# BIA 810 PROJECT REPORT: RECOMMENDATION SYSTEM

**Yubo Feng, Wei Yang, Zi Wang**
Stevens Institute of Technology
yfeng4@stevens.edu, wyang17@stevens.edu, zwang210@stevens.edu

**ABSTRACT**

With technology improvement, more and more things have been categorized into Big Data field. Not only because advanced technology equipment can process more and more byte, but also because more and more things around human beings in daily life have been informational, the things that are usually ignored and have been recorded by personal laptop and cellphone. After pioneers realized the value of big data gathered with general usage, they led us into Age of Big Data without even a little alarm that the advanced technology about Big Data will eventually change our life. How that is happen? While it is actually pretty easy. With the development of portable electronic devices including laptop and mobile phone, everyone will left track when he get out, shopping, jogging, searching and working. Every step you click on internet will be record by back-end server. Furthermore, lawmaker cannot capture this rapid process and can only sue the tech-giant company and legislate from former cases. Recommendation system is one of classic technology among cutting edge, which will use information from customer to make recommendation goods and advertisements.

## 1 Introduction

Our project is a new recommendation system model. Traditional recommendation systems are based on wide memorization model or embedding based model, which only can recommend the commodities customer purchased before. But new e-commerce companies need recommendation system to provide fresh new commodities from new stock keeping unit. So we need a new model to fit requirement.

## 2 Dataset and Features

### 2.1 Dataset description

We use public datasets from JD.com. This datasets contain seven different databases. Sku_data contains all click history from customers among one month. Delivery_data contains commodities delivery records among one month. Order_data restore log for a success purchase behavior. Compared to order databases, click_data contain every log records whether these records convert to success purchase behavior. Among seven databases we focus on sku_data, orde_data, click_data and user_data.

1. User_ID: each individual user have its own identical user_ID
2. request_time: the time point (accurate to seconds) that each click movement occur.
3. order_text: time point (accurate to seconds) that each purchase occur.

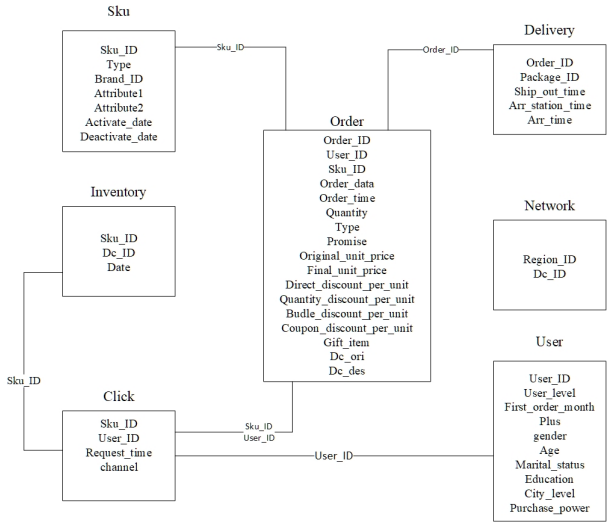


Figure 1: Datasets Realtionship

# 3 Exploratory Data Analysis (EDA)

## 3.1 Data imbalance

After we explore our click_data, we find data imbalance. most of sku only have 1 or 2 click history. And 80% click are occupied by 20% users. So if we directly use click data, we use lose most of outliers and some of maybe extremely important. The problem is occured when we view our user-click join process. If we join user data within sparse format, we will get sparse outcome. So we use a new method to view our data.

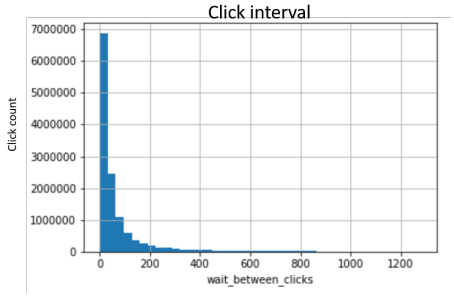


Figure 2: click imbalance

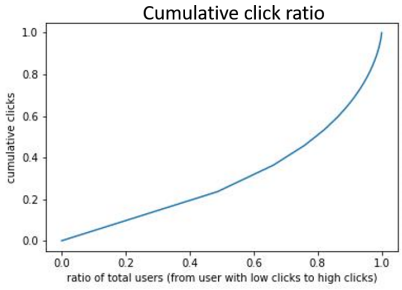


Figure 3: cumulative click ratio

# 4 Methods

## 4.1 Word2Vector

In order to process word into neural network, we should transform every single English word into numeric. We use word2vector to find relation between words. Furthermore, we can use similar method in a sequence model

## 4.2 Sequence2Vector

We use n-gram to extract sequence and use sequence2vector to generate final prepossessing data. N-gram have different choices. Depending on shopping customer behavior, we should use at least 3-gram. We analysis dataset joined

from click_data and sku_data, we find 90% log data have at least 5 click history. Finally, we decide use 5-gram to build our sequence model

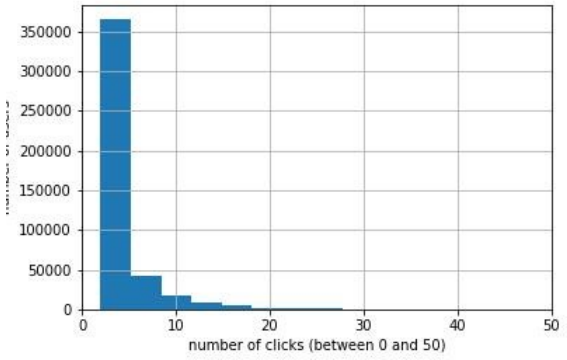


Figure 4: Number of Clicks

### 4.3 User Concatenate

Instead of directly using use_ID to sequence2vector, we should use concatenate user data as our input. Because we find sparse user distribution in EDA period. 50% user data and click data only have one row log record. If these data processed by neural network, they have a great chance to be ignore. So we use user features to concatenate user from 14000 rows to 7000 rows.

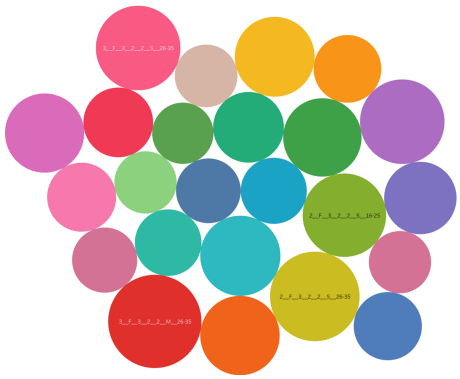

Figure 5: User Concatenate

1. User Level: 1-3
2. Gender: Female or Male
3. Education: 1-3 3 is highest
4. City Level: 1-5, 5 represent Beijing, Shanghai
5. Purchase Power
6. Martial Status
7. age

### 4.4 Pretraining

We use 5-gram and negative sampling to pretraining and build our sequence data. Suppose we have center word V3, we will use former v1 and v2, after v4 and v5 to build a basic. To accelerate train process, we use negative sampling to complete our sequence. The loss function is list as follow[1].

$$\begin{aligned}\operatorname{argmax}_\theta \sum_{(l,c)\in\mathcal{D}_p} \log \frac{1}{1+e^{-\mathbf{v}_c'\mathbf{v}_l}} + \sum_{(l,c)\in\mathcal{D}_n} \log \frac{1}{1+e^{\mathbf{v}_c'\mathbf{v}_l}} \\ + \log \frac{1}{1+e^{-\mathbf{v}_l'\mathbf{v}_l}} + \sum_{(l,m_n)\in\mathcal{D}_{m_n}} \log \frac{1}{1+e^{\mathbf{v}_{m_n}'\mathbf{v}_l}}\end{aligned} \tag{1}$$

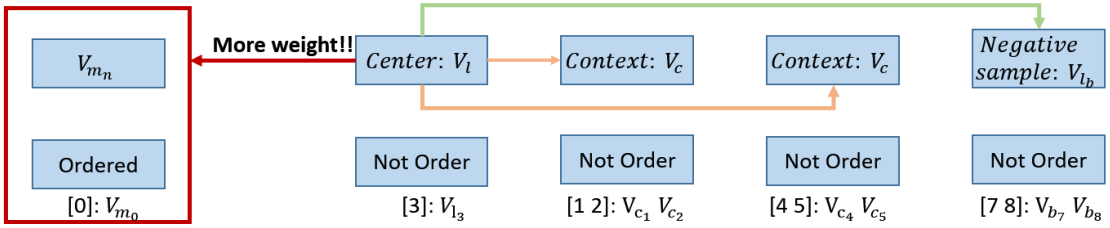


Figure 6: Build sequence with negative sampling

Here is final sequence result. Center is v4, context_1 is v1 etc.

| | order_item | center | context_1 | context_2 | context_3 | context_4 | neg_1 | neg_2 | neg_3 | neg_4 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 32194 | 27053 | 32194 | 32194 | 18710 | 32194 | 5480 | 29768 | 9324 | 31062 |
| 1 | 32194 | 18710 | 32194 | 27053 | 32194 | 32194 | 10150 | 12158 | 18840 | 14523 |
| 2 | 32194 | 29832 | 32194 | 32194 | 32194 | 32194 | 22519 | 9073 | 18854 | 20995 |
| 3 | 32194 | 22523 | 32194 | 32194 | 32194 | 32194 | 22120 | 1315 | 1858 | 3476 |
| 4 | 32194 | 21935 | 32194 | 32194 | 24836 | 21935 | 20832 | 28681 | 6794 | 12871 |
| 5 | 32194 | 24836 | 32194 | 21935 | 21935 | 32194 | 7351 | 25457 | 18994 | 30773 |

Figure 7: Sequence Result

### 4.5 Neural Network Model

We build a simple Neural network model to train our model. It consist with five part. First is input layer, which receive our prepossessing data from previous step. Second, we split user sequence into three part and differ it with sequence data to get three different big categories. When we analysis our sku vector dimension visualization, we believe sku can be split into three category after simple analysis, for instance, digital items, routine commodities and daily necessities. Third, we flatten the vector and concatenate them into one mix vector. Finally we let the mix vector get through a dense layer and use sigmoid output a probability to represent recommend or not.

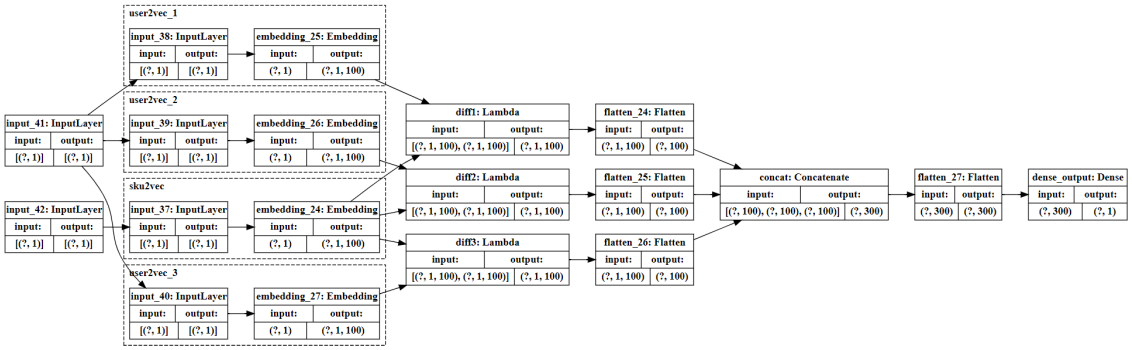


Figure 8: Neural Network Model

## 5 Other comparison RNNs and CNNs

## 6 Results and Conclusion

Our model loss general is 23%. The final part fluctuation is caused by and model accuracy is general 90%

Table 1: Result on our model

| | Precision | Recall | F1-score |
|---|---|---|---|
| Not Recommend (0) | 0.90 | 0.92 | 0.91 |
| Recommend (1) | 0.92 | 0.89 | 0.91 |
| | | | |
| Accuracy | | | 0.91 |
| Macro avg | 0.91 | 0.91 | 0.91 |
| Weighted avg | 0.91 | 0.19 | 0.91 |

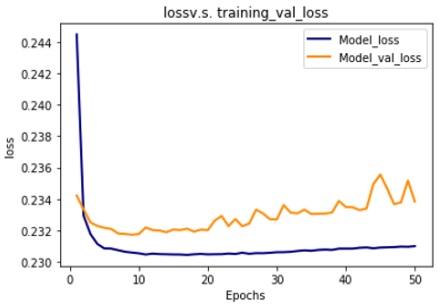


Figure 9: Model Loss

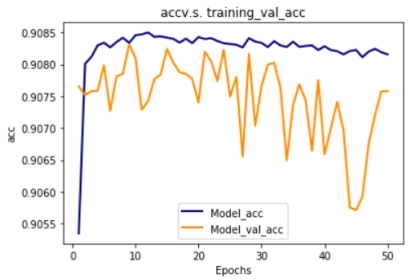


Figure 10: Model Accuracy

Conclusion: Our model obtained higher accuracy than RNN based model

## References

[1] M Grbovic, et al. "Real-time personalization using embeddings for search ranking at airbnb-kdd axiog." KDD '18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data MiningJuly 2018 Pages 311–320

[2] HT Cheng,et al. "Wide Deep Learning for Recommender Systems." Proceedings of the 1st Workshop on Deep Learning for Recommender SystemsSeptember 2016 Pages 7–10