Proceedings of the 2nd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/75/20240526

House price prediction

Jiajin Kong

Auckland University of Technology, Auckland, 1010, New Zealand

12138oliver@gmail.com

Abstract. In recent years, more and more facts have proven that neural network models have good performance in machine learning, and researchers always use RNN and CNN models to solve complex problems, like how RNN can utilize historical information to predict further weather, identify human notes, and analyze text. The CNN model is always used to identify signatures, process picture images, and classify items. How I find that in the linear prediction field, people always use linear regression to predict results. For example, in house price prediction, people will use linear regression or the random forest model to predict house price. However, the author think neural network model can also have good performance in house prices. So, this paper is about building a better neural network model to predict house prices. This paper describes how to process data before training two models and compare their prediction accuracy. After that, this paper describes new neural network and train this model, which, after training, will output its accuracy. Luckily, after a long-time training, the author gets a neural network model to predict house price, it has a higher accuracy in predicting house price than two tradition models that we used.

Keywords: predict house price, neural network model, linear regression, random forest, accuracy.

1. Introduction

This paper is aiming to find a better model to solve the house price prediction problem. The linear regression model and the random forest model are the two most widely used models to solve the house price prediction model. Linear regression, with its simplicity and transparency, provides a baseline understanding of the linear relationships within the data. Meanwhile, the random forest, as an ensemble method, offers the flexibility to capture non-linear patterns and intricate interactions among features. So, this paper will compare those two models first and get an exact number about their accuracy. Then, the author will introduce a neural network model to achieve higher accuracy. Here are some practical implications for a better model. A housing price prediction model can provide buyers, sellers, and real estate professionals with information about the feature trends in house prices: Investors can use housing price forecasts to evaluate the feasibility of potential real estate investments. This helps them choose suitable investment opportunities and maximize their return on investment. Governments and urban planners can use housing price forecasts to formulate urban planning and housing policies. This helps promote sustainable urban development and community building.

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/75/20240526

2. Literature Review

Regression is a type of supervised learning algorithm used for predicting continuous numerical values based on input features. In particular, linear regression is a useful tool for predicting a quantitative response [1]. We always use linear regression to predict house price. Because it is easy to understand and suitable for simple relationships, it works well when there is a linear relationship between features and targets. The general single-equation linear regression model, which is the universal set containing simple (two-variable) regression and multiple regression as complementary subsets [2]. Multivariate linear regression and polynomial regression. The basic model for MLR is: $y = \beta 0 + \beta 1x1 + \cdots + \beta mxm + \varepsilon$. In this equation, $\beta 0$ is the intercept term, representing the expected value of the data when all independent variables are zero. $\beta 1, \beta 2, \ldots, \beta m$ are regression coefficients, indicating the extent of influence of each independent variable on the dependent variable. ε is the error term. The goal of the entire model is to adjust the regression coefficients to minimize the error term, thereby enabling the model to best fit the data. Multiple techniques have been developed to modify and improve the prediction accuracy of least squares regression models [3]. The model of a polynomial is: $y = \beta 0 + \beta 1x + \beta 2x2 + \cdots + \beta hxh + \varepsilon$. Normally, MLR can process data with a large number of variables, and polynomials can process data that does not conform to linear laws. In house price prediction, people always choose MLR because different features have different effects on the final price, and the relationship between features does not always confirm the linear rule. Here a model called linear regression in sklearn.linear_model Linear regression. This class is consisted of class definition and description, defining the constructor and attributes, the most important method called the fit method and the fitting process.

Random forests can capture nonlinear relationships and have good interpretability. On this basis, random forests improve stability and accuracy. A decision tree is a tree-structured model in which each leaf is associated with a binary-valued prediction and each internal node is a decision node associated with an attribute $a \in P$ and a threshold value $v \in R$ [4]. So we also use take this model into consider solving house price prediction. Probabilistic Random Forest (PRF) algorithm treats the features and labels as probability distribution functions, rather than deterministic quantities [5]. Here is a model in sklearn.ensemble called RandomForestRegressor, which included class definition, parameters including n_estimators (number of trees), criterion (splitting criterion), max_depth (maximum depth of trees), min_samples_split and some else, and most important is here are some methods including fit method, predict method, and score, and something else. As an expedient measure, many people use Grid Search CV to predict house price. The grid search is originally an exhaustive search based on a defined subset of the hyper-parameter space [6]. Grid Search Tree can predict house price more precisely, because users can choose several features before making predictions and reduce the randomness of the tree.

3. Methodology

As for the neural network model, the author would like to create a new model to be a subclass of torch.nn. torch.nn.Module is the base class for all neural network modules.

3.1. Data Processing and Basic Model Testing

The author chooses to import Pandas as pd to process the dataset. the author download a dataset from the website in CSV format and use pd to use this dataset. Irrespective of the ML algorithms used, data errors can adversely affect the quality of the generated model [7]. The first step is deal data with some missing features. Having some missing data is a normal thing, and we must deal with this problem at beginning because if a data has missing features, it will have a huge influence on model. The author uses the "dropna" function to complete this process. After deleting data with some features missing, the author divides the dataset into x and y; x is all the features that will have an influence on price, and y means price. The next step is dividing x and y into x_train, x_test, y_train, y_test. The train data is four times larger than test data, and the author uses the split function to achieve this. The next step is normalizing some features. It has been reported in the literature that normalization could improve performance [8]. The author chooses total rooms, total bedrooms, population, and household to stander

DOI: 10.54254/2755-2721/75/20240526

in this situation. The final step is to perform one-hot encoding. This can make some features enable to join model training. In this dataset, the author converts each category in the 'ocean proximity' column into a binary column (0 or 1) for each category. And drops the original 'ocean proximity' column from the train data since I have now replaced it with the dummy variables.

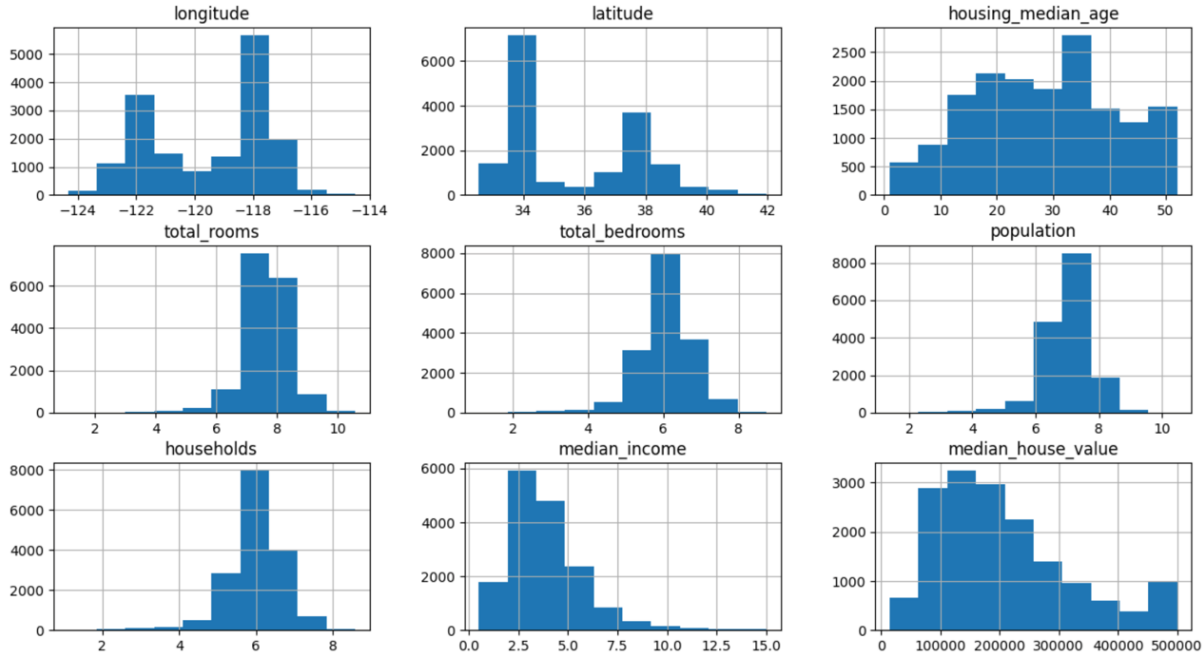


Figure 1. Exploring the distribution of values in different features

The author imports Linear Regression from sklearn.linear_model, import RandomForestRegressor from sklearn.ensemble, and from sklearn.model_selection the author imports GridSearchCV and then gives some values for GridSearchCV. To train those three models, the author uses the "fit" function, after a moment of waiting, we can get our trained model.

After training, we get three trained models. Let me show details about this training. The author chooses to use the "score" function to get the accuracy rate of models, and use "feature_importances" to show each weight in the model. Then, the author uses "predict(x_test)" to get the result and compare it with the true value(y_test). The first model is linear Regression; this model has a little high bias. Normally, its value is around 0.7, higher than the other two models. Next is Random Forest Regression, this accuracy is always higher than Linear Regression, maybe because this dataset has too many features. The last one is GridSearchCV; sometimes, it has higher accuracy than RandomForestRegressor and other times it has lower accuracy, it depends on which value we defined for it. So, to get a good performance, it is important to choose a sensible value for GridSearchCV.

3.2. Designing the New Model

After introducing the three regression models that we usually use, the author tries to define a neural network model and test it to see if it has higher accuracy. The author defines his own model as a subclass of nn.Moudle. The first method defined is __init__, and this is an initialization method. This model includes fully connected layer and ReLU Activation Function. Here are some details about this model. The first is input layer that accepts size of input. The input feature of size is connected to the first hidden layer with 512 neurons. The number of neurons in the subsequent hidden layer gradually decreases (256, 128, 64,32). The final layer gets 32 neurons and output a neuron that represents the predicted housing price. The ReLU activation function was applied after each hidden layer to introduce the nonlinear characteristics of the model. In order to reduce the risk of overfitting, the author adds a step called dropout in each layer. This function can help prevent the network from overly relying on specific

DOI: 10.54254/2755-2721/75/20240526

neurons to improve model generalization. This function achieves this goal by dropping out a random portion of neurons in the network by setting them to zero during training but retaining all neurons during testing. Then the author defines the forward pass method: __forward__, Pass the input x through the first linear layer (self.fc1). Apply the ReLU activation function to the output.

Repeat this pattern for each subsequent linear layer and ReLU activation until reaching the final linear layer (self.fc7).

To ensure this model have higher accuracy, the next step is the circuit training model, to achieve this goal, the author has changed the learning rate, neurons in each layer, layer's number, training times, and the startup function of the model. After a long time, the author finally gets a model that can predict more accurately than the traditional model we use.

4. Simulation Results

To show the accuracy of each model, the author chooses to use MSE. MSE can be used if there are outliers that need to be detected. In fact, MSE is great for attributing larger weights to such points, thanks to the L2 norm: clearly, if the model eventually outputs a single very bad prediction, the squaring part of the function magnifies the error [9]. $\mathrm{MSE} = (1/n) * \Sigma(y_i - \hat{y}_i)^2$, n is the number of samples, y_i is the actual observed value, $\hat{y}_i$ is the predicted value by the model.

The author uses mean_squared_error method to input y_test and (model's name). Predict(x_test_s)'s result to get a value.

As a result, the author gets from the program, Random Forest MSE: 2828475761.961905 Grid Search MSE: 2806549573.7166433 and linear regression's bias is much larger than the forest model. Then the author spends more energy on training my model after a long-time test, the author finds that when the learning rate is 0.0018 and the training times higher than 3000, this model can get higher accuracy. After each training, the author records the MSE value and compares it with the minimum error obtained so far. If the error is lower, replace the recorded model with the new model from this iteration, and update the minimum error with the current error.

Table 1. Training process

The process of training	
Training times	MSE value
1-1100	50000000000-10000000000
1100-1700	10000000000-4000000000
1700-2100	4000000000-3000000
2100-3000	3000000000-2000000

Then the author have tested three times: the first result is 2217174016.0; the second result is 2671392512.0, and the third result is 2661729024.0. Each of them is more reliable than Linear Regression and Random Forest.

Table 2. Final accuracy

Model	Neural model	Random Forest Tree	Grid Search Tree
Value	2217174016.0	2828475761.9	2806549573.7

5. Discussion

Data preprocessing plays a crucial role in enhancing the accuracy of a model before training. The effectiveness of these applications relies heavily on the quantity and quality of data used [10]. It involves handling problematic missing data, normalizing variables with large ranges, and other operations that contribute to minimizing bias and improving overall model performance. Failure to preprocess data adequately can lead to significant bias in the model.

DOI: 10.54254/2755-2721/75/20240526

Each model has its advantages and is suitable for specific scenarios. Linear regression is straightforward and easy to use, particularly when the number of influencing factors is limited. On the other hand, Random Forest Regression can outperform linear regression when dealing with numerous influencing factors, albeit at the cost of increased prediction time. While it may take longer to make predictions using Random Forest, its higher accuracy often justifies the trade-off.

Grid Search CV is a valuable tool for model tuning, ensuring the selection of reasonable hyperparameters. The trade-off lies in the number of values chosen; a higher number generally leads to increased accuracy, but this is not always the case.

In tasks like housing price prediction, where the influencing factors are relatively few, linear regression is a commonly chosen and effective model. However, despite its simplicity and widespread use, the author aims to explore the development of a neural network model for this task. Neural networks offer advantages in handling data normalization, providing more room for improvement. While regression models are more commonly used for housing price prediction due to their simplicity and efficiency, developing a neural network model adds value by allowing for more structured data formatting and potential performance enhancement.

6. Conclusion

For real estate predictions, where real-time updates may not be crucial and the validity of a single prediction lasts for an extended period, neural network models can be beneficial. Although neural networks have their drawbacks, such as longer prediction times, the extended validity of each prediction mitigates this concern. Hence, despite the prevalence of regression algorithms in housing price prediction, exploring neural network models remains meaningful.

In conclusion, while traditional regression models like linear regression and Random Forest Regression are well-established for housing price prediction, the integration of neural network models introduces a different perspective. The unique capabilities of neural networks in handling data formatting and their potential for performance improvement make them a valuable addition to the toolkit, particularly in scenarios where real-time updates are not critical, and the focus is on accurate and enduring predictions. And obviously, if the author trains the new model enough long time, it can predict house prices better. It can be an option when people need to predict house prices with enough time, and want to get a very precise prediction.

Building a neural network model requires a solid grasp of its underlying principles and mechanisms, which I am actively working towards acquiring. The journey involves not only comprehending the theoretical foundations but also gaining practical insights into the workflow of neural networks in PyTorch. This holistic understanding is essential for creating models that are not only effective but also well-suited for the intricacies of the task at hand-predicting housing prices.

References

[1] James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). Linear regression. In An introduction to statistical learning: With applications in python (pp. 69-134). Cham: Springer International Publishing.

[2] Poole, M. A., & O'Farrell, P. N. (1971). The assumptions of the linear regression model. Transactions of the Institute of British Geographers, 145-158.

[3] Santiago, C. B., Guo, J. Y., & Sigman, M. S. (2018). Predictive and mechanistic multivariate linear regression models for reaction development. Chemical science, 9(9), 2398-2412.

[4] Brophy, J., & Lowd, D. (2021, July). Machine unlearning for random forests. In International Conference on Machine Learning (pp. 1092-1104). PMLR.

[5] Reis, I., Baron, D., & Shahaf, S. (2018). Probabilistic random forest: A machine learning algorithm for noisy data sets. The Astronomical Journal, 157(1), 16.

[6] Syarif, I., Prugel-Bennett, A., & Wills, G. (2016). SVM parameter optimization using grid search and genetic algorithm to improve classification performance. TELKOMNIKA (Telecommunication Computing Electronics and Control), 14(4), 1502-1509.

DOI: 10.54254/2755-2721/75/20240526

[7] Breck, E., Polyzotis, N., Roy, S., Whang, S., & Zinkevich, M. (2019, March). Data Validation for Machine Learning. In MLSys.
[8] Jin, J., Li, M., & Jin, L. (2015). Data normalization to accelerate training for linear neural net to predict tropical cyclone tracks. Mathematical Problems in Engineering, 2015.
[9] Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science, 7, e623.
[10] Amato, A., & Di Lecce, V. (2023). Data preprocessing impact on machine learning algorithm performance. Open Computer Science, 13(1), 20220278.