Car Resale Price Prediction Tool: A ML Approach

Name:	**Rahul Kulkarni**
Registration No./Roll No.:	21150
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	August 17, 2023
Date of Submission:	November 19, 2023

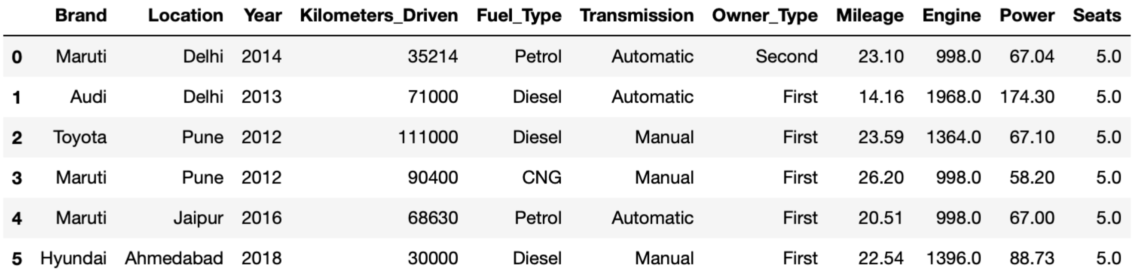
1 Introduction

Used Car price prediction is a challenging task that has attracted a lot of attention in recent years due to the rise of online E-Commerce in the automobile resale and dealership market, with the advent of machine learning and deep learning algorithms, it is now possible to develop predictive models that can analyze historical data of pre-owned cars and forecast future prices with reasonable accuracy.

Problem Statement: The objective of this project is to estimate the prices (lakhs) of used cars of different brands of India based on various factors and condition of car.

Dataset: The dataset we are working with has 5417 training data points. It contains both categorical and continuous features. The categorical features include the brand, location, fuel type, transmission, owner number, and number of seats in the car, while the continuous features consist of the year of purchase, kilometers-driven, mileage, power, and engine capacity of the car.

We imputed in missing values using the median method from SimpleImputer and removed outliers. To normalize the data and better accuracy and training results, I used one-hot-encoding and Standard Scaler and then trained the data on several predefined methods to obtain the best-performing model. This ensures that the data is not disturbed and that the focus remains on the main data points.



	Brand	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats
0	Maruti	Delhi	2014	35214	Petrol	Automatic	Second	23.10	998.0	67.04	5.0
1	Audi	Delhi	2013	71000	Diesel	Automatic	First	14.16	1968.0	174.30	5.0
2	Toyota	Pune	2012	111000	Diesel	Manual	First	23.59	1364.0	67.10	5.0
3	Maruti	Pune	2012	90400	CNG	Manual	First	26.20	998.0	58.20	5.0
4	Maruti	Jaipur	2016	68630	Petrol	Automatic	First	20.51	998.0	67.00	5.0
5	Hyundai	Ahmedabad	2018	30000	Diesel	Manual	First	22.54	1396.0	88.73	5.0

Figure 1: Overview of Data Set after Data Extraction

2 Data Pre-processing

I simplified the Brand column by removing the model name and replacing it with the brand name, which drastically reduced the number of features after encoding to just 29, this helped remove potential outliners which could distract the model. Also the Brand of the Car plays a important role in estimating the prize of the car. I also removed units in the Mileage, Engine, and Power columns. For example, "25.3 kmpl" [String DataType] was converted to 25.30 [Float DataType] which will be helpful for numerical manipulation during regression

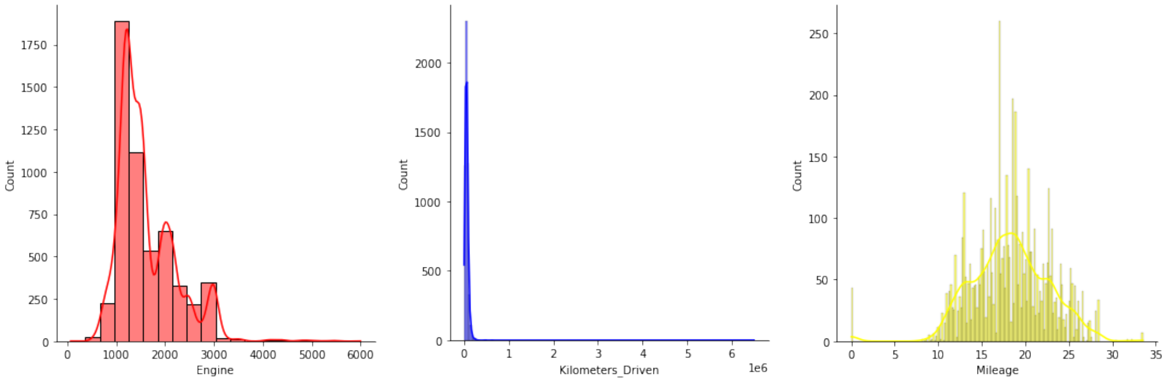


Figure 2: Dataset features with Outliners

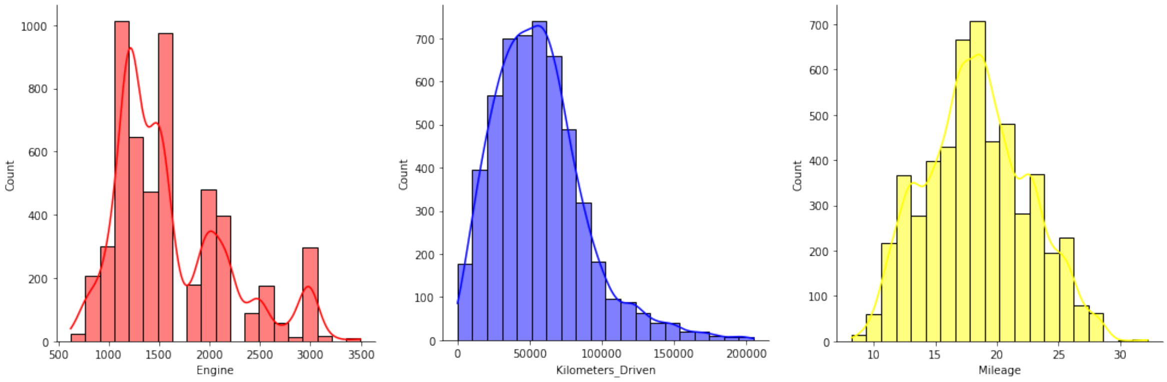


Figure 3: Dataset features without Outliners

3 Methods

Models used: For our regression problem, I used Linear Regression, Random Forest Regression, Adaboost Regression, Gradient Boosting Regression and KNN Regressor to train the model and compared them on the basis of Root Mean Squared Error (MSE), and R2 score. After training and Implementing the model in all the cases, I applied Cross Validation and Hyper Parameter Tuning to get the best performing Parameters for our model.

4 Experimental Setup

I initiated a comprehensive parameter tuning process for each model by constructing a parameter grid tailored to the specific characteristics of the model by using GridSearchCV, and systematically explore various combinations of hyperparameters through five cycles of cross-validation, ultimately identifying the optimal set that maximizes performance metric such as the R2 Score.

The hyperparameter tuning extends to diverse models, where, for instance, Random Forest Regression undergoes fine-tuning of parameters like criterion, the number of estimators, maximum tree depth, number of leafs and splitting criteria (dominant hyperparameter). KNN Regression is optimized through adjustments to number of nearest neighbours. AdaBoost Regression by tuning of parameters numbers of estimators and learning rate, while Gradient Boosting is tuned on number of estimators, learning rate and maximum depth. Subsequently, after determining the best-performing parameters, I evaluated these models using the R2 Score, to find out the optimal model for our regression problem

The correlation matrix also helped to visualise the linearity between different types of features

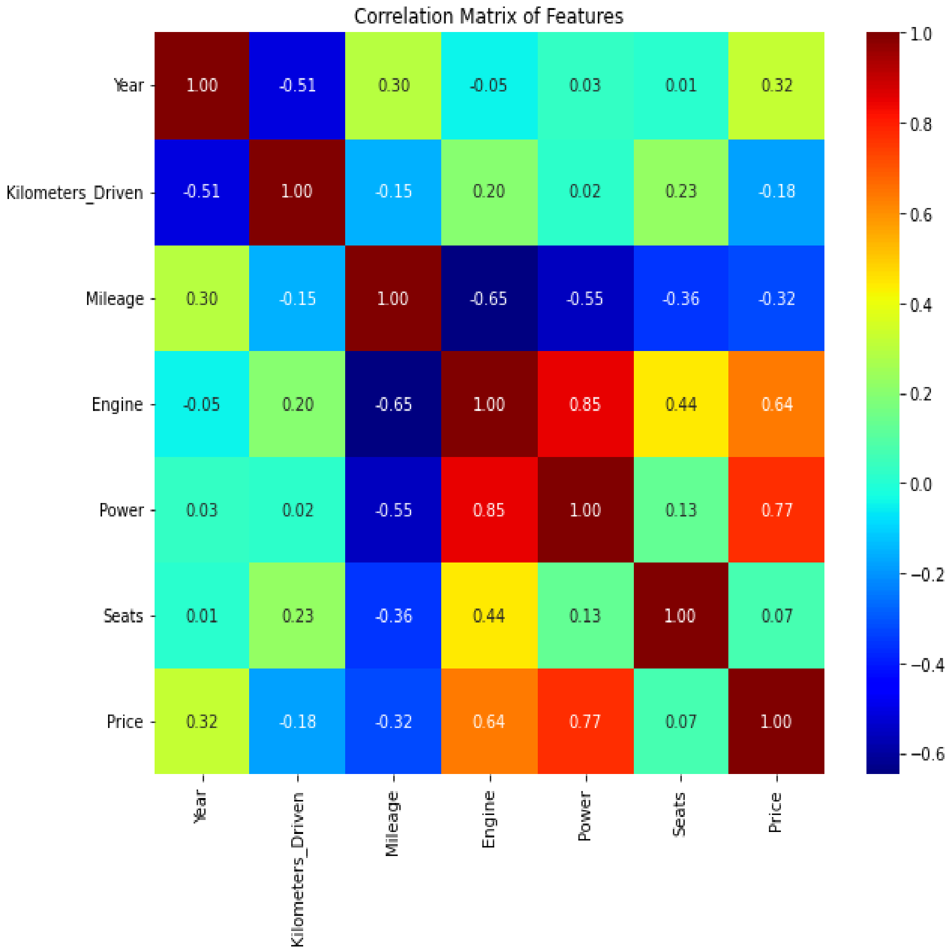
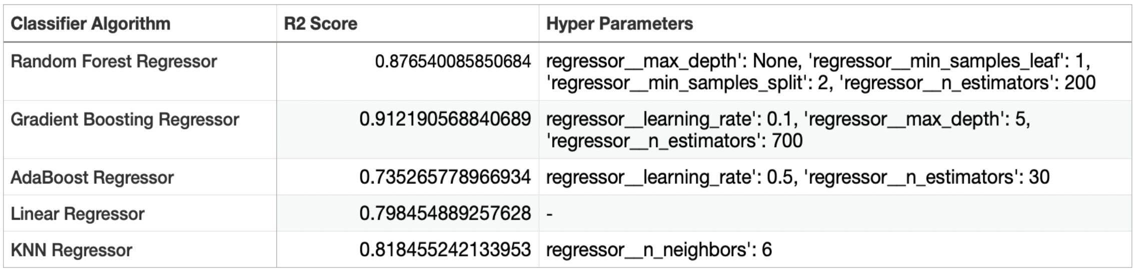


Figure 4: Correlation Matrix

5 Results and Discussion

The Experimental results can be seen in the table given below, of each of the model and method used. From the table, we saw that Gradient Boost Regressor, gave us the best performing metrics, with an R2 score of 91.21 percent. Among all of the regressors, Adaboost Regressor gave the least results, by changing the hyper parameters and fine tuning of Algorithms, a better result was obtain than just running bare algorithm.

The predicted vs original labels of the best performing model can be seen in the plot Figure 6.



Classifier Algorithm	R2 Score	Hyper Parameters
Random Forest Regressor	0.876540085850684	regressor__max_depth': None, 'regressor__min_samples_leaf': 1, 'regressor__min_samples_split': 2, 'regressor__n_estimators': 200
Gradient Boosting Regressor	0.912190568840689	regressor__learning_rate': 0.1, 'regressor__max_depth': 5, 'regressor__n_estimators': 700
AdaBoost Regressor	0.735265778966934	regressor__learning_rate': 0.5, 'regressor__n_estimators': 30
Linear Regressor	0.798454889257628	-
KNN Regressor	0.818455242133953	regressor__n_neighbors': 6

Figure 5: Model Evaluation and Best Hyper Parameter

References

https://ieeexplore.ieee.org/abstract/document/9800719/ https://escholarship.org/uc/item/35x3v9t4 Leo

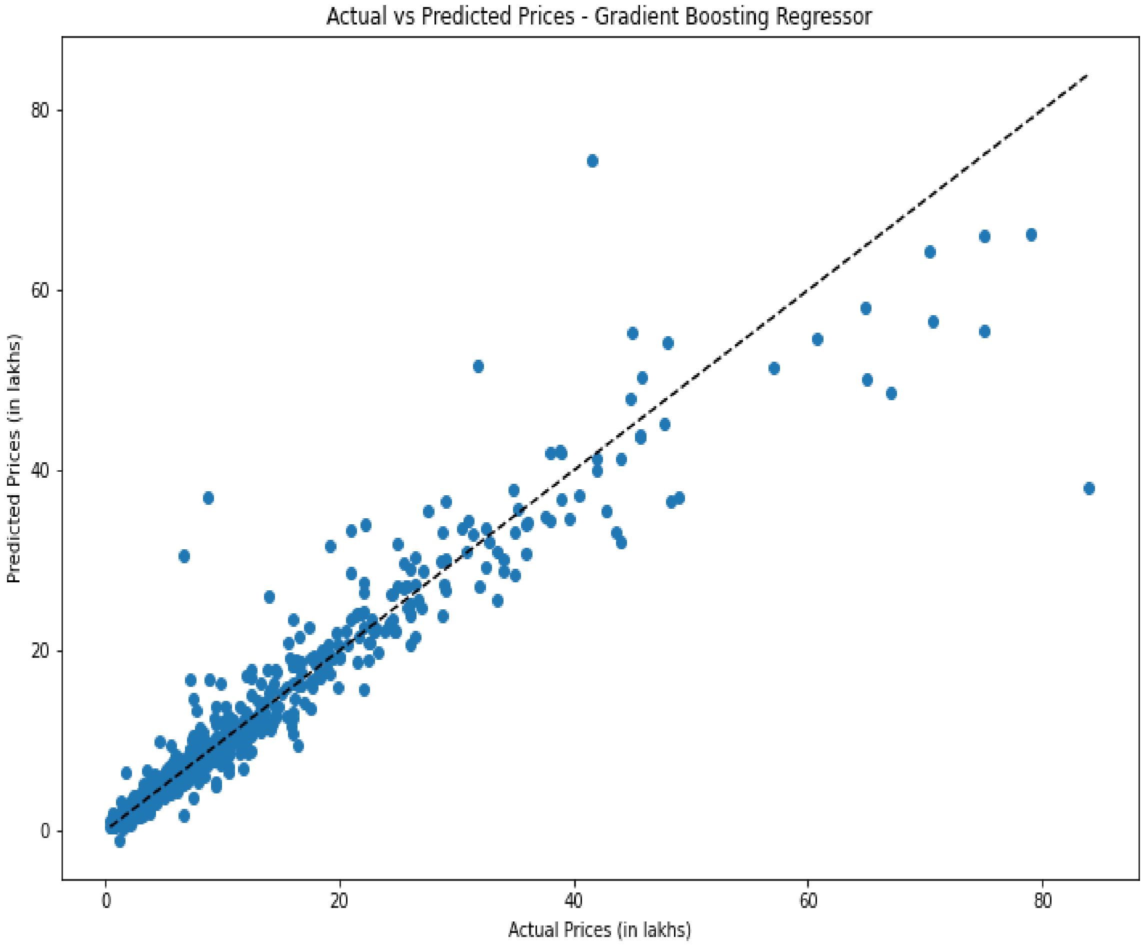


Figure 6: Predicted vs Actual Prices

Breiman. Random Forests. Machine learning, 45(1):5–32, 2001. gitHub: https://github.com/Rahul-Kulkarnii/ML-project