car price

March 12, 2022

1 Libraries

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

import warnings
warnings.filterwarnings("ignore")
```

2 Dataset

[2]:
```
df = pd.read_csv('CarPrice_Assignment.csv')
df.head()
```

[2]:
```
   car_ID  symboling                   CarName fueltype aspiration doornumber  \
0       1          3        alfa-romero giulia      gas        std        two
1       2          3       alfa-romero stelvio      gas        std        two
2       3          1  alfa-romero Quadrifoglio      gas        std        two
3       4          2               audi 100 ls      gas        std       four
4       5          2                audi 100ls      gas        std       four

       carbody drivewheel enginelocation  wheelbase  …  enginesize  \
0  convertible        rwd          front       88.6  …         130
1  convertible        rwd          front       88.6  …         130
2    hatchback        rwd          front       94.5  …         152
3        sedan        fwd          front       99.8  …         109
4        sedan        4wd          front       99.4  …         136

   fuelsystem  boreratio  stroke compressionratio horsepower  peakrpm citympg  \
0        mpfi       3.47    2.68              9.0        111     5000      21
1        mpfi       3.47    2.68              9.0        111     5000      21
2        mpfi       2.68    3.47              9.0        154     5000      19
3        mpfi       3.19    3.40             10.0        102     5500      24
4        mpfi       3.19    3.40              8.0        115     5500      18

   highwaympg    price
```

```
0          27  13495.0
1          27  16500.0
2          26  16500.0
3          30  13950.0
4          22  17450.0

[5 rows x 26 columns]
```

[3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   car_ID            205 non-null    int64
 1   symboling         205 non-null    int64
 2   CarName           205 non-null    object
 3   fueltype          205 non-null    object
 4   aspiration        205 non-null    object
 5   doornumber        205 non-null    object
 6   carbody           205 non-null    object
 7   drivewheel        205 non-null    object
 8   enginelocation    205 non-null    object
 9   wheelbase         205 non-null    float64
 10  carlength         205 non-null    float64
 11  carwidth          205 non-null    float64
 12  carheight         205 non-null    float64
 13  curbweight        205 non-null    int64
 14  enginetype        205 non-null    object
 15  cylindernumber    205 non-null    object
 16  enginesize        205 non-null    int64
 17  fuelsystem        205 non-null    object
 18  boreratio         205 non-null    float64
 19  stroke            205 non-null    float64
 20  compressionratio  205 non-null    float64
 21  horsepower        205 non-null    int64
 22  peakrpm           205 non-null    int64
 23  citympg           205 non-null    int64
 24  highwaympg        205 non-null    int64
 25  price             205 non-null    float64
dtypes: float64(8), int64(8), object(10)
memory usage: 41.8+ KB
```

[4]:
```
df.columns
```

[4]:
```
Index(['car_ID', 'symboling', 'CarName', 'fueltype', 'aspiration',
       'doornumber', 'carbody', 'drivewheel', 'enginelocation', 'wheelbase',
       'carlength', 'carwidth', 'carheight', 'curbweight', 'enginetype',
       'cylindernumber', 'enginesize', 'fuelsystem', 'boreratio', 'stroke',
       'compressionratio', 'horsepower', 'peakrpm', 'citympg', 'highwaympg',
       'price'],
      dtype='object')
```

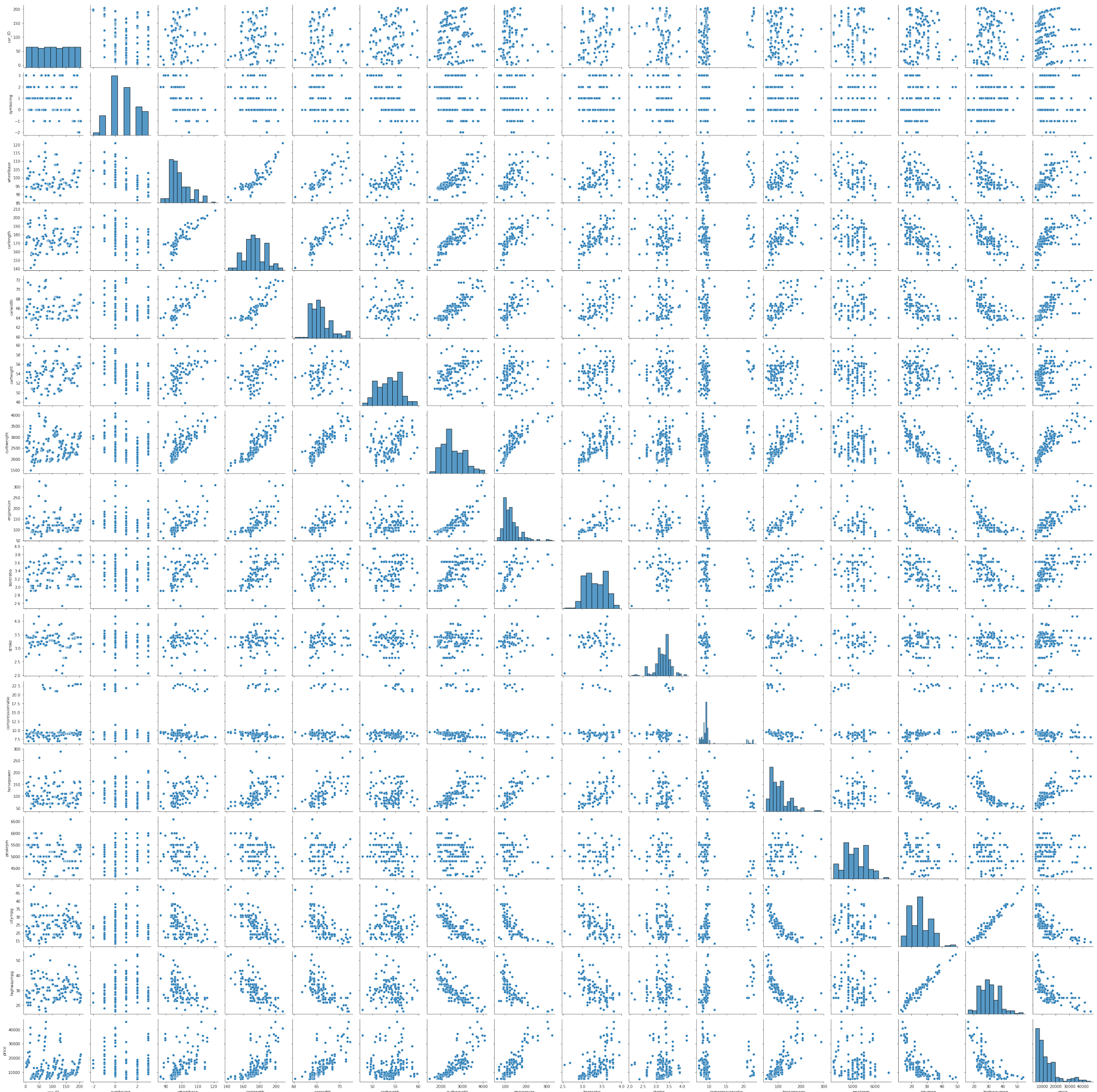
[5]:
```
cars_numeric = df.select_dtypes(include=['float64', 'int64'])
cars_numeric.head()
```

[5]:
```
   car_ID  symboling  wheelbase  carlength  carwidth  carheight  curbweight  \
0       1          3       88.6      168.8      64.1       48.8        2548
1       2          3       88.6      168.8      64.1       48.8        2548
2       3          1       94.5      171.2      65.5       52.4        2823
3       4          2       99.8      176.6      66.2       54.3        2337
4       5          2       99.4      176.6      66.4       54.3        2824

   enginesize  boreratio  stroke  compressionratio  horsepower  peakrpm  \
0         130       3.47    2.68               9.0         111     5000
1         130       3.47    2.68               9.0         111     5000
2         152       2.68    3.47               9.0         154     5000
3         109       3.19    3.40              10.0         102     5500
4         136       3.19    3.40               8.0         115     5500

   citympg  highwaympg    price
0       21          27  13495.0
1       21          27  16500.0
2       19          26  16500.0
3       24          30  13950.0
4       18          22  17450.0
```

[6]:
```
sns.pairplot(cars_numeric)
plt.show()
```



[7]:
```
cars_numeric.corr()
```

[7]:

	car_ID	symboling	wheelbase	carlength	carwidth	\
car_ID	1.000000	-0.151621	0.129729	0.170636	0.052387	
symboling	-0.151621	1.000000	-0.531954	-0.357612	-0.232919	
wheelbase	0.129729	-0.531954	1.000000	0.874587	0.795144	
carlength	0.170636	-0.357612	0.874587	1.000000	0.841118	
carwidth	0.052387	-0.232919	0.795144	0.841118	1.000000	
carheight	0.255960	-0.541038	0.589435	0.491029	0.279210	
curbweight	0.071962	-0.227691	0.776386	0.877728	0.867032	
enginesize	-0.033930	-0.105790	0.569329	0.683360	0.735433	
boreratio	0.260064	-0.130051	0.488750	0.606454	0.559150	
stroke	-0.160824	-0.008735	0.160959	0.129533	0.182942	

compressionratio	0.150276	-0.178515	0.249786	0.158414	0.181129
horsepower	-0.015006	0.070873	0.353294	0.552623	0.640732
peakrpm	-0.203789	0.273606	-0.360469	-0.287242	-0.220012
citympg	0.015940	-0.035823	-0.470414	-0.670909	-0.642704
highwaympg	0.011255	0.034606	-0.544082	-0.704662	-0.677218
price	-0.109093	-0.079978	0.577816	0.682920	0.759325

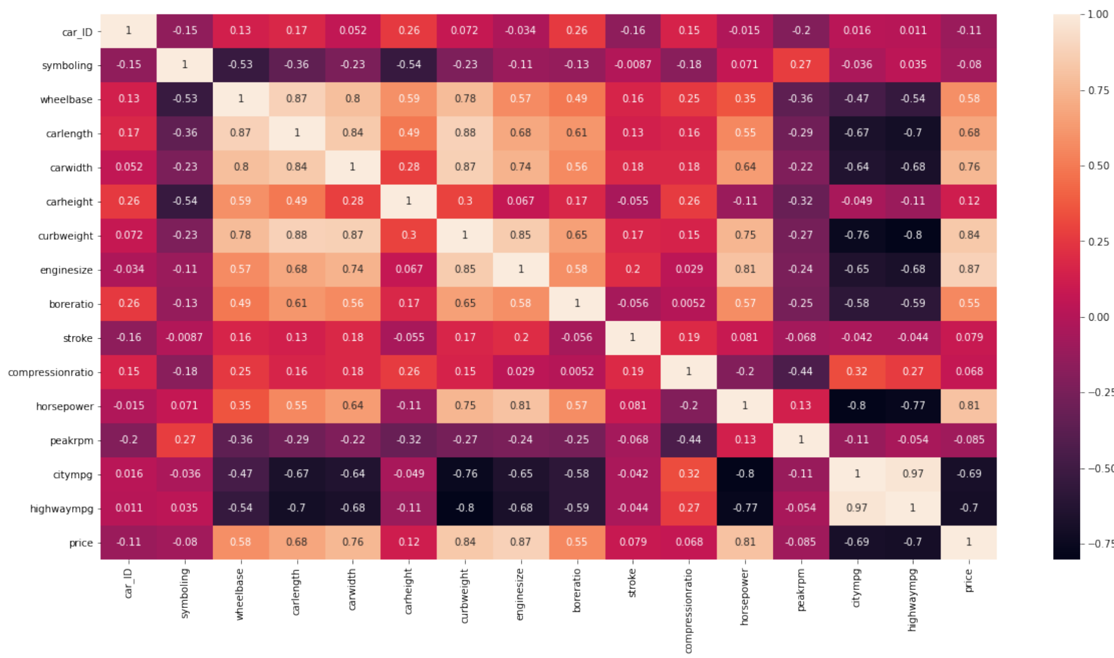
	carheight	curbweight	enginesize	boreratio	stroke
car_ID	0.255960	0.071962	-0.033930	0.260064	-0.160824
symboling	-0.541038	-0.227691	-0.105790	-0.130051	-0.008735
wheelbase	0.589435	0.776386	0.569329	0.488750	0.160959
carlength	0.491029	0.877728	0.683360	0.606454	0.129533
carwidth	0.279210	0.867032	0.735433	0.559150	0.182942
carheight	1.000000	0.295572	0.067149	0.171071	-0.055307
curbweight	0.295572	1.000000	0.850594	0.648480	0.168790
enginesize	0.067149	0.850594	1.000000	0.583774	0.203129
boreratio	0.171071	0.648480	0.583774	1.000000	-0.055909
stroke	-0.055307	0.168790	0.203129	-0.055909	1.000000
compressionratio	0.261214	0.151362	0.028971	0.005197	0.186110
horsepower	-0.108802	0.750739	0.809769	0.573677	0.080940
peakrpm	-0.320411	-0.266243	-0.244660	-0.254976	-0.067964
citympg	-0.048640	-0.757414	-0.653658	-0.584532	-0.042145
highwaympg	-0.107358	-0.797465	-0.677470	-0.587012	-0.043931
price	0.119336	0.835305	0.874145	0.553173	0.079443

	compressionratio	horsepower	peakrpm	citympg
car_ID	0.150276	-0.015006	-0.203789	0.015940
symboling	-0.178515	0.070873	0.273606	-0.035823
wheelbase	0.249786	0.353294	-0.360469	-0.470414
carlength	0.158414	0.552623	-0.287242	-0.670909
carwidth	0.181129	0.640732	-0.220012	-0.642704
carheight	0.261214	-0.108802	-0.320411	-0.048640
curbweight	0.151362	0.750739	-0.266243	-0.757414
enginesize	0.028971	0.809769	-0.244660	-0.653658
boreratio	0.005197	0.573677	-0.254976	-0.584532
stroke	0.186110	0.080940	-0.067964	-0.042145
compressionratio	1.000000	-0.204326	-0.435741	0.324701
horsepower	-0.204326	1.000000	0.131073	-0.801456
peakrpm	-0.435741	0.131073	1.000000	-0.113544
citympg	0.324701	-0.801456	-0.113544	1.000000
highwaympg	0.265201	-0.770544	-0.054275	0.971337
price	0.067984	0.808139	-0.085267	-0.685751

	highwaympg	price
car_ID	0.011255	-0.109093
symboling	0.034606	-0.079978
wheelbase	-0.544082	0.577816

```
carlength         -0.704662  0.682920
carwidth          -0.677218  0.759325
carheight         -0.107358  0.119336
curbweight        -0.797465  0.835305
enginesize        -0.677470  0.874145
boreratio         -0.587012  0.553173
stroke            -0.043931  0.079443
compressionratio   0.265201  0.067984
horsepower        -0.770544  0.808139
peakrpm           -0.054275 -0.085267
citympg            0.971337 -0.685751
highwaympg         1.000000 -0.697599
price             -0.697599  1.000000
```

[8]:
```python
plt.figure(figsize=(20,10))
sns.heatmap(cars_numeric.corr(), annot=True)
plt.show()
```



[9]:
```python
df['car_company'] = df['CarName'].apply(lambda x : x.split(' ')[0])
df.head()
```

[9]:
```
   car_ID  symboling                   CarName fueltype aspiration doornumber  \
0       1          3        alfa-romero giulia      gas        std        two
1       2          3       alfa-romero stelvio      gas        std        two
2       3          1  alfa-romero Quadrifoglio      gas        std        two
3       4          2               audi 100 ls      gas        std       four
```

```
4       5           2                audi  100ls      gas         std        four

      carbody drivewheel enginelocation  wheelbase  …  fuelsystem  \
0  convertible        rwd          front       88.6  …        mpfi
1  convertible        rwd          front       88.6  …        mpfi
2    hatchback        rwd          front       94.5  …        mpfi
3        sedan        fwd          front       99.8  …        mpfi
4        sedan        4wd          front       99.4  …        mpfi

   boreratio  stroke  compressionratio horsepower peakrpm  citympg highwaympg  \
0       3.47    2.68               9.0        111    5000       21         27
1       3.47    2.68               9.0        111    5000       21         27
2       2.68    3.47               9.0        154    5000       19         26
3       3.19    3.40              10.0        102    5500       24         30
4       3.19    3.40               8.0        115    5500       18         22

     price  car_company
0  13495.0  alfa-romero
1  16500.0  alfa-romero
2  16500.0  alfa-romero
3  13950.0         audi
4  17450.0         audi

[5 rows x 27 columns]
```

3 Spelling mistake fixing

[10]:
```
df.loc[(df['car_company']=='vw') | (df['car_company']=='vokswagen'),␣
 ↪'car_company']='volkswagen'
df.loc[(df['car_company']=='toyouta'), 'car_company']='toyota'
df.loc[(df['car_company']=='maxda'), 'car_company']='mazda'
df.loc[(df['car_company']=='Nissan'), 'car_company']='nissan'
df.loc[(df['car_company']=='porcshce'), 'car_company']='porsche'
# df['car_company'] = df['car_company'].str.replace('maxda', 'mazda')
```

[11]:
```
df['car_company'].value_counts()
```

[11]:
```
toyota         32
nissan         18
mazda          17
mitsubishi     13
honda          13
volkswagen     12
subaru         12
peugeot        11
volvo          11
```

```
dodge            9
buick            8
bmw              8
audi             7
plymouth         7
saab             6
porsche          5
isuzu            4
jaguar           3
chevrolet        3
alfa-romero      3
renault          2
mercury          1
Name: car_company, dtype: int64
```

[12]:
```
df = df.drop(['CarName'], axis=1)
```

[13]:
```
df['cylindernumber'].value_counts()
```

[13]:
```
four      159
six        24
five       11
eight       5
two         4
three       1
twelve      1
Name: cylindernumber, dtype: int64
```

[14]:
```
def number(x):
    return x.map({'four':4,
                  'six':6,
                  'five':5,
                  'eight':8,
                  'two':2,
                  'three':3,
                  'twelve':12})
```

[15]:
```
df['cylindernumber'] = df[['cylindernumber']].apply(number)
```

[16]:
```
df['doornumber'] = df[['doornumber']].apply(number)
```

[17]:
```
df_category = df.select_dtypes(include=['object'])
df_category
```

[17]:
```
    fueltype aspiration      carbody drivewheel enginelocation enginetype  \
0        gas        std  convertible        rwd          front       dohc
1        gas        std  convertible        rwd          front       dohc
```

```
2        gas      std  hatchback      rwd          front      ohcv
3        gas      std      sedan      fwd          front       ohc
4        gas      std      sedan      4wd          front       ohc
..       …        …        …          …            …            …
200      gas      std      sedan      rwd          front       ohc
201      gas    turbo      sedan      rwd          front       ohc
202      gas      std      sedan      rwd          front      ohcv
203   diesel    turbo      sedan      rwd          front       ohc
204      gas    turbo      sedan      rwd          front       ohc

    fuelsystem  car_company
0         mpfi  alfa-romero
1         mpfi  alfa-romero
2         mpfi  alfa-romero
3         mpfi         audi
4         mpfi         audi
..         …           …
200       mpfi        volvo
201       mpfi        volvo
202       mpfi        volvo
203        idi        volvo
204       mpfi        volvo

[205 rows x 8 columns]
```

[18]:
```python
df_dummies = pd.get_dummies(df_category, drop_first=True)
df_dummies.head()
```

[18]:
```
   fueltype_gas  aspiration_turbo  carbody_hardtop  carbody_hatchback  \
0             1                 0                0                  0
1             1                 0                0                  0
2             1                 0                0                  1
3             1                 0                0                  0
4             1                 0                0                  0

   carbody_sedan  carbody_wagon  drivewheel_fwd  drivewheel_rwd  \
0              0              0               0               1
1              0              0               0               1
2              0              0               0               1
3              1              0               1               0
4              1              0               0               0

   enginelocation_rear  enginetype_dohcv  …  car_company_nissan  \
0                    0                 0  …                   0
1                    0                 0  …                   0
2                    0                 0  …                   0
3                    0                 0  …                   0
```

```
4                    0                  0  …                      0

   car_company_peugeot  car_company_plymouth  car_company_porsche  \
0                    0                     0                    0
1                    0                     0                    0
2                    0                     0                    0
3                    0                     0                    0
4                    0                     0                    0

   car_company_renault  car_company_saab  car_company_subaru  \
0                    0                 0                   0
1                    0                 0                   0
2                    0                 0                   0
3                    0                 0                   0
4                    0                 0                   0

   car_company_toyota  car_company_volkswagen  car_company_volvo
0                   0                       0                  0
1                   0                       0                  0
2                   0                       0                  0
3                   0                       0                  0
4                   0                       0                  0

[5 rows x 43 columns]
```

[19]:
```
df = df.drop(list(df_category.columns), axis=1)
df.head()
```

[19]:
```
   car_ID  symboling  doornumber  wheelbase  carlength  carwidth  carheight  \
0       1          3           2       88.6      168.8      64.1       48.8
1       2          3           2       88.6      168.8      64.1       48.8
2       3          1           2       94.5      171.2      65.5       52.4
3       4          2           4       99.8      176.6      66.2       54.3
4       5          2           4       99.4      176.6      66.4       54.3

   curbweight  cylindernumber  enginesize  boreratio  stroke  \
0        2548               4         130       3.47    2.68
1        2548               4         130       3.47    2.68
2        2823               6         152       2.68    3.47
3        2337               4         109       3.19    3.40
4        2824               5         136       3.19    3.40

   compressionratio  horsepower  peakrpm  citympg  highwaympg    price
0               9.0         111     5000       21          27  13495.0
1               9.0         111     5000       21          27  16500.0
2               9.0         154     5000       19          26  16500.0
3              10.0         102     5500       24          30  13950.0
```

```
4                 8.0         115      5500       18            22  17450.0
```

[20]:
```python
df = pd.concat([df, df_dummies], axis=1)
df.head()
```

[20]:
```
   car_ID  symboling  doornumber  wheelbase  carlength  carwidth  carheight  \
0       1          3           2       88.6      168.8      64.1       48.8
1       2          3           2       88.6      168.8      64.1       48.8
2       3          1           2       94.5      171.2      65.5       52.4
3       4          2           4       99.8      176.6      66.2       54.3
4       5          2           4       99.4      176.6      66.4       54.3

   curbweight  cylindernumber  enginesize  …  car_company_nissan  \
0        2548               4         130  …                   0
1        2548               4         130  …                   0
2        2823               6         152  …                   0
3        2337               4         109  …                   0
4        2824               5         136  …                   0

   car_company_peugeot  car_company_plymouth  car_company_porsche  \
0                    0                     0                    0
1                    0                     0                    0
2                    0                     0                    0
3                    0                     0                    0
4                    0                     0                    0

   car_company_renault  car_company_saab  car_company_subaru  \
0                    0                 0                   0
1                    0                 0                   0
2                    0                 0                   0
3                    0                 0                   0
4                    0                 0                   0

   car_company_toyota  car_company_volkswagen  car_company_volvo
0                   0                       0                  0
1                   0                       0                  0
2                   0                       0                  0
3                   0                       0                  0
4                   0                       0                  0

[5 rows x 61 columns]
```

[21]:
```python
18+43
```

[21]: 61

[22]:
```python
df.drop(['car_ID'], axis=1, inplace=True)
```

[23]:
```
df.head()
```

[23]:
```
   symboling  doornumber  wheelbase  carlength  carwidth  carheight  \
0          3           2       88.6      168.8      64.1       48.8
1          3           2       88.6      168.8      64.1       48.8
2          1           2       94.5      171.2      65.5       52.4
3          2           4       99.8      176.6      66.2       54.3
4          2           4       99.4      176.6      66.4       54.3

   curbweight  cylindernumber  enginesize  boreratio  …  car_company_nissan  \
0        2548               4         130       3.47  …                   0
1        2548               4         130       3.47  …                   0
2        2823               6         152       2.68  …                   0
3        2337               4         109       3.19  …                   0
4        2824               5         136       3.19  …                   0

   car_company_peugeot  car_company_plymouth  car_company_porsche  \
0                    0                     0                    0
1                    0                     0                    0
2                    0                     0                    0
3                    0                     0                    0
4                    0                     0                    0

   car_company_renault  car_company_saab  car_company_subaru  \
0                    0                 0                   0
1                    0                 0                   0
2                    0                 0                   0
3                    0                 0                   0
4                    0                 0                   0

   car_company_toyota  car_company_volkswagen  car_company_volvo
0                   0                       0                  0
1                   0                       0                  0
2                   0                       0                  0
3                   0                       0                  0
4                   0                       0                  0

[5 rows x 60 columns]
```

4 Model Building

[24]:
```
from sklearn.model_selection import train_test_split
```

[25]:
```
df_train, df_test = train_test_split(df, train_size=0.7, random_state=100)
```

5 Standard Scaler

[26]:
```
from sklearn.preprocessing import StandardScaler
```

[27]:
```
sc = StandardScaler()
```

[28]:
```
varlist = ['symboling', 'doornumber', 'wheelbase', 'carlength', 'carwidth',
           'carheight', 'curbweight', 'cylindernumber', 'enginesize',␣
 ↪'boreratio',
           'stroke', 'compressionratio', 'horsepower', 'peakrpm', 'citympg',
           'highwaympg']
```

[29]:
```
df_train[varlist] = sc.fit_transform(df_train[varlist])
#df_test = sc.transform(df_train[varlist])
```

[30]:
```
df_train.head()
```

[30]:
```
     symboling  doornumber  wheelbase  carlength  carwidth  carheight  \
122   0.170159    0.887412  -0.811836  -0.487238 -0.924500  -1.134628
125   1.848278   -1.126872  -0.677177  -0.359789  1.114978  -1.382026
166   0.170159   -1.126872  -0.677177  -0.375720 -0.833856  -0.392434
1     1.848278   -1.126872  -1.670284  -0.367754 -0.788535  -1.959288
199  -1.507960    0.887412   0.972390   1.225364  0.616439   1.627983

     curbweight  cylindernumber  enginesize  boreratio  …  \
122   -0.642128       -0.351431   -0.660242  -1.297329  …
125    0.439415       -0.351431    0.637806   2.432256  …
166   -0.441296       -0.351431   -0.660242  -0.259197  …
1      0.015642       -0.351431    0.123485   0.625138  …
199    1.137720       -0.351431    0.123485   1.201877  …

     car_company_nissan  car_company_peugeot  car_company_plymouth  \
122                   0                    0                     1
125                   0                    0                     0
166                   0                    0                     0
1                     0                    0                     0
199                   0                    0                     0

     car_company_porsche  car_company_renault  car_company_saab  \
122                    0                    0                 0
125                    1                    0                 0
166                    0                    0                 0
1                      0                    0                 0
199                    0                    0                 0

     car_company_subaru  car_company_toyota  car_company_volkswagen  \
122                   0                   0                       0
```

```
125                    0                   0                        0
166                    0                   1                        0
1                      0                   0                        0
199                    0                   0                        0

     car_company_volvo
122                  0
125                  0
166                  0
1                    0
199                  1

[5 rows x 60 columns]
```

[31]:
```python
y_train = df_train.pop('price')
```

[32]:
```python
X_train = df_train
```

6 Model(Liniar Regression)

[33]:
```python
from sklearn.linear_model import LinearRegression
```

[34]:
```python
lm = LinearRegression()
lm.fit(X_train, y_train)
```

[34]:
```
LinearRegression()
```

[35]:
```python
lm.score(X_train, y_train)
```

[35]:
```
0.9712064047413826
```

[36]:
```python
lm.coef_
```

[36]:
```
array([-6.88567127e+01,  1.59791915e+02,  1.67620993e+03, -9.40501426e+02,
        1.66728468e+03, -1.32048230e+03,  2.11588268e+03, -2.24692316e+03,
        7.76550622e+03, -2.46121683e+03, -8.48196473e+02, -3.46772929e+03,
       -9.97487191e+02,  1.47072292e+03,  4.74859336e+02,  6.18660770e+02,
       -5.55423212e+03,  3.05402080e+03, -4.42195370e+03, -4.93031210e+03,
       -4.14820586e+03, -3.45584500e+03, -4.90769833e+02,  3.55723578e+02,
        7.66348624e+03,  7.02233939e+03,  7.73848811e+03,  2.20125928e+03,
        4.74919992e+03,  6.31832027e+01,  8.71702434e+03,  9.33222641e+02,
       -2.37609763e+03,  5.55423212e+03, -1.81898940e-12, -3.11421930e+02,
       -4.55560035e+02,  5.45696821e-12, -9.98396594e+02,  7.91156079e+03,
        8.74512513e+02, -4.59870841e+03, -5.59057700e+03, -3.90189453e+03,
       -2.38532312e+03, -1.93079078e+03, -1.22727036e+03,  0.00000000e+00,
       -6.29519219e+03, -1.93401393e+03, -1.03724527e+04, -5.55044819e+03,
        6.18313769e+03, -2.57784497e+03,  5.48694996e+03, -2.91428633e+03,
```

```
        -1.38182873e+03, -1.37415291e+03,  1.45115753e+02])
```

[37]:
```python
lm.intercept_
```

[37]:
```
20603.961509963374
```

[38]:
```python
from sklearn.feature_selection import RFE
```

[39]:
```python
lm = LinearRegression()
rfe1 = RFE(lm, 15)
rfe1.fit(X_train, y_train)
```

[39]:
```
RFE(estimator=LinearRegression(), n_features_to_select=15)
```

[40]:
```python
rfe1.ranking_
```

[40]:
```
array([41, 39, 28, 31,  1, 26, 17, 32,  1, 18, 33,  1, 23, 24, 37, 30,  1,
       16, 14, 12, 13, 15, 35, 40,  1, 22,  1, 20,  1, 42,  1, 36, 10,  1,
       43, 27, 19, 44,  9,  1, 34,  8,  6,  4, 25, 29,  1, 45,  1,  2,  1,
        7, 11,  1, 21,  1,  3,  5, 38])
```

[41]:
```python
rfe1.support_
```

[41]:
```
array([False, False, False, False,  True, False, False, False,  True,
       False, False,  True, False, False, False, False,  True, False,
       False, False, False, False, False, False,  True, False,  True,
       False,  True, False,  True, False, False,  True, False, False,
       False, False, False,  True, False, False, False, False, False,
       False,  True, False,  True, False,  True, False, False,  True,
       False,  True, False, False, False])
```

[42]:
```python
import statsmodels.api as sm
```

[43]:
```python
col1 = X_train.columns[rfe1.support_]
```

[44]:
```python
X_train_rfe1 = X_train[col1]
```

[45]:
```python
X_train_rfe1.head()
```

[45]:
```
     carwidth  enginesize  compressionratio  fueltype_gas  \
122 -0.924500   -0.660242         -0.172569             1
125  1.114978    0.637806         -0.146125             1
166 -0.833856   -0.660242         -0.172569             1
1   -0.788535    0.123485         -0.278345             1
199  0.616439    0.123485         -0.675002             1

     enginelocation_rear  enginetype_l  enginetype_ohcf  enginetype_rotor  \
122                    0             0                0                 0
```

```
125                      0             0                0                 0
166                      0             0                0                 0
1                        0             0                0                 0
199                      0             0                0                 0

     fuelsystem_idi  car_company_bmw  car_company_mazda  \
122               0                0                  0
125               0                0                  0
166               0                0                  0
1                 0                0                  0
199               0                0                  0

     car_company_mitsubishi  car_company_peugeot  car_company_renault  \
122                       0                    0                    0
125                       0                    0                    0
166                       0                    0                    0
1                         0                    0                    0
199                       0                    0                    0

     car_company_subaru
122                   0
125                   0
166                   0
1                     0
199                   0
```

[46]:

```python
# new columns const added
X_train_rfe1 = sm.add_constant(X_train_rfe1)
```

[47]:

```python
X_train_rfe1.head()
```

[47]:

```
     const  carwidth  enginesize  compressionratio  fueltype_gas  \
122    1.0 -0.924500   -0.660242         -0.172569             1
125    1.0  1.114978    0.637806         -0.146125             1
166    1.0 -0.833856   -0.660242         -0.172569             1
1      1.0 -0.788535    0.123485         -0.278345             1
199    1.0  0.616439    0.123485         -0.675002             1

     enginelocation_rear  enginetype_l  enginetype_ohcf  enginetype_rotor  \
122                    0             0                0                 0
125                    0             0                0                 0
166                    0             0                0                 0
1                      0             0                0                 0
199                    0             0                0                 0

     fuelsystem_idi  car_company_bmw  car_company_mazda  \
122               0                0                  0
```

```
125                0                 0                  0
166                0                 0                  0
1                  0                 0                  0
199                0                 0                  0

     car_company_mitsubishi  car_company_peugeot  car_company_renault  \
122                       0                    0                    0
125                       0                    0                    0
166                       0                    0                    0
1                         0                    0                    0
199                       0                    0                    0

     car_company_subaru
122                   0
125                   0
166                   0
1                     0
199                   0
```

[48]:
```
lm1 = sm.OLS(y_train, X_train_rfe1).fit()
```

[49]:
```
lm1.summary()
```

[49]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.920
Model:                            OLS   Adj. R-squared:                  0.912
Method:                 Least Squares   F-statistic:                     114.1
Date:                Sat, 12 Mar 2022   Prob (F-statistic):           4.59e-64
Time:                        20:09:00   Log-Likelihood:                -1303.5
No. Observations:                 143   AIC:                             2635.
Df Residuals:                     129   BIC:                             2676.
Df Model:                          13
Covariance Type:            nonrobust
================================================================================
==========
                             coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
----------
const                   1.242e+04   1263.541      9.829      0.000    9919.615
1.49e+04
carwidth                3348.1798    356.058      9.403      0.000    2643.710
4052.649
enginesize              3726.7427    351.763     10.594      0.000    3030.772
```

```
4422.714
compressionratio       -3504.8498   1260.300     -2.781      0.006    -5998.385
-1011.314
fueltype_gas            -525.2133   1616.959     -0.325      0.746    -3724.406
2673.979
enginelocation_rear     1.159e+04   1638.588      7.071      0.000     8345.074
1.48e+04
enginetype_l            6988.6523   2392.209      2.921      0.004     2255.608
1.17e+04
enginetype_ohcf         5007.1933    847.950      5.905      0.000     3329.504
6684.883
enginetype_rotor        7418.4990   1461.610      5.076      0.000     4526.667
1.03e+04
fuelsystem_idi          1.294e+04   2869.216      4.512      0.000     7267.961
1.86e+04
car_company_bmw         8307.1217   1020.468      8.141      0.000     6288.100
1.03e+04
car_company_mazda      -1897.4175    825.214     -2.299      0.023    -3530.123
-264.712
car_company_mitsubishi -3077.6988    875.141     -3.517      0.001    -4809.186
-1346.211
car_company_peugeot    -1.097e+04   2629.810     -4.171      0.000    -1.62e+04
-5764.481
car_company_renault    -5273.2963   1661.724     -3.173      0.002    -8561.058
-1985.535
car_company_subaru     -6579.8676    951.545     -6.915      0.000    -8462.522
-4697.213
==============================================================================
Omnibus:                        8.408   Durbin-Watson:                   1.997
Prob(Omnibus):                  0.015   Jarque-Bera (JB):                9.825
Skew:                           0.399   Prob(JB):                      0.00735
Kurtosis:                       4.005   Cond. No.                     3.60e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The smallest eigenvalue is 2.23e-31. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
"""
```

[50]:
```
vif=pd.DataFrame()
```

[51]:
```
vif['Features'] = X_train_rfe1.columns
vif
```

[51]:

	Features
0	const
1	carwidth
2	enginesize
3	compressionratio
4	fueltype_gas
5	enginelocation_rear
6	enginetype_l
7	enginetype_ohcf
8	enginetype_rotor
9	fuelsystem_idi
10	car_company_bmw
11	car_company_mazda
12	car_company_mitsubishi
13	car_company_peugeot
14	car_company_renault
15	car_company_subaru

[52]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

[53]:
```
vif['VIF'] = [variance_inflation_factor(X_train_rfe1.values, i) for i in␣
↪range(X_train_rfe1.shape[1])]
```

[54]:
```
vif['VIF'] = round(vif['VIF'], 2)
```

[55]:
```
vif.sort_values(by='VIF', ascending=False)
```

[55]:

	Features	VIF
4	fueltype_gas	inf
5	enginelocation_rear	inf
7	enginetype_ohcf	inf
9	fuelsystem_idi	inf
15	car_company_subaru	inf
3	compressionratio	42.32
13	car_company_peugeot	9.73
6	enginetype_l	8.99
1	carwidth	3.38
2	enginesize	3.30
8	enginetype_rotor	1.55
11	car_company_mazda	1.50
12	car_company_mitsubishi	1.20
10	car_company_bmw	1.12
14	car_company_renault	1.01
0	const	0.00

[56]:
```
lm = LinearRegression()
rfe2 = RFE(lm, 10)
```

```
rfe2.fit(X_train, y_train)
```

[56]: RFE(estimator=LinearRegression(), n_features_to_select=10)

[57]:
```
rfe2.ranking_
```

[57]:
```
array([46, 44, 33, 36,  1, 31, 22, 37,  1, 23, 38,  4, 28, 29, 42, 35,  3,
       21, 19, 17, 18, 20, 40, 45,  1, 27,  1, 25,  1, 47,  1, 41, 15,  2,
       48, 32, 24, 49, 14,  1, 39, 13, 11,  9, 30, 34,  6, 50,  5,  7,  1,
       12, 16,  1, 26,  1,  8, 10, 43])
```

[58]:
```
col2 = X_train.columns[rfe2.support_]
X_train_rfe2 = X_train[col2]
X_train_rfe2 = sm.add_constant(X_train_rfe2)
lm2 = sm.OLS(y_train, X_train_rfe2).fit()
print(lm2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.907
Model:                            OLS   Adj. R-squared:                  0.901
Method:                 Least Squares   F-statistic:                     144.3
Date:                Sat, 12 Mar 2022   Prob (F-statistic):           3.98e-64
Time:                        20:09:02   Log-Likelihood:                -1314.2
No. Observations:                 143   AIC:                             2648.
Df Residuals:                     133   BIC:                             2678.
Df Model:                           9
Covariance Type:            nonrobust
================================================================================
=======
                          coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-------
const                1.266e+04    235.167     53.845      0.000    1.22e+04
1.31e+04
carwidth             3586.8272    365.982      9.801      0.000    2862.929
4310.725
enginesize           3739.2569    369.354     10.124      0.000    3008.690
4469.824
enginelocation_rear  1.131e+04   1735.038      6.519      0.000    7878.367
1.47e+04
enginetype_l         7351.4558   2533.602      2.902      0.004    2340.089
1.24e+04
enginetype_ohcf      5097.7417    897.846      5.678      0.000    3321.837
6873.646
enginetype_rotor     5388.9829   1337.401      4.029      0.000    2743.655
8034.311
```

```
car_company_bmw      8749.4458   1071.995      8.162      0.000    6629.081
1.09e+04
car_company_peugeot -9788.1466   2757.167     -3.550      0.001   -1.52e+04
-4334.578
car_company_renault -4866.7944   1757.035     -2.770      0.006   -8342.141
-1391.448
car_company_subaru  -6212.4633   1003.232     -6.192      0.000   -8196.816
-4228.110
==============================================================================
Omnibus:                        5.615   Durbin-Watson:                   1.942
Prob(Omnibus):                  0.060   Jarque-Bera (JB):                5.456
Skew:                           0.349   Prob(JB):                       0.0654
Kurtosis:                       3.655   Cond. No.                     2.00e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The smallest eigenvalue is 6.3e-31. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

[59]:
```python
vif=pd.DataFrame()
vif['Features'] = X_train_rfe2.columns
vif['VIF'] = [variance_inflation_factor(X_train_rfe2.values, i) for i in␣
 ↪range(X_train_rfe2.shape[1])]
vif.sort_values(by='VIF', ascending=False)
# showing vif
```

[59]:

	Features	VIF
3	enginelocation_rear	inf
5	enginetype_ohcf	inf
10	car_company_subaru	inf
8	car_company_peugeot	9.494513
4	enginetype_l	8.952555
2	enginesize	3.226114
1	carwidth	3.167483
0	const	1.307825
6	enginetype_rotor	1.150062
7	car_company_bmw	1.092396
9	car_company_renault	1.006775

[60]:
```python
X_train_rfe2.drop('car_company_subaru', axis=1, inplace=True)
```

[61]:
```python
X_train_rfe2 = sm.add_constant(X_train_rfe2)
lm2 = sm.OLS(y_train, X_train_rfe2).fit()
print(lm2.summary())
```

```
                            OLS Regression Results
```

```
==============================================================================
Dep. Variable:                  price   R-squared:                       0.907
Model:                            OLS   Adj. R-squared:                  0.901
Method:                 Least Squares   F-statistic:                     144.3
Date:                Sat, 12 Mar 2022   Prob (F-statistic):           3.98e-64
Time:                        20:09:03   Log-Likelihood:                -1314.2
No. Observations:                 143   AIC:                             2648.
Df Residuals:                     133   BIC:                             2678.
Df Model:                           9
Covariance Type:            nonrobust
================================================================================
=======
                          coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-------
const                1.266e+04    235.167     53.845      0.000    1.22e+04
1.31e+04
carwidth             3586.8272    365.982      9.801      0.000    2862.929
4310.725
enginesize           3739.2569    369.354     10.124      0.000    3008.690
4469.824
enginelocation_rear  1.752e+04   2688.407      6.518      0.000    1.22e+04
2.28e+04
enginetype_l         7351.4558   2533.602      2.902      0.004    2340.089
1.24e+04
enginetype_ohcf     -1114.7216    784.120     -1.422      0.157   -2665.681
436.238
enginetype_rotor     5388.9829   1337.401      4.029      0.000    2743.655
8034.311
car_company_bmw      8749.4458   1071.995      8.162      0.000    6629.081
1.09e+04
car_company_peugeot -9788.1466   2757.167     -3.550      0.001   -1.52e+04
-4334.578
car_company_renault -4866.7944   1757.035     -2.770      0.006   -8342.141
-1391.448
==============================================================================
Omnibus:                        5.615   Durbin-Watson:                   1.942
Prob(Omnibus):                  0.060   Jarque-Bera (JB):                5.456
Skew:                           0.349   Prob(JB):                       0.0654
Kurtosis:                       3.655   Cond. No.                         23.8
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[62]:
```
vif=pd.DataFrame()
vif['Features'] = X_train_rfe2.columns
vif['VIF'] = [variance_inflation_factor(X_train_rfe2.values, i) for i in␣
↪range(X_train_rfe2.shape[1])]
vif.sort_values(by='VIF', ascending=False)
# showing vif
```

[62]:

	Features	VIF
8	car_company_peugeot	9.494513
4	enginetype_l	8.952555
2	enginesize	3.226114
1	carwidth	3.167483
0	const	1.307825
3	enginelocation_rear	1.186865
6	enginetype_rotor	1.150062
5	enginetype_ohcf	1.117740
7	car_company_bmw	1.092396
9	car_company_renault	1.006775

[63]:
```
X_train_rfe2.drop('enginetype_ohcf', axis=1, inplace=True)
```

[64]:
```
X_train_rfe2 = sm.add_constant(X_train_rfe2)
lm2 = sm.OLS(y_train, X_train_rfe2).fit()
print(lm2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.906
Model:                            OLS   Adj. R-squared:                  0.900
Method:                 Least Squares   F-statistic:                     160.8
Date:                Sat, 12 Mar 2022   Prob (F-statistic):           8.22e-65
Time:                        20:09:03   Log-Likelihood:                -1315.3
No. Observations:                 143   AIC:                             2649.
Df Residuals:                     134   BIC:                             2675.
Df Model:                           8
Covariance Type:            nonrobust
================================================================================
=======
                          coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-------
const                1.256e+04    225.167     55.790      0.000    1.21e+04
1.3e+04
carwidth             3575.4156    367.285      9.735      0.000    2848.989
4301.842
enginesize           3788.6562    369.114     10.264      0.000    3058.614
4518.699
```

```
enginelocation_rear  1.642e+04   2583.971      6.355      0.000     1.13e+04
2.15e+04
enginetype_l         7500.5753   2541.056      2.952      0.004     2474.810
1.25e+04
enginetype_rotor     5552.1175   1337.536      4.151      0.000     2906.704
8197.531
car_company_bmw      8800.1077   1075.476      8.183      0.000     6673.003
1.09e+04
car_company_peugeot -9839.1754   2767.416     -3.555      0.001    -1.53e+04
-4365.708
car_company_renault -4771.2505   1762.425     -2.707      0.008    -8257.020
-1285.481
==============================================================================
Omnibus:                        5.533   Durbin-Watson:                   1.944
Prob(Omnibus):                  0.063   Jarque-Bera (JB):                5.168
Skew:                           0.374   Prob(JB):                       0.0755
Kurtosis:                       3.555   Cond. No.                         23.8
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[65]:
```python
vif=pd.DataFrame()
vif['Features'] = X_train_rfe2.columns
vif['VIF'] = [variance_inflation_factor(X_train_rfe2.values, i) for i in␣
↪range(X_train_rfe2.shape[1])]
vif.sort_values(by='VIF', ascending=False)
# showing vif
```

[65]:

	Features	VIF
7	car_company_peugeot	9.492904
4	enginetype_l	8.937210
2	enginesize	3.197559
1	carwidth	3.165959
0	const	1.189897
5	enginetype_rotor	1.141595
6	car_company_bmw	1.091189
3	enginelocation_rear	1.088153
8	car_company_renault	1.005302

[66]:
```python
X_train_rfe2.drop('car_company_peugeot', axis=1, inplace=True)
```

[67]:
```python
X_train_rfe2 = sm.add_constant(X_train_rfe2)
lm2 = sm.OLS(y_train, X_train_rfe2).fit()
print(lm2.summary())
```

```
                            OLS Regression Results
```

```
==============================================================================
Dep. Variable:                  price   R-squared:                       0.897
Model:                            OLS   Adj. R-squared:                  0.891
Method:                 Least Squares   F-statistic:                     167.5
Date:                Sat, 12 Mar 2022   Prob (F-statistic):           2.49e-63
Time:                        20:09:04   Log-Likelihood:                -1321.7
No. Observations:                 143   AIC:                             2659.
Df Residuals:                     135   BIC:                             2683.
Df Model:                           7
Covariance Type:            nonrobust
================================================================================
=======
                          coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-------
const                1.254e+04    234.591     53.458      0.000    1.21e+04
1.3e+04
carwidth             3201.2839    366.745      8.729      0.000    2475.975
3926.593
enginesize           3968.7863    381.057     10.415      0.000    3215.172
4722.400
enginelocation_rear  1.599e+04   2690.180      5.946      0.000    1.07e+04
2.13e+04
enginetype_l         -963.1972    926.277     -1.040      0.300   -2795.089
868.694
enginetype_rotor     5781.1882   1392.391      4.152      0.000    3027.467
8534.910
car_company_bmw      8767.2497   1120.844      7.822      0.000    6550.566
1.1e+04
car_company_renault -4660.5447   1836.552     -2.538      0.012   -8292.678
-1028.411
==============================================================================
Omnibus:                       10.615   Durbin-Watson:                   1.972
Prob(Omnibus):                  0.005   Jarque-Bera (JB):               11.115
Skew:                           0.570   Prob(JB):                      0.00386
Kurtosis:                       3.752   Cond. No.                         16.6
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[68]:

```
vif=pd.DataFrame()
vif['Features'] = X_train_rfe2.columns
vif['VIF'] = [variance_inflation_factor(X_train_rfe2.values, i) for i in␣
 ↪range(X_train_rfe2.shape[1])]
```

```
vif.sort_values(by='VIF', ascending=False)
# showing vif
```

[68]:

	Features	VIF
2	enginesize	3.137317
1	carwidth	2.906076
0	const	1.189050
5	enginetype_rotor	1.138946
4	enginetype_l	1.093290
6	car_company_bmw	1.091108
3	enginelocation_rear	1.085818
7	car_company_renault	1.004989

[69]:

```
X_train_rfe2.drop('enginetype_l', axis=1, inplace=True)
```

[70]:

```
X_train_rfe2 = sm.add_constant(X_train_rfe2)
lm2 = sm.OLS(y_train, X_train_rfe2).fit()
print(lm2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.896
Model:                            OLS   Adj. R-squared:                  0.891
Method:                 Least Squares   F-statistic:                     195.2
Date:                Sat, 12 Mar 2022   Prob (F-statistic):           2.92e-64
Time:                        20:09:04   Log-Likelihood:                -1322.3
No. Observations:                 143   AIC:                             2659.
Df Residuals:                     136   BIC:                             2679.
Df Model:                           6
Covariance Type:            nonrobust
================================================================================
=======
                          coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-------
const                1.247e+04    225.791     55.247      0.000     1.2e+04
1.29e+04
carwidth             3094.8228    352.270      8.785      0.000    2398.186
3791.459
enginesize           4048.8588    373.307     10.846      0.000    3310.621
4787.097
enginelocation_rear  1.589e+04   2688.922      5.908      0.000    1.06e+04
2.12e+04
enginetype_rotor     5943.7489   1384.001      4.295      0.000    3206.803
8680.695
car_company_bmw      8786.7032   1121.023      7.838      0.000    6569.813
1.1e+04
```

```
car_company_renault -4573.6566   1835.198     -2.492      0.014    -8202.872
-944.441
==============================================================================
Omnibus:                        7.920   Durbin-Watson:                   1.970
Prob(Omnibus):                  0.019   Jarque-Bera (JB):                7.687
Skew:                           0.497   Prob(JB):                       0.0214
Kurtosis:                       3.549   Cond. No.                         16.6
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[71]:
```python
vif=pd.DataFrame()
vif['Features'] = X_train_rfe2.columns
vif['VIF'] = [variance_inflation_factor(X_train_rfe2.values, i) for i in range(X_train_rfe2.shape[1])]
vif.sort_values(by='VIF', ascending=False)
# showing vif
```

[71]:

	Features	VIF
2	enginesize	3.009203
1	carwidth	2.679605
4	enginetype_rotor	1.124589
0	const	1.100862
5	car_company_bmw	1.090804
3	enginelocation_rear	1.084154
6	car_company_renault	1.002908

7 Prediction

[72]:
```python
df_test[varlist]=sc.transform(df_test[varlist])
```

[73]:
```python
y_test = df_test.pop('price')
```

[78]:
```python
X_test = df_test
```

[79]:
```python
col2
```

[79]:
```
Index(['carwidth', 'enginesize', 'enginelocation_rear', 'enginetype_l',
       'enginetype_ohcf', 'enginetype_rotor', 'car_company_bmw',
       'car_company_peugeot', 'car_company_renault', 'car_company_subaru'],
      dtype='object')
```

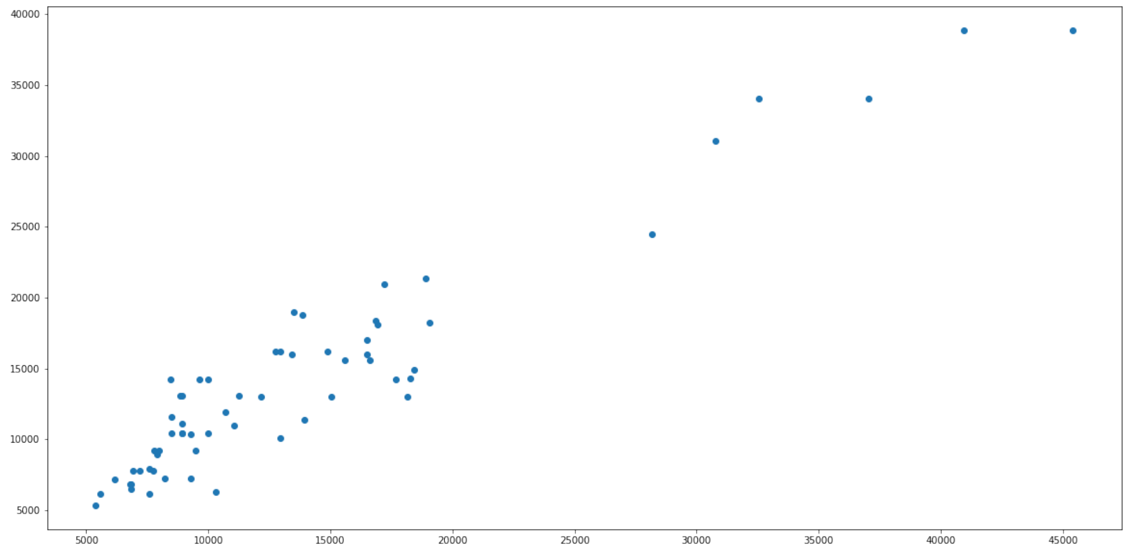
[80]:
```python
X_test_rfe2 = X_test[col2]
```

[81]:
```
X_test_rfe2 = X_test_rfe2.drop(['enginetype_l', 'car_company_peugeot',␣
↪'enginetype_ohcf', 'car_company_subaru'], axis=1)
```

[82]:
```
X_test_rfe2 = sm.add_constant(X_test_rfe2)
```

[83]:
```
y_pred = lm2.predict(X_test_rfe2)
```

[86]:
```
plt.figure(figsize=(20,10))
plt.scatter(y_test, y_pred)
plt.show()
```



[87]:
```
from sklearn.metrics import r2_score
```

[88]:
```
r2_score(y_test, y_pred)
```

[88]: 0.8997211435182687

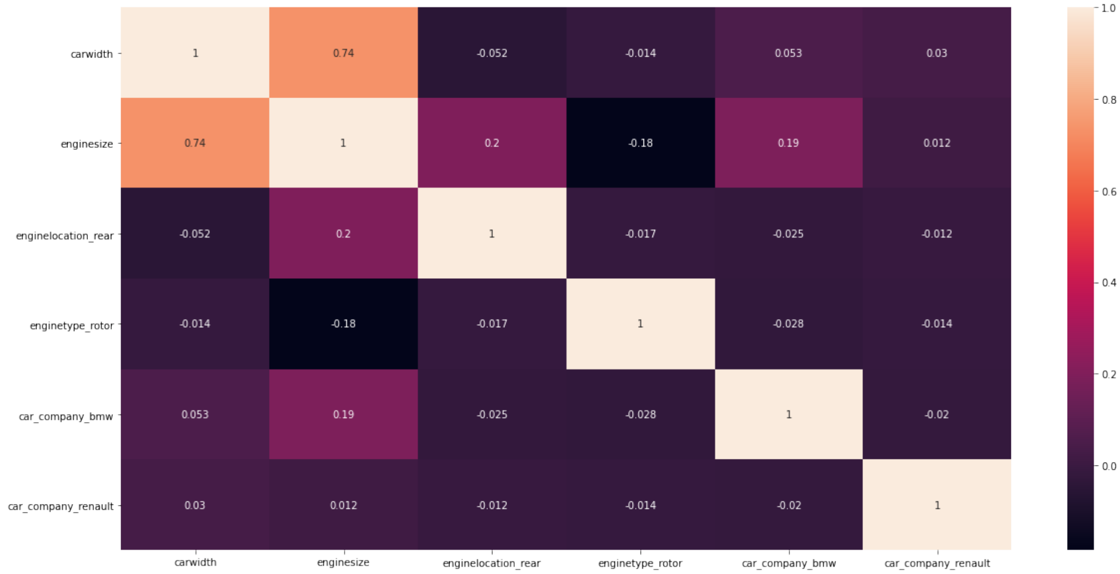
[89]:
```
col2
```

[89]:
```
Index(['carwidth', 'enginesize', 'enginelocation_rear', 'enginetype_l',
       'enginetype_ohcf', 'enginetype_rotor', 'car_company_bmw',
       'car_company_peugeot', 'car_company_renault', 'car_company_subaru'],
      dtype='object')
```

[91]:
```
col2 = col2.drop(['enginetype_l', 'car_company_peugeot', 'enginetype_ohcf',␣
↪'car_company_subaru'])
```

[95]:
```
plt.figure(figsize=(20,10))
sns.heatmap(df[col2].corr(), annot=True)
```

```
plt.show()
```



carwidth
enginesize
enginelocation_rear
enginetype_rotor
car_company_bmw
car_company_renault
1 0.74 -0.052 -0.014 0.053 0.03
0.74 1 0.2 -0.18 0.19 0.012
-0.052 0.2 1 -0.017 -0.025 -0.012
-0.014 -0.18 -0.017 1 -0.028 -0.014
0.053 0.19 -0.025 -0.028 1 -0.02
0.03 0.012 -0.012 -0.014 -0.02 1
1.0
0.8
0.6
0.4
0.2
0.0