Airbnb Pricing Optimization: A Data-Driven Approach

15.C571 Final Project Technical Report

Authors:

Aleksandar Jovanovic-Hacon
Camila Moran-Hidalgo
Sky Pulling

December 2024

Problem Background and Motivation

Hosts on Airbnb face the dual challenge of setting prices that are low enough to attract bookings while ensuring sufficient profit to cover the operational costs associated with owning a rental property - tax rate and cleaning fees. This project tackles the problem of optimizing Airbnb nightly pricing to maximize revenue while remaining competitive enough to avoid overbooking.

Airbnb hosts, especially part-time hosts, struggle to take factors such as seasonality, location, guest reviews, and amenities into account when they lack sufficient data or experience. Unlike professional hosts who have a wealth of booking data, guest feedback, and historical trends to inform their pricing decisions, part-time hosts are particularly prone to suboptimal pricing.

Pricing optimization is a critical issue for hosts and the broader Airbnb platform. Overpricing may deter potential guests, driving them to book with different hosts or consider hotels, leading to vacancies and an unsustainable loss in revenue. Inconsistent pricing can also have adverse effects on marketing and customer relations, potentially undermining trust in the listing. Conversely, when daily rates fail to reflect market demand (determined by seasonality or local events), hosts miss valuable opportunities to maximize their earnings during peak periods. Supporting part-time hosts in overcoming these disadvantages is essential to maintaining a diverse and competitive Airbnb marketplace, where smaller operators can thrive alongside professional hosts.

Pricing optimization is a critical issue for hosts and the broader Airbnb platform. For hosts, effective pricing directly impacts revenue, as prices that are too high result in unbooked properties, while prices that are too low lead to missed earning opportunities. Beyond individual hosts, pricing consistency and fairness are essential to maintain guest trust and satisfaction on the platform. Guests are more likely to return to Airbnb when they perceive pricing as fair, and this loyalty directly affects Airbnb's competitiveness in the short-term rental market.

Moreover, price optimization has broader societal and economic implications. Our model incorporates property tax and home value data from New York, ensuring fair pricing in line with local data, which can also contribute to long term sustainable tourism. Ultimately, optimizing pricing fosters a more equitable, transparent, and competitive short-term rental platform.

Data and Preprocessing

We incorporated data from the Inside Airbnb Open Data dataset[1], enriching it with additional features like NYC property tax calculations based on property classes, market values, and assessed values. This allows the model to reflect real-world costs that hosts must consider in their pricing. The data used in this analysis was sourced from Inside Airbnb and

included listings, calendar, and review data. These datasets provide a comprehensive view of Airbnb properties, their availability, pricing, and associated reviews, making them ideal for modeling demand. The listings dataset contains detailed information on properties, including price, number of reviews, and various features such as location, amenities, and room types. The calendar dataset captures the availability of listings over time, along with their daily prices. The reviews dataset includes guest reviews, providing a qualitative assessment of the properties.

After importing the datasets, we performed several preprocessing steps to ensure data consistency and readiness for modeling. Missing values were carefully handled: for example, any `NaN` values in critical columns like `price` or `review_scores_rating` were replaced with their respective column means to avoid bias from missing data. Extreme values (e.g., infinities) were replaced with the mean values of their respective columns. The goal was to preserve the structure of the data while ensuring it remained clean and suitable for downstream analysis.

To make the data compatible with the logistic regression model and ensure interpretability, we scaled key variables of interest. Instead of min-max normalization, we standardized these variables to have a mean of 0 and a standard deviation of 1 using z-score normalization. This transformation avoids issues where extreme values dominate the scaling process and ensures that variables with different ranges contribute equally to the model. The transformations applied were as follows:

$$\text{Price (scaled)} = \frac{\text{Price} - \mu_{\text{Price}}}{\sigma_{\text{Price}}}$$

$$\text{Review Scores Rating (scaled)} = \frac{\text{Review Scores Rating} - \mu_{\text{Review Scores Rating}}}{\sigma_{\text{Review Scores Rating}}}$$

The scaled features had the following summary statistics:

- **Price (scaled):** Mean = 0.0000, Standard Deviation = 1.0000
- **Review Scores Rating (scaled):** Mean = 0.0000, Standard Deviation = 1.0000

Additionally, columns not directly relevant to the analysis were filtered out to focus on demand proxies such as `availability_30` (the number of days a listing is available in the next 30 days) and `demand_proxy_1`, which captures normalized demand using the formula:

$$\text{Demand Proxy (1)} = 1 - \frac{\text{availability\_30}}{30}$$

This ensures that demand is inversely related to the number of days a listing is available.
To address the challenges faced by Airbnb hosts, particularly part-time operators, we developed a data-driven optimization model designed to dynamically adjust nightly pricing. Our approach integrates key factors that influence pricing, including operational costs (cleaning fees and daily property taxes), demand sensitivity, guest reviews, and local market conditions. By employing a non-linear optimization framework using Gurobi, we formulated the problem with an objective function that maximizes expected revenue, constrained by the

need to meet operational costs and achieve target occupancy rates. Central to our method is a logistic demand model that predicts the likelihood of bookings as a function of price, adjusted for factors such as review scores and market trends.

Logistic Regression Model

The logistic regression model was designed to predict demand as a function of `price` and scaled `review_scores_rating`. The model is defined by the following equation:

$$f(p, r; \alpha, \gamma) = \frac{1}{1 + \exp(\alpha \cdot p + \gamma \cdot (1 - r))}$$

where p represents the price, r represents the scaled review scores, and α and γ are the parameters to be estimated [2]. Each parameter has a specific interpretation:

- α: measures the sensitivity of demand to changes in price. A higher absolute value of α indicates greater sensitivity to price.
- γ: measures the sensitivity of demand to changes in review scores. A higher absolute value of γ indicates greater sensitivity to ratings.

The function incorporates the term $1-r$, emphasizing that higher ratings should correlate with higher demand.

Parameter Estimation

The parameters were estimated using the `scipy.optimize.curve_fit` function. Initial guesses for the parameters were set to $[0.1, 0.1]$ to guide the optimization algorithm. The optimization was constrained to enforce $\alpha > 0$, ensuring that demand behaves as expected with respect to price, while γ was unconstrained. The covariance matrix was used to calculate the standard deviations of the parameter estimates.

After fitting the model, the estimated parameters were $\alpha = 0.102, \gamma = 0.255$. These values indicate the direction and magnitude of the relationships between the independent variables (`price` and `review_scores_rating`) and the dependent variable (`demand_proxy_1`).

The results of the logistic regression model suggest the following:

1. **Price Sensitivity (α)**: Demand decreases significantly as price increases, as evidenced by the positive value of α. This aligns with economic expectations that higher prices deter bookings.
2. **Review Scores Sensitivity (γ)**: Demand modestly increases as review scores improve, reflected by the negative value of γ. Since the function uses $1 - r$, a lower r (indicating worse ratings) corresponds to lower demand, confirming the positive impact of higher ratings.

3. **Baseline Demand**: The overall function provides a good estimate of how demand responds to these variables, with a consistent relationship between price, ratings, and demand.

These insights suggest that pricing strategies and maintaining high ratings are critical for optimizing demand. Price reductions can significantly boost demand, while maintaining high review scores ensures consistent interest from potential customers.

Constraints

Property Taxes

Why Property Taxes Matter Estimating property taxes is essential for our pricing model, as we aim to set a minimum nightly rate for Airbnbs that exceeds their daily property tax costs. This ensures that hosts can cover property tax expenses while remaining competitive in the market.

Steps to Calculate Property Taxes

1. Determining Unit Value: We used Zillow's estimated home values [5] (as of September 2024) categorized by neighborhood and number of bedrooms. The following steps outline the process:

- **Bedroom Categories:** Properties were grouped into 1, 2, 3, 4, or 5+ bedrooms. These categories were integrated into a dataset (`home_value_data`), and listings were adjusted to match this format.
- **Adjustments for Missing/Invalid Data:**
  - Listings with 0 bedrooms or missing bedroom data were assigned 1 bedroom.
  - Private rooms without specified bedrooms were assumed to have 1 bedroom.
  - For other empty entries, the number of bedrooms was estimated using the ceiling of half the accommodation capacity.
- **Neighborhood Matching:**
  - The listing dataset contained 225 neighborhoods, while the Zillow dataset had 162 regions.
  - 145 neighborhoods were matched by name (accounting for case and spelling differences), while the remaining matches were determined using Google Maps proximity.
- **Merging Datasets:** Using neighborhood (`RegionName`) and bedroom count, the datasets were joined to assign property values to each listing. Missing property values were filled using historical Zillow data or averaged values for similar NYC units.

2. Calculating Property Tax: Property taxes were calculated using the formula:

$$\text{Daily Property Tax} = \frac{\text{Estimated Market Value} \times \text{Assessment Ratio} \times \text{Effective Tax Rate}}{365}$$

The process accounts for NYC's tax structure:

- **Tax Classifications:** NYC properties fall into four classes [3], but Airbnb properties were classified as either:
  - **Class 1:** Residential properties with 1–3 units (applied to non-Manhattan units with 3+ bedrooms).
  - **Class 2:** Residential properties with more than 3 units (applied to Manhattan properties and non-Manhattan units with 2 or fewer bedrooms).
- **Assessment Ratios and Effective Tax Rates:**[4]
  - Class 1: Assessment ratio = 6%, effective tax rate = 0.753%.
  - Class 2: Assessment ratio = 45%, effective tax rate = 4.212%.

This calculation ensures that our pricing model incorporates realistic daily property tax estimates, providing hosts with a minimum pricing floor that reflects real-world costs.

Modeling and Methods

Canonical Optimization Problem

The non-robust optimization problem aims to determine the optimal price p that maximizes revenue for a given listing. The revenue is modeled as the product of price and demand:

$$\max_{p} \; R(p) = p \cdot d(p),$$

where the demand function $d(p)$ is given by the logistic function for which α and γ were fit to the data:

$$d(p; \alpha, \gamma, r_{\text{scaled}}) = \frac{1}{1 + \exp\left(\alpha(p - p_{\text{current}}) + \gamma(1 - r_{\text{scaled}})\right)}.$$

The optimization is subject to the following constraints:

- The price must cover the `daily_tax` and `daily_upkeep`:

$$p \geq \texttt{daily\_tax} + \texttt{daily\_upkeep}.$$

- The demand must exceed a minimum threshold of one booking per month to ensure viability:

$$d(p; \alpha, \gamma, r_{\text{scaled}}) \geq \frac{12}{365}.$$

Robust Reformulation

To account for parameter uncertainty, the robust optimization problem considers multiple scenarios corresponding to the extremes of confidence intervals for α and γ. The parameters are assumed to lie within the following bounds:

$$\alpha \in [\alpha_{\text{lb}}, \alpha_{\text{ub}}], \quad \gamma \in [\gamma_{\text{lb}}, \gamma_{\text{ub}}],$$

where:

$$\alpha_{\text{lb}} = \alpha - z \cdot SE_\alpha, \quad \alpha_{\text{ub}} = \alpha + z \cdot SE_\alpha,$$
$$\gamma_{\text{lb}} = \gamma - z \cdot SE_\gamma, \quad \gamma_{\text{ub}} = \gamma + z \cdot SE_\gamma,$$

and $z = 1.96$ for a 95% confidence interval.

The robust reformulation maximizes the worst-case revenue z, defined as the minimum revenue across all scenarios:

$$\max_p \ z \quad \text{subject to:}$$

$$z \leq p \cdot \frac{1}{1 + \exp\left(\alpha(p - p_{\text{current}}) + \gamma(1 - r_{\text{scaled}})\right)}, \quad \forall \alpha \in [\alpha_{\text{lb}}, \alpha_{\text{ub}}], \quad \gamma \in [\gamma_{\text{lb}}, \gamma_{\text{ub}}],$$

, and:

$$\frac{1}{1 + \exp\left(\alpha(p - p_{\text{current}}) + \gamma(1 - r_{\text{scaled}})\right)} \geq \frac{12}{365}, \quad \forall \alpha \in [\alpha_{\text{lb}}, \alpha_{\text{ub}}], \quad \gamma \in [\gamma_{\text{lb}}, \gamma_{\text{ub}}].$$

Objective Function with Regularization

To improve numerical stability and prevent excessively large values for p, a regularization term was added to the objective function:

$$\text{Objective: } \max\left(z - \epsilon p^2\right), \quad \epsilon = 10^{-6}.$$

Solution Methodology

The optimization was implemented in Julia using the `JuMP` package with the `Ipopt` solver. For each listing, the optimization was performed across all scenarios, and the robust optimal price was stored for further analysis. Listings for which no feasible solution could be found were flagged and excluded from the results.

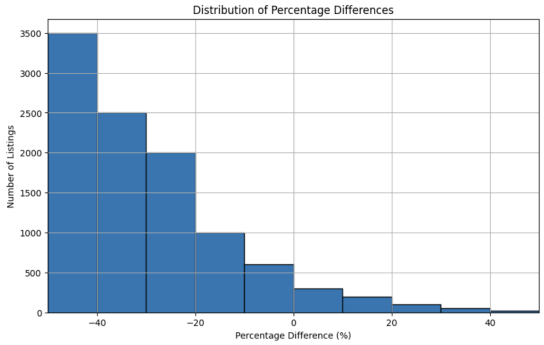
Results and Discussion

Baseline Revenue Calculation

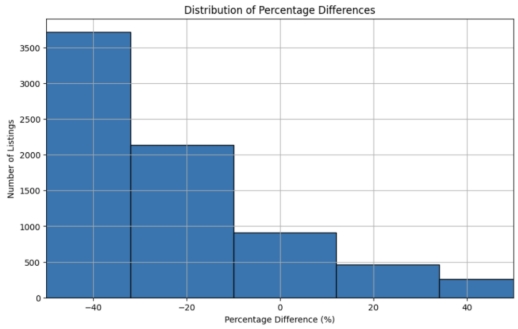
To establish a baseline for evaluating the performance of our pricing optimization model, we calculated the total revenue generated by each Airbnb listing over a one-month period, from September 5th to October 5th. The purpose of this calculation was to measure how much money was actually made under the current pricing strategies employed by Airbnb hosts.

The dataset used for this analysis was filtered to include only listings with valid pricing and availability information. The revenue for each listing was calculated as the sum of nightly prices for all booked days within the period. This baseline serves as a reference point to evaluate whether our optimized pricing model can improve revenue outcomes for Airbnb hosts.

Canonical Optimization Results



(a) Histogram for non-robust optimization results.



(b) Histogram for robust optimization results.

Figure 1: Comparison of percentage difference histograms for non-robust and robust optimization approaches.

The general optimization model was aimed at maximizing revenue by adjusting prices according to a logistic demand function using the sensitivity to price change and the sensitivity of rating (α and γ) calculated above. The results highlight significant revenue potential:

- **58% of listings** experienced an increase in revenue.
- The **median revenue gain of 32%** shows substantial growth for many listings.

However, 41.65% of the listings saw decreases in revenue, underscoring the model's dependence on accurate demand predictions. Thus, since we were unable to perform the AB testing and measure the change in demand directly given a change in price, it is likely that our α and γ parameters may not be exactly tuned to the true sensitivity of demand.

Robust Optimization Results

The robust optimization model prioritized a risk-averse strategy, ensuring stability in demand while minimizing the risk of significant revenue loss.

- Only **29.41% of the listings** experienced revenue gains.
- The majority, **70.59% of the listings**, saw decreases in revenue, with the median change being a **decrease of 28.27%**.

- However, the **mean revenue change was a 129.74% increase**, indicating variability in the results across the listings.

This conservative approach is ideal for listings where maintaining steady demand is critical. By accounting for uncertainty in demand parameters, robust optimization minimizes worst-case revenue losses, providing a safety-first pricing strategy.

In general,the results demonstrate a trade-off between risk and reward in pricing strategies. General optimization is best suited for listings with reliable demand predictions, as it maximizes revenue potential while accepting the risk of overpricing. In contrast, robust optimization is designed for risk-averse hosts, prioritizing consistent bookings and minimizing the impact of demand variability, although this comes at the expense of potential revenue growth.

Future Work

Enhancing the Robust Optimization Model

Given additional time, we would enhance our robust optimization model by incorporating parameters that capture the unique attributes of each listing. This enhancement involves ranking neighborhoods, amenities, and property characteristics numerically and integrating these scores into the existing logistic function. For neighborhoods, we propose assigning a numeric score $S_{\text{neighborhood}}$ based on factors such as proximity to attractions, walkability, safety, and historical demand patterns, representing the relative demand potential of each area. Property amenities, such as Wi-Fi, parking, and pool availability, would be assigned numeric scores $S_{\text{amenities}}$, weighted by their historical impact on guest preferences to streamline the inclusion of amenity-related factors. Similarly, key property characteristics, including the number of bedrooms (S_{bedrooms}), bathrooms ($S_{\text{bathrooms}}$), and overall size, would be standardized into numeric scores to ensure consistency in model inputs. These enhancements would allow the model to better account for listing-specific attributes, improving its accuracy and applicability.

Enhanced Logistic Function

To incorporate these new features, we propose extending our logistic demand function as follows:

$$D(p) = \frac{1}{1 + \exp\left(-\left(\alpha + \gamma p + \beta_1 S_{\text{neighborhood}} + \beta_2 S_{\text{amenities}} + \beta_3 S_{\text{characteristics}}\right)\right)}$$

where $D(p)$ represents demand as a function of price p, α as the baseline demand intercept and γ as the price sensitivity coefficient. β_1, β_2, and β_3 are coefficients that represent the weights assigned to the neighborhood score ($S_{\text{neighborhood}}$), amenities score ($S_{\text{amenities}}$), and property characteristics score ($S_{\text{characteristics}}$), respectively. These numeric scores capture specific aspects of the listing's location, features, and overall appeal.

Integrating these parameters into the model offers several benefits. It enhances demand predictions by incorporating listing-specific and location-specific factors, allowing for a more tailored pricing strategy that accounts for each listing's unique attributes. This approach not only improves the accuracy of the model but also optimizes revenue potential while maintaining robustness under demand uncertainty. Together, these enhancements make the model more applicable and effective in real-world pricing scenarios.

Bibliography

References

[1] Inside Airbnb. Get the data @ONLINE, 2024.

[2] Steven Berry and Ariel Pakes. The pure characteristics demand model. *International Economic Review*, 48(4):1193–1225, 2007.

[3] NYC Department of Finance. Definitions of property assessment terms @ONLINE, 2024.

[4] New York State Department of Property and Taxatino. Overall full-value tax rates by county (all taxing purposes): 2012-2021 @ONLINE, 2021.

[5] Zillow. Zillow home value index data @ONLINE, 2024.