In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
import statsmodels.formula.api as smf
```

In [2]:
```
salary=pd.read_csv('Salary_Data.csv')
salary
```

Out[2]:

	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 [3]:
```
salary.describe()
```

Out[3]:

	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

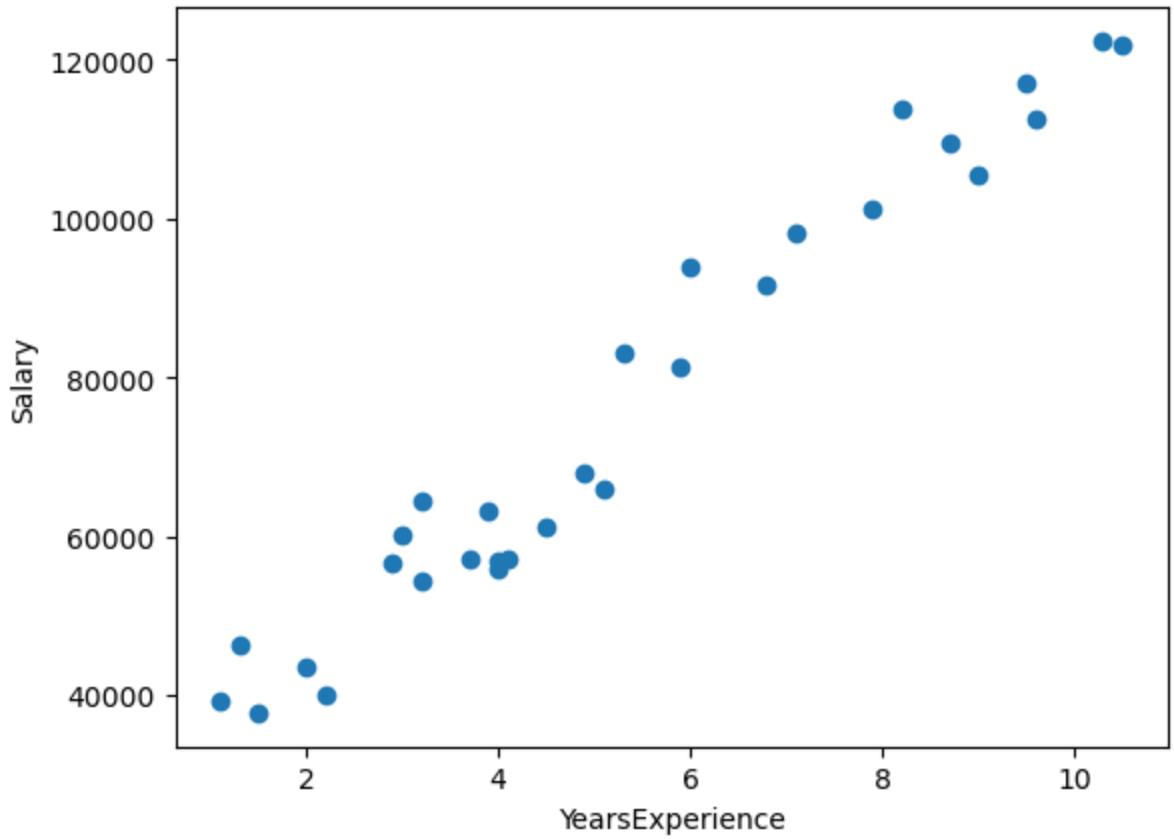
In [4]:
```
salary.corr()
```

Out[4]:

	YearsExperience	Salary
YearsExperience	1.000000	0.978242
Salary	0.978242	1.000000

In [5]:
```
x=salary.YearsExperience
y=salary.Salary
plt.scatter(x,y)
plt.xlabel('YearsExperience')
plt.ylabel('Salary')
```

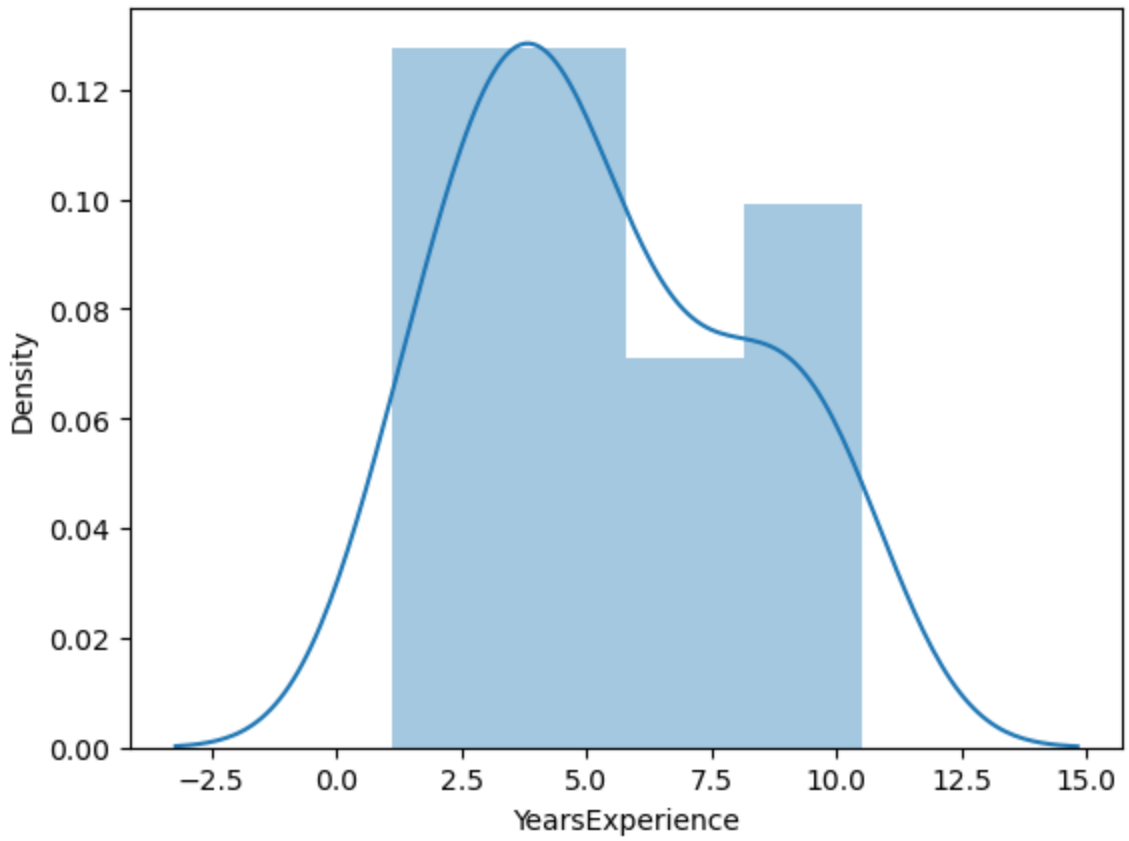
Out[5]:
```
Text(0, 0.5, 'Salary')
```



In [6]:
```
sns.distplot(salary['YearsExperience'])
```

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

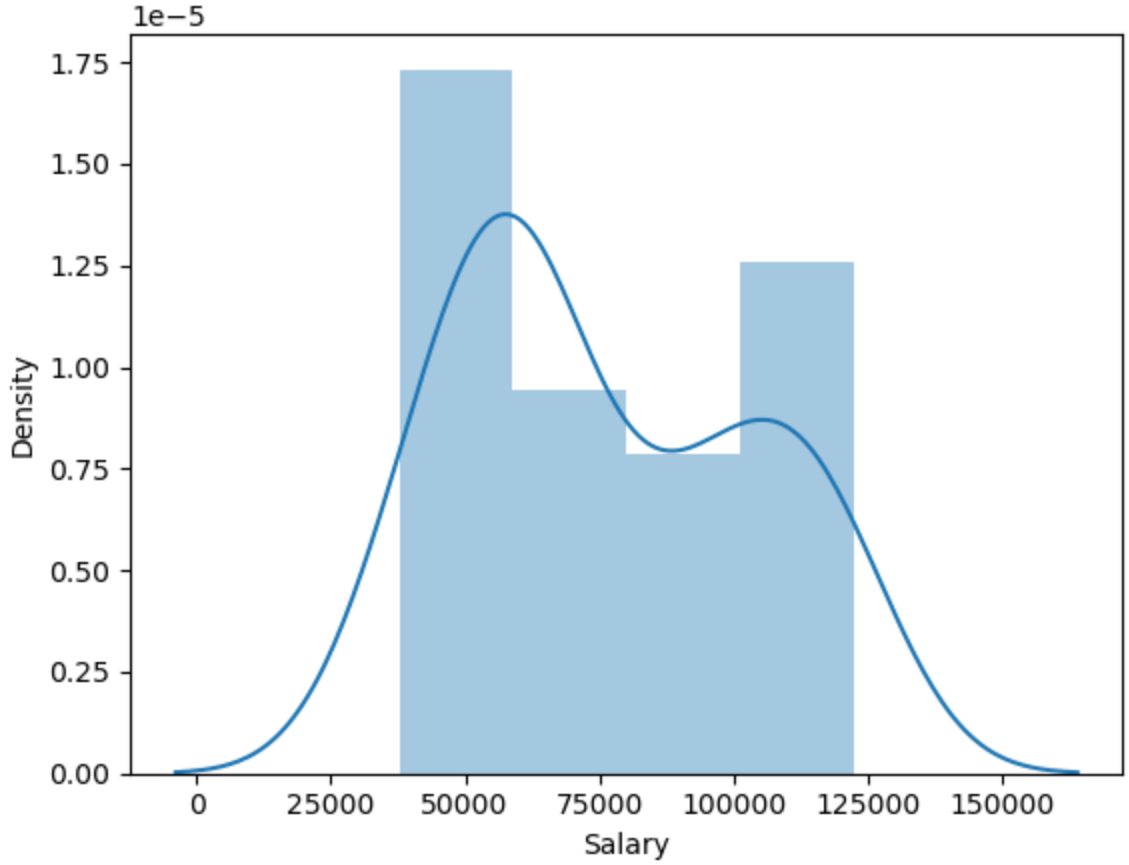
Out[6]: <AxesSubplot:xlabel='YearsExperience', ylabel='Density'>



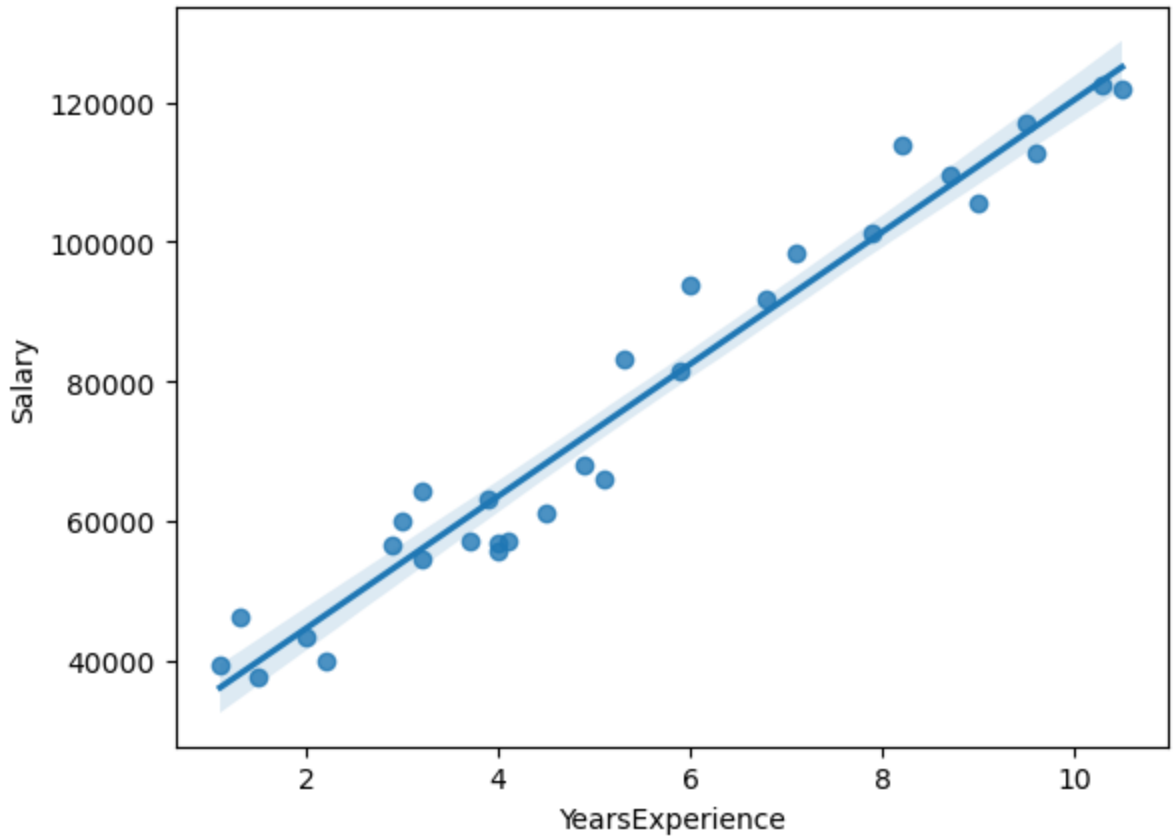
In [7]:
```
sns.distplot(salary['Salary'])
```

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

Out[7]: <AxesSubplot:xlabel='Salary', ylabel='Density'>



In [8]:
```
sns.regplot(x="YearsExperience", y="Salary", data=salary);
```



In [9]:
```
import statsmodels.formula.api as smf
model = smf.ols("Salary~YearsExperience",data = salary).fit()
model.summary()
```

Out[9]:

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:	Sun, 28 Jan 2024	**Prob (F-statistic):**	1.14e-20
Time:	20:56:06	**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 [10]:
```
pred=model.params
```

In [11]:
```
print(model.tvalues, '\n', model.pvalues)
```

```
Intercept          11.346940
YearsExperience    24.950094
dtype: float64
 Intercept          5.511950e-12
YearsExperience    1.143068e-20
dtype: float64
```

In [12]:
```
(model.rsquared,model.rsquared_adj)
```

Out[12]:
```
(0.9569566641435086, 0.9554194021486339)
```

In [13]:
```
newsalary=pd.Series([30,40])
```

In [14]:
```
data_pred=pd.DataFrame(newsalary,columns=['YearsExperience'])
data_pred
```

Out[14]:

	YearsExperience
0	30
1	40

In [15]:
```
data_pred=pd.DataFrame(newsalary,columns=['YearsExperience'])
```

Out[15]:

	YearsExperience
0	30
1	40

In [ ]: