,Jieun Hwang, Samir Hatem, Xiaorou Liang, Yejin You
December 14th, 2018
IEOR 142

Prediction of Airbnb Listing Price in San Francisco

Introduction

Airbnb is a web and mobile-based service that allows its users to arrange or offer lodging. For such service, monitoring and understanding the underlying pricing dynamics of the market is crucial for both hosts and guests. As the number of Airbnb users continues to grow, hosts may find it difficult to properly price their property.

The purpose of this report is to explore possible models for predicting Airbnb listing prices per night in the San Francisco area. Many factors, such as number of bedrooms, location, and host response rate were considered in model formulation. We wanted to build a model that would allow hosts to appropriately price their listings. A successful model would prevent the hosts from both overpricing and underpricing the listings, thus optimizing the host's profitability while maintaining affordability to the users. With the model, the users would be able to understand what features of an Airbnb listing are most significant in determining the price.

Methodology

We used Airbnb dataset[1] from Open Data Soft, which had 89 features and 8680 observations. We first looked through the 'Listing Url' column, which included many expired URLs. We then utilized web scraping to filter out the expired URLs, which left us with 2770 observations. Next, we webscraped the Airbnb listing price per night. We cleaned the data by dropping the columns that had either a significant number of missing values, many repeated values, or the same information as other columns. We dealt with missing values by either replacing them with the average value of the column or dropping them when they contained strings since missing string values are difficult to replace.

We then combined crime data[2] scraped from Areavibes with the Airbnb data. In the crime dataset, SF was zoned into multiple regions, with each zone scored based on the crime rating. We expected that the crime rate would have an effect on the listing price. So we used 'Neighbourhood Cleansed' column, which contains SF regions, and converted them into numerical scores from 0 to 11 ('F': 0, 'D-': 1, 'D': 2, 'D+': 3, 'C-': 4, 'C': 5, 'C+': 6, 'B-': 7, 'B': 8, 'B+':9, 'A-': 10, 'A': 11). For further details on data processing , see the Appendix. After cleaning the data and combining the two datasets, we had the final data set with 43 columns including the response variable, 'Price,' and 2289 observations (10 columns: text and 33 columns: non-text). We refer to the 33 non-text columns as numeric columns in the rest of this report. See the Appendix for the data dictionary containing the list of features and their descriptions of the cleaned dataset.

The cleaned dataset contains 10 columns with text: Summary, Space, Description, Notes, Neighborhood Overview, Transit, Access, Interaction, House Rules, Host About. To avoid redundancy and enhance efficiency, 3 columns that did not overlap with other columns were chosen for bag of words analysis. Among

[1]https://public.opendatasoft.com/explore/dataset/airbnb-listings/table/?disjunctive.host_verifications&disjunctive.amenities&disjunctive.features&refine.city=San+Francisco
[2]https://www.areavibes.com/san+francisco-ca/mission/livability/?ll=37.76009+-122.4145

the four columns Summary, Space, Description, and Notes, only Summary is chosen because all four columns contain similar information. The column Neighborhood Overview, which contains information about the subjective safety information of the neighborhood provided by the host can be disregarded from bag of words because a more objective crime ratings feature was added to the dataset. The column Transit contains subjective information, which can be replaced by the more objective feature Review Scores Location, which contains ratings on the overall convenience of the listing by previous guests. The text column Access contains similar information as the column Accommodates, so Access can be removed to avoid redundancy. Similarly, since the column Interaction contains comparable information to the Review Score Communication column, it can be disregarded as well. Columns House Rules and Host About do not overlap with other columns so they are kept for bag of words. Thus, columns Summary, House Rules, and Host About are the 3 columns chosen for bag of words.

To create the bag of words, each column is converted into an individual corpus and processed separately with all terms turned into lower case, and punctuations and stopped words removed. Then, the texts are stemmed and a sparsity of 0.90 is used for each column so that terms are kept only if they appear in 10% or more of the listings for each column. This process resulted in a total of 128 terms (independent variables) from all 3 columns. Then, these 128 terms are combined with the other 33 non-text columns for model building and further analysis. By doing so, the final cleaned dataset has a total of 161 columns including the response variable Price.

Initial Attempts

We initially attempted to analyze and run models on our text and numerical data separately. The reason was that we mistakenly thought that applying bag of words would result in many more features than our numerical features. Through combining them we in turn assume that the weight, or the importance of the bagged word features is the same as all the numerical features. However , after consideration, we realized that the relative importance of the features will be naturally displayed in the outcomes of our models. From all the models ran on the data, Random Forest performed the best. Similarly, the codes and results of each model using only bag of words from text columns as predictors or only the numerical columns can be found in the Appendix.

Model Formulation

After completing the initial data cleaning and processing, we randomly split the data into 70% training data (1602 listings) and 30% test data (687 listings). Several models were then assessed in the process of model formulation:

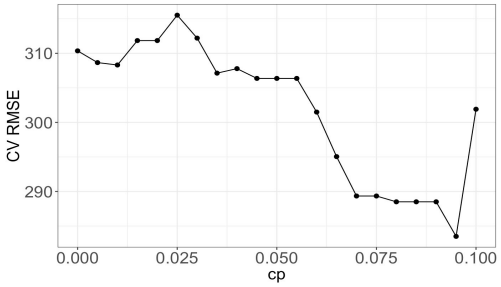
1. Baseline Model
   The baseline model predicts simplistically using the average price of the training set listings. The prediction, or the average price, was $215.36.
   $$\bar{y} = \frac{y_1+y_2+\ldots+y_n}{y_n} = \frac{y_1+y_2+\ldots+y_{1602}}{y_{1602}} = 215.3558$$
2. Backward Stepwise Regression Model
   The linear regression model predicts the price, which is modeled as a linear function of the independent variables. Backward stepwise regression was used for feature selection to account for multicollinearity and statistical insignificance. See Appendix for the selected features.
   The form of the resulting linear regression model should be:

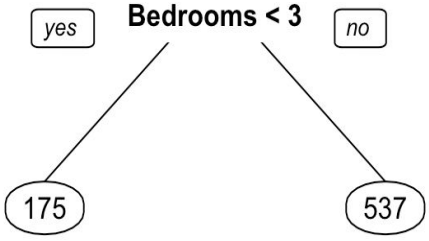
$$Price = \beta_0 + \beta_1 Bedrooms + \beta_2 Bathrooms + \beta_3 Availability.365 + \beta_4 Accommodates + \ldots$$

From analyzing the output, we can see that the most important features according this model are host response time, number of bedrooms and bathrooms, and ratings. Also, some important terms identified by the model are “san”, “10pm”, and “rule”.

3. Cross-Validated CART Model

   The CART model builds a tree by splitting on the independent variables. The complexity parameter (cp) used to prune the CART tree was chosen by 5-fold cross-validation using the metric RMSE. Every 0.005 between 0 and 0.1 was tested for the cp values. Observing the plot, it is evident that a cp value of 0.095 should be selected because it results in the lowest RMSE.

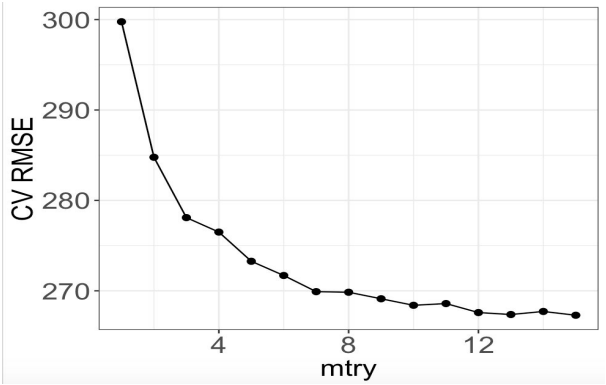


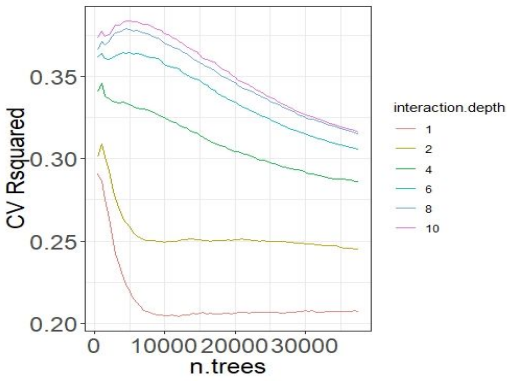


   Running this final model with cp = .095, we get that the number of bedrooms is the most important predictor in that listings with more bedrooms have higher prices. This simple structure indicates that CART is not a good model.

4. Random Forest Model

   The random forest model was trained with the mtry(m) value chosen from 1 to 15 through 5-fold cross-validation. From the plot, it is evident that an mtry of 15 gives the minimum RMSE.





Random Forest Model Boosting Model

5. Boosting Model

   The boosting model was trained with interaction.depth (maximum number of splits) from possible values of 1,2,4,6,8,10 and n.trees (total number of trees) from possible values of 500 to 37500 selected through 5-fold cross-validation. The idea is to select the parameters that give the highest R-squared. Thus, the chosen parameters were 3000 for n.trees and 10 for interaction.depth.

For further details, see Appendix for the model formulation code.

Results

The following table shows the performance metrics of each model:

Merged dataset metrics:

Model	MAE	RMSE	OSR2
Baseline	115.2296	219.1013	0
Stepwise Regression	108.5496	232.2336	-0.9163871
CART	101.7905	182.0866	0.3092725
Random Forest	77.26419	163.4805	0.4432213
Boosting	227.9264	261.9051	-1.437367

95% CI for merged dataset metrics:

Model	MAE	RMSE	OSR2
Stepwise Regression	(92.3, 122.3)	(151.3, 325.0)	(-2.1439, 0.8429)
CART	(89.9, 112.3)	(130.5, 224.9)	(0.2156, 0.3895)
Random Forest	(65.54, 87.08)	(101.7, 213.3)	(0.2958, 0.5695)
Boosting	(218.2, 237.5)	(250.4, 273.4)	(-1.922, -0.727)

Looking at the metrics for the models and their bootstraps, we chose the cross-validated random forest model. Some of the most important features that our model showed included the Host response time, number of bedrooms and bathrooms, and the listing's cancellation policy.

Impact/further explorations

We wanted to add value to the marketplace by creating a systematic methodology by which both host profit and the value for the user could be maximized. Although our model had seen some success especially when utilizing cross-validated random forest it's still far from perfect and hence has room for further improvement. One feature that would potentially result in significant improvement would be accounting for seasonality where obtaining the change in price listings throughout the year would allow our model to become much more robust. The second improvement would be to scale our model such that it is not limited to the San Francisco region.

Appendix

Data Dictionary (Features and Descriptions)

Column Name	**Column Content**
Summary	Overall summary of the listing. In text format.
Space	More information about the listing, most specifically the physical attributes of the place. In text format.
Description	Detailed information about the listing. In text format.
Notes	Additional notes/ information provided by the host about the listing. In text format.
Neighborhood Overview	Information about the subjective safety information of the neighborhood provided by the host. In text format.
Transit	Subjective information provided by the host regarding the availability and proximity of the listing to public transits. In text format.
Access	Information about areas in the listing the guest has access to. In text format.
Interaction	Types of interaction offered by host. In text format.
House Rules	Rules guests must abide to. In text format.
Host Location	Where the host is. Categorical. If the host resides in SF, then takes a value of 1. If not, takes a value of 0
Host About	Information about the host. In text format.
Host Response Time	How long it takes the host to respond to rental inquiries. Categorical and takes possible categories of : 'within an hour', 'within a day', 'within a few hours', or 'a few days or more'
Host Verifications	Number of verifications. Numerical
Neighbourhood Cleansed	Neighbourhood in which the listing is located. Categorical and takes possible categories of : 'Castro/Upper Market', 'Bernal Heights', 'South of Market', 'Potrero Hill', 'Inner Richmond', 'Marina', 'Bayview','Western Addition', 'Seacliff', 'Haight Ashbury', 'Russian Hill', 'Noe Valley', 'Inner Sunset', 'Excelsior', 'Outer Mission', 'Downtown/Civic Center', 'Outer Richmond', 'Financial District', 'Ocean View', 'Mission', 'West of Twin Peaks', 'Twin Peaks', 'Pacific Heights', 'Outer Sunset', 'North Beach', 'Diamond Heights', 'Parkside', 'Presidio Heights', 'Crocker Amazon', 'Nob Hill', 'Glen Park', 'Visitacion Valley', 'Chinatown', 'Golden Gate Park', 'Lakeshore', or 'Presidio'.
Property Type	Property type of the listing. Categorical and takes possible categories of 'Apartment', 'Condominium', 'Bed & Breakfast', 'House', 'Townhouse',

	'Other', 'Loft', 'Cabin', 'Bungalow', 'Guesthouse', 'Timeshare', 'Boutique hotel', 'Dorm', 'Treehouse', or 'Lighthouse'
Room Type	Nature of the listing. Categorical and takes possible possible categories of 'Private room', 'Entire home/apt', or 'Shared room'
Accommodates	How many people (guests + visitors) the listing can accommodate / fit. Numerical.
Bathrooms	Num of bathroom. Numerical
Bedrooms	Num of bedroom. Numerical
Beds	Num of bed. Numerical
Bed Type	Type of bed offered in the listing. Possible categories of 'Real Bed', 'Futon', 'Airbed', 'Pull-out Sofa', or 'Couch'
Price	Current listing price. Numerical Nov, 16 2018
Guests Included	How many guests the listing can house (visitors not included). Numerical
Minimum Nights	Minimum number of nights guests must book for. Numerical
Maximum Nights	Maximum number of nights guests can book for. Numerical
Availability 365	How many days the listing is available for in a year. Numerical
Number Reviews	Number of reviews. Numerical
Review Scores Ratings	Average score of ratings from 1- 10. Numerical
Review Scores Accuracy	Averaged ratings of accuracy of the post from 1-10. Numerical
Review Scores Cleanliness	Average rating of cleanliness of listing from 1- 10. Numerical
Review Scores Checkin	Average rating of the check-in process from 1- 10. Numerical
Review Scores Communication	Average rating of communication of the host from 1- 10. Numerical
Review Scores Location	How good or convenient was the listing location from 1- 10. Numerical
Review Scores Value	The overall score of the experience the guests had from 1- 10. Numerical
Cancellation Policy	Cancellation policy. Categorical and possible categories are 'strict', 'flexible', 'moderate', 'super_strict_30', 'super_strict_60'
Calculated Host Listings Count	How many listings by the same host. Numerical
Reviews per Month	Number of reviews per month. Numerical
duration_rev	the time difference between the first and the last review. Numerical

Time Since Last Review	Time passed since the last review using day of web-scraping as reference. Numerical
Number of Amenities	Total number of amenities per listing. Numerical
Host Duration	How long the host has been on Airbnb. Numerical
Crime	Alphanumeric rating of the neighborhood's crime score. Numerical from 0-11. ('F': 0, 'D-': 1, 'D': 2, 'D+': 3, 'C-': 4, 'C': 5, 'C+': 6, 'B-': 7, 'B': 8, 'B+':9, 'A-': 10, 'A': 11). F is the lowest rating while A is the highest.

Data Process Detail

1. Dropped the columns
    - Column has almost all missing data points
      ID, Scrape ID, Picture Url, Experiences Offered, Host ID, Host Response Rate, Host Acceptance Rate , Neighbourhood Group Cleansed, Cleaning Fee, Extra People, Has Availability, Geolocation, Features, Longitude, Latitude, Weekly Price, Monthly Price,Security Deposit, Square Feet
    - Column has all same values or URL
      Unnamed: 0.1, Listing Url, Name, Host URL, Host Name, Unnamed: 0, Host Thumbnail Url, Calendar last Scraped, License, Jurisdiction Names, Thumbnail Url, Medium Url, Host Picture Url, XL Picture Url, Last Scraped, Country, City, State, Zipcode, expire, Market, Country Code, Smart Location
    - Replaced the column
        - Dropped Host Total Listings Count and Host Listing Count because they contain same information so we used Calculated host listings count instead
        - Dropped Availability 30, Availability 60, Availability 90 because we used Availability 365 instead
        - Dropped Neighborhood and Street because we used Neighbourhood Cleansed instead
        - Dropped Host Neighborhood because we used host location instead

2 . Working with Missing data points
    - Dropped missing values because they were string columns and had NaN values less than 10% of total number of observations
      Host Since, First Review, Host Location, Last Review, Bathrooms, Beds, Host Response Time, Host Neighbourhood
    - Replaced missing values with average of columns. These columns had NaN Values less than 8
      Review Scores Rating, Review Scores Accuracy, Review Scores Cleanliness, Review Scores Checkin, Review Scores Communication, Review Scores Location, Review Scores Value

3. Cleaning Columns
    - Host location: When host location is San Francisco, return 1 otherwise 0.
    - Maximum Nights: removed some outliers (10000, 999999, 2147483647)

- First Reviews and Last Reviews: 'Duration_rev' (the time difference between the first and the last review), 'Time since last review' (time difference between day of acces
- Amenities: changed list of elements to total number of amenities per listing
- Host Since: changed to fractional year
- Host Verifications: changed list of elements to total number of Host verifications per listing

Model Parameters for Text Analysis

1. Backward Stepwise Regression

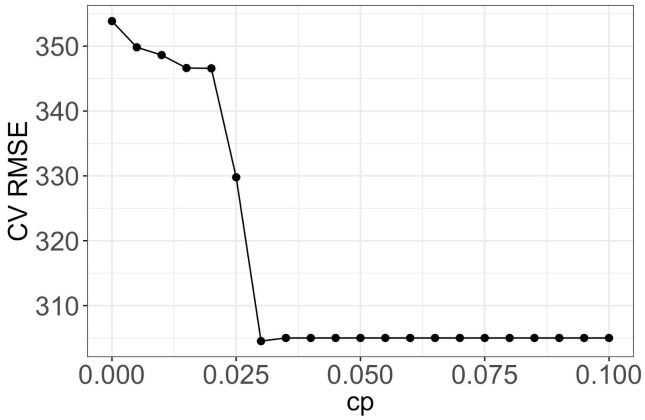
   The following is the R output of the model is as follows:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  216.335     16.640  13.001  < 2e-16 ***
home          24.001     15.525   1.546 0.122302
floor         33.129     21.311   1.555 0.120248
privat       -59.453     14.034  -4.236  2.4e-05 ***
enjoy         58.858     25.025   2.352 0.018795 *
share        -68.461     24.960  -2.743 0.006159 **
victorian     52.865     26.875   1.967 0.049346 *
mission      -35.230     20.520  -1.717 0.086192 .
walk          47.408     16.115   2.942 0.003310 **
public       -38.957     24.648  -1.581 0.114186
access       -33.944     21.934  -1.548 0.121920
clean        -25.232     16.493  -1.530 0.126258
make         -38.988     23.018  -1.694 0.090502 .
parti         41.591     16.774   2.479 0.013262 *
check        -24.163     13.120  -1.842 0.065712 .
use           38.176     12.948   2.948 0.003240 **
quiet.1      -49.983     19.555  -2.556 0.010678 *
X10pm        101.980     27.437   3.717 0.000209 ***
home.1        26.112     13.475   1.938 0.052821 .
rule          32.043     19.484   1.645 0.100248
room.1       -51.121     18.263  -2.799 0.005185 **
citi.1       -25.448     15.483  -1.644 0.100447
francisco.1  -99.737     41.860  -2.383 0.017306 *
love.1       -14.631      9.072  -1.613 0.106981
san.1        131.121     40.660   3.225 0.001286 **
can.1        -36.171     19.372  -1.867 0.062060 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 347.3 on 1594 degrees of freedom
Multiple R-squared:  0.08161,   Adjusted R-squared:  0.06721
F-statistic: 5.666 on 25 and 1594 DF,  p-value: < 2.2e-16
```

   It is evident that from all the terms the algorithm kept, only terms [private, enjoy, share, victorian, walk, parti, use, quiet, 10pm, room, francisco, san] are statistically significant with a p-value of <= 0.05.

2. Cross-Validated CART Model

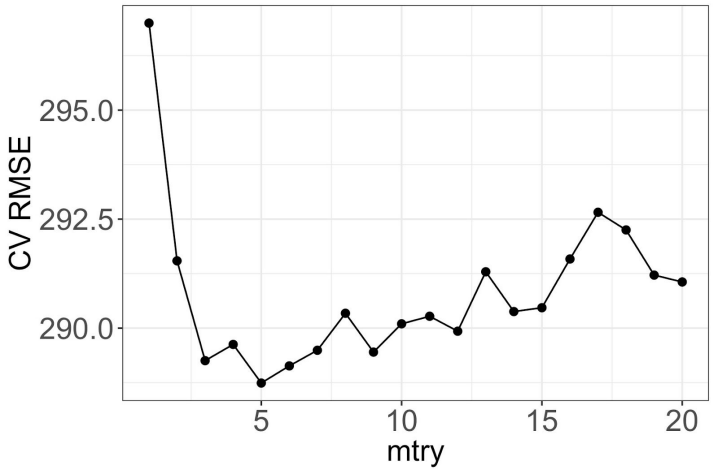
   The CART model builds a tree by splitting on the independent variables. The complexity parameter (cp) used to prune the CART tree was chosen by 5-fold cross-validation using the metric RMSE. Every 0.005 between 0 and 0.1 was tested for the cp values and the cp value resulting in the lowest RMSE is chosen. By plotting the complexity parameter against the RMSE, it is evident that a cp value of 0.03 should be chosen because it results in the lowest RMSE.



Running the final model with a chosen cp of 0.03 produces the following result. It is clear that the CART model is no better than the baseline model since like the baseline, CART only predicts the average of the training set (215) , hence the 0 R-squared we got.

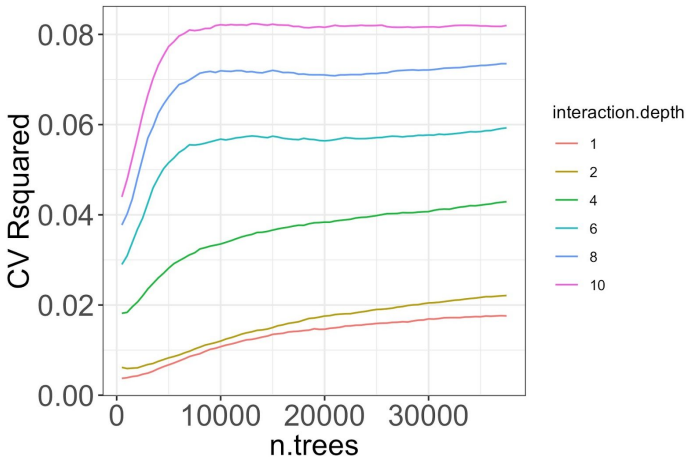
3. Random Forest

   The random forest model was trained with the mtry (m) value chosen through 5-fold cross-validation from possible values from 1 to 20. The mtry value giving the lowest RMSE is chosen. By plotting mtry against RMSE, one can see that a mtry value of 5 provides the minimum RMSE. So, the final model uses a mtry of 5:



4. Boosting

   The boosting model was trained with interaction.depth (maximum number of splits) and n.trees (total number of trees) parameters selected through 5-fold cross-validation. The interaction.depth and n.trees parameters that give the highest R-squared value are selected. From the output shown below, it is clear that an interaction.depth of 10 outperforms other values. Also, a value of 10000 should be chosen for n.trees because as the number of trees increase from 0 to 10000, R-squared also increases but as the number of trees increase beyond 10000, R-squared stays flat. So, 10000 trees are chosen:



5. Results

The following table summarizes the performance metrics for all 5 models trained using only the text data:

Model	MAE	RMSE	OSR2
Baseline	115.9681	219.1013	0
Stepwise Regression	121.9779	176.3504	-0.1156362
CART	115.9681	166.961	0
Random Forest	102.4672	150.2023	0.1906752
Boosting	107.3319	160.9937	0.07020379

It is evident that Random Forest outperforms all other models in that it has the lowest MAE and RMSE and the highest OSR2.

Model Parameters for Numerical Analysis

1. Backward Stepwise Regression

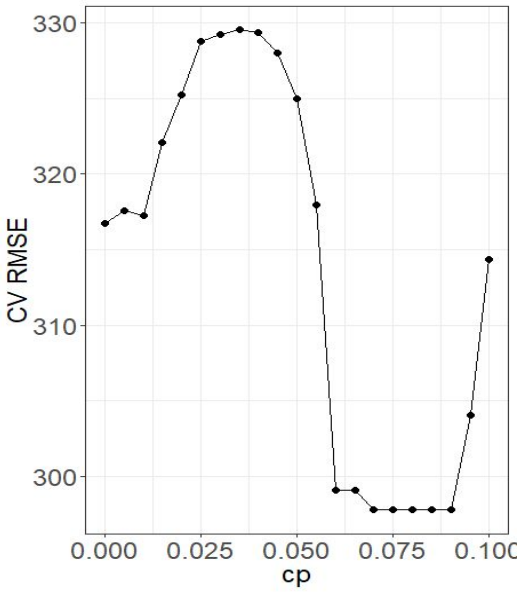
   The following is the R output of the model is as follows:

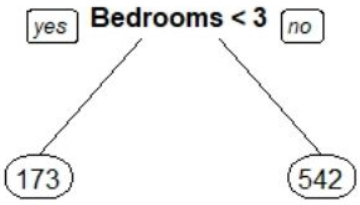
```
Call:
lm(formula = Price ~ Host.Response.Time + Host.Location + Host.Verifications +
    Room.Type + Accommodates + Bathrooms + Bedrooms + Minimum.Nights +
    Maximum.Nights + Availability.365 + Number.of.Reviews + Review.Scores.Rating +
    Review.Scores.Cleanliness + Review.Scores.Location + Review.Scores.Value +
    Cancellation.Policy + Number.of.Amenities, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-2297.1   -66.6   -15.6    36.6  7749.3

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                         1.221e+03  2.247e+02   5.433  6.4e-08 ***
Host.Response.Timewithin a day     -2.086e+03  1.523e+02 -13.695  < 2e-16 ***
Host.Response.Timewithin a few hours -2.072e+03  1.513e+02 -13.697  < 2e-16 ***
Host.Response.Timewithin an hour   -2.048e+03  1.510e+02 -13.563  < 2e-16 ***
Host.Location                      -6.327e+01  2.612e+01  -2.423 0.015522 *
Host.Verifications                  1.335e+01  6.944e+00   1.923 0.054622 .
Room.TypePrivate room              -4.163e+01  1.867e+01  -2.230 0.025903 *
Room.TypeShared room               -1.399e+02  5.673e+01  -2.467 0.013739 *
Accommodates                        1.890e+01  6.418e+00   2.945 0.003275 **
Bathrooms                           5.301e+01  1.580e+01   3.355 0.000812 ***
Bedrooms                            7.007e+01  1.362e+01   5.146  3.0e-07 ***
Minimum.Nights                     -1.473e+00  8.037e-01  -1.833 0.066947 .
Maximum.Nights                     -2.158e-02  1.467e-02  -1.471 0.141572
Availability.365                    1.409e-01  6.331e-02   2.226 0.026173 *
Number.of.Reviews                  -3.719e-01  1.336e-01  -2.783 0.005449 **
Review.Scores.Rating                6.353e+00  2.411e+00   2.635 0.008504 **
Review.Scores.Cleanliness           3.921e+01  1.731e+01   2.265 0.023643 *
Review.Scores.Location              4.099e+01    1.389e+01 2.952 0.003203 **
Review.Scores.Value                -4.850e+01  1.656e+01  -2.930 0.003442 **
Cancellation.Policymoderate         1.139e+01  2.463e+01   0.462 0.643947
Cancellation.Policystrict           1.103e+01  2.404e+01   0.459 0.646373
Cancellation.Policysuper_strict_30  2.326e+01  1.119e+02   0.208 0.835370
Cancellation.Policysuper_strict_60  6.523e+02  1.780e+02   3.664 0.000256 ***
Number.of.Amenities                -4.267e+00  1.681e+00  -2.538 0.011234 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 298.3 on 1578 degrees of freedom
Multiple R-squared:  0.2796,    Adjusted R-squared:  0.2691
F-statistic: 26.63 on 23 and 1578 DF,  p-value: < 2.2e-16
```

Observing the above output, only Host Verification, Minimum Nights, Maximum Nights and Cancellation Policy that are not super strict are insignificant with p-values >= 0.05.

2. CART Model

By plotting the complexity parameter against the RMSE, it is evident that a cp value of 0.095 should be chosen because it results in the lowest RMSE.

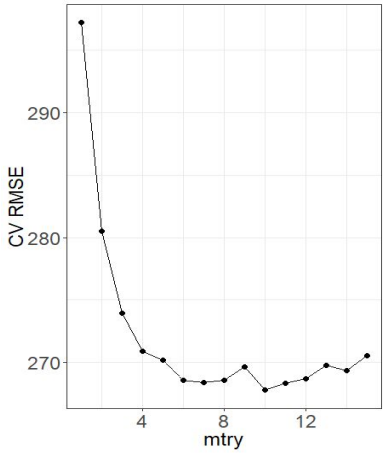




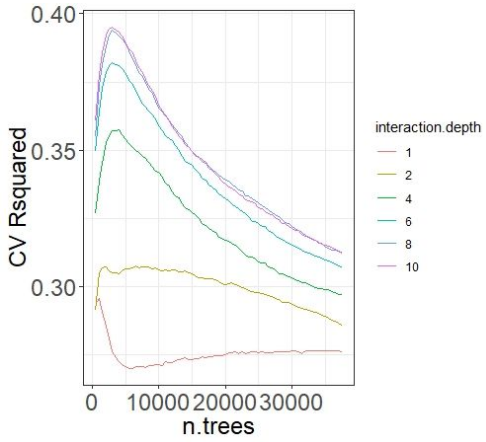
Running the final model with a chosen cp of 0.095 produces the following result. It is clear that the CART model is not a good model since it only indicates one feature Bedrooms as the only important predictor in that listings with more rooms are more expensive:

3. Random Forest

   The random forest model was trained with the mtry (m) value chosen through 5-fold cross-validation from possible values from 1 to 15. The mtry value giving the lowest RMSE is chosen. By plotting mtry against RMSE, one can see that a mtry value of 10 provides the minimum RMSE. So, the final model uses a mtry of 10:



4. Boosting



The boosting model was trained with interaction.depth (maximum number of splits) and n.trees (total number of trees) parameters selected through 5-fold cross-validation. The interaction.depth and n.trees parameters that give the highest R-squared value are selected. From the plot shown above, it is clear that an interaction.depth of 10 outperforms other values and around 3000 tree also outperforms other tree values.

5. Results

Numerical analysis metrics:

Model	MAE	RMSE	OSR2
Baseline	115.2296	219.1013	0
Stepwise Regression	93.0127850	188.7569135	0.2577394
CART	101.7904971	182.0866194	0.309272589
Random Forest	77.1334718	168.7546723	0.4067165
Boosting	296.2199	326.2602	-2.782344

95% Confidence Interval for Metrics:

Model	MAE	RMSE	OSR2
Stepwise Regression	(79.65, 104.38)	(128.1, 247.0)	(0.0148, 0.6043)
CART	(89.9, 112.2)	(130.1, 225.8)	(0.2154, 0.3892)
Random Forest	(65.26, 87.50)	(112.9, 216.5)	(0.2374, 0.5601)
Boosting	(286.0, 306.4)	(315.3, 337.4)	(-3.609, -1.566)

From the the first table, cross-validated random forest outperforms all other models with the highest R-squared and lowest MAE and RMSE. The second table shows the 95% confidence intervals for the corresponding metrics.

Combined Dataset (text and numerical columns) Regression Model Output

```
Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                       9.560e+02  2.333e+02   4.098 4.39e-05 ***
Host.Response.Timewithin a day   -1.970e+03  1.513e+02 -13.025  < 2e-16 ***
Host.Response.Timewithin a few hours -1.947e+03  1.504e+02 -12.944  < 2e-16 ***
Host.Response.Timewithin an hour -1.925e+03  1.500e+02 -12.833  < 2e-16 ***
Host.Location                    -6.318e+01  2.628e+01  -2.405 0.016304 *
Host.Verifications                1.289e+01  6.952e+00   1.854 0.063862 .
Room.TypePrivate room            -2.519e+01  1.874e+01  -1.344 0.179070
Room.TypeShared room             -1.452e+02  5.857e+01  -2.480 0.013256 *
Accommodates                      2.161e+01  6.414e+00   3.370 0.000771 ***
Bathrooms                         5.315e+01  1.571e+01   3.382 0.000737 ***
Bedrooms                          6.721e+01  1.373e+01   4.896 1.08e-06 ***
Minimum.Nights                   -2.072e+00  8.110e-01  -2.554 0.010735 *
Availability.365                  1.142e-01  6.271e-02   1.821 0.068731 .
Number.of.Reviews                -4.004e-01  1.363e-01  -2.936 0.003369 **
Review.Scores.Rating              6.746e+00  2.382e+00   2.832 0.004690 **
Review.Scores.Cleanliness         4.001e+01  1.719e+01   2.328 0.020043 *
Review.Scores.Location            4.458e+01  1.391e+01   3.204 0.001383 **
Review.Scores.Value              -5.294e+01  1.640e+01  -3.228 0.001272 **
Cancellation.Policymoderate       1.370e+01  2.435e+01   0.563 0.573713
Cancellation.Policystrict         9.973e+00  2.408e+01   0.414 0.678846
Cancellation.Policysuper_strict_30 -6.216e+01  1.202e+02  -0.517 0.605116
Cancellation.Policysuper_strict_60  5.762e+02  1.768e+02   3.258 0.001144 **
Time.since.last.review            3.769e+01  2.622e+01   1.437 0.150803
Number.of.Amenities              -3.492e+00  1.698e+00  -2.057 0.039885 *
full                             -3.034e+01  1.970e+01  -1.540 0.123828
live                             -3.536e+01  1.915e+01  -1.846 0.065076 .
enjoy                             4.230e+01  2.073e+01   2.040 0.041473 *
park                             -1.761e+01  1.081e+01  -1.628 0.103623
downtown                          3.040e+01  1.968e+01   1.544 0.122701
mission                          -2.651e+01  1.789e+01  -1.482 0.138555
walk                              3.577e+01  1.612e+01   2.219 0.026620 *
place                             3.202e+01  1.429e+01   2.240 0.025236 *
love                             -4.368e+01  2.267e+01  -1.927 0.054162 .
distanc                          -5.900e+01  2.754e+01  -2.143 0.032299 *
clean                            -2.367e+01  1.433e+01  -1.652 0.098756 .
parti                             2.463e+01  1.526e+01   1.614 0.106742
check                            -2.203e+01  1.295e+01  -1.701 0.089050 .
hous.1                           -2.399e+01  1.040e+01  -2.306 0.021256 *
use                               2.660e+01  1.114e+01   2.387 0.017090 *
quiet.1                          -3.870e+01  1.637e+01  -2.364 0.018185 *
X10pm                             8.001e+01  2.385e+01   3.354 0.000815 ***
rule                              6.186e+01  1.957e+01   3.162 0.001599 **
citi.1                           -2.474e+01  1.230e+01  -2.011 0.044506 *
francisco.1                      -7.809e+01  3.471e+01  -2.250 0.024619 *
san.1                             9.484e+01  3.384e+01   2.803 0.005132 **
look                              5.195e+01  2.208e+01   2.353 0.018740 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 293.1 on 1556 degrees of freedom
Multiple R-squared:  0.3141,    Adjusted R-squared:  0.2943
F-statistic: 15.84 on 45 and 1556 DF,  p-value: < 2.2e-16
```