~~~
---
title: "BA501HW_Xiangyu_Zeng"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```



```{r}
#Load packages that will be used later
library(data.table)
library(sqldf)
library(dplyr)
library(glmnet)
library(caTools)
library(ROCR)



#Read in the data ; I shorten the name as to be more concise
customer<-fread("customer_table.csv")
orders<- fread("order_table.csv")
product<- fread("product_table.csv")
category<- fread("category_table.csv")

#### to change from scientific notation to the actual number
customer[,customer_id := as.character(customer$customer_id)]
orders[,customer_id := as.character(orders$customer_id)]
orders[,order_id := as.character(orders$order_id)]
orders[,product_id := as.character(orders$product_id)]
product[,product_id := as.character(product$product_id)]

```


```{r}
#1.Select out customers in the customer table who
#only made one purchase before 2016/12/22
#I use filter function to select those customers,
#mutate to add the new variables to the results

before = filter(order_table,order_date<20161222)
purchase = mutate(group_by(before,customer_id),count=n())
~~~

~~~
onepurchase = filter(purchase,count==1)

#The head of the results
head(onepurchase)

```


```{r}
#1.
#Select customers who made only one purchase before 2016/12/22;
onepurchase<-orders %>%
        filter(order_date<20161222,order_amount>0) %>%
        group_by(customer_id) %>%
        mutate(count=n()) %>%
        filter(count==1)
#The head of the results
head(onepurchase)
```


```{r}
#2. For users in step 1, select users who did not purchase
#anything from 2016/12/12 and 2017/02/22
#aka elect users go dormant for 3 months in onepurchase

#Base on experiences from the previous homework, run sql
#code in r, where I make left joins betweeen the two tables
#to choose the targeted dormant customer group.
dormant<-sqldf("select b.customer_id,b.product_id,b.order_amount
               from onepurchase b
               left join orders o on o.customer_id=b.customer_id
               and o.order_date>=20161222
               and o.order_date<20170222
               where o.customer_id is NULL")
#The head of the results
head(dormant)

```


```{r}
#2. double check
#Select users go dormant for 3 months in onepurchase;
is_buyer<-orders %>%
        filter(order_date>=20161222,order_date<20170222,order_amount>0) %>%
~~~

~~~~
    group_by(customer_id) %>%
    summarise(sum(order_amount))
dormant<-onepurchase[!(onepurchase$customer_id %in% is_buyer$customer_id),]

#The head of the results
head(dormant)

```


```{r}
#3.For users in step2, if they purchase anything betweeen 2017
#/02/22 amd 2017/05/22
#Flag users who come back as 1
back = sqldf("select distinct d.customer_id,d.product_id,d.order_amount
      from dormant d
      join orders o on o.customer_id=d.customer_id
      where d.order_date>=20170222 AND d.order_date<=20170522")
back$back = 1

#Flag users who do not come back as 0
notback = sqldf("select d.customer_id,d.product_id,d.order_amount
      from dormant d
       left join orders o on o.customer_id=d.customer_id
        and d.order_date>=20170222 and d.order_date<=20170522
        where o.customer_id is NULL")
notback$back = 0

sample = rbind(back,notback)
#The head of the results
head(sample)

```



```{r}
#4.Run a logistic regression using the rest of customer
#features in the customer table, plus anything related to
#their orders before 2016/12/22, including order amount,
#product they purchased, category they purchased.
#a. You have to remove the character type features
#(unless you know how to deal with them), as logistic
#regression could only run a numeric value
#b. You have to deal with missing values (users whose
~~~~

```
#features have ‘NA’). For now you could just simply
#remove those customers

#Fill in features
sample_features<-inner_join(sample,customer,by="customer_id")

#handle characters
sample_features$country<-as.numeric(as.factor(sample_features$country))
sample_features$gender<-as.numeric(as.factor(sample_features$gender))
sample_features<-sample_features[,-c(1,11,12,16,117)]

#Delete NAs
data<-sample_features[complete.cases(sample_features),]

#Prepare datasets
set.seed(1234)
split<-sample.split(data$back, SplitRatio = 0.6)
train<-subset(data, split == TRUE)
test<-subset(data, split == FALSE)

#logistic regression
model<-glm(back~.,family=binomial, train)
summary(model)
```