instacart\_training\_dataset\_prep

# Libraries

# Mode Function

getmode <- function(v){  
 uv <- unique(v)  
 uv[which.max(tabulate(match(v,uv)))]}  
  
# Hours difference - hours run from 0 to 23  
#circ\_shift\_dow<- function(x) {   
# ifelse(x>3,x-4,x+3)}  
  
hrDiff <- function(a,b) {  
 diff = ifelse(a>b,a-b,b-a)   
 diff = min(diff,24-diff)   
 return(diff)}  
  
dowDiff <- function(a,b) {  
 diff = ifelse(a>b,a-b,b-a)   
 diff = min(diff,7-diff)   
 return(diff)}

# Read files

orders =fread("data/orders.csv")  
train\_orders =fread("data/order\_products\_\_train.csv")  
prior\_orders=fread("data/order\_products\_\_prior.csv")

# Calculate number of days using the days\_since\_prev\_order from each order

orders %>%   
 mutate(days\_since\_prior\_order = ifelse(is.na(days\_since\_prior\_order),0,  
 days\_since\_prior\_order)) %>%   
 mutate(O\_day = ave(days\_since\_prior\_order,user\_id,FUN = cumsum)) %>% setDT()->orders

# Building user set;

# train user set   
orders[orders$eval\_set == "train",c(2,4,8)] ->user\_set  
  
  
colnames(user\_set)[2] = "U\_total\_orders"  
colnames(user\_set)[3] = "U\_last\_O\_day"

# Extract all orders from user ; limit to last 100 orders if necessary

user\_set %>%  
 head(10) ->user\_set  
  
user\_set %>%  
 left\_join(orders[orders$eval\_set=="prior",], by = "user\_id") -> user\_orders  
user\_orders[,-5] ->user\_orders  
  
#user\_orders$order\_dt = user\_orders$start\_dt + days(user\_orders$O\_day)  
#user\_orders[,names(user\_orders) %in% c("user\_id","order\_id","order\_number", #"order\_dow","order\_hour\_of\_day","days\_since\_prior\_order","order\_dt")] -> user\_orders

# Build User\_Features

1. mostly ordered hour of the day  
 2. mostly ordered dow  
 3. average days between orders

user\_orders %>%   
 group\_by(user\_id) %>%   
 mutate(U\_mostly\_ord\_hod = getmode(order\_hour\_of\_day),  
 U\_mostly\_ord\_dow = getmode(order\_dow)) %>%  
 group\_by(order\_id) %>%   
 mutate(O\_ord\_dow\_abs\_var= dowDiff(order\_dow,U\_mostly\_ord\_dow),  
 O\_ord\_hod\_abs\_var= hrDiff(order\_hour\_of\_day,U\_mostly\_ord\_hod) ) %>%   
 group\_by(user\_id,U\_total\_orders,U\_mostly\_ord\_hod,U\_mostly\_ord\_dow) %>%   
 summarise(U\_mostly\_ord\_hod\_var = median(O\_ord\_hod\_abs\_var),  
 U\_mostly\_ord\_dow\_var = median(O\_ord\_dow\_abs\_var))->user\_feature  
  
# U\_dspo\_mean = round(mean(days\_since\_prior\_order)),  
# U\_dspo\_sd = round(sd(days\_since\_prior\_order)),  
# U\_days\_since\_last\_purchase = as.numeric(U\_last\_order\_day - O\_day))

# Add order details from prior\_order file

products = fread("data/products.csv")  
  
user\_orders %>%   
 left\_join(prior\_orders, by="order\_id") %>% setDT() -> user\_prior\_orders  
rm(prior\_orders)  
rm(user\_orders)  
  
user\_prior\_orders %>%   
 left\_join(products) -> user\_prior\_orders

## Joining, by = "product\_id"

# Limit the number of prior orders  
user\_prior\_orders %>%   
# filter(x-order\_dt > 180) %>%   
 group\_by(order\_id) %>%   
 mutate(O\_reord\_count = sum(reordered),  
 O\_cart\_size = n(),  
 O\_aisles = n\_distinct(aisle\_id),  
 O\_dept = n\_distinct(department\_id)) -> user\_prior\_orders  
  
user\_prior\_orders %>%   
 group\_by(order\_id,O\_cart\_size,O\_reord\_count,O\_aisles,O\_dept) %>%  
 mutate(O\_reord\_ratio\_in\_cart = round(O\_reord\_count/O\_cart\_size,2) ,  
 O\_aisle\_ratio\_in\_cart = round(O\_aisles / O\_cart\_size,2),  
 O\_dept\_ratio\_in\_cart = round(O\_dept/O\_cart\_size,2)) -> orders\_features

# More user features

orders\_features %>%  
 group\_by(user\_id) %>%   
 summarise(U\_avg\_reord\_ratio\_in\_cart = round(mean(O\_reord\_ratio\_in\_cart),2),  
 U\_avg\_aise\_ratio\_in\_cart = round(mean(O\_aisle\_ratio\_in\_cart),2),  
 U\_avg\_dept\_ratio\_in\_cart = round(mean(O\_dept\_ratio\_in\_cart),2),  
 U\_avg\_cart\_size = round(mean(O\_cart\_size),1)) -> tmp\_uf  
  
user\_feature %>%   
 merge(tmp\_uf) -> user\_feature  
rm(tmp\_uf)  
rm(orders\_features)

### Gather the complete list of products the customer ever purchased

#user\_prior\_orders %>%   
# group\_by(user\_id,product\_id) %>%   
# summarize(UP\_last\_purchased = max(O\_day) ) ->user\_prod\_cross  
#write.csv(user\_prod\_cross,"user\_prod\_cross\_complete.csv")

# user\_cross\_product to identify first time orders

user\_prior\_orders %>% group\_by(user\_id,product\_id) ->user\_prod\_feature\_for\_new

# User\_product\_cross features

Summarize at cross-product-level 1, number of orders 2. last day ordered 3. mostly ordered dow 4. mostly ordered hod 5. repeat frequency 6. percentage or orders of the product 7. Days since last order placed as a ratio of reorder frequency

user\_prior\_orders %>%   
 filter(U\_last\_O\_day - O\_day<=180) ->user\_prior\_orders  
  
user\_prior\_orders %>%   
 group\_by(user\_id,product\_id) %>%   
 mutate(UP\_mostly\_ord\_dow = getmode(order\_dow),  
 UP\_mostly\_ord\_hod = getmode(order\_hour\_of\_day)) %>%   
 group\_by(order\_id,user\_id,product\_id) %>%   
 mutate(UPO\_ord\_dow\_abs\_var = dowDiff(order\_dow,UP\_mostly\_ord\_dow),  
 UPO\_ord\_hod\_abs\_var= hrDiff(order\_hour\_of\_day,UP\_mostly\_ord\_hod)) ->UP\_tmp  
  
UP\_tmp %>%   
 group\_by(user\_id,product\_id,#U\_last\_O\_day,  
 UP\_mostly\_ord\_dow,UP\_mostly\_ord\_hod,aisle\_id,department\_id) %>%   
 summarize(UP\_prd\_ord\_count = n(),  
 UP\_days\_since\_last\_purchase = first(U\_last\_O\_day) - max(O\_day),  
 UP\_ord\_dow\_abs\_var = median(UPO\_ord\_dow\_abs\_var),  
 UP\_ord\_hod\_abs\_var= median(UPO\_ord\_hod\_abs\_var),  
 UP\_pur\_duration = max(O\_day) - min(O\_day),  
 UP\_add\_to\_cart\_order =   
 round(mean(add\_to\_cart\_order/O\_cart\_size),2)) %>%   
 mutate(UP\_Pur\_freq\_days =round(UP\_pur\_duration/UP\_prd\_ord\_count,1)) %>%   
 mutate(UP\_pur\_due = ifelse(UP\_Pur\_freq\_days==0,0,   
 round(UP\_days\_since\_last\_purchase/UP\_Pur\_freq\_days,2 ))) -> UP\_tmp  
  
 UP\_tmp[,c("user\_id"   
 ,"product\_id"   
 ,"UP\_prd\_ord\_count"   
 ,"UP\_days\_since\_last\_purchase"  
 ,"UP\_mostly\_ord\_dow"  
 ,"UP\_ord\_dow\_abs\_var"   
 ,"UP\_mostly\_ord\_hod"  
 ,"UP\_ord\_hod\_abs\_var"   
 ,"UP\_pur\_duration"   
 ,"UP\_add\_to\_cart\_order"   
 ,"UP\_Pur\_freq\_days"   
 ,"UP\_pur\_due"   
 ,"aisle\_id"   
 ,"department\_id")] ->user\_prod\_feature   
  
  
rm(UP\_tmp)

# User\_aisle features

user\_prior\_orders %>%   
 group\_by(user\_id,aisle\_id) %>%   
 mutate(UA\_mostly\_ord\_dow = getmode(order\_dow),  
 UA\_mostly\_ord\_hod = getmode(order\_hour\_of\_day)) %>%   
 group\_by(order\_id,user\_id,aisle\_id) %>%   
 mutate(UAO\_aisle\_size = n(),  
 UAO\_ord\_dow\_abs\_var = dowDiff(order\_dow,UA\_mostly\_ord\_dow),  
 UAO\_ord\_hod\_abs\_var= hrDiff(order\_hour\_of\_day,UA\_mostly\_ord\_hod)) ->UA\_tmp  
  
UA\_tmp %>%   
 group\_by(user\_id,aisle\_id,#U\_last\_O\_day,  
 UA\_mostly\_ord\_dow,UA\_mostly\_ord\_hod) %>%   
 summarize(UA\_aisle\_ord\_count = n(),  
 UA\_days\_since\_last\_purchase = first(U\_last\_O\_day) - max(O\_day),  
 UA\_ord\_dow\_abs\_var = median(UAO\_ord\_dow\_abs\_var),  
 UA\_ord\_hod\_abs\_var= median(UAO\_ord\_hod\_abs\_var),  
 UA\_pur\_duration = max(O\_day) - min(O\_day),  
 UA\_add\_to\_cart\_order = round(mean(add\_to\_cart\_order/O\_cart\_size),2)) %>%   
 mutate(UA\_Pur\_freq\_days =round(UA\_pur\_duration/UA\_aisle\_ord\_count,1)) %>%   
 mutate(UA\_pur\_due = ifelse(UA\_Pur\_freq\_days==0,0,   
 round(UA\_days\_since\_last\_purchase/UA\_Pur\_freq\_days,2 ))) -> UA\_tmp  
  
 UA\_tmp[,c("user\_id"   
 ,"aisle\_id"   
 ,"UA\_aisle\_ord\_count"   
 ,"UA\_days\_since\_last\_purchase"  
 ,"UA\_mostly\_ord\_dow"  
 ,"UA\_ord\_dow\_abs\_var"   
 ,"UA\_mostly\_ord\_hod"  
 ,"UA\_ord\_hod\_abs\_var"   
 ,"UA\_pur\_duration"   
 ,"UA\_add\_to\_cart\_order"   
 ,"UA\_Pur\_freq\_days"   
 ,"UA\_pur\_due")] ->user\_aisle\_feature   
  
rm(UA\_tmp)

# User Department features

user\_prior\_orders %>%   
 group\_by(user\_id,department\_id) %>%   
 mutate(UD\_mostly\_ord\_dow = getmode(order\_dow),  
 UD\_mostly\_ord\_hod = getmode(order\_hour\_of\_day)) %>%   
 group\_by(order\_id,user\_id,department\_id) %>%   
 mutate(UDO\_dept\_size = n(),  
 UDO\_ord\_dow\_abs\_var = dowDiff(order\_dow,UD\_mostly\_ord\_dow),  
 UDO\_ord\_hod\_abs\_var= hrDiff(order\_hour\_of\_day,UD\_mostly\_ord\_hod)) ->UD\_tmp  
  
UD\_tmp %>%   
 group\_by(user\_id,department\_id,#U\_last\_O\_day,  
 UD\_mostly\_ord\_dow,UD\_mostly\_ord\_hod) %>%   
 summarize(UD\_dept\_ord\_count = n(),  
 UD\_days\_since\_last\_purchase = first(U\_last\_O\_day) - max(O\_day),  
 UD\_ord\_dow\_abs\_var = median(UDO\_ord\_dow\_abs\_var),  
 UD\_ord\_hod\_abs\_var= median(UDO\_ord\_hod\_abs\_var),  
 UD\_pur\_duration = max(O\_day) - min(O\_day),  
 UD\_add\_to\_cart\_order = round(mean(add\_to\_cart\_order/O\_cart\_size),2)) %>%   
 mutate(UD\_Pur\_freq\_days =round(UD\_pur\_duration/UD\_dept\_ord\_count,1)) %>%   
 mutate(UD\_pur\_due = ifelse(UD\_Pur\_freq\_days==0,0,   
 round(UD\_days\_since\_last\_purchase/UD\_Pur\_freq\_days,2 ))) -> UD\_tmp  
  
 UD\_tmp[,c("user\_id"   
 ,"department\_id"   
 ,"UD\_dept\_ord\_count"   
 ,"UD\_days\_since\_last\_purchase"  
 ,"UD\_mostly\_ord\_dow"  
 ,"UD\_ord\_dow\_abs\_var"   
 ,"UD\_mostly\_ord\_hod"  
 ,"UD\_ord\_hod\_abs\_var"   
 ,"UD\_pur\_duration"   
 ,"UD\_add\_to\_cart\_order"   
 ,"UD\_Pur\_freq\_days"   
 ,"UD\_pur\_due")] ->user\_dept\_feature   
  
rm(UD\_tmp)  
rm(user\_prior\_orders)

# Build Future order\_features

orders[orders$eval\_set == "train",c(2,5,6,7,8)] ->user\_FUT\_features  
  
colnames(user\_FUT\_features)=c("user\_id","FUT\_order\_dow","FUT\_order\_hod","FUT\_days\_since","FUT\_O\_day")  
  
#write.csv(user\_FUT\_features, "user\_FUT\_features.csv",col.names = NA)

# Training dataset

user\_prod\_feature %>%   
 left\_join(user\_feature, by="user\_id") %>%   
 left\_join(user\_aisle\_feature, by = c("user\_id", "aisle\_id")) %>%   
 left\_join(user\_dept\_feature, by = c("user\_id", "department\_id")) %>%   
 left\_join(user\_FUT\_features,by=c("user\_id")) %>% setDT() ->prior\_orders\_features  
  
rm(user\_prod\_feature)

# Prepare future orders(outcomes) for training

# train user set   
orders[orders$eval\_set == "train",c(1,2)] ->future\_ord  
future\_ord[future\_ord$user\_id %in% user\_set$user\_id, ] ->future\_ord  
  
future\_ord %>%   
 left\_join(train\_orders) %>%   
 select(user\_id,product\_id,reordered) ->future\_ord

## Joining, by = "order\_id"

# Final training file for predicting reorders

prior\_orders\_features %>%   
 left\_join(future\_ord,by = c("user\_id", "product\_id")) %>%   
 mutate(ORDERED = ifelse(is.na(reordered),0,1) ) -> training\_data  
  
  
apply(training\_data[,c("UP\_mostly\_ord\_hod","FUT\_order\_hod")], 1 ,  
 function(x) hrDiff(x[1],x[2])) ->  
 training\_data$UP\_hod\_diff  
  
apply(training\_data[,c("UP\_mostly\_ord\_dow","FUT\_order\_dow")], 1 ,  
 function(x) dowDiff(x[1],x[2])) ->   
 training\_data$UP\_dow\_diff  
  
write.csv(training\_data, "training\_data10k.csv")

# Training data for New orders

future\_ord %>%   
 left\_join(products,by="product\_id") %>%   
 left\_join(user\_feature, by="user\_id") %>%   
 left\_join(user\_aisle\_feature, by = c("user\_id", "aisle\_id")) %>%   
 left\_join(user\_dept\_feature, by = c("user\_id", "department\_id")) %>%   
 left\_join(user\_FUT\_features,by=c("user\_id")) %>% setDT() %>%   
 anti\_join(user\_prod\_feature\_for\_new, by=c("user\_id","product\_id")) ->new\_order\_training\_data  
  
rm(user\_feature)  
rm(user\_aisle\_feature)  
rm(user\_dept\_feature)  
rm(user\_FUT\_features)  
  
  
write.csv(new\_order\_training\_data,"new\_order\_training\_data.csv")