Price Prediction and Qualities Responsible for Becoming a Superhost: An Airbnb Case Study

Digvijay Karamchandani
Department of CSE, UCSD
PID: A53220055
dkaramch@eng.ucsd.edu

Chandana Lakshminarayana
Department of CSE, UCSD
PID: A53213890
clakshmi@eng.ucsd.edu

Kriti Aggarwal
Department of CSE, UCSD
PID: A53214465
kriti@eng.ucsd.edu

Saksham Sharma
Department of CSE, UCSD
PID: A53220021
sas111@eng.ucsd.edu

ABSTRACT

In this project, we consider the problem of predicting prices for Airbnb rentals from various features available from the Boston Airbnb open dataset. We also consider the problem of finding out the factors which are responsible for becoming a superhost on Airbnb. The project involves finding out the appropriate features for each of the above mentioned tasks. Once the features are extracted, they are trained using machine learning techniques such as Linear regression, Random Forest regression and Gradient Boosting regression for price prediction. For superhost prediction task, the extracted features are trained using SVM classifier and Decision Tree classifier. For the price prediction task, Gradient Boosting regression preformed the best, yielding a test mean absolute error (MAE) of 0.27509908. Decision Tree classifier performed the best for the superhost prediction task, yielding a test MAE of 0.09413.

KEYWORDS

Linear regression, Random Forest regression and Gradient Boosting regression, SVM classifier, Decision Tree classifier, mean absolute error

1 INTRODUCTION

Now a days, many people use online websites to list or rent short-term lodging including vacation rentals, apartment rentals, homestays, hostel beds, or hotel rooms. Airbnb is one such site which does not own any lodging, but is merely a broker and receives percentage service fees (commissions) from both the guests and hosts in conjunction with every booking. It has over 3 million lodging listings in 65,000 cities and 191 countries, and the cost of lodging is set by the host[2]. But, the problem is for a new user who wants to rent short-term lodging but might not know the appropriate price to ask for. Also, he might not know about the qualities he should possess in order to become a superhost on Airbnb.

The price of a rental depends on many features such as number of bedrooms & bathrooms, neighborhood, room type, latitude & longitude, reviews per month etc. So, our goal is to determine the relevant features by performing the exploratory analysis on the dataset. After extracting the relevant features, we plan to train them using five different machine learning techniques which are as follows:

(1) Linear regression
(2) Ridge regression
(3) Lasso regression
(4) Random Forest regression
(5) Gradient Boosting regression

Comparison of the different models and reasoning about them is provided by subsequent sections.

Airbnb also provides a "Superhost" badge to its users on the basis of few benchmarks. Some of the benchmarks are as follows:

(1) High response rate: Superhosts respond to guests quickly and maintain a 90% response rate or higher.
(2) Commitment: Superhosts honor confirmed reservations-they rarely cancel.
(3) 5 star reviews: Superhosts provide listings that inspire enthusiastic reviews. At least 80% of their reviews need to be 5 stars.
(4) Experience: Superhosts complete at least 10 trips in their listings in a year.

Apart from the above mentioned benchmarks, we plan to explore other characteristics that can help a user to become a superhost. Once, the relevant features are extracted, we will train them using two machine learning techniques: (1) SVM classifier (2) Decision Tree classifier. Further details about the superhost prediction task are given in subsequent sections.

2 LITERATURE REVIEW

The dataset which we are using for our predictive tasks is the Boston AIRBNB Open Data[1]. The dataset is a part of Inside Airbnb and the original source can be found at [4]. The data visualization of the Airbnb Boston listings is shown in figure 1.

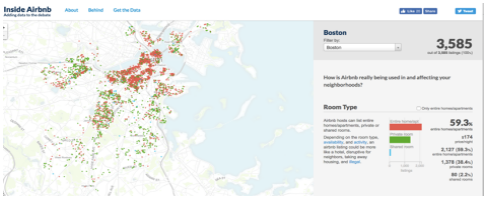

Figure 1: Airbnb Boston listings data visualization

2.1 Inspiration behind the dataset

(1) Can you describe the vibe of each Boston neighborhood using listing descriptions?
(2) What are the busiest times of the year to visit Boston? By how much do prices spike?
(3) Is there a general upward trend of both new Airbnb listings and total Airbnb visitors to visit Boston?

2.2 Previous Work

The previous work done on this dataset were based on the below problem statements:

(1) Modeling price prediction
(2) Effect of holiday seasons on pricing and price spikes
(3) Sentiment analysis and collocation of reviews

For the first predictive task of modeling price predictions, the kaggle site has an activity for the same task[5] where the user performed linear regression on important features such as neighborhood, and room related features such as number of bedrooms, bathrooms, etc. We used this as the baseline for our predictive task.

We studied on what models work best for similar predictive tasks and found that if one takes several predictions and aggregates them, then the resulting aggregate forecast in many cases will outperform the individual forecasts. Bagging, boosting and stacking are all based exactly on this idea. If the aim is purely prediction, then in most cases this is the best one can do. The drawback of this method is that it is a black-box approach that returns the result but does not help one to understand and interpret it. It is also more computationally intensive than any other method since one has to build multiple predictors to come up with a final prediction [6],[7].

We decided to apply ensemble techniques like gradient boosting and random forests for our task as the literature suggests these methods to be state-of-the-art. We also need to engineer and extract the features which are most relevant for price prediction.

For the second task of finding characteristics of a super host, we found limited literature. We decided to extract the characteristics by two approaches. The first approach was to find aspect of review text related to highly and lowly rated reviews. For this we found Latent Dirichlet allocation and word clouds to be commonly used methods [8].

The second approach was to classify super host and non super host on various attributes, to find differentiating features which are significant. From past literature on similar tasks, we found that binary classifiers like SVM and decision trees along with proper feature engineering was to work well with our case [9].

3 DATASET

In this section we study the dataset and perform analysis on it.

3.1 Basic Statistics and Properties

We used Boston Airbnb Open Dataset[1] for our case study. This dataset provides the following information:

(1) Listings: A csv file which has listings across various properties in Boston. (3,585 listings)
(2) Calendar: A csv file which has details on booking date and price for each listing. (1,048,575 bookings)
(3) Review: A csv file which has review provided for a particular listing id by a particular reviewer. (68,275 reviews)

We used all the above mentioned datasets for our project. The basic schema of the datasets is shown in figure 3

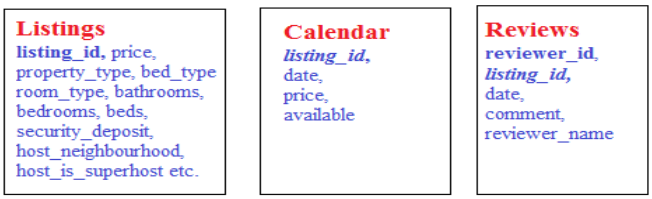

Figure 2: Schema of datasets.

3.2 Exploratory Data Analysis

To develop a deeper insight into how prices correlate with respect to different features such as room configurations (number of bedrooms, room type), day of the year, neighborhood etc. we carried out extensive exploratory analysis on our data. Few snapshots and interesting findings are given in brief in following sub sections.

There were a total of 3586 listings in the Airbnb Boston data. After cleaning the data(removing null rows, incomplete data, etc.) we were left with only 2725 listings. Out of these hosts, only 16.8% of the hosts were super hosts. Hence, the dataset was unbalanced with respect to the type of the host. To further find how super hosts differ from normal hosts, we plotted the attributes with number of super and normal hosts. We found that there were several factors that had a very high correlation with the host being a super host viz a viz cleanliness score, number of reviews, rating score, acceptance rate of the host, response rate of the user, rating of the host location, whether the host has been verified or not.

3.2.1 *Price versus neighborhood*

. The price that the customers pay for a property is proportional to the neighborhood. To find this correlation in our dataset, we plotted prices (y axis) versus the neighborhoods sorted with respect to price (x axis) as shown in figure 4.The most expensive neighborhood turns out to be Miami beach followed by Downtown. And the cheapest neighborhoods are South Boston and Chelsea. Figure 4 captures an interesting relationship between prices and how neighborhood affects it.

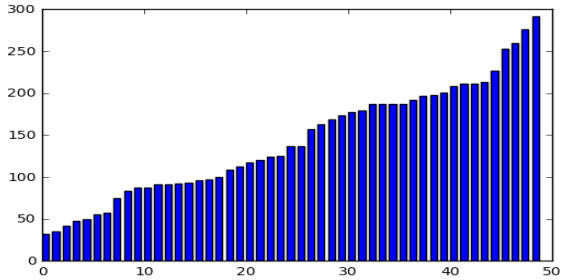

Figure 4: Prices versus neighborhood

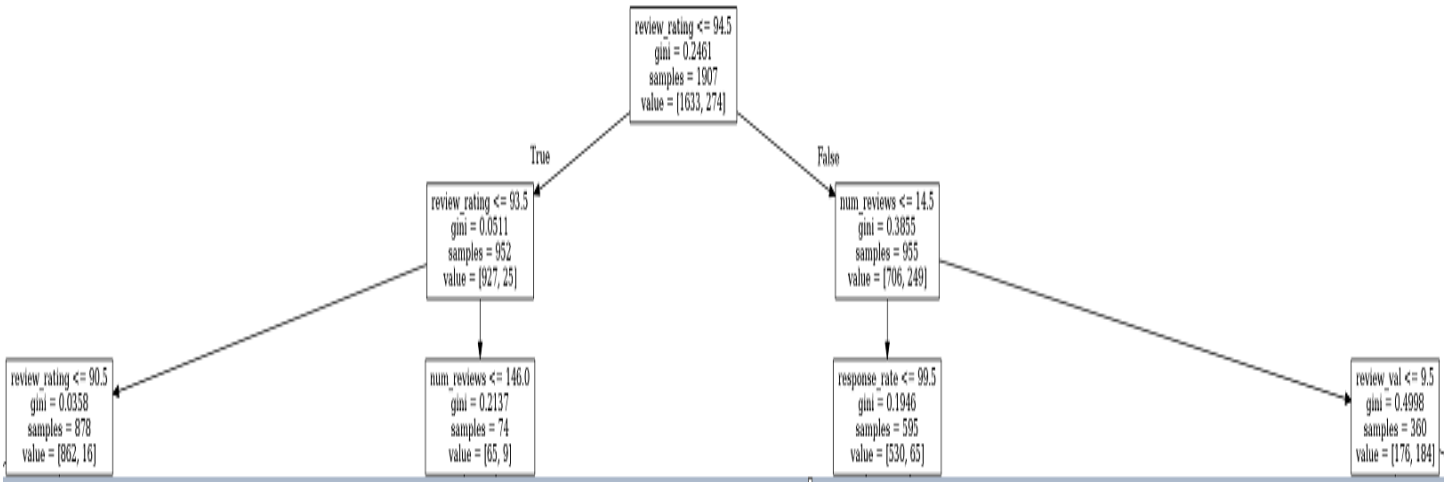

Figure 3: Decision tree for predicting a super host.

3.2.2 Price versus room type

. The price of the listing also depends on the configurations of the property being rented out. These configurations include type of room, number of bedrooms/bathrooms. Figure 5 shows price (y axis) versus the type of room sorted with respect to price (x axis). The room type with highest price was found to be Townhouse followed by Bread and Breakfast. And room types like Condominium, Dorms and Boat had lesser price. The notion of prices being affected by the configurations of the room is reaffirmed by figure 5.

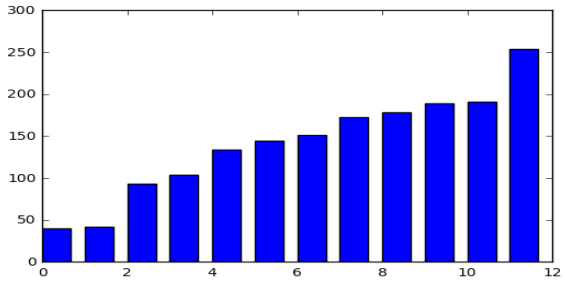

Figure 5: Price versus room type

3.2.3 Price versus other important features

. We saw that the prices usually spiked during the holiday season. We categorized holiday season to four types, New Years, Christmas, Thanks Giving and New Year's Eve. The price spiked mostly during New Years and Thanks Giving compared to the other seasons.

3.2.4 Word clouds of reviews

. For the predictive task of finding attributes which constitute a superhost, we analyzed the reviews dataset for finding difference between reviews which received high and low ratings. We decided on the distribution of high and low reviews by observing that the mean of superhost reviews was around 97%. Figure 6 and 7 depicts the wordclouds based on bigram frequency for review text rated high and low. The unigram model did not give much information as the same words were repeated for both. From bigram wordcloud for low ratings we can say that dissatisfied customers stressed on host cancellations, shared bathrooms and inconvenience of location. While from the bigram wordcloud for high ratings we can say that the satisfied customers mentioned about great hosts, convenience of location, strong recommendations and possible revisits.

Figure 6: Word cloud for reviews with high ratings (greater that 95.0)

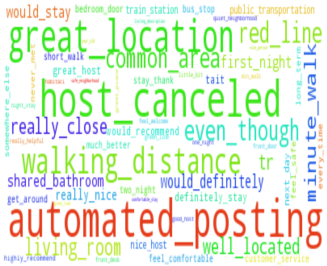

Figure 7: Word cloud for reviews with low ratings (lower than 70.0)

3.2.5 *Number of Hosts vs various features*

. We segregated the hosts into super hosts and normal hosts. Then we made the following plots:

(1) (a)Number of normal hosts versus rating and (b)Number of super hosts versus ratings (Figure 8). We found that the super hosts had very high ratings(mostly 10.0) as compared to the normal hosts which had the ratings in the range 6.0-10.0.
(2) (a)Number of normal hosts versus number of reviews and (b)Number of super hosts versus number of reviews (Figure 9). We found that the guests write high number of reviews for super hosts than for the normal hosts.
(3) (a)Number of normal hosts versus cleanliness rating and (b)Number of super hosts versus cleanliness rating (Figure 10). We found that the super hosts had very high cleanliness ratings(mostly 10.0) as compared to the normal hosts which had the ratings in the range 6.0-10.0.
(4) (a)Number of normal hosts versus rate of acceptance and (b)Number of super hosts versus rate of acceptance (Figure 11).We found that the acceptance rate of super hosts is almost 100% while for the normal hosts the parameter value varies from 70-100%
(5) (a)Number of normal hosts versus location rating and (b)Number of super hosts versus location rating (Figure 12). We found that the super hosts had very high location ratings(mostly 10.0) as compared to the normal hosts which had the ratings in the range 6.0-10.0.
(6) (a)Number of normal hosts versus check-in rating and (b)Number of super hosts versus check-in rating (Figure 13).
(7) (a)Number of normal hosts versus whether host is verified and (b)Number of super hosts versus whether host is verified (Figure 14).We found that most of the super hosts were verified by Airbnb while many of the normal hosts were not verified.
(8) (a)Number of normal hosts versus communication rating and (b)Number of super hosts versus communication rating (Figure 15). We found that the super hosts had very high communication ratings(mostly 10.0) as compared to the normal hosts which had the ratings in the range 6.0-10.0.
(9) (a)Number of normal hosts versus number of amenities provided and (b)Number of super hosts versus number of amenities provided (Figure 16). We found that there was no direct correlation between the number of amenities provided by a host and the host being a super host.

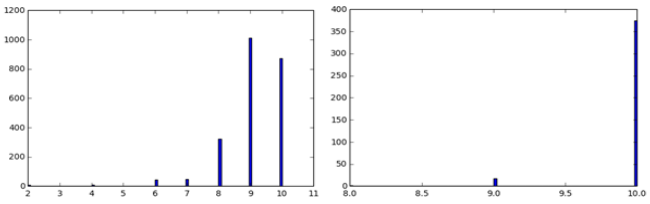

Figure 8: Number of normal hosts vs ratings (left) and Number of super host vs ratings (right)

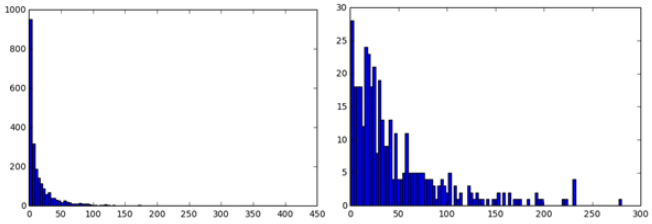

Figure 9: Number of normal hosts vs number of reviews (left) and Number of super host vs number of reviews (right)

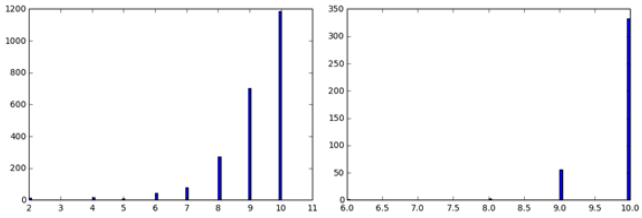

Figure 10: Number of normal hosts vs cleanliness rating (left) and Number of super host vs cleanliness rating (right)

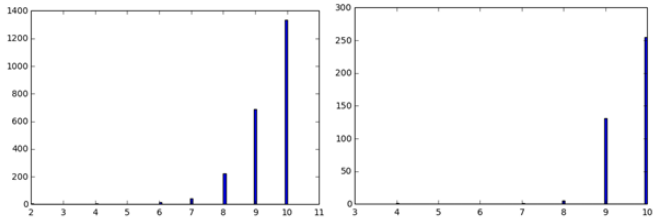

Figure 11: Number of normal hosts vs rate of acceptance (left) and Number of super host vs rate of acceptance (right)

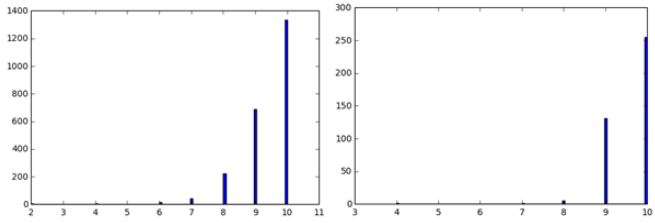

Figure 12: Number of normal hosts vs location rating (left) and Number of super host vs location rating (right)

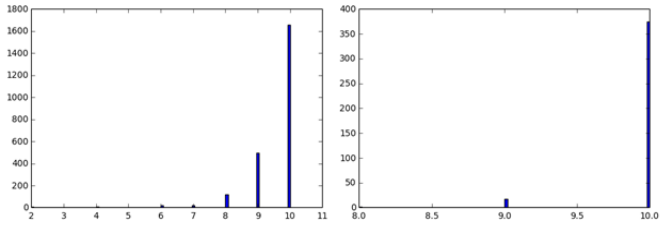

Figure 13: Number of normal hosts vs check-in rating (left) and Number of super host vs check-in rating (right)

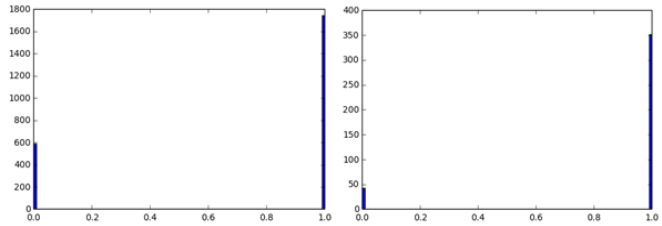

Figure 14: Number of normal hosts vs whether host is verified (left) and Number of super host vs whether host is verified (right)

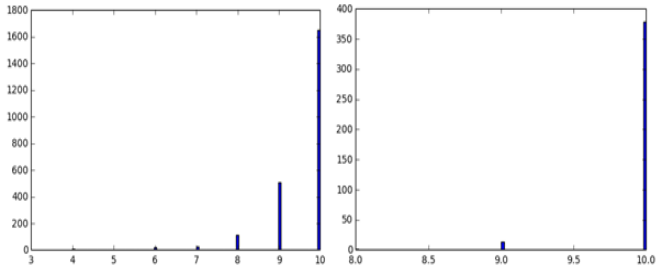

Figure 15: Number of normal hosts vs communication rating (left) and Number of super host vs communication rating (right)

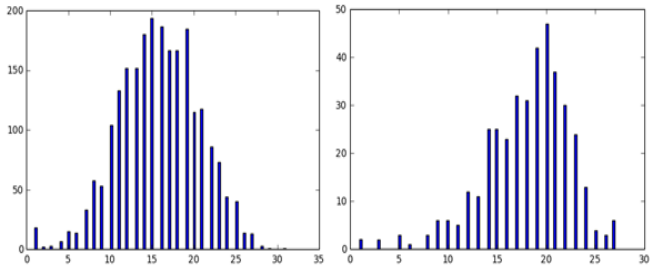

Figure 16: Number of normal hosts vs number of amenities (left) and Number of super host vs number of amenities (right)

4 PREDICTIVE TASKS

We identified two predictive tasks that we could perform using the dataset. One was to model the prices based on the configuration of the property, location and the season of the booking. The second task is to predict the characteristics needed for a host to promote to a super host.

4.1 Modeling price

The dataset provided rich insights into the features that had a strong influence on how the property would be priced. We built a model to predict the prices with respect to these features. A few notable features that could be used are, room type, bedrooms/bathrooms, holiday season price spike and security deposit.

4.2 Modeling super hosts

Airbnb gives certain hosts a special status called fiSuperhostfi badge. These are the hosts, who provide a shining example to other hosts and extraordinary experiences for their guests. A fiSuperhostfi badge provides a number of benefits to both Airbnb and the host. Hence, itfis paramount for both the business of Airbnb and hosts to find out what differentiates a host from a super host.

We did an in-depth study of Airbnb reviews and listings data to gain more insights into this problem. Though there are a set of guidelines provided by Airbnb on how to become a super host, we found there are many more latent factors that goes into determining if a host would become a super host. In our study, we visualized various features to determine their relationship with making the user a super host. At the end of the study, we also ran a SVM classifier and Decision Tree Classifier to determine if the new user is a super host.

4.3 Data cleaning and pre-processing

We did the following data cleaning and pre-processing:

(1) Room Type: 3 types of rooms, Private Room, Entire home / apartment, Shared room. This feature was converted to a categorical feature ranging from 0 to 2.
(2) Property Type: 12 types of property, ex. Townhouse, Villa, Guest-house. This feature was converted to a categorical feature ranging from 0 to 11.
(3) Host Response Rate, Host Acceptance Rate: Remove '%' and stray characters.
(4) Bathrooms: The number of bathrooms that a property has ranges from 0 to 6 bathrooms
(5) Bedrooms: The number of bedrooms that a property has ranges from 0 to 5 bedrooms.
(6) Beds: The number of beds in the property ranges from 0 to 16.
(7) Bed type: 5 types of beds, ex. Pull-out Sofa, Couch etc. This feature was converted to a categorical feature ranging from 0 to 5.
(8) New Year's price spike: 0 or 1 depending on whether there was an increase in price compared to the original listing price.

(9) Thanksgiving price spike: 0 or 1 depending on whether there was an increase in price compared to the original listing price.
(10) Christmas price spike: 0 or 1 depending on whether there was an increase in price compared to the original listing price.
(11) New Year's Eve: 0 or 1 depending on whether there was an increase in price compared to the original listing price.
(12) Price, Security deposit, Cleaning Fee: Removed Null and illegal values Strip '$' and remove ',', and rstrip '.' at the end
(13) Host_neighborhood: There were 50 neighborhoods like "Miami Beach", "Financial District" etc which were converted to a categorical feature ranging from 0 to 49.

4.4 Relevant Features

4.4.1 Relevant features for modeling prices

. From the exploratory analysis of the dataset, we found that the features related to neighborhood, room configurations and holiday season affected the prices the most. The features are as listed below:

(1) Property_type: 12 types of property, ex. Townhouse, Villa, Guesthouse. This feature was converted to a categorical feature ranging from 0 to 11.
(2) Room_type: 3 types of rooms, Private Room, Entire home / apartment, Shared room. This feature was converted to a categorical feature ranging from 0 to 2.
(3) Bathrooms: The number of bathrooms that a property has ranges from 0 to 6 bathrooms.
(4) Bedrooms: The number of bathrooms that a property has ranges from 0 to 5 bedrooms.
(5) Beds: The number of beds in the property ranges from 0 to 16.
(6) Bed_type: 5 types of beds, ex. Pull-out Sofa, Couch etc. This feature was converted to a categorical feature ranging from 0 to 5.
(7) New Year's price spike: 0 or 1 depending on whether there was an increase in price compared to the original listing price.
(8) Thanksgiving price spike: 0 or 1 depending on whether there was an increase in price compared to the original listing price.
(9) Christmas price spike: 0 or 1 depending on whether there was an increase in price compared to the original listing price.
(10) New Year's Eve: 0 or 1 depending on whether there was an increase in price compared to the original listing price.
(11) Security deposit: amount of money paid as deposit.
(12) Host_neighborhood: neighborhood of the property. There were 50 neighborhoods like "Miami Beach", "Financial District" etc which were converted to a categorical feature ranging from 0 to 49.

4.4.2 Modeling super host

. For predicting whether a user can be a super host, we explored various features and found the followings as important:

(1) number_of_reviews: Number of reviews written by the guests for a host.
(2) acceptance_rate: The rate of acceptance of the host. It ranges between 0-100.
(3) response_rate: It captures the responsiveness of the host. The value of this parameter ranges between 0-100.
(4) rating_score_cleanliness: The cleanliness rating of the host.(0-10)
(5) rating_score_location: It captures how good/convenient is the location of the host.(0-10)
(6) rating_score_communication: It captures how responsive the host is.(0-10)
(7) rating_score_value: It captures the overall rating of the host.(0-10)
(8) rating_score_checkin: How convenient is it for the guest for the host to checkin.
(9) is_host_verified: Is the host verified by Airbnb.(0 or 1)

5 MODELS

Here are our experiments with different models for both our predictive tasks.

5.1 Modeling prices

(1) Linear Regression: The model built on linear regression was used as baseline. We encountered over-fitting with this model as there were few entries in the dataset which could possibly be skewed.
(2) Ridge Regression: We needed to improve on the previous model as there was over-fitting. We used Ridge Regression which has L2 regularization in it.
(3) Lasso Regression: In cases where relevant information is smeared over large parts of the spectrum asking the regularization to drop variates will low co-efficient is not a particularly sensible approach. Two parameters which are very well correlated maybe dropped by Lasso Regression.
(4) Random Forest: Random forests are an ensemble learning method for regression, that operate by constructing a multiple decision trees at training time and outputting the mean prediction of the individual trees as the final prediction. Random decision forests prevents decision trees' over-fitting by optimizing the tuning parameters that governs the number of features that are randomly chosen to grow each tree.
(5) Gradient Boosting: Gradient Boosting is an ensemble technique in which weak predictors are combined in building a better model. These weak predictors learn from the misclassifications from the previous steps and better in the next steps by boosting the importance of incorrectly predicted data points. The aggregate forecast got from each of the weak learners will be much better than each of the learners alone.

5.2 Modeling super host

We used two different models to predict if the new user is a super host or not.

5.2.1 SVM classifier

. SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided

by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.[10]

5.2.2 Decision Tree Classifier

. Decision tree learning uses a decision tree as a predictive model which maps observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modeling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a finite set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels[3].
A decision tree is a simple representation for classifying examples. In the task of classifying super hosts, there is a single target feature which is whether the host is a super host or not. Each internal node in a decision tree is labeled with an input feature. Each leaf of the tree is labeled with a class.

6 RESULTS

6.1 Modeling prices

As a first step towards building the models, we built the models based on Linear Regression, Ridge Regression, Lasso Regression, Random forest and Gradient Boosting. These models were initially built without cross-validation.

Mean Absolute Error (MAE) was used as a metric to compare the models. As there could be properties with very high or very low price (outliers), Mean Squared Error (MSE) on our predictions would penalize error rate much more whereas MAE would not.

As shown in the Table 1, Linear Regression and Ridge Regression gave comparable errors. The Ridge regression was done with different regularization parameter and the best value of the parameter was 1. As the errors of the two models are comparable, the regularization added in Ridge did not impact the model a lot in terms of error rate.

Lasso Regression performed the worst on our model as shown in the Table 1. This model would have dropped many features that could add value to the model as the L1 norm used in Lasso drops the features with lesser significance. Although some features may not be as significant as the others, they added value to the model and it is not suitable to drop them off and hence Lasso Regression wasnfit useful.

The ensemble techniques like Random Forest and Gradient Boosting performed well and were comparable in performance as shown in Table 1. As seen from the literature review, these methods which are state-of-the-art performed well on our predictive task as well.

As a next step, we introduced 10 fold cross-validation while building the models. The comparison between the models before and after cross-validation are shown in the . The accuracy of each of the models improved after cross-validation than before. The best value of regularization parameter for Ridge Regression was 1 and it produced an accuracy of 0.585. The Random Forest Regression with 80% of the features as maximum features in each tree, depth of the tree as 10 and number of samples per leaf as 2, the accuracy after cross-validation improved by more than 4%.

As seen from Table 2, Gradient boosting with 80% of the features as maximum features in each tree, depth of the tree as 5 and learning rate as 0.1, the accuracy after cross-validation improved by 2%. Model based on Gradient Boosting performed better than any other model with an accuracy of 72.21%

Table 1: Training and test error while modeling price prediction

Model	Training error	Test error
Linear Regression	0.3046	0.3173
Ridge Regression	0.3047	0.3174
Lasso Regression	0.5039	0.5074
Random Forest Regressor	0.1394	0.2760
Gradient Boosting Regressor	0.2379	0.2751

Table 2: Accuracy while modeling price prediction with cross validation

Model	Before validation	After Validation
Ridge Regression	0.5853	0.5867
Random Forest Regressor	0.6538	0.69145
Gradient Boosting Regressor	0.7018	0.7221

6.2 Modeling super host

The problem of modeling a super host is rather challenging. The major challenge being that there are not many distinguishing parameters between the super hosts and the normal hosts. To deal with this challenge, we performed exploratory analysis and found the various features that were distributed differently between the two type of hosts. We also encountered the problem of imbalanced dataset as the number of super hosts was just 16% of the total hosts.

To deal with the problem of imbalanced dataset we first experimented with using SVM classifier. We used different penalties for the negative and positive cases so that the decision boundary is not biased towards minority samples.

We were able to get $\tilde{2}0\%$ error for both training and test set using SVM classifier. Though precision and recall are better measure of accuracy for unbalanced dataset. Hence, we calculated both precision and recall. We found that the SVM model gave very high recall for both the super host and normal hosts, While the precision for the super host class was just $\tilde{4}1\%$ where as for the normal host it was $\tilde{9}7\%$.

To improve on our precision, we experimented with decision trees which are not only effective for classification tasks but also require very low tuning of parameters for their working. We were able to reach a precision of $\tilde{7}1\%$ using decision trees classifier with a max depth of 5 for the decision tree.

With regard to the features that were helpful in predicting whether a host is a super host or not. We found that there a number of features that come into play in determining the badge of a host. Airbnb lists a couple of attributes that are paramount for becoming a super host. Through our analysis we validated these attributes

and found several other latent attributes that determines whether a host would become a super host. The important latent features being:

- Number of reviews written by the guest for a host
- Level of cleanliness of the host
- The responsiveness of the host
- If the host is verified by Airbnb
- How good is the location of the listing of the host
- How comfortable is the checkin procedure of the host

We also performed text analysis of the reviews given by the guests for hosts to gain insights into other attributes that differentiate a super host from the host which could not be otherwise captured by numerical scores. For doing that, we formed bigram word clouds for high and low rated reviews and observed the following:

The low rated reviews had frequently occurring bigrams like: Host Cancelled, Shared Bathroom, Never met, Even though, etc. The highly rated reviews had frequently occurring bigrams like: Great host, highly recommend, would definitely, great location etc. From the above, we inferred that cancellations, location and interaction of host with guest, were the major factors that influenced ratings. Table 3

Table 3: Training and test error while modeling super hosts

Model	Training error	Test error
SVM Classifier	0.206	0.2053
Decision Tree Classifier	0.1027	0.09413

Table 4

Table 4: Statistics while modeling super hosts

Host type	Model	Precision	Recall
Non-SuperHost	SVM Classifier	0.97158082	0.78254649
SuperHost	SVM Classifier	0.40392157	0.86554622
Non-SuperHost	Decision Tree Classifier	0.93074792	0.96137339
SuperHost	Decision Tree Classifier	0.71875	0.57983193

7 CONCLUSIONS

In this project, we performed two predictive tasks. First we studied the features that influenced price of the listings. Second, we studied what attributes make a super host.

For modeling prices, we studied various models. Ensemble techniques like Random Forest and Gradient Boosting proved to be better than most other models. We also performed sentiment analysis on the review text provided by the customers to assess if it added value to the model. However, the features based on sentiment analysis did not influence the model.

To derive characteristics associated with a super host, we performed exploratory data analysis to find that cleanliness, location, communication, number of reviews were major characteristics through which we can distinguish a super host. We verified our analysis, by building classifiers such as SVM and decision trees, and found our results were in line with the features we derived from our analysis.

The attempt to extract topics from review comments which differentiate between normal/super host did not work as expected. This is because both the normal and super host shared the same sentiments/word vector distribution when it came to reviews. However, when we analyzed good and bad reviews, we captured a fair picture of factors which influenced review ratings.

8 REFERENCES

[1] https://www.kaggle.com/airbnb/boston
[2] https://en.wikipedia.org/wiki/Airbnb
[3] https://en.wikipedia.org/wiki/Decision_tree_learning
[4] http://insideairbnb.com/get-the-data.html
[5] https://www.kaggle.com/residentmario/d/airbnb/boston/modeling-prices/
[6] Winkler, RL. and Makridakis, S. (1983). The Combination of Forecasts. J. R. Statis. Soc. A. 146(2), 150-157.
[7] Makridakis, S. and Winkler, R.L. (1983). Averages of Forecasts: Some Empirical Results.Management Science, 29(9) 987-996.
[8] https://www.cs.cornell.edu/home/cardie/papers/masa-sentire-2011.pdf
[9] http://res.cloudinary.com/general-assembly-profiles/image/upload/v1474479496/geovpl78jbsx7dzqycde.pdf
[10]https://en.wikipedia.org/wiki/Support_vector_machine