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/placemnet/YUVA/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]:
```
a=data['sales']
b=data.drop(['sales','Unnamed: 0'],axis=1)
b
```

Out[3]:

	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 [4]:
```
from sklearn.model_selection import train_test_split
a_train,a_test,b_train,b_test=train_test_split(a,b,test_size=0.1,random_state=42)
b_test
```

Out[4]:

	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
115	75.1	35.0	52.7
69	216.8	43.9	27.2
170	50.0	11.6	18.4
174	222.4	3.4	13.1
45	175.1	22.5	31.5
66	31.5	24.6	2.2
182	56.2	5.7	29.7
165	234.5	3.4	84.8
78	5.4	29.9	9.4
186	139.5	2.1	26.6
177	170.2	7.8	35.2
56	7.3	28.1	41.4
152	197.6	23.3	14.2
82	75.3	20.3	32.5
68	237.4	27.5	11.0

Lasso

In [5]:
```
from sklearn.linear_model import Lasso
from sklearn.model_selection import GridSearchCV

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(b_train, a_train)
```

Out[5]:

▾ **GridSearchCV**

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

▾ **estimator: Lasso**

Lasso()

▾ Lasso

Lasso()

In [6]:
```
lasso_regressor.best_params_
```

Out[6]:
```
{'alpha': 1}
```

In [7]:
```
lasso=Lasso(alpha=1)
lasso.fit(b_train,a_train)
a_pred_lasso=lasso.predict(b_test)
```

In [8]:
```
from sklearn.metrics import r2_score    #to check the efficiency
r2_score(a_test,a_pred_lasso)
```

Out[8]:
```
0.903778595031101
```

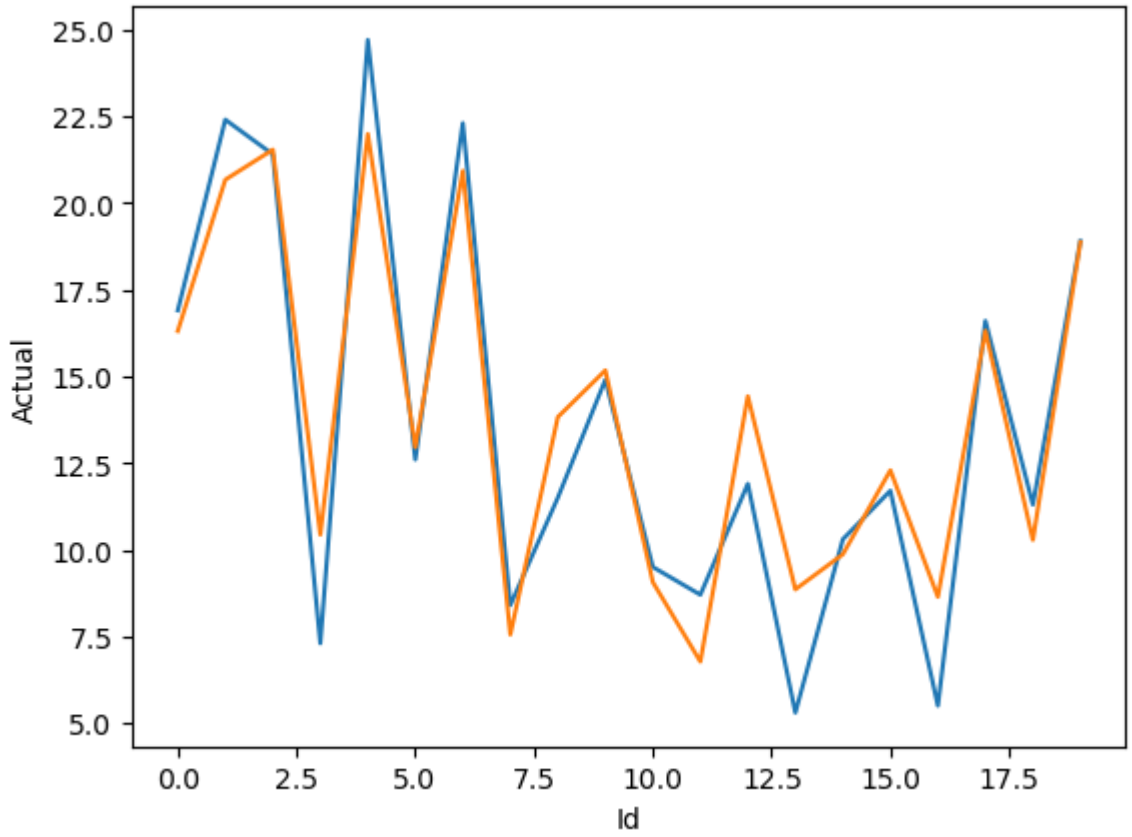
In [9]:
```
results=pd.DataFrame(columns=['Actual','Predicted'])   #to compare the actual and pedicted price
results['Actual']=a_test
results['Predicted']=a_pred_lasso
results=results.reset_index()
results['Id']=results.index
results.head()
```

Out[9]:

	index	Actual	Predicted	Id
0	95	16.9	16.305758	0
1	15	22.4	20.668405	1
2	30	21.4	21.534602	2
3	158	7.3	10.435192	3
4	128	24.7	21.985764	4

In [10]:
```
import seaborn as sns
sns.lineplot(x='Id',y='Actual',data=results.head(50))
sns.lineplot(x='Id',y='Predicted',data=results.head(50))
plt.plot()
```

Out[10]: []



In [ ]: