In [30]:
```
import pandas as pd
import numpy as np
```

In [31]:
```
df = pd.read_csv('E:/Work/P/Workspace/SupL_1/import2.csv',names =['symboling',
 'normalized-losses', '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', 'b
ore', 'stroke',
           'compression-ratio', 'horsepower', 'peak-rpm', 'city-mpg', 'highway
-mpg', 'price'])
df_head=df.head(10)
```

In [33]:
```
df_head
```

Out[33]:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location
0	3	NaN	alfa-romero	gas	std	two	convertible	rwd	front
1	3	NaN	alfa-romero	gas	std	two	convertible	rwd	front
2	1	NaN	alfa-romero	gas	std	two	hatchback	rwd	front
3	2	164.0	audi	gas	std	four	sedan	fwd	front
4	2	164.0	audi	gas	std	four	sedan	4wd	front
5	2	NaN	audi	gas	std	two	sedan	fwd	front
6	1	158.0	audi	gas	std	four	sedan	fwd	front
7	1	NaN	audi	gas	std	four	wagon	fwd	front
8	1	158.0	audi	gas	turbo	four	sedan	fwd	front
9	0	NaN	audi	gas	turbo	two	hatchback	4wd	front

10 rows × 26 columns

In [34]:
```
#Q 2:Explain the problem statement. What are you predicting and what attribute
s you have to predict?
#Ans : We can use this data to predict the price of the car based on various a
ttributes
```

In [35]:

```
#Q 3:Browse a sample record from the dataframe. Are there any missing values?
df.iloc[14]
#The normalized losses value is missing for this tuple
```

Out[35]:

```
symboling                1
normalized-losses      NaN
make                   bmw
fuel-type              gas
aspiration             std
num-of-doors          four
body-style           sedan
drive-wheels           rwd
engine-location      front
wheel-base           103.5
length                 189
width                 66.9
height                55.7
curb-weight           3055
engine-type            ohc
num-of-cylinders       six
engine-size            164
fuel-system           mpfi
bore                  3.31
stroke                3.19
compression-ratio        9
horsepower             121
peak-rpm              4250
city-mpg                20
highway-mpg             25
price                24565
Name: 14, dtype: object
```

In [36]:

```
# #Q 4: How many records are available in the data set and how many attribute
s.
# Do you think the depth (number of records) is sufficient given the breadth?
# In other words, is the sample likely to be a good representative of the univ
erse?
#Ans : Number of Records - 205
df.shape[0]
```

Out[36]:

```
205
```

In [37]:

```
#Ans : Number of Columns - 26
df.shape[1]
```

Out[37]:

```
26
```

In [ ]:

```
#Que: Do you think the depth (number of records) is sufficient given the breadth?
#In other words, is the sample likely to be a good representative of the universe?

#Ans : No the number of records is not sufficient for precise prediction compared to
#the given number of attributes. The sample is a mediocre representative of the universe.
```

In [ ]:

```
#Que : Analyse the data distribution for the various attributes and share your observations ?
```

In [53]:

```
df['price'].corr(df['symboling'])
```

Out[53]:

```
-0.082391187169623584
```

In [54]:

```
df['price'].corr(df['wheel-base'])
```

Out[54]:

```
0.58464182226550787
```

In [55]:

```
df['price'].corr(df['length' ])
```

Out[55]:

```
0.69062838044836405
```

In [56]:

```
df['price'].corr(df['width'])
```

Out[56]:

```
0.75126534405226697
```

In [57]:

```
df['price'].corr(df['height'])
```

Out[57]:

```
0.13548630756805977
```

In [58]:

```
df['price'].corr(df['curb-weight'])
```

Out[58]:

```
0.83441452577028474
```

In [59]:

```
df['price'].corr(df['engine-size'])
```

Out[59]:

```
0.87233516744551831
```

In [60]:

```
df['price'].corr(df['bore'])
```

Out[60]:

```
0.54343586641885455
```

In [61]:

```
df['price'].corr(df['stroke'])
```

Out[61]:

```
0.082309827389704937
```

In [62]:

```
df['price'].corr(df['compression-ratio'])
```

Out[62]:

```
0.07110732668194146
```

In [63]:
```
df['price'].corr(df['horsepower'])
```

Out[63]:
```
0.81053308213220654
```

In [64]:
```
df['price'].corr(df['highway-mpg'])
```

Out[64]:
```
-0.70469226505895299
```

In [65]:
```
#Ans : The following attributes have positive correlation with attribute pricing.
#(i.e the values that impact the price of the car)
        #wheel-base
        #length
        #width
        #height
        #curb-weight
        #engine-size
        #bore
        #stroke
        #compression-ratio
        #horsepower
#The following attributes have negative correlation with attribute pricing.
#(i.e the values that don't impact the price of the car)
        #symboling
        #highway-mpg
```

In [66]:
```
#Que : Are there any independent attributes which have |R| close to 1?
#Ans : engine-size attribute has |R| close to 1
```

In [67]:
```
#Que : Which attributes seem to have stronger relation with the dependent variable (Price of the car)?
#Ans : horsepower,engine-size and curb-weight
```

In [ ]:
```
#Que : Given the above analysis, which algorithm is likely to give a better accuracy? Why?
#Ans : logistic regression
```