תרגול פרק 4 - רגרסיה לינארית על ידי Gradient Descent

תרגיל 1:

הטבלה הבאה מציגה נתונים מימיה הראשונים של חברת ביגוד איטלקית בשם בנטון. כל שורה בטבלה מציגה את המכירות של בנטון במשך שנה ואת הסכום שהוצא על פרסום באותה שנה.

Year	Sales (Million Euro)	Advertising (Million Euro)
1	651	23
2	762	26
3	856	30
4	1,063	34
5	1,190	43
6	1,298	48
7	1,421	52
8	1,440	57
9	1,518	58

ממשו מכונה לומדת מבוססת **Gradient Descent** שתענה על השאלה הבא: אם הוצאות הפרסום של חברת בנטון יוגדלו במיליון יורו, בכמה צפויים לגדול מכירות החברה?

לצורך מימוש התרגיל נבצע את השלבים הבאים:

- בשלב הראשון נקלוט את נתוני הטבלה לתוך מערך numpy.
- נפעיל את הפעולה GradientDescent כדי לקבל את הפונקציה הלינארית y=mx+b
- נחשב את צפי המכירות בייחס להשקעה של מיליון יורו בפרסם.

לרשותכם הפעולה GradientDescent כפי שלמדנו.

```
def GradientDescent(x,y,learning_rate=0.01, epochs=20):
    m=0
    b=0
    for _ in range(epochs):
        for i in range(len(x)):
            xi = x[i]
            yi = y[i]
            guess = m * xi + b
            error = guess - yi
            m = m - (error * xi) * learning_rate
```

```
        b = b - error * learning_rate
return m,b
```

טיפים:

ערכים גדולים מדי מביאים אתכם לחישובים במספרים גדולים מאוד. ניתן לחלק את כלל נתונים בערך קבוע. מצד אחד נשמור את היחס בין הנתונים ומצד שני נחשב מספרים קטנים יותר.

חשוב למצוא ערכים אופטימליים ל- learningRate ו- epochs. האם תצליחו למצוא דרך לעשות זאת?