Experiment 4

Title:

Implement Regression model without using ML libraries.

Context:

Linear Regression is a type of predictive analysis algorithm that shows a linear relationship between the dependent variable(x) and the independent variable(y).

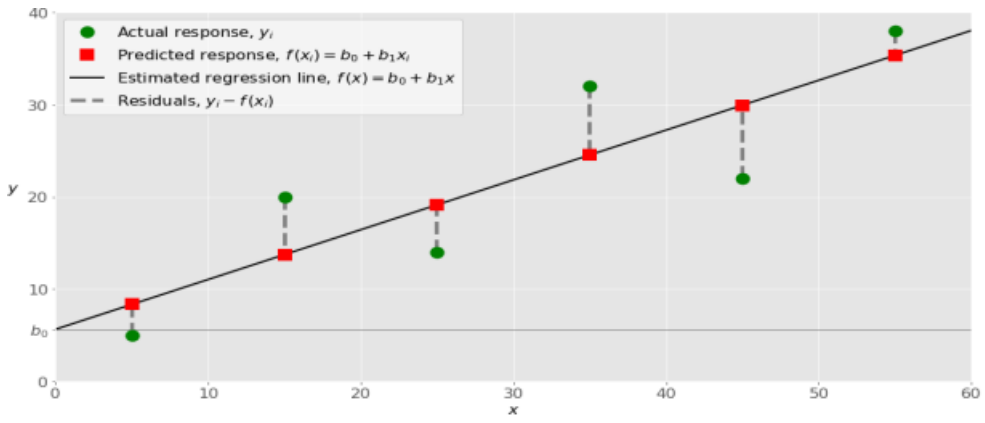
Based on the given data points, we try to plot a straight line that fits the points the best, the equation of a straight line is: $y = m*x + c$

x: input data points

y: dependent variable

m: bias or slope of the regression line

c: intercept, shows the point where the estimate



The slope(m) is calculated by the formula:

$$m = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Calculating the accuracy:

$$RSS = \sum_{i=1}^{n}(y_i - f(x_i))^2 \qquad \text{TSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

$$R^2 = 1 - \frac{RSS}{TSS}$$

Dataset Descrtiption:

This Visual Crossing global weather dataset provides easy access to decades of historical weather data, model-based 15-day forecasts, and long-range weather patterns based on statistical climate modeling. This allows our machine learning model to quickly and easily provide the weather data that you need for any project worldwide.

Code & Output:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
import operator
import io

from google.colab import files
uploaded=files.upload()
```

```
<IPython.core.display.HTML object>
```

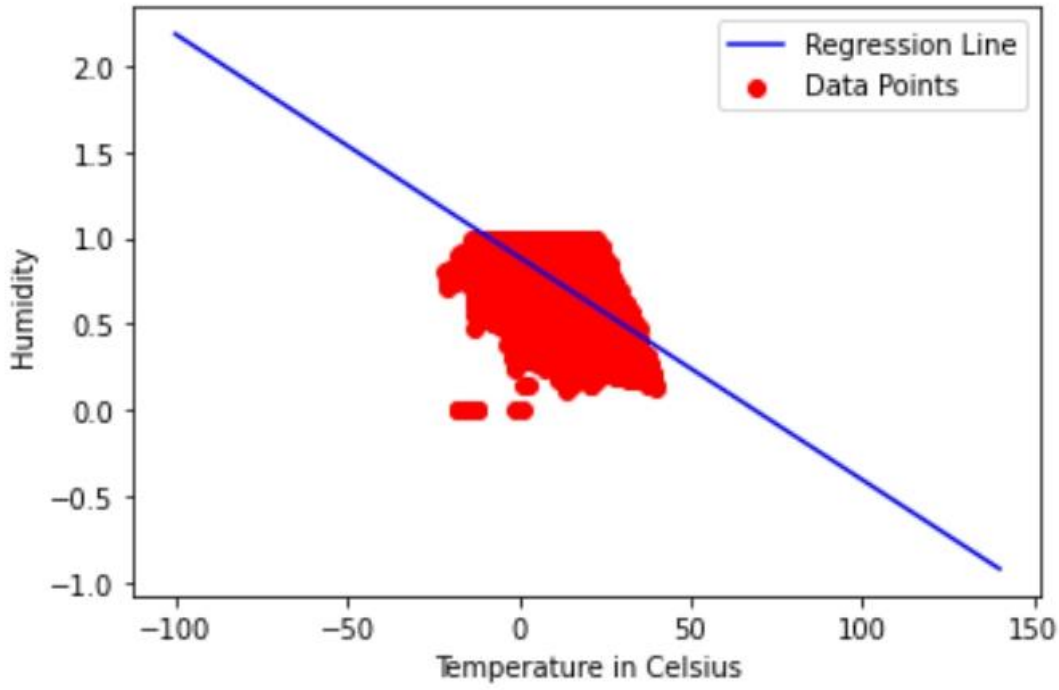
```
df=pd.read_csv(io.BytesIO(uploaded["weatherHistory.csv"]))
print(df.head())
```

```
                  Formatted Date        Summary  Temperature(C)  \
0  2006-04-01 00:00:00.000 +0200  Partly Cloudy        9.472222
1  2006-04-01 01:00:00.000 +0200  Partly Cloudy        9.355556
2  2006-04-01 02:00:00.000 +0200  Mostly Cloudy        9.377778
3  2006-04-01 03:00:00.000 +0200  Partly Cloudy        8.288889
4  2006-04-01 04:00:00.000 +0200  Mostly Cloudy        8.755556

   Apparent Temperature (C)  Humidity  Wind Speed (km/h)
0                  7.388889      0.89            14.1197
1                  7.227778      0.86            14.2646
2                  9.377778      0.89             3.9284
3                  5.944444      0.83            14.1036
4                  6.977778      0.83            11.0446
```

```
x=df['Temperature(C)'].values
y=df['Humidity'].values
mean_of_x=np.mean(x)
mean_of_y=np.mean(y)

count=len(x)
k=0
l=0

for i in range(count):
  k+=(x[i]-mean_of_x)*(y[i]-mean_of_y)
  l+=(x[i]-mean_of_x)**2
m=k/l
c=mean_of_y-(m*mean_of_x)
print("m=",m,"\n")
print("c=",c,"\n")
```

```
m= -0.012939114669653631

c= 0.8892972609551082
```

```
max_of_x=np.max(x)+100
min_of_x=np.min(y)-100
X=np.linspace(min_of_x,max_of_x,100)
Y=c+m*X
plt.plot(X, Y, color='blue', label='Regression Line')
plt.scatter(x, y, color='red', label='Data Points')
plt.xlabel('Temperature in Celsius')
plt.ylabel('Humidity')
plt.legend()
plt.show()
```



```
s=0
p=0
for i in range(int(count)):
  y_predict=c+m*x[i]
  s+=(y[i]-mean_of_y)**2
  p+=(y_predict-mean_of_y)**2
r2=p/s
print(r2)
```

```
0.39974597409263735
```