R Notebook

Parametros:

Measure = F1 measure
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure_residual
Filter keys = imba.rate
Filter values = 0.001

```
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        :600   Mode :logical
##  classif.randomForest:600   FALSE:1440
##  classif.rusboost    :  0   TRUE :360
##  classif.xgboost     :600   NA's :0
##
##
##
##                             measure       sampling    underbagging
##  Accuracy                       :   0   ADASYN: 360   Mode :logical
##  Area under the curve           :   0   FALSE :1080   FALSE:1440
##  F1 measure                     :1800   SMOTE : 360   TRUE :360
##  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.:0.1444   1st Qu.:0.0000   1st Qu.:0.02254
##  Median :0.8072   Median :0.3333   Median :0.21133
##  Mean   :0.6196   Mean   :0.4116   Mean   :0.32573
##  3rd Qu.:0.9987   3rd Qu.:0.8000   3rd Qu.:0.59294
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##  NA's   :60       NA's   :60       NA's   :60
##  iteration_count                      dataset       imba.rate
##  Min.   :1       abalone                  :  45   Min.   :0.001
##  1st Qu.:1       adult                    :  45   1st Qu.:0.001
##  Median :2       bank                     :  45   Median :0.001
##  Mean   :2       car                      :  45   Mean   :0.001
##  3rd Qu.:3       cardiotocography-10clases:  45   3rd Qu.:0.001
##  Max.   :3       cardiotocography-3clases :  45   Max.   :0.001
##  NA's   :60      (Other)                  :1530
```

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] 40 15
```

```
# Renomeando a variavel
df = df_tec_wide_residual

head(df)
```

```
##   ADASYN, FALSE, FALSE, classif.ksvm
## 1                        0.101905890
## 2                        0.083783692
## 3                        0.004065052
## 4                        0.034684412
## 5                        0.000000000
## 6                        0.032084899
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                 0.07466998
## 2                                         NA
## 3                                 0.01617013
## 4                                 0.11469128
## 5                                 0.05854360
## 6                                 0.39394571
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.03799017                        0.03448112
## 2                            0.35860150                        0.07719857
## 3                            0.08812821                        0.00000000
## 4                            0.11101906                        0.10702333
## 5                            0.33683835                        0.13077942
## 6                            0.43433032                        0.08378653
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                              0.0005124263
## 2                              0.3814761838
## 3                              0.0000000000
## 4                              0.1256252991
## 5                              0.1333688405
## 6                              0.4087131894
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                          0.001536492                       0.75056379
## 2                          0.338343423                       0.56827651
## 3                          0.010796355                       0.63842237
## 4                          0.097788598                       0.69934641
## 5                          0.123638344                       0.01990050
## 6                          0.433479009                       0.05623665
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                                0.7453236
## 2                                0.8501727
## 3                                0.7112993
## 4                                0.1989499
## 5                                0.7346306
## 6                                0.6434506
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.7348548                       0.03255014
## 2                           0.8479729                       0.08452969
## 3                           0.6880691                       0.00000000
## 4                           0.2501528                       0.10702333
## 5                           0.7137299                       0.13077942
## 6                           0.5361180                       0.08378653
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                             0.0005124263
## 2                             0.3628587010
## 3                             0.0000000000
## 4                             0.1034862753
## 5                             0.1598001860
## 6                             0.4035051554
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                        0.0005124263                       0.097251914
## 2                        0.3426039783                       0.090957308
## 3                        0.0067696714                       0.006769571
## 4                        0.0881277020                       0.077783026
## 5                        0.1156781553                       0.010101010
## 6                        0.4309032656                       0.016512026
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                                0.07952866
## 2                                        NA
## 3                                0.04168425
## 4                                0.11900875
## 5                                0.03980100
## 6                                0.35936253
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                           0.03895394
## 2                           0.33593996
## 3                           0.07489552
## 4                           0.08043011
## 5                           0.31420828
## 6                           0.45266705
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.02349
##  Mean   :0.12999
##  3rd Qu.:0.15527
##  Max.   :0.87407
##  NA's   :2
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.00000
##  1st Qu.:0.07242
##  Median :0.30115
##  Mean   :0.34750
##  3rd Qu.:0.55764
##  Max.   :0.95284
##  NA's   :7
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.01342                      Min.   :0.00000
##  1st Qu.:0.10530                      1st Qu.:0.00000
##  Median :0.41085                      Median :0.08607
##  Mean   :0.42564                      Mean   :0.19190
##  3rd Qu.:0.69588                      3rd Qu.:0.24274
##  Max.   :0.96197                      Max.   :0.88492
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.00000
##  1st Qu.:0.02265
##  Median :0.16357
##  Mean   :0.28066
##  3rd Qu.:0.46176
##  Max.   :0.90114
##  NA's   :2
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.00000                     Min.   :0.0003698
##  1st Qu.:0.04094                     1st Qu.:0.0522234
##  Median :0.25322                     Median :0.4759320
##  Mean   :0.31416                     Mean   :0.4073185
##  3rd Qu.:0.48773                     3rd Qu.:0.6620882
##  Max.   :0.90794                     Max.   :0.9371130
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.006542
##  1st Qu.:0.248023
##  Median :0.689560
##  Mean   :0.566743
##  3rd Qu.:0.877669
##  Max.   :0.981822
##  NA's   :1
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.004514                    Min.   :0.00000
##  1st Qu.:0.246839                    1st Qu.:0.00000
##  Median :0.670794                    Median :0.08416
##  Mean   :0.565092                   Mean   :0.18858
##  3rd Qu.:0.862288                   3rd Qu.:0.24274
##  Max.   :0.960136                   Max.   :0.88492
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.00000
##  1st Qu.:0.02238
##  Median :0.16167
##  Mean   :0.26334
##  3rd Qu.:0.40896
##  Max.   :0.88753
##  NA's   :1
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.00000                     Min.   :0.0000000
##  1st Qu.:0.04215                     1st Qu.:0.0005234
##  Median :0.23620                     Median :0.0475759
##  Mean   :0.31816                     Mean   :0.1289498
##  3rd Qu.:0.48482                     3rd Qu.:0.1339689
##  Max.   :0.90326                     Max.   :0.6924416
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.00000
##  1st Qu.:0.04168
##  Median :0.20099
##  Mean   :0.31822
##  3rd Qu.:0.59058
##  Max.   :0.93333
##  NA's   :7
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.00000
##  1st Qu.:0.09062
##  Median :0.43634
##  Mean   :0.43460
##  3rd Qu.:0.71944
##  Max.   :0.95598
##
```

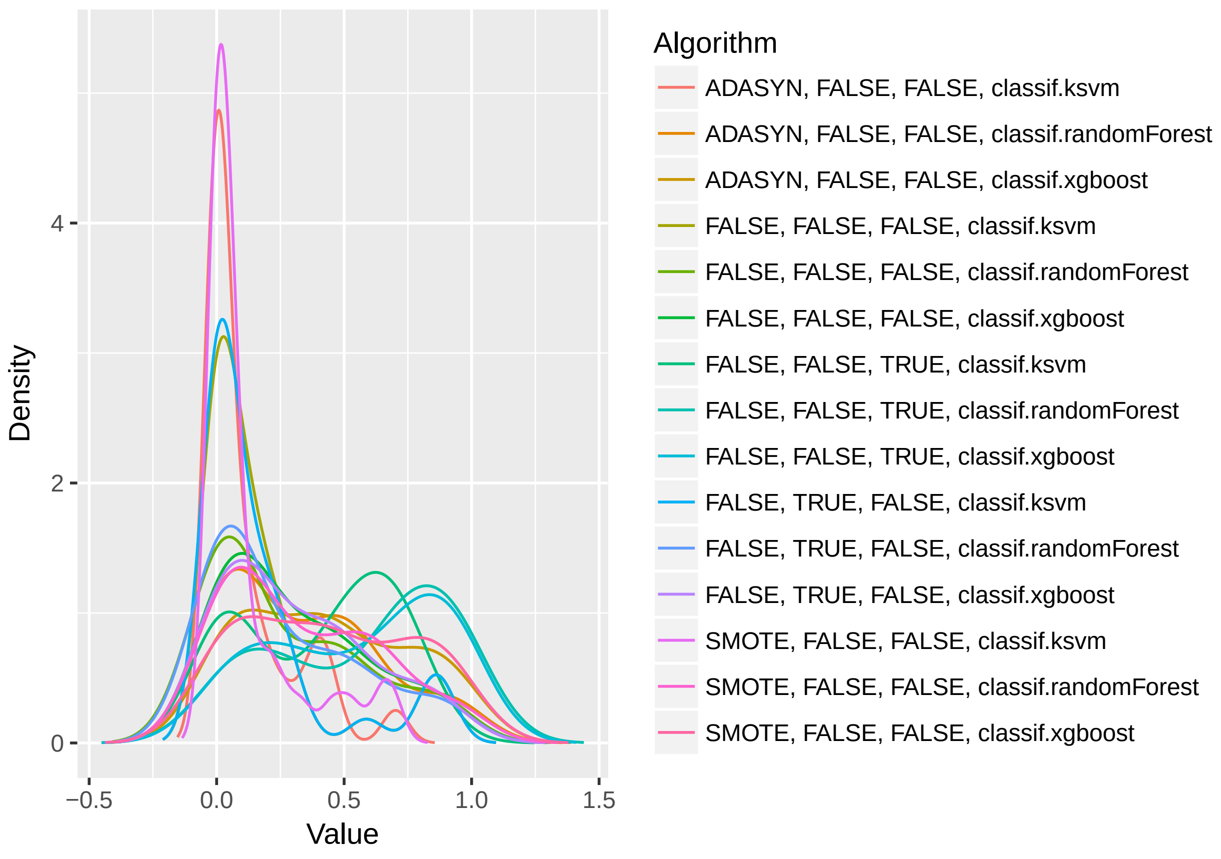
Verificando a média de cada coluna selecionada

```
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 = 0.129992981439635"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.347499805052361"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.425642771342025"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.191895194681838"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.280655929773395"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.314160504533314"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.407318499678718"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.566743232938743"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.565091698285878"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.188584864404373"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.263341243601662"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.318155745625504"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.128949800109106"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.318219063001826"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.434601042804444"
```

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 = 193.89, 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,]                               TRUE
##  [8,]                               TRUE
##  [9,]                               TRUE
## [10,]                              FALSE
## [11,]                              FALSE
## [12,]                               TRUE
## [13,]                              FALSE
## [14,]                               TRUE
## [15,]                               TRUE
##       ADASYN, FALSE, FALSE, classif.randomForest
##  [1,]                                       TRUE
##  [2,]                                      FALSE
##  [3,]                                       TRUE
##  [4,]                                      FALSE
##  [5,]                                      FALSE
##  [6,]                                      FALSE
##  [7,]                                      FALSE
##  [8,]                                       TRUE
##  [9,]                                       TRUE
## [10,]                                      FALSE
## [11,]                                      FALSE
## [12,]                                      FALSE
## [13,]                                      FALSE
## [14,]                                      FALSE
## [15,]                                      FALSE
##       ADASYN, FALSE, FALSE, classif.xgboost
##  [1,]                                  TRUE
##  [2,]                                  TRUE
##  [3,]                                 FALSE
##  [4,]                                  TRUE
##  [5,]                                  TRUE
##  [6,]                                 FALSE
##  [7,]                                 FALSE
##  [8,]                                 FALSE
##  [9,]                                 FALSE
## [10,]                                  TRUE
## [11,]                                  TRUE
## [12,]                                 FALSE
## [13,]                                  TRUE
## [14,]                                  TRUE
## [15,]                                 FALSE
##       FALSE, FALSE, FALSE, classif.ksvm
##  [1,]                             FALSE
##  [2,]                             FALSE
##  [3,]                              TRUE
##  [4,]                             FALSE
##  [5,]                             FALSE
##  [6,]                             FALSE
##  [7,]                              TRUE
##  [8,]                              TRUE
##  [9,]                              TRUE
## [10,]                             FALSE
## [11,]                             FALSE
## [12,]                             FALSE
## [13,]                             FALSE
## [14,]                             FALSE
## [15,]                              TRUE
##       FALSE, FALSE, FALSE, classif.randomForest
##  [1,]                                      TRUE
##  [2,]                                     FALSE
##  [3,]                                      TRUE
##  [4,]                                     FALSE
##  [5,]                                     FALSE
##  [6,]                                     FALSE
##  [7,]                                     FALSE
##  [8,]                                      TRUE
##  [9,]                                      TRUE
## [10,]                                     FALSE
## [11,]                                     FALSE
## [12,]                                     FALSE
## [13,]                                     FALSE
## [14,]                                     FALSE
## [15,]                                      TRUE
##       FALSE, FALSE, FALSE, classif.xgboost
##  [1,]                                 TRUE
##  [2,]                                FALSE
##  [3,]                                FALSE
##  [4,]                                FALSE
##  [5,]                                FALSE
##  [6,]                                FALSE
##  [7,]                                FALSE
##  [8,]                                 TRUE
##  [9,]                                 TRUE
## [10,]                                FALSE
## [11,]                                FALSE
## [12,]                                FALSE
## [13,]                                 TRUE
## [14,]                                FALSE
## [15,]                                FALSE
##       FALSE, FALSE, TRUE, classif.ksvm
##  [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, TRUE, classif.randomForest
##  [1,]                                     TRUE
##  [2,]                                     TRUE
##  [3,]                                    FALSE
##  [4,]                                     TRUE
##  [5,]                                     TRUE
##  [6,]                                     TRUE
##  [7,]                                    FALSE
##  [8,]                                    FALSE
##  [9,]                                    FALSE
## [10,]                                     TRUE
## [11,]                                     TRUE
## [12,]                                     TRUE
## [13,]                                     TRUE
## [14,]                                     TRUE
## [15,]                                    FALSE
##       FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  [1,]                                TRUE                            FALSE
##  [2,]                                TRUE                            FALSE
##  [3,]                               FALSE                             TRUE
##  [4,]                                TRUE                            FALSE
##  [5,]                                TRUE                            FALSE
##  [6,]                                TRUE                            FALSE
##  [7,]                               FALSE                             TRUE
##  [8,]                               FALSE                             TRUE
##  [9,]                               FALSE                             TRUE
## [10,]                                TRUE                            FALSE
## [11,]                                TRUE                            FALSE
## [12,]                                TRUE                            FALSE
## [13,]                                TRUE                            FALSE
## [14,]                                TRUE                            FALSE
## [15,]                               FALSE                             TRUE
##       FALSE, TRUE, FALSE, classif.randomForest
##  [1,]                                    FALSE
##  [2,]                                    FALSE
##  [3,]                                     TRUE
##  [4,]                                    FALSE
##  [5,]                                    FALSE
##  [6,]                                    FALSE
##  [7,]                                    FALSE
##  [8,]                                     TRUE
##  [9,]                                     TRUE
## [10,]                                    FALSE
## [11,]                                    FALSE
## [12,]                                FALSE
## [13,]                                FALSE
## [14,]                                FALSE
## [15,]                                 TRUE
##       FALSE, TRUE, FALSE, classif.xgboost
##  [1,]                            TRUE
##  [2,]                           FALSE
##  [3,]                           FALSE
##  [4,]                           FALSE
##  [5,]                           FALSE
##  [6,]                           FALSE
##  [7,]                           FALSE
##  [8,]                            TRUE
##  [9,]                            TRUE
## [10,]                           FALSE
## [11,]                           FALSE
## [12,]                           FALSE
## [13,]                            TRUE
## [14,]                           FALSE
## [15,]                           FALSE
##       SMOTE, FALSE, FALSE, classif.ksvm
##  [1,]                         FALSE
##  [2,]                         FALSE
##  [3,]                          TRUE
##  [4,]                         FALSE
##  [5,]                         FALSE
##  [6,]                          TRUE
##  [7,]                          TRUE
##  [8,]                          TRUE
##  [9,]                          TRUE
## [10,]                         FALSE
## [11,]                         FALSE
## [12,]                          TRUE
## [13,]                         FALSE
## [14,]                         FALSE
## [15,]                          TRUE
##       SMOTE, FALSE, FALSE, classif.randomForest
##  [1,]                                  TRUE
##  [2,]                                 FALSE
##  [3,]                                  TRUE
##  [4,]                                 FALSE
##  [5,]                                 FALSE
##  [6,]                                 FALSE
##  [7,]                                 FALSE
##  [8,]                                  TRUE
##  [9,]                                  TRUE
## [10,]                                 FALSE
## [11,]                                 FALSE
## [12,]                                 FALSE
## [13,]                                 FALSE
## [14,]                                 FALSE
## [15,]                                  TRUE
##       SMOTE, FALSE, FALSE, classif.xgboost
##  [1,]                             TRUE
##  [2,]                               FALSE
##  [3,]                               FALSE
##  [4,]                                TRUE
##  [5,]                                TRUE
##  [6,]                               FALSE
##  [7,]                               FALSE
##  [8,]                               FALSE
##  [9,]                               FALSE
## [10,]                                TRUE
## [11,]                                TRUE
## [12,]                               FALSE
## [13,]                                TRUE
## [14,]                                TRUE
## [15,]                               FALSE
```

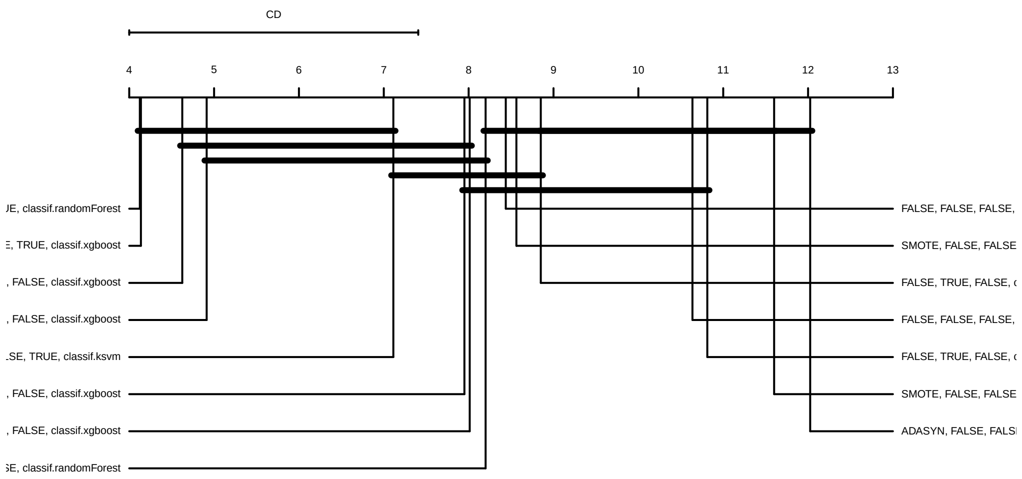
Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm 
##                                    12.0250 
## ADASYN, FALSE, FALSE, classif.randomForest 
##                                     8.2000 
##      ADASYN, FALSE, FALSE, classif.xgboost 
##                                     4.6250 
##          FALSE, FALSE, FALSE, classif.ksvm 
##                                    10.6375 
##  FALSE, FALSE, FALSE, classif.randomForest 
##                                     8.4375 
##       FALSE, FALSE, FALSE, classif.xgboost 
##                                     8.0125 
##           FALSE, FALSE, TRUE, classif.ksvm 
##                                     7.1125 
##   FALSE, FALSE, TRUE, classif.randomForest 
##                                     4.1250 
##        FALSE, FALSE, TRUE, classif.xgboost 
##                                     4.1375 
##           FALSE, TRUE, FALSE, classif.ksvm 
##                                    10.8125 
##   FALSE, TRUE, FALSE, classif.randomForest 
##                                     8.8500 
##        FALSE, TRUE, FALSE, classif.xgboost 
##                                     7.9500 
##          SMOTE, FALSE, FALSE, classif.ksvm 
##                                    11.6000 
##  SMOTE, FALSE, FALSE, classif.randomForest 
##                                     8.5625 
##       SMOTE, FALSE, FALSE, classif.xgboost 
##                                     4.9125 
```

Plotando grafico de Critical Diference

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


CD
JE, classif.randomForest
E, TRUE, classif.xgboost
, FALSE, classif.xgboost
, FALSE, classif.xgboost
.SE, TRUE, classif.ksvm
, FALSE, classif.xgboost
, FALSE, classif.xgboost
SE, classif.randomForest
FALSE, FALSE, FALSE,
SMOTE, FALSE, FALSE
FALSE, TRUE, FALSE, c
FALSE, FALSE, FALSE,
FALSE, TRUE, FALSE, c
SMOTE, FALSE, FALSE
ADASYN, FALSE, FALSI