Chloe Ho

DATA 300

Final Project:

Predicting the price of Airbnb Listings in New York City

A. Data Preprocessing

B. Linear Regression

C. Decision Trees and Tree-based Methods

D. Support Vector Machine

E. Conclusion

A. Data Preprocessing

1. Introduction to the data set

For the final project, I use the "New York City Airbnb Open Data" data set. This dataset features Airbnb listings in New York City, including 48,895 sample units with 15 variables:

id: a unique identifier for each Airbnb listing

name: the title of the Airbnb listing

host_id: a unique identifier for each Airbnb host

host_name*:* the name of the Airbnb host

neighbourhood_group: the borough in which the listing is located

neighbourhood: the specific neighbourhood in which the listing is located

latitude: the latitude coordinate of the listing

longitude: the longitude coordinate of the listing

room_type: the type of room (entire home, private room, or shared room)

price: the price for the listing per night

minimum_nights: the minimum number of nights required for a booking

number_of_reviews: the total number of reviews for the listing

reviews_per_month: the average number of reviews per month for the listing

availability_365: the number of days in a year that the listing is available for booking

calculated_host_listing_count: the number of listings of each Airbnb host

2. Missing values

These variables have missing values: ***name, host_name,*** and ***reviews_per_month***. However, since the two categorical variables ***name*** and ***host_name*** are unnecessary for analysis and modeling, I decided to drop these variables. Meanwhile, to handle missing values of ***reviews_per_month***, I replaced NA with 0, indicating that the listing has no reviews.

3. Variables type

Regarding variable type, ***neighbourhood_group***, and ***room_type*** were originally stored as character types but were converted to factor for analysis. Note that ***neighbourhood*** had 221 categories, which is quite large and may not be suitable for inclusion in the model. Therefore, variable ***neighbourhood*** was removed from the dataset.

B. Regression

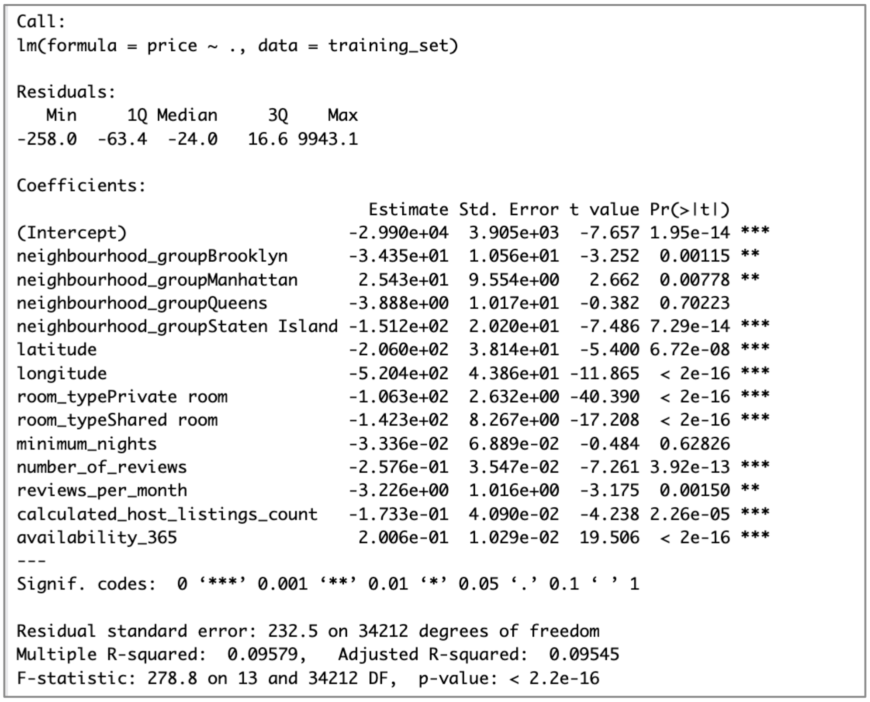
1. Regression goal

My major goal for this data set is to use regression modeling to predict the ***price*** of an Airbnb listing in New York City. Columns ***id***, ***name***, ***host_id***, and ***host_name*** were removed from the data set because they are irrelevant to the analysis. The response variable is the price of the Airbnb listing.

In terms of predictive capacity, some predictors are more likely to be associated with ***price*** than others. Because entire homes are often more expensive than private or shared rooms, the ***room_type*** variable may be a strong predictor of price. Similarly, some boroughs in New York City are more expensive than others, so the ***neighbourhood_group*** variable may be a strong predictor of price.

2. Linear Regression

The dataset was divided into a training set and a testing set, with 80% of the samples allocated for training and 20% for testing. A simple linear regression model was fitted to the training data: the model included all nine predictor variables, including 2 categorical variables (***neighbourhood_group*** and ***room_type***) and the remaining 7 continuous variables, with ***price*** as the response. Here is the summary of the linear regression model with all predictors:

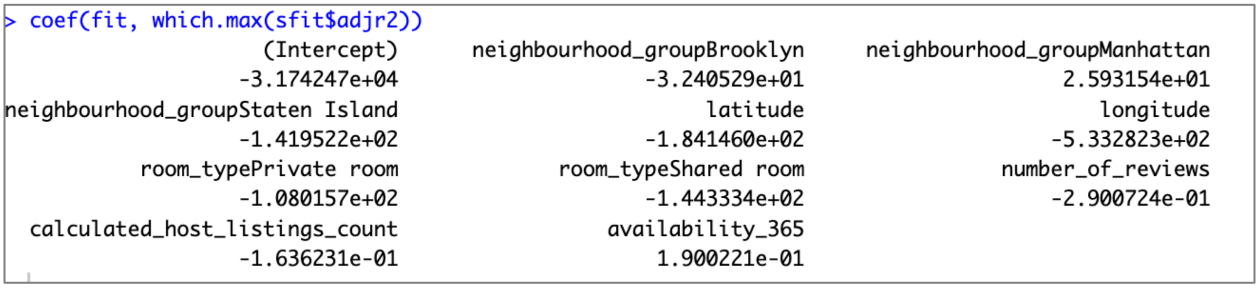


```
Call:
lm(formula = price ~ ., data = training_set)

Residuals:
   Min     1Q Median     3Q    Max 
-258.0  -63.4  -24.0   16.6 9943.1 

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)                    -2.990e+04  3.905e+03  -7.657 1.95e-14 ***
neighbourhood_groupBrooklyn    -3.435e+01  1.056e+01  -3.252  0.00115 ** 
neighbourhood_groupManhattan    2.543e+01  9.554e+00   2.662  0.00778 ** 
neighbourhood_groupQueens      -3.888e+00  1.017e+01  -0.382  0.70223    
neighbourhood_groupStaten Island -1.512e+02  2.020e+01  -7.486 7.29e-14 ***
latitude                       -2.060e+02  3.814e+01  -5.400 6.72e-08 ***
longitude                      -5.204e+02  4.386e+01 -11.865  < 2e-16 ***
room_typePrivate room          -1.063e+02  2.632e+00 -40.390  < 2e-16 ***
room_typeShared room           -1.423e+02  8.267e+00 -17.208  < 2e-16 ***
minimum_nights                 -3.336e-02  6.889e-02  -0.484  0.62826    
number_of_reviews              -2.576e-01  3.547e-02  -7.261 3.92e-13 ***
reviews_per_month              -3.226e+00  1.016e+00  -3.175  0.00150 ** 
calculated_host_listings_count  -1.733e-01  4.090e-02  -4.238 2.26e-05 ***
availability_365                2.006e-01  1.029e-02  19.506  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 232.5 on 34212 degrees of freedom
Multiple R-squared:  0.09579,	Adjusted R-squared:  0.09545 
F-statistic: 278.8 on 13 and 34212 DF,  p-value: < 2.2e-16
```

Figure 1: Summary of All-predictor model

Based on the summary of All-predictor Linear Regression model, the residual standard error (RSE) is 232.5. With the exception of ***minimum_nights***, the majority of variables are statistically significant at a 0.001 threshold. Using this model, I predicted the price for the test data and calculated the test root-mean-square error (RMSE) for the all-predictor linear regression model. The test RMSE was 238.39, indicating a difference of $238.39 between the predicted and actual price in the test set.

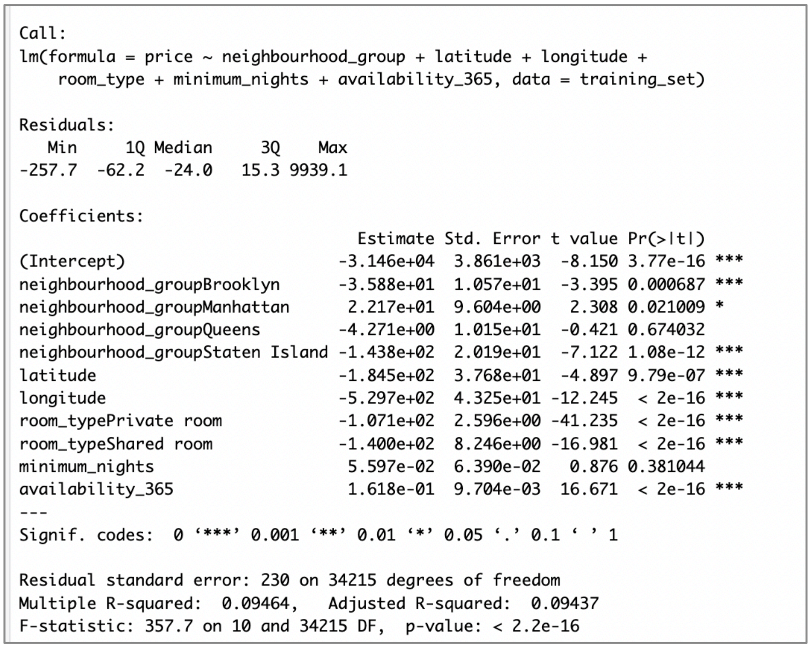
While there is a possibility that all predictors are associated with the response variable, it is more likely that the response is solely associated with a subset of the predictors. To identify the optimal set of predictors, I perform best subset selection using the regsubsets function. The analysis reveals that the maximum number of predictors to be included in the model is 10. Below are the 10 coefficients necessary to construct the best-fitting model:



```
> coef(fit, which.max(sfit$adjr2))
                    (Intercept)      neighbourhood_groupBrooklyn     neighbourhood_groupManhattan 
                  -3.174247e+04                    -3.240529e+01                     2.593154e+01 
neighbourhood_groupStaten Island                        latitude                        longitude 
                  -1.419522e+02                    -1.841460e+02                    -5.332823e+02 
            room_typePrivate room             room_typeShared room                number_of_reviews 
                  -1.080157e+02                    -1.443334e+02                    -2.900724e-01 
  calculated_host_listings_count                 availability_365 
                  -1.636231e-01                     1.900221e-01 
```

Figure 2: 10 optimal coefficients for Best-subset Model

Next, I fit a linear regression using the best subset of predictors, including ***neighbourhood_group***, ***latitude***, ***longitude***, ***room_type***, ***minimum_nights***, and ***availability_365***:



```
Call:
lm(formula = price ~ neighbourhood_group + latitude + longitude + 
    room_type + minimum_nights + availability_365, data = training_set)

Residuals:
   Min     1Q Median     3Q    Max 
-257.7  -62.2  -24.0   15.3 9939.1 

Coefficients:
                                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)                      -3.146e+04  3.861e+03  -8.150 3.77e-16 ***
neighbourhood_groupBrooklyn      -3.588e+01  1.057e+01  -3.395 0.000687 ***
neighbourhood_groupManhattan      2.217e+01  9.604e+00   2.308 0.021009 *  
neighbourhood_groupQueens        -4.271e+00  1.015e+01  -0.421 0.674032    
neighbourhood_groupStaten Island -1.438e+02  2.019e+01  -7.122 1.08e-12 ***
latitude                         -1.845e+02  3.768e+01  -4.897 9.79e-07 ***
longitude                        -5.297e+02  4.325e+01 -12.245  < 2e-16 ***
room_typePrivate room            -1.071e+02  2.596e+00 -41.235  < 2e-16 ***
room_typeShared room             -1.400e+02  8.246e+00 -16.981  < 2e-16 ***
minimum_nights                    5.597e-02  6.390e-02   0.876 0.381044    
availability_365                  1.618e-01  9.704e-03  16.671  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 230 on 34215 degrees of freedom
Multiple R-squared:  0.09464,	Adjusted R-squared:  0.09437 
F-statistic: 357.7 on 10 and 34215 DF,  p-value: < 2.2e-16
```

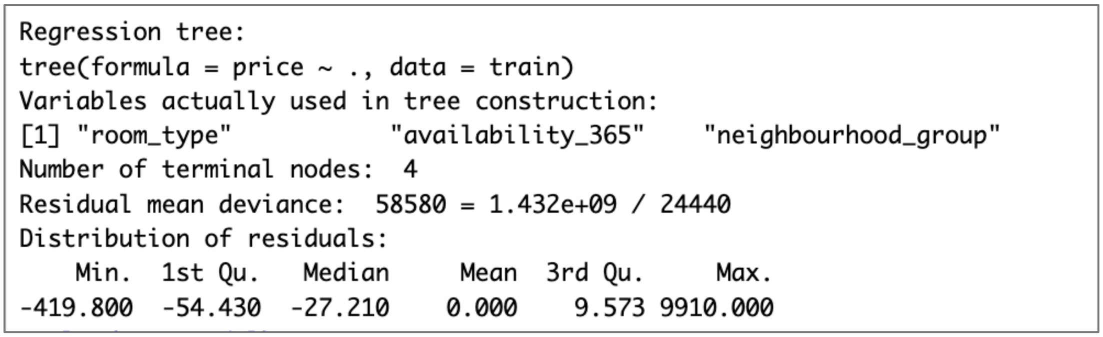
Figure 3: Summary of Best-subset Model

The test RMSE for the Best-subset Model is 238.05, which is quite similar to the test RMSE of All-predictor Model. However, it is worth noting that the Best-subset is marginally better as it has a slightly smaller test RMSE, indicating a better fit to the data.

C. Decision Tree and Tree-based Methods

1. Fitting Regression Trees

Before fitting a Regression Tree to the Airbnb data set, I decide to only include variables that are relevant to modeling. I select ***neighbourhood_group**, **room_type**, **minimum_nights**, **number_of_reviews**, **reviews_per_month**, **calculated_host_listings_count**,* and ***availability_365*** columns because they are likely to be the most informative for predicting the price of an Airbnb listing which produces a more accurate and robust model. First, I create a training set, then fit the tree to the training data. The tree has a total of 4 nodes:



```
Regression tree:
tree(formula = price ~ ., data = train)
Variables actually used in tree construction:
[1] "room_type"           "availability_365"    "neighbourhood_group"
Number of terminal nodes:  4
Residual mean deviance:  58580 = 1.432e+09 / 24440
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-419.800  -54.430  -27.210    0.000    9.573 9910.000
```

Figure 4: Summary of the Full Tree Regression on the training set

The summary indicates that only three of the variables have been used in constructing the tree. This is because the tree-constructing algorithm automatically selects variables that may help achieve a balance between model accuracy and model complexity, reducing the risk of overfitting. I now plot the tree:

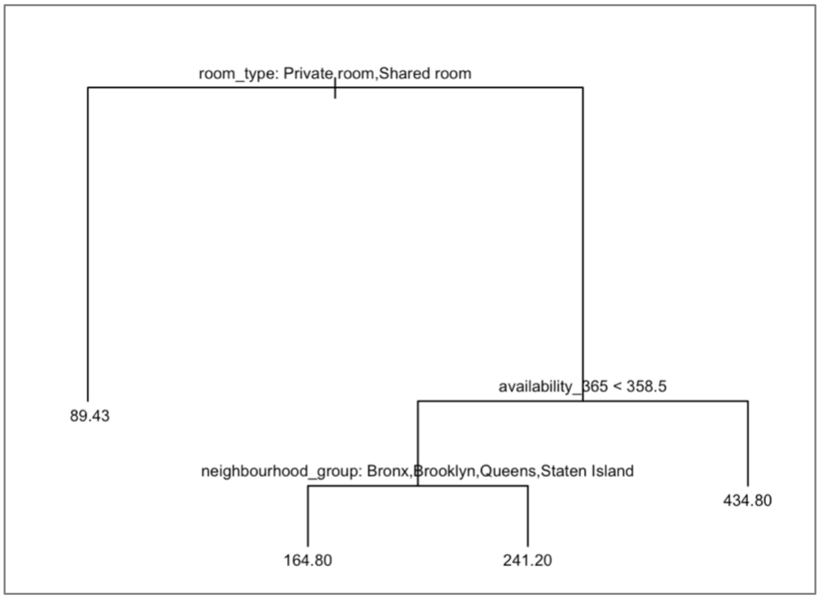

room_type: Private room,Shared room
availability_365 < 358.5
89.43
neighbourhood_group: Bronx,Brooklyn,Queens,Staten Island
434.80
164.80
241.20

Figure 5: Fully-constructed Tree

It can be inferred from the plot that ***room_type*** and ***availability_365*** are the two most important factors in predicting Airbnb listing prices, since they appear at the top of the tree (because they provided the best split of the data). Specifically, the variable ***room_type*** refers to the type of room of the Airbnb listing, including entire home, private room, or shared room, while ***availability_365*** means the number of days in a year that the listing is available for booking. The tree suggests that under-1-year stays in private or shared rooms are associated with lower prices per night compared to entire-home rentals. Furthermore, the model reveals that Airbnb rentals located in Manhattan tend to have a higher nightly price in comparison to rentals in other neighborhoods, such as the Bronx, Brooklyn, Queens, and Staten Island. After that, I predict and evaluate the test RMSE of the unpruned tree in comparison to the Baseline Test RMSE, standing at 237.0901 and 242.2283 respectively.

2. **Cross-validation pruning**

In order to test if pruning the tree will improve performance, I plot a graph to select the optimal level of complexity with cross-validation. The selected model with the lowest cross-validation error appears to be the model with 4 terminal nodes. Since the optimal tree is the Fully-grown Tree without any pruning, the test RMSE is the same as the unpruned tree.

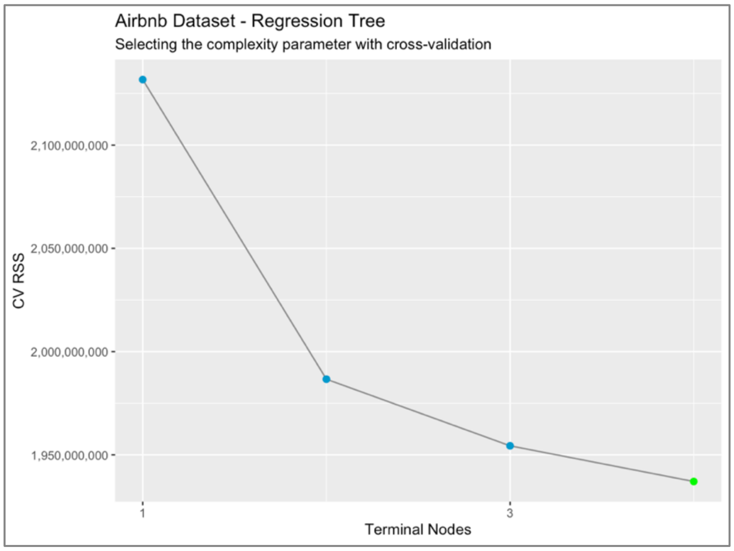
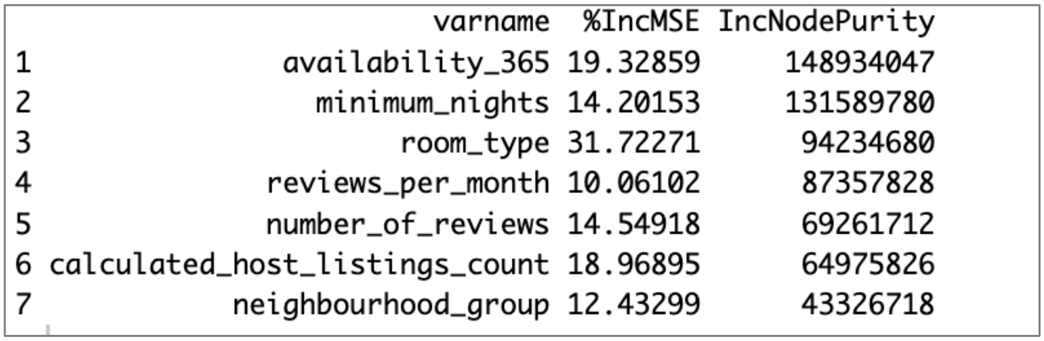


Figure 6: Plot of optimal complexity parameter with cross validation

3. Bagging and Random Forests

Here I apply Bagging to the Airbnb data with 3 variables tried at each split (mtry = 3), obtaining a Bagged Regression Tree with a test RMSE of 212.4323. Compared to the fully-grown tree, this represents a decrease of about 12% in test RMSE, suggesting an improvement in predictive power. The variable importance statistics are provided below to aid in understanding the contribution of each variable to the model.



```
                         varname  %IncMSE IncNodePurity
1                availability_365 19.32859     148934047
2                  minimum_nights 14.20153     131589780
3                       room_type 31.72271      94234680
4               reviews_per_month 10.06102      87357828
5               number_of_reviews 14.54918      69261712
6 calculated_host_listings_count 18.96895      64975826
7             neighbourhood_group 12.43299      43326718
```

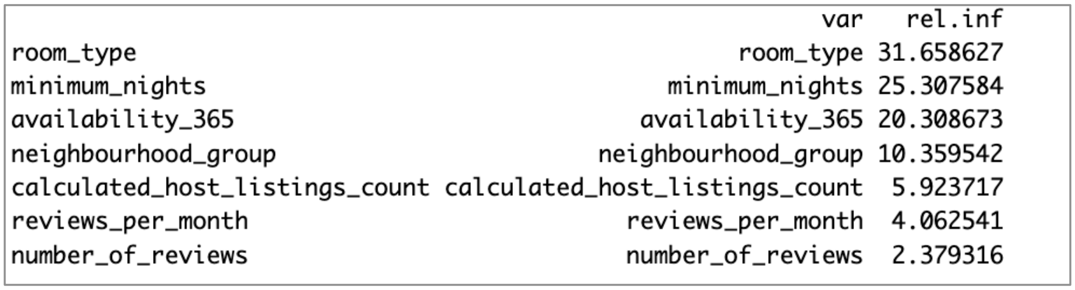
Figure 7: Variable importance

Two measures of variable importance are reported, namely IncMSE (Increase in Mean Squared Error) and IncNodePurity (Increase in Node Purity). While IncMSE evaluates how important a predictor variable is for a model's accuracy, IncNodePurity measures how well a predictor separates the response variable into more homogeneous groups in a given node. Based on the two measures, ***availability_365***, ***minimum_nights,*** and ***room_type*** are the three most important variables. This result is consistent with the top splits of the Full Regression Tree.

Growing a Random Forest model proceeds in exactly the same way, except that we use smaller number of variables tried at each split (mtry = 2). The RMSE for Random Forest is 207.0531, which is by far the optimal model with the lowest RMSE.

4. Boosting

Another tree-based method that I am using for predicting an Airbnb listing price is Boosting. First, I use cross-validation to select the shrinkage parameter for Boosting, which is 0.001, and decide that model will have 5,000 trees. Below are the relative influence statistics and a relative influence plot of the model:



```
                                                          var   rel.inf
room_type                                           room_type 31.658627
minimum_nights                                 minimum_nights 25.307584
availability_365                             availability_365 20.308673
neighbourhood_group                       neighbourhood_group 10.359542
calculated_host_listings_count calculated_host_listings_count  5.923717
reviews_per_month                           reviews_per_month  4.062541
number_of_reviews                           number_of_reviews  2.379316
```

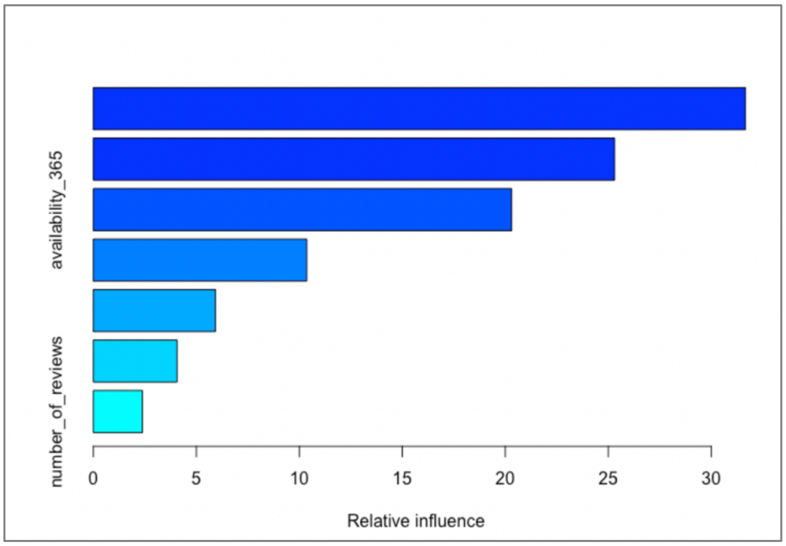


Figure 8: Statistics Summary and Relative Influence Plot

From the summary and the plot, ***room_type*** and ***minimum_nights*** are by far the most important variables with the relative influence of 31.65 and 25.3 respectively. In this case, as we might expect, median Airbnb prices are higher for entire-home/apartment rentals in comparison to private/shared room rentals. Moreover, prices are decreasing when minimum nights get longer.

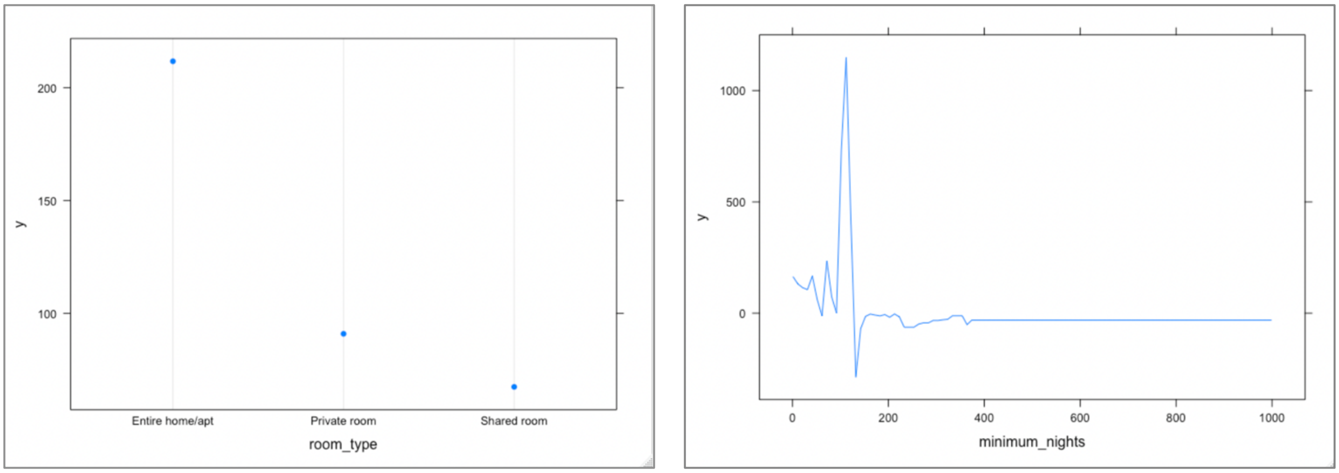


Figure 9: Plots of room_type and minimum_nights in Boosting model

I then use the Boosting model to predict price on the test set. The test RMSE using boosting is 210.4895, suggesting that the average difference between the predicted and actual prices on the test set is approximately 210 dollars. Here is the table displaying the test RMSE of decision trees and other tree-based methods used to predict the price of an Airbnb listing in New York:

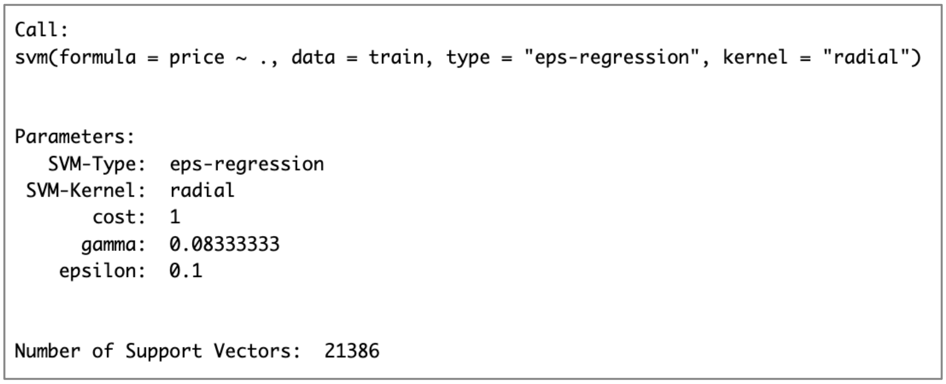
Method	Decision Trees	Bagging	Random Forests	Boosting
Test RMSE	237.09	212.43	207.05	210.48

Figure 10: Table of Test MRE using Decision Trees and other Tree-based methods

The test RMSE for Boosting is 210.48, which is superior to the test RMSE of Random Forests and Bagging.

D. Support Vector Machine

First, I define and fit the Support Machine Vector (SVM) model with default parameters on train data, and obtain some basic information about the SVM fit using summary():



```
Call:
svm(formula = price ~ ., data = train, type = "eps-regression", kernel = "radial")


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.08333333
    epsilon:  0.1


Number of Support Vectors:  21386
```

Figure 11: Summary of SVM Model

This tells us that a radial kernel was used with cost = 1, and that there were 21,386 support vectors. Next, I will predict the test data and plot the results to compare visually.

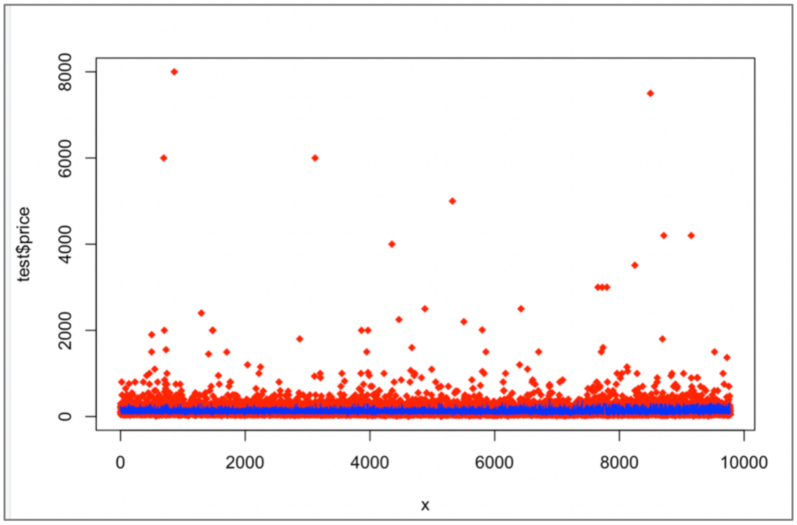
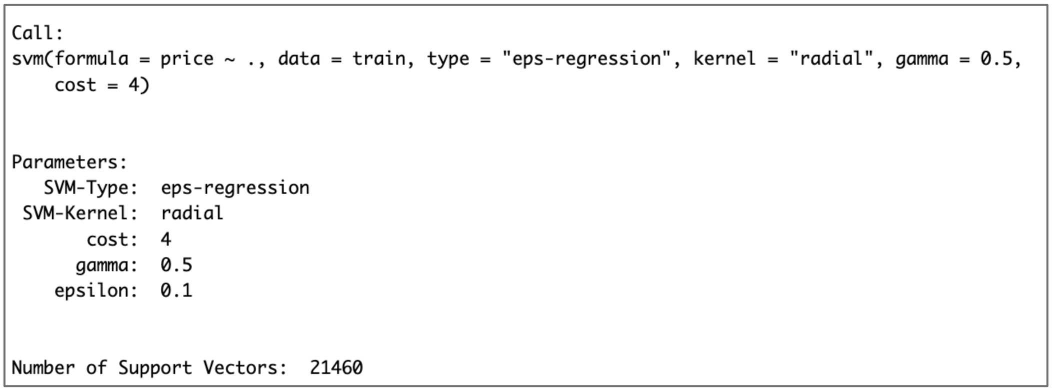


Figure 12: SVM Model with default parameters

The red dots represent the actual values, while the blue dots correspond to the predicted values. Upon initial analysis, the model appears to perform well on the data since the majority of the observation points are closely aligned with the actual observations, aside from a few outliers. The test RMSE for this SVM model, using default parameters, is 219.431.

By default, the SVM function in R assumes a maximum allowed error (epsilon) of 0.1. To prevent overfitting, I utilize a penalty function by training models with varying allowable error and cost parameters. To optimize the hyperparameters of the SVM model, I randomly sample 20% of the original dataset and perform cross-validation. After tuning, the best model is found to have cost, gamma, and epsilon values of 4, 0.5, and 0.1, respectively. Applying the tuned SVM to the training set yielded the following results:



```
Call:
svm(formula = price ~ ., data = train, type = "eps-regression", kernel = "radial", gamma = 0.5,
    cost = 4)


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  4
      gamma:  0.5
    epsilon:  0.1


Number of Support Vectors:  21460
```

Figure 13: Summary of tuned SVM Model

Next, I utilize the tuned SVM model to predict prices on the test set, achieving a RMSE of 204.5. This indicates that the SVM model outperforms all other models by a significant margin.

E. Conclusion

In summary, I employ seven different models to predict the price of Airbnb listings in New York City. These models include Linear Regression with all predictors, Linear Regression with best subset variable selection, Decision Trees, Bagging, Random Forests, Boosting, and Support Vector Machine. The following table provides a summary of each model's test performance:

Model	LR best-subset	Decision Trees	Bagging	Random Forests	Boosting	SVM
Test RMSE	238.05	237.09	212.43	207.05	210.48	204.49

Figure 14: Model Accuracy Comparison

The table provides a comprehensive overview of the performance of each model in predicting the price of an Airbnb listing in New York City. It shows that Support Vector Machine (SVM) outperforms all other models in terms of test RMSE. This result suggests that SVM is a reliable model for predicting the price of an Airbnb listing in the given dataset.

While SVM is the best overall performer, it is notable that the other models also demonstrate reasonable performance, with test RMSEs ranging from 207 to 239. It's important to consider the strengths and limitations of each model, as well as the specific needs and goals of the analysis. For example, linear regression with best subset variable selection is a simpler and more interpretable model, while ensemble models like random forests or boosting can capture complex interactions and nonlinear relationships between predictors. The choice of the best model ultimately depends on various factors, highlighting the need to carefully consider model complexity and interpretability when selecting a model for a specific analysis.