electronics

Article

Used Car Price Prediction Based on the Iterative Framework of XGBoost+LightGBM

Baoyang Cui [1,2,3], Zhonglin Ye [1,2,3], Haixing Zhao [1,2,3,*], Zhuome Renqing [1,2,3], Lei Meng [1,2,3] and Yanlin Yang [1,2,3]

[1] The State Key Laboratory of Tibetan Intelligent Information Processing and Application, Xining 810008, China
[2] School of Computer, Qinghai Normal University, Xining 810008, China
[3] Tibetan Information Processing and Machine Translation Key Laboratory of Qinghai Province, Qinghai Normal University, Xining 810008, China
[*] Correspondence: hxzhao@qhnu.edu.cn

Abstract: To better address the problem of the low prediction accuracy of used car prices under a large number of features and big data and improve the accuracy of existing deep learning models, an iterative framework combining XGBoost and LightGBM is proposed in this paper. First, the relevant data processing is carried out for the initial recognition features. Then, by training the deep residual network, the predicted results are fused with the original features as new features. Finally, the new feature group is input into the iteration framework for training, the iteration is stopped, and the results are output when the performance reaches the highest value. These experimental results show that the combination of the deep residual network and iterative framework has a better prediction accuracy than the random forest and deep residual network. At the same time, by combining the existing mainstream methods with the iterative framework, it is verified that the iterative framework proposed in this paper can be applied to other models and greatly improve the prediction performance of other models.

Keywords: XGBoost; LightGBM; deep residual network; price; prediction; used car

check for updates

Citation: Cui, B.; Ye, Z.; Zhao, H.; Renqing, Z.; Meng, L.; Yang, Y. Used Car Price Prediction Based on the Iterative Framework of XGBoost+LightGBM. *Electronics* **2022**, *11*, 2932. https://doi.org/10.3390/electronics11182932

Academic Editors: Abdeldjalil Ouahabi and Kenji Suzuki

Received: 28 July 2022
Accepted: 13 September 2022
Published: 16 September 2022

1. Introduction

As of 2020, China's car ownership exceeds 280 million, and the huge car ownership has created great development space for the used car market. At the same time, with the implementation of China VI (the national sixth phase of motor vehicle pollutant emission standards is to implement the relevant laws and control the environmental pollution of compression-ignited and gas-fuel ignited engine vehicle exhaust) and the current widespread popularity of electric vehicles, the value of the retention rate of newly purchased cars is low, and many of the cars currently flowing in the market will be processed, which will lead to the accelerated development of the used car market.

In the context of the era of big data, people's lives are slowly being digitized and Internetized. The Internet turns most commodity transactions into e-commerce; that is, many transactions can be carried out on the Internet, such as Taobao, JD, Pinduoduo and many other e-commerce platforms, making people's lives more convenient. Similarly, the Internet has also turned used car trading into e-commerce, such as 58.com and other used car trading platforms. The e-commerce of used car transactions has further pushed the used car market to a climax, and various problems have gradually emerged, such as the lack of a unified standard for judging the value of used car assets. Used cars are affected not only by the basic configuration of the body—brand, car system and power—but also by the condition, mileage and age of the car, resulting in differences between the prices of used cars—one car, one condition and one price. Therefore, how to effectively assess the price of used cars requires a scientific valuation conversion method.

With the rapid development of machine learning methods and deep learning methods, it is possible to automate price prediction. Zhang et al. [1] introduced a complete integrated empirical modal decomposition algorithm with adaptive noise to extract the characteristics of stock price time series on the time scale and combined the attention mechanism with the neural network of the gated recurrent unit to predict stock prices and the Shanghai composite index. Cao et al. [2] designed an online short-term rental market price prediction model based on XGBoost and introduced a SHAP model to explain the characteristics. Fathalla et al. [3] proposed using a deep neural network based on LSTM and convoluted neural networks to predict price prediction, and the proposed model achieved a better average absolute error accuracy score than SVM. Based on the regression algorithm of random forest, Yan et al. [4] combined the Pearson coefficient and an improved grid search method to predict stock prices. Yao et al. [5] used HP filtering to decompose the stock index price time series into long-term trends and short-term fluctuations, used the LSTM neural network model to learn its sequence characteristics and make predictions, and fused the prediction results to obtain the stock index price prediction results. Kky et al. [6] proposed a hybrid GA-XGBoost prediction system with an enhanced feature engineering process consisting of feature set expansion, data preparation, and optimal feature set selection using the hybrid GA-XGBoost algorithm. Le et al. [7] propose to apply k-means, a clustering technique, together with Gradient Boost and XGBoost models to improve the prediction performance. Hernández-Casas et al. [8] successfully used an artificial neural network model to predict Mexican lobster export prices. Xu et al. [9] found that the displacement-prediction effect of XGBoost algorithm is better than that of SVR and RNNs in the sliding process of landslide with a large displacement value and small numbers of samples.

The used car market can be traced back to the beginning of the last century. After decades of development, the ratio of new cars to used cars has reached 1:3, and such a rapid development rate has caused a wave of research on the price of used cars by scholars in various fields. Purohit [10] studied the correlation between the used car market and the new car market and found that the price of used cars decreases as the number of new cars increases, and its consumers are more inclined to buy models with slow depreciation. Hansen et al. [11] developed a statistical model for the resale price prediction of used cars and found that random forests were more appropriate through empirical studies. Hu et al. [12] proposed the replacement cost method, which evaluates the price of the car on the basis of the classification treatment of the vehicle, but because the method is based on the price of the new car and there are many uncertain factors in the automobile market, there is uncertainty about the effect of the model. Andrews et al. [13] analyzed the relevant information of the eBay used car auction market through a linear regression method and concluded that car property rights and auction duration have a greater impact on price. To study the impact of the characteristics of the used car on the price, Richardson [14] analyzed multiple characteristics through a multiple regression analysis and concluded that the hybrid vehicle is more valuable. Qiang et al. [15] believed that when using the traditional methods to process massive data, there would be problems such as a slow processing speed and low accuracy, so they proposed a comprehensive optimization method based on the combination of a BP neural network and nonlinear curve fitting; the nonlinear curve fitting was carried out on the output of the BP neural network model, which greatly improved the accuracy of the price prediction of used cars. Zhang [16] established a multiple linear regression model and an artificial neural network model based on the crawled data of nearly 5000 used cars and found that the prediction accuracy of the artificial neural network model was higher by comparison. Liu [17] proposed predicting the price with the help of artificial neural networks without considering the vehicle condition, and then combining the prediction results with the vehicle condition factors to make a secondary prediction of the price, which obtained good results.

To further improve the accuracy of used car price prediction and improve the accuracy of existing mainstream models, this paper proposes an iterative framework of XGBoost+LightGBM. First, by training the deep residual network model, the training results are fused with the original features as new features. The resulting features are then repeatedly trained in the iterative framework to improve the prediction accuracy of the model. Finally, the effect of the model framework is verified by experiments, and the universality of the framework is verified by combining it with a variety of existing mainstream methods.

2. Models and Methods

2.1. Deep Residual Network

ResNet (deep residual network) is a network model that has been improved based on a deep convolutional neural network [18], and by learning the development process of a deep convolutional neural network, it can be found that the expression ability of the network increases with increasing the network depth. He Kaiming et al. [19–21] proved through experiments that for two network structures with the same time complexity, the deeper the structure, the better the performance. However, as the number of network layers increases to a certain extent, the network's performance degrades significantly. To solve this problem, a deep residual learning network is proposed.

Figure 1 shows a schematic diagram of the residual unit of the residual network. It can be seen from the figure that the output of the residual unit is obtained by adding the output of multiple convolutional layers and the original input, and then activated by the ReLU function, and the residual network is obtained by multiple residual units.

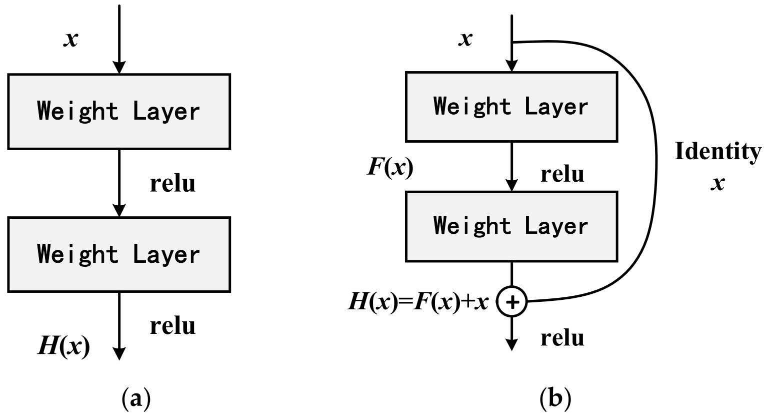

Figure 1. Residual unit. (**a**) Standard network structure diagram; (**b**) ResNet structure diagram.

For a residual unit, there is a functional relationship, as follows:

$$y_l = h(x_l) + F(x_l, W_l)\ (W_l = \left\{ W_{l,k}|_{1\le k\le K} \right\}, \tag{1}$$

$$x_{l+1} = f(y_l), \tag{2}$$

where y_l is the output of the lth residual unit and x_l is the input of the lth residual unit.

Let $h(x_l) = x_l, x_{l+1} = y_l$, then we have

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i, W_i), \tag{3}$$

That is, the input of the Lth residual unit can be expressed as the input of a residual unit of a certain layer and the sum of all the complex maps in the middle of it. If ε denotes the loss function, then by calculation, the backpropagation is calculated:

$$\frac{\partial \varepsilon}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=l}^{L-1} F(x_i, W_i)\right), \tag{4}$$

Obviously, at this time, the successive multiplication disappears, that is, the vanishing gradient will not exist.

The precondition of the above analysis is $h(x_l) = x_l, x_{l+1} = y_l$; to achieve this effect, the residual structure must be reconstructed. The improved residual structure fuses the activation layer into the residual branch and uses the residual unit preactivated by ReLU to satisfy the preconditions of the above analysis.

2.2. XGBoost

The XGBoost model [22] is a tree-based gradient boosting integrated model that is actually composed of multiple classification regression trees (CART) that learn the residual value of the sum of the target value and the predicted values of all previous decision trees, make a common decision after all decision trees have completed training, and finally accumulate the samples in the previous prediction results as the final prediction result.

During the XGBoost model training phase, each new tree is trained based on the tree that has been trained, and when a decision tree is generated, it needs to be pruned to prevent overfitting. To reduce the error, the XGBoost model trains the error obtained by each tree as the input of the next tree for training again, and gradually reduces the prediction error so that the model prediction result is gradually forced to the true value.

Suppose that (x_i, y_i) is the sample used in training. Then, the XGBoost prediction model can be expressed as

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i),\ f_k \in F, \tag{5}$$

where $x \in R^m$, $y \in R$, x represents the eigenvector, y represents the sample label, and $f_k(x_i)$ represents the kth decision tree.

The corresponding objective function is defined as follows:

$$Obj(0) = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{6}$$

The objective function $Obj(0)$ consists of two parts: the first part is the loss function determined by the specific task to assess the accuracy of the model's predictions; the second part is the regularization term, which is used to reduce the possibility of overfitting linearity in the model; The function is as follows:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|\omega\|^2, \tag{7}$$

where γ is the leaf node coefficient, and the function is to make XGBoost optimize and adjust the objective function, which is equivalent to a pre-pruning operation (that is, γT controls the complexity of the tree, and the larger the value, the greater the objective function value, which then suppresses the complexity of the model). While λ is the coefficient of the squared mode of $L2$, it also suppresses overfitting, and the entire $L2$ regularization term is used to control the leaf node weight fraction. If the value of the regularization term is 0, then the objective function is GBDT.

The XGBoost model adds regularization terms to the objective function and reduces the possibility of overfitting. It uses not only the first derivative, but also the second derivative to increase the accuracy of the loss function and customize the loss.

2.3. LightGBM

LightGBM [23] is a high-speed, distributed, and high-performance gradient boosting framework; its full name is Lightweight Efficient Gradient Boosting Tree (LGBM). It is based on a decision tree algorithm that can be used to complete sorting, classification, regression, and many other machine learning tasks. Under the condition of not reducing the accuracy, the speed is increased by approximately ten times, and the memory occupied is reduced by approximately three times, which has the advantages of high training efficiency, low memory occupation, high precision, and support for parallelism, and GPU can be used to process large-scale data. The main technical details of LGBM are as follows [24]:

(1) Histogram Algorithm

LightGBM is a decision tree algorithm based on histograms, which has great advantages in memory consumption and computational cost. The principle is as follows: first, discretize the continuous floating-point eigenvalues, convert them to k integers, and construct them as a histogram (bins) with a width value equal to k. When traversing the data, the discretized value is used as an index, and the statistics are accumulated in the histogram. After a data traversal, the required statistic has been accumulated in the histogram and then iterated in conjunction with the index of the histogram to search for the best partition point. Therefore, when performing node splitting, it is not necessary to calculate all the data when traversing each feature, but only k times, and the complexity is optimized from O(data * feature) to O(k* feature), which greatly accelerates the speed of training.

(2) Leafwise

The traditional GBDT algorithm splits each node of each larger node in parallel at the same time as the decision tree. This growth method is called levelwise growth, as shown in Figure 2, which treats each leaf of the same layer equally. However, some leaves in normal training do not need to continue to split and search, resulting in a waste of resources and excessive time expenditure.

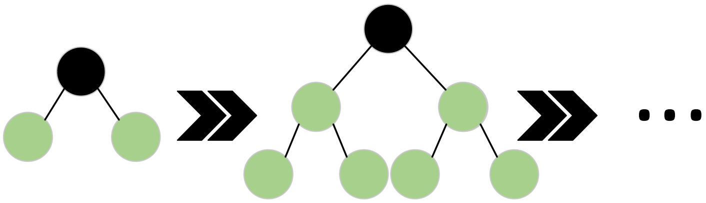

Figure 2. Schematic diagram of growth by layer. (Black nodes represent generated nodes in the tree; green nodes represent newly generated nodes at the current time step).

Therefore, LGBM proposes a more efficient strategy of leafwise growth. This method selects only the leaves with the largest splitting gain when splitting each layer of leaves, which can effectively reduce the time expenditure, as shown in Figure 3.

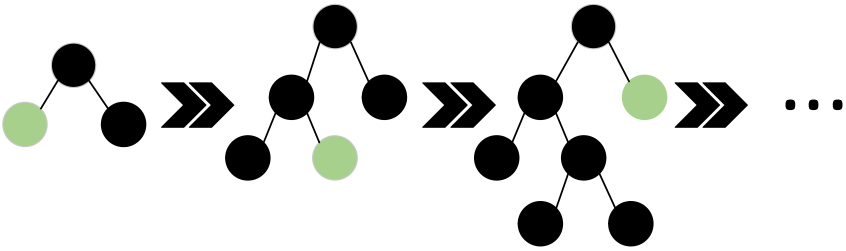

Figure 3. Schematic diagram of growth by layer. (Black nodes represent generated nodes in the tree; green nodes represent nodes selected for splitting at the current time step).

(3) Histogram Error Acceleration

Generally, as shown in Figure 4, after the black node splits into green child nodes, the histogram of the blue node can be obtained directly from the black node minus the green node, which greatly reduces the time overhead.

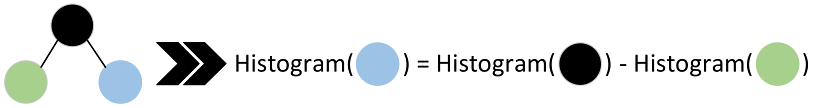

Figure 4. Histogram acceleration diagram.

(4) Gradient-Based One-Sided Sampling (GOSS)

LGBM is trained based on gradients, where the GBDT algorithm inputs all samples into the next classifier for training, but the optimization of the loss function is determined by those sample points with very large gradients, so LGBM uses the GOSS method to retain all large gradient samples and randomly selects some small gradient samples proportionally to participate in subsequent training. If there are 1 million pieces of data, of which 50,000 are large gradient samples (the gradient size threshold can be determined by yourself) and the rest are small gradient samples, the algorithm keeps these 50,000 samples and then randomly takes $x\%$ (x is a hyperparameter) samples from the small gradient sample size, which can also improve the speed.

2.4. DXL Model

To further improve the performance of the price prediction model, this paper combines the machine learning model and the deep learning model to design an XGB+LGBM iterative framework for multiple models. In this paper, a combination of ResNet and iterative framework is chosen to verify the performance of the XGB+LGBM iterative framework, and the combined model is named DXL (Deep Residual Network-XGB+LGBM).

Whether it is a machine learning model or a deep learning model, the downstream task is completed by learning the characteristics of the data. Then, we determine whether the repeated training will obtain better results by combining the obtained results as new features with the original features into a group of new features.

As an important step in the development of artificial intelligence, the effect of neural networks has been widely recognized by scholars in various fields, but it is limited by the size of the data scale, and overlarge-scale data will also reduce the training speed, so it is not suitable for iterative loops. However, XGBoost and LightGBM have been widely used in a variety of prediction tasks in recent years, and both are extension algorithms based on decision trees, which assume that they have the same sensitivity to the same features. Multimodel alternate iterative training also avoids the excessive dependence of a single model iterative learning on new features.

Through the above analysis, the XGBoost and LightGBM models are combined to obtain the XGB+LGBM iterative framework, and the new features are used for iterative learning training. Figure 5 is the flowchart of the DXL model, and the specific steps of the model are as follows:

Step 1: The original data are preprocessed and feature engineered to obtain the original features.

Step 2: The original feature is trained into the deep residual network, and the result is combined with the original feature to form a new set of features.

Step 3: Input the new features into XGBoost for training, evaluate the results, and output the results if the peak is reached; otherwise, combine the results as new features with the original features to form a new set of features.

Step 4: Input the new features into LightGBM for training, evaluate the results, and output the results if the peak is reached; otherwise, combine the results as new features with the original features to obtain a new set of features, and then complete Step 3.

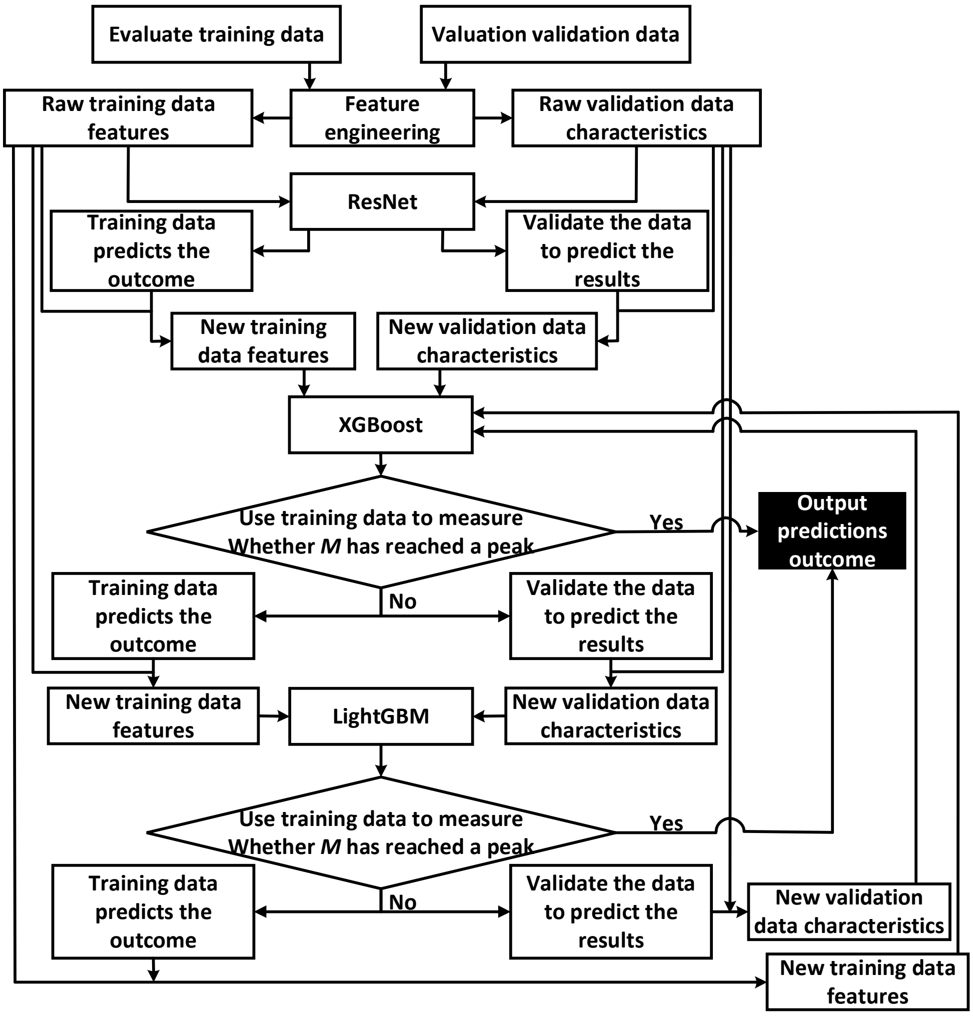
Evaluate training data
Valuation validation data
Raw training data features
Feature engineering
Raw validation data characteristics
ResNet
Training data predicts the outcome
Validate the data to predict the results
New training data features
New validation data characteristics
XGBoost
Use training data to measure Whether M has reached a peak
Yes
Output predictions outcome
No
Training data predicts the outcome
Validate the data to predict the results
New training data features
LightGBM
New validation data characteristics
Use training data to measure Whether M has reached a peak
Yes
No
Training data predicts the outcome
Validate the data to predict the results
New validation data characteristics
New training data features

Figure 5. DXL model flowchart.

The DXL Algorithm 1 pseudocode is as follows:

Algorithm 1: DXL algorithm

```
    Input: Train_data and Test_data (No label);
    Output: Prediction results Pr of Test_data;
    Training ResNet models by using Train_data;
    The results of Train_data and Test_data predicted by using the trained ResNet model;
    Add the prediction result as a new feature pre_result to the original feature;
    min_loss = ∞;
    index = 1;
    while loss < min_loss do:
        min_loss = loss;
        if index%2 == 1:
            Training with XGB;
        else:
            Training with LGBM;
Calculate the average loss value between the predicted and actual values of Train_data;
The pre_result feature is replaced with the current round prediction Pr;
end
    Output prediction results Pr;
```

3. Experiments and Analysis of Results

The data used in this research are derived from the car valuation training data and verification data used in the 2021 MathorCup big data competition, with a total of 30,000 training data, each with 20 features and 15 anonymous features. The basic characteristic properties of the data are shown in Table 1. Through the simple statistics and observations of the data, it is found that there are some serious missing characteristics that need to be processed.

Table 1. Basic characteristics of the data.

ID	Features	Type	ID	Features	Type
1	carid	int	12	registerDate	string
2	tradeTime	string	13	licenseDat	string
3	brand	int	14	country	int
4	serial	int	15	maketype	int
5	model	int	16	modelyear	int
6	mileage	float	17	displacement	float
7	color	int	18	gearbox	int
8	cityId	int	19	oiltype	int
9	carCode	int	20	newprice	float
10	transferCount	int	21	anonymousFeature*15	int or string
11	seatings	int	22	price	float

3.1. Feature Engineering

3.1.1. Special Feature Treatment

Process that time data, such as 2011-3-7 to 20110307. Anonymous feature 11 is processed and classified into integer data according to the data. By observing and guessing that the anonymous feature 12 is a volume feature, calculate the volume of the data of feature 12, such as 4220 × 1740 × 1625, and replace the original data. To facilitate training, the obtained data are divided by 109 and two decimal places are retained.

3.1.2. Missing Value Treatment and One-Hot Coding

Through statistics, it is found that there are 15 attributes with missing values, excluding features with missing values accounting for more than 15%, and less than 15% of the features are national standard code, year model, manufacturer type, anonymous feature 1, anonymous feature 11 and anonymous feature 13, and the missing value of the integer data in these features takes the majority of the feature, while for floating-point data, the missing value takes the average value of the feature. The missing data are shown in Figure 6.

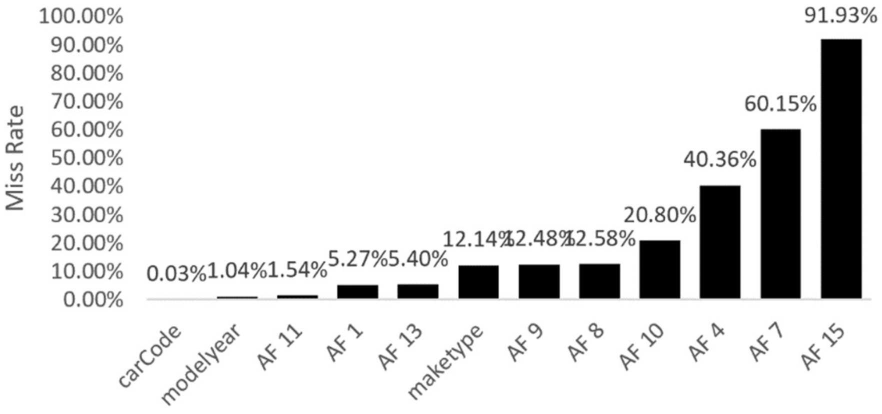

Figure 6. Data missing condition.

At the same time, most of the data with feature type integers are encoded uniquely, and a total of 79 features are obtained.

3.2. Model Parameter Setting

The DXL model uses ResNet, XGBoost and LightGBM models. For ResNet, we first use four linear layers, and the output of each layer uses the ReLU activation function, which constitutes a residual block. The model consists of four residual blocks, two self-attentive layers and one linear layer, for a total of 38 layers. One self-attentive layer is used after every two residual blocks. In addition, a Dropout of p = 0.1 is applied to the input of the last layer. Finally, the linear layer is used to output the prediction results.

Some important parameters of the XGBoost model are set as shown in Table 2.

Table 2. XGBoost partial parameter setting.

Parameters	Parameter Description	Value
n_estimators	The number of learners.	150
learning_rate	The step size when updating the weights for each iteration.	0.1
gamma	Specifies the minimum loss function descent value required for node splitting.	0
subsample	Sample subsampling. Randomly select a certain percentage of samples to train the tree.	0.8
colsample_bytree	Feature sampling rate. The proportion of features sampled when building the tree.	0.9
max_depth	Tree depth. The maximum depth of each basic learner tree.	8

The parameters n_estimators, learning_rate and max_depth of the LightGBM model are consistent with XGBoost, and the rest of the parameters are set as shown in Table 3.

Table 3. LightGBM partial parameter setting.

Parameters	Parameter Description	Value
feature_fraction	Proportion of randomly selected features.	0.8
bagging_fraction	Randomly selecting a portion of the data without resampling.	0.8
bagging_freq	The value k represents the execution of bagging per k iteration.	2
num_leaves	The maximum number of leaf nodes of the tree.	32

3.3. Evaluation Indicators

This article directly takes the evaluation indicator M used in this data contest and calculates it as shown in the following formula:

$$M = 0.2 * (1 - Mape) + 0.8 * Accuracy_5, \tag{8}$$

where $Mape$ is the average relative error and is the error accuracy of 5%. The calculation formulas are as follows:

$$Ape = \frac{|\hat{y} - y|}{y}, \tag{9}$$

$$Mape = \frac{1}{m}\sum_{i=1}^{m} Ape_i, \tag{10}$$

$$Accuracy_5 = \frac{count(Ape \leq 0.05)}{count(total)}, \tag{11}$$

where Ape is the relative error, $y = (y_1, y_2, \cdots, y_m)$ is the true value, the model predicted value is $\hat{y} = (\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_m)$, $count(Ape \leq 0.05)$ is the sample size of Ape within 5% of the relative error, and $count(total)$ is the total number of samples.

3.4. Used Car Price Forecast and Comparative Analysis

After training the model by the decile cross-validation method, the model effect of the XGB+LGBM iteration frame 30 times is obtained, as shown in Figure 7.

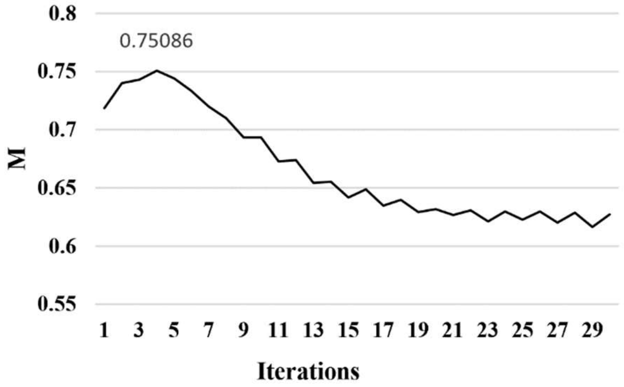

Figure 7. Graph of the number of iterations versus the results.

From Figure 7, it can be observed that as the number of model iterations increases, the evaluation standard M results rise and reach a peak, and as the number of iterations continues to increase, the evaluation standard results show a certain degree of decline and then tend to stabilize. This shows that the effect of the model proposed in this paper increases with the increase in the number of model iterations and tends to be stable after a certain decrease in the effect after reaching the peak, so this paper sets the model termination condition to end the experiment after reaching the peak. As seen in Figure 7, for this dataset, the evaluation criterion M reaches a peak of 0.75 when iterating four times, that is, the accuracy of the model proposed in this article on the used car valuation task reaches 75%.

Table 4 shows the training results of the test set on other methods.

Table 4. The accuracy of the test set on other models.

Model	Accuracy (%)
linear regression	23
Random forest	51
LightGBM	55
XGBoost	53
ResNet	63
DXL	75

From the observation in Table 4, it can be seen that the method proposed in this paper is superior to the current mainstream car price prediction method, which is 52 percentage points higher than the linear regression method and 17 percentage points higher than the current better depth residual network.

Table 5 shows the comparison of the effects of other models and the effect comparison after adding the XGB+LGBM iterative framework.

Table 5. Add XGB+LGBM iterative framework before and after the effect comparison.

Model	Effect (%)	Add XGB+LGBM Iterations Frame Effects (%)
Random forest	51	63
XGBoost	53	65
LightGBM	55	66
ResNet	63	75

From Table 5, we can see that the effects of the random forest, XGBoost model, LightGBM model and depth residual network have been greatly improved by nearly

12 percentage points after the additive XGB+LGBM iterative framework. This shows that the iterative framework proposed in this paper can be applied to other models and greatly improve the prediction performance of other models, and the algorithm framework is universal and generalizable.

4. Conclusions

With the rapid growth of the economy, the car ownership rate is rising, which also bestows the used car market with excellent development prospects, but the accuracy of existing models in the price prediction task is average. Taking the real car data provided by the MathorCup big data competition as the research object, this paper constructs a model framework based on multimodel fusion. First, by training the deep residual network, the prediction results are fused with the original features as new features. Second, the combined features are input into the XGB+LGBM iterative framework, and the prediction results are output when the M index reaches its peak through cross-validation. Then, the effectiveness of the DXL algorithm is verified by comparative experiments with five mainstream algorithms: linear regression, random forest, XGB, LGBM and depth residual network. Finally, by combining the four mainstream algorithms with the XGB+LGBM iterative framework, it is verified that the XGB+LGBM iteration framework can be applied to other models and greatly improve the performance of the original model.

Author Contributions: Algorithm design and conception, B.C.; experimental instruction and paper revision, Z.Y. and H.Z.; data collection, Z.R.; literature research and analysis, L.M. and Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China, grant number 2020YFC1523300; the Youth Program of the Natural Science Foundation of Qinghai Province, grant number 2021-ZJ-946Q; and the National Natural Science Foundation for Young Scholars of China, grant number No. 62007019.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, Q.Y.; Yan, D.M.; Han, J.D. Research on Stock Price Prediction Combined with Deep Learning and Decomposition Algorithm. *Comput. Eng. Appl.* **2021**, *57*, 56–64.
2. Cao, R.; Liao, B.; Li, M.; Sun, R.N. Predicting Prices and Analyzing Features of Online Short-Term Rentals Based on XGBoost. *Data Anal. Knowl. Discov.* **2021**, *5*, 51–65.
3. Fathalla, A.; Salah, A.; Li, K.; Li, K.; Francesco, P. Deep end-to-end learning for price prediction of second-hand items. *Knowl. Inf. Syst.* **2020**, *62*, 4541–4568. [CrossRef]
4. Yan, Z.X.; Qing, C.; Song, G. Random Forest Model Stock Price Prediction Based on Pearson Feature Selection. *Comput. Eng. Appl.* **2021**, *57*, 286–296.
5. Yao, Y.; Zhang, C.Y. Stock Index Price Forecasting Method Based on HP Filter. *Comput. Eng. Appl.* **2021**, *57*, 296–304.
6. Kky, A.; Sang, W.; Dw, A. Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process. *Expert Syst. Appl.* **2021**, *186*, 115716.
7. Le, H.T.; Huy, T.D.; Le, A.N. Clustering helps to improve price prediction in online booking systems. *Int. J. Web Inf. Syst.* **2021**, *17*, 45–53.
8. Hernández-Casas, S.; Beltrán-Morales, L.F.; Vargas-López, V.G.; Vergara-Solana, F.; Seijo, J.C. Price Forecast for Mexican Red Spiny Lobster (*Panulirus* spp.) Using Artificial Neural Networks (ANNs). *Appl. Sci.* **2022**, *12*, 6044. [CrossRef]
9. Xu, J.; Jiang, Y.; Yang, C. Landslide Displacement Prediction during the Sliding Process Using XGBoost, SVR and RNNs. *Appl. Sci.* **2022**, *12*, 6056. [CrossRef]
10. Purohit, D. Exploring the Relationship between the Markets for New and Used Durable Goods: The Case of Automobiles. *Mark. Sci.* **1992**, *11*, 154–167. [CrossRef]
11. Hansen, J.V.; Mcdonald, J.B.; Turley, R.S. Partially adaptive robust estimation of regression models and applications. *Eur. J. Oper. Res.* **2006**, *170*, 132–143. [CrossRef]
12. Hu, N.; Chen, Z.H.; Jia, J.S.; Wang, Y. Study on The Value Evaluation Methods of Used Vehicle. *Shanghai Auto* **2005**, *12*, 16–19.

13. Andrews, T.; Benzing, C. The Determinants of Price in Internet Auctions of Used Cars. *Atl. Econ. J.* **2007**, *5*, 43–57. [CrossRef]
14. Richardson, M. Determinants of Used Car Resale Value. Doctoral Dissertation, The Colorado College, Colorado, CO, USA, 2009.
15. Gongqi, S.; Yansong, W.; Qiang, Z. New Model for Residual Value Prediction of the Used Car Based on BP Neural Network and Nonlinear Curve Fit. In Proceedings of the 2011 3rd International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Shanghai, China, 6–7 January 2011.
16. Zhang, Y.S. *A Used Cars' Price Forecasting Model Based on Artificial Neural Network*; Tianjin University: Tianjin, China, 2018.
17. Liu, S. Research on Used Car Price Evaluation Method Based on Neural Network. *Auto Ind. Res.* **2019**, *1*, 21–24.
18. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
19. He, K.; Jian, S. Convolutional neural networks at constrained time cost. In Proceedings of the 2015 28th Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
20. He, K.; Zhang, X.; Ren, S. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 29th Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26–30 June 2016.
21. He, K.; Zhang, X.; Ren, S. Identity Mappings in Deep Residual Networks. In Proceedings of the 2016 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016.
22. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.
23. Wang, D.; Zhang, Y.; Zhao, Y. LightGBM: An Effective miRNA Classification Method in Breast Cancer Patients. In Proceedings of the 2017 5th International Conference on Bioinformatics and Computational Biology (ICBCB), Hong Kong, China, 6–8 January 2017.
24. Xiang, X.J. *Research on Second-Hand Car Forecast Based on Machine Learning*; Southwest University: Chongqing, China, 2021.