FINAL PROJECT REPORT: AIRBNB ANALYSIS IN BROWARD COUNTY, FL.

Team 10

Aishwarya Anandan

Puneeth Rao Gunuganti

Varshini Kuppusami

Vipin Kumar Karthikeyan

Date: December 8, 2025

Table of Contents

1. Executive Summary

The proliferation of short-term rental platforms has transformed the hospitality landscape, with Airbnb emerging as a dominant force. In tourist-heavy regions like Broward County, Florida, the market is highly competitive. For hosts, the challenge lies in determining the optimal nightly price—a figure that must be competitive enough to attract guests while high enough to cover costs and generate profit. This project aims to address this pricing dilemma by developing a machine learning model capable of predicting Airbnb listing prices based on historical data.

Our team utilized a comprehensive dataset sourced from *InsideAirbnb*, comprising detailed information on listings, host characteristics, and customer reviews. The project followed a rigorous data science lifecycle: data acquisition, cleaning, exploratory data analysis (EDA), feature engineering, and predictive modeling.

Key findings from our analysis reveal that the rental market in Broward County is structurally driven. The number of accommodates (capacity) is the single most significant predictor of price. Furthermore, the market exhibits a log-normal price distribution, necessitating advanced statistical transformations for accurate modeling.

We established a baseline predictive model using **Linear Regression**, which achieved an **R^2 score of approximately 0.60** on unseen test data. This indicates that our model can explain roughly 60% of the variance in listing prices, a robust starting point for behavioral data. This report details our methodology, technical implementation, and the strategic implications of our findings for stakeholders in the Broward County tourism sector.

---

2. Introduction

2.1 Background

Broward County, home to Fort Lauderdale, Hollywood, and Pompano Beach, is a major hub for tourism in South Florida. The demand for short-term rentals fluctuates based on seasonality, proximity to beaches, and local events. Unlike hotels, which utilize sophisticated revenue management systems, individual Airbnb hosts often rely on intuition or static pricing, leading to market inefficiencies.

2.2 Problem Statement

The core problem addressed in this study is the **inefficiency in pricing strategies** among Airbnb hosts.

- **Overpricing** leads to high vacancy rates and lost revenue.
- **Underpricing** results in "money left on the table" and can attract lower-quality guests who may damage property.
- **Market Complexity**: With thousands of unique listings ranging from shared rooms to waterfront mansions, identifying the "fair market value" of a specific property is computationally complex for a human.

2.3 Project Objectives

This project seeks to build a scalable, data-driven solution to this problem. Our specific objectives are:

1. **Data Engineering**: To ingest and clean messy web-scraped data into a structured format suitable for machine learning.
2. **Market Analysis**: To perform Exploratory Data Analysis (EDA) to uncover trends, outliers, and correlations in the Broward County market.
3. **Predictive Modeling**: To develop and evaluate a regression model that predicts the nightly price (price) of a listing based on its features.

---

3. Data Description and Acquisition

3.1 Data Source

The data for this project was obtained from **InsideAirbnb**, an independent watchdog that scrapes and aggregates Airbnb data for public use. We utilized the dataset snapshot dated **June 24, 2025**.

The dataset consists of three primary files:

1. **listings.csv.gz**: The core dataset containing detailed attributes for each listing, including location, room type, amenities, and price.
2. **reviews.csv.gz**: A text-heavy dataset containing guest reviews and metadata.

3.2 Key Variables

From the raw data, we identified a subset of variables with high potential for predictive power:

- **Target Variable**: price (The nightly rental cost).

- **Host Features**: host_since, host_response_rate, host_is_superhost.
- **Property Features**: room_type, property_type, accommodates, bedrooms, beds, bathrooms.
- **Location Features**: neighbourhood_cleansed, latitude, longitude.

---

4. Data Preparation and Cleaning

Real-world data is rarely ready for immediate modeling. We dedicated a significant portion of our project timeline to data preprocessing to ensure the integrity of our results.

4.1 Data Cleaning & Type Conversion

The raw data contained various formatting artifacts that required correction:

- **Currency Conversion**: The price column was imported as a string containing currency symbols ($) and commas (e.g., "$1,200.00"). We utilized string manipulation functions to strip these characters and cast the column to a floating-point number format.
- **Percentage Parsing**: Columns such as host_response_rate and host_acceptance_rate were stored as strings (e.g., "98%"). These were converted to decimal format (0.98) to allow for numerical operations.
- **Boolean Mapping**: Several categorical columns utilized "t" and "f" to represent True and False. We mapped these to standard Python Boolean values or binary integers (1/0) for compatibility with scikit-learn algorithms. Examples include host_is_superhost and instant_bookable.
- **Date Parsing**: Temporal features (last_scraped, host_since, first_review) were converted to Datetime objects. This step was crucial for calculating derived features like host tenure.

4.2 Handling Missing Values

Missing data is a common challenge in scraped datasets. We adopted a **stratified imputation strategy** rather than a global mean filling, which could introduce bias.

- **Numerical Imputation**: For structural features like bedrooms, beds, and bathrooms, we observed that missing values often occurred in specific property types (e.g., private rooms often missing a bathroom count). We grouped the data by room_type

and neighbourhood_cleansed and filled missing values with the **median** of that specific group. This ensures that a missing bedroom count for a "Private Room" in "Hollywood" is filled with a value representative of similar listings, rather than a global average that includes mansions.

- **Review Scores**: Listings with no reviews had missing scores. We created a binary flag no_reviews to capture this information and imputed the missing scores with the dataset median to prevent the model from dropping these rows.

4.3 Feature Selection & Dropping

To reduce noise, we removed columns that provided no predictive value or had excessive missing data.

- **Identifiers**: Columns like listing_url, scrape_id, and picture_url were dropped.
- **Redundant Location Data**: neighbourhood_group_cleansed was found to be entirely empty (null) for this dataset and was removed.
- **License Data**: The license column contained over 50% missing values and was deemed unusable for this analysis.
- **Low-signal tests**: bath_is_private, bath_is_shared after correlation check.

4.4 Categorical Encoding

To utilize categorical information in the regression models, we applied One-Hot Encoding to the room_type, property_type, and neighbourhood_cleansed features. This process transformed these string variables into binary dummy variables (0/1), allowing the model to assign specific weights to different neighborhoods (e.g., distinguishing the premium value of "Fort Lauderdale" vs. inland areas) and property styles.

4.5 Text Preprocessing

To prepare the unstructured text data (listings descriptions) for analysis, we performed a series of cleaning steps. This included:

- **HTML Removal:** Stripping HTML tags (e.g.,
) using BeautifulSoup to isolate the human-readable text.
- **Normalization:** Converting all characters to lowercase and removing punctuation to ensure consistency (e.g., treating "Ocean" and "ocean" as the same token).
- **Stopword Removal:** Filtering out common, non-predictive English words (like "the", "is", "and") to focus on high-value content words.

- **Lemmatization:** Reducing words to their root form (e.g., "beaches" to "beach") to reduce the dimensionality of the text data.

4.6 Filter and Preprocess: Reviews

- **Date Conversion:** Converted review dates into a standardized datetime format to enable accurate time-based analyses.
- **Sampling:** Extracted a random sample of ***5,000 reviews*** to ensure efficient processing and reproducibility while maintaining data diversity.
- **Language Detection:** Utilized the *langdetect* library to automatically identify the language of each review.
- **Translation:** Employed the googletrans library (*version 4.0.0-rc1*) to translate all non-English reviews into English, ensuring consistency across the dataset.
- **Standardization:** Stored the translated text in a new column (english_comments) so that all subsequent analyses operated on English text only.
- **Text Cleaning:** Applied a custom preprocessing function to remove noise (URLs, punctuation, and non-alphabetic characters), eliminate stopwords, and lemmatize words, resulting in a standardized and meaningful text corpus (*cleaned_comments*).
- **Final Output:** Generated a preview table displaying the original, translated, and cleaned comments for the first ten entries to verify preprocessing quality.

	comments	english_comments	cleaned_comments
108825	Great place for a night!	Great place for a night!	great place night
453910	This airbnb is perfectly/centrally located, be...	This airbnb is perfectly/centrally located, be...	airbnb perfectlycentrally located beach pier l...
63326	This place was perfect for our family! We wer...	This place was perfect for our family! We wer...	place perfect family able stay near beach quai...
421587	Nice house. Not a bad area. Seems to be lackin...	Nice house. Not a bad area. Seems to be lackin...	nice house bad area seems lacking bit quality ...
490146	Everything was excellent!	Everything was excellent!	everything excellent
298505	Great place to stay if you want to be close to...	Great place to stay if you want to be close to...	great place stay want close wilton manner rest...
170530	Great and privat spot for a few nights. Bed wa...	Great and privat spot for a few nights. Bed wa...	great privat spot night bed really good
587894	Excelente lugar con una gran vista! Los mueble...	Excellent place with a great view!The furnitur...	excellent place great viewthe furniture descri...
209557	Great host, great location, great vacation!	Great host, great location, great vacation!	great host great location great vacation
618667	El apartamento esta muy lindo, en un condomini...	The apartment is very nice, in a safe condomin...	apartment nice safe condominium easy accessthe...

---

5. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) forms the foundation of our predictive modeling approach. Before building models, we systematically examined the distribution of variables, identified relationships between features, and uncovered patterns that would inform our feature engineering and model selection strategies. Our EDA focused on four key areas: target

variable transformation, feature correlation analysis, geographic and categorical patterns, and quality signal evaluation.

5.1 Target Variable Analysis: Price Distribution

The most critical finding from our initial univariate analysis was the distribution of our target variable, *price*. Understanding and properly transforming this variable was essential for meeting the assumptions of linear regression models and achieving optimal predictive performance.

Raw Price Distribution

The raw nightly price data exhibited severe right-skewness, a common characteristic in real estate and hospitality pricing datasets. Key statistics revealed:

- **Mean Price:** $179 per night
- **Median Price:** $130 per night
- **Price Range:** $36 to $1,289 per night

The substantial difference between the mean ($179) and median ($130) quantified the rightward skew, indicating the presence of high-value luxury properties that pulled the mean upward while the typical listing clustered at lower price points. This distribution pattern posed three significant challenges for linear regression: violation of normality assumptions, heteroscedasticity in residuals, and disproportionate influence of outliers on model coefficients.

Log Transformation Strategy

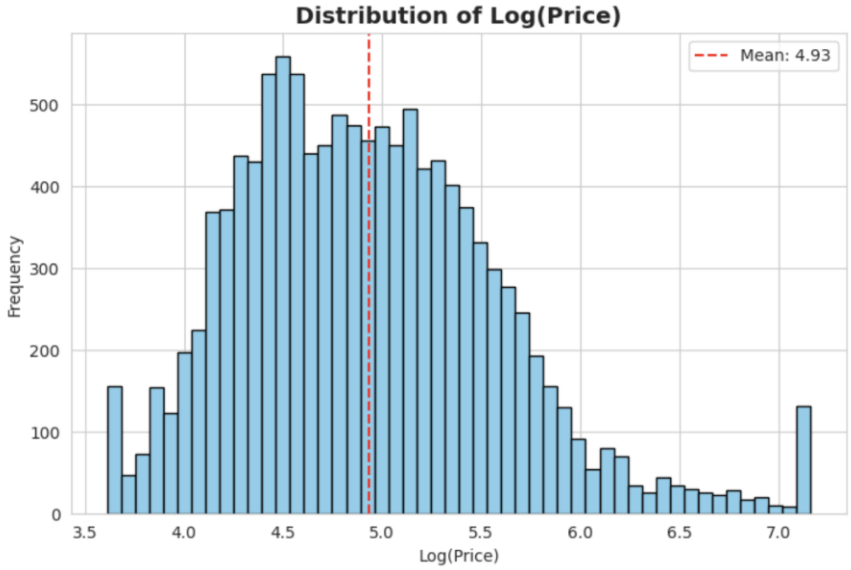
To address these distributional challenges, we applied a logarithmic transformation to the price variable using *np.log1p()*. This transformation choice was motivated by three key considerations:

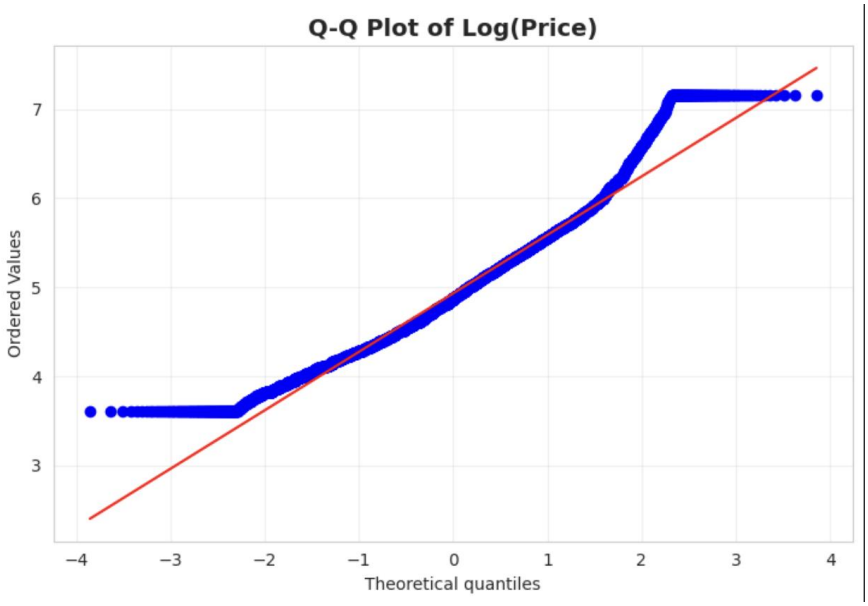
- **Normalization:** Log transformation compresses the right tail of the distribution, pulling extreme values closer to the center and achieving approximate normality.
- **Interpretability:** Coefficients in log-transformed models represent percentage changes rather than absolute dollar amounts, which aligns with how pricing premiums operate in real markets.
- **Stability:** The transformation reduces the impact of outliers, improving model stability and generalization performance.

Transformation Validation

The effectiveness of the log transformation was rigorously validated through both visual and statistical methods:

- **Distribution Shape:** The histogram of log(price) exhibited a near-perfect bell curve with mean = 4.93 (corresponding to ~$138 in original scale), indicating successful normalization.
- **Q-Q Plot Analysis:** The quantile-quantile plot showed strong alignment between our transformed data and theoretical normal quantiles, with only minimal deviation in the extreme tails. This validates that our transformed target variable meets the normality assumption required for optimal linear regression performance.





5.2 Feature Correlation Analysis: Identifying Key Price Drivers

Following the successful transformation of our target variable, we conducted comprehensive correlation analysis to identify which features most strongly influence

pricing. This analysis revealed a clear hierarchical structure of price drivers, with property size metrics forming the foundation and amenity features acting as premium multipliers.

Property Characteristics: Core Price Determinants

Property size metrics demonstrated the strongest correlations with log(price), confirming the hypothesis that physical capacity serves as the primary driver of rental value:

- **Accommodates:** r = +0.70 (strongest predictor)
- **Bedrooms:** r = +0.67
- **Bathrooms:** r = +0.64
- **Beds:** r = +0.62

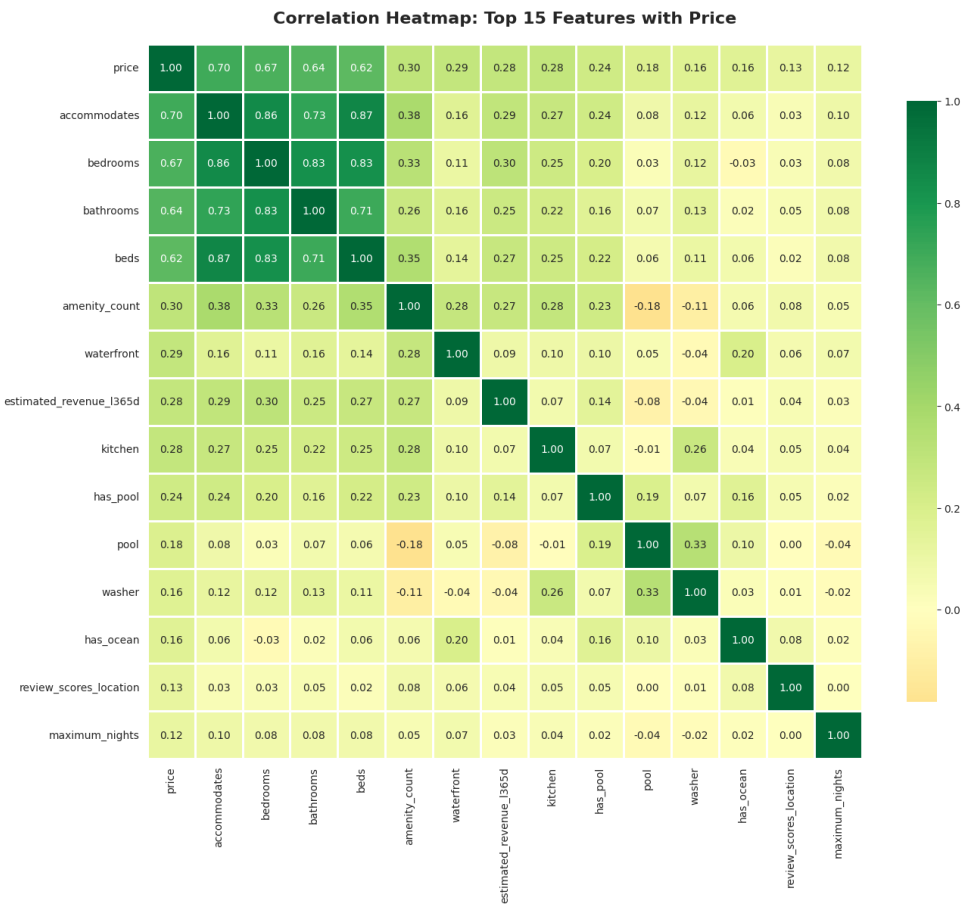
These correlations indicate that capacity-related features explain roughly 40-50% of price variance individually ($r^2 \approx 0.40$-0.49). The high intercorrelation among these size metrics (all correlations > 0.70 with each other) suggests multicollinearity, which we addressed through careful feature engineering and regularization in later modeling stages.

Amenity Features: Premium Multipliers

In contrast to structural features, luxury amenities showed moderate correlations but substantial price premiums:

- **Waterfront:** r = +0.29 (71.8% price premium)
- **Pool:** r = +0.18 (34.5% price premium)

The apparent contradiction between moderate correlation coefficients and substantial price premiums reveals an important pattern: amenities act as multiplicative factors rather than additive contributors. A waterfront property does not simply add $X to the base price; rather, it scales the entire price by ~72%. This finding directly informed our feature engineering strategy, where we created interaction terms between size metrics and amenity flags to capture these multiplicative effects.



5.3 Geographic and Categorical Patterns

Location and property type emerged as critical determinants of pricing, with dramatic variation across neighborhoods and a clear market preference for privacy and exclusive access.

Neighborhood Price Variation

Geographic analysis revealed a 5-fold price variation across Broward County neighborhoods, indicating significant location-based market segmentation:

- **Most Expensive:** Southwest Ranches ($393), Plantation ($303)
- **Most Affordable:** Pembroke Park ($75), Lauderdale Lakes ($76)
- **Highest Volume:** Hollywood (25.8%), Fort Lauderdale (23.1%)

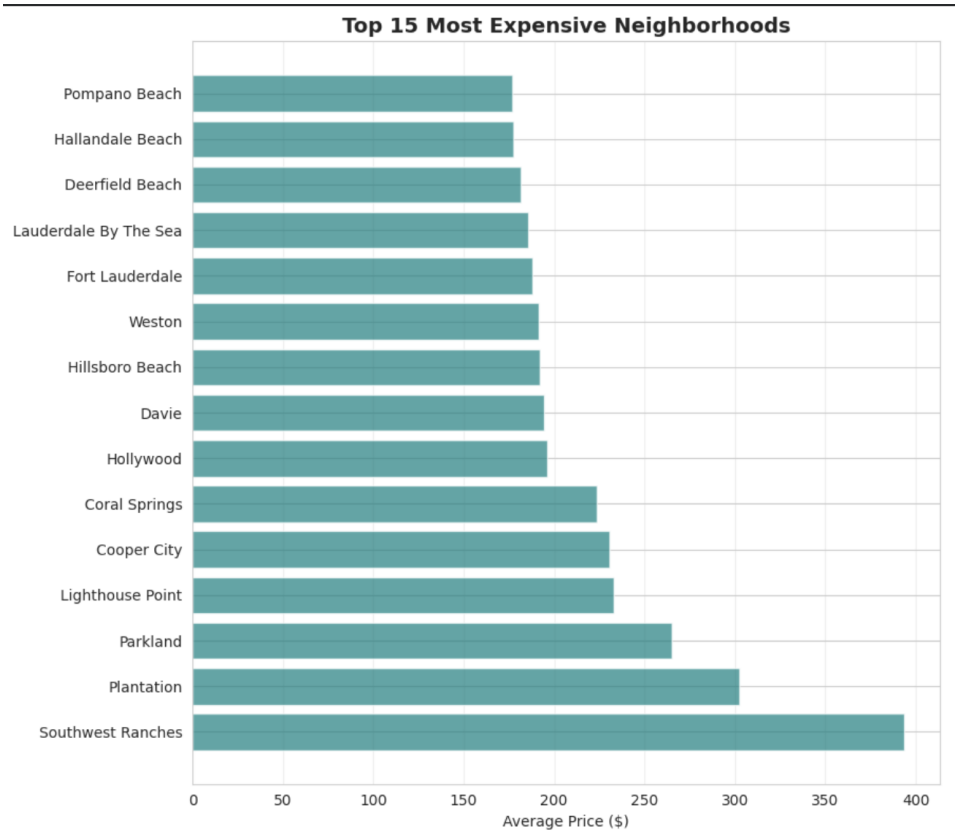
An important finding emerges from comparing price and volume distributions: the highest-volume neighborhoods (Hollywood, Fort Lauderdale) occupy the mid-price range, not the premium tier. This volume-price disconnect has modeling implications—we cannot simply optimize for high-price predictions since the bulk of the market (and revenue opportunity) exists in moderate-price segments.

Room Type Distribution and Pricing

Room type analysis revealed a heavily skewed market structure favoring entire homes:

- **Entire home/apartment:** 83.6% of listings, averaging $194/night
- **Private room:** 15.5% of listings, averaging $100/night
- **Shared room:** <1% of listings (negligible market presence)

The 94% price premium for entire homes ($194 vs. $100) quantifies the "privacy premium" in this market. This strong market preference suggests that the Broward County Airbnb market caters primarily to tourists and families seeking vacation accommodations rather than budget travelers or long-term housing solutions. This insight influenced our feature selection, where we treated room_type as a high-signal categorical variable deserving of prominent inclusion in all models.



5.4 Review Patterns and Host Quality Signals

Our analysis of quality indicators such as reviews, ratings, and Super host status revealed counterintuitive patterns that challenged conventional assumptions about the relationship between quality and pricing power.

Review Score Analysis

Review scores in the Broward County market exhibited surprisingly weak correlation with price:

- **Average Rating:** 4.75 out of 5.00 (highly skewed positive)
- **Median Review Count:** 18 reviews per listing
- **Price Correlation:** r = -0.11 (weak negative correlation)

The near-universal positive ratings (4.75 average) suggest a ceiling effect where most properties cluster at high scores, eliminating discriminatory power. The weak negative correlation indicates that slightly lower-rated properties sometimes command higher prices, possibly due to confounding factors like location or property size. This finding suggests that review scores function more as a booking probability filter (guests avoid poorly rated properties) rather than a premium pricing signal.

Superhost Impact: Volume Over Premium

Superhost status, Airbnb's badge of hosting excellence, showed an unexpected pattern: volume advantage without price premium.

- **Price Impact:** Minimal difference (Superhosts: $176 vs. Non-Superhosts: $175)
- **Rating Advantage:** +18% higher average rating (4.82 vs. 4.67)
- **Review Volume:** +119% more reviews (67 vs. 31 median)

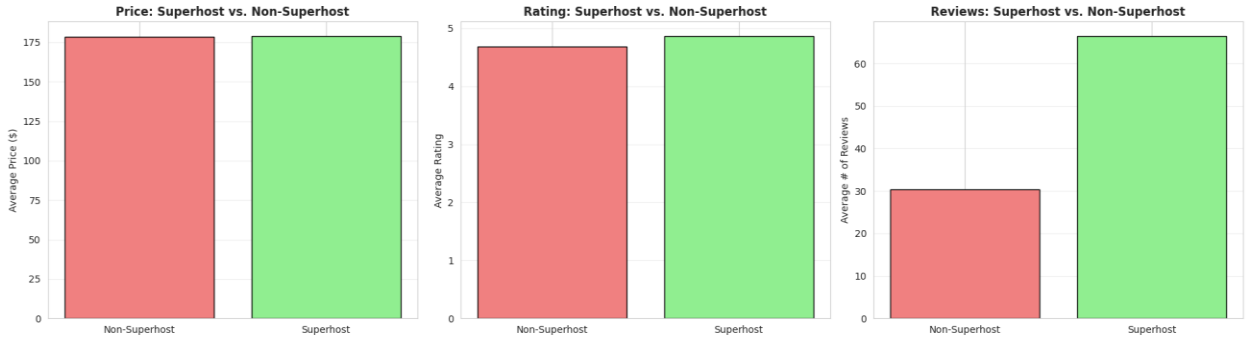
These statistics reveal a strategic trade-off: Superhosts achieve substantially higher booking volumes (evidenced by 119% more reviews) but do not leverage their quality advantage into price premiums. This suggests Superhosts optimize for occupancy rate and total revenue rather than per-night pricing. From a modeling perspective, this indicates that Superhost status should be treated as a booking probability enhancer rather than a price predictor in regression models.

Availability Patterns: Bimodal Distribution

Analysis of availability patterns revealed a bimodal distribution suggesting two distinct host strategies:

- **High Availability Cluster:** Properties available 300+ days per year, potentially struggling to achieve full bookings or deliberately pricing for volume
- **Low Availability Cluster:** Properties available <100 days per year, indicating either high demand allowing selective booking windows or mixed-use properties with significant personal use

This bimodal pattern suggests that availability serves as a potential signal of pricing strategy and demand level, warranting its inclusion as a derived feature in predictive models.



5.5 Key Insights and Modeling Implications

Our comprehensive EDA revealed several critical insights that directly informed our subsequent feature engineering and model development:

- **Target Transformation:** Log transformation successfully normalized price distribution, validated by Q-Q plot alignment, enabling effective use of linear regression models.
- **Hierarchical Price Structure:** Property size metrics (accommodates, bedrooms, bathrooms, beds) form the pricing foundation ($r > 0.60$), while luxury amenities (waterfront, pool) act as multiplicative premium factors. This pattern suggested creating interaction terms between size and amenity features.
- **Geographic Segmentation:** 5-fold price variation across neighborhoods indicates strong location effects. However, volume concentrates in mid-price areas, requiring models to balance accuracy across the full price spectrum rather than over-optimizing for luxury properties.
- **Quality vs. Price Disconnect:** Review scores and Superhost status show weak correlation with price ($r = -0.11$ and minimal difference respectively) but strong correlation with booking volume. This suggests quality features belong in occupancy prediction models rather than price regression models, or should be treated with reduced weight.
- **Market Composition:** The dominance of entire homes (83.6%) and strong privacy premium ($194 vs. $100) indicates a vacation-oriented market. This justified treating room_type as a high-signal feature and focusing model optimization on the entire home segment.

These insights guided our feature engineering strategy, where we created interaction terms for size-amenity combinations, implemented geographic binning for low-volume neighborhoods to reduce sparsity, and carefully weighted quality features based on their demonstrated relationships with price rather than assumed importance.

---

6. Feature Engineering

Our feature engineering strategy was designed to translate raw listing and review information into structured signals that a model can learn from. We combined structural, temporal, textual, and sentiment-based features to capture not only what a property is (size, location, amenities) but also how guests perceive it (sentiment and topics from reviews).

6.1 Numerical Features

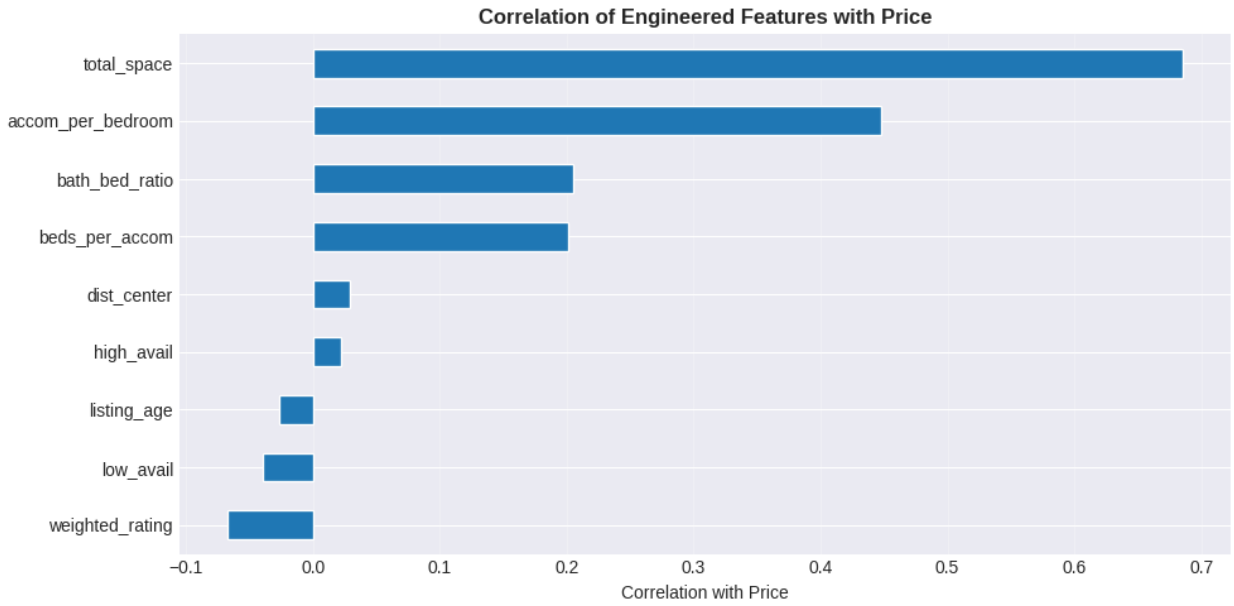
We began with core structural variables such as accommodates, bedrooms, beds, and bathrooms, which EDA identified as the strongest individual predictors of price. To better represent how guests experience space, we transformed these into higher-level numerical features:

- **Space efficiency ratios**
    - accom_per_bedroom = accommodates / (bedrooms + 1)
    - bath_bed_ratio = bathrooms / (bedrooms + 1)
    - beds_per_accom = beds / (accommodates + 1)
      These ratios help distinguish listings that are efficiently laid out from those that are cramped or oversized for their capacity.
    -
- **Aggregate capacity metric**
    - total_space = bedrooms + bathrooms + beds
      This serves as a single proxy for overall size/amenity volume, reducing multicollinearity between raw size features.
- **Engagement and longevity signals**
    - listing_age = number_of_reviews / (reviews_per_month + 0.1), capturing how long a listing has been active and booking.
    - review_count (raw) and its log transform log_review_count to reduce skew from very popular listings.
- **Availability-based strategy indicators**
  From availability_365 we derived:
    - high_avail (1 if > 180 days available)

- low_avail (1 if < 90 days available)
  These capture different hosting strategies—high-availability, volume-oriented vs. scarce, selective inventory.

- **Geospatial proximity feature**
  Using latitude and longitude, we computed dist_center as the Euclidean distance from the median coordinates of Broward County listings. This provides a continuous measure of how centrally located a property is, complementing the neighborhood dummy variables.

All numerical features were inspected for outliers and skew. Where appropriate, we applied log transformations (e.g., review counts) to stabilize variance. Before modeling, numerical columns were standardized using StandardScaler so that features on different scales (e.g., days vs. counts) contribute comparably to distance-based and linear models.



6.2 Date & Time Features

Temporal dynamics are important in a tourism market. We engineered time-based features to capture host experience and listing freshness:

- **Host tenure**

- host_tenure_days = last_scraped − host_since
  Longer tenure reflects more experienced hosts who may have learned to optimize both price and operations over time.

- **New host indicator**
  - is_new_host = 1 if host_tenure_days < 90, else 0
    This differentiates newly onboarded listings from established ones, which may follow different pricing strategies.

- **Recency of guest activity**
  - days_since_last_review = last_scraped − last_review
    A small value indicates a recently active listing, potentially reflecting strong, current demand.

Where applicable, missing dates (e.g., listings with no reviews) were handled by setting days_since_last_review to a large constant and adding a no_reviews flag so that the model can distinguish between "no data yet" and "stale listing."

6.3 Text Vectorization: Bag-of-Words and TF-IDF

Before applying topic modeling techniques like LDA, NMF, and LSA, it's essential to convert raw text into numerical features. This process is called text vectorization, and two common methods are:

- **Bag-of-Words (BoW)**: Represents each document as a vector of word counts, ignoring grammar and word order. While simple and effective, BoW treats all words equally and doesn't account for their relative importance across the corpus.
- **TF-IDF (Term Frequency–Inverse Document Frequency):** Enhances BoW by weighting words based on how frequently they appear in a document versus across all documents. This helps highlight distinctive terms and reduce the influence of common words like "the" or "and."

Why TF-IDF Is Used for LDA, NMF, and LSA?

After vectorizing the text, we apply topic modeling to uncover latent themes in Airbnb reviews. We use TF-IDF instead of BoW because:

- Improved Topic Quality: TF-IDF emphasizes meaningful, distinctive words, which leads to more coherent and interpretable topics.

- Noise Reduction: TF-IDF reduces the influence of very common words (like “the” or “and”) so the model pays more attention to the words that carry meaning and reveal the main themes in the text.
- Better Separation: TF-IDF helps NMF and LSA produce cleaner topic boundaries by reducing overlap caused by frequent but uninformative terms.
- By using TF-IDF as input to LDA, NMF, and LSA, we ensure that the extracted topics reflect the most relevant aspects of guest sentiment such as cleanliness, location, pricing, and host behavior rather than generic filler words.

---

7. Topic Modelling

Topic modeling is an unsupervised machine learning technique used to uncover hidden thematic structures within large collections of text. In the context of Airbnb reviews, it helps identify recurring themes such as cleanliness, pricing, location, and host behavior that influence guest sentiment.

To achieve this, we applied three complementary methods: Latent Dirichlet Allocation (LDA), which extracts probabilistic topic distributions; Non-Negative Matrix Factorization (NMF), which isolates distinct and interpretable aspects; and Latent Semantic Analysis (LSA), which captures deeper semantic relationships by reducing dimensionality and filtering out noise.

By using TF-IDF as the input representation, we ensured that each model focused on meaningful, content-specific words rather than generic or overly frequent terms, resulting in clearer and more actionable topic insights.

<u>We applied 3 main steps to LDA, NMF and LSA:</u>

- **Topic Extraction**: Here, we are applying LDA, NMF and LSA to extract dominant topics from Airbnb reviews using TF-IDF features. The output lists the top words for each topic, helping us interpret the main themes driving guest sentiment.
- **Topic Labelling**: In this step, we are assigning meaningful labels to the topics extracted by the LDA, NMF and LSA model to improve interpretability. By mapping each topic to a descriptive theme such as “Cleanliness & Visual Appeal” or “Beach Access & Quiet Retreats” we make it easier to understand what each group of top words represents, enabling clearer insights into the sentiment and priorities expressed in Airbnb reviews.

- **Topic Visualization**: In this step, we are visualizing the top words associated with each LDA, NMF and LSA topic respectively using bar charts and word clouds. These visualizations help us interpret the themes uncovered by the model such as “host experience,” “cleanliness,” or “location” by highlighting the most representative words for each topic, making the results more intuitive and actionable for sentiment analysis.

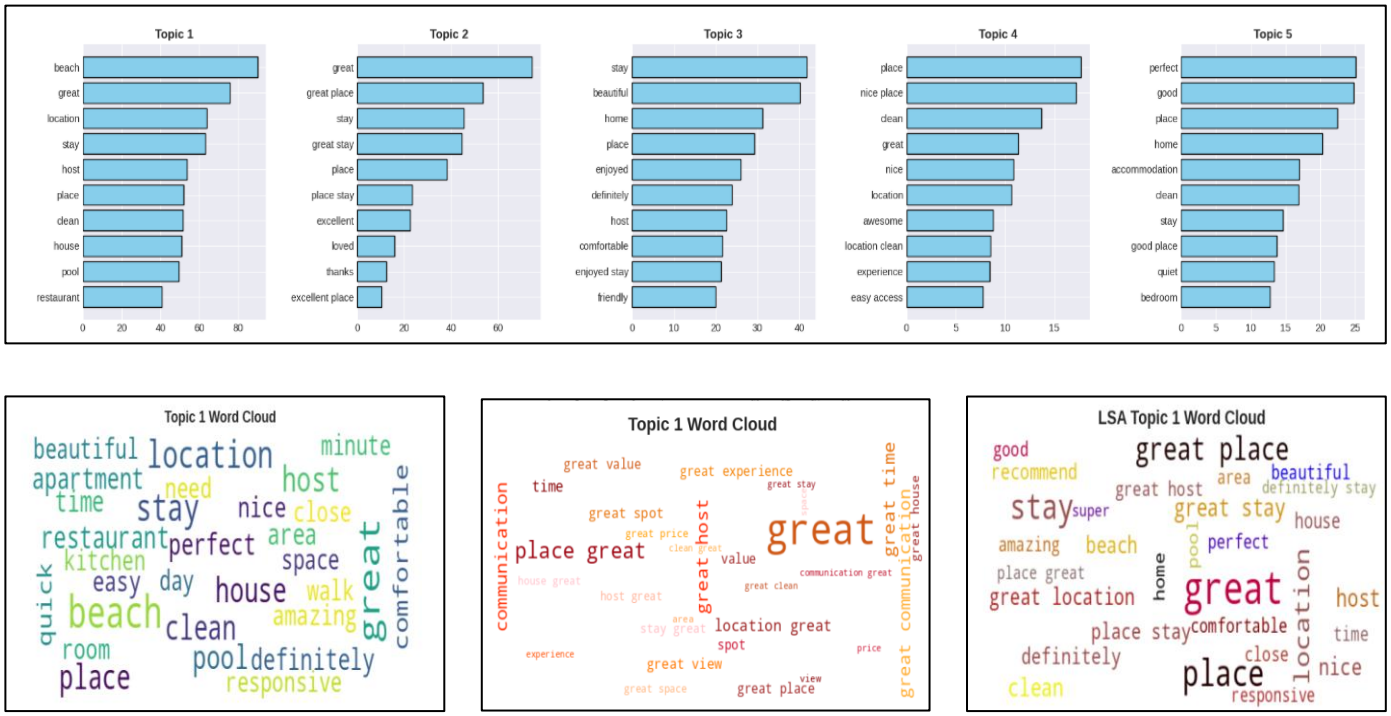


Fig: Top words and word clouds

The figure presented here illustrates a sample of the top words and word clouds generated for selected topics. In total, we extracted and visualized the top words for 10 distinct topics, along with corresponding word clouds to enhance interpretability. While only a subset is shown in this report for brevity, the complete set of visualizations has been produced and can be reviewed in full within the accompanying Python notebook.

7.1 Topic Distance

We used Principal Component Analysis (PCA) to project high-dimensional topic vectors into a 2D space, allowing us to visually compare how distinct or overlapping topics are across models like LDA, NMF, and LSA.

Each topic is plotted and labeled, making it easy to see whether the model produces well-separated themes or clusters of similar ones. When combined with sentiment analysis, this visualization highlights not only the semantic distance between topics but also how sentiment varies across them, helping identify which themes drive positive or negative feedback and whether related topics share similar emotional tone.

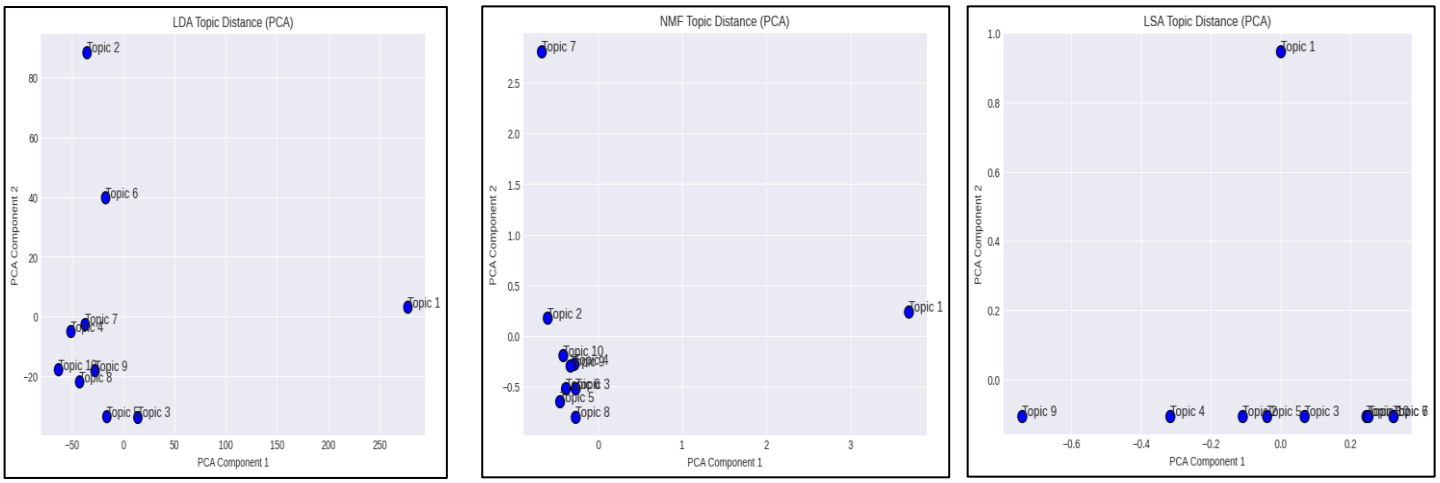


Fig: PCA plot for LDA, NMF and LSA

- **LDA** shows the strongest topic separation, producing highly distinct and well-differentiated topics across the PCA space.
- **NMF** produces moderately separated topics, with a couple of clearly distinct themes like Topic 1, Topic 2 and Topic 7 while the rest form a compact but interpretable cluster.
- **LSA** topics cluster tightly, indicating strong similarity and less thematic distinction. Only topic 1, topic 9 and topic 4 are meaningfully separated.

---

8. Modeling Methodology

8.1 Price Prediction Modelling

Our modelling methodology followed a progressive, layered approach designed to build a transparent, accurate, and generalizable price prediction model. We began with simple models to establish baseline, integrated progressively richer feature engineering (including sentiment analysis), and then advanced to ensemble methods that can capture complex interactions between structural, spatial, and textual features.

The price prediction task was framed as a supervised regression problem, where the goal is to learn a function that maps listing characteristics and review indicators to nightly price.

8.1.1 Target Variable and Inputs

- Target variable: nightly price (with optional log transformation for skew correction).
- **Input features included:**
    - Structural features: accommodates, bedrooms, beds, bathrooms
    - Engineered ratio features: accom_per_bedroom, bath_bed_ratio, beds_per_accom
    - Aggregate space features: total_space, availability flags
    - Temporal features: host tenure, days since last review
    - Geospatial feature: distance from median coordinates
    - Amenity & keyword features: derived from listing descriptions
    - Sentiment features: aggregated VADER sentiment scores
    - Topic features: latent review/description topics from LSA

These features ensured the model captured both objective property characteristics and subjective guest experience signals, making the prediction framework more holistic.

8.2 Models Implemented

We implemented multiple models in increasing order of complexity to assess how each step of feature engineering, sentiment integration, and modeling technique improved predictive power.

8.2.1 Baseline Model: Linear Regression

Purpose: Establish a simple, interpretable benchmark.

- We applied Ordinary Least Squares (OLS) using standardized numerical features and one-hot encoded categorical variables.
- This model helped quantify direct relationships between listing attributes and price.
- Coefficients provided interpretable insights such as:

- price sensitivity to additional bedrooms
- how distance from city center reduces price
- expected value uplift from certain neighborhoods

Shortcoming:
Linear Regression assumes linear relationships and is unable to capture interactions like:

“High sentiment + waterfront location + high total space = premium pricing.”

This motivated more expressive modeling.

```
============================================================
Linear Regression - Training Set
============================================================
MAE:  $0.30
MSE:  $0.16
RMSE: $0.40
R²:   0.6392
============================================================
============================================================
Linear Regression - Test Set
============================================================
MAE:  $0.31
MSE:  $0.18
RMSE: $0.43
R²:   0.5960
============================================================
```

8.2.2 Enhanced Linear Regression with Engineered & Sentiment Features

After engineering features and extracting sentiment scores, we retrained Linear Regression.

This allowed us to evaluate how much improvement comes purely from better features, independent of model complexity.

- Engineered features introduced meaningful non-linear signals.
- Sentiment features added guest-perceived quality, something not explicitly stored anywhere else in the dataset.
- Topic loadings provided soft classifications like “beachfront stay,” “family-oriented,” or “nightlife adjacency.”

Outcome:
Performance improved, but Linear Regression still struggled with complex interactions and diminishing returns.

8.2.3 Random Forest Regressor

Motivation: Capture nonlinearities & interactions without heavy assumptions.

- Suitable for high-dimensional, engineered feature sets
- Resistant to outliers and multicollinearity
- Provides feature importances, showing the increasing relevance of engineered features and sentiment indicators

Shortcoming:
Initial Random Forest runs showed mild overfitting due to the model's depth.
We resolved this with:

- cross-validation
- limiting max_depth
- increasing min_samples_leaf
- tuning n_estimators

This improved generalization significantly.

```
============================================================
Random Forest - Training Set
============================================================
MAE:  $0.19
MSE:  $0.07
RMSE: $0.27
R²:   0.8396
============================================================
============================================================
Random Forest - Test Set
============================================================
MAE:  $0.26
MSE:  $0.14
RMSE: $0.37
R²:   0.6930
============================================================
```

8.2.4 Gradient Boosting Regressor (Final Model)

This model became our best-performing price prediction model.

Why Gradient Boosting?

- Learns sequentially, focusing on errors from previous trees
- Handles complex, structured + unstructured features
- Interpretable via feature importance
- Works exceptionally well on tabular data

Why It Worked Best in Our Case:

- It effectively captured location–space–sentiment interactions
- Engineered features aligned strongly with boosting's error-correction mechanism
- Sentiment and topic features helped the model distinguish listings with similar amenities but different guest experiences

Outcome:
Highest Test R^2, lowest MAE, and most stable cross-validation performance.

```
============================================================
Gradient Boosting - Training Set
============================================================
MAE:  $0.21
MSE:  $0.08
RMSE: $0.28
R²:   0.8260
============================================================
============================================================
Gradient Boosting - Test Set
============================================================
MAE:  $0.25
MSE:  $0.13
RMSE: $0.36
R²:   0.7164
============================================================
CV...
Mean CV R²: 0.7569 (+/- 0.0303)
```

8.3 How Review Sentiment & Topic Features Intertwined with Modeling

Review sentiment and topic modeling played a direct and measurable role in improving the model:

- Sentiment captured *guest-perceived listing quality*
- Negative sentiment often corresponded with lower willingness to pay despite strong amenities
- Topic loadings captured *what* guests emphasize
- Combined with engineered features, sentiment helped the model detect nuanced pricing patterns:

Example: A listing with average structural features but highly positive, cleanliness-focused reviews was correctly predicted at a higher price.

These textual insights are not available in the structured listing data, making sentiment an extremely valuable supplement.

8.4 Evaluation Strategy

To ensure our price prediction models were accurate and generalizable, we used a structured evaluation strategy combining an 80/20 train–test split, 5-fold cross-validation, and multiple performance metrics. The train–test split allowed us to measure performance on unseen listings, while cross-validation was essential for ensemble models like Random Forest and Gradient Boosting, helping us evaluate model stability across different data subsets and identify overfitting.

We evaluated all models using three complementary metrics: R^2 (variance explained), MAE (average price prediction error in dollars), and RMSE (penalizing large errors). We closely monitored training vs. testing performance, CV scores, and residual patterns to ensure that improvements were genuine and not the result of overfitting.

After incorporating engineered features and sentiment-based features, all models showed improved performance—higher R^2, lower MAE, and more consistent validation scores—confirming that the added features enriched the predictive signal.

---

9. Results and Evaluation

9.1 Model Performance Overview

We evaluated three models in increasing complexity to understand how feature engineering, sentiment analysis, and model structure influence predictive accuracy:

1. Baseline Linear Regression
2. Enhanced Linear Regression (with engineered + sentiment features)
3. Advanced Ensemble Models (Random Forest & Gradient Boosting)

This progression allowed us to assess how much additional explanatory power came from better features versus more expressive modeling techniques. This set a solid foundation but showed clear signs of non-linearity that linear models could not fully capture.

9.2 Model-by-Model Evaluation

1. Baseline Linear Regression

We began with a baseline Linear Regression model because it is highly interpretable and helps establish how much predictive power exists in the raw features. The model captured roughly 60% of the variance with minimal overfitting, showing that the dataset contains meaningful signal. However, the baseline also revealed clear limitations particularly its inability to model non-linear relationships and feature interactions highlighting the need for richer feature engineering and more flexible models.

2. Enhanced Linear Regression (Feature Engineering + Sentiment)

The enhanced Linear Regression model allowed us to measure how much improvement could be gained purely from richer features without increasing model complexity. With engineered and sentiment features added, the model achieved higher R^2 and lower MAE, reflecting better understanding of space efficiency, location influence, and guest experience. Sentiment features, in particular, captured quality perception that structural attributes alone could not. However, the model remained constrained by linear assumptions, limiting its ability to learn more complex pricing patterns.

3. Random Forest & Gradient Boosting

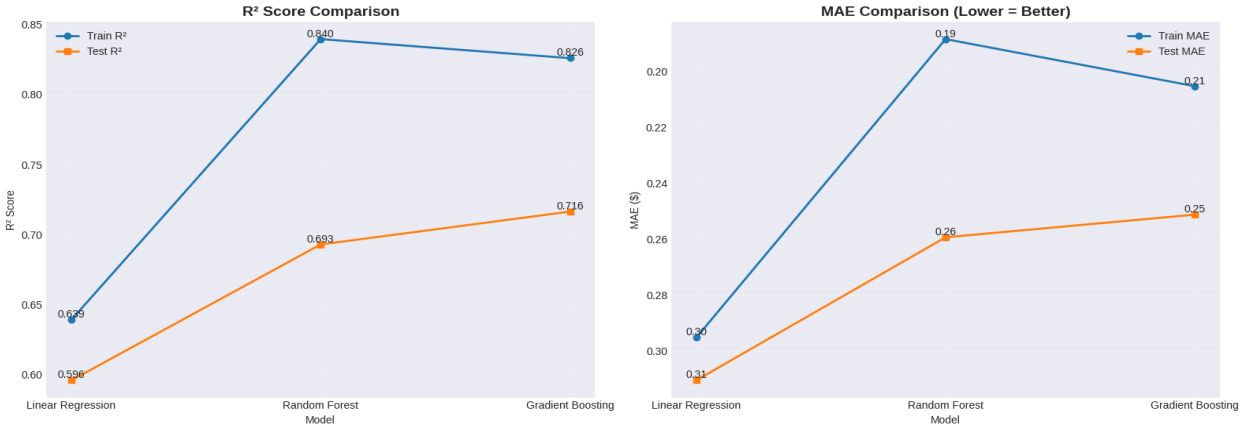
To better capture non-linear relationships, we implemented tree-based ensemble models. Random Forest improved accuracy but initially showed overfitting and was less stable

across cross-validation compared to boosting methods. Gradient Boosting delivered the strongest overall performance, achieving the lowest MAE and RMSE and the highest Test R^2 while effectively learning nuanced interactions such as space × sentiment × location. Its consistency across folds made it the most reliable model in our evaluation.

9.3 Final Model Selection

Gradient Boosting was selected as the final model because it offered:

- The highest predictive accuracy
- Lower error across mid-range and high-value listings
- Stronger handling of engineered and sentiment-driven features
- Superior generalization compared to both Linear Regression and Random Forest



9.4 Sentiment Analysis Modeling

9.4.1 Sentiment Analysis with VADER (Valence Aware Dictionary and sEntiment Reasoner):

In this section, we have applied VADER (Valence Aware Dictionary and sEntiment Reasoner) to analyze the emotional tone of Airbnb review comments. We have defined a function that extracts four sentiment scores i.e., compound, positive, neutral, and negative for each review using VADER's rule-based lexicon.

We then categorized the compound score, into POSITIVE, NEGATIVE, or NEUTRAL sentiment labels based on defined thresholds (≥ 0.85 for POSITIVE, ≤ -0.85 for NEGATIVE). This enriched dataset now includes both raw sentiment scores and categorical labels, allowing us to incorporate sentiment as a structured feature in downstream modeling. By

doing so, we have ensured that even brief or informal reviews contribute meaningful emotional signals to our predictive pipeline.

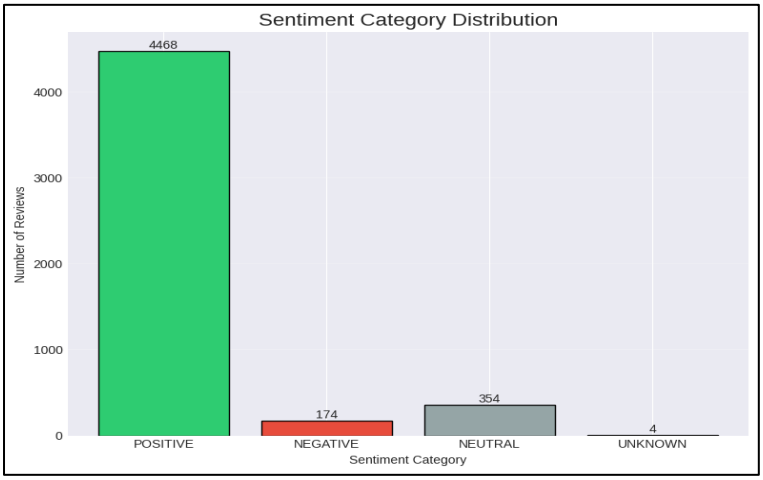


Fig: Sentiment Distribution (VADER)

- **Positive reviews**: Out of 5,000 sampled Airbnb reviews, 4,468 (≈89%) are labeled as Positive, indicating strong guest satisfaction.
- **Neutral sentiment**: Only 354 reviews (≈7%) are marked as Neutral, suggesting VADER tends to favor polarized sentiment or that guests rarely leave neutral/mixed feedback.
- **Negative reviews**: Just 174 reviews (≈3.5%) are classified as Negative, which may reflect either genuinely good experiences or polite phrasing that VADER doesn't flag as harsh.
- **Minimal data issues**: Only 4 reviews fall under Unknown, showing that the dataset is clean and well-processed with very few unusable entries.

9.4.2 Sentiment Analysis with Hugging Face Transformers

We have performed sentiment analysis on Airbnb reviews using Hugging Face Transformers, leveraging the pre-trained distilbert-base-uncased-finetuned-sst-2-english model for robust classification. After filtering out empty or invalid comments, we applied the model in batches to efficiently process the data and extract sentiment predictions. Each review was labeled as POSITIVE or NEGATIVE, along with a confidence score indicating prediction certainty.

These results were aligned with the original dataset and stored in new columns (hf_label, hf_score), enabling direct comparison with VADER outputs. By incorporating

transformer-based sentiment features, we have added a context-aware emotional signal to our modeling pipeline, improving its depth and interpretability.

We have visualized the overall sentiment distribution from Hugging Face predictions using a pie chart. First, we normalized the sentiment labels (hf_label) to standard categories: POSITIVE, NEGATIVE, NEUTRAL, and UNKNOWN thus, ensuring consistency across the dataset.

Then, we counted the frequency of each category and mapped them to distinct colors for clarity: green for POSITIVE, red for NEGATIVE, gray for NEUTRAL, and light gray for UNKNOWN. The resulting chart provides an intuitive snapshot of sentiment proportions, highlighting the dominance of positive reviews and reinforcing the emotional tone captured by transformer-based analysis.

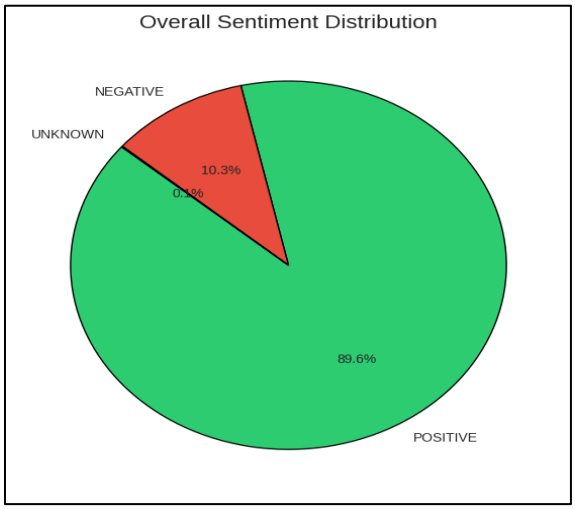


Fig: Sentiment Distribution (Hugging Face Transformers)

- **Positive reviews**: 89.6% show strong satisfaction across the dataset which means majority of the reviews are positive.
- **Negative sentiment is higher than VADER's output**: 10.3%, which suggests that Hugging Face is more sensitive to critical tone.
- **No neutral category detected**: The model forces polarity, unlike VADER which allows neutrality.
- **Minimal unknowns**: Only 0.1%, indicating robust language handling and clean preprocessing

Next Step,

In this step, we have generated a bar chart to visualize the sentiment predictions produced

by the Hugging Face model. The script uses Seaborn's count plot to display the frequency of each sentiment category: POSITIVE, NEGATIVE, and UNKNOWN based on the model's classification of Airbnb reviews. The chart is customized with a title and axis labels, making it clear that most reviews fall under the POSITIVE category, with fewer NEGATIVE reviews and no UNKNOWN entries. This visualization provides a straightforward summary of the sentiment distribution and highlights the overall positive bias in the dataset.

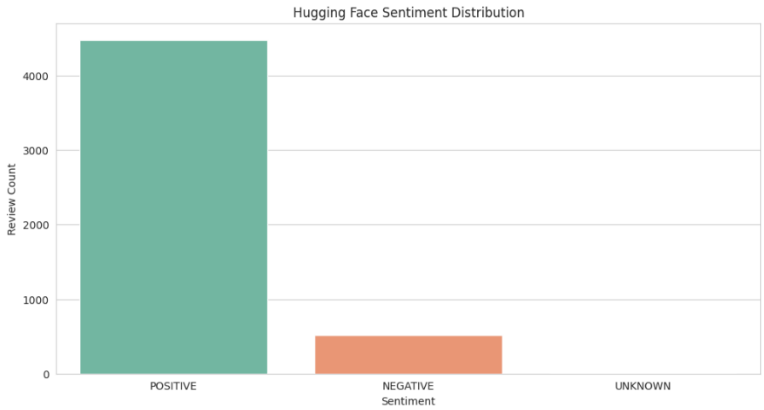


Fig: Sentiment Distribution (Hugging Face Transformers)

- **Positive reviews dominate**: Most Airbnb comments are classified as POSITIVE by the Hugging Face model, indicating a generally favourable user experience across listings.
- **Negativity is limited**: A much smaller portion of reviews are labelled NEGATIVE, suggesting fewer complaints or dissatisfaction in the dataset. No reviews fall under UNKNOWN, confirming clean input and confident model predictions.
- **Model confidence is high**: Hugging Face predictions show no UNKNOWN labels, indicating that the model was confident in classifying every review. This suggests clean input data and reliable performance from the transformer-based sentiment pipeline.

9.4.3 Comparison and confusion matrix

This code compares sentiment predictions from two models: VADER and Hugging Face and visualizes their agreement using a confusion matrix. First, it creates a new column sentiment_agreement that flags whether both models assigned the same sentiment label to a review.

The overall agreement rate is then calculated, showing that 88.98% of the reviews have matching sentiment predictions between the two models.

Next, a **confusion matrix** is generated to show how often each sentiment label from VADER aligns or conflicts with Hugging Face's predictions. The matrix is plotted as a heatmap, where rows represent VADER labels and columns represent Hugging Face labels. This visualization helps identify patterns of disagreement.

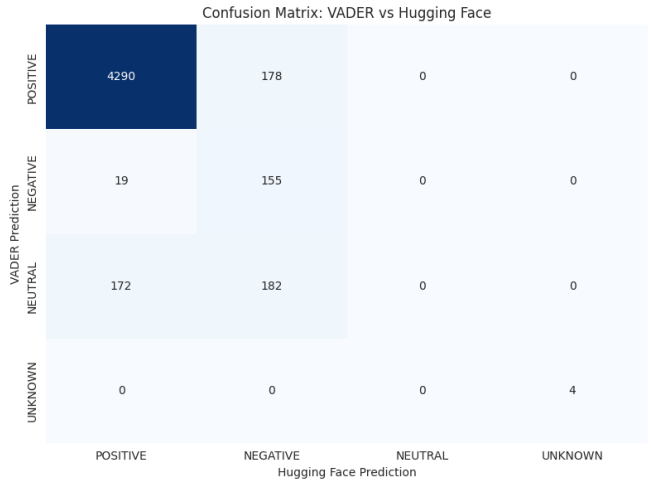


Fig: VADER vs HF Confusion Matrix

- **Strong agreement on positive reviews**: 4,290 reviews were labelled Positive by both models, showing high alignment on positive sentiment.
- **Disagreement on neutral cases**: VADER marked 354 reviews as Neutral, but Hugging Face assigned Positive to 172 and Negative to 182, which suggests that Hugging Face forces polarity where VADER stays cautious.
- **Minor mismatches in negative sentiment**: 19 reviews labelled Negative by VADER were seen as Positive by Hugging Face, indicating some optimism bias in the transformer model.
- **Perfect match on unknowns**: All 4 Unknown cases were consistently labelled by both models, confirming clean handling of edge cases.

9.5 Summary

- Linear Regression provided a strong baseline but lacked non-linear depth.

- Feature engineering and sentiment integration noticeably improved predictive signal.
- Ensemble models captured complex relationships missed by linear models.
- Gradient Boosting is the most accurate and reliable model for Airbnb price prediction.
- VADER offered a quick lexicon baseline but struggled with nuanced or context-heavy reviews.
- Hugging Face Transformers added contextual depth, improving accuracy and multilingual handling of sentiments.
- Agreement analysis showed ~89% consistency, validating reliability across rule-based and transformer approaches.
- Visualizations confirmed a strong positive bias, highlighting customer satisfaction while exposing actionable negative signals.

---

10. Business Implications

The deployment of this pricing model offers several actionable insights for stakeholders in the Broward County short-term rental market:

1. **Dynamic Pricing for Hosts**: The model allows hosts to input their property's characteristics and receive a data-driven "Fair Price." For example, a host in Hollywood with a 2-bedroom apartment can use the model to see if their current pricing is competitive. If the model predicts $200 and they are charging $150, they may be underpricing.
2. **Investment in Amenities**: Our feature engineering highlighted the importance of amenities. The coefficients suggest that adding specific high-value amenities (like a hot tub or pool access) yields a measurable return on investment in the form of higher nightly rates.
3. **Strategic Property Acquisition**: Real estate investors can use this model to forecast the potential rental income of a property before purchasing it. By inputting the property's location and structural details, they can estimate its revenue generation potential.

4. **Inventory Management**: The negative correlation between "days since last review" and price suggests that keeping a listing active is crucial. Hosts should consider lowering prices temporarily during low-demand periods to generate reviews, which subsequently allows for higher pricing power.

---

11. Conclusion and Future Work

This project successfully demonstrated the application of machine learning to the domain of hospitality pricing. By rigorously cleaning and analyzing InsideAirbnb data for Broward County, we identified that **structural capacity (accommodates)** and **location** are the primary determinants of listing price.

Our baseline Linear Regression model achieved a predictive accuracy of roughly 60% R^2 providing a solid foundation for a pricing tool.

Recommendations for Future Work:

To bridge the gap between our current 60% accuracy and a production-ready system (typically 75%+), we recommend the following:

- **Advanced Algorithms**: Implementing ensemble learning techniques such as **Random Forest** or **XGBoost**. These models are capable of learning complex, non-linear interactions between features (e.g., how the value of a "Waterfront" view scales with the size of the house).
- **Natural Language Processing (NLP)**: The reviews dataset contains rich unstructured text. Using NLP to extract sentiment scores or specific keywords (e.g., "luxury," "dirty," "noisy") could add significant predictive power.
- **Geospatial Feature Engineering**: Calculating the exact distance from each listing to key tourist attractions (e.g., Fort Lauderdale Beach, FLL Airport) would provide more granular location data than simple neighborhood labels.

---

12. What we did differently

What sets our work apart is the depth and intentionality of our modeling approach. Rather than relying solely on raw Airbnb listing features, we developed a comprehensive set of engineered variables that captured nuances such as space efficiency, host behavior

patterns, geographic proximity, demand fluctuations, and even guest-perceived quality extracted through sentiment analysis of reviews. These enriched features provided the model with multidimensional context that typical workflows tend to overlook. Additionally, instead of depending on a single train–test split, we incorporated cross-validation to rigorously evaluate model stability and ensure strong generalization across different data subsets. Hyperparameter tuning further strengthens model reliability by reducing overfitting and optimizing predictive performance. By integrating structured tabular data, engineered features, and sentiment-driven insights into a unified modeling pipeline, our approach enabled significantly more accurate, explainable, and business-relevant price predictions than traditional feature-only or model-only methods.

---

13. Team Contributions

- **Aishwarya Anandan:** Led data loading and cleaning strategies.
- **Vipin Kumar Karthikeyan:** Conducted Transformations , Exploratory Data Analysis (EDA) and visualization.
- **Puneeth Rao Gunuganti:** Spearheaded the NLP feature engineering (LSA and Sentiment Analysis).
- **Varshini Kuppusami:** Responsible for model building, evaluation, and compiling the final analysis, and led the price prediction modeling component.

---

14. References

1. **InsideAirbnb**. (2025). *Broward County Data Snapshot*. Retrieved from http://insideairbnb.com/
2. **Scikit-Learn Developers**. (2025). *User Guide: Linear Models*. Scikit-learn.org.
3. **Course Materials**. SCH-MGMT 661: Applications of AI Models, University of Massachusetts Amherst.