**The Prediction of Booking Destination of New User on Airbnb**

**Chien-Wei Lin, Po-Kang Huang**

**Abstract**

New users on Airbnb can book a 34,000+ cities across 190+ countries. By accurately prediction where a new user will book their first travel destination, Airbnb can share more information with their community, decrease the average time to first booking. This project aims to analyze Airbnb dataset and design a machine learning flow to do the prediction task. The task was assigned by Kaggle competition, which also provided training and test dataset for this project. We plan to learn the models and processes used by several winners in the competition, and try to combine all advantages of various machine learning methods.

**Methodology**

1. Dataset
2. Description of dataset

The dataset provided by Airbnb is a list of users’ information, including their demographics, web session records, and some statistics data. The whole dataset contains the following 5 files:

1. train-users and test-users

These two files respectively contain 213451 training samples and 62096 testing samples with 16 (country-destination, which is the label) and 15 (without label) properties, which are some basic information of some user data in their history records. The properties contain: id, date-account-created, date-first-booking, gender, age, signup-method, signup-flow, language, affiliate-channel, affiliate-provider, first-affiliate-tracked, signup- app, first-device-type, first-browser, country-destination, timestamp-first-active.

1. Observation and analysis of Data
2. Country Destination

Based on figure.1, we found that 64% of the users ended up booking nothing which indicated as NDF in *country\_destination*. Furthermore, there are 97.97% of those users who didn’t booked in the Airbnb didn’t have the record of *data\_first\_booking*. Therefore we turn to the users who had booked in Airbnb to ensure the effectiveness. The result is that all the users who didn’t book in Airbnb have NDF on the *country\_destination*. To predict the country destination, we want to classify the users who booked in the airbnb and those did not. As the previous observation, we found out a useful feature: *data\_first\_booking*

Fig. 1 percentage of the destination

1. Range of Age

From Fig.2, we can observe that most of the users who booked on Airbnb are in the range between 20 to 45 years old.

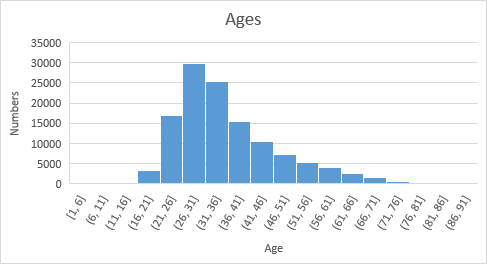


Fig.2 Age of users without missing data and outliers

1. Data pre-processing
2. One-Hot Encoding

For all the features in our raw data, they are string form which is harder for computation. Moreover, many of our feature can be represented by numerical values. For example, *gender* contains male, female, other, and unknown in raw data. It will not be proper to assign 0, 1, 2, 3 to represent male, female, other, and unknown, respectively. Instead of using numerical value, we translate this feature from 1-dimension vector into 4-dimension vector. Each dimension will be 1 if it matches the description of the feature, and will be 0 otherwise. If User1 is male, then then User1 has the 4-dimension vector as Table 1. For most of our feature, we translate each 1-dimension features into multiple dimension binary features

|  |  |  |  |
| --- | --- | --- | --- |
| Male | Female | Other | Unknown |
| 0 | 0 | 1 | 0 |

Table 1

1. Age

We first represented the age with its number directly, which is a trivial result. We then found out that it was the age distribution which matters. From Fig.2, we assume that children and seniors have less probability to use the Airbnb to plan a trip. After discarding the data with and , we create 15-dimension vector, which counted the index between adjacent pair in . From Fig.2, we assume children and seniors has less probability to use Airbnb on planning a trip. Therefore, we focus on the range of age on 20 to 40.

1. Royalty of Using Apple Product

We first treated these three features, *first\_device\_type, first\_browser,* and *signup\_app*, separately, which didn’t improve our performance. After analysis, we noticed that they have some relations when it comes to Apple product, such as iPhone, iPad, Macbook, or Safari. As a result, we shrink these three feature into one feature. If one of the features contains the iOS, Mac, or other Apple related product, then we set this feature to 1, or 0 otherwise.

1. Learning Model
2. Model selection

After analyzing our dataset, one can notice that the distribution of destination country in the whole dataset is extremely unbalanced. There are two destinations countries account for a large proportion of the dataset, NDF (no destination found) and US. NDF account for about 58%, and US account for about 29% of the whole data. Since these two labels account for a huge part of data set, we got a poor performance if we simply use a single-layer multi-class classifier to predict the destination country.

Moreover, we also found that all the users with NDF have missing data for their date-first-book feature, which means those users did not even book on the Airbnb, and, of course, they are impossible to book any country. Thus, we will have a strong confidence to predict NDF for users whose date-first-book is missing. By this reason, before starting our training process, we first divide the dataset into two groups, with date-first-book and without date-first-book, then we can discard NDF prediction and focus on the predictions of non-NDF.

For the remaining data of non-NDF, which has a large proportion of data whose destination countries are US. So we decide to develop a two-layer classifier to handle this condition. The first layer is a binary classifier, whose goal is to distinguish users between US and non-US. The second layer is a multiclass classifier, whose goal is to predict the most 5 possible destination countries for each user whose destination countries are non-US.

1. Binary Classifier

For our binary classifier, we combine SVM, Naïve Bayes and logistic regression models together. We used greedy method to train each model by cross-validation, choosing the one with the best performance, and we predict the result by majority voting. We first tried using linear regression model, but the linear regression is not appropriate for binary classification, because the linear regression model assumes that the outcome is continuous, binary outcomes obviously violate the assumption. The reason why not using polynomial regression is time-expensive.

1. Multiclass Classifier

The Multiclass classifier should predict top 5 possible booking destination countries for each user. We use SVM and logistic regression model together as our multiclass classifier. For the SVM model, we can directly calculate the probability of each destination countries for a user. For the logistic regression model, the one-vs-all classification is used to calculate the probability of each classes. That is, given 10 possible countries (without NDF and US) for each user, there will be 10 binary classifiers for each class. For each binary classifier, it will separate users with a particular destination country and all the other users. Then we could get 10 probabilities of each countries by normalizing the probabilities of all the classes for a user. Combining two lists of probabilities from two models for a user, we sorted the probabilities and pick 5 countries with highest probabilities as its prediction outcome.

1. Evaluation model
2. Binary Evaluation

For the binary classification, where we try to distinguish the users into US and non-US, we simply give the score of 1 to the user whose prediction country matching the ground truth country, otherwise, we give the score of 0 to the incorrect predictions.

1. Multiclass Evaluation

For the multiclass evaluation, we follow the evaluation metric for the Kaggle competition, NDCG (Normalized discounted cumulative gain) @k where k=5. The NDCG evaluation is shown as:

whereis the relevance of the result at position , in our case

The term is the maximum possible ideal for a given set of queries. All calculations are relative values on the interval . Using this evaluation metric, we predict a list of 5 destination countries with highest probability, and compare, in order, the countries on the list with the ground truth country to calculate its evaluation score. The ground truth country is marked with relevance = 1, while the rest have relevance = 0. For example, if for a particular user with the prediction list and the ground truth country , the evaluation score becomes:

The maximum score for each user is 1 and minimum 0. The score is determined by the order of the prediction countries in the prediction list. If the correct prediction is at first position, the score will be 1, and the score decreases if the position of the correct country goes larger. This method is often used to measure effectiveness of web search engine algorithms or related applications, which avoids the advantages of arbitrarily giving the score 1 or 0.

1. Cross validation and testing data set

After eliminating the instance without *data\_first\_booking,* we segmented 15% of the total data as the testing set and the remaining data as training set shown in Fig 2.

Fig. 3

Numbers of instance in training set and testing set

To evaluate and validate our model, we use the 10-fold cross-validation. In this way, the training data set can both be used to train and test our model. In addition, we also used 9-fold cross-validation to average the US/non-US prediction output in our logistic, SVM, and naïve Bayes binary classifiers. For each model, we have 9 different learners. The prediction will be based on the voting results from 9 different learners.

1. Bagging

For both the binary classifier and multiclass classifier, we use bagging to increase the accuracy with 9 bags. The size of each bag is picked by Gaussian distribution with numbers of instances and in each bag.

1. Description of whole model

**Result**

**Discussion**

1. Class Imbalance (70% US v.s. 30% non-US)
2. Disparity in Frequency of the Observed Classes

Since the performance of binary classifier will influence the outcome of multiclass classifier in our model, first step was to increase the accuracy of the binary classifier. At first, we had 70% validation accuracy on US/non-US binary classifier. In order to have better result, we used cross-validation and bagging. However, other than increased the accuracy, the performance became worse. Based on this observation, we noticed that our destination prediction on the validation set were all US. The 70% accuracy was the percentage of US in validation set.

1. Up-Sampling and Down-Sampling

In our classification problem, the class imbalance had a significant negative impact on model fitting. Our solution for this problem was to use up-sampling, which randomly sampled the instance from minor class with replacement to have the same size of the majority class, and down-sampling, which randomly select the subset of the majority class to have the same size of minority class.

1. Overfitting
2. Random Forest
3. Ada boost
4. Feature Selection

**Conclusion**

1. Performance
2. Future Directions
3. Learning from this project

**Reference**