CS-452 HW1 Report

Car Pricing Prediction by Linear Regression

Mustafa Alper Sayan

S015674

1. Introduction

In this assignment we are expected to do linear regression analysis on the car pricing dataset. We are expected understand the data, clean the data and implement linear regression to predict a cars price given a set of features. Then evaluate the success of the models by using metrics.

2. Methodology

a. The steps of the homework are as follows:

   a. Load the data
   b. Giving information about the dataset
   c. Visualization of the dataset for each dependent feature with independent feature
   d. Feature engineering
   e. Training LR models for 3 different setups
   f. Evaluating the models with different metrics
   g. Printing general formula for the models
   h. Discussing coefficients

b. Linear Regression

Regression analysis is a statistical method performed to estimate the level effect of an independent variable (x) on a dependent variable (y). It is used to understand the relationships between a set of independent variables and dependent variables.
As an outcome of regression analysis, we get an equation called regression equation. The Formula is as follows for one dependent variable and one independent variable:

$$Y = bx + a$$

$a = intercept,\ b = slope$ in the above equations are parameters and they remain constant as x and y changes. By determining values of an and b we can calculate the value of y for a given x. For our case we will use this equation for prediction.

Linear regression is a supervised learning technique and it assumes the dependence of Y on X1, …, Xn is linear. To use linear regression data is linear and multi-collinearity is low. For the case in the homework we used multiple linear regression and it can be mathematically expressed as:

$$y = b0 + b1X1 + b2X2 + \cdots + bn\,Xn + E$$

Where:

y: dependent variable
b0: intercept
b1: coefficient of x1 (independent variable)
b2: coefficient of x2 (independent variable)
…
bn: coefficient of xn (independent variable)
E: Error

Regression line is a straight line that best fits the data meaning that the line attempts to define to define the value of y for a given x. The best fit regression line attempts to minimize the sum of the squared distances between the observed data points and the predicted ones. The formula for best fitting line or regression line is as follows:

$$y = a + bx$$

Where:

'b' is the slope of the line
'a' is the y-intercept
'x' is an explanatory variable
'y' is a dependent variable

For evaluating the success of a linear regression model following metrics is used:

c. Mean absolute Error (MAE): is the mean absolute value of the errors, calculated as follows:

$$MAE = \frac{1}{n}\sum_{j=1}^{n} |y_j - \ y_j|$$

d. Mean squared Error (MSE): is the mean of the squared errors and is calculated as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i)^2$$

e. Root Mean Squared Error (RMSE): is the square root of the mean of the squared errors, calculated as follows:

$$MSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i)^2}$$

3. Implementation Details

The details of this task as follows:
• Giving information about the linear regression
• Giving information about the dataset (e.g. what are the features? how many rows/columns? Numeric/categoric features? Any missing information in rows? The statistical information about the dataset and more
• Checking out the data (e.g. head, description, info, # of missing rows, correlation)
• Visualizing the data for each dependent feature with independent feature (e.g. Owner Type vs. Selling Price, Transmission vs. Selling Price)
• Feature engineering
• Training LR models for 3 different setups
• Evaluate the models with different metrics (mean-squared error, mean-absolute error, root-mean-squared error, r2 score)
• Visualize the results for 3 different setups for each metrics
• Printing the general formula of the model with best performing setup
• Discussing the model coefficients for 3 different setups
• Discussing the extracted information which may be useful for, for example, a car seller company, that's why we try to use ML or predictive analysis on such a case.

4. Results

Information about the car pricing dataset

a. what are the features? (before converting categorical variables to dummy variables)

Dependent features = 'year', 'km_driven', 'fuel', 'seller_type', 'transmission', 'owner', 'mileage', 'engine', 'max_power', 'seats'
Independent features = 'selling_price'

b. How many rows/columns?

Rows = 8128
Columns = 12

c. Numeric/categorical features:

Numeric Features = 'year', 'km_driven', 'mileage', 'engine', 'max_power', 'seats'

Categorical features = 'fuel', 'seller_type', 'transmission', 'owner'

d. Any missing rows?

Below are some of the missing values

```
name              0
year              0
selling_price     0
km_driven         0
fuel              0
seller_type       0
transmission      0
owner             0
mileage         221
engine          221
max_power       215
seats           221
dtype: int64
```

e. Information about the dataset

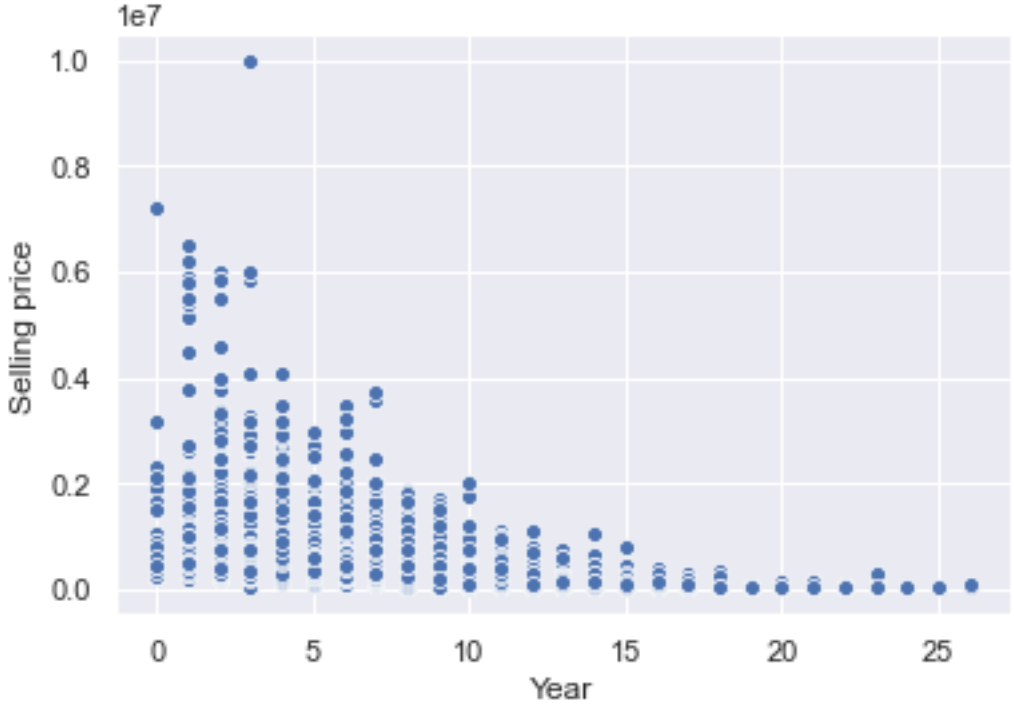
```
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           8128 non-null   object
 1   year           8128 non-null   int64
 2   selling_price  8128 non-null   int64
 3   km_driven      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  seats          7907 non-null   float64
dtypes: float64(1), int64(3), object(8)
memory usage: 762.1+ KB
```

f. Description about the dataset

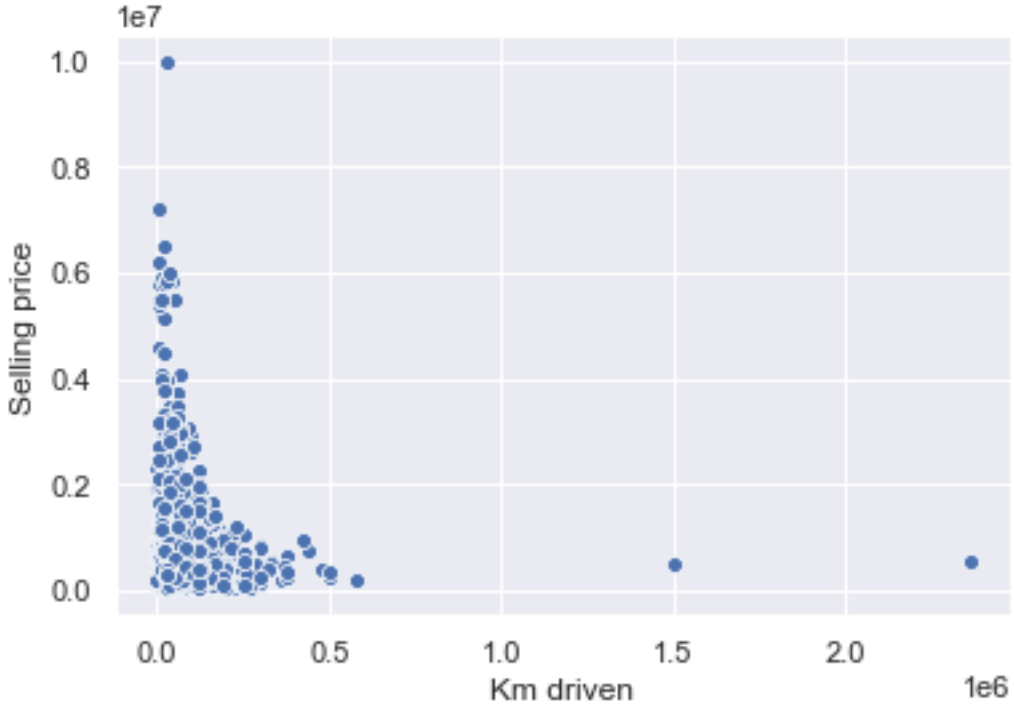
	year	selling_price	km_driven	seats
count	8128.000000	8.128000e+03	8.128000e+03	7907.000000
mean	2013.804011	6.382718e+05	6.981951e+04	5.416719
std	4.044249	8.062534e+05	5.655055e+04	0.959588
min	1983.000000	2.999900e+04	1.000000e+00	2.000000
25%	2011.000000	2.549990e+05	3.500000e+04	5.000000
50%	2015.000000	4.500000e+05	6.000000e+04	5.000000
75%	2017.000000	6.750000e+05	9.800000e+04	5.000000
max	2020.000000	1.000000e+07	2.360457e+06	14.000000

Visualization of the dataset witch each independent variable (Exploratory Data Analysis)

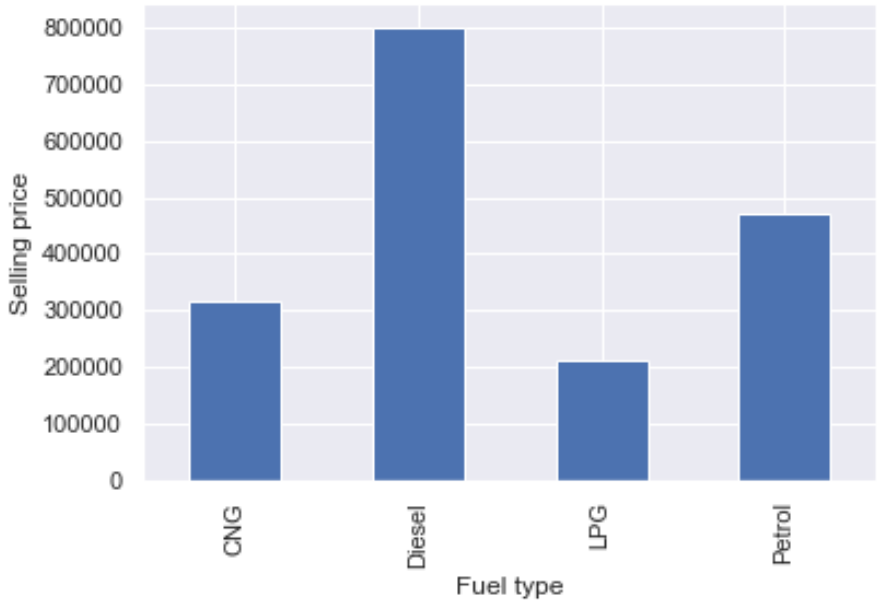
g. X = year, y = selling_price (year = negative influence on price)



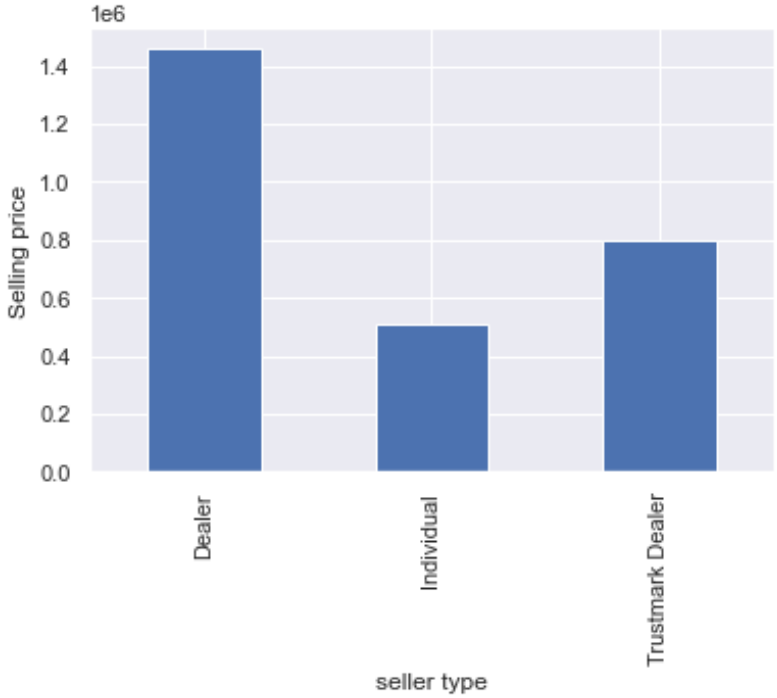
h. X = km_driven, y = selling_price (km driven = negative influence on price)



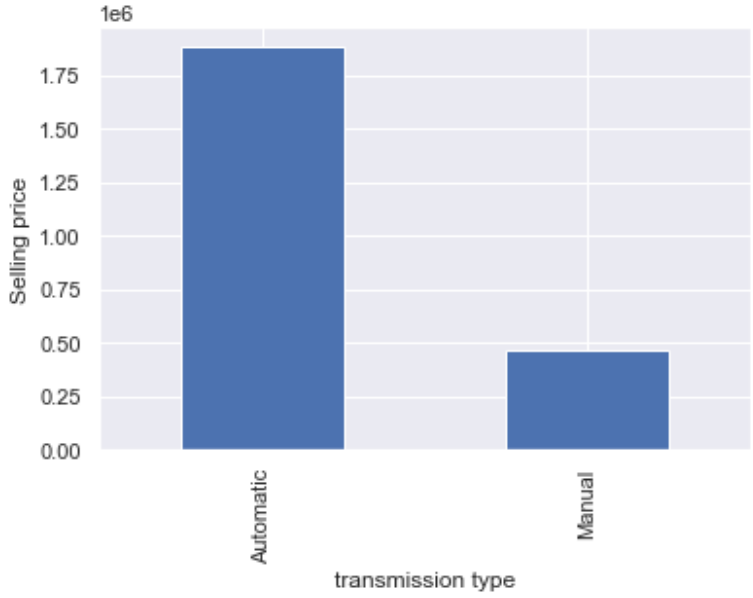
i. X = fuel, y = selling_price (diesel has positive influence on the price)



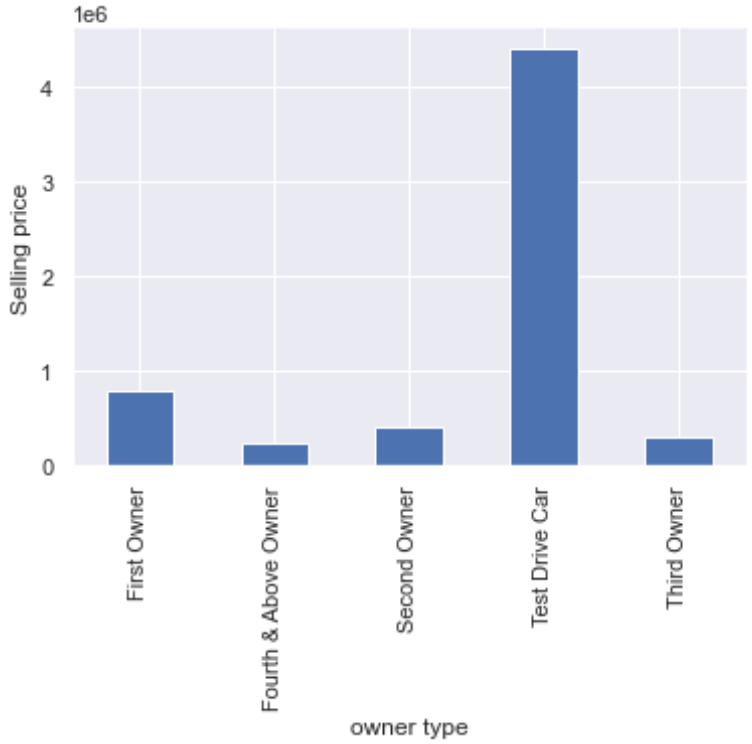
j. X = seller_type, y = selling_price (Dealer = Positive influence, individual = negative)



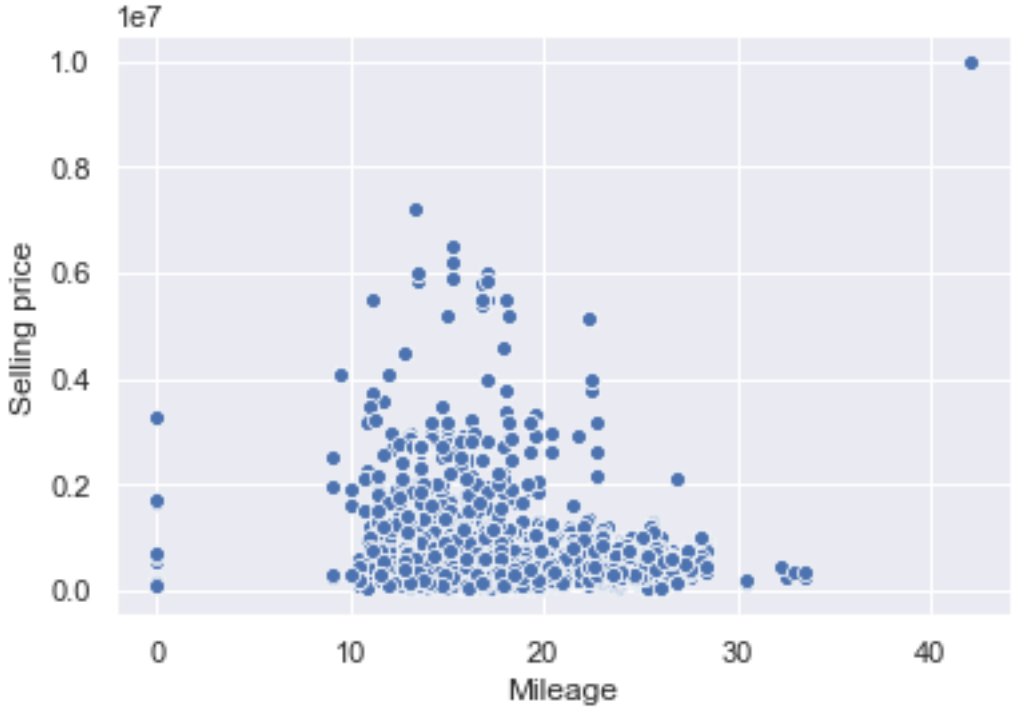
k. X= transmission, y = selling_price (automatic= positive, Manual= negative inf on y)



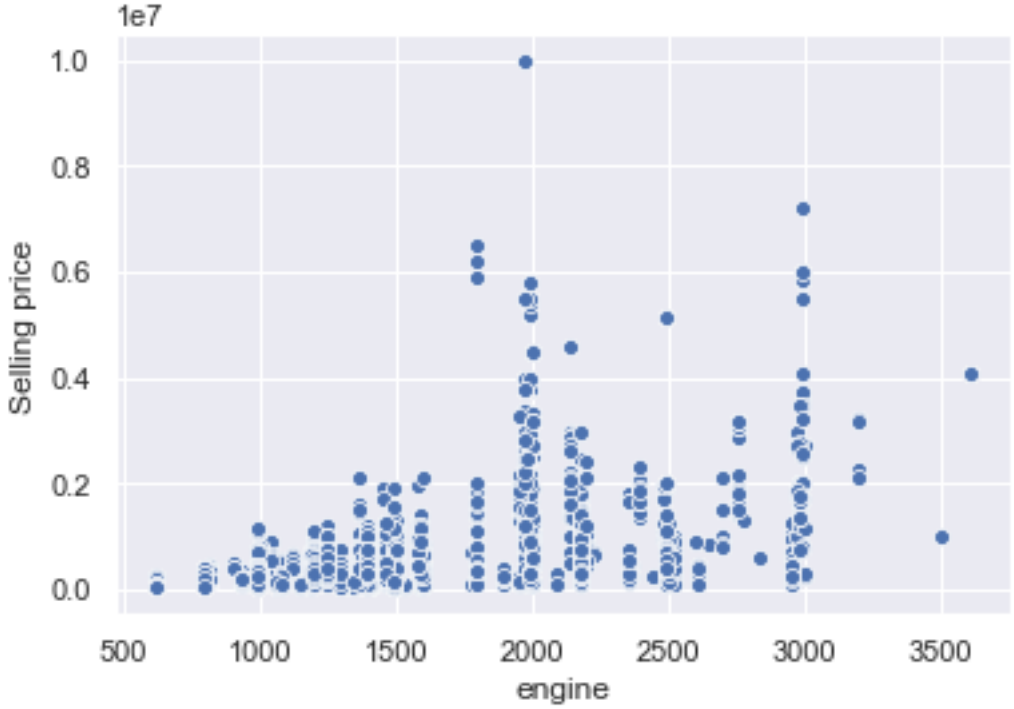
l. X= owner, y= selling_price



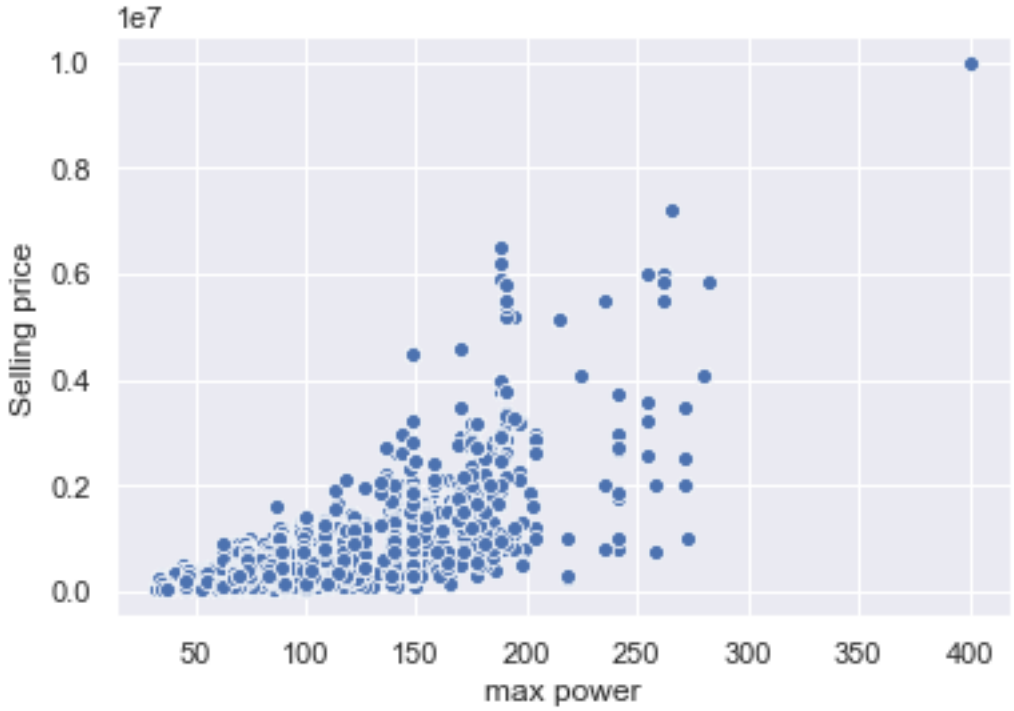
m. X= mileage, y= selling_price



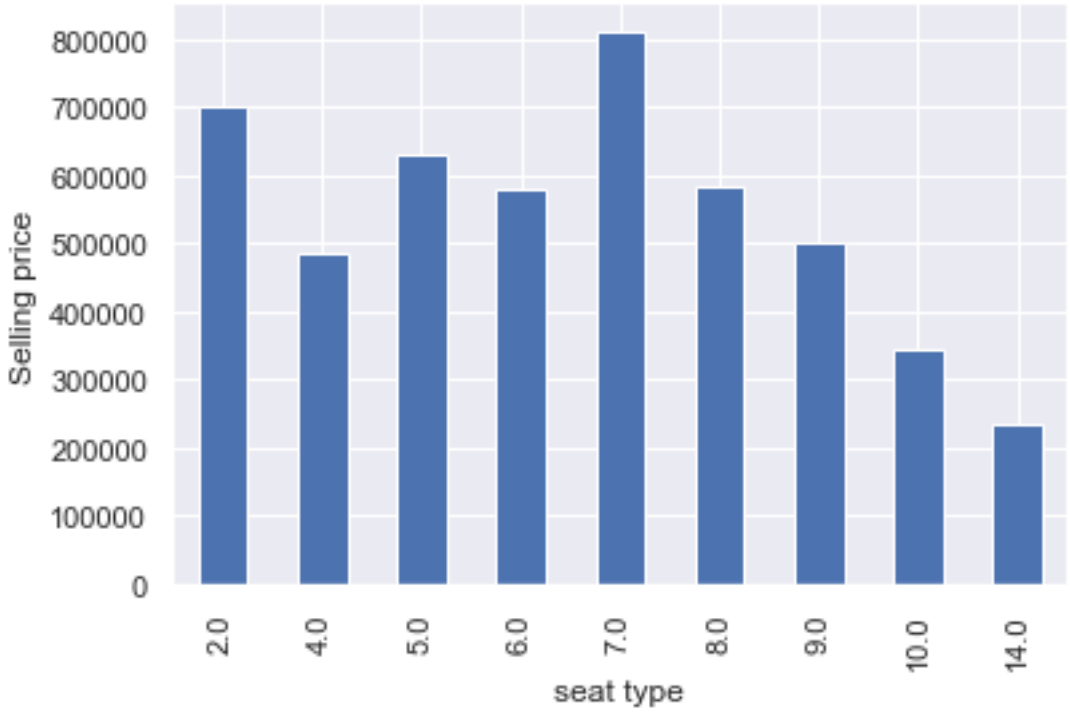
n. X= engine, y= selling_price



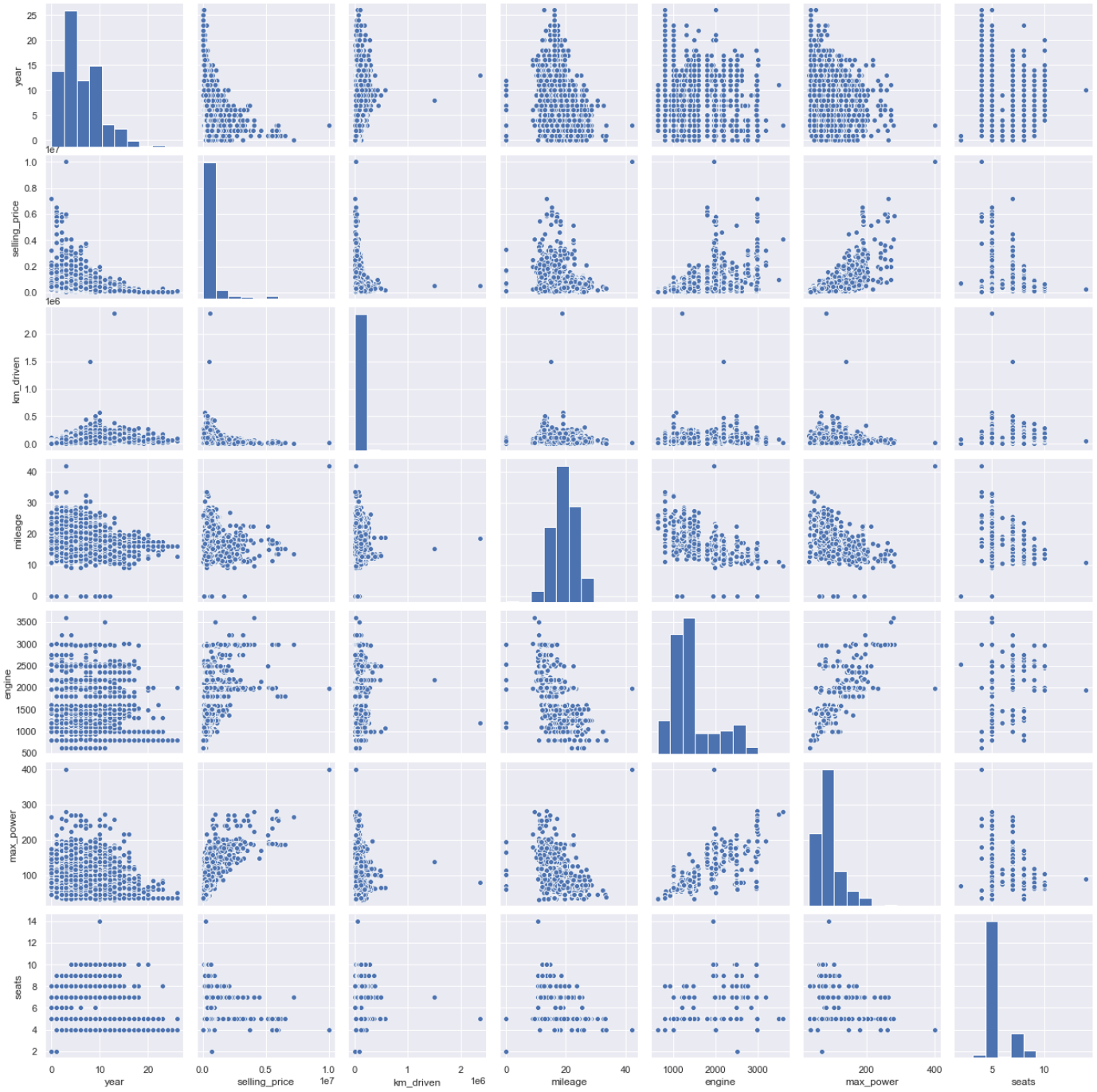
o. X= max_power, y= price



p. X= seats, y= price

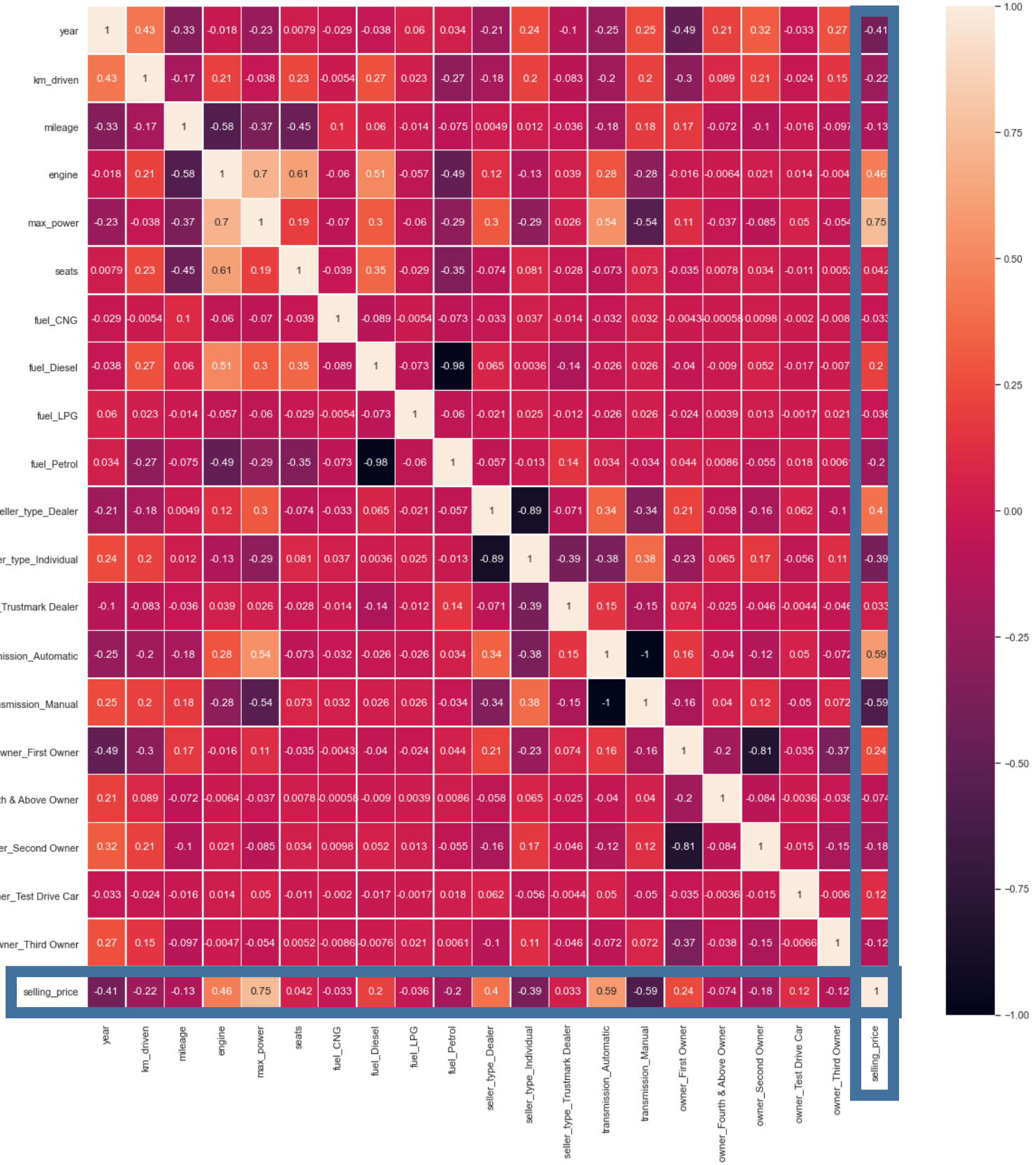


q. Pair plot


year
selling_price
km_driven
mileage
engine
max_power
seats
1e7
1e6

Feature engineering

r. Correlation table with using heatmap is as follows (values closer to absolute value of 1 in the column selling_price has better explanatory power, and should be prioritized on the selection of linear regression features):



s. Chosen feature sets:

   i. F1 = ['year', 'engine', 'max_power', 'transmission_Automatic', 'transmission_Manual']
   ii. F2 = ['seats', 'fuel_CNG', 'fuel_LPG', 'seller_type_Trustmark Dealer', 'owner_Fourth & Above Owner']
   iii. F3 = ['year', 'engine', 'max_power', 'transmission_Automatic', 'transmission_Manual' , 'km_driven','seller_type_Individual', 'fuel_Diesel', 'fuel_Petrol']

Evaluating each model

t. General formula for the linear models (**Best performing model is F3**)

   i. F1= y= x_1*-1246192.484+x_2*-191899.317+x_3*5164808.457+x_4*259956.233+x_5*-259956.233+357863.74251297204

   ii. F2= y= x_1*380294.698+x_2*-326423.05+x_3*-413004.664+x_4*105517.541+x_5*-430338.718+556522.5613735698

   iii. F3= y= x_1*-996762.562+x_2*-235209.077+x_3*5020071.848+x_4*228764.119+x_5*-228764.119+x_6*-2295432.097+x_7*-218023.028+x_8*-91210.937+x_9*-195599.279+697210.2873432674

u. Most influencing features for each formula (**discussing coefficients**):

   i. F1= X_3 (max_power) > X_1 (year) > X_4 (transmission_Automatic) = X_5 (transmission_Manual) > X_2 (engine)

   ii. F2 = X_5 (owner_Fourth & Above Owner) > X_3 (fuel_LPG) > X_1 (seats) > X_2 (fuel_CNG) > X_4 (seller_type_Trustmark Dealer)

   iii. F3 = X_3 (max_power) > X_6 (km_driven) > X_1 (year) > X_4 (transmission_Automatic) = X_5 (transmission_Manual) > X_7 (seller_type_Individual) > X_9 (fuel_Petrol) > X_8 (fuel_Diesel)

   iv. Discussion of coefficients:

   Let us examine X_3 in the feature set F1 is has this coefficient 5164808.457, in the feature set F3 X_3 has coefficient 5020071.848, this value is lower than the

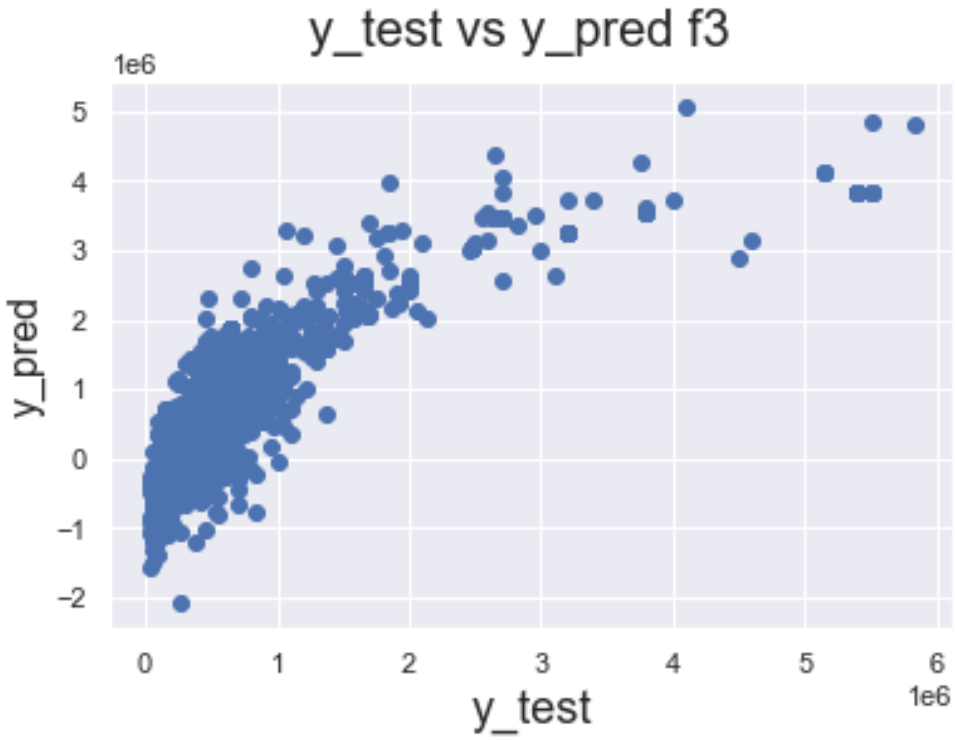
value in F1 meaning we decreased the importance of the X_3 feature by introducing new features to the feature set. Since we did not include many unnecessary features the drop in the importance of the value is not significant, but if we were to add more unnecessary features we would decrease the power of our model.

v. Success metrics for each feature set:
    i. F1
        1. Mean Squared Error = 402949895448.3637
        2. Absolute squared error = 482242.2692989118
        3. Root mean squared error = 634783.3452827537
        4. R2_score = 0.26207647757314045
    ii. F2
        1. Mean Squared Error = 538284237585.93787
        2. Absolute squared error = 391275.7243367282
        3. Root mean squared error = 733678.5655761914
        4. R2_score = 0.014238233703244951
    iii. F3
        1. Mean Squared Error = 304618179620.85095
        2. Absolute squared error = 420324.4960253233
        3. Root mean squared error = 551922.2586749432
        4. R2_score = 0.4421516852586419

    iv. Discussion of the success metrics:
    From looking at the success metrics above we can determine that the best performing model is F3. Because r2_score is highest and for the other metrics has the lowest value.

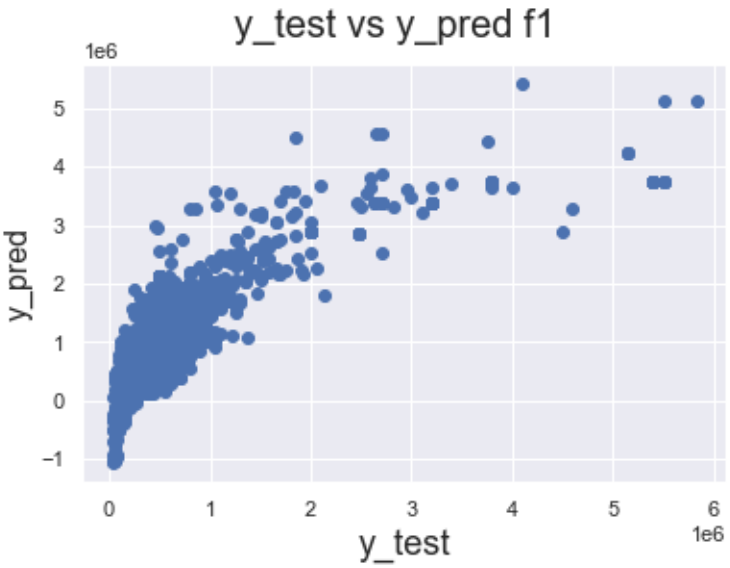
Visualizing results

w. Performing best



x. Performing worst

y. Performing okey



5. Conclusion

In this assignment linear regression model is used to predict the value of a selling price of a given car with selected features. I used correlation metric to tune best possible features to use in the regression meaning max power feature of a car with correlation 0.75 with respect to selling price should have more explanatory power then feature mileage which has correlation score 0.13 with respect to selling price. More could have been done to improve the performance of the model such as removing outliers (noise) from the training dataset. I purposely selected F2 with the worst correlation scores to see the difference in the metrics. As expected F2 performed worst in the scoring metrics. If we use all the features to train our linear regression model the performance is lower compared to F3 meaning, we over fitted our model and we are vulnerable to noise from the training data. Hence we can justify using 9 features to describe our data is enough.

If we were to give these results as a report to a car seller, we would use the correlation matrix and look at the selling price column the values between 0 – 1 would have positive correlation with the selling price of a car with values that are closer to 1 being influencing the price of a car more. Values between -1 – 0 would have a negative effect on the price of car.

we would say that features that are increasing the price of a car with the following order is:

max_power > automatic transmission > engine > seller type dealer > owner is first owner > fuel is diesel > test drive car

Features that are decreasing the price of a car witch the following order is:

Manual transmission > age of the car > seller type individual > km driven > petrol fuel > second owner > mileage