Used Car Price Prediction

Name:	**Shivam Chopra**
Registration No./Roll No.:	21338
Institute/University Name:	IISER Bhopal
Program/Stream:	DSE
Problem Release date:	August 18, 2023
Date of Submission:	November 19, 2023

1 Introduction

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.

Dataset: The dataset conists of 5417 training data points. In the given dataset, we are given both discrete and continuous type features. Discrete features is composed of brand of car, location of car, fuel type , transmission, owner number, and number of seats in the car. Continuous features consists of Year of purchase, kilometers driven in the car, mileage, power and engine capacity of the car. The features are not highly correlated with one another, and to build a linear relation with the target variable we use log transformation on the target variable.

In brief, we are filling the missing values using median method and also replace the outliers with median, which will prevent the data from getting disturbed and shifting its focus on outliers. We checked for outliers using IQR (Inter Quartile Range) method. The data before and after replacing the outliers can be in Figure 1. and Figure 2. Now, we use one-hot-encoding and Standard Scaler to normalize the data to prevent false accuracy and training results and train the data on several predefined methods, after getting the best preforming model, we tried to boost the performance of the model using ensemble techniques. After successful training and testing of data, we get our best performing Model as Random Forest Regressor with Adaptive Boosting with mean square error of 0.0416.

2 Some more pre-processing on data set

For the **Brand column**, we removed the exact model name of the car, as in brand name and replaced it with just the brand or the company owning the model of the car, which brought down the total of unique classes in the column Brand from 1787 to 29. Then, using similar methods we removed the units in the columns Mileage, Engine and Power. Example, 17.1 kmpl is converted to just 17.1

Linearity and correlation: After removing the outliers, it is important to check the relation of the target variable(Price) with the other continuous features, to prevent overfitting. After, checking the correlation of the data with various features, the data was not directly correlated with any feature, so no case of overfitting was shown. However, after plotting the data with various other features, we also see that target is randomly related with other features, so to make the data linearly related, we use log transformation on the target variable, Price and add a new column as our target variable, as log price. The correlation matrix and the linearity of data before and after log transformation can be seen in fig 3, fig4, fig 5 respectively.

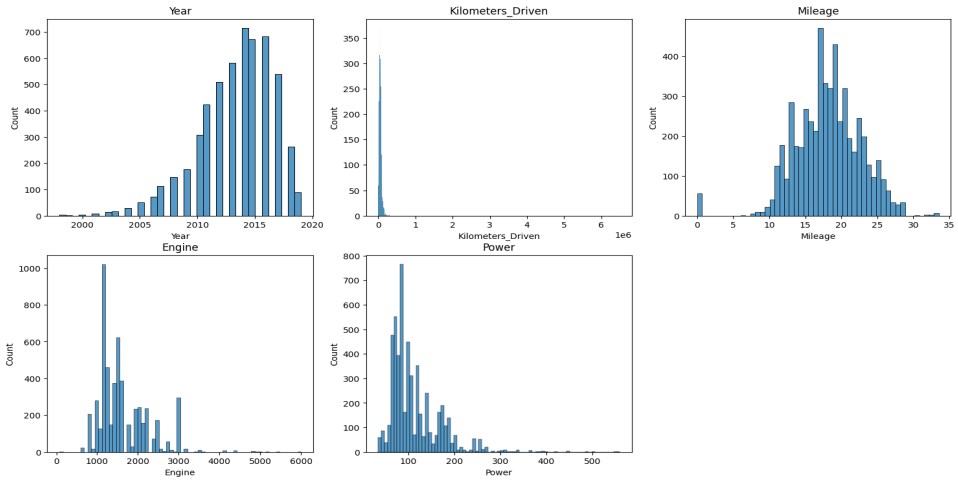

Figure 1: Data Set features with outliers

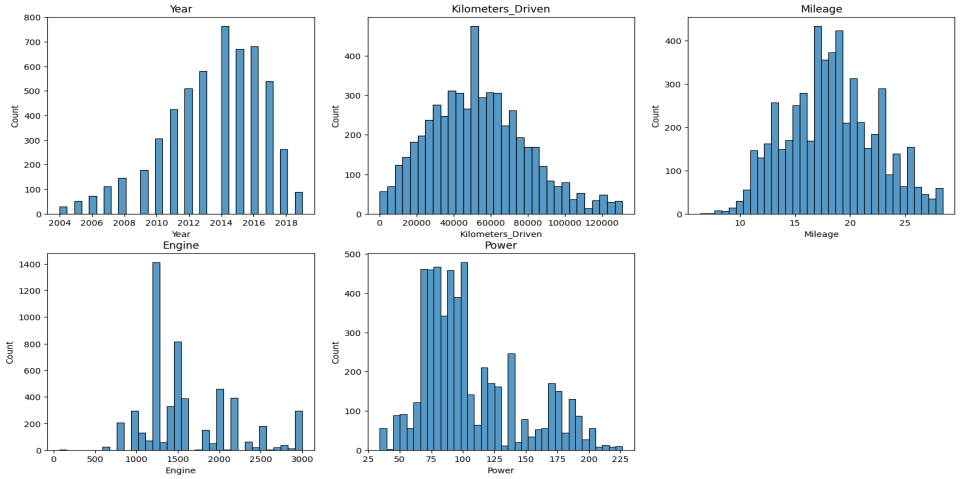

Figure 2: Data Set features after removing outliers

3 Methods

Models used: For our analysis, I used Linear Regression, Random Forest Regression, Decision Tree Regression, Ridge regression and SVM Regression to train the model and compared them on the basis of Mean Absolute Error (MAE), Mean Squared Error (MSE), and R2 score.

After training and getting the best parameters in all the cases, we applied Adaptive Boosting and Bagging method on the best performing classifier model to got better results.

4 Experimental Setup

For each and every model, we first make a grid of parameters for the particular model and then using GridSearch find the best parameters by running five cycles of cross-validation and based on the best parameters, we evaluate the preformance of the model based on certain parameters, like Mean Absolute Error(MAE), Mean Squared Error(MSE) and R2 Score.

Hyperparameters: We used GridSearchCV to find the best working parameters in various models. For Random Forest Regression, we tuned parameters like, criterion, n estimators, max depth and max features For Decision Tree Regression, we tuned, criterion, max features, max depth and ccp alpha, which is simply cost complexity pruning. For Support Vector Machine Regression, we tuned C and kernel parameters. For ridge regression, we just tuned solver.

After calculating the best performing parameters, we use ensemble methods, like Adaptive Boosting and Bagging method to further improve the score and performance of the best performing model.

	Year	Kilometers_Driven	Mileage	Engine	Power	\
Year	1.000000	-0.479688	0.293139	-0.057073	0.020308	
Kilometers_Driven	-0.479688	1.000000	-0.121144	0.166480	0.059363	
Mileage	0.293139	-0.121144	1.000000	-0.620689	-0.497743	
Engine	-0.057073	0.166480	-0.620689	1.000000	0.686464	
Power	0.020308	0.059363	-0.497743	0.686464	1.000000	
Price	0.295520	-0.182909	-0.331490	0.589943	0.492306	

	Price
Year	0.295520
Kilometers_Driven	-0.182909
Mileage	-0.331490
Engine	0.589943
Power	0.492306
Price	1.000000

Figure 3: Correlation matrix

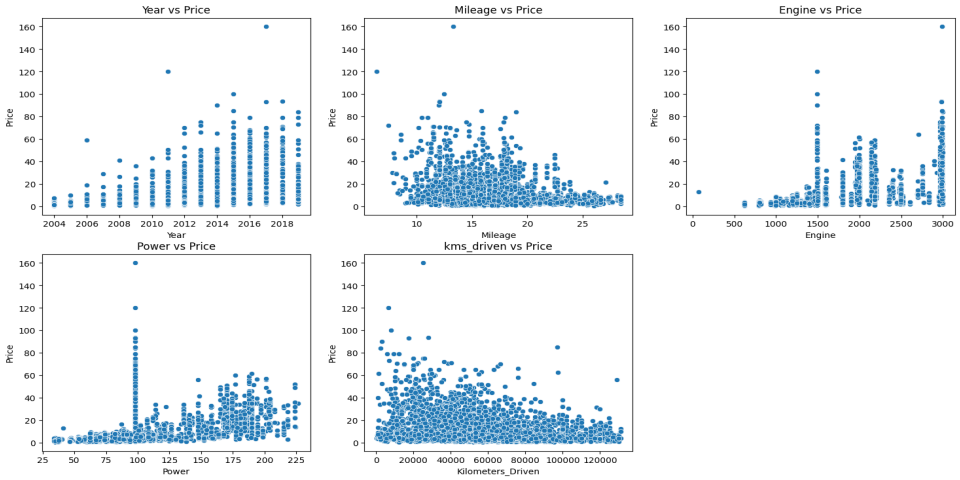

Figure 4: Dataset before log transformation

5 Results and Discussion

The Experimental results can be seen in the table 1, of each of the model and method used. From the table, we saw that Random Forest Regressor, when used with Adaptive Boosting gave us the best performing metrics, with an R2 score of 94.32 percent and MSE and MAE of 0.0416 and 0.1452 respectively. Among all of the classifiers, decision tree gave the least results, we are able to improve the results by changing the hyperparameters that we are tuning and removing ccp from the tuned parameters, however, even after changing the hyperparameters, Random Forest regressor was the best performing model. We tried more ensemble methods on the RFR like bagging method, but RFR with adaptive boosting gave the best results. On using feature extraction methods like PCA, the performance of the model is reduced due to low correlation of the features.

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

Table 1: Performance Of Different Classifiers Using All Features

Classifier	MAE	MSE	R2 Score
Random Forest Regressor(RFR)	0.1502	0.045	0.9375
Decision Tree Regressor	0.346	0.204	0.7212
Linear Regression	0.199	0.073	0.899
Ridge regression	0.232	0.113	0.846
Support Vector Machine	0.157	0.052	0.9287
Adaptive Boosting on Random Forest Regression	0.1452	0.0416	0.9432
Bagging method on RFR	0.156	0.05	0.9317

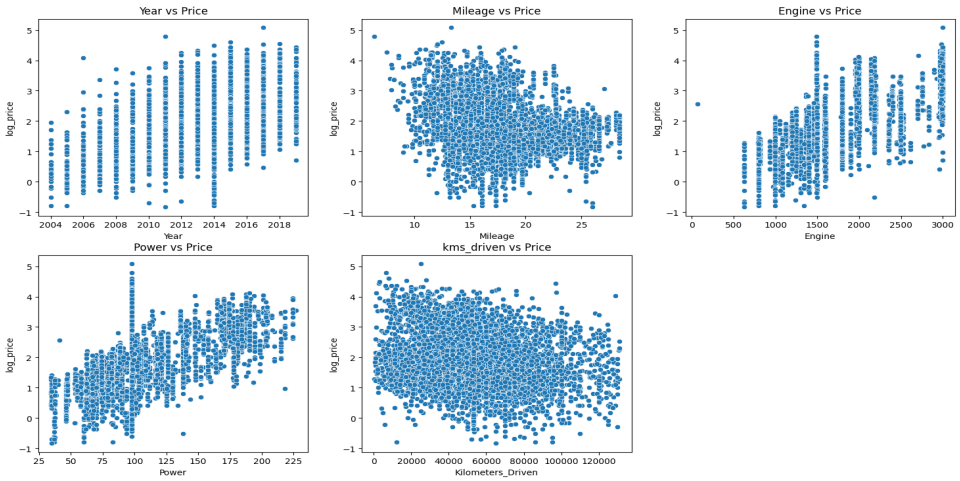

Figure 5: Dataset after log transformation of target price

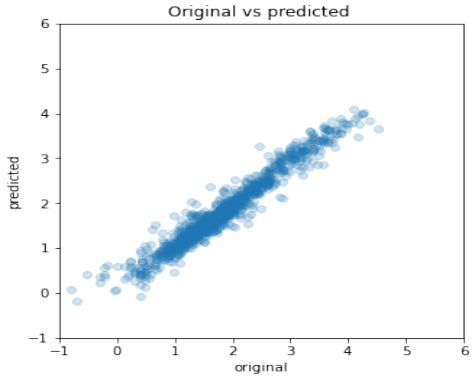

Figure 6: Original vs Predicted target price in Adaptive Boosting

6 Conclusion

In conclusion, we see that random forest regressor performed best with the given parameters and adaptive boosting. with a validation score of 0.9432 on the R2 score and 0.0416 on the Mean squared error. We can classify SVM as the second best performing model and maybe perform or try to tweak more of its parameters to boost its performance. We can try to tweak the target variable by maybe using some other transformation instead of log transformation, to build better relation between the target variable and other features.

References

https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

https://towardsdatascience.com/adaptive-boosting-a-stepwise-explanation-of-the-algorithm-50b75c3729c1

Github Link:
https://github.com/LeehZen/Machine-Learning-Project-21338.git