# Used Car Price Prediction Based on Machine Learning Algorithms

Mingliang Wei

Beijing Wuzi University, Beijing, 101149, China
2421130079@bwu.edu.cn

**Abstract.** Accurate used car price prediction has become increasingly important as the second-hand vehicle market continues to grow. Buyers and sellers often struggle to agree on fair prices because of information gaps. Expert-based pricing, while common, tends to be slow and inconsistent across different markets. Machine learning could help, but it is not obvious which algorithm works best for this particular problem. Five methods were compared here, namely Linear Regression, k-Nearest Neighbor (KNN), Random Forest, XGBoost, and a Combined Boosting model that blends Random Forest and XGBoost outputs through weighted averaging. The experiments used a Kaggle dataset of 4,009 records with 12 vehicle attributes, split 70/30 into training and test sets. Mean Squared Error served as the evaluation metric. Ensemble methods clearly beat linear and instance-based models. XGBoost gave the best results among single models, yet the Combined Boosting approach achieved the lowest test error overall. The fused model appears to capture price patterns more reliably than any individual algorithm alone.

**Keywords:** used car price prediction, ensemble learning, XGBoost, random forest, combined boosting

## 1 Introduction

The used car market has grown fast. More buyers now prefer pre-owned vehicles for cost reasons, but pricing remains a headache. Sellers and buyers often disagree on what a car is worth because no one has complete information [1]. Dealers still price cars based on experience, which works for common models but breaks down for rare ones. Machine learning can help by finding patterns in vehicle data that humans miss [2]. Picking the right algorithm is tricky, though. Car prices depend on many factors, including brand, year, mileage, and fuel type, and these interact in non-linear ways. Categorical data like brand names need special handling, and market prices fluctuate. All of this makes automated pricing hard to get right.

© The Author(s) 2026
L. C. Yoong (ed.), *Proceedings of the 2026 2nd International Conference on Financial Innovation and Marketing Management (FIMM 2026)*, Advances in Economics, Business and Management Research 408,
https://doi.org/10.2991/978-94-6239-801-6_140

Some work has already been done on predicting used car prices with machine learning. Bukvic et al tested linear regression and decision trees on used vehicle data and achieved a prediction accuracy of 95% after model training, showing that supervised learning can pick up price trends from vehicle attributes [3]. Hankar et al tried several regression methods and found gradient boosting regressor gave the highest R-squared value of 0.9725 and the lowest root mean square error of 0.05 among all tested models [4]. Das Adhikary et al built an ensemble model using LightGBM, which beat single models on prediction accuracy and showed that combining learners works well for this task [5]. Nandan and Ghosh used a bagging regressor to estimate resale values [6]. Chandana et al built meta-ensemble models with feature engineering to make predictions more stable [7]. Most of these studies focused on one or two methods, though. No one has compared linear models, instance-based methods, bagging, boosting, and a fused approach side by side using the same data and setup. That gap is what this paper tries to fill.

Five algorithms are compared here, namely Linear Regression, k-Nearest Neighbor (KNN), Random Forest, XGBoost, and a Combined Boosting model that fuses Random Forest and XGBoost predictions. All are tested on the same Kaggle dataset of 4,009 used car records. Mean Squared Error measures which method predicts prices best.

## 2 Methodology

### 2.1 Dataset Description and Selection

The data comes from a public Kaggle dataset with 4,009 used car records and 12 attributes [8]. Numeric fields include year, mileage, and engine size. Categorical fields cover brand, fuel type, and transmission. Price is the target variable, so this is a regression task. Kaggle was chosen because its datasets are well-documented and have been used in many prior studies, which means the data quality is reasonable, and others can reproduce the work. The sample size is large enough to train several models without running into memory issues, and the features cover the main factors that affect resale value, including year, mileage, brand, and engine size. This makes the dataset a decent proxy for the broader used car market.

### 2.2 Models and Algorithms

Linear Regression is the simplest model tested. It assumes price changes linearly with each feature, fitting a straight line through the data by minimizing squared errors. The model is easy to understand and runs fast, so it makes a good baseline [9].

KNN works by finding the five cars in the training data that are most similar to the one being priced, then averaging their prices. Similarity is measured with Euclidean distance

after normalizing the features. KNN can pick up non-linear patterns because it does not assume any particular shape for the data [10].

Random Forest builds many decision trees on random subsets of the data, then averages their predictions. Breiman came up with this idea, and it turned out to work much better than single trees for messy real-world data [1]. Each tree sees only a random sample of rows and a random subset of features at each split, which keeps individual trees from memorizing the training data. The ensemble handles both numeric and categorical inputs well without much preprocessing [11].

XGBoost adds trees one at a time, with each new tree trying to fix the mistakes of all previous trees. Chen and Guestrin designed the algorithm with built-in regularization that penalizes overly complex models, which helps prevent overfitting without sacrificing too much accuracy [2]. XGBoost also handles missing values on its own and supports parallel processing, making it both fast and practical. It has won many machine learning competitions and is a standard tool for tabular data problems [12].

The Combined Boosting model simply takes a weighted average of Random Forest and XGBoost predictions, with weights 0.4 and 0.6 respectively. The idea is that Random Forest brings stability through bagging while XGBoost brings accuracy through gradient boosting, and mixing them might capture benefits from both. The weights were chosen based on how each model performed on cross-validation data, with XGBoost receiving the heavier weight because it individually predicted better.

### 2.3 Experimental Design

The experimental design follows a standard pipeline. Missing values and outliers are removed from the dataset, followed by one-hot encoding to convert categorical features into numerical binary vectors. StandardScaler normalization is then applied to transform all numerical features into a uniform range. The preprocessed dataset is split into 70% training (2,806 samples) and 30% testing (1,203 samples), with random state set to 42 for reproducibility. Five-fold cross-validation is employed during training. Each model uses standard hyperparameters, specifically KNN with k=5, Random Forest with 100 estimators, and XGBoost with 100 estimators and a learning rate of 0.1. Mean Squared Error is adopted as the evaluation metric, measuring the average squared difference between predicted and actual prices.

Fig. 1 illustrates the complete experimental pipeline, from raw data ingestion through preprocessing, model training, and final evaluation.

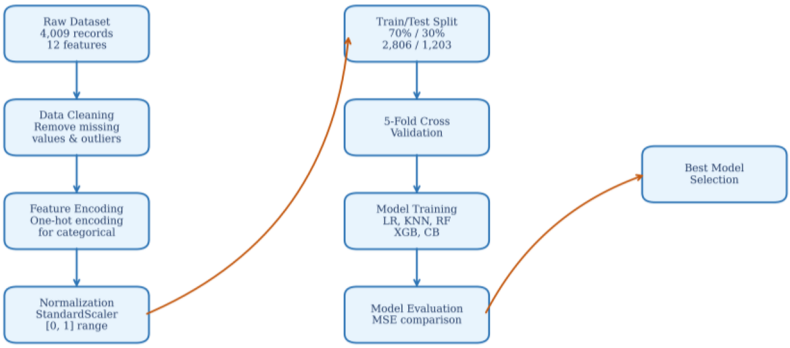



**Fig. 1.** Experimental pipeline of the comparative study (Photo credit: Original).

## 3 Results

**Table 1.** The MSE for all five models across training, cross-validation, and test sets, with Fig. 2 visualizing the same results as grouped bar charts.

| Model | Training MSE | Cross-validation MSE | Test MSE |
|---|---|---|---|
| Linear Regression | 3.56 | 3.61 | 3.59 |
| KNN | 2.12 | 2.37 | 2.41 |
| Random Forest | 1.01 | 1.35 | 1.42 |
| XGBoost | 0.87 | 1.19 | 1.23 |
| Combined Boosting | 0.92 | 1.15 | 1.18 |

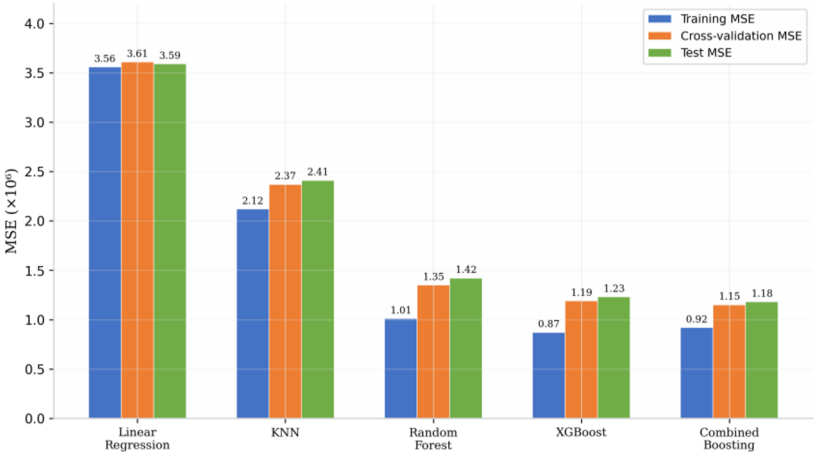



**Fig. 2.** Performance comparison of five machine learning models (Photo credit: Original).

As Table 1, Fig. 2, Linear Regression missed the mark by a wide margin, since car prices do not follow straight lines. KNN did better but overfitted, its training error being much lower than its test error, meaning it memorized training examples rather than learning general rules. Random Forest cut the error roughly in half compared to KNN, showing that bagging helps. XGBoost beat Random Forest, likely because boosting focuses on hard cases rather than averaging everything. The Combined Boosting model edged out XGBoost by a small margin, suggesting that mixing two different ensemble strategies adds something extra. The 0.6 weight on XGBoost makes sense since it individually performed best.

# 4 Discussion

## 4.1 Analysis of Model Advantages and Limitations

Looking at the numbers, each method has clear pros and cons. Linear Regression had the highest error because car prices do not follow straight lines [9]. KNN could model non-linear patterns, but overfitted badly and ran slowly because every prediction requires a full search through the training data [10]. Random Forest did well overall, with bagging keeping variance in check. Individual trees can still fixate on dominant features, though. XGBoost fixed this by boosting, where each tree learns from prior mistakes, and the built-in regularization kept overfitting under control [2]. The Combined Boosting model took things one step further by mixing Random Forest and XGBoost outputs, building on what Tanga et al showed, that stacking different ensemble types can push accuracy higher than any single model [12].

## 4.2 Model Optimization Recommendations

There are ways to push these models further. Random Forest and XGBoost could benefit from proper hyperparameter tuning. Grid search or Bayesian optimization might find better settings than the defaults used here. Fayyaz et al showed that feature engineering, things like polynomial expansion and automated selection, helps Random Forest accuracy for car pricing [13]. For the Combined Boosting model, the fixed weights (0.4/0.6) could be replaced with a meta-learner that adjusts them based on input. Feature importance from tree models also deserves attention. Huang used random forest feature analysis and found a clear ranking of price factors, with mileage at the top [14]. That kind of insight could guide the creation of new features that better capture what really drives used car prices.

### 4.3 Limitations of the Experimental Design

The study has several shortcomings worth mentioning. The dataset only covers basic vehicle specs, including brand, year, and mileage. It says nothing about condition ratings, service history, accidents, or local market trends. Velidi et al found that adding such features meaningfully improves ensemble predictions, so their absence here likely caps model performance [15]. Using a single Kaggle dataset also raises questions about whether these results would hold elsewhere. Netayawijit et al showed how the split train and test data affect rankings, so a different split might change which model looks best [16]. Deep learning was not tested either, though neural networks have worked well on similar regression problems. The Combined Boosting weights were fixed at 0.4/0.6. Adaptive weights might work better for some market segments.

### 4.4 Future Research Directions

There is still room to push this work further. Adding more features would help, including vehicle condition ratings, service history, or even photos of the cars. Neural networks could also be worth trying, since deep learning has shown promise on tabular data when paired with the right preprocessing. Velidi et al found that bringing in extra data sources improved their results, and a similar approach might work here [15]. On the practical side, turning the model into a real-time pricing tool for dealerships would be a natural next step. Testing on data from other countries or regions would also matter, since car markets differ. The Combined Boosting weights were fixed in this study, but letting them adapt to local market conditions could make the model more flexible.

## 5 Conclusion

Five machine learning methods were tested on used car price data, namely Linear Regression, KNN, Random Forest, XGBoost, and a Combined Boosting model. Ensemble methods came out on top, with XGBoost being the best single model. The Combined Boosting approach, which mixes Random Forest and XGBoost predictions, performed even better. The study has its limits. The dataset only includes basic vehicle specs, not condition or service records. Deep learning models were not tested. And everything was run on one dataset, so it is unclear how well the results transfer to other markets. Future work could add more features, try neural network hybrids, and test across different regions to see if the findings hold up. If these improvements pan out, the resulting models could help both buyers and sellers make better pricing decisions in real-world used car transactions.

## References


1. Breiman, L.: Random Forests. Machine Learning 45, 5–32 (2001)
2. Chen, T., Guestrin, C.: XGBoost: A Scalable Tree Boosting System. In: Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
3. Bukvić, L., Pa š agić Š krinjar, J., Fratrović, T., Abramović, B.: Price Prediction and Classification of Used-Vehicles Using Supervised Machine Learning. Sustainability 14(24), 17034 (2022)
4. Hankar, M., Birjali, M., Beni-Hssane, A.: Used Car Price Prediction Using Machine Learning: A Case Study. In: Proc. 2022 11th International Symposium on Signal, Image, Video and Communications (2022)
5. Das Adhikary, D.R., Sahu, R., Shaik, M.S.A., et al.: Prediction of Used Car Prices Using Machine Learning. In: Nature-Inspired Techniques in Machine Learning and Deep Learning, pp. 131–142. Springer (2022)
6. Nandan, M., Ghosh, D.: Pre-owned Car Price Prediction by Employing Machine Learning Techniques. Journal of Decision Analytics and Intelligent Computing 3(1), 1–15 (2023)
7. Chandana, Y., Aishwarya, G., Sirisha, P., et al.: Pre-Owned Car Price Prediction via Cross-Validated Meta-Ensembles and Feature Composites. In: Proc. 2025 1st IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (2025)
8. Kaggle: Used Car Dataset, https://www.kaggle.com/datasets/avikasliwal/used-cars-dataset, last accessed 2026/05/19
9. Qu, K.: Research on Linear Regression Algorithm. MATEC Web of Conferences 404, 01046 (2024)
10. Cunningham, P., Delany, S.J.: K-nearest Neighbour Classifiers: A Tutorial. ACM Computing Surveys 54(6), 1–25 (2022)
11. Nuzulia, R., Misbullah, A., Farsiah, L., et al.: Comparative Analysis of XGBoost and Random Forest for Used Car Price Prediction. In: Proc. 2024 International Conference on Artificial Intelligence and Big Data. IEEE (2024)
12. Tanga, Y.M.P., Simanjuntak, R.P.R., Mesran, M., et al.: Improving Car Price Prediction Performance Using Stacking Ensemble Learning Based on ANN and Random Forest. Journal of Soft Computing Exploration 5(2), 181–190 (2024)
13. Fayyaz, I., Ali, G.G.M.N., Khairunnesa, S.S.: Advanced Feature Engineering and Machine Learning Techniques for High Accurate Price Prediction of Heterogeneous Pre-Own Cars. Vehicles 7(3), 94 (2025)
14. Huang, J.: Price Prediction and Analysis of Price Influencing Factors for Second-Hand Car Sales in AutoTrader Based on XGBoost Algorithm. World Scientific Research Journal 11(3), 28–38 (2025)
15. Velidi, S., Vaartha, N.C., Gupta, K.V.V.N.S., et al.: Enhanced Used Car Price Prediction Using Ensemble Machine Learning Techniques. In: Proc. International Conference on Smart Computing and Societal Applications, pp. 45–58. Springer (2025)
16. Netayawijit, P., Chansanam, W., Sorn-In, K.: An Optimized Data Partitioning Framework for Benchmarking Ensemble and Linear Regression Models in Used-Vehicle Price Prediction. Engineering, Technology & Applied Science Research 16(1), 16878–16885 (2026)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

