R Notebook

Parametros:

Measure = F1 measure
Columns = sampling, weight_space, underbagging, learner
Performance = tuning_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                          0.9605491
## 2                          0.9752661
## 3                          0.9830445
## 4                          0.9137324
## 5                          1.0000000
## 6                          0.9853090
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.9554395
## 2                                         NA
## 3                                  0.9911964
## 4                                  0.9970588
## 5                                  1.0000000
## 6                                  0.9863006
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.9671649                          0.06039800
## 2                            0.9841667                          0.13794827
## 3                            0.9887081                          0.16704345
## 4                            0.9877406                          0.00000000
## 5                            1.0000000                          0.95555556
## 6                            0.9842412                          0.03917554
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                 0.0000000
## 2                                 0.4296021
```

```
## 3                                0.5044974
## 4                                0.4208995
## 5                                1.0000000
## 6                                0.0773216
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                           0.0000000                       0.09971807
## 2                           0.4279426                       0.17195080
## 3                           0.3623074                       0.27798341
## 4                           0.4632275                       0.13845698
## 5                           1.0000000                       0.75687831
## 6                           0.1444711                       0.07882308
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                             0.1037997
## 2                             0.2021382
## 3                             0.2457998
## 4                             0.4089093
## 5                             0.5016656
## 6                             0.1951329
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.1002549                       0.04769641
## 2                           0.2128582                       0.15451572
## 3                           0.2227367                       0.16704345
## 4                           0.3981344                       0.00000000
## 5                           0.4551948                       0.95555556
## 6                           0.1944995                       0.03917554
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                            0.00000000
## 2                            0.43674218
## 3                            0.44920635
## 4                            0.58042328
## 5                            1.00000000
## 6                            0.09222036
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                          0.01777778                         0.9620655
## 2                          0.41147172                         0.9764353
## 3                          0.30708859                         0.9819709
## 4                          0.51816578                         0.8456461
## 5                          1.00000000                         1.0000000
## 6                          0.15768203                         0.9837050
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                             0.9530177
## 2                                    NA
## 3                             0.9930181
## 4                             0.9949722
## 5                             1.0000000
## 6                             0.9818218
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                           0.9660527
## 2                           0.9849976
## 3                           0.9881106
## 4                           0.9890111
## 5                           1.0000000
## 6                           0.9848897
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.8172
##  1st Qu.:0.9790
##  Median :0.9943
##  Mean   :0.9808
##  3rd Qu.:0.9977
##  Max.   :1.0000
##  NA's   :2
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.8141
##  1st Qu.:0.9840
##  Median :0.9961
##  Mean   :0.9871
##  3rd Qu.:0.9992
##  Max.   :1.0000
##  NA's   :6
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.8141                        Min.   :0.000000
##  1st Qu.:0.9767                        1st Qu.:0.005051
##  Median :0.9935                        Median :0.162666
##  Mean   :0.9832                        Mean   :0.288185
##  3rd Qu.:0.9982                        3rd Qu.:0.581997
##  Max.   :1.0000                        Max.   :0.959982
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.00000
##  1st Qu.:0.08806
##  Median :0.55752
##  Mean   :0.52133
##  3rd Qu.:0.87090
##  Max.   :1.00000
##  NA's   :1
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.0000                       Min.   :0.0564
##  1st Qu.:0.3453                       1st Qu.:0.1451
##  Median :0.6064                       Median :0.3266
##  Mean   :0.5771                       Mean   :0.3893
##  3rd Qu.:0.8640                       3rd Qu.:0.6055
##  Max.   :1.0000                       Max.   :0.9476
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.05959
##  1st Qu.:0.19274
##  Median :0.36648
##  Mean   :0.42234
##  3rd Qu.:0.64927
##  Max.   :1.00000
##  NA's   :2
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.06061                     Min.   :0.0000
##  1st Qu.:0.19857                     1st Qu.:0.0000
##  Median :0.35150                     Median :0.1564
```

```
##  Mean   :0.39152                     Mean   :0.2713
##  3rd Qu.:0.57327                     3rd Qu.:0.5362
##  Max.   :1.00000                     Max.   :0.9600
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.00000
##  1st Qu.:0.09222
##  Median :0.58042
##  Mean   :0.52139
##  3rd Qu.:0.86883
##  Max.   :1.00000
##  NA's   :1
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000                      Min.   :0.8167
##  1st Qu.:0.3322                      1st Qu.:0.9767
##  Median :0.6328                      Median :0.9922
##  Mean   :0.5782                      Mean   :0.9805
##  3rd Qu.:0.8686                      3rd Qu.:0.9976
##  Max.   :1.0000                      Max.   :1.0000
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.8159
##  1st Qu.:0.9826
##  Median :0.9959
##  Mean   :0.9877
##  3rd Qu.:0.9993
##  Max.   :1.0000
##  NA's   :5
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.8159
##  1st Qu.:0.9821
##  Median :0.9945
##  Mean   :0.9867
##  3rd Qu.:0.9982
##  Max.   :1.0000
##
```

Verificando a média de cada coluna selecionada

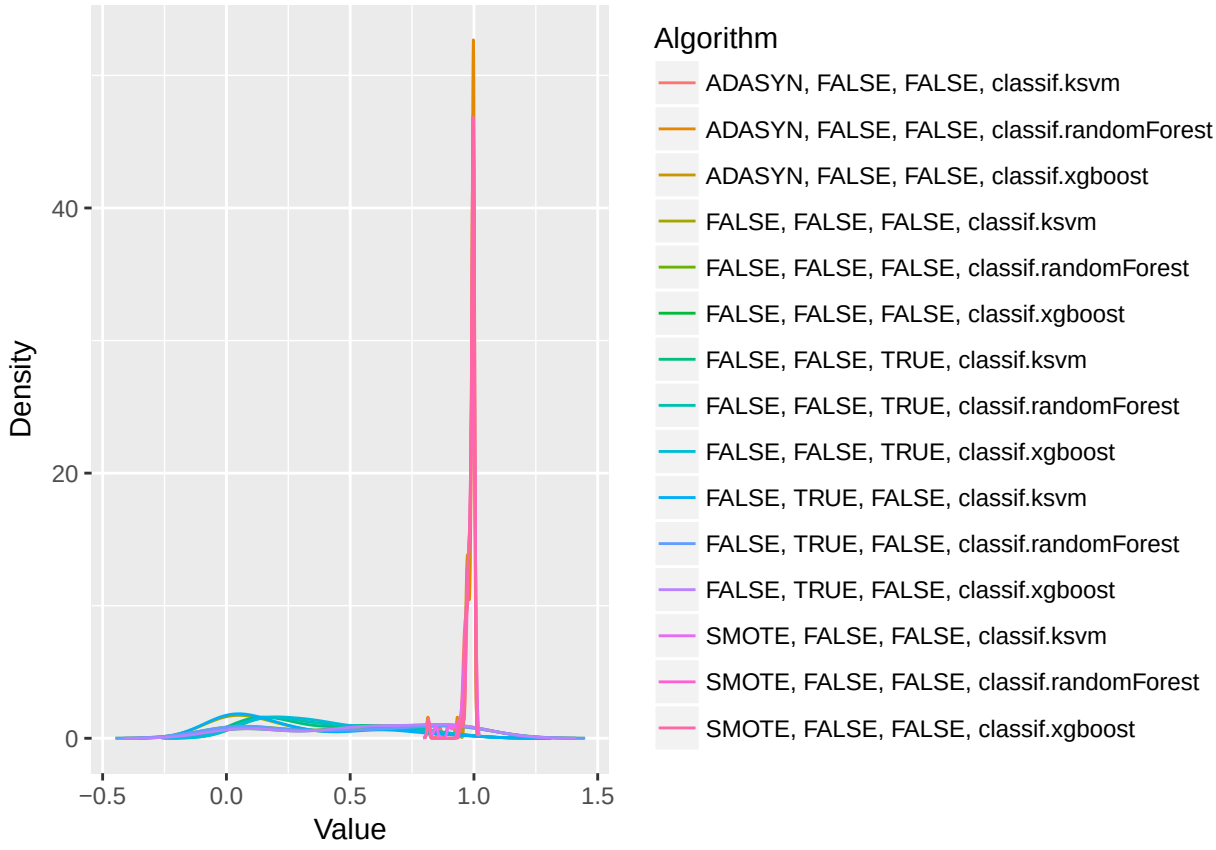
```
for(i in (1:dim(df)[2])){
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.980778211351388"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.987142008891316"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.983163588387305"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.288184857545814"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.521329243686241"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.577074112409825"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.38928315708718"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.422339892519806"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.391519438244507"
```

```
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.27130554460841"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.521388956122506"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.578248440440479"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.980548522469664"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.987674081114817"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.986721016814"
```

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 = 587.66, 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,]                               FALSE
##  [6,]                                TRUE
##  [7,]                               FALSE
##  [8,]                               FALSE
##  [9,]                               FALSE
## [10,]                               FALSE
## [11,]                               FALSE
## [12,]                                TRUE
## [13,]                                TRUE
## [14,]                                TRUE
## [15,]                                TRUE
##       FALSE, FALSE, FALSE, classif.randomForest
##  [1,]                                      TRUE
##  [2,]                                      TRUE
##  [3,]                                      TRUE
##  [4,]                                     FALSE
##  [5,]                                     FALSE
##  [6,]                                     FALSE
##  [7,]                                     FALSE
##  [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,]                                 FALSE
##  [8,]                                 FALSE
##  [9,]                                  TRUE
## [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,]                              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.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,]                                TRUE                               TRUE
##  [7,]                               FALSE                              FALSE
##  [8,]                               FALSE                              FALSE
##  [9,]                               FALSE                              FALSE
## [10,]                               FALSE                              FALSE
## [11,]                               FALSE                              FALSE
## [12,]                                TRUE                               TRUE
## [13,]                                TRUE                               TRUE
## [14,]                                TRUE                               TRUE
## [15,]                                TRUE                               TRUE
##       FALSE, TRUE, FALSE, 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, TRUE, FALSE, classif.xgboost
##  [1,]                                   TRUE
##  [2,]                                   TRUE
##  [3,]                                   TRUE
##  [4,]                                   TRUE
##  [5,]                                  FALSE
##  [6,]                                  FALSE
##  [7,]                                  FALSE
##  [8,]                                  FALSE
##  [9,]                                   TRUE
## [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
##                                   4.128788
## ADASYN, FALSE, FALSE, classif.randomForest
##                                   4.196970
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                   3.863636
##          FALSE, FALSE, FALSE, classif.ksvm
##                                  12.424242
##  FALSE, FALSE, FALSE, classif.randomForest
##                                   9.856061
##       FALSE, FALSE, FALSE, classif.xgboost
##                                   8.787879
##           FALSE, FALSE, TRUE, classif.ksvm
##                                  11.196970
##   FALSE, FALSE, TRUE, classif.randomForest
##                                  11.030303
##        FALSE, FALSE, TRUE, classif.xgboost
##                                  11.492424
##           FALSE, TRUE, FALSE, classif.ksvm
##                                  12.560606
##   FALSE, TRUE, FALSE, classif.randomForest
##                                  10.053030
##        FALSE, TRUE, FALSE, classif.xgboost
##                                   8.727273
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   3.598485
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                   4.159091
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                   3.924242
```

Plotando grafico de Critical Diference

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