Linear Regression Model for Airbnb Price Prediction

Contents

Linear Regression on Airbnb Dataset

1. Introduction

The aim of this project is to to predict the price of San Francisco Airbnb listings from features in a given data set. To achieve this goal, we need to build a simple linear regression model to mathematically represent the correlation between one variable Y (Price) and other predictor variables X_i (features). We can then use this linear regression model to predict the Y (Price) given only the selected feature variables X_i. This mathematical equation can be described as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_i X_i, \; where \; \beta_i \; are \; the \; regression \; coefficients$$

In order to assess the prediction accuracy of our model on fresh data, we sampled one training set and one testing set from the given data set. We used the training set to build the linear regression model, then used the model to predict price of the listings in the testing set.

2. Data Selection

The given data set contains 108 different variables that can be used as the predictors in our linear regression model. If we choose only one or two variables, our result will be overfitting, which means our model corresponds too closely to the given data set, and therefore fails to reliably predict fresh data. If we use all 108 variables in our model, the sample variance will become too large, which means the mean absolute prediction error will increase. Because of this bias/variance trade-off, we need to select only a subset of features as our predictors.

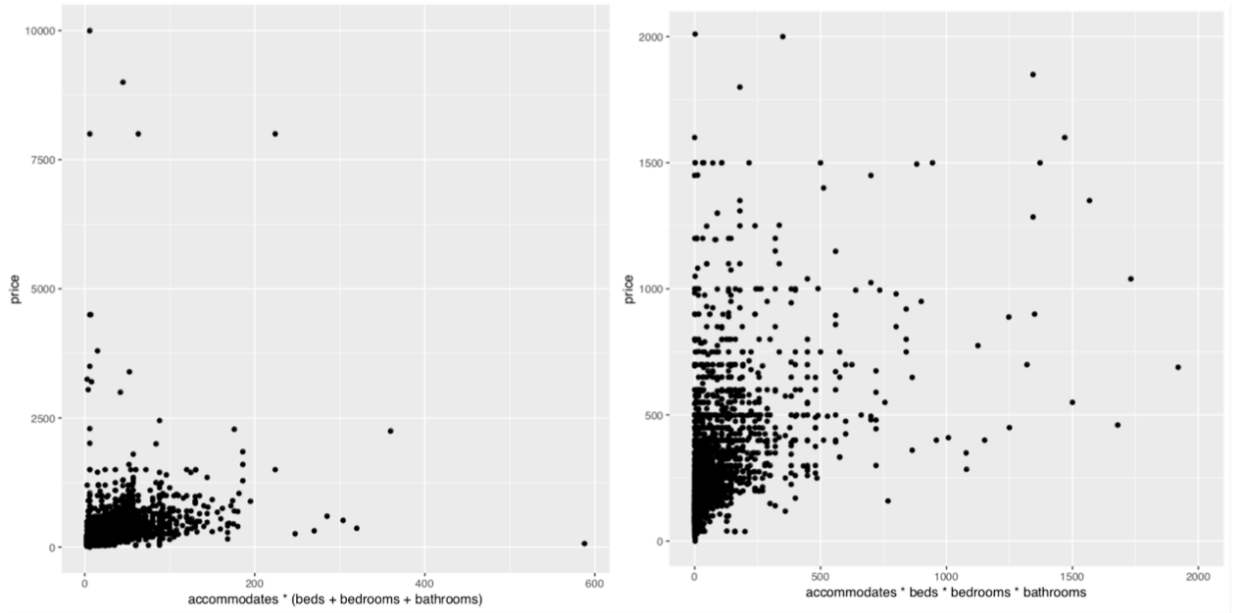
Since real-world data can be messy and contain missing values, we performed the first round of data selection by examining each variable's number of missing values. After printing out the number of missing values for each variable, we decided to only consider the variables whose number of missing values are less than 100:

`listing_url,scrape_id,last_scraped, name, description, experiences_offered, picture_url, host_id, host_url, host_name`
`host_since, host_location, host_response_time, host_response_rate, host_acceptance_rate, host_is_superhost, host_thumbnail_url, host_picture_url, host_listings_count, host_total_listings_count, host_verifications, host_has_profile_pic, host_identity_verified, street, neighbourhood_cleansed, city, state, market, smart_location, country_code country, latitude, longitude, is_location_exact, property_type, room_type, accommodates, bathrooms, bedrooms, beds, bed_type, amenities, price, guests_included, extra_people, minimum_nights, maximum_nights, minimum_minimum_nights, maximum_minimum_nights, minimum_maximum_nights, maximum_maximum_nights, minimum_nights_avg_ntm, maximum_nights_avg_ntm, calendar_updated, has_availability, availability_30, availability_60, availability_90, availability_365, calendar_last_scraped, number_of_reviews, number_of_reviews_ltm, requires_license, jurisdiction_names, instant_bookable, is_business_travel_ready, cancellation_policy, require_guest_profile_picture, require_guest_phone_verification, calculated_host_listings_count, calculated_host_listings_count_entire_homes, calculated_host_listings_count_private_rooms, calculated_host_listings_count_shared_rooms`

Based on common sense, we selected the following set of variables that might be related the price:

`neighbourhood_cleansed, property_type, room_type, accommodates, bathrooms, bedrooms, beds, bed_type, amenities, extra_people`

Then we plotted each variable against price to see if there's a possible correlation. After trying out several linear and non-linear combinations of these variables, we decide the following variable combinations have positive correlations to price:



After several trials, we discovered that if the variable 'longitude' is put into the model as a numeric variable, the prediction error will decrease. Therefore, we also added it as one of our predictors. Our final list of predictors are:

`neighbourhood_cleansed, property_type, room_type, accommodates, bathrooms, bedrooms, beds, bed_type, amenities, extra_people, longitude`

3. Data Wrangling

To make the selected data useful in our model, we also need to fix the missing values and convert category variables into dummy variables.

3.1 Fixing Missing Values

In our selected variables, only three variables bathrooms, bedrooms, beds have missing values. Since a common house will have at least one bathroom and one bedroom with one bed, we filled all the empty cells with 1.

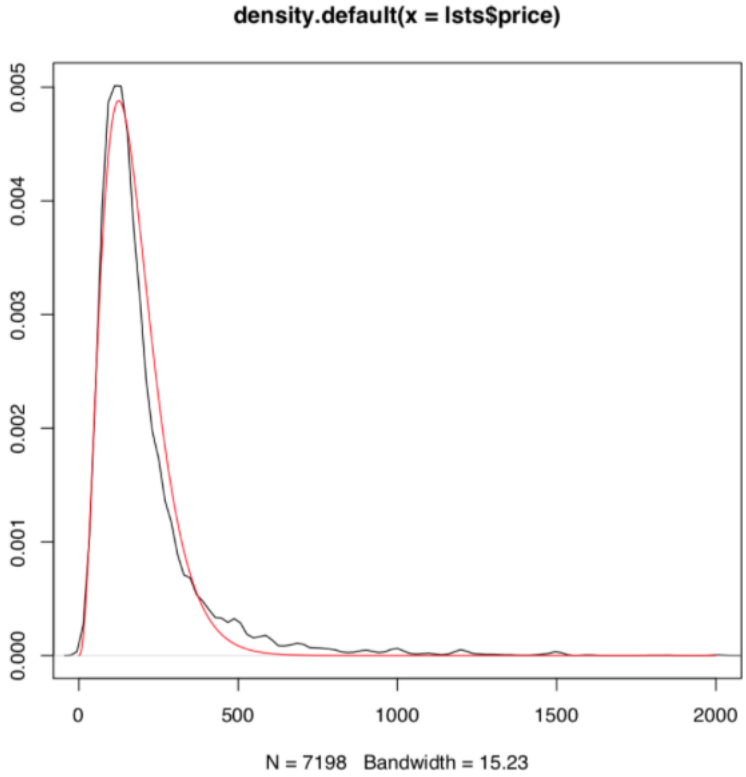
3.2 Converting Category Variables

We noticed that the variables neighbourhood_cleansed, property_type, room_type, bed_type are category variables, so we simply convert them into dummy variables by R function as.numeric(). Also, there are some categories that have too many items inside, (e.g. 36 different neighborhoods). So we decided to also turn those categories into more generalized dummy variables. As a result, we got two dummy variables for three neighborhood types, two dummy variables for three house types, and one simple dummy variable for room types, and one simple variable for bed type.

3.3 Removing Outlier Values

Because the data might be messy, meaning there exist outlier values that are distant from other values. These outlier values can increases the sample variance which will also increase the prediction error. Therefore, we

wanted to remove these outliers from the training set. We plotted the density of price in the given data set and found out it fits the gamma distribution (dgamma(x,3.5,0.02)).



To make our prediction more accurate, we calculated qgamma(0.99,3.5,0.02) = 461.88, where 99% of the data in this distribution have price $\leq$ 461.88, and removed the sample with price $\geq$ 461.88 from the training set.

4. Model Construction

We used lm() function in R to build the linear regression model and use predict() function to make predictions. In R, the lm() function fits a line to the data which is the closest to all data points. In other words, lm() fits the line in a way that minimizes the sum of all squared difference between the line and each data point.

```
Call:
lm(formula = price ~ accommodates * (beds + bedrooms + bathrooms) +
    (accommodates * beds * bedrooms * bathrooms) + longitude +
    extra_people + house + other + room_type + bed_type + area1 +
    area2, data = trndta)

Residuals:
    Min      1Q  Median      3Q     Max
-218.11  -39.69  -11.36   28.29  333.10

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                        4.130e+04  3.823e+03  10.803  < 2e-16 ***
accommodates                       1.267e+01  3.722e+00   3.402 0.000672 ***
beds                              -2.269e+00  7.960e+00  -0.285 0.775621
bedrooms                          -5.461e+00  9.091e+00  -0.601 0.548028
bathrooms                         -3.683e+01  6.554e+00  -5.620 2.00e-08 ***
longitude                          3.360e+02  3.123e+01  10.759  < 2e-16 ***
extra_people                       1.597e-01  3.016e-02   5.295 1.23e-07 ***
house                             -2.232e+01  1.945e+00 -11.475  < 2e-16 ***
other                             -1.931e+01  7.360e+00  -2.624 0.008713 **
room_type                         -3.981e+01  1.833e+00 -21.717  < 2e-16 ***
bed_type                           7.691e-01  3.525e+00   0.218 0.827292
area1                              2.068e+01  2.907e+00   7.114 1.26e-12 ***
area2                              2.722e+01  2.209e+00  12.323  < 2e-16 ***
accommodates:beds                 -3.806e-01  1.246e+00  -0.306 0.759946
accommodates:bedrooms              6.721e-01  2.307e+00   0.291 0.770814
accommodates:bathrooms             7.703e+00  2.479e+00   3.107 0.001901 **
beds:bedrooms                      4.073e+00  4.468e+00   0.911 0.362071
beds:bathrooms                     1.441e+00  4.297e+00   0.335 0.737327
bedrooms:bathrooms                 2.560e+01  5.762e+00   4.444 9.00e-06 ***
accommodates:beds:bedrooms         1.345e-01  3.809e-01   0.353 0.724024
accommodates:beds:bathrooms       -1.127e+00  6.229e-01  -1.809 0.070544 .
accommodates:bedrooms:bathrooms   -1.964e+00  1.246e+00  -1.577 0.114959
beds:bedrooms:bathrooms           -7.362e-01  2.125e+00  -0.346 0.729032
accommodates:beds:bedrooms:bathrooms  7.456e-02  1.265e-01   0.589 0.555680
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 62.95 on 5758 degrees of freedom
Multiple R-squared:  0.5074,    Adjusted R-squared:  0.5054
F-statistic: 257.9 on 23 and 5758 DF,  p-value: < 2.2e-16
```

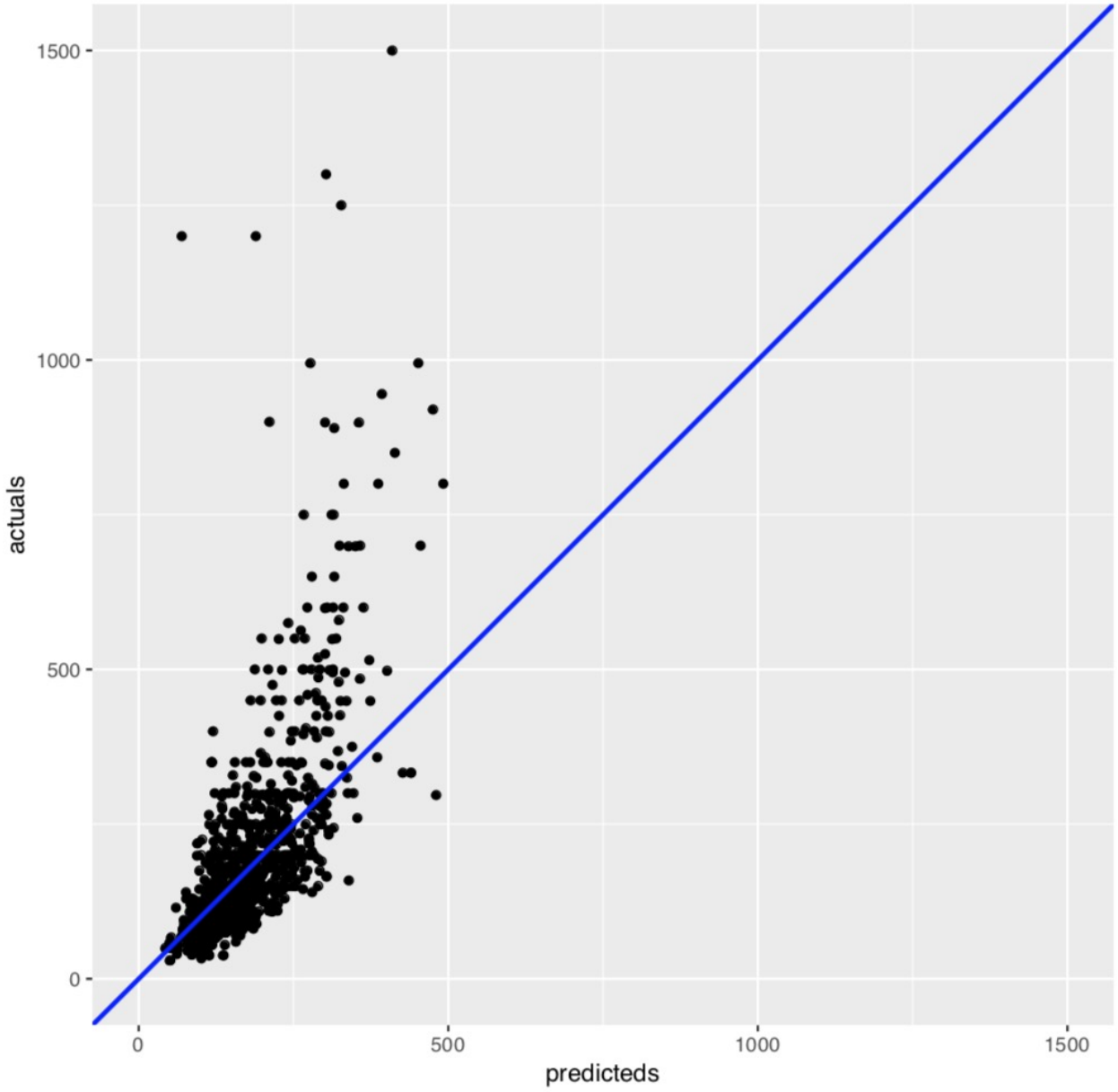
- Coefficients: Estimate and Std. Error:
  - The intercept means the expected price of a listing if all the predictors X_i were zero.
  - The “Estimate” values for each variable are the regression coefficients β_i. It means the effect of each feature variable on price.
  - The coefficient standard errors tell us the average variation of the estimated coefficients from the actual average of our response variable.

We then used the R function predict() function to make predictions from our model. The function predict() takes as input our linear regression model, the returned lm object from lm(), and the values of the selected predictor variable in the testing set. This is effectively calculating the linear regression formula:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_i X_i$$

5. Results

We plot the predicted price against the actual price for the testing data set and get the following graph:



The blue line is the line y = x, which means a perfect prediction. This plot shows a linear relation between the predicted price and the actual price and suggests that our prediction is somewhat accurate.

We calculated the mean absolute prediction error (MAE) and mean absolute percentage error (MAPE) to determine the accuracy of our model. The mathematical formula for MAE and MAPE are as follows:

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|Y_j - \hat{Y}_j|$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i - \hat{Y}_i|}{Y_i}$$

```
"mean absolute prediction error: 77.6205603851199"
"mean absolute percentage error: 33.5364376339157%"
```

The mean absolute prediction error shows on average, our absolute value difference between our prediction and the actual value is 77.62, which is acceptable considering the real world data is messy and noisy. The mean absolute percentage error shows on average, our prediction is making the error as large as 33.5% of the actual value. Since our training data size is limited, this error is acceptable.

6. Summary

In order to predict the price of Airbnb listings in San Francisco, we selected the variables neighbourhood_cleansed, property_type, room_type, accommodates, bathrooms, bedrooms, beds, bed_type, amenities, extra_people, and longitude from the given data set to build a linear regression model. We split the data set into one training set and one testing set to assess our model's prediction accuracy on fresh data. After some data manipulation, we managed to decrease our mean absolute prediction error down to 77.6 and our mean absolute percentage error down to 33.5%, which indicates our model's prediction ability is decent.