Özyeğin University

CS551 – Data Science with Python
Assignment-1

Car Pricing Prediction by Linear Regression

Furkan Cantürk

21.11.2020

1. INTRODUCTION

Linear regression is easy to understand, and to model. It might be implemented as a preliminary model to interpret linear relation between dependent variables (i.e., features) and an independent variable (i.e., target, respondent). Our car price dataset more than one feature, so, multiple linear regression is used in this report.

Multiple linear regression models can be depicted by the following equation.

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \varepsilon_i \qquad (1)$$

y is a dependent variable which is subject to predict. In linear regression, y is always a numerical variable and cannot be categorical. x_i's are independent variables which are taken into the consideration to predict y, where k is the number of independent variables. β_i's are coefficients which determine how much a unit change in x_i changes y while other variables remain constant. β_0 is the intercept value which is the model output if there is no effect of any independent variables on the dependent variable. ε_i's are the random errors, i.e., residuals.

There are some assumptions for linear regression:

Linearity: The true relationship between the dependent variable and independent variables is linear.

Multivariate Normality: Residuals are normally distributed around a zero mean with a constant variance. Also, they are independent, i.e., not dependent on any independent variable.

Homoscedasticity: Regression line has the same variance over all values of an independent variable.

Multicollinearity: Independent variables are not highly correlated with each other.

Unless data is consistent with the assumptions, performance of linear regression models deteriorates.

2. EXPLORATORY DATA ANALYSIS

Table 1. Sample from Car Pricing Data Set

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_power	seats
6463	Datsun GO A	2015	210000	80000	Petrol	Individual	Manual	First Owner	20.63 kmpl	1198 CC	67.06 bhp	5.0
1405	Tata Indica Vista Aqua 1.3 Quadrajet	2011	125000	85000	Diesel	Individual	Manual	Second Owner	18.0 kmpl	1248 CC	75 bhp	5.0
1079	Jaguar XF 2.0 Diesel Portfolio	2017	3200000	45000	Diesel	Dealer	Automatic	First Owner	19.33 kmpl	1999 CC	177 bhp	5.0
3985	Mahindra XUV500 W8 4WD	2012	595000	129000	Diesel	Individual	Manual	Third Owner	15.1 kmpl	2179 CC	140 bhp	7.0
5243	Maruti Omni E MPI STD BS IV	2014	155000	80000	Petrol	Individual	Manual	First Owner	16.8 kmpl	796 CC	34.2 bhp	8.0

Table 2. Number of non-missing values in each feature and the target variable

```
name           8128 non-null
year           8128 non-null
selling_price  8128 non-null
km_driven      8128 non-null
fuel           8128 non-null
seller_type    8128 non-null
transmission   8128 non-null
owner          8128 non-null
mileage        7907 non-null
engine         7907 non-null
max_power      7913 non-null
seats          7907 non-null
```

There are 8128 observations in the dataset and 6717 observations are unique. 11 features are listed in Table 1 and 2. Some observations have missing values for some features. Total number of these observations is 221, which is about 2.72% of the data. After dropping them, 7907 left.

km_driven, *age*, *mileage*, *engine*, *max_power*, and *seats* are the numerical features where *age* the converted feature, equivalent to 2020 - *year*. *name*, *fuel*, *seller_type*, *transmission*, and *owner* are the categorical features.

Table 3. Statistics of the Numerical Features and the Target Variable

	km_driven	age	mileage	engine	max_power	seats	selling_price
count	7906.00	7906.00	7906.00	7906.00	7906.00	7906.00	7906.00
mean	69188.66	6.02	19.42	1458.71	91.59	5.42	649813.72
std	56792.30	3.86	4.04	503.89	35.75	0.96	813582.75
min	1.00	0.00	0.00	624.00	32.80	2.00	29999.00
25%	35000.00	3.00	16.78	1197.00	68.05	5.00	270000.00
50%	60000.00	5.00	19.30	1248.00	82.00	5.00	450000.00
75%	95425.00	8.00	22.32	1582.00	102.00	5.00	690000.00
max	2360457.00	26.00	42.00	3604.00	400.00	14.00	10000000.00

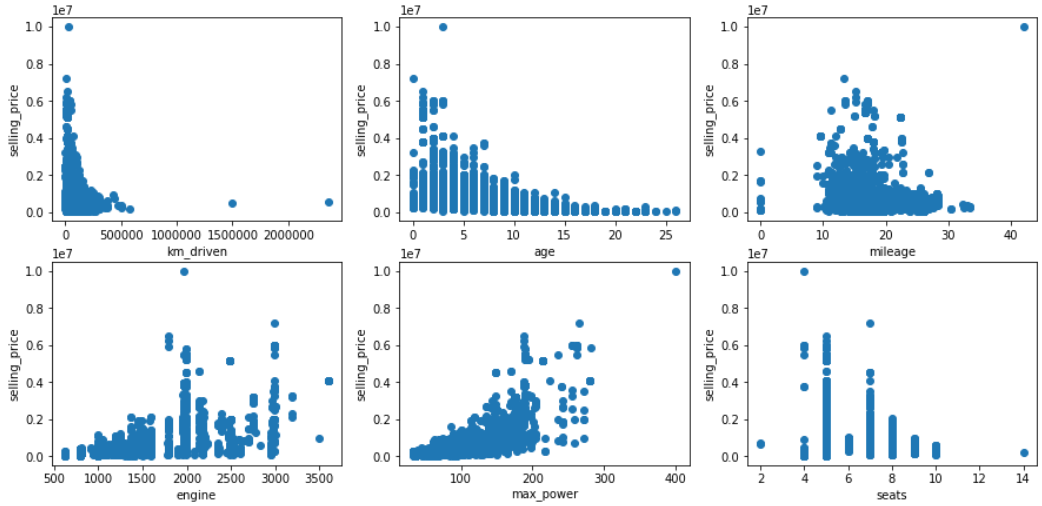

Figure 1. Distributions of Numerical Features Over Selling Price

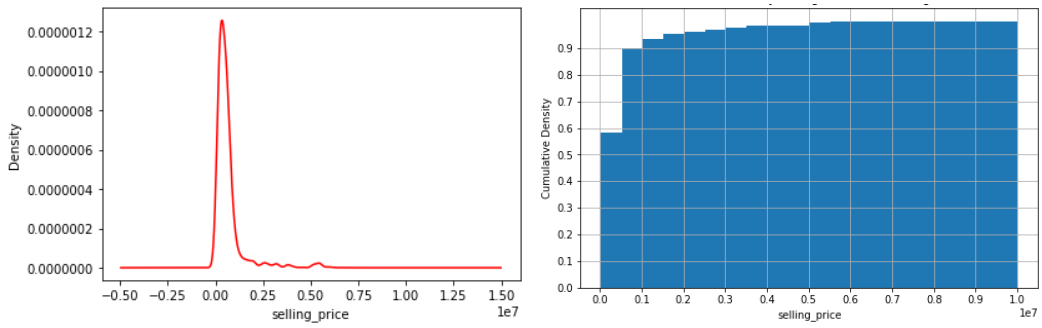

Figure 2. (i) Density Distribution and (ii) Cumulative Density Histogram of Car Selling Price

90% of the observations is accumulated up to 0.1 * 10^7 selling price according to Figure 2(i). Besides, they are not normally distributed based on the target. So, these show that a linear regression model to be fitted on this data is expected to yield high errors. Besides, *seats*, *mileage*, and *km_driven*, are not linearly correlated with the target.

Outliers should be removed, or Standard Normalization should be applied to data but there is only one outlier whose *selling_price* value is 10^7, which is ineffective alone to change regression fit against 7907 observations.

Table 4. Values Count of Categorical Features

name	1982
fuel	4
seller_type	3
transmission	2
owner	5

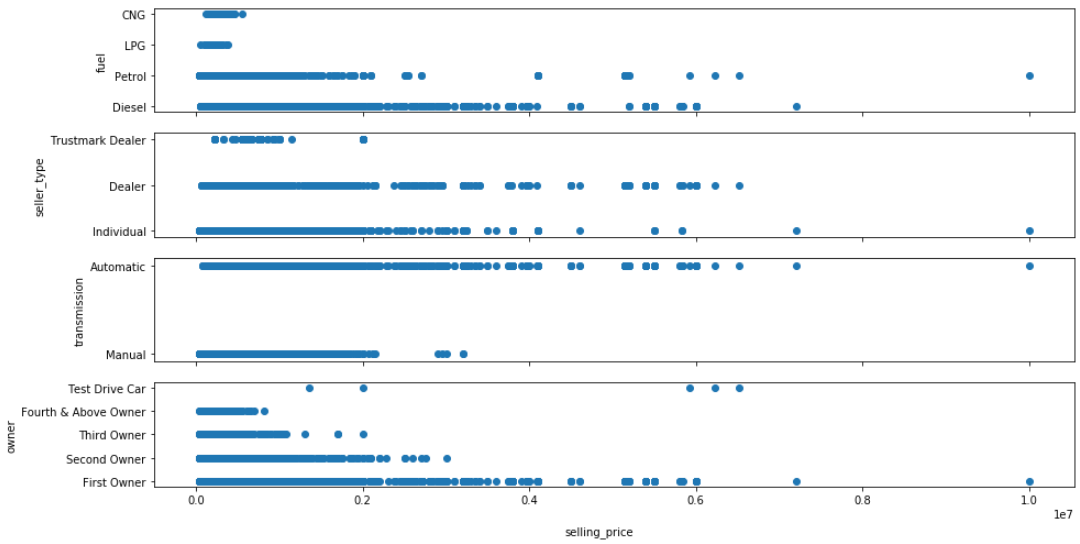

Figure 3. Price Distributions of Each Level of Each Categorical Feature Except "name"

To use categorical features in a linear regression model, they are needed to be convert to a numerical form. So, I encode each categorical feature as a group of bits representation with One-Hot Encoding. For example, fuel feature is encoded as [0,1,0,0] if its value is *LPG*. So, it is converted to 4 features to be used in the model.

name has 1982 different values. It means 1982 new features (after One-Hot Encoding) if *name* is chosen to use in car price prediction. However, using *name* directly in the modelling is problematic with 2 reasons. Firstly, this model, using *name* as a feature, is harder to understand individual effect of each feature on car price among thousands of features. Secondly, more importantly, this model will become quickly impractical as time passes since each new car represents a new feature with its unique name. So, for each a new car name, we will need to update the model by adding a new variable of new name. Also, a general model is aimed to predict the price of any car, not using a separate feature for each specific car name.

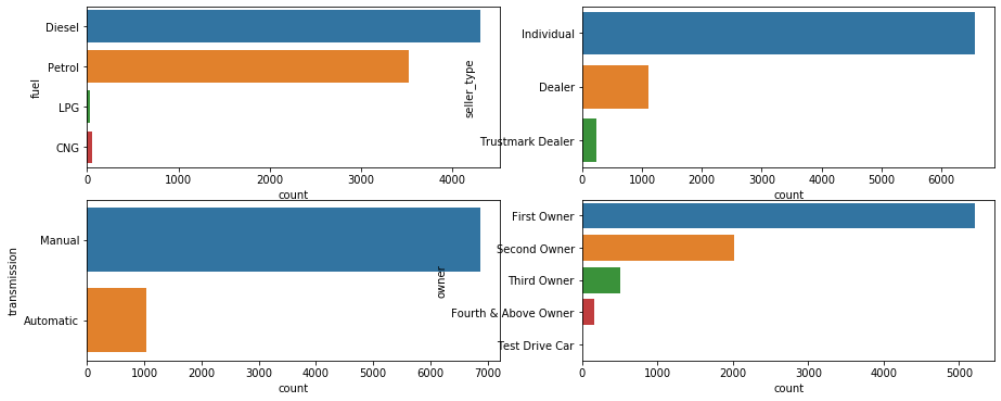

Figure 4. Value Counts of the Categorical Features Except "name"

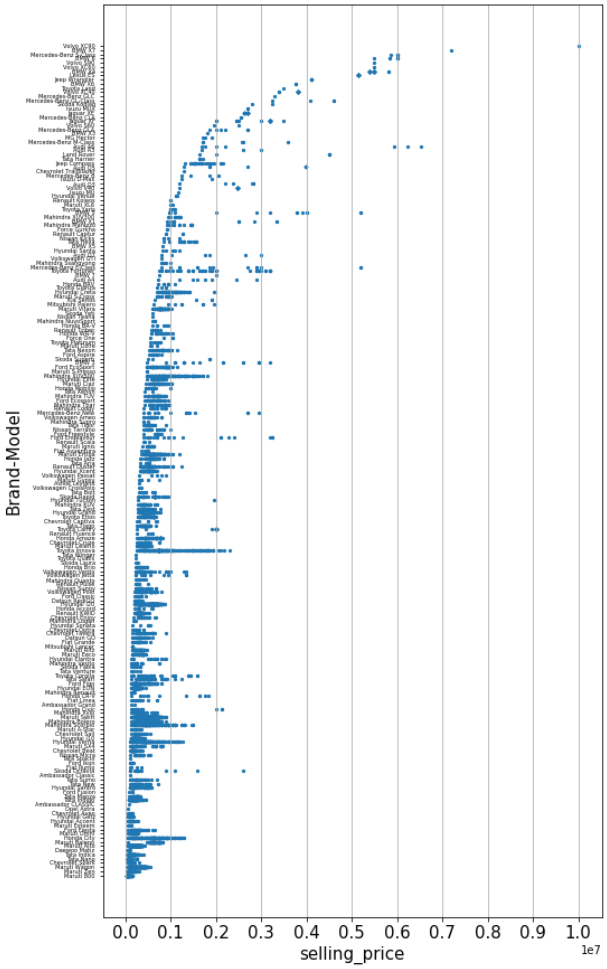

Figure 5. Price Distribution of Car Brand-Model Levels

Due to high cardinality of *name*, its value levels should be reduced if it is used in regression of car price. So, a new feature named *brand_model* is created from *name* by getting first two words in it. *brand_model* has 200 unique values. When price distribution of brand-model levels is inspected over Figure 5, we see that the brand-model levels whose prices are less than $0.2*10^7$ are distributed around an almost linear curve although some brand-model levels have a high variance in their prices. So, *brand_model* can be clustered in 2 with a new feature named *luxury*. Accordingly, the list of brandmodel types whose all observations' prices are larger than $0.2*10^7$ is extracted. If *brand_model* of an observation is in that list, its *luxury* value takes 1, otherwise 0.

```
Lux car brand-models:
['Mercedes-Benz B' 'Toyota Innova' 'Ford Endeavour' 'Jeep Compass'
 'Mercedes-Benz GLA' 'Honda Civic' 'BMW 3' 'Audi A6' 'Audi Q3' 'Jaguar XF'
 'Audi A4' 'Volvo V40' 'BMW X1' 'Volvo S60' 'Mercedes-Benz CLA'
 'Mercedes-Benz M-Class' 'Skoda Octavia' 'Audi A3' 'Jaguar XE' 'Audi Q7'
 'Mercedes-Benz New' 'Audi Q5' 'Mercedes-Benz E-Class' 'Isuzu MUX'
 'Toyota Fortuner' 'Skoda Kodiaq' 'BMW 5' 'Mercedes-Benz GL-Class'
 'Mercedes-Benz GLC' 'Volvo XC40' 'Toyota Land' 'BMW X6' 'Jeep Wrangler'
 'Land Rover' 'Lexus ES' 'BMW X4' 'Volvo XC60' 'Volvo S90' 'BMW 6'
 'Mercedes-Benz S-Class' 'BMW X7' 'Volvo XC90']
```

I note that this clustering might be done in more than 2 clusters and different split values (e.g. $0.1*10^7$, $0.3*10^7$ etc.) should be experimented, but optimizing the clustering of dataset is beyond the scope of report. So, this optimizing phase is skipped, and the split value was determined by just observation.

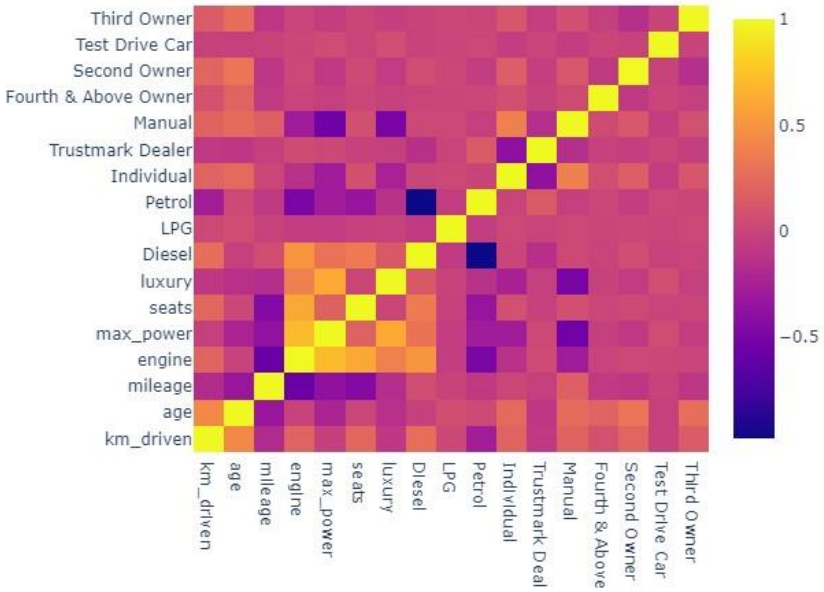

Figure 6. Heatmap of Correlation Matrix

max_power	0.749674
luxury	0.682865
Manual	0.590269
engine	0.455682
age	0.412302
Individual	0.386151
km_driven	0.222158
Diesel	0.204831
Petrol	0.195074
Second Owner	0.178786
mileage	0.126280
Test Drive Car	0.116081
Third Owner	0.115113
Fourth & Above Owner	0.073601
seats	0.041617
LPG	0.035978
Trustmark Dealer	0.032779

Table 5. Correlations with Car Price

engine, *Diesel*, *luxury*, *seats*, and *max_power* are the variables which highly correlated with each other according to the correlation matrix in Figure 6.

In Table 5, the first 6 variables have a correlation with the target at rates larger than 0.38. Correlation rates of the rest is less than 0.23. So, there is a clear separation of highly correlated and low correlated variables with the target.

3. MODELLING AND RESULTS

Considering the exploratory data analysis, 5 linear regression models are built. Before fitting the regression lines, all numerical inputs are normalized with min-max scaling. For every feature, the minimum value of that feature is transformed into 0, the maximum value is transformed into 1, and every other value gets transformed between 0 and 1 with formula: (value – min_value) / (max_value – min_value). Besides, all categorical features are one-hot encoded, as said in the previous section, but the first bit in the binary representation is dropped to avoid multicollinearity. For example, *transmission* is represented with only one bit, named *Manual*. If *Manual*=0 for an observation, then it means *transmission* value of that observation is *Automatic*. If *transmission* is represented with two bits, i.e. two separate features as *Manual* and *Automatic*, each bit correlates with each other since 0-1 is the reversed of 1-0 representation.

The transformed data set is splitted into training and test datasets where proportion of test data to all is 33%. Each model is trained with the same training dataset and tested with the same test dataset.

Table 6. Variables, Target, and Test Scores of the Models

	Model-1	**Model-2**	**Model-3**	**Model-4**	**Model-5**
x1	*max_power*	*max_power*	*max_power*	*max_power*	*max_power*
x2	*Manual*	*Manual*	*Manual*	*Manual*	*Manual*
x3	*engine*	*engine*	*engine*	*engine*	*Petrol*
x4	*age*	*age*	*age*	*age*	*age*
x5	*Individual*	*Individual*	*Individual*	*Individual*	*Individual*
x6		*luxury*		*luxury*	*luxury*
y	*selling_price*	*selling_price*	ln(*selling_price*)	ln(*selling_price*)	ln(*selling_price*)
R-Square	0.678	0.743	0.823	0.875	0.908
RMSE	464,055.111	414,022.749	343,496.846	289,219.81	248,503.33
MAE	273,839.53	238,327.976	155,315.837	139,399.806	129,646.441

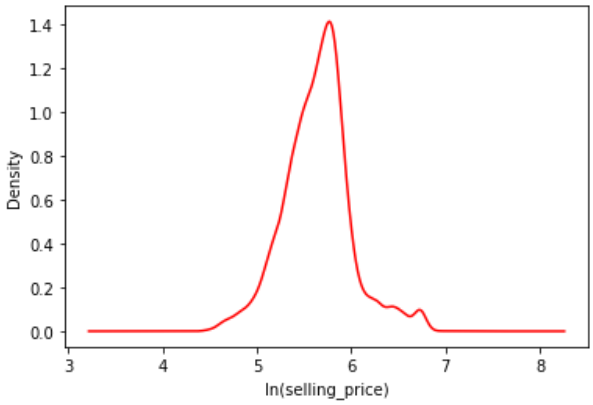

Figure 7. Density Distribution of ln(*selling_price*)

Model-1 is constructed with the first 5 variables in Table 5 except *luxury*, and contribution of *luxury* to prediction success is observed with Model-2.

Density distribution of prices was observed in Figure 2 (i), which is highly skewed. Logarithmic transformation is applied on the target so that a more normalized distribution is obtained as in Figure 7 to be more consistent with the linear regression assumptions. Performance difference of Model-1 and Model-3 shows the contribution to the prediction success made by this transformation.

Both transformation of *name* into *luxury* and logarithmic transformation of *selling_price* is applied with Model-4, which achieves better performance than the previous ones.

To build the last model, all variables are evaluated: Highly correlated variables with target are selected and the variables causing high multicollinearity are removed. The only difference of Model-5, compared to Model-4, is that *Petrol* is selected rather than *engine* since *engine* causes high multicollinearity as seen in Figure 6.

4. IMPLEMENTATION

All data visualization, pre-processing, and modelling is done with Python libraries such as Pandas, Numpy, Seaborn, Plotly, and Scikit-learn. The code is at Appendix as a Jupyter Notebook.

5. CONCLUSION

For a transformed observation (i.e. encoded categorical variables, and min-max scaled numerical variables), coefficients for the linear regression equation formula of Model-5 is demonstrated in Table 7. *Diesel, engine, seats, mileage, km_driven, LPG, Third Owner, Second Owner, Test Drive Car, Trustmark Dealer,* and *Fourth & Above Owner* are the discarded variables in Model-5, i.e. their coefficients are 0's.

Table 7. Coefficients of Model-5

	Coefficient
intercept	13.381813
age	-3.134177
max_power	3.619460
Petrol	-0.223856
Individual	-0.103842
Manual	-0.108115
luxury	0.451977

Linear equation of Model-5 as follows.

$$\ln(\text{price}) = 13.381813 - 3.134177 * \text{age} + 3.619460 * max\_power - 0.223856 * \text{Petrol} - 0.103842 * \text{Individual} - 0.108115 * \text{Manual} + 0.451977 * \text{luxury} \quad (2)$$

where 13.381813 is the intercept value, *age* and *max_power* are the scaled variables and *Petrol*, *Individual*, *Manual*, and *luxury* are binary variables indicating categorical features of observation accordingly.

To calculate a predicted car price with some error, the linear predictor becomes the power of e:

$$price = e^{13.381813 - 3.134177*age + 3.619460*maxpower - 0.223856*Petrol - 0.103842*Individual - 0.108115*Manual + 0.451977*luxury} \quad (3)$$

which is equal to

$$price = e^{13.381813} * e^{-3.134177*\text{age}} * e^{3.619460*\text{maxpower}} * e^{-0.223856*\text{Petrol}} * e^{-0.103842*\text{Individual}} * e^{-0.108115*\text{Manual}} * e^{0.451977*\text{luxury}} \quad (4)$$

(4) provides the following information about car price prediction.

One-unit change in (scaled) *age* changes car price by a factor of 0.043536, which is equal to $e^{-3.134177}$, while other variables remain same (w.o.v.r.s).

One-unit change in (scaled) *max_power* changes car price by a factor of 37.317423 w.o.v.r.s.

If *fuel* is *Petrol*, it changes car price by a factor of 0.799431 w.o.v.r.s.

If *seller_type* is *Individual*, it changes car price by a factor of 0.901368 w.o.v.r.s.

If *transmission* is *Manual*, it changes car price by a factor of 0.897524 w.o.v.r.s.

If *brand_model* is in *luxury*, it changes car price by a factor of 1.571415 w.o.v.r.s.

If all variables are 0, then the predicted price is the intercept value 648,108.078, which is equal to $e^{13.381813}$.

Example

Table 8. An Observation in the Test Dataset

	name	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_power	seats	age	brand_model
0	Maruti Swift Dzire VDI	450000	145500	Diesel	Individual	Manual	First Owner	23.40	1248.0	74.00	5.0	6	Maruti Swift

For the observation represented in Table 8, the inputs are *age*=6, *max_power*=74, *Petrol*=0, *Individual*=1, *Manual*=1, and *luxury*=0 since its brand-model is not in the lux car brand-model list. After *age* and *max_power* values are multiplied with some scaling factor, all inputs are plugged in (3). This price prediction result gives 527,548.22. If brand-model of this observation was in the lux car list, the predicted price would be 828,997.36 which is equal to 527,548.22 * 1.571415 (1.571415 is the luxury factor as stated above).