QMST 5334 Stat Methods: Final Project

Chinmay Bake

Venkata Sowjanya Koka

1. Introduction and Problem Statement

Cookie Cat is a hugely popular mobile application game. As players of the game would progress through different levels of the game, they might occasionally encounter gates that force them to wait a non-trivial amount of time or make an in-app purchases to make progress in the game. Apart from driving sales of the in-app purchases, these gates serve the important purpose of giving players an enforced break from playing the game, hopefully resulting in that the player's enjoyment of the game being increased.

The question which the developers of the game are trying to evaluate is - ***Initially, the first gate was placed at level 30. The gate for certain reason could be moved to level 40! What the developers would like to know that would this change impact retention of the customers at the end of 7 days?***

2. Data Definition Library:

- We first try to clearly define the definition of each column we have in our dataset –
- Total records - 90,189
- **userid** - a unique number that identifies each player.
- **version** – whether the player had Gate 30 or Gate 40 assigned to it
- **sum_gamerounds** - the number of game rounds played by the player during the first week after installation
- **retention_1** - did the player come back and play 1 day after installing?
- **retention_7** - did the player come back and play 7 days after installing?

When a player installed the game, he or she was randomly assigned to either gate_30 or gate_40

3. Exploratory Data Analysis & Defining the Baseline Metrics:

First, we try evaluating if each player has a unique identity or not.

```
> UniqueUserID <- unique(Game$userid); length(UniqueUserID)
[1] 90189
```

The length of unique userIDs corresponds to the total number of records implying that we do not have any evidence of duplication in userIDs. Now, we analyze descriptive statistics for both the versions. For that, we split the data with datapoints corresponding to each version

Gate 30 -

```
> summary(Data_Gate30)
     userid            version      sum_gamerounds     retention_1   retention_7
 Min.   :     116   gate_30:44700   Min.   :    0.00   FALSE:24666   FALSE:36198
 1st Qu.:2505469   gate_40:    0   1st Qu.:    5.00   TRUE :20034   TRUE : 8502
 Median :4983631                   Median :   17.00
 Mean   :4987564                   Mean   :   52.46
 3rd Qu.:7481497                   3rd Qu.:   50.00
 Max.   :9999710                   Max.   :49854.00
```

Gate 40 –

```
> summary(Data_Gate40)
     userid            version      sum_gamerounds     retention_1   retention_7
 Min.   :     377   gate_30:    0   Min.   :    0.0   FALSE:25370   FALSE:37210
 1st Qu.:2517171   gate_40:45489   1st Qu.:    5.0   TRUE :20119   TRUE : 8279
 Median :5007329                   Median :   16.0
 Mean   :5009073                   Mean   :   51.3
 3rd Qu.:7510762                   3rd Qu.:   52.0
 Max.   :9999861                   Max.   :2640.0
```

The difference we would analyze first after looking at the above two images would be the retention counts at the end of the first week or the **count of TRUE values in retention_7 column**. For Gate 30, they correspond to 8502 and for Gate 40 they correspond to 8279. What it would mean that 8502 players were retained when Gate 30 was assigned to them and 8279 players were retained when Gate 40 was assigned to them.

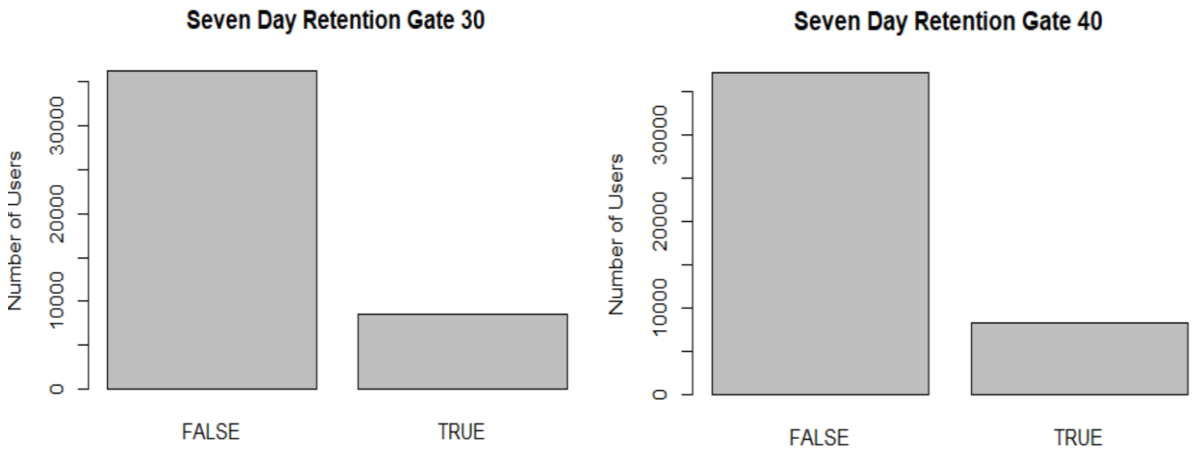
Of course, these numbers are misguiding as there is an imbalance in total records within both the gates. So, to correctly interpret it, we define our retention metric as follows –

NEXT PAGE

$$Retention\ Ratio = \frac{Retention\ TRUE\ values}{Total\ Values\ in\ Retention\ Column}$$

Based on the above formula, we compute that retention ratio of **Gate 30** is **0.19 or 19% of the players who were assigned gate 30 continued playing the game at the end of the first week.**

Similarly, the retention ratio for **Gate 40** was found as **0.182 or 18% of the players who were assigned gate 40 continued playing the game at the end of the first week.**



4. Approach:

We observe that there is a **difference of 0.8%** within the weekly retention of Gate 30 and Gate 40, with Gate 40 having lesser retentions. What we try to answer now is that if the difference is **significant or not?**

We would be constructing a **logistic regression model**, where we would try to probabilistically classify if a player would be retained at the end of 7 days or not, given the number of times the player has

played the game during the first week and the gate version he/she has been assigned to. In a nutshell we are trying to find the below probability -

$$P(Retention7 \mid GameRounds \cap GateVersion)$$

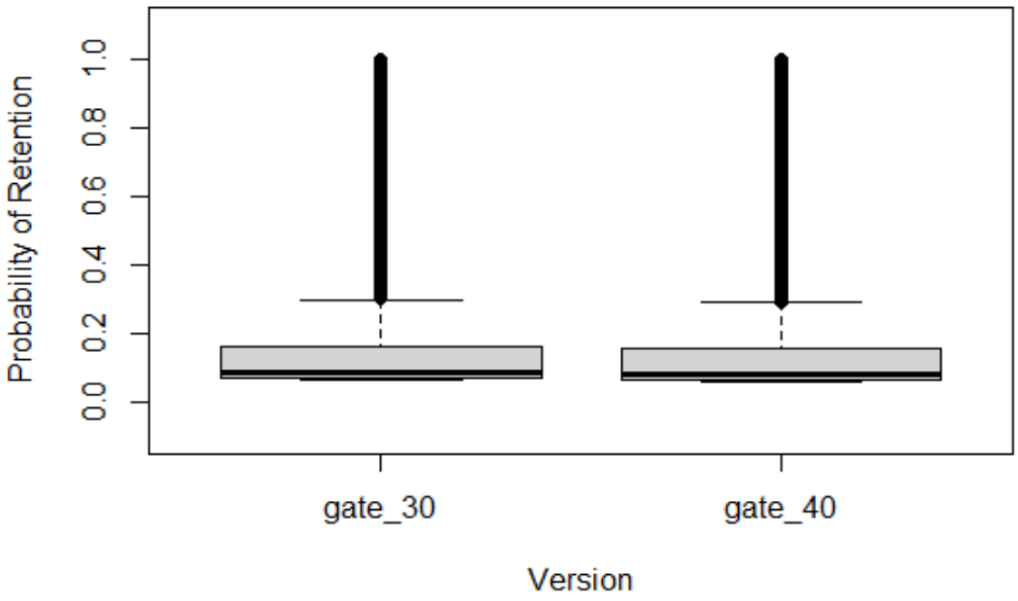
The coefficient estimate of each gate version would speak for the likelihood of retention, but with respect to the change from one gate version to other.

Apart from building an explanatory logistic regression model, we would also split the data into training and testing samples and build a predictive model which could help us predict in future if the player would be retained or not.

5. Interpretations and Results

```
Coefficients:
                Estimate Std. Error  z value Pr(>|z|)
(Intercept)    -2.6931956  0.0182019 -147.963  < 2e-16 ***
sum_gamerounds  0.0208641  0.0001817  114.843  < 2e-16 ***
versiongate_40 -0.0832847  0.0214180   -3.889 0.000101 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Version column had an input type ‘Factor’ while reading into R studio. The two levels of factors were Gate 30 and Gate 40. Now, the interpretation of the coefficient estimate for Version 40 in the above summary results of a logistic regression model is as follows – **moving to Gate 40 from Gate 30 will decrease the log odds of retaining a player by 0.0832. Which essentially would imply that the overall probability of retention would be reduced if we move from Gate 30 to Gate 40. Also, from above summary, version Gate 40 being a significant predictor, we could conclude that changes related to it could significantly impact the retention by the end of 7 days.**



Finally, we will summarize the results which we obtained from the predictive classification model. We set the threshold probability to 0.5, meaning that if the probability of retention is greater than 0.5 then prediction would be that the player is retained. If lesser than 0.5, it would be the other way round. The first predictive model has 2 predictors – sum_gamerounds and version. We observe that all the predictors are significant, and accuracy obtained is 74.89%. The performance of the model is evaluated using the test data. Lastly, we try adding another predictor to the model – retention_1, to see if the accuracy improves. The new accuracy is 77.84% and no improvement is observed over the previous model.

6. References

- Dr. Mendez's class notes & meetings!
- https://www.kaggle.com/yufengsui/mobile-games-ab-testing
- https://blog.minitab.com/blog/adventures-in-statistics-2/how-to-interpret-regression-analysis-results-p-values-and-coefficients