



MATEMATİK BÖLÜMÜ

PYTHON'DA NUMPY PAKET İNCELEMESİ VE GRAFİK ÇİZİMİ

ASRA TAŞ 18025062

Danışman: DR. ÖĞR. ÜYESİ ELİF TEKİN TARIM

ÖZET

Yazımızın içeriği temel seviyede Python 3 programlama dili bilen kişiler esas alınarak yazılmıştır. Dört bölümden oluşan bu çalışmada "Giriş" bölümünde Python 3 programlama dilinde en sık kullanılan fonksiyonlar, döngüler, demetler, diziler anlatılmıştır. İkinci bölümünde ise Python'ın en sık kullanılan kütüphanesi olan NumPy Kütüphanesi anlatılmıştır. Üçüncü bölümde veri analizinde, sunumlarda veya projelerde veri görselleştirme için kullanılan Matplotlib Kütüphanesi anlatılmıştır. En son bölümümüzde ise NumPy ve Matplotlib kütüphanelerimizin ortak kullandığımız uygulama örneklerine yer verilmiştir.

NEDEN PYTHON?

Python 90'lı yılların başında Guido Van Rossum tarafından geliştirilmiş bir programlama dilidir. Python nesne yönelimli, yorumsal, modüler ve etkileşimli, yüksek seviyeli bir dildir. Yüksek seviye bir dil olduğu için öğrenilmesi kolaylaşmaktadır. Yapılmak istenen işler hızlı bir şekilde daha az kod satırıyla yapılabilir.

Python'un desteklendiği platformlar oldukça fazladır. Unix, Linux, Mac, Windows, Amiga, Symbian gibi birçok işletim sistemiyle uyumludur. Python, mühendislikten finansa kadar birçok alanda kullanılmaktadır.

-Python, sahip olduğu geniş kütüphane sayesinde oldukça popüler hale gelmiştir.

-Açık kaynaklıdır. Ücretsiz ve sade olması sayesinde diğer programlara göre çok basittir.

-Ayrı bir derleyiciye ihtiyaç duymaz. Python'da derleme işlemi ortadan kaldırıldığı için hızlı bir şekilde program geliştirilebilir.

-Daha düzenli bir kod dizilimine sahiptir. Ayrıca basit ve temiz syntaxa (söz dizimi) sahip olması birçok açıdan avantaj sağlamaktadır.

-Çok büyük yazılımların seri bir şekilde programlanması gerektiğinde C veya C++ gibi dillere göre daha kullanışlıdır.

-Python dilinin taşınabilirlik sağlaması herhangi bir ortamda yazılan bir programı, başka ortamlarda da çalıştırılabilir hale getirmektedir.

-Python'un standart kütüphaneleri birçok Internet protokolünü destekler.

NUMPY KÜTÜPHANESİ

NumPy (Numerical Python) bilimsel hesaplamaları hızlı bir şekilde yapmamızı sağlayan ve veri biliminde kullanılırken sentetik veri üretmeyi kolaylaştıran Python'da bulunan bir matematik kütüphanesidir. Bellekte sürekli yer kaplayan art arda sıralanmış aynı türden verilerin oluşturduğu kümeye dizi (array) denir. NumPy'in temelini NumPy dizileri oluşturur. NumPy dizileri Python listelerine benzer fakat hız ve işlevsellik açısından Python listelerinden daha kullanışlıdır. Ayrıca Python listelerinden farklı olarak NumPy dizileri homojen yapıda olmalıdır yani dizi içindeki tüm elemanlar aynı veri tipinden olmalıdır.

NumPy yükledikten sonra, import anahtar sözcüğünü ekleyerek onu uygulamalarınıza dâhil edebilirsiniz. NumPy genellikle np alias (takma adı) altında içe aktarılır. Alias Python'da, aynı şeye atfta bulunmak için alternatif bir addir. içe aktarılan as anahtar sözcüğüyle bir alias oluşturabilirsiniz. Listeyi NumPy dizisine dönüştürmek için np.array() metodu kullanılır. Ayrıca "ndarray" olarak da isimlendirilmiştir.

Dizilerde Boyutlar	Dizi Elemanlarına Erişim
Programlamada bir değişken sadece bir değer tutabilir, eğer bu değişken dizi değişken (array) ise birden fazla değer tutabilme özelliğine sahiptir. Örneğin 1-D dizi bir vektör, 2-D dizi bir matrix ve 3-D dizi bir küp gibidir. Bu durumda 4-D dizi küplerin vektörü olarak tanımlanabiliriz. Benzer şekilde 5-D dizi küplerin matrix'i, 6-D dizi küpleri küpü olarak düşünebiliriz.	Dizi indeksleme aslında bir dizi elemanına erişmek demektir. Bir dizi elemanına indeks numarasına bakarak erişebilirsiniz. NumPy dizilerindeki indeksler 0 ile başlar, yani ilk elemanın indeksi 0, ikincisinin indeksi 1 ve böyle devam eder.

Random Sayı Üretme
NumPy'da random sayı üretmek için bir random modülü bulunmaktadır. Bu modül aracılığıyla belirli aralıklarda, tam sayı (int), ondalıklı sayı (float) veya dizi (array) şeklinde random sayılar üretmek mümkündür.

İki NumPy dizisinin/matrisinin nokta çarpımını veya bir diğer adıyla skaler çarpımını bulmak için np.dot() metodu kullanılır.	İki NumPy dizisinin/matrisinin vektörel çarpımını bulmak için np.matmul() metodu kullanılır.
<pre>import numpy as np X = np.array([[1,8,3],[9,6,7]]) Y = X.min(axis=1) print(Y) [1 6]</pre>	Verilen eksendeki en küçük değerdeki elemanı bulmak için np.min() metodu kullanılır.

Bir diziyi kopyalamak istediğimizde np.copy() metodu kullanılır. Kopya dizi içerisinde orijinal diziden gelen veriler vardır ve kopya dizi üzerinde yapılan değişiklikler orijinal diziyi etkilemez. Aynı şekilde orijinal dizide yapılan değişiklikler kopya diziyi etkilemez.
<pre>import numpy as np A = np.array([2,5,0,3,9,1,3]) B = A.copy() A[0]=9 print("Matrix A:",A) print("Matrix B:",B) Matrix A: [9 5 0 3 9 1 3] Matrix B: [2 5 0 3 9 1 3]</pre>

- Bir dizinin görünümünü oluşturmak istediğimizde np.view() metodu kullanılır. Dizi görünümünde veriler yoktur ve dizi görünümünde yapılan değişiklikler orijinal diziyi etkiler. Benzer şekilde orijinal dizide yapılan değişiklikler görünümü etkiler.
- İki NumPy dizinin/matrisinin toplarken np.add() metodunu kullanırız.
- İki veya daha fazla dizideki ortak elemanları bulmak için np.intersect1d() metodu kullanılır.
- Eksendeki en büyük ve en küçük değerler arasındaki farkı bulmak istediğimizde np.ptp() metodunu kullanırız.
- Sıralı bir dizide bir elemanın indis numarasını bulmak için np.searchsorted() metodunu kullanırız.

İkinci dizide olup ikinci dizide olmayan elemanları np.setdiff1d() metodu ile buluruz.	Eksendeki elemanları sıralamak istediğimizde np.sort() metodunu kullanırız.
<pre>import numpy as np A = np.array([2,1,9,5,6,7,0]) B = np.array([0,1,2,6,8,9,3]) C = np.setdiff1d(A,B) print(C) [5 7]</pre>	Koşula uyan elemanın indisini verirken np.where() metodu kullanılır.

MATPLOTLIB İLE VERİLERİ GÖRSELLEŞTİRME

Python dili veri analizinde yaygın olarak kullanılmaktadır ve veri analizinde büyük çaplı verileri veya listeleri grafiklerle gösterirken Python dilinde "matplotlib" grafik çizimi paketi kullanılır. Matplotlib, Python'da düşük seviyeli bir çizim kitaplığıdır. Matplotlib paketine ait metotlar ile 1D (1boyutlu), 2D (2 boyutlu) ve 3D (3 boyutlu) çizimler yapılabilir.

Çizgi Grafiği	Pasta Dilimi Grafiği	Çubuk Dilimi Grafiği	Nokta (Dağılım) Grafiği	Histogram Grafiği	Yığılmış Alan Grafiği
Çizgi grafikleri çizdirmek için 'plot' komutu kullanılır. Varsayılan olarak çizilen grafikler çizgi grafikleridir. Fakat farklı türde çizgi ve işaretlere sahip grafikler çizdirmek için 'plot' komutuna çizgi ve renk kodları eklenebilir.	Pasta Grafik, yalnızca bir dizi veriyi görüntüleyebilen dairesel bir istatistiksel çizimdir. Grafiğin alanı, verilen verilerin toplam yüzdesidir. Pastanın dilimlerinin alanı, veri bölümlerinin yüzdesini temsil eder.	Bir çubuk grafik, farklı kategoriler arasındaki karşılaştırmaları gösterir. Grafiğin bir eksenini, karşılaştırılan belirli kategorileri gösterir ve diğer eksen ölçülen bir değeri temsil eder. Çubuk grafiklerini çizdirmek için 'bar' komutu kullanılır.	Nokta (dağılım) grafiği çizimlerinde plot() metodu yerine scatter() metodu kullanılır. Bu iki metot birlikte de kullanılabilir.	Python'da histogram eğrilerinin elde edilmesinde matplotlib.pyplot paketi içerisinde yer alan "hist" metodu kullanılır.	Aynı grafik üzerinde birkaç grubun değerinin genişlemesini görüntülemek için temel alan grafiğinin büyütülmesidir. Her grubun değerleri üst üste görüntülenir.

UYGULAMA

Örnek 1. Euler Metodunu kullanarak, $xy' - y = x^2 \sin(x)$, $y(-\frac{\pi}{2}) = 0$, $[-\frac{\pi}{2}, 10]$ aralığında $n = 27$ adımda yaklaşımını bul.

Python Kodu	Çıktı
<pre>import numpy as np from matplotlib import pyplot as plt x0=-np.divide(np.pi,2) y0=0 xp=10 n=27 step_h=(xp-x0)/(n-1) x=np.linspace(x0,xp,n) y=np.zeros(n) y[0]=y0 for k in range(1,n): y[k]=y[k-1]+step_h*(np.divide(y[k-1],x[k-1])+(x[k-1])*(np.sin(x[k-1]))) plt.plot(x,y,'o') plt.xlabel("x-value") plt.ylabel("y-value") plt.title("Approximate Solution with Euler's Method") plt.show()</pre>	<pre>-1.5707963267948966 0.0 -1.1257656988412466 0.6990524757008195 -0.6807350708875968 0.8749089728046469 -0.235704442933947 0.4936018011996627 0.20932618501970301 -0.4138655391491306 0.654356812973353 -1.2743919081283885 1.0993874409270026 -1.9638668589356463 1.5444180688806252 -2.322940594856259 1.9894486868343026 -2.3052315599179236 2.4344793247879526 -2.011998055277782 2.8795099527416026 -1.6759640952187207 3.3245405806952526 -1.6029661760548366 3.7695712086489017 -2.0867110670985656 4.214601836602552 -3.3186576030407586 4.659632464556202 -5.317086461475434 5.104663092509852 -7.895703034412756 5.549693720463502 -10.68323528269886 5.994724348417152 -13.193364919467882 6.439754976370802 -15.1516743781324808 6.884785604324451 -15.516743781324808 7.32981632278102 -14.785666200310141 7.774846860231751 -12.85933566471649 8.219877488185402 -10.14618489244194 8.66490811613905 -7.279562687710384 9.1099387440927 -4.99721553528096 9.55496937204635 -3.98590268595458 10.0 -4.72359376657091</pre>

KAYNAKÇA

- [1] Çobanoğlu, B., "Herkes için Python", 1. Baskı, Pusula Basımevi, İstanbul, 2018
- [2] Karaçay, T., "Python 3", 1. Baskı, Kubaan Matbaacılık, Ankara, 2009
- [3] <https://numpy.org>