R Notebook

Parametros:

Measure = Accuracy
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure_residual
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                       :2970   ADASYN: 594   Mode :logical
##  Area under the curve           :   0   FALSE :1782   FALSE:2376
##  F1 measure                     :   0   SMOTE : 594   TRUE :594
##  G-mean                         :   0                 NA's :0
##  Matthews correlation coefficient:   0
##
##
##  tuning_measure    holdout_measure   holdout_measure_residual
##  Min.   :0.09041   Min.   :0.02655   Min.   :0.0346
##  1st Qu.:0.96926   1st Qu.:0.96647   1st Qu.:0.3599
##  Median :0.98130   Median :0.97619   Median :0.6882
##  Mean   :0.95405   Mean   :0.94750   Mean   :0.6478
##  3rd Qu.:0.99560   3rd Qu.:0.99045   3rd Qu.:0.9438
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
##  NA's   :57        NA's   :57        NA's   :57
##  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   :57      (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                          0.3810826
## 2                          0.4831227
## 3                          0.7721519
## 4                          0.9807692
## 5                          0.2203548
## 6                          0.6854839
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.3803536
## 2                                         NA
## 3                                  0.8804501
## 4                                  0.9743590
## 5                                  0.2838469
## 6                                         NA
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.3429925                          0.3311463
## 2                            0.5878330                          0.4176651
## 3                            0.8157525                          0.7074543
## 4                            0.9839744                          0.9807692
## 5                            0.2931839                          0.2819795
## 6                            0.6970430                          0.6610215
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                 0.3154729
## 2                                        NA
```

```
## 3                                0.8284107
## 4                                0.9775641
## 5                                0.2987862
## 6                                0.6618280
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                           0.3156552                         0.6504465
## 2                           0.5584004                         0.5304448
## 3                           0.7862166                         0.7172996
## 4                           0.9807692                         0.8301282
## 5                           0.2577031                         0.2698413
## 6                           0.6610215                         0.5295699
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                                0.6655732
## 2                                0.8230483
## 3                                0.8888889
## 4                                0.9326923
## 5                                0.3856209
## 6                                0.7943548
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.6387826                         0.3214872
## 2                           0.8250725                         0.4185678
## 3                           0.8101266                         0.7074543
## 4                           0.9262821                         0.9807692
## 5                           0.4257703                         0.2819795
## 6                           0.7862903                         0.6610215
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                                0.3156552
## 2                                0.5670441
## 3                                0.8438819
## 4                                0.9839744
## 5                                0.2987862
## 6                                0.6596774
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                           0.3154729                         0.3728814
## 2                           0.5552820                         0.4875267
## 3                           0.7623066                         0.7566807
## 4                           0.9839744                         0.9807692
## 5                           0.2577031                         0.2885154
## 6                           0.6610215                         0.6634409
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                                0.3865500
## 2                                0.5870124
## 3                                0.8579466
## 4                                0.9743590
## 5                                0.2969188
## 6                                0.6938172
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                           0.3439038
## 2                           0.5828546
## 3                           0.8213783
## 4                           0.9647436
## 5                           0.2931839
## 6                           0.6940860
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.03709
##  1st Qu.:0.33198
##  Median :0.47672
##  Mean   :0.57868
##  3rd Qu.:0.91727
##  Max.   :0.99992
##  NA's   :2
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.03934
##  1st Qu.:0.37805
##  Median :0.63892
##  Mean   :0.65223
##  3rd Qu.:0.94647
##  Max.   :0.99987
##  NA's   :7
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.05205                       Min.   :0.03709
##  1st Qu.:0.40718                       1st Qu.:0.30583
##  Median :0.75889                       Median :0.47920
##  Mean   :0.68537                       Mean   :0.58807
##  3rd Qu.:0.94394                       3rd Qu.:0.94689
##  Max.   :0.99992                       Max.   :0.99992
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.07857
##  1st Qu.:0.33062
##  Median :0.64180
##  Mean   :0.62558
##  3rd Qu.:0.94887
##  Max.   :1.00000
##  NA's   :2
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.06025                      Min.   :0.1530
##  1st Qu.:0.36436                      1st Qu.:0.4562
##  Median :0.65692                      Median :0.6576
##  Mean   :0.64564                      Mean   :0.6538
##  3rd Qu.:0.96092                      3rd Qu.:0.8325
##  Max.   :0.99992                      Max.   :0.9983
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.2038
##  1st Qu.:0.6370
##  Median :0.8498
##  Mean   :0.7548
##  3rd Qu.:0.9266
##  Max.   :0.9998
##  NA's   :2
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.1649                      Min.   :0.03709
##  1st Qu.:0.6244                      1st Qu.:0.30565
##  Median :0.8354                      Median :0.47655
```

```
##  Mean   :0.7413                     Mean   :0.58540
##  3rd Qu.:0.9223                     3rd Qu.:0.94689
##  Max.   :0.9998                     Max.   :0.99992
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.09006
##  1st Qu.:0.33333
##  Median :0.65968
##  Mean   :0.63454
##  3rd Qu.:0.94840
##  Max.   :0.99987
##  NA's   :1
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.07042                    Min.   :0.03709
##  1st Qu.:0.35966                    1st Qu.:0.31756
##  Median :0.65597                    Median :0.45541
##  Mean   :0.64628                    Mean   :0.58942
##  3rd Qu.:0.96168                    3rd Qu.:0.92888
##  Max.   :0.99992                    Max.   :0.99992
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.04552
##  1st Qu.:0.38655
##  Median :0.65876
##  Mean   :0.65378
##  3rd Qu.:0.94148
##  Max.   :0.99988
##  NA's   :5
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.05609
##  1st Qu.:0.41188
##  Median :0.76136
##  Mean   :0.68353
##  3rd Qu.:0.94910
##  Max.   :1.00000
##
```

Verificando a média de cada coluna selecionada

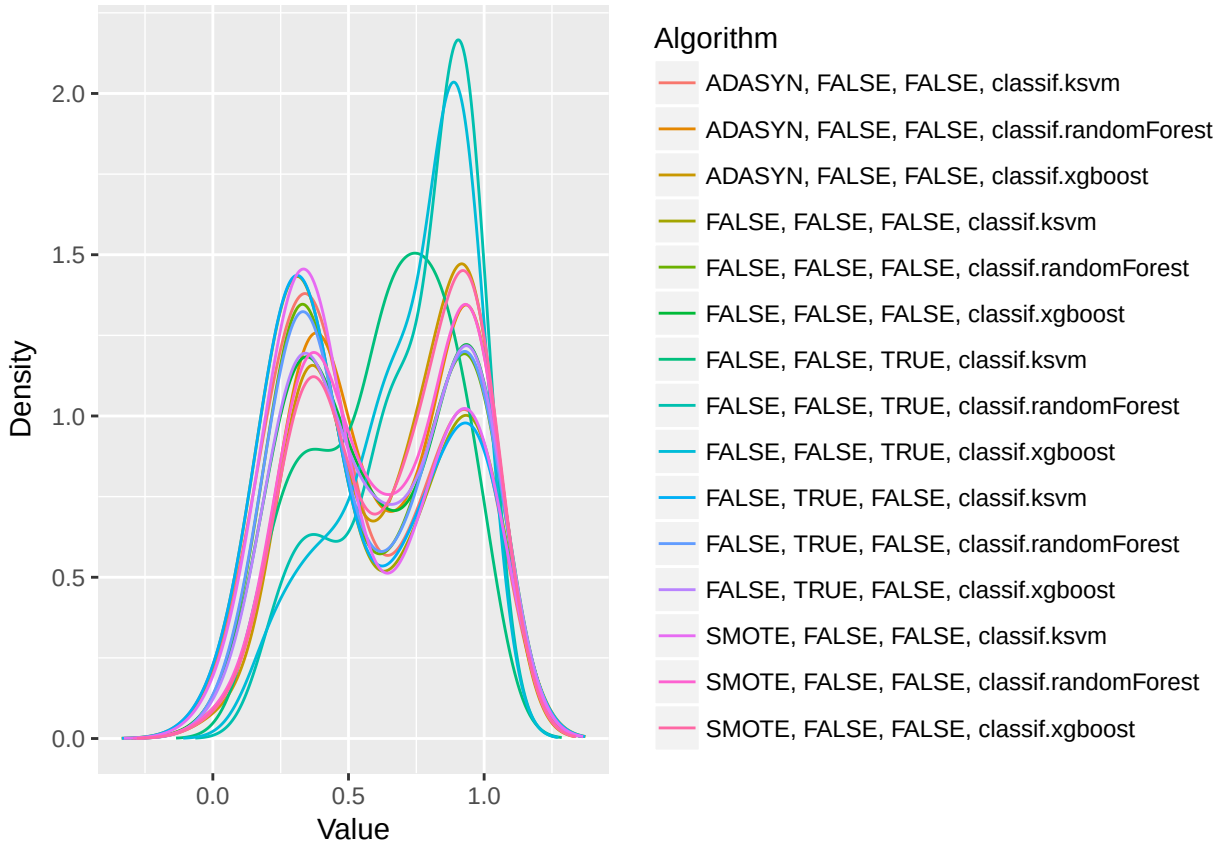
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.578684657297001"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.652230985561985"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.685367616034072"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.588071491527805"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.625576726604515"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.64563518661074"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.653808818891572"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.754799001458355"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.741313255801762"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.585401219001062"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.63454069319953"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.646283779594415"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.589417994142654"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.65377553206798"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.683531346088038"
```

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 = 122.45, 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,]                               TRUE
##  [4,]                              FALSE
##  [5,]                              FALSE
##  [6,]                              FALSE
##  [7,]                              FALSE
##  [8,]                               TRUE
##  [9,]                               TRUE
## [10,]                              FALSE
## [11,]                              FALSE
## [12,]                              FALSE
## [13,]                              FALSE
## [14,]                               TRUE
## [15,]                               TRUE
##       ADASYN, FALSE, FALSE, classif.randomForest
##  [1,]                                      FALSE
##  [2,]                                      FALSE
##  [3,]                                      FALSE
##  [4,]                                      FALSE
##  [5,]                                      FALSE
##  [6,]                                      FALSE
##  [7,]                                      FALSE
##  [8,]                                      FALSE
##  [9,]                                      FALSE
## [10,]                                      FALSE
## [11,]                                      FALSE
## [12,]                                      FALSE
## [13,]                                      FALSE
## [14,]                                      FALSE
## [15,]                                      FALSE
##       ADASYN, FALSE, FALSE, classif.xgboost
##  [1,]                                 TRUE
##  [2,]                                FALSE
##  [3,]                                FALSE
##  [4,]                                 TRUE
##  [5,]                                 TRUE
##  [6,]                                FALSE
##  [7,]                                 TRUE
##  [8,]                                FALSE
##  [9,]                                FALSE
## [10,]                                 TRUE
## [11,]                                 TRUE
## [12,]                                FALSE
## [13,]                                 TRUE
## [14,]                                FALSE
## [15,]                                FALSE
```

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

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

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

Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                   10.257576
## ADASYN, FALSE, FALSE, classif.randomForest
##                                    7.893939
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                    5.446970
##          FALSE, FALSE, FALSE, classif.ksvm
##                                   10.083333
##  FALSE, FALSE, FALSE, classif.randomForest
##                                    8.818182
##       FALSE, FALSE, FALSE, classif.xgboost
##                                    7.909091
##           FALSE, FALSE, TRUE, classif.ksvm
##                                    8.250000
##   FALSE, FALSE, TRUE, classif.randomForest
##                                    6.015152
##        FALSE, FALSE, TRUE, classif.xgboost
##                                    6.121212
##           FALSE, TRUE, FALSE, classif.ksvm
##                                   10.159091
##   FALSE, TRUE, FALSE, classif.randomForest
##                                    8.545455
##        FALSE, TRUE, FALSE, classif.xgboost
##                                    7.750000
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                    9.545455
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                    7.500000
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                    5.704545
```

Plotando grafico de Critical Diference

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