In [10]:
```
import pandas as pd
import seaborn as sns
from sklearn import linear_model
from word2number import w2n
```

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

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

Out[3]:

	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 [7]:
```
a=df.bedrooms.median()
```

In [10]:
```
df['bedrooms'].fillna(a,inplace=True)
```

In [11]:
```
df
```

Out[11]:

	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 [12]:
```
reg=linear_model.LinearRegression()
```

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

Out[13]:
```
▾ LinearRegression
LinearRegression()
```

In [14]:
```
reg.predict([[3000,3,40]])
```

```
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[14]:
```
array([498408.25158031])
```

In [15]:
```
# this is the model to predict the price of house based on mutiple variable
# here we learn to fill empty value with median
```

In [16]:
```
# now we are gonna build the model to predict the salary based on experience, test score,and interview score
```

In [66]:
```
ram=pd.read_csv("hiring.csv")
```

In [67]:
```
ram
```

Out[67]:

	experience	test_score(out of 10)	interview_score(out of 10)	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 [68]:
```
ram.experience.fillna("zero",inplace=True)
```

In [ ]:
```

```

In [69]:
```
ram
```

Out[69]:

	experience	test_score(out of 10)	interview_score(out of 10)	salary($)
0	zero	8.0	9	50000
1	zero	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 [70]:
```
def add(word):
    return w2n.word_to_num(word)

ram['experience'] = ram['experience'].apply(add)
# here i learn to convert the string number to int number column
```

In [73]:
```
ram = ram.rename(columns={'test_score(out of 10)': 'testscore', 'interview_score(out of 10)': 'interview score',
                "salary($)":"salary"})

#here we change the column name
```

In [74]:
```
ram
```

Out[74]:

	experience	testscore	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	NaN	7	72000
7	11	7.0	8	80000

In [77]:
```
b=ram.testscore.median()
```

In [79]:
```
ram.testscore.fillna(b,inplace=True)
```

In [80]:
```
ram
```

Out[80]:

	experience	testscore	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 [86]:
```
hari=linear_model.LinearRegression()
```

hari.fit(ram[['experience','testscore','interview score']],ram.salary)

In [87]:
```
hari.fit(ram[['experience','testscore','interview score']],ram.salary)
```

Out[87]:
```
▾ LinearRegression
LinearRegression()
```

In [89]:
```
hari.predict([[2,9,6]])
```

```
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[89]:
```
array([53205.96797671])
```

In [ ]: