Midterm Progress Quick Overview

Kehao Yao & Yuan Yao

In this project, our goal is to predict the probability that the target users finally decide to buy the advertising item, given the information about the advertising items, target users and item providers.

I Dataset

i) General idea

Our dataset is provided by Alibaba and contains over 500,000 advertising data. It mainly includes five different aspects, which are basic information, advertising items, target users, contexts and merchants. With this information, hopefully, we will be able to predict how likely is a customer to buy the advertising item after watching the advertisement provided by the platform.

ii) Data cleaning and describing

We first process the missing values by various means. For merchants' rating, we replace them with median value, while for users' age, we drop the missing values, since users from different age groups can act very differently in online shopping behaviors.

To better understand our data, we run a few preliminary analyses. Regarding the users' features, we visualize the data of users' age, gender and occupations. Since all the original data is encrypted by the data provider, we can only plot the data in group codes.

According to the graphs, female consumers make up a majority, over 75%, of all consumers. Users' occupation is generally divided into four categories, among which the category '2005' stands out and the category '2002' follows.

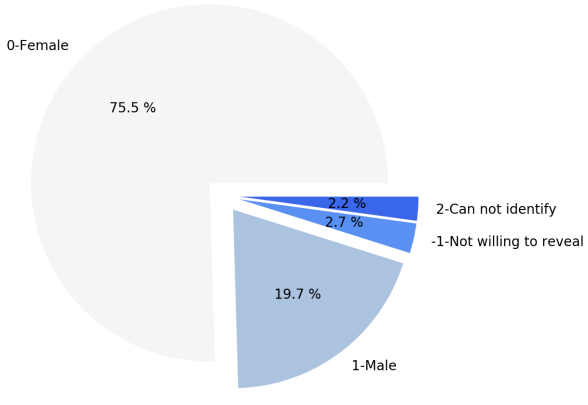

Fig 1. Users' gender

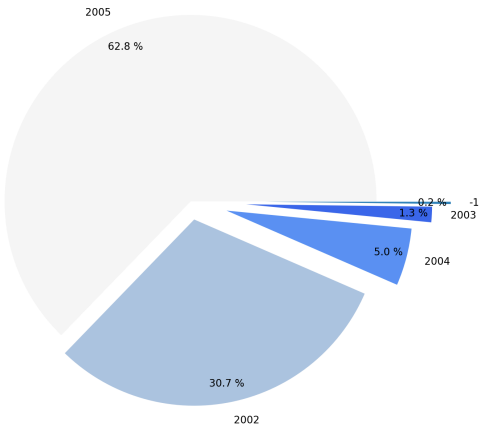

Fig 2. Users' occupation

Users' ages mainly fall into the categories from 1002 to 1005. Please see the bar chart as below.

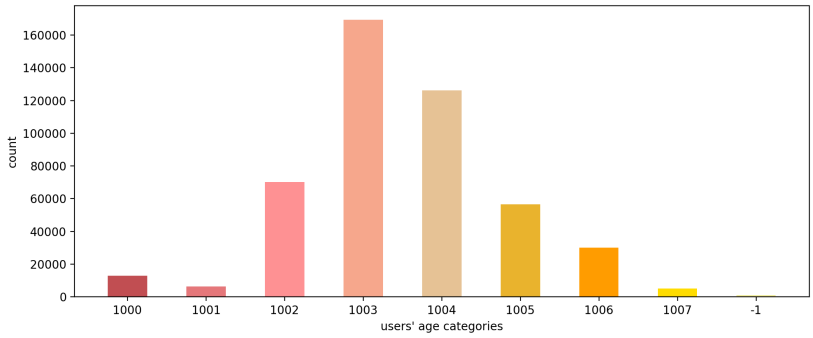

Fig 3. Users' age

II Feature engineering

We compare the properties of the advertising items and the wanted items of users, which are predicted by users' searching words. The item is more likely to be purchased as their intersection increases. Therefore, we generate a new feature "intersection". With this new feature, the impact of overlapped information on the purchase rate is better expressed.

Also, we conduct correlation analysis on numerous values like merchant score. It is obvious to see that, in the training set, the merchants' score on service is highly correlated with their score on delivery. Similar results are also observed in merchants' score on service and description of items. Therefore, we adopt a linear combination method with the help of linear regression, which integrates the highly correlative features into one feature, and receive a better result in the model performance.

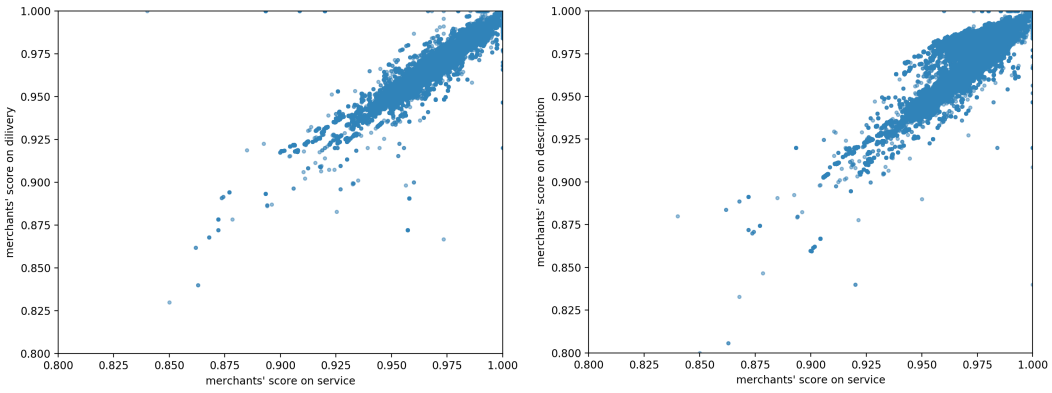

Fig 4. The correlation of merchants' score features

As to the features like item_id, we apply one-hot encoding to better fit them for the model. We replaced some ID features with their appearing frequencies using the count function.

```
#replace ID feature with appearing times
counts = pd.DataFrame(pd.value_counts(data['item_id']))
data['item_id'] = data['item_id'].replace(counts.index.tolist(),counts['item_id'].tolist())
counts = pd.DataFrame(pd.value_counts(data['item_brand_id']))
data['item_brand_id'] = data['item_brand_id'].replace(counts.index.tolist(),counts['item_brand_id'].tolist())
counts = pd.DataFrame(pd.value_counts(data['shop_id']))
data['shop_id'] = data['shop_id'].replace(counts.index.tolist(),counts['shop_id'].tolist())
counts = pd.DataFrame(pd.value_counts(data['item_city_id']))
data['item_city_id'] = data['item_city_id'].replace(counts.index.tolist(),counts['item_city_id'].tolist())
```

Fig 5. Replacing ID features with their appearing frequencies

Finally, we segment some continuous variables. The selection of the segmentation points is related to the analysis of the feature data.

```
data['shop_score_service0'] = data['shop_score_service'].apply(lambda x: 2 if x > 0.979 else x)
data['shop_score_service0'] = data['shop_score_service0'].apply(lambda x: 1 if 0.979 >= x > 0.967 else x)
data['shop_score_service0'] = data['shop_score_service0'].apply(lambda x: 0 if  x <= 0.967 else x)
```

Fig 6. Using lambda to segment continuous variables

III Model constructing

After reading relevant papers and researching the dataset, we decide to combine GBDT (gradient boosting decision tree) and LR (Logistic Regression) to build our model.

We plan to avoid over-fitting and under-fitting with the following methods:
1) feature engineering; 2) less complexed model; 3) solving the imbalanced data

Regarding to the imbalanced data, it is shown that purchase rate is extremely low, meaning that only a few data are marked with label "1". We apply a few ways to deal with that problem:

i) Use decision tree algorithm

Fortunately, decision trees perform well on imbalanced data. They make the predictions by learning a hierarchy of if/else questions and this can force both classes to be considered into our model.

ii) Resampling Techniques — Oversample minority class

We will use the resampling module from Scikit-Learn to randomly replicate samples from the minority class.

iii) Resampling techniques — Undersample majority class

Undersampling is to remove some observations of the majority class. Undersampling can be a good choice since we have a ton of data. However, a drawback is that we could be removing information that are valuable. This could lead to poor generalization and underfitting to the test set.

We will test the effectiveness of the models by calculating the logarithmic loss as below, where p_i is the estimated conversion rate of the i-th sample.

$$\log loss = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1-y_i)\log(1-p_i))$$

IV Midterm Conclusion and Future plans

Above all, with combined effort in feature engineering and model construction, we build a complete prediction model and gradually gain a deeper insight of the problem. Despite what we have completed, the following tasks remain to be done in the future.

1) trying other models like lightgbm and xgboost
2) dealing with unbalanced samples
3) processing time series data
4) introducing PCA for feature engineering