```
import pandas as pd
```

```
df = pd.read_csv("https://raw.githubusercontent.com/AmenaNajeeb/Data/master/Automobile_data.csv")
```

```
df.head(10)
```

	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	...	engine-size	fuel-system	bore	stroke	co
0	NaN	alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	...	130	mpfi	3.47	2.68	
1	NaN	alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	...	130	mpfi	3.47	2.68	
2	NaN	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	171.2	...	152	mpfi	2.68	3.47	
3	164.0	audi	gas	std	four	sedan	fwd	front	99.8	176.6	...	109	mpfi	3.19	3.40	
4	164.0	audi	gas	std	four	sedan	4wd	front	99.4	176.6	...	136	mpfi	3.19	3.40	
5	NaN	audi	gas	std	two	sedan	fwd	front	99.8	177.3	...	136	mpfi	3.19	3.40	
6	158.0	audi	gas	std	four	sedan	fwd	front	105.8	192.7	...	136	mpfi	3.19	3.40	
7	NaN	audi	gas	std	four	wagon	fwd	front	105.8	192.7	...	136	mpfi	3.19	3.40	
8	158.0	audi	gas	turbo	four	sedan	fwd	front	105.8	192.7	...	131	mpfi	3.13	3.40	
9	NaN	audi	gas	turbo	two	hatchback	4wd	front	99.5	178.2	...	131	mpfi	3.13	3.40	

10 rows × 25 columns

```
df.shape
```

```
(30, 25)
```

```
df.isnull().sum()
```

```
normalized-losses    10
make                  0
fuel-type             0
aspiration            0
num-of-doors          0
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                  0
stroke                0
compression-ratio     0
horsepower            0
peak-rpm              0
city-mpg              0
highway-mpg           0
price                 0
dtype: int64
```

```
# normalized-losses has the highest number of null values
```

```
df['normalized-losses'].fillna(value=df['normalized-losses'].mean(),inplace=True)
```

```
df[['SalesPrice','Lakhs']]=(df['price'].str.split("L",expand=True))
```

```
df["engine-type"].value_counts()
```

```
ohc     26
dohc     3
```

```
ohcv     1
Name: engine-type, dtype: int64
```

```
from sklearn import preprocessing
```

```
le = preprocessing.LabelEncoder()
df["engine-type"]=le.fit_transform(df["engine-type"])
```

```
df["engine-type"].value_counts()
```

```
1    26
0     3
2     1
Name: engine-type, dtype: int64
```

```
df["num-of-cylinders"].value_counts()
```

```
four     16
six       7
five      6
three     1
Name: num-of-cylinders, dtype: int64
```

```
df["num-of-cylinders"]=le.fit_transform(df["num-of-cylinders"])
```

```
df["num-of-cylinders"].value_counts()
```

```
1    16
2     7
0     6
3     1
Name: num-of-cylinders, dtype: int64
```

```
df=df.drop(["Lakhs","aspiration","normalized-losses","fuel-type"],axis=1)
```

```
df["stroke-bore-ratio"]=df["stroke"]/df["bore"]
```

```
df.columns
```

```
Index(['make', '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', 'SalesPrice', 'stroke-bore-ratio'],
      dtype='object')
```

```
df["Avg_City_Mpg_Per_Make"]=df.groupby("make")["city-mpg"].transform("mean")
```

```
df["Avg_City_Mpg_Per_Make"]
```

```
0     20.333333
1     20.333333
2     20.333333
3     18.857143
4     18.857143
5     18.857143
6     18.857143
7     18.857143
8     18.857143
9     18.857143
10    19.375000
11    19.375000
12    19.375000
13    19.375000
14    19.375000
15    19.375000
16    19.375000
17    19.375000
18    41.000000
19    41.000000
20    41.000000
21    28.000000
22    28.000000
23    28.000000
24    28.000000
25    28.000000
26    28.000000
27    28.000000
28    28.000000
```

```
29    28.000000
Name: Avg_City_Mpg_Per_Make, dtype: float64
```

```
df["Avg_Highway_Mpg_Per_Body_Style"]=df.groupby("body-style")["highway-mpg"].transform("mean")
```

```
df["Avg_Highway_Mpg_Per_Body_Style"]
```

```
0     27.000000
1     27.000000
2     35.000000
3     27.882353
4     27.882353
5     27.882353
6     27.882353
7     27.500000
8     27.882353
9     35.000000
10    27.882353
11    27.882353
12    27.882353
13    27.882353
14    27.882353
15    27.882353
16    27.882353
17    27.882353
18    35.000000
19    35.000000
20    27.882353
21    35.000000
22    35.000000
23    35.000000
24    35.000000
25    27.882353
26    27.882353
27    27.882353
28    27.500000
29    35.000000
Name: Avg_Highway_Mpg_Per_Body_Style, dtype: float64
```

```
# Adding a new feature "Avg_mpg" as it is a useful parameter for getting a good idea about the automobile's overall performance
df["Avg_mpg"]=(df["city-mpg"]+df["highway-mpg"])/2
```

```
df["Avg_mpg"]
```

```
0     24.0
1     24.0
2     22.5
3     27.0
4     20.0
5     22.0
6     22.0
7     22.0
8     18.5
9     19.0
10    26.0
11    26.0
12    24.5
13    24.5
14    22.5
15    19.0
16    19.0
17    17.5
18    50.0
19    40.5
20    40.5
21    39.0
22    34.5
23    27.0
24    34.5
25    34.5
26    34.5
27    27.0
28    27.0
29    21.5
Name: Avg_mpg, dtype: float64
```

```
df.shape
```

```
(30, 27)
```

```
# Shape of dataset before feature Engineering = (30,25)
# Shape of dataset after feature Engineering = (30,27)
```

Double-click (or enter) to edit

✓ 0s completed at 12:48 PM