Churn Prediction for Game Industry Based on Cohort Classification Ensemble

Evgenii Tsymbalov[1,2]

[1] National Research University Higher School of Economics, Moscow, Russia
[2] Webgames, Moscow, Russia
etsymbalov@gmail.com
http://corpwebgames.com/en/

Abstract. In this paper, we present a cohort-based classification approach to the churn prediction for social on-line games. The original metric is proposed and tested on real data showing a good increase in revenue by churn preventing. The core of the approach contains such components as tree-based ensemble classifiers and threshold optimization by decision boundary.

Keywords: Churn prediction, ensemble classification, cohort-based prediction, on-line games, game analytics

1 Introduction

The churn prediction is a real problem, which can be found in businesses that deal with a permanent stream of customers using subscription services: banking [1,2], telecommunication [3], and entertainment industries, with increasing popularity of game analytics (e.g. [4]). The focus of business development shifts from the attraction of new customers to retention of the old ones. Therefore modeling users' outflow can be used to plan company's tactics and strategies. However, it is often important to not simply know the outflow indicators on a macro level, but to predict the churn probability for every customer to use the spot interventions.

The definition of churn and the churners varies depending on a specific problem. Here we define the churners as users who were absent for 30 days and more. Moreover, in this research we are interested in users who were absent for at least three days and made at least one transaction. Such restrictions are motivated by the low revenue of the late returners and recommendations from social platforms, e.g., Facebook does not recommend to send out notifications to the players with more than 28 days of absence.

In this paper, we propose a cohort-based classification approach to the churn prediction for social on-line games. Our data processing pipeline includes feature engineering and selection along with optimization of specific metrics. Thus, we introduce a meta-metric, which penalizes undesired outcomes of the cohort test procedure; it is designed to reflect real-life experience of using the prediction models. All the steps of the model training pipeline include optimization of classifiers; the final step optimizes the whole ensemble on meta-metric values.

In Section 2, we introduce our cohort-based ensemble method for churn prediction. In Section 3, we provide the reader with the results of experimental evaluation. Section 4 concludes the paper.

2 Proposed Method

In a few words, the churn prediction problem can be stated as follows: given the data about users, who recently stopped playing, predict whether a particular user will abandon the game or not. This information is used for sending app-to-user notifications to those users, who are likely to stop playing our game at all. These notifications also offer some in-game goods to the user, usually some in-game currency, rare valuable items or discounts.

2.1 Meta-metric

This problem is rather different from the standard data mining classification problem due to several evaluation requirements, which result in so-called "cohort test". It is defined as follows (see Fig. 1 for short description):

1. We take the cohort of players who had been absent for 3 days by the given date D.
2. A model, that solves the classification problem for a given day of absence, predicts for every player, whether or not he or she becomes a churner.
3. Those, who were predicted as churners (both true and false churners), are eliminated from the next steps.
4. We take the cohort of players who had been absent have been absent for 4 days by the given date $(D+1)$ and had not been eliminated in the previous part.
5. Steps 2, 3, 4 are repeated until the date $(D + 29)$ or no more users left in cohort.
6. We evaluate the meta-metric, using the information about successful and unsuccessful predictions at each step as well as users left after Step 5 (if there are any).

Final challenges are connected with the result of the cohort test and its influence on the game balance:

- We do not want to send prizes (bonuses) to people who are likely to return by themselves, because it may ruin the game balance.
- We want the players to be returned as soon as possible. The probability of the player's future transactions as well as the average revenue dramatically decreases with the growing number of days the player is absent for.
- We assume that we should not send notifications to the same player twice.

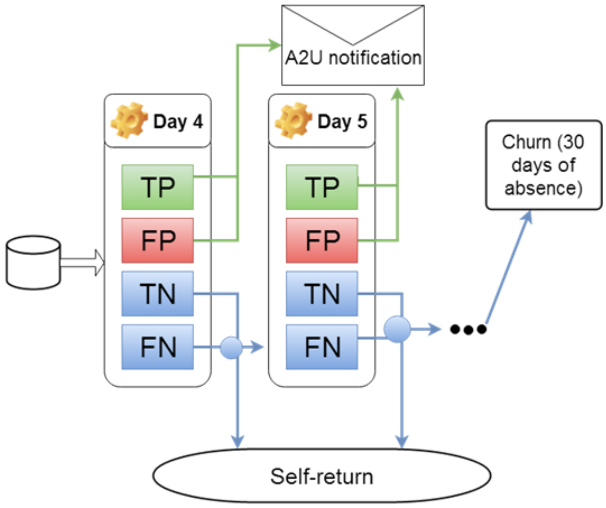

Fig. 1. The cohort test evaluation scheme. The users, defined by a classification algorithm as churners, are provided with app-to-user notification, while those users who remain go to the next step of the algorithm according to Eq. (2).

To answer these requirements, the cohort-based meta-metric is introduced:

$$MM = \sum_{day=4\ldots 26} (\gamma^{day-4} TP_{day} - \alpha FP_{day}) - \beta(TP_{27} + FP_{27}), 0 < \gamma \leq 1 \quad (1)$$

This metric can be interpreted as a weighted number of returned players, with the reward for early return and the penalty for type I error (marking a user who will return as a churner). Note that different goods offered to a user with notifications, as well as various social platforms and projects require different parameters for the meta-metric.

The question of determining coefficients α, β, γ for a given project should be discussed at both levels, analytical and managerial. While α could be estimated by approximating the probability of a player being a payer after he or she returns, β and γ could be estimated by experts (like we do in Webgames now) or based on the data from the previous experiments (which is unavailable for this research).

2.2 Problem statement

Let $C^D_{date} = (P_1, \ldots, P_{N_{date}})$ be data of D-th day cohort of players on date *date*, where $D = 4 \ldots 26$, $P_j \in \mathbb{R}^{|F| \times D}$ are all player's features up to day $(D + date)$, F is the set of player's features at a particular day.

For every user in a cohort, our algorithm A makes a decision S: on which day $D = 4 \ldots 26$ send out the notification (or not to send, then $D = 27$):

$$A : (C^j_{date})_{j=4\ldots 26} \longmapsto S = S(D_1, \ldots, D_N); \quad (2)$$

Our goal is to find the decision, which maximizes the meta-metric (1):

$$S = \arg\max_{S'} MM((C^j_{date})_{j=4\ldots26}, S') \tag{3}$$

2.3 Ensemble of classifiers

To classify a user on a given day, we use the set of classifiers $\{C_4, C_5, \ldots, C_{26}\}$, one for the data for each day the user absents. So the classifier C_4 is trained on the users who have been absent for 3 days and evaluated on the 4-th day of absence; the classifier C_5 is trained on the users who have been absent for 4 days, etc. This results in total of 23 classifiers.

2.4 Optimization and greedy solution

Since the main problem requires the complex elimination procedure, therefore it seems to be complicated (if even possible) to optimize the meta-metric analytically. Instead, we use a greedy approach, with models for solving Problem (3) optimized by the cohort ensemble classifiers parameters in terms of ROC AUC metric, and these classifiers thresholds (decision boundaries) are optimized by the meta-metric.

3 Experiments

3.1 Data description

We used data from the Ghost Tales project on Facebook during the period 11.2015 – 03.2016, with the last month as a test subset. Hereby the features described below are mostly game-independent (for free-to-play games), so this feature set could be used for other projects as well.

Here we treat a player on his/her different days of absence as different players, which allows us to increase the number of objects in the dataset by more than 10 times, resulting in 2.7M records in total.

Typically, features could be divided into three main groups:

- Personal features: year of birth, country, etc.
- Behavioral features: the total number of active days for a player, the number of days a player is absent, the number of completed quests, etc.
- Transactional features (based on users payments).

3.2 Data processing

Our data processing includes the following steps:

1. Feature engineering. We have calculated pairwise ratios between all the features with the same measurement units (e.g. number of days with payments divided by total days for a given player) and added some other feature combinations based on expert knowledge.

2. Deletion of low variance features.
3. Selection of the best features based on F-test.
   The number of the best features selected may vary for different base classifiers. Here we used Extra Random Forest classifier [5] and logistic regression. We have also tried to use Gradient Boosting algorithms (XGBoost [6] and AdaBoost), but these classifiers shows similar or even worse results and do not support effective parallelization. We do not use algorithms based on neural nets due to their long training time, which is undesirable for weekly model retraining.

3.3 Single classifier optimization

We have performed 10-fold cross-validation procedure on the training set with optimization by ROC AUC score. The parameters used during the grid search are summarised below.

The list of ensemble trees' parameters (taken according to the real-life resource constraints):

- number of trees in ensemble: 50 – 500;
- tree depth: 5 – 20;
- minimal sample split: 2 – 15.

Logistic regression's parameters:

- penalty type: l1 or l2;
- C (regularization constant): 0.1 – 300.

The performance summary for selected values of the parameters of Extra Random Forest classifier is given in Table 1.

Table 1. Optimized parameters and classification metrics on the test set for the extra trees based classifier for selected days.

Day	Number of trees	Tree depth	Minimal sample split	Precision	Recall	F1-score
4	500	10	6	0.73	0.75	0.74
7	500	10	10	0.66	0.66	0.66
10	300	15	6	0.64	0.65	0
14	200	15	6	0.68	0.7	0.69
18	500	15	10	0.63	0.79	0.7

3.4 Ensemble optimization

The simplest method to continue optimization is to set decision boundaries (thresholds) for all the classifiers to the same value (which can be found via cross-validation, for example). However, it is easy to see that we should reduce the decision boundaries of classifiers for several first days, because the users, who returned on the first days, have more impact on the meta-metric. Since 23-parameter optimization (for every classifier) is a computationally exhaustive problem, we reduced the number of parameters by considering the threshold as a function of day. We optimized the thresholds for different types of such functions:

- linear: $threshold = A \cdot day + B$;
- exponential: $threshold = A \cdot exp(B \cdot day)$;
- quadratic: $threshold = A \cdot day^2 + B \cdot day + C$ (where A, B, and C are constants)

using differential evolution as a global optimization algorithm. However, the optimization on various cohorts shows that exponential and quadratic approximations tends to reduce to linear, see Table 2.

Table 2. Optimal parameters found by differential evolution algorithm for the cohort of users absent from 2016-02-21.

Approximation function	A	B	C
Linear	0.4921	0.0141	-
Exponential	0.5416	0.0207	-
Quadratic	0.0002	0.0138	0.5344

The optimal parameters found by linear optimization results in 3-7% increase of the meta-metric, compared to the optimized parameters for the constant decision boundary, see Fig. 2.

4 Conclusion

In this research, we have proposed a cohort-based ensemble model for the churn prediction. The final part of this model includes optimization of the original meta-metric, which is designed to reflect the real-life experience in usage of prediction models that rely on a cohort test; other steps are also constructed by taking real-life resource constraints into account. Various numerical experiments show the importance of the steps used in the model training pipeline.

During this research, the algorithm for the weekly model evaluation has been developed and implemented in the Webgames company. Mostly automated, this algorithm requires human assistance only at the step of the meta-metric parameters' choice.

We assume it can be used in various areas for churn prediction. Thus, the obtained results form the groundwork for model improvement and generalization in other areas such as telecommunication and banking industries.

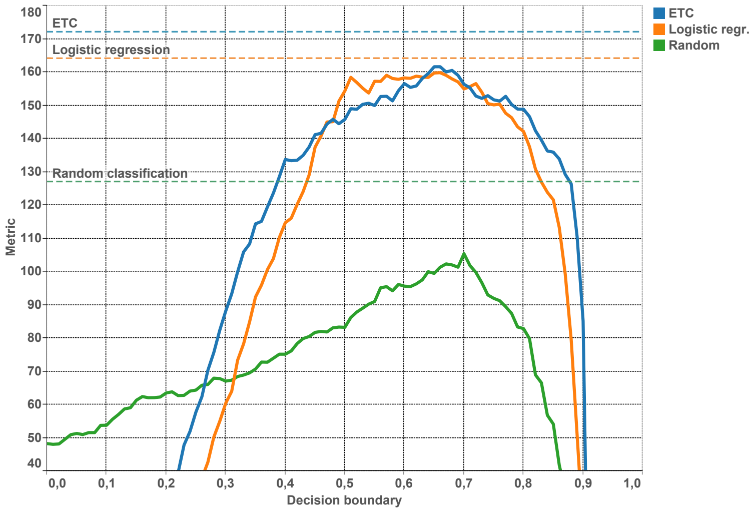

Fig. 2. Meta-metric values on α=0.8, β=1.2, γ=0.87. (The Ghost Tales on Facebook with diamonds prize) for the cohort absent from 2016-02-27. The solid line shows the meta-metric value for the case of the constant threshold, the dashed lines correspond to linearly optimized values of the metric.

Acknowledgment. We would like to thank Webgames' Analytics department for the provided data and feedback. We are also grateful to Mehdi Kaytoue from INSA-Lyon (France), and Evgeniy Sokolov and Peter Romov from Yandex Data Factory (Moscow, Russia) for their advice. Last but not least we would like to thank Dmitry Ignatov (HSE, Moscow) for his supervision.

References

1. Nie, G., Rowe, W., Zhang, L., Tian, Y., Shi, Y.: Credit card churn forecasting by logistic regression and decision tree. Expert Systems with Applications **38**(12) (2011) 15273–15285
2. Mutanen, T., Ahola, J., Nousiainen, S.: Customer churn prediction-a case study in retail banking. In: Proc. of ECML/PKDD Workshop on Practical Data Mining. (2006) 13–19
3. Ahn, J.H., Han, S.P., Lee, Y.S.: Customer churn analysis: Churn determinants and mediation effects of partial defection in the Korean mobile telecommunications service industry. Telecommunications policy **30**(10) (2006) 552–568
4. Bosc, G., Kaytoue-Uberall, M., Raïssi, C., Boulicaut, J., Tan, P.: Mining balanced sequential patterns in RTS games. In: ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic. (2014) 975–976
5. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine learning **63**(1) (2006) 3–42
6. Chen, T., He, T.: XGBoost: eXtreme Gradient Boosting. R package version 0.4-2 (2015)