Data Analysis Take-Home Assigment

Tram Pham

08/31/2021

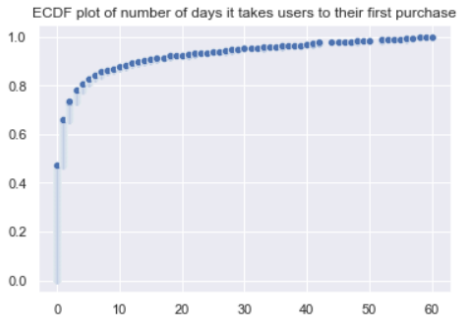
Data Quality Issues:

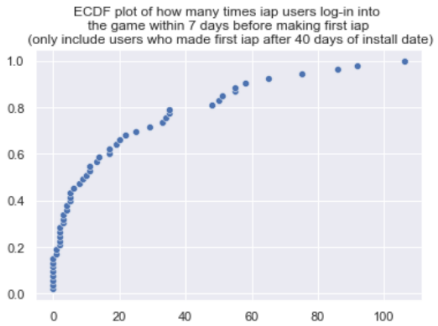
- Several users made in-app purchases, but those purchases were not recorded in the spendevent.csv
- Many non-paying users don't have any records in spendevents.csv, even though they are very active in the game (based on their session_num).
- A lot of duplicates in session.csv, iaps.csv, and spendevent.csv

Assumption: I assume we want to find potential groups at the **current time (i.e. '2019-05-06')** in the dataset.

- **Definition:** *active user:* users that play the game at least 1 time within the most 7 recent days (i.e. log in into the game from 2019-04-29 to 2019-05-06).

Problem 1: Return a group of potential users





Insights from the plots: ~90% of paying users made their first iap within the first 10 days of joining the game.

- Our potential group of users must be active within the most 10 recent days since users who are not active within 10 recent days are most likely to churn. 1733 non-paying users satisfy the condition.
- The most recent date in the data is 2019-05-06 and all the install_date starts from 2019-03-01 to 2019-03-07. Hence, all active users are in their 50-60th days of playing the game. We will look back 7 days since the last active day to observe their characteristic. We see that 60% of them have less than 50 gems, and around 40% of them play the game at least 20 times.
- Promotion group (for sample data): send a promotion to users who are still active within the last 10 days, i.e. (from 2019-04-29 to 2019-05-06), have less than 50 gems, and playing the game at least 15 times within 7 recent days. 309 users satisfy the above condition. (I extracted the group and put it at the end of "retention_model.ipynb")
- **Reason:** Because all users in our potential groups are within their 50-60th days of playing the game, we consider the paying users who made their first iap after 40 days in-game. We try to the pattern in paying users and choose non-payers whose patterns are similar to paying users. These paying users are very active in-game (40% of them play the game at least 20 times within 7 days), 60% of them have less than 50 gems before making a purchase, and 20% have from 50 to 75 gems. Still, they decided to buy more gems because they would spend them after purchasing them (e.g., user 5546) and have some gems left in the account. However, it is better to promote users who have a small number of gems and actively engage in the game.
- In our sample data, all the active users within the most recent 10 days are on their 50th day of the game. However, it is better target active users within the range of 7-30th day of their lifetime.
- We shouldn't send out the promotion within the first 6 days since that may help a portion of players to buy at a lower price than the price they are willing to pay for, but we may so do to keep several potential churn users to stay with us longer (there are 9k users quit the game after 6 days of installing)

Problem 2: Build a simple machine learning model to return the probability of user conversion.
As discussed in problem 1, we will remove all users who made a first iap within 7 days from the installed date since this group contains most users who will buy gems without promotion.
Modeling: (assume current time is '2019-05-06'. Recall that all the install_date starts from 2019-03-01 to 2019-03-07)

- In our sample data, all active people within the most recent 7 days are already on their 50-60th day of the game. The chance they convert to paying users is not high. However, I couldn't find any active users who are in the 7-30th days of their game. And we don't want to target non-active users since they already churn. Here is what I think is the best:

Train data (for our sample data):

- Take all the paying users who made their first purchase after 7 days → label 1 (239 users). For each of these users, the lookbacks window is 7 days before they made their first iap (we want to learn their behavior of paying users before they make a purchase). We will calculate features based on 7 days lookback window.
- For those who are not active within the most recent 10-25 days → label 0 (we chose so because those users were playing the game for a reasonable amount of time (they were in the game in around 40 days and they haven't made any iap and they churned). (total of 1518 users)

Test data:

- We will predict on those who are active within the last 7 days (all of them are in their 50-60th day of game) (total of 1733 users)

Alternative train data (general version):

- if we have more data, I think it is better to take all the non-paying users who haven't converted after 60 days as label 0 as well, since, after 60 days, it is a minimal chance they will convert. We will be targeting people in their 7 – 30th days of the game, have a small number of gems, and actively playing the game. Label 1 is from the paying users we recorded.

Model:

- Calculate features for each class
- Train a simple logistic model with class_weights and evaluate model

Feature Engineering:

- For label 1, we look back 7 days before the day users made their first iap.
- For label 0, we look back 7 before the last active day.
- Potential group: we look back 7 days since the most recent date. Use the model to predict the probability of this group.
- We will be building a very basic model. It contains a few features relating to the number of sessions, what time in the day users play the game, changes in the number of gems, number of certain spendtype, max chapter, how many days since they installed, unique story, etc.

```
evaluation(y_test, clf.predict(X_test))
```

```
Confusion Matrix for :
 [[144   8]
 [  9  15]]
=============
Accuracy    : 0.9034090909090909
Specificity :  0.9473684210526315
Sensitivity :  0.625
```

This model can be improved a lot by doing more EDA on data, build more features and tune model