R Notebook

Parametros:

Measure = Area under the curve
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           :2970   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.   :0.3023   Min.   :0.0000   Min.   :0.00057
##  1st Qu.:0.9338   1st Qu.:0.8603   1st Qu.:0.69645
##  Median :0.9963   Median :0.9835   Median :0.89271
##  Mean   :0.9356   Mean   :0.8947   Mean   :0.82476
##  3rd Qu.:0.9999   3rd Qu.:0.9998   3rd Qu.:0.98444
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##  NA's   :66       NA's   :66       NA's   :66
##  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   :66      (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                          0.6127365
## 2                                 NA
## 3                          0.8458946
## 4                          0.7083333
## 5                          1.0000000
## 6                          0.7669113
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.6578104
## 2                                  0.8869102
## 3                                  0.9883578
## 4                                  0.9365079
## 5                                  1.0000000
## 6                                  0.8761618
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                             0.6776939                          0.6855853
## 2                             0.8905205                                 NA
## 3                             0.9564951                          0.9166667
## 4                             0.9563492                          0.5000000
## 5                             1.0000000                          1.0000000
## 6                             0.8728337                          0.7530486
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                 0.6779007
## 2                                 0.9005918
```

```
## 3                                0.9966299
## 4                                0.9861111
## 5                                1.0000000
## 6                                0.8677498
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                           0.7682208                         0.6796409
## 2                           0.9224770                         0.8562167
## 3                           0.9791667                         0.8602941
## 4                           0.9920635                         0.4722222
## 5                           1.0000000                         1.0000000
## 6                           0.9034504                         0.7020702
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                             0.7256108
## 2                             0.8816846
## 3                             0.9840686
## 4                             0.9761905
## 5                             1.0000000
## 6                             0.8829053
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.7254730                         0.6863779
## 2                           0.9045471                                NA
## 3                           0.9206495                         0.9166667
## 4                           0.9841270                         0.5000000
## 5                           1.0000000                         1.0000000
## 6                           0.8564640                         0.7530486
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                             0.6210758
## 2                             0.8952598
## 3                             0.9840686
## 4                             0.9523810
## 5                             1.0000000
## 6                             0.8467068
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                           0.7695303                         0.5259313
## 2                           0.9241239                                NA
## 3                           0.9889706                         0.9252451
## 4                           0.9265873                         0.2678571
## 5                           1.0000000                         1.0000000
## 6                           0.8837352                         0.7597484
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                             0.7177539
## 2                                    NA
## 3                             0.9895833
## 4                             0.9484127
## 5                             1.0000000
## 6                             0.8978249
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                           0.6558117
## 2                           0.8897241
## 3                           0.9485294
## 4                           0.9861111
## 5                           1.0000000
## 6                           0.8877184
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.3593
##  1st Qu.:0.7250
##  Median :0.9492
##  Mean   :0.8529
##  3rd Qu.:0.9920
##  Max.   :1.0000
##  NA's   :3
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.5843
##  1st Qu.:0.8780
##  Median :0.9871
##  Mean   :0.9206
##  3rd Qu.:0.9989
##  Max.   :1.0000
##  NA's   :5
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.4740                        Min.   :0.3333
##  1st Qu.:0.8831                        1st Qu.:0.7154
##  Median :0.9767                        Median :0.9466
##  Mean   :0.9165                        Mean   :0.8494
##  3rd Qu.:0.9997                        3rd Qu.:0.9974
##  Max.   :1.0000                        Max.   :1.0000
##                                        NA's   :1
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.2924
##  1st Qu.:0.8953
##  Median :0.9861
##  Mean   :0.9172
##  3rd Qu.:0.9998
##  Max.   :1.0000
##  NA's   :1
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.4439                       Min.   :0.4413
##  1st Qu.:0.8946                       1st Qu.:0.7924
##  Median :0.9840                       Median :0.8877
##  Mean   :0.9204                       Mean   :0.8525
##  3rd Qu.:0.9991                       3rd Qu.:0.9688
##  Max.   :1.0000                       Max.   :1.0000
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.5187
##  1st Qu.:0.8812
##  Median :0.9793
##  Mean   :0.9197
##  3rd Qu.:0.9984
##  Max.   :1.0000
##  NA's   :2
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.4707                      Min.   :0.3333
##  1st Qu.:0.8677                      1st Qu.:0.7039
##  Median :0.9758                      Median :0.9496
```

```
##  Mean   :0.9109                     Mean   :0.8390
##  3rd Qu.:0.9969                     3rd Qu.:0.9974
##  Max.   :1.0000                     Max.   :1.0000
##                                     NA's   :1
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.3369
##  1st Qu.:0.8703
##  Median :0.9848
##  Mean   :0.9054
##  3rd Qu.:0.9996
##  Max.   :1.0000
##  NA's   :4
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.3793                     Min.   :0.2679
##  1st Qu.:0.9208                     1st Qu.:0.7099
##  Median :0.9846                     Median :0.9262
##  Mean   :0.9208                     Mean   :0.8424
##  3rd Qu.:0.9997                     3rd Qu.:0.9917
##  Max.   :1.0000                     Max.   :1.0000
##                                     NA's   :2
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.6535
##  1st Qu.:0.8955
##  Median :0.9837
##  Mean   :0.9313
##  3rd Qu.:0.9996
##  Max.   :1.0000
##  NA's   :3
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.3896
##  1st Qu.:0.8918
##  Median :0.9854
##  Mean   :0.9223
##  3rd Qu.:0.9987
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

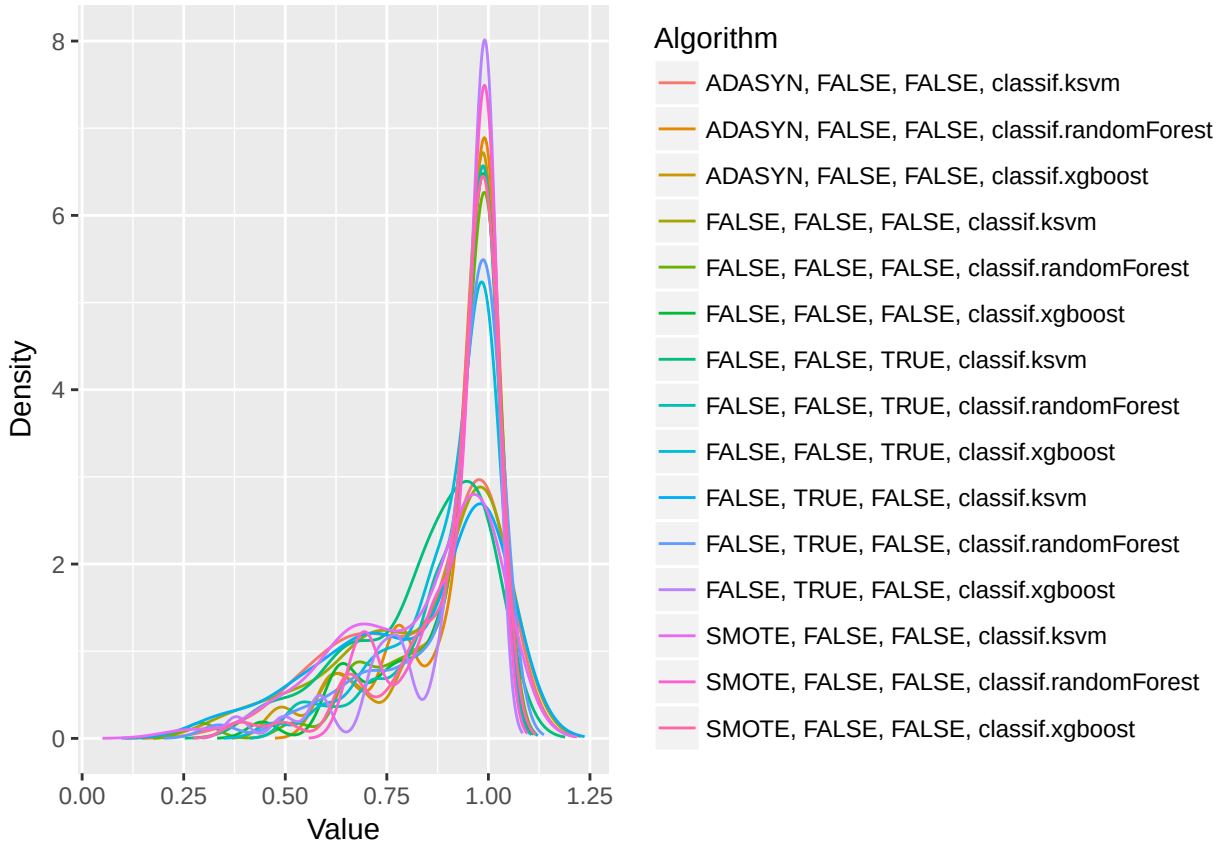
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.852903565015728"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.920628247479544"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.91651026974206"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.849372101874096"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.917245866711597"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.920388146954629"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.852491589774782"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.919732556969207"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.910925359680222"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.83901773037164"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.905428429199096"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.920769031641273"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.842395807810363"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.931259090456176"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.922262125325961"
```

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 = 180.05, 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,]                               TRUE
##  [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,]                                      FALSE
## [10,]                                       TRUE
## [11,]                                      FALSE
## [12,]                                      FALSE
## [13,]                                       TRUE
## [14,]                                      FALSE
## [15,]                                      FALSE
##       ADASYN, FALSE, FALSE, classif.xgboost
##  [1,]                                  TRUE
##  [2,]                                 FALSE
##  [3,]                                 FALSE
##  [4,]                                 FALSE
##  [5,]                                 FALSE
##  [6,]                                 FALSE
##  [7,]                                  TRUE
##  [8,]                                 FALSE
##  [9,]                                 FALSE
## [10,]                                  TRUE
## [11,]                                 FALSE
## [12,]                                 FALSE
## [13,]                                  TRUE
## [14,]                                 FALSE
## [15,]                                 FALSE
```

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

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

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

Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                   10.515152
## ADASYN, FALSE, FALSE, classif.randomForest
##                                    6.462121
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                    6.727273
##          FALSE, FALSE, FALSE, classif.ksvm
##                                    9.371212
##  FALSE, FALSE, FALSE, classif.randomForest
##                                    5.765152
##       FALSE, FALSE, FALSE, classif.xgboost
##                                    6.704545
##           FALSE, FALSE, TRUE, classif.ksvm
##                                   11.575758
##   FALSE, FALSE, TRUE, classif.randomForest
##                                    7.643939
##        FALSE, FALSE, TRUE, classif.xgboost
##                                    8.583333
##           FALSE, TRUE, FALSE, classif.ksvm
##                                    9.537879
##   FALSE, TRUE, FALSE, classif.randomForest
##                                    7.393939
##        FALSE, TRUE, FALSE, classif.xgboost
##                                    6.189394
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   11.143939
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                    6.295455
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                    6.090909
```

Plotando grafico de Critical Diference

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