Analysis on New York Airbnb Rent

Semester: Fall 2020

Qianyi Zhang
CS513-A
CWID: 10455276
Computer Science
Graduate Student

Zichong Wang
CS513-A
CWID:10464881
Computer Science
Graduate Student

Zheng Liu
CS513-A
CWID:10455895
Computer Engieering
Graduate Student

Ziming Zhang
CS513-A
CWID:10455301
Computer Science
Graduate Student

Instructor : Prof. Khasha Dehnad

Overview

- House price is different over time and location leads to high cost for compare lots of house price.
- Common expenses of compare different house price
  - Room type
  - Longitude and latitude
  - Neighborhood group
- Customer often compare the similar house.
- The reasons for house price fluctuations are similar.

Problem Description

Goal

- Help customer more fast to get the good price for rental a house.
- To show how the different reasons to effect the house price.
- Provide factors to the house owner with which they can improve the house price increase.

Problem Description

Summary

- Predict the price for a house from the different location.
- Factors considered
  - Room type
  - House’s longitude and latitude
  - Neighborhood group
- Performed data preparation, data analysis & data modelling.
- Developed machine learning models to predict the attrition.

Approaches / Techniques Used

1. KNN (Zheng Liu)
2. kkNN (Qianyi Zhang)
3. Naïve Bayes (Ziming Zhang)
4. Decision Tree (Ziming Zhang)
5. SVM (Qianyi Zhang)
6. ANN (Zheng Liu)
7. Random Forest (Zichong Wang)
8. C5.0 (Zichong Wang)

Solution

- Read the data set and analyze the data for some absent value.
- Convert data based on the algorithm requirement.
- Normalize the data for certain algorithms.
- Apply algorithms.
- Train the data.
- Predict the price for the house.
- Create visualization for some models.

Supervised Learning

KNN - K - Nearest-Neighbor

- The data points are predicted based on how its k neareast neighbor data are classified.

- The target variable is the “price” of airbnb in NYC based on correlated featrues which are :
  - Latitude & Longitude (location)
  - room type
  - reviews & availability (popularity)

- Normalization : Min - Max Normalization

KNN - K - Nearest-Neighbor Result

```
# Get 70% data as training set, 30% data as testing set.
idx <- sort(sample(nrow(data),as.integer(.3*nrow(data))))
training <- data[-idx,] # train 70% of the data
test <- data[idx,] # test 30% of the data
```

```
[1] "k =  5  accuracy =  0.667094565682499"
[1] "k =  15  accuracy =  0.694765368706319"
[1] "k =  25  accuracy =  0.69248324062188"
[1] "k =  35  accuracy =  0.696191698759093"
[1] "k =  45  accuracy =  0.694622735701041"
[1] "k =  55  accuracy =  0.693196405648267"
[1] "k =  65  accuracy =  0.690486378547996"
[1] "k =  75  accuracy =  0.692197974611325"
[1] "k =  85  accuracy =  0.691056910569106"
[1] "k =  95  accuracy =  0.694052203679932"
[1] "k =  105  accuracy =  0.691342176579661"
[1] "k =  115  accuracy =  0.689630580516332"
[1] "k =  125  accuracy =  0.689487947511054"
[1] "k =  135  accuracy =  0.68863214947939"
[1] "k =  145  accuracy =  0.687776351447725"
[1] "k =  155  accuracy =  0.687633718442448"
[1] "k =  165  accuracy =  0.684923691342177"
[1] "k =  175  accuracy =  0.685066324347454"
[1] "k =  185  accuracy =  0.684067893310512"
[1] "k =  195  accuracy =  0.682926829268293"
```

- Total data tested = 7012 (30% of original dataset)

- Table of prediction

- K = 35 can get the best performance.
- Accuracy: ~69.6%

Supervised Learning

kkNN

- Investigate whether neighbourhood_group, neighbourhood, room_type have an impact on price_level

```
> summary(data)
   neighbourhood_group             neighbourhood                  room_type          price_level
 Bronx        : 1090    Williamsburg      : 3919   Entire home/apt:25407   [0, 100)  :21866
 Brooklyn     :20095    Bedford-Stuyvesant: 3710   Private room   :22319   [100, 200):17233
 Manhattan    :21660    Harlem            : 2658   Shared room    : 1158   [200, +)  : 9785
 Queens       : 5666    Bushwick          : 2462
 Staten Island:  373    Upper West Side   : 1971
                        Hell's Kitchen    : 1958
                        (Other)           :32206
```

- K = 3

```
table(test$price_level,fit)
            fit
             [0, 100) [100, 200) [200, +)
[0, 100)         4809       1667       56
[100, 200)       1214       3614      418
[200, +)          225       2200      462
```

Accuracy: 60.6%

- K = 10

```
table(test$price_level,fit)
            fit
             [0, 100) [100, 200) [200, +)
[0, 100)         5336       1135       61
[100, 200)       1280       3370      596
[200, +)          227       2045      615
```

Accuracy: 63.6%

Supervised Learning

Naïve Bayes

- Assume that all the featrues are independent of each other
- The dependent variable: neighbourhood_group, neighbourhood, latitude, longitude, room type.

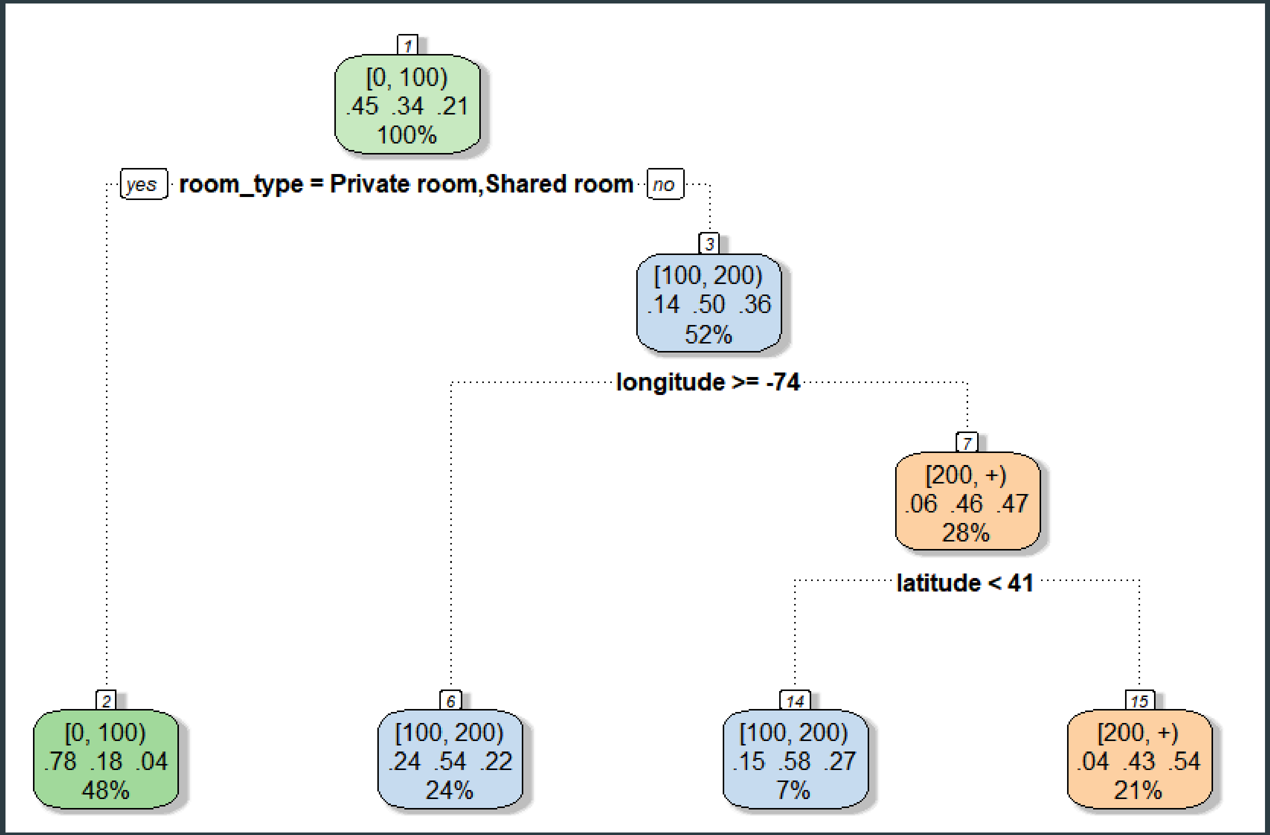
```
> conf_matrix

predict        [0, 100) [100, 200) [200, +)
  [0, 100)        12001       2563      590
  [100, 200)       2996       6433     2608
  [200, +)          398       2983     3646
> prop.table(conf_matrix)

predict          [0, 100) [100, 200)   [200, +)
  [0, 100)    0.35072184 0.07490210 0.01724239
  [100, 200)  0.08755626 0.18800047 0.07621720
  [200, +)    0.01163131 0.08717634 0.10655211
> #Measure the accuracy
> accuracy <- function(x){sum(diag(x)/(sum(rowSums(x))))}
> print(paste("Accuracy:" , accuracy(conf_matrix)))
[1] "Accuracy: 0.645274416973523"
```

Accuracy: 64.53%

Supervised Learning

Decision Trees

- Decision tree is a basic classification and regression method. The decision tree model has a tree structure. It can be thought of as a collection of if-then rules.
- Use the CART algorithm.
- The dependent variable : latitude, longitude, room type.




```
> table(Actual=test_data[,4],CART=pred)
            CART
Actual        [0, 100) [100, 200) [200, +)
  [0, 100)       12738       2250      272
  [100, 200)      3050       5908     3242
  [200, +)         645       2262     3851
> # Compare the prediction to actual data and calculate the accuracy
> right<-(test_data[,4]==pred)
> accuracy<-sum(right)/length(right)
> print(paste("Accuracy :" , accuracy))
[1] "Accuracy : 0.657460985446256"
```

Accuracy: 65.75%

Supervised Learning

Support Vector Machines (SVM)

- Investigate whether neighbourhood_group, neighbourhood, room_type have an impact on price_level

```
> summary(data)
   neighbourhood_group              neighbourhood                room_type            price_level
 Bronx        : 1090    Williamsburg      : 3919   Entire home/apt:25407   [0, 100)  :21866
 Brooklyn     :20095    Bedford-Stuyvesant: 3710   Private room   :22319   [100, 200):17233
 Manhattan    :21660    Harlem            : 2658   Shared room    : 1158   [200, +)  : 9785
 Queens       : 5666    Bushwick          : 2462
 Staten Island:  373    Upper West Side   : 1971
                        Hell's Kitchen    : 1958
                        (Other)           :32206
```

```
              true
pred            [0, 100) [100, 200) [200, +)
  [0, 100)          5582       1242      309
  [100, 200)        1039       3480     2038
  [200, +)            22        344      609
```

Accuracy: 65.9%

Supervised Learning

Artificial Neural Network

- Stimulate the behaviour of biological system composed of neurons.

- The target variable is the “price” of aribnb in NYC based on correlated featrues which are :

  - Latitude & Longitude (location)
  - room type
  - reviews & availability (popularity)

- Normalization : Min - Max Normalization

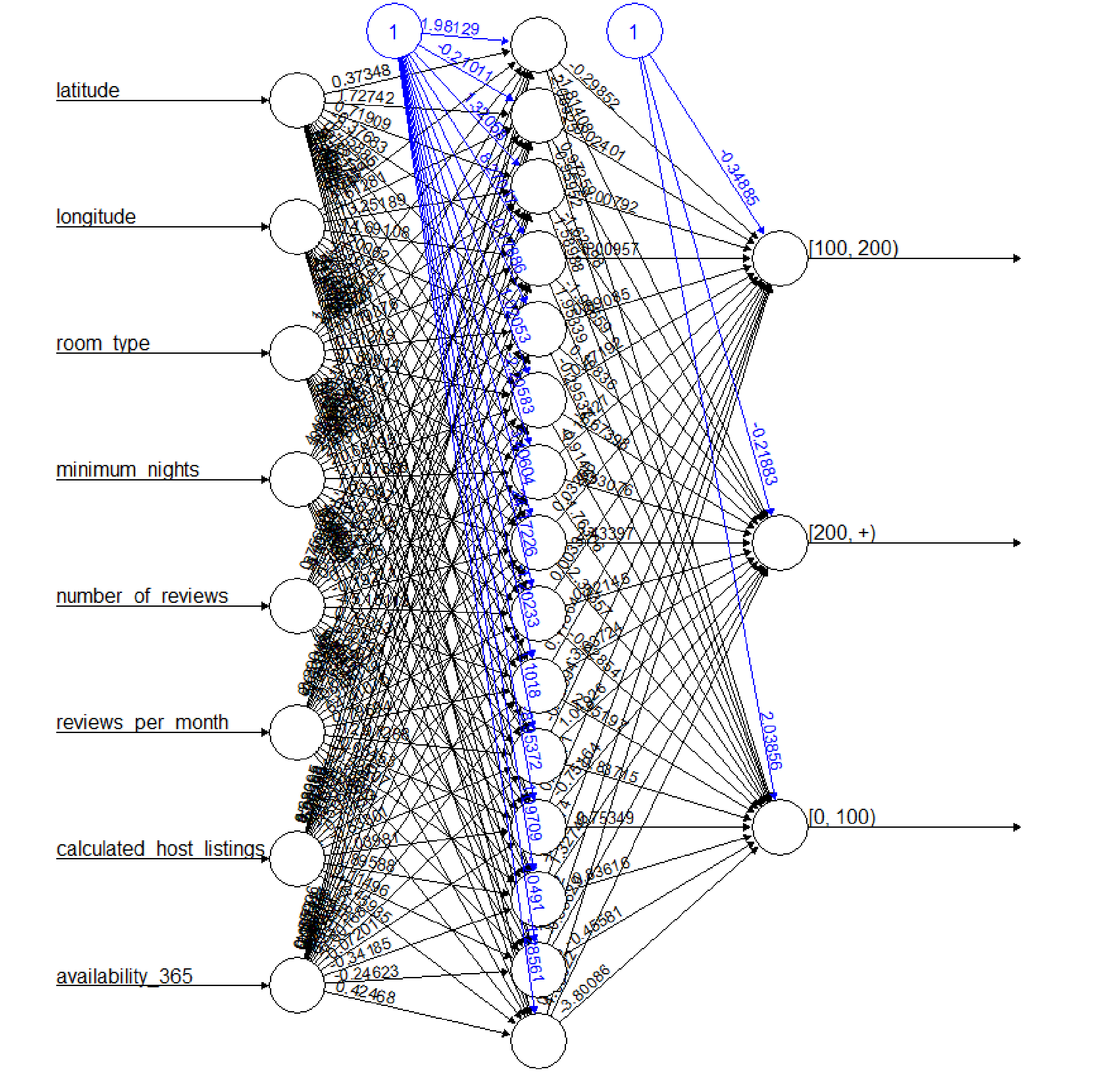
- Hidden Node : 15

Supervised Learning

Artificial Neural Network

Accuracy ~ 69.8%

```
             predition
Actual        [100, 200) [200, +)
  [0, 100)             1        0
  [100, 200)        2190      416
  [200, +)          1695     2710
> wrong<- (test$price_level!=ann_cat)
> error_rate<-sum(wrong)/length(wrong)
> error_rate
[1] 0.3011979
> accuracy <- 1 - error_rate
> accuracy
[1] 0.6988021
```



Supervised Learning

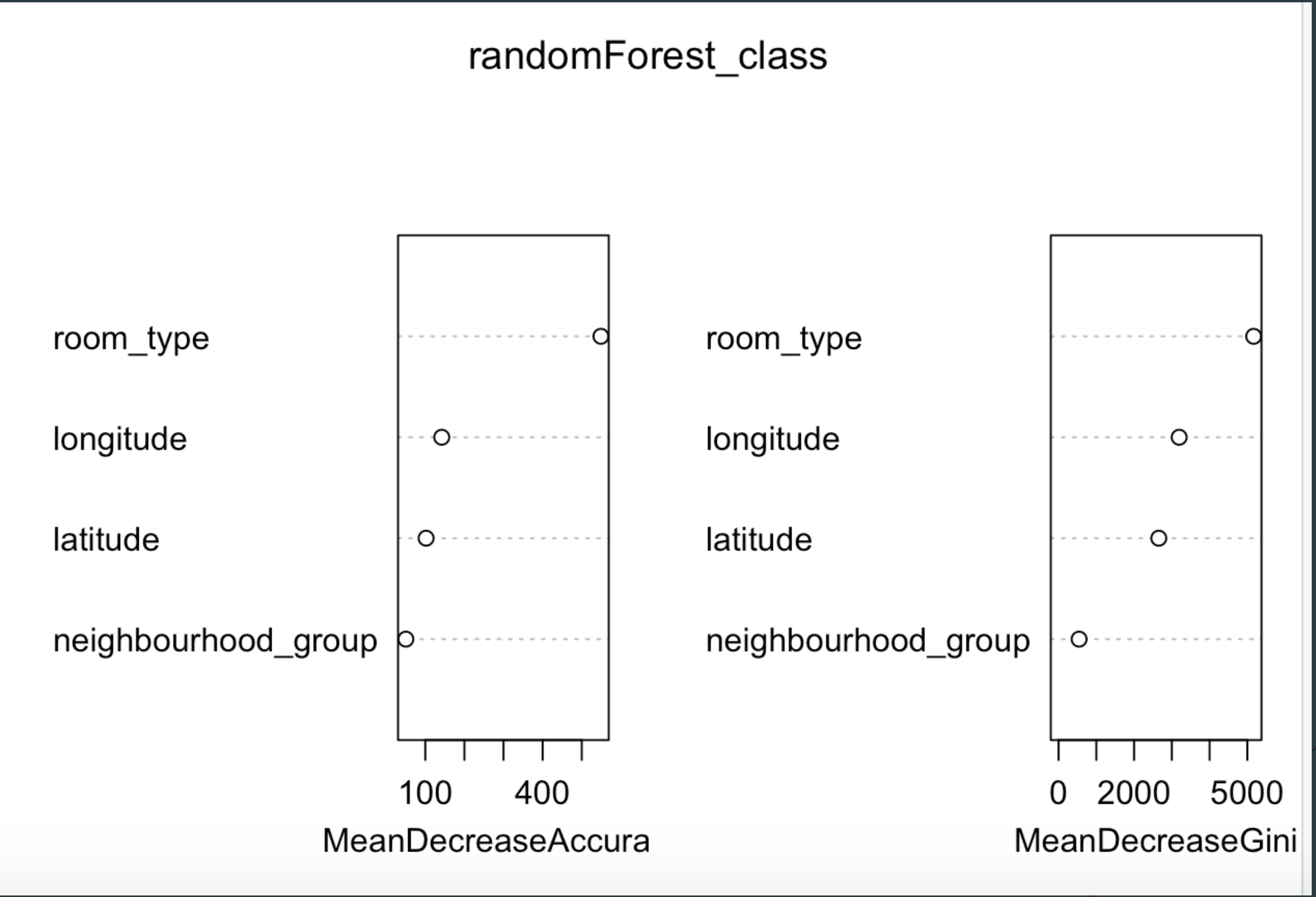
Random Forest

- It is unexcelled in accuracy among current algorithms.
- It runs efficiently on large data bases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.

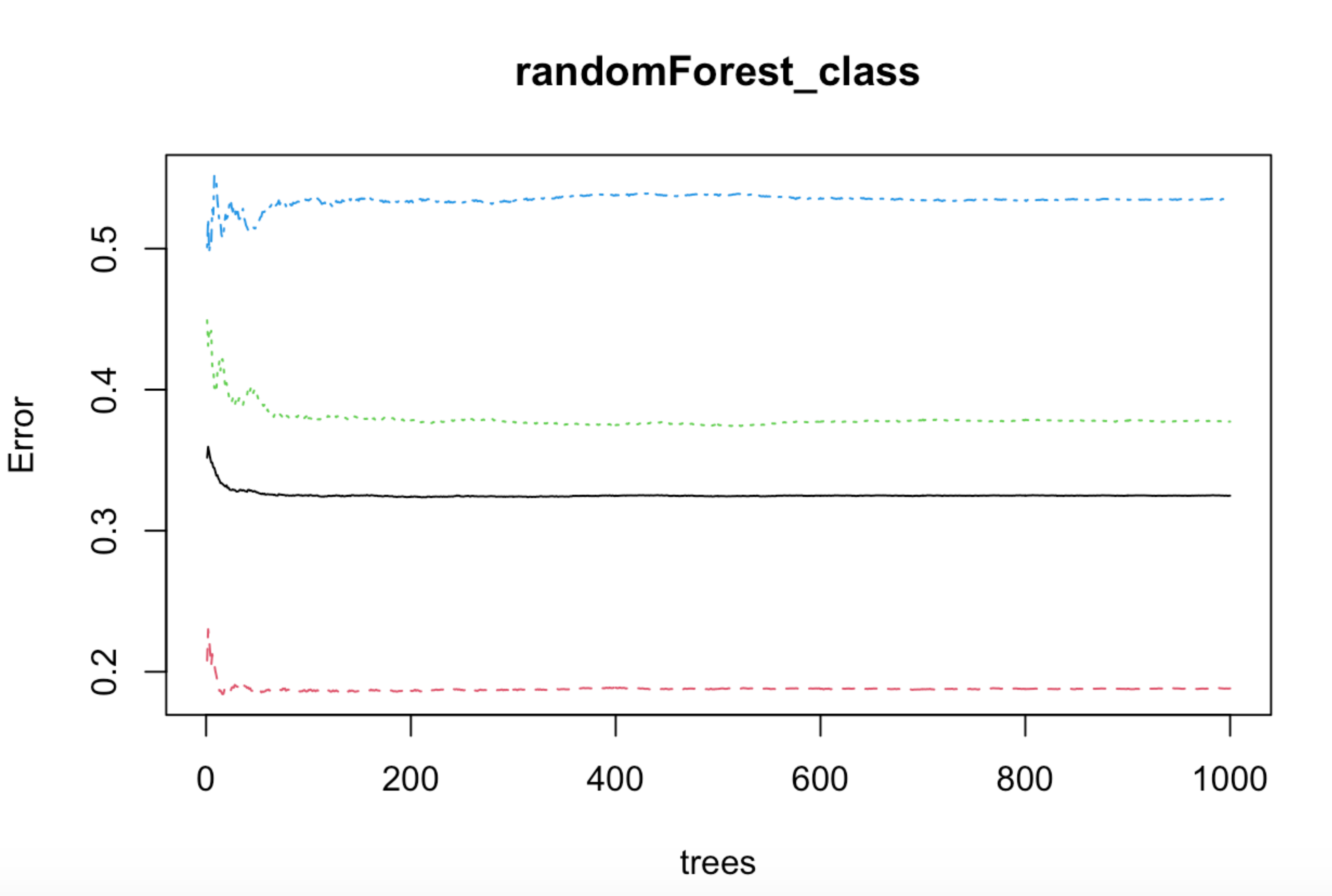
```
            Prediction
             [0, 100) [100, 200) [200, +)
  [0, 100)       4503        959       68
  [100, 200)      824       2688      787
  [200, +)        153       1140     1099

Overall Statistics

               Accuracy : 0.6783
                 95% CI : (0.67, 0.6866)
    No Information Rate : 0.4484
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4878
```

Accuracy: 67.83%

Supervised Learning



Supervised Learning



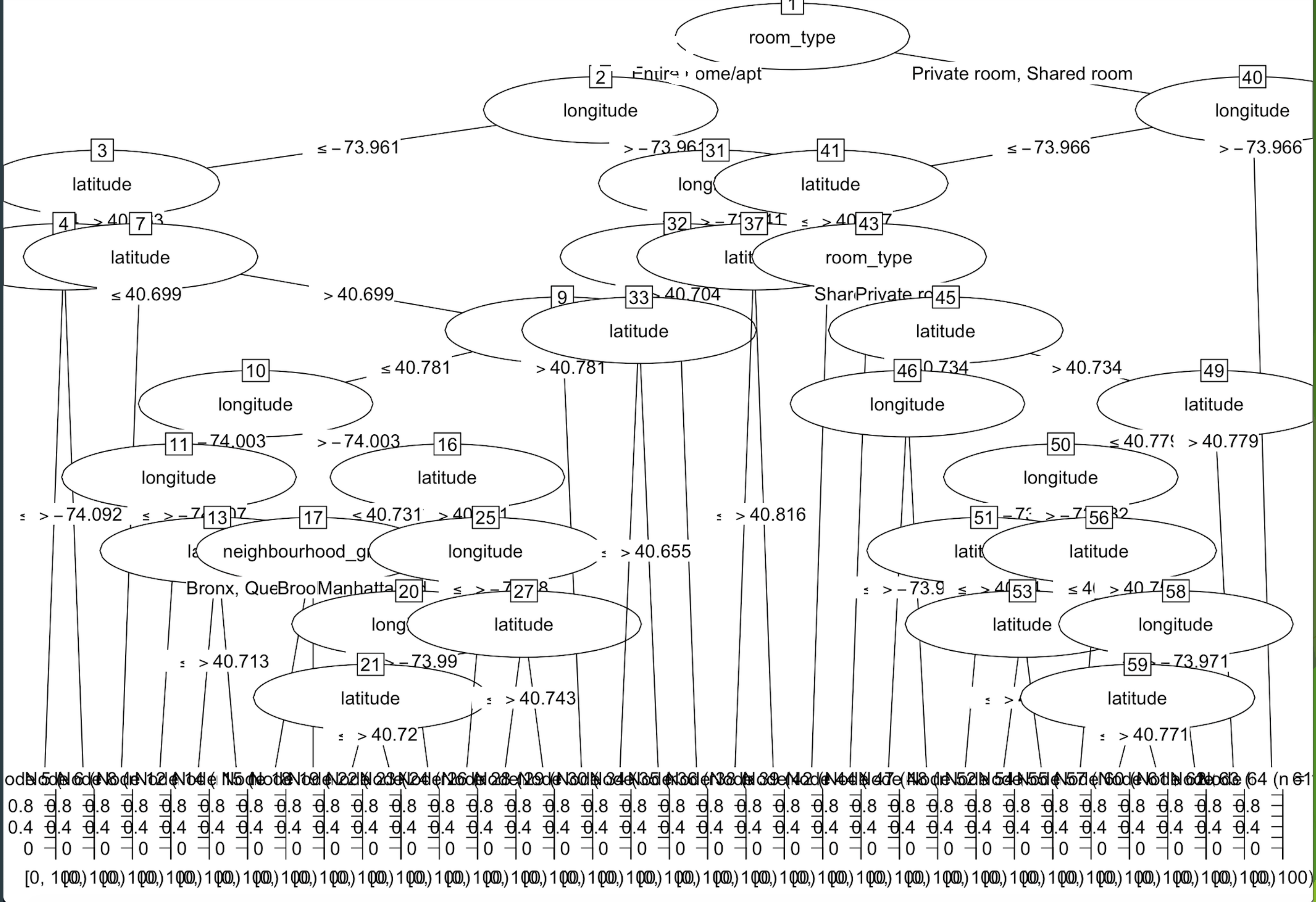
Supervised Learning

C5.0

Accuracy: 68.1%

- C5.0 is a classification algorithm that C4.5 is applied to large data sets, mainly in terms of execution efficiency and memory usage.
- It works by splitting the sample based on the field that provides the maximum information game.
- Use Boosting to improve model accuracy.
- The C5.0 model is very good to fix the missing data and many input fields.

```
Confusion Matrix and Statistics

            Reference
Prediction   [0, 100) [100, 200) [200, +)
  [0, 100)       4571        903       56
  [100, 200)      926       2619      705
  [200, +)        158       1150     1133

Overall Statistics

               Accuracy : 0.681
                 95% CI : (0.6727, 0.6893)
    No Information Rate : 0.4627
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4911

 Mcnemar's Test P-Value : < 2.2e-16
```


1
room_type
2
longitude
Private room, Shared room
40
longitude
3
latitude
≤ −73.961
31
41
latitude
≤ −73.966
> −73.966
4
7
latitude
32
37
43
room_type
≤ 40.699
> 40.699
9
33
latitude
45
latitude
10
longitude
≤ 40.781
> 40.781
46
longitude
> 40.734
49
latitude
11
> −74.003
16
latitude
50
longitude
> 40.779
longitude
> −74.092
13
17
neighbourhood_gr
25
longitude
≤ 40.731
> 40.816
51
56
latitude
> 40.655
20
27
latitude
53
latitude
58
longitude
> 40.713
21
latitude
> −73.99
> 40.743
59
latitude
> −73.971
> 40.72
> 40.771

Conclusion

- We should choose the reasonable model based on the characteristics of the data. The difference in the results of the same data using different models may be very large.

- The training set with more data can make model be more accurate.

- When we filtered the data first, we have initially selected several important factors that may affect rental prices

- It can be known from the results predicted by the models, the location(neighborhood group) and neighborhood are the two main factors leading to different rental prices.

- The effect of room type on rental prices is obvious.

Future Work

- We can estimate a reasonable price based on various information about the house for new landlords.

- We can quickly match suitable house according to tenants' various requirements for house.

- When more and more features are added to the house, we can try new machine learning techniques to discover new features affecting rental prices.