R Notebook

Parametros:

Measure = F1 measure
Columns = sampling, weight_space, underbagging, learner
Performance = tuning_measure
Filter keys = NULL
Filter values = NULL

```
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)){
  ds = filter_at(ds, .vars = params$filter_keys, .vars_predicate = any_vars(. == params$filter_values))
}

summary(ds)
```

```
##                 learner     weight_space
##  classif.ksvm        :3420   Mode :logical
##  classif.randomForest:3420   FALSE:8208
##  classif.rusboost    :   0   TRUE :2052
##  classif.xgboost     :3420   NA's :0
##
##
##
##                             measure        sampling    underbagging
##  Accuracy                       :    0   ADASYN:2052   Mode :logical
##  Area under the curve           :    0   FALSE :6156   FALSE:8208
##  F1 measure                     :10260   SMOTE :2052   TRUE :2052
##  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.:0.2739   1st Qu.:0.0000   1st Qu.:0.04287
##  Median :0.8197   Median :0.4500   Median :0.28466
##  Mean   :0.6468   Mean   :0.4554   Mean   :0.36600
##  3rd Qu.:0.9944   3rd Qu.:0.8075   3rd Qu.:0.68235
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##  NA's   :216      NA's   :216      NA's   :216
##  iteration_count                      dataset       imba.rate
##  Min.   :1       abalone                  : 180   Min.   :0.0010
##  1st Qu.:1       adult                    : 180   1st Qu.:0.0100
##  Median :2       bank                     : 180   Median :0.0300
##  Mean   :2       car                      : 180   Mean   :0.0286
##  3rd Qu.:3       cardiotocography-10clases: 180   3rd Qu.:0.0500
##  Max.   :3       cardiotocography-3clases : 180   Max.   :0.0500
##  NA's   :216     (Other)                  :9180
```

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] 228  15
```

```
# Renomeando a variavel
df = df_tec_wide_residual

summary(df)
```

```
##  ADASYN, FALSE, FALSE, classif.ksvm
##  Min.   :0.7913
##  1st Qu.:0.9770
##  Median :0.9948
##  Mean   :0.9793
##  3rd Qu.:0.9987
##  Max.   :1.0000
##  NA's   :7
##  ADASYN, FALSE, FALSE, classif.randomForest
##  Min.   :0.7179
##  1st Qu.:0.9863
##  Median :0.9967
##  Mean   :0.9864
##  3rd Qu.:0.9992
##  Max.   :1.0000
##  NA's   :27
##  ADASYN, FALSE, FALSE, classif.xgboost FALSE, FALSE, FALSE, classif.ksvm
##  Min.   :0.6463                        Min.   :0.0000
##  1st Qu.:0.9841                        1st Qu.:0.0000
##  Median :0.9941                        Median :0.2032
##  Mean   :0.9808                        Mean   :0.2988
##  3rd Qu.:0.9987                        3rd Qu.:0.5279
##  Max.   :1.0000                        Max.   :1.0000
##
```

```
##  FALSE, FALSE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.1517
##  Median :0.4848
##  Mean   :0.5099
##  3rd Qu.:0.8632
##  Max.   :1.0000
##  NA's   :5
##  FALSE, FALSE, FALSE, classif.xgboost FALSE, FALSE, TRUE, classif.ksvm
##  Min.   :0.0000                       Min.   :0.02691
##  1st Qu.:0.2234                       1st Qu.:0.12822
##  Median :0.6119                       Median :0.34859
##  Mean   :0.5547                       Mean   :0.39206
##  3rd Qu.:0.8545                       3rd Qu.:0.61889
##  Max.   :1.0000                       Max.   :1.00000
##
##  FALSE, FALSE, TRUE, classif.randomForest
##  Min.   :0.02829
##  1st Qu.:0.15674
##  Median :0.31751
##  Mean   :0.39558
##  3rd Qu.:0.60451
##  Max.   :1.00000
##  NA's   :6
##  FALSE, FALSE, TRUE, classif.xgboost FALSE, TRUE, FALSE, classif.ksvm
##  Min.   :0.02172                     Min.   :0.0000
##  1st Qu.:0.14244                     1st Qu.:0.0000
##  Median :0.28592                     Median :0.1880
##  Mean   :0.36171                     Mean   :0.2919
##  3rd Qu.:0.54209                     3rd Qu.:0.5018
##  Max.   :1.00000                     Max.   :1.0000
##
##  FALSE, TRUE, FALSE, classif.randomForest
##  Min.   :0.0000
##  1st Qu.:0.1667
##  Median :0.4685
##  Mean   :0.5134
##  3rd Qu.:0.8749
##  Max.   :1.0000
##  NA's   :7
##  FALSE, TRUE, FALSE, classif.xgboost SMOTE, FALSE, FALSE, classif.ksvm
##  Min.   :0.0000                      Min.   :0.7607
##  1st Qu.:0.2283                      1st Qu.:0.9775
##  Median :0.6056                      Median :0.9952
##  Mean   :0.5531                      Mean   :0.9785
##  3rd Qu.:0.8559                      3rd Qu.:0.9994
##  Max.   :1.0000                      Max.   :1.0000
##
##  SMOTE, FALSE, FALSE, classif.randomForest
##  Min.   :0.7311
##  1st Qu.:0.9867
##  Median :0.9971
##  Mean   :0.9870
##  3rd Qu.:0.9994
```

```
##  Max.   :1.0000
##  NA's   :20
##  SMOTE, FALSE, FALSE, classif.xgboost
##  Min.   :0.6908
##  1st Qu.:0.9851
##  Median :0.9949
##  Mean   :0.9859
##  3rd Qu.:0.9987
##  Max.   :1.0000
##
```

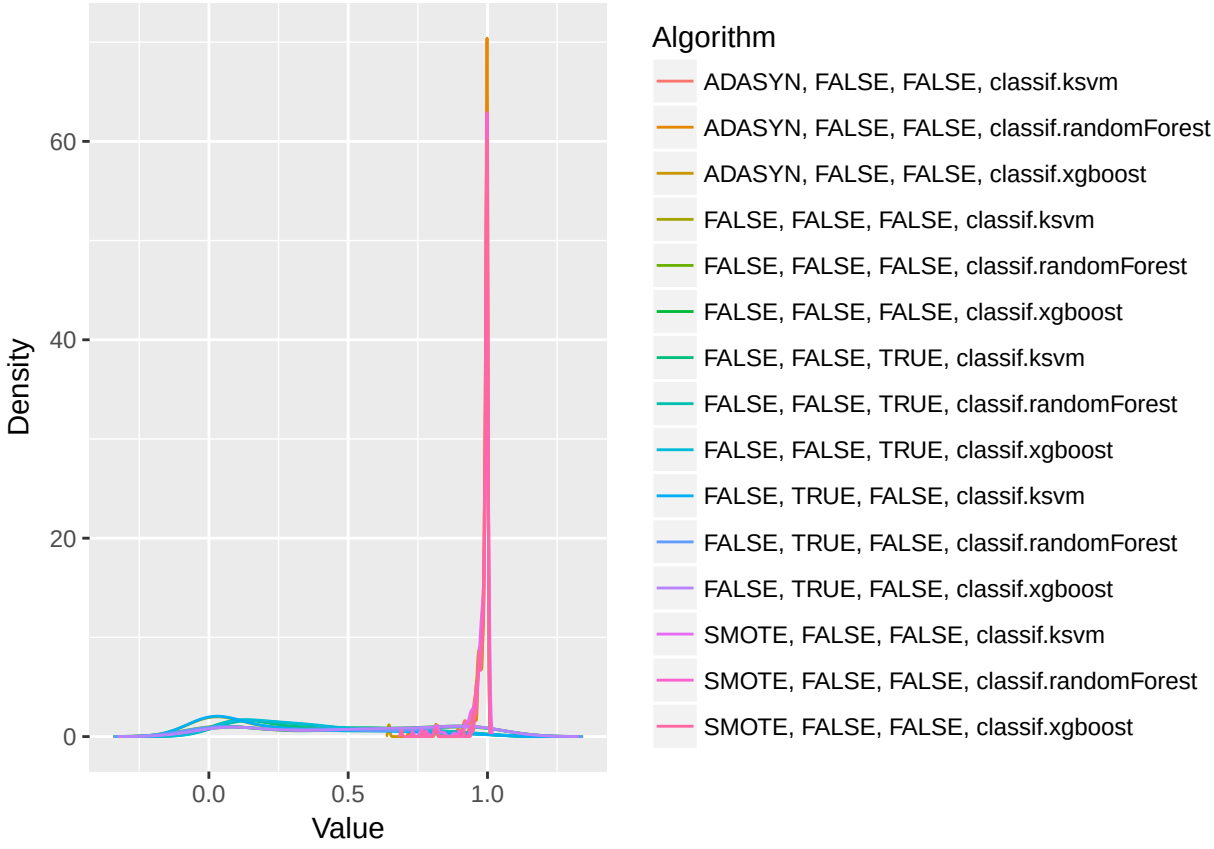
Verificando a média de cada coluna selecionada

```
for(i in (1:dim(df)[2])){
  #print(df[,i])
  print(paste("Media da coluna ", colnames(df)[i], " = ", mean(df[,i], na.rm = TRUE), sep=""))
}
```

```
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.ksvm = 0.979283131965443"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.randomForest = 0.986425350724291"
## [1] "Media da coluna ADASYN, FALSE, FALSE, classif.xgboost = 0.980773021311217"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.ksvm = 0.298822286155046"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.randomForest = 0.509855108081001"
## [1] "Media da coluna FALSE, FALSE, FALSE, classif.xgboost = 0.554723922554986"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.ksvm = 0.39206162711166"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.randomForest = 0.395581456764756"
## [1] "Media da coluna FALSE, FALSE, TRUE, classif.xgboost = 0.361712319826771"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.ksvm = 0.291938977502276"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.randomForest = 0.513352739370822"
## [1] "Media da coluna FALSE, TRUE, FALSE, classif.xgboost = 0.553101496540203"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.ksvm = 0.978508419002339"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.randomForest = 0.986950934161897"
## [1] "Media da coluna SMOTE, FALSE, FALSE, classif.xgboost = 0.985868886576608"
```

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 = 1939, 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,]                              FALSE
##  [3,]                              FALSE
##  [4,]                               TRUE
##  [5,]                               TRUE
##  [6,]                               TRUE
##  [7,]                               TRUE
```

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

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

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

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

Plotando grafico de Critical Diference

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