Veri bilimi kodlar

# VERİ BİLİMİNE GİRİŞ

## Veri Bilimi Nedir?

### İstatistik

### Makine Öğrenmesi

### Veri İşleme

### Bilgisayar bilimleri

### Alan uzmanlığı

## Veri biliminin gelişimi nasıl oldu?

## Veri bilimcisi kimdir

### Programlama ve veri tabanı

### Python

### R

### SQL

### JAVA

### MATLAB

### C++

### HADOOP VE HIVE

### SAS

### TABLEAU

### BigQuery

## Dünyada Veri Bilimci

# PYTHON’A GİRİŞ

## Python programlamaya giriş

## Veri türleri

### string: Karakter dizileri

print("Merhaba veri bilimi")

## Merhaba veri bilimi

print("Merhaba\  
 veri\  
 bilimi")

## Merhaba veri bilimi

### integer: Tam sayılar

print(55)

## 55

print(123456789+1)

## 123456790

55

## 55

### float: Reel Sayılar

5.5

## 5.5

-55.03

## -55.03

5.03e55

## 5.03e+55

45e-111

## 4.5e-110

type(55.03)

## <class 'float'>

### complex: Karmaşık sayılar

55 + 3j

## (55+3j)

type(55 + 3j)

## <class 'complex'>

### bool: True, False

type(True)

## <class 'bool'>

type(False)

## <class 'bool'>

## Operatörler

### Aritmetik Operatörler

7+5

## 12

117/34

## 3.4411764705882355

3\*\*4

## 81

45//4

## 11

45%4

## 1

### Fonksiyonlar

print("Merhaba Veri Bilimi Dersi")

## Merhaba Veri Bilimi Dersi

llllllllllllllllllllllllllllllllllllllll

x = -2019  
abs(x)

## 2019

d = -3.54632  
int(d)

## -3

notlar = [21,34,57,75,81,83,87,90,93,99,31,37,41]  
len(notlar)

## 13

llllllllllllllllllllllllllllllllllllllll

def fonk1(x):  
 return x\*x+5  
fonk1(3)

## 14

def fonk2(x,y,z):  
 return x\*\*2 + y\*\*3 + z\*\*-2  
fonk2(3, 55, 12)

## 166384.00694444444

for i in range(10):  
 print(i)

## 0  
## 1  
## 2  
## 3  
## 4  
## 5  
## 6  
## 7  
## 8  
## 9

def ortalama(x):  
 "x değişkenlerinin ortalamasını hesaplama"  
 top\_x = sum(x)  
 sayı\_x = len(x)  
 return top\_x/sayı\_x  
  
ortalama([11,22,33,44,55])

## 33.0

llllllllllllllllllllllllllllllllllllllll

def harmonik\_ortalama(s):  
 "s tane sayının harmonik ortalamasını hesaplama"  
 n = len(s)  
 sayılar = [1/s[i] for i in range(0,n)]  
 harmonik\_ortalama = n/sum(sayılar)  
 return harmonik\_ortalama  
  
harmonik\_ortalama([10,20,30,40,50,60])

## 24.489795918367346

### Karşılaştırma

23 == 77

## False

34 <= 55

## True

4 != 5

## True

### CSV formatındaki Veriyi yükleme

import pandas as pd  
import numpy as np  
  
data = pd.read\_csv("C:/Users/PC/Desktop/Veri Bilimi Kitap/fuel.csv")  
data.head()

## Model Year Division ... Hwy FE Comb FE  
## 0 2016 ALFA ROMEO ... 34 28  
## 1 2016 Aston Martin Lagonda Ltd ... 18 14  
## 2 2016 Aston Martin Lagonda Ltd ... 21 16  
## 3 2016 Aston Martin Lagonda Ltd ... 19 15  
## 4 2016 Aston Martin Lagonda Ltd ... 21 16  
##   
## [5 rows x 8 columns]

data = pd.read\_csv(r"C:\Users\PC\Desktop\Veri Bilimi Kitap\fuel.csv")  
data.head()

## Model Year Division ... Hwy FE Comb FE  
## 0 2016 ALFA ROMEO ... 34 28  
## 1 2016 Aston Martin Lagonda Ltd ... 18 14  
## 2 2016 Aston Martin Lagonda Ltd ... 21 16  
## 3 2016 Aston Martin Lagonda Ltd ... 19 15  
## 4 2016 Aston Martin Lagonda Ltd ... 21 16  
##   
## [5 rows x 8 columns]

llllllllllllllllllllllllllllllllllllllll

print(data.shape)

## (1205, 8)

print(data.columns)

## Index(['Model Year', 'Division', 'Carline', 'Eng Displ', '# Cyl', 'City FE',  
## 'Hwy FE', 'Comb FE'],  
## dtype='object')

data.head(6)

## Model Year Division ... Hwy FE Comb FE  
## 0 2016 ALFA ROMEO ... 34 28  
## 1 2016 Aston Martin Lagonda Ltd ... 18 14  
## 2 2016 Aston Martin Lagonda Ltd ... 21 16  
## 3 2016 Aston Martin Lagonda Ltd ... 19 15  
## 4 2016 Aston Martin Lagonda Ltd ... 21 16  
## 5 2016 Aston Martin Lagonda Ltd ... 19 15  
##   
## [6 rows x 8 columns]

data.tail(3)

## Model Year Division Carline ... City FE Hwy FE Comb FE  
## 1202 2016 Volkswagen Touareg ... 17 23 19  
## 1203 2016 Volvo XC90 AWD ... 20 25 22  
## 1204 2016 Volvo XC90 AWD ... 22 25 23  
##   
## [3 rows x 8 columns]

print(data.describe())

## Model Year Eng Displ ... Hwy FE Comb FE  
## count 1205.0 1205.000000 ... 1205.000000 1205.000000  
## mean 2016.0 3.150041 ... 28.215768 23.295436  
## std 0.0 1.336714 ... 5.987383 5.567975  
## min 2016.0 0.900000 ... 13.000000 12.000000  
## 25% 2016.0 2.000000 ... 24.000000 19.000000  
## 50% 2016.0 3.000000 ... 28.000000 23.000000  
## 75% 2016.0 3.700000 ... 32.000000 27.000000  
## max 2016.0 8.400000 ... 53.000000 56.000000  
##   
## [8 rows x 6 columns]

data.describe(include = ['O'])

## Division Carline  
## count 1205 1205  
## unique 43 768  
## top BMW Jetta  
## freq 107 6

# PYTHON’DA VERİ DEPOLAMA

## PYTHON liste

import numpy as np  
veri1 = np.arange(6) # 0'dan 5'e kadar  
print(veri1) # veriyi yazdırma

## [0 1 2 3 4 5]

print(veri1.shape) # verinin büyüklüğünü gösterme

## (6,)

veri2 = np.arange(10,22,3) # 10'dan 22'ye kadar 3'er artırımlı veri seti  
print(veri2)

## [10 13 16 19]

veri3 = np.zeros(6) # tüm değerleri sıfır olan dizi  
print(veri3)

## [0. 0. 0. 0. 0. 0.]

print(veri3.shape)

## (6,)

veri4 = np.zeros((3,4)) # tüm değerleri sıfır olan 3x4 matris  
print(veri4)

## [[0. 0. 0. 0.]  
## [0. 0. 0. 0.]  
## [0. 0. 0. 0.]]

print(veri4.shape)

## (3, 4)

veri5 = np.full((3,4),5) # tüm deperleri beş olan 3x4 matris  
print(veri5)

## [[5 5 5 5]  
## [5 5 5 5]  
## [5 5 5 5]]

print(veri5.shape)

## (3, 4)

veri6 = np.eye(5) # 5x5 birim matris  
print(veri6)

## [[1. 0. 0. 0. 0.]  
## [0. 1. 0. 0. 0.]  
## [0. 0. 1. 0. 0.]  
## [0. 0. 0. 1. 0.]  
## [0. 0. 0. 0. 1.]]

print(veri6.shape)

## (5, 5)

veri7 = np.random.random((3,4)) # [0,1) yarı açık aralığından  
print(veri7) # 3x4 boyutunda veri üretme

## [[0.91027051 0.62150431 0.92218913 0.14837906]  
## [0.49122571 0.92960565 0.50891534 0.04387901]  
## [0.36687253 0.21804907 0.59610152 0.51454252]]

print(veri7.shape)

## (3, 4)

liste1 = [12,22,14,15,16] # Bir Python listesinden bir NumPy dizisi oluşturma  
a1 = np.array(liste1)  
print(a1)

## [12 22 14 15 16]

print(a1[0]) # dizinin 1. elemanını yazdır

## 12

print(a1[1])

## 22

print(a1[4]) # dizinin 5. elemanını yazdır

## 16

liste1 = [12,22,14,15,16]  
liste2 = [102,212,141,315,716]  
a2 = np.array([liste1, liste2]) # iki diziyi sıralama  
print(a2)

## [[ 12 22 14 15 16]  
## [102 212 141 315 716]]

sayılar = np.arange(15)  
print(sayılar)

## [ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14]

çift\_sayılar = sayılar[sayılar % 2 == 0] # 2'ye bölümünden kalan 0 olan sayılar  
print(çift\_sayılar)

## [ 0 2 4 6 8 10 12 14]

a3 = np.array([[11,21,33,43,75],  
 [4,5,61,17,8],  
 [19,8,77,6,51]])  
print(a3)

## [[11 21 33 43 75]  
## [ 4 5 61 17 8]  
## [19 8 77 6 51]]

print(" ")

b3 = a3[0:3,:3] # ilk üç satırı ve ilk üç sütunu al  
print(b3)

## [[11 21 33]  
## [ 4 5 61]  
## [19 8 77]]

x1 = np.array([2,3,5])  
x2 = np.array([7,4,2])  
np.dot(x1, x2) # iç (skaler) çarpım: 2x7 + 3x4 + 5x2

## 36

m1 = np.array([[11,21,3], [21,4,5]])  
m2 = np.array([[7,8], [9,10], [11,12]])  
print(m1)

## [[11 21 3]  
## [21 4 5]]

print(m2)

## [[ 7 8]  
## [ 9 10]  
## [11 12]]

print(np.dot(m1,m2)) # matris çarpımı

## [[299 334]  
## [238 268]]

m3 = np.array([(11,22,33,44), (4,5,6,7), (27,28,29,30)])  
print(m3)

## [[11 22 33 44]  
## [ 4 5 6 7]  
## [27 28 29 30]]

print(m3.cumsum()) # Kümülatif toplamları bir dizi halinde yazma

## [ 11 33 66 110 114 119 125 132 159 187 216 246]

yaşlar = np.array([44,22,37,56,13,77,61])  
sıralanmış\_yaşlar = np.sort(yaşlar) # sort:sıralama komutu  
print(yaşlar) # orjinal veriyi yazdırma

## [44 22 37 56 13 77 61]

kardeşler = np.array(["Recep", "Fatma", "Şaban", "Emine", "Selvi"])  
yaş = np.array([60,57,55,51,49])  
boy = np.array([188,165,190,169,170])  
diz = np.argsort(yaş) # "yaş" ve "boy" değişkenlerine göre "kardeşler" verisini sıralama  
diz1 = np.argsort(boy)  
print(kardeşler[diz])

## ['Selvi' 'Emine' 'Şaban' 'Fatma' 'Recep']

print(kardeşler[diz1])

## ['Fatma' 'Emine' 'Selvi' 'Recep' 'Şaban']

ters\_diz = np.argsort(kardeşler)[::-1] # Büyüklüğe ya da alfabetik olarak tersine sıralama  
print(yaş[ters\_diz])

## [55 49 60 57 51]

print(boy[ters\_diz])  
  
# bir referansa göre kopyalama:

## [190 170 188 165 169]

liste1 = [[15, 25, 34, 47], [54, 64, 75, 85], [154, 164, 715, 825]]  
d1 = np.array(liste1)  
print(d1)

## [[ 15 25 34 47]  
## [ 54 64 75 85]  
## [154 164 715 825]]

d2 = d1  
print(d2)

## [[ 15 25 34 47]  
## [ 54 64 75 85]  
## [154 164 715 825]]

d2[0][0] = 999 # d2 matrisinin 1. satır 1. sütun elemanını 999 olarak değiştirme  
d2[1][3] = -888 # d2 matrisinin 2. satır 4. sütun elemanını -888 olarak değiştirme  
  
d2.shape = 2,-1 # 3x3 boyutundaki d2 matrisi 2 satıra indirgeme  
print(d2)

## [[ 999 25 34 47 54 64]  
## [ 75 -888 154 164 715 825]]

## pandas serileri

import pandas as pd

llllllllllllllllllllllllllllllllllllllll

df = pd.DataFrame({  
 "İsim": ["Melih", "Kübra", "Büşra", "Kürşat", "Gülnur", "Feyzi", "Esra", "Zişan", "Mustafa", "Zeynep", "Harnunnisa"],  
 "Yaş": [37,36,33,29,27,25,25,20,19,16,14],  
 "Cinsiyet": ["erkek", "kadın", "kadın", "erkek", "kadın", "erkek", "kadın", "kadın", "erkek", "kadın", "kadın"]  
})  
df

## İsim Yaş Cinsiyet  
## 0 Melih 37 erkek  
## 1 Kübra 36 kadın  
## 2 Büşra 33 kadın  
## 3 Kürşat 29 erkek  
## 4 Gülnur 27 kadın  
## 5 Feyzi 25 erkek  
## 6 Esra 25 kadın  
## 7 Zişan 20 kadın  
## 8 Mustafa 19 erkek  
## 9 Zeynep 16 kadın  
## 10 Harnunnisa 14 kadın

llllllllllllllllllllllllllllllllllllllll

df["Cinsiyet"]

## 0 erkek  
## 1 kadın  
## 2 kadın  
## 3 erkek  
## 4 kadın  
## 5 erkek  
## 6 kadın  
## 7 kadın  
## 8 erkek  
## 9 kadın  
## 10 kadın  
## Name: Cinsiyet, dtype: object

llllllllllllllllllllllllllllllllllllllll

df["Yaş"].max()

## 37

Yaş = pd.Series([37,36,33,29,25,25,20,19,16,14], name = "Yaş")  
Yaş

## 0 37  
## 1 36  
## 2 33  
## 3 29  
## 4 25  
## 5 25  
## 6 20  
## 7 19  
## 8 16  
## 9 14  
## Name: Yaş, dtype: int64

Yaş.max()

## 37

llllllllllllllllllllllllllllllllllllllll

df.describe()

## Yaş  
## count 11.000000  
## mean 25.545455  
## std 7.802098  
## min 14.000000  
## 25% 19.500000  
## 50% 25.000000  
## 75% 31.000000  
## max 37.000000

veri = {"isim": ["Hüsnü", "Nevzat", "Dr.Erol", "Talat", "Yüksel", "Adem", "İzzet"],  
"yıl": [1969, 1965, 1967, 1972, 1971, 1972, 1975],  
"doğum\_yeri": ["Çarşamba", "Havza", "Araklı", "Karabük", "Sürmene", "Of", "Hatay"],  
"çocuk\_sayısı": [2,4,1,2,3,3,5]}  
veri = pd.DataFrame(veri, index = [1,2,3,4,5,6,7])  
print(veri)

## isim yıl doğum\_yeri çocuk\_sayısı  
## 1 Hüsnü 1969 Çarşamba 2  
## 2 Nevzat 1965 Havza 4  
## 3 Dr.Erol 1967 Araklı 1  
## 4 Talat 1972 Karabük 2  
## 5 Yüksel 1971 Sürmene 3  
## 6 Adem 1972 Of 3  
## 7 İzzet 1975 Hatay 5

Satırları kaldırma:

print(veri.drop([4,6])) # indeks olarak kodlanan 4. ve 6. satırları kaldırma

## isim yıl doğum\_yeri çocuk\_sayısı  
## 1 Hüsnü 1969 Çarşamba 2  
## 2 Nevzat 1965 Havza 4  
## 3 Dr.Erol 1967 Araklı 1  
## 5 Yüksel 1971 Sürmene 3  
## 7 İzzet 1975 Hatay 5

veya

print(veri.drop([4,6])) # 4 ve 6. satırları silme

## isim yıl doğum\_yeri çocuk\_sayısı  
## 1 Hüsnü 1969 Çarşamba 2  
## 2 Nevzat 1965 Havza 4  
## 3 Dr.Erol 1967 Araklı 1  
## 5 Yüksel 1971 Sürmene 3  
## 7 İzzet 1975 Hatay 5

print(veri[veri.isim != "Dr.Erol"]) # Dr.Erol'a ait satırı kaldırma

## isim yıl doğum\_yeri çocuk\_sayısı  
## 1 Hüsnü 1969 Çarşamba 2  
## 2 Nevzat 1965 Havza 4  
## 4 Talat 1972 Karabük 2  
## 5 Yüksel 1971 Sürmene 3  
## 6 Adem 1972 Of 3  
## 7 İzzet 1975 Hatay 5

Sütunları kaldırma:

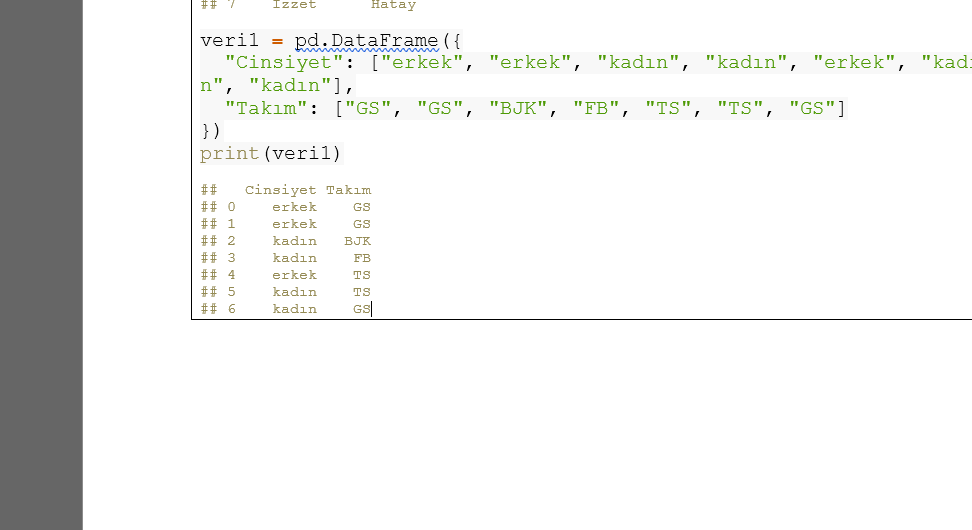


Figure .

Table .

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

print(veri.drop("doğum\_yeri", axis = 1)) # Dopum yeri sütununu kaldırma

## isim yıl çocuk\_sayısı  
## 1 Hüsnü 1969 2  
## 2 Nevzat 1965 4  
## 3 Dr.Erol 1967 1  
## 4 Talat 1972 2  
## 5 Yüksel 1971 3  
## 6 Adem 1972 3  
## 7 İzzet 1975 5

print(veri.drop(veri.columns[[1,3]], axis = 1)) # 2. ve 4. sütunu silme

## isim doğum\_yeri  
## 1 Hüsnü Çarşamba  
## 2 Nevzat Havza  
## 3 Dr.Erol Araklı  
## 4 Talat Karabük  
## 5 Yüksel Sürmene  
## 6 Adem Of  
## 7 İzzet Hatay

veri1 = pd.DataFrame({  
 "Cinsiyet": ["erkek", "erkek", "kadın", "kadın", "erkek", "kadın", "kadın"],  
 "Takım": ["GS", "GS", "BJK", "FB", "TS", "TS", "GS"]  
})  
print(veri1)

## Cinsiyet Takım  
## 0 erkek GS  
## 1 erkek GS  
## 2 kadın BJK  
## 3 kadın FB  
## 4 erkek TS  
## 5 kadın TS  
## 6 kadın GS