گروه 7
درین رستمی – 97243034
شهریار مربی – 97243064

گزارش تمرین دوم:

در تابع main بعد از اینکه داده index شده باشد با وزن ها و تتاهای رندوم (برای تابع manual) 50 تا نمونه از نتایج بدست آمده را با فایل qrel خود مقایسه میکنیم.

```
random_theta = [1, 3, 2, 9]
samples_count = 50
x = []
y = []
for i in range(samples_count):
    x.append([int(random.random() * 10) for _ in range(4)])
for i in range(samples_count):
    y.append(int(evaluate(x[i])['query']['map'] * 10000))

print("starting ez optimization\n")
optimize(x, y, random_theta)
print("starting manual optimization\n")
manual_optimize(x, y, random_theta)
```

تابع اول که optimize نام دارد، از تابع optimize در ai.ez استفاده میکند که در آن با gradient descent موجود در کتابخانه sklearn و با کمک scipy به ما بهترین وزن هایی که بدست آورده است را نشان میدهد. (ez=easy چون از کتابخانه های از قبل درست شده استفاده کردیم =)))

تابع دوم که manual_optimize نام دارد با gradient descent ای که خودمان در فایل ai.manual پیاده سازی کرده ایم این کار را انجام میدهد به طوری که با تابع minibatch_gradient_descent تتاهای خوب و با find_extremum بهترین x ها (همان چیزی که ما میخواهیم) را پیدا میکند. (ما اول این تابع را نوشته بودیم و بعد که گفتید میشه از کتابخانه های دیگر استفاده کرد فایل ez را اضافه کردیم و گفتیم آن روش رو هم امتحان کنیم و در آخر هر دو روش را نگه داریم)