R Notebook

Parametros:

Measure = Matthews correlation coefficient
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                          :   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

```
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                          0.8927019
## 2                          0.9218761
## 3                          0.9080674
## 4                          0.7246147
## 5                          1.0000000
## 6                          0.9556729
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.8782120
## 2                                         NA
## 3                                  0.9865259
## 4                                  0.9866066
## 5                                  1.0000000
## 6                                  0.9496729
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                             0.9147539                          0.08104013
## 2                             0.9469375                          0.24483618
## 3                             0.9764872                          0.29346248
## 4                             0.9685016                          0.00000000
## 5                             1.0000000                          1.00000000
## 6                             0.9467044                          0.10956287
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                 0.0000000
## 2                                 0.4854582
```

```
## 3                                0.7292073
## 4                                0.7804829
## 5                                1.0000000
## 6                                0.1939237
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                           0.02330986                       0.10888103
## 2                           0.48961416                       0.27697309
## 3                           0.68320251                       0.30736929
## 4                           0.67382183                       0.08270852
## 5                           1.00000000                       0.90275267
## 6                           0.21903979                       0.17235130
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                               0.1599893
## 2                                      NA
## 3                               0.4720989
## 4                               0.5621467
## 5                               0.9488613
## 6                               0.3217028
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.1592943                        0.07590171
## 2                           0.3418694                        0.27339314
## 3                           0.4237077                        0.34428936
## 4                           0.6124266                        0.00000000
## 5                           0.7487916                        1.00000000
## 6                           0.3168558                        0.09975332
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                               0.0000000
## 2                                      NA
## 3                               0.7259167
## 4                               0.6469110
## 5                               1.0000000
## 6                               0.1939237
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                          0.02060981                         0.8913934
## 2                          0.49816198                         0.9208809
## 3                          0.64962959                         0.9241607
## 4                          0.68830632                         0.6839427
## 5                          1.00000000                         1.0000000
## 6                          0.21542464                         0.9705928
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                                 0.8809365
## 2                                 0.9358251
## 3                                 0.9759713
## 4                                 0.9867289
## 5                                 1.0000000
## 6                                 0.9530885
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                            0.9077481
## 2                            0.9494704
## 3                            0.9796283
## 4                            0.9554301
## 5                            1.0000000
## 6                            0.9440836
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.5483
##  1st Qu.:0.9196
##  Median :0.9590
##  Mean   :0.9330
##  3rd Qu.:0.9899
##  Max.   :1.0000
##  NA's   :1
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.4102
##  1st Qu.:0.9455
##  Median :0.9826
##  Mean   :0.9519
##  3rd Qu.:0.9967
##  Max.   :1.0000
##  NA's   :7
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.4273                        Min.   :-0.00263
##  1st Qu.:0.9338                        1st Qu.: 0.00000
##  Median :0.9753                        Median : 0.22396
##  Mean   :0.9460                        Mean   : 0.30194
##  3rd Qu.:0.9923                        3rd Qu.: 0.49531
##  Max.   :1.0000                        Max.   : 1.00000
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.2241
##  Median :0.6288
##  Mean   :0.5636
##  3rd Qu.:0.8255
##  Max.   :1.0000
##
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.0000                       Min.   :0.02716
##  1st Qu.:0.2768                       1st Qu.:0.21828
##  Median :0.6695                       Median :0.41094
##  Mean   :0.5872                       Mean   :0.43668
##  3rd Qu.:0.8344                       3rd Qu.:0.65310
##  Max.   :1.0000                       Max.   :0.98137
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.01566
##  1st Qu.:0.29475
##  Median :0.50366
##  Mean   :0.50843
##  3rd Qu.:0.72086
##  Max.   :1.00000
##  NA's   :3
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.004119                    Min.   :-0.00263
##  1st Qu.:0.257100                    1st Qu.: 0.00000
##  Median :0.450802                    Median : 0.22083
```

```
##  Mean   :0.476384                    Mean   : 0.29654
##  3rd Qu.:0.699798                    3rd Qu.: 0.48690
##  Max.   :1.000000                    Max.   : 1.00000
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :-0.002021
##  1st Qu.: 0.212861
##  Median : 0.611868
##  Mean   : 0.562607
##  3rd Qu.: 0.873895
##  Max.   : 1.000000
##  NA's   :3
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000                      Min.   :0.5423
##  1st Qu.:0.2843                      1st Qu.:0.9196
##  Median :0.6515                      Median :0.9589
##  Mean   :0.5801                      Mean   :0.9296
##  3rd Qu.:0.8382                      3rd Qu.:0.9864
##  Max.   :1.0000                      Max.   :1.0000
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.4552
##  1st Qu.:0.9446
##  Median :0.9813
##  Mean   :0.9527
##  3rd Qu.:0.9948
##  Max.   :1.0000
##  NA's   :4
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.3909
##  1st Qu.:0.9363
##  Median :0.9757
##  Mean   :0.9487
##  3rd Qu.:0.9929
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

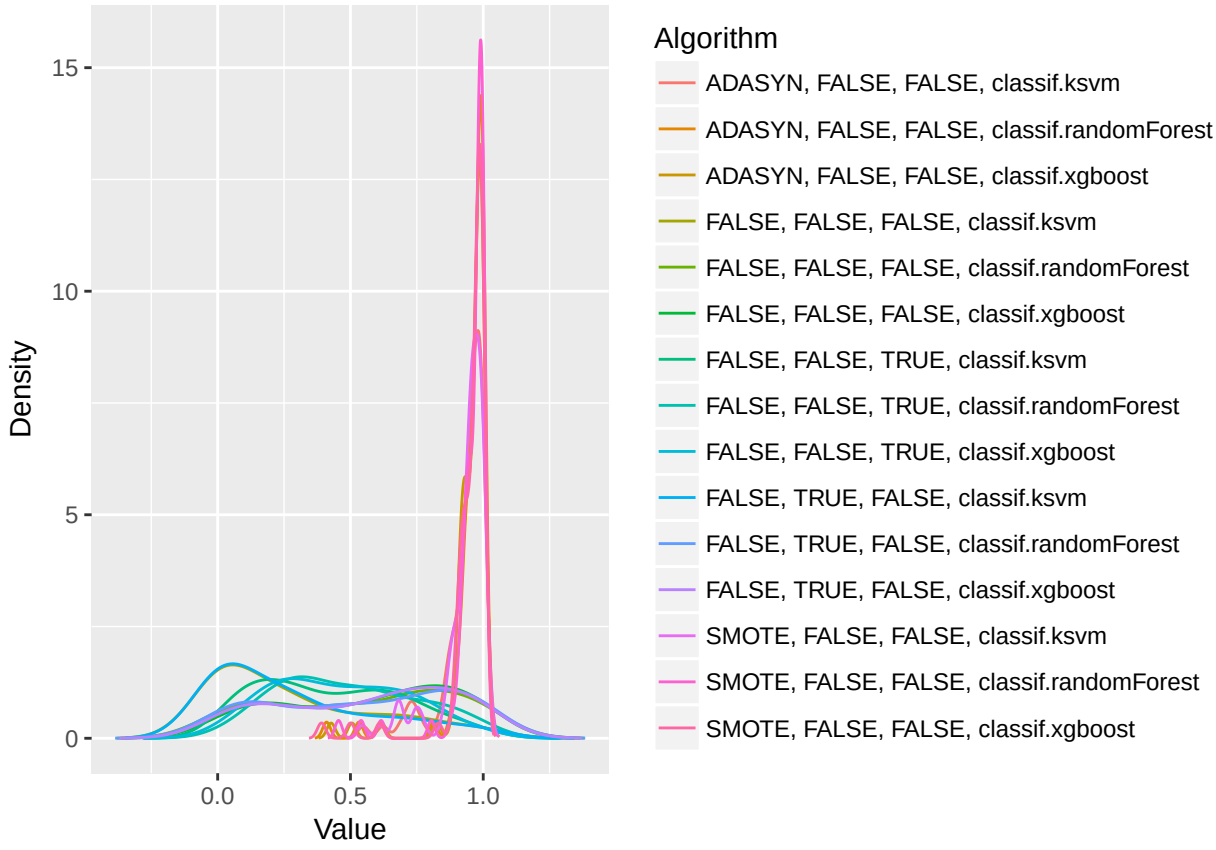
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.932983074715977"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.951918405863464"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.945978658855029"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.301939372449071"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.563594360772488"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.587222752029455"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.436678945879101"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.508434848734606"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.476383878748302"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.296536950997029"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.562607316588684"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.580121740445508"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.929603142989257"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.952742208547565"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.948749596253218"
```

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 = 761.83, 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,]                                     FALSE
##  [8,]                                     FALSE
##  [9,]                                     FALSE
## [10,]                                      TRUE
## [11,]                                     FALSE
## [12,]                                     FALSE
## [13,]                                      TRUE
## [14,]                                      TRUE
## [15,]                                      TRUE
##       FALSE, FALSE, FALSE, classif.xgboost
##  [1,]                                  TRUE
##  [2,]                                  TRUE
##  [3,]                                  TRUE
##  [4,]                                  TRUE
##  [5,]                                 FALSE
##  [6,]                                 FALSE
##  [7,]                                  TRUE
##  [8,]                                 FALSE
##  [9,]                                  TRUE
## [10,]                                  TRUE
## [11,]                                 FALSE
## [12,]                                 FALSE
## [13,]                                  TRUE
## [14,]                                  TRUE
## [15,]                                  TRUE
##       FALSE, FALSE, TRUE, classif.ksvm
##  [1,]                               TRUE
##  [2,]                               TRUE
##  [3,]                               TRUE
##  [4,]                              FALSE
##  [5,]                              FALSE
```

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

```
## [12,]                                    FALSE
## [13,]                                     TRUE
## [14,]                                     TRUE
## [15,]                                     TRUE
##       FALSE, TRUE, FALSE, classif.xgboost
##  [1,]                                   TRUE
##  [2,]                                   TRUE
##  [3,]                                   TRUE
##  [4,]                                   TRUE
##  [5,]                                  FALSE
##  [6,]                                  FALSE
##  [7,]                                  FALSE
##  [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
##                                   4.274390
## ADASYN, FALSE, FALSE, classif.randomForest
##                                   3.914634
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                   3.878049
##          FALSE, FALSE, FALSE, classif.ksvm
##                                  12.750000
##  FALSE, FALSE, FALSE, classif.randomForest
##                                   9.682927
##       FALSE, FALSE, FALSE, classif.xgboost
##                                   8.884146
##           FALSE, FALSE, TRUE, classif.ksvm
##                                  11.451220
##   FALSE, FALSE, TRUE, classif.randomForest
##                                  10.542683
##        FALSE, FALSE, TRUE, classif.xgboost
##                                  11.317073
##           FALSE, TRUE, FALSE, classif.ksvm
##                                  12.859756
##   FALSE, TRUE, FALSE, classif.randomForest
##                                   9.920732
##        FALSE, TRUE, FALSE, classif.xgboost
##                                   9.225610
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   3.981707
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                   3.512195
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                   3.804878
```

Plotando grafico de Critical Diference

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