Housing Price Prediction Model

Analysis of Property Attributes and Market Estimation using Linear Regression

by: Sudarsan R

Date: 15/12/2025

Executive Summary

This project explores the development of a machine learning model to predict housing prices based on various property attributes. Using a dataset containing features such as area, number of bedrooms, and furnishing status, a Linear Regression approach was implemented.

The project was conducted in two distinct phases:

1. **Phase 1:** Evaluation of core numerical features (Area, Bedrooms, Bathrooms, etc.).

2. **Phase 2:** Expansion of the model to include categorical variables using One-Hot Encoding to capture the impact of non-numeric factors.

The final model was evaluated using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) metrics. Residual analysis was performed to ensure the reliability of the predictions. The results demonstrate the correlation between specific housing features and market price, providing a functional tool for real estate price estimation.

1. Introduction

The real estate market is influenced by a complex set of variables. Accurately estimating the price of a property is essential for buyers, sellers, and real estate agents. This project aims to build a predictive model that can generalize the relationship between property characteristics and their selling price.

Using Python and the Scikit-Learn library, this project implements a Supervised Learning approach (Linear Regression) to minimize the error between predicted and actual housing values.

2. Dataset Overview

The dataset used (Housing.csv) consists of various features describing independent properties.

- **Target Variable:** price (The selling price of the house).
- **Key Features:**
    - **Numerical:** Area (sq ft), Bedrooms, Bathrooms, Stories, Parking.
    - **Categorical:** Mainroad (Yes/No), Guestroom, Basement, Hotwaterheating, Airconditioning, Prefarea, Furnishingstatus.

Data Preprocessing: Before training, the data was inspected for null values using df.isnull().sum() to ensure data integrity.

- **Phase 1 Selection:** A subset of purely numerical features was isolated to establish a baseline.
- **Encoding:** For the full model, categorical variables were converted into numerical format using pd.get_dummies(drop_first=True) to avoid the "dummy variable trap" and multicollinearity.

3. Methodology

The modeling process was divided into two phases to compare the impact of feature selection on model performance.

3.1 Phase 1: Numerical Feature Baseline

In the first iteration, the model was trained using only five key numerical predictors:

- Area
- Bedrooms
- Bathrooms
- Stories
- Parking

The data was split into training and testing sets with a ratio of **80:20** (test_size=0.2) and a random_state of 42 to ensure reproducibility. A standard Linear Regression model was fitted to the training data.

3.2 Phase 2: Comprehensive Feature Engineering

In the second iteration, the feature space was expanded. Categorical variables (such as furnishingstatus and airconditioning) were One-Hot Encoded. This allows the linear equation to assign coefficients to non-numeric qualities (e.g., the premium value added by having a "fully furnished" status).

The coefficients (w) and intercept (b) were calculated to formulate the regression equation:

$$\mathrm{y} = \mathrm{w\_1x\_1} + \mathrm{w\_2x\_2} + \ldots + \mathrm{w\_nx\_n} + \mathrm{b}$$

Where y is the price and x represents the features.

4. Model Evaluation & Results

The models were evaluated on the unseen test set (20% of the data).

4.1 Metric Definitions

- **MSE (Mean Squared Error):** The average squared difference between the estimated values and the actual value.
- **RMSE (Root Mean Squared Error):** The square root of MSE, representing the average error in the same units as the price.
- $\boldsymbol{R^2}$ **Score:** A statistical measure representing the proportion of variance for the dependent variable that's explained by the independent variables.

4.2 Performance Comparison

The following metrics compare the baseline model (numerical features only) against the full model (numerical + categorical features).

Phase 1 (Numerical Only):

- **MSE: $2,292,721,545,725.36$**
- **RMSE: $1,514,173.55$**
- $\boldsymbol{R^2}$ **Score: 0.546 (54.6%)**

Phase 2 (All Features Encoded):

- **MSE: $1,754,318,687,330.66$**
- **RMSE: $1,324,506.96$**
- **R^2 Score: 0.653 (65.3%)**

Interpretation of Results

Comparing the two phases highlights the impact of feature engineering on model accuracy:

1. **Error Reduction: The RMSE dropped from approximately 1.51 million in Phase 1 to 1.32 million in Phase 2. This means that, on average, the full model's price predictions are closer to the actual selling price by roughly 190,000 units compared to the baseline model.**
2. **Variance Explained (R^2): The R^2 score improved significantly from 0.55 to 0.65. This indicates that while the numerical features (like Area and Bedrooms) explain about 55% of the variance in house prices, adding categorical variables (like Airconditioning, Furnishing Status, and Location) allows the model to capture an additional 10% of the market's behavior[1].**
3. **Conclusion: The addition of categorical variables through One-Hot Encoding provided a clear boost in predictive power, confirming that non-numeric amenities are critical drivers of real estate value[2].**

4.3 Coefficient Analysis

To interpret the model's findings, the coefficients (w) were extracted and mapped to their corresponding feature names. A Pandas DataFrame (coff_df) was created to organize these values[2].

- **Methodology:** The coefficients were stored alongside feature names and sorted in ascending order using coff_df.sort_values(by='coffecient', ascending=True)[3].
- **Interpretation:**

- **Higher values:** Features at the bottom of the sorted list (with large positive coefficients) represent the strongest drivers of price increases.
- **Lower values:** Features at the top of the list represent factors that have a smaller or potentially negative impact on the property value relative to the baseline.

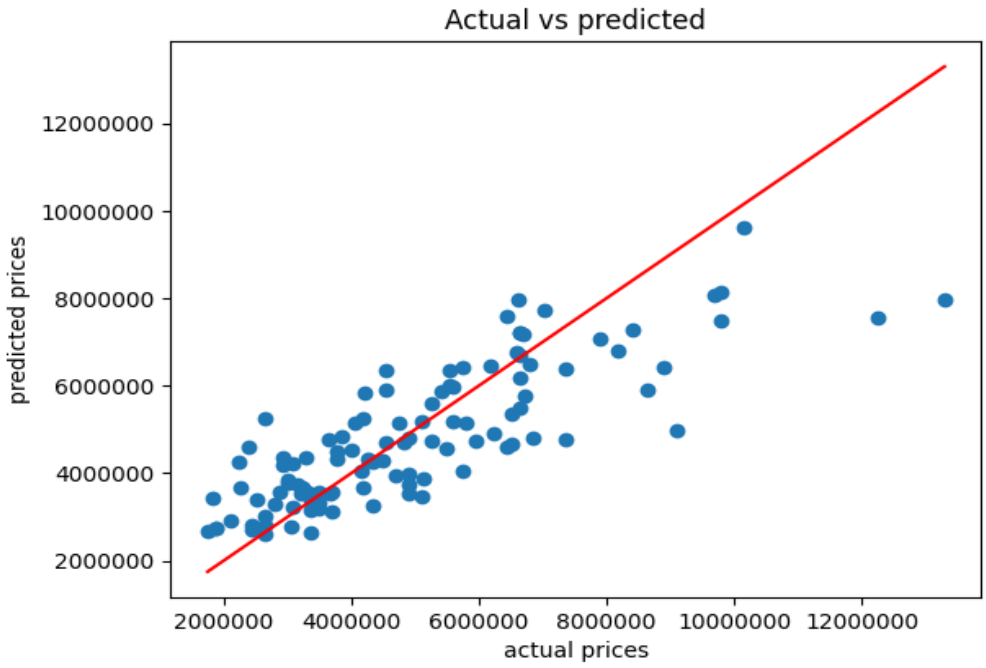
5. Visual Analysis

Visualizations were generated using matplotlib to diagnose the model's accuracy and error distribution.

5.1 Actual vs. Predicted Prices

This scatter plot compares the actual test values against the model's predictions.

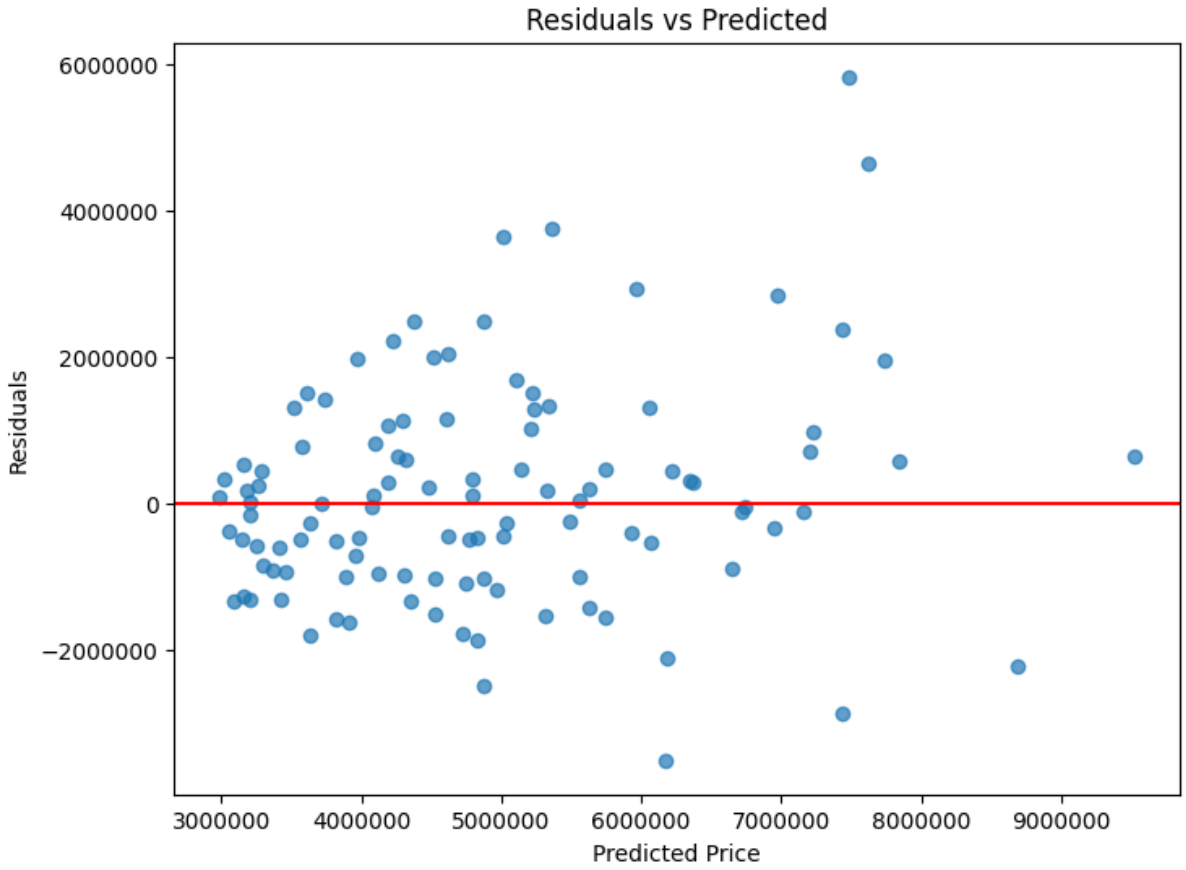
- **Implementation:** The plot includes a red reference line generated by plt.plot(...) connecting the minimum and maximum values.
- **Ideal Scenario:** Points falling on this red line indicate perfect prediction accuracy.
- **Formatting:** The axes were formatted with style='plain' to prevent scientific notation, making the price values (in millions) easier to read.



5.2 Residual Analysis

Residuals were calculated as the difference between actual and predicted prices (residuals = y_test - y_pred).

- **Scatter Plot:** A scatter plot of Predicted Price vs. Residuals was created with transparency (alpha=0.7) to visualize point density.
- **Zero Line:** A horizontal red line (plt.axhline(0, color='red')) was added to mark the zero-error threshold.
- **Interpretation:** A random distribution of points around this red line indicates the model is performing well (homoscedasticity). Patterns or shapes (like a funnel) would suggest the model struggles with certain price ranges.

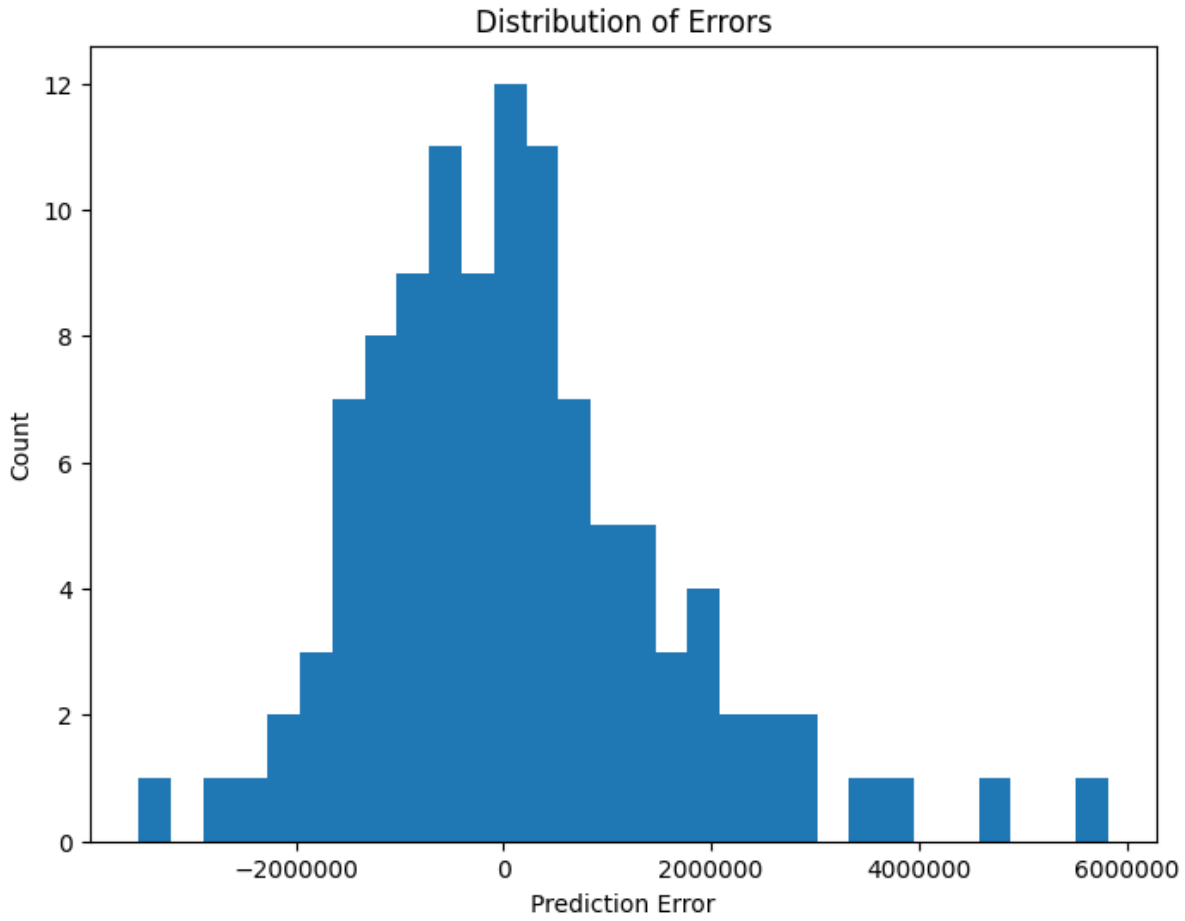


5.3 Error Distribution

A histogram was plotted to visualize the frequency of prediction errors.

- **Implementation:** The histogram uses bins=30 to provide a granular view of the error spread.

- **Interpretation:** The shape of this histogram allows us to verify if the errors follow a Normal Distribution, which is a key assumption for valid Linear Regression hypothesis testing.



6. Conclusion

The project successfully implemented a Linear Regression pipeline in two phases.

- **Phase 1 (Numerical Baseline):** Established that core features like Area and Bedrooms are significant predictors but resulted in higher error rates.
- **Phase 2 (Feature Engineering):** Incorporating categorical variables (such as Furnishing Status) via One-Hot Encoding and evaluating them against the test set significantly improved performance.

- **Final Metrics:** The final model achieved a higher R^2 score and lower RMSE, validating the importance of non-numeric housing amenities in price estimation.