**DİCLE ÜNİVERSİTESİ**

**ERGANİ MESLEK YÜKSEKOKULU**

**RAPOR**

NUMPY CREATİNG ARRAYS

**HAZIRLAYAN**

SEDA NUR AYDIN

**DANIŞMAN**

FARUK AKSOY

29 MAYIS

DİYARBAKIR

**İÇİNDEKİLER**

1. **NUMPY NEDİR**
2. **NEDEN NUMPY KULLANILIR**

**2.1 NUMPY NEDEN LİSTELERDEN DAHA UZUN**

**3. NUMPY KURULUMU**

**3.1 NUMPYİ UYGULAMAYA DAHİL ETMEK**

**3.2 NUMPY NP OLARAK KULLANMAK**

**3.3 NUMPY SÜRÜMÜNÜ KONTROL ETMEK**

**4. NUMPY DİZİ OLUŞTURMA**

**4.1 NUMPY NDARRAY NESNESİ OLUŞTURMAK**

**4.2 DİZİLERDE BOYUTLAR**

**4.3 0-D DİZİLER**

**4.4 1-D DİZİLER**

**4.5 2-D DİZİLER**

**4.6 3-D DİZİLER**

**5. BOYUT SAYISINI KONTROL ETMEK**

**6. DAHA YÜKSEK BOYUTLU DİZİLER**

1. **NUMPY NEDİR**

Numpy, dizilerde çalışmak için kullanılan bir Python kütüphanesidir. Ayrıca doğrusal cebir, fourier dönüşümü ve matrisler alanında çalışmak için de gerekli işlevlere sahiptir. Numpy, 2005 yılında Travis Oliphant tarafından oluşturulmuştur. Açık kaynak kodlu bir projedir ve özgürce kullanabilirsiniz. Numpy, Numeriacal Python (Sayısal Python) ‘un kısaltmasıdır.

**2. NEDEN NUMPY KULLANILIR**

Python’da dizilerin amacına hizmet eden listelerimiz var ancak işlenmesi yavaştır. Numpy, geleneksel Python listelerinden 50 kata kadar daha hızlı bir dizi nesnesi sağlamayı amaçlamaktadır. Numpy’deki dizi nesnesi(array) ndarray olarak adlandırılır ve ndarray, çalışmayı çok kolaylaştıran birçok destekleyici işlev sağlar. Diziler, hız ve kaynakların çok önemli olduğu veri biliminde çok sık kullanılır.

**2.1 NUMPY NEDEN LİSTELERDEN DAHA HIZLI**

Numpy bir Python kütüphanesidir ve kısmen Python’da yazılmıştır ancak hızlı hesaplama gerektiren parçaların çoğu C veya C ++ ile yazılmıştır.

**3. NUMPY KURULUMU**

Bir sistemde Python ve PIP kuruluyorsa Numpy’nin kurulumu çok kolaydır. Bu komutu kullanarak kurulumu yapabilirsiniz:

Pip install numpy

Anacanda, Spyder zaten Numpy’nin kurulu olduğu bir Python dağıtımı kullanın.

**3.1 NUMPYİ UYGULAMAYA DAHİL ETMEK**

Numpy yüklendikten sonra import anahtar sözcüğünü ekleyerek onu uygulamalarınıza dahil edebilirsiniz.

Import numpy

Artık Numpy içe aktarılmış ve kullanıma hazırdır. Aşağıdaki kodla test edebilirsiniz.

Import numpy

ar = numpy . array([1, 2, 3, 4, 5])

print(arr)

**3.2 NUMPY NP OLARAK KULLANMAK**

Numpy genellikle np alias (takma adı) altında içe aktarılır. Alias Pyhon’da aynı şeye atıfta bulunmak için alternatif bir addır. İçe aktarırken as anahtar sözcüğüyle bir alias oluşturabilirsiniz.

import numpy as np

arr = np. array([1, 2, 3, 4, 5])

print(arr)

**3.3 NUMPY SÜRÜMÜNÜ KONTROL ETME**

Sürüm bilgisi version özniteliği altında saklanır.

print(np.\_version\_)

**4. NUMPY DİZİ OLUŞTURMA**

**4.1 NUMPY NDARRAY NESNESİ OLUŞTURMAK**

Numpy, dizilerle çalışmak için kullanılır. Numpy’deki dizi nesnesine ndarraydenir. Array() işlevini kullanarak bir Numpy nesnesi oluşturabiliriz.

Arr = np.array([1, 2, 3, 4, 5])

Print(arr)

Print(type(arr))

**4.2 DİZİLERDE BOYUTLAR**

Dizilerdeki bir boyut bir dizi derinliği düzeyidir (iç içe diziler).

**4.3 0-D DİZİLER**

0-D dizileri veya Skalarlar, bir dizideki öğelerdir. Bir dizideki her değer bir 0-D dizisidir.

arr = np.array(42)

print(arr)

**4.4 1-D DİZİLER**

Öğeleri olarak 0-D dizileri olan bir dizi tek boyutlu veya 1-D dizi olarak adlandırılır. Bunlar en yaygın ve temel dizilerdir.

arr = np.array([1, 2, 3, 4, 5])

print(arr)

**4.5 2-D DİZİLER**

Öğeleri olarak 1 boyutlu dizilere sahip bir diziye 2 boyutlu dizi denir. Bunlar genellikle matris veya 2.derece tensörleri temsil etmek için kullanılır.

arr = np.array([[1, 2, 3], [4, 5, 6]])

print(arr)

**4.6 3-D DİZİLER**

Elamanları olarak 2 boyutlu dizilere (matrisler) sahip olan bir dizi olarak adlandırılır. Bunlar genellikle 3. dereceden bir tensörü temsil etmek için kullanılır.

arr = np.array([[[1, 2, 3], [4, 5, 6]], [[1,2,3], [4,5,6]]])

print(arr)

**5. BOYUT SAYISINI KONTROL ETMEK**

Numpy dizileri, dizinin kaç boyuta sahip olduğunu bize söyleyen bir tamsayı döndüren ndim niteliğini sağlar. Dizilerin kaç boyutta olduğunu kontrol etmek için kullanabiliriz.

a = np.array(42)

b = np.array([1, 2, 3, 4, 5])

c = np.array([[1, 2, 3], [4, 5, 6]])

d = np.array([[[1, 2, 3], [4,5,6]], [[1, 2, 3], [4, 5, 6]]])

print(a.ndim) # 0

print(b.ndim) # 1

print(c.ndim) # 2

print(d.ndim) #3

**6. DAHA YÜKSEK BOYUTLU DİZİLER**

Bir dizi herhangi bir sayıda boyuta sahip olabilir. Dizi oluşturulduğunda ndim bağımsız değişkenini kullanarak boyutların sayısını tanımlayabiliriz. Örneğin 5 boyutlu bir dizi oluşturmak için aşağıdaki kullanıma bakabilirsiniz:

arr = np.array([1, 2, 3, 4], ndim=5)

print(arr) # [[[[[1 2 3 4]]]]]

print(‘Boyut sayısı: ‘ arr.ndim)

Bu dizide en içteki boyut (5.boyut) 4 öğeye, 4. boyut vektör olan 1 öğeye, 3. boyut vektörün bulunduğu matris olan 1 öğeye, 2. boyutun ise 3 boyutlu dizi olan 1 öğesi ve 1.dim, 4D dizisi olan 1 öğeye sahiptir.

**KAYNAKÇA :**

[**https://numpy.org/doc/stable/user/absolute\_beginners.html**](https://numpy.org/doc/stable/user/absolute_beginners.html)

[**https://www.classcentral.com/report/best-numpy-courses/**](https://www.classcentral.com/report/best-numpy-courses/)

[**https://analyticsindiamag.com/10-best-online-resources-to-learn-numpy/**](https://analyticsindiamag.com/10-best-online-resources-to-learn-numpy/)

[**https://realpython.com/numpy-tutorial/**](https://realpython.com/numpy-tutorial/)