Housing Price Prediction System Based on Historical Transaction Data

Ten Cua Ban
Department of Computer Science
University Name
City, Country
email@example.com

***Abstract*—In the modern real estate market, accurate property valuation is a critical determinant for both buyers and sellers to maximize economic returns. This project aims to develop a Machine Learning model to predict housing prices based on input features such as area, location (district), number of bedrooms, number of bathrooms, and legal status. Data was automatically harvested from nhatot.com utilizing the crawl4ai tool. Through a structured methodology comprising data collection, preprocessing, model selection, and refinement, a robust predictive model was constructed. Random Forest, LightGBM, and CatBoost algorithms were trained and evaluated. The model enhancement strategy involved employing the Optuna library for hyperparameter tuning. The results demonstrate the superiority of Boosting models in handling complex tabular data.**

***Index Terms*—Housing Price Prediction, Machine Learning, CatBoost, LightGBM, Random Forest, Real Estate Valuation.**

I. Introduction

A. Motivation

Real estate valuation is a critical task for various stakeholders, including buyers, sellers, and investors. Traditional methods of valuation often rely on manual appraisal, which can be subjective and time-consuming.

Imagine you are a real estate investor or a first-time homebuyer with a big question: what truly drives the price of a property? Is it merely the location, the square footage, or the number of bedrooms? The answer is often a complex interplay of non-linear factors. This project aims to solve this uncertainty by leveraging data-driven approaches.

B. Rationale

We hypothesize that the integration of tree-based machine learning models and Bayesian optimization techniques (Optuna) will yield superior performance compared to traditional methods.

Specifically, CatBoost's robust handling of categorical data and LightGBM's computational efficiency are expected to effectively address the heterogeneity inherent in the Vietnamese real estate market data.

C. Overview

In this project, we follow a structured methodology to build and evaluate our predictive model. We first collect a large dataset of historical housing transactions.

We then pre-process the data to handle missing values, outliers, and specifically geospatial categorical variables. We perform data analysis to explore the data and understand its characteristics.

We then select several machine learning algorithms suitable for regression tasks, such as Random Forest, LightGBM, and CatBoost. We train and test our models using cross-validation and compare their performance using metrics such as R^2 and MAPE. The resulting model offers promising results and can be used to predict housing prices with high accuracy.

II. Literature Review

In this section, we will see the definitions of Label Encoder, One-hot encoder, RandomSearch optimzed by Optuna library, Train Test Split and provide some literature content on the models which we are going to use. We will also explain the evaluation metrics 'R² score' and 'RMSE'.

A. Label Encoder

Label Encoder is a utility method to convert categorical data into numerical data. It assigns each unique category in the data to an integer value, making the data more suitable for algorithmic processing.

B. One-hot Encoder

One-hot encoding is a technique used to transform categorical data into a binary numerical format that machine learning algorithms can process. This process involves creating new columns for each unique category, assigning a value of 1 to the corresponding category and 0 to the rest. The objective is to enable machine learning models to effectively interpret and utilize label-based variables.

C. RandomSearch optimzed by Optuna library

Random Search selects hyperparameter combinations stochastically from a defined distribution, making it more efficient than Grid Search for high-dimensional spaces. When implemented via Optuna, this technique is significantly enhanced by Pruning. Optuna monitors intermediate training results and automatically terminates unpromising random trials early. This intelligent resource management allows the model to explore a larger volume of hyperparameter candidates within a fixed computational budget compared to traditional random search.

D. Train Test Split

The 'train test split' function from the 'sklearn.model selection' module is a utility that divides a dataset into randomized training and testing subsets. Each subset is distinct, meaning that no data point can be present in both subsets. This allows for the model to be trained on one subset of the data, and then validated on an entirely separate subset. In our case, we applied an 80/20 split on our dataset for training and testing our models.

E. Random Forest

Random Forest is a machine learning method that functions by constructing an ensemble of multiple decision trees during the training phase. For regression tasks, the output is the mean value of the individual trees. This technique helps reduce variance and prevents overfitting.

F. Gradient Boosting Frameworks

Unlike Random Forest (Bagging), Boosting builds models sequentially.

- **LightGBM:** A gradient boosting framework that utilizes tree-based learning algorithms. Unlike other algorithms that grow trees level-wise, LightGBM adopts a leaf-wise growth strategy. It is designed to be distributed and efficient with faster training speed and higher.
- **CatBoost:** A high-performance library for gradient boosting on decision trees. It is particularly powerful for data with categorical features, using an algorithm called Ordered Boosting to prevent target leakage.

G. Evaluation Metrics

- **R^2 score:** Also known as the coefficient of determination, is a statistical measure that shows the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. It provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model.
- **RMSE:** Root Mean Squared Error (RMSE) is a widely used metric for evaluating regression models. It measures the average magnitude of the errors between the predicted values and the actual values. Because the errors are squared before being averaged, RMSE gives a relatively high weight to large errors, making it particularly useful when large prediction errors are undesirable.

III. Data Acquisition Strategy

Instead of relying on static, pre-existing datasets which may be outdated, we implemented a dynamic data acquisition strategy to harvest real-time data from actual market sources.

A. Web Crawling Implementation

We developed an automated web scraping pipeline using the *Crawl4AI* library to collect real-time real estate data from *Nhatot.com*—one of the largest real estate portals in Vietnam. The rationale behind this approach is to ensure the model is trained on the latest market data, accurately reflecting economic fluctuations and supply-demand dynamics that legacy datasets fail to capture.

The data fields extracted are categorized in Table I.

TABLE I
EXTRACTED DATA FIELDS

Category	Data Fields
Location	City, District, Ward
Physical	Area (m^2), Bedrooms, Bathrooms, Floors, Width, Length, Direction
Financial	Selling Price
Legal & Status	House Type, Legal Documents, Furnishing Status

B. Crawling Techniques

The crawler was engineered with advanced techniques to maximize efficiency and reliability:

- **Asynchronous Crawling:** Utilizing *asyncio* to crawl multiple pages simultaneously, significantly reducing data collection time from hours to minutes.
- **Browser Stealth Mode:** Configuration of User-Agent, viewport, and other parameters to mimic human behavior and bypass anti-bot mechanisms.
- **JSON-LD Extraction:** Extracting listing URLs directly from structured data rather than manually parsing raw HTML, ensuring higher accuracy.
- **Concurrency Control:** Limiting the number of simultaneous requests (maximum 10) to prevent server overload.
- **Checkpointing (Periodic Saving):** Periodically saving data to disk to prevent data loss in the event of network failures or system crashes.

C. Performance

As a result of this pipeline, we successfully collected **7,669 listings** across 310 search pages, with an average processing time of approximately 2 seconds per listing.

IV. Data Analysis and Preprocessing

To ensure model efficacy, we applied a "Clean-First, Transform-Later" strategy. This section details the rigorous procedures applied to the raw data.

A. Dataset Overview

The housing price dataset was collected from the *NhaTot* real estate platform, comprising 7,669 initial entries with 13 attributes describing property characteristics including area, dimensions, room counts, location, and legal documentation status. The target variable is the property price in billion Vietnamese Dong (VND). Through systematic preprocessing, the final curated dataset contains **5,972 samples** with **18 engineered features** suitable for regression modeling.

B. Initial Data Quality Assessment

Prior to preprocessing, a comprehensive assessment of data quality was conducted to identify key challenges that would inform subsequent cleaning strategies.

1) Missing Value Analysis: As shown in Table II, the analysis revealed significant missingness in several key attributes, particularly *Hng* (Direction, 74.11% missing) and *Tình trng ni tht* (Furnishing Status, 47.80% missing). This necessitated specialized imputation strategies to preserve data integrity while maintaining sufficient sample size.

TABLE II
MISSING VALUES DISTRIBUTION IN RAW DATASET

Feature	NaN Count	NaN Percentage (%)
Huong	4,740	74.11%
Tinh trang noi that	3,057	47.80%
So tang	2,183	34.13%
So phong ve sinh	1,759	27.50%
Chieu dai (m)	1,498	23.42%
Chieu ngang (m)	1,212	18.95%
So phong ngu	140	2.19%
Phuong/Xa	1	0.02%
Dien tich (m^2)	1	0.02%
Giay to phap ly	1	0.02%
Gia (ty VN)	1	0.02%
Thanh pho	0	0.00%
Loai hinh	0	0.00%

2) Distribution Analysis and Outlier Detection: Visual inspection of numerical variables revealed pronounced long-tail distributions across multiple dimensions. As illustrated in Fig. 1, the distribution of *Dien tich* (m^2) demonstrates a concentration of properties in the lower range (predominantly 10-200 m^2) with a significant rightward skew extending to extreme values exceeding 1,000 m^2. The boxplot analysis identified 419 outliers (6.55%) for area alone based on the conventional $1.5 \times IQR$ criterion.

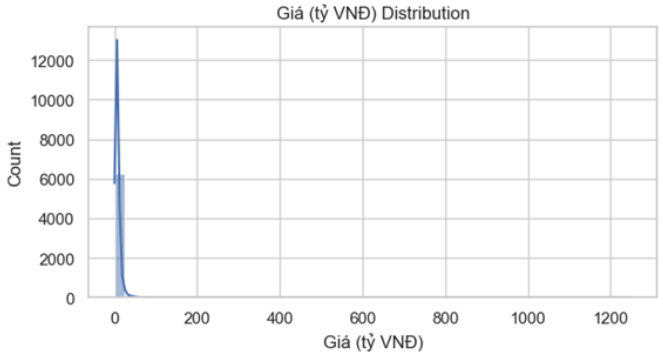

Fig. 1. Distribution of Price and Area showing long-tail behavior.

Similar distribution patterns were observed for price, width, length, and other numeric features. These distributions present dual challenges: (1) numerous extreme values that can disproportionately influence regression models, and (2) inherent skewness that violates normality assumptions. Consequently, a multi-stage outlier treatment strategy was designed.

C. Data Cleaning and Imputation

1) Handling Missing Values and Duplicates: Rows with entirely missing values were first removed. To balance data quality and sample retention, entries with more than six missing values (out of 13 attributes) were excluded, reducing the dataset to 6,396 samples. Subsequently, exact duplicate records were identified and deduplicated, preserving only the first occurrence.

2) Numerical Standardization: The price column, originally in mixed textual formats (e.g., "3,5 ty", "750 trieu"), was standardized into numerical values (in billion VND). Other numerical attributes were converted to numeric types, with commas replaced by periods for consistency.

3) Outlier Treatment: A two-stage outlier removal strategy was implemented:

- **Domain-based filtering:** Applied reasonable thresholds informed by real-estate market knowledge:
  - Price range: $0.2 - 200$ billion VND
  - Area: $10 - 1,500$ m^2
  - Width: $2 - 100$ m; Length: $3 - 150$ m
  - Room/Floors: Reasonable limits (0-25)
- **Statistical filtering:** The Interquartile Range (IQR) method with a multiplier of 3.0 was applied to the price and area columns to remove extreme statistical outliers.

These steps yielded a cleaned dataset of **5,971 samples**.

4) Advanced Missing-Value Imputation: Remaining missing values were imputed using context-aware strategies:

- **Numerical attributes:** Width and length were estimated from area and the median aspect ratio. Bedroom and bathroom counts were imputed based on area and group medians. Floor count used hierarchical median imputation.
- **Categorical attributes:** Common columns were filled with the mode. Columns with high missing rates (direction, furnishing status) were assigned a new category "*Khong xac dinh*" (Unknown).

D. Feature Engineering

To enhance predictive signals, three new features were constructed:

- **Tong_phong:** Total number of rooms (bedrooms + bathrooms), representing the functional capacity.
- **Aspect_ratio:** Width-to-length ratio, capturing the shape characteristics.
- **Dien_tich_per_phong:** Area per room, indicating spaciousness.

E. Categorical Encoding

Categorical variables were transformed using techniques appropriate to their cardinality:

- **Target Encoding with Smoothing:** Applied to *City* and *Ward/Commune* using Out-of-Fold (OOF) Target Encoding with 5-fold cross-validation.
- **One-Hot Encoding:** Applied to the *property type* variable.
- **Ordinal Encoding:** The *legal documentation* variable was mapped to an ordinal scale (e.g., "Da co so" $\rightarrow$ 4, "Khong co so" $\rightarrow$ 0).

The final dataset comprises **5,971 samples** and **18 features**.

V. Model Selection

A. Theoretical Framework for Model Selection

The selection of appropriate machine learning models is grounded in the intrinsic characteristics of the real estate dataset, as revealed through comprehensive exploratory data analysis (EDA). Empirical evidence derived from Fig. 2 and Fig. 3 highlights several critical properties that directly inform the modeling strategy.

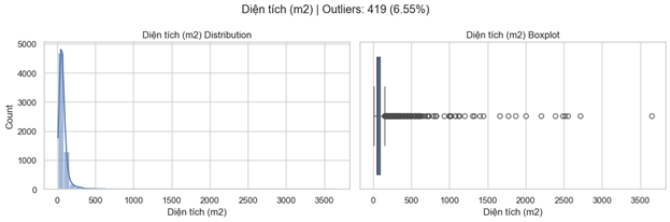

Fig. 2. Pronounced long-tail distribution of the Area variable.

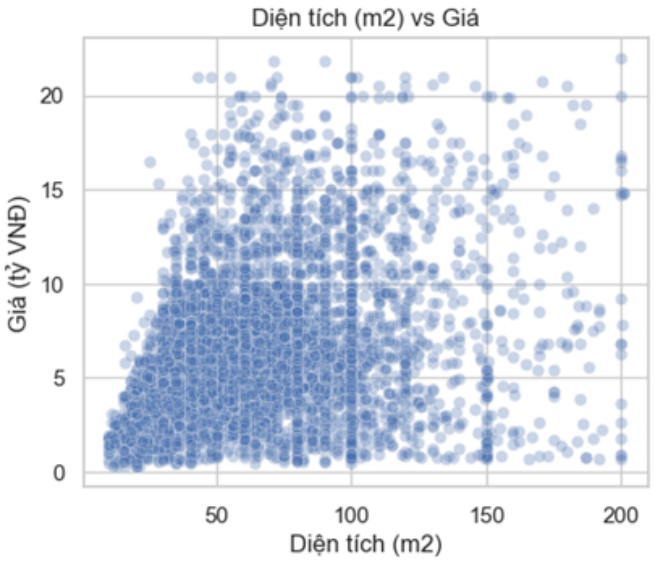

Fig. 3. Non-linear, heteroscedastic relationship between Area and Price.

Fig. 2 reveals a pronounced long-tail distribution of the Area variable, with 419 outliers (6.55%) identified using conventional statistical criteria. Fig. 3 demonstrates a non-linear, heteroscedastic relationship between area and price, characterized by a fan-shaped scatter pattern in which price variability increases with area. Similar patterns are consistently observed across other feature–target relationships. Collectively, these observations establish the following key data characteristics:

- **Non-Gaussian Target Distribution:** The target variable violates the assumptions of normality and homoscedasticity required by linear regression models.
- **Complex Non-linear and Conditional Relationships:** Feature–target interactions exhibit threshold effects, interaction terms, and segment-specific behaviors that are better represented by decision rules than by continuous parametric functions.
- **Structurally Significant Outliers:** Extreme values correspond to legitimate market phenomena (e.g., luxury properties or large land parcels) rather than measurement errors, necessitating preservation rather than removal.
- **High-Dimensional Categorical Feature Space:** The dataset includes numerous categorical, ordinal, and one-hot encoded variables with complex interactions.
- **Non-Random Missingness:** High missing rates in features such as property direction (74.11%) and furnishing status (47.80%) reflect systematic reporting behavior rather than random data loss.
- **Multicollinearity Among Predictors:** Engineered variables (e.g., area-per-room, aspect ratio) exhibit substantial inter-correlations that challenge parametric modeling assumptions.
- **Rule-Based Pricing Structure:** Real estate valuation inherently follows conditional logic (e.g., location $\times$ property type $\times$ size), favoring hierarchical decision structures over smooth functional mappings.

B. Rationale for Tree-Based Ensemble Selection

Given the above characteristics, traditional linear models and shallow neural networks are ill-suited for this task. Instead, tree-based ensemble methods—specifically Random Forest, LightGBM, and CatBoost—are selected due to their strong theoretical alignment with the observed data properties.

- **Robustness to Distributional Assumptions:** Tree-based ensembles impose no assumptions regarding feature distributions, variance homogeneity, or error normality. They naturally accommodate long-tail target distributions and heteroscedasticity through piecewise constant approximations rather than global parametric forms.
- **Native Modeling of Non-linearities and Feature Interactions:** Decision tree hierarchies automatically capture:
  - Non-linear threshold effects (e.g., price jumps at specific area ranges).
  - High-order feature interactions (e.g., location $\times$ property type $\times$ size).
  - Segment-dependent decision logic across different market regimes.
- **Integrated Missing Value Handling:** All selected algorithms incorporate built-in mechanisms for missing value processing during both training and inference:
  - *Random Forest:* Utilizes surrogate splits to approximate missing feature behavior.
  - *LightGBM:* Assigns default split directions for missing values.
  - *CatBoost:* Employs ordered boosting and dedicated missing value handling strategies.

  This capability is particularly critical given the extent and non-random nature of missingness in the dataset.
- **Robustness to Multicollinearity:** Tree-based models evaluate predictors locally at each split, rendering them

inherently robust to correlated features. This property mitigates instability arising from highly correlated engineered variables.

- **Effective Handling of Categorical Variables:**
  - *CatBoost:* Provides native categorical feature processing without extensive preprocessing.
  - *LightGBM:* Efficiently handles large, encoded feature spaces via histogram-based learning.
  - *Random Forest:* Naturally accommodates categorical splits through impurity-based criteria.
- **Outlier Robustness via Ensemble Aggregation:** Aggregation across multiple trees reduces sensitivity to extreme observations, preserving informative outliers while preventing disproportionate influence from any single data point.

Based on these empirical observations, tree-based ensemble models emerge as the most theoretically appropriate model family for capturing the complex, non-linear, and rule-driven structure of real estate pricing.

VI. Model Improvement

To maximize predictive performance, we implemented several optimization techniques.

A. Analytical Framework for Feature Evaluation

Feature selection is conducted through a structured analytical framework that integrates correlation analysis, distributional inspection, and domain knowledge from real estate valuation. The objective is to retain features that provide meaningful predictive signals while removing redundant or weakly informative variables, thereby improving model robustness, interpretability, and generalization.

Exploratory data analysis—including correlation heatmaps, boxplots, and count–median scatter plots—serves as the primary basis for evaluating feature relevance. Decisions are guided not only by statistical association with the target variable but also by economic plausibility and observed market behavior.

1) Numerical Feature Analysis and Selection: **Correlation-Based Evaluation:** The correlation heatmap (Fig. 4) reveals clear differences in the strength of association between numerical features and property price.

Features retained due to strong or moderate correlation and clear real-world relevance include:

- *Dien tích (m^2):* Exhibits the strongest association with price, reflecting the fundamental role of land and floor area in property valuation.
- *Chieu ngang (m)* and *Chieu dai (m):* Capture land geometry and frontage characteristics, which are known to influence accessibility and usability.
- *So tang:* Reflects vertical capacity and usable space.
- *Tong_phong:* Acts as a proxy for functional capacity and property scale.

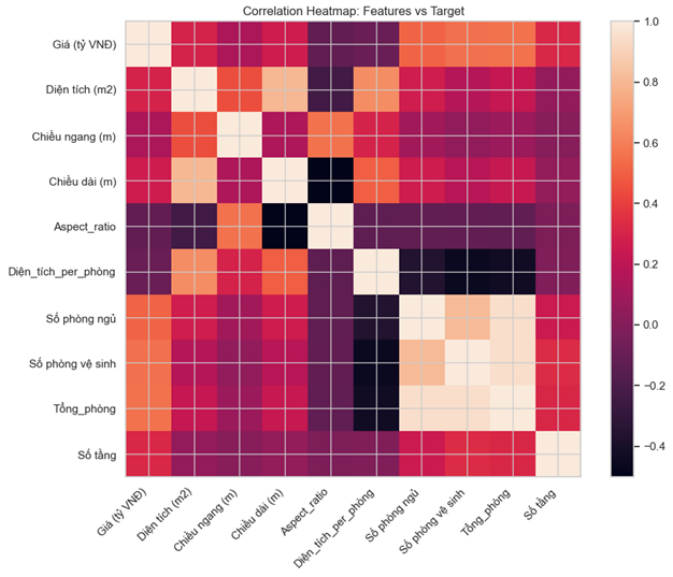

Fig. 4. Correlation heatmap between numerical features and property price.

Conversely, *Aspect_ratio* and *Dien_tich_per_phong* were excluded due to negligible correlation or redundant information, suggesting that absolute scale dominates pricing dynamics over ratio-based features.

Multicollinearity Assessment: A strong multicollinearity cluster was observed among room-related variables (*So phong ngu*, *So phong ve sinh*, *Tong_phong*). Pairwise correlations within this group are consistently high, indicating they represent overlapping aspects of property capacity. To address this, *Tong_phong* is retained as the most comprehensive representation, while the individual room counts are excluded to reduce dimensionality and prevent over-weighting a single underlying factor.

2) Categorical Feature Analysis and Selection: **Legal Documentation Effects:** Boxplot analysis (Fig. 5) reveals substantial stratification. Properties with formal legal documentation occupy higher price ranges, while incomplete documentation corresponds to lower, compressed distributions. This confirms legal status is a critical predictor reflecting market risk perception.

Property Type Segmentation: As shown in Fig. 6, price distributions demonstrate strong market segmentation. *Nha biet thu* (Villas) shows high median prices and variance, while *Nha ngo, hem* (Alley houses) exhibit lower medians. This confirms that property type encodes discrete pricing regimes.

Geographic Hierarchy Effects: Geographic analysis highlights non-linear location effects. While city-level data shows convergence around regional benchmarks, ward-level analysis (Fig. 7) reveals strong heterogeneity. Distinct clusters emerge: high-count/moderate-price established zones versus low-count/high-price premium areas. This justifies including both city and ward-level features.

3) Final Feature Set Composition: Based on the analytical evaluation, the final feature set selected for model training

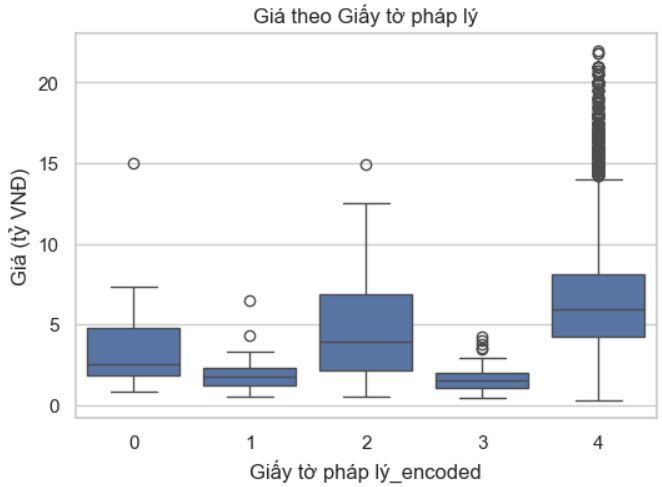

Fig. 5. Price distribution by legal documentation status.

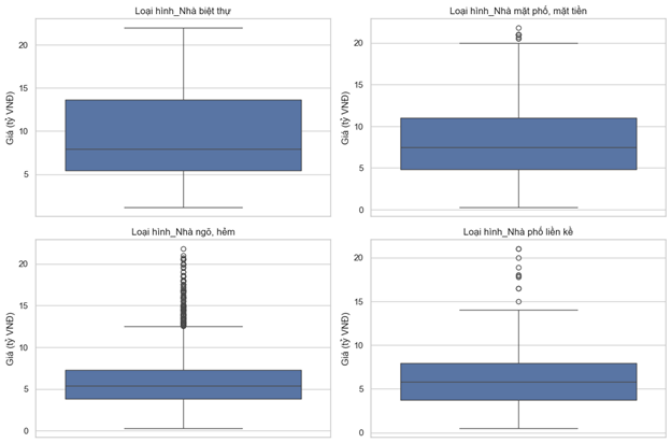

Fig. 6. Price distribution by property type categories.

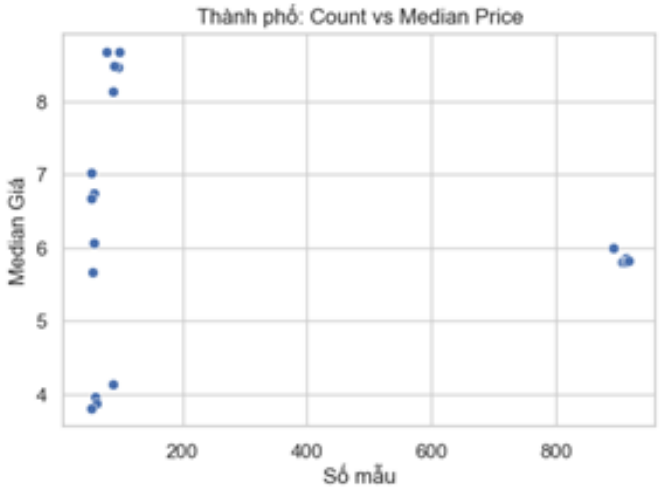

Fig. 7. Sample count versus median price at city and ward levels.

includes:

- **Core Numerical Features:** *Dien tích (m^2), Chieu ngang (m), Chieu dai (m), So tang, Tong_phong*.
- **Categorical Features:** *Giay to phap ly, Loai hinh nha, Thanh pho, Phuong/Xa*.

Features excluded (e.g., *So phong ngu*, *Aspect_ratio*) were removed due to redundancy or weak empirical association to ensure a parsimonious model.

B. Hyperparameter Tuning

Now, let's delve into the details of hyperparameter tuning. Unlike the exhaustive approach of GridSearchCV, we utilized Optuna, a next-generation automatic hyperparameter optimization software framework. In our case, we employed Optuna to perform an efficient search over the hyperparameter space for our regression models (Random Forest, LightGBM, and CatBoost). Here's how it works:

- We define an objective function that encapsulates the model training process. Inside this function, instead of a fixed grid, we define a dynamic search space for hyperparameters (e.g., learning rate, max depth, n estimators) using probabilistic sampling distributions provided by the trial object.
- The Optuna study systematically explores different combinations of these hyperparameters. We specifically utilized the Random Search sampler to explore the high-dimensional parameter space efficiently, employing 5-fold cross-validation within each trial to evaluate performance stability.
- The best combination of parameters, determined by the lowest validation error (RMSE), is extracted from the study using the study.best params attribute.
- Armed with these optimal parameters, we initialize the final model configuration.
- The model is then trained on the full training data and evaluated on both the training and test datasets to validate its accuracy and generalization capability.

By following this process, we aim to create a robust and high-performing model while significantly reducing the computational cost compared to traditional grid search methods.

C. Implementing CatBoostRegressor using CatBoost Library

CatBoost (Categorical Boosting) is a high-performance open-source library based on Gradient Boosting Decision Trees. In this project, we employed the `CatBoostRegressor` not only for its superior handling of categorical features but also for its robust internal strategies for dealing with missing data (*NaN*), which appeared frequently in our dataset under the label "Undefined".

1) Data Strategy and Mechanisms: Handling Missing Values (NaN) as Information: In features such as *Direction* and *Interior condition*, a significant portion of the data was labeled as "Unknown" or "Undefined". Instead of treating "Unknown" as a standard categorical label (which effectively imputes it as a distinct category), we converted these values to `NaN` (Not a Number).

This allows us to leverage CatBoost's internal **Min/Max Strategy** for missing values:

- During tree construction, for every split candidate, CatBoost tests two scenarios: treating `NaN` as the minimum value or assigning missing values to either side of the split in a way that maximizes the reduction in the loss function.
- It selects the scenario that yields the best improvement in the loss function.

This mechanism enables the model to capture latent semantic patterns associated with missing attributes, thereby improving predictive performance without requiring manual imputation.

Ordered Target Encoding: For a large range of feature values such as Location and Interior condition, manual One-Hot Encoding creates a sparse matrix. CatBoost addresses this issue using **Ordered Target Encoding**, which computes a statistic for each category based on the target values of previous rows in a random permutation, smoothed by a prior. This approach effectively extracts meaningful signals from location data while preventing target leakage.

2) Implementation Steps: To train and evaluate the model, we followed these structured steps:

- **Data Preparation:** We identified columns containing "Unknown" values (e.g., *Direction*, *Interior condition*) and replaced them with `NaN`. Location features (Location, Interior condition) were passed directly to the model as categorical feature indices.
- **Dataset Splitting:** The dataset was split into training and testing sets with an 80/20 ratio to ensure robust validation.
- **Hyperparameter Optimization with Optuna:** We utilized **Optuna** to search for optimal hyperparameters (Learning Rate, Depth, L2 Regularization, etc.) that maximize the model's predictive accuracy.
- **Training the Final Model:** Using the best hyperparameters, we initialized the `CatBoostRegressor`. The `nan_mode` parameter was set to 'Min' (default) to allow the model to optimally handle the missing values introduced.
- **Evaluation and Visualization:** The model was trained on the training set and validated on the test set. We monitored the learning process using both **RMSE** (loss function) and the R^2 **score** (evaluation metric).

Model Selection Justification. CatBoost was selected over traditional regression models and other tree-based algorithms due to its native support for categorical features, robust handling of missing values, and ability to model complex non-linear interactions. These properties are particularly well-suited for real estate datasets, which typically contain high-cardinality location attributes and incomplete legal information.

The following plot illustrates the learning curve of the model. The steady improvement of the validation R^2 score indicates that the model effectively learned both categorical patterns and information encoded in missing values.

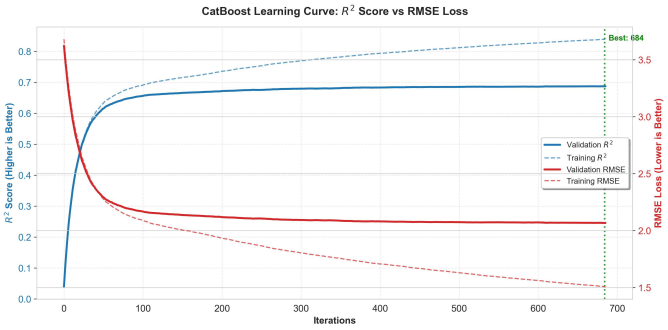

Fig. 8. Training and Validation R^2 Score Curve of the CatBoost Model.

The convergence between training and validation R^2 scores throughout the training process indicates stable learning behavior and suggests that the model does not suffer from significant overfitting.

Furthermore, feature importance analysis reveals that *Area* and location-related features are the primary drivers of price prediction, followed by attributes with missing values such as *Direction* and *Legal Status*. This observation indicates that the model successfully utilized information encoded through the `NaN` handling mechanism.

VII. Model Evaluation

We will evaluate the performance of our models using appropriate metrics such as Root Mean Squared Error (RMSE) and Coefficient of Determination (R^2). Taken together, these two metrics provide a comprehensive view of your model's performance.

Root Mean Squared Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data. It represents the square root of the average of squared differences between prediction and actual observation.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

RMSE is particularly useful when large errors are particularly undesirable, as it gives a relatively high weight to large errors. A lower RMSE value indicates a better fit.

R^2 ScoreThe R^2 score, also known as the coefficient of determination, is a statistical measure that shows the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

It provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model. An R^2 score of 1 indicates that the regression predictions perfectly fit the data.Comparison of ModelsWe compare the performance of our baseline Random Forest model against the gradient boosting techniques (LightGBM and CatBoost) optimized via Optuna. The results are summarized in Table III.

TABLE III
MODEL EVALUATION RESULTS

Model	Dataset	R^2 Score	RMSE
Random Forest	Training Set	0.6558	2.1786
	Testing Set	0.6515	2.1818
LightGBM	Training Set	0.6876	2.0992
	Testing Set	0.6841	2.0775
CatBoost	Training Set	0.6934	2.0785
	Testing Set	**0.6911**	**2.0542**

VIII. CONCLUSION

In this study, we successfully established an end-to-end real estate price prediction pipeline, ranging from automated data collection on `nhatot.com` using `crawl4ai` to the deployment of advanced machine learning models.

Based on the experimental comparison among Random Forest, LightGBM, and CatBoost, we identified **CatBoost Regressor** (optimized via Optuna's Random Search strategy) as the best-performing model. It effectively handled high-cardinality categorical features (e.g., location, ward) and achieved the best balance between accuracy and generalization:

- **Final Testing Accuracy (R^2 Score):** 0.6911
- **Final Root Mean Squared Error (RMSE):** 2.0542

The success of this approach confirms the efficiency of using Optuna to replace traditional GridSearch, significantly reducing computational costs while discovering superior hyperparameters.

Future work will focus on integrating multimodal learning, specifically employing Convolutional Neural Networks (CNNs) to extract features from house images, combining them with the current tabular features to further enhance prediction accuracy.

REFERENCES

[1] Breiman, L. "Random Forests". Machine Learning, 45(1), 5-32, 2001.
[2] Ke, G., et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". NIPS, 2017.
[3] Prokhorenkova, L., et al. "CatBoost: unbiased boosting with categorical features". NIPS, 2018.