CAR PRICE PREDICTION

DSCI 4780/6780 Final Project Report

Abdul Afrid Mohammed

Matta Preethi Reddy

Krishna Chaitanya Pulipati

04-22-2023

Table of Contents

1 Business Understanding 3

1.1 Business Problem 3

1.2 Dataset 3

1.3 Proposed Analytics Solution 4

2 Data Exploration and Preprocessing 4

2.1 Data Quality Report 4

2.2 Missing Values and Outliers 8

2.3 Normalization 10

2.4 Feature Selection and Transformations 10

3. Model Selection and Evaluation 12

3.1 Evaluation Metrics 13

3.2 Models 13

3.3 Evaluation 14

3.3.1 Evaluation Settings and Sampling 14

3.3.2 Hyper-parameter Optimization 14

3.3.3 Evaluation 14

3.3.4 UI

4 **Results and Conclusion** 18

1 Business Understanding

The transportation industry is crucial in developed nations, with cars being referred to as the "industry of industries". The popularity of cars among people is increasing, but due to factors like affordability or economic conditions, many prefer to buy used cars. However, predicting used car prices accurately requires expert knowledge due to their dependence on various factors and features. Buying a used car from a dealer can be frustrating, as some dealers use deceitful tactics to close deals. This project aims to equip consumers with the right tools to guide them in their shopping experience and avoid falling victim to such tactics. Used car prices are not constant in the market, and buyers and sellers need an intelligent system to predict the correct price efficiently.

Determining the price of a used car is challenging because many factors affect a used vehicle's price on the market. It is difficult to decide whether a used car is worth the posted price when viewing listings online. From the seller's perspective, it is also challenging to price a used car appropriately. This project's primary objective is to predict car prices using features that are highly correlated with the price feature. We want to understand the factors that influence used car prices and model the car's price using available independent variables. To achieve this goal, we will use supervised machine learning techniques, which can analyze vast amounts of data to identify patterns and make accurate predictions.

1.1 Business Problem

The price of a car is dependent on the features it offers, such as the model, mileage, fuel type, and interior. The objective of this problem is to predict the price of a car based on its features. By doing so, we can estimate the cost of a car based on its characteristics and provide a good estimate of the price range it falls in. This prediction model can help in establishing transparency between car sellers and buyers in the market. Consumers can know the actual worth of a car or a desired car by providing the program with a set of attributes to predict the car's price. This can help consumers make informed decisions while buying or selling cars. We aim to use machine learning techniques to analyze and understand the various factors that influence car prices and build a predictive model that accurately predicts car prices based on these factors.

1.2 Dataset

The Car Price Prediction dataset, which can be found on Kaggle, consists of 19237 rows and 18 columns. The dataset contains a variety of descriptive features, including continuous and categorical variables. The target feature, "Price," is a continuous variable, while the remaining columns describe the car's characteristics, such as the manufacturer, model, category, leather interior, fuel type, and gearbox type, to name a few.

The continuous features in the dataset include Levy, Product Year, Engine Volume, Mileage, Cylinders, Doors, and Airbags.

The categorical features, on the other hand, include Manufacturer, Model, Category, Leather Interior, Fuel Type, Gearbox Type, Drive Wheels, Wheel, Color, and Turbo Engine.

To get a better understanding of each feature and how they relate to the target variable, detailed descriptions are provided in the Data Quality reports, which can be found in the upcoming chapters.

Overall, the Car Price Prediction dataset is a comprehensive collection of features that can be used to create models for predicting the price of cars.

1.3 Proposed Analytics Solution

This project deals with a dataset that comprises of car prices as the target feature and other descriptive features possessed by each car. The project will commence by importing necessary libraries, loading the dataset, and analysing it. Further, bar charts and histograms will be employed to visualize the data, which will aid in gaining a better understanding of the data and identifying features that are highly correlated with the target feature, which is the price of the car.

The next step would be Feature Selection using correlation and chi-square tests, and thereafter, applying supervised machine learning techniques to train a model on the selected features. To assess the performance of the models, the evaluation metric RMSE will be used, and the model with the best performance will be selected. This will enable accurate prediction of the price of a car. Ultimately, the project aims to provide buyers and sellers in the car market with a reliable tool to estimate the value of a car based on its features.

2 Data Exploration and Pre-Processing

The dataset used in this project is unprocessed and requires extensive cleaning. The following steps were taken to clean the data:

- Identify missing values in the "Levy" feature by examining the unique values of each feature. Replace any instances of "-" with NaN and change the feature's data type from "object" to "float64".
- The "Engine Volume" feature contains the string value "Turbo", which makes it an object type. Split this feature into a new feature called "Turbo Engine" that indicates whether the engine is turbo or not. The feature's data type was then changed from "object" to "float64".
- The string "km" was removed from the "Mileage" feature, and its data type was changed from "object" to "int64".
- The "Doors" feature was examined, and its integer data was extracted and converted to a data type of "int64".
- The "No of Cylinders" feature's data type was changed from "object" to "int64".
- After these cleaning steps were taken, a cleaned dataset was created and used to generate Data Quality Reports for both the continuous and categorical features.

2.1 Data Quality Report

The Data Quality Report for each Categorical Feature is shown in Table 1 below. This report gives important information such as Total Count, % of Missing Values, Mode, Mode Frequency, 2nd Mode, 2nd Mode Frequency, etc, of the Categorical Features.

Table 1. Data Quality Report for Categorical Features:

Feature	Desc	Count	% of Missing	Card.	Mode	Mode Freq.	Mode %	2nd Mode	2nd Mode Freq.	2nd Mode %
Manufacturer	Manufacturer Details	19237	0	65	HYUNDAI	3769	19.59245205	TOYOTA	3662	19.03623226
Model	Model of the Car	19237	0	1590	Prius	1083	5.629775953	Sonata	1079	5.60898269
Category	Category of the Car	19237	0	11	Sedan	8736	45.41248635	Jeep	5473	28.45038208
Leather interior	Is it Leather interior	19237	0	2	Yes	13954	72.53729792	No	5283	27.46270208
Fuel type	Type of Fuel	19237	0	7	Petrol	10150	52.76290482	Diesel	4036	20.98040235
Gear box type	Which type of Gear Box is used?	19237	0	4	Automatic	13514	70.25003899	Tiptronic	3102	16.12517544
Drive wheels	Drive wheels	19237	0	3	Front	12874	66.92311691	4x4	4058	21.0947653
Wheel	Wheel Type	19237	0	2	Left wheel	17753	92.28569943	Right-hand drive	1484	7.714300567
Color	Car Color	19237	0	16	Black	5033	26.16312315	White	4489	23.33523938
Turbo Engine	Is the engine type Turbo	19237	0	2	No	17306	89.9620523	Yes	1931	10.0379477

Table 2. Data Quality Report for Continuous Features:

The Data Quality Report for each Continuous Feature is shown in Table 2 below. This report gives important information such as Total Count, % of Missing Values, Minimum, Maximum, Median, Standard Deviation, Quartiles etc of the Continuous Features.

Feature	**Desc**	**Count**	**% of Missing**	**Card.**	**Min.**	**Q1**	**Median**	**Q3**	**Max.**	**Mean**	**Std.Dev.**
Price	Price of the Car	19237	0	2315	1	5331	13172	22075	26307500	18555.92722	190581.2697
Levy	Levy: Tax	19237	30.24899932	559	87	640	781	1058	11714	906.8381279	461.8670512
Prod. year	Production year of the Car	19237	0	54	1939	2009	2012	2015	2020	2010.912824	5.668672994
Engine volume	Volume of the engine	19237	0	65	0	1.8	2	2.5	20	2.307989811	0.8778045085
Mileage	Mileage of the car	19237	0	7687	0	70139	126000	188888	2147483647	1532235.688	48403869.38
Cylinders	No of Cylinders	19237	0	13	1	4	4	4	16	4.582991111	1.199933168
Doors	No of Doors	19237	0	3	2	4	4	4	6	3.932525862	0.4285274981
Airbags	No of Airbags	19237	0	17	0	4	6	12	16	6.582627229	4.320168395

Figure 1. Visualizations of Categorical Features in Dataset using Barplots:

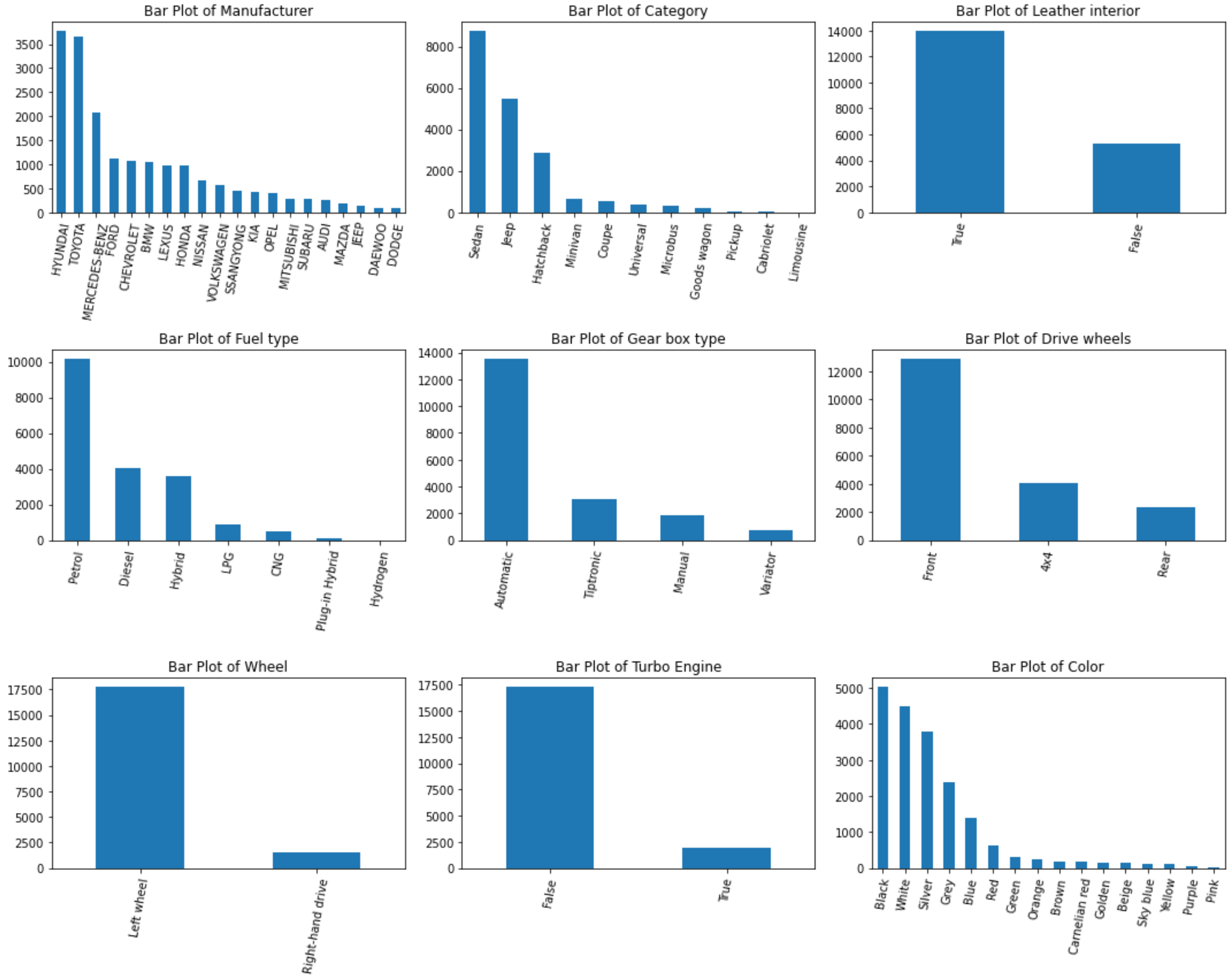
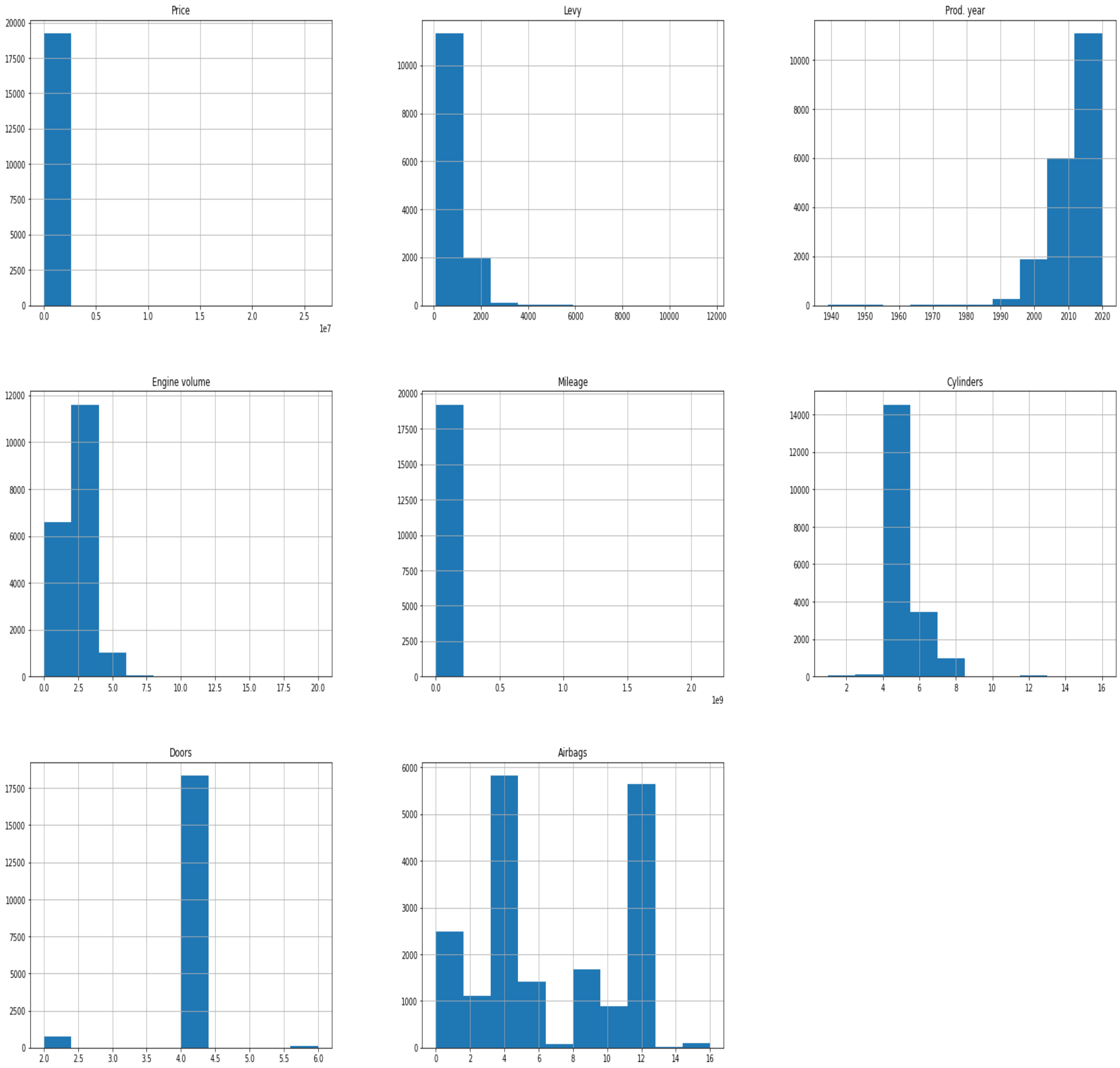

Bar Plot of Manufacturer
3500
3000
2500
2000
1500
1000
500
0
HYUNDAI
TOYOTA
MERCEDES-BENZ
FORD
CHEVROLET
BMW
LEXUS
HONDA
NISSAN
VOLKSWAGEN
SSANGYONG
KIA
OPEL
MITSUBISHI
SUBARU
AUDI
MAZDA
JEEP
DAEWOO
DODGE
Bar Plot of Category
8000
6000
4000
2000
0
Sedan
Jeep
Hatchback
Minivan
Coupe
Universal
Microbus
Goods wagon
Pickup
Cabriolet
Limousine
Bar Plot of Leather interior
14000
12000
10000
8000
6000
4000
2000
0
True
False
Bar Plot of Fuel type
10000
8000
6000
4000
2000
0
Petrol
Diesel
Hybrid
LPG
CNG
Plug-in Hybrid
Hydrogen
Bar Plot of Gear box type
14000
12000
10000
8000
6000
4000
2000
0
Automatic
Tiptronic
Manual
Variator
Bar Plot of Drive wheels
12000
10000
8000
6000
4000
2000
0
Front
4x4
Rear
Bar Plot of Wheel
17500
15000
12500
10000
7500
5000
2500
0
Left wheel
Right-hand drive
Bar Plot of Turbo Engine
17500
15000
12500
10000
7500
5000
2500
0
False
True
Bar Plot of Color
5000
4000
3000
2000
1000
0
Black
White
Silver
Grey
Blue
Red
Green
Orange
Brown
Carnelian red
Golden
Beige
Sky blue
Yellow
Purple
Pink

Figure 2. Visualizations of Continuous Features in Dataset using Histograms:



2.2 Missing Values and Outliers

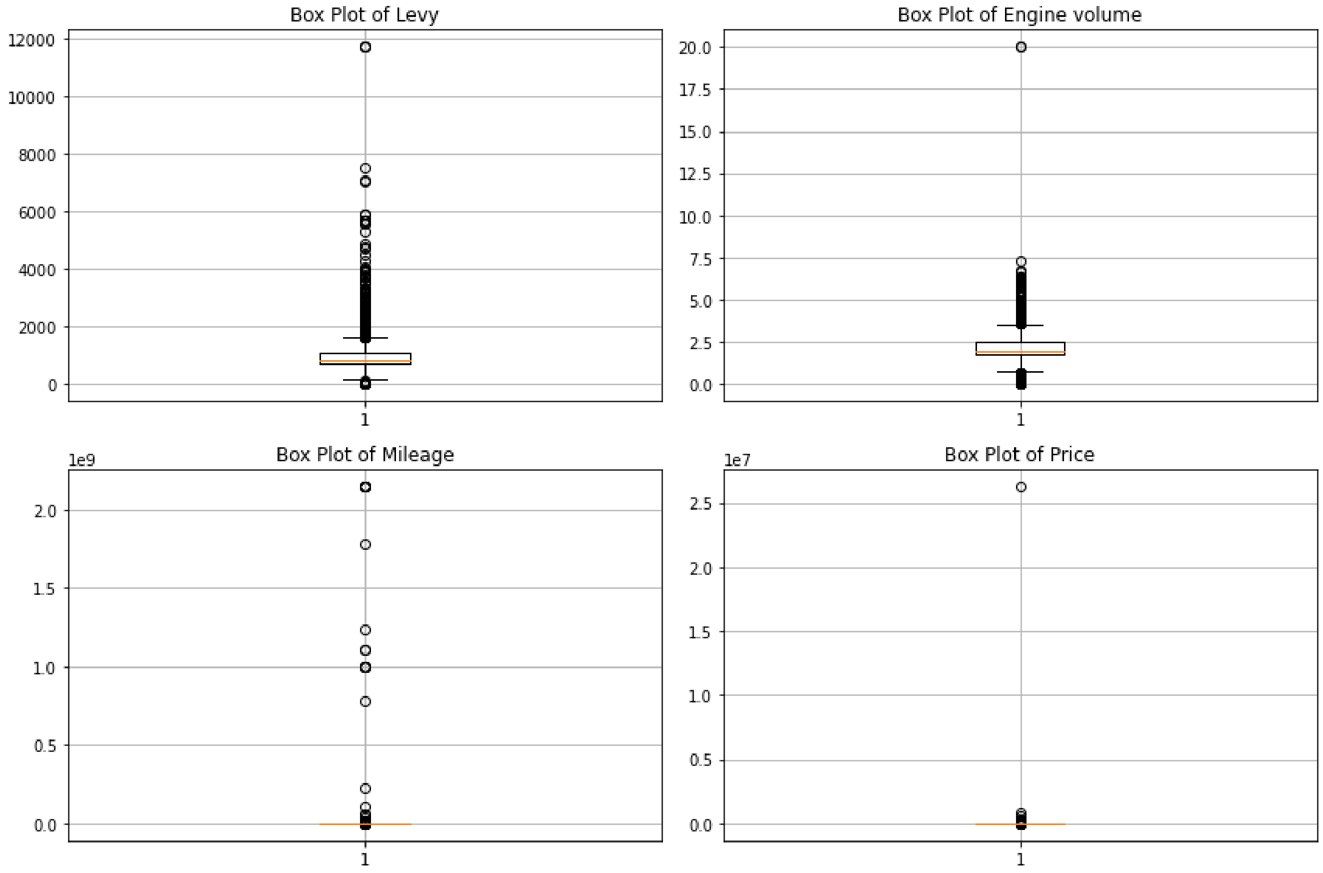
Handling Missing Values:

After analysing the dataset, we found that the "Levy" feature has 30.25% missing values. To handle this, we calculated the average levy for each manufacturer and replaced the NaN values with the mean value of the respective manufacturer.

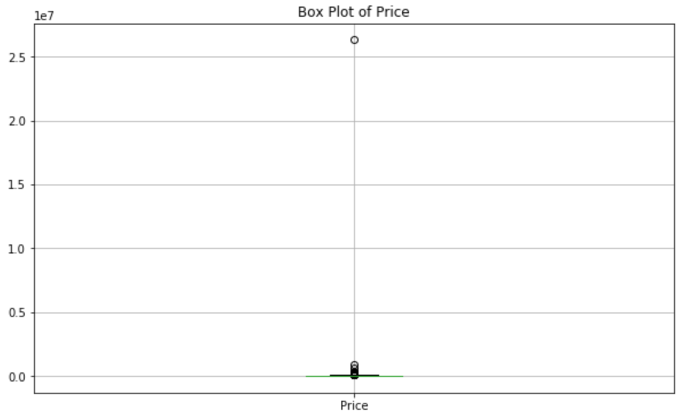
Handling Outliers:

We also noticed that the "Levy" and "Mileage" features contain outliers based on the box plots. Since most of the data is in exponential distribution form, we used complete case analysis and removed the outliers to handle them.

Figure 3. Visualizations of Box Plots:



In our analysis, we took into consideration the target feature of Price, as some cars have a price under 500, which would make their mean far away from the actual value.



The number of Updated values for each feature is given below:

Levy: No of removed values: 53

Mileage: No of removed values: 125

2.3 Normalization

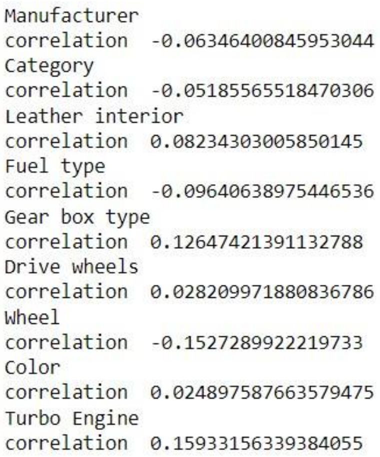
Normalization is essential for machine learning algorithms that calculate distances between data. If not scaled, the feature with a higher value range starts dominating when calculating distances. As we have removed the outliers in our data we have used StandardScaler for features 'Levy', and 'Prod. year' 'Engine volume', 'Mileage', and 'Airbags'.

$$z = \frac{x - \mu}{\sigma}$$

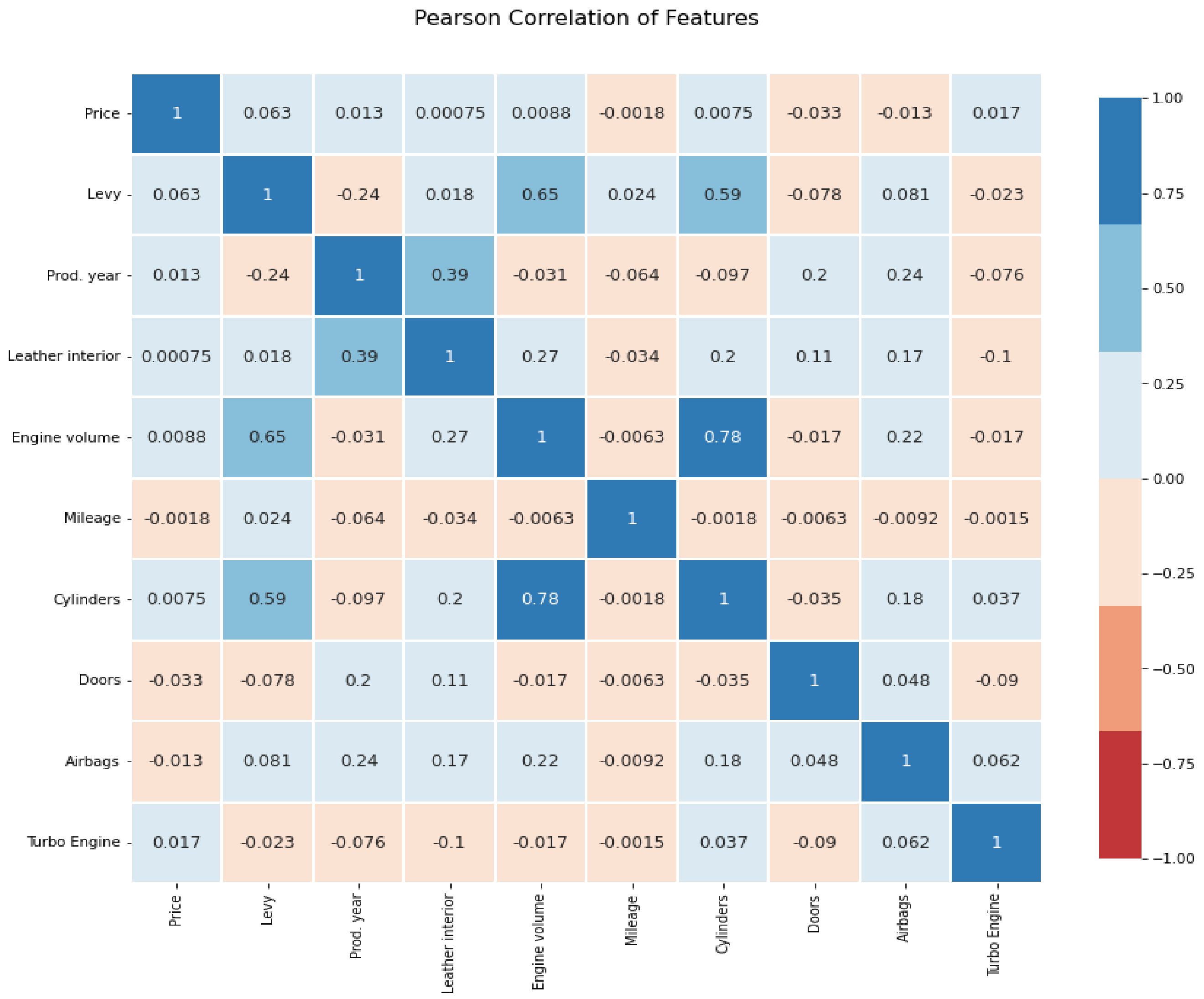
where xi is the ith instance, μ is the mean, and σ is the standard deviation. It results in a distribution standard deviation of 1.

2.4 Feature Selection and Transformations

We have conducted a Correlation test on categorical columns and continuous features:

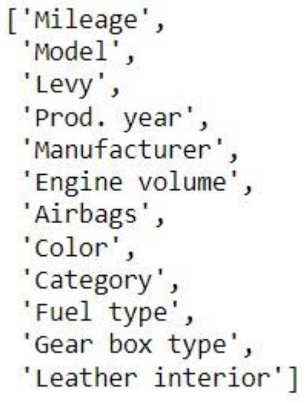


```
Manufacturer
correlation  -0.06346400845953044
Category
correlation  -0.05185565518470306
Leather interior
correlation  0.08234303005850145
Fuel type
correlation  -0.09640638975446536
Gear box type
correlation  0.12647421391132788
Drive wheels
correlation  0.028209971880836786
Wheel
correlation  -0.1527289922219733
Color
correlation  0.024897587663579475
Turbo Engine
correlation  0.15933156339384055
```


Pearson Correlation of Features

	Price	Levy	Prod. year	Leather interior	Engine volume	Mileage	Cylinders	Doors	Airbags	Turbo Engine
Price	1	0.063	0.013	0.00075	0.0088	-0.0018	0.0075	-0.033	-0.013	0.017
Levy	0.063	1	-0.24	0.018	0.65	0.024	0.59	-0.078	0.081	-0.023
Prod. year	0.013	-0.24	1	0.39	-0.031	-0.064	-0.097	0.2	0.24	-0.076
Leather interior	0.00075	0.018	0.39	1	0.27	-0.034	0.2	0.11	0.17	-0.1
Engine volume	0.0088	0.65	-0.031	0.27	1	-0.0063	0.78	-0.017	0.22	-0.017
Mileage	-0.0018	0.024	-0.064	-0.034	-0.0063	1	-0.0018	-0.0063	-0.0092	-0.0015
Cylinders	0.0075	0.59	-0.097	0.2	0.78	-0.0018	1	-0.035	0.18	0.037
Doors	-0.033	-0.078	0.2	0.11	-0.017	-0.0063	-0.035	1	0.048	-0.09
Airbags	-0.013	0.081	0.24	0.17	0.22	-0.0092	0.18	0.048	1	0.062
Turbo Engine	0.017	-0.023	-0.076	-0.1	-0.017	-0.0015	0.037	-0.09	0.062	1

The correlation value measures the strength and direction of the relationship between two variables. A correlation value closer to zero indicates a weak association between the variables. In this case, the target class (car price) is being analysed in relation to different features, and it was found that the 'Color' feature has the weakest correlation with the target, followed by the 'Drive Wheels' feature. A positive correlation suggests that as the target (car price) increases, the feature value increases as well, while a negative correlation indicates that as the target increases, the feature value decreases and vice versa. Feature selection using Information gain:



```
['Mileage',
 'Model',
 'Levy',
 'Prod. year',
 'Manufacturer',
 'Engine volume',
 'Airbags',
 'Color',
 'Category',
 'Fuel type',
 'Gear box type',
 'Leather interior']
```

We selected the top 12 features having maximum information gain, as the RMSE value does not seem to increase with the increasing number of features, and it is said: Occam’s Razor: “With all things being equal, the simplest explanation tends to be the right one.”
William of Ockham (1288-1348)

Transformations:

In machine learning, categorical variables are those which contain discrete values, such as color or type. To use such variables in a machine learning model, they need to be transformed into numerical values. One way to do this is by using one-hot encoding, which transforms each category into a binary vector, where only one element is 1 and the rest are 0s.

In this case, the dataset contained categorical values, and the one-hot encoding method was used to transform them into numerical values. The original dataset had 18 columns and 16,728 rows. After applying the one-hot encoding technique, the dataset was transformed into a new dataset with 88 columns and the same number of rows. This means that each categorical feature was transformed into multiple columns, each corresponding to a possible category value.

shape of the dataset before encoding = (16728, 18)
shape of the dataset after encoding = (16728, 88)

3 Model Selection and Evaluation

Model Selection and Evaluation is a crucial step in the process of developing a machine-learning model. It involves analyzing the performance of different models and selecting the best one based on various metrics. Model Evaluation is used to test the model's ability to make accurate predictions on new, unseen data. A simple model may not be able to capture complex patterns in the data, while a complex model may overfit the data, meaning it memorizes the training data but fails to generalize well to new data. Therefore, model selection involves finding the right balance between complexity and performance by selecting a model that can accurately represent the data while also being flexible enough to generalize well.

To prepare the data for training, categorical variables are encoded, meaning they are converted into numerical values, and continuous variables are normalized, meaning their values are scaled to be within a certain range. This ensures that all variables are in the same format and have the same impact on the model's performance. Once the data is prepared, it is ready for the training phase, where the model is trained on the dataset to learn the patterns in the data and make accurate predictions on new, unseen data.

3.1 Evaluation metrics

The Evaluation metrics of a regression problem include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). As MSE squares, the units of it won't be suitable to use as an evaluation metric. We use RMSE as the unit of the error would be the same as the target feature.

$$RMSE = \sqrt{\frac{\Sigma_i^N (x_i - \hat{x}_i)^2}{N}}$$

where x-cap is the mean and xi is the instance and N is the number of features.

3.2 Models

In this business problem, our primary goal is to predict the price of the car which is a continuous target feature. So, this is a regression problem and can be solved by using several regression algorithms. A regression model provides a function that describes the relationship between one or more independent variables and a target variable.

The models that we have used for the car price prediction are as follows:

Linear Regression - Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data.

K-Nearest Neighbours (KNN) - K-Nearest Neighbour is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure.

ADA boost classifier - AdaBoost is an ensemble learning method. AdaBoost uses an iterative approach to learn from the mistakes of weak classifiers and turn them into strong ones.

Support Vector Regression (SVR) - Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to find the best-fit line. In SVR, the best-fit line is the hyperplane that has the maximum number of points.

Random forest Regressor - A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

3.3 Evaluation

3.3.1 Evaluation Settings and Sampling

The data sample is split into a 'k' number of smaller samples; hence the name: K-fold Cross Validation. with the value of k = 10 and the split is shuffled. This gives us random instances without any order. The models are trained on these splits and then are evaluated using the mean squared error metrics taking the root of it gives us root mean squared error.

3.3.2 Hyper-parameter Optimization

The Hyper-parameter Optimization for the models is explained below:

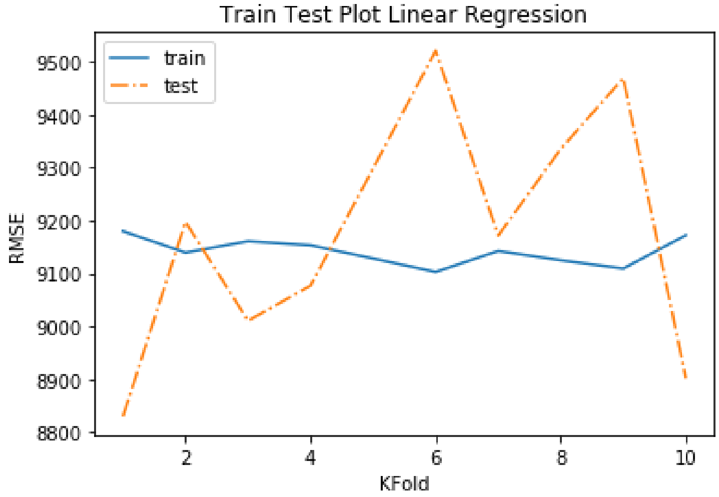
KNN: For KNN we changed the value of the neighbors from 1 to 20 incrementing each time by 2. We found out that the best neighbor value would be 1. We applied a GridCV search for KNN, and the optimal parameters were 'algorithm': 'auto', 'n_neighbors': 200, 'weights': 'distance'.

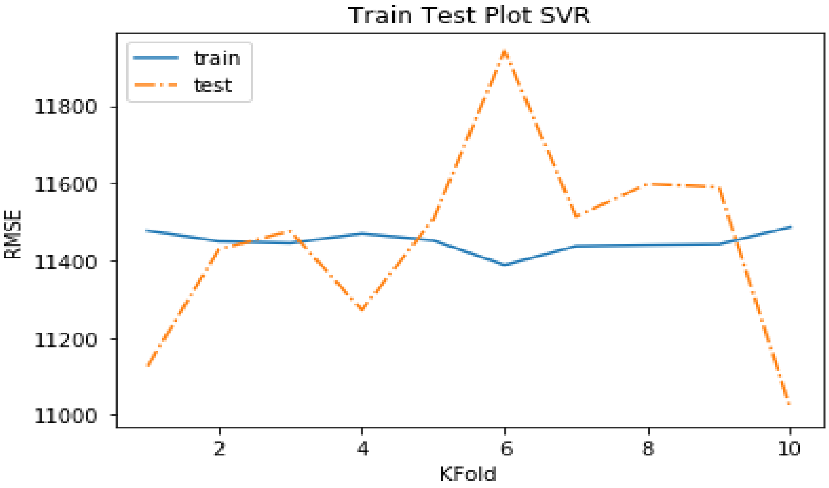
ADA boost classifier: For ADA boost classifier with base_estimator as Decision Tree and range of depth from 1 to 20 incrementing each time by 2. We found out that the best depth value would be 13.

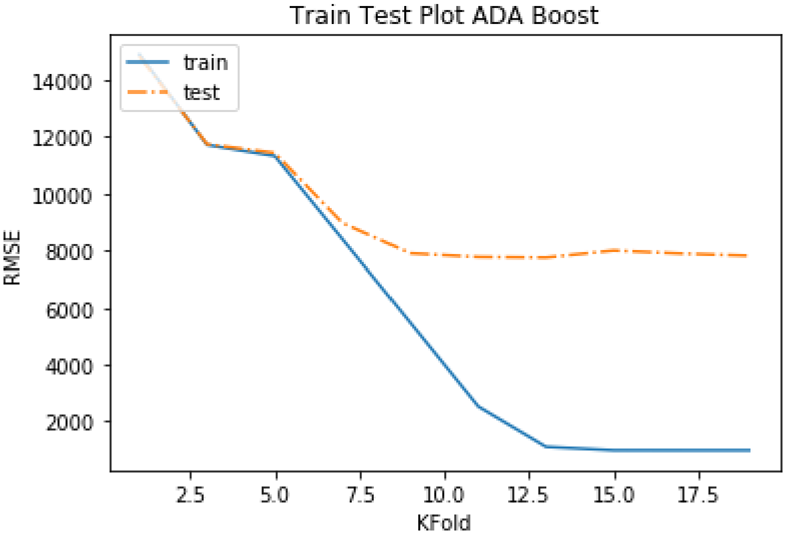
Random Forest Regressor: For Random Forest Regressor we changed the value of the depth from 1 to 20 incrementing each time by 2. We found out that the best depth value would be 25. Then, after training it we found out that the RMSE for Random Forest Regressor was minimum, but it was slightly memorizing the pattern on data and not generalizing it. So we used GridCV to go through all the parameters and and give us optimal features depending upon the data, The best parameters for it were 'max_depth': 8, 'max_features': 'sqrt', 'n_estimators': 500, 'random_state': 18.

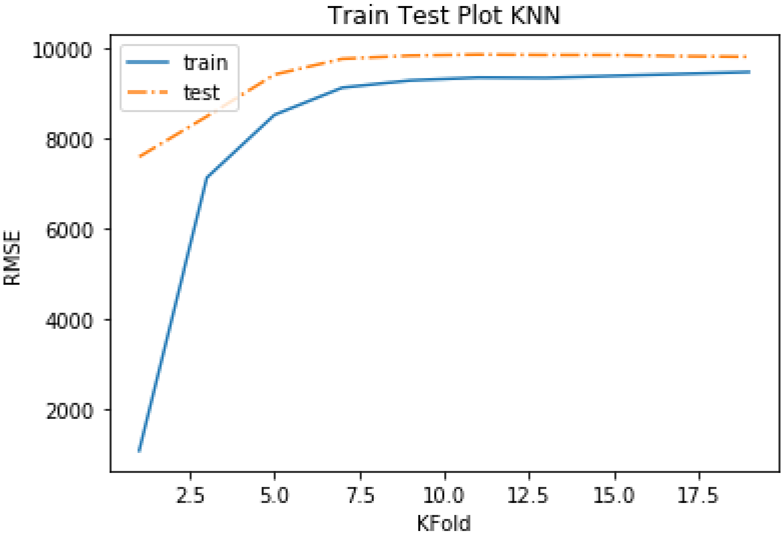
3.3.3 Evaluation

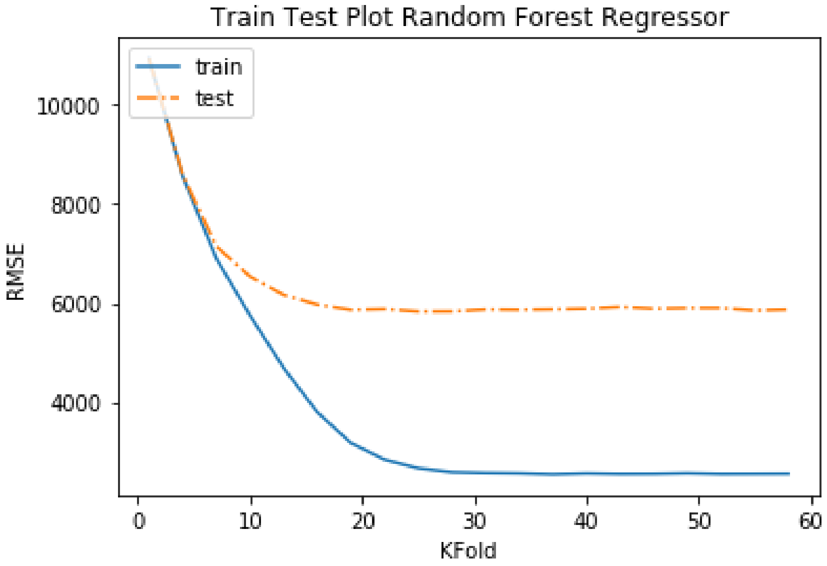
The train and test graph for each algorithm is given below with the x-axis as k and y as RMSE score:



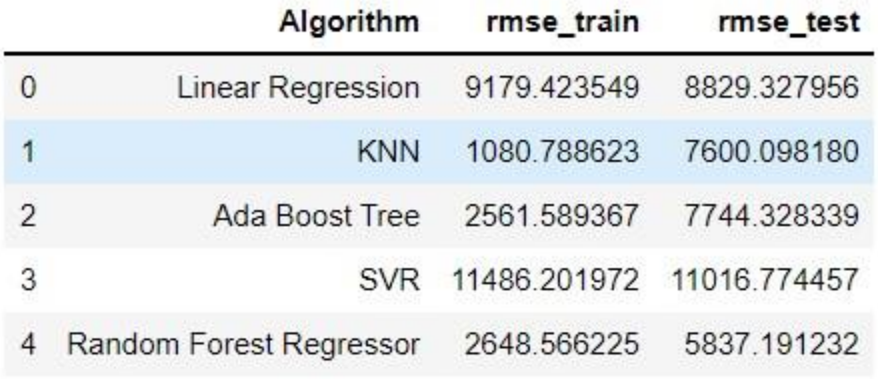

Train Test Plot SVR
train
test
11800
11600
11400
11200
11000
RMSE
2
4
6
8
10
KFold


Train Test Plot ADA Boost
train
test
14000
12000
10000
8000
6000
4000
2000
RMSE
2.5
5.0
7.5
10.0
12.5
15.0
17.5
KFold


Train Test Plot KNN
train
test
10000
8000
6000
4000
2000
RMSE
2.5
5.0
7.5
10.0
12.5
15.0
17.5
KFold



After training, we have taken the minimum of the average of the root mean square of training and testing for each algorithm trained on 10 k-fold splits. Below are the algorithms and their RMSE value for train and test samples.



	Algorithm	rmse_train	rmse_test
0	Linear Regression	9179.423549	8829.327956
1	KNN	1080.788623	7600.098180
2	Ada Boost Tree	2561.589367	7744.328339
3	SVR	11486.201972	11016.774457
4	Random Forest Regressor	2648.566225	5837.191232

Random forest Regressor slightly overfits the data but gives Lower RMSE we use GridCV on it to find optimal parameters and trained on those parameters The training and testing accuracy were:

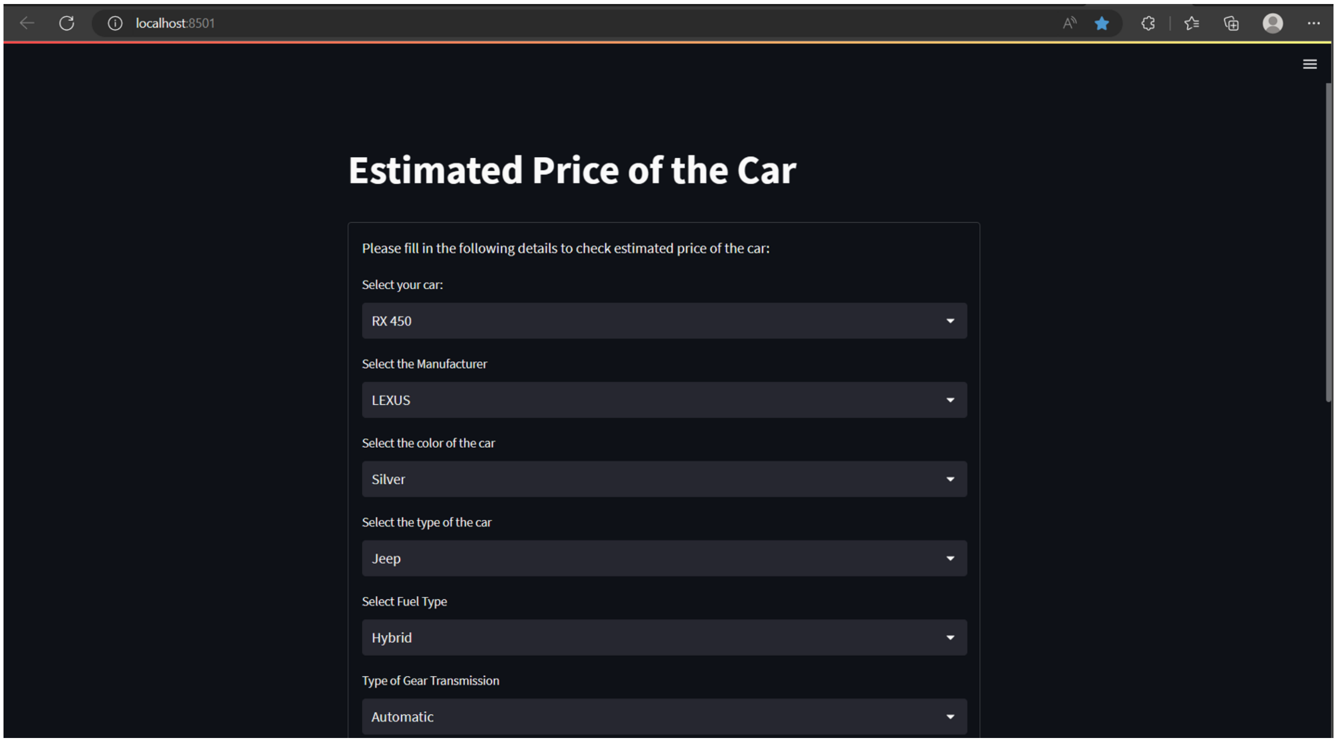
Train RMSE: 8035.985625349999 || Test RMSE: 8141.879769237276.

As it has the least RMSE and generalizes well on data we choose Random Forest Regressor as our model algorithm.

3.3.4 UI

I built a user interface for my project using Streamlit, which allowed me to easily create interactive visualizations and deploy my application to the web. With Streamlit, I was able to quickly develop a user-friendly interface that allowed users to input data, view results, and interact with the model in real time. However, this is not a full-fledged website yet and it needs some fine-tuning to be fully operational and deployed on the web.

Here, we can open the website and enter the details of the car we are willing to buy. It will give the estimated price of the car.



Estimated Price of the Car

Please fill in the following details to check estimated price of the car:

Select your car: RX 450

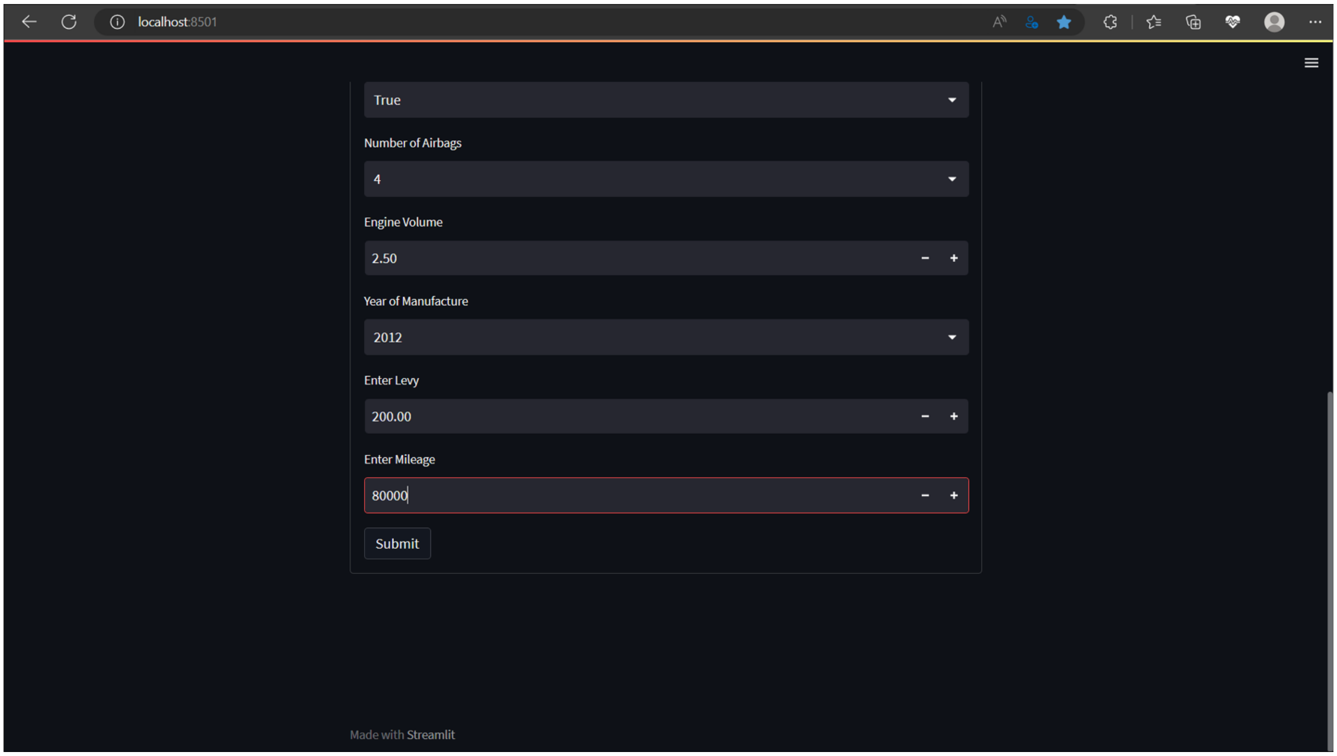
Select the Manufacturer: LEXUS

Select the color of the car: Silver

Select the type of the car: Jeep

Select Fuel Type: Hybrid

Type of Gear Transmission: Automatic



True

Number of Airbags: 4

Engine Volume: 2.50

Year of Manufacture: 2012

Enter Levy: 200.00

Enter Mileage: 80000

4 Results and Conclusion

The main goal of this project was to predict car prices based on features that are highly correlated with the price feature. Through our analysis, we found that 'Mileage' and 'Model' were the most correlated features, while 'Doors' and 'Turbo Engine' had little or no correlation with car prices.

After testing different machine learning algorithms, we found that Linear Regression performed well in terms of generalization with an RMSE of 8829.33 on test data. However, KNN and Decision Tree overfitted the data, while SVR generalized well with similar RMSE on both training and testing data, but it is having very high RMSE values. Random Forest Regressor was slightly overfitted, but we were able to improve its performance by tuning the hyperparameters using GridCV.

Overall, we chose Random Forest Regressor as our model algorithm to predict car prices based on its features, as it performed better at generalizing the data than the other algorithms.