Employee Salary Prediction using Linear Regression

In [2]:

```
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
%matplotlib inline
```

In [4]:

```
#reading the data
data = pd.read_csv('Salary.csv')
```

In [52]:

```
data.head()
```

Out[52]:

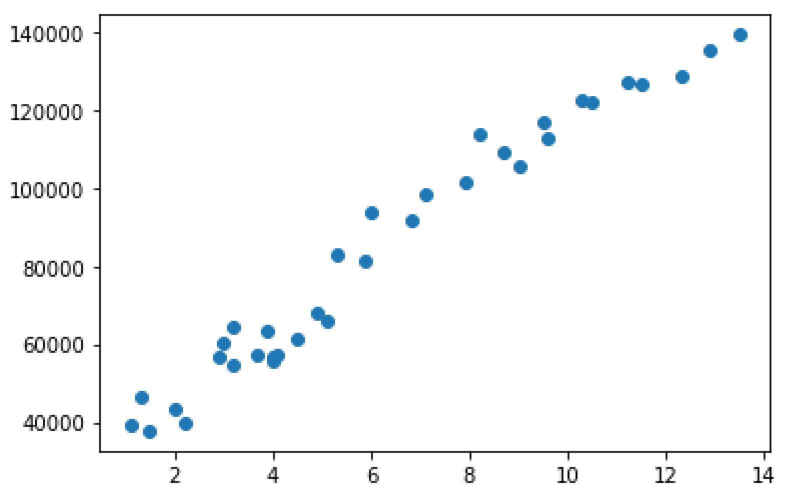
	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

In [13]:

```
#visualising the data inorder to find out about the trends in it
plt.scatter(data['YearsExperience'],data['Salary'])
```

Out[13]:

```
<matplotlib.collections.PathCollection at 0x1907a3230d0>
```



In [14]:

```
#storing the experience values and the salaries in particular variables
X =data.iloc[:,:-1].values
y=data.iloc[:,-1].values
```

In [15]:

```
X
```

Out[15]:

```
array([[ 1.1],
       [ 1.3],
       [ 1.5],
       [ 2. ],
       [ 2.2],
       [ 2.9],
       [ 3. ],
       [ 3.2],
       [ 3.2],
       [ 3.7],
       [ 3.9],
       [ 4. ],
       [ 4. ],
       [ 4.1],
       [ 4.5],
       [ 4.9],
       [ 5.1],
       [ 5.3],
       [ 5.9],
       [ 6. ],
       [ 6.8],
       [ 7.1],
       [ 7.9],
       [ 8.2],
       [ 8.7],
       [ 9. ],
       [ 9.5],
       [ 9.6],
       [10.3],
       [10.5],
       [11.2],
       [11.5],
       [12.3],
       [12.9],
       [13.5]])
```

In [16]:

```
y
```

Out[16]:

```
array([ 39343,  46205,  37731,  43525,  39891,  56642,  60150,  54445,
        64445,  57189,  63218,  55794,  56957,  57081,  61111,  67938,
        66029,  83088,  81363,  93940,  91738,  98273, 101302, 113812,
       109431, 105582, 116969, 112635, 122391, 121872, 127345, 126756,
       128765, 135675, 139465], dtype=int64)
```

In [17]:

```
#using sklearns train_test_split in order to divide the data into training and testing set and derive insights
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

In [18]:

```
from sklearn.linear_model import LinearRegression
#applying the linear regression method in the data in order to find out the optimal fit
linReg = LinearRegression()
linReg.fit(X_train,y_train)
```

Out[18]:

```
LinearRegression()
```

In [19]:

```
linReg
```

Out[19]:

```
LinearRegression()
```

In [20]:

```
linAns =linReg.predict(X_test)
```

In [21]:

```
linAns
```

Out[21]:

```
array([120197.8256403 ,  88644.21802942,  74146.61453254, 118492.2252289 ,
        98025.02029212,  72441.01412114,  63913.01206415,  43445.80712736,
        64765.81226984, 112522.623789  , 107405.82255481])
```

In [22]:

```
X_test
```

Out[22]:

```
array([[10.5],
       [ 6.8],
       [ 5.1],
       [10.3],
       [ 7.9],
       [ 4.9],
       [ 3.9],
       [ 1.5],
       [ 4. ],
       [ 9.6],
       [ 9. ]])
```

In [23]:

```
linAns
```

Out[23]:

```
array([120197.8256403 ,  88644.21802942,  74146.61453254, 118492.2252289 ,
        98025.02029212,  72441.01412114,  63913.01206415,  43445.80712736,
        64765.81226984, 112522.623789  , 107405.82255481])
```

In [24]:

```
y_test
```

Out[24]:

```
array([121872,  91738,  66029, 122391, 101302,  67938,  63218,  37731,
        55794, 112635, 105582], dtype=int64)
```

In [25]:

```
#Finding the difference between the predicted salary and the actual salary
#the values which have the smallest positive value will be closest to the actual output
difference = y_test - linAns
```

In [26]:

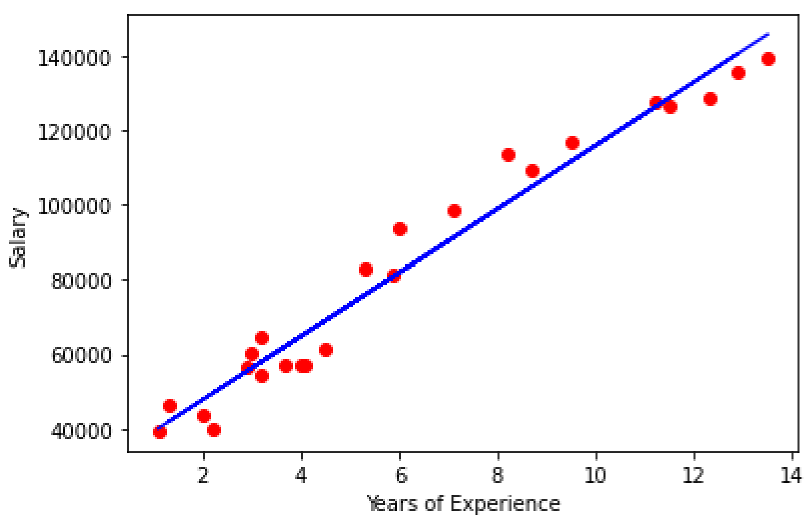
```
difference
```

Out[26]:

```
array([ 1674.1743597 ,  3093.78197058, -8117.61453254,  3898.7747711 ,
        3276.97970788, -4503.01412114,  -695.01206415, -5714.80712736,
       -8971.81226984,   112.376211  , -1823.82255481])
```

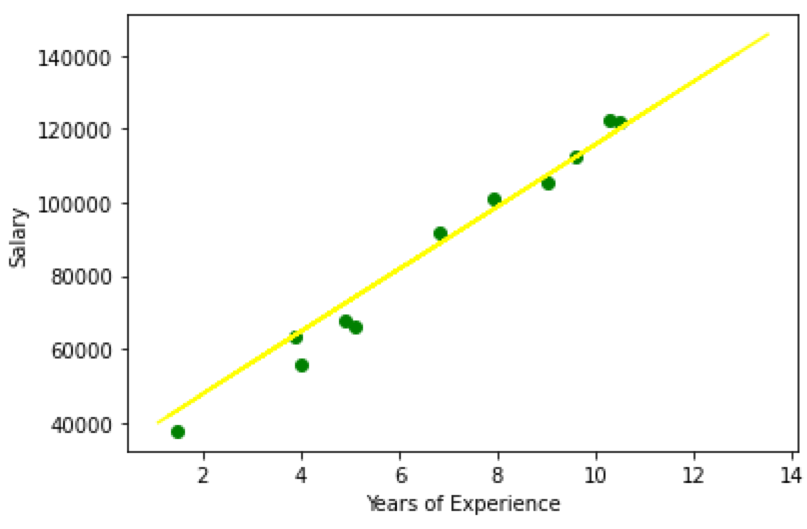
In [28]:

```
#Data visualization with training dataset
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, linReg.predict(X_train), color = 'blue')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```



In [29]:

```
#data visualization with the test data
plt.scatter(X_test,y_test,color='green')
plt.plot(X_train,linReg.predict(X_train),color='yellow')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```


Salary
Years of Experience