

Predicting Austin Airbnb Listing Survival Using Machine Learning

December 1, 2025

1 Introduction

Every year, millions of Americans travel across the country in search of new experiences, such as exploring new cities, visiting friends and family, or just escaping for a weekend. In the past few decades, these travelers have increasingly turned to non-traditional hotel housing, looking to Short-Term Rentals (STRs) for places to stay. These STRs offer the benefit of unique styles, local neighborhoods, and places that feel more personal. During the last decade, Airbnb has solidified itself as a platform for travelers to find these short term rentals, seeing an increase from 52 million bookings in 2016 to 491 million in 2024 (Curry, 2025). The number of listings has also increased with this demand, with the count moving from 2 million in 2016 to 8 million in 2024 (Kumar, 2025). However, hosts are forced to make a business decision after each stay: should they continue running their listing, or should remove it from the platform altogether?

Not every listing survives. We define *survival* as the persistence of a listing, while *churn* refers to the closing of one. Some listings survive for years or decades, while others churn after only a few months. For example, a recent STR study showed that 79% of listings survive more than one year, while only 62% of listings survive more than two years (Lighthouse, 2023).

Can machine learning predict whether a listing survives or churns? Can we identify the factors that increase survival versus those that cause churn? This paper uses machine learning techniques to predict Airbnb survival in Austin, Texas. By identifying the features that influence listing survival the most, the analysis provides a better look at which listings persist and why.

2 Research Background

The increase in peer-to-peer rental platforms like Airbnb has changed the STR market around the world. With this, there has been an expansion of academic research into Airbnb and other STRs, with a large part of it focused primarily on the operational practices (e.g., trust, pricing strategy, amenities) and impacts on other urban tourism and housing markets (Ding et al., 2023). For instance, studies of major U.S. cities have shown that Airbnb activity can influence housing pressure in the form of rent gap, as a New York City study shows the Airbnb-induced gentrification of different neighborhoods (Wachsmuth & Weisler, 2018). Similarly, there has been extensive research on the drivers of Airbnb prices, like a large, multi-city pricing analysis by Wang and Nicolau (2017) that found attributes such as being a superhost (one with ≥ 10 reservations, $\geq 90\%$ response rate, $< 1\%$ cancellation rate, and

≥ 4.8 overall rating (Airbnb, n.d.)), the number of properties a host had, and whether they had a profile picture significantly influence nightly rates across major Airbnb markets.

Despite the vast research on different sections of the Airbnb market, there has been less work on listing longevity or survival, and why some listings stay active over time while others leave the platform. In existing research, Choi and Won (2023) studied the influence of factors like unit type, location, and the number of bedrooms on STR longevity in both urban and rural markets. They found that rural STRs have a better chance at survival than similar urban listings, but that STRs with the main purpose of rental have a better survival probability than "Occasional STRs", independent of location. Similarly, a Shanghai-based study examining Airbnb performance during COVID-19 found a decline in the overall probability of survival from 56.84% to 50.08% (Hu et al., 2024). Interestingly, they showed that trust factors, like high response rates and review ratings, could partially mitigate the effect of something like a pandemic. They also used a variety of predictive models to find key determinants of listing survival in Shanghai, noting that a listing's age and local COVID-19 incidence were the strongest indicators in forecasting a listing's success.

While the general research on Airbnb, along with the survival studies, provide good insights on host performance and listing longevity, there are several important gaps in the existing literature. Most of the work examines non-U.S. markets or analyzes performance outcomes like occupancy and pricing. Because of this, we know relatively little about the determinants of listing survival in major U.S. cities, including Austin.

3 Data and Materials

3.1 Dataset Descriptions

This study uses data in the form of two snapshots from Austin, Texas. The more recent snapshot is a December 2024 dataset from InsideAirbnb, a website that aggregates data directly from Airbnb's public facing website and provides consistency across cities around the world (Inside Airbnb, 2024). The earlier snapshot is from a March 2023 public Kaggle dataset that is taken directly from InsideAirbnb, but is more accessible than the historical InsideAirbnb data (Banachewicz, 2023). The data from both sources contains information on listing characteristics, host attributes, pricing, reviews, and availability, and together the datasets provide a 21-month time interval. The older dataset contains 14,369 active listings, while the newer one contains 15,501 active listings. Using the listing ID as a unique identifier common to both datasets, the two snapshots allow analysis of which listings survived and which churned between 2023 and 2024.

3.2 Feature Construction and Preprocessing

In order to model listing survival, features were extracted from the 2023 snapshot prior to labeling listings as survived or churned. The raw data included 75 columns, many of which were unsuitable and incomplete for predictive modeling. To ensure a consistent dataset for our modeling, we selected a subset of features that better reflected host activity, listing characteristics, amenities, pricing, and demand. For numeric features like price, minimum nights, maximum nights, and reviews per month, we clipped and cleaned them to remove extreme outliers (e.g. >30 minimum nights, prices above the 99th percentile) which prevented model distortion while keeping meaningful variation among active listings. Host activity features, which included the length of a host’s tenure on Airbnb, their number of listings, and super-host status were kept and converted to numerical values, with boolean fields expressed as 0/1 values.

Attributes that represent key amenities such as WiFi, air conditioning, heating, and kitchen access were extracted from the raw amenities string and converted to binary (0/1) indicators. The `property_type` variable was encoded into three categories: `is_entire_place`, `is_private_room`, and `is_hotel_room`. Because there were 25+ types of properties, one-hot encoding types with very few occurrences (e.g., Earthen Home, Casa Particular, Tent) would cause sparsity and exploding dimensionality with features that have little predictive value. We simplified the review-related variables by keeping only the overall review score to avoid multicollinearity with subcategory ratings. Also, as shown in Fig. 1, the distribution of surviving and churning listings is not strongly influenced by zip code, so we removed the zip code feature from the dataset. Finally, all NaN numeric fields were populated with the median of that category, as shown by Thomas et al. (2020). After completing these steps, the data had strictly numeric values and no missing values, giving us a fully processed dataset ready for model survival prediction.

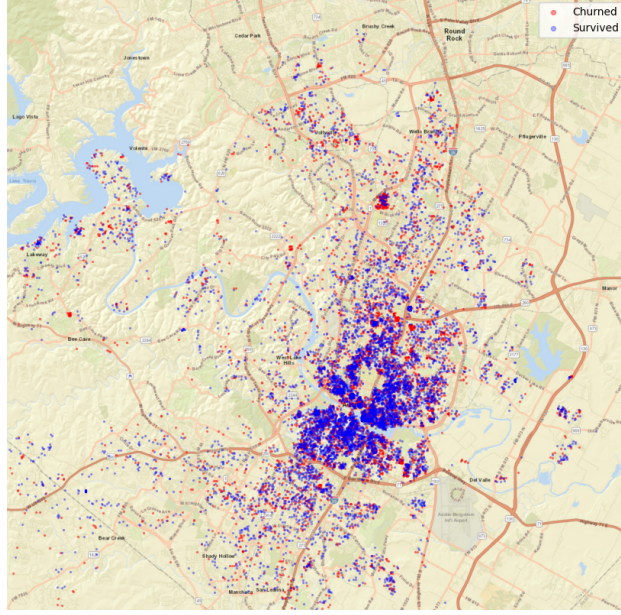


Figure 1: Distribution of survivors (blue) and churners (red) in the greater Austin area.

Before modeling, we examined potential remaining multicollinearity across the features using a correlation heatmap from the pandas library. The heatmap in Figure 2 shows that most features are not strongly correlated, with the strongest relationships between variables that represent a listing's size (`accommodates`, `beds`, `bathrooms`, `bedrooms`), as well as a moderate correlation between these variables and `price`. These correlations are expected, as they show overlapping aspects of capacity, and generally a larger listing will cost more. The slight correlation between a listing being an entire place and these capacity features also makes sense, as you would likely need a full property to hold more people.

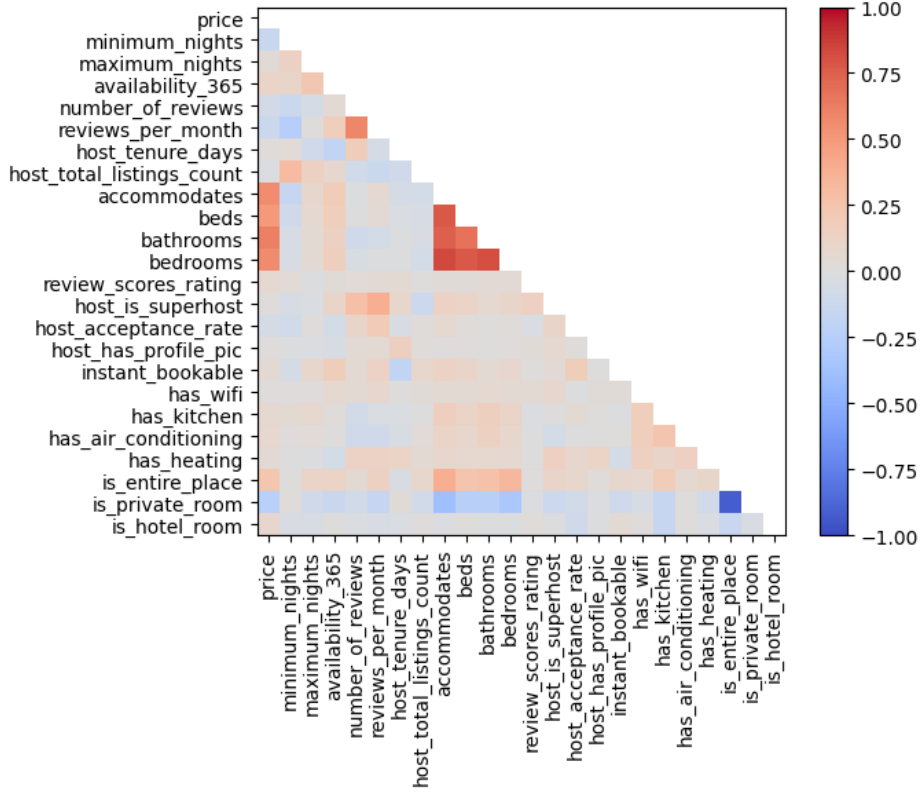


Figure 2: Correlation heatmap of all features in the dataset.

For continuous-continuous correlation, the pandas *corr()* method uses the Pearson correlation coefficient by default (pandas Development Team, 2024). For binary-continuous correlation, the point-biserial correlation coefficient is used, but since our binary variable is coded as 0/1, point-biserial is equivalent to the Pearson correlation coefficient (Field, 2017). Similarly, for binary-binary pairs, Pearson’s correlation formula reduces to the ϕ coefficient, which is standard for measuring association between two dichotomous variables (Duan et al., 2014).

3.3 Survival Label Definition

To determine whether an Airbnb listing survived between the 2023 and 2024 snapshots, we used the listing ID as the identifier across the datasets. Because they were both generated based on Airbnb’s public facing data, listing IDs remain the same over time. A listing was labeled as survived if its ID was present in both snapshots, which indicated that it remained active on Airbnb through the 21-month period. On the other hand, if a listing was in the 2023 dataset but not in the 2024 dataset, it was considered churned. More formally, for each

listing ID i ,

$$\text{survived}_i = \begin{cases} 1, & \text{if listing ID } i \text{ exists in both the 2023 and 2024 datasets,} \\ 0, & \text{if listing ID } i \text{ exists in 2023 dataset but not in 2024 dataset.} \end{cases}$$

For modeling, only listings from the 2023 dataset were included so that survival could be evaluated consistently. This ensured that the prediction task focused only on listings that had a chance to survive or churn. In Austin, 63% of listings that were active in March 2023 remained active in December 2024, while 37% churned.

4 Methods

4.1 Problem Formulation

One of the main goals of this paper is to predict whether an Airbnb listing that is active in March 2023 will still be active in December 2024. We formulate this as a binary classification, where each listing can be represented by a feature vector X_i that contains the attributes extracted from the 2023 dataset described in Section 3.2. The true outcome variable, survived_i , is the survival label from Section 3.3. The model learns the conditional probability

$$\hat{y}_i = P(\text{survived}_i = 1 \mid X_i),$$

which allows us to classify listings as survived or churned based on a threshold.

To address this problem, we used three supervised machine learning algorithms: Logistic Regression, Random Forest, and eXtreme Gradient Boosting (XGBoost). Logistic Regression is an interpretable baseline that shows which features increase and decrease the probability of survival. However, it cannot capture non-linear relationships or interactions between features unless they are manually engineered, which hinders its predictive performance on complex tabular data. On the other hand, ensemble tree methods like Random Forest and gradient-boosted trees can handle non-linear relationships and typically perform better than Logistic Regression on tabular data (Caruana & Niculescu-Mizil, 2006). XGBoost, which is based on gradient-boosted trees, also consistently outperforms Logistic Regression across multiple evaluation metrics, as Yarmohammadtoosky and Attota (2024) showed in a study.

We intentionally did not include a feed-forward neural network in our model pipeline. While neural networks are strong for image, text, and unstructured data with many dimensions, they often under-perform tree ensemble models like XGBoost on tabular data and require significantly more tuning (Shwartz-Ziv & Armon, 2022).

4.2 Models

4.2.1 Logistic Regression

Logistic Regression stands as the baseline model for this study. It models the probability that an Airbnb listing survives as a sigmoid function of a linear combination of features (Molnar, 2025):

$$P(Y^{(i)} = 1) = \text{logistic}(x^{(i)T}\beta) = \frac{1}{1 + \exp\left(-(\beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_p x_p^{(i)})\right)}.$$

The logistic function converts values into a probability between 0 and 1, which makes it easy to do a binary classification task like survival versus churn. In other words, it calculates a value based on the attributes of each listing and then uses the sigmoid curve to determine how likely it is that the listing remains active in 2024. We use the standard 0.5 threshold for our analysis, where probabilities above 0.5 are classified as survived and those below 0.5 are classified as churned. Also, because the classes in the dataset are imbalanced (37% churn, 63% survive), the Logistic Regression model was trained with the option `class_weight='balanced'`, which prevents the model from directly favoring the majority class.

4.2.2 Random Forest

Random Forest is an ensemble tree model that combines predictions from many decision trees trained on bootstrapped samples of the original data. Each tree gets a slightly different set of rows and features, and makes a prediction from that part of the data (Breiman, 2001). Since we are doing binary classification, the aggregation of all trees is done based on majority voting (1 for survived, 0 for churned). Similarly, Random Forest has been shown to deal well with the non-linear relationships that Logistic Regression struggles with, particularly in large, complex datasets like Airbnb listings (Pittala et al., 2024). Random Forest also provides built-in feature importance scores, which allows us to look at how individual listing characteristics affect survival prediction. As with Logistic Regression, we use `class_weight='balanced'` to help with the uneven distribution of the original data.

4.2.3 XGBoost

The third model we used for predicting listing survival is eXtreme Gradient Boosting, or XGBoost. Similar to Random Forest, it is based on decision trees, but builds the trees one at a time instead of averaging multiple independent trees. In gradient boosting, each new tree attempts to correct the mistakes of the previous tree by looking at the residual between

the predicted and actual value, which essentially fits each new tree to the gradient of the loss function (Clark & Lee, n.d.). Formally, it represents the prediction as an additive model

$$\hat{y}_i = \sum_{k=1}^K f_k(X_i),$$

where each f_k is a smaller decision tree that improves the model each iteration.

XGBoost optimizes this boosting scheme by adding a regularization term, which prevents the trees from becoming too tall or specific to the training (overfitting). As shown in the original XGBoost paper, it combines the normal gradient boosting prediction loss with a penalty for tree complexity (Chen & Guestrin, 2016):

$$\mathcal{L} = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k),$$

where the first term is a loss function for the difference between the prediction \hat{y}_i and the actual y_i . The second term Ω is the penalty for the model’s complexity and is one of the main reasons that overfitting is avoided, as more simple trees are often selected. Because of the sequential training and complexity control, XGBoost does well at modeling nonlinear relationships and interactions, which is important for our partially collinear Airbnb data.

4.3 Evaluation Metrics

To evaluate the performance of each of our three models, we use five standard binary classification metrics: accuracy, precision, recall, F1 score, and area under the receiver-operating characteristic curve (ROC-AUC). Since our data is slightly imbalanced, it is helpful to have separate metrics for different aspects of prediction.

First, accuracy measures the proportion of total predictions that the model gets correct (Google Developers, n.d.). In our case, this is the proportion of listings that were correctly predicted as survived or churned. However, since the data does have an imbalance in favor of survived listings, the model could seem more accurate just by predicting all listings as survived. Next, precision measures how often each model’s positive predictions are correct. In our case, this is the proportion of listings classified as "survived" that actually survived. Recall, or the true positive rate, is the proportion of surviving listings that were correctly identified. It is important to note that precision and recall have a tradeoff, where increasing one decreases the other, and vice versa (Upadhyay, 2020).

The F1 score is especially important for our analysis since it balances precision and recall

into one number, which provides a more stable performance metric when false positives and false negatives have ramifications (Lipton et al., 2014). For instance, misclassifying a survivor may cause a listing owner to neglect a major problem, while misclassifying a churning may cause unnecessary distress. Finally, ROC-AUC classifies how well each model separates survivors and churners across all thresholds, rather than just the fixed cutoff of 0.5. A higher AUC shows that the model consistently assigns higher survival probabilities to survivors than to churners, telling us how well the model distinguishes the two classes overall. In combination, these metrics give us a complete idea of model performance in both threshold-dependent and threshold-independent settings.

5 Results

5.1 Model Performance

All three models were trained on the 2023 dataset and tested on a 20% split of the cleaned data, which was held out of training. Table 1 shows the accuracy, precision, recall, F1 score, and ROC-AUC for each model. Overall, both tree-based models performed substantially better than the baseline Logistic Regression, but there were only relatively small differences between Random Forest and XGBoost.

Table 1: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1 Score	ROC-AUC
Logistic Regression	0.650	0.752	0.669	0.708	0.696
Random Forest	0.745	0.745	0.907	0.818	0.809
XGBoost	0.746	0.768	0.859	0.811	0.803

Logistic Regression had an accuracy of 0.650 and F1 score of 0.708, which shows that a linear model can detect some structure in the data. However, it had a recall of only 0.669, meaning that it misses a substantial of listings that actually survived.

Overall, Random Forest performed the best, with the highest recall (0.907), F1 score (0.818), and ROC-AUC (0.809). The high recall indicates that that it correctly found the majority of listings that stayed active in 2024, and the strong ROC-AUC shows a very good separation of survivors, independent of probability thresholds. We do note that Random Forest slightly under-performs Logistic Regression in precision, which reflects the tradeoff where Random Forest assumes more false positives in order to have higher recall.

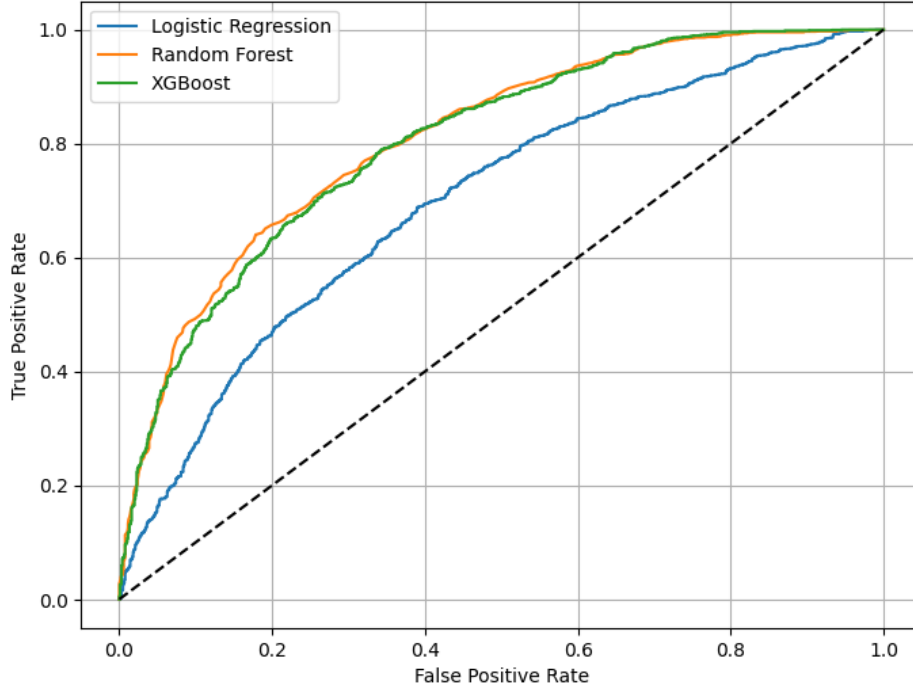


Figure 3: ROC curves for all models.

Lastly, XGBoost slightly outperformed Random Forest in a few metrics, but fell short in others. It achieved the highest accuracy and precision, while barely under-performing Random Forest in F1 score and ROC-AUC (0.007 and 0.006 differences, respectively). On the other hand, it had a recall score of 0.048 less than Random Forest, meaning it could not correctly identify as many surviving listings.

Figure 3 shows the ROC curves for all three models. These curves give a visual comparison of false-positive rates across different thresholds. As with other metrics, the ensemble models clearly outperformed Logistic Regression with ROC. Random Forest and XGBoost have nearly identical curves, but Random Forest was slightly higher between false-positive rates of about 0.40-0.65.

Across every metric except precision, Random Forest and XGBoost substantially outperformed Logistic Regression. Random Forest scored highest in most categories, but XGBoost remained close in prediction performance.

5.2 Feature Importance

To understand which host and listing features most influenced XGBoost’s survival predictions, we plotted the feature importance scores using the model’s learned statistics. Figure 4 shows how much the top 10 features contributed to reducing loss during XGBoost training.

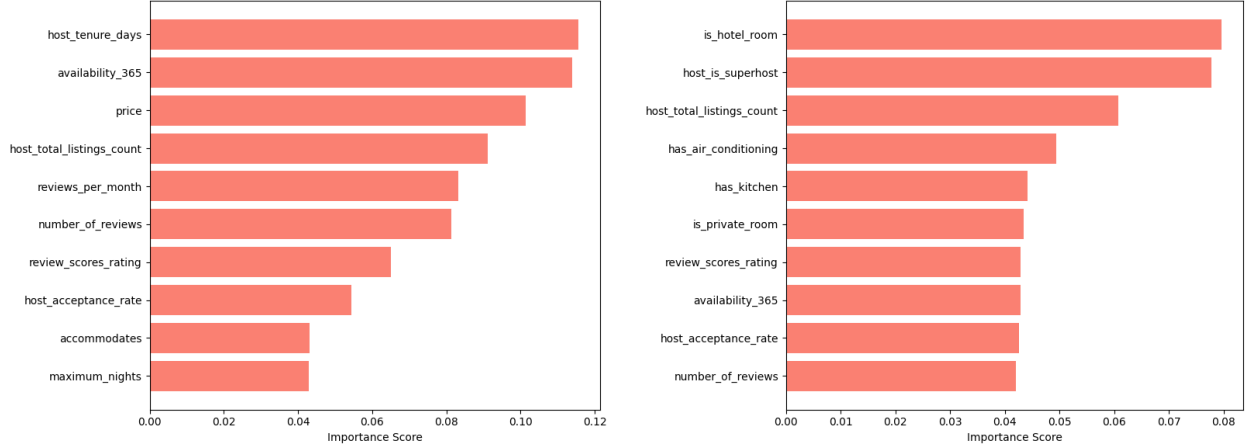


Figure 4: Comparison of feature importance for top 10 features of Random Forest (left) and XGBoost (right).

For Random Forest, we see that the length of a host’s tenure on Airbnb (`host_tenure_days`) and the availability in the next year (`availability_365`) had the most effect on the model’s prediction, with feature importance scores of around 0.12. Price and the host’s number of listings (`host_total_listing_count`) were the next biggest factors, coming in at 0.10 and 0.09, respectively. On the other hand, XGBoost’s survival prediction was most influenced by whether a listing was a hotel room (`is_hotel_room`), and whether the host is a superhost (`host_is_superhost`), with importance scores of around 0.08. XGBoost was also strongly influenced by the number of listings a host has (`host_listings_count`), as well as amenities like air conditioning (`has_air_conditioning`) and kitchen (`has_kitchen`).

While these importance values show which features were most influential overall, they do not show whether lower or higher values of a feature increase or decrease the probability of survival. Since XGBoost performs competitively and better supports detailed interpretability tools, we will use it for analyzing how features influence survival prediction. In order to understand the direction and magnitude of feature influence, we make use of Shapley Additive exPlanations (SHAP) values. These values hold the importance that a model gives to each feature (Marcílio & Eler, 2020). The SHAP plot in Figure 5 shows the overall importance ordering as well as how the value of each feature affects XGBoost’s survival predictions across all listings.

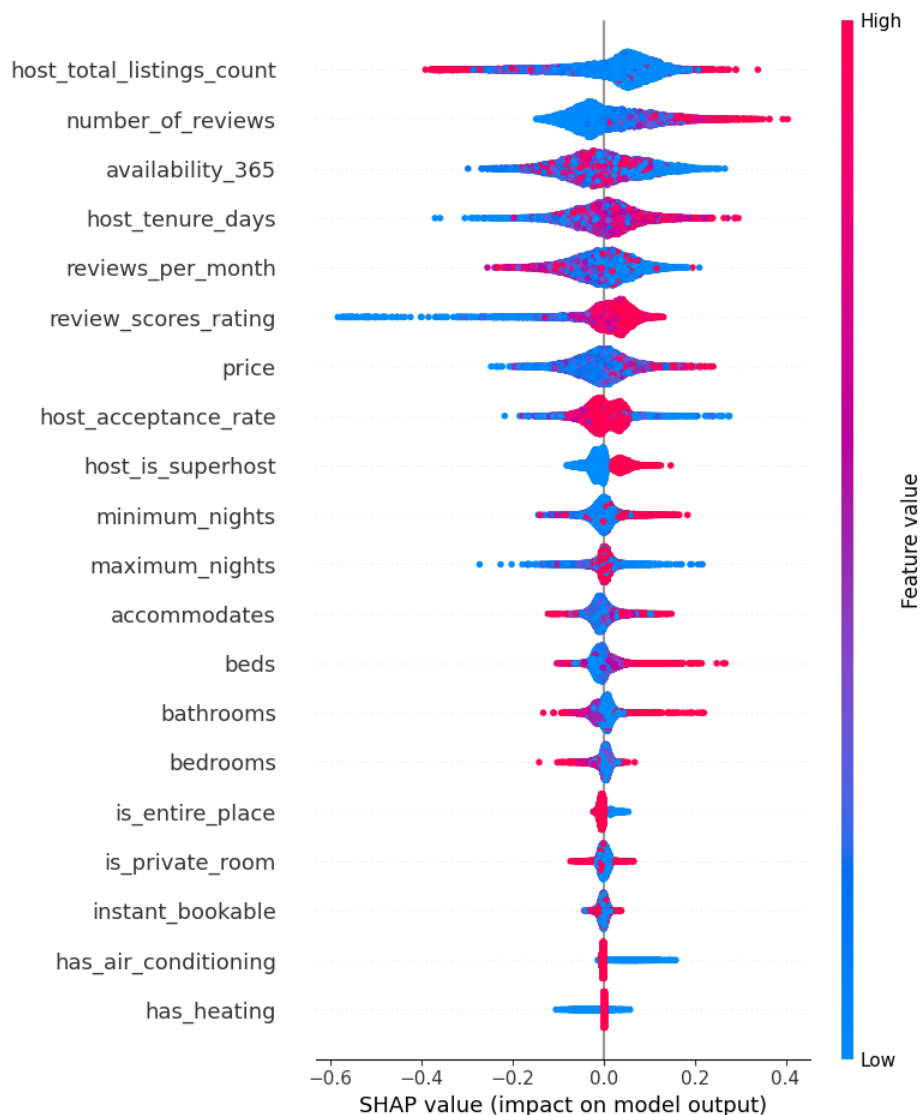


Figure 5: SHAP values ordered by influence (descending).

To interpret Figure 4, note that the features are shown in importance-descending order. Similarly, for each feature, red dots represent a high value for that feature, blue dots a low value, and position along the x-axis shows whether they influenced the model to predict churn (negative SHAP value) or survival (positive SHAP value). If the SHAP value is around 0.0, it means that when the model was conditioned on that feature, it did not change the expected prediction (Lundberg & Lee, 2017). A host having a high total number of listings can cause survival or churn, as we will discuss in the next section. A high number of reviews, long host Airbnb tenure, high review scores, and the host being a superhost all generally influenced the model to predict survival. On the other hand, number of reviews per month and price are almost inverses of each other, with the majority of their values not influencing

survival prediction but some of the high values causing the model to predict churn and survive, respectively. Other features, like a listing being an entire place, private room, or instant-bookable have tighter SHAP distributions, indicating weaker overall influence.

As a complement to the SHAP plot, we created partial dependence plots (PDPs) for several of the highest-ranked features in order to show how changing each variable independently, while all others stay fixed, affects the average predicted survival probability. Figure 6 shows PDPs for features including the number of listings a host has, total reviews, a host’s tenure and acceptance rate on Airbnb, the price, and the minimum nights for a stay. The black lines on the x-axis represent scikit-learn sampled listings that have that feature value.

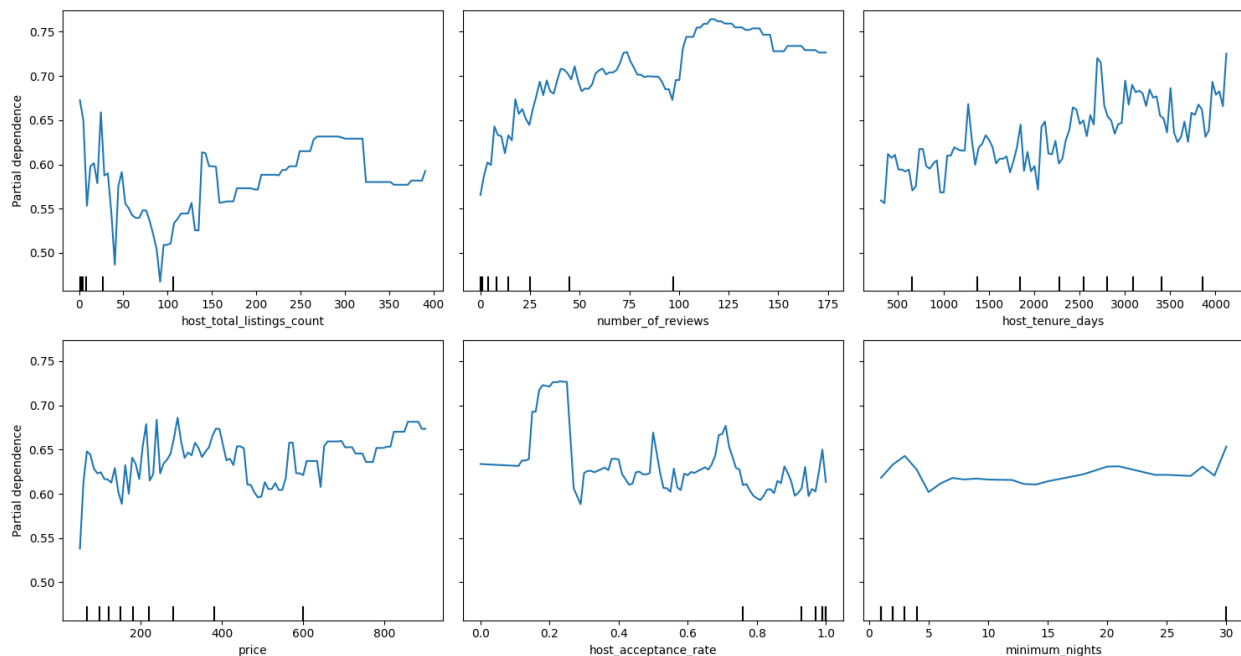


Figure 6: Partial dependence plots showing predicted survival probability across selected features.

In Figure 6, we see that survival probabilities for listings whose owners have few properties is slightly higher than those whose owners have many (>150) listings, and much higher than ones where the owner has between 10 and 150 listings. In terms of reviews and a host’s tenure on Airbnb, the PDP plots show that more total reviews and a longer tenure provide a generally increasing linear survival probability. Price also has a small positive trend, indicating higher-priced listings survive a bit more often. Acceptance rate and minimum night PDPs are fairly horizontal, meaning that these features do not strongly influence predicted survival. Although there are some fluctuations, the overall trend shows that decreasing or increasing these values does not consistently lower or raise survival probability.

Altogether, importance rankings, SHAP values, and PDPs give a detailed idea of how

XGBoost uses different host and listing characteristics when predicting survival. These results show which features the model relies on most, along with how these features change the prediction on average.

6 Discussion

The results from this study show that tree ensemble models can accurately predict whether an Austin Airbnb listing will survive over a 21-month period. Because Random Forest and XGBoost strongly outperformed Logistic Regression, we can tell that there are non-linear relationships between listing features, host behavior, and survival outcomes. This means that survival is not tied to any single individual feature, but instead how these features work together in cultivating a listing. In practice, every Airbnb host has different listings and acts differently on the platform, and these kinds of patterns are handled much better by tree-based models than a linear model.

One of the clearest pictures in the analysis is that experience matters a lot. The feature importance charts for both Random Forest and XGBoost show that these qualities (in the form of tenure, superhost, total listings, etc.) drastically impact the models predictions of survival. This is expected, as a host who has had more visitors knows what to expect and how to build a good experience.

The SHAP chart gives us one of the best ideas of how features influence the model's prediction. For one, a host who has many listings may be part a professional operation that stays incredibly active but does not hesitate to close a listing if it is struggling, which is why we see red points on both side of the SHAP scale. The red values for number of reviews, host tenure, and superhost being positive SHAP values are also intuitive, as this means a host and listing has been around for a long time and stays active on the platform. Also, availability over the next 365 days having a distribution of high values with negative SHAP values can be seen as less early bookings corresponding to less profit and higher churn probability. Lastly, although the distribution of price is mostly in the middle, the slight pattern of higher prices and higher survival prediction could come from a few things. Properties may be maintained more consistently, can host groups or families, or are treated as second homes for people with higher resources, which would all likely increase survival probability. On the other side, lower-priced listings may come from new hosts or hosts who recently lowered the price in order to attract more visitors, both of which indicate volatility.

Turning to the Partial Dependency Plots (PDPs), we can speculate from the total listings graph that hosts with one listing are more personally involved with their property. We can also reason that large operators generally see success because they treat it like a business

and have established systems, but mid-sized hosts might be experimenting with multiple listings but end up failing to keep them afloat. On the other hand, the slightly positive-trending price PDP may also reflect the model’s learned association between higher prices and other features, like strong reviews. Number of reviews and host tenure days again follow our analysis that experience matters. The more reviews a listing has and the longer the host has been active on Airbnb, the more likely it is to survive, which follows the idea that longer-standing hosts who have already built experience and a reputation often keep the listing running.

7 Conclusion

This study was constructed to predict whether an Airbnb listing in Austin would survive over a 21-month period and what host and listing characteristics contribute the most to this survival. We use two snapshots of Austin Airbnb data: a December 2024 InsideAirbnb Austin dataset for finding survivors, and a March 2023 Kaggle Austin dataset (which was originally from InsideAirbnb) for modeling. We trained Logistic Regression, Random Forest, and XGBoost models on a cleaned version of the 2023 dataset to classify each listing as surviving or churning by December 2024.

The first finding was that the tree ensemble models (Random Forest and XGBoost) significantly outperformed Logistic Regression, which indicates that the survival of an Airbnb listing is dependent on non-linear relationships and interactions between features, rather than any one feature alone. Both tree models had an accuracy of 0.75, F1 score of 0.815, and area under the receiver operating characteristic curve of 0.805, indicating that they distinguished survivors and churners well enough to generalize unseen data.

Our second discovery was that host-experience characteristics played the biggest role in predicting survival. Features such as the number of listings a host has, the length of their tenure, the total number of reviews on a listing, and the listing’s average review score were the strongest effectors of survival. A SHAP plot and PDPs helped show that higher values of the latter features actually pushed predictions towards survival. The charts also showed that listings with hosts who have either very few or very many listings are pushed toward survival, while ones with mid-size hosts are pushed towards churn. Listing-specific features like price, amenities, and minimum nights did not have a lot of influence overall, indicating that long-term host behavior and consistency is more important than the physical traits of the listing itself.

Despite the models performing well, there are still several limitations to this study. First, it does not consider higher factors like legal policy shifts or pricing strategy changes. Next, it

does not consider when churned listings exited the market, which prevents us from looking at survival as a continuous event. Finally, it only considers Austin data, while other cities and states in the U.S. may have other regulations that affect the host and listing characteristics.

Future work could expand on this paper by incorporating time-series data, which would allow for a more specific time-to-event analysis with longitudinal survival models (Singer & Willett, 2003). Another next step could be to apply the same modeling pipeline to other U.S. cities that have less Airbnb regulation, such as Milwaukee or Columbus, and determine how the feature importance lists differ (Mathvisor, 2024). These extensions would help determine whether the driving features in Austin survival prediction generalize over time and to other geographic markets.

References

- Airbnb. (n.d.). *What's required to be a superhost*. Retrieved from <https://www.airbnb.com/help/article/829>
- Banachewicz, K. (2023). *Inside airbnb - usa*. Retrieved from <https://www.kaggle.com/datasets/konradb/inside-airbnb-usa>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5-32. doi: doi.org/10.1023/A:1010933404324
- Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. *Association for Computing Machinery*, 23, 161–168. doi: 10.1145/1143844.1143865
- Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (p. 785–794). ACM. doi: 10.1145/2939672.2939785
- Choi, S., & Won, J. (2023). Exploring the survival mechanisms of short-term rentals in virginia: A comparative analysis of rural versus non-rural markets. *Sustainability*, 15, 12651. doi: 10.3390/su151612651
- Clark, B., & Lee, F. (n.d.). *What is gradient boosting?* Retrieved from <https://www.ibm.com/think/topics/gradient-boosting>
- Curry, D. (2025). *Airbnb revenue and usage statistics (2025)*. Retrieved from <https://www.businessofapps.com/data/airbnb-statistics/>
- Ding, K., Niu, Y., & Choo, W. C. (2023). The evolution of airbnb research: A systematic literature review using structural topic modeling. *Heliyon*, 9, e17090. doi: 10.1016/j.heliyon.2023.e17090
- Duan, L., Street, W. N., Liu, Y., Xu, S., & Wu, B. (2014, September). Selecting the right correlation measure for binary data. *ACM Trans. Knowl. Discov. Data*, 9(2), 1-28. doi: 10.1145/2637484
- Field, A. (2017). *Discovering statistics using ibm spss statistics* (Fifth ed.). Sage Publications.
- Google Developers. (n.d.). *Classification: Accuracy, recall, precision, and related metrics*. Retrieved from <https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall>
- Hu, M., Yang, L., Park, J., & Liu, M. (2024). Survival determinants and prediction for airbnb listings. *International Journal of Hospitality Management*, 128, 104132. doi: 10.1016/j.ijhm.2025.104132
- Inside Airbnb. (2024). *Get the data*. Retrieved from <https://insideairbnb.com/get-the>

-data/

- Kumar, N. (2025). *Airbnb statistics [2025] – users growth data*. Retrieved from <https://www.demandsage.com/airbnb-statistics/>
- Lighthouse. (2023). *2023 ota trends across channels: Airbnb, tripadvisor vrbo*. Retrieved from <https://www.mylighthouse.com/resources/blog/2023-ota-trends-across-channels-airbnb-tripadvisor-vrbo>
- Lipton, Z. C., Elkan, C., & Narayanaswamy, B. (2014). Thresholding classifiers to maximize f1 score. *arXiv*. Retrieved from <https://arxiv.org/abs/1402.1892>
- Lundberg, S., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *arXiv*. Retrieved from <https://arxiv.org/abs/1705.07874>
- Marcílio, W. E., & Eler, D. M. (2020). From explanations to feature selection: assessing shap values as feature selection mechanism. In *2020 33rd sibgrapi conference on graphics, patterns and images (sibgrapi)* (p. 340-347). doi: 10.1109/SIBGRAPI51738.2020.00053
- Mathvisor. (2024). *20 cities with no airbnb regulation*. Retrieved from <https://www.mashvisor.com/blog/cities-no-airbnb-legal-issues/>
- Molnar, C. (2025). *Interpretable machine learning: A guide for making black box models explainable* (3rd ed.).
- pandas Development Team. (2024). *pandas.dataframe.corr* [Computer software manual]. Retrieved from <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html> (Version 2.2.3. Accessed 2025-03-01)
- Pittala, T. S. S. R., Meleti, U. M. R., & Vasireddy, H. (2024). Unveiling patterns in european airbnb prices: A comprehensive analytical study using machine learning techniques. *arXiv*, 2407.01555. Retrieved from <https://arxiv.org/abs/2407.01555>
- Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Elsevier Science Publishers B. V.*, 81, 84–90. doi: 10.1016/j.inffus.2021.11.011
- Singer, J. D., & Willett, J. B. (2003). *Applied longitudinal data analysis: Modeling change and event occurrence* (3rd ed.). Oxford Academic Press. doi: 10.1093/acprof:oso/9780195152968.001.0001
- Thomas, R. M., Bruin, W., Zhutovsky, P., & van Wingen, G. (2020). Chapter 14 - dealing with missing data, small sample sizes, and heterogeneity in machine learning studies of brain disorders. In A. Mechelli & S. Vieira (Eds.), *Machine learning* (p. 249-266). Academic Press. Retrieved from <https://www.sciencedirect.com/science/article/pii/B9780128157398000146> doi: <https://doi.org/10.1016/B978-0-12-815739-8.00014>
- Upadhyay, A. (2020). *Precision/recall tradeoff*. Retrieved from <https://medium.com/>

analytics-vidhya/precision-recall-tradeoff-79e892d43134

- Wachsmuth, D., & Weisler, A. (2018). Airbnb and the rent gap: Gentrification through the sharing economy. *Environnment and Planning A*, 50, 1147-1170. doi: 10.1177/0308518X18778038
- Wang, D., & Nicolau, J. L. (2017). Price determinants of sharing economy based accommodation rental: A study of listings from 33 cities on airbnb.com. *International Journal of Hospitality Management*, 50, 120-131.
- Yarmohammadtoosky, S., & Attota, D. C. (2024). Optimizing fintech marketing: A comparative study of logistic regression and xgboost. *arXiv*, 2412.16333. doi: 10.48550/arXiv.2412.16333