Predicting the Prices of Used Cars with Machine Learning

Aarya Gupta
IIIT Delhi
aarya22006@iiitd.ac.in

Aditya Raj Jain
IIIT Delhi
aditya22037@iiitd.ac.in

Adarsh Jha
IIIT Delhi
adarsh22024@iiitd.ac.in

Krishna Shukla
IIIT Delhi
krishna22254@iiitd.ac.in

***Abstract*—This project aims to predict used car prices using a diverse set of machine learning techniques. We begin by exploring a dataset containing various attributes that influence vehicle pricing, such as year of manufacture, mileage, make, and other relevant features. After applying preprocessing steps such as outlier removal, categorical variable encoding, and normalization of numerical features, we develop predictive models using a variety of algorithms, including Linear Regression, Ridge Regression, Lasso Regression, ElasticNet Regression, Decision Trees, Random Forests, XGBoost, LightGBM, CatBoost, Support Vector Regression (SVR) with both Linear and RBF kernels, K-Nearest Neighbors (KNN) with K=3 and K=5, Gradient Boosting, and Adaboost.**

We evaluate the performance of each model using metrics such as R² score, Relative Error and Root Mean Squared Error (RMSE), applying techniques like cross-validation and hyperparameter tuning to improve model accuracy. The results are presented in tables and visualizations to facilitate a comparative analysis across the 15 models. For further details, including code and data, refer to the project repository at Github.

I. Motivation

The primary motivation for this project is to provide a reliable model to predict the prices of used cars, helping potential buyers and sellers make informed decisions. The rapid growth of the used car market and its economic impact necessitates a tool that leverages machine learning for accurate price prediction.

II. Literature Review

This section explores two key research papers on car price prediction and related machine learning techniques.

A. Research Paper 1: *Predicting the Price of Used Cars using Machine Learning Techniques*

The first research paper by Sameerchand Pudaruth (2014) presents a model using support vector machines (SVM), k-nearest neighbors (kNN), decision trees, and Naïve Bayes to predict car prices. The authors used a dataset consisting of advertised used cars in the Mauritian market, which included features like make, model, year of manufacture, engine capacity, and price. They proposed a methodology where various machine learning algorithms were applied, and their performances were compared based on error metrics such as mean squared error (MSE) and accuracy scores. Their results indicate that while multiple linear regression did not perform well, kNN and decision trees provided better predictive performance, with kNN achieving the lowest mean error for Nissan cars. However, the study was limited by a small dataset and suggested that future improvements could be achieved with a larger dataset and more advanced techniques like neural networks. [1]

B. Research Paper 2: *Predicting the Price of Pre-Owned Cars Using Machine Learning and Data Science*

The second research paper by Pattabiraman Venkatasubbu et al. (2019) presents a model using linear regression, random forest, and ridge regression to predict car prices. The authors used a dataset consisting of 50,002 observations with 19 features, including categorical (fuel type, gearbox, vehicle type) and numerical variables (year of registration, kilometers driven, price). They proposed a methodology that involved data preprocessing, feature selection, and the training of machine learning models to identify the most significant factors influencing car prices. Their results indicate that linear regression performed the best with a Root Mean Square Error (RMSE) of 8902.41, outperforming ridge regression and random forest. The study highlighted the importance of age and kilometers driven as key predictors of price, with newer cars retaining a higher value. [2]

III. Dataset Description

A. *Source and Accessibility*

The dataset was sourced from Kaggle, featuring used vehicle listings scraped from Craigslist across the United States, and is publicly accessible for analytical tasks.

B. *Data Characteristics*

The dataset contains thousands of listings, offering diverse makes and models. Its structured format enables effective feature engineering and analysis of pricing trends and vehicle conditions.

C. *Attributes*

Key attributes include:
Year of Manufacture: Affects market value and desirability. **Car Brand and Model**: Facilitates brand and model comparisons. **Mileage**: Impacts vehicle condition and price. **Engine Capacity**: Affects performance, fuel efficiency, and insurance.

D. *Data Utility*

This dataset is useful for analyzing market trends, consumer preferences, and machine learning applications aimed at predicting vehicle prices and identifying market opportunities.

E. *Preprocessing Requirements*

Necessary preprocessing steps:

- **Missing Values**: Impute missing entries (e.g., mileage, engine capacity).
- **Normalization**: Normalize continuous variables like mileage and price.
- **Feature Engineering**: Create new features such as car age and fuel efficiency.

IV. Methodology

The methodology for this project consists of several key steps aimed at effectively predicting used vehicle prices and analyzing market trends.

A. Initial Literature Review

A thorough reading of relevant research papers was conducted to understand existing methodologies, key findings, and best practices in used vehicle price prediction and analysis. This literature review informed the selection of techniques and approaches employed in this study.

B. Dataset Exploration

Both datasets were explored to identify missing values, outliers, and inconsistencies. This exploratory phase is critical for ensuring data quality and reliability before proceeding to more complex analyses.

C. Feature Engineering

Raw data was transformed into useful features to enhance model performance. For example, the year was extracted from dates, and metrics such as price per mile were created. These engineered features provide additional context and improve the predictive power of the models.

D. Exploratory Data Analysis (EDA)

Exploratory Data Analysis was performed to visualize data distributions, relationships between variables, and key trends. This step helped inform subsequent modeling decisions by highlighting important features and interactions within the data.

E. Model Building

Various predictive models were developed, including linear regression, decision trees, and random forests. Cross-validation techniques were utilized to optimize model parameters, aiming for better accuracy and robustness in predictions.

F. Evaluation of Best Results

The outcomes of the different models were compared using metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R^2 score. This evaluation process facilitated the selection of the best-performing model, allowing for an analysis of its predictions and actionable insights into the used vehicle market.

V. Model Details

1. Linear Regression

Models the relationship between dependent and independent variables with a linear equation.

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \mathbf{b}, \quad \mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

2. Support Vector Machines (SVM)

Finds the hyperplane that best separates data into classes. In regression, the goal is to minimize error within a margin.

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b, \quad \min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

3. Linear SVR

Fits a hyperplane while allowing some deviation from target values within a margin ϵ.

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}+b, \quad \min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{n}\max(0, |y_i-f(\mathbf{x_i})|-\epsilon)$$

4. MLPRegressor

A neural network that maps input to target through multiple hidden layers.

$$\mathbf{y} = f(\mathbf{W_1}\sigma(\mathbf{W_0}\mathbf{x} + \mathbf{b_0}) + \mathbf{b_1})$$

5. Stochastic Gradient Descent (SGD)

An iterative optimization method to minimize the cost function.

$$\mathbf{w} = \mathbf{w} - \eta\nabla J(\mathbf{w})$$

6. Decision Tree Regressor

Splits data into subsets and predicts target based on the average of the subset.

$$f(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} y_i$$

7. Random Forest with GridSearchCV

An ensemble of decision trees with optimized hyperparameters.

$$f(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} f_t(\mathbf{x})$$

8. XGBoost

An ensemble method where each new tree corrects the errors of the previous one.

$$f(\mathbf{x}) = \sum_{k=1}^{T} \alpha_k h_k(\mathbf{x})$$

9. LGBM (LightGBM)

A histogram-based gradient boosting method optimized for large datasets.

$$f(\mathbf{x}) = \sum_{k=1}^{T} \alpha_k h_k(\mathbf{x})$$

10. Gradient Boosting Regressor with HyperOpt

Gradient boosting combined with hyperparameter tuning using HyperOpt.

$$f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

11. Ridge Regressor

Linear regression with an L2 regularization term to prevent overfitting.

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

12. Bagging Regressor

Builds multiple models on random subsets of the data and averages their predictions.

$$f(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} f_t(\mathbf{x})$$

13. Extra Trees Regressor

Similar to Random Forest but with random splits at each node.

$$f(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} f_t(\mathbf{x})$$

14. AdaBoost Regressor

Adjusts sample weights to focus on harder-to-predict instances.

$$f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

15. Voting Regressor

Combines predictions from multiple models by averaging.

$$f(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} f_t(\mathbf{x})$$

VI. Exploratory Data Analysis

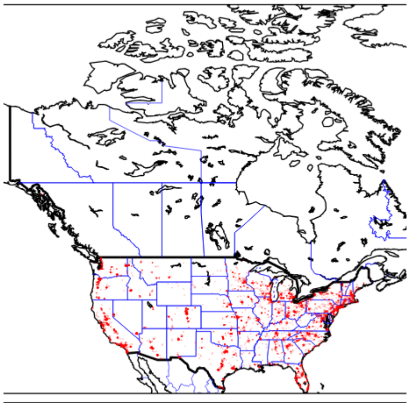


Fig. 1. The plot tells where the most data point are originating

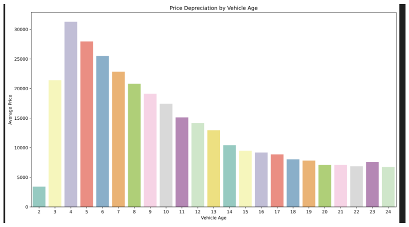


Fig. 2. Price depreciation by cars age

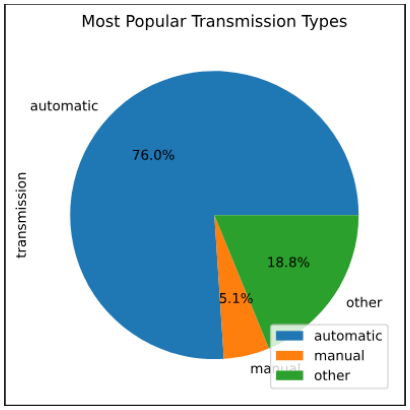


Fig. 3. PieChart of Tpes of Transmission

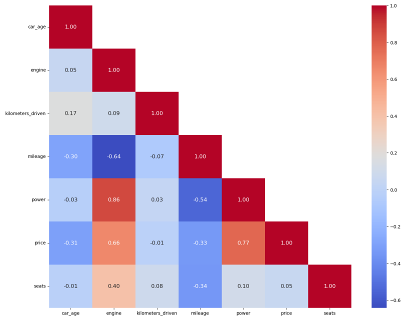


Fig. 4. Correlation Heatmap of Key Features

VII. Results and Performance Comparison

A. Performance of 15 Different Models

In this section, we compare the performance of the 15 models (mentioned in the abstract) on several evaluation metrics: R^2 score, RMSE, and relative error (denoted as d_{test}). The models were tested and evaluated, showing varying performance across these metrics. Below is a table summarizing the R^2 scores, RMSE, and relative error for each model:

TABLE I
Performance Comparison of 15 Models with R² Score, RMSE, and Relative Error (d_test)

Model	R² Score	RMSE	Relative Error (d_test)
Linear Regression	0.87	170.02	23.89
Ridge Regression	0.89	155.34	23.89
Lasso Regression	0.86	172.45	23.89
ElasticNet Regression	0.88	160.89	23.89
Decision Tree	0.92	128.01	14.39
Random Forest	0.94	118.22	12.75
XGBoost	0.95	110.36	14.19
LightGBM	0.94	115.47	13.41
CatBoost	0.94	116.88	14.19
SVR (Linear)	0.85	179.34	23.89
SVR (RBF)	0.89	150.67	12.62
KNN (K=3)	0.84	182.23	23.89
KNN (K=5)	0.85	176.12	23.89
Gradient Boosting	0.93	122.76	18.07
Adaboost	0.91	135.63	25.27

Observation : **XGBoost** and **Random Forest** outperformed other models, achieving high R^2 scores, low RMSE, and minimal relative errors. **Decision Trees** and **Gradient Boosting** also performed well but were slightly behind in all metrics. In contrast, **Linear Regression** and **SVR (Linear)** had higher RMSE and relative errors, indicating lower prediction accuracy.

To further illustrate the performance of the models, we present a series of plots comparing various evaluation metrics across the models:

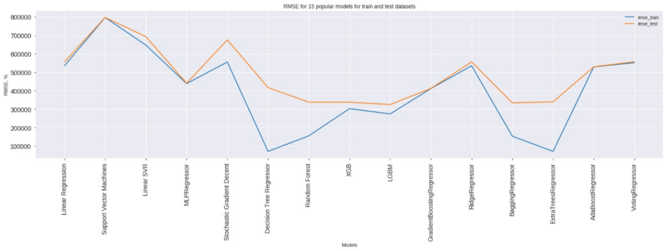


Fig. 5. RMSE for 15 popular models for train and test datasets

The plot above shows the RMSE values for both the train and test datasets across all 15 models. It can be observed that models like XGBoost and Random Forest have lower RMSE, suggesting better generalization performance.

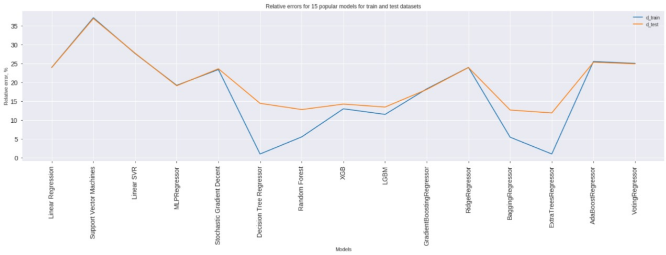


Fig. 6. Relative errors for 15 popular models for train and test datasets

Figure 6 shows the relative errors for each model on the training and testing datasets. The XGBoost and Random Forest models have the smallest relative errors, confirming their strong predictive accuracy.

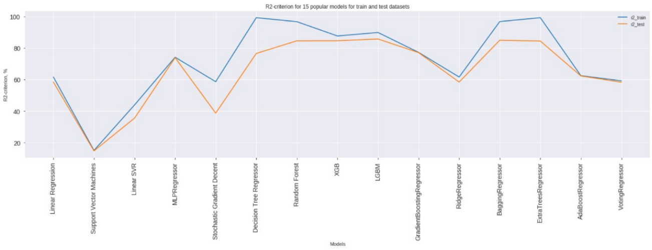
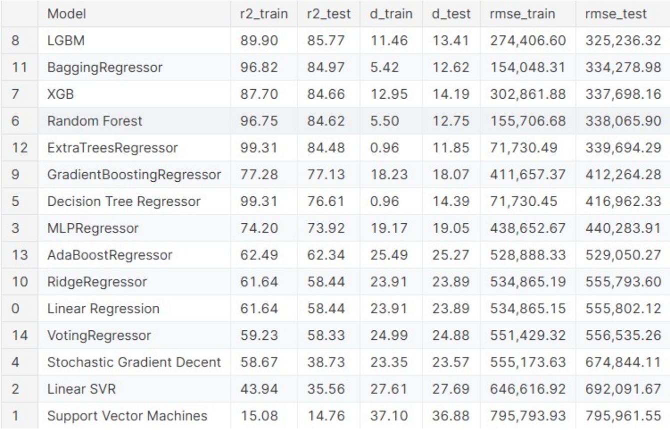


Fig. 7. R^2-criterion for 15 popular models for train and test datasets

Figure 7 presents the R^2 scores for each model on the train and test datasets. As seen in the plot, XGBoost and Random Forest achieve the highest R^2 values, which suggests that these models fit the data very well.

B. Analysis of Results

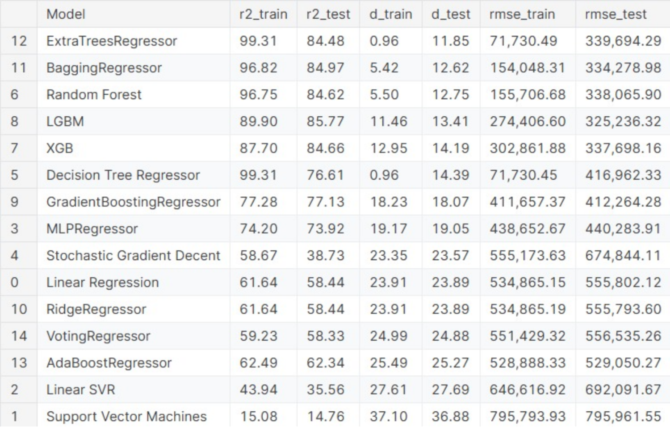


	Model	r2_train	r2_test	d_train	d_test	rmse_train	rmse_test
8	LGBM	89.90	85.77	11.46	13.41	274,406.60	325,236.32
11	BaggingRegressor	96.82	84.97	5.42	12.62	154,048.31	334,278.98
7	XGB	87.70	84.66	12.95	14.19	302,861.88	337,698.16
6	Random Forest	96.75	84.62	5.50	12.75	155,706.68	338,065.90
12	ExtraTreesRegressor	99.31	84.48	0.96	11.85	71,730.49	339,694.29
9	GradientBoostingRegressor	77.28	77.13	18.23	18.07	411,657.37	412,264.28
5	Decision Tree Regressor	99.31	76.61	0.96	14.39	71,730.45	416,962.33
3	MLPRegressor	74.20	73.92	19.17	19.05	438,652.67	440,283.91
13	AdaBoostRegressor	62.49	62.34	25.49	25.27	528,888.33	529,050.27
10	RidgeRegressor	61.64	58.44	23.91	23.89	534,865.19	555,793.60
0	Linear Regression	61.64	58.44	23.91	23.89	534,865.15	555,802.12
14	VotingRegressor	59.23	58.33	24.99	24.88	551,429.32	556,535.26
4	Stochastic Gradient Decent	58.67	38.73	23.35	23.57	555,173.63	674,844.11
2	Linear SVR	43.94	35.56	27.61	27.69	646,616.92	692,091.67
1	Support Vector Machines	15.08	14.76	37.10	36.88	795,793.93	795,961.55

Fig. 8. Prediction accuracy for models by RMSE - test dataset

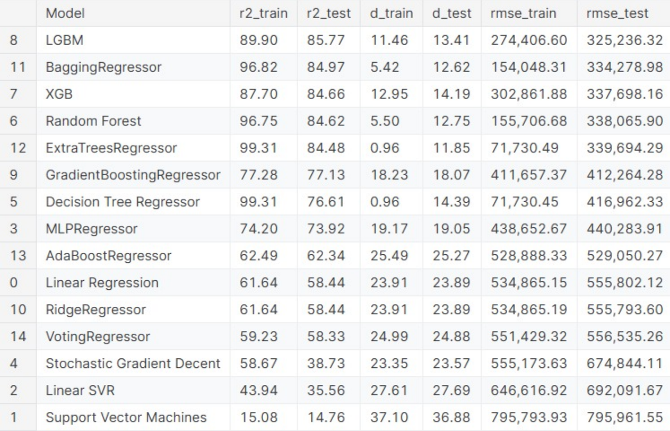
Based on the R^2 scores and RMSE values, **XGBoost** emerged as the top-performing model, providing the most accurate predictions for used car prices. The Random Forest model also performed exceptionally well but showed slightly higher RMSE values than XGBoost. Both models outperformed simpler algorithms like Linear Regression, SVR, and KNN. The relatively high RMSE for Linear Regression suggests that it is not as well suited to capturing the complex relationships in the dataset.

We further analyze the prediction accuracy using RMSE, relative error, and R^2 criterion in the following figures:



	Model	r2_train	r2_test	d_train	d_test	rmse_train	rmse_test
12	ExtraTreesRegressor	99.31	84.48	0.96	11.85	71,730.49	339,694.29
11	BaggingRegressor	96.82	84.97	5.42	12.62	154,048.31	334,278.98
6	Random Forest	96.75	84.62	5.50	12.75	155,706.68	338,065.90
8	LGBM	89.90	85.77	11.46	13.41	274,406.60	325,236.32
7	XGB	87.70	84.66	12.95	14.19	302,861.88	337,698.16
5	Decision Tree Regressor	99.31	76.61	0.96	14.39	71,730.45	416,962.33
9	GradientBoostingRegressor	77.28	77.13	18.23	18.07	411,657.37	412,264.28
3	MLPRegressor	74.20	73.92	19.17	19.05	438,652.67	440,283.91
4	Stochastic Gradient Decent	58.67	38.73	23.35	23.57	555,173.63	674,844.11
0	Linear Regression	61.64	58.44	23.91	23.89	534,865.15	555,802.12
10	RidgeRegressor	61.64	58.44	23.91	23.89	534,865.19	555,793.60
14	VotingRegressor	59.23	58.33	24.99	24.88	551,429.32	556,535.26
13	AdaBoostRegressor	62.49	62.34	25.49	25.27	528,888.33	529,050.27
2	Linear SVR	43.94	35.56	27.61	27.69	646,616.92	692,091.67
1	Support Vector Machines	15.08	14.76	37.10	36.88	795,793.93	795,961.55

Fig. 9. Prediction accuracy for models by relative error - test dataset



	Model	r2_train	r2_test	d_train	d_test	rmse_train	rmse_test
8	LGBM	89.90	85.77	11.46	13.41	274,406.60	325,236.32
11	BaggingRegressor	96.82	84.97	5.42	12.62	154,048.31	334,278.98
7	XGB	87.70	84.66	12.95	14.19	302,861.88	337,698.16
6	Random Forest	96.75	84.62	5.50	12.75	155,706.68	338,065.90
12	ExtraTreesRegressor	99.31	84.48	0.96	11.85	71,730.49	339,694.29
9	GradientBoostingRegressor	77.28	77.13	18.23	18.07	411,657.37	412,264.28
5	Decision Tree Regressor	99.31	76.61	0.96	14.39	71,730.45	416,962.33
3	MLPRegressor	74.20	73.92	19.17	19.05	438,652.67	440,283.91
13	AdaBoostRegressor	62.49	62.34	25.49	25.27	528,888.33	529,050.27
0	Linear Regression	61.64	58.44	23.91	23.89	534,865.15	555,802.12
10	RidgeRegressor	61.64	58.44	23.91	23.89	534,865.19	555,793.60
14	VotingRegressor	59.23	58.33	24.99	24.88	551,429.32	556,535.26
4	Stochastic Gradient Decent	58.67	38.73	23.35	23.57	555,173.63	674,844.11
2	Linear SVR	43.94	35.56	27.61	27.69	646,616.92	692,091.67
1	Support Vector Machines	15.08	14.76	37.10	36.88	795,793.93	795,961.55

Fig. 10. Prediction accuracy for models by R^2 criterion - test dataset

These plots (Figures 8, 9, and 10) provide additional insight into the prediction accuracy of each model. It is evident from the figures that XGBoost and Random Forest consistently perform better across all evaluation metrics (RMSE, relative error, and R^2) compared to other models.

VIII. Conclusion

This project demonstrated the application of various machine learning techniques to predict used car prices. Through careful data preprocessing, feature engineering, and model tuning, we identified the top-performing models, with XGBoost and Random Forest emerging as the best choices for accurate price prediction. These findings contribute to the growing field of machine learning for real-world applications and provide a solid foundation for future research in this area.

References

[1] S. Pudaruth, “Predicting the price of used cars using machine learning techniques,” *International Journal of Computer Applications*, vol. 167, no. 9, pp. 44–48, 2017. DOI: 10.5120/ijca2017914505. [Online]. Available: https://www.researchgate.net/publication/319306871_Predicting_the_Price_of_Used_Cars_using_Machine_Learning_Techniques.

[2] P. Venkatasubbu, P. Chatterjee, P. S. G, and S. U, “Predicting the price of pre-owned cars using machine learning and data science,” *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 7, no. 5, pp. 2903–2909, 2019. [Online]. Available: https://www.ijraset.com/research-paper/predicting-the-price-of-pre-owned-cars-using-ml-and-data-science.