In [19]:
```
import pandas as pd
import numpy as np
import matplotlib as mp
import seaborn as sns
```

In [240…
```
path='C:\\Users\\naman\\Downloads\\1621693814_135ebefb28731e37c622\\auto-mpg.
file1= pd.read_csv(path)
```

In [232…
```
file1
```

Out[232…

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino
…	…	…	…	…	…	…	…	…	…
393	27.0	4	140.0	86	2790	15.6	82	1	ford mustang gl
394	44.0	4	97.0	52	2130	24.6	82	2	vw pickup
395	32.0	4	135.0	84	2295	11.6	82	1	dodge rampage
396	28.0	4	120.0	79	2625	18.6	82	1	ford ranger
397	31.0	4	119.0	82	2720	19.4	82	1	chevy s-10

398 rows × 9 columns

In [241…
```
file1.head()
```

Out[241…

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
									malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst

In [233…
```
file1.index
```

Out[233…
```
RangeIndex(start=0, stop=398, step=1)
```

In [234…
```
file1.shape
```

Out[234…
```
(398, 9)
```

In [242…
```
file1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64  
 2   displacement  398 non-null    float64
 3   horsepower    398 non-null    object 
 4   weight        398 non-null    int64  
 5   acceleration  398 non-null    float64
 6   model year    398 non-null    int64  
 7   origin        398 non-null    int64  
 8   car name      398 non-null    object 
dtypes: float64(3), int64(4), object(2)
memory usage: 28.1+ KB
```

In [243…
```
file1['horsepower']
```

Out[243…
```
0      130
1      165
2      150
3      150
4      140
      ...
393     86
394     52
395     84
396     79
397     82
Name: horsepower, Length: 398, dtype: object
```

In [244...
```
##Data cleaning

file1=file1.replace('?',np.NaN)
file1["horsepower"].iloc[354]
#file1["horsepower"].iloc[336]
#file1["horsepower"].iloc[354]
```

Out[244... nan

In [245...
```
#Type conversion
file1["horsepower"]=file1["horsepower"].astype(float)
```

In [246...
```
file1
```

Out[246...

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130.0	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	1	ford torino
...	...	...	...	...	...	...	...	...	...
393	27.0	4	140.0	86.0	2790	15.6	82	1	ford mustang gl
394	44.0	4	97.0	52.0	2130	24.6	82	2	vw pickup
395	32.0	4	135.0	84.0	2295	11.6	82	1	dodge rampage
396	28.0	4	120.0	79.0	2625	18.6	82	1	ford ranger
397	31.0	4	119.0	82.0	2720	19.4	82	1	chevy s-10

398 rows × 9 columns

In [247...
```
file1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
```

```
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64
 2   displacement  398 non-null    float64
 3   horsepower    392 non-null    float64
 4   weight        398 non-null    int64
 5   acceleration  398 non-null    float64
 6   model year    398 non-null    int64
 7   origin        398 non-null    int64
 8   car name      398 non-null    object
dtypes: float64(4), int64(4), object(1)
memory usage: 28.1+ KB
```

In [248...

```
file1=file1.dropna()
```

In [249...

```
file1
```

Out[249...

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130.0	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	1	ford torino
...	...	...	...	...	...	...	...	...	...
393	27.0	4	140.0	86.0	2790	15.6	82	1	ford mustang gl
394	44.0	4	97.0	52.0	2130	24.6	82	2	vw pickup
395	32.0	4	135.0	84.0	2295	11.6	82	1	dodge rampage
396	28.0	4	120.0	79.0	2625	18.6	82	1	ford ranger
397	31.0	4	119.0	82.0	2720	19.4	82	1	chevy s-10

392 rows × 9 columns

In [250...

```
file1.describe()
```

Out[250...

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	
count	392.000000	392.000000	392.000000	392.000000	392.000000	392.000000	392.000000	392.
mean	23.445918	5.471939	194.411990	104.469388	2977.584184	15.541327	75.979592	1.
std	7.805007	1.705783	104.644004	38.491160	849.402560	2.758864	3.683737	0.
min	9.000000	3.000000	68.000000	46.000000	1613.000000	8.000000	70.000000	1.
25%	17.000000	4.000000	105.000000	75.000000	2225.250000	13.775000	73.000000	1.
50%	22.750000	4.000000	151.000000	93.500000	2803.500000	15.500000	76.000000	1.
75%	29.000000	8.000000	275.750000	126.000000	3614.750000	17.025000	79.000000	2.
max	46.600000	8.000000	455.000000	230.000000	5140.000000	24.800000	82.000000	3.

In [251...

```
file1.corr()
```

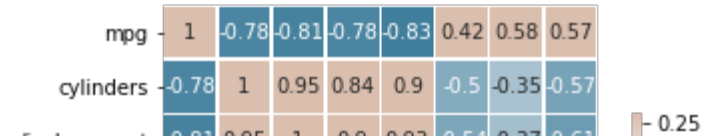
Out[251...

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	
mpg	1.000000	-0.777618	-0.805127	-0.778427	-0.832244	0.423329	0.580541	0.5
cylinders	-0.777618	1.000000	0.950823	0.842983	0.897527	-0.504683	-0.345647	-0.5
displacement	-0.805127	0.950823	1.000000	0.897257	0.932994	-0.543800	-0.369855	-0.6
horsepower	-0.778427	0.842983	0.897257	1.000000	0.864538	-0.689196	-0.416361	-0.4
weight	-0.832244	0.897527	0.932994	0.864538	1.000000	-0.416839	-0.309120	-0.5
acceleration	0.423329	-0.504683	-0.543800	-0.689196	-0.416839	1.000000	0.290316	0.2
model year	0.580541	-0.345647	-0.369855	-0.416361	-0.309120	0.290316	1.000000	0.1
origin	0.565209	-0.568932	-0.614535	-0.455171	-0.585005	0.212746	0.181528	1.0

In [252...

```
sns.heatmap(file1.corr(),cmap = sns.diverging_palette(230, 30, as_cmap=True),v
            square=True, linewidths=.5, cbar_kws={"shrink": .5}, annot=True)
```

Out[252... <AxesSubplot:>



In [253...
```
file1.drop('car name', inplace=True,axis=1)
```

```
C:\Users\naman\anaconda3\lib\site-packages\pandas\core\frame.py:4308: SettingW
ithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  return super().drop(
```

In [255...
```
file1
```

Out[255...

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin
0	18.0	8	307.0	130.0	3504	12.0	70	1
1	15.0	8	350.0	165.0	3693	11.5	70	1
2	18.0	8	318.0	150.0	3436	11.0	70	1
3	16.0	8	304.0	150.0	3433	12.0	70	1
4	17.0	8	302.0	140.0	3449	10.5	70	1
...	...	...	...	...	...	...	...	...
393	27.0	4	140.0	86.0	2790	15.6	82	1
394	44.0	4	97.0	52.0	2130	24.6	82	2
395	32.0	4	135.0	84.0	2295	11.6	82	1
396	28.0	4	120.0	79.0	2625	18.6	82	1
397	31.0	4	119.0	82.0	2720	19.4	82	1

392 rows × 8 columns

In [256...
```
y = file1.iloc[:,1].values
y
```

Out[256...
```
array([8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 4, 6, 6, 6, 4, 4, 4, 4,
       4, 4, 6, 8, 8, 8, 8, 4, 4, 4, 6, 6, 6, 6, 6, 8, 8, 8, 8, 8, 8, 8,
       6, 4, 6, 6, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 8, 8, 8, 8, 8,
       8, 8, 8, 8, 3, 8, 8, 8, 8, 4, 4, 4, 4, 4, 4, 4, 4, 4, 8, 8, 8, 8,
       8, 8, 8, 8, 8, 8, 8, 8, 6, 6, 6, 6, 6, 4, 8, 8, 8, 8, 6, 4, 4, 4,
       3, 4, 6, 4, 8, 8, 4, 4, 4, 4, 8, 4, 6, 8, 6, 6, 6, 4, 4, 4, 4, 6,
       6, 6, 8, 8, 8, 8, 8, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 6, 6, 6,
       8, 8, 8, 8, 6, 6, 6, 6, 6, 8, 8, 4, 4, 6, 4, 4, 4, 4, 6, 4, 6, 4,
       4, 4, 4, 4, 4, 4, 4, 4, 4, 8, 8, 8, 8, 6, 6, 6, 6, 4, 4, 4, 4, 6,
       6, 6, 6, 4, 4, 4, 4, 4, 8, 4, 6, 6, 8, 8, 8, 8, 4, 4, 4, 4, 4, 8,
       8, 8, 8, 6, 6, 6, 6, 8, 8, 8, 8, 4, 4, 4, 4, 4, 4, 4, 4, 6, 4, 3,
       4, 4, 4, 4, 4, 8, 8, 8, 6, 6, 6, 4, 6, 6, 6, 6, 6, 6, 8, 6, 8, 8,
       4, 4, 4, 4, 4, 4, 4, 4, 5, 6, 4, 6, 4, 4, 6, 6, 4, 6, 6, 8, 8, 8,
```

```
8, 8, 8, 8, 8, 4, 4, 4, 4, 5, 8, 4, 8, 4, 4, 4, 4, 4, 6, 6, 4, 4,
4, 4, 4, 4, 4, 4, 6, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 4, 4, 4, 4,
6, 3, 4, 4, 4, 4, 4, 6, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 6, 6, 6, 6, 8, 6, 6, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 6, 6, 4, 6, 4, 4, 4, 4, 4, 4, 4, 4])  dtype=int64)
```

In [257…
```
file1
```

Out[257…

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin
0	18.0	8	307.0	130.0	3504	12.0	70	1
1	15.0	8	350.0	165.0	3693	11.5	70	1
2	18.0	8	318.0	150.0	3436	11.0	70	1
3	16.0	8	304.0	150.0	3433	12.0	70	1
4	17.0	8	302.0	140.0	3449	10.5	70	1
...	...	...	...	...	...	...	...	...
393	27.0	4	140.0	86.0	2790	15.6	82	1
394	44.0	4	97.0	52.0	2130	24.6	82	2
395	32.0	4	135.0	84.0	2295	11.6	82	1
396	28.0	4	120.0	79.0	2625	18.6	82	1
397	31.0	4	119.0	82.0	2720	19.4	82	1

392 rows × 8 columns

In [258…
```
file2=file1.copy()
file2.drop('mpg', inplace=True,axis=1)
x = file2.iloc[:,:].values
x
```

Out[258…
```
array([[  8. , 307. , 130. , ...,  12. ,  70. ,   1. ],
       [  8. , 350. , 165. , ...,  11.5,  70. ,   1. ],
       [  8. , 318. , 150. , ...,  11. ,  70. ,   1. ],
       ...,
       [  4. , 135. ,  84. , ...,  11.6,  82. ,   1. ],
       [  4. , 120. ,  79. , ...,  18.6,  82. ,   1. ],
       [  4. , 119. ,  82. , ...,  19.4,  82. ,   1. ]])
```

In [259…
```
from sklearn.model_selection import train_test_split
```

In [279…
```
x_train,x_test,y_train,y_test = train_test_split(x,y,train_size=0.3,random_sta
```

In [280…
```
x_train.shape
```

Out[280…
```
(117, 7)
```

In [281…
```
y_train.shape
```

Out[281… (117,)

In [282…
```
x_test.shape
```

Out[282… (275, 7)

In [283…
```
y_test.shape
```

Out[283… (275,)

In [284…
```
from sklearn.linear_model import LinearRegression
```

In [285…
```
regressor = LinearRegression()
```

In [286…
```
regressor.fit(x_train,y_train)
```

Out[286… LinearRegression()

In [287…
```
y_predict = regressor.predict(x_test)
```

In [288…
```
y_predict
```

Out[288…
```
array([4., 4., 8., 4., 4., 6., 4., 4., 6., 4., 4., 8., 4., 8., 8., 4., 4.,
       8., 4., 4., 8., 6., 4., 6., 4., 8., 4., 4., 6., 8., 6., 4., 4., 4.,
       6., 4., 4., 8., 8., 8., 8., 4., 4., 8., 4., 8., 4., 8., 8., 6., 4.,
       8., 8., 4., 4., 6., 6., 6., 4., 4., 6., 4., 4., 8., 4., 6., 8., 4.,
       4., 4., 4., 4., 8., 8., 6., 8., 4., 8., 4., 4., 6., 4., 6., 4., 8.,
       8., 4., 4., 4., 3., 4., 8., 4., 6., 6., 4., 4., 4., 4., 4., 8., 4.,
       4., 4., 4., 6., 4., 8., 4., 8., 8., 4., 4., 8., 4., 4., 4., 6., 4.,
       6., 4., 4., 8., 6., 8., 8., 4., 4., 4., 4., 6., 6., 4., 4., 8., 4.,
       8., 6., 8., 4., 4., 4., 8., 4., 6., 4., 8., 4., 6., 8., 4., 6., 6.,
       6., 4., 6., 6., 6., 8., 8., 4., 8., 6., 4., 6., 4., 6., 4., 6., 6.,
       8., 8., 4., 6., 4., 4., 6., 4., 4., 4., 4., 8., 8., 4., 8., 6., 4.,
       8., 8., 8., 6., 4., 5., 6., 6., 6., 8., 6., 6., 6., 6., 4., 8., 4.,
       4., 4., 5., 8., 4., 6., 4., 4., 4., 8., 8., 3., 8., 4., 6., 6., 8.,
       6., 4., 6., 4., 4., 4., 8., 6., 4., 3., 4., 8., 6., 6., 8., 4., 4.,
       4., 4., 4., 4., 4., 4., 8., 6., 4., 8., 4., 8., 4., 8., 4., 8., 4.,
       4., 4., 6., 4., 8., 4., 6., 4., 4., 5., 4., 4., 4., 4., 4., 6., 8.,
       4., 8., 4.])
```

In [289…
```
y_test
```

Out[289…
```
array([4, 4, 8, 4, 4, 6, 4, 4, 6, 4, 4, 8, 4, 8, 8, 4, 4, 8, 4, 4, 8, 6,
       4, 6, 4, 8, 4, 4, 6, 8, 6, 4, 4, 4, 6, 4, 4, 8, 8, 8, 8, 4, 4, 8,
       4, 8, 4, 8, 8, 6, 4, 8, 8, 4, 4, 6, 6, 6, 4, 4, 6, 4, 4, 8, 4, 6,
       8, 4, 4, 4, 4, 4, 8, 8, 6, 8, 4, 8, 4, 4, 6, 4, 6, 4, 8, 8, 4, 4,
       4, 3, 4, 8, 4, 6, 6, 4, 4, 4, 4, 4, 8, 4, 4, 4, 4, 6, 4, 8, 4, 8,
```

```
8, 4, 4, 8, 4, 4, 4, 6, 4, 6, 4, 4, 8, 6, 8, 8, 4, 4, 4, 4, 6, 6,
4, 4, 8, 4, 8, 6, 8, 4, 4, 4, 8, 4, 6, 4, 8, 4, 6, 8, 4, 6, 6, 6,
4, 6, 6, 6, 8, 8, 4, 8, 6, 4, 6, 4, 6, 4, 6, 6, 8, 8, 4, 6, 4, 4,
6, 4, 4, 4, 4, 8, 8, 4, 8, 6, 4, 8, 8, 8, 6, 4, 5, 6, 6, 6, 8, 6,
6, 6, 6, 4, 8, 4, 4, 4, 5, 8, 4, 6, 4, 4, 4, 8, 8, 3, 8, 4, 6, 6,
8, 6, 4, 6, 4, 4, 4, 8, 6, 4, 3, 4, 8, 6, 6, 8, 4, 4, 4, 4, 4, 4,
4, 4, 8, 6, 4, 8, 4, 8, 4, 8, 4, 8, 4, 4, 4, 6, 4, 8, 4, 6, 4, 4,
5, 4, 4, 4, 4, 4, 6, 8, 4, 8, 4], dtype=int64)
```

In [290…
```
regressor.intercept_
```

Out[290… 2.220446049250313e-14

In [273…
```
regressor.coef_
```

Out[273…
```
array([ 1.00000000e+00,  1.65902511e-17,  1.90654205e-17, -1.45563237e-17,
       -2.72962599e-17, -5.74718043e-17,  3.33677444e-16])
```

In [291…
```
#Model Evaluation
from sklearn import metrics
```

In [292…
```
MAE = metrics.mean_absolute_error(y_test,y_predict)
MAE
```

Out[292… 3.5091121927424946e-15

In [293…
```
MSE = metrics.mean_squared_error(y_test,y_predict)
MSE
```

Out[293… 1.6236908868280843e-29

In [294…
```
RMSE = np.sqrt(MSE)
RMSE
```

Out[294… 4.0295047919416656e-15

In [295…
```
R2 = metrics.r2_score(y_test,y_predict)
R2
```

Out[295… 1.0

In [ ]: