R Notebook

Parametros:

Measure = F1 measure
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure
Filter keys = imba.rate
Filter values = 0.03

```
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        :990   Mode :logical
##  classif.randomForest:990   FALSE:2376
##  classif.rusboost    :  0   TRUE :594
##  classif.xgboost     :990   NA's :0
##
##
##
##                              measure       sampling    underbagging
##  Accuracy                        :   0   ADASYN: 594   Mode :logical
##  Area under the curve            :   0   FALSE :1782   FALSE:2376
##  F1 measure                      :2970   SMOTE : 594   TRUE :594
##  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.:0.2788   1st Qu.:0.0481   1st Qu.:0.04815
##  Median :0.8296   Median :0.4840   Median :0.28571
##  Mean   :0.6542   Mean   :0.4646   Mean   :0.37464
##  3rd Qu.:0.9927   3rd Qu.:0.8000   3rd Qu.:0.70061
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##  NA's   :51       NA's   :51       NA's   :51
##  iteration_count          dataset       imba.rate
##  Min.   :1       abalone      :  45   Min.   :0.03
##  1st Qu.:1       adult        :  45   1st Qu.:0.03
##  Median :2       annealing    :  45   Median :0.03
##  Mean   :2       arrhythmia   :  45   Mean   :0.03
##  3rd Qu.:3       balance-scale:  45   3rd Qu.:0.03
##  Max.   :3       bank         :  45   Max.   :0.03
##  NA's   :51      (Other)      :2700
```

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] 66 15
```

```
# Renomeando a variavel
df = df_tec_wide_residual

head(df)
```

```
##   ADASYN, FALSE, FALSE, classif.ksvm
## 1                         0.03849057
## 2                         0.11510026
## 3                         0.16190476
## 4                         0.00000000
## 5                         0.66666667
## 6                         0.11827957
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                 0.03100775
## 2                                         NA
## 3                                 0.70000000
## 4                                 0.00000000
## 5                                 1.00000000
## 6                                 0.14533884
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                             0.0500000                          0.04164767
## 2                             0.3967302                          0.16774773
## 3                             0.5571429                          0.30158730
## 4                             0.4841270                          0.00000000
## 5                             1.0000000                          1.00000000
## 6                             0.2157503                          0.02222222
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                0.00000000
## 2                                0.42131460
```

```
## 3                                0.64761905
## 4                                0.66666667
## 5                                1.00000000
## 6                                0.02380952
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                           0.00000000                        0.08464646
## 2                           0.38793021                        0.17717435
## 3                           0.19047619                        0.30875817
## 4                           0.26666667                        0.07407407
## 5                           1.00000000                        0.68888889
## 6                           0.06349206                        0.07426864
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                              0.08826039
## 2                              0.20290920
## 3                              0.32777778
## 4                              0.32671958
## 5                              0.77777778
## 6                              0.19241883
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.1017349                        0.06740848
## 2                           0.2206182                        0.17763202
## 3                           0.2559801                        0.30158730
## 4                           0.4026936                        0.00000000
## 5                           0.4111111                        1.00000000
## 6                           0.1760133                        0.02222222
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                              0.00000000
## 2                              0.42297016
## 3                              0.75555556
## 4                              0.22222222
## 5                              1.00000000
## 6                              0.04103535
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                           0.0000000                         0.05409754
## 2                           0.4267146                         0.13195365
## 3                           0.1333333                         0.20634921
## 4                           0.2222222                         0.00000000
## 5                           1.0000000                         0.88888889
## 6                           0.2276547                         0.00000000
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                              0.05626016
## 2                                      NA
## 3                              0.48148148
## 4                              0.00000000
## 5                              1.00000000
## 6                              0.12257345
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                           0.04679803
## 2                           0.40713248
## 3                           0.46296296
## 4                           0.27777778
## 5                           1.00000000
## 6                           0.07525172
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.1719
##  Mean   :0.3076
##  3rd Qu.:0.5965
##  Max.   :1.0000
##  NA's   :2
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.2602
##  Median :0.6820
##  Mean   :0.5774
##  3rd Qu.:0.9077
##  Max.   :1.0000
##  NA's   :6
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000                        Min.   :0.0000
##  1st Qu.:0.3425                        1st Qu.:0.0000
##  Median :0.7613                        Median :0.2063
##  Mean   :0.6170                        Mean   :0.3299
##  3rd Qu.:0.9042                        3rd Qu.:0.5886
##  Max.   :1.0000                        Max.   :1.0000
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.1667
##  Median :0.6476
##  Mean   :0.5331
##  3rd Qu.:0.8667
##  Max.   :1.0000
##  NA's   :1
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.0000                       Min.   :0.04591
##  1st Qu.:0.1871                       1st Qu.:0.12026
##  Median :0.6379                       Median :0.28237
##  Mean   :0.5566                       Mean   :0.36006
##  3rd Qu.:0.8814                       3rd Qu.:0.54015
##  Max.   :1.0000                       Max.   :1.00000
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.04946
##  1st Qu.:0.18526
##  Median :0.33907
##  Mean   :0.41859
##  3rd Qu.:0.65635
##  Max.   :1.00000
##  NA's   :2
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.0441                      Min.   :0.0000
##  1st Qu.:0.1672                      1st Qu.:0.0000
##  Median :0.3753                      Median :0.1721
```

```
##  Mean   :0.3995                      Mean   :0.3096
##  3rd Qu.:0.5845                      3rd Qu.:0.5864
##  Max.   :1.0000                      Max.   :1.0000
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.1111
##  Median :0.5667
##  Mean   :0.5221
##  3rd Qu.:0.8889
##  Max.   :1.0000
##  NA's   :1
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000                      Min.   :0.0000
##  1st Qu.:0.2236                      1st Qu.:0.0000
##  Median :0.6186                      Median :0.1680
##  Mean   :0.5663                      Mean   :0.2939
##  3rd Qu.:0.8967                      3rd Qu.:0.5969
##  Max.   :1.0000                      Max.   :0.9833
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.1667
##  Median :0.6574
##  Mean   :0.5738
##  3rd Qu.:0.9444
##  Max.   :1.0000
##  NA's   :5
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.0000
##  1st Qu.:0.3346
##  Median :0.7016
##  Mean   :0.6179
##  3rd Qu.:0.9331
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

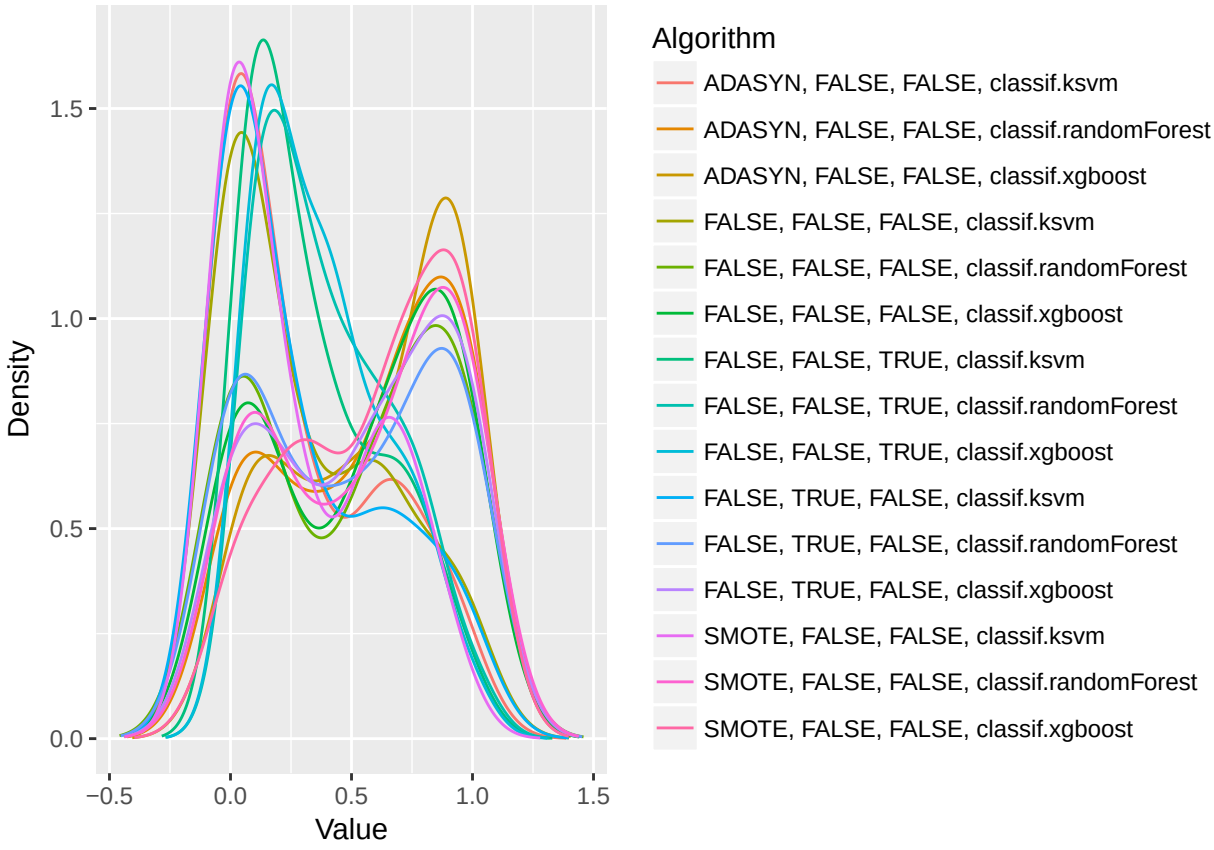
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.307580569550132"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.57743688206446"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.617031768286606"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.329863639121736"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.533077154028337"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.556624482997447"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.360063762038108"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.418589573591692"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.399453803913878"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.309593608142317"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.522113195721044"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.566305292026373"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.293912848477795"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.573784164338312"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.617900688046487"
```

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 = 252.84, 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,]                              FALSE
##  [8,]                              FALSE
##  [9,]                              FALSE
## [10,]                              FALSE
## [11,]                               TRUE
## [12,]                               TRUE
## [13,]                              FALSE
## [14,]                               TRUE
## [15,]                               TRUE
##       ADASYN, FALSE, FALSE, classif.randomForest
##  [1,]                                       TRUE
##  [2,]                                      FALSE
##  [3,]                                      FALSE
##  [4,]                                       TRUE
##  [5,]                                      FALSE
##  [6,]                                      FALSE
##  [7,]                                       TRUE
##  [8,]                                      FALSE
##  [9,]                                       TRUE
## [10,]                                       TRUE
## [11,]                                      FALSE
## [12,]                                      FALSE
## [13,]                                       TRUE
## [14,]                                      FALSE
## [15,]                                      FALSE
##       ADASYN, FALSE, FALSE, classif.xgboost
##  [1,]                                  TRUE
##  [2,]                                 FALSE
##  [3,]                                 FALSE
##  [4,]                                  TRUE
##  [5,]                                 FALSE
##  [6,]                                 FALSE
##  [7,]                                  TRUE
##  [8,]                                  TRUE
##  [9,]                                  TRUE
## [10,]                                  TRUE
## [11,]                                  TRUE
## [12,]                                 FALSE
## [13,]                                  TRUE
## [14,]                                 FALSE
## [15,]                                 FALSE
```

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

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

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

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

Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                   11.295455
## ADASYN, FALSE, FALSE, classif.randomForest
##                                    6.348485
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                    4.696970
##          FALSE, FALSE, FALSE, classif.ksvm
##                                   10.356061
##  FALSE, FALSE, FALSE, classif.randomForest
##                                    7.325758
##       FALSE, FALSE, FALSE, classif.xgboost
##                                    6.590909
##           FALSE, FALSE, TRUE, classif.ksvm
##                                    9.537879
##   FALSE, FALSE, TRUE, classif.randomForest
##                                    8.984848
##        FALSE, FALSE, TRUE, classif.xgboost
##                                    9.181818
##           FALSE, TRUE, FALSE, classif.ksvm
##                                   10.628788
##   FALSE, TRUE, FALSE, classif.randomForest
##                                    7.371212
##        FALSE, TRUE, FALSE, classif.xgboost
##                                    5.651515
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   11.272727
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                    6.409091
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                    4.348485
```

Plotando grafico de Critical Diference

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