AirBnB Next Booking Challenge

ISYE 7406 Data Mining and Statistical Learning

1 Abstract

AirBnB hosted a challenge for the Kaggle community to predict the country first-time AirBnB users would choose as their first booking destination- a challenge the growing data science and analytics team is constantly looking to tackle. With an increasing number of competitors and an aim to revolutionize how consumers imagine leisure travel, the company's goal was to improve its recommendation system for customers and ensure their supply of home shares met travelers' demand. Our group took this chance to explore how machine learning and predictive analysis impacts our daily lives such as in booking a vacation home. The dataset consisted of 12 variables, 11 of which were predictors, and the last variable was the country or dependent variable. We completed the same challenge by performing exploratory data analysis on the raw data, cleaning and factorizing the qualitative variables, and running several machine learning models. We evaluated the results that variable selection produced and compared the test errors after running numerous models. We ran Monte Carlo Cross-Validation on the models and found the best predictive model to be gradient boosting. This model predicted the destination of a first-time AirBnB user's first booking with an 88.51% accuracy, which would provide great benefits for the company's advertising and supply chain departments. We documented each step as discussed in this process in this report, as well as future steps we would take and the lessons we learned throughout the project.

2 Introduction

AirBnB offers stays for travelers in over 34,000 cities and in more than 190 countries. With a rapidly growing and ever-competitive travel industry bolstered by the expansion of home shares, AirBnB would benefit by knowing where a first-time booker might choose for a number of reasons. Economically, by being able to accurately predict the region where a user might want to travel, AirBnB can then determine the demand and ensure that there is ample supply to meet it. Advertising-wise, AirBnB can then suggest users individualized content based on activities and highlights from similar previous users. Lastly, by identifying which countries saw an uptick in first-time bookers, AirBnB could use that information to advertise to other first-time prospective travelers and subsequently shorten the amount of time for this group to purchase.

We want to predict the country destination column which indicates the first booking for users. To do this, we will need to perform feature selection and/or up-sampling of the training data since it seems that the target classes are heavily unbalanced. We will determine if there is any data cleaning and conversions needed amongst the variables and then model various algorithms on the data.

Given a variety of characteristics about AirBnB users, we will run various machine learning models to best predict where a new user's first stay will be. These characteristics included demographics, website session data, and other statistics to determine the country they are most likely to choose as their first destination. The response variable is made up of twelve different categories, ten of which are abbreviated country names with the last two as 'NDF' or 'no destination found', meaning there was not a booking, and 'other'.

Below are the different classification models we will use:

- Multinomal Logistic Regression (based on all features)
- Random Forest
- KNN
- AdaBoost
- LDA
- Naive Bayes

For each model, we will compute accuracies by finding the testing errors. We will use Monte Carlo Cross-Validation to represent the outcome of our modeling efforts by providing the corresponding mean values. We will then analyze which methods were most effective, and what we would suggest using to best predict which country a first-time booker might choose.

3 Problem Statement and Data Source

The data was obtained directly from the Kaggle website. The data has been split into two sets for us, one for training and one for testing. We will be using the training data to train our model then validate different models using the website on the hidden class labels. We found this dataset while looking for similar datasets on the Kaggle website, as well as other websites when researching different interesting datasets that would have real-world applications. We also discovered that this dataset was particularly of interest to us based on the potential financial advantages for determining which countries are most likely to see first-time bookers using AirBnB. Additionally, given that our group members are in different locals, with one of us in Egypt and two in the U.S., we wanted to do a research project related to traveling and international destinations, and stumbled upon this dataset.

All users in this dataset are from the USA and the ten countries are: 'US', 'FR', 'CA', 'GB', 'ES', 'IT', 'PT', 'NL', 'DE', and 'AU'. 'Other' indicates that a booking was made but is not one of the ten above countries, and is different from 'NDF', where a booking was not made.
Attribute Information:

Variable	Description
Gender	
Age	
Signup Method	Basic, Facebook, or Google
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	The first marketing user interacted with before signing up
Signup App	Web or mobile app
First Device Type	Desk or mobile phone or tablet
First Browser	Web or mobile browser
Country Destination	Target variable

4 Exploratory Data Analysis

Analyzing the data, we find some interesting insights regarding user behavior. For example, as we can see in Fig.1 when we look at the age groups of users, we find that the most engaged age group is from 40 to 54- not youth- which might indicate a preference towards family trips.

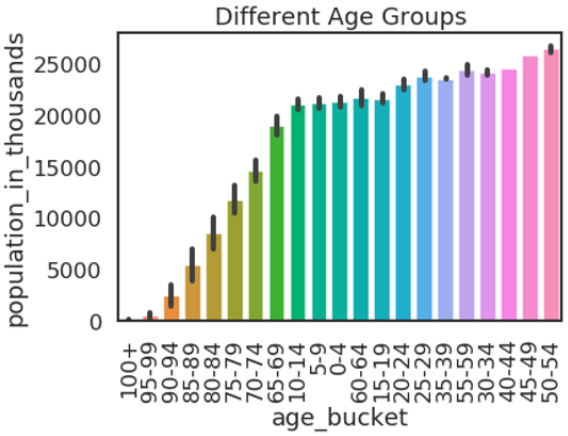

Figure 1: The most dominant age buckets is 40 to 54.

Additionally, we can see in Fig.2, there is a large increase in new accounts in the year 2014. Although there is a great increase in number of new accounts created between the years 2013 and 2014, there is a very slight increase in bookings that are actually made during the same time period, suggesting that new users didn't find what they liked on the app.

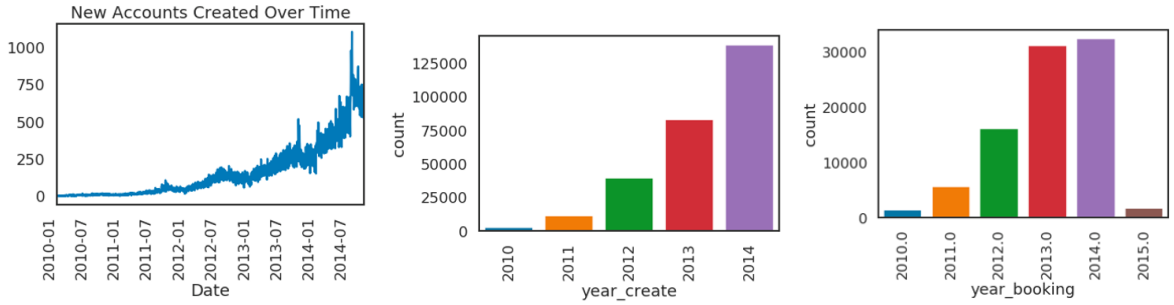

Figure 2: New accounts opening and booking.

Finally, we can see in figures 3 and 4 that most users use apple products, suggesting high welfare (because their prices are relatively high) or that the nationality of most users are American (where these devices are most popular), with the latter idea most likely because the destination is usually also in the U.S. (domestic).

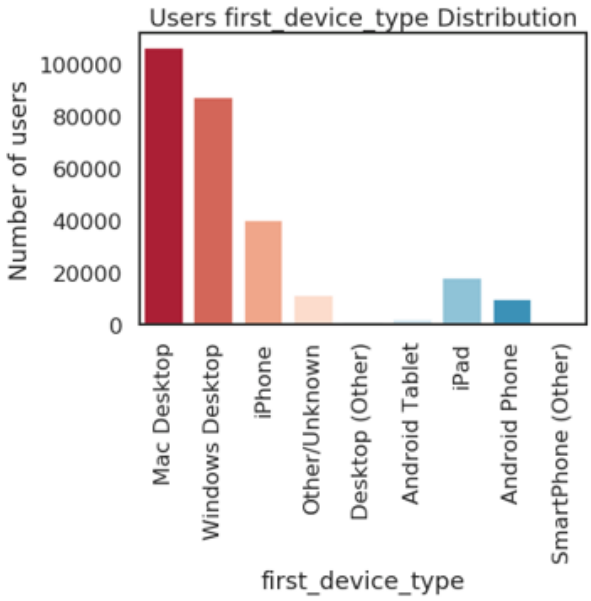

Figure 3: Users Sign up Devices.

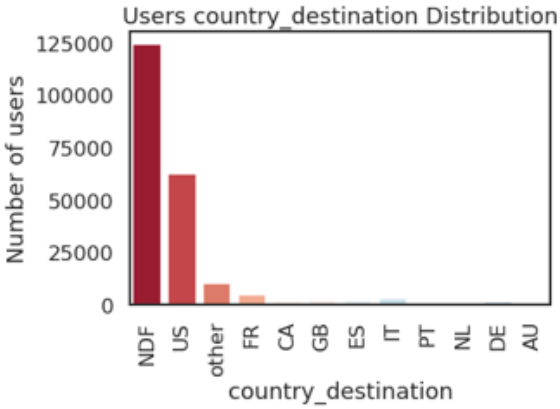

Figure 4: User Booking destination.

5 Proposed Methodology

As for methodology, we will begin by cleaning up the training dataset. We found that the dataset itself was large enough to utilize in full for this project and disregarded the testing dataset given in order to make our own with an 80/20 split. The large amount of training data given from this dataset is better than using a smaller dataset, as performance improves with more information. More data leads to lower variances during cross-validation and thus in turn provides a better predictive performance amongst the models. After cleaning up the missing and unknown values, we were left with a dataset that has 51,982 rows and 12 columns (with one being our response variable, country_destination). We then used as.factor() which converted our qualitative response variable into a factor, which created quantitative values which were easy to use for modeling. Therefore, we were left with a dataset that was labeled as follows:

Country Destination	Gender	Age	Sign Up Method	Sign Up Flow	Language	Affiliate Channel
CA	FEMALE		basic		EN	direct
DE	MALE		facebook		ES	other
ES	OTHER		google		DE	seo
FR					KO	sem-non-brand
GB					FR	content
IT					ZH	sem-brand
NL					NL	remarketing
other					IT	api
US					SV	
AU					PT	
PT					RU	
NDF					JA	
					TR	
					HU	
					NO	
					DA	
					CS	
					PL	
					TH	
					FI	
					EL	
					CA	
					IS	

Affiliate Provider	First Affiliate Tracked	Sign Up App	First Device Type	First Browser
direct	untracked a	Web	Windows Desktop	Android Browser
craiglist	omg	Android	Mac Desktop	Chrome
google	linked	Mo1	Desktop (other)	Chrome Mobile
other	tracked-other	iOS	iPhone	Chromium
facebook	marketing		iPad	Firefox
vast	product		Android Phone	Mobile Safari
bing	local ops		Android Tablet	RockMelt
padmapper			Other/Unknown	IE
yahoo			SmartPhone (other)	Safari
5-open-graph				Mobile 8
email marketing				Yandex.Browser
gsp				Sogou Explorer
naver				CoolNovo
baidu				AOL Explorer
meetup				Maxthon
yandex				Opera
				Silk
				IceWeasel
				Mobile 5
				BlackBerry Browser
				Apple Mail
				SiteKiosk
				Iron
				NetNewsWire
				Kindle Browser
				Camino
				SeaMonkey
				Pale Moon
				Stainless
				16 Mini
				TenFourFox
				Avant Browser
				TheWorld Browser

We will run a forward stepwise logistic regression using AIC/BIC for variable selection. We chose this method as it is one of the most widely used variable selection techniques. Our first model that we will test is multinomial linear regression utilizing all the factors against our response variable using the nnet package in R. This is a typical starting point and best used to classify our potential user's first booking destination using all our variables as predictors.

Next, we will run random forest using the randomForest package in R with the default parameters since the algorithm is known to effectively handle large datasets while performing optimally in predicting the classification of our response variable. This algorithm was chosen over a simple decision tree because it should perform better by combining multiple decision trees into one, and the default parameter uses 500 trees to optimally produce the best result. We will also tune this model to provide the most optimal result since it typically requires rigorous training methods. This will involve changing the number of trees, node size, as well as the number of variables to randomly sample at each of the splits.

The third algorithm ran will be KNN, using the class package in R, where we will perform hypertuning to find the most optimal values of k. Also known for its high predictive accuracy, this algorithm is one of the simplest classification algorithms because it classifies the data points based on the distance from the nearest neighbors. We will provide a separate table for the individual k cluster values and their associated testing errors for easy comparison.

Next will be the gradient boosting algorithm, using the gbm package in R, which additively builds the decision trees to find the one with the most optimal testing error and thus produce the best model.

We hypothesize that this will provide an optimal result because of its ability to correct the weaknesses of the previous trees.

Next is LDA, run using the MASS package in R, and is generally used to classify patterns between two classes and it seeks to understand data through the use of unknown data, determining the similarity between the two. LDA is useful as it often produces robust, decent, and interpretable classification results. We expect this to perform worse than MLR as although the model is used for classification, LDA is used for modeling continuous variables while we have categorical variables.

Lastly is Naive Bayes, modeled using the e1071 package in R, and used for classification tasks with categorial input values. Naive Bayes is highly scalable with the number of predictors and data points. It is fast and can be used to make real-time predictions and is not sensitive to irrelevant features. Naive Bayes should perform well as it runs under the assumption that the variables are independent and LDA runs under the assumption that the covariance matrix is identical for all the features.

Finally, we will also incorporate Monte Carlo Cross-Validation (CV) on each of these algorithms, meaning, we will be splitting the dataset randomly to best fit the appropriate model. This will typically provide better predictive accuracy because the results of the splits are averaged to better the model's ability to predict unknown, unclassified data. We will evaluate the mean and variance values given by CV and determine which method we would advise to use for classification purposes. Finally, we will then discuss any and all conclusions and challenges based on the results of the implementations and appropriate suggestions of what algorithms would be best to use in this case.

6 Analysis and Results

Model (Feature Selection)	Testing Error	Model with Cross Validation
Multinomial Linear Regression	0.2889573	0.2957445
KNN (k=3)	0.1131781	0.1224609
LDA	0.2884727	0.2962678
Naive Bayes	0.2989521	0.3058869

We ran a forward stepwise logistic regression using AIC/BIC for variable selection. Both AIC and BIC showed the most important variable to be signup_method, followed by first_affiliate_tracked, gender, and language. We ran MLR, KNN, LDA, and Naive Bayes with variable selection as random forest and boosting auto-selects variables for prediction. The table above shows our results. None performed better than boosting- the model we found to be optimal. However, running our models using only these variables did not improve performance significantly and we felt that discluding the other variables could leave out crucial predictive value. Therefore, we ran and evaluated all models without manual variable selection and discuss their results in the conclusion.

Model	Testing Error	Model with Cross Validation
Multinomial Linear Regression	0.3016055	0.2966896
Random Forest	0.3333333	0.3433333
KNN (k=3)	0.1281087	0.1854141
Boosting	0.1165995	0.1148831
LDA	0.2930046	0.2915711
Naive Bayes	0.3126084	0.3146980

Model	Testing Error	Model with Cross Validation
KNN (k=1)	0.1019857	0.1565600
KNN (k=3)	0.1281087	0.1854141
KNN (k=5)	0.1403509	0.1981461
KNN (k=7)	0.1540389	0.2053740
KNN (k=9)	0.1652207	0.2107791
KNN (k=11)	0.170908	0.2150503
KNN (k=13)	0.1791016	0.2184641
KNN (k=15)	0.1845961	0.2215207

Of the models and selections we made for modeling, the tables above demonstrate the results of our experimentation. Using these visuals, it is easy for potential business-critical teams to determine which modeling techniques would be preferable, and if CV should be used or not. The above chart shows the results of running the six modeling techniques, with and without CV and their associated testing errors. The chart below it shows the results of running various k cluster values with and without CV as well, and as the top table demonstrates, k = 3 clusters provided the best results.

7 Conclusion

Our results showed that boosting provided the most optimal testing error, and performed the best even with cross-validation. We also noticed that random forest proved to be the weakest model, both with and without the use of cross-validation. We wanted to note the inefficiency of the random forest model, and the large number of trees combined with the 100 loops for CV took several hours to run. This makes sense with respect to what we know about the two models, being that gradient boosting is typically better than random forest. Additionally, we know that boosting is known for over-fitting which is why tuning the model typically provides better results, which we also see evident here as the testing error was 11.66% without tuning and 11.49% with tuning- an improvement.

Additionally, we know that linear regression will typically also outperform random forest, which we observed in our results. In general, we saw a trend where cross-validation worsened our models' results, except in the case of boosting, LDA, and MLR. Even still, the improvement amongst these models was minuscule. This was a somewhat shocking find because we typically find that CV provides better performance results. However, this is not always the case as although CV is a good demonstration of how your model is performing, it does not prevent the model from over-fitting. The solution for over-fitting is hyperparameter tuning, which is typically done outside of CV, and combined with feature selection, could overcome the over-fitting. Moreover, we found that after both feature engineering and hyperparameter tuning, we reported the best results and noticed these slight improvements with the use of CV.

For KNN, we tuned the model to find the best number of k clusters. We noticed when k was 9 or higher, the testing error plateaued with cross-validation. Although when $k = 1$ performed the best after cross-validation, we did not choose it as the best model because its only neighbor is itself (and in theory, should have 0% test error). Therefore, we found that $k = 3$ yields the smallest testing error and performed the best among the various number of k clusters when using cross-validation. Additionally, for each test, we incremented k clusters by 2 because it is recommended to use an odd number to avoid ties in classification efforts. We then cross-validated our results to confirm the optimal k cluster value. We again saw similar results as to when we did not perform cross-validation. We found that although $k = 1$ had the lowest test error, we chose $k = 3$ to be the best because of the low testing error value and similar plateau at $k = 9$ clusters.

Overall, we found that boosting was the best method to use in order to classify a future AirBnB renter's first country of choice for booking. We can see that KNN with $k = 3$ is a close second choice and performed better without the use of cross-validation. Additionally, the other four methods produced similar results and were far outperformed by boosting and KNN. Surprisingly, MLR and LDA performed about the same. Naive Bayes may not have performed well due to the large feature list- the model may not produce accuracy because the likelihood would be distributed and may not follow the Gaussian or other distribution. This further aligns with what we already know since gradient boosting is one of the best machine learning methods out there due to its ability to tune weak models by auto-selecting the variables for prediction.

For all of the methods, we noticed that the testing errors were higher than the training errors. This means that the training errors are not indicative of accurate predictive performance which is why we omitted these values from our report. As we learned in lectures, this is because it is known that training errors always underestimate the Expected Predictive Error. To further improve upon this project, we would suggest running additional models with and without cross-validation and comparing them to see if any outperform boosting. Below, we will delve into what we learned throughout this process, the challenges we encountered, and future developments that could be made to improve predictive performance.

7.1 Lessons We Have Learned

We learned that in order to truly optimize a data model you have to understand the problem set at hand and make comparisons with other models in order to evaluate the results and give weight to the solutions. After finding a dataset we all agreed to work with, we performed exploratory data analysis on the raw data- this is key because it allowed us to evaluate the data without making assumptions. This gave us insight into the patterns and relationships between the variables, which we know is important in order to determine which relationships may be needed for variable selection. We noticed quickly that we would need to perform feature engineering by converting the qualitative variables into integer values in order to run the models. Using as.factor() was something we did not have experience with before this project and was something we found to be incredibly useful.

We initially chose this dataset because of the real-world applications it could provide which is necessary as we will be doing similar evaluations in future industry settings. This ability to apply what we are learning in courses to real-life business settings is our goal within this program and something we built upon within this assignment. We also learned how to collaboratively work on one dataset, together. In class, we focused on individually choosing and implementing modeling techniques, and this project gave us experience in doing so with a team; which is much more similar to how data modeling is managed in industry fields. We learned how to divide up the work evenly, how to manage our time, and present the important key information we discovered. We found that although more models and hyperparameter tuning may have occurred within the modeling process, it is important to only present the information that is pertinent to the project and that would be best understood and relevant to the audience.

We believe that it would have been helpful to have been provided some examples of hyperparameter tuning in the lectures as it is a crucial component in building models. It would have also benefited to learn about different approaches to validation other than MC Cross Validation to ensure our results are not biased or overfitting. While we have learned enormously in this class about how to interpret and present our findings in a report, we would have liked focusing on other means of evaluating results other than comparing test errors. Overall, the lectures could include an introductory module on the entire modeling process and how to produce a report rather than on the actual models which we have learned in the introductory class in this program.

7.2 Challenges

We encountered a few challenges in the modeling process. We determined that we needed to factorize the variables in order to use them in our experimentation, which led us to use as.factor() on the predictor variables when running the models. Feature engineering was a concept that was rather new to us, especially on a dataset of this size. In running the models on our data, we noticed that some algorithms took a lengthy amount of time to run- random forest, for example, took hours to complete when ran with CV due to running 100 loops. Additionally, determining which algorithms would work best and that we should include was something that we had to determine on our own, something we hadn't done for any of the previous homework assignments. Although our table of results is limited to five algorithms, this is not to say that is all we ran on our data. One of the other challenges we had was to simplify our results to present which would narrow down the importance of our findings. Additionally, although it is best to use a large dataset such that we did, it can also be a challenge working with the large number of rows and columns of possible variables for selection.

7.3 Future Improvements

In the future, we would continue making comparisons between different algorithms in order to find the most optimal. To do so, we would hyper-tune the algorithms and include feature selection when appropriate in order to find the best results. We would continue to run the algorithms with and without cross-validation and expand upon our table in order to easily view which solutions were the best/worst. Assuming we were presenting our findings to a non-technical audience, we would gather their feedback and implement changes accordingly based on what they thought we should tweak or add to our analysis.

8 Bibliography and Credits

1. https://www.kaggle.com/competitions/airbnb-recruiting-new-user-bookings/data?select=countries.csv.zip