

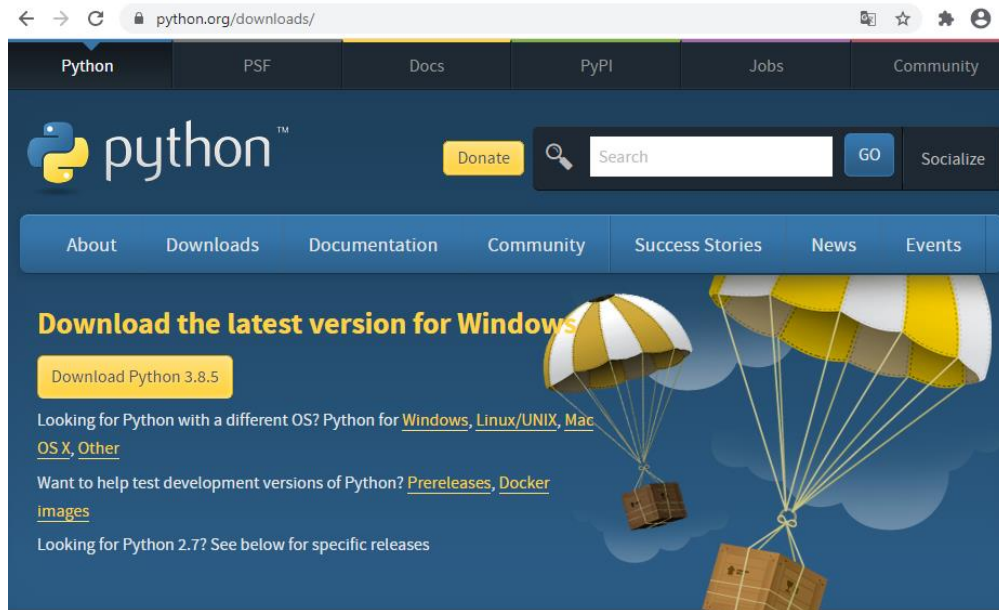
Makine öğrenmesi



PROGRAMLAMAYA GİRİŞ

Python ücretsiz bir programlama dilidir. Guido Van Rossum adlı Google’da çalışmış Hollandalı bir programcı tarafından 90’lı yılların başında geliştirilmeye başlanmıştır. Adı “The Monty Python” adlı bir İngiliz komedi grubundan esinlenilmiştir. Python, öğrenmesi kolay, tamamen açık kaynak ücretsiz bir programlama dilidir. Python diğer programlama dillerine göre daha az kod ile işlem yapması avantajına sahiptir ve bütün işletim sistemleri ile uyum içerisinde çalışmaktadır. İçerisinde birçok kütüphaneyi barındırmaktadır. Bu kaynaklarla daha az kod yazmak mümkündür. Python ile masaüstünde çalışan uygulamalar geliştirilebileceği gibi web üzerinde çalışan uygulamalar geliştirmek mümkündür.

En son Python programı sürümü olan Python 3.8.5 <http://www.python.org/downloads> adresinden yüklenebilir.

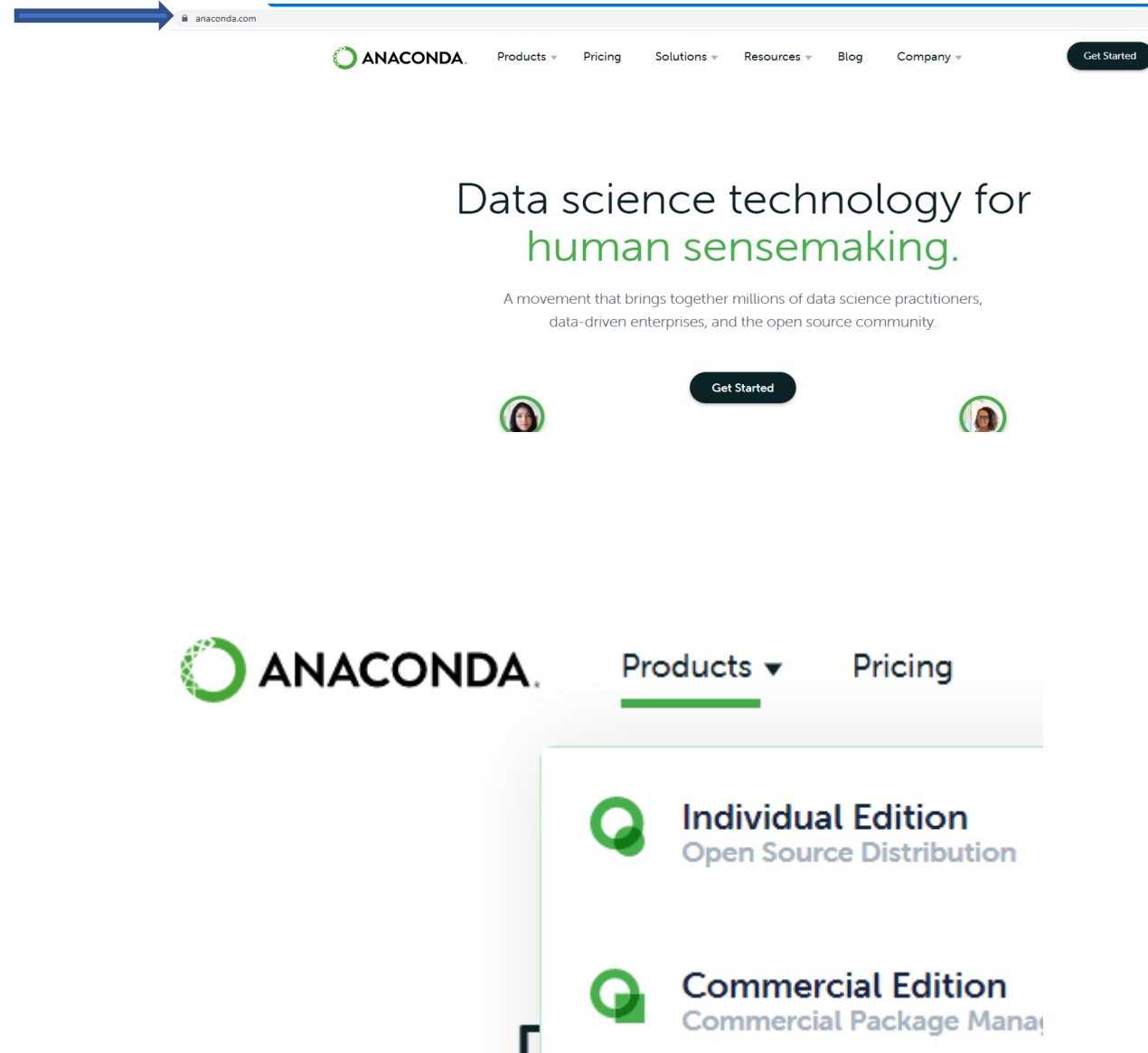


Anaconda, veri bilimi ve benzeri bilimsel uygulamalar için **python** kullanmak isteyenlere hazırlanmış tümleşik bir **python** dağıtımıdır. **Anaconda dağıtımı**, 1.500'den fazla paketin yanı sıra **conda** paketi ve sanal çevre yöneticisiyle birlikte gelir. Ayrıca komut satırı arabirimi (CLI) grafiksel bir alternatif olarak bir GUI, **Anaconda Navigator** içerir.

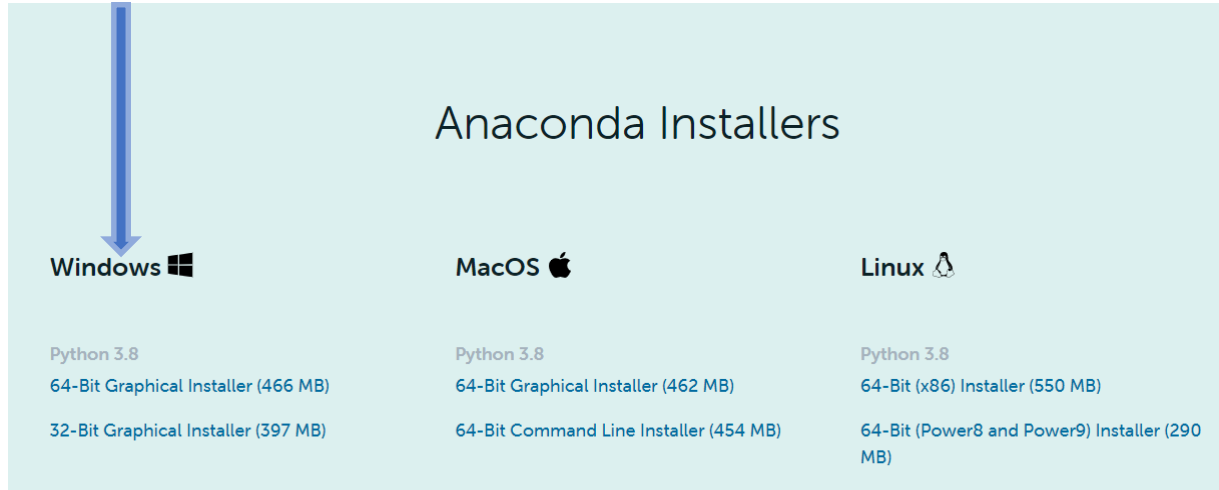
Conda ve pip paket yöneticisi arasındaki en büyük fark, paket gereksinimlerinin yönetim farkıdır, bu da Python veri bilimi ve condanın varlığının nedeni için önemli bir zorluktur. Pip bir paket yüklediğinde, önceden yüklenmiş paketlerle çakışıp çakışmadığını kontrol etmeden, bağımlı Python paketlerini otomatik olarak kurar. Mevcut kurulumun durumuna bakılmaksızın bir paket ve

Makine öğrenmesi

bağımlılıklarından herhangi birini kuracaktır. Bu nedenle, örneğin Google Tensorflow'un çalışan bir kurulumuna sahip bir kullanıcı, gerekli olan numpy kütüphanesinin Tensorflow tarafından kullanılandan farklı bir sürümünü gerektiren farklı bir paket yüklemek için pip kullanmayı bıraktığını görebilir. Bazı durumlarda, paket çalışıyor gibi görünebilir ancak ayrıntılı olarak farklı sonuçlar verebilir.



Makine öğrenmesi



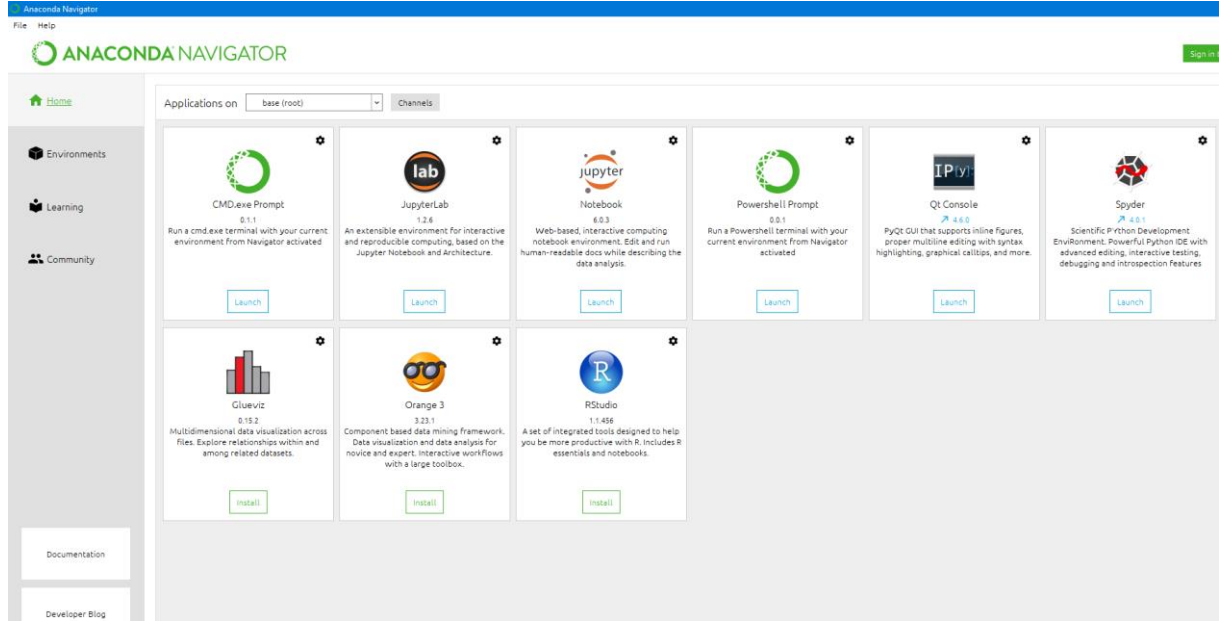
Program yüklemesinden sonra aşağıda gösterilen Anaconda giriş paneli oluşturulur. Veri bilimi, yapay zeka vb konularında sıkça kullanılan kütüphanelerin yanı sıra **jupyter notebook** ve **spyder** gibi araçları da barındırır. Programı anaconda.com adresinden kendi işletim sisteminize uygun versiyonu indirerek kurabilirsiniz. Anaconda'yı kurduğunuzda sisteminizde **python**, **jupyter notebook** ve **spyder** da kurulmuş olacak.

Anaconda Navigator, Anaconda dağıtımında bulunan ve kullanıcıların komut satırı komutlarını kullanmadan uygulamaları başlatmasını ve conda paketlerini, ortamları ve kanalları yönetmesini sağlayan bir masaüstü grafik kullanıcı arabirimidir (GUI) . Navigator, paketleri Anaconda Cloud'da veya yerel bir Anaconda Deposunda arayabilir, bir ortama kurabilir, paketleri çalıştırabilir ve güncelleyebilir. Windows, macOS ve Linux için kullanılabilir .

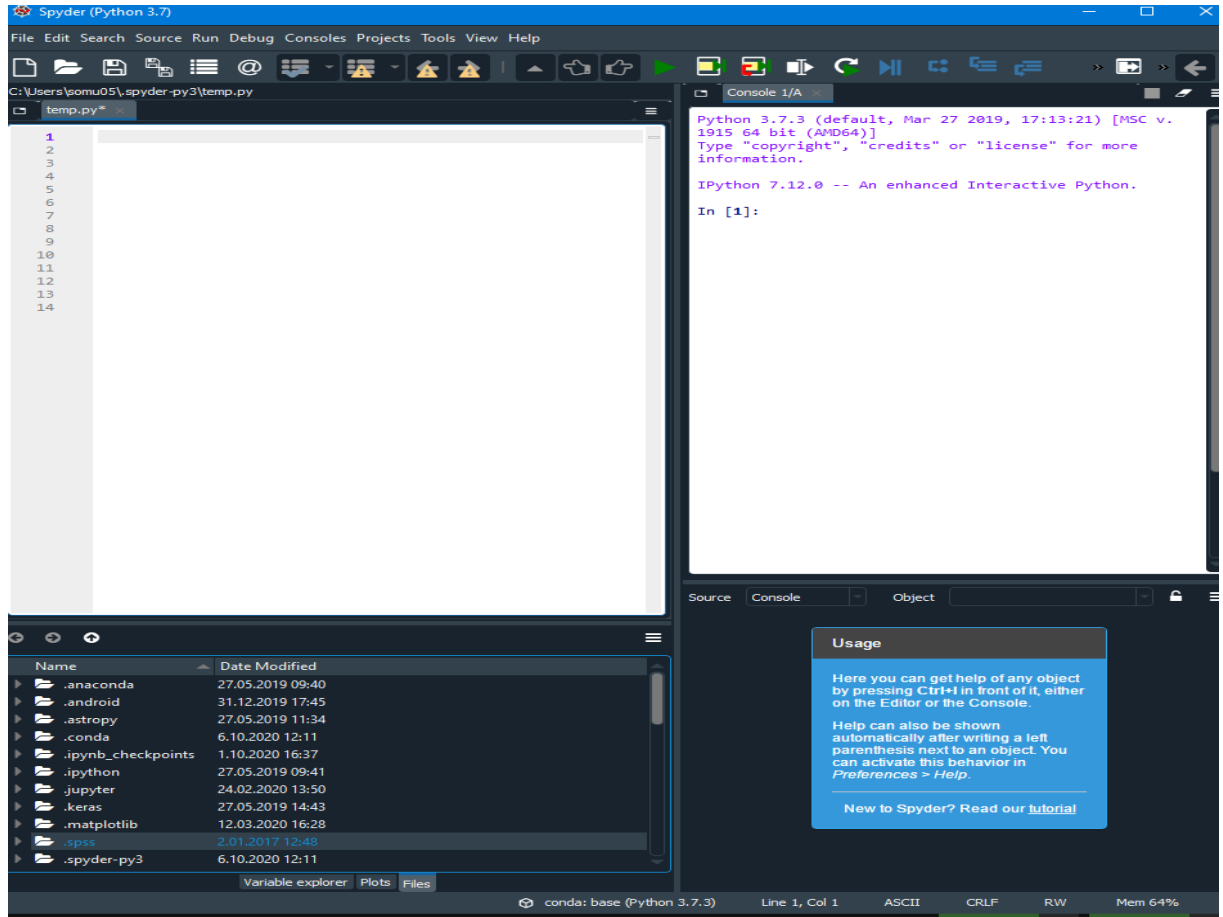
Navigator uygulamasında aşağıdaki uygulamalar varsayılan olarak mevcuttur:

- JupyterLab
- Jupyter Not Defteri
- QtConsole
- Spyder
- Glue_ (yazılım)
- Orange
- RStudio
- Visual Studio Code

Makine öğrenmesi

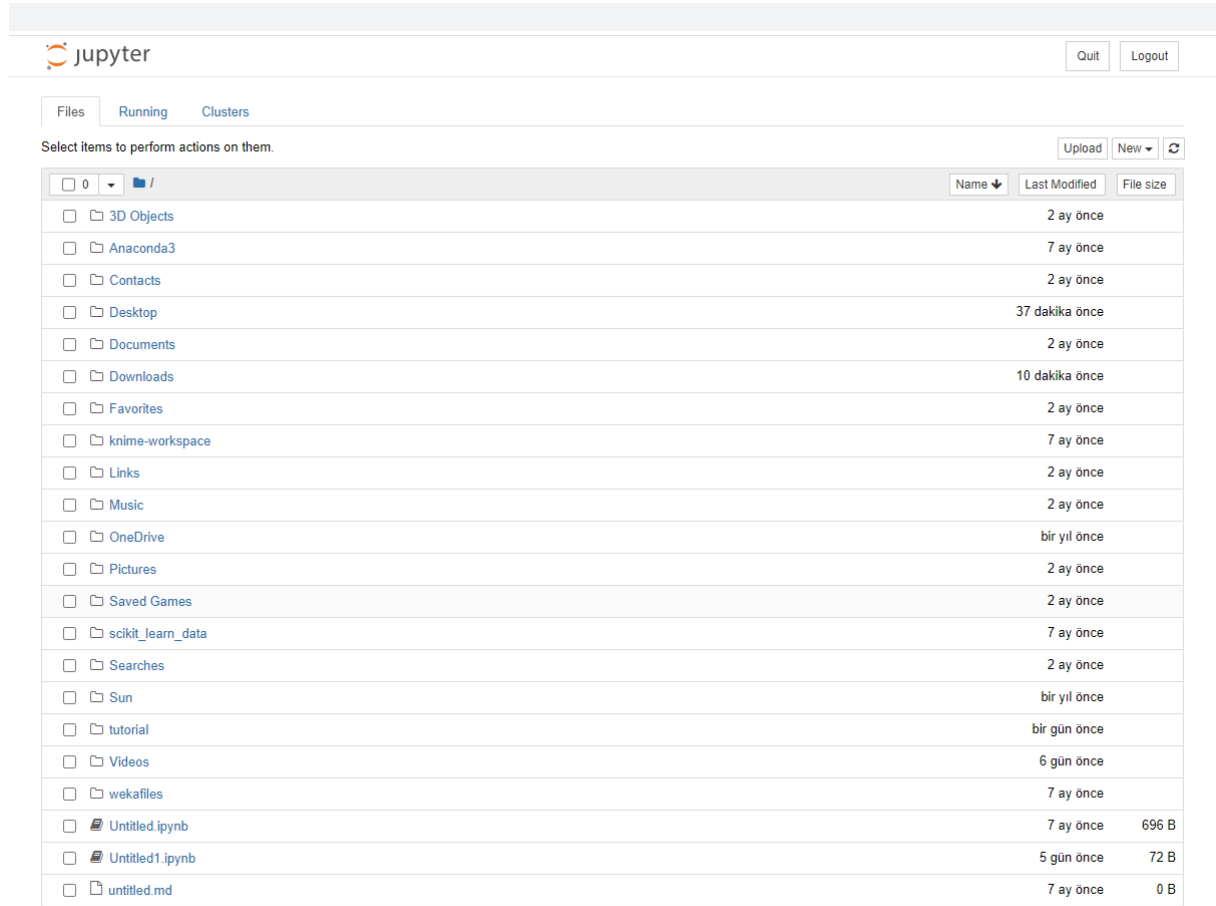


Spyder(Python 3.7) menüsü aşağıda verilmiştir.



Jupyter notebook menüü aşağıda verilmiştir.

Makine öğrenmesi

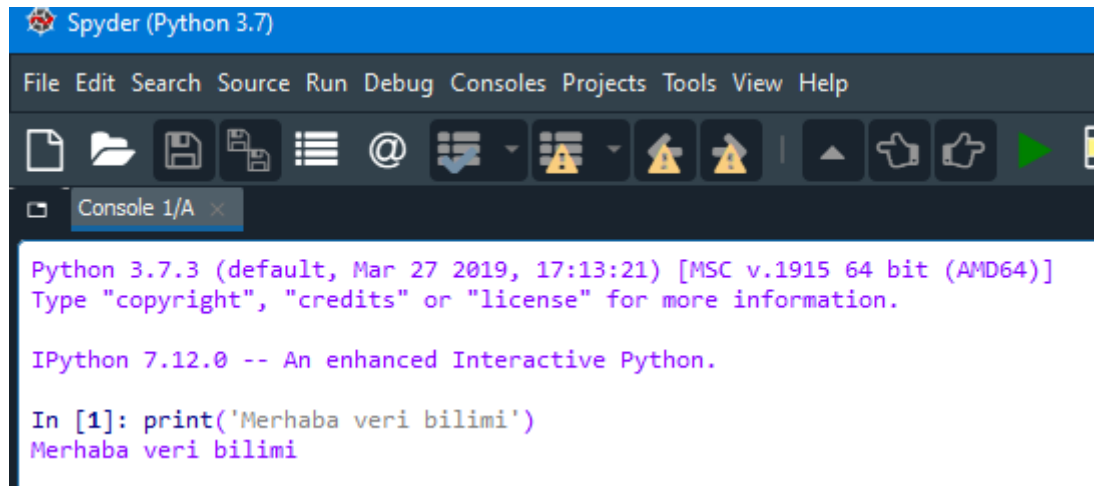


JupyterLab interface showing a file browser. The interface includes a top bar with the Jupyter logo and 'Quit' and 'Logout' buttons. Below the top bar are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a list of files and folders. The list is titled 'Select items to perform actions on them.' and includes buttons for 'Upload', 'New', and a refresh icon. The list has columns for 'Name', 'Last Modified', and 'File size'. The files and folders listed are:

Name	Last Modified	File size
0		
3D Objects	2 ay önce	
Anaconda3	7 ay önce	
Contacts	2 ay önce	
Desktop	37 dakika önce	
Documents	2 ay önce	
Downloads	10 dakika önce	
Favorites	2 ay önce	
knime-workspace	7 ay önce	
Links	2 ay önce	
Music	2 ay önce	
OneDrive	bir yıl önce	
Pictures	2 ay önce	
Saved Games	2 ay önce	
scikit_learn_data	7 ay önce	
Searches	2 ay önce	
Sun	bir yıl önce	
tutorial	bir gün önce	
Videos	6 gün önce	
wekafiles	7 ay önce	
Untitled.ipynb	7 ay önce	696 B
Untitled1.ipynb	5 gün önce	72 B
untitled.md	7 ay önce	0 B

VERİ TÜRLERİ:

1. **string**: Karakter dizileri. Tırnak içindeki her türlü karakterlerdir



```
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.12.0 -- An enhanced Interactive Python.

In [1]: print('Merhaba veri bilimi')
Merhaba veri bilimi
```

Makine öğrenmesi

```
In [2]: print("Merhaba veri bilimi")
Merhaba veri bilimi
```

```
In [5]: print("merhaba\
...: veri\
...: bilimi")
merhaba veri bilimi
```

\ ile aynı ifade birden fazla
satırda yazılabilir.

Not: string fonksiyonunda çift tırnak kullanmak şart değildir. Tek tırnak veya üç tırnak da kullanabiliriz. Üç tırnak için, üç tek tırnak veya üç çift tırnak karakterleri kullanılabilir.

2. integer: Tam sayılar. Pozitif, negatif ya da sıfır değeri alabilir. Kesirli değer içermez.

Örnek: 73, 0, -105

```
In [6]: print(55)
55

In [7]: print(123456789+1)
123456790
```

```
In [10]: 55
Out[10]: 55
```

Print komutu olmadan da tamsayı yazılabilir

3. float: Reel Sayılar. Kayan noktalı sayılar da denir. Ancak burada virgül yerine nokta kullanmamız gerekir. Örnek: 55.03, 4.0, -3.55, 0.0

```
In [11]: 5.5      In [12]: -55.03
Out[11]: 5.5      Out[12]: -55.03

In [14]: 5.03e55  In [15]: 45e-111
Out[14]: 5.03e+55 Out[15]: 4.5e-110
```

```
In [13]: type(55.03)
Out[13]: float
```

Veri türünü "type" komutu ile öğrenebiliriz

4. complex: Karmaşık sayılardır. Ör: 55+3j

```
In [16]: 55+3j
Out[16]: (55+3j)

In [17]: type(55+3j)
Out[17]: complex
```

5. bool: Herhangi bir ifadenin doğruluğunu veya yanlışlığını sorgular. True, False

Makine öğrenmesi

```
In [18]: type(True)
Out[18]: bool

In [19]: type(False)
Out[19]: bool
```

OPERATÖRLER :

Aritmetik Operatörler:

Toplama : + Çıkarma : - Çarpma : * Bölme : / Tam Bölme: // Üs Alma : ** Mod : %

```
In [21]: 7+5
Out[21]: 12

In [22]: 117/34
Out[22]: 3.4411764705882355

In [23]: 3**4
Out[23]: 81

In [24]: 45//4
Out[24]: 11

In [25]: 45%4
Out[25]: 1
```

FONKSİYONLAR:

1.print():

```
In [20]: print("Merhaba VERİ BİLİMİ DERSİ")
Merhaba VERİ BİLİMİ DERSİ
```

2. type(): Verilerin tipini sorgular

3. str(): Verileri string'e (karakter dizisi) çevirir.

4. int(): Verileri integer'a (tamsayı) çevirir.

5. float(): Verileri float'a (reel sayı) çevirir.

6. len(): stringlerin uzunluğunu ölçer

Diğer en çok kullanılan fonksiyonlar aşağıdadır.

Makine öğrenmesi

<code>abs()</code>	<code>dict()</code>	<code>help()</code>	<code>min()</code>	<code>setattr()</code>
<code>all()</code>	<code>dir()</code>	<code>hex()</code>	<code>next()</code>	<code>slice()</code>
<code>any()</code>	<code>divmod()</code>	<code>id()</code>	<code>object()</code>	<code>sorted()</code>
<code>ascii()</code>	<code>enumerate()</code>	<code>input()</code>	<code>oct()</code>	<code>staticmethod()</code>
<code>bin()</code>	<code>eval()</code>	<code>int()</code>	<code>open()</code>	<code>str()</code>
<code>bool()</code>	<code>exec()</code>	<code>isinstance()</code>	<code>ord()</code>	<code>sum()</code>
<code>bytearray()</code>	<code>filter()</code>	<code>issubclass()</code>	<code>pow()</code>	<code>super()</code>
<code>bytes()</code>	<code>float()</code>	<code>iter()</code>	<code>print()</code>	<code>tuple()</code>
<code>callable()</code>	<code>format()</code>	<code>len()</code>	<code>property()</code>	<code>type()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>list()</code>	<code>range()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>locals()</code>	<code>repr()</code>	<code>zip()</code>
<code>compile()</code>	<code>globals()</code>	<code>map()</code>	<code>reversed()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hasattr()</code>	<code>max()</code>	<code>round()</code>	
<code>delattr()</code>	<code>hash()</code>	<code>memoryview()</code>	<code>set()</code>	

```
In [12]: x = -2019  
         abs(x)
```

```
Out[12]: 2019
```

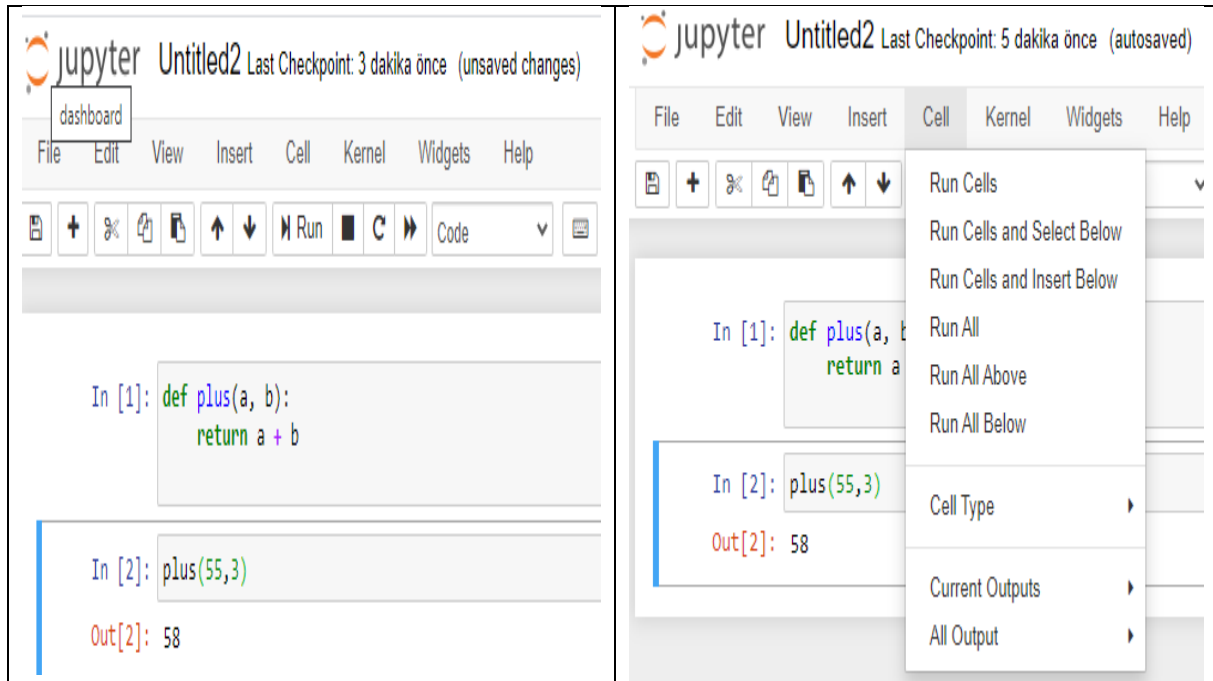
```
In [13]: d = -3.54632  
         int(d)
```

```
Out[13]: -3
```

```
In [14]: notlar = [21,34,57,75,81,83,87,90,93,99,31,37,41]  
         len(notlar)
```

```
Out[14]: 13
```


Makine öğrenmesi



```
In [6]: def fonk1(x):  
        return x*x+5
```

```
In [7]: fonk1(3)
```

```
Out[7]: 14
```

```
In [9]: def fonk2(x,y,z):  
        return x**2+y**3+z**2
```

```
In [10]: fonk2(3,55,12)
```

```
Out[10]: 166384.00694444444
```

```
In [11]: for i in range(10):  
        print(i)
```

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

```
In [15]: def ortalama(x):  
        "x değişkenlerinin ortalamasını hesaplama."  
        top_x = sum(x)  
        say1_x = len(x)  
        return top_x / say1_x
```

```
In [17]: ortalama([11,22,33,44,55])
```

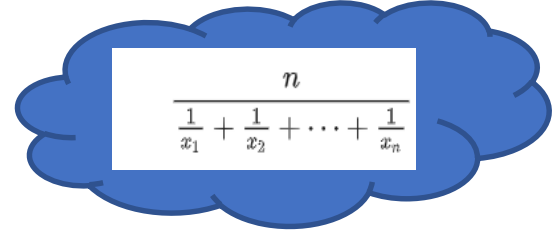
```
Out[17]: 33.0
```

Makine öğrenmesi

UYGULAMA:

n tane $X_1, X_2, X_3, \dots, X_n$ verildiğinde harmonik ortalamasını hesaplayalım.

HARMONİK ORTALAMA


$$\frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}$$

```
In [19]: 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
```

```
In [20]: harmonik_ortalama([10,20,30,40,50,60])
```

```
Out[20]: 24.489795918367346
```

KARŞILAŞTIRMA:

<	..den küçük
<=	..den küçük veya eşit
>	...den büyük
>=	..den büyük veya eşit
==	eşit
!=	eşit değil

```
In [21]: 23==77
```

```
Out[21]: False
```

```
In [22]: 34<=55
```

```
Out[22]: True
```

```
In [23]: 4!=5
```

```
Out[23]: True
```

Makine öğrenmesi

CSV formatındaki Veriyi yükleme :

Veri bilimi projeleri, verinin keşfedilmesi ve temizlenmesi ile başlar ve bu işlemler projelerin en çok zaman alan kısımlarıdır. Dolayısıyla verinin keşfi ve temizlenmesi sırasında işleri kolaylaştıracak bir takım kütüphanelere ihtiyaç duyulur. Pandas ile veri analizi yaparken kullanacağımız temel veri yapıları Seriler ve DataFrame'lerdir. Pandas, Python'da 2 veya 3 boyutlu veri dizileri oluşturmanıza izin veren bir başka Python kütüphanesi olan numpy üzerine kurulmuş güçlü bir veri analizi kütüphanesidir. Pandas farklı formattaki dosyalarıda DataFrame nesnesine okuma imkanı sağlıyor burada çalışacağımız dosya türü “.csv” olduğu için sadece bu tür dosyaların okunması ile ilgileneceğiz.

Örnek bir .csv dosyası aşağıda verilmiştir.

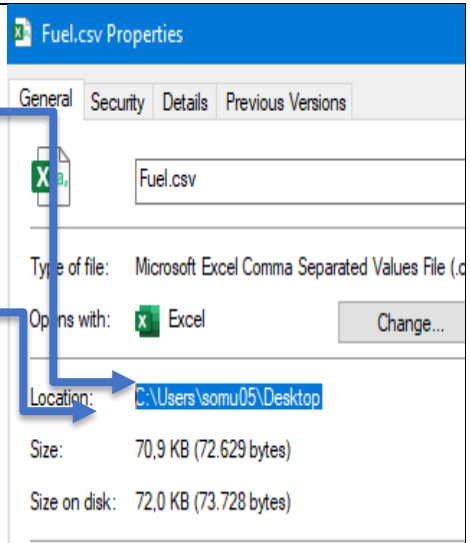
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V					
1	MODEL	YEAR	MAKE	MODEL	VEHICLE	CLASS	ENGINE	SIZE	CYLINDERS	TRANSMISSION	FUEL	TYPE	FUEL	CONSUMPTION	CITY	FUEL	CONSUMPTION	HWY	FUEL	CONSUMPTION	COMB	FUEL	CONSUMPTION	COMB	MPG	CO2	EMISSIONS
2	2014	ACURA	ILX	COMPACT	2.4	ASS	Z	9.9	6.7	8.5	33	196															
3	2014	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	7.7	9.6	29	221														
4	2014	ACURA	ILX	HYBRID	COMPACT	1.5	4	AV7	Z	6.5	8.5	9.48	136														
5	2014	ACURA	MDX	4WD	SUV	- SMALL	3.5	6	AS6	Z	12.7	9.1	11.1	25	255												
6	2014	ACURA	RDX	AWD	SUV	- SMALL	3.5	6	AS6	Z	12.1	8.7	10.6	27	244												
7	2014	ACURA	RLX	MID-SIZE	3.5	6	AS6	Z	11.9	7.7	10.28	230															
8	2014	ACURA	TL	MID-SIZE	3.5	6	AS6	Z	11.8	8.1	10.1	28	232														
9	2014	ACURA	TL	AWD	MID-SIZE	3.7	6	AS6	Z	12.8	9.11	1.25	255														
10	2014	ACURA	TL	AWD	MID-SIZE	3.7	6	M6	Z	13.4	9.5	11.6	24	267													
11	2014	ACURA	TSX	COMPACT	2.4	4	ASS	Z	10.6	7.5	9.2	31	212														
12	2014	ACURA	TSX	COMPACT	2.4	4	M6	Z	11.2	8.1	9.8	29	225														
13	2014	ACURA	TSX	COMPACT	3.5	6	ASS	Z	12.1	8.3	10.4	27	239														
14	2014	ASTON	MARTIN	DB9	MINICOMPACT	5.9	12	A6	Z	18	12.6	15.6	18	359													
15	2014	ASTON	MARTIN	RAPIDE	SUBCOMPACT	5.9	12	A6	Z	18	12.6	15.6	18	359													
16	2014	ASTON	MARTIN	V8 VANTAGE	TWO-SEATER	4.7	8	AM7	Z	17.4	11.3	14.7	19	338													
17	2014	ASTON	MARTIN	V8 VANTAGE	TWO-SEATER	4.7	8	M6	Z	18.1	12.2	15.4	18	354													
18	2014	ASTON	MARTIN	V8 VANTAGE S	TWO-SEATER	4.7	8	AM7	Z	17.4	11.3	14.7	19	338													
19	2014	ASTON	MARTIN	V8 VANTAGE S	TWO-SEATER	4.7	8	M6	Z	18.1	12.2	15.4	18	354													
20	2014	ASTON	MARTIN	VANQUISH	MINICOMPACT	5.9	12	A6	Z	18	12.6	15.6	18	359													
21	2014	AUDI	A4	COMPACT	2.4	AV8	Z	9.9	7.4	8.8	32	202															
22	2014	AUDI	A4	QUATTRO	COMPACT	2.4	AS8	Z	11.5	8.1	10.28	230															
23	2014	AUDI	A4	QUATTRO	COMPACT	2.4	M6	Z	10.8	7.5	9.3	30	214														
24	2014	AUDI	A5	CABRIOLET	QUATTRO	SUBCOMPACT	2.4	AS8	Z	11.5	8.1	10.28	230														
25	2014	AUDI	A5	QUATTRO	SUBCOMPACT	2.4	AS8	Z	11.5	8.1	10.28	230															
26	2014	AUDI	A5	QUATTRO	SUBCOMPACT	2.4	M6	Z	10.8	7.5	9.3	30	214														
27	2014	AUDI	A6	QUATTRO	MID-SIZE	2.4	AS8	Z	12.8	1.10	2.28	235															
28	2014	AUDI	A6	QUATTRO	MID-SIZE	3.6	AS8	Z	12.8	8.6	10.9	26	251														
29	2014	AUDI	A6	QUATTRO	TDI CLEAN DIESEL	MID-SIZE	3.6	AS8	D	9.8	6.4	8.3	34	224													
30	2014	AUDI	A7	QUATTRO	MID-SIZE	3.6	AS8	Z	13.1	8.8	11.2	25	258														
31	2014	AUDI	A7	QUATTRO	TDI CLEAN DIESEL	MID-SIZE	3.6	AS8	D	9.8	6.4	8.3	34	224													
32	2014	AUDI	A8	MID-SIZE	3.6	AS8	Z	13.1	8.8	11.2	25	258															
33	2014	AUDI	A8	MID-SIZE	4.8	AS8	Z	13.5	8.6	11.3	25	260															
34	2014	AUDI	A8	TDI CLEAN DIESEL	MID-SIZE	3.6	AS8	D	10	6.4	8.4	34	227														
35	2014	AUDI	A8	FULL-SIZE	3.6	AS8	Z	13.1	8.8	11.2	25	258															

Makine öğrenmesi

```
In [12]: import pandas as pd
import numpy as np
data=pd.read_csv("C:/Users/somu05/Desktop/fuel.csv")
data.head()
```

VEYA;

```
In [13]: import pandas as pd
import numpy as np
data=pd.read_csv(r"C:\Users\somu05\Desktop\fuel.csv")
data.head()
```



Dosyamız artık data isminde bir DataFrame nesnesine dönüştü.

Out[13]:	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINE SIZE	CYLINDERS	TRANSMISSION	FUELTYPE	FUELCONSUMPTION_CITY	FUELCONSUMPTION_HW
0	2014	ACURA	ILX	COMPACT	2.0	4	AS5	Z	9.9	6
1	2014	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	7
2	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	6.0	5
3	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	12.7	9
4	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	8

data'nın satır ve sütun sayısını öğrenmek için,

```
In [14]: print(data.shape)
```

(1067, 13)

1067 satır ve 13 sütun

Data veri setindeki sütunların isimlerini yazdırmak için

```
In [15]: print(data.columns)
```

```
Index(['MODELYEAR', 'MAKE', 'MODEL', 'VEHICLECLASS', 'ENGINE SIZE', 'CYLINDERS',
      'TRANSMISSION', 'FUELTYPE', 'FUELCONSUMPTION_CITY',
      'FUELCONSUMPTION_HWY', 'FUELCONSUMPTION_COMB',
      'FUELCONSUMPTION_COMB_MPG', 'CO2EMISSIONS'],
      dtype='object')
```

Makine öğrenmesi

Veri setinin ilk altı satırını yazdırmak için..

In [16]: `data.head(6)`

Out[16]:

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINE SIZE	CYLINDERS	TRANSMISSION	FUELTYPE	FUELCONSUMPTION_CITY	FUELCONSUMPTION_Hwy
0	2014	ACURA	ILX	COMPACT	2.0	4	AS5	Z	9.9	13.4
1	2014	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	15.4
2	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	6.0	9.7
3	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	12.7	18.7
4	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	17.9
5	2014	ACURA	RLX	MID-SIZE	3.5	6	AS6	Z	11.9	17.4

In [17]: `data.tail(3)`

Out[17]:

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINE SIZE	CYLINDERS	TRANSMISSION	FUELTYPE	FUELCONSUMPTION_CITY	FUELCONSUMPTION_Hwy
1064	2014	VOLVO	XC70 AWD	SUV - SMALL	3.0	6	AS6	X	13.4	19.4
1065	2014	VOLVO	XC70 AWD	SUV - SMALL	3.2	6	AS6	X	12.9	18.8
1066	2014	VOLVO	XC90 AWD	SUV - STANDARD	3.2	6	AS6	X	14.9	21.4

Data veri seti içindeki sayısal verilerin sayı, ortalama, standart sapma, minimum, 1.çeyrek, 3.çeyrek ve maksimum gibi istatistiksel hesaplamaları yapmak için “`print(data.describe())`” komutu kullanılır.

In [18]: `print(data.describe())`

```
count    1067.000000  1067.000000  1067.000000  1067.000000  \
mean      2014.000000   3.346298   5.794752   13.296532
std         0.000000   1.415895   1.797447   4.101253
min      2014.000000   1.000000   3.000000   4.600000
25%      2014.000000   2.000000   4.000000  10.250000
50%      2014.000000   3.400000   6.000000  12.600000
75%      2014.000000   4.300000   8.000000  15.550000
max      2014.000000   8.400000  12.000000  30.200000

count    1067.000000  1067.000000  1067.000000  \
mean         9.474602  11.580881  26.441425
std         2.794510   3.485595   7.468702
min         4.900000   4.700000  11.000000
25%         7.500000   9.000000  21.000000
50%         8.800000  10.900000  26.000000
75%        10.850000  13.350000  31.000000
max        20.500000  25.800000  60.000000

count    1067.000000
mean      256.228679
std       63.372304
min      108.000000
25%      207.000000
50%      251.000000
75%      294.000000
max      488.000000
```

Makine öğrenmesi

Sayısal olmayan tüm sütunların istatistiksel özetini include=['O'] parametresi kullanılarak elde edilebilir.

```
In [21]: data.describe(include=['O'])
```

Out[21]:

	MAKE	MODEL	VEHICLECLASS	TRANSMISSION	FUELTYPE
count	1067	1067	1067	1067	1067
unique	39	663	16	22	4
top	FORD	F150 FFV 4X4	MID-SIZE	A6	X
freq	90	8	178	222	514