R Notebook

Parametros:

Measure = Area under the curve
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           :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                          0.9879838
## 2                                 NA
## 3                          0.9943687
## 4                          0.9962909
## 5                          1.0000000
## 6                          0.9995258
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.9922782
## 2                                  0.9979965
## 3                                  0.9988598
## 4                                  0.9999385
## 5                                  1.0000000
## 6                                  0.9993270
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                             0.9947308                          0.5950041
## 2                             0.9979218                                 NA
## 3                             0.9993358                          0.6989927
## 4                             0.9993193                          0.5000000
## 5                             1.0000000                          0.9995693
## 6                             0.9987106                          0.7797869
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                 0.6663678
## 2                                 0.8966458
```

```
## 3                                0.9757173
## 4                                0.9248947
## 5                                1.0000000
## 6                                0.8696595
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                           0.7238225                         0.6592267
## 2                           0.9147722                         0.8362549
## 3                           0.9459066                         0.8418271
## 4                           0.9233865                         0.6194164
## 5                           1.0000000                         0.9993614
## 6                           0.8698905                         0.7357850
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                             0.7166933
## 2                             0.8961590
## 3                             0.9447322
## 4                             0.9605933
## 5                             1.0000000
## 6                             0.8623656
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.7317528                         0.5958792
## 2                           0.9026472                                NA
## 3                           0.9114799                         0.6989927
## 4                           0.9683462                         0.5000000
## 5                           1.0000000                         0.9995693
## 6                           0.8614712                         0.7797869
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                             0.6620182
## 2                             0.8916592
## 3                             0.9675346
## 4                             0.9449771
## 5                             1.0000000
## 6                             0.8793182
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                           0.7237752                         0.9897728
## 2                           0.9144373                                NA
## 3                           0.9594818                         0.9943640
## 4                           0.9354511                         0.9941427
## 5                           1.0000000                         1.0000000
## 6                           0.8642584                         0.9995878
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                             0.9923479
## 2                                    NA
## 3                             0.9990911
## 4                             0.9999282
## 5                             1.0000000
## 6                             0.9991872
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                           0.9947563
## 2                           0.9978589
## 3                           0.9991896
## 4                           0.9989549
## 5                           1.0000000
## 6                           0.9987570
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.7768
##  1st Qu.:0.9988
##  Median :1.0000
##  Mean   :0.9939
##  3rd Qu.:1.0000
##  Max.   :1.0000
##  NA's   :3
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.7697
##  1st Qu.:0.9988
##  Median :0.9999
##  Mean   :0.9950
##  3rd Qu.:1.0000
##  Max.   :1.0000
##  NA's   :5
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.7745                        Min.   :0.5000
##  1st Qu.:0.9979                        1st Qu.:0.7406
##  Median :0.9998                        Median :0.9238
##  Mean   :0.9949                        Mean   :0.8545
##  3rd Qu.:1.0000                        3rd Qu.:0.9928
##  Max.   :1.0000                        Max.   :1.0000
##                                        NA's   :1
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.4574
##  1st Qu.:0.8966
##  Median :0.9809
##  Mean   :0.9192
##  3rd Qu.:0.9983
##  Max.   :1.0000
##  NA's   :1
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.4694                       Min.   :0.5126
##  1st Qu.:0.8952                       1st Qu.:0.7732
##  Median :0.9780                       Median :0.8870
##  Mean   :0.9230                       Mean   :0.8516
##  3rd Qu.:0.9984                       3rd Qu.:0.9545
##  Max.   :1.0000                       Max.   :0.9994
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.4610
##  1st Qu.:0.8853
##  Median :0.9745
##  Mean   :0.9208
##  3rd Qu.:0.9960
##  Max.   :1.0000
##  NA's   :2
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.4985                      Min.   :0.5000
##  1st Qu.:0.8896                      1st Qu.:0.7362
##  Median :0.9701                      Median :0.9238
```

```
##  Mean   :0.9094                     Mean   :0.8504
##  3rd Qu.:0.9931                     3rd Qu.:0.9929
##  Max.   :1.0000                     Max.   :1.0000
##                                     NA's   :1
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.4652
##  1st Qu.:0.8838
##  Median :0.9782
##  Mean   :0.9173
##  3rd Qu.:0.9984
##  Max.   :1.0000
##  NA's   :4
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.4728                     Min.   :0.8739
##  1st Qu.:0.8870                     1st Qu.:0.9994
##  Median :0.9756                     Median :1.0000
##  Mean   :0.9232                     Mean   :0.9955
##  3rd Qu.:0.9982                     3rd Qu.:1.0000
##  Max.   :1.0000                     Max.   :1.0000
##                                     NA's   :2
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.8748
##  1st Qu.:0.9990
##  Median :0.9999
##  Mean   :0.9968
##  3rd Qu.:1.0000
##  Max.   :1.0000
##  NA's   :3
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.8744
##  1st Qu.:0.9973
##  Median :0.9997
##  Mean   :0.9965
##  3rd Qu.:1.0000
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

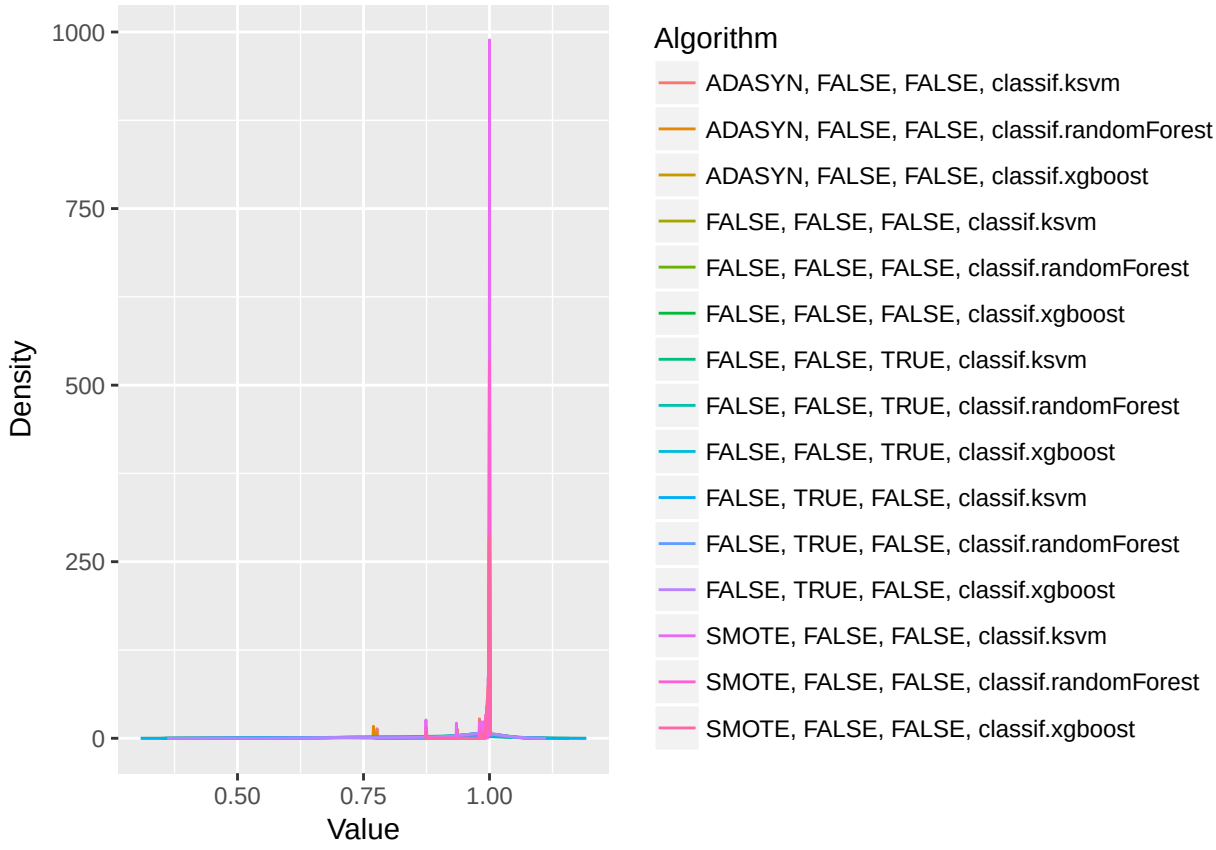
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.993896342527872"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.994989993964252"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.994856800854009"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.854453446965661"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.919196968703294"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.922973531796803"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.851639317308536"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.920783184575221"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.909429796989083"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.85042133345092"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.917320521561858"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.923202201895085"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.995543116624745"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.996818838546643"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.99649222492951"
```

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 = 561.71, 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,]                                TRUE
##  [6,]                                TRUE
##  [7,]                               FALSE
##  [8,]                               FALSE
##  [9,]                               FALSE
## [10,]                               FALSE
## [11,]                                TRUE
## [12,]                                TRUE
## [13,]                                TRUE
## [14,]                                TRUE
## [15,]                                TRUE
##       FALSE, FALSE, FALSE, classif.randomForest
##  [1,]                                      TRUE
##  [2,]                                      TRUE
##  [3,]                                      TRUE
##  [4,]                                      TRUE
##  [5,]                                     FALSE
##  [6,]                                     FALSE
##  [7,]                                      TRUE
##  [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,]                                FALSE
## [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,]                              TRUE
```

```
##  [6,]                              TRUE
##  [7,]                             FALSE
##  [8,]                              TRUE
##  [9,]                             FALSE
## [10,]                             FALSE
## [11,]                              TRUE
## [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,]                                     TRUE
##  [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,]                               FALSE                               TRUE
##  [7,]                               FALSE                              FALSE
##  [8,]                               FALSE                              FALSE
##  [9,]                               FALSE                              FALSE
## [10,]                               FALSE                              FALSE
## [11,]                               FALSE                               TRUE
## [12,]                               FALSE                               TRUE
## [13,]                                TRUE                               TRUE
## [14,]                                TRUE                               TRUE
## [15,]                                TRUE                               TRUE
##       FALSE, TRUE, FALSE, classif.randomForest
##  [1,]                                    TRUE
##  [2,]                                    TRUE
##  [3,]                                    TRUE
##  [4,]                                    TRUE
##  [5,]                                   FALSE
##  [6,]                                   FALSE
##  [7,]                                    TRUE
##  [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,]                                  FALSE
## [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
##                                   3.977273
## ADASYN, FALSE, FALSE, classif.randomForest
##                                   4.212121
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                   4.780303
##          FALSE, FALSE, FALSE, classif.ksvm
##                                  12.257576
##  FALSE, FALSE, FALSE, classif.randomForest
##                                   9.053030
##       FALSE, FALSE, FALSE, classif.xgboost
##                                   9.462121
##           FALSE, FALSE, TRUE, classif.ksvm
##                                  13.060606
##   FALSE, FALSE, TRUE, classif.randomForest
##                                   9.931818
##        FALSE, FALSE, TRUE, classif.xgboost
##                                  10.734848
##           FALSE, TRUE, FALSE, classif.ksvm
##                                  12.303030
##   FALSE, TRUE, FALSE, classif.randomForest
##                                   9.227273
##        FALSE, TRUE, FALSE, classif.xgboost
##                                   9.113636
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   3.227273
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                   4.068182
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                   4.590909
```

Plotando grafico de Critical Diference

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