R Notebook

Parametros:

Measure = F1 measure
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                     :10260   SMOTE :2052   TRUE :2052
##  G-mean                         :    0                 NA's :0
##  Matthews correlation coefficient:    0
##
##
##  tuning_measure   holdout_measure  holdout_measure_residual
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.2739   1st Qu.:0.0000   1st Qu.:0.04287
##  Median :0.8197   Median :0.4500   Median :0.28466
##  Mean   :0.6468   Mean   :0.4554   Mean   :0.36600
##  3rd Qu.:0.9944   3rd Qu.:0.8075   3rd Qu.:0.68235
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##  NA's   :216      NA's   :216      NA's   :216
##  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   :216     (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                         0.10190589
## 2                         0.10190589
## 3                         0.21605452
## 4                         0.23727721
## 5                         0.08378369
## 6                         0.08378369
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                 0.07466998
## 2                                 0.07466998
## 3                                 0.20140359
## 4                                 0.23887424
## 5                                         NA
## 6                                         NA
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.03799017                          0.03448112
## 2                            0.03799017                          0.03448112
## 3                            0.09169446                          0.14887811
## 4                            0.15183135                          0.19111774
## 5                            0.35860150                          0.07719857
## 6                            0.35860150                          0.07719857
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                              0.0005124263
## 2                              0.0005124263
```

```
## 3                              0.0000000000
## 4                              0.0049632477
## 5                              0.3814761838
## 6                              0.3814761838
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                          0.001536492                         0.7505638
## 2                          0.001536492                         0.7505638
## 3                          0.010556046                         0.7212287
## 4                          0.018575041                         0.6914606
## 5                          0.338343423                         0.5682765
## 6                          0.338343423                         0.5682765
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                               0.7453236
## 2                               0.7453236
## 3                               0.7198507
## 4                               0.7295545
## 5                               0.8501727
## 6                               0.8501727
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                          0.7348548                         0.03255014
## 2                          0.7348548                         0.03255014
## 3                          0.7287353                         0.12147113
## 4                          0.7342854                         0.18547876
## 5                          0.8479729                         0.08452969
## 6                          0.8479729                         0.08452969
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                            0.0005124263
## 2                            0.0005124263
## 3                            0.0015936255
## 4                            0.0049632477
## 5                            0.3628587010
## 6                                      NA
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                         0.0005124263                         0.09725191
## 2                         0.0005124263                         0.09725191
## 3                         0.0015923567                         0.19840898
## 4                         0.0120329137                         0.25263278
## 5                         0.3426039783                         0.09095731
## 6                         0.3426039783                         0.09095731
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                               0.07952866
## 2                               0.07952866
## 3                               0.20997048
## 4                               0.23593746
## 5                                       NA
## 6                               0.24268382
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                          0.03895394
## 2                          0.03895394
## 3                          0.10398276
## 4                          0.14186432
## 5                          0.33593996
## 6                          0.33593996
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.000000
##  1st Qu.:0.002186
##  Median :0.083784
##  Mean   :0.200389
##  3rd Qu.:0.314711
##  Max.   :0.989520
##  NA's   :7
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.1032
##  Median :0.3578
##  Mean   :0.3941
##  3rd Qu.:0.6036
##  Max.   :0.9922
##  NA's   :27
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000                        Min.   :0.0000
##  1st Qu.:0.1261                        1st Qu.:0.0000
##  Median :0.4000                        Median :0.1033
##  Mean   :0.4501                        Mean   :0.2038
##  3rd Qu.:0.7600                        3rd Qu.:0.2686
##  Max.   :0.9975                        Max.   :0.9949
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.00000
##  1st Qu.:0.04393
##  Median :0.26320
##  Mean   :0.33697
##  3rd Qu.:0.52490
##  Max.   :1.00000
##  NA's   :5
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.00000                      Min.   :0.0001813
##  1st Qu.:0.06354                      1st Qu.:0.1770854
##  Median :0.28825                      Median :0.4761142
##  Mean   :0.36424                      Mean   :0.4575081
##  3rd Qu.:0.64321                      3rd Qu.:0.7176554
##  Max.   :0.99746                      Max.   :0.9895300
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.003676
##  1st Qu.:0.297097
##  Median :0.671195
##  Mean   :0.578258
##  3rd Qu.:0.872340
##  Max.   :0.983805
##  NA's   :6
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.001115                    Min.   :0.0000
##  1st Qu.:0.271380                    1st Qu.:0.0000
##  Median :0.688069                    Median :0.0955
```

```
##  Mean   :0.572520                    Mean   :0.1956
##  3rd Qu.:0.851709                    3rd Qu.:0.2581
##  Max.   :0.982204                    Max.   :0.9949
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.00000
##  1st Qu.:0.04393
##  Median :0.24886
##  Mean   :0.33315
##  3rd Qu.:0.56950
##  Max.   :1.00000
##  NA's   :7
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.00000                     Min.   :0.000000
##  1st Qu.:0.07262                     1st Qu.:0.006659
##  Median :0.30130                     Median :0.093235
##  Mean   :0.36291                     Mean   :0.194141
##  3rd Qu.:0.63019                     3rd Qu.:0.309480
##  Max.   :1.00000                     Max.   :0.980870
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.1112
##  Median :0.3235
##  Mean   :0.3967
##  3rd Qu.:0.6502
##  Max.   :0.9975
##  NA's   :20
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.0000
##  1st Qu.:0.1497
##  Median :0.4211
##  Mean   :0.4544
##  3rd Qu.:0.7770
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

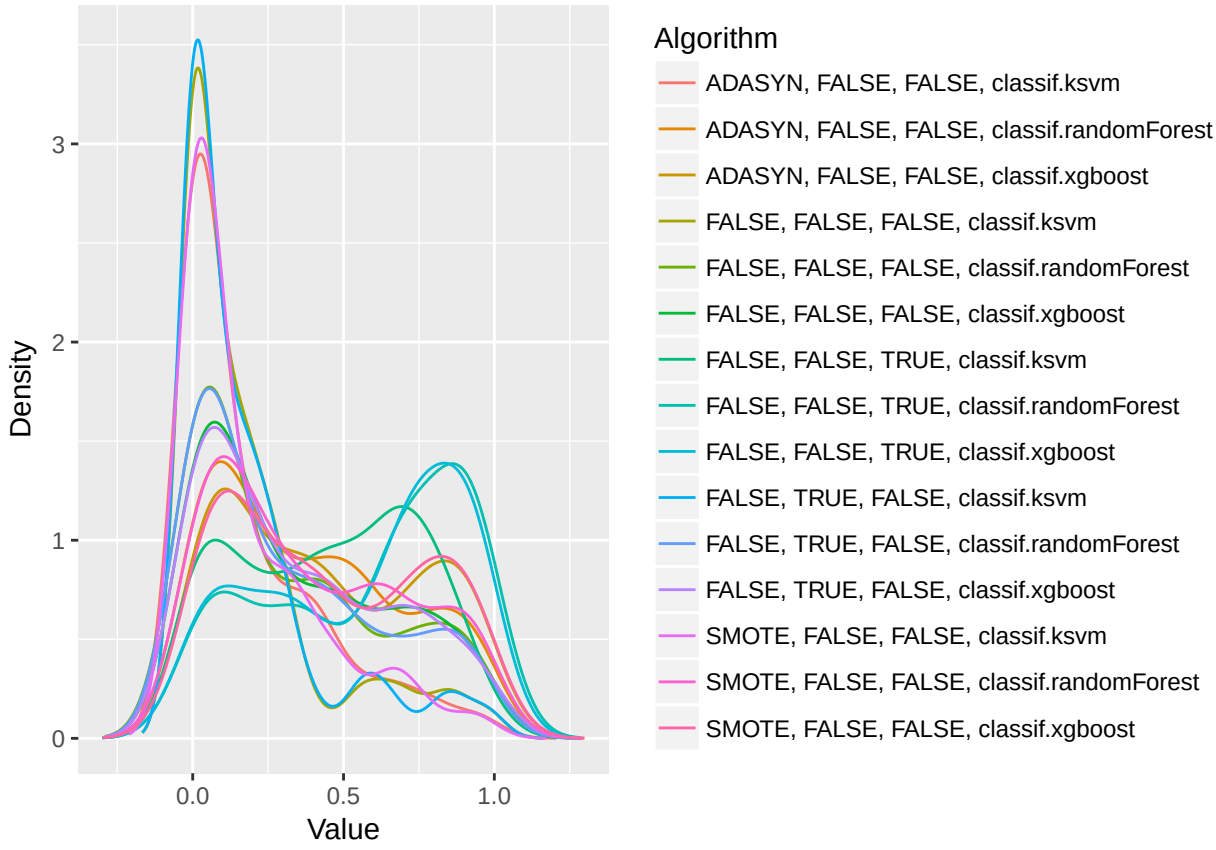
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.200388805339403"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.394142361433443"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.450143438246752"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.203750931957246"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.336968826836848"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.364243232689986"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.457508051594853"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.578257817245342"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.572519970741519"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.195606550543352"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.333148822135626"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.362911237893325"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.194141338237353"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.396717720545962"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.454423094002101"
```

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 = 963.19, 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,]                                 FALSE
##  [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,]                                      TRUE
##  [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,]                                 TRUE
##  [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,]                            FALSE
##  [4,]                             TRUE
##  [5,]                             TRUE
```

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

```
## [12,]                                    FALSE
## [13,]                                     TRUE
## [14,]                                     TRUE
## [15,]                                     TRUE
##       FALSE, TRUE, FALSE, classif.xgboost
##  [1,]                                TRUE
##  [2,]                               FALSE
##  [3,]                                TRUE
##  [4,]                                TRUE
##  [5,]                               FALSE
##  [6,]                               FALSE
##  [7,]                                TRUE
##  [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,]                                      TRUE
## [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,]                               FALSE
##  [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
##                                   11.269737
## ADASYN, FALSE, FALSE, classif.randomForest
##                                    7.697368
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                    5.293860
##          FALSE, FALSE, FALSE, classif.ksvm
##                                   11.078947
##  FALSE, FALSE, FALSE, classif.randomForest
##                                    8.688596
##       FALSE, FALSE, FALSE, classif.xgboost
##                                    8.144737
##           FALSE, FALSE, TRUE, classif.ksvm
##                                    6.559211
##   FALSE, FALSE, TRUE, classif.randomForest
##                                    4.357456
##        FALSE, FALSE, TRUE, classif.xgboost
##                                    4.537281
##           FALSE, TRUE, FALSE, classif.ksvm
##                                   11.254386
##   FALSE, TRUE, FALSE, classif.randomForest
##                                    9.024123
##        FALSE, TRUE, FALSE, classif.xgboost
##                                    8.258772
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   11.087719
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                    7.502193
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                    5.245614
```

Plotando grafico de Critical Diference

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