

Projelerim (/github/murderuo/Projelerim/tree/python) / kernel.ipynb (/github/murderuo/Projelerim/tree/python/kernel.ipynb)

```
In [1]: import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory

import os
print(os.listdir("../input"))

# Any results you write to the current directory are saved as output.
```

```
['avocado.csv']
```

```
In [2]: data=pd.read_csv("../input/avocado.csv")

data.head(10)
```

Out[2]:

	Unnamed: 0	Date	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags	XLarge Bags	type	year
0	0	2015-12-27	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	conventional	2015
1	1	2015-12-20	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	conventional	2015
2	2	2015-12-13	0.93	118220.22	794.70	109149.67	130.50	8145.35	8042.21	103.14	0.0	conventional	2015
3	3	2015-12-06	1.08	78992.15	1132.00	71976.41	72.58	5811.16	5677.40	133.76	0.0	conventional	2015
4	4	2015-11-29	1.28	51039.60	941.48	43838.39	75.78	6183.95	5986.26	197.69	0.0	conventional	2015
5	5	2015-11-22	1.26	55979.78	1184.27	48067.99	43.61	6683.91	6556.47	127.44	0.0	conventional	2015
6	6	2015-11-15	0.99	83453.76	1368.92	73672.72	93.26	8318.86	8196.81	122.05	0.0	conventional	2015
7	7	2015-11-08	0.98	109428.33	703.75	101815.36	80.00	6829.22	6266.85	562.37	0.0	conventional	2015
8	8	2015-11-01	1.02	99811.42	1022.15	87315.57	85.34	11388.36	11104.53	283.83	0.0	conventional	2015
9	9	2015-10-25	1.07	74338.76	842.40	64757.44	113.00	8625.92	8061.47	564.45	0.0	conventional	2015

```
In [3]: data.columns=[i.replace(" ","_") for i in data.columns]
data.columns=[i.replace(":","") for i in data.columns]
data.columns=[i.lower() for i in data.columns]
data.columns=[i.replace("unnamed_0","number") for i in data.columns]
```

```
data.columns
```

```
Out[3]: Index(['number', 'date', 'averageprice', 'total_volume', '4046', '4225',
              '4770', 'total_bags', 'small_bags', 'large_bags', 'xlarge_bags', 'type',
              'year', 'region'],
              dtype='object')
```

```
In [4]: cropped_data=data.head(15)

cropped_data
```

Out[4]:

	number	date	averageprice	total_volume	4046	4225	4770	total_bags	small_bags	large_bags	xlarge_bags
0	0	2015-12-27	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0
1	1	2015-12-20	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0
2	2	2015-12-13	0.93	118220.22	794.70	109149.67	130.50	8145.35	8042.21	103.14	0.0
3	3	2015-12-06	1.08	78992.15	1132.00	71976.41	72.58	5811.16	5677.40	133.76	0.0
4	4	2015-11-29	1.28	51039.60	941.48	43838.39	75.78	6183.95	5986.26	197.69	0.0
5	5	2015-11-22	1.26	55979.78	1184.27	48067.99	43.61	6683.91	6556.47	127.44	0.0
6	6	2015-11-15	0.99	83453.76	1368.92	73672.72	93.26	8318.86	8196.81	122.05	0.0
7	7	2015-11-08	0.98	109428.33	703.75	101815.36	80.00	6829.22	6266.85	562.37	0.0
8	8	2015-11-01	1.02	99811.42	1022.15	87315.57	85.34	11388.36	11104.53	283.83	0.0
9	9	2015-10-25	1.07	74338.76	842.40	64757.44	113.00	8625.92	8061.47	564.45	0.0
10	10	2015-10-18	1.12	84843.44	924.86	75595.85	117.07	8205.66	7877.86	327.80	0.0
11	11	2015-10-11	1.28	64489.17	1582.03	52677.92	105.32	10123.90	9866.27	257.63	0.0
12	12	2015-10-04	1.31	61007.10	2268.32	49880.67	101.36	8756.75	8379.98	376.77	0.0
13	13	2015-09-27	0.99	106803.39	1204.88	99409.21	154.84	6034.46	5888.87	145.59	0.0
14	14	2015-09-20	1.33	69759.01	1028.03	59313.12	150.50	9267.36	8489.10	778.26	0.0

```
In [5]: d_data=cropped_data.drop(columns=["date","4770","total_bags","small_bags","xlarge_bags","type","year","reg:

d_data=d_data.drop(columns="large_bags")

d_data
d_data.columns
```

Out[5]: Index(['number', 'averageprice', 'total_volume', '4046', '4225'], dtype='object')

```
In [6]: averageprice=input("Value of your average price column: ")
total_volume=input("Value of your total volume column: ")
ff6=input("Value of your 4046 column: ")
f2t5=input("Value of your 4225 column: ")
input_dict={"number":len(d_data),"averageprice":[averageprice],"total_volume":[total_volume],"4046":[ff6],
input_data=pd.DataFrame(input_dict)

d_data=d_data.append(input_data,ignore_index=True)
d_data

Value of your average price column: 23
Value of your total volume column: 23
Value of your 4046 column: 23
Value of your 4225 column: 23
```

Out[6]:

	number	averageprice	total_volume	4046	4225
0	0	1.33	64236.6	1036.74	54454.8
1	1	1.35	54877	674.28	44638.8
2	2	0.93	118220	794.7	109150
3	3	1.08	78992.1	1132	71976.4
4	4	1.28	51039.6	941.48	43838.4
5	5	1.26	55979.8	1184.27	48068
6	6	0.99	83453.8	1368.92	73672.7
7	7	0.98	109428	703.75	101815
8	8	1.02	99811.4	1022.15	87315.6
9	9	1.07	74338.8	842.4	64757.4
10	10	1.12	84843.4	924.86	75595.9
11	11	1.28	64489.2	1582.03	52677.9
12	12	1.31	61007.1	2268.32	49880.7
13	13	0.99	106803	1204.88	99409.2
14	14	1.33	69759	1028.03	59313.1
15	15	23	23	23	23

SORU 2 CEVAP: Mehmet daha mantıklı sorular sormuştur. Çünkü tuttuğumuz sayı 98 olsa idi Ahmet bize 98 kere tuttuğumuz sayıyı soracaktı ve buda uzun sürecekti. Ancak Mehmet in düşüncesinden hareketle; tuttuğumuz sayının 50 olup olmadığını değilse büyük olup olmadığını soracak(adım 1), aldığı cevaba göre kalan sayıların içinden (51-100)bir tahmin daha yapacak. Bu sefer 75 sayısı olup olmadığını değilse büyük olup olmadığını soracak (adım 2). Yine aynı şekilde kalan sayılar arasından tahmin etmeye çalışacak ve 87 sayısı olup olmadığını soracak değilse büyük olup olmadığını soracak (adım 3). Kalan sayıların (88-100) ortasındaki sayı (94) olup olmadığını soracak (adım 4). Yine yanılırsa bu sefer de kalan sayılardan (95-100) bir tahmin yapacak ve en çok 3 adım daha sonra yani toplamda 6 adımda tahmin ettiğimiz 98 sayısını bulabilecektir.

SORU 4 CEVAP

1936 daki seçimlerde yanılmasının sebebi ilk olarak çalışma alanının dar olması. Şöyleki; otomobillerin tarih sahnesine çıkışı ile 1936 yılındaki seçim arasında çok bir zaman yok (30 küsür yıl)özellikle o zamanki teknolojiyi düşünürsek o yıldaki araç sahibinin nüfusa oranının az olduğunu düşünüyorum. Aynı şekilde telefon sahiplerinin de . Böylece 1936 yılındaki alınan geri bildirimlerin toplam nüfusa oranı çok azdı. İkinci olarak o tarihte araç ve telefon sahibi olan kişiler muhtemelen varlıklı kişilerdi. İlk düşüncemde azınlıkta olan bu kişilerin toplumun genel görüşünü ya da hayatını yaşamadığını da varsayabilirim.

SORU 5 CEVAP İşlemi matematiksel fonksiyonda yazamadım ancak pytonda bunu yapabilirim.

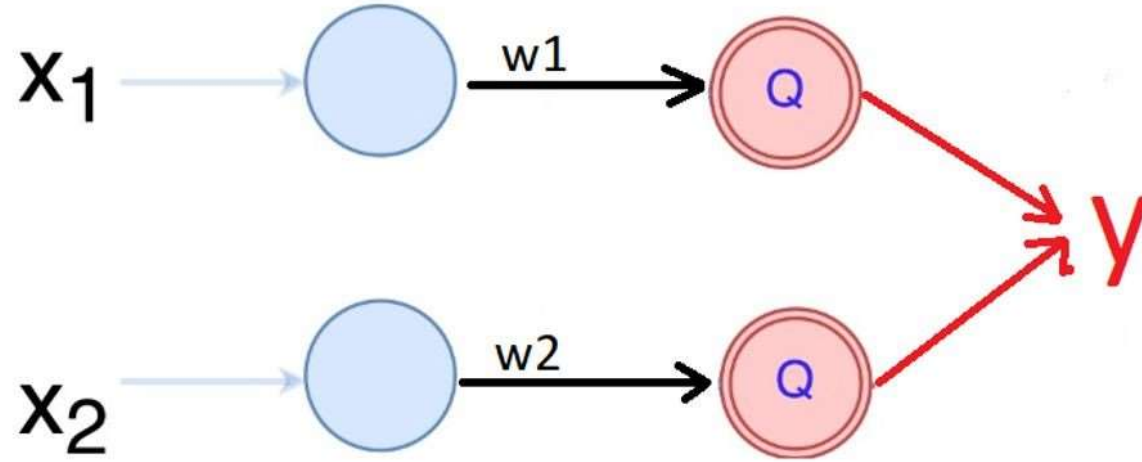
```
In [7]: def kenar(a,b):
        if a > b:
            if a == b: print("karedir")
            elif a < 0 and b < 0: print("pozitif bir değeri giriniz.")
            else:
                mod = a % b
                while mod != 0:
                    a = b
                    b = mod
                    mod = a % b
                print(b)
        else:
            print("Kısa uzun kenar değişimi yapılarak hesaplandı")
            c = 0
            c = a
            a = b
            b = c
            kenar(a,b)

b = int(input("kısa kenarı giriniz: "))
a = int(input("uzun kenarı giriniz: "))

kenar(a,b)

kısa kenarı giriniz: 12
uzun kenarı giriniz: 5
Kısa uzun kenar değişimi yapılarak hesaplandı
1
```

SORU 6 CEVAP



Yapay sinir ağında $w_1=0.9$, $w_2=1.1$ ve $Q \geq 1$ için $f(x_1)=(x_1 \times w_1)$ ve $g(x_2)=(x_2 \times w_2)$ fonksiyonları vardır

x_1	x_2	$f(x_1)$	$g(x_2)$	$f(x_1) \text{ or } g(x_2) \geq 1$	y değeri
0	0	0×0.9	0×1.1	HAYIR	0
0	1	0×0.9	1×1.1	<u>Evet</u> $g(x_2)$	1
1	0	1×0.9	0×1.1	<u>Evet</u> $f(x_1)$	1
1	1	1×0.9	1×1.1	<u>Evet</u> $g(x_2)$	1

SORU 7 CEVAP

S-eğrisi zaman değişim grafiğini göstermektedir. Ve bu egride gözüktüğü gibi mısırların patladıktan sonraki değişimi hariç tutulmuştur. Bu haliyle bence eksik bir grafikdir.

SORU 8 CEVAP

S-eğrisinin bu grafiklere göre avantajı değişimin keskin olmamasıdır.