EnPress Publisher

Comparison of Ridge Regression and GA-RF Models for Boston House Price Prediction

Liang Ye
Jiangnan University, Wuxi 214000, China.

Abstract: The purpose of this paper is to explore the performance of ridge regression and the random forest model improved by genetic algorithm in predicting the Boston house price data set and conduct a comparative analysis. To achieve it, the data is divided into training set and test set according to the ratio of 70-30. The RidgeCV library is used to select the best regularization parameter for the Ridge regression model, and for the random forest model, the genetic algorithm is used to optimize the model's hyperparameters. The result shows that compared with ridge regression, the random forest model improved by genetic algorithm can perform better in the regression problem of Boston house prices.

Keywords: Ridge Regression; Random Forest; Genetic Algorithm; Model Comparison

1. Introduction

1.1 Problem Description

In reality, the house price of an area is influenced by a combination of many factors, including location, traffic conditions, distance from the city center, school district housing, etc. Therefore, a comprehensive analysis of the surrounding environment and an accurate prediction of local house prices are of great importance in the purchase of houses by residents and the selection of sites for real estate construction. This paper aims to analyze 13 key factors influencing house prices with the Boston-house price dataset, and build a ridge regression model and a random forest model to explore the role of different influencing factors on local house prices.

1.2 Model Frame Diagram

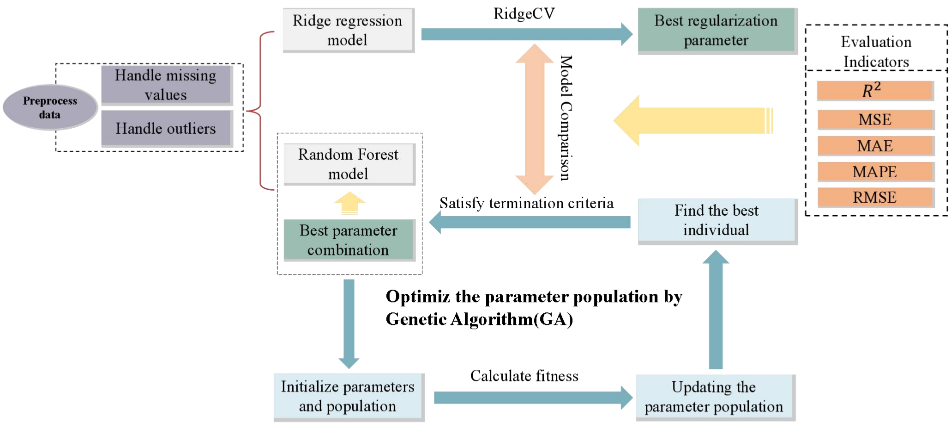
Preprocess data
Handle missing values
Handle outliers
Ridge regression model
RidgeCV
Best regularization parameter
Evaluation Indicators
R^2
MSE
MAE
MAPE
RMSE
Model Comparison
Random Forest model
Best parameter combination
Satisfy termination criteria
Find the best individual
Optimiz the parameter population by Genetic Algorithm(GA)
Initialize parameters and population
Calculate fitness
Updating the parameter population

Fig 1. Model Framework

2. Data Description

The Boston house price dataset is a classical machine learning dataset, which consists of 506 samples, 13 independent variable features and 1 target house price feature. As shown in Table 1 and Fig 2, CHAS and RAD are categorical, while others are continuous.

Some characteristics have missing values, and only a few variables follow a normal distribution. Data distribution is uneven or concentrated in certain ranges. Hence, further analysis requires handling outliers and appropriate normalization to ensure modeling process applicability.

Table 2. Statistical Table

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
count	486	486	486	486	506	506	486	506	506	506	506	506	486	506
mean	3.61	11.21	11.08	0.07	0.55	6.28	68.52	3.80	9.55	408.24	18.46	356.67	12.72	22.53
std	8.72	23.39	6.84	0.26	0.12	0.70	28.00	2.11	8.71	168.54	2.16	91.29	7.16	9.20
min	0.01	0.00	0.46	0.00	0.39	3.56	2.90	1.13	1.00	187.00	12.60	0.32	1.73	5.00
25%	0.08	0.00	5.19	0.00	0.45	5.89	45.18	2.10	4.00	279.00	17.40	375.38	7.13	17.03
50%	0.25	0.00	9.69	0.00	0.54	6.21	76.80	3.21	5.00	330.00	19.05	391.44	11.43	21.20
75%	3.56	12.50	18.10	0.00	0.62	6.62	93.98	5.19	24.00	666.00	20.20	396.23	16.96	25.00
max	88.98	100.00	27.74	1.00	0.87	8.78	100.00	12.13	24.00	711.00	22.00	396.90	37.97	50.00

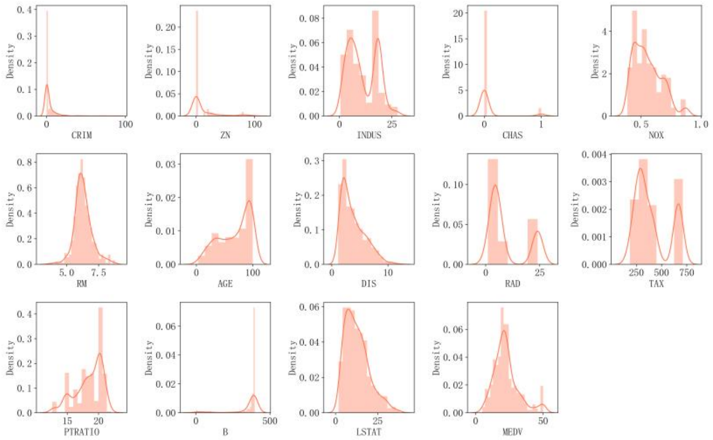

Fig 2. Histogram and Kernel Density Plot

3. Evaluation Indicators

In order to evaluate the accuracy of the prediction results and to compare the predictive power of each model, this paper selects five indicators, namely the coefficient of determination (R^2), the mean square logarithmic error (MSLE), the mean absolute error (MAE), the mean absolute percentage error (MAPE) and the root mean square error (RMSE). The calculation formulas are as follows:

$$R^2(y,\hat{y}) = 1 - \frac{\Sigma_{i=1}^{n}(y_i - \hat{y_i})^2}{\Sigma_{i=1}^{n}(y_i - \bar{y_i})^2} \#(1)$$

$$MSLE(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}\left(log(1+y_i) - log(1+\hat{y}_i)\right)^2 \#(2)$$

$$/\, MAE(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y_i} - y_i\right| /(3)$$

$$/\, MAPE(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\widehat{y_i} - y_i}{y_i}\right| /(4)$$

$$/\, RMSE(y,\hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\widehat{y_i} - y_i)^2} /(5)$$

4. Data Preprocessing

4.1 Handling Missing Values

By drawing the missing value distribution map (as shown in **Fig 4**), it is found that there are missing values in the features CRIM, ZN, INDUS, CHAS, AGE and LSTAT, and the distribution positions of these missing values are inconsistent. If these missing values are simply deleted, a large amount of original information may be lost, thus affecting the accuracy and credibility of subsequent analysis. Therefore, in this paper, the data imputation method is used to deal with the missing values of each feature.

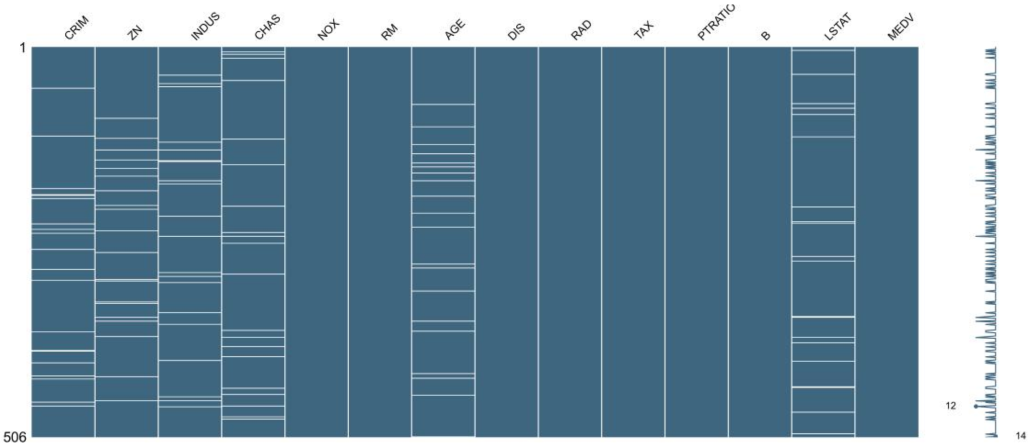

Fig 3. Missing Value Matrix Plot

Data imputation is a common approach for handling missing values, which involves inferring the missing data values using a prescribed method and then inserting them into the original data. Specifically, in the context of the presented **Fig 4**, the CHAS feature is discrete and has a concentrated distribution, making the mode an appropriate choice for filling in missing values.

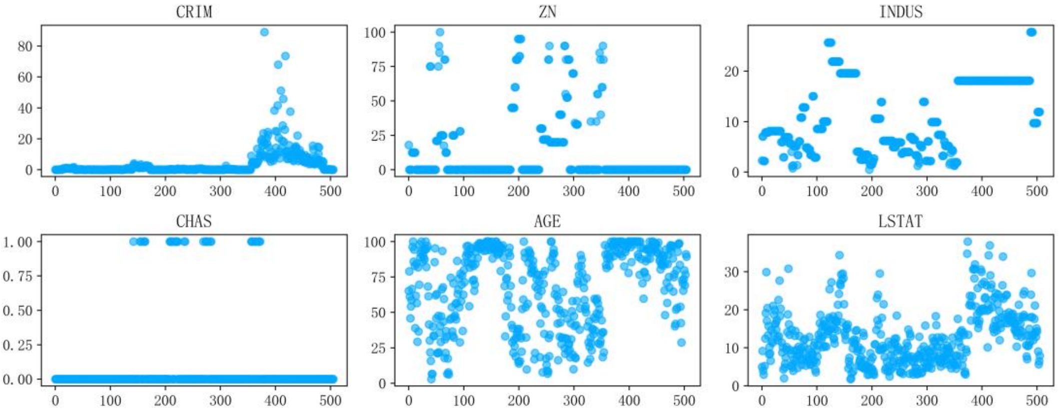

Fig 4. Feature Scatterplot with Missing Values

Missing values for continuous-valued features CRIM, ZN, and INDUS, with concentrated distributions, can be imputed using the mean value before and after them. For AGE and LSTAT, which have relatively uniform distributions, missing values can be filled with the overall mean feature value. This data imputation is a well-established technique in data analysis and machine learning, relying on accurate imputation methods and dataset characteristics.

4.2 Outlier Treatment

Fig 5 shows outliers in the independent variable features CRIM, ZN, RM, DIS, PTRATIO, B, and LSTAT. To preserve original information and maintain analysis reliability, this study focuses on highly correlated features with the target housing price. Pearson correlation coefficients are calculated for each feature to identify the ones with the most significant impact on the target price, effectively reducing outlier influence with minimal information loss.

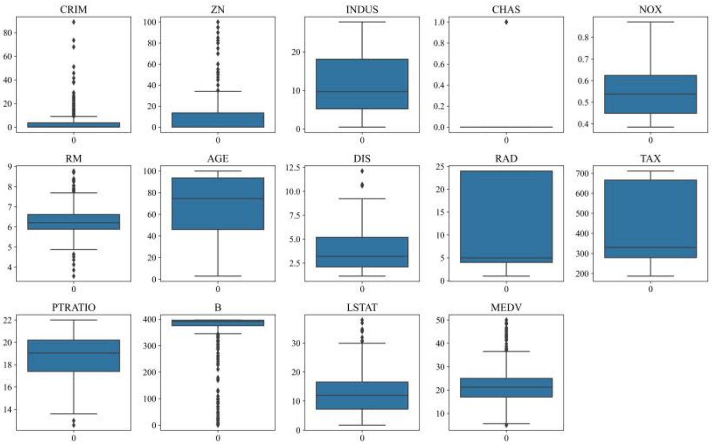

Fig 5. Boxplots of each feature

The Pearson correlation coefficient is widely used to calculate the degree of correlation between two variables. The coefficient was proposed by Carl Pearson in 1895 [1] , and its calculation formula is as follows:

$$/r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \#(6)/$$

where n is the number of samples, x_i and y_i are values of x and y for the i-th sample, and $\bar{x}$ and $\bar{y}$ represent the mean of x and y, respectively. The coefficient ranges from -1 to 1, with -1 indicating complete negative correlation, 1 denoting complete positive correlation, and 0 indicating no correlation. Larger absolute values signify stronger feature correlations. **Table 2** shows the Pearson correlation coefficients between each variable and the target housing price obtained through calculations. LSTAT and RM are found to have a higher correlation with housing prices. Consequently, this study chooses to delete samples with outliers in LSTAT and RM, resulting in retaining 468 samples.

Table 2. The Pearson correlation coefficient value of each feature to house price

Features	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
Corr-coef	-0.39	0.37	-0.48	0.18	-0.43	0.70	-0.38	0.25	-0.38	-0.47	-0.51	0.33	-0.72

5. Model Building

5.1 Ridge Regression Model

After analyzing the characteristic correlation of the Boston house price data set and drawing it into a heat map (as shown in **Fig 6**), it is found that there is a high correlation between some variables. This correlation may lead to approximate multicollinearity between variables, which affects the stability and accuracy of the linear regression model and reduces the interpretability of the model.

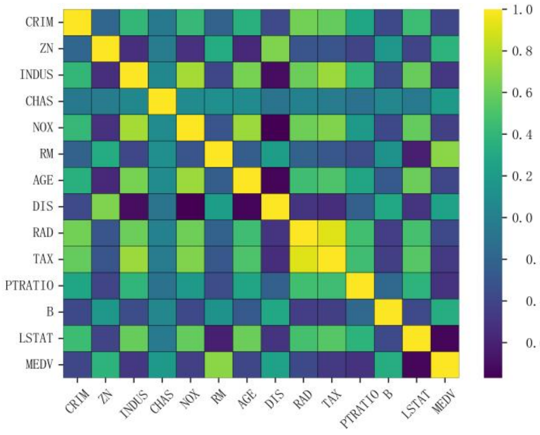

Fig 6. Correlation coefficient heat map of each feature

This study adopts ridge regression, proposed by Robert Tibshirani [2], as the modeling method. Ridge regression is a regularization technique for linear regression, effectively handling multicollinearity by adding an L2 norm penalty term to the regression equation. This reduces unstable coefficients in the model and controls its complexity. The hyperparameter λ adjusts the penalty's size: larger λ means a smaller model complexity, and vice versa. Proper tuning of λ is crucial for optimal model performance.

$$/\min_{\beta_0,\beta_j}\frac{1}{2n}\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{i}^{p}\beta_j^2\,\#(7)/$$

In this model, the data is divided into 70% training set and 30% test set. In order to achieve the best effect of the model, the RidgeCV library that comes with Python is used to select the value of the regularization parameter, and finally the best regularization parameter is 17. Based on this, a ridge regression model is built using the training set. The model achieved a training set fit of 0.74 and a test set fit of 0.71, indicating better performance on the test set. Cross-validation was used to evaluate the model's generalization performance, resulting in an average fit of 0.69. The model evaluation indicators for ridge regression on each dataset are summarized in **Table 3**, demonstrating its strong generalization performance and accurate predictions for unknown data.

Table 3. Performance of ridge regression on different data sets

	Full Dataset	Training Set	Test Set
R^2	0.692	0.744	0.716
MSE	17.882	18.389	16.706
MAE	2.793	2.729	2.940
MAPE	0.137	0.139	0.134
RMSE	4.229	4.288	4.087

5.2 GA-RF Model

5.2.1 Random Forest

The Random Forest algorithm was proposed by Leo Breiman in 2001 [3], which is an ensemble learning algorithm based on Bagging that incorporates random attribute selection during the training process of decision trees. Unlike traditional decision trees that use all attributes at each node, random forest selects a random subset of k attributes from the set of attributes at each node of each decision tree, and determines the optimal attribute for node splitting based on this subset. At the same time, random forest training does not use all samples for modeling every time, but randomly selects N samples for training with replacement, and then builds a decision tree, and uses the selected samples for the root node of the decision tree [4].

The random attribute selection method increases diversity among decision trees, improving the random forest ensemble model. Utilizing the Bagging integration method, random forest mitigates overfitting risk and enhances model generalization. It performs data resampling, constructs multiple decision trees, and integrates their predictions for the final output value using the simple average method:/$\widehat{y}_i=\frac{1}{B}\sum_{i=1}^{B}f_b(x_i)\,\#(8)/$

where B represents the number of decision trees, x_i represents the feature set of each sample, $f_b(x_i)$ represents the output of sample i under the b decision tree, and $\widehat{y}_i$ is the final output

value of sample i.

5.2.2 Genetic Algorithm

Genetic Algorithm (GA), proposed by John Holland in 1975 [5], is an adaptive global optimization search algorithm inspired by the genetic and evolutionary process of organisms in nature. It transforms the optimization problem into a genotype optimization problem, iteratively optimizing the genotype by simulating natural genetic processes to find the optimal solution. Each individual is represented as a genotype, randomly initialized, and its fitness is evaluated by a fitness function. Through operations like crossover, mutation, and selection, individuals evolve, leading to the discovery of better solutions.

It can be used to solve various optimization problems, such as function optimization, combinatorial optimization, parameter optimization, etc. Several scholars have combined genetic algorithms with other models for the purpose of finding optimal parameters quickly [6,7].

5.2.3 GA-RF Model

Parameter tuning is a critical aspect in machine learning algorithms. The random forest algorithm contains a large number of parameters, such as n_estimators(the number of decision trees), max_depth(the depth of the tree), min_samples_split(the minimum number of split samples), and min_samples_leaf(the minimum number of samples of leaf nodes), which directly affect the performance of the model. What's more, as an ensemble algorithm, the speed of training the model can be much slower compared to a single learner. To solve the problem, heuristic algorithms are widely used in machine learning algorithms. This paper combines genetic algorithm with random forest to optimize the four important parameters, through selection, inheritance, and variation operations of the genetic algorithm to achieve the best performance of the model. The improved model can enhance the solution quality and speed without the need to exhaust all possible parameter combinations.

5.2.4 Training Model

The preprocessed data is divided into a 70% training set and a 30% test set. The GA-RF model is trained on the training set and evaluated using the test set. Cross-validation is also performed on different data sets to assess the model's generalization ability and robustness. The model achieves a 0.97 fit on the training set and 0.85 on the test set. **Table 4** presents the model evaluation indicators for GA-RF on each data set. Additionally, cross-validation results in a final fit of 0.83, demonstrating that the genetic algorithm-enhanced random forest model performs well in predicting Boston housing prices with good stability and reliability.

Table 4. Performance of Random Forest on different data sets

	Full Dataset	Training Set	Test Set
R^2	0.939	0.973	0.850
MSE	3.599	1.678	8.054
MAE	1.196	0.857	1.982
MAPE	0.060	0.044	0.099
RMSE	1.897	1.295	2.838

6. Model Comparison and Conclusion

Based on the results in **Fig 9**, it is possible to compare the fitting effect of random forest and ridge regression on the original data set and each model evaluation index (normalized). It is found that the random forest improved by genetic algorithm performs well in all aspects on the Boston house price regression, and its fitting effect is significantly better than that of the ridge regression model.

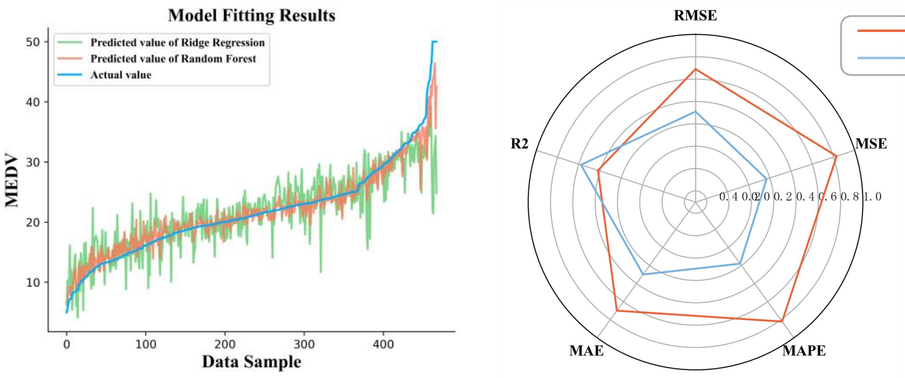

Fig 7. Model Comparison

So it can be concluded that although the Ridge Regression model can effectively fit linear data, real-world data is often non-linear. As an ensemble non-linear regression model, Random Forest performs better in handling such data compared to ordinary linear models. Therefore, when selecting models to fit real-world data, it is necessary to conduct a linear test to ensure the best model is utilized to achieve the optimal performance.

References

[1] Pearson K. (1895). "Notes on regression and inheritance in the case of two parents". Proceedings of the Royal Society of London. 58: 240–242.

[2] Hoerl AE., & Kennard RW. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55-67.

[3] Breiman L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

[4] Zhihua Z. Machine learning[M]: Beijing: Tsinghua University Press, 2016.

[5] Holland JH. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press.

[6] Khan MN, Ghafoor U, Abdullah A, et al. Prediction of thermal diffusivity of volcanic rocks using machine learning and genetic algorithm hybrid strategy[J]. International Journal of Thermal Sciences, 2023, 192: 108403.

[7] Huo ZG, Zha XT, Lu MY, Ma TQ, Lu ZC, Prediction of Carbon Emission of the Transportation Sector in Jiangsu Province-Regression Prediction Model Based on GA-SVM, Sustainability 15(4) (2023).

About the author: Liang Ye (2002), male, Han, Anhui, undergraduate, Jiangnan University, Wuxi, Jiangsu, 214000, research direction: Machine learning.