Forecasting mobile games' retention using Weka

Roxana Ioana STIRCU
Bucharest University of Economic Studies
roxana.stircu@gmail.com

Abstract: *In the actual market, when thousands of mobile, PC or console games are released every year, developing and publishing a successful and profitable game is a very challenging process. The gaming industry is very competitive, and all the distribution channels are full of projects competing for players. More and more companies are investing a lot of time and resources in developing an effective way to save and store all the data used and generated by their game's users. In order to develop effective and successful projects, companies adopted a lot of tools and techniques from other domains, like Statistics, Business Intelligence, or Project Management. The method most currently used is Analytics, defined as the process of discovering and communicating patterns in data, to better understand players' behavior, analyze their in-game interaction, and predicting their next in-game actions. This represents a huge step forward for the gaming industry, towards successful projects and user-tailored gaming experience. In this article the problem of users' retention is discussed, and a regression model is proposed in order to forecast players' retention, and prevent players from leaving the game.*
Keywords: *game analytics, metrics, user behavior, Weka, linear regression, forecast, players' retention.*

1 Introduction

Game Analytics was well documented and all the important concepts were defined by M. S. El-Nasr (editor) et al. [1], which covers a variety of analytics topics. The book is focused on behavioral telemetry and its role in the game development process and research. Authored by more than 50 experts in the field, the book covers data mining, visualization, monetization, and user research.

The process of user research is crucial in developing a game for success, and the first thing to mention is the frequency of the player's come backs in game, and the daily time spent playing. There is a direct connection between these two metrics and the following in-game activity of the player, as also mentioned in S. K. Hui paper [2] about gamer retention, defining retention as "a key input to gamer lifetime value".

In this paper is studied the retention and its properties, and a forecast of the next week values of retention is made. This forecast can be very useful on the long term, because developers are able to predict which users are about to quit the game, and take measures to prevent this and improve game experience.

2. Game analytics context

In the context of data analysis, a very important part is represented by the prediction models created using machine learning and forecasting methods, as described in [4] and [5]. Prediction represents the process of forecasting future values of a time series based on the known values, and are widely used in areas like financial markets, healthcare, marketing, social/products networks, military operations, or national economies. Prediction can be hard because of noise, or not having the right or enough data to train the model, but this problems can be resolved by using a moving average to smoothen the time series, and the data can be pre-processed and cleaned during this steps.

Machine learning represents the automatically learning process to make accurate predictions, based on previous observations (see [6] for details). The

process of classification (classify samples into predefined set of categories) can be represented as seen in *Figure 1*, listed below:

Fig. 1. The process of classification

Examples of classification problems can be given from a sort of areas, like natural-language processing, market segmentation, bioinformatics, face recognition, or text categorization. Among the advantages of solving problems using ML is the fact that machine learning algorithms are often more accurate than human-crafted rules, and are very flexible (can be applied to any learning task). One disadvantage is represented by the fact that is needed a lot of labeled data for the process of prediction to be more accurate.

The rules of data analysis can also be applied to data generated from video/mobile games, and this is mainly known as game analytics. The basic tools for game analysis are represented by KPI's (or Key Performance Indicators).

This are the basic metrics defined according to each game (please see [1] and [3] for detailed definitions and examples), and are the most important metrics to be tracked over time, like:

- DAU, abbreviation of Daily Active Users, defined as the total number of unique users that were active, calculated on a daily basis; active is defined as any action made in game, marked by a session – including only opening/closing the game;
- MAU, abbreviation of Monthly Active Users, defined similarly to DAU, the only difference is the fact that the total number of active users is calculated during an entire month;
- ARPU, abbreviation of Average Revenue Per User, defined as the average value of the game revenue for each user (calculated as the total revenue of the game divided by the total number of users); is also used a derivate of ARPU, named ARPPU (Average Revenue Per Paying User), calculated as the average value of the revenue only for the users that made at least one purchase in game;
- Retention, defined as the number of players that come back in the game, and it's calculated every day; most important values of retention are for the first day, the third day, and seventh day, and the thirtieth day;
- (Average) Session Length, defined as the (average) value of a player's session in game; a game session is defined by the moment when the player opens the game and the moment when he closes the game, and represents the amount of time between these two time stamps.

The most important KPI in this case is Retention, which represents the number of players that come back in game within a given timeframe. This value is crucial for any game developer, as it measures the rate of success over time for the game, according to sessions and number of players. A small value for retention indicates the fact that the players are not retained/do not come back in the game after the given period of time.

In the case of predicting a small retention value and identifying the users that are about to quit the game in the following days, some CRM campaigns or other type of player engaging actions should be taken into consideration, like:

- Promotions: represented by special offers and special prices of certain packages or boosters offered in the game's store; this can also represent an offer for a smaller price of the next month subscription or for the next mission (details are strictly defined according to the game characteristics);
- Newsletters and Invites: very important for the game features success, and is a good channel leading directly to the player's email inbox; usually via newsletter are sent articles about the game's users, development, new features, promotions, community, and invites to all the game related events;
- Rewards: this is one of the most effective ways to engage or re-engage a player in the game, and represents the process of sending rewards and boosters for free to certain users (usually this users are about to quit the game because they are having a hard time to passing a game level, and by receiving this reward they manage to advance in game).

3. Retention forecast

The first step in improving the game success by improving the retention rate (by campaigns) is to identify users that are about to quit the game (churn users), because can be a direct connection between decreasing user activity and churn rate.

To be able to use the information about churn users and predict retention for the following timeframe, one of the solutions can be to use machine learning based on historical tracking data. In this case we will be using historical data to train the defined model, and then apply the model on new data to obtain retention predictions and churn users.

The data used for this model should be extracted from a mobile game, using metrics defined according to each game. In this case we are interested in the following KPI:

- Sessions Per Day, defined as the total number of sessions a player has during a day, where sessions are defined as the total time between the moment when the player opens and closes the game;
- Time in game per day, defined as the total time a player spends in game during a day, and calculated as the sum of the sessions he had during that day, because these parameters represent best the retention of the players.

The data is automatically generated, using a Matlab function (*randi*). The values of sessions/day and time in game/day (in seconds), are created respecting the following conditions, displayed below in *Table 1*, where day 1 corresponds to the first day after the game installation day. In order to simulate real data generated from a game, we should use the known characteristics of users' data.

Table 1. Average Retention values

Day Number	(Average) Retention Range
Day 1	60-65%
Day 3	50-55%
Day 7	40-45%
Day 30	20%
Day 60	5%

We need to observe user behavior: for each user we have several time series, and the users with too few data should be ignored. The chosen metrics must explain the churn decision, in this case we defined daily numbers of sessions, and daily sessions length. The created model can be updated anytime with other metrics in order to improve performance.

For the case when the data is too noisy, we should use rolling sums over the last period of time or maybe moving averages, to eliminate the noise and made the model more accurate.

After filtering and grouping the data, we continue the process by creating the time series: the data is processed and converted to match the structure of an *.arff* file and saved with this extension, and the file is imported in Weka, to complete the preprocessing part (this is completed using the "Preprocess" tool from Weka).

To create the *.arff* file is used the structure described below, where the two attributes are defined: day, which is numeric and represents the day's number, and the time spent in game each of the days (numeric as well, counted in seconds). The data section is started with the mark "*@data*", and the attributes are listed below, separated by commas, one entry on each line of the file.

```
@relation UsersHistory

@attribute day numeric
@attribute time numeric

@data
1,4646
2,4509
3,4473
4,4357
5,4282
6,4254
7,4237
8,3767
9,3707
10,3541
...
```

We can visualize the data imported in Weka using the Visualize tool, after setting the colors, and other parameters for the selected chart. The values of daily time in game of the selected player is plotted as displayed in *Figure 2*, having the day's number displayed on OX (values from 1, representing the first day in game, to 42, representing the 42th day in game), and the total daily time spent in game, measured in seconds (values between 0 and 4865).

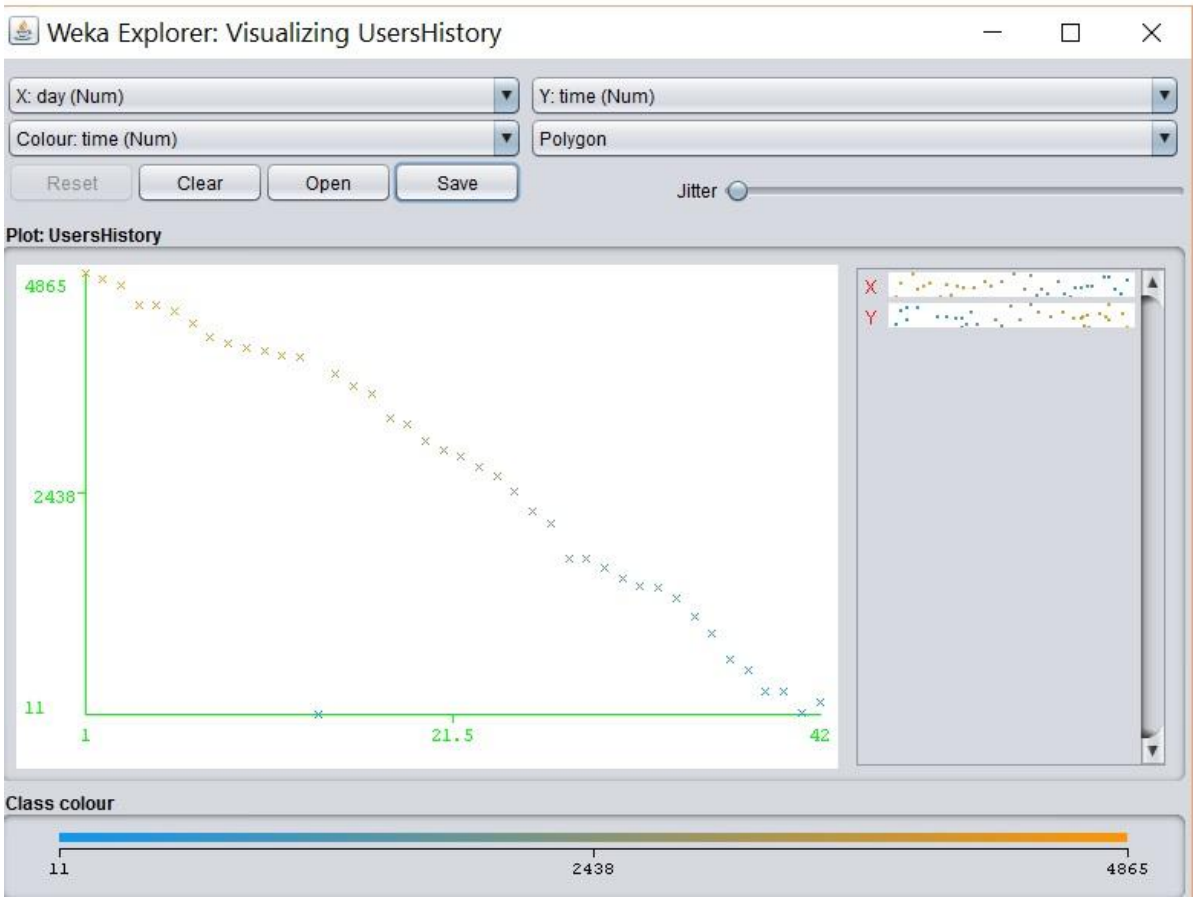

Fig. 2. Example of player retention during 6 weeks

For the forecast of the future values of "time in game" and retention, we use Weka's tool named "Forecast". In the first case the prediction is made using Linear

Regression as the base learner, but we also try time series forecasting with other different algorithms, for the first 6 weeks.
In order to test the forecasted values and measure the accuracy of the prediction, we compare the predicted values with the real values for the following 5 days. The forecast is made for a player with 6 weeks of history, including days with no game activity (this means that time spent in game equals 0), and the previous 4 days with no game activity at all.
The scheme used to forecast time in game is Linear Regression Model as presented in paper [4], defined below in the code section. The variable used for the regression model is the daily sessions length value, and is used to predict the expected values for the retention in the next 5 days. After choosing the desired model, *LinearRegression,* the data that it should use to build the model is loaded (the *arff* file described above), and we select "Use training set" – to specify that we want the desire model to be build based on the supplied training set. After this, we choose the dependent variable (the one we want to predict), which is the total daily time a player spends in game. The built regression model output and the predicted values are described below.

```
=== Run information ===
Scheme:
     LinearRegression -S 0 -R
1.0E-8 -num-decimal-places 4
-batch-size 1000
Lagged and derived variable
options:
     -F [time] -L 1 -M 7 -G
day -dayofweek -weekend
Relation:     UsersHistory
Instances:    42
Attributes:   2
              day
              time
Transformed training data:
              day
              time
              Lag_time-1
              Lag_time-2
              Lag_time-3
              Lag_time-4
              Lag_time-5
              Lag_time-6
              Lag_time-7
              day^2
              day^3
              day*Lag_time-1
              day*Lag_time-2
              day*Lag_time-3
              day*Lag_time-4
              day*Lag_time-5
              day*Lag_time-6
              day*Lag_time-7
time:
Linear Regression Model
time =
   -127.4628 * day +
     -0.2469 * Lag_time-5 +
      0.3209 * Lag_time-6 +
     -0.2378 * Lag_time-7 +
     -0.0099 * day*Lag_time-3
+
     -0.0107 * day*Lag_time-4
+
   6260.8175
```

The results of the forecast model described above are the predicted values for the next 5 days:

```
43*         620.361
44*          718.34
45*        321.7085
46*        115.7735
47*       -376.7138
```

Which means that the model forecasts, based on the history of the first 6 weeks, that the player may comeback in game and quit after 4 days.
We can improve this forecasting process by trying to use other learn algorithms, and compare the results afterwards for all the algorithms. Other way of improving the process can be done by speed up data processing, and by automating the prediction process.

4. Conclusions

This kind of model very useful to predict and prevent players churn – this thing is done in this case by selectively incentivize users that are about to quit the game. This characteristic is marked in this case by the low engagement of a player, represented in

the small values of retention vector. Using the predicted values, we can defined triggers, and automatically start campaigns and promotions, send gifts or rewards to avoid players churn, and to improve the game's success.

References

[1]. Magy Seif El-Nasr (Editor), Anders Drachen (Editor), Alessandro Canossa (Editor), "Game Analytics: Maximizing the Value of Player Data", Springer (2013).
[2]. Sam K. Hui, "Understanding Gamer Retention in Social Games using Aggregate DAU and MAU data: A Bayesian Data Augmentation Approach", online (2013).
[3]. Magy Seif El-Nasr, Alessandro Canossa, Anders Drachen ,"Chapter I2: Game Analytics – The Basics", online (2011).
[4]. Vasant Dhar, "Data Science and Prediction", Communications of the ACM, Vol. 56, No. 12 (2013).
[5]. John Langford, "Tutorial on Practical Prediction Theory for Classification", Journal of Machine Learning Research, 6, 273-306 (2005).
[6]. Rob Schapire, "Machine Learning Algorithms for Classification", online.

Ioana Roxana STIRCU, graduated the Faculty of Mathematics and Informatics of the University of Bucharest in 2010, and gained the Master title in Cryptography and Codes Theory at the same university