In [5]:
```
import numpy as np
import pandas as pd
df=pd.read_csv('Salary_data.csv')
```

In [6]:
```
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: 608.0 bytes
```

In [7]:
```
df.dropna(inplace=True)
```

In [8]:
```
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: 608.0 bytes
```

In [9]:
```
df.describe()
```

Out[9]:

	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 [10]:
```
features=df.iloc[:,[0]].values
label=df.iloc[:,[1]].values
```

In [14]:

```
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,ra
```

In [15]:

```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train,y_train)
```

Out[15]:

```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [16]:

```
model.score(x_train,y_train)
```

Out[16]:

```
0.9645401573418146
```

In [17]:

```
model.score(x_test,y_test)
```

Out[17]:

```
0.9024461774180497
```

In [18]:

```
model.coef_
```

Out[18]:

```
array([[9423.81532303]])
```

In [19]:

```
model.intercept_
```

Out[19]:

```
array([25321.58301178])
```

In [20]:

```
import pickle
pickle.dump(model,open('SalaryPred.model','wb'))
```

In [21]:

```
model=pickle.load(open('SalaryPred.model','rb'))
```

In [ ]:

```
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)
```

In [ ]:

In [ ]:

```
print("Estimated Salary for {} years of experience is {}: " .format(yr_of_exp,
```

In [ ]: