R Notebook

Parametros:

Measure = Matthews correlation coefficient
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure
Filter keys = imba.rate
Filter values = 0.03

```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        :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                       :   0   SMOTE : 594   TRUE :594
##  G-mean                           :   0                 NA's :0
##  Matthews correlation coefficient:2970
##
##
##  tuning_measure     holdout_measure   holdout_measure_residual
##  Min.   :-0.05673   Min.   :-0.1757   Min.   :-0.4658
##  1st Qu.: 0.33347   1st Qu.: 0.0000   1st Qu.: 0.0391
##  Median : 0.83196   Median : 0.5030   Median : 0.2116
##  Mean   : 0.66187   Mean   : 0.4753   Mean   : 0.3111
##  3rd Qu.: 0.98596   3rd Qu.: 0.8126   3rd Qu.: 0.5286
##  Max.   : 1.00000   Max.   : 1.0000   Max.   : 1.0000
##  NA's   :48         NA's   :48        NA's   :48
##  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   :48      (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                       0.0004984056
## 2                       0.0882254134
## 3                       0.1438516324
## 4                       0.0000000000
## 5                       0.6666666667
## 6                       0.1158290535
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                               -0.003266437
## 2                                0.281634007
## 3                                0.579496962
## 4                               -0.005578849
## 5                                1.000000000
## 6                                0.135553899
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.03051619                      -0.007549758
## 2                            0.39385913                       0.146444566
## 3                            0.55028487                       0.184077857
## 4                            0.63523335                       0.000000000
## 5                            1.00000000                       1.000000000
## 6                            0.12698099                       0.023322357
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                               0.000000000
## 2                               0.469422453
## 3                               0.648010158
## 4                               0.666666667
## 5                               1.000000000
## 6                              -0.006915521
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                         -0.003371752                       0.08179207
## 2                          0.459268050                       0.22390593
## 3                          0.179229084                       0.33456239
## 4                          0.264961794                       0.06922774
## 5                          1.000000000                       0.69390964
## 6                          0.061768996                       0.08601424
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                               0.08784083
## 2                               0.26667419
## 3                               0.43525472
## 4                               0.38737909
## 5                               0.80611000
## 6                               0.25340519
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.1369229                       0.02304014
## 2                           0.2770432                       0.16296220
## 3                           0.3398779                       0.18407786
## 4                           0.4838392                       0.00000000
## 5                           0.4844443                       1.00000000
## 6                           0.2195653                       0.02332236
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                               0.00000000
## 2                               0.47858740
## 3                               0.77430184
## 4                               0.23287938
## 5                               1.00000000
## 6                               0.05547975
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                        -0.007158714                       0.018807164
## 2                         0.505558089                       0.105571021
## 3                         0.278411314                       0.196904632
## 4                         0.234311674                       0.000000000
## 5                         1.000000000                       0.900548805
## 6                         0.085861206                      -0.007023383
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                                0.02384633
## 2                                0.25952828
## 3                                0.46655072
## 4                               -0.01115770
## 5                                1.00000000
## 6                                0.10655223
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                           0.02496716
## 2                           0.41335425
## 3                           0.44780733
## 4                           0.26367954
## 5                           1.00000000
## 6                           0.06085747
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :-0.05271
##  1st Qu.: 0.00000
##  Median : 0.18318
##  Mean   : 0.31432
##  3rd Qu.: 0.61074
##  Max.   : 1.00000
##  NA's   :2
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :-0.03659
##  1st Qu.: 0.27348
##  Median : 0.67012
##  Mean   : 0.57264
##  3rd Qu.: 0.90622
##  Max.   : 1.00000
##  NA's   :4
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :-0.03545                      Min.   :-0.02749
##  1st Qu.: 0.33394                      1st Qu.: 0.00000
##  Median : 0.74292                      Median : 0.19991
##  Mean   : 0.61244                      Mean   : 0.33782
##  3rd Qu.: 0.90604                      3rd Qu.: 0.62881
##  Max.   : 1.00000                      Max.   : 1.00000
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :-0.01694
##  1st Qu.: 0.18891
##  Median : 0.65240
##  Mean   : 0.53893
##  3rd Qu.: 0.86172
##  Max.   : 1.00000
##  NA's   :2
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :-0.02329                     Min.   :-0.001398
##  1st Qu.: 0.19820                     1st Qu.: 0.151769
##  Median : 0.66636                     Median : 0.292077
##  Mean   : 0.56706                     Mean   : 0.377472
##  3rd Qu.: 0.89310                     3rd Qu.: 0.561228
##  Max.   : 1.00000                     Max.   : 1.000000
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :-0.01063
##  1st Qu.: 0.25493
##  Median : 0.42364
##  Mean   : 0.45629
##  3rd Qu.: 0.67576
##  Max.   : 1.00000
##  NA's   :2
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :-0.06218                    Min.   :-0.03234
##  1st Qu.: 0.24456                    1st Qu.: 0.00000
##  Median : 0.44184                    Median : 0.19061
##  Mean   : 0.44256                    Mean   : 0.31345
##  3rd Qu.: 0.62718                    3rd Qu.: 0.61580
##  Max.   : 1.00000                    Max.   : 1.00000
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :-0.01536
##  1st Qu.: 0.16897
##  Median : 0.58129
##  Mean   : 0.53543
##  3rd Qu.: 0.89025
##  Max.   : 1.00000
##  NA's   :2
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :-0.01536                    Min.   :-0.05605
##  1st Qu.: 0.24534                    1st Qu.: 0.00000
##  Median : 0.63612                    Median : 0.19055
##  Mean   : 0.57481                    Mean   : 0.29865
##  3rd Qu.: 0.88669                    3rd Qu.: 0.58657
##  Max.   : 1.00000                    Max.   : 0.98306
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :-0.0333
##  1st Qu.: 0.1848
##  Median : 0.6701
##  Mean   : 0.5744
##  3rd Qu.: 0.9309
##  Max.   : 1.0000
##  NA's   :4
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :-0.02848
##  1st Qu.: 0.33047
##  Median : 0.70453
##  Mean   : 0.62302
##  3rd Qu.: 0.92464
##  Max.   : 1.00000
##
```

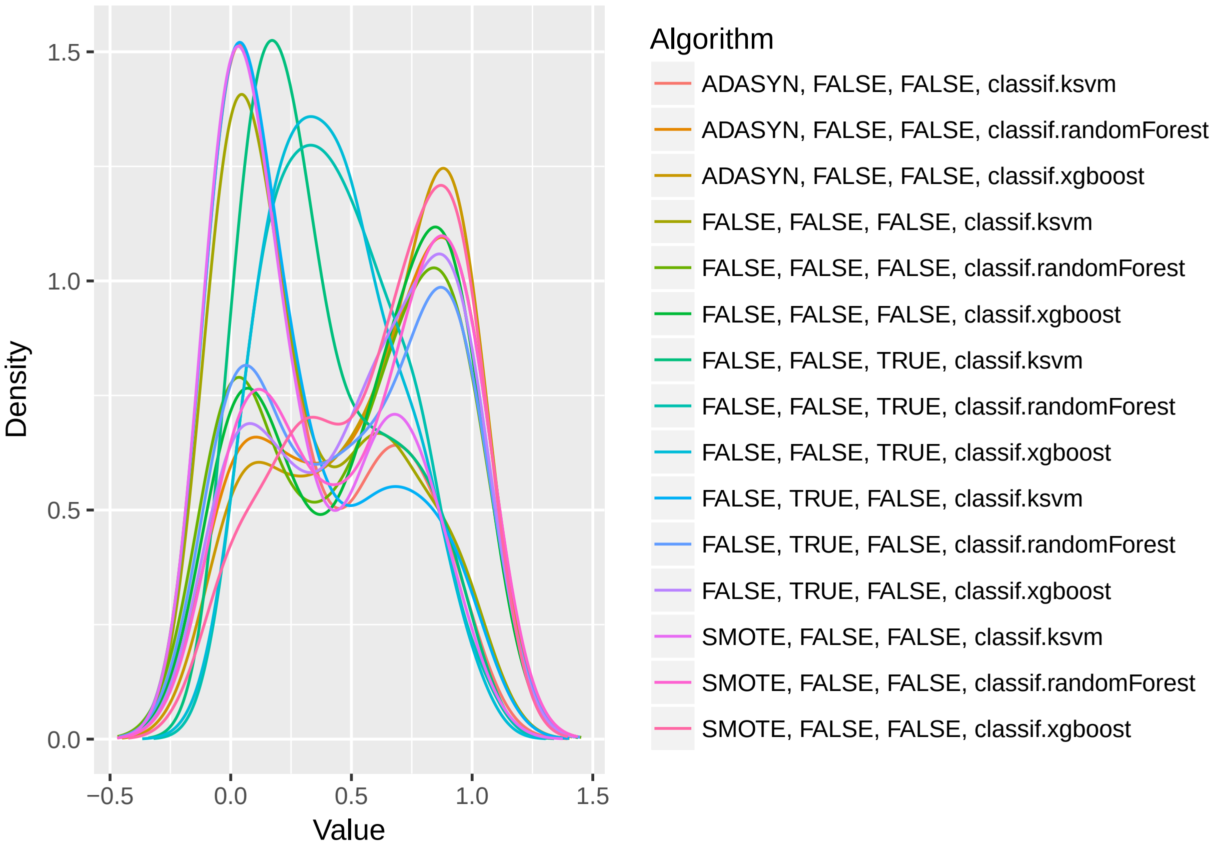
Verificando a média de cada coluna selecionada

```
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 = 0.314324422019971"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.572635610148972"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.612436645152332"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.33782362074705"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.538933940072554"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.567056380341611"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.377472226673554"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.456291185936359"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.442562977789507"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.313452789882849"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.535434300254414"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.574809901960192"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.298653225455856"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.574449391839337"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.623015345791242"
```

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 = 249.76, df = 14, p-value < 2.2e-16
```

Testando as diferencas par a par

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

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

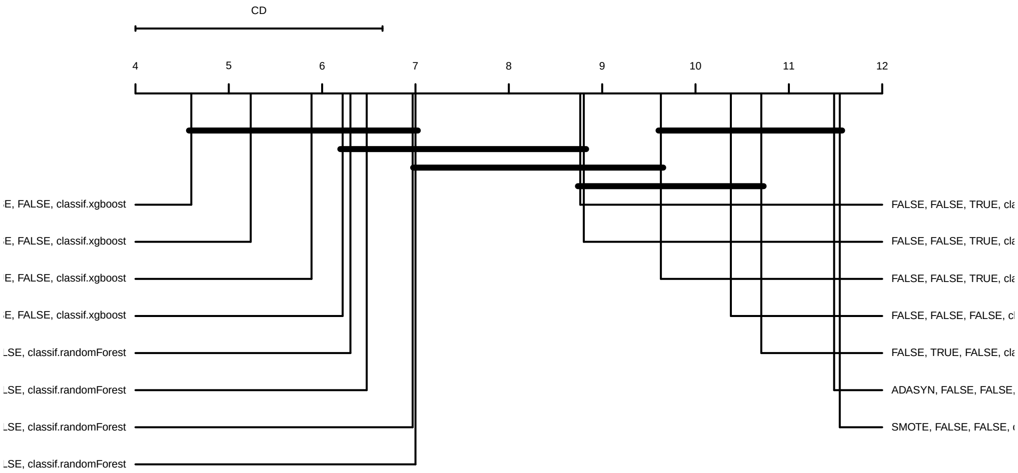
Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                  11.484848
## ADASYN, FALSE, FALSE, classif.randomForest
##                                   6.477273
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                   5.234848
##          FALSE, FALSE, FALSE, classif.ksvm
##                                  10.378788
##  FALSE, FALSE, FALSE, classif.randomForest
##                                   7.000000
##       FALSE, FALSE, FALSE, classif.xgboost
##                                   6.219697
##           FALSE, FALSE, TRUE, classif.ksvm
##                                   9.628788
##   FALSE, FALSE, TRUE, classif.randomForest
##                                   8.765152
##        FALSE, FALSE, TRUE, classif.xgboost
##                                   8.803030
##           FALSE, TRUE, FALSE, classif.ksvm
##                                  10.704545
##   FALSE, TRUE, FALSE, classif.randomForest
##                                   6.969697
##        FALSE, TRUE, FALSE, classif.xgboost
##                                   5.886364
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                  11.545455
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                   6.303030
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                   4.598485
```

Plotando grafico de Critical Diference

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


CD
E, FALSE, classif.xgboost
E, FALSE, classif.xgboost
E, FALSE, classif.xgboost
E, FALSE, classif.xgboost
LSE, classif.randomForest
LSE, classif.randomForest
LSE, classif.randomForest
LSE, classif.randomForest
FALSE, FALSE, TRUE, cla
FALSE, FALSE, TRUE, cla
FALSE, FALSE, TRUE, cla
FALSE, FALSE, FALSE, cl
FALSE, TRUE, FALSE, cla
ADASYN, FALSE, FALSE,
SMOTE, FALSE, FALSE,