Bengaluru Real Estate Price Prediction

A Machine Learning Approach to Predicting Residential Property Prices

13,320	7,251	0.831	3
Raw Records	After Cleaning	Best CV R^2	Models Compared

MGSC 661 Project Report

1. Introduction

The Bengaluru housing market is one of the most dynamic real estate markets in India, with property prices varying significantly based on neighborhood, property configuration, and available amenities. This project develops a predictive model for residential property prices (in Lakhs INR) using a publicly available dataset of 13,320 Bengaluru property listings.

The analysis follows a structured data science pipeline: raw data cleaning, feature engineering, multi-stage outlier removal, and model selection via cross-validated grid search. The goal is to identify the regression algorithm that best generalizes to unseen property data.

2. Dataset Description

The dataset (Bengaluru_House_Data.csv) contains 13,320 listings with 9 original features. The target variable is **price** in Lakhs INR (1 Lakh = 100,000 INR).

Feature	Description
area_type	Type of area (Super built-up, Built-up, Plot, Carpet)
availability	When the property is available
location	Neighborhood / locality in Bengaluru
size	Number of bedrooms (e.g., 2 BHK, 4 Bedroom)
society	Name of the housing society
total_sqft	Total area in square feet (may contain ranges)
bath	Number of bathrooms
balcony	Number of balconies
price	Price in Lakhs INR (target)

Initial inspection revealed 1 missing location value, 16 missing size values, and 73 missing bathroom values. These were dropped, yielding 13,246 clean records for further processing.

3. Methodology

3.1 Data Cleaning

• **Column removal:** Dropped *availability*, *society*, *balcony*, and *area_type* as they provided limited predictive signal or had excessive cardinality.

• **Missing value removal:** Dropped rows with any null values (13,320 to 13,246 records).

• **BHK extraction:** Parsed the *size* column (e.g., '2 BHK', '4 Bedroom') into a numeric *bhk* feature.

• **Square footage conversion:** The *total_sqft* column contained ranges (e.g., '1133 - 1384') and non-numeric entries. A custom parser averaged range endpoints and converted values to float, dropping unconvertible entries.

3.2 Feature Engineering

• **Price per square foot:** Computed *price_per_sqft = price * 100,000 / total_sqft* for outlier analysis.

• **Location consolidation:** Locations with 10 or fewer listings were grouped into an 'other' category to reduce the high cardinality of location features.

• **One-hot encoding:** Location was one-hot encoded (dropping the 'other' category as baseline) for model input.

3.3 Outlier Removal (Multi-Stage)

A four-stage outlier removal process was applied, progressively filtering the data from 13,246 down to 7,251 records:

Stage	Rule	Records After
1. Min sqft per BHK	Remove if total_sqft / bhk < 300	12,502
2. Price per sqft	Remove beyond +/- 1 SD per location	10,241
3. Cross-BHK anomaly	Remove if higher-BHK price < lower-BHK mean	~7,500
4. Bathroom constraint	Remove if bath > bhk + 2	7,251

The price per sqft removal was performed at the location level, computing the mean and standard deviation of price_per_sqft for each neighborhood and retaining only those listings within one standard deviation. The cross-BHK anomaly detection ensured that, within the same locality, a property with more bedrooms was not cheaper per square foot than the average of properties with fewer bedrooms.

3.4 Model Selection

Three regression algorithms were compared using **GridSearchCV** with 5-fold **ShuffleSplit** cross-validation (test size = 33%):

• **Linear Regression** — hyperparameter: fit_intercept [True, False]

• **Decision Tree Regressor** — hyperparameter: max_depth [1, 5, 10]

- **Lasso Regression** — hyperparameters: alpha [1, 2], selection [random, cyclic]

4. Results

4.1 Model Comparison

Model	Best CV R^2	Best Parameters
Linear Regression	0.831	fit_intercept: False
Decision Tree	0.743	max_depth: 10
Lasso	0.704	alpha: 1, selection: random

Linear Regression achieved the highest cross-validated R^2 score of **0.831**, outperforming Decision Tree Regressor by approximately 9 percentage points and Lasso Regression by approximately 13 percentage points.

4.2 Linear Regression — Detailed Evaluation

The selected Linear Regression model was further evaluated on a held-out test set (33% split, random_state=42):

Metric	Value
Test Set R^2	0.808
CV Fold 1	0.824
CV Fold 2	0.797
CV Fold 3	0.862
CV Fold 4	0.824
CV Fold 5	0.846
CV Mean R^2	0.831

The cross-validation scores ranged from 0.797 to 0.862, indicating consistent generalization with relatively low variance across folds. The test set R^2 of 0.808 is closely aligned with the cross-validated estimate, confirming that the model does not overfit.

5. Key Findings

• **Data quality was a major challenge:** The raw dataset contained mixed-format square footage entries, unrealistic outliers (e.g., 43 bedrooms in a small property), and high-cardinality location features. Careful multi-stage cleaning was essential.

• **Outlier removal significantly improved model quality:** Removing nearly 45% of records through principled domain-driven rules (minimum sqft per BHK, per-location price distributions, cross-BHK consistency, and bathroom constraints) produced a cleaner dataset that enabled better model performance.

• **Linear Regression outperformed tree-based and regularized models:** After proper feature engineering and outlier removal, the linear relationship between features and price was strong enough that simple Linear Regression achieved the best R^2. Decision Trees likely overfit or underfit at the tested depths, while Lasso's L1 penalty was too aggressive given the one-hot encoded location features.

• **Location is a critical predictor:** One-hot encoding of Bengaluru localities, even after consolidating rare neighborhoods, contributed substantially to model accuracy.

---

6. Conclusion

This project demonstrates that a well-executed data cleaning and feature engineering pipeline can enable even simple models to achieve strong predictive performance. The final Linear Regression model explains approximately 83% of the variance in Bengaluru property prices using just four core features: location, total square footage, number of bedrooms, and number of bathrooms.

Future work could explore ensemble methods (e.g., Random Forest, Gradient Boosting) with broader hyperparameter search spaces, incorporate additional features such as proximity to amenities or transit, and apply the model to a deployment-ready prediction service.

End of Report