Airbnb Revenue Prediction: A Machine Learning Model for Northeast U.S. Listings

Business Problem

Airbnb hosts face substantial uncertainty in forecasting potential revenue due to variability in property features, host behavior, guest reviews, and fluctuating demand. The objective of this project is to develop a predictive model capable of estimating annual revenue (represented by estimated_revenue_l365d) for Airbnb listings. By generating accurate revenue predictions, this model enables hosts to make informed decisions concerning pricing strategies, amenity offerings, and operational practices. Furthermore, the model provides valuable insights for Airbnb as a platform via improved host onboarding, targeted market strategies, and optimization.

Background / History

The explosion of short-term vacation rentals through platforms such as Airbnb has introduced a highly dynamic rental market. Revenue generation for individual listings varies significantly based on numerous factors, including geographic location, property type, host responsiveness, and guest reviews. Leveraging detailed datasets from Inside Airbnb (Inside Airbnb, n.d.), this project builds machine learning models to forecast revenue.

Data Explanation

The data used in this analysis were obtained from Inside Airbnb, specifically for the Rhode Island, Boston, Cambridge, and New York City markets. The dataset consists of multiple sources. The listings.csv file has comprehensive property and host information across 48,433 records and 80 attributes, including price, property characteristics, amenities, and host metrics. The reviews.csv file has over 1.5 million reviews, providing text data for sentiment analysis.

Data preparation was performed prior to modeling. Several columns with excessive missing values, such as calendar_updated and license, were removed. Monetary fields, such as price, were cleaned to ensure numeric consistency, and rows with missing target values were excluded. Missing review scores were imputed using median values.

Sentiment analysis was performed on the review text using the VADER algorithm (Hutto & Gilbert, 2014) to extract an aggregate sentiment score for each listing. This additional feature captured guest satisfaction patterns that are not reflected in numeric review ratings alone. Further feature engineering included conversion of percentage fields (such as host_response_rate) to numeric format, mapping of boolean fields (e.g., host_is_superhost) to binary values, parsing of the bathrooms_text field into discrete counts for private/shared full and half baths, and expansion of the amenities field into multiple binary indicator columns. Host tenure was calculated as

the number of days since the host account was created (host_since_days). Finally, categorical variables, such as region and property type, were label-encoded to enable modeling. Over 200 engineered features were incorporated into the final dataset for model training.

Figure 1 displays the distribution of listing prices, revealing a right-skewed pattern with a long tail toward higher-priced properties.

Figure 1

Methods

Following exploratory data analysis (EDA), multiple modeling approaches were evaluated to determine the most effective revenue prediction method. The dataset was divided into training and testing subsets using an 80/20 split to assess model performance on unseen data.

Figure 2 presents a correlation heatmap illustrating relationships among key numeric features.

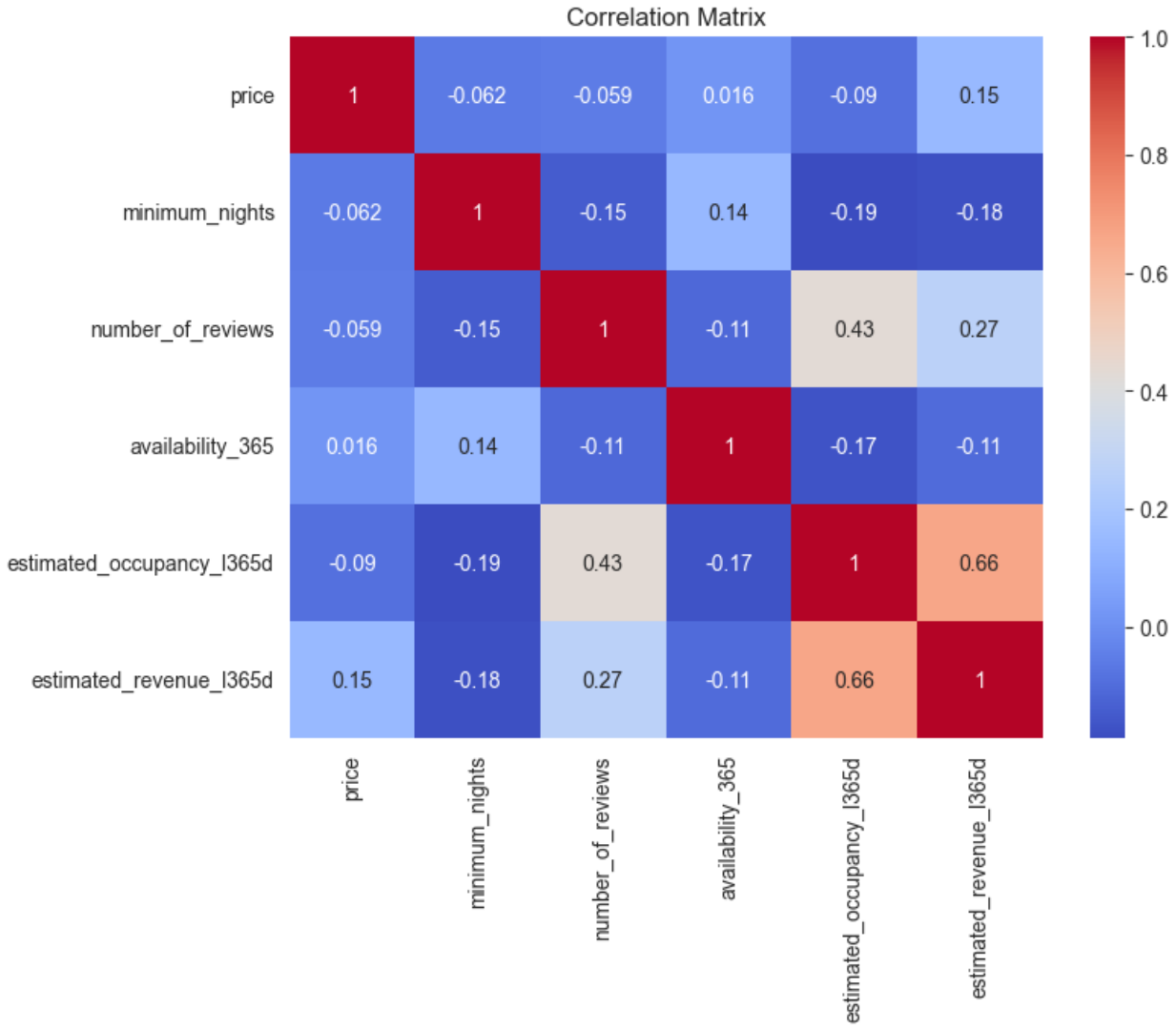

Figure 2

The primary model evaluated was a Random Forest Regressor. Additionally, a HistGradientBoosting Regressor was tested with feature scaling applied to help model stability. Lastly, ElasticNet regression was tested using scaled features to serve as a linear performance benchmark.

Model performance was evaluated using Root Mean Squared Error (RMSE) and R^2, implemented using scikit-learn (Pedregosa et al., 2011), to assess model accuracy and variance explained. A log transformation of the revenue target was tested but the final results were reported using the original dollar scale for interpretability.

Analysis

The Random Forest Regressor has the best performance out of the models tested. The Random Forest model achieved an RMSE of $12,978.97 and an R^2 of 0.778, successfully explaining approximately 78% of the variance in annual revenue. The HistGradientBoosting Regressor performed similarly, with an RMSE of $13,753.88 and an R^2 of 0.751. The ElasticNet model underperformed by comparison, getting an RMSE of $23,324.33 and an R^2 of 0.283, confirming that the problem requires non-linear modeling techniques.

The Random Forest feature importance analysis indicated that daily listing price was the most dominant predictor, followed by review count and minimum nights required for booking. Host responsiveness (acceptance rate), geographic region, and host tenure were also significant.

Figure 3 displays revenue distributions across regions, demonstrating differences in median and variance.

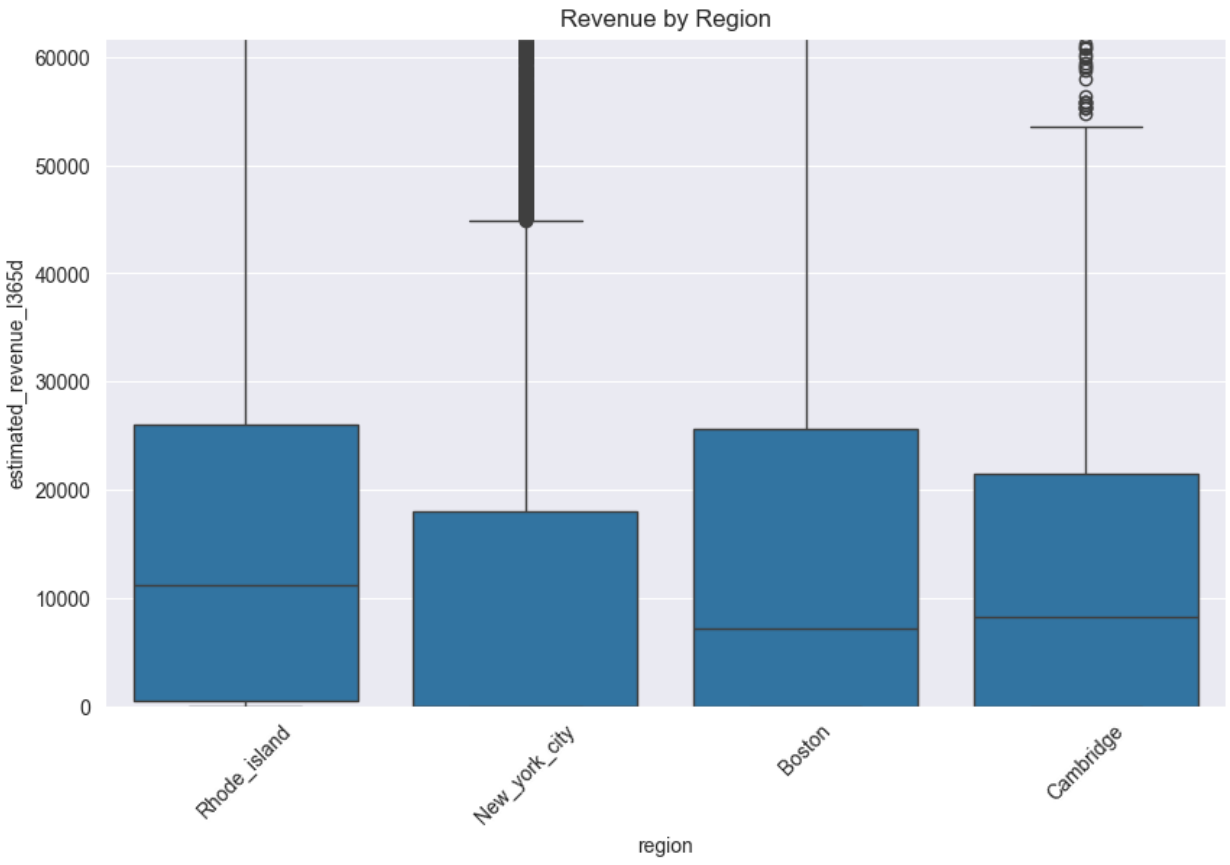

Figure 3

Conclusion

The predictive model demonstrates that Airbnb revenue can be estimated with high accuracy using a machine learning approach. The Random Forest model uses complex interactions across a wide range of features, outperforming linear methods and demonstrating generalization to unseen data. These insights can provide actionable recommendations for both hosts and the platform itself. Hosts can optimize pricing, enhance guest experiences, and prioritize key amenities known to drive revenue, while Airbnb could use these findings to support host onboarding, refine market strategies, and improve platform offerings.

Assumptions

This analysis is based on several important assumptions. It assumes that the estimated_revenue_l365d variable accurately reflects true annual revenue for each listing. The VADER sentiment analysis model is assumed to provide a meaningful representation of guest satisfaction based on review text. Additionally, it is assumed that hosts accurately report amenities and property details.

Limitations

Although data cleaning was performed, some listings contained missing or inconsistent fields that could not be used. The model does not account for seasonality since booking patterns vary throughout the year. The amenity data could have introduced noise, since uncommon amenities were represented across very few listings.

Challenges

The analysis had multiple technical challenges. Parsing the unstructured amenities field and the varied formatting of the bathrooms_text field required text processing and error handling for consistent feature extraction. Managing the high dimensionality of the resulting feature set while minimizing overfitting required reworking feature engineering and validation. Additionally, skewed distributions in important variables, such as price and revenue, required some consideration while modeling.

Future Uses / Additional Applications

Future enhancements could include the incorporation of calendar booking data to directly model seasonality and dynamic pricing fluctuations. However, when trying to get this data the files were corrupted. Hyperparameter tuning for advanced gradient boosting models could get some performance improvements. Interactive dashboards could allow hosts to simulate revenue scenarios under various configurations. The model could also be extended to support pricing optimization algorithms to maximize potential revenue.

Recommendations

Hosts should focus on setting competitive prices, getting positive guest reviews, maintaining high host responsiveness, and offering high-demand amenities such as Wi-Fi and air conditioning. For Airbnb as a platform, integrating revenue prediction tools into host dashboards and providing real-time revenue simulation could improve host engagement.

Implementation Plan

The project is fully reproducible via a Jupyter notebook environment, with all data cleaning, feature engineering, and model training steps documented. The cleaned datasets were saved as snapshots, allowing for quick retraining or exploring additional models.

Ethical Assessment

The model avoids direct bias against new or small-scale hosts by relying on listing features rather than historical revenue alone. Review sentiment analysis was conducted solely on publicly available guest feedback.

References

Hutto, C. J., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media, 8*(1), 216–225.

Inside Airbnb. (n.d.). Inside Airbnb: Adding data to the debate. Retrieved from http://insideairbnb.com/

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Appendix A: Feature Engineering Details

Boolean and percentage fields such as host_is_superhost, host_identity_verified, host_response_rate, and host_acceptance_rate were converted to numeric format for modeling. The host_response_time field was mapped to an ordinal scale to reflect responsiveness levels. Host verification methods were parsed into separate binary columns. Rare property types were grouped and label-encoded, while the complex bathrooms_text field was decomposed into counts of private/shared full and half baths. Amenities were parsed into over 700 binary indicator features to capture available property offerings. Host tenure was calculated as the number of days since account creation. Categorical fields such as region and room_type were label-encoded. Finally, review features including total review count, average sentiment score, and

average review length were aggregated for each listing to enrich the model's understanding of guest perceptions.

Audience Questions

Stakeholders could ask several questions, including the model's accuracy for listings with few reviews, its ability to account for seasonal revenue variation, and the specific amenities that most strongly influence revenue outcomes. Additional question could focus on the handling of pricing outliers, applicability to other regions, computational resource requirements for a production deploy, recommended retraining frequency, performance across different property types, and the prospect of generating optimal pricing recommendations.