Airbnb new user bookings

6202 final project

I. Introduction

In this report, we use the dataset from Airbnb new user bookings in a Kaggle competition to deal with the classification problem. The label for this dataset is first destination the users went to. We use four models to train the data and check their performance by using test data and using Kaggle to score our training result. The four models we use are Naïve Bayes, decision tree, random forest and support vector machine (SVM). We first doing the data exploration for this Airbnb dataset and then preprocessing for training model. Next, we start fit the model and explain the results we get. At last, we reach a conclusion for the work we have done.

II. Data Exploration

Airbnb new user bookings dataset in Kaggle provides first booking destination decisions of users with their information. By exploring the data, we want to predict the first booking destination more accurately based on users information. In the train dataset, there are 213451 users with their decision of first booking destination and 15 features, which include 1. date variables: account created date, first active date, first booking date;

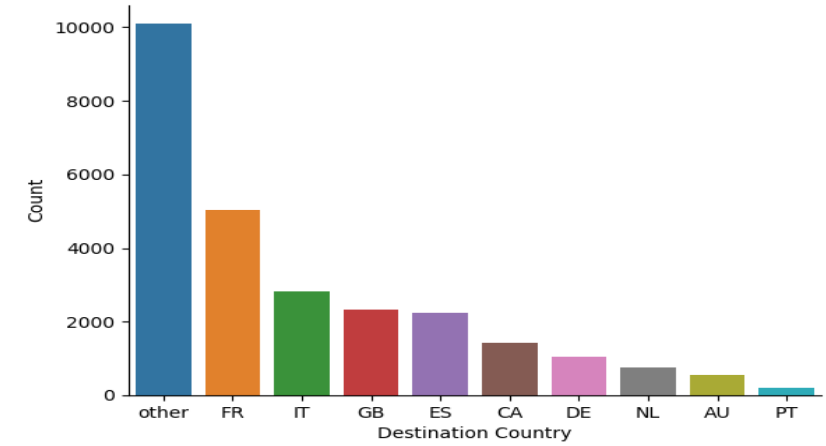
2. basic information: user id, gender, age, language preference;

3. access to Airbnb: sign up method, sign up flow, affiliate channel, affiliate provider, first affiliate tracked, sign up app, first device type and first browser. In the test dataset, there are 62096 users with 15 features which are same as the train dataset, while first booking date are all missing value. Before creating features, let us take a closer look at the train data.

i. Destinations

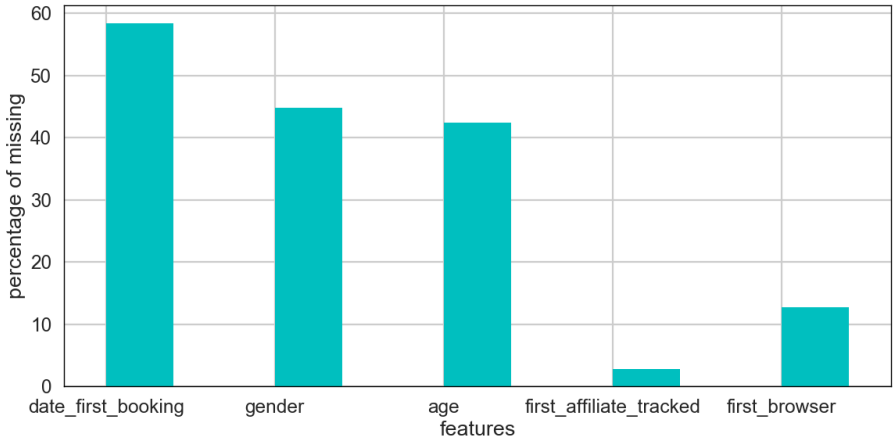
Among all 213451 users in the train data, 58.35% of users just browsing but have not book in Airbnb yet; 29.22% of users chose the United States as their first destination. The remaining 12.43% of users chose many different countries as destinations, whose distribution is shown in below figure. The destinations' location

and their counts are also shown in the map with pins in the location and marked with total population of who choose it as first destination. We also could know from the map that except the U.S., Canada and Australia, most of users in Airbnb would choose areas around Europe as their first destination.



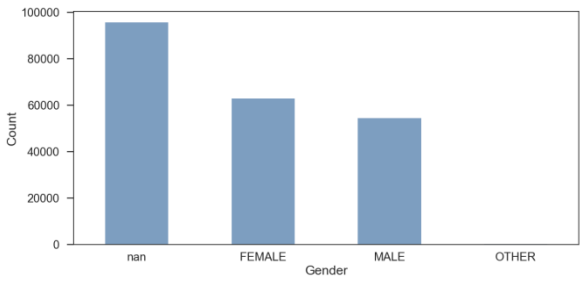
ii. Missing value

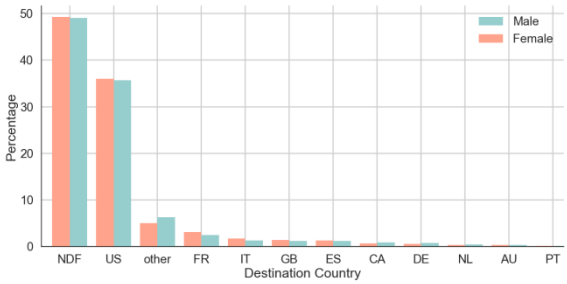
After replace “unknown” as missing value, the percentages of missing value in features with missing value are shown in figure below. As the missing value percentage of first booking date in train data is nearly 60% and there is no first date booking information in test data, we will delete this feature in model training.



iii. Basic information

First, we look at the distribution of gender. It seems like many people do not want to specify their gender in user information. As the number of Nan (which replace unknown) is larger than male and female, we would take it as a separate category in gender feature. Among those who specify their gender, female users are larger than male users. And when we look at the gender distribution on destination, we find that male and female do not have big difference in the first destination decision.

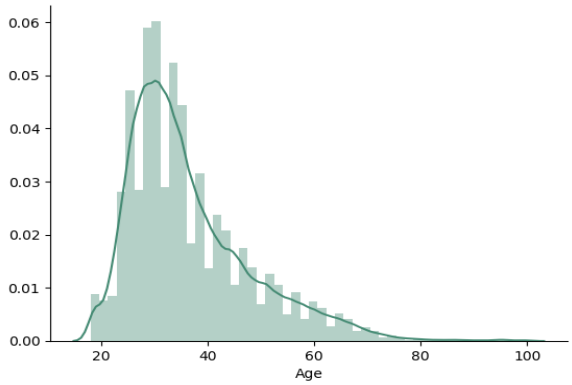


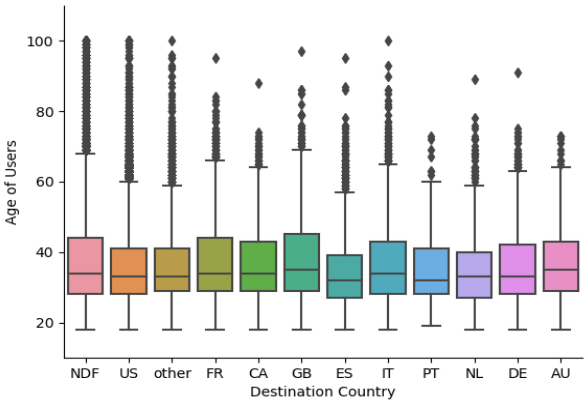


By calculating we know 96.66% of users in the train dataset prefer to use English. That might be why except for those who did not book, most of people chose the United States as their first destination.

Age value shows that there might be some input error of this feature, because Airbnb only allow people who are 18 or older to create an account, but there are 158 users below age 18. Also, the max of age is 2014, which is unreasonable as an age number. So we guess some values might be recorded wrongfully with birth year instead of age. After dealing with those outliers, the distribution of age between 18

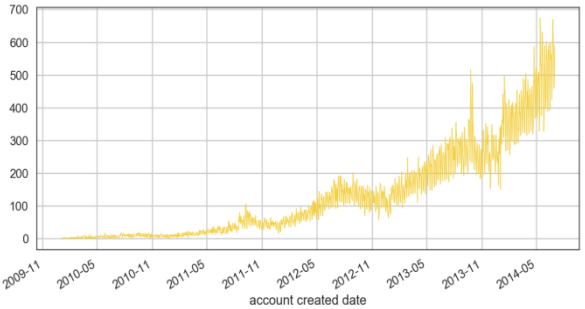
and 100 shows that users of Airbnb centers on age 25 to age 40. When we look at the age distribution in different country destinations, Spain seems to be chosen by more younger users as their first destination in Airbnb, the Great Britain seems attract users in a wider range of age.

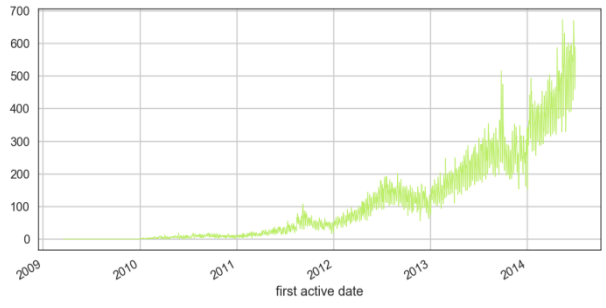




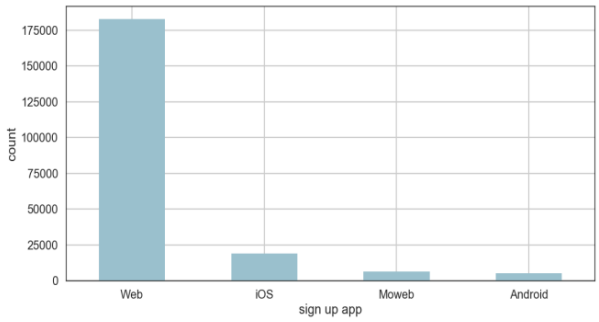
iv. Date features

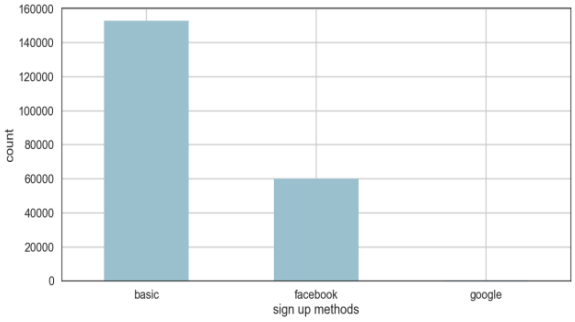
Except for feature -- first booking date, there are two features include date information: account created date and first active date. First active date could be earlier than account created date, as people might browse before signing up. After plotting them out, we find that the distributions of these two features are pretty similar and both of them have obviously increasing trend. Both of these two plots indicate that Airbnb thrive after year 2012. The wave in two figures also indicates that August to October seem to be most active period among each year. The wave in two figure also indicates that August to October seem to be most active period among each year.

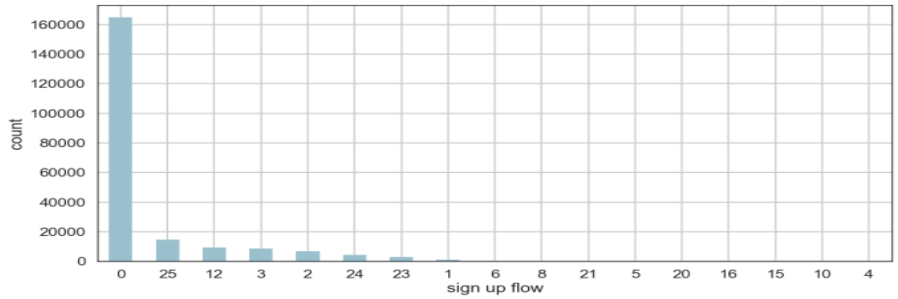




v. Other features

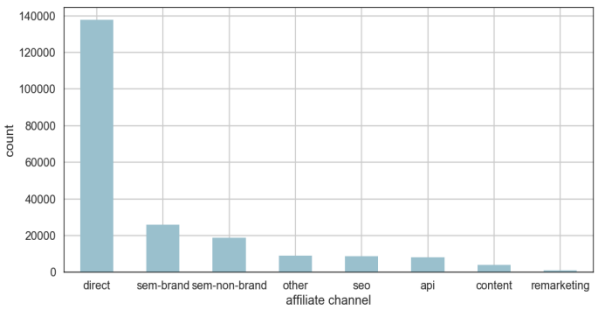


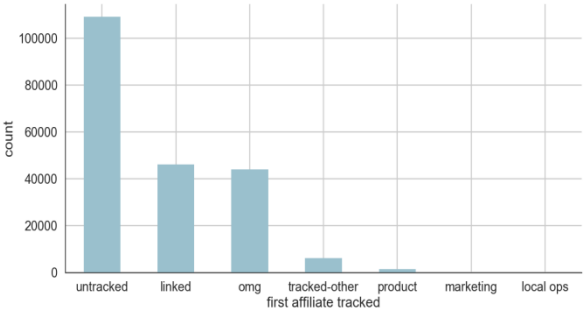


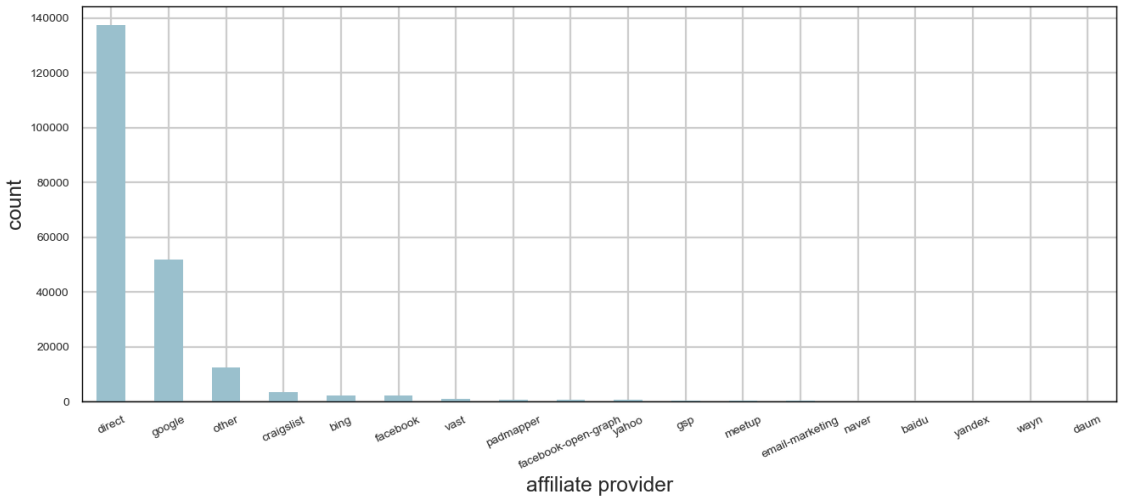


The first three figures above represent the distributions of three features about Airbnb account sign up: sign up app, sign up method and sign up flow. By calculation, we know that 85.6% of users signed up through web, the other three accesses – iOS, Moweb and Android only share with a small percentage. For sign up method, 71.63% signed up by basic method, 28.11% signed up by Facebook and only 0.26% signed up by google. For the sign up flow, we know that 77.18% of users came to signup up from the page 0.

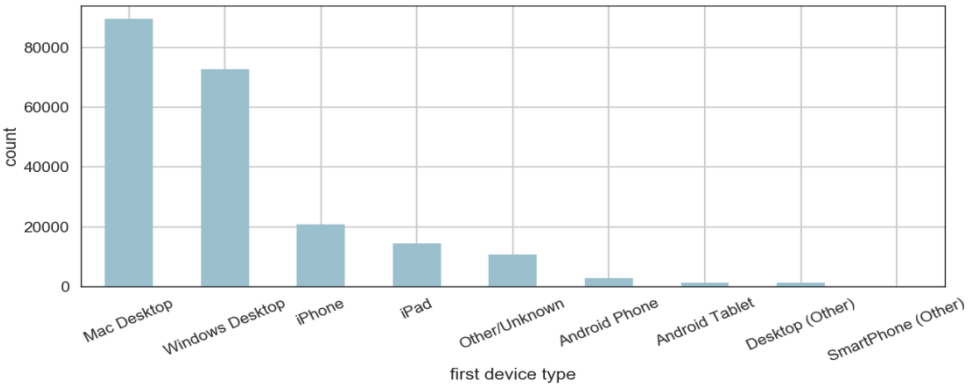
Three figures below show the distribution of three features about affiliate. By calculating their percentage in each category, we find most users came to Airbnb directly instead of going through other affiliates. There is 64.52% of users are marked as “direct” in affiliate channel; 64.38% of users are marked as “direct” in affiliate provider.



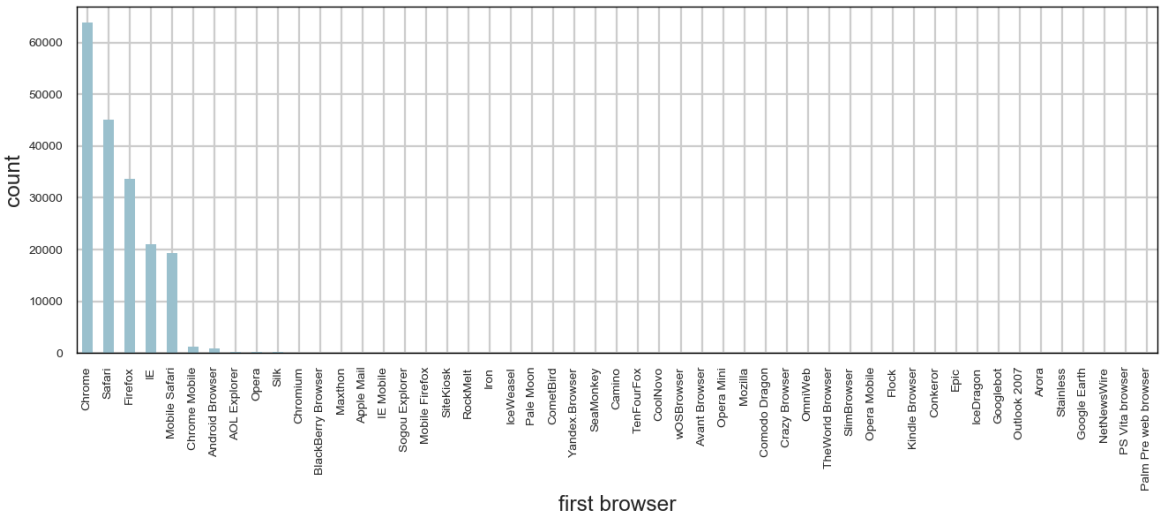




From the distribution of first device type feature, we know Mac Desktop users account for 41.98%, which is the highest share. Besides, Windows Desktop users account for 34.07%, iPhone account for 9.73%, iPad accounts for 6.72% among whole users.



From the distribution of users' first browser, we know 29.91% of users using Chrome to access Airbnb, 21.16% using Safari, which are two main shares among all the browsers listed in the train data. Also, Firefox accounts for 15.77%, IE accounts for 9.87% and Mobile Safari accounts for 9.03%.



III. Algorithm of methods

i. Naïve Bayes

Naïve Bayes is a classification technique based on conditional probability, which often used as a baseline for more complex models. The basic assumptions of using Naïve Bayes are independence between features and all features contributing equally to the target. Most of other deep learning methods would treat features with difference importance. Let us take a closer look at the algorithm of this method.

Based on Bayes Rule, we could calculate the probability of class given certain condition of features.

$$\mathrm{P(A|B)} = \frac{P(B|A)P(A)}{P(B)}$$

$$\mathrm{P(Class|Features)} = \frac{P(\mathrm{Features|Class})P(\mathrm{Class})}{P(\mathrm{Features})}$$

$P(\mathrm{Features|Class}) - -$ conditional probability of the features given the class

$P(\mathrm{Class}) - -$ probability of the class

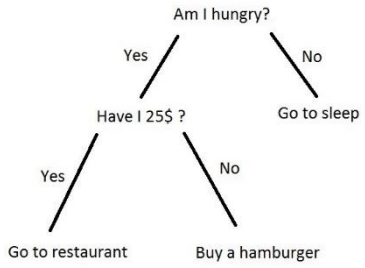
$P(\mathrm{Features}) - -$ probability of features

From previous data (train data), we could calculate the conditional probability of the features given the class label, probability of the class and probability of features. By independence assumption, probability of predictors could be estimated directly by multiplying the individual relative frequencies of each predictor.

$$\mathrm{P(C|F)} = \frac{[P(\mathrm{f1|C})P(\mathrm{f2|C}) \ldots P(\mathrm{f1|C})]P(\mathrm{Class})}{P(\mathrm{F})}$$

ii. Decision Tree

Decision trees can be applied to both regression and classification problems. As the name suggests, a decision tree is a tree-like structure where internal nodes represent a test on a feature, each branch represents outcome of a feature, and each leaf node represents class label, and the decision is made after computing all features. A path from root to leaf represents classification rules. A simple decision tree example is just like the figure below.



In this case we can get the rules that lead to buy a hamburger, that is:

- I am hungry.
- I don't have 25$.

Decision tree model's output is easy to interpret. Another advantage of a decision tree is that it is a non-parametric model so it requires little data preparation. Other techniques often require data normalization, dummy variables need to be created and blank values to be removed. Moreover, it can handle data of different types, including continuous, categorical, ordinal, and binary. It is also useful for detecting important variables, interactions, and identifying outliers

How the tree splits and grows?

The base algorithm is known as a greedy algorithm, in which the tree is constructed in a top-down recursive divide-and-conquer manner.

- At start, all the training examples are at the root.
- Input data is partitioned recursively based on selected attributes.
- Test attributes at each node are selected on the basis of a heuristic or statistical impurity measure example, gini, or information gain(entropy).
- Gini=$1 - \sum_i (p_i)^2$,where $p_i$is the probability of each label.
- Entropy = $-\mathrm{plog2(p)} - \mathrm{qlog2(q)}$,where p and q represent the probability of

success/failure respectively in a given model.

Conditions for stopping partitioning

- All samples for a given node belong to the same class.
- There are no remaining attributes for further partitioning. Majority voting is employed for classifying the leaf.
- There are no samples left.

Decision trees are simple to use, easy to understand, and offer many advantages compared to other decision-making tools, but they still don't find wide acceptance, due to its instability, overfitting and other limitations.

iii. Random forest

In order to fix the overfitting problem of decision tree, random forest was proposed. Random forest is an ensemble method. Ensemble methods aggregated more or less predictions of many different algorithms in order to increase predictive accuracy, random forest is an ensemble of decision trees. Individual decision tree sometimes will not result in best decision, so we need to combine those weak learners to get an ensemble learner.

We first draw n bootstrap samples from the original data, then grow a classification tree for each of the bootstrap samples. At each node, rather than choosing the best split among all predictors, randomly sample m of the predictors and choose the best split from among those variables. Optimizing the algorithm could be achieved by setting the number of trees grown and track the change of classification error rate. Then we could predict new data by aggregating the prediction of n trees. Moreover, random forest could output features importance by determining which variables are closest to the root of tree and then moving out.

Support Vector Machine

The key objective of SVM is to draw a hyperplane that separates the two classes optimal. In a classification problem, there is the possibility of different hyperplanes. However the objective of SVM is to find the one which gives us the highest margin.

To maximize the margin, we need to minimize $\left(\frac{2}{\|w\|}\right)$ subject to

$$y_i\ (\mathrm{W}X_i + b) - 1 \geq 0$$

for all i.

The final SVM equation can be written mathematically as

$$\mathrm{L} = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j X_i X_j$$

IV. Data preprocessing

Next, we focus on how to extract and build attributes from the training dataset.

First, we merge the two datasets, train and test, in order to dealing with data more efficiently. Then we drop out the feature date_first_booking, which does not be contained in the test dataset.

i. NaN processing

When confronted a part of the data set that had a small amount of missing values, deleting whole rows or columns was impractical. Usually the missing data comes in the way of NaN, but in the data above we can see that the gender column, language column and first_browser column containing some values being -unknown- and first_affiliate_tracked column containing some values being untracked. We transformed those values into NaN first and then summarize the percentage of unknowns in each field. The missing values for all attributes in the train and test dataset are shown below:

Attribute name	Percentage of missing value
Age	42.412365
Country_destination	22.535538
First_affiliate_tracked	2.208335
First_browser	16.111226
Gender	46.990169

The table above shows that there are 42% in age attribute being missing values. And There are 46% in gender attribute being missing value. It means that the missing

value play an important role in the gender attribute, therefore we decided to set these missing vales as -1. We treat the -unknown- as another category. Then we use the same method to deal with other attributes expect for age attribute.

We focus on age attribute independently. From the result of exploration part containing density distribution for age attribute. It is shown that there are some age values being greater than 1000, which is obvious impossible. This might be caused by using the born year as age. So we transfer these 'born-year' data as 'age' data. What's more, the Airbnb require the users should be older than 18 years old. Then we also set these people whose age is younger than 18 as Nan, whose values are -1. And for some people whose ages are bigger than 100, we also set its age as -1.

ii. Feature engineering

- Date-related attributes

The raw data just put a series of numbers in the date-related attributes, date_account_created and timestamp_first_active. And then we split the date_account_created into day, week, month, year, which form four new features. At the same time, splitting the timestamp_first_active into day, weekday, week, month and year, which form five new features. And based on these two attributes we create a new feature called time_lag, which is the difference value between these two.

- Age attribute

After dealing with the NaN in the age column, we create age_group feature to set 14 categories, which divide the age into different goup. It will help us to know different age groups have different preference when using Airbnb.

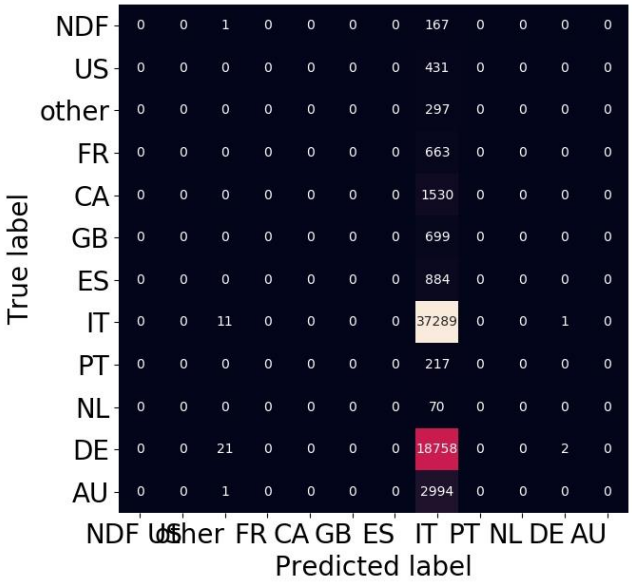
- Other dummy attributes

There are some remaining dummy attributes containing gender, signup_method, signup_flow, language, affiliate_channel, affiliate_provider, first_affiliate_tracked, signup_app, first_device_type and first_brower. We encode these category attributes.

V. Model Engineering

i. Naïve Bayes

We split data into 30% and 70% two groups. The group has 70% of whole train data as train, 30% of train data as X test. After fitting the Naïve Bayes, we get an accuracy of 58.24 and confusion matrix of prediction on X _test, which shown in below. The accuracy on X test is pretty low. We guest it is because some features in the data are not independence with each other, and some features might not have a stationary mean or standard deviation. From the confusion matrix, we could find that when predicting for label, only Italy and Germany have correct true positive prediction. Also, most of true label Germany ones would be wrongfully labels as Italy.



Also, after upload the file into Kaggle, we get a score of 0.84255. It is a little bit weird that we get a pretty low accuracy in X test but getting a high score in Kaggle.

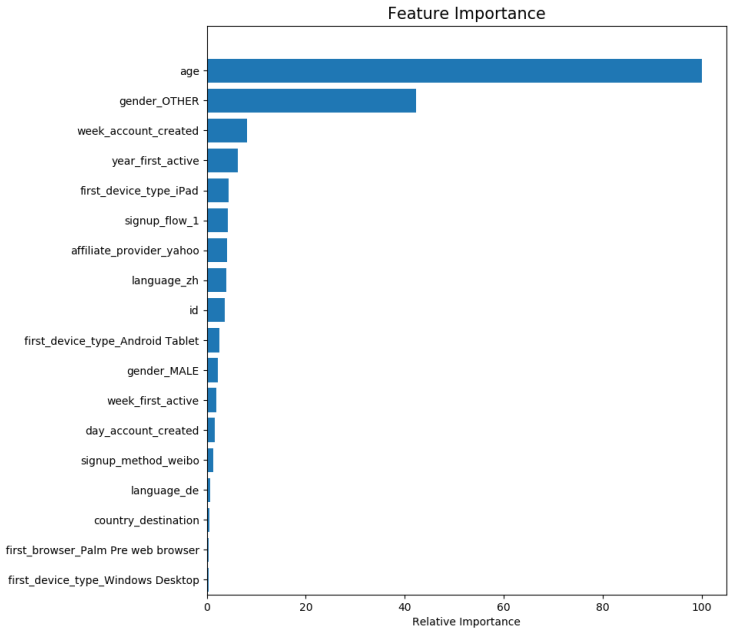
Name	Submitted	Wait time	Execution time	Score
sub_nb1.csv	5 hours ago	0 seconds	9 seconds	0.84255

ii. Decision tree

We use decision tree for classification, the splits are chosen so as to minimize entropy or Gini impurity in the resulting subsets. The resulting classifier separates

the feature space into distinct subsets. Prediction of an observation is made based on which subset the observation falls into. The decision tree is suitable for this kind of problem, because this is a multiclass classification problem.

The result of feature importance of the attributes is shown below:



It shows that by using decision tree the most important feature is age attribute and second important attribute is gender_OTHER. And the relatively useless attribute is first browser used by users. From the result of decision tree, we notice that the root node for this training method is age_group attribute. And during the fitting process, we use both entropy and gini method to build the tree. The accuracy of these two methods are shown below:

Results Using Gini Index:

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	168
1	0.00	0.00	0.00	431
2	0.00	0.00	0.00	297
3	0.00	0.00	0.00	663
4	0.00	0.00	0.00	1530
5	0.00	0.00	0.00	699
6	0.00	0.00	0.00	884
7	0.70	0.83	0.76	37301
8	0.00	0.00	0.00	217
9	0.00	0.00	0.00	70
10	0.49	0.51	0.50	18781
11	0.00	0.00	0.00	2995
avg / total	0.55	0.64	0.59	64036

Accuracy : 63.523642950840156

Results Using Entropy:

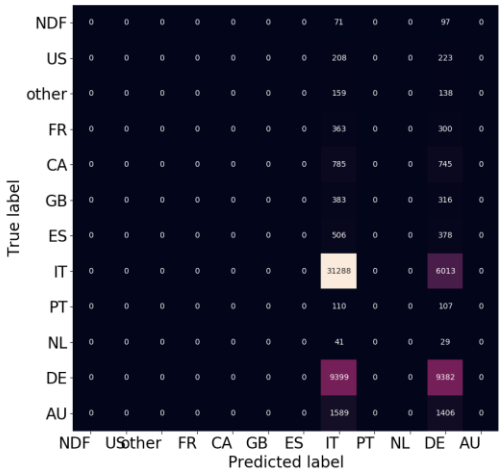
Classification Report:

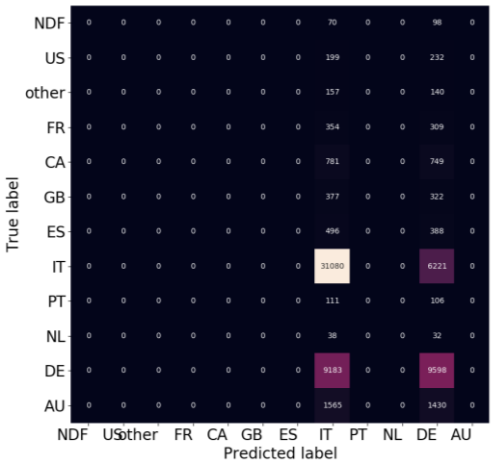
	precision	recall	f1-score	support
0	0.00	0.00	0.00	168
1	0.00	0.00	0.00	431
2	0.00	0.00	0.00	297
3	0.00	0.00	0.00	663
4	0.00	0.00	0.00	1530
5	0.00	0.00	0.00	699
6	0.00	0.00	0.00	884
7	0.70	0.84	0.76	37301
8	0.00	0.00	0.00	217
9	0.00	0.00	0.00	70
10	0.49	0.50	0.49	18781
11	0.00	0.00	0.00	2995
avg / total	0.55	0.64	0.59	64036

Accuracy : 63.5111499781373

The results tell us that using these two methods make no big difference for accuracy of decision tree. In fact, using entropy is computational expensive. But we still fit two models by using both methods.

And about the results of confusion matrix are shown below:





The confusion matrices are derived from entropy and gini index from left to right. The results of both seem similar. They both show that the decision tree misclassified IT and DE. There are 23.100% of IT has been classified to DE. And there are 39.058% of DE has been misclassified to IT. It shows that many predicted labels has been classified to IT and DE incorrectly.

The results in Kaggle are as following:

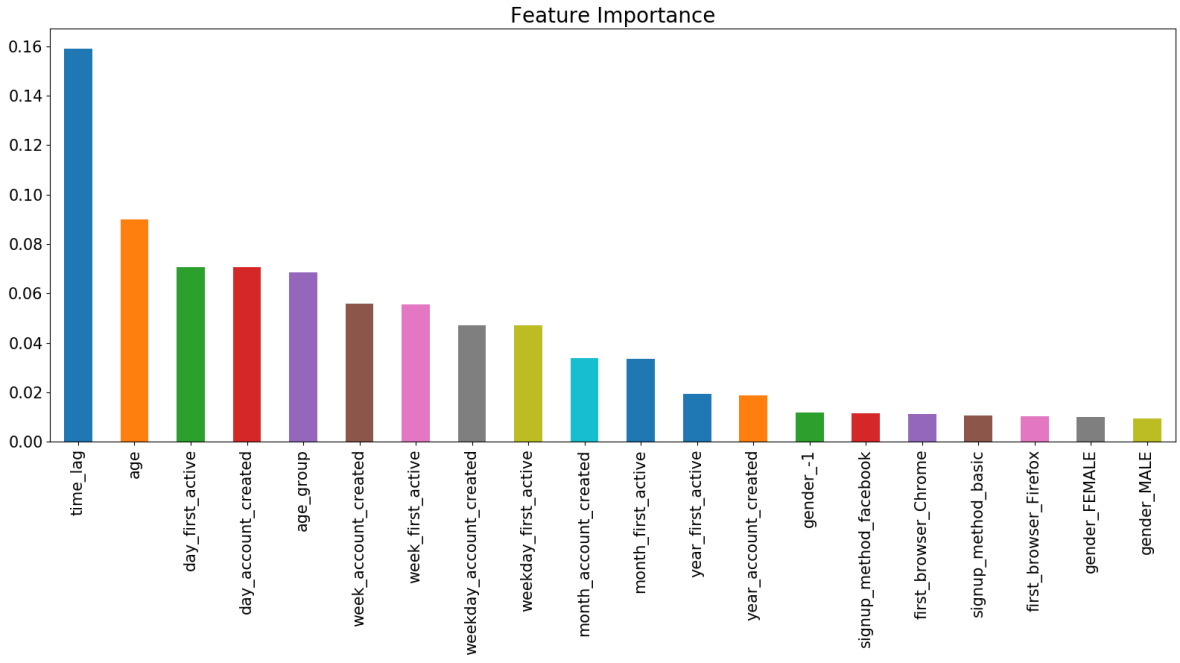
sub_dt_en.csv 4 hours ago by Linl0907 add submission details	0.86884	0.86403
sub_dt.csv 4 hours ago by Linl0907 add submission details	0.86916	0.86394

The results show that the decision tree performed well on this destination classification problem.

iii. Random Forest

A random forest is simply a collection of decision trees whose results are aggregated into one final result. When inputting a training dataset with features and labels into a decision tree, it will formulate some set of rules, which will be used to make the predictions. Compared with decision tree, the Random Forest randomly selects observations and features to build several decision trees and then averages the results.

The result of feature importance of the attributes is shown below:



From the plot above, we can see that the most important feature is time_lag, which is the time difference between the users create an account to the users first

browsing Airbnb. And the second significant feature is age, which indicates that the destination of Airbnb users' choice is sensitive to the age attribute. We just show the first twenty important features above. The result of feature importance also told us that sign-up method, the first browser the users used and the gender of users are not that important for the destination the users chose.

The result of the accuracy for the random forest as following:

Results Using All Features:

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	168
1	0.00	0.00	0.00	431
2	0.00	0.00	0.00	297
3	0.02	0.00	0.01	663
4	0.05	0.01	0.01	1530
5	0.03	0.00	0.01	699
6	0.03	0.00	0.01	884
7	0.67	0.81	0.73	37301
8	0.00	0.00	0.00	217
9	0.00	0.00	0.00	70
10	0.46	0.45	0.45	18781
11	0.06	0.01	0.02	2995
avg / total	0.53	0.60	0.56	64036

Accuracy : 60.10837653819726

Results Using K features:

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	168
1	0.00	0.00	0.00	431
2	0.01	0.01	0.01	297
3	0.01	0.01	0.01	663
4	0.02	0.01	0.02	1530
5	0.02	0.01	0.01	699
6	0.02	0.01	0.01	884
7	0.65	0.70	0.68	37301
8	0.00	0.00	0.00	217
9	0.00	0.00	0.00	70
10	0.39	0.42	0.41	18781
11	0.06	0.03	0.04	2995
avg / total	0.50	0.53	0.52	64036

Accuracy : 53.4402523580486

By using all features in the dataset, the accuracy is 0.601. Then we used 15 most important attributes to fit the model. The result was supposed to be better than using all features, since we dropped the useless features and remain those who are significant. But the fact is, using 15 important features perform worse than using all of them. The accuracy reduced to 0.534 when using these features. It seems that the accuracy will decrease dramatically if we use less features to fit models, even if these features are not that important.

Then we tested the prediction in Kaggle made by using the model trained by all of the features. The result is shown below:

sub_rf.csv a few seconds ago by Linl0907 add submission details	0.86036	0.85726

The result seems well in this classification problem. The score is no better than the one which got by decision tree algorithm, which is 0.86884. So random forest did not show its advantage in this dataset. Random forest theoretically performs better

than decision tree. The reason maybe that the parameters that I set randomly are not the best one to fit this dataset. I think after tuning the parameter, random forest could give a higher score of accuracy.

iv. SVM

The accuracy score of SVM is shown below, which indicates that it performs well just like decision tree.

```
Results Using SVM:

Classification Report:
             precision    recall  f1-score   support

          0       0.00      0.00      0.00       168
          1       0.00      0.00      0.00       431
          2       0.00      0.00      0.00       297
          3       0.00      0.00      0.00       663
          4       0.00      0.00      0.00      1530
          5       0.00      0.00      0.00       699
          6       0.00      0.00      0.00       884
          7       0.68      0.86      0.76     37301
          8       0.00      0.00      0.00       217
          9       0.00      0.00      0.00        70
         10       0.49      0.45      0.47     18781
         11       0.00      0.00      0.00      2995

avg / total       0.54      0.63      0.58     64036

accuracy: 63.155100256105946
```

And the score given by Kaggle is, which shows below.

Although SVM shows a high AUC score on Kaggle. I don't think it is an appropriate algorithm according to this dataset. Since the dataset is too big to train and it is a multi-class problem, which leads time consuming when fitting the model. And it costs the computer more than 1 hour to train 70% of the dataset. Compared to decision tree, SVM is more computational expensive and less accurate.

VI. Conclusion

From the results above, we could see that decision tree performs better. It is outstanding not only for its accuracy, and it is less expensive than the other models when manipulated on the computer. By comparing the performance of different methods with their accuracy, we could find that decision tree and support vector

machine work better in splitting test data, and the difference between those methods in accuracy is big. However, when we upload predictions on test to Kaggle, there is not big difference, while decision tree still works the best.

VII. reference

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112). New York: springer.

Watt, J., Borhani, R., & Katsaggelos, A. (2016). Machine learning refined : foundations, algorithms, and applications . Cambridge, United Kingdom: Cambridge University Press.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. Cambridge, Massachusetts: The MIT Press.

Swamynathan, M. (2017). Mastering Machine Learning with Python in Six Steps: A Practical Implementation Guide to Predictive Data Analytics Using Python. Berkeley, CA: Apress. doi:10.1007/978-1-4842-2866-1