simple-linear-regression

March 6, 2025

SIMPLE LINEAR REGRESSION

Regression ->what is regression Regression is a statistical approach to find the correlation between variables(dependent and independent).

Regression algorithms give you continuous output.then you must choose one of the regression algorithm to bulid model.

types of regression 1.simple linear regresson 2.multi linear regression 3.polynomial regression 4.decision tree regressor 5.random forest regressor

simple linear regreession

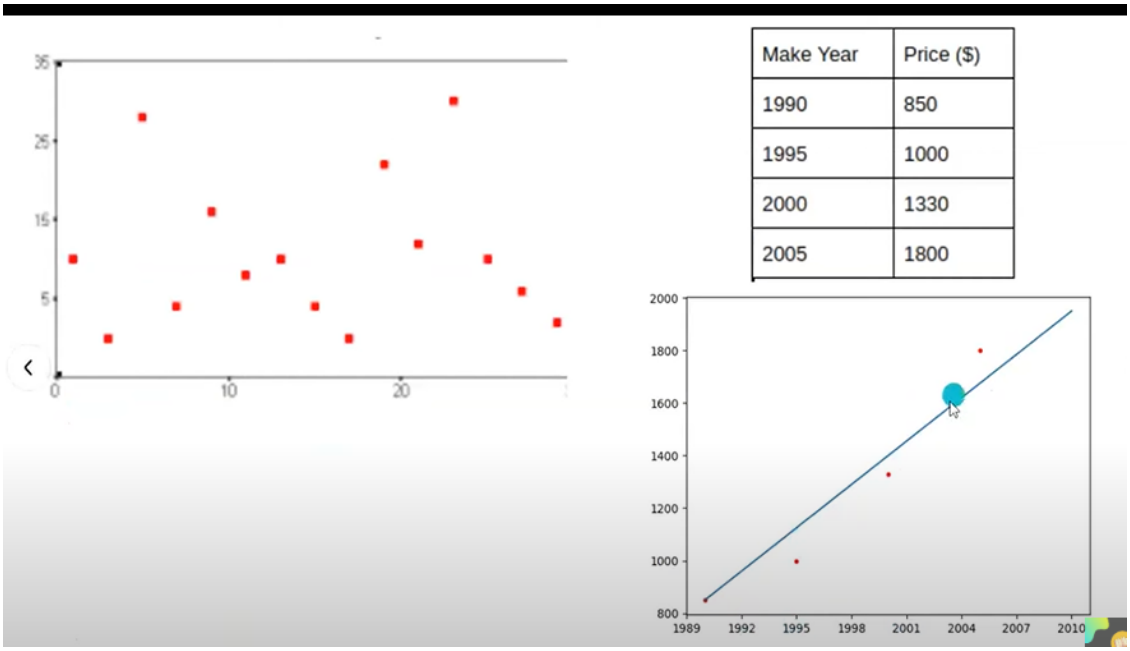
A simplr linear regression model only has two eplanatory variables 1.dependent variable anotherone 2. independent variable.think of two dimensional sapce.where the horizontal axis represents the independent variable-x and vertical axis represnts the dependent variable-y

where we can apply SLR. 1.Linearity

it assumes that the relation btween the data points is linear.

[12]:
```
from PIL import Image
Image.open("C:/Users/kamis/Pictures/Screenshots/Screenshot 2025-03-01 161135.
 ↪png")
```

[12]:



2.independence

There must be no relationship among the different values of the independent variable. the value is unique.

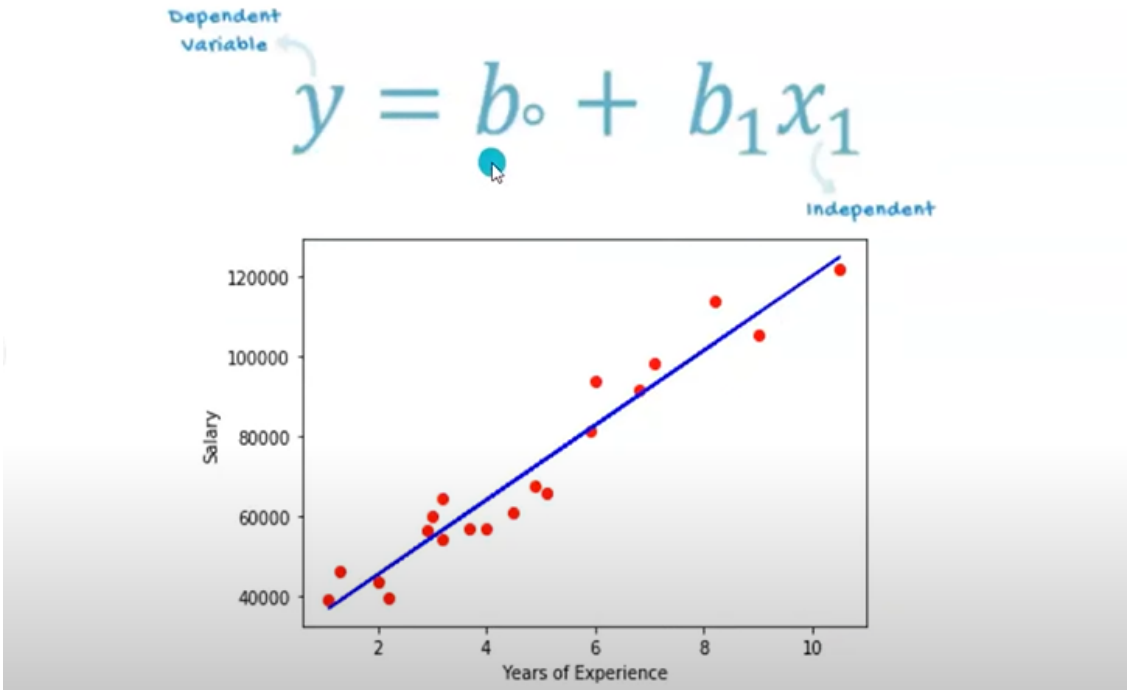
3.Normality

data points follow a normal distribution.if we plot the histogram fo the data,we should be able to draw a skewed line.

simple linear regresson equation

[24]:
```
Image.open("C:/Users/kamis/Pictures/Screenshots/Screenshot 2025-03-01 162724.
↪png")
```

[24]:



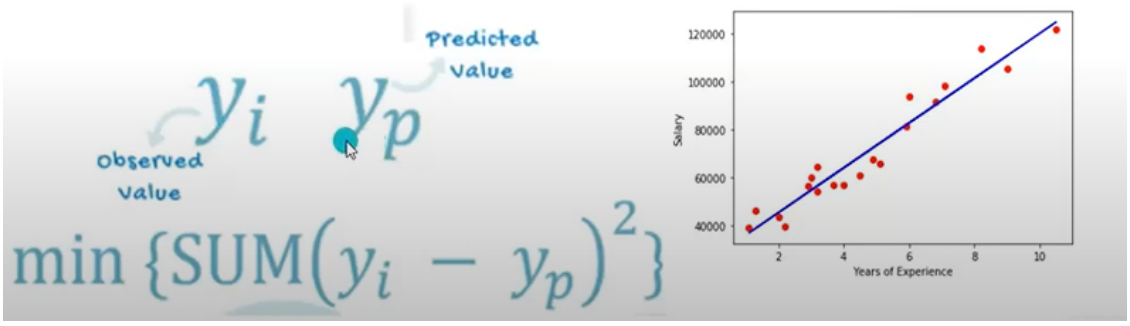
where do you find b0 and b1 values?

we are trying to minimize the errors in predicton by finding the best fit line.

we are trying to minimize the length btween the observed value(yi) and predicted valuve from our model (yp). #here observed value means actual output value.

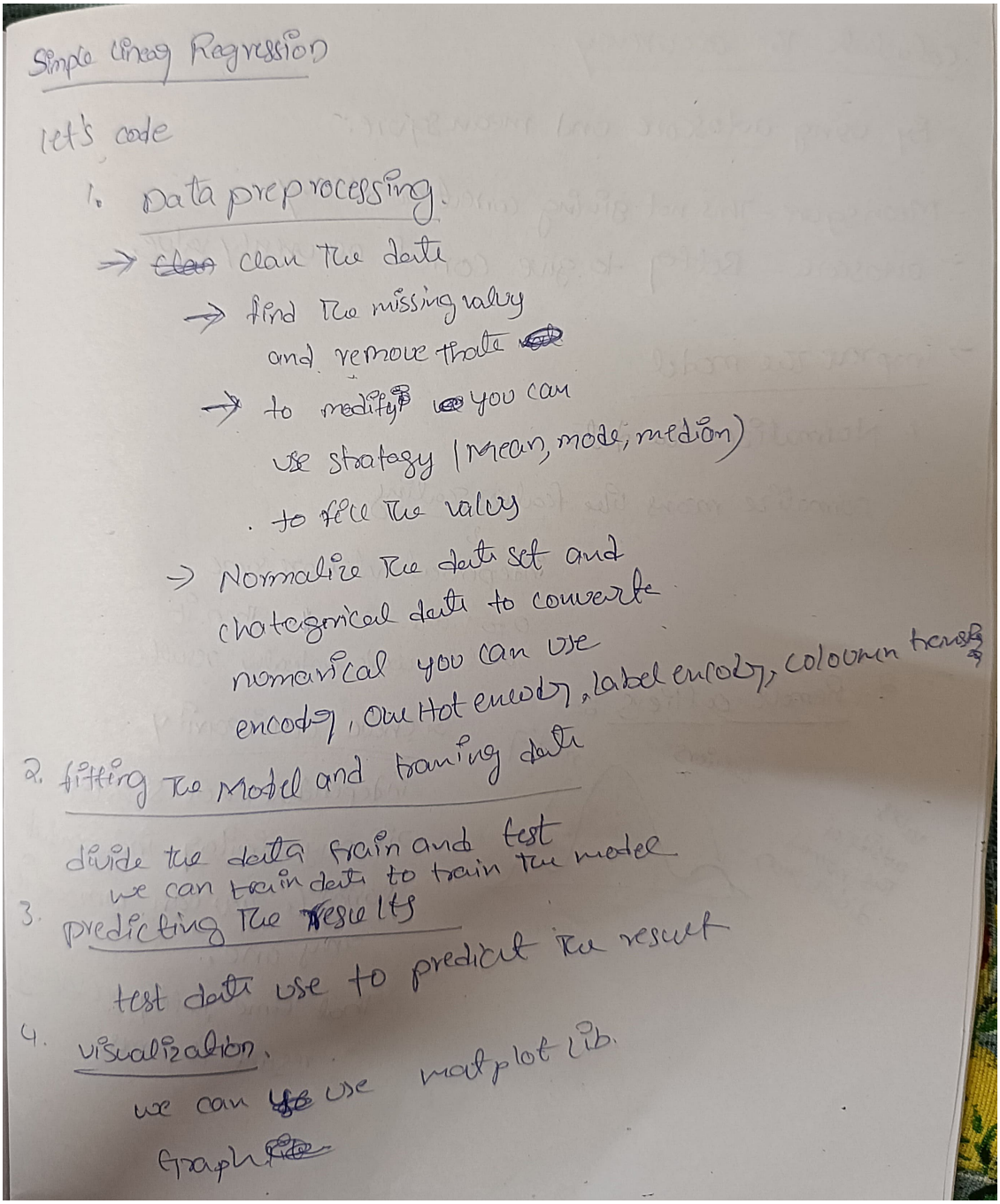
[28]:
```
Image.open("C:/Users/kamis/Pictures/Screenshots/Screenshot 2025-03-01 164012.
↪png")
```

[28]:



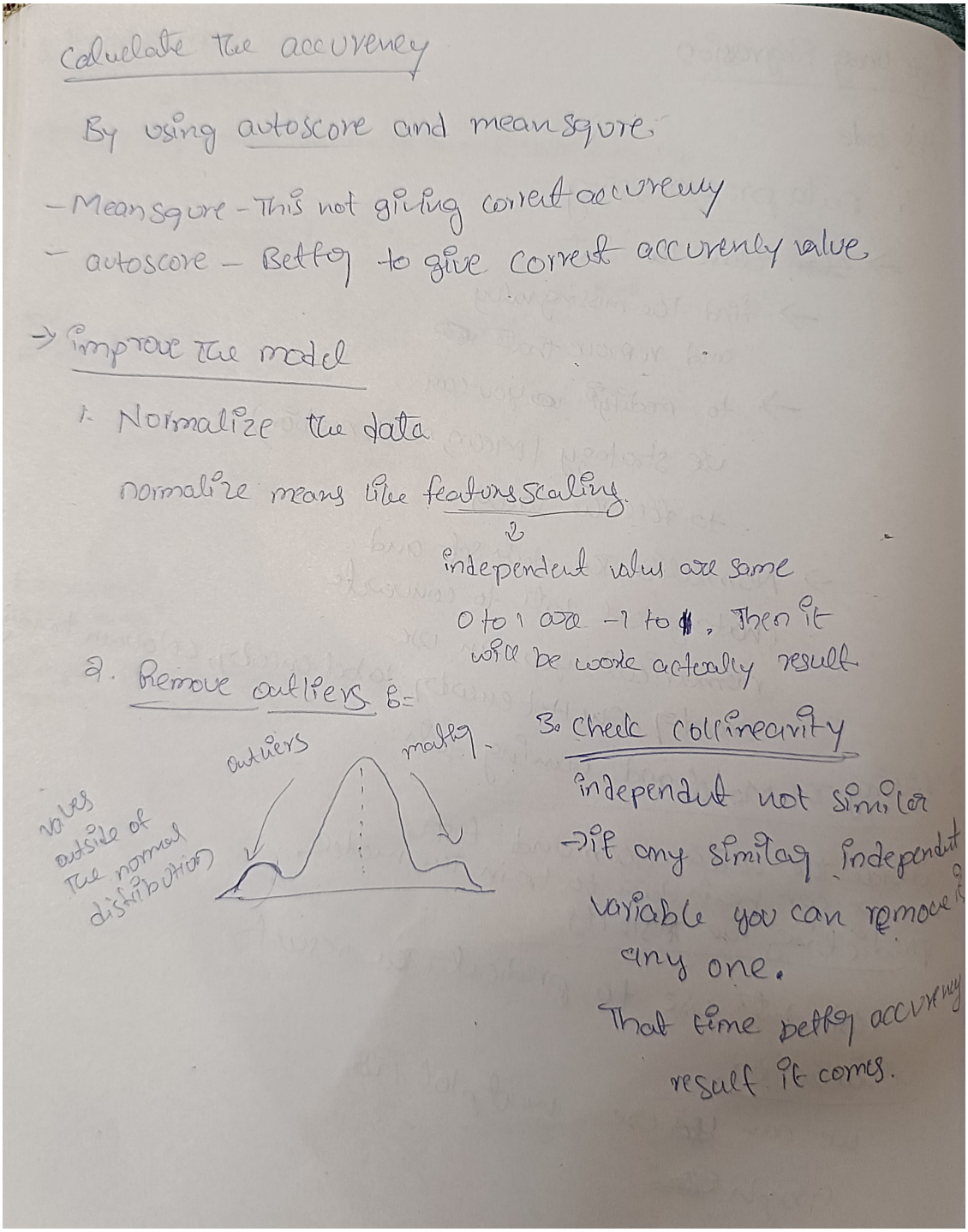
[34]:
```
Image.open("C:/Users/kamis/Pictures/SLR-2.jpg")
```

[34]:



[32]:
```
Image.open("C:/Users/kamis/Pictures/SLR-1.jpg")
```

[32]:



[1]:
```
import numpy as np # numpy using dealing with arrays
import pandas as pd # pandas using read dataset and manipulating data frames
import matplotlib.pyplot as plt # matplotlib using visulisation
```

[3]:
```
dataset=pd.read_csv("C:/Users/kamis/OneDrive/Desktop/ml-notes/Salary_Data.csv")
```

[5]:
```
dataset.head() #showing the top the dataset
```

[5]:
```
   YearsExperience  Salary
0              1.1   39343
1              1.3   46205
2              1.5   37731
3              2.0   43525
4              2.0   43525
```

[7]:
```
dataset.isna().sum() #any missing value you can use this
```

[7]:
```
YearsExperience    0
Salary             0
dtype: int64
```

[9]:
```
x=dataset.iloc[:,:1].values
y=dataset.iloc[:,-1].values
```

[11]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

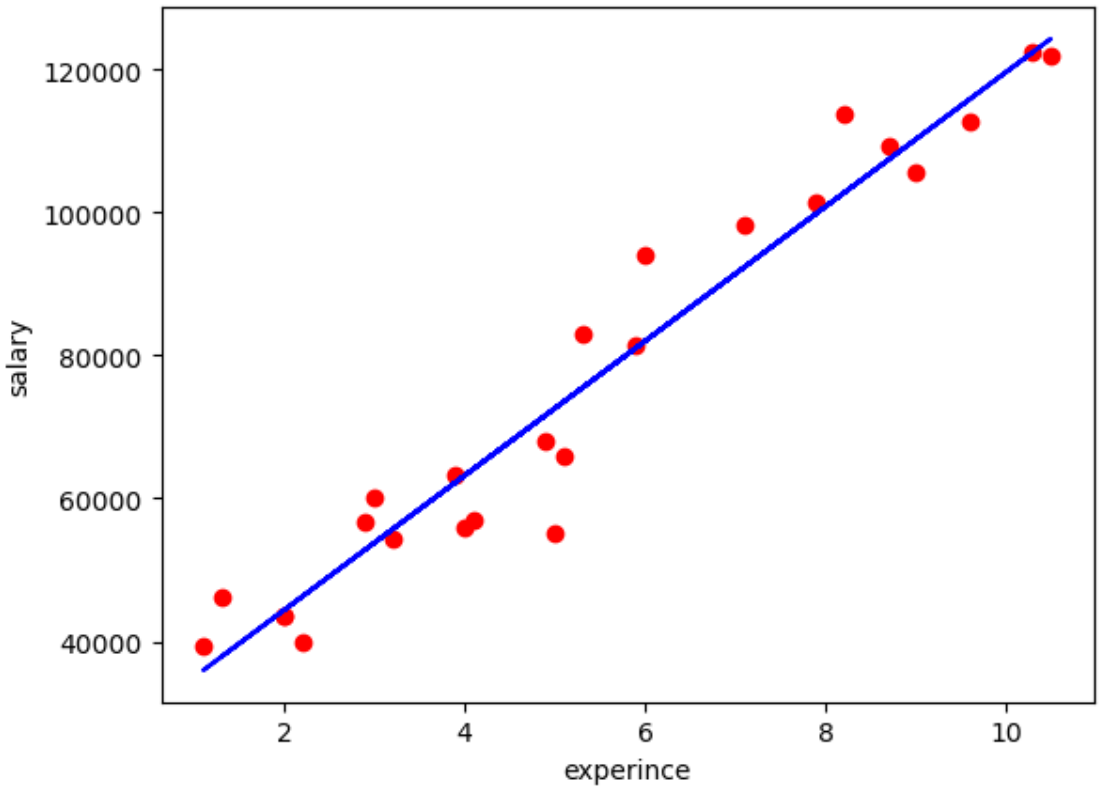
[13]:
```
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(x_train,y_train)
```

[13]:
```
LinearRegression()
```

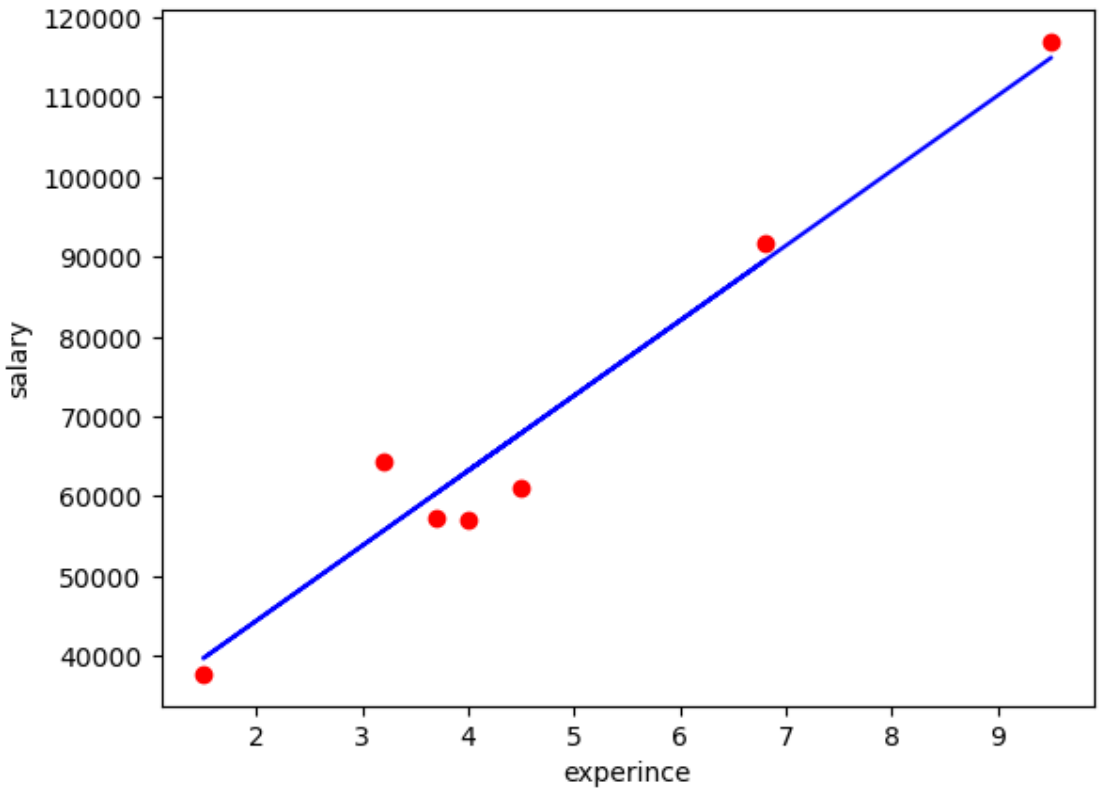
[15]:
```
y_pred=regressor.predict(x_test)
```

[21]:
```
import warnings
warnings.filterwarnings('ignore')
```

[31]:
```
plt.scatter(x_train,y_train, color='red')
plt.plot(x_train,regressor.predict(x_train) ,color='blue')
plt.xlabel('experince')
plt.ylabel('salary')
plt.show()
```



[33]:
```
plt.scatter(x_test,y_test,color='red')
plt.plot(x_test,regressor.predict(x_test),color='blue')
plt.xlabel('experince')
plt.ylabel('salary')
plt.show()
```


120000
110000
100000
90000
80000
70000
60000
50000
40000
salary
2
3
4
5
6
7
8
9
experince

[ ]:

[ ]:

[ ]:

[ ]: