Practical - 3

Shivam Tawari A-58

Aim : Write a python program to calculate R squared.

Theory :

R- squared is a statistical measure of how close the data are to the fitted regression line.

The concept of R- squared is useful in feature selection.
Correlation (otherwise known as "R") is a number between 1 and -1 where a value of +1 implies than an increase in R results in some increase in a decrease in y, and 0 means that there isn't any relationship between x and y.

R^2 is the percentage of variation explained by the relationship between two variables.

The variance of the entire dataset is equal to the sum of the distance between every data point and the mean squared. The difference is squared such that the points below the mean don't cancel out with the points above the mean.

$$var(mean) = sum\,(pi - mean)^2$$

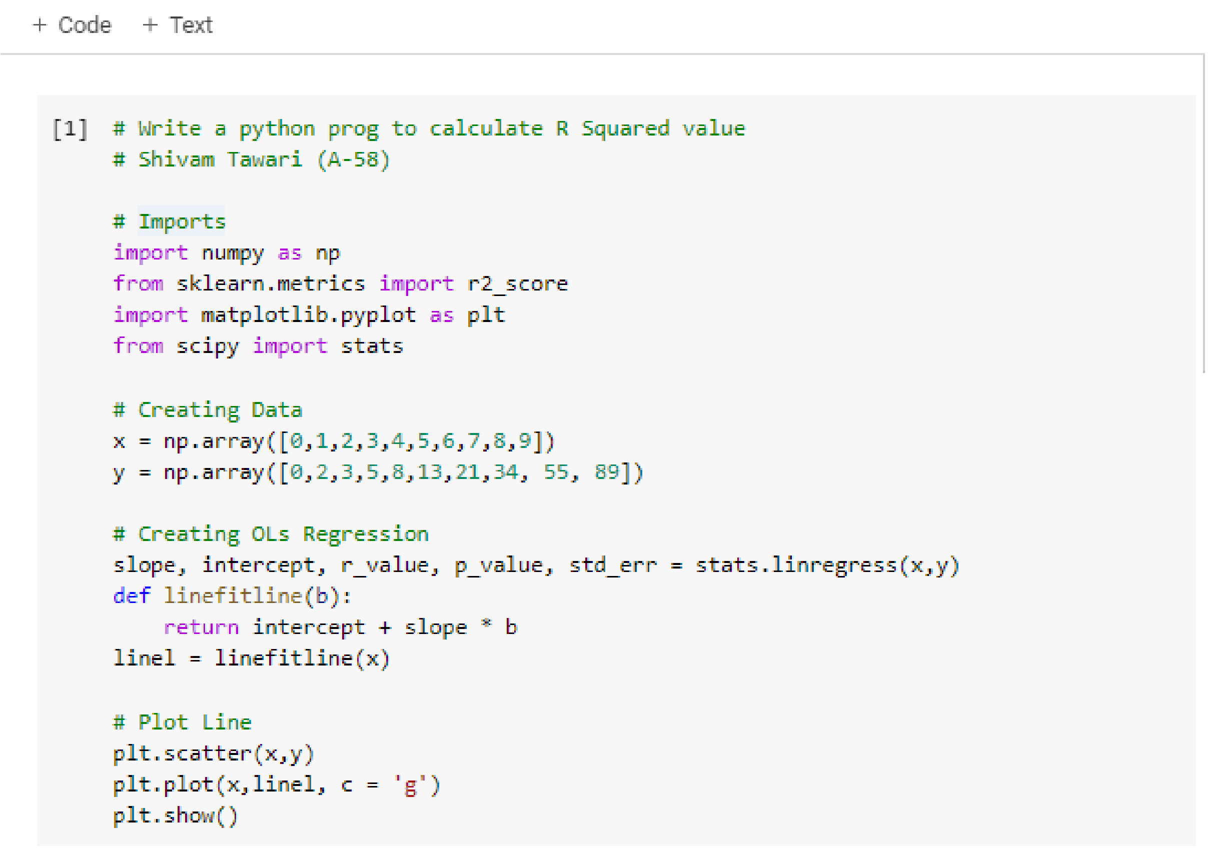
R-squared formula:

$$R^2 = \frac{\text{Total variation} - \text{Unexplained var}}{\text{Total variation}}$$

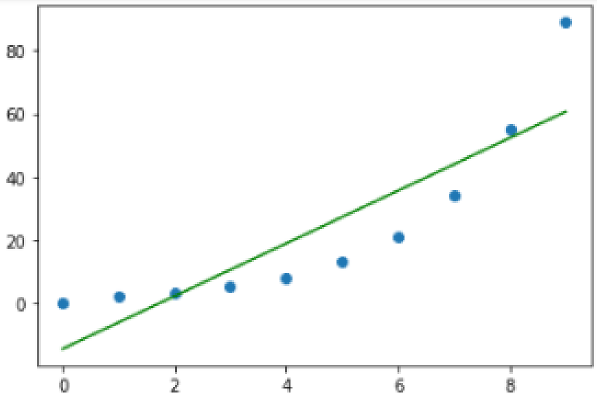
- Code:

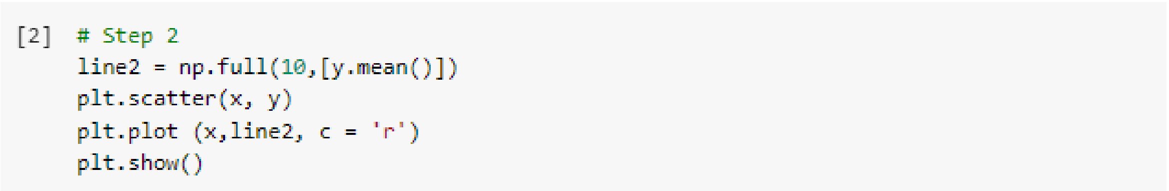
```
import numpy as np
import matplotlib.pyplot as plt.
from sklearn.metrics import r2-score
from scipy import stats

x = np.arrange (0, 10)
y = np.array ([0, 2, 3, 5, 8, 13, 21, 34, 55, 89])

Slope, intercept, r-value, p-value, std-err =
                              stats.linregress(x,y)

def linefitline (b):
    return intercept + slope xb

line1 = linefitline (x)

plt.scatter (x,y)
plt.plot (x, line1, e = 'g')
plt. show ()

line 2= np. full (10, [y. mean()])
plt. scatter (x, y)
plt. plot (x, line 2 , e = 'r')
plt. show ()

differences_line1 = linefitline (x) - y
np. sum (differences_line1 **2)

differences_line2 = line2 - y
np. sum (differences_line2 **2)

r2 = r2_score (y, linefitline (x))
print ('The R squared value is:', r2)
```

Conclusion: Hence, we have successfully implemented the calculation of R squared.

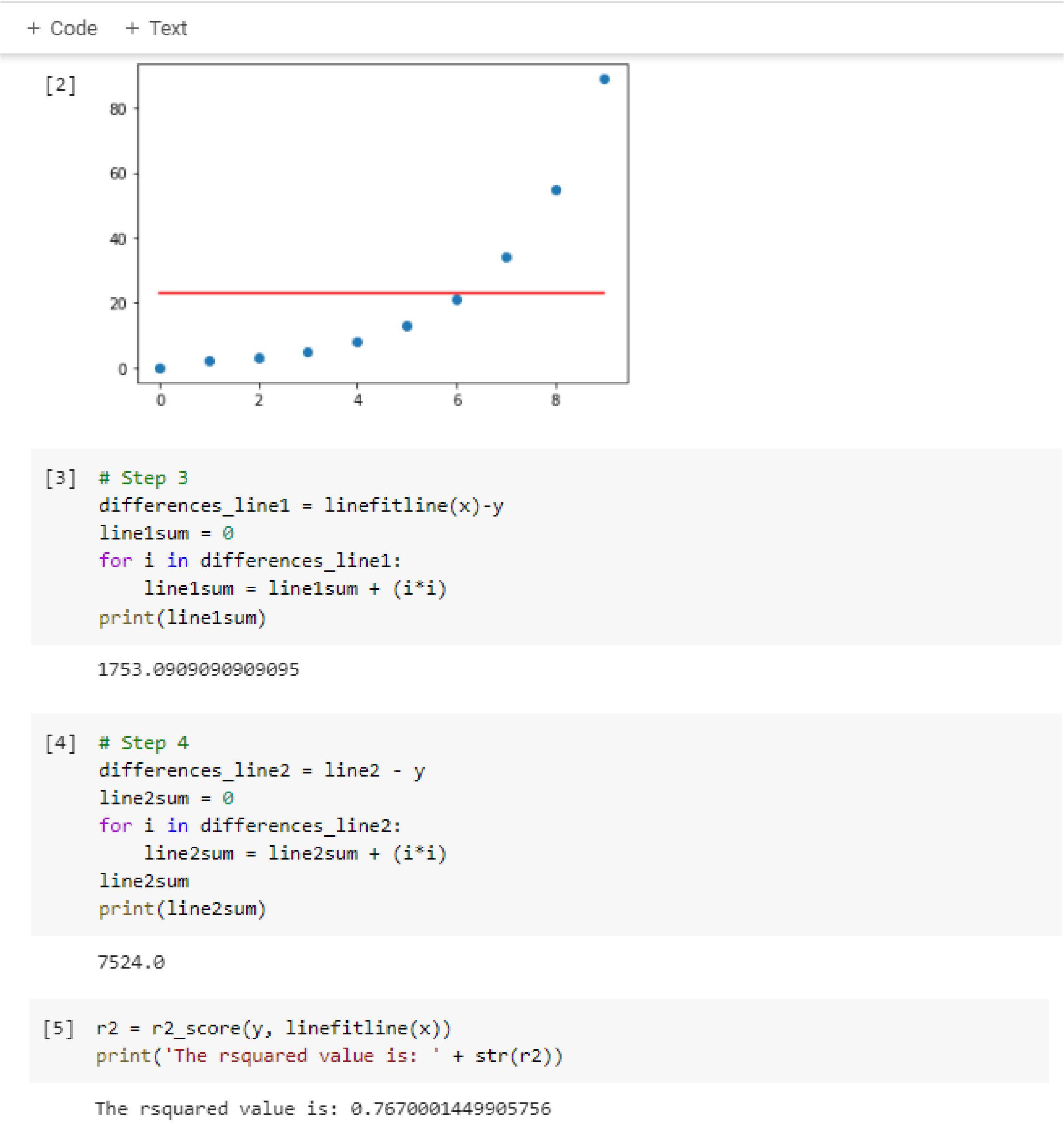


```
[1]  # Write a python prog to calculate R Squared value
     # Shivam Tawari (A-58)

     # Imports
     import numpy as np
     from sklearn.metrics import r2_score
     import matplotlib.pyplot as plt
     from scipy import stats

     # Creating Data
     x = np.array([0,1,2,3,4,5,6,7,8,9])
     y = np.array([0,2,3,5,8,13,21,34, 55, 89])

     # Creating OLs Regression
     slope, intercept, r_value, p_value, std_err = stats.linregress(x,y)
     def linefitline(b):
         return intercept + slope * b
     linel = linefitline(x)

     # Plot Line
     plt.scatter(x,y)
     plt.plot(x,linel, c = 'g')
     plt.show()
```





```
[2]  # Step 2
     line2 = np.full(10,[y.mean()])
     plt.scatter(x, y)
     plt.plot (x,line2, c = 'r')
     plt.show()
```



```
[3]  # Step 3
     differences_line1 = linefitline(x)-y
     line1sum = 0
     for i in differences_line1:
         line1sum = line1sum + (i*i)
     print(line1sum)
```

```
1753.0909090909095
```

```
[4]  # Step 4
     differences_line2 = line2 - y
     line2sum = 0
     for i in differences_line2:
         line2sum = line2sum + (i*i)
     line2sum
     print(line2sum)
```

```
7524.0
```

```
[5]  r2 = r2_score(y, linefitline(x))
     print('The rsquared value is: ' + str(r2))
```

```
The rsquared value is: 0.7670001449905756
```