6th Conference on Information and Knowledge Technology (IKT 2014), May 28-30, 2014, Shahrood University of Technology, Tehran, Iran

A Study on Prediction of User's Tendency Toward Purchases in Websites Based on Behavior Models

Emad Gohari Boroujerdi, Soroush Mehri, Saeed Sadeghi Garmaroudi,
Mohammad Pezeshki, Farid Rashidi Mehrabadi, SeyyedSalim Malakouti,
and Shahram Khadivi
Department of Computer Engineering and Information Technology
Amirkabir University of Technology
Tehran, Iran 15875–4413
Emails: {e.gohari.b, soroushmehr, ssadeghig, m.pezeshki,
frashidi, salim_m, khadivi}@aut.ac.ir

***Abstract*—Nowadays customers would rather buy their needs online than visiting a retail store because of many reasons such as saving time. Therefore, in order to increase efficiency of online shopping websites, many companies have invested in researches toward prediction of users purchases and recommendation systems that may help and motivate a user to buy products that he may be interested in. However, most efforts in this area has been around classification and predictions based on users interests in specific types of products. In this paper, we have studied efficiency of numerous algorithms toward building a classification model to predict the probability of a complete purchase by users only based on their behavior models in the system and regardless of their interest. Therefore, we experimented accuracy of different algorithms and proposed a novel classification model that is able to predict whether a user will be interested in buying a certain set of products that are placed in the online shopping cart or not.**

Keywords*—Feature Selection, Classification, User Behavior, Knowledge Discovery, Machine Learning, Data Mining, Purchase Intention

I. Introduction

There are different ways of advertising to people online. Some involves displaying the same advertisements to everyone who visits a particular website. The aim of this approach is telling customer about products or services that customer is likely to be interested in. Organizations and companies have always employed information about their customers to market goods and services to them. This has appeared to have a great influence on customers' satisfactory and ease of purchase and is an effective approach to improving companies' online markets. Data mining tasks are used to help both markets and customers to provide better recommendations. Many data mining and machine learning researchers have worked on order prediction and classification of customers, commonly treating it as a basic voting process (asking customers to LIKE the more desirable goods) and recommend other possible desirable products based on their user interest models.

In this paper, we have studied efficiency of different machine learning algorithms toward building more accurate user models that result in better product recommendation. As a consequence, much of our work is based on data pre-processing and extraction of related and suitable features and merging results of individual classifiers. Therefore, first and foremost, we concentrate on finding significant features in the datasets. After pre-processing, the next phase is to utilize classifiers in order to predict customers' purchasing decision using voting methods among chosen classifiers.

However, the main contribution of this study is the fact that we have built our prediction model only based on features describing users behavior and not anything related to his/her preferences and types of products that he/she has purchased before. By user behavior attributes, we mean attributes which are derived from the frequency of user clicks ,web surfing model of the user, etc. In some shopping websites such as eBay customer may desire to buy products in various categories which the common methods of prediction need to consider user interests and available products. In contrast, in our model the only features that is related to the products are those that inform us of the total cost of the shopping basket, minimum and maximum cost of the item in a shopping basket. (See Appendix)

The remainder of the paper enumerates issues and describes how they are manifested in purchase prediction problem. The final sections of this paper discuss the results and suggestions for future works.

II. Problem Description

The studied data set[1] is split into two sets: A training set consisted of 426335 transactions[2] (equal to 50000 sessions) and a test set consisted of 44902 transactions (which consists of 10000 sessions). Each transaction which has 23 attributes

[1]The data set is provided by *Prudsys* company in Spring 2013. The data set is available upon request.

[2]Data were provided in transaction format which means that some sequential transactions present one session that is related to the purchase process of a customer. In other words, our data set consisted of 426335 transactions which present 50000 sessions.

978-1-4799-5660-9/14/$31.00

showing a session's status in a moment during shopping process of a particular customer.

Also, attributes are in both nominal and numerical formats. In addition, the data have different kinds of features which need to be treated with different strategies. Features lie in three different groups. The first group is **user-related** features like *UserAge* and *Gender*, the second one is **session-related** features like *StartHour* of a session and the third one includes **time-related** like number of purchased items (such as *BasketCount*) and value of items in the basket (such as *ClickedSumPrice*). It is important to note that unlike recommender systems, we are not dealing with specific goods. The only information on goods in a "basket" are sum, minimum, maximum of prices, and the number of items.

Customer records (transactions) are separated into two different classes, customers who had ordered selected goods, labeled as Y, and those customers who did not order their selected goods, labeled as N.

III. Proposed Model

As mentioned earlier, structure of studied data set is not like usual classification problems. Hence, our model is new from the perspective of handling this data set. There are seemingly no algorithm that can operate on this data set with its unique problem description. We used logged data from an online purchasing website in which every sample is called a *session* and contains one or usually more than one chronologically ordered *transactions*. Due to the fact that many of the attributes are numeric, we are able to deploy them toward defining new meaning-full features which are able to boost the results in a significant way. Subsequently, before learning and applying classifiers, it is necessary to pre-process the data. To fulfill this aim, *merging transactions into session in a meaningful way*, *feature extraction*, *feature selection and elimination*, *filling missing values*, and *outlier detection* are required to be done. These steps are explained in the following section. A graphical depiction of the final model is shown in Fig. 1.

According to the results provided in Table. II, some of the classifiers show better prediction for Y-labeled instances and some others for N-labeled ones. Therefore, To obtain a more accurate prediction, it is needed to combine the results of different classifiers by voting.

A. Data Pre-processing

One of the most important steps of the knowledge discovery from data is data pre-processing. In the following parts, we are going to describe all of the exploited methods:

Filling Missing Values: As a general fact, the less missing values, the more accurate classifiers can achieve by exploiting related methods.[1] At first, by using fully available features, the training data is clustered. Then, in each cluster, the missing values of each numerical feature are filled by the average of its available part, and for nominal ones the major values are used. Among EM, OPTICS [2], and K-Means, the best clustering algorithms, based on *silhouette coefficient*, was the EM with three clusters for output.[3]

Merging Transactions: As explained in previous section, transactions show a session's status in different moments during shopping process. The sequential dependency between transactions makes it unsuitable for direct use. Based on our experimental results (Section 4), it is clear that without any data manipulation, the results would be weak and inaccurate. In sequential data, changes of a particular feature during a session can be transformed into important information for understanding user's behavior and building prediction model. Each session from our dataset is a number of transactions, each with its own values. However, to make existing classifiers work on this dataset, we decided to transform data by merging transactions of each session into a single record.

It is needed to merge features of time-related group (section 2). Different actions can be taken for transformation. For some features, since that its last value in all transactions demonstrates the whole customers' behavior, their sequences are eliminated and their last value is preserved. For other attributes we added some statistical values calculated based on sequential features such as *correlation* or *standard deviation*. This technique enabled us to create more features for training the prediction model and to reduce the number of records in data set which is more convenient for running algorithms.

Feature Extraction: Finding useful and informative features/attributes is playing a central role in pre-processing phase. To discover these features we applied two different approaches:

- ***Firstly,*** during merging transactions step, we made 68 new features. All of them are derived from original features of data set. However, as well as statistical functions like standard deviation, some heuristic methods were used to create hand-crafted features which we thought would improve the prediction model. For instance, we used the number of clicks on products and the elapsed time to build a feature showing click rate of customer in the website. In some other cases, it was found useful to convert a nominal attribute with n different values to n boolean attributes with value of 0 or 1. Finally, we had 82 features to work on. By using these hand-crafted features in combination with other new features in our model, we succeeded in getting the highest possible results. (explained in Section 4)
- ***Secondly,*** we utilized an automated feature extraction tool called Deep Belief Network (DBN) [4]. To achieve state-of-the-art results, we combined feature engineering with feature learning. The idea is to learn higher-level features on top of hand-crafted ones. In other words, we utilized feature learning, or representation learning which is learning representations and transformations of the data that somehow make it easier to extract useful information out of it, e.g., when building classifiers or other predictors as follows in next parts of this paper.
  To reach this aim, we created a DBN with 4 hidden layers, respectively 100, 50, 25, and 5 nodes in each layer. We trained this network by 50 batches of data, each included 100 sessions. Finally, we again added these new 5 features to our set of features. It should be mentioned that although an interpretation can be made for these new 5 features, since it is not necessary, we waive this part.

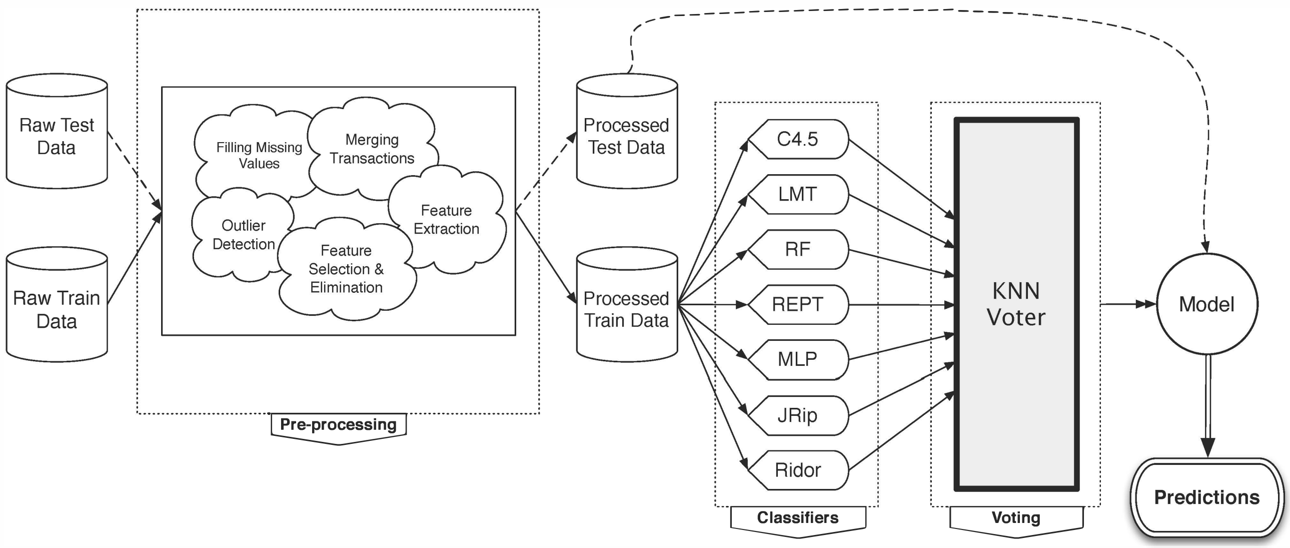

Fig. 1. The sequence of proposed approach is *pre-processing*, *classifying* and *KNN voting*. The most important part is pre-processing step and is applied on both *train* and *test* data to continue in the classification processes.

Feature Selection & Elimination: Feature selection is process of selecting a subset of relevant and useful features. The main idea of feature selection is to eliminate a subset of features which are not very effective in the prediction model. This technique can noticeably reduce the complexity of building the prediction model.

For this phase we have used *backward elimination*. [5] This algorithm starts with the full set of attributes and in each step, we have removed the worst attribute, from F-measure's point of view, among remaining ones. Therefore, primary features are reduced to 45 selected features (about 44% reduction in number of features) and the results have been improved (section 4).

Outlier Detection: One of the negative characteristics of natural data is outliers. It can affect the training algorithm and its prediction model would be misleading. [6] Here, elimination is performed by human analysis. We realized some inconsistencies among features of some sessions which did not obey the common rules extracted from data. For instance, it was founded that a specific customer had visited the online shopping website in a very high rate. This customer's number of visits was 49 times in 3 days while average visit rate of customers was about 5.

Therefore, these kind of sessions have been tagged as outliers and ignored in order to achieve a better result. We believe this does not have negative effects on our final results because existence of such users is not unusual. For instance, a salesperson at a retail store may visit the website frequently to update his information about products prices.

B. Classification Methods

There are numerous classifiers that can be used to predict customer's intention for purchase, among which, tree-based, rule-based, and some other groups stand out. As our experiments on dataset show (Table. II), rule- and tree-based algorithms would have better results.

Amongst mentioned methods, ConjunctiveRule, Ridor [7], and JRip from rule-based family, C4.5 [8], LMT (Logistic Model Trees) [9], ADTree (Alternating Decision Tree) [10], FT (Functional Trees) [11], RandomForest[12], and RandomTree from tree-based family, Logistic Regression, RBFNetwork (Radial-Basis Function Network), Multilayer Perceptron, SPegasos (SVM using stochastic gradient descent), and SVM from function-based family [13] have been tested. The results are represented in Table. II. Detailed information about analyzing and selection of proper algorithms for voting is available in section 4.

C. Voting

After performing multiple pre-processing techniques on the data, different classifiers were used in order to predict purchase intention of customers which gave us different accuracies. Based on predicted instances, algorithms from one group, like tree-based algorithms, were similar to each other and those from different groups of tree-based and rule-based had different results. For example, there was a subset of instances from data set which all tree-based classifiers incorrectly predicted them as non-purchasing; while, this subset already had been correctly predicted by rule-based classifiers. Thus, voting on the results of these two different families of classifiers increases the chance of correct prediction for this subset.

Therefore, we utilized a Weighted Voter to combine our classification models. However, the fact that the classifiers' predicted labels did not vary in a high extent, resulted in only a very small increase in our final accuracy. Most of the classifiers were involved with this issue. Therefore, to solve this problem and gain better result from our voter, we trained a KNN classification model. It was built on top of our classifiers' outputs. We used 7 classifiers which were chosen for building the voter and the best accuracy reached with K = 2 in KNN model based on experimental results.

TABLE I.
STEP-BY-STEP EXPERIMENTAL RESULTS. AS IT IS CLEAR, EACH OF THE PRE-PROCESSING STEPS ARE IMPROVING THE PERFORMANCE. IT IS NOTICEABLE THAT THE PERFORMANCE OF THE ALGORITHM IS IMPROVED SIGNIFICANTLY WHEN THE RECORDS ARE MERGED AND NEW TASK-RELEVANT FEATURES ARE ADDED TO THE DATASET.

Steps of work	Dataset Properties	Precision			Recall			F-Measure		
		Y	N	Avg.	Y	N	Avg.	Y	N	Avg.
Raw Data	# of attributes: 23 # of records: 429013 trans. Algorithm: Reptree	73.1	43.7	64.1	81.8	32.0	66.5	77.2	36.9	64.8
Filling Missing Values	# of attributes: 23 # of records: 429013 trans. Algorithm: Reptree	75.3	45.7	66.2	77.6	42.5	66.9	76.5	44.1	66.5
Merging Transactions	# of attributes: 82 # of records: 50000 sessions Algorithm: Reptree	95.1	97.8	96.6	97.1	96.2	96.6	96.0	97.0	96.6
Feature Selection	# of attributes: 45 # of records: 50000 sessions Algorithm: Reptree	95.5	98.2	96.9	98.0	96.0	96.9	96.7	97.1	96.9
Feature Extraction (DBN)	# of attributes: 50 # of records: 50000 sessions Algorithm: Reptree	95.7	98.3	97.0	98.0	96.1	97.0	96.8	97.2	97.0

TABLE II.
NUMERICAL RESULTS OF TESTED ALGORITHMS

Category	Name	Precision			Recall			F-Measure		
		Y	N	Avg.	Y	N	Avg.	Y	N	Avg.
Decision Tree	FT	94.9	98.2	96.8	97.6	96.1	96.7	96.3	97.1	96.7
	C4.5	95.5	98.3	97.0	98.0	96.0	96.9	96.7	97.1	96.9
	ADTrees	94.5	97.2	95.9	96.8	95.2	95.9	95.7	96.2	95.9
	LMT	95.3	97.7	96.5	96.9	96.5	96.7	96.0	97.0	96.6
	RandomForest	95.8	98.0	97.0	97.8	96.3	97.0	96.7	97.1	97.0
	RandomTree	94.2	94.8	94.5	94.0	95.0	94.5	94.1	94.9	94.5
	REPTree	95.7	98.3	97.0	98.0	96.1	97.0	96.8	97.2	97.0
Function	Logistic	91.8	97.6	94.9	97.4	92.5	94.8	94.5	95.0	94.8
	RBFNetwork	90.2	97.1	93.9	96.9	90.9	93.7	93.4	93.9	93.7
	MultilayerPerceptron	95.0	97.7	96.4	97.4	95.5	96.4	96.2	96.6	96.4
	SPegasos	91.0	97.4	94.4	97.1	91.7	94.2	94.0	94.5	94.3
	SMO	89.6	97.3	94.0	96.6	91.6	93.7	93.0	94.3	93.7
Rule-based	ConjunctiveRule	96.2	80.7	87.8	73.0	97.5	86.1	83.0	88.3	85.8
	JRip	95.5	98.2	97.0	98.0	96.0	96.9	96.7	97.1	96.9
	Ridor	94.1	98.3	96.5	97.8	95.4	96.5	96.0	96.9	96.5

D. *Using the model*

Because this model should be able to predict whether a costumer would buy in its session or not, it seems to be important to describe how it would be used in a potential online website. Assuming that a new user is visiting the website, whether he is a new user or a returning user with a *UserID*. The logging system records every action of the user, including changes in value of basket, purchasing steps and so forth. At the beginning, we can create a new record representing the merged transactions of the user's session after needed *pre-processes*. Certainly, in the beginning of a session this record just includes user- and session-related, but not time-related attributes. As the user surfs the website, new transactions will be added to its session and its merged record will be updated by recalculating the time-related attributes from transactions of this session so far. Then this record is ready to be classified in our model.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The outcomes of individual tested algorithms (mentioned in section 3.2) were optimized by applying brute force on their parameters (Fig. 2 shows an example for the corresponding results of C4.5 classifier). We have chosen seven of these algorithm with the highest prediction accuracy to be passed to the voting algorithm. Doing so made final model to achieve much better precision and recall than each of individual ones. Using heterogenous algorithms with different power (as indicated in the above-mentioned table) to predict various parts of data set was central for achieving such a result. In other words, the weak point of one algorithm is the strong point of another algorithm. According to reasons mentioned in the preceding sections, C4.5, LMT, RandomForest, REPTree, MultiLayer Perceptron, JRip, and Ridor have been chosen which covered the most part of the data set with correct predictions. Therefore, records which were misclassified by one algorithm, were classified correctly using another one. However, as explained before, there are some very unique and odd behaviors among users that could not be modeled.

V. CONCLUSION

This paper reported on a study in "predicting purchase" in online shopping based on surfing behavior of customers.

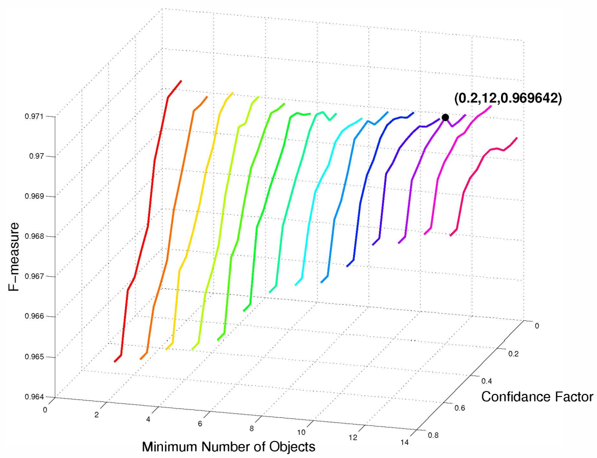

Fig. 2. An example of brute forcing on the C4.5 parameters. The maximum F-measure point, which is highlighted on the figure is $(C = 0.2, M = 12, F = 0.9696)$.

TABLE III.
NUMERICAL RESULTS OF VOTERS

Voter	Precision	Recall	F-Measure
Weighted	97.11	97.19	97.14
KNN	97.21	97.20	97.20

We used logged data from an online purchasing website and showed that it is feasible to classify a session of a visitor into "purchasing" and "non-purchasing". Despite the absence of information on key variables such as *UserAge* and *LastOrder*, our model performed surprisingly well. Most notably, our voting algorithm alongside with feature extraction and merging transactions, achieved **97.20%** accuracy (Table. III).

Our algorithms were drawn from statistics (filling missing values and feature extraction) and data mining/machine learning (rule- and tree-based algorithms, clustering, weighted voting, and KNN voting). Each algorithm was tailored to the problem at hand (e.g., we devised an appropriate statistical feature extraction, merging transactions, classifier selection for voting), and the algorithms were combined using a voter to improve their predictive accuracy.

To wrap it up, we believe that we have been able to propose a classification model for predicting user's purchase decision based on their behaviors. And we find it tremendously important since in websites such as eBay, users tend to buy various products and may not fit in a specific category. Therefore, recommender systems or method which tend to build models based on users' interests may not be so effective in this areas. However, we were able to predict user purchase with a high desirable accuracy.

VI. FUTURE WORKS

Future works will include enhancing the quality of both hand-crafted and extracted features. It would be useful if we eliminate those features which are not related to the purchase prediction task. Furthermore, eliminating non-relevant features would help us to use some slow classifiers such as SVM that were not practical previously because of the large number of input data dimensions. Moreover, applying some state-of-the-art methods such as those mentioned in [14] would enhance the efficiency of the classifiers significantly. These areas will be addressed in future phases of this study.

REFERENCES

[1] E. Acuna, C. Rodriguez, "The treatment of missing values and its effect on classifier accuracy," In Classification, Clustering, and Data Mining Applications, pp. 639-647 (2004).

[2] M. Ankerst,M. M. Breunig,H. P. Kriegel, J. Sander, "OPTICS: ordering points to identify the clustering structure," ACM SIGMOD Record, 28(2), pp. 49-60 (1999).

[3] I. H. Witten, E. Frank, "Data Mining: Practical machine learning tools and techniques," Morgan Kaufmann (2005).

[4] Y. Bengio, A. C. Courville, & P. Vincent, "Unsupervised feature learning and deep learning: A review and new perspectives," CoRR abs/1206.5538 (2012).

[5] I. Guyon, & A. Elisseeff, "An introduction to variable and feature selection," The Journal of Machine Learning Research, 3, pp. 1157-1182 (2003).

[6] V. J. Hodge, & J. Austin, "A survey of outlier detection methodologies," Artificial Intelligence Review, 22(2), pp. 85-126 (2004).

[7] P. Compton, R. Jansen, "A philosophical basis for knowledge acquisition," Knowledge acquisition, 2(3), pp. 241-258 (1990).

[8] J. R. Quinlan, "C4. 5: programs for machine learning (Vol. 1)," Morgan kaufmann (1993).

[9] N. Landwehr, M. Hall, E. Frank, "Logistic model trees," Machine Learning, 59(1-2), pp. 161-205 (2005).

[10] Y. Freund, L. Mason, "The alternating decision tree learning algorithm," ICML, pp. 124-133 (1999).

[11] J. Gama, "Functional trees," Machine Learning, 55(3), pp. 219-250 (2004).

[12] L. Breiman, "Random forests," Machine learning, 45(1), pp. 5-32 (2001).

[13] T. Joachims, "Training linear SVMs in linear time," In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 217-226 (2006).

[14] J. Honaker, G. King, "What to do about missing values in time-series cross-section data," American Journal of Political Science, 54(2), pp. 561-581 (2010).

APPENDIX

TABLE IV.
FEATURES OF RAW DATASET

	Feature	Description
1	sessionNo	running number of the session
2	startHour	hour in which the session has begun
3	startWeekday	day of week in which the session has begun
4	duration	time in seconds passed since start of the session
5	cCount	number of the products clicked on
6	cMinPrice	lowest price of a product clicked on
7	cMaxPrice	highest price of a product clicked on
8	cSumPrice	sum of of all products clicked on
9	bCount	number of products in basket
10	bMinPrice	lowest price of all products put in the shopping basket
11	bMaxPrice	highest price of all products put in the shopping basket
12	bSumPrice	sum of the prices of all products in the basket
13	bStep	purchase processing step
14	onlineStatus	Indication whether the customer is online
15	availability	delivery status
16	customerID	customer number
17	maxVal	maximum admissible purchase price for the customer
18	customerScore	customer evaluation from the point of view of the shop
19	accountLifetime	lifetime of the customer's account in months
20	payments	number of payments effected by the customer
21	age	age of the customer
22	address	form of address (1:Mr, 2:Mrs, 3:Company)
23	lastOrder	time in days passed since the last order

TABLE V.
SOME OF THE EXTRACTED FEATURES FROM ORIGINAL DATASET WHICH WERE USED TO BUILD BEHAVIORAL MODEL OF USERS (THESE FEATURES ARE DESIGNED TO SHOW USERS' BEHAVIOR IN SHOPPING PROCESS)

	Feature	Description
1	cCountLast	last number of clicks
2	cCountAvgIncrease	mean of increases in number of clicks
3	cSumLast	sum of last clicks
4	cSumAvgIncrease	mean of increases in number of clicks
5	cMinLast	least number of last clicks in a session
6	cMinAvgIncrease	mean of increases in least number of clicks
7	cMaxLast	highest number of clicks in last basket
8	cMaxAvgIncrease	mean of increases in highest number of clicks
9	bCountLast	last number of products in last basket
10	bCountAvgIncrease	mean of increases in number of products
11	bSumLast	sum of last products
12	bSumAvgIncrease	mean of increases in number of products in last basket
13	bMinLast	least number of last products in a session
14	bMinAvgIncrease	mean of increases in least number of products
15	bMaxLast	highest number of products in last basket
16	bMaxAvgIncrease	mean of increases in highest number of products
17	bStepBehavior	mean of customer's purchasing step
18	availabilityLast	last availability
19	availabilityMost	most common availability
20	onlineStatusLast	last status of customer
21	onlineStatusMost	most common status
22	onlineStatusRatio	number of y status divided by number of n status
23	avgCCountPerSec	mean of number of clicks in each second
24	avgBCountPerSec	mean of number of products selected in each second
25	cvCCount	standard deviation of clicks divided by mean of clicks
26	cvBCount	standard deviation of products divided by mean of products