Effect of ad-variety and session-level history on customer response to mobile advertising

Aakanksha Baid

2/25/2020

## 1. Descriptive Analysis

#load data  
  
load("variety\_train.RData")  
load("variety\_test.RData")  
  
summary(variety\_train)

## click timeofday imptotal ctruser   
## Min. :0.0000 Min. :1.000 Min. : 0.0 Min. :0.0000013   
## 1st Qu.:0.0000 1st Qu.:2.000 1st Qu.:147.0 1st Qu.:0.0512066   
## Median :0.0000 Median :3.000 Median :218.0 Median :0.1048145   
## Mean :0.1134 Mean :2.799 Mean :222.3 Mean :0.1164985   
## 3rd Qu.:0.0000 3rd Qu.:4.000 3rd Qu.:291.0 3rd Qu.:0.1684327   
## Max. :1.0000 Max. :4.000 Max. :686.0 Max. :0.5133861   
## varietytotal adimptotal variety rep   
## Min. : 5.00 Min. : 0.00 Min. :1.000 Min. :0.000   
## 1st Qu.:19.00 1st Qu.:13.00 1st Qu.:3.000 1st Qu.:1.000   
## Median :24.00 Median :19.00 Median :4.000 Median :2.000   
## Mean :23.91 Mean :19.69 Mean :3.714 Mean :1.711   
## 3rd Qu.:28.00 3rd Qu.:26.00 3rd Qu.:4.000 3rd Qu.:3.000   
## Max. :68.00 Max. :60.00 Max. :7.000 Max. :6.000   
## adimpsession   
## Min. :0.000   
## 1st Qu.:2.000   
## Median :3.000   
## Mean :2.893   
## 3rd Qu.:4.000   
## Max. :7.000

str(variety\_train)

## 'data.frame': 63283 obs. of 9 variables:  
## $ click : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ timeofday : int 1 2 4 4 2 4 4 1 4 4 ...  
## $ imptotal : int 159 244 324 96 119 304 289 184 244 387 ...  
## $ ctruser : num 0.03469 0.21833 0.00459 0.00637 0.02491 ...  
## $ varietytotal: int 19 14 14 15 20 23 26 16 18 21 ...  
## $ adimptotal : int 17 27 36 8 8 28 27 15 22 38 ...  
## $ variety : int 2 1 2 3 2 5 4 2 3 3 ...  
## $ rep : int 3 6 5 1 4 1 1 4 4 4 ...  
## $ adimpsession: int 5 7 6 3 6 2 3 5 5 5 ...

## 1a What is the observed CTR in the data and the average of the users’ past click through rate? Are these numbers as expected?

# observed ctr  
ctr\_avg <- mean(variety\_train$click)  
ctr\_avg

## [1] 0.1134112

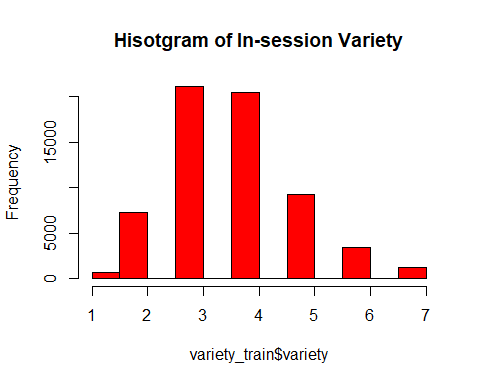
# average of the users’ CTR  
ctr\_user <- mean(variety\_train$ctruser)  
ctr\_user

## [1] 0.1164985

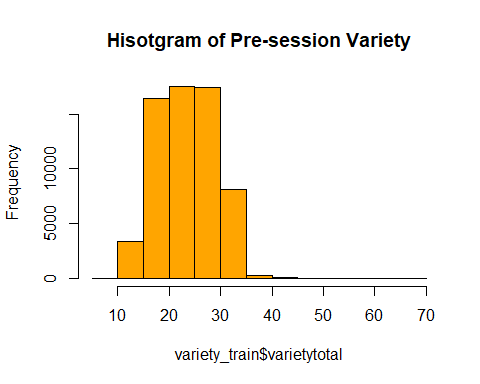
The mean of clicks is 0.1134 or observed CTR in the data is 11.34% while the average user click through rate is 11.65%. Though the average clicks and average user click through rate are close, these numbers are not as expected as they are higher than the typical CTR of ads, which is aound 2%. As the observed CTR and average user CTR are high they are considered above average CTR.

## 1b Plot historgrams of in-session variety (variety), and pre-session variety (varietytotal). What do you infer from the plots?

# histograms  
hist(variety\_train$variety, main = 'Hisotgram of In-session Variety',col = 'red')



hist(variety\_train$varietytotal, main = 'Hisotgram of Pre-session Variety',col = 'orange')



From the in-session histogram, I infer that users are typically shown less than 5 distinct ads while majority of distinct ads are between 3 - 4. From the pre-session histogram, I can infer that users have typically seen between 10 to 35 distinct ads prior to this session. So the user typically has seen more distinct ads prior to the session than during the session.

## 1c Run a correlation test between the two in-session variables (variety) and (rep)? What do you infer from the sign and the magnitude of the correlation?

# correlation  
cor(variety\_train$variety, variety\_train$rep)

## [1] -0.7011752

cor.test(variety\_train$variety, variety\_train$rep)

##   
## Pearson's product-moment correlation  
##   
## data: variety\_train$variety and variety\_train$rep  
## t = -247.39, df = 63281, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.7051143 -0.6971927  
## sample estimates:  
## cor   
## -0.7011752

From the Pearson’s correlation test between variety and rep, the correlation coefficient = -0.701152. As the p value of test < 2.2e-16, which is less than the significance level of 0.05, so we can conclude that variety and rep are significantly negatively correlated (due to the negative sign). As magnitude of correlation (0.7) is higher than 0.6, the correlation is strong. Thus variety and rep are strongly negatively correlated with a significant correlation between them. This is intuitive as variety of ads or distinct ads increases, the repeats or number of times the ad is replaced with the same ad within session decreases.

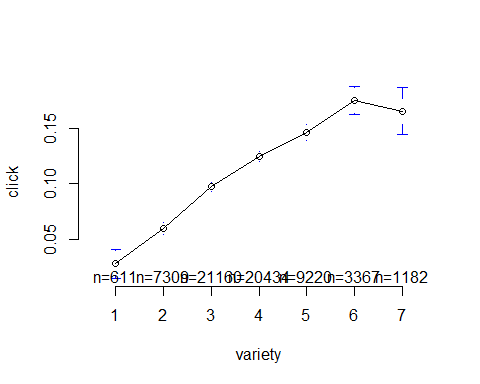
## 1d Plot the average or mean CTR at each level of in-session variety. Now based on this graph, interpret the relationship between in-session variety and click? Are you more or less likely to click if you have seen a higher variety of ads previously?

library(gplots)

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

# Plot the mean of mean CTR by variety  
plotmeans(click ~ variety, data = variety\_train,frame=FALSE, mean.lables = TRUE)

 Based on the graph, I infer that in-session variety is positively related to clicks. As in-session variety increases, the number of impression clicks also increases. You are more likely to click if you have seen a higher variety of ads previously in the session because of this positive relationship. This trend is observed till variety of ads is 6 after which the number of clicks declines. It could be because after 6 distinct ads, the user gets annoyed by the too many ads and stops clicking after that.

## 1e Based on how the experiment was run, do you think this effect is causal? That is, is variety causing the changes in CTR that you see in the graph or is this simply a correlation between CTR and variety?

# correlation  
cor(variety\_train$variety, variety\_train$click)

## [1] 0.09435889

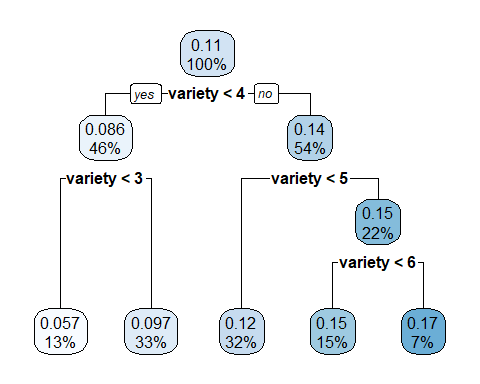
Based on the experiment setup, this effect is causal as seeing upto 6 distinct ads previously leads to more clicks. This experiement was run on a sample of active users who have a higher propensity to click as more ads are shown to them which gives the positive relationship between clicks and variety. After seeing 6 distinct ads, impression clicks decline as maybe the users get annoyed with seeing too may distinct ads. Also, variety and observed CTR are weakly correlated at 0.09. Had this effect been due to correlation between CTR and variety, then the clicks would have increased with increase in distinct number of ads and not decline after 6 distinct ads. Thus this causation is not due to correlation.

## 2a Estimate a CART model (to predict click) with the three within-session behavioral history variables on the training data.

library(rpart)  
  
behavioral1.model <- click ~ variety + rep + adimpsession  
  
behavioral1.tree <- rpart(formula = behavioral1.model,   
 data = variety\_train, control = rpart.control(cp = 0.00032))

## 2b Visualize this CART model and give a short overview of your findings.

library(rpart.plot)  
rpart.plot(behavioral1.tree)

 The tree has 5 leaves and only uses 1 variable for splitting - variety. Thus only variety matters here. The variables rep and adimpsession are omitted. The tree is pretty balanced with 46% data contains clicks when the ad variety is less than 4 with the average click probability = 0.086. When ad variety is between 3 and 4, 33% of data contains clicks with average click probability = 0.097.

## 2c Predict on the test dataset with this CART model and store the predictions in a column named ‘withinsession.CART.pred’.

behavioral1.CART.prediction <- predict(behavioral1.tree, variety\_test)  
variety\_test$withinsession.CART.pred <- behavioral1.CART.prediction

## 2d Estimate an XGBoost model (to predict click) with the three within-session behavioral history variables using the training dataset.

library(xgboost)  
  
col.behavioral1 = c(7,8,9)  
  
xgb1.behavioral <- xgboost(data = data.matrix(variety\_train[,col.behavioral1]),   
 label = variety\_train[,1],   
 eta = 0.1,  
 max\_depth = 6,   
 nround=100,   
 subsample = 1,  
 colsample\_bytree = 1,  
 num\_class = 1,  
 min\_child\_weight = 5,  
 gamma = 5,  
 nthread = 30,  
 eval\_metric = "logloss",  
 objective = "binary:logistic",  
 verbose = 0  
 )

## 2e Predict on the test dataset with this XGBoost model and store the predictions in a column named ‘withinsession.xgb.pred’

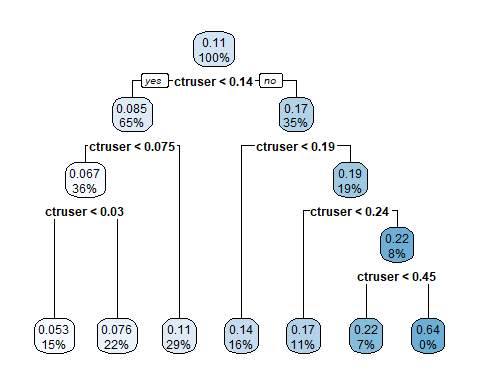
variety\_test$withinsession.xgb.pred <- predict(xgb1.behavioral, data.matrix(variety\_test[,col.behavioral1]))

## 3a Estimate a CART model (to predict click) with the four pre-session behavioral history variables on the training data.

behavioral2.model <- click ~ imptotal + ctruser + varietytotal + adimptotal  
  
behavioral2.tree <- rpart(formula = behavioral2.model,   
 data = variety\_train, control = rpart.control(cp = 0.00032))

## 3b Visualize this CART model and give a short overview of your findings.

rpart.plot(behavioral2.tree)



The tree has 7 leaves and only uses 1 variable for splitting - ctruser. Thus only ctruser matters. The variables varietytotal, adimptotal and imptotal are omitted. The tree is split unbalanced with 65% data contains clicks when the average user CTR prior to this session is less than 0.14 with the average click probability = 0.085. When user’s prior session CTR is between 0.075 and 0.14, 29% of data contains clicks with average click probability = 0.11.

## 3c Predict on the test dataset with this CART model and store the predictions in a column named ‘presession.CART.pred’.

behavioral2.CART.prediction <- predict(behavioral2.tree, variety\_test)  
variety\_test$presession.CART.pred <- behavioral2.CART.prediction

## 3d Estimate an XGBoost model (to predict click) with the four pre-session behavioral history variables using the training dataset.

col.behavioral2 = c(3,4,5,6)  
  
xgb2.behavioral <- xgboost(data = data.matrix(variety\_train[,col.behavioral2]),   
 label = variety\_train[,1], method="class",  
 eta = 0.1,  
 max\_depth = 6,   
 nround=100,   
 subsample = 1,  
 colsample\_bytree = 1,  
 num\_class = 1,  
 min\_child\_weight = 5,  
 gamma = 5,  
 # nthread = 30,  
 # eval\_metric = "logloss",  
 objective = "binary:logistic"  
 ,verbose = 0  
 )

## 3e Predict on the test dataset with this XGBoost model and store the predictions in a column named ‘presession.xgb.pred’

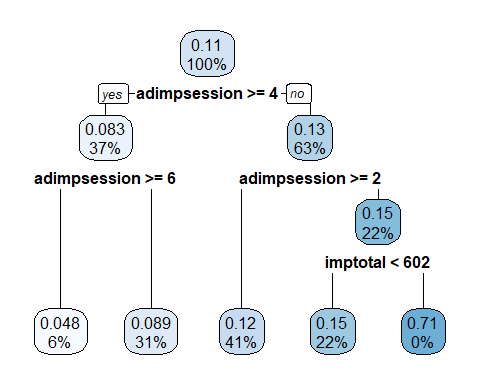
variety\_test$presession.xgb.pred <- predict(xgb2.behavioral, type = 'class', data.matrix(variety\_test[,col.behavioral2]))

## 4a Estimate a CART model (to predict click) with all the impression-level variables on the training data.

behavioral3.model <- click ~ imptotal + adimptotal + adimpsession  
  
behavioral3.tree <- rpart(formula = behavioral3.model,   
 data = variety\_train, control = rpart.control(cp = 0.00032))

## 4b Visualize this CART model and give a short overview of your findings.

rpart.plot(behavioral3.tree)



The tree has 5 leaves and only uses 2 of the 3 impression variables for splitting - adimpsession and imptotal. Thus only adimpsession and imptotal matters. The variable adimptotal is omitted. The tree is split unbalanced with 37% data contains clicks when the number of times EA’s ad shown earlier in session is 4 or more with the average click probability = 0.083. When adimpsession is between 2 and 4, 41% of data contains clicks with average click probability = 0.12.

## 4c Predict on the test dataset with this CART model and store the predictions in a column named ‘full.CART.pred’.

behavioral3.CART.prediction <- predict(behavioral3.tree, variety\_test)  
variety\_test$full.CART.pred <- behavioral3.CART.prediction

## 4d Estimate an XGBoost model (to predict click) with all the variables using the training dataset.

col.behavioral3 = c(3,6,9)  
  
xgb3.behavioral <- xgboost(data = data.matrix(variety\_train[,col.behavioral3]),   
 label = variety\_train[,1],   
 eta = 0.1,  
 max\_depth = 4,   
 nround=100,   
 subsample = 1,  
 colsample\_bytree = 1,  
 num\_class = 1,  
 min\_child\_weight = 5,  
 gamma = 5,  
 nthread = 30,  
 eval\_metric = "logloss",  
 objective = "binary:logistic",  
 verbose = 0  
 )

## 4e Predict on the test dataset with this XGBoost model and store the predictions in a column named ‘full.xgb.pred’

variety\_test$full.xgb.pred <- predict(xgb3.behavioral, data.matrix(variety\_test[,col.behavioral3]))

## 5a First, use Area Under the Curve (AUC) to evaluate the performance of the six models presented above

library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

#CART1  
auc.withinsession.CART = roc(variety\_test$click, variety\_test$withinsession.CART.pred)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

auc.withinsession.CART

##   
## Call:  
## roc.default(response = variety\_test$click, predictor = variety\_test$withinsession.CART.pred)  
##   
## Data: variety\_test$withinsession.CART.pred in 18719 controls (variety\_test$click 0) < 2376 cases (variety\_test$click 1).  
## Area under the curve: 0.5763

#XgBoost1  
auc.withinsession.xgb = roc(variety\_test$click, variety\_test$withinsession.xgb.pred)

## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases

auc.withinsession.xgb

##   
## Call:  
## roc.default(response = variety\_test$click, predictor = variety\_test$withinsession.xgb.pred)  
##   
## Data: variety\_test$withinsession.xgb.pred in 18719 controls (variety\_test$click 0) < 2376 cases (variety\_test$click 1).  
## Area under the curve: 0.5834

#CART2  
auc.presession.CART = roc(variety\_test$click, variety\_test$presession.CART.pred)

## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases

auc.presession.CART

##   
## Call:  
## roc.default(response = variety\_test$click, predictor = variety\_test$presession.CART.pred)  
##   
## Data: variety\_test$presession.CART.pred in 18719 controls (variety\_test$click 0) < 2376 cases (variety\_test$click 1).  
## Area under the curve: 0.6385

#XgBoost2  
auc.presession.xgb = roc(variety\_test$click, variety\_test$presession.xgb.pred)

## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases

auc.presession.xgb

##   
## Call:  
## roc.default(response = variety\_test$click, predictor = variety\_test$presession.xgb.pred)  
##   
## Data: variety\_test$presession.xgb.pred in 18719 controls (variety\_test$click 0) < 2376 cases (variety\_test$click 1).  
## Area under the curve: 0.6425

#CART full  
auc.full.CART = roc(variety\_test$click, variety\_test$full.CART.pred)

## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases

auc.full.CART

##   
## Call:  
## roc.default(response = variety\_test$click, predictor = variety\_test$full.CART.pred)  
##   
## Data: variety\_test$full.CART.pred in 18719 controls (variety\_test$click 0) < 2376 cases (variety\_test$click 1).  
## Area under the curve: 0.5641

#XgBoost full  
auc.full.xgb = roc(variety\_test$click, variety\_test$full.xgb.pred)

## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases

auc.full.xgb

##   
## Call:  
## roc.default(response = variety\_test$click, predictor = variety\_test$full.xgb.pred)  
##   
## Data: variety\_test$full.xgb.pred in 18719 controls (variety\_test$click 0) < 2376 cases (variety\_test$click 1).  
## Area under the curve: 0.5672

Table for comparison -

|  |  |  |  |
| --- | --- | --- | --- |
|  | Within Session | Pre Session | Full |
| CART | 0.6385 | 0.6385 | 0.5641 |
| XGBoost | 0.5834 | 0.6425 | 0.5672 |

Summary -

1. We see that the AUC of the within session CART model is 0.6385, which is same as that from the pre session CART model. The AUC for the full CART model is 0.5641, which is lower than that of the within and pre session CART models. So both within session and pre-session CART models are equally valuable and more valuable than full impression CART model.
2. The AUC of the within session XgBoost model is low at 0.5834 and that of full XgBoost model is even lower at 0.5672. The pre session XgBoost models has the highest AUC = 0.6425 and is the most valuable model.
3. Also, for both CART and XgBoost, the pre session models have the highest AUC than within session and full model. This implies that the user’s ad exposure and behavior before the session is more valuable than that within the same session and even than that of full data on impressions.
4. But XgBoost outperforms CART in 2 out of 3 session scenarios, with higher AUC in pre session. So XgBoost is generally a better model than CART.
5. Also, the predictive performance of the models is good but not that high as AUC is not quite good (< 0.7).

## 5b Next, use Relative Information Gain (RIG) to evaluate the performance of the six models presented above

#function  
RIG <- function(pred,actual){  
 mean.outcome = mean(actual)  
 pred = pmin(pmax(pred, 0.0000001), 1-0.0000001)  
 llpred = mean(-log(pred)\*actual-log(1-pred)\*(1-actual))  
 llbase = mean(-log(mean.outcome)\*actual-log(1-mean.outcome)\*(1-actual))  
 rig = (1- llpred/llbase)\*100  
 return(rig)  
}  
  
#CART RIG's  
RIG(variety\_test$withinsession.CART.pred,variety\_test$click)

## [1] 1.21679

RIG(variety\_test$presession.CART.pred,variety\_test$click)

## [1] 3.47516

RIG(variety\_test$full.CART.pred,variety\_test$click)

## [1] 0.8106216

# XgBoost RIG's  
RIG(variety\_test$withinsession.xgb.pred,variety\_test$click)

## [1] 1.3396

RIG(variety\_test$presession.xgb.pred,variety\_test$click)

## [1] 3.535669

RIG(variety\_test$full.xgb.pred,variety\_test$click)

## [1] 0.9427142

Summary -

1. For the within session and pre session CART, the same score of 3.47516 means that we have 3.48% information gain relative to the case where we do not use any information except the average CTR. For the full impression CART model, the relative information gain drop to 0.81% as compared to the case where we do not use any information except the average CTR and is the lowest score even when compared to the XgBoost models. This suggests that within and pre session targeting information is more valuable than full impression for CART models.
2. For XgBoost models, the pre session score of 3.53 is highest meaning we have 3.53% information gain relative to the case where we do not use any information except the average CTR. The full Xg Boost score is lowest among the 3 XgBoost scores at 0.94%, just like the CART full model.
3. XgBoost outperforms CART in information gain compared to no informationn except average CTR in the 2 scenarios pre session and full impression. For within session, CART model score is better.
4. Pre session RIG are highest as compared to the within session and full impression models.
5. Pre-session XgBoost score reaches 3.53% RIG which is the highest than all targeting models and is the most valuable model.

Table fo comparison for RIG -

|  |  |  |  |
| --- | --- | --- | --- |
|  | Within Session | Pre Session | Full |
| CART | 3.47516 | 3.47516 | 0.8106216 |
| XGBoost | 1.3396 | 3.535669 | 0.9427142 |

## 5c Compare the performance of different models and summarize your findings on the relative predictive ability of the six models. What is the best model among these six?

The qualitative results from RIG table are exactly the same as that from the AUC table. 1. Relatively, within and pre session models have a better predictive power than the full impression model. The prior session model is the best as compared to the full imression and within session model. 2. XgBoost model give the best predictive performance except in the case of within session model where CART is better. Hence we should use XgBoost model prior to the session for most business purposes. 3. Overall, this suggests that, irrespective of the evaluation metric used, the XGBoost model that only consider the user’s ad exposure and behavior before the session is the best predictive model as it has the highest AUC and RIG.

In terms of predictive performance - Prior sesion > Within Session > Full impressions

## 6a What is the relative value of within-session user history vs. pre-session user history?

While both within session and pre session user history are valuable, the pre session user history is more valuable and has more benefits for behavioral targeting -

1. Prior session user history has larger data on ads and response and can give more accurate predictions than the within session data
2. Prior session user history helps advertisers decide advertising budget and make better decisions to maximize revenue for future
3. Prior session user history helps advertisers in dynamic retargeting and remarketing the correct users for the current session and for the future

## 6b What is the effect (positive or negative) of within-session variety on users’ ad response?

As seen from 1d, as within-session variety increases, the users’ ad response increases. So the effect of within-session variety is positive on users’ ad response. The more distinct ads the user is shown, the more impression clicks are generated as users need to be reminded multiple times through ads and need more stimulus to click the ad and respond. However, this trend is seen upto 6 distinct ads after which the clicks decline. This could be due to -

1. In the case when a user is repeatedly targeted with too many distinct ads during a session, users get irritated and do not click after a certain threshold of seeing the distinct ads.
2. After seeing an ad repeatedly during a session, it does not make any additional or significant impact on clicks after crossing the threshold. So within session user history can be costly to maintain if the right threshold of ads that generate clicks is not known.

So advertisers have to carefully set the optimum number of distinct ads shown to a user to maximize clicks that minimize cost and hence give the maximum ROI.

## 7a Identify the top 5000 of impressions with the highest predicted CTR (based on the best model that you identified in the previous question) and store these impressions in a separate dataframe.

As Pre session XgBoost model has the best predictive power, top impressions are identified using this model.

#sort pre session xgboost and get top 5000  
 temp <- variety\_test[order(-variety\_test$presession.xgb.pred),]  
top\_5000 <-temp[1:5000,]

## 7b What is the average CTR for these 5000 impressions? What is the average predicted CTR of these impressions based on your best model. Is your model-predicted average CTR close or similar to the true CTR observed in this subset of the data?

avg\_ctr <- mean(top\_5000$click)  
avg\_ctr

## [1] 0.1874

# Predicted CTR  
avg\_pred\_ctr <- mean(top\_5000$presession.xgb.pred)  
avg\_pred\_ctr

## [1] 0.1785241

The average CTR for these 5000 impressions is 18.74% whereas the average predicted CTR of these impressions based on the best xgBoost pre session model is lower at 17.85%. The model-predicted average CTR is only slightly lower than the true CTR observed in this subset of the data.

## 7c ROI calculation on test data

cost = 0.05  
price = 2  
  
# i Baseline RoI  
  
#marginal gain  
mg\_base = price\*mean(variety\_test$click)\*nrow(variety\_test)  
  
#marketing spend  
ms\_base = cost\*nrow(variety\_test)  
  
base\_ROI = (mg\_base - ms\_base)/ms\_base  
base\_ROI

## [1] 3.505333

# ii New RoI  
  
#marginal gain  
mg\_new = price\*mean(top\_5000$click)\*nrow(top\_5000)  
  
#marketing spend  
ms\_new = cost\*nrow(top\_5000)  
  
new\_ROI = (mg\_new - ms\_new)/ms\_new  
new\_ROI

## [1] 6.496

Baseline ROI = $3.5 New ROI = $6.5 New ROI based on the top 5000 impressions is higher than the baseline ROI by $3.

## 7d How should EA distribute this money between advertising and price promotions. Specifically, how many of the top impressions should EA buy (consider only multiples of 500, e.g., 500 impressions, 1000 impressions and so on), and what is the revenue and cost of this advertising spend? And how much should EA invest in price promotions?

EA should buy those number of top impressions that generate higher ROI than that of advertising or greater than 5.

#check in multiples of 500 which number of impressions maximize ROI  
  
# As variety test data has 21095 impressions, checking the ROI of highest multiple of 500 or from 20000 impressions-  
  
top\_20000 <-temp[1:20000,]  
mg\_new\_20000 = price\*mean(top\_20000$click)\*nrow(top\_20000)  
ms\_new\_20000 = cost\*nrow(top\_20000)  
new\_ROI\_20000 = (mg\_new\_20000 - ms\_new\_20000)/ms\_new\_20000  
new\_ROI\_20000

## [1] 3.662

# As ROI from 5000 impressions is $6.5, checking if a higher number of impression still gives higher ROI than price promotion ROI of $5  
 top\_10000 <-temp[1:10000,]  
#marginal gain  
mg\_new\_10000 = price\*mean(top\_10000$click)\*nrow(top\_10000)  
  
#marketing spend  
ms\_new\_10000 = cost\*nrow(top\_10000)  
  
new\_ROI\_10000 = (mg\_new\_10000 - ms\_new\_10000)/ms\_new\_10000  
new\_ROI\_10000

## [1] 5.216

#11000  
 top\_2 <-temp[1:11000,]  
#marginal gain  
mg\_new2 = price\*mean(top\_2$click)\*nrow(top\_2)  
mg\_new2

## [1] 3326

#marketing spend  
ms\_new2 = cost\*nrow(top\_2)  
ms\_new2

## [1] 550

new\_ROI2 = (mg\_new2 - ms\_new2)/ms\_new2  
new\_ROI2

## [1] 5.047273

As ROI from all 20,000 impressions is $3.662 which is less than ROI from price promotion ROI of $5, a smaller selection of ads should be made. Using a trial and error, the impressions that generate ROI > 5 lie between 10000 and 12500. The minimum impressions that lead to ROI > $5 is 11000. After 11,000 all multiples of 500 impressions give ROI less than 5. So EA should buy the top 11,000 impressions.

THe ROI from buying 11000 impressions is $5.04 The revenue of this advertising on top 11000 impressions is $3326. The cost of advertising on top 11000 impressions is $550.

So EA should allocate $550 to advertising on top 11000 impressions as these impressions give higher ROI than price promotion. The remainder amount of the $1000 budget (= $1000 - $550 = $450) or $450 should be invested in price promotions.