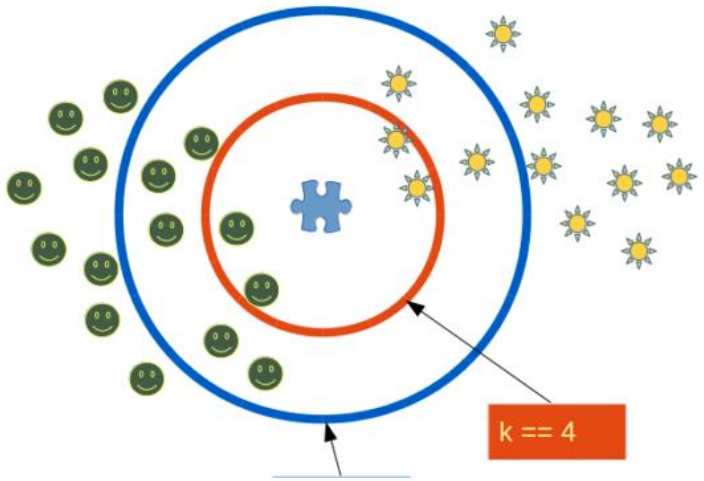
Örnek Tabanlı Öğrenme (Instance-Based Learning)

K-En Yakın Komşu (K-Nearest Neighbor)



K-En Yakın Komşu (K-Nearest Neighbor – K-NN)

- K-NN algoritması, Thomas M. Cover ve Peter E. Hart tarafından önerilen, örnek veri noktasının bulunduğu sınıfın ve en yakın komşunun, k değerine göre belirlendiği bir sınıflandırma yöntemidir.

- Denetimli (supervised) öğrenmede sınıflandırma ve regresyon için kullanılan algoritmalardan biridir.

- Diğer denetimli öğrenme algoritmalarının aksine, eğitim aşamasına sahip değildir. Eğitim ve test hemen hemen aynı şeydir.

K-En Yakın Komşu Avantajları

- Anlaşılması ve uygulanması oldukça kolaydır.
- Eğitim basamağı yoktur.
- Hem Sınıflandırma hem de Regresyon için kullanılabilir.
- K-NN hafıza bazlı bir yaklaşımdır. Sınıflandırıcı, yeni eğitim verilerini toplarken derhal adapte olur.
- Algoritma, gerçek zamanlı kullanım sırasında girdideki değişikliklere hızlı bir şekilde cevap verir.
- Çok sınıflı problem için uygulanması çok kolaydır. K-NN ekstra çaba göstermeden çoklu sınıfa uyarlanır.
- Kullanıcıya K-NN modelini oluştururken değişik uzaklık ölçütlerini seçme esnekliğini verir.

K-En Yakın Komşu Dezavantajları

- Bütün örneklerin saklanması için çok hafızaya ihtiyaç vardır.
- k-NN, dengesiz verilerde iyi performans göstermez.
- Veri kümesi büyüdükçe ve öznitelik sayısı arttıkça işlem yükü artar, algoritma verimliliği ve hızı azalır.
- Yeni bir örneği sınıflandırmak çok zaman alır (gelen yeni bir örneğin diğer örneklerle mesafesinin hesaplanması ve karşılaştırılması)
- K-NN, veri kümesindeki eksik değer sorunu ile başa çıkma kabiliyetine sahip değildir.
- Performans k komşu sayısı, uzaklık ölçütü ve öznitelik sayısı gibi parametre ve özelliklere bağlı olarak etkilenir.

K-En Yakın Komşu (K-Nearest Neighbor)

- **Benzerlik ile öğrenme:**
- Yeni karşılaşılan bir örnek, eğitim veri kümesinde yer alan örnekler ile arasındaki benzerliğe göre sınıflandırılmaktadır.

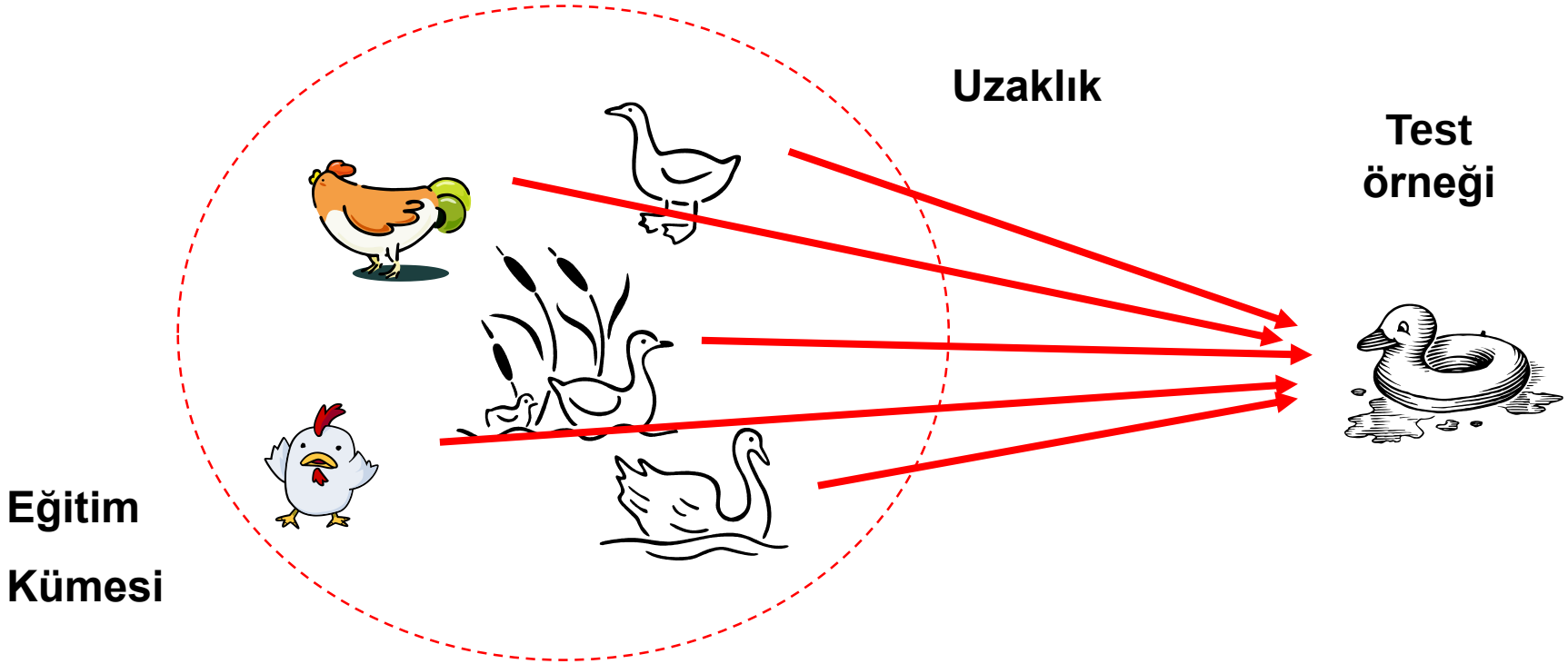
- Bir örnek, komşularının çoğunluk oyuyla sınıflandırılır; örnek, en yakın komşuları arasında en yaygın olan sınıfa verilir.

- k: komşu sayısı (pozitif bir tam sayı)

- Eğer k = 1 ise, örnek ona en yakın komşunun sınıfına atanır.

K-En Yakın Komşu (K-Nearest Neighbor)

Yeni bir örnek hangi sınıfta dahil olacak?
Yeni örneğin eğitim kümesi içerisinde yer alan bütün örnekler ile arasındaki uzaklığı (distance) hesaplanır.
Yeni örnek için eğitim kümesi içerisindeki en yakın k örnek seçilir.
Yeni örnek en yakın k komşusu arasında en fazla olan sınıfa atanır.



K-En Yakın Komşu Parametreleri

- K-NN algoritmasının performansında etkili ve önemli parametreler **uzaklık ölçütü, komşu sayısı (k) ve ağırlıklandırma** yöntemidir.

- K-NN algoritmasının performansı için kritik öneme sahip noktalardan birisi örnekler arası uzaklığın nasıl ölçümleneceğidir.

- Uzaklık ölçütü olarak genellikle Euclidean uzaklığı (Euclidean Distance) alınır ya da (Minkowski, Manhattan, Mahalanobis, Chebyschev, Dilca vs.) bir başka uzaklık ölçütü kullanılarak hesaplanabilir.

Uzaklık Ölçütleri

- İki veri noktası arasındaki farklılığı ölçer.
- Uzaklık ölçülerinin özellikleri :

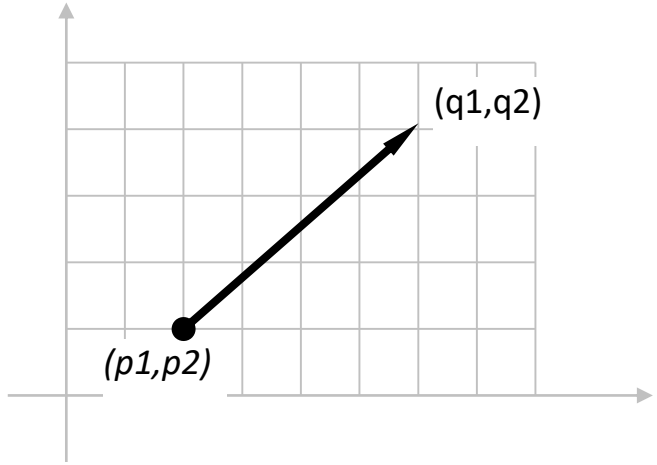
- d(X, Y) pozitif kesindir: eğer $(X \neq Y)$, $d(X, Y) > 0$
  eğer $(X = Y)$, $d(X, Y) = 0$

- d(X, Y) simetriktir: $d(X, Y) = d(Y, X)$

- d(X, Y) üçgen eşitsizliğini sağlar: $d(X, Y) + d(Y, Z) \geq d(X, Z)$
- bir üçgenin herhangi bir kenarı, diğer iki kenarın toplamından küçük veya eşit olmalı.

Euclidean uzaklıklığı (Euclidean Distance)

- Euclidean Distance: uzaydaki herhangi iki nokta arasındaki uzaklığın aralarında çizilen düz bir çizginin uzunluğuna tekabül ettiği 1, 2, 3 veya daha yüksek boyutlara da genelleştirilebilen doğrusal bir metriktir.
- Bir boyutlu Euclidean uzayında, iki nokta arasındaki uzaklık, sadece koordinatlar arasındaki farkın mutlak değeridir.
- İki boyutlu Euclidean uzayında P ve Q olarak iki nokta alınır. P koordinatları (p1, p2) ve Q koordinatları (q1, q2) olarak tanımlanırsa Euclidean uzaklığı;



$$d(P,Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$$

- n boyutlu Euclidean uzayında;

$$d(P,Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots.. + (p_n - q_n)^2}$$

$$d(P,Q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

Euclidean uzaklıklığı (Euclidean Distance)

Euclidean uzaklığı, iki nokta arasındaki doğrusal uzaklıktır.

P (p_1, p_2) ve Q (q_1, q_2) x-y düzleminde 2 nokta ise;

Euclidean dist: $d(P,Q)=((p_1, p_2), (q_1, q_2)) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$

$d(P,Q) = ((2, -1), (-2, 2)) = \sqrt{(2 - (-2))^2 + ((-1) - 2)^2}$

$= \sqrt{(2 + 2)^2 + (-1 - 2)^2}$

$= \sqrt{(4)^2 + (-3)^2}$

$= \sqrt{16 + 9}$

$= \sqrt{25}$

$= 5$

Euclidean uzaklıklığı (Euclidean Distance)

Data	Var1	Var2	Var3
Case1	1	1	1
Case2	1	1	0
Case3	2	2	2
Case4	10	10	10
Case5	11	11	11
Case6	10	5	0

Euclidean distance	Case1	Case2	Case3	Case4	Case5	Case6
Case1	0	1,000	1,732	15,588	17,321	9,899
Case2	1,000	0	2,449	16,186	17,916	9,849
Case3	1,732	2,449	0	13,856	15,588	8,775
Case4	15,588	16,186	13,856	0	1,732	11,180
Case5	17,321	17,916	15,588	1,732	0	12,570
Case6	9,899	9,849	8,775	11,180	12,570	0

Euclidean dist ((case1), (case2)) = $\sqrt{(1-1)^2 + (1-1)^2+(1-0)^2} = 1$

Euclidean dist ((case1), (case3)) = $\sqrt{(1-2)^2 + (1-2)^2+(1-2)^2} = 1{,}732$

Euclidean dist ((case1), (case4)) = $\sqrt{(1-10)^2 + (1-10)^2+(1-10)^2} = 15{,}588$

Euclidean uzaklıklığı (Euclidean Distance)

Fırat:

Yaş=35

Gelir=35bin

Kredi kartı sayısı=3

- Necla:
- Yaş=22
- Gelir=50bin
- Kredi kartı sayısı=2

Euclidean Distance (Fırat, Necla)=sqrt $[(35-22)^2+(35bin-50bin)^2+(3-2)^2]$

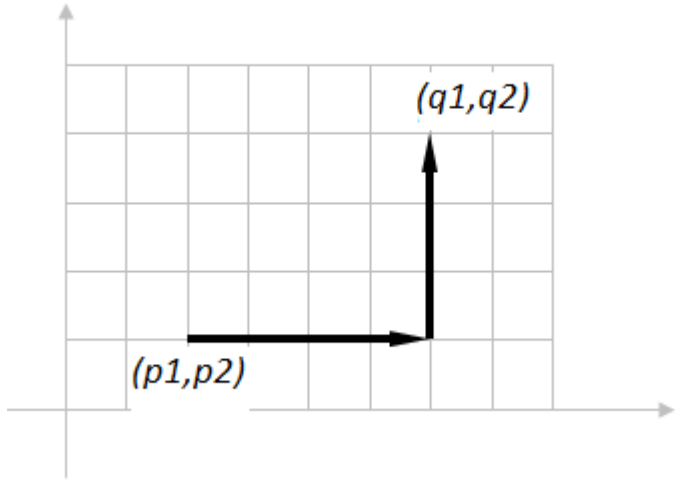
Manhattan uzaklıklığı (Manhattan Distance)

- İki boyutlu Manhattan uzayında P ve Q olarak iki nokta alınır. P koordinatları (p1, p2) ve Q koordinatları (q1, q2) tanımlanırsa Manhattan uzaklığı;

$$d(P,Q) = |p_1 - q_1| + |p_2 - q_2| + \ldots\ldots + |p_n - q_n|$$

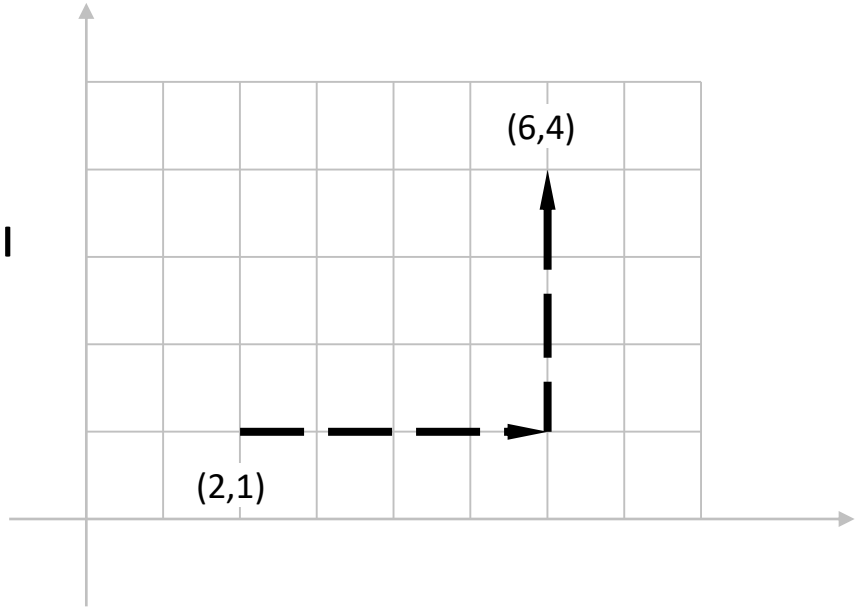
$$d_1(P,Q) = \sum_{i=1}^{n} |p_i - q_i|$$

- denklemi ile hesaplanır. Denklemde n değişken sayısı, pi ve qi sırasıyla P=(p1,p2,p3,…pn) ve Q=(q1,q2,q3,…qn) vektörlerinin değişkenleridir.
- Burada iki nokta arasındaki uzaklık bu noktaların koordinatlarının mutlak farklarının toplamı ile ölçülür.
- Bu uzaklık ölçüsü ***şehir blok uzaklığı (city block distance)*** gibi bir isimle de anılır.



Manhattan uzaklıklığı (Manhattan Distance)

- (p1, p2) ve (q1, q2) düzlem üzerinde 2 nokta ise;
- (2,1) ve (6,4) noktaları arasındaki Manhattan uzaklığı
- $d(P,Q)=|p1 - q1| + |p2 - q2|$
- $d(P,Q) = |2 - 6| + |1 - 4| = 4 + 3 = 7$



Manhattan uzaklıklığı (Manhattan Distance)

Data	Var1	Var2	Var3
Case1	1	1	1
Case2	1	1	0
Case3	2	2	2
Case4	10	10	10
Case5	11	11	11
Case6	10	5	0

Manhattan distance	Case1	Case2	Case3	Case4	Case5	Case6
Case1	0	1	3	27	30	14
Case2	1	0	4	28	31	13
Case3	3	3	0	24	27	13
Case4	27	28	24	0	3	15
Case5	30	31	27	3	0	18
Case6	14	13	13	15	18	0

Minkowski uzaklıklığı (Minkowski Distance)

- Minkowski Distance : $$d_r\left(P,Q\right)=\left\{\sum_{i=1}^{n}\left|p_i-q_i\right|^r\right\}^{\frac{1}{r}}$$

$$d_r(p,q)=((p_1-q_1)^r+(p_2-q_2)^r+.....+(p_n-q_n)^r)^{\frac{1}{r}}$$

- denklemi ile hesaplanır. Denklemde n değişken sayısı, pi ve qi sırasıyla P=(p1, p2, p3,…pn) ve Q =(q1,q2, q3,…qn) vektörlerinin değişkenleridir.
- r: Minkowski uzaklık indeksi
- r, tipik olarak 1 ile 2 arasında bir değere ayarlanır. r'nin 1'den küçük değerleri için, yukarıdaki formül, üçgen eşitsizliği karşılanmadığından geçerli bir mesafe ölçümü tanımlamaz.
- r = 1, olduğunda Minkowshi uzaklığı, Manhattan uzaklığı ile aynıdır.
- r = 2, olduğunda Minkowshi uzaklığı, Euclidean uzaklığı ile aynıdır.
- Minkowski uzaklığı Euclidean ve Manhattan uzaklıklarının bir genellemesidir.

Minkowski uzaklıklığı (Minkowski Distance)

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

Manhattan Distance

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Euclidean Distance

L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Chebyshev distance

Distance Matrix

Diğer Uzaklık Ölçüleri

Numerical Data

- [ ] Chessboard Distance
- [ ] Bray Curtis Distance
- [ ] Canberra Distance
- [ ] Cosine Distance
- [ ] Correlation Distance
- [ ] Binary Distance
- [ ] Time Warping Distance

Boolean Data

- [ ] Hamming Distance
- [ ] Jaccard Dissimilarity
- [ ] Matching Dissimilarity
- [ ] Dice Dissimilarity

String Data

- [ ] Edit Distance
- [ ] Damerau-Levenshtein
- [ ] Hamming Distance
- [ ] Smith-Waterman Similarity
- [ ] Needleman-Wunsch Similarity

Images & Colors

- [ ] Image Distance
- [ ] Color Distance

k-En Yakın Komşu Yöntemi

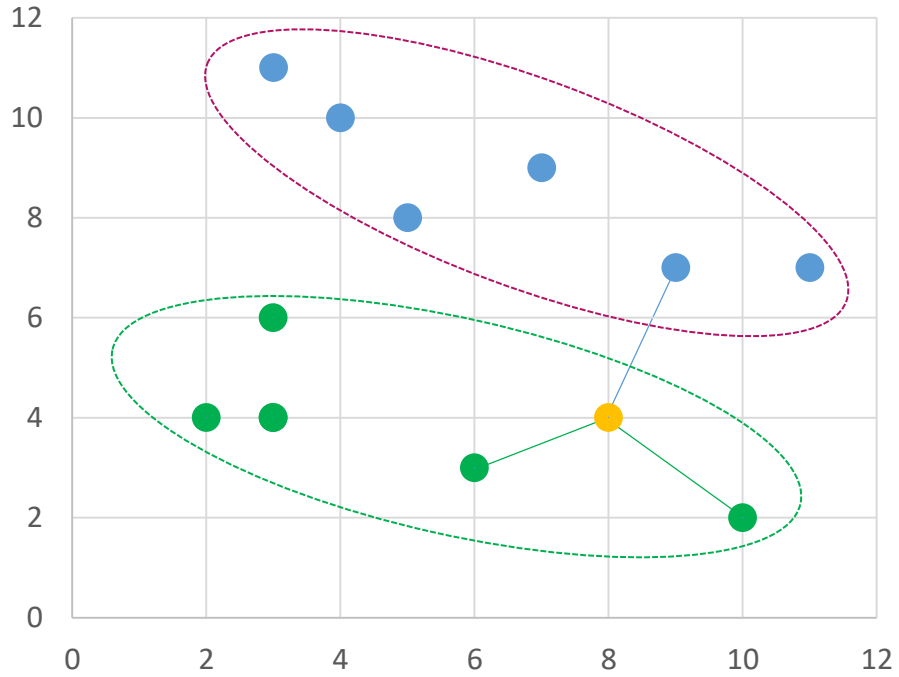
Bu el yalın 3 komşu problemi

(2,4) ile test edilecek (8,4) noktaları arasındaki Öklid uzaklığının değeri

$$d((2,4)(8,4)) = \sqrt{(2-8)^2 + (4-4)^2} = \mathbf{6,00}$$

x1	x2	Küme	Uzaklık	3-En Yakın Komşu
2	4	A	6,00	
3	6	A	5,39	
3	4	A	5,00	
4	10	B	7,21	
5	8	B	5,00	
6	3	A	2,24	1
7	9	B	5,10	
9	7	B	3,16	3
11	7	B	4,24	
10	2	A	2,83	2
3	11	B	8,60	

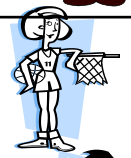
Yeni nokta (8,4) için en yakın 3 komşu



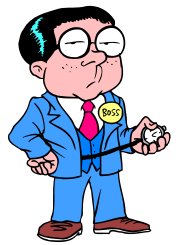
k-En Yakın Komşu Yöntemi

Müşteri	Yaş	Gelir	Kredi kartı sayısı	Sınıf
Fırat 	35	35bin	3	A
Necla 	22	50bin	2	B
Gönül 	63	200bin	1	A
Ayhan 	59	170bin	1	B
Neslihan 	25	40bin	4	B
Erdal 	37	45bin	2	?

k-En Yakın Komşu Yöntemi

Müşteri	Yaş	Gelir (bin)	Kredi Kartı sayısı	Sınıf	Erdal'dan uzaklık
Fırat	35	 35	3	A	sqrt $[(35-37)^2+(35-45)^2+(3-2)^2]=10.25$
Necla	22	 50	2	B	sqrt $[(22-37)^2+(50-45)^2+(2-2)^2]=15.81$
Gönül	63	 200	1	A	sqrt $[(63-37)^2+(200-45)^2+(1-2)^2]=157.17$
Ayhan	59	 170	1	B	sqrt $[(59-37)^2+(170-45)^2+(1-2)^2]=126.92$
Neslihan	25	 40	4	B	sqrt $[(25-37)^2+(40-45)^2+(4-2)^2]=13.15$
Erdal	37	 45	2	**B**	

Örneklerin Normalizasyonu



Fırat:
yaş=35
gelir=150bin
Kredi kartı sayısı=3

Necla:
yaş=22
gelir=215bin
Kredi kartı sayısı=2

olsaydı

Distance (Fırat, Necla)=sqrt $[(35-22)^2$**$+(150{,}000-215{,}000)^2$** $+(3-2)^2]$

- Komşular arasındaki mesafeyi hesaplarken bazı özellikler baskın olmaktadır. Örneğimizde gelir özelliğinin olduğu gibi. Bu tip değerlerin normalize edilmesi önemlidir.
- Örnek: gelir

  yüksek gelir = 500bin

  Fırat'ın geliri 150/500, Necla'nın geliri de 215/500 olarak normalize edilebilir.

Örneklerin Normalizasyonu

$$a_i = \frac{X_i - \min X_i}{\max X_i - \min X_i}$$

$$a_i = \frac{X_i - X_{ort}}{\max X_i - \min X_i}$$

a_i : Normalize edilmiş değer

X_i : Veri değeri

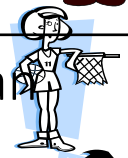
X_{ort} : Verilerin Ortalaması

min X_i : Veri içerisindeki min değer

max Xi : Veri içerisindeki max değer

Örneklerin Normalizasyonu

$$a_i = \frac{X_i - \min X_i}{\max X_i - \min X_i}$$

Müşteri	Yaş	Gelir (bin)	Kredi Kartı sayısı	Sınıf
 Fırat	(35-22)/(63-22)=0.317	(35-35)/(200-35)=0	(3-1)/(4-1)=0.66	A
 Necla	(22-22)/(63-22)=0	(50-35)/200-(35)=0.090	(2-1)/(4-1)=0.333	B
 Gönül	(63-22)/(63-22)=1	(200-35)/(200-35)=1	(1-1)/(4-1)=0	A
 Ayhan	(59-22)/(63-22)=0.902	(170-35)/(200-35)=0.818	(1-1)/(4-1)=0	B
 Neslihan	(25-22)/(63-22)=0.073	(40-35)/(200-35)=0.030	(4-1)/(4-1)=1	B
 Erdal	(37-22)/(63-22)=0.365	(45-35)/(200-35)=0.030	(2-1)/(4-1)=0.333	**B**

Z-Skor tekniği ile Örneklerin Normalizasyonu

- Öznitelik değerlerinin normalizasyonu çok çeşitli şekillerle yapılmaktadır.
- Bunlardan biri de **Z-Skor** tekniğidir.

- Z-Skor normalizasyonu, her bir öznitelik değerinden, ortalamanın farkının alınması ve elde edilen farkın standart sapmaya bölünmesidir.

- Böylece ham veriler standart verilere dönüştürülerek, ölçü birimi farklılığı ortadan kaldırılmış olur.

- Normalize edilmiş değerler ile öznitelikler arası karşılaştırma daha kolaydır.

Z-Skor tekniği ile Örneklerin Normalizasyonu

Z-Skor Formülü :

$$Z_i = \frac{(X_i - X_{ort})}{S}$$

Z_i : Z-Skor

X_i : Veri Değeri

X_{ort} : Verilerin Ortalaması $X_{ort} = \frac{\sum X_i}{n}$

S : Standart Sapma $S = \sqrt{\frac{\sum (X_i - X_{ort})^2}{n}}$

Z-Skor Standart Sapma ile çarpılıp, ortalama ile toplanırsa ham değerler elde edilir.

Örneklerin Normalizasyonu

Uzaklık (distance) normalde nümerik değerler kullanılarak hesaplanır

$d = \text{sqrt}\,[(35-37)^2+(35-45)^2+(3-2)^2]=\mathbf{10.25}$

Kategorik(Nominal) bir öznitelik bulunursa?

Örnek: Evli

Kategorik örnekler söz konusu olduğunda, **Hamming mesafesi** kullanılmalıdır.

Ayrıca, veri kümesinde sayısal ve kategorik örneklerin bir karışımı olduğunda değerlerin normalizasyonu meselesi tekrar ortaya çıkmaktadır.

Müşteri	Evli	Gelir (bin)	Kredi Kartı sayısı	Sınıf
Fırat	evet	35	3	A
Necla	hayır	50	2	B
Gönül	hayır	200	1	A
Ayhan	evet	170	1	B
Neslihan	hayır	40	4	B
Erdal	evet	45	2	

Hamming mesafesi

- Hamming mesafesi aynı uzunluktaki iki string arasında, birbirine dönüşmesi için gerekli olan yer değiştirme sayısını verir.
- Yani basitçe bir stringin diğer stringden ne kadar farklı olduğunu gösterir.

Male <-> Male -> Hamming mesafesi = 0

Male <-> Fmale -> Hamming mesafesi = 1

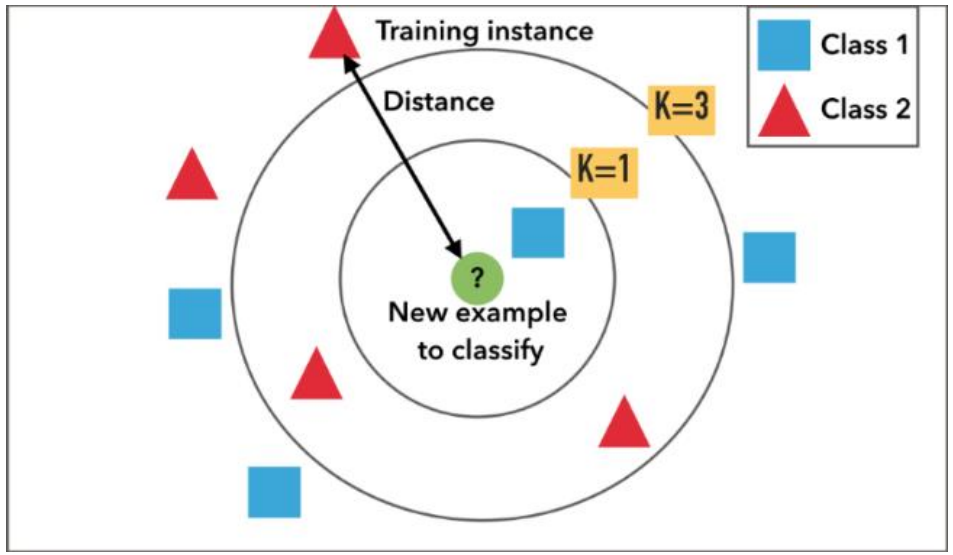
düğün <-> düşün -> Hamming mesafesi = 1

00111 <-> 11001 -> Hamming mesafesi = 4

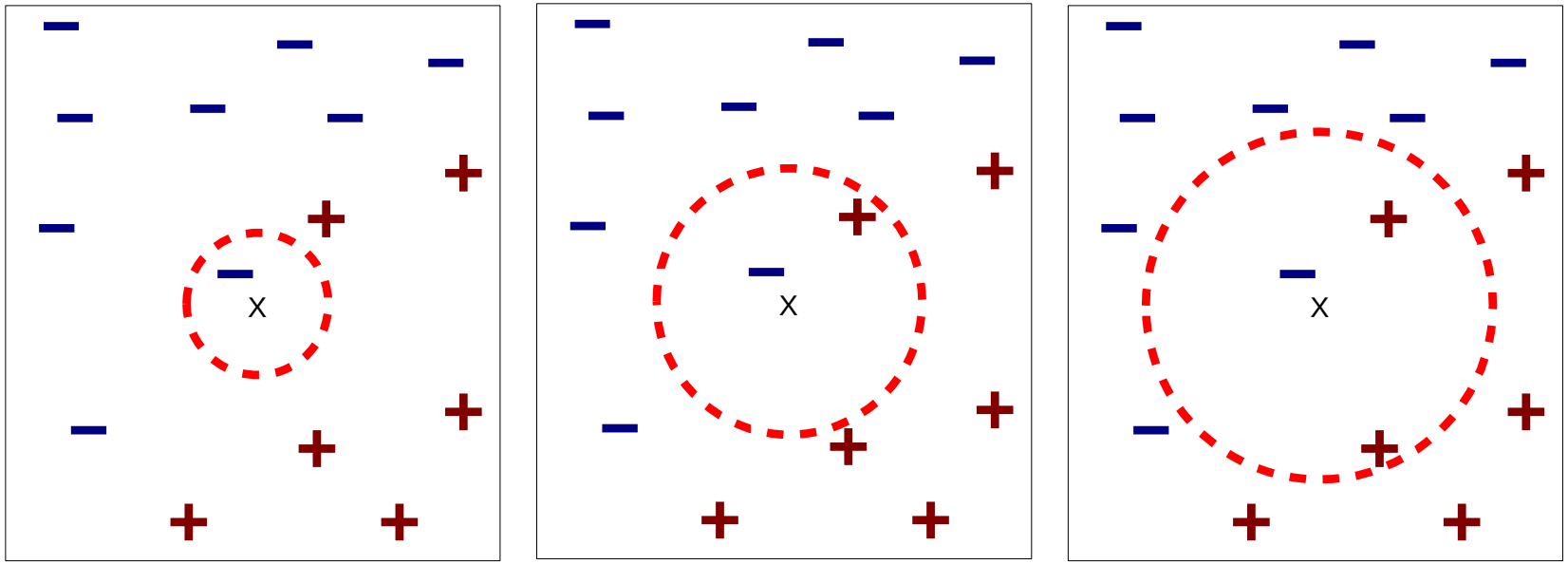
0122 <-> 1220 -> Hamming mesafesi = 3

Komşu Sayısı

- K-NN algoritmasında, komşu sayısı (k) parametresinin değerine dayalı olarak sınıflandırma yapılmaktadır. Sınıflandırma, doğru k seçimine duyarlıdır.
- Sınıflandırma sürecinde k=1 için, sadece en yakın komşunun bulunduğu sınıfa atanırken,
- k sayısı örnek sayısına (N) yaklaştıkça veri setinde yer alan tüm veriler dikkate alınmakta ve oylamaya göre seçim yapılmaktadır.
- Çoğu veri kümesi için optimal K, 3-10 arasında seçilebilir.



Komşu Sayısı



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

Örnek x'in en yakın K-komşuları, x'e en küçük mesafeye sahip veri noktalarıdır.

Oylama

- Uzaklık fonksiyonunu kullanarak test verisinin en yakın komşularını elde ettikten sonra, test verisinin sınıfını öngörmek için komşuların oylamasına başvurulur.
- Bunun için iki yaklaşım yaygındır.

- **Çoğunluk oyu (Majority voting):** Bu yaklaşımda, tüm oylar eşittir. En çok oyu olan sınıfa atama yapılır.
- **Ağırlıklı oylama (Weighted voting) :** Bu yaklaşımda, daha yakın komşular daha yüksek oy alır.

Ağırlıklı oylama (Weighted voting)

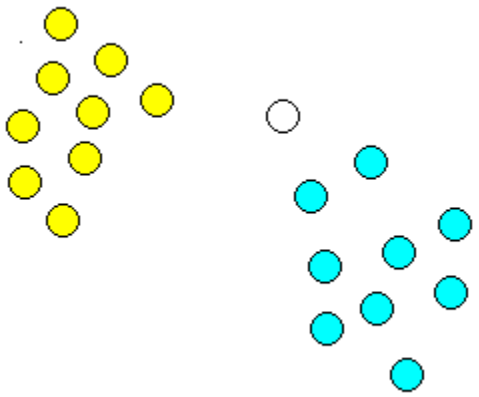
- Komşular için ağırlık değerleri atanması ile sınıflandırılmakta olan örneğe yakın olan komşuların, daha uzak olanlara göre çoğunluk oylamasında daha fazla katkıda bulunmaları amaçlanır.

- En çok kullanılan ağırlık değeri atama yöntemleri, her bir komşunun ağırlığının, $1/d$ ya da $1/d^2$ şeklinde alınmasıdır. (d, komşuya olan uzaklıktır)

- $w_i = 1/d$ veya $1/d^2$

K-En Yakın Komşu (K-Nearest Neighbor)

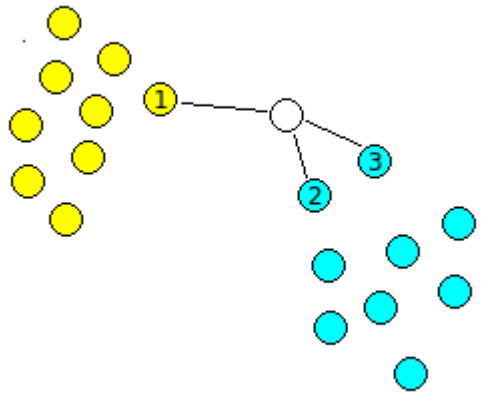
- Aşağıdaki gibi bir veri kümesinin olduğunu düşünelim. Bu veri kümesindeki örnekler sarı ve mavi olmak üzere iki sınıfa ayrılmaktadır.
- Sınıfı bilinmeyen bir örnek verildiği zaman bu örneği bir sınıfa atamak için K-NN sınıflandırma metodu kullanılabilir.



K-En Yakın Komşu (K-Nearest Neighbor)

- Sınıfı bilinmeyen örneğin K adet en yakın komşusuna bakılır.

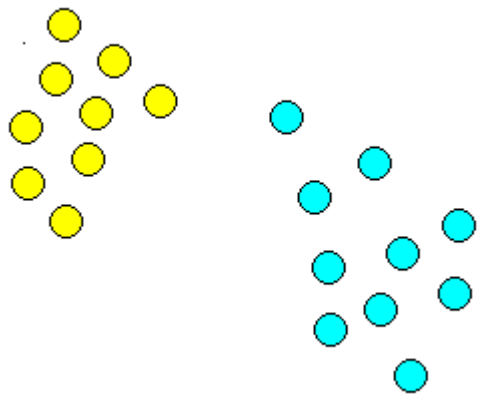
- Örneğin, K=3 seçerek en yakın 3 komşuya bakılmalıdır.
- Bunun için bütün veri kümesindeki bütün örneklerin sınıfı bilinmeyen örneğe olan uzaklığına hesaplanıp bunlardan en küçük olan 3 tanesi seçilir.



K-En Yakın Komşu (K-Nearest Neighbor)

- Bir örnek, komşularının çoğunluk oyuyla sınıflandırılır. Dolaysıyla bulunan en yakın K=3 komşunun sınıflarına bakarak sınıfı bilinmeyen örnek çoğunluk olan sınıfa atanır.

- En yakın 3 komşudan 2 tanesinin mavi 1 tanesinin sarı sınıfına ait olduğunu varsayarsak sınıfı bilinmeyen örnek çoğunluk olan mavi sınıfına atanır.



Örnek-1

Aşağıdaki tablo X, Y öznitelik değerlerinden ve Z sınıf değerlerinden oluşmaktadır. Bu örnek değerleriyle yola çıkarak yeni verilen test örnek değerinin hangi sınıfa ait olduğunu k-en yakın komşu yöntemiyle bulalım.

Test için verilen örnek değeri X=7, Y=3;

k= 4 için işlem yapalım.

(Problem için (7,3) noktasına en yakın komşu değerler aranmalı.)

X	Y	Z
1	3	Negatif
2	5	Pozitif
2	3	Pozitif
3	9	Negatif
4	7	Negatif
5	2	Pozitif
6	8	Pozitif
8	6	Negatif
10	6	Negatif
11	1	Negatif

Örnek-1

X	Y	Z
1	3	Negatif
2	5	Pozitif
2	3	Pozitif
3	9	Negatif
4	7	Negatif
5	2	Pozitif
6	8	Pozitif
8	6	Negatif
10	6	Negatif
11	1	Negatif

Öklit bağıntısına göre her bir örnek değeri için uzaklıklar hesaplanmalı.

(7, 3) noktasının tüm örnek değerleri ile arasındaki uzaklıkları hesaplanırsa.

$$d(i, j) = \sqrt{(1-7)^2 + (3-3)^2} = 6.00$$

$$d(i, j) = \sqrt{(2-7)^2 + (5-3)^2} = 5{,}39$$

$$d(i, j) = \sqrt{(2-7)^2 + (3-3)^2} = 5{,}00$$

...

Örnek-1

Örnek değerlerlerin (7, 3) noktasına olan uzaklığı…

X	Y	Uzaklık
1	3	6
2	5	5,39
2	3	5
3	9	7,21
4	7	5
5	2	2,24
6	8	5,10
8	6	3,16
10	6	4,24
9	1	2,83

En küçük uzaklıkların belirlenmesi için satırlar sıralanarak en küçük **k=4** tanesi belirleniyor.

Belirlenen dört nokta **(7, 3)** noktasına en yakın değerlerdir.

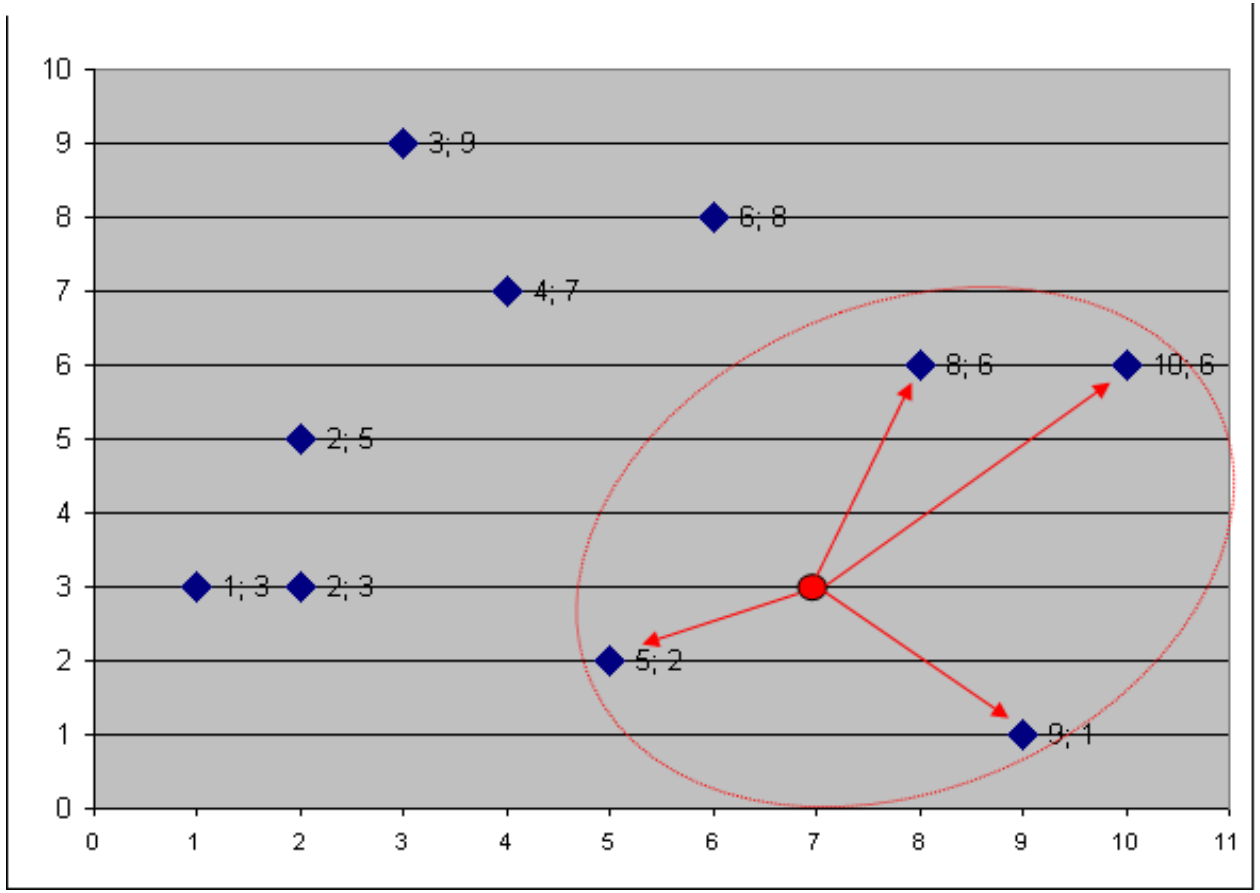
Uzaklık değerlerine göre k=4 komşu değerlerin belirlenmesi

X	Y	Uzaklık	Sıralama
1	3	6	9
2	5	5,39	8
2	3	5	6
3	9	7,21	10
4	7	5	5
5	2	2,24	1
6	8	5,10	7
8	6	3,16	3
10	6	4,24	4
9	1	2,83	2

Örnek-1

X	Y	Uzaklık	Sıralama
1	3	6	9
2	5	5,39	8
2	3	5	6
3	9	7,21	10
4	7	5	5
5	2	2,24	1
6	8	5,10	7
8	6	3,16	3
10	6	4,24	4
9	1	2,83	2

(7, 3) Noktasına komşu olan en yakın dört örnek değerinin koordinat sistemi üzerindeki gösterimi



Örnek-1

En küçük satırlara ilişkin sınıfların belirlenmesi işlemi örnek değerlerinin içinde **hangi değerin baskın olduğuna göre** karar verilir.

X	Y	Z
1	3	Negatif
2	5	Pozitif
2	3	Pozitif
3	9	Negatif
4	7	Negatif
5	2	Pozitif
6	8	Pozitif
8	6	Negatif
10	6	Negatif
9	1	Negatif

Örnek değerlerin içinde **bir pozitif** ve **üç negatif** değer olduğundan **(7, 3)** noktasının sınıfı **negatif** olarak belirlenir.

(7,3) noktasının Sınıfı Negatif olarak belirlenir.

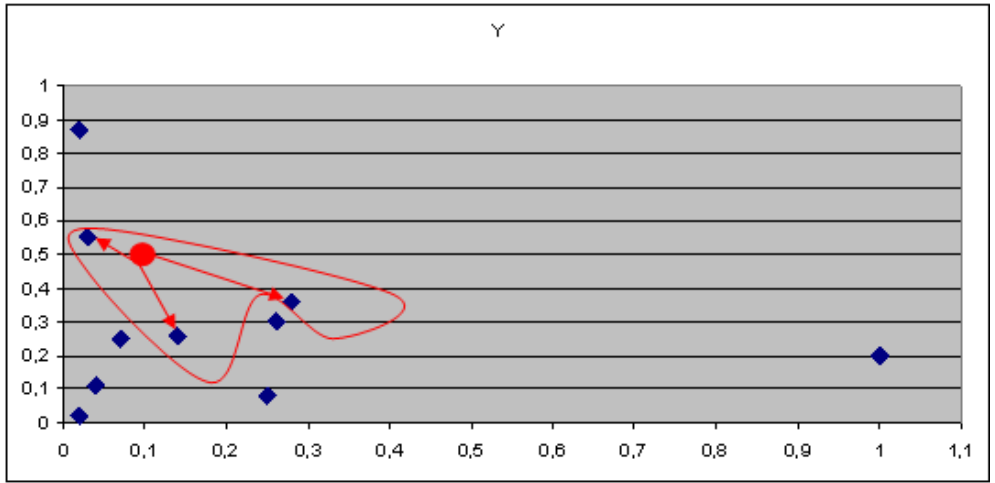
Örnek-2

X= 0,10 Y=0,50 olarak verilen test örnek değerlerinin hangi sınıfa dahil olduğunu k-en yakın komşu algoritmasından Ağırlıklı Oylama Yöntemini uygulayarak bulunuz. k=3 olarak seçilecektir.

X	Y	BAKİYE
0,07	0,25	ARTI
0,02	0,02	ARTI
0,25	0,08	ARTI
1	0,2	EKSİ
0,26	0,3	ARTI
0,14	0,26	ARTI
0,28	0,36	ARTI
0,04	0,11	EKSİ
0,03	0,55	ARTI
0,02	0,87	EKSİ

Örnek-2

X	Y	UZAKLIK	SIRA
0,07	0,25	0,25	4
0,02	0,02	0,49	9
0,25	0,08	0,45	8
1	0,2	0,95	10
0,26	0,3	0,26	5
0,14	0,26	0,24	3
0,28	0,36	0,23	2
0,04	0,11	0,39	7
0,03	0,55	0,09	1
0,02	0,87	0,38	6



X	Y	UZAKLIK	SIRA	BAKIYE
0,07	0,25	0,25	4	ARTI
0,02	0,02	0,49	9	ARTI
0,25	0,08	0,45	8	ARTI
1	0,2	0,95	10	EKSİ
0,26	0,3	0,26	5	ARTI
0,14	0,26	0,24	3	ARTI
0,28	0,36	0,23	2	ARTI
0,04	0,11	0,39	7	EKSİ
0,03	0,55	0,09	1	ARTI
0,02	0,87	0,38	6	EKSİ

En küçük uzaklıkların belirlenmesi

k=3 olarak seçilen örneklerin belirlenmesi

Örnek-2

Ağırlıklı Oylama Yöntemin Uygulanması :

$$d(i,j)' = \frac{1}{d(i,j)^2}$$

$$d(9,örnek1)' = \frac{1}{(0,09)^2} = 135,14$$

$$d(7,örnek2)' = \frac{1}{(0,23)^2} = 19,23$$

$$d(6,örnek3)' = \frac{1}{(0,24)^2} = 16,89$$

X	Y	UZAKLIK	AGIRLIKLI OYLAMA	SIRA
0,07	0,25	0,25	15,77	4
0,02	0,02	0,49	4,22	9
0,25	0,08	0,45	5,03	8
1	0,2	0,95	1,11	10
0,26	0,3	0,26	15,24	5
0,14	0,26	0,24	16,89	3
0,28	0,36	0,23	19,23	2
0,04	0,11	0,39	6,42	7
0,03	0,55	0,09	135,14	1
0,02	0,87	0,38	6,98	6

Örnek-2

$w_i = 1/d^2$

$W4 = 1/(0,25)^2 = 15,77$
$W9 = 1/(0,49)^2 = 4,22$
$W9 = 1/(0,09)^2 = 135,14$

.....

Ağırlıklı uzaklık değerleri tablo üzerinde gösterilirse

X	Y	UZAKLIK	AGIRLIKLI OYLAMA	SIRA	BAKIYE
0,07	0,25	0,25	15,77	4	ARTI
0,02	0,02	0,49	4,22	9	ARTI
0,25	0,08	0,45	5,03	8	ARTI
1	0,2	0,95	1,11	10	EKSİ
0,26	0,3	0,26	15,24	5	ARTI
0,14	0,26	0,24	16,89	3	ARTI
0,28	0,36	0,23	19,23	2	ARTI
0,04	0,11	0,39	6,42	7	EKSİ
0,03	0,55	0,09	135,14	1	ARTI
0,02	0,87	0,38	6,98	6	EKSİ

Bakiyeler içinde hepsi ARTI olduğu için aranan test değerinin sınıfının da **ARTI**'ya ait olduğu belirlenir.

Örnek-3

Test örneği:

height =161cm weight = 61kg

ise hangi sınıfa dahil olur?

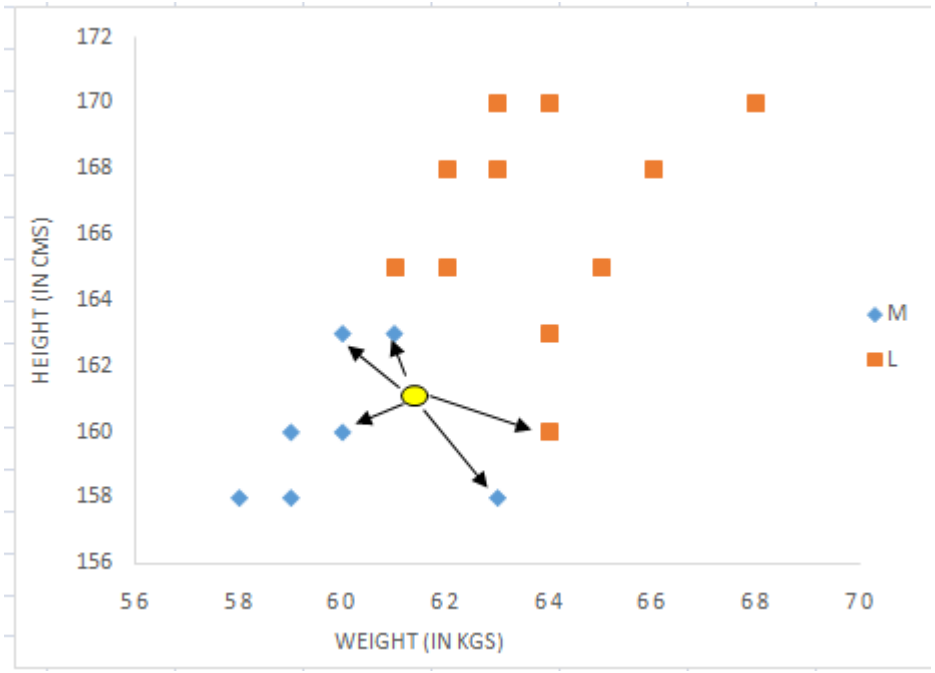
Height (in cms)	Weight (in kgs)	T Shirt Size
158	58	M
158	59	M
158	63	M
160	59	M
160	60	M
163	60	M
163	61	M
160	64	L
163	64	L
165	61	L
165	62	L
165	65	L
168	62	L
168	63	L
168	66	L
170	63	L
170	64	L
170	68	L

Örnek-3

k=5 olsun. Bu durumda algoritma test örneğine en yakın 5 örneği arar, yani öznitelik bakımından test örneğine en çok benzeyen bu 5 örneğin hangi kategoriye ait olduğunu görür.

Bu en yakın 5 komşunun 4'ünde 'Medium (M)' T shirt boyutu ve 1 tanesinde ise ‘Large (L)' T shirt boyutu vardır.

O zaman test örneği için en iyi tahmin 'Medium (M)' .



=SQRT((A21-A6)^2+(B21-B6)^2)

	A	B	C	D	E
1	Height (in cms)	Weight (in kgs)	T Shirt Size	Distance	
2	158	58	M	4.2	
3	158	59	M	3.6	
4	158	63	M	3.6	
5	160	59	**M**	2.2	**3**
6	160	60	**M**	1.4	**1**
7	163	60	**M**	2.2	**3**
8	163	61	**M**	2.0	**2**
9	160	64	**L**	3.2	**5**
10	163	64	L	3.6	
11	165	61	L	4.0	
12	165	62	L	4.1	
13	165	65	L	5.7	
14	168	62	L	7.1	
15	168	63	L	7.3	
16	168	66	L	8.6	
17	170	63	L	9.2	
18	170	64	L	9.5	
19	170	68	L	11.4	
20					
21	**161**	**61**			

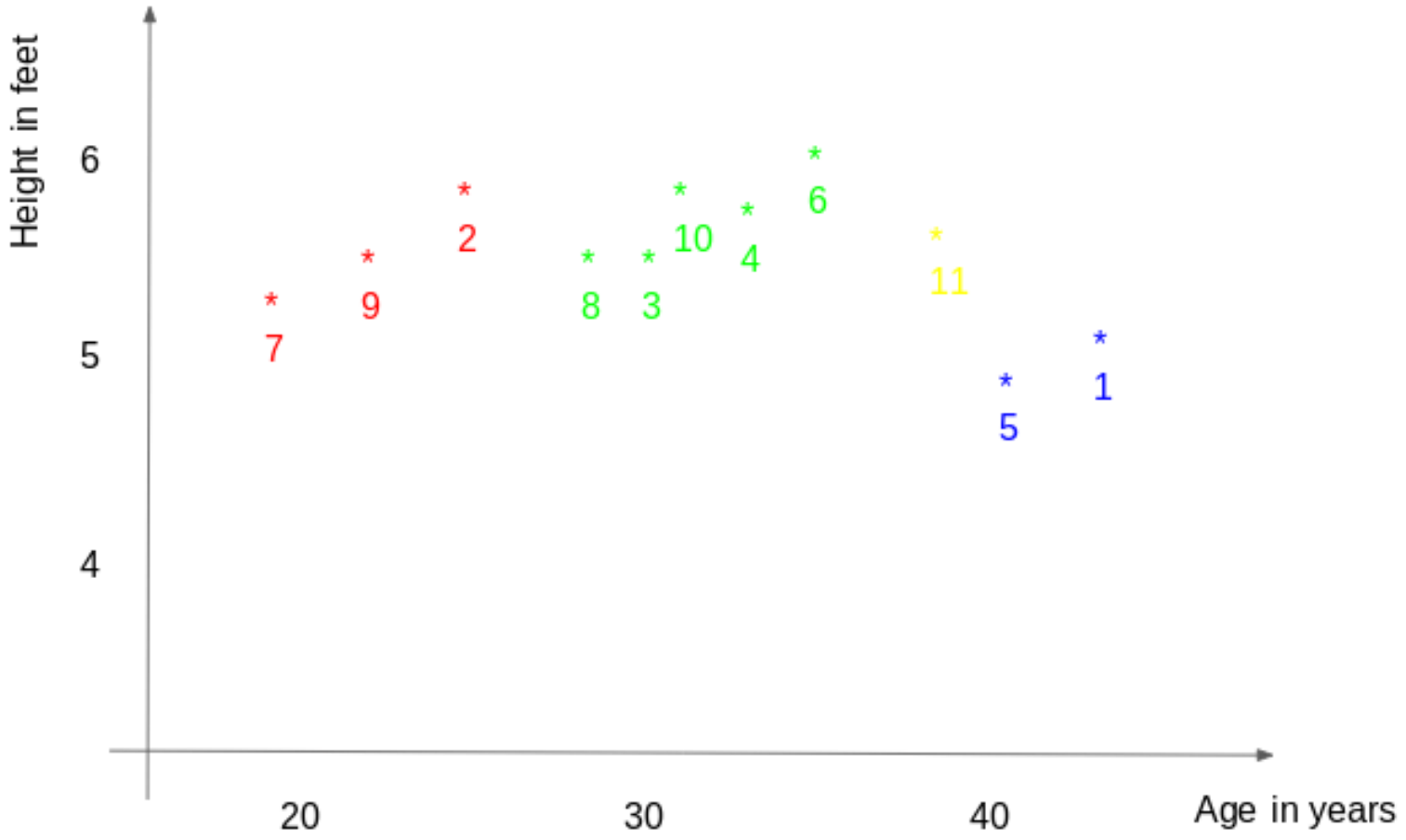
Örnek-4

- Test örneği ID 11 için: height =5,5 (feet) age = 38 (yıl) weight=? (kg)

ID	Height	Age	Weight
1	5	45	77
2	5.11	26	47
3	5.6	30	55
4	5.9	34	59
5	4.8	40	72
6	5.8	36	60
7	5.3	19	40
8	5.8	28	60
9	5.5	23	45
10	5.6	32	58
11	5.5	38	?

Örnek-4

- Aşağıdaki grafikte, y ekseni bir kişinin yüksekliğini (feet olarak) ve x ekseni yaşı (yıl olarak) gösterir. Noktalar, ID değerlerine göre numaralandırılmıştır. Sarı nokta (ID 11) test noktasıdır.



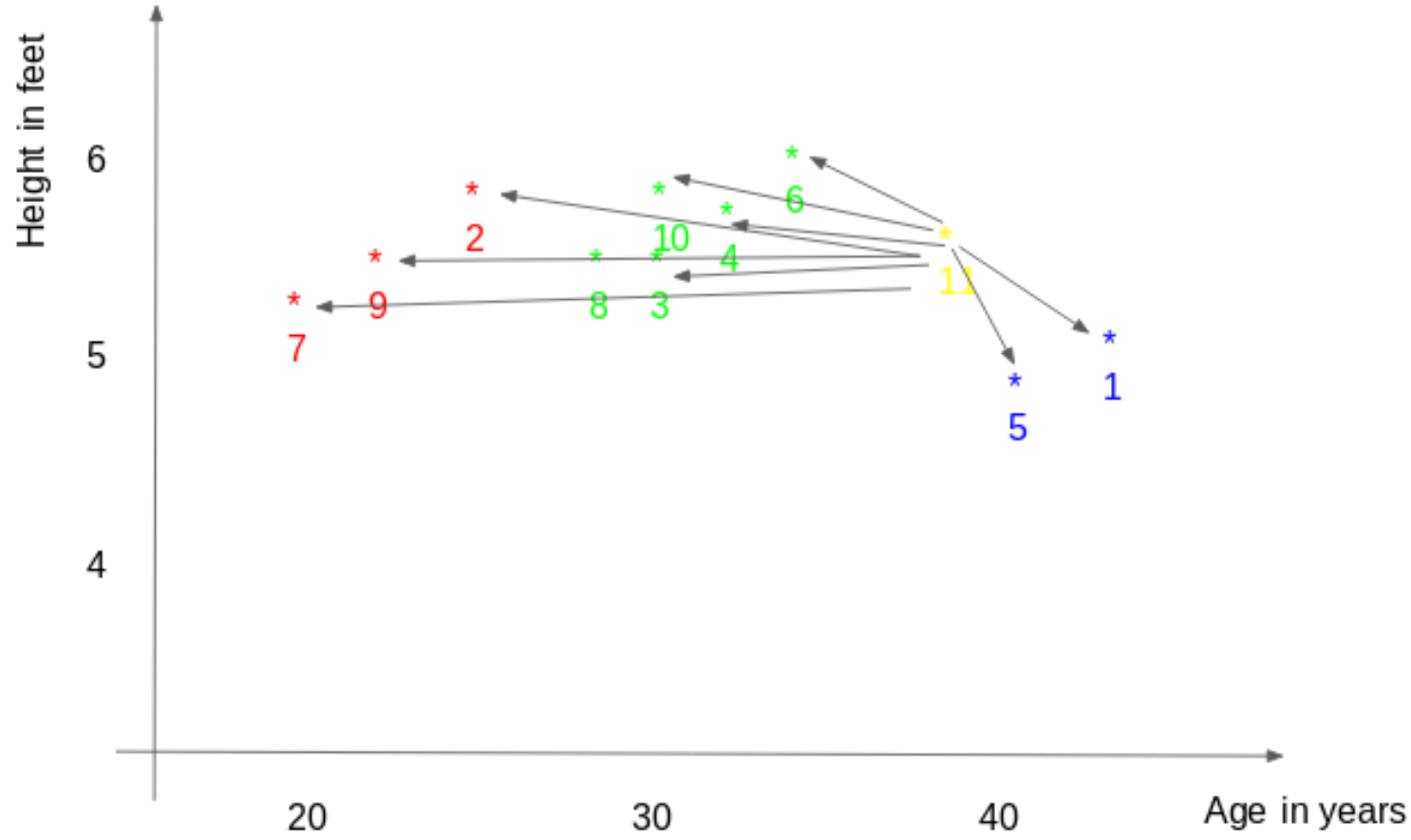
Örnek-4

- ID11'in ağırlığının belirlenmesi isteniyor. ID11'in 5. ve 1. noktalara daha yakın olması nedeniyle, bu ID'lere benzer bir ağırlığa sahip olması gerektiği, muhtemelen 72-77 kg (tablodaki ID1 ve ID5 ağırlıkları) arasında olması gerektiği düşünülebilir.

- Bu aslında mantıklı geliyor, ancak algoritmanın değerleri nasıl öngördüğünü ayrıca hesaplayalım.

Örnek-4

- İlk önce, test örneği ile veri kümesindeki her örnek arasındaki mesafe hesaplanır.

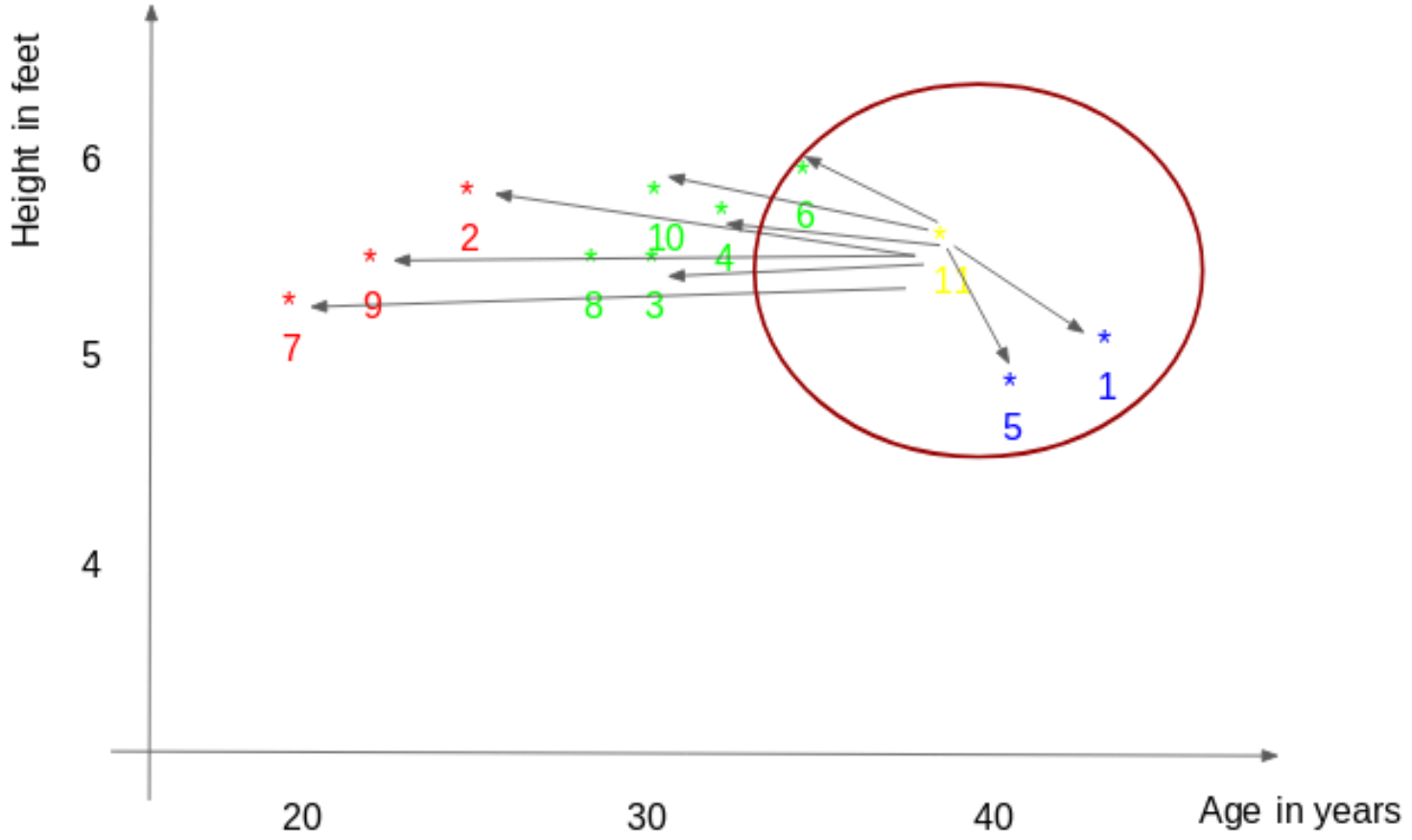


Örnek-4

- k=3 olsun. Bu durumda algoritma test örneğine en yakın 3 örneği arar. Bu örnekte, k değeri 3 için 1, 5, 6 numaralı noktalar seçilecektir.

ID	Height	Age	Weight
1	5	45	77
5	4.8	40	72
6	5.8	36	60

1, 5, 6 veri noktalarının ortalaması, 11. nokta için öngörüdür.
Burada
ID11 = (77 + 72 + 60) / 3
ID11 = 69,66 kg olur.

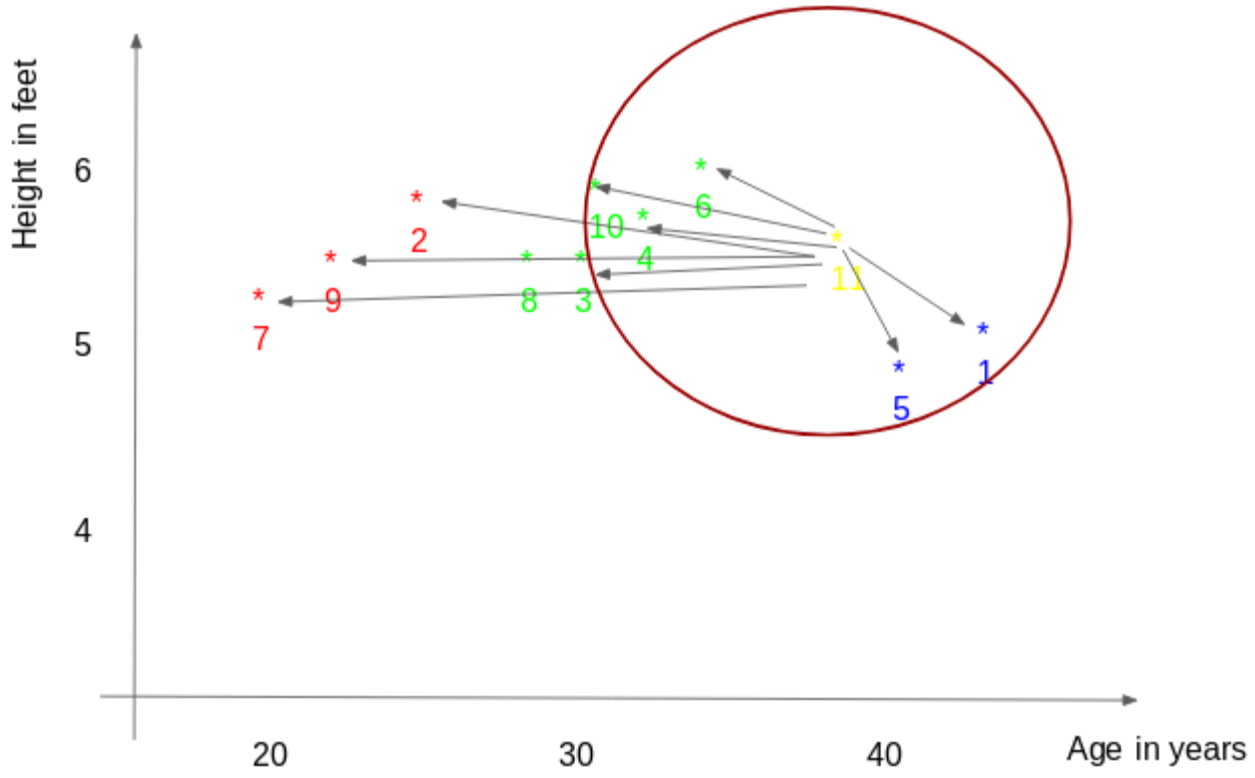


Örnek-4

- Eğer k = 5 seçilirse, en yakın noktalar ID1, ID4, ID5, ID6, ID10 olacaktır.

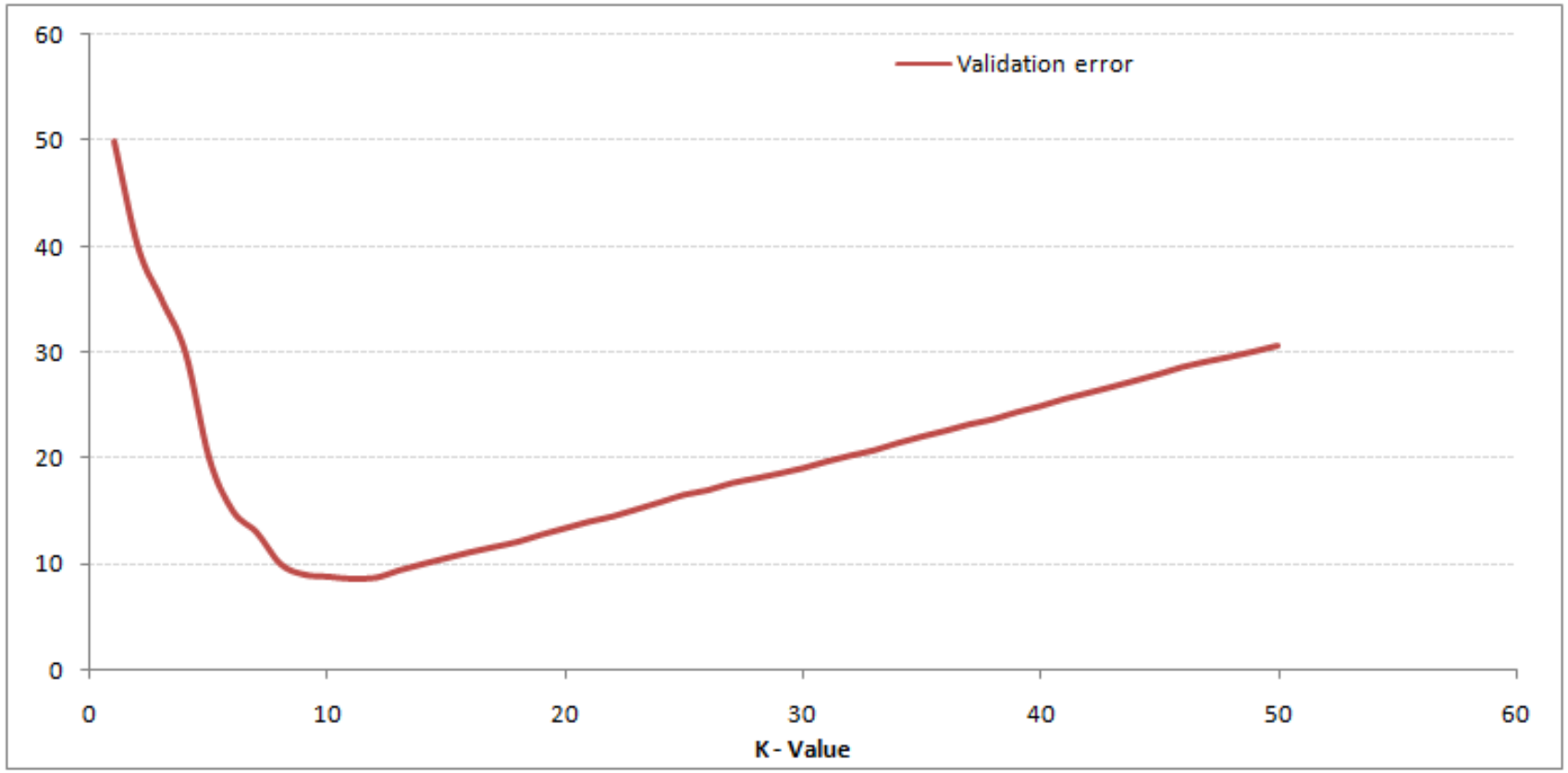
ID	Height	Age	Weight
1	5	45	77
4	5.9	34	59
5	4.8	40	72
6	5.8	36	60
10	5.6	32	58

- ID 11 = (77+59+72+60+58)/5
- ID 11 = 65,2 kg



Örnek-4

- k'nın farklı değerleri için test hatası grafikte gösterilmiştir. Yüksek bir k değeri için model test setinde düşük performans gösterir. Test hatası eğrisi k = 9 değerinde bir minimuma ulaşır. Bu k değeri modelin optimum değeridir (farklı veri setleri için değişecektir). Bu eğri ‘dirsek eğrisi‘ (elbow curve) olarak bilinir ve genellikle k değerini belirlemek için kullanılır.



SONUÇ

- KNN metodunun test aşaması, zaman ve bellek açısından yavaş ve maliyetlidir.
- Öngörü için tüm eğitim veri kümesini saklamak için geniş bellek gerektirir.
- KNN en yakın komşuları bulmak için iki veri noktası arasında Euclidean uzaklığını kullandığı zaman verilerin ölçeklendirmesi gerekir. Çünkü Euclidean uzaklığı büyük değerlere duyarlıdır.
- **Daha iyi sonuç elde etmek için, verileri normalleştirmek şiddetle tavsiye edilir.** Genel olarak, normalizasyon aralığı 0 -1 arasında alınır.
- KNN, çok sayıda öznitelikten ziyade daha düşük sayıda öznitelik ile daha iyi performans gösterir.
- Ayrıca KNN büyük boyutlu veriler için çok da uygun değildir.