I. Project Overview

These days booking for a holiday has become easier with the online booking facility offered by various travel/holiday companies. With the online booking option, customer can choose and compare the available options and finalize on the best suitable option without a need to physically visit the booking office.

For travel companies such as Thomas cook, AirBnB, it would be advantageous to offer best suitable option to its customers based on certain understanding about their customers.

If the travel companies could understand the type of customer and could predict the likelihood of booking to a particular destination or type of holidays, this will help the travel companies to display the most favorable option.

A. Problem statement

To start with a constructive example, I shall be picking up Airbnb as a travel company which offers online holiday booking portal.

First criteria for the company is to predict in which country a new customer will make his or her first booking. The task is to make use of data from existing customers and try to understand the features which will influence the new customers to make booking at a particular country. These features could be popularity of the destination, age, gender of the customer and various other factors. The dataset contains most of the information required for the machine learning model to learn.

So in this project I shall be using the data from Kaggle to train various machine learning models and test the model performance against the testing data. The task would be to learn from the features presented in the training data and understand which factors influence the customer to book his/her holiday at a particular destination country and use this learning to predict destination country on a testing data.

B. Metrics

Once the machine learning model is trained it is important to verify the learning model. This is done by exercising the trained model against a new set of data, we call it testing data and see how good the model can predict. There are many metrics available for measuring the predictions such as accuracy, F1- score.

Accuracy is a measure of number of correct predictions made by the model divided by the total number of predictions made, whereas F1-score includes calculation of two important factors: *precision* and *recall*.

Where *precision* is defined as the fraction of relevant instances among the retrieved instances and it is calculated as follows:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive + False positive}}$$

And *recall* is defined as the fraction of relevant instances that have been retrieved over total relevant instances in the dataset and it is calculated as follows:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive + False negative}}$$

F1 score is calculated as follows:

$$\text{F1} = \frac{2 * \text{precision}* \text{recall}}{\text{precision + recall}}$$

Precision will let us calculate the percent of positive predictions in the testing dataset, whereas recall will let us calculate the percent of the positive cases in the complete dataset. Basically recall considers the positive cases in the complete dataset. For instance, if the training dataset contains 1000 samples and there are 100 positive samples. If we happen to pick 200 samples and found that 60 of them were identified correctly, then our precision will be

Actual Results

		Positive	Negative
Predicted results	Positive	60	40
	Negative	140	760

$$\text{Precision} = \frac{60}{60 + 40} = 60\ \%$$

$$\text{Whereas recall will be} = \frac{60}{140 + 60} = 30\%$$

So it is important to calculate not only the percent of correct results we find in the testing data (precision), but it is also important to calculate the percentage of positive cases in the overall dataset (recall). Hence I shall be using F1-score as accuracy metrics to measure the performance of models.

II. Analysis

A. Data Exploration

The dataset is used from Kaggle competition. Following are the datasets available from the website and these will be used as part of the project:

1. train_users.csv - the training set of customer booking. This dataset contains following features:

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	: Male/Female, but some records are marked as unknown.
Age	: Age of the customer.
signup_method	: Mechanism used for sign up with AirBnB.
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's the first marketing the user interacted with before the signing up
signup_app	: website
first_device_type	: device type used for booking.
first_browser	: Browser used
country_destination	: this is the target variable we train the model and predict against the testing data.

2. sessions.csv - web sessions log for users. The sessions.csv contains following features:

user_id	: to be joined with the column 'id' in users table
Action	: Action performed on the website, could be lookup, search results
action_type	: Action type, could be click, data displayed.
action_detail	: Details of the action.
device_type	: Device used, could be desktop/any mobile device
secs_elapsed	: time spent on the action

3. countries.csv - summary statistics of destination countries in this dataset and their locations:

country_destination	: Country name
lat_destination	: Latitude
lng_destination	: Longitude
distance_km	: Distance in km.

- destination_km2 : Distance in km2.
- destination_language : Language spoken in destination country
- language_levenshtein_distance:Similarity in terms of language distance

4. age_gender_bkts.csv - summary statistics of users' and it contains following set of features:

- age group : Age group of people visiting the destination.
- Gender : Gender of the customer.
- destination country : Destination country
- Year : Year of booking made.
- population (in thousands): Population.

B. Exploratory Visualization

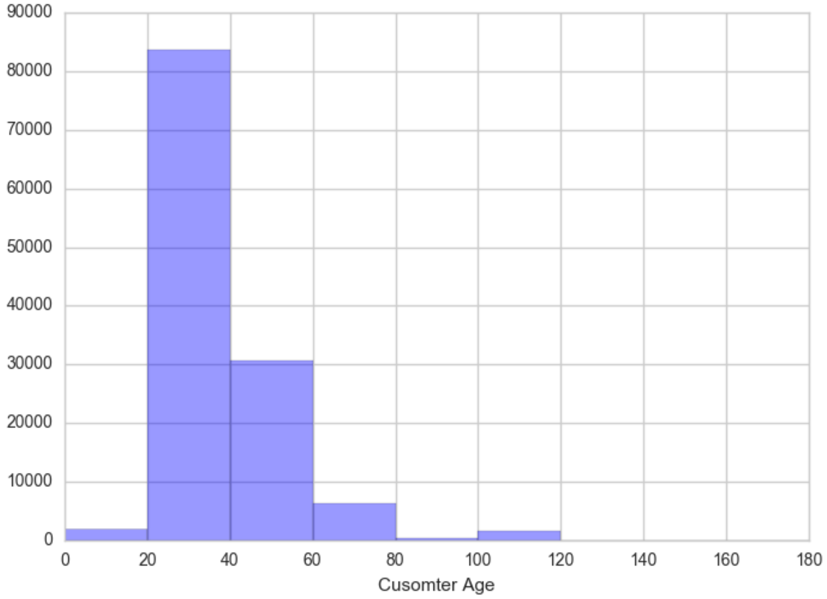
1. Training dataset

The training dataset contains majority of the training data. Each row represents one customer, with the columns containing various information such the customers' age, agender, date of first booking, device used and other information as described in the previous section. The dataset also contains the 'country_destination', which will be predicted in this project.

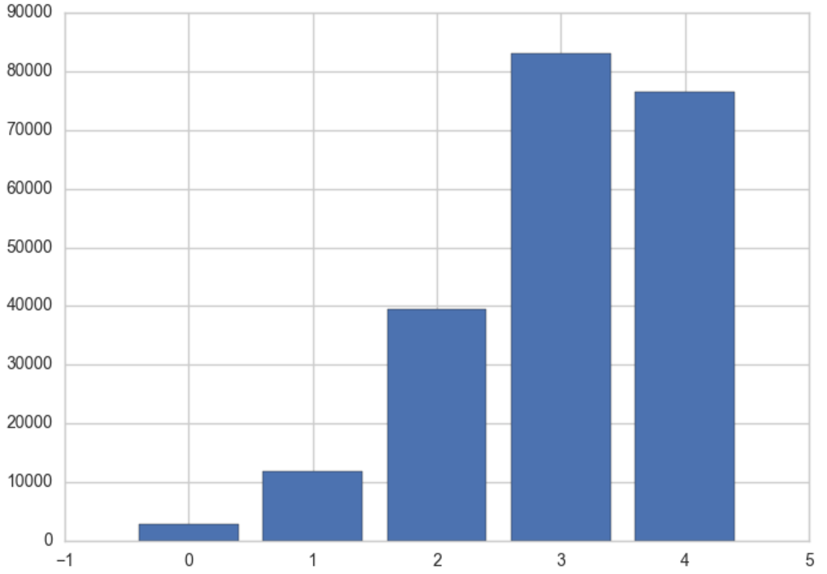
The training data spans over 4 years, starting from 28th June 2010 to 30th June 2014. Each customer is identified by means of unique id field. This field can be used as primary key while trying to match relevant information from various csv files in the dataset.

Looking at the dataset, following observations can be made:

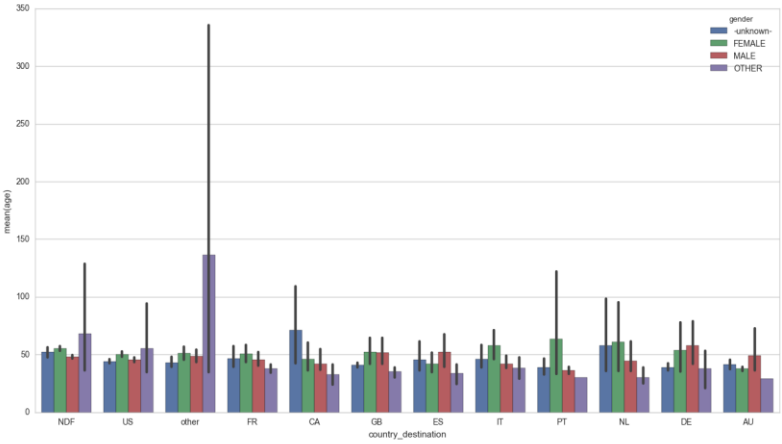
- Firstly, the date_first_booking feature in the dataset contains date of the first booking made by the customer. However not all customers made the booking, so this field is filled with empty value for those customers who hasn't made their booking, this is represented as 'NDF' (No Destination Found) in the 'country_destination' field.
- Secondly, the age groups of customers. Below fig represents the customer age groups in the dataset. We can clearly see that some customers have specified 0 or less than 10 as their age and some of them specified their age greater than 100. I think we cannot rely on the data entered by those customers and it may not be useful for modeling. Basically we will cleanup this along with other features. This is something we will be handling as part of the implementation.



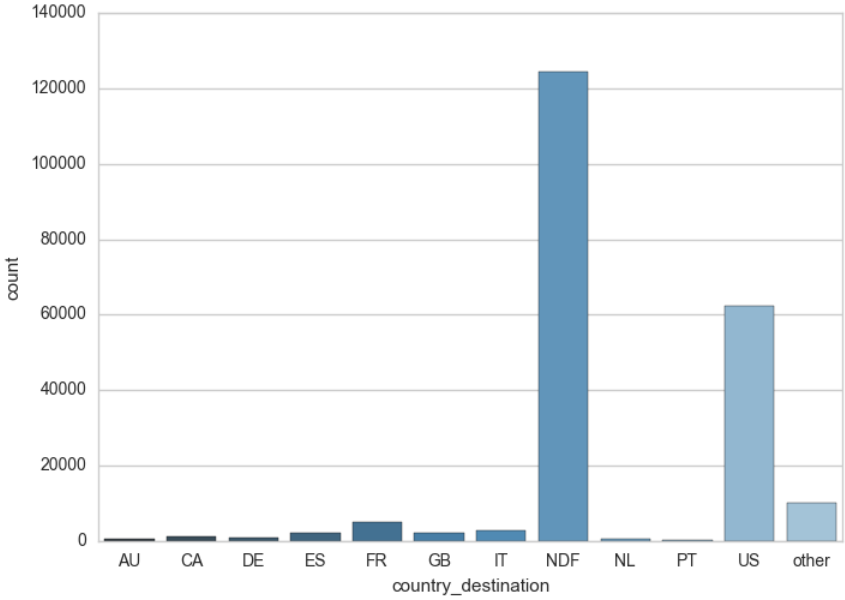
- Third observation is about the number of bookings made in a year. Below fig depicts that the number of bookings has seen increasing trend year-on-year. We might notice that the last element in the bar graph is smaller, this is because we have data only for the first half of the year, hence we are showing first 6 months. I expect that there will be many more bookings in the second half of the year.



- Fourth observation is about relationship between country destination and age group and gender. Looking into these two features (age and gender) we found that females outnumber when it comes to booking and its true for most of the countries.



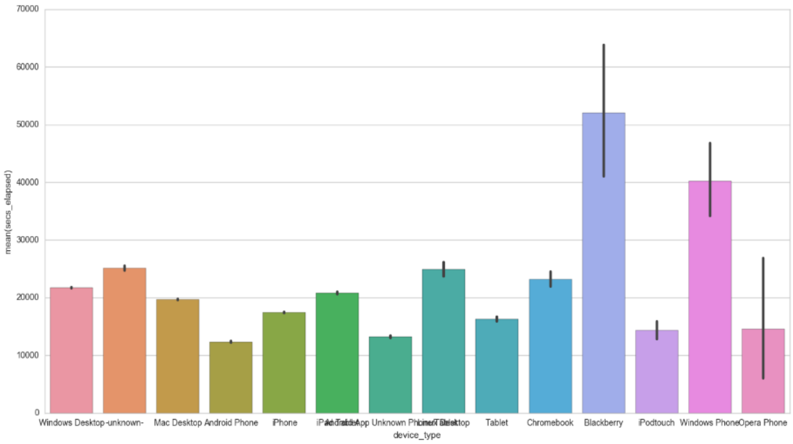
- Finally, when we look at the most desirable destination among the customers, we found that US is at the top of the list followed by other European countries. However, majority of people did not manage to make any bookings. Below figure shows the matplot:



2. sessions.csv

The sessions csv file captures exhaustive list of actions performed by the user. These actions could be clicking various links in order to create wish list, searching for destination, modifying the search criteria and other various actions. In addition to this, the CSV file provides time spent during each action on a particular device, which will help us in understanding intended action by the customer and the device used for making the booking. For example, if a customer spends lot of time in exploring the destination, which is more likely to make booking around that location/destination when compared with spending less time in exploring the destination.

Below bar chart depicts data extracted from the sessions.csv



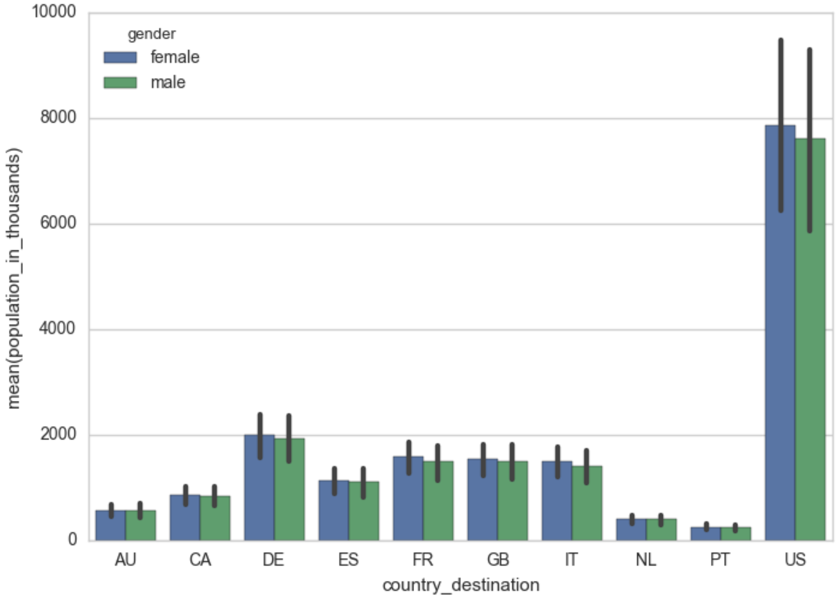
As we can see that customers have spent majority of their time on mobile devices rather on desktop device.

3. countries.csv

The countries csv file captures details about the destination, its location in terms of longitude and latitude. In addition to this it all also describes language spoken in the destination. This piece of information could be taken into account when the customer considers booking his holiday.

4. Age-Gender-buckets.csv

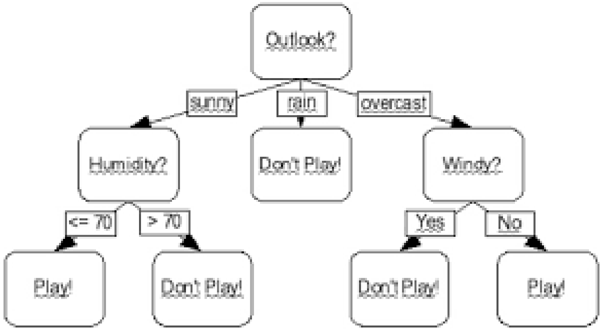
This csv file contains valuable information such as range of age groups of people (male/female) visiting the destination country. In addition to it the file also contains information about population visiting the country for a given age group in a year. This will help the model to learn the choices done by different age groups, male/females. Apart from this we can also deduce favorite destination by looking at the population count in each destination country.



C. Algorithms and Techniques

There are plenty of models such as Decision Trees, SVM, KNN, Linear Regression that could be used to classify input data. However not all of them provide the best results due to nature of classification problem. Some classification models would take more time to train the model and also models are meant for solving particular set of classification problems. For example, SVM is best suited for text prediction, whereas linear regression model expects the data to be linearly separable. However, Decision tree does not have any such requirement.

In the simple case, we shall be picking up Decision tree supervised classification model. Decision tree builds classification in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A sample decision tree about to play outdoor or not is shown below. The features include outlook, humidity and windy.



The biggest problem about decision trees is the problem of overfitting. In the example above, parameters have been set to stop model splitting once the population of records at a given node gets too small (minimum split) and when a certain number of splits have occurred ('maximum depth'). These values have been set to prevent the tree from growing too large. The reason for this is that if the tree gets too large, it will start modelling random noise and hence will not work for data not in the training dataset (it will not 'generalize' well)

There is another variant of decision tree, called Boosting, which overcomes issues of decision tree. Boosting works by building trees iteratively such that each instance of the tree learns from earlier tree and this process goes on. The depth/number of trees are configurable. Each time a tree is built, it finds the data which are misclassified by that tree and assigns a heavy weight (high importance) than the ones which were correctly classified. The following tree will build a new tree based on these weights and this process continues depending on the number of trees configured in the model. This model has got following advantages:

a. This model is based on iterative model, whereby the model builds with basic set and learns from earlier trees.
b. Can handle large amounts of data
c. XGBoost is more flexible, i.e. has more customizable parameters
d. XGBoost is fast, in terms of learning.

For these reasons, I will be exercising advanced version of decision trees called as XGBoost model.

D. Benchmark

In addition to predicting the first most likely destination for AirBnB customer, I would like to compare the performance of some of the supervised models such as Decision Trees, K-NN, SVM, Gaussian and XGBoost. The comparison will be based on **model evaluation done using** F1-score. Finally, I will try to improve the prediction by fine tuning the hyper parameters for XGBoost.

Since DecisionTree alogirthm is our first classification model, the results of this model which will be set as a benchmark and then it will be compared against advanced version of decision tree, which is XGBoost model.

As we know that the dataset is used from Kaggle competition, the output/target results are not published. So I shall be just using training data and split that into train and test. The F1-score for decision tree is as follows.

Country	**Precision**	**Recall**	**F1-score**	**Support**
AU	0.00	0.00	0.00	30
CA	0.00	0.00	0.00	90
DE	0.00	0.00	0.00	49

ES	0.02	0.02	0.02	149
FR	0.03	0.04	0.03	274
GB	0.01	0.01	0.01	151
IT	0.03	0.03	0.03	203
NDF	0.75	0.73	0.74	9046
NL	0.00	0.00	0.00	52
PT	0.00	0.00	0.00	18
US	0.42	0.43	0.43	3971
OTHER	0.07	0.07	0.07	730
avg / total	***0.57***	***0.57***	***0.57***	***14763***

III. Methodology

A. Data Cleansing

The very first step in data processing involves analysing the data and finding out any inconsistency. As mentioned in earlier sections that the AirBnB dataset is a real data feed by customers, it is obvious that some of the data is either missing or it is not in the format expected for data analysis/ may contain some incorrect values, so we will be carrying out data cleansing.

The first feature is 'age', which contains values less than 10 and some were more than 100. Customers with age between 16 and 85 are filtered out and remaining customers age is filled with mean of all the valid customer's age. However there are many options to fill with such as :

- A constant value that has meaning within the domain, such as 0 or -1, distinct from all other values.
- A value from another randomly selected record.
- A median or mode value for the column.

I choose to use mean of all the valid customer's age and assigned them to customer whose age was out of range (16-85)

The 'date_first_booking' feature contains values for only customers who made booking, while for others this field is blank and the destination_country is NDF. So I removed this feature, otherwise the model will get biased.

Since the dataset contains booking information, so there are many date related features, such as 'date_account_created', 'timestamp_first_active' and 'first_affiliate_track'. The values in these fields are re-formatted to pandas date time format, because it will be easy to perform various operations such as comparison, differences.

B. Normalization

Next step in data preprocessing is normalization, which is one of the important step in data processing, whereby we transform categorical features from ordinal values into nominal values. Normalization is done in order to improve the performance of classification models. I will be normalizing all the categorical features present in the dataset:

1. gender
2. signup_method
3. signup_flow
4. Language
5. affiliate_channel
6. affiliate_provider
7. first_affiliate_tracked
8. signup_app
9. first_device_type
10. first_browser

There are two ways one could carry out normalization, one-hot encoding and LabelEncoder. One-hot encoding translates the categorical data into binary representation resulting one column for each categorical data, whereas latter method "condenses" the information by changing things to integers.

One of the advantage of one-hot encoding is that all our categorical data will be at the same (hamming) distance from each other, whereas label encoder might assign different values for each of the levels in the features, which could be misleading. Based on this, we will be using one-hot encoding methodology to normalize the data.

In this method, the range of values assigned to each feature shall be split into columns. For instance, signup_method takes up two values "basic" and "Facebook". As a result of normalization, we end up having two columns (signup_method_basic and signup_method_facebook) in the dataset, one for each option. As a result of this step, we end up having many features in the dataset.

C. Implementation

Before we start training the model using the data from the previous step, we need to extract relevant data which is spread across multiple csv files and stich them together in one big data, which can then be feed to the supervised model for training.

As mentioned earlier the sessions csv file describes about user sessions made using booking for each customer, which has got very relevant information – time spent in each session, which is key feature in predicting the classification outcome. I noticed that there are multiple rows for each customer, each row describing different session, possibly using different device. Before I merge the data into single database, I would like to refactor some of the information. Each customer row has got same information except the time spent and the device. So I would like to categorize the device type into "main" and "backup" device and sum up the time spent into either "main" or "backup" category, depending on the amount of time spent during each session. Once this is done, we need to club these two data sets.

Again I used one hot encoding to "device_type" feature and converted into main_device and backup_device. Next we need to merge these two data ("main" and "backup") into single pandas dataframe. Finally, we merge the information extracted from sessions information with the training data into one big training data using pandas concat API.

Once we have the training data available, we then train various supervised models (SVM, KNN, Gaussian and XGBoost) making use of sklearn library.

Finally, we evaluate the performance of model using sklearn classification report and the results are mentioned in the results section.

D. Refinement

In this section we will go through the various tuning parameters for XGBoost algorithm. As we know that XGBoost has got plenty of arguments and most of them has got default values, but still there are quite a few options which we need to define, for instance, number of trees, depth of tree and some other parameters. Tuning the parameters is bit time consuming process, so I have used GridSearchCV to run through various tuning parameters and let the model pick the best among them.

Following parameters of XGBoost were tuned:

i. max_depth: [3, 10] - The maximum depth of a tree. This parameter is used to control over-fitting as higher depth will allow model to learn relations very specific to a particular sample.

ii. learning_rate: [0.1, 0.3]- learning rate. Makes the model more robust by shrinking the weights on each step

iii. n_estimators: [50, 100] - Number of boosted trees to fit.

IV. Results

A. Model Evaluation

As mentioned earlier, this is a AirBnB competition, whereby the outcome of the testing data is not published, hence I have to make use of just the training data. The training data is split into training and testing data in the ratio of 80-20. The 20% of the data is used for validation, whereas 80% of the data is trained using K fold cross validation.

B. Justification

1. XGBoost results

   Best score: 0.703312334891

country	precision	recall	f1-score	support
AU	0.00	0.00	0.00	30
CA	0.00	0.00	0.00	90
DE	0.00	0.00	0.00	49
ES	0.00	0.00	0.00	149
FR	0.00	0.00	0.00	274
GB	0.00	0.00	0.00	151
IT	0.00	0.00	0.00	203
NDF	0.77	0.88	0.82	9046
NL	0.00	0.00	0.00	52
PT	0.00	0.00	0.00	18
US	0.54	0.61	0.57	3971
OTHER	0.00	0.00	0.00	730
avg / total	***0.62***	***0.70***	***0.66***	***14763***

2. Training the KNN using training data

   Predicting the results using KNN: 0.511074984759

country	precision	recall	f1-score	support
AU	0.00	0.00	0.00	30
CA	0.01	0.02	0.01	90
DE	0.00	0.00	0.00	49
ES	0.01	0.01	0.01	149
FR	0.01	0.03	0.02	274
GB	0.01	0.02	0.01	151
IT	0.01	0.02	0.01	203
NDF	0.63	0.78	0.69	9046
NL	0.00	0.00	0.00	52
PT	0.00	0.00	0.00	18
US	0.31	0.13	0.18	3971
OTHER	0.12	0.01	0.01	730

avg / total	***0.47***	***0.51***	***0.47***	***14763***

3. Training the MLPClassifier training data

   Predicting the results using MLPClassifier: 0.615254352096

country	**precision**	**recall**	**f1-score**	**support**
AU	0.00	0.00	0.00	30
CA	0.00	0.00	0.00	90
DE	0.00	0.00	0.00	49
ES	0.00	0.00	0.00	149
FR	0.00	0.00	0.00	274
GB	0.00	0.00	0.00	151
IT	0.00	0.00	0.00	203
NDF	0.64	0.92	0.76	9046
NL	0.00	0.00	0.00	52
PT	0.00	0.00	0.00	18
US	0.44	0.20	0.28	3971
OTHER	0.00	0.00	0.00	730
avg / total	***0.51***	***0.62***	***0.54***	***14763***

4. Training the GaussianNB training data

   Predicting the results using GaussianNB: 0.600487705751

country	**precision**	**recall**	**f1-score**	**support**
AU	0.00	0.00	0.00	30
CA	0.00	0.00	0.00	90
DE	0.00	0.00	0.00	49
ES	0.00	0.00	0.00	149
FR	0.05	0.02	0.04	274
GB	0.00	0.00	0.00	151
IT	0.00	0.00	0.00	203
NDF	0.63	0.94	0.75	9046
NL	0.00	0.00	0.00	52
PT	0.00	0.00	0.00	18
US	0.37	0.09	0.15	3971
OTHER	0.00	0.00	0.00	730
avg / total	***0.48***	***0.60***	***0.50***	***14763***

v. Conclusion

As part of the project, I was able to exercise various classification models to predict the first destination country made by AirBnB's customer.

A. Reflection

I felt that the whole project was very challenging, starting from the size of the dataset to finalizing the models. Basically the project involved following things:

1. Understanding various datasets available from Kaggle website.
2. Researched on various performance metrics supported and zeroed down to F1 score.
3. Cleansing the dataset.
4. Performing feature engineering by picking up the right set of features and discarding other parameters.
5. Tried to use various classification models, especially I was focusing on multiclass classification models.

I found all the steps described above were quite challenging.

B. Improvement

One thing I would like to try to make use of all the data available in the competition, unfortunately I could use only two datasets (training and sessions) to carry out prediction. I think given some more time, I would have exercised analyzing other datasets (countries and age-gender-bkts) to perform further enhancements to the training data.