In [14]:

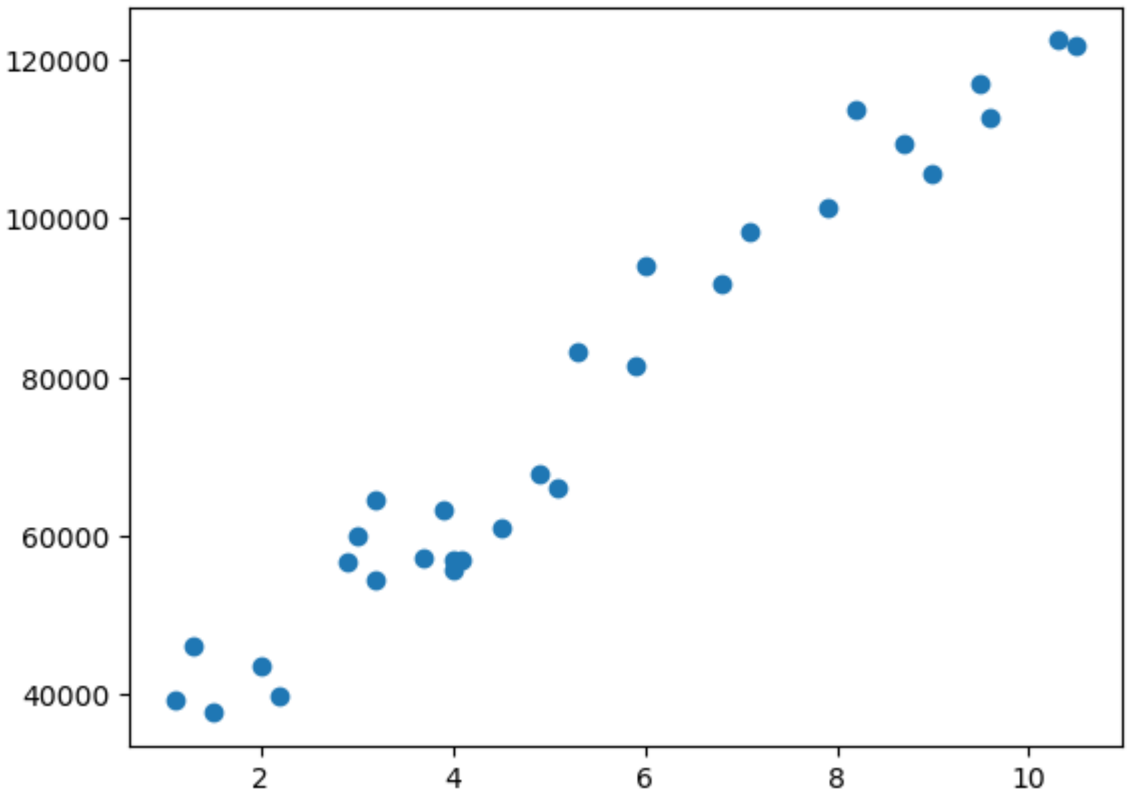
```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv('C:/Users/Asus/Downloads/Salary_Data.csv')
```

In [15]:

```
plt.scatter(df['YearsExperience'],df['Salary'])
```

Out[15]:

```
<matplotlib.collections.PathCollection at 0x259b08a2690>
```



In [16]:

```
from sklearn.cluster import KMeans
```

In [17]:

```
km=KMeans(n_clusters=4)
km
```

Out[17]:

```
▾       KMeans

KMeans(n_clusters=4)
```

In [18]:

```
y_pred=km.fit_predict(df[['YearsExperience','Salary']])
y_pred
```

```
C:\Users\Asus\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWa
rning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the
value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\Asus\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarn
ing: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable
OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[18]:

```
array([0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3,
       3, 1, 1, 1, 1, 1, 1, 1])
```

In [19]:

```
df['Cluster']=y_pred
```

In [20]:

```
df
```

Out[20]:

	YearsExperience	Salary	Cluster
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	2
6	3.0	60150	2
7	3.2	54445	2
8	3.2	64445	2
9	3.7	57189	2
10	3.9	63218	2
11	4.0	55794	2
12	4.0	56957	2
13	4.1	57081	2
14	4.5	61111	2
15	4.9	67938	2
16	5.1	66029	2
17	5.3	83088	3
18	5.9	81363	3
19	6.0	93940	3
20	6.8	91738	3
21	7.1	98273	3
22	7.9	101302	3
23	8.2	113812	1
24	8.7	109431	1
25	9.0	105582	1
26	9.5	116969	1
27	9.6	112635	1
28	10.3	122391	1
29	10.5	121872	1

In [21]:

```
km.cluster_centers_
```

Out[21]:

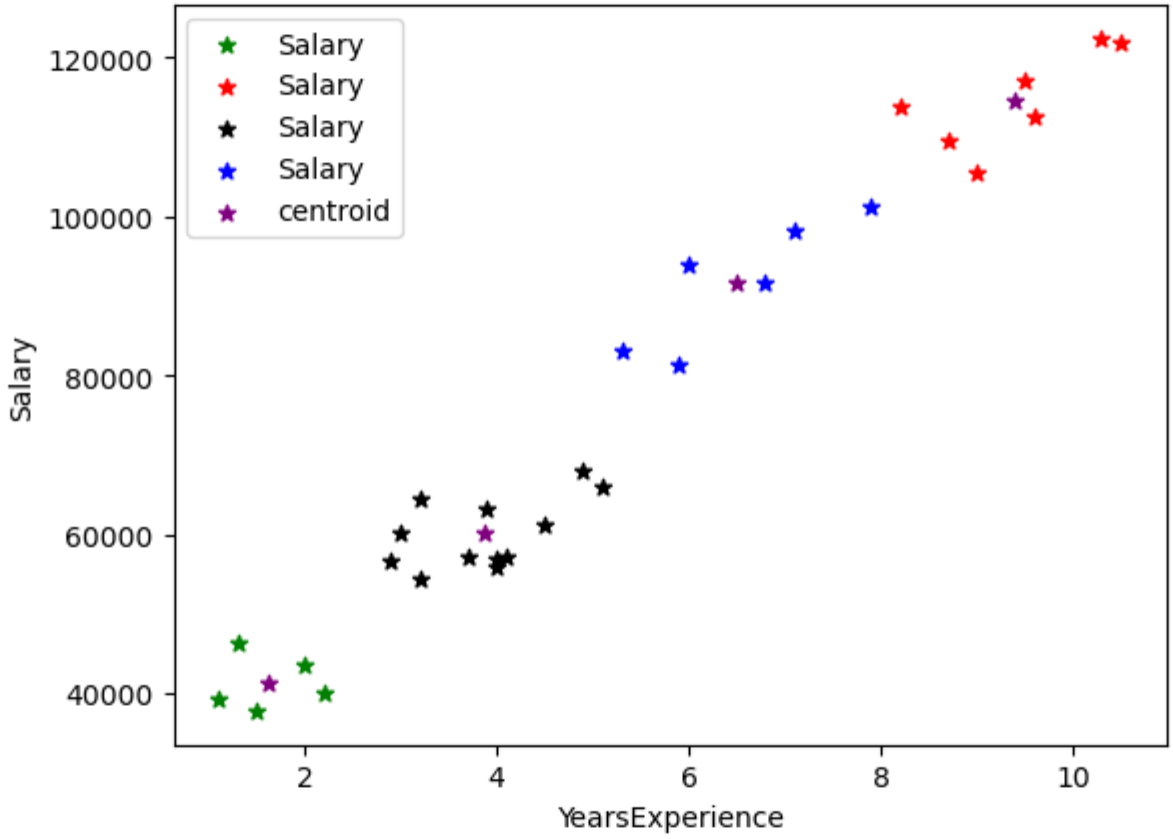
```
array([[1.62000000e+00, 4.13390000e+04],
       [9.40000000e+00, 1.14670286e+05],
       [3.87500000e+00, 6.00832500e+04],
       [6.50000000e+00, 9.16173333e+04]])
```

In [22]:

```
df1=df[df.Cluster==0]
df2=df[df.Cluster==1]
df3=df[df.Cluster==2]
df4=df[df.Cluster==3]
plt.scatter(df1.YearsExperience,df1['Salary'],color='green',marker='*',label='Salar
plt.scatter(df2.YearsExperience,df2['Salary'],color='red',marker='*',label='Salary'
plt.scatter(df3.YearsExperience,df3['Salary'],color='black',marker='*',label='Salar
plt.scatter(df4.YearsExperience,df4['Salary'],color='blue',marker='*',label='Salary
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker
plt.xlabel('YearsExperience')
plt.ylabel('Salary')
plt.legend()
```

Out[22]:

```
<matplotlib.legend.Legend at 0x259b7170590>
```



In [23]:

```
df1
```

Out[23]:

	YearsExperience	Salary	Cluster
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

In [24]:

```
df2
```

Out[24]:

	YearsExperience	Salary	Cluster
23	8.2	113812	1
24	8.7	109431	1
25	9.0	105582	1
26	9.5	116969	1
27	9.6	112635	1
28	10.3	122391	1
29	10.5	121872	1

In [25]:

```
df3
```

Out[25]:

	YearsExperience	Salary	Cluster
5	2.9	56642	2
6	3.0	60150	2
7	3.2	54445	2
8	3.2	64445	2
9	3.7	57189	2
10	3.9	63218	2
11	4.0	55794	2
12	4.0	56957	2
13	4.1	57081	2
14	4.5	61111	2
15	4.9	67938	2
16	5.1	66029	2

In [26]:

```
df4
```

Out[26]:

	YearsExperience	Salary	Cluster
17	5.3	83088	3
18	5.9	81363	3
19	6.0	93940	3
20	6.8	91738	3
21	7.1	98273	3
22	7.9	101302	3