

House Price Prediction Using Machine Learning: Analysis of XGBoost and Gradient Boosting Models

An Li

*Faculty of Science and Technology, Beijing Normal-Hong Kong Baptist University, Zhuhai, China
s230026070@mail.uic.edu.cn*

Abstract: This study investigates the optimization and application of Distributed Machine Learning (DML) techniques, essential for addressing the computational challenges of large-scale datasets in modern applications such as autonomous driving and natural language processing. The primary objective is to evaluate the performance and characteristics of key DML methods, including MapReduce, Parallelized Stochastic Gradient Descent (PSGD), and Federated Learning (FL). A comprehensive analytical framework is employed to explore the underlying mechanisms of each method. For MapReduce, the study examines data partitioning, mapping, shuffling, and reducing processes to assess its efficiency in managing large datasets. In PSGD, the research analyzes the methods of data allocation, local gradient computation, and parameter aggregation to highlight its parallelization benefits and identify potential bottlenecks. FL focuses on its privacy-preserving model training process, which includes client-server communication, local updates, and global aggregation. Experimental results using the ImageNet dataset and Gboard user log data show that while MapReduce offers fault tolerance, PSGD accelerates training on large datasets, and FL ensures data privacy. However, limitations such as communication overhead and convergence issues remain, underscoring the trade-offs between efficiency, scalability, and privacy in DML applications.

Keywords: Federated Learning, Machine Learning, Parallelized Stochastic Gradient Descent

1. Introduction

This paper focuses on using machine learning technology to predict house prices. House price forecasting is the core issue in economics and real estate. Accurate price forecasting enables buyers to make decisions more scientifically, optimizes resource allocation, and promotes the development of the estate market. With the rapid growth of data science, regression analysis, feature engineering, and ensemble learning methods have been widely used in housing price prediction. This study compares the performance of various regression models and integrated algorithms. It provides a tool for housing price prediction to provide valuable scientific references and support for buyers.

Recent studies have highlighted various machine learning techniques for predicting housing prices. Alshammari found that Random Forest (RF) outperforms other models like Linear Regression (LR) and Decision Trees (DT), while Support Vector Regression (SVR) performs poorly in comparison [1]. Wang et al. demonstrated that Artificial Neural Networks (ANN) can predict housing prices in Singapore. However, their performance is inconsistent due to varying initial conditions, suggesting a need for better feature selection and testing [2]. Tang et al. found that as an ensemble learning

algorithm, Random Forest provides high accuracy and minimal error, making it reliable for large-scale predictions [3]. Joseph et al. used Extreme Gradient Boosting (XGBoost) for real-time house price prediction in Bengaluru, achieving 91.77% accuracy [4]. Kin applied machine learning techniques, including ANN and Chatbots, for housing price predictions in Malaysia, aiding buyers and sellers in understanding market trends [5]. Saini also concluded that models like XGBoost outperform others in predicting housing prices, especially regarding accuracy [6]. These studies confirm that Random Forest and XGBoost are among the most effective models for housing price prediction.

The main objective of this study is to compare the performance of different machine learning models in predicting house prices. Specifically, first, data preprocessing techniques such as data visualisation and handling missing values are applied to prepare the dataset for modelling. Second, multiple regression models, including RF, SVR, and XGBoost, predict house prices based on various features in this dataset. Third, the predictive performance of the different models is analyzed and compared using evaluation metrics like R2 and accuracy rate. In addition, feature importance is assessed to identify which characteristics most significantly affect house prices. Meanwhile, the study also addresses challenges in improving the accuracy of predictions by tuning model parameters and optimizing the feature selection process. The experimental results demonstrate that XGBoost outperforms other models in prediction accuracy, highlighting the importance of gradient-boosting techniques in this context.

2. Methodology

2.1. Dataset description and preprocessing

The data set used in this project is from the Kaggle platform called "House Prices-Advanced Regression Techniques"[7]. The goal is to predict the sales price of a property. This dataset contains 81 numerical and categorical features, such as house size, year of construction, garage size, etc. Data preprocessing: First, for missing values, numerical features are filled with the median, and categorical features are filled with the mode; Then, the class variables are processed by unique thermal coding. Then, the logarithmic transformation of the target variable, sale price, and numerical characteristics is carried out to approximate a normal distribution. In addition, Grid search cross-validation (GridSearchCV) is used for hyperparameter tuning to optimize the model performance [8]. The application of data sets mainly focuses on predicting house prices by analyzing property characteristics. Distribution characteristics of the Sale price and outliers of the data are shown in Figure 1.

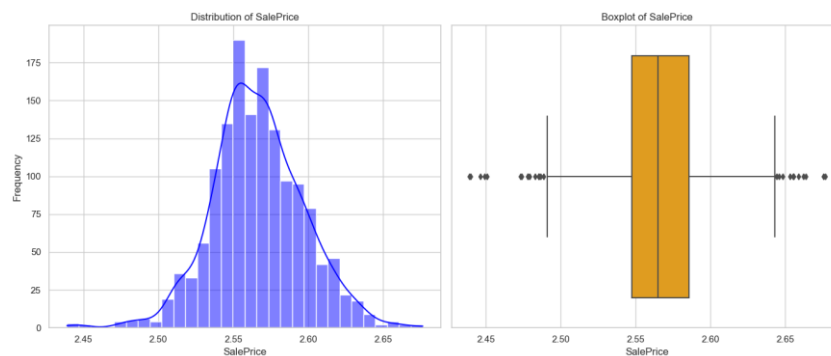


Figure 1: Visualization of distribution characteristics of sales prices and outliers of data (picture credit: original)

2.2. Proposed approach

Firstly, the data set is preprocessed, including missing value processing, category variable coding, logarithmic transformation and numerical feature standardization. The training with multiple regression models, including linear regression, decision trees, random forests, XGBoost, SVR, Lasso regression, Ridge regression, gradient lift, etc., evaluated model performance using R² scores. The preliminary comparison results show that the R² scores of XGBoost and gradient lift models are the highest, so these two models are selected for separate optimization. The overall process is shown in Figure 2.

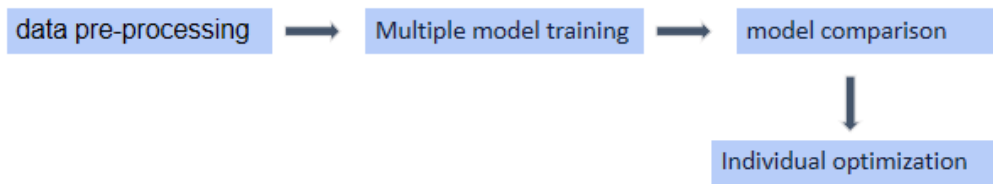


Figure 2: Overall process (picture credit: original)

2.2.1. Introduction of XGBoost

XGBoost is an efficient ensemble learning algorithm based on Gradient Boosting Decision Trees (GBDT), which is designed to provide efficient, flexible, and portable implementations for classification and regression tasks. The core idea of XGBoost is to improve the model's accuracy by calculating the gradient of the objective function at each iteration and splitting the features that can best reduce the prediction error.

XGBoost model supports various loss functions and evaluation metrics, and allows Lasso (L1) and Ridge (L2) regularization to control model complexity and avoid overfitting. Moreover, XGBoost can also process missing values automatically, reducing the reliance on data preprocessing by learning to choose the appropriate path when data is missing. In addition, XGBoost limits the maximum depth of the decision tree by tree pruning technology, which is a practical approach to prevent overfitting and improve the model's generalisation ability.

In this project, the training process for XGBoost consists of five main steps. First, do some data preprocessing. The missing values are filled in, and the numerical features are logarithmically transformed to make the data closer to a normal distribution. The data set is divided into a training set and a test set. Moreover, standardizing the numerical and class features is encoded by a unique heat. Next, the model is trained, and hyperparameter tuning is carried out through GridSearchCV to optimize multiple hyperparameters of the model and select the best model. Finally, using R² score of the test set to evaluate the model performance to ensure the accuracy and stability of the model. The training process is shown in Figure 3.

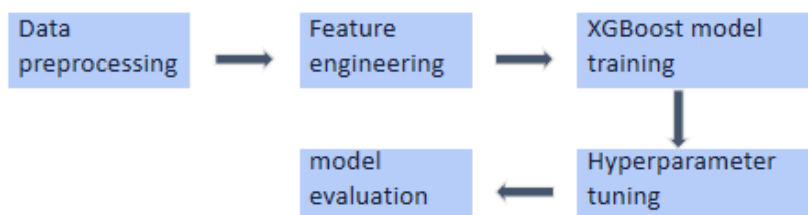


Figure 3: Training process of XGBoost (picture credit: original)

2.2.2. Introduction of gradient boosting

Gradient Boosting is an ensemble learning method based on decision trees and belongs to the Boosting algorithm. The algorithm is iteratively trained by multiple weak learners (usually decision trees), with each new decision tree designed to correct the prediction error of the previous one. The core idea of Gradient Boosting is to minimize the loss function by using the gradient descent method and update the model's prediction results by using the objective function's gradient in each iteration.

Gradient Boosting has high flexibility, supports various loss functions and evaluation indicators, and can adapt to multiple regression tasks. It can adapt to data, mainly when it contains a certain amount of noise, so it can effectively train. It also has some ability to prevent overfitting.

The training process of gradient lifting is gradual; in each round of training, the new decision tree is adjusted based on the result of the previous tree. By learning step by step to reduce errors, gradient lifting can build a robust classifier. The data preprocessing method is consistent with the XGBoost model, handling missing values (categorical features filled with 'Missing', numerical features filled with the median) and applying a logarithmic transformation to numerical features. After preprocessing, the dataset is split into training and test sets. Using GridSearchCV to perform Hyperparameter tuning, optimizing parameters such as the number of estimators, learning rate, and maximum depth to improve the model's performance. Unlike the XGBoost model training, the Gradient Boosting model training uses 5-fold cross-validation to evaluate different hyperparameter combinations, reducing model variance and ensuring robust performance. The prediction of model accuracy is assessed using R^2 scores. The training process is shown in Figure 4.

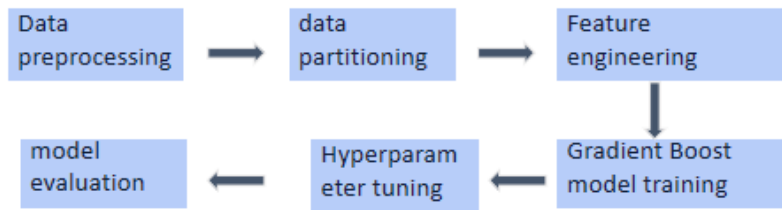


Figure 4: Training process of Gradient Boosting (picture credit: original)

2.2.3. Loss function

In XGBoost, the loss function is the squared error loss function. This loss function measures the squared difference between the predicted and actual values of the model, which means the model is trained by minimizing the error between each prediction. For a prediction problem, the squared error loss function is the squared variance between a single prediction and the actual value. One characteristic of square error loss is that it is sensitive to significant errors, so significant errors will get more attention in the model training process, which helps the model minimize the occurrence of substantial errors:

$$\text{Squared Error Loss} = (y_{true} - y_{pred})^2 \quad (1)$$

Similarly, Gradient Boosting uses mean squared error (MSE) as the default loss function. Gradient Boosting gradually reduces errors by optimizing the loss function, with each iteration training a new tree based on the current residual. The mean square error is the average of the squared errors of all sample points.

$$MSR = \frac{1}{n} \sum_{i=1}^n (y_{true} - y_{pred})^2 \quad (2)$$

2.3. Implementation details

This project used a variety of standard machine learning libraries and tools to optimize models. These libraries provide powerful support for data processing, model training and evaluation, which helps to improve prediction accuracy and model performance. First, the data preprocessing section uses pandas and NumPy, which are helpful in data cleaning, conversion, and numerical calculations. Missing values and normalizing data ensure the data is properly processed before entering the model. One-Hot Encoder encodes categorical variables to fit them into regression models.

GridSearchCV finds the best combination of hyperparameters, and estimators choose numbers of 200, 300, and 500 trees to evaluate model performance. The learning rate is 0.01, 0.05 and 0.1; the maximum depth of the tree is 3, 5, 7 and 10, respectively; the minimum sample number of the split node is set to 2, 5 and 10; the minimum sample number of the leaf node is set to 1, 2 and 4, and the subsample is 0.8, 0.9 and 1.0. By tuning these hyperparameters and selecting the model, this project can optimize the model's performance and choose the most suitable combination of hyperparameters for prediction.

3. Results and discussion

This paper first discusses the importance of housing price prediction in economics and real estate, emphasizing that accurate prediction can help buyers make scientific decisions, optimize resource allocation, and promote market development. The data contains 81 features, and the data preprocessing process is detailed, including missing value processing, logarithm transformation of numerical features, and hyperparameter tuning steps. Regarding model selection, two machine learning models, XGBoost and gradient Lift, are discussed in detail, explaining their principles, advantages, and how to improve the prediction accuracy through optimization. Finally, by comparing the performance of different models, the XGBoost and gradient lift models are selected, and the R^2 score is used to evaluate models.

3.1. Comparison of model performance in house price prediction

As shown in Figure 5, the chart shows the R^2 scores of the different house price forecasting models, with the optimized XGBoost and gradient boosting models performing best, whose scores are close to 1, showing their superior performance in prediction accuracy. This indicates that optimizing hyperparameters and model structure enables the ensemble learning model to effectively capture complex nonlinear relationships in housing price data and reduce overfitting, thus significantly improving prediction accuracy. Other models, such as Random Forest, Lasso Regression, and Ridge Regression, whose score is high in the 0.8 to 0.9 range, failed to achieve the effect of optimized integrated models, showing that these models still have certain limitations when facing complex data. In contrast, the Decision Tree has a low score of about 0.7, indicating that it cannot take full advantage of the complex features of the data, resulting in significantly inadequate prediction accuracy. Overall, the ensemble learning model improves the prediction accuracy and enhances the model's generalization ability by combining multiple models and parameter optimization. However, it is difficult for a simple model to obtain ideal results in the face of multidimensional and nonlinear data. These findings highlight the importance of ensemble learning models in complex regression tasks such as house price prediction.

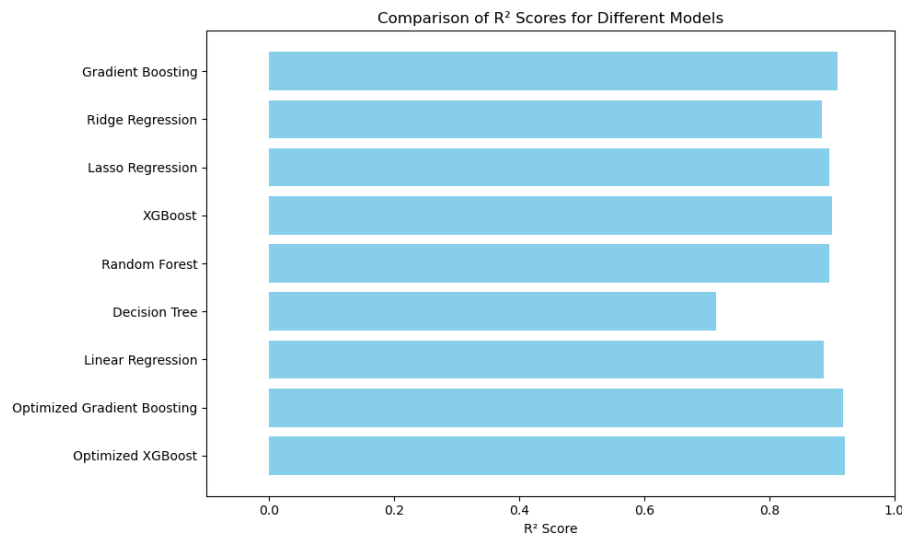


Figure 5: Comparison of R^2 scores for different models (picture credit: original)

3.2. Distribution of sales prices and normality analysis

As shown in Figure 6, sales prices exhibit a typical unimodal distribution, with most houses clustered around 2.55, which fits the pattern of a nearly normal distribution. The black curve of the density map is the fitted standard distribution curve, showing the symmetry and concentration of the data. This indicates that most houses are relatively close in price, and there are fewer houses at extreme prices, which conforms to the common law of the real estate market.

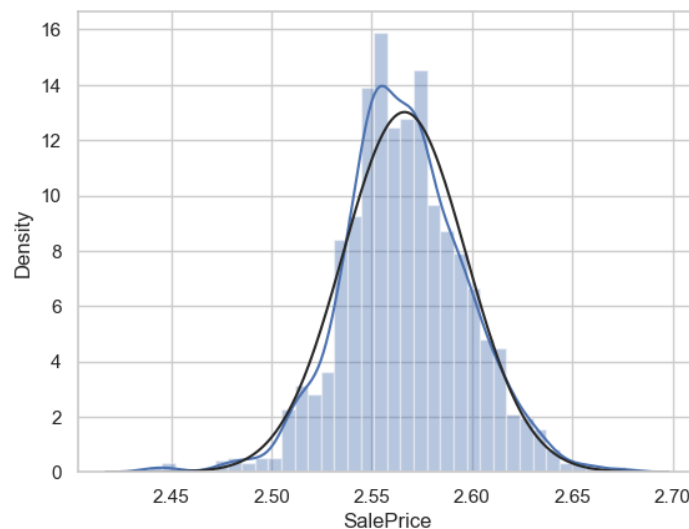


Figure 6: Sales price distribution map (density map) (picture credit: original)

3.3. Normality test and outlier analysis of sales price data

Figure 7 (Q-Q plot) shows sales price data compared to the theoretical normal distribution. Researchers can see from this plot that most data points are located near the linear trend line in red, indicating that the sales price data is roughly in line with the normal distribution. However, at the low and high data points, the data diverges, especially the tails in this plot, showing a specific skew, which

suggests that while most house prices conform to a normal distribution, there are some outliers, especially in extreme price ranges. This data distribution makes more demands on the model.

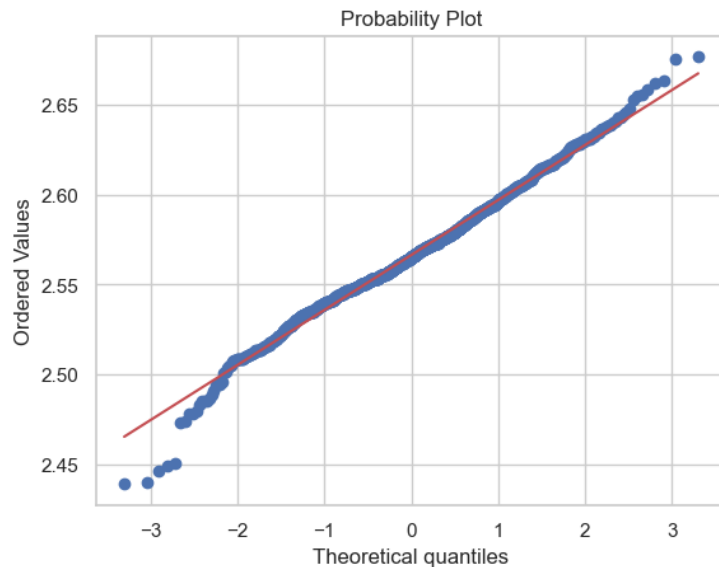


Figure 7: Normal probability graph (Q-Q graph) (picture credit: original)

The distribution characteristics of sales price data are analyzed to compare the performance of different house price prediction models. The results show that optimized integrated learning models, such as XGBoost and gradient boosting, are excellent in predicting complex nonlinear relationships and improving prediction accuracy.

4. Conclusion

This paper introduces machine learning techniques for house price prediction to enhance forecasting accuracy in the real estate market. By employing advanced data science methods, the research seeks to provide valuable tools for buyers, investors, and market analysts to support decision-making. The proposed approach utilizes XGBoost and Gradient Boosting models to analyze house prices based on various property characteristics, such as house size, year of construction, and garage size. The modelling process includes data preprocessing steps, such as handling missing values, applying logarithmic transformations to numerical features, and standardizing the data. Hyperparameters are optimized using GridSearchCV to improve model performance. Extensive experiments demonstrate that XGBoost and Gradient Boosting outperform other regression models, such as Random Forest and Decision Trees, achieving near-perfect prediction accuracy. These results underscore the effectiveness of ensemble learning models in capturing complex nonlinear relationships in housing data. Future research will explore macroeconomic factors, such as interest rates, to examine their impact on house prices and further enhance model stability through advanced feature engineering.

References

- [1] Alshammari, T. (2023). *Evaluating machine learning algorithms for predicting house prices in Saudi Arabia. International Conference on Smart Computing and Application*, 1-5.
- [2] Xu, X., Zhang, Y. (2022). *Residential housing price index forecasting via neural networks. Neural Computing and Applications*, 34(17), 14763-14776.
- [3] Mora-Garcia, R.T., Cespedes-Lopez, M.F., Perez-Sanchez, V.R. (2022). *Housing price prediction using machine learning algorithms in COVID-19 times. Land*, 11(11), 2100.

- [4] Joseph, L.M.I.L., Reddy, E.H., Srinidhi, M., et al. (2024). *Predicting Real-Time House Prices: A Machine Learning Approach Using XGBoost Algorithm*. *Asia Pacific Conference on Innovation in Technology*, 1-4.
- [5] Chee-Kin, C., Arabee, B.A.S.Z., and Batcha, N.K., (2022). *Machine Learning based House Price Prediction Model*. *International Conference on Edge Computing and Applications, Tamilnadu, India*, 1423-1426.
- [6] Zaki, J., Nayyar, A., Dalal, S., et al. (2022). *House price prediction using hedonic pricing model and machine learning techniques*. *Concurrency and computation: practice and experience*, 34(27), e7342.
- [7] Kaggle. (2018). *House Prices - Advanced Regression Techniques*. Retrieved from <https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques>.
- [8] Adnan, M., Alarood, A.A.S., Uddin, M.I., et al. (2022). *Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models*. *PeerJ Computer Science*, 8, e803.