hackathon.R

alfrz

2022-09-30

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4  
## ✔ tibble 3.1.7 ✔ dplyr 1.0.9  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.0  
## ✔ readr 2.1.2 ✔ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 0.2.0 ──

## ✔ broom 1.0.0 ✔ rsample 1.0.0  
## ✔ dials 1.0.0 ✔ tune 0.2.0  
## ✔ infer 1.0.2 ✔ workflows 0.2.6  
## ✔ modeldata 1.0.0 ✔ workflowsets 0.2.1  
## ✔ parsnip 1.0.0 ✔ yardstick 1.0.0  
## ✔ recipes 1.0.0

## Warning: package 'broom' was built under R version 4.2.1

## Warning: package 'modeldata' was built under R version 4.2.1

## Warning: package 'recipes' was built under R version 4.2.1

## Warning: package 'rsample' was built under R version 4.2.1

## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## ✖ scales::discard() masks purrr::discard()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ recipes::fixed() masks stringr::fixed()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ yardstick::spec() masks readr::spec()  
## ✖ recipes::step() masks stats::step()  
## • Use tidymodels\_prefer() to resolve common conflicts.

library(themis)  
tidymodels\_prefer()  
  
# Data --------------------------------------------------------------------  
loc <- 'D:/data-mining-ta20222023'  
  
df\_train <- read.csv(file.path(loc, "training.csv"))  
df\_test <- read.csv(file.path(loc, "testing2.csv"), sep = ';')  
  
glimpse(df\_train)

## Rows: 18,890  
## Columns: 18  
## $ ID <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,…  
## $ X1 <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …  
## $ X2 <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ X3 <int> 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, …  
## $ X4 <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …  
## $ X5 <int> 50, 60, 40, 30, 30, 45, 45, 45, 45, 48, 29, 29, 60, 45, 30, 43, 43…  
## $ X6 <int> 22, 30, 19, 18, 18, 17, 17, 15, 15, 22, 12, 12, 27, 15, 13, 16, 14…  
## $ X7 <int> 2, 2, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ X8 <int> 23, 35, 20, 19, 19, 21, 21, 17, 17, 22, 14, 14, 28, 17, 13, 17, 15…  
## $ X9 <int> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10…  
## $ X10 <int> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10…  
## $ X11 <int> 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ X12 <int> 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, …  
## $ X13 <int> 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 1, 2, 2, 2, 1, …  
## $ X14 <int> 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, …  
## $ X15 <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, …  
## $ X16 <int> 1, 1, 1, 1, 1, 0, 0, 0, 0, 2, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, …  
## $ Y <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, …

glimpse(df\_test)

## Rows: 3,799  
## Columns: 17  
## $ X.ID <int> 1, 5, 9, 10, 11, 12, 14, 20, 21, 29, 31, 32, 33, 34, 43, 50, 60, …  
## $ X1 <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,…  
## $ X2 <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…  
## $ X3 <int> 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…  
## $ X4 <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1,…  
## $ X5 <int> 50, 30, 45, 48, 29, 29, 45, 35, 35, 48, 48, 49, 40, 40, 40, 16, 7…  
## $ X6 <int> 22, 18, 15, 22, 12, 12, 15, 14, 17, 24, 17, 14, 19, 19, 14, 14, 1…  
## $ X7 <int> 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…  
## $ X8 <int> 23, 19, 17, 22, 14, 14, 17, 16, 16, 24, 18, 15, 20, 20, 14, 15, 1…  
## $ X9 <int> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 0, 0, 10, 10, 0, …  
## $ X10 <int> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 1…  
## $ X11 <int> 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 2, 2, 1, 1, 2, 1, 1, 1, 1,…  
## $ X12 <int> 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 2, 1, 2,…  
## $ X13 <int> 1, 1, 2, 2, 2, 2, 1, 2, 2, 1, 1, 1, 1, 2, 2, 2, 1, 2, 1, 1, 1, 2,…  
## $ X14 <int> 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1,…  
## $ X15 <int> 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…  
## $ X16 <int> 1, 1, 0, 2, 1, 1, 0, 1, 0, 2, 2, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1,…

# Data Prepocessing -------------------------------------------------------  
unique(df\_train$Y)

## [1] 1 2

df\_train <- df\_train %>%   
 mutate(Y = as.factor(Y)) %>%   
 mutate\_at(vars(X1:X4, X7, X9:X16), ~ as.factor(.x))   
  
df\_test <- df\_test %>%   
 mutate\_at(vars(X1:X4, X7, X9:X16), ~ as.factor(.x)) %>%   
 rename(ID = X.ID)  
  
df\_train %>%   
 glimpse()

## Rows: 18,890  
## Columns: 18  
## $ ID <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,…  
## $ X1 <fct> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …  
## $ X2 <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ X3 <fct> 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, …  
## $ X4 <fct> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …  
## $ X5 <int> 50, 60, 40, 30, 30, 45, 45, 45, 45, 48, 29, 29, 60, 45, 30, 43, 43…  
## $ X6 <int> 22, 30, 19, 18, 18, 17, 17, 15, 15, 22, 12, 12, 27, 15, 13, 16, 14…  
## $ X7 <fct> 2, 2, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ X8 <int> 23, 35, 20, 19, 19, 21, 21, 17, 17, 22, 14, 14, 28, 17, 13, 17, 15…  
## $ X9 <fct> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10…  
## $ X10 <fct> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10…  
## $ X11 <fct> 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ X12 <fct> 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, …  
## $ X13 <fct> 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 1, 2, 2, 2, 1, …  
## $ X14 <fct> 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, …  
## $ X15 <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, …  
## $ X16 <fct> 1, 1, 1, 1, 1, 0, 0, 0, 0, 2, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, …  
## $ Y <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, …

# Metric ------------------------------------------------------------------  
my\_metric <- metric\_set(accuracy)  
  
  
# train test split --------------------------------------------------------  
set.seed(1)  
splits <- initial\_split(df\_train, prop = 0.7, strata = Y)  
splits

## <Training/Testing/Total>  
## <13222/5668/18890>

train\_set <- training(splits)  
test\_set <- testing(splits)  
  
set.seed(1)  
train\_fold <- vfold\_cv(train\_set, v = 3, strata = Y)  
  
  
  
# EDA ---------------------------------------------------------------------  
colSums(is.na(df\_train))

## ID X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 Y   
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

# model -------------------------------------------------------------------  
logreg\_spec <-  
 logistic\_reg(penalty = tune(), mixture = tune()) %>%  
 set\_engine('glmnet')  
  
rf\_spec <-  
 rand\_forest(mtry = tune(), min\_n = tune(), trees = tune()) %>%  
 set\_engine('ranger') %>%  
 set\_mode('classification')  
  
xgb\_spec <-  
 boost\_tree(tree\_depth = tune(), trees = tune(), learn\_rate = tune(), min\_n = tune(), loss\_reduction = tune(), sample\_size = tune(), stop\_iter = tune()) %>%  
 set\_engine('xgboost') %>%  
 set\_mode('classification')  
  
mlp\_spec <-  
 mlp(hidden\_units = tune(), penalty = tune(), epochs = tune()) %>%  
 set\_engine('nnet') %>%  
 set\_mode('classification')  
  
  
  
svm\_rbf\_spec <-  
 svm\_rbf(cost = tune(), rbf\_sigma = tune(), margin = tune()) %>%  
 set\_engine('kernlab') %>%  
 set\_mode('classification')  
  
svm\_poly\_spec <-  
 svm\_poly(cost = tune(), degree = tune(), scale\_factor = tune(), margin = tune()) %>%  
 set\_engine('kernlab') %>%  
 set\_mode('classification')  
  
# recipe and workflow -----------------------------------------------------  
my\_recipe <-   
 recipe(Y ~ ., data = train\_set) %>%   
 update\_role(ID, new\_role = "id") %>%   
 step\_dummy(all\_nominal\_predictors()) %>%   
 step\_zv() %>%   
 step\_smote(Y, over\_ratio = 0.9)   
  
  
my\_workflow <-   
 workflow\_set(  
 preproc = list(my\_recipe),   
 models = list(  
 logreg = logreg\_spec  
 # nnet = mlp\_spec,  
 # xgb = xgb\_spec,  
 # rf = rf\_spec  
 # svm\_rbf = svm\_rbf\_spec  
 # svm\_poly = svm\_poly\_spec  
 )  
 )  
  
  
# tuning ------------------------------------------------------------------  
grid\_ctrl <-  
 control\_grid(  
 # parallel\_over = "resampling",  
 save\_pred = TRUE,  
 verbose = T,  
 allow\_par = T  
 )  
  
  
library(doParallel)

## Loading required package: foreach

##   
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':  
##   
## accumulate, when

## Loading required package: iterators

## Loading required package: parallel

all\_cores <- parallel::detectCores(logical = FALSE)  
cl <- makePSOCKcluster(all\_cores)  
registerDoParallel(cl)  
  
grid\_results <-  
 my\_workflow %>%  
 workflow\_map(  
 seed = 1,  
 resamples = train\_fold,  
 grid = 50,  
 control = grid\_ctrl,  
 metrics = my\_metric,  
 verbose = TRUE  
 )

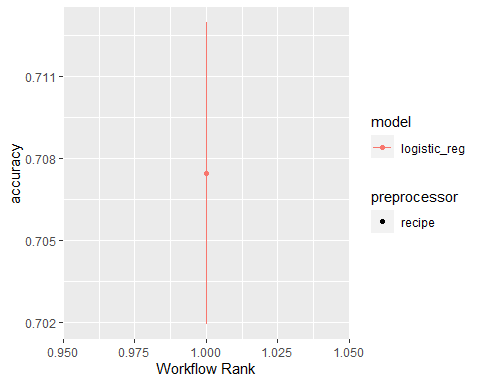
## i 1 of 1 tuning: recipe\_logreg

## ✔ 1 of 1 tuning: recipe\_logreg (1m 44.8s)

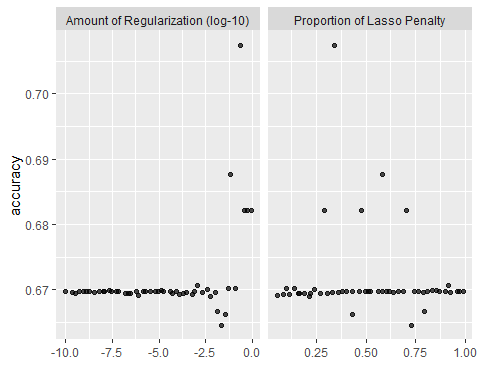
# res1 <- grid\_results  
# res2 <- grid\_results  
  
# Hasil -------------------------------------------------------------------  
metric\_name <- 'accuracy'  
  
grid\_results %>%   
 rank\_results() %>%   
 filter(.metric == metric\_name) %>%   
 select(model, .config, accuracy = mean, rank)

## # A tibble: 50 × 4  
## model .config accuracy rank  
## <chr> <chr> <dbl> <int>  
## 1 logistic\_reg Preprocessor1\_Model16 0.707 1  
## 2 logistic\_reg Preprocessor1\_Model28 0.688 2  
## 3 logistic\_reg Preprocessor1\_Model23 0.682 3  
## 4 logistic\_reg Preprocessor1\_Model13 0.682 4  
## 5 logistic\_reg Preprocessor1\_Model35 0.682 5  
## 6 logistic\_reg Preprocessor1\_Model46 0.671 6  
## 7 logistic\_reg Preprocessor1\_Model05 0.670 7  
## 8 logistic\_reg Preprocessor1\_Model03 0.670 8  
## 9 logistic\_reg Preprocessor1\_Model11 0.670 9  
## 10 logistic\_reg Preprocessor1\_Model42 0.670 10  
## # … with 40 more rows

autoplot(  
 grid\_results,  
 rank\_metric = metric\_name, # <- how to order models  
 metric = metric\_name, # <- which metric to visualize  
 select\_best = TRUE # <- one point per workflow  
)



autoplot(  
 grid\_results,  
 id = "recipe\_logreg",  
 metric = metric\_name  
)



# finalizing model --------------------------------------------------------  
best\_model <- 'recipe\_logreg'  
  
best\_results <-   
 grid\_results %>%   
 extract\_workflow\_set\_result(best\_model) %>%   
 select\_best(metric = metric\_name)  
  
best\_results

## # A tibble: 1 × 3  
## penalty mixture .config   
## <dbl> <dbl> <chr>   
## 1 0.226 0.343 Preprocessor1\_Model16

final\_wf <-   
 grid\_results %>%   
 extract\_workflow(best\_model) %>%   
 finalize\_workflow(best\_results)  
  
final\_wf

## ══ Workflow ════════════════════════════════════════════════════════════════════  
## Preprocessor: Recipe  
## Model: logistic\_reg()  
##   
## ── Preprocessor ────────────────────────────────────────────────────────────────  
## 3 Recipe Steps  
##   
## • step\_dummy()  
## • step\_zv()  
## • step\_smote()  
##   
## ── Model ───────────────────────────────────────────────────────────────────────  
## Logistic Regression Model Specification (classification)  
##   
## Main Arguments:  
## penalty = 0.225892019025795  
## mixture = 0.343290190990083  
##   
## Computational engine: glmnet

final\_fit <- final\_wf %>% fit(df\_train)   
final\_fit

## ══ Workflow [trained] ══════════════════════════════════════════════════════════  
## Preprocessor: Recipe  
## Model: logistic\_reg()  
##   
## ── Preprocessor ────────────────────────────────────────────────────────────────  
## 3 Recipe Steps  
##   
## • step\_dummy()  
## • step\_zv()  
## • step\_smote()  
##   
## ── Model ───────────────────────────────────────────────────────────────────────  
##   
## Call: glmnet::glmnet(x = maybe\_matrix(x), y = y, family = "binomial", alpha = ~0.343290190990083)   
##   
## Df %Dev Lambda  
## 1 0 0.00 0.32790  
## 2 1 0.36 0.29880  
## 3 3 0.94 0.27230  
## 4 3 1.73 0.24810  
## 5 3 2.46 0.22600  
## 6 3 3.14 0.20600  
## 7 3 3.75 0.18770  
## 8 3 4.31 0.17100  
## 9 4 4.85 0.15580  
## 10 4 5.40 0.14200  
## 11 5 5.90 0.12930  
## 12 6 6.39 0.11790  
## 13 7 6.86 0.10740  
## 14 7 7.29 0.09785  
## 15 8 7.71 0.08915  
## 16 8 8.08 0.08123  
## 17 8 8.41 0.07402  
## 18 8 8.71 0.06744  
## 19 8 8.96 0.06145  
## 20 9 9.20 0.05599  
## 21 10 9.41 0.05102  
## 22 11 9.65 0.04649  
## 23 11 9.88 0.04236  
## 24 11 10.07 0.03859  
## 25 11 10.24 0.03516  
## 26 12 10.39 0.03204  
## 27 13 10.52 0.02919  
## 28 13 10.64 0.02660  
## 29 13 10.74 0.02424  
## 30 14 10.83 0.02208  
## 31 16 10.93 0.02012  
## 32 16 11.03 0.01833  
## 33 17 11.12 0.01671  
## 34 17 11.20 0.01522  
## 35 17 11.27 0.01387  
## 36 17 11.34 0.01264  
## 37 18 11.39 0.01151  
## 38 18 11.44 0.01049  
## 39 18 11.48 0.00956  
## 40 18 11.52 0.00871  
## 41 18 11.55 0.00794  
## 42 18 11.58 0.00723  
## 43 19 11.60 0.00659  
## 44 19 11.62 0.00600  
## 45 19 11.64 0.00547  
## 46 19 11.65 0.00498  
##   
## ...  
## and 17 more lines.

# Evaluasi ----------------------------------------------------------------  
# train  
df\_train %>%   
 bind\_cols(predict(final\_fit, .)) %>%   
 my\_metric(truth = Y, estimate = .pred\_class)

## # A tibble: 1 × 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 accuracy binary 0.712

df\_train %>%   
 bind\_cols(predict(final\_fit, .)) %>%   
 conf\_mat(truth = Y, estimate = .pred\_class)

## Truth  
## Prediction 1 2  
## 1 11725 4281  
## 2 1160 1724

# test  
test\_set %>%   
 bind\_cols(predict(final\_fit, new\_data = .)) %>%   
 my\_metric(truth = Y, estimate = .pred\_class)

## # A tibble: 1 × 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 accuracy binary 0.719

test\_set %>%   
 bind\_cols(predict(final\_fit, .)) %>%   
 conf\_mat(truth = Y, estimate = .pred\_class)

## Truth  
## Prediction 1 2  
## 1 3548 1273  
## 2 318 529

# Prediction --------------------------------------------------------------  
preds <- predict(final\_fit, new\_data = df\_test) %>% pull()  
preds

## [1] 2 1 1 1 1 1 1 1 1 1 1 2 2 2 2 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1  
## [38] 1 1 1 1 1 1 2 2 2 1 1 1 1 2 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2  
## [75] 1 1 1 2 1 1 1 1 2 1 1 1 2 2 1 2 2 2 1 1 1 1 1 1 1 2 1 2 2 1 1 1 2 1 1 1 1  
## [112] 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 1 2 1  
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [186] 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 2 2 1 1 1 1 1 1 1  
## [223] 1 1 2 2 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 2 1 1 2 1 1 1 1  
## [260] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2  
## [297] 1 1 2 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 2 1 2 1 1 1 1 2 1 1 2  
## [334] 2 1 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1  
## [371] 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 2 2 2  
## [408] 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 2 1 1 1 1 1 1  
## [445] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 2 1 2 1 1 1 2 1 1 1 1 1  
## [482] 1 2 1 1 1 2 1 1 1 1 2 2 2 1 2 1 1 1 2 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 2  
## [519] 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 2 2 1 1 1 1 1 2 1 2 1 2 1 1 1 2 1 1  
## [556] 1 1 2 2 2 2 1 1 2 1 1 1 1 2 1 1 1 1 2 1 2 2 2 1 1 1 1 1 1 1 1 2 2 2 1 1 2  
## [593] 1 1 1 2 2 1 1 2 1 2 1 1 2 2 1 1 1 1 2 1 1 1 1 2 2 1 1 1 2 2 1 2 2 1 1 1 2  
## [630] 1 1 2 1 1 1 1 2 1 1 2 1 2 1 1 2 1 2 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1  
## [667] 2 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 2 2 1 2 1 1 1 1 1 2 2 1 1 2  
## [704] 1 1 1 2 2 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 2 1 2 1 1 1 1 1 1 2 1 1  
## [741] 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1  
## [778] 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1  
## [815] 1 2 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 2 1  
## [852] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2 2 1 1 1 1 2 1 1 1 1 2 2 1 1 1 1 2 1  
## [889] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 2 2 1 2 2 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1  
## [926] 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 1  
## [963] 2 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [1000] 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2  
## [1037] 2 1 1 1 2 1 1 1 1 1 1 2 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1  
## [1074] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1  
## [1111] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1  
## [1148] 1 1 2 1 2 2 2 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 2 1 1 2 2 1 1 1 1 1 1 2 1 2 1  
## [1185] 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1  
## [1222] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 1 1 1 1 1 1 2 1 1 1 1 2 2 2 1 1 1 1 1 1  
## [1259] 1 1 1 1 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1  
## [1296] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1  
## [1333] 1 1 1 2 2 2 2 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [1370] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1  
## [1407] 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 2 2 1 1 2 1 1 1 2 1 1 1 1 2 1 1 1 1 2 1 1 1  
## [1444] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1  
## [1481] 1 1 1 1 1 1 1 2 2 1 2 1 2 1 1 1 1 2 1 1 2 2 1 1 1 1 2 1 1 1 1 1 1 2 2 1 1  
## [1518] 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 2 2 2 1 2 1 1 1  
## [1555] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [1592] 1 1 1 1 2 2 1 1 1 1 2 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1  
## [1629] 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1  
## [1666] 1 1 1 1 1 2 1 2 2 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1  
## [1703] 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 2 2 2 1 1 1 2 1 1 1 1 2 1 1 2 2 1 1 1  
## [1740] 1 1 1 1 1 1 2 1 1 1 2 1 2 2 2 2 2 1 1 1 1 1 2 1 2 2 1 2 2 1 1 1 1 1 2 1 1  
## [1777] 1 1 2 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1  
## [1814] 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [1851] 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 1 1 2 1 2 1 2 2 1 2 1 1 1 1 2 1  
## [1888] 2 2 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 2 1 1 2 2 2 2 1  
## [1925] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1  
## [1962] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 2 1 2 1 1 1  
## [1999] 1 1 1 1 1 1 1 2 2 1 1 1 1 2 2 1 1 2 1 2 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1  
## [2036] 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1  
## [2073] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1  
## [2110] 2 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1  
## [2147] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1  
## [2184] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 2 2 1 2 1 1 1  
## [2221] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1  
## [2258] 1 1 2 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1  
## [2295] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1  
## [2332] 1 1 1 1 1 2 1 1 2 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1  
## [2369] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [2406] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [2443] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1  
## [2480] 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [2517] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1  
## [2554] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1  
## [2591] 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [2628] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [2665] 1 1 1 1 1 1 1 1 1 2 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [2702] 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1  
## [2739] 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 2 1 1 1 1  
## [2776] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [2813] 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [2850] 2 2 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 2  
## [2887] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [2924] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [2961] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1  
## [2998] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [3035] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1  
## [3072] 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 2 1 1 2 2 2 2 1 1 1 1  
## [3109] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1  
## [3146] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1  
## [3183] 1 1 1 1 1 1 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [3220] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [3257] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1  
## [3294] 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [3331] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 2 1 1 1 1  
## [3368] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [3405] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1  
## [3442] 1 1 1 1 1 1 2 1 1 2 2 2 2 1 1 1 1 2 2 2 2 1 1 2 1 2 1 2 1 1 2 2 2 1 2 1 1  
## [3479] 1 1 2 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 2 1 1 2 2 1 2 2 1 2 2 1 1 1 1 1  
## [3516] 2 2 2 1 1 1 2 2 2 2 2 1 1 1 1 1 2 2 2 2 2 1 1 2 2 1 1 1 2 2 2 1 2 2 1 1 1  
## [3553] 1 1 1 1 2 2 1 2 2 2 1 1 1 1 1 1 2 2 1 1 1 1 2 2 2 1 2 1 1 1 2 1 2 1 2 1 2  
## [3590] 1 1 1 1 1 2 1 2 2 1 2 2 2 1 1 2 2 1 1 2 2 1 2 2 1 1 2 1 1 1 1 1 1 2 1 1 1  
## [3627] 1 2 1 2 1 2 2 1 1 2 1 1 2 2 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 1 2 1 2 1 2 2 1  
## [3664] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1  
## [3701] 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 1 1  
## [3738] 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [3775] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2  
## Levels: 1 2

table(preds)

## preds  
## 1 2   
## 3213 586

# Submission --------------------------------------------------------------  
hasil <- df\_test %>%   
 select(ID) %>%   
 mutate(Y = preds)  
  
write.csv(hasil, 'E:/sub.csv', row.names = F, quote = F)   
  
# system(  
# 'kaggle competitions submit -c classification-data-challenge -f D:/\_\_Datasets/sub.csv -m "Message"'  
# )