An Exploratory Study in Survival Rates of Online Game Players

Yang Zhang

May 10, 2018

Introduction

With the advances in computer and internet, gaming online becomes a common part of everyday life. People play online games together with friends, or in most cases with strangers for entertainment. In fact, NBC News reported that there were mover 1 billion hours of League of Legends played in a month in 2012 [1]. That was only one game among many others with even more popularity.

As a result, we are interested in what makes people want to play the game. Or in other words, what makes a player decide to stay in community and keep playing. And what would make the player quit the game. One of the most intuitive approaches is to compare between games with different designs, and compare the user playing time. We can then investigate the effect of design factors and the interactions between them on a user's playing time. However, this would require reliable and uniformed sources from multiple games, which is beyond our reach. Instead, we focus on one game – Defense of the Ancients (DotA) 2, and compare between players. Specifically, we want to investigate what happened in the games that made a DotA player decided to not play the game anymore. Equivalently, it will also inform us what will make a player keep playing the game.

Data structure

Our data comes from the OpenDotA website, where we can get access to over a million DotA players data ever since their first game. We downloaded 35352 players' data (https://www.dropbox.com/s/cbraofsl18oavyr/download.tar.gz?dl=0). Each player has one json file which contains all games that player was ever involved. Each game has a data point in the json file, formatted as follows:


```
"match_id":3835021371,"player_slot":4,"radiant_win":false,"duration":2986,"game_mode":22,
"lobby_type":7,"hero_id":96,"start_time":1523822948,"version":null,"kills":0,"deaths":9,
"assists":19,"skill":null,"leaver_status":0,"party_size":null
```


Where we have the match id, the player's side, game result, game duration, game mode, lobby type, hero id, game start time, version, number of kills, number of deaths, number of assists, skill, leaver status, and party size. A full document of these data can be found here [2].

Analysis plan

To process the data, we first removed data from players who have played for shorter than two months. These players were often new to the game and thus were unstable and can be unpredictable when it comes to why they quitted the game. To better model the population, we removed data points from these players from the data set.

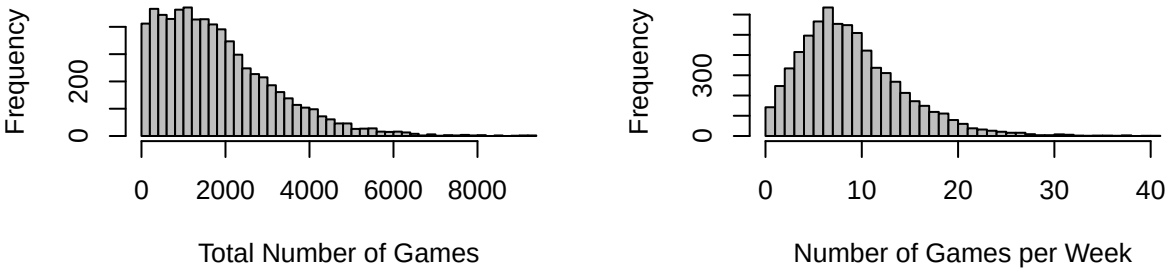
We then calculate the performance dependent variables – one set of variables for each player. Specifically, the performance dependent variables include average kills, average deaths, average assists, average game duration, average winning percentage, hero selection diversity (i.e., we counted the number of unrepeated heroes selected by that user), average game finish rate, game frequency (i.e., number of games per week), game history (i.e., total number of weeks playing DotA2), number of games (i.e., total number of games played), number of weeks not played since the last game.

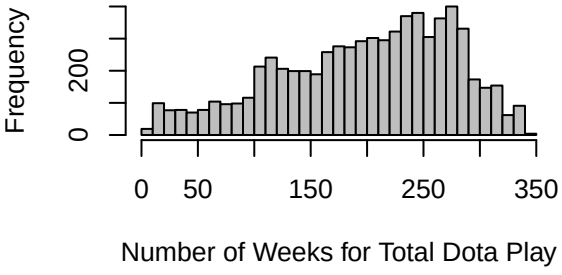
We also want to pay attention to that player's last period of game play before she quitted the game. For example, we calculated Z-scores of performance characteristics (e.g., kills/deaths/assists) of the player against her average performance throughout her entire gaming history. We also calculated the average performance characteristics of the player's last period of game play such as winning percentage, game finish rate, and game frequency. In our data processing, we defined the last week and the last month as the last period of game play. The above characteristics are calculated for the last week and the last month respectively.

Table 1. shows all dependent variables we calculated:

Symbol	Meaning
AverageKills	Average kill count in a game
AverageDeaths	Average death count in a game
AverageAssists	Average assist count in a game
AverageDuration	Average duration for a game (second)
AverageWinningPercentage	Average winning percentage
HeroSelectionDiversity	The total number of unrepeated heroes selected
GameFinishRate	Average game finish rate
GameHistory	The total number of weeks of game play
NumberOfGame	The total number of games played
NotPlaySinceLastGame	Number of weeks the player has not played since the last one
STDKills	The standard deviation of kill count in a game
STDDeaths	The standard deviation of death count in a game
STDAssists	The standard deviation of assist count in a game
STDDuration	The standard deviation of duration count in a game
LWZKills	Average z-scores of kill count in the last week
LWZDeaths	Average z-scores of death count in the last week
LWZAssists	Average z-scores of assist count in the last week
LWZDuration	Average z-scores of duration count in the last week
LMZKills	Average z-scores of kill count in the last month
LMZDeaths	Average z-scores of death count in the last month
LMZAssists	Average z-scores of assist count in the last month
LMZDuration	Average z-scores of duration count in the last month
LWGameFrequency	Number of games played in the last week
LWWinningPercentage	Winning percentage in the last week
LWGameFinishRate	Game finish rate in the last week
LMGameFrequency	Number of games played in the last month
LMWinningPercentage	Winning percentage in the last month
LMGameFinishRate	Game finish rate in the last month

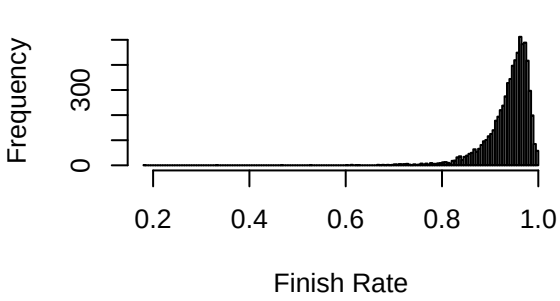
To tell the data summary story, we first plot histograms of the first 20% exploratory data. The histograms are plotted below.

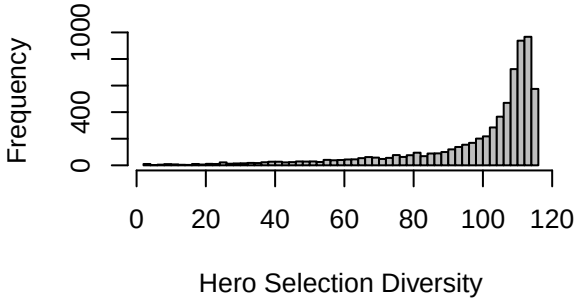


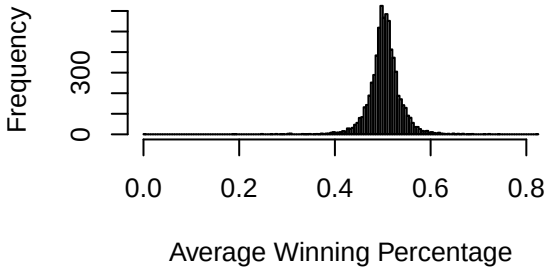


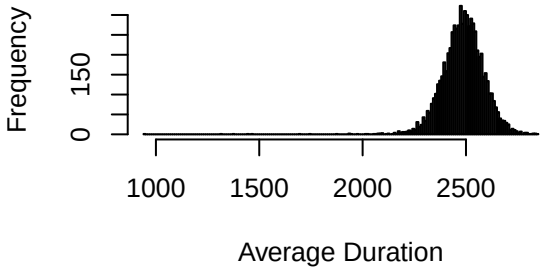
We can see from these plots that the total number of games played by each player and the number of games per week are more like normal distributions however with only one tail on the right. The mojority of players play 0-3000 games in our samples, while most of the players play 3-15 games every week. This indicates a log transformation might be needed for these dependent variables. However, for the number of weeks for the

total DotA play, we did not see a normal distribution.

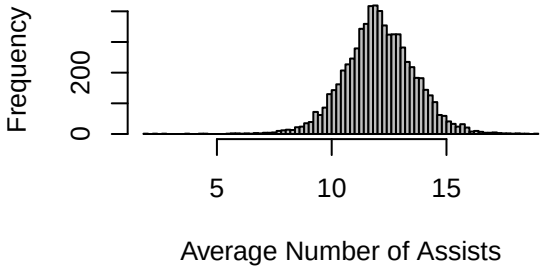


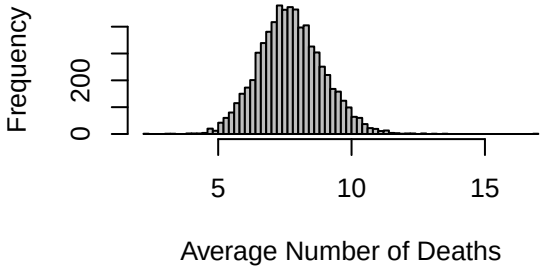


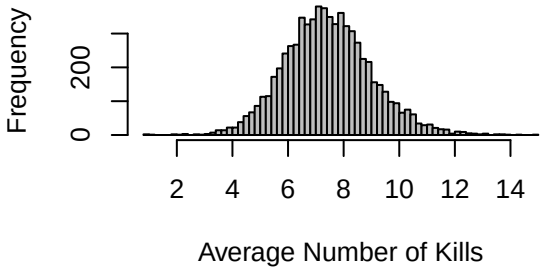


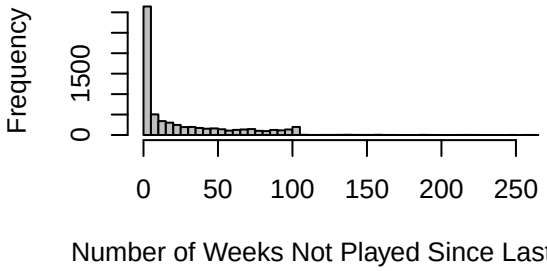


From the above histograms we can see clearly a normal distribution for the average winning percentage and average duration. In fact, the DotA2 platform automatically adjust the game difficulty for a player through e.g., matching the player with opponents with different levels to control the winning percentage to be 50%. The finish rate and the hero selection diversity have also normal distributions but the right tails are limited by the upper boundary. As we can see that most of the players have a game finish rate between 85% to 100%. And most of the players have used more than 100 heros during her game play history.









From the above plots we can clearly see normal distributions of game performance including average number of assists, deaths and kills. Most players have average number of assists ranging from 10-15, while 6-10 for deaths, and roughly the same for kills.

Since most of the player are still active in the game, so we see most players have not played for less than 5 weeks. We set a threshold to 48 weeks (i.e., 4 months). If a player has not played for longer than 4 months, she is counted as an inactive player. This threshold is selected based on experience, for e.g., an international travel or an internship might be roughly 3 months where a player might not be able to play during that period, but able to after. Inevitably, we have an imbalanced data set. We counted the number of active players and the number of inactive players in the exploritory data set and the result is shown in the following table:

Quitted?	Frequency
no	5345
yes	1535

To avoid having an imbalanced data set, we randomly downsampled the active player to match the number of the inactive players and move on to the next stage.

The most intuitive model is merely using winning percentage as the independent variable. A player might lose interests in the game simply because she lost to many games, which can be reflected in the winning percentage. So we first build logistic regressions based on average winning percentation, last week winning percentage, last month winning percentage, and the difference between average and last week winning percentage, and the diifference between average and last month winning percentation. We set a threshold of 50% on the the probability outputted by the model to predict if a player is inactive. We calculate the prediction accuracies which are show in the table below:

Model	Accuracy
average WP	0.5361564
last week WP	0.5166124
last month WP	0.5247557
average WP - last week WP	0.5123779
average WP - last month WP	0.5159609

This result indicates that the winning percentage along cannot well predict if a user is likely to quit the game since the prediction accuracy is almost the same as prior possiblity (i.e. 50%). We then select more independent variables, build the following model and evaluate their accuracies. Specifically, we build logistic regression models based on 1) only last period statistics, 2) only all period statistics, 3) combined statistics (all statistics). The prediction performance is shown in the table below:

Model	Accuracy
only last period stats	0.5625407
only all period stats	0.7925081
combined	0.8000000
combined and reduced	0.7990228

Since not all independent variables can explain the dependent variable well, we reduced the number of independent variables and selected the ones that are significant in the summary of the model using the combined all period stats and last period stats independent variable set. This model is shown as follows:

$$log(\frac{p(x)}{1-p(x)}) = \beta_0 + (\beta_1 * AverageAssists) + (\beta_2 * AverageDuration) + (\beta_3 * log(NumberOfGame) +$$

$$(\beta_4 * log(GameFrequency)) + (\beta_5 * GameHistory) + (\beta_6 * HeroSelectionDiversity) + (\beta_7 * STDKills) +$$

$$(\beta_8 * STDAssists) + (\beta_9 * LWZKills) + (\beta_{10} * LWZAssists) + (\beta_{11} * LMZAssists) + (\beta_{12} * DiffLMGameFrequency)$$

We also run a predicion evaluation using the exploritary data and the result is shown in the above table. As we can see the prediction accuracy of this model is roughtly as good as the all combined model. So we selected this model to move on to the validation phase.

```
folds <- rep(1:5, length.out = nrow(second_60))
set.seed <- 200
folds <- sample(folds)
accuracy <- c(0,0,0,0,0)

 for(i in 1:5){
    train <- filter(second_60, folds != i)

    down_train <- downSample(x = train[, colnames(train)],
                         y = factor(train$Active))

    test <- filter(second_60, folds == i)
    down_train$ActiveNumeric <- (down_train$Active == FALSE)*1
    test$ActiveNumeric <- (test$Active == FALSE)*1

    glm_all_reduced <- glm(ActiveNumeric ~ AverageAssists+ AverageDuration + log(NumberOfGame) + log(Ga

    valid_p1 <- predict(glm_all_reduced, type = "response", newdata = test)
    y_pred <- ifelse(valid_p1 > 0.5, FALSE, TRUE)
    y_act <- test$Active
    accuracy[i] <- mean(y_pred == y_act)
 }

  mean(accuracy)
```

```
## [1] 0.8121908
```

```
  sd(accuracy)
```

```
## [1] 0.005386877
```

The 5-fold cross-validation shows an average predication accuracy of 0.858 with a standard deviation of 0.005.

Finally, we combine the exploritory and the validation dataset to train a final model, while using the rest 20% of all data points as the test set. The result is shown as follows.

```
combined_80 <- rbind(first_20,second_60)

down_train <- downSample(x = combined_80[, colnames(combined_80)],
                         y = factor(combined_80$Active))
test <- third_20

down_train$ActiveNumeric <- (down_train$Active == FALSE)*1

test$ActiveNumeric <- (test$Active == FALSE)*1
```

```
glm_all_reduced <- glm(ActiveNumeric ~ AverageAssists+ AverageDuration + log(NumberOfGame) + log(GameFr

    test_pred <- predict(glm_all_reduced, type = "response", newdata = test)
    y_pred <- ifelse(test_pred > 0.5, FALSE, TRUE)
    y_act <- test$Active
    accuracy <- mean(y_pred == y_act)

    library(caret)
    library(e1071)
    confusionMatrix(data=table(y_pred, y_act))
```

```
## Confusion Matrix and Statistics
##
##        y_act
## y_pred  FALSE TRUE
##   FALSE  1162  985
##   TRUE    318 4414
##
##                Accuracy : 0.8106
##                  95% CI : (0.8011, 0.8198)
##     No Information Rate : 0.7849
##     P-Value [Acc > NIR] : 7.394e-08
##
##                   Kappa : 0.518
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7851
##             Specificity : 0.8176
##          Pos Pred Value : 0.5412
##          Neg Pred Value : 0.9328
##              Prevalence : 0.2151
##          Detection Rate : 0.1689
##    Detection Prevalence : 0.3121
##       Balanced Accuracy : 0.8013
##
##        'Positive' Class : FALSE
##
```

We achieved an accuracy of 0.816. We also calculated the confusion matrix and the result indicates an false positive rate of 0.213 where inactive players are predicted as active players. The false negative rate is 0.176 where active players are predicted as inactive players.

Reference

[1] NBC News. https://www.nbcnews.com/tech/tech-news/league-legends-players-log-1-billion-hours-month-flna1C6423906

[2] OpenDotA API Documentation https://docs.opendota.com/#