R Notebook

Parametros:

Measure = Matthews correlation coefficient
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure_residual
Filter keys = NULL
Filter values = NULL

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

Tratamento dos dados

Carregando data set compilado

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

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

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

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

Filtrando pela metrica

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

Filtrando o data set

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

summary(ds)
```

```
##                  learner     weight_space
##  classif.ksvm        :3420   Mode :logical
##  classif.randomForest:3420   FALSE:8208
##  classif.rusboost    :   0   TRUE :2052
##  classif.xgboost     :3420   NA's :0
##
##
##
##                             measure        sampling    underbagging
##  Accuracy                       :    0   ADASYN:2052   Mode :logical
##  Area under the curve           :    0   FALSE :6156   FALSE:8208
##  F1 measure                     :    0   SMOTE :2052   TRUE :2052
##  G-mean                         :    0                 NA's :0
##  Matthews correlation coefficient:10260
##
##
##  tuning_measure    holdout_measure   holdout_measure_residual
##  Min.   :-0.1277   Min.   :-0.2120   Min.   :-0.46576
##  1st Qu.: 0.3307   1st Qu.: 0.0000   1st Qu.: 0.03886
##  Median : 0.8174   Median : 0.4907   Median : 0.21377
##  Mean   : 0.6548   Mean   : 0.4657   Mean   : 0.30966
##  3rd Qu.: 0.9890   3rd Qu.: 0.8152   3rd Qu.: 0.53139
##  Max.   : 1.0000   Max.   : 1.0000   Max.   : 1.00000
##  NA's   :225       NA's   :225       NA's   :225
##  iteration_count                      dataset       imba.rate
##  Min.   :1       abalone                  : 180   Min.   :0.0010
##  1st Qu.:1       adult                    : 180   1st Qu.:0.0100
##  Median :2       bank                     : 180   Median :0.0300
##  Mean   :2       car                      : 180   Mean   :0.0286
##  3rd Qu.:3       cardiotocography-10clases: 180   3rd Qu.:0.0500
##  Max.   :3       cardiotocography-3clases : 180   Max.   :0.0500
##  NA's   :225     (Other)                  :9180
```

Computando as médias das iteracoes

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

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

Criando dataframe

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

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

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

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

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

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

```
## [1] 228  15
```

```
# Renomeando a variavel
df = df_tec_wide_residual

head(df)
```

```
##   ADASYN, FALSE, FALSE, classif.ksvm
## 1                          0.07016844
## 2                          0.07016844
## 3                          0.10164023
## 4                          0.09430275
## 5                          0.10155397
## 6                          0.10155397
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                 0.04455766
## 2                                 0.04455766
## 3                                 0.11489099
## 4                                 0.12967379
## 5                                 0.20471450
## 6                                 0.19368796
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.02455997                          0.03701282
## 2                            0.02455997                          0.03701282
## 3                            0.07341364                          0.05529460
## 4                            0.08434280                          0.08009145
## 5                            0.30955469                          0.09710987
## 6                            0.30955469                          0.09710987
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                               0.005300235
## 2                               0.005300235
```

```
## 3                                0.000000000
## 4                                0.000000000
## 5                                0.323620415
## 6                                         NA
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                          0.005300235                         0.1761902
## 2                          0.005300235                         0.1761902
## 3                          0.021544091                         0.2486599
## 4                          0.017705096                         0.2031376
## 5                          0.302563452                         0.3763465
## 6                          0.302563452                         0.3763465
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                                0.2150826
## 2                                0.2150826
## 3                                0.2697597
## 4                                0.2713467
## 5                                0.6258826
## 6                                0.6258826
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                          0.2024389                        0.002457572
## 2                          0.2024389                        0.002457572
## 3                          0.2477941                        0.054696787
## 4                          0.2861170                        0.076287922
## 5                          0.6245448                        0.108197202
## 6                          0.6245448                        0.108197202
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                               0.005300235
## 2                               0.005300235
## 3                               0.005292689
## 4                               0.000000000
## 5                                        NA
## 6                                        NA
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                         0.005300235                         0.06360859
## 2                         0.005300235                         0.06360859
## 3                         0.005382185                         0.07731173
## 4                         0.007719047                         0.11483245
## 5                         0.302725524                         0.10857900
## 6                         0.302725524                         0.10857900
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                                0.06250968
## 2                                0.06250968
## 3                                0.12042316
## 4                                0.11153157
## 5                                        NA
## 6                                0.22879628
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                          0.02420618
## 2                          0.02420618
## 3                          0.08605067
## 4                          0.07239314
## 5                          0.29725558
## 6                          0.29725558
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :-0.26464
##  1st Qu.: 0.00000
##  Median : 0.07768
##  Mean   : 0.18062
##  3rd Qu.: 0.27174
##  Max.   : 0.98633
##  NA's   :7
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :-0.2970
##  1st Qu.: 0.0928
##  Median : 0.2709
##  Mean   : 0.3391
##  3rd Qu.: 0.5375
##  Max.   : 0.9782
##  NA's   :25
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :-0.3498                       Min.   :-0.26935
##  1st Qu.: 0.1146                       1st Qu.: 0.00000
##  Median : 0.3567                       Median : 0.08433
##  Mean   : 0.3848                       Mean   : 0.19174
##  3rd Qu.: 0.5931                       3rd Qu.: 0.27632
##  Max.   : 0.9974                       Max.   : 0.99489
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :-0.34919
##  1st Qu.: 0.05113
##  Median : 0.22407
##  Mean   : 0.30998
##  3rd Qu.: 0.52248
##  Max.   : 1.00000
##  NA's   :6
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :-0.3993                      Min.   :-0.22965
##  1st Qu.: 0.0723                      1st Qu.: 0.09735
##  Median : 0.2587                      Median : 0.25955
##  Mean   : 0.3317                      Mean   : 0.31703
##  3rd Qu.: 0.5229                      3rd Qu.: 0.53837
##  Max.   : 0.9974                      Max.   : 0.98633
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :-0.3249
##  1st Qu.: 0.1536
##  Median : 0.4176
##  Mean   : 0.4396
##  3rd Qu.: 0.7456
##  Max.   : 0.9688
##  NA's   :6
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :-0.3331                     Min.   :-0.26935
##  1st Qu.: 0.1607                     1st Qu.: 0.00000
##  Median : 0.4246                     Median : 0.07843
```

```
##  Mean   : 0.4251                    Mean   : 0.18542
##  3rd Qu.: 0.7047                    3rd Qu.: 0.26770
##  Max.   : 0.9636                    Max.   : 0.99489
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :-0.34302
##  1st Qu.: 0.04879
##  Median : 0.21926
##  Mean   : 0.31299
##  3rd Qu.: 0.54270
##  Max.   : 1.00000
##  NA's   :11
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :-0.39928                   Min.   :-0.237905
##  1st Qu.: 0.06576                   1st Qu.: 0.001932
##  Median : 0.26070                   Median : 0.089675
##  Mean   : 0.32776                   Mean   : 0.175722
##  3rd Qu.: 0.51001                   3rd Qu.: 0.212975
##  Max.   : 1.00000                   Max.   : 0.973741
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :-0.31686
##  1st Qu.: 0.09336
##  Median : 0.27270
##  Mean   : 0.34339
##  3rd Qu.: 0.52838
##  Max.   : 0.99743
##  NA's   :20
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :-0.3323
##  1st Qu.: 0.1180
##  Median : 0.3591
##  Mean   : 0.3857
##  3rd Qu.: 0.6141
##  Max.   : 1.0000
##
```

Verificando a média de cada coluna selecionada

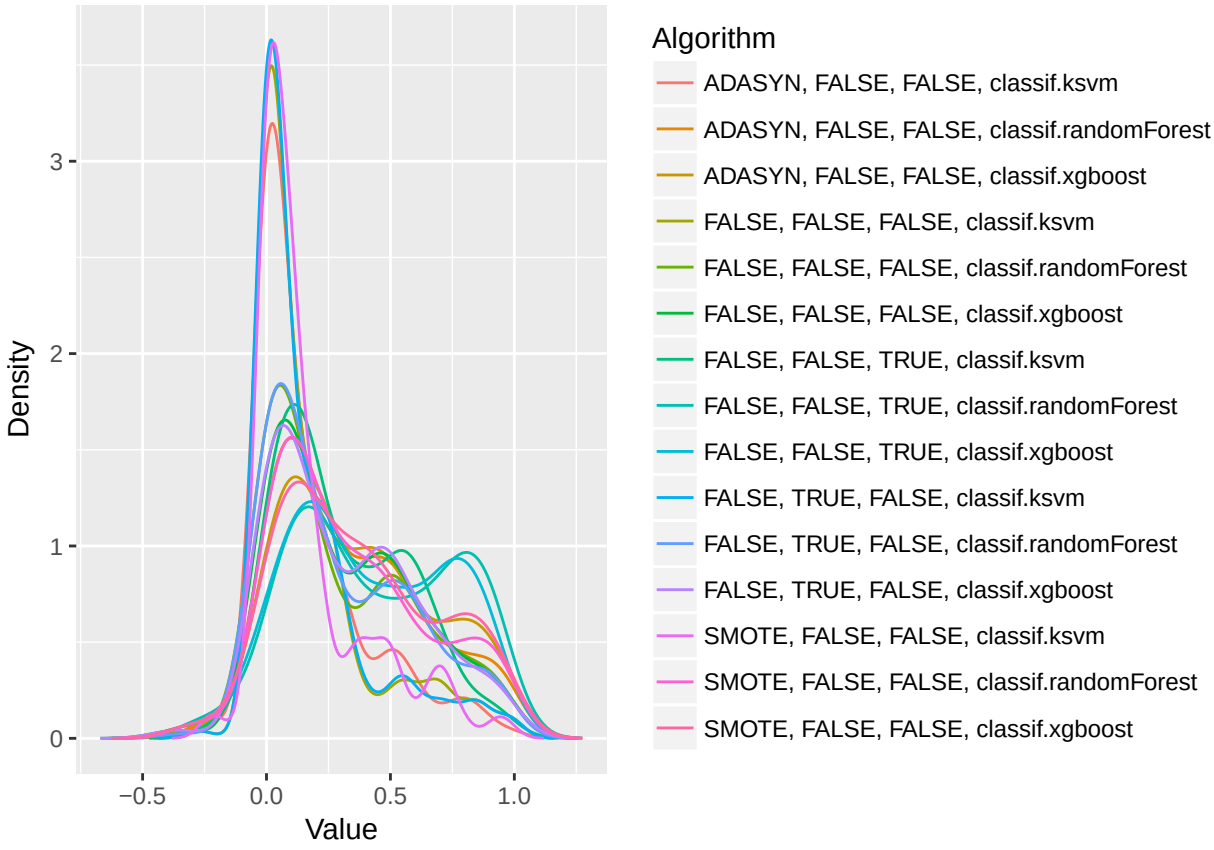
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.180617652152928"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.339060159434773"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.384772769319656"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.19173654981395"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.309979346079034"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.331714283128261"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.317030859125428"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.4396094715883"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.425088546457107"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.185423789347126"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.312994297886636"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.327758676268591"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.175721887326245"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.343393298166142"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.385749642682616"
```

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 = 733.43, df = 14, p-value < 2.2e-16
```

Testando as diferencas par a par

```
test <- nemenyiTest (df, alpha=0.05)
abs(test$diff.matrix) > test$statistic
```

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

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

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

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

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

Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                   11.164474
## ADASYN, FALSE, FALSE, classif.randomForest
##                                    7.541667
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                    5.379386
##          FALSE, FALSE, FALSE, classif.ksvm
##                                   10.697368
##  FALSE, FALSE, FALSE, classif.randomForest
##                                    8.359649
##       FALSE, FALSE, FALSE, classif.xgboost
##                                    7.486842
##           FALSE, FALSE, TRUE, classif.ksvm
##                                    7.732456
##   FALSE, FALSE, TRUE, classif.randomForest
##                                    5.081140
##        FALSE, FALSE, TRUE, classif.xgboost
##                                    5.344298
##           FALSE, TRUE, FALSE, classif.ksvm
##                                   10.850877
##   FALSE, TRUE, FALSE, classif.randomForest
##                                    8.739035
##        FALSE, TRUE, FALSE, classif.xgboost
##                                    7.857456
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   10.861842
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                    7.513158
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                    5.390351
```

Plotando grafico de Critical Diference

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