ML - PROJECT 2 - VEHICLE PRICE PREDICTION

our algorithm will take vehicle details like mileage, engine type, no of doors, length, width, height, engine capacity, etc.... and our algorithm will predict PRICE of the vehicle.

step1 - load data

In [1]:
```
import pandas as pd

auto_data = pd.read_csv('auto.txt')
auto_data.head()
```

Out[1]:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	.
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	.
1	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	.
2	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	.
3	2	164	audi	gas	std	four	sedan	fwd	front	99.8	.
4	2	164	audi	gas	std	four	sedan	4wd	front	99.4	.

5 rows × 26 columns

step 2 - clean data

In [2]:
```
# you can observe there are some columns with values ? lets clean these.
import numpy as np
auto_data = auto_data.replace('?',np.nan)
auto_data.head()
```

Out[2]:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	.
0	3	NaN	alfa-romero	gas	std	two	convertible	rwd	front	88.6	.
1	3	NaN	alfa-romero	gas	std	two	convertible	rwd	front	88.6	.
2	1	NaN	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	.
3	2	164	audi	gas	std	four	sedan	fwd	front	99.8	.
4	2	164	audi	gas	std	four	sedan	4wd	front	99.4	.

5 rows × 26 columns

In [3]:
```
auto_data['price'].describe() #lets see what is the data type of price column
```

Out[3]:
```
count      201
unique     186
top       8921
freq         2
Name: price, dtype: object
```

In [4]:
```
auto_data['price'] = pd.to_numeric(auto_data['price'], errors='coerce') #coerce
auto_data['price'].describe()
```

Out[4]:
```
count      201.000000
mean     13207.129353
std       7947.066342
min       5118.000000
25%       7775.000000
50%      10295.000000
75%      16500.000000
max      45400.000000
Name: price, dtype: float64
```

In [5]:
```
# let us remove unwanted columns -- which are not useful.
```

In [6]:
```
auto_data = auto_data.drop('normalized-losses', axis=1)
auto_data.head()
```

Out[6]:

	symboling	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	...
0	3	alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	...
1	3	alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	...
2	1	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	171.2	...
3	2	audi	gas	std	four	sedan	fwd	front	99.8	176.6	...
4	2	audi	gas	std	four	sedan	4wd	front	99.4	176.6	...

5 rows × 25 columns

In [7]:
```
auto_data.columns
```

Out[7]:
```
Index(['symboling', 'make', 'fuel-type', 'aspiration', 'num-of-doors',
       'body-style', 'drive-wheels', 'engine-location', 'wheel-base', 'length',
       'width', 'height', 'curb-weight', 'engine-type', 'num-of-cylinders',
       'engine-size', 'fuel-system', 'bore', 'stroke', 'compression-ratio',
       'horsepower', 'peak-rpm', 'city-mpg', 'highway-mpg', 'price'],
      dtype='object')
```

In [8]:
```
auto_data['horsepower'].describe()
```

Out[8]:
```
count     203
unique     59
top        68
freq       19
Name: horsepower, dtype: object
```

In [9]:
```
auto_data['horsepower'] = pd.to_numeric(auto_data['horsepower'], errors='coerce')
auto_data['horsepower'].describe()
```

Out[9]:
```
count    203.000000
mean     104.256158
std       39.714369
min       48.000000
25%       70.000000
50%       95.000000
75%      116.000000
max      288.000000
Name: horsepower, dtype: float64
```

In [10]:
```
auto_data.columns
```

Out[10]:
```
Index(['symboling', 'make', 'fuel-type', 'aspiration', 'num-of-doors',
       'body-style', 'drive-wheels', 'engine-location', 'wheel-base', 'length',
       'width', 'height', 'curb-weight', 'engine-type', 'num-of-cylinders',
       'engine-size', 'fuel-system', 'bore', 'stroke', 'compression-ratio',
       'horsepower', 'peak-rpm', 'city-mpg', 'highway-mpg', 'price'],
      dtype='object')
```

In [11]:
```
auto_data['price']
```

Out[11]:
```
0      13495.0
1      16500.0
2      16500.0
3      13950.0
4      17450.0
        ...
200    16845.0
201    19045.0
202    21485.0
203    22470.0
204    22625.0
Name: price, Length: 205, dtype: float64
```

In [12]:
```
auto_data['bore'] = pd.to_numeric(auto_data['bore'], errors='coerce')#cnvrt to fl
auto_data['bore'].describe()
```

Out[12]:
```
count    201.000000
mean       3.329751
std        0.273539
min        2.540000
25%        3.150000
50%        3.310000
75%        3.590000
max        3.940000
Name: bore, dtype: float64
```

In [13]:
```
auto_data['stroke'] = pd.to_numeric(auto_data['stroke'], errors='coerce')#cnvrt t
auto_data['stroke'].describe()
```

Out[13]:
```
count    201.000000
mean       3.255423
std        0.316717
min        2.070000
25%        3.110000
50%        3.290000
75%        3.410000
max        4.170000
Name: stroke, dtype: float64
```

In [14]:

```
auto_data['peak-rpm'] = pd.to_numeric(auto_data['peak-rpm'], errors='coerce')#cn
auto_data['peak-rpm'].describe()
```

Out[14]:

```
count     203.000000
mean     5125.369458
std       479.334560
min      4150.000000
25%      4800.000000
50%      5200.000000
75%      5500.000000
max      6600.000000
Name: peak-rpm, dtype: float64
```

In [15]:

```
auto_data['num-of-cylinders'].describe()
```

Out[15]:

```
count      205
unique       7
top       four
freq       159
Name: num-of-cylinders, dtype: object
```

In [16]:

```
auto_data['num-of-cylinders']
```

Out[16]:

```
0      four
1      four
2       six
3      four
4      five
       ...
200    four
201    four
202     six
203     six
204    four
Name: num-of-cylinders, Length: 205, dtype: object
```

In [17]:

```
cylinders_dict = {
    'two':2,
    'three':3,
    'four':4,
    'five':5,
    'six':6,
    'eight':8,
    'twelve':12
}
auto_data['num-of-cylinders'].replace(cylinders_dict, inplace = True)
auto_data['num-of-cylinders'].head()
```

Out[17]:

```
0    4
1    4
2    6
3    4
4    5
Name: num-of-cylinders, dtype: int64
```

In [18]:

```
a = {'1bbl':1, '2bbl':2, '4bbl':4, 'idi':5, 'mfi':6, 'mpfi':7,
               'spdi':8, 'spfi':9}
auto_data['fuel-system'].replace(a, inplace = True)

b = {'dohc':1, 'dohcv':2, 'l':3, 'ohc':4, 'ohcf':5, 'ohcv':6,
              'rotor':7}
auto_data['engine-type'].replace(b, inplace = True)

c = {'front':1, 'rear':2}
auto_data['engine-location'].replace(c, inplace = True)

d = {'4wd':1, 'fwd':2, 'rwd':3}
auto_data['drive-wheels'].replace(d, inplace = True)

e = {
    'alfa-romero' : 1,'audi' : 2,'bmw': 3,'chevrolet' :4,'dodge':5,
        'honda':6, 'isuzu':7,'jaguar':8, 'mazda':9, 'mercedes-benz':10,
        'mercury':11,'mitsubishi':12, 'nissan':13, 'peugot':14,
        'plymouth':15,'porsche':16, 'renault':17, 'saab':18, 'subaru':19,
       'toyota':20, 'volkswagen':21, 'volvo':22
}
auto_data['make'].replace(e, inplace = True)

f = {'convertible':1,'hardtop':2, 'hatchback':3, 'sedan':4, 'wagon':5}
auto_data['body-style'].replace(f, inplace = True)

g = {'four':4, 'two':2}
auto_data['num-of-doors'].replace(g, inplace = True)

h = {'std':0, 'turbo':1}
auto_data['aspiration'].replace(h, inplace = True)

auto_data.head(10)
```

Out[18]:

	symboling	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	...	engine-size
0	3	1	gas	0	2.0	1	3	1	88.6	168.8	...	130
1	3	1	gas	0	2.0	1	3	1	88.6	168.8	...	130
2	1	1	gas	0	2.0	3	3	1	94.5	171.2	...	152
3	2	2	gas	0	4.0	4	2	1	99.8	176.6	...	109
4	2	2	gas	0	4.0	4	1	1	99.4	176.6	...	136
5	2	2	gas	0	2.0	4	2	1	99.8	177.3	...	136
6	1	2	gas	0	4.0	4	2	1	105.8	192.7	...	136
7	1	2	gas	0	4.0	5	2	1	105.8	192.7	...	136
8	1	2	gas	1	4.0	4	2	1	105.8	192.7	...	131
9	0	2	gas	1	2.0	3	1	1	99.5	178.2	...	131

10 rows × 25 columns

In [19]:

```
i = {'gas':0, 'diesel':1}
auto_data['fuel-type'].replace(i, inplace = True)

auto_data.head(10)
```

Out[19]:

	symboling	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	...	engine-size
0	3	1	0	0	2.0	1	3	1	88.6	168.8	...	130
1	3	1	0	0	2.0	1	3	1	88.6	168.8	...	130
2	1	1	0	0	2.0	3	3	1	94.5	171.2	...	152
3	2	2	0	0	4.0	4	2	1	99.8	176.6	...	109
4	2	2	0	0	4.0	4	1	1	99.4	176.6	...	136
5	2	2	0	0	2.0	4	2	1	99.8	177.3	...	136
6	1	2	0	0	4.0	4	2	1	105.8	192.7	...	136
7	1	2	0	0	4.0	5	2	1	105.8	192.7	...	136
8	1	2	0	1	4.0	4	2	1	105.8	192.7	...	131
9	0	2	0	1	2.0	3	1	1	99.5	178.2	...	131

10 rows × 25 columns

In [20]:
```
auto_data.isnull()
```

Out[20]:

	symboling	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	...	engi si
0	False	False	False	False	False	False	False	False	False	False	...	Fal
1	False	False	False	False	False	False	False	False	False	False	...	Fal
2	False	False	False	False	False	False	False	False	False	False	...	Fal
3	False	False	False	False	False	False	False	False	False	False	...	Fal
4	False	False	False	False	False	False	False	False	False	False	...	Fal
...	...	...	...	...	...	...	...	...	...	...	...	
200	False	False	False	False	False	False	False	False	False	False	...	Fal
201	False	False	False	False	False	False	False	False	False	False	...	Fal
202	False	False	False	False	False	False	False	False	False	False	...	Fal
203	False	False	False	False	False	False	False	False	False	False	...	Fal
204	False	False	False	False	False	False	False	False	False	False	...	Fal

205 rows × 25 columns

In [21]:
```
auto_data.isna().sum()
```

Out[21]:
```
symboling            0
make                 0
fuel-type            0
aspiration           0
num-of-doors         2
body-style           0
drive-wheels         0
engine-location      0
wheel-base           0
length               0
width                0
height               0
curb-weight          0
engine-type          0
num-of-cylinders     0
engine-size          0
fuel-system          0
bore                 4
stroke               4
compression-ratio    0
horsepower           2
peak-rpm             2
city-mpg             0
highway-mpg          0
price                4
dtype: int64
```

In [ ]: