Linear Regression Analysis on Car Selling Price

Prepared by: Monty Xu, Vishwas Prabhu, Hanna Lee

Statement of the research:

Our dataset contains various information about cars. This analysis aims to find the most suitable predictors, both categorical and numerical, in predicting an appropriate selling price of the car in the regression model. We will focus on the idea of linear regression and how to perform linear regression modelling in Python environment.

In the beginning, we clean the dataset and chose the initial columns we want to perform the regression process, followed by potential data structural problem checking. Next, we checked the potential model assumption violations. In the end, we performed the best subsets method to choose the best model.

Summary of methods being used in the analysis:

- To check for multicollinearity → VIF score method
- To check for influential points → Externally Studentized Residuals Plot and Cook's Distance Calculation
- To check for heteroscedasticity → Plot of Residuals and Breusch-Pagan Test
- To check for non-normality residuals → Normal-Probability Plot (Q-Q plot) and Jarque-Bera Test
- Model-selection → Best subset atomic procedure using AIC/BIC and Adjusted R^2

Description of Dataset:

This dataset contains information about used cars, and it contains 13 columns which include “name”, “year”, “km_driven” etc., and 8128 rows in total.

The picture below is the information table of the dataset.

```
carsdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           8128 non-null   object
 1   year           8128 non-null   int64
 2   selling_price  8128 non-null   int64
 3   km_driven      8128 non-null   int64
 4   fuel           8128 non-null   object
 5   seller_type    8128 non-null   object
 6   transmission   8128 non-null   object
 7   owner          8128 non-null   object
 8   mileage        7907 non-null   object
 9   engine         7907 non-null   object
 10  max_power      7913 non-null   object
 11  torque         7906 non-null   object
 12  seats          7907 non-null   float64
dtypes: float64(1), int64(3), object(9)
memory usage: 825.6+ KB
```

First, let us take a look at each column.

1. Target variable: Selling price.

```
carsdata['selling_price']
```

```
0       450000
1       370000
2       158000
3       225000
4       130000
         ...
8123    320000
8124    135000
8125    382000
8126    290000
8127    290000
Name: selling_price, Length: 8128, dtype: int64
```

```
carsdata['selling_price'] = carsdata['selling_price'] / 1000
carsdata['selling_price']
```

```
0       450.0
1       370.0
2       158.0
3       225.0
4       130.0
        ...
8123    320.0
8124    135.0
8125    382.0
8126    290.0
8127    290.0
Name: selling_price, Length: 8128, dtype: float64
```

The selling price is numeric data and we converted the price to price/per thousand.

2. Name of cars

```
carsdata['name']
```

```
0             Maruti Swift Dzire VDI
1         Skoda Rapid 1.5 TDI Ambition
2           Honda City 2017-2020 EXi
3          Hyundai i20 Sportz Diesel
4             Maruti Swift VXI BSIII
                        ...
8123               Hyundai i20 Magna
8124           Hyundai Verna CRDi SX
8125          Maruti Swift Dzire ZDi
8126                 Tata Indigo CR4
8127                 Tata Indigo CR4
Name: name, Length: 8128, dtype: object
```

We can see that the datatype of the “name” column is ‘string’, and it has too many categories. It seems “name” doesn’t have a significant impact on selling price so we dropped this column.

3. Production year of a car (Numeric variable)

The “year” column shows the production year of a car and we converted the “year” column to the “age” which compared to the year 2020.

4. Kilometers driven (Numeric variable)

The "km_driven" column shows how many kilometers the car has travelled when it is being sold.

5. Fuel type (Categorical variable)

```
carsdata['fuel'].value_counts()
```

```
Diesel    4402
Petrol    3631
CNG         57
LPG         38
Name: fuel, dtype: int64
```

The fuel type is a categorical variable, and we kept the two main categories to perform analysis.

6. Seller type (Dropped)

Since most of the cars in our dataset are from individual sellers, we just dropped this column.

7. Transmission (Categorical variable)

The “transmission” column tells if the gear transmission of the car is automatic or manual. It’s a categorical variable and has two categories.

8. Owner (Categorical variable)

```
carsdata['owner'].value_counts()
```

```
First Owner             5238
Second Owner            2073
Third Owner              547
Fourth & Above Owner     170
Test Drive Car             5
Name: owner, dtype: int64
```

We combined the third owner and fourth and above owner as “Third and above” categories and dropped “Test Drive Car”. After transformation, we only have 3 categories.

9. Mileage (Numeric variable)

The datatype of “mileage” is string, which consists of a number and its units. We kept the numeric part.

10. Engine (Numeric variable)

Same as the mileage, the datatype of “engine” is string, and we kept the numeric part.

11. Max power(Dropped)

The max power is similar to the “engine”, so we dropped this column.

12. Torque(Dropped)

Torque is a physical concept, and we just dropped it.

13. Seats (Dropped)

The “seats” has too many categories and most of the cats have 5 seats, so we dropped this column.

After selecting all the columns, we also dropped the data records which include null values.

In conclusion, we have a target variable: Selling price, four numerical variables, including “age”, “km_driven”, “mileage” and “engine”, and three categorical variables, including “fuel”, “transmission” and “owner”.

Potential problems:
We run a few tests to try to identify modeling problems including data structural problems and model assumption violation as listed below:

Data Structural Problems:

- Multicollinearity
- Influential Points

Model Assumption Violations:

- Heteroscedasticity
- Non-normality residuals

Test for multicollinearity:
We run Variance Inflation Factor (VIF) test to check how much the variance of our predictors is inflated in the coefficient estimates.

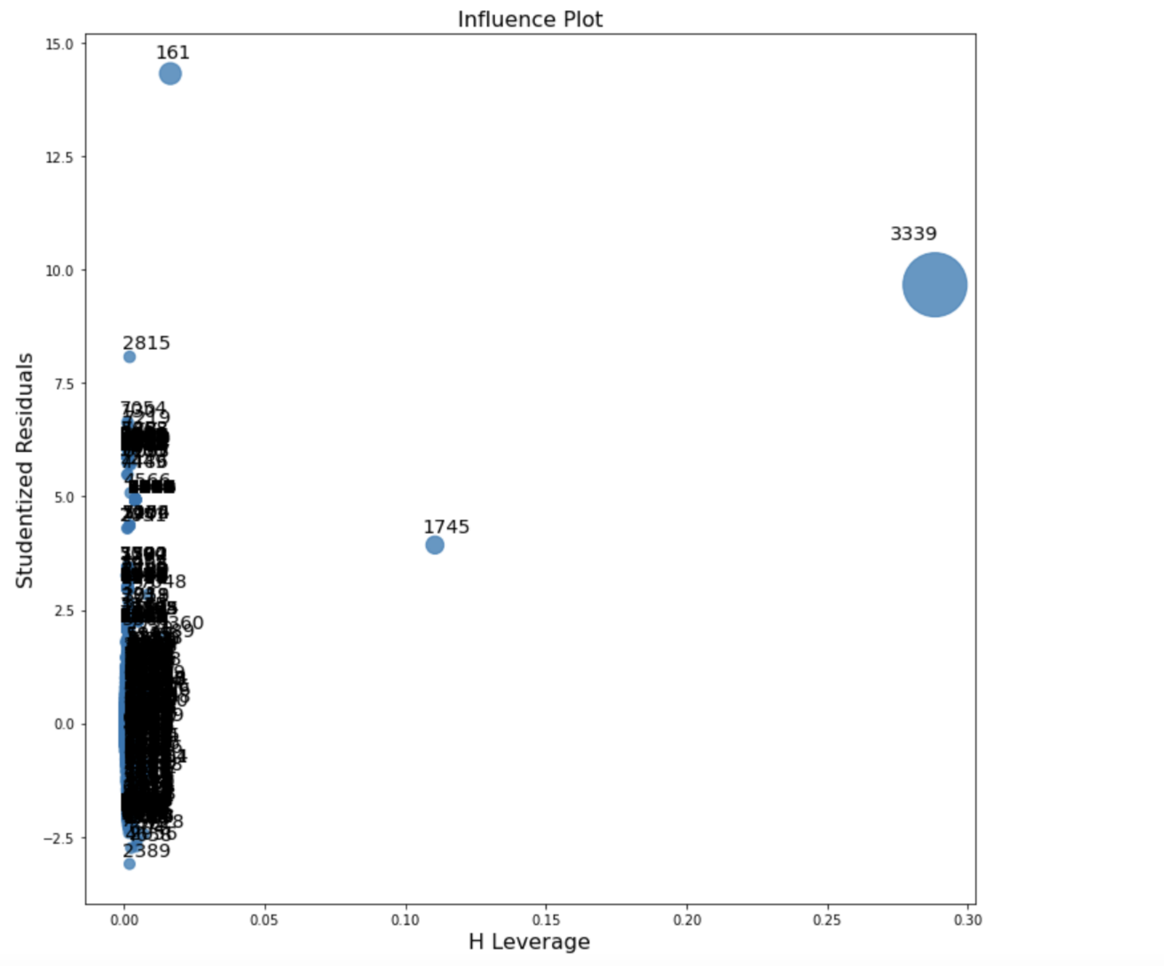
<u>VIF score for data:</u>

	VIF Factor	features
0	204.827539	Intercept
1	2.041134	C(fuel)[T.Petrol]
2	1.248729	C(transmission)[T.Manual]
3	1.260143	C(owner)[T.Second Owner]
4	1.271157	C(owner)[T.Third & Above Owner]
5	1.888147	age
6	1.419133	km_driven
7	2.600718	mileage
8	3.293355	engine

Result:
Based on our test on the dataset, there are no VIF scores that are higher than 10. There is no serious multicollinearity problem in this data.

Test for influential points:
We plot the 'Externally Studentized Residuals Influence Plot' to check if there are any influential points visually that could potentially influence our model and we also calculated the Cook's distance to detect influential points.

<u>Externally Studentized Residuals Influence Plot and calculation from notebook:</u>



<u>Detection through Cook's distance (Di) calculation:</u>
Cook's distance (Di) for ith observation is calculated as follow:

$$D_i = \frac{\sum(\hat{Y}_j - \hat{Y}_{j(i)})^2}{p \cdot MSE} = \frac{e_i^2}{p \cdot MSE} \cdot \frac{h_{ii}}{(1-h_{ii})^2}$$

Di measures directly how much all the fitted values change after deleting that particular i^{th} observation. Rule of thumb that are used to decide whether the calculated Di is flagged as an influential point is when Di > 4/n.

Result:
From the 'Externally Studentized Residuals Influence Plot', we can see that there are some outliers in the plot but from the plot, we are unsure of exactly how many points are influential points. Through the externally studentized residuals calculation in the notebook, using externally studentized residuals threshold value, 318 influential points were found.

Upon calculating the Cook's distance for each observation (based on the calculation on the notebook), we can see that there are 533 numbers of influential points.

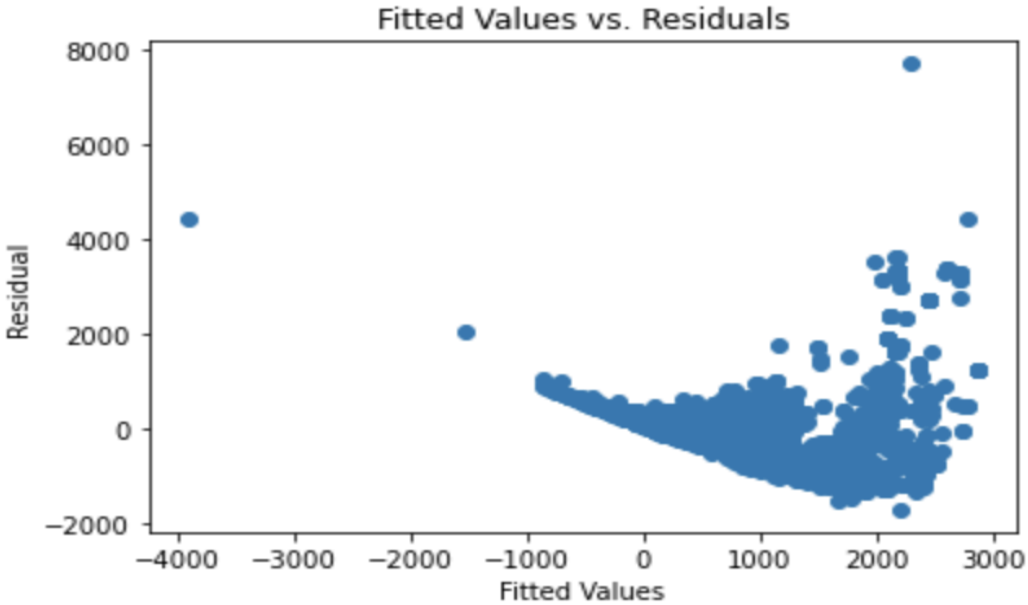
Our approach to confirm these influential points is to identify those influential points that are flagged in both externally studentized residuals and in the Cook's distance calculation, which is a total of 316 points that are found using both methods.

Solution:
Besides reporting the observations, we also analyze the model result with and without the influential points.

Also, we note that influential points would highly influence the Breusch-Pagan test result which we will carry out next to test heteroskedasticity. Hence, we would perform the Breusch-Pagan test with and without the influential points separately.

Test for heteroscedasticity:
We plotted the 'Residual versus Fitted Value Plot' to check if there are obvious changes of bandwidth in the plot and carried out the Breusch-Pagan Test to check the change of the variance of error of the data as the predictors change.

Residual versus Fitted Value Plot before dropping influential points:



Breusch-Pagan Test:
Null hypothesis: No relation between error term and predictors
Alternate hypothesis: Significant relationship between error term and predictors

Breusch-Pagan Results from notebook:

```
{'LM Statistic': 1671.827653088665, 'LM-Test p-value': 0.0}
```

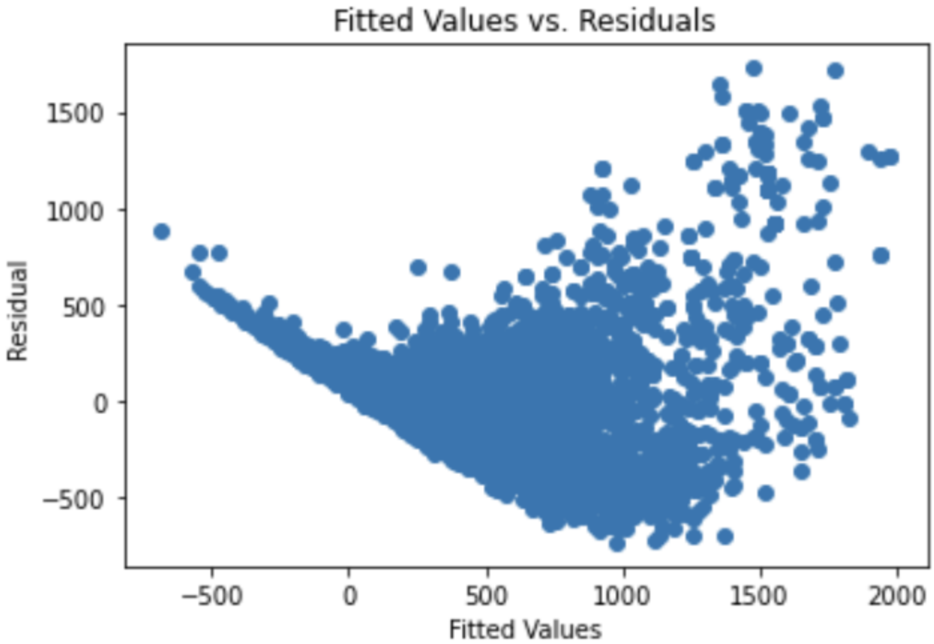
Since p-value < alpha =0.05, we reject the null hypothesis and conclude that there is a significant heteroscedasticity problem.

Result:
Visually on the 'Residual versus Fitted Value Plot', there is an obvious change of bandwidth of the residuals in the plot, which suggests the possibility of heteroscedasticity.

Solution:

1. Dropping influential points

Residual versus Fitted Value Plot after dropping influential points:



Breusch-Pagan Test:

Null hypothesis: No relation between error term and predictors

Alternate hypothesis: Significant relationship between error term and predictors

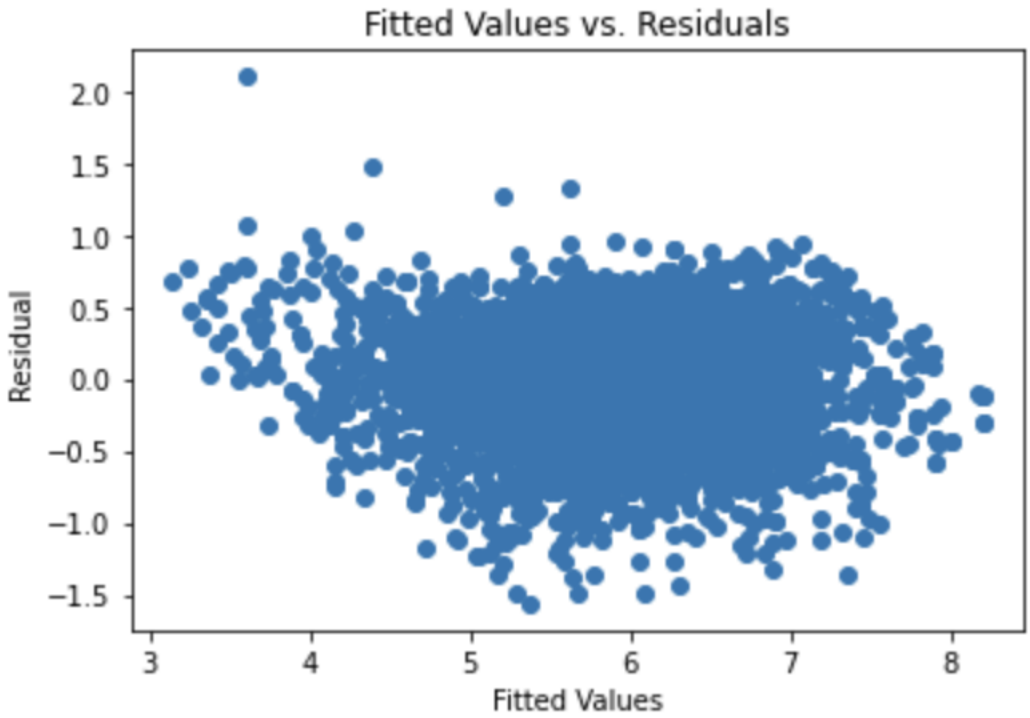
Breusch-Pagan Results from notebook:

```
{'LM Statistic': 2382.553931940368, 'LM-Test p-value': 0.0}
```

Since p-value < alpha =0.05 we still reject the null hypothesis and conclude that there is a significant heteroscedasticity problem, which means dropping influential points doesn't improve the heteroscedasticity problem.

2. Transforming Y to log(Y)

Residual versus Fitted Value Plot after transforming Y to log(Y):



Breusch-Pagan Test:
Null hypothesis: No relation between error term and predictors
Alternate hypothesis: Significant relationship between error term and predictors

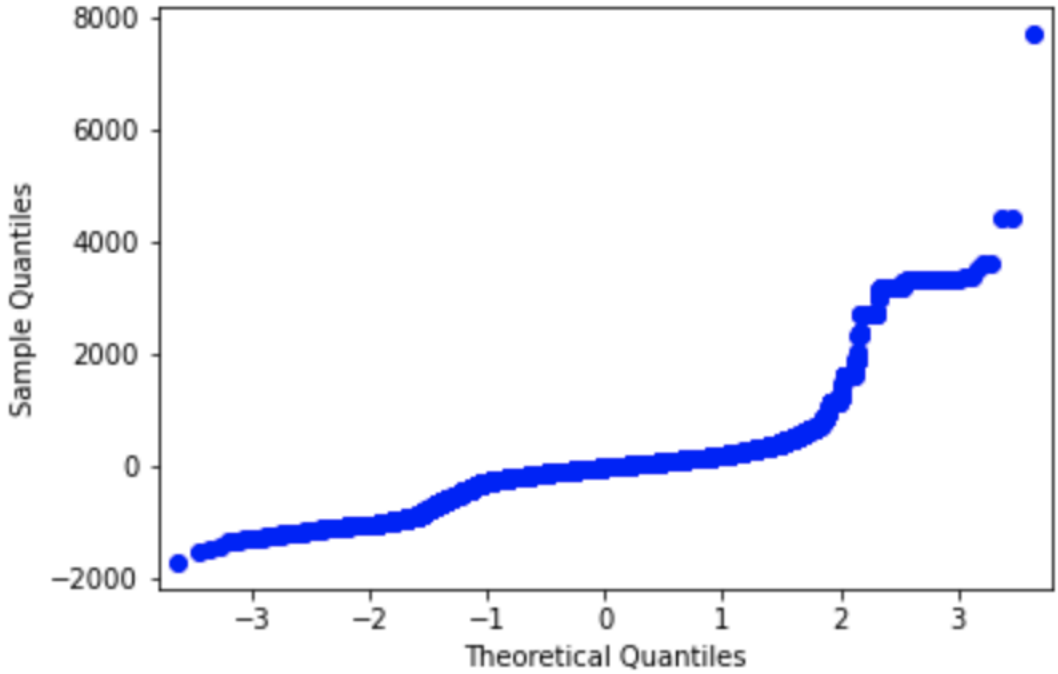
Breusch-Pagan Results from notebook:

```
{'LM Statistic': 410.4218297297747, 'LM-Test p-value': 1.1036884101508006e-83}
```

Since p-value < alpha =0.05 we still reject the null hypothesis and conclude that there is a significant heteroscedasticity problem, which means converting Y to log(Y) doesn't improve the heteroscedasticity problem either.

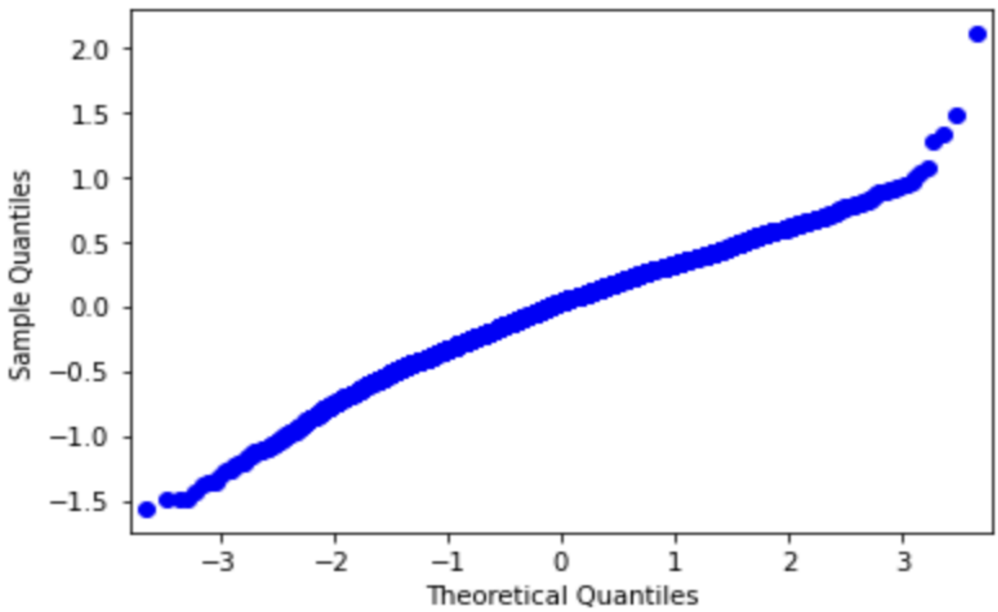
To conclude, after dropping influential points and converting Y to log(Y), the p-values of the BP test are still close to 0, which means both ways of dropping influential points and converting Y to log(Y) don't improve the heteroscedasticity problem either. Hence, we still have the heteroscedasticity problem in our model.

Test for non-normality residuals:
We plotted the 'Normal-Probability Plot' (Q-Q plot) to check if the dataset is normally distributed and we further check the Jarque-Bera test results for normality as well.



Result:
The resulting plot is not approximate linear on the diagonal of the plot suggests non-normality, and the Jarque-Bera test results also suggest non-normality.

Solution:
We performed a natural-log transformation on y, and then the diagonal of the Q-Q plot improved significantly visually, as shown below. However, the Jarque-Bera test results is still close to zero showing non-normality.



Summary of potential problems after exploring the data:

Potential Problems	**Test to Detect**	**Result of Test: Problem Present in Data (Yes/No)**	**Solution**
Multicollinearity	VIF scores	No	-
Influential Points	Externally Studentized Residuals Plot and Cook's Distance Calculation	Yes	Report the observed influential points and perform model analysis with and without the influential points identified.
Heteroscedasticity	Plot of Residuals and Breusch-Pagan Test	Yes	Log-transformation on y
Non-normality residuals	Normal-Probability Plot (Q-Q plot)	Yes	Log-transformation on y

Model selection:
We are using the best subset atomic search procedure for every possible combination of models. In our case we will get 127 possible combinations since we have 7 predictors in total for our analysis. We will choose the best model based on several criteria like AIC, BIC and adjusted R2 separately.
We have included the complete list of models in the appendix.

Using AIC/BIC criteria we get the following result for best subset of predictors:

	model	predictors	adj_rsq	AIC	BIC
0	selling_price~age	1	0.494399	13881.917158	13895.844502
9	selling_price~age+ engine	2	0.748729	8419.261088	8440.152105
38	selling_price~age+ engine+ C(transmission)	3	0.789108	7051.345127	7079.199816
79	selling_price~age+ engine+ C(fuel)+ C(transmission)	4	0.797310	6742.384080	6777.202442
112	selling_price~age+ engine+ C(fuel)+ C(transmission)+ C(owner)	5	0.799779	6648.620175	6697.365880
123	selling_price~age+ km_driven+ engine+ C(fuel)+ C(transmission)+ C(owner)	6	0.801361	6587.637096	6643.346474
126	selling_price~age+ km_driven+ mileage+ engine+ C(fuel)+ C(transmission)+ C(owner)	7	0.801953	6565.312935	6627.985986

Using adjusted R2 criteria we get the exact same table as above after selecting for subset of predictors.

Result:
As seen from the table above we get the full model as the model of choice irrespective of the selection criteria.

Final Regression:
Based on the analysis we performed and model selection process, we chose the below full model as our final choice.

Log (Y) ~ age + km_driven + mileage + engine + C(fuel) + C(transmission) + C(owner)

The summary table as shown below:

OLS Regression Results

Dep. Variable:	np.log(selling_price)	**R-squared:**	0.802
Model:	OLS	**Adj. R-squared:**	0.802
Method:	Least Squares	**F-statistic:**	3956.
Date:	Fri, 15 Oct 2021	**Prob (F-statistic):**	0.00
Time:	17:09:22	**Log-Likelihood:**	-3273.7
No. Observations:	7814	**AIC:**	6565.
Df Residuals:	7805	**BIC:**	6628.
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	6.3020	0.060	105.745	0.000	6.185	6.419
C(fuel)[T.Petrol]	-0.1697	0.012	-14.190	0.000	-0.193	-0.146
C(transmission)[T.Manual]	-0.5628	0.014	-41.036	0.000	-0.590	-0.536
C(owner)[T.Second Owner]	-0.0874	0.011	-8.142	0.000	-0.108	-0.066
C(owner)[T.Third & Above Owner]	-0.1084	0.017	-6.427	0.000	-0.142	-0.075
age	-0.1229	0.001	-82.967	0.000	-0.126	-0.120
km_driven	-6.997e-07	8.72e-08	-8.022	0.000	-8.71e-07	-5.29e-07
mileage	0.0083	0.002	4.933	0.000	0.005	0.012
engine	0.0007	1.5e-05	46.426	0.000	0.001	0.001

Omnibus:	254.746	**Durbin-Watson:**	1.809
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	536.723
Skew:	-0.211	**Prob(JB):**	2.83e-117
Kurtosis:	4.213	**Cond. No.**	1.30e+06

Summary of our findings:
We have a combination of categorical predictors as well as numerical predictors in our dataset. We have analyzed the effect of each of the predictors and their levels (in the case of categorical predictors).

Following are the observations from the analysis:

- Fuel type petrol has negative influence on selling price in comparison to diesel option.
- Transmission type manual has a negative effect on selling price in comparison to automatic.
- Second owner and above have negative effect on selling price in comparison to first owner (first owner is the reference level).
- Age of the vehicle and distance driven has a negative effect on selling price of the vehicle.
- Mileage and the engine size positively affect the selling price of the vehicle on average.

Discussion on lingering problem of the data structure:

We still have heteroscedasticity problem in our model that hasn't been solved. This problem will cause the beta vector (coefficient vector) to not be the "best" anymore, and MSE will not be a reliable estimate of variance of error terms.

We also have non-normality problem in our model. As seen from the QQ plot of log(Y) model, the distribution of error terms doesn't extremely departures from normality. And since we have more than 7000 data records, we conclude that non-normality problem may not cause serious problems.

References:

Applied Linear Statistical Models- 5th Edition by Kutner, Nachtsheim, and Neter.

Dataset: https://www.kaggle.com/nehalbirla/vehicle-dataset-from-cardekho?select=Car+details+v3.csv

Glossary:
Table: List of all possible models for predicting selling price of car

ID	Model	Predictors	adj_rsq	AIC	BIC
0	selling_price~age	1	0.49	13882.0	13896.0
1	selling_price~km_driven	1	0.06	18723.0	18737.0
2	selling_price~mileage	1	0.0	19204.0	19218.0
3	selling_price~engine	1	0.27	16793.0	16807.0
4	selling_price~C(fuel)	1	0.1	18381.0	18395.0
5	selling_price~C(transmission)	1	0.25	16953.0	16967.0
6	selling_price~C(owner)	1	0.16	17861.0	17882.0
7	selling_price~age+ km_driven	2	0.5	13828.0	13849.0
8	selling_price~age+ mileage	2	0.57	12569.0	12590.0
9	selling_price~age+ engine	2	0.75	8419.0	8440.0
10	selling_price~age+ C(fuel)	2	0.58	12441.0	12462.0
11	selling_price~age+ C(transmission)	2	0.61	11900.0	11921.0
12	selling_price~age+ C(owner)	2	0.5	13858.0	13886.0
13	selling_price~km_driven+ mileage	2	0.07	18675.0	18696.0
14	selling_price~km_driven+ engine	2	0.4	15255.0	15276.0
15	selling_price~km_driven+ C(fuel)	2	0.22	17257.0	17278.0
16	selling_price~km_driven+ C(transmission)	2	0.27	16721.0	16742.0
17	selling_price~km_driven+ C(owner)	2	0.18	17700.0	17727.0
18	selling_price~mileage+ engine	2	0.37	15576.0	15596.0
19	selling_price~mileage+ C(fuel)	2	0.1	18357.0	18378.0
20	selling_price~mileage+ C(transmission)	2	0.25	16922.0	16942.0
21	selling_price~mileage+ C(owner)	2	0.17	17759.0	17787.0
22	selling_price~engine+ C(fuel)	2	0.27	16747.0	16768.0
23	selling_price~engine+ C(transmission)	2	0.4	15171.0	15192.0
24	selling_price~engine+ C(owner)	2	0.43	14817.0	14845.0
25	selling_price~C(fuel)+ C(transmission)	2	0.36	15709.0	15730.0
26	selling_price~C(fuel)+ C(owner)	2	0.27	16771.0	16799.0
27	selling_price~C(transmission)+ C(owner)	2	0.36	15785.0	15812.0
28	selling_price~age+ km_driven+ mileage	3	0.58	12526.0	12554.0
29	selling_price~age+ km_driven+ engine	3	0.75	8300.0	8328.0
30	selling_price~age+ km_driven+ C(fuel)	3	0.58	12419.0	12447.0
31	selling_price~age+ km_driven+ C(transmission)	3	0.62	11712.0	11740.0
32	selling_price~age+ km_driven+ C(owner)	3	0.5	13794.0	13829.0

33	selling_price~age+ mileage+ engine	3	0.75	8369.0	8397.0
34	selling_price~age+ mileage+ C(fuel)	3	0.67	10588.0	10616.0
35	selling_price~age+ mileage+ C(transmission)	3	0.64	11129.0	11157.0
36	selling_price~age+ mileage+ C(owner)	3	0.57	12536.0	12571.0
37	selling_price~age+ engine+ C(fuel)	3	0.75	8361.0	8389.0
38	selling_price~age+ engine+ C(transmission)	3	0.79	7051.0	7079.0
39	selling_price~age+ engine+ C(owner)	3	0.75	8325.0	8359.0
40	selling_price~age+ C(fuel)+ C(transmission)	3	0.7	9779.0	9806.0
41	selling_price~age+ C(fuel)+ C(owner)	3	0.58	12373.0	12408.0
42	selling_price~age+ C(transmission)+ C(owner)	3	0.61	11886.0	11921.0
43	selling_price~km_driven+ mileage+ engine	3	0.49	13972.0	14000.0
44	selling_price~km_driven+ mileage+ C(fuel)	3	0.24	17094.0	17122.0
45	selling_price~km_driven+ mileage+ C(transmission)	3	0.27	16716.0	16744.0
46	selling_price~km_driven+ mileage+ C(owner)	3	0.19	17556.0	17591.0
47	selling_price~km_driven+ engine+ C(fuel)	3	0.42	14989.0	15016.0
48	selling_price~km_driven+ engine+ C(transmission)	3	0.48	14152.0	14180.0
49	selling_price~km_driven+ engine+ C(owner)	3	0.49	13912.0	13947.0
50	selling_price~km_driven+ C(fuel)+ C(transmission)	3	0.42	14906.0	14934.0
51	selling_price~km_driven+ C(fuel)+ C(owner)	3	0.32	16169.0	16204.0
52	selling_price~km_driven+ C(transmission)+ C(owner)	3	0.36	15741.0	15776.0
53	selling_price~mileage+ engine+ C(fuel)	3	0.39	15407.0	15435.0
54	selling_price~mileage+ engine+ C(transmission)	3	0.51	13565.0	13593.0
55	selling_price~mileage+ engine+ C(owner)	3	0.49	13933.0	13968.0
56	selling_price~mileage+ C(fuel)+ C(transmission)	3	0.36	15695.0	15723.0
57	selling_price~mileage+ C(fuel)+ C(owner)	3	0.29	16588.0	16623.0
58	selling_price~mileage+ C(transmission)+ C(owner)	3	0.36	15785.0	15820.0
59	selling_price~engine+ C(fuel)+ C(transmission)	3	0.43	14888.0	14916.0
60	selling_price~engine+ C(fuel)+ C(owner)	3	0.44	14731.0	14766.0
61	selling_price~engine+ C(transmission)+ C(owner)	3	0.52	13411.0	13446.0
62	selling_price~C(fuel)+ C(transmission)+ C(owner)	3	0.47	14227.0	14262.0
63	selling_price~age+ km_driven+ mileage+ engine	4	0.76	8221.0	8256.0
64	selling_price~age+ km_driven+ mileage+ C(fuel)	4	0.67	10520.0	10555.0
65	selling_price~age+ km_driven+ mileage+ C(transmission)	4	0.65	10977.0	11012.0
66	selling_price~age+ km_driven+ mileage+ C(owner)	4	0.58	12483.0	12525.0
67	selling_price~age+ km_driven+ engine+ C(fuel)	4	0.76	8184.0	8219.0
68	selling_price~age+ km_driven+ engine+ C(transmission)	4	0.79	7031.0	7066.0
69	selling_price~age+ km_driven+ engine+ C(owner)	4	0.76	8225.0	8267.0
70	selling_price~age+ km_driven+ C(fuel)+ C(transmission)	4	0.7	9781.0	9815.0
71	selling_price~age+ km_driven+ C(fuel)+ C(owner)	4	0.58	12358.0	12400.0
72	selling_price~age+ km_driven+ C(transmission)+ C(owner)	4	0.62	11684.0	11725.0
73	selling_price~age+ mileage+ engine+ C(fuel)	4	0.75	8350.0	8385.0
74	selling_price~age+ mileage+ engine+ C(transmission)	4	0.79	6868.0	6903.0
75	selling_price~age+ mileage+ engine+ C(owner)	4	0.75	8272.0	8313.0
76	selling_price~age+ mileage+ C(fuel)+ C(transmission)	4	0.74	8557.0	8592.0

77	selling_price~age+ mileage+ C(fuel)+ C(owner)	4	0.67	10490.0	10532.0
78	selling_price~age+ mileage+ C(transmission)+ C(owner)	4	0.65	11108.0	11150.0
79	selling_price~age+ engine+ C(fuel)+ C(transmission)	4	0.8	6742.0	6777.0
80	selling_price~age+ engine+ C(fuel)+ C(owner)	4	0.75	8255.0	8297.0
81	selling_price~age+ engine+ C(transmission)+ C(owner)	4	0.79	6978.0	7020.0
82	selling_price~age+ C(fuel)+ C(transmission)+ C(owner)	4	0.7	9722.0	9764.0
83	selling_price~km_driven+ mileage+ engine+ C(fuel)	4	0.49	13965.0	14000.0
84	selling_price~km_driven+ mileage+ engine+ C(transmission)	4	0.58	12526.0	12561.0
85	selling_price~km_driven+ mileage+ engine+ C(owner)	4	0.55	12919.0	12961.0
86	selling_price~km_driven+ mileage+ C(fuel)+ C(transmission)	4	0.42	14898.0	14933.0
87	selling_price~km_driven+ mileage+ C(fuel)+ C(owner)	4	0.35	15833.0	15875.0
88	selling_price~km_driven+ mileage+ C(transmission)+ C(owner)	4	0.36	15737.0	15779.0
89	selling_price~km_driven+ engine+ C(fuel)+ C(transmission)	4	0.51	13596.0	13630.0
90	selling_price~km_driven+ engine+ C(fuel)+ C(owner)	4	0.51	13638.0	13680.0
91	selling_price~km_driven+ engine+ C(transmission)+ C(owner)	4	0.56	12850.0	12891.0
92	selling_price~km_driven+ C(fuel)+ C(transmission)+ C(owner)	4	0.5	13828.0	13870.0
93	selling_price~mileage+ engine+ C(fuel)+ C(transmission)	4	0.52	13552.0	13586.0
94	selling_price~mileage+ engine+ C(fuel)+ C(owner)	4	0.49	13879.0	13921.0
95	selling_price~mileage+ engine+ C(transmission)+ C(owner)	4	0.59	12168.0	12209.0
96	selling_price~mileage+ C(fuel)+ C(transmission)+ C(owner)	4	0.47	14209.0	14251.0
97	selling_price~engine+ C(fuel)+ C(transmission)+ C(owner)	4	0.55	13061.0	13102.0
98	selling_price~age+ km_driven+ mileage+ engine+ C(fuel)	5	0.76	8172.0	8214.0
99	selling_price~age+ km_driven+ mileage+ engine+ C(transmission)	5	0.8	6828.0	6870.0
100	selling_price~age+ km_driven+ mileage+ engine+ C(owner)	5	0.76	8146.0	8195.0
101	selling_price~age+ km_driven+ mileage+ C(fuel)+ C(transmission)	5	0.74	8546.0	8588.0
102	selling_price~age+ km_driven+ mileage+ C(fuel)+ C(owner)	5	0.67	10436.0	10485.0
103	selling_price~age+ km_driven+ mileage+ C(transmission)+ C(owner)	5	0.65	10941.0	10990.0
104	selling_price~age+ km_driven+ engine+ C(fuel)+ C(transmission)	5	0.8	6668.0	6710.0
105	selling_price~age+ km_driven+ engine+ C(fuel)+ C(owner)	5	0.76	8101.0	8149.0
106	selling_price~age+ km_driven+ engine+ C(transmission)+ C(owner)	5	0.79	6965.0	7014.0
107	selling_price~age+ km_driven+ C(fuel)+ C(transmission)+ C(owner)	5	0.7	9724.0	9773.0
108	selling_price~age+ mileage+ engine+ C(fuel)+ C(transmission)	5	0.8	6719.0	6761.0
109	selling_price~age+ mileage+ engine+ C(fuel)+ C(owner)	5	0.75	8246.0	8295.0
110	selling_price~age+ mileage+ engine+ C(transmission)+ C(owner)	5	0.8	6792.0	6841.0
111	selling_price~age+ mileage+ C(fuel)+ C(transmission)+ C(owner)	5	0.75	8475.0	8523.0
112	selling_price~age+ engine+ C(fuel)+ C(transmission)+ C(owner)	5	0.8	6649.0	6697.0
113	selling_price~km_driven+ mileage+ engine+ C(fuel)+ C(transmission)	5	0.58	12517.0	12558.0
114	selling_price~km_driven+ mileage+ engine+ C(fuel)+ C(owner)	5	0.55	12921.0	12970.0
115	selling_price~km_driven+ mileage+ engine+ C(transmission)+ C(owner)	5	0.63	11538.0	11587.0
116	selling_price~km_driven+ mileage+ C(fuel)+ C(transmission)+ C(owner)	5	0.5	13754.0	13803.0
117	selling_price~km_driven+ engine+ C(fuel)+ C(transmission)+ C(owner)	5	0.59	12282.0	12331.0
118	selling_price~mileage+ engine+ C(fuel)+ C(transmission)+ C(owner)	5	0.59	12169.0	12218.0
119	selling_price~age+ km_driven+ mileage+ engine+ C(fuel)+ C(transmission)	6	0.8	6643.0	6692.0
120	selling_price~age+ km_driven+ mileage+ engine+ C(fuel)+ C(owner)	6	0.76	8090.0	8146.0

121	selling_price~age+ km_driven+ mileage+ engine+ C(transmission)+ C(owner)	6	0.8	6762.0	6818.0
122	selling_price~age+ km_driven+ mileage+ C(fuel)+ C(transmission)+ C(owner)	6	0.75	8469.0	8525.0
123	selling_price~age+ km_driven+ engine+ C(fuel)+ C(transmission)+ C(owner)	6	0.8	6588.0	6643.0
124	selling_price~age+ mileage+ engine+ C(fuel)+ C(transmission)+ C(owner)	6	0.8	6627.0	6683.0
125	selling_price~km_driven+ mileage+ engine+ C(fuel)+ C(transmission)+ C(owner)	6	0.63	11504.0	11560.0
126	selling_price~age+ km_driven+ mileage+ engine+ C(fuel)+ C(transmission)+ C(owner)	7	0.8	6565.0	6628.0