Wargaming: Analyzing WoT gamers’ behavior
Wargaming

And more!
Wargaming in the Czech Republic

Wargaming Czech Republic
Newest addition to the Wargaming Empire

Wargaming Prague
QA, Data Warehouse, Distributed Development, Back office support, Global Procurement and Legal services

Wargaming Brno
Prototyping and development of World of Tanks gameplay

Friendly and fun environment
Collaborate on cutting edge projects with colleagues from around the globe
Exceptional opportunities for professional growth
BI in the product lifecycle

Strategic Intelligence

User Research

WGAN Support Analytics

Publishing Analytics

Data Science

Competitive Intelligence

New Game Analytics

Proof of concept / MVP

Telemetry

Alpha

Beta

Events #N

Patch #N

Idea

Launch

Live Operations

Data Services
Core focus of Data Science

- Develop models and algorithms in support of all BI functions
- Support regional publishing and game analysts with complex product analyses
- Support Player Relationship Management Globally
- Explore new technologies, methodologies, and develop new tools
How data science supports each BI team

- **Strategic Intelligence**
  - Telemetry Input
  - Feature Analysis
  - Life Cycle analysis

- **User Research**
  - Player Satisfaction
  - User Profiling
  - PRMP Surveys

- **Game Support**
  - Cheat/Bot Detection
  - User Segmentation
  - Progression Models

- **Publishing Support**
  - CRM Support
  - CS Models
  - LTV Models
Data science tools

- Engineered Solution
- Massive Parallelization
- Model Management
- Limited Algorithms

- GPU Processing
- Training Parallelization
- No model management
- Cutting-edge Algorithms

- Open Source Solution
- In-depth Customization
- Limited Model Management
- Cutting-edge Algorithms
Analyzing player behavior in non-contractual settings

• How many games is each user expected to play in the next 90 days/3 months?

• Translate RFM measures to the gaming domain –
  • Recency: When was the last time user A played the game?
  • Frequency: How often does user A play the game?
  • Monetary / Intensity: Every time user A has a game play session, how many games on average does he/she play?
Analyzing player behavior in non-contractual settings

- The “Buy till you die” (BTYD) family of models
  - Probabilistic models for user behavior in non-contractual settings
  - Developed by the marketing research community
  - Common theme: Recurrent survival model which allows users to churn from the process


The “Buy Till You Die” family of models

- Model how long a user is active
  - Dropout rate parameter $\mu_i$
- Model how many sessions and games the user plays while he/she is active
  - Playing rate parameter $\lambda_i$
- Many different variations based upon different choices of modeling playing and dropout behavior
The “Buy Till You Die” family of models

- Works with just activity log data
  - Privacy concerns, users don’t want to share personal information, users misreport information, registration process should be less intrusive.
- Parameters determined using maximum likelihood estimation
- Useful for summarizing and predicting population level trends
Predicting user behavior with BTYD

- Dataset consists of all EU region players joining between 1st Feb 2016 and 1st May 2016

- All users’ games, for each day, are tracked till 1st May 2017
  - Number of users = 331,811
  - Number of games = 206,897,542

- Data from 1st Feb 2016 – 31st Jan 2017 was used to predict number of games each user will play in the next 90 days
Predicting user behavior with BTYD

• RFM based features prepared for each user
  • Frequency: Number of active days
  • Recency: Number of days since last login
  • Intensity: Number of games played
  • Tenure: Number of days between first and last active days
Predicting user behavior with BTYD

- Total number of games played predicted by BTYD: 28,852,874
- Total number of games actually played: 31,329,849
Individual predictions with BTYD

- *Whales*: Few users having a very large number of games
- Exclude top 10% players based on number of games played
  - Total number of users = 300,399 (90%)
  - Total number of games played = 68,834,316 (33.27%)
- *Pareto principle*
Error Metrics for evaluating individual predictions

- **Root mean squared error (RMS Error)** = Square root of the mean of squared error
  - Squared error for a given user \( u_i \) is \( e_i = (y_i - \hat{y}_i)^2 \)

- **Mean absolute error (ABS Error)** = Mean of absolute error
  - Absolute Error for a given user \( u_i \) is \( e_i = |y_i - \hat{y}_i| \)

- **Mean relative absolute error (Rel. ABS Error)** = Mean of relative absolute error
  - Relative absolute error for a given user \( u_i \) is \( e_i = \frac{|y_i - \hat{y}_i|}{y_i} \)
  - But \( y_i = 0 \) is an issue and so we use \( e_i = \frac{|y_i - \hat{y}_i|}{\max(1, y_i)} \)
Predicting user behavior with BTYD
Machine learning for predicting user behavior

- **Gradient Boosting**
  - State of the art before Deep Networks, still continue to be one of the best machine learning techniques for learning from data
  - Used to win several data science competitions/challenges

- **Core idea**
  - Train a model on data
  - Train another model which learns where the first one makes mistakes
  - Train another model which learns where the combined model makes mistakes
  - Keep repeating the previous step!


Predicting user behavior with Gradient Boosting

Individual predictions using GrBoosted model
BTYD vs Gradient boosting for predicting number of games played

- Does significantly better than BTYD on the individual predictions but loses some of population level properties captured by BTYD

<table>
<thead>
<tr>
<th></th>
<th>BTYD</th>
<th>GrBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS Error</td>
<td>80.57</td>
<td>12.27</td>
</tr>
<tr>
<td>Mean ABS Error</td>
<td>15.09</td>
<td>5.78</td>
</tr>
<tr>
<td>Mean Rel. ABS Error</td>
<td>5.38</td>
<td>3.52</td>
</tr>
</tbody>
</table>
Comparing response variable distributions – Gradient Boosting

Log transformed response variable distribution from BTYD and GrBoost vs Actual
Combining Gradient Boosting and BTYD

- Boosting fixes the larger errors in BTYD predictions but does so at the cost of overestimating the number of users having fewer games.
- BTYD is better at capturing the overall distribution of the response variable but does poorly on individual estimations.
- Can we have the best of both worlds?
Ensemble of Gradient Boosting and BTYD

RFM data from user $u_i$ → Gradient Boosting Model → Decision Tree based prediction selector → Regularized Gradient boosting → Gradient Boost Prediction for user $u_i$ → BTYD Model → BTYD Prediction for user $u_i$
Combining Gradient Boosting and BTYD

• Learn a decision tree on when to use BTYD predictions and when to use boosting predictions!
  • *RegGrBoost*: Regularizing gradient boosted predictions with BTYD predictions

• For each user use decision tree model to decide whether to use BTYD prediction or Gradient Boosted prediction!
Regularized Gradient boosting for predicting number of games played

• Improved predictions on users having fewer number of games played

<table>
<thead>
<tr>
<th></th>
<th>BTYD</th>
<th>GrBoost</th>
<th>Reg-GrBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS Error</td>
<td>80.57</td>
<td>12.27</td>
<td>12.13 (98.86%)</td>
</tr>
<tr>
<td>Mean ABS Error</td>
<td>15.09</td>
<td>5.78</td>
<td>3.91 (67.65%)</td>
</tr>
<tr>
<td>Mean Rel. ABS Error</td>
<td>5.38</td>
<td>3.52</td>
<td>1.47 (41.76%)</td>
</tr>
</tbody>
</table>
Comparing response variable distributions – GrBoost vs RegGrBoost
Log Rel. ABS error quantile plot for all methods

Response variable error quantiles from different methods

Method
- BTYD
- GrBoost
- RegGrBoost
Log squared error quantile plot for all methods

Response variable error quantiles from different methods
Conclusions and Future directions

• BTYD and Gradient boosting combined to produce a model improving on both
  • Decision tree model used to learn whether to use the response from gradient boosting or the average user behavior predicted by BTYD
  • Can it be applied to other scenarios?
    • Explore from a Machine Learning theory perspective
• Largest errors due to *Winback* phenomenon
Conclusions and Future directions

• Plan to use this in support of our new mobile products
• Work with publishing and CRM teams for ROI based evaluation
• Other successful models developed in the past which have been empirically verified to provide ROI lifts for WoT, WoWS and other games
THANK YOU!

• www.wargaming.com

• Questions/Answers
Appendix I – Details on the “Buy Till You Die” family of models

• **Pareto/NBD Model**
  • While customer is active, number of transactions in time $t \sim \text{Poisson}(\lambda_i t)$
  • Transaction rate for customers $\lambda_i \sim \text{Gamma}(r, \alpha)$
  • Each customer has an unobserved lifetime $\tau_i \sim \text{Exponential}(\mu_i)$
  • Dropout rate for customers $\mu_i \sim \text{Gamma}(s, \beta)$
  • Transaction and dropout rates vary independently across customers

• **Gamma-Gamma** spending model to estimate expected spend per transaction

• Many different variations based on different choices of modeling transaction and dropout behavior
Appendix II - Gradient boosting for predicting number of games played

• Training data from 1\textsuperscript{st} Feb 2016 – 31\textsuperscript{st} Jan 2017 further split into two parts-
  • ML Training data from 1\textsuperscript{st} Feb 2016 – 1\textsuperscript{st} Nov 2016
  • ML validation data from 2\textsuperscript{nd} Nov 2016 – 31\textsuperscript{st} Jan 2017

• For each user
  • Prepare RFM based features from ML Training data
  • Use corresponding number of games played in ML validation data as response

• Use to train gradient boosted regression tree model
  • Measure prediction errors on test period 1\textsuperscript{st} Feb 2017 – 1\textsuperscript{st} May 2017
Appendix III - Regularized Gradient Boosting predictions

Individual predictions using RegGrBoosted model
Appendix IV – Decision Tree for choosing prediction model (0: BTYD, 1: Gradient Boost)