```
# Corrida general del Workflow Baseline

# limpio la memoria
rm(list = ls(all.names = TRUE)) # remove all objects
gc(full = TRUE, verbose= FALSE) # garbage collection

require("rlang")
require("yaml")
require("data.table")

if( !exists("envg") ) envg <- env()  # global environment

envg$EXPENV <- list()
envg$EXPENV$bucket_dir <- "~/buckets/b1"
envg$EXPENV$exp_dir <- "~/buckets/b1/expw/"
envg$EXPENV$wf_dir <- "~/buckets/b1/flow/"
envg$EXPENV$repo_dir <- "~/dmeyf2024/"
envg$EXPENV$datasets_dir <- "~/buckets/b1/datasets/"
envg$EXPENV$messenger <- "~/install/zulip_enviar.sh"

# lugar para alternar semillas 799891, 799921, 799961, 799991, 800011
envg$EXPENV$semilla_primigenia <- 799991

# leo el unico parametro del script
args <- commandArgs(trailingOnly=TRUE)
envg$EXPENV$scriptname <- args[1]

#------------------------------------------------------------------------------
# Error catching

options(error = function() {
  traceback(20)
  options(error = NULL)

  cat(format(Sys.time(), "%Y%m%d %H%M%S"), "\n",
    file = "z-Rabort.txt",
    append = TRUE
    )

  stop("exiting after script error")
})
#------------------------------------------------------------------------------
# inicializaciones varias

dir.create( envg$EXPENV$exp_dir, showWarnings = FALSE)
dir.create( envg$EXPENV$wf_dir, showWarnings = FALSE)
#------------------------------------------------------------------------------

# cargo las  "librerias" mlog y exp_lib

mlog <- paste0( envg$EXPENV$repo_dir,"/src/lib/mlog.r")
source( mlog )

exp_lib <- paste0( envg$EXPENV$repo_dir,"/src/lib/exp_lib.r")
source( exp_lib )

#------------------------------------------------------------------------------
# Incorporacion Dataset
# deterministico, SIN random
```

```
DT_incorporar_dataset <- function( arch_dataset )
{
  if( -1 == (param_local <- exp_init())$resultado ) return( 0 ) # linea fija

  param_local$meta$script <- "/src/wf-etapas/z1101_DT_incorporar_dataset.r"

  param_local$archivo_absoluto <- arch_dataset
  param_local$primarykey <- c("numero_de_cliente", "foto_mes" )
  param_local$entity_id <- c("numero_de_cliente" )
  param_local$periodo <- c("foto_mes" )
  param_local$clase <- c("clase_ternaria" )

  param_local$semilla <- NULL  # no usa semilla, es deterministico

  return( exp_correr_script( param_local ) ) # linea fija}
}
#------------------------------------------------------------------------------
# Catastophe Analysis  Baseline
# deterministico, SIN random
# MachineLearning EstadisticaClasica Ninguno

CA_catastrophe_base <- function( pinputexps, metodo )
{
  if( -1 == (param_local <- exp_init())$resultado ) return( 0 ) # linea fija

  param_local$meta$script <- "/src/wf-etapas/z1201_CA_reparar_dataset.r"

  # Opciones MachineLearning EstadisticaClasica Ninguno MICE
  param_local$metodo <- metodo
  param_local$semilla <- NULL  # no usa semilla, es deterministico

  return( exp_correr_script( param_local ) ) # linea fija}
}
#------------------------------------------------------------------------------
# Feature Engineering Intra Mes   Baseline
# deterministico, SIN random

FEintra_manual_base <- function( pinputexps )
{
  if( -1 == (param_local <- exp_init())$resultado ) return( 0 ) # linea fija


  param_local$meta$script <- "/src/wf-etapas/z1301_FE_intrames_manual.r"

  param_local$semilla <- NULL  # no usa semilla, es deterministico

  return( exp_correr_script( param_local ) ) # linea fija
}
#------------------------------------------------------------------------------
# Data Drifting Baseline
# deterministico, SIN random

DR_drifting_base <- function( pinputexps, metodo)
{
  if( -1 == (param_local <- exp_init())$resultado ) return( 0 ) # linea fija


  param_local$meta$script <- "/src/wf-etapas/z1401_DR_corregir_drifting.r"

  # valores posibles
  #  "ninguno", "rank_simple", "rank_cero_fijo", "deflacion", "estandarizar"
```

```
  param_local$metodo <- metodo
  param_local$semilla <- NULL  # no usa semilla, es deterministico

  return( exp_correr_script( param_local ) ) # linea fija
}
#------------------------------------------------------------------------------
# Feature Engineering Historico  Baseline
# deterministico, SIN random

FEhist_base <- function( pinputexps)
{
  if( -1 == (param_local <- exp_init())$resultado ) return( 0 ) # linea fija


  param_local$meta$script <- "/src/wf-etapas/z1501_FE_historia.r"

  param_local$lag1 <- TRUE
  param_local$lag2 <- FALSE # no me engraso con los lags de orden 2
  param_local$lag3 <- FALSE # no me engraso con los lags de orden 3

  # no me engraso las manos con las tendencias
  param_local$Tendencias1$run <- TRUE  # FALSE, no corre nada de lo que sigue
  param_local$Tendencias1$ventana <- 6
  param_local$Tendencias1$tendencia <- TRUE
  param_local$Tendencias1$minimo <- FALSE
  param_local$Tendencias1$maximo <- FALSE
  param_local$Tendencias1$promedio <- FALSE
  param_local$Tendencias1$ratioavg <- FALSE
  param_local$Tendencias1$ratiomax <- FALSE

  # no me engraso las manos con las tendencias de segundo orden
  param_local$Tendencias2$run <- FALSE
  param_local$Tendencias2$ventana <- 12
  param_local$Tendencias2$tendencia <- FALSE
  param_local$Tendencias2$minimo <- FALSE
  param_local$Tendencias2$maximo <- FALSE
  param_local$Tendencias2$promedio <- FALSE
  param_local$Tendencias2$ratioavg <- FALSE
  param_local$Tendencias2$ratiomax <- FALSE

  param_local$semilla <- NULL # no usa semilla, es deterministico

  return( exp_correr_script( param_local ) ) # linea fija
}
#------------------------------------------------------------------------------
#  Agregado de variables de Random Forest, corrido desde LightGBM
#  atencion, parmetros para generar variables, NO para buen modelo
#  azaroso, utiliza semilla

FErf_attributes_base <- function( pinputexps,
  arbolitos,
  hojas_por_arbol,
  datos_por_hoja,
  mtry_ratio
)
{
  if( -1 == (param_local <- exp_init())$resultado ) return( 0 )# linea fija


  param_local$meta$script <- "/src/wf-etapas/z1311_FE_rfatributes.r"
```

```
  # Parametros de un LightGBM que se genera para estimar la column importance
  param_local$train$clase01_valor1 <- c( "BAJA+2", "BAJA+1")
  param_local$train$training <- c( 202101, 202102, 202103)

  # parametros para que LightGBM se comporte como Random Forest
  param_local$lgb_param <- list(
    # parametros que se pueden cambiar
    num_iterations = arbolitos,
    num_leaves  = hojas_por_arbol,
    min_data_in_leaf = datos_por_hoja,
    feature_fraction_bynode  = mtry_ratio,

    # para que LightGBM emule Random Forest
    boosting = "rf",
    bagging_fraction = ( 1.0 - 1.0/exp(1.0) ),
    bagging_freq = 1.0,
    feature_fraction = 1.0,

    # genericos de LightGBM
    max_bin = 31L,
    objective = "binary",
    first_metric_only = TRUE,
    boost_from_average = TRUE,
    feature_pre_filter = FALSE,
    force_row_wise = TRUE,
    verbosity = -100,
    max_depth = -1L,
    min_gain_to_split = 0.0,
    min_sum_hessian_in_leaf = 0.001,
    lambda_l1 = 0.0,
    lambda_l2 = 0.0,

    pos_bagging_fraction = 1.0,
    neg_bagging_fraction = 1.0,
    is_unbalance = FALSE,
    scale_pos_weight = 1.0,

    drop_rate = 0.1,
    max_drop = 50,
    skip_drop = 0.5,

    extra_trees = FALSE
  )


  return( exp_correr_script( param_local ) ) # linea fija
}
#------------------------------------------------------------------------------
# Canaritos Asesinos   Baseline
#  azaroso, utiliza semilla

CN_canaritos_asesinos_base <- function( pinputexps, ratio, desvio)
{
  if( -1 == (param_local <- exp_init())$resultado ) return( 0 )# linea fija


  param_local$meta$script <- "/src/wf-etapas/z1601_CN_canaritos_asesinos.r"

  # Parametros de un LightGBM que se genera para estimar la column importance
  param_local$train$clase01_valor1 <- c( "BAJA+2", "BAJA+1")
  param_local$train$positivos <- c( "BAJA+2")
```

```
  param_local$train$training <- c( 202101, 202102, 202103)
  param_local$train$validation <- c( 202105 )
  param_local$train$undersampling <- 0.1
  param_local$train$gan1 <- 273000
  param_local$train$gan0 <-  -7000


  # ratio varia de 0.0 a 2.0
  # desvio varia de -4.0 a 4.0
  param_local$CanaritosAsesinos$ratio <- ratio
  # desvios estandar de la media, para el cutoff
  param_local$CanaritosAsesinos$desvios <- desvio

  return( exp_correr_script( param_local ) ) # linea fija
}
#------------------------------------------------------------------------------
# Training Strategy  Baseline
#   y solo incluyo en el dataset al 20% de los CONTINUA
#  azaroso, utiliza semilla

TS_strategy_base8 <- function( pinputexps )
{
  if( -1 == (param_local <- exp_init())$resultado ) return( 0 )# linea fija

  param_local$meta$script <- "/src/wf-etapas/z2101_TS_training_strategy.r"


  param_local$future <- c(202108)

  param_local$final_train$undersampling <- 1.0
  param_local$final_train$clase_minoritaria <- c( "BAJA+1", "BAJA+2")
  param_local$final_train$training <- c(202106, 202105, 202104,
    202103, 202102, 202101)


  param_local$train$training <- c(202104, 202103, 202102,
    202101, 202012, 202011)
  param_local$train$validation <- c(202105)
  param_local$train$testing <- c(202106)

  # Atencion  0.2  de  undersampling de la clase mayoritaria,  los CONTINUA
  # 1.0 significa NO undersampling
  param_local$train$undersampling <- 0.2
  param_local$train$clase_minoritaria <- c( "BAJA+1", "BAJA+2")

  return( exp_correr_script( param_local ) ) # linea fija
}
#------------------------------------------------------------------------------
# Hyperparamteter Tuning Baseline
#  donde la Bayuesian Optimization solo considera 4 hiperparámetros
#  azaroso, utiliza semilla
#  puede llegar a recibir  bypass, que por default esta en false

HT_tuning_base <- function( pinputexps, bo_iteraciones, bypass=FALSE)
{
  if( -1 == (param_local <- exp_init(pbypass=bypass))$resultado ) return( 0 ) #
linea fija bypass

  param_local$meta$script <- "/src/wf-etapas/z2201_HT_lightgbm_gan.r"

  # En caso que se haga cross validation, se usa esta cantidad de folds
```

```
  param_local$lgb_crossvalidation_folds <- 5

  param_local$train$clase01_valor1 <- c( "BAJA+2", "BAJA+1")
  param_local$train$positivos <- c( "BAJA+2")
  param_local$train$gan1 <- 273000
  param_local$train$gan0 <-  -7000
  param_local$train$meseta <- 2001

  # Hiperparametros  del LightGBM
  #  los que tienen un solo valor son los que van fijos
  #  los que tienen un vector,  son los que participan de la Bayesian Optimization

  param_local$lgb_param <- list(
    boosting = "gbdt", # puede ir  dart  , ni pruebe random_forest
    objective = "binary",
    metric = "custom",
    first_metric_only = TRUE,
    boost_from_average = TRUE,
    feature_pre_filter = FALSE,
    force_row_wise = TRUE, # para reducir warnings
    verbosity = -100,
    max_depth = -1L, # -1 significa no limitar,  por ahora lo dejo fijo
    min_gain_to_split = 0.0, # min_gain_to_split >= 0.0
    min_sum_hessian_in_leaf = 0.001, #  min_sum_hessian_in_leaf >= 0.0
    lambda_l1 = 0.0, # lambda_l1 >= 0.0
    lambda_l2 = 0.0, # lambda_l2 >= 0.0
    max_bin = 31L, # lo debo dejar fijo, no participa de la BO
    num_iterations = 9999, # un numero muy grande, lo limita early_stopping_rounds

    bagging_fraction = 1.0, # 0.0 < bagging_fraction <= 1.0
    pos_bagging_fraction = 1.0, # 0.0 < pos_bagging_fraction <= 1.0
    neg_bagging_fraction = 1.0, # 0.0 < neg_bagging_fraction <= 1.0
    is_unbalance = FALSE, #
    scale_pos_weight = 1.0, # scale_pos_weight > 0.0

    drop_rate = 0.1, # 0.0 < neg_bagging_fraction <= 1.0
    max_drop = 50, # <=0 means no limit
    skip_drop = 0.5, # 0.0 <= skip_drop <= 1.0


    # # quantized me rompió
    # use_quantized_grad = TRUE, # enabling this will discretize (quantize) the
gradients and hessians into bins
    # num_grad_quant_bins =  4,
    # quant_train_renew_leaf = TRUE,

    extra_trees = FALSE,

    # Parte variable
    learning_rate = c( 0.02, 0.3 ),
    feature_fraction = c( 0.5, 0.9 ),
    num_leaves = c( 8L, 2048L,  "integer" ),
    min_data_in_leaf = c( 100L, 10000L, "integer" )


  )



  # iteraciones de la Optimizacion Bayesiana
  param_local$bo_iteraciones <- bo_iteraciones
```

```
  return( exp_correr_script( param_local ) ) # linea fija
}
#------------------------------------------------------------------------------
# proceso FM_final_models_base  Baseline
#  azaroso, utiliza semilla

FM_final_models_lightgbm <- function( pinputexps, ranks, qsemillas )
{
  if( -1 == (param_local <- exp_init())$resultado ) return( 0 )# linea fija

  param_local$meta$script <- "/src/wf-etapas/z2301_FM_final_models_lightgbm.r"

  # Que modelos quiero, segun su posicion en el ranking de la Bayesian
Optimizacion, ordenado por metrica descendente
  param_local$modelos_rank <- ranks
  param_local$metrica_order <- -1  # ordeno por el campo metrica en forma
DESCENDENTE

  # Que modelos quiero, segun su iteracion_bayesiana de la Bayesian Optimizacion,
SIN ordenar
  param_local$modelos_iteracion <- c()

  param_local$train$clase01_valor1 <- c( "BAJA+2", "BAJA+1")
  param_local$train$positivos <- c( "BAJA+2")

  # default 20 semillas
  param_local$qsemillas <- qsemillas

  return( exp_correr_script( param_local ) ) # linea fija
}
#------------------------------------------------------------------------------
# proceso ZZ_final  Baseline
# deterministico, SIN random

SC_scoring <- function( pinputexps )
{
  if( -1 == (param_local <- exp_init())$resultado ) return( 0 )# linea fija

  param_local$meta$script <- "/src/wf-etapas/z2401_SC_scoring_lightgbm.r"

  param_local$semilla <- NULL  # no usa semilla, es deterministico

  return( exp_correr_script( param_local ) ) # linea fija
}
#------------------------------------------------------------------------------
# proceso KA_evaluate_kaggle
# deterministico, SIN random

KA_evaluate_kaggle <- function( pinputexps )
{
  if( -1 == (param_local <- exp_init())$resultado ) return( 0 )# linea fija

  param_local$meta$script <- "/src/wf-etapas/z2601_KA_evaluate_kaggle.r"

  param_local$semilla <- NULL  # no usa semilla, es deterministico

  param_local$isems_submit <- 1:20 # misterioso parametro, no preguntar

  param_local$envios_desde <-   9000L
  param_local$envios_hasta <-  13000L
```

```
  param_local$envios_salto <-   500L
  param_local$competition <- "dm-ey-f-2024-segunda"

  return( exp_correr_script( param_local ) ) # linea fija
}
#------------------------------------------------------------------------------
#------------------------------------------------------------------------------
# A partir de ahora comienza la seccion de Workflows Completos
#------------------------------------------------------------------------------
# Este es el  Workflow Baseline
# Que predice 202108 donde NO conozco la clase

wf_base <- function( pnombrewf )
{
  param_local <- exp_wf_init( pnombrewf ) # linea workflow inicial fija

  # Etapa especificacion dataset de la Segunda Competencia Kaggle
  DT_incorporar_dataset( "~/buckets/b1/datasets/competencia_02.csv.gz")

  # Etapas preprocesamiento
  CA_catastrophe_base( metodo="MachineLearning")
  FEintra_manual_base()
  DR_drifting_base(metodo="rank_cero_fijo")
  FEhist_base()

  FErf_attributes_base( arbolitos= 20,
    hojas_por_arbol= 16,
    datos_por_hoja= 1000,
    mtry_ratio= 0.2
  )

  # CN_canaritos_asesinos_base(ratio=0.2, desvio=4.0)

  # ACÁ INSERTO CLUSTERS

  # Etapas modelado
  ts8 <- TS_strategy_base8()
  ht <- HT_tuning_base( bo_iteraciones = 40 )  # iteraciones inteligentes

  # Etapas finales
  fm <- FM_final_models_lightgbm( c(ht, ts8), ranks=c(1,2,3), qsemillas=5 )
  SC_scoring( c(fm, ts8) )
  KA_evaluate_kaggle()  # genera archivos para Kaggle

  return( exp_wf_end() ) # linea workflow final fija
}
#------------------------------------------------------------------------------
#------------------------------------------------------------------------------
# Aqui comienza el programa

# llamo al workflow con future = 202108
wf_base()
```