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

Armandas Bartas, Alex Romanus, Braeden Norman, Gabriel Lanzaro

2021-12-07

Motivation

This dataset is interesting to us because it combines our love for statistics with our love for vacation planning. Statistical analysis of this data will provide insights while comparing prices and booking accommodations.

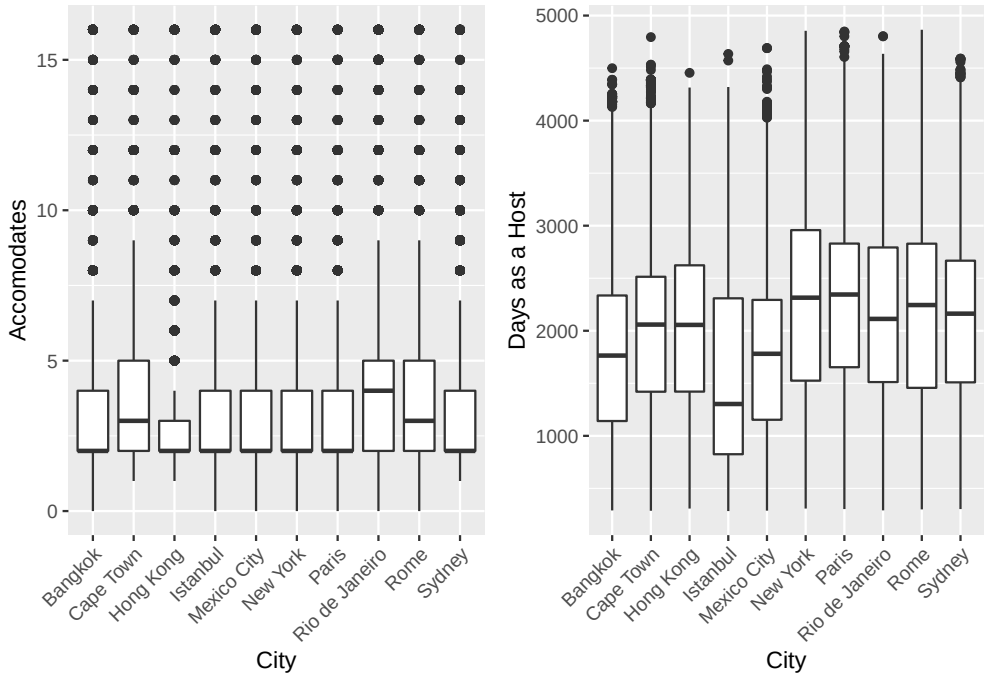
Introduction

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 contains 280 000 listings including, but not limited to: host info, geographical data, price, number of bedrooms, amenities, and review scores.

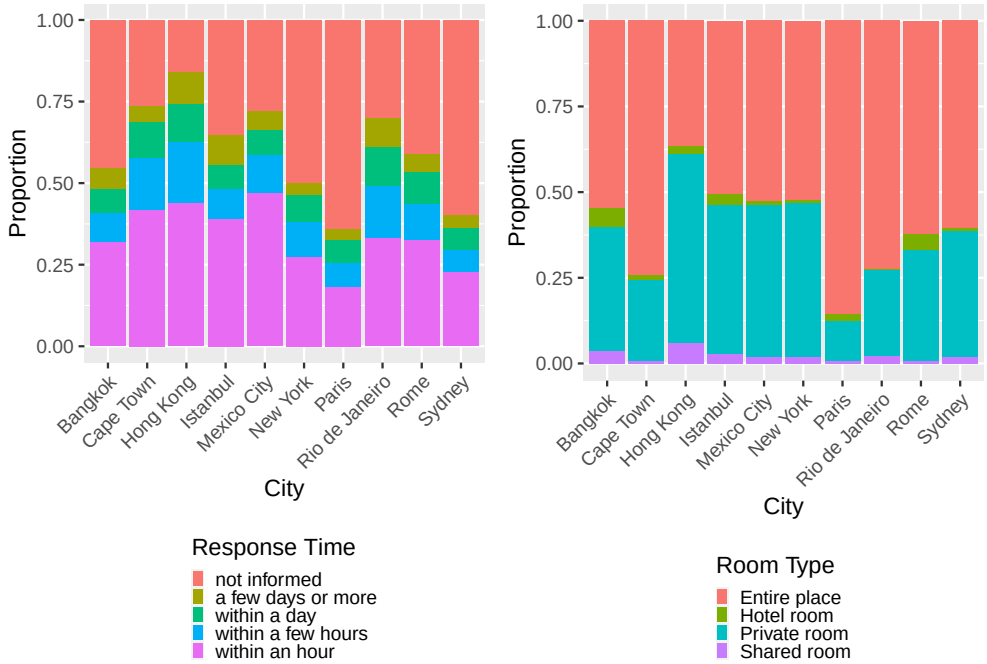
The analysis can then reveal important aspects regarding how different attributes may characterize each city, for example:
- Which amenities are more important for each city when selecting a property?
- Does the host profile differ among different cities?
- Which types of accommodation are more common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

Exploratory Analysis

Several insights can be obtained by plotting different variables against the cities. For example, the next figure shows boxplots that present (1) the number of guests the listing accommodates and (2) for how long the host has been renting properties in AirBnB. 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. For Hong Kong, however, the accommodations tend to be for fewer guests, which shows that listings might be tiny and that the city is more appropriate for business trips. In addition, 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 an average for the number of days as a host variable that is considerably higher than the average for Istambul. It might show that AirBnB has only been widely used in Istambul for a shorter amount of time.



The next plot shows the proportion per city of (1) response time and (2) room type for different categories. The first plot shows that hosts in Mexico City and Hong Long tend to have the highest response times, whereas hosts in Paris and New York have the lowest response times. The second 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. This room type analysis for Hong Kong is consistent with the previous plots (i.e., the number of guests a property can accommodate). The room type variable can also provide information regarding the trip purpose (e.g., business, family, group). Cities such as Paris are preferred for group trips, whereas Hong Kong is more appropriate for business trips.



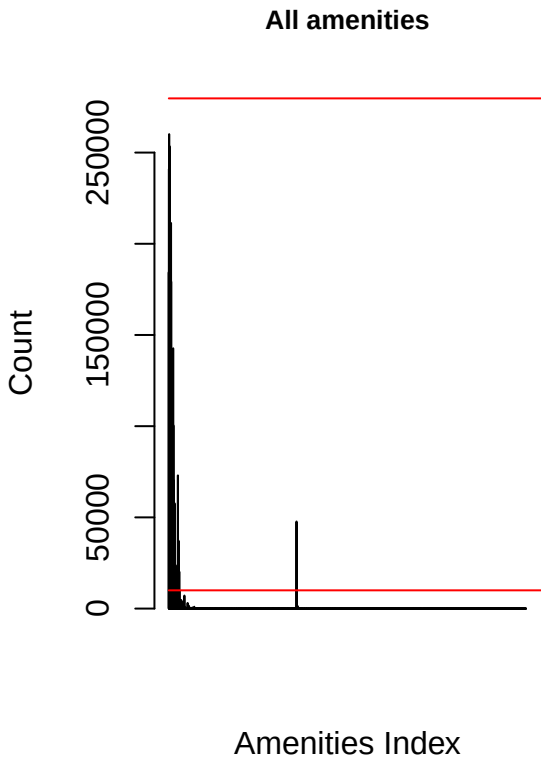
Exploring Amenities

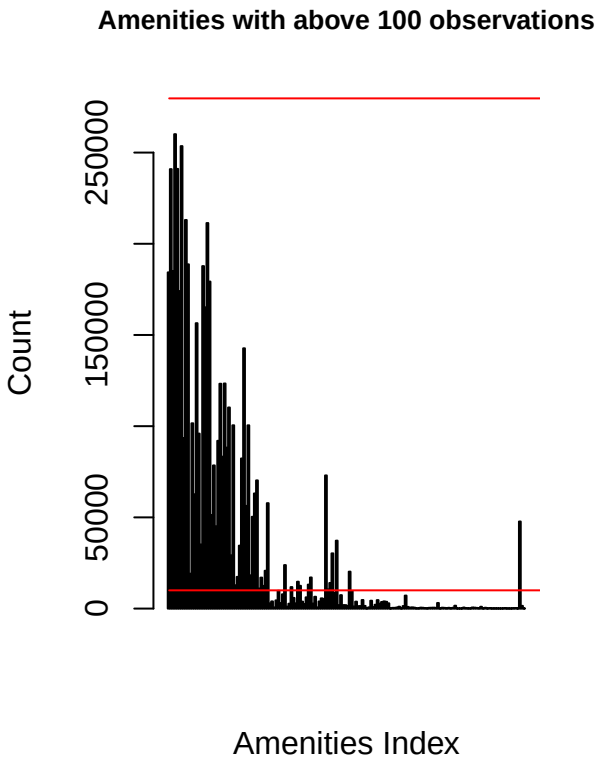
In the original dataset, there is a column with a list of string of possible amenities a listing has. Here we explored these lists to extract useful information for our predictions. For example, these are the first 5 observations:

[“Heating”, “Kitchen”, “Washer”, “Wifi”, “Long term stays allowed”], [“Shampoo”, “Heating”, “Kitchen”, “Essentials”, “Washer”, “Dryer”, “Wifi”, “Long term stays allowed”], [“Heating”, “TV”, “Kitchen”, “Washer”, “Wifi”, “Long term stays allowed”], [“Heating”, “TV”, “Kitchen”, “Wifi”, “Long term stays allowed”], [“Heating”, “TV”, “Kitchen”, “Essentials”, “Hair dryer”, “Washer”, “Dryer”, “Bathtub”, “Wifi”, “Elevator”, “Long term stays allowed”, “Cable TV”]

We found the list of all amenities and graphically determined which to include in the updated dataset. For each included amenity, we added new columns labeling whether this listing has these amenities or not.

The first graph below shows the count of all amenities. There are 2865 different amenities in the dataset. 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 second graph gives all amenities with a count over 100. The top red line is at the total number of observations in the dataset, and the bottom line is at 10,000. 10k was decided to be a good number to remove all amenities with less observations. This would leave us with a more reasonable size of amenities to add to our dataset as new columns (60 new columns after reduction).





All observations were updated with new columns: 1 for has amenity and 2 for not. Here is a quick preview of what the updated dataset looks like.

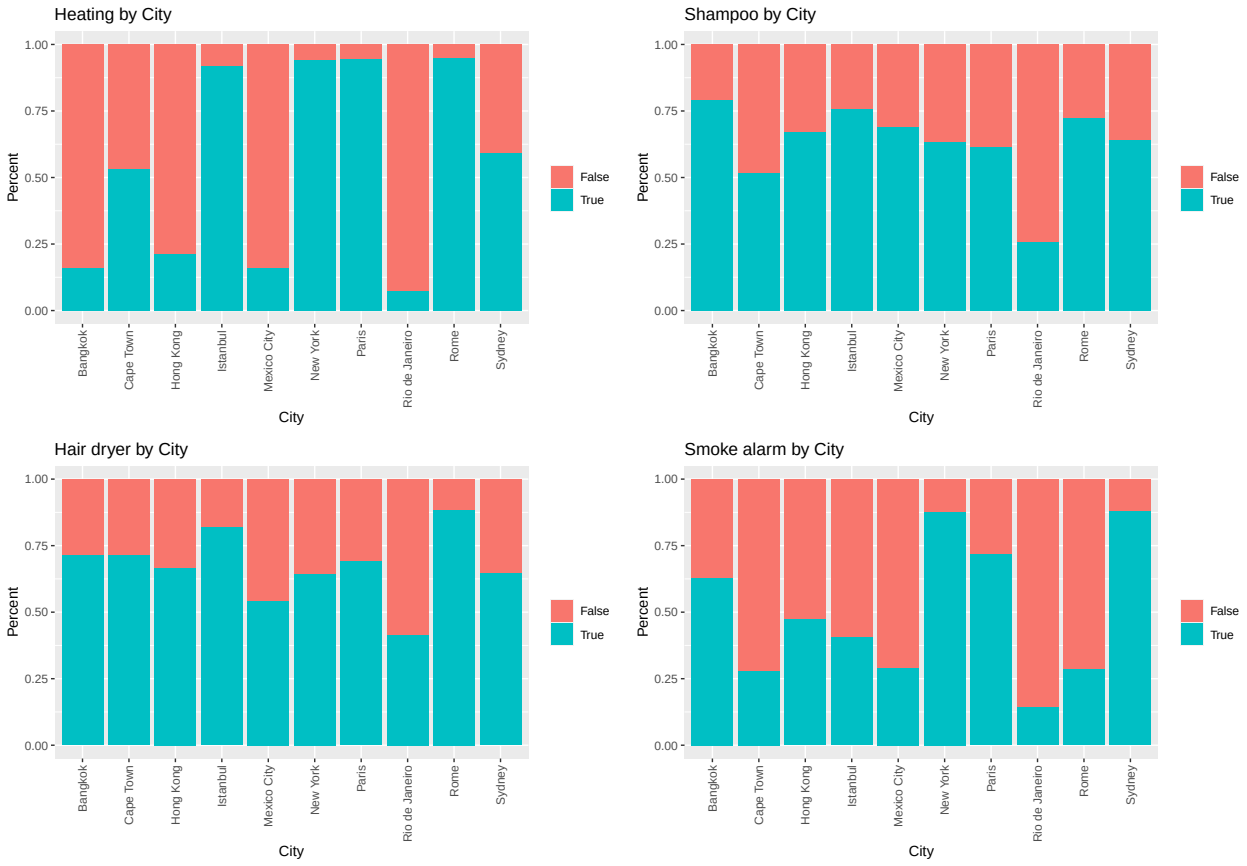
```
##   listing_id  host_id  city Heating Kitchen Washer Wifi
## 1     281420  1466919 Paris       1       1      1    1
```

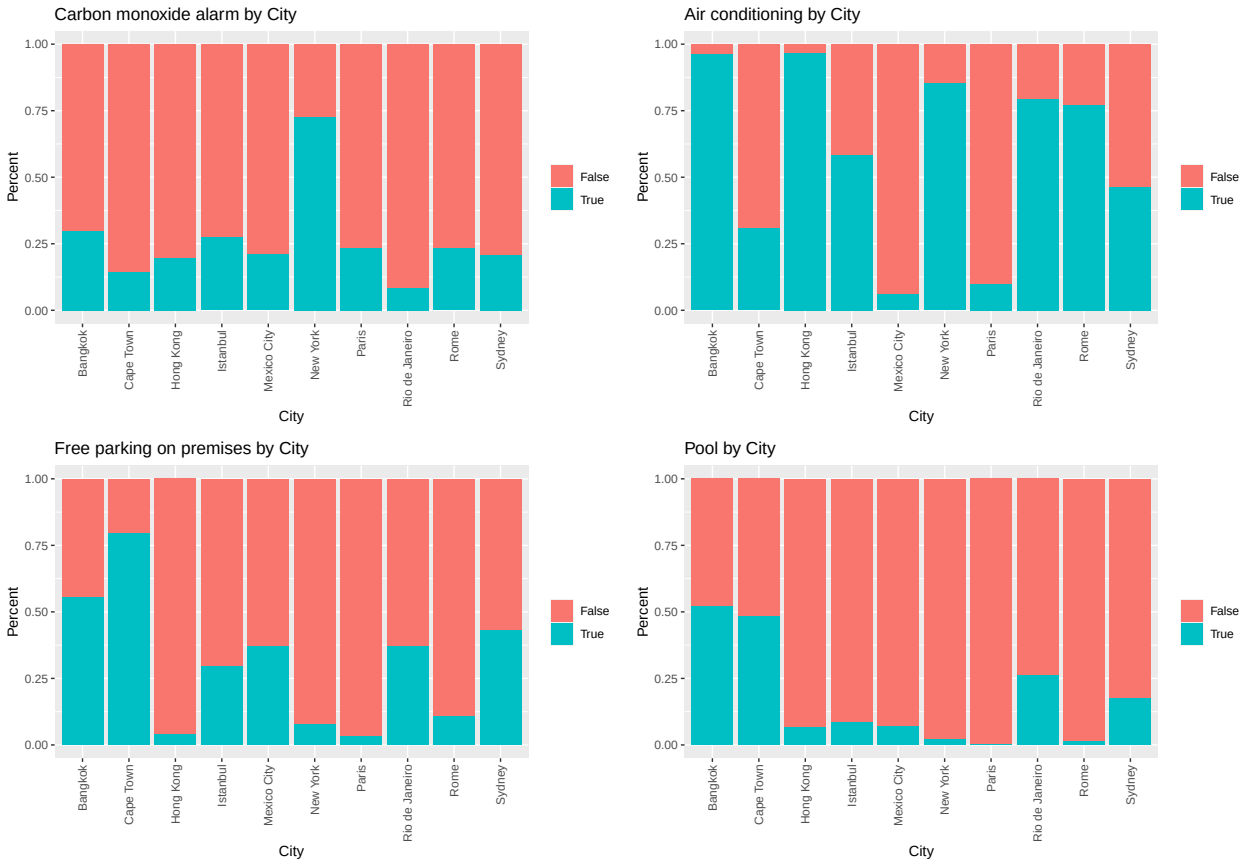
```
## 2    3705183 10328771 Paris       1       1       1    1
## 3    4082273 19252768 Paris       1       1       1    1
## 4    4797344 10668311 Paris       1       1       0    1
## 5    4823489 24837558 Paris       1       1       1    1
```

Below are the percent TRUE/FALSE values of the selected amenities for each city. Since we are trying to determine the city, based on the listings, seeing the different amenities by city will give us an understanding of how useful these amenities will be in our model. The graphs selected were:

Heating, Shampoo, Hair dryer, Smoke alarm, Carbon monoxide alarm, Air conditioning, Free parking on premises, Pool

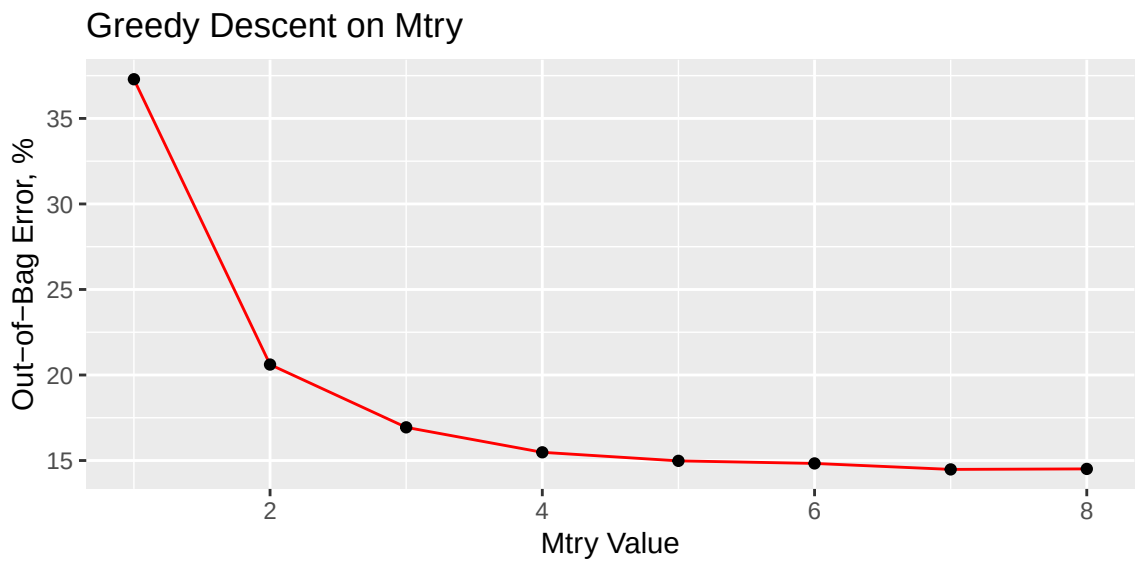
For example, in Heating, we only have 4 cities that have almost no listings with heating (Bangkok, Hong Kong, Mexico City, and Rio de Janeiro) If we also look at Air conditioning, we see that only 2, maybe, 3 have little to no listings with AC (Cape Town, Mexico City, and Paris). Now, if we take these two into account, we can see that given no AC and no Heating, we can be almost positive the listing is Mexico City.



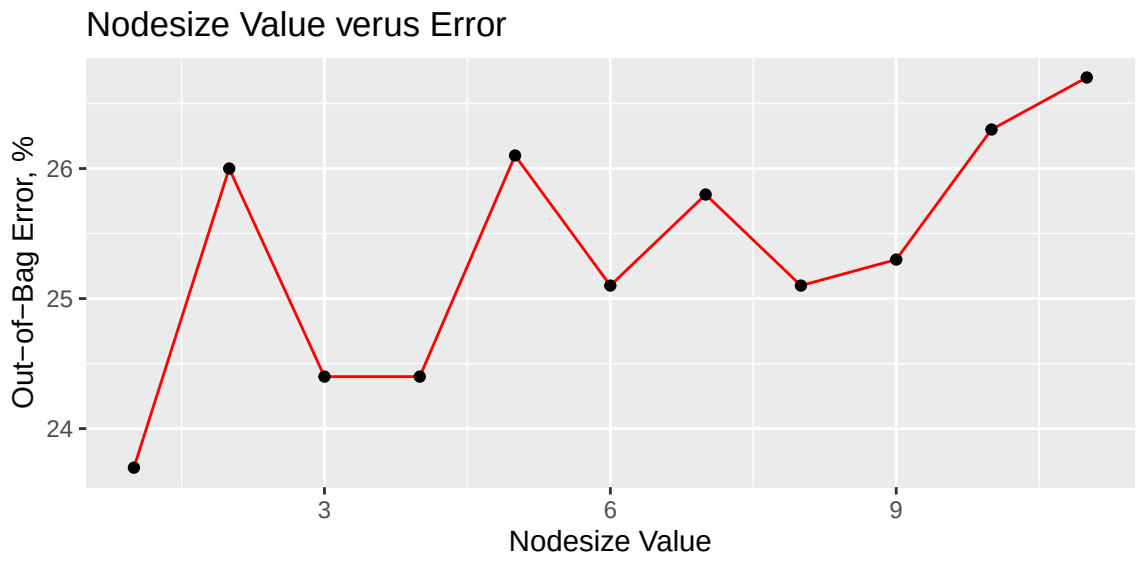


Random Forest

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 doing so no longer improved the out of bag error. This analysis, and the following on nodesize, 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.

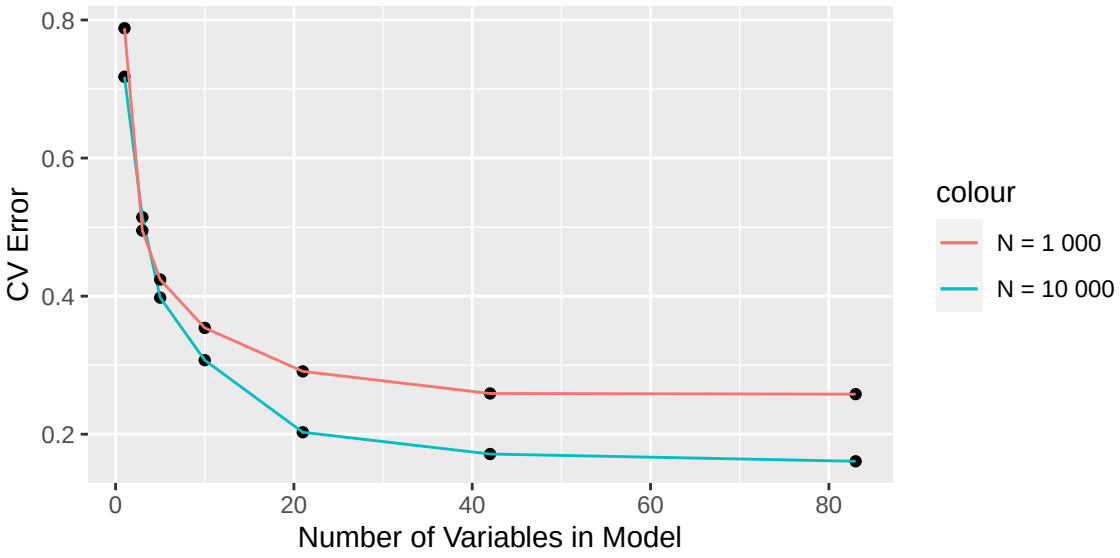


A greedy descent was also done on the maxnodes parameter - the maximum number of terminal nodes in each tree. However, larger values or maxnodes took too long to compute and analysis was abandoned. Nodesize - the minimum size of terminal nodes, was found to be a better tuning parameter to modify. Trying 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. My hypothesis was that weighing the harder-to-predict classes more heavily would improve the classification error - it did not. For completeness, I tried weighing those classes more lightly compared to the easier-to-predict classes. This also did not have any substantial effect on the error.

Variable selection was evaluated using the rfcv() function with a reduced data set. 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 elimination was larger when fitted on a larger data set. The reduced model with 42 predictor variables, and the full model with 83 variables, were then fitted on the full data set (complete cases only).



Results

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 ##%. 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.