

PYTHON DASTURLASH TILIDA CHIZIQLI REGRESSIYA MASALALARINI YECHISH

Muhammademinov Alijon Azizjon o'g'li

Andijon davlat universiteti talabasi

Annotatsiya: *Bizga ma'lumki dasturlash tillarining ichida Python dasturlash tilining o'rni nihoyatda katta. Bundan tashqari Python dasturlash tili yordamida ko'plab sun'iy intellekt masalalari ham yechiladi. Ularning ichidagi eng asosiy mavzulardan biri bu Chiziqli regressiya bo'lib uning yordamida sun'iy intellektning bir qator masalalari hal qilinadi. Bu mavzu ham aynan shunga qaratilgan.*

Kalit so'zlar: *Chiziqli regressiya, Python tili, kutubxonalar, numpy, chiziqli funksiya, qiymatlarning bog'lanishi, sun'iy intellekt, datasetlar.*

Dastlab chiziqli regressiya haqida umumiy tushunchalar bilan tanishib chiqish kerak. Chiziqli regressiya tushunchasi haqida asosiy tushunchalar bilan "Chiziqli algebra" fanida chuqur o'rganish mumkin.

Regressiya - biror tasodifiy miqdor o'rta qiymatining boshqa bir tasodifiy miqdoriga bog'liqligi. Funktsional bog'lanishda miqdorning har bir qiymatiga u miqdorning bitta qiymati mos kelsa, regressiv bog'lanishda x ning y qiymatiga turli hollarda turli qiymatlar mos keladi. Regressiya tushunchasi faqat tasodifiy miqdorlar uchun emas, tasodifiy vektorlarga nisbatan ham kiritiladi. Regressiya iborasini ingliz olimi F.Galton kiritgan (1886-yil). Regressiya tabiat va jamiyat hodisalarini o'rganishda muhim matematik vosita hisoblanadi.

Yuqoridagi ma'lumotlardan foydalanib aytish mumkinki chiziqli regressiya o'z-o'zidan chiziqli funksiya bilan bog'liqdir. Bunda aynan $y = mx + n$ chiziqli funksiya tenglamasidan foydalaniladi. Buni Python dasturlash tili bilan bog'lab biron masalani yechib ko'raylik va shu orqali mavzuni yoritaylik.

Bu kabi masalalar uchun Numpy kutubxonasi katta yordam beradi. Quyidagi misol chiziqli regressiya masalasini yechish uchun bir nechta funksiyalarni qamrab oladi:

```
import numpy as np
```

```
# X va y ni tayyorlash
```

```
X = np.array([[1, 1], [1, 2], [2, 2], [2, 3]])
```

```
y = np.dot(X, np.array([1, 2])) + 3
```

```
# chiziqli regressiya modelini yaratish va bajarish
```

```
reg = np.linalg.lstsq(X, y, rcond=None)[0]
```

```
# natijani chiqarish
```

```
print(reg)
```

Yuqoridagi kod yordamida aynan shu mavzu masalasini ko'rish mumkin. Natija esa konsol oynasida ko'rinadi. Yoki yana bir misolni ko'rib chiqaylik:

```
import numpy as np

# ma'lumotlar
x1 = [0.1, 0.2, 0.3, 0.4, 0.5]
x2 = [0.2, 0.4, 0.6, 0.8, 1.0]
x3 = [0.1, 0.3, 0.5, 0.7, 0.9]
y = [2.1, 4.2, 6.1, 8.5, 10.4]
```

```
# X ni tayyorlash
X = np.column_stack((x1, x2, x3))
```

```
# chiziqli regressiya modelini yaratish va bajarish
reg = np.linalg.lstsq(X, y, rcond=None)[0]
```

```
# natijani chiqarish
print(reg)
```

Bu kod ham yuqoridagi malumotlardan foydalanadi va chiziqli regressiya masalasini yechishning ikkinchi ko'rinishidir.

Bizning avvalgi misolimizda biz Numpy kutubxonasidagi “np.linalg.lstsq” funksiyasidan foydalandik, bu esa chiziqli regressiya uchun bir nechta boshqa metodlardan farqli hisoblanadi. Shu asnoda bu maqola uchun boshqa bir usulni ko'rib chiqamiz. Chiziqli regressiya uchun boshqa usullardan biri Gradient qisqartma usuli bo'lib, bu usulda regressiya ko'rsatkichlari yig'indisini eng kam qiluvchi holatda hisoblanadi. Gradient qisqartmasi usulini python yordamida quyidagi shaklda yozish mumkin:

```
import numpy as np

# ma'lumotlar
x1 = [0.1, 0.2, 0.3, 0.4, 0.5]
x2 = [0.2, 0.4, 0.6, 0.8, 1.0]
x3 = [0.1, 0.3, 0.5, 0.7, 0.9]
y = [2.1, 4.2, 6.1, 8.5, 10.4]
```

```
# X ni tayyorlash  
X = np.column_stack((x1, x2, x3))
```

```
# Gradient qisqartma usulini yaratish  
def gradient_descent(X, y, alpha=0.1, num_iters=100):  
    m, n = X.shape  
    theta = np.ones(n)  
    y = np.array(y)
```

```
    for i in range(num_iters):  
        h = np.dot(X, theta)  
        loss = h - y  
        gradient = np.dot(X.T, loss) / m  
        theta -= alpha * gradient  
    return theta
```

```
# gradient qisqartma usulini bajarish  
theta = gradient_descent(X, y)
```

```
# natijani chiqarish  
print(theta)
```

Bu kod yuqoridagi malumotlardan foydalanadi va Gradient qisqartma usuli yordamida chiziqli regressiya masalasini yechadi. Natija esa konsol oynasiga chiqadi.

FOYDALANILGAN ADABIYOTLAR:

- 1) Mathematical aspects of deep learning.(PHILIPP GROHS)
- 2) The World Almanac and Book of Facts 1993 (1993)
- 3) Actionable Recourse in Linear Classification(2019)