R Notebook

Parametros:

Measure = Matthews correlation coefficient
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure_residual
Filter keys = imba.rate
Filter values = 0.05

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

Tratamento dos dados

Carregando data set compilado

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

```r
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

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

Filtrando o data set

```r
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         :1230   Mode :logical
##  classif.randomForest:1230   FALSE:2952
##  classif.rusboost     :   0   TRUE :738
##  classif.xgboost      :1230   NA's :0
##
##
##
##                                   measure        sampling    underbagging
##  Accuracy                        :   0   ADASYN: 738   Mode :logical
##  Area under the curve            :   0   FALSE :2214   FALSE:2952
##  F1 measure                      :   0   SMOTE : 738   TRUE :738
##  G-mean                          :   0                 NA's :0
##  Matthews correlation coefficient:3690
##
##
##  tuning_measure     holdout_measure    holdout_measure_residual
##  Min.   :-0.1277   Min.   :-0.21201   Min.   :-0.45710
##  1st Qu.: 0.3764   1st Qu.: 0.06131   1st Qu.: 0.05637
##  Median : 0.8057   Median : 0.55190   Median : 0.23378
##  Mean   : 0.6629   Mean   : 0.49274   Mean   : 0.32193
##  3rd Qu.: 0.9728   3rd Qu.: 0.82456   3rd Qu.: 0.56442
##  Max.   : 1.0000   Max.   : 1.00000   Max.   : 1.00000
##  NA's   :54        NA's   :54         NA's   :54
##  iteration_count         dataset       imba.rate
##  Min.   :1      abalone      :  45   Min.   :0.05
##  1st Qu.:1      adult        :  45   1st Qu.:0.05
##  Median :2      annealing    :  45   Median :0.05
##  Mean   :2      arrhythmia   :  45   Mean   :0.05
##  3rd Qu.:3      balance-scale:  45   3rd Qu.:0.05
##  Max.   :3      bank         :  45   Max.   :0.05
##  NA's   :54     (Other)      :3420
```

Computando as médias das iteracoes

```r
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

```r
# 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$performan

# 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] 82 15
```

```r
# Renomeando a variavel
df = df_tec_wide_residual

head(df)
```

```
##   ADASYN, FALSE, FALSE, classif.ksvm
## 1                        0.09430275
## 2                        0.23594212
## 3                        0.39737200
## 4                        0.00000000
## 5                        0.14032488
## 6                        0.05815156
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                 0.12967379
## 2                                         NA
## 3                                 0.78835469
## 4                                 0.22905351
## 5                                 0.06984462
## 6                                 0.35650890
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.08434280                       0.08009145
## 2                            0.43217195                       0.29868169
## 3                            0.78819309                       0.31222405
## 4                            0.30572505                       0.00000000
## 5                            0.08978642                       0.14801251
## 6                            0.32094006                       0.19892100
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                0.00000000
## 2                                0.38674028
```

```
## 3                                     0.78287453
## 4                                     0.24122912
## 5                                     0.09485503
## 6                                     0.22645356
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                         0.01770510                        0.20313757
## 2                         0.38214676                        0.55327083
## 3                         0.68241727                        0.37539770
## 4                         0.36622368                        0.07136273
## 5                         0.08978642                        0.11577975
## 6                         0.28503349                        0.36690665
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                         0.2713467
## 2                               NA
## 3                         0.7886780
## 4                         0.2151090
## 5                         0.1529797
## 6                         0.5856299
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                         0.2861170                        0.07628792
## 2                         0.6615847                        0.29457094
## 3                         0.6790479                        0.36193425
## 4                         0.2275119                        0.00000000
## 5                         0.0429144                        0.14801251
## 6                         0.5743372                        0.14294212
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                         0.0000000
## 2                               NA
## 3                         0.7789185
## 4                         0.3065597
## 5                         0.1033385
## 6                         0.2264536
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                        0.007719047                        0.11483245
## 2                        0.373972504                        0.24240609
## 3                        0.654695739                        0.41403016
## 4                        0.355877657                        0.00000000
## 5                        0.089786421                        0.09045042
## 6                        0.247234213                        0.11319614
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                         0.11153157
## 2                         0.42923182
## 3                         0.82047885
## 4                         0.25552467
## 5                         0.07474221
## 6                         0.34517556
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                         0.07239314
## 2                         0.43088774
## 3                         0.76448714
## 4                         0.34740267
## 5                         0.07869554
## 6                         0.33380303
```

```r
summary(df)
```

```
##   ADASYN, FALSE, FALSE, classif.ksvm
##   Min.    :-0.26464
##   1st Qu.: 0.02509
##   Median : 0.11755
##   Mean    : 0.21514
##   3rd Qu.: 0.31422
##   Max.    : 0.98633
##   NA's    :1
##   ADASYN, FALSE, FALSE, classif.randomForest
##   Min.    :-0.2970
##   1st Qu.: 0.1186
##   Median : 0.3001
##   Mean    : 0.3606
##   3rd Qu.: 0.5600
##   Max.    : 0.9782
##   NA's    :7
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##   Min.    :-0.3498                      Min.    :-0.26935
##   1st Qu.: 0.1334                       1st Qu.: 0.00000
##   Median : 0.3270                       Median : 0.08951
##   Mean    : 0.3803                      Mean    : 0.19345
##   3rd Qu.: 0.6308                       3rd Qu.: 0.29753
##   Max.    : 0.9868                      Max.    : 0.99489
##
##   FALSE, FALSE, FALSE, classif.randomForest
##   Min.    :-0.34920
##   1st Qu.: 0.07321
##   Median : 0.24615
##   Mean    : 0.33689
##   3rd Qu.: 0.56177
##   Max.    : 0.96884
##
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##   Min.    :-0.39928                     Min.    :-0.2297
##   1st Qu.: 0.08988                      1st Qu.: 0.1054
##   Median : 0.29534                      Median : 0.2537
##   Mean    : 0.35279                     Mean    : 0.3227
##   3rd Qu.: 0.56354                      3rd Qu.: 0.5528
##   Max.    : 0.98175                     Max.    : 0.9863
##
##   FALSE, FALSE, TRUE, classif.randomForest
##   Min.    :-0.3249
##   1st Qu.: 0.1655
##   Median : 0.3731
##   Mean    : 0.4215
##   3rd Qu.: 0.7123
##   Max.    : 0.9688
##   NA's    :3
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##   Min.    :-0.3331                     Min.    :-0.26935
##   1st Qu.: 0.1689                      1st Qu.: 0.00000
##   Median : 0.3883                      Median : 0.08749
```

```
##   Mean   : 0.4089                    Mean    : 0.18691
##   3rd Qu.: 0.6770                    3rd Qu.: 0.29662
##   Max.   : 0.9636                    Max.    : 0.99489
##
##   FALSE, TRUE, FALSE, classif.randomForest
##   Min.   :-0.34302
##   1st Qu.: 0.08386
##   Median : 0.25441
##   Mean   : 0.34523
##   3rd Qu.: 0.57812
##   Max.   : 1.00000
##   NA's   :3
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##   Min.   :-0.39928                    Min.    :-0.23791
##   1st Qu.: 0.09907                    1st Qu.: 0.02052
##   Median : 0.27821                    Median : 0.11410
##   Mean   : 0.34301                    Mean    : 0.20912
##   3rd Qu.: 0.56347                    3rd Qu.: 0.31721
##   Max.   : 1.00000                    Max.    : 0.96126
##
##   SMOTE, FALSE, FALSE, classif.randomForest
##   Min.   :-0.2854
##   1st Qu.: 0.1260
##   Median : 0.3203
##   Mean   : 0.3777
##   3rd Qu.: 0.6254
##   Max.   : 0.9792
##   NA's   :4
##   SMOTE, FALSE, FALSE, classif.xgboost
##   Min.   :-0.3323
##   1st Qu.: 0.1402
##   Median : 0.3270
##   Mean   : 0.3839
##   3rd Qu.: 0.6440
##   Max.   : 0.9848
##
```

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 = 0.215139691812362"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.360639993535549"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.380323512172679"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.193454681939607"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.336890219023616"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.352786996234241"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.32272866048926"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.421478217086802"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.408924428963741"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.186906720798796"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.345225544655046"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.343012116409775"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.209119896222376"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.377665371388864"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.38389880379007"
```

Fazendo teste de normalidade

```
plotDensities(data = na.omit(df))
```
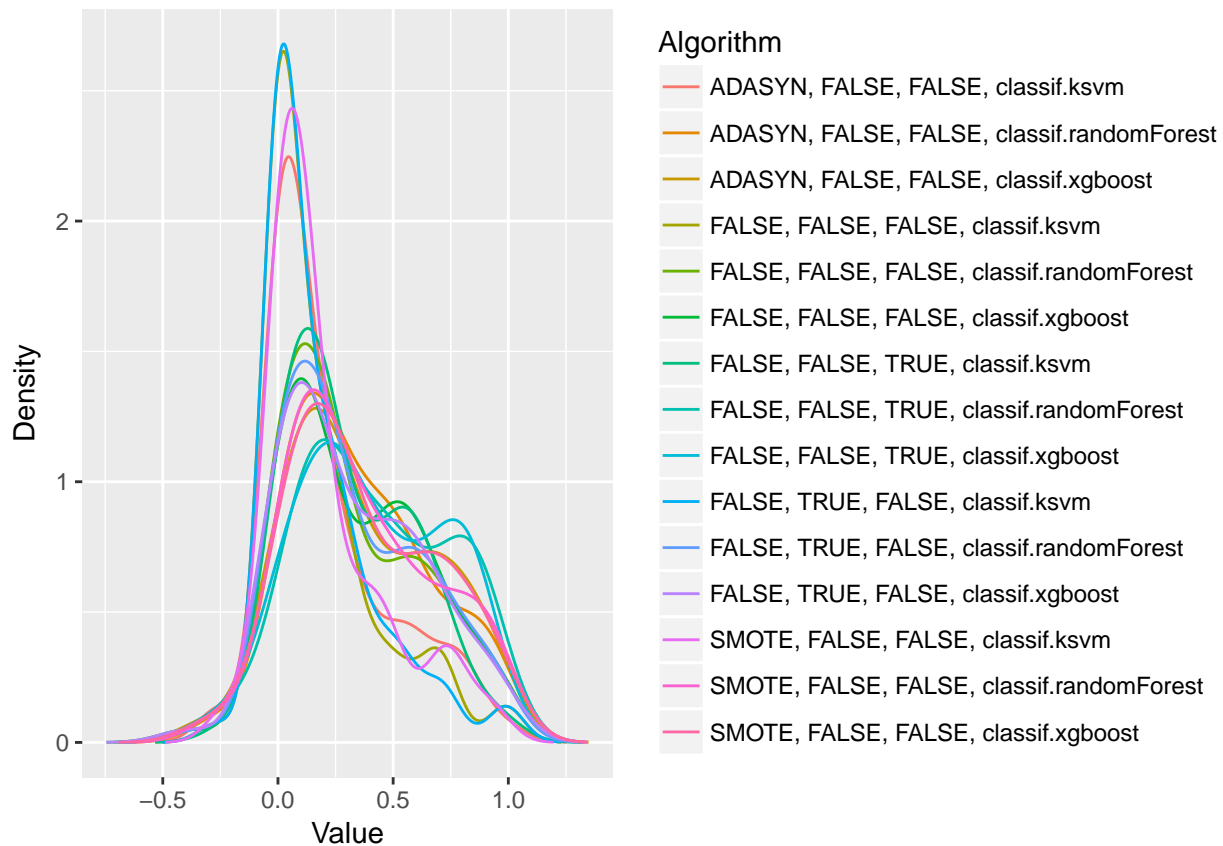

Testando as diferencas

```
friedmanTest(df)
```

```
##
##  Friedman's rank sum test
##
## data:  df
## Friedman's chi-squared = 197.88, 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,]                             FALSE
##  [6,]                              TRUE
##  [7,]                              TRUE
##  [8,]                              TRUE
##  [9,]                              TRUE
## [10,]                             FALSE
## [11,]                             FALSE
## [12,]                             FALSE
## [13,]                             FALSE
## [14,]                              TRUE
## [15,]                              TRUE
##         ADASYN, FALSE, FALSE, classif.randomForest
##  [1,]                                     TRUE
##  [2,]                                    FALSE
##  [3,]                                    FALSE
##  [4,]                                     TRUE
##  [5,]                                    FALSE
##  [6,]                                    FALSE
##  [7,]                                    FALSE
##  [8,]                                    FALSE
##  [9,]                                    FALSE
## [10,]                                     TRUE
## [11,]                                    FALSE
## [12,]                                    FALSE
## [13,]                                     TRUE
## [14,]                                    FALSE
## [15,]                                    FALSE
##         ADASYN, FALSE, FALSE, classif.xgboost
##  [1,]                                 TRUE
##  [2,]                                FALSE
##  [3,]                                FALSE
##  [4,]                                 TRUE
##  [5,]                                FALSE
##  [6,]                                FALSE
##  [7,]                                FALSE
##  [8,]                                FALSE
##  [9,]                                FALSE
## [10,]                                 TRUE
## [11,]                                FALSE
## [12,]                                FALSE
## [13,]                                 TRUE
## [14,]                                FALSE
## [15,]                                FALSE
```

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

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

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

Plotando os ranks

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

```
##          ADASYN, FALSE, FALSE, classif.ksvm
##                                    10.280488
## ADASYN, FALSE, FALSE, classif.randomForest
##                                     7.237805
##       ADASYN, FALSE, FALSE, classif.xgboost
##                                     6.207317
##           FALSE, FALSE, FALSE, classif.ksvm
##                                    10.810976
##  FALSE, FALSE, FALSE, classif.randomForest
##                                     8.091463
##       FALSE, FALSE, FALSE, classif.xgboost
##                                     7.609756
##           FALSE, FALSE, TRUE, classif.ksvm
##                                     7.451220
##   FALSE, FALSE, TRUE, classif.randomForest
##                                     5.518293
##       FALSE, FALSE, TRUE, classif.xgboost
##                                     6.030488
##           FALSE, TRUE, FALSE, classif.ksvm
##                                    10.932927
##   FALSE, TRUE, FALSE, classif.randomForest
##                                     8.310976
##       FALSE, TRUE, FALSE, classif.xgboost
##                                     8.341463
##           SMOTE, FALSE, FALSE, classif.ksvm
##                                    10.439024
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                     6.993902
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                     5.743902
```

Plotando grafico de Critical Diference

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