A Practical Framework for Real Life Webshop Sales Promotion Targeting

Gábor Kőrösi and Tamás Vinkó
University of Szeged, Institute of Informatics, Hungary
E-mail: korosig@inf.u-szeged.hu, tvinko@inf.u-szeged.hu

Technical paper

Keywords: promotion targeting, behavior analysis, hybrid recommendation system

Received: February 28, 2020

In recent years, online marketing has become increasingly extensive and effective. Product recommender systems are often deployed by e-commerce websites to improve user experience and increase sales. To address this, more and more e-commerce started to use machine learning models to predict customers' purchase behaviors. In the scientific literature there are only few real-life *studies to date which give solutions for recommendation systems for online advertising. The demand from the owners of such websites is given, however, it is hard for them to choose a method or model to predict from an endless number of options for some specific circumstances. The aim of this paper is to propose a practical guideline as a hybrid approach that predicts customers' purchase behaviors and helps to target advertisement, sales form in user level. To this end, we have designed a robust hybrid model to predict interested sales form based on user behavior within a large e-commerce website. The paper details a real-life practical solution and build a structure that can be used in a large variety of e-commerce systems.*

Povzetek: Opisan je razvoj modela nakupovanja po spletu z namenom ciljnega oglaševanja.

1 Introduction

One of the most important and dynamically developing areas today is e-commerce and related services. While in a traditional shop tracking customer is difficult (e.g.,loyalty card program), a webshop's back-end offers countless solutions to solve this problem. For example, we could use cookies, spent checking, newsletter and product tracking [4, 15, 1]. The main driving force behind this fast evolution is the fact that we can understand and anticipate user behavior better, and we can answer the related questions in real-time. The key goal is to get the highest response from users by spending as little money and time on it as possible, and create customer-oriented services [2]. That is named personalization and targeting [13], where the objective is to find the best matching ads or form of sales promotion to be displayed for each user. The solution is not new, as we could see similar solutions at the first generation webshops, but nowadays the amount of data is much higher than before.

When the task is to efficiently process huge amount of data, it is useful to try and find a solution in those research papers written based on similar task. For example, [26] analyzing clickstream, [15] uses email sending history, while [1] collects user activity to predict user's future behavior. At first glance, the task does not seem to be a difficult one, as using data mining and data science in e-commerce is not new, and there is a huge amount of papers published with the same purpose. These papers refer to a waste amount of machine learning (ML) tools and solutions which are able to help with this optimization. For instance, classification can predict the occurrence of an event, or regression techniques that can help us to predict the time or amount of money the user will spend on the website. More sophisticated solutions are offered by collaborative filtering or content-based approaches. The repository of toolkits may seem endless, but solving a problem is never the same, and it is seldom enough to use just one tool to solve a problem. Many recent publications have introduced some kind of hybrid solution for this complex problem in which one has to combine and embed simple methods to find a proper model. For example, in [5] we could see a typical hybrid recommendation model that integrates user-based and item-based collaborative filtering, content-based filtering together with contextual information to get rid of the disadvantages of each approach.

Thorough literature review on these subjects can lead to an impression that most of the scientific papers are theoretical model descriptions instead of accurate and practical model descriptions. One could find vague model formulations that make it difficult or impossible to rebuild a presented solution in real life. Along this line of thought, we have concluded that, besides theoretical models, there is a huge demand for publications that document a case study and provide the opportunity for anyone to reproduce it the results on their database. Our goal is to make and document a case study that demonstrates an ML-based recommendation system, which classifies users and provides an individual-level approach for ads form. Based on our literature review we found that a hybrid recommendation system

provides the most accurate solution for that. We combined and embedded various classification and regression models, including Logistic Regression, Random Forest, GBM, and XGBoost to get the most accurate solution.

The rest of the paper describes our approach as follows. Sections 2 and 3 describe the background of the problem. In Section 4 the dataset and the generated features are detailed. The model ensemble is briefly described in Section 5. In Section 7, the importance of features is studied, top features are listed, and our solution is given. Finally, Section 8 concludes the results of the study.

2 Background

As data is increasing, more and more companies are demanding high quality solutions from their data scientists. The use of recommendation systems has become a daily concept in product suggestion, product group selection, promotional message content generation which is supported by machine learning techniques. Common examples of applications include the recommendation of movies (e.g., Netflix, Amazon Prime Video), music (e.g., Pandora), videos (e.g., YouTube), news content (e.g., Outbrain) or advertisement (e.g., Google) [25].

In this paper we give a detailed description of a recommendation system which can make user-level marketing letter or offer sales promotion. Note that *recommendation system* is a quite general concept. It could be based on the collaborative filter solution, the content-based method, the classification or regression, and their embedding in different depths and widths. What follows is an outline of what a recommendation system might consist of.

Collaborative filtering (CF) is probably one of the most used and well-known technologies. Behind the basic idea, the solution is that based on users' historical data, the users are put into an n-th dimensional space which makes possible to then measure the distance between them. In light of this, we could make recommendations based on the data of the users closest to each other [7]. This CF technique proved its power, but on the other hand, a huge amount of work pointed out the disadvantages of it. These are the followings: cold start problem, data sparsity, and scalability [29].

Besides collaborative filtering, the second most popular solution is the content-based method. It is a technique which operates with unique characteristics and behaviors of each customer, and in turn, delivering personalized content for each user, based on their content consumption history across channels. Another interesting way is the community-based method. This approach assumes that the content coming from a user's friends or authoritative users is more likely to be interesting for a user than the rest.

While collaborative filtering and content-based models, used only a static 'user states' we could find many papers which are using uni- or multivariate user event sequences, time-series to build a predictive model. Koehn *et al*. [20] divided the user event sequence prediction problem into four groups, namely the 'predict the product group', 'classify a sequence', 'predict the outcome of an incomplete session', and 'click-through rate prediction'. In our work we are focusing to predicting the users' interest, which was created based on some initial observations on the users' purchase behavior during the shopping process, meaning that our task is rather similar to the 'predict the product group' task of the recommender systems. Koehn *et al*. [20] summarized the methods of event sequence data preprocessing, highlighting their advantages and disadvantages. One of the most often implemented methods is to create aggregated, cumulated data, which, however, results in data loss and requires manual feature engineering by the domain experts. Another common method is to create sequence segments or sliding a window, where we use only a chunk/fixed-length part of the data. Lastly, there are neural networks and embedding layers, where we can work with partially or completely raw data. In the field of sequence prediction approach, we could find many papers.

Perhaps one of the most promising paper which related to our work is created by Yu *et al*. [28]. They used recurrent neural network on sequenced data to identify web shop users habits and made the next basket recommendation. They applied recurrent layers in the temporal domain and proved their effectiveness for handling the temporal dimension for time series classification. Deep learning based (DLL) solution with time series have proven efficiency in many areas, however, web-shop log data often includes variables that contain mixed continuous and discrete variables. Even these kinds of data can be easily handled by a decision tree-based solution, in neural network this is not so easy. In deep learning based approaches, the discrete-valued sequences must be transformed into the numeric space. Using one-hot encoding might not prove to be overly useful, as it explores the dimensionality of the input feature vector and dramatically increases its sparsity. Inspired by Natural Language Processing, we managed to transform our categorical data into a dense space utilizing embeddings. These methods encode categories as vectors based on contextual similarities and then feed them into the recurrent or convolutional neural network. The embedded vectors are usually trained together with the time-series/sequence model training process [21]. The embedding of discrete-valued sequences was successfully applied in user behavior analysis. For instance, An *et al*. [3] presented their neural user embedding approach which was capable of learning informative user embeddings by using the unlabeled browsing-behavior. Koehn *et al*. [20] proposed their impressive clickstream classification results where they applied RNN architectures and embedding layers. Cheng *et al*. [9] introduced the Wide and Deep feature representation method. In their terminology, Wide representations were one-hot encoded features which could memorize sparse feature coincidences, while Deep representations consisted of dense embeddings which gave generalization power to deep learning systems.

Although content-based and community-based methods have proven their worth in many case studies, in our case it was almost impossible to apply these methods due to the lack of data. Another approach could be the deep learning based solution, but as many paper shows (e.g. [8, 17]) when the dataset is based only short sequences (as our dataset), a traditional ML model can outperform a DLL based model. Based on these paper even a XGboost based classification model or regression would provide a good solution in an optimum prediction system, but unlike simple patterns, things are always more complicated in real life.

To solve the backward of the traditional and DLL based methods, the concept of hybrid or combined systems are becoming more popular in many papers. Bozanta and Kutlu [5] summarized that while each filtering approach has different drawbacks, a hybrid approaches combines the existing approaches and aim to minimize or remove the drawbacks of existing approaches, which may occur when they are used individually. The exact description does not exist for a hybrid solution, but we could certainly use the aforementioned tools at different depths and widths. There are quite many papers proving that a hybrid approach provides a better solution than the single method, see, e.g., in [5, 7, 12]. Thus, we have chosen this solution for our workflow, and we decided to use use such a hybrid model for our system which used both regression and classification method.

Our goal was to solve the problem of predicting the user behaviors about the sales promotion. Similar goals is solved by Martínez *et al*. [23] and Liu *et al*. [22]. They created a model that can predict future customer behavior which based on the set of customer-relevant features that derives from times and values of previous purchases. As our solution, they apply machine learning algorithms including logistic Lasso regression, the extreme learning machine and gradient tree boosting for predicting whether the customer makes a purchase in the upcoming month. Although these two cited papers are very similar to the solution we used, however, unlike them, we tried to create a prediction algorithm not just by using one method but by a (hybrid) combination of them.

3 Problem statement

The main objective of this paper is to solve the problem of predicting the purchase behaviors of users who have known the history on an e-commerce website. More closely, we aim at forecasting which ads group or form of sales promotion user will most likely to use based on purchase history and profile information. This form of sales promotion could be: buy two, get one free; price deal; sampling, etc.

Although we did not directly use others' work to design our system, the solution we came up with is strikingly similar to the description of [29]. That is, a predictive system would help in several practical scenarios such as

- build a cold start recommender system, by providing high-level recommendations to users who connect for the first time to an e-commerce website;
- improve existing product recommendation engines, by providing category-level priors that can guide the recommender system to and domains of interest for the user;
- provide e-commerce companies with tools for targeted email/social media campaigns.

Our paper has two main goals. The first is to explore which information is correlated with the form of sales promotion which the users most likely to use (see in Table 1 for an illustrative example.) Based on this we have built and tested a hybrid model which optimizes a user-level table, in order to propose the form of sales promotion to users that fit the best to their interests and preferences, see Table 2. The second goal is to back-test and document well each critical point of hybrid machine learning algorithms which could be used as a base structure for those who want to replicate our model or build a similar system.

4 Datasets

We have used data which has been recorded from a health and beauty webshop. The data has contained near millions of users, from different markets (countries), however, in order to obtain the richest data possible, we have filtered it by the oldest market which includes $230,000$ user-profiles and their purchase history. Data consists of seven years of user interaction logs with the webshop. Each event has a user identifier, a timestamp, and an event type. The purchase data contains 5 categories of events: pageview of a product, basket view, buy, ordered timestamp, and delivered timestamp.

There are around 240 different types of products. In the case of a buy or a basket view, we have information about the price and extra details. An average customer has been used the shop two or three times yearly, which leads to very sparse and high dimensional dataset. This is not surprising as it is extremely common in recommender systems [25]. As a solution, there are two obvious ways to reduce the dimensionality of the data: either by marginalizing the time (aggregate pageviews per user over the period) or the product pageviews (aggregate products viewed per time frame) [26]. In this work, we follow both approaches.

As a first step, our solution connected unique events with sessions. We used homogeneous like purchase history only and heterogeneous example clicks, profile data in nature. These events are then cleansed and ordered by their timestamps to form the action chain.

As a next step, we transformed unique events into a feature list (e.g., number of purchases, the distance between two logins, etc.). Beside of the evident data (number of, sum of, mean of purchases), the script accumulated other data such as:

Table 1: Illustration: problem statement as a binary classification.

1st purchase	2nd purchase	3rd purchase	···	nth purchase
		time of prediction	⇒	Likely to buy with *sales promotion* (user who use more than 50% promotion for buying something)

Table 2: Illustration: problem statement as a recursion; the distribution of sales promotion types.

1st purchase	2nd purchase	3rd purchase	···	nth purchase	
		Time of prediction	⇒	SPType1 SPType2 ⋮ SPType$n$	35% 25% ⋮ 50%

– distance (in time) between first and second, third, etc. actions;

– number of purchases in first, second, etc. months;

– increase or decrease in purchases compared to the previous month by month;

– the reaction times between advertising letters and a purchase.

4.1 Feature engineering

One of the most important steps for better performance of a classifier is to preprocess the data correctly. Besides the regular data cleaning process, we transformed features by scaling each feature to a given range with min-max scaling. As a last preprocessing step, we calculated feature importance with tree-based ensemble method namely `ExtraTreesClassifier` [14]. Based on the obtained results by this method, our model uses only the top 20 features, which significantly increased the accuracy of the results.

5 Methodology

In order to handle the popularity-bias, we divided the problem into two subtasks:

i) predict if a user is sensitive for the sales promotion or not, and

ii) predict which kind of form of sales promotion is more interested in it.

As a solution, we have created a hybrid model which used both regression and classification method, see Figure 1.

The recommendation model returns two lists. The first list gives information about the users, if they are likely to use or not any of the forms of sales (the sensitivity for sales promotion). The second list provides us with the data to calculate the probability for every sale (which form of sales likely to use).

For the results we propose a novel hybrid recommendation algorithm where similarity measurement is performed between a user and form of sales on features derived from their profile and history information. As a result, we obtain a table where every single user gets his/her predicted value, as we can see in Table 3.

6 Experimental setup

As we want to use raw log data to make a prediction for recommendation, we have to handle the data sparsity problem. As already mentioned, our dataset contains $230,000$. However, only $33,000$ of them have data of sufficient quality. So, in our experiment, we used only this reduced and filtered dataset. To conduct experiments, we split the entire dataset into test (20%) and training (80%) sets.

In the first step, we have trained various classification models, including Logistic Regression [11], Random Forest [6], LightGBM [19], and XGBoost [8], where grid search was used to select the optimal parameters. As the final results proved, XGBoost classifier and XGBRegressor performed the best. Additionally, the majority of classifier (MC) [18] is used as a baseline for comparison with the above learning algorithms.

For the regression problem, we used the central tendency measure as the baseline for all predictions. Based on these, we inspected the hybrid models using the training set and adjusted the predictive algorithms' parameters achieving the best performance on the validation set. Predictions were made for each instance in the test set and the forecasted results were compared with the true values by computing corresponding performance metrics. To obtain the best evaluations we have used K-fold validation where

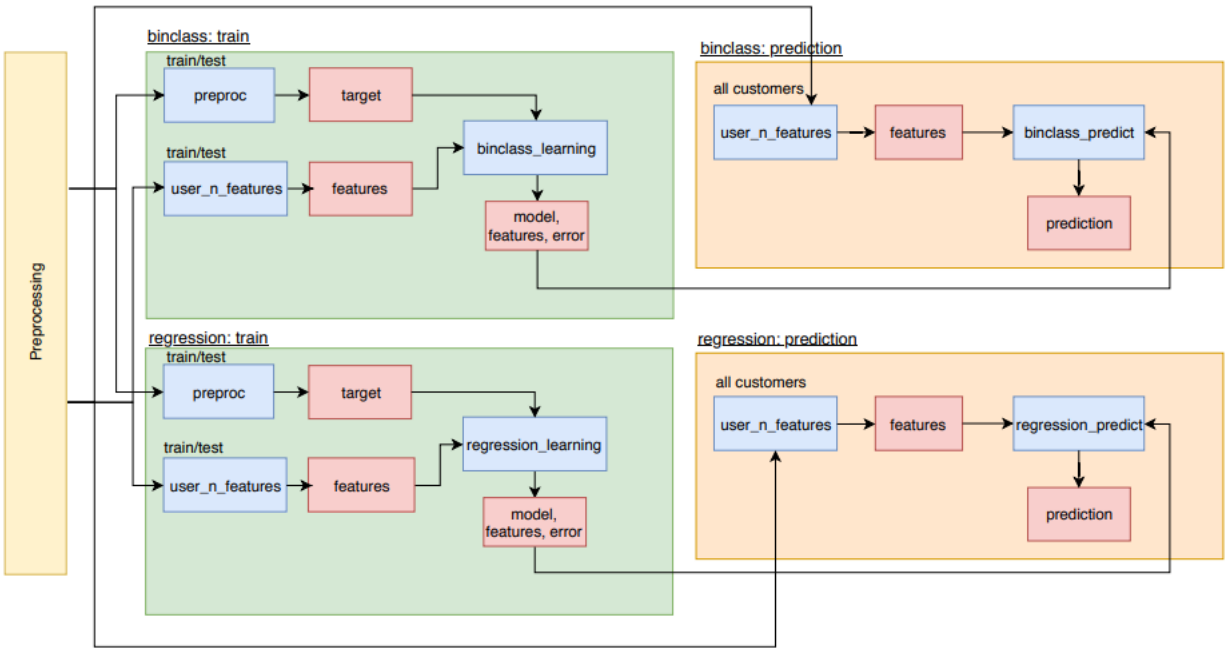

Figure 1: State diagram of our hybrid solution.

Table 3: Example of model outcome.

user id	likely to use sales promotions	likely to use sales promotion type				
		type1	type2	type3	type4	type5
1000	YES	35%	50%	5%	3%	7%
1001	NO	0%	0%	0%	0%	0%

both training and validation sets were also used for prediction.

Handling problem with an ensemble classification and regression tree

The first goal is to predict if a user is likely to use or not a sales promotion, which is a binary classification problem. To find the best solution we have trained and tested classification models as many times as we could. In the end, we have found that the XGBoost ensemble classifier [8] gives the best results. It is not surprising, because tree boosting is a highly effective and widely used machine learning method.

Another important feature is that the algorithm has a good performance as it includes an efficient linear model solver and can also exploit parallel computing capabilities [8]. Ensemble learning to provide a systematic solution to merge the power of multiple learners. The prediction value of XGB can have different interpretations, depending on the task, i.e., regression or classification. XGB is a tree ensemble model which set of classification and regression trees. It could classify our data into one of a finite number of values, that while called a regression (nonlinear model). Besides XGB, we compared our results with Linear regression [10], Lasso [24] and Ridge regression [16].

7 Results

Classification. It is well known that the main problem of the recommendation system is the cold start problem. It could appear when the user has started his/her initial steps, or in our case when a shop owner started a new sales promotion type, which makes very sparse data. To solve this problem, we filtered (dropped out) those users and promotions from the training dataset which has too sparse or no data. Based on our model, we made a binary classification with XGBoost to predict user likely to use a sales promotion or not. The parameters of the estimator used to apply optimization by cross-validated grid-search over a parameter grid.

To find the most accurate model, we have tried more models and settings. The results are reported in Table 4, where the window size (number of purchase) was 3 for all

Table 4: Results of classifications.

model	ACC	F1	precision	recall
Baseline	0.587	0.342	0.351	0.337
Logreg_all	0.676	0.409	0.620	0.306
Logreg_top10	0.685	0.404	0.661	0.291
XGBoost_all	0.706	0.527	0.652	0.436
XGBoost_top10	0.703	0.518	0.657	0.419
XGBoost_all_HPT	0.768	0.519	0.666	0.423
XGBoost_top10_HPT	0.771	0.509	0.658	0.417
XGBoost_top10_HPT(4)	0.790	0.624	0.713	0.554

Table 5: Error rates of regression models.

model	Sales promotion Type1			Sales promotion Type2			Sales promotion Type3		
	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
Baseline_CV	5.840	53.568	7.313	9.970	161.948	12.723	12.679	256.261	16.001
DNN	5.906	53.275	7.298	9.870	158.492	12.589	12.600	259.132	16.097
LR_all_CV	5.039	46.029	6.779	8.927	131.045	11.442	11.368	206.408	14.365
LGBMReg_CV	4.715	42.469	6.551	8.446	118.202	10.869	10.946	191.677	13.843
StackReg_CV_TOP	4.778	43.153	6.564	8.720	125.506	11.200	11.092	196.392	14.013
LR_CV_TOP	4.986	44.844	6.691	8.829	127.842	11.301	11.234	203.471	14.284
LGBMReg_CV_TOP	4.700	42.349	6.501	8.602	112.589	11.067	10.895	191.120	13.824

the methods, except in the last configuration.

During the first phase, we have used XGboost with all, and with only the top-10 features, which achieved 70% accuracy. To improve this, we have applied hyperparameter tuning, namely cross-validated grid-search over a parameter grid which gains better accuracy.

We wanted to make further improvements, but the sparsity of the data did not allow it. The main problem is that we want to predict user feature habits as soon as possible. For that reason, we used the user's first 3 purchase history to train the model, but this was (as expected) not enough to improve the results. To get better results, we need more data, such as we expected it. The solution to this problem is simple: we have to wait for more information, or encourage clients to fill the profile table. To prove this concept, we have trained our model with the user's first 4 purchases, which achieves 0.79 accuracy (last row in Table 4).

To find another way for this challenge, we changed our method like many researchers suggest: if we don't have accurate enough classification model, we have to change our point of view. To use this idea, we retested our solution as a regression with XGBRegression (as a regression problem). As a result, it affords $RMSE = 16.77$, which is not offering better outcome, because if we transform this result into a classification result, we got accuracy: 0.686, precision: 0.578, recall: 0.546, and F1: 0.562.

Regression. In our second phase, we were looking forward to determining which type of sales promotion will prefer most of our users (see in Figure 1). It is a regression problem, where we have to predict every SP type for every user. To make a measurable result, we did not test all the types of SP, instead of that, we chose only 3 types of promotion:

- Type1 is an SP type which has a long history in our webshop;
- Type2, which has only a year background, and
- Type3 is the youngest SP type (less than 6 months is using).

Based on this idea we obtained the results reported in Table 5.

To get the best outcome, we tested more models with different settings, like linear regression (LR), LightGBM (LGBMReg), and a simple deep neural network (DNN). In the initial step, our model used all ($n = 129$) normalized, scaled and skewed feature sets. Based on this method LGMBReg made the most accurate solution.

As a second step, we wanted to increase our model's accuracy. To solve this, we wanted to find the most important features. For this purpose, we used wrapper method, namely backward elimination. As the name suggests, we gave all the possible data to the model at first. We track the performance of the model and then repetitively remove the worst performing features one by one until the overall performance of the model comes in a suitable range. To calculate feature importance, we are using the ordinary least squares (OLS) model [27]. After many attempts and settings, the best solution is made by LGMBReg which is a tree-based regression model, which made a much more accurate model than the random choice.

Discussion. Our problem and its solution to predict acceptance of the sales promotion is unique, since we do not predict a repeat purchase but a reaction to advertising letters. Regardless, we wanted to somehow compare the performance of our model with other models as well. The results, and methodology of our paper is similar to the results obtained by Martínez *et al*. [23], so we compared our results with theirs. Our goal was to predict if a user is likely to use or not a sales promotion, which was same as their binary classification problem. While our model reaches 79% accuracy, their solution reached 86.68%. The difference in accuracy between the two models is not surprising, since we used only the first 4 purchases, they used 24 months for the same task. As they noted, it is difficult to make an accurate prediction model from short data and few purchases, however, over time, as data is collected, we could produce more accurate results.

8 Conclusions

In this work, the goal was to build and share a structure of the model for predicting user habits about using sales promotions. As we saw in the literature review it is not a trivial case. There is a lot of gaps that we have to handle, for example, feature with different types (time, numeric, categorical, etc.) or scale. Based on human habits, the webshop's data is often log scaled, and sparse which makes it difficult for the model to find optimal parameters. There are now countless solutions to deal with this problem, like scaling, normalizing, skewing data, or find the most relevant features. Based on these methods, finally we identified a solution for our problem with relatively good accuracy results. For the classification problem we have found that XGBoost gives the best model, while the second solution is not that clear. Based on our results as at first glance LightGBM (LGBM) could be the right choice.

Before making our decision, we need to know the structure of the model. LGBM is a very popular solution, because of its speed and accuracy. It has happened because LGMB grows tree vertically while other algorithms like Xgboost, Gboost grow trees horizontally. Put it differently, LGBM grows tree leaf-wise while another algorithm grows level-wise. LGBM is giving the best solution for our task, but there is some gap, which overshadows our success. However, it is sensitive to overfitting, especially on small dataset. There is no threshold on the number of rows but researchers suggest to use it only for data with $10,000+$ rows. This model hence cannot be used for new promotions that only used by a small amount of user. In the light of this, in the final model, we used linear regression which gives almost the same results as LGBM.

References

[1] Ahmed, A., Low, Y., Aly, M., Josifovski, V., and Smola, A. J. Scalable distributed inference of dynamic user interests for behavioral targeting, In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, 2011
https://doi.org/10.1145/2020408.2020433

[2] Aly, M., Hatch, A., Josifovski, V., and Narayanan, V. K. Web-scale user modeling for targeting. Proceedings of the 21st International Conference on World Wide Web, 2012
https://doi.org/10.1145/2187980.2187982

[3] An, M., Kim, S. Neural User Embedding from Browsing Events. In: Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track. ECML PKDD 2020
https://doi.org/10.1007/978-3-030-67667-4_11

[4] Banerjee, A., and Ghosh, J. Clickstream Clustering using Weighted Longest Common Subsequences. In: The Web Mining Workshop at the 1st SIAM Conference on Data Mining, 2001

[5] Bozanta, A., and Kutlu, B. Developing a Contextually Personalized Hybrid Recommender System, Mobile Information Systems, Article ID 3258916, 2018
https://doi.org/10.1155/2018/3258916

[6] Breiman L. Random forests - random features. Technical Report 567, Statistics Department, University of California, Berkeley, 1999

[7] Burke, R. Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12(4), 331-370, 2002
https://doi.org/10.1023/A:1021240730564

[8] Chen T. and Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794, 2016
https://doi.org/10.1145/2939672.2939785

[9] Cheng, H., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, T., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., Shah H. Wide & Deep Learning for Recommender Systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 2016 https://doi.org/10.1145/2988450.2988454

[10] Cook R.D. Detection of influential observations in linear regression. Technometrics, 19(1):15-18, 1977
https://doi.org/10.2307/1268249

[11] John N. D. and Ratcliff D. Generalized iterative scaling for log-linear models. The Annals of Mathematical Statistics, 43(5):1470-1480, 1972

[12] Çano, E., Morisio, M. Hybrid recommender systems: A systematic literature review. Intelligent Data Analysis, 21(6), 1487-1524, 2017 https://doi.org/10.3233/IDA-163209

[13] Essex, D. Matchmaker, matchmaker. Communications of the ACM, 52(5):16-17, 2009. https://doi.org/10.1145/1506409.1506415

[14] P. Geurts, D. Ernst., and L. Wehenkel, Extremely randomized trees. Machine Learning, 63(1), 3-42, 2006. https://doi.org/10.1007/s10994-006-6226-1

[15] Grbovic, M., Radosavljevic, V., Djuric, N., Bhamidipati, N., Savla, J., Bhagwan, V., and Sharp, D. E-commerce in Your Inbox: Product Recommendations at Scale. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015 https://doi.org/10.1145/2783258.2788627

[16] Hoerl, A. E., Kennard, R. W., Ridge Regression: Applications to Non-Orthogonal Problems. Technometrics 12(1), 69-82, 1970 https://doi.org/10.2307/1267352

[17] A. Ibrahem, A. Osman, A. N. Ahmed, M. F. Chow, Y. F. Huang, A. El-Shafie. Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia, Ain Shams Engineering Journal, 12(2), 1545–1556, 2021

[18] James, G. Majority vote classifiers: theory and applications. PhD thesis, Stanford University, 1998

[19] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T. Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: Advances in neural information processing systems, pp. 3146-3154, 2017.

[20] Koehn, D., Lessmann, S., Schaal, M. Predicting online shopping behaviour from clickstream data using deep learning. Expert Systems with Applications, 150, 113342, 2020 https://doi.org/10.1016/j.eswa.2020.113342

[21] Li, Z., Kulhanek, R., Wang, S., Zhao, Y., Wu, S. Slim Embedding Layers for Recurrent Neural Language Models. In: Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[22] G. Liu, T. T. Nguyen, G. Zhao, W. Zha, J. Yang, J. Cao, M. Wu, P. Zhao, W. Chen. Repeat Buyer Prediction for E-Commerce. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, 2016. https://doi.org/10.1145/2939672.2939674

[23] A. Martínez, C. Schmuck, S. Pereverzyev, C. Pirker, M. Haltmeier. A machine learning framework for customer purchase prediction in the non-contractual setting. European Journal of Operational Research, 281(3)588–596, 2020 https://doi.org/10.1016/j.ejor.2018.04.034

[24] Park, T., Casella, G., The Bayesian Lasso. Journal of the American Statistical Association 103(482), 681-686, 2008 https://doi.org/10.1198/016214508000000337

[25] Sidana, S. Recommendation systems for online advertising. Computers and Society [cs.CY]. Université Grenoble Alpes, 2018.

[26] Vieira, A. Predicting online user behaviour using deep learning algorithms. arXiv preprint arXiv:1511.06247, 2015

[27] Weiss, A. A Comparison of Ordinary Least Squares and Least Absolute Error Estimation. Econometric Theory, 4(3), 517-527, 1988 https://doi.org/10.1017/S0266466600013438

[28] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan. A dynamic recurrent model for next basket recommendation. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 729–732, 2016 https://doi.org/10.1145/2911451.2914683

[29] Zhang, Y., and Pennacchiotti, M. Predicting purchase behaviors from social media, In: Proceedings of the 22nd International Conference on World Wide Web, 2013 https://doi.org/10.1145/2488388.2488521