In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

In [2]:
```
df = pd.read_csv("salary_data.csv")
```

In [3]:
```
df.head()
```

Out[3]:

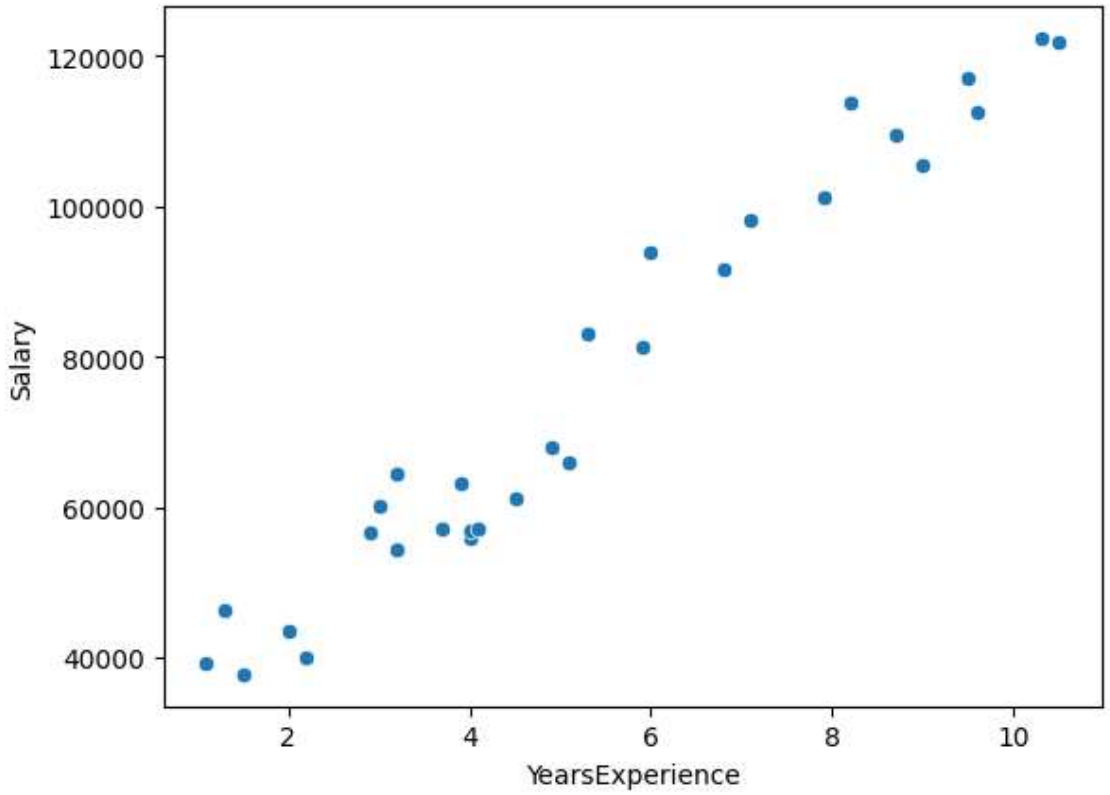
	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

In [4]:
```
X = df.iloc[:, 0]
Y = df.iloc[:, 1]
```

In [5]:
```
X = np.array(X).reshape(-1, 1)
Y = np.array(Y).reshape(-1, 1)
```

In [6]:
```
sns.scatterplot(x = df.iloc[:, 0], y = df.iloc[:, 1], data = df)
```

Out[6]: <Axes: xlabel='YearsExperience', ylabel='Salary'>



In [7]:
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X, Y)
```

Out[7]:
```
LinearRegression
LinearRegression()
```

In [8]:
```
print(lr.intercept_, lr.coef_)
```

```
[25792.20019867] [[9449.96232146]]
```

In [9]:
```
lr.predict([[5]])
```

Out[9]:
```
array([[73042.01180594]])
```

In [10]:
```
import joblib
```

In [11]:
```
joblib.dump(lr, "model.pkl")
# joblib.dump(var_object, "file_name"):- lr variable will stored on disk
# with file name as "model.pkl"
# .pkl :- extension for pickle library
```

Out[11]:
```
['model.pkl']
```

In [12]:
```
m = joblib.load("model.pkl")
# joblib.load("file_name")  # load model object in variable 'm'
```

In [13]:
```
m.predict([[5]])
```

Out[13]:
```
array([[73042.01180594]])
```

In [ ]:

In [14]:
```
# ========================================================================
```

In [ ]:

In [15]:
```
n = df.shape[0]
n
```

Out[15]:
```
30
```

In [16]:
```
summation_X = 0
summation_Y = 0
summation_XY = 0
summation_X2 = 0
for i in range(n):
    summation_X += df.iloc[i, 0]
    summation_Y += df.iloc[i, 1]
    summation_XY += (df.iloc[i, 0] * df.iloc[i, 1])
    summation_X2 += df.iloc[i, 0]**2
print(summation_X)
print(summation_Y)
print(summation_XY)
print(summation_X2)
```

```
159.4
2280090
14321961.0
1080.5
```

In [17]:
```
summation_Y = df["YearsExperience"].sum()  # or df.iloc[:, 1].sum()
print(summation_X)
```

```
159.4
```

In [18]:
```
summation_Y = df["Salary"].sum()  # or df.iloc[:, 1].sum()
print(summation_Y)
```

```
2280090
```

In [19]:
```
summation_XY = (df["YearsExperience"] * df["Salary"]).sum()
print(summation_XY)
```

```
14321961.0
```

In [20]:
```
summation_X2 = (df["YearsExperience"]**2).sum()
print(summation_X2)
```

```
1080.5
```

In [21]:
```
b_num = (n * summation_XY) - (summation_X * summation_Y)
b_den = (n * summation_X2) - (summation_X**2)
b = b_num / b_den
b
```

Out[21]:
```
9449.962321455077
```

In [22]:
```
a = (summation_Y - (b * summation_X)) / n
a
```

Out[22]:
```
25792.200198668685
```

In [23]:
```
def predict(X):
    Y = a + b * X
    return Y
```

In [24]:
```
predict(5)
```

Out[24]:
```
73042.01180594407
```

In [ ]:
```

```