In [39]:

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

exploratory data analysis

In [40]:

```
data=pd.read_csv("/home/placement/Downloads/Advertising.csv")
```

In [41]:

```
data.head()
```

Out[41]:

	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

Lasso

In [42]:

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

In [43]:

```
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 [44]:

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

Out[44]:

```
Lasso()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [45]:

```
y_pred=reg.predict(x_test)
```

In [46]:

```
y_pred
```

Out[46]:

```
array([16.52356688, 21.07183096, 21.63653184, 10.73467623, 22.18465223,
       13.23922119, 21.16664756,  7.4431686 , 13.52971188, 15.20686921,
        9.06623028,  6.63289843, 14.40121642,  8.95402791,  9.6407048 ,
       12.14981955,  8.84314265, 16.26648664, 10.32853903, 18.85137298,
       19.70493689, 13.5213463 , 12.41160087, 21.46562312,  7.78549068,
        5.78265455, 20.87376833, 11.96631497,  9.18377863,  8.5245863 ,
       12.43342356, 10.04074515, 21.58122748, 12.49489552, 18.26811877,
       20.16474367, 14.14870494, 20.95277886, 10.97979616,  4.53957393,
        9.67385982, 12.60613272, 10.14475068,  8.20968861, 13.42902186,
        5.36899823,  9.33781109, 14.18378647,  8.82071962, 11.68378102,
       15.56752612, 11.83221277, 13.07693412, 10.97649377,  6.53276607,
        9.92514199,  9.50755099, 24.21620593,  7.71785174, 12.42302645,
       17.64869313, 15.30224537, 11.48429434, 11.04853032, 16.71492188,
        6.9526154 ])
```

In [47]:

```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_lasso)
```

Out[47]:

```
0.8589177083282906
```

In [48]:

```
from sklearn.metrics import mean_squared_error
lasso_Error=mean_squared_error(y_pred_lasso,y_test)
lasso_Error
```

Out[48]:

```
3.6411878779973614
```

In [49]:

```python
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[49]:

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

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [50]:

```python
lasso_regressor.best_params_
```

Out[50]:

```
{'alpha': 1}
```

In [51]:

```python
lasso=Lasso(alpha=1)
lasso.fit(x_train,y_train)
y_pred_lasso=lasso.predict(x_test)
```

In [52]:

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

Out[52]:

```
0.8589177083282906
```

In [53]:

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

Out[53]:

```
3.6411878779973614
```

In [ ]:

```python

```

In [55]:

```
Results=pd.DataFrame(columns=['Actual','Predicted'])
Results['Actual']=y_test
Results['Predicted']=y_pred_lasso
Results=Results.reset_index()
Results['ID']=Results.index
Results.head(10)
```

Out[55]:

	index	Actual	Predicted	ID
0	95	16.9	16.523567	0
1	15	22.4	21.071831	1
2	30	21.4	21.636532	2
3	158	7.3	10.734676	3
4	128	24.7	22.184652	4
5	115	12.6	13.239221	5
6	69	22.3	21.166648	6
7	170	8.4	7.443169	7
8	174	11.5	13.529712	8
9	45	14.9	15.206869	9

In [56]:

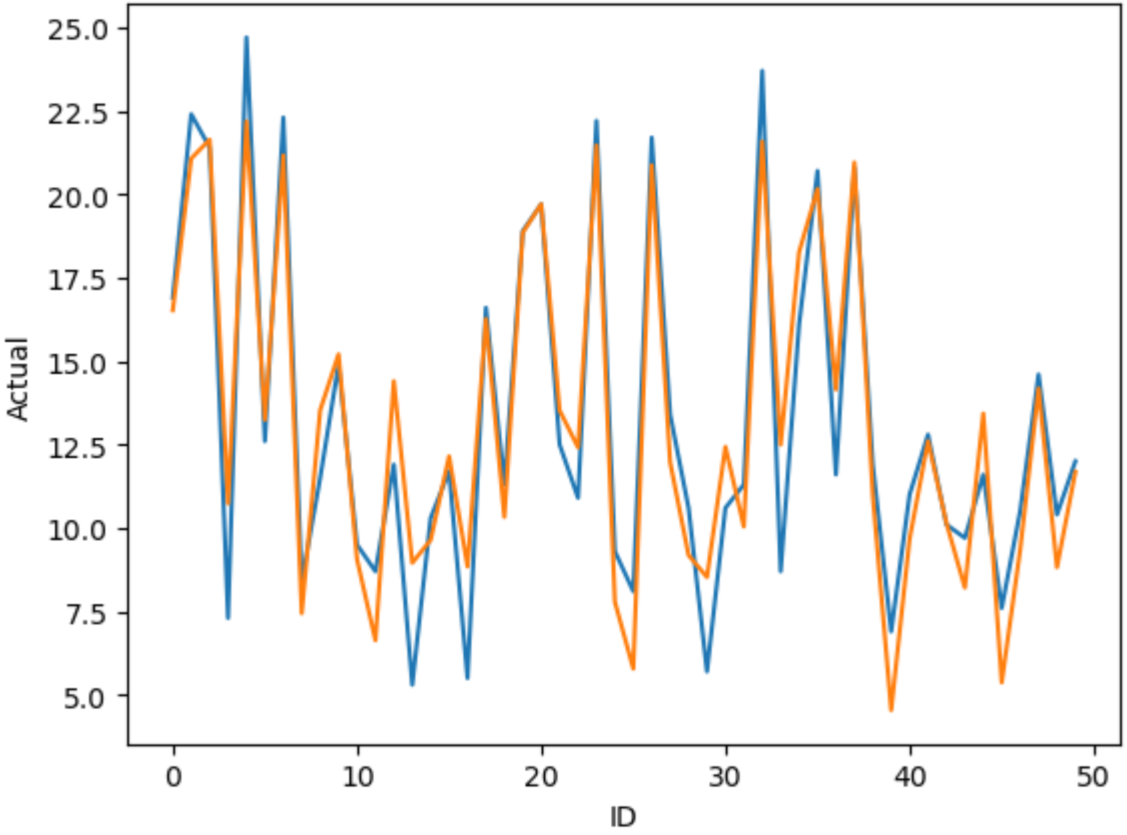
```
import seaborn  as sns
import matplotlib.pyplot as plt
```

In [57]:

```
sns.lineplot(x='ID',y='Actual',data=Results.head(50))
sns.lineplot(x='ID',y='Predicted',data=Results.head(50))
plt.plot()
```

Out[57]:

```
[]
```



In [ ]:

```

```