Name: Michael Hu and David Zhao **Course:** ORIE 4741

AirBnb Destination Prediction

Introduction

With the rise of Airbnb, finding places to stay during trips has never been easier, cheaper, or more centralized. We were interested in determining if the demographics, search histories, and other user data available on Airbnb could be used to accurately predict their next booking location by country. This idea originated from a kaggle competition hosted in 2014, but we intended to build onto the original drive of the competition. Possessing the ability to accurately predict where a user wants to book their next trip would allow travel companies to provide targeted ads related to the predicted location, theoretically improving click through rates. As an example, if we knew a user lived in the US and was likely intending to travel outside the US, we could give them targeted ads selling European outlet adaptors. Also, proactively knowing when surges in travelers to certain destinations will occur would give companies like Airbnb reason to incentivize more hosts to rent out their spaces in the destinations in question so there would be enough space available for everyone.

In this project, we processed a messy dataset split across multiple files to create a final numerical dataset we could train models with. There were various problems with the data, such as missing values, merging tables, imbalanced classes, and non-numerical features, that had to be addressed. After preparing the data, we tested a wide variety of feature sets and models to reach our final random forest approach with a classification accuracy of 92% on our test set.

Data

Our dataset consisted of four spreadsheets that could be joined together on certain columns, not unlike the tables of a relational database. The first csv consisted of data on 200,000 users and their booking data with fields such as age, gender, language, id, dates of various events like account creation or booking, signup channel, devices, and, of course, the destination of their booking. Although only 75,000 rows were labeled with the data we needed, this was our most abundant source of information, as it offered interesting data for describing a user both directly (demographic information) and indirectly (device choice may indicate financial status). The second table consisted of very broad user session data, covering very general actions like when a user visits their profile or clicks a search result. Although we would have liked the session data to have information such as search queries, we had to make do with a simple log of event triggers. This data, unlike the values from the first table, described user behavior such as how much time they spent planning their booking.

Unlike the first two spreadsheets, the last two do not describe users and actually could not be directly used to train our models. These final two tables consisted of geographical and demographic information on the countries we were trying to predict, and thus could not be joined with data in a test set without destinations for each user. While seemingly useless initially, we rationalized that this data may be useful for imputing missing user data such as age or gender. Specifically, there may be some correlation between the size of a country and gender, the number of males in their early 20's in the country and age, etc. Although these ideas seemed fairly unlikely, they were nonetheless worth exploring.

At this point, it shouldn't be surprising that a significant portion of our project was dedicated towards extracting features of interest from the given raw data such as the number of search

actions a user took or whether or not they made their booking on a weekend. With so much data to work with, our approach to data preprocessing was to coalesce as much data into a numerical table as possible initially, and keep or drop features during model selection. Not only did we try to make as much raw data numerical as possible, we also explored the various tables for patterns that may be worth abstracting into features. For instance, we were initially interested in whether bookings on various national holidays were important.

Data Exploration

Diving into the data, we had a lot of intuitions we wanted to confirm via visualizations or summary statistics. The most important of these was patterns involving age and gender as those were the features with the most missing values and we wanted to know as much about the data before making any imputations. For example, if we were to fill in missing ages with just the average ages of people visiting the corresponding destination, if would be important to know if the age distribution for that country was more normal or if all the ages were actually in the tails of the distribution. As Figure 1 shows, there are trends between age, gender and destinations (young women are more likely to visit France than old men) and ages are generally distributed with a longer tail. It's apparent from this visualization that using the averaging method may be reasonable for addressing missing ages.

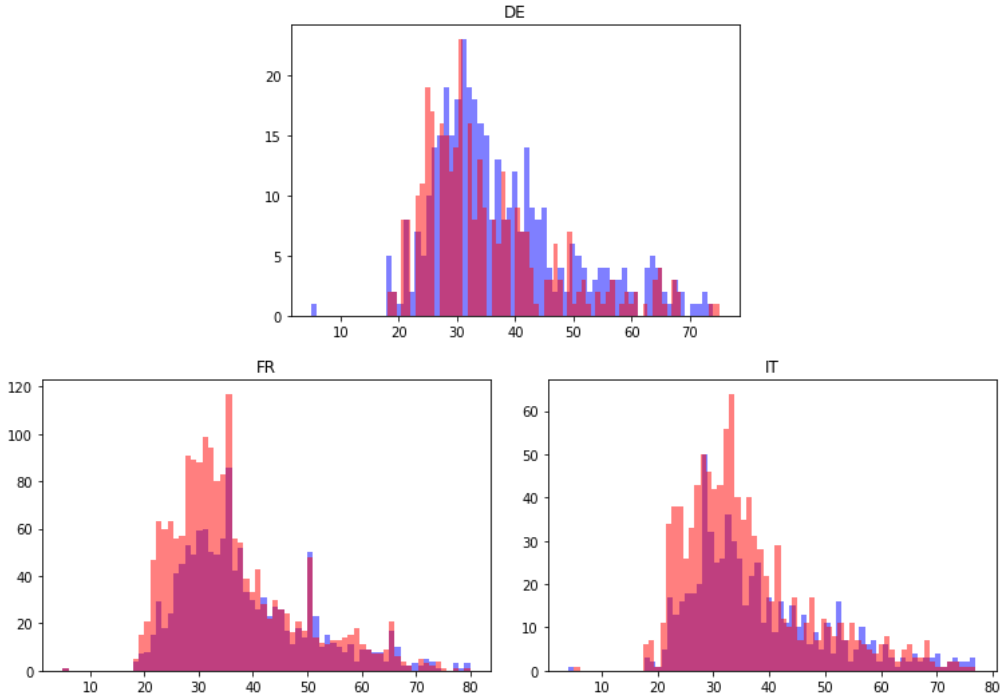

Figure 1: It appears that travelers to Germany (DE) who are older than 30 tend to be male. Travelers to France (FR) under 35 tend to be female, and travelers to Italy (IT) are mostly female. The majority of travelers appear to be 25 to 35 years in age.

As previously mentioned, one of our initial hypotheses was that dates would hold a lot of interesting patterns. Looking at summary statistics of booking dates broken up by destination, we found that, surprisingly, the most common booking days were generally nondescript. The exceptions to this in the US were Thanksgiving, New Years Day, and September 11th; The former two both had minimal bookings while the opposite was true for 9/11. In fact, 9/11 received the most bookings of any other day in 2013, primarily in US locations. Disregarding

holidays, there were also clear trends for bookings in the late summer and on weekends as Figure 2 shows. The effect on days in non-US countries was not as obvious, as there was significantly less data for many of the compared to the US; however, the fact that most bookings in France in 2013 peaked in the early summer showed that trends did vary a bit between countries.

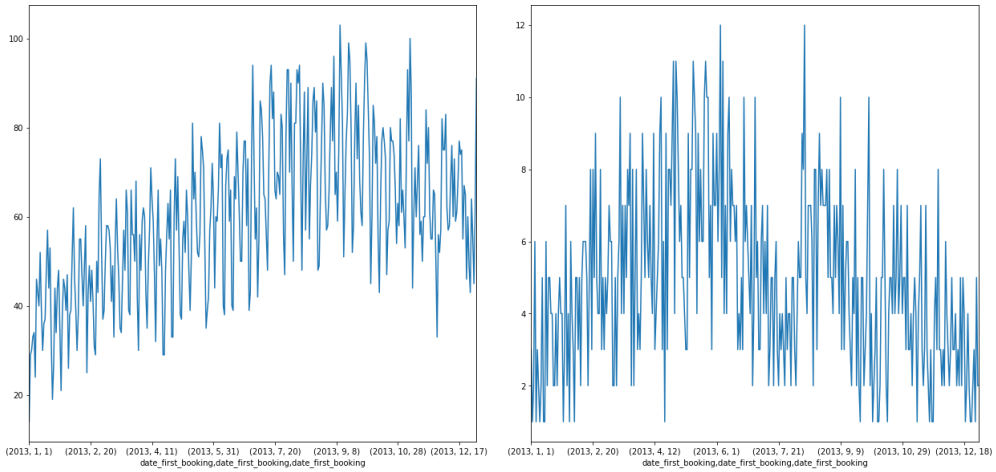

Figure 2. Booking dates in 2013 for the United States (left) and France (right)

In addition to age, gender, and time, we played with ideas for how the other user data could be used. The devices used by users were fairly evenly split, with 37k OSX users and 26k Windows sessions. There are a few iOS and Android data points as well, from mobile sessions. Looking at web session data, we saw that most users are split between Macs, Windows, and iPhones. Language had a less even split, with 97% of users in our dataset speaking English. The second and third most common languages were Chinese and French, each accounting for around .5% of users. The user session data was generally fairly dull in terms of content, as descriptions for actions such as "show page" or "search" did not offer much insight when summarized.

Data Preprocessing

To merge all of the tables we were given into a single dataset, we used the pandas python library to perform joins with various columns $userid, destination$ as keys. Then, to make our dataset numerical, we performed one-hot encodings of all the categorical features. This one-hot encoding was nice, as it provided a way to deal with missing gender data by encoding missing values as $0, 0$ where men were $0, 1$ and women were $1, 0$. To handle dates, we used pandas to convert each date string into datetime object which allowed us to break apart the dates into year, month, day, and is-weekend columns.

After combining and transforming the data, we decided to drop any one-hot encoded features that only had 1's for a few data points, as those features seemed like they would just add noise to our model instead of helping learn some concept like how people with blackberry phones generally traveled. We also had to drop any data with an undefined or "-unknown-" destination country from our final dataset, although that data could still be used for imputing missing user data. In the end, we extracted 92 features from the given data.

Imputing Missing Values

In our user data, the ages of users ranged from 2 to 2014. Assuming many of these ages were bogus answers, we decided to limit the range of our age feature to 15 to 105 and see if we could

replace fake ages with reasonably accurate predicted ones. To do this, we tried three approaches: averaging, regression, and matrix completion.

The results of averaging and linear regression to predict age were more or less the same: they only returned ages between 35 and 38. This means that our regression model was underfitting our data, as it wasn't able to find much of a pattern at all. Using the average ages per country to fill in missing data would likely be safe, as our data visualizations showed, but we wanted to see if we could do better with matrix completion.

For matrix completion, we tried using the Generalized Low Rank Models julia library from homework 6 with huber loss, non-negative regularization, and k=10 with all the user features joined with country features. The resulting prediction matrix was interesting in that it appeared to have altered the mean and variance of the age distribution as if normalization on that column had been done at some point. This would make sense because having features with disproportionate variances would mess with most dimensionality reduction algorithms. Generally the GLRM age predictions were in the ranges [2, 5] or [23, 25]. In general, predictions from the lower range corresponded to ages that were below 30, but this wasn't always the case as a 19 year old got a predicted age of 24.35, for example. This same pattern appeared even when we tried different features and parameters, and, although it was interesting, we decided to take the safer approach and stick with the regression approach to filling in missing ages.

Classification

Our approach to predicting destination countries was to try to make the best models we could with logistic regression, SVMs, and decision trees, adjusting feature sets and regularizers as needed. We first split our data into training and test sets with stratified sampling, using model selection based only on the training data and not touching the test data until the very end to evaluate our model. This was done to minimize any bias our final model would have towards the data in our test set. We then split the training set into training a validation sets and used k-fold cross validation to tune parameters and the validation set scores to compare models. These data processing acrobatics had to be done due to oversampling we performed on the training set due to a class imbalance we found in our dataset.

Dealing With Imbalanced Classes

When training our initial models, we came across a strange pattern where most or all of our predicted destinations were for the US class. As can be seen from Figure 3, simply guessing US all the time would produce an accuracy of over 79%.

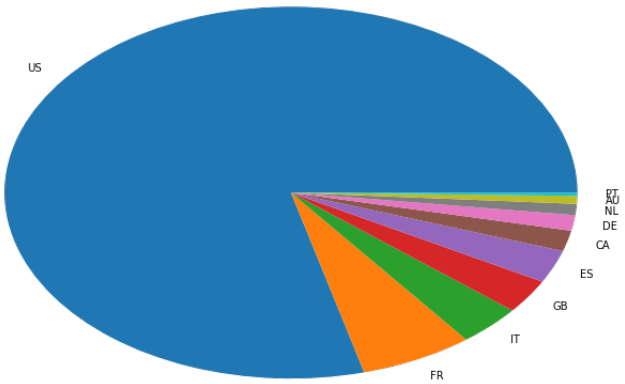

Figure 3. The portions each destination takes up of the dataset. US: 0.791, FR: 0.064, IT: 0.036, GB: 0.030, ES: 0.029, CA: 0.018, DE: 0.014, NL: 0.010, AU: 0.007, PT: 0.003

To make our models able to fit the contents of our data and not the distribution of it, we tried two approaches: oversampling non-US data, and applying weights to classes in loss functions. To see if it would make our lives any easier, we also tried solving the simpler binary classification problem of differentiating between if someone was going to book an AirBnb in the US or not.

Binary classification unless specified otherwise	Training Set	Validation Set
Logistic Regression	.797	.799
Logistic Regression + class weights	.561	.558
Logistic Regression + oversampling	.900	.900
SVM	.797	.799
SVM + class weights	.554	.551
SVM + oversampling	.900	.900
Random Forest	.992	.755
Reduced Random Forest + class weights	.991	.768
Reduced Random Forest + cw + os	.999	.907
RRF + cw, multiclass	.985	.793
RRF + cw + os, multiclass	.998	.921

Figure 4. Results from tuning various classification algorithms for predicting US vs non-US. The last two are approaches for the multiclass problem, after we decided to select random forest

As Figure 4 shows, all models appeared to perform well at first, but logistic regression and support vector machines quickly dropped off in accuracy when we started generating larger losses for non-US destinations when they were misclassified via class weights. Because random forests performed best after the application of class weights, we chose to move forward with tuning the random forests algorithm.

It is worth mentioning at this point, our unfortunate stint with the trap of improper oversampling. In Figure 4, it is apparent that all of the models that used oversampling performed extraordinarily well, and didn't even look like they are overfitting the data. Unfortunately, this was not the case. Oversampling is a technique used to artificially boost the amount of data from underrepresented classes in a dataset. The technique we used, Synthetic Minority Over-sampling Technique (SMOTE), looked at the k-nearest neighbors of each underrepresented point, and added random points between each the point and each of its neighbors to the dataset until all the classes were balanced. To perform oversampling correctly, it's important to oversample after splitting a dataset up into training and validation, which we did not do. The repercussions of this are that it's highly likely similar samples around a point will appear in both the training and validation set, causing a model to overfit without appearing to do so until it is run on a held-out set that wasn't oversampled.

Classification

The model we chose to use for this problem was random forest: a forest of decision trees, each assigned 9 random features from our set of 92 total features. This is an ensemble method that uses the results of many decision trees to form a prediction. By looking at how frequently features appeared in well performing trees, we were also able to use random forests to filter out features we didn't need and reduce the noisiness of our dataset for produce feature importance rankings like in Figure 5. Although decision trees are different from logistic regression of svms in that they don't support regularization functions, we still managed to stop our model from overfitting the data too much via limiting the depths of our trees, as discussed in lecture. Our

final random forest ended up using 33 of the original 92 features, and achieved a classification accuracy of 92.1% on our test set.

1	age
2	account created day
3	first booking day
4	first booking month
5	date account created month
6	female
7	male
8	first browser Chrome
9	secs elapsed
10	is weekend

Figure 5. Top 10 features ranked by importance

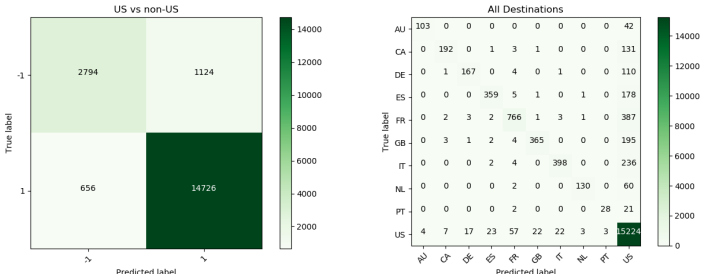

Figure 6. Confusion matrix of US vs non-US problem and also multiclass

Results

Our results indicate that we have a model that does significantly better than a baseline guess of US (a solid 79%). Looking at the confusion matrices for both binary and multi-class destinations, we see very promising results, even for countries where we had significantly less data. Although good progress was made to combat the discrepancy in class representation, our model still needs more data for non-US bookings. Looking at the multinomial confusion matrix, we see that most of the errors were between US and the next three most-visited places.

Most Important Features

As our model's performance showed much improvement after pruning some features, our model suggests that certain features are very useful for our future understanding of destination prediction. Figure 5 showed that age, time, and gender in that order were likely to be the most important features. Interestingly, aside from age, the dates that bookings were made played the largest roles, larger than even months which encode the time of year. This may imply that the time of month a trip is booked is more important than the time of year. This could be true if holidays were really important factors in trip planning, or if many business trips were planned at the beginnings of months. An unexpectedly important feature according to our model was the choice of Google Chrome as the user's browser. Looking at our data, roughly one third of the points from each class used Google Chrome to make a booking. We speculated that some combination of Google Chrome and some other feature like age may be the source of some interesting pattern.

Moving Forward

Based on the results of this project, we think that our model can be good enough for making business decisions. However, as the 8% misclassification rate would suggest, there is still much work to be done in collecting diverse data and to investigate the relationships between the various features and the destination.

References

We used the numpy, pandas, scikit-learn, and imbalanced-learn python libraries and the GeneralLowRankModels julia library.

For more information about oversampling, we recommend imbalanced-learn's guide.