

```
In [ ]: from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

### მარტივი წრფივი რეგრესია

ამ მაგალითში გამოვსახავთ მარტივი წრფივი რეგრესიის ამოცანას კლასიკური განტოლების  $y=mx+b$  გამოყენებით. ყურადღებით გავივლით ამ პროცესის თითოეულ საფეხურს.

### imports

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [ ]: # df=pd.read_csv("Advertising.csv")
df=pd.read_csv("/content/drive/MyDrive/dataset-students/Advertising.csv")
```

```
In [ ]: df.head()
```

Out[4]:

	TV	radio	newspaper	sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

```
In [ ]: df.tail()
```

Out[5]:

	TV	radio	newspaper	sales
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	9.7
197	177.0	9.3	6.4	12.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	13.4

არის თუ არა კავშირი საერთო სარეკლამო დანახარჯებსა და გაყიდვებს შორის?

```
In [ ]: ▶ df["total_spend"] = df["TV"] + df["radio"] + df["newspaper"]  
df
```

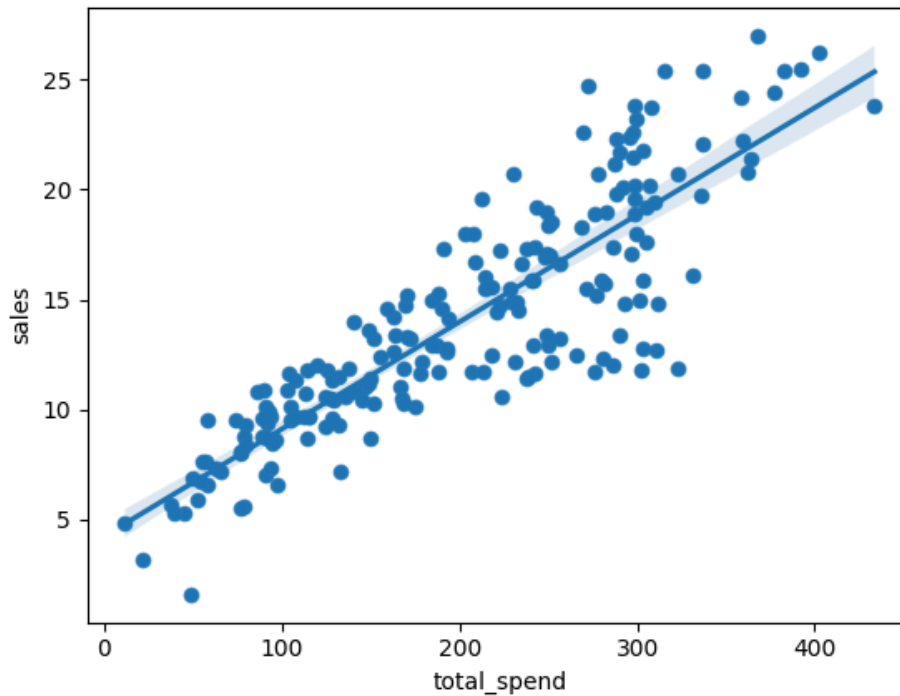
```
Out[6]:
```

	TV	radio	newspaper	sales	total_spend
0	230.1	37.8	69.2	22.1	337.1
1	44.5	39.3	45.1	10.4	128.9
2	17.2	45.9	69.3	9.3	132.4
3	151.5	41.3	58.5	18.5	251.3
4	180.8	10.8	58.4	12.9	250.0
...	...	...	...	...	...
195	38.2	3.7	13.8	7.6	55.7
196	94.2	4.9	8.1	9.7	107.2
197	177.0	9.3	6.4	12.8	192.7
198	283.6	42.0	66.2	25.5	391.8
199	232.1	8.6	8.7	13.4	249.4

200 rows × 5 columns

```
In [ ]: sns.scatterplot(x="total_spend", y="sales", data=df)
sns.regplot(x="total_spend", y="sales", data=df)
```

Out[7]: <Axes: xlabel='total\_spend', ylabel='sales'>



### უმცირესი კვადრატების მეთოდი

\*\* რა კითხვაზე პასუხობს წრფივი რეგრესია. \*\*

**პროგნოზირება:** თუ ვინმე გეგმავს 200 დოლარის დახარჯვას რეკლამაში, გავიგოთ როგორი გაყიდვები შეიძლება ჰქონდეს მას მომავალში? ჩვენ გავამარტივეთ პრობლემას ყველა ინდივიდუალური ნიშნის გაერთიანებით ერთ მახასიათებელში „სარეკლამო ჯამური ხარჯები“. მრავალფუნქციურ პრობლემას ცოტა მოგვიანებით დავუბრუნდებით.

ამისათვის ჩვენ უნდა გადავწყვიტოთ შემდეგი განტოლება:  $y = ax + b$

ვეძებთ  $a$  და  $b$  კოეფიციენტებს და ვხსნით განტოლების ზოგად ფორმას: მოცემული გვაქვს მნიშვნელობების მატრიცები - ცნობილია გაყიდვების მატრიცა  $y$  და ცნობილია რეკლამის დანახარჯის მატრიცა  $X$ . უცნობი არის  $a$  და  $b$  კოეფიციენტები, ხოლო მატრიცების შემთხვევაში ეს არის არაერთი რიცხვი, არამედ რამდენიმე რიცხვი. ამრიგად, ჩვენ ვძუშაობთ არა ერთი, არამედ რამდენიმე ნიშნით. ეს კოეფიციენტები ასევე დაგვცხმარება გავიგოთ, რომელ მახასიათებელს აქვს ყველაზე დიდი უნარი, წინასწარ განსაზღვროს სამიზნე  $y$  მნიშვნელობა. აქ, ზემოთ მონიშნული ნიშნით არის სამიზნე მნიშვნელობის შეფასება (პროგნოზირება), ხოლო  $y$  არის ჭეშმარიტი მნიშვნელობა, რომელიც ცნობილია ზოგიერთი პუნქტისთვის. ჩვენ შეგვიძლია გამოვიყენოთ NumPy განტოლების ამოსახსნელად.

```
In [ ]: x = df['total_spend']  
        y = df['sales']
```

```
In [ ]: # help(np.polyfit)  
        # Nump-ში არის გამზადებული ფუნქცია - polyfit(), რომელიც გამოიყენება პოლინომიური ფუნქციის ფარგლებში.  
        # ის იყენებს უმცირეს კვადრატების მეთოდს ფუნქციის პოლინომის ასაგებად.  
        # პოლინომი ნიშნავს, რომ შეგვიძლია მივუთითოთ არა მარტო  $X$ , არამედ  $X$ -ის უფრო მაღალი ხარისხი.
```

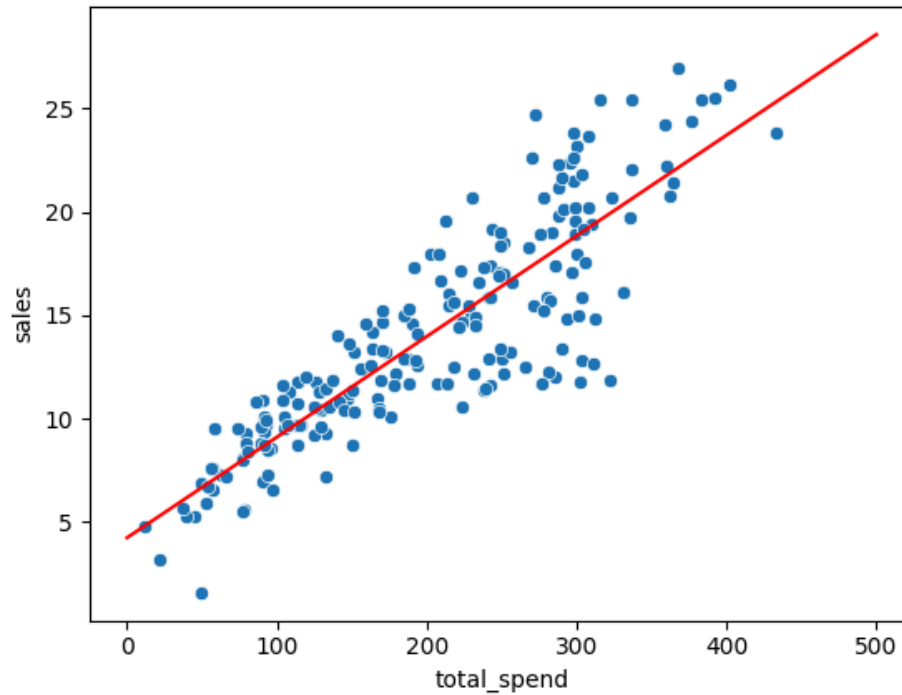
```
In [ ]: # უპირველეს ყოვლისა გამოვთვალოთ კოეფიციენტები (პოლინომის ხარისხები)!  
        np.polyfit(x, y, 1)
```

```
Out[10]: array([0.04868788, 4.24302822])
```

```
In [ ]: # ბიუჯეტის პოტენციური ხარჯვა მომავალში  
        potential_spend = np.linspace(0,500,100)
```

```
In [ ]: ▶ predicted_sales = 0.04868788 * potential_spend + 4.24302822
sns.scatterplot(x='total_spend',y='sales',data=df)
plt.plot(potential_spend,predicted_sales,color='red')
```

Out[20]: [ <matplotlib.lines.Line2D at 0x79023b7c0b50> ]



მომდევნო სარეკლამო კამპანიაში მხოლოდ 200 დოლარის დახარჯვას ვგეგმავთ. რამდენი ერთეული პროდუქტის გაყიდვას ველოდებით ამ რეკლამით?

```
In [ ]: ▶ spend = 200
predicted_sales = 0.04868788 * spend + 4.24302822
predicted_sales
```

Out[21]: 13.98060422

სურვილის შემთხვევაში შეგვიძლია გამოვიყენოთ უფრო მაღალი რიგის პოლინომები, მაგრამ გაუმჯობესდება თუ არა პროგნოზი? არის სიტუაციები, როცა მოდელი ძალიან კარგად დატრენინგებულია ან პირიქით. როგორ განვსაზღვროთ ტრენინგი კარგად არის ჩატარებული თუ ცუდად?

```
In [ ]: # y = B1 * x + B0

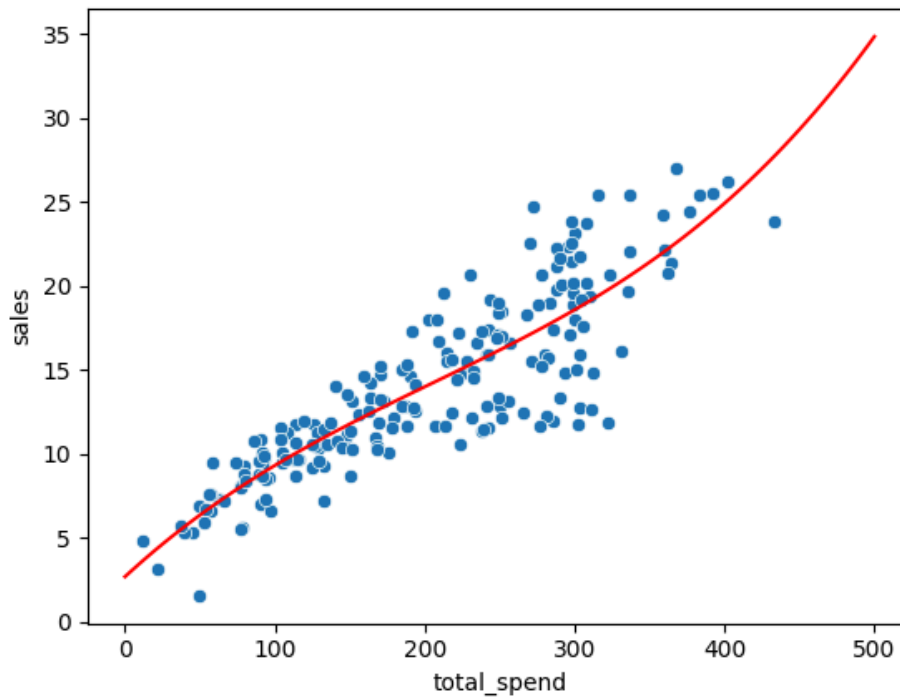
# y = B3 * x^3 + B2 * x^2 + B1 * x + B0
np.polyfit(x,y,3)
```

Out[15]: array([ 3.07615033e-07, -1.89392449e-04, 8.20886302e-02, 2.70495053e+00])

```
In [ ]: # ბიუჯეტის პოტენციური ხარჯვა მომავალში
potential_spend= np.linspace(0,500,100)
```

```
In [ ]: predicted_sales=3.07615033e-07*potential_spend**3 + \
-1.89392449e-04*potential_spend**2 + \
8.20886302e-02*potential_spend + \
2.70495053e+00
sns.scatterplot(x='total_spend',y='sales',data=df)
plt.plot(potential_spend,predicted_sales,color='red')
```

Out[23]: [matplotlib.lines.Line2D at 0x79023b62e170]



გრაფიკზე ჩანს, რომ არ არის წრფივად დამოკიდებული.

საწყისი მონაცემებში არის რამდენიმე მახასიათებელი (3) ნიშანი და არა რეკლამის მთლიანი დანახარჯი. შეგვიძლია გავიმეოროთ პროცესი და მივიღოთ უფრო ზუსტი

```
In [ ]: ▶ x = df[["TV", "radio", "newspaper"]]
        y = df['sales']
```

```
In [ ]: ▶ # აქ ჩვენ X მასივში გადავცემთ სამ ცალკეულ მახასიათებელს
        np.polyfit(x,y,1)
        # ეს ბრძანება გამოიწვევს შეცდომას - იხილეთ კომენტარი ქვემოთ
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-23-7b257a9170d4> in <cell line: 2>()
      1 # აქ ჩვენ X მასივში გადავცემთ სამ ცალკეულ მახასიათებელს
----> 2 np.polyfit(x,y,1)
      3 # ეს ბრძანება გამოიწვევს შეცდომას - იხილეთ კომენტარი ქვემოთ

/usr/local/lib/python3.10/dist-packages/numpy/core/overrides.py in polyfit(*args, **kwargs)

/usr/local/lib/python3.10/dist-packages/numpy/lib/polynomial.py in polyfit(x, y, deg, rcond, full, w, cov)
    634         raise ValueError("expected deg >= 0")
    635     if x.ndim != 1:
--> 636         raise TypeError("expected 1D vector for x")
    637     if x.size == 0:
    638         raise TypeError("expected non-empty vector for x")

TypeError: expected 1D vector for x
```

Polyfit() მუშაობს მხოლოდ ერთგანზომილებიანი X მასივით! ჩვენ უნდა გადავიდეთ უფრო მძლავრ ბიბლიოთეკაში...