Exploring New York City Airbnb Open Data

Mengyao Liu

Dec. 1st, 2019

Keywords: business — internet — regression analysis — classification analysis — New York City — hotels and accommodations

1 Executive Summary

We aim to answer two questions. The first question is to predict the best price range of a room/house for a host on Airbnb using regression analysis to make sure the room/house will be rented more than two thirds of the time of a year. The second question is to predict whether a room/house on Airbnb is popular or not.

To answer the first question, we have build regression models using least squares regression and tree methods to predict `price` given `neighborhood_group`, `room_type` and `minimum_nights`. According to both models, we can see some common trends like "Entire home/apt" usually can be more expensive than "Private room" and "Share room" and listings in Manhattan are usually more expensive than in other neighborhoods.

To answer the second question, we have build classification models using logistic regression, linear discriminant analysis (LDA) and tree methods to predict whether a room/house on Airbnb is popular given `neighborhood_group`, `room_type`, `price`, `minimum_nights`, `number_of_reviews`, and `reviews_per_month`. The best test error rate logistic regression can achieve is $\sim 20\%$. The test error rate for LDA is $\sim 35\%$. However, the assumption of all the quantitative variables being multivariate normal distributed is not met. So we don't consider LDA anymore. The best test error rate of the tree methods is $\sim 28\%$. According to logistic regression and the tree methods, we can see some common trends like the lower `minimum_nights` and `reviews_per_month` are, the more likely a listing is to be popular.

2 Data Processing & Cleaning

We have found that there are 11 observations that have `price = 0`. This is unusual and we do not aim to investigate the reasons here. Thus we remove those 11 observations. Additionally, we set `reviews_per_month` of those observations with missing `reviews_per_month` to zero since they have `number_of_reviews = 0`.

We define "Popular" as `availability_365` < 122. We define "Unpopular" as `availability_365` $\geqslant 122$. In the sample of 48,884 observations, we have 30,632 observations in class "Popular" and 18,252 observations in class "Unpopular".

3 Exploratory Data Analysis

3.1 Regression

To investigate pricing, we plot the distribution of `price` by `neighbourhood_group` (see Figure 1), `room_type` (see Figure 2), and `minimum_nights` (see Figure 3), respectively. Note that we only use observations that are "Popular". We made these plots to gain insight on the significance of each predictor to help determine whether to reduce any of them later as well as insight on the signs of the coefficients.

Apparently, there is difference in `price` between different `neighbourhood_group` and `room_type`. From Figure 1, it can be shown that the rooms/houses located in Manhattan tend to have the highest price while the rooms/houses located in Bronx tend to be priced low. From Figure 2, there is a clear order of price among the three room types with "Entire home/apt" pricing the highest, followed by "Private room", and "Shared room" with the lowest price. Therefore, we can expect `neighbourhood_group` and `room_type` will be strong predictors for `price`. The variation of `price` with `minimum_nights` based on Figure 3 is not obvious so far.

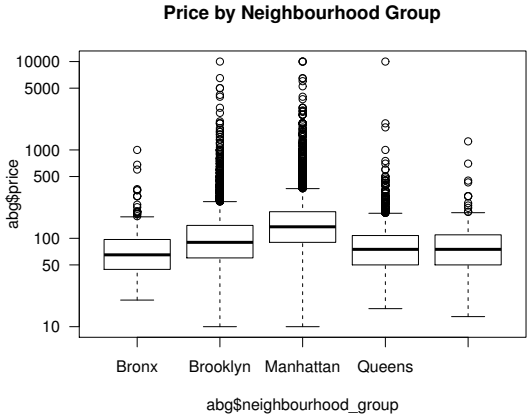

Figure 1

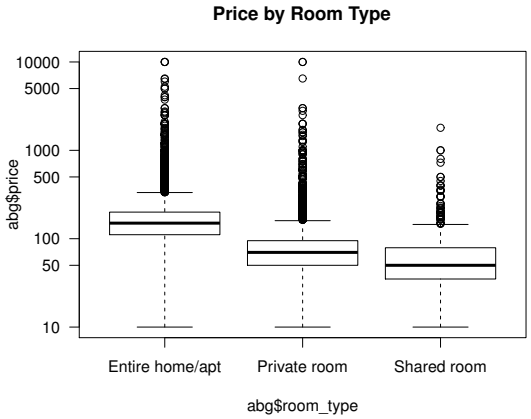

Figure 2

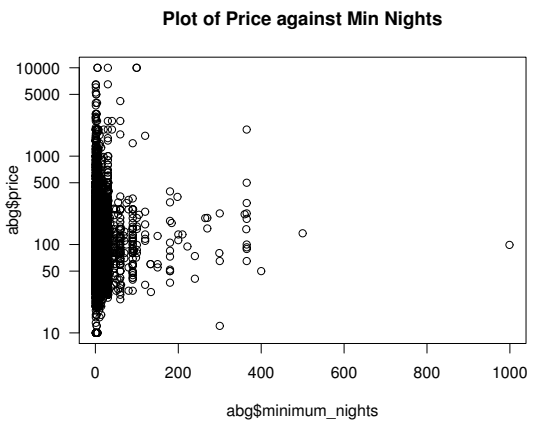

Figure 3

3.2 Classification

To investigate the influence of different predictors on popularity, we plot the distribution of `price` (see Figure 4), `minimum_nights` (see Figure 5), `number_of_reviews` (see Figure 6), and `number_of_reviews_per_month` (see Figure 7) by `popularity`, respectively, where the right panel is a zoom-in plot of the left panel. We also plot the distribution of `availability_365` by `neighborhood_group` (see Figure 8) and `room_type` (see Figure 9). We made these plots to gain insight on the significance of each predictor to help determine whether to reduce any of them later as well as insight on the signs of the coefficients.

From those plots, we can tell that a room/house tends to be popular if it has lower `price`, `minimum_nights`, `number_of_reviews`, and `reviews_per_month`. Also there seems to be a difference in `popularity` between different `neighborhood_group` and `room_type`: listings in Brooklyn, Manhattan, and Queens are more popular than those in Bronx and Staten Island; "Entire homes/apts" and "Private rooms" are more popular than "Shared rooms".

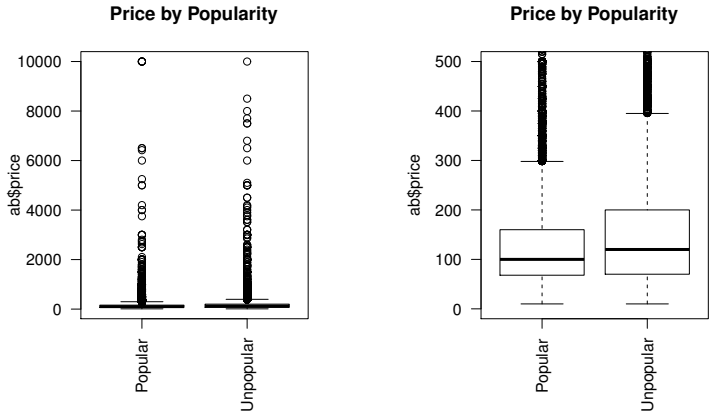

Figure 4

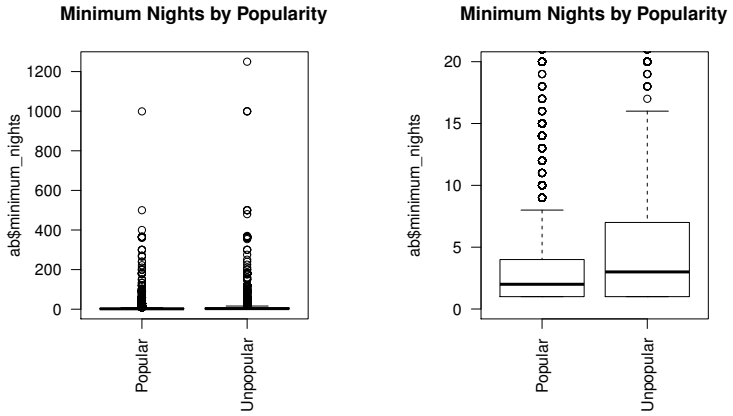

Figure 5

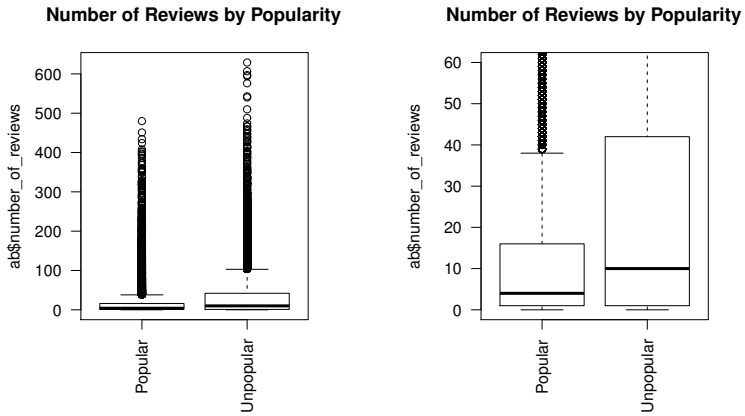

Figure 6

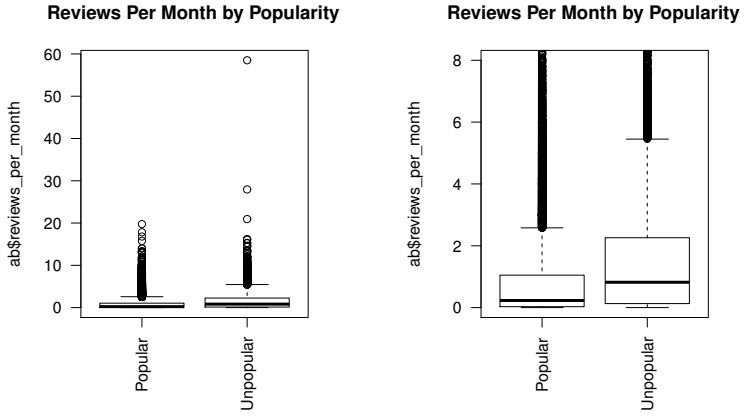

Figure 7

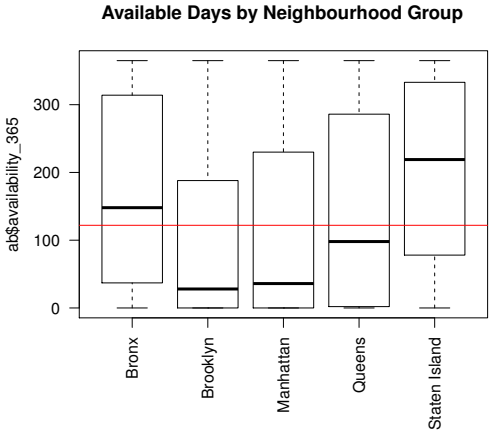

Figure 8

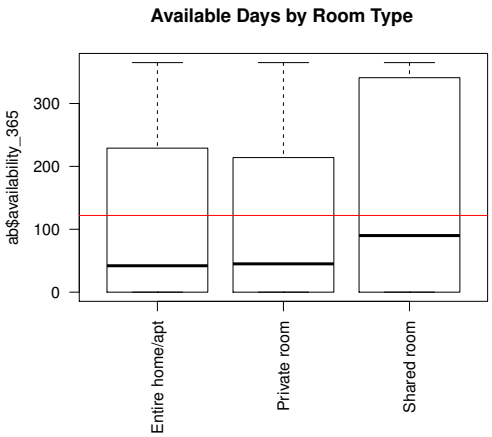

Figure 9

4 Analysis from Regression, Classification, Trees

4.1 Regression

4.1.1 Linear Regression

We perform least squares regression on the 30,632 "Popular" observations and the results are shown in Figure 10. Note based on earlier Box Cox output, we raise the response variable to an exponent of -0.28. According to the residual plots shown in Figure 11, the variance appears constant and the residuals are close to 0. Although there are some high leverage points seen from the leverage plot in Figure 11, they are not influential so that is not a problem. According to the p-values the coefficients of all the predictors are significant at 0.1% level except the coefficient for the last class of `neighborhood_group` is significant at 10% level. Thus there is no need to reduce any predictors.

```
Call:
lm(formula = price^(-0.28) ~ room_type + minimum_nights + neighbourhood_group)

Residuals:
      Min        1Q    Median        3Q       Max
-0.257405 -0.019473  0.000803  0.022134  0.292496

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    2.740e-01  1.587e-03 172.619  < 2e-16 ***
room_typePrivate room          5.738e-02  4.116e-04 139.412  < 2e-16 ***
room_typeShared room           8.276e-02  1.464e-03  56.529  < 2e-16 ***
minimum_nights                 1.291e-04  1.436e-05   8.992  < 2e-16 ***
neighbourhood_groupBrooklyn   -1.997e-02  1.595e-03 -12.523  < 2e-16 ***
neighbourhood_groupManhattan  -4.205e-02  1.596e-03 -26.346  < 2e-16 ***
neighbourhood_groupQueens     -1.100e-02  1.693e-03  -6.498 8.29e-11 ***
neighbourhood_groupStaten Island -6.320e-03  3.453e-03  -1.830   0.0672 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03524 on 30624 degrees of freedom
Multiple R-squared:  0.4746,    Adjusted R-squared:  0.4744
F-statistic:  3951 on 7 and 30624 DF,  p-value: < 2.2e-16
```

Figure 10

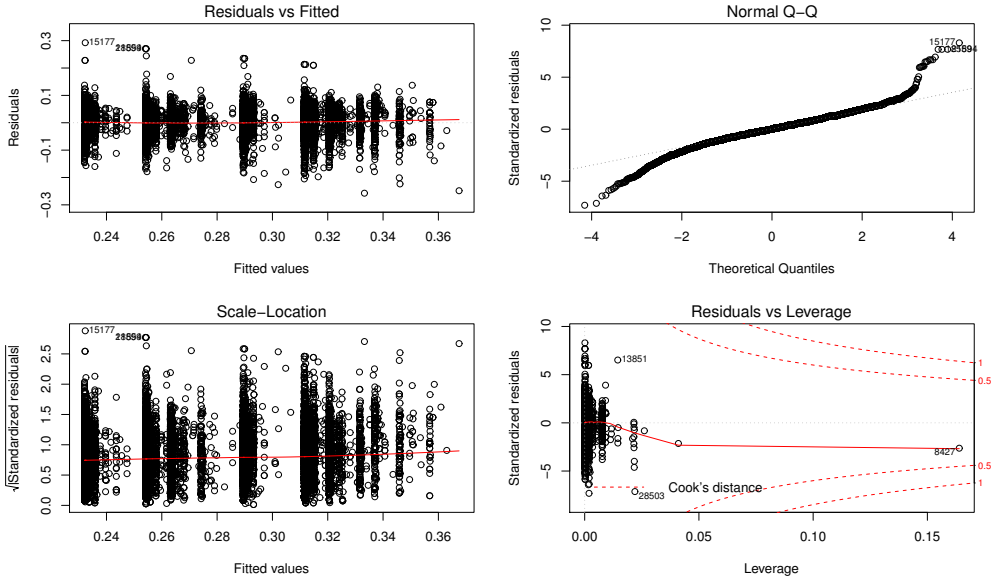

Figure 11

In order to investigate the influence of `room_type` and `neighborhood_group` on `price` more precisely, we compare the means in `price` between each pair of `room_type` in Figure 12 and `neighborhood_group` in Figure 13, respectively. From Figure 12, we see that all the coefficients are significant at 0.1% and positive. That means different `room_type` definitely makes a difference. From Figure 13, we can see not all pairs are significantly different.

```
        Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: lm(formula = price^(-0.28) ~ room_type + minimum_nights + neighbourhood_group)

Linear Hypotheses:
                                    Estimate Std. Error t value Pr(>|t|)
Private room - Entire home/apt == 0 0.0573760  0.0004116  139.41   <2e-16 ***
Shared room - Entire home/apt == 0  0.0827602  0.0014640   56.53   <2e-16 ***
Shared room - Private room == 0     0.0253841  0.0014662   17.31   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)
```

Figure 12

```
        Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: lm(formula = price^(-0.28) ~ room_type + minimum_nights + neighbourhood_group)

Linear Hypotheses:
                                     Estimate Std. Error t value Pr(>|t|)
Brooklyn - Bronx == 0           -0.0199713  0.0015948 -12.523   <0.001 ***
Manhattan - Bronx == 0          -0.0420502  0.0015961 -26.346   <0.001 ***
Queens - Bronx == 0             -0.0110008  0.0016931  -6.498   <0.001 ***
Staten Island - Bronx == 0      -0.0063199  0.0034533  -1.830    0.305
Manhattan - Brooklyn == 0       -0.0220789  0.0004314 -51.182   <0.001 ***
Queens - Brooklyn == 0           0.0089705  0.0007160  12.529   <0.001 ***
Staten Island - Brooklyn == 0    0.0136514  0.0030944   4.412   <0.001 ***
Queens - Manhattan == 0          0.0310494  0.0007191  43.179   <0.001 ***
Staten Island - Manhattan == 0   0.0357303  0.0030942  11.547   <0.001 ***
Staten Island - Queens == 0      0.0046809  0.0031468   1.488    0.517
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)
```

Figure 13

In summary, a higher `minimum_nights` tends to lead to a lower `price`. The order of `price` based on `room_type` assuming other predictors hold the same should be Entirehome/apt > Privateroom > Shareroom. The order of `price` based on `neighborhood_group` assuming other predictors hold the same should be Manhattan > Brooklyn > Queens > StatenIsland $\gtrsim$ Bronx.

4.1.2 Tree methods

We randomly split the 30,632 “Popular” observations into a training set and a test set of equal sizes. Then we fit a regression tree on the training set using recursive binary splitting. The results are shown in Figure 14 and Fig 15. All the three predictors are used. The tree has 5 terminal nodes. It tells us that if you have a “Private room” or “Shared room”, if `minimum_nights` < 94.5, `price` should be set around 83.32 to reach an average level; if `minimum_nights` > 94.5 and the room/house is located in Brooklyn, the average `price` of popular listings is around 83.17; if `minimum_nights` $>$ 94.5 and the room/house is not located n Brooklyn, you can even set `price` at 2780. Otherwise, if you have a “Entire home/apt”, and it is not located in Manhattan, `price` should be set around 161.60; if the house is located in Manhattan, `price` should be set around 216.90. The test MSE

is 45952.85. We notice the node with a price of 2780 is quite odd. After examination, we find the branch with `minimum_nights` > 94.5 only have 11 observations and the node of 2780 results from 5 observations. We will talk more about these odd observations below.

```
Regression tree:
tree(formula = price ~ room_type + minimum_nights + neighbourhood_group,
    data = train)
Number of terminal nodes:  5
Residual mean deviance:  33580 = 514200000 / 15310
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-2730.00   -46.64   -20.32     0.00    13.36  7219.00
```

Figure 14

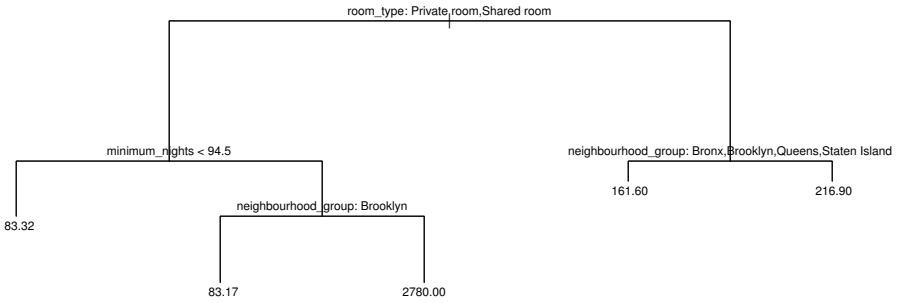

Figure 15

Then we seek to improve the performance of the tree by using pruning, bagging and random forests. Here we present the results of random forest since it gives the best test MSE. We set $m = 1$ and the importance result is essentially the same as shown in Figure 16 that `room_type` and `minimum_nights` are more important than `neighborhood_group`. The test MSE from random forest is 43505.16.

From § 3 we already know that there are a number observations with extremely high `minimum_nights` and `price`, which is not normal. We thus carry out an experiment and see how the tree would change if we only target on normal sample and exclude the observations with `minimum_nights` > 30 and `price` > 400. The two numbers are based on the histograms of `minimum_nights` and `price`. There are 14,819 observations left in the training set and 14,805 observations left in the test set then. The new tree is shown in Fig 17. This time the tree only has 4 nodes. The number of observations in each node is of the same magnitude. The price predicted also looks more reasonable. The test MSE is 3349.79. The results tell us still "Entire home/apt" is more expensive than "Private room" and "Shared room". And a listing in Manhattan tends to be more expensive than other locations. We then perform random forests. The importance of the variables has also changed as shown in Figure 18. `room_type` and `neighborhood_group` are more important than `minimum_nights`. The test MSE is 3376.989.

4.1.3 Discussion

The results of linear regression and tree methods basically agree. Linear regression shows that `neighborhood_group`, `room_type` and `minimum_nights` are all significant predictors. Tree methods

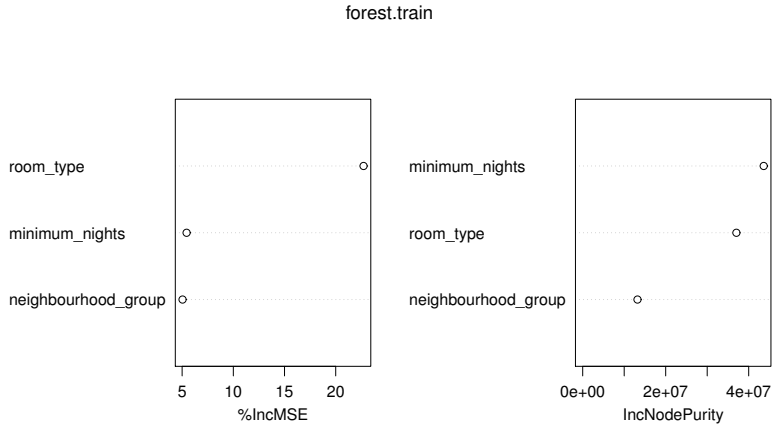

Figure 16

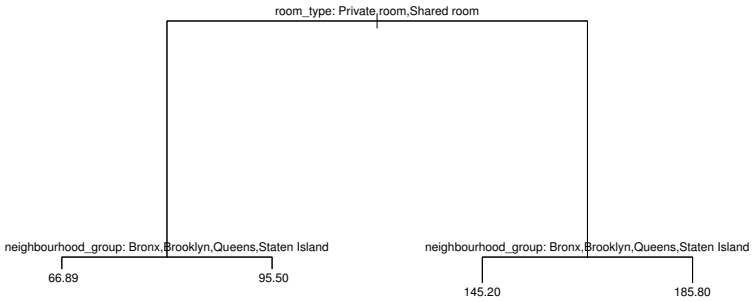

Figure 17

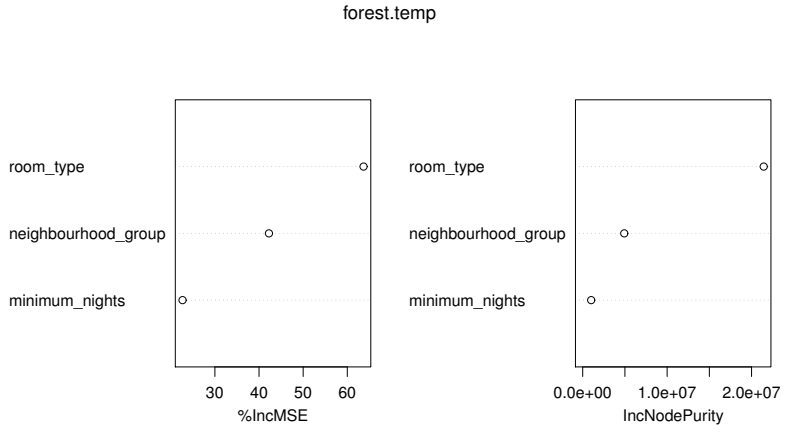

Figure 18

show that `room_type` may be the most important predictor. Both methods show that a listing of "Entire home/apt" tends to have a higher price than other room types and a listing in Manhattan tends to have a higher price than in other locations. Linear regression also shows that `price` will decrease with increasing `minimum_nights`. This is reflected in the tree derived from recursive binary splitting with the whole sample, but not revealed in the tree when we only consider normal sample.

4.2 Classification

We randomly split the data into a training set (60% of the observations) and a test set (40% of the observations). Note that in the models "Unpopular" is the positive class while "Popular" is the negative class.

4.2.1 Logistic Regression

We perform logistic regression on the training set and the results are shown in Figure 19. We see all the predictors are very significant. The coefficients for `price`, `minimum_nights`, `number_of_reviews`, and `reviews_per_month` are all positive, which indicates a lower probability of being "Popular". The coefficients for `neighborhood_group` of Brooklyn, Manhattan, and Queens are negative, indicating the rooms/houses in those three neighborhoods are more popular than those in Bronx and Staten Island. The coefficients for `room_type` of "Private room" and "Shared room" are positive, indicating "Entire homes/apts" are more popular. The coefficient for "Shared room" is even larger than that of "Private room", indicating "Shared room" is the most unpopular room type. These results all agree with the plots in §3.

We test the model on the test set. The ROC curve is shown in Figure 20. The AUC is 0.738. The test error rate using k fold cross-validation is 0.2032 with k = 5 and 0.2032 with k = 10. We show the confusion matrix on the test set in Table 1. While the overall error rate is $\sim 30\%$, a little higher than the k fold cross-validation test error, which is expected, we notice that the false negative rate (FNR) is as high as $\sim 60\%$. The false positive rate (FPR) is only $\sim 7\%$. That indicates our overall error rate is dominated by FNR and unpopular listings are likely to be classified as popular. Thus we should adopt a lower threshold to improve the prediction accuracy.

Table 1: Confusion Matrix of Logistic Regression

	Popular (pred)	Unpopular (pred)
Popular	10769	1536
Unpopular	4374	2875

4.2.2 Tree Methods

We fit a classification tree on the training set using recursive binary splitting. The results are shown in Figure 21 and Fig 22. The tree has 3 terminal nodes. Surprisingly, the tree only uses two predictors `minimum_nights` and `reviews_per_month` but essentially only `minimum_nights`. The tree predicts that a house/room is likely to be popular if `minimum_nights` < 27.5 and unpopular otherwise. We notice the two nodes under the branch of `reviews_per_month` both return "Popular". After examination, we find that the node of `reviews_per_month` < 0.505 has a probability of "Popular" of 0.80 while the other node only has a probability of 0.53. So the node purity is still quite different.

```
Call:
glm(formula = popularity ~ price + minimum_nights + number_of_reviews +
    reviews_per_month + neighbourhood_group + room_type, family = binomial,
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.4904  -0.8599  -0.7116   1.1377   1.8741

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -0.8001472  0.0864650  -9.254  < 2e-16 ***
price                            0.0021607  0.0001136  19.027  < 2e-16 ***
minimum_nights                   0.0490055  0.0014520  33.751  < 2e-16 ***
number_of_reviews                0.0077555  0.0003859  20.095  < 2e-16 ***
reviews_per_month                0.1768877  0.0103267  17.129  < 2e-16 ***
neighbourhood_groupBrooklyn     -0.9623660  0.0837381 -11.493  < 2e-16 ***
neighbourhood_groupManhattan    -0.9366377  0.0841933 -11.125  < 2e-16 ***
neighbourhood_groupQueens       -0.3442836  0.0886613  -3.883 0.000103 ***
neighbourhood_groupStaten Island  0.3206833  0.1661349   1.930 0.053575 .
room_typePrivate room            0.1943730  0.0294764   6.594 4.28e-11 ***
room_typeShared room             0.7853235  0.0830221   9.459  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 38812  on 29329  degrees of freedom
Residual deviance: 35304  on 29319  degrees of freedom
AIC: 35326

Number of Fisher Scoring iterations: 6
```

Figure 19

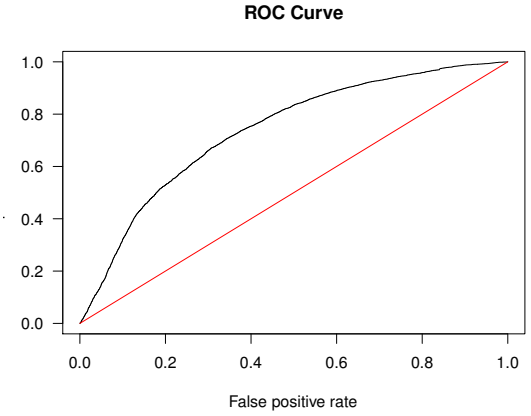

Figure 20

The confusion matrix on the test data is shown in Table 2 based on a threshold of 0.5, which is the threshold adopted by the tree method. The test error rate is 0.323. Similar to logistic regression, the FNR is as high as $\sim 80\%$. This indicates we should lower the threshold so that for example the node of `reviews_per_month` $\geqslant 0.505$ can be classified as "Unpopular".

```
Classification tree:
tree(formula = popularity ~ price + minimum_nights + number_of_reviews +
    reviews_per_month + neighbourhood_group + room_type, data = train)
Variables actually used in tree construction:
[1] "minimum_nights"    "reviews_per_month"
Number of terminal nodes:  3
Residual mean deviance:  1.179 = 34580 / 29330
Misclassification error rate: 0.3239 = 9499 / 29330
```

Figure 21

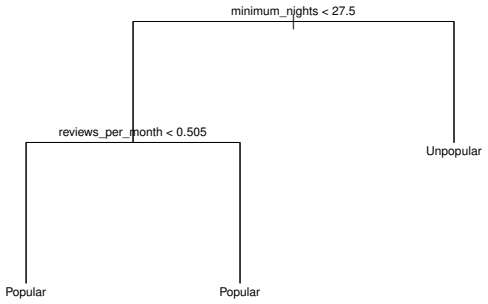

Figure 22

Table 2: Confusion Matrix of Recursive Binary Splitting Tree

	Popular (pred)	Unpopular (pred)
Popular	11751	554
Unpopular	5771	1478

Then we seek to improve the performance of the tree by using pruning, bagging and random forests. Here we present the results of random forest since it gives the best test error rate. We set $m = 3$. From Figure 23, we can see `minimum_nights` and `reviews_per_month` result in the highest mean decrease in accuracy if excluded. `reviews_per_month` and `price` result in the highest mean decrease in Gini index followed by `number_of_reviews` and `minimum_nights`. It seems `reviews_per_month` is the most important predictor and `minimum_nights` and `price` are somewhat important. The test error rate from random forest is 0.284.

Again we carry out an experiment to exclude the observations with `minimum_nights` > 30 and `price` > 400. The tree is the same as Fig 22 and the probability under each node is also similar.

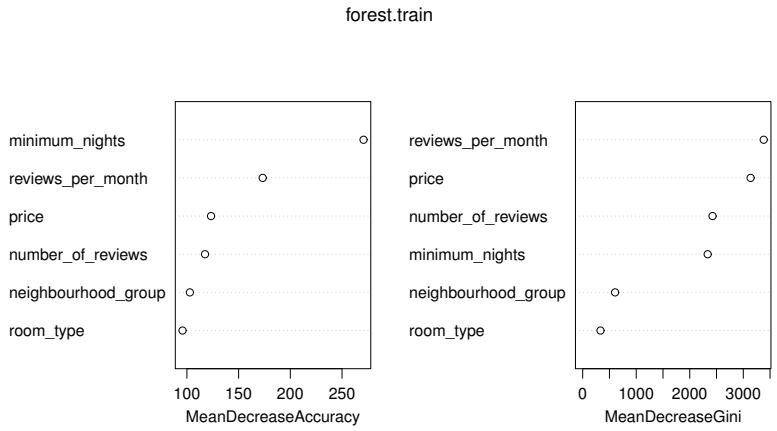

Figure 23

4.2.3 Discussion

While logistic regression and tree methods agree in the sense that a listing tends to be popular with lower `minimum_nights` and `reviews_per_month`, they do not quite agree on the significance of other predictors. Logistic Regression shows that all the predictors are significant and can reveal the correlation trends. Tree methods only make use of `minimum_nights` and `reviews_per_month` either for the whole sample or for the normal sample. An interesting result is that unpopular rooms/houses tend to have more reviews (higher `number_of_reviews` and `reviews_per_month`), which implies that people may be more keen to post complaints than compliments.

Logistic regression has a test error rate $\sim 20\%$ and the tree methods have a test error rate $\sim 28\%$. But both models suffer from really high FNR when adopting a threshold of 0.5 and the overall error rate is dominated by the FNR. So adopting a lower threshold will improve the models.

5 Future Work

If given more time, we may think of a way to determine a test MSE range beforehand for the regression problem to better evaluate the regression models. Another aspect is that now the spatial information is all represented by `neighborhood_group`. If we take the longitude and latitude of individual listings into consideration, we will see more correlation and more clear impact on price and popularity based on the location.