instacart\_cross\_prod\_prep

# Libraries

# Read files

O =fread("data/orders.csv")  
O\_T=fread("data/order\_products\_\_train.csv")  
O\_P=fread("data/order\_products\_\_prior.csv")  
P = fread("data/products.csv")  
D = fread("data/departments.csv")  
A = fread("data/aisles.csv")

Total Orders - 32,14,874 Total customers - 2,06,209 = 7500(test) + 131209(train)

# Product file cleanup

gsub("Constant Comment","",P$product\_name) -> P$product\_name  
#grep("[\\]",P$product\_name)  
gsub('[\\]','',P$product\_name) -> P$product\_name  
#grep("\"\"",P$product\_name)  
gsub("\"\"","",P$product\_name) -> P$product\_name

# Building train/test user set

Find train user IDs to build train data;

# train user set   
train\_user <- as.matrix(unique(O[O$eval\_set == "train",]$user\_id))   
as.data.frame(train\_user) ->train\_user  
names(train\_user) = "user\_id"  
# test user set   
test\_user <- as.matrix(unique(O[O$eval\_set == "test",]$user\_id))   
as.data.frame(test\_user) ->test\_user  
names(test\_user) = "user\_id"

# sample it later

#test/train sampling  
#set.seed(1234)  
#idx = sample(1:nrow(train\_user),.7\*nrow(train\_user))  
#train\_user\_sample = train\_user[idx,] # 91846 records  
#test\_user\_sample = train\_user[-idx,] # 39363 records

# train\_orders # 15,24,629 orders # 8547

# test\_orders # 6,53,957 orders # 8128

# Extract complete order details;limit to 1000 users

#print(paste("starting query" , Sys.time()))  
  
train\_user %>%  
 head(100) %>%   
 left\_join(O, by = "user\_id") %>%   
 mutate(days\_since\_prior\_order = ifelse(is.na(days\_since\_prior\_order),  
 0,days\_since\_prior\_order)) %>%   
 mutate(cum\_purchase\_day =ave(days\_since\_prior\_order,user\_id,FUN=cumsum)) -> user\_orders   
#colnames(prior\_orders)[8] = "cum\_purchase\_day"  
  
# Pick up the last 10 orders   
user\_orders %>%  
 group\_by(user\_id) %>%  
 summarise(max\_order\_number = max(order\_number)) ->user\_max\_order\_number  
user\_orders %>%   
 left\_join(user\_max\_order\_number) ->user\_orders

## Joining, by = "user\_id"

user\_orders %>%  
 filter(order\_number + 10 >= max\_order\_number) ->user\_orders\_last10  
  
# Merge with order\_product\_prior dataset   
user\_orders\_last10 %>%   
 left\_join(O\_P, by ="order\_id") -> prior\_orders  
  
  
# Summarize at customer-product-level  
prior\_orders %>%   
 select(user\_id,product\_id,order\_id,add\_to\_cart\_order,days\_since\_prior\_order,  
 cum\_purchase\_day,order\_dow,order\_hour\_of\_day,  
 cum\_purchase\_day ) %>%  
 group\_by(user\_id,product\_id ) %>%   
 mutate(prodPurch\_lag = cum\_purchase\_day - lag(cum\_purchase\_day)) %>%  
 mutate(prodPurch\_lag = ifelse(is.na(prodPurch\_lag),0,prodPurch\_lag)) %>%  
 summarise(prodPurchCount=n(),  
 prodAddtocartOrder=round(median(add\_to\_cart\_order)),  
 prodPurch\_lag = median(prodPurch\_lag),  
# prodPurchDay = mode(puchase\_day),  
# prodPurchDOW = mode(order\_dow),  
# prodPurchHOD = mode(order\_hour\_of\_day),  
 prodLastPurch = max(cum\_purchase\_day)) -> Cust\_Prod\_Feature  
  
Cust\_Prod\_Feature %>%   
 mutate(prodNextPurch = prodLastPurch+ prodPurch\_lag) ->Cust\_Prod\_Feature

# Extract future order details

train\_user %>%   
 left\_join(O[O$eval\_set=="train",]) ->Cust\_future\_orders

## Joining, by = "user\_id"

Cust\_future\_orders %>%   
 left\_join(O\_T) ->Cust\_future\_orders

## Joining, by = "order\_id"

Cust\_future\_orders$reordered = Cust\_future\_orders$reordered +1   
paste0("FUT", colnames(Cust\_future\_orders)) -> colnames(Cust\_future\_orders)  
  
colnames(Cust\_future\_orders)[1] = "user\_id"  
colnames(Cust\_future\_orders)[8] = "product\_id"  
  
Rem\_cols = c("FUTeval\_set","FUTorder\_id")  
Cust\_future\_orders[,!colnames(Cust\_future\_orders) %in% Rem\_cols]-> Cust\_future\_orders

# \*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\* \*\*\* \*\*\*\*\*\*

# Reduce the size to 100 customers

train\_user[1:10,] %>% as.data.frame() ->users10  
colnames(users10) <- "user\_id"  
# Merge customer, product CROSS PRODUCT to the future orders(determine reorders or NOT)  
  
users10 %>%   
 left\_join(Cust\_future\_orders, all.x= TRUE) ->Cust\_future\_orders\_10

## Joining, by = "user\_id"

users10 %>%   
 merge(P,by=NULL) ->Cust\_Prod\_cross  
dim(P)

## [1] 49688 4

Cust\_Prod\_cross %>%   
left\_join(Cust\_future\_orders\_10, all.x= TRUE) ->Cust\_future\_orders\_10

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

Cust\_future\_orders\_10 %>%   
 mutate(FUTreordered = ifelse(is.na(FUTreordered),0,1))-> Cust\_Prod\_cross\_with\_Y  
  
Cust\_Prod\_cross\_with\_Y %>%   
 left\_join(Cust\_Prod\_Feature) ->Cust\_Prod\_cross\_with\_Y

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

#length(unique(Cust\_Prod\_cross\_with\_Y$user\_id))  
#Cust\_Prod\_cross\_with\_Y[is.na(Cust\_Prod\_cross\_with\_Y$user\_id),] %>% View()

# Extract purchase predictions(Y) for extracted users

write.csv(Cust\_Prod\_cross\_with\_Y , "Cust\_Prod\_cross\_with\_Y.csv")