In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

Exploratory Data Analysis

In [2]:

```
#load data
data=pd.read_csv("/home/placement/Desktop/BhanuSiva4K8/Advertising.csv")
data.head()
```

Out[2]:

	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

In [3]:

```
list(data)
```

Out[3]:

```
['Unnamed: 0', 'TV', 'radio', 'newspaper', 'sales']
```

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

Out[4]:

	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 [5]:
```
data.isna().sum()
```

Out[5]:
```
Unnamed: 0    0
TV            0
radio         0
newspaper     0
sales         0
dtype: int64
```

In [6]:
```
data=data.drop(['Unnamed: 0'],axis=1)
data
```

Out[6]:

	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 [7]:
```
list(data)
```

Out[7]:
```
['TV', 'radio', 'newspaper', 'sales']
```

In [8]:
```
y=data['sales']
x=data.drop('sales',axis=1)
```

In [9]:

```
x
```

Out[9]:

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

200 rows × 3 columns

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]:
```
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]:
```
from sklearn.linear_model import Lasso
reg=Lasso()
reg.fit(x_train,y_train)
```

Out[12]:
```
▾ Lasso

Lasso()
```

In [13]:
```
y_pred=reg.predict(x_test)
y_pred
```

Out[13]:
```
array([16.52391951, 21.05821853, 21.624966  , 10.7457245 , 22.18826932,
       13.24310226, 21.16115506,  7.45487482, 13.54176537, 15.19735958,
        9.05895919,  6.64726211, 14.41534178,  8.94924498,  9.65557133,
       12.16355273,  8.83613562, 16.27480625, 10.32558284, 18.84483264,
       19.71142443, 13.51103813, 12.41920349, 21.46659417,  7.77792701,
        5.7752475 , 20.87123457, 11.95956215,  9.18798248,  8.52886947,
       12.41703422, 10.02587396, 21.57497619, 12.493691  , 18.27719607,
       20.16422674, 14.15387342, 20.96077769, 10.98977443,  4.53493832,
        9.66985435, 12.62189921, 10.13425798,  8.19584981, 13.43210386,
        5.38482786,  9.32599787, 14.16911116,  8.82848335, 11.68070192,
       15.57033379, 11.84410473, 13.06276807, 10.98222598,  6.53498206,
        9.9204947 ,  9.51373461, 24.21697767,  7.72986857, 12.42107899,
       17.65952642, 15.30134052, 11.47270716, 11.0350857 , 16.7024047 ,
        6.95562135])
```

In [14]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

Out[14]:
```
0.8589079527148957
```

In [15]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test,y_pred)
```

Out[15]:
```
3.641439660278575
```

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]}

lasso_regressor = GridSearchCV(lasso, parameters)

lasso_regressor.fit(x_train, y_train)
```

Out[16]:

```
▸   GridSearchCV
▸ estimator: Lasso
      ▸ Lasso
```

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]:

```
r2_score(y_test,y_pred_lasso)
```

Out[19]:

```
0.8589079527148957
```

In [20]:

```
Lasso_Error=mean_squared_error(y_pred_lasso,y_test)
Lasso_Error
```

Out[20]:

```
3.641439660278575
```

In [21]:

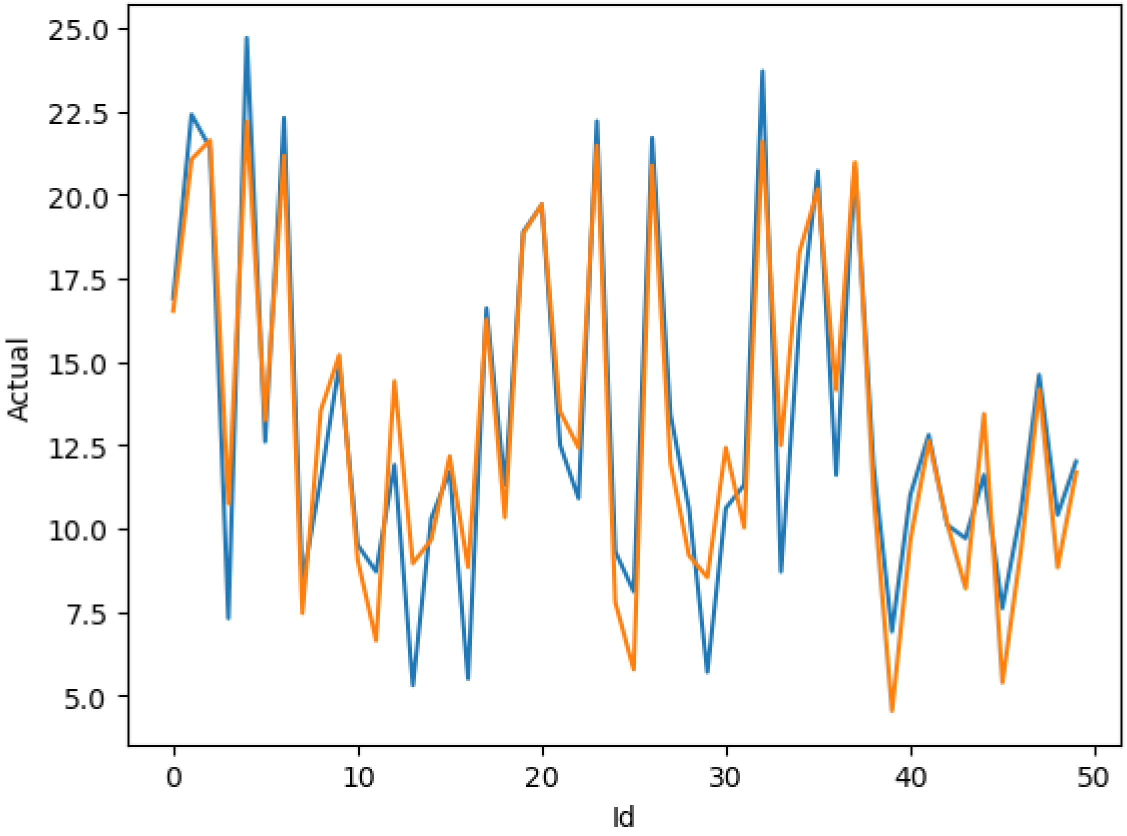
```
Results=pd.DataFrame(columns=['Actual','predicted'])
Results['Actual']=y_test
Results['predicted']=y_pred_lasso
#Result['km']=x_test['km']
Results=Results.reset_index()
Results['Id']=Results.index
Results.head(10)
```

Out[21]:

	index	Actual	predicted	Id
0	95	16.9	16.523920	0
1	15	22.4	21.058219	1
2	30	21.4	21.624966	2
3	158	7.3	10.745724	3
4	128	24.7	22.188269	4
5	115	12.6	13.243102	5
6	69	22.3	21.161155	6
7	170	8.4	7.454875	7
8	174	11.5	13.541765	8
9	45	14.9	15.197360	9

```
In [22]: import seaborn as sns
         import matplotlib.pyplot as plt
         sns.lineplot(x='Id',y='Actual',data=Results.head(50))
         sns.lineplot(x='Id',y='predicted',data=Results.head(50))
         plt.plot()

Out[22]: []
```



In [ ]: