

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
```

```
# Outcome = 1 Diabet/Şeker Hastası
```

```
# Outcome = 0 Sağlıklı
```

```
data = pd.read_csv("diabetes.csv")
```

```
data.head()
```

```
# Veri setimizdeki ilk 5 veriyi listeliyoruz:
```

```
seker_hastalari = data[data.Outcome == 1] # Outcome 1, veri setimizdeki hastaları tanımlıyor.
```

```
saglikli_insanlar = data[data.Outcome == 0] # Outcome 0 veri setimizdeki sağlıklı insanları tanımlıyor.
```

```
# Şimdilik sadece glouucose'a bakarak örnek bir çizim yapalım:
```

```
# Programımızın sonunda makine öğrenme modelimiz sadece glikoza değil tüm diğer verilere bakarak bir tahmin yapacaktır..
```

```
plt.scatter(saglikli_insanlar.Age, saglikli_insanlar.Glucose, color="green", label="sağlıklı", alpha = 0.6)
```

```
plt.scatter(seker_hastalari.Age, seker_hastalari.Glucose, color="red", label="diabet hastası", alpha = 0.6)
```

```
plt.xlabel("Age") # Yaş
```

```
plt.ylabel("Glucose") # Glikoz oranı
```

```
plt.legend()
```

```
plt.show()
```

```
# x ve y eksenlerini belirleyelim
y = data.Outcome.values
x_ham_veri = data.drop(["Outcome"],axis=1) # Tablomuzdan outcome yi kaldırdık.
# Outcome sütununu(dependent variable, sonuç) çıkarıp sadece independent variables bırakıyoruz
# Çünkü KNN algoritması x değerleri içerisinde gruplandırma yapacak..
```

```
# NORMALİZATİON yapıyoruz - x_ham_veri içerisindeki değerleri sadece 0 ve 1 arasında olacak
şekilde hepsini güncelliyoruz

# Bunu yapma nedenimiz örneğin yaş dataları 80 e kadar giderken hamilelik 3-4-5 şeklinde gidiyor.
Fakat bizim hepsine aynı ağırlığı vermemiz gerekiyor.

# Bu nedenle Normalizasyon yapıyoruz. Böylece KNN algoritması yanılmayacak!!!

x = (x_ham_veri - np.min(x_ham_veri))/(np.max(x_ham_veri)-np.min(x_ham_veri))
```

```
print("Normalization öncesi ham veriler:\n")
print(x_ham_veri.head()) # önce
```

```
print("\n\nNormalization sonrası yapay zekaya eğitim için vereceğimiz veriler:\n")
print(x.head()) # sonra
```

TEST - TRAIN İŞLEMİ

train datamız ile test datamızı ayırıyoruz

train datamız sistemin sağlıklı insan ile hasta insanı ayırt etmesini öğrenmek için kullanılacak -->
Yapay zeka eğitme işlemi

test datamız ise algoritmamızın ne kadar doğru tahmin yapabildiğini ölçmek için kullanılacak -->
Yapay zeka test işlemi

x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.25,random_state=1) # 768 hastanın
%25 tanesini test için kullanacağım

knn modelimizi oluşturuyoruz.

knn = KNeighborsClassifier(n_neighbors = 3) # n_neighbors = k # K Değerimiz (En yakın kaç komşuya
bakılacak?)

knn.fit(x_train,y_train) # eğitme (fit)

prediction = knn.predict(x_test) #test (prediction)

print("K=3 için Test verilerimizin doğrulama testi sonucu ", knn.score(x_test, y_test))

k kaç olmalı ?

en iyi k değerini belirleyelim..

sayac = 1

for k in range(1,11):

 knn_yeni = KNeighborsClassifier(n_neighbors = k)

 knn_yeni.fit(x_train,y_train)

 print(sayac, " ", "Doğruluk oranı: %", knn_yeni.score(x_test,y_test)*100)

 sayac += 1

#En doğru sonuç k=7