information

MDPI

*Article*

# Application of Machine Learning Techniques to Predict the Price of Pre-Owned Cars in Bangladesh

**Fahad Rahman Amik, Akash Lanard, Ahnaf Ismat and Sifat Momen ***

Department of Electrical and Computer Engineering, North South University, Dhaka 1229, Bangladesh; fahad.rahman1@northsouth.edu (F.R.A.); akash.lanard@northsouth.edu (A.L.); ahnaf.ismat@northsouth.edu (A.I.)
* Correspondence: sifat.momen@northsouth.edu

**Abstract:** Pre-owned cars (i.e., cars with one or more previous retail owners) are extremely popular in Bangladesh. Customers who plan to purchase a pre-owned car often struggle to find a car within a budget as well as to predict the price of a particular pre-owned car. Currently, Bangladesh lacks online services that can provide assistance to customers purchasing pre-owned cars. A good prediction of prices of pre-owned cars can help customers greatly in making an informed decision about buying a pre-owned car. In this article, we look into this problem and develop a forecasting system (using machine learning techniques) that helps a potential buyer to estimate the price of a pre-owned car he is interested in. A dataset is collected and pre-processed. Exploratory data analysis has been performed. Following that, various machine learning regression algorithms, including linear regression, LASSO (Least Absolute Shrinkage and Selection Operator) regression, decision tree, random forest, and extreme gradient boosting have been applied. After evaluating the performance of each method, the best-performing model (XGBoost) was chosen. This model is capable of properly predicting prices more than 91% of the time. Finally, the model has been deployed as a web application in a local machine so that this can be later made available to end users.

**Keywords:** exploratory data analysis; feature selection; model deployment; overestimation; pre-owned cars; regression; root-mean-squared error; underestimation



**Citation:** Amik, F.R.; Lanard, A.; Ismat, A.; Momen, S. Application of Machine Learning Techniques to Predict the Price of Pre-Owned Cars in Bangladesh. *Information* **2021**, *12*, 514. https://doi.org/10.3390/info12120514

Academic Editor: Gennady Agre

Received: 22 October 2021
Accepted: 3 December 2021
Published: 9 December 2021



## 1. Introduction

Pre-owned cars (also known as second-hand cars or used cars) are vehicles with one or more previous retail owners. These type of cars are extremely popular in developing countries such Bangladesh. The price of a pre-owned car depends on various factors. As a consequence, customers who plan to purchase a pre-owned car often struggle to find an appropriate car within a budget. Even if a customer knows the type of car they want to purchase, it becomes challenging for them to estimate the price of the car. In this article, we look into this problem and present a machine learning approach to forecast the price of pre-owned cars in Bangladesh. The model described in this article will aid a prospective buyer to make a more informed decision while buying a pre-owned car.

### *1.1. Used Car Market in Bangladesh*

Bangladesh is a relatively small country in South Asia. At the moment, it has a population of around 166 million, placing it 8th on the list of nations by population [1]. The country accounts for 2.11% of the world's total population and has 12 million people in the middle-income bracket, which is growing at a rapid rate of 10% each year [2]. It is a rapidly developing country with 39.4% of the population living in cities.

Over the last decade, Bangladesh has steadily improved its Gross Domestic Product (GDP) per capita [3]. GDP is the total value of all goods and services produced in a country in a given year, and it is the most comprehensive measure of a country's overall economic performance. The economy of Bangladesh expanded 5.47% in the fiscal year 2020–2021,

following a revised 3.51% growth in the previous period. In the same year, in the midst of the pandemic, the per capita income of Bangladesh increased from $2064 to $2227 [4], which demonstrates that the purchasing power of its citizens has risen, as seen by the establishment of numerous small and medium-sized businesses. Additionally, with the emergence of ride-sharing services [5] such as Uber, Pathao, and Shohoz, the demand for pre-owned vehicles has been found to increase substantially [6].

Conventionally, purchasing a vehicle in Bangladesh has not been financially feasible for middle-income families partly due to its exorbitant cost. The high cost of car ownership is partially attributable to the hefty import tariffs imposed by the government [7]. The duties, as set by the government's national budget for fiscal year 2009–2010, were 250% for cars between 1600 cc and 2000 cc, 350% for cars between 2750 cc and 4000 cc and 500% for cars above 4000 cc [8]. As a consequence, a sizable portion of vehicle buyers, whether for personal or commercial purposes, largely rely on pre-owned cars rather than brand new ones. According to market insiders of Bangladesh, the current market size for reconditioned cars is around BDT 5000 crore (USD 587$M$), and each year the size of the market increases by 15% to 20% [9]. Recently, the demand for vehicles in Bangladesh has been found to rise over this pandemic period [10].

### *1.2. Applicability of Machine Learning in Predicting the Price of a Pre-Owned Car*

The price of a pre-owned car depends on various factors including model year, mileage, condition, equipment, etc. [11]. Since the price depends on so many factors, it is difficult to estimate it directly using rule-based algorithms. A more feasible strategy is to use inductive based learning to learn the price from the dataset. Hence, a machine learning approach [12] is very suitable for this application.

### *1.3. Research Goal*

The goals of this research are summarized as follows:

1. To collect dataset pertaining to pre-owned cars and to identify the prominent features that can be used to predict the price of cars.
2. To reliably predict the price of pre-owned cars using machine learning approaches.
3. To deploy the model as a web application in a local machine so that it can be later made available to end users.

The key contribution in this article lies in (i) collecting a real dataset pertaining to pre-owned cars and then preparing the dataset extensively, which can be used in (ii) developing an accurate predictive model. In addition to this, other key contributions include (iii) assessing how well the model predicts by comparing the predicted output on unseen data and (iv) deploying the model so that it can be used in the future by end users.

Rest of the article is organized as follows: related works in this field of study are reviewed in Section 2. Section 3 is dedicated to explaining all the steps of our research methodology. Useful results and findings are discussed in Section 4. The model's deployment procedure is described in detail in Section 5. Finally, the article is concluded in Section 6.

## 2. Related Work

Listiani [13] forecasted the price of a used vehicle by taking the vehicle's depreciation into consideration. Support Vector Regression is used, which is a dimension-independent data mining approach. Following that, the forecasting accuracy is compared to the statistical regression model. A fully automated technique for tuning and implementing SVR is devised, drawing on concepts from evolutionary search. The entire project, which employs a machine learning technique, is based on real-world data from a major German automaker.

Pal et al. [14] forecasted the price of secondhand vehicles using a random forest regressor trained on a Kaggle dataset. The model was selected following extensive exploratory data analysis to ascertain the effect of each variable on pricing. To train the data, a Random Forest was constructed with 500 decision trees. Empirical results demonstrate that the

model can predict with a training accuracy of 95.82% and a testing accuracy of 83.63%. The model can reliably forecast vehicle prices by selecting the most prominent attributes in the dataset, such as original price, kilometer, brand, and vehicleType, and then filtering out outliers and irrelevant features.

Gajera and colleagues [15] aimed to develop a statistical model capable of forecasting the price of a used vehicle. The accuracy with which these models forecast is utilized to identify the most appropriate method. The authors trained the model using a dataset of 92,386 entries. Features including kilometers driven, year of registration, fuel type, car model, fiscal strength, car brand, and gear type were found to have contributed towards the price of a vehicle. The authors used five regressors including K Nearest Neighbors (KNN), Random Forest, XGBoost, Decision Tree, and Linear regressors. Empirical results indicate that Random Forest has the lowest root mean square error (RMSE) as well as the highest R-squared value: 0.93.

Venkatasubbu et al. [16] examined the use of machine learning algorithms such as lasso regression, multiple regression, and regression trees to create a machine learning model capable of predicting the price of a used vehicle based on historical consumer data and a specified set of characteristics. The data were compiled from the 2005 Central Edition of the Kelly Blue Book and includes 804 records for 2005 General Motors vehicles. When the prediction accuracy of these models was compared, it was discovered that the prediction error rate for all models was far less than the acceptable 5% error rate. However, with further examination, it was also found that the mean error rate of the regression tree model was greater than the mean error rate of the multiple regression and lasso regression models. This was validated using Analysis of Variance (ANOVA). Additionally, the post hoc test indicated that the error rates for multiple regression and lasso regression models are not statistically different.

Monburinon and colleagues [17] compared the performance of regression models based on supervised machine learning. Multiple linear regression, random forest regression, and gradient boosted regression trees were used to construct a price model for used vehicles. Each model was trained using the same test data gathered from German e-commerce websites on the used vehicle industry. The highest performance was discovered for gradient-boosted regression trees with a mean absolute error (MAE) of 0.28, followed by random forest regression with an MSE of 0.35 and Multiple Linear Regression with an MSE of 0.55.

Lessmann et al. [11] explored how different variations in modeling procedures can affect the price prediction accuracy. After comparing different algorithms with their benchmarks, they concluded that ensemble methods are better overall for predicting the price of used cars.

Gegic et al. [18] investigates several characteristics for predicting used vehicle values in Bosnia and Herzegovina in a reliable and accurate manner. The authors used three machine learning approaches to construct a prediction model: Artificial Neural Networks, Support Vector Machines, and Random Forests. Instead of working independently, the approaches were integrated to work as an ensemble. Online scraper was utilized to collect data for the prediction from the web portal https://www.autopijaca.ba/, (accessed on 13 October 2021) [19]. After that, the respective performances of several algorithms were compared. It was established that the accuracy of a single machine learning algorithm on the data set was less than 50%. However, when the machine learning algorithms were implemented in combination with one another, the accuracy went up to 92.38%. When compared to a single machine learning algorithm, this is a substantial improvement. However, the suggested system has the disadvantage of consuming significantly more computing resources than a single machine learning method. The authors also pointed out that although data cleansing is one of the procedures that improves prediction performance, it is insufficient for complicated data sets like the one used in their study.

Samruddhi and colleagues [20] presented a supervised machine learning model for estimating the pricing of second-hand vehicles by utilizing the K-Nearest Neighbor (KNN)

method, which they believe is suited for small data sets. The model was trained on the data, and its accuracy was evaluated using various ratios of train to test data. The same model is cross-validated to determine the model's performance using the K-Fold cross validation approach. The model was trained using a dataset available in Kaggle. The accuracy of the K-Nearest Neighbor algorithm was found to be 85%, whereas the accuracy of linear regression was found to be 71%. Additionally, the suggested model is tested with five and ten folds using the K Fold Method. The experimental study demonstrates that the suggested model is optimally suited.

Several other initiatives have been undertaken in the context of connected and autonomous cars [21], as well as feature selection using optimization techniques [22,23] which are applicable in the context of selecting features in various datasets.

## 3. Research Methodology

The approach adopted in this work is outlined in Figure 1:

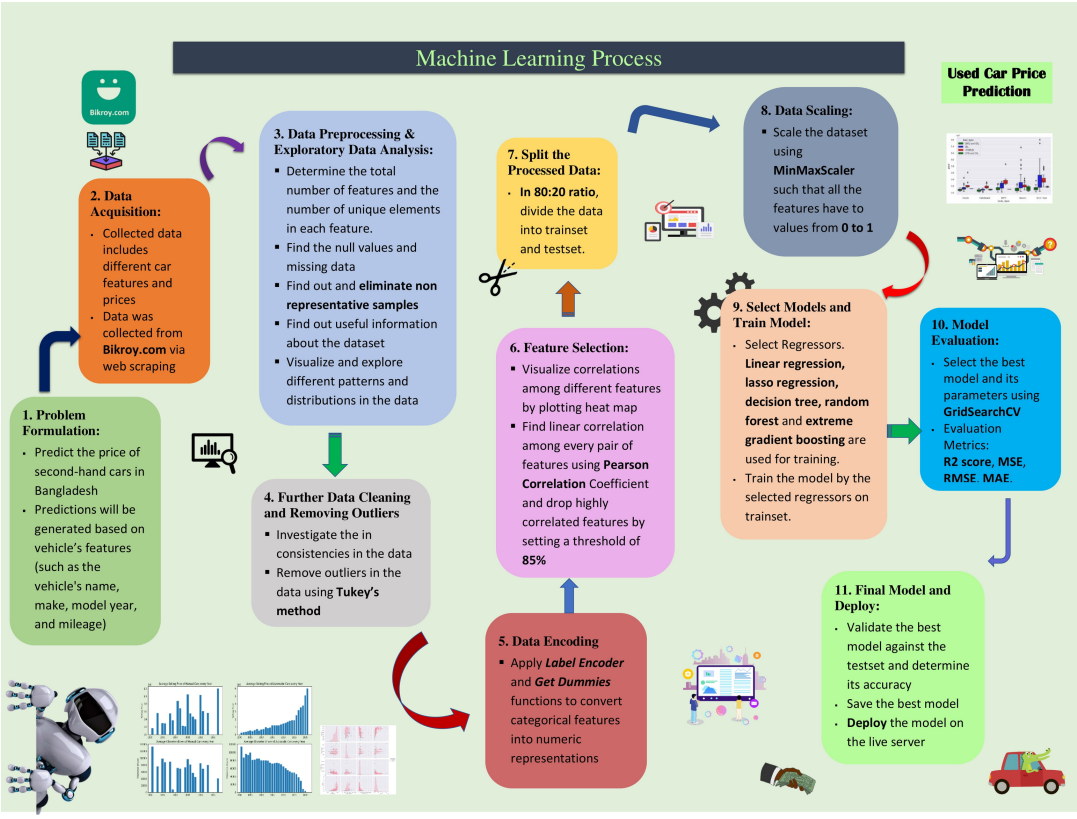


**Figure 1.** Flow Chart of Research Methodology.

### *3.1. Data Acquisition*

We collected the dataset by scraping https://bikroy.com/ (accessed on 13 October 2021) [24], which is a prominent online market place in Bangladesh. We chose this platform since it was one of the first platforms in Bangladesh for buying and selling pre-owned vehicles and had the largest and the most comprehensive collection of relevant data at the time of our data acquisition phase.

The data were collected through web scraping. Web scraping is the automated process of gathering data from a website using bots. We used a Google Chrome internet browser extension called Web Scraper (accessed on 13 October 2021) [25] to perform this task. Bikroy.com (accessed on 13 October 2021) has no objections to their website's data being scraped. The initial dataset has 1209 instances and 10 features (shown in Table 1) including the target class. The dataset along with the python codes that were used to generate results can be found in [26].

**Table 1.** Description of selected attributes.

| Variable Name | Description |
|---|---|
| car_name | Name of the car |
| brand | Car brand |
| car_model | Model of the car |
| ine model_year | Model year |
| transmission | Transmission (automatic or manual) |
| body_type | Body type |
| fuel_type | Fuel type |
| engine_capacity | Capacity of the engine (in cc) |
| kilometers_run | Kilometers run by the car |
| price | Price of the car (in taka) |

### *3.2. Pre-Processing*

The dataset collected may not always be in a format that is suitable for machine learning algorithms to operate on. To ensure that machine learning algorithms can be effectively used, the data pre-processing stage is crucial. Upon analyzing our initial dataset, we identified a few problems that needed to be addressed.

For instance, the feature *car_name* had 1108 unique values. In this case, the values did not adhere to any specific format and occasionally contained superfluous information. Additionally, because it was a nominal variable with an excessive number of unique values, this feature was ineffective for the purpose of forecasting. As a result, this feature was eliminated. Table 2 shows the unique values for each feature in the dataset.

**Table 2.** Number of unique values for all the features.

| Feature Name | Unique Values |
|---|---|
| car_name | 1108 |
| brand | 26 |
| car_model | 123 |
| model_year | 35 |
| transmission | 2 |
| body_type | 7 |
| fuel_type | 24 |
| engine_capacity | 51 |
| kilometers_run | 640 |

After removing the aforementioned feature, we investigated the missing entries per feature. Feature *body_type* had 18 missing entries. As the number of such instances was low compared to the number of records we had, we eliminated these instances.

Following that, we assessed the data for consistency. Three features, namely the *car_model*, *body_type*, and *fuel_type* were identified to require further pre-processing.

In feature *car_model*, there were several values that were unique, the majority of which occurred just once. As a result of this small number, non-representative samples were formed. Therefore, we eliminated these instances.

In feature *body_type*, there were only 2 instances of type *Convertible* (shown in Table 3). Due to the small number of data inside this category, there were insufficient instances to represent it. As a result, all occurrences falling under this category were eliminated.

Another challenge was to resolve the inconsistency in the values of feature *fuel_type*. In Bikroy.com, the feature *fuel_type* did not follow any specific format, and this resulted in the data being inconsistent. Instances that belonged to the same category were given input under different names. Additionally, a very small number of instances belonged to some categories (shown in Table 4). As they formed a non-representative sample, instances that belonged to these categories were eliminated as well.

**Table 3.** Value counts of feature body type.

| Body_Type | Value Counts |
|---|---|
| Saloon | 606 |
| MPV | 193 |
| SUV/4 × 4 | 183 |
| Estate | 77 |
| Hatchback | 76 |
| Convertible | 2 |

**Table 4.** Before processing the feature fuel type.

| Fuel_Type | Value Counts |
|---|---|
| CNG, Octane | 391 |
| Octane | 246 |
| Petrol, Octane | 116 |
| Hybrid, Octane | 93 |
| Petrol, Hybrid, Octane | 62 |
| Hybrid | 55 |
| Petrol, CNG, Octane | 52 |
| Petrol, CNG | 28 |
| Diesel | 24 |
| Octane, LPG | 22 |
| Petrol | 21 |
| CNG | 6 |
| Octane, Other fuel type | 4 |
| Petrol, Octane, LPG | 3 |
| Petrol, Other fuel type | 2 |
| LPG | 1 |
| Petrol, LPG | 1 |
| Petrol, Hybrid, Octane, LPG | 1 |
| Petrol, CNG, Octane, LPG | 1 |
| CNG, Hybrid | 1 |
| Petrol, Hybrid | 1 |
| Diesel, Petrol | 1 |
| Hybrid, Octane, LPG | 1 |
| Petrol, Octane, Other fuel type | 1 |

Finally, values of the feature *fuel_type* were mapped to four broader values according to their relevance. They are *CNG and Oil*, *Oil*, *Hybrid*, and *LPG and Oil*. For example: *Octane*, *Petrol*, *Diesel* has been mapped to a broader category *Oil*. Cars that run with both CNG and oil are mapped to category *CNG and Oil*. Other categories are likewise mapped in a similar manner (shown in Table 5).

**Table 5.** After processing the feature fuel type.

| Fuel_Type | Description | Value Counts |
|---|---|---|
| CNG and Oil | cars that run on both CNG and Oil | 477 |
| Oil | cars that run on oil | 411 |
| Hybrid | Hybrid cars | 210 |
| LPG and Oil | cars that run on both LPG and oil | 25 |

### *3.3. Exploratory Data Analysis*

By examining the pre-processed data, several trends in Bangladesh's used vehicle market are found. To begin, it became clear that cars with *automatic transmission* are considerably more prevalent than cars with *manual transmission* (shown in Figure 2).

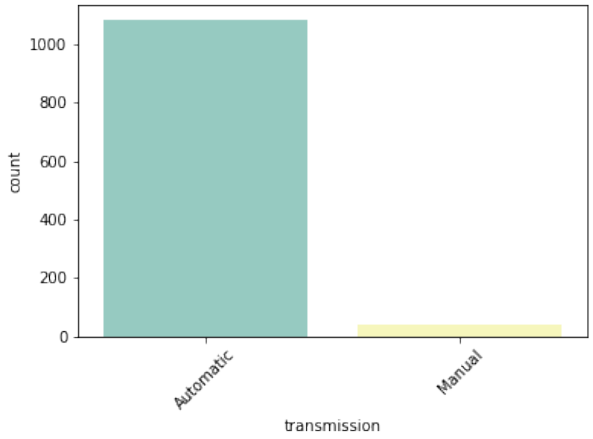


**Figure 2.** Cars with *Automatic* vs. *Manual Transmission*.

If *body_type* feature is considered, *Saloon* is the most prevalent in Bangladesh, followed by *MPV and SUV*, which are almost equally prevalent. In comparison, *Hatchback* and *Estate* are less commonly available (shown in Figure 3).

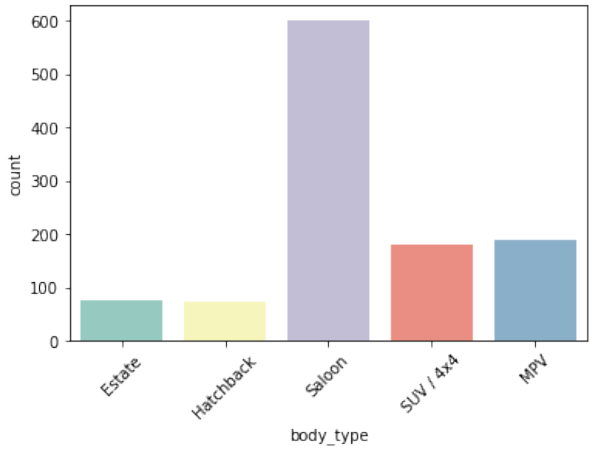


**Figure 3.** Number of cars with different *body_type* feature.

When *fuel_type* feature is taken into account, *CNG and Oil* is the most commonly configured type of car in Bangladesh. It is followed by *Oil*, *Hybrid* and *LPG and Oil* (shown in Figure 4).

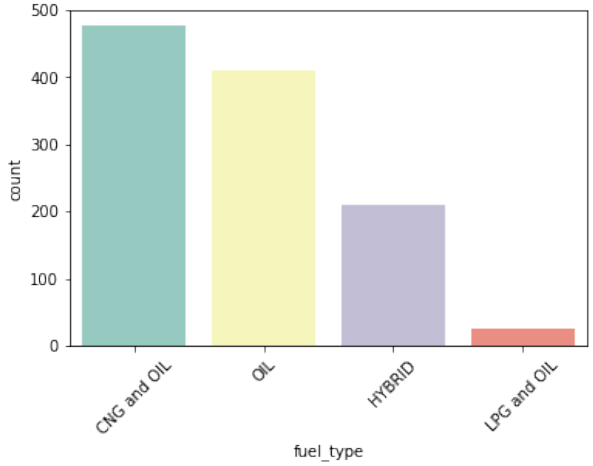


**Figure 4.** Number of cars with different *fuel_type* feature.

By analysis of the linear model plot, price is found to have a linear relation with *fuel_type* and *model_year* (shown in Figure 5).

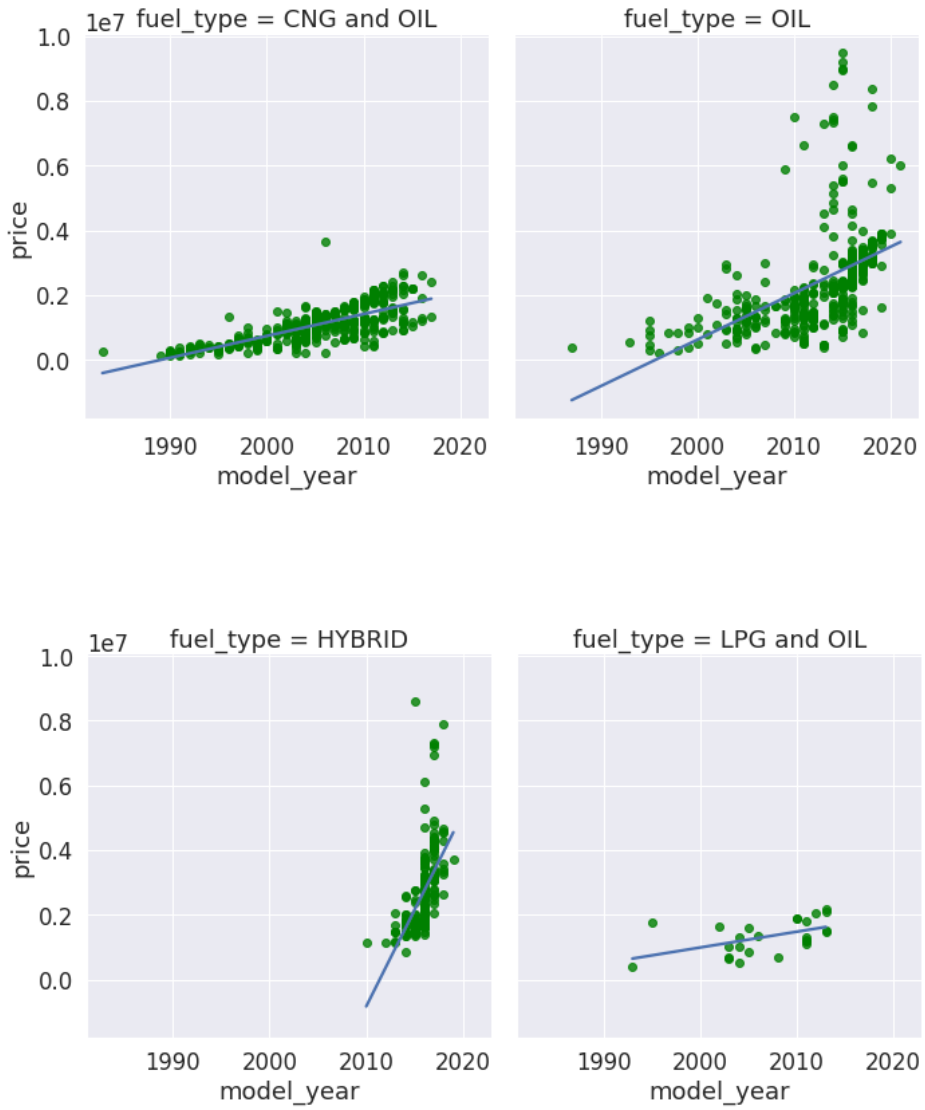


**Figure 5.** Linear model plot for *model_year* and *price*.

Box plots of various features (shown in Figure 6) give the following important information regarding the average price of used cars in Bangladesh:

1. If *transmission* is considered, cars with *automatic transmission* have a higher average price than cars with *manual transmission*.
2. If *body_type* is considered, *SUV* has the highest average price. It is followed by *MPV*, *Saloon*, *Estate*, and *Hatchback*.
3. If *fuel_type* is considered, *Hybrid* cars have highest average price.

All of the above observations corroborate our intuition.

Upon analyzing the facet grid (shown in Figure 7) for different *body_type* features against *model_year*, the following information can be derived:

1. Newer cars have a higher price for all *body_type*.
2. Within different *body_type*, cars with *automatic transmission* have a higher price than cars with *manual transmission*.
3. Some outliers are present for *Saloon* and *SUV* cars.

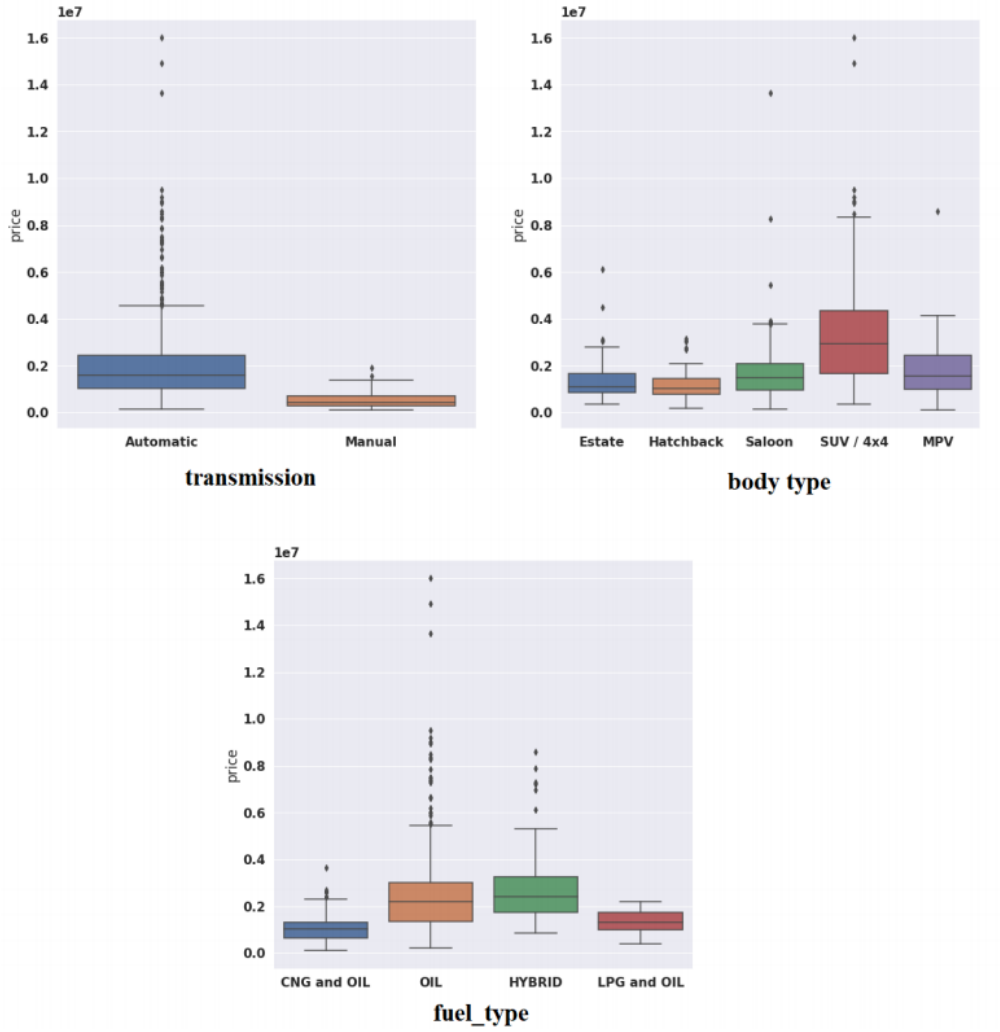


**Figure 6.** Box plot for *transmission*, *body_type*, *fuel_type*, and *model_year* features.

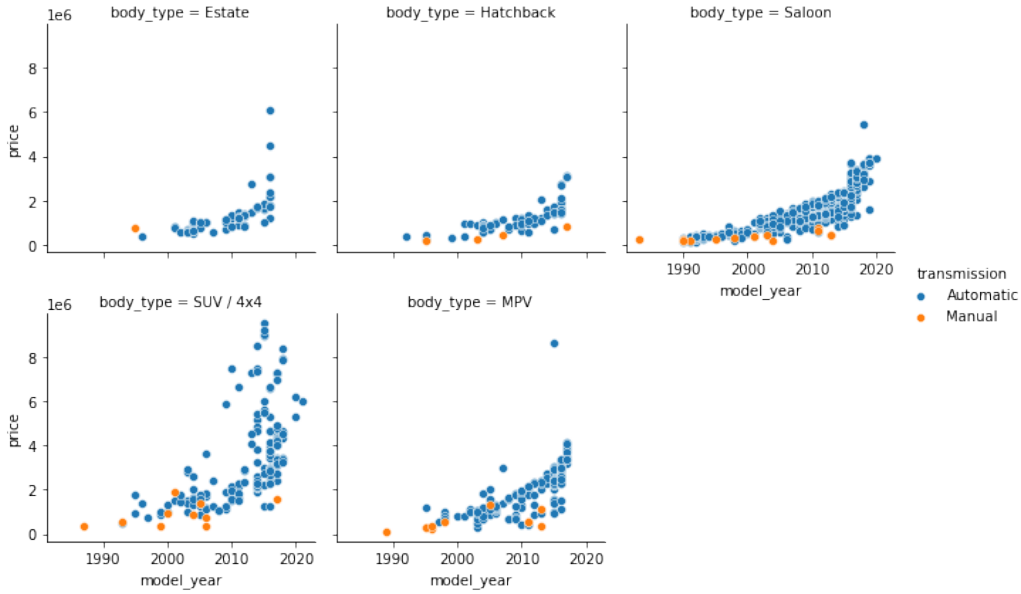


**Figure 7.** Facet grid for different *body_type* features against *model_year*.

Upon examining the box plots (shown in Figure 8) for different *body_type* and *fuel_type* features, more relevant information has been obtained. For instance, *Hybrid* cars, in general,

have been found to have a higher price for most *body_type* features. In the case of *Saloon body_type*, cars with *Oil fuel_type* have been observed to have a higher average price.

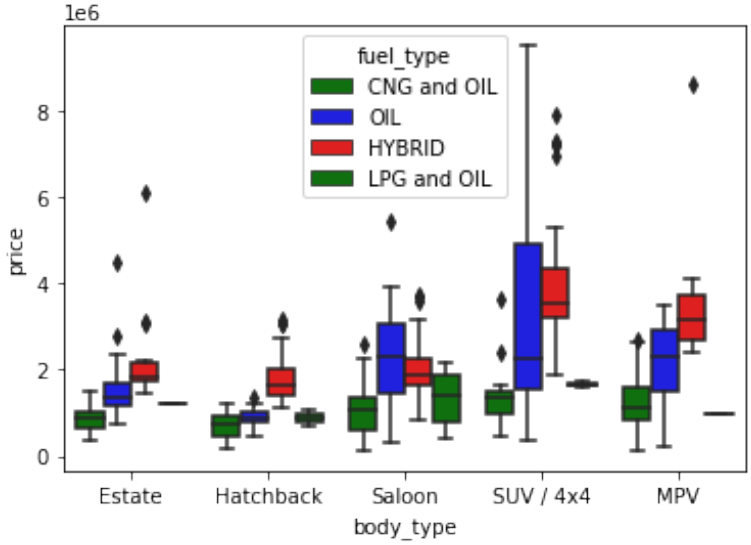


**Figure 8.** Box plots for different *body_type* and *fuel_type* features.

It has been previously established that newer cars have a higher average selling price. If we look at the *model_year* and selling price for *automatic* and *manual transmission* cars separately (shown in Figure 9), we can observe a noteworthy trend. That is, the selling price of cars with *automatic transmission* increases non-linearly for newer *model_year*. However, in the case of cars with *manual transmission*, although the average selling price has an upward trend, the bar chart is wavy for newer *model_year* due to the absence of occurrences for distinct *model_year* instances.

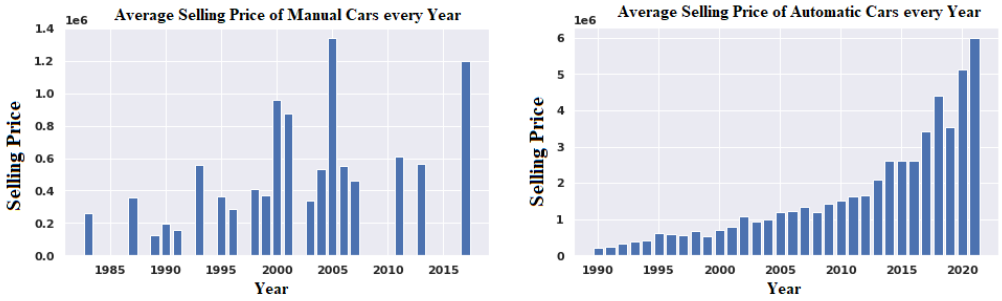


**Figure 9.** Average selling price of *automatic* and *manual* cars of each *model_year*.

By analyzing the pair plots, one can determine pair-wise correlation among the features. From the pair plot of cars of different *body_type*, we have observed that *Saloon* cars have a higher price range and higher *kilometers_run*. It is also observed that *SUV* type cars have a wider price range (shown in Figure 10).

By analyzing the pair plot of *automatic* and *manual transmission* cars, it is observed that although distribution of *automatic* cars are skewed to the right, these cars have a higher price range compared to *manual* type cars. We can also see a growth in the number of automatic cars between the years 2015 and 2020 (shown in Figure 11).

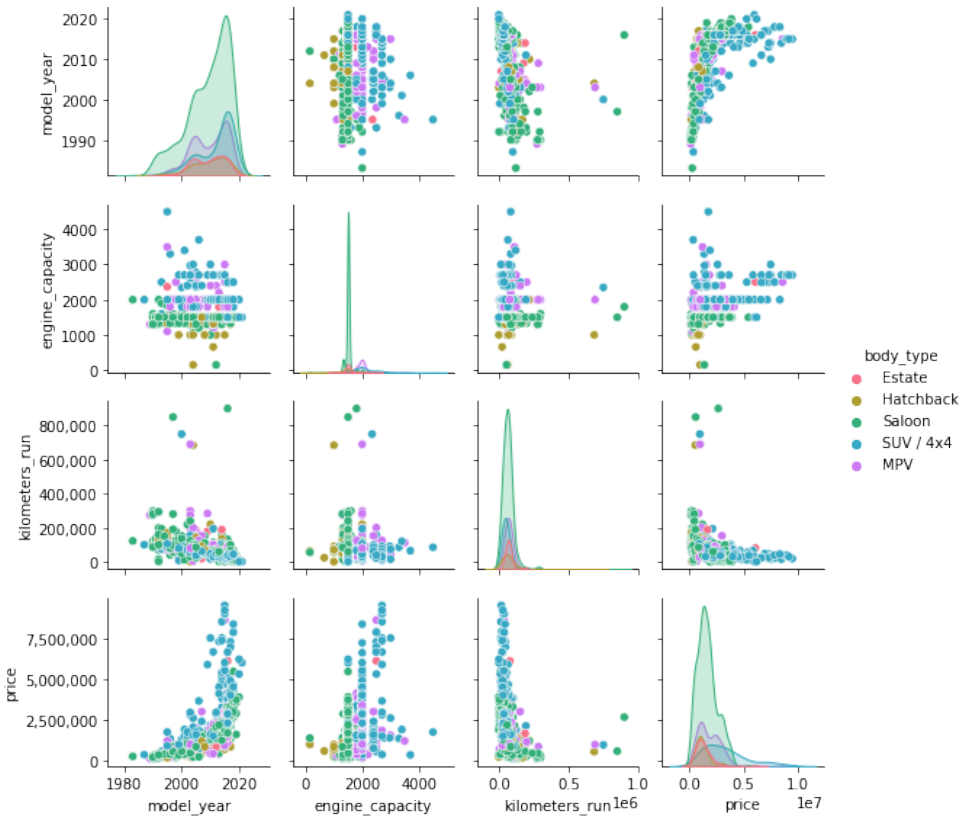


**Figure 10.** Pair plot of cars of different *body_type*.

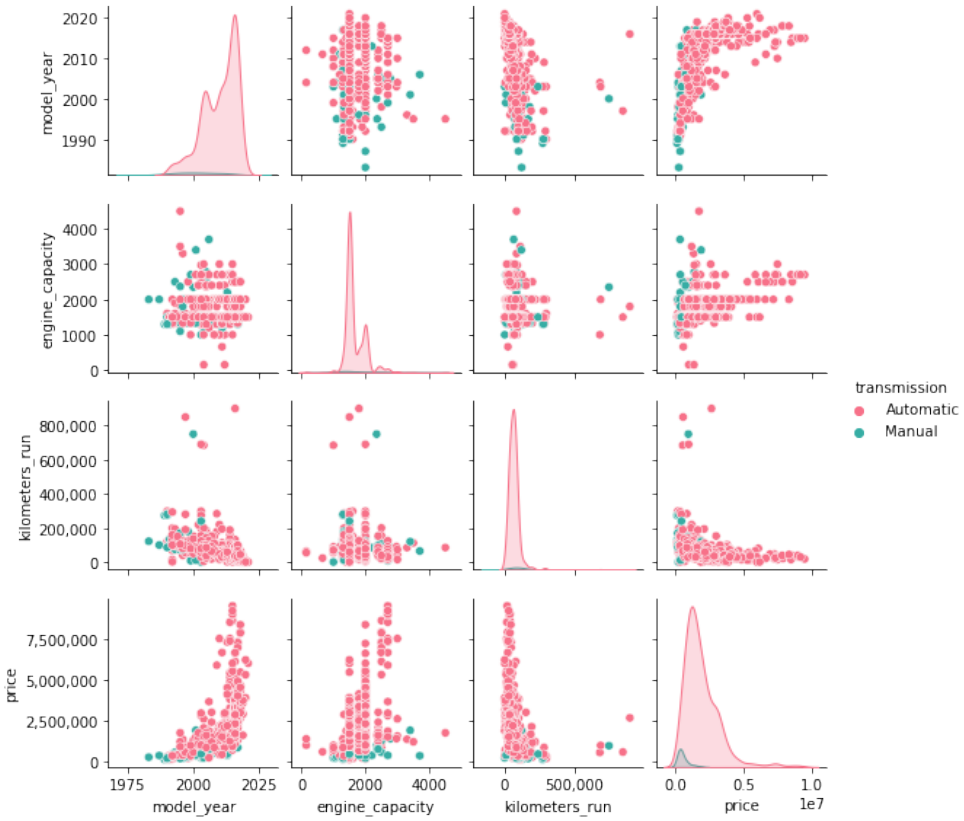


**Figure 11.** Pair plot of *automatic* and *manual* cars.

Finally, if we analyze the pair plot of *automatic* and *manual* cars with different *fuel_type* features, it is observed that the selling price of *Oil* type cars, in both *manual* and *automatic* models, are more spread than *CNG and Oil*, *Hybrid*, and *LPG and Oil* type cars. This means that *Oil*-type cars have a higher average price and a broader range of selling price (shown in Figures 12 and 13).

### *3.4. Further Data Cleaning and Removing Outliers*

While conducting exploratory data analysis, some outliers were identified. In feature *body_type*, outliers were present in *Saloon* and *SUV* (shown in Figure 7). In *Saloon* type, instances having a price higher than 7,500,000 are labeled as outliers. Such instances have been removed from the dataset. Similarly, instances that have a price higher than 12,500,000 in *SUV* have also been removed.

Some outliers were also discovered while analyzing *kilometers_run* in the distribution plot for *automatic* and *manual* cars (shown in Figure 14). Comparably, when the selling prices of *automatic* and *manual* cars were analyzed according to *model_year* (shown in Figure 9), such exceptions were again detected. These outliers were removed using Tukey's method [27].

The following steps are used to determine the outliers:

- Step 1: The lower and upper fences are calculated using Equations (1) and (2);
- Step 2: All data outside the range $[L_f, U_f]$ are treated as outliers.

$$L_f = Q_1 - 1.5 \times IQR \quad (1)$$

$$U_f = Q_3 + 1.5 \times IQR \quad (2)$$

where $IQR$ = interquartile range, which is equal to upper quartile—lower quartile.

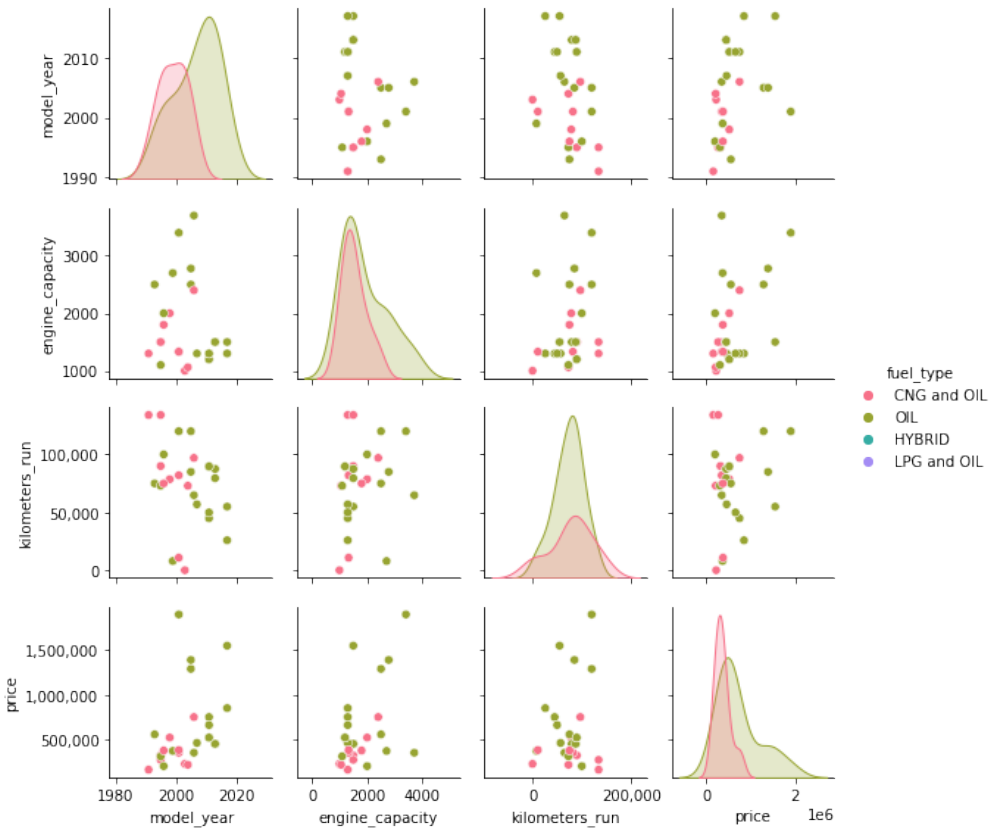


**Figure 12.** Pair plot of *manual* cars with different *fuel_type* features.

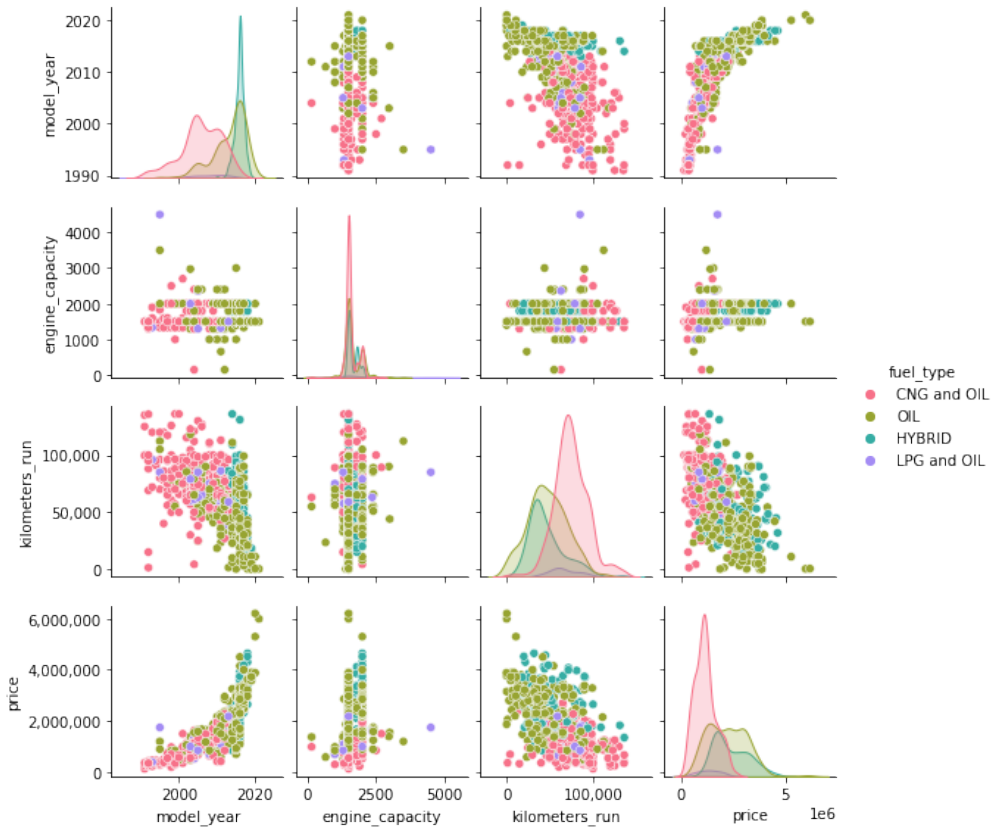


**Figure 13.** Pair plot of *automatic* cars with different *fuel_type* features.

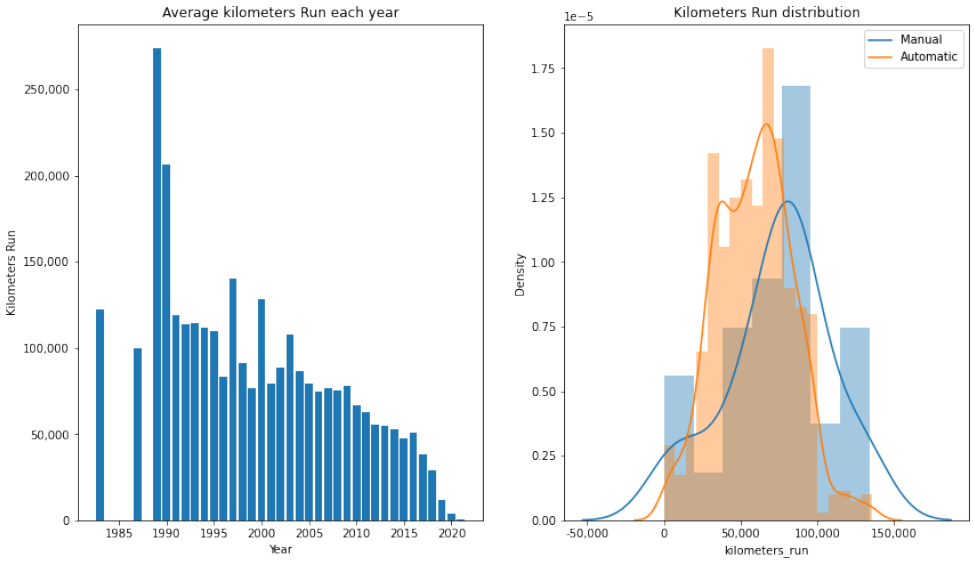


**Figure 14.** Average *kilometers_run* for each *model_year* and their distribution plot.

Finally, it was observed that there is discrepancy in the information pertaining to vehicles with *model_year* before 1990 (shown in Figure 14). As a result, these occurrences were eliminated as well.

After further cleaning, if we now plot the data (in Figure 15) and compare with Figures 9 and 14, we can see that the data have become significantly cleaner, with fewer outliers.

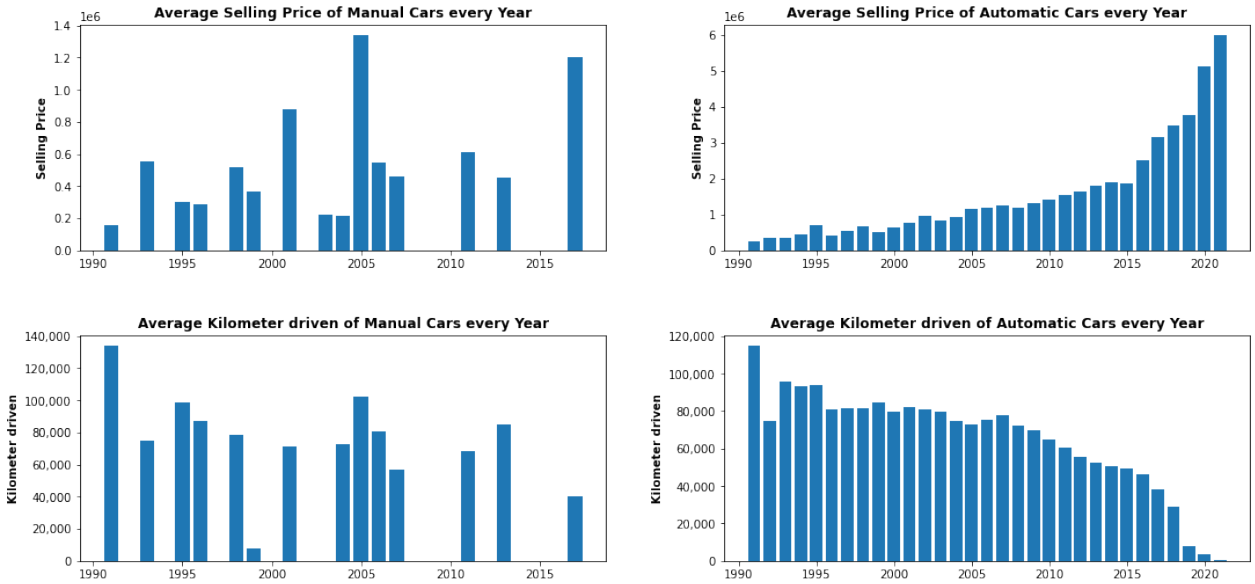


**Figure 15.** Average selling price and average *kilometers_run* for *automatic* and *manual* cars.

### *3.5. Data Encoding*

There were a number of categorical values such as: *brand*, *body_type*, *transmission*, *car_model* and *fuel_type* in our dataset. To be able to train our model, we created a numerical representation of these categorical values.

We used label encoding to get a numerical representation of the features whose number of distinct values exceeding 4. Label encoding means converting categorical features into numerical values. For other categorical features, we expanded the dimensions to convert them into numerical values.

In features *brand*, *car_model* and *body_type* there are several possible distinct values. Hence, we utilized label encoding to compress the data into integer numbers. We used LabelEncoder [28] from scikit-learn [29] to accomplish this task.

*Transmission* can only have two distinct values, and *body_type* can only have four distinct values. Therefore, we expanded the dimensions by using the same number of variables. We used the get_dummies [30] function from pandas [31] to accomplish this task. Pandas is a software library written for the Python programming language for data manipulation and analysis. After data encoding, we were left with a dataset containing 12 features.

### *3.6. Feature Selection*

When it comes to estimating the price of used cars, not every feature available to us will have the same effect. Certain characteristics may be more critical than others, while others may be irrelevant to the pricing. As a result, it is critical to deal with just the most prominent features. We have already removed one feature (*car_name*) during our pre-processing phase. We can visualize correlations of the remaining features from plotting a heatmap (shown in Figure 16).

Next, we measured the correlation between each pair of features to detect and accordingly drop the highly correlated ones. High correlation between a pair of features indicates that the two features carry the same information, and thus it is not required to have both of them. We used the Pearson correlation co-efficient [32] to measure the strength and direction of a linear relationship between two features. Setting a threshold of 85%, we searched for the highly correlated features from the correlation matrix. The feature *Manual* was removed because it was found to be highly correlated with feature *automatic*.

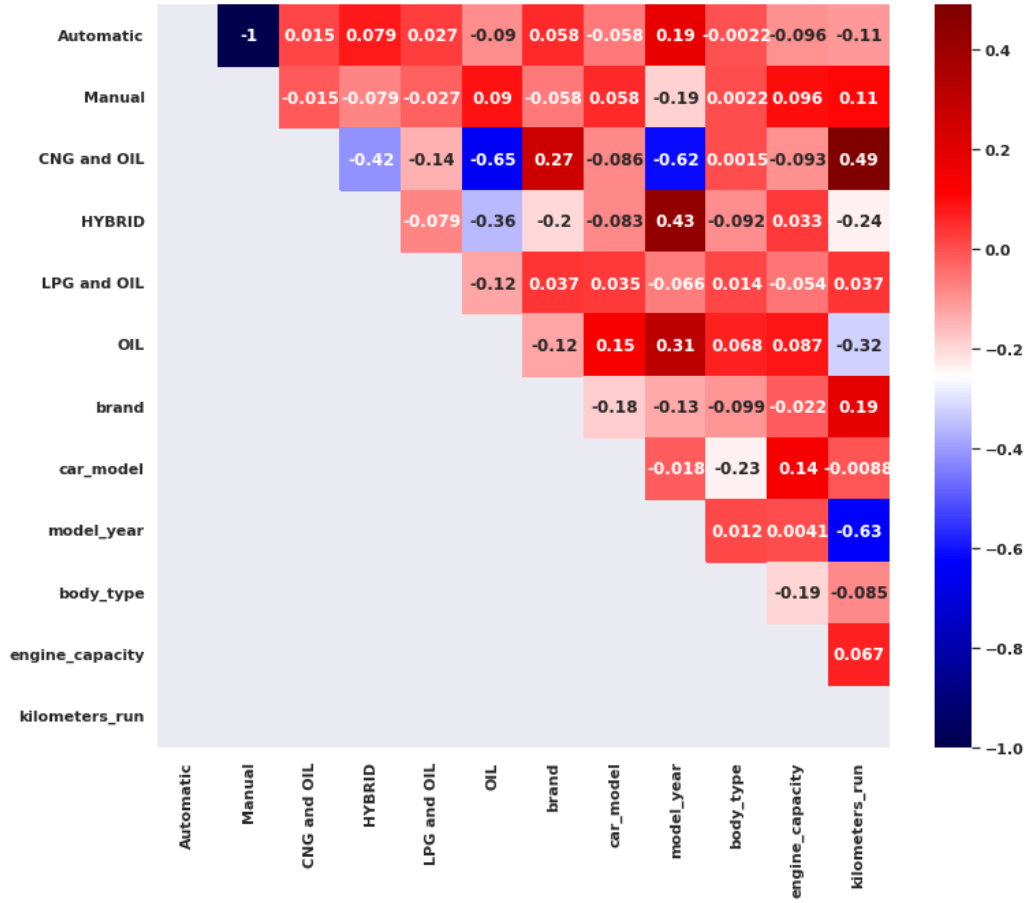


**Figure 16.** Heat map between every pair of features.

The following algorithm (Algorithm 1) is used to detect highly correlated features:

**Algorithm 1** Algorithm for finding correlated features.

```
function CORRELATION(dataset, threshold)
    correlated_features ← set()                          ▷ Set of correlated features
    correlation_matrix ← dataset.corr()   ▷ Correlation matrix using Pearson correlation
    for every distinct pair of features do
        if difference in the correlation matrix > threshold then
            correlated_features.insert(feature_name)
        end if
    end for
    return correlated_features
end function
```

### *3.7. Data Splitting*

It is critical to divide the dataset into train and test sets throughout the machine learning process. By segmenting the data, we may assess how effectively a model responds to new data. By separating the test and train data, the model will be unable to view the test data in advance. This enables us to assess the regressors' performance accurately while evaluating them on the test set. Our dataset was divided in a 80:20 ratio. Additionally, we partitioned the dataset with a fixed value for the hyper-parameter *random_state* to ensure that the results obtained are reproducible.

### *3.8. Data Scaling*

In our dataset, there are different features with different ranges of values. These differences may lead to the dominance of some features over others while training the data. In order to avoid this situation, it is important to scale our dataset. We used the MinMaxScaler [33] function from Scikit-learn library [34] to scale our dataset. By doing so, all features were transformed into the range [0, 1] meaning that the minimum and maximum value of a feature is going to be 0 and 1, respectively. The features that were scaled are: *Automatic*, *CNG and Oil*, *Hybrid*, *LPG and Oil*, *Oil*, *body_type*, *engine_capacity*, and *kilometers_run*. The formula for min max scaling (Equation (3)) is as follows:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3}$$

The data scaling procedure is shown in Figure 17.

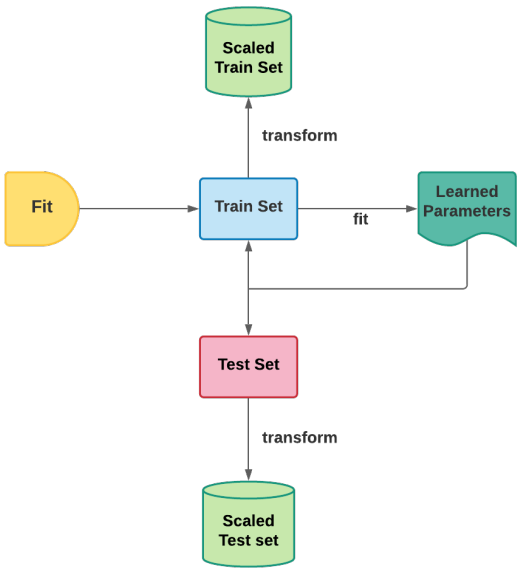


**Figure 17.** Data scaling procedure.

We utilized the fit operation on the training set to learn the parameters. We transformed the training set into a scaled training set by applying these learned parameters to it. Additionally, we used the same learned parameters to turn the test set into a scaled test set.

### *3.9. Regressors Used*

In this study, five regression algorithms were used in total. They are linear regression, lasso regression, decision tree, Random Forest, and extreme gradient boosting.

#### 3.9.1. Linear Regression

Linear Regression [35] is a regression model that assumes a linear relationship between the features ($X$) and the target class ($y$). With this model, target class is derived from the linear combination of the input variables. Based on number of variables, it can be a univariate linear regression or multivariate linear regression.

A linear regression line has an Equation (Equation (4)) of the following form:

$$\hat{y} = a + bX \tag{4}$$

Here, $\hat{y}$ is the predicted dependent variable and X is the independent variable. The slope of the line is $b$ and $a$ is the intercept.

#### 3.9.2. LASSO Regression

LASSO (Least Absolute Shrinkage and Selection Operator) [36] is a type of linear regression model that uses shrinkage. Shrinkage is the central point where the data are shrunk like the mean. This regression is also called L1 regularizer and is particularly effective when there are fewer parameters and high multicollinearity.

The goal of LASSO regression is to minimize the following (Equation (5)):

$$Cost(\beta) = \sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \quad (5)$$

Here, $\lambda$ denotes the amount of shrinkage.

#### 3.9.3. Decision Tree

Decision Tree [37] is a supervised machine learning technique that builds regression or classification models. It uses a tree-like structure where the leaf nodes are the outcomes. All the other nodes except the leaf nodes are called decision nodes where further splits are made depending on yes/no questions. The goal of the decision tree is to create a model that can be used to predict the value of target variable by learning simple decision rules from the previous data. How a decision tree splits the data is often determined by entropy or Gini index. In this work, we used Gini index to split the data in decision tree.

Following equation show the formula for Gini index (Equation (6)):

$$G(S) = 1 - \sum_{i=1}^{c}(p_i)^2 \quad (6)$$

where $S$ is the subset of the training data and $p_i$ is the probability of the class.

#### 3.9.4. Random Forest

Random Forest [38] is a ensemble learning method that uses multiple decision trees in order to create classification or regression model. Random forest consists of a large number of decision trees that work as an ensemble. Each individual trees predict the value of the target class, and their predictions are combined in order to get more accurate prediction. Random forest is a supervised machine learning algorithm that can be used to solve both classification and regression problems.

#### 3.9.5. Extreme Gradient Boosting

Extreme Gradient Boosting is based on Gradient Boosting algorithm [39] which is a machine learning algorithm that is used to solve both classification and regression problems. XGBoost is an open-source library that provides an efficient and effective implementation of the gradient boosting algorithm. It uses several weak prediction models and combines them to improve the performance. Typically, multiple decision trees are used. In extreme gradient boosting, a more regularized model is used to control overfitting, which further improves its performance.

## 4. Results

The dataset was trained with the aforementioned five regressors. To determine which regressors perform the best, they were assessed against a variety of performance indicators. We evaluated the regressors' performance using three distinct metrics. They are: $R^2$ score, Root Mean Squared Error, and Mean Absolute Error.

$R^2$ score measures how close the data fit with the regression line. It is closely related to Mean Squared Error. Following Equation (Equation (7)) is used to calculate $R^2$ score:

$$R^2 = 1 - \frac{RSS}{TSS} \quad (7)$$

Here, $RSS$ = sum of squares of residuals, and $TSS$ = total sum of squares.

Root Mean Squared Error is the standard deviation of the prediction errors. It is calculated using the following Equation (Equation (8)):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - a_i)^2} \tag{8}$$

Here, $p_i$ is the predicted value and $a_i$ is the actual value.

Mean Absolute Error is a measure of average distance between the real data and the predicted data. It is calculated by using the following Equation (Equation (9)):

$$\text{MAE} = \frac{\sum_{i=1}^{n}|p_i - a_i|}{n} \tag{9}$$

We determined the best parameters for each regressor using hyper parameter tuning. We accomplished this by using GridSearchCV [40] from scitit-learn [29]. GridSearchCV performs an exhaustive search over specified parameter values over an estimator. Scores and best parameters for each regressor are shown in the following table (Table 6).

To measure the overall performance of each regressor, five-fold cross validation is used. The data set is divided into five folds in this case. The first iteration uses the first fold to test the model, while the remaining folds are utilized to train the model. The second iteration uses the second fold as the testing set and the remaining folds as the training set. This process is repeated until each fold of the five folds has been used as the testing set. Our results indicate that the RMSE, and MAE values were excessively large. Thus, we took the logarithm of these numbers to make them more concise and easy to compare.

$R^2$ Score, Log Root Mean Squared Error, and Log Mean Absolute Error for each regressors are summarized in the table below (Table 7). The best performances have been highlighted in bold font.

**Table 6.** Parameter Space and Best Parameters of Models.

| Model | Parameter Space | Best Parameters |
|---|---|---|
| Linear Regression | normalize: True, False | normalize: False |
| Lasso Regression | alpha: 1, 2<br>selection: random, cyclic | alpha: 2<br>selection: random |
| Decision Tree | criterion: mse,<br>friedman_mse<br>max_depth: 1–21<br>splitter: best,<br>random | criterion: mse<br>max_depth: 9<br>splitter: best |
| Random Forest | criterion: mse,<br>friedman_mse, mae<br>n_estimators: 0, 5, 10, ... 100 | criterion:<br>friedman_mse<br>n_estimators: 60 |
| XGBoost | colsample_bytree: 0–1<br>criterion: mse,<br>friedman_mse, mae<br>eta: 0.1–0.01<br>max_depth: 1–6<br>n_estimators: 0, 5, 10, ... 100 | colsample_bytree: 0.8<br>criterion: mse<br>eta: 0.1<br>max_depth: 5<br>n_estimators: 95 |

**Table 7.** Performance of classifiers on test set.

| Model | $R^2$ Score (%) | Log RMSE (%) | Log MAE (%) |
|---|---|---|---|
| Linear Reg. | $70.48 \pm 3.75$ | 13.15 | 12.57 |
| Lasso Reg. | $70.56 \pm 3.75$ | 13.15 | 12.55 |
| Decision Tree | $85.36 \pm 4.17$ | 12.80 | 11.77 |
| Random Forest | $90.14 \pm 1.96$ | 12.60 | **11.63** |
| XGBoost | **91.32 ±1.84** | **12.53** | 11.72 |

Empirical results show that XGBoost has the highest $R^2$ score followed by Random Forest, which is quite close to that of XGBoost. Decision tree has a moderate score, while linear and lasso regressions have comparatively low scores. It may be noted here that in terms of the Log Mean Absolute Error (LMAE) metric, the Random Forest performs better than XGBoost.

To compare the $R^2$ scores of the regressors, a bar plot is given below (Figure 18):

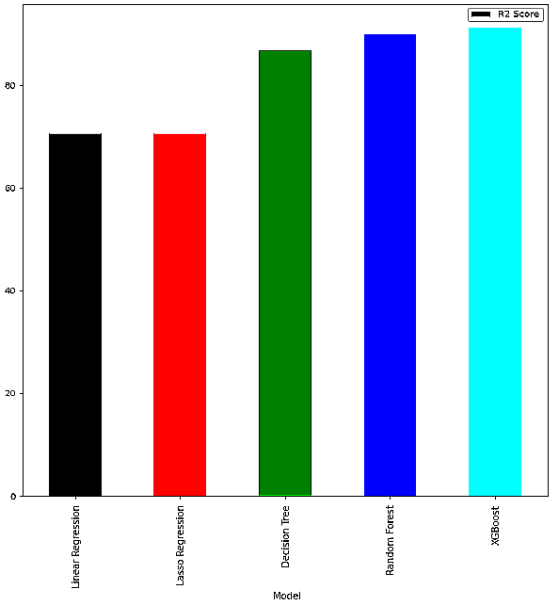


**Figure 18.** Bar plot comparing $R^2$ scores of different regressors.

Comparison among Log Root Mean Squared Error and Log Mean Absolute Error scores of the classifiers are plotted below (Figure 19):

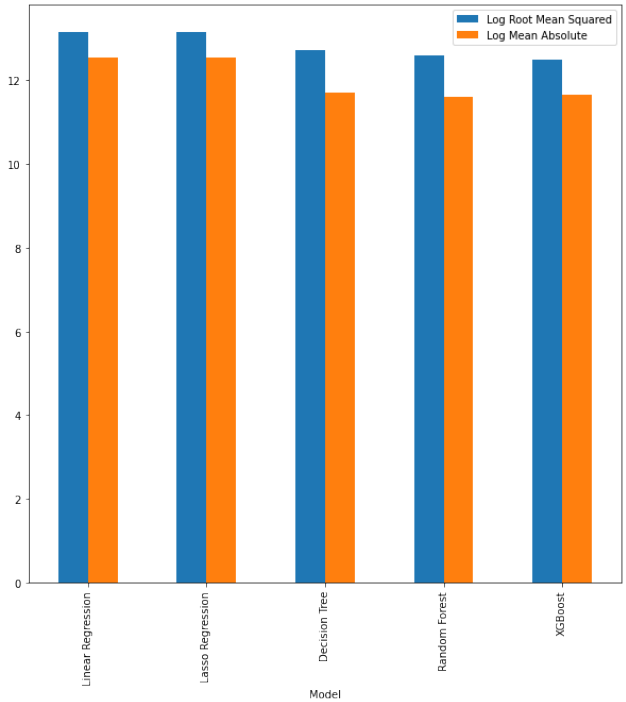


**Figure 19.** Comparison among log root mean squared error and log mean absolute error scores of the classifiers.

*Investigation of the XGBoost Model on Price Estimation*

To investigate how well the XGBoost model predicts price of a car, we collected 100 fresh samples from bikroy.com. These samples had not been used in the training phase—this does not make the prediction biased. The relevant features of these records are taken as input, and XGBoost model is used to estimate the prices of these records. Figure 20 shows how the predicted price varies with the actual price. The Pearson's correlation between the predicted and the actual price is found to be 0.988, which indicates that the model has been very effective in predicting the price of the cars.

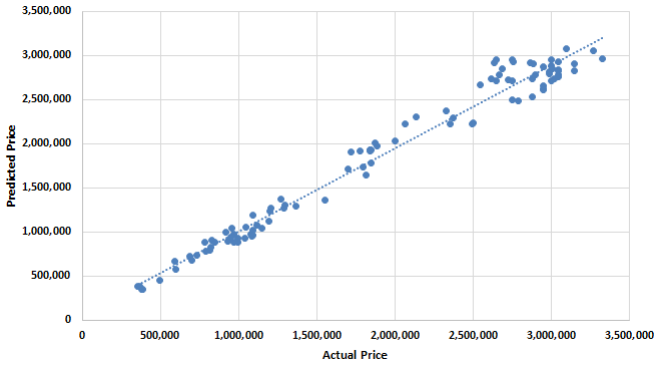


**Figure 20.** Predicted price versus actual price of 100 records.

In order to investigate whether the price of a car is overestimated or underestimated by the model, we follow a very simple rule. If the predicted price of a car is found to be larger than the actual price, then the estimation made is treated as overestimation. However, if the price predicted by the model is found to be smaller than the actual price, then it is treated as underestimation. Since the probability of predicted price to be exactly equal to actual price is almost zero, we will always treat a prediction to be either overestimation or underestimation. The absolute price differences between the predicted and actual cars are then determined. Table 8 shows how the absolute percentage difference in price is distributed. The absolute percentage difference varies between 0.11% and 12.95%. Figure 21 shows the distribution in a stacked bar chart.

**Table 8.** Overestimation and underestimation of the predicted price.

| Absolute Percentage Difference | Number of Cars | Number of Overestimation | Number of Underestimation |
|---|---|---|---|
| 0–4 | 30 | 14 | 16 |
| 4–8 | 35 | 17 | 18 |
| 8–12 | 30 | 8 | 22 |
| 12–13 | 5 | 2 | 3 |

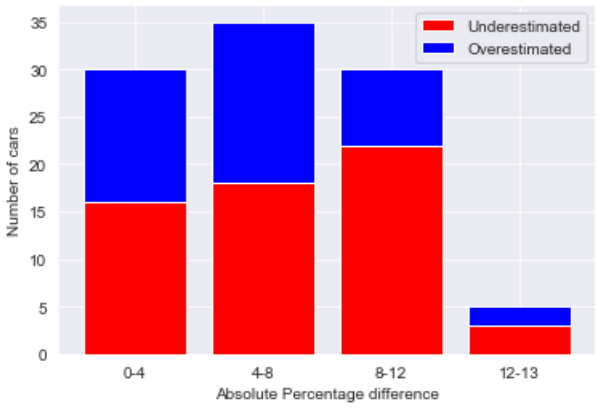


**Figure 21.** Number of overestimated and underestimated cars.

## 5. Deployment of the Model

The architecture of the deployment model is given in Figure 22. In order to deploy our model, an HTML page (Shown in Figure 23) was designed with the following inputs: *brand*, *model_year*, *body_type*, *transmission*, *fuel_type*, *engine_capacity* and *mileage*. When an input field is submitted, the API which was developed in Python [41] using Flask [42], gets a POST request with the provided inputs.

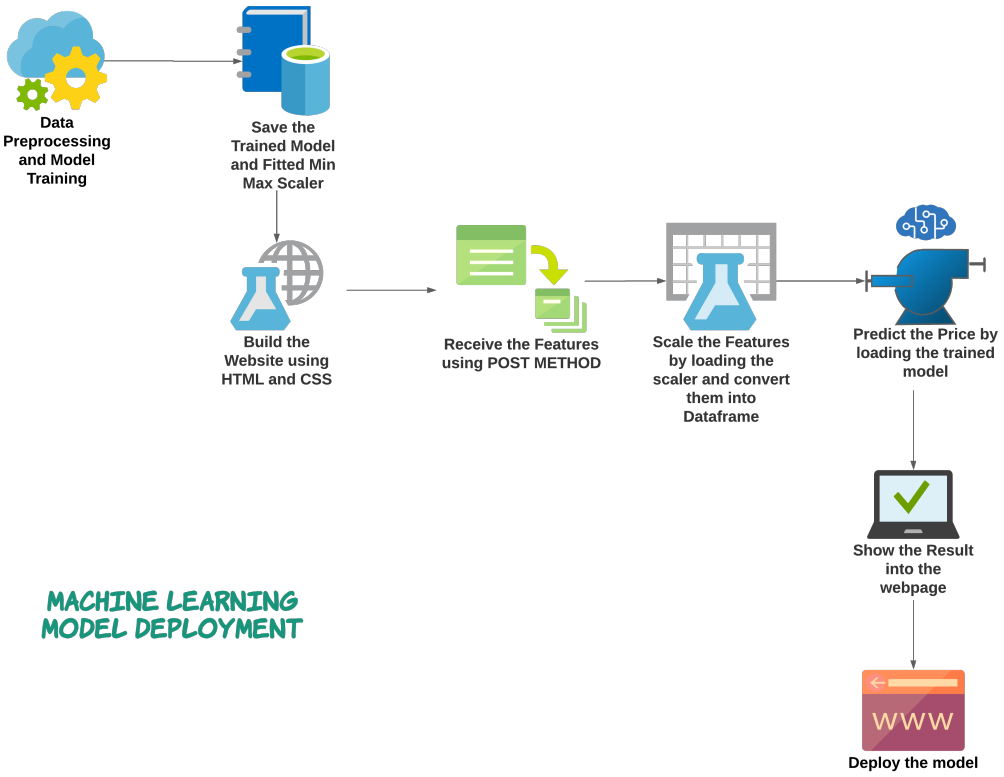


**Figure 22.** Architectural diagram of the deployment model.

These input fields were converted to a NumPy [43] ndarray, which was then transformed into a dataframe. Then, we loaded the pre-trained model XGBoost, which we previously designated as the most accurate. After that, the model was used to forecast the price using the dataframe. Following prediction, the output was utilized to render another HTML page that displays the outcome as a price (Shown in Figure 24).

To summarize, the deployment phase comprises the following stages:

1. The XGBoost model is trained with the pre-processed data using the fine-tuned hyperparameters. Then, this model and the fitted min–max scalar are saved.
2. The given inputs collected from the HTML web page are passed to the Python Flask API via a POST request.
3. The saved model and the fitted min–max scalar are loaded.
4. The inputs are processed and prepared with the help of the loaded min-max scalar within the API.
5. The model is imported and populated with the processed inputs. The model generates a prediction for us, which is displayed in another HTML web page.

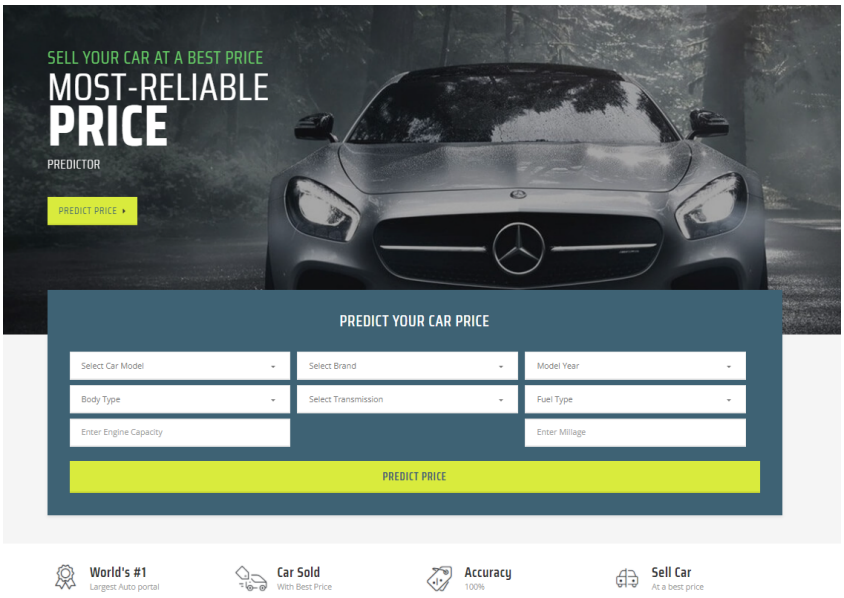


**Figure 23.** Screenshot of the webpage.

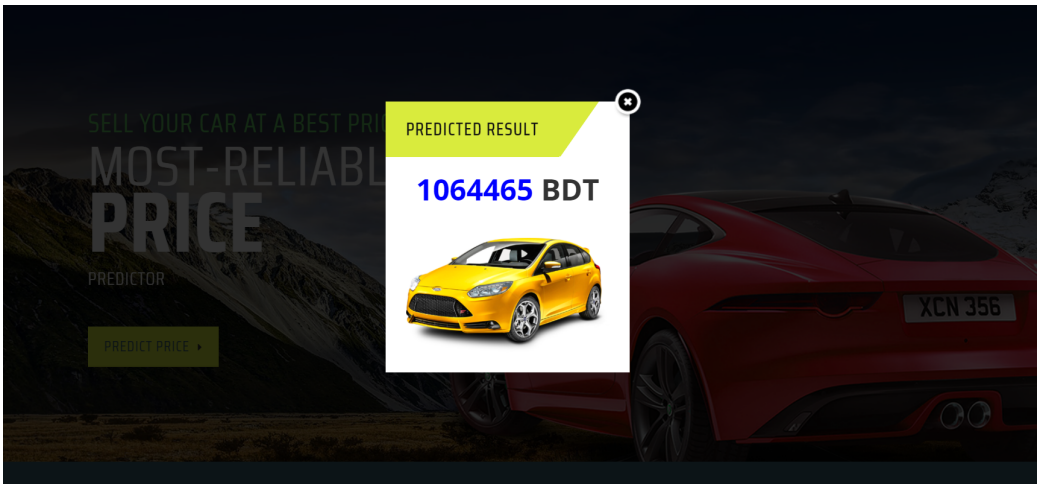


**Figure 24.** Predicted result.

## 6. Conclusions

This article discusses a machine learning approach for forecasting the price of pre-owned cars that is applicable to the Bangladeshi context. By removing outliers and irrelevant characteristics from the dataset, the most important features for this prediction were determined to be *brand*, *car model*, *model year*, *transmission*, *body type*, *fuel type*, *engine capacity*, and *kilometers run*. The dataset used in this research was collected from Bikroy.com, and using exploratory data analysis, previously undiscovered market trends and characteristics were discovered. After experimenting with a variety of machine learning algorithms on our obtained data, we discover that XGBoost, with an $R^2$ score of $(91.32 \pm 1.84)\%$, is the best effective regressor for forecasting resale prices of pre-owned cars. Additionally, our model was deployed as a web application on a local machine, from which it may subsequently be made available to end users in the future. Furthermore, in our future work, we plan to collect a much larger dataset with greater number of relevant features.

**Author Contributions:** Conceptualization, S.M., F.R.A., A.L.; data curation, F.R.A., A.L., S.M.; formal analysis, F.R.A., A.L., S.M., A.I.; investigation, F.R.A., A.L., S.M., A.I.; methodology, F.R.A., A.L., S.M., A.I.; software, F.R.A.; supervision, S.M. All authors read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available in github repository [26].

**Conflicts of Interest:** The authors declare no conflict of interest.

## References


1. Bangladesh Population. Available online: http://srv1.worldometers.info/world-population/bangladesh-population/ (accessed on 13 October 2021).
2. Haq, R.A. A brief look at the auto industry in Bangladesh. *The Daily Star*, 13 February 2021.
3. Bank, W. Bangladesh Development Update. April 2013. Available online: https://openknowledge.worldbank.org/handle/10986/16497 (accessed on 13 October 2021).
4. Imam, S.H. Bangladesh surpasses India on per capita income. *The Financial Express*, 24 May 2021.
5. Islam, S.; Huda, E.; Nasrin, F.; Freelanch Researcher, M. Ride-sharing Service in Bangladesh: Contemporary States and Prospects. *Int. J. Bus. Manag.* **2019**, *14*, 65–75. [CrossRef]
6. Holy, I.J. Bangladesh Automotive Industry: A Roadmap to the Future. 2020. Available online: https://www.lightcastlebd.com/insights/2020/07/bangladesh-automotive-industry-a-roadmap-to-the-future (accessed on 13 October 2021).
7. Hasan, M. Reconditioned car imports take a nosedive: Industry people cite high tariff, rising trend of ridesharing as major factors. *Dhaka Tribune*, 21 September 2019.
8. Ahmmed, M.; Ullah, M.H. Analysis of the National Budget of Bangladesh 2010–2011: Excellencies and Constraints. 2019. Available online: https://research.usc.edu.au/esploro/outputs/journalArticle/Analysis-of-the-National-Budget-of/99451299902621 (accessed on 13 October 2021).
9. Anik, S.S.B. Budget FY19: Used car prices may rise, hybrid cars to become cheaper. *Dhaka Tribune*, 9 June 2018.
10. Anwari, N.; Ahmed, M.T.; Islam, M.R.; Hadiuzzaman, M.; Amin, S. Exploring the travel behavior changes caused by the COVID-19 crisis: A case study for a developing country. *Transp. Res. Interdiscip. Perspect.* **2021**, *9*, 100334. [CrossRef]
11. Lessmann, S.; Voß, S. Car resale price forecasting: The impact of regression method, private information, and heterogeneity on forecast accuracy. *Int. J. Forecast.* **2017**, *33*, 864–877. [CrossRef]
12. Mackenzie, A. The production of prediction: What does machine learning want? *Eur. J. Cult. Stud.* **2015**, *18*, 429–445. [CrossRef]
13. Listiani, M. Support Vector Regression Analysis for Price Prediction in a Car Leasing Application. Unpublished. 2009. Available online: https://www.ifis.uni-luebeck.de/~moeller/publist-sts-pw-andm/source/papers/2009/list09.pdf (accessed on 13 October 2021).
14. Pal, N.; Arora, P.; Kohli, P.; Sundararaman, D.; Palakurthy, S.S. How much is my car worth? A methodology for predicting used cars' prices using random forest. In Proceedings of the Future of Information and Communication Conference, Singapore, 5–6 April 2018; pp. 413–422.
15. Gajera, P.; Gondaliya, A.; Kavathiya, J. Old Car Price Prediction With Machine Learning. *Int. Res. J. Mod. Eng. Technol. Sci.* **2021**, *3*, 284–290.
16. Venkatasubbu, P.; Ganesh, M. Used Cars Price Prediction using Supervised Learning Techniques. *Int. J. Eng. Adv. Technol. (IJEAT)* **2019**, *9*, 216–223.
17. Monburinon, N.; Chertchom, P.; Kaewkiriya, T.; Rungpheung, S.; Buya, S.; Boonpou, P. Prediction of prices for used car by using regression models. In Proceedings of the 2018 5th International Conference on Business and Industrial Research (ICBIR), Bangkok, Thailand,17–18 May 2018; pp. 115–119.
18. Gegic, E.; Isakovic, B.; Keco, D.; Masetic, Z.; Kevric, J. Car price prediction using machine learning techniques. *TEM J.* **2019**, *8*, 113.
19. Autopijaca. Available online: https://www.autopijaca.ba/ (accessed on 13 October 2021).
20. Samruddhi, K.; Kumar, R.A. Used Car Price Prediction using K-Nearest Neighbor Based Model. *Int. J. Innov. Res. Appl. Sci. Eng. (IJIRASE)* **2020**, *4*, 629–632.
21. Rathee, G.; Sharma, A.; Iqbal, R.; Aloqaily, M.; Jaglan, N.; Kumar, R. A blockchain framework for securing connected and autonomous vehicles. *Sensors* **2019**, *19*, 3165. [CrossRef] [PubMed]
22. Dhiman, G.; Oliva, D.; Kaur, A.; Singh, K.K.; Vimal, S.; Sharma, A.; Cengiz, K. BEPO: A novel binary emperor penguin optimizer for automatic feature selection. *Knowl.-Based Syst.* **2021**, *211*, 106560. [CrossRef]
23. Dhiman, G.; Singh, K.K.; Soni, M.; Nagar, A.; Dehghani, M.; Slowik, A.; Kaur, A.; Sharma, A.; Houssein, E.H.; Cengiz, K. MOSOA: A new multi-objective seagull optimization algorithm. *Expert Syst. Appl.* **2021**, *167*, 114150. [CrossRef]
24. Bikroy.com. Available online: https://bikroy.com/ (accessed on 13 October 2021).
25. Web Scraper. Available online: https://chrome.google.com/webstore/detail/web-scraper-free-web-scra/jnhgnonknehpejjnehehllkliplmbmhn?hl=en (accessed on 13 October 2021).
26. Dataset and Codes. Available online: https://github.com/Amik-TJ/cse_445_used_car_price_prediction_using_machine_learning/tree/main/Experiment_Notebook_Dataset (accessed on 8 December 2021).

27. Seo, S. A Review and Comparison of Methods for Detecting Outliers in Univariate Data Sets. Ph.D. Thesis, University of Pittsburgh, Pittsburgh, PA, USA, 2006.
28. LabelEncoder. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html (accessed on 13 October 2021).
29. Scikit-Learn. Available online: https://scikit-learn.org/stable/ (accessed on 13 October 2021).
30. Get_dummies. Available online: https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html (accessed on 13 October 2021).
31. Pandas. Available online: https://pandas.pydata.org/ (accessed on 13 October 2021).
32. Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–4.
33. MinMaxScaler. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html (accessed on 13 October 2021).
34. Bisong, E. Introduction to Scikit-learn. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 215–229.
35. Montgomery, D.C.; Peck, E.A.; Vining, G.G. *Introduction to Linear Regression Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2021.
36. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B* **1996**, *58*, 267–288. [CrossRef]
37. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
38. Liaw, A.; Wiener, M. Classification and regression by randomForest. *R News* **2002**, *2*, 18–22.
39. Friedman, J.H. Stochastic gradient boosting. *Comput. Stat. Data Anal.* **2002**, *38*, 367–378. [CrossRef]
40. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
41. Oliphant, T.E. Python for scientific computing. *Comput. Sci. Eng.* **2007**, *9*, 10–20. [CrossRef]
42. Aslam, F.A.; Mohammed, H.N.; Mohd, J.M.; Gulamgaus, M.A.; Lok, P. Efficient way of web development using python and flask. *Int. J. Adv. Res. Comput. Sci.* **2015**, *6*, 54–57.
43. Oliphant, T.E. *A Guide to NumPy*; Trelgol Publishing: Spanish Fork, UT, USA, 2006; Volume 1.