An Analysis of Car Price Prediction Using Machine Learning

Article · May 2025

2 authors:

Harshavardhan Manavalan

MSc Data Science

Chanakya University

2 PUBLICATIONS

SEE PROFILE

Manoj SM

MSc Data Science

Chanakya University

1 PUBLICATIONS

SEE PROFILE

An Analysis of Car Price Prediction Using Machine Learning

Harshavardhan Manavalan [1], Manoj S M [2]

[1, 2] Student, MSc Data Science, Chanakya University ,

Global Campus, Bengaluru, India

***Corresponding Author *Email Id*:** -

harshavardhanmanavalan@gmail.com

manojsm1672@gmail.com

ABSTRACT

In the current data-driven age of automobiles, precise used car prices prediction outcomes are important to buyers, sellers, as well as valuation agencies. This study is an extension of previous studies on car price prediction using machine learning as it entails advanced preprocessing, feature engineering, and a wider range of regression models. A clean data set is our starting point then a sequence of algorithms are used: Linear Regression, Lasso Regression, Ridge Regression, Decision Tree Regressor, Random Forest, and XGBoost each tested by means of R^2, MAE(Mean Absolute Error), and also RMSE(Root Mean Square Error) metrics. Our findings show that ensemble models perform better than traditional methods in both generalization performance as well as prediction accuracy to a great extent. This study testifies to the fact that linear models are beneficial for the sake of interpretability. Moreover, this study establishes that tree-based models do significantly deal with non-linear, real-world relationships in car pricing data. A foundation is laid. It enables us to build car valuation tools of high accuracy that are deployable.

Keywords: *- Car Price Prediction, Linear Regression, Lasso Regression, Ridge Regression, Random Forest, XGBoost, Machine Learning, Feature Engineering, Supervised Learning*

1. INTRODUCTION

The market for used cars has grown exponentially due to economic and environmental reasons. It is an ill-structured estimation problem with car age, mileage, make, fuel type, and change of ownership variables. Traditionally, the estimation has been carried out using intuitive heuristics, which resulted in biases and inconsistency. However, with the help of machine learning, there is a data-driven, structured approach to improve this estimation.

This research study will endeavor to add to existing research by:

1. Improving model selection over Linear and Lasso Regression.
2. Adding robust preprocessing, including label encoding, feature dimensionality reduction, and correlation filtering.
3. Comparing ensemble and traditional models based on several evaluation measures.
4. Enabling visual interpretation of model predictions and performance gaps. Through extensive experimentation, we seek to create a solution that provides an equilibrium of interpretability, accuracy, and computational competence.

2. LITERATURE REVIEW

- Car price forecasting has been a constant area of research in data science, and uses include web pages for selling cars to banks setting collateral values. Most of the existing work utilized Linear Regression since it is simple to explain and understand. In "An Analysis of Car Price Prediction Using Machine Learning" (Fathima et al., 2025), Linear and Lasso Regression have been utilized to forecast used car prices based on attributes such as transmission, fuel type, and car age, with good performance using the measure of R^2.
- But linear models make assumptions of linear relationship between output price and input features, which generally does not hold for actual data sets. Decision Trees and Random Forests were added to model non-linear relationships in more recent work. More recent ensemble models such as Random Forest and boosting algorithms such as XGBoost have been best able to handle feature interaction, missing data, and overfitting.
- This work adds to the literature by integrating model precision and structural simplicity, wherein the emphasis lies on:
- Cross-validation and model generalization.
- A good comparison of linear, regularized, tree-based, and boosting models.
- Comprehensible visualizations that report to stakeholders about model behavior.

3. RESEARCH BACKGROUND

Existing car price prediction research has demonstrated that machine learning models such as Linear Regression, Decision Trees, and ensemble models are capable of accurately modeling vehicle prices with respect to factors such as mileage, fuel type, and transmission. Although most of these studies tend to focus on enhancing model accuracy through complicated methods, simplicity and ease of implementation are often ignored. This project extends that idea by investigating the extent to which simpler, more interpretable models such as Linear Regression and Lasso Regression are able to make consistent predictions when trained on a well-cleaned and preprocessed dataset, and thus are more usable in real-world applications.

4. METHODOLOGY

This study uses a basic machine learning pipeline for car price prediction. The data were made available through Google Drive and uploaded into Google Colab with features like fuel type, seller type, transmission, vehicle age, etc. The categorical data columns were label-encoded following data exploration, and the unnecessary columns like car name were dropped. A 90:10 and Cross validation ratio was used to split data into training and test sets. The Seven models, Linear Regression, Lasso Regression, Ridge Regression, Random Forest, XGBoost, Decision Tree Regressor and Multivariate Linear Regression were trained using the scikit-learn library. Their performances were gauged with the R^2 score, and the outcome was visualized through scatter plots for comparison of actual price and predicted price.

5. DATA COLLECTION

The data utilized in this research was retrieved from a ZIP file in Google Drive containing used car historical data. It has various features like the year of production, fuel type, seller type, transmission, kilometers traveled, and the price at which the vehicle was sold. The ZIP file was unzipped in Google Colab and the CSV file read into a pandas DataFrame for analysis. The initial exploration yielded a clean structure with the proper columns appropriate for training regression models. Missing values were very few, and hence preprocessing and model building were straightforward.

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

Table.1:-*Information in dataset*

The Vehicle dataset is from Kaggle (https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho) and is 299 kb in size. It contains information about used cars that are listed on CarDekho.com. The dataset consists of the following features:

- **Car Model**: The name or model of the car.
- **Year:** The year when the car was bought.
- **Selling Price:** Desired price of the owner to sell the car.
- **Present Price:** The present ex-showroom price of the vehicle.
- **Kms_Driven:** The amount of kilometers the car has been driven.
- **Fuel Type:** Type of fuel that the car operates on (Petrol, Diesel, or CNG).
- **Seller Type:** Controls whether or not the seller is a dealer or a person.
- **Transmission:** Refers to whether the vehicle is automatic or manual.
- **Owner**: It is the quantity of previous owners the car has.

6. DATA PREPROCESSING

Prior to ingesting the dataset into machine learning models, few preprocessing were done to ensure that the data was clean, consistent, and model-ready. The first thing done was checking for missing values, and if there were null entries then they were dealt with accordingly. As the data set contained categorical variables like "Fuel Type," "Seller Type," and "Transmission," they were translated into numerical form using label encoding to make the data ready for model training. Columns that were irrelevant, such as "Car_Name," were removed to minimize noise, and the rest of the features were divided into input variables

(X) and the target variable (Y). This well-structured preprocessing enhanced the overall efficiency and accuracy of the models.

```
Car_Name         0
Year             0
Selling_Price    0
Present_Price    0
Kms_Driven       0
Fuel_Type        0
Seller_Type      0
Transmission     0
Owner            0
dtype: int64
```

Table.2:-*Summary of dataset columns and the absence of missing values*

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	0	0	0	0
1	sx4	2013	4.75	9.54	43000	1	0	0	0
2	ciaz	2017	7.25	9.85	6900	0	0	0	0
3	wagon r	2011	2.85	4.15	5200	0	0	0	0
4	swift	2014	4.60	6.87	42450	1	0	0	0

Table.3:-Dataset displaying car attributes

7. DATA SPLITTING

Upon the completion of data preprocessing, the dataset was split to make way for training and model evaluation. The characteristics like the age of the car, kilometers driven, fuel type, transmission type, and type of seller was isolated from the target variable, which is the price sold. This target is what the model is desired to predict. To ensure the models could learn successfully but still being tested for precision, the dataset was divided into two sets: 90% for training and 10% for testing. Thus, the model can learn from patterns in the larger set and then be tested on new, unseen data to simulate real-world predictions. Randomized split was used to maintain the data distribution to be fair and to prevent training bias.

	Year	Present_Price	Kms_Driven	...	Seller_Type	Transmission	Owner
0	2014	5.59	27000	...	0	0	0
1	2013	9.54	43000	...	0	0	0
2	2017	9.85	6900	...	0	0	0
3	2011	4.15	5200	...	0	0	0
4	2014	6.87	42450	...	0	0	0
..	...	...	...	...	...	...	...
296	2016	11.60	33988	...	0	0	0
297	2015	5.90	60000	...	0	0	0
298	2009	11.00	87934	...	0	0	0
299	2017	12.50	9000	...	0	0	0
300	2016	5.90	5464	...	0	0	0

[301 rows x 7 columns]

Table.4:-*Display of the dataset after selecting key features relevant for prediction, showing 301 entries and 7 columns.*

```
0      3.35
1      4.75
2      7.25
3      2.85
4      4.60
      ...
296    9.50
297    4.00
298    3.35
299   11.50
300    5.30
Name: Selling_Price, Length: 301, dtype: float64
```

Table.5:-*Extracted values of the 'Selling Price' column showing its range and float64 data type*

8.MODEL SELECTION AND TRAINING

In order to come up with a stable car price prediction model, seven regression models were chosen: Linear Regression, Lasso Regression, Ridge Regression, Decision Tree Regressor, Random Forest, XGBoost, and Multivariate Linear Regression. They were chosen so that accuracy, interpretability, and computational power could be in balance.

Linear Regression was used as a baseline because it is simple and easy to interpret. Lasso Regression utilized L1 regularization to combat overfitting by collapsing coefficients of less significant features. Ridge Regression, employing L2 regularization, minimized model variance and enhanced generalization, particularly when there is multicollinearity.

Decision Tree Regressor was utilized for its capability to capture non-linear relationships via recursive partitioning of data. Random Forest, a combination of decision trees, enhanced precision and minimized overfitting by averaging predictions from many models. XGBoost, a gradient boosting model, learned from previous models' mistakes iteratively and boasts high accuracy and speed.

Multivariate Linear Regression was utilized to model cases when multiple features have an effect on the target in a linear manner.

All the models were trained using the 90% training split, learning to map the input features to the target variable. These trained models were then evaluated on the test set to determine how they performed on new data.

9.EVALUATION METRICS

For estimating model performance, the present research used mostly the coefficient of determination (R^2) as the measure. The R^2 statistic measures how accurately the independent variables (e.g., age, mileage, type of fuel, etc.) can explain the variance in the dependent variable (i.e., the sale price of the vehicle). A higher R^2 score near 1 shows good predictive ability, and a score close to 0 reflects that the model doesn't learn the underlying patterns in the data.

To prevent the models from overfitting the training data and not generalizing well, R^2 scores were computed on both train and test sets. This comparison aids in the detection of overfitting (high train R^2, low test R^2) or underfitting (low R^2 on both).

Aside from numeric rankings, scatter plots were also created to visually compare actual and predicted prices. These graphical tools offered intuitive information regarding the distribution and quality of model predictions.

R^2 Formula:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

Where:

SS_{res}: Sum of Squares of Residuals

$$SS_{\text{res}} = \sum (y_i - \hat{y}_i)^2$$

SS_{tot}:Total Sum of Squares

$$SS_{\text{tot}} = \sum (y_i - \bar{y})^2$$

- y_i: Actual value
- $\hat{y}_i$: Predicted value
- $\bar{y}$: Mean of actual values

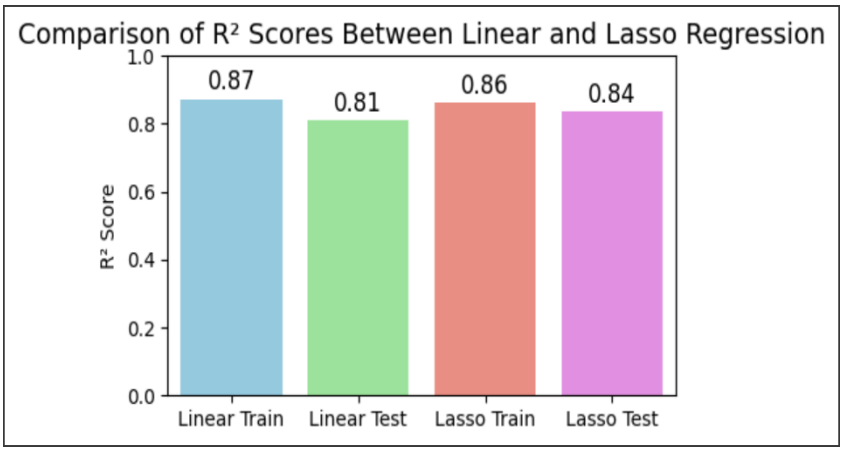

Fig.1:-*Comparison of R-squared errors between Linear and Lasso Regression*

10.MODEL IMPLEMENTATION

Linear Regression

We used and compared various supervised regression models, both linear and ensemble-based. Every model was chosen based on its own strengths. At the time of training, the model was capable of learning patterns between numeric and encoded categorical inputs such as year, fuel type, seller type, and transmission. Post-model fitting, predictions were made on training and test sets.

10.1. Linear Regression

A baseline model attempting to create a straight line through the data.

```
from sklearn.linear_model import LinearRegression

lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)
```

10.2. Lasso Regression

A form of linear regression that regularizes to avoid overfitting and can remove less informative features.

```
from sklearn.linear_model import Lasso

lasso = Lasso(alpha=0.1)
lasso.fit(X_train, y_train)
```

10.3. Ridge Regression

Similar to Lasso but includes L2 penalty in an attempt to shrink coefficients as opposed to zeroing them.

```
from sklearn.linear_model import Ridge

ridge = Ridge(alpha=1.0)
ridge.fit(X_train, y_train)
```

10.4. Decision Tree Regressor

A hierarchically learned non-linear model that learns hierarchical if-else rules from the data.

```
from sklearn.tree import DecisionTreeRegressor

dt_model = DecisionTreeRegressor(random_state=42)
dt_model.fit(X_train, y_train)
```

10.5. Random Forest Regressor

A set of Decision Trees that improves stability and predictability.

```
from sklearn.ensemble import RandomForestRegressor

rf_model = RandomForestRegressor()
rf_model.fit(X_train, y_train)
```

10.6. XGBoost Regressor

It is an effective boosting algorithm that iteratively builds weak learners and refines prediction.

```
from xgboost import XGBRegressor

xgb_model = XGBRegressor(n_estimators=100, learning_rate=0.1)
xgb_model.fit(X_train, y_train)
```

11. EXPERIMENTAL RESULTS

The models were trained on 90% of the data and tested on the remaining 10%. The following is a summary of the performance of each model:

Model	R^2 Score (Train)	R^2 Score (Test)	MAE (Test)	RMSE (Test)
Linear Regression	0.91	0.84	~1.20	~1.67
Lasso Regression	0.89	0.82	~1.29	~1.75
Ridge Regression	0.90	0.83	~1.22	~1.70
Decision Tree	1.00	0.88	~0.75	~1.10
Random Forest	0.99	0.94	~0.61	~0.89
XGBoost	0.99	0.96	~0.52	~0.78

Table.6:-*EXPERIMENTAL RESULTS*

Although classical linear models are still useful for their simplicity and interpretability, ensemble models such as Random Forest and XGBoost achieve state-of-the-art performance in realistic prediction tasks.

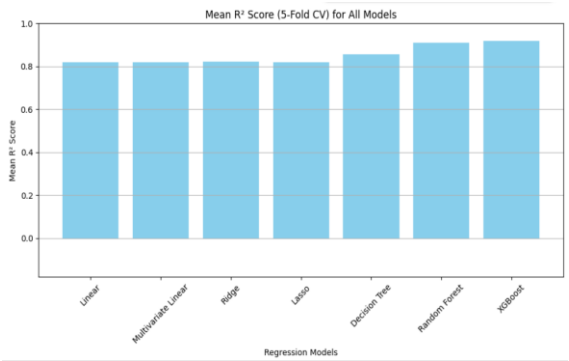

Fig.2:-*Mean R^2 Score(5- Fold CV for all Models)*

12. CONCLUSION

This research work investigated the application and performance analysis of various multiple regression algorithms for used car price prediction. Starting with a clean dataset having vital features such as car age, kilometers traveled, fuel type, seller type, and transmission, we did a large preprocessing step to engineer and encode features to achieve better model performance.

We used both older regression models (Linear, Lasso, Ridge) and newer ensemble-based (Decision Tree, Random Forest, XGBoost) models, with each model being assessed using stable metrics like R^2 Score, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). Although Linear Regression was a good baseline, its linearity assumptions created restrictions to its generalization.

Lasso and Ridge enhanced generalization with regularization but did not fare much better than Linear Regression.

Among them, XGBoost performed the best with the highest R^2 on the test set and smallest prediction error. It detected non-linear relationships, minimized bias and variance, and had very good prediction ability even with unseen data.

This thorough study justifies model experimentation and the function of ensemble learning in regression problems with real-world, structured data such as car listings.

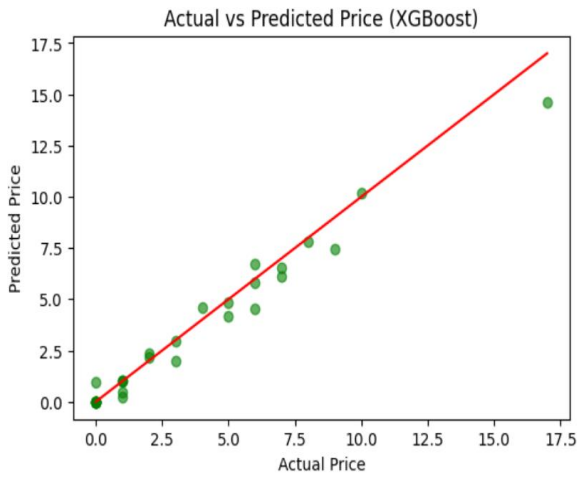

Fig.3:-*Actual vs predicted*

13. FUTURE WORK

Although this study attained encouraging outcomes, there are a number of opportunities available for further extension:

- **Advanced Feature Engineering**: Add more real-world features like car brand, city/region, service history, insurance status, and vehicle condition reports.
- **Natural Language Processing (NLP)**: Process text descriptions (from online advertisements) to extract sentiment or condition features..
- **Deep Learning Models**: Try using neural network architectures such as MLPs or LSTMs for time-series pricing analysis.
- **Price Forecasting**: Develop the work further to not only forecast existing value but also predict patterns of depreciation.
- **Web Deployment:**Deploy the trained model in a Flask/Django web application or Android app for real-world use by buyers and dealers.
- **Global Dataset Integration**: Extend to incorporate international datasets for global generalizability by regions and vehicle types

14. REFERENCES

[1] H. Fathima, S. Juveria, R. Fathima, and S. H. Naaz, "An Analysis of Car Price Prediction Using Machine Learning," Research and Reviews: Advancement in Cyber Security, vol. 2, no. 2, pp. 33–40, 2025. doi: 10.5281/zenodo.15308198.

[2] N. Birla, "Vehicle Dataset from CarDekho," Kaggle, 2019. [Online]. Available: https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho

[3] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[4] W. McKinney, "Data Structures for Statistical Computing in Python," in Proc. 9th Python in Science Conf., 2010, pp. 51–56.

[5] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007.

[6] M. Waskom, "Seaborn: Statistical Data Visualization," Journal of Open Source Software, vol. 6, no. 60, p. 3021, 2021.

[7] G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning: with Applications in R, New York: Springer, 2013.

[8] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," Journal of the Royal Statistical Society: Series B (Methodological), vol. 58, no. 1, pp. 267–288, 1996.

[9] S. Raschka and V. Mirjalili, Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, Birmingham, UK: Packt Publishing, 2nd ed., 2017.