Comparing Machine Learning Models for House Price Prediction: Linear Regression, Decision Tree, and Random Forest

Yuhan Jing*

College of Engineering, Texas A&M University, College Station, TX 77840, USA

Abstract. Predicting housing prices accurately is crucial for informed real estate decisions. This paper compares the performance of three machine learning models—Linear Regression, Decision Tree, and Random Forest—on the California Housing Prices dataset from scikit-learn. Through comprehensive data preprocessing and hyperparameter tuning using grid search with cross-validation, the paper evaluates model performance using Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE), and R-squared (R^2) metrics. Experimental results show that Random Forest achieves the highest prediction accuracy with the smallest MAE (0.33), MSE (0.26), and RMSE (0.51) values, and the R^2 value (0.78) closest to 1. The Decision Tree model demonstrates moderate accuracy but shows signs of overfitting, while Linear Regression exhibits the lowest accuracy but best robustness with minimal difference between training and test performance. These findings suggest that ensemble methods like Random Forest are particularly effective for housing price prediction, though model choice should consider the specific trade-offs between accuracy and robustness. The study provides valuable insights for selecting appropriate models in real estate price prediction applications.

1 Introduction

In modern society, buying a house is a concern for many adults. When purchasing a house, people consider not only housing needs but also price. Factors influencing housing prices can be broadly categorized into three categories. The first category includes property conditions, such as age, building materials, floor space, and parking availability; the second category is surrounding environmental conditions, including public amenities, transportation, and educational institutions; and the third category is location, including proximity to the Central Business District, accessibility to public transportation, and distance to the nearest retail center [1-2].

Using a rational approach to inferring housing prices can help people make more informed financial decisions and better manage their personal finances [3]. The real estate industry is greatly influenced by artificial intelligence (AI). Various techniques are currently used to predict housing prices, for example, linear regression and other prediction models.

* Corresponding author: yhj24@tamu.edu

These algorithms can consider factors such as location, floor space, number of bedrooms and bathrooms, and other property-related characteristics. This information is invaluable to real estate agents, investors, and homeowners who need to make informed market decisions [4].

Due to technological advancements, a wide variety of prediction models have emerged, making it difficult to determine which model to use for predicting housing prices. Different models produce varying results. Therefore, it is crucial to select the most accurate and appropriate model based on the specific conditions of the property. This article will use California Housing Prices data to extract features such as area and age, and compare three models. Using model training, a single dataset was employed to train all models and preprocessing conditions, and their performance is compared using metrics such as prediction accuracy and overfitting test.

2 Related Work

Early housing price prediction studies predominantly employed linear regression, with Multiple Linear Regression (MLR) serving as the most prevalent approach for modeling relationships between continuous dependent variables and multiple independent variables [5]. Theoretically, however, multiple regression-based models emphasize statistical inference over prediction, leading research to primarily focus on identifying significant influencing factors, assessing property economic values, and establishing causal relationships among regressors. While methodologically substantial, this approach demonstrates inherent limitations in statistical prediction of real estate pricing.

Recent technological advancements have accelerated machine learning development. As a crucial artificial intelligence subfield, machine learning addresses knowledge extraction from data and connects this process to broader inference concepts [6]. The learning procedure typically involves two phases: identifying underlying dataset relationships and applying these relationships to forecast future system outputs [7]. Machine learning methodologies fall into three primary categories: supervised learning with known input-output mappings (including the three experimental models), unsupervised learning for pattern discovery without labels (e.g., K-means, Principal Component Analysis, t-Distributed Stochastic Neighbor Embedding), and reinforcement learning where agent-like models learn strategies through environmental interactions (e.g., Q-learning, Deep Q Network). The evolution of machine learning has popularized decision trees and random forests, enhancing prediction accuracy through hierarchical feature space partitioning via binary splits that optimize information gain [8]. These tree-based structures facilitate sample classification or regression through sequential decision rules and leaf node assignments [9]. As supervised learning models, random forests have demonstrated remarkable performance, with Adetunji et al. reporting ±5 error margin predictions on the Boston housing dataset from the University of California, Irvine (UCI) Machine Learning Repository [10]. Some studies have also used neural networks, but these methods are computationally expensive.

However, comparative research on different models is limited. Even when predicting the same housing price data, different models can produce different results. Therefore, this article focuses on comparing the prediction accuracy, computational efficiency, and interpretability of different models, thereby helping people more accurately predict housing prices and make better decisions.

3 Methodology

3.1 Data preprocessing and environment preparation

This article uses the California Housing Prices dataset provided by scikit-learn. The dataset has 20,640 samples and 9 feature columns, and contains eight input features and one target variable. The target variable in this dataset is the median house value, standardized in units of $100,000. Critical predictive features encompass median income levels, housing age, along with average room counts and bedroom quantities per household. These selected characteristics capture fundamental dimensions of housing quality, spatial attributes, and socioeconomic factors that collectively influence regional housing valuations. The median income parameter particularly reflects purchasing power constraints within geographical segments, while structural characteristics like room dimensions and property age contribute to quantifying physical asset quality. This feature combination establishes a multidimensional framework for analyzing housing price determinants across California's diverse real estate market.

This experiment uses Python, scikit-learn, pandas, and matplotlib. Use scikit-learn's fetch_california_housing() to load the data and convert it to a Pandas Dataframe. This dataset is relatively clean, and no missing values were found after checking for missing values. The imputation of missing values is performed using the mean or median of the dataset. The distribution of income and house prices was examined, and the Interquartile Range was used to check for outliers. If any were present, these were removed or truncated. Some features have significantly different dimensions (e.g., income vs. latitude), so normalization was performed to normalize all continuous variables to a mean of 0 and a variance of 1. The paper screened features using correlation coefficients or model-based importance, retaining only those with a significant impact on housing price prediction, such as median income and average number of rooms, while discarding features with low correlation. Finally, the data is divided into an 80% training set and a 20% test set.

3.2 Model selection and training procedure

This experiment selected three models. Linear Regression, as one of the simplest and most classic regression methods, served as a comparison. Decision Tree can handle nonlinear relationships and offers better feature interpretation. Random Forest is an ensemble method that is generally more accurate and stable than a single model. The goal of this experiment was to use these three models to predict housing prices using the same California Housing Price data set and compare their performance. This experiment employed three training steps: training/test set splitting, cross-validation, and hyperparameter setting.

For model implementation, the three algorithms were imported from standard libraries and applied to the dataset, which was partitioned into training and test sets with an 80:20 ratio. Following model training on the training subset, predictive performance was assessed using the test set.

The evaluation employed 5-fold cross-validation, where the dataset was divided into five equal segments. In each iteration, one segment served as the validation set while the remaining four formed the training set. This process ensured every segment functioned as validation data exactly once, with the final performance metric derived from averaging all five validation results.

For hyperparameter settings, a set of hyperparameters to be adjusted is defined for each model. For the decision tree model, the maximum depth is set to {5, 10, 15}, and the minimum number of splits is set to {2, 5, 10}. For the random forest model, the tree count is fixed at {100, 200, 300}, and the maximum depth is defined at {10, 20, None}. The paper uses a grid search approach to train and validate each hyperparameter combination, employing cross-validation during training to ensure robustness of parameter selection. The paper compares the average validation set performance (e.g., MSE, R^2) of different parameter combinations to select the best-performing hyperparameter configuration. Finally, the paper uses the optimal hyperparameters for training and evaluating the final model.

3.3 Evaluation metrics

This experiment uses Cross Validation Mean Squared Error (CV MSE) and best_params to evaluate the prediction results of different models on the training set. CV MSE is used to test different parameter combinations on the training set. best_params selects the parameter configuration with the smallest error based on the CV MSE values of all parameter combinations as the optimal parameter configuration for training the model. The following is the formula for CV MSE, where k is the number of folds to divide the data into.

$$CV\ MSE\ = \frac{1}{k}\sum_{j=1}^{k} MSE_j \tag{1}$$

MAE measures the average magnitude of errors between predicted and actual values, providing an intuitive assessment of prediction accuracy. Its advantages are that it is easy to interpret and is not easily affected by abnormally large squared errors. The following is the formula for MAE, where y_i represents the true value, $\hat{y}_i$ denotes the predicted value, and n is the total number of samples.

$$MAE = \frac{1}{n}\sum\nolimits_{i=1}^{n} \backslash y_i - \hat{y}_i \backslash \tag{2}$$

MSE quantifies prediction accuracy by averaging the squared differences between predicted and actual values. This squaring operation makes the metric particularly sensitive to larger errors, as it amplifies their contribution to the overall score. The following is the formula for MSE.

$$MSE\ = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3}$$

RMSE emphasizes the impact of large prediction errors by calculating the square root of the average squared differences. This characteristic makes it particularly sensitive to significant deviations, effectively measuring a model's ability to handle extreme values. The following is the formula for RMSE.

$$RMSE\ = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4}$$

The R^2 Score determination coefficient model can explain the proportion of the target variance and more intuitively illustrate the goodness of fit. The following is the formula for R^2.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \overline{y})^2} \quad (5)$$

Finally, an overfitting test is performed, calculating key metrics (MAE, RMSE) on both the machine and test sets. If the training error is significantly smaller than the test error, the model is considered overfitted.

4 Result and Analysis

4.1 Results

When evaluating the training set in the normalized target space, Linear Regression achieved a CV MSE of 0.55, Decision Tree achieved a CV MSE of 0.87, and Random Forest achieved a CV MSE of 0.42. The Decision Tree achieved the highest CV MSE. The optimal hyperparameters identified through tuning were as follows: for the Decision Tree model, a maximum depth of 10 with a minimum of 10 samples required for splitting; for the Random Forest model, an unlimited maximum tree depth with 300 estimators. Fig. 1 shows the CV MSE values for the three models.

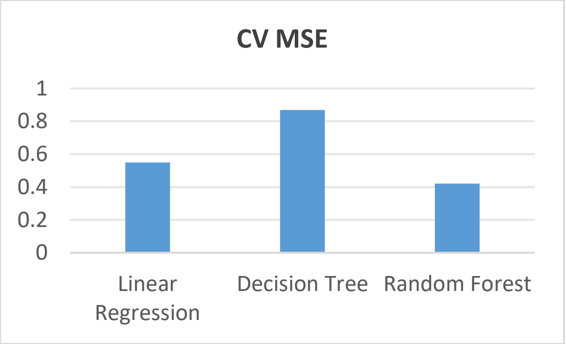

Fig. 1. CV MSE values of three models (Photo credit: Original).

According to experimental results, in the normalized target space, the MAE value for Linear Regression is 0.53, Decision Tree is 0.45, and Random Forest is 0.33. Linear Regression has the largest MAE value, while Random Forest has the smallest. The MAE values of the three models are shown in Fig. 2.

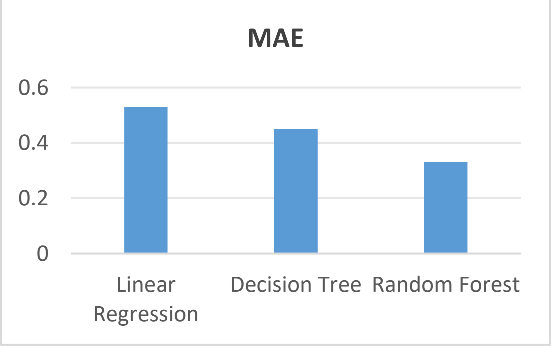

Fig. 2. MAE values of three models (Photo credit: Original).

In the normalized target space, the MSE value for Linear Regression is 0.53, Decision Tree is 0.48, and Random Forest is 0.26. Linear Regression has the largest MSE value, while Random Forest has the smallest. The MSE value for Decision Tree decreases by 9.43% compared to Linear Regression, while the MSE value for Random Forest decreases by 50.94%. Fig. 3 shows the MSE values for the three models.

Fig. 3. MSE values of three models (Photo credit: Original).

In the normalized target space, the RMSE value for Linear Regression is 0.73, Decision Tree is 0.70, and Random Forest is 0.51. The RMSE values for Linear Regression and Decision Tree are relatively close, with Random Forest having the smallest RMSE, which is 30.13% lower than the Linear Regression model. Fig. 4 shows the RMSE values for the three models.

Fig. 4. RMSE values of three models (Photo credit: Original).

In the normalized target space, Linear Regression achieves an R^2 of 0.55, Decision Tree achieves an R^2 of 0.58, and Random Forest achieves an R^2 of 0.78. The R^2 values for Decision Tree and Linear Regression are relatively close, with only a 5.45% increase compared to the Linear Regression model. Random Forest has the highest R^2 value, increasing by 41.82% compared to the Linear Regression model. Fig. 5 shows the R^2 values for the three models.

Fig. 5. R^2 values of three models (Photo credit: Original).

In the normalized target space, the evaluation indicators for the Linear Regression training and test sets are relatively close, indicating that Linear Regression is relatively robust. The MSE, RMSE, and MAE for the Decision Tree training set are all 0, and the R^2 is 1, but the test set results differ significantly from the training set, indicating that the Decision Tree is overfitting. The performance gap between training and test sets for Random Forest was intermediate—wider than that observed for Linear Regression but narrower than for Decision Tree. This comparative pattern across the three models is illustrated in Fig. 6, which presents their respective evaluation metrics on both datasets.

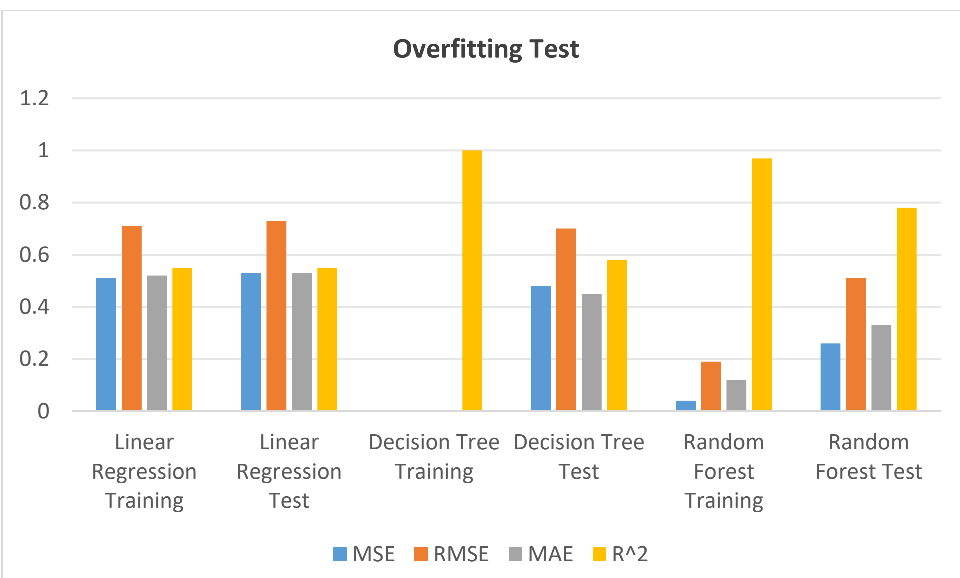

Fig. 6. Overfitting test of three models (Photo Credit: Original).

4.2 Analysis

A comprehensive comparison of model performance demonstrates clear hierarchical patterns in both predictive accuracy and generalization capability. Among the three evaluated approaches, Random Forest achieved superior predictive precision, as evidenced by its optimal performance across all error metrics (lowest MSE, MAE, and RMSE) and the highest explanatory power (R^2 closest to 1). The Decision Tree model secured an intermediate position, outperforming Linear Regression in accuracy metrics while falling short of Random Forest's results.

Regarding model robustness, the analysis reveals an inverse relationship between training performance and generalization capability. While exhibiting perfect metrics on training data, the Decision Tree manifested substantial performance degradation on test data, indicating pronounced overfitting. In contrast, Linear Regression maintained the most consistent performance between training and test sets, confirming its superior stability despite lower absolute accuracy. Random Forest presented a balanced profile—demonstrating marginally reduced consistency compared to Linear Regression, but significantly better generalization than the Decision Tree.

This systematic evaluation thus establishes Random Forest as the optimal choice for prediction tasks prioritizing accuracy, while identifying Linear Regression as the most suitable option when model stability is paramount.

5 Conclusion

This experiment compared the prediction results of three models on the same set of housing price data. After data processing, the dataset was partitioned into two parts: The dataset was partitioned using an 80-20 split for training and testing, respectively. Through systematic cross-validation and hyperparameter optimization, optimal parameter configurations were identified for model training and evaluation. Comparative analysis of test set performance metrics—including MSE, RMSE, MAE, and R^2—demonstrated that the Random Forest model achieved superior predictive accuracy, consistently outperforming the other two approaches across all evaluation criteria. Decision Tree was second, and Linear Regression had the lowest prediction accuracy. Linear Regression had the smallest difference in all metrics between the training and test sets, indicating the most robust model. The Random Forest model performed second, while the Decision Tree model performed best on the training set, achieving an MSE, RMSE, and MAE of 0 and an R^2 of 1. However, the difference in all metrics between the training and test sets was the largest, indicating overfitting. Due to time and other constraints, this experiment was conducted only on California Housing Prices data; the models may not be applicable to other regions. In the future, if conditions permit, the paper can consider using deep learning methods or adding more socioeconomic characteristics (such as income level, traffic conditions, etc.) to obtain more comprehensive and accurate prediction results, thereby helping people better predict housing prices and make more informed economic decisions.

References

1. Soltani, M. Heydari, F. Aghaei, et al., Housing price prediction incorporating spatio-temporal dependency into machine learning algorithms. Cities 131, 103941 (2022)
2. P.Y. Wang, C.T. Chen, J.W. Su, et al., Deep learning model for house price prediction using heterogeneous data analysis along with joint self-attention mechanism. IEEE Access 9, 55244-55259 (2021)
3. P.F. Pai, W.C. Wang, Using machine learning models and actual transaction data for predicting real estate prices. Applied Sciences 10, 5832 (2020)
4. K.P. Fourkiotis, A. Tsadiras, Comparing machine learning techniques for house price prediction, in Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations, (2023), pp. 292-303

5. Q. Zhang, Housing price prediction based on multiple linear regression. Scientific Programming 2021, 7678931 (2021)
6. K. Kourou, T.P. Exarchos, K.P. Exarchos, et al., Machine learning applications in cancer prognosis and prediction. Computational and structural biotechnology journal 13, 8-17 (2015)
7. W.K.O. Ho, B.S. Tang, S.W. Wong, Predicting property prices with machine learning algorithms. Journal of property research 38, 48-70 (2021)
8. S. Lahmiri, S. Bekiros, C. Avdoulas, A comparative assessment of machine learning methods for predicting housing prices using Bayesian optimization. Decision Analytics Journal 6, 100166 (2023)
9. Z. Zhang, Decision trees for objective house price prediction, in Proceedings of the 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), (2021), pp. 280-283
10. A.B. Adetunji, O.N. Akande, F.A. Ajala, et al., House price prediction using random forest machine learning technique. Procedia Computer Science 199, 806-813 (2022)