Izabela Litwin, Yi Sophia Wen

Data Mining Final Project: Airbnb Predictions

3rd May 2019

INTRODUCTION

In this report, we are interested in understanding the AirBnB Seattle dataset better through the lens of classification, regression and clustering algorithms. We want to predict AirBnB Seattle listings prices with the lowest error in order to help people price new listings. Also, each AirBnB listing also has a review score or undesirable or desirable, and our goal is to design a classifier for this variable that has the highest accuracy on new data. The data is sourced from the Airbnb site.

EXPLORATION

Data Exploration

We are working on 2 datasets: Price and Review.

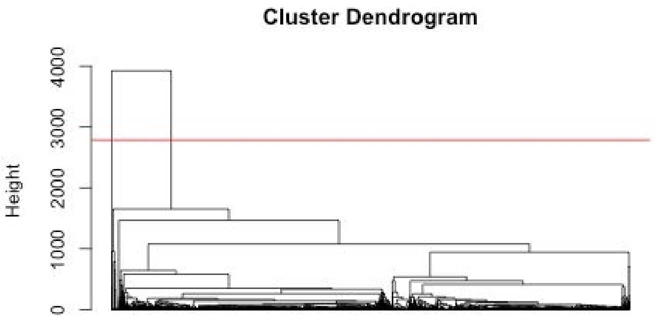
Price dataset has 5200 listings and 24 variables including listing's id, whether host is a superhost, whether their identity is verified, the host's response rate, number of host's listings, listing's neighborhood group, latitude, longitude, room type, property type, guests included, accommodates, number of bathrooms, bedrooms, beds, bed type, amenities, cleaning fee, number of minimum nights, number of maximum nights, whether the listing is instant bookble, the cancelation policy and the price. Listing's price is our response variable.

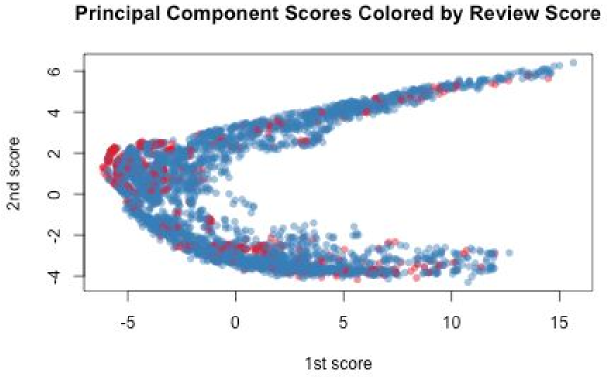
Review dataset has 4043 listings and 24 variables that are the same as those in price. Additionally, it has a review score variable discretized into two classes {0, 1} corresponding to {undesirable, desirable}. Review score is our response variable.

To explore the data, we decided to take the following unsupervised approaches on the continuous variables in our datasets:

- Hierarchical clustering with average linkage
- K-means clustering (k=2 from a scree plot)
- Principal component analysis (with identified an elbow at 2)

The most interesting structure that we detected comes from hierarchical clustering. In particular, when we cut the dendrogram so that there are 2 clusters and assign 2 different labels to them, the review scores predicted by the clustering have the misclassification error of only 14.37%. For PCA, we created a scree plot, identified an elbow at 2 and plotted the principal components. We were not able to find any meaningful structure in the PCA analysis.


Cluster Dendrogram


Principal Component Scores Colored by Review Score

Data Pre-Processing

Geographical Coordinates

We turned the longitude-latitude features of Airbnb listings in Seattle into Euclidean distances to 30 landmarks and large tour sites. The 30 historic reservations and tour sites are selected based on their popularity according to Tripadvisor and City Pass and on their size according to our intuition. The 30 chosen landmarks are as follows:

> *Space Needle, Hiram Chittenden Locks, Columbia Center, Seattle Center, Amazon Spheres, Mt. Rainer National Park, Pioneer Square, Geocaching HQ, University District, Queen Anne Book Company, Daybreak Star Cultural Center, Dunn Gardens, Pacific Science Center, Museum of Pop Culture, Seattle Aquarium, Argosy Cruises, Chihuly Garden and Glass, Woodland Park and Zoo, Pike Place Market, Washington Park Arboretum, Safeco Field, Century Link Field, Olympic Sculpture Park, Myrtle Edwards Park, Kerry Park, Lake Union, Paramount Theatre, Seattle Public Library, Elliot Bay Book Company, Seattle Great Wheel*

To get the distances in kilometers, we used the function ***spDistsN1*** under the R package ***np***.

Amenities

The Amenities variable in both datasets is a series of quotes enclosing multi-word amenities. In order to extract the predictive power of this variable, we stripped those quotes and converted them to categorical variables corresponding to unique amenities. For Price, there were 181 different amenities. For Review, there were 183 different amenities. We decided to evaluate the importance of those categories and limit amenities to only most important variables.

With Price dataset, we run a linear regression of price on all 181 amenities/categorical variables and ordered the variables by the t value (or equivalently p value). We chose only 30 most significant variables (from most significant to least significant):

> *Pool, Pack n Play travel crib, indoor fireplace, gym, hot tub, extra pillows and blankets, cable tv, internet, luggage drop off allowed, suitable for events, bed linens, tv, wheelchair accessible, dishwasher, buzzer wireless intercom, host greets you, paid parking on premises, oven, crib, essentials, family kid friendly, private entrance, fire extinguisher, single level home, pets live on this property, coffeemaker, free street parking, paid parking off premises, elevator, standing valet*

With Review dataset, we run a logistic regression of review score on all 183 amenities/categorical variables and ordered the variables by the z score (or equivalently p value). We chose only 30 most significant variables (from most significant to least significant):

> *Air Conditioning, luggage drop off allowed, tv, extra pillows and blankets, building staff, smartlock, dishes and silverware, carbon monoxide detector, host greets you, translation missing en hosting amenity 50, shower chair, kitchen, breakfast, pool, ethernet connection, wifi, stair gates, hot tub, lake access, lockbox, cleaning before checkout, bed linens, table corner guards, toilet, private living room, long term stays allowed, accessible height toilet, full kitchen, wide clearance to bed, beach essentials*

Other Variables

In our data cleansing process, we also dropped the ID variables as they do not add any predictive power to our models. Also, we dropped the “$” and “%” signs to make our analysis easier.

SUPERVISED ANALYSIS

In order to get the best predictive performance in both Price and Review datasets, we decided to run a wide selection of classification and regression models and compare their errors. We performed a 10 fold cross validation on each model we run. We tried to understand the base rates (fitted naive predictors) in order to compare our classification model accuracies to a benchmark (Naive Bayes). We diagnosed the bias/variance tradeoff problems, plotted model complexity curves, regularization curves and compared various models. We tuned each model with a set of several tuning parameters.

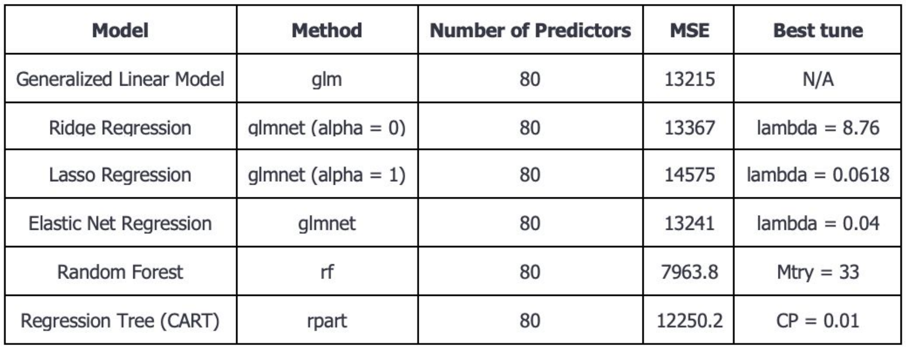
Regression

We run 6 different regression models with 80 predictor variables (20 initial variables without ID variables, initial amenities variable and geographical variables, 30 amenity categorical variables and 30 variables with distances to different landmarks). We engineered the amenities and landmark features as described in the data exploration section.

We ran the following 6 regression algorithms: Generalized Linear Regression (GLM), Ridge, Lasso, Elastic Net, Random Forest, and Regression Tree (CART). For Ridge, Lasso and Elastic Net, we used the function ***train*** under R package ***caret***, and tuned the parameters (lambda) for these algorithms with 10-fold cross validation.

For Random Forest, we firstly tuned the forest with 25, 35, 45 and 60 variables to be split at each node (also known as mtry value) on a 10-fold cross validation. We found out that the RMSE value drops when the mtry value increases from 25 to 35, then the RMSE increases when mtry value increases from 35 to 60. After that, we tuned the forest with 25, 27, 29, 31, 33, 35 mtry values on a 10-fold cross validation to find the number of trees that offer the least mean-squared error. For Regression Tree (CART), we used function ***rpart*** under R package ***rpart*** and to find the best Complexity Parameter (CP) that offers the smallest test error.

With the set of predictor variables we ran 6 regression algorithms. We tuned parameters for all of the regression models with 10-fold cross validation. We used function ***train*** under R package ***caret*** to tune the parameter lambda and to train models. The Mean-Squared-Errors and the final tuning parameters are listed in the table below:

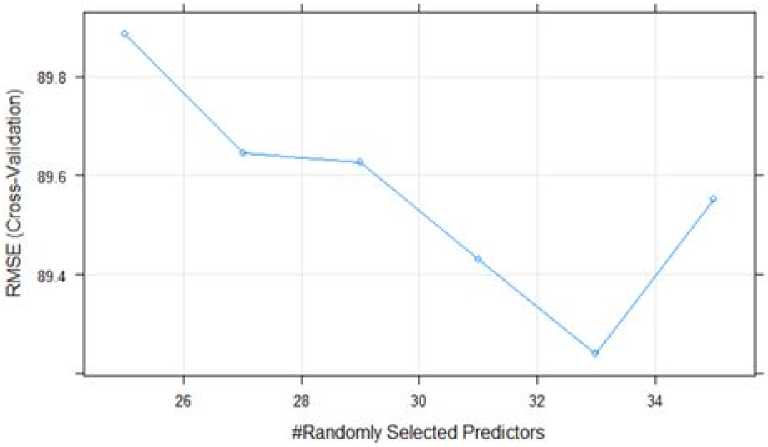


Model	Method	Number of Predictors	MSE	Best tune
Generalized Linear Model	glm	80	13215	N/A
Ridge Regression	glmnet (alpha = 0)	80	13367	lambda = 8.76
Lasso Regression	glmnet (alpha = 1)	80	14575	lambda = 0.0618
Elastic Net Regression	glmnet	80	13241	lambda = 0.04
Random Forest	rf	80	7963.8	Mtry = 33
Regression Tree (CART)	rpart	80	12250.2	CP = 0.01

According to the table above, the random forest model with max. 33 variables split at each node (mtry value) and 80 predictors has the least mean-squared cross validation error on the train set with 10-fold cross validation. The best cross validation error we got is around:

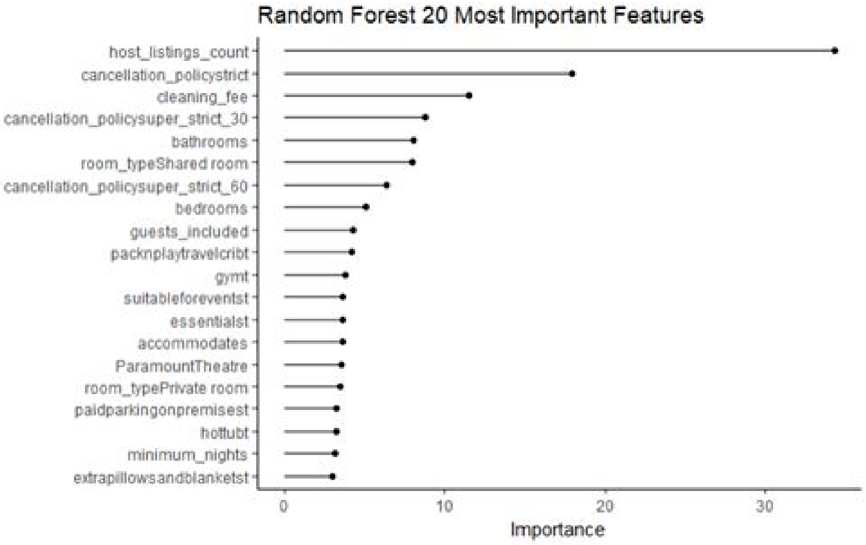
$$MSE = RMSE^2 = 89.24^2 = 7963.82$$

The mtry versus corresponding RMSE that we obtained from the random forest model is visualized as below.



To explain the nature of the relationship between the predictors in the random forest and the predictions itself, we created a variable importance plot measured by the random forest. The importance of features was computed using ***varImp*** function from ***caret*** package in R. Variable importance is calculated by sum of the decrease in error when split by a variable. In the graph below, the relative importance is presented as the variable importance divided by the highest variable importance value so that values are bounded between 0 and 100%.

The 20 most important features that are affecting our price predictions the most are visualized below.



It appears that the number of host listings, the cancellation policy and the cleaning fee are the most important factors predicting prices in the random forest model. Additional features that significantly affect our price predictions are the cancellation policy (super strict 30), the number of bathrooms and type of the room (shared room).

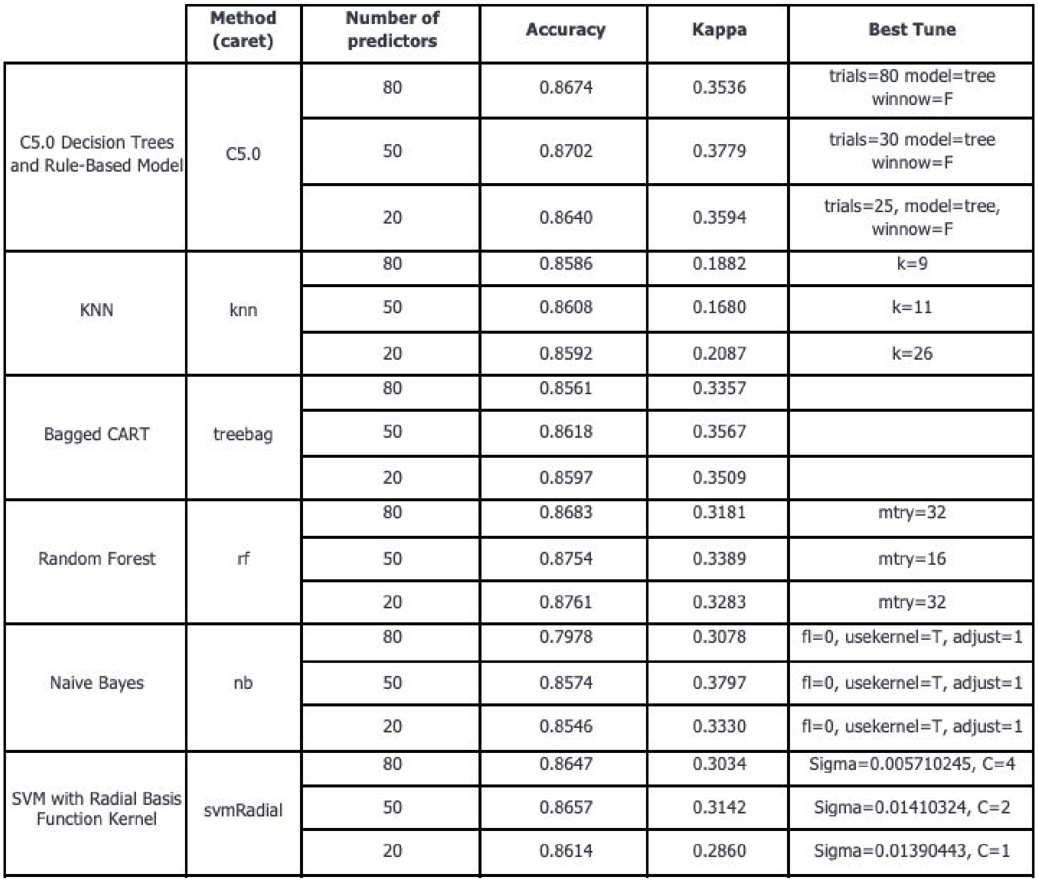
Classification

We run 10 classification models with 80 predictor variables (20 initial variables without ID variables, initial amenities variable and geographical variables, 30 amenity categorical variables and 30 variables with distances to different landmarks), 50 predictor variables (20 initial variables without ID and 30 amenity categorical variables) and 20 predictor variables (20 initial variables without ID). Similar as with regressions, we engineered the amenities and landmark features as described in the data exploration section.

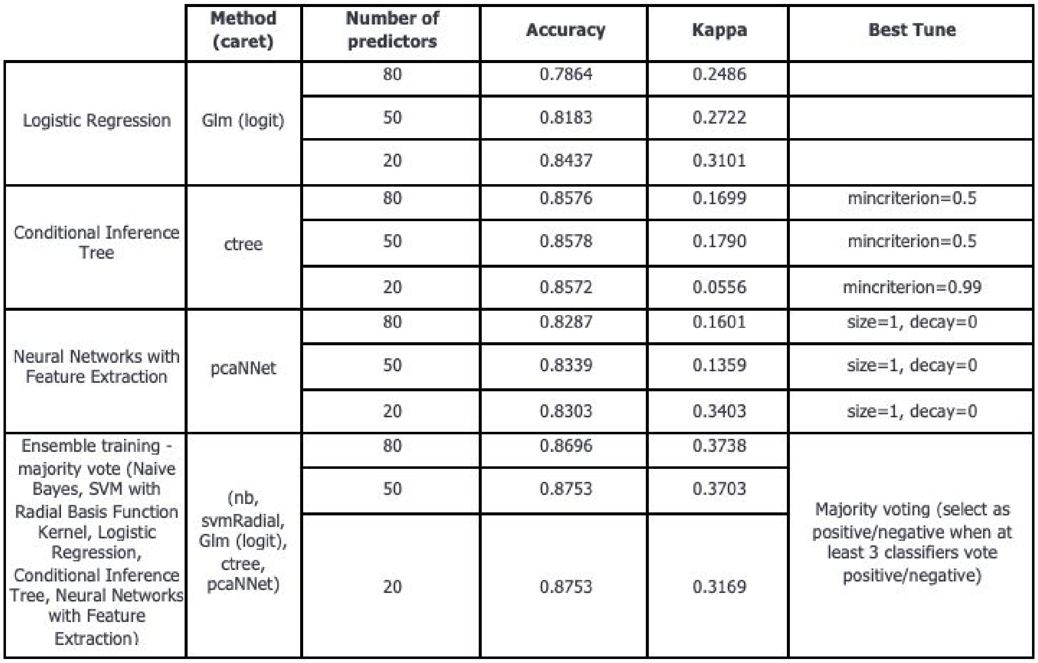
Overall, we ran the following algorithms: C5.0 Decision Trees and Rule-Based Model, KNN, Bagged CART, Random Forest, Naive Bayes, SVM with Radial Basis Function Kernel, Logistic Regression, Conditional Inference Tree, Neural Networks with Feature Extraction and an ensemble with the last 5 models and the majority vote rule (select as positive/negative when at least 3 classifiers vote positive/negative).

Since 85.65% of the data set has review score of 1, the dataset is imbalanced and the classifier can obtain high accuracy simply by always guessing the most frequent class. In order to account for that while comparing the models accuracies, we used the kappa statistics which adjusts accuracy by accounting for the possibility of a correct prediction by chance alone. Kappa values range from 0 to a maximum of 1, which indicates perfect agreement between the model's predictions and the true values.

With all 3 sets of variables, we run all 10 algorithms with separate tuning processes. We used ***caret*** package and 10-fold cross validation to obtain accuracies and kappas listed in the table below. To tune the parameters, we supplied the ***train*** function from ***caret*** package with matrices of possible tuning parameters and compared the cross validation accuracies produced by each model for each tuning parameter. We chose parameters that were producing models with the highest accuracies and kappas. The final tuning parameters are listed in Best Tune column in the tables below.

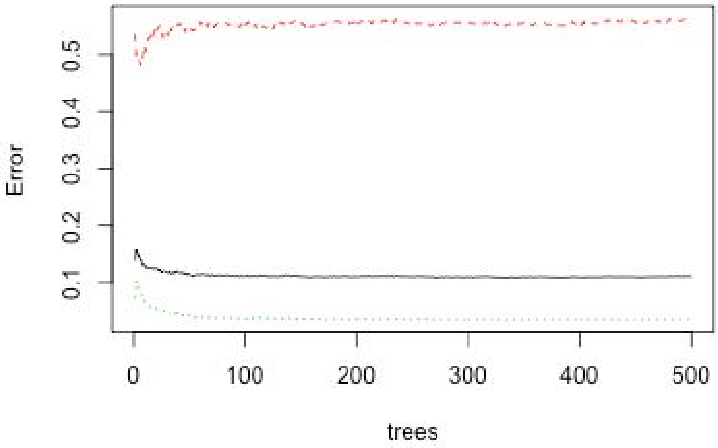


	Method (caret)	Number of predictors	Accuracy	Kappa	Best Tune
C5.0 Decision Trees and Rule-Based Model	C5.0	80	0.8674	0.3536	trials=80 model=tree winnow=F
		50	0.8702	0.3779	trials=30 model=tree winnow=F
		20	0.8640	0.3594	trials=25, model=tree, winnow=F
KNN	knn	80	0.8586	0.1882	k=9
		50	0.8608	0.1680	k=11
		20	0.8592	0.2087	k=26
Bagged CART	treebag	80	0.8561	0.3357	
		50	0.8618	0.3567	
		20	0.8597	0.3509	
Random Forest	rf	80	0.8683	0.3181	mtry=32
		50	0.8754	0.3389	mtry=16
		20	0.8761	0.3283	mtry=32
Naive Bayes	nb	80	0.7978	0.3078	fl=0, usekernel=T, adjust=1
		50	0.8574	0.3797	fl=0, usekernel=T, adjust=1
		20	0.8546	0.3330	fl=0, usekernel=T, adjust=1
SVM with Radial Basis Function Kernel	svmRadial	80	0.8647	0.3034	Sigma=0.005710245, C=4
		50	0.8657	0.3142	Sigma=0.01410324, C=2
		20	0.8614	0.2860	Sigma=0.01390443, C=1



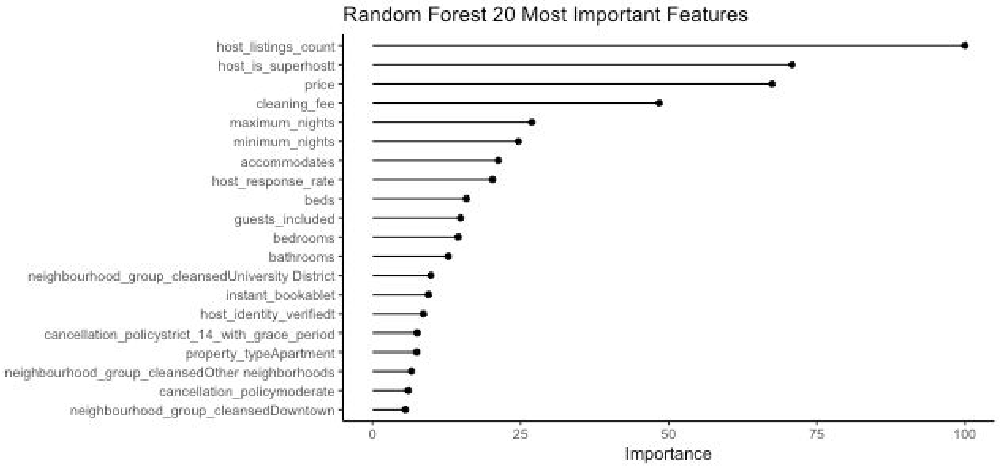
	Method (caret)	Number of predictors	Accuracy	Kappa	Best Tune
Logistic Regression	Glm (logit)	80	0.7864	0.2486	
		50	0.8183	0.2722	
		20	0.8437	0.3101	
Conditional Inference Tree	ctree	80	0.8576	0.1699	mincriterion=0.5
		50	0.8578	0.1790	mincriterion=0.5
		20	0.8572	0.0556	mincriterion=0.99
Neural Networks with Feature Extraction	pcaNNet	80	0.8287	0.1601	size=1, decay=0
		50	0.8339	0.1359	size=1, decay=0
		20	0.8303	0.3403	size=1, decay=0
Ensemble training - majority vote (Naive Bayes, SVM with Radial Basis Function Kernel, Logistic Regression, Conditional Inference Tree, Neural Networks with Feature Extraction)	(nb, svmRadial, Glm (logit), ctree, pcaNNet)	80	0.8696	0.3738	Majority voting (select as positive/negative when at least 3 classifiers vote positive/negative)
		50	0.8753	0.3703	
		20	0.8753	0.3169	

We decided to choose the random forest with 20 variables to make the final predictions because it has the highest accuracy (87.61%) on the test data in 10-fold cross validation and one of the highest kappas (32.83%). This is higher than the Naive Bayes accuracy of 85.46%. For this model, the best tuning uses the number of variables available for splitting at each tree node equal to 32. Below are the train and test errors of our final model as function of the number of trees.



The chosen model randomly selects a set of possible variables at each node. The performance is much better, but interpretation is a bit more difficult. We made predictions on given test_review set based on the final model using the predict function in R.

To explain the nature of the relationship between the predictors in the random forest and the predictions itself, we created a variable importance plot. Below are the most important features ranked in the order of importance (computed exactly as described in the Regression section).



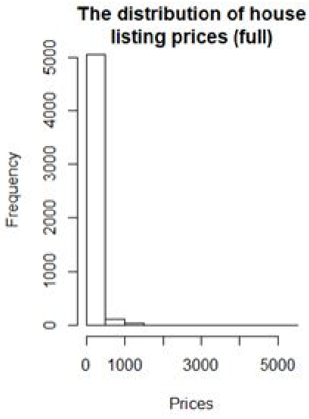
It appears that the host listings count, whether the host is a superhost and the price are the most important features in the chosen random forest model. Those are the variables that drive the model's predictions the most. Other variables significantly affecting our predictions are the cleaning fee, maximum number of nights and minimum number of nights.

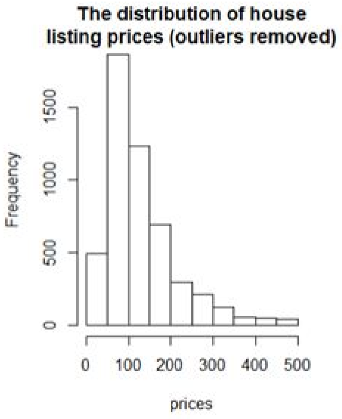
ANALYSIS OF RESULTS

Regression

If we predict the prices with the final model on the entire train set, the mean-squared of residuals is only 2135.81 (around $46). On average, those listings with larger than $40 residuals are priced significantly higher, have higher cleaning fee and are more suitable for events, with higher percentage of "Pack and Play Travel Crib", "Gym", "Hot Tub", "indoor fireplace", "cable tv", "Internet", which are important in predicting prices (see section "Supervised Learning: Regression"). This observation tells us that, the residuals of our final model may be dependent on the predictors, because it seems that the larger certain predictors are, the larger the price residuals become. In addition, it is also likely that the listings with higher prices may have higher prediction errors than the listings with lower prices, because it seems that expensive listings have larger residuals. Also, collinearity between variables may occur as well because some predictors clearly correlates to each other by intuition. E.g. "cable tv" and "tv"; "Pack and Play Travel Crib" and "Crib".

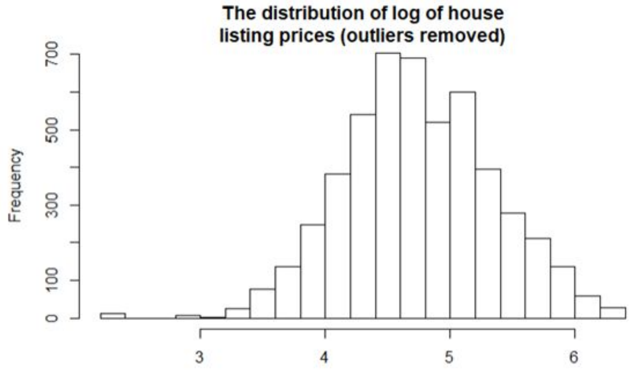
To validate the ideas above, we plotted the distribution of prices in the train set. Then, we found out that the listing prices are very right-skewed. The graph on the left below shows the distribution of prices of the full train set. The majority of prices listed are below $500 and the number of listings with price above $500 is only 147 out of the 5200 listings. After removing the outliers, the distribution of prices looks much less skewed as you may see on the right.





Furthermore, our final random forest model predicts well when the listing prices are less than $500. The corresponding mean-squared residual is 589.63 (around $24) for listings cheaper than $500. For listings with prices higher than $500, the mean-squared residual is 55284.38 (around $235). Hence, to improve the prediction, we may want to analyze and predict prices of the low/average price listings (less than $500 per stay) and luxurious listings (more than $500 per stay) separately.

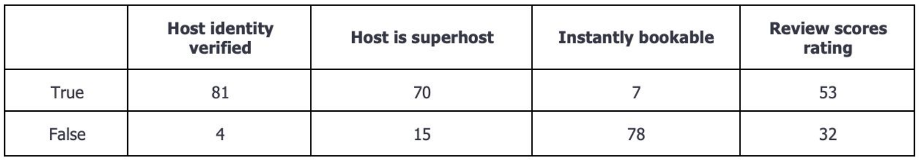
In addition, we also observe that the distribution of listing prices lower than $500 per stay is still right-skewed. After a log transformation of the prices, the distribution of prices looks much more likely to be normal (graphed below). Hence, we may also consider to train a regressor on a log-transformed response and see if the assumptions for random forest would be better held moving forward.



Overall, if we had more time, we would remove the outliers (listings with prices higher than $500) for the train set, reduce collinearity by variable selection, and try log transformations of the prices to improve the residual distribution and the mean squared error of the model.

Classification

If we run the final model on the entire set, it appears like only 2% of observations are misclassified. Those are mostly listings were the host is not a superhost, host's identity is not verified and where property/room type is an entire home/apartment in downtown Seattle, minimum nights of 1-5 and maximum nights of 1125 with prices up to $200/night and above $900/night that are instantly bookable. The vast majority of misclassified listings is not instantly bookable and have a review score of 0. A table with the number of misclassified observations in selected categories is included below.



	Host identity verified	Host is superhost	Instantly bookable	Review scores rating
True	81	70	7	53
False	4	15	78	32

The model is doing well on predicting the review score of listings located in other neighborhoods, maximum nights up to 400, moderate cancellation policy, with review score of 1, up to 5 guests included, entire home/apartment room type, up to 2 beds, 1 bedroom with prices up to $400 that are not instantly bookable.

If we had more time, we would try to extract more variables that are not included in the initial dataset. For example, we would try to include variables like distance to the airport, Seattle tax districts and distance to best local restaurants and conference centers. That could potentially improve the model's predictive performance. Additionally, we believe that we could achieve better predictive power of our models if we had a more balanced train dataset (more listings with a review score of 0). Right now, only 14.35% of the observations have a review score of 0 and 62.35% of misclassified observations are those with the true review score of 0.

EXTRA CREDIT

We performed a similar analysis for Airbnb listings in Asheville, NC.

Data Processing

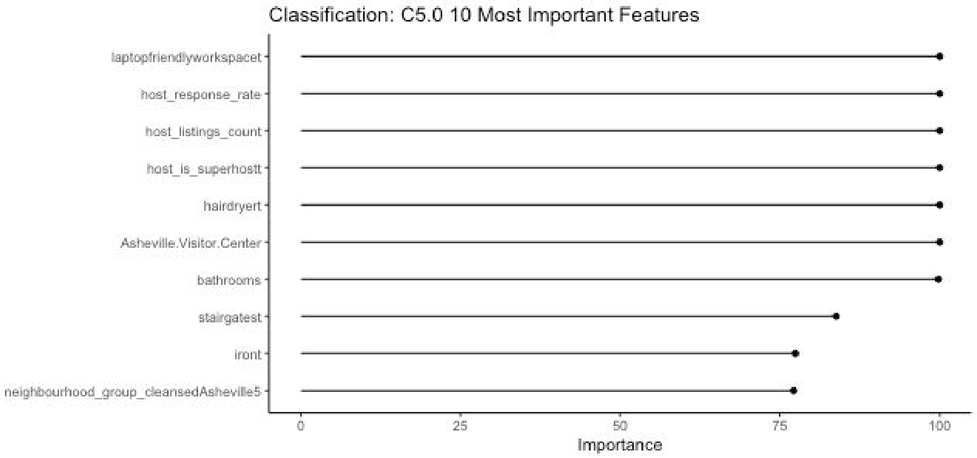
We retrieved listings data for Asheville, NC from the Airbnb site. We removed any columns that do not exist in the initial datasets for Seattle. For the cleaning fee was not present for a listing, we filled in 0. We dropped the incomplete data points. In Price dataset, we removed 3 outliers with prices of $10k/night. We then selected 19 most popular landmarks from TripAdvisor and computed distances to those landmarks just like we did in the dataset for Seattle. We also processed amenities in the same way as we did for listings in Seattle. Additionally, we retrieved the neighbourhood_group_cleansed variable from the zip codes provided in the neighbourhood_cleansed variable. Overall, we identified 68 independent variables and 1843 observations for Price dataset and 73 independent variables and 1555 observations for Review dataset.

Then, we run the algorithms that proved most effective on the dataset in Seattle. For each model, we performed a 10-fold cross validation. The results are displayed in tables below.

Classification

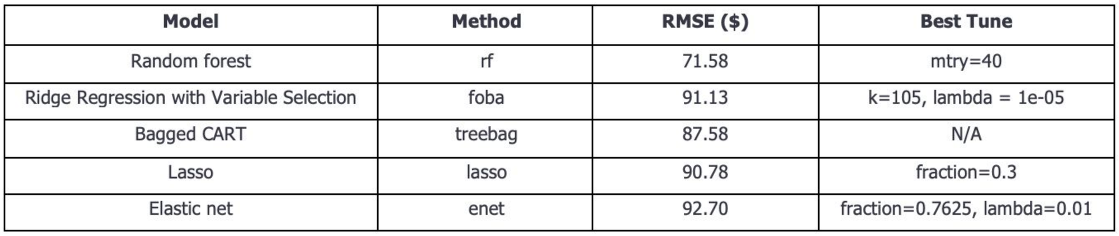
Model	Method	Accuracy	Kappa	Best Tune
Random Forest	rf	0.9627	0.2285	mtry=2
C5.0 Decision Trees and Rule-Based Model	C5.0	0.9646	0.3614	trials=40, model=nrules, winnow=T
Bagged CART	treebag	0.9582	0.3115	N/A
Naive Bayes	nb	0.9511	0.2807	usekernel=T
Decision tree	ctree	0.9576	0.1640	mincriterion=0.99

For classification, C5.0 performs best as it has the highest accuracy (96.27%) and highest kappa (22.85%). This is higher than the Naive Bayes accuracy of 95.11%. The final tuning parameters are 40 boosting iterations, the fact that the tree was decomposed into a rule-based model and predictor winnowing (i.e feature selection) was used. Below are the 10 most important features ranked in the order of importance (computed exactly as described in the Regression section).



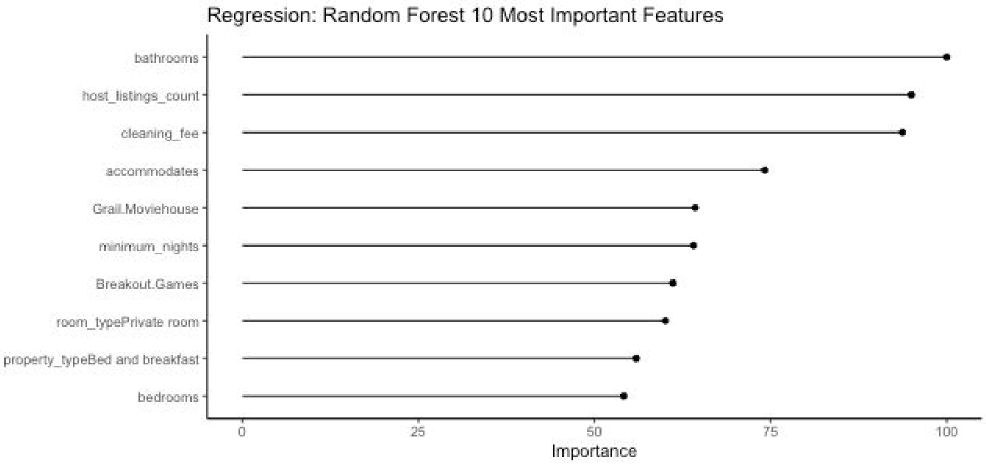
It appears that the laptop-friendly workspace, host response rate and the number of host's listings are the most important features in the chosen model. Those are the variables that drive the model's predictions the most. Other variables significantly affecting our predictions are whether the host is a superhost, whether the listing has a hairdryer and the distance to the Asheville Visitor Center.

Regression



Model	Method	RMSE ($)	Best Tune
Random forest	rf	71.58	mtry=40
Ridge Regression with Variable Selection	foba	91.13	k=105, lambda = 1e-05
Bagged CART	treebag	87.58	N/A
Lasso	lasso	90.78	fraction=0.3
Elastic net	enet	92.70	fraction=0.7625, lambda=0.01

For regression, random forest performs best as it has the lowest RMSE ($71.58). The best tuning uses the number of variables available for splitting at each tree node equal to 40. Below are the 10 most important features ranked in the order of importance (computed exactly as described in the Regression section).



It appears that the number of bathrooms, host listings count and cleaning fee are the most important features in the chosen model. Those are the variables that drive the model's predictions the most. Other variables significantly affecting our predictions are the number of people a listing accommodates, the distance to Grail Moviehouse and the number of minimum nights per stay.

Analysis of Results

After comparing the feature importance graphs for Seattle and Asheville, we can notice that for Seattle the most important features affecting the prices and review scores are features closely connected with sightseeing in a city with high living cost (host listings count, price, cleaning fee and strict cancellation policy). For Asheville, it looks like most important features are those that people look at when they go on business travels (i.e. laptop friendly workspace, hairdryer, number of bathrooms, host listings count or host response rate).