Predicting CTR depending on session length

2023-05-18

## R Markdown

# We have two datasets – variety\_train.RData and variety\_test.RData for training and testing, respectively.

# The columns in both datasets are defined below:

# 1. **click** – Indiator for whether the impression is clicked. 1 implies that a click occured and zero implies

# that a click did not occur.

# 2. **timeofday** – Can take one of four values – 1 for late night (12 am to 6 am), 2 for morning (6 am to 12

# pm), 3 for afternoon (12 pm to 6 pm), and 4 for early night (6 pm to 12 am).

# 3. **imptotal** – Total number of impressions shown to this user prior to this session. It captures how active

# the user has been on her mobile device.

# 4. **ctruser** – the average CTR the user has had prior to the session.

# 5. **varietytotal** – total number of distinct ads the user has seen prior to this session.

# 6. **adimptotal** – total number of impressions of client’s ad shown to the user prior to this session.

# 7. **variety** – number of distinct ads shown earlier in the session.

# 8. **rep** – number of times the ad is replaced with the same ad earlier within this session.

# 9. **adimpsession** – number of times client’s ad has been shown earlier in the session.

install.packages(“usethis”)

load(“variety\_test.RData”) load(“variety\_train.RData”)

# **EDA**

#*getting summary of the training data* summary(variety\_train)

# The observed CTR of the user is 0.1134 whereas the average ctruser is 0.1164.

# The observed CTR and the ctruser is almost the same with very minor difference.

# Having said so the CTR seems very high (11.34% and 11.64%).

# This is a little unusual , but probably because the sample is of active users,

# who tend to spend more time on apps and are more willing to engage with an ad.

# *understanding how the test and train data is distributed*

hist(variety\_trainvarietytotal, xlim=c(5,70), ylim=c(0,20000))

# From the first histogram we see that there is a little bit of variation in the number of distinct ads shown

# in a particualr session, but mostly a user would see 3-4 distinct ads within a session. The data is a little

# left skewed.

# From the second histograms we see that the total number of distinct ads seen by most user in previous

# sessions ranges from 15 to 30 and the data is right skewed.

#*understanding if any correlation exists between in-session variables ~ variety and rep*

Cor <- cor.test(formula = ~ variety\_trainrep)

# The coefficient of correlation is pretty high, and the sign is negative (-0.70). Additionally p-value < 2.2e-16

# This means there is a significant -ve correlation between **variety** – number of distinct ads shown earlier in the session.

# and **rep** – number of times the ad is replaced with the same ad earlier within this session

# This means that as the repetition of ads increases, the distinctiveness of ad shown in

# session decreases. This is obvious and pretty intuitive because when the same ad is shown

# more than once the the probability of showing distinct (new) ad within a session would decrease

# assuming every eighth ad is Client ad and for the sake of argument we assume that one session

# is a eight ad session.

# to check if customer is more or less likely to click if you have seen a higher variety

# of ads previously?

install.packages(“gplots”, repos=“<https://cran.rstudio.com>”) library(gplots) plotmeans(variety\_trainvariety)

#The difference between 6TH AD shown and 7 add showns is not significant, and therefore we cannot # tell if the effect of variety and click stays positive at 7 ads are shown.

# **CART Analysis**

# What previous session behaviors affect the click probability. Predicting the behavior.( contextual)

install.packages(“rpart” , repos=“<https://cran.rstudio.com>”) library(rpart) behavioral.context <- click ~ variety + rep + adimpsession behavioral.tree = rpart(formula= behavioral.context, data= variety\_train, control=rpart.control(cp=0.00032)) install.packages(“rpart.plot”) library(rpart.plot) rpart.plot(behavioral.tree)

# The tree has 5 leaves. This tree tells us that the average probability of click for all

# data is 0.11 Variety looks like the single most important variable that matters. Repeat

# and AdImp are the omitted variables. Looks like users are more likely to click when they see a

# variety of ads. When the variety is more than 4 per in session ads, the probability that 54%

# of impression will be click is 0.14, which is a significantly high number.

# Going down the same branch we see the left leaf which suggests, 32% of impression have a 0.12

# probability of being clicked on the ad and see less than 5 distinct ads in session

behavioral.prediction = predict(behavioral.tree, variety\_test) variety\_test$behavioral.pred = behavioral.prediction head(variety\_test)

# predicting based on behavior before the session ( behavioral)

behavioral.prior <- click ~ imptotal + adimptotal + varietytotal + ctruser behavioral.prior.tree = rpart(formula= behavioral.prior, data= variety\_train, control=rpart.control(cp=0.00032))

library(rpart.plot) rpart.plot(behavioral.prior.tree)

# This tree has seven leaves. It looks like CTR is the main variable in this tree and

# rest of the variables are omitted. This kind of makes sense, because if in the past session

# a user has higher probability of clicking on an ad then ctruser can help predict what the

# user might do in other session based on his behavior data. This can be even seen in the tree.

# If ctruser is higher than 0.45 then an impression has 64% probability of receiving a click.

# If CTR < 0.45 7 % of impressions still have a 0.22 probabaility of getting clicked.

behavioral.prior.prediction = predict(behavioral.prior.tree, variety\_test) variety\_test$behavioral.prior.pred = behavioral.prior.prediction head(variety\_test)

#Full tree ( contexual + behavioral)

behavioral.full <- click ~ variety + rep + adimpsession + imptotal + adimptotal + varietytotal + ctruser + timeofday

behavioral.full.tree = rpart(formula= behavioral.full, data= variety\_train, control=rpart.control(cp=0.00032))

rpart.plot(behavioral.full.tree)

# There are 17 leaves in the tree above. The variables that matter are ctuser,

# variety and adimpsession. The omitted variables are rep, varietytotal, timeofday,

# adimptotal, imptotal. The tree is unbalanced. if the ctruser is greater than 0.14

# then there is 17% chance that 35% of the impression will receive clicks. When adimpsession

# is greater than or equal to 2.5 there is 14% chance that 10% of the impresssion will be

# clicked.

behavioral.full.prediction = predict(behavioral.full.tree, variety\_test) variety\_test$behavioral.full.pred = behavioral.full.prediction

head(variety\_test)

# model comparision

#Baseline CTR base\_mean = mean(variety\_test$click)

# mean of behavioral data with no click in session

beh\_NC = mean(variety\_testclick==0])

# mean of behavioral data with click in session

beh\_C = mean(variety\_testclick==1])

# mean of behavioral data with no click in prior session

Prior\_NC = mean(variety\_testclick==0])

# mean of behavioral data with click in prior session

Prior\_C = mean(variety\_testclick==1])

# mean of behavioral data with no click in for full tree session

F\_NC = mean(variety\_testclick==0])

# mean of behavioral data with click in for full tree session

F\_C = mean(variety\_testclick==1])