R Notebook

Parametros:

Measure = Matthews correlation coefficient
Columns = sampling, weight_space, underbagging, learner
Performance = tuning_measure
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            :   0   FALSE :1782   FALSE:2376
##  F1 measure                      :   0   SMOTE : 594   TRUE :594
##  G-mean                          :   0                 NA's :0
##  Matthews correlation coefficient:2970
##
##
##  tuning_measure     holdout_measure   holdout_measure_residual
##  Min.   :-0.05673   Min.   :-0.1757   Min.   :-0.4658
##  1st Qu.: 0.33347   1st Qu.: 0.0000   1st Qu.: 0.0391
##  Median : 0.83196   Median : 0.5030   Median : 0.2116
##  Mean   : 0.66187   Mean   : 0.4753   Mean   : 0.3111
##  3rd Qu.: 0.98596   3rd Qu.: 0.8126   3rd Qu.: 0.5286
##  Max.   : 1.00000   Max.   : 1.0000   Max.   : 1.0000
##  NA's   :48         NA's   :48        NA's   :48
##  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   :48      (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                          0.9202676
## 2                          0.9501846
## 3                          0.9664565
## 4                          0.8624084
## 5                          1.0000000
## 6                          0.9705260
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.9103009
## 2                                  0.9577139
## 3                                  0.9847317
## 4                                  0.9941001
## 5                                  1.0000000
## 6                                  0.9725192
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.9344153                          0.02820236
## 2                            0.9685591                          0.11635362
## 3                            0.9773841                          0.17638584
## 4                            0.9736076                          0.00000000
## 5                            1.0000000                          0.95818163
## 6                            0.9691481                          0.05285858
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                0.02094578
## 2                                0.47846662
```

```
## 3                                0.51635002
## 4                                0.45302968
## 5                                1.00000000
## 6                                0.08104800
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                          0.005814705                        0.11727958
## 2                          0.497229873                        0.20937678
## 3                          0.417950912                        0.29672610
## 4                          0.471018509                        0.11395088
## 5                          1.000000000                        0.77180093
## 6                          0.161216151                        0.09049135
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                                0.1234075
## 2                                0.2709790
## 3                                0.3265577
## 4                                0.4757866
## 5                                0.5592691
## 6                                0.2518343
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                          0.1112294                        0.01746777
## 2                          0.2811698                        0.14047268
## 3                          0.2940804                        0.17638584
## 4                          0.4562134                        0.00000000
## 5                          0.5214770                        0.95818163
## 6                          0.2482089                        0.05285858
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                                0.0000000
## 2                                0.4852789
## 3                                0.4957134
## 4                                0.6224115
## 5                                1.0000000
## 6                                0.1212584
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                          0.05558296                         0.9235585
## 2                          0.48579438                         0.9528485
## 3                          0.36819488                         0.9648315
## 4                          0.52625410                         0.7663273
## 5                          1.00000000                         1.0000000
## 6                          0.16226147                         0.9688113
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                                 0.9052261
## 2                                 0.9592306
## 3                                 0.9860182
## 4                                 0.9900608
## 5                                 1.0000000
## 6                                 0.9635796
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                           0.9357333
## 2                           0.9703027
## 3                           0.9761847
## 4                           0.9782432
## 5                           1.0000000
## 6                           0.9698831
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.6183
##  1st Qu.:0.9577
##  Median :0.9886
##  Mean   :0.9638
##  3rd Qu.:0.9954
##  Max.   :1.0000
##  NA's   :2
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.6108
##  1st Qu.:0.9646
##  Median :0.9922
##  Mean   :0.9742
##  3rd Qu.:0.9982
##  Max.   :1.0000
##  NA's   :4
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.6105                        Min.   :-0.001814
##  1st Qu.:0.9572                        1st Qu.: 0.009720
##  Median :0.9859                        Median : 0.154596
##  Mean   :0.9692                        Mean   : 0.299413
##  3rd Qu.:0.9965                        3rd Qu.: 0.614705
##  Max.   :1.0000                        Max.   : 0.960041
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.1395
##  Median :0.5711
##  Mean   :0.5348
##  3rd Qu.:0.8705
##  Max.   :1.0000
##  NA's   :2
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :-0.004855                    Min.   :0.01683
##  1st Qu.: 0.384607                    1st Qu.:0.18052
##  Median : 0.640134                    Median :0.35660
##  Mean   : 0.592192                    Mean   :0.40576
##  3rd Qu.: 0.870799                    3rd Qu.:0.63116
##  Max.   : 1.000000                    Max.   :0.94840
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.006784
##  1st Qu.:0.254509
##  Median :0.445206
##  Mean   :0.461211
##  3rd Qu.:0.681317
##  Max.   :1.000000
##  NA's   :2
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.03527                     Min.   :-0.002485
##  1st Qu.:0.26582                     1st Qu.: 0.002466
##  Median :0.42119                     Median : 0.136639
```

```
##  Mean   :0.43045                     Mean   : 0.282068
##  3rd Qu.:0.61390                     3rd Qu.: 0.577214
##  Max.   :1.00000                     Max.   : 0.960041
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :-0.002005
##  1st Qu.: 0.144148
##  Median : 0.566851
##  Mean   : 0.538192
##  3rd Qu.: 0.876502
##  Max.   : 1.000000
##  NA's   :2
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :-0.001833                   Min.   :0.6181
##  1st Qu.: 0.389216                   1st Qu.:0.9545
##  Median : 0.640756                   Median :0.9849
##  Mean   : 0.598015                   Mean   :0.9638
##  3rd Qu.: 0.869774                   3rd Qu.:0.9954
##  Max.   : 1.000000                   Max.   :1.0000
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.6164
##  1st Qu.:0.9645
##  Median :0.9914
##  Mean   :0.9750
##  3rd Qu.:0.9989
##  Max.   :1.0000
##  NA's   :4
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.6164
##  1st Qu.:0.9645
##  Median :0.9896
##  Mean   :0.9733
##  3rd Qu.:0.9964
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

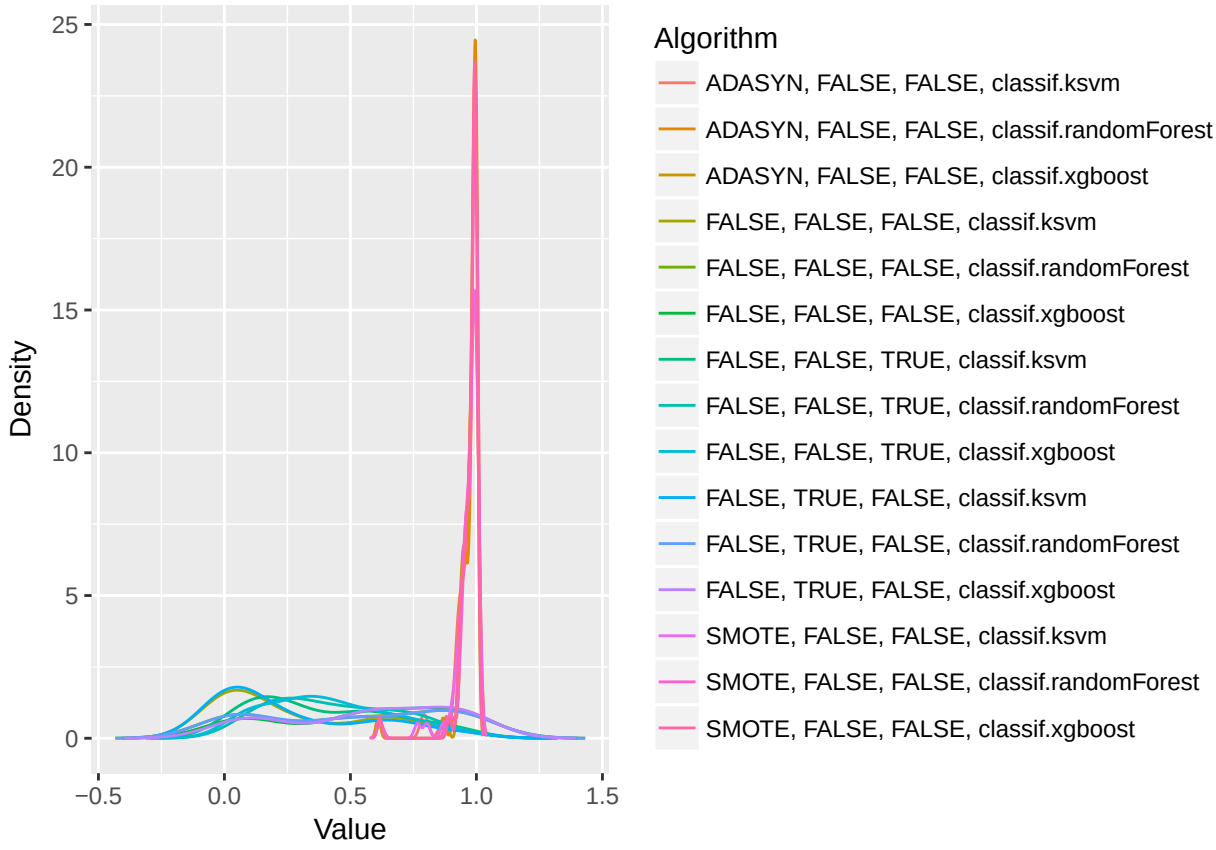
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.963784294528457"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.974154997424581"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.969169764732366"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.29941264429869"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.534816715535788"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.592192099426845"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.405760027677724"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.461210499420144"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.430450344978768"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.282068201423063"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.538192477158144"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.598015050122279"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.963786704952045"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.975016861275657"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.973338655492306"
```

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 = 591, 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,]                               FALSE
##  [3,]                               FALSE
##  [4,]                                TRUE
##  [5,]                                TRUE
##  [6,]                                TRUE
##  [7,]                                TRUE
##  [8,]                                TRUE
##  [9,]                                TRUE
## [10,]                                TRUE
## [11,]                                TRUE
## [12,]                                TRUE
## [13,]                               FALSE
## [14,]                               FALSE
## [15,]                               FALSE
##       ADASYN, FALSE, FALSE, classif.randomForest
##  [1,]                                      FALSE
##  [2,]                                      FALSE
##  [3,]                                      FALSE
##  [4,]                                       TRUE
##  [5,]                                       TRUE
##  [6,]                                       TRUE
##  [7,]                                       TRUE
##  [8,]                                       TRUE
##  [9,]                                       TRUE
## [10,]                                       TRUE
## [11,]                                       TRUE
## [12,]                                       TRUE
## [13,]                                      FALSE
## [14,]                                      FALSE
## [15,]                                      FALSE
##       ADASYN, FALSE, FALSE, classif.xgboost
##  [1,]                                 FALSE
##  [2,]                                 FALSE
##  [3,]                                 FALSE
##  [4,]                                  TRUE
##  [5,]                                  TRUE
##  [6,]                                  TRUE
##  [7,]                                  TRUE
##  [8,]                                  TRUE
##  [9,]                                  TRUE
## [10,]                                  TRUE
## [11,]                                  TRUE
## [12,]                                  TRUE
## [13,]                                 FALSE
## [14,]                                 FALSE
## [15,]                                 FALSE
```

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

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

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

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

Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                   4.204545
## ADASYN, FALSE, FALSE, classif.randomForest
##                                   3.939394
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                   4.015152
##          FALSE, FALSE, FALSE, classif.ksvm
##                                  12.446970
##  FALSE, FALSE, FALSE, classif.randomForest
##                                  10.000000
##       FALSE, FALSE, FALSE, classif.xgboost
##                                   8.772727
##           FALSE, FALSE, TRUE, classif.ksvm
##                                  11.500000
##   FALSE, FALSE, TRUE, classif.randomForest
##                                  10.833333
##        FALSE, FALSE, TRUE, classif.xgboost
##                                  11.492424
##           FALSE, TRUE, FALSE, classif.ksvm
##                                  12.659091
##   FALSE, TRUE, FALSE, classif.randomForest
##                                   9.954545
##        FALSE, TRUE, FALSE, classif.xgboost
##                                   8.393939
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   3.704545
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                   4.068182
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                   4.015152
```

Plotando grafico de Critical Diference

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