Linear Regression: Sergio Garcia, Devon O'Brien,

Decision Trees: Justin Cao, Robert Hines, Austin Strain

December 6th, 2019

Dr. Wang

AirBnB Final Report

Introduction (Question 1)

Dataset Description (Part A)

We're analyzing the AirBnB listings in New York dataset. AirBnB is a website/app comparable to a hotel. It allows users to rent out rooms in their house/apartment or their entire house/apartment, and allows renters to find rooms/houses. This dataset provides a variety of identifiers about each listing, such as location and price. We removed about half of the listings (private rooms and shared rooms) to only look at entire homes.

Question Formulation (Part B)

We wanted to see if we could make a reliable *inference* on the price to rent an entire private home in New York City through AirBnB using the following predictors per listing: borough, number of reviews, minimum required nights per booking, number of reviews per month, and the annual number of days available for booking.

Analysis (Question 2)

Primary Question (Part A)

QUESTION: Which of the following variables affect the price of an AirBnB listing in New York:

- Borough
- Number of reviews
- Minimum required nights per booking
- Number of reviews per month
- Annual number of days available for booking.

We'll be using linear regression and decision trees to answer this question. Expected advantages of linear regression include:

1. Simplicity
2. Interpretability
3. Scientific acceptance
4. Widespread availability

Expected advantages of decision trees include:

1. Forcing the consideration of all possible outcomes of a decision
2. Trace each decision tree path to a conclusion
3. Creates a comprehensive analysis of the consequences along each branch
4. Identifies decision nodes that require further analysis

Linear Regression Model (Part B)

1. $\hat{y} = b_0 + b_1x_1 + b_2x_2 + \ldots + b_px_p$

   $\hat{y}$ = 91.61172 + (66.76417)(neighbourhood_group2) + (121.70396)(neighbourhood_group3) + (-0.32010)(number_of_reviews) + (-0.57637)(minimum_nights) + (0.27772)(availability_365)

2. A number of predictors were initially excluded because we wouldn't expect them to be of any use (listing ID, name of host, host ID, host name, latitude, longitude)
3. ```
   n <- nrow(AirBnBNYC)
   p <- ncol(AirBnBNYC)
   set.seed(1)  #seed is set to reflect iteration. Next will be 2, then 3, etc.
   train <- sample(n, 0.8*n)
   AirBnBNYC.train = AirBnBNYC[train, 1:p]
   AirBnBNYC.train.labels <- AirBnBNYC[train, p+1]
   AirBnBNYC.test = AirBnBNYC[-train, 1:p]
   AirBnBNYC.test.labels <- AirBnBNYC[-train, p+1]
   ```

4. ```
   lm.fit = lm(price ~ neighbourhood_group + number_of_reviews +
   minimum_nights + availability_365, data = AirBnBNYC.train)
   lm.probs = predict(lm.fit, AirBnBNYC.test, type="response")
   lm.probs
   predict_err <- AirBnBNYC.test$price-lm.probs
   predict_mse <-
   sum((AirBnBNYC.test$price-lm.probs)^2)/nrow(AirBnBNYC.test)
   Predict_mse
   ```

5. Test MSEs:

   ```
   [1] 122949.50 124179.80  70838.61  72164.95  62463.38
   [6]  43514.34  62042.81  66193.60 101390.60  82636.21
   ```

Mean of Test MSEs:

```
[1] 80837.38
```

Decision Tree Model (Part B)

1. price ~ neighbourhood_group + number_of_reviews + minimum_nights + reviews_per_month + availability_365
2. We used the same predictors as the linear regression for consistency sake. We wanted to test the decision tree's performance vs the linear regression's performance. To make the performance as good as we could make it, we tried pruning, random forests, and bagging as different approaches for validation.
3. ```
   library(readxl)
   AirBnBNYC <- read_excel("AirBnBNYC.xlsx")
   data = AirBnBNYC
   data$neighbourhood = factor(data$neighbourhood)
   data$neighbourhood_group = factor(data$neighbourhood_group)
   data$reviews_per_month[is.na(data$reviews_per_month)] = 0

   set.seed(1)
   pd = sample(2,nrow(data),replace = TRUE, prob = c(0.8,0.2))
   train = data[pd==1,]
   test = data[pd==2,]
   ```

4. ```
   tree = tree(price ~ neighbourhood_group + number_of_reviews +
   minimum_nights + reviews_per_month + availability_365,
   data=train)
   tree.test_mse = mean((predict(tree, test) - test$price)^2)
   ```

5. ```
   test_mses = integer(10)
   for (i in 1:10) {
     set.seed(i)
     pd = sample(2,nrow(data),replace = TRUE, prob = c(0.8,0.2))
     train = data[pd==1,]
     test = data[pd==2,]
     tree = tree(price ~ neighbourhood_group + number_of_reviews +
       minimum_nights + reviews_per_month + availability_365,
       data=train)
     test_mses[i] = mean((predict(tree, test) - test$price)^2)
   }
   test_mses
   mean(test_mses)
   ```

   Result of Building 10 Decision Trees:
   ```

```
> test_mses
 [1] 88657.69 75935.70 58734.86 72323.59 54365.49
 [6] 92396.14 95583.50 72985.85 85537.73 86895.71
> mean(test_mses)
[1] 78341.62
```

Inference Interpretation (Part D)

Linear Regression (Part 1, 2)

```
Call:
lm(formula = price ~ neighbourhood_group + number_of_reviews +
    minimum_nights + reviews_per_month + availability_365, data = AirBnBNYC)

Residuals:
   Min     1Q Median     3Q    Max
-233.0  -72.9  -31.5   24.6 9846.3

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                    92.14642   12.72389   7.242 4.58e-13 ***
neighbourhood_groupBrooklyn    66.22946   12.70408   5.213 1.87e-07 ***
neighbourhood_groupManhattan  121.16925   12.66402   9.568  < 2e-16 ***
neighbourhood_groupQueens      24.51722   13.47678   1.819   0.0689 .
neighbourhood_groupStaten Island -0.53471   21.71873  -0.025   0.9804
number_of_reviews              -0.32010    0.04027  -7.949 1.98e-15 ***
minimum_nights                 -0.57637    0.07875  -7.319 2.60e-13 ***
reviews_per_month              -1.26368    1.19859  -1.054   0.2918
availability_365                0.27772    0.01276  21.756  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 218.2 on 20323 degrees of freedom
  (5077 observations deleted due to missingness)
Multiple R-squared:  0.04833,   Adjusted R-squared:  0.04796
F-statistic:   129 on 8 and 20323 DF,  p-value: < 2.2e-16
```

As the overall p-value is essentially 0, we can conclude that there is a significant relationship between this set of predictors and the response (price).

We interpret b_i as the average effect of Y (the predictor) of a one unit increase in X_j, holding all other predictors fixed.

$b_0 \approx 92.15$, our intercept, is what we expect if all predictors = 0

$b_1 \approx 66.23$, if located in Brooklyn, we expect an increase of 66.23 in price

$b_2 \approx 121.17$, if located in Manhattan, we expect an increase of 121.17 in price

$b_3 \approx -0.32$, for each review, we expect the price to decrease by 0.32

$b_4 \approx -0.57$, for each minimum night greater than 1, we expect the price to decrease by 0.57

$b_5 \approx 0.28$, for each day the unit is available annually, we expect the price to increase by 0.28

Because of these predictors in our model, the remaining are insignificant:

- Neighbourhood_group (Bronx, Queens, Staten Island)
- Reviews_per_month

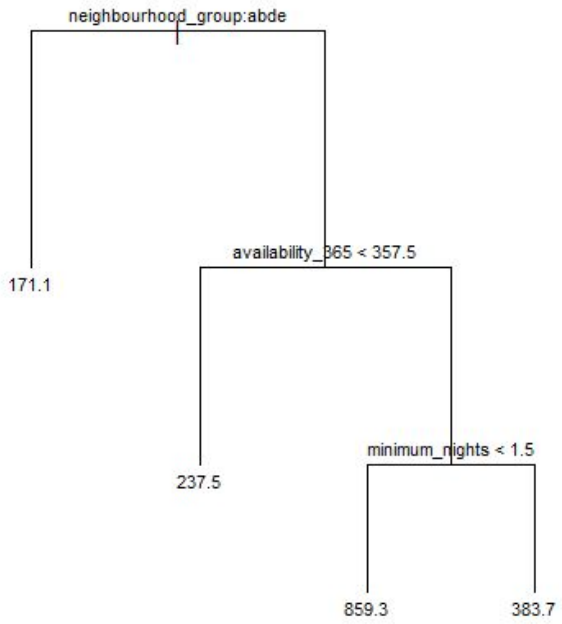
Decision Tree (Part 1, 2)

```
> summary(tree)

Regression tree:
tree(formula = price ~ neighbourhood_group + number_of_reviews +
    minimum_nights + reviews_per_month + availability_365, data =
train)
Variables actually used in tree construction:
[1] "neighbourhood_group" "availability_365"
[3] "minimum_nights"
Number of terminal nodes:  4
Residual mean deviance:  74880 = 1.517e+09 / 20260
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-789.30  -86.09  -41.09    0.00   14.49 9829.00

> summary(pruned)

Regression tree:
tree(formula = price ~ neighbourhood_group + number_of_reviews +
    minimum_nights + reviews_per_month + availability_365, data =
train)
Variables actually used in tree construction:
[1] "neighbourhood_group" "availability_365"    "minimum_nights"
Number of terminal nodes:  4
Residual mean deviance:  74880 = 1.517e+09 / 20260
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-789.30  -86.09  -41.09    0.00   14.49 9829.00
```



Our decision tree made splits according to the following three predictors:

- neighbourhood_group
- availability_365
- minimum_nights

These predictors appear to be significant enough to warrant dividing the training data into different divisions. Therefore, these predictors have some form of importance for influencing the price of a listing.

While pruning the tree did significantly improve the test MSE, pruning the tree did not consistently change this particular pattern of splits.

Concluding Remarks (Part E)

For inference performance the decision tree ended up performing better, although not by much. We guessed initially that the linear model would perform badly and the decision tree would

perform considerably better, since we knew the models were probably not linearly related. Our guess in terms of the linear model was correct, as almost none of the predictors were linear.

As for the decision tree, we believe that the greedy nature of the decision trees did a good job at looking at the important factors, but did poorly in producing a globally optimal solution for the data. We also believe that overfitting was a huge issue for our tree. The test MSE was 14000 over the train MSE, showing obvious overfitting.

Extra Remarks on Validation Approaches (Part E)

After performing the primary investigation with linear regression and decision trees, we tried to combat the extreme overfitting of the models by using a set of validation approaches. With decision trees, we attempted to use pruning, random forest, and bagging. For linear models, we computed the mean of multiple linear models to find a central model. Versions of these models can be found in the provided code (warning: random forest and bagging both take a significant amount of time to compute).

Pruning did appear to significantly affect the test MSE, almost by 40000 in certain cases. This is shown in an early response.

Both bagging and random forest did show improved training and test MSEs. Thus, it may have been a better idea to use either of these two models instead of a decision tree. Even with pruning the decision tree, bagging and random forest approaches showed more promising results since IncNodePurity and %IncMSE can be used to better justify which predictors influence the price of a listing.

Below are some of the results from each of the models:

```
> importance(bagging)
                     %IncMSE IncNodePurity
neighbourhood_group 15.35023      61147399
number_of_reviews   22.43062     157449941
minimum_nights      20.92564     186658767
reviews_per_month   18.52248     225253114
availability_365    33.87704     380596129

> importance(rf)
                     %IncMSE IncNodePurity
neighbourhood_group 14.60808      32633492
number_of_reviews   12.32326      39350215
minimum_nights      11.72192      35134303
reviews_per_month   11.42947      56242606
availability_365    19.09031      75149876
```

```
> pruned.test_mse
[1] 46378.21
> tree.test_mse
[1] 88657.69
```