Feature Article

User Modeling for Churn Prediction in E-Commerce

Patrik Berger and Michal Kompan
Faculty of Informatics and Information Technologies, Slovak University of Technology

***Abstract*—In the domain of e-commerce, acquiring a new customer is generally more expensive than keeping the existing ones. A successful prediction of churn of a specific customer provides an opportunity to change his/her decision to leave. In this paper, we propose a novel complex user model focused on the user churn intent prediction. The idea of our model is based on composing of multiple sets of features representing user's interaction with the web application. The performance of our model is evaluated indirectly by the prediction of churn, using real data from online retailers. The results show that the prediction using the proposed model outperforms the churn prediction based on baseline models across two domains.**

The recent growth of the market and the technology advancement led to the increased number of competitors providing online services to its users. Let the users who generate profit for a retailer by using the application be called customers. Since the number of paying customers is one of the key metrics of a success of an online retailer, maintaining user base is a widely adopted process. According to F. Reichheld, increasing the customer retention in financial services by 5% produced more than 25% profit increase.[1] Several approaches to the retention increase are used. Based on the time span of strategy of retaining customers, we differentiate between long-term and short-term approach.

The long-term strategy focuses mostly on improving provided service and consists of four steps:[2] data collection, analysis, recommendations, and action. The analysis is usually done by online store's business analyst. His/her task is to create recommendations based on results of the analysis. Those recommendations are then passed to the development team, which changes the web-application accordingly.

While the long-term approach is oriented on an improvement of the retailer's web-application and

Digital Object Identifier 10.1109/MIS.2019.2895788
Date of publication 28 January 2019; date of current version 3 May 2019.

business that is behind it, from the short-term perspective, the main task is to identify users that are most probably about to churn. Commonly used approach of predicting the customer's churn is to first identify the set of attributes describing him/her and consequently modeling his/her behavior—a user model. The user model is according to Brusilovsky defined as a set of information connected with the user behavior, attitudes, and stereotypes.[3] Depending on the goal for which the user model is build, it should consider user's preferences, interests, goals, knowledge, interaction history, etc..[4] The user modeling is usually researched from the system personalization point of view, while long- and short-term models were proposed in the literature.[5] The user model can then be used as a base for prediction using machine learning methods.[6]

Problem Statement: Most of state-of-the-art works that studied the user churn, defined the churn either as ending a session or quitting the service across the sessions. Both of those approaches tend to employ a specific user and prediction models to predict whether a user will churn. Generally, the prediction of a session end needs to be conducted in real time or at last online time; thus, the prediction model needs to be fast. To achieve the speed and a reasonable accuracy, it requires a user model capable to capture dynamic behavior aspects.[7]

In this paper, we focus on the user churn across multiple sessions. Given the data from user's past k sessions, we try to determine whether he/she will remain retained for the $(k+1)$ th session. In other words, the model is evaluated by the binary classification task, which answers the question: "Is this the last user session?"

Main contributions presented in this paper are as follows:

- scalable user model capturing customer's behavior from the churn intent perspective.
- application of the model in the user churn prediction task.
- evaluation of the model on real-world e-commerce datasets.

RELATED WORK

User churn is the subject of study across multiple domains. Churn of donors in crowd-funding platform for education projects DonorsChoose.org was studied.[8] The authors considered a donor churner after not making any donation during a period of one year. They created a complex model composed of four categories of attributes describing: time, donor, project, and the teacher responsible for the project. The performance of complex model, using all categories of attributes, was compared to models using the categories individually employing Area Under Receiver Curve (ROC AUC) metric. Their results show that the complex model, achieving 74% ROC AUC, outperformed simple models. The most important category of attributes turned out to be the one describing the donor.

A usage of multiple time-based attributes was analyzed in detail[9]. The subject of the study was an web-application recommending the users' websites that they might like. They found out that the churn of the users was not connected to all the recommended sites. In fact, it was mainly influenced by only a few that interested or did not interest them the most. As a result, the most accurate model, achieving 71% prediction accuracy, only used time attributes related particularly to those sites.

The retention of users of Yahoo Answers was studied.[6] The authors focused on new users during their first days in the application. Their goal was to predict their churn during the first week of usage of the application. They proposed a complex model composed of four categories of attributes that described: the question that the user responded to, the provided answer, the feedback to the answer, and the demographics of the user. Authors used the model to compare the accuracy of the churn prediction achieved by multiple machine learning methods. Although the random forest achieved the most accurate prediction (75.8% ROC AUC), the authors preferred faster linear regression with comparable results.

Multiple works studying the user churn prediction focused particularly on practical problems that occur across application domains. Lazarov and Capota[10] provide comparison of multiple machine learning methods that are commonly used to predict the churn. Their work confirms the superiority of neural networks in this area. Abbasimehr *et al.*[11] simple sampling techniques achieved better results compared to more sophisticated approaches of balancing the dataset.

Several approaches considering multiple user action types were proposed for the personalization or classification tasks, e.g., collective matrix factorization.[12,13] However, the interpretability of such latent approaches is quite low. Moreover, it is beneficial to predict a user churn within his/her actual session (to be able to take actions) and thus the performance issues are extremely important.

To sum up, user models including various attributes sets seem to perform better in the context of churn prediction tasks.[14] These sets are usually derived by a domain knowledge of specific application. Attributes and user model are usually represented as a vector, which represents and/or models user's preferences.[15,16] Among approaches which are used for the classification, neural network, SVM or the linear regression are mostly used.

PROPOSED MODEL

We composed our model of multiple attribute categories, each of them is offering a unique perspective on a user's interaction with an application. We included six categories of attributes that could be extracted from generally available reactive web interaction logs of most of the online retailers.[17]

Let S be a set of attributes describing a given session, P a set of attributes representing purchases of the customer, B a set of attributes describing customer's behavior tendencies, F a set of attributes describing the frequency of customer's interaction with the web-application, A a set of attributes describing the actions executed during the session, and R a set of attributes describing ratings of items. The user model UM is then defined as a vector

$$\mathrm{UM} = (S, P, B, F, A, R). \quad (1)$$

Each of attributes set is characterized by various attributes. Let X be a vector of all attribute values ax (the list of all attributes is presented in Table 1) for the set of all attributes in XA (e.g., purchase, total time spent, actions count, average actions time), then

$$X = ax_1, ax_2, \ldots, ax_n, n \in N$$
$$ax_i, i \leq n \in XA, N = |XA|, X \in S, P, B, F, A, R. \quad (2)$$

Table 1 details the constructed attributes for individual categories. As a result, after each session, the user model is updated for each user—all attributes are recalculated. Value of attributes last sum paid, session length change, session gap change is initialized to 0 for a new user. Time since last visit if it is user's first visit and time since last purchase if a user did not buy any products yet are set to -1. Both attributes describing behavior tendencies are calculated relatively to the average value extracted from previous sessions of the user. We present the formulas for each attribute from Table 1 in the Appendix.

RESULTS

We evaluated our proposed model indirectly—by an application in the user churn prediction task. The evaluation of user models is typically realized in an indirect way (based on the task, the user model is defined for). This is a standard methodology in user modeling field.[18]

Data: To explore proposed model, we used transaction data from an e-shop (https://www.zlavadna.sk) (referred as *Dataset 1*). As Dataset 1 is not publicly available, we used also generally available, anonymized dataset (https://www.kaggle.com/retailrocket/ecommerce-dataset) (referred as the *Dataset 2*). To overcome the cold-start problem, i.e., little or no user actions, only users with more than four actions (and thus at last one session made) were considered. Next, we took one month interval and create user model to all users based on their activity within this month. Following month was used to indicate whether a user returned or not, which is, obviously, the predicted variable. As a result, we obtain approx. 160k of users (75% labeled as returning and 25% as leaving) for Dataset 1 and approx. 13k of users (12% labeled as returning and 88% as leaving) for Dataset 2.

Methodology: To explore the usefulness of proposed model and its specific attributes, we conduct several offline experiments. Similarly,[19] we used a reference model (RM) based on the

Table 1. **List of all attributes of proposed model with corresponding attribute sets.**

Session - S
session order - the order of user's last session
last session time - the time in seconds between last and first action of user's last session **no. of actions within session** - number of user's actions, i.e., clicks within the last session **average time on action** - last session time divided by the number of actions within last session **mobile device** - a binary flag referring to whether the mobile device was used in the last session **day of the week** - a numeric representation of a day of the week in which last session was performed **day of the month** - a numeric representation of a day of the month in which last session was performed **weekend** - a binary flag referring to whether the last session was performed during the weekend
Purchases - P
number of purchases - the number of user's previous sessions with a purchase event **total sum paid** - the amount of all user's purchases **last sum paid** - the amount of user's last purchase **session to purchase ratio** - the number of session made by the user divided by the total number of purchases he/she made **add to basket to purchase ratio** - the total number of add to basket events divided by the total number of purchases he/she made **add to basket since last purchase** - the total number of add to basket events made by the user since his/her last purchase event
Behavior changes - B
session length change - the difference between the last session duration and the user's average session duration **number of session actions change** - the difference between the number of user's actions in the last session and the average number of actions in his/her other sessions **session gap change** - the difference between user's last and previous session gap time (time between two sessions) and his/her average session gap
Interaction with application - F
number of sessions since last purchase - the number of sessions between user's last session and his/her most recent session with the purchase event **time since last purchase** - the time in seconds between user's first action of actual session and the last action of his/her most recent session with the purchase event **time since last visit** - the time in seconds between last action in user's previous session and his/her first action in actual session
Actions made in interaction - A
add to basket - a binary flag referring to whether the add to basket event is present in actual session **purchase** - a binary flag referring to whether the purchase event is present in actual session
Rating - R
average rating of last purchased item - the global average rating (across all users) of most recent bought item by the user **last rating** - the last rating given by the user **last user rating to average rating difference** - the difference between user's last item rating and the item global rating (across all users) **last user rating to average user rating difference** - the difference between user's last item rating and his/her average rating on all items **average user rating** - the user's average rating on all items

Table 2. Comparison of several under sampling approaches using SVM classifier. Tags indicate that the cluster identifier was used in the classification as additional feature. Several metrics are reported Acc - accuracy, Prec - precision, Rec - recall, ROC - Receiver operating characteristic curve.

Model	Acc	Prec	Rec	ROC
Unbalanced	0.622	0.342	0.622	0.614
Random sampling	0.608	0.609	0.627	0.608
Birch - basic	0.798	0.855	0.718	0.798
Birch - tags	**0.813**	**0.888**	**0.716**	**0.813**
DBScan - basic	0.647	0.615	0.787	0.648
DBScan - tags	0.782	0.882	0.651	0.782
KMeans - basic	0.757	0.854	0.619	0.756
KMeans - tags	0.764	0.852	0.638	0.764

session order. Next, we evaluated and report the influence of each attribute set included in the model based on various standard metrics (Accuracy, Precision, Recall, ROC). We also evaluated the attributes importance. As the data in the task of the churn prediction are highly unbalanced, we used the clustering to help the predictor. The threefold cross validation is used in all experiments.

Clustering for Data Undersampling

There are two basic options for reduction of data unbalance—under or oversampling.[20] Due to the validity, the downsampling is usually used. It is clear that candidates which should be "removed" (undersampled) have to be selected. Often a naive, i.e., random undersampling is used. However, this may result in selecting many similar instances and omitting important ones.

Because of this, we employed a clustering to the selection of downsampling candidates. The idea is to create k clusters of similar users. Next, to create a undersampled sample, based on a few users, representing each cluster. As we select some candidates from all clusters, all user "types," i.e., preferences are used for training.

For this reason, we experimented with several clustering approaches (*K*-means, DBScan, Birch). As a result, we used several instances from each cluster to create an undersampled train set. Thanks to such approach, we ensured that a classifier covers as much data patterns in the learning phase as possible.

Moreover, thanks to the clustering, we have an information of instances similarity—encoded by the cluster identifier. We hypothesize that including such information (new attribute with cluster identifier), will help the classifier to reach better performance.

As we can see (see Table 2), the usage of a clustering significantly improves the results. Among the compared algorithms, the best performance was achieved using the Birch clustering algorithm, which we used in further experiments. Similarly, considering the cluster identifier (i.e., instances similarities) helped the classifier to obtain better performance as we expected.

Table 3. Comparison of different attributes sets and the complete model for the churn prediction task (SVM classifier). several metrics are reported Acc - accuracy, Prec - precision, Rec - recall, ROC - Receiver operating characteristic curve.

	Dataset 1				Dataset 2			
Model	Acc	Prec	Rec	ROC	Acc	Prec	Rec	ROC
RM	0.80	0.51	0.40	0.50	0.65	0.74	0.65	0.65
RM + purchases P	0.81	0.60	0.50	0.61	0.63	0.75	0.63	0.63
RM + behavior B	**0.87**	0.82	0.59	0.56	0.65	0.75	0.65	0.65
RM + interaction F	0.82	0.85	0.72	0.60	0.69	**0.78**	0.69	0.69
RM + actions A	0.79	0.59	0.54	0.65	0.64	0.65	0.64	0.69
RM + ratings R	0.77	0.63	**0.88**	0.53	0.65	0.74	0.65	0.65
Complete model	0.84	**1.00**	0.62	**0.81**	**0.73**	**0.78**	**0.73**	**0.73**

Table 4. Results of customer churn prediction based on various information amount represented by session count performed by user in the train set. Several metrics are reported Acc - accuracy, Prec - precision, Rec - recall, ROC - Receiver operating characteristic curve.

Session	Dataset 1				Dataset 2			
count	Acc	Prec	Rec	ROC	Acc	Prec	Rec	ROC
1	0.65	0.99	0.11	0.55	0.70	0.75	0.70	0.64
2	0.94	1.00	0.92	0.96	0.76	0.76	0.76	0.64
3	0.99	1.00	0.99	1.00	0.90	0.86	0.90	0.52
4	1.00	1.00	1.00	1.00	0.95	0.93	0.95	0.49
5 +	1.00	1.00	1.00	1.00	1.00	0.99	1.00	0.50

Churn Prediction

To better understand the customer's behavior, our goal was to quantify the importance of each attribute set (S, P, B, F, A, R). To do so, we performed classifications using only the attributes of each set. To get a baseline for results, we use a RM[19] containing only the attributes describing a user session (set S).

For the classification task, we used the SVM with *rbf* kernel. Table 3 shows results for various model parts and datasets used. As we can see for Dataset 1 the RM with ratings R achieved the highest recall, our proposed complete model outperformed every other variant by values of other metrics. To bring even more insight into the reasons behind this, we calculated the correlations between predicted variable and the individual attributes. The most correlated attributes refer to session number, time since last visit, total sum paid, or number of purchases. On the contrary, lowest correlation was found with attributes as day of the week, mobile device, or weekend. Similar pattern can be observed for the Dataset 2. The complete model obtains best results comparing each model part standalone. This supports our hypothesis that the model is able to capture users' churn tendencies. The model is designed to cover generally available information about user transactions. However, the optimal subset of attributes should be identified for each domain (e.g., by using random forest classifier, correlations).

Further, we explored the performance of proposed model with respect to various amount of history of users. It is clear that a new user which performed only few actions can be hardly captured by the model. This is supported by obtained results (see Table 4). As we can see, with increasing number of sessions performed by a user, the prediction performance significantly increases. On the contrary, two sessions performed seems to be sufficient for our model to capture user behavior patterns and next results in good prediction performance (Table 4—Dataset 1). Similar pattern can be observed for Dataset 2. However, the ratio between returning and nonreturning customers is in comparison to Dataset 1 quite low. This together with the Dataset 2 anonymization process which brings some noise, results in lower results.

To conclude, proposed model captures user behavior and tendencies to churn. Obtained results support the relevance of model attributes, while the best performing subset should be found for each application domain.

CONCLUSIONS

The acquisition of a new customer is 6–7 times more expensive as retaining an old one.[1] More and more businesses try to identify users with tendencies to churn. As a result, various actions may be performed in order to retain specific user.

In this paper, we proposed a novel user model, which can be used for the prediction of the user's intent to churn. Due to modeling the user's interaction with a web application in a complex way, rather than simply, the model reflects user's behavior during his/her sessions.

We found out that attributes related to the frequency of user's interaction provided the best indicator of the churn intent. The one-shoot buyers show different behavioral patterns in terms of correlation between money spent over the given period and the tendency to churn.

The typical problem of the churn prediction is hugely unbalanced data. We addressed this problem by utilizing a clustering approach to reduce the major class and to preserve distinctive information in the undersampling process. Thanks to this idea, we were able to learn classification model on the optimally sampled data.

Furthermore, the conducted experiments proved efficiency of our model in the task of prediction of user's churn. Trying to identify the last sessions, we achieved precision of 100.0% and recall of 62.0% for Dataset 1; and precision of 78.0% and recall of 73.0% for Dataset 2. However, when exploring the influence of the amount of user interaction—number of sessions performed in the train set, we found out that with increasing number of user sessions, the performance of the model significantly increases. Using proposed model on users with at last two sessions results for Dataset 1 and at last three sessions for Dataset 2 to satisfactory predictions. This is a promising result (overcoming state-of-the-art approaches), which proved the validity of proposed model and its attributes across various datasets.

We believe that considering the content of items within specific domain may increase the success of churn intent prediction. This will, however, increase the model domain dependence and computation complexity.

APPENDIX: USER MODEL ATTRIBUTES

Let the $Se_{u,n}$ be the nth session Se of a user u. Let $Ac_{u,n,m}$ be the sequence of m actions made by a user u during his/her nth session then

$$\text{session order} = |Se_u| \tag{3}$$

$$\begin{aligned}\text{last session time} &= \text{timestamp}(Ac_{u,\text{last},m}) \\ &- \text{timestamp}(Ac_{u,\text{last},1}),\ m = |Ac_{u,\text{last}}|\end{aligned} \tag{4}$$

$$\text{no. of actions within session} = |Ac_{u,\text{last}}| \tag{5}$$

$$\text{average time on action} = \frac{\text{session time}}{|Ac_{u,\text{last}}|} \tag{6}$$

$$\begin{aligned}&\text{mobile device} = 1\text{ , if a mobile device was used,}\\ &\quad 0 \text{ otherwise}\end{aligned} \tag{7}$$

$$\begin{aligned}&\text{day of the week}\\ &\quad = \text{get\_day\_of\_the\_week}(\text{timestamp}(Ac_{u,\text{last}}))\end{aligned} \tag{8}$$

$$\begin{aligned}&\text{day of the month}\\ &\quad = \text{get\_day\_of\_the\_month}(\text{timestamp}(Ac_{u,\text{last}}))\end{aligned} \tag{9}$$

$$\begin{aligned}&\text{weekend} = 1\text{ , if day of the week is Saturday}\\ &\quad \text{or Sunday},\ 0 \text{ otherwise.}\end{aligned} \tag{10}$$

Let $PurchAc_{u,n}$, $PurchAc \in Ac_{u,n}$ be the set of purchases made by the user u during the nth session. Similarly, let $AddAc_{u,n}$, $AddAc \in Ac_{u,n}$ be the set of add to basket actions made by the user u during the nth session

$$\text{number of purchases} = \sum_{i=1}^{\text{last}-1} |PurchAc_{u,i}| \tag{11}$$

$$\text{total sum paid} = \sum_{i=1}^{\text{last}} \text{get\_price}(PurchAc_{u,i}) \tag{12}$$

$$\text{last sum paid} = \text{get\_price}(PurchAc_{u,last}) \tag{13}$$

$$\text{session to purchase ratio} = \frac{|Se_u|}{\text{number of purchases}} \tag{14}$$

$$\begin{aligned}&\text{add to basket to purchase ratio}\\ &\quad = \frac{\sum_{i=1}^{\text{last}} |AddAc_{u,i}|}{\text{number of purchases}}\end{aligned} \tag{15}$$

$$\begin{aligned}&\text{add to basket since last purchase}\\ &\quad = \sum_{i=\text{last}(PurchAc_u)}^{\text{last}} |AddAc_{u,i}|\end{aligned} \tag{16}$$

$$\begin{aligned}&\text{session length change} = \text{last session time}\\ &- \frac{\sum_{i=1}^{\text{last}-1} \text{timestamp}(Ac_{u,i,m}) - \text{timestamp}(Ac_{u,i,1})}{\text{session order}},\\ &m = |Ac_{u,i}|\end{aligned} \tag{17}$$

$$\begin{aligned}&\text{number of session actions change}\\ &\quad = |Ac_{u,\text{last}}| - \frac{\sum_{i=1}^{\text{last}-1} |Ac_{u,i}|}{\text{session order}}\end{aligned} \tag{18}$$

$$\text{session gap change} = \text{timestamp}(Ac_{u,\text{last},1}) - \text{timestamp}(Ac_{u,\text{last}-1,m}) - \frac{\sum_{i=1}^{\text{last}-1} \text{timestamp}(Ac_{u,i,1}) - \text{timestamp}(Ac_{u,i-1,m})}{\text{session order}}, \quad m = |Ac_{u,i}| \tag{19}$$

$$\text{number of sessions since last purchase} = \text{session order} - \text{get\_order}(\text{last}(PurchAc_{u,i})) \tag{20}$$

$$\text{time since last purchase} = \text{timestamp}(Ac_{u,\text{last},1})) - \text{timestamp}(\text{last}(PurchAc_{u,i,m})) \tag{21}$$

$$\text{time since last visit} = \text{timestamp}(Ac_{u,\text{last},1})) - \text{timestamp}(Ac_{u,\text{last}-1,m}),\ m = |Ac_{u,i}| \tag{22}$$

$$\text{add to basket} = 1 \text{ if add to basket event is present in actual session},\ 0 \text{ otherwise} \tag{23}$$

$$\text{purchase} = 1 \text{ if purchase event is present in actual session},\ 0 \text{ otherwise} \tag{24}$$

$$\text{average rating of last purchased item} = \text{get\_avg\_rating}(\text{last}(PurchAc_{u,i})). \tag{25}$$

Let $\text{Rat}_{it,u} \in \text{Rat}_{it}$ be the rating of item it made by a user u and Rat_u the set of all ratings given by the user u, then:

$$\text{last rating} = \text{last}(\text{Rat}_{it,u}) \tag{26}$$

$$\text{last user rating to average rating difference} = \text{last}(\text{Rat}_{it,u}) - \text{get\_avg\_rating}(\text{Rat}_{it}) \tag{27}$$

$$\text{last user rating to average user rating difference} = \text{last}(\text{Rat}_{it,u}) - \text{get\_avg\_rating}(\text{Rat}_u) \tag{28}$$

$$\text{average user rating} = \text{get\_avg\_rating}(\text{Rat}_u). \tag{29}$$

ACKNOWLEDGMENTS

This work was supported by the Scientific Grant Agency of the Slovak Republic under Grant VG 1/0667/18, the Slovak Research and Development Agency under Contract APVV-15-0508 and APVV-17-0267. The authors would also like to thank the Ph.D. student P. Gaspar for his support.

REFERENCES

1. F. Reichheld, *Prescription for Cutting Costs*. Boston, MA, USA: Bain & Company, 2014.
2. J. Lee, M. Podlaseck, E. Schonberg, and R. Hoch, "Visualization and analysis of clickstream data of online stores for understanding web merchandising," *Data Min. Knowl. Discov.*, vol. 34, no. 1/2, pp. 59–84, 2001.
3. P. Brusilovsky, "Methods and techniques of adaptive hypermedia," *J. User Modeling User-Adapted Inter.*, vol. 6, no. 2/3, pp. 87–129, 1996.
4. T. Kramar, M. Barla, and M. Bielikov, "Personalizing search using socially enhanced interest model built from the stream of users activity," *J. Web Eng.*, vol. 12, no. 1/2, pp. 65–92, 2013.
5. M. Mezghani, C. A. Zayani, I. Amous, and F. Gargouri, "A user profile modelling using social annotations: A survey," in *Proc. 21st Int. Conf. World Wide Web*, 2012 pp. 969–976.
6. G. Dror, D. Pelleg, O. Rokhlenko, and I. Szpektor, "Churn prediction in new users of Yahoo! answers," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 829–834.
7. M. Kompan, O. Kassak, and M. Bielikova, "Beyond user preferences: The short-term behaviour modelling," in *Proc. 1st Workshop Temporal Reasoning Recommender Syst. Co-located 11th Int. Conf. Recommender Syst.*, Como, Italy, Aug. 31, 2017, pp. 1–3.
8. T. Althoff and J. Leskovec, "Donor retention in online crowdfunding communities: A case study of DonorsChoose.org," in *Proc. 24th Int. Conf. World Wide Web. Switzerland*, 2015, pp. 34–44.
9. K. S. Dave, V. Vaingankar, S. Kolar, and V. Varma, "Timespent based models for predicting user retention," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 331–342.
10. V. Lazarov and M. Capota, "Churn prediction. Business analytics course," *TUM Computer Science*, 2007, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.462.7201
11. H. Abbasimehr, M. Setak, and M. J. Tarokh, "A comparative assessment of the performance of ensemble learning in customer churn prediction," *Int. Arab J. Inf. Technol.*, vol. 11, no. 6, pp. 599–606, 2014.
12. A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 650–658.

13. D. Yin, S. Guo, B. Chidlovskii, B. D. Davison, C. Archambeau, and G. Bouchard, "Connecting comments and tags: Improved modeling of social tagging systems," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 547–556.
14. W.-H. Au, K. C. C. Chan, and X. Yao, "A novel evolutionary data mining algorithm with applications to churn prediction," *IEEE Trans. Evol. Comput.*, vol. 7, no. 6, pp. 532–545, Dec. 2003.
15. C. Senot, D. Kostadinov, M. Bouzid, J. Picault, A. Aghasaryan, and C. Bernier, "Analysis of strategies for building group profiles," in *Proc. Int. Conf. User Modeling, Adaptation Personalization*, 2010, pp. 40–51.
16. P. De Bra *et al.*, "AHA! The adaptive hypermedia architecture," in *Proc. 14th ACM Conf. Hypertext Hypermedia*, 2003, pp. 81–84.
17. P. N. Huntington and H. R. Jamali, "Website usage metrics: A re-assessment of session data," *Inf. Process. Manage., Int. J.*, vol. 44, no. 1, pp. 358–372, 2008.
18. G. Piao, "Towards comprehensive user modeling on the social web for personalized link recommendations," in *Proc. Conf. User Modeling Adaptation Personalization*, 2016, pp. 333–336.
19. M. R. Khan and J. E. Blumenstock, "Predictors without borders: Behavioral modeling of product adoption in three developing countries," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 145–154.
20. I. H. Witten, E. Frank, and M. H. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann, 2011.

Patrik Berger received the master's degree from the Slovak University of Technology in Bratislava, Slovakia, in 2018. His research interests include the areas of user modeling and machine learning. Contact him at patrik.berger@stuba.sk.

Michal Kompan is currently an associate professor at the Institute of Informatics and Software Engineering, Slovak University of Technology in Bratislava, Slovakia. He received the Ph.D. degree in 2014 from the University of Technology in Bratislava. His research interests include the areas of personalized recommenders for single or group of users and user modeling. He is a member of IEEE, and ACM. He is the corresponding author. Contact him at michal.kompan@stuba.sk.