Used Car Price Prediction Using Machine Learning Models

Code Repository: https://github.com/yanran0307/STATS-507-Final-Project-Repository

Yanran Chen
Department of Statistics
College of Literature, Science, and the Arts, University of Michigan
Ann Arbor, USA
yanranc@umich.edu

***Abstract*—This project used a dataset from Kaggle to create a machine learning workflow for predicting used car prices. It included several stages such as data cleaning, feature extraction, and feature engineering, modeling, etc. Of the three models we used—linear regression, random forest, and XGBoost—XGBoost performed the best, with an R² of about 0.85. The analysis of feature importance showed that mileage, year, car age, and engine displacement are the most important factors affecting price. This further supports the effectiveness of tree models in valuing used cars.**

***Index Terms*—Used car price prediction, machine learning, XGBoost, random forest, regression modeling**

I. Introduction

In recent years, the used car market has continued to expand, and it has become more convenient for people to buy and sell used cars. However, some new problems have also emerged. The prices of the same car often differ significantly on different platforms, and the pricing process often relies on the salesperson's experience or the buyer's subjective judgment, lacking transparent and systematic data support. For ordinary consumers, without relevant professional knowledge, it is difficult to judge whether the price of a used car is ”reasonable,” and it is also difficult to identify vehicles with prices that are too high or too low.

This project was inspired by my own experience buying a used car in Ann Arbor. I noticed that factors such as brand, mileage, year, and accident history significantly impacted prices, but there was a lack of data-driven pricing tools available for both buyers and sellers. Therefore, the goal of this project is to build a machine learning workflow for used car price prediction based on the Kaggle public dataset ”used_cars.csv”. The system will compare various regression models (including linear regression, random forest, and XGBoost) and, through feature importance analysis, explain which factors have the greatest impact on used car prices, thus providing consumers and sellers with a more intuitive pricing reference.

Existing research has demonstrated that using machine learning to predict used car prices is a feasible and effective approach. For example, Kang et al. [1], based on approximately 3 million used car data points from CarGurus in the United States, compared Linear Regression, Random Forest Regression, and Gradient Boosting Tree models on the Spark big data platform. The results showed that gradient boosting tree performed best in RMSE and R², while random forest, with similar accuracy, had a shorter training time, making it suitable as a model that balances efficiency and performance.

Another study by AlShared [2] focused on the UAE used car market. By crawling local website data and performing missing value processing, feature encoding, and correlation analysis, they trained random forest regression, linear regression, and Bagging Regression models. Random forest achieved approximately 95% R² and the lowest error rate, and was selected as the primary model. These studies demonstrate that, given sufficient data cleaning and modeling, tree models typically outperform simple linear models. This is helpful for me to choose models for my project.

Compared to these studies, this project places greater emphasis on a complete data science workflow and the interpretability of the results. On one hand, we systematically cleaned and performed feature engineering on raw string fields such as price, mileage, and engine, and compared various models under a unified preprocessing workflow. On the other hand, through feature importance analysis using a new model XGBoost and Random Forest, we explored the marginal impact of features on price, answering questions like ”which factors are most valuable” from a data perspective. In summary, this research aims to build a well-structured and interpretable used car price prediction system, and provide a reusable baseline for future larger-scale applications.

II. Data & Methods

A. Dataset Description

In this project, we use the ”used_cars.csv” dataset on Kaggle, which contains approximately 4,000 used car records and 12 raw feature fields. The data includes brand, model, model year, mileage, fuel type, transmission, exterior color (ext_col), interior color (int_col), engine description, accident history, title status (clean_title), and price.

These fields contain the core dimensions affecting used car valuation, including numerical variables (e.g., price, mileage, model year), as well as a large number of categorical variables and text descriptions (e.g., color, engine description). The structure of the data has many non-standardized fields. For example, prices are presented as strings (containing dollar sign and ”,”), mileage includes the ”mi.” suffix, and the engine field contains multiple specifications. Therefore, systematic data cleaning and feature extraction are important before modeling.

Before modeling, data is split into a feature matrix X and target y. Then it is further divided into a training set and a test set in an 80/20 ratio to provide independent samples for model evaluation.

B. Problem Definition

This project can be viewed as a typical supervised regression problem: predicting the fair market price of a used car based on its various characteristics (e.g., brand, mileage, year, engine, accident history, etc.). Our goal is to make the model's predicted price as close as possible to the actual transaction price, and to use metrics such as MAE, RMSE, and R² to measure the model's performance. Also, we place special focus on the model's interpretability. We want to explore ”which features have the greatest impact on price” from a data perspective.

C. Workflow Overview

The project follows this complete workflow:

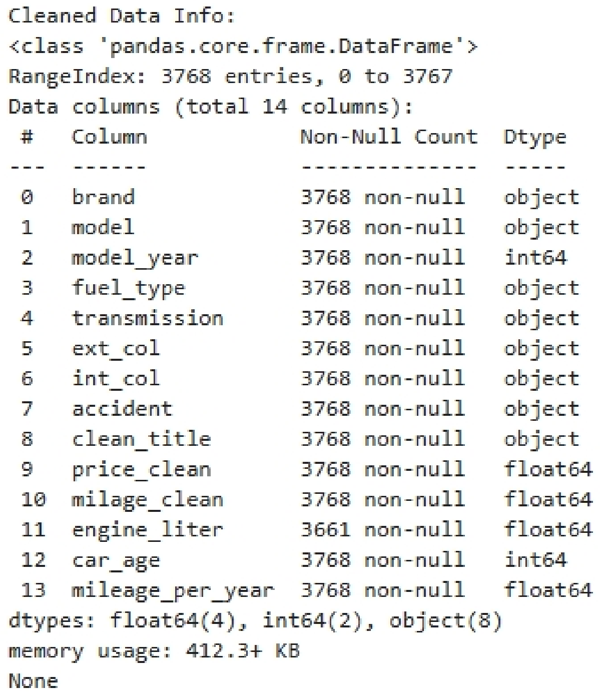
- Data Cleaning
- Feature Engineering
- Encoding & Preprocessing Pipeline
- Modeling
- Model Evaluation
- Feature Importance

D. Data Cleaning

Before starting modeling, we systematically cleaned and standardized the raw data to ensure all features could be correctly identified and used by the model. First, for numeric fields stored as strings, such as price and mileage, we performed structured processing: removing dollar signs, commas, and irrelevant characters like ”mi.”, and converting them to floating-point numbers. The engine description field contains various information such as displacement, horsepower, and number of cylinders. We used regular expressions to extract the displacement (engine_liter) and filled missing values using the median of vehicles of the same brand.

Second, to maintain uniformity of categorical variables, we uniformly converted the text fields such as brand, model, fuel type, and color to lowercase and trimmed spaces. Simultaneously, accident records and title status were mapped to a standard yes/no format to eliminate noise introduced by textual differences.

Furthermore, to reduce the effect of extreme values on modeling, we removed some obviously abnormal data based on business logic, such as vehicle samples with prices exceeding \$100,000, mileage exceeding 300,000 miles, or age exceeding 50 years. The entire cleaning process ensured the integrity and consistency of the data, giving a solid foundation for subsequent feature engineering and modeling.



```
Cleaned Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3768 entries, 0 to 3767
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   brand            3768 non-null   object
 1   model            3768 non-null   object
 2   model_year       3768 non-null   int64
 3   fuel_type        3768 non-null   object
 4   transmission     3768 non-null   object
 5   ext_col          3768 non-null   object
 6   int_col          3768 non-null   object
 7   accident         3768 non-null   object
 8   clean_title      3768 non-null   object
 9   price_clean      3768 non-null   float64
 10  milage_clean     3768 non-null   float64
 11  engine_liter     3661 non-null   float64
 12  car_age          3768 non-null   int64
 13  mileage_per_year 3768 non-null   float64
dtypes: float64(4), int64(2), object(8)
memory usage: 412.3+ KB
None
```

Fig. 1. Cleaned Dataset Summary After Data Cleaning

E. Feature Engineering

After data cleaning, to improve the model's ability to characterize price changes, we performed further feature engineering on vehicle attributes. First, based on the original mileage and model_year fields, we constructed two key derived variables: car_age and mileage_per_year. The car_age directly reflects the depreciation trend of a vehicle over time, while mileage_per_year is more comparable than the original mileage, avoiding scale differences caused by different car ages. We created these two variables through this formula:

- **car_age** = current year – model year
- **mileage_per_year** = mileage / (car_age + 1)

Secondly, the engine displacement (engine_liter) extracted from the engine description (engine) is also incorporated into the model to reflect the differences in vehicle power performance. Furthermore, to help the model better understand category information, all discrete fields (such as brand, model, fuel type, transmission, interior/exterior color, accident history, and title status) are cleaned and uniformly processed by ColumnTransformer for One-Hot Encoding, converting them into numerical features that the machine learning model can directly use.

F. Modeling

After completing data preprocessing and feature engineering, we trained and compared multiple regression models using a unified modeling workflow. All models were based on the same ColumnTransformer preprocessing pipeline, which included One-Hot Encoding and numerical imputation, ensuring consistency and fairness in comparisons between different models. This pipeline, as the first step, automatically encodes categorical features, imputes missing values, and inputs numerical features, making the modeling process more standardized and reproducible, while also effectively avoiding potential data leakage between the training and test sets.

We selected three representative regression models for comparison:

(1) Linear Regression, the most basic linear baseline model, is used to test whether the used car pricing problem is suitable for being described by a linear assumption.

(2) Random Forest Regressor, a tree model based on ensemble learning, can capture the nonlinear relationship between features well and is relatively robust.

(3) XGBoost Regressor, the core model of this project, is widely used in structured data prediction tasks due to its powerful nonlinear modeling ability, feature interaction capture ability, and high prediction performance. In terms of parameter tuning, we appropriately optimized key parameters of XGBoost such as n_estimators, max_depth, and learning_rate to achieve a balance between prediction performance and training efficiency.

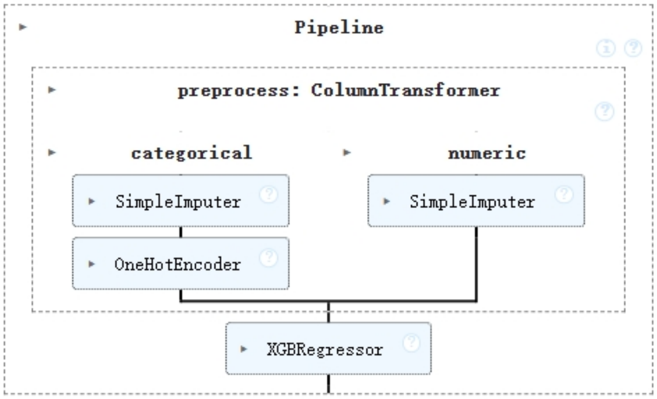


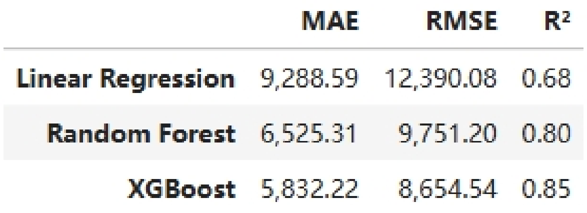
Fig. 2. End-to-End Modeling Pipeline with ColumnTransformer and XGBoost

By integrating these three models into the same pipeline, we constructed an end-to-end, structured modeling process, laying the foundation for subsequent performance evaluation and feature importance analysis.

III. Results

After training the three models separately, we systematically evaluated their predictive performance based on the test set, using MAE, RMSE, and R² to measure the model's fitting ability and error level. The results show that the linear regression model performed the weakest, with a high RMSE and an R² of only around 0.68. This indicates that the simple linear assumption is insufficient to characterize the complex nonlinear structure of used car prices. In contrast, the random forest regression model showed a significant improvement, with both RMSE and MAE lower than the linear model, and an R² of approximately 0.80, demonstrating the advantage of tree models in handling mixed-type features and nonlinear relationships.

The best performing of the three was the XGBoost model, which not only achieved the lowest MAE and RMSE but also an R² of around 0.85. This indicates that the model can more fully capture the interaction relationships between vehicle attributes and effectively reduce prediction errors in the mid-to-high price range. The scatter plot of Actual vs Predicted also shows that XGBoost's points are closest to the ideal diagonal; while Random Forest performs well, it has higher dispersion; and Linear Regression's error is the most dispersed.



	MAE	RMSE	R²
Linear Regression	9,288.59	12,390.08	0.68
Random Forest	6,525.31	9,751.20	0.80
XGBoost	5,832.22	8,654.54	0.85

Fig. 3. Comparison of Model Performance (MAE, RMSE, R^2)

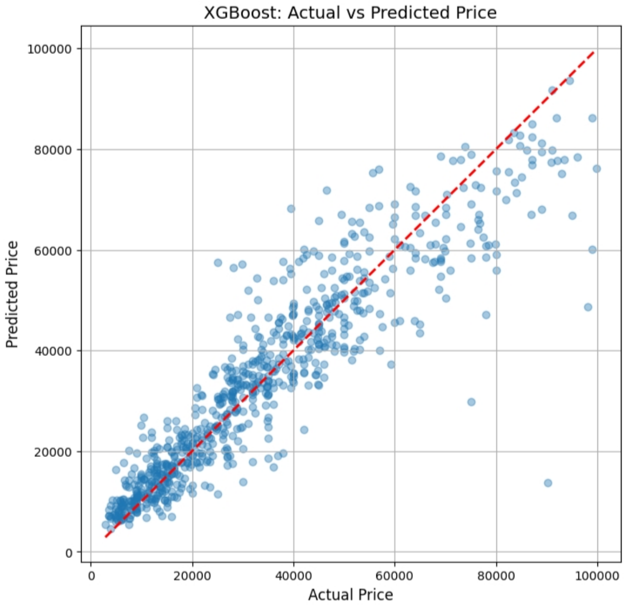


Fig. 4. XGBoost: Actual vs Predicted Price

Regarding interpretability, we performed feature importance analysis and extracted the importance ranking of the top 20 features based on XGBoost and Random Forest. The results show that mileage, model year, car age, and engine displacement are the factors with the greatest impact on price, which is consistent with the valuation logic of the real used car market. Furthermore, some brand and model features encoded with One-Hot (such as Porsche, BMW, Rivian, etc.) also rank highly in the tree model, reflecting the influence of brand premium; while fields such as certain exterior or interior colors

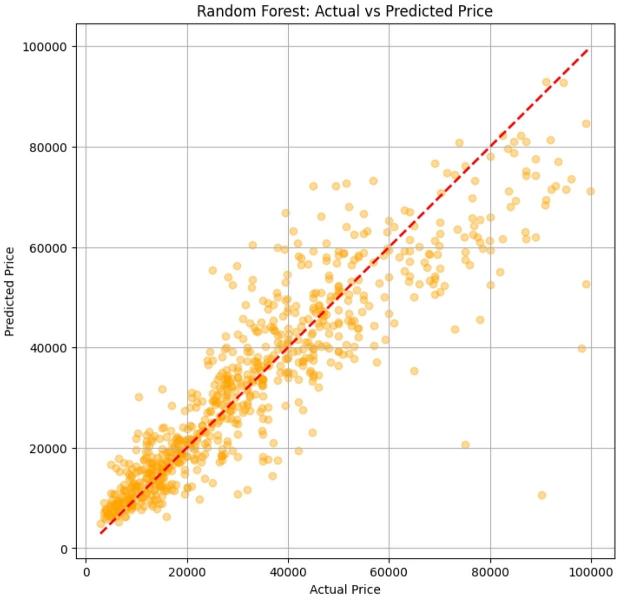


Fig. 5. Random Forest: Actual vs Predicted Price

have lower importance, indicating that their impact on price is relatively limited.

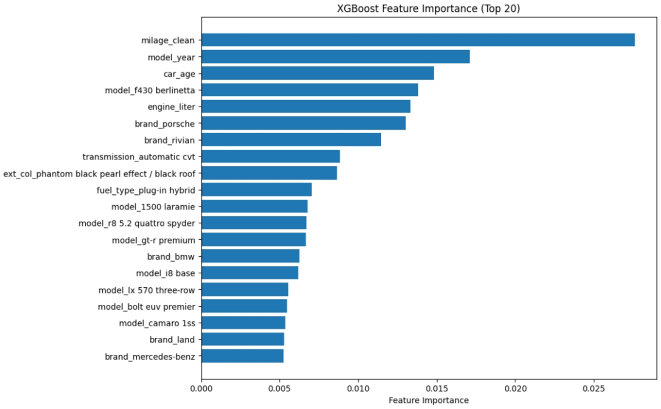


Fig. 6. XGBoost Feature Importance (Top 20)

Overall, the experimental results show that the tree model can more comprehensively utilize the non-linear patterns in structured used car data, while XGBoost shows a significant advantage in capturing complex relationships between features.

IV. Conclusion

This project, based on a used car dataset from Kaggle, implemented a complete machine learning workflow, covering data cleaning, feature engineering, model training, and interpretability analysis. Through a systematic comparison of three types of models, we found that tree models have a significant advantage in handling complex structured features, with XGBoost achieving the best prediction performance due to its powerful nonlinear modeling and feature interaction capture capabilities.

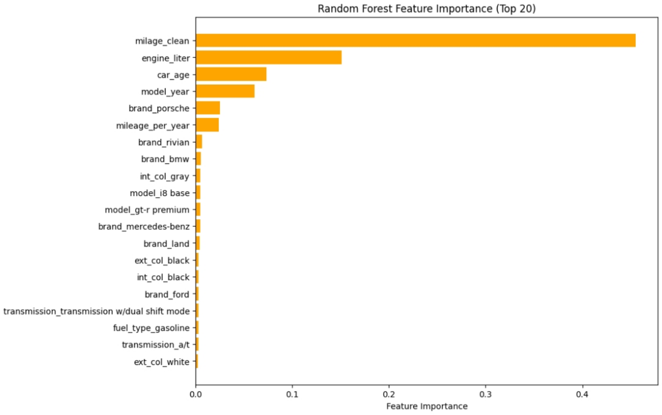


Fig. 7. Random Forest Feature Importance (Top 20)

Feature importance analysis further revealed key factors influencing used car prices, such as mileage, model year, car age, and engine displacement, while also confirming the premium effect inherent in brand and model.

In summary, this research built a high-performance and well-structured price prediction workflow. It also showed how machine learning can be applied to used car valuation. Future research could increase the data scale, add more features like text or images, and look into more complex models such as deep learning to create a better and more precise price prediction system.

References

[1] J. I. Kang, H. Parekh, P. Ramdas, S. Lee and J. Woo, “Comparing Regression Models Predicting the Price of Used Cars in Big Data,” 2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Yeosu, Republic of Korea, 2022, pp. 1–4, doi: 10.1109/ICCE-Asia57006.2022.9954633.

[2] A. AlShared, “Used cars price prediction and valuation using data mining techniques,” RIT Digital Institutional Repository, https://repository.rit.edu/theses/11086/ (accessed Dec. 3, 2025).