R Notebook

Parametros:

Measure = Area under the curve
Columns = sampling, weight_space, underbagging, learner
Performance = holdout_measure_residual
Filter keys = imba.rate
Filter values = 0.01

```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

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

Filtrando o data set

```r
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            :1800   FALSE :1080   FALSE:1440
##  F1 measure                       :   0   SMOTE : 360   TRUE :360
##  G-mean                           :   0                 NA's :0
##  Matthews correlation coefficient:   0
##
##
##  tuning_measure   holdout_measure  holdout_measure_residual
##  Min.   :0.3866   Min.   :0.2139   Min.   :0.3092
##  1st Qu.:0.9529   1st Qu.:0.8909   1st Qu.:0.7392
##  Median :0.9993   Median :0.9916   Median :0.9067
##  Mean   :0.9498   Mean   :0.9120   Mean   :0.8469
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.9842
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  NA's   :51       NA's   :51       NA's   :51
##  iteration_count                      dataset       imba.rate
##  Min.   :1       abalone                 :  45   Min.   :0.01
##  1st Qu.:1       adult                   :  45   1st Qu.:0.01
##  Median :2       bank                    :  45   Median :0.01
##  Mean   :2       car                     :  45   Mean   :0.01
##  3rd Qu.:3       cardiotocography-10clases:  45   3rd Qu.:0.01
##  Max.   :3       cardiotocography-3clases :  45   Max.   :0.01
##  NA's   :51      (Other)                  :1530
```

Computando as médias das iteracoes

```r
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] 40 15
```

```
# Renomeando a variavel
df = df_tec_wide_residual

summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.4882
##  1st Qu.:0.7250
##  Median :0.8243
##  Mean   :0.8102
##  3rd Qu.:0.9394
##  Max.   :1.0000
##  NA's   :3
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.6035
##  1st Qu.:0.8480
##  Median :0.9435
##  Mean   :0.8922
##  3rd Qu.:0.9805
##  Max.   :1.0000
##  NA's   :6
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.5704                        Min.   :0.4549
##  1st Qu.:0.7888                        1st Qu.:0.7339
##  Median :0.9261                        Median :0.8486
##  Mean   :0.8619                        Mean   :0.8197
##  3rd Qu.:0.9854                        3rd Qu.:0.9742
##  Max.   :1.0000                        Max.   :1.0000
##
```

```
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.4953
##  1st Qu.:0.8222
##  Median :0.9411
##  Mean   :0.8726
##  3rd Qu.:0.9808
##  Max.   :1.0000
##  NA's   :1
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.5704                       Min.   :0.5009
##  1st Qu.:0.8118                       1st Qu.:0.6258
##  Median :0.9225                       Median :0.7851
##  Mean   :0.8761                       Mean   :0.7641
##  3rd Qu.:0.9855                       3rd Qu.:0.8949
##  Max.   :1.0000                       Max.   :0.9986
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.5415
##  1st Qu.:0.7837
##  Median :0.9200
##  Mean   :0.8658
##  3rd Qu.:0.9849
##  Max.   :1.0000
##  NA's   :1
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.5153                      Min.   :0.4988
##  1st Qu.:0.7695                      1st Qu.:0.7339
##  Median :0.9026                      Median :0.8471
##  Mean   :0.8594                      Mean   :0.8207
##  3rd Qu.:0.9755                      3rd Qu.:0.9742
##  Max.   :1.0000                      Max.   :1.0000
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.5008
##  1st Qu.:0.7928
##  Median :0.9506
##  Mean   :0.8801
##  3rd Qu.:0.9932
##  Max.   :1.0000
##  NA's   :2
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.5437                      Min.   :0.4858
##  1st Qu.:0.7935                      1st Qu.:0.6569
##  Median :0.9184                      Median :0.8160
##  Mean   :0.8725                      Mean   :0.7930
##  3rd Qu.:0.9794                      3rd Qu.:0.9451
##  Max.   :1.0000                      Max.   :1.0000
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.5061
##  1st Qu.:0.7645
##  Median :0.9316
##  Mean   :0.8627
##  3rd Qu.:0.9883
##  Max.   :1.0000
##  NA's   :4
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.5112
##  1st Qu.:0.7790
##  Median :0.9260
##  Mean   :0.8617
##  3rd Qu.:0.9868
##  Max.   :1.0000
##
```

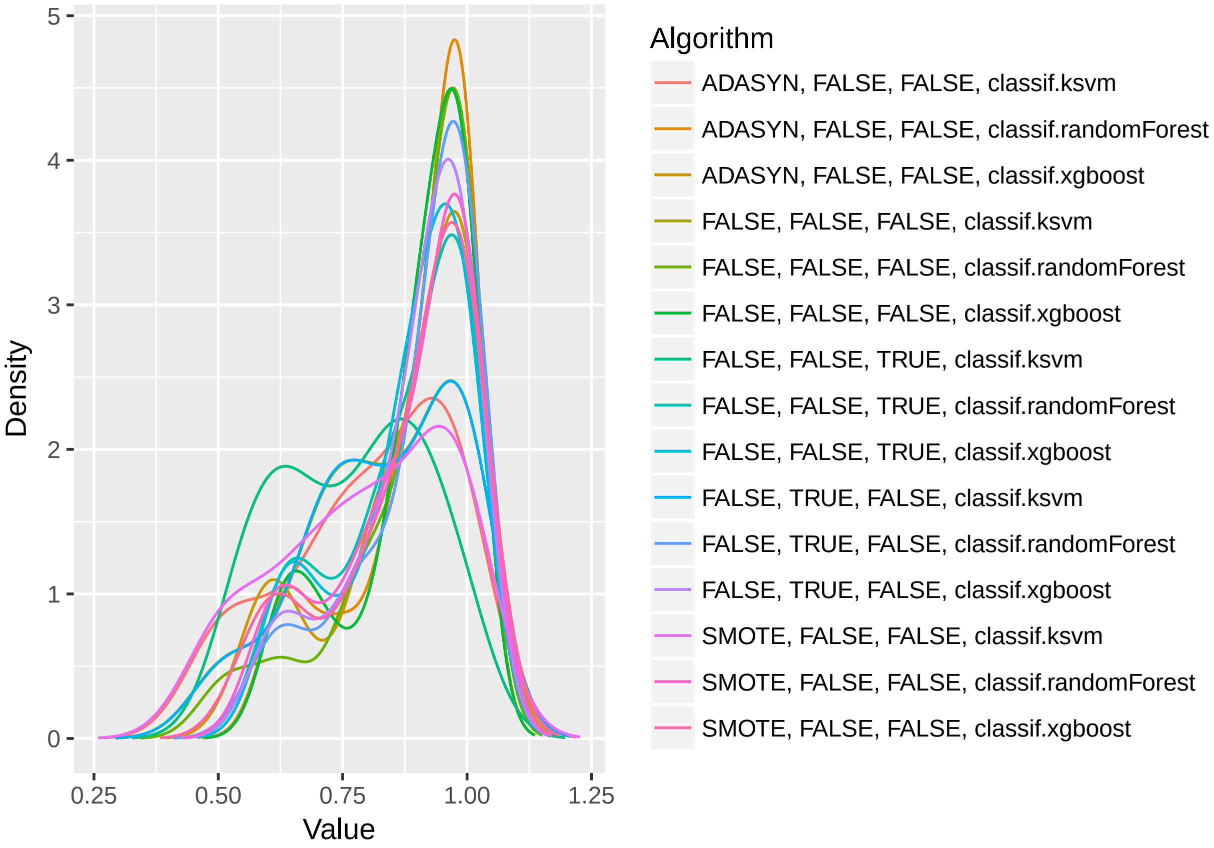
Verificando a média de cada coluna selecionada

```
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 = 0.810192517620008"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.892226457440384"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.861931463889398"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.819684267348074"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.872551115921721"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.876095504804726"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.764052695360382"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.865787028371761"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.859448061063698"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.820687355828643"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.880062700886546"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.872496664911349"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.792984752389399"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.862701109173019"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.861650382410199"
```

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 = 87.312, df = 14, p-value = 1.219e-12
```

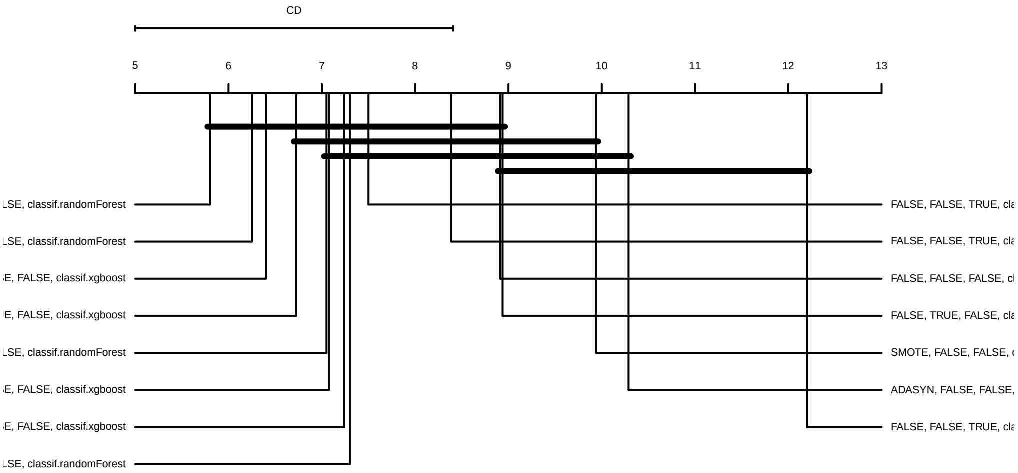
Testando as diferencas par a par

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

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

Plotando grafico de Critical Diference

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


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