```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import warnings
        warnings.filterwarnings("ignore")
```

Exploratory Data Analysis

```
In [2]: data=pd.read_csv("/home/placement/Downloads/Advertising.csv")
```

```
In [3]: data.describe()
```

Out[3]:

	Unnamed: 0	TV	radio	newspaper	sales
count	200.000000	200.000000	200.000000	200.000000	200.000000
mean	100.500000	147.042500	23.264000	30.554000	14.022500
std	57.879185	85.854236	14.846809	21.778621	5.217457
min	1.000000	0.700000	0.000000	0.300000	1.600000
25%	50.750000	74.375000	9.975000	12.750000	10.375000
50%	100.500000	149.750000	22.900000	25.750000	12.900000
75%	150.250000	218.825000	36.525000	45.100000	17.400000
max	200.000000	296.400000	49.600000	114.000000	27.000000

In [4]: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  200 non-null    int64
 1   TV          200 non-null    float64
 2   radio       200 non-null    float64
 3   newspaper   200 non-null    float64
 4   sales       200 non-null    float64
dtypes: float64(4), int64(1)
memory usage: 7.9 KB
```

In [5]: `data.head(10)`

Out[5]:

	Unnamed: 0	TV	radio	newspaper	sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9
5	6	8.7	48.9	75.0	7.2
6	7	57.5	32.8	23.5	11.8
7	8	120.2	19.6	11.6	13.2
8	9	8.6	2.1	1.0	4.8
9	10	199.8	2.6	21.2	10.6

In [6]: `data.shape`

Out[6]: (200, 5)

In [7]:
```python
data.isna().sum()
```

Out[7]:
```
Unnamed: 0    0
TV            0
radio         0
newspaper     0
sales         0
dtype: int64
```

In [8]:
```python
data1=data.drop(['Unnamed: 0'],axis=1)
data1
```

Out[8]:

	TV	radio	newspaper	sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9
...	...	...	...	...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	9.7
197	177.0	9.3	6.4	12.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	13.4

200 rows × 4 columns

In [9]:
```python
y=data1['sales']
x=data1.drop('sales',axis=1)
```

In [10]: y

Out[10]: 0   22.1
         1   10.4
         2   9.3
         3   18.5
         4   12.9
                  ...
         195   7.6
         196   9.7
         197   12.8
         198   25.5
         199   13.4
         Name: sales, Length: 200, dtype: float64

In [11]:
```python
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [12]:
```python
x_test.head(5)
```

Out[12]:

	TV	radio	newspaper
95	163.3	31.6	52.9
15	195.4	47.7	52.9
30	292.9	28.3	43.2
158	11.7	36.9	45.2
128	220.3	49.0	3.2

In [13]: `y_test.head(5)`

Out[13]:
```
95      16.9
15      22.4
30      21.4
158      7.3
128     24.7
Name: sales, dtype: float64
```

In [14]: `x_train.head(5)`

Out[14]:

	TV	radio	newspaper
42	293.6	27.7	1.8
189	18.7	12.1	23.4
90	134.3	4.9	9.3
136	25.6	39.0	9.3
51	100.4	9.6	3.6

In [15]: `y_train.head(5)`

Out[15]:
```
42      20.7
189      6.7
90      11.2
136      9.5
51      10.7
Name: sales, dtype: float64
```

```
In [16]:  from sklearn.model_selection import GridSearchCV
          from sklearn.linear_model import Lasso
          lasso=Lasso()
          parameters={'alpha':[1e-15,1e-10,1e-8,1e-4,1e-3,1e-2,1,5,10,20,30]}
          lasso_regressor=GridSearchCV(lasso,parameters)
          lasso_regressor.fit(x_train,y_train)
```

```
Out[16]:  GridSearchCV(estimator=Lasso(),
                       param_grid={'alpha': [1e-15, 1e-10, 1e-08, 0.0001, 0.001, 0.01, 1,
                                             5, 10, 20, 30]})
```

```
In [17]:  lasso_regressor.best_params_
```

```
Out[17]:  {'alpha': 1}
```

```
In [18]:  lasso=Lasso(alpha=1)
          lasso.fit(x_train,y_train)
          y_pred_lasso=lasso.predict(x_test)
```

```
In [19]:  from sklearn.metrics import r2_score
          r2_score(y_test,y_pred_lasso)
```

```
Out[19]:  0.8589079527148957
```

```
In [20]:  from sklearn.metrics import mean_squared_error
          Lasso_Error=mean_squared_error(y_pred_lasso,y_test)
          Lasso_Error
```

```
Out[20]:  3.641439660278575
```

```
In [ ]:
```