Importing Libraries

In [21]:
```
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import statsmodels.formula.api as smf
```

Business Understanding

```
    prediction Model for Salary hike
```

Data Collection

In [4]:
```
salary_data=pd.read_csv('Salary_Data.csv')
salary_data
```

Out[4]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0
13	4.1	57081.0
14	4.5	61111.0
15	4.9	67938.0
16	5.1	66029.0
17	5.3	83088.0
18	5.9	81363.0
19	6.0	93940.0
20	6.8	91738.0
21	7.1	98273.0
22	7.9	101302.0
23	8.2	113812.0
24	8.7	109431.0
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

Data Understanding

In [7]:
```
salary_data.shape
```

Out[7]:
```
(30, 2)
```

In [8]:
```
salary_data.dtypes
```

Out[8]:
```
YearsExperience    float64
Salary             float64
dtype: object
```

In [9]:
```
salary_data.isna().sum()
```

Out[9]:
```
YearsExperience    0
Salary             0
dtype: int64
```

In [11]:
```
salary_data.describe(include='all')
```

Out[11]:

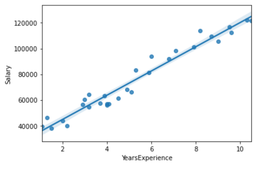
	YearsExperience	Salary
count	30.000000	30.000000
mean	5.313333	76003.000000
std	2.837888	27414.429785
min	1.100000	37731.000000
25%	3.200000	56720.750000
50%	4.700000	65237.000000
75%	7.700000	100544.750000
max	10.500000	122391.000000

Checking Assumptions for matching

In [40]:
```
plt.scatter(x = 'YearsExperience', y = 'Salary',data = salary_data)
plt.title('YearsExperience Vs Salary')
plt.xlabel('YearsExperience')
plt.ylabel('Salary')
plt.show()
```

In [18]:
```
sns.regplot(x = 'YearsExperience',y='Salary',data =salary_data)
```

Out[18]:
```
<AxesSubplot:xlabel='YearsExperience', ylabel='Salary'>
```



Model Training and Model Testing

In [30]:
```
linear_model = smf.ols(formula = 'Salary~YearsExperience', data = salary_data).fit()
#Model Training
```

Check for the deliverables of the training time

In [31]:
```
linear_model.params
```

Out[31]:
```
Intercept          25792.200199
YearsExperience     9449.962321
dtype: float64
```

In [32]:
```
 linear_model.tvalues , linear_model.pvalues
```

Out[32]:
```
(Intercept          11.346940
 YearsExperience    24.950094
 dtype: float64,
 Intercept          5.511950e-12
 YearsExperience    1.143068e-20
 dtype: float64)
```

In [33]:
```
linear_model.rsquared , linear_model.rsquared_adj
```

Out[33]:
```
(0.9569566641435086, 0.9554194021486339)
```

Model Predictions

In [34]:
```
# Manual prediction for say 3 Years Experience
Salary = (25792.200199) + (9449.962321)*(3)
Salary
```

Out[34]:
```
54142.087162
```

Automatic Prediction for say 3 & 5 Years Experience

In [35]:
```
new_data=pd.Series([3,5])
new_data
```

Out[35]:
```
0    3
1    5
dtype: int64
```

In [36]:
```
data_pred=pd.DataFrame(new_data,columns=['YearsExperience'])
data_pred
```

Out[36]:

	YearsExperience
0	3
1	5

In [38]:
```
linear_model.predict(data_pred)
```

Out[38]:
```
0    54142.087163
1    73042.011806
dtype: float64
```

In [ ]:
```

```