R Notebook

Parametros:

Measure = Area under the curve
Columns = sampling, weight_space, underbagging, learner
Performance = tuning_measure
Filter keys = imba.rate
Filter values = 0.05

```
library("scmamp")
library(dplyr)
```

Tratamento dos dados

Carregando data set compilado

```
ds = read.csv("/home/rodrigo/Dropbox/UNICAMP/IC/estudo_cost_learning/SummaryResults/summary_compilation_

ds = filter(ds, learner != "classif.rusboost")
summary(ds)
```

```
##                  learner      weight_space
##  classif.ksvm        :17100   Mode :logical
##  classif.randomForest:17100   FALSE:41040
##  classif.rusboost    :    0   TRUE :10260
##  classif.xgboost     :17100   NA's :0
##
##
##
##                             measure        sampling     underbagging
##  Accuracy                       :10260   ADASYN:10260   Mode :logical
##  Area under the curve           :10260   FALSE :30780   FALSE:41040
##  F1 measure                     :10260   SMOTE :10260   TRUE :10260
##  G-mean                         :10260                  NA's :0
##  Matthews correlation coefficient:10260
##
##
##  tuning_measure    holdout_measure   holdout_measure_residual
##  Min.   :-0.1277   Min.   :-0.2120   Min.   :-0.4658
##  1st Qu.: 0.6911   1st Qu.: 0.4001   1st Qu.: 0.1994
##  Median : 0.9700   Median : 0.8571   Median : 0.5581
##  Mean   : 0.7903   Mean   : 0.6718   Mean   : 0.5298
##  3rd Qu.: 0.9975   3rd Qu.: 0.9900   3rd Qu.: 0.8755
##  Max.   : 1.0000   Max.   : 1.0000   Max.   : 1.0000
##  NA's   :1077      NA's   :1077      NA's   :1077
##  iteration_count                      dataset        imba.rate
##  Min.   :1       abalone                 :  900   Min.   :0.0010
##  1st Qu.:1       adult                   :  900   1st Qu.:0.0100
##  Median :2       bank                    :  900   Median :0.0300
##  Mean   :2       car                     :  900   Mean   :0.0286
```

```
##  3rd Qu.:3       cardiotocography-10clases:  900   3rd Qu.:0.0500
##  Max.   :3       cardiotocography-3clases :  900   Max.   :0.0500
##  NA's   :1077    (Other)                  :45900
```

Filtrando pela metrica

```
ds = filter(ds, measure == params$measure)
```

Filtrando o data set

```
if(params$filter_keys != 'NULL' && !is.null(params$filter_keys)){
  dots = paste0(params$filter_keys," == '",params$filter_values,"'")
  ds = filter_(ds, .dots = dots)
}

summary(ds)
```

```
##                  learner     weight_space
##  classif.ksvm        :1230   Mode :logical
##  classif.randomForest:1230   FALSE:2952
##  classif.rusboost    :   0   TRUE :738
##  classif.xgboost     :1230   NA's :0
##
##
##
##                              measure       sampling    underbagging
##  Accuracy                        :   0   ADASYN: 738   Mode :logical
##  Area under the curve            :3690   FALSE :2214   FALSE:2952
##  F1 measure                      :   0   SMOTE : 738   TRUE :738
##  G-mean                          :   0                 NA's :0
##  Matthews correlation coefficient:   0
##
##
##  tuning_measure   holdout_measure  holdout_measure_residual
##  Min.   :0.3977   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.9145   1st Qu.:0.8175   1st Qu.:0.6976
##  Median :0.9932   Median :0.9755   Median :0.8806
##  Mean   :0.9282   Mean   :0.8846   Mean   :0.8211
##  3rd Qu.:0.9997   3rd Qu.:0.9992   3rd Qu.:0.9784
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  NA's   :84       NA's   :84       NA's   :84
##  iteration_count          dataset       imba.rate
##  Min.   :1       abalone      :  45   Min.   :0.05
##  1st Qu.:1       adult        :  45   1st Qu.:0.05
##  Median :2       annealing    :  45   Median :0.05
##  Mean   :2       arrhythmia   :  45   Mean   :0.05
##  3rd Qu.:3       balance-scale:  45   3rd Qu.:0.05
##  Max.   :3       bank         :  45   Max.   :0.05
##  NA's   :84      (Other)      :3420
```

Computando as médias das iteracoes

```
ds = group_by(ds, learner , weight_space , measure , sampling , underbagging , dataset , imba.rate)
ds = summarise(ds, tuning_measure = mean(tuning_measure), holdout_measure = mean(holdout_measure),
               holdout_measure_residual = mean(holdout_measure_residual))

ds = as.data.frame(ds)
```

Criando dataframe

```
# Dividindo o ds em n, um para cada técnica
splited_df = ds %>% group_by_at(.vars = params$columns) %>% do(vals = as.data.frame(.)) %>% select(vals)

# Juntando cada uma das partes horizontalmente em um data set
df_tec_wide = do.call("cbind", splited_df)

# Renomeando duplicacao de nomes
colnames(df_tec_wide) = make.unique(colnames(df_tec_wide))

# Selecionando apenas as medidas da performance escolhida
df_tec_wide_residual = select(df_tec_wide, matches(paste("^", params$performance, "$|", params$performa

# Renomeando colunas
new_names = NULL
for(i in (1:length(splited_df))){
  id = toString(sapply(splited_df[[i]][1, params$columns], as.character))
  new_names = c(new_names, id)
}
colnames(df_tec_wide_residual) = new_names

# Verificando a dimensao do df
dim(df_tec_wide_residual)
```

```
## [1] 82 15
```

```
# Renomeando a variavel
df = df_tec_wide_residual

head(df)
```

```
##   ADASYN, FALSE, FALSE, classif.ksvm
## 1                          0.9815223
## 2                                 NA
## 3                          0.9920167
## 4                          0.9827796
## 5                          1.0000000
## 6                          0.9988989
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.9849170
## 2                                  0.9955712
## 3                                  0.9993568
## 4                                  0.9999242
## 5                                  1.0000000
## 6                                  0.9979093
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                             0.9914382                          0.6418531
## 2                             0.9956298                                 NA
## 3                             0.9991018                          0.7864112
## 4                             0.9992371                          0.5000000
## 5                             1.0000000                          1.0000000
## 6                             0.9970050                          0.7924629
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                 0.6964474
## 2                                 0.8981041
```

```
## 3                                0.9905911
## 4                                0.9612795
## 5                                1.0000000
## 6                                0.8801934
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                           0.7168914                         0.6658918
## 2                           0.9165578                         0.8291724
## 3                           0.9814664                         0.7307403
## 4                           0.9870230                         0.5811588
## 5                           1.0000000                         1.0000000
## 6                           0.8842474                         0.7546013
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                               0.7309119
## 2                                      NA
## 3                               0.9795330
## 4                               0.9840067
## 5                               1.0000000
## 6                               0.8729236
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.7259539                                NA
## 2                           0.9061667                                NA
## 3                           0.9457564                         0.7864112
## 4                           0.9791667                         0.5000000
## 5                           1.0000000                         1.0000000
## 6                           0.8744592                         0.7924629
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                               0.6964474
## 2                               0.8997472
## 3                               0.9879044
## 4                               0.9656285
## 5                               1.0000000
## 6                               0.8801934
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                           0.7164017                         0.9803999
## 2                           0.9175758                                NA
## 3                           0.9851009                         0.9948641
## 4                           0.9779742                         0.9720947
## 5                           1.0000000                         1.0000000
## 6                           0.8861514                         0.9989664
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                               0.9866418
## 2                                      NA
## 3                               0.9989449
## 4                               0.9999490
## 5                               1.0000000
## 6                               0.9976390
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                           0.9913887
## 2                           0.9959374
## 3                           0.9992231
## 4                           0.9988394
## 5                           1.0000000
## 6                           0.9968077
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.7759
##  1st Qu.:0.9950
##  Median :0.9997
##  Mean   :0.9890
##  3rd Qu.:1.0000
##  Max.   :1.0000
##  NA's   :5
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.7745
##  1st Qu.:0.9964
##  Median :0.9997
##  Mean   :0.9901
##  3rd Qu.:1.0000
##  Max.   :1.0000
##  NA's   :3
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.7724                        Min.   :0.4719
##  1st Qu.:0.9942                        1st Qu.:0.6785
##  Median :0.9992                        Median :0.9017
##  Mean   :0.9891                        Mean   :0.8440
##  3rd Qu.:0.9999                        3rd Qu.:0.9909
##  Max.   :1.0000                        Max.   :1.0000
##                                        NA's   :4
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.4564
##  1st Qu.:0.8761
##  Median :0.9724
##  Mean   :0.9092
##  3rd Qu.:0.9961
##  Max.   :1.0000
##  NA's   :2
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.4689                       Min.   :0.5226
##  1st Qu.:0.8617                       1st Qu.:0.7595
##  Median :0.9693                       Median :0.8729
##  Mean   :0.9105                       Mean   :0.8462
##  3rd Qu.:0.9962                       3rd Qu.:0.9517
##  Max.   :1.0000                       Max.   :1.0000
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.4968
##  1st Qu.:0.8620
##  Median :0.9618
##  Mean   :0.9106
##  3rd Qu.:0.9926
##  Max.   :1.0000
##  NA's   :3
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.5068                      Min.   :0.4719
##  1st Qu.:0.8637                      1st Qu.:0.7062
##  Median :0.9543                      Median :0.8928
```

```
##  Mean   :0.9039                     Mean   :0.8439
##  3rd Qu.:0.9924                     3rd Qu.:0.9909
##  Max.   :1.0000                     Max.   :1.0000
##                                     NA's   :4
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.4950
##  1st Qu.:0.8826
##  Median :0.9746
##  Mean   :0.9089
##  3rd Qu.:0.9974
##  Max.   :1.0000
##
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.4655                     Min.   :0.7440
##  1st Qu.:0.8833                     1st Qu.:0.9974
##  Median :0.9702                     Median :0.9998
##  Mean   :0.9102                     Mean   :0.9893
##  3rd Qu.:0.9968                     3rd Qu.:1.0000
##  Max.   :1.0000                     Max.   :1.0000
##                                     NA's   :3
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.7577
##  1st Qu.:0.9972
##  Median :0.9997
##  Mean   :0.9903
##  3rd Qu.:1.0000
##  Max.   :1.0000
##  NA's   :4
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.7667
##  1st Qu.:0.9936
##  Median :0.9990
##  Mean   :0.9895
##  3rd Qu.:0.9999
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

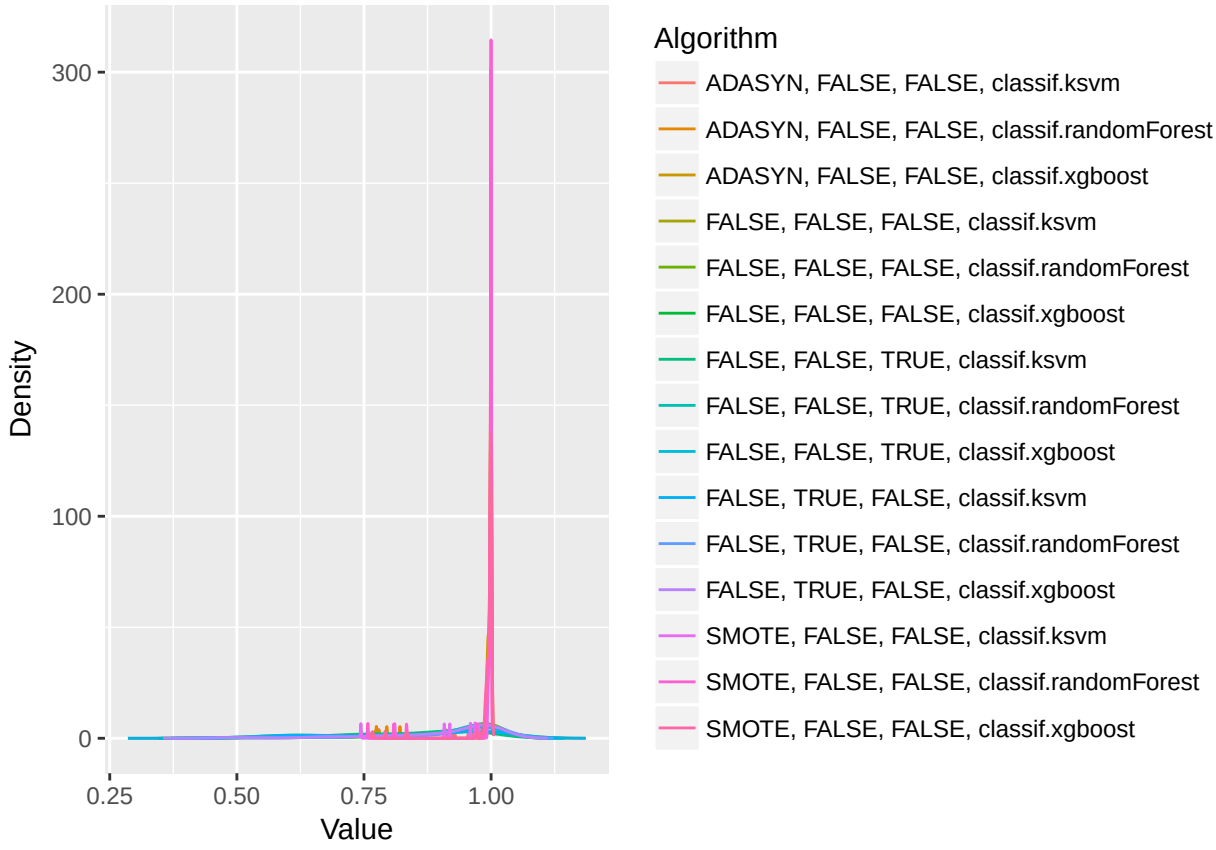
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.988978624104941"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.990068369450773"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.989116521783164"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.843966535459925"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.909210032275"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.91050638048228"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.846213459687723"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.91061825660726"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.903864409836207"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.843919085556466"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.90886997417648"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.910215097579915"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.989302699006571"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.990282947095929"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.989506447237661"
```

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 = 697.72, df = 14, p-value < 2.2e-16
```

Testando as diferencas par a par

```
test <- nemenyiTest (df, alpha=0.05)
abs(test$diff.matrix) > test$statistic
```

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

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

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

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

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

Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                   4.121951
## ADASYN, FALSE, FALSE, classif.randomForest
##                                   3.682927
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                   4.646341
##          FALSE, FALSE, FALSE, classif.ksvm
##                                  11.987805
##  FALSE, FALSE, FALSE, classif.randomForest
##                                   9.182927
##       FALSE, FALSE, FALSE, classif.xgboost
##                                   9.256098
##           FALSE, FALSE, TRUE, classif.ksvm
##                                  12.853659
##   FALSE, FALSE, TRUE, classif.randomForest
##                                  10.506098
##        FALSE, FALSE, TRUE, classif.xgboost
##                                  10.786585
##           FALSE, TRUE, FALSE, classif.ksvm
##                                  12.158537
##   FALSE, TRUE, FALSE, classif.randomForest
##                                   9.146341
##        FALSE, TRUE, FALSE, classif.xgboost
##                                   9.536585
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   3.585366
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                   3.750000
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                   4.798780
```

Plotando grafico de Critical Diference

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