R Notebook

Parametros:

Measure = Area under the curve
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure_residual
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

```r
# 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
```

```r
# Renomeando a variavel
df = df_tec_wide_residual

head(df)
```

```
##   ADASYN, FALSE, FALSE, classif.ksvm
## 1                          0.5948389
## 2                                 NA
## 3                          0.8072758
## 4                          0.5739972
## 5                          0.8808254
## 6                          0.7789270
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.6927146
## 2                                  0.8809462
## 3                                  0.9771092
## 4                                  0.9574689
## 5                                  0.6068844
## 6                                  0.8810023
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                             0.6819788                         0.6479328
## 2                             0.9062335                                NA
## 3                             0.9730628                         0.8307921
## 4                             0.9066390                         0.5000000
## 5                             0.5451291                         0.8701104
## 6                             0.8740440                         0.7816640
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                 0.6678776
## 2                                 0.8941163
## 3                                 0.9909297
## 4                                 0.9052559
## 5                                 0.6190888
## 6                                 0.8897089
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                            0.6806404                        0.6628553
## 2                            0.9180063                        0.8251103
## 3                            0.9843616                        0.7774846
## 4                            0.9567773                        0.6625173
## 5                            0.5881527                        0.7139014
## 6                            0.8668094                        0.7655071
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                                0.6966937
## 2                                       NA
## 3                                0.9798655
## 4                                0.9508990
## 5                                0.6253509
## 6                                0.8738974
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.6898095                               NA
## 2                           0.9053025                               NA
## 3                           0.9546681                        0.8307921
## 4                           0.9595436                        0.5000000
## 5                           0.6214361                        0.8701104
## 6                           0.8644446                        0.7816640
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                                0.6678776
## 2                                0.8932671
## 3                                0.9890531
## 4                                0.9747580
## 5                                0.6037495
## 6                                0.8897089
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                           0.6830128                         0.5862406
## 2                           0.9193893                                NA
## 3                           0.9848307                         0.8195129
## 4                           0.9591978                         0.5276625
## 5                           0.5881527                         0.8969212
## 6                           0.8644223                         0.7748179
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                                 0.6991028
## 2                                        NA
## 3                                 0.9889749
## 4                                 0.9132089
## 5                                 0.6350599
## 6                                 0.8895640
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                            0.6722286
## 2                            0.9034426
## 3                            0.9711862
## 4                            0.9343015
## 5                            0.6556320
## 6                            0.8809182
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.3506
##  1st Qu.:0.6630
##  Median :0.8385
##  Mean   :0.7949
##  3rd Qu.:0.9657
##  Max.   :0.9999
##  NA's   :5
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.3880
##  1st Qu.:0.7603
##  Median :0.9269
##  Mean   :0.8538
##  3rd Qu.:0.9855
##  Max.   :1.0000
##  NA's   :3
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.4087                        Min.   :0.4167
##  1st Qu.:0.6880                        1st Qu.:0.6353
##  Median :0.8681                        Median :0.8295
##  Mean   :0.8256                        Mean   :0.7861
##  3rd Qu.:0.9788                        3rd Qu.:0.9560
##  Max.   :0.9999                        Max.   :1.0000
##                                        NA's   :4
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.3885
##  1st Qu.:0.7914
##  Median :0.9147
##  Mean   :0.8524
##  3rd Qu.:0.9806
##  Max.   :1.0000
##  NA's   :2
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.4229                       Min.   :0.3869
##  1st Qu.:0.7144                       1st Qu.:0.6528
##  Median :0.8960                       Median :0.7696
##  Mean   :0.8399                       Mean   :0.7629
##  3rd Qu.:0.9753                       3rd Qu.:0.9024
##  Max.   :1.0000                       Max.   :0.9997
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.3757
##  1st Qu.:0.7087
##  Median :0.8839
##  Mean   :0.8247
##  3rd Qu.:0.9686
##  Max.   :1.0000
##  NA's   :3
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.3653                      Min.   :0.4167
##  1st Qu.:0.6900                      1st Qu.:0.6394
##  Median :0.8747                      Median :0.8295
##  Mean   :0.8170                      Mean   :0.7855
##  3rd Qu.:0.9724                      3rd Qu.:0.9560
##  Max.   :1.0000                      Max.   :1.0000
##                                      NA's   :4
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.4130
##  1st Qu.:0.7698
##  Median :0.9312
##  Mean   :0.8511
##  3rd Qu.:0.9829
##  Max.   :1.0000
##
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.4143                      Min.   :0.3945
##  1st Qu.:0.7155                      1st Qu.:0.6722
##  Median :0.8900                      Median :0.8195
##  Mean   :0.8377                      Mean   :0.7914
##  3rd Qu.:0.9759                      3rd Qu.:0.9520
##  Max.   :1.0000                      Max.   :1.0000
##                                      NA's   :3
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.4036
##  1st Qu.:0.7712
##  Median :0.9152
##  Mean   :0.8534
##  3rd Qu.:0.9864
##  Max.   :1.0000
##  NA's   :4
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.4457
##  1st Qu.:0.7202
##  Median :0.8740
##  Mean   :0.8375
##  3rd Qu.:0.9797
##  Max.   :0.9999
##
```

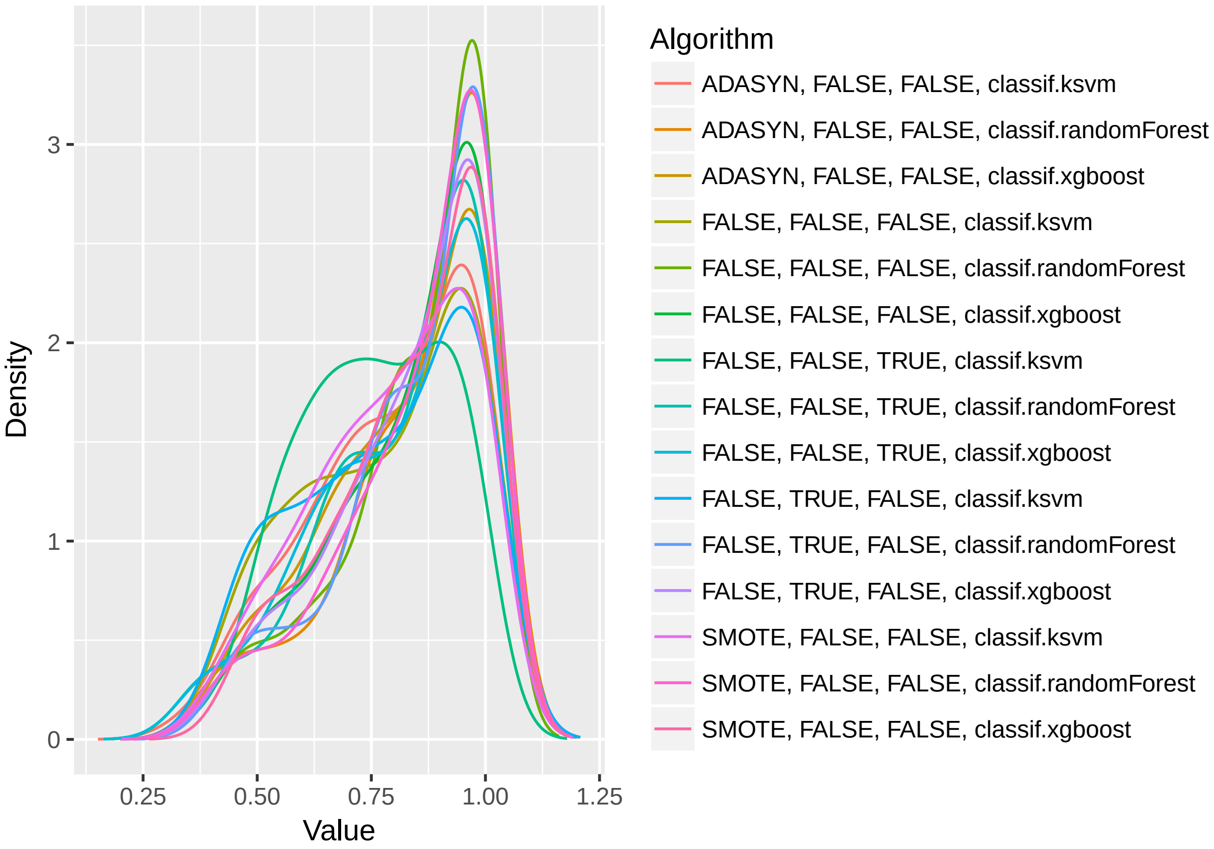
Verificando a média de cada coluna selecionada

```r
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 = 0.794908244987678"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.853807377275889"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.825635026660744"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.78605256570506"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.852398353253037"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.839899674001894"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.762887851728796"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.824740957713654"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.817013259180041"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.785507595146507"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.85112000282138"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.837668718003242"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.791392698052683"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.85342624969326"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.837479958094399"
```

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 = 206.47, 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,]                              FALSE
##  [4,]                              FALSE
##  [5,]                               TRUE
##  [6,]                               TRUE
##  [7,]                              FALSE
##  [8,]                              FALSE
##  [9,]                              FALSE
## [10,]                              FALSE
## [11,]                               TRUE
## [12,]                               TRUE
## [13,]                              FALSE
## [14,]                               TRUE
## [15,]                               TRUE
##       ADASYN, FALSE, FALSE, classif.randomForest
##  [1,]                                       TRUE
##  [2,]                                      FALSE
##  [3,]                                       TRUE
##  [4,]                                       TRUE
##  [5,]                                      FALSE
##  [6,]                                      FALSE
##  [7,]                                       TRUE
##  [8,]                                       TRUE
##  [9,]                                       TRUE
## [10,]                                       TRUE
## [11,]                                      FALSE
## [12,]                                      FALSE
## [13,]                                       TRUE
## [14,]                                      FALSE
## [15,]                                      FALSE
##       ADASYN, FALSE, FALSE, classif.xgboost
##  [1,]                                 FALSE
##  [2,]                                  TRUE
##  [3,]                                 FALSE
##  [4,]                                 FALSE
##  [5,]                                 FALSE
##  [6,]                                 FALSE
##  [7,]                                  TRUE
##  [8,]                                 FALSE
##  [9,]                                 FALSE
## [10,]                                 FALSE
## [11,]                                  TRUE
## [12,]                                 FALSE
## [13,]                                 FALSE
## [14,]                                  TRUE
## [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,]                              TRUE
## [12,]                              TRUE
## [13,]                             FALSE
## [14,]                              TRUE
## [15,]                              TRUE
##       FALSE, FALSE, FALSE, classif.randomForest
##  [1,]                                      TRUE
##  [2,]                                     FALSE
##  [3,]                                     FALSE
##  [4,]                                      TRUE
##  [5,]                                     FALSE
##  [6,]                                     FALSE
##  [7,]                                      TRUE
##  [8,]                                      TRUE
##  [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,]                                    FALSE
##  [2,]                                     TRUE
##  [3,]                                    FALSE
##  [4,]                                    FALSE
##  [5,]                                     TRUE
##  [6,]                                    FALSE
##  [7,]                                     TRUE
##  [8,]                                    FALSE
##  [9,]                                    FALSE
## [10,]                                    FALSE
## [11,]                                     TRUE
## [12,]                                    FALSE
## [13,]                                    FALSE
## [14,]                                     TRUE
## [15,]                                    FALSE
##       FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  [1,]                               FALSE                            FALSE
##  [2,]                                TRUE                             TRUE
##  [3,]                               FALSE                            FALSE
##  [4,]                               FALSE                            FALSE
##  [5,]                                TRUE                             TRUE
##  [6,]                               FALSE                             TRUE
##  [7,]                                TRUE                            FALSE
##  [8,]                               FALSE                            FALSE
##  [9,]                               FALSE                            FALSE
## [10,]                               FALSE                            FALSE
## [11,]                                TRUE                             TRUE
## [12,]                               FALSE                             TRUE
## [13,]                               FALSE                            FALSE
## [14,]                                TRUE                             TRUE
## [15,]                               FALSE                             TRUE
##       FALSE, TRUE, FALSE, classif.randomForest
##  [1,]                                     TRUE
##  [2,]                                    FALSE
##  [3,]                                     TRUE
##  [4,]                                     TRUE
##  [5,]                                    FALSE
##  [6,]                                    FALSE
##  [7,]                                     TRUE
##  [8,]                                     TRUE
##  [9,]                                     TRUE
## [10,]                                     TRUE
## [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,]                               FALSE
## [14,]                               FALSE
## [15,]                               FALSE
##       SMOTE, 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,]                              TRUE
## [12,]                             FALSE
## [13,]                             FALSE
## [14,]                              TRUE
## [15,]                              TRUE
##       SMOTE, FALSE, FALSE, classif.randomForest
##  [1,]                                      TRUE
##  [2,]                                     FALSE
##  [3,]                                      TRUE
##  [4,]                                      TRUE
##  [5,]                                     FALSE
##  [6,]                                     FALSE
##  [7,]                                      TRUE
##  [8,]                                      TRUE
##  [9,]                                      TRUE
## [10,]                                      TRUE
## [11,]                                     FALSE
## [12,]                                     FALSE
## [13,]                                      TRUE
## [14,]                                     FALSE
## [15,]                                     FALSE
##       SMOTE, FALSE, FALSE, classif.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
```

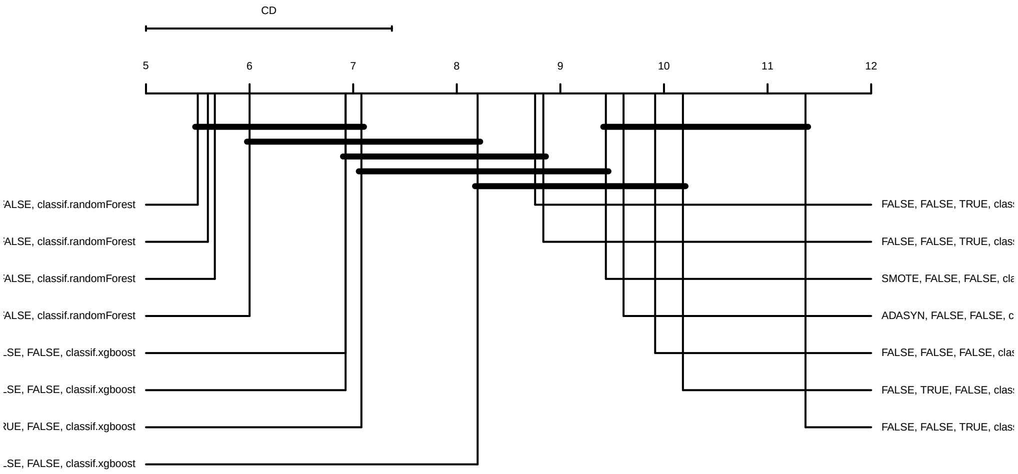
Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                   9.609756
## ADASYN, FALSE, FALSE, classif.randomForest
##                                   5.597561
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                   8.201220
##          FALSE, FALSE, FALSE, classif.ksvm
##                                   9.914634
##  FALSE, FALSE, FALSE, classif.randomForest
##                                   6.000000
##       FALSE, FALSE, FALSE, classif.xgboost
##                                   6.926829
##           FALSE, FALSE, TRUE, classif.ksvm
##                                  11.365854
##   FALSE, FALSE, TRUE, classif.randomForest
##                                   8.835366
##        FALSE, FALSE, TRUE, classif.xgboost
##                                   8.756098
##           FALSE, TRUE, FALSE, classif.ksvm
##                                  10.182927
##   FALSE, TRUE, FALSE, classif.randomForest
##                                   5.664634
##        FALSE, TRUE, FALSE, classif.xgboost
##                                   7.079268
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                   9.439024
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                   5.500000
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                   6.926829
```

Plotando grafico de Critical Diference

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


CD
ALSE, classif.randomForest
ALSE, classif.randomForest
ALSE, classif.randomForest
ALSE, classif.randomForest
LSE, FALSE, classif.xgboost
LSE, FALSE, classif.xgboost
RUE, FALSE, classif.xgboost
LSE, FALSE, classif.xgboost
FALSE, FALSE, TRUE, clas
FALSE, FALSE, TRUE, clas
SMOTE, FALSE, FALSE, cla
ADASYN, FALSE, FALSE, c
FALSE, FALSE, FALSE, clas
FALSE, TRUE, FALSE, clas
FALSE, FALSE, TRUE, clas