Airbnb Analysis

- based on Central Area, Seattle

AD 699 Data Mining for Business Analytics
Instructor: Greg Page
Presented by: Farah Eid
Fengnian Zhao
Kaming Yip
Shangze Li
Xinran Chen
Date: August 7, 2019

Agenda

Project Overview

- Neighborhood Introduction
- Project Goal and Objectives

Data Preparation & Exploration

- Missing Values
- Summary Statistics
- Visualization

Predication

- Multiple Regression

Classification

- K- nearest neighbors
- Naive Bayes
- Classification Tree

Clustering

- K-mean Analysis

Conclusion

- Overview
- Outlook

Project Overview

Neighborhood Introduction

“The **Central Area** is a mostly residential district in Seattle located east of First Hill (12th Avenue and Rainier Avenue); west of Madrona , Leschi and Mt. Baker; south of Capitol Hill, and north of Rainier Valley. Historically, the Central District has been one of Seattle's most racially and ethnically diverse neighborhoods,[2] and was once the center of Seattle's black community and a major hub of African-American businesses.[3]”

Main Objectives

1. Understand the nature of Airbnb listings and hosts in Central Area Neighborhood
2. Help potential renters have a better renting experience by using our models and findings!

1

Data Preparation & Exploration

Missing Values

Method 1: Delete the inefficient column(s)

```
> # square_feet
> summary(CentralArea$square_feet)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
      1     850    1200    1050    1300    2100     360
> CentralArea <- select(CentralArea, -(square_feet))
```

In the case of a particular column named “square_feet”, we find out that, among 369 rows, only 9 of them provide this information, which is far too inefficient to extract reliable information.

Method 2: Imputation

In most of the cases in this project, imputation, instead of deleting whether a specific column or several rows, is one of the most prevailing methods to handle the missing values

Summary Statistics

Q1. How is the price of Airbnb located in Central Area, Seattle?

Q2. How many levels exist for host performance metrics?

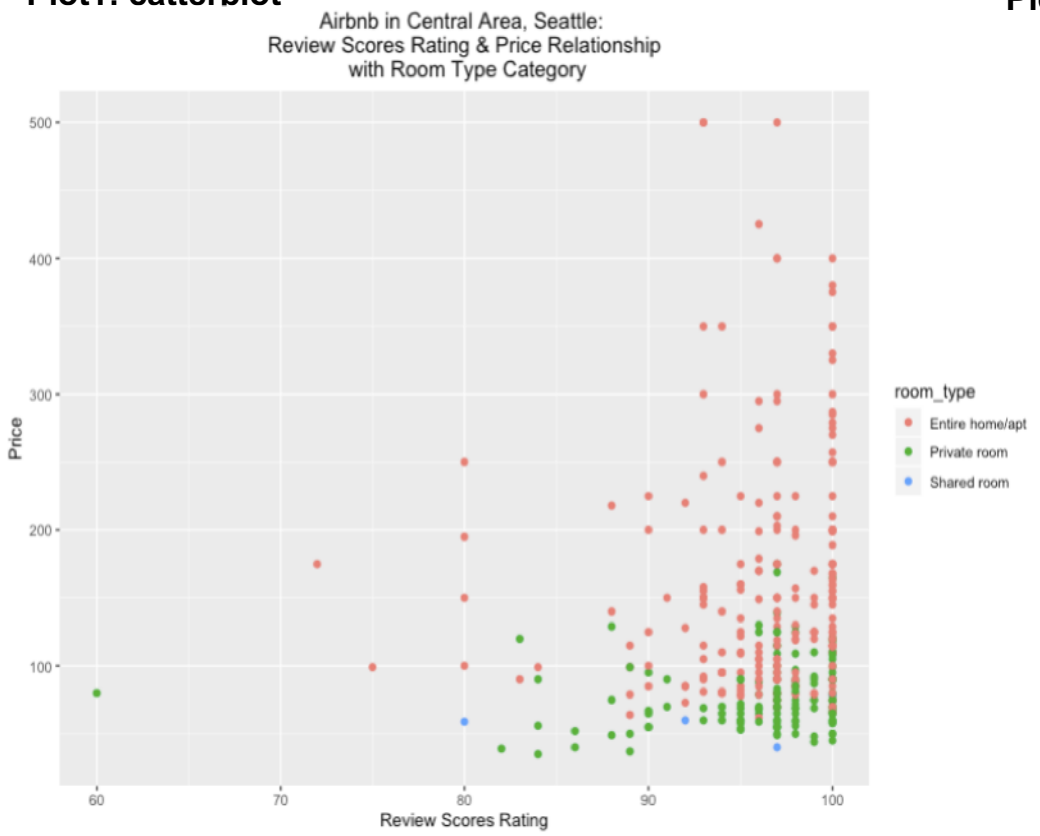
Q3. Among the overall Airbnb hosts in this neighborhood, how many of them are superhosts?

Q4. How is the number and the price of each property type in the neighborhood?

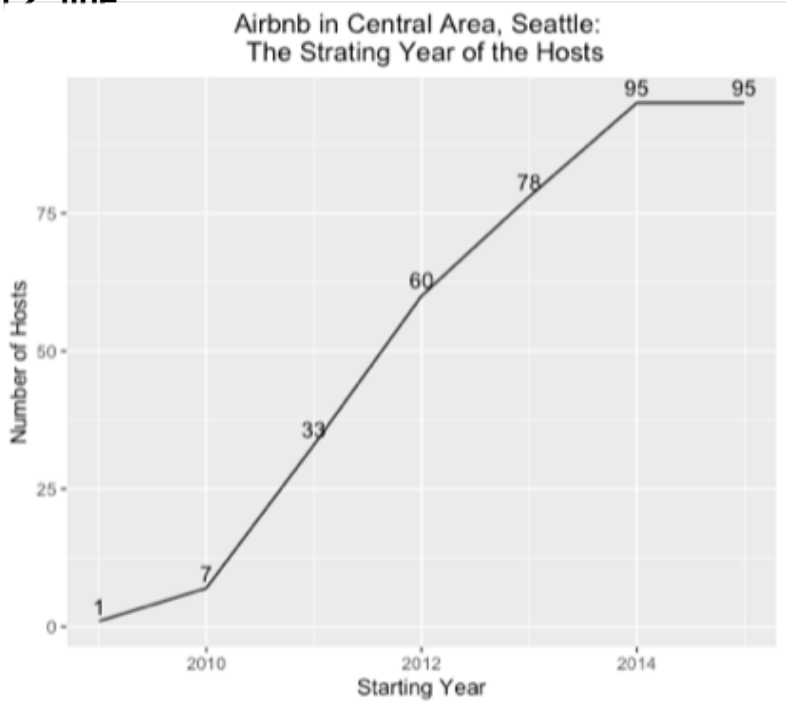
Q5. What is the trend of the host number in this neighborhood when time went by?

Visualization

Plot1: catterplot

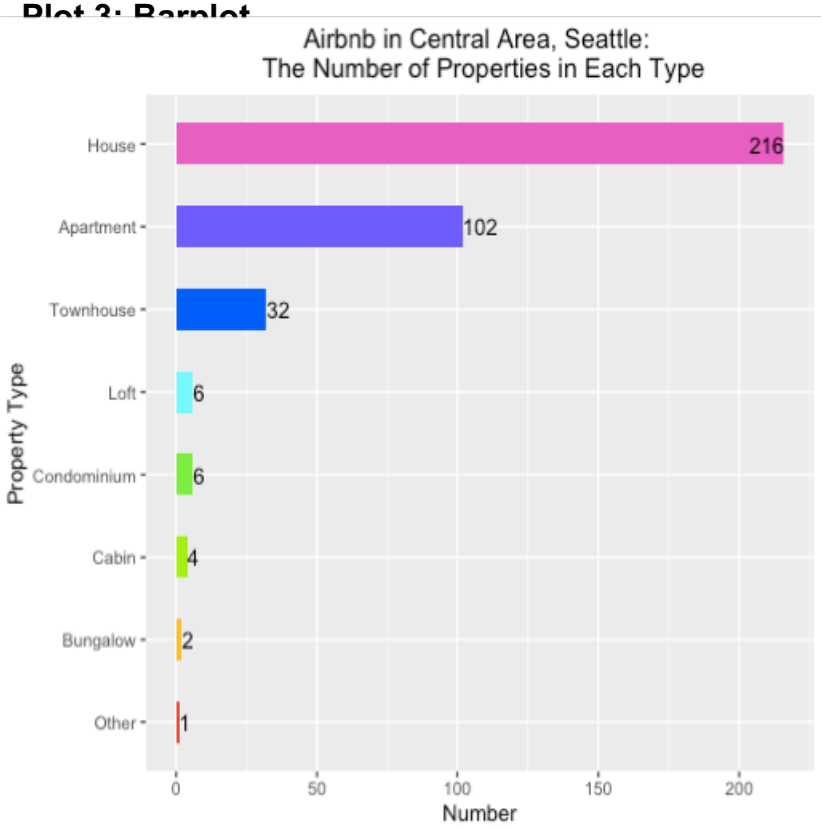


Plot 2: line

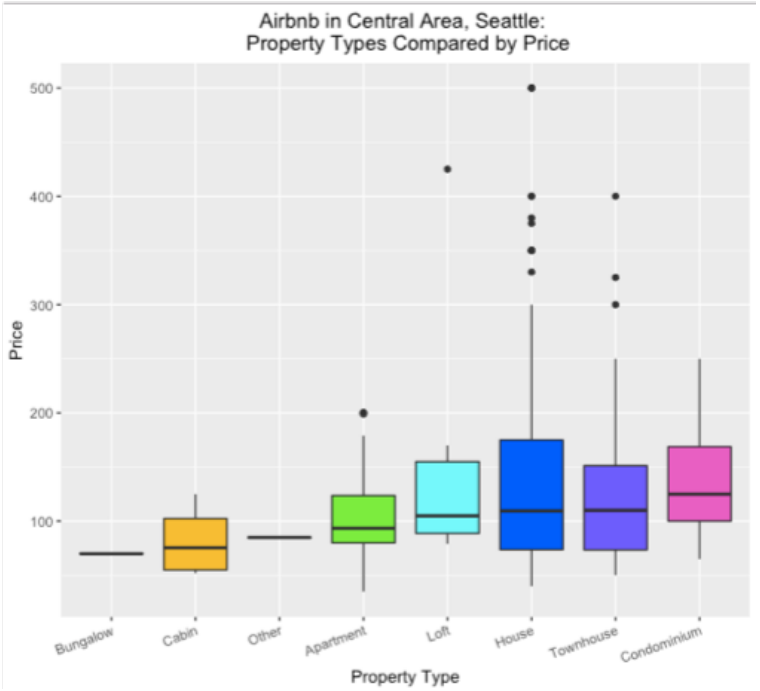


Visualization

Plot 3: Barplot

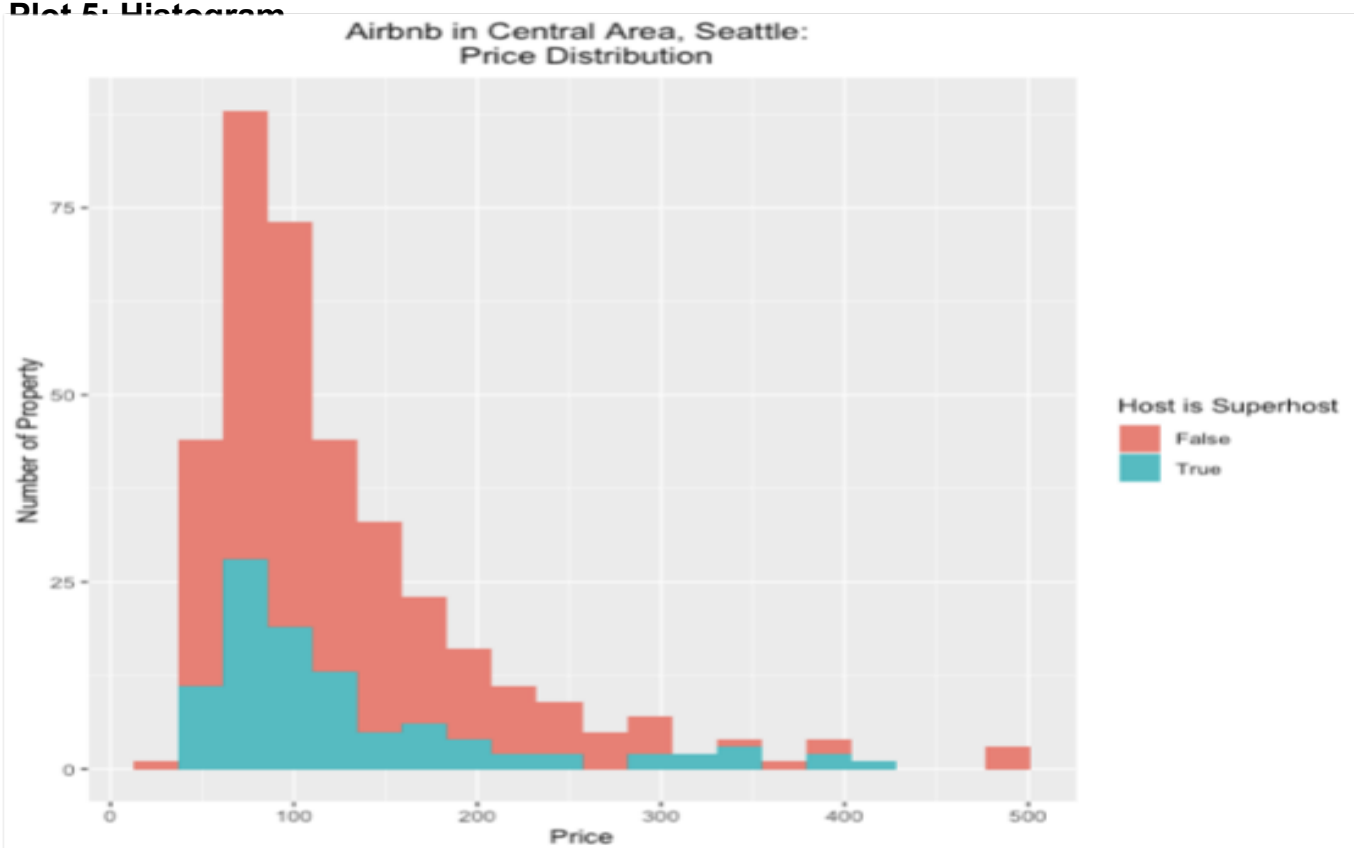


Plot 4: Boxplot



Visualization

Plot 5: Histogram

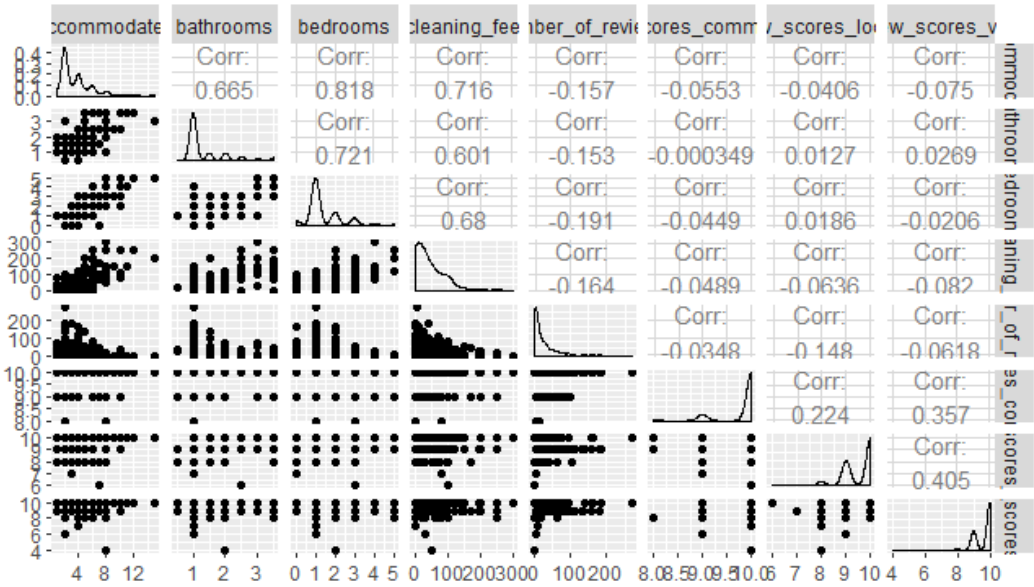


2

Prediction

Prediction
- Multiple Linear Regression

- **Selecting Significant Predictors**



- **Final Predictors**

```
Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  -50.29524   78.54209  -0.640    0.523
room_typePrivate room        -33.25416    6.62462  -5.020 1.09e-06 ***
room_typeShared room         -38.82395   29.54296  -1.314    0.190
bathrooms                     28.09801    5.91363   4.751 3.71e-06 ***
bedrooms                      22.90252    4.80911   4.762 3.54e-06 ***
cleaning_fee                   0.44361    0.08285   5.355 2.21e-07 ***
review_scores_communication   14.14865    8.61663   1.642    0.102
review_scores_value           -4.38679    5.45119  -0.805    0.422
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 41.04 on 213 degrees of freedom
Multiple R-squared:  0.7318,    Adjusted R-squared:  0.723
F-statistic: 83.02 on 7 and 213 DF,  p-value: < 2.2e-16
```

- **Accuracy**

```
> accuracy(Prediction_TrainPre, Prediction_Train$price)
                     ME      RMSE       MAE        MPE      MAPE
Test set -3.537555e-13 40.29146 28.09696 -7.015867 23.31265
```

3

Classification

Classification
- k-nearest Neighbors

- K-nearest neighbors

1. Identify significant predictors:
2. Normalization and Data Prep
3. Choose Optimal K:
4. Create Test Neighborhood:
5. Test Neighborhood:

Classification
- k-nearest Neighbors

Predict Cancellation policy

- Question for audience:
  - Which predictors do you think are important?
- Predictors: cleaning fee, total host listings, security deposit

Low Model Accuracy

- Numerical predictors did not have a high impact on cancellation policy
- 57.04%

Filter

	k	accuracy
11	11	0.5704698
8	8	0.5637584
13	13	0.5637584
12	12	0.5570470
10	10	0.5503356
7	7	0.5436242
14	14	0.5436242
9	9	0.5369128
5	5	0.5033557
1	1	0.4966443
4	4	0.4899329
6	6	0.4765101
3	3	0.4429530

```
> varImp(model,scale=TRUE)
                           Overall
accommodates            13.1707995
availability_365         7.5315762
availability_60         15.6889960
availability_90         11.3105452
beds                    14.5663217
cleaning_fee            51.3329311
extra_people            14.0081812
guests_included         13.1783735
host_total_listings_count 31.3834072
maximum_nights          15.8105985
minimum_nights           5.7309754
price                   17.7757519
review_scores_accuracy   1.7270597
review_scores_location   0.8813474
review_scores_rating     8.3272604
reviews_per_month       28.2368118
security_deposit        30.9291199
weekly_price             6.9776698
```

Classification
- k-nearest Neighbors

Create Test Neighborhood

- Test<-data.frame(cleaning_fee=195,host_total_listings_count=6,security_deposit=995)

Test Neighborhood

```
> nn2
[1] strict
attr(,"nn.index")
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
[1,]  158    6  160  184   50  117  218  131    3    66   196
attr(,"nn.dist")
     [,1] [,2]     [,3]     [,4]     [,5]     [,6]    [,7]
[1,]    0    0 2.108818 2.695255 3.134691 3.194627 3.24229
         [,8]     [,9]    [,10]    [,11]
[1,] 3.333298 3.376756 3.376756 3.443967
Levels: strict
> View(Test.norm)
```

Classification
- Naive Bayes

```
> summary(Bayse$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   35.0    75.0   100.0   128.3   155.0   500.0
> summary(Bayse$extra_people)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0     0.0     0.0    11.7    20.0    75.0
> summary(Bayse$number_of_reviews)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    2.00   10.00   22.88   30.00  270.00
> summary(Bayse$review_scores_rating)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  60.00   95.00   97.00   96.14  100.00  100.00
> summary(Bayse$review_scores_accuracy)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  6.000  10.000  10.000   9.794  10.000  10.000
> summary(Bayse$review_scores_location)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  6.000   9.000  10.000   9.558  10.000  10.000
```

This is our non-factorial data, which turns these data into factors

```
'data.frame':   369 obs. of  15 variables:
 $ host_is_superhost              : Factor w/ 3 levels "","f","t": 2 2 2 2 2 2 2 3 2 ...
 $ host_has_profile_pic           : Factor w/ 3 levels "","f","t": 3 3 3 3 3 3 3 3 3 ...
 $ host_identity_verified         : Factor w/ 3 levels "","f","t": 3 3 3 3 3 3 3 3 3 ...
 $ is_location_exact              : Factor w/ 2 levels "f","t": 1 2 2 2 2 2 2 2 2 ...
 $ price                          : num  90 88 125 145 275 60 95 140 250 225 ...
 $ extra_people                   : num  0 0 20 25 0 15 50 0 25 25 ...
 $ has_availability               : Factor w/ 1 level "t": 1 1 1 1 1 1 1 1 1 ...
 $ number_of_reviews              : int  0 8 132 3 2 7 93 24 7 11 ...
 $ review_scores_rating           : int  97 98 95 93 100 94 94 88 97 95 ...
 $ review_scores_accuracy         : int  10 10 10 8 9 9 10 10 9 9 ...
 $ review_scores_location         : int  10 10 9 10 9 9 9 9 10 9 ...
 $ instant_bookable               : Factor w/ 2 levels "f","t": 1 2 1 1 1 1 1 1 1 1 ...
 $ cancellation_policy            : Factor w/ 3 levels "flexible","moderate",..: 3 2 2 3 1
 $ require_guest_profile_picture  : Factor w/ 2 levels "f","t": 1 1 1 1 2 1 1 1 1 1 ...
 $ require_guest_phone_verification: Factor w/ 2 levels "f","t": 1 1 1 1 2 1 1 1 1 1 ...
>
```

Classification
- Naive Bayes

```
> Bayse_T

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        f         t
0.8823529 0.1176471

Conditional probabilities:
   host_is_superhost
Y                   f         t
  f 0.0000000 0.7743590 0.2256410
  t 0.0000000 0.7307692 0.2692308

   host_has_profile_pic
Y     f t
  f 0 0 1
  t 0 0 1

   host_identity_verified
Y                   f         t
  f 0.0000000 0.1897436 0.8102564
  t 0.0000000 0.2692308 0.7307692

   is_location_exact
Y            f         t
  f 0.07692308 0.92307692
  t 0.00000000 1.00000000

   price
Y   Low Price Favorable Price High Price
  f 0.5076923       0.2564103  0.2358974
  t 0.5769231       0.2307692  0.1923077
```

```
   extra_people
Y      Small    Medium     Large
  f 0.2065217 0.3043478 0.4891304
  t 0.4375000 0.2500000 0.3125000

   has_availability
Y   t
  f 1
  t 1

   number_of_reviews
Y   Especially Less      Less  Abundant
  f       0.5641026 0.2307692 0.2051282
  t       0.3846154 0.3076923 0.3076923

   review_scores_rating
Y   Negative Comment Ordinary Comment Positive Comment
  f        0.2974359        0.2461538        0.4564103
  t        0.5384615        0.1538462        0.3076923

   review_scores_accuracy
Y   Negative Comment Ordinary Comment Positive Comment
  f       0.02051282       0.18461538       0.79487179
  t       0.00000000       0.34615385       0.65384615

   review_scores_location
Y   Negative Comment Ordinary Comment Positive Comment
  f       0.05641026       0.31282051       0.63076923
  t       0.11538462       0.46153846       0.42307692

   cancellation_policy
Y    flexible  moderate    strict
  f 0.3487179 0.2923077 0.3589744
  t 0.1538462 0.5384615 0.3076923

   require_guest_profile_picture
Y           f         t
  f 0.94358974 0.05641026
  t 0.88461538 0.11538462
```

```
require_guest_phone_verification
Y          f         t
  f 0.93846154 0.06153846
  t 0.88461538 0.11538462
```

Naive Bayes Model

Classification - Naive Bayes

```
> confusionMatrix(Bayse_Train$instant_bookable,Bayse_T_PC)
Confusion Matrix and Statistics

          Reference
Prediction   f   t
         f 187   8
         t  19   7

               Accuracy : 0.8778
                 95% CI : (0.8272, 0.9179)
    No Information Rate : 0.9321
    P-Value [Acc > NIR] : 0.99887

                  Kappa : 0.2794

 Mcnemar's Test P-Value : 0.05429

            Sensitivity : 0.9078
            Specificity : 0.4667
         Pos Pred Value : 0.9590
         Neg Pred Value : 0.2692
             Prevalence : 0.9321
         Detection Rate : 0.8462
   Detection Prevalence : 0.8824
      Balanced Accuracy : 0.6872

       'Positive' Class : f
```

```
> confusionMatrix(Bayse_Valid$instant_bookable,Bayse_V_PC)
Confusion Matrix and Statistics

          Reference
Prediction   f   t
         f 122   6
         t  18   2

               Accuracy : 0.8378
                 95% CI : (0.7684, 0.8933)
    No Information Rate : 0.9459
    P-Value [Acc > NIR] : 1.00000

                  Kappa : 0.0711

 Mcnemar's Test P-Value : 0.02474

            Sensitivity : 0.8714
            Specificity : 0.2500
         Pos Pred Value : 0.9531
         Neg Pred Value : 0.1000
             Prevalence : 0.9459
         Detection Rate : 0.8243
   Detection Prevalence : 0.8649
      Balanced Accuracy : 0.5607

       'Positive' Class : f
```

Our model prediction accuracy reaches 87.78%

The accuracy of the verification set is 83.78%, which is close to 87.78%. The difference between our validation set and the training set is 4%. Prove that our model is very accurate.

Classification
- Naive Bayes

host_is_superhost	host_has_profile_pic	host_identity_verified
t	t	t

is_location_exact	price	extra_people	has_availability	number_of_reviews
t	High Price	Medium	f	Abundant

review_scores_rating	review_scores_accuracy	review_scores_location	instant_bookable
Positive Comment	Positive Comment	Ordinary Comment	t

cancellation_policy	require_guest_profile_picture	require_guest_phone_verification
moderate	t	t

In the end, the probability of getting an instant bookable is 47.4%. The probability of not getting instant bookable is 52.6%.

```
> New_Bayse2
                f         t
[1,] 0.5256182 0.4743818
```

Classification
- Classification Tree

Variable: property type, room type
accommodates, guests included
bathrooms, bedrooms, beds
price, extra people
review scores rating, review scores of cleanliness

Classification
- Classification Tree

369 records -> 319 records

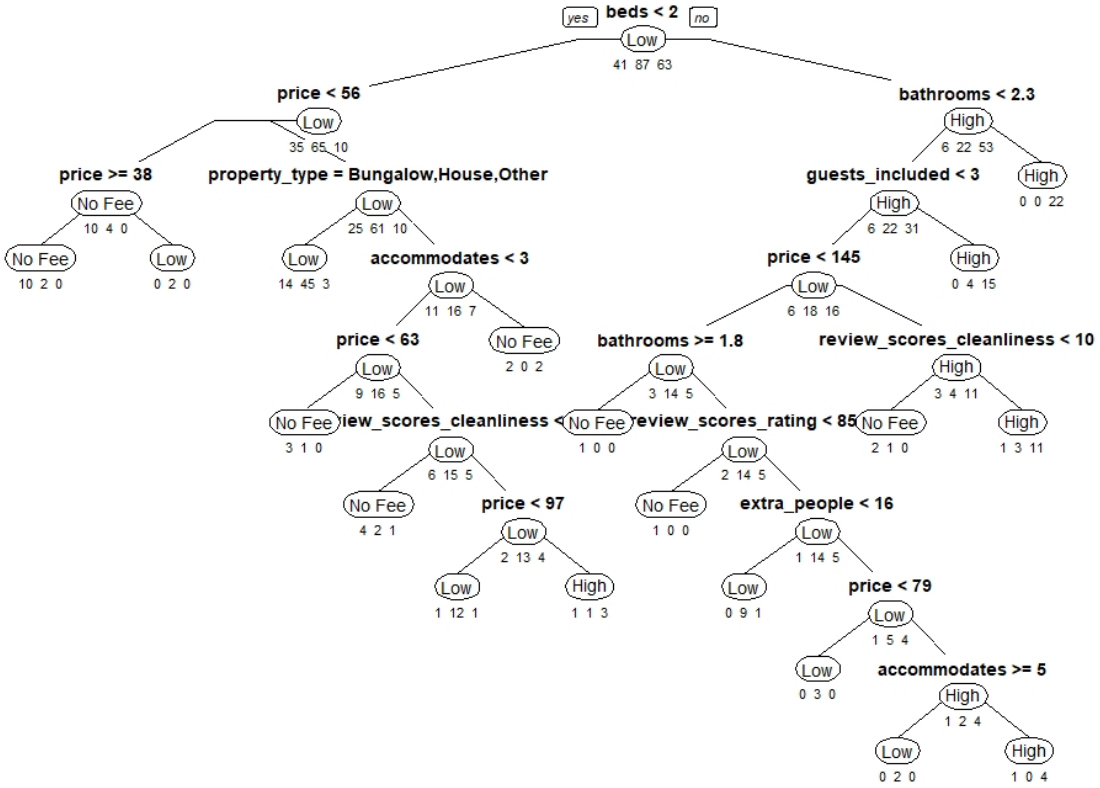
```
> map(Tree, ~sum(is.na(.)))
$property_type
[1] 0

$room_type
[1] 0

$accommodates
[1] 0

$bathrooms
[1] 0

$bedrooms
[1] 0

$beds
[1] 0

$price
[1] 0

$cleaning_fee
[1] 0

$guests_included
[1] 0

$extra_people
[1] 0

$review_scores_rating
[1] 50

$review_scores_cleanliness
[1] 50
```

Classification
- Classification Tree

```
> table(Tree2$cleaning_fee)

No Fee     Low    High
    73     144     102
```

3 groups: No Fee (0)
Low (1-50,cheap, acceptable)
High (51-250, expensive, unacceptable)

Classification
- Classification Tree

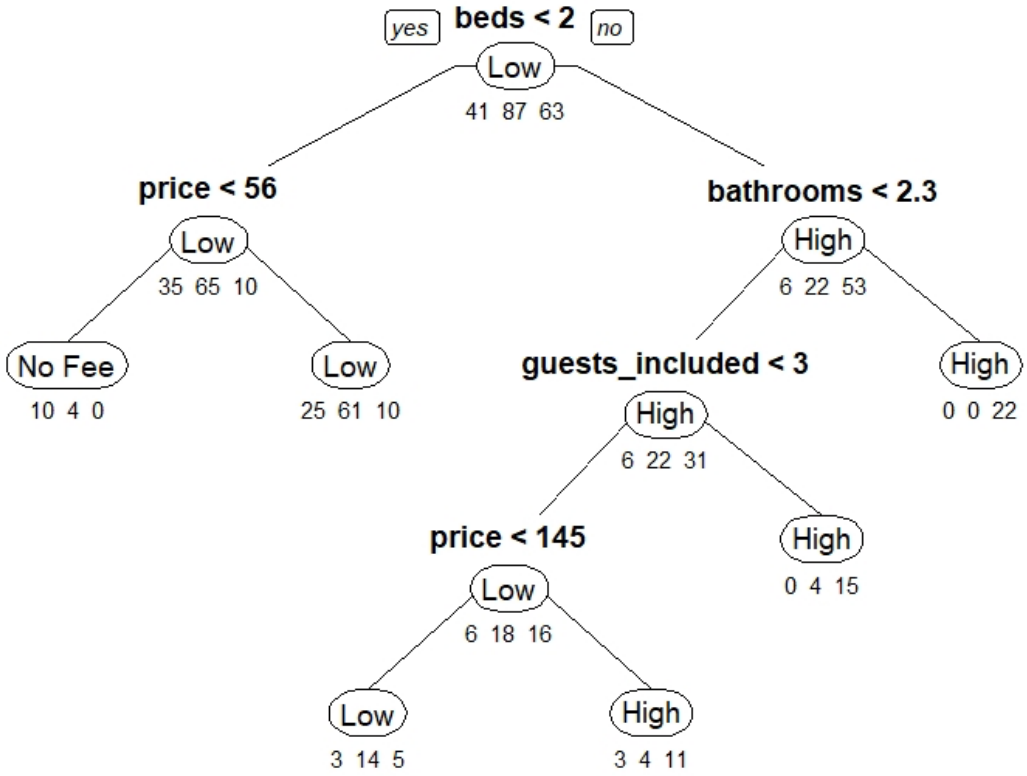


Classification
- Classification Tree

```
> printcp(cv.ct)

Classification tree:
rpart(formula = cleaning_fee ~ ., data = Tree.train, method = "class",
    cp = 1e-05, minsplit = 5, xval = 191)

Variables actually used in tree construction:
[1] accommodates              bathrooms                 beds                      extra_people
[5] guests_included           price                     property_type             review_scores_cleanliness
[9] review_scores_rating

Root node error: 104/191 = 0.5445

n= 191

         CP nsplit rel error  xerror     xstd
1 0.2980769      0   1.00000 1.00000 0.066180
2 0.0576923      1   0.70192 0.74038 0.065185
3 0.0288462      2   0.64423 0.69231 0.064401
4 0.0192308      5   0.55769 0.59615 0.062222
5 0.0153846      7   0.51923 0.66346 0.063834
6 0.0128205     12   0.44231 0.75000 0.065319
7 0.0096154     15   0.40385 0.81731 0.066041
8 0.0064103     22   0.33654 0.85577 0.066290
9 0.0000100     34   0.25000 0.90385 0.066435
```

cp=0.0192308

Classification
- Classification Tree



Classification
- Classification Tree

```
> confusionMatrix(Tree.pruned.train.ct.pred,Tree.train$cleaning_fee)
Confusion Matrix and Statistics

          Reference
Prediction No Fee Low High
    No Fee      10   4    0
    Low         28  75   15
    High         3   8   48

Overall Statistics

               Accuracy : 0.6963
                 95% CI : (0.6258, 0.7606)
    No Information Rate : 0.4555
    P-Value [Acc > NIR] : 1.520e-11

                  Kappa : 0.4947

 Mcnemar's Test P-Value : 3.793e-05

Statistics by Class:

                     Class: No Fee Class: Low Class: High
Sensitivity                0.24390     0.8621      0.7619
Specificity                0.97333     0.5865      0.9141
Pos Pred Value             0.71429     0.6356      0.8136
Neg Pred Value             0.82486     0.8356      0.8864
Prevalence                 0.21466     0.4555      0.3298
Detection Rate             0.05236     0.3927      0.2513
Detection Prevalence       0.07330     0.6178      0.3089
Balanced Accuracy          0.60862     0.7243      0.8380
```

```
> confusionMatrix(Tree.pruned.valid.ct.pred,Tree.valid$cleaning_fee)
Confusion Matrix and Statistics

          Reference
Prediction No Fee Low High
    No Fee      2   4    0
    Low        28  48    8
    High        2   5   31

Overall Statistics

               Accuracy : 0.6328
                 95% CI : (0.5431, 0.7162)
    No Information Rate : 0.4453
    P-Value [Acc > NIR] : 1.517e-05

                  Kappa : 0.3937

 Mcnemar's Test P-Value : 0.000122

Statistics by Class:

                     Class: No Fee Class: Low Class: High
Sensitivity                0.06250     0.8421      0.7949
Specificity                0.95833     0.4930      0.9213
Pos Pred Value             0.33333     0.5714      0.8158
Neg Pred Value             0.75410     0.7955      0.9111
Prevalence                 0.25000     0.4453      0.3047
Detection Rate             0.01562     0.3750      0.2422
Detection Prevalence       0.04688     0.6562      0.2969
Balanced Accuracy          0.51042     0.6675      0.8581
```

4

Clustering

Choose variables

First, we chose “accommodates”, “bathrooms”, “bedrooms”, “beds”, “price” and “review_scores_rating” as the variables which are meaningful to our clustering step.

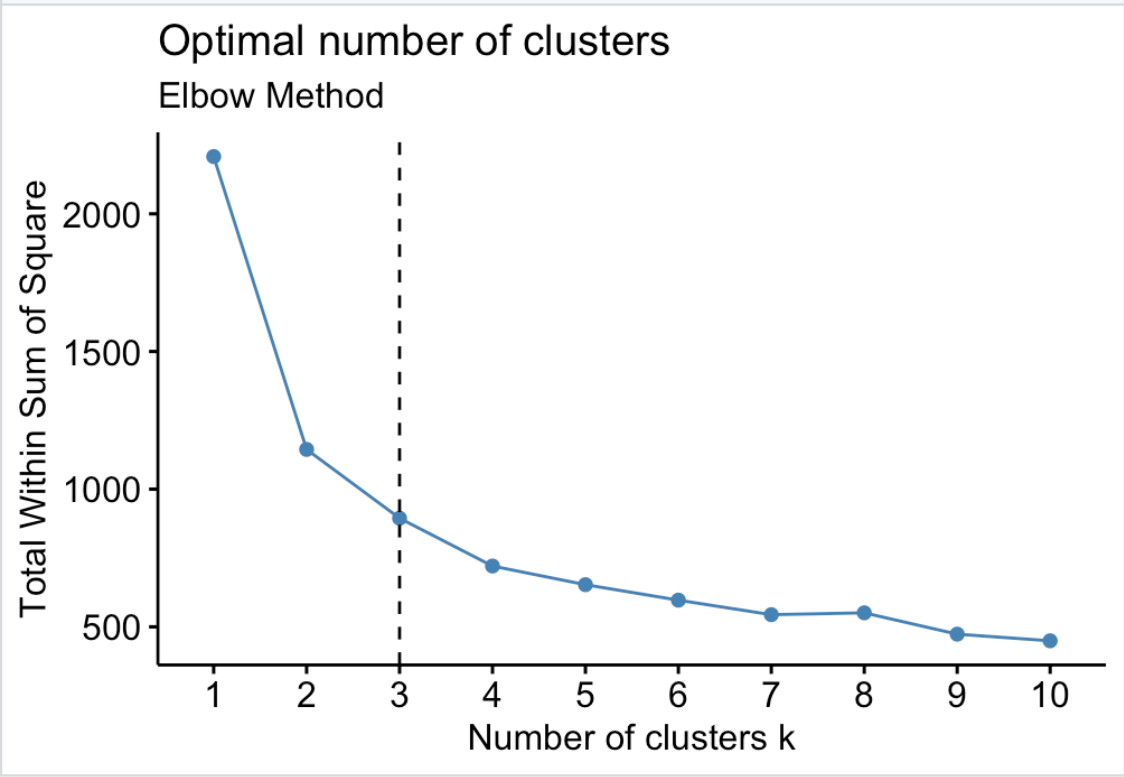
```
> CA2 <- CentralArea[, c(1,34,35,36,37,40,58)]
> CA3 <- drop_na(CA2)
> anyNA(CA3)
[1] FALSE
> CA.cluster <- CA3
> names(CA.cluster)
[1] "id"                  "accommodates"        "bathrooms"           "bedrooms"            "beds"
[6] "price"               "review_scores_rating"
```

Normalize the data

Second, we name the “id” as the row in the frame and normalized the data for the later clustering.

	accommodates	bathrooms	bedrooms	beds	price	review_scores_rating
10035644	-0.7141233	-0.5657699	-0.4588410	-0.6423402	-0.477678628	0.17886778
8293287	-0.7141233	-0.5657699	-0.4588410	-0.6423402	-0.502597468	0.38773553
910784	0.7051487	-0.5657699	0.5927883	0.1451134	-0.041598931	-0.23886773
7071222	0.7051487	0.8740170	0.5927883	0.1451134	0.207589468	-0.65660323
8687716	1.1782393	3.0336973	1.6444176	0.9325670	1.827314060	0.80547103
7430679	-1.1872139	-0.5657699	-0.4588410	-0.6423402	-0.851461226	-0.44773548

Determine the number of clusters



Third, we choose k-mean clustering and determined the optimal number of 3 the clusters by “fviz_nbcluster” function.The figure shows there tends to be stable when the number of clusters is larger than 3.

K-means analysis

We labeled the three cluster as “Party Preferred”, “Family Preferred” and “Couple or Business”.

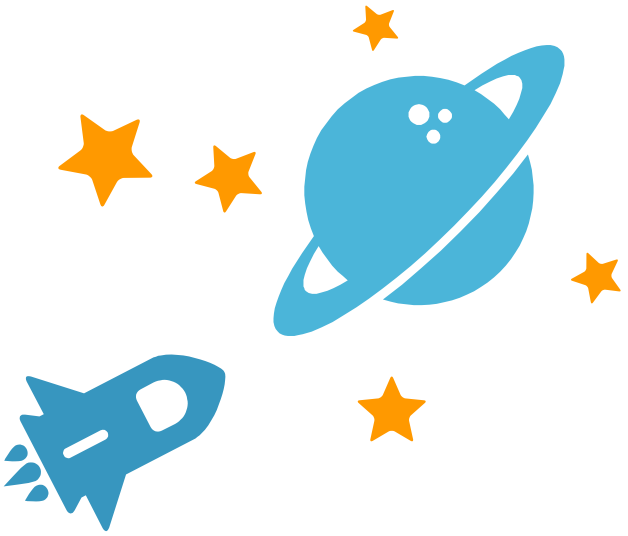
```
> km$centers
  accommodates   bathrooms   bedrooms       beds      price review_scores_rating
1    2.0497221  2.27591471  2.1702322  2.1344698  2.1424062           0.15138518
2    0.6773198  0.09483819  0.6422768  0.5434723  0.4538462           0.04126079
3   -0.5506570 -0.38433335 -0.5571640 -0.5175000 -0.4877576          -0.03764148
> dist(km$centers)
         1        2
2 3.790761
3 5.938351 2.279188
```

5

Conclusion

Conclusion

1. Importance of Data Cleaning
2. Insight from summary statistics and visualization
3. How our work can benefit future renters :)



THANK YOU

FOR YOUR PATIENCE!