2024

Data, Modeling, and Analysis

Table of Contents

Executive Summary:

The project revolves around predicting optimal neighborhoods in the five boroughs of New York City (NYC) for Airbnb listings, utilizing a dataset encompassing Airbnb's 2019 listing activity in NYC. The objective is to explore the impact of various neighborhood variables on Airbnb prices, offering valuable insights for optimizing Airbnb locations. However, the initial dataset proved insufficient for effective predictive modeling, leading to comprehensive data cleaning and transformation. Hence, the dataset went under data cleaning and transformation. External variables, including crime rate, education level, household median value and proximity from Airbnb location to key sites etc. were collected to enrich the dataset. Exploratory data analysis and outlier handling were performed using descriptive statistics and histograms, resulting in either the removal or replacement of outliers. A linear regression model was built using R, iterating through multiple rounds of data cleaning and model refinement to achieve optimal performance. Additionally, stepwise regression was employed to address insignificant variables, but there was no major change observed in the results, The project concludes with recommendations and policy suggestions for dataset enhancement to further improve predictive modeling accuracy and usability.

Problem Statement:

This dataset aims to predict the good neighborhoods in the 5 boroughs of New York City (NYC) based on various factors such as the property value, air quality index, income, crime rate, land size

and unemployment rate. This prediction helps us in drawing actionable insights as it aims to find how a good neighborhood affects Airbnb prices.

Researchers, analysts, and businesses may seek to address issues such as optimizing Airbnb locations, understanding market trends, or scouting good neighborhoods. Therefore, the central problem is how to leverage the variables in this dataset, including neighborhood data, room types, pricing, review metrics, and more, to derive meaningful conclusions and facilitate data-driven solutions within the realm of Airbnb's operations in NYC during 2019.

Dataset Description:

This dataset offers a comprehensive overview of Airbnb's listing activity and associated metrics in NYC for the year 2019. Airbnb is a platform facilitating unique and personalized travel experiences since 2008 and has redefined the way guests and hosts connect.

The original dataset is stored in a CSV file format and contains vital information on hosts, geographical availability, predictive metrics, and the means to derive valuable conclusions. It consists of 16 columns and 48,895 rows originally.

Table 1: Data Dictionary

Column Name	Description	Type of Variable
ID	Unique identifier for each listing	Identifier
Name	Descriptive name or title of the property	Categorical
Host ID	IDs of the property owners	Categorical
Host Name	Name of host	Categorical
Borough	Where the property is located	Categorical
Neighborhood	The specific neighborhood within the borough.	Categorical
Income	Median Income of the specific neighborhood	Numeric
Crime rate	Crime rate of the specific neighborhood	Numeric
Population (by race)	Percentage distribution of population by race	Numeric
Property Price ($)	Property Price in each neighborhood	Numeric
Education Level	Percentage of the population with high school educational level	Numeric
Land Size	Total size of a specific neighborhood in sq ft	Numeric
Unemployment Rate	Unemployment rate in a specific neighborhood	Numeric
Latitude/Longitude	Geographic lat/long for each property	Numeric
Room Type	Type of room available for rent	Categorical
Price	Price of Air BnB in $	Numeric

Assumptions:

The following assumptions provide a foundation for applying linear regression to the dataset. Ensuring they are met enhances the reliability and interpretability of the model, allowing for meaningful predictions regarding prices based on selected features and neighborhood attributes.

- **Linearity:** The relationship between the dependent variable (price) and independent variables is linear.
- **No Multicollinearity**: Each independent variable in the dataset is not strongly correlated with another independent variable.

Detailed Methodology:

1. **Data Cleaning and Transformation:**

Data Deletion

- Irrelevant columns such as last review and reviews per month were deleted to simplify the data set.

Data Transformation

- We simplified our data by narrowing down the neighborhoods in terms of their frequency count. We chose to focus only on neighborhoods that had a count of 500 or above which resulted in 24 neighborhoods.

Data Collection

- We gathered information from external sites, for example trip advisor, on hospitals, malls, subways, parks and tourist destinations in each borough, as well as found latitude and longitude for each site to facilitate geospatial analysis and establish relationships with the neighborhood properties. Our assumption for taking this step was that this proximity would define the value of property of Air Bnb.
- Moreover, we gathered other variables such as crime rate, population, median household value, education level and land size for the 24 shortlisted neighborhoods to create more meaningful features for analysis. We took the average value of NYC for the neighborhoods with a frequency of more than 500. These variables helped us in defining a good neighborhood which can create an impact on the property price of Air Bnb.

2. Exploratory Data Analysis and Outlier Handling:

- Utilized descriptive statistics and histograms to identify outliers.
- The average of the column was used to replace outliers in each column.
- Employed histograms to visualize the distribution of property prices of Air Bnb. Figure 1 shows a skewed histogram; hence we deployed a fixed top and bottom percentile to remove the outliers. Calculated the 90th and 10th percentiles to define the outlier range. Hence, we removed the outliers, specifically 9938 rows for improved data integrity and model performance. Figure 2 shows the histogram after the removal of outliers.

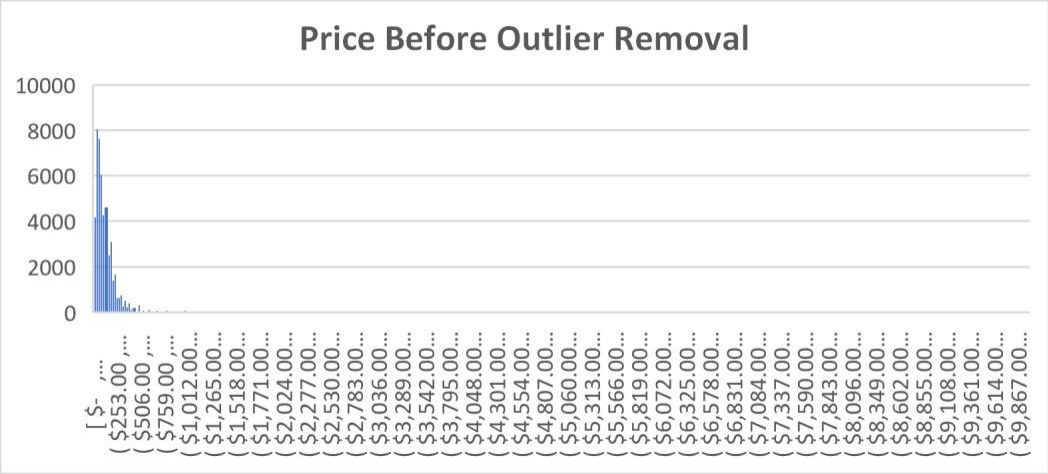

Figure 1: y-variable before data cleaning

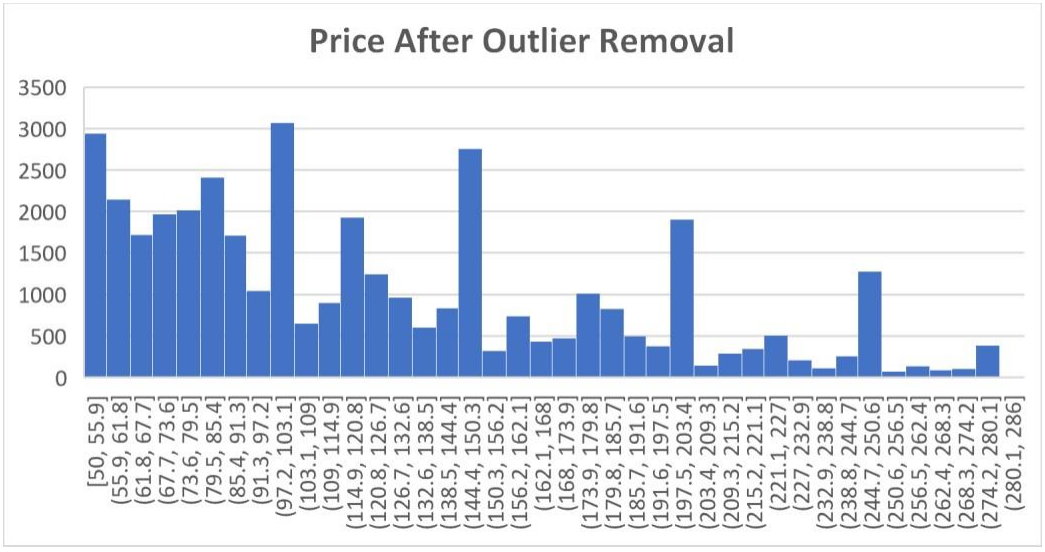

Figure 2: y-variable after data cleaning

3. Feature Engineering:

External Variables Integration

- We used SQL and R to incorporate the distance between each Air Bnb in a neighborhood to the destinations. The destinations include subway stations within 1km, hospitals, parks, malls, and tourist destinations all within 5km. This helped us in identifying the relationship between the price of property and the proximity of each site. These new variables were incorporated in the dataset by the Vlookup function.
- Next, we incorporated the above listed new external variables (crime rate, population etc.) in the data set for analysis. We did this by applying the Vlookup formula.

New Variables

- We created new variables named (Entire home/ Private Room and Shared room) in an attempt to convert the categorical variable to binary so that it easier to run the linear regression and establish the correlation with the final property price.

Model Building

- There were 16 independent variables and 1 dependent variable. Since the dependent (y-variable) was continuous, we decided to build a linear regression model on it.
- The dataset was huge, so we used R for building the model. After data cleaning and feature engineering as mentioned above, we build our initial model on R. It was followed by more data cleaning and rebuilding the model until we achieved an optimum value for r-square and MAPE.
- After linear regression, we also ran a stepwise regression to cater for insignificant variables. However, there was not much difference observed in the results.

Model Results

- Final r-square = 13.17%
- Final MAPE = 42.55%

```
> summary(model)

Call:
lm(formula = Price ~ ., data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-137.00  -41.99  -11.93   33.30  189.92

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     2.480e+01  2.796e+00   8.869  < 2e-16 ***
Income                         -6.961e-05  2.016e-05  -3.454 0.000554 ***
Crime.Rate..per.1000.residents.  3.107e+00  1.006e-01  30.891  < 2e-16 ***
Population..by.race.             9.512e-01  2.717e-02  35.006  < 2e-16 ***
Property.Price                   2.465e-06  1.592e-07  15.481  < 2e-16 ***
Educational.Level                1.609e-01  1.664e-02   9.666  < 2e-16 ***
Land.Size..sq.ft.               -2.310e-07  2.149e-08 -10.749  < 2e-16 ***
Unemployment.Rate                1.109e+00  1.852e-01   5.986 2.17e-09 ***
Minimum_Nights                   8.408e-03  1.360e-02   0.618 0.536332
Number_of_Reviews               -3.563e-02  6.171e-03  -5.774 7.82e-09 ***
Calculated_Host_Listings_Count   2.248e-01  9.736e-03  23.092  < 2e-16 ***
Availability_365                 1.904e-02  2.248e-03   8.468  < 2e-16 ***
Hospital_Count_5km               3.178e-01  3.525e-01   0.902 0.367226
Subway_Count_1km                -1.400e+00  4.805e-01  -2.914 0.003574 **
Park_Count_5km                  -9.961e-03  7.108e-02  -0.140 0.888545
Mall_Count_5km                  -1.912e-01  5.854e-02  -3.267 0.001089 **
Tourist_Dest_Count_5km           5.815e-03  5.516e-01   0.011 0.991588
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 53.81 on 39441 degrees of freedom
Multiple R-squared:  0.1317,    Adjusted R-squared:  0.1313
F-statistic: 373.9 on 16 and 39441 DF,  p-value: < 2.2e-16
```

```
> predicted_values <- predict(model, data)
> actual_values <- data$Price
> mape <- mean(abs((actual_values - predicted_values) / actual_values)) * 100
> print(mape)
[1] 42.551
```

Policy Suggestions

The regression results derived from the Airbnb data can offer valuable insights for policy suggestions:

1. **Minimum Nights (Positive Coefficient):** Encourage longer minimum stay durations by providing incentives or discounts for guests booking stays. This might attract guests seeking more extended stays, potentially increasing booking stability and reducing turnover.

2. **Number of Reviews (Negative Coefficient):** Focus on improving the guest experience to minimize negative reviews. Implement measures such as prompt issue resolution, improved amenities, and better communication to enhance guest satisfaction and subsequently reduce the likelihood of negative reviews.

3. **Host Listings Count (Positive Coefficient):** Encourage hosts to manage multiple listings effectively by providing resources or tools for efficient management. It might help in boosting the supply of available properties and providing guests with more options, potentially improving overall booking rates.

4. **Availability 365 (Positive Coefficient):** Explore strategies to maintain high availability throughout the year. This could involve incentivizing hosts to keep their properties available, ensuring a steady supply of accommodation for potential guests.

5. **Proximity to Facilities (Mixed Coefficients):** Invest in areas where there are ample amenities nearby (hospitals, parks, malls, etc.), as these aspects positively influence bookings. However, consider other factors like outlier coefficients, ensuring that such facilities aren't overly dominant or outliers skewing the general trend.

6. **Price Influence (Negative Coefficients):** Review pricing strategies and consider adjustments if needed, particularly for instances where pricing negatively affects bookings. This might involve dynamic pricing models or promotions during off-peak periods to increase bookings without sacrificing revenue.

7. **Subway Access (Zero Coefficient):** While subway access might not significantly impact bookings according to the model, ensure that transportation facilities are highlighted to potential guests for convenience and ease of access.

Scope for Further Work

1. **Coefficient Analysis:** Look at the coefficients of the variables to understand their impact on the dependent variable. Focus on those with higher magnitude, like 18.8, 30.9, 839000, and 83.3, as they seem to have more substantial effects.

2. **Statistical Significance**: Pay attention to the p-values associated with each coefficient. Coefficients with p-values less than 0.05 are typically considered statistically significant. Variables with low p-values are more likely to have a significant impact on the output.

3. **Outliers Detection:** Investigate further if there are any outliers in *minimum nights, no. of reviews, and calculated hosts listing count*, affecting the model's performance. These can distort the coefficients and affect the model's accuracy.

4. **Model Fit:** Consider other model evaluation metrics beyond R-squared, like RMSE (Root Mean Squared Error), to understand the average error of the model predictions.

5. **Feature Engineering:** If feasible, explore new variables like air quality index, green space etc, of NYC and merge with the data to see the impact on the model results.

6. **Regularization:** Consider using regularization techniques like Lasso or Ridge regression to handle multicollinearity and prevent overfitting if there are many features in the dataset.

Conclusion

In pursuit of refining our Airbnb price prediction model applied to the New York dataset, the XGBoost model can be preferred. The current model's performance, as denoted by an r-square value of 13% and a MAPE of 42%, signifies substantial room for enhancement. XGBoost, distinguished for its efficacy in handling complex non-linear relationships within data, offers robust capabilities. Its adeptness in managing missing data, facilitating feature importance assessments, and employing regularization techniques positions it as an attractive alternative for optimizing our predictive model. By harnessing XGBoost's capabilities and diligently fine-tuning its hyperparameters, a significant amelioration in the model's precision in forecasting Airbnb prices within the New York market is anticipated.

Appendices

Appendix A: R Console screenshots for Outlier Removal, Linear Regression Model and K-Folds Cross Validation

```r
data<- read.csv("FINAL.csv")
str(data)
data$Price <- gsub("[$,]", "", data$Price) # Removing $ and ,
data$Price <- as.numeric(data$Price) # Convert to numeric
# Remove $ and % signs and convert to numeric
data$Income <- as.numeric(gsub("[$,]", "", as.character(data$Income)))
data$Crime.Rate..per.1000.residents. <- as.numeric(gsub("%", "", as.character(data$Crime.Rate..per.1000.residents.)))
data$Population..by.race. <- as.numeric(gsub("%", "", as.character(data$Population..by.race.)))
data$Property.Price <- as.numeric(gsub("[$,]", "", as.character(data$Property.Price)))
data$Educational.Level <- as.numeric(gsub("%", "", as.character(data$Educational.Level)))
data$Unemployment.Rate <- as.numeric(gsub("%", "", as.character(data$Unemployment.Rate)))
data$Property.Price <- as.numeric(gsub("[$,]", "", as.character(data$Property.Price)))
data <- data[!is.na(data$Host_ID), ]
data <- data[, !names(data) %in% "Host_ID"]
data <- data[, -c((ncol(data) - 9):ncol(data))]

# Assuming your data frame is named 'data'
income_column <- data$Income

# Calculate the 10th and 90th percentile
p10 <- quantile(income_column, 0.1)
p90 <- quantile(income_column, 0.90)

# Calculate the average of the "Income" column
average_income <- mean(income_column, na.rm = TRUE)

# Replace values below the 10th percentile and above the 90th percentile with the average income
income_column[income_column < p10] <- average_income
income_column[income_column > p90] <- average_income

# Update the column in the original data frame
data$Income <- income_column
# Assuming 'data' is your dataframe
income_data <- data$Income

# Create a histogram for the 'Income' column
hist(income_data, breaks = 20, col = "skyblue", xlab = "Income", ylab = "Frequency", main = "Distribution of Income")


# Assuming 'data' is your dataframe
crime_rate_column <- data$Crime.Rate..per.1000.residents.

# Replace values 40.4 and 83.5 with 18.79
crime_rate_column[crime_rate_column == 40.4 | crime_rate_column == 83.5] <- 18.79

```

```r

# Replace all crime rates less than 8.4 with 18.79
crime_rate_column[crime_rate_column < 8.4] <- 18.79

# Update the column in the original data frame
data$Crime.Rate..per.1000.residents. <- crime_rate_column

# Create a histogram for the modified 'Crime.Rate..per.1000.residents.' column
hist(data$Crime.Rate..per.1000.residents., breaks = 20, col = "skyblue",
     xlab = "Crime Rate per 1000 Residents", ylab = "Frequency",
     main = "Distribution of Crime Rate (Filtered)")

# Assuming 'data' is your dataframe
property_price_column <- data$Property.Price

# Calculate the average excluding NA values
average_property_price <- mean(property_price_column, na.rm = TRUE)

# Replace NA values with the calculated average
property_price_column[is.na(property_price_column)] <- average_property_price

# Update the column in the original data frame
data$Property.Price <- property_price_column

# Assuming 'data' is your dataframe
property_price_column <- data$Property.Price

# Create a histogram for the modified 'Property.Price' column
hist(property_price_column, breaks = 20, col = "skyblue",
     xlab = "Property Price", ylab = "Frequency",
     main = "Distribution of Property Prices")

# Assuming 'data' is your dataframe
unemployment_rate_column <- data$Unemployment.Rate

# Calculate the 10th and 90th percentile
p10 <- quantile(unemployment_rate_column, 0.1)
p90 <- quantile(unemployment_rate_column, 0.9)

# Calculate the average of the column
average_unemployment_rate <- mean(unemployment_rate_column, na.rm = TRUE)

# Replace values below the 5th percentile and above the 95th percentile with the average
unemployment_rate_column[unemployment_rate_column < p10] <- average_unemployment_rate
unemployment_rate_column[unemployment_rate_column > p90] <- average_unemployment_rate
```

```r
# Update the column in the original data frame
data$Unemployment.Rate <- unemployment_rate_column

# Assuming 'data' is your dataframe
unemployment_rate_column <- data$Unemployment.Rate

# Create a histogram for the modified 'Unemployment.Rate' column
hist(unemployment_rate_column, breaks = 20, col = "skyblue",
     xlab = "Unemployment Rate", ylab = "Frequency",
     main = "Distribution of Unemployment Rate")
summary(data)

# Linear Regression Model
model <- lm(Price ~ ., data = data)
print(model)
summary(model)

# Get the R-squared value
r_squared <- summary(model)$r.squared
# Print the R-squared value
print(r_squared)

# MAPE
predicted_values <- predict(model, data)
actual_values <- data$Price
mape <- mean(abs((actual_values - predicted_values) / actual_values)) * 100
print(mape)

# Step Wise Regression
stepwise_model <- step(model, direction = "both")
summary(stepwise_model)
r_squared_step <- summary(stepwise_model)$r.squared

# Step Wise Mape
predicted_value_step <- predict(stepwise_model, data = data)
mape_step <- mean(abs(data$Price - predicted_value_step) / data$Price) * 100
print(mape_step)

# Save Data as csv
write.csv(data, "AAMD PROJECT 1 FINAL GROUP 11.csv", row.names = FALSE)

## CROSS VALIDATION USING K-FOLDS
install.packages("tidyverse")
library(tidyverse)
```

```r
install.packages("caret")
library(caret)

num_folds <- 5

# Define the control parameters for k-fold cross-validation
train_control <- trainControl(method = "cv", number = num_folds)

# Create a model using k-fold cross-validation (e.g., linear regression as an example)
model1 <- train(Price ~ ., data = data, method = "lm", trControl = train_control)

# Get cross-validation results
cv_results <- model1$resample
print(cv_results)

```

Appendix B: Structure of the data before and after Feature Engineering

Before Feature Engineering

```
> str(data)
'data.frame':   39458 obs. of  28 variables:
 $ Host_ID                          : int  2787 2845 4632 4869 7192 7322 7356 896
7 7490 7549 ...
 $ Price                            : chr  " $149 " " $225 " " $150 " " $89 " ...
 $ Income                           : chr  " $76,607 " " $98,510 " " $47,990 " "
 $98,510 " ...
 $ Crime.Rate..per.1000.residents.: chr  "18.8%" "16.4%" "17.7%" "0.0%" ...
 $ Population..by.race.             : chr  "30.9%" "51.6%" "17.5%" "44.8%" ...
 $ Property.Price                   : chr  " $839,000 " " $6,400,000 " " $2,257,5
00 " " $- " ...
 $ Educational.Level                : chr  "83.30%" "78.00%" "17.00%" "0.00%" ...
 $ Land.Size..sq.ft.                : int  42408780 62839709 40338738 15200000 25
000826 42408780 77607794 23435280 50631572 42408780 ...
 $ Unemployment.Rate                : chr  "4.30%" "9.47%" "15.90%" "0.00%" ...
 $ Minimum_Nights                   : int  1 1 3 1 10 3 45 2 2 1 ...
 $ Number_of_Reviews                : int  9 45 0 270 9 74 49 430 118 160 ...
 $ Calculated_Host_Listings_Count : int  6 2 1 1 1 1 1 1 1 4 ...
 $ Availability_365                 : int  365 355 365 194 0 129 0 220 0 188 ...
 $ Hospital_Count_5km               : int  2 0 0 2 0 2 0 0 0 2 ...
 $ Subway_Count_1km                 : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Park_Count_5km                   : int  0 14 0 0 0 0 0 0 14 0 ...
 $ Mall_Count_5km                   : int  4 16 2 0 0 0 0 0 0 0 ...
 $ Tourist_Dest_Count_5km           : int  0 0 0 0 0 0 0 0 0 0 ...
 $ X                                : logi  NA NA NA NA NA NA ...
 $ X.124                            : chr  "" "" "SUMMARY OUTPUT" "" ...
 $ X.1                              : chr  "" "" "" "" ...
 $ X.284                            : chr  " $50 " "" "" "" ...
 $ X.2                              : chr  "" "" "" "" ...
 $ X.3                              : chr  "" "" "" "" ...
 $ X.4                              : chr  "" "" "" "" ...
 $ X.5                              : chr  "" "" "" "" ...
 $ X.6                              : chr  "" "" "" "" ...
 $ X.7                              : chr  "" "" "" "" ...
```

After Feature Engineering

```
> str(data)
'data.frame':   39458 obs. of  17 variables:
 $ Price                          : num  149 225 150 89 80 200 60 79 79 150 ...
 $ Income                         : num  76607 98510 86384 98510 86384 ...
 $ Crime.Rate..per.1000.residents.: num  18.8 16.4 17.7 18.8 22.3 ...
 $ Population..by.race.           : num  30.9 51.6 17.5 44.8 13.9 30.9 32.7 51.6 63.6 30.9 ...
 $ Property.Price                 : num  839000 6400000 2257500 2862348 3128450 ...
 $ Educational.Level              : num  83.3 78 17 0 37 83.3 43 12.7 72.8 83.3 ...
 $ Land.Size..sq.ft.              : int  42408780 62839709 40338738 15200000 25000826 42408780 7760
7794 23435280 50631572 42408780 ...
 $ Unemployment.Rate              : num  4.3 9.47 8.43 8.43 12.19 ...
 $ Minimum_Nights                 : int  1 1 3 1 10 3 45 2 2 1 ...
 $ Number_of_Reviews              : int  9 45 0 270 9 74 49 430 118 160 ...
 $ Calculated_Host_Listings_Count : int  6 2 1 1 1 1 1 1 1 4 ...
 $ Availability_365               : int  365 355 365 194 0 129 0 220 0 188 ...
 $ Hospital_Count_5km             : int  2 0 0 2 0 2 0 0 0 2 ...
 $ Subway_Count_1km               : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Park_Count_5km                 : int  0 14 0 0 0 0 0 0 14 0 ...
 $ Mall_Count_5km                 : int  4 16 2 0 0 0 0 0 0 0 ...
 $ Tourist_Dest_Count_5km         : int  0 0 0 0 0 0 0 0 0 0 ...
```

Appendix C: Linear Regression Model Output and MAPE

```
> summary(model)

Call:
lm(formula = Price ~ ., data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-137.00  -41.99  -11.93   33.30  189.92

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                      2.480e+01  2.796e+00   8.869  < 2e-16 ***
Income                          -6.961e-05  2.016e-05  -3.454 0.000554 ***
Crime.Rate..per.1000.residents.  3.107e+00  1.006e-01  30.891  < 2e-16 ***
Population..by.race.             9.512e-01  2.717e-02  35.006  < 2e-16 ***
Property.Price                   2.465e-06  1.592e-07  15.481  < 2e-16 ***
Educational.Level                1.609e-01  1.664e-02   9.666  < 2e-16 ***
Land.Size..sq.ft.               -2.310e-07  2.149e-08 -10.749  < 2e-16 ***
Unemployment.Rate                1.109e+00  1.852e-01   5.986 2.17e-09 ***
Minimum_Nights                   8.408e-03  1.360e-02   0.618 0.536332
Number_of_Reviews               -3.563e-02  6.171e-03  -5.774 7.82e-09 ***
Calculated_Host_Listings_Count   2.248e-01  9.736e-03  23.092  < 2e-16 ***
Availability_365                 1.904e-02  2.248e-03   8.468  < 2e-16 ***
Hospital_Count_5km               3.178e-01  3.525e-01   0.902 0.367226
Subway_Count_1km                -1.400e+00  4.805e-01  -2.914 0.003574 **
Park_Count_5km                  -9.961e-03  7.108e-02  -0.140 0.888545
Mall_Count_5km                  -1.912e-01  5.854e-02  -3.267 0.001089 **
Tourist_Dest_Count_5km           5.815e-03  5.516e-01   0.011 0.991588
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 53.81 on 39441 degrees of freedom
Multiple R-squared:  0.1317,    Adjusted R-squared:  0.1313
F-statistic: 373.9 on 16 and 39441 DF,  p-value: < 2.2e-16
```

```
> predicted_values <- predict(model, data)
> actual_values <- data$Price
> mape <- mean(abs((actual_values - predicted_values) / actual_values)) * 100
> print(mape)
[1] 42.551
```

Appendix D: Linear Regression Stepwise Model Output and MAPE

```
> summary(stepwise_model)

Call:
lm(formula = Price ~ Income + Crime.Rate..per.1000.residents. +
    Population..by.race. + Property.Price + Educational.Level +
    Land.Size..sq.ft. + Unemployment.Rate + Number_of_Reviews +
    Calculated_Host_Listings_Count + Availability_365 + Subway_Count_1km +
    Mall_Count_5km, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-136.91  -42.01  -11.90   33.32  189.76

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     2.490e+01  2.790e+00   8.923  < 2e-16 ***
Income                         -6.961e-05  2.015e-05  -3.454 0.000553 ***
Crime.Rate..per.1000.residents.  3.107e+00  1.006e-01  30.893  < 2e-16 ***
Population..by.race.             9.516e-01  2.717e-02  35.026  < 2e-16 ***
Property.Price                   2.466e-06  1.592e-07  15.493  < 2e-16 ***
Educational.Level                1.610e-01  1.664e-02   9.677  < 2e-16 ***
Land.Size..sq.ft.               -2.311e-07  2.148e-08 -10.755  < 2e-16 ***
Unemployment.Rate                1.111e+00  1.852e-01   5.998 2.01e-09 ***
Number_of_Reviews               -3.586e-02  6.138e-03  -5.842 5.20e-09 ***
Calculated_Host_Listings_Count   2.273e-01  9.438e-03  24.079  < 2e-16 ***
Availability_365                 1.920e-02  2.224e-03   8.633  < 2e-16 ***
Subway_Count_1km                -1.395e+00  4.800e-01  -2.905 0.003672 **
Mall_Count_5km                  -1.840e-01  5.697e-02  -3.229 0.001243 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 53.81 on 39445 degrees of freedom
Multiple R-squared:  0.1317,    Adjusted R-squared:  0.1314
F-statistic: 498.4 on 12 and 39445 DF,  p-value: < 2.2e-16

> predicted_value_step <- predict(stepwise_model, data = data)
> mape_step <- mean(abs(data$Price - predicted_value_step) / data$Price) * 100
> r_squared_step <- summary(stepwise_model)$r.squared
> print(mape_step)
[1] 42.55224
```

Appendix E: K-Folds Cross Validation

```
> print(cv_results)
      RMSE  Rsquared      MAE Resample
1 53.84239 0.1312921 43.94962    Fold1
2 53.78529 0.1254386 43.79023    Fold2
3 54.35936 0.1295743 44.13748    Fold3
4 54.05807 0.1260831 44.03919    Fold4
5 53.04478 0.1435535 43.43724    Fold5
```