PROJECT REPORT

PROJECT 2: House Price Prediction Using Linear Regression

K D Liyanage

1. Introduction

Predicting housing prices is one of the most common and practical applications of machine learning. This project uses the Boston Housing Dataset to build a Linear Regression model that predicts the median value of owner-occupied homes based on various socioeconomic and structural features.

The main objectives of the project are:

- To explore and preprocess the dataset.
- To analyze feature relationships.
- To build and train a Linear Regression model.
- To evaluate its performance using standard regression metrics.

2. Dataset Description

The Boston Housing dataset contains several numerical and categorical attributes that influence home prices. Key columns used in this project include:

- **crim** – Per capita crime rate by town
- **zn** – Proportion of large residential lots
- **indus** – Proportion of non-retail business acres
- **chas** – Charles River dummy variable (1 = near river)
- **nox** – Nitrogen oxide concentration
- **rm** – Average number of rooms per dwelling
- **age** – Proportion of units built before 1940
- **dis** – Distance to employment centers
- **rad** – Highway accessibility index
- **tax** – Property tax rate per $10,000
- **ptratio** – Student-teacher ratio
- **lstat** – % lower-status population
- **medv** – *Target variable* (Median home value)

These features capture economic, environmental, and structural factors affecting housing prices.

3. Data Preprocessing

To prepare the dataset for modeling, the following steps were performed:

3.1 Loading the dataset

The dataset was loaded using pandas into a DataFrame for analysis.

3.2 Checking dataset structure

- Used .info() and .describe() to inspect the shape, data types, and descriptive statistics.
- Verified that there were no missing values.

3.3 Handling Missing Values

- Checked for null values.
- Imputed or removed rows/columns where necessary.

3.4 Encoding Categorical Features

- The chas variable was already binary.
- Other datasets may require LabelEncoder or OneHotEncoding.

3.5 Feature selection

- The target variable is **medv**.

  All other columns were treated as predictors.

3.6 Train-Test Split

- Data was split into:

  80% Training

  20% Testing
- Ensures fair evaluation of model performance on unseen data.

3.7 Feature Scaling

- Applied **StandardScaler** to normalize numerical features.
- Scaling helps improve model training stability.

4. Exploratory Data Analysis (EDA)

The following key visualizations were generated to understand relationships and patterns:

4.1 Distribution of numerical features

Most features were visualized to understand:

- Skewness
- Outliers
- Spread and central tendency

4.2 Correlation heatmap

A heatmap was used to observe linear relationships.

Key observations:

- **rm** had a strong positive correlation with **medv**.
- **lstat** had a strong negative correlation with **medv**.
- Strong multicollinearity existed among some features (e.g., **tax**, **rad**).

These insights guided the expectations for the regression model.

5. Model Building

5.1 Linear Regression Model

- Used LinearRegression from scikit-learn.
- Trained the model on the scaled training dataset.

5.2 Predictions

- Obtained predicted house prices for the test dataset.
- Stored results for performance evaluation and visualization.

6. Model Evaluation

The model was evaluated using standard regression metrics:

- **Mean Absolute Error (MAE)**
- **Mean Squared Error (MSE)**

  Measures average squared prediction error.
- **R^2 Score**

  Measures the proportion of variance explained by the model.

A **Predicted vs Actual Price Plot** was generated, showing:

- How closely the predictions align with the true prices
- The overall performance and error distribution

The model performed reasonably well and captured major trends in pricing.

7. Optimization (Advanced)

To enhance model performance, additional regression techniques were tested:

7.1 Polynomial Regression

- Captures nonlinear relationships
- Slightly improved accuracy depending on degree

7.2 Ridge Regression

- Adds L2 regularization to reduce overfitting

7.3 Lasso Regression

- Adds L1 regularization and performs feature selection

Performance metrics were compared to identify which model provided the best balance of accuracy and generalization.

8. Conclusion

This project provided a full experience in developing a machine learning regression model for predicting house prices. The Linear Regression model demonstrated solid performance, and optimization techniques highlighted how model tuning can improve results. Key skills developed include data preprocessing, visualization, regression modeling, and interpretation of evaluation metrics.

9. Key Takeaways

- Learned the complete workflow of a supervised ML project
- Understood how to clean and preprocess real-world data
- Improved skills in visualization and feature analysis
- Built and evaluated a regression model using scikit-learn
- Explored advanced techniques such as Ridge, Lasso, and Polynomial Regression
- Strengthened understanding of how numerical features influence house prices