Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 14, No. 2, 2025

Research on the Pricing Method of Automotive Aftermarket Parts Based on Distributed Computing Architecture

Qingyi Dong[1, 2], Chao Zhang[1, 2, *], Tengjian Yang[1, 2], Peipei Zhu[1, 2]

[1]China Automotive Technology and Research Center Co., Ltd. Tianjin, China
[2]China Auto Information Technology (Tianjin) Co., Ltd. Tianjin, China
[*]Corresponding author: m13502037247_3@163.com

Abstract: With the rapid development of the automobile industry, the scale of the automobile aftersales parts market continues to expand, and its pricing is of great significance to both enterprises and consumers. The traditional pricing method has many limitations, it is difficult to adapt to the dynamic changes of the market. In this paper, an automatic pricing tool for automotive aftermarket parts based on distributed computing architecture is deeply studied, aiming to improve the accuracy and efficiency of pricing. Firstly, the cost and sales price functions of auto aftersales parts are determined. Based on the characteristics of discrete and multi-dimensional data of auto aftersales parts, new features are constructed by using feature engineering method, including data acquisition and preprocessing, model architecture design, etc. Then, it uses the calculation principle of Spark and Spark Shuffle to build distributed computing system and distributed training strategy. Finally, the model is evaluated and optimized, and the existing risks and countermeasures are proposed. The method in this paper can effectively solve many problems faced by the traditional pricing, and improve the accuracy and real-time pricing. It will bring significant economic and social benefits to the enterprise, and promote the healthy development of the automobile aftermarket.

Keywords: Distributed; Feature engineering; Training strategy; Aftermarket parts for cars.

1. Background

As an important means of transportation in modern society, the number of cars continues to grow, and the automobile after-sales parts market has also developed rapidly. The pricing of after-sale parts directly affects the profits of automobile maintenance enterprises, the maintenance costs of consumers and the competition pattern of the entire market. However, the traditional pricing method mainly relies on manual experience and market research, which is difficult to consider many complex factors, resulting in pricing is not accurate enough to adapt to the changes of the market in real time.

At present, there are mainly the following ways to price auto aftermarket parts: cost-plus-based pricing, market-oriented pricing and competitor pricing. The cost plus pricing simply considers the cost and expected profit of the parts, ignoring the market demand and competition factors; Market-oriented pricing focuses on changes in market demand, but lacks the ability to analyze and process market data in depth; Competitor pricing mainly refers to the prices of competitors, which is easy to lead to price wars and market chaos. Automobile after-sales parts data has the characteristics of diversity, dynamic and large-scale, different brands, models, specifications of the parts of its characteristics and price factors are not the same, while the market dynamics and consumer demand are constantly changing, which makes the processing and analysis of data more difficult. The price of spare parts is affected by a variety of factors, such as brand, quality, market demand, supply, etc., there are complex causal relationships and mutual effects between these factors, it is difficult to use traditional statistical methods for accurate modeling. Based on the above factors, this paper cites the distributed architecture method. Distributed computing and distributed storage technology can store massive after-sale parts data on multiple nodes, and improve data processing power and computing efficiency through parallel processing.

The distributed architecture method is used in all walks of life. For example[1], in order to solve the problems that the existing researches on spatial data models do not involve the behavior characteristics of entities and lack the description of individual behaviors, an innovative database system is proposed, whose underlying architecture is specially built for the distributed computing of dynamic behaviors. The system adopts "mixed-match" data storage architecture, combined with "master-slave" database storage strategy, in order to optimize the data storage and management process. This realization breaks through the limitation of traditional dynamic behavior as process simulation, provides strong distributed computing support for the dynamic behavior of spatio-temporal objects, and shows significant performance advantages. The literature[2] introduces distributed computing to solve the problem of insufficient traditional stand-alone processing capacity, and analyzes data security and privacy protection in combination with the characteristics of large-scale data processing, so as to provide technical support for promoting digital transformation and information construction. In order to efficiently store and manage big data of water conservancy and geospatial, a distributed storage and computing architecture based on Hadoop and Spark is proposed in the literature[3]. At the same time, the memory and cache mechanism of Spark are rationally used to design and implement a distributed storage and computing method for spatial big data. Based on the DEM data of a hydropower station reservoir area, the calculation method proposed in this paper is verified, and the method has higher computing efficiency and good scalability.

To sum up, this paper aims to build an auto aftermarket

parts automatic pricing tool based on distributed computing, fully exploit and utilize massive aftermarket parts data, comprehensively consider the influence of various factors on prices, improve the accuracy and real-time pricing, provide scientific and reasonable pricing decision support for enterprises, and promote the healthy development of the automobile aftermarket.

2. Research Methods

The research method of this paper includes four parts: the determination of accessory price calculation method, distributed calculation principle, model evaluation and optimization, model tuning and risk control. Among them, the determination of the calculation method of the spare parts price is mainly based on the characteristics of the automobile after-sale parts data, to generate new characteristics; The principle of distributed computing mainly includes SPARK memory scheduling mechanism, SPARK Shuffle principle and other computer science content; Model evaluation and optimization use accuracy, mean square error (MSE), mean absolute error (MAE) and other indicators to evaluate the pricing accuracy of the model. Risk control mainly focuses on the research of data privacy and security, model interpretability, data update and model adaptability, etc. The research methods in this paper are shown in Figure 1 below.

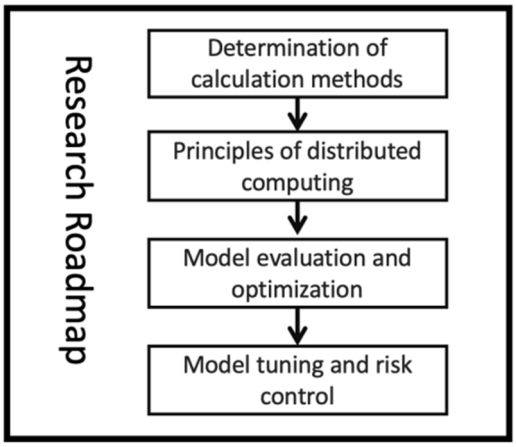

Figure 1. Roadmap of research methods in this paper

2.1. Determine the calculation method of after-sale parts price

2.1.1. Cost and selling price pricing model

The calculation method of after-sale parts price mainly includes auto parts cost pricing model and parts sales pricing function, as follows.

(1) Construct the auto parts cost pricing model function, as shown in formula (1) below.

$$p = f(x_i), i = 1, \cdots, n \tag{1}$$

Where, p is the cost price of auto parts, f is the fitting polynomial function, x_i is the index after simplified pricing evaluation, and n is the number of indicators after simplified pricing.

(2) Construct the parts sales pricing function

Construct the auto parts sales pricing model function, as shown in the following formula (2).

$$p' = p \times (1 + \%m) \tag{2}$$

Where, p' is the sale price of auto parts, p is the cost pricing function of new energy auto parts, m is the profit percentage (compared to cost pricing).

2.1.2. Feature engineering

Before feature engineering, it is necessary to clean the collected data, remove duplicate, invalid and wrong data records, and then carry out pre-processing operations, such as data standardization, missing value filling, feature extraction, etc. The data after cleaning and preprocessing is stored in[4] a distributed file system or database.

Feature engineering is mainly to generate new features based on the characteristics of auto aftermarket parts data. The brand, model and other classification features of the unique thermal coding, the numerical characteristics of the normalization process, at the same time for different features cross combination, generate new features, in order to better describe the relationship[5] between the parts and the price. For discrete data, in order to facilitate the following calculation, it is necessary to carry out numerical processing of factor indicators. The numerical processing calculation function of the characteristic index is shown in the following formula (3).

$$s = \frac{\max\_s - \min\_s}{num(feature)} \times index(feature) + \min\_s \tag{3}$$

Where, s is the processed data, $\max\_s$ is the upper limit of numerical processing, $\min\_s$ is the lower limit of numerical processing, $num(feature)$ is the number of eigenvalues of an influencing factor, and $index(feature)$ is the ranking of the index eigenvalues.

2.2. Principle of distributed computing

Distributed computing and distributed storage technology can store massive after-sale parts data on multiple nodes, and improve data processing capacity and computing efficiency through parallel processing. At the same time, the distributed architecture has good scalability and fault tolerance, and can adapt to the rapid growth of data volume and the high load of processing tasks.

In the actual operation process, distributed deep learning frameworks such as TensorFlow and PyTorch are used to design a reasonable distributed training scheme combined with the model parallelism of data parallel and the update strategy of synchronization and asynchronism of parameters. The data is processed in parallel on multiple computing nodes, and the parameters of the model are updated synchronously or asynchronously to improve the training speed of the model.

2.2.1. Spark memory scheduling mechanism

In Spark, an application includes three basic concepts: Job, Stage and Task. A Spark application is usually composed of multiple jobs. It is a logical unit, which represents a complete logical execution process in the user code. Job refers to a set of RDD conversion operations and action operations. These operations form a directed acyclic graph. Whenever an action operation is invoked, Spark will trigger a new job on the DAG graph. Typically, each job will correspond to one or more input RDDS, and one or more output RDDS. When a job is

broken down into smaller units, it is called a phase, and a phase is made up of a group of tasks with the same computational logic that can be executed in parallel and without relying on the output of other tasks. Phase division is carried out according to the dependency relationship between RDD. Narrow dependency does not result in phase division, while wide dependency involves Shuffle operation of data in different partitions, so it involves phase division. When encountering a wide dependency (such as Shuffle operation), the phase is divided into the smallest execution unit Task. The Spark execution engine will try its best to optimize the data transmission and calculation process between phases to improve the overall execution efficiency, which involves the Spark memory scheduling content[6] that will be studied in the following sections. The Spark architecture diagram is shown in Figure 2.

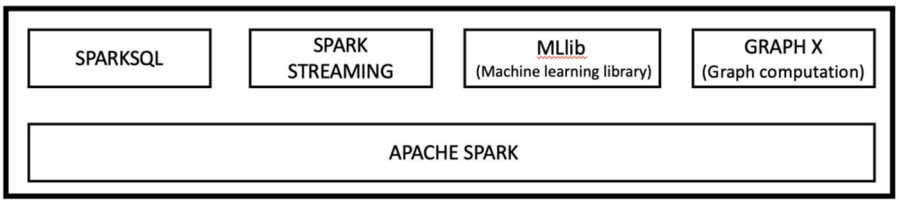

Figure 2. Spark architecture diagram

2.2.2. Basic Principle of Spark Shuffle

Spark Shuffle refers to a data repartitioning operation in the Spark computing framework, which is used to transfer data from nodes in the previous computing phase to nodes in the next computing phase in distributed computing. The Shuffle operation is very common in Spark computing, especially when dealing with large data sets. It is an essential step in many data-intensive computing tasks, such as grouping and merging by primary key. The core goal of Spark Shuffle is to efficiently and evenly distribute data to each compute node through reasonable data distribution policies and network transmission technologies, thereby improving the execution efficiency and performance[7] of the entire computing task. The schematic diagram of Spark Shuffle is shown in Figure 3.

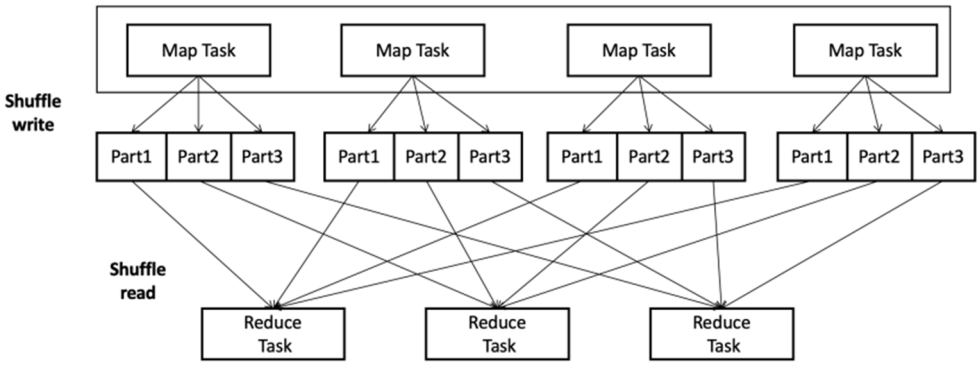

Figure 3. Schematic diagram of Spark Shuffle

2.3. Model evaluation

(1) K-fold cross-validation

The cross-validation method is used to evaluate the generalization ability of the model and ensure the stability and validity of the model on different data sets. Cross-validation is a method of evaluating the performance of a model by training and testing it multiple times. It divides the data set into subsets, and then in turn uses one of them as the test set and the rest as the training set[8-9].

Method: Divide the data set into K subsets, selecting one of them at a time as the test set and the remaining K-1 subsets as the training set. Repeat this process K times, each time selecting a different subset as the test set.

(2) Performance evaluation indicators

For different types of models and tasks, different performance evaluation indicators can be used to measure the performance of the model. Use the following performance evaluation metrics and their formulas[8-9]:

① Accuracy definition: the proportion of the number of samples correctly predicted by the model to the total number of samples. Formula: Accuracy = (true example + true counterexample) / (true example + true counterexample + false positive example + false counterexample);

② Accuracy rate definition: the proportion of positive samples that are actually positive samples in all instances predicted as positive samples. Formula: precision rate = true examples / (true examples + false positive examples);

③ Recall is defined as the percentage of all instances that are actually positive samples that are correctly predicted to be positive samples. Formula: Recall rate = True examples / (true examples + false counterexamples).

2.4. Model tuning and risk control

2.4.1. Model tuning

According to the evaluation results, the model was optimized by adjusting the model structure, increasing the amount of data and improving the training algorithm. The model is updated regularly to adapt to new data and market changes, so that the model is always efficient and accurate.

(1) Model structure optimization

Model structure tuning refers to improving the performance of the model by changing the structure of the model. In machine learning and deep learning, the structure of a model has a crucial impact on its performance. In this paper, the following methods are adopted to optimize the

structure of the model and improve the performance of the model.

In the neural network model, the structure is optimized based on the adjustment of the number of layers, the number of nodes, and the activation function. In various tree models, adjusting the depth of the tree, the number of leaf nodes, splitting criteria and other methods are used to optimize the model structure.

(2) Hyperparameter tuning

The goal of hyperparameter tuning is to find an optimal set of hyperparameter configurations such that the model performs best on a given data set and task. This usually involves conducting a search across multiple hyperparameters, evaluating the model's performance under each configuration, and selecting the optimal configuration based on the performance results. The following are commonly used hyperparameter tuning in engineering.

① Grid Search: full parameter combination traversal, suitable for small parameter space.

② Random Search: efficiently explore large parameter Spaces.

③ Bayesian Optimization: guided search based on probabilistic models.

In some neural network evolution algorithms, the learning rate, regularization coefficient (L1/L2), the maximum depth of the tree. Batch size, number of iterations are tuned.

2.4.2. Risk control

(1) Data privacy and security issues: In the process of pricing automotive aftermarket parts, issues such as model interpretability and data update and model adaptability involve a large amount of data information, such as sales data. There are potential data privacy and security risks. Therefore, in this paper, operations regarding data privacy and security access are carried out in full compliance with data laws.

(2) Model interpretability problems: The deep learning model is a black box model, and its internal decision-making mechanism is difficult to explain. In this paper, the interpretability of the model is improved based on code and expository text, and users' trust in the model is increased.

(3) Data update and model adaptability: the market data of automobile aftermarket parts is in dynamic change, new products appear, the market supply and demand relationship changes, if the original model can not update and process the data in time, it may lead to inaccurate prediction results. Therefore, this paper designs a dynamic monitoring model update and learning mechanism at any time, so that the model can quickly adapt to the new data, maintain continuous forecasting ability.

3. Discussion and Analysis

The experimental results show that the pricing model based on distributed computing architecture has outstanding performance in accuracy, mean square error, average absolute error and other evaluation indicators in comparison with traditional pricing models, and can more accurately capture the relationship between after-sale parts prices and various influencing factors. At the same time, distributed computing significantly improves the training speed and scalability of the model. Through the analysis of the importance of different features, the actual role of each influencing factor in pricing is deeply understood, and key information is provided for further optimization of the model and pricing strategy. Therefore, the auto after-sale parts automatic pricing tool based on distributed and deep learning proposed in this paper can effectively solve many problems faced by traditional pricing methods and improve the accuracy and real-time pricing. It will bring significant economic and social benefits to enterprises, and promote the healthy development of the automobile after-sales market.

References

[1] Zeng Mengxiong, Zhang Zheng, Zhang Jiangshui, et al. [J]. Journal of Science and Technology of Surveying and Mapping, 2024, 40 (06): 658-665. (in Chinese)

[2] Wang Lei, Chen Ying. Research on large-scale data processing technology in distributed computing environment [J]. China Science and Technology Investment, 2024, (26): 27-29.

[3] Yuan Zewen, Zhou Guocheng, Zhou Shengjie, et al. Application of distributed storage and calculation method in big data of water conservancy geospatial [J]. Surveying, Mapping and Spatial Geographic Information, 2024, 47 (07): 10-13.

[4] Xie Yuting, Yang Wei, Qin Jie, et al. Power User Behavior Profiling based on unsupervised Learning and Feature Engineering [J]. Journal of Electric Power Systems and Automation, 2025, 37 (01): 112-119. DOI:10.19635/j.cnki.csu-epsa.001460.

[5] LIU Yulin, BAI Yang, Cui Bin, et al. Review of Automated feature Engineering for Machine Learning [J]. Journal of Computer Applications and Software, 2025, 42 (01): 1-10+40.

[6] Ma Xing. Spark distributed computing platform performance optimization research [D]. University of electronic science and technology, 2024. The DOI: 10.27005 /, dc nki. Gdzku. 2024.003029.

[7] Huang Xu. Spark platform performance and resource optimization technology research [D]. Hebei university, 2023. The DOI: 10.27103 /, dc nki. Ghebu. 2023.002273.

[8] LI Changqi. Research on Intelligent Risk Assessment Model of Driving Behavior Based on Machine Learning Technology [J]. Automotive Knowledge, 2025, 25 (02): 87-90.

[9] Yang Dan, Yin Liyan. Meter based on hidden markov model long-term stability assessment study [J]. China's new technology and new products, 2025, (3) : 52-54, DOI: 10.13612 / j.carol carroll nki CNTP. 2025.03.021.