AirBnB Price Prediction

Nicholas Wong
University of North Carolina at Chapel Hill
`nicwjh@ad.unc.edu`

Abstract

AirBnB, an online marketplace facilitating homestays and experiences, has grown to own approximately a quarter of the reservation-and-online-booking market and a fifth of the vacation rental market in recent years. As it is a platform that facilitates interactions between buyers and sellers, price is naturally a primary covariate of interest for both parties involved in the transaction. In this paper, we propose a machine learning and econometrics based approach for the price prediction of AirBnB listings in major U.S. cities. For price prediction, we deploy 4 different supervised methods - forward selection, LASSO regression, polynomial regression, and boosting. These methods are specifically chosen to be split approximately evenly between linear and nonlinear methods. The motive of this study is to attempt to build a price prediction model that can prognosticate the price of an AirBnB listing with maximum accuracy given a vector of input features. Experiments are run on the AirBnB listings in Major U.S. Cities Dataset sourced from the Kaggle repository in [4]. For performance evaluation of our regression models, cross-validation is used to estimate prediction error with Root Mean Squared Error (RMSE) as a measurement metric. Results show that amongst the models tested, boosting is the best model for predicting price of AirBnB listings. Additionally, input features indicative of maximum occupancy or location are found to be most significant.

1 Introduction

In many ways, AirBnB has ushered in a new generation of lodging. Since it was founded in 2008, AirBnB has become one of the most successful pioneers of the sharing economy, transforming the travel industry around the world. By helping more than 200 million guests find individual hosts, AirBnB has shaken up the hospitality business and urban real estate markets. As of 2023, AirBnB has a 28% share in the reservation-and-online-booking market. This growth has brought AirBnB from nothing to a $75bn firm in just 15 years[2].

Today, AirBnB has over 150 million users and hosts more than half a billion guests per year[3]. With 5.6 million listings spanning over 220 countries and regions, it has become the go-to vacation renting service for many travellers today. Given its ubiquity, proper pricing of AirBnB listings has never been more important. It is of primary interest to both buyers and sellers in the AirBnB marketplace. In an economic context, buyers are seeking to yield the best utility for their dollar spent, or the best "bang-per-buck". Sellers, on the other hand, are trying to maximize the return on their real estate investment in the form of profit. The ability to accurately forecast what would be deemed fair-value for an AirBnB listing is thus immensely valuable for both parties in the vacation home and rental market.

This project aims to explore both linear and nonlinear regression models for price prediction of AirBnB listings in the U.S. Specifically, we aim to compare the predictive performance of forward selection, LASSO regression, polynomial regression, and boosting by evaluating their performance through Root Mean-Squared Error (RMSE). With the identification of an accurate predictive model, buyers/sellers on the AirBnB platform can obtain accurate predictions of listing price given relevant input features. By performing model selection and examining the significance of input features remaining in the model, we can also comment on the relative variable importance. In other words, we can identify the most significant subset of variables that allows us to predict the price of an AirBnB listing with maximum accuracy. Consumers then are able to use the information gleaned from our measure of variable importance as a yardstick to assess fair-valued listings.

2 Literature Review

Prior to beginning, we surveyed other projects that used machine learning algorithms to explore a similar problem space in economics. Much of the work in this area is centered around the concept of hedonic regression. Hedonic regression is a regression technique used to determine the value of a good, service, or asset by fractionating the product into constituent parts or characteristics. This is done to determine the contributory value of each characteristic separately through regression analysis. The regression model should be able to place values and weights on each component or contributory factor to determine the value of the product/service[9]. Yazdani in [12] for instance compares deep learning methods to hedonic methods for real estate price prediction - a common example where hedonic pricing methods are used as the piece of land can be determined by both internal factors as well as characteristics of its surrounding environment[8]. In this project, we apply linear regression in a similar fashion whereby 'hedonic' features are able to be removed from price. In other words, given the attributes of a particular AirBnB listing, our models would be able to price the AirBnB listing even if an exact match of attributes is not found in the dataset. However, while hedonic price regressions have been primarily used for inference, we seek to utilize machine learning methods for prediction.

Some economics studies we surveyed use deep learning via Artificial Neural Networks (ANNs) for price prediction. Awotunde et al. in [11] adapts Long Short-Term Memory (LSTM) to build a cryptocurrency price prediction model. Many other financial economics projects also favor ANNs due to their flexibility and ability to adapt to the stochastic nature of financial fluctuations. For example, Li et al. in [5] explores the application of ANNs in forecasting market prices. For this project, we exclude the use of ANNs/deep learning methods for several reasons. Neural networks are a class of highly flexible learning methods that deal well with high-dimensional and noisy data. However, flexible methods come at a cost of being less interpretable than their less flexible counterparts. Additionally, the model's flexibility makes the data to parameter ratio paramount - regularization of neural networks is key to avoid the risk of overfitting. Hence, we exclude the implementation of neural networks in this research work to prioritize model interpretability over minimizing bias.

In this paper, we present a machine learning based approach to price prediction that differs from much of existing economics literature in scope of methods applied. Many other studies exploring a similar problem space, such as Li et al. in [5], utilize a narrow set of methods in their work. For the case of Li, their work is focused on exploring the implementation of ANNs in-depth. For our work, having a breadth of methods is prioritized to better suit our research goals. Specifically, in this project, both linear as well as nonlinear methods are used to span a range of model flexibility. Fitting models of different flexibilities is important due to the bias-variance tradeoff implicit in fitting flexible models with more degrees of freedom. Minimizing bias often comes at the cost of an increase in variance. In the absence of knowledge of the true form of our data, fitting a wide scope of models gives us a better chance of optimizing the bias-variance tradeoff and choosing an appropriate model.

3 Data

This project uses panel data from a 2017 AirBnB survey, sourced from the Kaggle repository in [4]. The dataset contains 74,111 observations and 28 features, one of which is price in a logarithmic scale. Data preprocessing occurs in several stages. As natural language processing is not a focus of this project, descriptive features requiring substantial amounts of text parsing are removed. Price is exponentiated to a regular scale from a logarithmic scale, *NA* values are removed, strings are cleaned for special characters, relevant character variables are typecasted to factors, and the data is split into a training and test set in an 80/20 ratio. Factor levels for all factor variables are then equated between the training and test sets to avoid factor discrepancies between them. The result of this process of data cleaning is a training dataset with 38,111 observations and a test dataset with 9,558 observations. Both train and test datasets have 22 features, one of which is the target variable *price*. Details on the full list of features are available in the appendix7.

Tuning parameters are selected and models are trained using the 80% training data. The 20% training data is used as unseen held-out data for model evaluation.

It is important to first acknowledge some limitations of our data. Due to missing variables, missing observations, and unclean data, approximately a third of the original 74,111 observations are not able to be used for either training or testing. As model prediction accuracy typically improves when it is trained on more data, this limitation may possibly result in an overestimate of our test RMSE.

In the rest of this section, we present a brief exploratory data analysis of the *price* target variable. Across all observations, the *price* variable has a minimum of 1, mean of 148.8, median of 110, and maximum of 1999. This positive-skewed distribution is illustrated below.

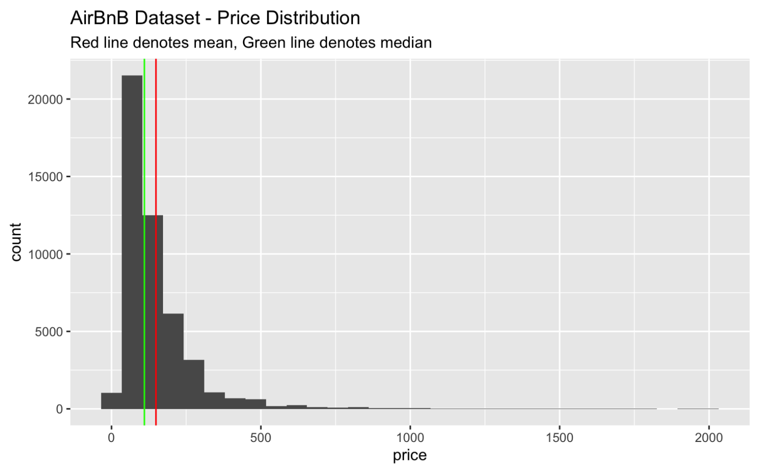


Figure A: Distribution of Price Variable

We also have to acknowledge existing geographic differences in AirBnB pricing. There is a nontrivial difference in the distribution of price when the observations are stratified by geographic region (*city*). High cost of living areas such as New York City (NYC) and Los Angeles (LA) have a price distribution that is skewed upwards.

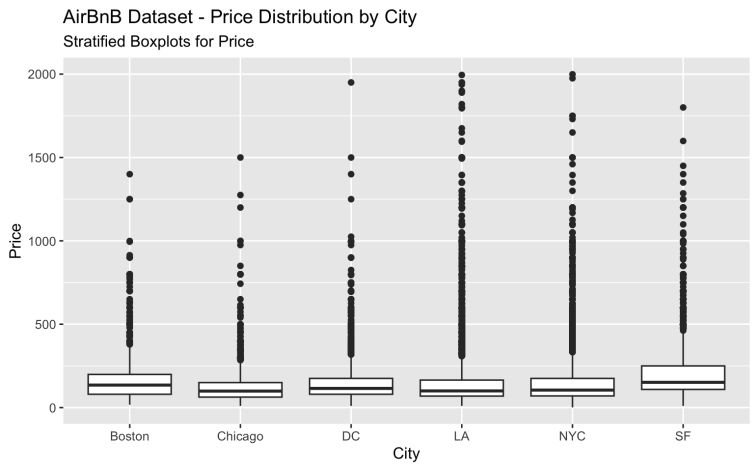


Figure B: Stratified Boxplot of Price

To conclude this exploratory data analysis, we explore the correlations the *price* variable has with the quantitative features in the dataset.

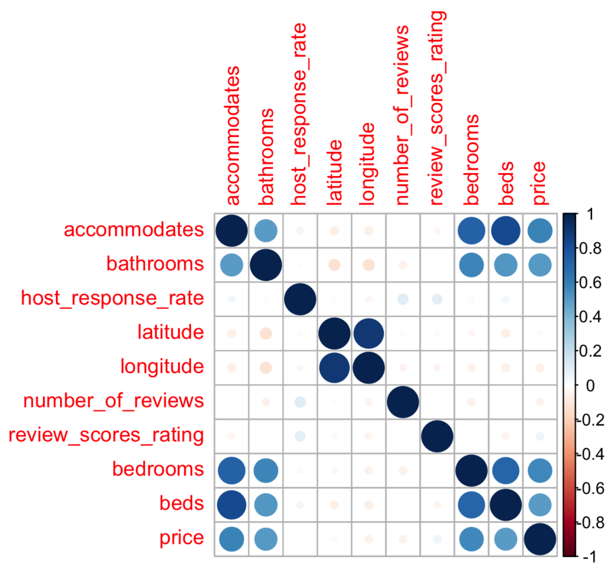


Figure C: Correlation Plot of Price

Notably, *price* has the highest correlation with the feature *accommodates*. For this reason, *accommodates* is used as the input feature for fitting our polynomial regression method. The features *bathrooms*, *bedrooms*, and *beds* are closely behind. As all of the quantitative input features with high correlations with *price* are signals of the maximum occupancy of the AirBnB listing, we can form a preliminary hypothesis that the best quantitative predictor of AirBnB listing price is a measure of the maximum occupancy of the unit. Moreover, since all the quantitative variables identified above are signals of maximum occupancy, there is significant overlap in how these features add predictive power to a model. Thus, we opt to implement a polynomial regression with just a single feature (*accommodates*) as opposed to multiple features for parsimony and interpretability.

4 Models/Methods

As a high-level overview of our methodology, the data is first read and preprocessed. After data preprocessing, for models that need to be tuned, we conduct cross-validation on the training data to choose the tuning parameters for our methods. We then use a validation set approach to assess predictive performance on the held-out data for the following methods: forward selection, LASSO regression, polynomial regression, and boosting. Finally, results from our experiments are compared in a comparative analysis.

The implementation of the project is written in the R programming language, primarily using the classification and regression training (*caret*) package to streamline the process for creating our predictive models[10].

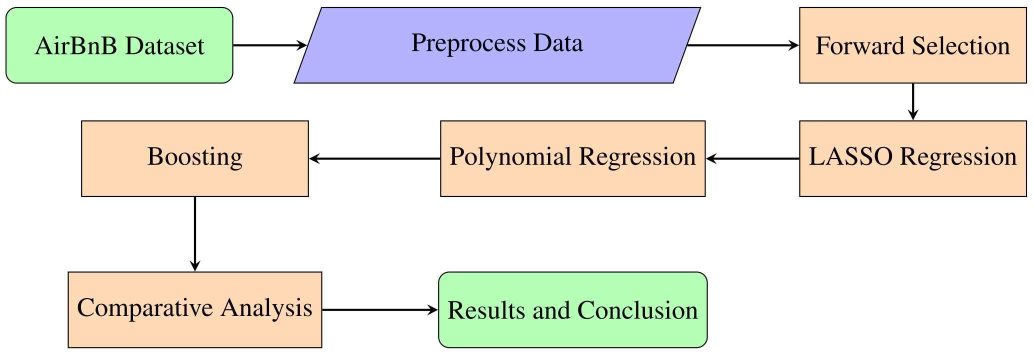


Table 1: Architecture of the Proposed System

These individual methods are expanded upon in greater detail within the subsections below. In this section, we discuss model assumptions, details of our methodology, and our process of cross-validation for tuning parameter selection.

The first two methods we deploy utilize a linear regression framework, where we assume the parametric form:

$$\hat{Y}_i = X^T\hat{\beta} \tag{1}$$

Due to the large amount of variables present in our dataset, for our linear models, forward selection and LASSO are chosen for their ability to perform shrinkage and feature selection.

4.1 Forward Selection

We deploy forward selection, a form of stepwise regression, as our first regressive model. This builds a parsimonious linear regression model iteratively where we begin with the null model and add the best predictor at every step. This algorithm is stopped by the *stepAIC* function when we no longer see an improvement in the Akaike Information Criterion (AIC) metric. Consequently, a feature subset is formed through this iterative process where features are sequentially selected in a greedy fashion. As it is a greedy selection that adds the best predictor at every step, we acknowledge the limitation that forward selection does not provide the flexibility to remove the features that have already been added in case they have become obsolete after the addition of new features.

4.2 Linear Regression with L1 Regularization

We deploy a linear regression model with L1 regularization, also known as LASSO regression, as our second regression model. We opt for LASSO regression as the L1 regularization penalty has the ability to perform feature selection. With p = 21 possible input features, some form of shrinkage is pertinent for this research work. The implemented L1 penalty shrinks some coefficients to zero - this effectively removes the input feature from the model and allows for shrinkage[13]. We deploy LASSO regression, with loss defined as:

$$Loss = Error(Y - \widehat{Y}) + \lambda \sum_{1}^{n} |w_i|$$

The λ tuning parameter, controlling the regularization penalty, is particularly pertinent for LASSO because different values of λ will result in different features being selected. This effectively makes LASSO a backwards greedy selection algorithm where the least useful features are the first to be eliminated by the λ penalty. To accomplish this, we deploy repeated 5-fold cross-validation on the 80% training data (10 repeats). We optimize with the $\lambda \sum_{1}^{n} |w_i|$ penalty, select the remaining features with nonzero coefficients, train an unpenalized model with those features, and compare performance via RMSE. We repeat this process for 16 different values of λ ($\lambda \in$ (0,0.3) with increments of 0.02) on each of the 5 folds and choose the λ that gives the best performance by RMSE. λ = 0.12 is chosen as the outcome of this process of cross-validation.

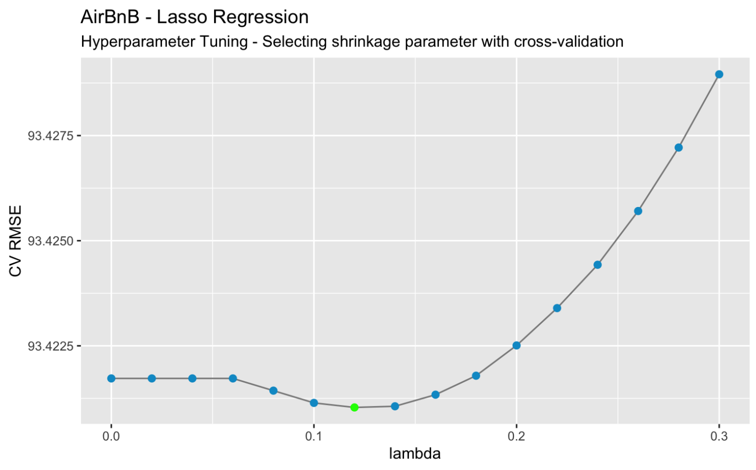


Figure D: Cross-validation RMSE for tuning shrinkage parameter for LASSO

4.3 Polynomial Regression

The methods defined above are fundamentally linear regression methods applied with shrinkage. We deploy polynomial regression as our first nonlinear regression method to predict price of AirBnB listings. It is important to note, however, that while we fit a nonlinear model to the data using polynomial regression, it is a linear statistical estimation problem where E(Y|X) is linear in the unknown parameters estimated from the data.

For a single variable polynomial regression there are two primary tuning parameters that we need to choose: the input feature for which to apply the polynomial regression and the degree of the polynomial.

To choose the input feature for the polynomial regression, we conduct an exploratory data analysis to find the feature most correlated with price. From this, the feature *accommodates* is found to be most highly correlated with the target variable *price*. This process of data exploration is expounded upon in greater detail in 3.

Subsequently, we apply repeated 10-fold cross-validation on the 80% training data to choose the degree of our polynomial. With input feature *accommodates* as our predictor, a model is fit for each polynomial degree (ranging from 1-10) using each of the 10 folds as a test set once. This process is repeated 5 times and averaged using the *caret* package. The average cross-validation RMSE over this repeated 10-fold CV is calculated for each polynomial degree and the polynomial degree with the lowest cross-validation RMSE is then chosen as the degree of our polynomial. Degree = 6 is chosen as the outcome of this process of hyperparameter tuning.

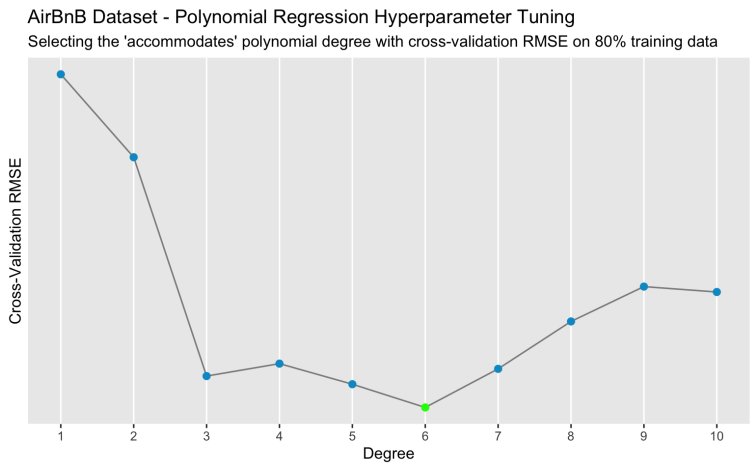


Figure E: Cross-validation RMSE for tuning degree of polynomial

4.4 Boosting

Decision trees are a non-parametric supervised machine learning method that can be used for regression. We apply a gradient boosting algorithm to decision trees, used as our "weak learner" in this ensemble method, where we use a gradient descent algorithm to minimize loss when adding models to the ensemble[6]. At each step, a new weak model is trained to predict the "error" and improve the ensemble model. This iterative process is stopped when a stopping criterion is met. For our project, this stopping criterion is defined as a maximum number of iterations[7].

We apply gradient boosting through the *gbm* function in the *caret* package. In the *caret* package, there are several tuning parameters for the *gbm* function: number of iterations (*n.trees*), complexity of the tree (*interaction.depth*), learning rate (*shrinkage*), and minimum number of training set samples in a node to commence splitting(*n.minobsinnode*).

We set these tuning parameters to be: *n.trees* = 1000, *interaction.depth* = 2, and *n.minobsinnode* = 10. For the shrinkage tuning parameter, we test a sequence of 61 different λ values $\in [10^{-6}, 1]$ through a validation set approach to find the optimal shrinkage parameter that achieves the lowest cross-validated RMSE. λ = 0.2512 achieves the lowest test error and is chosen as a result of this process of cross-validation.

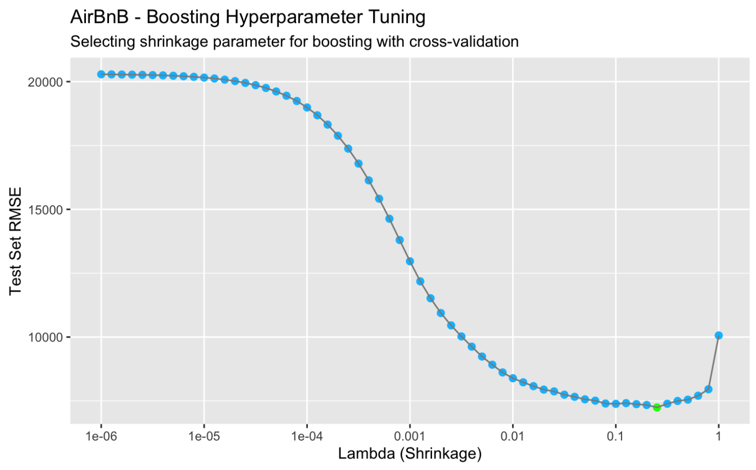


Figure F: Gradient Boosting cross-validation RMSE for choosing λ shrinkage parameter

4.5 Evaluation

Our models are evaluated Root Mean Squared Error (RMSE), which is derived from the closely related Mean Squared Error (MSE). MSE is a metric that measures the average of the squares of the errors - that is, the average squared difference between the estimated values and the actual value. RMSE is simply the square root of MSE. We provide details of MSE and RMSE computation below.

$$MSE = \sum_{i=1}^{D} (x_i - y_i)^2 \tag{2}$$

$$RMSE = \sqrt{MSE} \tag{3}$$

We opt to use RMSE rather than Residual Sum of Squares (RSS) for evaluating prediction accuracy for several reasons. As compared to RSS, RMSE allows for more convenient comparisons between training and cross-validation results. RSS as a metric is dependent on the number of observations the residuals are being calculated over. Consequently, it will be lower for K-fold cross-validation since each fold comprises only a portion of the data. The resulting RSS metric (for K-fold cross-validation) will be $\sim$1/K the size of the training RSS on the full data. With RMSE, we can look at the results for cross-validation on the same scale as with the full training data. This allows for more interpretable results when assessing prediction accuracy between our models or choosing tuning parameters - particularly when K-fold cross-validation is involved.

While we use K-fold cross-validation on the 80% training data for choosing most of our tuning parameters, we opt for a validation set approach for assessing our models. The number of parameters and size of our dataset renders implementing K-fold cross-validation methods computationally infeasible, particularly for our gradient boosting algorithm. As a result, for homogeneity and easier comparisons, all models are tested with a validation set approach on the 20% held-out data.

5 Results and Comparative Analysis

We begin this comparative analysis with a comparison of cross-validation RMSE across our four fitted models. Forward selection has a RMSE of 98.32, LASSO has a RMSE of 100.07, polynomial regression has a RMSE of 115.88, and boosting has a RMSE of 85.11. These results are summarized in the following bar plot.

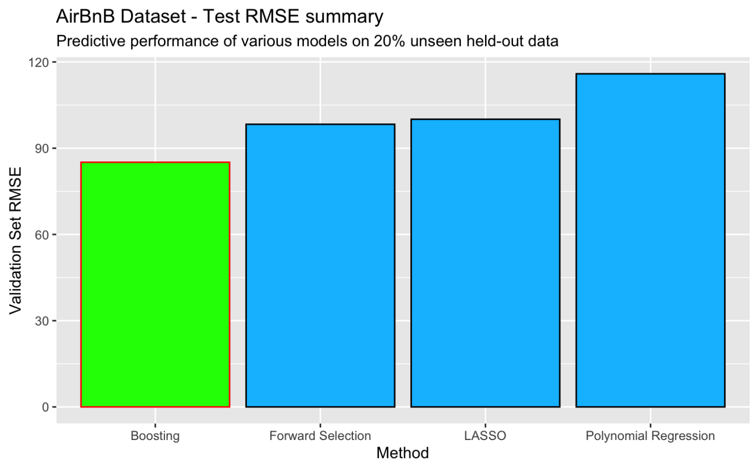


Figure G: Summary of RMSE across all models

Forward selection and LASSO achieve comparable predictive performance. Our boosting algorithm results in a substantial increase in performance through a 15% decrease in RMSE as compared to forward selection and LASSO. We hypothesize that nonlinearities are likely to be present in the data that are being explained by our nonlinear boosting method, resulting in a lower test RMSE. This is confirmed by the plot of residual vs. fitted values from our forward selection model.

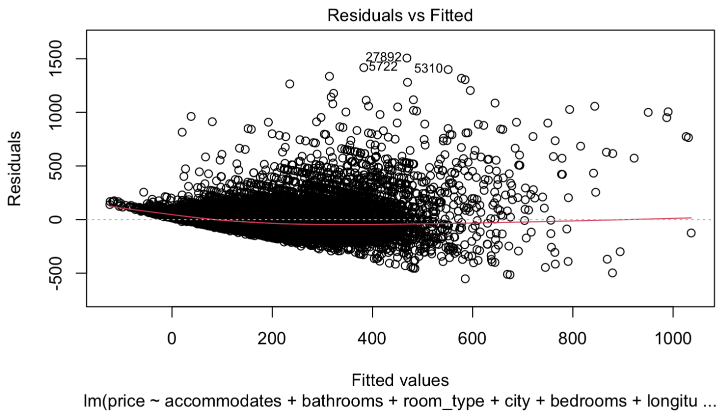


Figure H: Residual vs Fitted plot for Forward Selection

From the plot of residual vs fitted values, we see that there is a definite nonlinear trend. Additionally, the data appears to fan our in a band of nonlinear width, which is indicative of heteroscedasticity/non-constant variance.

Across all models, our polynomial model achieves the worst performance by RMSE. Since our polynomial model is built simply by using the quantitative predictor with highest correlation with *price*, we can conclude that the addition of qualitative/categorical predictors contributes significant predictive power in forecasting *price*.

Next, we will compare the significance of individual features. To do this, we present a table contrasting the features selected by forward selection against the features selected by LASSO as a result of the applied shrinkage[1]. We also present a plot of variable importance to visualize the 10 most important predictors in the boosted model. A full summary of the LASSO and forward selection models with significance levels for individual coefficients is available in the appendix7.

Table 2: Features Selected by LASSO vs Forward Selection

Features selected by LASSO	Features selected by forward selection
property_type	**property_type**
room_type	**room_type**
accommodates	**accommodates**
bathrooms	**bathrooms**
cancellation_policy	**cancellation_policy**
cleaning_fee	**cleaning_fee**
city	**city**
first_review	**first_review**
host_has_profile_pic	**host_has_profile_pic**
host_response_rate	**host_response_rate**
instant_bookable	**instant_bookable**
last_review	**last_review**
latitude	**latitude**
longitude	**longitude**
number_of_reviews	**number_of_reviews**
bed_type	bedrooms
host_identity_verified	review_scores_rating
host_since	beds

[1]Features in common are denoted in bold

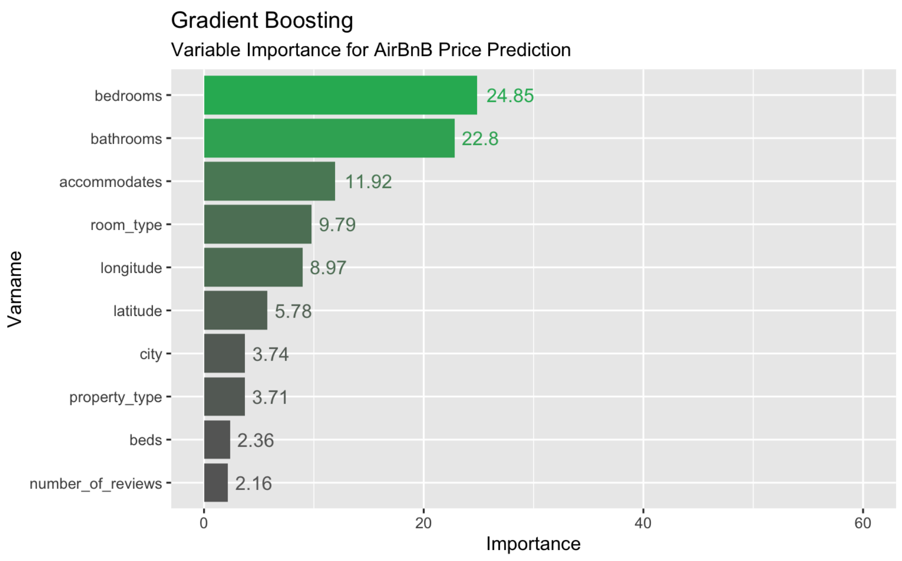


Figure I: Variable Importance for Boosting Algorithm

Results confirm the preliminary hypothesis from our exploratory data analysis. Specifically, quantitative predictors that are signals of maximum occupancy (*bedrooms*, *bathrooms*, *accommodates*) appear to be most significant for forecasting the price of an AirBnB listing across forward selection, LASSO, and boosting. For robustness, we now present a table of the overlapping features across the 10 most important features for boosting, features selected by LASSO, and features selected by forward selection. Features presented in this table are significant across three different models and are the best candidates as input features for predicting price.

Table 3: Overlapping Features across LASSO, Forward Selection, Boosting

Final List of Overlapping Features
bathrooms
longitude
accommodates
room_type
latitude
property_type
city
number_of_reviews

Geography has a significant impact on AirBnB pricing, as confirmed by the inclusion of *latitude*, *longitude*, and *city* as significant predictors across all three models. This conclusion agrees with our exploratory data analysis exploring the distribution of price of AirBnB listings when stratified by city3.

Besides maximum occupancy and geography, other features that appear to be significant for forecasting price of AirBnB listings are the type of booking (*room_type*, *property_type*) and how well reviewed the host is(*number_of_reviews*).

6 Conclusion

The ubiquity of AirBnB has rendered it a mainstay in the vacation home rental space for the last decade. The development of machine learning algorithms that can assist with accurate prediction of AirBnB prices could prove instrumental in ensuring fair-valued transactions for both parties in the AirBnB marketplace.

In accordance with our obtained results, we have the highest likelihood of obtaining better prediction of AirBnB list price when it applied to the boosting model. We attribute this to nonlinearities present in the data which allow boosting, a nonlinear method, to outperform our penalized linear regression methods. The polynomial regression model, on the other hand, achieves the worst performance out of all our tested models. This can be attributed to the significance of categorical predictors - omitting key categorical predictors in favor of solely quantitative predictors in a polynomial regression results in poor prediction accuracy.

In terms of individual features, features indicative of maximum occupancy (*bathrooms*, *accommodates*) and features indicative of geographic location (*longitude*, *latitude*, *city*) carry the most significance for predicting AirBnB list price. This trait is common across boosting, forward selection, and LASSO.

Since our most flexible model has the lowest test RMSE, a potential area for future expansion of this research work is in fitting more flexible methods for price prediction. These could potentially include the implementation of neural networks for AirBnB price prediction, generalized additive models (GAMs), or nonparametric kernel regressions. As mentioned briefly in 2, in this project, we opt for model interpretability over minimizing bias to better suit our research goals. However, with nonlinear trends discovered in the data, fitting and comparing additional flexible methods could potentially result in better predictive performance. The inclusion of appropriate interaction terms, which we do not implement here, could also plausibly prove to be informative.

A limitation of our work, however, is the potential for unaccounted variations in AirBnB pricing. In this project, we are able to account for geographic variations through the inclusion of relevant geographic features. Geographic variations in pricing, however, are not the only variations that could plausibly be present. Seasonal variations, for example, could potentially be present and unaccounted for. This provides another avenue for potential expansion of this research work. As an example, since our data is labelled, one could potentially combine both supervised and unsupervised approaches in a sequential fashion by first implementing a supervised prediction model to identify important features. Subsequently, Ward's method could be applied through an unsupervised hierarchical cluster analysis to understand and type clusters[1]. These clusters could potentially expose unique demographic conditions that drive cluster creation and account for variation.

References

[1] *14.7 - Ward's Method | STAT 505 — online.stat.psu.edu*. `https://online.stat.psu.edu/stat505/lesson/14/14.7`. [Accessed 19-11-2023].

[2] *Airbnb (ABNB) - Market capitalization — companiesmarketcap.com*. `https://companiesmarketcap.com/airbnb/marketcap/`. [Accessed 08-11-2023].

[3] *Airbnb (ABNB) - Statistics*. `https://www.thezebra.com/resources/home/airbnb-statistics/`. [Accessed 08-11-2023].

[4] *AirBnB listings in major US cities — kaggle.com*. `https://www.kaggle.com/datasets/rudymizrahi/airbnb-listings-in-major-us-cities-deloitte-ml`. [Accessed 06-11-2023].

[5] *Applications of Artificial Neural Networks in Financial Economics: A Survey — ieeexplore.ieee.org*. `https://ieeexplore.ieee.org/abstract/document/5692701`. [Accessed 14-11-2023].

[6] Chirag Chadha. *Bagging, Boosting, and Gradient Boosting — towardsdatascience.com*. `https://towardsdatascience.com/bagging-boosting-and-gradient-boosting-1a8f135a5f4e`. [Accessed 16-11-2023].

[7] *Gradient Boosted Decision Trees | Machine Learning | Google for Developers — developers.google.com*. `https://developers.google.com/machine-learning/decision-forests/intro-to-gbdt`. [Accessed 16-11-2023].

[8] *Hedonic Pricing: Definition, How the Model Is Used, and Example — investopedia.com*. `https://www.investopedia.com/terms/h/hedonicpricing.asp`. [Accessed 16-11-2023].

[9] *Hedonic Regression Method — corporatefinanceinstitute.com*. `https://corporatefinanceinstitute.com/resources/data-science/hedonic-regression-method/`. [Accessed 08-11-2023].

[10] Max Kuhn. *The caret Package — topepo.github.io*. `https://topepo.github.io/caret/`. [Accessed 16-11-2023].

[11] *Machine Learning Algorithm for Cryptocurrencies Price Prediction — link.springer.com.* `https://link.springer.com/chapter/10.1007/978-3-030-72236-4_17`. [Accessed 11-11-2023].

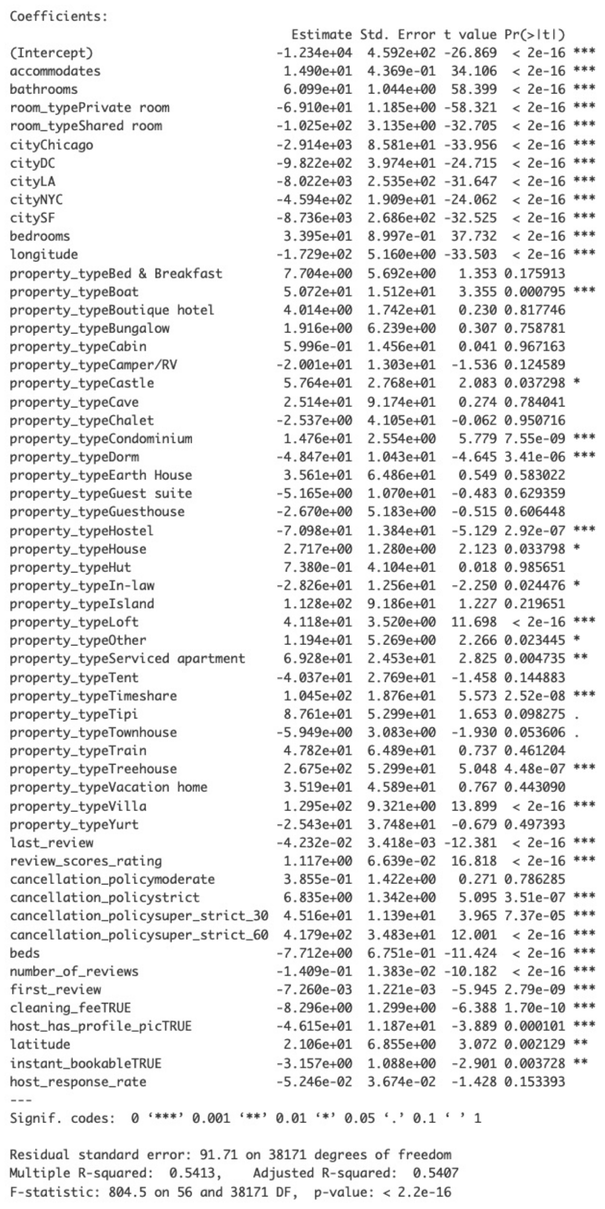
[12] *Machine Learning, Deep Learning, and Hedonic Methods for Real Estate Price Prediction — arxiv.org.* `https://arxiv.org/abs/2110.07151`. [Accessed 16-11-2023].

[13] Great Learning Team. *A Complete understanding of LASSO Regression — mygreatlearning.com.* `https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/`. [Accessed 01-11-2023].

7 Appendix

Table 4: Full List of Input Features

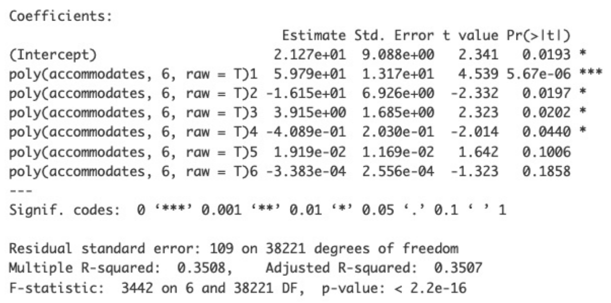
Feature	Description
log_price	price in log scale
price	price of listing
property_type	factor; type of property
room_type	factor; type of room
accommodates	maximum number of occupants
bathrooms	number of bathrooms in listing
bed_type	factor; type of bed in listing
cancellation_policy	factor; strict, moderate, flexible
cleaning_fee	boolean indicating whether a cleaning fee is charged
city	factor; city of listing
first_review	date of first review
host_has_profile_pic	boolean indicating whether host has a profile picture
host_identity_verified	boolean indicating whether host identity is verified
host_response_rate	numeric; 0-100
host_since	date of host registration on site
instant_bookable	boolean indicating whether listing can be instantly booked
last_review	date of most recent review
latitude	numeric; latitude measurement of listing
longitude	numeric; longitude measurement of listing
number_of_reviews	number of reviews for the listing at the time of survey
review_scores_rating	numeric; 0-100
bedrooms	number of bedrooms in the listing
beds	number of beds in the listing



```
Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)                     -1.234e+04  4.592e+02 -26.869  < 2e-16 ***
accommodates                     1.490e+01  4.369e-01  34.106  < 2e-16 ***
bathrooms                        6.099e+01  1.044e+00  58.399  < 2e-16 ***
room_typePrivate room           -6.910e+01  1.185e+00 -58.321  < 2e-16 ***
room_typeShared room            -1.025e+02  3.135e+00 -32.705  < 2e-16 ***
cityChicago                     -2.914e+03  8.581e+01 -33.956  < 2e-16 ***
cityDC                          -9.822e+02  3.974e+01 -24.715  < 2e-16 ***
cityLA                          -8.022e+03  2.535e+02 -31.647  < 2e-16 ***
cityNYC                         -4.594e+02  1.909e+01 -24.062  < 2e-16 ***
citySF                          -8.736e+03  2.686e+02 -32.525  < 2e-16 ***
bedrooms                         3.395e+01  8.997e-01  37.732  < 2e-16 ***
longitude                       -1.729e+02  5.160e+00 -33.503  < 2e-16 ***
property_typeBed & Breakfast     7.704e+00  5.692e+00   1.353 0.175913    
property_typeBoat                5.072e+01  1.512e+01   3.355 0.000795 ***
property_typeBoutique hotel      4.014e+00  1.742e+01   0.230 0.817746    
property_typeBungalow            1.916e+00  6.239e+00   0.307 0.758781    
property_typeCabin               5.996e-01  1.456e+01   0.041 0.967163    
property_typeCamper/RV          -2.001e+01  1.303e+01  -1.536 0.124589    
property_typeCastle              5.764e+01  2.768e+01   2.083 0.037298 *  
property_typeCave                2.514e+01  9.174e+01   0.274 0.784041    
property_typeChalet             -2.537e+00  4.105e+01  -0.062 0.950716    
property_typeCondominium         1.476e+01  2.554e+00   5.779 7.55e-09 ***
property_typeDorm               -4.847e+01  1.043e+01  -4.645 3.41e-06 ***
property_typeEarth House         3.561e+01  6.486e+01   0.549 0.583022    
property_typeGuest suite        -5.165e+00  1.070e+01  -0.483 0.629359    
property_typeGuesthouse         -2.670e+00  5.183e+00  -0.515 0.606448    
property_typeHostel             -7.098e+01  1.384e+01  -5.129 2.92e-07 ***
property_typeHouse               2.717e+00  1.280e+00   2.123 0.033798 *  
property_typeHut                 7.380e-01  4.104e+01   0.018 0.985651    
property_typeIn-law             -2.826e+01  1.256e+01  -2.250 0.024476 *  
property_typeIsland              1.128e+02  9.186e+01   1.227 0.219651    
property_typeLoft                4.118e+01  3.520e+00  11.698  < 2e-16 ***
property_typeOther               1.194e+01  5.269e+00   2.266 0.023445 *  
property_typeServiced apartment  6.928e+01  2.453e+01   2.825 0.004735 ** 
property_typeTent               -4.037e+01  2.769e+01  -1.458 0.144883    
property_typeTimeshare           1.045e+02  1.876e+01   5.573 2.52e-08 ***
property_typeTipi                8.761e+01  5.299e+01   1.653 0.098275 .  
property_typeTownhouse          -5.949e+00  3.083e+00  -1.930 0.053606 .  
property_typeTrain               4.782e+01  6.489e+01   0.737 0.461204    
property_typeTreehouse           2.675e+02  5.299e+01   5.048 4.48e-07 ***
property_typeVacation home       3.519e+01  4.589e+01   0.767 0.443090    
property_typeVilla               1.295e+02  9.321e+00  13.899  < 2e-16 ***
property_typeYurt               -2.543e+01  3.748e+01  -0.679 0.497393    
last_review                     -4.232e-02  3.418e-03 -12.381  < 2e-16 ***
review_scores_rating             1.117e+00  6.639e-02  16.818  < 2e-16 ***
cancellation_policymoderate      3.855e-01  1.422e+00   0.271 0.786285    
cancellation_policystrict        6.835e+00  1.342e+00   5.095 3.51e-07 ***
cancellation_policysuper_strict_30  4.516e+01  1.139e+01   3.965 7.37e-05 ***
cancellation_policysuper_strict_60  4.179e+02  3.483e+01  12.001  < 2e-16 ***
beds                            -7.712e+00  6.751e-01 -11.424  < 2e-16 ***
number_of_reviews               -1.409e-01  1.383e-02 -10.182  < 2e-16 ***
first_review                    -7.260e-03  1.221e-03  -5.945 2.79e-09 ***
cleaning_feeTRUE                -8.296e+00  1.299e+00  -6.388 1.70e-10 ***
host_has_profile_picTRUE        -4.615e+01  1.187e+01  -3.889 0.000101 ***
latitude                         2.106e+01  6.855e+00   3.072 0.002129 ** 
instant_bookableTRUE            -3.157e+00  1.088e+00  -2.901 0.003728 ** 
host_response_rate              -5.246e-02  3.674e-02  -1.428 0.153393    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 91.71 on 38171 degrees of freedom
Multiple R-squared:  0.5413,	Adjusted R-squared:  0.5407 
F-statistic: 804.5 on 56 and 38171 DF,  p-value: < 2.2e-16
```

Figure J: Forward Selection Model Summary



```
Coefficients:
                               Estimate Std. Error t value Pr(>|t|)    
(Intercept)                   2.127e+01  9.088e+00   2.341   0.0193 *  
poly(accommodates, 6, raw = T)1  5.979e+01  1.317e+01   4.539 5.67e-06 ***
poly(accommodates, 6, raw = T)2 -1.615e+01  6.926e+00  -2.332   0.0197 *  
poly(accommodates, 6, raw = T)3  3.915e+00  1.685e+00   2.323   0.0202 *  
poly(accommodates, 6, raw = T)4 -4.089e-01  2.030e-01  -2.014   0.0440 *  
poly(accommodates, 6, raw = T)5  1.919e-02  1.169e-02   1.642   0.1006    
poly(accommodates, 6, raw = T)6 -3.383e-04  2.556e-04  -1.323   0.1858    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 109 on 38221 degrees of freedom
Multiple R-squared:  0.3508,	Adjusted R-squared:  0.3507 
F-statistic:  3442 on 6 and 38221 DF,  p-value: < 2.2e-16
```

Figure K: Polynomial Regression Model Summary