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 machine learning techniques. We begin by exploring the dataset to identify key attributes that influence vehicle pricing, such as year of manufacture, mileage, and make. After applying preprocessing methods to clean and prepare the data—such as filtering outliers, encoding categorical variables, and normalizing numerical features—we develop predictive models using Linear Regression, Ridge Regression, Decision Trees, and Random Forests. We evaluate each model's performance using metrics like R² score and Root Mean Squared Error (RMSE), employing cross-validation and hyperparameter tuning to enhance accuracy. The results are presented in tables and visualizations, facilitating a comparative analysis of the models.**

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 utilized for this project was sourced from Kaggle, where it features a comprehensive collection of used vehicle listings scraped from Craigslist across the United States. This rich dataset is accessible online, providing a valuable resource for various analytical tasks.

B. *Data Characteristics*

The dataset encompasses thousands of listings, reflecting a diverse array of makes and models. Its structured format facilitates effective feature engineering, enabling deeper insights into pricing trends and vehicle conditions across different regions.

C. *Attributes*

The dataset includes several key attributes that provide critical insights into used vehicle listings. The **Year of Manufacture** indicates when each vehicle was produced, which can significantly influence its market value and desirability. The **Car Brand and Model** specifies the manufacturer and model of the vehicle, allowing for comparisons across different brands and types. **Mileage** measures how much the vehicle has been driven, typically affecting its condition and price; lower mileage often correlates with a higher resale value. Finally, **Engine Capacity** denotes the size of the vehicle's engine, which can impact performance, fuel efficiency, and insurance costs. Together, these attributes enable comprehensive analysis and comparisons within the used vehicle market.

D. *Data Utility*

This dataset is particularly useful for analyzing market trends and consumer preferences within the used vehicle market. Its rich feature set supports detailed comparisons and is ideal for machine learning applications aimed at predicting vehicle prices and identifying market opportunities.

E. *Visualization*

Visualization of the dataset aids in understanding the distribution of various features, as demonstrated in the following figure:

F. Preprocessing Requirements

The dataset requires the following preprocessing steps:

- **Missing Values:** Imputation for missing entries in certain fields like mileage and engine capacity.
- **Normalization:** Continuous variables such as mileage and price need normalization.
- **Feature Engineering:** Creating new features like age of the car and fuel efficiency metrics.

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

A. Data Collection & Preprocessing

- We obtained this dataset from Kaggle, after removing the features which had more than 40% of missing data and ended up with 14 parameters for training the model.
- Initially, there were 117169 data points. After performing preprocessing (such as removing extreme outliers and feature engineering), we are left with 45268 data points.
- We have performed standardization using standard scalar to perform feature scaling (which is necessary in models such as Ridge.
- We are converting categorical features to numerical features, using Label encoder from scikit-learn.

B. Data Splitting

- The dataset is divided into training and testing sets (e.g., 70% for training, 10% for testing and 10% for validation).
- We are performing 5 fold Cross Validation, to get a proper bound of error for each model.

TABLE I
STATISTICAL SUMMARY OF VEHICLE DATASET

Statistic	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
year	114.75	2.56	111.00	113.00	115.00	117.00	122.00
manufacturer	18.61	11.23	0.00	10.00	14.00	29.00	41.00
condition	1.21	1.18	0.00	0.00	2.00	2.00	5.00
cylinders	4.45	1.25	0.00	3.00	5.00	5.00	7.00
fuel	2.00	0.48	0.00	2.00	2.00	2.00	4.00
odometer	16.19	11.40	0.00	7.00	16.00	23.00	469.00
transmission	0.39	0.77	0.00	0.00	0.00	0.00	2.00
drive	0.75	0.75	0.00	0.00	1.00	1.00	2.00
type	6.11	4.05	0.00	2.00	8.00	9.00	12.00
paint_color	5.66	4.10	0.00	1.00	8.00	10.00	11.00
state	23.90	14.90	0.00	9.00	23.00	37.00	50.00

C. Model Selection

Evaluation Metrics: For each model, we have calculated **R2 score, Relative error** & **RMSE** for test & train data.

- **Linear Regression**: To establish a baseline model due to its simplicity and interpretability.
- **Ridge Regressor**: To handle multicollinearity among features and regularize the model, preventing overfitting.
- **Decision Tree Regressor**: To capture non-linear relationships and interpret feature importance.
- **Random Forest**: To improve prediction accuracy and generalization by averaging the outputs of multiple decision trees.

D. Linear Regression

- **Purpose of Linear Regression**: Models the relationship between independent variables (e.g., car age, mileage) and a dependent variable (car prices) using a linear combination of features.

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

  [3]
- **Model Training**: The fit() function estimates feature coefficients, showing how each feature impacts the target variable (price).
- **Prediction and Evaluation**: The acc_model function computes R², relative error, and RMSE to evaluate model performance on training and testing datasets.

- **Strengths**: Simple to implement and interpret, reveals each feature's influence on the target through coefficients. **Limitations**: Struggles with non-linear relationships and is sensitive to multicollinearity.

E. Ridge Regressor

- **Purpose of Cross-Validation**: Analyze how cross-validation folds impact Ridge Regression performance (mean RMSE) to find the optimal CV value for better generalization.
- **Feature Engineering**: Modifying features like odometer can improve model performance by providing more meaningful input data.
- **Defining Target and Features**: Clearly separating the target variable from the features facilitates training the model effectively.

F. Decision Tree Regressor

- **Purpose of Decision Tree Regressor**: It models the relationship between input features (e.g., age, mileage, brand) and a continuous target variable (car price) by creating a tree structure. Each node represents a condition on a feature, and the leaves correspond to predicted car prices. [4]
- **Training the Model**: The model is trained using the training dataset, where the fit() function finds the optimal splits at each node. These splits aim to reduce the variance in the target variable (car prices), improving the accuracy of predictions at the leaf nodes.
- **Key Characteristics**: Decision Trees effectively capture non-linear relationships and complex interactions between variables. However, they tend to overfit when too deep, performing well on training data but often failing to generalize to unseen testing data. Pruning or setting maximum depth limits can help mitigate overfitting.

G. Random Forest Regressor

- **Purpose of Random Forest Regressor**: Random Forests combine multiple Decision Trees to improve prediction accuracy. Each tree uses a subset of data, and the final prediction averages the trees' outputs, reducing overfitting and enhancing stability.
- **Hyperparameter Tuning with GridSearchCV**: GridSearchCV tests various values of n_estimators (number of trees) with 5-fold cross-validation. It identifies the best value of n_estimators by maximizing the R² score, which measures explained variance.
- **Model Training and Selection**: After GridSearchCV finds the optimal n_estimators, the Random Forest model is trained using the fit() function on the training data for more accurate predictions.
- **Advantages of Using Random Forest**:
  Robustness: Less prone to overfitting than individual Decision Trees, making it suitable for complex regression tasks like car price prediction.
  Feature Importance: Offers insights into which features influence the target variable most.
  Scalability: Flexible and can be fine-tuned for various dataset sizes and computational resources by adjusting n_estimators.

VI. Exploratory Data Analysis

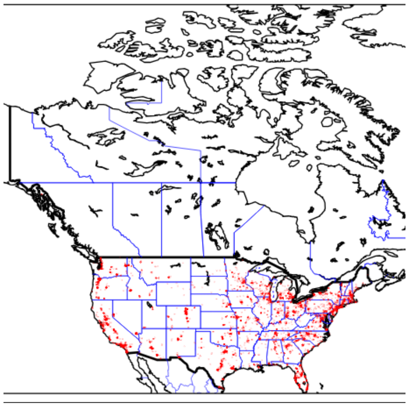

Fig. 1. The plot tells where the most data point are originating

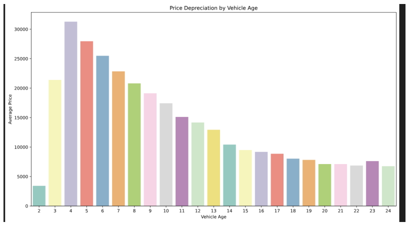

Fig. 2. Price depreciation by cars age

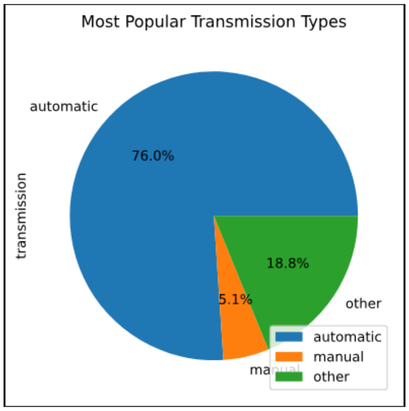

Fig. 3. PieChart of Tpes of Transmission

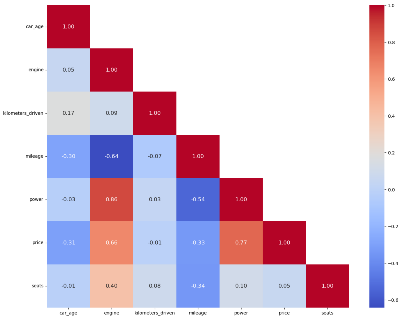

Fig. 4. Correlation Heatmap of Key Features

VII. Results

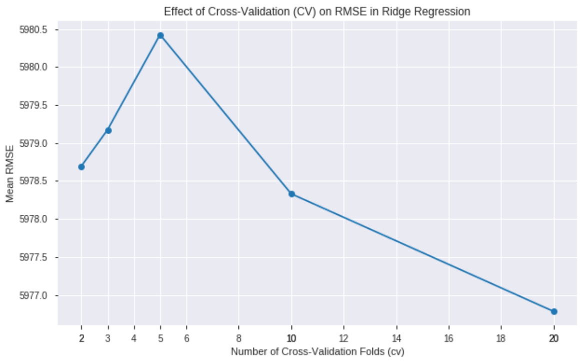

Fig. 5. Trend of RMSE with varying CV folds illustrating the bias-variance trade-off in Ridge Regression.

Effect of Cross-Validation (CV) on RMSE in Ridge Regression

Few CV Folds (2–5)::

- **Higher bias, lower variance:** Larger training sets improve performance, but smaller validation sets may lead to overfitting and higher RMSE.

Optimal CV Folds (around 10)::

- **Balanced bias and variance:** A suitable mix of training and validation sizes enhances generalization, resulting in decreased RMSE.

Many CV Folds (20+)::

- **Lower bias, higher variance:** Smaller training sets increase variance, causing fluctuations in performance and rising RMSE.

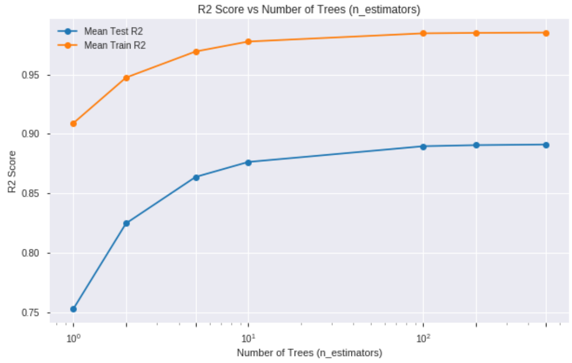

Fig. 6. R^2 Score vs Number of Trees (n_estimators)

R^2 Score vs Number of Trees (n_estimators)

The following key insights can be drawn from the plot:

- **Training R^2 Score (orange line)**: Rapid increase with the number of trees, plateauing above 0.95 after around 50 trees, indicating effective fitting but diminishing returns beyond this point.
- **Test R^2 Score (blue line)**: Increases more slowly, leveling off between 0.88 and 0.90 after about 30 trees, suggesting good generalization with sufficient model complexity.
- **Overfitting Consideration**: The gap between training and test R^2 scores indicates better fitting to training data, but high test scores show a balance between model complexity and generalization.

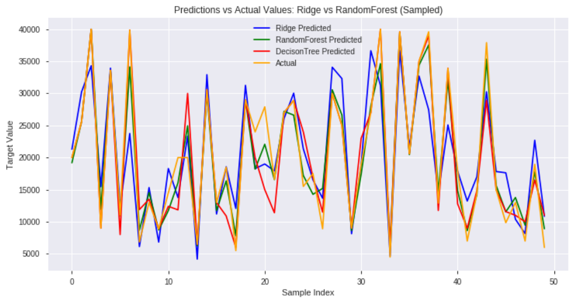

Fig. 7. Predictions vs Actual Values: Ridge vs Random Forest vs Decision Tree (Sampled on 100 Test points)

The following key observations can be made:

- **Random Forest (green line)**: Follows actual values closely, particularly at peaks and troughs, indicating better capture of non-linear patterns and superior generalization compared to Ridge Regression.
- **Decision Tree (red line)**: Aligns well with actual values but exhibits more fluctuations, suggesting overfitting. It fits data points more precisely than Ridge Regression but may generalize less effectively than Random Forest.
- **General Performance**: Random Forest strikes the best balance between flexibility and generalization, followed by the Decision Tree, while Ridge Regression struggles with data complexity.

TABLE II
Performance Comparison of Models

Model	Metric	Train (%)	Test (%)	Diff. (%)
Linear Regression	R^2 Score	67.32	68.24	0.92
	Relative Error	22.38	22.29	-0.09
Ridge Regression	R^2 Score	67.32	68.24	0.92
	Relative Error	22.38	22.29	-0.09
Decision Tree	R^2 Score	100.0	79.71	-20.29
	Relative Error	0.0	13.31	13.31
Random Forest	R^2 Score	98.61	90.15	-8.46
	Relative Error	3.75	10.18	6.43

A. *Conclusion*

We learned how to implement models in real-life scenarios and delved into the exciting world of machine learning. We faced challenges during implementation and focused on creating beautiful visualizations at each step.

All team members participated in meetings where we discussed various topics, and work was equally distributed among us. Aarya Gupta (Literature Review), Aditya Raj Jain (Model Details), Adarsh Jha (Results), Krishna Shukla (EDA)

We need to apply more models and find the best one. We also need to focus more on model-building with Indian datasets.

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.

[3] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.