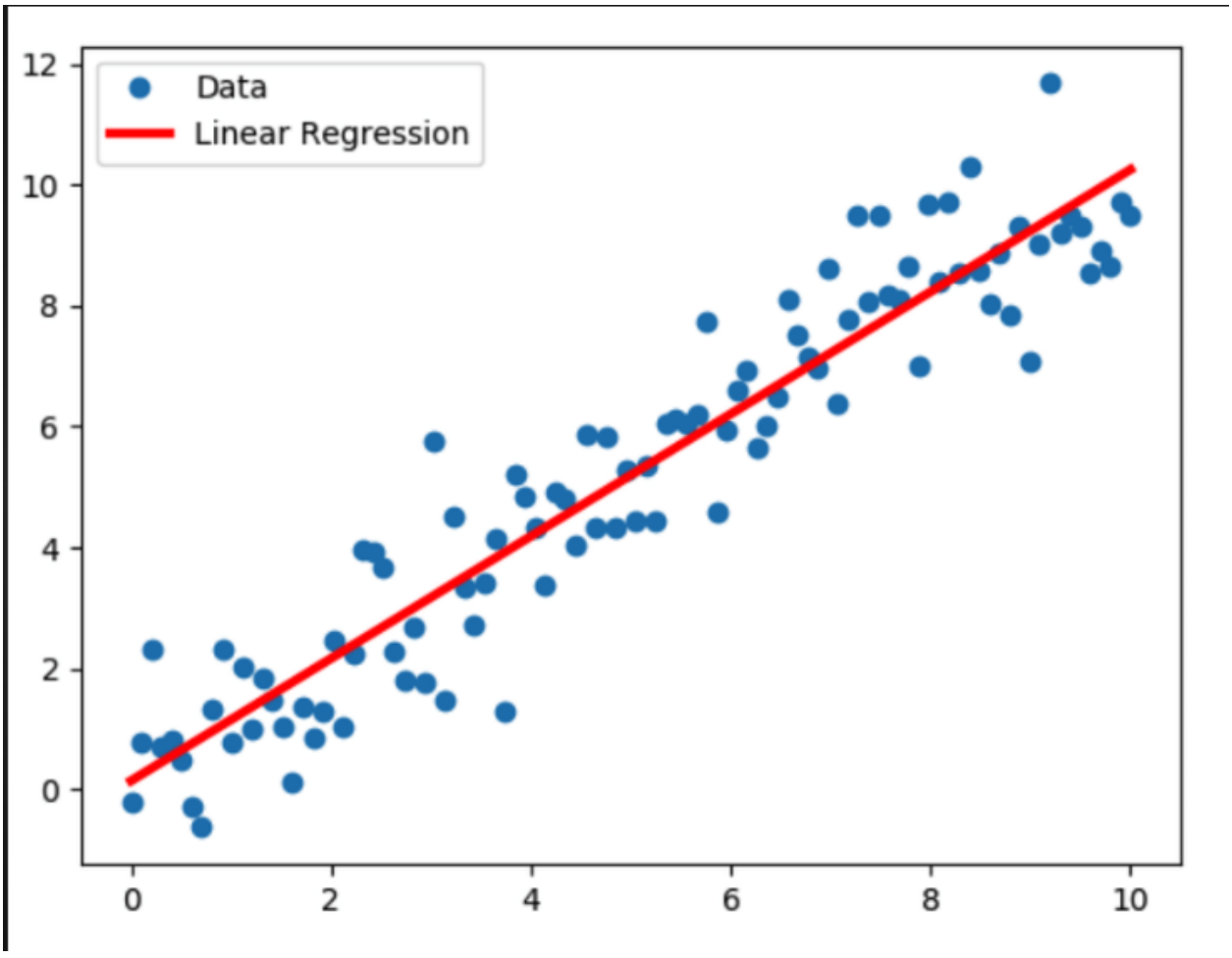
machine learning - linear regression

linear regression is a slope that is drawn in a graph between scattered datasets which predicts the event that is going to happen in the future with previous data's of a particular class.

for example:



we use sci-kit learn linear model to import linear regression in python .

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_diabetes
```

here we use the diabetes dataset , therefore..

```
diabetes = load_diabetes()
```

create a dataframe of the dataset

```
data= pd.DataFrame(diabetes.data , columns=diabetes.feature_names)
```

create a column for the target value

```
data["target"] = diabetes.target
```

now split the dataset as x and y

```
x = data.drop('target' , axis = 1)
y= data['target']
```

split the x and y datasets as testing and training datasets using train_test_split from sci-kit learn

```
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=42)
```

```
print("training shape : " , x_train.shape  , y_train.shape)
print("testing shape : " , x_test.shape , y_test.shape)
```

use the linearregression and fit the x and y training datasets

```
from sklearn.linear_model import LinearRegression

lr = LinearRegression()
lr.fit(x_train,y_train)

y_pred = lr.predict(x_test) #use the x_test to predict y values
```

now calculate r2 score , mse and mae using the sklearn metric library

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

r2 = r2_score(y_test , y_pred)
mse = mean_squared_error(y_test , y_pred)
mae = mean_absolute_error(y_test , y_pred)
intercept = lr.intercept_

print("r2 score : " , r2)
print("mean_squared_error : " , mse)
print(" mean_absolute_error : " , mae)
print("intercept : " , intercept)
```

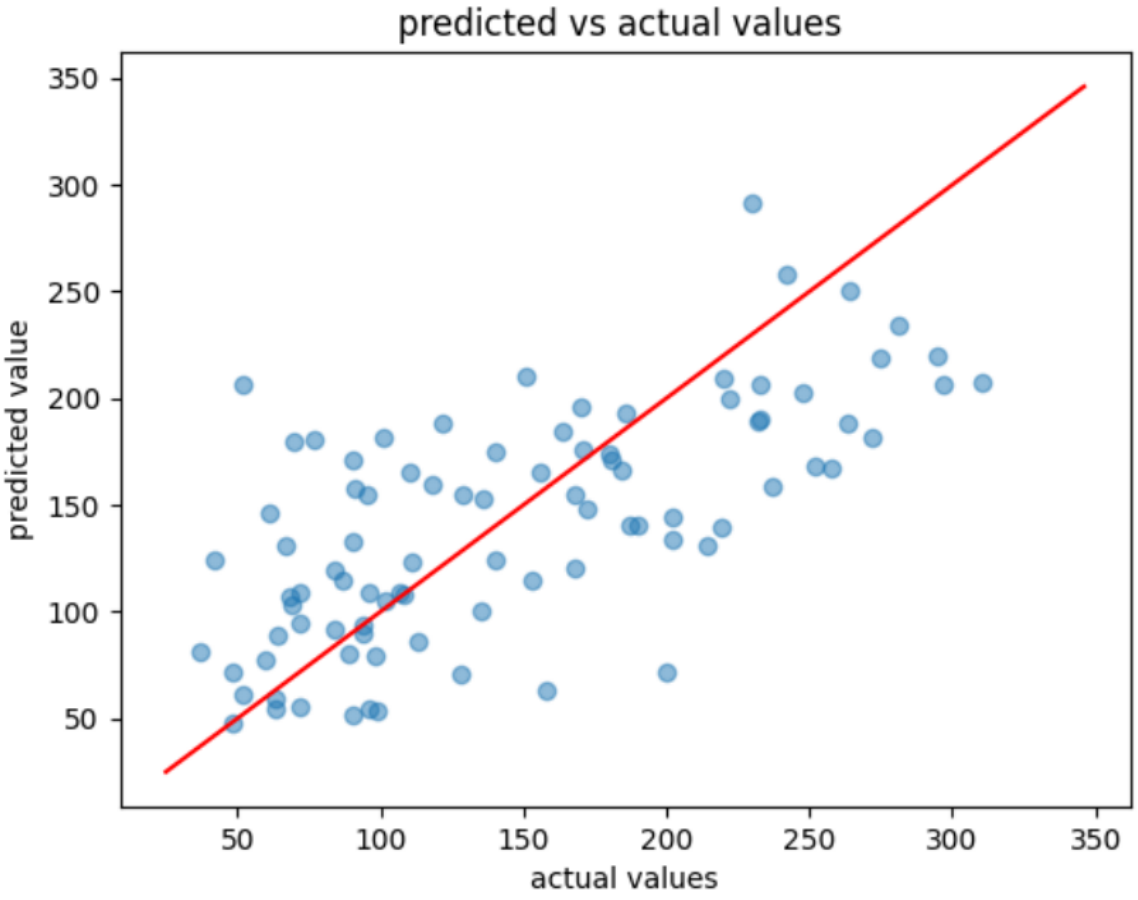
now plot the graph

```
import matplotlib.pyplot as plt

plt.scatter(y_test , y_pred , alpha=0.5)
plt.plot([y.min(),y.max()],[y.min(),y.max()] , color = "red")
plt.xlabel("actual values")
plt.ylabel("predicted value")
plt.title("predicted vs actual values " )
```

the output will be :

```
r2 score :  0.4526027629719195
mean_squared_error :  2900.1936284934814
 mean_absolute_error :  42.79409467959994
intercept :  151.34560453985995
```



this is the finaly linearly predicted graph .

end.