R Notebook

Parametros:

Measure = Matthews correlation coefficient
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure
Filter keys = imba.rate
Filter values = 0.05

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

Tratamento dos dados

Carregando data set compilado

```r
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            :   0   FALSE :2214   FALSE:2952
##  F1 measure                      :   0   SMOTE : 738   TRUE :738
##  G-mean                          :   0                 NA's :0
##  Matthews correlation coefficient:3690
##
##
##  tuning_measure    holdout_measure    holdout_measure_residual
##  Min.   :-0.1277   Min.   :-0.21201   Min.   :-0.45710
##  1st Qu.: 0.3764   1st Qu.: 0.06131   1st Qu.: 0.05637
##  Median : 0.8057   Median : 0.55190   Median : 0.23378
##  Mean   : 0.6629   Mean   : 0.49274   Mean   : 0.32193
##  3rd Qu.: 0.9728   3rd Qu.: 0.82456   3rd Qu.: 0.56442
##  Max.   : 1.0000   Max.   : 1.00000   Max.   : 1.00000
##  NA's   :54        NA's   :54         NA's   :54
##  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   :54      (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$performan

# 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                         0.05919056
## 2                         0.14927451
## 3                         0.44081601
## 4                         0.00000000
## 5                         1.00000000
## 6                         0.01727820
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                 0.06176052
## 2                                         NA
## 3                                 0.74696564
## 4                                 0.23333333
## 5                                 1.00000000
## 6                                 0.27693530
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.04406711                         0.1037074
## 2                            0.46104374                         0.2609339
## 3                            0.81291184                         0.3065053
## 4                            0.73604068                         0.0000000
## 5                            1.00000000                         1.0000000
## 6                            0.26844033                         0.2013082
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                 0.0000000
## 2                                 0.5019371
## 3                                 0.7917338
## 4                                 0.5929357
## 5                                 1.0000000
## 6                                 0.2003312
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                           0.05577801                        0.1222062
## 2                           0.53030447                        0.2801961
## 3                           0.60056656                        0.4408298
## 4                           0.70000000                        0.1702715
## 5                           1.00000000                        0.9530776
## 6                           0.28916269                        0.1850007
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                                0.1346849
## 2                                       NA
## 3                                0.5041621
## 4                                0.4530208
## 5                                0.9061553
## 6                                0.3247754
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.1454173                       0.07585303
## 2                           0.3594248                       0.29437987
## 3                           0.4687353                       0.32667105
## 4                           0.4907112                       0.00000000
## 5                           0.7747336                       1.00000000
## 6                           0.3258895                       0.14506241
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                                0.0000000
## 2                                       NA
## 3                                0.7084555
## 4                                0.6625713
## 5                                1.0000000
## 6                                0.2003312
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                         -0.01555568                        0.09021363
## 2                          0.50686751                        0.18853292
## 3                          0.52512959                        0.36382909
## 4                          0.72653061                        0.00000000
## 5                          1.00000000                        0.79290891
## 6                          0.22089840                        0.08378119
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                                 0.0721108
## 2                                 0.3708813
## 3                                 0.8314980
## 4                                 0.3931973
## 5                                 1.0000000
## 6                                 0.2201364
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                            0.1173197
## 2                            0.4709793
## 3                            0.8662987
## 4                            0.8000000
## 5                            1.0000000
## 6                            0.2666692
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :-0.06657
##  1st Qu.: 0.00000
##  Median : 0.23266
##  Mean   : 0.33557
##  3rd Qu.: 0.66667
##  Max.   : 1.00000
##  NA's   :1
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :-0.05198
##  1st Qu.: 0.31298
##  Median : 0.67374
##  Mean   : 0.58860
##  3rd Qu.: 0.86650
##  Max.   : 1.00000
##  NA's   :7
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :-0.06053                      Min.   :-0.04044
##  1st Qu.: 0.35119                      1st Qu.: 0.00000
##  Median : 0.71543                      Median : 0.19993
##  Mean   : 0.60412                      Mean   : 0.30638
##  3rd Qu.: 0.88058                      3rd Qu.: 0.57985
##  Max.   : 1.00000                      Max.   : 1.00000
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :-0.02449
##  1st Qu.: 0.19079
##  Median : 0.64768
##  Mean   : 0.56150
##  3rd Qu.: 0.87827
##  Max.   : 1.00000
##
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :-0.03192                     Min.   :-0.0266
##  1st Qu.: 0.28134                     1st Qu.: 0.1743
##  Median : 0.68114                     Median : 0.4388
##  Mean   : 0.58262                     Mean   : 0.4353
##  3rd Qu.: 0.86716                     3rd Qu.: 0.6325
##  Max.   : 1.00000                     Max.   : 0.9796
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :-0.06927
##  1st Qu.: 0.28683
##  Median : 0.50510
##  Mean   : 0.50954
##  3rd Qu.: 0.75626
##  Max.   : 1.00000
##  NA's   :3
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :-0.01178                    Min.   :-0.03836
##  1st Qu.: 0.31915                    1st Qu.: 0.00000
##  Median : 0.47370                    Median : 0.18920
##  Mean   : 0.49921                    Mean   : 0.30136
##  3rd Qu.: 0.70084                    3rd Qu.: 0.55969
##  Max.   : 1.00000                    Max.   : 1.00000
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :-0.02178
##  1st Qu.: 0.24112
##  Median : 0.68293
##  Mean   : 0.57046
##  3rd Qu.: 0.87654
##  Max.   : 1.00000
##  NA's   :3
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :-0.01983                    Min.   :-0.04101
##  1st Qu.: 0.20960                    1st Qu.: 0.00000
##  Median : 0.67668                    Median : 0.25935
##  Mean   : 0.56559                    Mean   : 0.33800
##  3rd Qu.: 0.87877                    3rd Qu.: 0.65092
##  Max.   : 1.00000                    Max.   : 1.00000
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :-0.0748
##  1st Qu.: 0.2542
##  Median : 0.6959
##  Mean   : 0.5822
##  3rd Qu.: 0.9216
##  Max.   : 1.0000
##  NA's   :4
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :-0.03402
##  1st Qu.: 0.39289
##  Median : 0.72657
##  Mean   : 0.62482
##  3rd Qu.: 0.91781
##  Max.   : 1.00000
##
```

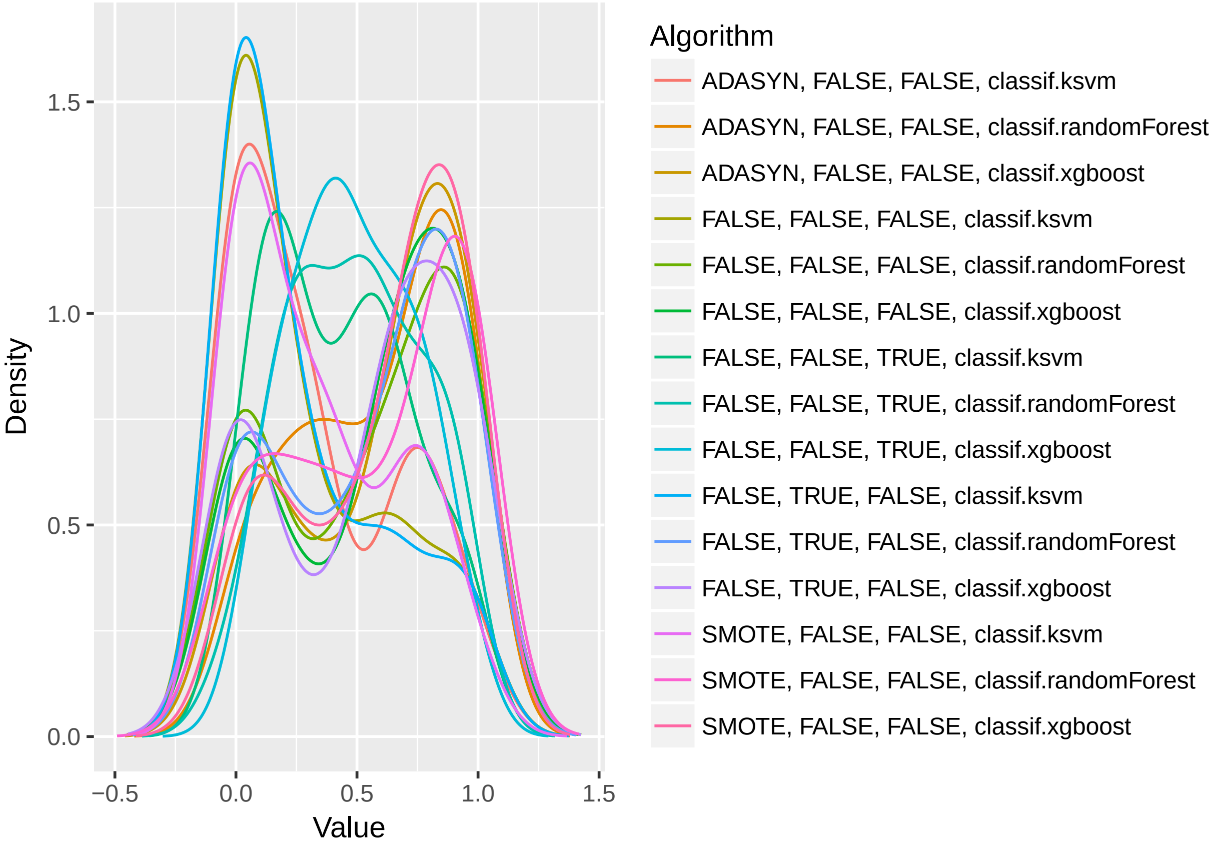
Verificando a média de cada coluna selecionada

```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 = 0.335572197834641"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.588595972988164"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.604122892911288"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.306380187943668"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.561499513381357"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.582620136319011"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.435278394058719"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.509540788091048"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.499208851875283"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.301364765853798"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.570461325051862"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.565591860844364"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.338001786891773"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.582171663710331"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.624815250426171"
```

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 = 254.96, 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,]                              FALSE
##  [9,]                               TRUE
## [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,]                                  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.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, FALSE, classif.randomForest
##  [1,]                                      TRUE
##  [2,]                                     FALSE
##  [3,]                                     FALSE
##  [4,]                                      TRUE
##  [5,]                                     FALSE
##  [6,]                                     FALSE
##  [7,]                                     FALSE
##  [8,]                                     FALSE
##  [9,]                                     FALSE
## [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,]                            FALSE
##  [6,]                             TRUE
##  [7,]                            FALSE
##  [8,]                            FALSE
##  [9,]                            FALSE
## [10,]                            FALSE
## [11,]                            FALSE
## [12,]                            FALSE
## [13,]                            FALSE
## [14,]                             TRUE
## [15,]                             TRUE
##       FALSE, FALSE, TRUE, classif.randomForest
##  [1,]                                    FALSE
##  [2,]                                    FALSE
##  [3,]                                     TRUE
##  [4,]                                     TRUE
##  [5,]                                    FALSE
##  [6,]                                    FALSE
##  [7,]                                    FALSE
##  [8,]                                    FALSE
##  [9,]                                    FALSE
## [10,]                                     TRUE
## [11,]                                    FALSE
## [12,]                                    FALSE
## [13,]                                    FALSE
## [14,]                                    FALSE
## [15,]                                     TRUE
##       FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  [1,]                                TRUE                            FALSE
##  [2,]                               FALSE                             TRUE
##  [3,]                                TRUE                             TRUE
##  [4,]                                TRUE                            FALSE
##  [5,]                               FALSE                             TRUE
##  [6,]                               FALSE                             TRUE
##  [7,]                               FALSE                            FALSE
##  [8,]                               FALSE                             TRUE
##  [9,]                               FALSE                             TRUE
## [10,]                                TRUE                            FALSE
## [11,]                               FALSE                             TRUE
## [12,]                               FALSE                             TRUE
## [13,]                                TRUE                            FALSE
## [14,]                               FALSE                             TRUE
## [15,]                                TRUE                             TRUE
##       FALSE, TRUE, FALSE, classif.randomForest
##  [1,]                                     TRUE
##  [2,]                                    FALSE
##  [3,]                                    FALSE
##  [4,]                                     TRUE
##  [5,]                                    FALSE
##  [6,]                                    FALSE
##  [7,]                                    FALSE
##  [8,]                                    FALSE
##  [9,]                                    FALSE
## [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,]                               FALSE
##  [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,]                             FALSE
##  [9,]                              TRUE
## [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,]                               TRUE
##  [9,]                               TRUE
## [10,]                               TRUE
## [11,]                              FALSE
## [12,]                              FALSE
## [13,]                               TRUE
## [14,]                              FALSE
## [15,]                              FALSE
```

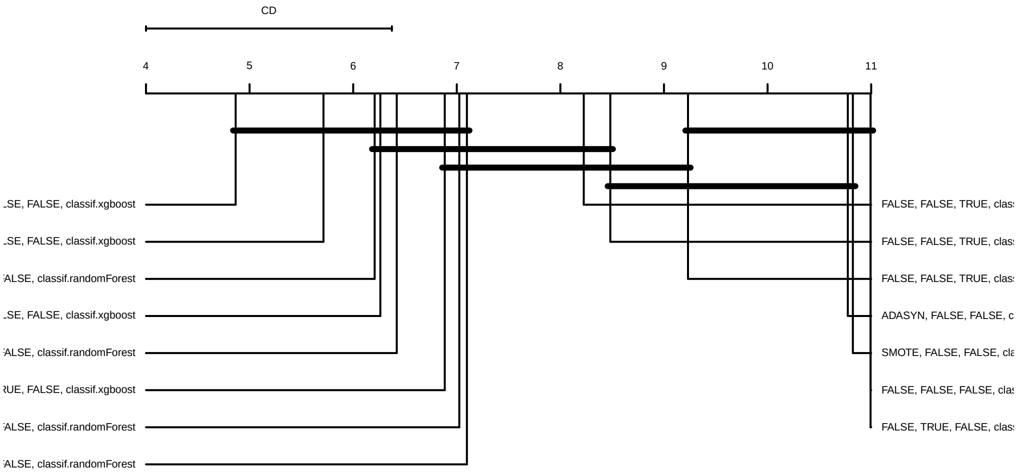
Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm 
##                                  10.774390 
## ADASYN, FALSE, FALSE, classif.randomForest 
##                                   6.420732 
##      ADASYN, FALSE, FALSE, classif.xgboost 
##                                   5.713415 
##          FALSE, FALSE, FALSE, classif.ksvm 
##                                  10.993902 
##  FALSE, FALSE, FALSE, classif.randomForest 
##                                   7.097561 
##       FALSE, FALSE, FALSE, classif.xgboost 
##                                   6.262195 
##           FALSE, FALSE, TRUE, classif.ksvm 
##                                   9.231707 
##   FALSE, FALSE, TRUE, classif.randomForest 
##                                   8.481707 
##        FALSE, FALSE, TRUE, classif.xgboost 
##                                   8.225610 
##           FALSE, TRUE, FALSE, classif.ksvm 
##                                  10.993902 
##   FALSE, TRUE, FALSE, classif.randomForest 
##                                   7.024390 
##        FALSE, TRUE, FALSE, classif.xgboost 
##                                   6.884146 
##          SMOTE, FALSE, FALSE, classif.ksvm 
##                                  10.823171 
##  SMOTE, FALSE, FALSE, classif.randomForest 
##                                   6.207317 
##       SMOTE, FALSE, FALSE, classif.xgboost 
##                                   4.865854
```

Plotando grafico de Critical Diference

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


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