Graduate School of Engineering and Science

CS 552 – Data Science with Python

2020 FALL

Instructor : Dr. Reyhan AYDOĞAN

Assignment 1 : Car Pricing Prediction by Linear Regression

Salih Emre KAYNAR – S011496

Introduction

This assignment demonstrates a car pricing prediction by “linear regression” in Python. Linear Regression is a very common used approach for predicting a specific value based on previously known values. Linear regression model assumes a linear relationship between the input variables and the single output variable.

The aim of this assigment is finding a linear regression model for predicting the selling price of each car and find a relationship between the selling price and other features.

Methodology

Python 3.8.6 64-bit version is used on Jupyter IPython notebook in this assignment. The purpose of using ipython notebook is, we can benefit an interactive environment for coding.

For linear regression method we need to use some libraries such as : pandas (for dataframe work), numpy(for other libraries), matplotlib (for plotting), seaborn (for plotting), math(for mathematical operations), and the most important sklearn (for linear regression).

Data for the used cars is already given by the instructor and used for implementing linear regression model.

Implementation Details

Before starting linear regression, all the data must be fetched and checked for any misinformation that may affect our predictions.

First, we import the given “`car-pricing-data.csv`” with pandas `read_csv()`. This method imports the data to our notebook as a dataframe in order to process it.

After importing the data , a first looking the data will give us some idea about the dataset with pandas head() method as shown in the Table 1 below.

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_power	seats
0	Maruti Swift Dzire VDI	2014	450000	145500	Diesel	Individual	Manual	First Owner	23.4 kmpl	1248 CC	74 bhp	5.0
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	Diesel	Individual	Manual	Second Owner	21.14 kmpl	1498 CC	103.52 bhp	5.0
2	Honda City 2017-2020 EXi	2006	158000	140000	Petrol	Individual	Manual	Third Owner	17.7 kmpl	1497 CC	78 bhp	5.0
3	Hyundai i20 Sportz Diesel	2010	225000	127000	Diesel	Individual	Manual	First Owner	23.0 kmpl	1396 CC	90 bhp	5.0
4	Maruti Swift VXI BSIII	2007	130000	120000	Petrol	Individual	Manual	First Owner	16.1 kmpl	1298 CC	88.2 bhp	5.0

Table 1 – First 5 rows of the given data with pandas head()

In order to get more info of the feautures and stuff pandas info() is used.

```
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
```

Table 2 pandas info() output

```
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
```

Table 3 Number of missing data

As show above Table 1 we can see that the number of non-null counts are not equal. With that information we can understand that there is missing information in some of the rows. So `isnull().sum()` is used to see where are the missing data in Table 2. These missing data will manipulate our prediction rate in a bad way, so we need to get rid of them.

pandas `df.dropna(inplace=True)` is used for getting rid of rows with missing information. After that `df.isnull().sum()` is called again in order to be sure that there is no missing information.

As we can see from Table 2, we have object data types where it should be a numerical value. When we look at Table 1, there are units of some data with its numerical value. These features are “mileage(kmpl), engine(CC) and max_power (bhp)”. We need to change the data types of these columns in order to use it in linear regression.

For statistical data and linear regression, we need to have our all columns in numerical data type. To reach this goal we’ll be removing the unit strings at columns, changing categorical features in to dummies and removing that column.

Removing the letters of the data with unit is done by `str.replace("unit_lettes","")`. After that we have some rows with an empty string (`''`). While the all units are the same, unit names are added to name of the columns with unit. Then we’re changing the year value with age by taking 2020 as current year.

Now we don’t have any letters or characters in our data so, we can change their data types to `float` and `int` types with `.astype(data_type)`. After checking that the data is correctly converted, we can now proceed to statistical analysis of the data.

Before starting linear regression, it will be good to check some statistical data. These datas may show us some relation with our dependent features and selling price. `pandas describe()` gives us some basic statistical information about the data as shown below Table 4.

	age	selling_price	km_driven	mileage_kmpl	engine_CC	max_power_bhp	seats
count	7889.000000	7.889000e+03	7.889000e+03	7889.000000	7889.000000	7889.000000	7889.000000
mean	6.012169	6.496753e+05	6.919859e+04	19.461709	1458.378628	91.588665	5.418050
std	3.863460	8.134766e+05	5.682769e+04	3.938527	503.299977	35.731275	0.958526
min	0.000000	2.999900e+04	1.000000e+00	9.000000	624.000000	32.800000	4.000000
25%	3.000000	2.700000e+05	3.500000e+04	16.780000	1197.000000	68.050000	5.000000
50%	5.000000	4.500000e+05	6.000000e+04	19.330000	1248.000000	82.000000	5.000000
75%	8.000000	6.900000e+05	9.550000e+04	22.320000	1582.000000	102.000000	5.000000
max	26.000000	1.000000e+07	2.360457e+06	42.000000	3604.000000	400.000000	14.000000

Table 4 Basic stastical data about given file

`Pandas df.corrwith()` gives us a nice information about correlation between a target value with all other features which in our case the target value is 'selling_price'.

age	-0.412429
selling_price	1.000000
km_driven	-0.222696
mileage_kmpl	-0.128868
engine_CC	0.455077
max_power_bhp	0.749194
seats	0.041865

Table 5 Correlation rate

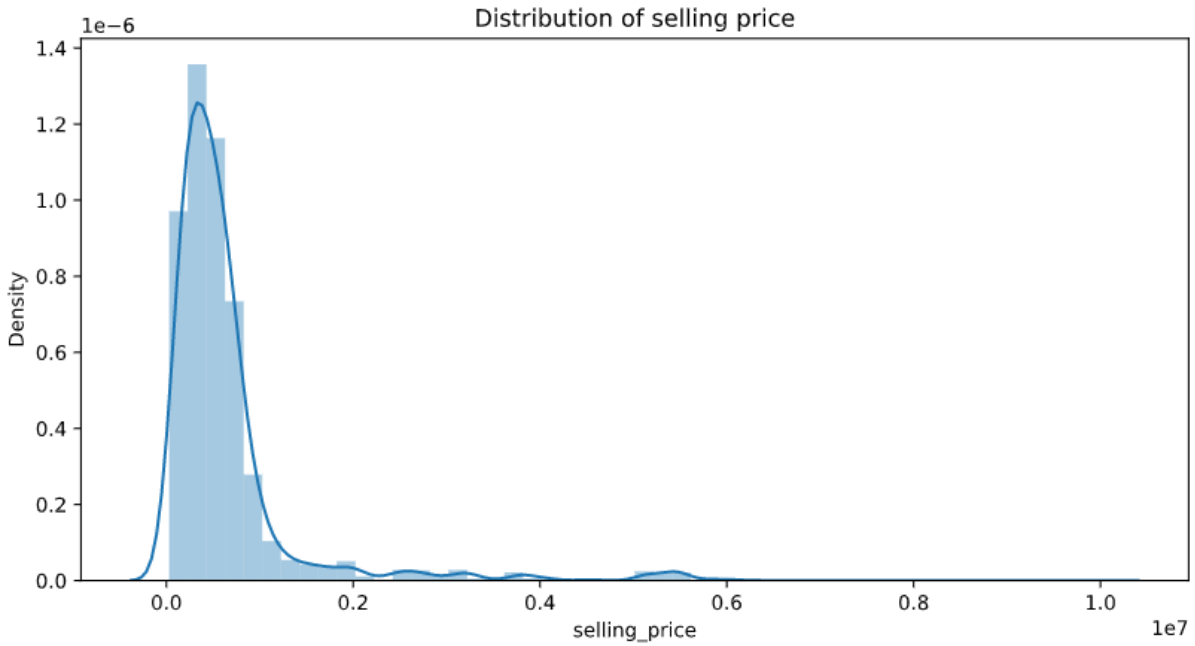
We can use Table 5 in order to select the data to make a linear regresssion practice in the later on next section.

Implementation of the statiscal data (e.g. distribution of selling price, age vs. selling price etc..) is done by seaborn library. These statistical data helps us to see the relationship with of every dependent features with indepent feature. The next statistical results of will be discussed in results part.

Another object type features we have are `name, fuel, seller type, transmission and owner`. These categorical values have converted into one-hot dummy format. With this method we can now insert out categorical features into linear regression.

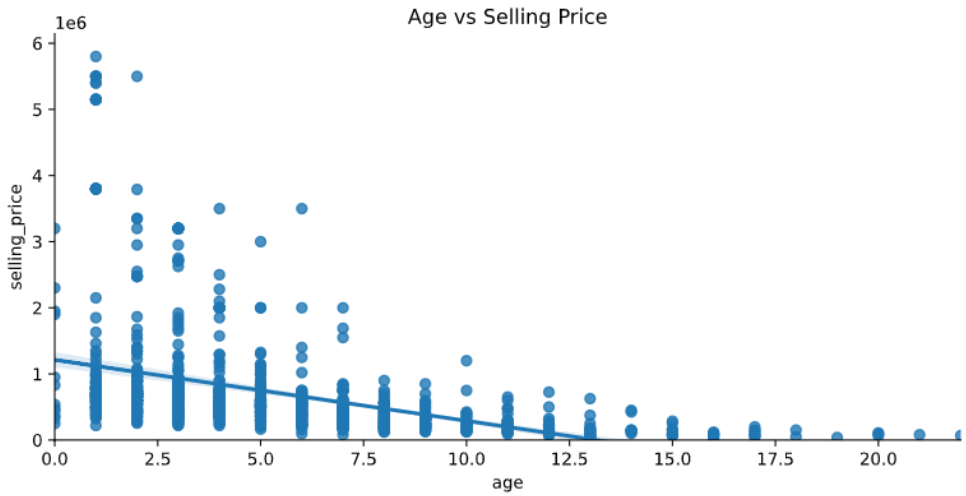
Lastly in linear regression part, the features are split into 3 different settings and the linear regression model is trained for that 3 different setups. Each model is evaluated with mean-squared error, mean-absolute error, root-mean-squared error and r2 score. For each model %20 of data is split for testing and %80 for training.

Before building a Linear Regression model for predicting our selling price, a quick look at statistics may improve our score on linear regression. At a first look distribution of selling price gives us an idea about the density of selling price as shown in Graph 1 below.



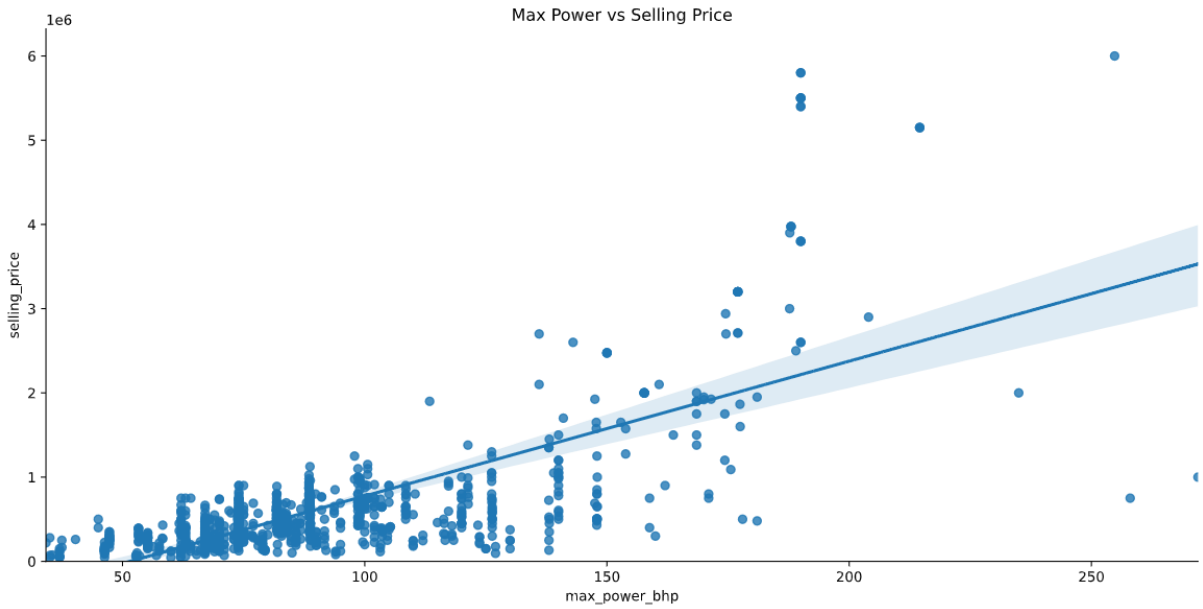
Graph 1 Distribution of Selling Price

In Table 5 above, we see the corrleation rates of each feature with numerical data type. All of the features gives us a very good information about the necessary data to take place in our multiple linear regression model.



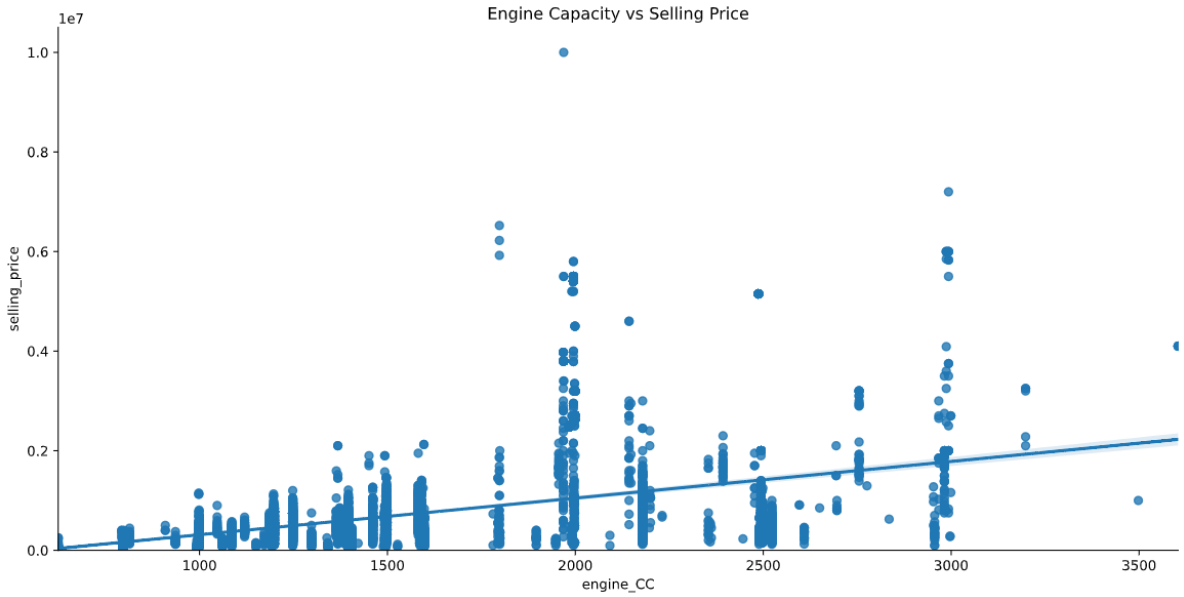
Graph 2 Age vs Selling Price

As we look at Graph 2 above, we can see an inverse correlation between age and selling price in a rate of 0.41. We can say that the price is higher for newer cars.



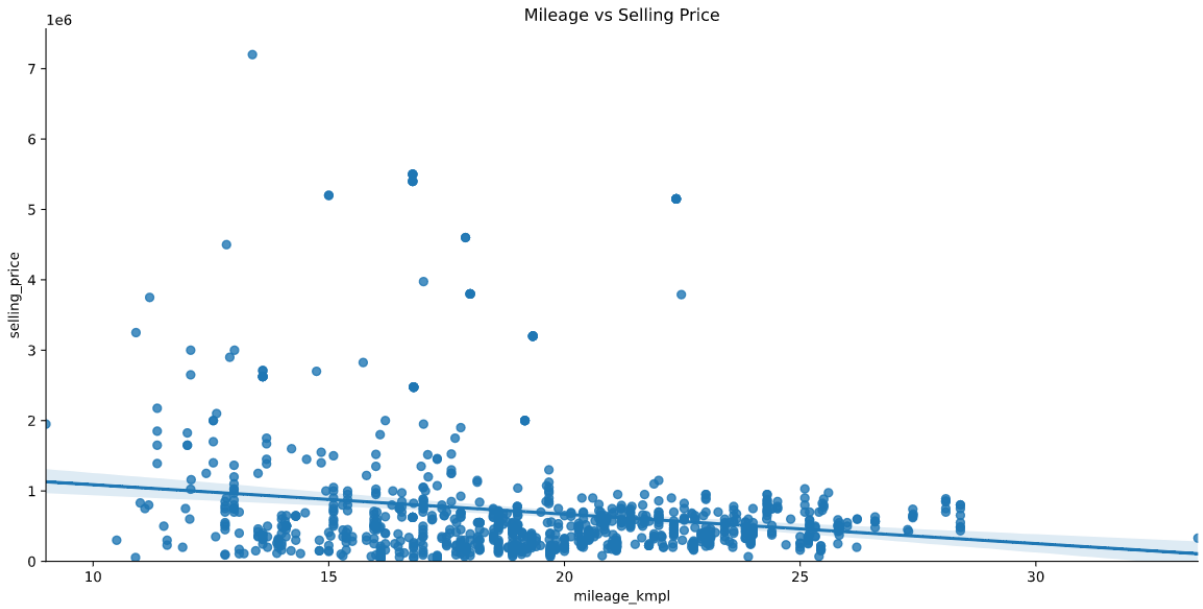
Graph 3 Max Power (bhp) vs Selling Price

In Graph 3 we can see a high correlation rate between maximum power and selling price with a rate of 0.75. This feauture looks like the used as a first dependent feature in LR part.



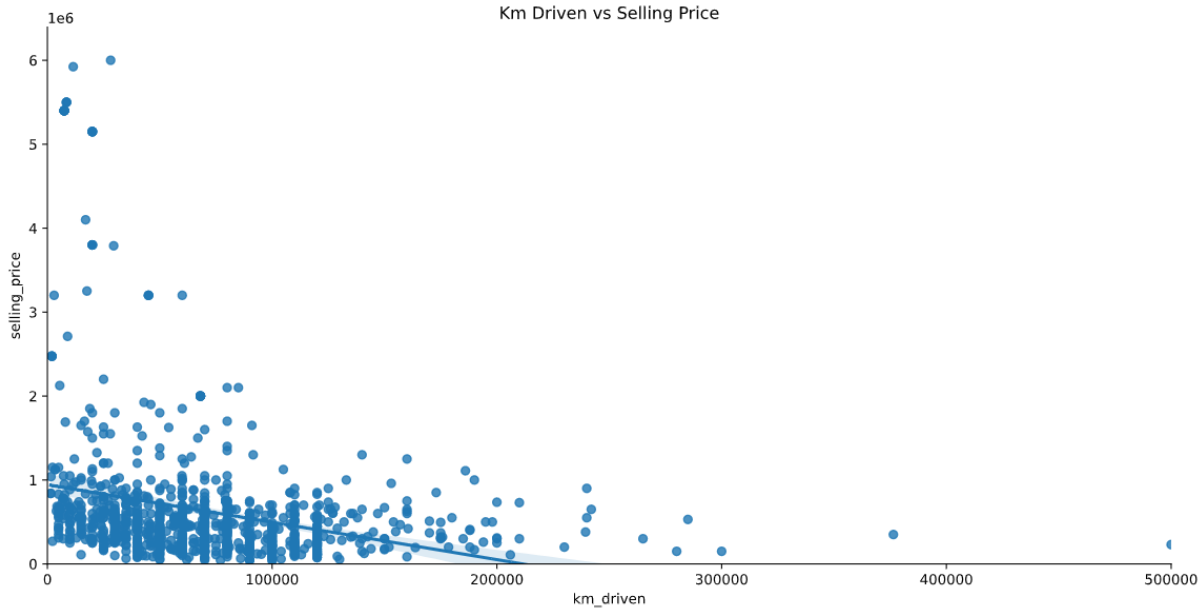
Graph 4 Engine Capacity vs Selling Price

Engine Capacity vs Selling Price Graph is directly correlated in a rate of 0.46 as shown in Graph 4.



Graph 5 Mileage vs Selling Price

We see a little correlation between the mileage and selling price in Graph 5 in a rate of 0.13



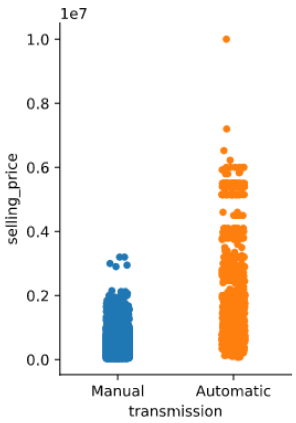
Graph 6 Kilometers driven vs Selling Price

As we can guees as the odometer value increases the price is going down. They're inversely correlated in a rate of 0.22.

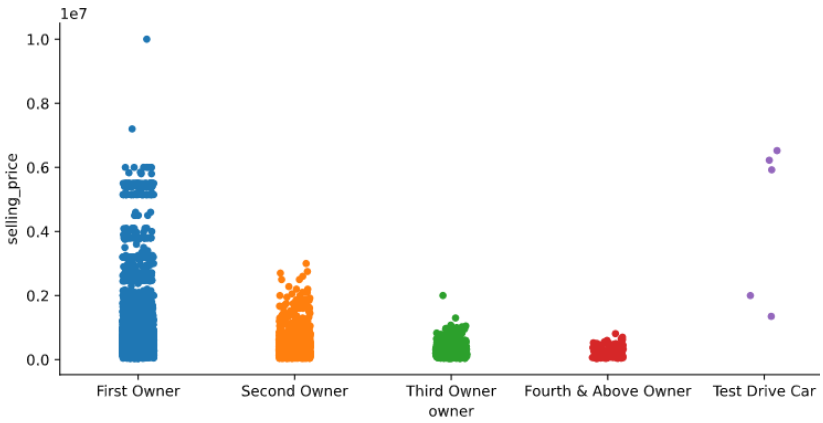
Seats vs Selling price has a correlation rate of 0.04 which is really close to 0 and may disturb our regression in a bad way.

When we put in an order of the correlation rates of the features discussed above, the highest correlating feature is max power, followed by engine capacity, age and km driven. Mileage and seats has a really small correlation rate.

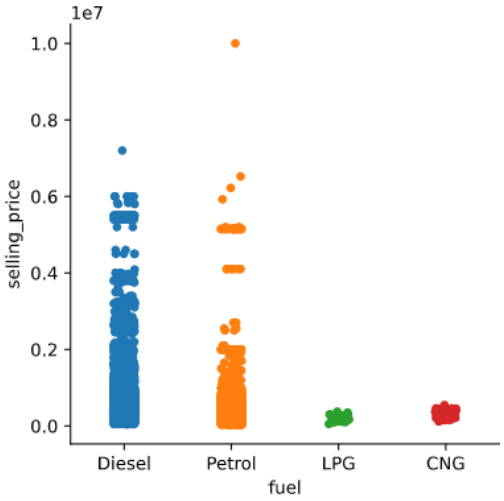
We have also the categorical values to be discussed. These values are transmission, fuel type, owner and seller type.



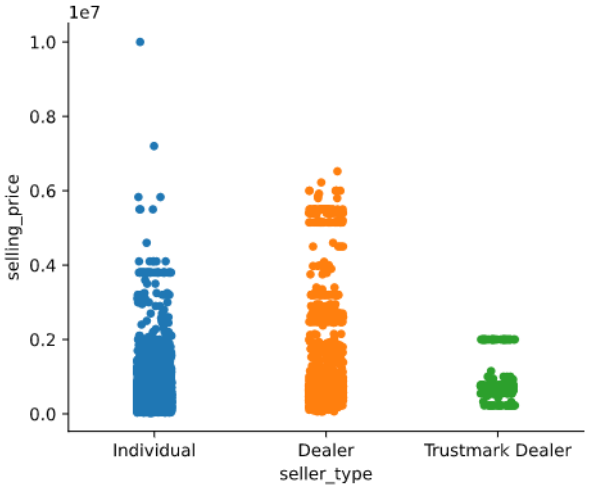
Graph 7 : Transmission vs Selling Price



Graph 8 : Owner and Selling price distribution



Graph 9 : Fuel vs Selling Price distribution



Graph 10 : Seller type vs Selling Price distribution

These 4 bar charts shows us visually how selling price is distributed between each categorical features.

In Graph 7 we can see that Automaric Transmission vehicles are more expensive than Manuals.

In Graph 8 Selling price is higher if the car is used by fewer owners.

In Graph 9 LPG and CNG fueled vehicles are cheaper than Petrol and Diesel.

In Graph 10 we can only say that Trustmark Dealer type sellers are cheaper than other so it won't help us the find correlation between seller type and selling price

Results

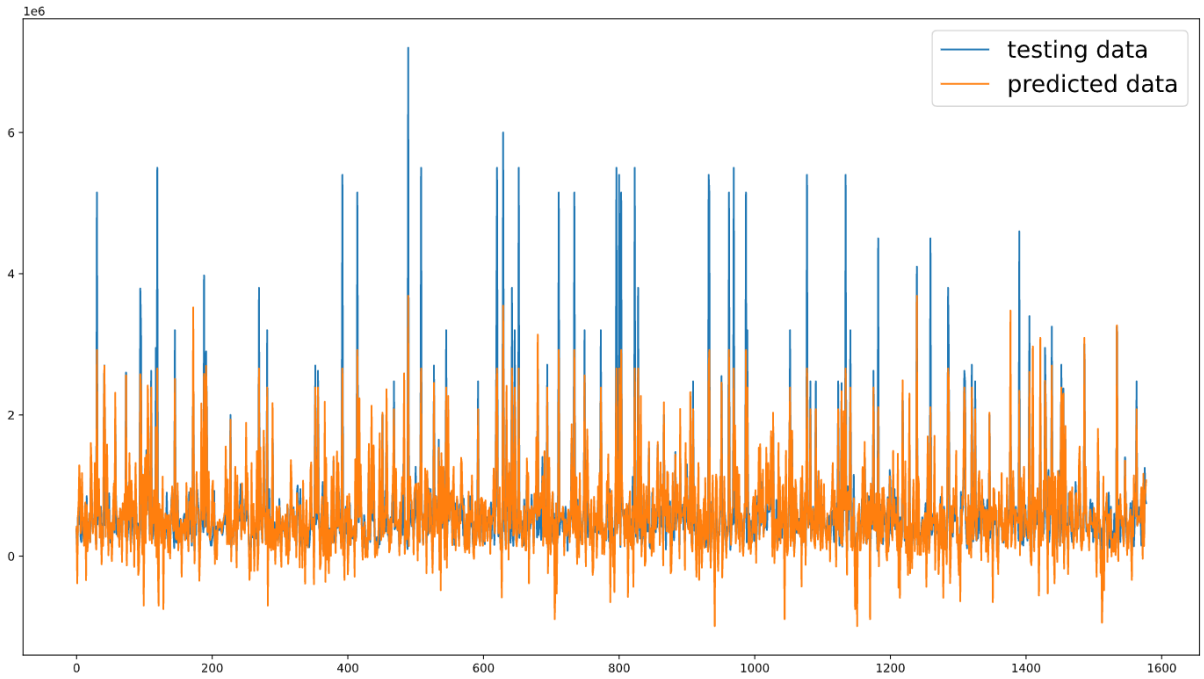
With all of that information we can now choose our feature set settings which has at least 4 feature in each setting. All of the data is split into 2 for training and testing. 80% of the data is reserved for training and the rest 20% is for testing purposes.

First Setting

As we saw above the correlation rates of each feature the first setting will be the most 4 highest correlation rate which were “max power”, “engine capacity”, “kilometers driven” and “age”.

With this setting we got these results :

- Mean Squared Error : 238888853880.32742
- Mean Absolute Error : 292521.00019099214
- Root Mean Squared Error : 488762.5741403769
- R_2 Score : 0.6328316279270444



Graph 11 Test vs Predicting data of Setting 1

The function for this model is

y = x_1*[6122821.63 -214879.65 -3022217.20 1143317.70]+[82386.50]

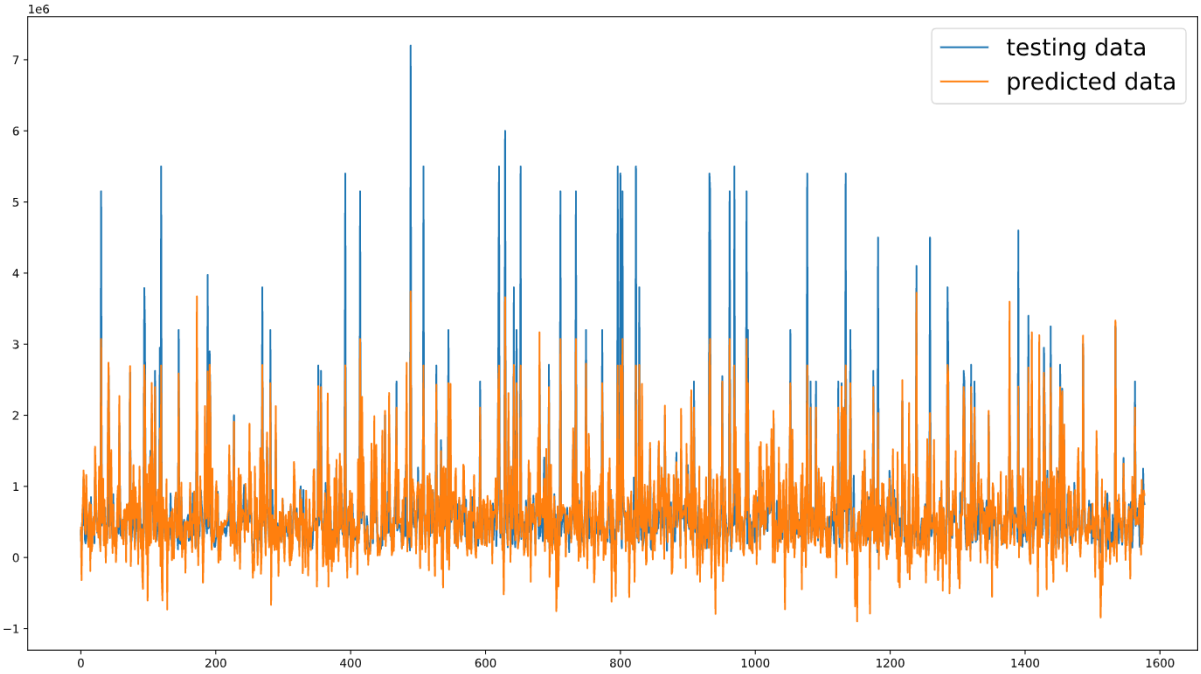
Second Setting

In first setting we used the most 4 correlated numeric features with a R2 Score : 0.63. In this setting we will be pushing some categorical values into set.

In Graph 7 we see a difference in selling price distribution between Automatic and Manual transmission, so, we'll be adding that value into the set and check the R2 score.

With this setting we got these results :

- Mean Squared Error : : 218941052035.0697
- Mean Absolute Error : 278051.9087452472
- Root Mean Squared Error : 467911.37198733445
- R_2 Score : 0.6634910823594654



Graph 12 Test vs Predicting data of Setting 2

Function for this model is :

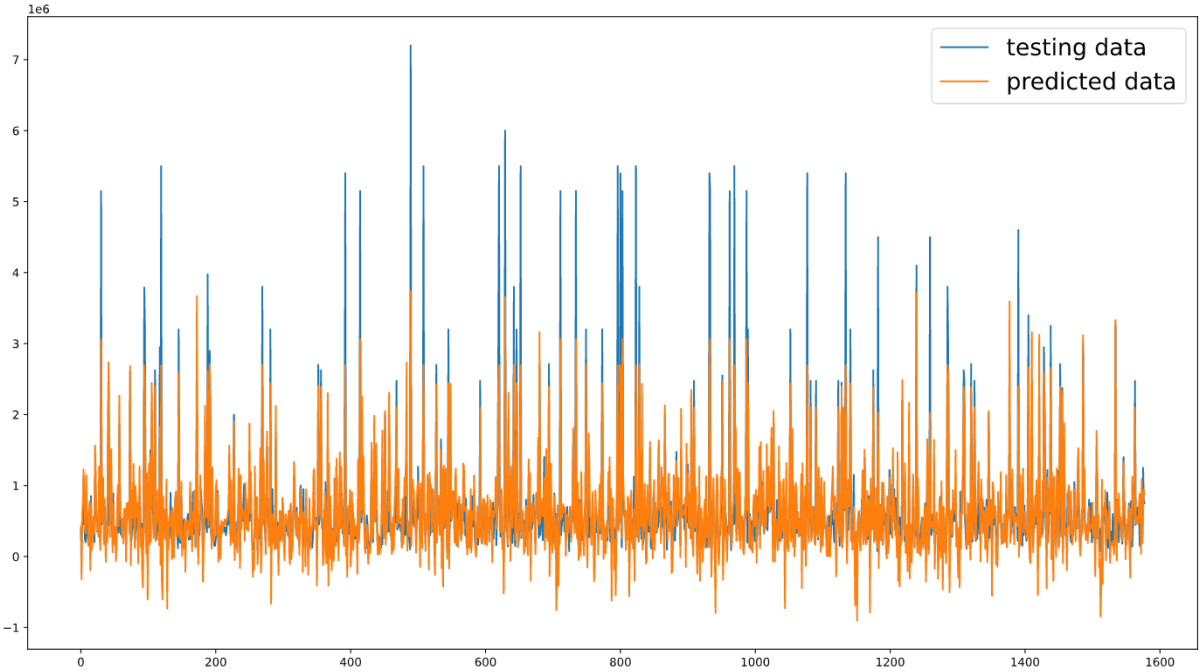
y = x_1*[5.271892 -1.08257 -2.309504 -1.072069 5.0567e+19 5.0567e+19]+[-5.0567e+19]

Third Setting

In third setting, engine capacity value is changed with mileage and fuel type is added into above 2nd setting.

With this setting we got these results :

- Mean Squared Error : : 218941052035.0697
- Mean Absolute Error : 278051.9087452472
- Root Mean Squared Error : 467911.37198733445
- R_2 Score : 0.6634910823594654



Graph 13 Test vs Predicting data of Setting 3

Function for this model is :

y = x_1*[5.21e+06 5.29e+05 -8.70e+05 -2.63e+06 -1.03e+19 -1.03e+19 3.44e+18 3.44e+18 3.44e+18 3.44e+18]+[6.85e+18]

Conclusion

In this report we have processed 7889 vehicles with unique features for a linear regression model for car pricing prediction.

As we can see from the Result part of this report as we add more variable into the features set our score has increased. But at the same time, we didn't use unrelated data like seats or seller type. We've checked for all relationship in statistical data part and eliminated them for linear regression model. Most correlating feature with selling price was max power. Max power positively influences the selling price of a car as seen above Graph 3. There were also features that inversely influencing the selling price prediction. For example, as seen above Graph 2 Age is inversely proportional to selling price. Our linear regression algorithm combined these data given to it and output the best predicted values for its own algorithm.

This model may be used for several applications for individual and even for professional in automotive industry market. Individuals may use this model for a market search for a new vehicle or may put his/her vehicle into market with the predicted price. Professionals may use this model for determining a price when entering a new market.