

Veri Madenciliği

DR. ŞAFAK KAYIKÇI

Yapay Zeka

- Hem bilişsel sistemleri simule etmeyi (akılcı/insan gibi düşünmeyi) hem de, akıllı sistemler yapılandırmayı (akılcı/insan gibi davranışma) kendine amaç edinmiş bir bilimsel disiplindir.
- Özette :
 - İnsan gibi düşünen / İnsan gibi davranışan
 - Akılcı (rasyonel) düşünen / Akılcı davranışanBilgisayar sistemleri ile ilgilenir.



Veri Madenciliği

- Başta internet ve sosyal medya olmak üzere, çok sayıda verinin (big data) analizi üzerine birçok yeni teknik geliştirilmiştir.
- Çok miktarda verinin varlığından bilgi çıkarımı ve faydalı bilgiyi ortaya çıkarmak için yapılan analitik işlemler sonucunda elde edilen analizlerde veri madenciliği kapsamındadır.
- Çok sayıda verinin varlığı ve bu verinin işlenmesinde insan kapasitesinin yeterli olmaması, devreye makinelerin girmesine yol açmıştır.
- Makine öğrenmesi ise, bir makinenin bir işi kendi kendine ve en iyi performansla gerçekleştirmesi amacıyla taşıyan yapay zeka ile ilişkili bir çalışma alanıdır.
- Başka bir debole makine öğrenmesi, bilgisayarın bir karar verme problemini kendi kendine çözebilir hale getirmektir.

Makine Öğrenmesi Faktörleri

1. Sunulan veri seti : sayısı ve çeşitliliği
 - Şehirdeki doktor ve kırsaldaki doktor
2. Sonuca etki eden değişkenleri bulundurması
 - Öğrenci Final notu, Final notunun 2 katı
3. Seçilen öğrenme stratejisi

Öğrenme Stratejileri

- Danışmalı Öğrenme (supervised learning)
- Danışmansız Öğrenme (unsupervised learning)
- Pekiştirmeli Öğrenme (reinforcement learning)

Danışmanlı Öğrenme

- Amaç bir dizi girdi değerine dayanarak çıktı değerinin tahmin edilmesidir.
- Girdi değerleri bir danışman (eğitmen) tarafından çıktı değerlerine etiketlenir.
- Her girdi vektörüne sonlu sayıdaki ayrık kategorilerden birini atama olan durumlar, ***sınıflandırma (classification)*** problemi olarak tanımlanır.
- Eğer istenen çıktı bir yada daha fazla sürekli değerden oluşmakta ise bu yönteme ***regresyon (regression)*** denilmektedir.

Örnek

Banka Risk Grubu (düşük, orta, yüksek)

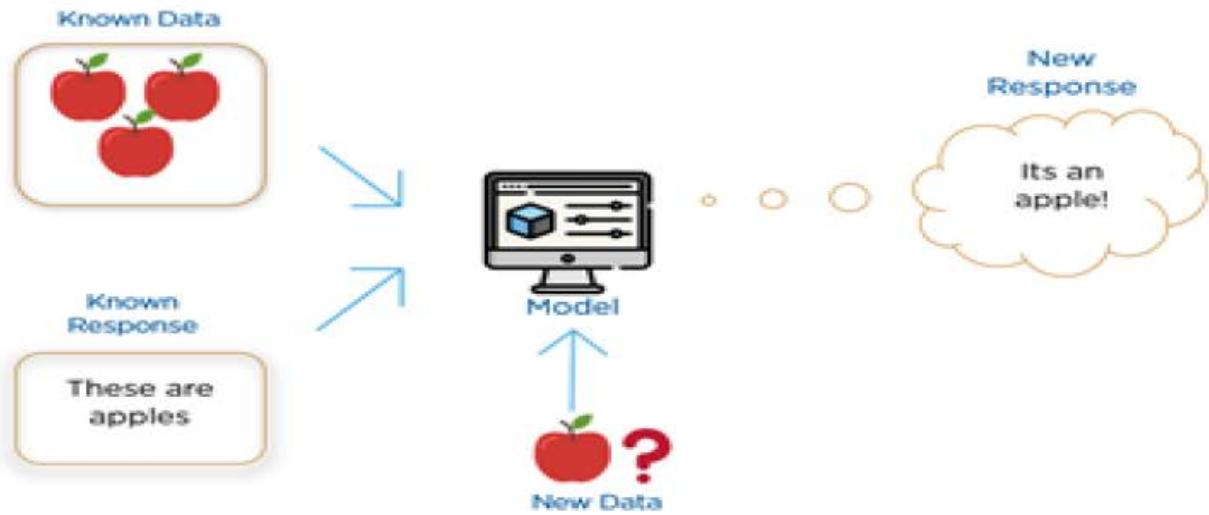
Yaş	Cinsiyet	Gelir (TL)	Borç Durumu	----	RİSK
30	Kadın	2500	Var	----	Yüksek
50	Erkek	3000	Yok	----	Düşük
----	----	----	----	----	----

45	Erkek	3500	Var	----	?
----	-------	------	-----	------	---

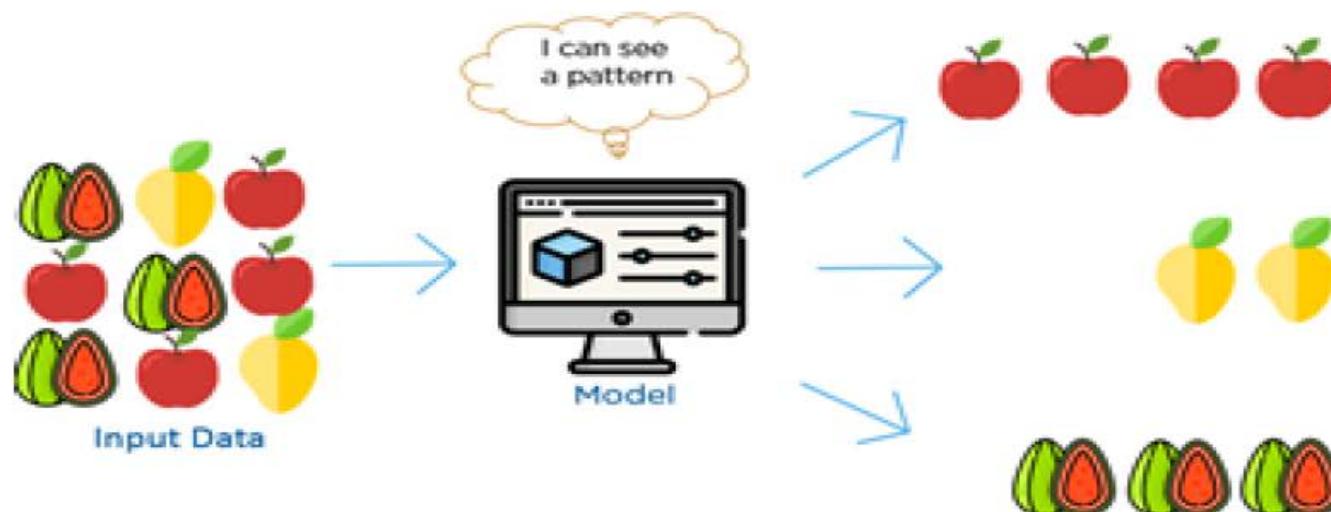
Danışmansız Öğrenme

- Çıktı değeri olmaksızın girdi değerleri arasındaki ilişki ve örüntülerin tanımlanmasıdır.
 - Girdi değerlerinin gruplanması, diğer bir deyle kümelenmesi (**clustering**) sağlanmaktadır.
 - Amaç:
 - Kümeleme
 - Girdi uzayında verinin dağılımını belirlemek – yoğunluk tahmini (density estimation)
 - Görselleştirme (visualization) ile
- Çok boyutlu uzayı, iki ya da üç boyutlu uzaya yansıtılmalıdır.

örn : müşteri segmentasyonları



supervised

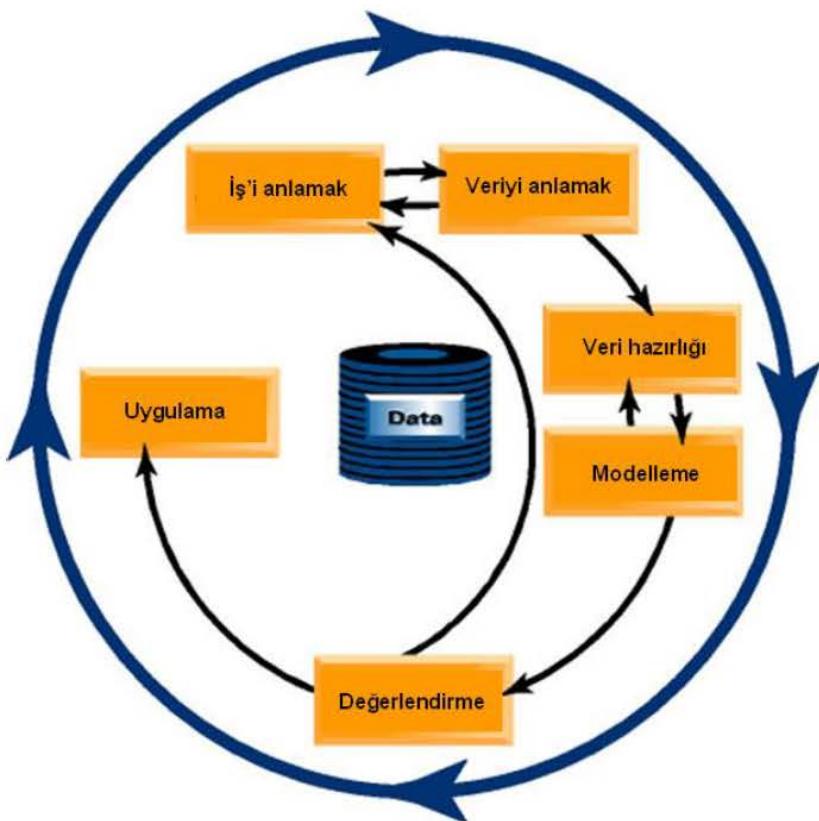


unsupervised

Pekiştirmeli Öğrenme

- Pekiştirmeli (yada eleştiri ile öğrenme) yaklaşımı, öğrenen makinenin davranışını , dinamik bir ortamda deneme-yanılma şeklinde bir etkileşime girerek öğrenmesidir.
- Geçici etiket (yada ödül) kullanılır. Bu geri bildirim, bir şeyin doğru yada yanlış olduğunu bildiren fakat neden yanlış olduğunu söylemeyen bir eleştirmen gibidir.

Makine Öğrenme Adımları



CRISP
Cross Industry Standard Process for Data Mining

1. İş'i anlama – Problemin tanımlanması

- Amaç belirlenmelidir : (hedef nitelik)
‘Çözülmek istenen problem nedir?’ ‘Makineden öğrenmesi istenen şey nedir?’
- Elimizde ne tür veriler var? Nicelik, nitelik

2.Veriyi Anlama

Bilgi Hiyerarşisi (D-I-K-W) (vebb)



Veri Tipleri

- **Nominal** : Kategorik tipteki değişkenlerdir. Örn: meslek grupları
- **İkili (Binary)** : Nominal değerlerin, sadece iki farklı değer aldığı durumlardır. Örn : Kadın-Erkek, Evli-Bekar, Evet-Hayır vb.
- **Sıralı (Ordinal)** : Nominal değerler mantıksal sıradan ise sıralı nitelik olarak adlandırılır. (az/orta/çok) ,(küçük/orta/büyük)
- **Tamsayı (Integer)** : Tamsayı tipindeki nitelikler olup, aritmetik işlemler yapmak mümkündür. Örn : evcil hayvan sayısı, takımın attığı gol sayısı

Veri tipleri gözden geçirildikten sonra, niteliklere ait ortalama, maksimum, minimum değerler, frekans bilgisi, mod, medyan gibi tanımlayıcı bilgiler incelenmelidir. Nitelikler arasındaki ilişkiler; kutu diyagramı (box plot), serpilme grafiği (scatter plt), pasta yada sütun grafiği gibi veriyi görselleştirme yolları ile tespit edilir.

3.Veriyi Hazırlama

- **Uç noktalar:** Uç noktalar, diğerlerinden bariz şekilde gözle görülür bir farkla ayrılmış gözlemlerdir. Şu şekillerde karşımıza çıkabilir:
 - Veri girişi sırasında karşılaşılan yada kodlama yapılrken meydana gelen prosedürel bir hata meydana gelmiş olabilir.
 - Sıra dışı bir olayın sonucu biçiminde olabilir.
 - Her bir değişken normal aralığında seyrederken, ilgili değişkenler biraraya geldiğinde bir gözlem üç nokta haline gelebilir.
- **Tekrar Eden Gözlemler :** Tekrar eden gözlemler bir yandan analizler sırasında vakit kaybı yaştatarak veri setinin gereksiz biçimde şişmesine neden olacaktır.
- **Kayıp Değerlerin Tamamlanması :** Bazı durumlarda ölçülemeyen değerler mevcut olabilir. Kayıp değerler içeren nitelik nümerik ise, tüm niteliğe ait ortalama değerin kayıp değere yazılmazı, kategorik ise en çok tekrar eden değerin kayıp değere yazılmazı gibi yöntemler izlenebilir. Kayıp değer sayısı çok fazla ise, ilgili nitelik analizden tamamıyla çıkarılabilir yada kayıp değerler tamamlanmadan önceki ve sonraki analiz sonuçları incelenerek, niteliğin dahil edilmiş edilmemesine karar verilebilir.

3.Veriyi Hazırlama

- **Normalizasyon :** Veri setinde, futbolcunun attığı gol sayısı ile transfer ücretinin tutulduğu bir durum için, bu numerik değerlerin değişim aralıkları birbirinden oldukça uzaktır. Büyük değerler, küçük değerleri domine ederek, sonuç üzerinde etkisi artmaktadır. Bu nedenle, özellikle uzaklık hesabına dayalı bazı algoritmalarla (k-en yakın komşuluk, k-ortalamalar vb) analizi olumsuz etkilemektedir.

Bu nedenle, bu değerler için bazı normalizasyon teknikleri geliştirilmiştir:

min-max normalization

$$v' = \frac{v - min_a}{max_a - min_a} (new_max - new_min) + new_min$$

- e.g. convert age=30 to range 0-1, when min=10,max=80.
 $new_age = (30-10)/(80-10) = 2/7$

z-score normalization

$$v' = \frac{v - mean_a}{stand_dev_a}$$

normalization by decimal scaling

$$v' = \frac{v}{10^j} \quad \text{Where } j \text{ is the smallest integer such that } \text{Max}(|v'|) < 1$$

4. Modelleme

- Model kurma, probleme ve öğrenme stratejisine uygun bir algoritmanın yardımı ile girdilerin istenilen çıktılara dönüştürülmesini ifade etmektedir.
- Bir model için tek bir model değil, birden fazla model inşa edilmekte, modellerin birbirleriyle performansları karşılaştırılmaktadır.
- Aynı öğrenme biçimini, farklı parametrelerle denenebilir.

5-Performans Değerlendirme Yöntemleri

- **Holdout (dışarıda tutma):** Veri setinin eğitim ve test olmak üzere iki parçaya ayrıldığı yöntemdir. Test setindeki veri, eğitim setinde kullanılan verinin dışındadır.
- Bu yöntemde iki temel dezavantaj vardır.
 - Veri setindeki gözlem sayısının az olması durumunda, test verisi için yeteri kadar örneğin ayırlamayacak olmasıdır.
 - Veri setinin eğitim ve test veri seti olarak sadece bir defaya mahsus ayrılmış olması ve elde edilen performansa da bu ayırım nedeniyle yeterince güvenilmemesidir.

5-Performans Değerlendirme Yöntemleri

- **Tekrarlı Holdout (repeated holdout)**: Kısaca holdout yönteminin, birkaç defa tekrarlanmasıdır. Bu yöntemde seçim işlemi her ne kadar rastgele yapılsa da, seçilmiş farklı veri setleri üst üste binebileceğinden tekrarlı holdout yöntemi elverişli görünmemektedir.
- **Tabakalı Öğrenme (Stratified sampling)**: Hedef niteliğin nominal tipte olduğu setlerde, hedef niteliğin bazı kategorilere ait örnekler az sayıda olabilir ya da hiç olmayıabilir. Bu gibi durumlarda, veri seti eğitim ve test olarak ayrılırken hedef niteliğin kategorilerine ait oranlarının korunması istenebilir. Bu nedenle tabakalı öğrenme tercih edilmektedir. Hedef niteliğin numerik olduğu durumlarda bu yöntemden faydalılmaz.

5-Performans Değerlendirme Yöntemleri

- **Üçlü ayırma (three-way split)** : Model seçimi ve peformans tahminin aynı anda gerçekleştirildiği yöntemdir. Veri; eğitim-doğrulama-test seti olmak üzere üçe ayrılır. Doğrulama veri setindeki örnekler ile kullanılan algoritmaya ait parametrelerin ince ayarı yapılmaktadır. Test veri seti ise nihai performans ölçümü için kullanılır.
- **K-kat çapraz geçerleme (k-fold cross validation)**: veri seti k eşit parçaya ayrılır. Elde edilen k parçasının her biri bir kez test veri seti, kalan $k-1$ parça ise eğitim veri seti olarak seçilir. k defa elde edilen performans ölçülerinin ortalaması alınır ve nihai performans elde edilir.

5-Performans Değerlendirme Yöntemleri

- **Monte Carlo Çapraz Geçerleme (Random sampling)**: Veri seti k defa bölünmekte, eğitim ve test veri setleri kullanıcının belirlediği orana göre oluşturulmaktadır. Çapraz geçerlemeden farkı, analizde oluşturulan k adet test veri setinde aynı noktaların tekrar edebilemesidir.
- **Bootstrap Örnekleme (bootstrap sampling)** : n adetli bir veri setinde, veri setinden eğitim seti için n defa rastgele örnek seçilmektedir. Ancak seçilen örnek veri setinden çıkarılmadan seçim işlemi sürdürülür. Bu nedenle eğitim setinde bir örnek birden fazla tekrar edebilir. Eğitim veri seti oluşturulduktan sonra, eğitim veri setine alınmayan bütün örnekler test veri setine aktarılır. Test veri setindeki her örnek yalnızca bir defa tekrar edebilecektir.

5- Performans Değerlendirme Öcütleri - Confusion Matrix (Kontenjans Tablosu – Hata Matrisi)

		Gerçek		
		Pozitif	Negatif	Toplam
Tahmin	Pozitif	Doğru Pozitif (dp)	Yanlış Pozitif (yp)	tPoz
	Negatif	Yanlış Negatif (yn)	Doğru Negatif (dn)	tNeg
	Toplam	poz	neg	m

Kalp hastalığı var mı / yok mu ?

- dp (tp) : Gerçekte kalp hastası olan hastalardan, modelin kalp hastası olarak tahmin ettiği hastaların sayısıdır.
- dn (tn) : Gerçekte kalp hastası olmayan hastalardan, modelin kalp hastası değildir biçiminde tahmin ettiği hastaların sayısıdır.
- yp (fp) : Gerçekte kalp hastası olan hastalardan, modelin kalp hastası değildir biçiminde tahmin ettiği hastaların sayısıdır.
- dn (fn) : Gerçekte kalp hastası olmayan hastalardan, modelin kalp hastası olarak tahmin ettiği hastaların sayısıdır.

Veri Madenciliği

DR. ŞAFAK KAYIKÇI

Hata Matrisi

Makine öğrenmesinde kullanılan sınıflandırma modellerinin performansını değerlendirmek için hedef niteliğe ait tahminlerin ve gerçek değerlerin karşılaştırıldığı hata matrisi sıkılıkla kullanılmaktadır.

1. Doğruya doğru demek (True Positive – TP) **DOĞRU**
2. Doğruya yanlış demek (True Negative – TN) **YANLIŞ**
3. Yanlışa doğru demek (False Positive – FP) **YANLIŞ**
4. Yanlışa yanlış demek (False Negative – FN) **DOĞRU**

Hastalık var mı ? E / H

n=165	Tahmin HAYIR	Tahmin EVET	
Gerçek HAYIR	50	10	60
Gerçek EVET	5	100	105
	55	110	

Bu matristen neler öğrenebiliriz?

- Öngörülen iki olası sınıf vardır: "evet" ve "hayır"
- Toplam 165 hasta için tahmin yapılmıştır
- Bu 165 davadan, sınıflandırıcı "evet" 110 kez ve "hayır" 55 kere tahmin etmiştir.
- Gerçekte, 105 kişi hastadır ve 60 hasta değildir.

Örnek

Hastalık var mı ? E / H

n=165	Tahmin HAYIR	Tahmin EVET	
Gerçek HAYIR	50	10	60
Gerçek EVET	5	100	105
	55	110	

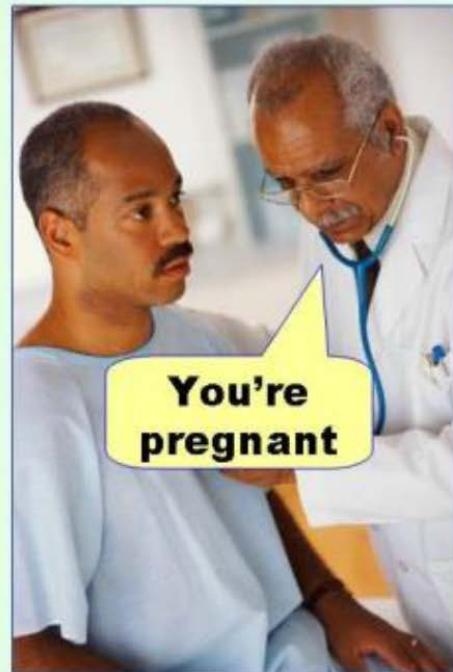
Terimler :

- Gerçek Pozitif (TP – True Positive) : Hasta tahmin ettiğimiz(Evet) ve aslında hasta olan kişiler (Evet) (100)
- Gerçek Negatif (TN – True Negative) : Hasta değil tahmin ettiğimiz (Hayır) ve aslında hasta olmayan kişiler (Hayır) (50)
- Yanlış Pozitif (FP – False Positive) : Hasta tahmin ettiğimiz (Evet) ama aslında hasta olmayan kişiler (Hayır) (10) (Tip 1 Hatası)
- Yanlış Negatif (FN – False Negatif) : Hasta değil tahmin ettiğimiz (Hayır) ve aslında hasta olan kişiler (Evet) (5) (Tip 2 Hatası)

- **Doğruluk(Accuracy)**: Toplamda, sınıflandırıcı ne kadar doğrudur?
$$(TP + TN) / \text{toplam} = (100 + 50) / 165 = 0.91$$
- **Sınıflandırılma (Hata) Oranı (Error Rate)**: Toplamda, sınıflandırıcı ne kadar yanlıstır?
$$(FP + FN) / \text{toplam} = (10 + 5) / 165 = 0,09$$
$$1 - \text{Doğruluk} = 1 - 0.91 = 0,09$$
- **Gerçek Pozitif Oran (True Pozitif Rate) (Hassasiyet (Sensitivity) veya Hatırlama(Recall) :**
Gerçekte evet olduğunda, ne kadar evet tahmin eder?
$$TP / \text{gerçek evet} = 100/105 = 0,95$$
- **Yanlış Pozitif Oran:** Aslında hayır olduğunda, ne kadar evet tahmin ediyor?
$$FP / \text{gerçek hayır} = 10/60 = 0,17$$
- **Gerçek Negatif Oran (Specificity- Özgünlük):** Aslında hayır olduğunda, , ne kadar hayır tahmin ediyor?
$$TN / \text{gerçek hayır} = 50/60 = 0,83$$
$$1 - \text{Yanlış Pozitif Oranı} = 1 - 0,17 = 0,83$$
- **Kesinlik(Precision):** Evet tahmin ettiğinde, ne kadar doğrudur?
$$TP / \text{tahmin edilen evet} = 100/110 = 0,91$$
- **Prevalans:** Evet koşulumuzörnekte ne sıklıkla görülür?
$$\text{Gerçek evet} / \text{toplam} = 105/165 = 0.64$$

Tip 1 (FP) – Tip 2 (FN) Hatalar

Type I error
(false positive)



Type II error
(false negative)



Veri Madenciliği

DR. ŞAFAK KAYIKÇI

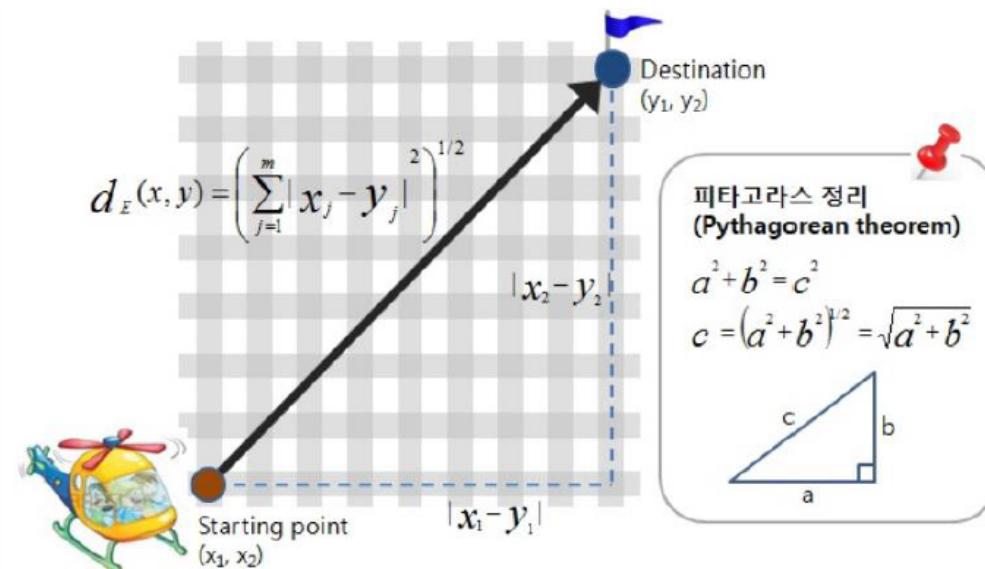
Uzaklık Ölçüleri

- Veriler arasındaki benzerliği ölçmek için uzaklık ölçüleri kullanılır.
- Bir uzaklık ölçüsü yada uzaklık fonksiyonu, x, y, z koordinatlarında reel değerli bir d fonksiyonu olmak üzere :
 - $d(x,y) \geq 0$ ve $d(x,y) = 0 \iff x = y$: Uzaklık her zaman pozitiftir ve koordinatlar için uzaklığın sıfır olmasının tek yolu koordinatların aynı olmalarıdır
 - $d(x,y) = d(y,x)$: komütatiflik
 - $d(x,z) \leq d(x,y) + d(y,z)$: üçüncü bir nokta, iki nokta arasındaki uzaklığı hiçbir zaman kısaltmaz (üçgen eşitsizliği)

Öklid Uzaklığı

Öklid uzaklığı, sınıflandırma ve kümeleme algoritmalarında en sık kullanılan uzaklık ölçütüdür. Öklid uzaklığı, iki nokta arasındaki doğrusal uzaklık olup herhangi iki nokta, P ve Q arasındaki Öklid uzaklığı $P=(x_1, x_2, \dots, x_n)$ ve $Q=(y_1, y_2, \dots, y_n)$ olmak üzere, Eşitlik bu şekilde sıralanır :

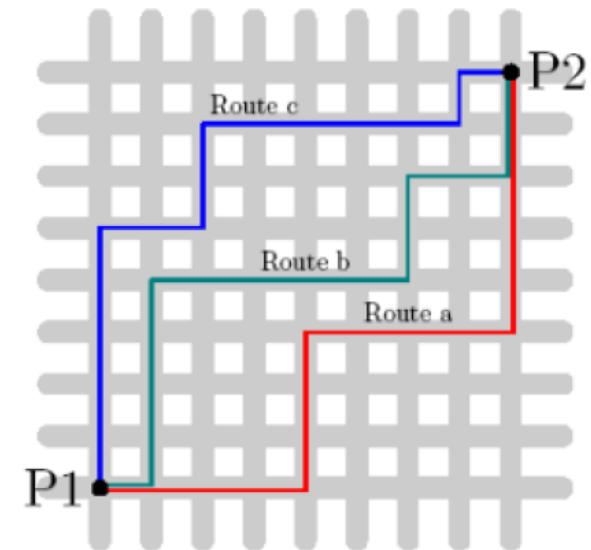
$$\left(\sqrt{\sum_{i=1}^n (x_i - y_i)^2} \right)$$



Manhattan Uzaklığı

Manhattan uzaklığı, n boyutlu iki nokta arasındaki farkların mutlak değerlerinin toplamıdır. Herhangi iki nokta, P ve Q arasındaki Manhattan uzaklığı $P=(x_1, x_2, \dots, x_n)$ ve $Q=(y_1, y_2, \dots, y_n)$ olmak üzere, Eşitlik şu şekilde hesaplanır:

$$d = \sum_{i=1}^n |x_i - y_i|$$



Minkowski Uzaklığı

Minkowski uzaklığı, Öklid uzayında tanımlı bir dizidir. Sınıflandırma, kümeleme gibi makine öğrenmesi, veri madenciliği uygulamalarında sıkılıkla kullanılan Öklid uzaklığı, Manhattan uzaklığı gibi uzaklık ölçütlerinin genelleştirilmiş halidir. Herhangi iki nokta P ve Q arasındaki Minkowski uzaklığı P=(x₁, x₂, ..., x_n) ve Q=(y₁, y₂, ..., y_n) olmak üzere :

$$\left(\sum_{i=1}^k (|x_i - y_i|)^q \right)^{1/q}$$

Genel olarak

✓ 1-norm distance

$$r = 1 \rightarrow d(x, y) = \sum_{j=1}^m |x_j - y_j|$$

✓ 맨하탄 거리
(Manhattan Distance)

✓ 2-norm distance

$$r = 2 \rightarrow d(x, y) = \left(\sum_{j=1}^m |x_j - y_j|^2 \right)^{1/2}$$

✓ 유클리드 거리
(Euclidean Distance)

(In the Euclidean space \mathbb{R}^n , the distance between two points)

✓ p-norm distance

$$r = p \rightarrow d(x, y) = \left(\sum_{j=1}^m |x_j - y_j|^p \right)^{1/p}$$

✓ Infinity norm distance

$$r \rightarrow \infty \rightarrow d(x, y) = \lim_{r \rightarrow \infty} \left(\sum_{j=1}^m |x_j - y_j|^r \right)^{1/r} = \max |x_j - y_j|$$

Kategorik Veriler

Bu uzaklık hesaplarını kullanabilmek için, verilerin nümerik olması gerekmektedir. Kategorik (nominal – binary) durumlarda kullanılırsa şu durum oluşur :

Gözlem	Cinsiyet	Sigara	Alkol
1	1	1	1
2	0	0	1
3	0	1	0
4	1	0	0
----	----	----	----

$$d_{\text{öklid}}(X_1, X_2) = \sqrt{(1 - 0)^2 + (1 - 0)^2 + (1 - 1)^2} = \sqrt{2}$$

$$d_{\text{öklid}}(X_1, X_3) = \sqrt{(1 - 0)^2 + (1 - 1)^2 + (1 - 0)^2} = \sqrt{2}$$

$$d_{\text{öklid}}(X_1, X_4) = \sqrt{(1 - 1)^2 + (1 - 0)^2 + (1 - 0)^2} = \sqrt{2}$$

Kategorik verilerin olduğu durumlarda : Simple Matching Distance, Jaccard Distance, Gower Benzerliği vb. kullanılabilir.

k -en yakın komşuluk algoritması (KNN)

Bana arkadaşını söyle, sana kim olduğunu söyleyeyim.

1951 –Fix ve Hodges–K nearest Neighbour

Yeni bir örneğin sınıfı, örneğin belirlenen bir k değerine göre, mevcut örneklem içindeki örneklerle olan uzaklığı hesaplanarak (k tane en yakın komşusu bulunarak) tespit edilir.

kNN

Avantajları

Eğitim hızlıdır

Öğrenme basit ve kolaydır

Gürültülü eğitim verisine gösterdiği direnç

Eğitim verisi fazla iken verimliliği

Dezavantajları

k değerine bağlıdır

Hesaplama karmaşası

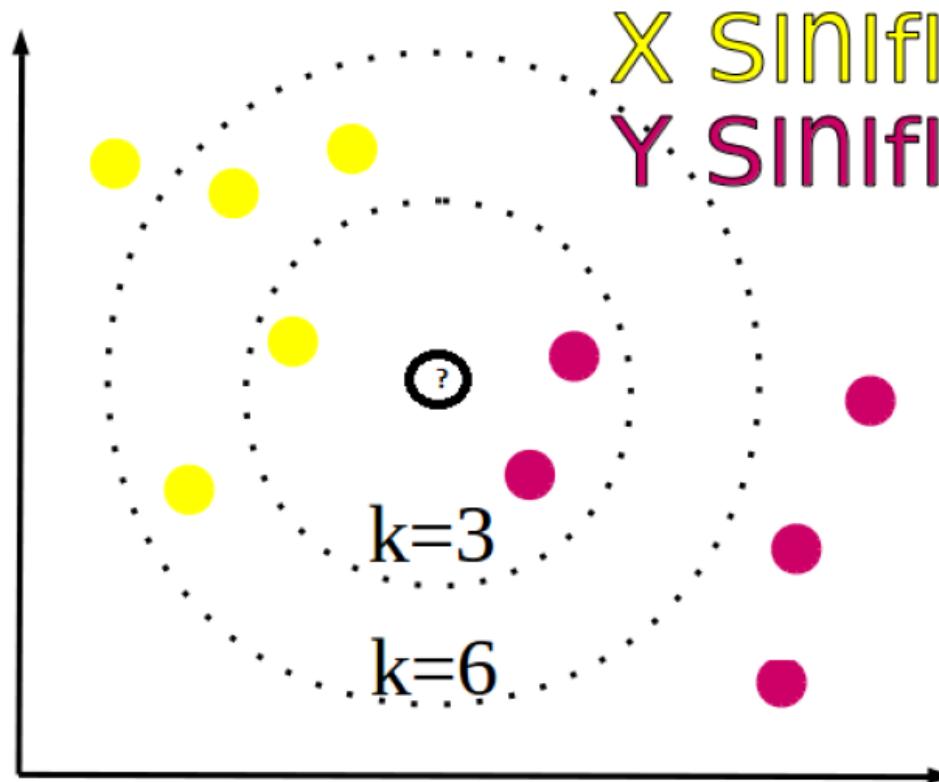
Hafıza kısıtlaması

İlgisiz nitelikler ile çabuk hataya düşmesi

kNN

- 1.Adım : Sınıfı belirlenmek istenen yeni örneğin, eğitim örnekleminde yer alan örneklerle uzaklığı hesaplanır
- 2.Adım : Hesaplanan uzaklıklar sıralanır, içlerinden en küçük (yakın) k tanesi seçilir
- 3.Adım : Yeni örneğin sınıfı için oylama (voting) yapılır. Oylamada yaygın olan iki yaklaşım mevuttur :
 - Çoğunluk oylaması (Majority Voting) : en çok tekrar eden sınıfın, yeni örneğin aranan sınıfı olduğu yöntemdir
 - Ters Mesafe Ağrılıklı Oylama (Inverse Distance Weighted Voting) : Birbirine yakın komşular daha yüksek almaktadır. Komşunun oyunu, o komşu ile sınıfı bulunmak istenen nokta(q) arasındaki uzaklığın tersi olarak belirlenmesidir. Sonunda tüm ağırlıklar toplanarak, en yüksek ağırlığa sahip sınıf seçilmektedir.

kNN



Örnek veriseti

Örnek	Yaş	Dinlenme Anındaki Kan Basıncı	Kolesterol	Max Kalp Atış Hızı	Kalp Hastalığı
1	55	110	275	169	VAR
2	65	150	268	151	VAR
3	50	120	225	140	YOK
4	40	130	351	115	YOK
5	60	125	203	170	YOK
6	48	100	305	165	YOK
7	25	103	185	120	VAR
8	53	170	270	155	VAR
9	67	140	328	128	VAR
10	72	115	400	172	VAR

Soru

Yaşı 70, kan basıncı 175,コレsterolü 200 ve en yüksek kalp atış hızı 150 olan bir hastada kalp rahatsızlığı olup olmadığını bulunuz?

Örnek	Yaş	Dinlenme Anındaki Kan Basıncı	Kolesterol	Max Kalp Atış Hızı	Kalp Hastalığı
1	55	110	275	169	VAR
2	65	150	268	151	VAR
3	50	120	225	140	YOK
4	40	130	351	115	YOK
5	60	125	203	170	YOK
6	48	100	305	165	YOK
7	25	103	185	120	VAR
8	53	170	270	155	VAR
9	67	140	328	128	VAR
10	72	115	400	172	VAR

11

70

175

200

150

?

hesaplama

$$\text{Öklid Uzaklığı : } \left(\sqrt{\sum_{i=1}^n (x_i - y_i)^2} \right) \quad k=3$$

$$d_{\text{öklid}}(x_1, x_{11}) = \sqrt{(55 - 70)^2 + (110 - 175)^2 + (275 - 200)^2 + (169 - 150)^2} = 102,6$$

$$d_{\text{öklid}}(x_1, x_{11}) = 102,6$$

$$d_{\text{öklid}}(x_2, x_{11}) = 72,63$$

$$d_{\text{öklid}}(x_3, x_{11}) = 64,42$$

$$d_{\text{öklid}}(x_4, x_{11}) = 164,17$$

$$d_{\text{öklid}}(x_5, x_{11}) = 54,85$$

$$d_{\text{öklid}}(x_6, x_{11}) = 131,75$$

$$d_{\text{öklid}}(x_7, x_{11}) = 91,29$$

$$d_{\text{öklid}}(x_8, x_{11}) = 72,38$$

$$d_{\text{öklid}}(x_9, x_{11}) = 134,54$$

$$d_{\text{öklid}}(x_{10}, x_{11}) = 209,97$$

Küçükten büyüğe : $d_{\text{öklid}}(x_5, x_{11}) < d_{\text{öklid}}(x_3, x_{11}) < d_{\text{öklid}}(x_8, x_{11})$

: Örnek 5 → YOK, Örnek 3 → YOK, Örnek 8 → Var

Çoğunluk oymalaya göre YOK olarak bulunur

soru 2 (Çoğunluk Oylaması)

Yaşı 45, kan basıncı 120, kolesterolü 110 ve en yüksek kalp atış hızı 100 olan bir hastada kalp rahatsızlığı olup olmadığını bulunuz? (k=3)

- $d_{\text{öklid}}(x_1, x_{11}) = 179,40$

- $d_{\text{öklid}}(x_2, x_{11}) = 169,90$

- $d_{\text{öklid}}(x_3, x_{11}) = 121,86$

- $d_{\text{öklid}}(x_4, x_{11}) = 241,73$

- $d_{\text{öklid}}(x_5, x_{11}) = 117,47$

- $d_{\text{öklid}}(x_6, x_{11}) = 206,54$

- $d_{\text{öklid}}(x_7, x_{11}) = 81,94$

- $d_{\text{öklid}}(x_8, x_{11}) = 176,60$

- $d_{\text{öklid}}(x_9, x_{11}) = 221,79$

- $d_{\text{öklid}}(x_{10}, x_{11}) = 300,06$

Küçükten büyüğe: Örnek 7 → VAR, Örnek 5 → YOK, Örnek 3 → YOK

Çoğunluk oymalaya göre YOK olarak bulunur

soru 2 (Ağırlıklı Oylama)

$d_{\text{öklid}}$	$\frac{1}{(d_{\text{öklid}})^2}$
$d_{\text{öklid}}(x_1, x_{11}) = 179,40$	0.000031
$d_{\text{öklid}}(x_2, x_{11}) = 169,90$	0.000035
$d_{\text{öklid}}(x_3, x_{11}) = 121,86$	0.000067
$d_{\text{öklid}}(x_4, x_{11}) = 241,73$	0.000017
$d_{\text{öklid}}(x_5, x_{11}) = 117,47$	0.000072
$d_{\text{öklid}}(x_6, x_{11}) = 206,54$	0.000023
$d_{\text{öklid}}(x_7, x_{11}) = 81,94$	0.000149
$d_{\text{öklid}}(x_8, x_{11}) = 176,60$	0.000032
$d_{\text{öklid}}(x_9, x_{11}) = 221,79$	0.000020
$d_{\text{öklid}}(x_{10}, x_{11}) = 300,06$	0.000011

Büyükten küçüğe :

Örnek 7 → VAR, Örnek 5 → YOK, Örnek 3 → YOK

YOK sınıfı toplamı (Örnek 3 + Örnek 5):

$$0.000067 + 0.000072 = \underline{\underline{0.000139}}$$

VAR sınıfı toplamı (Örnek 7) :

$$\underline{\underline{0.000149}}$$

0.000149 > 0.000139 olduğundan

ağırlıklı oylamaya göre
VAR olarak sınıflandırılır.

```

require("class") # load pre-installed package
## Loading required package: class

require("datasets")

data("iris") # load Iris Dataset
str(iris) #view structure of dataset
summary(iris) #view statistical summary of dataset
head(iris) #view top rows of dataset
set.seed(99) # required to reproduce the results
rnum<- sample(rep(1:150)) # randomly generate numbers from 1 to 150
rnum
iris<- iris[rnum,] #randomize "iris" dataset
head(iris)
# Normalize the dataset between values 0 and 1
normalize <- function(x) {
  return ((x-min(x)) / (max(x)-min(x)))
}

iris.new<- as.data.frame(lapply(iris[,c(1,2,3,4)],normalize))
head(iris.new)

```

```

# subset the dataset
iris.train<- iris.new[1:130,]
iris.train.target<- iris[1:130,5]
iris.test<- iris.new[131:150,]
iris.test.target<- iris[131:150,5]
summary(iris.new)
modell<- knn(train=iris.train, test=iris.test, cl=iris.train.target, k=16)
table(iris.test.target, modell, dnn = c("Tahmini Siniflar","Gercek Siniflar"))

```

Veri Madenciliği

DR. ŞAFAK KAYIKÇI

Karar Ağaçları (Decision Trees)

- Kararın verilebilmesi için ağaç biçiminde bir yapı oluşturmaktadır.
- **Karar düğümleri** (decision nodes) veri setinde karar vermek, sınıflandırmak için kullanılarak, bunlar iki yada daha fazla **dala** (branch) ayrılmaktadır. **Yaprak düğümleri** ise kararları tutmaktadır.
- Ağacın en tepesindeki düğüm, **kök düğüm** (root node) olarak adlandırılır.

Karar Ağaçları (DecisionTrees)

Karar ağaçlarının ID3, C4.5, CART gibi kullanılan algoritmalar, genelleştirme hatasını en aza indirmeyi amaçlamaktadır.

En uygun fonksiyon, bölme ölçülerini dikkate alarak seçilir. Bilgi Kazancı (Information Gain), Kazanç Oranı (Gain Ratio) ve Gini Endeksi (Gini Index) gibi tek değişkenli bölme kriterlerinin yanı sıra çok değişkenli bölme kriterleri de mevcuttur.

Karar Ağaçları (Avantaj –Dezavantaj)

- Sınıflandırma ve tahmin için popüler araçlardır.
- İnsanlar tarafından anlaşılabilen kurallar yaratır ve gösterir.
- Sürekli ve kategorik değişkenlerle çalışma becerileri vardır.
- Tahmin yada sınıflandırma için hangi alanların daha önemli olduğunu açıkça ortaya koyar.
- Sınıflandırma çabuk yapılırken, ağaç oluşturma süresi diğer sınıflandırıcırlara göre uzundur.
- Sürekli değerler içeren hedef değişkenlerin tahmini için uygun değildir.
- Az sayıda veri ve çok sayıda sınıf ile zayıf performans gösterir.

ID3 Karar Ağacı Algoritması

Yalnızca kategorik değerler için çalışmaktadır.

Karar ağacı oluştururken her adımda tüm niteliklere ait bilgi kazancı hesaplanmaktadır.

Bilgi kazancı hesaplanırken, belirsizliğin ölçüsü yada saf(sız)lık ölçüsü olarak tanımlanan ENTROPI'den faydalılmaktadır.

Entropi

- X , sınıf etiketleri bulunan eğitim kümесini, $C_i; i=1, \dots, k$ sınıf sayısını, $C_{i,x}$, X 'deki C_i sınıfına ait gözlemlerine kümесini, $|X|$ vs $|C_{i,x}|$ de sırası ile X ve $C_{i,x}$ 'deki gözlemlerin sayısı
- $p_i = \frac{|C_{i,x}|}{|X|}$, X 'deki C_i sınıfına ait bir gözlemin olasılığı
- $\text{Entropi}(X) = - \sum_{i=1}^k p_i \log_2(p_i)$

Örnek = ?

Entropi(Hedef Nitelik)

- $P_1 = P_{\text{düşük}} = \frac{4}{20}$

- $P_2 = P_{\text{orta}} = \frac{6}{20}$

- $P_3 = P_{\text{yüksek}} = \frac{10}{20}$

- $$\begin{aligned} \text{Entropi}(X) &= - \sum_{i=1}^k p_i \log_2(p_i) \\ &= - \left(\frac{4}{20} \log_2\left(\frac{4}{20}\right) + \frac{6}{20} \log_2\left(\frac{6}{20}\right) + \frac{10}{20} \log_2\left(\frac{10}{20}\right) \right) \\ &= 1.485475 \end{aligned}$$

Entropi(Herbir Nitelik)

$$\text{Entropi}_A(x) = \sum_{j=1}^v \frac{|X_j|}{|X|} * \text{Entropi}(X_j)$$

$$\text{Bilgi Kazancı}(A) = \text{Entropi}(X) - \text{Entropi}_A(X)$$

Bina Yaşı İçin

$$\text{Entropi}(X_{0-10}) = -\left(\frac{2}{6} \log_2\left(\frac{2}{6}\right) + \frac{3}{6} \log_2\left(\frac{3}{6}\right) + \frac{1}{6} \log_2\left(\frac{1}{6}\right)\right) = 1.4591$$

$$\text{Entropi}(X_{10-20}) = -\left(\frac{5}{9} \log_2\left(\frac{5}{9}\right) + \frac{2}{9} \log_2\left(\frac{2}{9}\right) + \frac{2}{9} \log_2\left(\frac{2}{9}\right)\right) = 1.4355$$

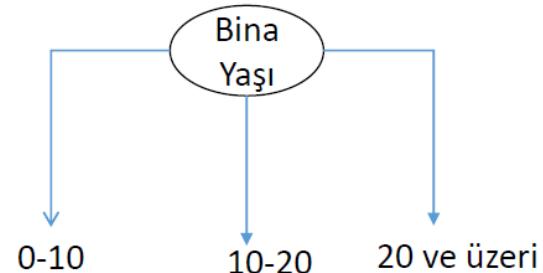
$$\text{Entropi}(X_{20 \text{ ve üzeri}}) = -\left(\frac{3}{5} \log_2\left(\frac{3}{5}\right) + \frac{1}{5} \log_2\left(\frac{1}{5}\right) + \frac{1}{5} \log_2\left(\frac{1}{5}\right)\right) = 1.3710$$

$$\begin{aligned}\text{Entropi}_{(\text{bina yaşı})} X &= \sum_{j=1}^3 \frac{|X_j|}{|X|} * \text{Entropi}(X_j) \\ &= \left(\frac{6}{20}\right)(1.4591) + \left(\frac{9}{20}\right)(1.4355) + \left(\frac{5}{20}\right)(1.3710) \\ &= 1.4265\end{aligned}$$

$$\text{Bilgi Kazancı}_{(\text{bina yaşı})} = \text{Entropi}(X) - \text{Entropi}_{(\text{bina yaşı})} X = 1.4855 - 1.4265 = 0.0590$$

Bilgi Kazançları

- Bilgi Kazancı $(\text{bina yaşı}) = 0.0590 \rightarrow$ En yüksek. Karar ağacı buradan başlar.
- Bilgi Kazancı $(\text{kat sayısı}) = 0.0357$
- Bilgi Kazancı $(\text{bakım}) = 0.0073$



Sonra diğer adımlara geçilir.

C4.5 Karar Ağacı Algoritması

- Hem kategorik hem de nümerik değerler için çalışmaktadır.

- Bölme Bilgisi $A(X) = - \sum_{j=1}^v \frac{|X_j|}{|X|} * \log_2\left(\frac{|X_j|}{|X|}\right)$

- Kazanç Oranı(A)= $\frac{Bilgi\ Kazancı(A)}{Bölme\ Bilgisi(A)}$

Örnek

- Değerler küçükten büyüğe sıralanır

5 – 5 – 5 – 8 – 8 – 8 – 12 – 12 – 12 – 15 – 15 – 17 – 17 – 17 – 17 – 25 – 25 – 30 – 30 – 40

- Tekrar eden değerler çıkarılır

5 – 8 – 12 – 15 – 17 – 25 – 30 - 40

- Entropi, Bölme Bilgisi ve Kazan Oranı hesaplanır

```
data("iris")
set.seed(1234)
SampleID <- sample(2, nrow(iris), replace = TRUE, prob = c(0.7, 0.3))
trainData <- iris[SampleID==1, ]
testData <- iris[SampleID==2, ]
install.packages("party") #party kütüphanesi
library(party)
iris_ctree <- ctree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width, data = trainData)
plot(iris_ctree, type="simple")
testPred <- predict(iris_ctree, newdata = testData)
table(testPred, testData$Species)
confusionMatrix(testPred,testData$Species) #caret kütüphanesi
```

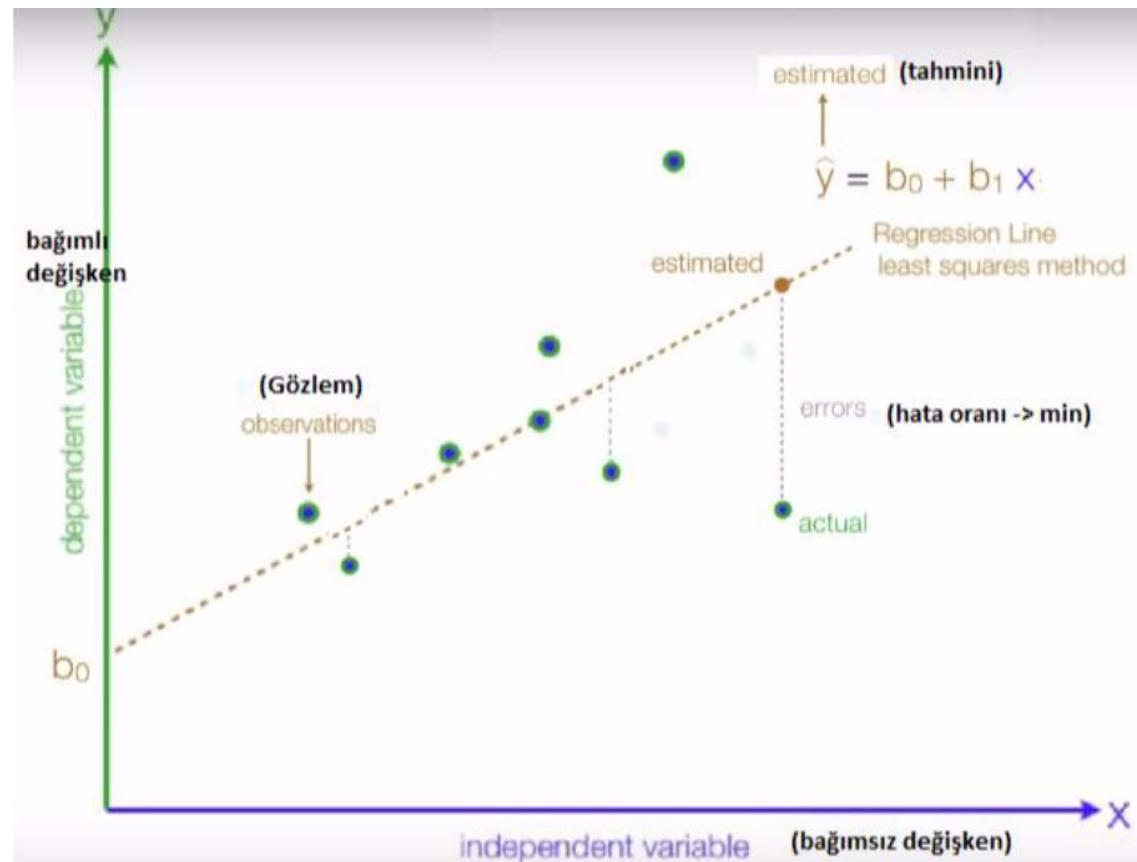
Veri Madenciliği

DR. ŞAFAK KAYIKÇI

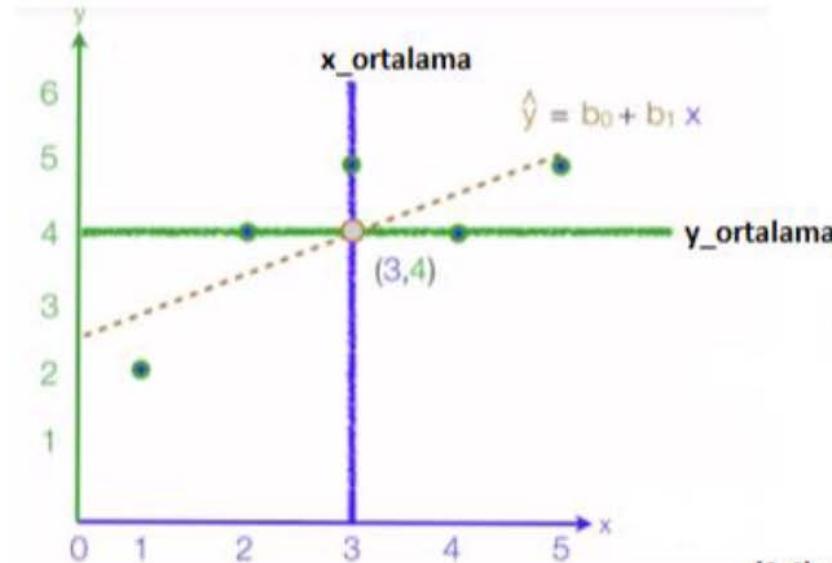
Lineer Regresyon – Logistik Regresyon

- regresyon = **bağlanım**
- Lineer regresyon, **nicel ve sürekli** veri arasındaki ilişkiyi özetleyen istatiksel bir metoddur. X ekseninde bağımsız değişkendir. Y eksemi ise tahmin edilen çıktı ise bağımlı değişkendir.
- Eğer tek bağımsız değişken var ise "Basit Doğrusal Regresyon" iki ve daha fazla bağımsız değişken var ise "Çoklu Doğrusal Regresyon" adı verilmektedir. Dereceli fonksiyonlara ise "Polinomsal Regresyon" adı verilmektedir.
- Logistik regresyon ise, **kategorik ve ayrık** veri arasındaki ilişkiyi özetleyen metotdur.

Lineer Regresyon



Lineer Regresyon -örnek



$$b_1 = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2} = \frac{6}{10} = .6$$

$$\hat{y} = b_0 + b_1 x$$

(3,4) ortalamar noktasından geçer

$$4 = b_0 + .6(3)$$

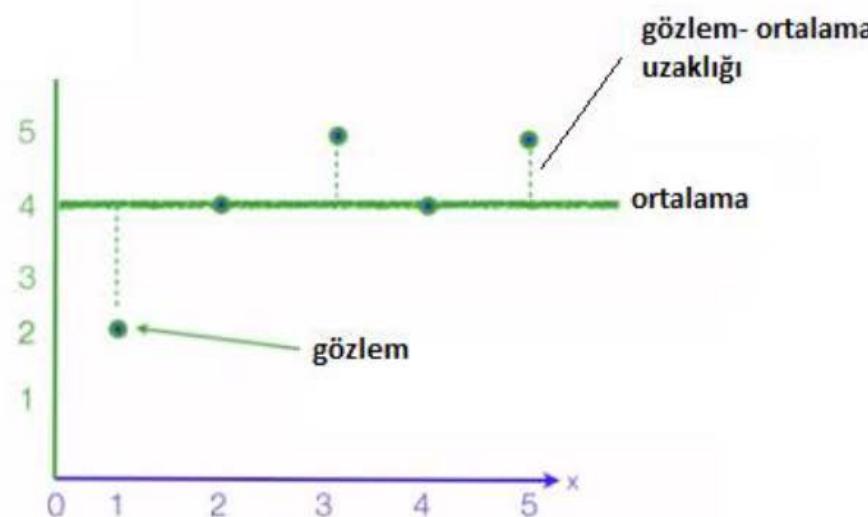
$$b_0 = 2.2$$

$$b_0 = 2.2$$

$$b_1 = .6$$

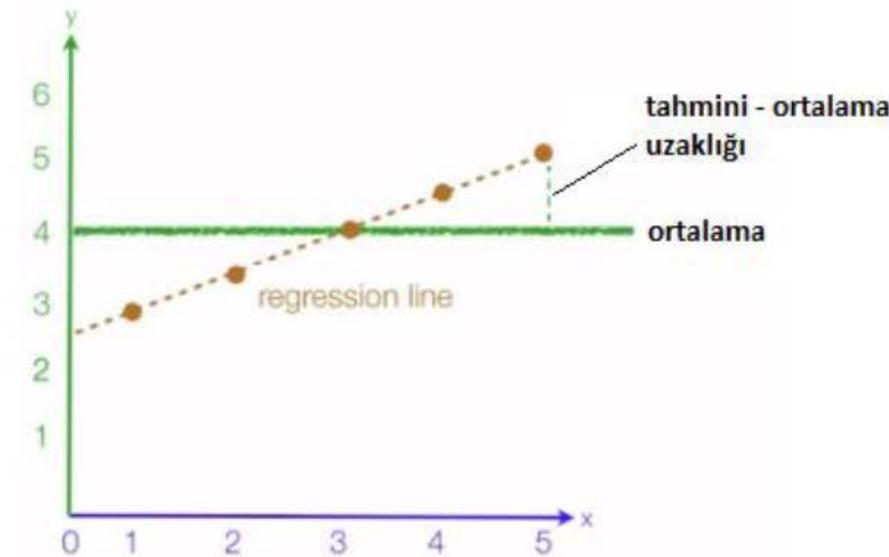
$$\hat{y} = 2.2 + .6x$$

Değerlendirme – R²



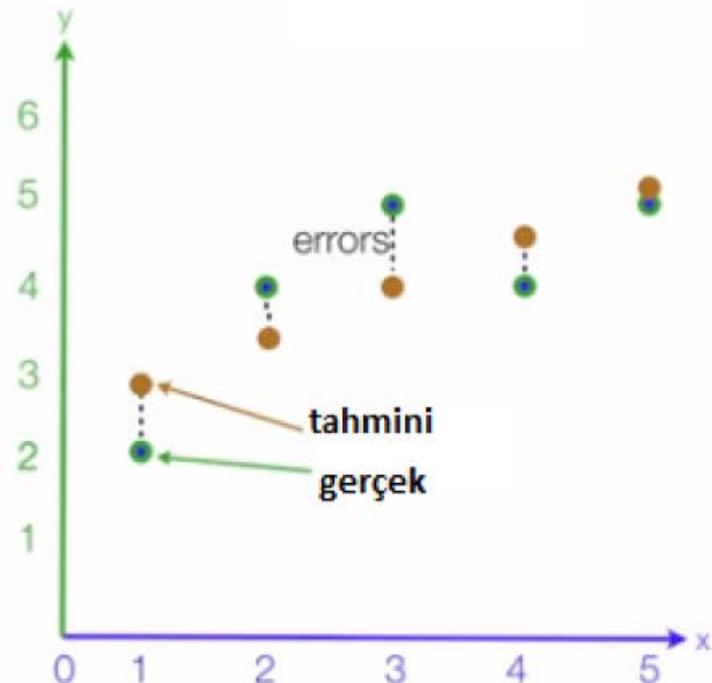
$$R^2 = \frac{\text{tahmini - ortalama uzaklığı}}{\text{gözlem - ortalama uzaklığı}}$$

$$R^2 = \frac{\sum (\hat{y} - \bar{y})^2}{\sum (y - \bar{y})^2} = \frac{3.6}{6} = .6$$



Değerlendirme - MSE

(Ortalama Karesel Hata) - Mean Square Error



$$\text{Standard Error of the Estimate} = \sqrt{\frac{\sum (\hat{y} - y)^2}{n - 2}}$$

$$= \sqrt{\frac{2.4}{3}} = \sqrt{.8} = .89$$

Lojistik Regresyon

odds (üstünlük) ve olasılık (p) → ?

		Kanser	
		E	H
Gen var mı?	E	23	117
	H	6	210

→ $\frac{23}{117}$ Gene sahip olan
kişinin, kanser üstünlüğü

→ $\frac{6}{210}$ Gene olmayan kişinin,
kanser üstünlüğü

Üstünlük oranlarının logaritması
log odds ratio → $\ln(\text{odds})$

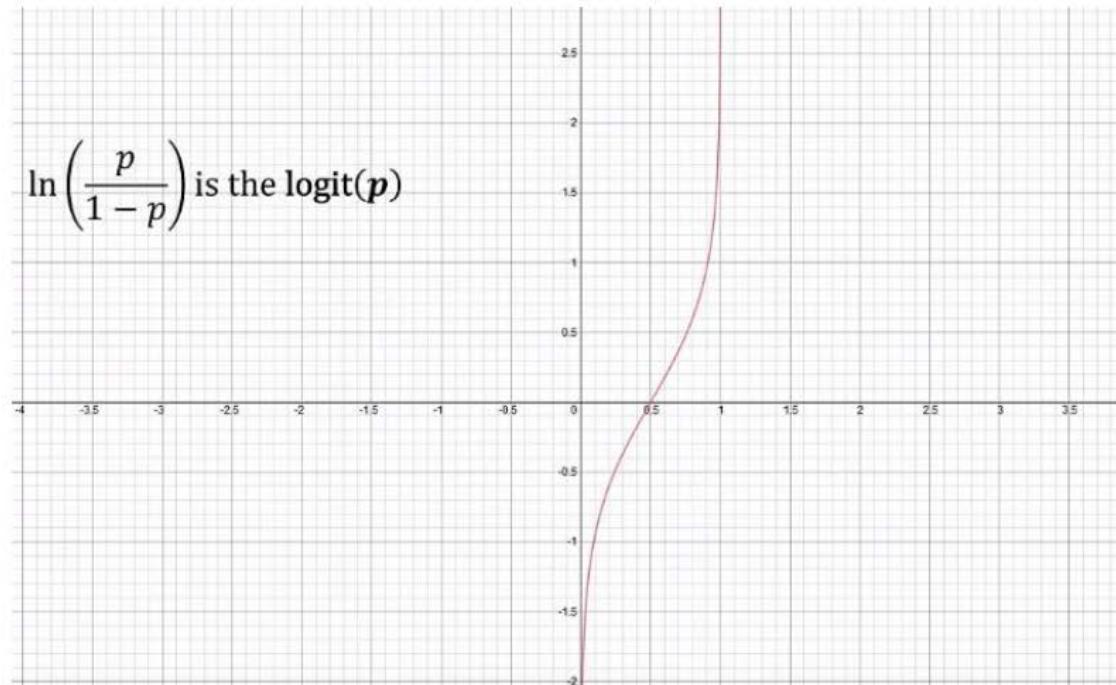
$$\frac{\frac{23}{117}}{\frac{6}{210}} = \frac{0.2}{0.03} = 6.88$$

$$\ln(6.88) = 1.93$$

- 1) Fisher's Exact Test
- 2) Chi-Square Test
- 3) The Wald Test

Lojistik Regresyon

$\ln(odds) \rightarrow \ln\left(\frac{p}{1-p}\right)$ is the **logit(p)** OR $\ln(p) - \ln(1-p) = \text{logit}(\mathbf{p})$



$$P = 0 \rightarrow \ln(0) = \text{tanımsız}$$

$$P = 1 \rightarrow \ln(\text{tanımsız}) = \text{tanımsız}$$

$$P = 0.5 \rightarrow \ln(1) = 0$$

Lojistik Regresyon

logit fonksiyonunda, 0-1 aralığındaki değerler X ekseninde.
Y ekseninde olması için, fonksiyonun tersini alıyoruz.

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right)$$

where p is between 0 and 1

$$\text{logit}^{-1}(\alpha) = \frac{1}{1 + e^{-\alpha}} = \frac{e^\alpha}{1 + e^\alpha}$$

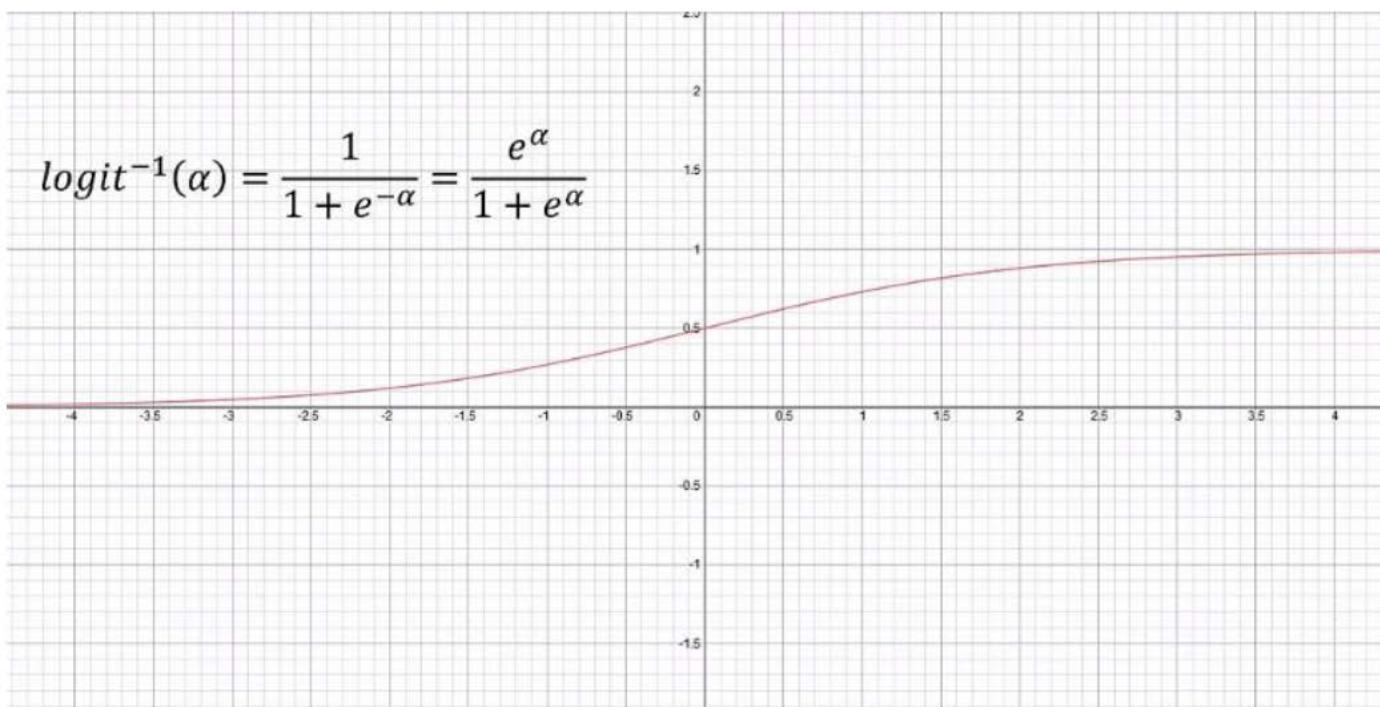
α = some number

Mean function:

(bazı kitaplarda)

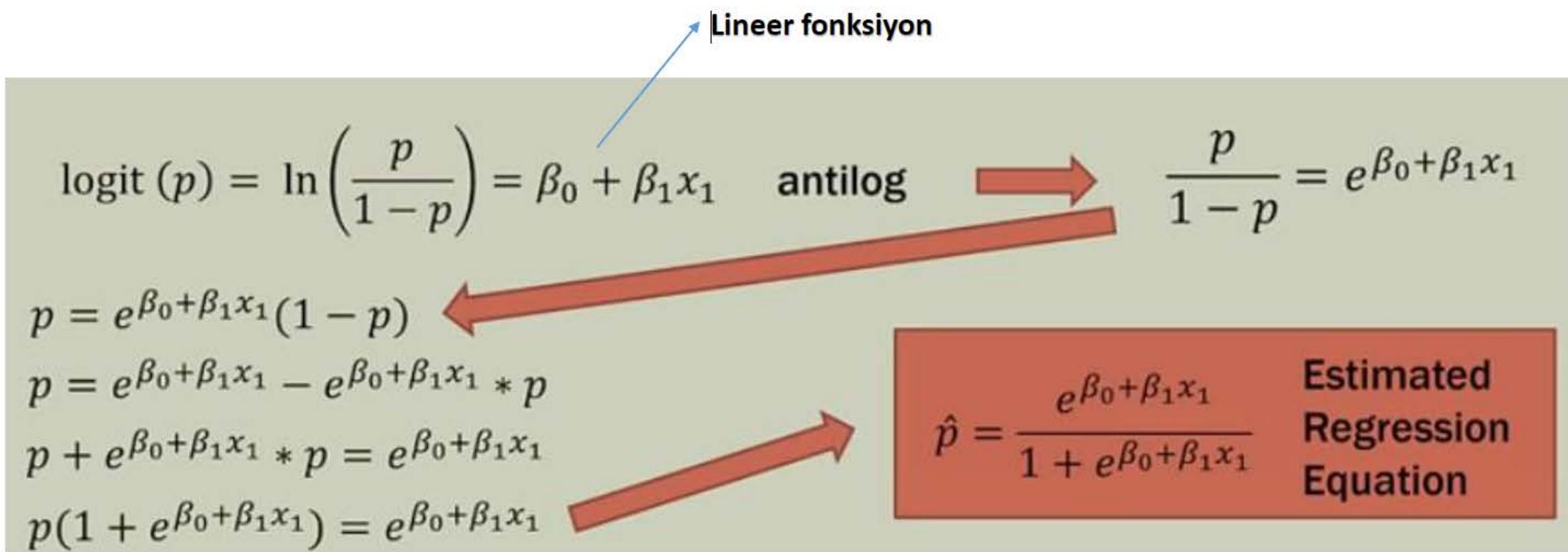
$$\mu_{y|x} = \frac{1}{1 + e^{-\alpha}} = \frac{e^\alpha}{1 + e^\alpha}$$

Lojistik Regresyon

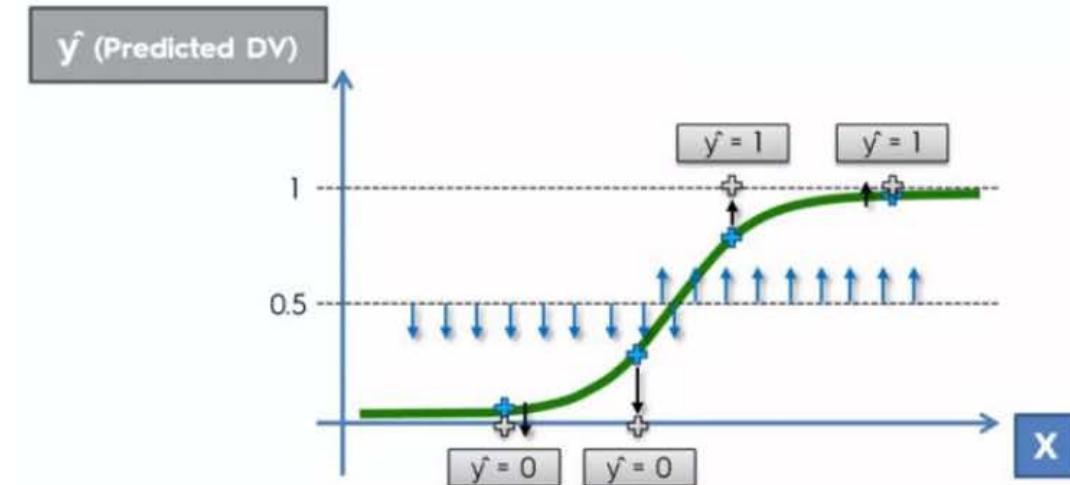
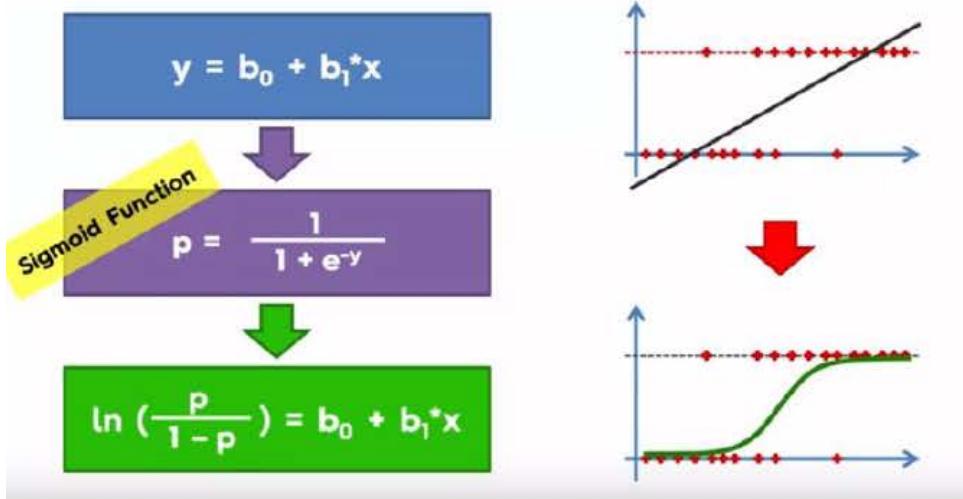


**sigmoid function curve
“S” curve**

Logistik Regresyon – tahmin denklemi



Logistik Regresyon - Özette



```
# boy  
151, 174, 138, 186, 128, 136, 179, 163, 152, 131  
  
# kilo  
63, 81, 56, 91, 47, 57, 76, 72, 62, 48
```

```
> x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)  
> y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)  
> relation <- lm(y~x)  
> print(relation)  
> print(summary(relation))
```

#170 boyundaki adamın tahmini kilosu

```
> a <- data.frame(x = 170)  
> result <- predict(relation,a)  
> print(result)
```

```
#grafik  
> plot(y,x,col = "blue",main = "Boy & Kilo Regresyon",  
+       abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Kilo (Kg)",ylab = "Boy (cm)"")
```

```
> input <- mtcars[,c("am","cyl","hp","wt")]
> print(head(input))
> am.data = glm(formula = am ~ cyl + hp + wt, data = input, family = binomial
)
> print(summary(am.data))
```

#Son sütundaki p değerinin "cyl" ve "hp" değişkenleri için 0,05'ten daha fazla olduğundan, "am" değişkeninin değerine katkıda bulunmalarının önemsiz olduğunu düşünüyoruz. Bu regresyon modelinde sadece ağırlık (ağırlık), "am" değerini etkiler.

Veri Madenciliği

DR. ŞAFAK KAYIKÇI

Bayes Sınıflandırıcılar

- Bayes Teoremi'ne (olasılık hesapları) dayanmaktadır.
- Her özellik (feature), birbirinden bağımsız olarak ele alınır. Bu yüzden 'naive' olarak adlandırılır.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Likelihood

Class Prior Probability

Posterior Probability

Predictor Prior Probability
(evidence)

The diagram illustrates the Bayes' Rule formula: $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$. The components are labeled as follows: Likelihood ($P(x|c)$), Class Prior Probability ($P(c)$), Posterior Probability ($P(c|x)$), Predictor Prior Probability (evidence) ($P(x)$), and the formula itself.

$$P(A|B) = \frac{P(A) P(B|A)}{P(B)}$$

$P(A|B)$ = B olayı gerçekleştiğinde A olayının gerçekleşme olasılığı
 $P(A)$ = A olayının gerçekleşme olasılığı
 $P(B|A)$ = A olayı gerçekleştiğinde B olayının gerçekleşme olasılığı
 $P(B)$ = B olayının gerçekleşme olasılığı

Avantaj - Dezavantaj

Avantaj

- Anlaşılması ve yapılması kolaydır
- Az sayıda veri seti ile bile kolay eğitilebilir
- Hızlıdır
- Alakasız özellikleri dikkate almaz

Dezavantaj

- Her özelliğin bağımsız olduğunu varsayar ama bu her zaman böyle değildir.

	Hava	Oyun
1	Güneşli	Hayır
2	Bulutlu	Evet
3	Yağmurlu	Evet
4	Güneşli	Evet
5	Güneşli	Evet
6	Bulutlu	Evet
7	Yağmurlu	Hayır
8	Yağmurlu	Hayır
9	Güneşli	Evet
10	Yağmurlu	Evet
11	Güneşli	Hayır
12	Bulutlu	Evet
13	Bulutlu	Evet
14	Yağmurlu	Hayır

Örnek -1

Hava	Hayır	Evet	Toplam		
Bulutlu	0	4	4	= 4 / 14	0,29
Yağmurlu	3	2	5	= 5 / 14	0,36
Güneşli	2	3	5	= 5 / 14	0,36
Toplam	5	9			
	= 5 / 14	= 9 / 14			
	0,36	0,64			

Soru : Oyuncunun yağmurlu günde oynama olasılığını bulunuz?

$$P(\text{Evet} | \text{Yağmurlu}) = \frac{P(\text{Evet}) * P(\text{Yağmurlu} | \text{Evet})}{P(\text{Yağmurlu})}$$

$$P(\text{Evet}) = 9 / 14 = 0,64$$

$$P(\text{Yağmurlu}) = 5 / 14 = 0,36$$

$$P(\text{Yağmurlu} | \text{Evet}) = 2 / 9 = 0,222$$

Böylece ,

$$P(\text{Evet} | \text{Yağmurlu}) = (0,64 * 0,222) / 0,36 = 0,39$$

Örnek -2

Meyve	Uzun	Tatlı	Sarı	Adet
Muz	400	350	450	500
Portakal	0	150	300	300
Diger	100	150	50	200
Toplam	500	650	800	1000

- Meyvelerin %50'si muz, %30 portakal, %20 diğer meyvelerdir.
- 500 muzun 400 (0.8) Uzun, 350 (0.7) Tatlı ve 450 (0.9) Sarıdır
- 300 portakalın 0 Uzun, 150 (0,5) Tatlı ve 300 (1) Sarıdır.
- Kalan 200 meyvenin 100 (0,5) Uzun, 150 (0,75) Tatlı ve 50 (0,25) Sarıdır

Meyve	Uzun	Uzun Değil	Tatlı	Tatlı Değil	Sarı	Sarı Değil	Toplam
Muz	400	100	350	150	450	50	500
Portakal	0	300	150	150	300	0	300
Diğer	100	100	150	50	50	150	200
Toplam	500	500	650	350	800	200	1000

Prior (önceki)

$$P(\text{Muz}) = 0.5 \ (500/1000)$$

$$P(\text{Portakal}) = 0.3$$

$$P(\text{Diğer}) = 0.2$$

Evidence (delil)

$$P(\text{Uzun}) = 0.5 \ (500 / 1000)$$

$$P(\text{Tatlı}) = 0.65$$

$$P(\text{Sarı}) = 0.8$$

Likehood (benzerlik)

$$P(\text{Uzun} | \text{Muz}) = 0.8 \ (400/500)$$

$$P(\text{Uzun} | \text{Portakal}) = 0$$

.....

$$P(\text{Sarı} | \text{Diğer}) = 0.25 \ (50/200)$$

$$P(\text{Sarı Değil} | \text{Diğer}) = 0.75 \ (150/200) \ \text{yada} \ (1-0.25)$$

Tahmin

uzun,tatlı ve sarı olan yeni meyve hangi sınıfına girer?

- P(Muz | Uzun,Tatlı,Sarı)

$$\begin{aligned} &= \frac{P(\text{Muz}) * P(\text{Uzun|Muz}) * P(\text{Tatlı|Muz}) * P(\text{Sarı|Muz})}{P(\text{Uzun}) * P(\text{Tatlı}) * P(\text{Sarı})} \\ &= \frac{0.5 * 0.8 * 0.7 * 0.9}{P(\text{Evidence})} \\ &= \frac{0.252}{P(\text{Evidence})} \end{aligned}$$

- P(Portakal | Uzun,Tatlı,Sarı) = 0
- P(Diger | Uzun,Tatlı,Sarı)

$$\begin{aligned} &= \frac{P(\text{Diger}) * P(\text{Uzun|Diger}) * P(\text{Tatlı|Diger}) * P(\text{Sarı|Diger})}{P(\text{Uzun}) * P(\text{Tatlı}) * P(\text{Sarı})} \\ &= \frac{0.2 * 0.5 * 0.75 * 0.25}{P(\text{Evidence})} \\ &= \frac{0,01875}{P(\text{Evidence})} \end{aligned}$$

0.252 > 0.01875 : Muz sınıfına dahil olur

```
> library("klaR")
> library("caret")
> data("iris")
> index = sample(nrow(iris), floor(nrow(iris) * 0.7)) #70/30 split.
> train = iris[index,]
> test = iris[-index,]
> xTrain = train[,-5]
> yTrain = train$Species
> xTest = test[,-5]
> yTest = test$Species
```

Aşağıdaki kod, 10 kat çapraz doğrulama kullanan bir Naive Bayes modeli üretecektir. x öznitelikler, y ise etiketlerdir. "nb", modele Naive Bayes'i kullanmalarını söyler.

```
> model = train(xTrain,yTrain,'nb',trControl=trainControl(method='cv',number=10))
> model

> table(predict(model$finalModel,xTest)$class,yTest)

> prop.table(table(predict(model$finalModel,xTest)$class,yTest))
```

Veri Madenciliği

DR. ŞAFAK KAYIKÇI

Özellikleri

- K-means en sık kullanılan kümeleme algoritmalarındandır. Uygulanması kolaydır.
 - Danışmansız öğrenme kategorisindedir.
 - Amaç : N adet veri nesnesinden oluşan bir veri kümesini giriş parametresi olarak verilen K adet kümeye bölmlemektir.
-
- Yöntem :
 1. Her kümenin merkez noktasını veya ortalamasını temsil etmek üzere K adet nesne rastgele seçilir.
 2. Kalan diğer nesneler, kümelerin ortalama değerlerine olan uzaklıkları dikkate alınarak en benzer oldukları kümelere dahil edilir.
 3. Her bir kümenin ortalama değeri hesaplanarak yeni küme merkezleri belirlenir.
 4. Tekrar nesnelerin merkeze uzaklıkları incelenir.
 5. Herhangi bir değişim olmayıncaya kadar algoritma tekrarlamaya devam eder.

Örnek

Örnek K=2 , uzaklık Öklid , rastgele ilk küme merkezleri(G1,G2)

	x	y
Gözlem 1	185	72
Gözlem 2	170	56
Gözlem 3	168	60
Gözlem 4	179	68
Gözlem 5	182	72
Gözlem 6	188	77

Centroid 1	Centroid 2	Atama
185	72	
0	21,9317122	1
21,9317122	0	2
20,80865205	4,472135955	2
7,211102551	15	1
3	20	1
5,830951895	27,65863337	1

Centroid 1	Centroid 2	Atama
183,5	72,25	
1,520690633	21,26029163	1
21,12610944	2,236067977	2
19,7563281	2,236067977	2
6,189709202	14,14213562	1
1,520690633	19,10497317	1
6,543126164	26,87005769	1

Örnek 2

Cluster the following eight points (with (x, y) representing locations) into three clusters A1(2, 10) A2(2, 5) A3(8, 4) A4(5, 8) A5(7, 5) A6(6, 4) A7(1, 2) A8(4, 9).

Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2).

The distance function between two points $a=(x_1, y_1)$ and $b=(x_2, y_2)$ is defined as: $\rho(a, b) = |x_2 - x_1| + |y_2 - y_1|$.

Örnek 2 :K=3 , uzaklık Manhattan, rastgele ilk küme merkezleri(A1,A4,A7)

Iteration 1

	Point	(2, 10)	(5, 8)	(1, 2)	
	Point	Dist Mean 1	Dist Mean 2	Dist Mean 3	Cluster
A1	(2, 10)	0	5	9	1
A2	(2, 5)	5	6	4	3
A3	(8, 4)	12	7	9	2
A4	(5, 8)	5	0	10	2
A5	(7, 5)	10	5	9	2
A6	(6, 4)	10	5	7	2
A7	(1, 2)	9	10	0	3
A8	(4, 9)	3	2	10	2

Cluster 1

(2, 10)

Cluster 2

(8, 4)

Cluster 3

(2, 5)

↑(5, 8)

(1, 2)

(7, 5)

(6, 4)

(4, 9)

Örnek 2 - devam

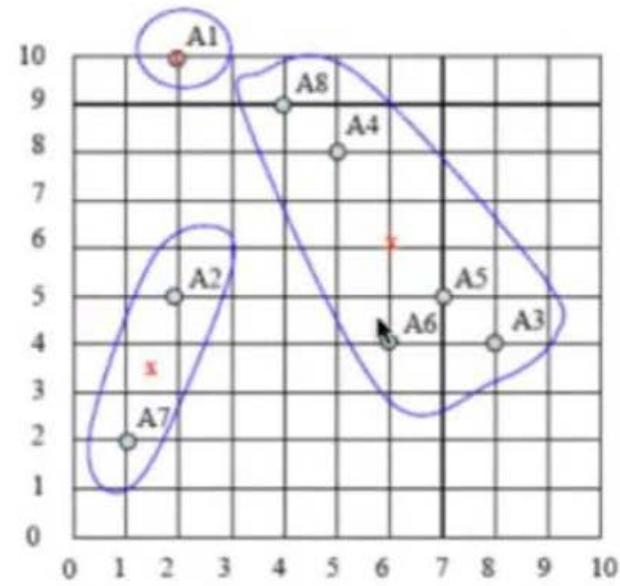
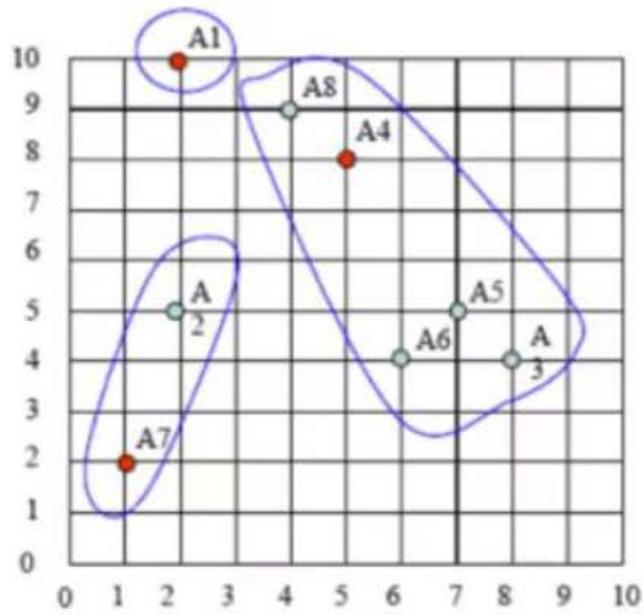
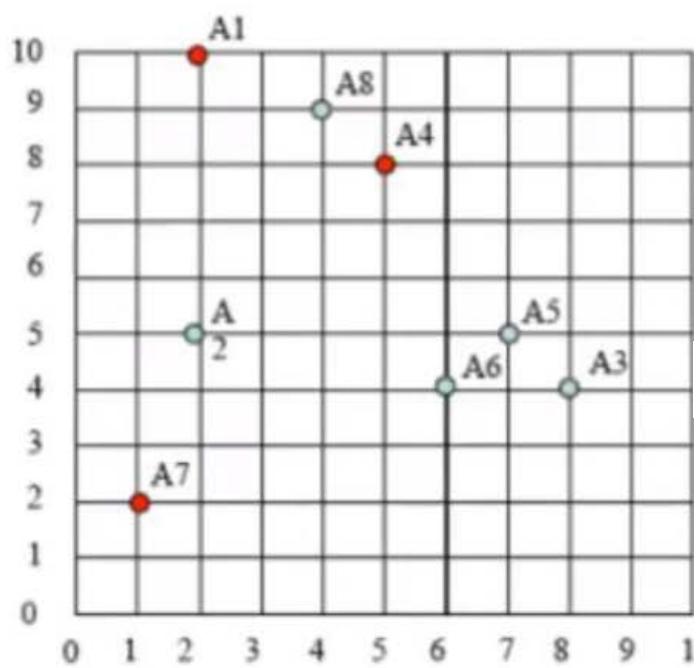
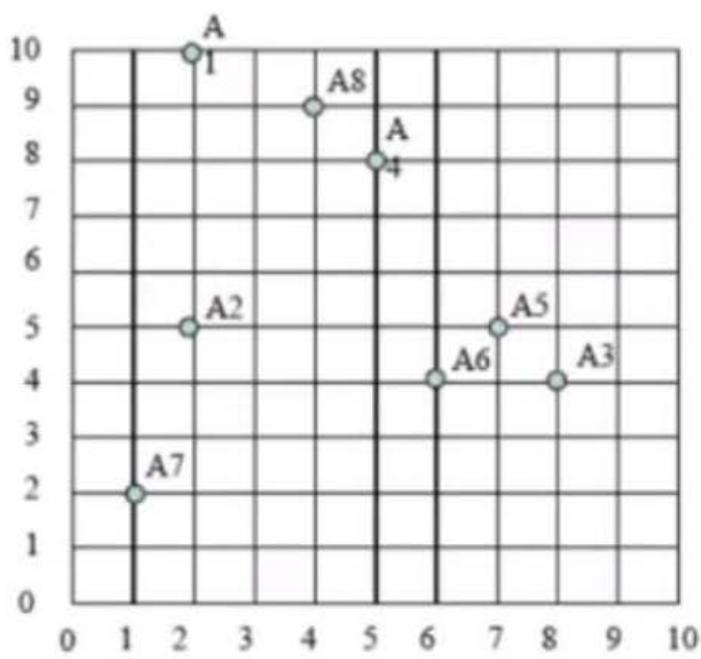
		(2, 10)	(6, 6)	(1.5, 3.5)	
	Point	Dist Mean 1	Dist Mean 2	Dist Mean 3	Cluster
A1	(2, 10)	0	8	7	1
A2	(2, 5)	5	5	2	3
A3	(8, 4)	12	4	7	2
A4	(5, 8)	5	3	8	2
A5	(7, 5)	10	2	7	2
A6	(6, 4)	10	2	5	2
A7	(1, 2)	9	9	2	3
A8	(4, 9)	3	5	8	1

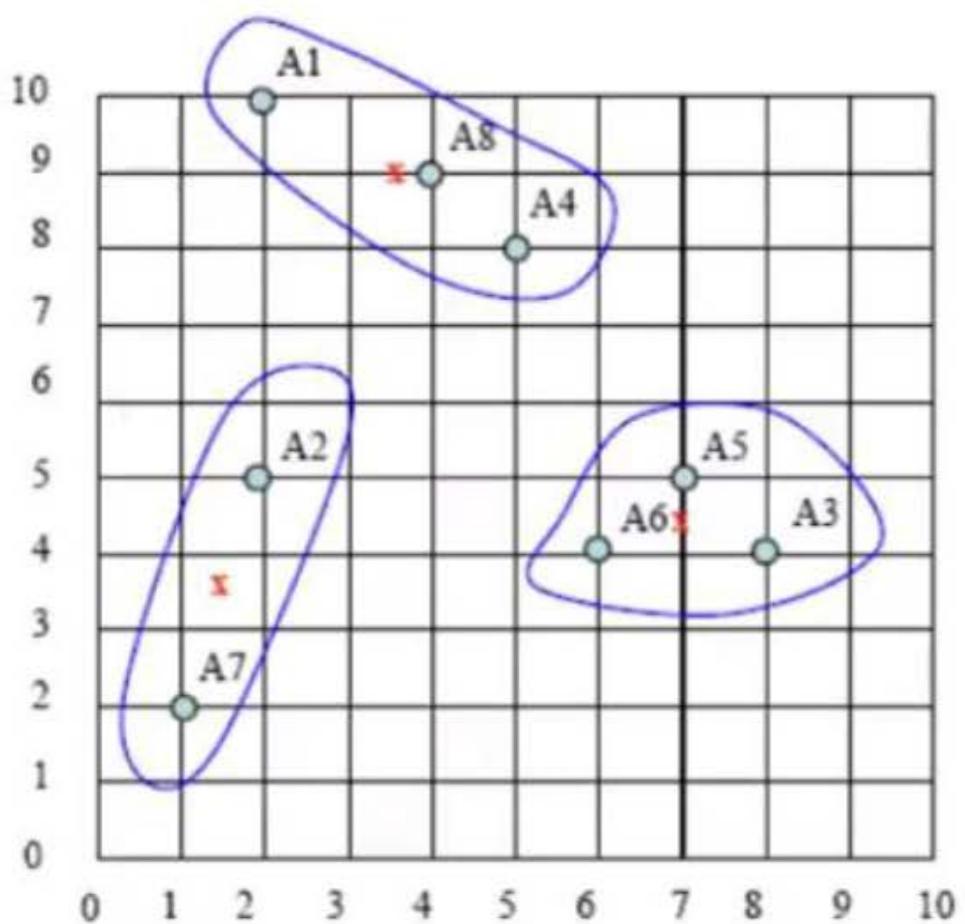
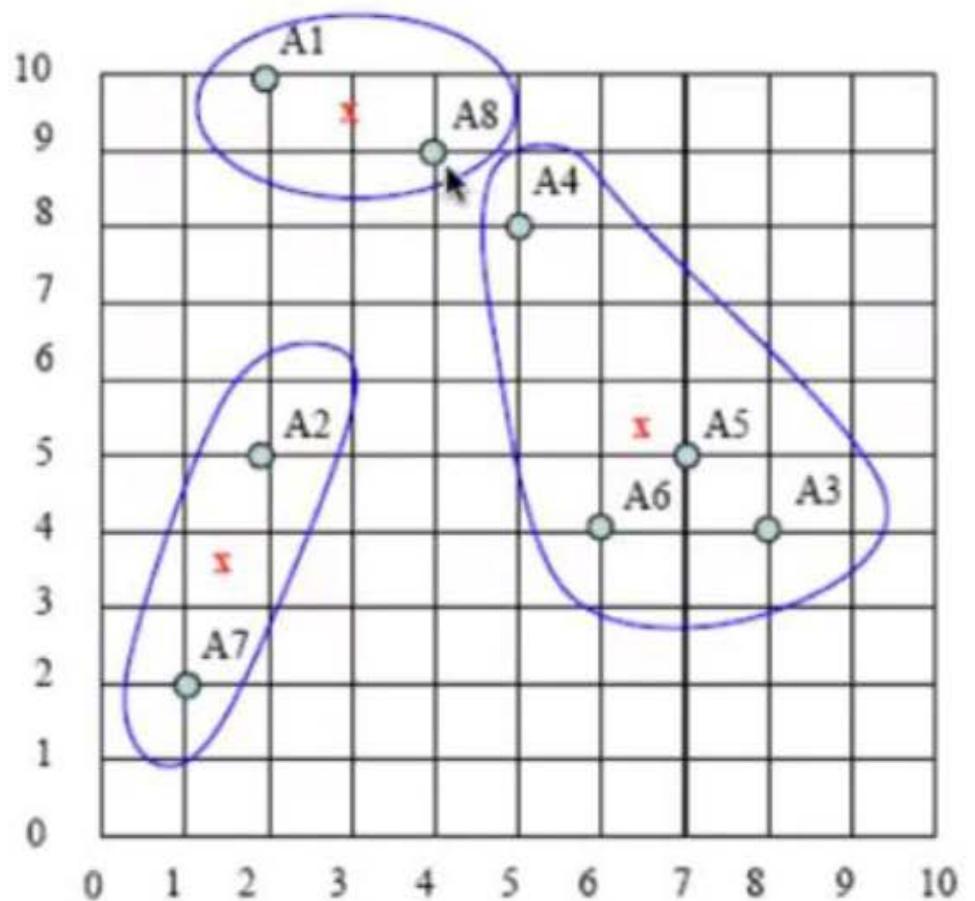
Örnek 2 - devam

		(3, 9.5)	(6.5 ,5.25)	(1.5, 3.5)	
	Point	Dist Mean 1	Dist Mean 2	Dist Mean 3	Cluster
A1	(2, 10)	1.5	9.25	7	1
A2	(2, 5)	5.5	4.75	2	3
A3	(8, 4)	10.5	2.75	7	2
A4	(5, 8)	3.5	4.25	8	1
A5	(7, 5)	8.5	0.75	7	2
A6	(6, 4)	8.5	1.75	5	2
A7	(1, 2)	9.5	8.75	2	3
A8	(4, 9)	1.5	6.25	8	1

Örnek 2 - devam

		(3.67, 9)	(7 ,4.3)	(1.5, 3.5)	
	Point	Dist Mean 1	Dist Mean 2	Dist Mean 3	Cluster
A1	(2, 10)	2.67	10.7	7	1
A2	(2, 5)	5.67	5.7	2	3
A3	(8, 4)	9.33	1.3	7	2
A4	(5, 8)	2.33	5.7	8	1
A5	(7, 5)	7.33	0.7	7	2
A6	(6, 4)	7.33	1.3	5	2
A7	(1, 2)	9.67	8.3	2	3
A8	(4, 9)	0.33	7.7	8	1





k-means animasyon



```

install.packages("ggplot2")
summary(iris)
library(ggplot2)
ggplot(iris,aes(x = Sepal.Length, y = Sepal.Width, col= Species)) + geom_point()
ggplot(iris,aes(x = Petal.Length, y = Petal.Width, col= Species)) + geom_point()

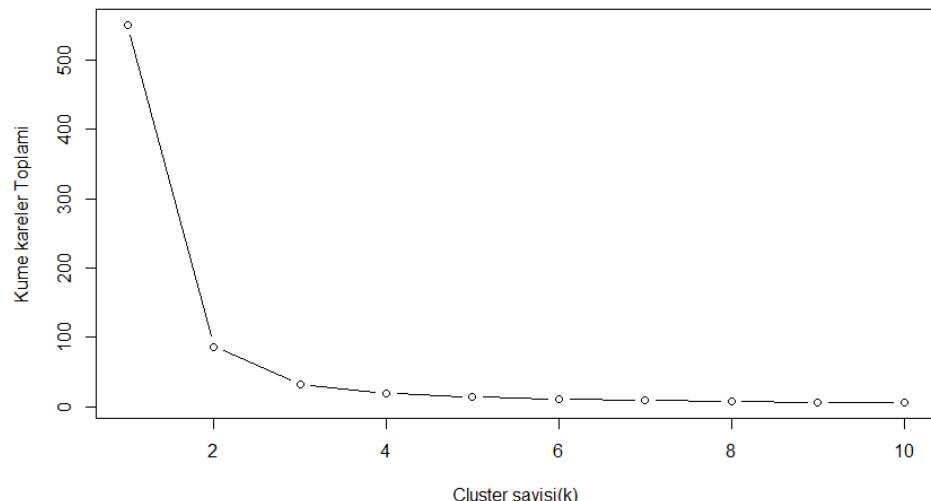
```

Optimum küme sayısının bulunması Küme içi küme kareleri toplamı vs küme grafiği, küme sayısı bize 3'te bir dirsek noktası gösterir. Bu nedenle, son modelin oluşturulmasında k için kullanılacak en iyi değerin 3 olduğunu söyleyebiliriz. Nstartvalue ayrıca 20 olarak tanımlanmıştır, bu R nin 20 farklı rastgele başlangıç ataması yapacaktır.

```

k.max = 10
sonuclar <- sapply(1:k.max,function(k){kmeans(iris[,3:4],k,nstart = 20,iter.max = 20)$tot.withinss})
sonuclar
plot(1:k.max,sonuclar, type= "b", xlab = "Cluster sayisi(k)", ylab = "Kume kareler Toplami")

```



```

icluster <- kmeans(iris[,3:4],3,nstart = 20)
table(icluster$cluster,iris$Species)

```

Veri Madenciliği

DR. ŞAFAK KAYIKÇI

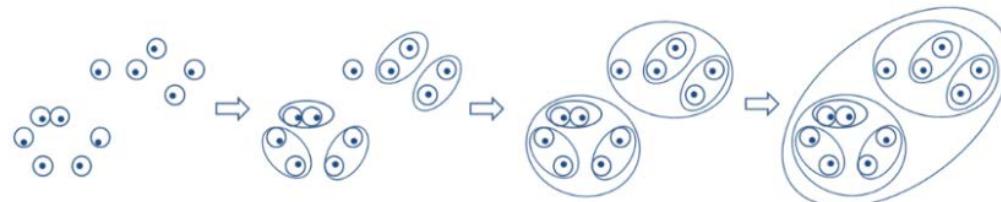
Hierarchical Clustering

- Küme sayısı önceden belli değildir

- İki çeşidi vardır :

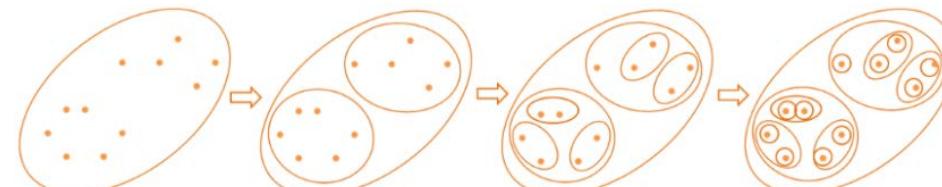
- birleştirici (bottom-up veya Agglomerative)

Birleştirici hiyerarşik kümelemede başlangıçta her bir değer bağımsız bir küme olarak değerlendirilir ve bu değerler çeşitli algoritmalarla birleştirilip her aşamada bir üst küme oluşturulur. En çok kullanılır.



- ayrıştırıcı (top-down veya Divisive)

Ayrıştırıcı kümelemede ise başlangıçta tüm değerler bir küme olarak değerlendirilip yine çeşitli ayrıştırma algoritmalarıyla alt kümeler elde edilir.



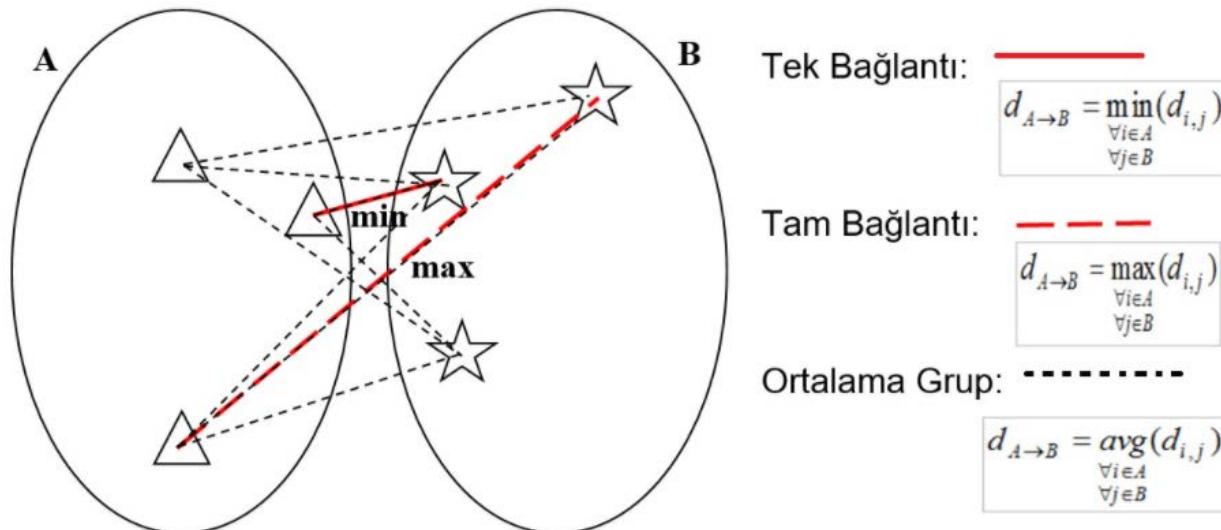
Mesafe Hesabı

Tek Bağlantı (Single Linkage) : İki küme arasındaki en yakın mesafeyi hesaplar. (en yakın komşu)

Tam Bağlantı (Complete Linkage): İki küme arasındaki en uzak mesafeyi hesaplar. (en uzak komşu)

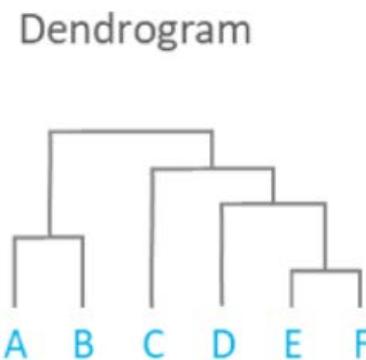
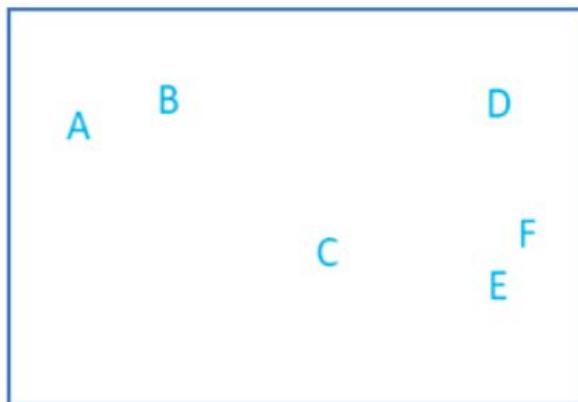
Ortalama Bağlantı (Average Linkage): İki küme arasındaki ortalama mesafeyi hesaplar.

Bunların dışında **ward**, **weighted**, **centroid** ve **median** yöntemleri vardır. **Seçilen yöntem sonucu etkiler.**



Dendogram

Dendrogram, benzer veri kümeleri arasındaki ilişkileri veya hiyerarşik kümelenmeyi gösteren bir ağaç diyagramıdır. Kaç tane küme oluşturacağımız bilgisini verir. **En uzun bacaktan çizilen yatay çizgi kümeye sayısını verir.**



1. E — F kümelenmiş (EF)
2. A — B kümelenmiş (AB)
3. D — EF kümelenmiş (DEF)
4. C — DEF kümelenmiş (CDEF)
5. AB — CDFE kümelenmiş (ABCDEF)

Örnek

Gönderilen ödevleri için kelime veya karakter benzerliği.

	1	2	3	4	5	6
1	0.000	0.458	0.177	0.419	0.4	0.161
2	0.458	0.000	0.486	0.263	0.263	0.462
3	0.177	0.486	0.000	0.000	0.300	0.004
4	0.419	0.263	0.300	0.000	0.000	0.276
5	0.419	0.263	0.300	0.000	0.000	0.276
6	0.161	0.462	0.004	0.276	0.276	0.000

	1	2	3	4	5	6
1						
2	0.458					
3	0.177	0.486				
4	0.419	0.263	0.300			
5	0.419	0.263	0.300	0.000		
6	0.161	0.462	0.004	0.276	0.276	

Birinci şekil ödevler arasındaki benzerlikleri göstermektedir. Örneğin 1 ve 2 nolu öğrenciler 0.458 oranında bir benzeşim bulunmaktadır. 2 – 1 nolu bakışta aynıdır, yanı simetriktir ve 1-1 nolu dokümanlar ise beklenildiği aynı dokümanlar olduğu için hiç bakmaya gerek yoktur. Bu noktada sadece şekil 2'deki veriler kullanılabilir.

Örnek

Matris üzerinde en küçük değer (maksimum benzerlik) 4 ve 5. öğrenci ödevlerinin benzerlik oranını gösteren 0 değeridir. Başka bir deyişle bu iki öğrencinin ödevleri tamamen aynıdır. 4 ve 5'ten yeni bir küme üretilebilir. Birleştirme işlemini yapılır ve yeni matrisimizi oluşturular.

	1	2	3	4	5	6
1						
2	0.458					
3	0.177	0.486				
4	0.419	0.263	0.300			
5	0.419	0.263	0.300	0.000		
6	0.161	0.462	0.004	0.276	0.276	

	1	2	3	4, 5	6
1					
2	0.458				
3	0.177	0.486			
4, 5	???	???	???	????	
6	0.161	0.462	0.004	???	

???: Algoritma tarafından belirlenecek değerler

Örnek

Bu örnekte min değerini kullanılacaktır. Min yani Tek Bağlantı hesabı:

$$d_{(4,5) \rightarrow 1} = \min(0.419, 0.419), \quad d_{(4,5) \rightarrow 2} = \min(0.263, 0.263), \quad d_{(4,5) \rightarrow 3} = \min(0.300, 0.300), \\ d_{(4,5) \rightarrow 6} = \min(0.276, 0.276)$$

Tek bağlantıya göre bu kümelerden en küçük değerler seçilir. İki ödev birbiriyle aynı olduğu için tüm benzerlikler aynı çıkmıştır.

İkinci döngüde, yeni oluşan matriste en küçük değer 0.004'tür. Başka bir deyişle 3 ve 6'inci öğrencilerden yeni bir küme oluşturulacaktır.

2. Döngü Başlangıç

	1	2	3	4, 5	6
1					
2	0.458				
3	0.177	0.486			
4, 5	0.419	0.263	0.300		
6	0.161	0.462	0.004	0.276	

2. Döngü Sonu

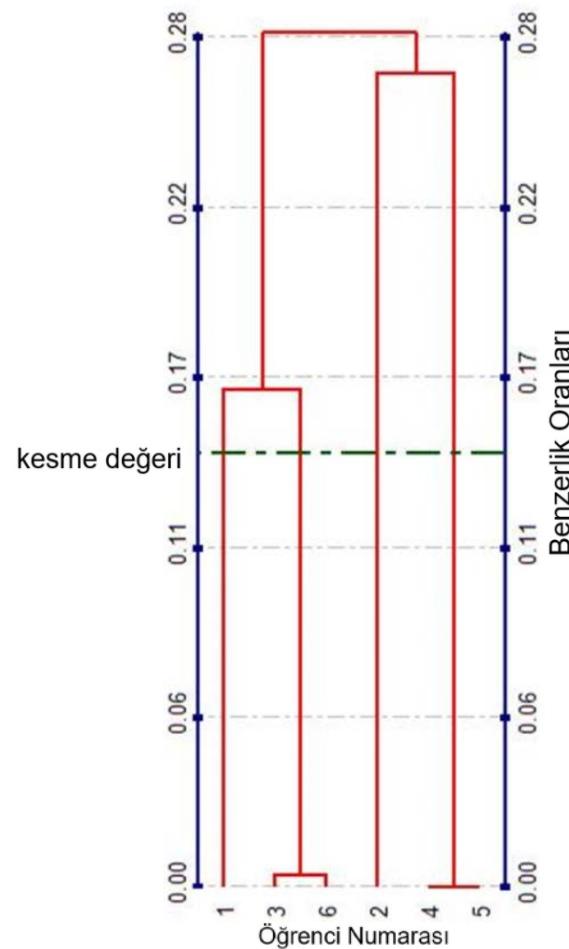
	1	2	3,6	4, 5
1				
2	0.458			
3, 6	0.161	0.462		
4, 5	0.419	0.263	0.276	

Örnek

Her döngünün sonunda küme sayısı 1 azalır ve küme sayısı 1 oluncaya kadar aynı işlemler tekrar eder. Her adım sonunda bir birleşim işlemi olur. Adım adım oluşan kümeler / ilişkileri ve seçilen minimum değerler aşağıda verilmiştir.

Adım	Kümeler	Minimum
0.	1, 2, 3, 4, 5, 6	-
1.	4, 5	0.000
2.	3, 6	0.004
3.	1, (3, 6)	0.161
4.	2, (4, 5)	0.263
5.	(1, (3, 6)), (2, (4, 5))	0.276

Örnek



Öncelikle (4, 5)'in 0 değeri çizilmiş ardından (3, 6)'nın 0.004 değeri çizilmiştir. Bu şekilde tüm kumeleme sonuçlarının çizimini yapılır.

Dendograma göre 4 ve 5 nolu öğrencilerin aynı ödevi gönderdiği 3 ile 6 nolu öğrencilerin ise çok az bir farklılık içerdigini görmektedir.

1 nolu öğrencinin 3 ve 6 nolu öğrencilere benzer bir ödev yaptığı görülmektedir. Bu benzerlik öğretim üyesi tarafından kopya oluşturacağı düşünülürse 0.161 büyük bir kesme değeri seçilebilir.

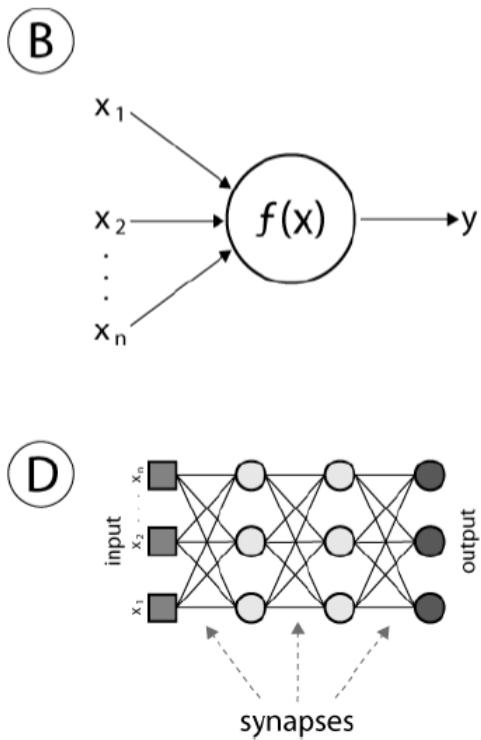
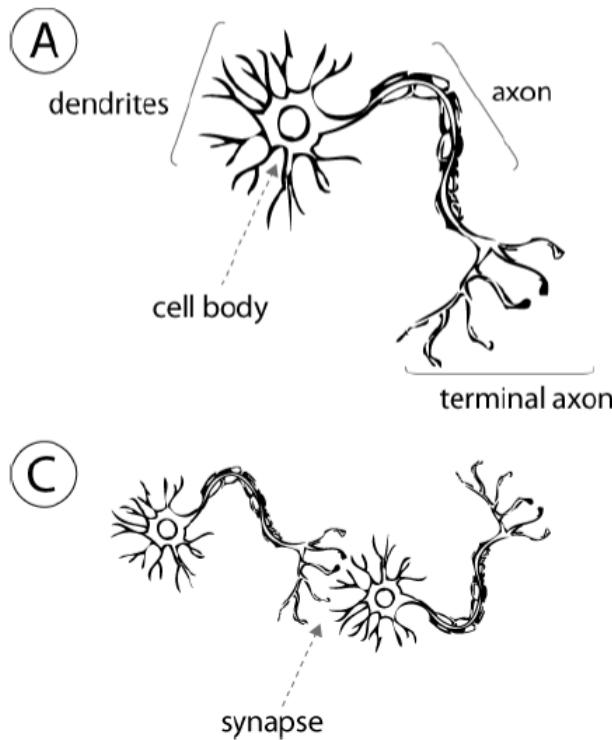
Eğer farklı bir ödev yaptığı düşüncesinde ise şekildeki gibi 0.15 kesme değerini seçebilir. Bu kesme değeri kullanıldığında 1, (3, 6), 2, (4, 5) olmak üzere 4 adet kume oluşturmaktadır. 1 ve 2 nolu öğrenciler orjinal ödev yaptığı görülmektedir.

```
#hclust, verileri bir mesafe matrisi biçiminde sağlamamızı gerektirir. Bunu dist kullanarak yapabiliriz.  
#Varsayılan olarak, tam bağlantı yöntemi kullanılır  
clusters <- hclust(dist(iris[, 3:4]))  
plot(clusters)  
  
#Toplam küme sayısı için en iyi seçeneklerin 3 veya 4 olduğunu şekilden görebiliyoruz  
#bunu yapmak için, cutree kullanarak ağaçları istenilen sayıda küme kesebiliriz  
clusterCut <- cutree(clusters, 3)  
table(clusterCut, iris$Species)  
  
#Bu sefer ortalama bağlantı yöntemini kullanalım  
clusters <- hclust(dist(iris[, 3:4]), method = 'average')  
plot(clusters)  
clusterCut <- cutree(clusters, 3)  
table(clusterCut, iris$Species)  
  
library(ggplot2)  
ggplot(iris, aes(Petal.Length, Petal.Width, color = iris$Species)) +  
  geom_point(alpha = 0.4, size = 3.5) + geom_point(col = clusterCut) +  
  scale_color_manual(values = c('black', 'red', 'green'))  
#İç rengin dış renkle eşleşmediği tüm noktalar yanlış kümelenmiş noktalardır.
```

Veri Madenciliği

DR. ŞAFAK KAYIKÇI

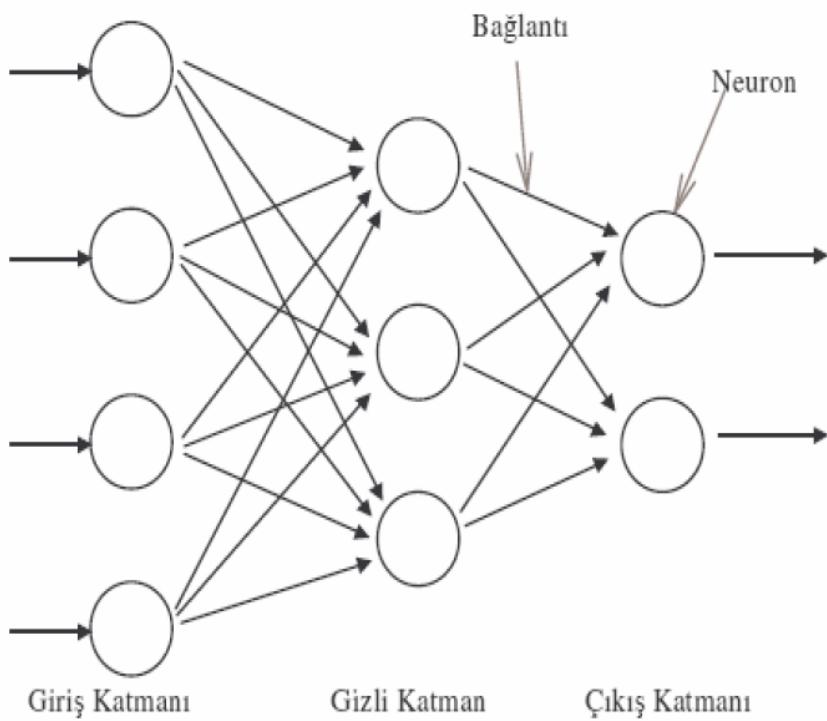
Yapay Sinir Ağları



YSA'lar, insan beyninin çalışma prensibi örnek alınarak geliştirilmeye çalışılmıştır ve aralarında yapısal olarak bazı benzerlikler vardır.

Sırasıyla nöronun sinapsisleri tarafından ağırlandırılmış giriş sinyallerini toplamak için bir toplayıcı, burada açıklanan işlemler bir doğrusal birleştirici oluşturmaktadır.

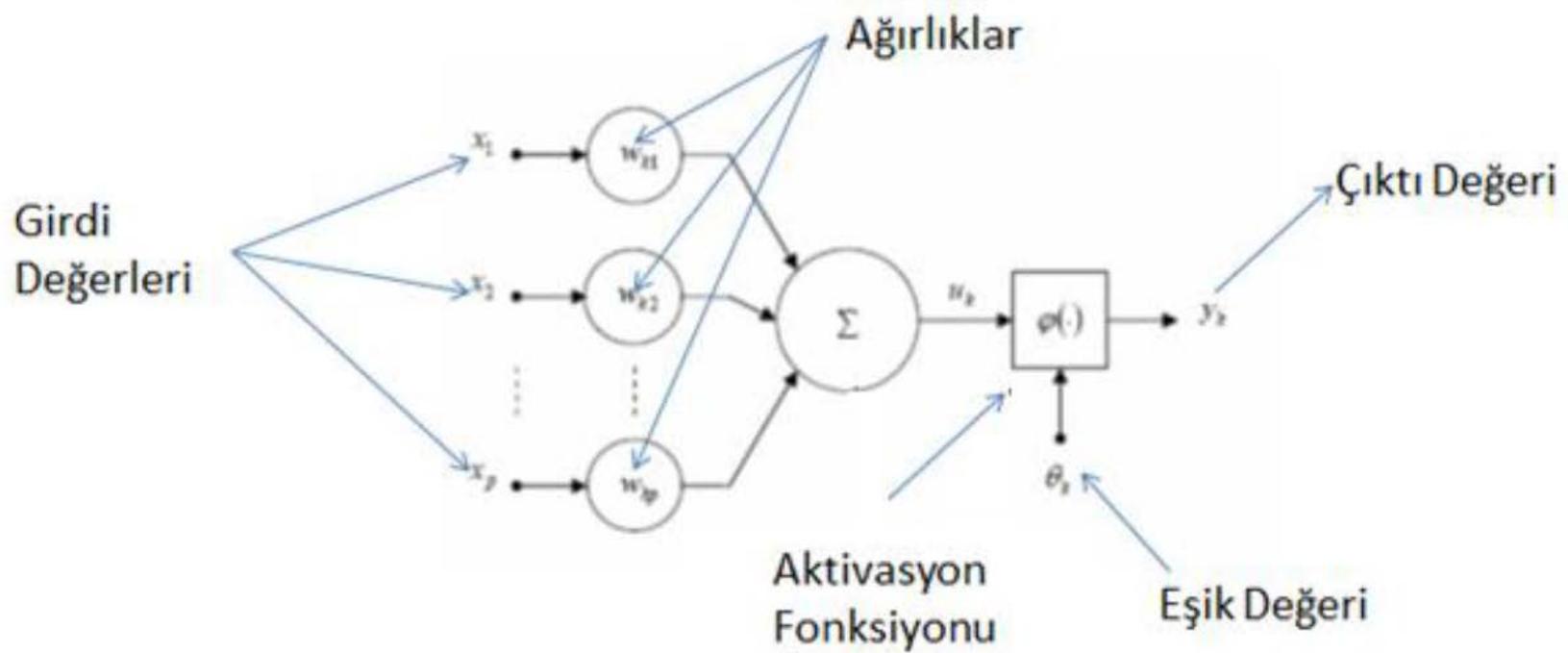
Yapay Sinir Ağları



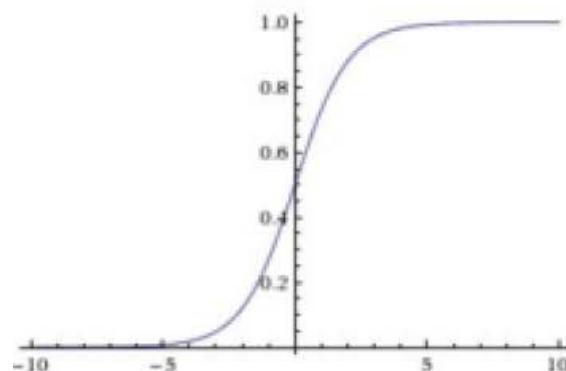
Katmanların değişik şekilde bir birleriyle bağlanması değişik ağ mimarilerini doğurur. YSA'lar üç katmadan oluşur. Bu katmanlar sırasıyla;

- a) **Girdi Katmanı:** Bu katmandaki proses elemanları dış dünyadan bilgileri alarak ara katmanlara transfer ederler. Bazı ağlarda girdi katmanında herhangi bir bilgi işleme olmaz.
- b) **Ara Katman (Gizli Katman) :** Girdi katmanından gelen bilgiler işlenerek çıktı katmanına gönderilirler. Bu bilgilerin işlenmesi ara katmanlarda gerçekleştirilir. Bir ağ içinde birden fazla ara katman olabilir.
- c) **Çıktı Katmanı:** Bu katmandaki proses elemanları ara katmandan gelen bilgileri isleyerek ağır girdi katmanından sunulan girdi seti için üretmesi gereken çıktıyı üretirler. Üretilen çıktı dış dünyaya gönderilir.

YAPISI



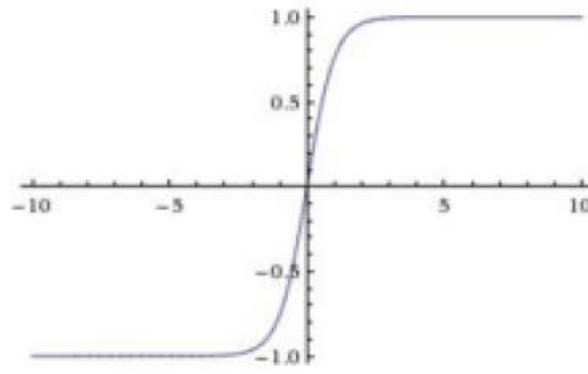
Aktivasyon Fonksiyonları



Sigmoid

$$\sigma(x) = 1 / (1 + \exp(-x))$$

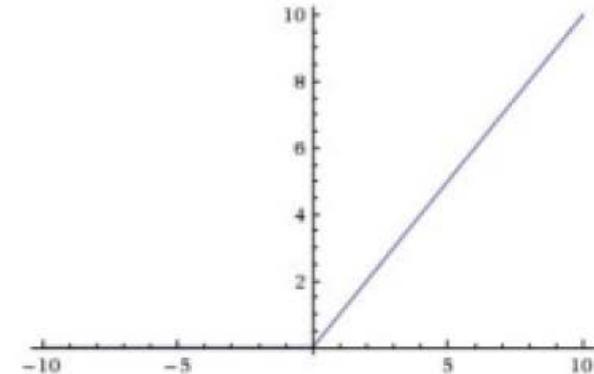
(0,1)



tanh

$$\tanh(x) = 2\sigma(2x) - 1$$

[-1,1]



ReLU

$$f(x) = \max(0, x)$$

Negatif değerleri 0 yapar

Eğitilmesi

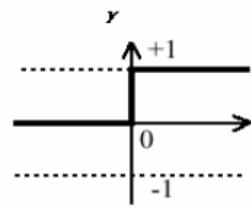
1. Ağa girdi setini ve ona karşılık olarak beklenen çıktı gösterilir (X, B). Birden fazla girdi değeri olabilir. Yani $X = (x_1, x_2, x_3, \dots, x_N)$ gibi. Çıktı değeri ise 1 ve 0 değerlerinden birini alır.

2. NET girdi hesaplanır : $NET = \sum_{i=1}^m w_i x_i$

3. Çıktı değeri hesaplanır.

$$\zeta = 1 \rightarrow Net > \phi$$

$$\zeta = 0 \rightarrow Net \leq \phi$$



Eşik aktivasyon fonksiyonu

4. Eğer gerçekleşen çıktı ile beklenen çıktı aynı ise nernangı bir değişiklik olmaz, öğrenme tamamlanır. Değilse ağırlıklar, bir öğrenme katsayısı oranında tekrar artırılır yada azaltılır.

$$w_n = w_0 + \lambda$$

$$w_n = w_0 - \lambda$$

Örnek

1.örnek: $X_1 = (x_1, x_2) = (1, 0)$, $B_1 = 1$

2.örnek: $X_2 = (x_1, x_2) = (0, 1)$, $B_2 = 0$

Ağırlıklar: $W = (w_1, w_2) = (1, 0)$

Eşik Değeri : $\phi = -1$

Öğrenme Katsayısı : $\lambda = 0.5$

- 1.iterasyon, 1.örnek ağa gösterilir

$$NET = w_1 * x_1 + w_2 * x_2 = 1 * 1 + 0 * 0 = 1$$

NET > ϕ (-1) olduğundan çıktı $C = 1$ olur.

$C = B_1$ (1) (beklenen değer olduğundan ağırlıklar değişmez)

- 2.iterasyon, 2.örnek ağa gösterilir

$$NET = w_1 * x_1 + w_2 * x_2 = 1 * 0 + 0 * 1 = 0$$

NET > ϕ olduğundan çıktı $C = 1$ olur.

$C \neq B_2$ (0) (beklenen değerden büyük, ağırlıklar azaltılır)

$$W_1 = w_1 - \lambda x_1 = 1 - 0.5 * 0 = 1$$

$$W_2 = w_2 - \lambda x_2 = 0 - 0.5 * 1 = -0.5$$

Örnek

- 3.iterasyon, 1.örnek ağa gösterilir

$$\text{NET} = w_1 * x_1 + w_2 * x_2 = 1 * 1 + (-0.5) * 0 = 1$$

NET > $\emptyset(-1)$ olduğundan çıktı $\zeta = 1$ olur.

$\zeta = B_1(1)$ (beklenen değer olduğundan ağırlıklar değişmez)

- 4.iterasyon, 2.örnek ağa gösterilir

$$\text{NET} = w_1 * x_1 + w_2 * x_2 = 1 * 0 + (-0.5) * 1 = -0.5$$

NET > $\emptyset(-1)$ olduğundan çıktı $\zeta = 1$ olur.

$\zeta \neq B_2(0)$ (beklenen değerden büyük, ağırlıklar azaltılır)

$$W_1 = w_1 - \lambda x_1 = 1 - 0.5 * 0 = 1$$

$$W_2 = w_2 - \lambda x_2 = -0.5 - 0.5 * 1 = -1$$

Örnek

- 5.İterasyon, 1.örnek ağa gösterilir

$$\text{NET} = w_1 * x_1 + w_2 * x_2 = 1 * 1 + (-1) * 0 = 1$$

NET > 0 (-1) olduğundan çıktı Ç = 1 olur.

Ç = B1 (1) (beklenen değer olduğundan ağırlıklar değişmez)

- 6.İterasyon, 2.örnek ağa gösterilir

$$\text{NET} = w_1 * x_1 + w_2 * x_2 = 1 * 0 + (-1) * 1 = -1$$

NET = 0(-1) olduğundan çıktı Ç = 0 olur.

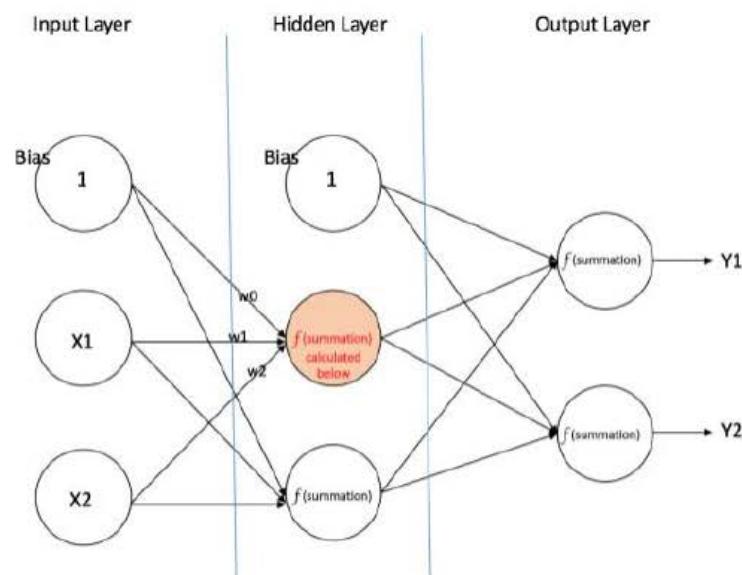
Ç = B2 (0) (beklenen değer olduğundan ağırlıklar değişmez)

W1 = 1

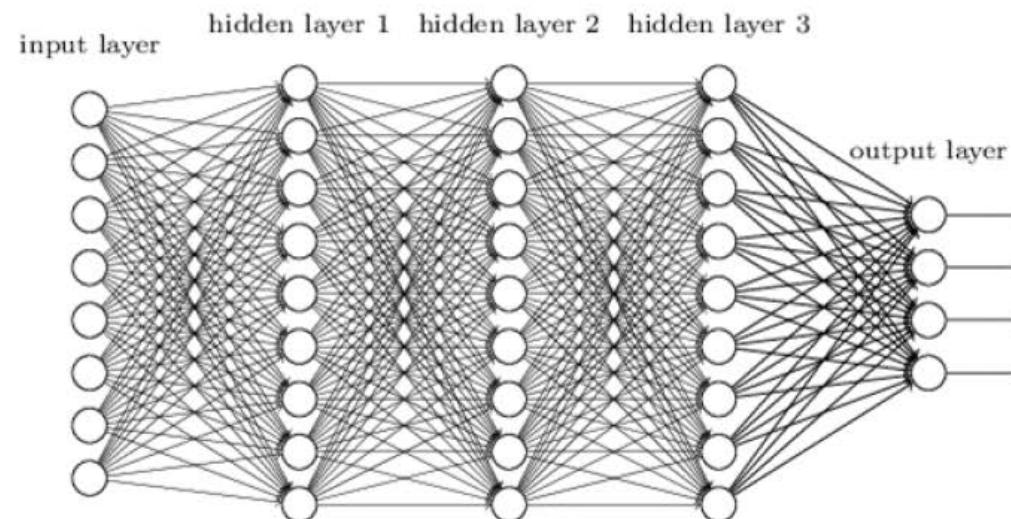
W2 = -1

Bu aşamadan sonra, ağırlıklar değiştirilmez. Sistem iterasyona girse de beklenen değerlerde sonuçlar vereceğinden öğrenme aşaması tamamlanmış olur.

Daha ilerisi...



Deep neural network



```
install.packages("neuralnet")
library("neuralnet")
data("iris")
iris.dataset <- iris
iris.dataset$setosa <- iris.dataset$Species=="setosa"
iris.dataset$virginica <- iris.dataset$Species == "virginica"
iris.dataset$versicolor <- iris.dataset$Species == "versicolor"
train <- sample(x = nrow(iris.dataset), size = nrow(iris)*0.5)
iristrain <- iris.dataset[train,]
irisvalid <- iris.dataset[-train,]
nn <- neuralnet(setosa+versicolor+virginica ~ Sepal.Length + Sepal.Width, data=iristrain, hidden=3, rep = 2, err.fct = "ce", linear.output = F, lifesign = "minimal", stepmax = 1000000)
plot(nn, rep="best")
comp <- compute(nn, irisvalid[-3:-8])
pred.weights <- comp$net.result
idx <- apply(pred.weights, 1, which.max)
pred <- c('setosa', 'versicolor', 'virginica')[idx]
table(pred, irisvalid$Species)
```

Veri Madenciliği

DR. ŞAFAK KAYIKÇI

Support Vector Machine (SVM)

Destek Vektör Makineleri

SVM, hem sınıflandırma hem de regresyon problemleri için kullanılan denetimli bir makine öğrenme algoritmasıdır.

SVM, hem doğrusal ayrılabilir veriler hem de doğrusal olmayan ayrılabilir veriler için kullanılır.

Doğrusal olmayan veriler için çekirdek fonksiyonları (kernel functions) kullanılır.

Doğrusal ayrılabilir veriler

1. Tek Boyutlu Uzay



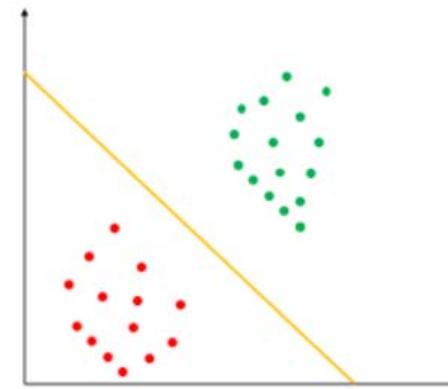
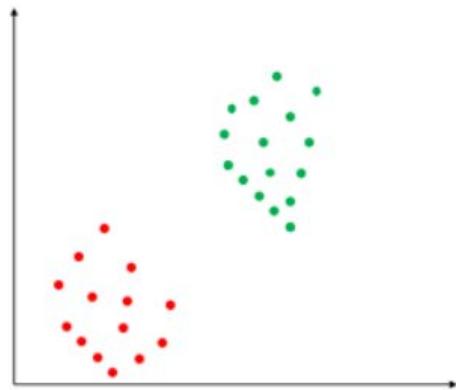
Kırmızı ve yeşil olmak üzere iki sınıfımız olduğunu varsayıyalım ve iki sınıf aracını ayıran bir sınır bulabilirsek, bunun doğrusal olarak ayrılabilir veri olduğu söylenir.



Burada, iki sınıf arasında sınır görevi gören bir nokta bulabiliz. Belirli bir noktanın solundaki veri noktaları kırmızı sınıfa ve bu belirli noktanın sağındaki yeşil sınıfa aittir.

Doğrusal ayrılabilir veriler

2. İki Boyutlu Uzay



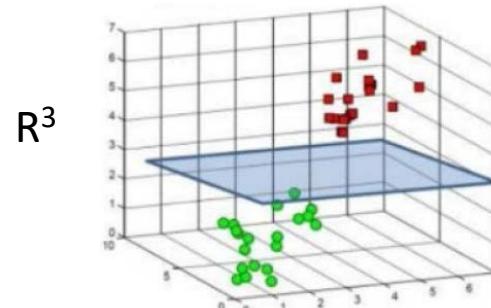
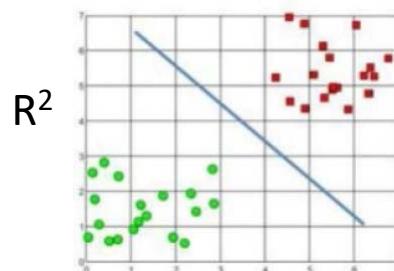
Benzer şekilde, iki boyutlu uzayda, iki sınıf arasında sınır görevi gören bir çizgi bulabiliriz.

Doğrusal ayrılabilir verilerde SVM

SVM, bir hiper düzlem (**hyperplane**) kullanarak iki sınıfı sınıflandırır. Hiper düzlemler, verileri sınıflandıran karar sınırlarıdır.

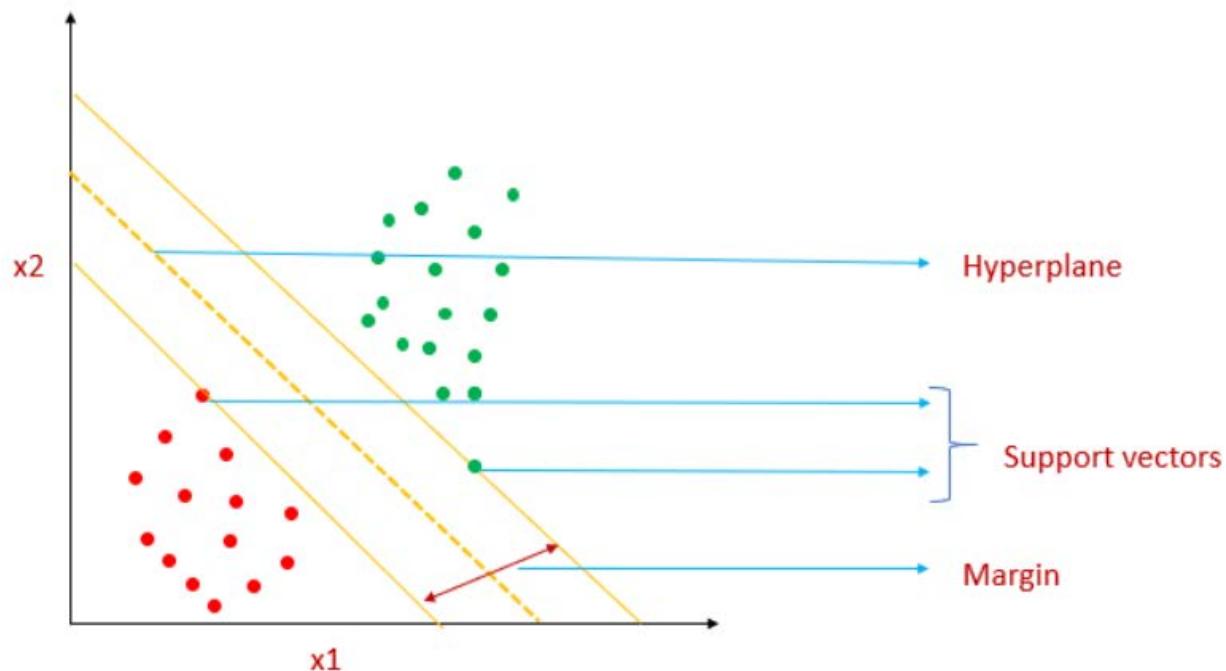
Bir hiper düzlem, n boyutlu bir uzayda n-1 boyutlu bir alt uzaydır.

- SVC, tek boyutlu bir uzayda tek bir noktadır
- SVC, iki boyutlu bir uzayda tek boyutlu bir çizgidir.
- SVC, üç boyutlu bir uzayda iki boyutlu bir düzlemdir.
- Veri noktaları üçten fazla boyutta olduğunda, SVC bir hiper düzlemdir.



SVM

SVM algoritması, her iki tarafta da maksimum marjla iki sınıfın ortasında giden en iyi hiper düzlemi bulur.



Terimler

Destek Vektörleri : Her iki sınıfı da hiper düzleme en yakın veri noktaları destek vektörleri olarak bilinir. Destek vektörü olan bir veri noktası kaldırılırsa, hiper düzlemin konumu değiştirilir. Destek vektörü olmayan bir veri noktası kaldırılırsa, model üzerinde hiçbir etkisi olmaz.

Marjin : Hiper düzlem ile destek vektörleri arasındaki mesafe, kenar boşluğu olarak bilinir.

Sabit Marj (Hard Marjin) : Sabit Marj, her iki sınıfı da hiper düzlemin her iki tarafında bulunan veri noktalarının marjin içinde olmadığı anlamına gelir.

Yumuşak Marj (Soft Marjin) : Veri noktaları doğrusal olarak ayrılabilir değilse, kenar boşluğu içinde bazı veri noktalarına izin verilir. Bu, soft marjin olarak bilinir.

SVM

SVM'nin arkasındaki temel ilke, iki sınıfı ayıran maksimum marjlı bir hiper düzlem çizmektir. İki boyutlu uzayda SVM kullanarak C1 ve C2 sınıflarını ayırmak istediğimizi varsayıyalım. Daha sonra bilinmeyen özellik vektörü X 'in hangi sınıfta olacağını tahmin edelim.

$$g(X) = w^T X + b = 0$$

Yukarıdaki doğrusal bir denklemde:

$w \rightarrow$ hiper düzleme dik olan ağırlık vektörünü belirtir. d -boyutlu uzayda hiper düzlemin yönünü temsil eder. Burada d özellik vektörünün boyutluluğudur.

$b \rightarrow$ d boyutlu uzayda hiper düzlemin konumunu temsil eder.

İki boyutlu bu doğrusal denklem, düz bir çizгиyi temsil eder. Üç boyutlu uzayda bir düzlemi ve 3'ten fazla boyutta bir alt düzlemi temsil eder.

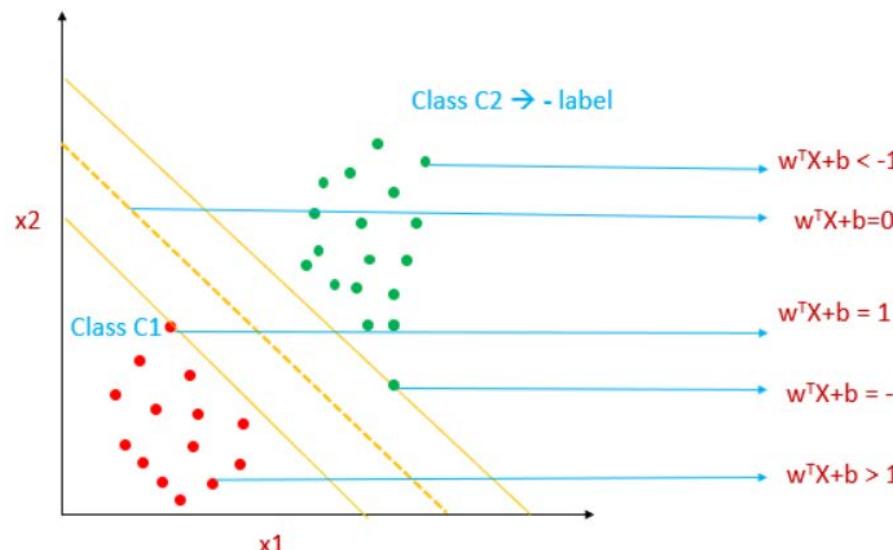
SVM

1. Özellik vektörü, hiper düzlemin pozitif tarafında yer alıyorsa : $g(X_i)=w^T X_i + b > 0$

2. Özellik vektörü, hiper düzlemin negatif tarafında yer alıyorsa: $g(X_i)=w^T X_i + b < 0$

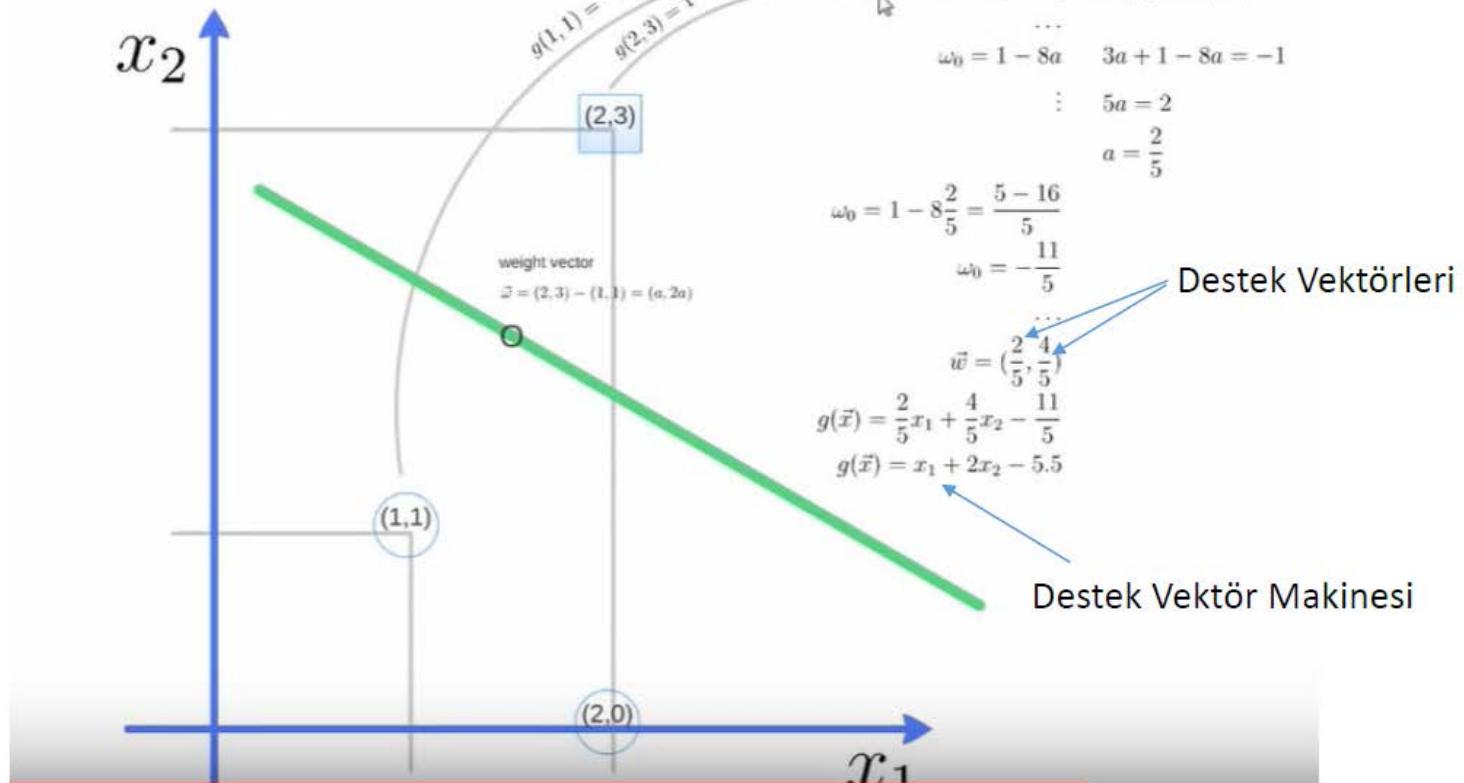
Sınıflandırma kuralı: $g(X_i)=w^T X_i + b > 0 \rightarrow X_i \in \text{Class C1}$

$g(X_i)=w^T X_i + b < 0 \rightarrow X_i \in \text{Class C2}$



Örnek

Example



SVM çalışması

■ Bilinmeyen öznitelik vektörünün değeri nasıl tahmin edilir?

Eğitim aşaması sırasında SVM, en iyi hiper düzlemi bulur ve o hiper düzlem için w ve b 'yi hesaplar. Bilinmeyen öznitelik vektörünün sınıfını bulmak için, SVM bu vektörün değerini denklemde uygulayarak tahmin eder. Değer negatifse, Sınıf C2'ye [Negatif etiket] aittir veya değer pozitifse Sınıf C1'e [pozitif etiket] aittir.

■ En iyi hiper düzlem nasıl bulunur?

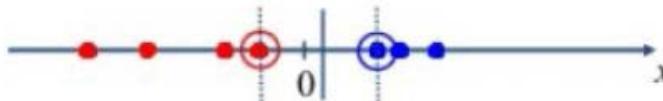
Eğitim aşaması sırasında, rastgele bir alt düzlemle başlanır ve bir hata olup olmadığını kontrol edilir. C1 sınıfına ait bir veri noktası C2 sınıfı olarak tahmin edilirse, m 'nin değerini değiştirilir ve hiper düzlemi, hata veri noktası doğru tarafa geri donecek şekilde döndürülür. Eğitim aşamasında, model sıfır eğitim hatası veren doğru w ve b 'yi bulacaktır.

■ Marji nasıl maksimize edilir?

İki sınıfı ayıran tüm olası hiper düzlemi bulduktan sonra, her bir hiper düzlem için w ve d 'yi hesaplanır. w , hiper düzlemin eğim vektörünü gösterir. d , en yakın veri noktasının alt düzleme olan mesafesini temsil eder. d değerini sıraladıktan sonra, her iki sınıfın en yakın veri noktasından maksimum mesafeye sahip olan hiper düzlemi seçilir.

Non Linear SVM

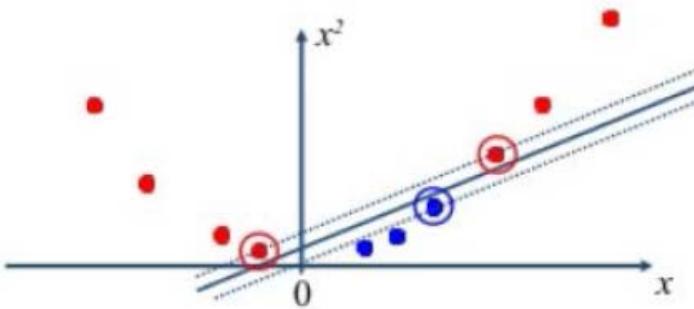
- Veri kümeleri lineer olarak ayrılabilir.



- Fakat veri kümeleri çok keskin(hard) ise ne yapacağız?



- Daha yüksek boyutlu uzayda veriyi haritalayabiliriz (ifade edebiliriz).



```
library("e1071")
plot(iris)
plot(iris$Sepal.Length, iris$Sepal.Width, col=iris$Species)
plot(iris$Petal.Length, iris$Petal.Width, col=iris$Species)
s<-sample(150, 100)
col<-c("Petal.Length", "Petal.Width", "Species")
iris_train<-iris[s,col]
iris_test<-iris[-s,col]
svmfit <- svm(Species ~., data = iris_train, kernel = "linear", cost = .1, scale = FALSE)
print(svmfit)
plot(svmfit, iris_train[,col])

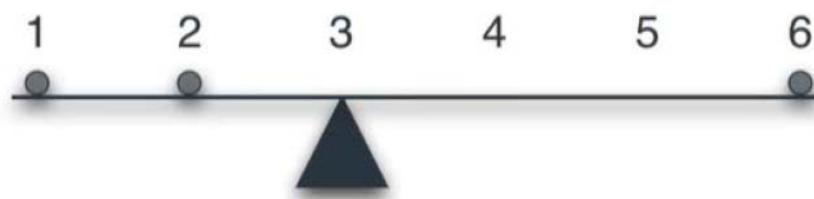
tuned <- tune(svm, Species ~., data = iris_train, kernel = "linear", ranges =
list(cost=c(0.001,0.01,.1,1,10,100)))
summary(tuned)

p <- predict(svmfit, iris_test[,col], type="class")
plot(p)
table(p, iris_test[,3])
mean(p== iris_test[,3])
```

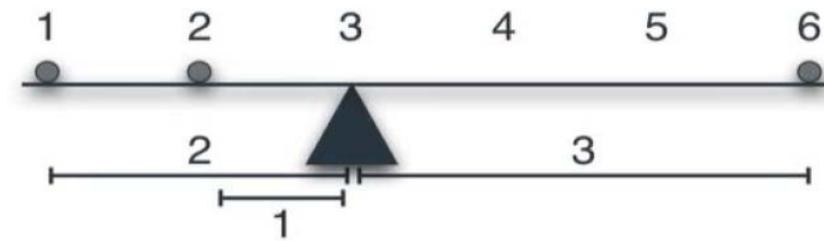
Veri Madenciliği

DR. ŞAFAK KAYIKÇI

Ortalama (mean) – Varyans (variance)

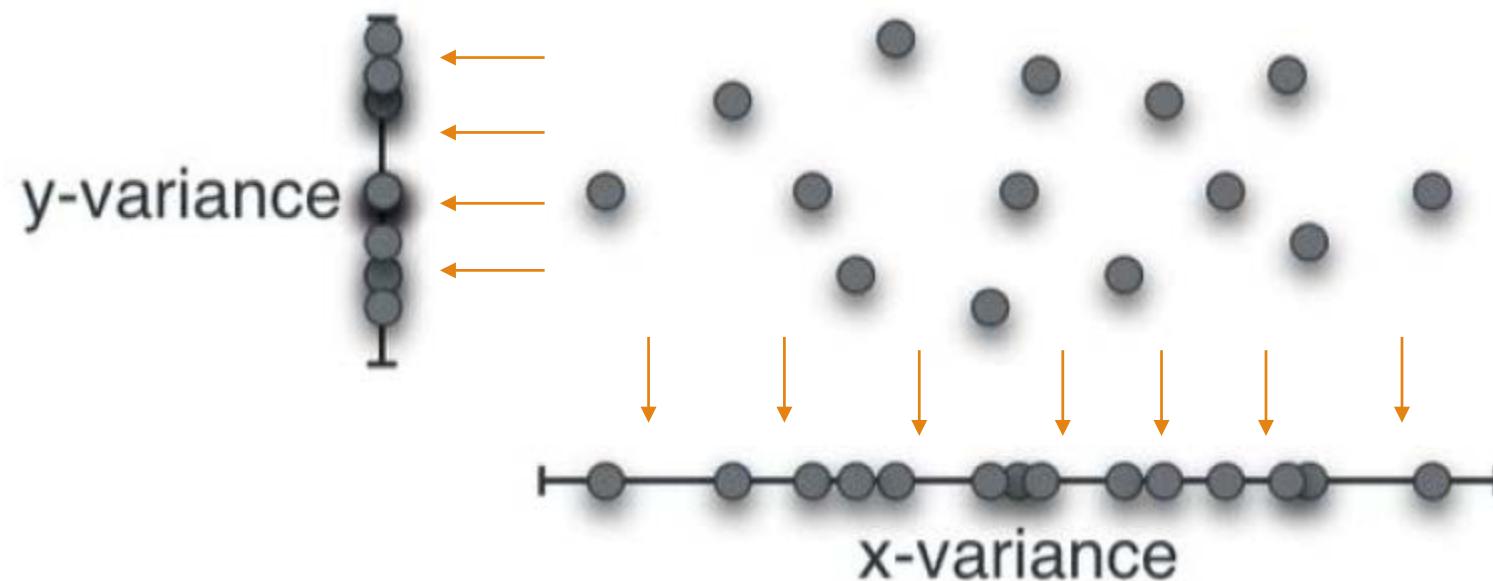


$$\text{Ortalama} = \frac{1+2+6}{3} = 3$$

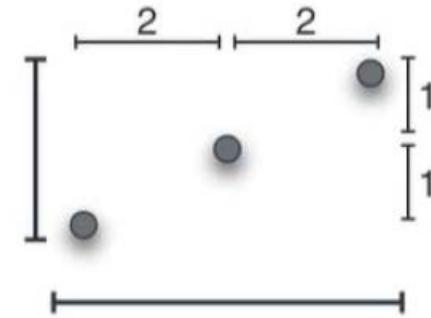
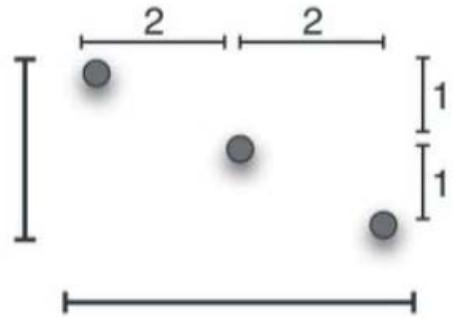


$$\text{Variance} = \frac{2^2 + 1^2 + 3^2}{3} = 14/3$$

x varyans – y varyans



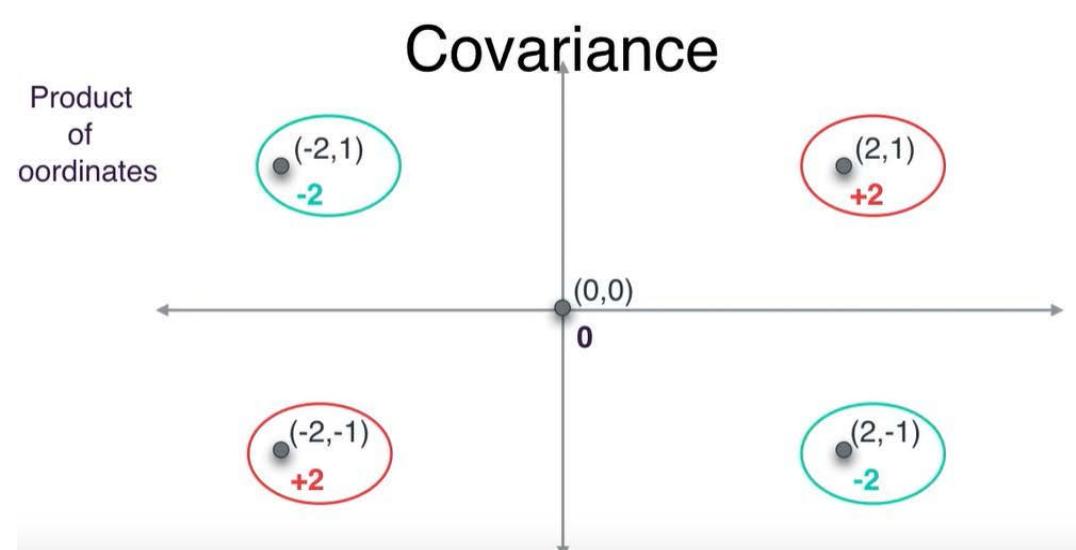
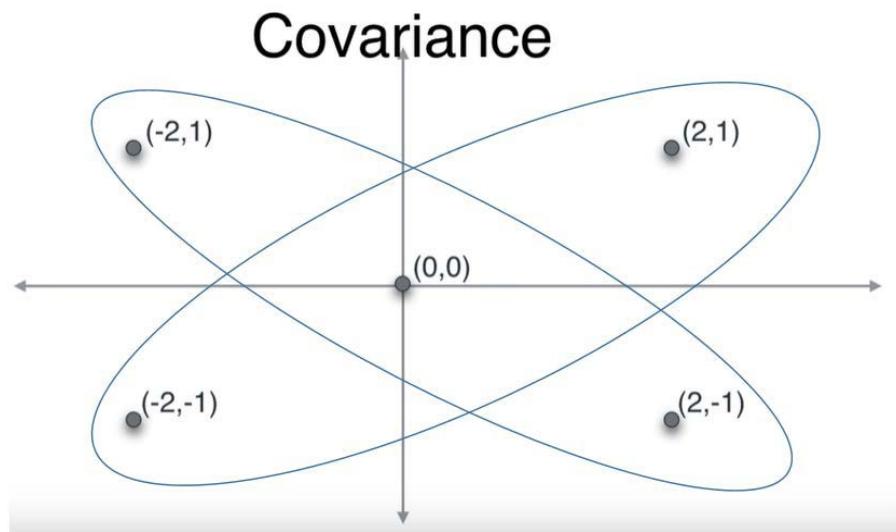
x varyans – y varyans



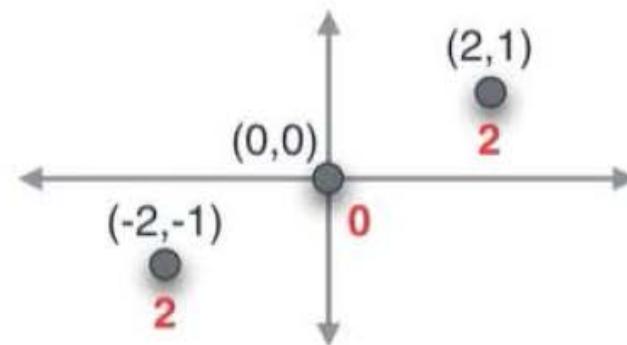
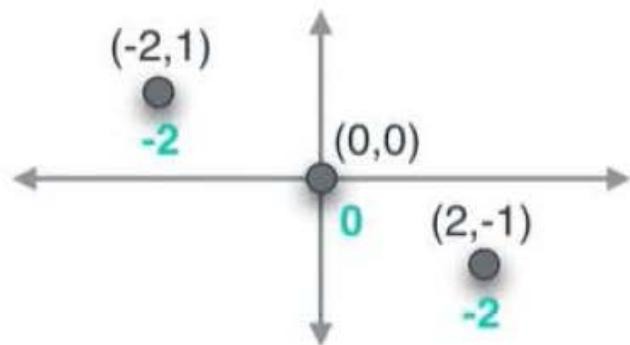
$$x\text{-variance} = \frac{2^2+0^2+2^2}{3} = 8/3$$

$$y\text{-variance} = \frac{1^2+0^2+1^2}{3} = 2/3$$

Kovaryans



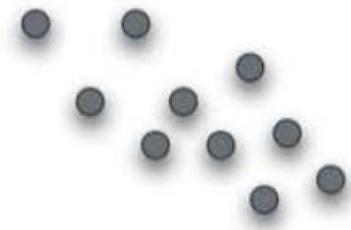
Kovaryans



$$\text{covariance} = \frac{(-2) + 0 + (-2)}{3} = -\frac{4}{3}$$

$$\text{covariance} = \frac{2 + 0 + 2}{3} = \frac{4}{3}$$

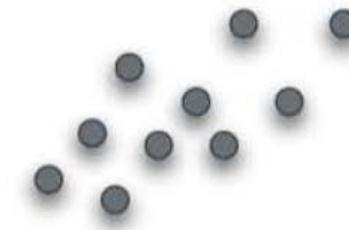
Kovaryans



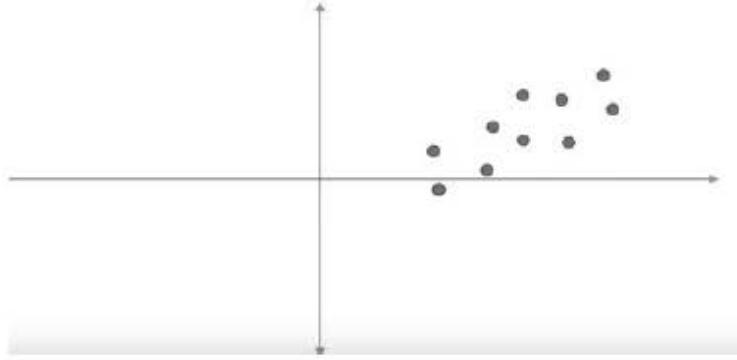
negative
covariance



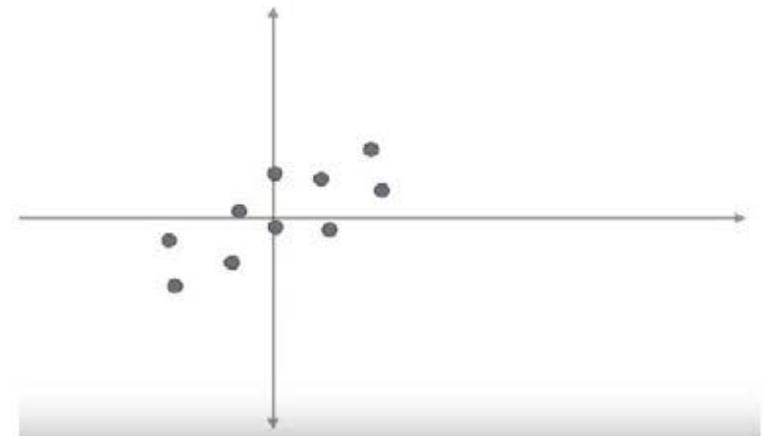
covariance zero
(or very small)



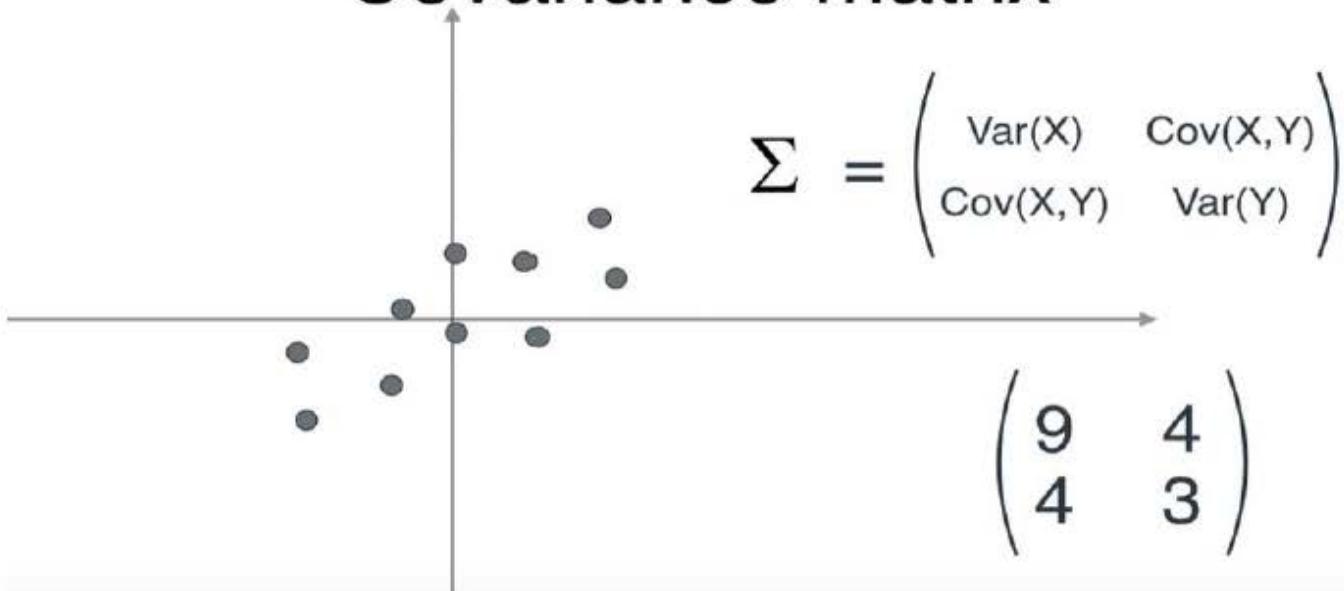
positive
covariance



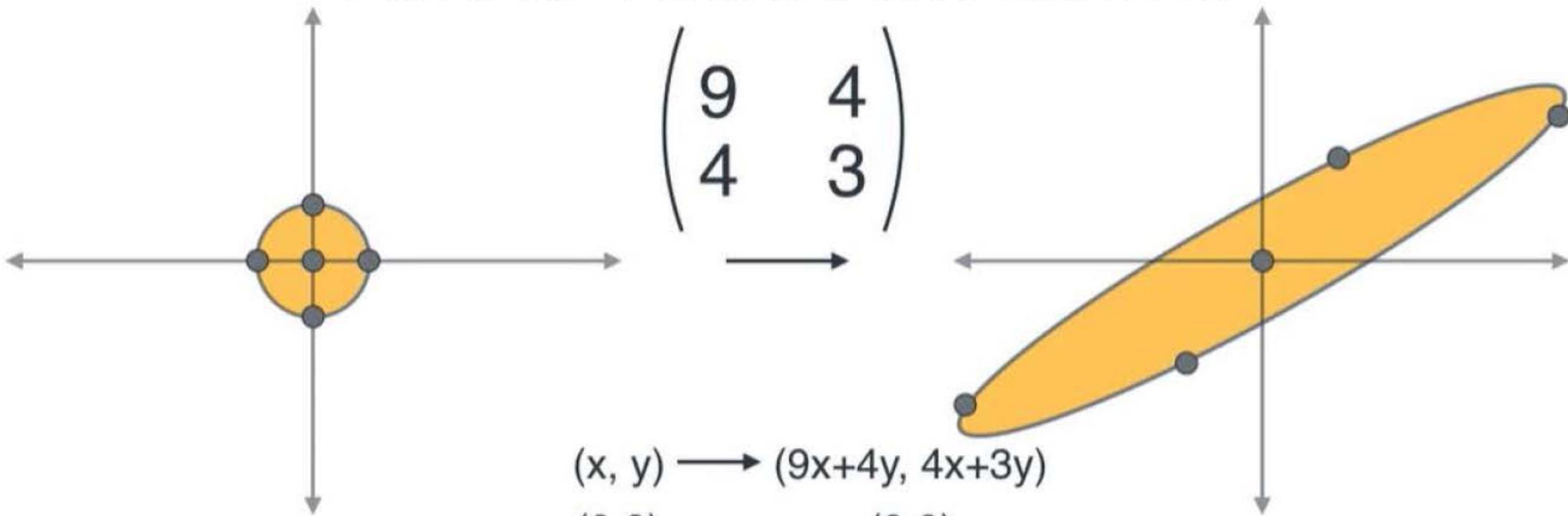
Orta noktasına göre
merkeze taşı



Covariance matrix



Linear Transformations

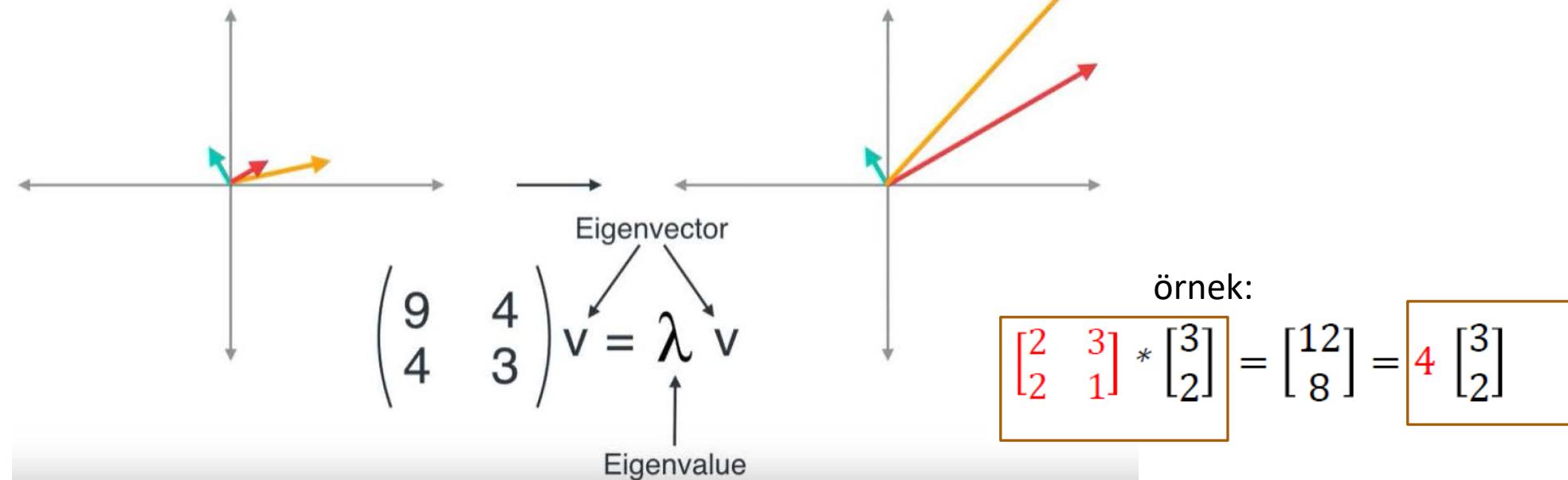


$$(x, y) \longrightarrow (9x+4y, 4x+3y)$$

(0,0)	(0,0)
(1,0)	(9,4)
(0,1)	(4,3)
(-1,0)	(-9,-4)
(0,-1)	(-4,-3)

Eigenvalue (özdeğer) - Eigenvector (özvektör)

- Bir vektör üzerine uygulanan matris, o vektörün hem büyüklüğünü hem de yönünü değiştirir. Buna rağmen, bir matris bazı belirli vektörler üzerine etkidiğinde onların sadece büyüklüğünü değiştirir, doğrultularını değiştirmez.
- Doğrultusu değişmeyen bu vektörler söz konusu matrisin **özvektörleri** olarak adlandırılır.
- Bir matris, bir özvektöru üzerinde etkidiğinde onun büyüklüğünü bir çarpan kadar katlar. Bu çarpan pozitif ise vektörün yönü değişmeden kalır, negatif ise vektörün yönü tersine döner (dikkat edilirse her iki durumda da vektörün doğrultusu değişmez.). Bu çarpana, söz konusu özvectöre ilişkin **özdeğer** denir.



Eigenvalues (özdeğerler)

Characteristic Polynomial

$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$
$$\begin{vmatrix} 9-x & 4 \\ 4 & 3-x \end{vmatrix} = (9-x)(3-x) - 16 = x^2 - 12x + 11$$
$$= (x-11)(x-1)$$

Eigenvalues 11 ve 1

Eigenvectors (özvektörler)

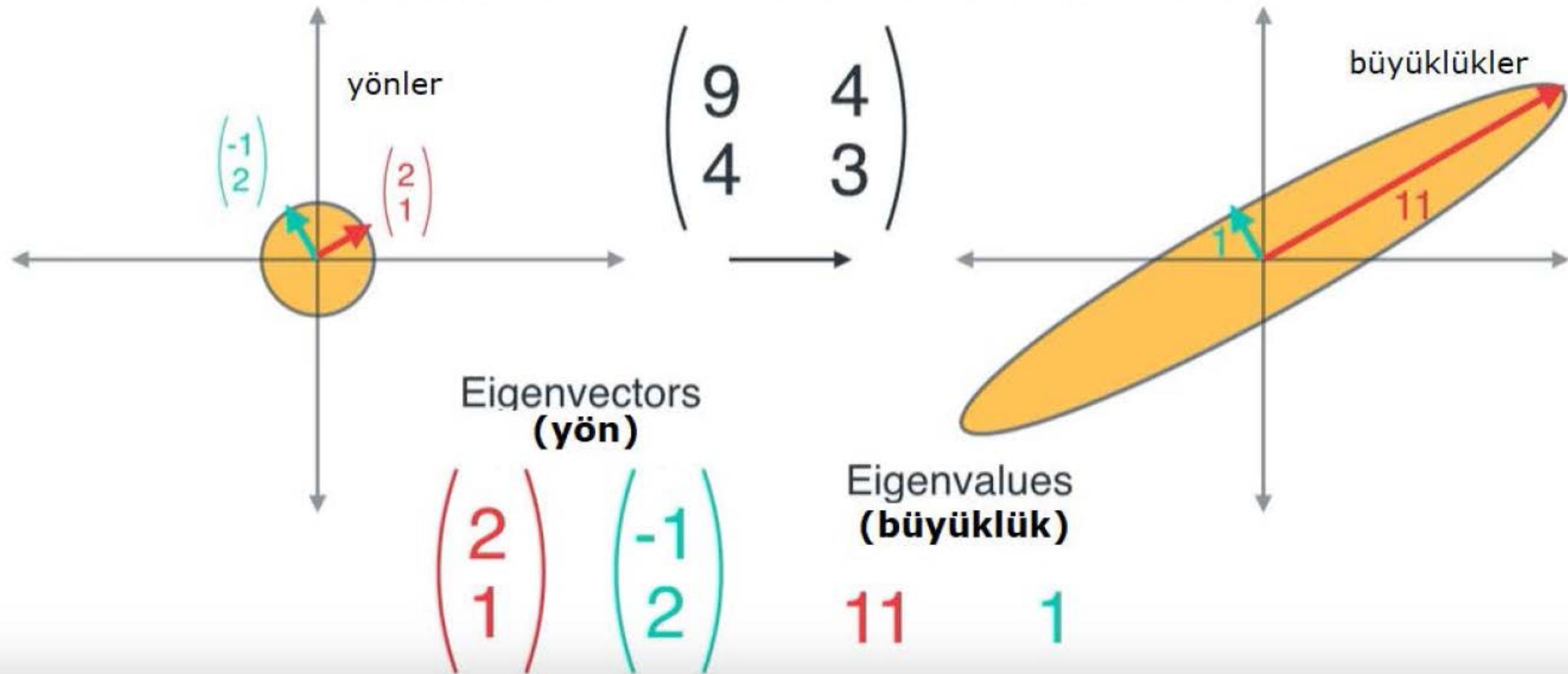
$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \textcolor{red}{11} \begin{pmatrix} u \\ v \end{pmatrix}$$

$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \textcolor{teal}{1} \begin{pmatrix} u \\ v \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \textcolor{red}{2} \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

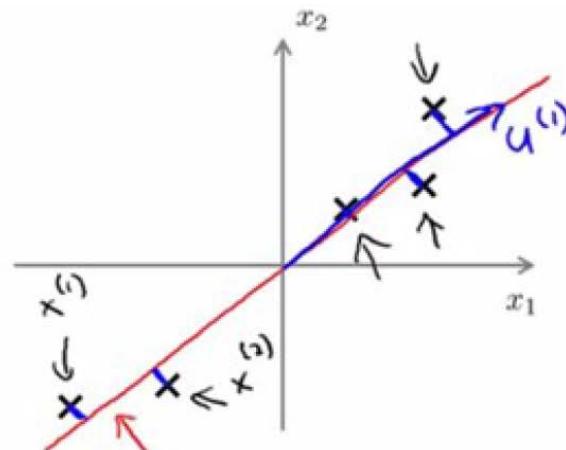
Linear Transformations



PCA – Temel Bileşen Analizi

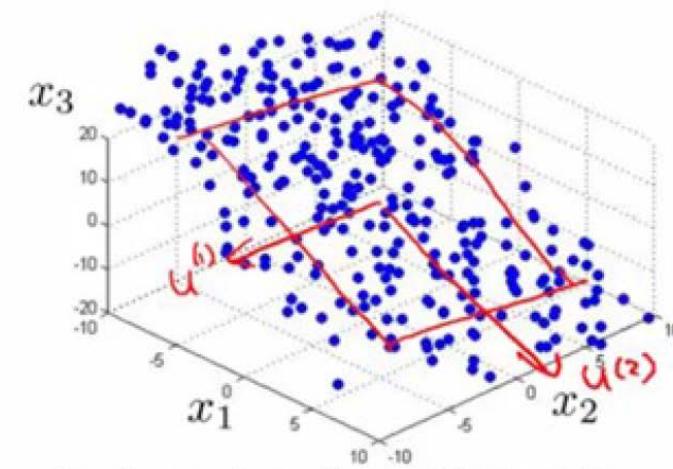
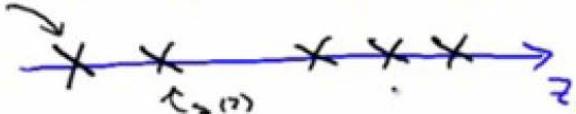
- Bir değişkenler kümesinin varyans-kovaryans yapısını, bu değişkenlerin doğrusal birleşimleri vasıtasıyla açıklayarak, boyut indirgenmesini ve yorumlanması sağlayan, çok değişkenli bir istatistik yöntemidir.
- Bu yöntemde p adet değişken; doğrusal, ortogonal ve birbirinden bağımsız olma özelliği taşıyan k ($k \leq p$) tane yeni değişken dönüştürilmektedir.
- Bu indirgenmede önemli varyans kaybı olmamaktadır. Aslında yeni oluşturulan bu k adet değişken, gerçek değişkenlerin doğrusal bileşimidir. Sıkıştırma algoritmalarında da PCA kullanılmaktadır.

PCA – Temel Bileşen Analizi



Reduce data from 2D to 1D

$$z^{(i)} \quad x^{(i)} \in \mathbb{R}^2 \rightarrow z^{(i)} \in \mathbb{R}$$



Reduce data from 3D to 2D

PCA işlem adımları

1. Veriler ortalamaya düzgünleştirilir (normalizasyon-sadece gerekli durumlarda).
2. Kovaryans matrisi hesaplanır.
3. Eigenvalue(özdeğerler) ve Eigenvector (özvektörler) hesaplanır.
4. İndirgenme için özellik vektörü seçilir ve indirgenme çarpımı yapılır.

Varyans Hesaplanması

Standart sapma:

- Veri setindeki verilerin, ortalamadan farklarının karelerinin toplamlarının $(n-1)$ 'e bölümünün karaköküdür.
- Ortalamanın, ne kadar gerçekçi olduğunu, verilerin bu ortalamadan ne kadar uzak/yakın olduğunu gösteren parametredir.

Varyans ise standart sapmanın karesidir.

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

Kovaryans Matrisinin Hesaplanması

- Kovaryans ise iki veri arasındaki değişimini hesaplayan bir parametredir. x ve y ile gösterilen iki dizi arasındaki kovaryans denklem:

$$\text{cov}_{(x,y)} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

- Kovaryans matrisi ise iki veri kümesi için hazırlanan bir matristir. k ile gösterimi:

$$k_{\text{cov}(x,y)} = \begin{bmatrix} \text{cov}_{x,x} & \text{cov}_{y,x} \\ \text{cov}_{x,y} & \text{cov}_{y,y} \end{bmatrix}$$

özdeğer ve özvektör hesaplanması

Örnek : $\begin{bmatrix} -7 & 7 \\ -5 & 5 \end{bmatrix}$ şeklinde verilmiş olan 2x2 matrisin özdeğer ve özvektörenin hesaplanması ?

- İlk olarak diagonellerden λ (lamda) çıkarılır

$$\begin{bmatrix} -7 - \lambda & 7 \\ -5 & 5 - \lambda \end{bmatrix}$$

- İçler dışlar çarpımı yapılarak sıfıra eşitlenir. λ değeri bulunur.

$$(-7 - \lambda)(5 - \lambda) - [(-5) 7] = 0$$

$$-35 + 7\lambda - 5\lambda + \lambda^2 + 35 = 0$$

$$\lambda^2 + 2\lambda = 0$$

$$\lambda_1 = -2, \lambda_2 = 0$$

özdeğer ve özvektör hesaplanması

- $\lambda_1 = -2$ için

$$\begin{bmatrix} -7 - (-2) & 7 \\ -5 & 5 - (-2) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$-5x + 7y = 0, -5x + 7y = 0$$
$$x = 7 \text{ ve } y = 5$$

- $\lambda_2 = 0$ için

$$\begin{bmatrix} -7 - (0) & 7 \\ -5 & 5 - (0) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$-7x + 7y = 0, -5x + 5y = 0$$
$$x = 1 \text{ ve } y = 1$$

Bu durumda öz vektörlerimiz:

[7 5] ve [1 1] olur. Bu iki özvektörden en güçlü olan indirgenme vektörü olarak seçilir.

Özelik vektörü seçimi ve indirgenme çarpımı

Elde edilen her iki vektör birlikte bir matris şeklinde kullanırsa elimizdeki iki dizi yine iki dizi olarak kalacaktır. Eğer biz bu vektörlerden büyük olanı seçersek, elimizdeki iki veri dizisi tek bir dizi haline dönüşecektir ve her iki dizinin ortak özelliklerini taşıyacaktır. İndirgenme işlemi :

$$A = B^t * C^t$$

A: indirgenmiş veri dizisini

B^t : seçilen özvektörün transpozesi

C^t: düzgünleştirilmiş orijinal veri kümесinin transpozesi

Örnek

sıra	x	y	x-xort	y-yort	$(x-xort)^2$	$(y-yort)^2$	$(x-xort)(y-yort)$
1	60	112	-232,1	-473	53870,41	223729	109783,3
2	255	545	-37,1	-40	1376,41	1600	1484
3	285	600	-7,1	15	50,41	225	-106,5
4	428	845	135,9	260	18468,81	67600	35334
5	265	500	-27,1	-85	734,41	7225	2303,5
6	306	605	13,9	20	193,21	400	278
7	326	645	33,9	60	1149,21	3600	2034
8	418	806	125,9	221	15850,81	48841	27823,9
9	220	456	-72,1	-129	5198,41	16641	9300,9
10	358	736	65,9	151	4342,81	22801	9950,9
ortalama:		292,1	585	Toplam:		101234,9	392662
		Toplam/9:		11248,32222	43629,11	198186	
				22020,66667			

$$\text{COV}_{(x,y)} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

$$k_{\text{cov}(x,y)} = \begin{bmatrix} \text{cov}_{x,x} & \text{cov}_{y,x} \\ \text{cov}_{x,y} & \text{cov}_{y,y} \end{bmatrix}$$

$$= \begin{bmatrix} 11248 & 22021 \\ 22021 & 43629 \end{bmatrix}$$

Örnek

$$\begin{bmatrix} 11248 - \lambda & 22021 \\ 22021 & 43629 - \lambda \end{bmatrix}$$

$$(11248 - \lambda)(43629 - \lambda) - (22021)(22021) = 0$$

$$\lambda_1 = 54770,84 \quad \lambda_2 = 106,161$$

Örnek

λ_1 yerine konduğunda :

$$\begin{bmatrix} 11248 - 54770,84 & 22021 \\ 22021 & 43629 - 54770,84 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -43522,84 & 22021 \\ 22021 & -11141,84 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -6,05595 \\ 1 \end{bmatrix}$$

λ_2 yerine konduğunda :

$$\begin{bmatrix} 11248 - 106,161 & 22021 \\ 22021 & 43629 - 106,161 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 11141,83 & 22021 \\ 22021 & 43522,83 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -8,53834 \\ 1 \end{bmatrix}$$

Bu özvektörlerden 2.sinin boyutu daha büyütür.

boyut indirgenmiş sonuç

$$\text{Sonuç} = B^t * C^t$$

$$= [-8,53834 \quad 1] * \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline & 60 & 255 & 285 & 428 & 265 & 306 & 326 & 418 & 220 & 358 \\ \hline & 112 & 545 & 600 & 845 & 500 & 605 & 645 & 806 & 456 & 736 \\ \hline \end{array}$$

$$= \boxed{-400,3004 \quad -1632,2767 \quad -1833,4269 \quad -2809,40952 \quad -1762,6601 \quad -2007,732 \quad -2138,49884 \quad -2763,03 \quad -1422,43 \quad -2320,73}$$

```
data_iris <- iris[1:4]
Cov_data <- cov(data_iris )
Eigen_data <- eigen(Cov_data)

PCA_data <- princomp(data_iris ,cor="False")
Eigen_data$values
PCA_data$sdev^2

PCA_data$loadings[,1:4]
Eigen_data$vectors

summary(PCA_data)
biplot (PCA_data)
screeplot(PCA_data, type="lines")
```

Veri Madenciliği

DR. ŞAFAK KAYIKÇI

Birliktelik Kuralları

- Bir arada geçen olayları çözümlemek veri madenciliği kapsamı içine girmektedir.
- Olayların birlikte gerçekleşme durumlarını çözümleyen veri madenciliği yöntemlerine ***birliktelik kuralları (association rules)*** adı verilmektedir.
- Bu yöntemler, birlikte olma kurallarını belirli olasılıklarla ortaya koyar.

Birliktelik Kuralları

- Birliktelik kurallarının en yaygın uygulaması perakende satışlarda müşterilerin satın alma eğilimlerini belirlemek amacıyla yapılmaktadır.
- Müşterilerin bir anda satın aldığı tüm ürünleri ele alarak satın alma eğilimini ortaya koyan uygulamalara "**Pazar sepet çözümlemesi**" adı verilir.
- Mağaza yöneticileri söz konusu ürünleri mağaza içerisinde birbirine yakın raflara yerleştirerek satışların artmasını sağlayabilir.

Destek (support) ve Güven (confidence) Ölçütleri

- Destek ve güven ölçütlerinin hesaplanmasında "destek sayısı" adı verilen bir değer kullanılır.
- "Kural Destek Ölçütü" bir ilişkinin tüm alışverişler içinde hangi oranda tekrarlandığını belirler.
- "Kural Güven Ölçütü", A ürün grubunu alan müşterilerin B ürün grubunu da alma olasılığını ortaya koyar.

Destek (support) Ölçütü

A ürün grubu alanların B ürün grubunu da alma durumu, yanı birliktelik kural $A \rightarrow B$ biçiminde gösterilir. Bu durumda kural destek ölçütü şu şekilde ifade edilebilir :

$$\text{destek } (A \rightarrow B) = \text{sayı}(A, B) / N$$

Burada $\text{sayı}(A, B)$ destek sayısı A ve B ürün gruplarını birlikte içeren alışveriş sayılarını göstermektedir. N ise tüm alışveriş sayısını göstermektedir.

Güven (Confidence) Ölçütü

A ve B ürün gruplarının birlikte satın alınması olasılığını ifade eden kural güven ölçütü de şu şekilde hesaplanır:

$$\text{güven } (A \rightarrow B) = \text{sayı}(A, B) / \text{sayı}(A)$$

Birliktelik kuralları belirlenirken destek ve güven ölçütlerinin yanı sıra, bu değerleri karşılaştırmak üzere eşik değerleri kullanılır. Hesaplanan destek ve güven ölçütlerinin **destek(esit)** ve **destek(güven)** değerlerinden büyük olması beklenir. Ölçüt değerleri ne kadar büyükse birliktelik kurallarının o derece güçlü olduğuna karar verilir.

Güven Ölçütü Hesabı

Bir mağazada 10 müşterinin bir defada yaptığı alışverişe dayanarak birliktelik kuralı şu şekilde olsun :

güven (ekmek,peynir → süt)

{ekmek,peynir,süt} ile ilgili destek sayısı (yani bu ürünü birlikte sayısı) 3 ve müşteri sayısı 10 ise

$$\begin{aligned}\underline{\text{destek(ekmek,peynir} \rightarrow \text{süt)}} &= \text{sayı(ekmek,peynir,süt)} / \text{müşteri sayısı} \\ &= 3/10 = 0.3 = 30\%\end{aligned}$$

Güven Ölçütü Hesabı

Bu kez {Ekmek,Peynir} ile ilgili destek sayısının (yani bu iki ürünü birlikte satın alanların sayısının) 4 olduğunu varsayıyalım. O halde kural güven ölçütü şu şekilde elde edilir:

$$\begin{aligned}\underline{\text{güven(ekmek,peynir} \rightarrow \text{süt)}} &= \text{sayı(ekmek,peynir,süt)} / \text{sayı(ekmek,peynir)} \\ &= 3/4 = 0.75 = 75\%\end{aligned}$$

Apriori Algoritması

1. Birliktelik çözümlemesinin yapılabilmesi için öncelikle destek ve güven ölçütlerini karşılaştırmak üzere eşik değerleri belirlenir. Uygulamadan elde edilen değerlerin bu eşik değerlere eşit yada büyük olması beklenir.
2. Her bir ürünün tekrar sayıları yani destek sayıları hesaplanır. Bu destek sayıları eşil destek sayısı ile karşılaştırılır. Eşik destek sayısından küçük olanlar çözümlemeden çıkarılır.
3. Yukarıdaki adımda seçilen ürünler bu kez ikişerli gruplandırılarak, bu grupların tekrar sayıları, yani destek sayıları elde edilir. Bu sayılar eşik destek sayıları ile karşılaştırılır. Eşik değerden küçük değerlere sahip satırlar çözümlemeden çıkarılır.

Apriori Algoritması

- 4) Bu kez üçerli, dörderli vb. gruplandırmalar yapılarak bu grupların destek sayıları elde edilir ve eşik değerleri karşılaştırılır. Eşik değerlere uygun olduğu sürece işlemlere devam edilir.
- 5) Ürün grubu belirlendikten sonra kural destek ölçütüne bakılarak birlikte kuralları türetilir ve bu kuralların her birisiyle ilgili olarak güven ölçütleri hesaplanır.

Örnek

Hangi ürünler hangi ürünlerle birlikte satın alma eğilimindedir?

<u>Müşteri</u>	<u>Aldığı Ürünler</u>
1	şeker, çay, ekmek
2	ekmek,peynir,zeytin,makarna
3	şeker,peynir,deterjan,ekmek,makarna
4	ekmek,peynir,çay,makarna
5	peynir,makarna,şeker,bira

Çözüm

1. Öncelikle eşik değerleri belirlenir.

destek(eşik) = %60

güven(eşik) = %75 olsun.

*destek(eşik)%60 olduğuna ve tüm müşteri sayısı 5 olduğuna göre
eşik destek sayısı $(0.60)(5) = 3$ olarak alınır.*

Çözüm

2. Her bir ürün için destek değerleri hesaplanır.

<u>Ürün</u>	<u>Sayı</u>
Şeker	3
Çay	2
Ekmek	4
Makarna	4
Peynir	4
Deterjan	1
Bira	1
Zeytin	1

3. Eşik destek sayısı 3 olduğu için, bu değerden küçük değerler çıkarılır.

<u>Ürün</u>	<u>Sayı</u>
Şeker	3
Çay	2
Ekmek	4
Makarna	4
Peynir	4
Deterjan	1
Bira	1
Zeytin	1

Çözüm

4. Kalan değerlerden ikili gruplar oluşturulur ve bunların destek sayıları hesaplanır.

<u>Ürün</u>	<u>Sayı</u>
Şeker,Ekmek	2
Şeker,Makarna	2
Şeker,Peynir	2
Ekmek,Makarna	3
Ekmek,Peynir	3
Makarna,Peynir	4

5. Tekrar eşik destek sayılarından küçük olanlar çıkartılır.

<u>Ürün</u>	<u>Sayı</u>
Şeker,Ekmek	2
Şeker,Makarna	2
Şeker,Peynir	2
Ekmek,Makarna	3
Ekmek,Peynir	3
Makarna,Peynir	4

Çözüm

6. En son elimizde üçlü grup kalmıştır. Bunun destek sayısı şu şekildedir.

<u>Ürün</u>	<u>Sayı</u>
Ekmek, Makarna, Peynir	3

7. Bundan sonra *birliktelik kuralları* elde edilebilir. Kurallarla birlikte **kural destek ölçütlerini** ve **kural güven ölçütleri** de hesaplanmalıdır.

$$\text{sayı}(A,B) = \text{sayı}(ekmek,makarna,peynir) = 3$$

Bu değere bağlı olarak kural destek ölçütü:

$$destek(A \rightarrow B) = \text{sayı}(ekmek,makarna,peynir) / N = 3/5 = 0.6$$

Bu destek ölçütü koşul olarak verdigimiz eşik değerden küçük değildir. O halde bu kural kullanılabilir. Kural destek sayılarına bağlı olarak birliktelik kuralları türeterek, bu kurallar için güven ölçütlerini elde edeceğiz.

Sonuçlar

Elde edilen {ekmek, makarna, peynir} kümesi göz önüne alınarak

$$\begin{aligned}\text{güven(ekmek,makarna} \rightarrow \text{peynir)} &= \text{sayı(ekmek,makarna,peynir)} / \text{sayı(ekmek,makarna)} \\ &= 3 / 3 = \%100\end{aligned}$$

$$\begin{aligned}\text{güven(ekmek} \rightarrow \text{peynir,makarna)} &= \text{sayı(ekmek,makarna,peynir)} / \text{sayı(ekmek)} \\ &= 3 / 4 = \%75\end{aligned}$$

$$\begin{aligned}\text{güven(peynir} \rightarrow \text{ekmek,makarna)} &= \text{sayı(ekmek,makarna,peynir)} / \text{sayı(peynir)} \\ &= 3 / 4 = \%75\end{aligned}$$

$$\begin{aligned}\text{güven(makarna} \rightarrow \text{ekmek,peynir)} &= \text{sayı(ekmek,makarna,peynir)} / \text{sayı(makarna)} \\ &= 3 / 4 = \%75\end{aligned}$$

Değerlendirme

<u>Birlikteklilik Kuralı</u>	<u>Anlamı</u>	<u>Güven</u>
ekmek & makarna --> peynir	Ekemek ve makarnanın bulunduğu ürün kümesinde peynirin bulunma olasılığı	100%
ekmek --> peynir & makarna	Ekmeğin olduğu kümede peynir ve makarnanın bulunma olasılığı	75%
peynir --> ekmek & makarna	Peynirin olduğu kümede ekmek ve makarnanın bulunma olasılığı	75%
makarna --> ekmek & peynir	Makarnanın olduğu kümede ekmek ve peynirin bulunma olasılığı	75%

```
install.packages("arules")
install.packages("arulesViz")
library(arules)
library(arulesViz)
data("Groceries")
summary(Groceries)
itemFrequencyPlot(Groceries,topN=20,type="absolute")

#support=0.001, minimum confidence=0.8
rules <- apriori(Groceries,parameter = list(supp = 0.001, conf = 0.80))
summary(rules)

#Oluşturulan kural sayısı: 410, Kuralların uzunluğa göre dağılımı: Çoğu kural 4 öğe uzunluğundadır
```

```
options(digits=2)
inspect(rules[1:10])

#Birisi yoğurt ve tahıl satın alırsa, tam yağlı süt de satın alma olasılığı% 81'dir.
plot(rules[1:10],method = "graph",control = list(type = "items"))
plot(rules[1:10],method = "paracord",control = list(reorder = TRUE))
plotly_arules(rules)
```