**Customer Repurchase Prediction**

1. Task description
2. Data source
3. Data probe
4. Feature generation
5. Model construction
6. Result
7. Discussion

GitHub: <https://github.com/NetLand-NTU/Shopee>

**Task description**

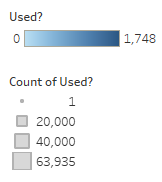
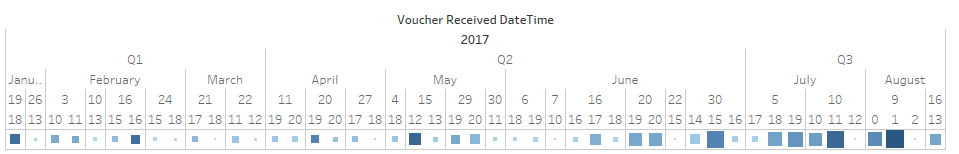
The main objective of the task is to predict which buyers will use the vouchers and continue to purchase after receiving coupons. The prediction is made based on a user’s purchase behavior generated from the shopping history.

**Data source**

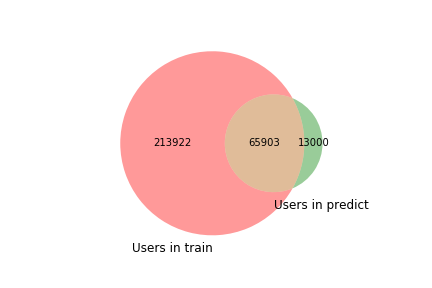
|  |  |  |
| --- | --- | --- |
| Source | Attributes | Time span |
| Training (710078,9) | userid, promotionid\_recieved (voucher\_code), voucher\_received\_time,used?,repurchased\_x (15d,30d,60d,90d) | 2017.1-8 (sparse) |
| Predict (78903,4) | userid, promotionid\_recieved (voucher\_code), voucher\_received\_time | 2017.8.16 |
| Transactions | Userid,shopid,total\_price,order\_time,promotionid\_used | 2015 Q2- 2017 Q3 |
| User profiles | Userid,registration\_time,is\_seller,gender,… |  |
| Likes | Userid,voucher\_received\_date,status,ctime |  |
| view\_log\_0-30 | Userid,promotionid\_received,,date,event\_name,count | 2017.4-2017.8 |
| Voucher\_distribution\_active\_date | Userid,promotionid\_received,time,date,active sessions counts | 2017.2(16)-2017.8(16) |
| Voucher\_mechanics | Promotionid,discount,max\_value |  |

**Data probe**

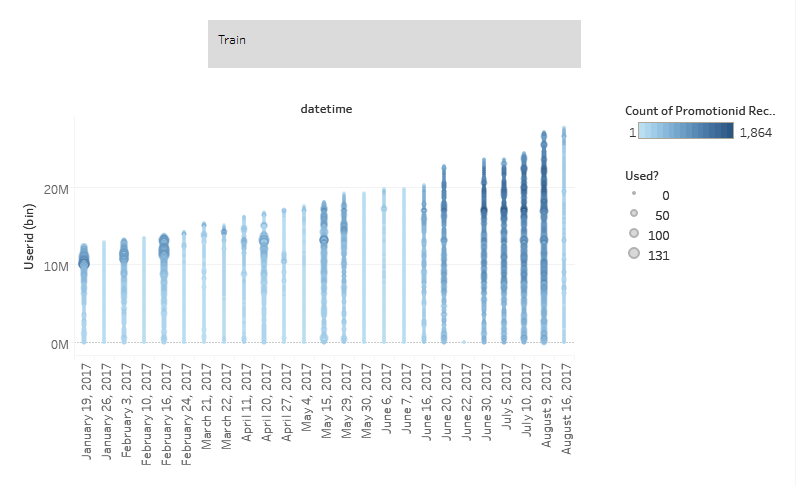
Training data & predict data

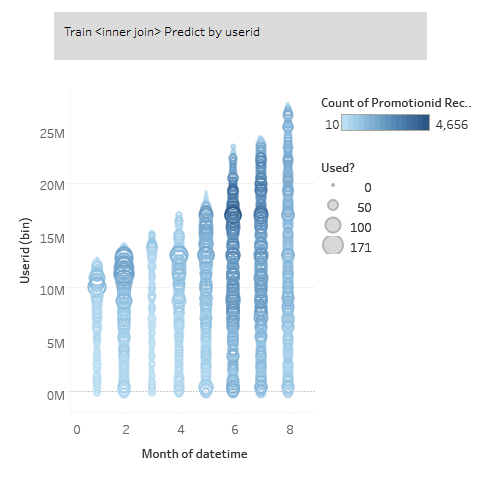
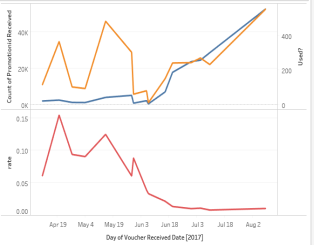


In the training data, the receiving date of vouchers is sparse from Jan to Aug 2017. Most of the coupons are sent to buyers in July and August, while the usage rate (red line) is lower than previous months.



The predict data are all gathered on 16 Aug 2017. There is an overlap of 65903 users with training data. Most of the shared users are new buyers appeared on 9th Aug.





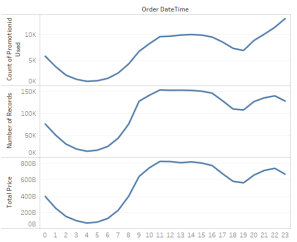
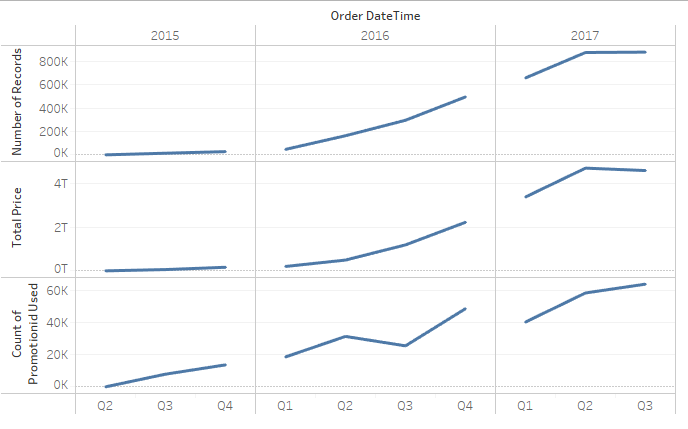
**Transaction history**

279803 (out of 279825 users in the training data) can be found in the transactions table.

78895 (out of 78903 users in the predict data) can be found in the transactions table.

We compared the transaction numbers, total spend and voucher usage in the scale of a year, quarter, month, day and hour. Except for the daily profile, the rest profiles are different in 2015, 2016 and 2017. Although the transaction data covers the shopping records in 2015 and 2016, we only focus on the items in 2017.

Some additional information, e.g., shop IDs, is provided in the transaction history. A potential link between promotion ID and shop ID may generate more features.

**Voucher mechanics**

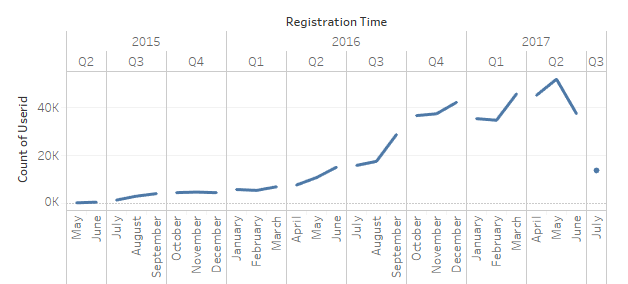
There are 94 unique promotion IDs, 92 in training data and 4 in predict data. Two mutual promotion IDs are only received on 16 Aug 2017. Thus promotion ID cannot be used as a feature. Based on the discount and max value of each promotion, we classified the IDs into 6 types.

|  |  |  |
| --- | --- | --- |
| Discount | Max\_value | Type |
| 20 | 1000000 | 1 |
| 50 | 2000000 | 2 |
| 50 | 1500000 | 3 |
| 20 | 1500000 | 4 |
| 50 | 1000000 | 5 |
| 30 | 1000000 | 6 |

?? not sure if it is better to remove records with voucher types of 3 and 4

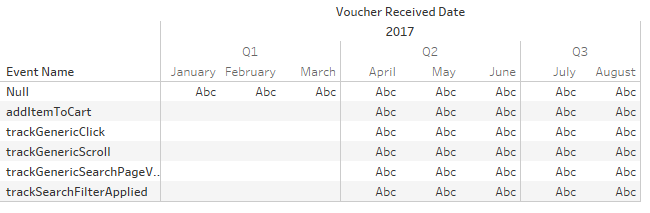
**User profiles**

Old customers are more experienced than younger ones. Thus they may prefer to use the vouchers and continue to purchase.



**View logs**

For the training data, we only use the records from April onward.



**Active sessions**



**Feature generation**

Three super groups (13 subgroups): user, voucher, date

All features are listed in naodong.docx

**Model construction**

1. Split data into training set, validation set, and test set

|  |  |  |
| --- | --- | --- |
| Set | Time span | N/P |
| Training set 1 | 4.11-8.9 | 34:1 |
| Training set 2 | 4.11-8.9 | 13:1 |
| Validation set | 8.16 (training data) | 33:1 |
| Predict set | 8.16 (predict data) | ? |

Training set 1 is used to construct the final model to predict the labels in predict set. Validation set is used to filter features and tune the parameter of Training set 2. The datasets are filtered using user IDs that shared by predict set and training set. 65350 (out of 78903 users) users remain.

1. Feature extraction
2. Normalization and add missing values

Z-score (MinMaxScale) is used to normalize continuous values.

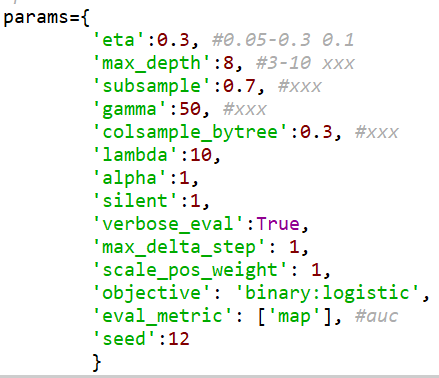
1. Feature reduction

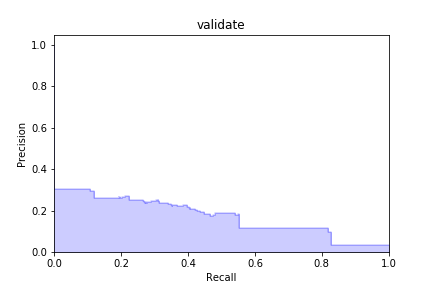
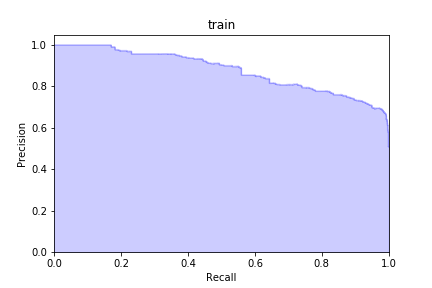
Select the top features by the importance

1. Training

**Result**

Xgboost is used to train the data. The prediction results are saved in “result.csv”.





**Discussion**

Likes table:

We didn’t use the likes information for the reason that even two shoppers have a similar preference for products they may not share the same purchase behavior. But if the shop and item relations are provided, we can generate connections between shops and items. The underlying information is if users prefer the products from one shop and have high transaction records, they may prefer to use the vouchers from the shop.

Balance data & data cleaning:

We have tried to balance the data using SMOTEENN, CNN, etc. But it took extremely long time. So we simply random sampled a proportion of negative samples from the original training data. Balancing data doesn’t help in this case.

After balancing the data, the training model becomes over fitted even we tuned the parameters (subsample, min\_child\_weight, lambda…) to prevent it. In summary, the prediction result is still low in the validation data.

Model selection:

Besides xgboost, RF, GBDT, LR and some blending techniques are all popular in customer preference predictions. We only trained a xgboost model that we believe a better algorithm will provide a limited improvement on the result. The selection of method is not important. The crucial part is how to split the data, extract effective and enough features to separate different groups, over-sampling/under-sampling the data and finally clean the noise in the training data.