Milestone 2: Report

Predictive Analysis of Los Angeles Airbnb Data

INST 737: Introduction to Data Science

Team 1: Members

Samarjith Jawaharlal Sathyanarayan

Sheryl Mathias

Ruchira Kapoor

Question 1. Linear Regressions

For research questions one [RQ 1] and two [RQ 2], linear regression does not apply. This is because the outcome of linear regression is typically a continuous variable. But the required outputs for questions one and two are discrete in nature, i.e., categorical in nature. Hence they are better suited as classification problems. With respect to research question three [RQ 3], the required output is continuous in nature and hence linear regression can be applied to it.

Independent Variable	Intercept	Coefficient	Predictive?	Type
host_response_rate	145.37282	0.03250	No	Integer
host_acceptance_rate	157.6918	-0.1045	No	Integer
accommodates	27.6658	36.4648	Yes	Integer
bathrooms	11.097	102.816	Yes	Integer
bedrooms	22.2073	97.1243	Yes	Integer
beds	59.6953	49.2761	Yes	Integer
number_of_reviews	151.36489	-0.22712	Yes	Integer
reviews_per_month	156.3791	-6.6605	Yes	Integer
review_scores_rating	21.2161	1.3040	Yes	Integer
host_is_superhost	146.724	4.620	No	Factor
host_has_profile_pic	160.14	-12.72	No	Factor

host_identity_verified	139.953	10.704	Yes	Factor
instant_bookable	151.611	-23.686	Yes	Factor

Table 1

For this question the variable “price” is considered as the dependent variable. We are going to choose a number of numeric and factor variables, both alone and in combination, as the independent variables. We start off question three by performing data cleaning on the “listing” dataset.

1. The variable price is a factor type and has a dollar symbol at the beginning. The dollar symbol is stripped from all the values and the column is converted into an integer type variable.
2. All null and “0” values are removed from the price column.
3. The security deposit and cleaning fee columns are also cleaned in the same way as the price column.
4. Host response rate and host acceptance rate columns are factor variables. They are stripped of their last character and are converted into an integer type.
5. All possible numeric datasets are chosen as independent variables.
6. In addition to all the numeric variables, categorical variables that have three or less than three levels are also chosen as the independent variables.
7. In total fifteen variables are considered, including the dependent variable.
8. The dataset is prunes only to contain thee fifteen variables.
9. The dataset is now divided into two datasets - test dataset and train dataset.
10. The train dataset contains about 70% of the total rows, while the test dataset contains the rest.
11. The rows are assigned on a random basis to the two datasets.

[A]. Table 1 shows all the individual independent variables and their corresponding intercept, coefficient, type and whether it is a predictive variable or not. According to the individual models, the independent variable with the most predictive feature is the bathrooms variable.

Since there are too many independent variables, we are going to consider only the top three variables that are the most predictive features. This results in the variables bathrooms, bedrooms and beds being considered. According to correlation, it can be observed that the variable bedrooms is the most accurate. It can be observed from the plots that the residuals are not completely normalized. This can be stated as one of the limitations. This can occur when the dataset considered has a large number of rows.

[B]. After performing simple linear regression for each independent variable, we now take all the independent variables together. In all, four models are created

1. Model 1 -> Independent variables include both categorical and numeric variables.
2. Model 2 -> Only the significant variables are chosen based on Model 1.
3. Model 3 -> Only numeric independent variables are considered.
4. Model 4 -> Only the significant variables are chosen based on Model 3.

It is observed that when we combine multiple independent variables to predict the dependent variable, the accuracy of the model increases.

```
Residuals:
    Min      1Q  Median      3Q     Max
-400.49  -43.56  -12.51   31.04  811.19

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             -132.54398   23.42967  -5.657 1.57e-08 ***
host_response_rate        -0.09152    0.06069  -1.508 0.131597
host_acceptance_rate       0.04901    0.04286   1.143 0.252876
accommodates              21.70374    0.61841  35.096  < 2e-16 ***
bathrooms                 30.38291    1.43675  21.147  < 2e-16 ***
beds                      -9.60264    0.91405 -10.506  < 2e-16 ***
bedrooms                  41.34638    1.39928  29.548  < 2e-16 ***
guests_included            5.46076    0.64486   8.468  < 2e-16 ***
number_of_reviews          0.04753    0.02486   1.912 0.055936 .
reviews_per_month         -4.74133    0.49907  -9.500  < 2e-16 ***
review_scores_rating       1.31909    0.09800  13.460  < 2e-16 ***
host_is_superhostt         7.54464    2.00156   3.769 0.000164 ***
host_has_profile_pict      6.08122   20.49420   0.297 0.766678
host_identity_verifiedt    3.01201    1.86735   1.613 0.106774
instant_bookablet        -14.57228    2.05526  -7.090 1.41e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 84.23 on 12305 degrees of freedom
  (5615 observations deleted due to missingness)
Multiple R-squared:  0.5342,    Adjusted R-squared:  0.5337
F-statistic:  1008 on 14 and 12305 DF,  p-value: < 2.2e-16
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-392.20  -43.15  -12.39   30.86  807.77 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)         -124.21699    8.52389 -14.573  < 2e-16 ***
accommodates          21.68114    0.59387  36.508  < 2e-16 ***
bathrooms             29.98802    1.36391  21.987  < 2e-16 ***
beds                  -9.79118    0.88126 -11.110  < 2e-16 ***
bedrooms              40.92760    1.32397  30.913  < 2e-16 ***
guests_included        5.66451    0.61220   9.253  < 2e-16 ***
reviews_per_month     -4.37041    0.41893 -10.432  < 2e-16 ***
review_scores_rating   1.27958    0.08921  14.343  < 2e-16 ***
host_is_superhostt     8.59386    1.87994   4.571 4.89e-06 ***
instant_bookablet    -13.60139    1.93464  -7.030 2.16e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 83.43 on 13567 degrees of freedom
  (4358 observations deleted due to missingness)
Multiple R-squared:  0.5297,	Adjusted R-squared:  0.5294 
F-statistic:  1698 on 9 and 13567 DF,  p-value: < 2.2e-16
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-396.73  -43.74  -12.21   31.03  817.85 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)         -1.351e+02  1.116e+01 -12.111  < 2e-16 ***
host_response_rate  -8.640e-02  6.032e-02  -1.432  0.15207    
host_acceptance_rate  3.225e-03  4.238e-02   0.076  0.93935    
accommodates         2.159e+01  6.187e-01  34.898  < 2e-16 ***
bathrooms            3.010e+01  1.440e+00  20.903  < 2e-16 ***
beds                -9.861e+00  9.156e-01 -10.769  < 2e-16 ***
bedrooms             4.190e+01  1.401e+00  29.905  < 2e-16 ***
guests_included      5.715e+00  6.458e-01   8.850  < 2e-16 ***
number_of_reviews    7.255e-02  2.449e-02   2.962  0.00306 ** 
reviews_per_month   -5.337e+00  4.878e-01 -10.940  < 2e-16 ***
review_scores_rating  1.470e+00  9.562e-02  15.371  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 84.46 on 12309 degrees of freedom
  (5615 observations deleted due to missingness)
Multiple R-squared:  0.5315,	Adjusted R-squared:  0.5311 
F-statistic:  1397 on 10 and 12309 DF,  p-value: < 2.2e-16
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-389.76  -43.45  -12.47   31.04  817.81 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)         -137.41582    8.36620 -16.425  < 2e-16 ***
accommodates          21.52529    0.59499  36.178  < 2e-16 ***
bathrooms             29.74593    1.36659  21.767  < 2e-16 ***
beds                 -10.03361    0.88271 -11.367  < 2e-16 ***
bedrooms              41.64249    1.32648  31.393  < 2e-16 ***
guests_included        5.79329    0.61404   9.435  < 2e-16 ***
number_of_reviews      0.06821    0.02360   2.890  0.00386 ** 
reviews_per_month     -5.37860    0.45127 -11.919  < 2e-16 ***
review_scores_rating   1.41784    0.08716  16.266  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 83.63 on 13568 degrees of freedom
  (4358 observations deleted due to missingness)
Multiple R-squared:  0.5274,	Adjusted R-squared:  0.5271 
F-statistic:  1893 on 8 and 13568 DF,  p-value: < 2.2e-16
```

The above screenshots show the coefficients for each variable of the four models. It is observed that, in all four models it is the variable bedrooms that that is the most predictive variable. The most efficient model is Model 1 followed by Model 3, Model 2 and Model 4. In terms of mean squared error, model 3 is the most accurate.

[C]. In order to prevent overfitting and improve the regression model, we fit the model with regularization with lasso penalty. The library glmnet is used for this purpose. Additional data cleaning is done.

1. Since regularization doesn’t support categorical variables, subsets of test and train datasets are created in which all the categorical variables are removed.
2. The cv.glmnet function does not handle null values. Hence all null values are removed from the newly created subsets.

It is observed that, even in this model it is the variable bedrooms that is the most predictive feature. This is in line with all the previous observations made. The efficiency is calculated, but it ranks the lowest among all the five models (although by a very small difference). Although the efficiency seems to be lower than the models without regularization, it is important to note that regularization equips the model to be more accommodative of new data points.

[D]. The same procedures from [A] to [C] are repeated three more times with different random test and train datasets. Interestingly the findings were similar to each other, in terms of the values, most predictive feature and the accuracies established.

Question 2. Logistic Regression and NB

a. ***With the knowledge gathered from Question1(b), compute a logistic regression model with respect to different sets of independent features on your training dataset and report.***

The following screenshot gives the summary of our logistic regression model, after excluding features that were not statistically significant. (p-value > 0.05)

```
glm(formula = get_more_visits ~ host_response_time + host_is_superhost +
    host_has_profile_pic + host_identity_verified + instant_bookable +
    as.integer(cleaning_fee) + cancellation_policy + require_guest_profile_picture +
    require_guest_phone_verification + charge_for_extra_people,
    family = binomial, data = train_rent_ltngs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9686   0.1909   0.4244   0.6444   1.7929

Coefficients:
                                   Estimate Std. Error z value Pr(>|z|)
(Intercept)                       -1.874e+00  3.442e-01  -5.444  5.2e-08 ***
host_response_timeN/A              3.891e-03  1.059e-01   0.037  0.97067
host_response_timewithin a day     9.266e-01  1.098e-01   8.440  < 2e-16 ***
host_response_timewithin a few hours 1.229e+00  1.065e-01  11.538  < 2e-16 ***
host_response_timewithin an hour   1.316e+00  1.044e-01  12.607  < 2e-16 ***
host_is_superhostt                 1.464e+00  9.942e-02  14.727  < 2e-16 ***
host_has_profile_pict              7.662e-01  3.306e-01   2.318  0.02047 *
host_identity_verifiedt            8.482e-01  4.222e-02  20.088  < 2e-16 ***
instant_bookablet                  1.828e-01  5.922e-02   3.087  0.00202 **
as.integer(cleaning_fee)          -2.420e-03  5.505e-04  -4.396  1.1e-05 ***
cancellation_policymoderate        1.082e+00  5.596e-02  19.332  < 2e-16 ***
cancellation_policystrict          1.051e+00  5.118e-02  20.528  < 2e-16 ***
cancellation_policysuper_strict_60 -1.148e+01  1.137e+02  -0.101  0.91955
require_guest_profile_picturet     1.991e-01  2.486e-01   0.801  0.42320
require_guest_phone_verificationt  5.814e-01  2.277e-01   2.553  0.01067 *
charge_for_extra_peopleYes         4.469e-01  4.321e-02  10.343  < 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: 19039  on 17934  degrees of freedom
Residual deviance: 15096  on 17919  degrees of freedom
AIC: 15128

Number of Fisher Scoring iterations: 10
```

<u>What's the intercept?</u>

As it can be seen from the “final_logit” logistic regression model, the intercept is -1.873846.

<u>What are the coefficients for each of the features?</u>

As it can be seen from the following screenshot, coefficient for host_response_time with a N/A value is 0.003892, for host_response_time within a day is 0.926616 and so on.

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9750   0.1843   0.4241   0.6453   1.9321

Coefficients:
                                   Estimate Std. Error z value Pr(>|z|)
(Intercept)                      -1.7513589  0.3584367  -4.886 1.03e-06 ***
host_response_timeN/A             0.0553258  0.1067069   0.518  0.60412
host_response_timewithin a day    0.9022605  0.1105030   8.165 3.21e-16 ***
host_response_timewithin a few hours 1.2304336  0.1072097  11.477  < 2e-16 ***
host_response_timewithin an hour  1.3331542  0.1051736  12.676  < 2e-16 ***
host_is_superhostt                1.5208994  0.1016511  14.962  < 2e-16 ***
host_has_profile_pict             0.6125364  0.3452620   1.774  0.07604 .
host_identity_verifiedt           0.8656711  0.0421721  20.527  < 2e-16 ***
instant_bookablet                 0.0976229  0.0589849   1.655  0.09791 .
as.integer(cleaning_fee)         -0.0025482  0.0005301  -4.807 1.53e-06 ***
cancellation_policymoderate       1.1709990  0.0564552  20.742  < 2e-16 ***
cancellation_policystrict         1.0559971  0.0509004  20.746  < 2e-16 ***
cancellation_policysuper_strict_60 -1.1484869  1.1665720  -0.984  0.32487
require_guest_profile_picturet    0.0359616  0.2359292   0.152  0.87885
require_guest_phone_verificationt 0.5626522  0.2180867   2.580  0.00988 **
charge_for_extra_peopleYes        0.4355147  0.0431534  10.092  < 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: 19101  on 17933  degrees of freedom
Residual deviance: 15131  on 17918  degrees of freedom
AIC: 15163

Number of Fisher Scoring iterations: 6
```

<u>Are they statistically significant?</u>

Host response time with a not applicable value, cancellation policy is under super strict 60 category, requirement of a guest profile picture is true and security deposit are not statistically significant features.

However, it can be observed that host response time within a day, within a few hours, within an hour, host being a super host, having a profile picture, having a verified identity, listing being instantly bookable, listing having a cleaning fee, listing having a moderate cancellation policy, listing having a strict cancellation policy, listing that requires a guest phone verification and host that charges for extra people for a listing are all statistically significant and have an effect on the listing getting a visit or not.

<u>What are the log-odds and odd ratios of the outcome for a unit increase in each independent variable?</u>

The coefficients give the change in the log-odds of the outcome for a one unit increase in the predictor.

1. For one unit increase in host response time within a day the odds of a listing getting a visit increases by a factor of 2.46. However, for a one unit increase in host response time within a day the log-odds of a listing getting a visit increases by a factor of 0.9.
2. For one unit increase in host response time within a few hours the odds of a listing getting a visit increases by a factor of 3.42. However, for a one unit increase in host response time within a few hours the log-odds of a listing getting a visit increases by a factor of 1.23.
3. For one unit increase in host response time within an hour the odds of a listing getting a visit increases by a factor of 3.79. However, for a one unit increase in host response time within an hour the log-odds of a listing getting a visit increases by a factor of 1.33.

4. For one unit increase in host being a super host the odds of a listing getting a visit increases by a factor of 4.57. However, for a one unit increase in host being a super host the log-odds of a listing getting a visit increases by a factor of 1.52.
5. For one unit increase in host having a profile picture the odds of a listing getting a visit increases by a factor of 1.84. However, for a one unit increase in host having a profile picture the log-odds of a listing getting a visit increases by a factor of 0.61.
6. For one unit increase in host having its identification verified the odds of a listing getting a visit increases by a factor of 2.37. However, for a one unit increase in host having its identification verified the log-odds of a listing getting a visit increases by a factor of 0.86.
7. For one unit increase in listing being instantly bookable verified the odds of a listing getting a visit increases by a factor of 1.1. However, for a one unit increase in listing being instantly bookable verified the log-odds of a listing getting a visit increases by a factor of 0.09.
8. For one unit increase in a listing having moderate cancellation policy the odds of a listing getting a visit increases by a factor of 3.22. However, for a one unit increase in a listing having moderate cancellation policy the log-odds of a listing getting a visit increases by a factor of 1.17.
9. For one unit increase in a listing having strict cancellation policy the odds of a listing getting a visit increases by a factor of 2.87. However, for a one unit increase in a listing having strict cancellation policy the log-odds of a listing getting a visit increases by a factor of 1.05.
10. For one unit increase in a listing having a requirement of guest phone verification the odds of a listing getting a visit increases by a factor of 1.75. However, for a one unit increase in a listing having a requirement of guest phone verification the log-odds of a listing getting a visit increases by a factor of 0.56.
11. For one unit increase in a listing having a charge for extra people the odds of a listing getting a visit increases by a factor of 1.54. However, for a one unit increase in a listing having a charge for extra people the log-odds of a listing getting a visit increases by a factor of 0.43.
12. For one unit increase in the cleaning fee for a listing the odds of a listing getting a visit increases by a factor of 0.99. However, for a one unit increase in the cleaning fee for a listing having a charge for extra people the log-odds of a listing getting a visit decreases by a factor of 0.002.

Odd ratios are as follows:

```
> exp(coef(final_logit))
                        (Intercept)              host_response_timeN/A
                          0.1735380                          1.0568849
   host_response_timewithin a day host_response_timewithin a few hours
                          2.4651693                          3.4227134
   host_response_timewithin an hour                   host_is_superhostt
                          3.7929886                          4.5763395
               host_has_profile_pict               host_identity_verifiedt
                          1.8451053                          2.3766004
                   instant_bookablet             as.integer(cleaning_fee)
                          1.1025469                          0.9974550
         cancellation_policymoderate            cancellation_policystrict
                          3.2252130                          2.8748403
   cancellation_policysuper_strict_60     require_guest_profile_picturet
                          0.3171162                          1.0366161
  require_guest_phone_verificationt          charge_for_extra_peopleYes
                          1.7553218                          1.5457584
```

Which are the most predictive features according to the training data?

The most predictive features from the model are:

Host response time within a day, within a few hours, within an hour, host being a super host, having a profile picture, having a verified identity, listing being instantly bookable, listing having a cleaning fee, listing having a moderate cancellation policy, listing having a strict cancellation policy, listing that requires a guest phone verification and host that charges for extra people for a listing.

Use the trained model to predict on your testing dataset. Explain your results.

Created a logistic regression model using three different training data sets generated randomly and used it respectively to predict three different testing data sets.

Calculated the accuracy for three different models. They are as follows:

First model accuracy: 0.8169637

Second model accuracy: 0.8114999

Third model accuracy: 0.8256797

Average accuracy for the logistic regression model is: 0.8180477 or 81.8%

By eliminating a few features like “host_has_profile_picture”, “instant_bookable” and “require_guest_profile_picture” the accuracy of the model increases by 1%. (Not a significant difference).

b. ***Next, we are going to report classification results using Naïve Bayes. Remember to categorize all your variables before running the classifier.***

Divide your dataset into training and testing set and train the classifier. Report the confusion matrix.

Cell Contents

N
N / Col Total

Total Observations in Table: 7688

predicted	actual cheap	expensive	moderate	Row Total
cheap	1937 0.752	418 0.165	1100 0.426	3455
expensive	113 0.044	1643 0.650	369 0.143	2125
moderate	525 0.204	468 0.185	1115 0.432	2108
Column Total	2575 0.335	2529 0.329	2584 0.336	7688

1. 1100 listings were predicted as “cheap” but they were actually “moderate” listings. Also, 525 listings were predicted as “moderate” but they were actually “cheap”. Thus we can say that 42.6% of the listings were incorrectly classified as “cheap” but were actually “moderate”. While, 20.4% of the listings were predicted as “moderate” but were actually “cheap”.
2. 369 listings were predicted as “expensive” but they were actually “moderate” listings. Also, 468 listings were predicted as “moderate” but they were actually “expensive”. Thus we can say that 14.3% of the listings were incorrectly classified as “expensive” but were actually “moderate”. While, 18.5% of the listings were incorrectly predicted as “moderate” but were actually “expensive”.
3. 418 listings were incorrectly predicted as “cheap” but they were actually “expensive” listings. Also, 113 listings were incorrectly predicted as “expensive” but they were actually “cheap”. Thus we can say that 16.5% of the listings were incorrectly classified as “cheap” but were actually “expensive”. While, 4% of the listings were incorrectly predicted as “expensive” but were actually “cheap”.
4. 1937 listings which is almost 75% of the listings were correctly classified as “cheap”.
5. 1643 listings which is almost 65% of the listings were correctly classified as “expensive”.
6. 1115 listings which is almost 43% of the listings were correctly classified as “moderate”.

Repeat the process above with the Laplace estimator. Do the results improve?

By using Laplace estimator, it can be observed that although the classifier didn’t improve much for predicting cheap and expensive type listings, there was an improvement in moderate type of listing.

1135 listings were correctly classified as “moderate”.

Question 3. Decision Trees and Random Forests

For implementing prediction using Decision Trees and Random Forests, we have used the data for research question 2: Predicting if a listing will get a review or not based on 9 independent variables.

a. ***Split your dataset into training and testing sets and show that the distribution after the split is similar to the original***

```
#Making the data random
set.seed(12345)
Q2_reviews_rand <- review_new[order(runif(25621)),]

#Splitting the data into Training and Testing (70:30 ratio)
Q2_reviews_rand_train = Q2_reviews_rand[1:17934, ]
Q2_reviews_rand_test = Q2_reviews_rand[17934:25621, ]
```

```
> prop.table(table(Q2_reviews_rand_test$review))

        0         1
0.2278876 0.7721124
> prop.table(table(Q2_reviews_rand_train$review))

       0        1
0.221646 0.778354
> prop.table(table(Q2_reviews_rand$review))

        0         1
0.2234885 0.7765115
```

The above code first randomizes the dataset and then divides it into a training and testing subsets in the ratio of 70:30.

After that to see that the distribution after the split is similar to the original, we use the prop.table command and as we can notice the values for the original, training and testing data set is similar. Thus, showing that the data has been split accurately.

b. ***Train a decision tree and interpret some of the results. Test the trained tree with the testing data and compare the confusion matrices obtained during the training and testing. Compute the percentage of accurately predicted values for both of the datasets.***

```r
#Creating the Decision Tree
library(C50)
str(Q2_reviews_rand_train)
Q2_reviews_rand_test$review <- as.factor(Q2_reviews_rand_test$review)
Q2_reviews_rand_train$review <- as.factor(Q2_reviews_rand_train$review)
Q2_reviews_model = C50::C5.0(Q2_reviews_rand_train[-10], Q2_reviews_rand_train$review)
Q2_reviews_model
summary(Q2_reviews_model)

#Prediction values
review_pred=predict(Q2_reviews_model,Q2_reviews_rand_test)
review_pred

#Creating the confusion matrix
library(gmodels)
CrossTable(Q2_reviews_rand_test$review,review_pred,prop.chisq=FALSE,prop.c=FALSE,prop.r=FALSE,dnn=c('actual review','predicted review'))
```

```
Decision tree:

host_is_superhost = t: 1 (2429.5/221.8)
host_is_superhost = f:
:...host_identity_verified = t: 1 (10034.1/2918.4)
    host_identity_verified = f:
    :...require_guest_phone_verification = t: 1 (71.1/16.7)
        require_guest_phone_verification = f:
        :...cancellation_policy in {flexible,super_strict_60}: 0 (2999.5/1120.7)
            cancellation_policy in {moderate,strict}: 1 (2399.8/1007.9)
```

<u>The above result can be interpreted as follows as a bunch of ifthen statements:</u>

If the host is a superhost, then the listing will get a review.

Else if the host is not a superhost and the host is verified then the listing will get a review.

If the host is a superhost and the host identity is not verified and the guest phone verification is required the listing will get a review; else if the guest phone verification is not required and the cancellation policy is either flexible or super strict the listing will not get a review; else if the cancellation policy is moderate or strict it will get a review.

<u>Structure of Decision Tree Model:</u>

Classification Tree

Number of samples: 17934

Number of predictors: 9

Tree size: 10

Confusion Matrix for Training Data:

```
Evaluation on training data (17934 cases):

            Decision Tree
          ----------------
          Size      Errors

             9 3249(18.1%)   <<


           (a)   (b)    <-classified as
          ----  ----
          1330  2645    (a): class 0
           604 13355    (b): class 1
```

The above matrix suggests that the training data has the prediction error rate of 18.1%. For all the values in the training set, it predicts 1330 listings to not get a review accurately whereas inaccurately predicts 2645 to not get a review. Similarly, it predicts 13355 listings to get a review and they do get a review and inaccurately predicts 604 listings to get a review.

Confusion Matrix for Testing Data:

```
Total Observations in Table:  7688

              | predicted review
actual review |         0 |         1 | Row Total |
--------------|-----------|-----------|-----------|
            0 |       565 |      1187 |      1752 |
              |     0.073 |     0.154 |           |
--------------|-----------|-----------|-----------|
            1 |       274 |      5662 |      5936 |
              |     0.036 |     0.736 |           |
--------------|-----------|-----------|-----------|
 Column Total |       839 |      6849 |      7688 |
--------------|-----------|-----------|-----------|
```

The testing data when used in the model created by the training data gives the above mentioned results. It can be interpreted as follows: the model predicted that 565 listings will not get a review and they did not get a review whereas it predicted that 274 listings will not get a review and they got a review. It predicted accurately that 5662 listings will get a review and inaccurately predicted that 1187 will get a review.

Comparison of Training and Testing Confusion Matrices with Computed Values:

	Accurately Predicted 0 (no review)	Accurately Predicted 1 (yes review)
Training Data	68.76%	83.46%
Testing Data	67.34%	82.66%
Difference	1.42%	0.8%

It can be observed that the difference in the accuracy percentages is not much. Thus, stating that our training model works efficiently with our training data.

c. ***Apply boosting with different number of trees and analyse the impact on the prediction results.***

<u>Boosting with trails = 7</u>

```
Evaluation on training data (17934 cases):

Trial       Decision Tree
-----     ----------------
          Size      Errors

   0         9 3249(18.1%)
   1         5 3586(20.0%)
   2         8 5242(29.2%)
   3         6 4552(25.4%)
   4         9 5064(28.2%)
   5         6 3472(19.4%)
   6         5 3358(18.7%)
boost          3125(17.4%)   <<


           (a)   (b)    <-classified as
          ----  ----
          1571  2404    (a): class 0
           721 13238    (b): class 1
```

The above result is obtained when we boost the tree by taking the number of trails as 7. The error rate drops from 18.1% to 17.4%.

<u>Boosting with trails = 5</u>

```
Trial       Decision Tree
-----     ----------------
          Size      Errors

   0         9 3249(18.1%)
   1         5 3586(20.0%)
   2         8 5242(29.2%)
   3         5 3561(19.9%)
   4         6 3563(19.9%)
boost          3231(18.0%)   <<


           (a)   (b)    <-classified as
          ----  ----
          1272  2703    (a): class 0
           528 13431    (b): class 1
```

The above result is obtained when we boost the tree by taking the number of trails as 5. The error rate drops from 18.1% to 18%.

<u>Comparison of Boosting with different trials</u>

It is observed that when the trial size is larger it decreases the error rate more. Thus, increasing the accuracy of the classifiers.

d. ***Bagging and Random Forests. Compare the features. What is the most important feature in each random forest?***

<u>Bagging</u>

```
Call:
 randomForest(formula = review ~ ., data = baggin_review, mtry = 9,      importance = TRUE, subset = trai
n)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 8

          Mean of squared residuals: 0.135437
                    % Var explained: 22.36
> |
```

After bagging, the number of trees formed are 500 with 8 splits. The variance explained is 22.36%. We now plot the predicted values with the testing data and get the plot shown below.

> mean((pred-baggin_review[-train,9])^2)

[1] 0.1349083

We get the mean squared error as 0.1349 in the case of bagging, where the number of splits are more.

<u>Random Forests</u>

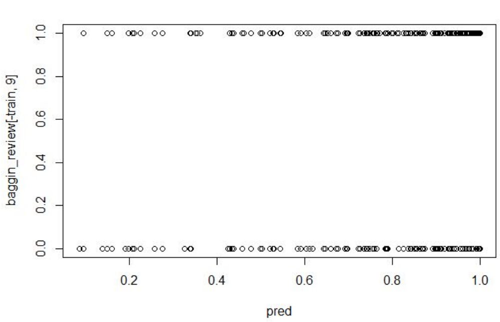
```
Call:
 randomForest(formula = review ~ ., data = baggin_review, mtry = 3,      importance = TRUE, subset = tra
n)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 0.1348612
                    % Var explained: 22.69
> |
```

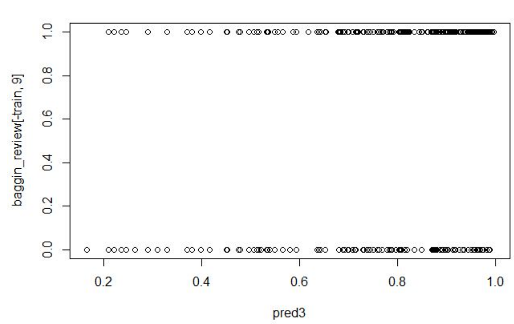
When we decrease the number of splits to 3 we get the variance as 22.69%.

> mean((pred3-baggin_review[-train,9])^2)

[1] 0.1338056

We get the mean squared error as 0.1338 which is lesser than that in bagging (0.1349), but there is not much of a difference.





The plots for both bagging and random forests clearly shows that there is ***no correlation***. This suggests that bagging is not an appropriate method for this particular dataset and research question. A reason why we obtained this is probably because the variable we are trying to predict has binary values of 0 and 1 only.

Question 4. Comparative Analysis

> ***Write a summary of all classifiers, their predictive quality and which one would you use for your system.***

Classification Technique for Research Question 2

Logistic Regression which is a binary classification technique has an accuracy of 81% for predicting whether a listing gets a visit or not.
Decision Trees are fairly accurate in predicting whether a listing gets a visit or not. It shows an error rate of 17% after boosting which is good. Which means an average accuracy of 83% of predicting whether a listing will get a review or not.
However, here we make an assumption that any listing with one or more reviews has got a visit (labelled as Yes) and a listing with 0 reviews has not got any visit (labelled as No).

Classification Technique for Research Question 3

While predicting the price for a listing, Naïve Bayes classifier may not be a very good classification technique as the data set is very large with nearly 26,000 listings. Naive Bayes can work really well with small data sets. Besides, Naive Bayes classifier makes a strong assumption about conditional independence of the features which might not be the case.

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Col Total |
|-------------------------|

Total Observations in Table:  7688

             | actual
   predicted |     cheap | expensive |  moderate | Row Total |
-------------|-----------|-----------|-----------|-----------|
       cheap |      1921 |       416 |      1084 |      3421 |
             |     0.746 |     0.164 |     0.420 |           |
-------------|-----------|-----------|-----------|-----------|
   expensive |       108 |      1641 |       365 |      2114 |
             |     0.042 |     0.649 |     0.141 |           |
-------------|-----------|-----------|-----------|-----------|
    moderate |       546 |       472 |      1135 |      2153 |
             |     0.212 |     0.187 |     0.439 |           |
-------------|-----------|-----------|-----------|-----------|
Column Total |      2575 |      2529 |      2584 |      7688 |
             |     0.335 |     0.329 |     0.336 |           |
-------------|-----------|-----------|-----------|-----------|
```