In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

MSE

MSE stands for Mean Squared Error. It represents the average of the squared difference between the original and predicted values in the data set. It measures the variance of the residuals.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

In the above formula: error = Y_i - Y^

RMSE

Root Mean Squared Error is the square root of Mean Squared error. It measures the standard deviation of residuals.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

Where:

∑ is the summation of all values

yi is the predicted value

y^ is observed or actual value

(yi — y^) 2 are the differences between predicted and observed values and squared

N is the total sample size

In [2]:
```
X = np.array([1714,1664,1760,1685,1693,1936,1810,1987,1962,2050])
```
This is a comment
```
y = np.array([2.40,2.52,2.54,2.74,2.83,3.71,3.71,3.73,3.76,3.81])
```

In [3]:
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.4)
```

In [4]:
```
linear_model = LinearRegression()

model = linear_model.fit(X_train.reshape(-1,1),y_train.reshape(-1,1))
```

In [5]:
```
predicted_gpa=[]

for x in X_test:
    predicted_gpa.append(model.predict([[x]]))
```

In [6]:
```
predicted_gpa = np.array(predicted_gpa)
predicted_gpa
```

Out[6]:
```
array([[[3.52320255]],

       [[2.58770442]],

       [[2.55690613]],

       [[3.623297  ]]])
```

In [7]:
```
actual_gpa = y_test
actual_gpa
```

Out[7]:
```
array([3.71, 2.83, 2.74, 3.76])
```

In [8]:
```
def mse(actual_data,predicted_data):
    if(len(actual_data)!=len(predicted_data)):
        print("Error")

    error_squared=[]
    mse = None
    for i in range(0,len(predicted_data)):
        error_squared.append(((actual_data[i]-predicted_data[i]))**2)
        mse = (1/len(actual_data))*np.sum(error_squared)
    return mse
```

In [9]:
```
mse(actual_gpa,predicted_gpa)
```

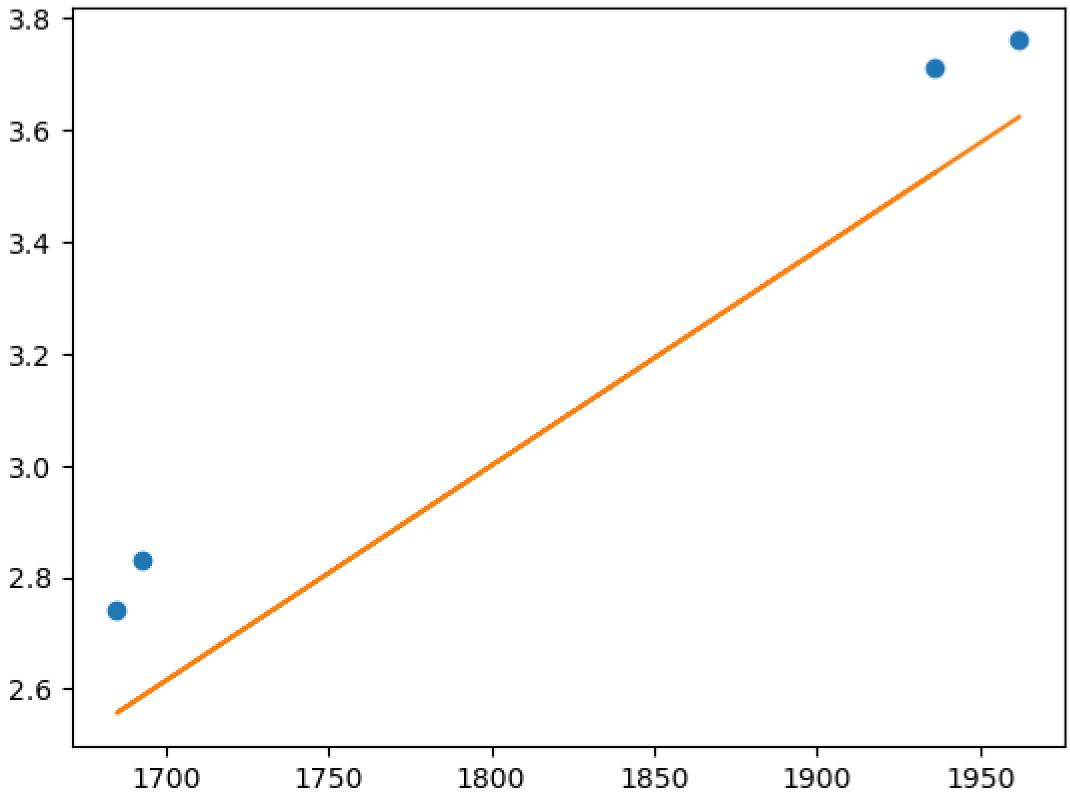
Out[9]:
```
0.03645287761129573
```

In [10]:
```
rmse = mse(actual_gpa,predicted_gpa)**0.5
rmse
```

Out[10]:
```
0.19092636698815524
```

In [11]:
```
plt.plot(X_test,y_test,"o")
plt.plot(X_test,model.predict(X_test.reshape(-1,1)))
```

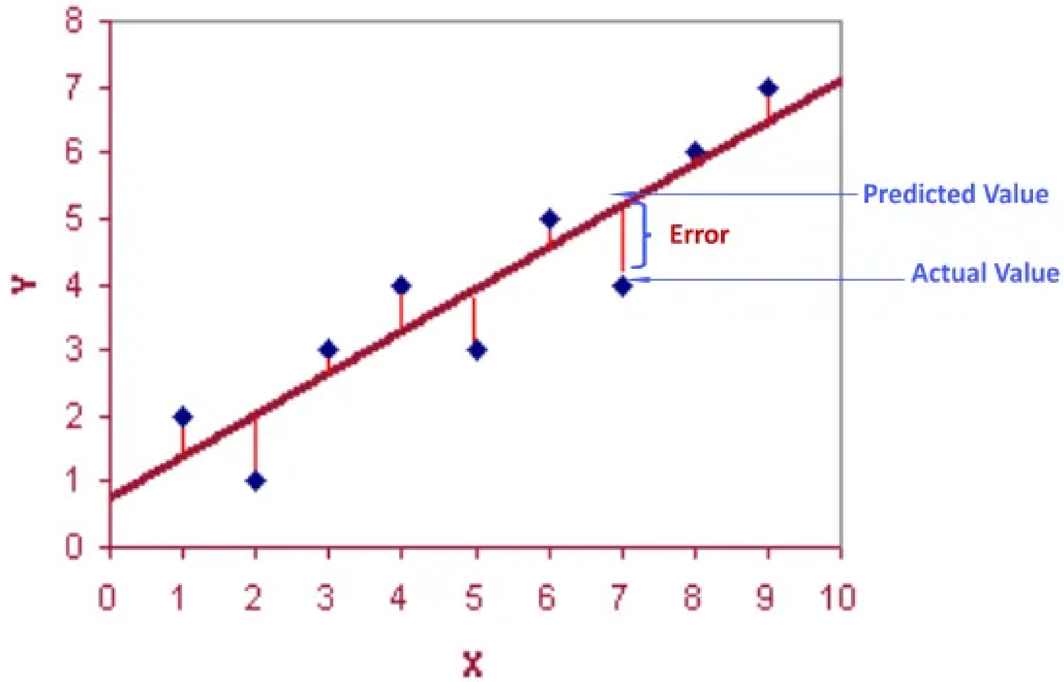
Out[11]: [<matplotlib.lines.Line2D at 0x7fa25032dbe0>]



Error = actual_value-predicted_value

Then when we calcuate error for every value and square it and calculate mean which is known as mean square error.

Incase of RMSE if we squre root the mse then we get rmse


Predicted Value
Error
Actual Value
Y
X
0
1
2
3
4
5
6
7
8
9
10