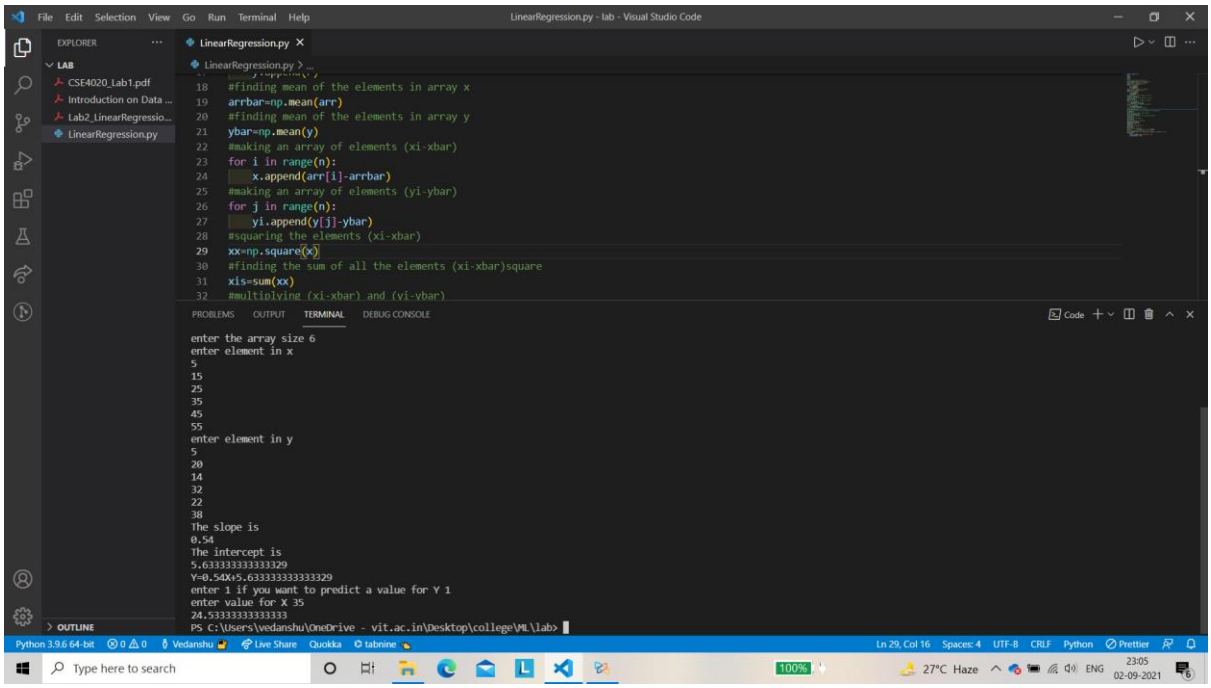
Linear Regression

VEDANSHU PATEL 20BCE0865

```
import numpy as np
x=[]
yi=[]
arr=[]
y=[]
# enter the length of the array
n=int(input("enter the array size "))
#entering the elements in column x
print("enter element in x ")
for i in range(n):
    q=int(input())
    arr.append(q)
#entering the elements in column y
print("enter element in y ")
for i in range(n):
    r=int(input())
    y.append(r)
#finding mean of the elements in array x
arrbar=np.mean(arr)
#finding mean of the elements in array y
ybar=np.mean(y)
#making an array of elements (xi-xbar)
for i in range(n):
    x.append(arr[i]-arrbar)
#making an array of elements (yi-ybar)
for j in range(n):
    yi.append(y[j]-ybar)
#squaring the elements (xi-xbar)
xx=np.square(x)
#finding the sum of all the elements (xi-xbar)square
xis=sum(xx)
#multiplying (xi-xbar) and (yi-ybar)
xy=np.multiply(x,yi)
#finding the summation of (xi-xbar)*(yi-ybar)
xys=sum(xy)
#calculating the slope
ans=xys/xis
#calculating the y intercept
www=ybar-ans*arrbar
print("The slope is")
print(ans)
```

```
print("The intercept is")
print(www)
#printing the equation
print('Y'+ '='+ str(ans)+'X' +'+'+ str(www))
#printing the predicted value of y for a x
zz=int(input("enter 1 if you want to predict a value for Y "))
if(zz==1):
    zzz=int(input("enter value for X "))
    print(ans*zzz + www)
```



In [1]:
```
!pip install -U scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\vedanshu\appdata\local
\programs\python\python39\lib\site-packages (0.24.2)
Requirement already satisfied: joblib>=0.11 in c:\users\vedanshu\appdata\local
\programs\python\python39\lib\site-packages (from scikit-learn) (1.0.1)
Requirement already satisfied: numpy>=1.13.3 in c:\users\vedanshu\appdata\local
\programs\python\python39\lib\site-packages (from scikit-learn) (1.21.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\vedanshu\appdat
a\local\programs\python\python39\lib\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: scipy>=0.19.1 in c:\users\vedanshu\appdata\local
\programs\python\python39\lib\site-packages (from scikit-learn) (1.7.1)

WARNING: You are using pip version 21.2.2; however, version 21.2.4 is availabl
e.
You should consider upgrading via the 'c:\users\vedanshu\appdata\local\programs
\python\python39\python.exe -m pip install --upgrade pip' command.
```

In [2]:
```
#Importing libraries and the class LinearRegression from sklearn
import numpy as np

from sklearn.linear_model import LinearRegression
import pandas as pd
```

In [3]:
```
#Providing data
x=np.array([5,15,25,35,45,55])
y=np.array([5,20,14,32,22,38])
print(x)
print(y)
```

```
[ 5 15 25 35 45 55]
[ 5 20 14 32 22 38]
```

In [4]:
```
#Reshaping the Data
x=np.array([5,15,25,35,45,55]).reshape((-1,1))
y=np.array([5,20,14,32,22,38])
print(x)
```

```
[[ 5]
 [15]
 [25]
 [35]
 [45]
 [55]]
```

In [5]:
```
print(y)
```

```
[ 5 20 14 32 22 38]
```

In [6]:
```
#Create a model and fit
#y=mx+C
model=LinearRegression()
model.fit(x,y)
```

Out[6]:
```
LinearRegression()
```

In [7]:
```
r_sq=model.score(x,y)
print('coefficient of determination:',r_sq)
```

```
coefficient of determination: 0.7158756137479542
```

In [8]:
```
#Printing the intercept
print('intercept:',model.intercept_)
```

```
intercept: 5.633333333333329
```

In [9]:
```
#Printing the slope
print('slope',model.coef_)
```

```
slope [0.54]
```

In [10]:
```
new_model=LinearRegression().fit(x,y.reshape((-1,1)))
print('intercept:',new_model.intercept_)
print('slope:',new_model.coef_)
```

```
intercept: [5.63333333]
slope: [[0.54]]
```

In [11]:
```
#printing the prediction
y_pred=model.predict(x)
print('predicted response:',y_pred, sep='\n')
print(x)
```

```
predicted response:
[ 8.33333333 13.73333333 19.13333333 24.53333333 29.93333333 35.33333333]
[[ 5]
 [15]
 [25]
 [35]
 [45]
 [55]]
```

In [12]:
```
y_pred=model.intercept_ + model.coef_ *x
print('predicted response:',y_pred,sep='\n')
```

```
predicted response:
[[ 8.33333333]
 [13.73333333]
 [19.13333333]
 [24.53333333]
 [29.93333333]
 [35.33333333]]
```

In [13]:
```
x_new= np.arange(5).reshape((-1,1))
print(x_new)
y_new=model.predict(x_new)
print(y_new)
```

```
[[0]
 [1]
 [2]
 [3]
 [4]]
[5.63333333 6.17333333 6.71333333 7.25333333 7.79333333]
```

In [ ]:
```

```