```
import pandas as pd

import numpy as np

df =
pd.read_csv(r'https://raw.githubusercontent.com/Sarvesh-S-Patil/Dataset/main/MPG.csv')

df.head()
```

```
    mpg  cylinders  displacement  horsepower  weight  acceleration  \
0  18.0          8         307.0       130.0    3504          12.0
1  15.0          8         350.0       165.0    3693          11.5
2  18.0          8         318.0       150.0    3436          11.0
3  16.0          8         304.0       150.0    3433          12.0
4  17.0          8         302.0       140.0    3449          10.5

   model_year origin                       name
0          70    usa  chevrolet chevelle malibu
1          70    usa          buick skylark 320
2          70    usa         plymouth satellite
3          70    usa              amc rebel sst
4          70    usa                ford torino
```

```
df.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    object
 8   name          398 non-null    object
dtypes: float64(4), int64(3), object(2)
memory usage: 28.1+ KB
```

```
df.describe()
```

```
              mpg   cylinders  displacement  horsepower
weight  \
count  398.000000  398.000000    398.000000  392.000000   398.000000

mean    23.514573    5.454774    193.425879  104.469388  2970.424623

std      7.815984    1.701004    104.269838   38.491160   846.841774
```

```
min      9.000000    3.000000     68.000000   46.000000  1613.000000

25%     17.500000    4.000000    104.250000   75.000000  2223.750000

50%     23.000000    4.000000    148.500000   93.500000  2803.500000

75%     29.000000    8.000000    262.000000  126.000000  3608.000000

max     46.600000    8.000000    455.000000  230.000000  5140.000000


       acceleration  model_year
count    398.000000  398.000000
mean      15.568090   76.010050
std        2.757689    3.697627
min        8.000000   70.000000
25%       13.825000   73.000000
50%       15.500000   76.000000
75%       17.175000   79.000000
max       24.800000   82.000000

df.nunique()

mpg             129
cylinders         5
displacement     82
horsepower       93
weight          351
acceleration     95
model_year       13
origin            3
name            305
dtype: int64

df.corr()

                   mpg  cylinders  displacement  horsepower    weight
\
mpg           1.000000  -0.775396     -0.804203   -0.778427 -0.831741

cylinders    -0.775396   1.000000      0.950721    0.842983  0.896017

displacement -0.804203   0.950721      1.000000    0.897257  0.932824

horsepower   -0.778427   0.842983      0.897257    1.000000  0.864538

weight       -0.831741   0.896017      0.932824    0.864538  1.000000

acceleration  0.420289  -0.505419     -0.543684   -0.689196 -0.417457
```

```
model_year    0.579267  -0.348746     -0.370164   -0.416361 -0.306564
```

```
              acceleration  model_year
mpg               0.420289    0.579267
cylinders        -0.505419   -0.348746
displacement     -0.543684   -0.370164
horsepower       -0.689196   -0.416361
weight           -0.417457   -0.306564
acceleration      1.000000    0.288137
model_year        0.288137    1.000000
```

```
df=df.dropna()
```

```
df.info()
```

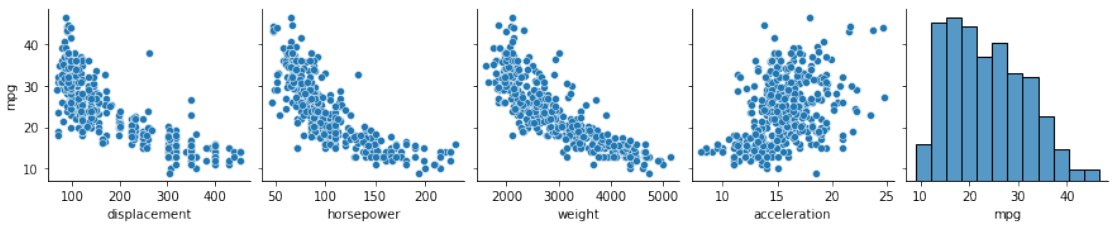
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 392 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   mpg           392 non-null    float64
 1   cylinders     392 non-null    int64  
 2   displacement  392 non-null    float64
 3   horsepower    392 non-null    float64
 4   weight        392 non-null    int64  
 5   acceleration  392 non-null    float64
 6   model_year    392 non-null    int64  
 7   origin        392 non-null    object 
 8   name          392 non-null    object 
dtypes: float64(4), int64(3), object(2)
memory usage: 30.6+ KB
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
sns.pairplot(df, x_vars =
['displacement','horsepower','weight','acceleration','mpg'], y_vars=
['mpg'])
```

```
<seaborn.axisgrid.PairGrid at 0x7f61c65f73d0>
```



```
df.columns
```

```
Index(['mpg', 'cylinders', 'displacement', 'horsepower', 'weight',
       'acceleration', 'model_year', 'origin', 'name'],
      dtype='object')
```

```
y = df['mpg']
```

```
y.shape
```

```
(392,)
```

```
X= df[['displacement','horsepower','weight','acceleration']]
```

```
X.shape
```

```
(392, 4)
```

```
from sklearn.preprocessing import StandardScaler
```

```
ss= StandardScaler()
```

```
X =ss.fit_transform(X)
```

```
pd.DataFrame(X).describe()
```

```
                  0             1             2             3
count  3.920000e+02  3.920000e+02  3.920000e+02  3.920000e+02
mean  -2.537653e-16 -4.392745e-16  5.607759e-17  6.117555e-16
std    1.001278e+00  1.001278e+00  1.001278e+00  1.001278e+00
min   -1.209563e+00 -1.520975e+00 -1.608575e+00 -2.736983e+00
25%   -8.555316e-01 -7.665929e-01 -8.868535e-01 -6.410551e-01
50%   -4.153842e-01 -2.853488e-01 -2.052109e-01 -1.499869e-02
75%    7.782764e-01  5.600800e-01  7.510927e-01  5.384714e-01
max    2.493416e+00  3.265452e+00  2.549061e+00  3.360262e+00
```

```
from sklearn.model_selection import train_test_split
```

```
X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.3,random_state=2529)
```

```
X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

```
((274, 4), (118, 4), (274,), (118,))
```

```
from sklearn.linear_model import LinearRegression
```

```
lr = LinearRegression()
```

```
lr.fit(X_train,y_train)
```

```
LinearRegression()
```

```
y_pred = lr.predict(X_test)
```

```
y_pred.shape
```

```
(118,)
```

```
lr.intercept_
```

```
23.485738559737584
```

```
lr.coef_
```

```
array([-1.05767743, -1.68734727, -4.10787617, -0.11495177])
```

```
from sklearn.metrics import
mean_absolute_error,mean_absolute_percentage_error,r2_score
```

```
mean_absolute_error(y_test,y_pred)
```

```
3.3286968643244106
```

```
mean_absolute_percentage_error(y_test,y_pred)
```

```
0.14713035779536746
```

```
r2_score(y_test,y_pred)
```

```
0.7031250746717692
```