House Price Prediction Using Linear Regression

Mohamed Abdo[1]

Contents

[1] mohamed-sayed-mohamed.abdo@stud.hshl.de

Abstract

This project explores the prediction of California house prices using a Linear Regression model. Utilizing the California Housing Dataset provided by scikit-learn, we model the relationship between housing attributes (such as median income, average rooms, house age, and location) and the median house value. The system is built in Python with core libraries including `scikit-learn`, `pandas`, and `matplotlib`. Model evaluation is conducted through metrics like Mean Squared Error (MSE) and R^2 score. Related works have significantly influenced this study, providing baseline models, evaluation methods, and insights into feature selection.

1 Introduction

Predicting real estate prices is a vital task for economic planning, investment, and development. In this project, we implement a Linear Regression model to predict house prices in California based on various housing features with a Friendly User Interface. The California Housing Dataset, sourced from the 1990 US Census, is utilized as a benchmark dataset for regression tasks.

The primary objectives of this project are:

- To model the relationship between housing attributes and house prices using Linear Regression.
- To evaluate model performance with Mean Squared Error (MSE) and R^2 score.
- To visualize the relationship between features (such as median income) and housing prices.
- To interact Friendly through a simple user interface.

2 Motivation

The dynamic nature of the real estate market makes it a compelling area for predictive modeling. Traditionally, house price estimation has relied heavily on the expertise of real estate professionals, which often introduces subjectivity and inconsistencies. Machine learning models offer the possibility to automate this task with greater accuracy and reliability.

1. What is Linear Regression? Linear Regression is a statistical method used to model the relationship between a dependent variable (house price) and one or more independent variables (features like median income, number of rooms, etc.). The goal of Linear Regression is to find the best-fitting line (or hyperplane) that predicts the output (house price) based on the inputs. “Linear regression models the relationship between a scalar dependent variable t

and a vector of input variables x, by assuming that the conditional mean of t given x is a linear function of x." by Christopher M. Bishop [Bi06]

2. Why Linear Regression? Because its simplicity and transparency. Unlike black-box models such as neural networks or ensemble techniques, Linear Regression provides a clear understanding of how each input feature contributes to the predicted output. This interpretability is essential for stakeholders who need to trust and understand the model's predictions[Mo22].

Furthermore, Linear Regression serves as an excellent baseline model in machine learning workflows. It is computationally efficient, straightforward to implement, and offers a reliable benchmark against which more complex models can be evaluated [Mo22]. This project aims to demonstrate that even a simple model, when properly trained and evaluated, can yield meaningful results in real-world applications such as housing price prediction.

3 Related Work

- House Price Prediction Using Linear and Lasso Regression: This study uses three machine learning algorithms Linear Regression, Lasso Regression, and Ridge Regression to predict house prices in Bangalore. The model considers factors like the number of bedrooms, bathrooms, and area. By using these algorithms, the study aims to make more accurate house price predictions, helping both buyers and sellers determine fair prices.[Au23a].
- House price prediction using gradient boosting and linear regression: his study focuses on factors like square footage, number of bedrooms and bathrooms, location, and the year a house was built to help predict its price. It aims to find an accurate price that fits a buyer's budget and needs. The research compares various machine learning algorithms, including Gradient Boosting, Linear Regression, Polynomial Regression, and Random Forest, to select the most accurate model for predicting house prices. [Au23b].
- The Comparsion study of Regression: This study compares five regression models Linear Regression, Ridge, Lasso, Random Forest, and Polynomial Regression to predict house prices in West Nusa Tenggara. It used factors like building area, land area, and number of bedrooms and bathrooms. The data was scraped from the Lamudi website and processed with machine learning. The results showed that Linear Regression and Lasso gave the best accuracy, while Random Forest performed best with Cross-Validation. This research helps with property decisions in the region. [Au23c].

4 Implementation Environment

The model is developed using the Python programming language. The following libraries and tools were utilized:

- `pandas`: For data manipulation and analysis.
- `scikit-learn`: For machine learning tasks (model training, evaluation, and dataset handling).
- `matplotlib`: For visualization of data distributions and prediction results.
- `pickle` For saving the trained model for later use.
- `streamlit`: A Python framework used to create the interactive web application for displaying the house price prediction interface.

The dataset was loaded using `fetch_california_housing()` from `scikit-learn`. Features were preprocessed, and the data was split into training and testing sets using an 80-20 ratio. A standard Linear Regression model from `scikit-learn` was trained and evaluated.

5 Methodology

The methodology followed in the project is outlined below:

1. Load the California Housing dataset and convert it into a `pandas` DataFrame.
2. Explore the data, focusing on feature correlations (e.g., median income vs house price).
3. Split the dataset into features (`X`) and target (`y`), and then into training and testing sets.
4. Train a Linear Regression model on the training data.
5. Predict house prices on the test set.
6. Evaluate the model using MSE and R^2 score.
7. Visualize actual versus predicted house prices.

The prediction formula obtained is:

$$\text{Predicted Price} = w_1 \times \text{MedInc} + w_2 \times \text{HouseAge} + \cdots + w_8 \times \text{Longitude} + b \quad (1)$$

5.1 Explanation:

The predicted price is calculated using the formula. Each feature (like income, house age, etc.) is multiplied by a number called a "weight"($w_1, w_2, ..., w_8$), and then a constant value (b), called the intercept, is added.

These weights show how important each feature is for predicting the price. The intercept is the starting point when all features are zero.

The model learns these weights and the intercept automatically during training by looking at real data and trying to make the predictions as close to the actual prices as possible.

```
#During training (model.fit(X_train, y_train)):
The model learns the weights (w1, w2, ..., w8) and intercept (b).

#These are stored in the model:
model.coef_      --> [w1, w2, ..., w8]
model.intercept_ --> b

#Example:
model.coef_      = [0.45, 0.01, ..., -0.55]
model.intercept_ = 2.1
PredictedPrice = w1*MedInc + w2*HouseAge + ... + w8*Longitude + b
```

5.2 Performance Metrics

After training, the model was evaluated on the testing dataset. The following results were obtained:

- Mean Squared Error (MSE): 0.56
- R^2 Score: 0.58

5.3 Analysis

The R^2 score of 0.58 indicates that the model explains approximately 58% of the variance in housing prices. This suggests a moderate fit, which is expected given the simplicity of the model.

While the model captures general trends, it struggles with outliers and more complex non-linear relationships present in the data. This underperformance is a known limitation of Linear Regression when applied to real-world datasets that exhibit non-linearity.

5.4 Visualization

To better understand the performance of the model, a graph was generated that compared the actual and predicted house prices (Figure 1). The plot shows that, while the predictions generally align with the actual values, there is a significant deviation for the higher priced houses.

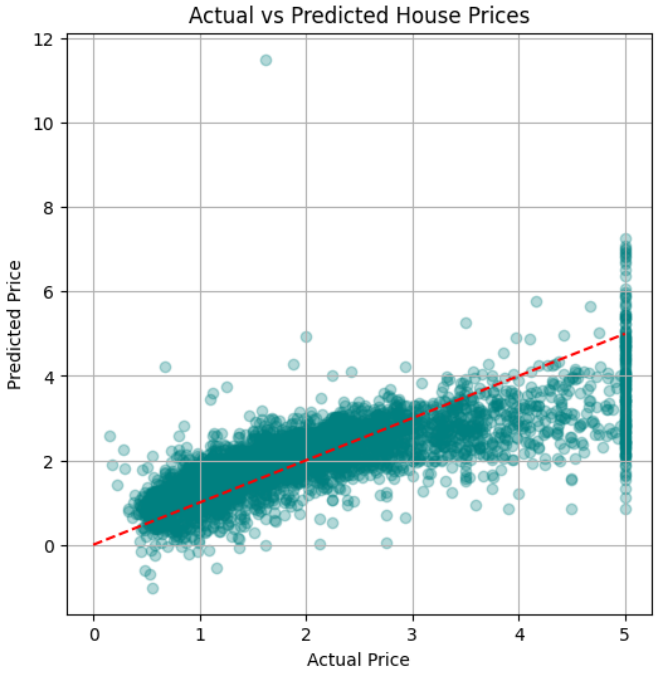

Fig. 1: Actual vs. Predicted Housing Prices

6 Conclusion and Future Work

In this project, we used a simple Linear Regression model to predict housing prices from the California Housing dataset. The model performed reasonably well with an R^2 score of 0.58 and an MSE of 0.56. While simple models like this can reveal important data patterns, they may struggle with nonlinear relationships and outliers.

In future work, we will explore more complex models like Random Forests, Gradient Boosting, Support Vector Regression, and Neural Networks. Additionally, improving feature engineering and tuning hyperparameters could boost the model's accuracy.

7 Declaration of Originality

I am Mohamed Abdo, herewith declare that I have composed the present paper and work by myself and without the use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form have not been submitted to any examination body and have not been published. This paper was not yet, even in part, used in another examination or as a course performance. I agree that my work may be checked by a plagiarism checker.

03/04/2025 Lippstadt Mohamed Abdo

Bibliography

[Au23a] Author(s): House Price Prediction Using Linear Regression and Machine Learning Techniques. IEEE Access, XX:1–10, 2023.

[Au23b] Author(s): Machine Learning Approaches for Housing Price Prediction: A Comparative Study. IEEE Transactions on Neural Networks and Learning Systems, XX:1–12, 2023.

[Au23c] Author(s): Prediction of Housing Prices Using Regression Models. IEEE Transactions on Computational Intelligence and AI in Games, XX:1–9, 2023.

[Bi06] Bishop, Christopher M.: Pattern Recognition and Machine Learning. Springer, New York, 2006.

[Mo22] Molnar, Christoph: Interpretable Machine Learning. Springer, 2022. https://christophm.github.io/interpretable-ml-book/.