In [45]:
```
import pandas as pd
from sklearn import linear_model
import numpy as np
```

In [2]:
```
df=pd.read_csv("homeprices.csv")
```

In [3]:
```
df
```

Out[3]:

	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 [4]:
```
pd.get_dummies(df.town)
#here i learn to create the dummy variable
#based on town value
```

Out[4]:

	monroe township	robinsville	west windsor
0	1	0	0
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
5	0	0	1
6	0	0	1
7	0	0	1
8	0	0	1
9	0	1	0
10	0	1	0
11	0	1	0
12	0	1	0

In [5]:
```
a=pd.get_dummies(df.town)
```

In [6]:
```
df=pd.concat([df,a],axis=1)
```

In [7]:
```
df
```

Out[7]:

	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 [19]:
```
df.drop(['town','west windsor'],axis=1,inplace=True)
'''here i drop the town column and one of the dummy column becasue according to rules we have to drop one dummy columns
'''
```

Out[19]:
```
'here i drop the town column and one of the dummy column becasue according to rules we have to drop one dummy columns\n'
```

In [20]:
```
model=linear_model.LinearRegression()
```

In [21]:
```
x=df.drop('price',axis=1)  # we drop the price column
```

In [22]:
```
y=df.price
```

In [23]:
```
df
```

Out[23]:

	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 [24]:
```
model.fit(x,y)  # we train model, we pass x and y
```

Out[24]:
```
LinearRegression()
```

In [26]:
```
model.predict([[2800,0,1]])  # here thats how we give input based on columns
# price of house 2800 area and town is robinsville
```

```
C:\Users\Asus\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[26]:
```
array([590775.63964739])
```

In [27]:
```
model.predict([[3400,0,0]])
#here the area is 3400 of wind windsor town
```

```
C:\Users\Asus\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[27]:
```
array([681241.66845839])
```

In [28]:
```
model.score(x,y) # this is done to see the accuracy of model
# maximum value is 1
```

Out[28]:
```
0.9573929037221872
```

In [29]:
```
# now we will learn how to use the one hot encoding from sklearn
```

In [30]:
```
# another way
```

In [32]:
```
from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()
```

In [33]:
```
le=LabelEncoder()
```

In [34]:
```

```

Out[34]:

	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 [35]:
```
dfg=pd.read_csv("homeprices.csv")
```

In [36]:
```
dfg
```

Out[36]:

	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 [38]:
```
dfg.town=le.fit_transform(dfg.town)  # we transform the town column
```

In [39]:
```
dfg
```

Out[39]:

	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 [50]:
```
x=dfg[['town','area']].values
```

In [51]:
```
x
```

Out[51]:
```
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 [52]:
```
y=dfg.price
```

In [53]:
```
y
```

Out[53]:
```
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 [54]:
```
md=linear_model.LinearRegression()
```

In [ ]:
```

```

In [ ]:
```

```

In [ ]: