In [76]:
```
import pandas as pd
import numpy as np
from sklearn import linear_model
from word2number import w2n
```

In [77]:
```
df=pd.read_excel('salary.xlsx')
```

In [78]:
```
df
```

Out[78]:

| | 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 [79]:
```
df.experience=df.experience.fillna("zero")
```

In [80]:
```
df
```

Out[80]:

| | 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 [89]:
```
df.experience = df.experience.apply(w2n.word_to_num)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-89-e811e5fd667d> in <module>
----> 1 df.experience = df.experience.apply(w2n.word_to_num)

c:\new folder\pyathon38\lib\site-packages\pandas\core\series.py in apply(self, func, convert_dtype, args, **kwargs)
   4354         dtype: float64
   4355         """
-> 4356         return SeriesApply(self, func, convert_dtype, args, kwargs).apply()
   4357
   4358     def _reduce(

c:\new folder\pyathon38\lib\site-packages\pandas\core\apply.py in apply(self)
   1034             return self.apply_str()
   1035
-> 1036         return self.apply_standard()
   1037
   1038     def agg(self):

c:\new folder\pyathon38\lib\site-packages\pandas\core\apply.py in apply_standard(self)
   1090                 # List[Union[Callable[..., Any], str]]]]"; expected
   1091                 # "Callable[[Any], Any]"
-> 1092                 mapped = lib.map_infer(
   1093                     values,
   1094                     f,  # type: ignore[arg-type]

c:\new folder\pyathon38\lib\site-packages\pandas\_libs\lib.pyx in pandas._libs.lib.map_infer()

c:\new folder\pyathon38\lib\site-packages\word2number\w2n.py in word_to_num(number_sentence)
    132 def word_to_num(number_sentence):
    133     if type(number_sentence) is not str:
--> 134         raise ValueError("Type of input is not string! Please enter a valid number word (eg. \'two million twenty three thousand and forty nine\')")
    135
    136     number_sentence = number_sentence.replace('-', ' ')

ValueError: Type of input is not string! Please enter a valid number word (eg. 'two million twenty three thousand and forty nine')
```

In [88]:
```
import math
median_test_score = math.floor(df['test_score(out of 10)'].mean())
median_test_score
```

Out[88]: 7

In [83]:
```
df['test_score(out of 10)'] = df['test_score(out of 10)'].fillna(median_test_score)
```

In [84]:
```
reg = linear_model.LinearRegression()
reg.fit(df[['experience','test_score(out of 10)','interview_score(out of 10)']],df['salary($)'])
```

Out[84]: LinearRegression()

In [90]:
```
reg.predict([[2,9,6]])
```

```
c:\new folder\pyathon38\lib\site-packages\sklearn\base.py:445: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[90]: array([53713.86677124])

In [93]:
```
reg.predict([[3,8,9]])
```

```
c:\new folder\pyathon38\lib\site-packages\sklearn\base.py:445: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[93]: array([60857.27438577])

In [ ]:
```

```