In [1]:
```
import pandas as pd
df=pd.read_csv("homeprices.csv")
df
```

Out[1]:

	town	area	price
0	monroe township	2600	550000
1	monroe township	3000	565000
2	monroe township	3200	610000
3	monroe township	3600	680000
4	monroe township	4000	725000
5	west windsor	2600	585000
6	west windsor	2800	615000
7	west windsor	3300	650000
8	west windsor	3600	710000
9	robinsville	2600	575000
10	robinsville	2900	600000
11	robinsville	3100	620000
12	robinsville	3600	695000

In [3]:
```
dummies=pd.get_dummies(df.town)
```

In [5]:
```
merged=pd.concat([df,dummies],axis=1)
merged
```

Out[5]:

	town	area	price	monroe township	robinsville	west windsor
0	monroe township	2600	550000	1	0	0
1	monroe township	3000	565000	1	0	0
2	monroe township	3200	610000	1	0	0
3	monroe township	3600	680000	1	0	0
4	monroe township	4000	725000	1	0	0
5	west windsor	2600	585000	0	0	1
6	west windsor	2800	615000	0	0	1
7	west windsor	3300	650000	0	0	1
8	west windsor	3600	710000	0	0	1
9	robinsville	2600	575000	0	1	0
10	robinsville	2900	600000	0	1	0
11	robinsville	3100	620000	0	1	0
12	robinsville	3600	695000	0	1	0

In [7]:
```
final = merged.drop(['town','west windsor'],axis='columns')
final
```

Out[7]:

	area	price	monroe township	robinsville
0	2600	550000	1	0
1	3000	565000	1	0
2	3200	610000	1	0
3	3600	680000	1	0
4	4000	725000	1	0
5	2600	585000	0	0
6	2800	615000	0	0
7	3300	650000	0	0
8	3600	710000	0	0
9	2600	575000	0	1
10	2900	600000	0	1
11	3100	620000	0	1
12	3600	695000	0	1

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

In [9]:
```
X = final.drop(['price'],axis='columns')
X
```

Out[9]:

	area	monroe township	robinsville
0	2600	1	0
1	3000	1	0
2	3200	1	0
3	3600	1	0
4	4000	1	0
5	2600	0	0
6	2800	0	0
7	3300	0	0
8	3600	0	0
9	2600	0	1
10	2900	0	1
11	3100	0	1
12	3600	0	1

In [10]:
```
y=final.price
y
```

Out[10]:
```
0     550000
1     565000
2     610000
3     680000
4     725000
5     585000
6     615000
7     650000
8     710000
9     575000
10    600000
11    620000
12    695000
Name: price, dtype: int64
```

In [11]:
```
model.fit(X,y)
```

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

In [14]:
```
import numpy as np
model.predict(np.array([[2800,0,1]]))
```

```
C:\Program Files\Python311\Lib\site-packages\sklearn\base.py:439: UserWa
rning: X does not have valid feature names, but LinearRegression was fit
ted with feature names
  warnings.warn(
```

Out[14]:
```
array([590775.63964739])
```

In [16]:
```
model.predict(np.array([[3400,0,0]]))
```

```
C:\Program Files\Python311\Lib\site-packages\sklearn\base.py:439: UserWa
rning: X does not have valid feature names, but LinearRegression was fit
ted with feature names
  warnings.warn(
```

Out[16]:
```
array([681241.66845839])
```

In [17]:
```
model.score(X,y)
```

Out[17]:
```
0.9573929037221872
```

In [18]:
```
df
```

Out[18]:

	town	area	price
0	monroe township	2600	550000
1	monroe township	3000	565000
2	monroe township	3200	610000
3	monroe township	3600	680000
4	monroe township	4000	725000
5	west windsor	2600	585000
6	west windsor	2800	615000
7	west windsor	3300	650000
8	west windsor	3600	710000
9	robinsville	2600	575000
10	robinsville	2900	600000
11	robinsville	3100	620000
12	robinsville	3600	695000

In [19]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

In [21]:
```
dfle = df
dfle.town=le.fit_transform(dfle.town)
df
```

Out[21]:

	town	area	price
0	0	2600	550000
1	0	3000	565000
2	0	3200	610000
3	0	3600	680000
4	0	4000	725000
5	2	2600	585000
6	2	2800	615000
7	2	3300	650000
8	2	3600	710000
9	1	2600	575000
10	1	2900	600000
11	1	3100	620000
12	1	3600	695000

In [24]:
```
X= df[['town','area']].values
X
```

Out[24]:
```
array([[   0, 2600],
       [   0, 3000],
       [   0, 3200],
       [   0, 3600],
       [   0, 4000],
       [   2, 2600],
       [   2, 2800],
       [   2, 3300],
       [   2, 3600],
       [   1, 2600],
       [   1, 2900],
       [   1, 3100],
       [   1, 3600]], dtype=int64)
```

In [25]:
```
y= dfle.price
```

In [31]:
```
from sklearn.preprocessing import OneHotEncoder
ohe =OneHotEncoder(categorical_features=[0])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call la
st)
Cell In[31], line 2
      1 from sklearn.preprocessing import OneHotEncoder
----> 2 ohe =OneHotEncoder(categorical_features=[0])

TypeError: OneHotEncoder.__init__() got an unexpected keyword argument
'categorical_features'
```

In [30]:
```
X=ohe.fit_transform(X).toarray()
X
```

Out[30]:
```
array([[0., 1., 1., 0., 1., 0., 0., 1., 1., 0., 1., 0., 1., 0., 1., 0.,
        1., 0., 1., 0., 1., 0., 1., 0.],
       [0., 1., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0., 0., 1., 1., 0.,
        1., 0., 1., 0., 1., 0., 1., 0.],
       [0., 1., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0.,
        0., 1., 1., 0., 1., 0., 1., 0.],
       [0., 1., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0.,
        1., 0., 1., 0., 0., 1., 1., 0.],
       [0., 1., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0.,
        1., 0., 1., 0., 1., 0., 0., 1.],
       [1., 0., 1., 0., 0., 1., 0., 1., 1., 0., 1., 0., 1., 0., 1., 0.,
        1., 0., 1., 0., 1., 0., 1., 0.],
       [1., 0., 1., 0., 0., 1., 1., 0., 0., 1., 1., 0., 1., 0., 1., 0.,
        1., 0., 1., 0., 1., 0., 1., 0.],
       [1., 0., 1., 0., 0., 1., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0.,
        1., 0., 0., 1., 1., 0., 1., 0.],
       [1., 0., 1., 0., 0., 1., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0.,
        1., 0., 1., 0., 0., 1., 1., 0.],
       [1., 0., 0., 1., 1., 0., 0., 1., 1., 0., 1., 0., 1., 0., 1., 0.,
        1., 0., 1., 0., 1., 0., 1., 0.],
       [1., 0., 0., 1., 1., 0., 1., 0., 0., 1., 1., 0., 1., 0., 1., 0.,
        1., 0., 1., 0., 1., 0., 1., 0.],
       [1., 0., 0., 1., 1., 0., 1., 0., 1., 0., 1., 0., 0., 1., 1., 0.,
        1., 0., 1., 0., 1., 0., 1., 0.],
       [1., 0., 0., 1., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0.,
        1., 0., 1., 0., 0., 1., 1., 0.]])
```

In [38]:
```
df1=pd.read_csv("carprices.csv")
df1
```

Out[38]:

	CarModel	Mileage	Sell Price	Age(yrs)
0	BMW X5	69000	18000	6
1	BMW X5	35000	34000	3
2	BMW X5	57000	26100	5
3	BMW X5	22500	40000	2
4	BMW X5	46000	31500	4
5	Audi A5	59000	29400	5
6	Audi A5	52000	32000	5
7	Audi A5	72000	19300	6
8	Audi A5	91000	12000	8
9	Mercedez Benz C class	67000	22000	6
10	Mercedez Benz C class	83000	20000	7
11	Mercedez Benz C class	79000	21000	7
12	Mercedez Benz C class	59000	33000	5

In [39]:
```
dummies1=pd.get_dummies(df1.CarModel)
dummies1
```

Out[39]:

	Audi A5	BMW X5	Mercedez Benz C class
0	0	1	0
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0
5	1	0	0
6	1	0	0
7	1	0	0
8	1	0	0
9	0	0	1
10	0	0	1
11	0	0	1
12	0	0	1

In [46]:
```
merged1 =pd.concat([df1,dummies1],axis=1)
merged1
```

Out[46]:

	CarModel	Mileage	Sell Price	Age(yrs)	Audi A5	BMW X5	Mercedez Benz C class
0	BMW X5	69000	18000	6	0	1	0
1	BMW X5	35000	34000	3	0	1	0
2	BMW X5	57000	26100	5	0	1	0
3	BMW X5	22500	40000	2	0	1	0
4	BMW X5	46000	31500	4	0	1	0
5	Audi A5	59000	29400	5	1	0	0
6	Audi A5	52000	32000	5	1	0	0
7	Audi A5	72000	19300	6	1	0	0
8	Audi A5	91000	12000	8	1	0	0
9	Mercedez Benz C class	67000	22000	6	0	0	1
10	Mercedez Benz C class	83000	20000	7	0	0	1
11	Mercedez Benz C class	79000	21000	7	0	0	1
12	Mercedez Benz C class	59000	33000	5	0	0	1

In [47]:
```
final1 = merged1.drop(['CarModel','BMW X5'],axis=1)
final1
```

Out[47]:

	Mileage	Sell Price	Age(yrs)	Audi A5	Mercedez Benz C class
0	69000	18000	6	0	0
1	35000	34000	3	0	0
2	57000	26100	5	0	0
3	22500	40000	2	0	0
4	46000	31500	4	0	0
5	59000	29400	5	1	0
6	52000	32000	5	1	0
7	72000	19300	6	1	0
8	91000	12000	8	1	0
9	67000	22000	6	0	1
10	83000	20000	7	0	1
11	79000	21000	7	0	1
12	59000	33000	5	0	1

In [48]:
```
from sklearn import linear_model
```

In [64]:
```
model1=linear_model.LinearRegression()
X=final1.drop(['Sell Price'],axis=1)
X
```

Out[64]:

	Mileage	Age(yrs)	Audi A5	Mercedez Benz C class
0	69000	6	0	0
1	35000	3	0	0
2	57000	5	0	0
3	22500	2	0	0
4	46000	4	0	0
5	59000	5	1	0
6	52000	5	1	0
7	72000	6	1	0
8	91000	8	1	0
9	67000	6	0	1
10	83000	7	0	1
11	79000	7	0	1
12	59000	5	0	1

In [62]:
```
y=final1['Sell Price']
y
```

Out[62]:
```
0     18000
1     34000
2     26100
3     40000
4     31500
5     29400
6     32000
7     19300
8     12000
9     22000
10    20000
11    21000
12    33000
Name: Sell Price, dtype: int64
```

In [77]:
```
model1.fit(X,y)
```

Out[77]:
```
▾ LinearRegression
LinearRegression()
```

In [79]:
```
model1.score(X,y)
```

Out[79]:
```
0.9417050937281082
```

In [74]:
```
model1.predict([[45000,4,0,1]])
```

```
C:\Program Files\Python311\Lib\site-packages\sklearn\base.py:439: UserWa
rning: X does not have valid feature names, but LinearRegression was fit
ted with feature names
  warnings.warn(
```

Out[74]:
```
array([36991.31721031])
```

In [75]:
```
model1.predict([[86000,7,0,0]])
```

```
C:\Program Files\Python311\Lib\site-packages\sklearn\base.py:439: UserWa
rning: X does not have valid feature names, but LinearRegression was fit
ted with feature names
  warnings.warn(
```

Out[75]:
```
array([11080.74313267])
```

In [ ]:
```

```