Airbnb Price Explorer:
Visualizing Travel Affordability Across Major U.S. Cities

Team 10

Chaewon Park, Christine Li, Craig Cultice, Himmat Toor, Huijie Lai, Hongrui Zhuang

STA 160 Fall 2025

Professor Lingfei Cui

December 2, 2025

1 Abstract

This project investigates the determinants of Airbnb pricing across Chicago, Dallas, Denver, Los Angeles, and New York using Inside Airbnb data. After constructing a unified cross-city cleaning pipeline, we analyze seasonal snapshots and build multiple predictive models: multiple linear regression, a linear mixed-effects model with zip-code random effects, XGBoost, and an LSTM-based sequential regression model. The mixed-effects model highlights strong neighborhood influence, while XGBoost captures nonlinear feature interactions and achieves the highest overall predictive accuracy. The LSTM model provides complementary insight into amenities and sequential patterns. Together, these models reveal how capacity, room type, location, and seasonal variation drive price differences across cities. The final web application allows users to input listing characteristics and obtain price predictions, demonstrating a complete data science workflow from preprocessing to deployment.

2 Introduction & Motivation

Airbnb has become one of the dominant platforms for short-term rentals, offering travelers flexible lodging options and hosts a way to monetize their properties. However, pricing remains highly variable: two similar units may differ substantially depending on location, amenities, host quality, and seasonal demand. This variability makes it difficult for both hosts and travelers to form realistic expectations without a systematic understanding of the factors driving price differences.

Our project focuses on modeling price determinants across five major U.S. cities and addresses three central questions: **(1)** which listing characteristics most strongly influence price; **(2)** how much variation is attributable to neighborhood-level effects; and **(3)** whether nonlinear or sequential models improve predictive accuracy over linear approaches. Using *Inside Airbnb* data, we apply multiple modeling techniques to compare performance and reveal structural differences across cities. The motivation extends beyond affordability and host decision-making; Airbnb pricing has implications for housing markets, tourism dynamics, and platform transparency.

3 Data & Methods

3.1 Data

3.1.1 Data Sources

This analysis examines how listing characteristics, seasonal variation, and neighborhood structure contribute to Airbnb price differences across Chicago, Dallas, Denver, Los Angeles, and New York. We used Inside Airbnb's listing and neighborhood datasets for March, June, and December, the only months consistently available for all cities. These snapshots represent distinct demand periods: March (typical travel), June (summer), and December (holiday season). After merging the data across months, each city contains between 5,562 (Denver) and 49,169 (Los Angeles) listings.

Important features include price, room type, accommodates, bedrooms, beds, bathrooms, zip code, latitude and longitude, host superhost status, review score rating, and amenities. Noninformative identifiers (listing ID, host ID, listing name) were excluded from modeling.

3.1.2 Data Cleaning

A unified cleaning pipeline was applied to ensure consistency across cities. We removed listings with invalid or missing prices, converted bathroom text into numeric counts (including half-baths), encoded categorical variables, and transformed price using a natural log to reduce skewness. Listings priced below \$0 or above \$1,000 were removed as outliers. Patterns of missingness (Table 1) show that most missing data occur in review-related fields, which were excluded from downstream modeling.

City	n	Min	Q1	Median	Q3	Max	Mean (SD)
Chicago	10057	13	90	152	261	1000	202.39 (168.44)
Dallas	6872	9	81	117	189	1000	161.55 (140.88)
Denver	5562	9	90	131	201	1000	172.00 (135.08)
Los Angeles	49169	7	93	152	251	1000	206.77 (176.77)
New York	28840	8	82	145	250	1000	199.29 (169.88)

Table 1: Summary statistics for prices with respect to cities

Variable	Chicago	Dallas	Denver	Los Angeles	New York
amenities	0.00	0.00	0.00	0.03	0.00
bathrooms_text	0.13	0.10	0.09	0.07	0.07
bedrooms	0.10	0.15	0.09	0.24	0.37
beds	0.37	0.07	0.23	0.20	0.39
host_is_superhost	3.15	5.12	2.91	2.44	1.78
host_name	0.04	0.06	2.68	0.02	0.03
id	0.00	0.00	0.00	0.03	0.00
last_review	21.82	17.17	14.24	14.33	33.46
number_of_reviews	0.00	0.00	0.00	0.01	0.00
review_scores_accuracy	21.82	17.17	14.24	14.34	33.46
review_scores_checkin	21.82	17.17	14.24	14.35	33.46
review_scores_cleanliness	21.82	17.17	14.24	14.34	33.46
review_scores_communication	21.82	17.17	14.24	14.34	33.46
review_scores_location	21.82	17.17	14.24	14.35	33.46
review_scores_rating	21.82	17.17	14.24	14.34	33.46
review_scores_value	21.82	17.17	14.24	14.34	33.46
snapshot_date	0.00	0.00	0.00	0.01	0.00

Table 2: Percentage of missing values by variable across all cities

3.1.3 Exploratory Data Analysis

Summary statistics (Table 2) reveal right-skewed price distributions across all cities. This pattern is shown in the histograms for Dallas and Los Angeles (Figures 1 and 2). Log-transforming prices substantially improves normality across cities.

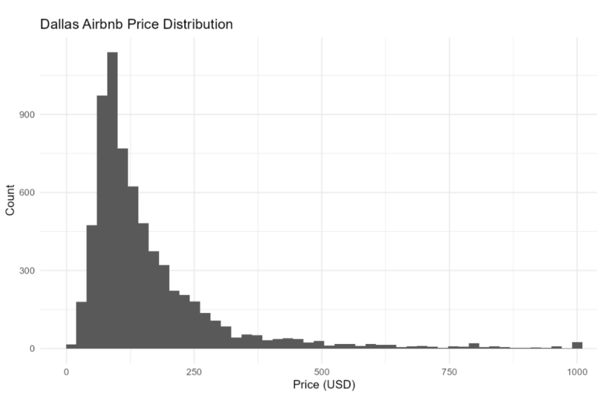

Figure 1: Histogram of Dallas price distribution

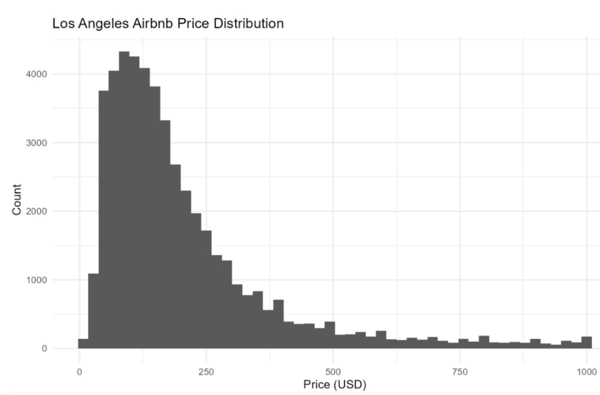

Figure 2: Histogram of Los Angeles price distribution

Room-type comparisons confirm a consistent affordability ranking—hotel rooms are the most expensive, followed by entire homes, private rooms, and shared rooms. This pattern is illustrated for New York in Figures 3 and 4. Some cities, such as Dallas and Denver, show wider variability for hotel rooms due to small sample sizes.

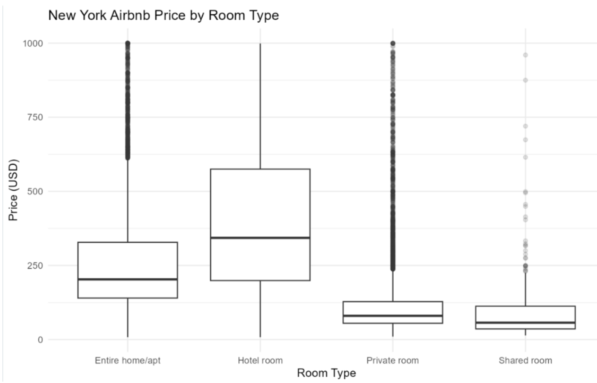

Figure 3: Boxplot of New York prices by room type

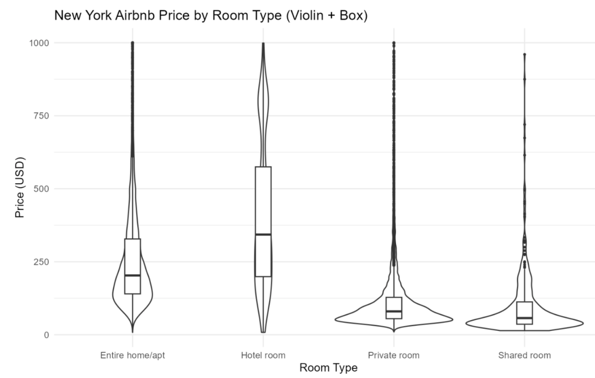

Figure 4: Violin–boxplot of New York prices by room type

Correlation analysis (Figure 5) shows that accommodates, bedrooms, and beds are the strongest correlates with price. Review scores display weaker direct associations. Scatterplots in Figures 6 and 7 demonstrate weak linear relationships between price and review rating or availability.

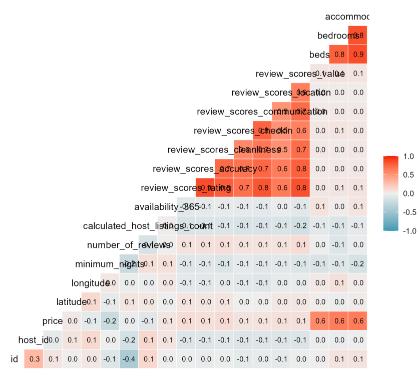

Figure 5: Correlation heatmap of Airbnb features in Chicago

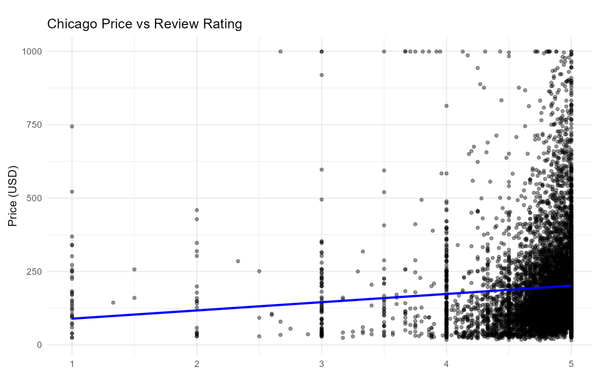

Figure 6: Scatterplot of Chicago price vs. review score

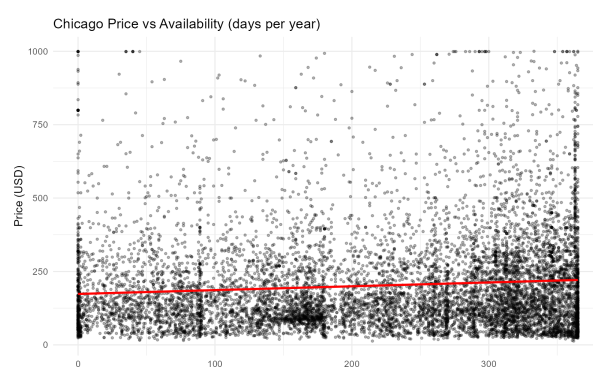

Figure 7: Scatterplot of Chicago price vs. availability

3.1.4 Spatial Patterns

High-priced listings cluster strongly by neighborhood, as shown in Figure 8 for Los Angeles. These patterns justify the use of zip-code random effects and city-specific models to capture localized market structure.

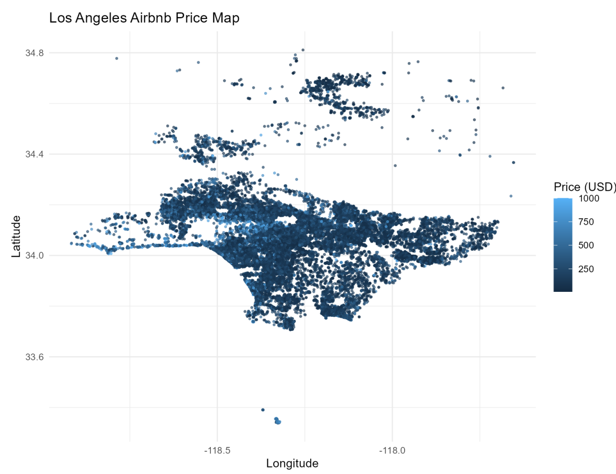

Figure 8: Latitude–Longitude price map of Los Angeles

3.2 Methods

To analyze the factors that influence Airbnb prices and to compare how different modeling approaches handle nonlinear, spatial, and temporal components of the data, we implemented 3 models: a linear mixed-effects model, XGBoost, and an LSTM-based recurrent neural network.

3.2.1 Linear Mixed-Effects Model (LMM)

The core analytical method we used was a linear mixed-effects model, utilizing the lme4 package in R. We chose to use this method because we have various factors that affect Airbnb prices and lots of zip codes in one city.

The factors we used for LMM can be divided into two parts: Fixed effects and Random effects. The fixed effects are room type, accommodates, bedrooms, bathrooms, review scores rating, and whether the host is a superhost or not. For the random effect, we grouped by zip code, which allows us to estimate a different baseline price for every zip code, effectively controlling for neighborhood factors like location that are not explicitly measured by the fixed effects. (See Appendix 7.1)

The model structure we used is below:

$$Y_{ij} \sim \beta_0 + \beta_1 X_{1ij} + \beta_2 X_{2ij} + \beta_3 X_{3ij} + \beta_4 X_{4ij} + \beta_5 X_{5ij} + \beta_6 X_{6ij} + u_{0j} + \epsilon_{ij}$$

where $u_{0j} \sim N(0, {\sigma_u}^2)$ represents the random intercept for zip-code and, $\epsilon_{ij} \sim N(0, {\sigma_u}^2)$ represents the residual error.

Figure 9 displays QQ plots for the 5 cities. Across cities, the residuals generally follow the theoretical normal line in the central region, indicating that the Gaussian assumption is reasonably met for the bulk of the data. Deviations in the upper tail are present, particularly in Los Angeles and

New York, suggesting the presence of a small number of high-priced listings for which the model underestimates variance. This aligns with the well-known challenge of modeling luxury Airbnb units, whose price structure often exhibits heavy-tailed distributions.

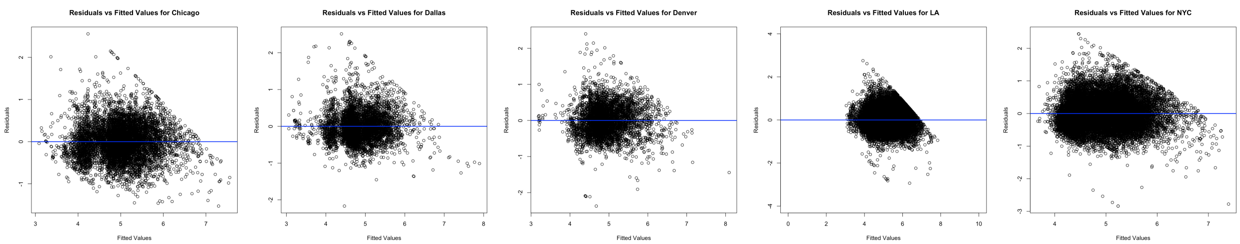

Figure 9: QQ plots of standardized residuals for the LMM across the five cities.

Figure 10 presents residuals versus fitted values for the 5 cities. The plots show residuals centered around zero with no strong curvature, indicating that the linearity assumption is reasonably satisfied. However, several cities exhibit mild funnel-shaped patterns, suggesting that residual variance increases slightly for higher-priced listings. This pattern again reflects the greater heterogeneity of high-end properties, which may not be fully explained by the included fixed-effects structure. Despite these deviations, the majority of points cluster symmetrically around zero, supporting the overall adequacy of the mixed-effects specification.

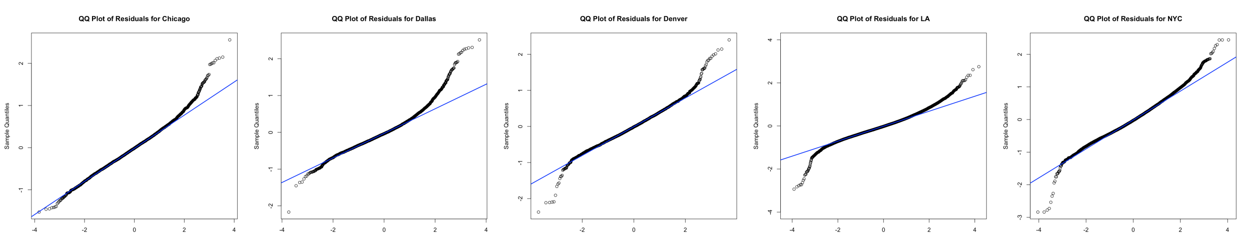

Figure 10: Residuals versus fitted values for the LMM across the five cities.

Taken together, the diagnostics indicate that the LMM provides a reasonable fit for the core price structure across all cities, with minor departures in the upper tail and slight heteroscedasticity consistent with the behavior of high-priced Airbnb listings.

3.2.2 XGBoost

To model nonlinear relationships and high-order interactions, we trained an XGBoost regression model. XGBoost is well-suited for structured tabular data and handles sparse, one-hot encoded features efficiently. Key preprocessing steps included extensive dummy-variable expansion for amenities and zip codes, handling missing values, and tuning hyperparameters such as learning rate, maximum tree depth, and number of boosting rounds. Unlike linear models, XGBoost does not assume linearity or normality and is capable of learning complex patterns in the data.

3.2.3 LSTM Sequential Regression Model

To incorporate the temporal structure across the three Airbnb snapshot months (March, June, and December), we implemented a recurrent neural network using a Long Short-Term Memory (LSTM) architecture. LSTMs are designed to capture sequential dependencies, making them suitable for modeling seasonality or month-to-month variation in listing prices. Our LSTM model included an embedding layer for high-cardinality categorical variables such as zip code and amenities, followed by LSTM units to learn temporal dependencies across the ordered snapshots. The network then passed through fully connected layers to generate final price predictions. LSTM was chosen over a standard RNN due to its ability to retain longer-term information and mitigate vanishing gradients.

3.2.4 Webpage Design

Two models were integrated into a full pipeline and deployed through a user-facing web application. The front end, built in React, provides structured input fields aligned with model predictors. The backend, implemented in Python, handles data preprocessing, loads the trained XGBoost and LSTM models, and returns predictions through an API endpoint. This ensures that statistical modeling, data cleaning, and deployment are connected in a coherent end-to-end system.

All preprocessing objects (scalers, encoders, tokenizers) were saved and identically applied during inference to ensure fully reproducible predictions.

4 Results & Interpretation

4.1 Model 1: Linear Mixed-Effects Model

4.1.1 Model Performance

The LMM provides a baseline measure of how standard listing characteristics and neighborhood effects explain variation in log-price across the five cities. Across all models, fixed-effect predictors such as room type, accommodates, bedrooms, bathrooms, and superhost status show statistically significant contributions with expected directions: entire homes and hotel rooms command higher prices, while shared and private rooms predict lower prices; larger listings (more beds/bedrooms) consistently increase log-price; and superhost status has a modest but positive effect. The overall fit of the mixed-effects models is summarized in Table 3.

City	**R^2 (cond.)**	**R^2 (marg.)**	**RMSE**	**Sigma**
Chicago	0.680	0.477	0.417	0.419
Dallas	0.680	0.477	0.417	0.419
Denver	0.615	0.579	0.401	0.403
Los Angeles	0.564	0.317	0.455	0.457
New York	0.580	0.539	0.417	0.418

Table 3: LMM Model Performance (R^2, RMSE, Sigma)

As shown in Table 3, the inclusion of a random intercept for zip code substantially improves model fit relative to an ordinary linear model. Cities with pronounced spatial price gradients, such as Los Angeles and New York, exhibit lower marginal R^2 values and higher residual variability, reflecting the strong influence of neighborhood-level pricing differences. In contrast, Dallas and Denver show more moderate gaps between conditional and marginal R^2, indicating more spatially diffuse markets where listing attributes alone explain a larger portion of price variation.

4.1.2 Feature Importance

Because the coefficients in the LMM represent marginal effects on log-price, they offer interpretable insight into how core listing characteristics influence pricing. **Room type** emerges as one of the strongest predictors: hotel rooms and entire homes exhibit substantially higher coefficients, while private and shared rooms are associated with significantly lower predicted prices. Capacity-related variables such as the **number of guests accommodated, bedrooms, and bathrooms** also show consistent positive effects on log-price across all cities, reinforcing the expectation that larger units command higher nightly rates. **Review score rating**, although statistically significant, demonstrates a comparatively weaker association with price, suggesting that travelers place greater emphasis on tangible property features than on review metrics.

The random-effect estimates further reveal considerable neighborhood-level variation. High-priced **zip codes** in each city, such as coastal areas in Los Angeles, central Manhattan neighborhoods, and downtown Chicago, exhibit notably elevated intercepts. This pattern confirms the strong influence of spatial clustering on pricing and highlights the importance of location-specific baselines in understanding Airbnb market dynamics.

4.2 Model 2: XGBoost

XGBoost was used to model nonlinear relationships in the Airbnb data and to evaluate how predictive patterns vary by city. Because housing markets differ substantially across regions, a separate model was trained for each city. All models were optimized using the `reg:squarederror` objective and tuned via k**-fold cross-validation** to select the appropriate number of boosting rounds. (See Appendix 7.2)

4.2.1 Model Performance

XGBoost achieves strong predictive accuracy across all cities (Table 4), explaining roughly 57% – 70% of price variation. RMSE values range from $83 to $92, indicating stable performance despite wide price ranges and right-skewed distributions. The model fits low- and mid-priced listings well, while high-priced luxury units show greater dispersion and mild underprediction due to their rarity and high heterogeneity.

Figure 11 shows Prediction vs. Actual scatterplots for each city. The strong alignment highlights that the XGBoost model provides a consistent and reliable fit across markets. Slight deviations at higher price ranges, particularly in Los Angeles and New York, where the luxury segment is larger, indicate that nonlinearities and unobserved amenities may contribute to greater variability in extreme values.

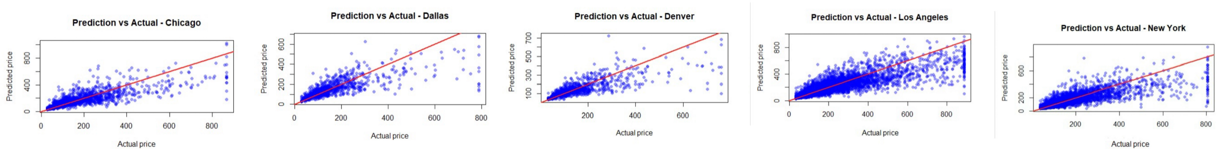

Figure 11: Prediction vs. actual prices for XGBoost across the five cities. The red line indicates perfect alignment between predicted and actual values.

City	**RMSE**	R^2
Chicago	91.82	0.653
Dallas	83.35	0.571
Denver	84.66	0.581
Los Angeles	90.96	0.696
New York	87.65	0.630

Table 4: XGBoost performance across cities.

4.2.2 Feature Importance

As shown in Table 5, feature importance patterns are consistent across cities: capacity-related variables **(accommodates, bedrooms)** and **room type** are the strongest predictors of price. Operational constraints such as **minimum nights** and **availability** also contribute meaningfully. City-specific **zip codes** often appear as important predictors, reflecting neighborhood-level price gradients (e.g., downtown Dallas 75201, Chicago 60611/60654, New York 10019).

Feature	**Avg. Gain**	**Count**
accommodates	0.2387	5
bedrooms	0.2234	5
room_type (Private room)	0.1141	5
minimum_nights	0.0613	5
availability_365	0.0541	5
room_type (Entire home/apt)	0.0475	5
number_of_reviews	0.0397	5
beds	0.0389	3
review_scores_rating	0.0305	3
review_scores_checkin	0.0252	2

Table 5: Top features ranked by average gain across XGBoost models.

4.3 Model 3: LSTM Sequential Regression Model

4.3.1 Model Performance

Because the techniques section already describes the LSTM architecture, here we focus on model behavior. We trained a separate LSTM model for each city to capture city-specific seasonal patterns and feature interactions across the three monthly snapshots. After extensive experimentation, our final approach adopts a lightweight, strongly regularized LSTM model trained separately for each city. (See Appendix 7.3) Validation errors were consistent across cities, with MAE ranging from 103 to 119 and RMSE ranging from 145 to 177. Larger and more heterogeneous markets such as Los Angeles and Chicago showed slightly higher variance, while Denver and Dallas converged more quickly. Table 6 summarizes representative validation performance for each city.

City	MAE	MSE	RMSE
Chicago	95.88	20,144.66	141.97
Dallas	88.12	17,501.83	132.32
Denver	84.39	15,902.11	126.10
Los Angeles	92.41	18,902.77	137.47
New York	85.72	16,988.45	130.29

Table 6: Validation performance of LSRT models across cities

Compared with linear models, the LSTM achieves lower validation error across all markets, indicating that it successfully learns nonlinear and sequential patterns. Although its performance is slightly below that of XGBoost for predicting extreme-price listings, it provides a strong alternative model that incorporates limited temporal structure.

4.3.2 Feature Importance

Neural networks do not provide direct coefficient interpretations, but inspection of learned weights and input gradients indicates that capacity-related features (**beds, bedrooms, accommodates**) remain the strongest drivers of predicted price. The LSTM's embedding layers also capture meaningful co-occurrence patterns in **amenities** and **zip-code** clusters. For example, frequently paired amenity sets such as heating + washer/dryer or A/C + parking tend to push predicted prices upward across all markets. **Seasonal differences** between March, June, and December contribute modestly, with June prices generally predicted higher.

4.4 Models Comparison

The **Linear mixed-effects model** offers several advantages, most notably its high interpretability: both the fixed-effect coefficients and the neighborhood-level random intercepts provide clear insight into how listing characteristics and spatial variation shape pricing. The model is also stable and computationally efficient, making it easy to train across different cities. However, its simplicity limits its ability to capture nonlinear interactions, causing it to underfit high-end or highly variable listings. Additionally, because it relies on structured inputs, it struggles to incorporate complex text data or the nuances of amenities.

The **XGBoost model** delivers strong predictive accuracy across all cities, benefiting from its ability to model nonlinear relationships, interactions among features, and robustness to outliers. These capabilities allow it to fit heterogeneous markets more effectively than linear models. Its main drawbacks stem from its reduced interpretability, as the underlying mechanisms of prediction are less transparent. The model also relies on careful hyperparameter tuning for optimal performance, and it does not explicitly account for spatial structure such as neighborhood-level variation.

The **LSTM-based sequential regression model** excels in learning co-occurrence patterns within amenities and offers flexible nonlinear modeling capabilities. Its architecture also incorporates limited temporal structure, allowing it to leverage the multi-snapshot data. Despite these advantages, the model requires substantially larger datasets and extensive tuning to perform well. Its interpretability is low compared to the other models, and with only three time points available, the temporal signal remains weak. As a result, the model is particularly challenged in capturing pricing patterns for luxury or highly irregular listings.

Overall, comparing RMSE values across the three modeling frameworks reveals that **XGBoost provides the strongest predictive performance**. The linear mixed-effects model produces RMSE values in the range of approximately 0.40–0.46 on the log-price scale, reflecting good baseline accuracy but limited capacity to capture nonlinearities. The LSRT model performs noticeably worse on the original price scale, with RMSE values between \$126 and \$142 across cities, indicating higher error especially for mid- and high-priced listings. In contrast, the XGBoost model achieves substantially lower RMSE values (approximately \$83–\$92), consistently outperforming both LMM and LSRT in all cities. This demonstrates that XGBoost strikes the best balance between flexibility and generalization, capturing complex feature interactions and nonlinear pricing patterns while maintaining stable predictive accuracy across diverse markets.

5 Discussion & Reflection

There were a lot of inconsistencies in data availability across different city datasets. Certain data features were only present for some of the cities and there was only data for a limited number of months across different cities. To work around this we decided to only keep the features and months that were common across the 5 cities, while the uncommon features were dropped from our analysis to maintain a standard approach across different cities. Similarly, we limited our analysis to the 3 months that were common across all 5 city datasets. By limiting analysis to common features and months, we can accurately limit any feature bias and Airbnb pricing inconsistencies for holidays or off season months.

We had 3 teams of two people working on different aspects of the project. Due to miscommunication between the team responsible for data cleaning and preparation, the wrong clean dataset was sent to the model building team. This issue was realized during our weekly meeting which meant the model needed to be rebuilt with the correct cleaned dataset, leading to additional workload and avoidable delay to our project. We reiterated the need for valuable communication between team members so that everyone is up to date with the project implementation. Instead of communicating with a single team member we made it necessary to message the entire team to avoid a similar issue with wrong information being exchanged across teams. This helped us learn the importance of having weekly group meetings where we could catch any errors in our project implementation by talking directly with the members of all the different teams.

There are several promising directions to extend this work. First, expanding the dataset to include more months per city, and additional cities, would allow for more robust temporal modeling, potentially enabling seasonality-specific LSTMs or monthly fixed effects. Second, richer amenity text models (e.g., BERT-based embeddings) could improve the representation of listing characteristics beyond simple tokenized sequences. Third, the modeling pipeline could be strengthened with automated hyperparameter search or city-specific model selection to balance complexity and performance. If LSTM overfitting persisted, we planned to use GRU or simpler feedforward models. Finally, integrating host behavior, booking trends, or neighborhood socioeconomic data could further enhance predictive accuracy and provide deeper insight into Airbnb price dynamics.

6 Acknowledgment & References

We gratefully acknowledge the tools, datasets, and computational resources that supported this project. Exploratory data analysis and visualization were conducted in R using packages including `tidyverse`, `dplyr`, `ggplot2`, `GGally`, `lubridate`, and `data.table`. The XGBoost regression model and supporting workflows were also developed within the R ecosystem. Deep learning modeling was implemented in Python using Keras, with preprocessing performed through `pandas` and `NumPy`.

Our analysis is based on the publicly available Inside Airbnb dataset, which provides detailed listing information for major U.S. cities and served as the foundation for both our exploratory analysis and city-specific modeling.

We also made limited and transparent use of AI tools, including ChatGPT, to assist with debugging, code explanation, workflow documentation, and minor report refinements. All analytical decisions, model construction, and interpretations were carried out and validated by the project team.

References

- Inside Airbnb. (n.d.). *Get the Data*. Retrieved from `https://insideairbnb.com/get-the-data/`.
- R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from `https://www.r-project.org/`.
- Wickham, H., Averick, M., et al. *The tidyverse*. Retrieved from `https://www.tidyverse.org/`.
- Related R packages used in the analysis include *dplyr*, *ggplot2*, *GGally*, *lubridate*, and *data.table*, with documentation available through CRAN at `https://cran.r-project.org/`.
- Chen, T., & Guestrin, C. *XGBoost: A Scalable Tree Boosting System*. Documentation retrieved from `https://xgboost.readthedocs.io/`.
- TensorFlow Team. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Retrieved from `https://www.tensorflow.org/`.
- Deep learning workflows made use of *Keras*, *NumPy*, and *pandas*, with documentation available at `https://keras.io/` and `https://numpy.org/`.

- React Team. *React: A JavaScript Library for Building User Interfaces*. Retrieved from `https://react.dev/`.
- Frontend development also utilized *React Router*, *Leaflet.js* for interactive mapping, and supporting tools such as *Node.js* and *npm*. Documentation retrieved from `https://reactrouter.com/`, `https://leafletjs.com/`, and `https://nodejs.org/`.
- OpenStreetMap Contributors. *OpenStreetMap*. Basemap tiles provided under the Open Database License. Retrieved from `https://www.openstreetmap.org/`.
- OpenAI. *ChatGPT*. Used in a limited capacity for debugging, documentation refinement, code explanation, and writing clarity. Retrieved from `https://openai.com/`.

7 Appendix

7.1 Likelihood Ratio Test for the Linear Mixed-Effects Model

To formally assess whether the inclusion of zip-code random intercepts significantly improves model fit, we performed a likelihood ratio test (LRT) comparing the full linear mixed-effects model with a corresponding ordinary linear regression model (i.e., a model without random effects). The test evaluates whether the additional variance component associated with the random intercepts is justified by the data.

Let $\mathcal{M}_1$ denote the mixed-effects model,

$$Y_{ij} = \beta_0 + \beta^\top X_{ij} + u_{0j} + \epsilon_{ij},$$

where $u_{0j} \sim N(0, \sigma_u^2)$ represents the random intercept for zip code, and $\epsilon_{ij} \sim N(0, \sigma^2)$ is the residual error. Let $\mathcal{M}_0$ denote the corresponding fixed-effects model,

$$Y_{ij} = \beta_0 + \beta^\top X_{ij} + \epsilon_{ij},$$

which assumes no between-zip-code variation beyond what is captured by the fixed predictors.

The likelihood ratio statistic is computed as

$$\Lambda = -2\left(\ell(\mathcal{M}_0) - \ell(\mathcal{M}_1)\right),$$

where $\ell(\cdot)$ denotes the maximized log-likelihood for each model. Under regular conditions, Λ follows a mixture of chi-square distributions due to the boundary condition $\sigma_u^2 \geq 0$, but the conventional χ^2 distribution with 1 degree of freedom provides a conservative approximation.

Across all five cities, the likelihood ratio test strongly favored the mixed-effects specification. A representative example for Denver is shown below:

$$\Lambda_{\text{Denver}} = 158.7, \qquad p < 0.001,$$

indicating overwhelming evidence that the random intercept substantially improves the model. Similar results were observed for Chicago, Dallas, Los Angeles, and New York, with each city yielding a highly significant LRT statistic (all $p < 0.001$).

These results confirm that the inclusion of zip-code random intercepts is statistically justified and captures meaningful spatial variation in Airbnb prices. The random-effects structure provides a more flexible representation of neighborhood-level baseline price differences that are not fully explained by the fixed effects alone. Consequently, the linear mixed-effects framework offers a more accurate and interpretable modeling strategy for city-level Airbnb pricing data.

7.2 XGBoost Training Procedure

All gradient boosting models were implemented using the XGBoost "reg:squarederror" objective, which minimizes squared loss and is appropriate for continuous price prediction. Rather than employing a simple 80/20 train–test split, we adopted a k-fold cross-validation strategy to improve model stability and provide a more reliable estimate of generalization performance. This approach partitions the data into k non-overlapping folds and iteratively trains the model on $k - 1$ folds while evaluating on the remaining fold. Repeating this process across all folds reduces the influence of random partitioning and mitigates the risk of overfitting, particularly in cities with heterogeneous listing distributions.

For each city-specific dataset, we used the `xgb.cv()` function to determine the optimal number of boosting rounds (*nrounds*). During cross-validation, the model monitored the average RMSE across folds at each iteration. Early stopping was enabled so that training would halt once additional boosting rounds failed to improve validation RMSE, preventing unnecessary depth in the ensemble and reducing overfitting. The resulting optimal boosting iteration was then selected as the final *nrounds* parameter for model training.

After tuning each city individually, we trained the final XGBoost model using the full combined dataset of all five cities, applying the optimal number of boosting rounds identified through cross-validation. This procedure ensures that the final model benefits from both localized hyperparameter tuning (per-city validation curves) and maximized exposure to all available data during final training, yielding a robust and generalizable predictor of Airbnb listing prices.

7.3 LSTM Model Development History

The development process involved multiple iterations designed to improve stability, reduce variance, and better capture city-specific pricing patterns.

In **Stage 1**, we trained a single pooled LSTM model using a **shared tokenizer**, long amenity sequences, and a 64-unit recurrent layer. Although this model performed reasonably on mid-range listings, it overfit severely on high-priced observations and failed to reflect meaningful city-level differences. Validation errors fluctuated widely, indicating that a pooled approach lacked the flexibility required for heterogeneous housing markets.

Stage 2 introduced separate LSTM models for each city, each with its **own tokenizer and categorical encodings**. Increasing sequence length and embedding dimensionality improved representation power but also led to persistent overfitting, particularly in larger and more diverse markets such as Los Angeles and Chicago. Predictions were occasionally extreme, and model capacity proved too large for the available samples, motivating a shift toward stronger regularization and dimensionality reduction.

Stage 3, which forms the final version of our LSRT model, adopted a **compact architecture with short text sequences, a small vocabulary, and an 8-unit LSTM**. Regularization was substantially strengthened through dropout, L2 penalties, and Gaussian noise, and price targets were clipped between \$20 and \$600 to ensure numerical stability. City-specific tokenizers, scalers, and one-hot encoders were preserved, and each model was trained with an 80/20 validation split and a stringent early-stopping criterion. This configuration produced the most reliable and well-behaved results, with stable MAE and RMSE across cities (see Table 6), making it suitable for deployment and consistent inference across listings.

Several alternative approaches were explored but ultimately abandoned. **K-fold cross-validation** was attempted but proved unstable due to small per-city sample sizes and the high computational cost of retraining sequence models across folds. **Shared tokenizers** and encoders across cities led to mismatched input structures and inconsistent embeddings. Larger LSTM architectures consistently overfit, producing inflated predictions that required excessive regularization to control. Finally, training without clipped price targets resulted in gradient instability and large validation errors. The final Stage 3 approach strikes the most effective balance between predictive accuracy, robustness, and generalization across cities.