final-hackathon.R

alfrz

2022-10-02

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()  
## • Dig deeper into tidy modeling with R at https://www.tmwr.org

library(themis)  
tidymodels\_prefer()  
  
# Data --------------------------------------------------------------------  
loc <- 'D:/Downloads/data-mining'  
list.files(path = loc)

## [1] "Submission\_Sample.csv" "testing.csv" "testing2.csv"   
## [4] "training.csv"

df\_train <- read.csv(file.path(loc, "training.csv"))  
df\_test <- read.csv(file.path(loc, "testing.csv"))  
  
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  
## $ ID <int> 1, 5, 9, 10, 11, 12, 14, 20, 21, 29, 31, 32, 33, 34, 43, 50, 60, 6…  
## $ 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, 70…  
## $ X6 <int> 22, 18, 15, 22, 12, 12, 15, 14, 17, 24, 17, 14, 19, 19, 14, 14, 16…  
## $ 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, 18…  
## $ X9 <int> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 0, 0, 10, 10, 0, 1…  
## $ 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, 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 -------------------------------------------------------  
table(df\_train$Y)

##   
## 1 2   
## 12885 6005

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))   
  
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, …

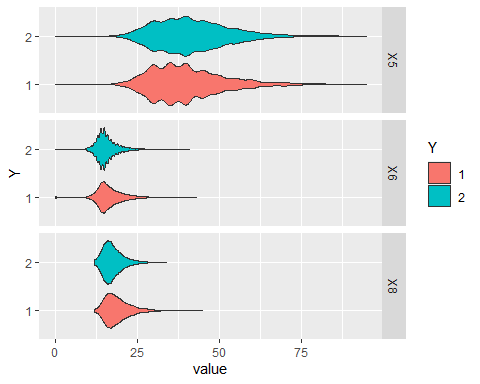
# EDA ---------------------------------------------------------------------  
colSums(is.na(df\_train))

## 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

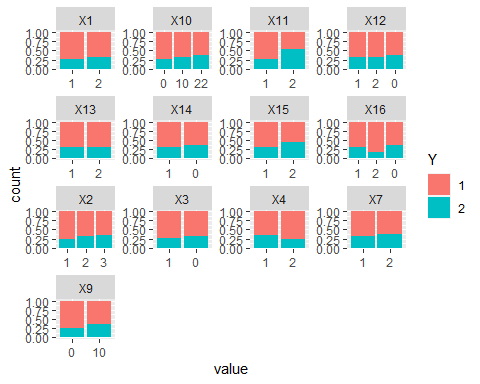
colSums(is.na(df\_test))

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

df\_train %>%   
 select(Y, X8, X5, X6) %>%   
 pivot\_longer(-Y) %>%   
 ggplot(aes(y = Y, x = value, fill = Y)) +  
 geom\_violin() +  
 facet\_grid(name~., scales = 'free')



df\_train %>%   
 select(Y, X1:X4, X7, X9:X16) %>%   
 pivot\_longer(-Y) %>%   
 ggplot(aes(x = value, fill = Y)) +  
 geom\_bar( position = position\_fill()) +  
 facet\_wrap(facets = ~name, scales = 'free', nrow = 4, ncol = 4)



# Metric ------------------------------------------------------------------  
my\_metric <- metric\_set(accuracy)  
  
  
# train test split --------------------------------------------------------  
set.seed(1)  
splits <- initial\_split(df\_train, prop = 0.8, strata = Y)  
splits

## <Training/Testing/Total>  
## <15112/3778/18890>

train\_set <- training(splits)  
test\_set <- testing(splits)  
  
set.seed(1)  
train\_fold <- vfold\_cv(df\_train, v = 3, strata = Y)  
  
  
# model -------------------------------------------------------------------  
rf\_spec <-  
 rand\_forest(mtry = tune(), min\_n = tune(), trees = tune()) %>%  
 set\_engine('ranger') %>%  
 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()   
  
  
my\_recipe %>%   
 summary() %>%   
 as.data.frame()

## variable type role source  
## 1 ID numeric id original  
## 2 X1 nominal predictor original  
## 3 X2 nominal predictor original  
## 4 X3 nominal predictor original  
## 5 X4 nominal predictor original  
## 6 X5 numeric predictor original  
## 7 X6 numeric predictor original  
## 8 X7 nominal predictor original  
## 9 X8 numeric predictor original  
## 10 X9 nominal predictor original  
## 11 X10 nominal predictor original  
## 12 X11 nominal predictor original  
## 13 X12 nominal predictor original  
## 14 X13 nominal predictor original  
## 15 X14 nominal predictor original  
## 16 X15 nominal predictor original  
## 17 X16 nominal predictor original  
## 18 Y nominal outcome original

my\_workflow <-   
 workflow() %>%   
 add\_model(rf\_spec) %>%   
 add\_recipe(my\_recipe)  
  
  
# Tuning ------------------------------------------------------------------  
my\_grid <- tibble(  
 mtry = c(11, 15),  
 min\_n = c(4, 2),  
 trees = c(700, 1100)  
)  
cross\_df(my\_grid)

## # A tibble: 8 × 3  
## mtry min\_n trees  
## <dbl> <dbl> <dbl>  
## 1 11 4 700  
## 2 15 4 700  
## 3 11 2 700  
## 4 15 2 700  
## 5 11 4 1100  
## 6 15 4 1100  
## 7 11 2 1100  
## 8 15 2 1100

library(doParallel)

## Loading required package: foreach

##   
## Attaching package: 'foreach'

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

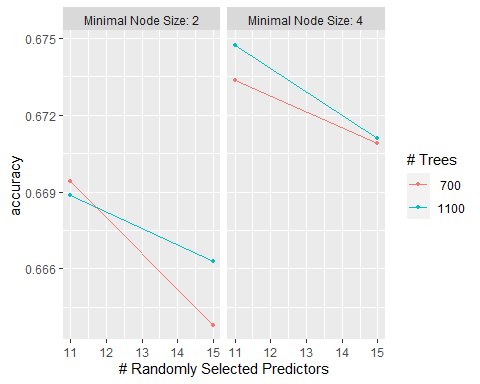
## Loading required package: iterators

## Loading required package: parallel

all\_cores <- parallel::detectCores(logical = FALSE)  
cl <- makePSOCKcluster(all\_cores)  
registerDoParallel(cl)  
  
# grid search  
set.seed(1)  
model\_res <-   
 my\_workflow %>%   
 tune\_grid(  
 resamples = train\_fold,  
 grid = cross\_df(my\_grid),  
 control = control\_grid(  
 save\_pred = TRUE,  
 verbose = T,  
 allow\_par = T  
 ),  
 metrics = my\_metric  
 )  
  
  
  
  
# hasil -------------------------------------------------------------------  
model\_res %>%   
 collect\_metrics() %>%   
 arrange(desc(mean))

## # A tibble: 8 × 9  
## mtry trees min\_n .metric .estimator mean n std\_err .config   
## <dbl> <dbl> <dbl> <chr> <chr> <dbl> <int> <dbl> <chr>   
## 1 11 1100 4 accuracy binary 0.675 3 0.00281 Preprocessor1\_Model5  
## 2 11 700 4 accuracy binary 0.673 3 0.00256 Preprocessor1\_Model1  
## 3 15 1100 4 accuracy binary 0.671 3 0.00213 Preprocessor1\_Model6  
## 4 15 700 4 accuracy binary 0.671 3 0.00132 Preprocessor1\_Model2  
## 5 11 700 2 accuracy binary 0.669 3 0.00247 Preprocessor1\_Model3  
## 6 11 1100 2 accuracy binary 0.669 3 0.00240 Preprocessor1\_Model7  
## 7 15 1100 2 accuracy binary 0.666 3 0.00147 Preprocessor1\_Model8  
## 8 15 700 2 accuracy binary 0.664 3 0.00157 Preprocessor1\_Model4

model\_res %>%   
 autoplot()



# Model Terbaik -----------------------------------------------------------  
best\_model <- model\_res %>%   
 select\_best()  
  
final\_wf <- my\_workflow %>%   
 finalize\_workflow(best\_model)  
  
final\_wf

## ══ Workflow ════════════════════════════════════════════════════════════════════  
## Preprocessor: Recipe  
## Model: rand\_forest()  
##   
## ── Preprocessor ────────────────────────────────────────────────────────────────  
## 1 Recipe Step  
##   
## • step\_zv()  
##   
## ── Model ───────────────────────────────────────────────────────────────────────  
## Random Forest Model Specification (classification)  
##   
## Main Arguments:  
## mtry = 11  
## trees = 1100  
## min\_n = 4  
##   
## Computational engine: ranger

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

## ══ Workflow [trained] ══════════════════════════════════════════════════════════  
## Preprocessor: Recipe  
## Model: rand\_forest()  
##   
## ── Preprocessor ────────────────────────────────────────────────────────────────  
## 1 Recipe Step  
##   
## • step\_zv()  
##   
## ── Model ───────────────────────────────────────────────────────────────────────  
## Ranger result  
##   
## Call:  
## ranger::ranger(x = maybe\_data\_frame(x), y = y, mtry = min\_cols(~11, x), num.trees = ~1100, min.node.size = min\_rows(~4, x), num.threads = 1, verbose = FALSE, seed = sample.int(10^5, 1), probability = TRUE)   
##   
## Type: Probability estimation   
## Number of trees: 1100   
## Sample size: 18890   
## Number of independent variables: 16   
## Mtry: 11   
## Target node size: 4   
## Variable importance mode: none   
## Splitrule: gini   
## OOB prediction error (Brier s.): 0.2210749

# Evaluasi ----------------------------------------------------------------  
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 0.921

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

## Truth  
## Prediction 1 2  
## 1 12489 1090  
## 2 396 4915

# 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 0.921

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

## Truth  
## Prediction 1 2  
## 1 2503 223  
## 2 74 978

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

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

# Submission --------------------------------------------------------------  
hasil <- df\_test %>%   
 select(ID) %>%   
 mutate(Y = preds)  
  
write.csv(hasil, 'D:/\_\_Datasets/sub.csv', row.names = F, quote = F)   
  
# system(  
# 'kaggle competitions submit -c data-mining-ta20222023 -f D:/\_\_Datasets/sub.csv -m "Message"'  
# )