In [ ]:
```
import numpy as np
import pandas as pd
df=pd.read_csv('Salary_data.csv')
df
```

In [19]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  30 non-null     float64
 1   Salary           30 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 612.0 bytes
```

In [3]:
```
df.dropna(inplace=True)
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  30 non-null     float64
 1   Salary           30 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 612.0 bytes
```

In [5]:
```
df.describe()
```

Out[5]:

	YearsExperience	Salary
count	30.000000	30.000000
mean	5.313333	76003.000000
std	2.837888	27414.429785
min	1.100000	37731.000000
25%	3.200000	56720.750000
50%	4.700000	65237.000000
75%	7.700000	100544.750000
max	10.500000	122391.000000

In [6]:
```
features=df.iloc[:,[0]].values
label=df.iloc[:,[1]].values
```

In [7]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(features,label,test_size=0.2,random_st
```

In [20]:
```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train,y_train)
```

Out[20]:
```
LinearRegression
LinearRegression()
```

In [21]:
```
model.score(x_train,y_train)
```

Out[21]: 0.9603182547438908

In [23]:
```
model.score(x_test,y_test)
```

Out[23]: 0.9184170849214232

In [24]:
```
model.coef_
```

Out[24]: array([[9281.30847068]])

In [25]:
```
model.intercept_
```

Out[25]: array([27166.73682891])

In [26]:
```
import pickle
pickle.dump(model,open('SalaryPred.model','wb'))
```

In [27]:
```
model=pickle.load(open('SalaryPred.model','rb'))
```

In [28]:
```
yr_of_exp=float(input("Enter Years of Experience: "))
yr_of_exp_NP=np.array([[yr_of_exp]])
Salary=model.predict(yr_of_exp_NP)
```

```
Enter Years of Experience: 44
```

In [ ]:

In [29]:
```
print("Estimated Salary for {} years of experience is {}: " .format(yr_of_exp,Salary)
```

```
Estimated Salary for 44.0 years of experience is [[435544.30953887]]:
```

In [ ]: