Justification of Airbnb Listings: Use of Machine Learning to Assess Fairness of San Diego Airbnb Prices

Hunter Blum
Applied Data Science Master's Program
Shiley Marcos School of Engineering / University of San Diego
hblum@sandiego.edu

Kyle Esteban Dalope
Applied Data Science Master's Program
Shiley Marcos School of Engineering / University of San Diego
kdalope@sandiego.edu

Nicholas Lee
Applied Data Science Master's Program
Shiley Marcos School of Engineering / University of San Diego
nvlee@sandiego.edu

ABSTRACT

As California rental prices continue to rise, renters have struggled to evaluate the factors affecting prices and determine a fair rate. A machine-learning model assessing Airbnb price fairness can serve as an integral tool to empower and inform consumers. Such a tool is warranted since hosts currently have a surplus of machine-learning tools readily accessible. This study focused on San Diego, CA Airbnb data from March 2022 to March 2023. The study included an exploratory data analysis, cleaning and engineering features, and the creation of a multitude of rental price models. Separate models were created based on two listing property types: single room rentals and entire home rentals. For both property types, gradient boosting regressions provided an optimal mix of performance and accessible feature importance. The results of this study showed a sentiment-based feature of the text on the listing page, property size, bedrooms, bathrooms, and weighted review scores all played a large role in influencing the predictions of the model. Finally, the models were deployed to a web application (fairbnb.streamlit.app), giving consumers a tool to evaluate San Diego rental prices.

KEYWORDS

Airbnb, machine learning, gradient boosting regression, price prediction, feature selection, rental, San Diego.

1 Introduction

Homeownership and renting in California has become increasingly unaffordable. In the previous year, 2022, the yearly average inflation rate significantly increased from approximately 2% to 6.1%, an all-time high for this century (U.S. Bureau of Labor Statistics, 2023). Many Californians found the cost-of-living has increased dramatically, thus exacerbating issues related to affordable housing. One rental service in particular, Airbnb, found itself under scrutiny by Lee (2016) for worsening the affordability of homes in the southern-California region. Airbnb was originally designed to be a short-term rental service alternative to traditional commercial establishments like hotels. However, unlike traditional establishments, which adjust prices to remain competitive, Airbnb rental listings are subject to the discretion of the host. Thus, hosts may charge exorbitant rates without the consumer's knowledge of a fair rate. Although hosts have machine learning tools such as Airbnb's own Smart Pricing tool to help maximize rental profits, consumers lack tools to fairly assess rental prices. Therefore, to prevent consumers from mistakenly overpaying for rent, this study established a machine learning tool that predicts a fair market price for a rental property.

2 Background

Numerous studies have used real estate data to model home, rental, and Airbnb prices (Cheung & Yiu, 2022; Garcia-López et al., 2020; Pai et al., 2020). However, a limited number of these studies focus on consumers' best interests. Rather, most studies have established machine learning models that would generate rental prices to "[guarantee] a financial success" for property owners (Heidari et al., 2021).

2.1 **Problem Identification and Motivation**

With rental prices increasing and subject to the discretion of individual hosts, the consumer incurs a growing risk of overpaying for rent. Other than a visual evaluation, consumers currently lack the tools needed to make fully informed decisions about rental prices. If rentals are continually over-priced, several consequences will ensue. For example, as Albouy et al. (2016) illustrated, as the proportion of income spent on housing increases, the proportion of available funds for basic necessities will decrease, thereby decreasing quality of life for consumers. Accordingly, if prices begin to exceed the consumers' willingness to pay, consumers will begin taking their business elsewhere.

While the intentions of this research were to inform and empower consumers, the economic benefits for Airbnb and its property hosts cannot be denied. Short-term rental services have stimulated the economy in a positive manner by increasing and sustaining tourism (Casamatta et al., 2022). As a result, it would be detrimental to Airbnb, its hosts, and the economy if Airbnb lost business solely because hosts prioritized maximizing profits over fair prices. Therefore, it is in everyone's best interest that prices are fair and fully transparent.

2.2 **Definition of Objectives**

Using data spanning from March 2022 to March 2023, the key objective of this research was to establish a web application equipped with an optimized machine learning algorithm to reliably predict San Diego Airbnb prices, with ongoing updates for data integration to maintain the application's reliability. The primary aim is to provide consumers with a fully transparent and accessible application for understanding the listing price relative to market value. By offering insights on which features such as property size and location influence rental prices, the study promotes transparency in the San Diego rental market.

2.2.1 Expand Consumer Knowledge

With regards to machine learning, the public has a limited understanding of models' inner workings and how algorithms reach conclusions. Thus, one key goal of this study is to identify the best features for determining San Diego rental prices and present how they influence the predicted price. Therefore, model interpretability must also be prioritized. To achieve this goal, the tool must be comprehensible for a non-technical user. If successful, this goal would ultimately build trust with the user and clearly communicate how the application created its prediction, thus empowering the consumer to create their own conclusions.

2.2.2 Optimize Model Features

Given that there are many factors for determining rental prices, this study builds upon previous research and continues to identify these key features. Optimized features will improve the overall performance of pricing models. In addition, feature reduction decreases the time needed to train and deploy a model.

2.2.3 Empower Consumers

The third objective was to provide consumers with an accessible and reliable tool for evaluating fair rental rates. Consumers can use this application in a way that enriches their livelihood (e.g., allowing them to budget better by finding fairly listed rentals). In addition, with increased use of this tool, hosts may be more wary of charging unfair, high prices.

3 Literature Review

Although Airbnb is a contemporary technology, several scholars have attempted to find an optimal rental pricing model (Kalhebasti et al., 2021; Lektorov et al., 2023; Tang and Songani, 2015). Throughout these attempts, many solutions have been proposed, particularly in feature and model selection. Both topics have evolved in recent years; however, researchers have yet to reach a consensus on an optimal methodology for feature and model selection.

3.1 **Feature Selection**

3.1.1 Geographical Features

Location is often considered one of the most important factors in the price of an Airbnb rental. However, geographical impact can vary on the scale and location of the dataset. Dhillon et al. (2021) found latitude and longitude did not have a strong linear relationship with rental price, despite evaluating U.S. 28 cities. In an analysis of the Nashville metropolitan area, Zhang et al. (2017) discovered spatial heterogeneity in explanatory features and saw performance improvements using geographic weighted regression over a general linear model, suggesting that geography can be integral to a city-based model. Due to each city's unique spatial features, the effect of geographic coordinates or neighborhood predictors should be evaluated for individual cities.

3.1.2 Text-Based Features

Airbnb listings also include a large amount of text data in the form of host descriptions and user reviews. Kalehbasti et al. (2021) achieved an R^2 score of 0.69 when including a TextBlob sentiment analysis of user reviews in their model. Similarly, Tang and Sangani (2015), used TextBlob on host description fields to generate sentiment analysis features. However, since TextBlob is a lexicon-based approach it may not capture the true meaning of our text features, particularly for missing words and negation phrases, such as *not bad* (Albrecht et al., 2021). Therefore, a pre-trained language model may offer more accurate sentiment analysis features and improved model performance. Kalehbasti et al. (2021) also suggested weighting sentiment features based on their recency to improve their relevance.

3.1.3 Feature Reduction

As more features are added to models, over-fitting becomes a larger issue. The two previous studies both experienced a high variance of errors due to the large amount of both geographic and text features. Previous research has used manual, correlative, Lasso regularization, selecting the lowest p-values from linear regression, and recursive feature selection (Dhillon et al., 2021; Kalehbasti et al., 2021; Tang and Sangani, 2015). Evaluating other filter, wrapper, and embedding based techniques could help improve model performance and interpretability.

3.2 Modeling Considerations

Earlier research has evaluated many machine learning models. Two studies found random forest models to have the lowest errors (Dhillon et al., 2021; Lektorov et al., 2023). Kalehbasti et al. (2021) concluded support vector regression was the best suited model; additionally, Tang and Sangani (2015) achieved high classification accuracy using a support vector classifier. Due to differing regions studied and different optimal models in prior studies, this study will evaluate multiple machine learning models.

3.2.1 Need for Multiple Models

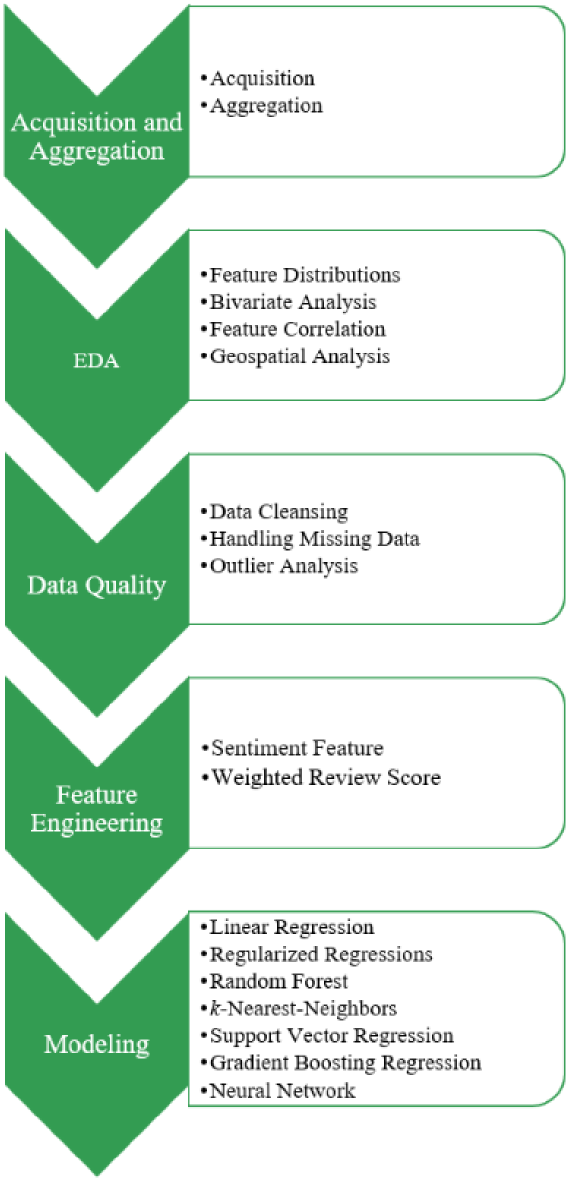
The room type of an Airbnb also has a large impact on price. One would expect an entire house to cost more than a single room in a non-private apartment. Voltes-Dorta & Sanchez-Medina (2020) concluded separating pricing models based on both property and room type is better than including them as dummy variables in a single model. Therefore, this project will create separate models for homes and single rooms.

4 Methodology

The core objective of the methodology section was to predict the nightly price of Airbnb rentals in San Diego, CA. This section follows five main steps: *Acquisition and Aggregation, Exploratory Data Analysis (EDA) Data Quality, Feature Engineering,* and *Modeling.* Each section incorporates a variety of subsections, exhibited in *Figure 1.*

Figure 1

Study Methodology Steps



4.1 Data Acquisition and Aggregation

4.1.1 Data Acquisition

The data for this study was acquired from Inside Airbnb, a project hosting Airbnb data for many cities. Four CSV files of Airbnb listings in San Diego were obtained, one for each yearly quarter from March, 2022 to March, 2023. To supplement the data, 2021 median income data based on local zip codes were obtained from the U.S. Census Bureau (2023).

4.1.2 Data Aggregation

The four CSV files contained data on all listings that existed during the quarterly scrape. Thus, if a listing spanned multiple quarters, the listing would be present in multiple files. To keep the most up-to-date information, the most recent observation from each unique listing identification number was retained.

After combination, the *zipcode* feature was created based on each listing's latitude and longitude. The *median_income* feature was joined from the Census data, using the *zipcode* features in each dataset.

4.2 Exploratory Data Analysis

The four quarterly data sets had a combined 18,627 unique listings and 75 variables, including the target feature *price*. To optimize the EDA process, features deemed redundant or unnecessary were dropped. Many of the dropped features related to non-functioning hyperlinks, such as *listing_url* and *host_url*, arbitrary identification numbers (*id, scrape_id,* and *host_id*), and features related to images, such as *picture_url* and *host_picture_url*.

With regards to data sensitivity concerns, the data is readily available to the public via Inside Airbnb (which is scraped from Airbnb, a public listing site) and the U.S. Census. For example, features, such as *host_name*, *host_location*, and *host_picture*, while personally identifiable, are disclosed at the discretion of the hosts. In addition, features that contain personal identifiable information were removed prior to model training, as they likely do not impact rental price. Thus, there is little concern that the data violates data privacy issues.

4.2.1 Feature Distributions

Fisher-Pearson coefficient of skewness and Fisher's kurtosis were calculated to evaluate the distribution of each feature. The target feature, *price,* was heavily skewed to the right, with a skewness value of 50. *Price* had a kurtosis of 4,445 indicating a very strong peak with a long tail of positive outliers

Of the 36 numeric predictors initially in the dataset, 27 had skews outside the interval [-1, 1]. The high skews of most features suggests that non-parmetric models should be explored.

4.2.2 Bivariate Analysis

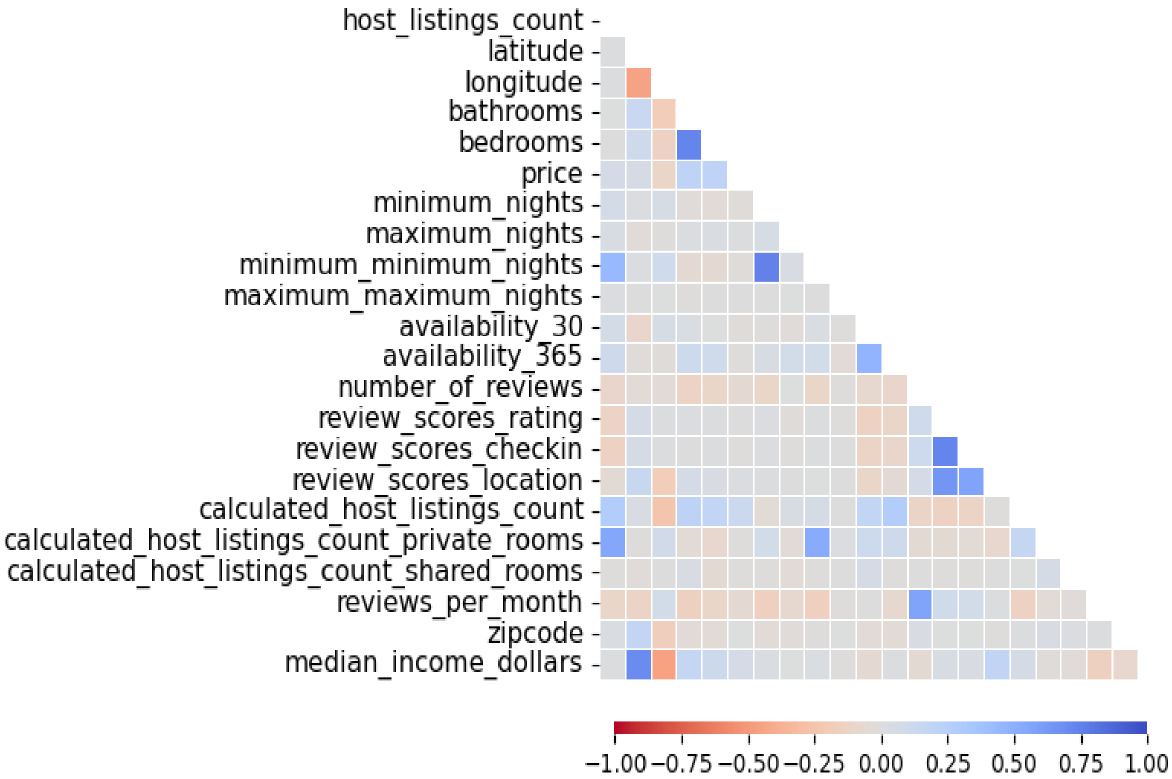
Scatterplots were made to assess the relationship between each numerical predictor to the outcome feature of interest, "price" (see Appendix Figure 6). These plots did not show any linear relationship between the numeric features and the target feature. However, the scatterplots showed that when a listing contains more 5- or 4-star reviews, the listing typically has a higher price. More interestingly, the fewer reviews a listing has received per month, the more expensive it is, suggesting that customers that pay more tend to write less reviews per month. Because these trends between price and review-based features held true across all review-based features, many of the review-based features, such as *review_scores_cleanliness*, were deemed redundant.

4.2.3 Feature Correlation

Pearson's correlation coefficient was calculated for every pairwise combination of features. A heatmap of these correlation coefficients revealed that many pairs of features were positively correlated (Figure 2). The correlation coefficients were filtered using an absolute threshold of 0.75. This revealed that 24 feature pairs had a correlation coefficient either greater than 0.75 or less than -0.75. Many of these correlated feature pairs contained features with redundant meanings, such as the feature pair *availability_60* and *availability_90* and feature pairs between review-based features. To reduce multicollinearity, one feature from each of the 24 feature pairs was selected for removal. However, because some features appeared multiple times through this filtering method, only 16 features were removed.

Figure 2

Pearson's Correlation Coefficient Heatmap



4.2.4 Geospatial Feature Analysis (Choropleths)

Choropleths were created to visualize descriptive statistics of features with relation to geospatial characteristics of the data (Figure 3). Specifically, two choropleths mapped average price and median income against San Diego zip codes to assess whether a median income and location impacted price. A visual assessment suggested that higher median incomes in the main area did not correspond to higher average rental prices. Between the two choropleths, very few regions suggest a direct relationship exists between median income and average price.

4.3 Data Quality

Data quality issues included: missing values, redundant columns, inaccurate data types, outliers, and the high number of features. To ensure data quality and mitigate bias, these features were either cleaned or removed.

4.3.1 Data Cleansing

The median income information provided by the U.S. Census was formatted as a string, containing symbols that were incompatible with numerical data types, such as the dollar sign ($) and the comma (,). These symbols were removed from the values so that median income may be accurately represented as a numerical feature rather than as a string.

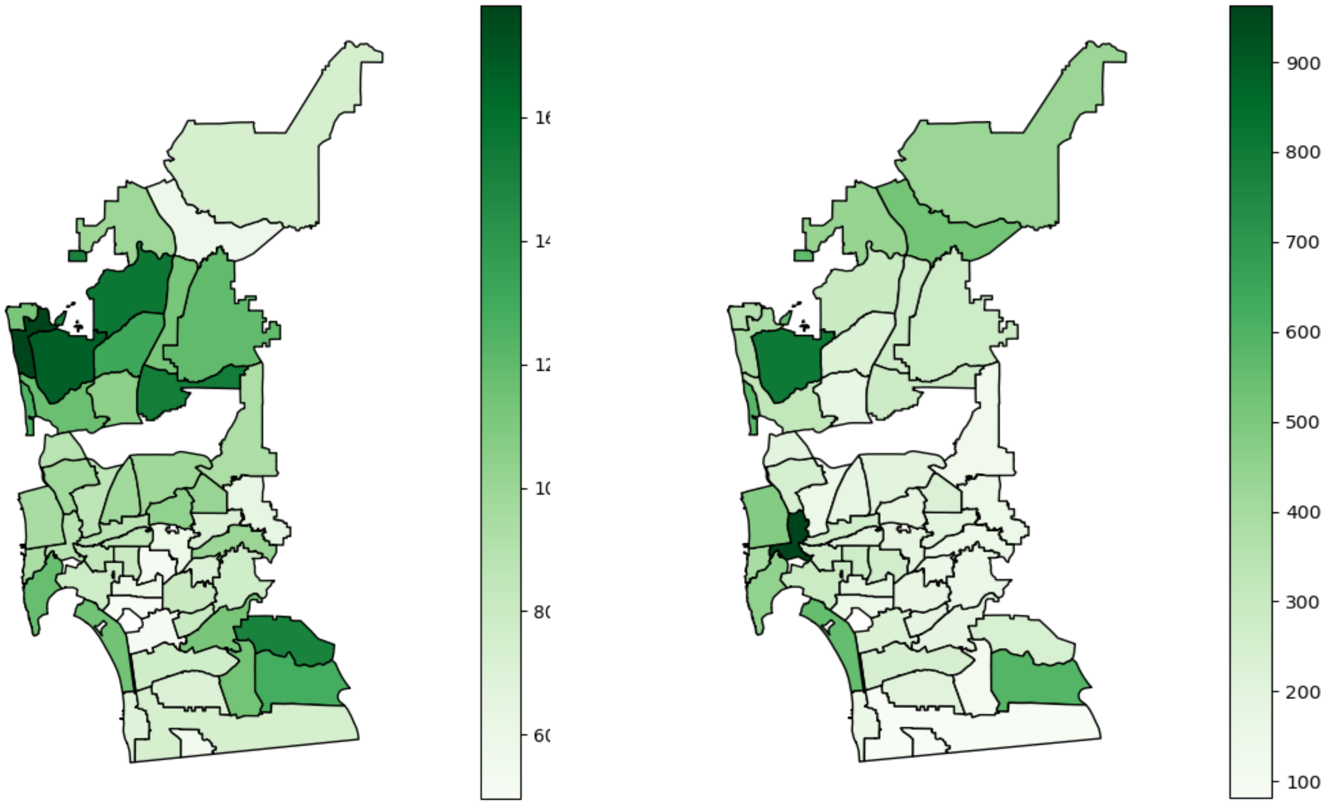
4.3.2 Handling Missing Data

Completeness of the data was evaluated by summing the number of missing values per feature. Two non-relevant features, *neighbourhood_group_cleansed* and *calendar_updated*, contained missing values for all records and were removed. More interestingly, the numerical feature, *bathrooms*, showing the number of bathrooms available at the property, was missing values for every record. The feature *bathroom_text*, which had information regarding the number of bathrooms in a string, was used to fill the numerical *bathrooms* feature.

For imputing the number of missing values in the *bedrooms* field, various strategies were employed. First, it was reasoned that if the *property type* was listed as a room or private room, then the listing only had a single bedroom. Next, every unique combination of the number of bedrooms and bathrooms from the data set were used to fill bedroom values based on the most common bedroom-bathroom combination. Lastly, some values were imputed by manually obtaining the bedroom value from Airbnb.

Figure 3

Average rent price (left) and median income (right) maps of San Diego zip codes.[1]



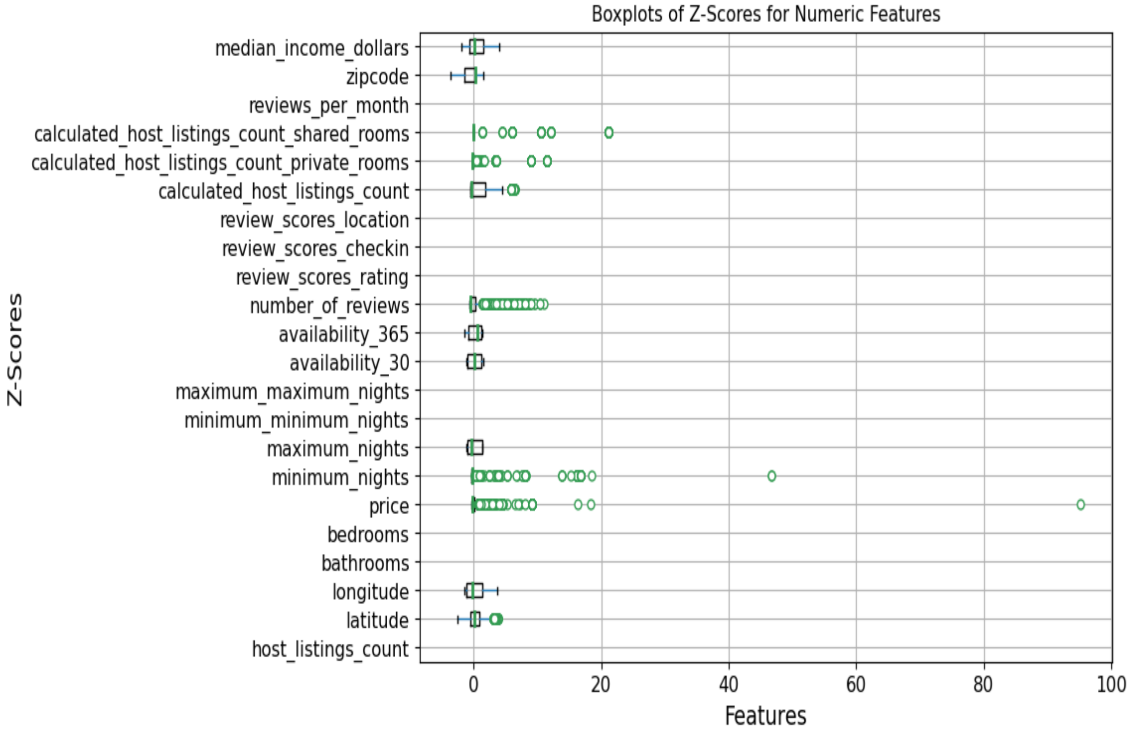
[1] Full maps including zip codes are included in *Figures 7* & 8 in the appendix.

53 records missing values in the fields *host_listings_count*, *minimum_minimum_night*, and *maximum_maximum_nights*, were removed, as removal of 0.2% of the data will likely not impact the model.

The 3,186 missing values in the feature *reviews_per_month* were filled with zeros to penalize the listings that did not receive monthly reviews. This topic is further discussed in section *4.4.2*, on engineering a weighted review-based feature.

Figure 4

Box Plots of numerical features based on Z-Scores.



4.3.3 Outlier Analysis

Outliers were defined as values that surpassed three standard deviations from the respective mean of a given feature. Scipy's *zscore* function was applied to each feature, which calculated the z-score for each value within each feature based on the feature's respective mean and standard deviation. Box plots were plotted to visualize the distribution of z-scores within each feature (Figure 4). Observations above $8,000 in the *price* feature were removed as they skewed the analysis. Obtaining more observations with a price above $8,000 in the future may allow for better performance for high-priced rentals.

4.4 Feature Engineering

Feature engineering was used to produce more meaningful features, which in turn would increase the ability of the models. The *zipcode* and *median_income* features were created in EDA. Likewise, two binary columns were established, "*property_type_binary*" and "*private*," which allowed more in-depth categorization of rental property types by the model.

4.4.1 Sentiment Feature

The feature *sentiment* was engineered to capture the sentiment of text-based variables.

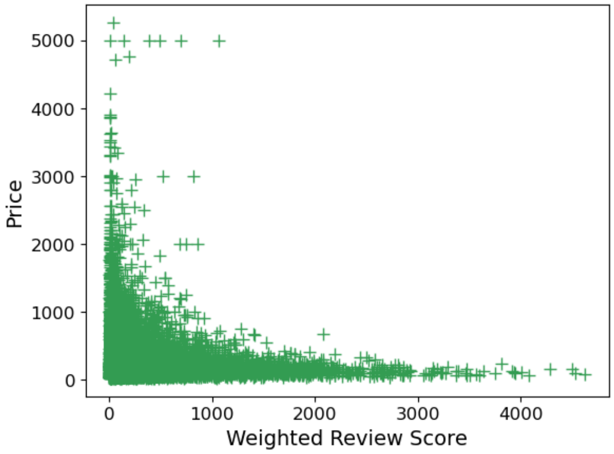
Text-based features *name, description,* and *neighborhood* were combined into a singular text feature. Then, a *DistilBERT base uncased fine-tuned SST-2* pre-trained model, in the *transformers* library, generated a positive or negative prediction for each observation. This prediction was then changed to the probability of the text being positive (0 being negative and 1 being positive) and stored as the *sentiment* feature.

4.4.2 Weighted Review Score Feature

While exploring the importance of reviews in predicting rental prices, the feature *weighted_review_score* was devised to capture the relationship between *review_scores_rating* and the *number_of _reviews*. It was derived by multiplying the two features for each rental property. The engineered feature resulted in a negative correlation with the rental prices, indicating the potential as a predictor valuable for modeling and further decreasing the number of features included (Figure 5).

Figure 5

Scatter plot of weighted review score against rental prices.



4.5 Modeling

As Voltes-Dorta & Sanchez-Medina (2020) found that separate models by property types led to higher quality predictions, the cleaned data set was partitioned between entire houses or single rooms. Then, a pipeline was established to further preprocess the partitioned data sets. The pipeline partitioned 75% of the dataset into a training set and 25% into a test set, separated the target variable from the data sets, and applied preprocessing steps based on the features' respective data types. The numerical predictors were Z-score standardized. By isolating *price* before applying the standardization, the original units of the target variable were preserved. Categorical predictors were each one-hot encoded. After encoding, the training data contained 50 features including the engineered features, property size and type variables, and availability measures.

Supervised models were trained on the training sets and evaluated on the test sets. Each model was trained with *Scikit-learn*, except for the linear regression which was trained with *statsmodels*. Models were tuned with a grid search and a five-fold cross-validation, optimizing for the lowest root mean squared error. However, for the gradient boosting regressions and the house-based neural networks, a randomized search method was used due to the high computation time. Incorporating cross-validation into hyperparameter tuning helped reduce risks of overfitting to the training data, particularly for the complex models such as the neural network.

4.5.1 Linear Regression

Two multiple ordinary least squares linear regression models were fit using all features. Both models' assumptions were evaluated using four diagnostic plots. The residual vs. fitted value plots for each model exhibited higher residuals as the fitted values increased, indicating an issue with heteroscedasticity. This was also indicated in the scale-location plots, which both were non-linear. The residual vs. fitted plots had high positive residuals, suggesting the models were undervaluing high-priced rentals. Each QQ plot had a heavy right-skew, so the residuals were not normally distributed. A transformation to *price* could solve this violated assumption. However, when natural log and log base 10 transformations were tested, the diagnostic plots and metrics did not show any improvements. This shows the data is likely best suited to non-linear modeling techniques. Finally, the residuals vs. leverage plots had many points with high leverages, indicating that a model more resistant to outliers could be more successful.

4.5.2 Regularized Regression

Three regularized linear regression methods were used for their embedded feature selection. Lasso regression adds the L1 regularization penalty to the sum of square errors (SSE). The L1 penalty is the sum of the absolute values of the coefficients multiplied by the tuning parameter, *alpha*. As the L1 penalty gets higher, more of the coefficients will shrink to zero, performing feature selection. *Alpha* was optimized ranging from 0.001 to 100.

Next, Ridge regression models, which used the L2 penalty for regularization, were trained. The L2 penalty adds the sum of squared values of the coefficients multiplied by the selected tuning parameter, *alpha*, to the SSE. This will shrink coefficients towards zero, but never to zero. Thus, the L2 penalty may reduce potential multicollinearity issues, without excluding features. The *alpha* parameter was optimized, using the same range as the Lasso regression.

The Elastic net regression was trained as a compromise between the two penalties, as the model adds both the L1 and L2 term to the SSE. The hyper parameter for the L1 term, *alpha*, was optimized from 0.001 to 100, while the hyper parameter for L2, *l1_ratio*, was evaluated from 0 to 1.

4.5.3 Random Forest

A random forest regression algorithm was trained to assess whether a non-linear model would have greater predictive power. As a random forest is an ensemble of decision trees, this algorithm was also selected for its ability to reduce potentially high variance and overfitting issues. The number of trees, *n_estimators*, was evaluated from 200 to 1000. The maximum depth of the trees, *max_depth,* was optimized ranging from two to ten. The minimum number of observations required to split an internal node, *min_samples_split,* was tuned from two to five. Finally, the minimum number of observations required at each leaf node, *min_samples_leaf,* was assessed from two to four.

4.5.4 k-Nearest-Neighbors (kNN)

KNN regression models were trained for their interpretable results and non-parametricism. The models were tuned with neighbors ranging from three to 29, uniform versus weighted distances to the neighbors, and Manhattan versus Euclidean distance.

4.5.5 Support Vector Regression (SVR)

SVR models were tuned for their robustness to outliers and their flexibility with kernels, allowing evaluation of different feature spaces. The SVR models were tuned evaluating *linear, poly, rbf,* and *sigmoid* kernels and *scale* versus *auto* gammas. Gamma serves as the kernel coefficient for the non-linear kernels. An L2 regularization penalty, *C,* was tuned from 0.01 to 1000. Lastly, *epsilon* was tuned from 0.01 to 1. Epsilon specifies the epsilon-tube, which specifies the regions in which errors are not penalized.

4.5.6 Gradient Boosting Regression

To take advantage of their general high performance and adaptability, gradient boosted regression models were trained. The number of boosting stages *n_estimators* was evaluated from 50 to 500. Learning rate, a hyperparameter controlling the contribution of each estimator, was assessed from 0.01 to 0.5. Maximum depth of the regression trees was optimized from three to seven. The number of observations to split an internal node, *min_samples_split*, was tuned from two to 20, while the observations required to be at a leaf node, *min_samples_leaf,* was evaluated from one to eight. *Subsample* was tuned from 0.5 to 1. A subsample below one causes the model to perform stochastic gradient boosting, and as the subsample decreases, bias is added into the estimator.

4.5.7 Neural Network

For the same reasons as the gradient boosting regression, multi-layer perceptron regression models were created. Hidden layer sizes ranging from 50 to 200 neurons and one to three hidden layers were evaluated. Three different hidden layer activation functions; *logistic, tahn,* and *relu* were assessed. The L2 regularization term, *alpha,* was tuned from 0. 001 to 0.5. Finally, the learning rate of the network was tuned from 0.001 to 0.5.

5 Results and Findings

After tuning the models to the training partitions, each model made predictions on the test partition. The predictions from the test sets and the known values were then used to derive three performance metrics, adjusted-R^2, root mean square error (RMSE), and mean absolute error (MAE), for each model. As the gradient boosting model had better results for two of the three metrics on the test set for both property types, the gradient boosting model was determined to be the model with the best predictive capabilities.

5.1 Model Test Set Evaluation

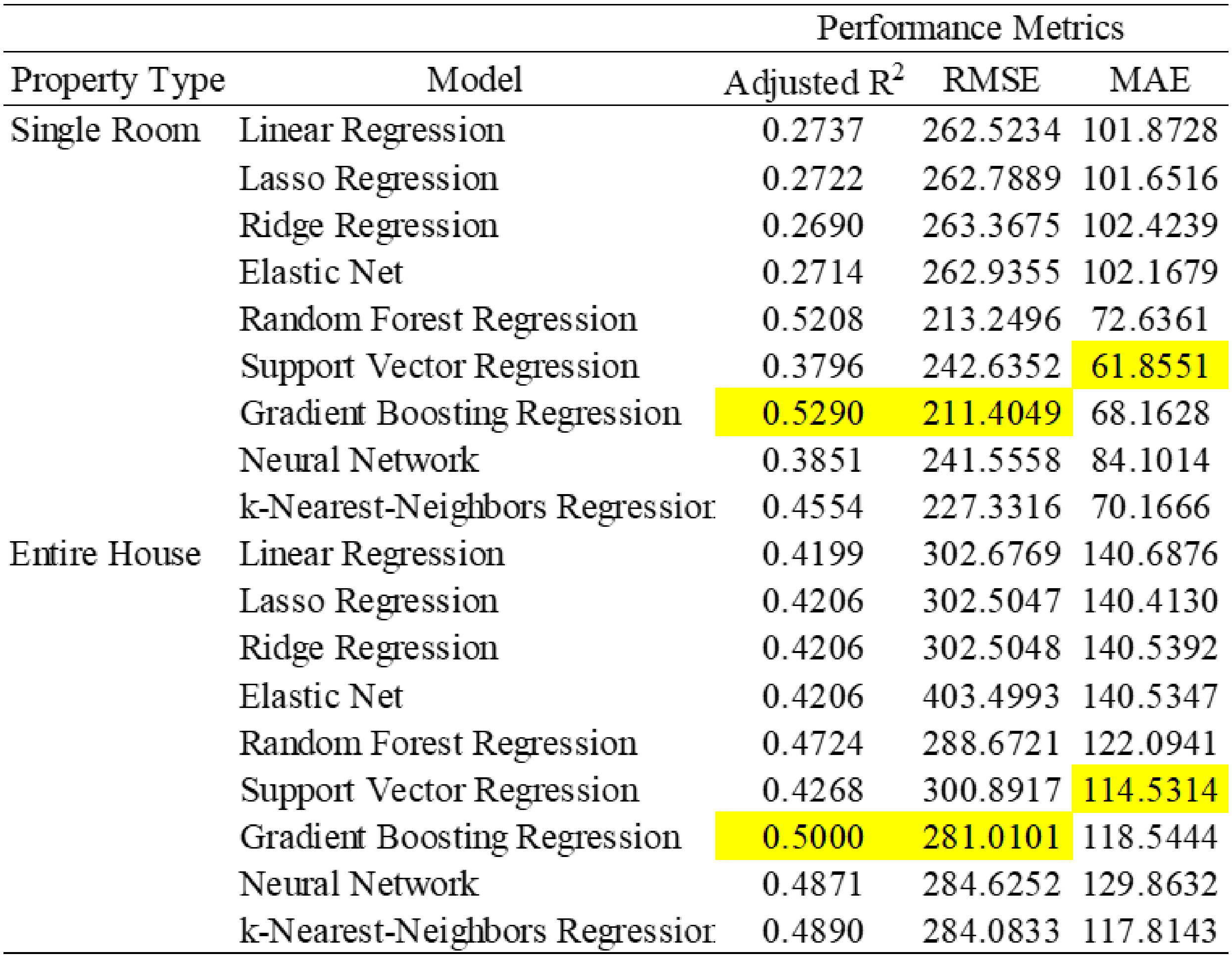
The models were evaluated against one another using three performance metrics, adjusted-R^2, root mean square error (RMSE), and mean absolute error. Adjusted-R^2 was used to interpret the variance captured by the models, while accounting for the number of predictor variables. Root mean square error was used to determine the quality of predictions based on the square root of the averaged squared errors, and MAE measured the model's respective average error. Root mean square error was selected as a performance metric since it is measured in the same units as the target variable; likewise, MAE was selected as a metric due to its robustness to outliers.

An optimal model would have an adjusted-R^2 value as close to one as possible, while minimizing RMSE and MAE. Table 5.1 shows the three metrics based on each model's performance on the test set data for each property type. The gradient boosting regression model had the highest adjusted-R^2 and lowest RMSE value for both property types. For single room properties the adjusted-R^2 value and RMSE were 0.5290 and 211.4049, respectively, whereas for entire home properties the adjusted-R^2 and RMSE values were 0.5000 and 281.0101, respectively. The SVR with an rbf kernel had the lowest MAE for both property types, 61.8551 for single room model and 114.5314 for entire house model. In addition to outperforming all other trained models in two out of three performance metrics, the gradient boosting regression also contained a built-in function for feature importance, a key goal of our model. As a result, gradient boosting regression is the optimal model for predicting prices of single room-type properties and entire house-type properties.

The gradient boosting regression models predicted Airbnb prices within a reasonable margin of error. This supports the initial hypothesis that a machine learning model can be utilized in helping renters assess both fair rental prices and the key determinants of these price

Table 5.1

Model Performance Metrics



		Performance Metrics		
Property Type	Model	Adjusted R^2	RMSE	MAE
Single Room	Linear Regression	0.2737	262.5234	101.8728
	Lasso Regression	0.2722	262.7889	101.6516
	Ridge Regression	0.2690	263.3675	102.4239
	Elastic Net	0.2714	262.9355	102.1679
	Random Forest Regression	0.5208	213.2496	72.6361
	Support Vector Regression	0.3796	242.6352	61.8551
	Gradient Boosting Regression	0.5290	211.4049	68.1628
	Neural Network	0.3851	241.5558	84.1014
	k-Nearest-Neighbors Regressior	0.4554	227.3316	70.1666
Entire House	Linear Regression	0.4199	302.6769	140.6876
	Lasso Regression	0.4206	302.5047	140.4130
	Ridge Regression	0.4206	302.5048	140.5392
	Elastic Net	0.4206	403.4993	140.5347
	Random Forest Regression	0.4724	288.6721	122.0941
	Support Vector Regression	0.4268	300.8917	114.5314
	Gradient Boosting Regression	0.5000	281.0101	118.5444
	Neural Network	0.4871	284.6252	129.8632
	k-Nearest-Neighbors Regressior	0.4890	284.0833	117.8143

5.2 Web Application Build and Deployment

Streamlit, a Python-based free and open-source framework to build machine learning web apps, was used to deploy the optimal models. Three main python files are necessary for the app: *main.py*, *prediction.py*, and *model.py* (a *prediction* file was not necessary in this app as the models were pickled). The data transformation pipelines were mirrored to preprocess the loaded data for each property type and saved into the *model* file. In the *main* file, the Gradient Boosting regression models were loaded and fitted to the pipelines to ensure consistency for the model deployment. To increase the user-friendly experience, the app build included two pages for data entry. The first page is the *URL Mode* and the second page is the *Manual Mode*. The URL mode allows the user to easily paste in an existing Airbnb URL (if applicable) and pulls the information from the modeling dataset to predict the price. If the property is not in the modeling dataset, the interface includes a second page for the user to manually input the information necessary to predict the price as well. The final page, *About,* includes a basic description of how gradient boosting regressions work and the feature importances for each model, allowing consumers to make their own conclusions on the model's predictions. The deployed web application was titled as *Fairbnb* and provides an easily accessible platform for users to input property details, receive estimated prices, and make informed decisions on Airbnb rental prices. The app also provides users with a reliable and convenient tool to understand the pricing dynamics of the Airbnb market within the San Diego area. In doing so, it influences consumers to optimize pricing strategies and understand listing prices relative to the market value.

6 Discussion

The RMSE of the gradient boosting regression model on the test data set was $211.40 for single rooms and $281.01 for entire homes. While this may seem like a large discrepancy, the MAE suggests the RMSE could be inflated due to outliers, with an MAE of $68.16 for single room-type listings and $118.54 for entire home-type listings. The results are aligned with the original objective of the study, which aimed to train a model capable of assessing fair rental prices while acknowledging the possibility of error. If the error metrics are minimized too close to zero, it could imply that the model has been trained to the listers' biases.

The performance metrics also indicate the model's predictions deviate further from the expected listing price if the property is an entire house than if the property is an individual room. This difference in predictive power by property type could be attributed to a multitude of factors, such as inadequate representation in the data set, non-representative features, or real-world differences in price setting biases by property types.

In prior research, Lektorov et al. (2023) and Dhillon et al. (2021) found random forest models yielded the lowest error when analyzing Airbnb prices. This study found the random forest algorithm reported comparable results to the gradient boosting regression model, with only slight differences in metrics. Given further optimization, these results could change. However, at this time, the results of the study have shown gradient boosting models can perform as well as random forest models in the prediction of Airbnb prices.

Through evaluating the variable importance in the context of feature creation and optimization, it was evident the engineering approaches enhanced each models' predictive capabilities. The sentiment-based feature was the most important predictor in the room-based model, while being seventh most important for the home-based model. In addition, the number of bedrooms and bathrooms were identified as key determinants in the models, particularly in the house-based model. Moreover, creating weighted review scores and median income features resulted in the establishment of key predictors, with each appearing as important features in both models.

This study included limitations that warrant acknowledgement to ensure the transparency of the findings. Due to limited data access and a vast array of potential features, it was out of the scope for this study to consider every potential feature. Likewise, consideration of hundreds of features could not be completed given limited computational resources. It should also be noted that the application developed within this study is only applicable to the region of San Diego, California, and is unlikely to be generalizable to other locations.

Furthermore, not all machine learning algorithms are easily explainable. To address this limitation, techniques like impurity-based feature importance were employed to identify the relative contribution of features to the model output and discuss the insights to end users. However, impurity-based feature importance does not show the direction of the relationship with *price.* Therefore, domain knowledge should be used to approximate the direction of each relationship.

6.1 Conclusion

In conclusion, this study used the gradient boosting regression models to successfully predict the Airbnb prices for both property types, single room and house, in the San Diego area. Upon evaluation via various performance metrics (adjusted R2, RMSE, and MAE), the gradient boosting regression models were selected as the final model because the models combined high predictive capability and integrated feature interpretability[2]. To provide transparency into rental pricing dynamics, the final models were deployed to the web application titled *Fairbnb*. Allowing transparency and insights into the factors that may influence rental prices, the application allows the consumers to make the informed decisions on fair rental prices. While this study contributes solely to the San Diego rental market, the work done here provides a baseline methodology to promote worldwide price fairness and consumer knowledge.

6.2 **Recommend Next Steps and Future Studies**

As previously noted, this study is limited to the exploration of a subset of the potential feature space. Future studies may expand upon the work done here by exploring the impact on prices by other features, either independently or in the presence of the features used here. Likewise, the models were optimized by searching for a limited hyperparameter space. Further hyperparameter tuning may be performed to potentially increase predictive power.

At this time, in *manual mode*, the application uses the mean value for features that the user cannot fill in manually, such as *sentiment* and *reviews_per_month.* While the application subsets the data to acquire mean values from listings similar to that of interest, better values would be acquired by scraping the listing of interest directly. Thus, the application can be expanded by incorporating a web scraping aspect to acquire more accurate data.

ACKNOWLEDGMENTS

We would like to thank Dr. Ebrahim Tarshizi for providing oversight and feedback. We would also like to thank the team at Inside Airbnb for providing the scraped Airbnb data. Lastly, we want to thank the SOLES writing center at the University of San Diego for their feedback on the written content of this article.

References

Albrecht, J., Ramachandran, S., & Winkler, C. (2021). *Blueprints for text analysis using Python: Machine Learning-based solutions for common real world (NLP) applications*. O'Reilly.

Albouy, D., Ehrlich, G., & Liu, Y. (2016, November 1). *Housing Demand, Cost-of-Living Inequality, and the Affordability Crisis*. National Bureau of Economic Research

[2] Model feature importance charts are included in *Figures 9 & 10* in the appendix.

Casamatta, G., Giannoni, S., Brunstein, D., & Jouve, J. (2022). Host type and pricing on Airbnb: Seasonality and perceived market power. *Tourism Management, 88*, 104433.

Cheung, K. S., & Yiu, C. Y. (2022). *The paradox of airbnb, crime and house prices: A reconciliation*. Tourism Economics, 135481662211028. https://doi.org/10.1177/13548166221102808

Dhillon, J., Eluri, N. P., Kaur, D., Chhipa, A., Gadupudi, A., Eravi, R. C., & Pirouz, M. (2021, January). Analysis of Airbnb Prices using Machine Learning Techniques. In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 0297–0303). IEEE.

Garcia-López, M.-À., Jofre-Monseny, J., Martínez-Mazza, R., & Segú, M. (2020). *Do short-term rental platforms affect housing markets?* Evidence from Airbnb in Barcelona. Journal of Urban Economics, 119, 103278. https://doi.org/10.1016/j.jue.2020.103278

Heidari, M., Zad, S., & Rafatirad, S. (2021, April). Ensemble of supervised and unsupervised learning models to predict a profitable business decision. In *2021 IEEE International IOT, Electronics and Mechatronics Conference* (pp. 1–6). IEEE.

Kalehbasti R., P., Nikolenko, L., & Rezaei, H. (2021, August). Airbnb price prediction using machine learning and sentiment analysis. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction* (pp. 173–184). Cham: Springer International.

Lee, D. (2016). How Airbnb short-term rentals exacerbate Los Angeles's affordable housing crisis: Analysis and policy recommendations. *Harvard Law and Policy Review, 10,* 229. https://harvardlpr.com/wp-content/uploads/sites/20/2016/02/10.1_10_Lee.pdf

Lektorov, A., Abdelfattah, E., & Joshi, S. (2023, March). Airbnb Rental Price Prediction Using Machine Learning Models. In *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 0339–0344). IEEE.

Pai, P.-F., & Wang, W.-C. (2020). *Using Machine Learning Models and Actual Transaction Data for Predicting Real Estate Prices*. Applied Sciences, 10(17), 5832. https://doi.org/10.3390/app10175832

Tang, E., & Sangani, K. (2015). *Neighborhood and price prediction for San Francisco Airbnb listings*. Departments of Computer science, Psychology, economics–Stanford University. https://cs229.stanford.edu/proj2015/236_report.pdf

U.S. Bureau of Labor Statistics. (2023). *CPI for All Urban Consumers (CPI-U)*. https://data.bls.gov/timeseries/CUUR0000SA0L1E?output_view=pct_12mths

U.S. Census Bureau. (2023). *S1901: Income in the past 12 months (In 2021 inflation-adjusted dollars).* https://data.census.gov/

Voltes-Dorta, A., & Sánchez-Medina, A. (2020). Drivers of Airbnb prices according to property/room type, season and location: A regression approach. *Journal of Hospitality and Tourism Management, 45*, 266–275. https://doi.org/10.1016/j.jhtm.2020.08.015

Zhang, Z., Chen, R. J., Han, L. D., & Yang, L. (2017). Key factors affecting the price of Airbnb listings: A geographically weighted approach. *Sustainability, 9*(9), 1635. https://www.mdpi.com/2071-1050/9/9/1635

Appendix

Figure 6

Scatter Plots of Features against Price below $8,000

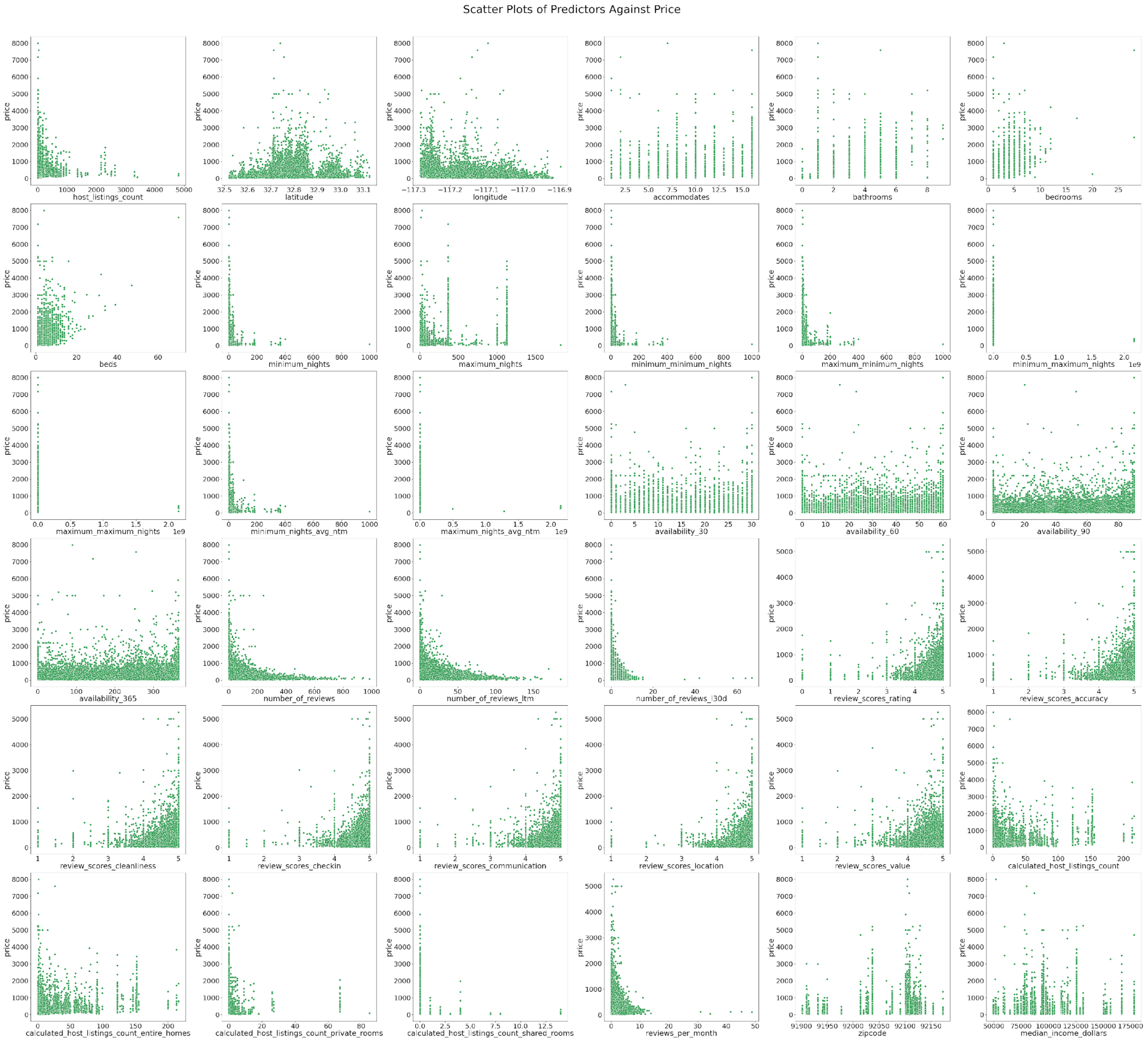


Figure 7

Average Airbnb Rental Price by Zip Code for San Diego, CA.

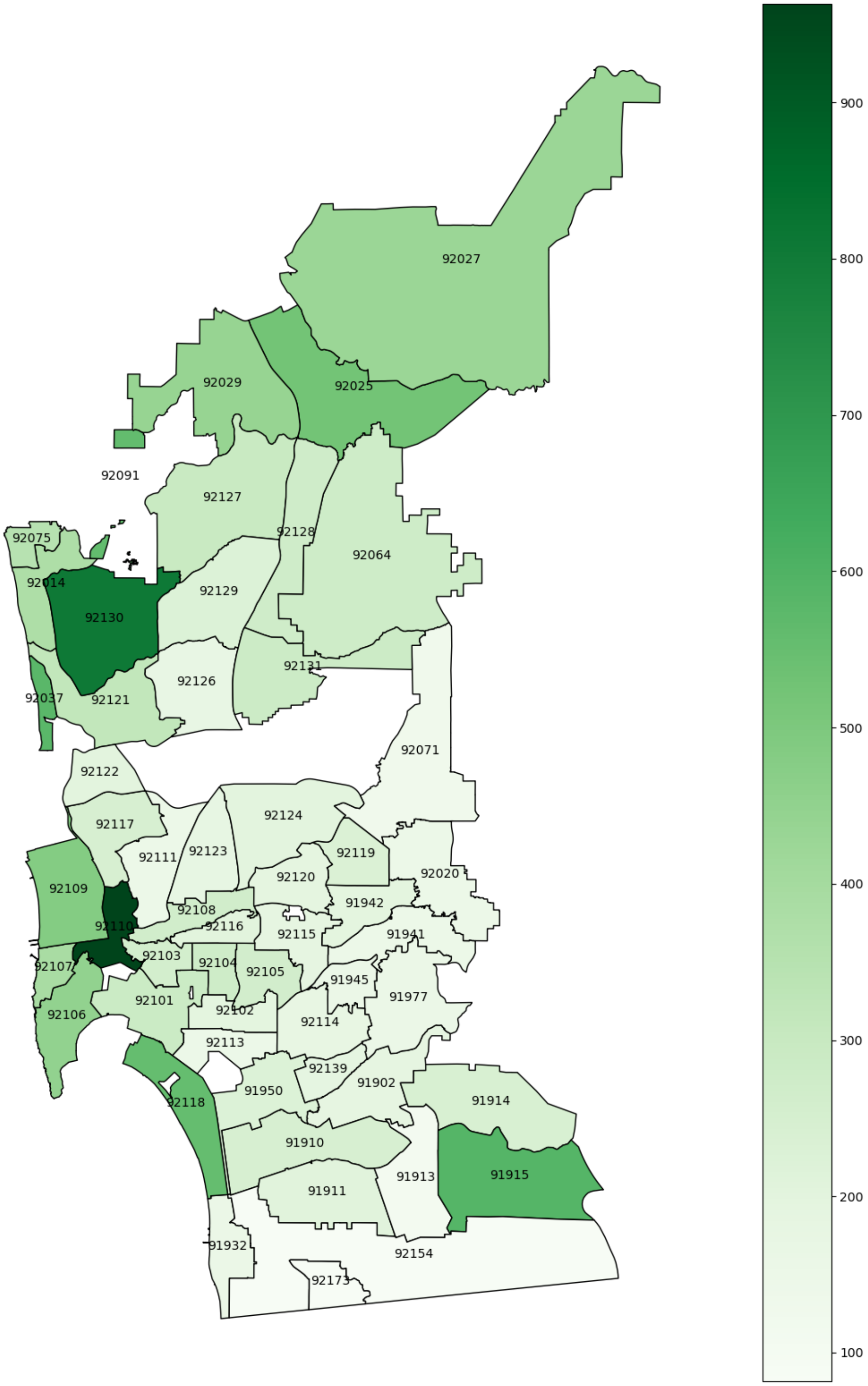


Figure 8

Median Income by Zip Code for San Diego, CA

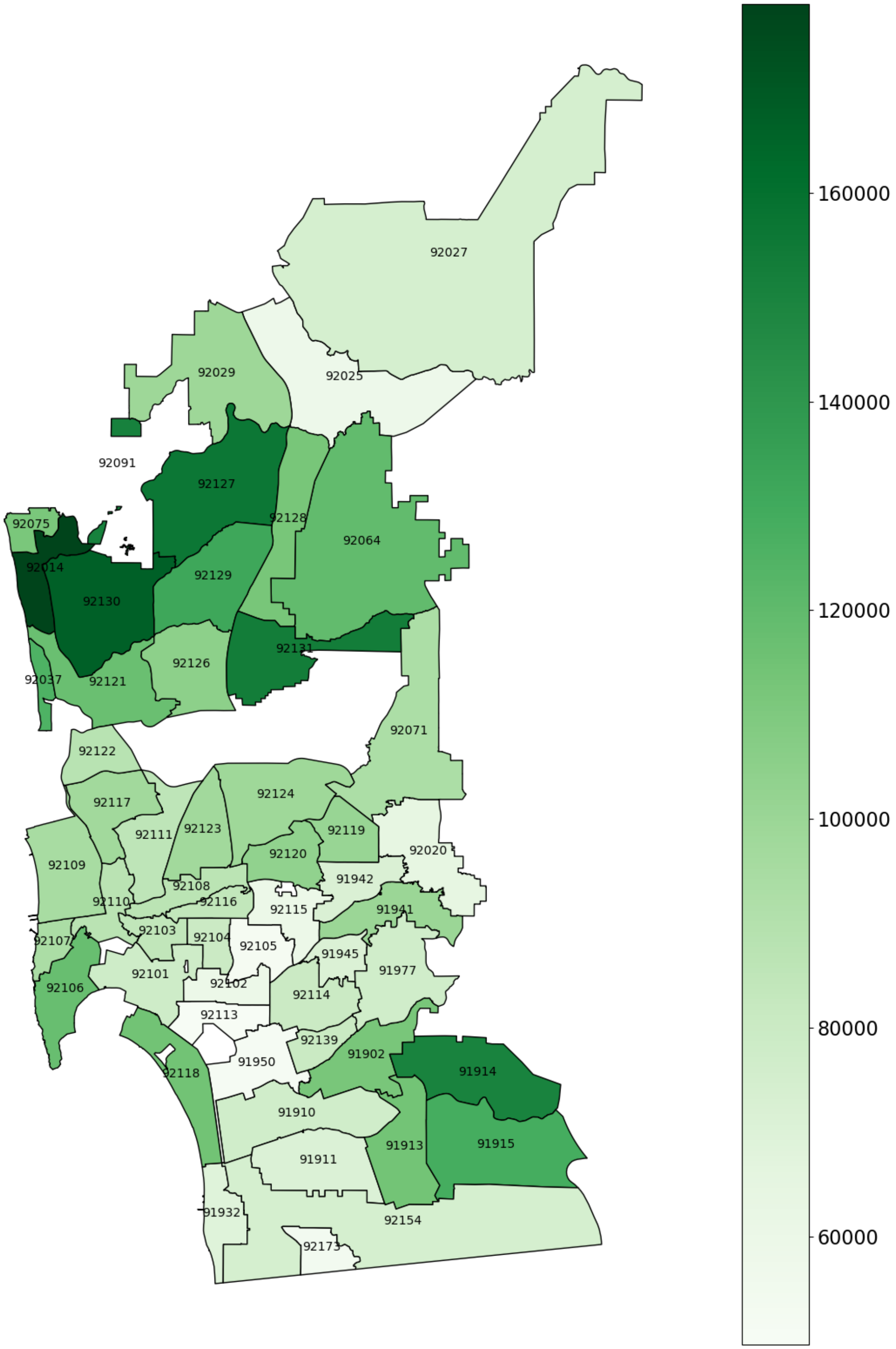


Figure 9

Feature Importance for the Room-Based Model

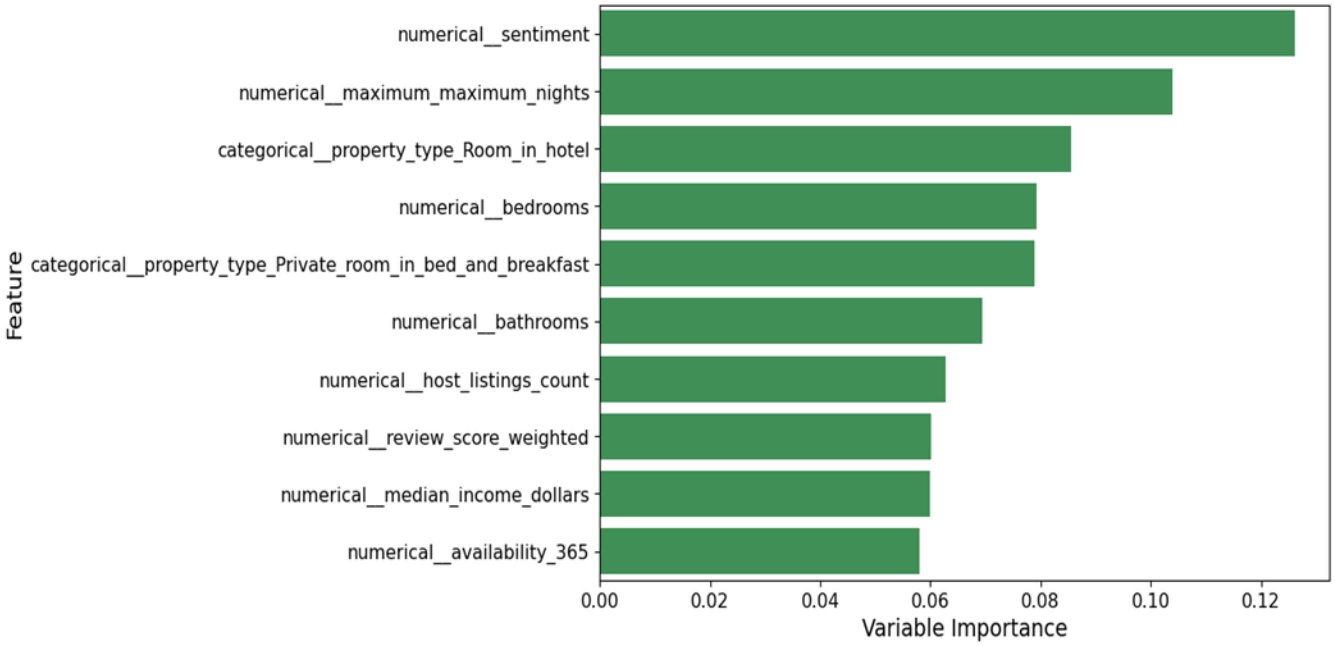
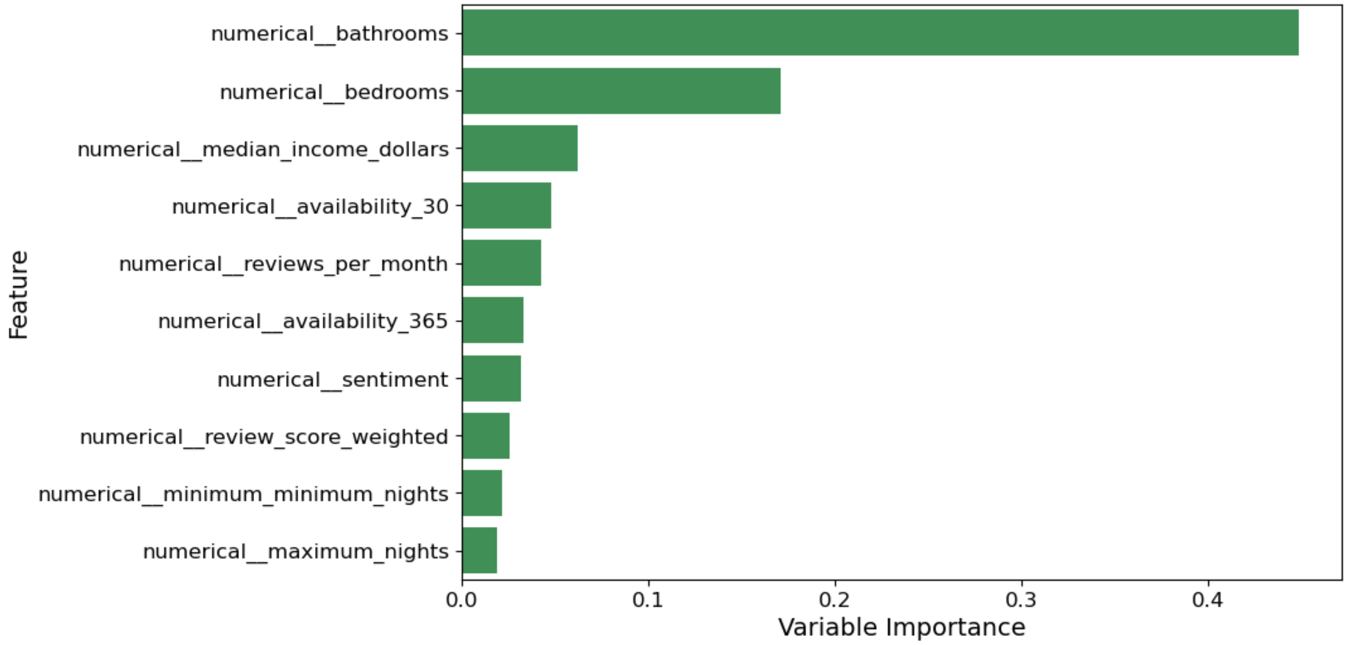


Figure 10

Feature Importance for the House-Based Model


numerical__bathrooms
numerical__bedrooms
numerical__median_income_dollars
numerical__availability_30
numerical__reviews_per_month
numerical__availability_365
numerical__sentiment
numerical__review_score_weighted
numerical__minimum_minimum_nights
numerical__maximum_nights
Feature
Variable Importance