Predicting Airbnb Nightly Prices Using Ensemble Data Mining Techniques

Final Project Report for YZV 311E - Data Mining

İbrahim Bancar
Department of Artificial Intelligence and Data Engineering
Istanbul Technical University

Hasan Kan
Department of Artificial Intelligence and Data Engineering
Istanbul Technical University

Alperen Sağlam
Department of Artificial Intelligence and Data Engineering
Istanbul Technical University

***Abstract*—Predicting Airbnb listing prices remains a challenging real-world problem that requires integrating diverse structured and unstructured data sources. This project presents a comprehensive data mining pipeline for nightly price prediction using Airbnb metadata, host characteristics, amenities, and contextual features. We implemented an end-to-end solution encompassing exploratory data analysis, robust preprocessing, strategic feature engineering, and ensemble modeling. Our final approach combines LightGBM and CatBoost models through weighted averaging, achieving a Root Mean Squared Logarithmic Error (RMSLE) of 0.4206 on the Kaggle leaderboard. Beyond predictive performance, this study emphasizes feature quality, preprocessing robustness, and model interpretability through SHAP analysis. The complete implementation is available on our GitHub repository.**

***Index Terms*—Airbnb price prediction, ensemble learning, feature engineering, LightGBM, CatBoost, RMSLE, interpretable machine learning**

I. Introduction

The exponential growth of online rental platforms has transformed the hospitality industry, with Airbnb serving over 4 million hosts across 220+ countries. Accurate price prediction is crucial for both hosts seeking competitive pricing strategies and guests looking for fair accommodation options. However, determining optimal nightly prices involves complex interactions between numerous factors including geographic location, property characteristics, host reputation, amenities, seasonal availability, and guest reviews.

A. Problem Definition

This project addresses the supervised regression problem of predicting the nightly price of Airbnb listings given heterogeneous input features. The dataset comprises 44,000+ training instances with 75+ attributes spanning numerical, categorical, textual, and geospatial domains. The challenge lies not only in achieving high predictive accuracy but also in handling data quality issues, extracting meaningful features from high-cardinality variables, and maintaining model interpretability.

B. Main Contributions

Our primary contributions include:

- **Comprehensive Data Pipeline:** A systematic preprocessing framework handling missing values, outliers, and feature transformations across diverse data types.
- **Strategic Feature Engineering:** Domain-informed feature creation including host reputation metrics, amenity aggregations, and location-based indicators that significantly improved model performance.
- **Robust Ensemble Architecture:** A weighted ensemble combining gradient-boosting models (LightGBM and CatBoost) that achieved superior generalization compared to individual models.
- **Interpretability Analysis:** SHAP-based explanations revealing that location, property type, and host reputation are the primary price drivers.
- **Production-Ready Implementation:** Clean, modular, and reproducible code with comprehensive documentation hosted on GitHub.

II. Related Work

A. Traditional vs. Machine Learning Approaches

Early Airbnb pricing research relied on hedonic pricing models from real estate economics. Camatti et al. [1] conducted a systematic comparison between linear methods (OLS, GLM) and artificial intelligence techniques, demonstrating that non-linear models achieve better generalization on Airbnb data. However, linear models remain valuable for baseline establishment and interpretability. Our work extends this by incorporating modern gradient boosting techniques and ensemble strategies.

B. Spatial and Location-Based Models

Geographic information significantly influences accommodation prices. Gyódi and Nawaro [2] analyzed 10 major European cities using spatial econometric models (SAR), revealing strong spatial dependencies that standard regression models cannot capture. Akalın and Alptekin [3] demonstrated that adding location-based features (proximity to city center, transportation hubs) substantially reduces prediction error in

Istanbul's Airbnb market. Our methodology incorporates similar location-aware features while focusing on model-agnostic approaches that generalize across different cities.

C. Text Mining and Sentiment Analysis

Unstructured textual data offers valuable predictive signals. Kalehbasti et al. [4] extracted sentiment and topic-based features from listing descriptions and reviews, improving prediction accuracy. Meijer [5] applied Latent Dirichlet Allocation (LDA) for topic modeling on guest reviews, showing that derived themes provide explanatory power beyond basic property features when integrated with XGBoost and SVR models. While our current implementation focuses on structured data, future extensions could incorporate similar text mining techniques.

D. Model Interpretability

High-performance ensemble models like XGBoost and LightGBM are often criticized as "black boxes." Lundberg and Lee [8] introduced SHAP (SHapley Additive exPlanations), a game-theoretic framework providing consistent feature importance explanations at both global and local levels. We adopt SHAP analysis to balance predictive performance with interpretability, identifying which features drive price predictions for individual listings.

E. Research Gap

While existing literature extensively explores individual modeling techniques, few studies systematically compare modern gradient boosting frameworks or investigate ensemble strategies for Airbnb price prediction. Our work addresses this gap by: (1) comparing LightGBM, CatBoost, and their ensemble on the same dataset, (2) providing detailed feature engineering insights, and (3) emphasizing reproducibility through public code sharing.

III. Dataset and Problem Setup

A. Dataset Overview

The competition dataset consists of Airbnb listings with the following characteristics:

- **Training Set:** 44,317 instances
- **Test Set:** 11,080 instances
- **Features:** 75 original attributes
- **Target Variable:** `price` (nightly rental cost in USD)

B. Feature Categories

The dataset exhibits significant heterogeneity across multiple dimensions:

Numerical Features: Continuous variables like `accommodates`, `bedrooms`, `bathrooms`, `beds`, `minimum_nights`, `maximum_nights`, and various review scores.

Categorical Features:

- Low cardinality: `room_type` (4 values), `instant_bookable` (2 values)
- Medium cardinality: `property_type` (87 values), `neighbourhood_cleansed` (223 values)
- High cardinality: `host_id` (31,149 unique hosts), `amenities` (40,000+ unique combinations)

Textual Features: Long-form text fields including `name`, `description`, `neighborhood_overview`, and `host_about`.

Temporal Features: Date fields like `host_since`, `first_review`, `last_review`, and `last_scraped`.

Geospatial Features: Continuous `latitude` and `longitude` coordinates.

C. Data Quality Issues

Several characteristics posed significant modeling challenges:

1) **Missing Values:** Critical features like `bedrooms` (6.2%), `bathrooms` (6.2%), and `beds` (2.3%) contained substantial missingness. Text fields like `host_about` (37%) and `neighborhood_overview` (41%) had higher missingness rates.
2) **Skewed Distributions:** The target variable `price` exhibited strong right skewness (skewness = 5.23), with extreme outliers reaching $10,000+ per night while the median was $95.
3) **High Cardinality:** Variables like `amenities` contained thousands of unique combinations, requiring sophisticated encoding strategies.
4) **Outliers:** Extreme values in `maximum_nights` (up to 1,125 days) and `host_listings_count` (hosts managing 500+ properties) required careful treatment.
5) **Imbalanced Categories:** Some property types (e.g., "Entire rental unit") dominated with 35,000+ instances, while others had only single-digit representation.

D. Evaluation Metric

Following the Kaggle competition specification, model performance is evaluated using Root Mean Squared Logarithmic Error (RMSLE):

$$\text{RMSLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1) - \log(a_i+1))^2} \quad (1)$$

where p_i represents predicted prices, a_i represents actual prices, and n is the number of instances. RMSLE penalizes relative errors rather than absolute errors, making it robust to extreme values and more suitable for skewed price distributions. We complemented RMSLE with Mean Absolute Error (MAE) and R^2 for comprehensive performance assessment.

IV. Methodology

Our modeling pipeline consisted of five sequential stages: data preprocessing, feature engineering, baseline model development, advanced ensemble modeling, and validation.

A. Data Preprocessing

1) Missing Value Treatment: We adopted a feature-specific imputation strategy:

- **Numerical features:** Median imputation for `bedrooms`, `bathrooms`, and `beds` to maintain robustness against outliers.
- **Categorical features:** Mode imputation for `property_type` and `neighbourhood_cleansed`.
- **Percentage features:** `host_response_rate` and `host_acceptance_rate` were converted from strings to numeric ratios [0,1], with missing values imputed to the median.
- **Text features:** Missing values in `description`, `host_about`, and `neighborhood_overview` were filled with placeholder text "Not Provided" for potential future text mining.
- **Target variable:** Rows with missing `price` values (124 instances) were removed as they represented target loss.

2) Outlier Detection and Treatment: Rather than aggressive outlier removal, we applied robust transformations:

- Log transformation of the target variable: log_price $= \log(price + 1)$ to reduce skewness and stabilize variance.
- Winsorization of extreme values in `maximum_nights` (capped at 99th percentile = 365 days).
- Capping `host_listings_count` at the 95th percentile to prevent influence from super-hosts managing hundreds of properties.

3) Encoding Strategies: Different encoding approaches were applied based on cardinality:

- **Binary encoding:** For `instant_bookable`, `host_is_superhost`, `host_has_profile_pic`, and `host_identity_verified`.
- **One-hot encoding:** For low-cardinality variables like `room_type` (4 categories).
- **Frequency encoding:** For medium-to-high cardinality features like `property_type` and `neighbourhood_cleansed`, replacing categories with their occurrence frequency.
- **Target encoding:** Applied to `neighbourhood_cleansed` with 5-fold cross-validation to prevent overfitting, replacing neighborhoods with their mean target value.

4) Feature Scaling: Continuous features were standardized using StandardScaler for linear models. Tree-based models (LightGBM, CatBoost) naturally handle unscaled features, so we maintained raw values for these algorithms to preserve interpretability.

B. Feature Engineering

Strategic feature creation significantly enhanced model performance:

1) Host Reputation Indicators: We aggregated host-related features into composite metrics:

- `host_quality_score`: Weighted combination of `host_response_rate`, `host_acceptance_rate`, and binary superhost status.
- `host_experience_days`: Days since `host_since` date, capturing host tenure.
- `host_listings_density`: Ratio of `host_total_listings_count` to `host_experience_days`, indicating host activity level.

2) Amenity Features: Rather than encoding 40,000+ unique amenity combinations, we created interpretable aggregations:

- `amenity_count`: Total number of amenities listed.
- `has_wifi`, `has_kitchen`, `has_parking`, `has_tv`: Binary indicators for high-impact amenities.
- `luxury_amenity_score`: Count of premium amenities (pool, gym, hot tub, etc.).

3) Location-Based Features: Geographic coordinates were enriched with spatial context:

- `distance_to_center`: Euclidean distance from city center (computed using median latitude/longitude).
- `neighbourhood_price_ratio`: Ratio of listing price to neighborhood median price (derived post-hoc for validation analysis).

4) Review and Availability Metrics: Temporal and feedback features were engineered:

- `review_recency_days`: Days since `last_review`.
- `review_frequency`: `number_of_reviews` divided by `host_experience_days`.
- `availability_ratio`: `availability_30` / 30, indicating calendar occupancy.

5) Property Configuration Features: Interaction terms captured property characteristics:

- `beds_per_bedroom`: Ratio indicating room density.
- `bathrooms_per_bedroom`: Luxury indicator.
- `is_entire_place`: Binary flag for entire property rentals (derived from `room_type`).

In total, we engineered 23 additional features, expanding the feature space to 98 dimensions.

C. Baseline Models

We established performance baselines using interpretable linear models:

1) Ridge Regression: A regularized linear model with L2 penalty to prevent overfitting. Hyperparameters were tuned using 5-fold cross-validation, testing $\alpha \in [0.01, 0.1, 1.0, 10, 100]$. Best performance: RMSLE = 0.5634 (local validation).

2) Lasso Regression: L1-regularized regression for feature selection. The optimal $\alpha = 0.01$ achieved RMSLE = 0.5698, performing slightly worse than Ridge but providing automatic feature selection (34 features zeroed out).

Key Insight: Linear models struggled with non-linear relationships between location, amenities, and price, motivating tree-based approaches.

D. Advanced Ensemble Models

1) LightGBM: LightGBM (Light Gradient Boosting Machine) [6] was selected for its efficiency and handling of categorical features. We employed Optuna for Bayesian hyperparameter optimization over 100 trials:

Search Space:

- `num_leaves`: [20, 50, 100, 150]
- `learning_rate`: [0.01, 0.05, 0.1]
- `max_depth`: [5, 10, 15, 20]
- `min_child_samples`: [10, 20, 50, 100]
- `subsample`: [0.6, 0.7, 0.8, 0.9, 1.0]
- `colsample_bytree`: [0.6, 0.7, 0.8, 0.9, 1.0]
- `reg_alpha`: [0, 0.01, 0.1, 1.0]
- `reg_lambda`: [0, 0.01, 0.1, 1.0]

Best Configuration:

- num_leaves: 100
- learning_rate: 0.05
- max_depth: 15
- n_estimators: 1000 (with early stopping)
- subsample: 0.8
- colsample_bytree: 0.8
- reg_alpha: 0.01, reg_lambda: 0.1

Performance:

- Local CV RMSLE: 0.4289
- Kaggle Leaderboard RMSLE: 0.43268

2) CatBoost: CatBoost [7] was chosen for its native handling of categorical features without extensive preprocessing. Similar Optuna optimization was performed:

Best Configuration:

- depth: 8
- learning_rate: 0.03
- iterations: 1500
- l2_leaf_reg: 3.0
- border_count: 128
- bagging_temperature: 0.5

Performance:

- Local CV RMSLE: 0.4251
- Kaggle Leaderboard RMSLE: 0.42783

CatBoost outperformed LightGBM both locally and on the leaderboard, likely due to superior categorical feature handling.

3) Ensemble Strategy: To leverage the complementary strengths of both models, we implemented a weighted averaging ensemble:

$$\hat{y}_{\text{ensemble}} = 0.5 \cdot \hat{y}_{\text{LightGBM}} + 0.5 \cdot \hat{y}_{\text{CatBoost}} \tag{2}$$

Equal weights were chosen after cross-validation experiments showed minimal improvement with optimized weights (tested ratios: 0.3:0.7, 0.4:0.6, 0.5:0.5, 0.6:0.4). The ensemble approach reduces variance and improves generalization.

Final Ensemble Performance:

- Kaggle Leaderboard RMSLE: **0.4206**

This represents a 1.3% improvement over the best individual model (CatBoost: 0.42783).

E. Validation Strategy

We employed rigorous validation protocols:

- **5-Fold Stratified Cross-Validation:** Data was split into 5 folds with stratification based on price quantiles to ensure balanced target distribution.
- **Out-of-Fold Predictions:** Models were trained on 4 folds and validated on the held-out fold, with predictions aggregated across all folds for unbiased performance estimation.
- **Early Stopping:** Gradient boosting models used 100 rounds of early stopping based on validation RMSLE to prevent overfitting.
- **Holdout Test Set:** Final ensemble was evaluated on Kaggle's hidden test set to assess real-world generalization.

V. Experimental Results

A. Model Comparison

Table I summarizes the performance of all models:

TABLE I
Model Performance Comparison

Model	Local CV RMSLE	Kaggle RMSLE	MAE
Ridge Regression	0.5634	0.5712	38.24
Lasso Regression	0.5698	0.5789	39.87
LightGBM	0.4289	0.43268	26.15
CatBoost	0.4251	0.42783	25.43
Ensemble	**N/A**	**0.4206**	**25.12**

B. Result Interpretation

1) Linear vs. Tree-Based Models: The substantial performance gap between linear baselines (RMSLE ≈ 0.57) and gradient boosting models (RMSLE ≈ 0.42) confirms the presence of strong non-linear relationships in the data. Linear models assume additive feature effects, failing to capture interactions such as:

- Location × Property Type (luxury apartments in city centers vs. budget rooms in suburbs)
- Bedrooms × Bathrooms (interaction indicating property quality)
- Host Experience × Superhost Status (reputation effects)

2) LightGBM vs. CatBoost: CatBoost's slight edge over LightGBM (0.42783 vs. 0.43268) can be attributed to:

1) **Categorical Feature Handling:** CatBoost's ordered target statistics for encoding high-cardinality features like `neighbourhood_cleansed` proved superior to LightGBM's native categorical support.
2) **Regularization:** CatBoost's symmetric tree structure and leaf-wise regularization reduced overfitting on the test set.
3) **Robustness:** CatBoost's default parameters are more conservative, requiring less hyperparameter tuning.

3) Ensemble Effectiveness: The ensemble's 1.3% RMSLE improvement demonstrates successful variance reduction. By averaging predictions from two models with different inductive biases:

- LightGBM's leaf-wise growth captured complex patterns in dense data regions.
- CatBoost's level-wise growth provided stable predictions in sparse regions.

This complementarity is particularly valuable in heterogeneous datasets where no single model excels uniformly.

C. Feature Importance Analysis

Using SHAP (SHapley Additive exPlanations), we analyzed feature contributions to model predictions. The top 10 most influential features were:

1) `neighbourhood_cleansed` (Target-Encoded)
2) `room_type`
3) `accommodates`
4) `bedrooms`
5) `latitude` / `longitude`
6) `bathrooms`
7) `property_type`
8) `host_is_superhost`
9) `review_scores_rating`
10) `amenity_count`

Key Insights:

- **Location Dominance:** Neighborhood and geographic coordinates accounted for ≈35% of predictive power, validating spatial econometric literature.
- **Property Configuration:** Structural features (bedrooms, bathrooms, accommodates) contributed ≈25%, with strong interaction effects (e.g., more bedrooms command higher prices primarily in entire homes).
- **Host Reputation:** Superhost status increased predicted prices by ≈8-12% on average, controlling for other features.
- **Amenities:** While individual amenities had weak effects, the aggregate count showed a moderate positive relationship (each additional amenity increased price by ≈1-2%).

D. Error Analysis

We investigated prediction errors by analyzing residuals:

1) Overestimation Cases: The model tended to overpredict prices for:

- **Budget Listings in Prime Locations:** Hosts intentionally pricing below market rate for faster bookings.
- **Properties with Misleading Descriptions:** Listings advertising luxury amenities but lacking actual quality (detectable only through guest reviews, not metadata).

2) Underestimation Cases: Underprediction occurred for:

- **Unique Properties:** Unusual property types (e.g., treehouses, boats) with few training examples.
- **Seasonal Premium Listings:** Properties in tourist destinations during peak season, where temporal effects weren't captured in static features.

3) High Residual Examples: Listings with RMSLE > 0.8 (top 5% of errors) often exhibited:

- Extreme luxury properties ($500+/night) with sparse representation in training data.
- Inconsistent data quality (e.g., missing critical features like bedrooms).
- Outlier pricing strategies (e.g., hosts testing market rates).

E. Cross-Validation Stability

To assess model stability, we examined RMSLE variance across folds:

- **LightGBM:** Mean CV RMSLE = 0.4289, Std = 0.0032 (stable)
- **CatBoost:** Mean CV RMSLE = 0.4251, Std = 0.0028 (very stable)

Low variance indicates robust generalization, with CatBoost showing slightly better consistency.

VI. Conclusion and Future Work

A. Key Findings

This project successfully developed an interpretable ensemble model for Airbnb price prediction, achieving competitive performance (RMSLE = 0.4206) while maintaining transparency through SHAP analysis. Our main findings include:

1) **Feature Engineering Impact:** Strategic feature creation (host reputation, amenity aggregations, location metrics) yielded ≈15% RMSLE improvement over raw features.
2) **Model Selection:** CatBoost demonstrated superior performance for high-cardinality categorical features, while ensemble averaging provided additional robustness.
3) **Location Primacy:** Geographic features (neighborhood, coordinates) dominated predictions, aligning with real estate economics literature.
4) **Interpretability:** SHAP analysis revealed intuitive relationships (e.g., more bedrooms → higher price in entire homes), validating model trustworthiness.

B. Lessons Learned from Data Mining Perspective

1) Data Quality Matters More Than Model Complexity: Extensive preprocessing (missing value imputation, outlier treatment, encoding strategies) provided larger gains than hyperparameter optimization. Investing 40% of project time in data cleaning was well-justified.

2) Feature Engineering Requires Domain Knowledge: Generic automated feature generation (e.g., polynomial terms) was less effective than domain-informed features (host reputation scores, amenity categories). Understanding the hospitality domain guided successful feature creation.

3) Ensemble Diversity is Key: Combining models with different biases (LightGBM's leaf-wise vs. CatBoost's level-wise) yielded better results than stacking similar models. Diversity in learning algorithms is more valuable than stacking depth.

4) Cross-Validation Prevents Overfitting: Rigorous 5-fold CV with early stopping prevented the ≈8% generalization gap observed in preliminary models without proper validation.

C. Model Limitations and Failure Modes

Our model performs poorly in the following scenarios:

- **Sparse Property Types:** Rare categories like "cave" or "castle" lack sufficient training data, leading to predictions regressing toward global means.
- **Temporal Dynamics:** Static features cannot capture seasonal price fluctuations (e.g., higher rates during holidays), resulting in underestimation during peak periods.
- **Text Signal Loss:** Ignoring listing descriptions discards valuable semantic information about property uniqueness and quality.
- **New Neighborhoods:** Test listings in neighborhoods absent from training data receive unreliable predictions due to lack of localized pricing information.
- **Luxury Segment:** Extreme high-end properties ($1000+/night) are systematically underestimated due to class imbalance and log transformation dampening extreme values.

D. Future Work

Several promising directions could further improve performance:

1) Advanced Text Mining:

- **Transformer-Based Embeddings:** Use BERT or domain-adapted models to encode `description` and `host_about` into semantic vectors.
- **Sentiment Analysis:** Extract sentiment scores from guest reviews using VADER or fine-tuned models to capture property reputation beyond star ratings.
- **Topic Modeling:** Apply LDA to identify latent themes (e.g., "family-friendly", "luxury", "budget") in descriptions.

2) Temporal Features:

- Incorporate seasonality indicators (month, day-of-week, holiday flags) if booking date data becomes available.
- Model price trends over time using time series features or recurrent neural networks.

3) External Data Integration:

- **POI Proximity:** Distance to landmarks, restaurants, entertainment venues (via APIs like Google Places).
- **Transportation Access:** Proximity to public transit stations, airports.
- **Crime Statistics:** Neighborhood safety scores from public datasets.
- **Economic Indicators:** Local median income, tourism statistics.

4) Model Architecture Extensions:

- **Neural Networks:** Explore deep learning with entity embeddings for high-cardinality categoricals.
- **Stacking:** Train meta-models (e.g., Ridge regression) on base model predictions for hierarchical ensembles.
- **Geospatial Models:** Implement Gaussian Process Regression or Spatial Autoregressive models to explicitly model spatial dependencies.

5) Fairness and Bias Analysis: Investigate potential discriminatory patterns in pricing across protected attributes (e.g., neighborhood demographics) and develop fairness-aware models to ensure equitable predictions.

VII. Individual Contributions

This project was a collaborative effort with clear role delineation:

A. İbrahim Bancar - Data Exploration & Preprocessing

- Conducted comprehensive exploratory data analysis (EDA) including univariate, bivariate, and multivariate analysis.
- Designed and implemented the preprocessing pipeline: missing value treatment, outlier detection, feature scaling, and encoding strategies.
- Performed data quality assessment, identifying inconsistencies and proposing resolution strategies.
- Created visualizations for data understanding (distributions, correlations, geographic plots).
- Documented all preprocessing decisions and maintained clean data documentation.
- Estimated effort: ≈35% of total project work.

B. Hasan Kan - Feature Engineering & Modeling

- Designed and implemented 23 engineered features (host reputation, amenities, location metrics).
- Developed baseline models (Ridge, Lasso) with hyperparameter tuning.
- Implemented LightGBM and CatBoost models with Optuna-based optimization.
- Conducted feature importance analysis using SHAP.
- Managed cross-validation experiments and early stopping strategies.
- Optimized model performance through iterative feature selection and hyperparameter refinement.
- Estimated effort: ≈35% of total project work.

C. Alperen Sağlam - Evaluation & Integration

- Designed evaluation framework with RMSLE, MAE, and R^2 metrics.
- Implemented the weighted ensemble averaging strategy.
- Conducted error analysis, identifying model failure modes and high-residual cases.
- Created performance comparison tables and visualizations for the report.
- Managed GitHub repository: code organization, documentation, version control, and reproducibility.
- Integrated final outputs, prepared Kaggle submissions, and coordinated team workflow.
- Drafted significant portions of the final report and presentation materials.
- Estimated effort: ≈30% of total project work.

All team members participated equally in weekly meetings, literature review, and strategic decision-making. The GitHub repository (https://github.com/ibrahim-bancar/YZV311_2526_

7) demonstrates consistent contributions from all members throughout the project timeline.

ACKNOWLEDGMENTS

We thank the YZV 311E course instructors and teaching assistants (Yaren Yılmaz Kargılı, Erhan Biçer, Abdullah Kavaklı) for their guidance and feedback throughout the project. We also acknowledge Kaggle for hosting the competition and Airbnb for providing the open dataset.

REFERENCES

[1] N. Camatti, G. di Tollo, G. Filograsso, and S. Ghilardi, "Predicting Airbnb pricing: a comparative analysis of artificial intelligence and traditional approaches," *Computational Management Science*, vol. 21, article 30, 2024.

[2] K. Gyódi and Ł. Nawaro, "Determinants of Airbnb prices in European cities: A spatial econometrics approach," *Tourism Management*, vol. 86, 104319, 2021.

[3] O. Akalın and G. I. Alptekin, "Enhancing Airbnb Price Predictions with Location-Based Data: A Case Study of Istanbul," in *Proc. 19th FedCSIS*, 2024, pp. 171–178.

[4] P. R. Kalehbasti, K. Neshat, and H. Sheikhzadeh, "Airbnb Price Prediction Using Machine Learning and Sentiment Analysis," arXiv:1907.12665, 2019.

[5] N. C. A. Meijer, "Improving Airbnb listing price prediction with sentiment analysis, review recency and topic modelling," Master's thesis, Tilburg University, 2022.

[6] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 3146–3154.

[7] L. Prokhorenkova et al., "CatBoost: unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems 31*, 2018, pp. 6638–6648.

[8] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 4765–4774.