import libraries

In [1]:

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

Exploratory Data Analysis

In [2]:

```
d=r.read_csv("/home/placement/Downloads/Advertising.csv")
d
```

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
...	...	...	...	...	...
195	196	38.2	3.7	13.8	7.6
196	197	94.2	4.9	8.1	9.7
197	198	177.0	9.3	6.4	12.8
198	199	283.6	42.0	66.2	25.5
199	200	232.1	8.6	8.7	13.4

In [3]:

```
#splitting the data to predict
y=d['sales']
x=d.drop(['sales','Unnamed: 0'],axis=1)
```

In [4]:

```
#spliting data to create the model
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 [5]:

```
#Lasso regression model
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, 20,30]}
lasso_regressor = GridSearchCV(lasso,parameters)
lasso_regressor.fit(x_train,y_train)
```

Out[5]:

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

In [6]:

```
lasso_regressor.best_params_
```

Out[6]:

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

In [7]:

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

In [8]:

```
from sklearn.metrics import r2_score #to  check the efficiency
r2_score(y_test,y_pred_lasso)
```

Out[8]:

```
0.8589079527148957
```

In [9]:

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

Out[9]:

```
3.641439660278575
```

In [10]:

```
results=r.DataFrame(columns=['Actual','Predicted']) #To compare the actual and predicted price
results['Actual']=y_test
results['Predicted']=y_pred_lasso
results=results.reset_index()
results['Id']=results.index
results
```

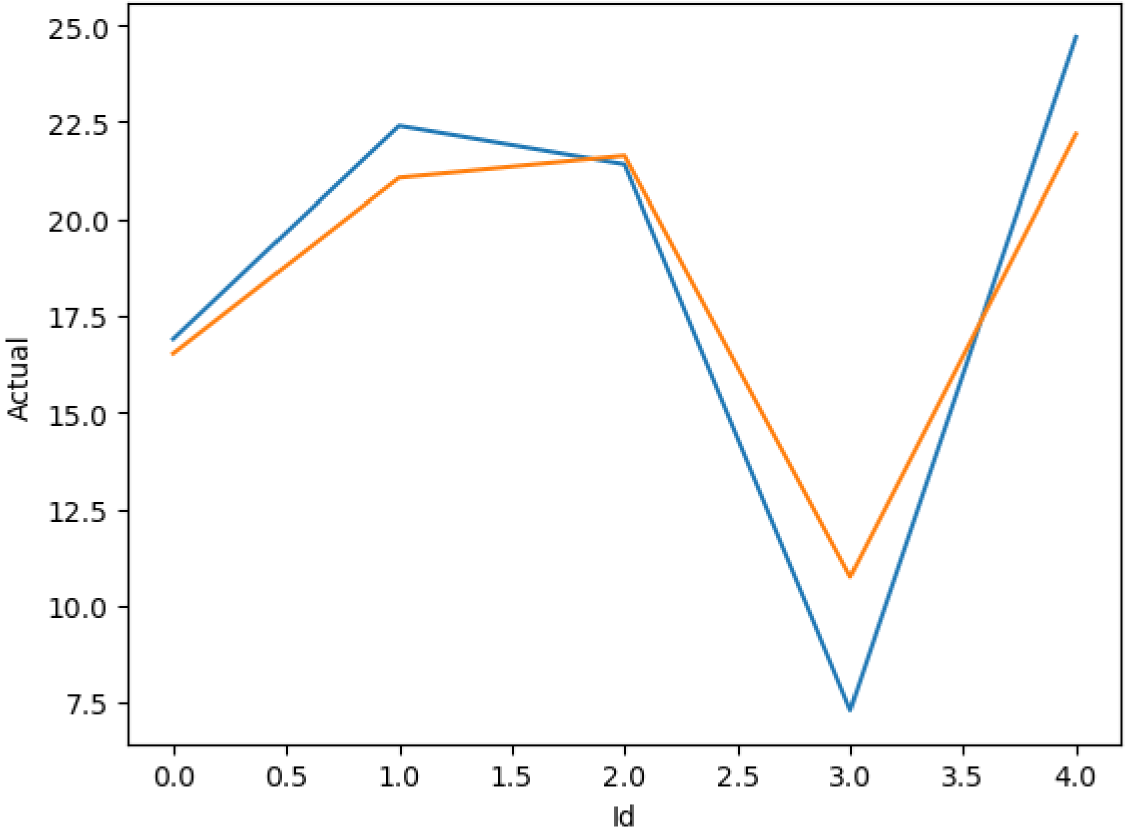
Out[10]:

	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
...	...	...	...	...
61	97	15.5	15.301341	61
62	31	11.9	11.472707	62
63	12	9.2	11.035086	63
64	35	12.8	16.702405	64
65	119	6.6	6.955621	65

66 rows × 4 columns

```
In [11]: import seaborn as sns
         import matplotlib.pyplot as plt
         sns.lineplot(x='Id',y='Actual',data=results.head(5))
         sns.lineplot(x='Id',y='Predicted',data=results.head(5))
         plt.plot()
```

```
Out[11]: []
```



In [ ]: