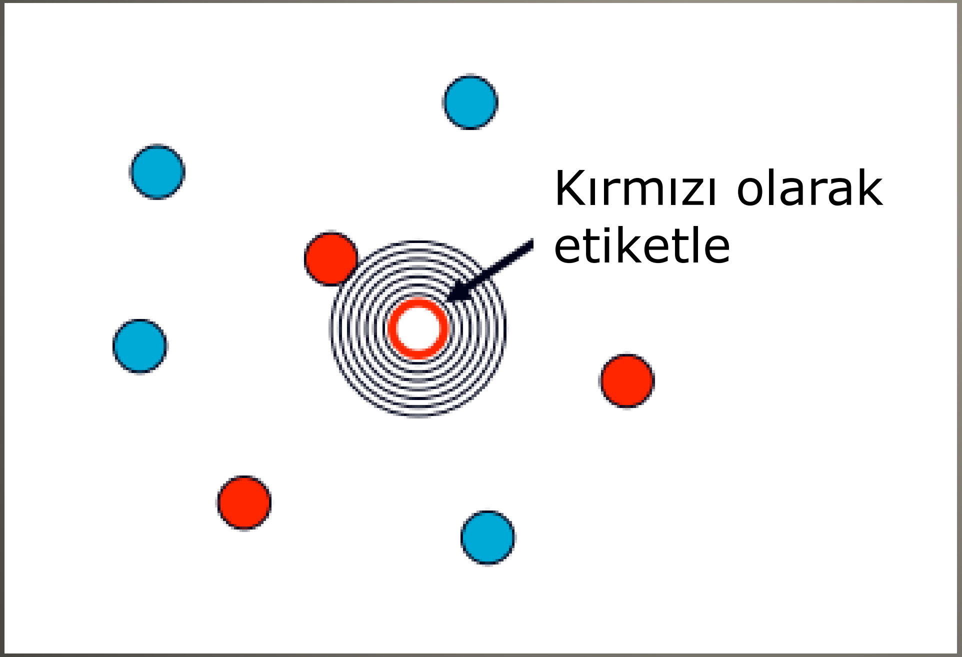
Makine Öğrenmesi

K- En Yakın Komşu Sınıflandırma

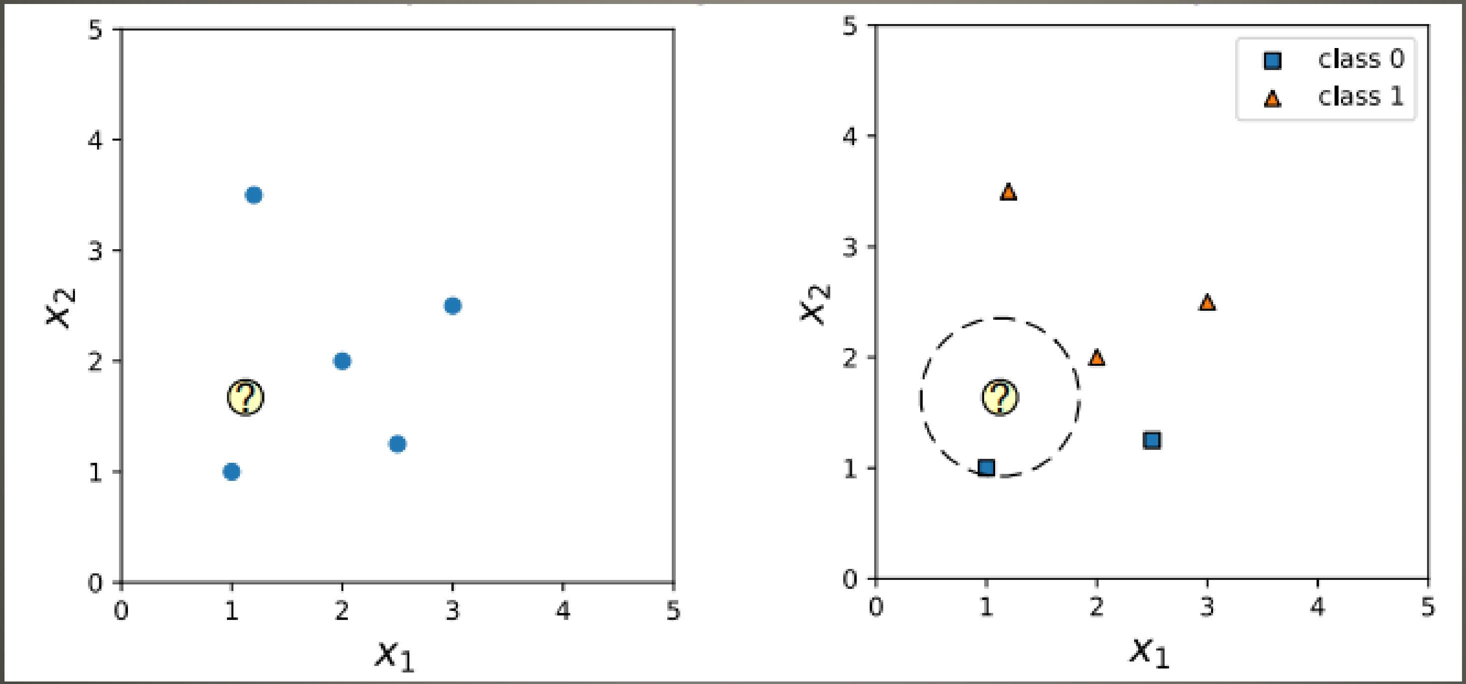
İlhan AYDIN

- Bütün makine öğrenmesi sınıflandırıcılarından en basitidir.
- Temel fikir: yeni bir noktası etiketi bilinen en yakın noktaya göre etiketle.



1- En Yakın Komşu

- Örnek tabanlı öğrenme tipi
  - Aynı zamanda hafıza tabanlı öğrenme olarak bilinir.
  - Görev: Hedefi tahmin et/yeni veri noktasının etiketi



Eğitim kümesindeki en benzer veri noktasına bakınız.

1- En Yakın Komşu

- 1-En Yakın Komşu Eğitim Adımı

Verilen : $\langle x^{[i]}, y^{[i]} \rangle \in \mathcal{D}$ $|\mathcal{D}| = n$

$$\langle x^{[q]}, ? \rangle$$

Tahmin algoritması:f($x^{[q]}$)

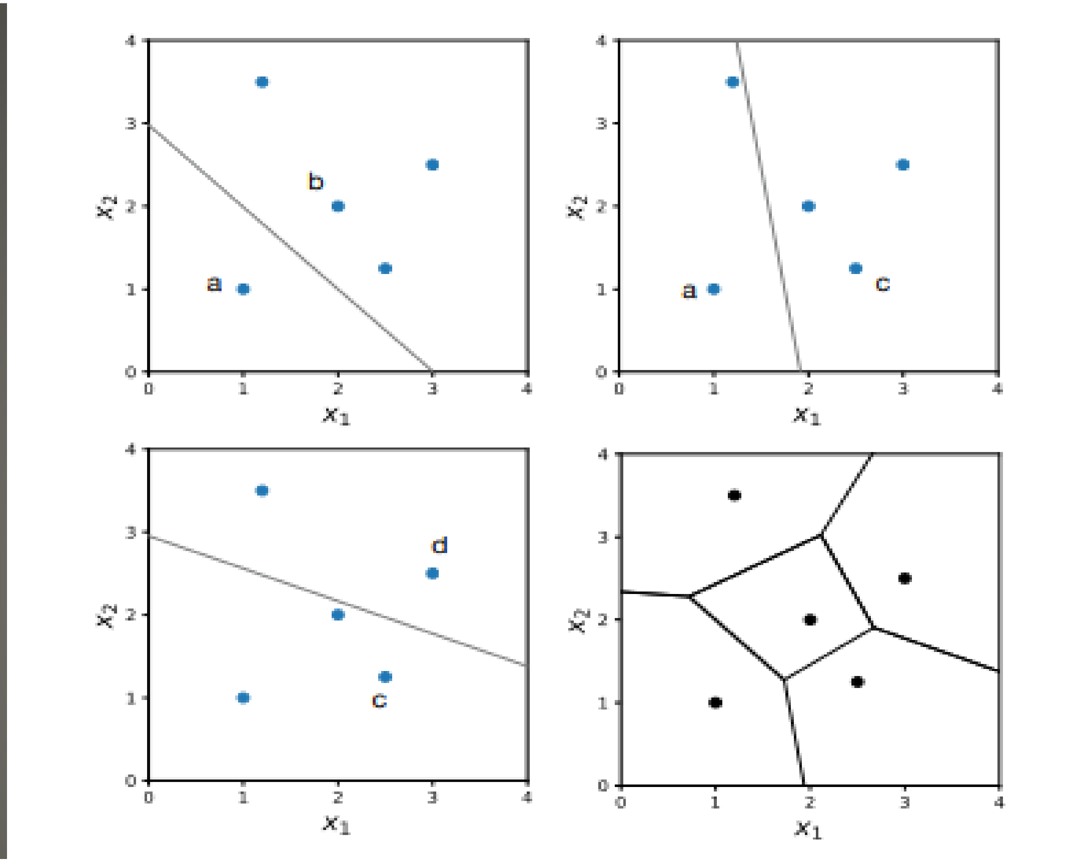
Sorgu noktası

```
Enyakin_nokta=None
enyakin_mesafe=∞
    For i=1…n:
        mevcut_mesafe=d(x^[i], x^[q])
        if mevcut_mesafe<enyakin_mesafe:
            Enyakin_mesafe=mevcut_mesafe
            Enyakin_nokta=x^[i]
    Return f(enyakin_nokta)
```

1- En Yakın Komşu

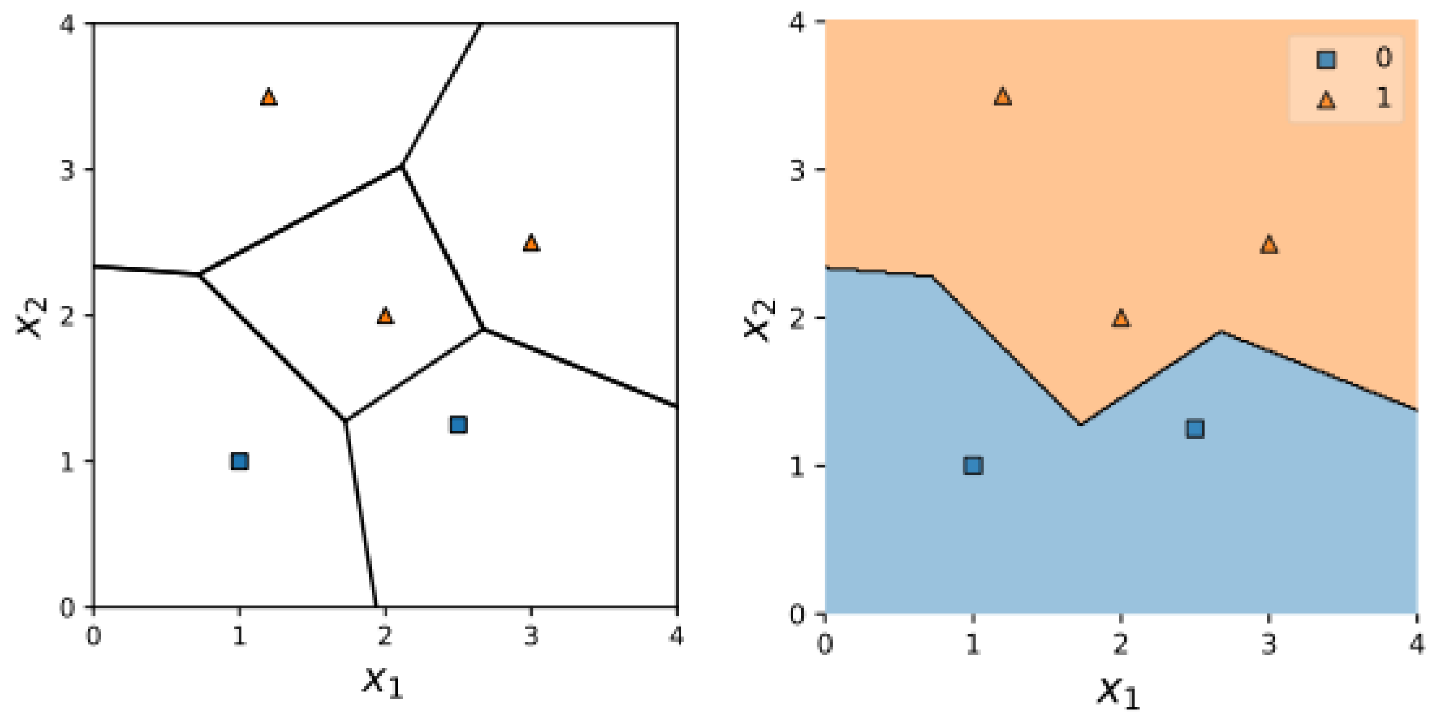
- En yakın komşu sınıflandırma modelinin karar sınırı nasıl oluşturulur?
- Öklid uzaklığı metriğini varsayarsak, herhangi iki eğitim örneği a ve b arasındaki karar sınırı düz bir çizgidir.
- Karar sınırında bir sorgu noktası bulunuyorsa, bunun hem eğitim örneği a hem de b'ye eşit uzaklıkta olduğu anlamına gelir.
- Bir çift nokta arasındaki karar sınırı düz bir çizgi iken, NN modelinin küresel düzeydeki karar sınırı, tüm eğitim seti göz önüne alındığında, birbirine bağlı, dışbükey çokyüzlüler kümesidir.
- Bir çokyüzlü içindeki tüm noktalar, içindeki eğitim örneğine en yakın olanlardır ve çokyüzlü dışındaki tüm noktalar, farklı bir eğitim örneğine daha yakındır.

1- En Yakın Komşu



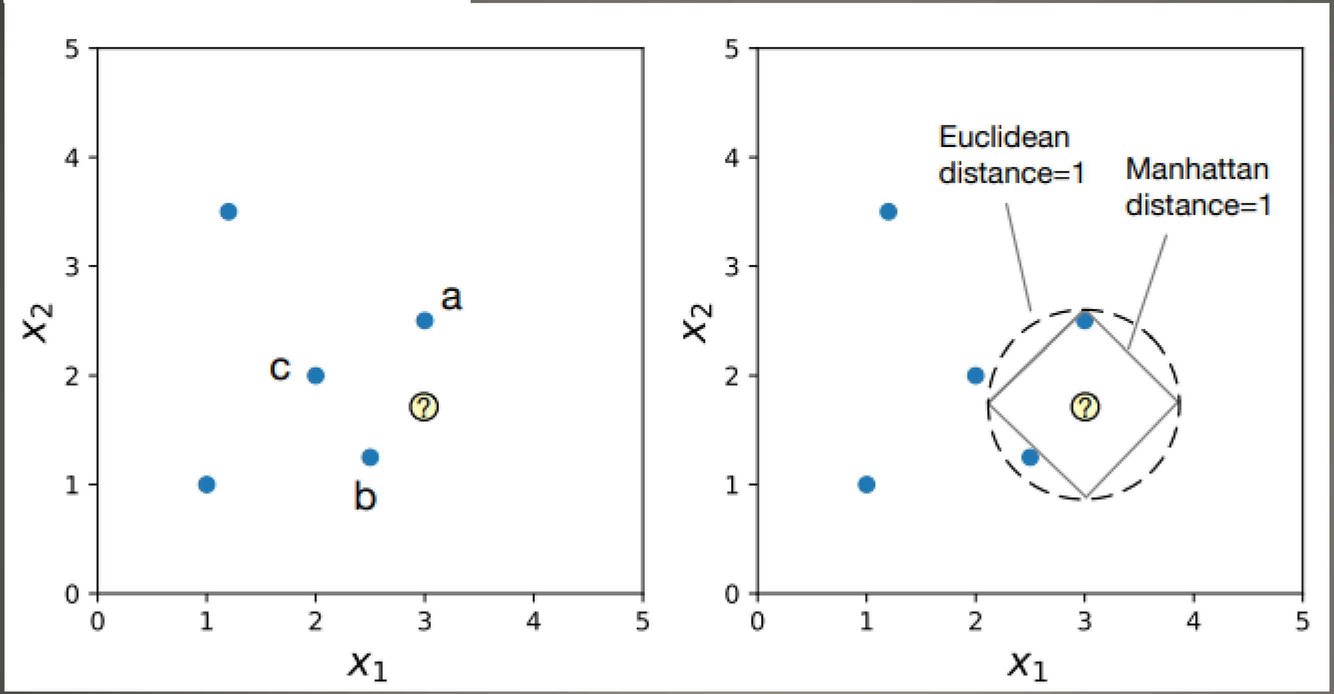
İki eğitim örneği (a & b, a & c ve c & d) ve sonuçta ortaya çıkan Voronoi diyagramı (sağ alt köşe) arasındaki doğrusal segmentler aracılığıyla iki boyutlu bir veri kümesinin (x1 ve x2 özellikleri) düzlem bölümlenmesinin gösterimi.

1- En Yakın Komşu



1- En Yakın Komşu

Hangisi en yakın?



Kullanılan mesafe ölçüm aracına göre değişir?

1- En Yakın Komşu

Euclidean

Manhattan

Minkowski: $d(\mathbf{x}^{[a]}, \mathbf{x}^{[b]}) = \left[\sum_{j=1}^{m} \left(\left| x_j^{[a]} - x_j^{[b]} \right| \right)^p \right]^{\frac{1}{p}}$

Mahalanobis

Cosine similarity

Bazı Ortak Sürekli mesafe Ölçümleri

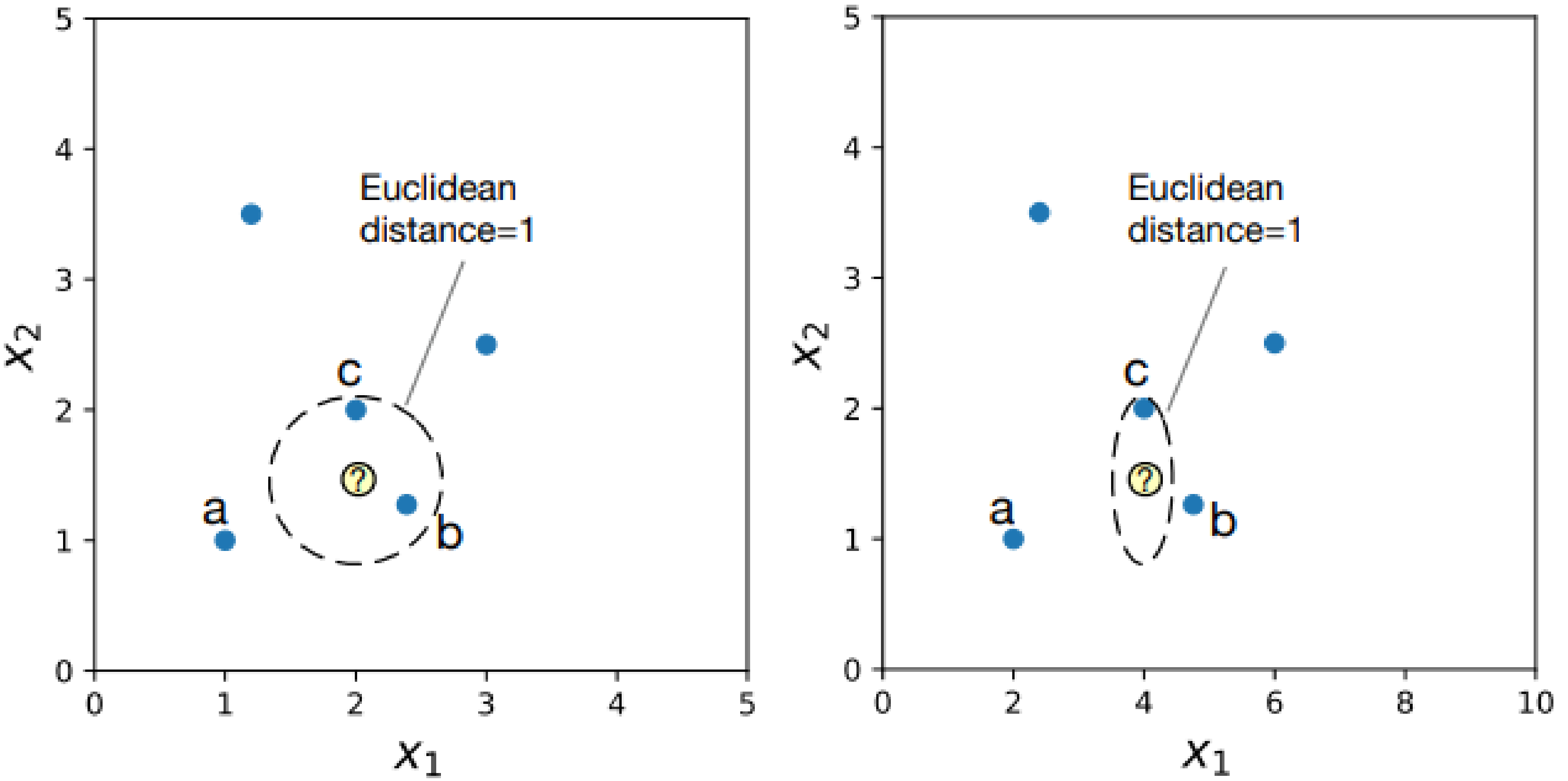
Hamming distance: $d(\mathbf{x}^{[a]}, \mathbf{x}^{[b]}) = \sum_{j=1}^{m} \left| x_j^{[a]} - x_j^{[b]} \right|$ where $x_j \in \{0,1\}$

Jaccard/Tanimoto similarity:

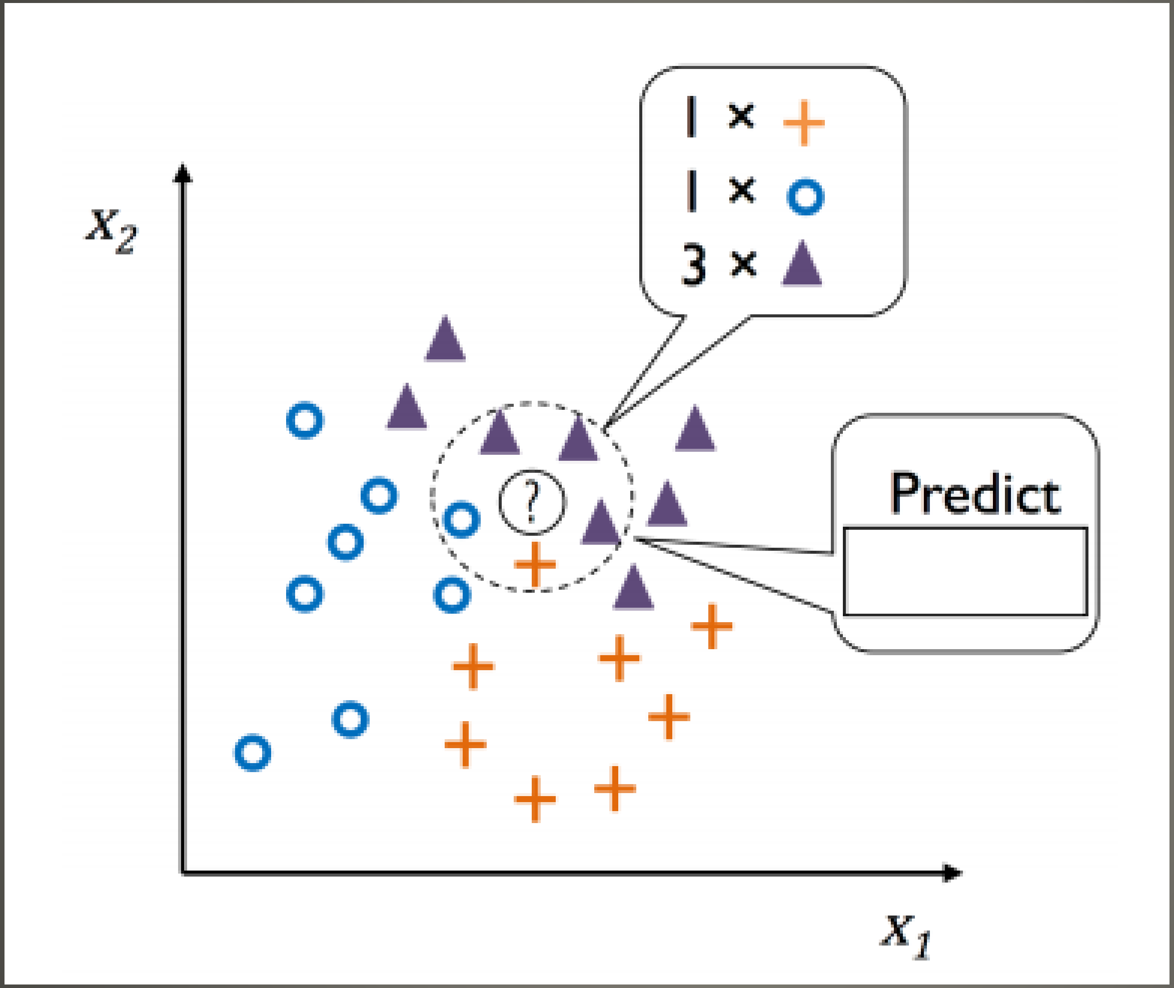
$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Dice: $D(A,B) = \frac{2|A \cap B|}{|A| + |B|}$

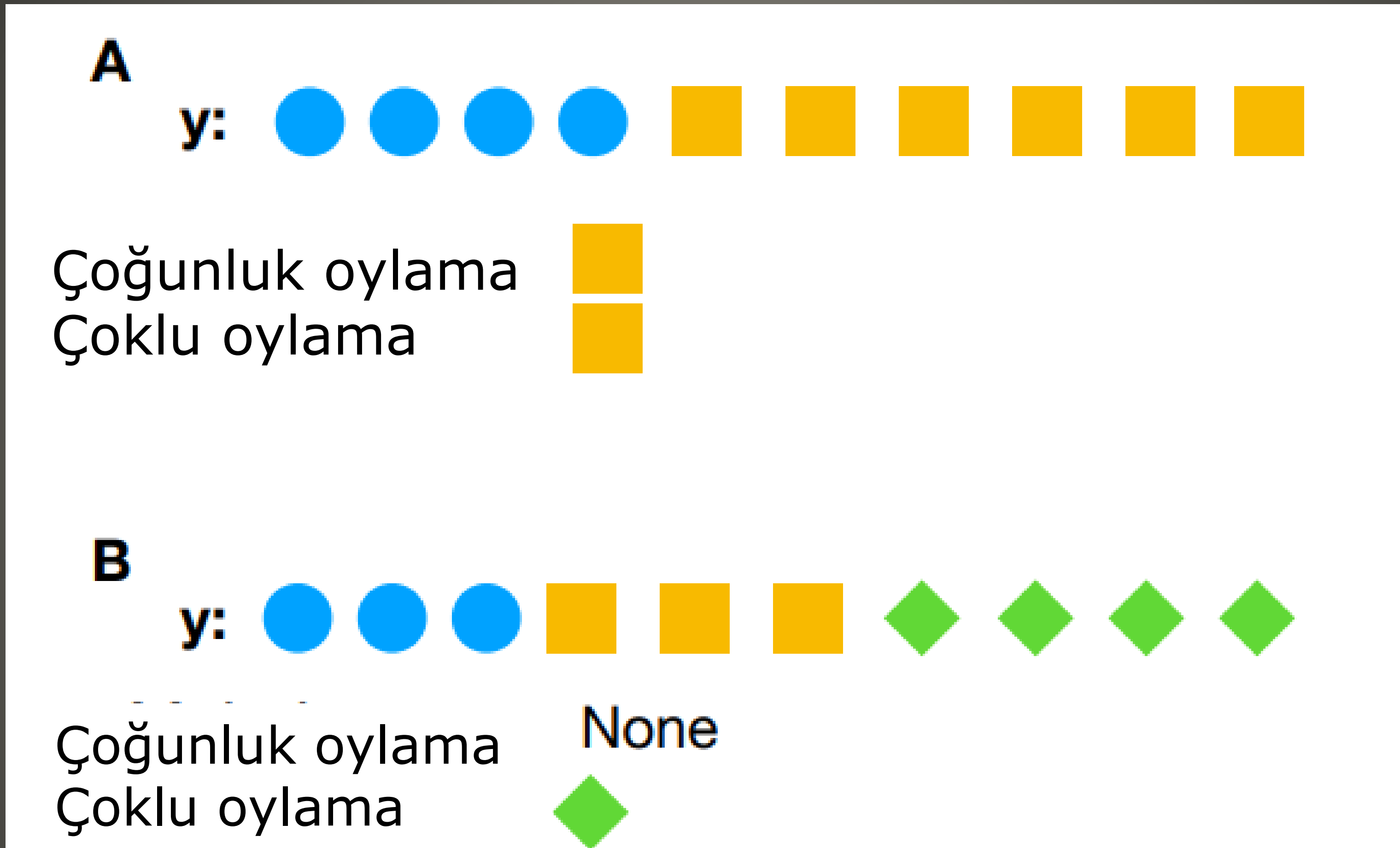
Bazı Ortak Ayrık mesafe Ölçümleri



Özellik Ölçekleme



K-En Yakın Komşu



K-En Yakın Komşu

K-En Yakın Komşu Özellik Ağırlıklandırma

- Her bir özellik boyutuna özellik ölçeklemeye eşdeğer bir ağırlık ekleyerek mesafe ölçümlerini değiştirebiliriz.
- Öklid mesafesi durumunda, bu aşağıdaki gibi görünecektir.

$$d_w(\mathbf{x}^{[a]}, \mathbf{x}^{[b]}) = \sqrt{\sum_{j=1}^{m} w_j \left(x_j^{[a]} - x_j^{[b]}\right)^2},$$

- Bunu kodda verimli bir şekilde uygulamak için, ağırlıklandırmayı bir dönüşüm matrisi olarak ifade edebiliriz; burada dönüşüm matrisi, m özellikleri için m ağırlık katsayılarından oluşan bir köşegen matristir:

$$\mathbf{W} \in \mathbb{R}^{m \times m} = \mathrm{diag}(w_1, w_2, ..., w_m).$$

$$d_w(\mathbf{x}^{[a]}, \mathbf{x}^{[b]}) = \sqrt{(\mathbf{x}^{[a]} - \mathbf{x}^{[b]})^T \mathbf{W} (\mathbf{x}^{[a]} - \mathbf{x}^{[b]})}.$$

- kNN'nin bir çeşidi, mesafe ağırlıklı kNN'dir.
- “Normal” kNN'de, tüm k komşular benzer şekilde çoğulcu oylama veya ortalamaya katılır.
- Ancak, özellikle bir komşu kümesini çevreleyen yarıçap büyükse, sorgu noktasına "daha yakın" olan komşulara daha güçlü bir ağırlık vermek isteyebiliriz.
- Örneğin, kNN sınıflandırmasında komşulara w ağırlığı atayabiliriz,:
- $w^{[i]} = \frac{1}{d(x^{[i]}, x^{[q]})}$

K-En Yakın Komşu Mesafe Ağırlıklandırma