**INSTACART PROJECT**

#LOADING THE LIBRARIES

library(dplyr)

library(data.table)

library(tidyr)

#LOADING THE DATA

path <- “/Users/keertannayak/Desktop/Keertan's Folder/Kaagle/InstaCart Dataset”

aisles <- fread(file.path(path,”aisles.csv”))

departments <- fread(file.path(path,”departments.csv”))

order\_prior <- fread(file.path(path,”order\_products\_\_prior”)

order\_train <- fread(file.path(path,”order\_products\_train”))

orders <- fread(file.path(path,”orders,csv”)

products <- fread(file.path(path,”products.csv”))

#DATA CONVERSION

aisles$aisle <- as.factor(aisles$aisle)

orders$eval\_set <- as.factor(orders$eval\_set)

products$product\_name <- as.factor(products$product\_name)

departments$department <- as.factor(departments$department)

products <- products %>%

  inner\_join(aisles) %>% inner\_join(departments) %>%

  select(-aisle\_id, -department\_id)

rm(aisles, departments)

order\_train$user\_id <- orders$user\_id[match(order\_train$order\_id, orders$order\_id)]

orders\_products <- orders %>% inner\_join(order\_prior, by = ”order\_id”)

rm(order\_prior)

#DEALING WITH THE PRODUCTS DATASET

prd <- orders\_products %>%

  arrange(user\_id, order\_number, product\_id) %>%

  group\_by(user\_id, product\_id) %>%

  mutate(product\_time = row\_number()) %>%

  ungroup() %>%

  group\_by(product\_id) %>%

  summarise(

    prod\_orders = n(),

    prod\_reorders = sum(reordered),

    prod\_first\_orders = sum(product\_time == 1),

    prod\_second\_orders = sum(product\_time == 2)

  )

prd$prod\_reorder\_probability <- prd$prod\_second\_orders / prd$prod\_first\_orders

prd$prod\_reorder\_times <- 1 + prd$prod\_reorders / prd$prod\_first\_orders

prd$prod\_reorder\_ratio <- prd$prod\_reorders / prd$prod\_orders

prd <- prd %>% select(-prod\_reorders, -prod\_first\_orders, -prod\_second\_orders)

rm(products)

# Users -------------------------------------------------------------------

users <- orders %>%

  filter(eval\_set == "prior") %>%

  group\_by(user\_id) %>%

  summarise(

    user\_orders = max(order\_number),

    user\_period = sum(days\_since\_prior\_order, na.rm = T),

    user\_mean\_days\_since\_prior = mean(days\_since\_prior\_order, na.rm = T)

  )

us <- orders\_products %>%

  group\_by(user\_id) %>%

  summarise(

    user\_total\_products = n(),

    user\_reorder\_ratio = sum(reordered == 1) / sum(order\_number > 1),

    user\_distinct\_products = n\_distinct(product\_id)

  )

users <- users %>% inner\_join(us)

users$user\_average\_basket <- users$user\_total\_products / users$user\_orders

us <- orders %>%

  filter(eval\_set != "prior") %>%

  select(user\_id, order\_id, eval\_set,

         time\_since\_last\_order = days\_since\_prior\_order)

users <- users %>% inner\_join(us)

rm(us)

#DEALING WITH THE ORDER\_PRODUCTS DATASET

data <- orders\_products %>%

  group\_by(user\_id, product\_id) %>%

  summarise(

    up\_orders = n(),

    up\_first\_order = min(order\_number),

    up\_last\_order = max(order\_number),

    up\_average\_cart\_position = mean(add\_to\_cart\_order))

rm(orders\_products, orders)

data <- data %>%

  inner\_join(prd, by = "product\_id") %>%

  inner\_join(users, by = "user\_id")

data$up\_order\_rate <- data$up\_orders / data$user\_orders

data$up\_orders\_since\_last\_order <- data$user\_orders - data$up\_last\_order

data$up\_order\_rate\_since\_first\_order <- data$up\_orders / (data$user\_orders - data$up\_first\_order + 1)

data <- data %>%

  left\_join(order\_tRAIN %>% select(user\_id, product\_id, reordered),

            by = c("user\_id", "product\_id"))

rm(order\_train, prd, users)

#DEALING WITH THE TRAIN & TEST DATASET

train <- as.data.frame(data[data$eval\_set == "train",])

train$eval\_set <- NULL

train$user\_id <- NULL

train$product\_id <- NULL

train$order\_id <- NULL

train$reordered[is.na(train$reordered)] <- 0

test <- as.data.frame(data[data$eval\_set == "test",])

test$eval\_set <- NULL

test$user\_id <- NULL

test$reordered <- NULL

rm(data)

#BUILDING THE MODEL

library(xgboost)

train\_2.0 <- train %>% sample\_frac(0.1)

X <- xgb.DMatrix(as.matrix(train\_2.0 %>% select(-reordered)), label = train\_2.0$reordered)

model <- xgboost(data = X, nrounds = 80)

importance <- xgb.importance(colnames(X), model = model)

xgb.ggplot.importance(importance)

rm(X, importance, train\_2.0)

#IMPLEMENTING THE MODEL ON TEST DATASET

X <- xgb.DMatrix(as.matrix(test %>% select(-order\_id, -product\_id)))

test$reordered <- predict(model, X)

test$reordered <- (test$reordered > 0.21) \* 1

submission <- test %>%

  filter(reordered == 1) %>%

  group\_by(order\_id) %>%

  summarise(

    products = paste(product\_id, collapse = " ")

  )

missing <- data.frame(

  order\_id = unique(test$order\_id[!test$order\_id %in% submission$order\_id]),

  products = "None"

)

submission <- submission %>% bind\_rows(missing) %>% arrange(order\_id)

write.csv(submission, file = "submit.csv", row.names = F)