R Notebook

Parametros:

Measure = F1 measure
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure_residual
Filter keys = imba.rate
Filter values = 0.03

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

Tratamento dos dados

Carregando data set compilado

```
ds = read.csv("/home/rodrigo/Dropbox/UNICAMP/IC/estudo_cost_learning/SummaryResults/summary_compilation

ds = filter(ds, learner != "classif.rusboost")
summary(ds)
```

```
##                    learner      weight_space
##  classif.ksvm        :17100   Mode :logical
##  classif.randomForest:17100   FALSE:41040
##  classif.rusboost    :    0   TRUE :10260
##  classif.xgboost     :17100   NA's :0
##
##
##
##                                measure        sampling     underbagging
##  Accuracy                          :10260   ADASYN:10260   Mode :logical
##  Area under the curve              :10260   FALSE :30780   FALSE:41040
##  F1 measure                        :10260   SMOTE :10260   TRUE :10260
##  G-mean                            :10260                  NA's :0
##  Matthews correlation coefficient:10260
##
##
##  tuning_measure    holdout_measure   holdout_measure_residual
##  Min.   :-0.1277   Min.   :-0.2120   Min.   :-0.4658
##  1st Qu.: 0.6911   1st Qu.: 0.4001   1st Qu.: 0.1994
##  Median : 0.9700   Median : 0.8571   Median : 0.5581
##  Mean   : 0.7903   Mean   : 0.6718   Mean   : 0.5298
##  3rd Qu.: 0.9975   3rd Qu.: 0.9900   3rd Qu.: 0.8755
##  Max.   : 1.0000   Max.   : 1.0000   Max.   : 1.0000
##  NA's   :1077      NA's   :1077      NA's   :1077
##  iteration_count                      dataset         imba.rate
##  Min.   :1       abalone                  :  900   Min.   :0.0010
##  1st Qu.:1       adult                    :  900   1st Qu.:0.0100
##  Median :2       bank                     :  900   Median :0.0300
##  Mean   :2       car                      :  900   Mean   :0.0286
##  3rd Qu.:3       cardiotocography-10clases:  900   3rd Qu.:0.0500
##  Max.   :3       cardiotocography-3clases :  900   Max.   :0.0500
##  NA's   :1077    (Other)                  :45900
```

Filtrando pela metrica

```
ds = filter(ds, measure == params$measure)
```

Filtrando o data set

```
if(params$filter_keys != 'NULL' && !is.null(params$filter_keys)){
  dots = paste0(params$filter_keys," == '",params$filter_values,"'")
  ds = filter_(ds, .dots = dots)
}

summary(ds)
```

```
##                 learner    weight_space
##  classif.ksvm        :990   Mode :logical
##  classif.randomForest:990   FALSE:2376
##  classif.rusboost    :  0   TRUE :594
##  classif.xgboost     :990   NA's :0
##
##
##
##                              measure       sampling    underbagging
##  Accuracy                        :   0   ADASYN: 594   Mode :logical
##  Area under the curve            :   0   FALSE :1782   FALSE:2376
##  F1 measure                      :2970   SMOTE : 594   TRUE :594
##  G-mean                          :   0                 NA's :0
##  Matthews correlation coefficient:   0
##
##
##  tuning_measure   holdout_measure  holdout_measure_residual
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.2788   1st Qu.:0.0481   1st Qu.:0.04815
##  Median :0.8296   Median :0.4840   Median :0.28571
##  Mean   :0.6542   Mean   :0.4646   Mean   :0.37464
##  3rd Qu.:0.9927   3rd Qu.:0.8000   3rd Qu.:0.70061
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##  NA's   :51       NA's   :51       NA's   :51
##  iteration_count         dataset       imba.rate
##  Min.   :1       abalone      :  45   Min.   :0.03
##  1st Qu.:1       adult        :  45   1st Qu.:0.03
##  Median :2       annealing    :  45   Median :0.03
##  Mean   :2       arrhythmia   :  45   Mean   :0.03
##  3rd Qu.:3       balance-scale:  45   3rd Qu.:0.03
##  Max.   :3       bank         :  45   Max.   :0.03
##  NA's   :51      (Other)      :2700
```

Computando as médias das iteracoes

```
ds = group_by(ds, learner , weight_space , measure , sampling , underbagging , dataset , imba.rate)
ds = summarise(ds, tuning_measure = mean(tuning_measure), holdout_measure = mean(holdout_measure),
               holdout_measure_residual = mean(holdout_measure_residual))

ds = as.data.frame(ds)
```

Criando dataframe

```
# Dividindo o ds em n, um para cada técnica
splited_df = ds %>% group_by_at(.vars = params$columns) %>% do(vals = as.data.frame(.)) %>% select(vals

# Juntando cada uma das partes horizontalmente em um data set
df_tec_wide = do.call("cbind", splited_df)

# Renomeando duplicacao de nomes
colnames(df_tec_wide) = make.unique(colnames(df_tec_wide))

# Selecionando apenas as medidas da performance escolhida
df_tec_wide_residual = select(df_tec_wide, matches(paste("^", params$performance, "$|", params$performa

# Renomeando colunas
new_names = NULL
for(i in (1:length(splited_df))){
  id = toString(sapply(splited_df[[i]][1, params$columns], as.character))
  new_names = c(new_names, id)
}
colnames(df_tec_wide_residual) = new_names

# Verificando a dimensao do df
dim(df_tec_wide_residual)
```

```
## [1] 66 15
```

```
# Renomeando a variavel
df = df_tec_wide_residual

head(df)
```

```
##   ADASYN, FALSE, FALSE, classif.ksvm
## 1                         0.21605452
## 2                         0.25222439
## 3                         0.43272429
## 4                         0.00000000
## 5                         0.01179527
## 6                         0.17141022
##   ADASYN, FALSE, FALSE, classif.randomForest
## 1                                  0.2014036
## 2                                         NA
## 3                                  0.7882299
## 4                                  0.0000000
## 5                                  0.1805687
## 6                                  0.2013623
##   ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
## 1                            0.09169446                        0.14887811
## 2                            0.47278517                        0.25340122
## 3                            0.55254625                        0.21114404
## 4                            0.48412698                        0.00000000
## 5                            0.19326460                        0.17351608
## 6                            0.22415097                        0.02262443
##   FALSE, FALSE, FALSE, classif.randomForest
## 1                                0.00000000
## 2                                0.42702445
## 3                                0.68421053
## 4                                0.66666667
## 5                                0.20762312
## 6                                0.06809004
##   FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
## 1                           0.01055605                       0.72122868
## 2                           0.40784939                       0.79040534
## 3                           0.51816486                       0.57989418
## 4                           0.26666667                       0.05797101
## 5                           0.10774411                       0.15452774
## 6                           0.12575546                       0.57195077
##   FALSE, FALSE, TRUE, classif.randomForest
## 1                                0.7198507
## 2                                0.8500476
## 3                                0.8393457
## 4                                0.3267196
## 5                                0.3895503
## 6                                0.7351026
##   FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
## 1                           0.7287353                       0.12147113
## 2                           0.8498716                       0.25573787
## 3                           0.7184903                       0.21114404
## 4                           0.4148148                       0.00000000
## 5                           0.4590204                       0.17351608
## 6                           0.7103109                       0.02262443
##   FALSE, TRUE, FALSE, classif.randomForest
## 1                              0.001593625
## 2                              0.426657416
## 3                              0.681883673
## 4                              0.222222222
## 5                              0.211287329
## 6                              0.065761489
##   FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
## 1                         0.001592357                        0.19840898
## 2                         0.420546937                        0.26357526
## 3                         0.511815854                        0.35061227
## 4                         0.222222222                        0.00000000
## 5                         0.107744108                        0.15009040
## 6                         0.123227841                        0.03068515
##   SMOTE, FALSE, FALSE, classif.randomForest
## 1                                 0.2099705
## 2                                        NA
## 3                                 0.6988345
## 4                                 0.0000000
## 5                                 0.2072956
## 6                                 0.2383112
##   SMOTE, FALSE, FALSE, classif.xgboost
## 1                            0.1039828
## 2                            0.4617937
## 3                            0.5992352
## 4                            0.2777778
## 5                            0.1932646
## 6                            0.2211338
```

```r
summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.00000
##  1st Qu.:0.01124
##  Median :0.13588
##  Mean   :0.20924
##  3rd Qu.:0.33839
##  Max.   :0.92728
##  NA's   :2
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.1011
##  Median :0.3570
##  Mean   :0.3972
##  3rd Qu.:0.6689
##  Max.   :0.9922
##  NA's   :6
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.01366                      Min.   :0.000000
##  1st Qu.:0.11040                      1st Qu.:0.002897
##  Median :0.39645                      Median :0.107671
##  Mean   :0.45372                      Mean   :0.199343
##  3rd Qu.:0.75826                      3rd Qu.:0.232915
##  Max.   :0.99746                      Max.   :0.975558
##
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.00000
##  1st Qu.:0.05789
##  Median :0.28145
##  Mean   :0.36246
##  3rd Qu.:0.65333
##  Max.   :1.00000
##  NA's   :1
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.00000                     Min.   :0.0001813
##  1st Qu.:0.05587                     1st Qu.:0.1602817
##  Median :0.33517                     Median :0.4955782
##  Mean   :0.37983                     Mean   :0.4717885
##  3rd Qu.:0.66549                     3rd Qu.:0.7636142
##  Max.   :0.99746                     Max.   :0.9758570
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.003676
##  1st Qu.:0.349687
##  Median :0.670067
##  Mean   :0.596381
##  3rd Qu.:0.874486
##  Max.   :0.979713
##  NA's   :2
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.003441                   Min.   :0.0000000
##  1st Qu.:0.290797                   1st Qu.:0.0004474
##  Median :0.693203                   Median :0.1076705
##  Mean   :0.577925                     Mean   :0.1883371
##  3rd Qu.:0.857315                     3rd Qu.:0.2313231
##  Max.   :0.974897                     Max.   :0.9755581
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.00000
##  1st Qu.:0.04889
##  Median :0.28847
##  Mean   :0.35090
##  3rd Qu.:0.62656
##  Max.   :0.99479
##  NA's   :1
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.00000                     Min.   :0.000000
##  1st Qu.:0.05852                     1st Qu.:0.006644
##  Median :0.36914                     Median :0.108291
##  Mean   :0.37924                     Mean   :0.200926
##  3rd Qu.:0.65192                     3rd Qu.:0.311243
##  Max.   :0.99746                     Max.   :0.980870
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.1174
##  Median :0.3447
##  Mean   :0.4053
##  3rd Qu.:0.6951
##  Max.   :0.9975
##  NA's   :5
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.008963
##  1st Qu.:0.130091
##  Median :0.407590
##  Mean   :0.452571
##  3rd Qu.:0.751440
##  Max.   :1.000000
##
```

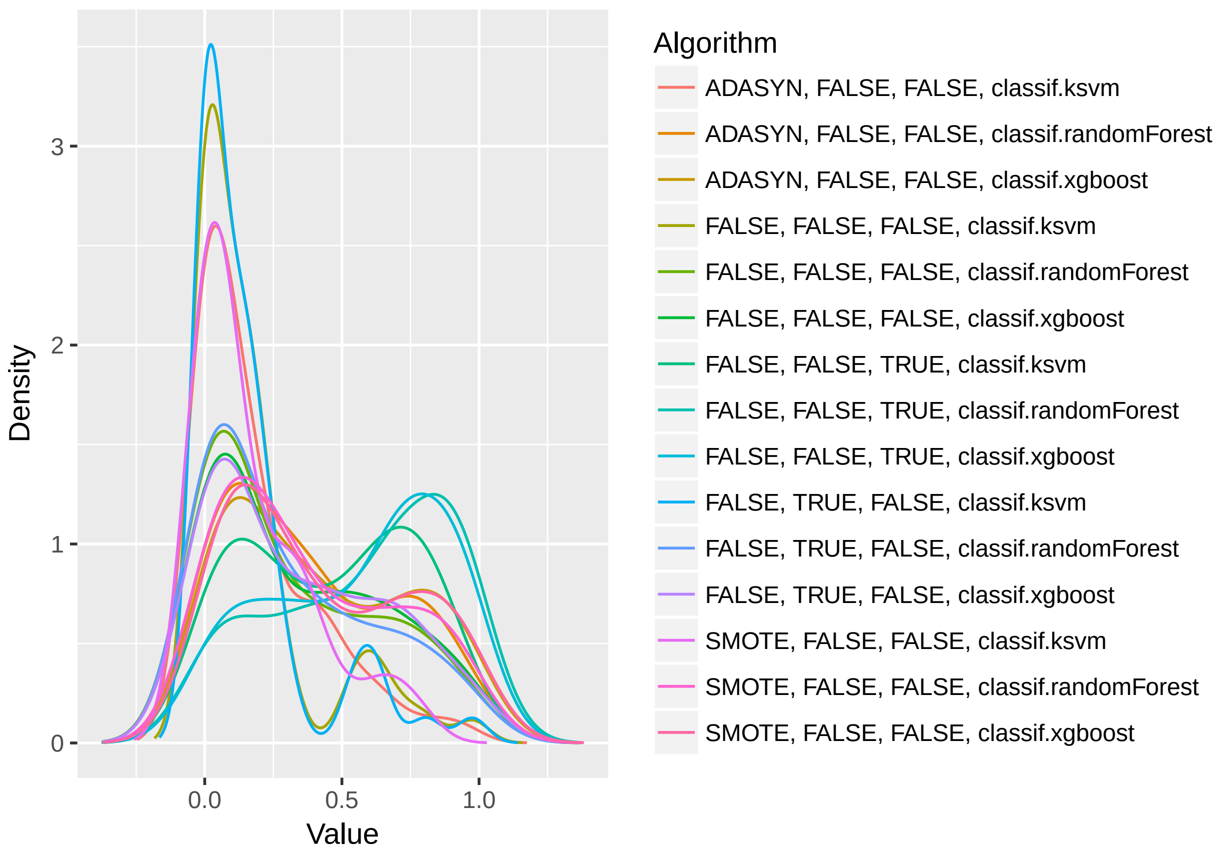
Verificando a média de cada coluna selecionada

```
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 = 0.209236054983801"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.397188536748764"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.453721019903491"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.199342907092154"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.362463625967697"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.379833571681706"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.47178849031539"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.596380859628329"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.577924935532121"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.18833708203037"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.350895051625677"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.379242774200312"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.200926355771355"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.405266686954057"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.452571401039357"
```

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 = 282.32, 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,]                              FALSE
##  [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,]                                      FALSE
##  [4,]                                       TRUE
##  [5,]                                      FALSE
##  [6,]                                      FALSE
##  [7,]                                      FALSE
##  [8,]                                       TRUE
##  [9,]                                       TRUE
## [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,]                                  TRUE
##  [6,]                                  TRUE
##  [7,]                                 FALSE
##  [8,]                                 FALSE
##  [9,]                                 FALSE
## [10,]                                  TRUE
## [11,]                                  TRUE
## [12,]                                  TRUE
## [13,]                                  TRUE
## [14,]                                 FALSE
## [15,]                                 FALSE
##        FALSE, FALSE, FALSE, classif.ksvm
##  [1,]                              FALSE
##  [2,]                               TRUE
##  [3,]                               TRUE
##  [4,]                              FALSE
##  [5,]                              FALSE
##  [6,]                               TRUE
##  [7,]                               TRUE
##  [8,]                               TRUE
##  [9,]                               TRUE
## [10,]                              FALSE
## [11,]                              FALSE
## [12,]                               TRUE
## [13,]                              FALSE
## [14,]                               TRUE
## [15,]                               TRUE
##        FALSE, FALSE, 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, FALSE, FALSE, classif.xgboost
##  [1,]                                  TRUE
##  [2,]                                 FALSE
##  [3,]                                  TRUE
##  [4,]                                  TRUE
##  [5,]                                 FALSE
##  [6,]                                 FALSE
##  [7,]                                 FALSE
##  [8,]                                  TRUE
##  [9,]                                  TRUE
## [10,]                                  TRUE
## [11,]                                 FALSE
## [12,]                                 FALSE
## [13,]                                  TRUE
## [14,]                                 FALSE
## [15,]                                  TRUE
##        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,]                             TRUE
## [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                             TRUE
##  [3,]                               FALSE                             TRUE
##  [4,]                                TRUE                            FALSE
##  [5,]                                TRUE                            FALSE
##  [6,]                                TRUE                             TRUE
##  [7,]                               FALSE                             TRUE
##  [8,]                               FALSE                             TRUE
##  [9,]                               FALSE                             TRUE
## [10,]                                TRUE                            FALSE
## [11,]                                TRUE                            FALSE
## [12,]                                TRUE                             TRUE
## [13,]                                TRUE                            FALSE
## [14,]                                TRUE                             TRUE
## [15,]                               FALSE                             TRUE
##       FALSE, TRUE, FALSE, classif.randomForest
##  [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, TRUE, FALSE, classif.xgboost
##  [1,]                              TRUE
##  [2,]                             FALSE
##  [3,]                              TRUE
##  [4,]                              TRUE
##  [5,]                             FALSE
##  [6,]                             FALSE
##  [7,]                             FALSE
##  [8,]                              TRUE
##  [9,]                              TRUE
## [10,]                              TRUE
## [11,]                             FALSE
## [12,]                             FALSE
## [13,]                              TRUE
## [14,]                             FALSE
## [15,]                              TRUE
##        SMOTE, FALSE, FALSE, classif.ksvm
##  [1,]                           FALSE
##  [2,]                            TRUE
##  [3,]                            TRUE
##  [4,]                           FALSE
##  [5,]                           FALSE
##  [6,]                            TRUE
##  [7,]                            TRUE
##  [8,]                            TRUE
##  [9,]                            TRUE
## [10,]                           FALSE
## [11,]                           FALSE
## [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,]                                   FALSE
##  [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,]                               TRUE
##  [6,]                               TRUE
##  [7,]                              FALSE
##  [8,]                              FALSE
##  [9,]                              FALSE
## [10,]                               TRUE
## [11,]                               TRUE
## [12,]                               TRUE
## [13,]                               TRUE
## [14,]                              FALSE
## [15,]                              FALSE
```

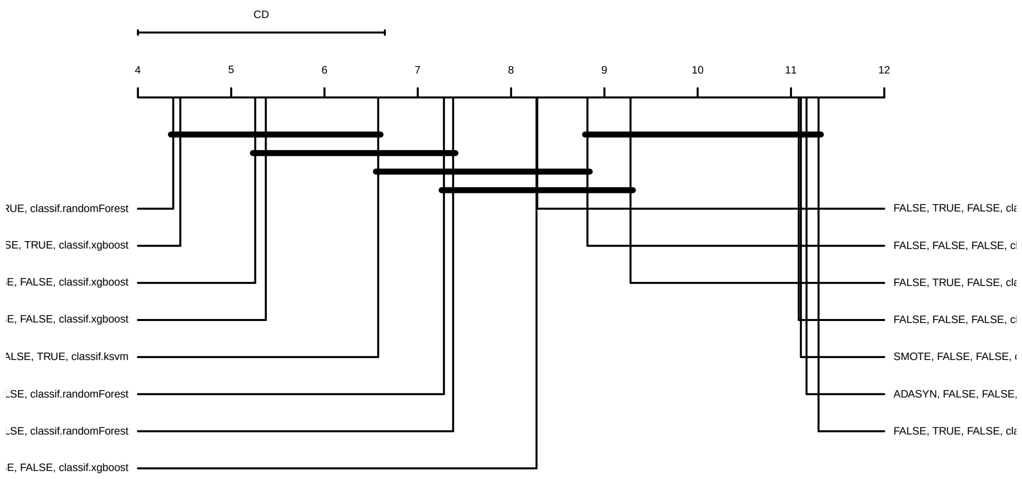
Plotando os ranks

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

```
##         ADASYN, FALSE, FALSE, classif.ksvm
##                                  11.166667
## ADASYN, FALSE, FALSE, classif.randomForest
##                                   7.378788
##      ADASYN, FALSE, FALSE, classif.xgboost
##                                   5.371212
##          FALSE, FALSE, FALSE, classif.ksvm
##                                  11.083333
##  FALSE, FALSE, FALSE, classif.randomForest
##                                   8.818182
##       FALSE, FALSE, FALSE, classif.xgboost
##                                   8.272727
##           FALSE, FALSE, TRUE, classif.ksvm
##                                   6.575758
##   FALSE, FALSE, TRUE, classif.randomForest
##                                   4.378788
##        FALSE, FALSE, TRUE, classif.xgboost
##                                   4.454545
##           FALSE, TRUE, FALSE, classif.ksvm
##                                  11.295455
##   FALSE, TRUE, FALSE, classif.randomForest
##                                   9.280303
##        FALSE, TRUE, FALSE, classif.xgboost
##                                   8.280303
##          SMOTE, FALSE, FALSE, classif.ksvm
##                                  11.106061
##  SMOTE, FALSE, FALSE, classif.randomForest
##                                   7.280303
##       SMOTE, FALSE, FALSE, classif.xgboost
##                                   5.257576
```

Plotando grafico de Critical Diference

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


CD
RUE, classif.randomForest
SE, TRUE, classif.xgboost
E, FALSE, classif.xgboost
E, FALSE, classif.xgboost
ALSE, TRUE, classif.ksvm
LSE, classif.randomForest
LSE, classif.randomForest
E, FALSE, classif.xgboost
FALSE, TRUE, FALSE, cla
FALSE, FALSE, FALSE, c
FALSE, TRUE, FALSE, cla
FALSE, FALSE, FALSE, c
SMOTE, FALSE, FALSE,
ADASYN, FALSE, FALSE,
FALSE, TRUE, FALSE, cla