Using Statistical Learning to Guide Decision Makings in NYC Airbnb

Jiaqi Li (jl5025), Yuqi Tu (yt2604), Xin Yin (xy2364)

INTRODUCTION

Airbnb is a marketplace for short term rentals and hospitality service, allowing customers to list part or all of your living space for others to rent. The company itself has grown rapidly from its founding in 2008 to a 30 billion dollar valuation in 2016 and is currently worth more than any hotel chain in the world.[1]

The main motivation behind our project is to assist customers to evaluate the Airbnb hosting under $1000 daily rent in the New York City (NYC). The evaluation could be done by predicting daily rental prices with regression modelings to customers in the city that would enable them to find the most suitable accommodation from the Airbnb listings. In addition, classification of review scores by using daily rental, the number of reviews, location, room types, minimum nights for booking, the number of housing that the hosts listed, and the availability in one year will also benefit customers to explore rental housing in NYC and find their favorite homestays.

For conducting the evaluation, The dataset 'nyc_airbnb.zip' is accessible from 'Inside Airbnb' on September 2, 2017 and will be used in this project. The version of the data that we use can be found here. This data contain a single dataframe with 40,753 rows of data on 17 variables. Inside Airbnb is a non-commercial set of tools and provides filters and key metrics so you can see how Airbnb is being used to compete with the residential housing market.

EXPLORATORY ANALYSIS

Locations for "Pricey" homestays, defined as listings costing more than the overall mean of all the prices, and rest labeled "affordable" locations can be seen in Fig. 1. Now, we check if a particular area in New York has more expensive stays than others. We observe that, in general, prices of homestays are higher in Manhattan.

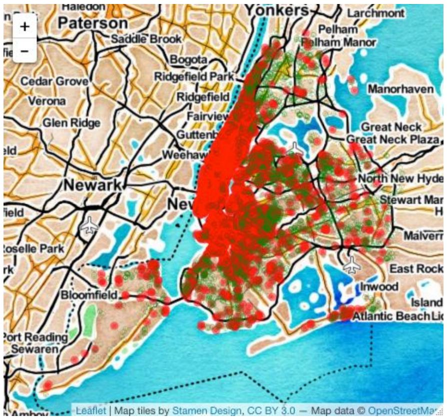


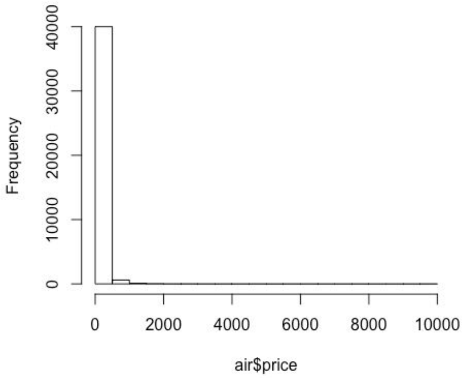
Figure 1. **Pricey vs. Affordable listings**

Note: Green are affordable and red are expensive homestays.

Dependent variable distribution

The first analysis of the datasets is looking into the distribution of the dependent variable, *Price*, among the listings. Fig. 2 shows a histogram of all prices in the data.

In case the data contains outliers (defined as listings with `prices >= 1000`, all listings to the right of value 1000), we want to exclude these extraordinarily expensive listings. This will change the histogram to Fig. 3.



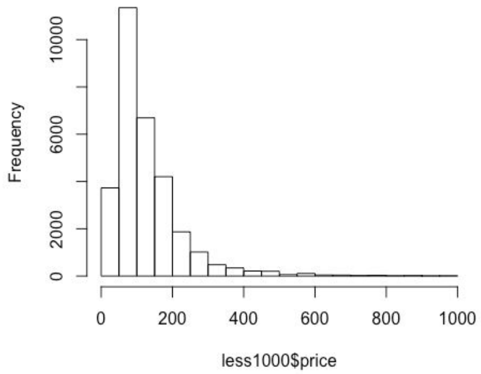
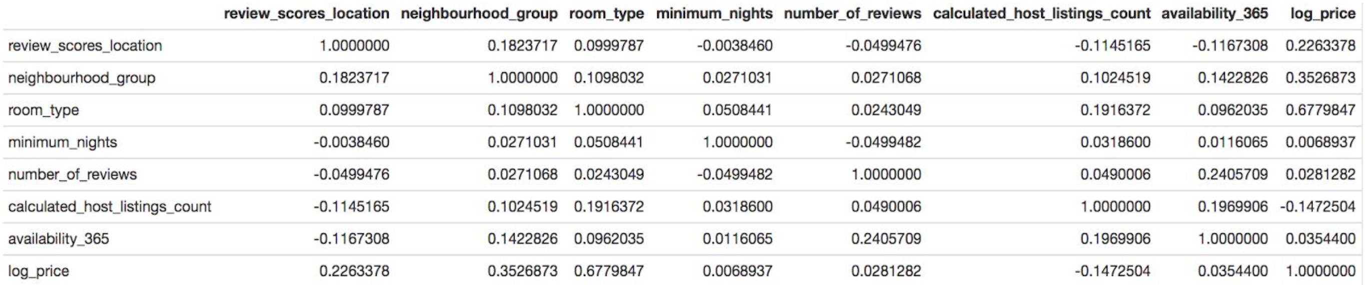


Figure 2. Histogram of *Price* distribution **Figure 3. Histogram of *Price<1000* distribution**

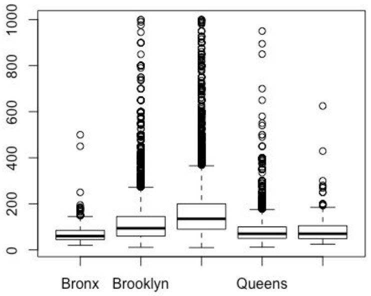
Correlation matrix

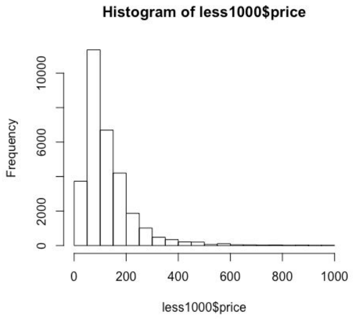
Table 1 shows correlation coefficients between sets of variables. This allows us to identify pairs with higher correlations:

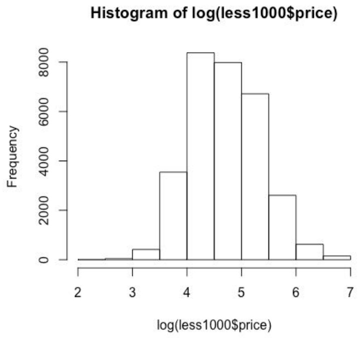


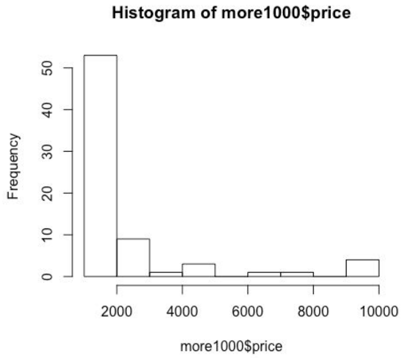
	review_scores_location	neighbourhood_group	room_type	minimum_nights	number_of_reviews	calculated_host_listings_count	availability_365	log_price
review_scores_location	1.0000000	0.1823717	0.0999787	-0.0038460	-0.0499476	-0.1145165	-0.1167308	0.2263378
neighbourhood_group	0.1823717	1.0000000	0.1098032	0.0271031	0.0271068	0.1024519	0.1422826	0.3526873
room_type	0.0999787	0.1098032	1.0000000	0.0508441	0.0243049	0.1916372	0.0962035	0.6779847
minimum_nights	-0.0038460	0.0271031	0.0508441	1.0000000	-0.0499482	0.0318600	0.0116065	0.0068937
number_of_reviews	-0.0499476	0.0271068	0.0243049	-0.0499482	1.0000000	0.0490006	0.2405709	0.0281282
calculated_host_listings_count	-0.1145165	0.1024519	0.1916372	0.0318600	0.0490006	1.0000000	0.1969906	-0.1472504
availability_365	-0.1167308	0.1422826	0.0962035	0.0116065	0.2405709	0.1969906	1.0000000	0.0354400
log_price	0.2263378	0.3526873	0.6779847	0.0068937	0.0281282	-0.1472504	0.0354400	1.0000000

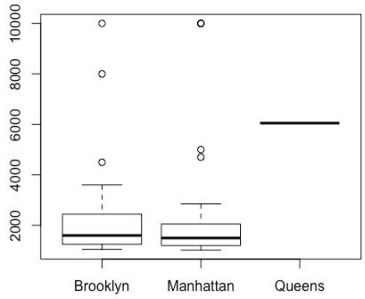
Table 1. Correlation between predictors of Airbnb data

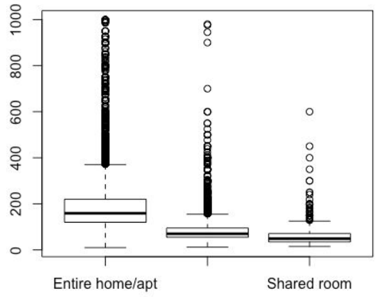












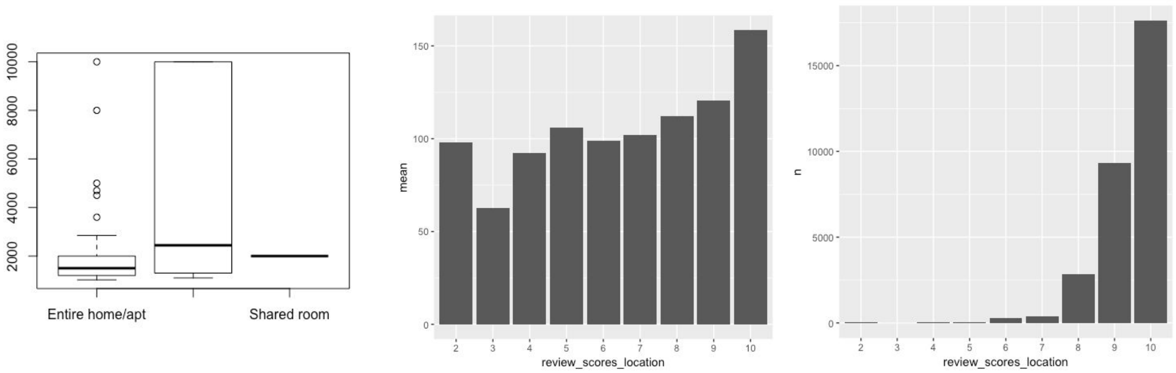


Figure 4. **Boxplots and histograms of some predictors of Airbnb data**

We can identify a couple of interesting observations: *Price* correlates strongly with *Review Scores* and *Neighbourhoods*. *Number of reviews* correlates strongly with *availability*. Furthermore, we see a higher correlation between *Room type* and the dependent variable *Price*. None of them is a surprise and confirm intuition. (Note: some variable correlations are shown as plots in Fig.4)

DATA CLEAN

We first removed NAs from dataset because most functions for regression and classification cannot take missing values, and then divided the "nyc_airbnb" dataset into two sub-dataset - price less than $1000 (`less1000`) and price greater than $1000 (`more1000`). Our project mainly focus on listings have price less than $1000, as mentioned in the exploration part that price greater than $1000 are in some extremes that has different pattern from price less than $1000. After data exploration, we selected relevant predictors to regression and classification respectively. `id`, `name`, `host_id`, and `host_name` were removed because they are nominal variables that are not meaningful in regression and classification in our project. `number_of_reviews` and `reviews_per_month` have correlation coefficient 0.49, so `reviews_per_month` is removed. `neighbourhood_group` and `neighbourhood` contain same information of neighbourhood besides that `neighbourhood` is more detailed. `neighbourhood group` is kept for simplicity.

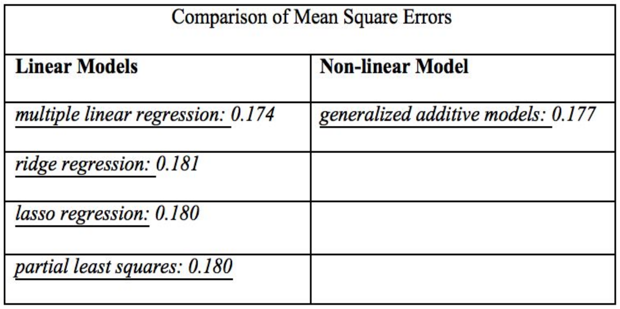
For regression, since price is skewed, we therefore take log transformation. `log_price` is used as response variable for regression, and `price` is removed. Final dataset for regression is `airbnb`. It contains 8 variables, including `review_scores_location`, `neighbourhood_group`, `room_type`, `minimum_nights`, `number_of_reviews`, `calculated_host_listings_count`, `availability_365` and `log_price`. For classification, we fit models to predict if the review score is perfect or not. We created a new variable `review_10` and assigned `reviews_scores_location` less than 10 to "`not10`" and `reviews_scores_location` greater than 10 to "`10`". `review_scores_location` is removed. The final dataset used for classification is `airbnb2` containing 9 variables - `latitude`, `longitude`, `room_type`, `price`, `minimum_nights`, `number_of_reviews`, `calculated_host_listings_count`, `availability_365`, and `review_10`.

METHODS & RESULTS

<u>Regression</u>

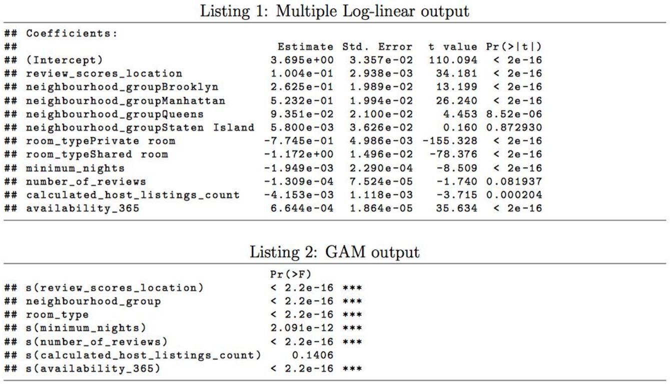
We first built some regression models, including linear and non-linear models. One kind of regression models, *Multiple linear regression,* assumes that there is no multicollinearity in the data. However, if there is multicollinearity, ridge and lasso regression are able to handle that. Our linear models include *multiple linear regression, ridge regression, lasso regression* and *partial least squares.* In *ridge regression*, the best tuning parameter *lambda* selected by 10-fold cross-validation is 0.042. The test error rate of *ridge regression* is 0.181. In *lasso regression*, the best tuning parameter *lambda* selected by 10-fold cross-validation is 0.00036. The test error rate of *lasso regression* is 0.180.

Then we used *generalized additive model* as the non-linear model. The results are summarized in the Table 2 and Table 3 below.



Comparison of Mean Square Errors	
Linear Models	**Non-linear Model**
multiple linear regression: 0.174	*generalized additive models: 0.177*
ridge regression: 0.181	
lasso regression: 0.180	
partial least squares: 0.180	

Table 2. Resulting MSE of regression models



Listing 1: Multiple Log-linear output

```
## Coefficients:
##                               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                  3.695e+00  3.357e-02  110.094  < 2e-16
## review_scores_location       1.004e-01  2.938e-03   34.181  < 2e-16
## neighbourhood_groupBrooklyn  2.625e-01  1.989e-02   13.199  < 2e-16
## neighbourhood_groupManhattan 5.232e-01  1.994e-02   26.240  < 2e-16
## neighbourhood_groupQueens    9.351e-02  2.100e-02    4.453 8.52e-06
## neighbourhood_groupStaten Island 5.800e-03  3.626e-02    0.160 0.872930
## room_typePrivate room       -7.745e-01  4.986e-03 -155.328  < 2e-16
## room_typeShared room        -1.172e+00  1.496e-02  -78.376  < 2e-16
## minimum_nights              -1.949e-03  2.290e-04   -8.509  < 2e-16
## number_of_reviews           -1.309e-04  7.524e-05   -1.740 0.081937
## calculated_host_listings_count -4.153e-03  1.118e-03   -3.715 0.000204
## availability_365             6.644e-04  1.864e-05   35.634  < 2e-16
```

Listing 2: GAM output

```
                                  Pr(>F)
## s(review_scores_location)       < 2.2e-16 ***
## neighbourhood_group             < 2.2e-16 ***
## room_type                       < 2.2e-16 ***
## s(minimum_nights)               2.091e-12 ***
## s(number_of_reviews)            < 2.2e-16 ***
## s(calculated_host_listings_count)   0.1406
## s(availability_365)             < 2.2e-16 ***
```

Table 3. Linear regression and GAM output

It looks like *multiple linear regression* and *generalized additive model* performed the best - MSE is 0.174 and 0.177 respectively. The test error rates of the shrinkage methods and the dimension reduction methods were slightly higher, which they were doing feature selection and using linear combinations of the original features to fit a linear model via least squares. It is not surprising that in this data the compressed explanatory information performed slightly worse than full explanatory information. This can be explained from another aspect - most of the predictors included were significant.

Classification

To classify the review scores by using daily rental, the number of reviews, location, room types, minimum nights for booking, the number of housing hosts listed, and the availability in one year, *logistic regression, linear discriminant analysis , quadratic discriminant analysis, k-nearest neighbors, tree methods*, including *classification tree, bagging, random Forest,* and *boosting*, *support vector classifier with linear and non-linear kernels* and compared to see which model performed best.

Logistics regression

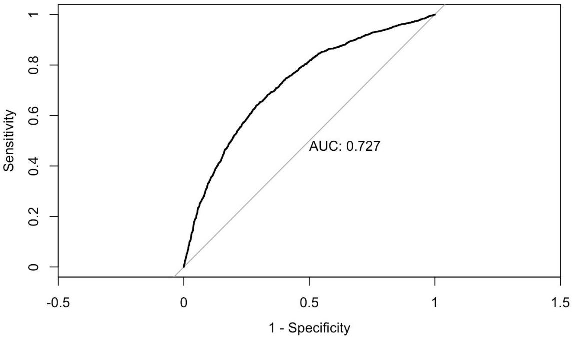


Figure 5. ROC curve of logistic regression for classification

Logistic regression can outperform LDA if gaussian assumptions are not met. First, Logistic regression does not require a linear relationship between the dependent and independent variables. Second, the error terms (residuals) do not need to be normally distributed. Third, homoscedasticity is not required. Finally, the dependent variable is not measured on an interval or ratio scale.

After fitting with logistic regression, the test error rate of logistics regression model is 0.3181 and the AUC of the test data is 0.727.

Linear Discriminant Analysis (LDA) & Quadratic Discriminant Analysis (QDA)

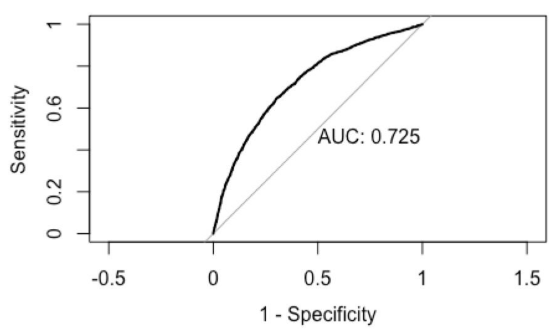


Figure 6. ROC curve of LDA model for classification

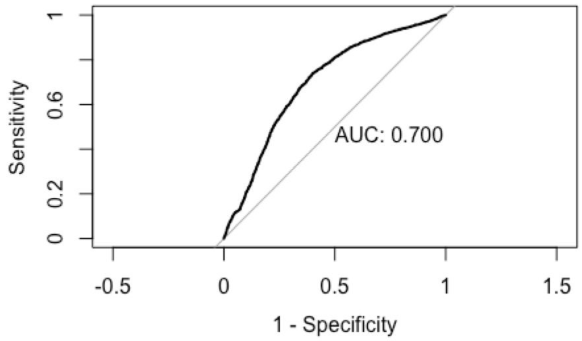


Figure 7. ROC curve of QDA model for classification

Review Score level / Variables	Longitude	Latitude	Number of Review	Availability in One Year	The number of housing that hosts listed	Minimum nights for booking
Review Score is 10	**0.00129**	**0.00209**	**852.51**	**17630**	**3.3457**	**60.112**
Review Score is not 10	**0.00189**	**0.00384**	**1357.0**	**18918**	**7.5810**	**224.44**

Table 4. Table of variances of different variables across different levels in the response (Review Score)

The LDA model needs to meet four assumptions, including multivariate normality, homogeneity of variance, multicollinearity, and independence. However, QDA does not need that variances among group variables are the same across levels of predictors as LDA does. After fitting with models, the test error rate of LDA model is 0.3209 and the AUC of the test data is 0.725 (Fig. 5). For QDA model, the test error rate is 0.3736 and AUC of the test data is 0.700 (Fig. 7). In addition, since some variables have similar variances across different levels of the response variable (Table 4), such as longitude, latitude and the number of housing that hosts listed, LDA should have a better performance than QDA according to their model assumptions. Since the rental housing observations are all come from the same city, it is also understandable that longitude and latitude have similar variances across different levels of the response variable.

Among these three parametric models, the logistics regression model provides the best result of the three methods since the test error rate is the lowest (31.81%). The three models have similar AUC from 0.700 to 0.725.

k-Nearest Neighbors (KNN)

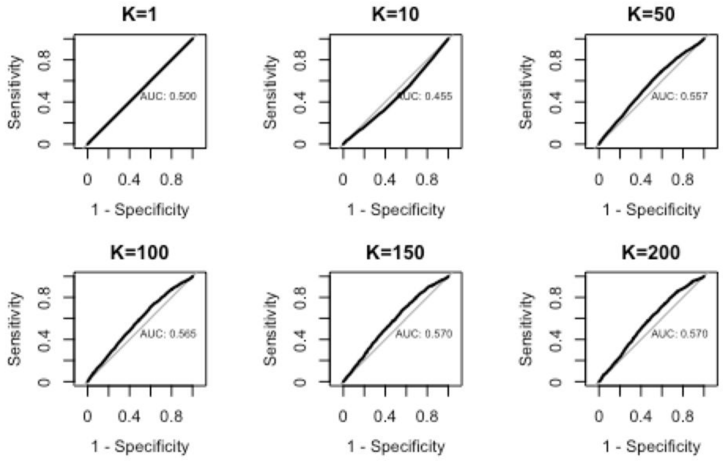


Figure 8. ROC curves of KNN models for classification

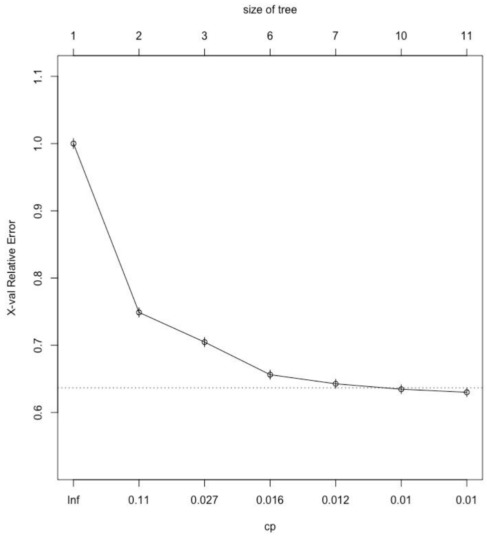
	Error rate of test data	AUC of test data
K = 1	41.21%	0.500
K = 10	36.54%	0.455
K = 50	34.24%	0.557
K = 100	33.93%	0.565
K = 150	33.93%	0.570
K = 200	33.95%	0.570

Table 5. Error rate and AUC of test data of KNN models for classification

KNN is a non-parametric technique, there is no assumption on the underlying data distribution. Based on Fig. 8 and the Table 5, the AUC increases with different k values.The highest AUC might happened when k=150 and k=200. KNN models use neighbors to classify each observation, the best prediction accuracy occured around k = 150 (test error rate is 0.3393). Therefore, KNN model performs best where k = 150.

Tree classification

The tree classification is built using best cp=0.01 (Fig. 9), chose by one-standard-error (1SE) rule. It corresponds to tree size=11 (Fig. 10). The first split criterion is longitude < -73.96. Taking the bottom left node as an example: for `longitude < -73.96` and `latitude >=40.66`, the predicted `review_10` will be classified as “`10`”. The number of observation in that branch is 11615 with deviance 2634. The probability of classify an airbnb’s review score as “`10`” within that area is 0.773. The test error rate of tree classification is 0.2482.



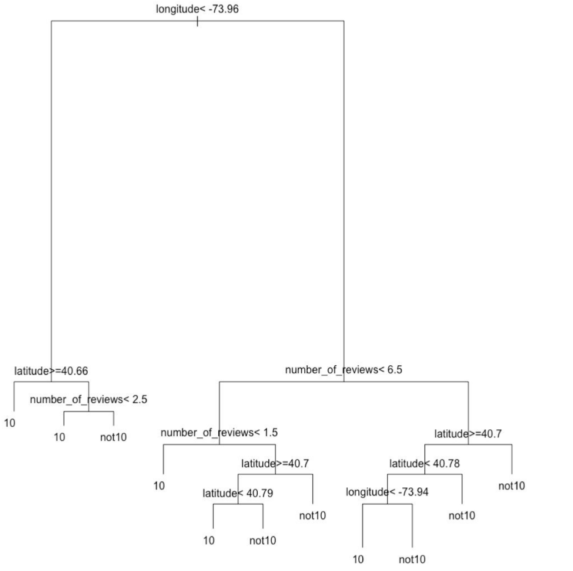


Figure 9. Cross-validation error rate of tree classification **Figure 10. Classification Tree using best CP**

Bagging & Random Forests

Bagging has OOB estimate of error rate is 24.27%. The test error rate is 0.2330. There is a improvement in predicted accuracy over prediction using a single tree.

Random forests generates the same test error rate as bagging of 0.2330. OOB estimate of error rate is 23.86%

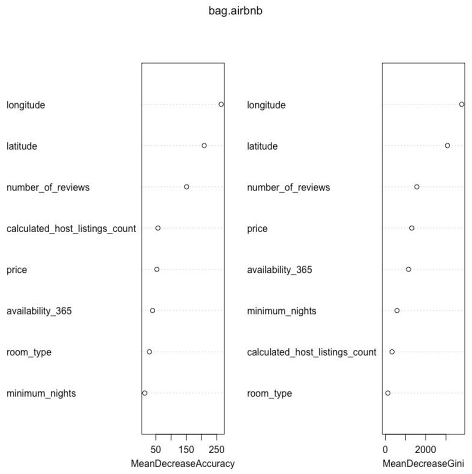


Figure 11. **Variable importance plot of bagging**

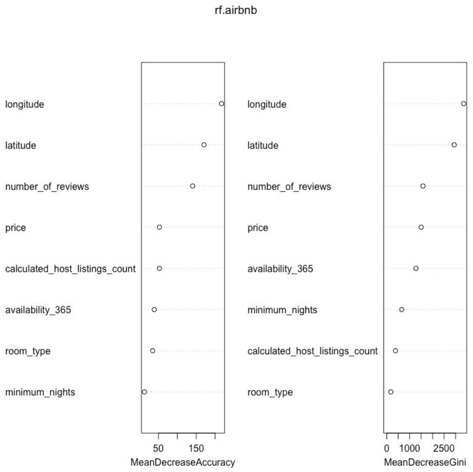


Figure 12. **Variable importance plot of random forest**

The two methods reported pretty similar variable importance: across all of the trees considered in the bagging and random forest, the longitude and latitude are by far the most important variables in predicting perfect review score (Fig. 11 & Fig. 12).

Boosting

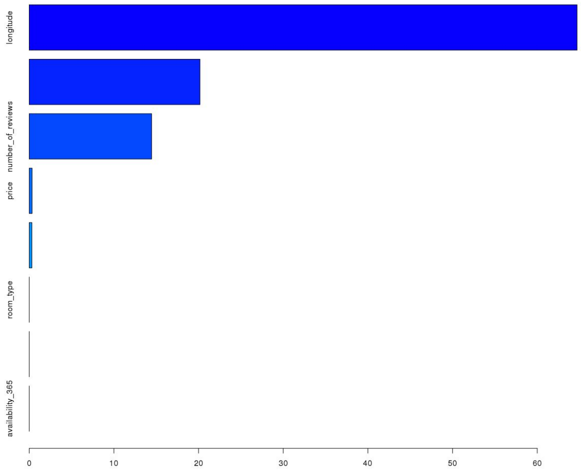


Figure 13. **Variable importance plot**

We see that `longitude` is the most important variables (Fig. 13); latitude and number of reviews are somewhat important. The test error rate of boosting is 0.2560 when using 3000 trees to build the model.

Support vector classifier with linear & non-linear kernel

In support vector classifier with linear kernel, the best tuning parameter cost selected by 10-fold cross-validation is 1. The test error rate of support vector classifier is 0.3216.

In support vector classifier with non-linear kernel, the best tuning parameter cost and gamma selected by cross-validation are 10 and 0.1 respectively. The test error rate is 0.2529.

Compared with linear kernel, non-linear kernel has better performance, which suggests the boundary between the two classes is non-linear.

SUMMARY

From the regression models, multiple linear regression performed the best whose test error is 0.174 and our data meets its normal assumption after transforming the outcome. *Generalized additive model* performed relatively moderate; *ridge and lasso regression* performed the worse. However, these models did not differ much and the test error rates were close. *Generalized additive model* is similar to *multiple linear regression* by using linear combinations of the original features to fit a linear model via least squares, which is not a surprise that their MSEs are closer. Considering we do not have many explanatory variables in this data and most of all variables are statistically significant, *ridge and lasso regression* would not take the advantage of their "shrinking power" to make them the best models.

	Logistic	LDA	QDA	KNN	Tree classification	Bagging	Random Forest	Boosting	SVC	SVM
Test error rate	0.3181	0.3209	0.3736	0.3393	0.2482	0.2330	0.2330	0.2560	0.326	0.2529
AUC	0.727	0.725	0.700	0.570						

Table 6. Classification model performances

Based on above classification models, longitude and latitude, which is the location of airbnb is the most important variable in classifying airbnbs' perfect review score. Fig. 14 shows that housing in downtown, midtown, upper west/east side Manhattan, Brooklyn, and northwest side of Queens have higher proportion of perfect review scores than other area of New York. It is a good reference for people who wish to choose housing based on ratings. Table 6 shows the performances of models used for classification. Bagging and random forest have the best performances in terms of test error rate, which is reasonable as bagging and random forest aggregating many decision trees therefore have higher predictive power.

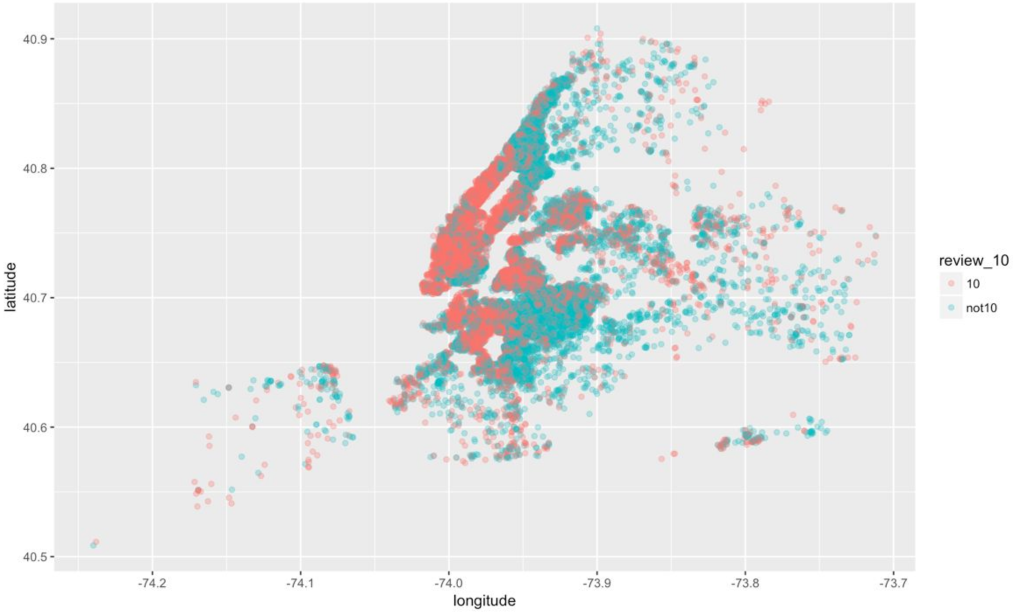


Figure 14. New York map with two levels of review score

REFERENCE

1. Devlin, Josh. "Machine Learning Fundamentals: Predicting Airbnb Prices." *Dataquest*, Dataquest, 6 Feb. 2018, www.dataquest.io/blog/machine-learning-tutorial/.