Iraqi Journal for Computer Science and Mathematics

Volume 5 | Issue 4 Article 25

2024

Enhancing Accuracy in Predicting Continuous Values through Regression

Ahmed Aljuboori
Computer Science Department, College of Education for Pure Science / Ibn Al-Haitham, University of Baghdad, Baghdad, Iraq AND Department of Computer Science, Dijlah University College, Baghdad, Iraq, a.s.aljuboori@ihcoedu.uobaghdad.edu.iq

M. M. A. Abdulrazzq
Computer Science, Faulty of Innovation & Technology, Taylor's University, Malysia

Recommended Citation

Aljuboori, Ahmed and Abdulrazzq, M. M. A. (2024) "Enhancing Accuracy in Predicting Continuous Values through Regression," *Iraqi Journal for Computer Science and Mathematics*: Vol. 5: Iss. 4, Article 25.
DOI: https://doi.org/10.52866/2788-7421.1227
Available at: https://ijcsm.researchcommons.org/ijcsm/vol5/iss4/25

IRAQI JOURNAL FOR COMPUTER SCIENCE AND MATHEMATICS 2025;5:326–336

REVIEW

Enhancing Accuracy in Predicting Continuous Values through Regression

Ahmed Aljuboori [a,b,*], **M. M. A. Abdulrazzq** [c]

[a] Computer Science Department, College of Education for Pure Science/Ibn Al-Haitham, University of Baghdad, Baghdad, Iraq
[b] Department of Computer Science, Dijlah University College, Baghdad, Iraq
[c] Computer Science, Faulty of Innovation & Technology, Taylor's University, Malysia

ABSTRACT

Enhancing the accuracy in predicting continuous values remains a significant challenge, especially when dealing with imbalanced data and choosing appropriate models. Regression techniques are widely used in data mining, and machine learning fields for this purpose. However, the traditional algorithms struggle to achieve high accuracy because of the limitations in dealing with complex data and imbalanced distribution. This study addresses these gaps by proposing a new framework that evaluates multiple regression models using the Boston House Pricing Dataset (BHD). The examined models involve simple linear, multiple linear, Polynomial, Lasso, Ridge, Random Forest, Keras and Gradient Boosting regression. The models are compared using evaluation metrics such as R-squared Score (R2), Mean Squared Error (MSE), and Mean Absolute Error (MAE). Among the examined models, the first promising outcomes indicate that Random Forest and Ridge regressors scored a high level of R2 i.e. 89.9 and 88.3, respectively. In addition, The Gradient Boosting model offers the best result of R2 92 with MSE 0.72 and MAE 2.00. To further enhance the accuracy of the best model, this research applies two techniques. Re-sampling and optimization using the RandomizedSearchCV tuned hyper-parameter improved R2 score to 93.2 with a better MSE of 0.015 and MAE of 0.82. These findings prove a significant improvement in model performance and offer a potential for practical application in real-world scenarios.

Keywords: Gradient Boosting, Keras, Lasso, Linear, Polynomial, Random Forest, Regression, Ridge

1. Introduction

Regression algorithms have become important in many fields, such as marketing, economics, finance, and healthcare because of their ability of supporting decision-making and data analysis. However, achieving high accuracy in predicting continuous values has long been a challenge due to data complexity and model selection. Especially with a noisy or imbalanced dataset that contains a nonlinear relationship [1, 2]. Addressing these challenges becomes crucial for house pricing and financial problems.

The Boston Housing Dataset (BHD), is widely used for regression research, due to its complexity and relevance to real-world problems. Predicting house pricing based on traditional features like location, crime statistics, and socio-economic [3] identifies the limitations of classic algorithms such as linear regression which often fails to capture nonlinear complex data. While classic regression approaches fail to capture nonlinear complex data, sophisticated techniques such as random forest and gradient boosting offer greater results but demand important parameter tuning to achieve optimal results [4, 5].

Studies tried to address these challenges, for example [3] tested simple linear regression, polynomial regression, ridge regression and lasso regression resulting in the best Ridge and Lasso's Regression registered R^2 scores of 88.28 and 89.79 respectively. Other research investigated ensemble and sophisticated

Received 13 September 2024; accepted 30 December 2024.
Available online 6 January 2025

* Corresponding author.
E-mail address: a.s.aljuboori@ihcoedu.uobaghdad.edu.iq (A. Aljuboori).

https://doi.org/10.52866/2788-7421.1227

machine learning techniques. [6] revealed that XGBoost is superior in capturing complicated associations, whereas support vector machines (SVM) are inferior in accuracy and interpretability [7]. The Authors suffer from choosing a suitable model with high accuracy, i.e. R^2 of 0.920, 0.570, 0.860, 0.640 and 0.910. Random forest approaches have also been frequently used due to their resilience, as noted by Sharma et al. [8], though reasonable tuning is required to avoid overfitting and efficiency when dealing with imbalanced data [9]. Despite Random Forest holds a lot of promise, it is not always sufficient to achieve the highest possible level of accuracy. As an instance, [10] discovered that merging random forest with other approaches enhanced outcomes, but it also brought to light the requirement to deal with the amount of time of training and memory. Similarly, [11] highlighted the potential of deep learning algorithms, such as Keras, to model complicated nonlinear interactions. However, this promise comes at the expense of increased processing intensity and longer training time. The efficiency of advanced gradient boosting methods was further proved by [12], who revealed that these methods could outperform conventional regression techniques when combined with hyperparameter tuning and data resampling procedures.

This research addresses these challenges by proposing a new framework that merges sophisticated regression and optimal preprocessing with the following key contributions. First, this research presents a robust Gradient Boosting technique using SMOTE-Tomek resampling and RandomizedSearchCV hyperparameter optimization to overcome regression model constraints and improve accuracy. Second, A comprehensive comparison of regression models, including linear regression, ridge regression, random forest, and deep learning techniques (Keras), evaluated using metrics such as R^2, MSE, and MAE [13, 14] and [7]. Third, Validation of the proposed model by illustrating its superiority over traditional techniques regarding accuracy and reliability.

This article is structured as follows: Section 1 presents the introduction. Section 2 states the literature review. Section 3 describes the methodology. Section 4 and Section 5 present the results and discussion, respectively. Section 6 states the conclusion and future work.

2. Related work

Regression techniques are frequently used in machine learning and data mining fields because of their capacity to handle complex data patterns. However, predicting continuous values using BHD was a challenge for many researchers. Several studies have examined different regression models to address this issue, each approach with its strengths and limitations.

[3] Implemented simple linear regression, polynomial, lasso, and ridge regression on the BHD. While Lasso regression outperformed among these models. The authors proved that lasso is computationally intensive because of cross-validation parameter tuning. Compared to this study, it limits the performance of complex data because it tends to oversimplify the structure.

[6] Evaluated the use of linear regression, random forest, XGBoost, and SVM for predicting BHD price. XGBoost presented a superior performance because it can deal with complex relationships, while the SVM struggled to achieve high accuracy. In contrast, the current research addresses this gap by integrating RandomizedSearchCV for hyperparameter optimization which refined model performance. Moreover, the finding of [6] were supported by [7] who reported butter R^2 of XGBoost with low accuracy of SVM. Both approaches are limited to hyper-parameter tuning or improving the classical models to increase the accuracy.

[8] Highlighted the effectiveness of random forest by capturing nonlinear relationships on BHD. While random forest scored a reasonable performance, the model limits to handle specific variables. [9] improved the random forest accuracy by integrating genetic algorithms. The results show reasonable performance with strong stability and reliability. Despite these improvements, the random forest model consumed a long training time and did not achieve the highest R^2 scores compared to the improved methods.

[10] Compared the linear regression, random forest and SVM repressor using R^2, MAE, and MSE indicators. The study discovered that random forest regression was the best among the tested models. However, the computational cost of assembling multiple forests increased the training time. This limitation sets the need for a more efficient method that can achieve high accuracy without extra computational costs.

The current research experimented with advanced techniques by integrating SMOTETomek resampling and RandomizedSearchCV into Gradient Boosting to address the gap in the literature. This framework not only deals with imbalanced data effectively but also tunes hyperparameters to achieve a high R^2. It also overcomes the Keras algorithm, presenting better accuracy than previous research for instance [9].

This research fills the gaps by selecting the best model, providing new experiment arguments, and

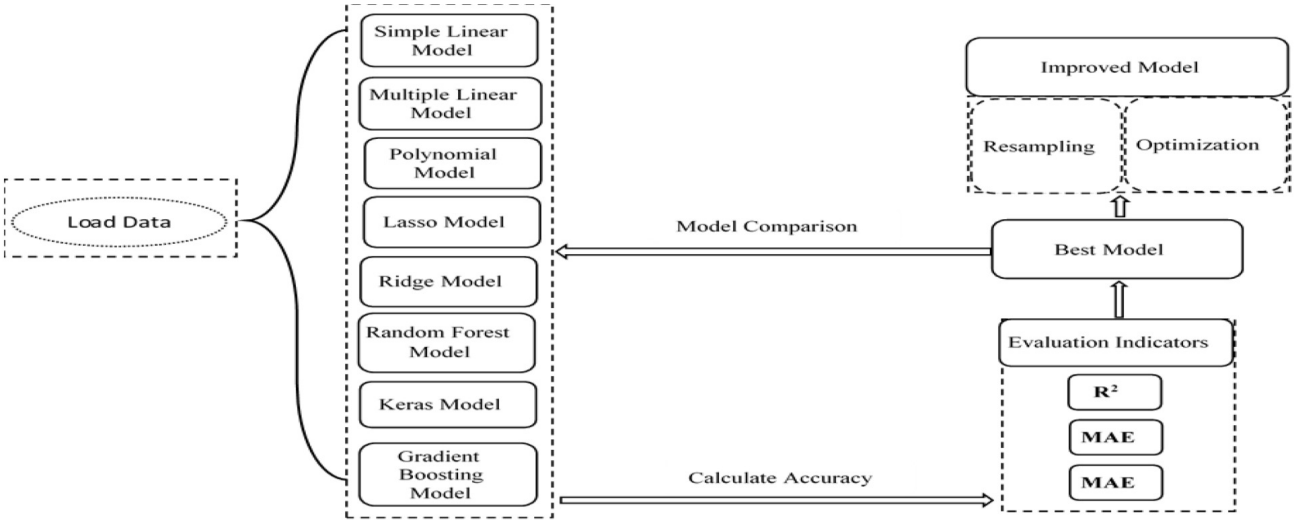

Fig. 1. Model framework.

proposing a new framework applied to the best model to improve the performance of the tested regression algorithms.

3. Methodology

This research compares eight regression techniques: simple linear, multiple, linear polynomial, lasso, ridge, random forest, Keras, and gradient boosting' regression. Boston housing prices dataset (BHD) is used as a benchmark for this research. BHD contains $n = 506$ observations with $p = 14$ features. This experimental research focuses on the algorithm that scores high accuracy of R^2 and on improving its performance using integrated procedures. The experiment started with linear regression, followed by other regressors to seek the best accuracy on the benchmarked BHD. The accuracy of the proposed framework is calculated through evaluation indicators R^2, MSE and MAE. The best model with high accuracy is then compared to the rest of the experimented models until the best results are reached. Finally, two techniques are applied to the best model, i.e., re-sampling and optimization, to improve the accuracy and fulfil the aim of this research, as shown in Fig. 1 each model is discussed as follows.

3.1. Linear regression

In simple linear regression, a linear relationship is established between the dependent variable y and a single independent variable X. This relationship is modeled by fitting a regression line represented by Eq. (1).

$$y = \beta_0 + \beta_1 X + \epsilon \tag{1}$$

β_1, β_1 refers to the vector of coefficients, and ϵ is the error rate [4].

Nevertheless, it is crucial to recognize that the simple linear regression model's predictions may not always be precise. The limitation of the model is overcome by utilizing the error term ϵ. In this study, Linear regression was examined first as a baseline for the relationship between variables. It was considered the starting point to determine a high level of accuracy. Still, due to the limit of a single predictor, a further method is tested to obtain the best model with the highest accuracy.

3.2. Multiple linear regression

Multiple Linear is an extension of simple linear Regression [15]. It models the relationship between several independent variables ($X1, X2, \ldots, X_p$) and the dependent variable y. It considers several features of the dependent variable compared to ordinary linear regression, as the latter only considers one independent variable. Eq. (2) shows the form of the MLR model.

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon \tag{2}$$

β_0 represents the intercept, while $\beta_1, \beta_2, \ldots, \beta_p$ are coefficient of each predictor and ϵ is the error term coefficient most used to resemble the data.

This algorithm has the potential to offer a more precise understanding of the correlation between each aspect and the result. It also showed a better relative

accuracy when compared to simple linear regression because of the use of multiple predictors, but further experiments are needed to fulfil the aim of this research.

3.3. Polynomial features and feature scaling

Polynomial regression enhances the original features by including additional variables of higher order [5]. To identify and extend the simple linear regression with only one feature, X^2, it is added as an extra feature to express the general form of this regressor, as shown in Eq. (3).

$$y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon \tag{3}$$

X^2 represents the n-degree of the polynomial feature. Including these polynomial features enables the model to deal with curves, bends, and the impact on the data. In addition, it improves its ability to detect complicated patterns.

In this study, the polynomial model has shown better performance when compared to the first two algorithms because an appropriate feature scaling procedure is employed to ensure the stability and accuracy of the tested model. Further experiments will be conducted to seek the best model with the best accuracy applied to the BHD.

3.4. Lasso regression

Lasso algorithm [14] eliminates a fundamental challenge in regression analysis, namely Overfitting. When a model becomes complicated by fitting noise, this could lead to poor generalization. The Lasso overcomes this issue by incorporating a penalty term into the linear regression equation, encouraging the model to select a subset of the most pertinent features while reducing the coefficients of less significant ones toward zero. In contrast, the simple linear regression aims to minimize the mean squared error (MSE) between predicted and actual y values. Lasso presents a regularization term but conducts a selection of variables by shrinking some coefficients to zero. The objective function is in Eq. (4).

$$L(\beta) = \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \sum_{j=1}^{p} |\beta j| \tag{4}$$

p is the number of predictors. λ is also known as $L1$, which refers to the regularization parameter controlling the shrinking degree. n is the number of observations. $|\beta_j|$ is the absolute value of the coefficient.

A drop in accuracy was noticed because the high λ resulted in an overfitting in the lasso model using BHD. In some experiments, however, under-fitting could occur because of missing significant features. Therefore, further experiments are required for better accuracy for the best model to be applied to the BHD.

3.5. Ridge regression

Ridge regression modifies linear regression models by adding regularizing terms to stop the overfitting issues [13]. Because it reduces the influence of correlated features on coefficient estimates, it enhances the stability of the model and is especially helpful when handling multi-collinearity or highly correlated features.

Ridge regression can reduce the impact of less relevant features by reducing their coefficients closer to zero. It selects the optimal value for finding the *L2* regularization that balances model complexity and performance as described in Eq. (5).

$$L(\beta) = \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \sum_{j=1}^{p} \beta_j^2 \tag{5}$$

Unlike L_1 regularization in Lasso Regression as a penalty term of the loss function, L_2, i.e. β_j^2 term, reduces the coefficients while maintaining their inclusion in the model. Ridge regression lowers variance and increases model stability, especially with multicollinearity. Ridge regression showed better accuracy than lasso but not the best in other experiments. Therefore, further research is required to fill the gap of stat-of-the-are.

3.6. Random forest regression

Random forest regression is an ensemble learning approach for regression applications [16]. It builds several decision trees during the training process. It produces the average prediction of all the individual trees to manage complicated datasets with high dimensionality and nonlinear correlations. It divides the feature space into areas recursively, giving each zone a constant value to reduce overfitting and decorrelating the different trees. The average of all the individual trees' forecasts makes up the prediction of a random forest regression model, as shown in Eq. (6).

$$\bar{Y} = \frac{1}{T} \sum_{t=1}^{n} h_t(X) \tag{6}$$

Where $\bar{Y}$ is the predicted value, T is the total number of trees. $H_t(X)$ is the prediction of the *t-th* tree.

In this research, random trees improved the predictive accuracy by controlling overfitting compared to previously examined approaches. This model has several benefits, such as better generalization, robustness to outliers, and parallelization training individual trees inside the forest.

3.7. Deep learning with keras algorithm

This study uses the Keras library [11] to apply Neural Network regression. Keras's model usually includes one input layer with one or more hidden layers to incorporate the regression process. In the implemented Keras on BHD, medv was the target variable. The input layer contained 128 neurons, and the first input layer contained 64 neurons and ReLU activation. The model continues with a multilayer perceptron (MLP) design for one hidden layer followed by two hidden layers. This design lets the model learn complex, nonlinear relationships between the input features and the target variable. Values shown in Eq. (7).

$$\bar{Y} = f(x) = W_2 \cdot ReLU(W_1 \cdot x + b_1) + b_2 \tag{7}$$

W_1 and W_2W are the weights addressed to connections between layers. b_1 and b_2 are bias vectors that allow the model to fit better. ReLU stands for Rectified Linear Unit, which activates the functions applied to assist hidden to deal with non-linearity. The dropout is adjusted to (0.2) between the hidden layers. This mechanism encourages the model to not depend strongly on a particular feature during the training to promote generalization as deep learning. The applied Keras resulted in reasonable accuracy performance but indicated that it is not the best model to predict the continuous values of the BHD. Further, research is conducted to achieve the aim of this study.

3.8. Gradient boosting regression

Gradient boosting regression is a powerful ML method that has gained widespread popularity in predictive modelling. It handles complex relationships in data and produces highly accurate predictions [12]. This regressor is an ensemble learning method that improves predictions by successively fitting numerous weak learners. It uses decision trees to create an additive model, as illustrated in Eq. (8).

$$\bar{Y} = \sum_{m=1}^{M} \gamma_m h_m(x_1) \tag{8}$$

$\bar{Y}$ represents the predicted values of the iteration *i-th*. M is the total number of the trees. γ_m is the weight applied to trees. $h_m(x_i)$ is the prediction of the trees for the required observation.

In this research, gradient-boosting regression scored with the best accuracy because weak learners were added one after the other; this reduced the residual errors from the previous step until a strong predictor was created. This made the model perform better compared to all experiments in this research. The learning rate of the shrinking technique prevents overfitting for further enhancement. In addition, it helps in feature selection and model interpretability.

3.9. Improving gradient boost regressor (re-sampling & optimization)

The experiments of this study have proved that the Gradient Boosting algorithm has achieved higher accuracy on the BHD when compared to the state-of-the-art. The proposed model suggests adding the re-sampling and optimization techniques to the classic gradient algorithm to improve the accuracy.

First, the SMOTETOMEK technique is applied to balance the sampling of the dataset. SMOTE produces adequate samples of the minority class, whereas TOMEK eliminates the nearest neighbors of the borderline to balance and clean the dataset, as shown in Eq. (9) [17].

$$x_{new} = x_i + \delta \cdot (x_{nn} - x_i) \tag{9}$$

x_{new} represents the new samples, while x_i is the minority class. x_{nn} as nearest neighbor subtracts the minority class of x_i. δ denotes the random number of distributions between zero and one.

Second, RandomizedSearchCV is conducted to tune the hyper-parameters. This technique samples a specific number of parameters randomly. It sets the specified distributions and assesses them through validation, as presented in Eq. (10).

$$\hat{\Theta} = argmin_{\theta \epsilon \Theta} \frac{1}{k} \sum_{k=1}^{K} L(y^K, f(X^K,)) \tag{10}$$

Integrating the two methods, SMOTETOMEK and RandomizedSearchCV, into a gradient-boosting algorithm improved the performance. The former in Eq. (9), as a preprocessing to the dataset, enhances the quality of the training to obtain a reliable model. The latter, in Eq. (10), is a tuning parameter that guarantees the tuning of the model optimally for better generalization, which improves accuracy. The following steps describe the new proposed model applied to the BHD.

Proposed algorithm.

Step 1:	Call libraries needed.
Step 2:	Load the dataset.
Step 3:	Make an optional Skewed Target Variable.
Step 4:	Divide the dataset into training sets and testing sets.
Step 5:	Re-sample training data using Tomek and SMOTE.
Step 6:	Set the GradientBoostingRegressor model's initialization.
Step 7:	Establish the RandomizedSearchCV parameter grid.
Step 8:	Conduct a random search.
Step 9:	Fit the model.
Step 10:	Predict the evaluation set.
Step 11:	Model evaluation.
Step 12:	Cross-Validation

The above steps describe a regression analysis approach that uses hyper-parameter optimization and re-sampling to address the high accuracy of the BHD.

Step 1 is to import libraries for data manipulation, such as sci-kit-learn and data handling tools.

Step 2 loads the regression dataset for analysis.

Step 3 to develop a skewed target variable (for testing objectives). This step is mainly utilized in experiments by purposefully distorting the target of the variable's distribution.

Step 4 Dividing dataset into testing and training: a random split function is used to split the dataset into training and testing sets. This division ensures the model is tested on unseen data (testing set) and train data (training set).

Step 5 was tested but did not achieve the highest accuracy. It starts by re-sampling training data to rebuild for imbalance: the training data's class imbalance is addressed by applying the SMOTETomek re-sampling technique. This strategy involves two approaches: SMOTE (Synthetic Minority Oversampling Technique): by producing artificial data points for the minority class, this technique corrects the imbalance. A re-sampled training dataset with a more balanced class distribution is created by using SMOTETomek. This step may enhance the performance in unbalanced regression issues.

Step 6, the GradientBoostingRegressor model is instantiated. This model is widely preferred for regression problems because of its versatility and capability to handle nonlinear correlations between data and the target variable.

In Step 7, the RandomizedSearchCV parameter grid is created to optimize the model's performance by adjusting hyperparameters. In this phase, a grid is designed to determine each hyper-parameter's potential values that must be adjusted. This grid defines the boundaries of the search space for the optimization technique.

Step 8 incorporates RandomizedSearchCV to determine the most optimal hyper-parameter configuration. This method efficiently analyses the specified parameter grid by randomly selecting a subset of hyper-parameter combinations and assessing their efficacy. The technique identifies the combination that produces the optimal performance on a validation set, a subset of the training data utilized for adjusting hyper-parameters.

Step 9 involves fitting the model using the hyper-parameters determined by the RandomizedSearchCV algorithm on the re-sampled training data, if applicable.

In step 10, the model is applied to the previously unseen testing data from step 4 to provide predictions. This step allows the regressor to assess the model's ability to make accurate predictions.

In Step 11, the model's efficacy is assessed using metrics appropriate for regression tasks. R^2 and (MSE, MAE) are examples of standard metrics. These metrics provide information on the accuracy and goodness-of-fit of the model by quantifying the gap between the predicted values and the actual target values.

Step 12. K-fold or stratified k-fold cross-validation is applied using several random data splits, and this technique iteratively repeats stages 4 through 11 of the process. Unlike a single split technique, each iteration offers an independent assessment of the model's performance, resulting in a more reliable and generalizable evaluation.

To improve the accuracy of the Gradient Boost Regressor, a re-sampling method of SMOTETomek is used in Step 5. It generates synthetic samples for the minority class and removes instances close to the majority class. The first high accuracy is reached 0.92.

The second fundamental part of the procedure in step 6 is using RandomizedSearchCV to carry out a thorough hyperparameter optimization. A predetermined grid of hyper-parameters, including the number of estimators, maximum depth, learning rate, subsample ratio, minimum samples needed for a split, minimum samples required for a leaf, and maximum features considered for a split, are searched across by this method. The search type can find the ideal hyper-parameters through hundreds of iterations to tune the model with the perfect configuration. The performance of this strategy is evaluated by calculating the (MSE), (MAE), and (R^2). This approach has shown a noticeable enhancement in predicting the accuracy, resulting in better accuracy.

3.10. Computational complexity of the proposed model

To evaluate the proposed model practically, it is important to highlight the computational complexity of the gradient boosting and RandomizedSearchCV [18].

Table 1. Computational complexity.

Regression model	Time	Memory usage	Notes
Linear	~ 0.2 second	~ 20 MB	Fast for small datasets.
Ridge	~ 0.2–0.3 seconds	~ 22 MB	Similar to Linear Regression
Lasso	~ 1 second	~ 30 MB	Additional time for feature selection.
Polynomial	~ 2 seconds	~ 50 MB	Increases complexity with coefficients degree.
Random forest	~ 5 seconds	~ 100 MB	Scales well; robust but slower.
Keras	~ 15 seconds	~ 250 MB	Need more training; computationally intensive.
Gradient boosting	~ 10 seconds	~ 150 MB	Accurate but slower for deep trees.
Proposed model	~ 11 seconds	~ 160 MB	Best accurate performance

First, the gradient boosting technique depends on the number of trees, samples and depth of each tree i.e. (M, N and d) as shown in Eq. (11).

$$O(M.N.d.logN) \tag{11}$$

Second, the process of hyperparameter optimization utilizing RandomizedSearchCV includes executing T as a random iteration across a parameter grid, in conjunction with a V which is a fold cross validation as shown in Eq. (12).

$$O(T \cdot V \cdot M \cdot N \cdot d \cdot logN) \tag{12}$$

Table 1 shows the computational efficiency of the tested models using standard hardware of a core i7 processor and 16 GB RAM. First of all, Linear and Ridge regressions are the fastest with approximately of ~0.2 seconds and ~(20–22) MB. Lasso and Polynomial take slightly longer times ~1 and ~2 seconds with ~30 and ~50 MBs. Random Forest takes higher time of ~5 seconds with moderate memory usage of ~100 MB. In contrast, Keras is the most intensive i.e. ~ 15 seconds because of the complexity of the Neural Networks. Finally, Gradient Boosting and the proposed Model present high accuracy as an advantage but require ~ (10–11) seconds and ~ (150–160) MB which is suitable for capturing complex patterns of the used dataset.

3.11. Evaluation and performance metrics

This research uses three metrics to examine the best model performance. i.e. MSE, MAE and R^2. The metrics used are used to evaluate the performance of the examined models. MSE calculates the average squared difference between observed and predicted values, which provides information on the variance of prediction errors. Conversely, MAE computes the average absolute distinctions between actual and predicted values, providing a simple description of prediction accuracy. R-squared measures the percentage of variance in the dependent variable compared to the independent ones. It is usually considered a model with a high R^2 value) and low error metrics (a low MSE and MAE) should be viewed as a better performance when compared to other models.

4. Results

The suggested Gradient Boosting framework showed better performance across all evaluated regression models. Fig. 2 illustrates the correlation analysis, highlighting the relationships among important attributes in the (BHD), hence signifying the dataset's necessity for sophisticated modeling approaches. Table 2 summarizes the results of each model, including a comprehensive comparison of R^2, MSE, and MAE.

Linear regression is evaluated as a baseline, scoring a R^2 of 74.9, an MSE of 0.09, and an MAE of 0.02. This score signifies a limited capacity to handle the dataset's nonlinear associations. Ridge Regression enhanced this performance, achieving a R^2 of 88.3. By using cross-validation in Ridge Regression, the optimal selection of regularization parameters was ensured, refining the model's predictive performance and bolstering its ability to generalize well to unseen data. However, its prediction accuracy remained lower compared to that of ensemble models. Likewise, Lasso Regression produced a R^2 of 65.0, indicating its propensity to underfit the data as a result of stringent feature selection as shown in Table 2. It became apparent the necessity to experiment with other regressors to obtain better results.

Multiple Linear Regression was tested as the second model, providing a slightly better score of R^2 76. as shown in Fig. 3. The model demonstrated an improved predictive capability by including more features but it did not the achieve target of this research.

Polynomial Regression proved competence in modeling nonlinear relationships, with R^2 of 83.0; yet, the computational complexity escalated significantly with higher polynomial degrees. Ensemble approaches such as Random Forest achieved notable enhancements, achieving R^2 of 89.9, displaying

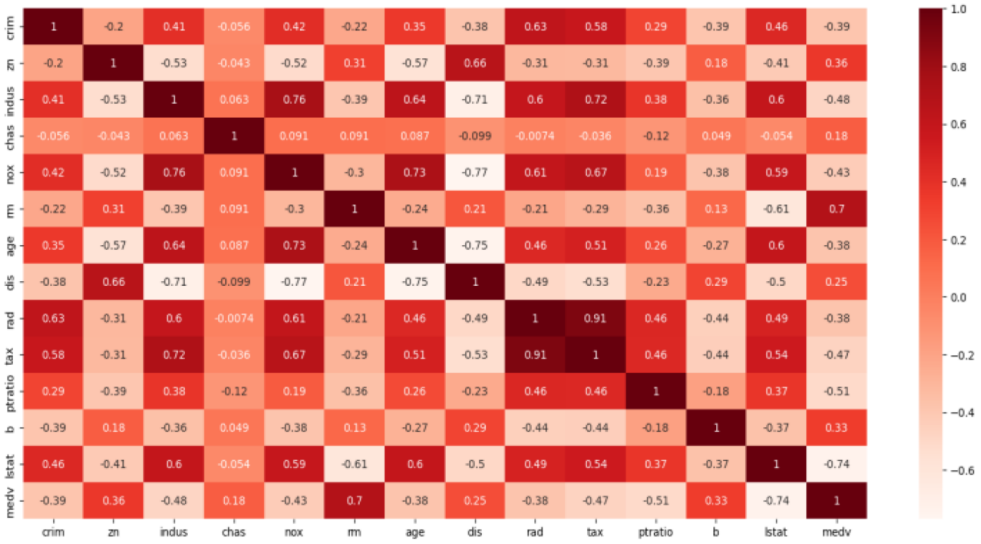

Fig. 2. Correlation of each feature.

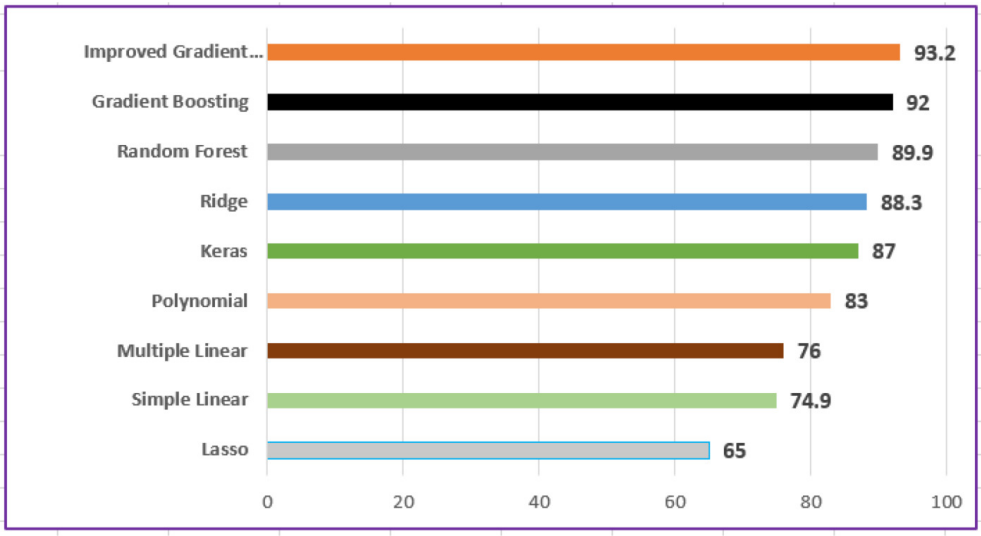

Fig. 3. Model performance.

resilience and adaptability. Nonetheless, the approach encountered difficulties with overfitting as the number of trees increased.

The Keras deep learning model, employing a neural network with three layers, scored R^2 of 87.0 but required consideration for processing resources. In addition, an extended training duration of around 19 seconds for the BHD. Although better performance of Keras model but did not provide a considerable advantage compared to classic approaches.

The suggested Gradient Boosting framework produced superior performance in general, illustrated by a R^2 of 93.2, an MSE of 0.015, and an MAE of 0.82. These scores demonstrate a significant improvement compared to all other models, as illustrated in Fig. 3. The enhanced performance is because of the inclusion of SMOTETomek resampling, which effectively addressed data imbalance. In addition, RandomizedSearchCV hyperparameter optimization is used, which refined the model variables for best accuracy.

5. Discussion

The performance of several regression models assessed by different authors provides a better understanding of each model. For this evaluation, the (R^2), (MSE), and (MAE) metrics are used, as shown in Table 2.

Table 2. Comparison of regressors.

Authors	Regression	R^2	MSE	MAE	Dataset
[6]	XGBoost	85.799520	0.628	2.936	Boston
	Random forest	81.971735	0.81	1.348	
	Simple linear	71.218184	3.090	19.074	
	SVM	59.001585	0.0001	0.009	
[8]	Random Forest	91.3	0.49	1.115	Boston
[19]	Ridge	69	17.882	2.793	Boston
	GA-RF model	91	3.599	1.196	
[10]	Simple linear	74.66	19.07	3.09	Boston
	Random forest	86.41	2.55	0.94	
	SVM	59.00	26.95	2.94	
[3]	Simple linear	73.66			Boston
	Polynomial	74.27			
	Ridge	88.28			
	Lasso	88.79			
[9]	Random forest	90	0.702	1.900	Boston
[7]	Simple linear	91	0.017	0.075	Boston
	Multilayer	64	0.066	0.179	
	Random forest	86	0.025	0.112	
	SVM	57	0.079	0.211	
	XGBoost	92	0.015	0.84	
The experiments of current study	Simple linear	74.9	0.09	0.02	Boston
	Multiple linear	76	2.50	10.0	
	Polynomial	83	5.50	1.80	
	Lasso	65	10.5	2.61	
	Ridge	88.3	5.01	1.71	
	Random forest	89.9	3.51	1.21	
	Keras	87	1.99	2.74	
	Gradient boosting	92	0.72	2.00	
	Improved gradient boosting	93.2	0.015	0.82	

[6] stated that XGBoost scores robust performance on the BHD, as demonstrated by its low MSE of 0.628, MAE of 2.936, and R^2 of 85.8%. With 82%, MSE of 0.81, and MAE of 1.348, Random Forest comes second with predictive solid accuracy. The superiority of ensemble methods in this situation is demonstrated by the poorer performance of linear regression and SVM, which noticeably have more errors.

[8] stated that the Random Forest model scores an astounding 91.3%, with an MSE of 0.49 and an MAE of 1.115. The strong R^2 indicates excellent model fit, and the comparatively low error metrics further support its effectiveness in handling this dataset.

[19] claimed that the Ridge Regression and GA-RF Model produce different outcomes for the Boston dataset. The performance of Ridge Regression is indicated by its R^2 of 69%, MSE of 17.882, and MAE of 2.793. With an of 91%, MSE of 3.599, and MAE of 1.196, on the other hand, the GA-RF Model performs remarkably well, highlighting the value of integrating genetic algorithms and random forests.

The outcomes in [10] highlighted the benefits of ensemble methods once more. With an 86.41%, Random Forest performs noticeably better than SVM-Regressor 59% and Linear Regression 74.66%. The predictive accuracy of Random Forest is further demonstrated by its reduced MSE and MAE (2.55 and 0.94, respectively).

The only numbers provided by the [3] are those for Ridge Regression (88.28%) and Lasso Regression (88.79%), which are considered to be strong competitors. Linear and Polynomial Regression have comparable, if somewhat lower, values of 73.66% and 74.27%.

[9] argued that Random Forests of 90%, MSE of 0.702, and MAE of 1.900 confirm its reliable performance in many investigations for the BHD.

[7] stated that the 91% and 92% scores, respectively, Linear Regression and XGBoost stand out among the several models this study examines using the Boston dataset. Their low MAEs (0.075 and 0.84) and MSEs (0.017 and 0.015) indicate their excellent predictive performance. Lower values are shown in Multilayer Perceptron, Random Forest, and SVM-Regressor, indicating less predictive accuracy.

The contrast in Table 2 shows that the results of R^2 of SVM are similar to those reported by the [6, 10] and [7]. Therefore, SVM was excluded from the proposed model because of the low level of R^2. [6, 10] and [3] share around 75% of R^2, similar to the proposed model. However, no significant increase was detected in the proposed model because the proposed

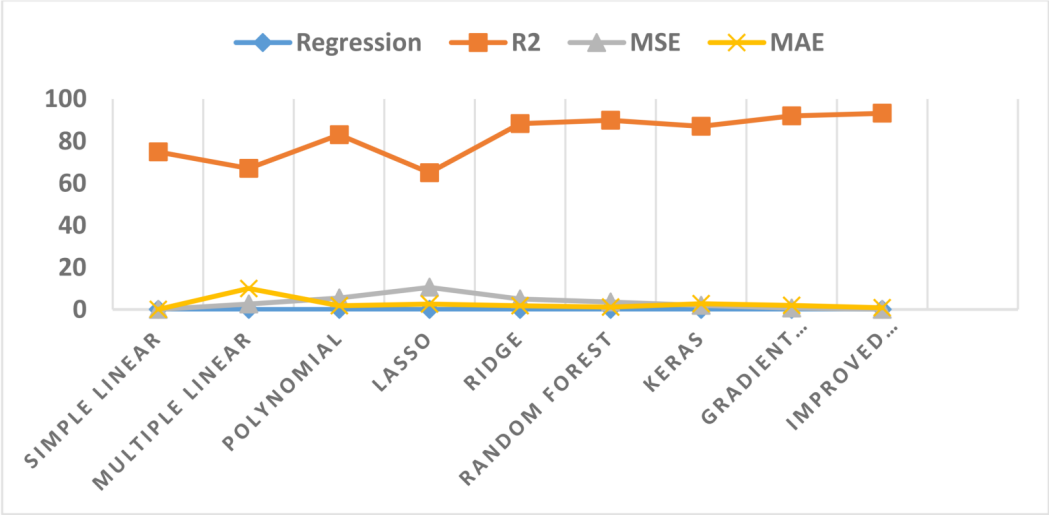

Fig. 4. Comparison of model performance metrics.

model scored R^2 of 74.9. Further analysis shows that the R^2 in [6] and [10] are similar to those reported in [7]. The random forest reveals that (80-86) % of R^2 shows promising results. However, in [8] and [9], reassuring results encouraged the author in this research to obtain a reasonable level of R^2, i.e. 89.9. Both results in [19] and [3] reveal that the R^2in Ridge regression scores of (69 and 88.28) respectively, whereas this research registered an increase of 88.3 with better performance. No rise of R^2 was detected in this research of Lasso, while [3] obtained notable performance. However, the experiments of the Polynomial model of the current study show a better level of R^2 83 when compared to [3].

Compared to the current research results in [7] and [6], Gradient Boosting outperforms the score of R^2 in XGBoost, as shown in Table 2. The R^2 of 0.92 in the Gradient Boosting technique using SMOTETomek scored a competitive level of accuracy. With optimization technique, the best obtaining of the greatest R^2 0.932%, lowest MSE (0.015) and MAE (0.82) as shown in Fig. 4. The comparison of the proposed model with previous sources highlights the benefit and originality of the current research. In addition, the results provide the importance of the Gradient Boosting model over the tested model in the current suggested framework. The improved Gradient Boosting appeared as the best solution due to its balance of accuracy and computational efficiency. Despite consuming ~11 seconds and ~160 MB of RAM, it beat other models by efficiently capturing complicated patterns. Thus, it is perfect for achieving great accuracy while requiring minimal resources. The proposed approach can be applied in real-world scenarios. It can be utilized in housing price prediction, and financial analysis, where accurate and reliable predictions are important for decision-making and data mining. The framework is suitable for applications that need precision without extra computational cost.

6. Conclusion

The ability of different regression approaches to accurately predict continuous values differs significantly, as can be seen by comparing them. Gradient Boosting performed better than all the other tested models, especially after optimization, with the lowest MSE of 0.015, the highest R^2 score of 93.2, and the lowest MAE of 0.82. It shows how well Gradient Boosting handles complex data patterns and generates accurate predictions. Random Forest and Ridge Regression also showed an outstanding performance, demonstrating that these models are appropriate for tasks requiring high prediction accuracy. However, the effectiveness of Lasso Regression and Linear Regression was comparatively lower, highlighting the necessity for more advanced techniques in specific situations.

The findings highlight the necessity of selecting and optimizing appropriate regression algorithms to improve the accuracy of continuous value predictions, providing functional visions for future research and application in various domains.

Further research will examine the Early Stopping approach in the training process to reduce the errors in the validation and prevent overfitting. In addition,

more advanced regression methods will be experimented with using different datasets to enhance the accuracy.

Funding

None

Acknowledgement

None

Conflicts of interest

The author declares no conflict of interest.

References

1. B. Dou, *et al.*, "Machine learning methods for small data challenges in molecular science," *Chemical Reviews*, vol. 123, no. 13, pp. 8736–8780, 2023.
2. A. Kulkarni, D. Chong, and F. A. Batarseh, "*5 - Foundations of data imbalance and solutions for a data democracy*," F. A. Batarseh, and R. B. T.-D. D. Yang, Eds., Academic Press, pp. 83–106, 2020. doi: 10.1016/B978-0-12-818366-3.00005-8.
3. S. Sanyal, S. K. Biswas, D. Das, M. Chakraborty, and B. Purkayastha, "Boston house price prediction using regression models," in *2022 2nd International Conference on Intelligent Technologies (CONIT)*, IEEE, pp. 1–6, 2022.
4. G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, "Linear regression," in *An introduction to statistical learning: With applications in python*, Springer, pp. 69–134, 2023.
5. L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "Polylanenet: Lane estimation via deep polynomial regression," in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, pp. 6150–6156, 2021.
6. Y. Chen, "Research on the prediction of Boston house price based on linear regression, random rorest, Xgboost and SVM models," vol. 21, pp. 27–37, 2023.
7. H. Sharma, H. Harsora, and B. Ogunleye, "An optimal house price prediction algorithm: XGBoost," *Analytics*, vol. 3, no. 1, pp. 30–45, 2024.
8. S. Sharma, D. Arora, G. Shankar, P. Sharma, and V. Motwani, "House price prediction using machine learning algorithm," in *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 982–986, 2023. doi: 10.1109/ICCMC56507.2023.10084197.
9. A. B. Adetunji, O. N. Akande, F. A. Ajala, O. Oyewo, Y. F. Akande, and G. Oluwadara, "House price prediction using random forest machine learning technique," *Procedia Computer Science*, vol. 199, pp. 806–813, 2022.
10. Z. Li, "Boston house price prediction based on machine leaning methods," *BCP Business & Management*, vol. 38, pp. 2883–2887, 2023. doi: 10.54691/bcpbm.v38i.4204.
11. J. Ott, M. Pritchard, N. Best, E. Linstead, M. Curcic, and P. Baldi, "A fortran-keras deep learning bridge for scientific computing," *Scientific Programming*, vol. 2020, p. 8888811, 2020. doi: 10.1155/2020/8888811.
12. J. T. Vieira, R. B. D. Pereira, C. H. Lauro, L. C. Brandão, and J. R. Ferreira, "Multi-objective evolutionary optimization of extreme gradient boosting regression models of the internal turning of PEEK tubes," *Expert Systems with Applications*, vol. 238, p. 122372, 2024.
13. B. M. Kibria and A. F. Lukman, "A new ridge-type estimator for the linear regression model: Simulations and applications," *Scientifica*, vol. 2020, 2020.
14. J. H. Lee, Z. Shi, and Z. Gao, "On LASSO for predictive regression," *Journal of Econometrics*, vol. 229, no. 2, pp. 322–349, 2022.
15. Q. Zhang, "Housing price prediction based on multiple linear regression," *Scientific Programming*, vol. 2021, p. 7678931, 2021. doi: 10.1155/2021/7678931.
16. D. A. Zema, M. Parhizkar, P. A. Plaza-Alvarez, X. Xu, and M. E. Lucas-Borja, "Using random forest and multiple-regression models to predict changes in surface runoff and soil erosion after prescribed fire," *Modeling Earth Systems and Environment*, vol. 10, no. 1, pp. 1215–1228, 2024.
17. K. Nugroho, A. R. Muslikh, S. W. Irianandа, and A. A. Ojugo, "Integrating SMOTE-Tomek and fusion learning with XGBoost Meta-learner for robust diabetes recognition," *Journal of Future Artificial Intelligence and Technologies*, vol. 1, no. 1, 2024.
18. N. Sakib, T. Paul, N. Anwari, and M.d. Hadiuzzaman, "Ensemble-based model to investigate factors influencing road crash fatality for imbalanced data," *Transportation Engineering*, vol. 18, p. 100284, 2024. doi: 10.1016/j.treng.2024.100284.
19. L. Ye, "Comparison of ridge regression and GA-RF models for Boston house price prediction," *International Journal of Mathematics and Systems Science*, vol. 6, no. 4, 2023.