### 



**Airbnb Recruiting - New User Bookings**

## ***Where will a new guest book their first travel experience?***

**Capstone Project Document**

**By Sumita Soundararajan**

***Foundations of Data Science Workshop by Springboard***

**Mentored by Anirban Ghosh, NOKIA**

## **Problem**

## Trying to predict in which country a new user on Airbnb, will make his or her first booking. There are 11 potential countries along with a 12th class - NDF (No Destination Found), indicating the user did not make any booking.

## New users on Airbnb can book a place to stay in 34,000+ cities across 190+ countries. By accurately predicting where a new user will book their first travel experience, Airbnb can share more personalized content with their community, decrease the average time to first booking, and better forecast demand.

## URL -> <https://www.kaggle.com/c/airbnb-recruiting-new-user-bookings>

## **Data**

A list of users along with their demographics, web session records, and some summary statistics are provided by Airbnb. The challenge is to predict which country a new user's first booking destination will be. All the users in this dataset are from the USA.

## URL -> <https://www.kaggle.com/c/airbnb-recruiting-new-user-bookings/data>

There are 12 possible outcomes of the destination country: 'US', 'FR', 'CA', 'GB', 'ES', 'IT', 'PT', 'NL','DE', 'AU', 'NDF' (no destination found), and 'other'. Destination 'NDF' is different from 'other' because 'other' means there was a booking, but is to a country not included in the list, while 'NDF' means there was no booking.

The data consists of user characteristics like language, age, browser, date-of-account-creation, OS, etc. for the train and test users. There is data on the actions taken by users on the website along with the details of the action and duration.

## **Approach**

The following steps highlight the strategy to be adopted for carrying out the analysis for the Capstone project:

1. Data Wrangling and Cleaning
   * Deal with missing values: Explore the data to determine if missing values should be discarded or hold significance
   * Dropping columns that are irrelevant to the analysis
   * Rearrange and transform dataset for cleaner analysis.
2. Exploratory Data Analysis
   * Perform Regression Analysis to determine factors that most influence the outcome of booking a particular destination
   * Identify patterns and correlation between the different variables
   * Use data visualization for graphical analysis to identify trends and answer questions on the dataset
3. Feature Engineering
   * The success of all Machine Learning algorithms depends on how the data is presented. The features in the data directly influences the predictive models used and the results achieved.
   * Identify great features that describe the structures inherent in the data and significantly affect the result
   * Transform data from a raw state to a state suitable for modelling
4. Predictive Analysis
   * Our machine learning approach will be supervised classification machine learning because the datasets are designed for this method since we have access to a training set with the correct outcomes and a test set without outcomes.
   * The algorithm that we are going to use have to be fast and not use up a lot of memory since we have a very large amount of data to process.
   * We also want multi classification which means that we want to basically find out how probable it is that the user wants to travel to all destinations and not just which destination has the highest probability.

**File descriptions**

The dataset we are researching is provided by Airbnb. The 5 datasets provided, contain a list of users along with their demographics, web session records, and some summary statistics.

1. train\_users.csv - the training set of users
2. test\_users.csv - the test set of users

## The training data consists of 213,451 rows with 16 columns while the test data has 62,096 rows with 15 columns. The values of 'country\_destination' are missing in the Test data and that is the value that is to be predicted.

* + id: user id
  + date\_account\_created: the date of account creation
  + timestamp\_first\_active: timestamp of the first activity, note that it can be earlier than date\_account\_created or date\_first\_booking because a user can search before signing up
  + date\_first\_booking: date of first booking
  + gender
  + age
  + signup\_method
  + signup\_flow: the page a user came to signup up from
  + language: international language preference
  + affiliate\_channel: what kind of paid marketing
  + affiliate\_provider: where the marketing is e.g. google, craigslist, other
  + first\_affiliate\_tracked: what is the first marketing the user interacted with before signing up
  + signup\_app
  + first\_device\_type
  + first\_browser
  + country\_destination: this is the target variable to be predicted

1. sessions.csv - 10,567,737 rows of 6 variables with multiple rows per user-id. Has details on the web sessions log depicting the browsing behavior of users
   * user\_id: to be joined with the column 'id' in users table
   * action
   * action\_type
   * action\_detail
   * device\_type
   * secs\_elapsed
2. countries.csv - Has summary statistics of destination countries in this dataset and their location information.
3. age\_gender\_bkts.csv - Has summary statistics of users' age group, gender, country of destination
4. sample\_submission.csv - Contains the correct format for submitting our predictions

The training and test sets are split by dates. In the test set, you will predict all the new users with first activities after 7/1/2014. In the sessions dataset, the data only dates back to 1/1/2014, while the user’s dataset dates back to 2010.

**Data Exploration**

## **Training User data**

## Country\_destination - has 12 valid classes with NDF meaning no booking. About 123,489 users ie 58% of users have not made a booking.

## 87,590 users have made a booking, out of that 61,457 users have chosen US – That is among those that book, 70% chose to do it in US. The rather imbalanced classes will be addressed during training the model.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PT** | **AU** | **DE** | **NL** | **CA** | **ES** | **GB** | **IT** | **FR** | **Other** | **US** | **NDF** |
| 83 | 152 | 250 | 247 | 440 | 707 | 731 | 979 | 1435 | 3655 | 20095 | 45041 |

## Age - there are many values in thousands and some in single digits. 41% of users have missing age. 23% of users with missing age have made a booking.

## Further analysis shows that the bulk of booking is done by users in age-group 30-60.

## The bookings drop significantly once the user’s age exceeds 60.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Age group** | **GB count** | **GB %** | **US count** | **US %** | **FR count** | **FR %** | **ES count** | **ES %** | **IT count** | **IT %** | **Other count** | **Other %** |
| 15-30 | 563 | 32 | 17,514 | 37 | 1,185 | 32 | 685 | 41 | 697 | 35 | 2,598 | 35 |
| 30-60 | 1,049 | 60 | 28,124 | 59 | 2,256 | 62 | 915 | 55 | 1,168 | 59 | 4,459 | 60 |
| >60 | 126 | 7 | 2,028 | 4 | 210 | 6 | 76 | 5 | 123 | 6 | 355 | 5 |
| **Total** | **1,738** |  | **47,666** |  | **3,651** |  | **1,676** |  | **1,988** |  | **7,412** |  |

## Language -The language spoken is distributed as shown below. It is not surprising that most users speak English since Airbnb is a company located in US and its customers are mostly Americans.

|  |  |  |
| --- | --- | --- |
| **Language** | **Count** | **Percent** |
| en | 206314 | 96.66% |
| zh | 1632 | 0.76% |
| fr | 1172 | 0.55% |
| es | 915 | 0.43% |
| ko | 747 | 0.35% |
| de | 732 | 0.34% |
| it | 514 | 0.24% |
| ru | 389 | 0.18% |

1. Gender –Users with 'unknown' gender book less frequently than those with a known one – either MALE or FEMALE. We can see that there are a lot of missing values for gender. Almost half of the users did not input their gender information.

|  |  |  |
| --- | --- | --- |
| **Gender** | **Count** | **Percent** |
| Unknown | 95,688 | 44.9% |
| Female | 63,041 | 29.6% |
| Male | 54,440 | 25.5% |
| Other | 12 | 0.0% |

## Below plot shows the booking done by each gender – Male or Female. There is no

## significant difference in the booking based on gender.

**Categorical variables:**

1. Users with the 'google' signup\_method book less frequently than 'basic' or 'facebook'
2. Users with signup\_flow 3 book more frequently than any other category while several have nearly 100% 'NDF'
3. Language - there are a large number of languages represented even though major bookings were done by users in US
4. Users with affiliate\_channel 'content' book less frequently than other categories
5. Users with affiliate\_provider 'craigslist', direct', and 'google' book more frequently than other categories (this begs the question as to why the google affiliate channel is more effective than the google sign up method)
6. Users with first\_affiliate\_tracked 'local ops' book less frequently than other categories
7. Users with signup\_app 'Web' booked the most frequently, while those with 'Android' booked the least
8. Users with first\_device\_type 'Mac\_Desktop' booked the most frequently, while those with 'Android Phone' booked the least

## **Sessions data**

1. 3563 rows with no valid user-ids were identified and dropped from processing.
2. There were many missing values, that were considered under the label - “Not Given”.
3. File consists of about

* 360 unique actions
* 11 unique action types
* 156 unique action details
* 14 unique device types

**Data Wrangling**

**Train.users –**

In Age column, there are many values in thousands and some in single digits. Assumed an age range of 15 to 100, and then assumed the 4-digit years from 1924-1995 as birth years.

## Replace all missing values to “NA”

## Replace all ages outside the valid range to “NA”

## Calculate the 4-digit years as birth year by subtracting given year from 2016 (commencement of competition in Kaggle)

## **Sessions –**

## Replace all NA or blanks in secs\_elapsed column to 0

## For columns, User\_id, Action, Action\_type, Action\_detail and Device type,

## Replace missing or blanks with “Not given”

## Replace “-unknown- “with “unknown”

**Feature Engineering**

The idea of “*Transforming Data*” from a raw state to a state suitable for modeling is considered feature engineering.

**Data Aggregation**

Considering the user’s browsing history, available in the Sessions dataset was voluminous, it was decided to use aggregated value of the categorical variables. Each categorical variable like Action, Action\_type, Action\_detail, Device\_type was grouped and their frequency was calculated.

Based on the frequency count, the percentage and cumulative percentage of each variable were identified. A minimum cut-off of 95% was decided based on the data. All information with cumulative percentage higher than the cut-off was deemed inconsequential and grouped under a new label called MISC (short for Miscellaneous).

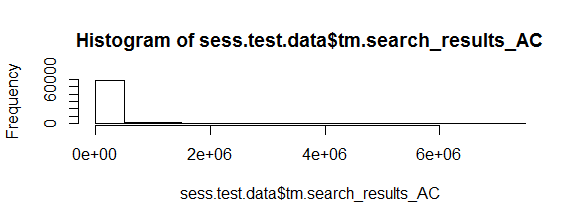
* Action - Out of 360 unique actions, 308 were renamed as MISC
* Action Type - Out of 11 action types, 5 were renamed as MISC
* Action Detail - Out of 156 action types, 119 were renamed as MISC
* Device type **-** Out of 14 device types, 8 were renamed as MISC

For each user and Action combination, the elapsed time and frequency were summarized. Similar processing was done on other variables – Action Type, Detail and Device type. The reshaped data formed the **Base feature dataset.**

The Base feature dataset was further merged with the country\_destination from the Training user dataset (~210 variables).

**Winsorization**

Further inspection of the data showed that more than 50% of the values in 174 variables corresponded to a constant (~0). A sample histogram of one such variable - tm.search\_results\_AC shows a skewed distribution.



The quantile function on the same variable shows the data values with different probabilities



Winsorization is the transformation of [statistics](https://en.wikipedia.org/wiki/Statistic) by limiting [extreme values](https://en.wikipedia.org/wiki/Extreme_value) in the [statistical](https://en.wikipedia.org/wiki/Statistics) data to reduce the effect of possibly spurious [outliers](https://en.wikipedia.org/wiki/Outliers).

To remove the outliers, the data was winsorized i.e., extreme data values were replaced with less extreme values. The values above 98% probability were replaced with 98th percentile value, likewise values below 2% probability were replaced with the 2nd percentile value.

The quantile function on the same variable shows the new values



The number of variables on the Base feature dataset is still 212 variables, quite large for running a model.

Selecting the right features in your data can mean the difference between mediocre performance with long training times and great performance with short training times.

The ***caret*** R package provides tools that automatically report on the relevance and importance of attributes in our data and even select the most important features.

**Near-zero Variance**

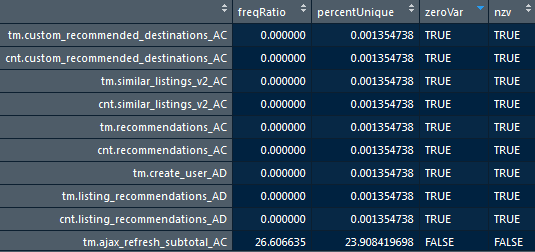
One of the ways to reduce the data is to eliminate variables/predictors with zero or near-zero variance using the function **nearZeroVar** from the *caret* R package, based on Kuhn's suggestion described below.

Near-zero variance means that the fraction of unique values over the sample size is low (say 10% and the ratio of the frequency of the most prevalent value to the frequency of the second most prevalent value is large (say around 20). If both of these criteria are true and the model in question is susceptible to this type of predictor, it may be advantageous to remove the variable from the model.

*-- Kuhn, M., & Johnson, K. (2013).*Applied predictive modeling,*New York, NY: Springer.*

From the below table, we could see that on calling the nearZeroVar function with the argument *saveMetrics = TRUE*, we have access to the frequency ratio and the percentage of unique values for each predictor, as well as flags that indicates if the variables are considered zero variance or near-zero variance predictors. By default, a predictor is classified as near-zero variance if the percentage of unique values in the samples is less than 10% and when the frequency ratio mentioned above is greater than 19.

About 9 predictors are eliminated based on this method. (base feature data ~203 variables)

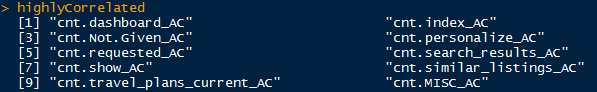


**Highly Correlated Variables**

If two numerical features are perfectly correlated, then one doesn't add any additional information (it is determined by the other). So, if the number of features is too high (relative to the training sample size), then it is beneficial to reduce the number of features through a [feature extraction](http://en.wikipedia.org/wiki/Feature_extraction) technique. Many methods perform better if highly correlated attributes are removed.

The **findCorrelation** function from the caret R package was used to analyze a correlation matrix of our data’s attributes report to identify attributes/variables that could be removed.

This function searched through a correlation matrix and returned a vector of integers corresponding to columns to remove variables with absolute correlation of 0.6. About 139 variables were found to be highly correlated and were removed from the base feature data.





The final base feature dataset has 64 variables that will be used to build models.

**Model Selection**

There is a total of 12 classes to be predicted. However, the data is not balanced because NDF and US counts for a large proportion of the data. The ratio of destination NDF to PT is 542 which shows that data is highly imbalanced. It was decided to use two methods – first the Logistic regression algorithm to build a model since it works best for binary classification problems.

The second was to address this multi-class classification problem with supervised learning of [Decision trees](http://scikit-learn.org/stable/modules/tree.html#tree). Decisions trees predict a target variable by learning simple decision rules detected in feature data, and are appropriate here for some of the following reasons:

* can handle numerical and categorical data
* can handle multiple classes
* can handle class imbalance
* can handle missing data

It was decided to build a total of 66 level binary classifiers to separate the data based on destinations – NDF and US, NDF and FR, US and DE and so on.

Before running the dataset through the machine learning module several preparations were made. Both the features we want to predict and the features we want to base our prediction had to be extracted. The selected features to base our prediction is already available in the Base feature dataset (~ 64 variables) extracted from the sessions dataset.

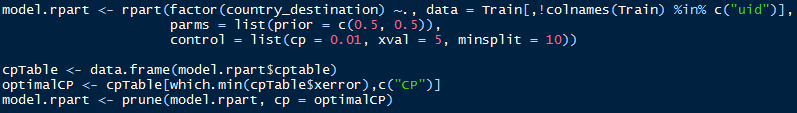
The Training set was created by extracting a list of user-ids and country\_destination from the original Training user dataset and performing an inner join of this set and the Base Feature dataset. The Test set on the other hand is created by extracting a list of user ids from the original test user set and performing an inner join of this set and the Base feature dataset total set which will naturally drop all rows with user ids not in the test-user id group.

This is not the only thing we did with the datasets. The first step was to drop the id column from the model building process and convert the country\_destination to a factor variable. Since the model works better with numeric values instead of strings, the country\_destination was encoded to be represented by a number instead of a string. The biggest class/destination was given a 0 and smaller one was given a 1.

First the model was built using Logistic regression method and the AUC values were computed.

Next, the model was built using Decision Tree algorithm. To validate the model, we used the CP value (Complexity parameter of the tree). The tree had to be pruned to avoid overfitting of the data. The convention is to have a small tree and the one with least cross validated error (*xerror*).

The optimal CP value for pruning the tree was chosen to be the first level with minimum *xerror*.



The process of making a prediction once we have the model is very simple, the only preparation we did was to configure the test-dataset the same way we did the training one. The prediction produced a multi-dimensional list where every row corresponded to a user on the same index in the test dataset and every column corresponded to 5 destinations in the order of the probability that the user would pick that destination first. The AUC value and cut point with maximum cost was calculated for every model.

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset** | **AUC using Logistic Regression** | **AUC using Decision Tree** | **Cut value (at max cost)** |
| US\_NDF | 0.72 | 0.69 | 0.6273792 |
| FR\_NDF | 0.71 | 0.7 | 0.4674754 |
| CA\_NDF | 0.67 | 0.63 | 0.7055138 |
| GB\_NDF | 0.72 | 0.68 | 0.6586309 |
| ES\_NDF | 0.75 | 0.68 | 0.6177689 |
| IT\_NDF | 0.76 | 0.73 | 0.5676834 |
| PT\_NDF | 0.61 | 0.69 | 0.9330074 |
| NL\_NDF | 0.68 | 0.6 | 0.7071798 |
| DE\_NDF | 0.69 | 0.67 | 0.7105922 |
| AU\_NDF | 0.65 | 0.66 | 0.59285 |
| OT\_NDF | 0.71 | 0.67 | 0.5581279 |
| US\_OT | 0.56 | 0.56 | 0.5488066 |
| FR\_OT | 0.59 | 0.55 | 0.5645038 |
| CA\_OT | 0.54 | 0.56 | 0.4041565 |
| GB\_OT | 0.56 | 0.53 | 0.5672346 |
| ES\_OT | 0.55 | 0.53 | 0.5954935 |
| IT\_OT | 0.58 | 0.56 | 0.5936055 |
| PT\_OT | 0.52 | 0.54 | 0.5494963 |
| NL\_OT | 0.55 | 0.53 | 0.5920496 |
| DE\_OT | 0.57 | 0.6 | 0.6381377 |
| AU\_OT | 0.52 | 0.55 | 0.1705224 |
| US\_AU | 0.56 | 0.49 | 0.4581197 |
| FR\_AU | 0.47 | 0.55 | 0.3441994 |
| CA\_AU | 0.52 | 0.56 | 0.278481 |
| GB\_AU | 0.53 | 0.52 | 0.7131028 |
| ES\_AU | 0.52 | 0.54 | 0.6742857 |
| IT\_AU | 0.52 | 0.53 | 0.334974 |
| PT\_AU | 0.48 | 0.49 | 0.2442159 |
| NL\_AU | 0.55 | 0.39 | 0 |
| DE\_AU | 0.51 | 0.55 | 0.6799438 |
| US\_DE | 0.6 | 0.55 | 0.1199074 |
| FR\_DE | 0.6 | 0.52 | 0.7304171 |
| CA\_DE | 0.56 | 0.58 | 0.6173469 |
| GB\_DE | 0.54 | 0.54 | 0.1838272 |
| ES\_DE | 0.53 | 0.53 | 0.725516 |
| IT\_DE | 0.59 | 0.57 | 0.6870305 |
| PT\_DE | 0.51 | 0.57 | 0.5889815 |
| NL\_DE | 0.58 | 0.55 | 0.6884206 |
| US\_NL | 0.56 | 0.55 | 0.7036259 |
| FR\_NL | 0.52 | 0.53 | 0.1118032 |
| CA\_NL | 0.55 | 0.49 | 0.5965293 |
| GB\_NL | 0.49 | 0.54 | 0.7589425 |
| ES\_NL | 0.53 | 0.52 | 0.6555556 |
| IT\_NL | 0.55 | 0.47 | 0.4440523 |
| PT\_NL | 0.53 | 0.52 | 0.1974522 |
| US\_PT | 0.6 | 0.46 | 0 |
| FR\_PT | 0.49 | 0.59 | 0.7402062 |
| CA\_PT | 0.48 | 0.41 | 0 |
| GB\_PT | 0.54 | 0.54 | 0.5537065 |
| ES\_PT | 0.47 | 0.45 | 0.9182879 |
| IT\_PT | 0.55 | 0.53 | 0.2023121 |
| US\_IT | 0.61 | 0.56 | 0.569388 |
| FR\_IT | 0.5 | 0.52 | 0.5850642 |
| CA\_IT | 0.54 | 0.5 | 0.5660511 |
| GB\_IT | 0.52 | 0.5 | 0.3580824 |
| ES\_IT | 0.56 | 0.52 | 0.3417015 |
| US\_ES | 0.56 | 0.55 | 0.4278547 |
| FR\_ES | 0.51 | 0.54 | 0.299229 |
| CA\_ES | 0.51 | 0.55 | 0.6407686 |
| GB\_ES | 0.52 | 0.55 | 0.5360247 |
| US\_GB | 0.57 | 0.55 | 0.5817031 |
| FR\_GB | 0.53 | 0.49 | 0.6176724 |
| CA\_GB | 0.53 | 0.5 | 0.3963899 |
| US\_CA | 0.52 | 0.52 | 0.6078301 |
| FR\_CA | 0.56 | 0.51 | 0.4368679 |
| US\_FR | 0.62 | 0.58 | 0.5811152 |

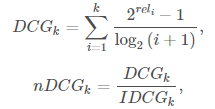
About 468 user ids on the Test data have no corresponding entry in sessions dataset. To the final Submission file, these user\_ids were appended with their prediction populated as “NDF”.

The competition required a certain format on the submission where we had to list every user and the five most likely destinations which then after uploading were evaluated using their NDCG system.

**Evaluation** **of Model**

To evaluate our solution, we are using Kaggle’s submission system. The system gives us a score between 0 and 1 where 1 is the perfect solution. Kaggle uses NDCG (Normalized Discounted Cumulative Gain) which is a system for measuring the quality of rankings.

The NDCG calculation is shown as:



where is th is the relevance of the result at position

is the maxis the maximum possible (ideal) DCG for a given set of queries. *k* is the number of predictions and *rel* is either 0 or 1, its 1 for the correct prediction and 0 otherwise. All NDCG calculations are relative values on the interval 0.0 to 1.0.

Here, for every user’s prediction, we are listing at most 5 countries in order. The score of the predictions are calculated from the position of the correct prediction. Each user prediction gets a normalized score between 0 and 1 and the total score is then calculated from the average of all user prediction scores. The score is only calculated from the position of the correct prediction and any predictions with lower probability will not reduce the score of the user prediction.

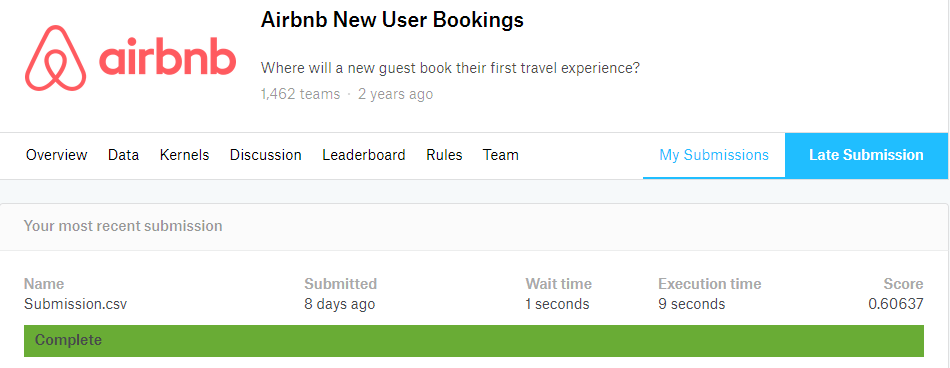
For example, if the correct destination is predicted as the most probable destination then we would get a score of 1 and if the correct destination was not part of the predicted destinations we get a score of 0. If the correct destination was predicted as the 2nd most likely destination we would get a score of 0.632. [(21-1)/log2(2+1)] = 0.632.

**Conclusion**

For every user in the dataset, the submission file contained two columns: id and country. For each new user, 5 predictions on the country of their first booking was made. The country\_destination predictions were ordered such that the most probable country\_destination went first. The file contained a header and had the following format:

id,country  
000am9932b,NDF  
000am9932b,US  
000am9932b,IT  
01wi37r0hw,FR  
etc.

The Submission File with predictions from the Decision Tree algorithm was uploaded to the Kaggle website that generated a score of 0.606.

****

The code could be found in the public git repository – <https://github.com/ssumita/Springboard-Capstone/tree/dev/Airbnb>.

**Ideas to explore**

The result on the prediction could have been improved with better knowledge on different machine learning algorithms. Due to time constraint, the only data preparation performed was modifying the data in the sessions dataset. There was no attempt made to explore or utilize user features from Training user dataset or Age-gender dataset which could have improved the predictions.

Could have analyzed the importance of the different features and minimized the number of features used in the final model, this would have sped up the training process and improved on the prediction.

Could have explored and evaluated more models using different algorithms.

**Recommendations to Airbnb**

* Improve demographic data to differentiate country destinations. Include more validation checks on Age and Gender and make these fields mandatory for users to book a reservation. This way more meaningful demographic data would be available for users who actually book reservations.
* Collect meaningful browser session activity for training users since this data was most helpful for predictions compared to the other datasets shared.  The sessions data was available only for newer users created during 2014 whereas users in the Training set dated back to 2010.
* Make the actions, action type, details in sessions dataset more meaningful since it was difficult to understand what they meant.