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

In [2]:
```
data=pd.read_csv("/home/placement/Desktop/usha g1/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.head()
```

Out[4]:

	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 [5]:
```
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 [6]:
```
data1=data.drop(['Unnamed: 0'],axis=1)
data1
```

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

In [8]:
```
y
```

Out[8]:
```
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 [9]:
```
list(x)
```

Out[9]:
```
['TV', 'radio', 'newspaper']
```

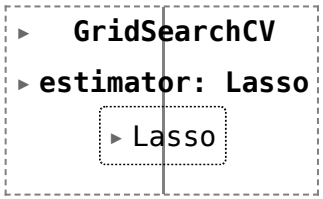
In [10]:
```
from sklearn.model_selection import train_test_split
x_train, x_test,y_train, y_test = train_test_split(x, y, test_size=0.33,random_state=42)
```

In [11]:
```
x_train.shape
```

Out[11]:
```
(134, 3)
```

In [12]:
```
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(x_train, y_train)
```

Out[12]:



In [13]:
```
lasso_regressor.best_params_
```

Out[13]:
```
{'alpha': 1}
```

In [14]:
```
lasso=Lasso(alpha=0.1)
lasso.fit(x_train,y_train)
y_pred_lasso=lasso.predict(x_test)
```

In [15]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_lasso)
```

Out[15]:
```
0.8559136390952934
```

In [16]:
```
from sklearn.metrics import mean_squared_error
lasso_Error=mean_squared_error(y_pred_lasso,y_test)
lasso_Error
```

Out[16]:
```
3.718719794627319
```

In [17]:
```
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[17]:

	index	Actual	Predicted	Id
0	95	16.9	16.580451	0
1	15	22.4	21.173432	1
2	30	21.4	21.663263	2
3	158	7.3	10.804369	3
4	128	24.7	22.245736	4
5	115	12.6	13.307456	5
6	69	22.3	21.231000	6
7	170	8.4	7.391095	7
8	174	11.5	13.449902	8
9	45	14.9	15.194742	9

In [18]:
```
sns.lineplot(x='Id',y='Actual',data=Results.head(50))
sns.lineplot(x='Id',y='Predicted',data=Results.head(50))
plt.plot()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[18], line 1
----> 1 sns.lineplot(x='Id',y='Actual',data=Results.head(50))
      2 sns.lineplot(x='Id',y='Predicted',data=Results.head(50))
      3 plt.plot()

NameError: name 'sns' is not defined
```

In [ ]: