AirBnB Listings: An in depth dive into the world of short-term sublets

Armandas Bartas, Alex Romanus, Braeden Norman, Gabriel Lanzaro

2021-12-07

Introduction

This project aims to investigate AirBnb listings and obtain insights into the most important features of short-term sublets. More specifically, the goal of this project is to classify the cities based on different attributes. The dataset, which was obtained from Kaggle, contains 10 cities from very distinct parts of the world: Bangkok, Cape Town, Hong Kong, Istambul, Mexico City, New York, Paris, Rio de Janeiro, Rome, and Sydney. The Airbnb data possesses 280 000 listings including information related to host info, geographical data, price, number of bedrooms, amenities, review scores, etc.

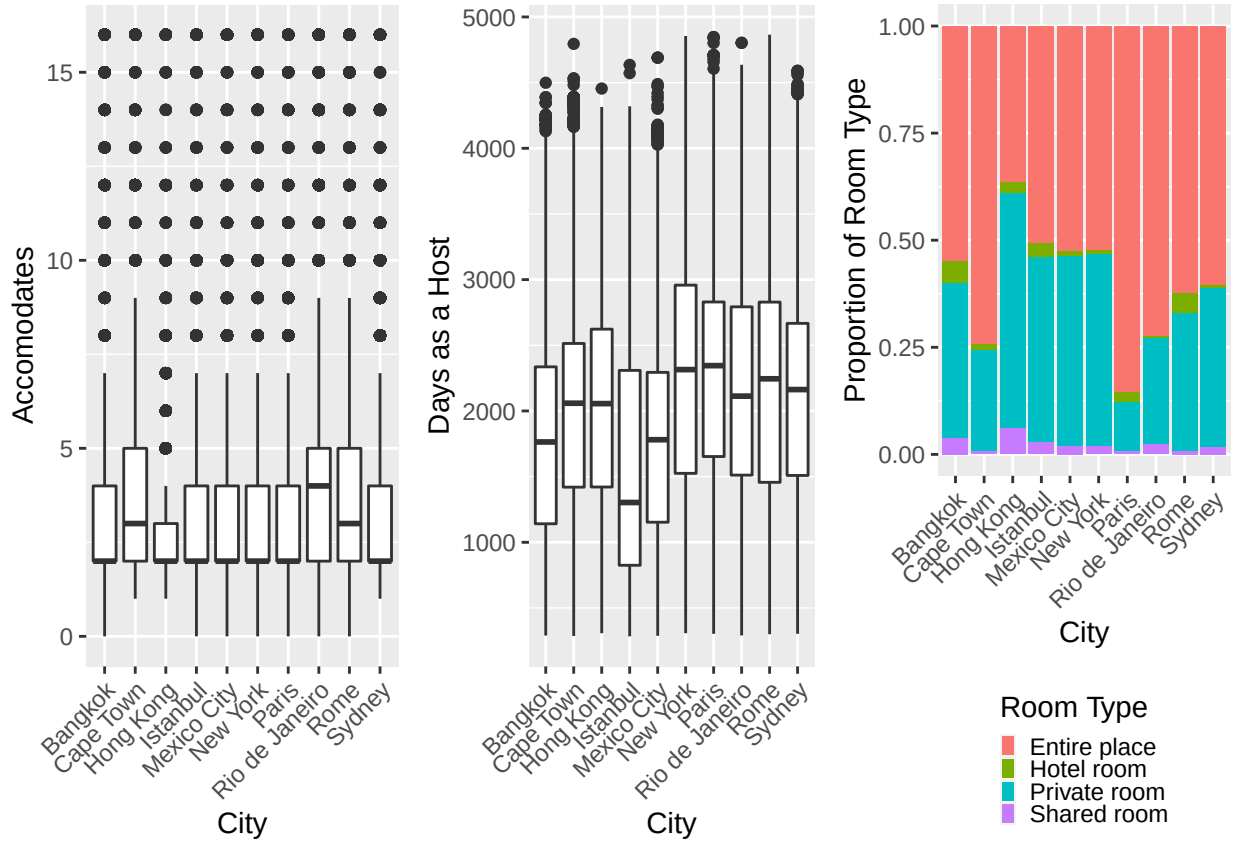
Previous research has shown that online review scores significantly contribute to selecting a place to stay (Zhao et al., 2015; Thomsen and Jeong, 2020). Moreover, the availability of locations with great review scores can influence the choice of a destination to spend a vacation. Therefore, it is crucial to develop models to further understand the destination selection. Local travel agencies can then target specific factors for improvement (e.g., reducing prices, setting mandatory amenities, demanding minimum review scores to keep hosts in the system).

The analysis can then reveal important aspects regarding how different attributes may characterize each city, for example:
- Which amenities are more important for each city when selecting a property?
- Does the host profile differ among different cities?
- Which types of accommodation are more common depending on the city?
- Can we predict the city based on different preferences related to the place to stay?
- What are the most important AirBnB variables to predict a city for destination?

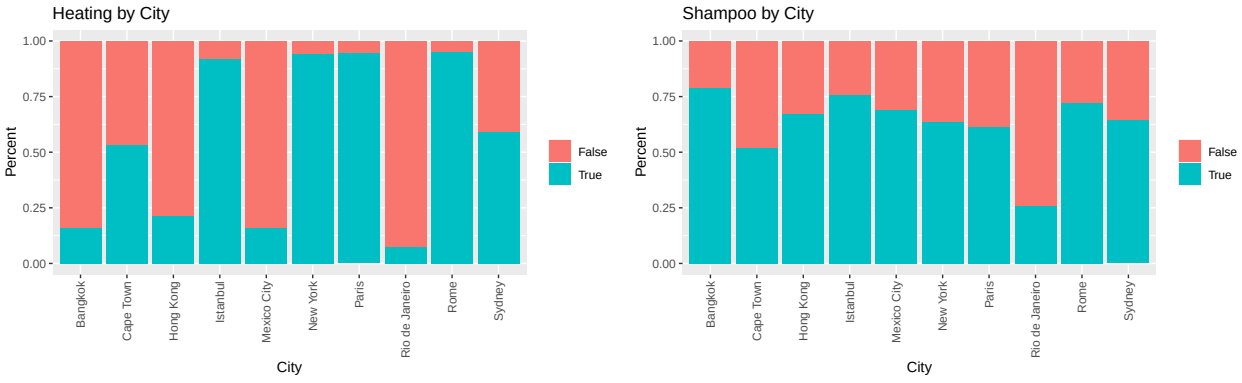
This project uses two classification algorithms to predict the cities: random forests and standard multinomial classification techniques. These methods have been selected as they provide different inferences about the data.

Exploratory Analysis

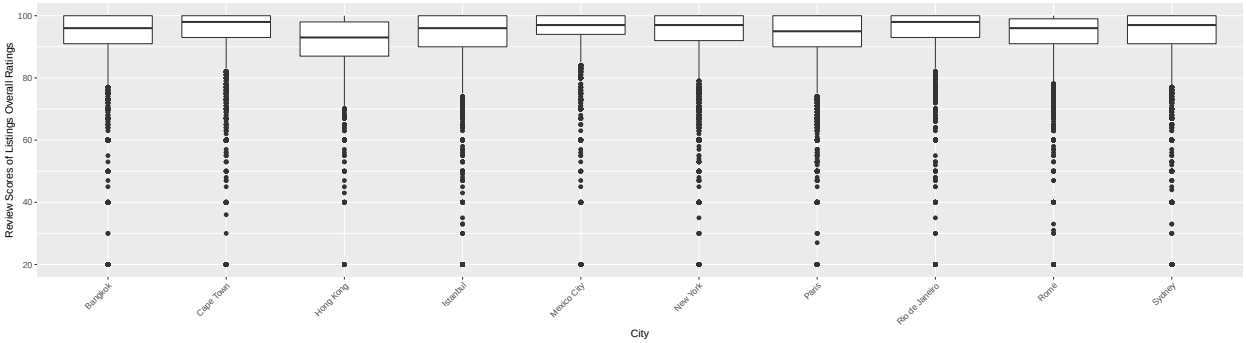
The exploratory analysis revealed some important information about the dataset. For example, the next figure shows graphs that present (1) the number of guests the listing accommodates, (2) for how long the host has been renting properties in AirBnB, and (3) the room types for different categories. The first boxplot indicates that cities such as Cape Town, Rio de Janeiro, and Rome tend to offer listings with more guests, which might be suitable for group or family trips. The second boxplot shows that AirBnb has been used in some cities for more time than in others. For example, New York and Paris have, on average, hosts with more AirBnb time than in Istambul. It might show that AirBnb has been widely used in Istambul only recently. The third plot shows that most of the accommodations in Paris and Cape Town are for the entire place, and most of the accommodations in Hong Kong are for private rooms.



In the original dataset, there is a column with a list of the possible amenities a listing has. There were 2865 different amenities and a lot of them only have a count of 1. So, to better view the distribution, we removed all amenities that had below 100 observations. The plots below represent the percent TRUE/FALSE values of 2 of the 60 included amenities in the updated dataset. For example, in Heating, we only have 4 cities that have almost no listings with heating (i.e., Bangkok, Hong Kong, Mexico City, and Rio de Janeiro). These cities tend to have the highest temperatures throughout the year.



The ratings were also evaluated. The high medians of Cape Town and Rio suggest that these cities may be easier to predict than the rest. Hong Kong also seems to have a significantly lower rating spread of ratings than the rest of the cities. Generally, it seems Airbnb customers seem to give high reviews.



The following table shows the proportion of reviews. The largest proportion of high reviews, i.e., 9s and 10s, belong to Paris at 24.99% and 26.09% respectively. Also, Istanbul has large shares of low reviews, with 23.15%, 28.57%, and 22.22% of 2s, 3s, and 4s, respectively, compared to the other cities. These heavy tails may make it easier to classify these two cities if review scores based on location are a significant predictor in the model.

city	10	2	3	4	5	6	7	8	9
Bangkok	3.98%	9.76%	0.00%	14.10%	12.50%	12.23%	18.85%	14.53%	10.84%
Cape Town	7.76%	7.87%	14.29%	4.27%	7.50%	6.35%	5.13%	5.03%	5.22%
Hong Kong	1.92%	4.57%	0.00%	5.13%	3.33%	2.23%	2.09%	2.07%	2.27%
Istanbul	5.61%	23.15%	28.57%	22.22%	23.33%	14.32%	11.90%	8.42%	5.78%
Mexico City	9.02%	11.02%	0.00%	3.42%	1.67%	5.27%	3.65%	2.75%	3.78%
New York	13.39%	10.08%	0.00%	11.54%	11.67%	13.51%	13.12%	14.71%	17.60%
Paris	26.09%	11.97%	14.29%	9.40%	10.00%	16.42%	20.76%	22.23%	24.99%
Rio de Janeiro	9.72%	7.24%	14.29%	8.55%	5.83%	7.70%	5.47%	5.87%	4.81%
Rome	9.85%	4.57%	14.29%	11.54%	17.50%	9.32%	11.29%	12.71%	15.69%
Sydney	12.64%	9.76%	14.29%	9.83%	6.67%	12.64%	7.73%	11.67%	9.01%

Results and Analysis

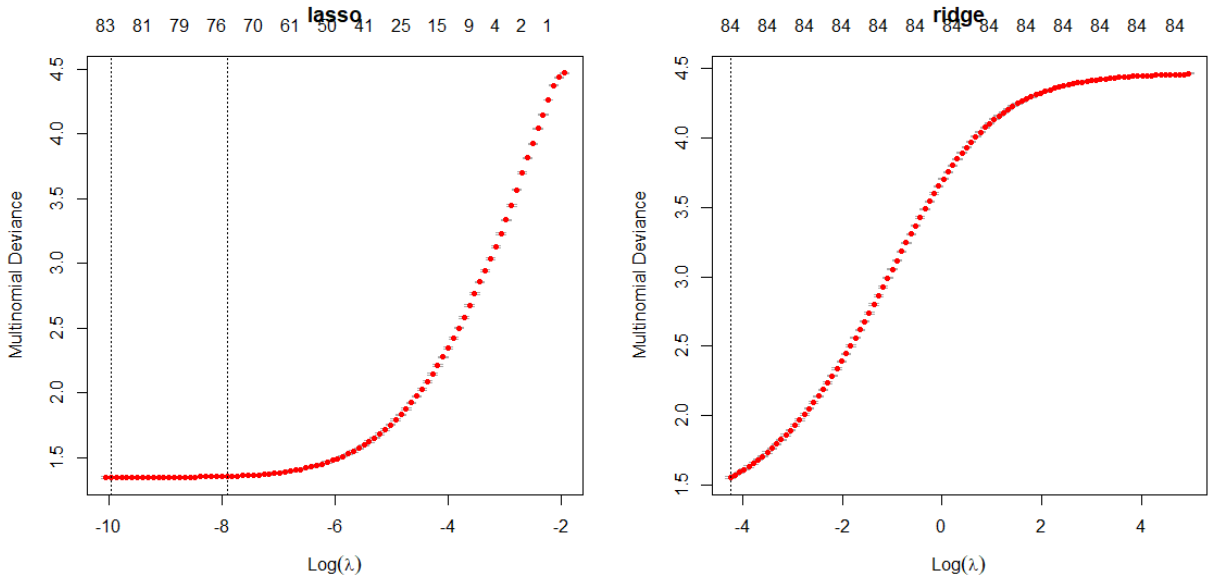
Standard Classification Techniques

This report considered classification models to determine the different cities based on the AirBnb data. Seven different classification models were applied to the dataset: (1) lasso logistic regression with default lambda, (2) lasso logistic regression with the lambda selected via cross-validation, (3) ridge logistic regression with default lambda, (4) ridge logistic regression with lambda selected via cross-validation, (5) linear discriminant analysis (LDA), (6) quadratic discriminant analysis (QDA), and (7) logistic regression.

The dataset was split into two sets: a training dataset with 80% of the data and a test dataset with 20% of the data. The errors were then computed using the test dataset. Out of the seven proposed models, the lasso logistic regression with lambda selected via cross-validation presented the best performance because of its lower error (i.e., higher accuracy).

	CV Lasso	Default Lasso	CV Ridge	Default Ridge	LDA	QDA	Logistic Regression
Error	0.2342	0.3219	0.244	0.5506	0.2619	0.31	0.532

The following plots show different values of lambda for the lasso and the ridge logistic regressions. It also shows the number of predictors that were used in each model. Lasso performs better because it uses a subset of the predictors, and some of the predictors were not useful in the final model. Therefore, the accuracy of the lasso model with fewer predictors outperformed the ridge model.



Preprocessing was also used to try to increase the model performance. The following tables shows the accuracies with different preprocessing techniques. By default, glmnet standardizes the data. Therefore, no differences were found between the accuracy reported previously and the accuracy with standardization. In addition, there were no great differences by implementing the min-max normalization.

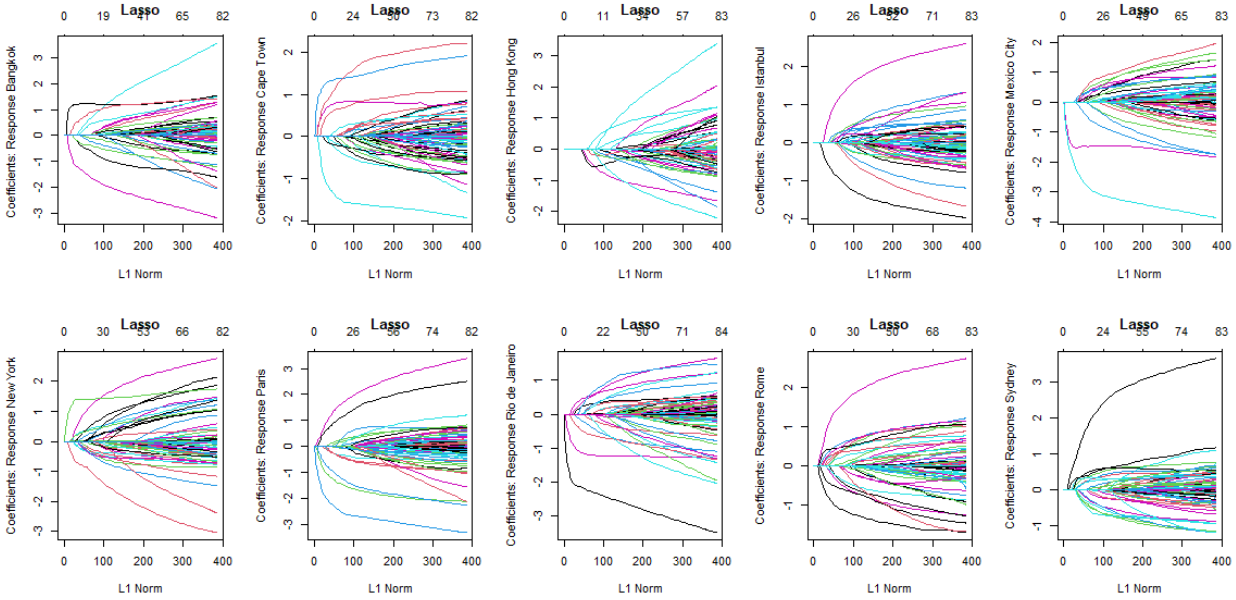
	CV Lasso	CV Lasso with Standardization	CV Lasso with Min-Max Normalization
Error	0.8566	0.2342	0.7842

The following tables show the accuracy of the model by city. The first table is a confusion matrix, and the second is the accuracy by city. Rio de Janeiro had the highest accuracy of all cities: it presented an accuracy of approximately 83%.

	BNK	CPT	HKG	IST	MXC	NYC	PAR	RIO	ROM	SYD
Bangkok	5041	84	267	117	59	21	23	366	47	144
Cape Town	57	6707	15	281	461	73	85	264	207	535
Hong Kong	170	6	1182	39	11	43	30	80	57	59
Istanbul	147	241	101	4516	198	353	398	93	787	350
Mexico City	53	735	17	204	8826	50	169	642	115	334
New York	46	139	128	561	95	10123	491	85	455	550
Paris	10	124	68	649	321	448	11624	53	639	574
Rio de Janeiro	387	342	210	120	420	64	68	9147	143	123
Rome	75	247	194	1069	160	276	505	184	9158	195
Sydney	121	632	70	346	211	425	297	81	114	5897

City	Accuracy
Bangkok	0.817
Cape Town	0.772
Hong Kong	0.705
Istanbul	0.629
Mexico City	0.792
New York	0.799
Paris	0.801
Rio de Janeiro	0.830
Rome	0.759
Sydney	0.720

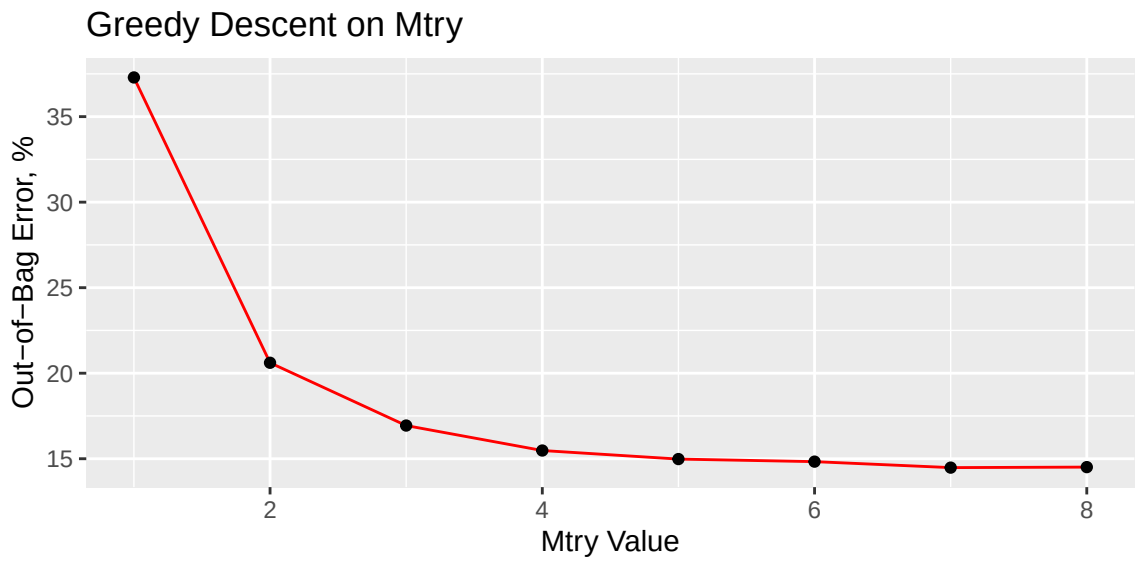
Finally, the plots below display the coefficient values for the different cities. The figures show that the coefficients may vary considerably depending on the city. For example, the coefficients for Sydney are very different from the coefficients for Cape Town. It shows the variables may exert different degrees of influence in different cities.



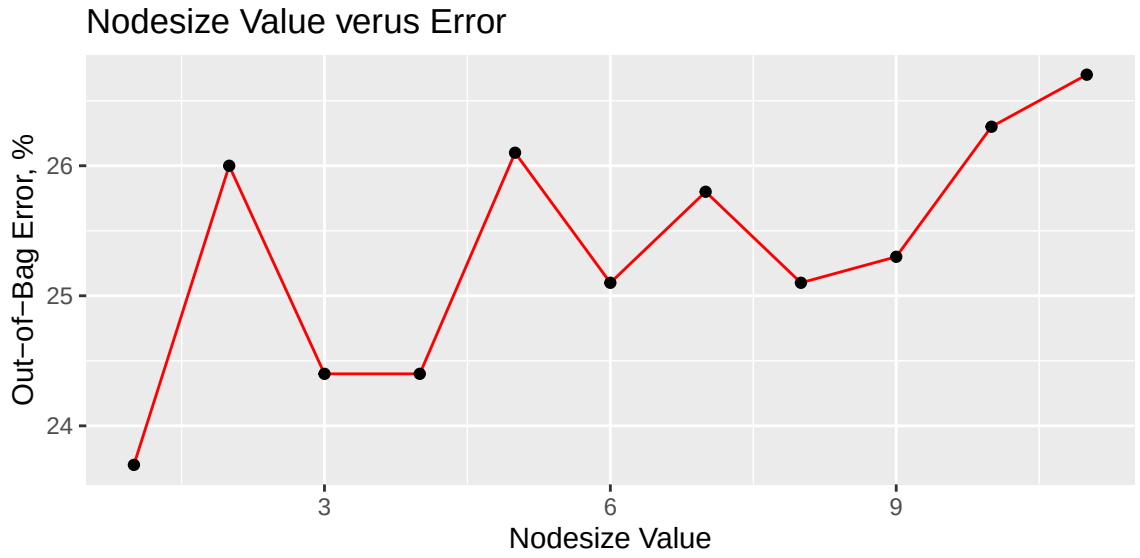
Random Forest Analysis

The most important tuning parameter with regards to random forest is mtry - the number of randomly selected predictors to use when building each individual tree. Greedy descent was performed on mtry, starting at mtry = 1, and adding +1 at each iteration until it no longer improved the out of bag error. This analysis, and the following on nodesize (i.e., the minimum size of terminal nodes), were done using a 10 000 row subset of the original data set in order to decrease computation time. Models with an mtry value of 7

performed slightly better than those with the default parameter of 9. No analysis was done on higher values of mtry.

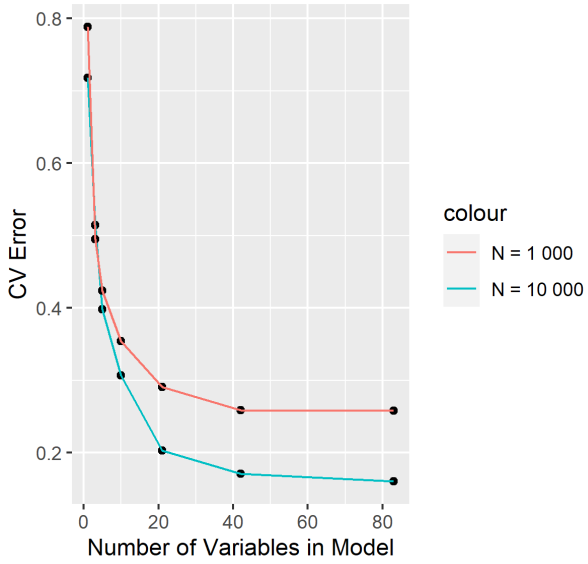


A greedy descent was also done on the maxnodes parameter - the maximum number of terminal nodes in each tree. However, larger values or maxnodes took too long to compute and the analysis was abandoned. Nodesize - the minimum size of terminal nodes, was found to be a better tuning parameter to modify. Trying different values for nodesize revealed that the default value for classification, 1, gave the best out-of-bag error.



Class weights were added in an attempt to improve the model. The hypothesis was that weighing the harder-to-predict classes more heavily would improve the classification error. However, it did occur. For completeness, those calsses were weighted more lightly compared to the easier-to-predict classes. This did not have any substantial effect on the error either.

Variable selection was evaluated using the rfcv() function with a reduced dataset. It was found that eliminating variables increased the out-of-bag error. It is worth noting that these reduced models would theoretically have lower variance but increased bias. It is also worth noting that the increase in error resulting from variable reduction was larger when fitted on a larger dataset. The reduced model with 42 predictor variables and the full model with 83 variables were then fitted on the full dataset (complete cases only).



Discussions

This report used different classification techniques to predict the cities based on an Airbnb dataset. 10 different cities from very distinct parts of the world were used. The explanatory analysis revealed that many variables (e.g., amenities, review scores, number of days as a host) influenced the cities. These variables were used to fit the classification models using traditional multinomial logistic models and random forest models.

After all the analysis, two random forest models were fitted with mtry set to 7. The first model was fitted over all variables and gave an out-of-bag classification error of 10.01%. The second model was fitted over the 42 variables marked most important by the random forest algorithm in the first model. It achieved an out-of-bag classification error of 11.03%. The easiest class to classify was Bangkok. The most difficult were Hong Kong and Istanbul, which often were mistaken for each-other. Sidney was also misclassified often as either New York or Paris. Overall, the random forest model performed very well and with minimal adjustments from the default parameters.

By comparing the random forest results to the traditional classification techniques (e.g., logistic regression), it can be seen that random forests presented better performance. In addition, random forests can provide additional and clearer inferences using the variance importance plot. Future works can include deep learning techniques as an attempt to further improve the performance of the model.

References

Thomsen, Chuhan Renee, and Miyoung Jeong. "An analysis of Airbnb online reviews: user experience in 16 US cities." Journal of Hospitality and Tourism Technology (2020).

Zhao, Xinyuan Roy, et al. "The influence of online reviews to online hotel booking intentions." International Journal of Contemporary Hospitality Management (2015).