

 **Used Car Price Prediction System** **“ Car Valuation Tool ”**

**Submitted by –**

**NAME: Jubin Mazumdar**

**REGISTRATION NUMBER: 12316322**

**ROLL NUMBER: 43**

**SECTION: 9P172**

## **1. Title of the Project**

Used Car Price Prediction using Machine Learning

## **2. Objective**

The objective of this project is to develop a machine learning-based system that predicts the selling price of used cars based on various features such as brand, fuel type, vehicle age, kilometers driven, engine capacity, and more.

The system helps users:

- Estimate the market value of their car
- Make better buying/selling decisions
- Get instant predictions via a web interface

## **3. Problem Statement**

In the used car market, determining the correct price is challenging due to multiple influencing factors. Sellers often overprice or underprice their vehicles due to lack of knowledge.

This project aims to solve this problem by:

- Analyzing historical car data
- Using machine learning to predict accurate prices

## **4. Dataset Description**

- Dataset used: CarDekho Dataset

<https://www.kaggle.com/datasets/manishkr1754/cardekho-used-car-data>

- Contains features such as:

- Brand
- Fuel Type
- Transmission Type
- Seller Type
- Vehicle Age
- Kilometers Driven
- Mileage

- Engine (CC)
  - Max Power (bhp)
  - Number of Seats
  - Selling Price (target variable)
- Preprocessing steps:
    - Removed unnecessary columns (car\_name, Unnamed: 0)
  - train\_model
    - Applied log transformation on target variable to normalize skewed data
    - Handled categorical and numerical features separately

## 5. Technologies Used

- Python
- Pandas & NumPy → Data processing
- Scikit-learn → Model building
- Streamlit → Web application
- Pickle → Model saving/loading

## 6. Machine Learning Approach

- Feature Engineering
  - Categorical features encoded using OneHotEncoder
  - Numerical features used directly
- Model Used
  - Random Forest Regressor

Reason:

- Handles non-linear relationships
- Works well with mixed data types
- Reduces overfitting using multiple trees

- Training Process
  - Data split into:
    - 80% Training
    - 20% Testing
  - Model parameters:
    - n\_estimators = 100
    - max\_depth = 15

## 7. Model Evaluation Metrics

The model is evaluated using:

- R<sup>2</sup> Score → Accuracy of prediction
- MAE (Mean Absolute Error) → Average prediction error
- RMSE (Root Mean Squared Error) → Penalizes large errors

These metrics ensure the model performs well on unseen data.

- Model Results:

MODEL PERFORMANCE	
R2 Score	: 0.9342
MAE	: ₹ 100,522
RMSE	: ₹ 222,629

- Interpretation:
  - The R<sup>2</sup> score of 0.93 indicates that the model explains 93.42% of the variance, which means high accuracy.
  - The MAE (~₹1 lakh) shows the average prediction error is reasonable for car prices.
  - The RMSE (~₹2.2 lakh) indicates some larger errors but still acceptable for real-world data.

## 8. System Architecture

- Training Phase
  1. Load dataset
  2. Preprocess data
  3. Encode categorical features
  4. Train Random Forest model
  5. Save:
    - model.pkl
    - encoder.pkl
    - columns.pkl
- Prediction Phase
  1. User inputs data via UI
  2. Data is encoded
  3. Model predicts log price
  4. Converted back using exponential
  5. Display final price

## 9. Web Application (Frontend)

The project includes a Streamlit-based UI app

Features:

- User-friendly interface
- Input fields:
  - Brand
  - Fuel Type
  - Transmission
  - Seller Type
  - Vehicle Age
  - KM Driven
  - Mileage

- Engine
- Max Power
- Seats
- Button: "Check Value"
- Output: Predicted price in Indian format (Lakh / Crore)

The screenshot shows the initial state of the Car Valuation Tool. It features a dark-themed UI with light-colored input fields. At the top center is the title "Car Valuation Tool" flanked by small car icons. Below the title is a subtitle: "Get an instant & accurate estimate for your car". The form contains several input fields:
 

- Brand:** A dropdown menu labeled "-- Select Brand --".
- Fuel Type:** A dropdown menu labeled "-- Select Fuel --".
- Mileage (km/l):** An input field with a placeholder "Enter mileage" and a numeric slider.
- Vehicle Age (years):** A slider with a value of 0.
- Transmission:** A dropdown menu labeled "-- Select --".
- Engine (CC):** An input field with a placeholder "Enter engine CC" and a numeric slider.
- Kilometers Driven:** An input field with a placeholder "Enter km driven" and a numeric slider.
- Seller Type:** A dropdown menu labeled "-- Select --".
- Max Power (bhp):** An input field with a placeholder "Enter max power" and a numeric slider.
- Seats:** A dropdown menu labeled "-- Select --".

 At the bottom left is a prominent red "Check Value" button.

The screenshot shows the Car Valuation Tool with specific values entered into the fields:
 

- Brand:** Maruti
- Fuel Type:** Petrol
- Mileage (km/l):** 40.0
- Vehicle Age (years):** 2
- Transmission:** Manual
- Engine (CC):** 500
- Kilometers Driven:** 1000
- Seller Type:** Dealer
- Max Power (bhp):** 200.0
- Seats:** 4

 At the bottom left is a red "Check Value" button. To the right of the button, a large gray callout box displays the estimated price: "Estimated Price: ₹ 34.35 Lakh" with a small car icon preceding the amount.

## **10. Key Functionalities**

- Real-time car price prediction
- Automatic feature encoding
- Log transformation handling
- Indian currency formatting (₹ Lakh / Crore)
- Error handling for missing files

## **11. Advantages**

- Fast and accurate predictions
- Easy to use interface
- Helps both buyers and sellers
- Reduces manual price estimation errors

## **12. Limitations**

- Limited dataset size
- Only supports selected brands
- Accuracy depends on data quality
- Does not include real-time market trends

## **13. Future Enhancements**

- Add more brands and features
- Deploy on cloud (AWS / Heroku)
- Add image-based car evaluation
- Use advanced models (XGBoost, Deep Learning)
- Real-time data integration

## **14. Conclusion**

This project successfully demonstrates how machine learning can be used to predict used car prices with reasonable accuracy. The integration of a trained model with a web interface makes the system practical and user-friendly.