Summary – KABAM Assignment

By Razzi Movassaghi

The question to be answered was to identify users that are likely to spend money in their game after finishing the tutorial. Three different sources of data were made available.

The major steps taken were**: Data Cleaning, Feature Engineering, Model Training and Tuning.**

An important issue was that the data is highly **imbalanced**. In other words, the number of users who spend some money (our positive class) is much smaller than the number of users who didn't spend any money.

Also, since the question asked was about whether a user will spend money or not (rather than how much one may spend), we treated it as a binary classification problem (we defined a new column called "spending_status" with 0 for no spending and 1 otherwise.)

Data Cleaning: After reading and combining data by the unique user identifier, we performed some data cleaning steps (dealing with NaNs, duplicates, outliers etc.).

The majority of missing data (NaNs) belonged to the feature that defined whether users completed the tutorial ('game_stats_tutorial_complete'). Since this is a key feature to filter the relevant data, samples with no values for this feature were dropped. More efforts can be done to impute data for this feature.

We also detected some outliers by looking at z-scores. However, we realized that while only around 9% of total data is detected as outliers, it includes around one-third of our valuable data in minority class (users with non-zero spending). Since we already had a very imbalanced data, we decided to keep the outliers. More investigation should be done regarding this issue.

Feature Engineering: First we looked at the numerical features, and removed the ones that were highly correlated to remove redundant information.

Then we looked at the categorical features, and chose some of them with reasonable number of unique values to generate dummies as new features.

This is by no means enough for feature engineering step that is one of the key steps of training a good model. However, we had to move on due to interest of time. More tests are needed to further improve on this issue.

Model Training and Tuning: First we tested a few models out of the box to see how they perform. The general steps were: Split data for training, validation and test, Over/Under Sampling of training data (as well as feature scaling if needed), and Evaluation.

Due to imbalanced nature of the data, we investigated precision, recall, and F1 scores to evaluate the models. Most of the chosen models performed similarly.

Then, for demonstration purposes. we chose one of the models (Random Forest classifier) to optimize some hyper-parameters. To avoid over-fitting, we performed a cross validation and studied the average score. As another example, we also optimized one of the parameters used in over-sampling step.

At the end, we evaluated our final model with optimized hyperparameters on the test data where some improvements were observed compared to the model with default values.

Overall, the model was able to predict the users who made a payment around 10 fold higher than the random expectation.

A better model can be built by doing further investigation.