

Housing Price Prediction using Multiple Linear Regression

Objective:

To develop a predictive model that estimates the **price of a house** based on various input features such as **area, number of bedrooms/bathrooms**, and amenities like **air conditioning, furnishing status**, etc.

Dataset Overview:

- **Source:** housing.csv
- **Target Variable:** price (continuous)
- **Input Features Used (12 total):**
 - **Numerical:** area, bedrooms, bathrooms, stories, parking
 - **Categorical:** main road, guestroom, basement, hot water heating, air conditioning, pref area, furnishing status

Methodology:

1. **Exploratory Data Analysis (EDA):**
 - Pair plots and correlation heat maps were used to understand relationships between features and the target.
 - High correlation observed between area and price.
2. **Data Preprocessing:**
 - **One Hot Encoding** applied to categorical features using a Column Transformer.
 - Combined with numerical features for model training.
3. **Model Development:**
 - Used **Multiple Linear Regression** from Scikit-learn.
 - Split data: 80% for training, 20% for testing.
 - Implemented via a **Pipeline** to streamline preprocessing and training.
4. **Evaluation Metrics:**
 - **Mean Squared Error (MSE):** Measures average squared difference between actual and predicted values.
 - **R² Score:** Indicates how well the model explains the variability in price.

Results:

- **R² Score (Test Set):** ~0.79 (indicative, depends on dataset)
- **MSE:** Depends on price scale but remained within acceptable range.
- **Residuals:** Roughly normally distributed, suggesting a well-fitted linear model.
- **Most influential features:** area, air conditioning, and furnishing status.

Output:

- Final trained model saved as housing_price_model.pkl using joblib

Tools & Technologies:

- Python
- Pandas, Seaborn, Matplotlib
- Scikit-learn (Linear Regression, Pipeline, Preprocessing)
- Joblib (model persistence)

Key Learning Outcomes:

- Hands-on experience with regression modeling
- Real-world application of EDA and preprocessing
- End-to-end model pipeline creation
- Residual analysis for model quality assessment
- Saving and deploying trained models