R Notebook

Parametros:

Measure = Area under the curve
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure_residual
Filter keys = imba.rate
Filter values = 0.03

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

Tratamento dos dados

Carregando data set compilado

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

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

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

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

Filtrando pela metrica

```
ds = filter(ds, measure == params$measure)
```

Filtrando o data set

```
if(params$filter_keys != 'NULL' && !is.null(params$filter_keys)){
  dots = paste0(params$filter_keys," == '",params$filter_values,"'")
  ds = filter_(ds, .dots = dots)
}

summary(ds)
```

```
##                  learner    weight_space
##  classif.ksvm        :990   Mode :logical
##  classif.randomForest:990   FALSE:2376
##  classif.rusboost    :  0   TRUE :594
##  classif.xgboost     :990   NA's :0
##
##
##
##                             measure       sampling    underbagging
##  Accuracy                       :   0   ADASYN: 594   Mode :logical
##  Area under the curve           :2970   FALSE :1782   FALSE:2376
##  F1 measure                     :   0   SMOTE : 594   TRUE :594
##  G-mean                         :   0                 NA's :0
##  Matthews correlation coefficient:   0
##
##
##  tuning_measure   holdout_measure  holdout_measure_residual
##  Min.   :0.3023   Min.   :0.0000   Min.   :0.00057
##  1st Qu.:0.9338   1st Qu.:0.8603   1st Qu.:0.69645
##  Median :0.9963   Median :0.9835   Median :0.89271
##  Mean   :0.9356   Mean   :0.8947   Mean   :0.82476
##  3rd Qu.:0.9999   3rd Qu.:0.9998   3rd Qu.:0.98444
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##  NA's   :66       NA's   :66       NA's   :66
##  iteration_count          dataset       imba.rate
##  Min.   :1       abalone      :  45   Min.   :0.03
##  1st Qu.:1       adult        :  45   1st Qu.:0.03
##  Median :2       annealing    :  45   Median :0.03
##  Mean   :2       arrhythmia   :  45   Mean   :0.03
##  3rd Qu.:3       balance-scale:  45   3rd Qu.:0.03
##  Max.   :3       bank         :  45   Max.   :0.03
##  NA's   :66      (Other)      :2700
```

Computando as médias das iteracoes

```
ds = group_by(ds, learner , weight_space , measure , sampling , underbagging , dataset , imba.rate)
ds = summarise(ds, tuning_measure = mean(tuning_measure), holdout_measure = mean(holdout_measure),
               holdout_measure_residual = mean(holdout_measure_residual))

ds = as.data.frame(ds)
```

Criando dataframe

```
# Dividindo o ds em n, um para cada técnica
splited_df = ds %>% group_by_at(.vars = params$columns) %>% do(vals = as.data.frame(.)) %>% select(vals)

# Juntando cada uma das partes horizontalmente em um data set
df_tec_wide = do.call("cbind", splited_df)

# Renomeando duplicacao de nomes
colnames(df_tec_wide) = make.unique(colnames(df_tec_wide))

# Selecionando apenas as medidas da performance escolhida
df_tec_wide_residual = select(df_tec_wide, matches(paste("^", params$performance, "$|", params$performa

# Renomeando colunas
new_names = NULL
for(i in (1:length(splited_df))){
  id = toString(sapply(splited_df[[i]][1, params$columns], as.character))
  new_names = c(new_names, id)
}
colnames(df_tec_wide_residual) = new_names

# Verificando a dimensao do df
dim(df_tec_wide_residual)
```

```
## [1] 66 15
```

```
# Renomeando a variavel
df = df_tec_wide_residual

head(df)
```

```
##   ADASYN, FALSE, FALSE, classif.ksvm
## 1                          0.5814637
## 2                                 NA
## 3                          0.8024917
## 4                          0.7091503
## 5                          0.8477891
## 6                          0.7722898
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.6998561
## 2                                  0.8728338
## 3                                  0.9828610
## 4                                  0.9428105
## 5                                  0.5724644
## 6                                  0.8797456
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                             0.6736275                          0.6445602
## 2                             0.8932010                                 NA
## 3                             0.9502517                          0.7863708
## 4                             0.9624183                          0.5000000
## 5                             0.5299938                          0.9006957
## 6                             0.8486947                          0.7787141
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                 0.6736512
## 2                                 0.8822224
```

```
## 3                                0.9851519
## 4                                0.9852941
## 5                                0.6311843
## 6                                0.8747629
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                           0.6883674                         0.6536723
## 2                           0.9123596                         0.8440126
## 3                           0.9680412                         0.8142570
## 4                           0.9934641                         0.4754902
## 5                           0.6291667                         0.6339981
## 6                           0.8655266                         0.7372121
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                               0.6852553
## 2                               0.8799091
## 3                               0.9662170
## 4                               0.9803922
## 5                               0.6170223
## 6                               0.8673811
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.6794753                         0.6538361
## 2                           0.8936782                                NA
## 3                           0.8926127                         0.7863708
## 4                           0.9869281                         0.5000000
## 5                           0.6310915                         0.9006957
## 6                           0.8512661                         0.7787141
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                               0.6735765
## 2                               0.8786111
## 3                               0.9832570
## 4                               0.9566993
## 5                               0.6291899
## 6                               0.8753356
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                           0.6886574                         0.5561325
## 2                           0.9122748                                NA
## 3                           0.9700492                         0.7961423
## 4                           0.9297386                         0.2777778
## 5                           0.6291667                         0.8827922
## 6                           0.8686244                         0.7637149
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                               0.6909349
## 2                                      NA
## 3                               0.9842186
## 4                               0.9493464
## 5                               0.5619048
## 6                               0.8725859
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                           0.6705265
## 2                           0.8914266
## 3                           0.9560071
## 4                           0.9869281
## 5                           0.6199985
## 6                           0.8454887
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.3781
##  1st Qu.:0.6948
##  Median :0.8653
##  Mean   :0.8100
##  3rd Qu.:0.9695
##  Max.   :0.9996
##  NA's   :3
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.3050
##  1st Qu.:0.7503
##  Median :0.9296
##  Mean   :0.8421
##  3rd Qu.:0.9859
##  Max.   :1.0000
##  NA's   :5
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.3300                        Min.   :0.3959
##  1st Qu.:0.7225                        1st Qu.:0.6743
##  Median :0.9250                        Median :0.8413
##  Mean   :0.8343                        Mean   :0.8016
##  3rd Qu.:0.9856                        3rd Qu.:0.9681
##  Max.   :1.0000                        Max.   :1.0000
##                                        NA's   :1
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.4142
##  1st Qu.:0.7629
##  Median :0.9277
##  Mean   :0.8608
##  3rd Qu.:0.9859
##  Max.   :1.0000
##  NA's   :1
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.3577                       Min.   :0.3864
##  1st Qu.:0.7227                       1st Qu.:0.6189
##  Median :0.9154                       Median :0.7652
##  Mean   :0.8376                       Mean   :0.7626
##  3rd Qu.:0.9738                       3rd Qu.:0.9079
##  Max.   :0.9999                       Max.   :0.9998
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.2777
##  1st Qu.:0.6945
##  Median :0.8893
##  Mean   :0.8294
##  3rd Qu.:0.9814
##  Max.   :1.0000
##  NA's   :2
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.3213                      Min.   :0.3959
##  1st Qu.:0.6848                      1st Qu.:0.6674
##  Median :0.9062                      Median :0.8352
```

```
##  Mean   :0.8205                     Mean   :0.7958
##  3rd Qu.:0.9767                     3rd Qu.:0.9643
##  Max.   :1.0000                     Max.   :1.0000
##                                     NA's   :1
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.4384
##  1st Qu.:0.7348
##  Median :0.9029
##  Mean   :0.8482
##  3rd Qu.:0.9821
##  Max.   :1.0000
##  NA's   :4
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.2916                     Min.   :0.2778
##  1st Qu.:0.7020                     1st Qu.:0.6818
##  Median :0.9231                     Median :0.8352
##  Mean   :0.8375                     Mean   :0.7975
##  3rd Qu.:0.9763                     3rd Qu.:0.9458
##  Max.   :0.9999                     Max.   :1.0000
##                                     NA's   :2
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.3383
##  1st Qu.:0.7537
##  Median :0.9339
##  Mean   :0.8535
##  3rd Qu.:0.9904
##  Max.   :1.0000
##  NA's   :3
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.2896
##  1st Qu.:0.7511
##  Median :0.9021
##  Mean   :0.8424
##  3rd Qu.:0.9827
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

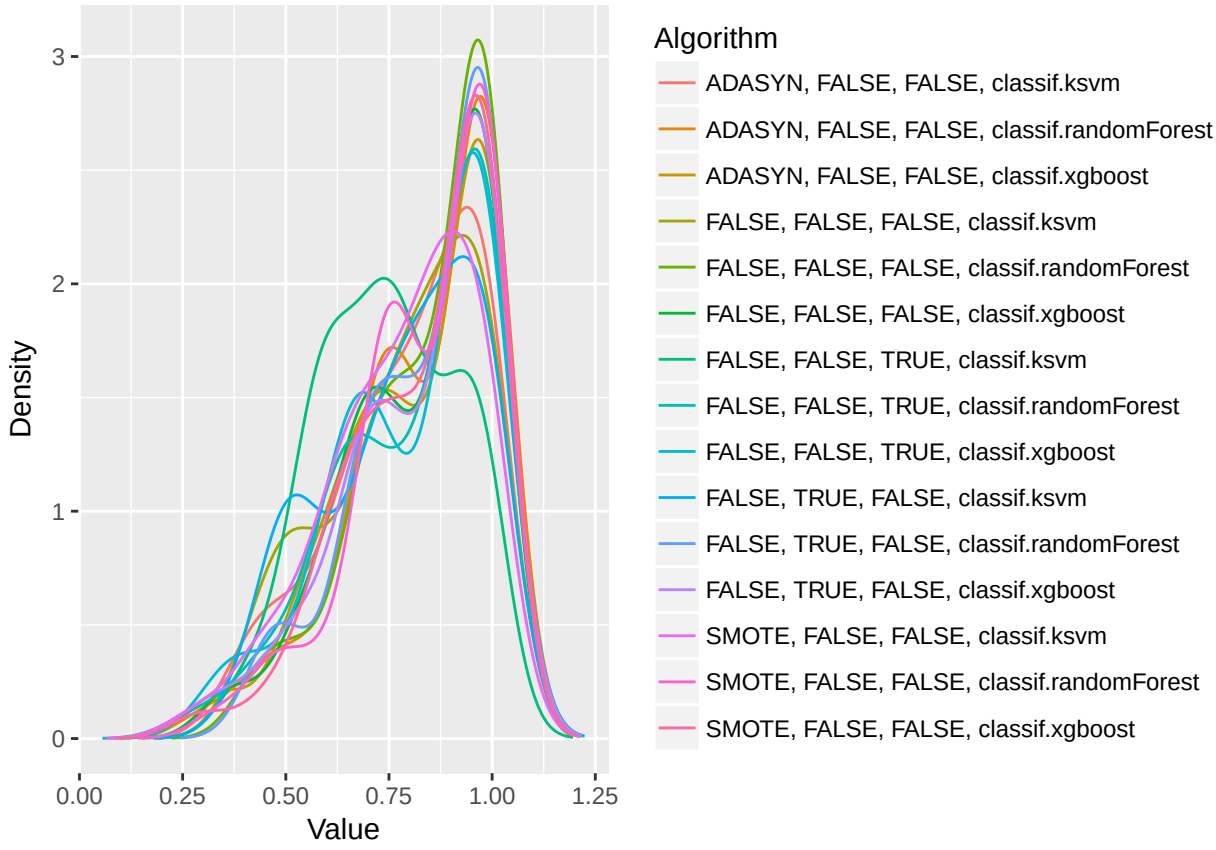
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.81002888811386"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.842087894930025"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.834262123965253"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.80164059172482"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.860811237724216"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.837639325782842"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.762625385991208"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.829400812031909"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.820473290301526"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.79582598135536"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.848222541686779"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.837452500301411"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.797488425998206"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.85354012036193"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.842360720807623"
```

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 = 134.15, 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,]                               FALSE
##  [4,]                               FALSE
##  [5,]                                TRUE
##  [6,]                               FALSE
##  [7,]                               FALSE
##  [8,]                               FALSE
##  [9,]                               FALSE
## [10,]                               FALSE
## [11,]                                TRUE
## [12,]                               FALSE
## [13,]                               FALSE
## [14,]                                TRUE
## [15,]                               FALSE
##       ADASYN, FALSE, FALSE, classif.randomForest
##  [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
##       ADASYN, FALSE, FALSE, classif.xgboost
##  [1,]                                 FALSE
##  [2,]                                 FALSE
##  [3,]                                 FALSE
##  [4,]                                 FALSE
##  [5,]                                 FALSE
##  [6,]                                 FALSE
##  [7,]                                  TRUE
##  [8,]                                 FALSE
##  [9,]                                 FALSE
## [10,]                                 FALSE
## [11,]                                 FALSE
## [12,]                                 FALSE
## [13,]                                  TRUE
## [14,]                                 FALSE
## [15,]                                 FALSE
```

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

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

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

Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                   9.393939
## ADASYN, FALSE, FALSE, classif.randomForest
##                                   6.106061
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                   7.272727
##          FALSE, FALSE, FALSE, classif.ksvm
##                                   9.030303
##  FALSE, FALSE, FALSE, classif.randomForest
##                                   5.772727
##       FALSE, FALSE, FALSE, classif.xgboost
##                                   7.621212
##           FALSE, FALSE, TRUE, classif.ksvm
##                                  11.621212
##   FALSE, FALSE, TRUE, classif.randomForest
##                                   8.522727
##        FALSE, FALSE, TRUE, classif.xgboost
##                                   8.757576
##           FALSE, TRUE, FALSE, classif.ksvm
##                                   9.045455
##   FALSE, TRUE, FALSE, classif.randomForest
##                                   6.712121
##        FALSE, TRUE, FALSE, classif.xgboost
##                                   7.356061
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                  10.196970
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                   5.719697
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                   6.871212
```

Plotando grafico de Critical Diference

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