Airbnb Price Prediction in Seattle, Washington

Riti Dahal

University of South Dakota

DSCI 519 Advanced Business Analytics Modeling

Dr. Hanus

November 2, 2025

Table of Contents

Tables

Introduction

Airbnb is a short-term rental website that has revolutionized the hospitality sector by allowing homeowners to make money by renting out vacant properties as well as providing travelers with more flexible lodging options. The main objective of this project is to create a predictive model that determines the key factors influencing listing prices for Airbnb listings in Seattle, Washington. The analysis uses SAS to explore both categorical and numerical factors, evaluate their statistical correlations with pricing, and build a regression model to measure their impact.

Understanding these pricing determinants provides meaningful insights for both hosts and the Airbnb platform. For hosts, accurate and data-driven pricing strategies can help them improve occupancy rates, increase revenue, and stay competitive within the local market. For Airbnb, these insights can strengthen automated pricing algorithms, enhance transparency, and highlight how property attributes and neighborhood dynamics shape traveler preferences. By integrating systematic data preparation, exploratory analysis, and predictive modeling, this project demonstrates how data analytics can be applied to support evidence-based decision-making in Seattle's short-term rental market.

Data Exploration

The dataset used for this analysis was obtained from the Inside Airbnb open data repository, which provides information about Airbnb listings across major cities. For this project, I have chosen Seattle, Washington. After importing the dataset into SAS, the first two rows of the PROC CONTENTS output show that there are a total of 6,862 observations and 79 variables in the Seattle dataset.

Table 1 Data Contents Overview

The CONTENTS Procedure

Data Set Name	WORK.SEATTLE	Observations	6862
Member Type	DATA	Variables	79
Engine	V9	Indexes	0
Created	10/30/2025 20:43:43	Observation Length	13568
Last Modified	10/30/2025 20:43:43	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

Table 2 Numeric Variables

Variable	Type	Description
accommodates	Numeric	Number of guests the property can accommodate
bathrooms	Numeric	Number of bathrooms in the listing
bedrooms	Numeric	Number of bedrooms
beds	Numeric	Number of beds available
price	Numeric	Nightly price of the listing
minimum_nights	Numeric	Minimum nights required for booking
maximum_nights	Numeric	Maximum number of nights allowed per booking
minimum_minimum_nights	Numeric	Minimum of minimum nights allowed
maximum_minimum_nights	Numeric	Maximum of minimum nights allowed
minimum_maximum_nights	Numeric	Minimum of maximum nights allowed
maximum_maximum_nights	Numeric	Maximum of maximum nights allowed
minimum_nights_avg_ntm	Numeric	Average minimum nights across time
maximum_nights_avg_ntm	Numeric	Average maximum nights across time
availability_30	Numeric	Availability within the next 30 days
availability_60	Numeric	Availability within the next 60 days
availability_90	Numeric	Availability within the next 90 days
availability_365	Numeric	Availability within the next 365 days
number_of_reviews	Numeric	Total number of reviews received
number_of_reviews_ltm	Numeric	Number of reviews in the last 12 months
number_of_reviews_l30d	Numeric	Number of reviews in the last 30 days
reviews_per_month	Numeric	Average number of reviews per month
host_listings_count	Numeric	Number of active listings by the host
calculated_host_listings_count	Numeric	Total listings under the host
calculated_host_listings_count_e	Numeric	Number of entire-home listings under the host
calculated_host_listings_count_p	Numeric	Number of private-room listings under the host
calculated_host_listings_count_s	Numeric	Number of shared-room listings under the host
estimated_occupancy_l365d	Numeric	Estimated occupancy rate in the past 365 days
estimated_revenue_l365d	Numeric	Estimated revenue generated in the past 365 days
review_scores_rating	Numeric	Overall guest rating score
review_scores_accuracy	Numeric	Rating for accuracy
review_scores_cleanliness	Numeric	Rating for cleanliness
review_scores_checkin	Numeric	Rating for check-in experience
review_scores_communication	Numeric	Rating for communication
review_scores_location	Numeric	Rating for location
review_scores_value	Numeric	Rating for value

Table 3 Categorical Variables

Variable	Type	Description
id	Character	Unique identifier for each listing
listing_url	Character	URL of the Airbnb listing

name	Character	Title of the listing
description	Character	Full text description of the listing
neighborhood_overview	Character	Description of the neighborhood
picture_url	Character	Main image URL of the listing
host_id	Character	Unique identifier for the host
host_name	Character	Name of the host
host_location	Character	Location of the host
host_about	Character	Host's profile description
host_response_time	Character	Average response time of the host
host_response_rate	Character	Host's response rate percentage
host_acceptance_rate	Character	Host's booking acceptance rate
host_is_superhost	Character	Indicates if the host is a Superhost
host_has_profile_pic	Character	Indicates if the host has a profile picture
host_identity_verified	Character	Indicates if host identity is verified
host_neighbourhood	Character	Neighborhood where host resides
host_verifications	Character	Types of verifications completed by host
neighbourhood	Character	General neighborhood name
neighbourhood_cleansed	Character	Cleaned neighborhood field used for grouping
neighbourhood_group_cleansed	Character	Broader neighborhood grouping
property_type	Character	Type of property (e.g., Apartment, House, Room)
room_type	Character	Type of room offered (e.g., Entire home, Private room)
amenities	Character	List of amenities available in the property
has_availability	Character	Indicates if the listing is currently available
instant_bookable	Character	Indicates if the property can be instantly booked
license	Character	Business license number of the listing
source	Character	Source of data extraction

The dataset from Airbnb Seattle includes a combination of categorical and numerical variables that describe location, host information, listing attributes, and guest ratings. While category variables like property_type, room_type, and neighbourhood_cleansed reflect qualitative characteristics of listings, numerical variables like accommodates, bedrooms, and availability capture quantifiable property features.

Dependent Variable Analysis

The PROC UNIVARIATE procedure is a great way to investigate a continuous variable.

Table 4 Moments and Basic Statistical Measures

The UNIVARIATE Procedure
Variable: price (price)

Moments			
N	6227	Sum Weights	6227
Mean	473.938494	Sum Observations	2951215
Std Deviation	3290.41086	Variance	10826803.6
Skewness	13.0224612	Kurtosis	170.315589
Uncorrected SS	6.88064E10	Corrected SS	6.74077E10
Coeff Variation	694.269595	Std Error Mean	41.6975652

Basic Statistical Measures			
Location		Variability	
Mean	473.9385	Std Deviation	3290
Median	177.0000	Variance	10826804
Mode	165.0000	Range	50024
		Interquartile Range	150.00000

The above table shows that the data contained 6227 observations and that the average value for price is $473.93. The standard deviation is larger than the mean value, which indicates that some big outliers are pulling the average price up. The skewness value of (13.02) indicates a right skewed distribution which will be further transformed to create a normal distribution. Likewise, there is a significant difference between the mean (473.93) and median (177.00) values. This difference is further evidence that outlier values are affecting the mean value. The range shows the difference between the highest and the lowest value. There is a $50024 per diem range for the price. The high kurtosis value (170.31) suggest heavy tails and a peaked distribution.

Table 5 Quartiles and Missing Values Tables

Quantiles (Definition 5)	
Level	Quantile
100% Max	50034
99%	1295
95%	509
90%	393
75% Q3	269
50% Median	177
25% Q1	119
10%	84
5%	67
1%	44
0% Min	10

Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	635	9.25	100.00

The quantiles of the variable shows the minimum and maximum values as well as values at certain percentage cutoff points. The 99% cutoff value is $1295 and the 100% value is $50034. This shows a large difference between them, meaning that the top 1% of the observations is extreme outlier points. It also shows that 635 observations of the total dataset have missing values for the price variable.

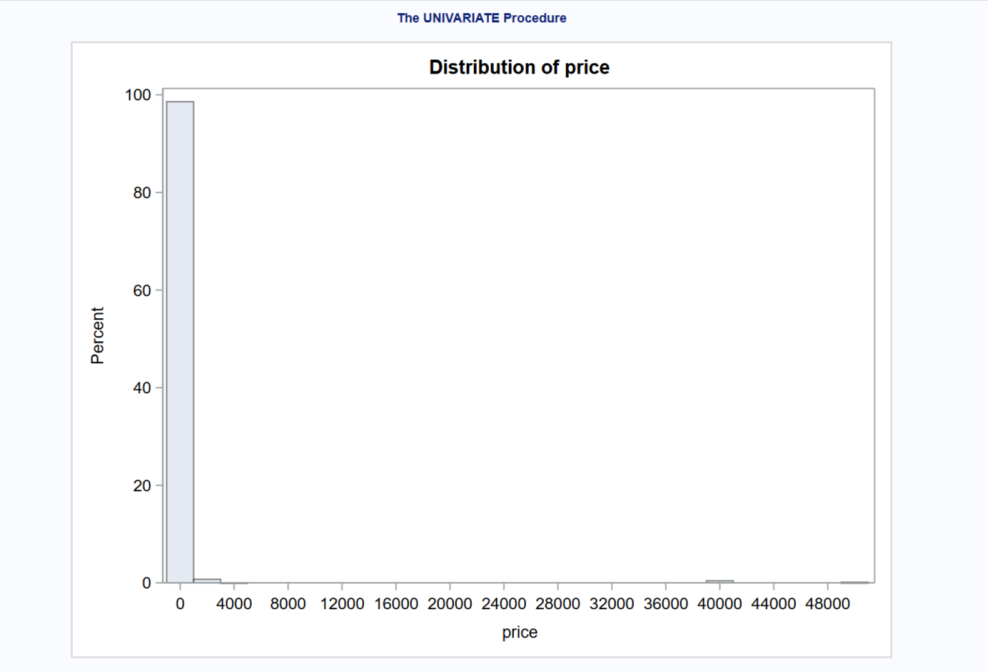


Figure 1 Histogram of Price

The histogram shows a highly skewed distribution with a very long right sided tail. This suggests that the price variable has significant outliers on the high end that needs to be fixed using different transformation.

Data Transformation and Eliminating

The right-skewed distribution of the price variable was addressed by a log transformation to improve my predictive modeling. As a way to address the presence of extreme values, I have eliminated values below $44 (1st percentile) and over $1,295 (99th percentile) based on a quantile analysis. In order to eliminate possible outliers while keeping a sizable amount of the dataset, this range was selected. The table below provides an insights from the summary statistics of distribution of the log transformed ‘Price’.

Table 6 Moments Table of Log Adjusted Dependent Variable

The UNIVARIATE Procedure
Variable: price_log

Moments			
N	6112	Sum Weights	6112
Mean	5.19598839	Sum Observations	31757.881
Std Deviation	0.58730194	Variance	0.34492357
Skewness	0.20724598	Kurtosis	-0.0142462
Uncorrected SS	167121.409	Corrected SS	2107.82796
Coeff Variation	11.302988	Std Error Mean	0.00751225

The data contained 6112 observations and that the average value for price_log is (5.19). The standard deviation (0.58) is smaller than the mean value, which indicates a more normalized distribution compared to the untransformed price variable. The skewness value of (0.20) indicates a nearly symmetric distribution with a slight right skew, demonstrating that the log transformation successfully reduced the extreme skewness present in the original price data. The relatively low kurtosis value of (-0.01) suggests that the distribution is approximately normal with neither heavy tails nor excessive peaking. The coefficient of variation (11.30%) indicates moderate variability in the log-transformed prices relative to the mean, suggesting reasonable consistency in pricing patterns across listings.

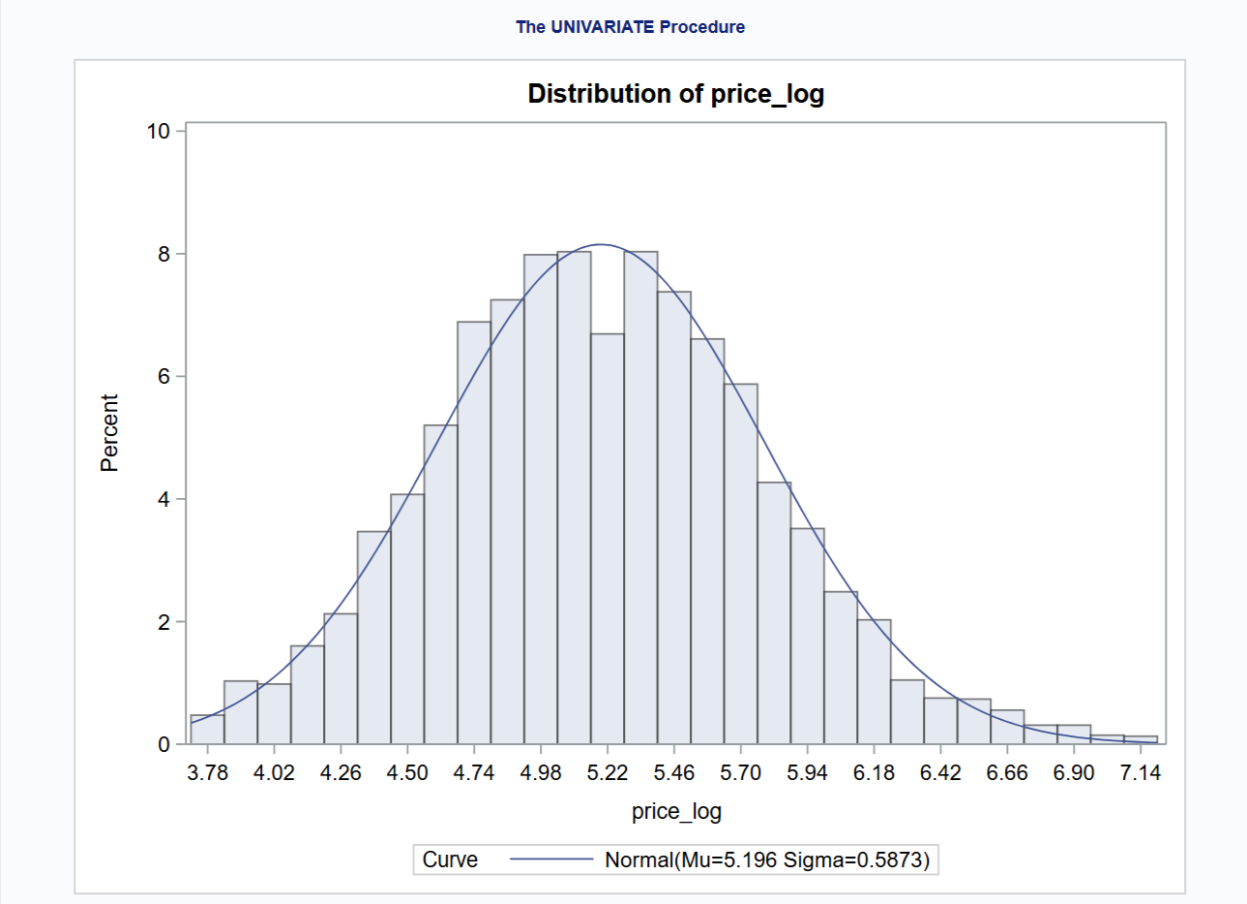


Figure 2 Histogram of Price (Log Adjusted)

As we can see that the log transformation has successfully addressed the right-skewed distribution of the 'Price' variable, providing a more symmetric and normalized distribution for further analysis.

Numeric Variable Transformation and Feature Engineering

For the analysis I have chosen 11 numerical variables as it capture the key factors that directly influence Airbnb pricing and booking behavior. The explanation is as below:

- A macro variable name ‘num_var’ has been created to store all the key numerical predicators such as accomodates, bedrooms, beds, bathrooms, minimum nights, maximum_nights, availability_30, availability_60, availability_90, availability_365 and host_listings_count. These variables directly influence the pricing decisions as it provides a detail information about the property size, property availability and host experience.

- Likewise variables like number_of_reviews and review_scores were excluded as it represents the performance after listing. Host_since, also represents information that is not directly correlated with pricing, thus, was excluded from the analysis.

Table 7 PROC MEANS Missing Data Analysis

The MEANS Procedure

Variable	Label	N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
host_listings_count	host_listings_count	5888	224	1.0000000	4889.00	300.7026155	3.0000000	1111.54
accommodates	accommodates	6112	0	1.0000000	16.0000000	4.0924411	4.0000000	2.6088094
bathrooms	bathrooms	6112	0	0	16.0000000	1.4490347	1.0000000	0.8028926
bedrooms	bedrooms	6105	7	0	17.0000000	1.6552007	1.0000000	1.2112403
beds	beds	6111	1	0	15.0000000	2.1903126	2.0000000	1.5332467
minimum_nights	minimum_nights	6112	0	1.0000000	365.0000000	11.0620092	2.0000000	18.6175519
maximum_nights	maximum_nights	6112	0	1.0000000	10000.00	403.2814136	365.0000000	437.5584186
availability_30	availability_30	6112	0	0	30.0000000	7.9136126	6.0000000	8.2585093
availability_60	availability_60	6112	0	0	60.0000000	21.4733312	19.0000000	17.1856252
availability_90	availability_90	6112	0	0	90.0000000	41.7177683	41.0000000	24.9002794
availability_365	availability_365	6112	0	0	365.0000000	211.8136453	230.0000000	107.4964174

From the Table 7, we can see that some of the variables like host_listing_count, bedrooms and beds have missing values. The missing values cause a problem when building a regression model as it will exclude any observation with single missing values. Therefore, to handle this I have used the missing value imputation technique. The explanation of the missing value imputation technique is as below:

- Host listing count has 224 missing values, so here I have directly imputed it with the global mean. This technique maximizes the sample size for analysis while retaining the variable's overall average by substituting the mean of the available observations for missing values.
- **For bedrooms and beds**

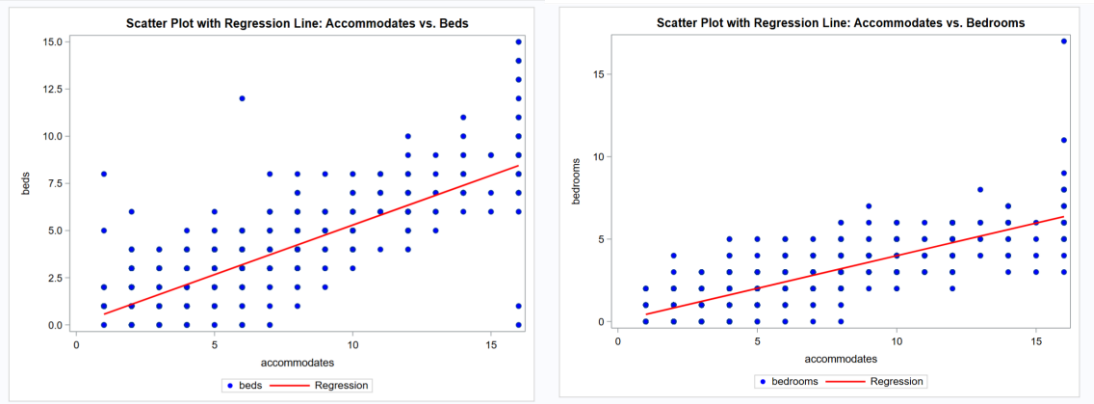


Figure 3 Scatter Plot with Regression Line

Based on the accommodates variable, a group mean imputation method was used to handle missing values in bedrooms and beds. A strong linear relationship between accommodates and both bedrooms and beds were demonstrated by scatter plots with fitted regression lines, indicating that accommodate was an appropriate grouping factor. If an entry was missing, it was replaced with the average number of bedrooms or beds among listings with the same amount of space.

Categorization of Numeric Variables

The variables of beds, bedrooms, and bathrooms showed 18, 16, and 17 unique levels, respectively. This indicates a wide variation in property size. Thus, to enhance interpretability and model stability, these variables were grouped into three broader categories, such as 0–1, 1–2, and more than 2. This grouping approach simplifies the analysis while preserving the logical progression of property capacity.

Frequency Distribution of Bathroom Categories

The FREQ Procedure

bath_cat	Frequency
0-1 Bathroom	4002
1-2 Bathroom	1340
More than 2 Bathroom	771

Frequency Distribution of Bedroom Categories

The FREQ Procedure

bed_cat	Frequency
0-1 Bedroom	3416
1-2 Bedroom	1506
More than 2 Bedroom	1191

Frequency Distribution of Bed Categories

The FREQ Procedure

beds_cat	Frequency
0-1 Bed	2528
1-2 Beds	1691
More than 2 Beds	1894

Figure 4 Frequency Distribution of Newly Created Variable

Collinearity Analysis and VIF

The correlation table shows that all the availability variables show extremely high correlations (r = 0.79 –0.93). The VIF also shows that the availability variable has VIFs above the threshold of 5. Likewise, other predictors, such as accommodates, minimum_nights, maximum_nights, and host_listings_count, exhibit only low to moderate correlations ($r < 0.4$), indicating they contribute unique information to the model. However, I have included the availability variables in the model as each one represents a different time frame for booking. These time frames can help hosts manage their listings and identify how demand changes over time.

Table 8 Correlation Table and VIF

Pearson Correlation Coefficients, N = 6112
Prob > |r| under H0: Rho=0

	accommodates	minimum_nights	maximum_nights	availability_30	availability_60	availability_90	availability_365	host_listings_count
accommodates	1.00000	-0.15369 <.0001	-0.04979 <.0001	0.06211 <.0001	0.07032 <.0001	0.07056 <.0001	0.04141 0.0012	-0.16597 <.0001
minimum_nights	-0.15369 <.0001	1.00000	0.25500 <.0001	-0.05935 <.0001	-0.09052 <.0001	-0.07147 <.0001	0.14085 <.0001	0.24690 <.0001
maximum_nights	-0.04979 <.0001	0.25500 <.0001	1.00000	-0.09219 <.0001	-0.10326 <.0001	-0.07633 <.0001	0.21157 <.0001	0.40019 <.0001
availability_30	0.06211 <.0001	-0.05935 <.0001	-0.09219 <.0001	1.00000	0.90061 <.0001	0.78999 <.0001	0.21469 <.0001	-0.16686 <.0001
availability_60	0.07032 <.0001	-0.09052 <.0001	-0.10326 <.0001	0.90061 <.0001	1.00000	0.93340 <.0001	0.26827 <.0001	-0.18795 <.0001
availability_90	0.07056 <.0001	-0.07147 <.0001	-0.07633 <.0001	0.78999 <.0001	0.93340 <.0001	1.00000	0.36418 <.0001	-0.17614 <.0001
availability_365	0.04141 0.0012	0.14085 <.0001	0.21157 <.0001	0.21469 <.0001	0.26827 <.0001	0.36418 <.0001	1.00000	0.09626 <.0001
host_listings_count	-0.16597 <.0001	0.24690 <.0001	0.40019 <.0001	-0.16686 <.0001	-0.18795 <.0001	-0.17614 <.0001	0.09626 <.0001	1.00000

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	Intercept	1	4.42721	0.01566	282.79	<.0001	0
accommodates	accommodates	1	0.15560	0.00202	76.94	<.0001	1.04972
minimum_nights	minimum_nights	1	-0.00191	0.00029489	-6.46	<.0001	1.13677
maximum_nights	maximum_nights	1	-0.00002118	0.00001329	-1.59	0.1111	1.27562
availability_30	availability_30	1	0.00843	0.00152	5.54	<.0001	5.95452
availability_60	availability_60	1	0.00165	0.00127	1.30	0.1944	17.93292
availability_90	availability_90	1	-0.00052876	0.00064567	-0.82	0.4129	9.74870
availability_365	availability_365	1	0.00015314	0.00005492	2.79	0.0053	1.31439
host_listings_count	host_listings_count	1	0.00016338	0.00000534	30.57	<.0001	1.28181

Standardizing Numeric Variables

To bring variables to the same scale, I have standardized the numeric variables which now have a mean 0 and standard deviation 1 for all the variables. This process helps to prevent the large range of variables from dominating the model and improves coefficient stability and interpretability. After the standardization, each coefficient displays the amount that the log (price) changes when a variable increases one standard deviation, or the average amount.

Table 9 Satndardized Numeric Variables

Verification of Standardized Numeric Variables

The MEANS Procedure

Variable	Label	N	Mean	Std Dev	Minimum	Maximum
accommodates	accommodates	6112	-2.03118E-14	1.0000000	-1.1853841	4.5643653
minimum_nights	minimum_nights	6112	-1.97686E-15	1.0000000	-0.5404582	19.0109845
maximum_nights	maximum_nights	6112	8.960165E-16	1.0000000	-0.9193776	21.9324282
availability_30	availability_30	6112	2.110005E-16	1.0000000	-0.9582374	2.6743794
availability_60	availability_60	6112	-4.44089E-16	1.0000000	-1.2494937	2.2417962
availability_90	availability_90	6112	-1.02957E-16	1.0000000	-1.6753936	1.9390237
availability_365	availability_365	6112	1.025576E-16	1.0000000	-1.9704252	1.4250368
host_listings_count	host_listings_count	6112	5.495904E-15	1.0000000	-0.2747110	4.2056876

Categorical Variable Transformation and Feature Engineering

For the analysis I have chosen 4 categorical variables to understand the qualitative features that may influence Airbnb listing prices in Seattle. These variables like property_type, room_type, neighbourhood_cleansed, instant_bookable reflect qualitative characteristics of listings. For instance, property and room type shows how hosts position their listings to target different market segments, ranging from budget-friendly shared rooms to high-end entire homes. The neighbourhood variable helps in identifying how Airbnb pricing is impacted by geographic location. Likewise, instant bookable variable indicates a simple booking method which may affect the competitiveness and rental rates.

Table 10 Number of Variable Levels

The FREQ Procedure

Number of Variable Levels				
Variable	Label	Levels	Missing Levels	Nonmissing Levels
property_type	property_type	52	1	51
room_type	room_type	4	1	3
neighbourhood_cleansed	neighbourhood_cleansed	89	1	88
instant_bookable	instant_bookable	3	1	2

The variable property_type contained the highest number of levels (52), followed by neighbourhood_cleansed (89). There were fewer levels for the variables room_type and instant_bookable, suggesting that their category structures were simpler. Each variable had one missing level, which was then eliminated to maintain the data accuracy.

Categorization of Categorical Variables

The original categorical variables had too many distinct levels. Thus, to make analysis easier and enhance model performance, the categories were combined into more comprehensive, meaningful groupings. For instance, the categories of rooms were grouped into Entire Unit and Room, while the types of properties were grouped into Apartment/Condo, House, Room, and Other. Likewise, neighborhoods were grouped according to market and geographic similarities

into four major groups: Downtown & Tourist Hub, Residential North, Residential South & West, and Other.

Table 11 Categorization of Categorical Variables

Categorization of Categorical Variables

The FREQ Procedure

property_group	Frequency
Apartment/Condo	2151
House	2970
Other	297
Room	694

room_group	Frequency
Entire Unit	5180
Room	932

neighborhood_group	Frequency
Downtown & Tourist Hub	1985
Other	1976
Residential North	1101
Residential South & West	1050

instant_group	Frequency
0	1993
1	4119

Data Partition

The dataset was split into training and testing sets using SAS's PROC SURVEYSELECT technique to ensure a reliable and targeted model evaluation. An 80/20 split was applied to maintain a balanced representation of listings across both sets. 80% of the dataset is used for training the prediction model, finding important predictors, and establishing correlations between variables. The testing dataset (20%) is set aside for performance validation on unobserved data, which aids in evaluating the model's capacity for generalization and prediction accuracy. Likewise, during the random sampling procedure, a fixed random seed (42) was used to guarantee reproducibility of results.

Model Building and Interpretability

After exploring, cleaning and transforming the data, three different predictive modelling techniques was used. For this I have used model such as LASSO Regression, Decision Tree and Random Forest to identify the factors that influence Airbnb listing prices in Seattle. The goal of comparing different models is to balance predictive accuracy and interpretability. Before modelling, a set of macro variables were created to ease the overall process.

LASSO Regression Model

The LASSO regression model was built using the PROC GLMSELECT with the log-transformed price (price_log) as the dependent variable. The model included both standardized numeric predictors that include accommodates, minimum_nights, maximum_nights,

availability_30, availability_60, availability_90, availability_365, and host_listings_count. Likewise, key categorical variables such as property_group, room_group, neighborhood_group, instant_group, bed_cat. Beds_cat and bath_cath are included in the model. LASSO regression involves selecting a subset of features which means it minimizes the influence of less significant predictors by shrinking their coefficient toward zero.

Table 12 LASSO Selection Summary

The GLMSELECT Procedure

LASSO Selection Summary

Step	Effect Entered	Effect Removed	Number Effects In	SBC
0	Intercept		1	-5177.8026
1	accommodates		2	-7037.8669
2	bed_cat_0-1 Bedroom		3	-7094.3773
3	property_group_Room		4	-7947.3630
4	host_listings_count		5	-8315.5959
5	bath_cat_0-1 Bathroom		6	-8316.4767
6	bed_cat_More than 2 Bedroom		7	-8509.5438
7	neighborhood_group_Downtown & Tourist Hub		8	-8741.5256
8	beds_cat_More than 2 Beds		9	-9298.0263
9	availability_365		10	-9355.7955
10	room_group_Entire Unit		11	-9392.9493
11	minimum_nights		12	-9404.3911
12	beds_cat_0-1 Bed		13	-9437.2329
13	instant_group_0		14	-9456.3814
14	bath_cat_More than 2 Bathroom		15	-9468.7410
15	property_group_Apartment/Condo		16	-9543.1432
16	neighborhood_group_Residential South & West		17	-9537.0800
17	maximum_nights		18	-9563.6176
18	neighborhood_group_Residential North		19	-9573.4020
19	property_group_Other		20	-9566.6860
20		room_group_Entire Unit	19	-9577.4996*
21	room_group_Entire Unit		20	-9569.3803

* Optimal Value of Criterion

The above selection summary shows that LASSO model begins with the intercept and progressively incorporates other predictors that enhanced the model. The SBC value steadily dropped as variables like accommodates, host listing count, bath_cat, property group and neighborhood group were added indicating improved model fit. The ideal balance between simplicity and precision was found at Step 21, when the SBC value was at its lowest. As we can see, 20 predictors that significantly influenced Airbnb pricing were included in the model. Among these, accommodates, property type, neighborhood group, and room type were the strongest and most reliable factors influencing listing prices in Seattle.

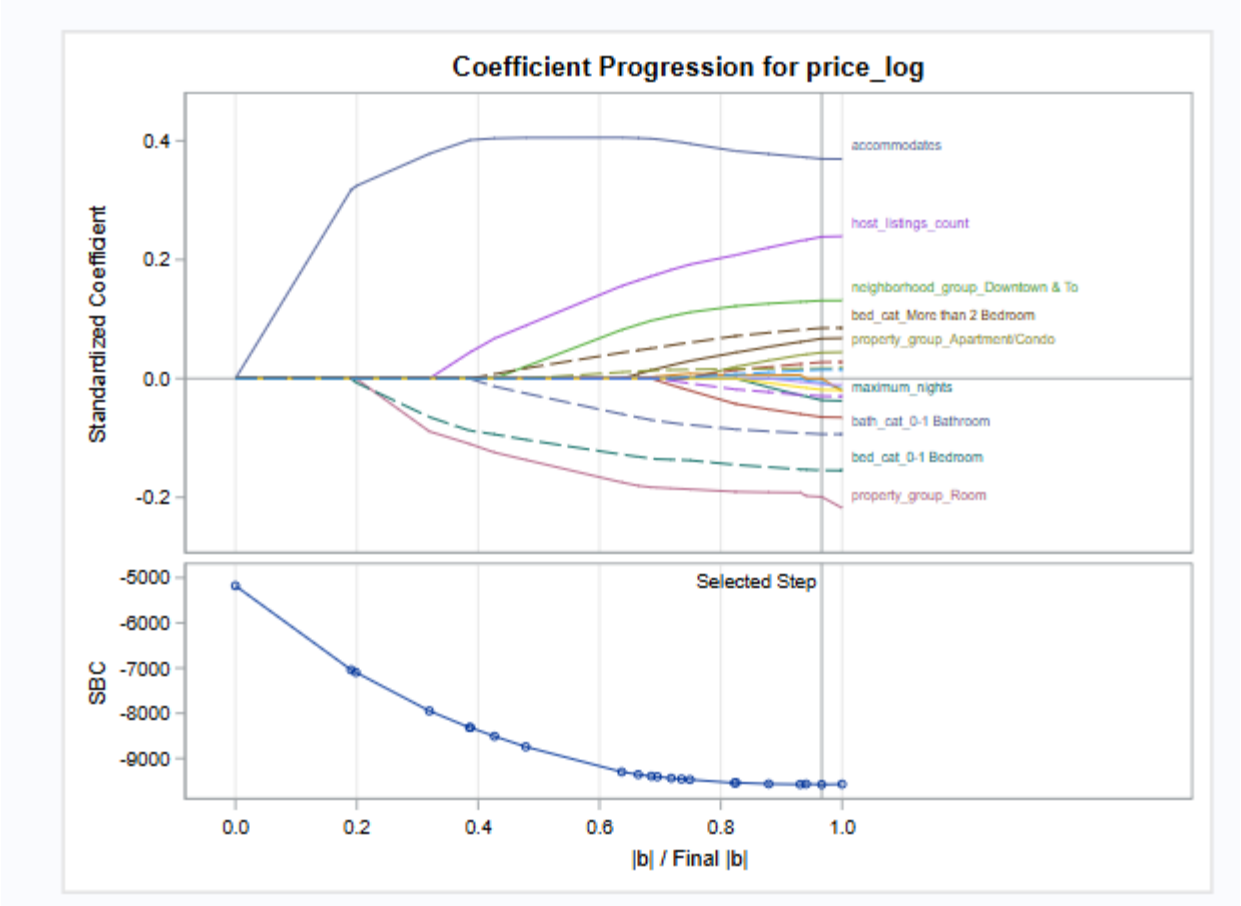


Figure 5 Coefficient Progression Plot

The above Coefficient Progression plot shows how the coefficients of each predictor change when the LASSO penalty increases. The top panel shows that two predictors accommodates, and host listing count are the most significant contributors while other predictors shrink toward zero but still have the predictive power. Likewise, the bottom panel indicates that the chosen model provides the best balance between simplicity and accuracy by showing the SBC gradually declining and reaching its lowest position close to the last step.

Table 13 Variable Selection

Parameter Estimates		
Parameter	DF	Estimate
Intercept	1	5.338917
accommodates	1	0.218104
minimum_nights	1	-0.038426
maximum_nights	1	-0.022480
availability_365	1	0.039617
host_listings_count	1	0.141710
property_group_Apartment/Condo	1	0.053815
property_group_Other	1	-0.018727
property_group_Room	1	-0.367030
neighborhood_group_Downtown & Tourist Hub	1	0.200174
neighborhood_group_Residential North	1	-0.018191
neighborhood_group_Residential South & West	1	-0.051118
bath_cat_0-1 Bathroom	1	-0.115930
bath_cat_More than 2 Bathroom	1	0.048375
bed_cat_0-1 Bedroom	1	-0.183237
bed_cat_More than 2 Bedroom	1	0.126040
beds_cat_0-1 Bed	1	-0.035286
beds_cat_More than 2 Beds	1	0.021777
instant_group_0	1	0.018648

Key Findings and Variable Importance

- The variable accommodates (Estimate = 0.2181) has the strongest positive effect on price. The coefficient indicates that, while all other variables are held constant, the log(price) increases by 0.2181 units for every standard deviation increase in the number of guests a listing can accommodate.
- The host listing count estimates (0.1417) indicating an increase in number of listings a host manages leads to a 0.14 increase in log price.
- Compared to the reference group (entire units), listings classified as private or shared rooms have a log (price) that is 0.37 cheaper. This suggests that the price of a shared room is significantly less than that of entire units.
- Categorized neighborhood groups by Downtown and Tourist Hub estimates (0.2002) highlighting the downtown area are costly compared to other groups.
- Property with 0-1 bedrooms estimates (-0.1832) indicating that smaller property with fewer bedrooms have low prices as compared to property with more than 2 bedrooms (estimates 0.126) being more expensive.
- When comparing rooms with 0-1 beds and more than 2 beds, it shows that price increases with the number of beds.

Overall, these variables represent the most significant factors of Airbnb pricing, reflecting that capacity, location, and property type are the primary determinants of listing value.

Regression Assumption

Regression assumptions were examined prior to assessing model performance to ensure the accuracy of the results. Diagnostic plots were used to evaluate the assumptions of linearity, independence, homoscedasticity, and normality.

Linearity Check

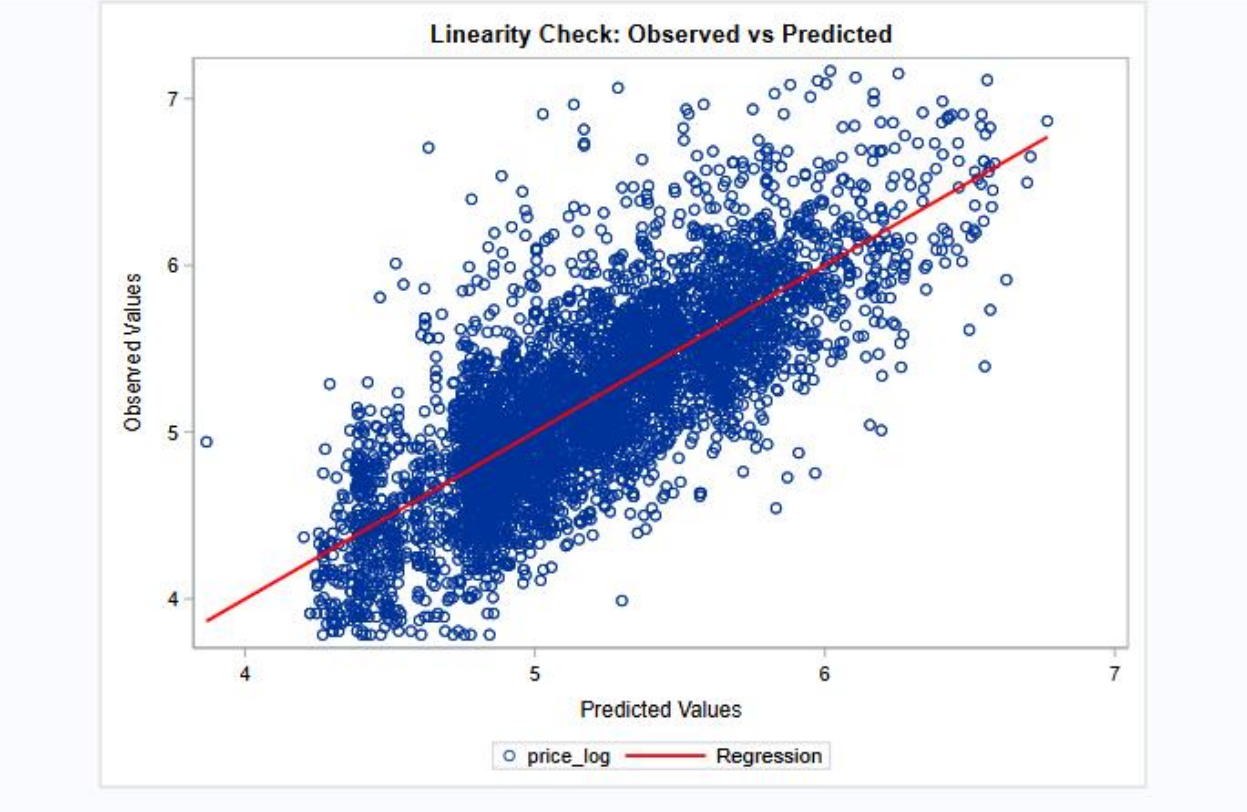


Figure 6 Linearity Check

This above plot checks the linearity of the model by comparing the observed and predicted log(price) values. The plot shows a strong linear relationship between the predicted and observed outcomes as the points are clustered around the red regression line. The model explains almost 61% of the variation in log(price), according to the R^2 value of (0.6059), and the Root MSE of (0.389) shows a low average prediction error. The model fits the data well and satisfies the linearity condition, as shown by the even distribution of data points around the line.

Normality Check: Histogram and Q-Q Plot of Residuals

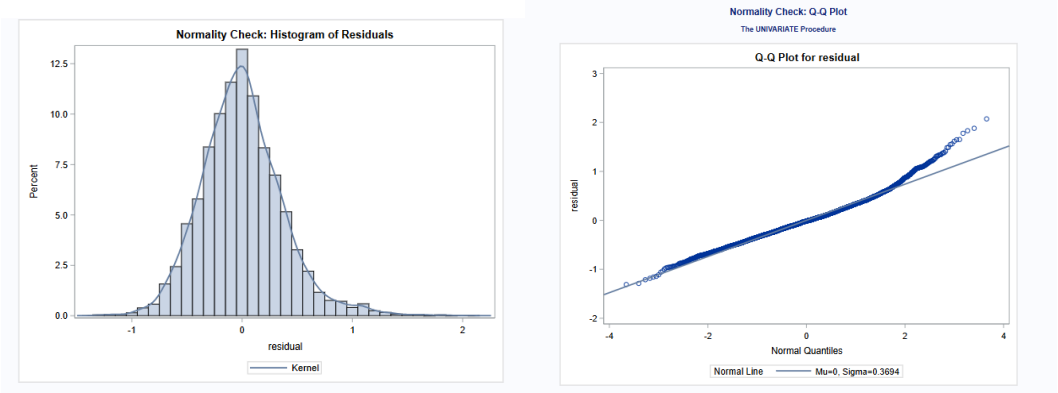


Figure 7 Normality Check: Histogram and Q-Q Plot of Residuals

From the graph we can see that the residuals are symmetrically distributed and closely follow the normal line. The near-linear Q-Q plot and bell-shaped histogram show that the residuals are approximately normal, satisfying the normality assumption of regression models.

Homoscedasticity Check: Residuals vs. Predicted Values

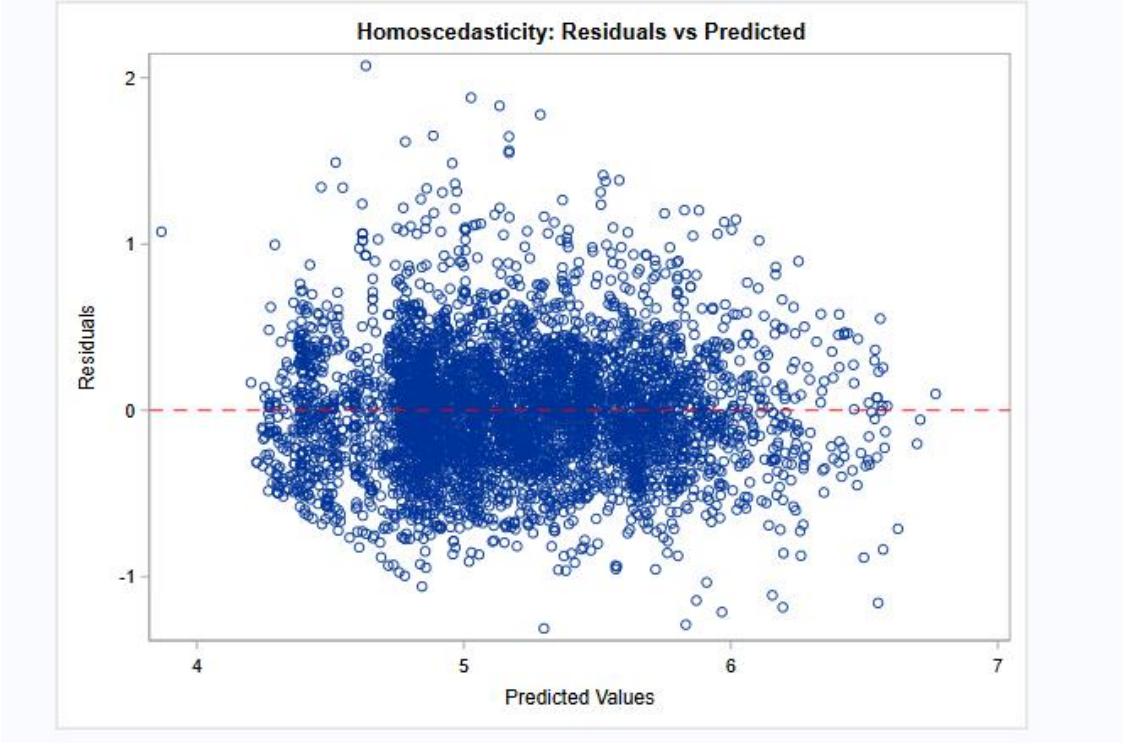


Figure 8 Homoscedasticity Check

The residuals are distributed uniformly around the zero line in the plot, with no apparent trend. This shows that the variance of residuals is constant across all predicted values, satisfying the regression model's homoscedasticity assumption.

Independence Check: Standardized Residuals vs. Predicted Values

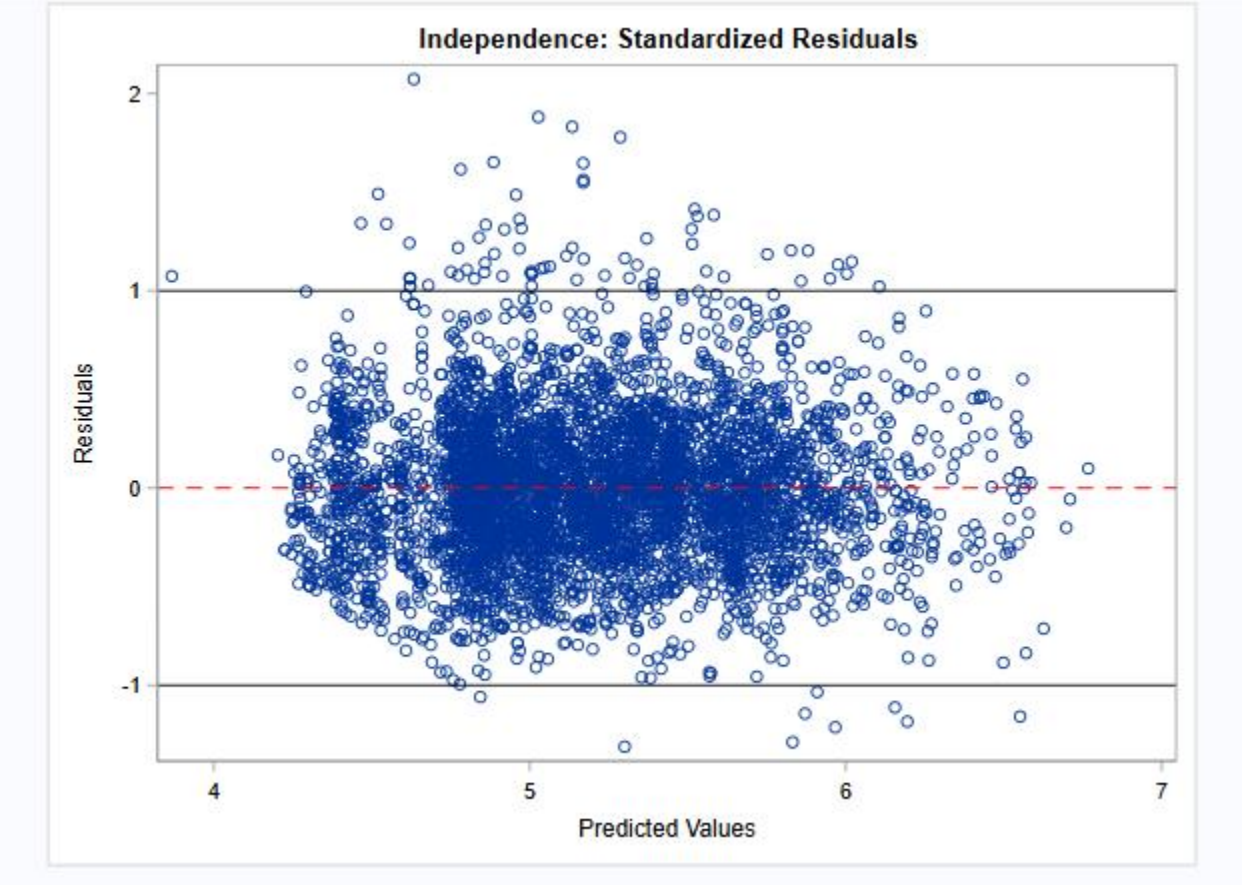


Figure 9 Independence Check

The residuals' independence from the predicted values is examined in this graphic. There is no autocorrelation or dependence among errors, as seen by the residuals' random distribution around the zero line and lack of noticeable trends or grouping. The spread is constant over the range of predicted values, indicating that the regression model's independence assumption is fulfilled.

Model Performance Evaluation

Training Accuracy

With an R-Square of (0.6059), the model explains 60% of the variation in the price log. The Adjusted R-Square (0.6058) shows that the fit is stable and not driven by overfitting. Likewise, the RMSE of (0.369) demonstrates that the average difference between the observed and predicted log(price) values is small, reflecting strong predictive accuracy. Furthermore, the F-statistic of 7513.97 with a p-value (< 0.0001) shows that the model is statistically significant, indicating that the chosen predictors together contribute for a large amount of price variation across listings.

Table 14 Training Accuracy

LASSO Model - R-Squared (Training Data)

The REG Procedure
Model: MODEL1
Dependent Variable: price_log

Number of Observations Read	4889
Number of Observations Used	4889

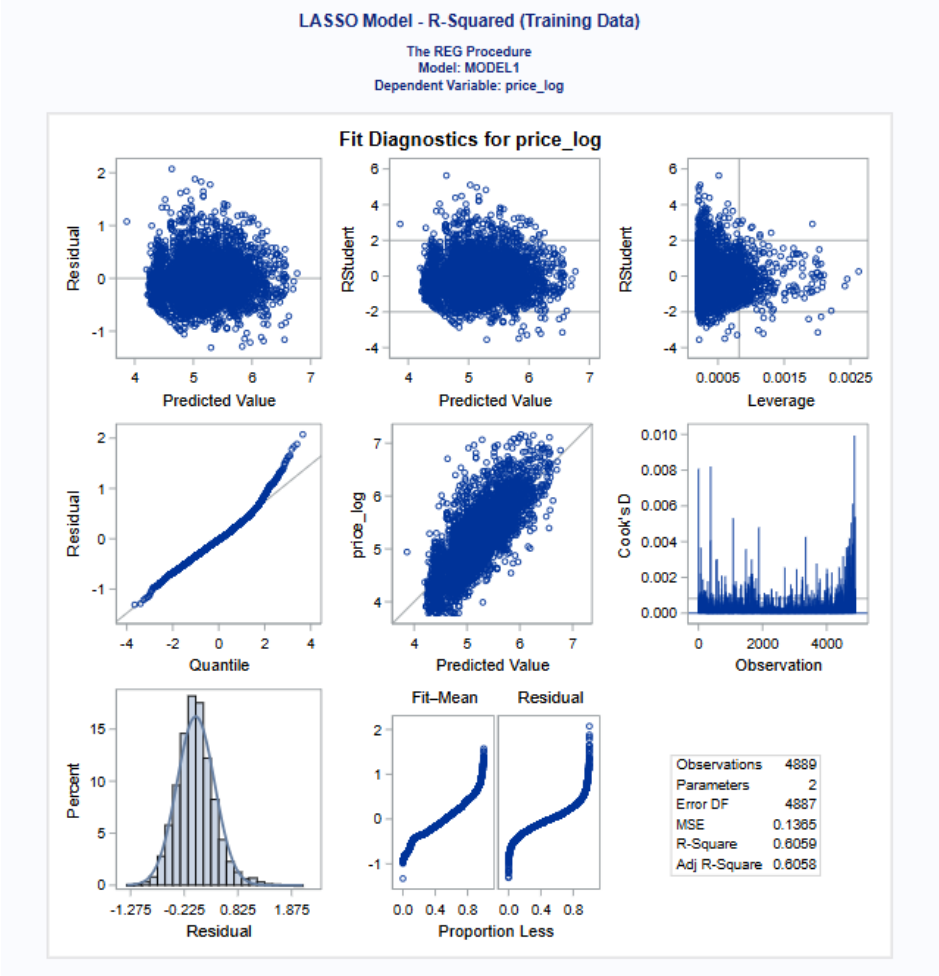
Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	1025.48704	1025.48704	7513.97	<.0001
Error	4887	666.96494	0.13648		
Corrected Total	4888	1692.45197			

Root MSE	0.36943	R-Square	0.6059
Dependent Mean	5.19401	Adj R-Sq	0.6058
Coeff Var	7.11259		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-0.00748	0.06024	-0.12	0.9012
p_price_log	1	1.00144	0.01155	86.68	<.0001

The diagnostic plots below show that residuals are randomly scattered around zero, indicating a good linear fit with constant variance. The Q–Q plot and histogram confirm that residuals are approximately normal, and the Cook's D plot shows no influential outliers. Overall, the LASSO model fits the data well and meets key regression assumptions.

Table 15 Fit Diagnostic Plots



Test Accuracy

The LASSO model's performance on the test dataset is summarized in this table, which demonstrates how effectively it generalizes new data. With an R-Square of (0.5974) and an Adjusted R-Square of (0.5970), the model performs similarly to the training performance, explaining almost 60% of the variation in log(price) for new listings. The RMSE of (0.3701) shows strong predictive accuracy with little error. Likewise, the F-value (1811.42) highlights that the model continues to be statistically significant when applied to the test data, demonstrating outstanding accuracy and consistency.

Table 16 Test Accuracy

LASSO Model - R-Squared (Test Data)

The REG Procedure
Model: MODEL1
Dependent Variable: price_log

Number of Observations Read	1223
Number of Observations Used	1223

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	248.06829	248.06829	1811.42	<.0001
Error	1221	167.21196	0.13695		
Corrected Total	1222	415.28025			

Root MSE	0.37006	R-Square	0.5974
Dependent Mean	5.20390	Adj R-Sq	0.5970
Coeff Var	7.11126		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	0.09481	0.12051	0.79	0.4316
p_price_log	1	0.97981	0.02302	42.56	<.0001

Decision Tree Model

For the analysis, the Decision Tree Model is created to capture non-linear correlations between variables and Airbnb listing pricing, using PROC HPSPLIT technique. In order to minimize variation in the targeted variable, log(price), this technique divides the data into smaller groups based on predictor variables.

From the table below, we can see that, in order to minimize variability among prices in each group, the model has split the data using variance criteria. To handle overfitting in the model, a cost-complexity pruning method was used where redundant branches of the tree were cut out. The final model turned out simpler and more interpretable with a maximal depth of 10 and 89 leaves after pruning, where at first it had 482 before pruning. It was trained using 4,889 observations, meaning it has utilized a full train data set effectively.

Table 17 Model Information

Decision Tree Model - Full Training Set

The HPSPLIT Procedure

Performance Information	
Execution Mode	Single-Machine
Number of Threads	2

Data Access Information			
Data	Engine	Role	Path
WORK.TRAIN	V9	Input	On Client
WORK.TREE_TRAIN_SCORED	V9	Output	On Client

Model Information	
Split Criterion Used	Variance
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	10
Tree Depth	10
Number of Leaves Before Pruning	462
Number of Leaves After Pruning	89

Number of Observations Read	4889
Number of Observations Used	4889

Cost Complexity Analysis

The plot below illustrates how the cost complexity parameter was used to prune the Decision Tree model to balance simplicity and accuracy. The ASE usually drops significantly before leveling off, which shows that adding more splits does not improve the performance. The most suitable tree size was found with a complexity parameter of (0.985) with 88 leaves, providing the lowest ASE value of (0.137). This point provides a well-balanced and comprehensible model by offering the optimal trade-off between reducing errors and preventing overfitting.

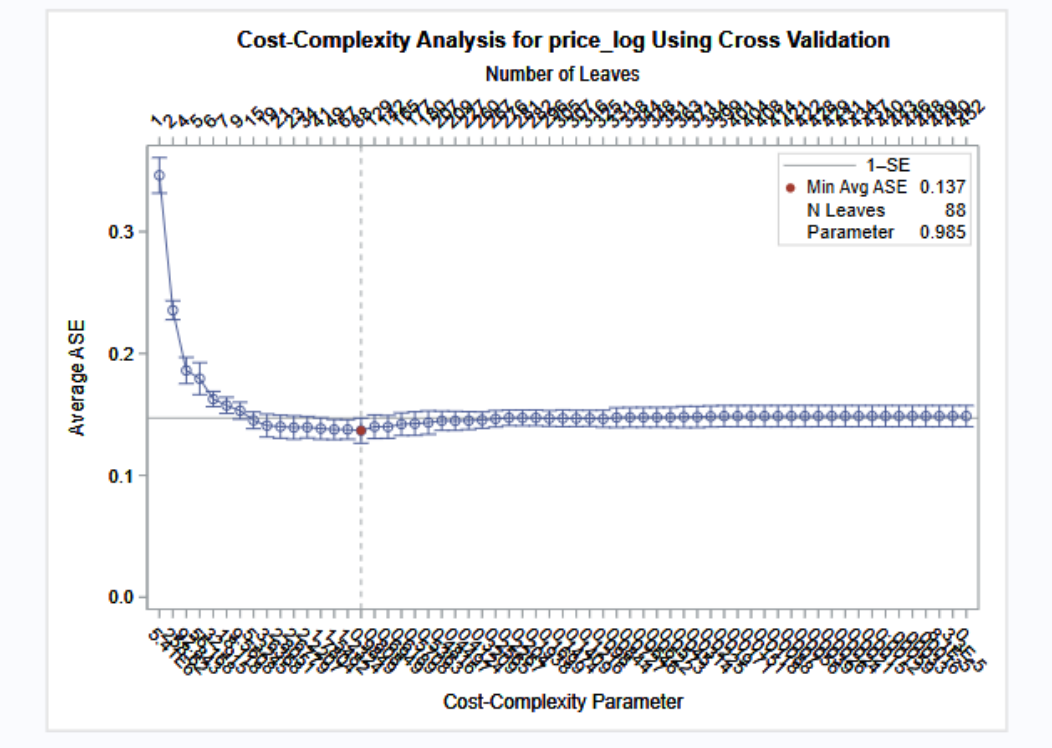


Figure 10 Cost Complexity Analysis

Regression Tree

This regression tree below shows the model structures of the data according to the key variables influencing Airbnb prices. The primary variations in listing pricing are captured by the top splits, such as accommodates and property_group. Other factors like neighborhood_group and bed_cat help to improve pricing predictions as the tree becomes deeper. The predicted log (price) is displayed on the color scale on the right; greater values are indicated by darker colors. In general, the highest predicted prices are seen in listings that are apartments, can accommodate more people, and are situated in the downtown area.

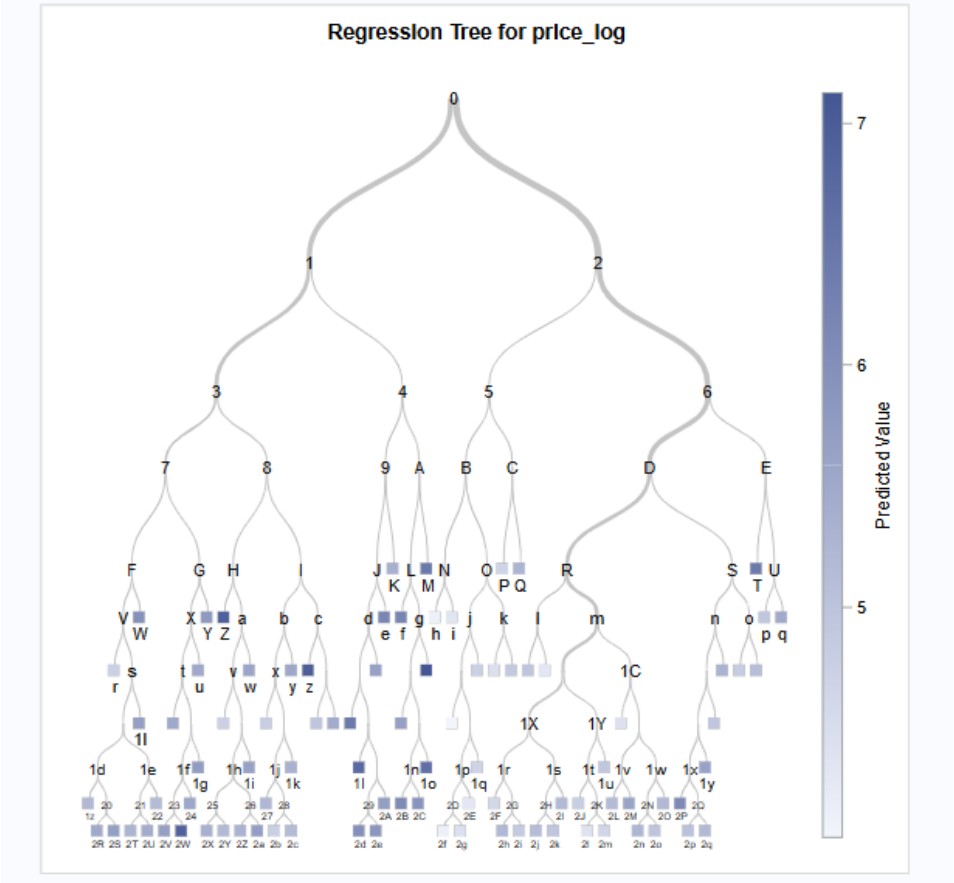


Figure 11 Regression Tree

Subtree

The subtree shows how the Decision Tree begins splitting data to predict Airbnb prices. The number of bedrooms is the most significant variable, as illustrated by the first split on bed_cat. The average log (price) values of listings with (0–1) bedrooms are often lower than those with (1–2) or more bedrooms. Listings are further divided according to guest capacity, property type, and location by the subsequent divides on accommodates, property_group, and neighborhood_group. Overall, the tree indicates that the most predicted prices are typically found in larger properties with more bedrooms, more guest capacity, and central locations.

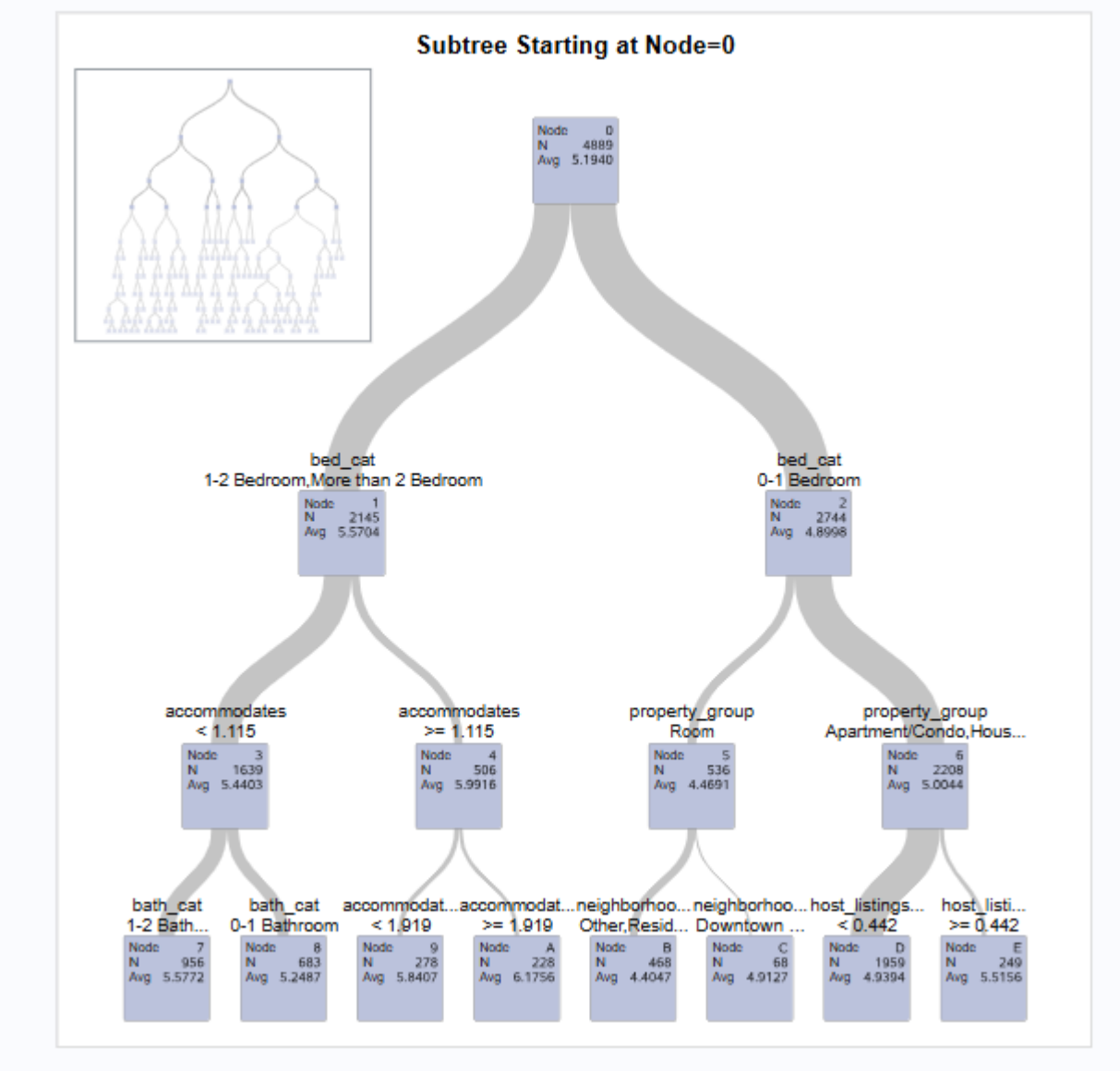


Figure 12 Subtree

Variable Importance for Decision Tree Model

The variable importance for the Decision Tree Model is shown below. The relative significance of each variable in the Decision Tree model prediction of Airbnb prices is displayed in this table. Bed_cat (23.29) is the most significant variable, indicating that the number of bedrooms has the biggest impact on price segmentation. Accommodates (13.57) and property_group (11.36) suggest that guest capacity and property type are also significant factors influencing pricing variations. Other factors like host_listings_count, bath_cat, and availability_365, have moderate importance. In comparison to property size, capacity, and location, the room group and instant group variables are of lower value, indicating a smaller impact in pricing prediction.

Table 18 Variable Importance of Decision Tree Model

Variable Importance				
Variable	Variable Label	Training		Count
		Relative	Importance	
bed_cat		1.0000	23.2931	2
accommodates	accommodates	0.5828	13.5760	13
property_group		0.4876	11.3576	7
host_listings_count	host_listings_count	0.4170	9.7135	10
bath_cat		0.3167	7.3780	7
availability_365	availability_365	0.3005	6.9999	23
neighborhood_group		0.2954	6.8809	6
minimum_nights	minimum_nights	0.2063	4.8053	6
maximum_nights	maximum_nights	0.1552	3.6139	3
beds_cat		0.1388	3.2338	4
room_group		0.1113	2.5916	5
instant_group		0.1095	2.5510	2

RMSE Summary of Decision Tree Model

From the table we can see the model performance of Decision Tree on both the training and test datasets. With just a slight difference between the two values, the model's RMSE of (0.3298) on the training set and (0.3675) on the test set demonstrated good prediction accuracy. The model correctly predicts actual price values, as shown by the true RMSE values (93.03) for training and (96.95) for testing. The Decision Tree shows good generalization and no overfitting because the RMSE values are similar for both datasets.

Decision Tree - RMSE Summary

model	dataset	rmse	true_rmse
Decision Tree	Training	0.3298	93.0302
Decision Tree	Test	0.3575	96.9516

Table 19 RMSE Summary of Decision Tree

Random Forest Model

To find patterns and connections between predictors and the log-transformed price, the Random Forest model was trained with the training dataset. The model's performance on untested data was assessed by applying it to the test dataset after training.

Model Information		
Parameter	Value	
Variables to Try	7	
Maximum Trees	300	
Actual Trees	300	
Inbag Fraction	0.6	(Default)
Prune Fraction	0	(Default)
Prune Threshold	0.1	(Default)
Leaf Fraction	0.00001	(Default)
Leaf Size Setting	10	
Leaf Size Used	10	
Category Bins	30	(Default)
Interval Bins	100	
Minimum Category Size	5	(Default)
Node Size	100000	(Default)
Maximum Depth	15	
Alpha	1	(Default)
Exhaustive	5000	(Default)
Rows of Sequence to Skip	5	(Default)
Split Criterion	.	Variance
Preselection Method	.	BinnedSearch
Missing Value Handling	.	Valid value

Table 20 Random Forest Output Table

The Random Forest output table contains a summary of the hyperparameter settings for the model. It captured the intricate correlations in the data as it was trained on 300 trees with a maximum depth of 15. The model assessed 7 factors at random at each split, increasing prediction accuracy through diversity and randomization. To prevent overfitting, each tree was constructed using 60% of the data (inbag fraction = 0.6). Each branch's level of intricacy was controlled by setting the leaf size to 10. The model handled missing values using a valid value technique and split using the variance criteria.

Table 21 Fit Statistic Table

Number of Trees	**Average Square Error (Train)**	**Average Square Error (OOB)**
50	0.162	0.154
100	0.140	0.128
150	0.121	0.117
200	0.117	0.116
250	0.116	0.116
300	0.116	0.116

From the fit statistics table, the model error rate (ASE) drops and eventually stabilizes as the number of trees increases. After about 250 trees, the error levels are eliminated, and the validation of ASE reaches (0.116), signifying that the model reached optimal accuracy. This pattern demonstrates that the Random Forest model is properly adjusted and does not overfit the data.

Variable Importance for Random Forest Model

Table 22 Loss Reduction Variable Importance

Loss Reduction Variable Importance

Variable	Number of Rules	MSE	OOB MSE	Absolute Error	OOB Absolute Error
bed_cat	940	0.070357	0.07000	0.055475	0.054761
accommodates	4635	0.059697	0.05580	0.052042	0.048404
property_group	2627	0.022460	0.02061	0.020592	0.018887
bath_cat	1988	0.015817	0.01497	0.015969	0.014947
host_listings_count	10737	0.021958	0.01423	0.027529	0.018996
beds_cat	1307	0.011176	0.01024	0.009658	0.008614
room_group	421	0.008416	0.00801	0.007261	0.006948
neighborhood_group	3172	0.009004	0.00679	0.010727	0.008325
minimum_nights	6687	0.008493	0.00345	0.011328	0.006192
instant_group	2334	0.002544	0.00107	0.003082	0.001508
availability_365	18814	0.013445	-0.00114	0.014775	-0.001819
maximum_nights	8871	0.004314	-0.00176	0.005182	-0.001777

From the table, we can see that bed_cat is the strongest predictor with the highest MSE (0.07) and OOB MSE (0.0700) indicating that the number of beds significantly influences the target variable. Accommodation seems to be the second highest predictor with MSE of (0.059) and OOB MSE of (0.054). The property group also plays a crucial role in predicting the Airbnb price as it shows the MSE of (0.024). Likewise, bath_cat ,host_listings_count, beds_cat show moderate importance with MSE of (0.018), (0.021), (0.011) respectively. Similar room_group, neighborhood_group, minimum_nights, and instant_group have much smaller MSE values, suggesting they have minimal influence.

RMSE Summary of Random Forest Model

Table 23 RMSE Summary of Random Forest

Random Forest - RMSE Summary

model	dataset	rmse	true_rmse
Random Forest	Training	0.3026	91.6150
Random Forest	Test	0.3304	94.2784

With an RMSE of (0.3028) on the training set and (0.3304) on the test set, the model demonstrated good consistency. The model works well when predicting actual price values, as demonstrated by the true RMSE values (91.82) for training and (94.28) for tests. The Random Forest is the most accurate model among those examined since it generalizes well and shows no evidence of overfitting due to a small variance between training and test errors.

Model Performance Comparison

Model Performance Comparison - All Models

model	dataset	rmse	true_rmse
LASSO Regression	Training	0.3694	104.5693
LASSO Regression	Test	0.3700	100.8395
Decision Tree	Training	0.3298	93.0302
Decision Tree	Test	0.3575	96.9516
Random Forest	Training	0.3026	91.6150
Random Forest	Test	0.3304	94.2784

For comparing the models, RMSE is used as it provides how well the model's predictions match actual price. The lower the RMSE the better the model. When comparing all three models, it shows that the Random Forest has the best predictive power with the lowest RMSE on both training (0.30) and test (0.33) data sets. The minimal difference between the training and test RMSE shows that it generalizes well without over fitting. Its ensemble method improves stability and lowers variance by averaging several trees.

The LASSO regression model shows the RMSE of (0.38) on the training set and (0.37) on the test set. This suggests that the model shows minimal over fitting, strong generalization, and consistent performance across training and test datasets. However, LASSO finds it difficult to adjust for complex, non-linear interactions among features that affect Airbnb prices because it mainly captures linear relationships. Due to this, its overall predictive accuracy is lower than that of tree-based models, which are more appropriate for managing non-linear patterns and variable interactions.

Likewise, the decision tree model shows the RMSE of (0.32) on the training set and (0.35) on the test set. This demonstrates that the model outperforms LASSO Regression in terms of accuracy by successfully capturing non-linear correlations between predictors and prices. The model works exceptionally well on the training data but loses some prediction accuracy when applied to unseen listings, given to the slightly higher RMSE on the test set.

Recommendation and Future Implications

The Random Forest model is recommended as the best model for predicting Airbnb prices in Seattle. As compared to other predictive model, Random Forest showed good generalization ability and the lowest test RMSE (0.3304). Models like LASSO are unable to adequately capture the complex, non-linear interactions between predictors like guest capacity, property type, and location given its ensemble structure. By offering hosts data-driven pricing recommendations, this approach can assist Airbnb in optimizing its dynamic pricing strategy from a business perspective Airbnb can increase host revenue and customer satisfaction by incorporating a model into the platform and recommending reasonable yet competitive pricing for both new and existing listings. In order to improve predicted accuracy, the model can be improved for future work by adding more factors like customer reviews, seasonal demand, and proximity to tourist attractions. Additionally, by expanding this framework to additional locations, Airbnb may be able to recognize local pricing patterns and create a scalable global pricing recommendation system.

Appendix

Summary Statistics for Numeric Variables

Summary Statistics for Numeric Variables

The MEANS Procedure

Variable	Label	N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
scrape_id	scrape_id	6112	0	2.0250621E13	2.0250621E13	2.0250621E13	2.0250621E13	0
last_scraped	last_scraped	6112	0	23913.00	23922.00	23913.77	23914.00	1.3609297
host_id	host_id	6112	0	2536.00	701082960	168915725	87918213.50	186787505
host_since	host_since	5888	224	17770.00	23905.00	20906.89	20687.50	1298.40
host_listings_count	host_listings_count	5888	224	1.0000000	4889.00	300.7026155	3.0000000	1111.54
host_total_listings_count	host_total_listings_count	5888	224	1.0000000	6586.00	367.0504416	5.0000000	1308.28
longitude	longitude	6112	0	-122.4194780	-122.2408700	-122.3350668	-122.3330067	0.0332479
accommodates	accommodates	6112	0	1.0000000	16.0000000	4.0924411	4.0000000	2.6088094
bathrooms	bathrooms	6112	0	0	16.0000000	1.4490347	1.0000000	0.8028926
bedrooms	bedrooms	6105	7	0	17.0000000	1.6552007	1.0000000	1.2112403
beds	beds	6111	1	0	15.0000000	2.1903126	2.0000000	1.5332467
price	price	6112	0	44.0000000	1295.00	216.0062173	177.0000000	147.3967169
minimum_nights	minimum_nights	6112	0	1.0000000	365.0000000	11.0620092	2.0000000	18.6175519
maximum_nights	maximum_nights	6112	0	1.0000000	10000.00	403.2814136	365.0000000	437.5584186
minimum_minimum_nights	minimum_minimum_nights	6112	0	1.0000000	365.0000000	10.1485602	2.0000000	17.5037930
maximum_minimum_nights	maximum_minimum_nights	6112	0	1.0000000	365.0000000	27.6982984	3.0000000	73.5378031
minimum_maximum_nights	minimum_maximum_nights	6112	0	1.0000000	10000.00	666.8121728	730.0000000	489.6925218
maximum_maximum_nights	maximum_maximum_nights	6112	0	2.0000000	10000.00	699.1472513	1125.00	481.3617598
minimum_nights_avg_ntm	minimum_nights_avg_ntm	6112	0	1.0000000	365.0000000	11.6774215	2.9000000	18.4906043
maximum_nights_avg_ntm	maximum_nights_avg_ntm	6112	0	2.0000000	10000.00	680.5125327	1084.10	481.7990229
availability_30	availability_30	6112	0	0	30.0000000	7.9136126	6.0000000	8.2585093
availability_60	availability_60	6112	0	0	60.0000000	21.4733312	19.0000000	17.1856252
availability_90	availability_90	6112	0	0	90.0000000	41.7177683	41.0000000	24.9002794
availability_365	availability_365	6112	0	0	365.0000000	211.8136453	230.0000000	107.4964174
calendar_last_scraped	calendar_last_scraped	6112	0	23913.00	23922.00	23913.77	23914.00	1.3609297
number_of_reviews	number_of_reviews	6112	0	0	1543.00	83.4478076	34.0000000	125.9041445
number_of_reviews_ltm	number_of_reviews_ltm	6112	0	0	682.0000000	18.5644634	11.0000000	24.3032329
number_of_reviews_l30d	number_of_reviews_l30d	6112	0	0	19.0000000	1.6071662	1.0000000	2.0835834
availability_eoy	availability_eoy	6112	0	0	194.0000000	115.7045157	129.0000000	52.4240735
number_of_reviews_ly	number_of_reviews_ly	6112	0	0	704.0000000	18.2066427	8.0000000	25.4064328
estimated_occupancy_l365d	estimated_occupancy_l365d	6112	0	0	255.0000000	112.9160668	102.0000000	95.2485985
estimated_revenue_l365d	estimated_revenue_l365d	6112	0	0	270300.00	22689.08	15660.00	25653.46
first_review	first_review	5346	766	18095.00	23912.00	22578.09	22934.00	1153.58
last_review	last_review	5346	766	20154.00	23921.00	23778.56	23893.00	335.7557250
review_scores_rating	review_scores_rating	5346	766	1.0000000	5.0000000	4.8154209	4.9100000	0.3328379
review_scores_accuracy	review_scores_accuracy	5346	766	1.0000000	5.0000000	4.8464516	4.9300000	0.2922074
review_scores_cleanliness	review_scores_cleanliness	5346	766	1.0000000	5.0000000	4.8250486	4.9200000	0.3241857
review_scores_checkin	review_scores_checkin	5346	766	1.0000000	5.0000000	4.8838103	4.9600000	0.2501547
review_scores_communication	review_scores_communication	5346	766	1.0000000	5.0000000	4.8894669	4.9700000	0.2780400
review_scores_location	review_scores_location	5346	766	1.0000000	5.0000000	4.8405612	4.9100000	0.2592050
review_scores_value	review_scores_value	5346	766	1.0000000	5.0000000	4.7310587	4.8200000	0.3516129
calculated_host_listings_count	calculated_host_listings_count	6112	0	1.0000000	341.0000000	30.4828207	2.0000000	77.8227726
calculated_host_listings_count_e	calculated_host_listings_count_entire_homes	6112	0	0	341.0000000	27.3447317	2.0000000	77.1074845
calculated_host_listings_count_p	calculated_host_listings_count_private_rooms	6112	0	0	66.0000000	3.1245092	0	10.8135918
calculated_host_listings_count_s	calculated_host_listings_count_shared_rooms	6112	0	0	8.0000000	0.0135798	0	0.2854323
reviews_per_month	reviews_per_month	5346	766	0.0100000	64.9100000	2.2195006	1.8550000	2.1208078
price_log		6112	0	3.7841896	7.1662660	5.1959884	5.1761497	0.5873019

Correlation Matric for Numeric Variables

Correlation Matrix of Numeric Variables

The CORR Procedure

8 Variables:	accommodates minimum_nights maximum_nights availability_30 availability_60 availability_90 availability_365 host_listings_count

Simple Statistics

Variable	N	Mean	Std Dev	Sum	Minimum	Maximum	Label
accommodates	6112	4.09244	2.60881	25013	1.00000	16.00000	accommodates
minimum_nights	6112	11.06201	18.61755	67611	1.00000	365.00000	minimum_nights
maximum_nights	6112	403.28141	437.55842	2464856	1.00000	10000	maximum_nights
availability_30	6112	7.91361	8.25851	48368	0	30.00000	availability_30
availability_60	6112	21.47333	17.18563	131245	0	60.00000	availability_60
availability_90	6112	41.71777	24.90028	254979	0	90.00000	availability_90
availability_365	6112	211.81365	107.49642	1294605	0	365.00000	availability_365
host_listings_count	6112	300.70262	1091	1837894	1.00000	4889	host_listings_count

Pearson Correlation Coefficients, N = 6112
Prob > |r| under H0: Rho=0

	accommodates	minimum_nights	maximum_nights	availability_30	availability_60	availability_90	availability_365	host_listings_count
accommodates	1.00000	-0.15369 <.0001	-0.04979 <.0001	0.06211 <.0001	0.07032 <.0001	0.07056 <.0001	0.04141 0.0012	-0.16597 <.0001
minimum_nights	-0.15369 <.0001	1.00000	0.25500 <.0001	-0.05935 <.0001	-0.09052 <.0001	-0.07147 <.0001	0.14085 <.0001	0.24690 <.0001
maximum_nights	-0.04979 <.0001	0.25500 <.0001	1.00000	-0.09219 <.0001	-0.10326 <.0001	-0.07633 <.0001	0.21157 <.0001	0.40019 <.0001
availability_30	0.06211 <.0001	-0.05935 <.0001	-0.09219 <.0001	1.00000	0.90061 <.0001	0.78999 <.0001	0.21469 <.0001	-0.16686 <.0001
availability_60	0.07032 <.0001	-0.09052 <.0001	-0.10326 <.0001	0.90061 <.0001	1.00000	0.93340 <.0001	0.26827 <.0001	-0.18795 <.0001
availability_90	0.07056 <.0001	-0.07147 <.0001	-0.07633 <.0001	0.78999 <.0001	0.93340 <.0001	1.00000	0.36418 <.0001	-0.17614 <.0001
availability_365	0.04141 0.0012	0.14085 <.0001	0.21157 <.0001	0.21469 <.0001	0.26827 <.0001	0.36418 <.0001	1.00000	0.09626 <.0001
host_listings_count	-0.16597 <.0001	0.24690 <.0001	0.40019 <.0001	-0.16686 <.0001	-0.18795 <.0001	-0.17614 <.0001	0.09626 <.0001	1.00000

Categorization of Categorical Variables

Categorization of Categorical Variables

The GLMSELECT Procedure

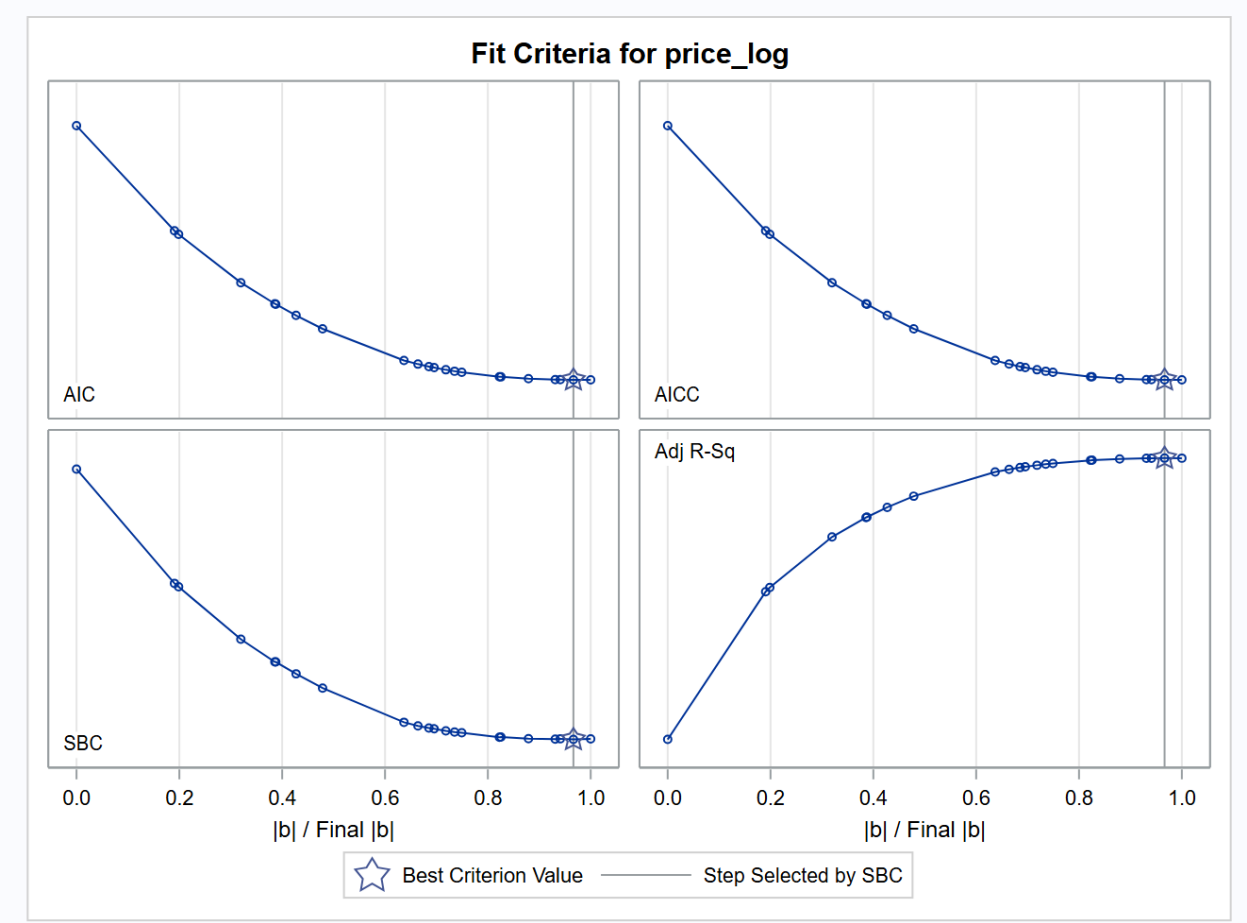
Data Set	WORK.TRAIN
Dependent Variable	price_log
Selection Method	LASSO
Stop Criterion	None
Choose Criterion	SBC
Effect Hierarchy Enforced	None

Number of Observations Read	4889
Number of Observations Used	4889

Class Level Information		
Class	**Levels**	**Values**
property_group	4	Apartment/Condo House Other Room
room_group	2	Entire Unit Room
neighborhood_group	4	Downtown & Tourist Hub Other Residential North Residential South & West
bath_cat	3	0-1 Bathroom 1-2 Bathroom More than 2 Bathroom
bed_cat	3	0-1 Bedroom 1-2 Bedroom More than 2 Bedroom
beds_cat	3	0-1 Bed 1-2 Beds More than 2 Beds
instant_group	2	0 1

Dimensions	
Number of Effects	13
Number of Effects after Splits	27
Number of Parameters	27

Fit Criteria



Training Set Performance Metrics

Obs	_TYPE_	_FREQ_	squared_error	trans_error	true_error	rmse	trans_rmse	true_rmse
1	0	4889	666.967	5265.86	72457.52	0.36935	1.03783	3.84974

Test Set Performance Metrics

Obs	_TYPE_	_FREQ_	squared_error	trans_error	true_error	rmse	trans_rmse	true_rmse
1	0	1223	167.452	1304.77	14292.41	0.37003	1.03289	3.41853

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
1	207	0.11713	0.14487
2	413	0.10570	0.14339
3	613	0.09949	0.13734
4	818	0.09768	0.13319
5	1022	0.09684	0.13091
6	1241	0.09582	0.12982
7	1442	0.09540	0.12853
8	1652	0.09475	0.12591
9	1863	0.09397	0.12445
10	2077	0.09403	0.12474
11	2279	0.09409	0.12347
12	2491	0.09347	0.12230
13	2699	0.09345	0.12204
14	2903	0.09331	0.12182
15	3109	0.09313	0.12144
16	3318	0.09310	0.12109
17	3535	0.09320	0.12077

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
18	3742	0.09313	0.12049
19	3950	0.09306	0.12020
20	4161	0.09286	0.12001
21	4368	0.09284	0.11991
22	4578	0.09282	0.11995
23	4782	0.09270	0.11976
24	4995	0.09267	0.11976
25	5210	0.09255	0.11963
26	5418	0.09246	0.11942
27	5628	0.09244	0.11929
28	5841	0.09231	0.11913
29	6047	0.09226	0.11906
30	6257	0.09229	0.11903
31	6470	0.09239	0.11905
32	6678	0.09238	0.11904
33	6879	0.09223	0.11882
34	7089	0.09217	0.11863

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
35	7305	0.09210	0.11851
36	7525	0.09220	0.11860
37	7739	0.09212	0.11861
38	7946	0.09210	0.11861
39	8152	0.09208	0.11851
40	8357	0.09198	0.11837
41	8573	0.09194	0.11818
42	8771	0.09198	0.11809
43	8982	0.09196	0.11794
44	9191	0.09194	0.11792
45	9395	0.09190	0.11786
46	9606	0.09189	0.11784
47	9818	0.09187	0.11786
48	10024	0.09185	0.11781
49	10233	0.09182	0.11777
50	10435	0.09182	0.11781
51	10637	0.09182	0.11772

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
52	10855	0.09181	0.11780
53	11059	0.09182	0.11773
54	11275	0.09176	0.11775
55	11486	0.09172	0.11766
56	11698	0.09174	0.11773
57	11918	0.09171	0.11768
58	12125	0.09172	0.11760
59	12340	0.09167	0.11756
60	12555	0.09167	0.11757
61	12762	0.09166	0.11752
62	12959	0.09170	0.11757
63	13165	0.09173	0.11758
64	13377	0.09171	0.11753
65	13575	0.09174	0.11752
66	13783	0.09174	0.11751
67	13983	0.09172	0.11747
68	14198	0.09174	0.11746

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
69	14410	0.09173	0.11744
70	14612	0.09176	0.11743
71	14832	0.09178	0.11746
72	15047	0.09173	0.11747
73	15251	0.09169	0.11739
74	15465	0.09166	0.11734
75	15674	0.09167	0.11732
76	15867	0.09168	0.11725
77	16082	0.09173	0.11728
78	16285	0.09175	0.11731
79	16501	0.09175	0.11730
80	16713	0.09179	0.11733
81	16923	0.09178	0.11733
82	17127	0.09176	0.11732
83	17345	0.09176	0.11730
84	17548	0.09177	0.11734
85	17768	0.09174	0.11732

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
86	17978	0.09172	0.11728
87	18191	0.09171	0.11722
88	18404	0.09171	0.11720
89	18619	0.09174	0.11724
90	18832	0.09174	0.11722
91	19050	0.09170	0.11717
92	19261	0.09174	0.11717
93	19466	0.09177	0.11719
94	19677	0.09179	0.11721
95	19896	0.09178	0.11722
96	20108	0.09178	0.11721
97	20327	0.09176	0.11715
98	20541	0.09176	0.11717
99	20753	0.09175	0.11719
100	20962	0.09174	0.11718
101	21173	0.09172	0.11718
102	21374	0.09173	0.11715

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
103	21585	0.09171	0.11713
104	21797	0.09171	0.11712
105	22006	0.09175	0.11718
106	22222	0.09172	0.11713
107	22437	0.09173	0.11714
108	22635	0.09176	0.11719
109	22850	0.09175	0.11719
110	23049	0.09177	0.11722
111	23243	0.09177	0.11719
112	23455	0.09177	0.11717
113	23661	0.09177	0.11714
114	23869	0.09174	0.11703
115	24080	0.09175	0.11706
116	24285	0.09179	0.11709
117	24488	0.09177	0.11709
118	24701	0.09177	0.11707
119	24918	0.09181	0.11714

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
120	25135	0.09180	0.11715
121	25340	0.09178	0.11712
122	25551	0.09177	0.11710
123	25758	0.09177	0.11709
124	25973	0.09177	0.11708
125	26176	0.09177	0.11707
126	26390	0.09177	0.11704
127	26592	0.09175	0.11701
128	26807	0.09176	0.11702
129	27008	0.09177	0.11701
130	27228	0.09178	0.11701
131	27432	0.09179	0.11701
132	27645	0.09179	0.11701
133	27848	0.09178	0.11698
134	28048	0.09177	0.11695
135	28263	0.09176	0.11693
136	28461	0.09177	0.11694

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
137	28664	0.09179	0.11692
138	28878	0.09176	0.11688
139	29089	0.09177	0.11688
140	29309	0.09176	0.11691
141	29516	0.09176	0.11692
142	29728	0.09176	0.11687
143	29932	0.09176	0.11687
144	30147	0.09176	0.11687
145	30367	0.09176	0.11689
146	30575	0.09175	0.11687
147	30783	0.09176	0.11688
148	30997	0.09176	0.11689
149	31210	0.09177	0.11690
150	31405	0.09177	0.11687
151	31617	0.09178	0.11688
152	31832	0.09177	0.11690
153	32041	0.09176	0.11688

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
154	32250	0.09174	0.11688
155	32453	0.09175	0.11686
156	32665	0.09177	0.11689
157	32870	0.09175	0.11682
158	33087	0.09175	0.11680
159	33300	0.09177	0.11683
160	33499	0.09178	0.11685
161	33707	0.09176	0.11683
162	33905	0.09176	0.11681
163	34116	0.09174	0.11678
164	34332	0.09174	0.11677
165	34533	0.09176	0.11678
166	34738	0.09176	0.11678
167	34943	0.09177	0.11678
168	35142	0.09177	0.11677
169	35355	0.09175	0.11674
170	35561	0.09174	0.11672

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
171	35761	0.09174	0.11671
172	35968	0.09173	0.11670
173	36173	0.09173	0.11671
174	36378	0.09172	0.11670
175	36591	0.09168	0.11667
176	36794	0.09170	0.11669
177	37010	0.09170	0.11668
178	37219	0.09173	0.11670
179	37434	0.09173	0.11670
180	37649	0.09172	0.11670
181	37864	0.09170	0.11668
182	38065	0.09170	0.11667
183	38286	0.09172	0.11666
184	38489	0.09169	0.11662
185	38703	0.09169	0.11661
186	38909	0.09168	0.11660
187	39113	0.09169	0.11659

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
188	39320	0.09167	0.11660
189	39533	0.09169	0.11661
190	39741	0.09169	0.11664
191	39951	0.09169	0.11666
192	40149	0.09170	0.11666
193	40361	0.09168	0.11662
194	40566	0.09167	0.11660
195	40778	0.09168	0.11660
196	40983	0.09167	0.11658
197	41190	0.09168	0.11661
198	41390	0.09167	0.11659
199	41598	0.09167	0.11660
200	41805	0.09167	0.11662
201	42020	0.09166	0.11663
202	42227	0.09165	0.11663
203	42435	0.09166	0.11665
204	42642	0.09167	0.11665

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
205	42843	0.09166	0.11661
206	43053	0.09165	0.11658
207	43262	0.09165	0.11657
208	43477	0.09164	0.11656
209	43689	0.09164	0.11656
210	43890	0.09164	0.11654
211	44096	0.09165	0.11654
212	44298	0.09165	0.11656
213	44505	0.09165	0.11654
214	44708	0.09166	0.11655
215	44924	0.09165	0.11652
216	45128	0.09164	0.11652
217	45342	0.09162	0.11650
218	45543	0.09164	0.11649
219	45746	0.09164	0.11649
220	45950	0.09163	0.11648
221	46167	0.09163	0.11649

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
222	46364	0.09163	0.11647
223	46584	0.09162	0.11646
224	46785	0.09163	0.11647
225	46986	0.09164	0.11648
226	47192	0.09164	0.11647
227	47405	0.09164	0.11647
228	47617	0.09163	0.11645
229	47833	0.09163	0.11647
230	48041	0.09161	0.11645
231	48250	0.09162	0.11644
232	48458	0.09162	0.11643
233	48663	0.09163	0.11643
234	48870	0.09162	0.11639
235	49077	0.09161	0.11638
236	49293	0.09161	0.11638
237	49510	0.09160	0.11637
238	49729	0.09161	0.11638

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
239	49938	0.09160	0.11634
240	50156	0.09160	0.11631
241	50365	0.09160	0.11632
242	50570	0.09161	0.11633
243	50778	0.09162	0.11633
244	50991	0.09161	0.11634
245	51215	0.09161	0.11633
246	51429	0.09161	0.11634
247	51646	0.09161	0.11635
248	51860	0.09160	0.11634
249	52065	0.09159	0.11634
250	52274	0.09159	0.11633
251	52485	0.09159	0.11633
252	52701	0.09158	0.11631
253	52909	0.09158	0.11630
254	53108	0.09159	0.11631
255	53319	0.09159	0.11631

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
256	53530	0.09160	0.11631
257	53733	0.09160	0.11631
258	53946	0.09159	0.11630
259	54159	0.09158	0.11630
260	54375	0.09159	0.11631
261	54597	0.09158	0.11629
262	54813	0.09159	0.11631
263	55019	0.09159	0.11631
264	55230	0.09158	0.11629
265	55442	0.09157	0.11631
266	55652	0.09157	0.11631
267	55874	0.09156	0.11629
268	56075	0.09157	0.11628
269	56276	0.09157	0.11629
270	56478	0.09157	0.11629
271	56685	0.09158	0.11630
272	56892	0.09159	0.11632

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
273	57108	0.09157	0.11631
274	57316	0.09156	0.11630
275	57527	0.09157	0.11632
276	57740	0.09157	0.11631
277	57936	0.09156	0.11629
278	58138	0.09157	0.11628
279	58350	0.09157	0.11626
280	58559	0.09157	0.11624
281	58774	0.09156	0.11623
282	58983	0.09156	0.11622
283	59196	0.09154	0.11622
284	59403	0.09155	0.11624
285	59620	0.09155	0.11624
286	59832	0.09155	0.11625
287	60050	0.09156	0.11625
288	60265	0.09156	0.11625
289	60475	0.09156	0.11624

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
290	60688	0.09155	0.11623
291	60891	0.09155	0.11621
292	61112	0.09155	0.11622
293	61331	0.09156	0.11621
294	61532	0.09155	0.11621
295	61751	0.09155	0.11622
296	61967	0.09155	0.11622
297	62180	0.09155	0.11621
298	62396	0.09155	0.11622
299	62613	0.09154	0.11623
300	62833	0.09154	0.11622