In [48]:
```
import pandas as pd
import numpy as np
from sklearn import linear_model
from word2number import w2n
```

In [47]:
```
df = pd.read_csv("homeprices.csv")
df
```

Out[47]:

	area	bedrooms	age	price
0	2600	3.0	20	550000
1	3000	4.0	15	565000
2	3200	NaN	18	610000
3	3600	3.0	30	595000
4	4000	5.0	8	760000
5	4100	6.0	8	810000

In [10]:
```
import math
median1=df.bedrooms.median()
```

In [11]:
```
df.fillna(median1,inplace=True)
```

In [12]:
```
df
```

Out[12]:

	area	bedrooms	age	price
0	2600	3.0	20	550000
1	3000	4.0	15	565000
2	3200	4.0	18	610000
3	3600	3.0	30	595000
4	4000	5.0	8	760000
5	4100	6.0	8	810000

In [13]:
```
reg = linear_model.LinearRegression()
reg.fit(df[['area','bedrooms','age']],df.price)
```

Out[13]:

▾ LinearRegression

LinearRegression()

In [14]:
```
reg.coef_
```

Out[14]:
```
array([  112.06244194, 23388.88007794, -3231.71790863])
```

In [19]:
```
reg.predict(np.array([[3000,4,2]]))
```

```
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[19]:
```
array([644602.4121863])
```

In [35]:
```
df1=pd.read_csv("hiring.csv")
df1
```

Out[35]:

	experience	test_score	interview_score	salary
0	NaN	8.0	9	50000
1	NaN	8.0	6	45000
2	five	6.0	7	60000
3	two	10.0	10	65000
4	seven	9.0	6	70000
5	three	7.0	10	62000
6	ten	NaN	7	72000
7	eleven	7.0	8	80000

In [36]:
```
import math
median1=df1.median().test_score
median1
```

```
C:\Users\KIIT\AppData\Local\Temp\ipykernel_10776\1676239188.py:2: Future
Warning: The default value of numeric_only in DataFrame.median is deprec
ated. In a future version, it will default to False. In addition, specif
ying 'numeric_only=None' is deprecated. Select only valid columns or spe
cify the value of numeric_only to silence this warning.
  median1=df1.median().test_score
```

Out[36]:
```
8.0
```

In [37]:
```
df1.fillna({
    "test_score": median1,
    "experience":0
},inplace=True)
```

In [53]:
```
df1 = df1.astype({'experience':'string'})
df1
df1.experience = df1.experience.apply(w2n.word_to_num)
df1
```

Out[53]:

	experience	test_score	interview_score	salary
0	0	8.0	9	50000
1	0	8.0	6	45000
2	5	6.0	7	60000
3	2	10.0	10	65000
4	7	9.0	6	70000
5	3	7.0	10	62000
6	10	8.0	7	72000
7	11	7.0	8	80000

In [55]:
```
reg1=linear_model.LinearRegression()
reg1.fit(df1[['experience','test_score','interview_score']],df1.salary)
```

Out[55]:

▾ LinearRegression

LinearRegression()

In [56]:
```
reg1.predict(np.array([[2,9,6]]))
```

```
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[56]:
```
array([53205.96797671])
```

In [57]:
```
reg1.predict(np.array([[12,10,10]]))
```

```
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[57]:
```
array([92002.18340611])
```

In [58]:
```
import pickle
```

In [60]:
```
with open('model_picckle','wb') as f:
    pickle.dump(reg,f)
```

In [64]:
```
with open('model_picckle','rb') as f:
    mp=pickle.load(f)
```

In [65]:
```
mp.predict(np.array([[3000,4,2]]))
```

```
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[65]:
```
array([644602.4121863])
```

In [ ]:

In [ ]: