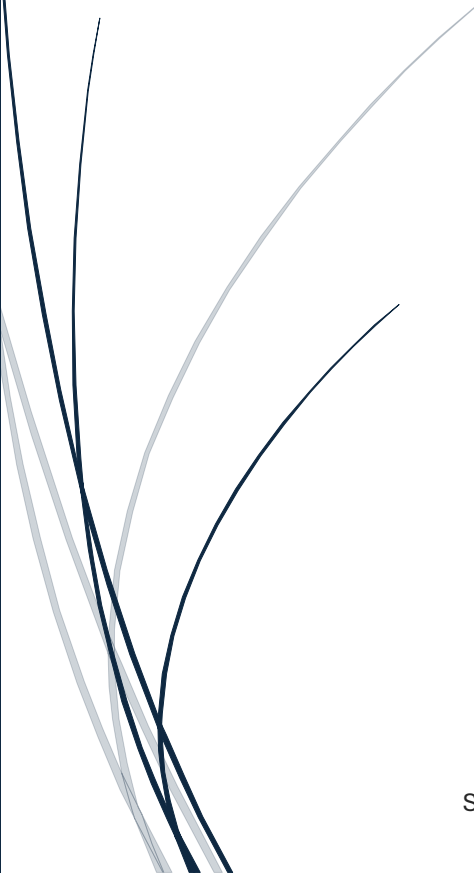


4/14/2025

Predicting Airbnb Listing Prices in New York City

A Comparative Analysis of Generalized Additive Models
and Gradient Boosting Machines



DUC ANH NGUYEN
UNIVERSITY OF CENTRAL FLORIDA
STA 5703 DATA MINING METHODOLOGY I

Table of Contents

1.	<i>Introduction</i>	2
1.1.	Problem Statement.....	2
1.2.	Background and Motivation	2
1.3.	Existing Approaches and Need for Further Analysis.....	2
2.	<i>Data Collection and Preparation</i>	3
2.1.	Data source.....	3
2.2.	Preprocessing and Transformations	3
2.2.1.	Handling Missing Values	3
2.2.2.	Transformation	3
2.2.3.	Variable Selection	4
3.	<i>Model Discussion</i>	4
3.1.	Justification for Model Selection	4
3.2.	Model Fit to Data Structure and Nature	5
3.3.	Discuss model assumptions and suitability	6
4.	<i>Methods</i>	7
4.1.	Generalized Additive Model (GAM)	7
4.2.	Gradient Boosting Machine (GBM)	7
5.	<i>Results of Data Analysis and Model Comparison</i>	9
5.1.	GAM Results (5-Fold Cross-Validation on Entire Dataset).....	9
5.2.	GBM Results (Training and Test Sets).....	10
5.3.	Compare and interpret model performance	13
6.	<i>Discussion</i>	13
6.1.	Reflection on Results	13
6.2.	Limitations and Assumptions	13
6.3.	Proposed Improvements and Future Work	14
7.	<i>References</i>	15
8.	<i>Appendices</i>	16

1. Introduction

1.1. Problem Statement

The problem addressed in this project is predicting the rental price of Airbnb listings in New York City using various listing attributes. Accurate price prediction helps hosts set competitive rates and assists guests in understanding price drivers. The focus is on building and comparing two predictive models to identify which best captures the relationship between rental price and features like location, room type, and availability.

1.2. Background and Motivation

Airbnb has transformed the short-term rental market by allowing property owners to list spaces directly to consumers. In a competitive market like New York City, rental prices vary widely based on location, property characteristics, and demand. Understanding the factors that influence pricing is valuable for both hosts aiming to optimize revenue and for platforms looking to offer pricing guidance. This project is motivated by the need to explore data-driven methods that can effectively model these pricing dynamics and improve decision-making for stakeholders.

1.3. Existing Approaches and Need for Further Analysis

Several methods have been applied to predict rental prices, ranging from simple linear regression to more complex machine learning techniques. Traditional methods like linear regression often struggle with capturing non-linear relationships between features, which are common in rental price dynamics. Recent approaches have incorporated tree-based methods, such as random forests and boosting, which can model these complex, non-linear relationships. Generalized Additive Models (GAMs) are another promising approach as they allow for flexibility in modeling non-linear effects while maintaining interpretability.

However, while these methods have shown success, the performance of boosting and GAM in the context of Airbnb rental price prediction has not been extensively compared. Further analysis is needed to assess the effectiveness of these models when applied to the unique characteristics of New York City's rental market, especially with the inclusion of various features like location, property attributes, and availability. This project aims to fill this gap by comparing the predictive

performance of GAM and boosting, providing insights into which method better captures the complexities of rental pricing.

2. Data Collection and Preparation

2.1. Data source

The dataset used for this project is the New York City Airbnb Open Data from Kaggle, which is publicly available and widely used for analyzing Airbnb listings. The data provides a comprehensive collection of listings in New York City, including details about the properties, their locations, and various other features such as price, reviews, and availability. This dataset was originally compiled to help analyze the short-term rental market in New York City, making it an ideal choice for the project's goal of predicting rental prices based on these attributes.

2.2. Preprocessing and Transformations

2.2.1. Handling Missing Values

The `reviews_per_month` column had missing values for listings that had no reviews. These missing values were imputed by replacing them with the mean value of the column, calculated from the available data.

2.2.2. Transformation

The target variable, `log_price`, was derived by applying a logarithmic transformation to the price variable using the formula: $\log(\text{price} + 1)$. This transformation was applied to address the right-skewed distribution of rental prices, stabilize variance, and improve the performance of models that assume normally distributed residuals. Adding 1 ensures that listings with a price of 0 do not result in undefined values during the transformation.

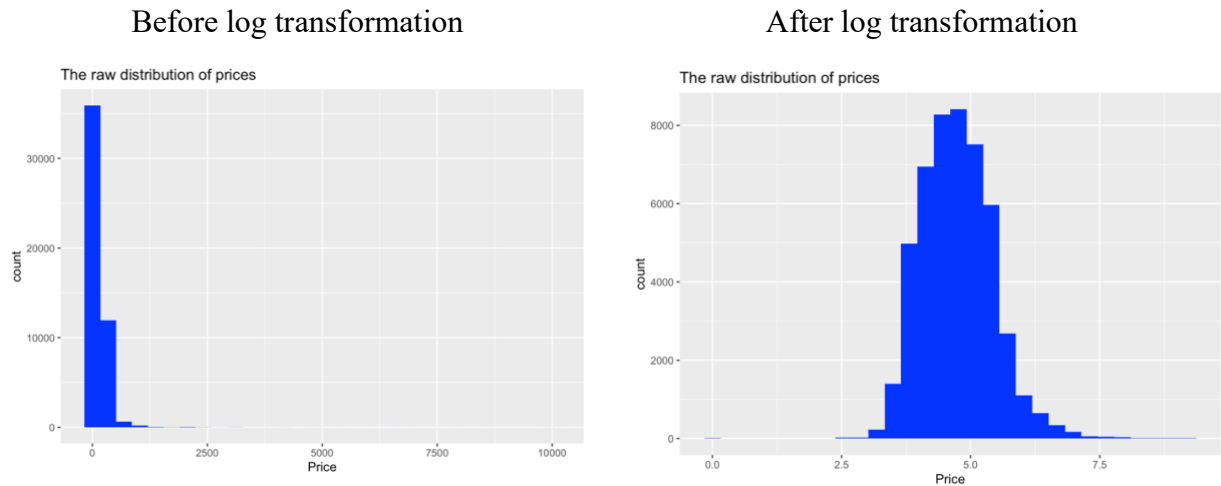


Figure 1: Log transformation on Price

2.2.3. Variable Selection

To improve model performance and reduce noise, a subset of variables was removed from the dataset. The columns `id`, `host_name`, `last_review`, and `name` were excluded, as they either provided little predictive value or were identifiers with no meaningful relationship to rental price.

3. Model Discussion

3.1. Justification for Model Selection

This project employs two distinct modeling approaches: Generalized Additive Models (GAM) and Gradient Boosting Machines (GBM). These models were selected due to their complementary strengths in capturing complex relationships between predictors and the target variable (`log_price`).

- GAM (Generalized Additive Model):

GAMs allow for flexible, non-linear relationships between predictors and the outcome by using smoothing functions. This is particularly useful for modeling continuous variables such as latitude, longitude, `minimum_nights`, and `availability_365`, which may not have a strictly linear effect on rental price. GAMs are interpretable and make fewer assumptions about the functional form of the

data, making them a strong candidate for understanding how each feature individually influences price.

- Gradient Boosting (GBM):

Boosting is a powerful ensemble technique that sequentially fits decision trees to minimize residual errors. GBM captures complex interactions between features and can handle non-linearities and heteroskedasticity effectively. It generally offers high predictive accuracy and is robust to outliers and irrelevant features due to its iterative refinement process. This model is well-suited for maximizing predictive performance on real-world data with noise and interactions, like Airbnb listings.

Using both models allows for a comparison between interpretability (GAM) and predictive power (GBM), helping to identify which modeling approach better fits the structure and complexity of Airbnb pricing data in New York City.

3.2. Model Fit to Data Structure and Nature

The structure of the dataset includes both **numerical predictors** (e.g., latitude, longitude, minimum_nights, availability_365) and **categorical features** (e.g., neighbourhood_group, room_type). Additionally, some relationships between predictors and the rental price are expected to be **nonlinear** (e.g., the effect of location or number of reviews on price may not be strictly linear).

To address these characteristics:

- Generalized Additive Model (GAM) was selected because it allows for nonlinear relationships through smooth functions (splines) while still maintaining interpretability. This makes it well-suited for capturing subtle variations in price influenced by continuous variables like location and availability without imposing strict parametric forms.
- Gradient Boosting (GBM) was chosen as a powerful ensemble tree-based method capable of automatically handling interactions and nonlinearities. GBM is robust to outliers and does not require explicit transformation or feature scaling, making it a good choice for capturing complex relationships in the data and improving predictive accuracy.

By using both models, we aim to balance interpretability (GAM) and predictive power (GBM) while accounting for the mixed data types and complex dynamics present in the dataset.

3.3. Discuss model assumptions and suitability

Generalized Additive Model (GAM) assumes that the relationship between the predictors and the response variable can be represented as a sum of smooth functions. Key assumptions include:

- **Additivity and smoothness:** Each predictor contributes additively and smoothly to the response.
- **Independence and homoscedasticity** of residuals.
- **Normal distribution of residuals**, especially important when using GAM for inference.

GAM is particularly suitable for our data because several predictors (like latitude, longitude, minimum_nights, and availability_365) are **continuous and likely to influence rental price in nonlinear ways**. GAM provides flexibility in capturing these effects while maintaining interpretability.

Gradient Boosting Machines (GBM) make fewer explicit assumptions about the data distribution. Instead, GBM builds an ensemble of decision trees sequentially to minimize the loss function. It is:

- **Nonparametric:** does not require linearity or specific distributions.
- **Robust to outliers** and can automatically detect interactions and non-linear effects.
- **Prone to overfitting** if not properly tuned (e.g., number of trees, depth, learning rate), but cross-validation can mitigate this.

GBM is highly suitable for this problem because of its ability to **model complex, high-order interactions** and **maximize prediction accuracy**, especially with mixed data types and nonlinear behavior.

4. Methods

This study compares two predictive models for estimating Airbnb rental prices in New York City: a **Generalized Additive Model (GAM)** and a **Gradient Boosting Machine (GBM)**. Both models are well-suited for regression tasks and allow for flexible modeling of complex relationships between predictors and the target variable. Each model was evaluated using **5-fold cross-validation** to ensure reliable performance estimates and reduce the risk of overfitting.

4.1. Generalized Additive Model (GAM)

We used the `mgcv` package in R to fit the GAM model. The dependent variable is `log_price`, derived by applying a log transformation to the original price variable ($\log(\text{price} + 1)$) to address right-skewness and stabilize variance. Smooth terms (`s()`) were applied to the continuous predictors, while categorical variables were included linearly.

```
set.seed(123)
folds <- vfold_cv(data, v = 5)

cv_results <- map_dfr(folds$splits, function(split) {
  train_split <- analysis(split)
  val_split <- assessment(split)

  gam_model <- gam(
    log_price ~ s(latitude) + s(longitude) + s(minimum_nights) +
      s(number_of_reviews) + s(reviews_per_month) +
      s(calculated_host_listings_count) + s(availability_365) +
      neighbour_group + room_type,
    data = train_split
  )
  preds <- predict(gam_model, newdata = val_split)
```

Code snippet 1: GAM with 5 folds CV

4.2. Gradient Boosting Machine (GBM)

GBM was implemented using the `caret` package in R, with the `gbm` method. All the same predictors as in the GAM were included for a fair comparison.

The model was tuned using `tuneLength = 5`, allowing the algorithm to explore 5 different combinations of parameters. The categorical variables `neighbourhood_group` and `room_type` were included as factors, and handled internally by the model.

```
set.seed(123)
control <- trainControl(method = "cv", number = 5)
gbm_model <- train(
  log_price ~ latitude + longitude + minimum_nights + number_of_reviews +
    reviews_per_month + calculated_host_listings_count + availability_365 +
    neighbourhood_group + room_type,
  data = train_data,
  method = "gbm",
  trControl = control,
  tuneLength = 5,
  verbose = FALSE
)
print(gbm_model)
```

Code snippet 2: GBM with 5 folds CV

5. Results of Data Analysis and Model Comparison

5.1. GAM Results (5-Fold Cross-Validation on Entire Dataset)

```
Family: gaussian
Link function: identity

Formula:
log_price ~ s(latitude) + s(longitude) + s(minimum_nights) +
  s(number_of_reviews) + s(reviews_per_month) + s(calculated_host_listings_count) +
  s(availability_365) + neighbourhood_group + room_type

Parametric coefficients:

```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.055523	0.022415	225.546	< 2e-16 ***
neighbourhood_groupBrooklyn	0.024140	0.025901	0.932	0.351
neighbourhood_groupManhattan	0.103321	0.021795	4.741	2.14e-06 ***
neighbourhood_groupQueens	-0.104530	0.023197	-4.506	6.62e-06 ***
neighbourhood_groupStaten Island	-0.002979	0.121558	-0.025	0.980
room_typePrivate room	-0.735149	0.005125	-143.431	< 2e-16 ***
room_typeShared room	-1.163041	0.015941	-72.960	< 2e-16 ***

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:

```

	edf	Ref.df	F	p-value
s(latitude)	8.569	8.945	229.56	<2e-16 ***
s(longitude)	8.401	8.880	175.97	<2e-16 ***
s(minimum_nights)	8.905	8.996	137.98	<2e-16 ***
s(number_of_reviews)	6.108	7.044	57.65	<2e-16 ***
s(reviews_per_month)	7.540	8.342	28.97	<2e-16 ***
s(calculated_host_listings_count)	7.803	8.510	22.13	<2e-16 ***
s(availability_365)	8.831	8.991	297.42	<2e-16 ***

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.553   Deviance explained = 55.4%
GCV = 0.21677   Scale est. = 0.21642   n = 39117
```

Code snippet 3: GAM model

The GAM model uses smooth splines to capture the non-linear relationships between the continuous predictors (like latitude, longitude, and minimum_nights) and the target variable. The significant parametric coefficients for neighbourhood_group and room_type provide useful insights into the influence of categorical variables on rental prices. Specifically, properties in Manhattan tend to have higher rental prices compared to other neighbourhoods, and shared rooms have a more significant negative impact on the price than private rooms.

Metric	Value
RMSE (Average)	0.4651

Adjusted R² (Average)	0.5536
AIC (Average)	51093.13
BIC (Average)	51637.82

Table 1: GAM model metrics

The Generalized Additive Model (GAM) demonstrates moderate predictive performance. The average RMSE across the five folds is 0.4651, indicating that on average, predictions differ from actual values by that magnitude on the log scale. The Adjusted R² of 0.5536 suggests the model explains over half of the variance in log-transformed price. The AIC and BIC values are reasonably low, showing a good trade-off between model fit and complexity.

5.2. GBM Results (Training and Test Sets)

```

Stochastic Gradient Boosting

39117 samples
  9 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 31295, 31295, 31294, 31292, 31292
Resampling results across tuning parameters:

interaction.depth  n.trees  RMSE      Rsquared  MAE
1                  50      0.5057970  0.4913537  0.3636289
1                  100     0.4833654  0.5283927  0.3449918
1                  150     0.4738119  0.5416858  0.3375333
1                  200     0.4690479  0.5485371  0.3341971
1                  250     0.4666565  0.5517995  0.3325737
2                   50     0.4801048  0.5351175  0.3425768
2                  100     0.4643370  0.5582257  0.3307043
2                  150     0.4592904  0.5656327  0.3272888
2                  200     0.4569101  0.5695199  0.3258748
2                  250     0.4554581  0.5720984  0.3247969
3                   50     0.4707650  0.5497830  0.3354047
3                  100     0.4579274  0.5684302  0.3262670
3                  150     0.4544735  0.5740498  0.3238697
3                  200     0.4523963  0.5777319  0.3224183
3                  250     0.4512094  0.5798783  0.3214524
4                   50     0.4657537  0.5576253  0.3317610
4                  100     0.4549156  0.5737034  0.3242083
4                  150     0.4517824  0.5790069  0.3218254
4                  200     0.4501156  0.5819272  0.3205804
4                  250     0.4489459  0.5840324  0.3197992
5                   50     0.4620750  0.5632416  0.3291552
5                  100     0.4524728  0.5779501  0.3224659
5                  150     0.4497843  0.5826146  0.3202886
5                  200     0.4480907  0.5856444  0.3189426
5                  250     0.4467777  0.5880243  0.3179834

Tuning parameter 'shrinkage' was held constant at a value of 0.1
Tuning parameter 'n.minobsinnode' was held constant at a value of 10
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were n.trees = 250, interaction.depth = 5, shrinkage = 0.1 and n.minobsinnode = 10.

```

Code snippet 4: GBM model

The optimal model was chosen using RMSE to minimize predictive error. The following parameters were tuned during the model training:

- `n.trees`: Number of trees in the model (ranging from 50 to 250).
- `interaction.depth`: Maximum depth of the trees (ranging from 1 to 5).
- `shrinkage`: A regularization parameter to control the model's learning rate (held constant at 0.1).
- `n.minobsinnode`: Minimum number of observations in a terminal node (held constant at 10).

The optimal model was selected based on the smallest RMSE, with the final parameter values being:

- `n.trees` = 250
- `interaction.depth` = 5
- `shrinkage` = 0.1
- `n.minobsinnode` = 10

	var <chr>	rel.inf <dbl>
room_typePrivate room	room_typePrivate room	48.97834882
longitude	longitude	13.03437708
room_typeShared room	room_typeShared room	10.49956878
latitude	latitude	8.73159804
availability_365	availability_365	5.84517929
minimum_nights	minimum_nights	3.94520960
neighbourhood_groupManhattan	neighbourhood_groupManhattan	3.35662901
number_of_reviews	number_of_reviews	2.31646769
calculated_host_listings_count	calculated_host_listings_count	2.27427586
reviews_per_month	reviews_per_month	0.87305383

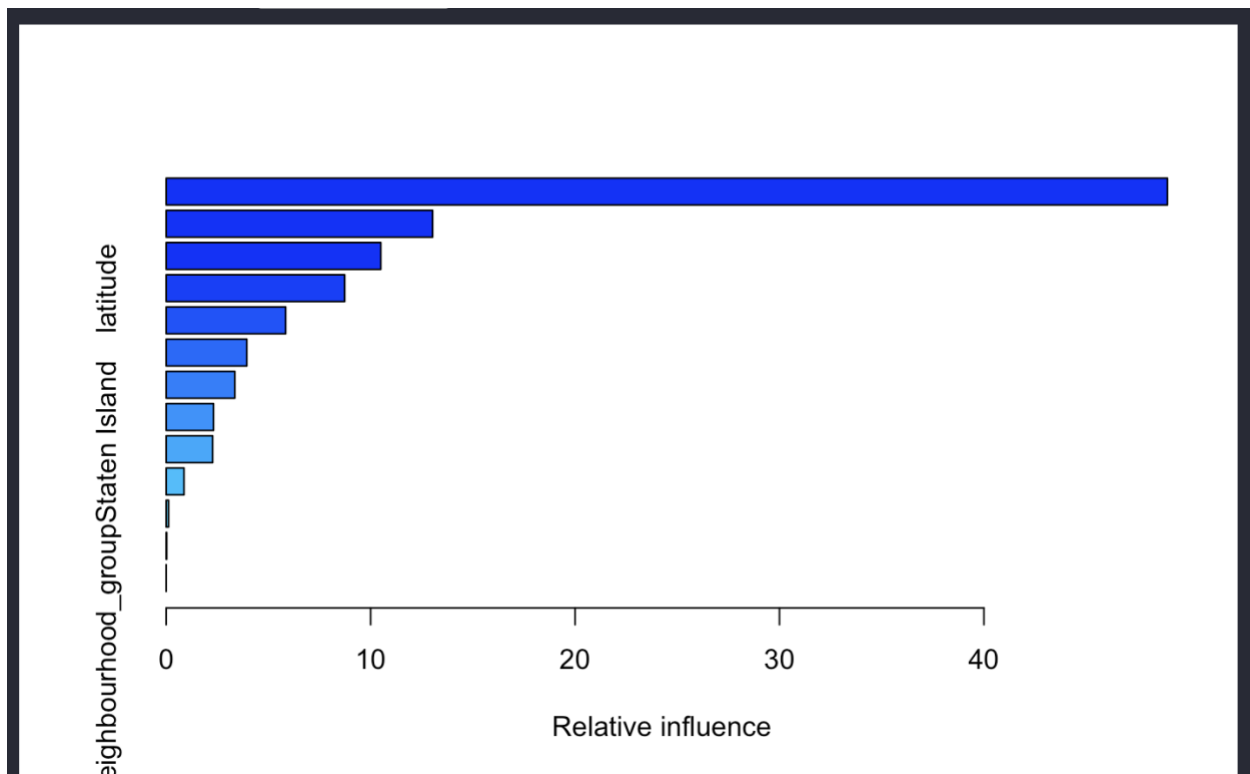


Figure 2: Code snippet 5: GBM Feature Importance

The most important features influencing rental prices in the GBM model are room type (both private and shared rooms) and geographical location (latitude and longitude). Private rooms and more central locations generally lead to higher prices.

On the other hand, neighborhood features (Brooklyn, Queens, and Staten Island) have minimal impact on price, suggesting that geographic coordinates already capture much of the location-related price variation.

In summary, room type and location are the primary drivers of rental price predictions, while neighborhood-specific factors are less influential.

Metric	Training Set	Test Set
RMSE	0.4357	0.4446
Adjusted R ²	0.6080	0.5875

Table 2: GBM model metrics

Gradient Boosting Machine (GBM), trained using 5-fold cross-validation, outperforms the Generalized Additive Model (GAM) in terms of both RMSE and Adjusted R^2 . The test RMSE for GBM is 0.4446, which is lower than GAM's average RMSE, and the Adjusted R^2 on the test set is 0.5875, indicating a higher level of explanatory power. AIC and BIC are not applicable to GBM, as it is a non-parametric model.

5.3. Compare and interpret model performance

In comparing the performance of the GAM and GBM models, GBM outperforms GAM in terms of predictive accuracy. The GBM model achieved a test RMSE of 0.445 and an Adjusted R^2 of 0.587, while GAM had a test RMSE of 0.466 and an Adjusted R^2 of 0.552. GBM excels at capturing complex, non-linear interactions between variables, which is likely why it performs better with this dataset. In contrast, GAM, while more interpretable, struggles with flexibility in modeling intricate relationships. The superior performance of GBM can be attributed to its boosting technique, which allows it to iteratively improve predictions, making it better suited for this kind of data.

6. Discussion

6.1. Reflection on Results

The GAM model provides interpretability by showing how location and property attributes affect prices, but the GBM model outperforms in terms of RMSE and adjusted R^2 , suggesting it captures complex relationships better. However, GBM might be prone to overfitting, as its performance in training is much better than on the test set.

6.2. Limitations and Assumptions

- Data Quality: Missing or incorrect data, especially for features like reviews or availability, could have impacted the models.
- Interpretability: GAM is easier to interpret, while GBM, though more accurate, lacks transparency.
- Overfitting: GBM's strong training performance could suggest overfitting.

- Feature Selection: Some features like room type and neighborhood group were more important, but further feature engineering could improve results.

6.3. Proposed Improvements and Future Work

- Data Enhancement: Adding more granular features (e.g., amenities or local events) could improve model accuracy.
- Feature Engineering: Creating interaction terms or new derived features could boost performance.
- Model Refinement: Further tuning GBM could reduce overfitting and improve generalization.
- Ensemble Methods: Combining GAM and GBM predictions through ensemble techniques might enhance accuracy.
- Exploring Other Models: Testing models like Random Forest or XGBoost could provide additional insights and improvements.

7. References

1. Kaggle. (n.d.). *New York City Airbnb Open Data*. Retrieved from <https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data>
2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning with Applications in R* (2nd ed.). Springer.

8. Appendices

Table 1: GAM model metrics	10
----------------------------	----

Table 2: GBM model metrics	12
----------------------------	----

Figure 1: Log transformation on Price	4
---------------------------------------	---

Figure 2: Code snippet 5: GBM Feature Importance	12
--------------------------------------------------	----

Code snippet 1: GAM with 5 folds CV	7
-------------------------------------	---

Code snippet 2: GBM with 5 folds CV	8
-------------------------------------	---

Code snippet 3: GAM model	9
---------------------------	---

Code snippet 4: GBM model	10
---------------------------	----