In [1]:
```
# Use Company dataset and apply ridge regression using the best value for penalty
```

In [2]:
```
import pandas as pd
```

In [3]:
```
df= pd.read_csv("company.csv")
df
```

Out[3]:

	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	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9
...	...	...	...	...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	14.0
197	177.0	9.3	6.4	14.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	18.4

200 rows × 4 columns

In [4]:
```
x=df.iloc[:,:-1]
x
```

Out[4]:

	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 [5]:
```
y=df.iloc[:,-1]
y
```

Out[5]:
```
0      22.1
1      10.4
2      12.0
3      16.5
4      17.9
       ...
195     7.6
196    14.0
197    14.8
198    25.5
199    18.4
Name: Sales, Length: 200, dtype: float64
```

In [8]:
```
from sklearn.linear_model import Ridge
model=Ridge()
```

In [19]:
```
from sklearn.preprocessing import MinMaxScaler
sc= MinMaxScaler()
x_sc=sc.fit_transform(x)
```

In [11]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.25, random_state=1)
model.fit(xtrain,ytrain)
```

Out[11]:
```
Ridge()
```

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 [15]:
```
from sklearn.model_selection import RepeatedKFold
```

In [18]:
```
cv = RepeatedKFold(n_splits = 10, n_repeats = 3, random_state = 1)
```

In [20]:
```
from sklearn.metrics import r2_score
ypred = model.predict(xtest)
r2_score(ytest, ypred)
```

Out[20]:
```
0.9048906885507663
```

In [22]:
```
model1= Ridge()
parms = {'alpha':[0.00001,0.0001,0.001,0.01]}
from sklearn.model_selection import GridSearchCV
search = GridSearchCV(model1, parms, cv=cv)
result = search.fit(x_sc, y)
result.best_params_
```

Out[22]:
```
{'alpha': 0.01}
```

In [23]:
```
model2 = Ridge(alpha=0.01)
model2.fit(xtrain, ytrain)
```

Out[23]:
```
Ridge(alpha=0.01)
```

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 [24]:
```
ypred2 = model2.predict(xtest)
r2_score(ytest,ypred2)
```

Out[24]:
```
0.9048917137810413
```

In [ ]:
```

```