# Recommendation Algorithms for User First Booking on Airbnb

Zhao Wang
Northeastern University
Seattle, WA
wang.zhao2@husky.neu.edu

Zerui Ma
Northeastern University
Seattle, WA
zeruima1989@gmail.com

Heng Xu
Northeastern University
Seattle, WA
xu.he@husky.neu.edu

## ABSTRACT

Airbnb has become more and more popular under the trend of sharing economy. In order to attract new users to place their first booking on Airbnb and offer a more personalized experience, we aim to build a recommendation system by using collaborative filtering approach, to predict which country these new users will make their first booking. Also the hosting data for various countries will be analyzed to provide better insight around booking area.

## Keywords

Learning Algorithms; Recommendation Systems; Collaborative Filtering; Cluster Models

## 1. INTRODUCTION

Data mining is the process of discovering interesting patterns from massive amounts of data. As a knowledge discovery process, it typically involves data cleaning, data integration, data selection, data transformation, pattern discovery, pattern evaluation, and knowledge presentation[1]. The science of learning plays a key role in the fields of data mining, statistics and artificial intelligence, intersecting with areas of engineering and other disciplines[2]. In a typical scenario, a quantitative or categorical outcome measurement is predicted based on a set of features.

Recommendation algorithms are a kind of learning algorithms which are widely used on e-commerce web sites. they use input about a customerś interests to generate a list of recommended items. Most recommendation algorithms are designed for finding similar customers, where they aggregate items from the similar customers, eliminates items the user has already purchased or rated, and recommends the remaining items to the user. The popular versions of these algorithms are collaborative filtering and cluster models. In the collaborative filtering, the similarity of customers is measured by the cosine of the angle between the two vectors which represent users' interests[3]. Using this algorithm to generate recommendations is computationally expensive, but it can be released by dimensionally reduction techniques[4]. To find the similar customers to the user, cluster models divide the customer base into many segments and treat the task as a classification problem[5]. Some algorithms classify users into multiple segments and describe the strength of each relationship[6]. Besides grouping the user to the similar customers, other algorithms such as search-based methods and item-to-item collaborative filtering focus on finding similar items[7]. Search- or content-based methods treat the recommendations problem as a search for related items[8].

User experience is now a critical factor to keep users and attract new users among web applications. That is the reason Airbnb wants to provide personalized and unique experience for its new users, thus Airbnb need an effective recommendation system to recommend a country for first-time booking.

However, the main challenge here is that Airbnb doesn't have the travel history or other type of the traveling data of new users, the only data available here is basic feature such as age, gender, session log etc., basically like a white paper to a recommendation system. While a typical recommendation system might make recommendation based on a few strongly related features, the system designed here need to focus on correctly classify similar users first, then trying to make recommendation with some relatively strong features. And that is why we choose collaborative filtering as our first-step approach.

## 2. DATASET DESCRIPTION

In this section, the data sets which are applied for the recommendation are introduced. The data is consisted of two parts: the first one is the list of usersf́irst booking destinations as well as their demographics and web session records; the other is the aggregated public host information dataset which is sourced from publicly available information from the Airbnb site. By analyzing the statistics of the source data, we can have an overview of the entire data and help to decide how to apply for the recommendation algorithms.

### 2.1 User data set

In the list of user first booking, each user is specified by a unique string id and each instance contains multiple types of properties for that user, which should be date, numerical, categorical, etc. There exists missing values in each fields, which should be handled before building recommendation model. The users whose destination countries are unavailable (indicated as NDF) are also need to be filtered out. The

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

details of user properties are described as Table.1.

**Table 1: User Data Statistics**

| field | type | description |
|---|---|---|
| id | string | unique for each user |
| date_account _created | date | |
| timestamp_first _active | timestampe | |
| date_first _booking | date | |
| gender | categorical | FEMALE, MALE, OTHER, unknown |
| age | numerical | 1 to 150 |
| signup_method | categorical | basic, facebook, google |
| signup_flow | numerical | 0 to 25 |
| language | categorical | en, zh, fr, es, ko, de, etc. |
| affiliate_channel | categorical | api, content, direct, etc |
| affiliate_provider | categorical | bing, facebook, google, etc |
| first_affiliate _tracked | categorical | linked, local ops, product, etc. |
| signup_app | categorical | Android, iOS, Moweb, Web |
| first_device_type | categorical | Android Phone, iPad, iPhone, Mac Desktop, etc. |
| first_browser | categorical | Chrome, Safari, Firefox, etc. |

Since most records in the dataset have no first booking destination, we only focus on the instances with available destination. After filtering the records without specific destinations, the numbers of the users for each country are as Table.2 described. The most number of remaining users is for other destinations, which is less useful for the recommendation. The instances with specific destinations are the data we can use to build the predication model for new users.

**Table 2: Users First Booking Destination**

| Destination Country | Population |
|---|---|
| AU | 537 |
| CA | 1425 |
| DE | 1059 |
| ES | 2243 |
| FR | 5013 |
| GB | 2318 |
| IT | 2827 |
| NL | 757 |
| PT | 217 |
| US | 62263 |
| other | 10075 |

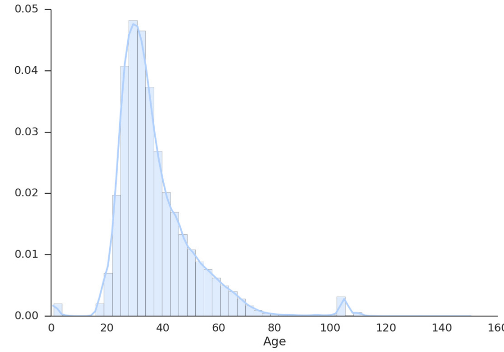


**Figure 1: Age Distribution for Users on Airbnb**

## 2.2 Hosts data set

The other data set is about detailed information of host listings among 16 countries available on Airbnb, although it covers almost all the countries in new user data set, but PT(Portugal) listing data is missing here.

This listing data set contains most of the public information available on Airbnb, and lots of features such as listing price, listing rating score, host registration information and neighbourhood etc., may help recommendation system identify better listings among target country for new users. However, due to the limitation of new user data, the recommendation on this part is most likely to be based on features weighted by common sense: highest rated, best available price etc..

Since listing price is showed in dollar, so the average listing price distribution is influenced by currency exchange rate in some extent, but besides that fact, the average listing price definitely reflects the popularity of these destination countries. BM(Bermuda) is a typical high-end vacation hot spot, but it's surprised to see UY(Uruguay) has such a high average price, maybe because the retrived UY listings is very limited. Other than these two countries, AU and US are the most expensive countries to go on Airbnb. It's interesting to see AU has higher average listing price over US, a possible explaination for this could be the number of listings in US is much greater than AU's, thus competetion over the host has lowered the average listing price in US.

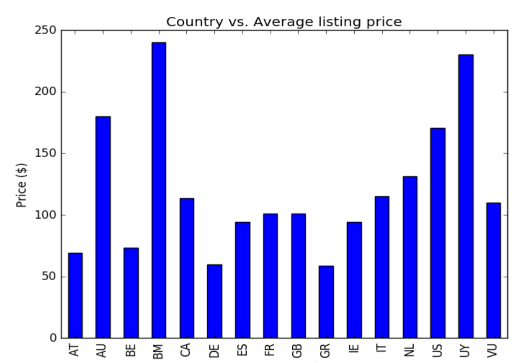


**Figure 2: Average price distribution among countries**

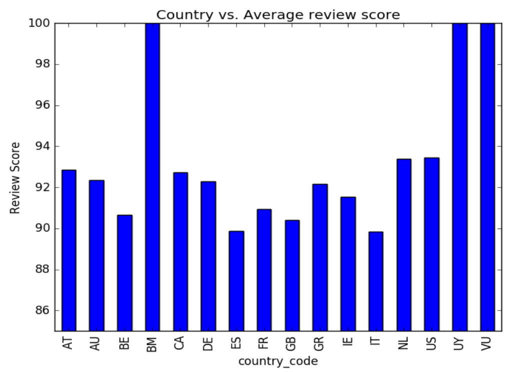


**Figure 3: Average rating score distribution among countries**

When comes to average review scores, most of countries get over score of 90. BM, UY and VU receives average score of 100, but since the number of listing of these 3 countries in this data set is very limited, it probably has large bias here. Besides these 3 countries, US and NL is probably best reviewed place to go on Airbnb, IT and ES is the worst reviewed, but still has average score of over 89.

# 3. PROPOSED ALGORITHMIC APPROACH

Using collaborative filtering to aggregate the similar users based on their first booking on Airbnb. The similarity between users is measured by cosine of the angle between the two vectors which indicate user properties. Dimensionality reduction should be applied to reduce computational complexity and speed up the algorithm.

## 3.1 Data preprocess

Since each

**Table 3: Data Preprocess**

| Processed Attribute | Description |
|---|---|
| Lag between date_first_booking and date_account_created | Divided into 4 categories =0, <0, >0, NA. |
| Lag between date_first_booking and timestamp_first_active | Divided into 3 categories =0, >0, NA |
| Age | Replace the missing values with the conditional mean; scale the age with mean and standard variance |
| Others | Original value |

## 3.2 Basic and Ensemble Classifiers

## 3.3 Collaborative Filtering

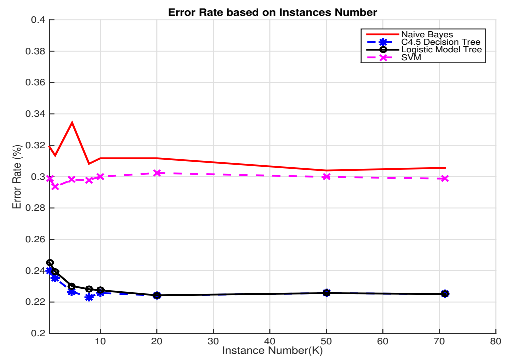


**Figure 4: Error Rates for Basic Classifier**

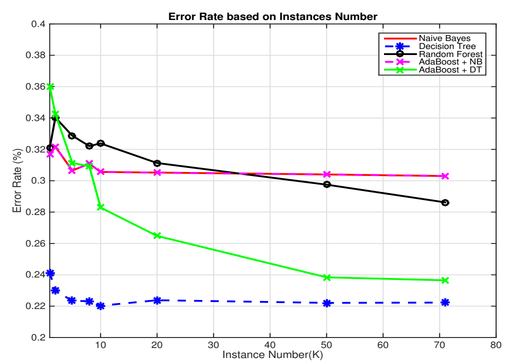


**Figure 5: Error Rates for Ensemble Classifier**

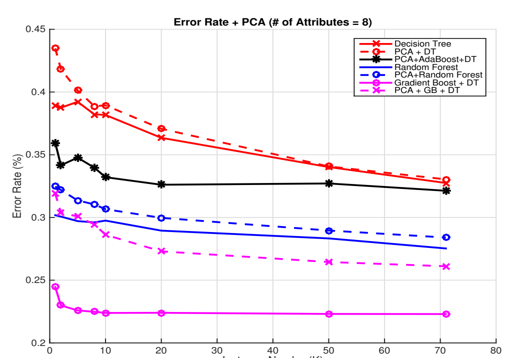


**Figure 6: PCA with Classifiers**

**Table 4: Clustering Parameters**

| Percentage Change of WGSS | Cluster Num | WGSS | Error Rate |
|---|---|---|---|
| < 0.3% | 10 | 114013 | 0.3036 |
| < 0.1% | 17 | 106895 | 0.2978 |
| < 0.03% | 67 | 84909 | 0.2700 |
| < 0.003% | 163 | 73112 | 0.2330 |
| < 0.001% | 424 | 55412 | 0.2137 |

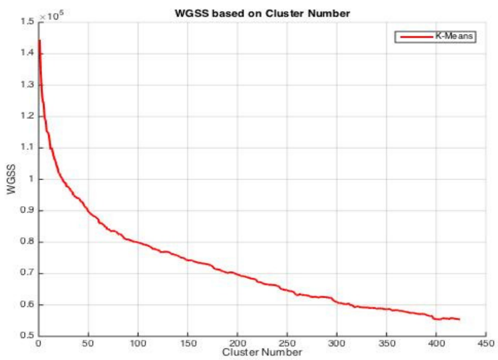


**Figure 7: WGSS based on Clusters**

### 3.4 Three Layer Ensemble

In previous section, results from basic ensemble methods didn't improved significantly from individual classifier results, so we tried to take advantage of different ensemble methods to build this three layer Ensemble methods[9].

First of all, in order to perform multilayer Ensemble, we need to partition dataset into 3 parts: training set 64%, validation set 16%, testing set 20%. And We will use X to denote features or predictions, y to denote labels.

For the first layer, we have used 6 individual classifiers: Logistic Regression, Random Forrest, Gradient Boosting, Deceision Tree, Extra Tree, K Nearest Neighbour. And all classifiers have been applied twice. First, Classifiers are trained on X_train, y_train and used to predict the class probabilities of X_valid. Next, Classifiers are trained on X = X_train + X_valid, y = y_train + y_valid and used to predict the class probabilities of X_test.

For the second layer, the predictions on X_valid from first layer are concatenated and used to create a new training set XV, y_valid. The predictions on X_test are concatenated to create a new test set XT, y_test. Then we train two following ensemble methods on XV, y_valid, and make predictions on XT.

First ensemble method is denoted as ENA, ENA computes optimized weights for each Prediction X made from first layer classifiers such that minimizes log_loss of test result.

Second ensemble method is denoted as ENB, ENB is similar to ENA, but the weights will be assigned with respect to the number of classes we have. For example, X1,X2,...,Xn is the set of predictions given, for each Xi = Xi1,Xi2,...,Xim, where m is the number of classes, and assign weights to each predictions with respect to different classes.

And in the end of this layer, we apply isotonic calibrated classifier obtained from scikit-learn package in python to genearate two new classifiers. So in second layer, we have obtained two ensemble classifiers and two isotonic calibrated classifier.

Then comes to third layer, we assign arbitrarily weights to each classifiers we got from last layer, and combine them into a new classifier. After trying few sets of weights, we found weights [2/13, 4/13, 2/13, 5/13] for [ENA, calibrated ENA, ENB, calibrated ENB] has lowered log loss. We should trying to test more weight distributions programmatically in order to find the best one, or trying to incorporate some statistical techniques here for covering wider range of weight distributions, but due to the time limits, this weight distribution is by far the best value for the result.

The final results for each layers are shown in the tables. Noticed that for first layer results, Gradient Boosting classifier has produced much better results compare with other classifiers we used in this experiment, it's bad for ensemble method since weight will be assigned heavily to Gradient Boosting classifier, other classifiers might get nealy zero weights, thus the ensemble methods might not improve a lot from the result of Gradient Boosting classifier.

The results from second layer proves this, for two ensemble methods ENA and ENB, while the log loss decreased for a very small amount, the error rate still remains the same, the calibrated ensemble methods improves a little on the error rate, but have increased on log loss. When comes to third layer results, noticed that althogh the log loss has decreased from 0.875 to 0.872, error rate however has increased from 0.2230 to 0.2231, so it's not very clear if this classifier has really improved the result from single Gradient Boosting classifier.

In the future version of this work, we might want to try more different classifiers to see if there's any other independent classifiers that can match with the result from Gradient Boosting classifier, because only then the ensemble methods can take advantage of few different classifiers and provide more preceise results.

**Table 5: Layer one results**

| Classifier | log_loss | error_rate |
|---|---|---|
| KNN | 2.3201224 | 0.2854003 |
| Gradient Boosting | 0.8753181 | 0.22308 |
| Extra tree | 6.0773485 | 0.3299713 |
| Random Forest | 3.7757482 | 0.2752578 |
| Logistic Regression | 1.1450557 | 0.29605 |
| DT | 8.9571967 | 0.3264777 |

**Table 6: Layer two results**

| Methods | log_loss | error_rate |
|---|---|---|
| EN_A | 0.875309 | 0.22308 |
| Calibrated_EN_A | 0.8933696 | 0.2230236 |
| EN_B | 0.875449 | 0.22308 |
| Calibrated_EN_B | 0.8932008 | 0.2230236 |

**Table 7: Layer three results**

| log_loss | error_rate |
|---|---|
| 0.8726795 | 0.2231927 |

# 4. CONCLUSION

Three layer ensemble methods should improve its performance by trying different classifiers such that their results can match with Gradient Boosting results, also weight distributions in third layer should be explored thoroughly to find the best result.

# 5. REFERENCES


[1] Han, J., Kamber, M., and Pei, J. , ”Data Mining: Concepts and Techniques”, 3rd Edition, 2011.

[2] Hastie T., Tibshirani R., and Friedman, J. , ”The Elements of Statistical Learning: Data Mining, Inference, and Prediction”, 2nd Edition, 2009.

[3] B.M. Sarwarm et al., ”Analysis of Recommendation Algorithms for E-Commerce”, ACM Conf. Electronic Commerce, ACM Press, 2000, pp.158-167.

[4] K. Goldberg et al., ”Eigentaste: A Constant Time Collaborative Filtering Algorithm”, Information Retrieval J., vol. 4, no. 2, July 2001, pp. 133-151.

[5] P.S. Bradley, U.M. Fayyad, and C. Reina, ”Scaling Clustering Algorithms to Large Databases”, Knowledge Discovery and Data Mining, Kluwer Academic, 1998, pp. 9-15.

[6] L. Ungar and D. Foster, ”Clustering Methods for Collaborative Filtering”, Proc. Workshop on Recommendation Systems, AAAI Press, 1998.

[7] G. Linden, B. Smith, and J. York, ”Amazon.com recommendations: item-to-item collaborative filtering”, Internet Computing 7:1, 2003, pp. 76-80.

[8] Leskovec, J., Rajaraman, A., and Ullman, ”Mining of Massive Datasets”, 2nd Edition, 2004, pp. 307-341.

[9] Pons, Sandro Vega., ”three_level_classification_architecture”, Kaggle, 25 Feb 2016. Web. 28 Apr 2016.