R Notebook

Parametros:

Measure = G-mean
Columns = sampling, weight_space, underbagging, learner
Performance = tuning_measure
Filter keys = imba.rate
Filter values = 0.05

```
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        :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                          :3690                 NA's :0
##  Matthews correlation coefficient:   0
##
##
##  tuning_measure   holdout_measure  holdout_measure_residual
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.6329   1st Qu.:0.3162   1st Qu.:0.2321
##  Median :0.9254   Median :0.7412   Median :0.5564
##  Mean   :0.7606   Mean   :0.6130   Mean   :0.5202
##  3rd Qu.:0.9872   3rd Qu.:0.9487   3rd Qu.:0.8165
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  NA's   :39       NA's   :39       NA's   :39
##  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   :39      (Other)      :3420
```

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] 82 15
```

```
# Renomeando a variavel
df = df_tec_wide_residual

head(df)
```

```
##   ADASYN, FALSE, FALSE, classif.ksvm
## 1                          0.9447230
## 2                          0.9607136
## 3                          0.9523195
## 4                          0.8260894
## 5                          1.0000000
## 6                          0.9772330
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.9383475
## 2                                  0.9682740
## 3                                  0.9931647
## 4                                  0.9932271
## 5                                  1.0000000
## 6                                  0.9745700
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.9572437                          0.3574773
## 2                            0.9734134                          0.4937165
## 3                            0.9880828                          0.4369027
## 4                            0.9822340                          0.0000000
## 5                            1.0000000                          1.0000000
## 6                            0.9733211                          0.2658176
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                0.01755665
## 2                                0.56635504
```

```
## 3                                0.82065958
## 4                                0.86595513
## 5                                1.00000000
## 6                                0.33705687
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                           0.07718019                         0.6189515
## 2                           0.58113759                         0.7772247
## 3                           0.75331643                         0.6839646
## 4                           0.78727363                         0.4579778
## 5                           1.00000000                         0.9825026
## 6                           0.39634228                         0.6629088
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                                0.6757496
## 2                                       NA
## 3                                0.9133856
## 4                                0.9107110
## 5                                0.9967793
## 6                                0.8127455
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.6717735                         0.3446430
## 2                           0.8330637                         0.4725211
## 3                           0.8754956                         0.4610769
## 4                           0.9364544                         0.0000000
## 5                           0.9692412                         1.0000000
## 6                           0.8110503                         0.2219618
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                              0.01755665
## 2                              0.57509025
## 3                              0.79932808
## 4                              0.69289848
## 5                              1.00000000
## 6                              0.33705687
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                          0.08755153                          0.9444674
## 2                          0.57539621                          0.9604414
## 3                          0.72967386                          0.9605666
## 4                          0.79565018                          0.7900353
## 5                          1.00000000                          1.0000000
## 6                          0.38143211                          0.9851849
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                                 0.9409278
## 2                                        NA
## 3                                 0.9880449
## 4                                 0.9932140
## 5                                 1.0000000
## 6                                 0.9764012
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                            0.9537588
## 2                            0.9744999
## 3                            0.9898346
## 4                            0.9856415
## 5                            1.0000000
## 6                            0.9719978
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.7000
##  1st Qu.:0.9564
##  Median :0.9790
##  Mean   :0.9617
##  3rd Qu.:0.9948
##  Max.   :1.0000
##  NA's   :1
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.6918
##  1st Qu.:0.9706
##  Median :0.9905
##  Mean   :0.9739
##  3rd Qu.:0.9983
##  Max.   :1.0000
##  NA's   :3
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.6674                        Min.   :0.0000
##  1st Qu.:0.9667                        1st Qu.:0.0000
##  Median :0.9875                        Median :0.2836
##  Mean   :0.9699                        Mean   :0.3505
##  3rd Qu.:0.9962                        3rd Qu.:0.6326
##  Max.   :1.0000                        Max.   :1.0000
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.3160
##  Median :0.6992
##  Mean   :0.5966
##  3rd Qu.:0.8875
##  Max.   :1.0000
##  NA's   :1
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.0000                       Min.   :0.3612
##  1st Qu.:0.4010                       1st Qu.:0.6306
##  Median :0.7388                       Median :0.7713
##  Mean   :0.6385                       Mean   :0.7361
##  3rd Qu.:0.8995                       3rd Qu.:0.8666
##  Max.   :1.0000                       Max.   :0.9961
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.5061
##  1st Qu.:0.7931
##  Median :0.9024
##  Mean   :0.8636
##  3rd Qu.:0.9644
##  Max.   :1.0000
##  NA's   :3
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.4794                      Min.   :0.0000
##  1st Qu.:0.7597                      1st Qu.:0.0000
##  Median :0.9002                      Median :0.2836
```

```
##  Mean   :0.8499                     Mean   :0.3399
##  3rd Qu.:0.9583                     3rd Qu.:0.5445
##  Max.   :1.0000                     Max.   :1.0000
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.3695
##  Median :0.6614
##  Mean   :0.6055
##  3rd Qu.:0.9035
##  Max.   :1.0000
##  NA's   :1
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000                     Min.   :0.7097
##  1st Qu.:0.4147                     1st Qu.:0.9558
##  Median :0.7152                     Median :0.9788
##  Mean   :0.6354                     Mean   :0.9598
##  3rd Qu.:0.8878                     3rd Qu.:0.9931
##  Max.   :1.0000                     Max.   :1.0000
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.6760
##  1st Qu.:0.9728
##  Median :0.9916
##  Mean   :0.9749
##  3rd Qu.:0.9976
##  Max.   :1.0000
##  NA's   :4
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.6787
##  1st Qu.:0.9683
##  Median :0.9873
##  Mean   :0.9727
##  3rd Qu.:0.9965
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

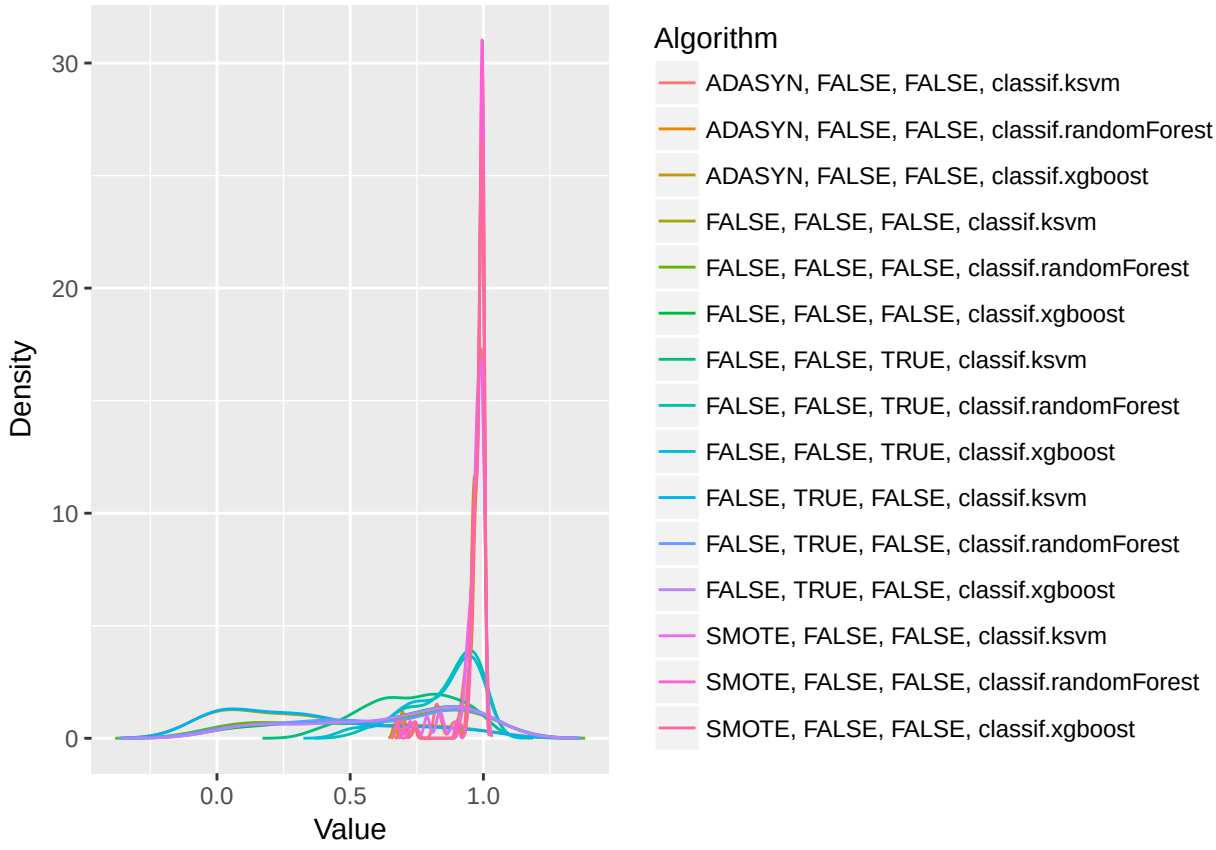
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.961690950079547"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.973873449178404"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.969865867217637"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.350530714908082"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.596642762531993"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.638535274233888"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.736136603391947"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.863559657214054"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.849937498115963"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.339937487852007"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.605544309315937"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.635447296447323"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.959768951755182"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.974853836432435"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.972732536694603"
```

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 = 822.9, 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,]                               TRUE
##  [8,]                               TRUE
##  [9,]                               TRUE
## [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,]                                      TRUE
##  [9,]                                      TRUE
## [10,]                                     FALSE
## [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,]                                FALSE
##  [8,]                                 TRUE
##  [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,]                              TRUE
##  [5,]                             FALSE
```

```
##  [6,]                            FALSE
##  [7,]                            FALSE
##  [8,]                             TRUE
##  [9,]                            FALSE
## [10,]                             TRUE
## [11,]                            FALSE
## [12,]                            FALSE
## [13,]                             TRUE
## [14,]                             TRUE
## [15,]                             TRUE
##       FALSE, FALSE, TRUE, classif.randomForest
##  [1,]                                     TRUE
##  [2,]                                     TRUE
##  [3,]                                     TRUE
##  [4,]                                     TRUE
##  [5,]                                     TRUE
##  [6,]                                     TRUE
##  [7,]                                     TRUE
##  [8,]                                    FALSE
##  [9,]                                    FALSE
## [10,]                                     TRUE
## [11,]                                     TRUE
## [12,]                                     TRUE
## [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,]                                TRUE                              FALSE
##  [5,]                                TRUE                              FALSE
##  [6,]                                TRUE                               TRUE
##  [7,]                               FALSE                               TRUE
##  [8,]                               FALSE                               TRUE
##  [9,]                               FALSE                               TRUE
## [10,]                                TRUE                              FALSE
## [11,]                                TRUE                              FALSE
## [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,]                                     TRUE
##  [9,]                                     TRUE
## [10,]                                    FALSE
## [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,]                                  FALSE
##  [8,]                                   TRUE
##  [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
##                                   4.603659
## ADASYN, FALSE, FALSE, classif.randomForest
##                                   3.536585
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                   3.975610
##          FALSE, FALSE, FALSE, classif.ksvm
##                                  13.347561
##  FALSE, FALSE, FALSE, classif.randomForest
##                                  11.414634
##       FALSE, FALSE, FALSE, classif.xgboost
##                                  10.506098
##           FALSE, FALSE, TRUE, classif.ksvm
##                                  10.121951
##   FALSE, FALSE, TRUE, classif.randomForest
##                                   7.615854
##        FALSE, FALSE, TRUE, classif.xgboost
##                                   7.926829
##           FALSE, TRUE, FALSE, classif.ksvm
##                                  13.493902
##   FALSE, TRUE, FALSE, classif.randomForest
##                                  11.396341
##        FALSE, TRUE, FALSE, classif.xgboost
##                                  10.652439
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   4.213415
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                   3.536585
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                   3.658537
```

Plotando grafico de Critical Diference

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