Analysis of Potential Customers and Products Recommendation based on E-commerce Log

Zihao Wang, Ph.D. in Information Studies

Long Island University

Abstract

The need for recommendation systems that provides personalized information has increased in e-commerce. This research developed 2 modules to explored potential customers with purchasing intend and analyze their preferable products based on the e-commerce log. In the first module, the K-nearest neighbor (KNN) and multi-layer perceptron (MLP) neural network are used to mine potential customers. In the second module, a first-order Markov chain model is designed to predict products that users are going to buy. The two modules are tested by the RecSys 2015 challenge dataset. In the first module, the KNN has better performance than MLP neural network. The accuracy of 90% means it can recognize potential customers. In the second module, the accuracy of 97% shows the model can predict users’ preferable products.

Keywords: information services, e-commerce, recommendation system, user preference, machine learning, neural network, K-nearest neighbor, Markov chain.

Analysis of Potential Customers and Products Recommendation based on E-commerce Log

# Introduction

Online service personalization has become growing research and commercial interest during the last 5 years (Romov & Sokolov, 2015). E-commerce recommendation systems can offer users’ interesting products. Both customers and business companies can benefit from personalization systems (Ben-Shimon et al., 2015). Customers can receive relevant products, retrieve information precisely, and achieve shopping goals faster. E-commerce vendors can generate more benefits based on the recommendation systems. However, how to precisely know users’ needs is a big challenge.

This study explored potential customers who have purchasing intend and what products they are going to buy. Two research questions are proposed: 1) Is a user going to buy items in a given session; 2) If a customer wants to buy, what would be the products this user will buy.

The organization of this paper is described as follows: the first chapter is the literature review. Methods for mining potential users with purchasing intend and for product recommendation are introduced. The second chapter is the methodology, the data collection, methods for finding potential customers and their preferable items are described. In the third chapter, the results of the 2 methods are described and discussed. Finally, the conclusion is summarized and the future applications are described. The process of this research is illustrated in Figure 1.

**Select users who have willingness to buy products**

**Select features for purchasing intend prediction**

**Use neural network and KNN to analysis if users want to buy products**

**Generate a set of users with purchasing intend**

**Calculate accuracy for training and testing dataset**

**Compare the performance between 2 models**

**Predict users items preferences if they want to buy products**

**Implement Markov chain to predict users’ preferable products**

**Generate a set of users with predicted users’ preferable products**

**Calculating accuracy of selected dataset**

Figure 1. The process of this research

# Literature Review

## Theory of mining potential users with purchasing intend

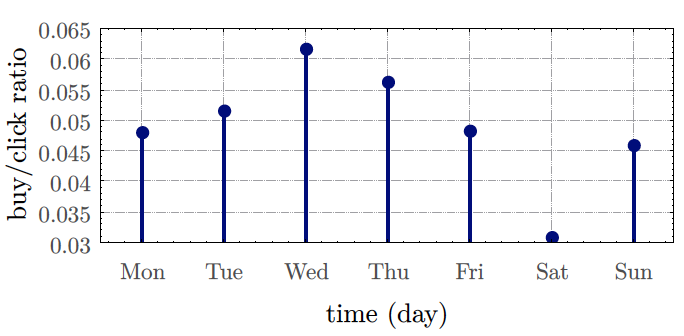
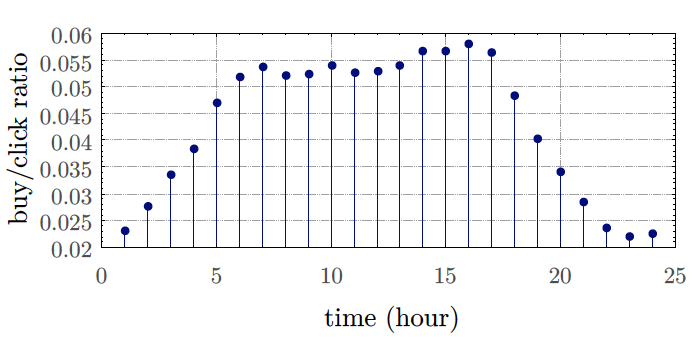
Romov and Sokolov proposed 7 features that are related to purchasing intend, see Table 1 (Romov & Sokolov, 2015).

Table 1. Features related to purchasing intend (Romov & Sokolov, 2015)

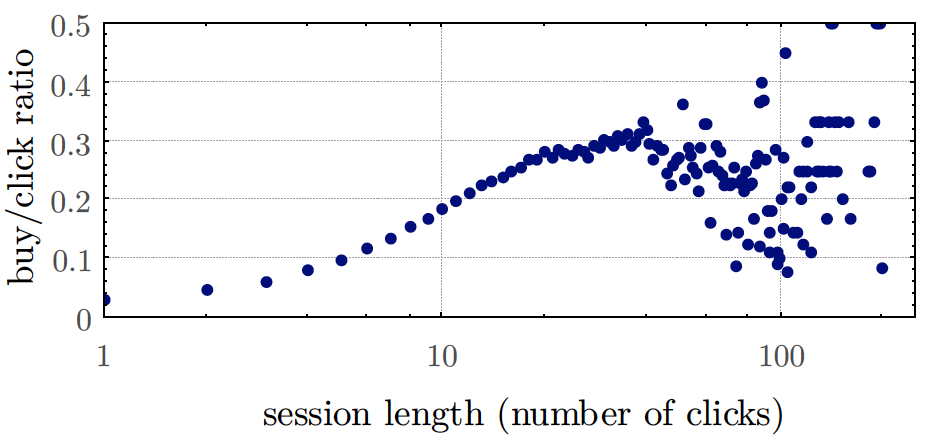
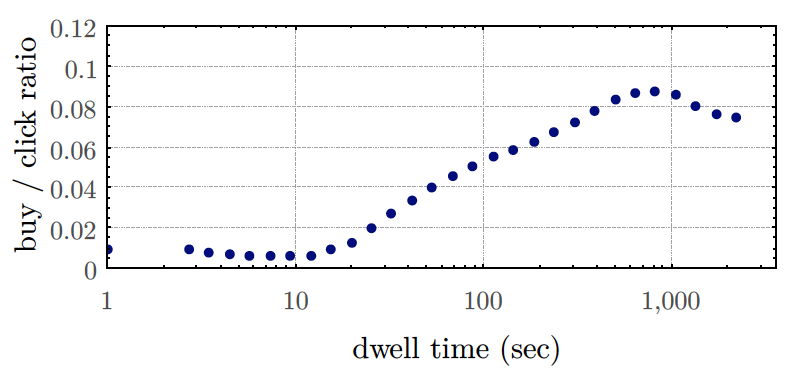
|  |  |  |
| --- | --- | --- |
|  | Feature Description | Number/Type |
| 1 | Numerical time features of the start/end of the session (month, day, hour, minute, second, etc.) | 2 × 7 Number |
| 2 | Categorical time features of the start/end of the session (month, day, month-day, month-day-hour, hour, minute, weekday) | 2 × 7 Category |
| 3 | Length of the session in seconds | 1 Number |
| 4 | Number of clicks, unique items, categories, and item-category pairs in the session | 4 Number |
| 5 | Top 10 items and top 5 categories by the number of clicks in the session | 15 Category |
| 6 | IDs of the first/last item clicked at least k = 1, 2 . . . , 6 times in the session | 12 Category |
| 7 | Vector of click numbers and total durations for 100 items and 50 categories that were the most popular in the whole training set | 150 × 2 Number |

The linear model also is used to find users with purchasing intend (Pálovics et al., 2015). It uses features of numbers of clicks on the item, the item, the day of the week, the hour of the day, dwell time on the item, and the maximum dwell time in the session. It used “buy/click ratio”, a fraction of the number of buyer sessions in some given amount of sessions, to analyze the relationship between each feature and the buy/click ratio for all sessions.

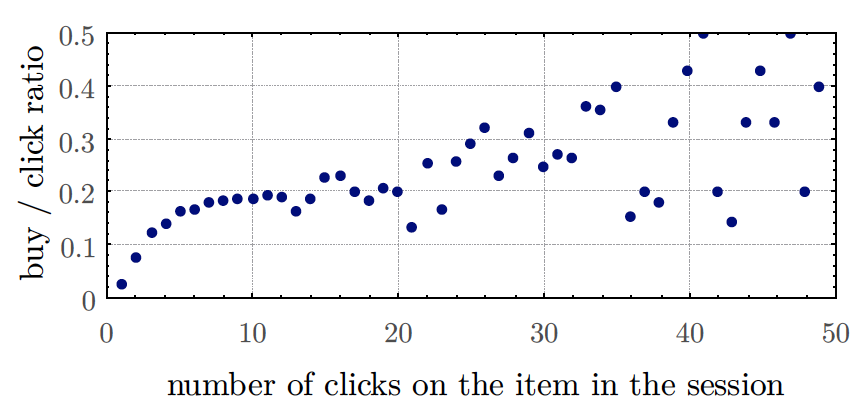
Figure 2 (a) showed the relationship between the week of days and the buy/click ratio. Most users want to buy products on Wednesday. Figure 2 (b) showed the relationship between hours of days and the buy/click ratio. Most people prefer shopping in the afternoon. Figure 2 (c) explored the relationship between session length and the buy/click ratio. If the session length between 1 and 55, with the increase of session length, the more probability that users want to buy. Figure 2 (d) showed the relationship between the dwell time and the buy/click ratio. The more dwell time, the more buy/click ratio. Figure 2 (e) showed the relationship between the number of items in a session and the buy/click ratio. The buy/click ratio increase when the number of searched products increases.

(a) (b)

(c) (d)



(e)

Figure 2. Relationship between each feature and the buy/click rate (Pálovics et al., 2015)

Multi-Layer Perceptron (MLP) neural network with a single hidden layer is used to find users with purchasing intend (Chen et al., 2015). It used the tanh activation function in the hidden layer and the SoftMax activation function in the output layer. The structure of the MLP neural network is shown in Figure 3.

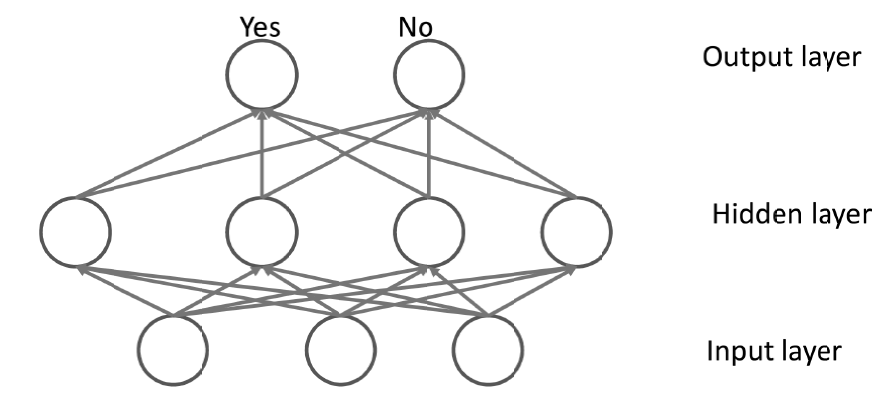
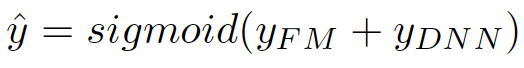


Figure 3. The single hidden layer MLP neural network (Chen et al., 2015)

DeepFM is proposed by combining a factorization machine with a neural network to predict the click-through rate in the recommender system (Guo et al., 2017). The input is a multi-dimension vector and the output is a value chosen from 0 or 1 which represents click behaviors (1 means user clicked the item, and 0 otherwise). The factorization model and deep learning model are combined and output a value by using the sigmoid function. The output function is defined as:



In this function, ,yFM is the output of the factorization machine and yDNN is the output of the neural network.

In the deep learning structure of DeepFM, there are 2 hidden layers. Each hidden layer has 32 neurons with a ReLU activation function. The structure is shown in Figure 4.



Figure 4. The structure of DeepFM (Guo et al., 2017)

## Theory of Customers preferable products prediction

#### Association rule

Association rule learning is a rule-based machine learning method for product recommendations. There are 2 important concepts for association rule learning: support and confidence. Definitions of these concepts are described below.

Let X, Y are item sets, X⇒Y is an association rule, and T is a set of transactions.

Support is an indication of how frequently the itemset appears in the dataset. The support of X with respect to T is defined as the proportion of transactions that contains X. It is defined as:

.

Confidence is an indication of how often the rule is true. It is defined as:

.

#### Markov chain

A hybrid recommendation system that unifies similarity models with Markov chains for sequential recommendation is developed (Yang et al., 2020). The system used both sequential patterns and item contents similarity to analyze the relationship between items.

First, it used a high-order Markov chain to model sequential patterns over time series. Second, the most frequent items are predicted by the high-order Markov chain. Third, the content similarity is used to recommend other similar products from the item list. The process of this study is shown in Figure 5.

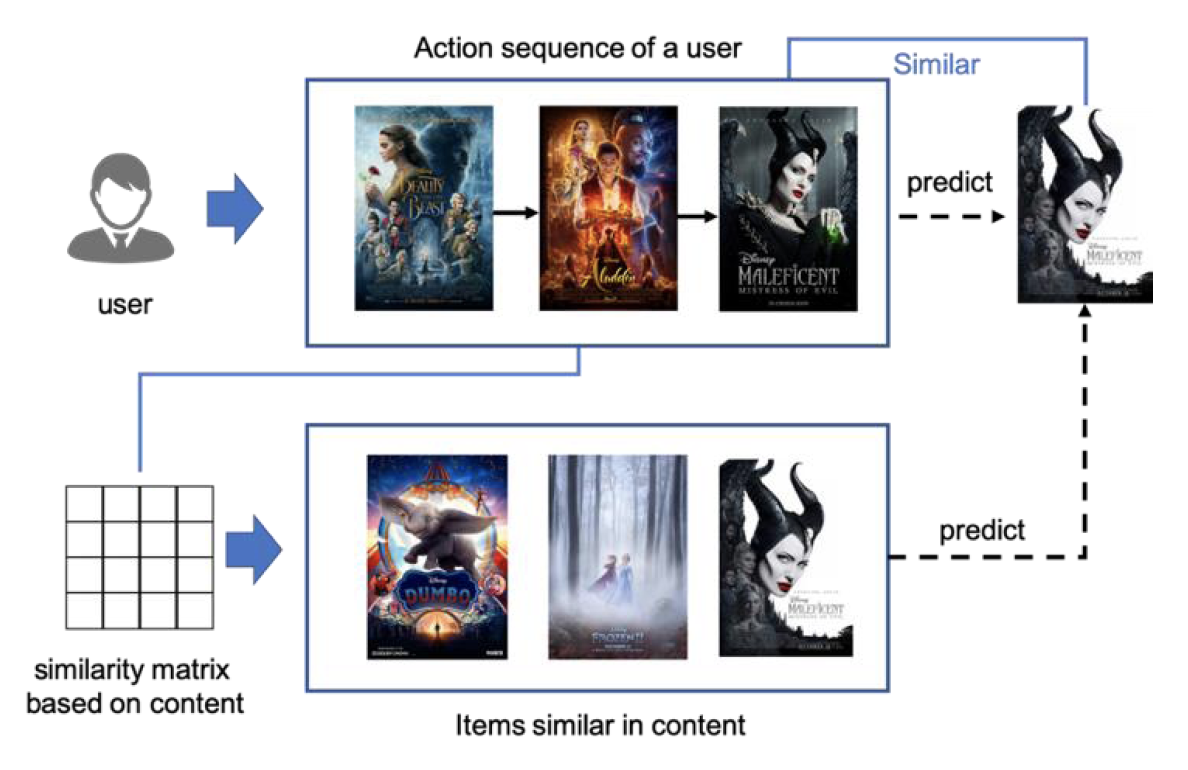


Figure 5. An example of recommendation system by Markov chain and similarity (Yang et al., 2020)

In Figure 5, a user searched a sequence of movies (Beauty and Beast, Aladdin, and Maleficent 1). The Markov chain predicts the move “Maleficent 1” where this user might watch next time. Based on the category of “Maleficent 1”, similar movies are recommended from the movie list, such as “Maleficent 2”.

The Markov chain represents sequential patterns by building the relationship between items and the transition probabilities from the transition matrix. The inner product between the vector of the probability distribution of the current state and the transition matrix represents the transition probability of the next step. Markov chain has better performance on the sparse transaction data.

# Methodology

### Data collection

This study used e-commerce search logs from RecSys 2015 challenge (Ben-Shimon et al., 2015). It provides 2 datasets: clicked events and bought events. The columns of clicked events dataset are described in Table 2.

Table 2. Description of clicked events dataset

|  |  |
| --- | --- |
| Item | Description |
| Session ID | The id of the session is represented as an integer number. One session ID represents an unique user. In one session there are one or many clicks |
| Timestamp | The time when the click occurred. |
| Item ID | The unique identifier of the item that has been clicked, represented as an integer number |
| Category | The context of the click |

An example of clicked event data is described in Figure 6.

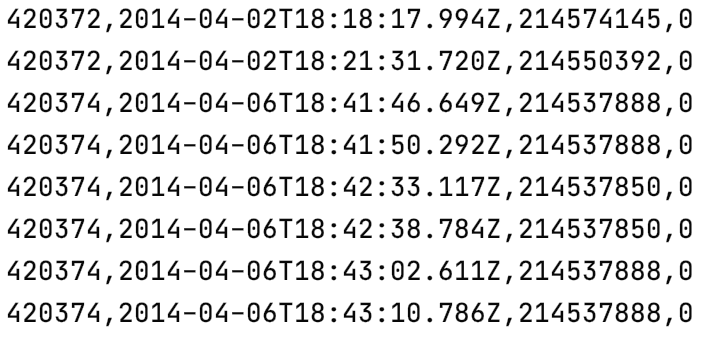


Figure 6. An example of click event data

The bought events dataset contains bought products. The columns are described in Table 3.

Table 3. The description of bought events data

|  |  |
| --- | --- |
| Item | Description |
| Session ID | The id of the session is represented as an integer number. One session ID represents an unique user. In one session there are one or many buying events |
| Timestamp | The time when the buy occurred. |
| Item ID | The unique identifier of item that has been bought, represented as an integer number |
| Price | The price of the item represented as an integer number |
| Quantity | The quantity in this buying, represented as an integer number |

An example of the bought events data is described in Figure 7.

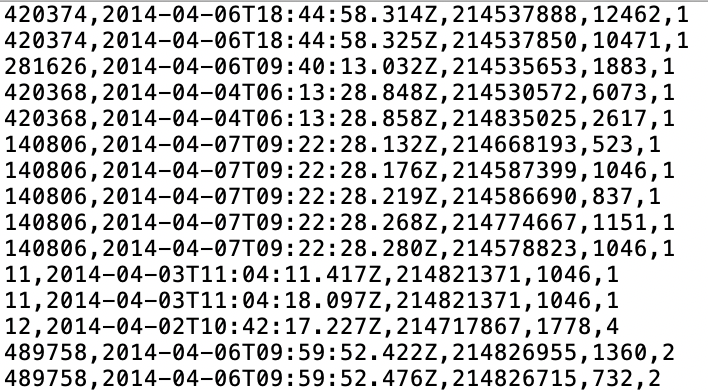


Figure 7. An example of the bought events data

The session id in the bought event data comes from the clicked event data. That is if an user had purchasing intend, the session id appeared in the bought event data.

### Mining potential customers with purchasing intend

The first step is finding effective features that can be used for training machine learning classifiers. This research implemented 2 machine learning models: MLP neural network and KNN classifier. The features used were 1) the day of the starting week, 2) the hour of the starting day, 3) the day of finishing week, 4) the hour of the finishing week, 5) numbers of clicks on the session, 6) maximum dwell time in the session, 7) numbers of items in the session. The feature vector for training is a vector with a length of 7. The description and types for each element are illustrated in Table 4.

Table 4. Features for purchasing intend analysis

|  |  |
| --- | --- |
| Feature Description | Number/Type |
| Time features of the start/end of the session (day of week, hour) | 4 Category |
| Length of the session in seconds (dwell time) | 1 Number |
| Number of clicks | 1 Number |
| Number of items | 1 Number |

#### Neural network method

This study implemented a MLP neural network and referred to the structure of DeepFM. A neural network with 3 hidden layers is constructed with tanh activation function in each hidden layer. The first and second hidden layers had 32 neurons and the third hidden layer had 16 neurons. The output layer had 2 neurons with a SoftMax activation function and cross-entropy loss function. The structure of the neural network is demonstrated in Figure 8.

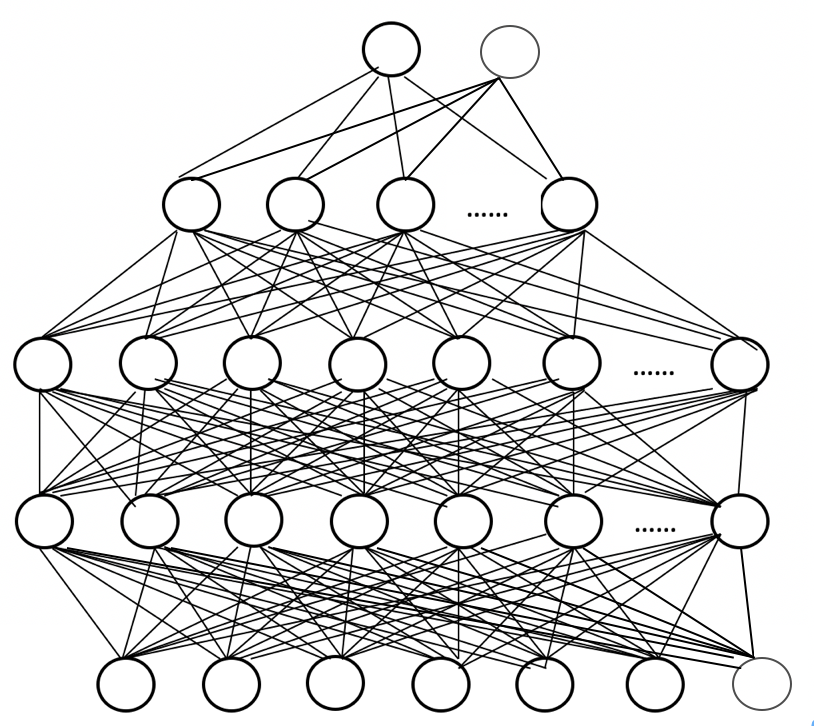


Figure 8. The structure of the neural network

#### K-nearest neighbor method

The K-nearest neighbor (KNN) algorithm is one of the machine learning algorithms of classification. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. Details are illustrated in Figure 9.



Figure 9. Illustration of KNN

The green point needs to be classified into blue or red. In this example, k=3. The green dot finds its closest 3 neighbors. In the 3 neighbors, the majority number of the category is the green dot’s category which is the triangle.

The chosen best k value of this study is illustrated in the result chapter.

### Customers preferable products prediction based on Markov chain

The first-order Markov chain is used to predict products that users would like to buy based on their search sequence. There are 4 steps in this part. The first step is selecting the unique selected product sequence (USP) set for each user. If the size of USP is less than 5, then all items in the USP are seen as the prediction items. Otherwise, the second step is calculating the item probability distribution and the transition matrix for the current state based on the current user’s unique selected products. Third, according to the Markov chain algorithm, the probability distribution of buying products is computed based on the inner product of the current probability distribution and the transition matrix. Forth, the searched products are sorted by the decreasing order of the probability distribution of buying products, and the first half of items are seen as the final prediction.

The process for selecting the unique selected products (USP) sequence for each user is described in Algorithm 1.

|  |
| --- |
| Algorithm 1. The process of selecting the unique selected products |
| for each user:  queue=empty  for item in searched sequence based on time series:  if item not in queue:  enqueue(item) |

A file that recorded all users and their unique selected products is generated from the clicked events dataset. An example of the file is shown in Table 5.

Table 5. An example of users and their unique selected products

|  |  |
| --- | --- |
| Session ID | Unique selected products |
| 1 | 214536502, 214536500, 214536506, 214577561 |
| 2 | 214662742, 214825110, 214757390, 214757407, 214551617 |
| 3 | 214716935, 214774687, 214832672 |
| 4 | 214836765, 214706482 |
| 6 | 214701242, 214826623 |

To get the probability distribution of the current state, the actual searched products (ASP) set is summarized for each user. The process is described in Algorithm 2.

|  |
| --- |
| Algorithm 2. The process to get the actual searched products |
| for each user:  queue=empty  for item in searched sequence based on time series:  enqueue(item) |

A file that recorded all users and their unique selected products is generated from the clicked events dataset. An example of the file is shown in Table 6. The user with session ID No. 2 searched product “214662742” appears once in Table 5 but twice in Table 6.

Table 6. An example of users and their actual selected products

|  |  |
| --- | --- |
| Session ID | Actual selected products |
| 1 | 214536502, 214536500, 214536506, 214577561 |
| 2 | 214662742, 214662742, 214825110, 214757390, 214757407, 214551617 |
| 3 | 214716935, 214774687, 214832672 |
| 4 | 214836765, 214706482 |
| 6 | 214701242, 214826623 |

In current state probability (CSP) distribution is defined as:

The transition matrix is the size of N × N matrix, where N is the size of USP set for each user. Rows and columns are the element order in USP set. the transition probability is defined as:

where means item x appeared before item y in the USP set.

The process of the first-order Markov chain is described in Algorithm 3.

|  |
| --- |
| Algorithm 3. The process of the first order Markov chain |
| all\_users\_USP=Algorithm1  all\_users\_ASP=Algorithm 2  for current\_user\_USP in All\_users\_USP:  if size(current\_user\_USP)<5:  prediction= current\_user\_USP  else:  get transistion matrix for current\_user\_USP  get initial item distribution for current\_user\_USP  compute the buying probability distribution for current\_user\_USP  sort current\_user\_USP by decrease order  prediction=first half items from the sorted current\_user\_USP  return prediction |

# Results

### Mining customers with purchasing intend

The clicked events dataset has 9,249,729 customers who are randomly shuffled and separated into training, validation, and testing datasets. The validation and testing dataset have 10,000 users in each part. And they are the same for testing neural network and KNN. The reminder 9,229,729 users are used for the training of the neural network. Within the 9,229,729 users, the top 1,000,000 users are selected for training the KNN model. The detailed description of the dataset is shown in Table 7.

Table 7. Dataset description

|  |  |  |  |
| --- | --- | --- | --- |
|  | Total number of users | Number of users with buy willingness | Number of users without buy willingness |
| Training MLP | 9,229,729 | 508,621 | 8,721,108 |
| Training KNN | 1,000,000 | 55,042 | 944,958 |
| Validation | 10000 | 529 | 9471 |
| Testing | 10000 | 546 | 9454 |
| Total | 9,249,729 | 509,696 | 8,740,033 |

The confusion matrix is used to evaluate model performance. It’s shown in Table 8.

Table 8. Confusion matrix

|  |  |  |
| --- | --- | --- |
| Predict  Actual | N | P |
| N | TN | FP |
| P | FN | TP |

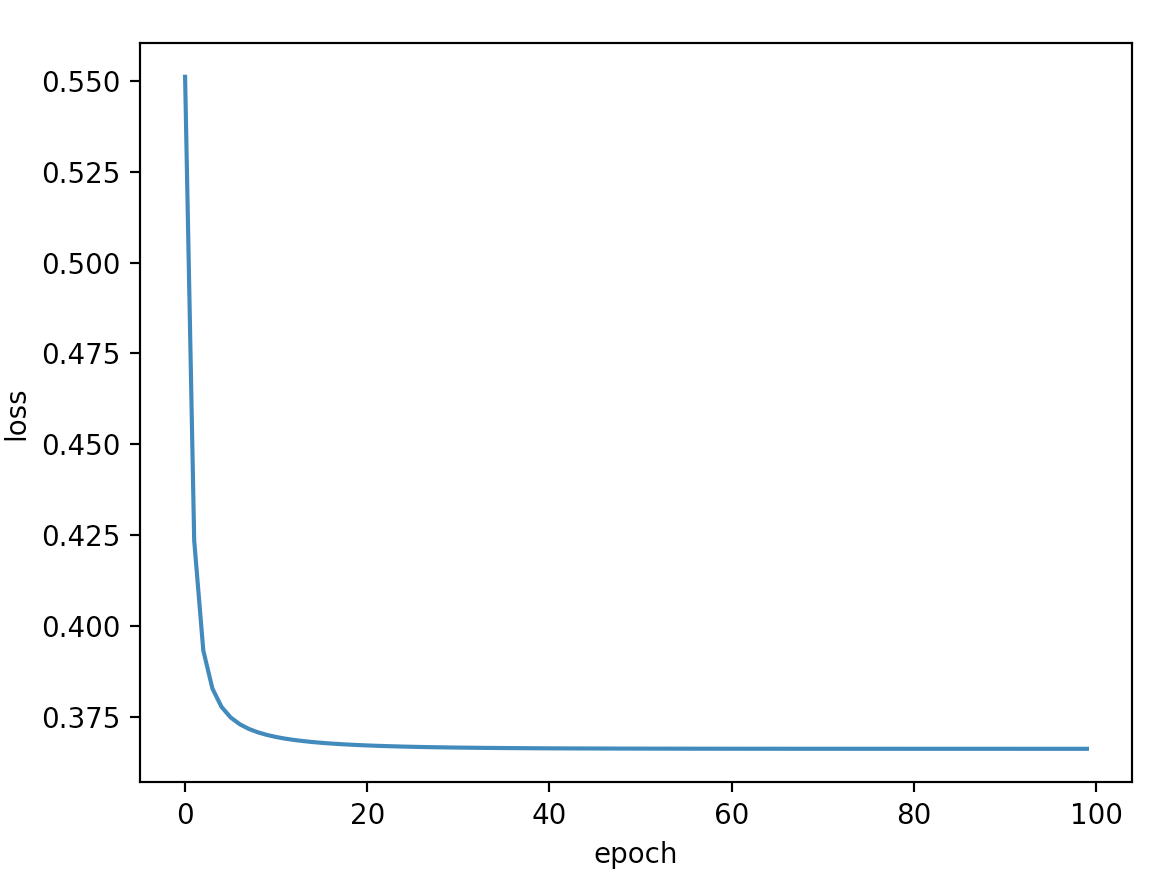
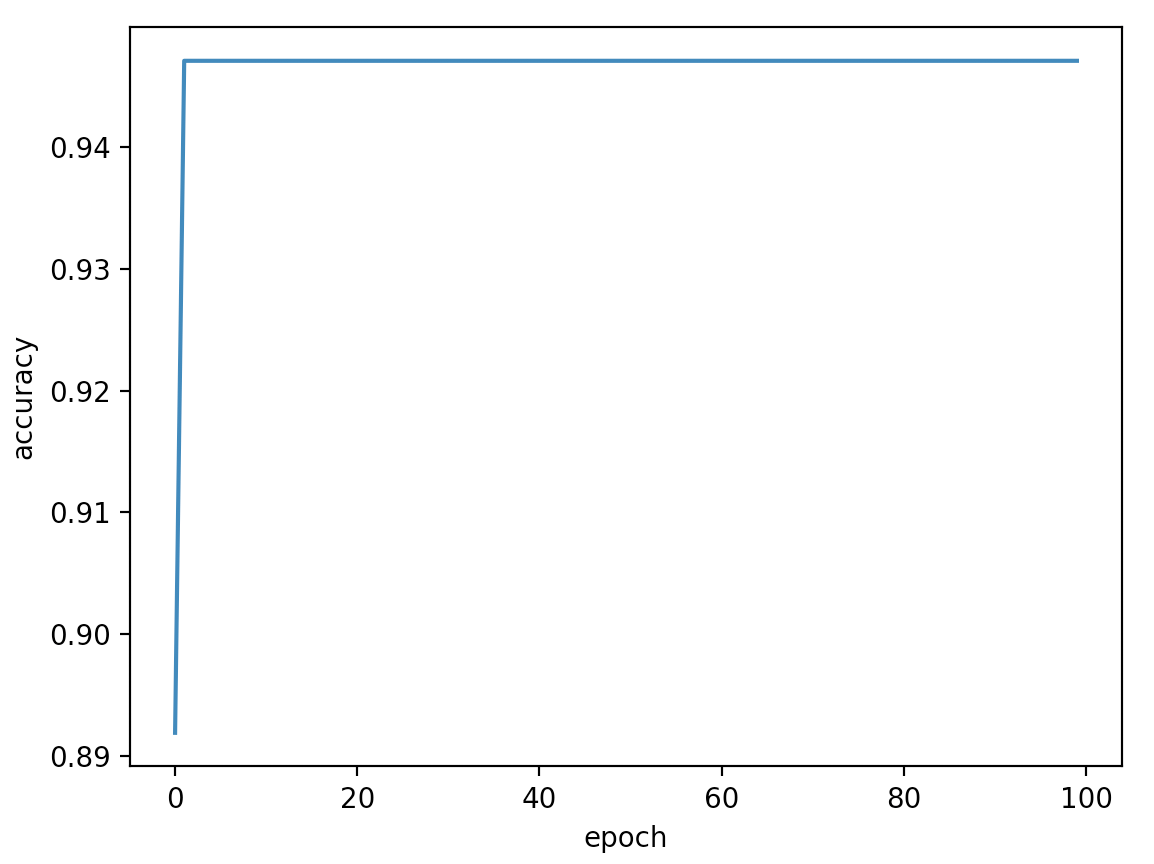
The accuracy and the true positive rate are defined as:

The neural network was trained 100 epochs with batch size 50,000 in each epoch. After the training process, the best result appeared on the initial state with the accuracy of 0.89, and the true positive rate of 0.03 for correctly predicting users with purchasing intend on the validation dataset. On the testing dataset, the respondent accuracy and true positive rate are 0.89 and 0.02. The confusion matrix based on the testing dataset is shown in Table 9.

Table 9. The confusion matrix of MLP neural network

|  |  |  |
| --- | --- | --- |
| Predict  Actual | Without purchase intend | With purchase intend |
| Without purchase intend | 8888 | 566 |
| With purchase intend | 537 | 9 |

The validation loss and accuracy during the 100 epochs are shown in Figure 10 (a) and Figure 10 (b).

(a) (b)

Figure 10. The trend of loss and accuracy of the validation dataset

The loss dramatically decreases within the first 5 epochs and the accuracy dramatically beyond 0.94 after the first epoch. This phenomenon demonstrates the effectiveness of the extracted features in Table 4. However, due to the number of users with purchasing intend is less than users without purchasing intend, the neural network can only detect a few costumers with purchasing intend.

In the KNN model, the k was chosen odd number between 1 and 20. The best k value is 1 with an accuracy of 0.9 and the true positive rate is 0.12 in the validation dataset. Respondent, the accuracy and true positive rate for the testing set are 0.9 and 0.13. The confusion matrix is shown in Table 10.

Table 10. Confusion matrix of KNN classifier

|  |  |  |
| --- | --- | --- |
| Predict  Actual | Without purchase intend | With purchase intend |
| Without purchase intend | 8948 | 506 |
| With purchase intend | 476 | 70 |

Comparing the neural network with the KNN model, the KNN model has better performance than the neural network because it used less training data and got better accuracy and recall rate.

### Customers preferable products prediction

The top 1000 customers with buying behavior in the clicked dataset are selected. For each customer, the actual bought products (ABP) set is generated from the bought products dataset. The process is the same as Algorithm 1. The predicted buying products (PBP) set is computed by the first-order Markov chain model. The accuracy for each user is defined as:

The average accuracy among the 1000 users is 0.97. For the 1000 users, the histogram of accuracy is described in Figure 11.

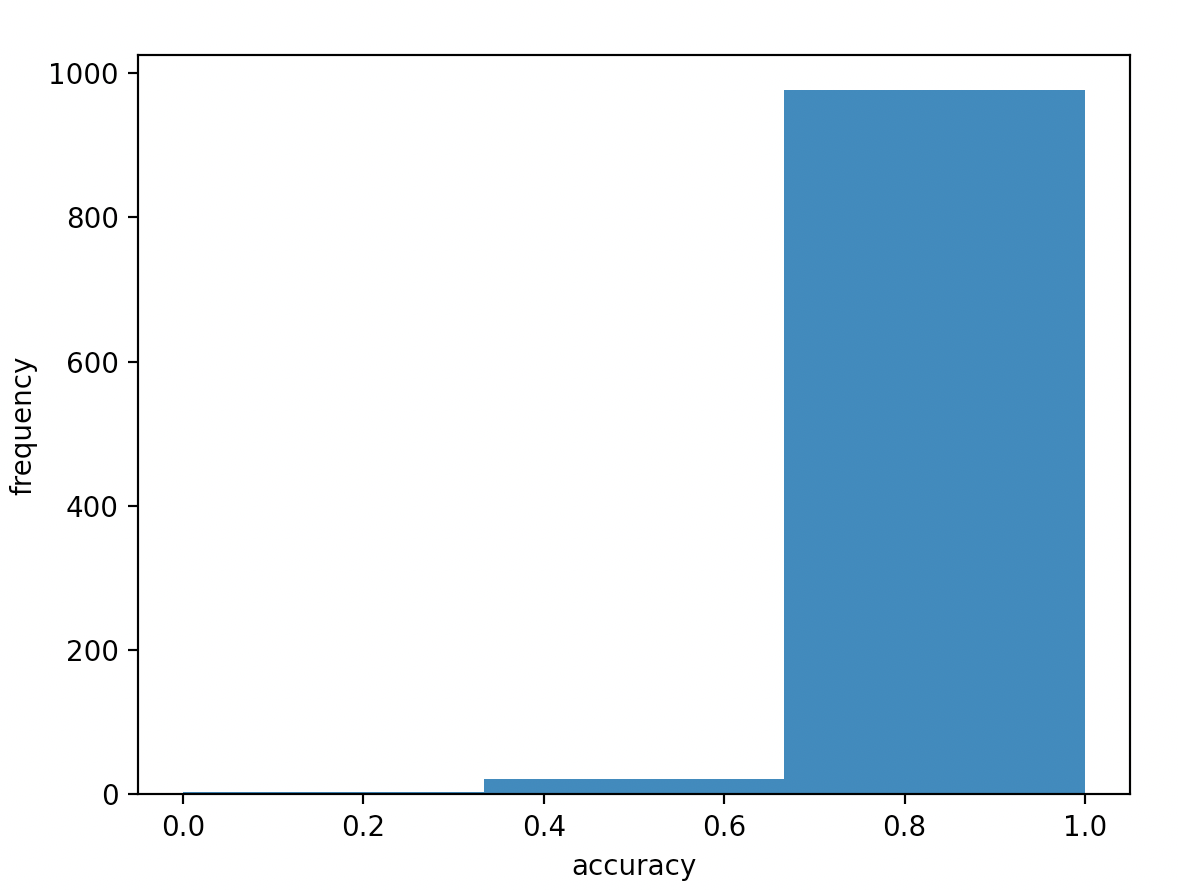


Figure 11. The histogram of accuracy

The histogram shows most users have an accuracy between 0.65 and 1.0. A few users have an accuracy between 0.35 and 0.65. The result shows this model can apply to real-world situations.

# Conclusion

This study implemented 2 modules for mining potential customers and predicting users’ interesting products. In the first task, the KNN method performs better than the MLP neural network. The KNN model has 90% accuracy and a true positive rate of 13%. It illustrates the capacity to recognized users with purchasing intend. But the true positive rate shows the model cannot find enough customers with purchase intention. Therefore, it should take more effort to real application. In the second task, the first-order Markov chain is implemented. It shows an excellent performance with an accuracy of 97%. It can precisely find users’ interesting products. And the histogram of accuracy illustrates it works well and can effectively apply to real applications. In the future, high-performance models for mining potential users are required to explore. A recommendation system that combined the 2 modules can be developed and applied to real e-commerce companies.

References

Ben-Shimon, D., Tsikinovsky, A., Friedmann, M., Shapira, B., Rokach, L., & Hoerle, J. (2015). RecSys Challenge 2015 and the YOOCHOOSE Dataset. *Proceedings of the 9th ACM Conference on Recommender Systems*, 357–358. https://doi.org/10.1145/2792838.2798723

Chen, W., Li, Z., & Zhang, M. (2015). Linear and Non-Linear Models for Purchase Prediction. *Proceedings of the 2015 International ACM Recommender Systems Challenge*, 1–4. https://doi.org/10.1145/2813448.2813518

Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. *ArXiv:1703.04247 [Cs]*. http://arxiv.org/abs/1703.04247

Pálovics, R., Szalai, P., Kocsis, L., Szabó, A., Frigó, E., Pap, J., Nyikes, Z. K., & Benczúr, A. A. (2015). Solving RecSys Challenge 2015 by Linear Models, Gradient Boosted Trees and Metric Optimization. *Proceedings of the 2015 International ACM Recommender Systems Challenge*, 1–4. https://doi.org/10.1145/2813448.2813513

Romov, P., & Sokolov, E. (2015). RecSys Challenge 2015: Ensemble learning with categorical features. *Proceedings of the 2015 International ACM Recommender Systems Challenge*, 1–4. https://doi.org/10.1145/2813448.2813510

Yang, Y., Jang, H.-J., & Kim, B. (2020). A Hybrid Recommender System for Sequential Recommendation: Combining Similarity Models With Markov Chains. *IEEE Access*, *8*, 190136–190146. https://doi.org/10.1109/ACCESS.2020.3027380