## 3                             0.6736275                          0.6445602
## 4                             0.6819788                          0.6479328
## 5                             0.8708447                          0.4549243
## 6                             0.8708447                          0.4549243
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                 0.6224639
## 2                                 0.6224639
```

```
## 3                                0.6736512
## 4                                0.6678776
## 5                                0.8601656
## 6                                0.8601656
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                           0.6375083                         0.6259505
## 2                           0.6375083                         0.6259505
## 3                           0.6883674                         0.6536723
## 4                           0.6806404                         0.6628553
## 5                           0.9023521                         0.7943273
## 6                           0.9023521                         0.7943273
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                               0.6705058
## 2                               0.6705058
## 3                               0.6852553
## 4                               0.6966937
## 5                               0.8795716
## 6                               0.8795716
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.6359470                         0.5137510
## 2                           0.6359470                         0.5137510
## 3                           0.6794753                         0.6538361
## 4                           0.6898095                                NA
## 5                           0.8885925                         0.5371265
## 6                           0.8885925                         0.5371265
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                               0.6178971
## 2                               0.6178971
## 3                               0.6735765
## 4                               0.6678776
## 5                                      NA
## 6                                      NA
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                           0.6375879                         0.5167415
## 2                           0.6375879                         0.5167415
## 3                           0.6886574                         0.5561325
## 4                           0.6830128                         0.5862406
## 5                           0.9015632                         0.6257369
## 6                           0.9015632                         0.6257369
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                               0.6600339
## 2                               0.6600339
## 3                               0.6909349
## 4                               0.6991028
## 5                               0.8664327
## 6                                      NA
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                           0.6217660
## 2                           0.6217660
## 3                           0.6705265
## 4                           0.6722286
## 5                           0.8685851
## 6                           0.8685851
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.3506
##  1st Qu.:0.6960
##  Median :0.8394
##  Mean   :0.8046
##  3rd Qu.:0.9605
##  Max.   :1.0000
##  NA's   :14
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.3050
##  1st Qu.:0.7699
##  Median :0.9323
##  Mean   :0.8631
##  3rd Qu.:0.9845
##  Max.   :1.0000
##  NA's   :20
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.3300                        Min.   :0.3959
##  1st Qu.:0.7157                        1st Qu.:0.6696
##  Median :0.9064                        Median :0.8382
##  Mean   :0.8409                        Mean   :0.8027
##  3rd Qu.:0.9835                        3rd Qu.:0.9680
##  Max.   :1.0000                        Max.   :1.0000
##                                        NA's   :5
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.3885
##  1st Qu.:0.7849
##  Median :0.9273
##  Mean   :0.8605
##  3rd Qu.:0.9844
##  Max.   :1.0000
##  NA's   :4
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.3577                       Min.   :0.3864
##  1st Qu.:0.7253                       1st Qu.:0.6270
##  Median :0.9123                       Median :0.7750
##  Mean   :0.8519                       Mean   :0.7632
##  3rd Qu.:0.9764                       3rd Qu.:0.9041
##  Max.   :1.0000                       Max.   :0.9998
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.2777
##  1st Qu.:0.7119
##  Median :0.8897
##  Mean   :0.8391
##  3rd Qu.:0.9800
##  Max.   :1.0000
##  NA's   :6
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.3213                      Min.   :0.3959
##  1st Qu.:0.7030                      1st Qu.:0.6696
##  Median :0.8955                      Median :0.8382
```

```
##  Mean   :0.8329                     Mean   :0.8011
##  3rd Qu.:0.9740                     3rd Qu.:0.9680
##  Max.   :1.0000                     Max.   :1.0000
##                                     NA's   :5
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.4130
##  1st Qu.:0.7703
##  Median :0.9311
##  Mean   :0.8601
##  3rd Qu.:0.9848
##  Max.   :1.0000
##  NA's   :9
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.2916                     Min.   :0.2778
##  1st Qu.:0.7494                     1st Qu.:0.6646
##  Median :0.9119                     Median :0.8266
##  Mean   :0.8498                     Mean   :0.7937
##  3rd Qu.:0.9771                     3rd Qu.:0.9466
##  Max.   :1.0000                     Max.   :1.0000
##                                     NA's   :5
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.3383
##  1st Qu.:0.7605
##  Median :0.9244
##  Mean   :0.8570
##  3rd Qu.:0.9876
##  Max.   :1.0000
##  NA's   :13
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.2896
##  1st Qu.:0.7557
##  Median :0.9041
##  Mean   :0.8474
##  3rd Qu.:0.9824
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

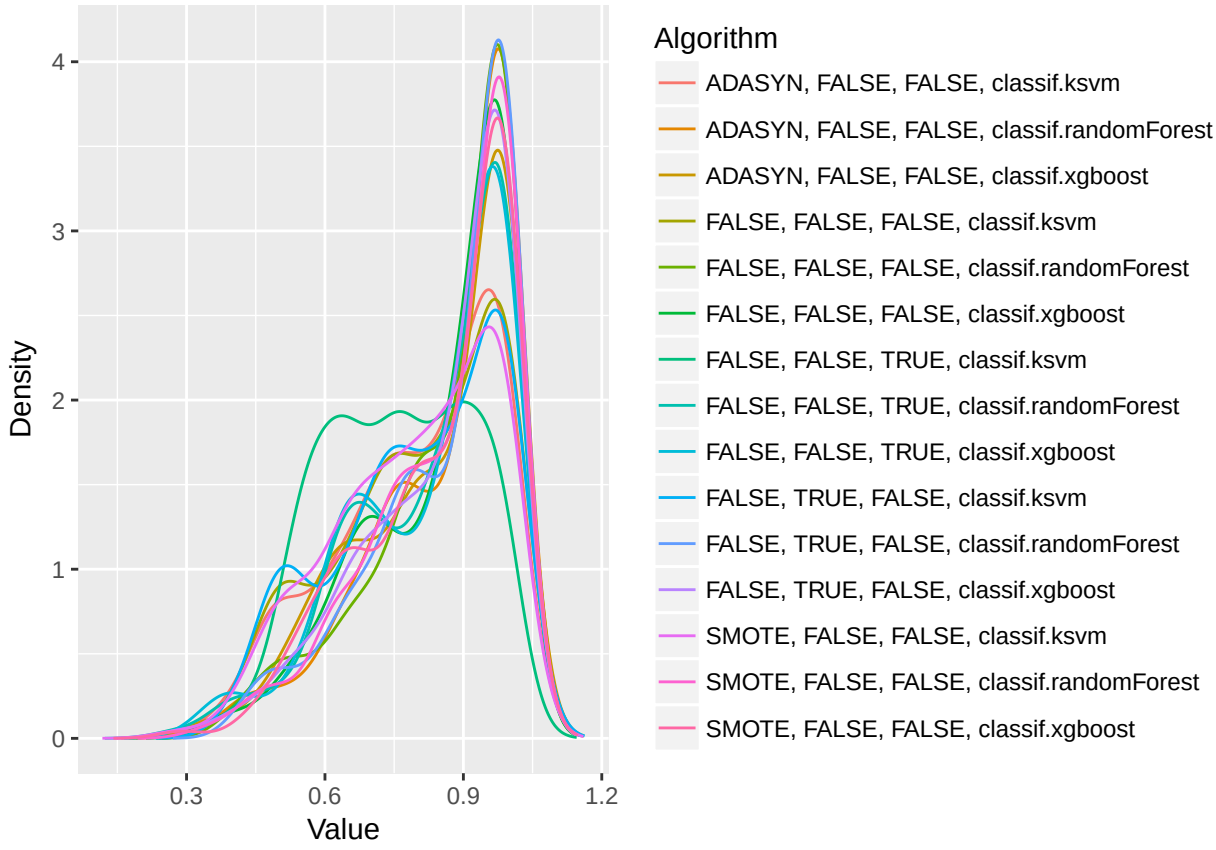
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.804644865042546"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.863138358332772"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.840867936311577"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.802661345179165"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.860489918436232"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.851945654185092"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.763220591868472"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.839132952019863"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.832904251305965"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.80113573845658"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.860119350493739"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.84982646091696"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.793713285207171"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.856974569462306"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.847373661095876"
```

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 = 482.47, 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,]                              FALSE
## [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,]                                  TRUE
##  [6,]                                 FALSE
##  [7,]                                  TRUE
##  [8,]                                 FALSE
##  [9,]                                 FALSE
## [10,]                                  TRUE
## [11,]                                  TRUE
## [12,]                                 FALSE
## [13,]                                  TRUE
## [14,]                                  TRUE
## [15,]                                 FALSE
```

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

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

```
##  [2,]                                FALSE
##  [3,]                                FALSE
##  [4,]                                 TRUE
##  [5,]                                FALSE
##  [6,]                                FALSE
##  [7,]                                 TRUE
##  [8,]                                FALSE
##  [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
##                                   9.793860
## ADASYN, FALSE, FALSE, classif.randomForest
##                                   6.232456
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                   7.537281
##          FALSE, FALSE, FALSE, classif.ksvm
##                                   9.311404
##  FALSE, FALSE, FALSE, classif.randomForest
##                                   6.026316
##       FALSE, FALSE, FALSE, classif.xgboost
##                                   6.938596
##           FALSE, FALSE, TRUE, classif.ksvm
##                                  11.745614
##   FALSE, FALSE, TRUE, classif.randomForest
##                                   8.263158
##        FALSE, FALSE, TRUE, classif.xgboost
##                                   8.631579
##           FALSE, TRUE, FALSE, classif.ksvm
##                                   9.421053
##   FALSE, TRUE, FALSE, classif.randomForest
##                                   6.094298
##        FALSE, TRUE, FALSE, classif.xgboost
##                                   7.030702
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   9.842105
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                   6.103070
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                   7.028509
```

Plotando grafico de Critical Diference

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