machine learning–vijay 2

January 14, 2023

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

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

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

[3]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
0	3	?	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111	21	27	13495
1	3	?	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111	21	27	16500
2	1	?	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	152	154	19	26	16500
3	2	164	audi	gas	sedan	fwd	front	66.2	54.3	ohc	109	102	24	30	13950
4	2	164	audi	gas	sedan	4wd	front	66.4	54.3	ohc	136	115	18	22	17450

[4]:
```
df.info() #checking that any columns dtype is correct or different
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype
```

```
---  ------              --------------  -----
 0   symboling           205 non-null    int64
 1   normalized-losses   205 non-null    object
 2   make                205 non-null    object
 3   fuel-type           205 non-null    object
 4   body-style          205 non-null    object
 5   drive-wheels        205 non-null    object
 6   engine-location     205 non-null    object
 7   width               205 non-null    float64
 8   height              205 non-null    float64
 9   engine-type         205 non-null    object
 10  engine-size         205 non-null    int64
 11  horsepower          205 non-null    object
 12  city-mpg            205 non-null    int64
 13  highway-mpg         205 non-null    int64
 14  price               205 non-null    int64
dtypes: float64(2), int64(5), object(8)
memory usage: 24.1+ KB
```

1 to know what are the values in a column

[5]:
```
df['normalized-losses'].value_counts() #to see 'normalized-losses' column␣
 ↪values because its dtype is object
```

[5]:
```
?      41
161    11
91      8
150     7
134     6
128     6
104     6
85      5
94      5
65      5
102     5
74      5
168     5
103     5
95      5
106     4
93      4
118     4
148     4
122     4
83      3
125     3
154     3
```

```
115     3
137     3
101     3
119     2
87      2
89      2
192     2
197     2
158     2
81      2
188     2
194     2
153     2
129     2
108     2
110     2
164     2
145     2
113     2
256     1
107     1
90      1
231     1
142     1
121     1
78      1
98      1
186     1
77      1
Name: normalized-losses, dtype: int64
```

2 to replace ? into nan

[6]:
```
df['normalized-losses'].replace('?',np.nan,inplace=True) #replace '?' with np.
↪nan,give inplace=True for save
```

[7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   symboling          205 non-null    int64
 1   normalized-losses  164 non-null    object
 2   make               205 non-null    object
 3   fuel-type          205 non-null    object
```

```
 4   body-style         205 non-null    object
 5   drive-wheels       205 non-null    object
 6   engine-location    205 non-null    object
 7   width              205 non-null    float64
 8   height             205 non-null    float64
 9   engine-type        205 non-null    object
 10  engine-size        205 non-null    int64
 11  horsepower         205 non-null    object
 12  city-mpg           205 non-null    int64
 13  highway-mpg        205 non-null    int64
 14  price              205 non-null    int64
dtypes: float64(2), int64(5), object(8)
memory usage: 24.1+ KB
```

3 to change datatype (object into int or float)

[8]:
```
df['normalized-losses']=df['normalized-losses'].astype('float64') #changing␣
↪dtype from 'object' to 'float64'
```

[9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   symboling          205 non-null    int64
 1   normalized-losses  164 non-null    float64
 2   make               205 non-null    object
 3   fuel-type          205 non-null    object
 4   body-style         205 non-null    object
 5   drive-wheels       205 non-null    object
 6   engine-location    205 non-null    object
 7   width              205 non-null    float64
 8   height             205 non-null    float64
 9   engine-type        205 non-null    object
 10  engine-size        205 non-null    int64
 11  horsepower         205 non-null    object
 12  city-mpg           205 non-null    int64
 13  highway-mpg        205 non-null    int64
 14  price              205 non-null    int64
dtypes: float64(3), int64(5), object(7)
memory usage: 24.1+ KB
```

4 dropna/fillna

[10]:
```
nmean=df['normalized-losses'].mean()
```

[11]:
```
nmean
```

[11]:
```
122.0
```

[12]:
```
df['normalized-losses'].fillna(nmean)
```

[12]:
```
0      122.0
1      122.0
2      122.0
3      164.0
4      164.0
       …
200     95.0
201     95.0
202     95.0
203     95.0
204     95.0
Name: normalized-losses, Length: 205, dtype: float64
```

[13]:
```
df['normalized-losses'].dropna()
```

[13]:
```
3      164.0
4      164.0
6      158.0
8      158.0
10     192.0
       …
200     95.0
201     95.0
202     95.0
203     95.0
204     95.0
Name: normalized-losses, Length: 164, dtype: float64
```

5 filling using simple imputer

[14]:
```
df['normalized-losses']
```

[14]:
```
0        NaN
1        NaN
2        NaN
3      164.0
4      164.0
```

```
      ...
200    95.0
201    95.0
202    95.0
203    95.0
204    95.0
Name: normalized-losses, Length: 205, dtype: float64
```

[15]:
```
from sklearn.impute import SimpleImputer
```

[16]:
```
si=SimpleImputer(missing_values=np.nan,strategy='mean')
```

[17]:
```
df[['normalized-losses']]=si.fit_transform(df[['normalized-losses']]) #give␣
↪mean value to 'normalized-losses' column.dont have inplace so assign
```

[18]:
```
df[['normalized-losses']].value_counts()
```

[18]:
```
normalized-losses
122.0               45
161.0               11
91.0                 8
150.0                7
128.0                6
104.0                6
134.0                6
95.0                 5
94.0                 5
74.0                 5
65.0                 5
103.0                5
85.0                 5
168.0                5
102.0                5
148.0                4
106.0                4
118.0                4
93.0                 4
101.0                3
154.0                3
115.0                3
83.0                 3
125.0                3
137.0                3
87.0                 2
188.0                2
158.0                2
153.0                2
```

```
81.0                  2
145.0                 2
192.0                 2
89.0                  2
129.0                 2
194.0                 2
197.0                 2
119.0                 2
113.0                 2
110.0                 2
108.0                 2
164.0                 2
186.0                 1
231.0                 1
142.0                 1
77.0                  1
78.0                  1
98.0                  1
90.0                  1
121.0                 1
107.0                 1
256.0                 1
dtype: int64
```

[19]:
```
df[['horsepower']].value_counts()
```

[19]:
```
horsepower
68            19
70            11
69            10
116            9
110            8
95             7
62             6
114            6
101            6
160            6
88             6
145            5
84             5
82             5
76             5
97             5
102            5
123            4
86             4
111            4
```

```
92              4
85              3
182             3
207             3
73              3
152             3
90              3
121             3
52              2
56              2
94              2
100             2
?               2
156             2
112             2
184             2
176             2
162             2
161             2
155             2
154             1
106             1
115             1
120             1
134             1
135             1
140             1
142             1
143             1
288             1
78              1
48              1
72              1
175             1
64              1
60              1
58              1
55              1
262             1
200             1
dtype: int64
```

[20]:
```
df['horsepower'].replace('?',np.nan,inplace=True)
```

[21]:
```
df['horsepower'].value_counts()
```

[21]:

68	19
70	11
69	10
116	9
110	8
95	7
114	6
160	6
101	6
62	6
88	6
145	5
76	5
97	5
82	5
84	5
102	5
92	4
111	4
123	4
86	4
207	3
182	3
90	3
121	3
152	3
85	3
73	3
161	2
94	2
56	2
112	2
184	2
155	2
156	2
52	2
100	2
162	2
176	2
140	1
115	1
134	1
78	1
48	1
288	1
143	1
142	1

```
200     1
58      1
55      1
60      1
175     1
154     1
72      1
120     1
64      1
135     1
262     1
106     1
Name: horsepower, dtype: int64
```

[22]:
```
df['horsepower']=df['horsepower'].astype('float64') #don't have inplace so␣
↪assign variable
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   symboling          205 non-null    int64
 1   normalized-losses  205 non-null    float64
 2   make               205 non-null    object
 3   fuel-type          205 non-null    object
 4   body-style         205 non-null    object
 5   drive-wheels       205 non-null    object
 6   engine-location    205 non-null    object
 7   width              205 non-null    float64
 8   height             205 non-null    float64
 9   engine-type        205 non-null    object
 10  engine-size        205 non-null    int64
 11  horsepower         203 non-null    float64
 12  city-mpg           205 non-null    int64
 13  highway-mpg        205 non-null    int64
 14  price              205 non-null    int64
dtypes: float64(4), int64(5), object(6)
memory usage: 24.1+ KB
```

[24]:
```
df['horsepower'].value_counts()
```

[24]:
```
68.0     19
70.0     11
69.0     10
116.0     9
```

110.0	8
95.0	7
114.0	6
160.0	6
101.0	6
62.0	6
88.0	6
145.0	5
76.0	5
97.0	5
82.0	5
84.0	5
102.0	5
92.0	4
111.0	4
123.0	4
86.0	4
207.0	3
182.0	3
90.0	3
121.0	3
152.0	3
85.0	3
73.0	3
161.0	2
94.0	2
56.0	2
112.0	2
184.0	2
155.0	2
156.0	2
52.0	2
100.0	2
162.0	2
176.0	2
140.0	1
115.0	1
134.0	1
78.0	1
48.0	1
288.0	1
143.0	1
142.0	1
200.0	1
58.0	1
55.0	1
60.0	1

```
175.0     1
154.0     1
72.0      1
120.0     1
64.0      1
135.0     1
262.0     1
106.0     1
Name: horsepower, dtype: int64
```

[25]:
```
df[['horsepower']]=si.fit_transform(df[['horsepower']])
```

[26]:
```
df['horsepower'].value_counts()
```

[26]:
```
68.000000     19
70.000000     11
69.000000     10
116.000000     9
110.000000     8
95.000000      7
88.000000      6
62.000000      6
101.000000     6
160.000000     6
114.000000     6
84.000000      5
97.000000      5
102.000000     5
145.000000     5
82.000000      5
76.000000      5
111.000000     4
92.000000      4
123.000000     4
86.000000      4
90.000000      3
73.000000      3
85.000000      3
207.000000     3
182.000000     3
121.000000     3
152.000000     3
112.000000     2
56.000000      2
161.000000     2
156.000000     2
94.000000      2
```

```
52.000000      2
104.256158     2
162.000000     2
155.000000     2
184.000000     2
100.000000     2
176.000000     2
55.000000      1
262.000000     1
134.000000     1
115.000000     1
140.000000     1
48.000000      1
58.000000      1
60.000000      1
78.000000      1
135.000000     1
200.000000     1
64.000000      1
120.000000     1
72.000000      1
154.000000     1
288.000000     1
143.000000     1
142.000000     1
175.000000     1
106.000000     1
Name: horsepower, dtype: int64
```

[ ]:

6 handling missing value—practise

[27]:
```
cf=pd.read_csv('cars.csv')
```

[28]:
```
cf
```

[28]:
```
     symboling normalized-losses         make fuel-type   body-style  \
0            3                 ?  alfa-romero       gas  convertible
1            3                 ?  alfa-romero       gas  convertible
2            1                 ?  alfa-romero       gas    hatchback
3            2               164         audi       gas        sedan
4            2               164         audi       gas        sedan
..         …                 …           …        …          …
200         -1                95        volvo       gas        sedan
201         -1                95        volvo       gas        sedan
```

```
202         -1                 95        volvo       gas        sedan
203         -1                 95        volvo    diesel        sedan
204         -1                 95        volvo       gas        sedan

    drive-wheels engine-location  width  height engine-type  engine-size  \
0            rwd           front   64.1    48.8        dohc          130
1            rwd           front   64.1    48.8        dohc          130
2            rwd           front   65.5    52.4        ohcv          152
3            fwd           front   66.2    54.3         ohc          109
4            4wd           front   66.4    54.3         ohc          136
..           …             …    …     …         …           …
200          rwd           front   68.9    55.5         ohc          141
201          rwd           front   68.8    55.5         ohc          141
202          rwd           front   68.9    55.5        ohcv          173
203          rwd           front   68.9    55.5         ohc          145
204          rwd           front   68.9    55.5         ohc          141

    horsepower  city-mpg  highway-mpg  price
0          111        21           27  13495
1          111        21           27  16500
2          154        19           26  16500
3          102        24           30  13950
4          115        18           22  17450
..         …       …           …    …
200        114        23           28  16845
201        160        19           25  19045
202        134        18           23  21485
203        106        26           27  22470
204        114        19           25  22625

[205 rows x 15 columns]
```

[29]:
```
cf.head(20)
```

[29]:
```
    symboling normalized-losses         make fuel-type   body-style  \
0           3                 ?  alfa-romero       gas  convertible
1           3                 ?  alfa-romero       gas  convertible
2           1                 ?  alfa-romero       gas    hatchback
3           2               164         audi       gas        sedan
4           2               164         audi       gas        sedan
5           2                 ?         audi       gas        sedan
6           1               158         audi       gas        sedan
7           1                 ?         audi       gas        wagon
8           1               158         audi       gas        sedan
9           0                 ?         audi       gas    hatchback
10          2               192          bmw       gas        sedan
11          0               192          bmw       gas        sedan
```

```
12          0                188          bmw       gas        sedan
13          0                188          bmw       gas        sedan
14          1                  ?          bmw       gas        sedan
15          0                  ?          bmw       gas        sedan
16          0                  ?          bmw       gas        sedan
17          0                  ?          bmw       gas        sedan
18          2                121    chevrolet       gas    hatchback
19          1                 98    chevrolet       gas    hatchback

   drive-wheels engine-location  width  height engine-type  engine-size  \
0           rwd           front   64.1    48.8        dohc          130
1           rwd           front   64.1    48.8        dohc          130
2           rwd           front   65.5    52.4        ohcv          152
3           fwd           front   66.2    54.3         ohc          109
4           4wd           front   66.4    54.3         ohc          136
5           fwd           front   66.3    53.1         ohc          136
6           fwd           front   71.4    55.7         ohc          136
7           fwd           front   71.4    55.7         ohc          136
8           fwd           front   71.4    55.9         ohc          131
9           4wd           front   67.9    52.0         ohc          131
10          rwd           front   64.8    54.3         ohc          108
11          rwd           front   64.8    54.3         ohc          108
12          rwd           front   64.8    54.3         ohc          164
13          rwd           front   64.8    54.3         ohc          164
14          rwd           front   66.9    55.7         ohc          164
15          rwd           front   66.9    55.7         ohc          209
16          rwd           front   67.9    53.7         ohc          209
17          rwd           front   70.9    56.3         ohc          209
18          fwd           front   60.3    53.2           l           61
19          fwd           front   63.6    52.0         ohc           90

   horsepower  city-mpg  highway-mpg  price
0         111        21           27  13495
1         111        21           27  16500
2         154        19           26  16500
3         102        24           30  13950
4         115        18           22  17450
5         110        19           25  15250
6         110        19           25  17710
7         110        19           25  18920
8         140        17           20  23875
9         160        16           22  12000
10        101        23           29  16430
11        101        23           29  16925
12        121        21           28  20970
13        121        21           28  21105
14        121        20           25  24565
```

```
15        182        16           22  30760
16        182        16           22  41315
17        182        15           20  36880
18         48        47           53   5151
19         70        38           43   6295
```

[30]:
```
cf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   symboling          205 non-null    int64
 1   normalized-losses  205 non-null    object
 2   make               205 non-null    object
 3   fuel-type          205 non-null    object
 4   body-style         205 non-null    object
 5   drive-wheels       205 non-null    object
 6   engine-location    205 non-null    object
 7   width              205 non-null    float64
 8   height             205 non-null    float64
 9   engine-type        205 non-null    object
 10  engine-size        205 non-null    int64
 11  horsepower         205 non-null    object
 12  city-mpg           205 non-null    int64
 13  highway-mpg        205 non-null    int64
 14  price              205 non-null    int64
dtypes: float64(2), int64(5), object(8)
memory usage: 24.1+ KB
```

[31]:
```
cf['normalized-losses'].value_counts()
```

[31]:
```
?      41
161    11
91      8
150     7
134     6
128     6
104     6
85      5
94      5
65      5
102     5
74      5
168     5
103     5
95      5
```

```
106     4
93      4
118     4
148     4
122     4
83      3
125     3
154     3
115     3
137     3
101     3
119     2
87      2
89      2
192     2
197     2
158     2
81      2
188     2
194     2
153     2
129     2
108     2
110     2
164     2
145     2
113     2
256     1
107     1
90      1
231     1
142     1
121     1
78      1
98      1
186     1
77      1
Name: normalized-losses, dtype: int64
```

[32]:
```
cf['normalized-losses'].replace('?',np.nan,inplace=True)
```

[33]:
```
cf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
```

```
 0   symboling          205 non-null    int64
 1   normalized-losses  164 non-null    object
 2   make               205 non-null    object
 3   fuel-type          205 non-null    object
 4   body-style         205 non-null    object
 5   drive-wheels       205 non-null    object
 6   engine-location    205 non-null    object
 7   width              205 non-null    float64
 8   height             205 non-null    float64
 9   engine-type        205 non-null    object
 10  engine-size        205 non-null    int64
 11  horsepower         205 non-null    object
 12  city-mpg           205 non-null    int64
 13  highway-mpg        205 non-null    int64
 14  price              205 non-null    int64
dtypes: float64(2), int64(5), object(8)
memory usage: 24.1+ KB
```

[34]:
```
cf['normalized-losses']=cf['normalized-losses'].astype('float64')
```

[35]:
```
cf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   symboling          205 non-null    int64
 1   normalized-losses  164 non-null    float64
 2   make               205 non-null    object
 3   fuel-type          205 non-null    object
 4   body-style         205 non-null    object
 5   drive-wheels       205 non-null    object
 6   engine-location    205 non-null    object
 7   width              205 non-null    float64
 8   height             205 non-null    float64
 9   engine-type        205 non-null    object
 10  engine-size        205 non-null    int64
 11  horsepower         205 non-null    object
 12  city-mpg           205 non-null    int64
 13  highway-mpg        205 non-null    int64
 14  price              205 non-null    int64
dtypes: float64(3), int64(5), object(7)
memory usage: 24.1+ KB
```

[36]:
```
nmedian=cf['normalized-losses'].median()
```

[37]:
```
nmedian
```

[37]:
```
115.0
```

[38]:
```
cf['normalized-losses'].fillna(nmedian)
```

[38]:
```
0      115.0
1      115.0
2      115.0
3      164.0
4      164.0
       …
200     95.0
201     95.0
202     95.0
203     95.0
204     95.0
Name: normalized-losses, Length: 205, dtype: float64
```

[39]:
```
cf['normalized-losses'].dropna()
```

[39]:
```
3      164.0
4      164.0
6      158.0
8      158.0
10     192.0
       …
200     95.0
201     95.0
202     95.0
203     95.0
204     95.0
Name: normalized-losses, Length: 164, dtype: float64
```

[40]:
```
from sklearn.impute import SimpleImputer
```

[41]:
```
si=SimpleImputer(missing_values=np.nan,strategy='median')
cf[['normalized-losses']]=si.fit_transform(cf[['normalized-losses']])
```

[42]:
```
cf['normalized-losses'].value_counts()
```

[42]:
```
115.0    44
161.0    11
91.0      8
150.0     7
134.0     6
128.0     6
104.0     6
85.0      5
94.0      5
```

```
65.0      5
102.0     5
74.0      5
168.0     5
103.0     5
95.0      5
106.0     4
93.0      4
118.0     4
148.0     4
122.0     4
83.0      3
125.0     3
154.0     3
137.0     3
101.0     3
188.0     2
119.0     2
89.0      2
192.0     2
197.0     2
158.0     2
81.0      2
87.0      2
153.0     2
129.0     2
108.0     2
110.0     2
164.0     2
145.0     2
194.0     2
113.0     2
78.0      1
256.0     1
107.0     1
90.0      1
77.0      1
142.0     1
121.0     1
98.0      1
186.0     1
231.0     1
Name: normalized-losses, dtype: int64
```

7 OUTLIER

[43]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   symboling          205 non-null    int64  
 1   normalized-losses  205 non-null    float64
 2   make               205 non-null    object 
 3   fuel-type          205 non-null    object 
 4   body-style         205 non-null    object 
 5   drive-wheels       205 non-null    object 
 6   engine-location    205 non-null    object 
 7   width              205 non-null    float64
 8   height             205 non-null    float64
 9   engine-type        205 non-null    object 
 10  engine-size        205 non-null    int64  
 11  horsepower         205 non-null    float64
 12  city-mpg           205 non-null    int64  
 13  highway-mpg        205 non-null    int64  
 14  price              205 non-null    int64  
dtypes: float64(4), int64(5), object(6)
memory usage: 24.1+ KB
```

8 split the table into input as feature and output as target

[44]:
```
feature=df.iloc[:,:-1]
target=df['price']                          #target=df.iloc[:,-1]
```

[45]:
```
feature
```

[45]:

	symboling	normalized-losses	make	fuel-type	body-style	\
0	3	122.0	alfa-romero	gas	convertible	
1	3	122.0	alfa-romero	gas	convertible	
2	1	122.0	alfa-romero	gas	hatchback	
3	2	164.0	audi	gas	sedan	
4	2	164.0	audi	gas	sedan	
..	…	…	…	…	…	
200	-1	95.0	volvo	gas	sedan	
201	-1	95.0	volvo	gas	sedan	
202	-1	95.0	volvo	gas	sedan	
203	-1	95.0	volvo	diesel	sedan	
204	-1	95.0	volvo	gas	sedan	

```
    drive-wheels engine-location  width  height engine-type  engine-size  \
0            rwd           front   64.1    48.8        dohc          130
1            rwd           front   64.1    48.8        dohc          130
2            rwd           front   65.5    52.4        ohcv          152
3            fwd           front   66.2    54.3         ohc          109
4            4wd           front   66.4    54.3         ohc          136
..           …             …    …     …         …           …
200          rwd           front   68.9    55.5         ohc          141
201          rwd           front   68.8    55.5         ohc          141
202          rwd           front   68.9    55.5        ohcv          173
203          rwd           front   68.9    55.5         ohc          145
204          rwd           front   68.9    55.5         ohc          141

     horsepower  city-mpg  highway-mpg
0         111.0        21           27
1         111.0        21           27
2         154.0        19           26
3         102.0        24           30
4         115.0        18           22
..          …       …          …
200       114.0        23           28
201       160.0        19           25
202       134.0        18           23
203       106.0        26           27
204       114.0        19           25

[205 rows x 14 columns]
```

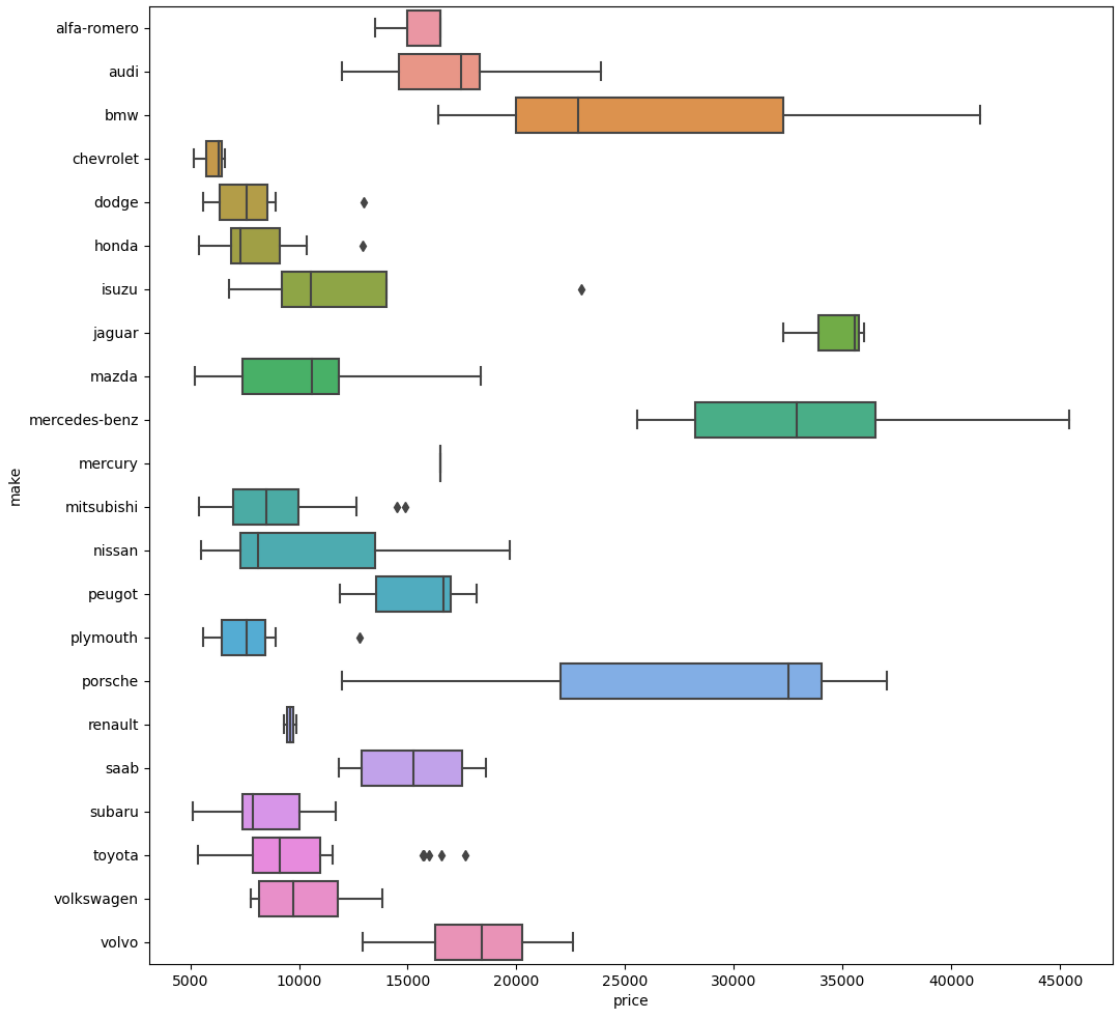
[46]:
```
target
```

[46]:
```
0      13495
1      16500
2      16500
3      13950
4      17450
       …
200    16845
201    19045
202    21485
203    22470
204    22625
Name: price, Length: 205, dtype: int64
```

9 finging outlier for each company(make–column)

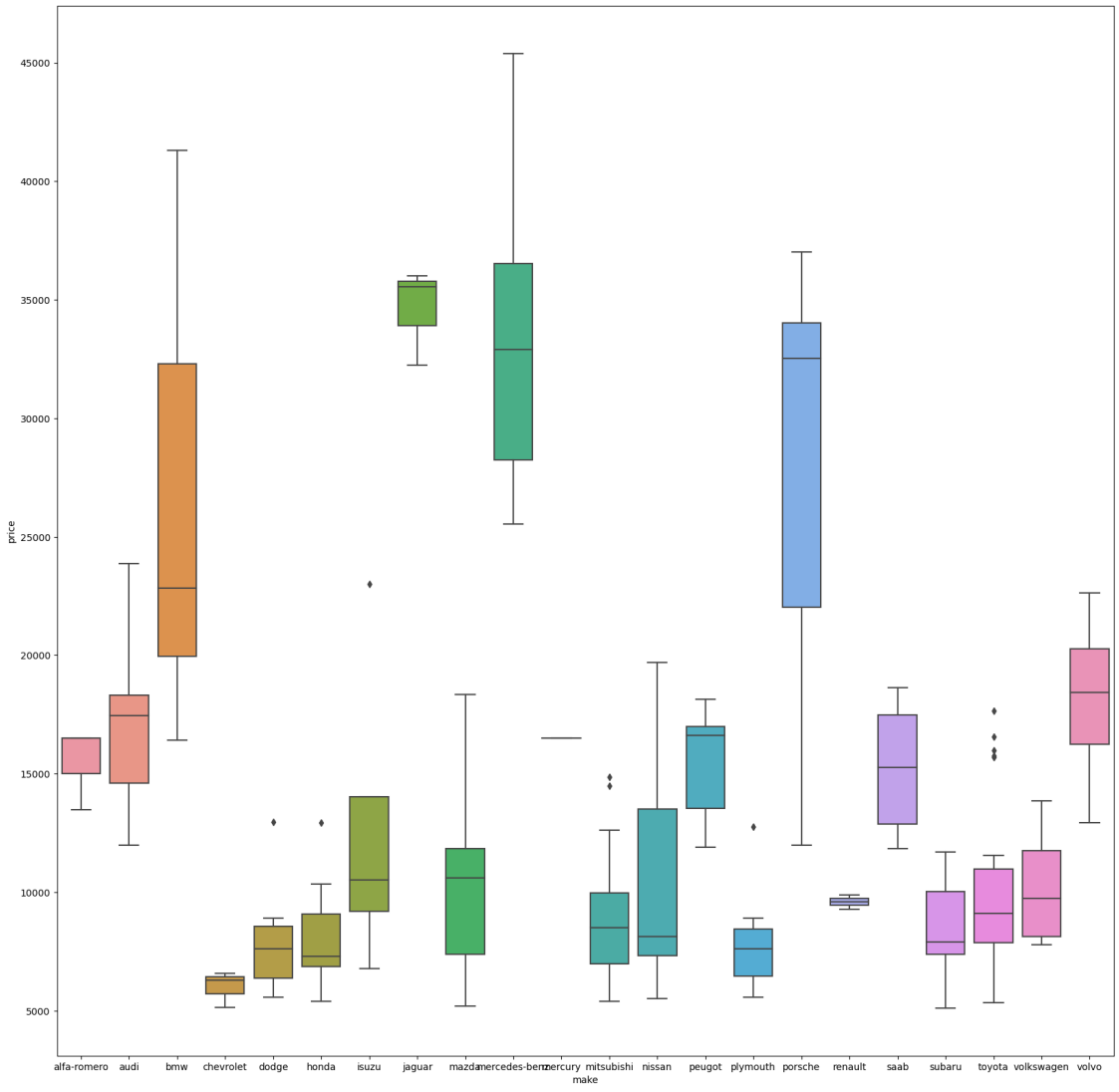
[47]:
```
plt.figure(figsize=(12,12))
sns.boxplot(data=feature,x=target,y='make')
```

[47]: <AxesSubplot:xlabel='price', ylabel='make'>



[48]:
```
plt.figure(figsize=(20,20))
sns.boxplot(data=feature,y=target,x='make')
```

[48]: <AxesSubplot:xlabel='make', ylabel='price'>



10 drop a outlier —— from 'honda' that are greater than 12000

[49]:
```
df.head()
```

[49]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size
0	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130
1	3	122.0	alfa-romero	gas	convertible						
2	1	122.0	alfa-romero	gas	hatchback						
3	2	164.0	audi	gas	sedan						
4	2	164.0	audi	gas	sedan						

```
1          rwd           front   64.1    48.8         dohc          130
2          rwd           front   65.5    52.4         ohcv          152
3          fwd           front   66.2    54.3          ohc          109
4          4wd           front   66.4    54.3          ohc          136

   horsepower  city-mpg  highway-mpg  price
0       111.0        21           27  13495
1       111.0        21           27  16500
2       154.0        19           26  16500
3       102.0        24           30  13950
4       115.0        18           22  17450
```

[50]:
```
df[(df['make']=='honda')&(df['price']>12000)]
```

[50]:
```
    symboling  normalized-losses   make fuel-type body-style drive-wheels  \
41          0               85.0  honda       gas      sedan          fwd

   engine-location  width  height engine-type  engine-size  horsepower  \
41           front   65.2    54.1         ohc          110       101.0

    city-mpg  highway-mpg  price
41        24           28  12945
```

[51]:
```
df.drop(41,axis=0,inplace=True) #drop a 41th row
```

[52]:
```
df.head(50) #41 index was deleted
```

[52]:
```
    symboling  normalized-losses         make fuel-type   body-style  \
0           3              122.0  alfa-romero       gas  convertible
1           3              122.0  alfa-romero       gas  convertible
2           1              122.0  alfa-romero       gas    hatchback
3           2              164.0         audi       gas        sedan
4           2              164.0         audi       gas        sedan
5           2              122.0         audi       gas        sedan
6           1              158.0         audi       gas        sedan
7           1              122.0         audi       gas        wagon
8           1              158.0         audi       gas        sedan
9           0              122.0         audi       gas    hatchback
10          2              192.0          bmw       gas        sedan
11          0              192.0          bmw       gas        sedan
12          0              188.0          bmw       gas        sedan
13          0              188.0          bmw       gas        sedan
14          1              122.0          bmw       gas        sedan
15          0              122.0          bmw       gas        sedan
16          0              122.0          bmw       gas        sedan
17          0              122.0          bmw       gas        sedan
18          2              121.0    chevrolet       gas    hatchback
```

```
19          1               98.0    chevrolet       gas    hatchback
20          0               81.0    chevrolet       gas        sedan
21          1              118.0        dodge       gas    hatchback
22          1              118.0        dodge       gas    hatchback
23          1              118.0        dodge       gas    hatchback
24          1              148.0        dodge       gas    hatchback
25          1              148.0        dodge       gas        sedan
26          1              148.0        dodge       gas        sedan
27          1              148.0        dodge       gas        sedan
28         -1              110.0        dodge       gas        wagon
29          3              145.0        dodge       gas    hatchback
30          2              137.0        honda       gas    hatchback
31          2              137.0        honda       gas    hatchback
32          1              101.0        honda       gas    hatchback
33          1              101.0        honda       gas    hatchback
34          1              101.0        honda       gas    hatchback
35          0              110.0        honda       gas        sedan
36          0               78.0        honda       gas        wagon
37          0              106.0        honda       gas    hatchback
38          0              106.0        honda       gas    hatchback
39          0               85.0        honda       gas        sedan
40          0               85.0        honda       gas        sedan
42          1              107.0        honda       gas        sedan
43          0              122.0        isuzu       gas        sedan
44          1              122.0        isuzu       gas        sedan
45          0              122.0        isuzu       gas        sedan
46          2              122.0        isuzu       gas    hatchback
47          0              145.0       jaguar       gas        sedan
48          0              122.0       jaguar       gas        sedan
49          0              122.0       jaguar       gas        sedan
50          1              104.0        mazda       gas    hatchback

   drive-wheels engine-location  width  height engine-type  engine-size  \
0           rwd           front   64.1    48.8        dohc          130
1           rwd           front   64.1    48.8        dohc          130
2           rwd           front   65.5    52.4        ohcv          152
3           fwd           front   66.2    54.3         ohc          109
4           4wd           front   66.4    54.3         ohc          136
5           fwd           front   66.3    53.1         ohc          136
6           fwd           front   71.4    55.7         ohc          136
7           fwd           front   71.4    55.7         ohc          136
8           fwd           front   71.4    55.9         ohc          131
9           4wd           front   67.9    52.0         ohc          131
10          rwd           front   64.8    54.3         ohc          108
11          rwd           front   64.8    54.3         ohc          108
12          rwd           front   64.8    54.3         ohc          164
13          rwd           front   64.8    54.3         ohc          164
```

14	rwd	front	66.9	55.7	ohc	164
15	rwd	front	66.9	55.7	ohc	209
16	rwd	front	67.9	53.7	ohc	209
17	rwd	front	70.9	56.3	ohc	209
18	fwd	front	60.3	53.2	l	61
19	fwd	front	63.6	52.0	ohc	90
20	fwd	front	63.6	52.0	ohc	90
21	fwd	front	63.8	50.8	ohc	90
22	fwd	front	63.8	50.8	ohc	90
23	fwd	front	63.8	50.8	ohc	98
24	fwd	front	63.8	50.6	ohc	90
25	fwd	front	63.8	50.6	ohc	90
26	fwd	front	63.8	50.6	ohc	90
27	fwd	front	63.8	50.6	ohc	98
28	fwd	front	64.6	59.8	ohc	122
29	fwd	front	66.3	50.2	ohc	156
30	fwd	front	63.9	50.8	ohc	92
31	fwd	front	63.9	50.8	ohc	92
32	fwd	front	64.0	52.6	ohc	79
33	fwd	front	64.0	52.6	ohc	92
34	fwd	front	64.0	52.6	ohc	92
35	fwd	front	64.0	54.5	ohc	92
36	fwd	front	63.9	58.3	ohc	92
37	fwd	front	65.2	53.3	ohc	110
38	fwd	front	65.2	53.3	ohc	110
39	fwd	front	65.2	54.1	ohc	110
40	fwd	front	62.5	54.1	ohc	110
42	fwd	front	66.0	51.0	ohc	110
43	rwd	front	61.8	53.5	ohc	111
44	fwd	front	63.6	52.0	ohc	90
45	fwd	front	63.6	52.0	ohc	90
46	rwd	front	65.2	51.4	ohc	119
47	rwd	front	69.6	52.8	dohc	258
48	rwd	front	69.6	52.8	dohc	258
49	rwd	front	70.6	47.8	ohcv	326
50	fwd	front	64.2	54.1	ohc	91

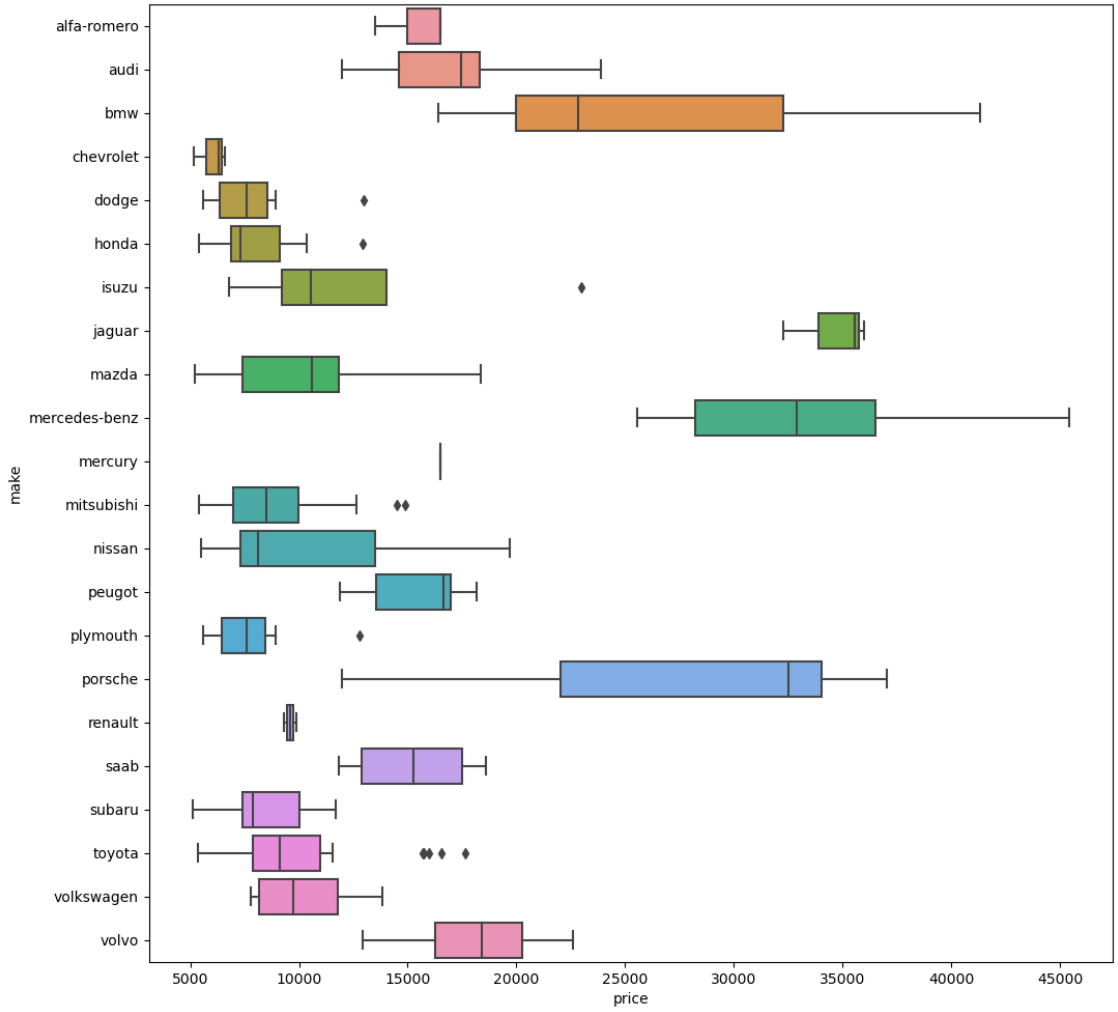
	horsepower	city-mpg	highway-mpg	price
0	111.0	21	27	13495
1	111.0	21	27	16500
2	154.0	19	26	16500
3	102.0	24	30	13950
4	115.0	18	22	17450
5	110.0	19	25	15250
6	110.0	19	25	17710
7	110.0	19	25	18920
8	140.0	17	20	23875

9	160.0	16	22	12000
10	101.0	23	29	16430
11	101.0	23	29	16925
12	121.0	21	28	20970
13	121.0	21	28	21105
14	121.0	20	25	24565
15	182.0	16	22	30760
16	182.0	16	22	41315
17	182.0	15	20	36880
18	48.0	47	53	5151
19	70.0	38	43	6295
20	70.0	38	43	6575
21	68.0	37	41	5572
22	68.0	31	38	6377
23	102.0	24	30	7957
24	68.0	31	38	6229
25	68.0	31	38	6692
26	68.0	31	38	7609
27	102.0	24	30	8558
28	88.0	24	30	8921
29	145.0	19	24	12964
30	58.0	49	54	6479
31	76.0	31	38	6855
32	60.0	38	42	5399
33	76.0	30	34	6529
34	76.0	30	34	7129
35	76.0	30	34	7295
36	76.0	30	34	7295
37	86.0	27	33	7895
38	86.0	27	33	9095
39	86.0	27	33	8845
40	86.0	27	33	10295
42	100.0	25	31	10345
43	78.0	24	29	6785
44	70.0	38	43	10000
45	70.0	38	43	23000
46	90.0	24	29	11048
47	176.0	15	19	32250
48	176.0	15	19	35550
49	262.0	13	17	36000
50	68.0	30	31	5195

[ ]:

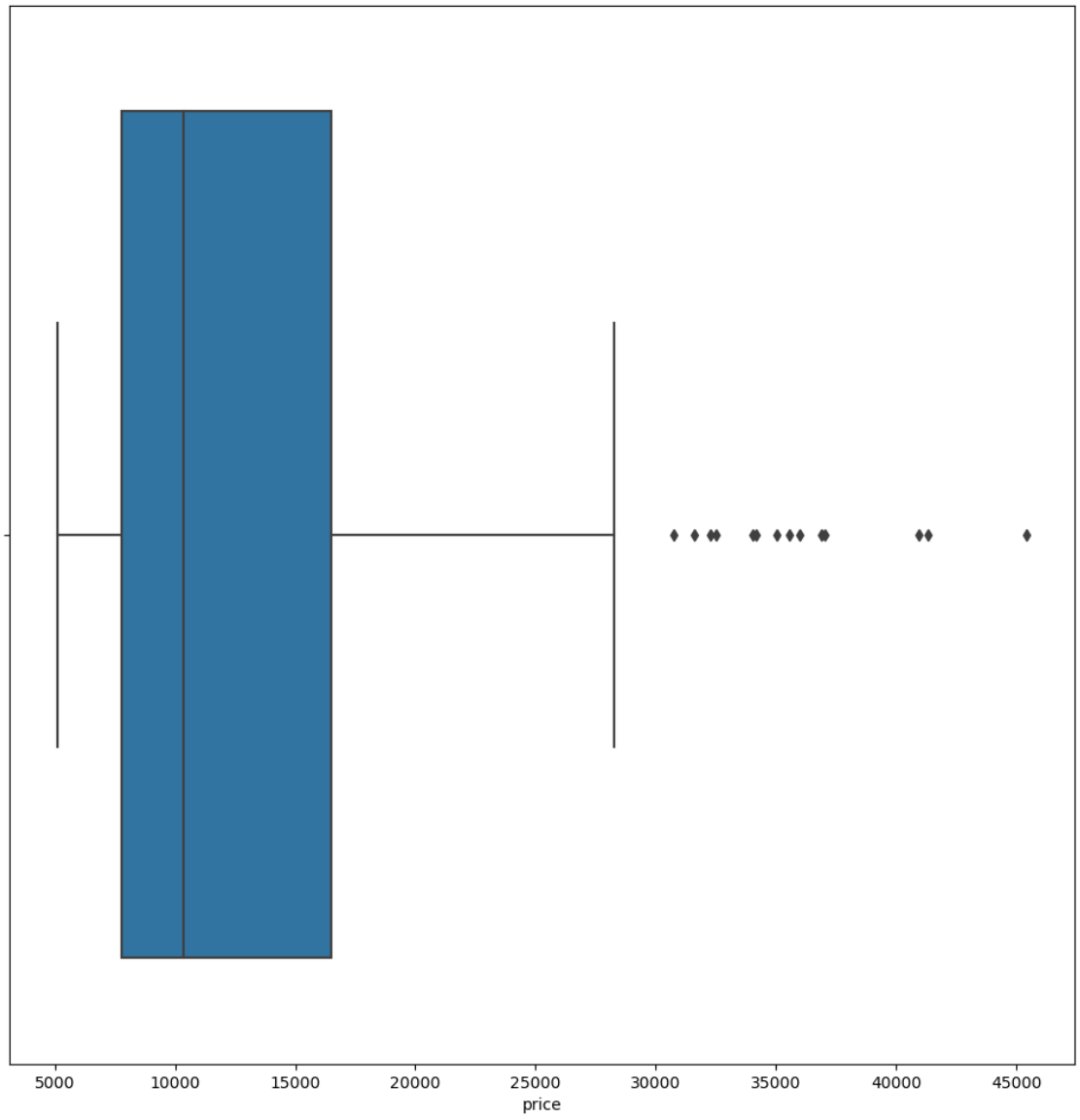
[53]:
```
plt.figure(figsize=(12,12))
sns.boxplot(data=feature,x=target,y='make')
```

[53]: <AxesSubplot:xlabel='price', ylabel='make'>



[54]:
```
plt.figure(figsize=(12,12))
sns.boxplot(data=(feature['make']=='dodge'),x=target)
```

[54]: <AxesSubplot:xlabel='price'>



11 to find null value in large dataset and drop

[55]:
```
ll=pd.read_csv('hp.train.csv')
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_20160\2353299982.py in <module>
----> 1 ll=pd.read_csv('hp.train.csv')

C:\ProgramData\Anaconda3\lib\site-packages\pandas\util\_decorators.py in␣
 ↪wrapper(*args, **kwargs)
```

```
    309                     stacklevel=stacklevel,
    310                 )
--> 311             return func(*args, **kwargs)
    312
    313         return wrapper

C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\parsers\readers.py in␣
 ↪read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col,␣
 ↪usecols, squeeze, prefix, mangle_dupe_cols, dtype, engine, converters,␣
 ↪true_values, false_values, skipinitialspace, skiprows, skipfooter, nrows,␣
 ↪na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates␣
 ↪infer_datetime_format, keep_date_col, date_parser, dayfirst, cache_dates,␣
 ↪iterator, chunksize, compression, thousands, decimal, lineterminator,␣
 ↪quotechar, quoting, doublequote, escapechar, comment, encoding,␣
 ↪encoding_errors, dialect, error_bad_lines, warn_bad_lines, on_bad_lines,␣
 ↪delim_whitespace, low_memory, memory_map, float_precision, storage_options)
    676     kwds.update(kwds_defaults)
    677
--> 678     return _read(filepath_or_buffer, kwds)
    679
    680

C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\parsers\readers.py in␣
 ↪_read(filepath_or_buffer, kwds)
    573
    574     # Create the parser.
--> 575     parser = TextFileReader(filepath_or_buffer, **kwds)
    576
    577     if chunksize or iterator:

C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\parsers\readers.py in␣
 ↪__init__(self, f, engine, **kwds)
    930
    931         self.handles: IOHandles | None = None
--> 932         self._engine = self._make_engine(f, self.engine)
    933
    934     def close(self):

C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\parsers\readers.py in␣
 ↪_make_engine(self, f, engine)
   1214             # "Union[str, PathLike[str], ReadCsvBuffer[bytes],␣
 ↪ReadCsvBuffer[str]]"
   1215             # , "str", "bool", "Any", "Any", "Any", "Any", "Any"
-> 1216             self.handles = get_handle(  # type: ignore[call-overload]
   1217                 f,
   1218                 mode,

C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\common.py in␣
 ↪get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text,␣
 ↪errors, storage_options)
```

```
    784         if ioargs.encoding and "b" not in ioargs.mode:
    785             # Encoding
--> 786             handle = open(
    787                 handle,
    788                 ioargs.mode,

FileNotFoundError: [Errno 2] No such file or directory: 'hp.train.csv'
```

[ ]:
```
ll.head(20)
```

[ ]:
```
ll.isna().sum() #to find null value is there or not
```

[ ]:
```
ll.dropna(how='all') #to drop a row if all columns are empty
```

[ ]:
```
ll.dropna(how='all',subset=['MSSubClass']) #to drop a row if 'MSSubClass' is␣
↪empty
```

12 to find skew

[56]:
```
from scipy.stats import skew
```

[57]:
```
colname=feature.select_dtypes(['int64','float64']).columns #give column name␣
↪that have int64,float64 column name
```

[58]:
```
colname
```

[58]:
```
Index(['symboling', 'normalized-losses', 'width', 'height', 'engine-size',
       'horsepower', 'city-mpg', 'highway-mpg'],
      dtype='object')
```

[65]:
```
skew(df['normalized-losses'])
```

[65]:
```
0.8457209081915792
```

[59]:
```
feature[colname]
```

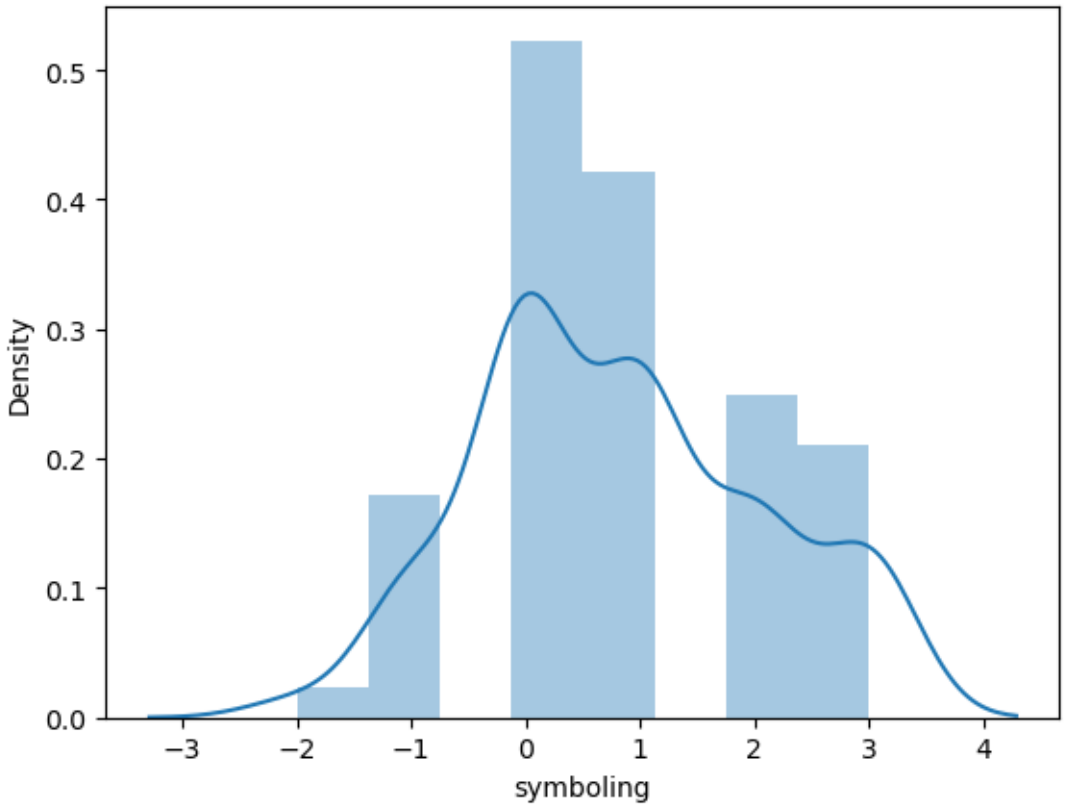
[59]:

	symboling	normalized-losses	width	height	engine-size	horsepower	\
0	3	122.0	64.1	48.8	130	111.0	
1	3	122.0	64.1	48.8	130	111.0	
2	1	122.0	65.5	52.4	152	154.0	
3	2	164.0	66.2	54.3	109	102.0	
4	2	164.0	66.4	54.3	136	115.0	
..	…	…	…	…	…	…	
200	-1	95.0	68.9	55.5	141	114.0	
201	-1	95.0	68.8	55.5	141	160.0	
202	-1	95.0	68.9	55.5	173	134.0	

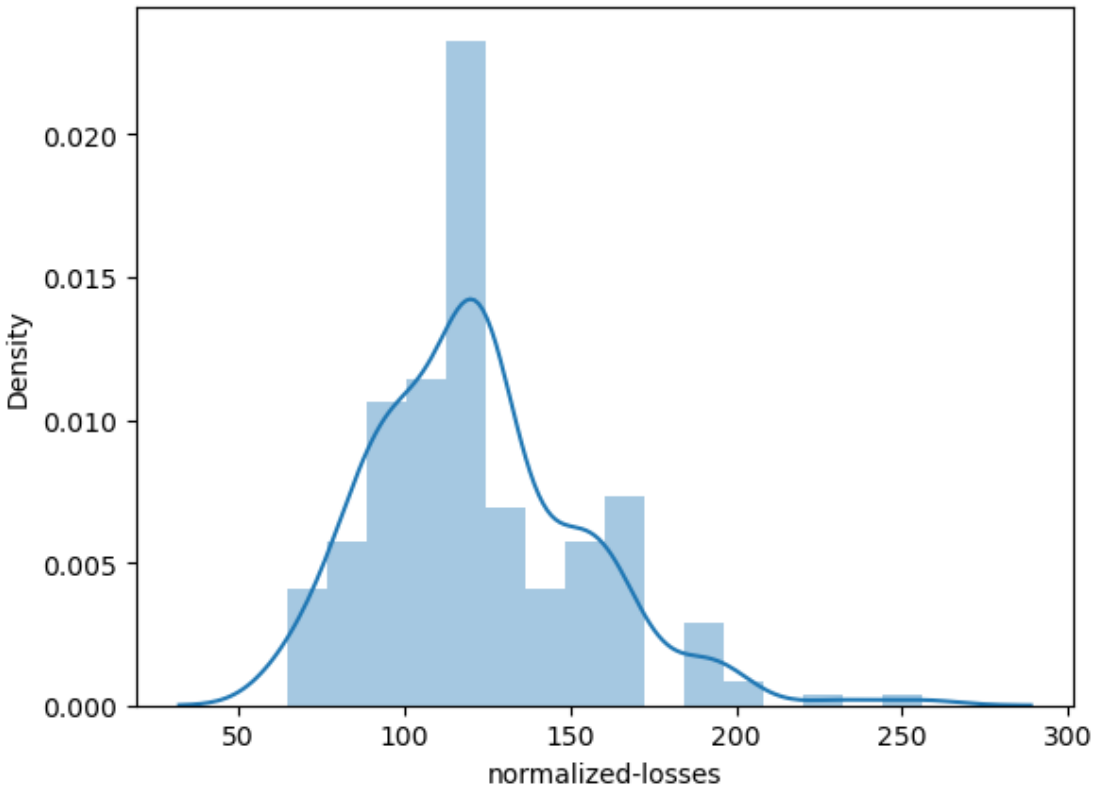
```
203         -1               95.0   68.9    55.5          145       106.0
204         -1               95.0   68.9    55.5          141       114.0

     city-mpg  highway-mpg
0          21           27
1          21           27
2          19           26
3          24           30
4          18           22
..        …            …
200        23           28
201        19           25
202        18           23
203        26           27
204        19           25

[205 rows x 8 columns]
```

[66]:
```
for i in feature[colname]:
    print(i)
    print(skew(feature[i]))
    #plt.figure()
    sns.distplot(feature[i]) #for display distplot to find have skew or not
    plt.show()
```

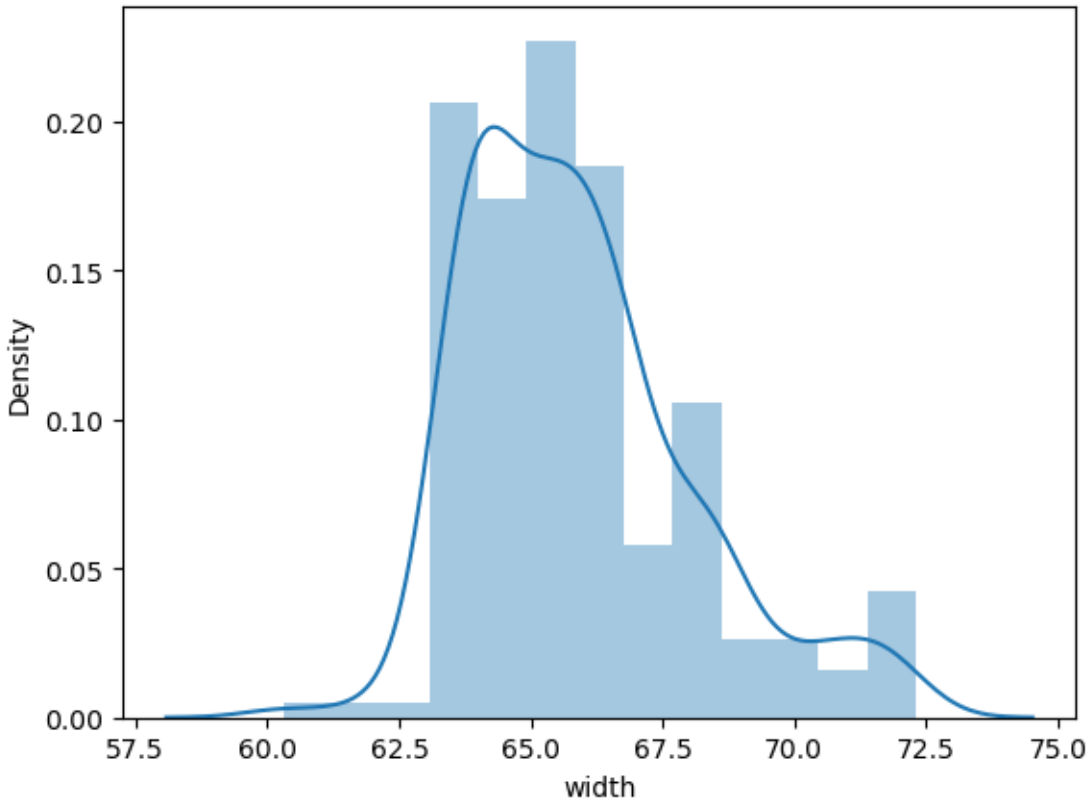
```
symboling
0.20952469094997359
```



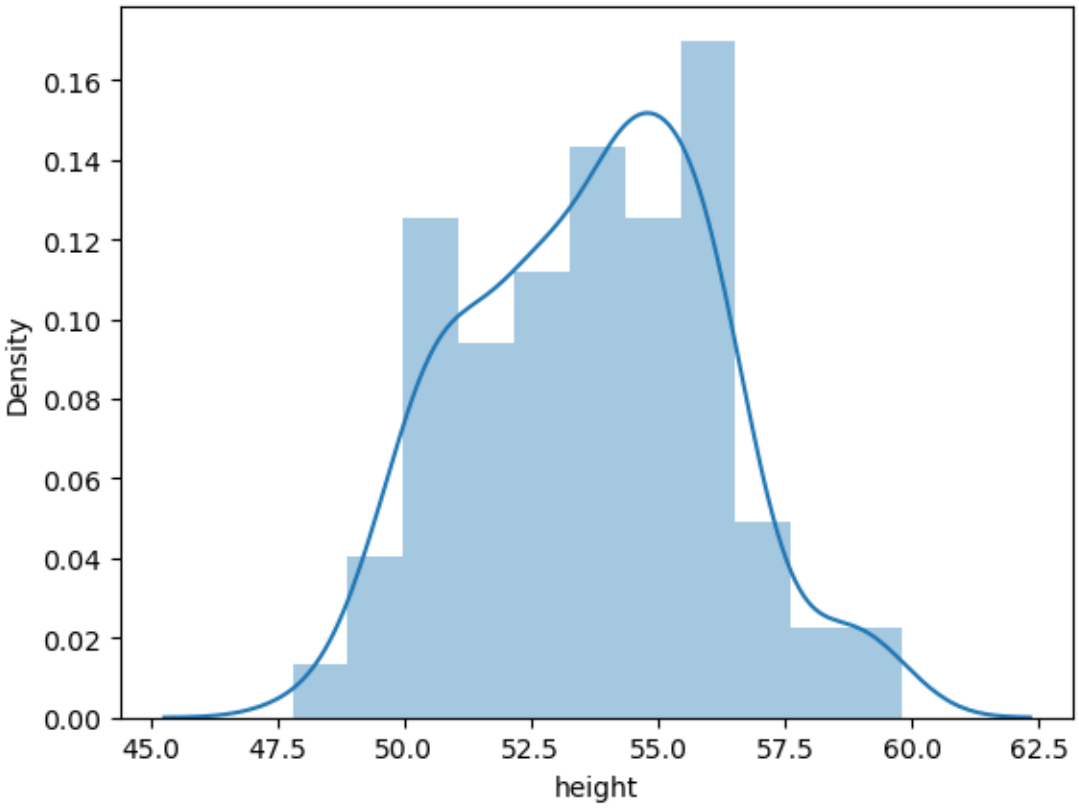

```
normalized-losses
0.8485348696008058
```



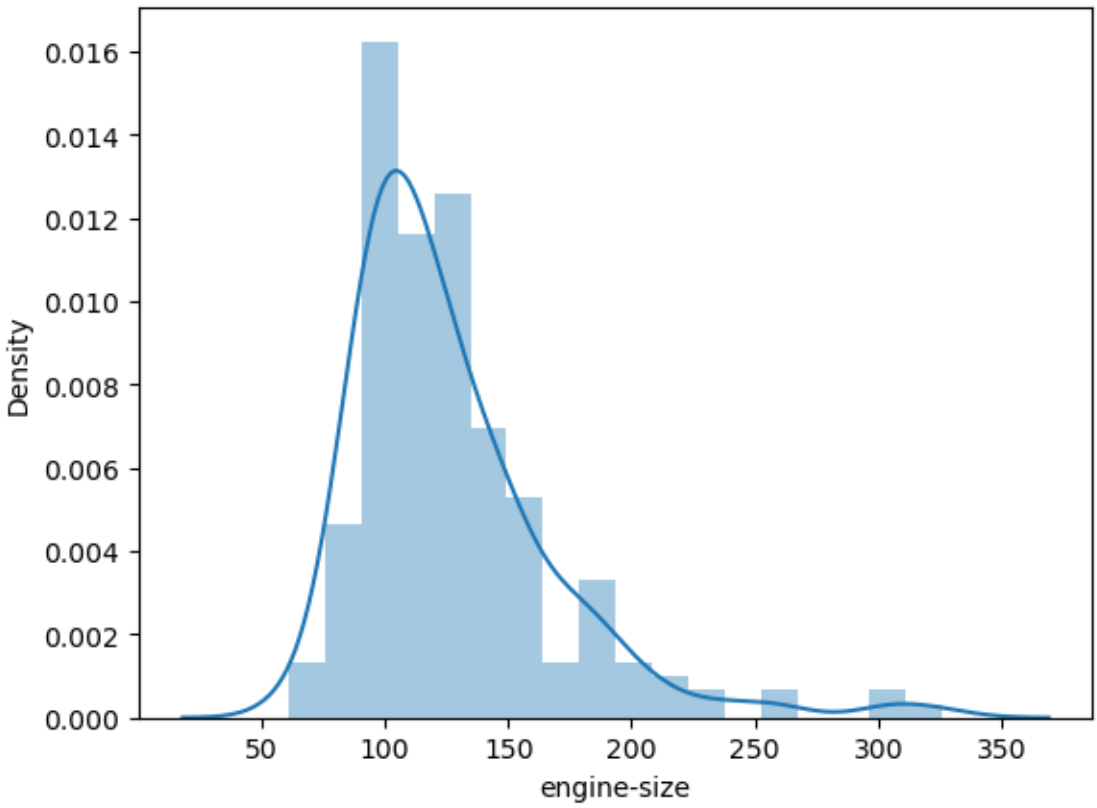

```
width
0.8973753485201392
```



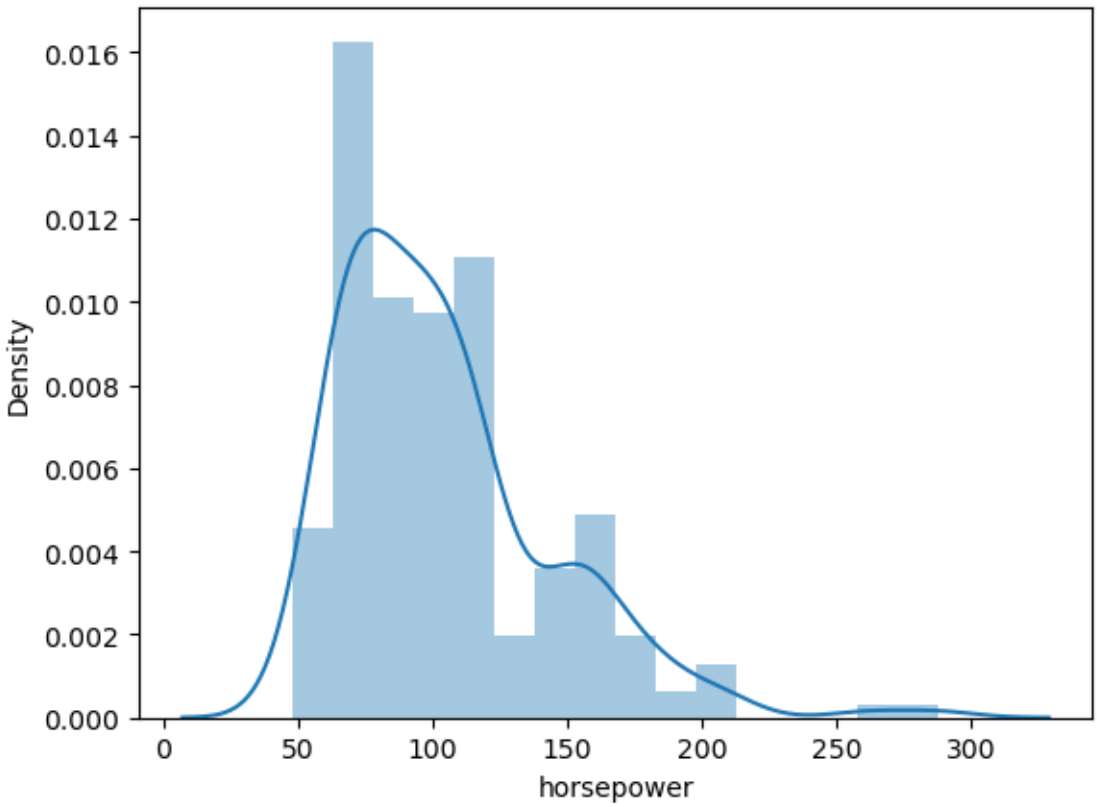

```
height
0.06265991683394276
```



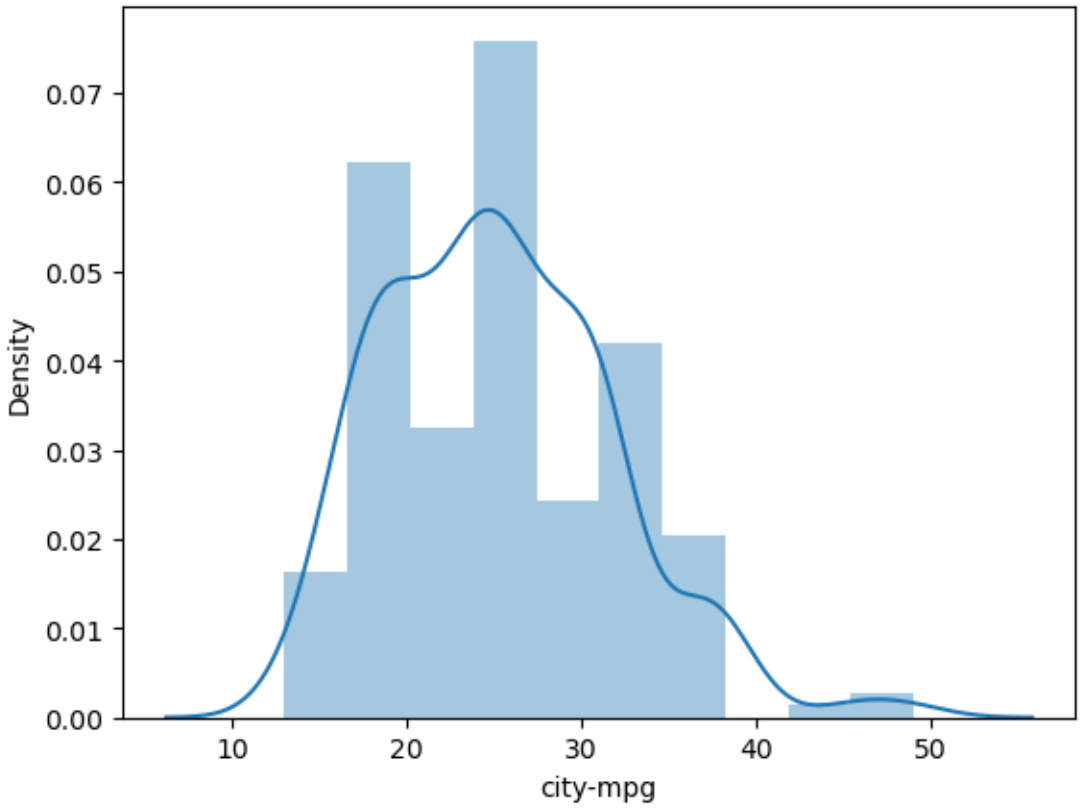

```
engine-size
1.9333748457840114
```



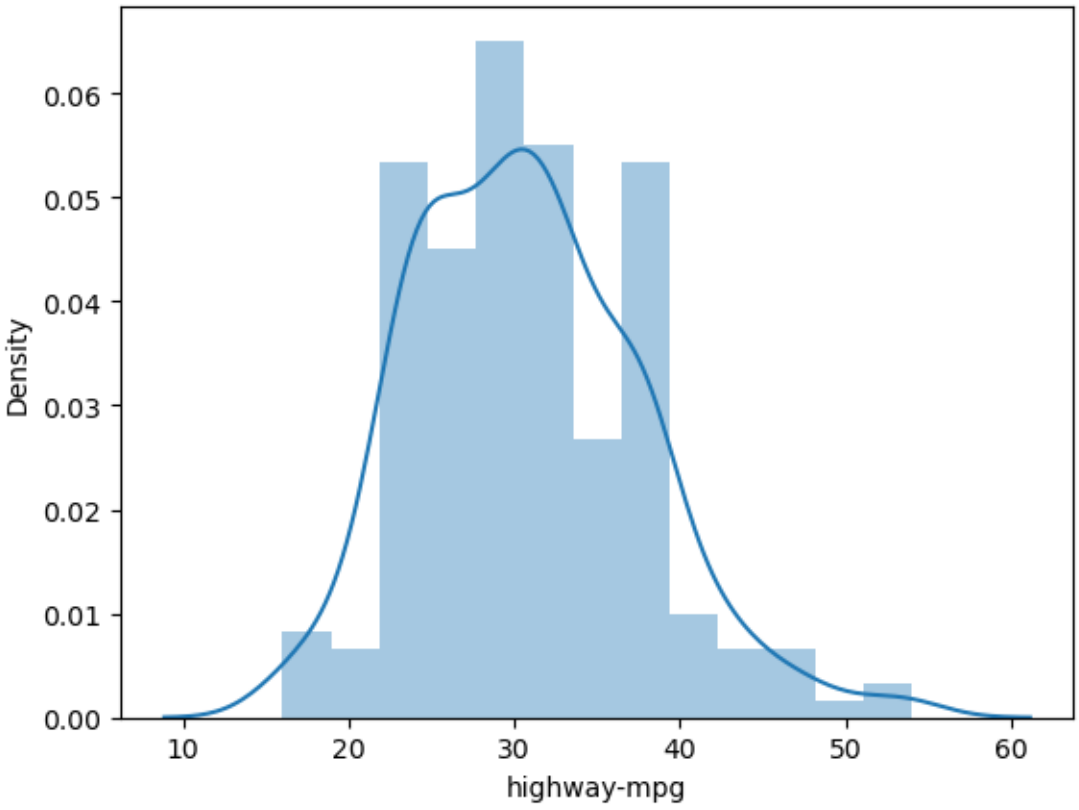

```
horsepower
1.3875147343096037
```




```
city-mpg
0.6588377533622138
```

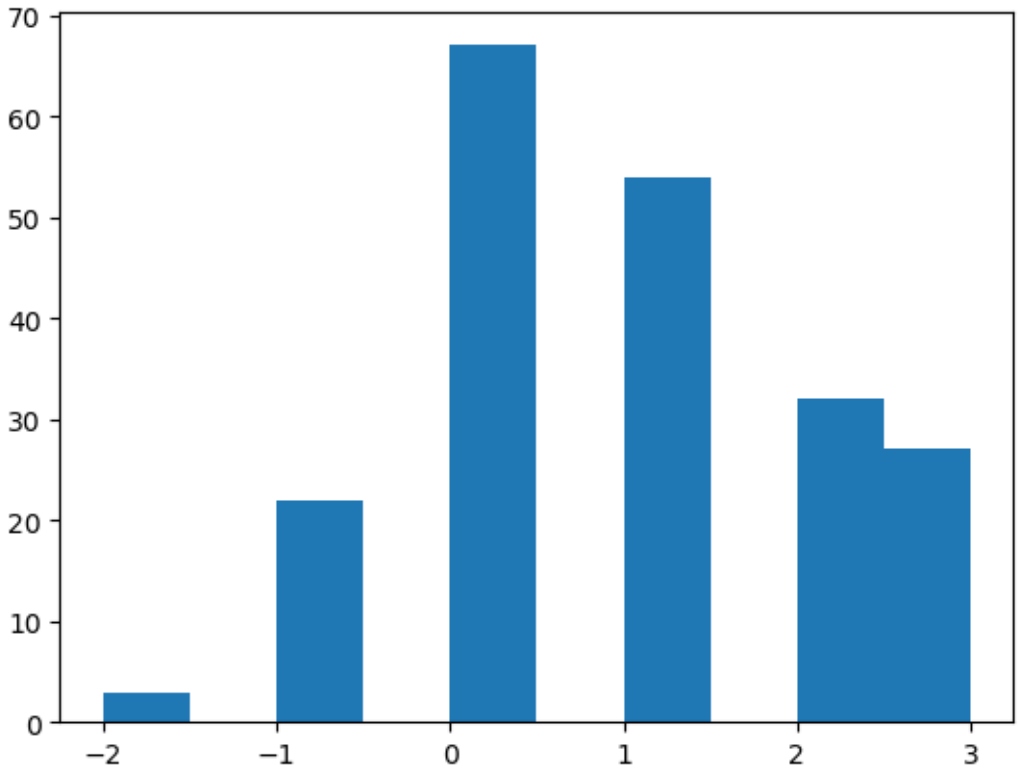



```
highway-mpg
0.5360379305163596
```

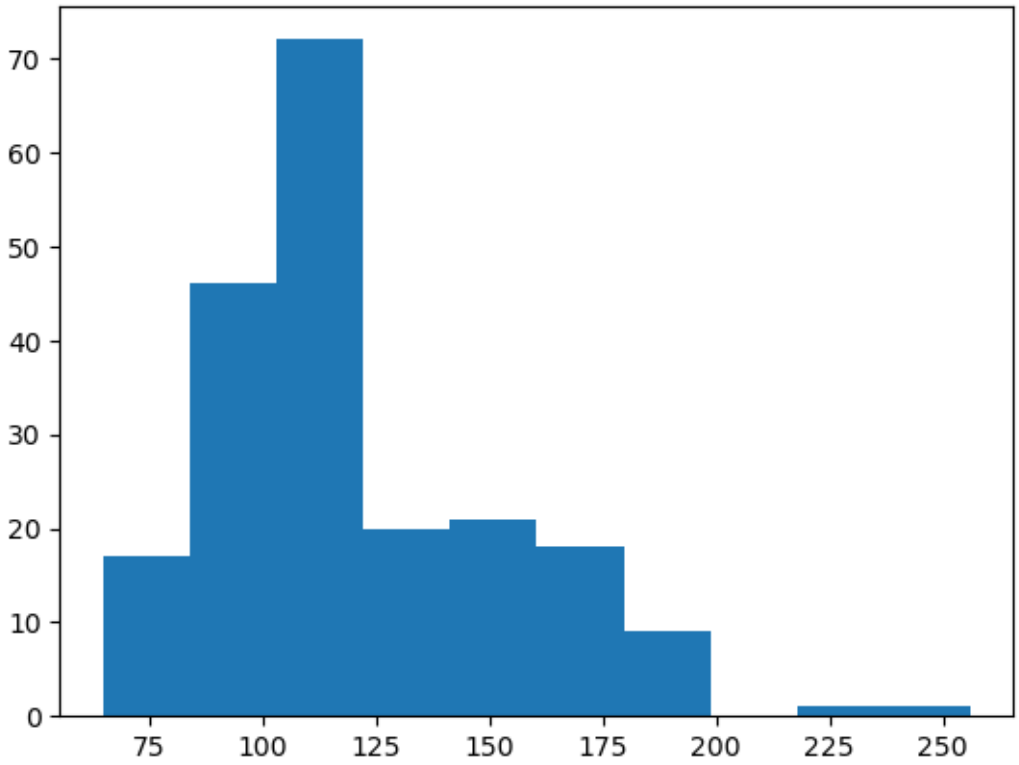


[62]:
```
for i in feature[colname]:
    print(i)
    print(skew(feature[i]))
    plt.figure()
    plt.hist(feature[i]) #for display distplot to find have skew or not
    plt.show()
```

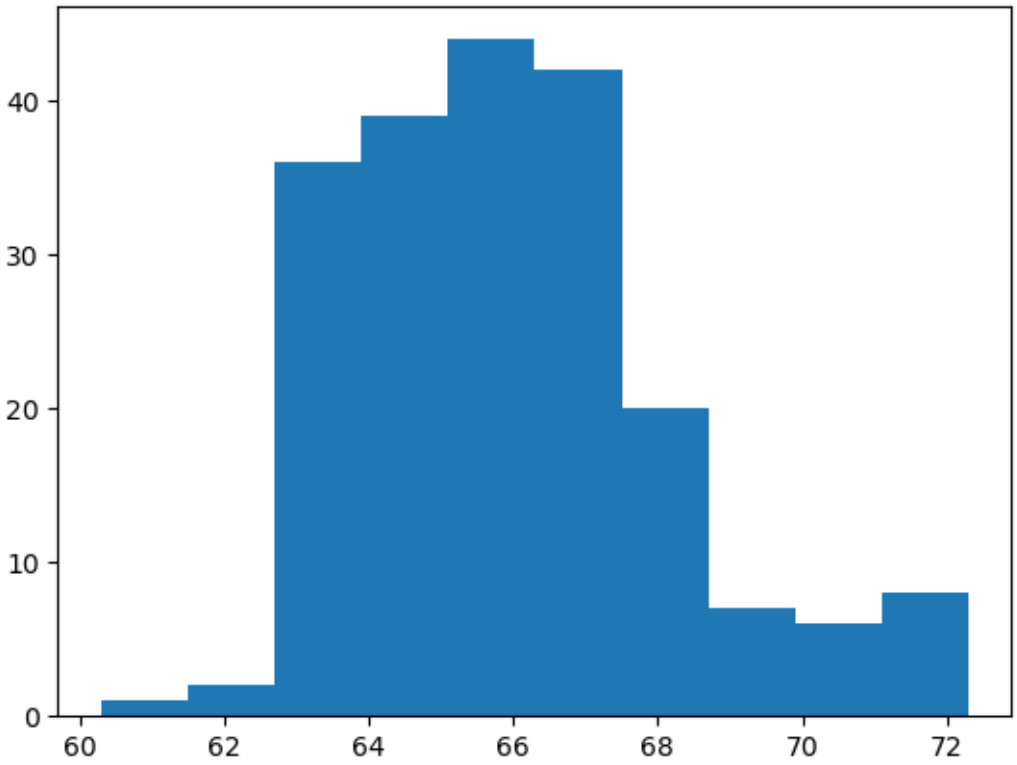
```
symboling
0.20952469094997359
```



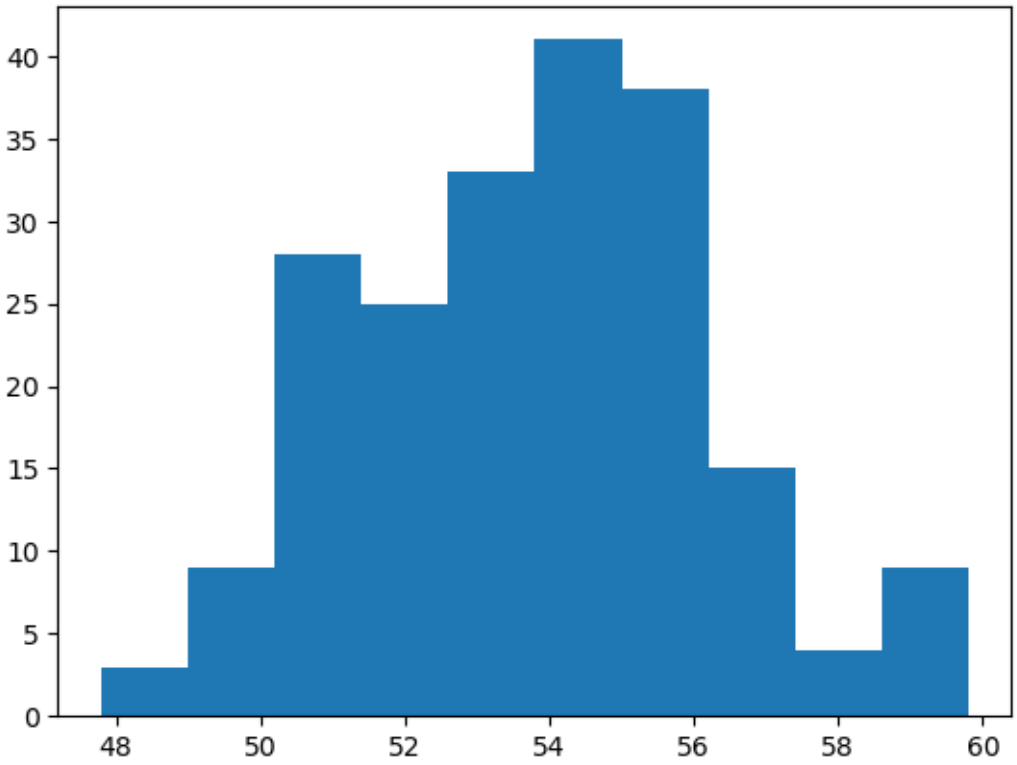

```
normalized-losses
0.8485348696008058
```



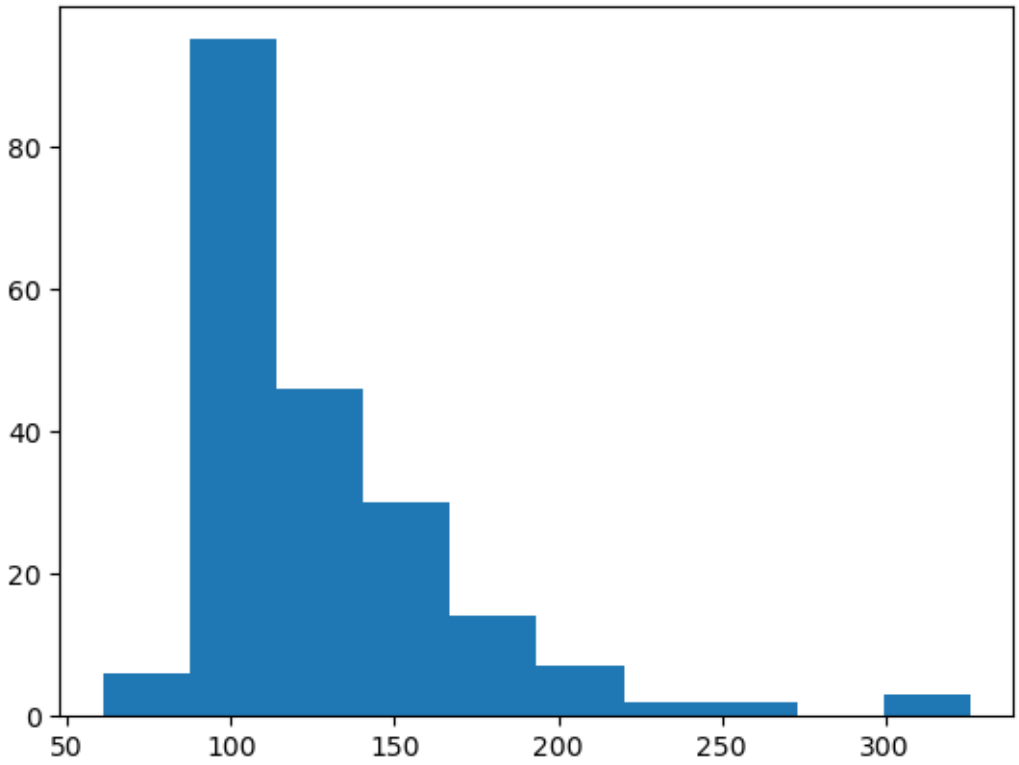

```
width
0.8973753485201392
```



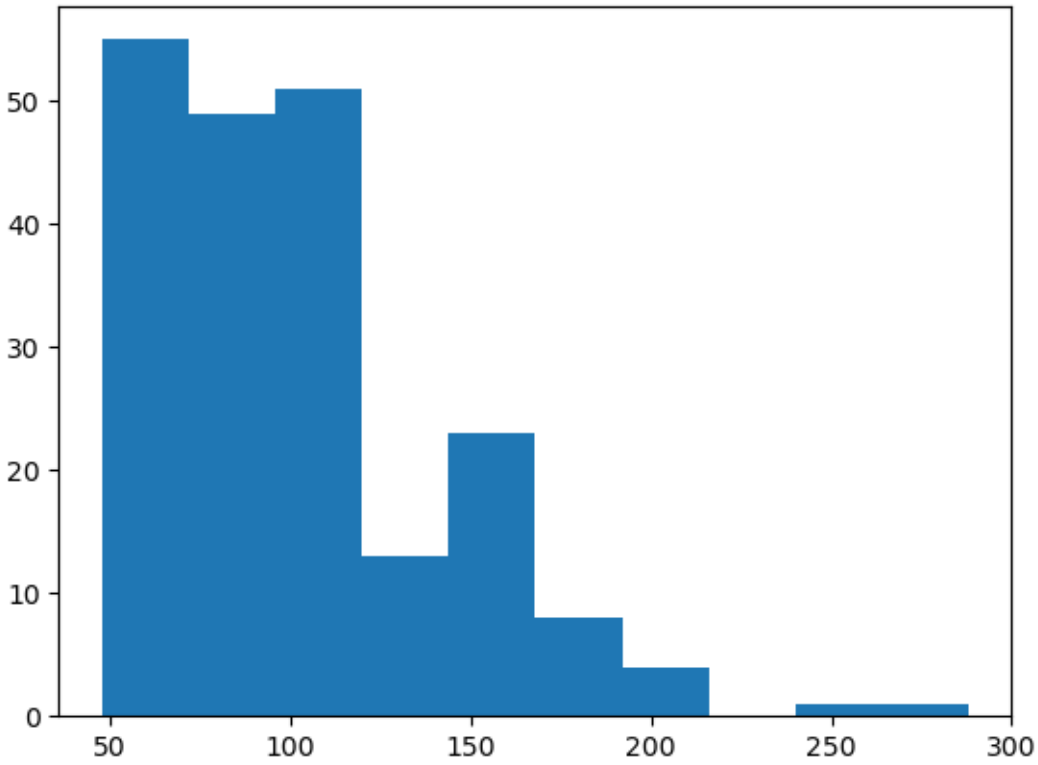

```
height
0.06265991683394276
```



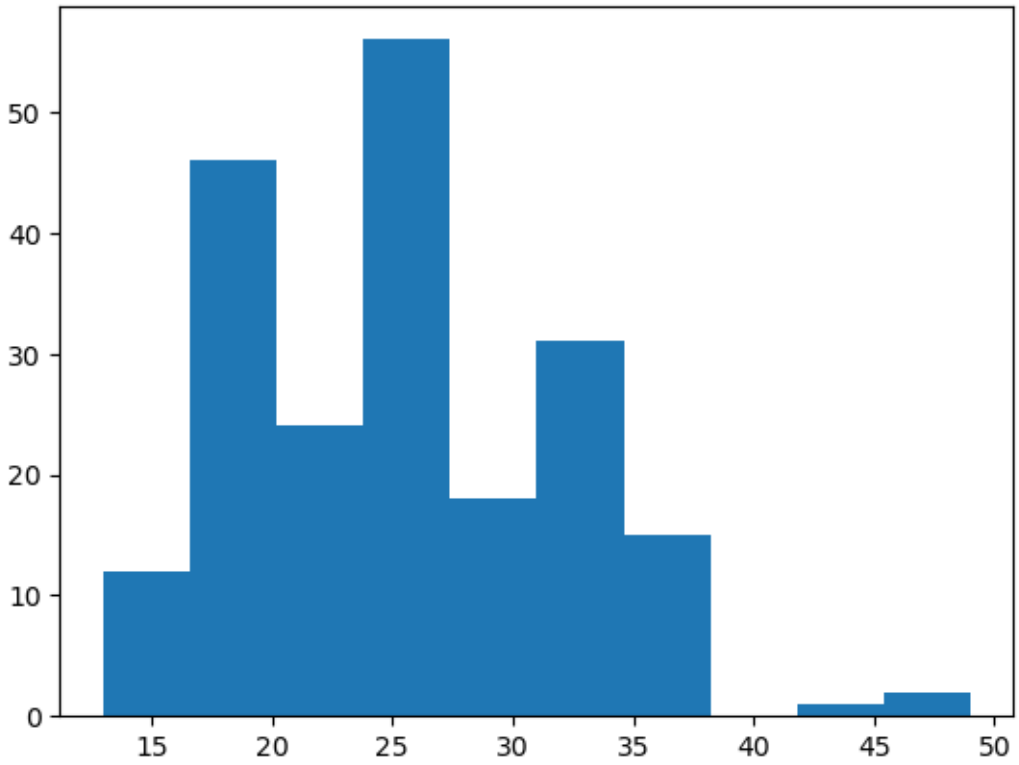

```
engine-size
1.9333748457840114
```



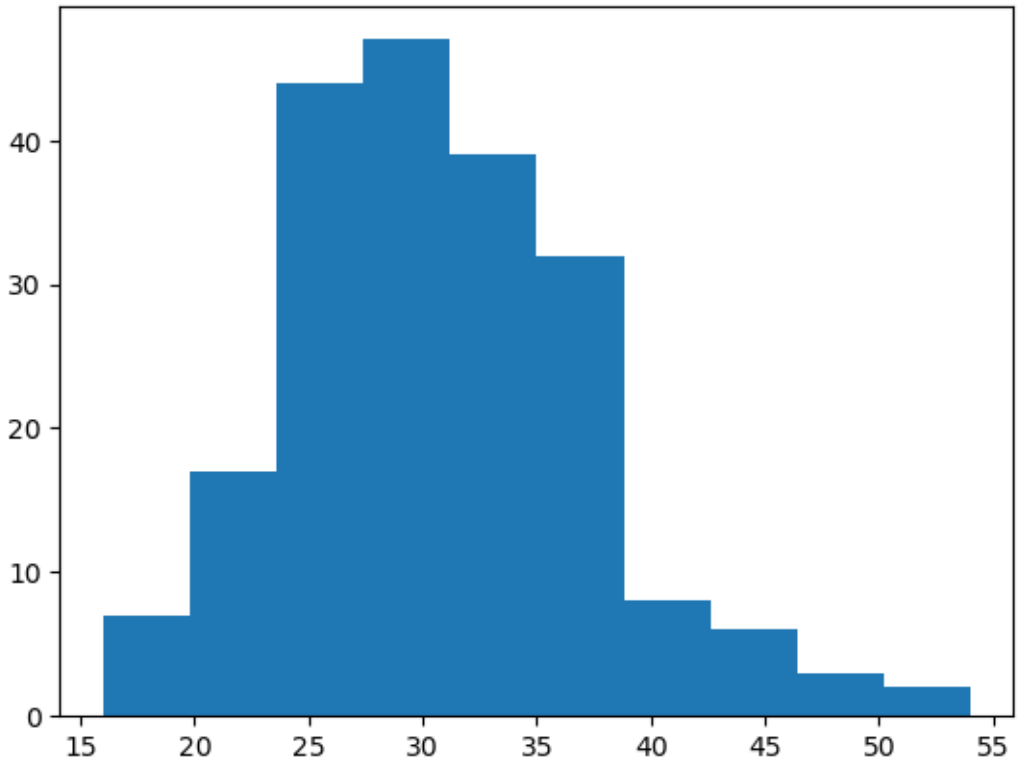

```
horsepower
1.3875147343096037
```




```
city-mpg
0.6588377533622138
```




```
highway-mpg
0.5360379305163596
```



[70]:
```
pd.concat([feature,target],axis=1).corr().style.background_gradient()
```

[70]:
```
<pandas.io.formats.style.Styler at 0x19ddfe17310>
```

[ ]:
```

```