In [87]:
```
import pandas as pd
import numpy as np
from sklearn import linear_model
from sklearn.model_selection import train_test_split
```

In [88]:
```
n1  = pd.read_csv("C:/Users/Najeeb Irfan/Desktop/mlpredict/new.csv" )
n1
```

Out[88]:

	Experience	Testscore	interview score	salary
0	0	8.0	9	50000
1	0	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 [89]:
```
n1.tail()
```

Out[89]:

	Experience	Testscore	interview score	salary
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 [90]:
```
dictionary = {"five": 5, "two": 2, "seven": 7, "three": 3, "ten": 10,"eleven":11}
```

In [92]:
```
n1 = n1.replace({"Experience":dictionary})
n1
```

Out[92]:

	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 [93]:
```
n1
```

Out[93]:

	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 [94]:
```
n1.Testscore.median()
```

Out[94]: 8.0

In [95]:
```
import math
median_Testscore = math.floor(n1.Testscore.median())
median_Testscore
```

Out[95]: 8

In [96]:
```
n1.Testscore = n1.Testscore.fillna(median_Testscore)
n1
```

Out[96]:

	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 [97]:
```
n1.shape
```

Out[97]: (8, 4)

In [98]:
```
n1.shape[0] != n1.shape[0]
```

Out[98]: False

In [99]:
```
F = (n1[['Experience','Testscore','interview score']])#This will be our feature
t = n1['salary']#target
```

In [100…
```
F_train, F_test, t_train, t_test = train_test_split(F, t, test_size=0.4, random_stat
```

In [101…
```

```

```
regressor = LinearRegression()
regressor.fit(F_train,t_train)
```

Out[101… LinearRegression()

In [102…
```
regressor.coef_
```

Out[102… array([3782.60869565, 2260.86956522, 2913.04347826])

In [103…
```
regressor.predict([[2,9,6]])
```

Out[103… array([51086.95652174])

In [104…
```
regressor.predict([[12,10,10]])
```

Out[104… array([102826.08695652])

In [105…
```
#Check the Accuracy of the Model
print('Train Accuracy:',regressor.score(F_train,t_train),'\nTest Acuracy:',regressor
```

```
Train Accuracy: 1.0
Test Acuracy: 0.7740278080176461
```

In [ ]: