**PANDAS (Panel Data System)**

Numpy eksikliklerini gidermek icin gelistirilmis bir kutuphane

Gorsellestirme yapmamiz, raporlar cikarmamiz, veriyi temizlememiz gerekebiliyor.

Veriyi modelleme icin, makine ogrenmesi ya da deepL icin hazirlamamiz istenebilir

Ya da veriyi anlamlandirip powerBI veya tableau’ya sokmamiz gerekebilir

Ikisi icin de pandas kullaniyoruz.

**Numpy ve Pandas farklari:**

-Numpy homojen tutuyordu, pandas’ta oyle degil.

-Pandasta, sutun ve satir indekslerini istedigimiz gibi isimlendirebiliyoruz.

-Numpy’in temeli array’ler iken, Pandas’ta ise, seriler ve data frame var. Data frame daha cok

-Daha yuksek performansli ve daha esnek

-Pandas daha cok ekonomi ile ilgili kodlanmaya baslanmis once

Panel data tipi var bir de:

**Pip install pandas**

**Import pandas as pd**

**Series**: 1D- tek boyutlu array’lere benziyor. Homojen data aliyor.

**Data frame:** 2D- heterojen data alabiliyor. 2 boyutlu

**Panel:** 3D arka arkaya paneller seklinde (3D array gibi)

**Series:**

Belirli index degerlerine sahip, belirli homojen data

**Icine liste vererek series olusturabiliriz.**

**Pandas.Series(data, index,dtype, copy)**

**Pd.series ([85,90,70,80]), name= ‘marks’**

**85,90.. da value olarak gecer.**

S=pd.Series(np.random.randn(5), index=[“a”, “b”, “c”, “d”, “e”] diyebiliriz mesela

Default 0’dan baslar ama istedigimiz degeri verebiliriz index olarak

Series’e de isim verebiliriz.

Numpy’da array olusturdugumuzda, index sayilarini getirmez.

Pandas’ta 0,1,2,3 gibi index’lerin numaralarini da gostererek value’lari gosterir.

|  |  |
| --- | --- |
| 0 | x |
| 1 | y |
| 2 | z |
| 3 | a |
| 4 | b |

**Data Frame**

Series’lerin birlesmesinden data frame olusuyor denebilir.

Index’ler de gosterilir basinda. 2D

Bir sutununda int bir sutununda str data tipinde heterojen data alabilir

|  |  |  |
| --- | --- | --- |
| 0 | x | 12 |
| 1 | y | 5 |
| 2 | z | 7 |
| 3 | a | 8 |
| 4 | b | 1 |

Iki dict formatinda yazdigimiz seriyi toplayarak, data frame olusturulabilir.

**Df=pd.DataFrame(d)**

Eger birinde 3 index digerinde 4 index olan 2 seriyi birlestirmek istersem bunu pandasta yapabiliyorum.

Arraylerin boyutlari birbirine uymadiginda, concatenate yaparken sorun cikiyordu np’de

Ama pandasta bunu yapabiliyoruz. NaN dedigimiz bilinmeyen bir degerle dolduruyor.

Index’ler default olarak 0,1,2,3 ama a,b,c,d olarak da isimlendirebiliyorsun

Ya da index’e 100, 101, 102 seklinde atama yapabilirsin

\*Python’da degisken isimlerine foo bar baz isimleri verilebilir. calisma notlari olusturmak icin bir isimlendirme olustururlar

**Scalar**: agirlik gibi sicaklik gibi degerler

**3 yontem:**

1. Dict item’lardan series olusturmak

**data={‘a’:0., ‘b’:1., ‘c’:2.}**

**S=pd.Series(data)**

Burda indexler a,b,c olur, value’lar 0.0, 1.0, 2.0

2.

**data={‘a’:0., ‘b’:1., ‘c’:2.}**

**S=pd.Series(data, index=[‘b’,’c’,’d’,’a’])**

Bu da d indexine NaN atayarak yapar.

3.

**S=pd.Series(5, index=[0,1,2,3]**

bu mesela tum indekslere 5 atayan bir seri

**Indexing and slicing**

s[a] yazinca a’ya karsilik hangi deger varsa o gelir

S[[‘a’, ‘c’, ‘d’] a,c,d. indexleri getirir.

Birden fazla degeri getirirken cift parantezle kullaniyorum

Ser[[0,2,4]] seklinde cagirdigimda, o indeksleri getirir.

Ser[[‘1st’, ‘3rd’ ‘5th’]] seklinde de yazabilirim.

Ser[ser>3] yazdigimda True False seklinde dondurur.

Ser[2:]=100

Dedigimde, 2. indeksten sonrasini 100 yap demek

Ser+ser gibi bunlari da toplayabiliyorum

Ser/2 gibi bolebiliyorum. Ortaya hep float deger cikar bolmede

Array’lerde kullandigimiz methodlari bunda da kullanabiliyoruz

**Ser.mean()**

**Np.mean(ser)** diyerek de ortalama alabiliyoruz

Index label’lar ayni isimde de olabilir

Oyle atarsan oyle olur.2 tane a, 2 tane b gibi

A’dan 2 tane atadin index olarak diyelim

1. Indexi cagirirsan 2 value’yu da verir

Ilkini cagirmak icin yanina[0] dersin, ikinciyi cagirmak icin [1] dersin

Iki seriyi birbiri ile carparken, degerlerden biri Nan ise carpim da NaN olacak

**Positional index** (sr[2])

**Label index** [‘India’]

**Negative index:** [-1] dedim mesela

**Slicing:** [1:3] 1’den basla 3’e kadar git demek,

**Row index**

**Column index**

Bir kosul ile de indeksleme islemi yapabilirz

Sutun>6 diyince sarti saglayanlari getiriyor.

**Data frame append:**

Append methodu ile data frame’leri birbirine ekleyebilirim.

Sutun indexleri farkli olsaydi, NaN olarak doldurararak ekler

**reindex**

**Fill\_value=black** dedim mesela NaN degeleri black ile doldur

**Method=’ffill’** bir oncekiyle dolduruyor

**Describe -omrun boyunca unutma!**

Veri hakkinda genel istatistiksel veri