In [26]:
```
import numpy as np
import pandas as pd
```

In [27]:
```
data=pd.read_csv("Salary_Data.csv")
data
```

Out[27]:

	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

In [28]:
```
data.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     float64
dtypes: float64(2)
memory usage: 608.0 bytes
```

In [29]:
```
# there are no null values
```

correlation

In [30]:
```
data.corr()
```

Out[30]:

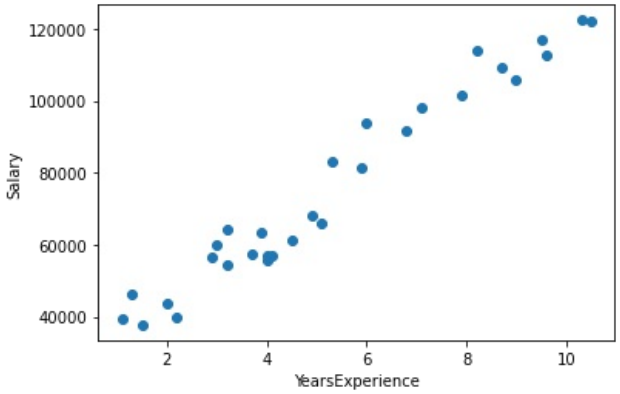
	YearsExperience	Salary
YearsExperience	1.000000	0.978242
Salary	0.978242	1.000000

In [31]:

```
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
x=data.YearsExperience
y=data.Salary
plt.scatter(x,y)
plt.xlabel("YearsExperience")
plt.ylabel("Salary")
```

Out[31]:

```
Text(0, 0.5, 'Salary')
```



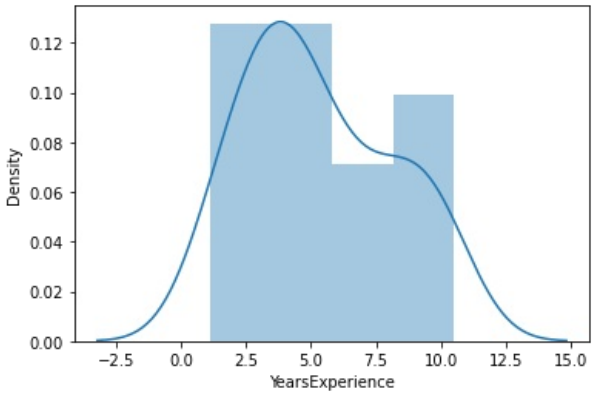
In [32]:

```
# to find distribution
import seaborn as sns
sns.distplot(data['YearsExperience'])
```

```
C:\Users\rajesh\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[32]:

```
<AxesSubplot:xlabel='YearsExperience', ylabel='Density'>
```



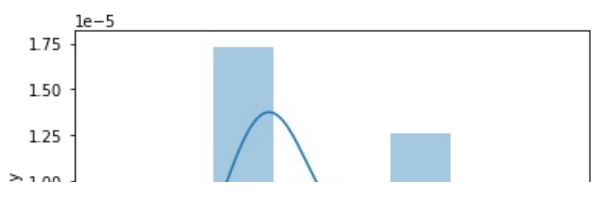
In [33]:

```
sns.distplot(data['Salary'])
```

```
C:\Users\rajesh\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[33]:

```
<AxesSubplot:xlabel='Salary', ylabel='Density'>
```



Build Model

In [34]:
```
import statsmodels.formula.api as smf
model=smf.ols("Salary~YearsExperience",data=data).fit()
model.summary()
```

Out[34]:

OLS Regression Results

Dep. Variable:	Salary	**R-squared:**	0.957
Model:	OLS	**Adj. R-squared:**	0.955
Method:	Least Squares	**F-statistic:**	622.5
Date:	Wed, 16 Feb 2022	**Prob (F-statistic):**	1.14e-20
Time:	11:37:21	**Log-Likelihood:**	-301.44
No. Observations:	30	**AIC:**	606.9
Df Residuals:	28	**BIC:**	609.7
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2.579e+04	2273.053	11.347	0.000	2.11e+04	3.04e+04
YearsExperience	9449.9623	378.755	24.950	0.000	8674.119	1.02e+04

Omnibus:	2.140	**Durbin-Watson:**	1.648
Prob(Omnibus):	0.343	**Jarque-Bera (JB):**	1.569
Skew:	0.363	**Prob(JB):**	0.456
Kurtosis:	2.147	**Cond. No.**	13.2

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [35]:
```
# as above data the probability is less  than 0.05 and rvalue .0.75
#predict data
```

In [36]:
```
pred=model.predict(data.YearsExperience)
pred
```

Out[36]:
```
0      36187.158752
1      38077.151217
2      39967.143681
3      44692.124842
4      46582.117306
5      53197.090931
6      54142.087163
7      56032.079627
8      56032.079627
9      60757.060788
10     62647.053252
11     63592.049484
12     63592.049484
13     64537.045717
14     68317.030645
15     72097.015574
16     73987.008038
17     75877.000502
```

```
18     81546.977895
19     82491.974127
20     90051.943985
21     92886.932681
22    100446.902538
23    103281.891235
24    108006.872395
25    110841.861092
26    115566.842252
27    116511.838485
28    123126.812110
29    125016.804574
dtype: float64
```

In [37]:

```
errors=data.Salary-pred
errors.mean()

model.resid
```

Out[37]:

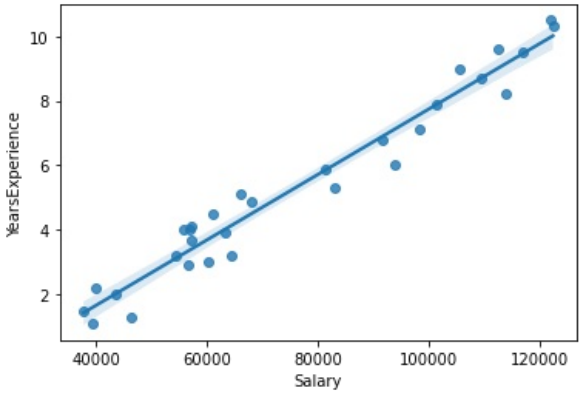
```
0      3155.841248
1      8127.848783
2     -2236.143681
3     -1167.124842
4     -6691.117306
5      3444.909069
6      6007.912837
7     -1587.079627
8      8412.920373
9     -3568.060788
10     570.946748
11    -7798.049484
12    -6635.049484
13    -7456.045717
14    -7206.030645
15    -4159.015574
16    -7958.008038
17     7210.999498
18     -183.977895
19    11448.025873
20     1686.056015
21     5386.067319
22      855.097462
23    10530.108765
24     1424.127605
25    -5259.861092
26     1402.157748
27    -3876.838485
28     -735.812110
29    -3144.804574
dtype: float64
```

In [38]:

```
sns.regplot(x="Salary",y="YearsExperience",data=data)
```

Out[38]:

```
<AxesSubplot:xlabel='Salary', ylabel='YearsExperience'>
```



In [39]:

```
#coefficents
model.params
```

Out[39]:

```
Intercept          25792.200199
YearsExperience     9449.962321
dtype: float64
```

In [40]:
```
#r values
(model.rsquared,model.rsquared_adj)
```

Out[40]:
```
(0.9569566641435086, 0.9554194021486339)
```

predicton

In [43]:
```
data1=pd.Series([300,200])
data_pred=pd.DataFrame(data1,columns=['YearsExperience'])
```

In [44]:
```
model.predict(data_pred)
```

Out[44]:
```
0    2.860781e+06
1    1.915785e+06
dtype: float64
```

In [ ]: