1. Importing important library

In [8]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

In [4]:

```
data1 = pd.read_csv('Salary_Data.csv',sep=',')
data1.head()
```

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

2. Understanding data

In [5]:

```
data1.shape
```

Out[5]:

```
(30, 2)
```

In [6]:

```
data1.dtypes
```

Out[6]:

```
YearsExperience    float64
Salary             float64
dtype: object
```

In [7]:

```
data1.isnull().sum()
```

Out[7]:

```
YearsExperience    0
Salary             0
dtype: int64
```

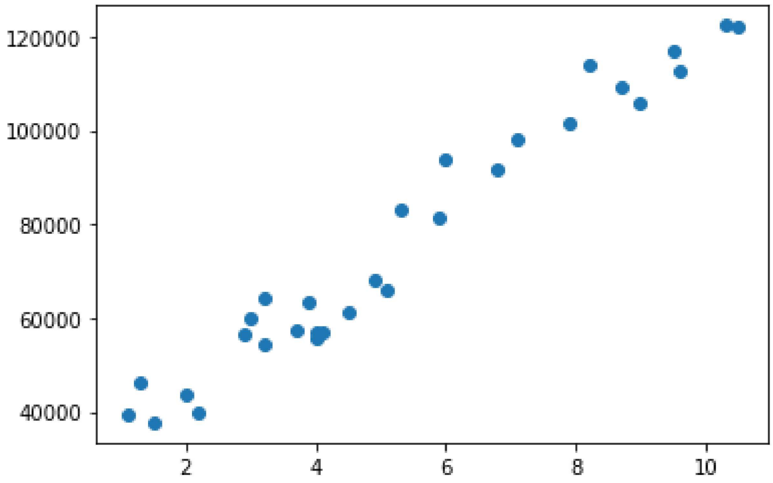
3.Checking assumptions

In [10]:

```
plt.scatter(x='YearsExperience',y='Salary',data=data1)
```

Out[10]:

```
<matplotlib.collections.PathCollection at 0x22389715af0>
```



In [11]:

```
## No autoregression and data following homoscadacity
```

In [12]:

```
data1.corr()
```

Out[12]:

	YearsExperience	Salary
YearsExperience	1.000000	0.978242
Salary	0.978242	1.000000

4. Model Building

In [15]:

```
from sklearn.model_selection import train_test_split
```

In [14]:

```
x = data1[['YearsExperience']]
y = data1[['Salary']]
```

In [19]:

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.20)
```

In [20]:

```
print(x_train.shape)
print(y_train.shape)
```

```
(24, 1)
(24, 1)
```

In [21]:

```
print(x_test.shape)
print(y_test.shape)
```

```
(6, 1)
(6, 1)
```

5.Model Training

In [22]:

```
from sklearn.linear_model import LinearRegression
```

In [24]:

```
linear_model = LinearRegression()
```

```
linear_model.fit(x_train,y_train)
```

Out[24]:

```
LinearRegression()
```

In [25]:

```
print('slope: ',linear_model.coef_)
print('intercept: ', linear_model.intercept_)
```

```
slope:  [[9657.2408511]]
intercept:  [24251.95383669]
```

6. Model Testing

In [26]:

```
y_predict = linear_model.predict(x_test)
```

In [27]:

```
y_predict
```

Out[27]:

```
array([[116961.46600726],
       [ 89921.19162418],
       [ 55155.12456021],
       [100544.15656039],
       [ 34874.9187729 ],
       [108269.94924127]])
```

In [28]:

```
y_test
```

Out[28]:

	Salary
27	112635.0
20	91738.0
8	64445.0
22	101302.0
0	39343.0
24	109431.0

7.Model Evaluation

In [29]:

```
from sklearn.metrics import mean_squared_error,r2_score
```

In [30]:

```
mean_squared_error(y_test,y_predict)
```

Out[30]:

```
21701166.978075575
```

In [31]:

```
r2_score(y_test,y_predict)
```

Out[31]:

```
0.9687423823822484
```