



Yapay Zeka Makine Öğrenmesi Derin Öğrenme



Sevdanur GENÇ

<https://github.com/SevdanurGENC>

<http://sevdanurgenc.com>



Ajanda

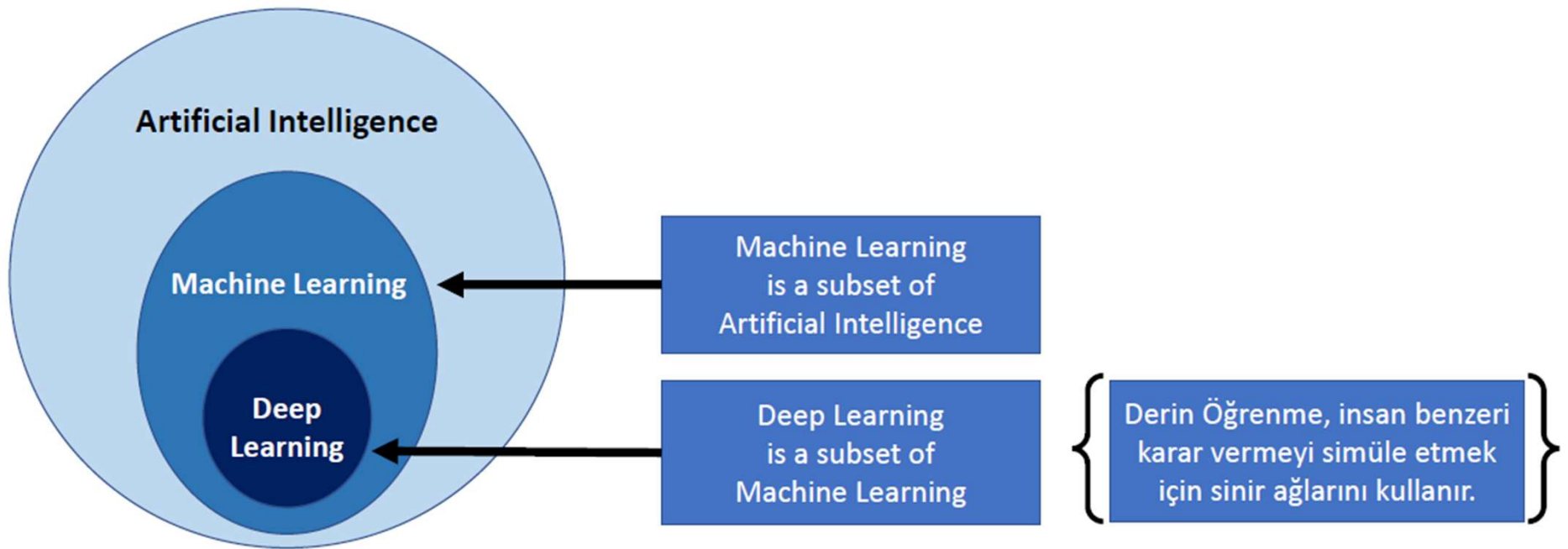
- **Yapay Zeka, Makine Öğrenmesi, Derin Öğrenme**
- **Makine Öğrenmesinin Yapay Zeka ile İlişkisi**
- **Sınıflandırma, Regresyon ve Kümeleme**
- **Makine Öğrenmesinde Öğrenme Modelleri**
- **Makine Öğrenmesi Algoritmaları**
- **Öznitelik Seçim Algoritmaları**
- **Sınıflandırmalar Algoritmaları için Performans Ölçüm Kriterleri**
- **Kuantum Makine Öğrenmesi (Quantum Machine Learning)**

Yapay Zeka



- Yapay Zeka, gerçek yaşam durumlarında karar veren makineler geliştirmek için, insan dillerini anlamak, etkileşimli oyunlar oynamak için bilgisayarların programlanmasıdır.
- İşini mükemmel yapan **canlı sistemlerini ve insan beynini model alan** yapay zeka çalışmaları; günlük hayatın farklı alanlarında ürünler vermesinin yanında, **tahmin, sınıflandırma, kümeleme** gibi amaçlar için de kullanılmaktadır.
- Genel anlamda yapay zekadan kastedilen; insan zekasının, sinir sistemi, gen yapısı gibi fizyolojik ve nörolojik yapısının ve doğal olayların modellenerek makineler (bilgisayar ve yazılımlara) aktarılmasıdır. Özetle yapay zeka; “**insan gibi düşünen, insan gibi davranan, akılcı (rasyonel) düşünen ve akılcı davranan**” canlıların zekice olarak kabul edilen davranışlarına sahip bilgisayar sistemleridir ve makine öğrenmesi bu anlamda yapay zekanın evresi olarak kabul edilmektedir.
- **Uzman sistemler, Genetik algoritmalar, Bulanık mantık, Yapay sinir ağları, Makine öğrenmesi** gibi teknikler, genel olarak yapay zeka teknolojileri olarak adlandırılmaktadır.

Yapay Zeka



Yapay Zeka, Makine Öğrenmesi, Derin Öğrenme



YAPAY ZEKA

1950'lerde açığa çıkmış olup, makinaların insanlar kadar kabiliyetli bir şekilde bazı işlemleri yapabilmesi olarak tanımlanabilir.



MAKİNE ÖĞRENMESİ

1980'lerde açığa çıkmış olup veri madenciliğın kullanılması ile beraber daha popüler hale gelmeye başlamıştır.



DERİN ÖĞRENME

2010'lu yıllarda kullanılmaya başlanmış, büyük ile tek bir katmanda değil, birçok katmanda makine öğreniminde kullanılan hesapları tek bir seferde yapma kabiliyetine sahiptir.

Yapay Zeka, Makine Öğrenmesi, Derin Öğrenme

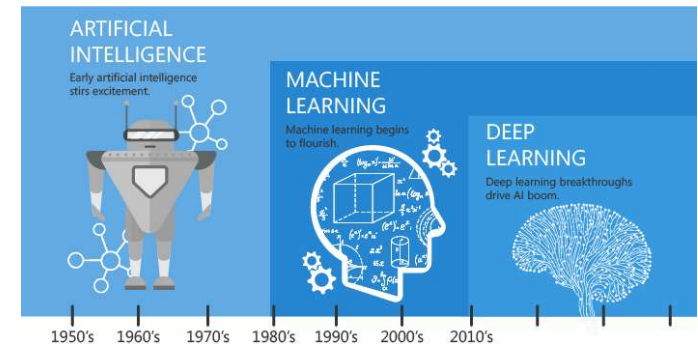


- **Yapay zekâ**, ister makine öğrenmesi kullansın ister kullanmasın **herhangi bir tahmin veya karar işlemini gerçekleştiren** teknolojilerin genel adıdır. Genel kanaatin aksine yapay zekâ makine öğrenmesi veya derin öğrenme algoritmaları olmaksızın da çalışan bir algoritma olabilir.
- Makine öğrenmesi algoritmaları ortaya çıkana kadar **yapay zekâ çalışmaları “hard-coded”** olarak nitelendirilen yani **tüm mantıksal ve matematiksel işlemlerin yazılımcı tarafından bizzat kodlandığı bir yapıya dayanmaktaydı**. Örneğin ilk satranç oyuncusu yapay zekâ algoritmaları tamamen böyleydi. Yapay zekânın bu türü sembolik yapay zekâ olarak adlandırılmaktadır.
- **Makine öğrenmesi**nde **algoritmalar tamamen veriden öğrenir**. Makine öğrenmesini hard-coded olarak kodlanmış sembolik **yapay zekâ algoritmalarından ayıran özellik** algoritmanın tamamen veriden öğrenmesidir. Doğal olarak akla makine öğrenmesi algoritmalarının verilerden öğrenmesinin riskli tarafları gelebilir.
- **Derin öğrenme** modeli, **verinin yapısına göre hangi parametrelere ne ağırlık verileceğini kendisi keşfetmektedir**. Derin öğrenme algoritması da **veriye dayalı öğrenme** gerçekleştirmekle birlikte, **öğrenme süreci** standart makine öğrenmesi algoritmalarında olduğu gibi tek bir matematiksel modele değil **sinirsel ağ (neural network)** olarak ifade edilen ağ diyagramlarına benzeyen yapıda geliştirilen hesaplamalarla çalışmaktadır.
- Her derin öğrenme algoritması bir makine öğrenmesi algoritmasıdır çünkü **verilerden öğrenme** gerçekleştirmektedir. Ancak her makine öğrenmesi algoritması derin öğrenme algoritması değildir; nitekim derin öğrenme, **makine öğrenmesinin spesifik bir türüdür**.



Makine Öğrenmesinin Yapay Zeka ile İlişkisi

- Bilimsel bir çaba olarak, makine öğrenmesi yapay zeka arayışından doğdu.
- Akademik bir disiplin olarak AI'nın ilk günlerinde, bazı araştırmacılar **makinenin verilerden öğrenmesini** istiyorlardı. Soruna çeşitli sembolik yöntemlerle ve daha sonra "sinir ağları" olarak adlandırılan yöntemle yaklaşmaya çalıştılar; bunlar çoğunlukla algılayıcılar ve daha sonra genelleştirilmiş doğrusal istatistik modellerinin yeniden icadı olarak bulunan diğer modellerdi. Olasılıksal akıl yürütme, özellikle otomatik tıbbi tanıda da kullanılmıştır.
- Bununla birlikte, mantıksal, bilgiye dayalı yaklaşıma artan vurgu, yapay zeka ve makine öğrenmesi arasında bir kaymaya neden oldu.
- Olasılıksal sistemler, veri toplama ve temsil etme ile ilgili teorik ve pratik problemlerle boğulmuştu. 1980'e gelindiğinde, uzman sistemler yapay zekaya hâkim olmuştu ve istatistikler kaybolmuştu.
- Yapay zeka içinde **sembolik / bilgiye dayalı öğrenme üzerine çalışmalar devam etmiş**, bu da tümevarımsal mantık programlamaya yol açmıştır, ancak daha istatistiksel araştırma çizgisi artık yapay zekâ alanının dışında, örüntü tanıma ve bilgi edinmede olmuştur.

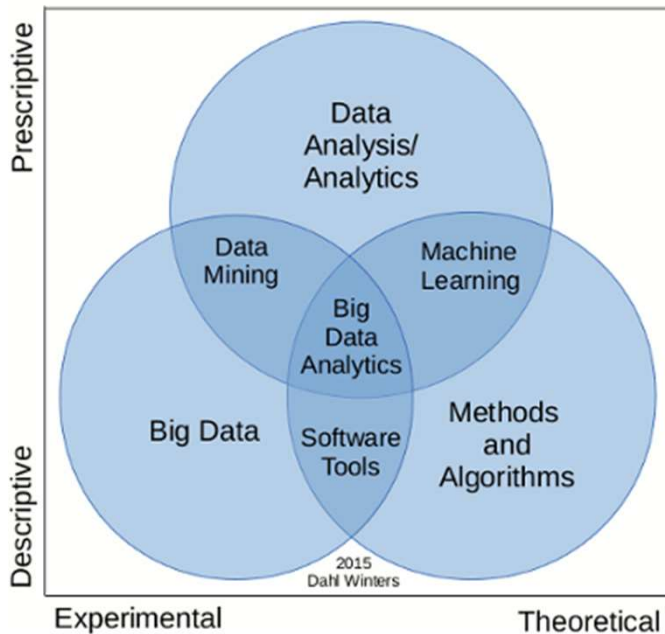




Veri Madenciliği ile İlişkisi

- Makine öğrenmesi algoritmaları veri madenciliği problemlerinin çözümüne yardımcı olur.
- Örneğin sınıflandırma bir veri madenciliği problemi iken sınıflandırma işleminde kullanılan karar ağaçları bir makine öğrenmesi algoritmasıdır.
- **Veri madenciliği** veri ön işleme ve yorumlama dahil sürecin tamamına verilen isim iken **makine öğrenmesi** veri ön işlem (pre-processing) ve veri son işleme (post-processing) karışmaz.

The Fields of Data Science



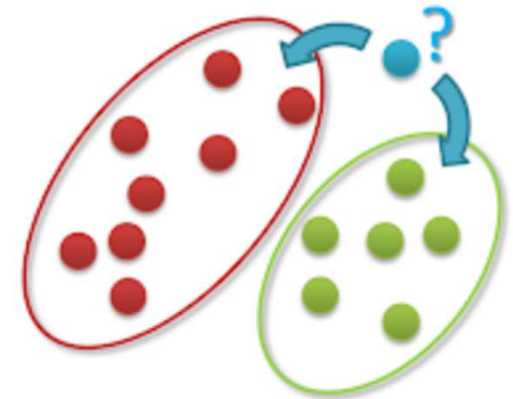
Parameter	Data Mining	Machine Learning
Types	Association Rules Classification Clustering Sequential Patterns Sequence Similarity	Supervised Un Supervised Reinforcement
Relationship	Forecasting Classifying Things Associating Similar Things Clustering into Groups Sequence Making	Automating Employee Access Control Protecting Animals Predicting Emergency wait room times Identifying Heart Failure Classroom will learn, Digital Guardian, City will help you live in it and many more.....



Sınıflandırma

- Yeni kayıtların daha önceden belirlenmiş sınıflara atanmasına **sınıflandırma** denir.
- Bir Sınıflandırma modeli, **tahmin** değişkenleri (predictors) yardımıyla, **hedef** (class) değişkeninin hesap edildiği fonksiyondur.
- Genellikle, veri seti eğitim ve test setlerine bölünür, eğitim seti ile model inşa edilir, test seti ile model doğrulama yapılır.
- **Eğitim seti (training set)** sınıflandırma modelinin **elde edileceği kayıtları** içerir. Eğitim setindeki her kayıt bir tanesi **sınıf (class) özneliği olan özneliliklerden** oluşur.
- **Test seti** bir sınıflandırma modelinin **doğruluğunu belirlemek için kullanılır**.
- **Sınıflandırma için kullanılabilecek makine öğrenmesi algoritmaları:**
 - Karar ağaçları (ID3, Rnd Tree, J48)
 - Kural tabanlı sınıflayıcılar (C4.5, C-RT, Decision Table)
 - Örnek tabanlı sınıflayıcılar (k-NN, KStar, LWL)
 - Bayes sınıflayıcı (Naive bayes, Naive bayes continuous, Bayes Net)
 - Yapay sinir ağları (Multilayer perceptron, Perceptron, LVQ)
 - Destek vektör makineleri (CVM, C-SVC, SMO)

Classification

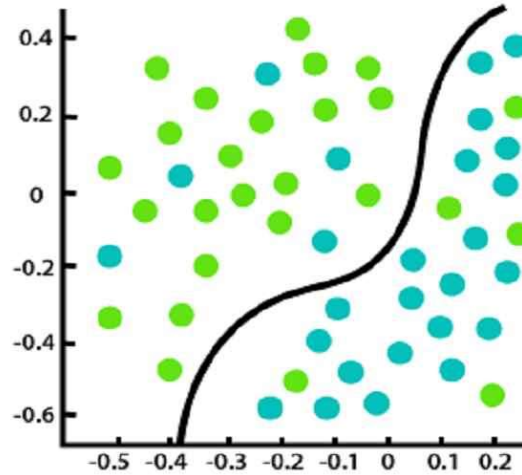
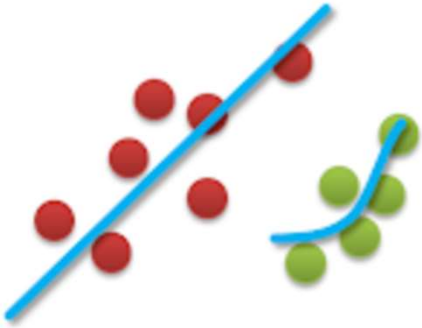




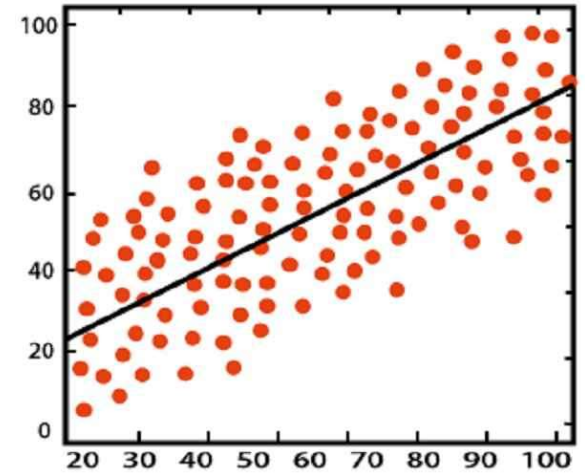
Sınıflandırma ve Regresyon

- Sınıflandırma ve regresyon aynı görevi yerine getirdiği için sıklıkla regresyon da bir sınıflayıcı olarak ele alınır.
- Aralarındaki en önemli fark **hedef değişkenle** ilgilidir.
- Eğer hedef değişkenin türü **NOMINAL** (categorical) ise **SINIFLANDIRMA**
- Eğer hedef değişken türü **SÜREKLİ** (continuous) ise **REGRESYON**
- Dolayısıyla regresyon, sonuçta sürekli değer üretir. Sadece lojistik regresyon nominal değer üretir.

Regression



Classification



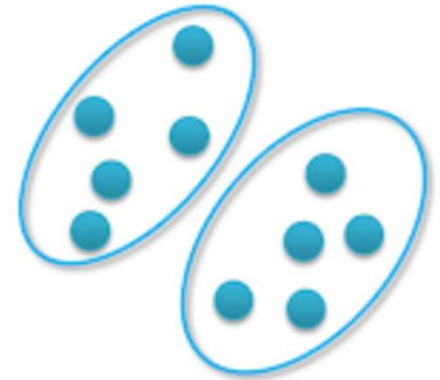
Regression



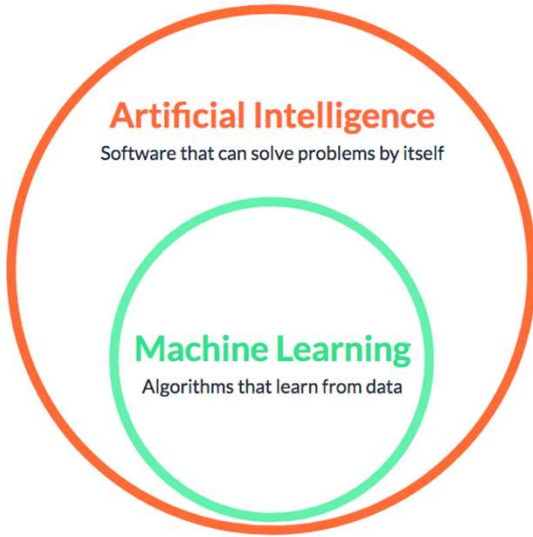
Kümeleme

- Elimizde, verileri sunan noktalar ve bu noktalar arasındaki benzerliği ölçen bir benzerlik ölçümü varken, kümelemenin amacı;
 - Birbirine daha çok benzeyen noktalar aynı kümede,
 - Birbirine benzemeyen noktalar ise farklı kümelerdedir
- gibi özellikleri sağlayan kümeleri bulmaktır.
- **Benzerlik Ölçümleri :**
 - Eğer öznitelikler **sürekli** değerler ise o zaman **Euclidean Distance**.
 - Diğerlerinde **probleme uygun** ölçümler kullanılır
- **Kümeleme algoritmaları :**
 - Merkez tabanlı kümeleme (k-means, k-medoids)
 - Hiyerarşik kümeleme (Hierarchical clusterer)
 - YSA ile kümeleme (Kohonen-SOM)

Clustering



Yapay Zeka ve Makine Öğrenmesi Arasındaki Farklar



- Yapay zekada **amaç** doğruluk değil **başarı şansını arttırmaktır**. Makine öğrenmesi başarıyı önemsemez. Amaç **doğruluğu arttırmaktır**.
- Yapay zeka akıllı çalışan **bir bilgisayar programıdır**. Makine öğrenmesi ise **makinenin veri aldığı (topladığı) ve bu veriler doğrultusunda öğrenen bir konsepttir**.
- Yapay zekada amaç karmaşık problemleri çözmek için **doğal zekayı simule etmektir**. Makine öğrenmesindeki amaç makinenin performansını en üst düzeye çıkarmak için belirli bir görev doğrultusunda toplanan verilerden bir şeyler öğrenmektir.
- Yapay zeka **karar verir**. Makine öğrenmesi, sistemin **toplanan verilerden yeni şeyler öğrenmesidir**.
- Yapay zeka **en uygun çözümü bulmak için çalışır**, Makine öğrenmesi **optimal** olup olmamasına yönelik olarak **sadece bir çözüm bulur**.
- Yapay zeka, **zekaya veya bilgeliğe yol açar**. Makine öğrenmesi **bilgiye yol açar**.

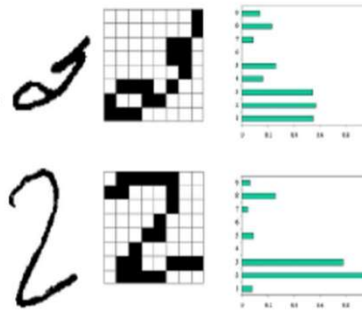
Makine Öğrenmesinin Başlıca Kullanım Alanları



- Sosyal medya
- Lojistik ve Üretim
- Finans
- Sağlık
- İnsan Kaynakları
- Satış ve Pazarlama
- Spam mail saptama
- Yüz tanıma
- Rakam tanıma
- Ses tanıma
- Ürün önerileri
- Hastalık tanısı
- Trafik tahmini
- Karakter analizi
- Kredi kartı sahtekarlık tespiti

Character Recognition

Want to learn how to recognize characters, even if written in different ways by different people



Face Recognition

Training examples of a person



Test images

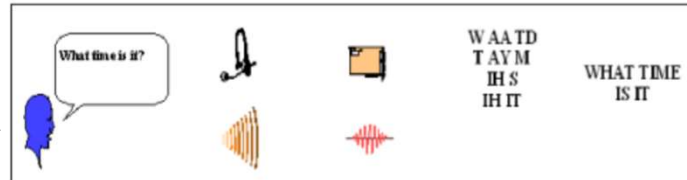


Model	Training Data Segmentation	Size Means	Error Rate
D=5(8,7)			15.5%
D=5(8,6)			17.5%
D=5(8,5)			22.5%

Figure 4.11: Segmentation and feature results for varying M

Speech Recognition

USER MICROPHONE SOUND CARD SPEECH RECOGNITION ENGINE SPEECH-AWARE APPLICATION

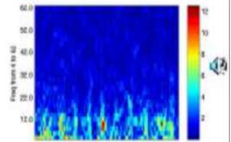


Interpreting Brainwaves

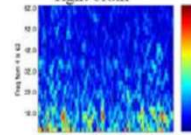
EEG electrodes reading brain waves:



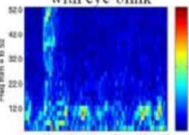
Rotation task, left brain



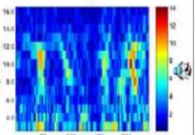
Rotation task, right brain



Resting task, with eye blink



Counting task



Makine Öğrenmesi (ML)



Makine öğrenmesi

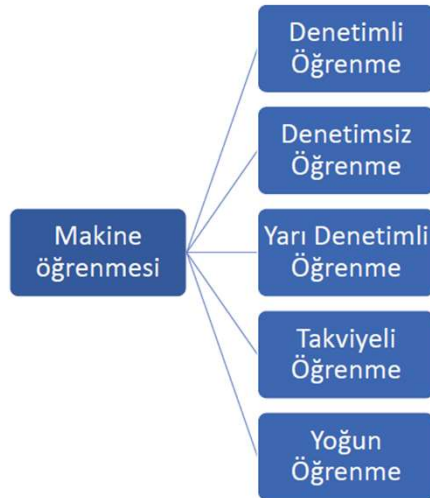
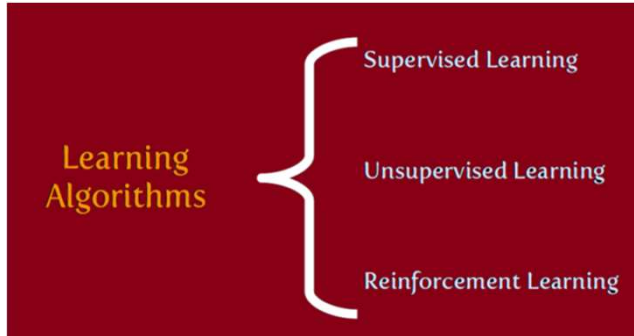
yapısal işlev olarak öğrenebilen ve veriler üzerinden tahmin yapabilen algoritmaların çalışma ve inşalarını araştıran bir sistemdir.

Makine öğrenmesi bilinen özelliklere dayanarak öğrenilen verilerden yapılan tahminler üzerine odaklanır.

Veri madenciliği ise verilerdeki (geçmiş) bilinmeyen özelliklerin keşfedilmesine odaklanır. Bu veri tabanlarında bilgi keşfi analizinin bir adımıdır.



Makine Öğrenmesinde Öğrenme Modelleri



Denetimli Öğrenme

- Etiketler sağlanmıştır, güçlü bir öğrenme sinyali vardır.
- Sınıflandırma, Regresyon.

Denetimsiz öğrenme

- Doğrudan öğrenme bilgisi yoktur. Sadece veride yapı bulmaya çalışıyoruz.
- Kümeleme, Boyutluluk azaltma.

Yarı denetimli Öğrenme

- Verilerin yalnızca bir kısmının etiketleri vardır.
- Örneğin. büyüyen bir çocuk.

Pekiştirmeli öğrenme

- Öğrenme bilgisi (skaler) bir ödüdür ve bir gecikmeyle gelebilir.
- Örneğin. labirentte bir fare, satranç oynamayı öğrenmeye çalışıyor.



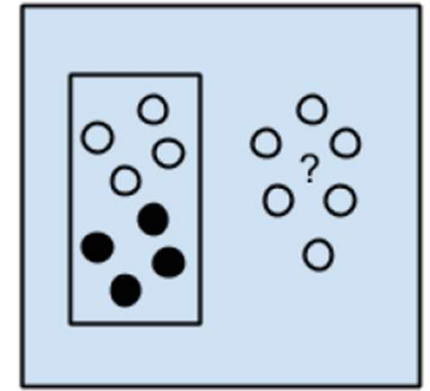
Makine Öğrenme Türleri

- Makine öğrenmesi algoritmalarının türleri yaklaşımlarına, girdi ve çıktı veri türlerine, çözmeleri amaçlanan görev veya sorun türlerine göre farklılık gösterir.
- **Eğitim modelleri:**
- Genellikle, makine öğrenmesi modelleri iyi performans gösterebilmeleri için **çok fazla veri gerektirir**.
- Bir makine öğrenmesi modeli eğilirken, bir eğitim setinden büyük, **temsili bir veri örneği toplanması gerekir**. Eğitim setindeki veriler, bir metin topluluğuna, bir resim koleksiyonuna ve bir hizmetin tek tek kullanıcılarından toplanan verilere kadar değişebilir. Aşırı sığdırma, bir makine öğrenme modeli eğitimi alırken dikkat edilmesi gereken bir şeydir.
- **Makine öğrenimi algoritmaları iki gruba ayrılır:**
 - Denetimli Öğrenme algoritmaları (Supervised Learning algorithms)
 - Denetimsiz Öğrenme algoritmaları (Unsupervised Learning algorithms)

Makine Öğrenme Türleri – Denetimli Öğrenme



- Hedef: Sınıfı veya değer etiketini tahmin edin.
- Denetimli öğrenme, etiketli eğitim verilerinden bir işlevin çıkarılmasıyla ilgili makine öğreniminin bir dalıdır. Eğitim verileri **bir dizi [giriş, hedef] çiftinden** oluşur; burada **giriş, özelliklerin bir vektörü olabilir ve hedef, işlevin çıktı vermesi** için ne istediğimizi belirtir. Hedefin türüne bağlı olarak, denetimli öğrenimi kabaca iki kategoriye ayırabiliriz: **Sınıflandırma ve Regresyon**.
- **Sınıflandırma, kategorik** hedefleri içerir; Görüntü sınıflandırması gibi bazı basit durumlardan makine çevirileri ve resim yazısı gibi bazı gelişmiş konulara kadar değişen örnekler.
- **Regresyon, sürekli** hedefleri içerir. Uygulamaları, tümü bu kategoriye giren stok tahmini, görüntü maskeleyme ve diğerlerini içerir.



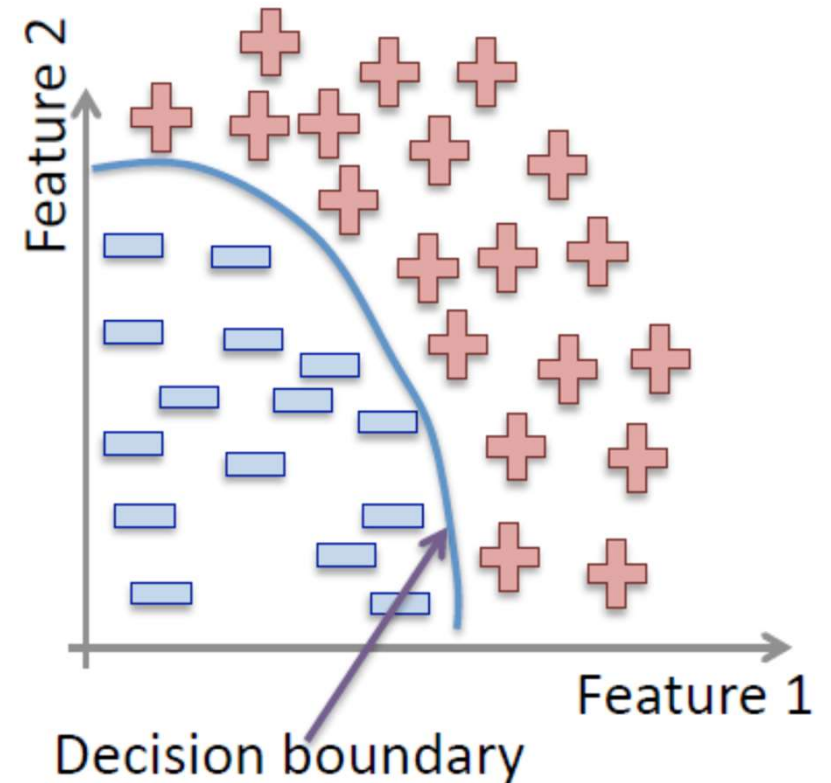
Supervised Learning
Algorithms

Makine Öğrenme Türleri – Denetimli Öğrenme



Denetimli öğrenme algoritmalarına örnek:

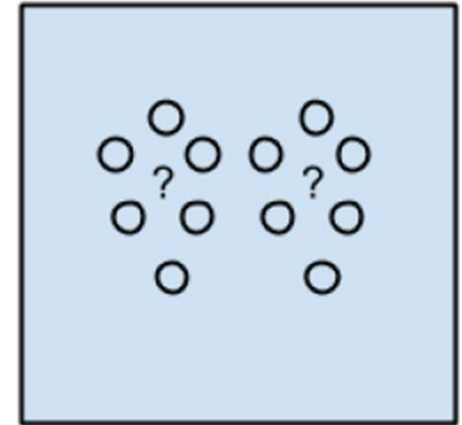
- K-En Yakın Komşular
- Doğrusal Regresyon
- Lojistik regresyon
- Rastgele Orman
- Gradyan Güçlendirilmiş Ağaçlar
- Destek Vektör Makineleri (SVM)
- Naive Bayes
- Nöral ağlar
- Karar ağaçları



Makine Öğrenme Türleri – Denetimsiz Öğrenme



- Öğrenme algoritmasına hiçbir **etiket verilmez** ve girdisinde yapı bulmak için tek başına bırakılır. Gözetimsiz öğrenme kendi içinde bir hedef (**verilerdeki gizli kalıpları keşfetme**) veya bir sona doğru bir araç (**özellik öğrenme**) olabilir.
 - k-means clustering
 - t-SNE (t-Distributed Stochastic Neighbor Embedding)
 - PCA (Principal Component Analysis)
 - Association rule
- Denetimli öğrenmenin aksine. Denetimsiz öğrenme, verilerdeki gizli yapıları tanımlayan bir işlev olan **etiketlenmemiş verilerden** kaynaklanır.
- **Denetimsiz öğrenme algoritmalarına örnekler:**
 - **Boyut Azaltma**
 - Yoğunluk Tahmini
 - Pazar Sepeti Analizi
 - Üretken düşmanlık ağları (GAN'lar)
 - **Kümeleme**

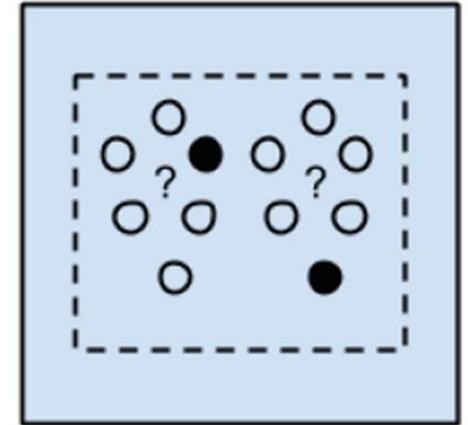


Unsupervised Learning
Algorithms

Makine Öğrenme Türleri – Yarı Denetimli Öğrenme

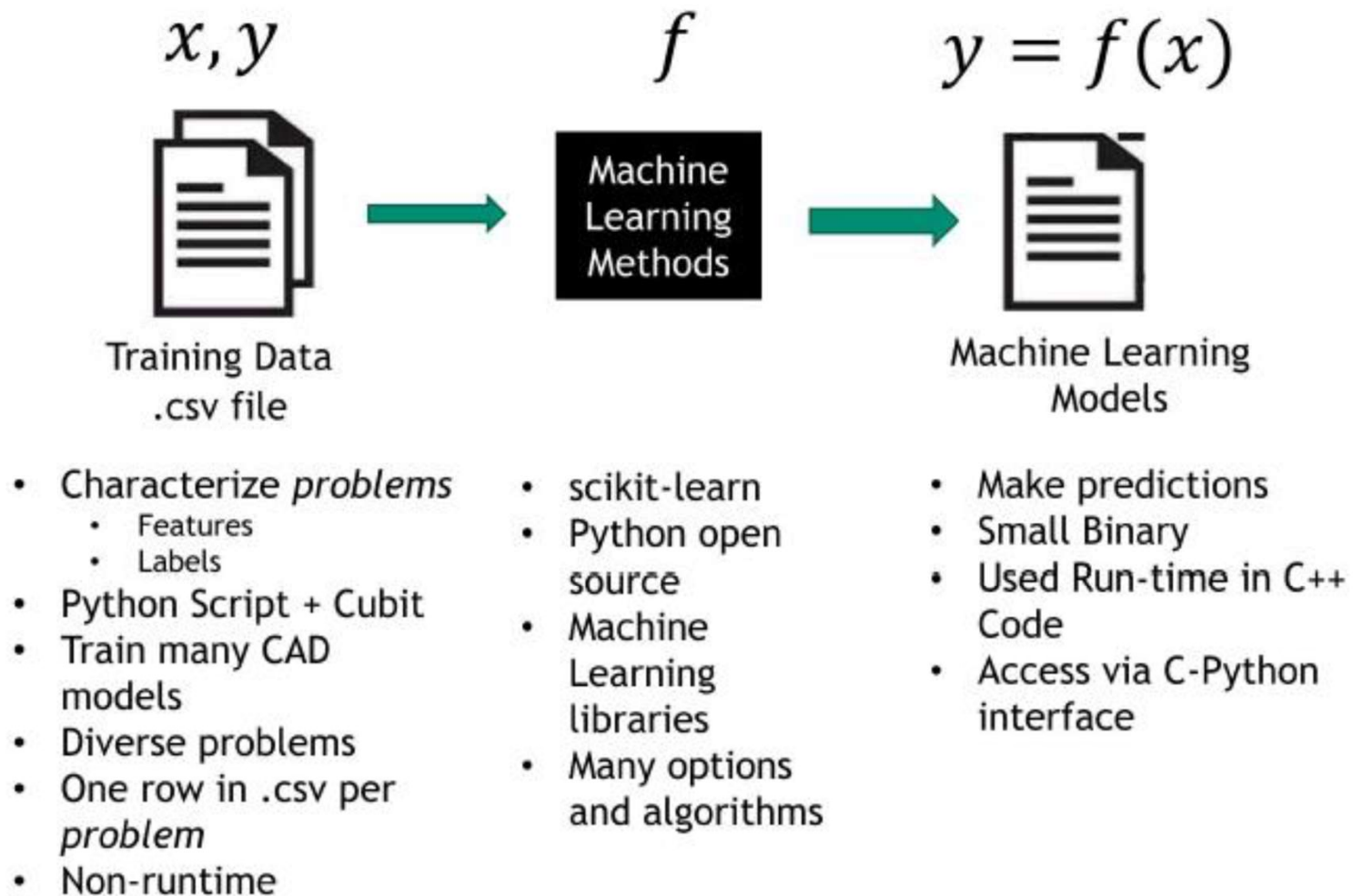


- Yarı denetimli öğrenme, denetimsiz öğrenme (herhangi bir etiketlenmiş öğrenme verisi olmadan) ve denetimli öğrenme (tamamen etiketlenmiş öğrenme verisi ile) arasındadır. Bazı öğrenme örnekleri öğrenme etiketlerinin eksik olmasına rağmen, birçok makine öğrenmesi araştırmacısı, etiketlenmemiş verilerin, az miktarda etiketlenmiş verilerle birlikte kullanıldığında, öğrenme doğruluğunda önemli bir gelişme sağlayabildiğini bulmuştur.
- Zayıf denetimli öğrenmede, öğrenme etiketleri gürültülü, sınırlı veya kesin değildir; bununla birlikte, bu etiketlerin elde edilmesi genellikle daha ucuzdur, bu da daha büyük etkili öğrenme setleriyle sonuçlanır.



Semi-supervised
Learning Algorithms

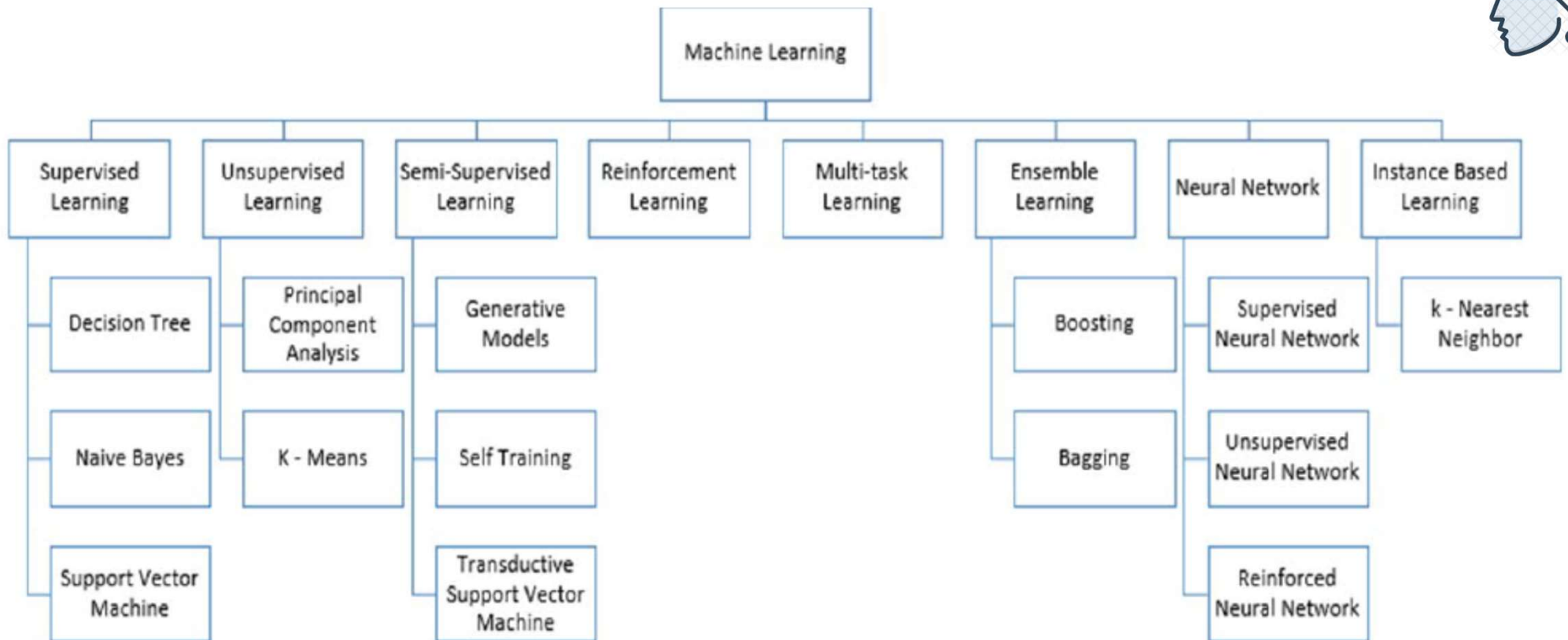
Machine Learning





Neden Python ?

- Python, popüler ve genel amaçlı bir programlama dilidir. Python kullanarak makine öğrenimi algoritmaları yazabiliriz ve bu iyi çalışır. Python'un veri bilimcileri arasında bu kadar popüler olmasının nedeni, Python'un hayatımızı daha rahat hale getiren çok çeşitli modül ve kütüphanelerin halihazırda uygulanmış olmasıdır.
 - Bazı heyecan verici Python kütüphanelerine kısaca bir göz atalım:
1. **Numpy:** Python'da n boyutlu dizilerle çalışmak için bir matematik kitaplığıdır. Hesaplamaları etkili ve verimli bir şekilde yapmamızı sağlar.
 2. **Scipy:** Sinyal işleme, optimizasyon, istatistik ve çok daha fazlasını içeren sayısal algoritmalar ve alana özgü araç kutusudur. Scipy, bilimsel ve yüksek performanslı hesaplamalar için işlevsel bir kitaplıktır.
 3. **Matplotlib:** 2D çizimin yanı sıra 3D çizim de sağlayan modaya uygun bir çizim paketidir.
 4. **Scikit-learn:** Python programlama dili için ücretsiz bir makine öğrenimi kitaplığıdır. Sınıflandırma, regresyon ve kümeleme algoritmalarının çoğuna sahiptir ve Numpy, Scipy gibi Python sayısal kütüphaneleri ile çalışır.





Makine Öğrenmesi Algoritmaları

1- Regresyon (Tahmin)

- **Sürekli** değerleri tahmin etmek için regresyon algoritmaları kullanılır.
 - Doğrusal Regresyon
 - Polinom Regresyon
 - Üstel Regresyon
 - Lojistik regresyon
 - Logaritmik Regresyon

2- Sınıflandırma

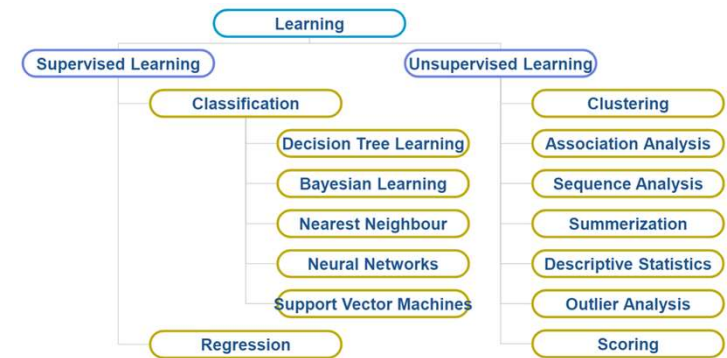
- Bir dizi ögenin sınıfını veya **kategorisini** tahmin etmek için sınıflandırma algoritmaları kullanılır.
 - K-En Yakın Komşular
 - Karar ağaçları
 - Rastgele Orman
 - Destek Vektör Makinesi
 - Naive Bayes

3- Kümeleme

- **Özetlemek** veya verileri yapılandırmak için kümeleme algoritmaları kullanılır.
 - K-means
 - DBSCAN
 - Mean Shift
 - Hierarchical

4- İlişkilendirme

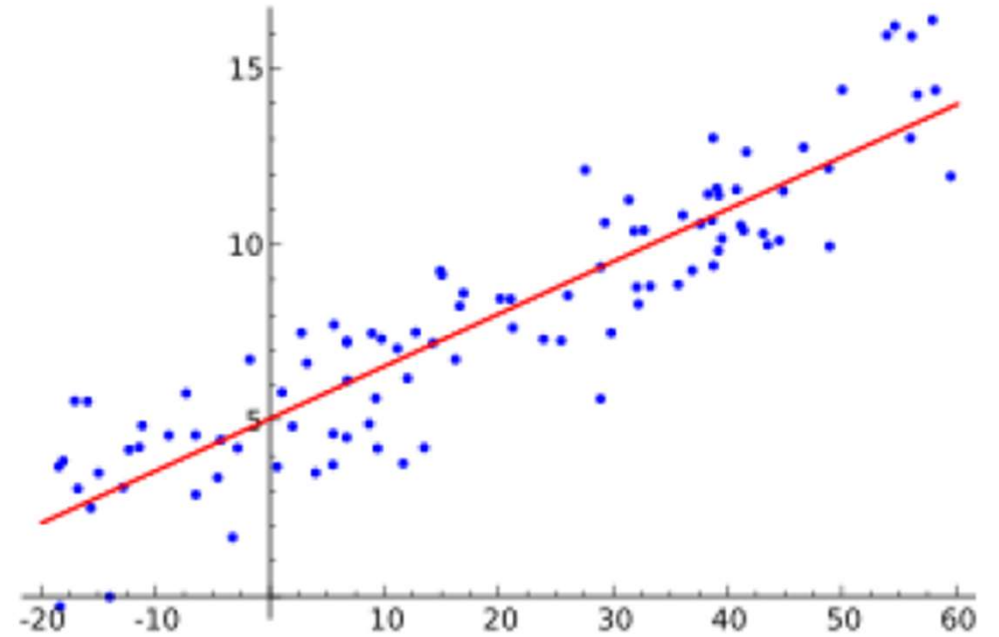
- Birlikte meydana gelen öğeleri veya olayları **ilişkilendirmek** için ilişkilendirme algoritmaları kullanıyoruz.
 - Apriori





1- Regresyon Algoritmaları

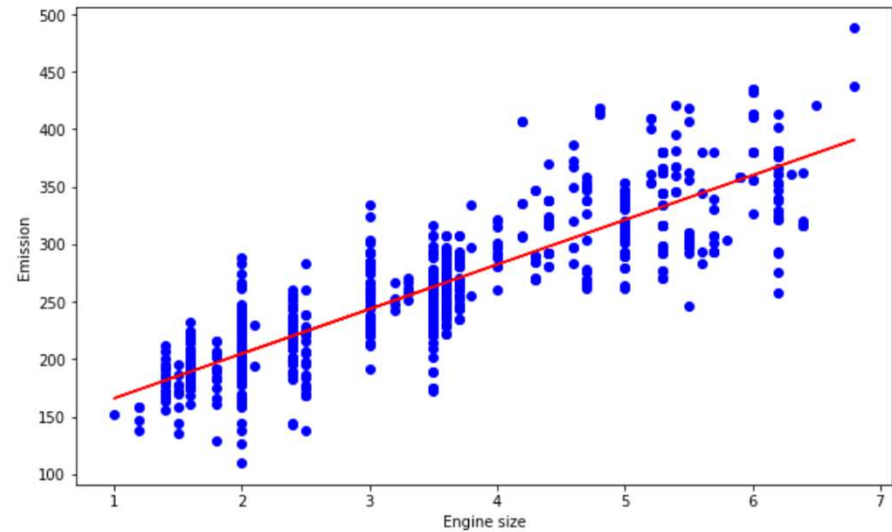
- **Sürekli** değerleri tahmin etmek için regresyon algoritmaları kullanılır.
- Regresyon algoritmaları:
 - Linear Regression (Linear)
 - Polynomial Regression (Polinom)
 - Exponential Regression (Üstel)
 - Logistic Regression (Logistik)
 - Logarithmic Regression (Logaritmik)





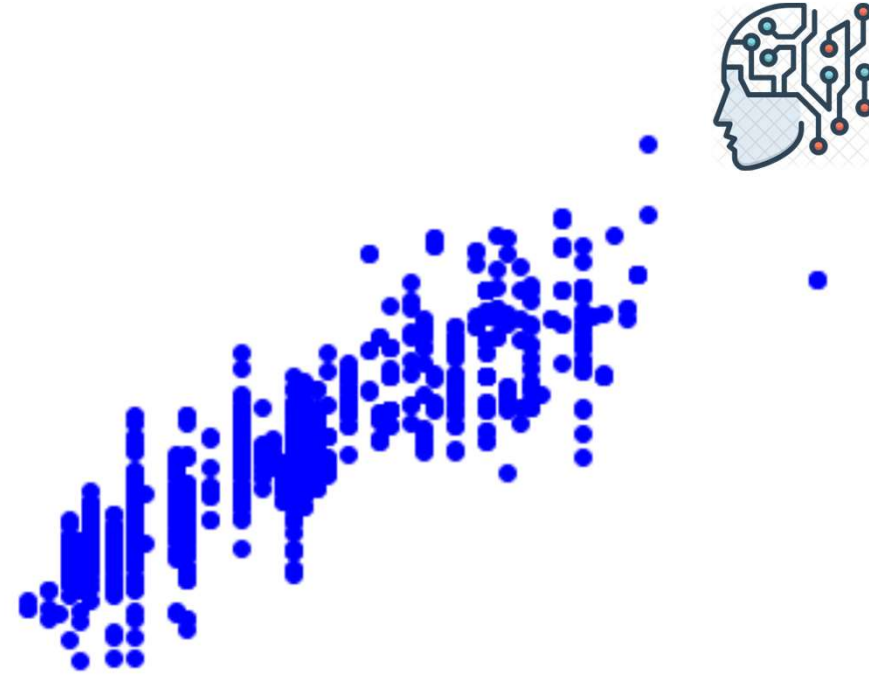
Linear Regression (Doğrusal Regresyon)

- Doğrusal regresyon, girdi özellikleri ile çıktı arasındaki ilişkiyi modelleyen **istatistiksel** bir yaklaşımdır. **Girdi** özelliklerine **bağımsız değişkenler** denir ve **çıktıya bağımlı değişken** adı verilir. Buradaki amacımız, girdi özelliklerine göre çıktının değerini, optimal katsayıları ile çarparak tahmin etmektir.
- Doğrusal regresyonun gerçek hayattan bazı örnekleri:
 - (1) Ürün satışlarını tahmin etmek.
 - (2) Ekonomik büyümeyi tahmin etmek.
 - (3) Petrol fiyatlarını tahmin etmek.
 - (4) Yeni bir arabanın emisyonunu tahmin etmek.
 - (5) Not ortalamasının üniversiteye girişler üzerindeki etkisi.
- **İki tür doğrusal regresyon vardır:**
 - 1. Basit Doğrusal Regresyon
 - 2. Çok Değişkenli Doğrusal Regresyon



Basit Doğrusal Regresyon

- Basit doğrusal regresyonda, çıktı / bağımlı değişkeni **yalnızca bir giriş özelliğine göre tahmin ederiz**. Basit doğrusal regresyon şu şekilde verilir:
- **Adım adım uygulama:**
 - a. Gerekli **kütüphaneler** aktarılır: Hesaplamalar için çeşitli kütüphaneler kullanacağımızdan, bunları aktarmamız gerekiyor.
 - b. **Veritabanı dosyası** okunur.
 - c. Değerler tahmin edilirken dikkate almak istenilen **özellikleri seçilir**.
 - d. Verilerin **grafiği** çizilir.
 - e. Bir modelin doğruluğunu kontrol etmek için, **veriler eğitim ve test veri setlerine ayrılır**.
 - f. **Model eğitilir**: Modelin eğitimi uygun regresyon çizgimi için katsayıların nasıl bulabileceğidir.
 - g. En uygun **çizgi** çizilir.
 - h. **Test veri seti** için bir **tahmin fonksiyonu** bulunur.
 - i. Test verilerinin **doğruluğu kontrol edilir**: Gerçek değerleri veri setindeki tahmin edilen değerlerle **karşılaştırılarak** bir modelin doğruluğu kontrol edilebilir.



$$Y = b_0 + b_1 * X_1$$

Where,

b_0 = constant or y – intercept of line

b_1 = coefficient of input feature

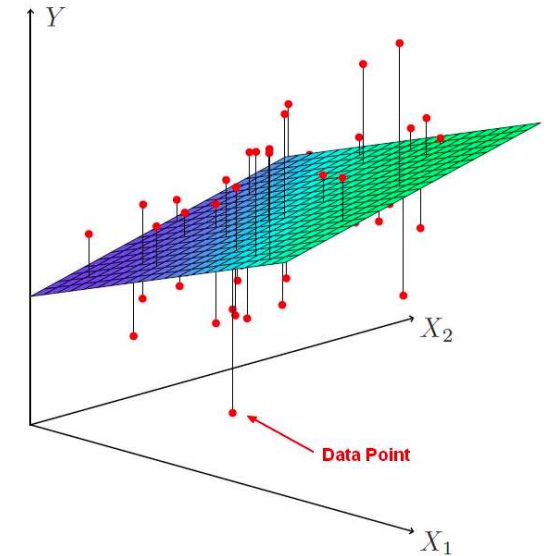
X_1 = input feature on which output is based

Y = output



Çok Değişkenli Doğrusal Regresyon

- Basit doğrusal regresyonda, çıktı özelliğinin değerinin tahmin edilmesi için yalnızca bir giriş özelliği dikkate alınır. Bununla birlikte, Çok Değişkenli Doğrusal Regresyonda, **çıkta birden fazla giriş özelliğine göre tahmin edilebilir.**
- **Adım adım uygulama:**
 - a. Gerekli **kütüphaneler** aktarılır.
 - b. **Veri tabanı dosyası** okunur.
 - c. X ve Y tanımlanır: X, istediğimiz bir **girdi**, Y ise **çıkta**nın değeri.
 - d. Veriler bir **test ve eğitim veri kümesine** bölünür.
 - e. **Model eğitilir.**
 - f. Giriş özelliklerinin **katsayılarını bulunur**: Şimdi hangi özelliğın çıkta değışkeni üzerinde daha önemli bir etkiye sahip olduğunu bilmemiz gerekiyor. Bunun için katsayı değeri bulunur. Negatif katsayının çıkta üzerinde ters etkisi olduğu anlamına geldiğı unutulmamalıdır. Özelliklerin değeri artarsa, çıkta değeri düşer.
 - g. Değerler **tahmin** edilir.
 - h. **Modelin doğruluğı test edilir**: Şimdi, burada basit ve çok değışkenli doğrusal regresyon için aynı veri seti kullanıldığına dikkat edilir. Çok değışkenli doğrusal regresyonun doğruluğının, basit doğrusal regresyonun doğruluğundan çok daha iyi olduğu fark edilebilir.



$$Y = b_0 + b_1 * X_1 + b_2 * X_2 + \dots + b_n * X_n$$

Where,

b_0 = constant or y intercept of line

b_1, b_2, b_n = coefficient of input feature

X_1, X_2, X_n = input features

Y = output



Polynomial Regression

- Polinom regresyonu, genelleştirilmiş bir doğrusal regresyon durumu olarak kabul edebilirsiniz. Çıktı ve girdiler arasındaki **polinom bağımlılığını** ve dolayısıyla **polinom tahmini** regresyon fonksiyonunu varsayarsınız.

$$Y = b_0 + b_1X + b_2X^2 + \dots + b_nX^n$$

Where,

Y = output

X = input feature

b_0, b_1, b_n = coefficients

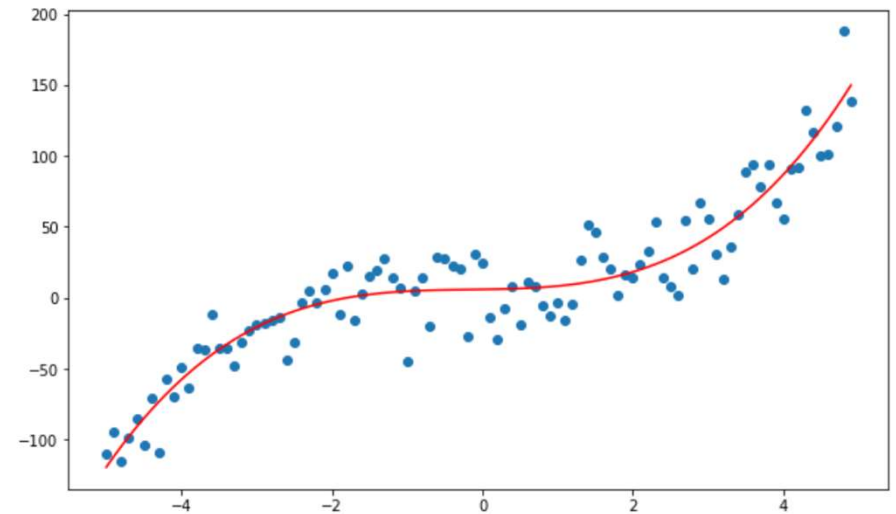
$$\theta = (X^T X)^{-1} \cdot (X^T y)$$

Yukarıdaki denklemde:

θ : onu en iyi tanımlayan hipotez parametreleri.

X : her örneğin giriş özelliği değeri.

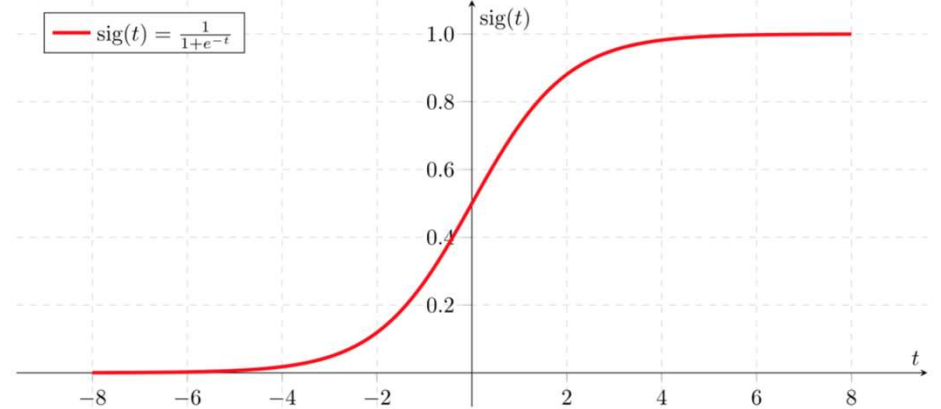
Y : Her örneğin çıktı değeri.





Logistic Regression

- Lojistik regresyon, temel bir sınıflandırma tekniğidir. Doğrusal sınıflandırıcılar grubuna aittir ve biraz **polinom** ve **doğrusal** regresyona benzer. Lojistik regresyon **hızlıdır** ve nispeten **karmaşık değildir** ve sonuçları yorumlamanız sizin için uygundur. Esasen ikili sınıflandırma için bir yöntem olmasına rağmen, çok sınıflı problemlere de uygulanabilir.
- Lojistik regresyonun ne olduğunu ve nasıl çalıştığını anlamak için **sigmoid işlevini** ve **doğal logaritma işlevini** anlamanız gerekir.
- **Sigmoid işlevi**, etki alanının çoğunda **0 veya 1'e çok yakın değerlere sahiptir**. Bu gerçek onu sınıflandırma yöntemlerinde uygulamaya uygun hale getirir.
- Bu görüntü, bazı değişken x 'nin sigmoid fonksiyonunu (veya S-şekilli eğri) gösterir.

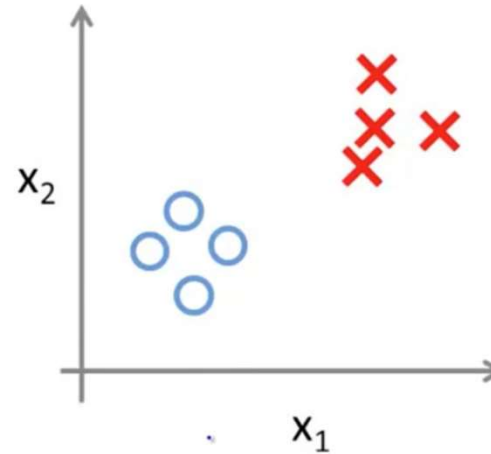




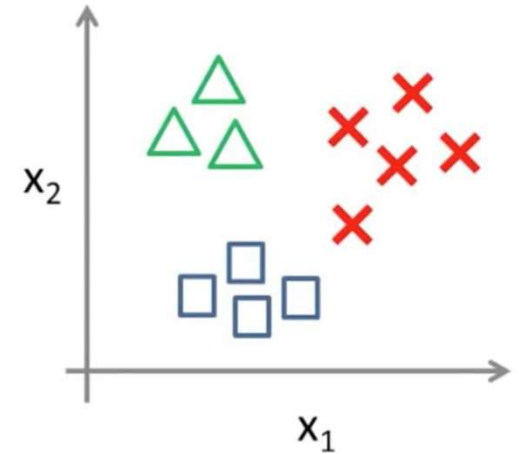
2- Sınıflandırma Algoritmaları

- Bir dizi ögenin **sınıfını** veya **kategorisini** tahmin etmek için sınıflandırma algoritmaları kullanılır.
- **Sınıflandırma algoritmaları:**
 - Decision Trees (Karar Ağaçları)
 - K-Nearest Neighbors (K-En Yakın komşular)
 - Random Forest (Rastgele Orman)
 - Support Vector Machine (Destek Vektör Makinesi)
 - Naive Bayes

Binary classification:

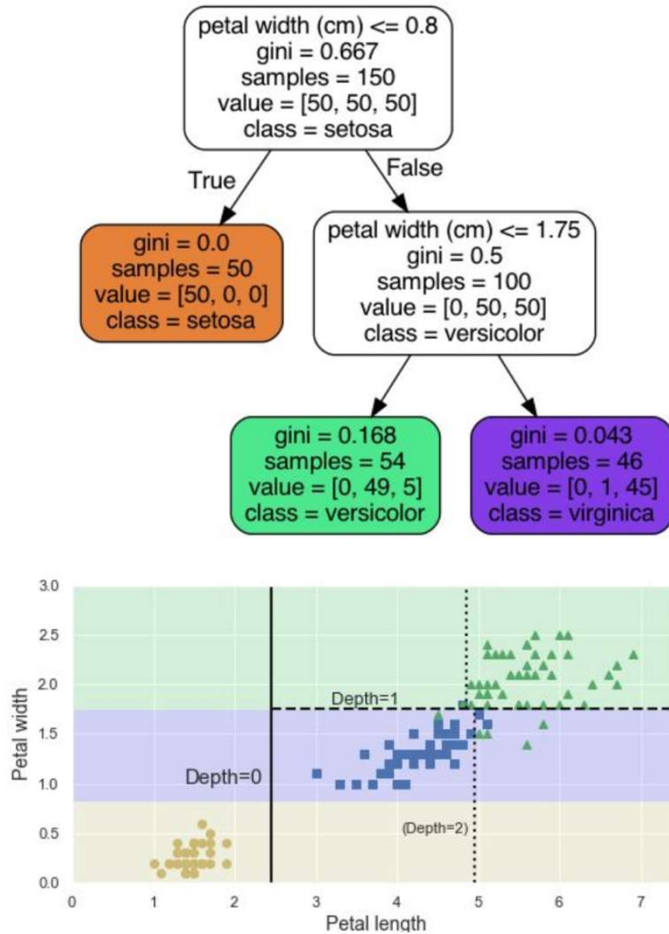


Multi-class classification:





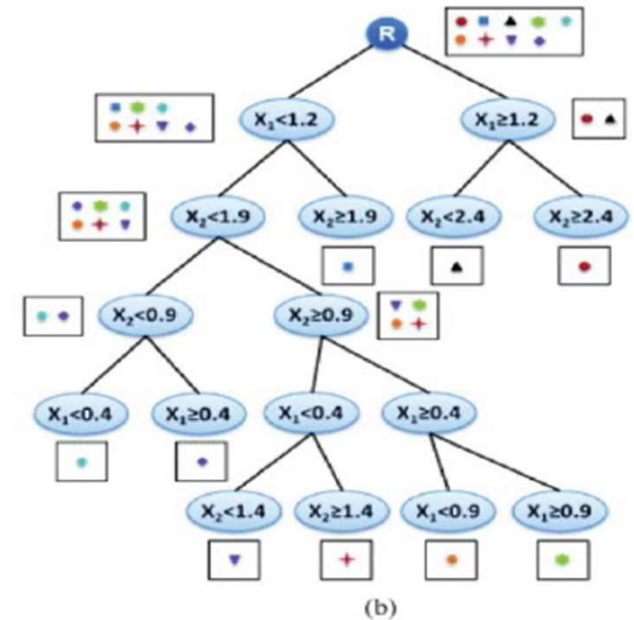
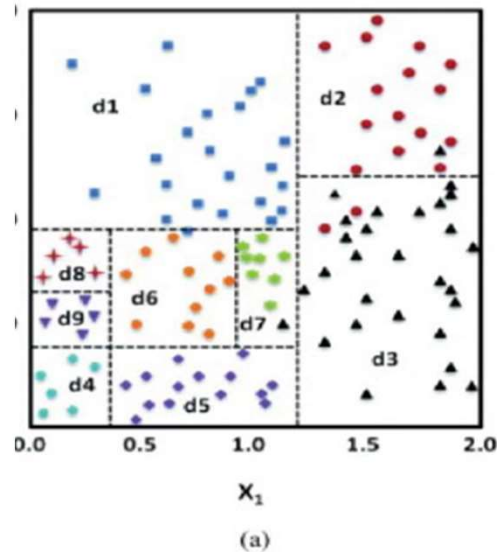
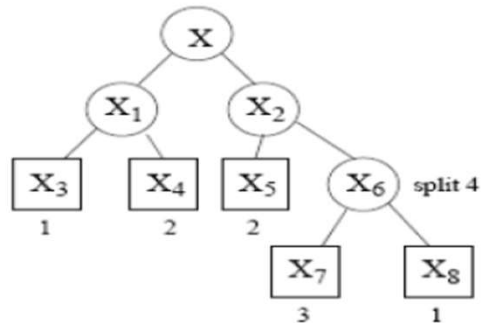
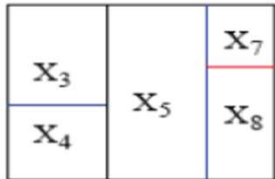
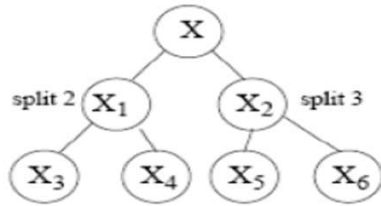
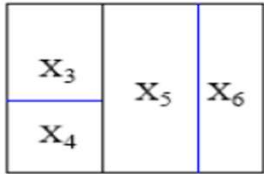
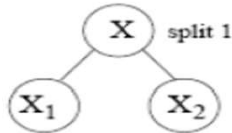
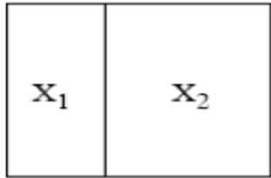
Decision Trees (Karar Ağaçları)



- Karar ağacı öğrenmesi, bir karar ağacını, bir öğeyle ilgili (dallarda temsil edilen) **gözlemlerden** öğenin **hedef** değeri (yapraklarda temsil edilen) ile ilgili **sonuçlara** gitmek için bir **tahmin modeli** olarak kullanır. İstatistik, veri madenciliği ve makine öğrenmesinde kullanılan öngörülü modelleme yaklaşımlarından biridir. **Hedef değişkenin ayrı bir değer kümesi alabileceği ağaç modellerine sınıflandırma ağaçları** denir; bu ağaç yapılarında yapraklar sınıf etiketlerini ve dallar bu sınıf etiketlerine yol açan özelliklerin birleşimlerini temsil eder. **Hedef değişkenin sürekli değerler alabileceği karar ağaçlarına (tipik olarak gerçek sayılar) regresyon ağaçları** denir. Karar analizinde, bir karar ağacı, kararları ve karar almayı görsel ve açık bir şekilde temsil etmek için kullanılabilir. Veri madenciliğinde, bir karar ağacı verileri tanımlar, ancak sonuçta ortaya çıkan sınıflandırma ağacı karar verme için bir girdi olabilir.
- Karar ağaçları metodu, giriş verisinin bir algoritma yardımıyla **gruplara bölünerek** tüm elemanlarının **aynı sınıf etiketine sahip olması** için yapılan **kümeleme** işlemidir. Giriş verisinin bir kümeleme algoritması yardımıyla tekrar tekrar gruplara bölünmesine dayanır. Grubun tüm elemanları aynı sınıf etiketine sahip olana kadar kümeleme işlemi derinlemesine devam eder.

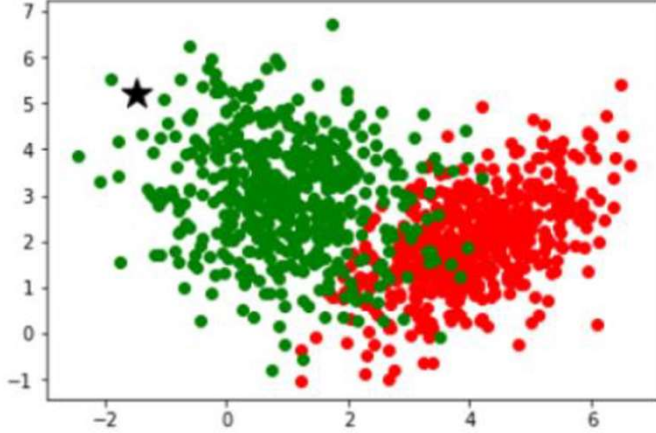


Decision Trees (Karar Ağaçları)





K-Nearest Neighbors (K - En Yakın Komşu)

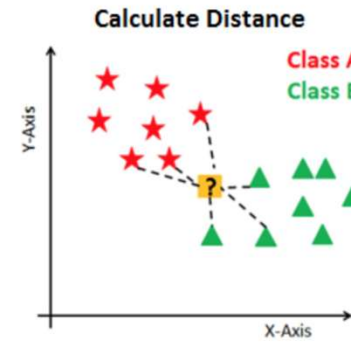
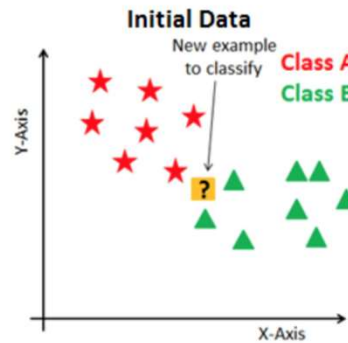
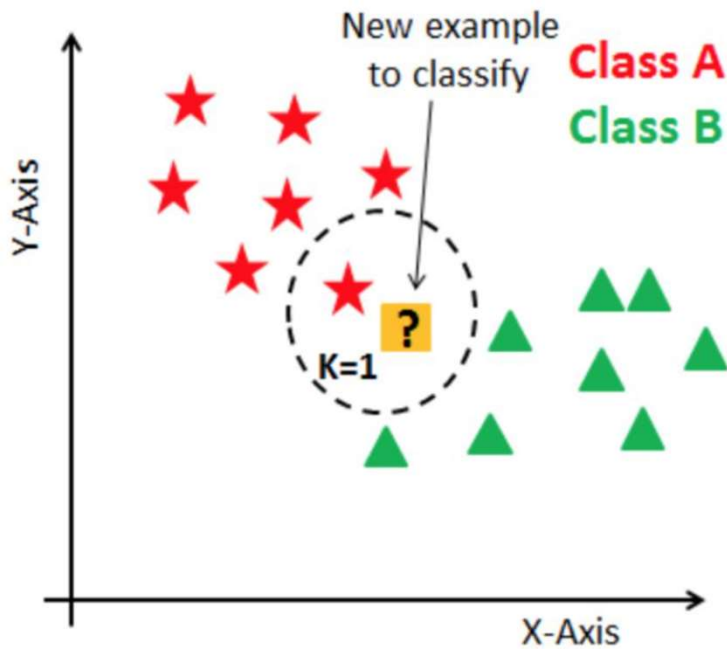


- K-En Yakın Komşu, aynı zamanda KNN olarak da bilinen, **regresyon ve sınıflandırma** problemleri için kullanılabilen denetimli bir öğrenme algoritmasıdır. KNN, makine öğreniminde sınıflandırma problemleri için yaygın olarak kullanılmaktadır. KNN, **birbirine yakın düşen her veri noktasının aynı sınıfa girdiğini varsayan bir prensiple çalışır**. Bu, benzer şeylerin birbirine yakın olduğu anlamına gelir.
- Bir tahmin yapmak istediğimizde, tüm veri setinde en yakın komşuları arar. Algoritmanın çalışmasında bir **K değeri** belirlenir. Bu K değerinin anlamı **bakılacak eleman sayısıdır**. Bir değer geldiğinde en yakın K kadar eleman alınarak gelen **değer arasındaki uzaklık hesaplanır**. Uzaklık hesaplama işleminde genelde **Öklid fonksiyonu** kullanılır. Öklid fonksiyonuna alternatif olarak **Manhattan, Minkowski** ve **Hamming** fonksiyonları da kullanılabilir. Uzaklık hesaplandıktan sonra **sıralanır** ve gelen değer uygun olan sınıfa atanır. İlgili uzaklıklardan **en yakın k komşu** ele alınır. Öznitelik değerlerine göre k komşu veya komşuların sınıfına atanır. Seçilen sınıf, tahmin edilmesi beklenen gözlem değerinin sınıfı olarak kabul edilir. Yani yeni veri etiketlenmiş (label) olur.

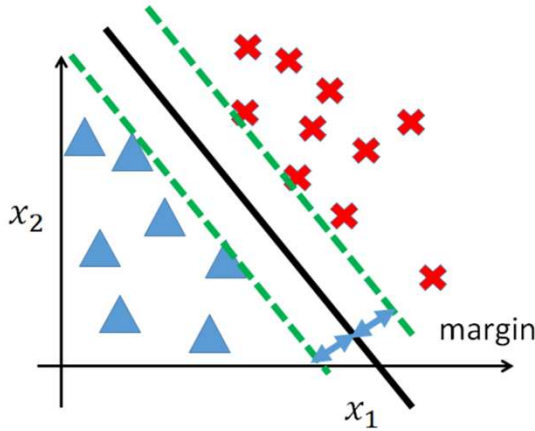


K-Nearest Neighbors (K - En Yakın Komşu)

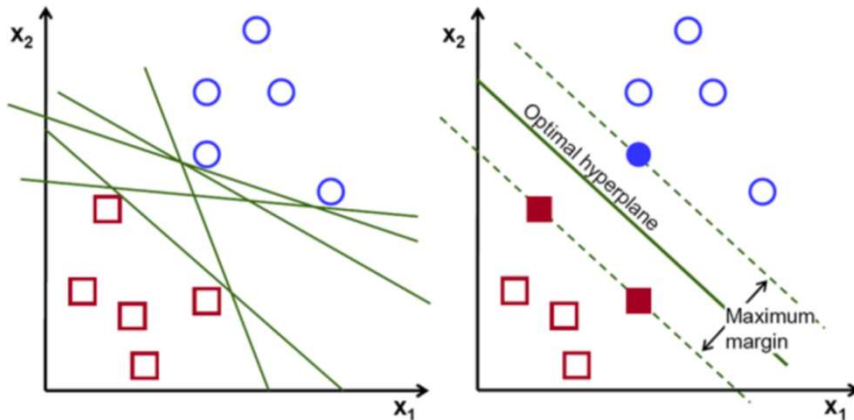
1. Uzaklığı Hesapla
2. Yakın Komşuları Bul
3. Etiket/Sınıf için Oy Ver



Support Vector Machine (Destek Vektör Makineleri)



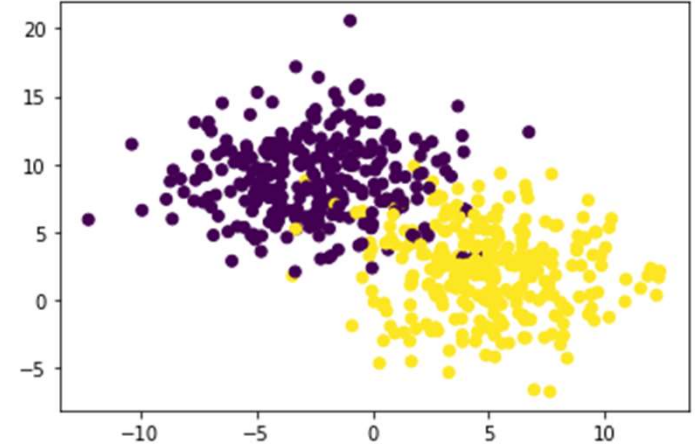
- **Sınıflandırma** ve **Regresyon** için kullanılan ilgili denetimli öğrenme yöntemleri kümesidir. Her biri iki kategoriden birine ait olarak işaretlenmiş bir dizi **eğitim örneği verildiğinde**, bir DVM eğitim algoritması yeni bir örneğin bir kategoriye mi yoksa diğerine mi düştüğünü tahmin eden bir model oluşturur. Bir DVM eğitim algoritması olasılık dışı, ikili, doğrusal bir sınıflandırıcıdır, ancak Platt ölçeklendirme gibi yöntemler olasılıklı bir sınıflandırma ayarında DVM'yi kullanmak için mevcuttur.
- Doğrusal sınıflandırmayı gerçekleştirmenin yanı sıra, DVM'ler, girdilerini yüksek boyutlu özellik alanlarına dolaylı olarak eşleyerek **çekirdek hilesi** adı verilen kullanarak doğrusal olmayan bir sınıflandırma gerçekleştirebilir.
 - Sınıflar arasındaki birbirine **en yakın ikili seçilir**. Bu noktalara destek vektörleri isimleri veriyoruz.
 - Destek vektörlerinden geçecek şekilde **doğrular çizilir**. Bu doğrulara **sınır çizgisi** adı verilir.
 - Her doğruya eşit uzakta çizilen doğruya **karar doğrusu** adı verilir. Karar doğrusuna hiper düzlem de denir.
 - DVM'lerde sınıflar **+1 ve -1** olarak etiketlenir. Karar doğrusunun üst kısmında kalan doğru denklemi, **$wx + b = 1$** , altında kalan doğru denklemi ise **$wx + b = -1$** olarak belirlenir. Karar doğrusu denklemi ise, **$wx + b = 0$** olur.





Naive Bayes

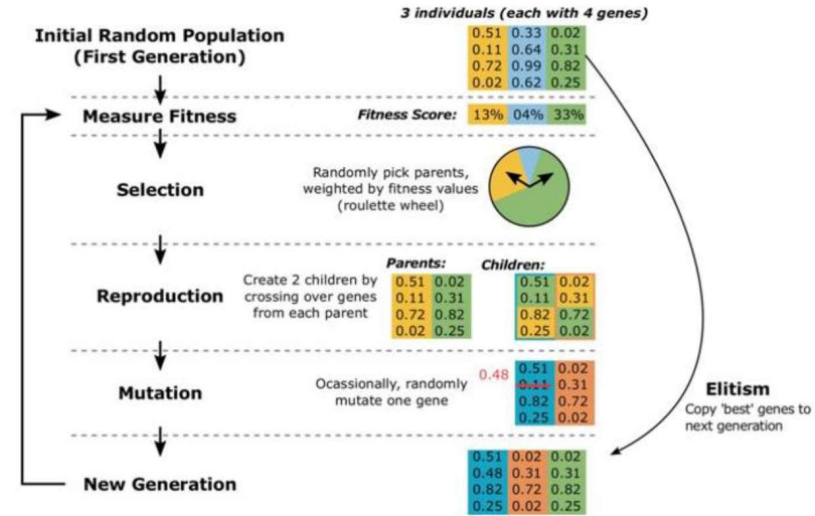
- Naive Bayes, sınıflandırıcılar oluşturmak için basit bir tekniktir: sınıf etiketlerinin bazı sonlu kümelerden çizildiği, özellik değerlerinin vektörleri olarak temsil edilen, problem örneklerine sınıf etiketleri atayan modeller. Bu tür sınıflandırıcıları eğitmek için tek bir algoritma yoktur, ancak ortak bir ilkeye dayanan bir algoritma ailesi vardır: tüm Naive Bayes sınıflandırıcıları, belirli bir özelliğin değerinin, sınıf değişkeni göz önüne alındığında diğer herhangi bir özelliğin değerinden **bağımsız olduğunu varsayar**. Örneğin bir meyve kırmızı, yuvarlak ve yaklaşık 10 cm çapındaysa elma olarak düşünülebilir. Bir Naive Bayes sınıflandırıcısı, renk, yuvarlaklık ve çap özellikleri arasındaki olası korelasyonlardan **bağımsız** olarak, bu özelliklerin her birinin bu meyvenin bir elma **olma olasılığına bağımsız olarak katkıda bulunduğunu düşünür**.
- Bayes ağı, düşünce ya da oluşan kanı veya yönlendirilmiş olasılıklı bir grafik modeldir. Örneğin, bir Bayes ağı, hastalıklar ve semptomlar arasındaki olasılık ilişkilerini temsil edebilir. Belirtiler verildiğinde, ağ çeşitli hastalıkların varlığının olasılıklarını hesaplamak için kullanılabilir. Çıkarım ve öğrenme yapan etkin algoritmalar mevcuttur. Konuşma sinyalleri veya protein dizileri gibi değişken dizilerini modelleyen Bayes ağlarına **dinamik Bayes şebekeleri** denir. Belirsizlik altında karar problemlerini temsil edebilen ve çözebilen Bayes ağlarının genellemelerine **etki diyagramları** denir.





Genetic algorithm (Genetik Algoritmalar)

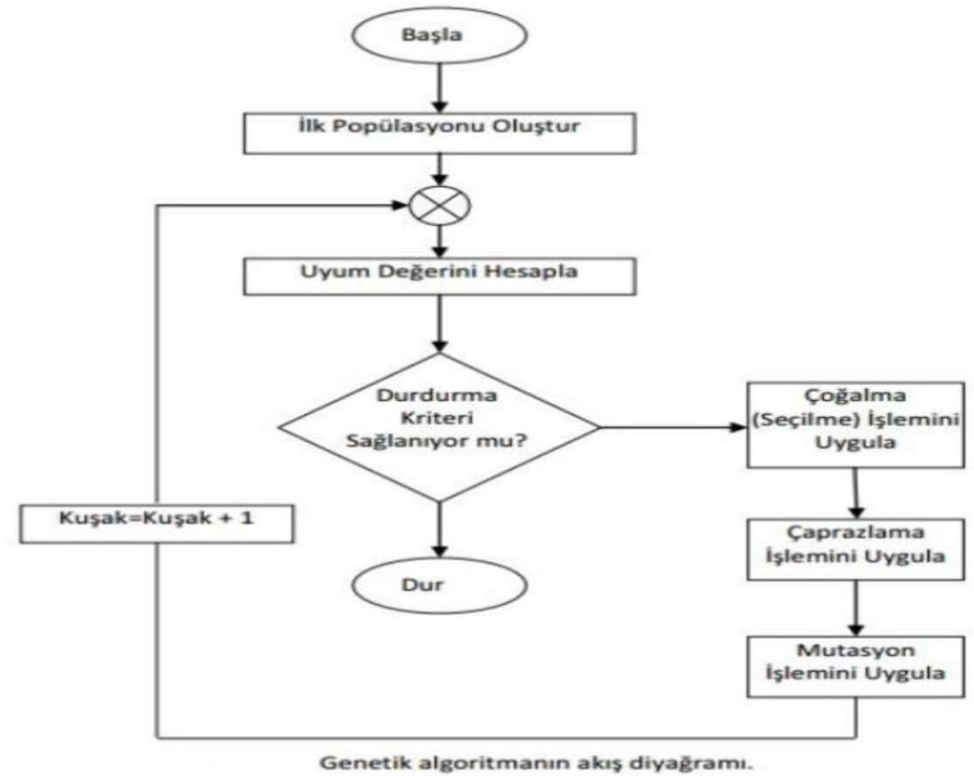
- Genetik algoritma (GA), belirli bir soruna iyi çözümler bulma umuduyla yeni **genotipler** üretmek için **mutasyon** ve **çaprazlama** gibi yöntemleri kullanarak **doğal seleksiyon sürecini taklit eden bir arama algoritması ve sezgisel tekniktir**.
- GEN:** Kendi başına anlamı olan ve genetik bilgi taşıyan en küçük genetik birimdir. Bir gen A, B gibi bir karakter olabileceği gibi 0 veya 1 ile ifade edilen bir bit veya bit dizisi olabilir. Örneğin bir cismin x koordinatındaki yerini gösteren bir gen 101 şeklinde ifade edilebilir
- KROMOZOM:** Bir ya da birden fazla genin bir araya gelmesiyle oluşurlar Probleme ait tüm bilgileri içerirler. Kromozomlar toplumdaki bireyler yada üyelere karşılık gelirler. Ele alınan problemde alternatif çözüm adayıdır.
- POPÜLASYON:** Kromozomlar veya bireyler topluluğudur. Popülasyon üzerinde durulan problem için alternatif çözümler kümesidir. Aynı anda bir popülasyonda ki birey sayısı sabit ve probleme göre kullanıcı tarafından belirlenir. Zayıf olan bireylerin yerini kuvvetli yeniler almaktadır.





Genetic algorithm (Genetik Algoritmalar)

- **Genetik Algoritmalar Nasıl Çalışır?**
- **1. Adım:** Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur. Çözüm grubuna biyolojideki benzerliği nedeniyle **populasyon** çözümlerin kodları da **kromozom** olarak adlandırılır. Bu adıma populasyonda bulunan **birey sayısı belirleyerek başlanır**. Bu sayı için bir standart yoktur. Genel olarak önerilen 100 - 300 aralığında bir büyüklüktür. Büyüklük seçiminde yapılan işlemlerin karmaşıklığı ve aramanın derinliği önemlidir. Populasyon bu işlemten sonra rasgele oluşturulur.
- **2. Adım:** Her **kromozomun ne kadar iyi olduğu** bulunur. Kromozomların ne kadar iyi olduğunu bulan fonksiyona **uygunluk fonksiyonu** denir. Bu fonksiyon işletilerek kromozomların uygunluklarının bulunmasına ise **hesaplama** (evaluation) adı verilir. Bu fonksiyon genetik algoritmanın beynini oluşturmaktadır. GA da probleme özel çalışan tek kısım bu fonksiyondur. Çoğu zaman GA'nın başarısı bu fonksiyonun verimli ve hassas olmasına bağlı olmaktadır.





Genetic algorithm (Genetik Algoritmalar)

Cut Point

P1 = [2 1 3 4 5 6 7]
P2 = [4 3 1 2 5 7 6] → O1 = [2 1 3 2 5 7 6]
O2 = [4 3 1 4 5 6 7]

Example 2. Partially Mapped Crossover

Cut Point 1

Cut Point 2

P1 = [2 1 3 4 5 6 7]
P2 = [4 3 1 2 5 7 6] → O1 = [4 3 1 2 5 6 7]
O2 = [2 1 3 4 5 7 6]

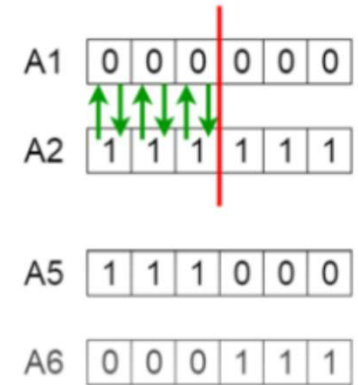
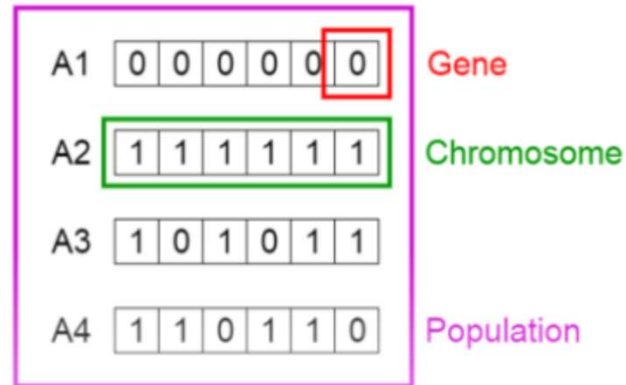
Cut Point 1

P1 = [2 1 3 4 5 6 7]
P2 = [4 3 1 2 5 7 6] → O1 = [2 1 4 3 5 7 6]
O2 = [4 3 2 1 5 6 7]

Cut Point 1

Cut Point 2

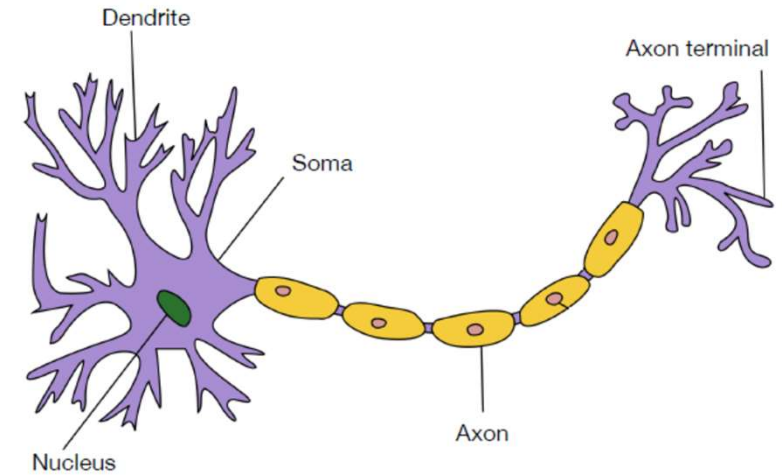
P1 = [2 1 3 4 5 6 7]
P2 = [4 3 1 2 5 7 6] → O1 = [3 4 5 1 2 7 6]



Artificial Neural Networks (Yapay Sinir Ağları)



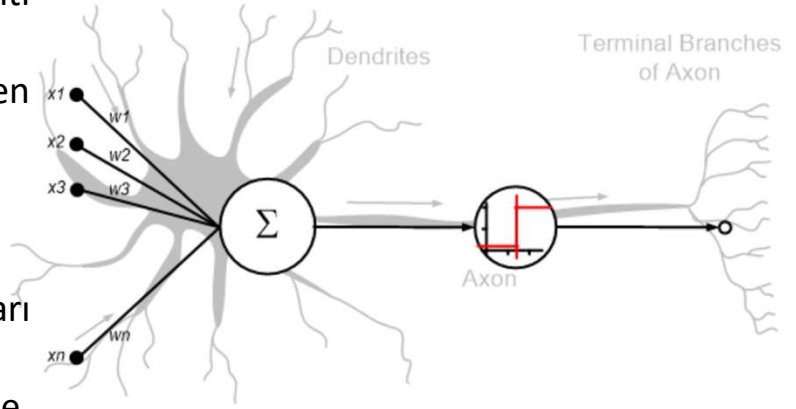
- **İnsan beyni:**
- Beynin nasıl çalıştığını daha yeni anlamaya başladık; daha yolun başındayız. Akıllı makineler yapmanın bir yolu, insan beynini özellikle organizasyonel prensiplerini taklit etmeye çalışmaktır.
- **Beyindeki Biyolojik Nöronlar:**
- İnsan beyni yaklaşık 100 milyar nörondan oluşur.
- **Dendritler:** hücrelere elektrik sinyalleri taşıyan sinir lifleri
- **Hücre gövdesi:** girdilerinin doğrusal olmayan bir işlevini hesaplar
- **Akson:** elektrik sinyalini hücre gövdesinden diğer nöronlara taşıyan tek uzun lif. Nöronun aksonu, diğer birçok nöronun dendritlerine bağlıdır. Nöronlar, dendritlerden elektrik sinyalleri alır ve bunları aksona gönderir.
- **Sinaps:** gücü hücreye girişi etkileyen kimyasal bir bağlantıyı düzenleyen, bir hücrenin aksonu ile diğerinin dendriti arasındaki temas noktası.



Artificial Neural Networks (Yapay Sinir Ağları)

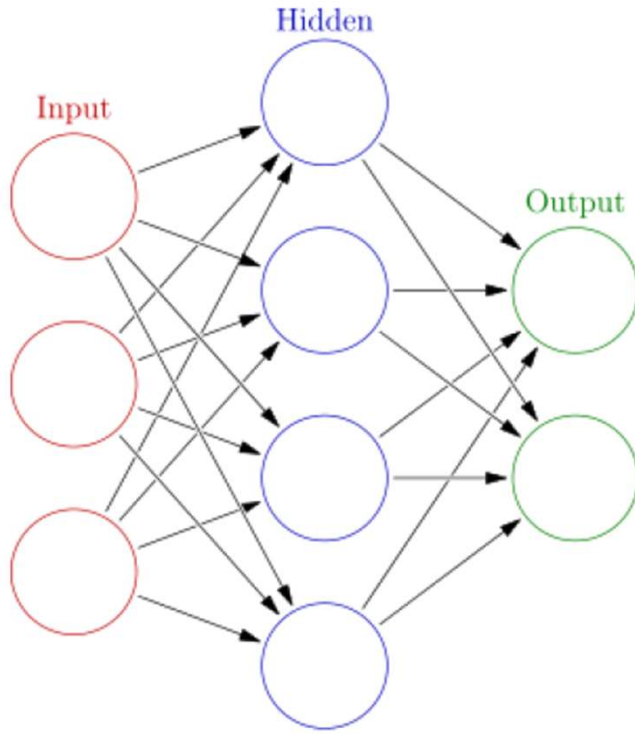


- İnsan beyninden ilham alan hesaplama model, Yapay Sinir Ağları:
- Basit işlem birimlerinden (nöronlar) oluşan büyük ölçüde paralel, dağıtılmış sistem
- Edinilen bilgiyi depolamak için nöronlar arasındaki sinaptik bağlantı güçleri kullanılır.
- Bilgi, ağ tarafından bir öğrenme süreci aracılığıyla çevresinden edinilir.
- Örneklerden öğrenmek: **etiketli** veya **etiketsiz**
- **Adaptivite**: bir şeyler öğrenmek için bağlantı güçlerini değiştirmek
- **Doğrusal olmama**: doğrusal olmayan aktivasyon fonksiyonları gereklidir
- **Hata toleransı**: Nöronlardan veya bağlantılardan biri hasar görürse, tüm ağ hala oldukça iyi çalışıyor.
- Bu nedenle, karakterize edilen sorunlar için klasik çözümlerden daha iyi alternatifler olabilirler: Yüksek boyutluluk, gürültülü, kesin olmayan veya eksik veriler; ve açıkça ifade edilmiş matematiksel bir çözüm veya algoritmanın olmaması

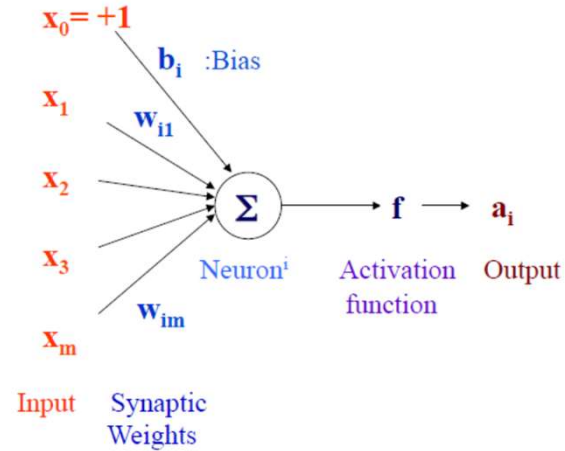




Artificial Neural Networks (Yapay Sinir Ağları)



- YSA yaklaşımının asıl **amacı**, problemleri bir insan beyninde olduğu gibi çözmektir. Bununla birlikte, zamanla dikkat, belirli görevlerin yerine getirilmesine ve biyolojiden sapmalara yol açtı. Yapay sinir ağları, **bilgisayar görme, konuşma tanıma, makine çevirisi, sosyal ağ filtreleme, oyun tahtası ve video oyunları ve tıbbi teşhis** gibi çeşitli görevlerde kullanılmıştır.
- Derin öğrenme** yapay bir sinir ağında **birden fazla gizli katmandan oluşur**. Bu yaklaşım, insan beyninin ışığı ve sesi görme ve işleme biçimini modellemeye çalışır. Derin öğrenmenin bazı başarılı uygulamaları **bilgisayarla görme ve konuşma tanımadır**.



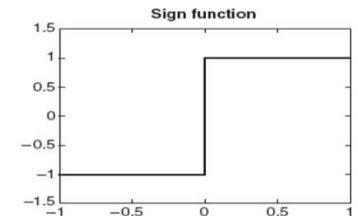
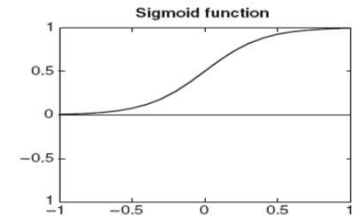
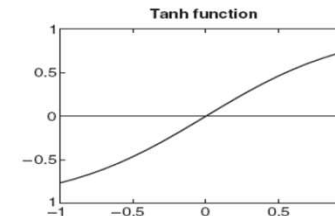
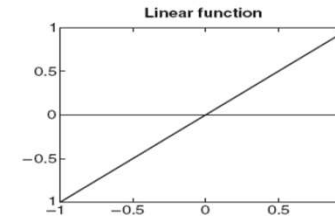
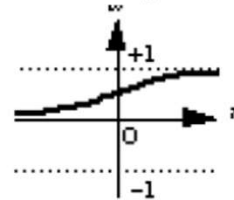
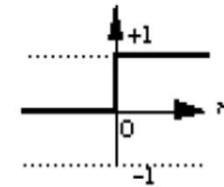
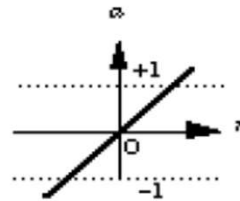
Artificial Neural Networks (Yapay Sinir Ağları)



- Aktivasyon fonksiyonları:
- Nöronun çıktısının genliğini sınırladığı için limit işlevi olarak da adlandırılır.

Many types of activations functions are used:

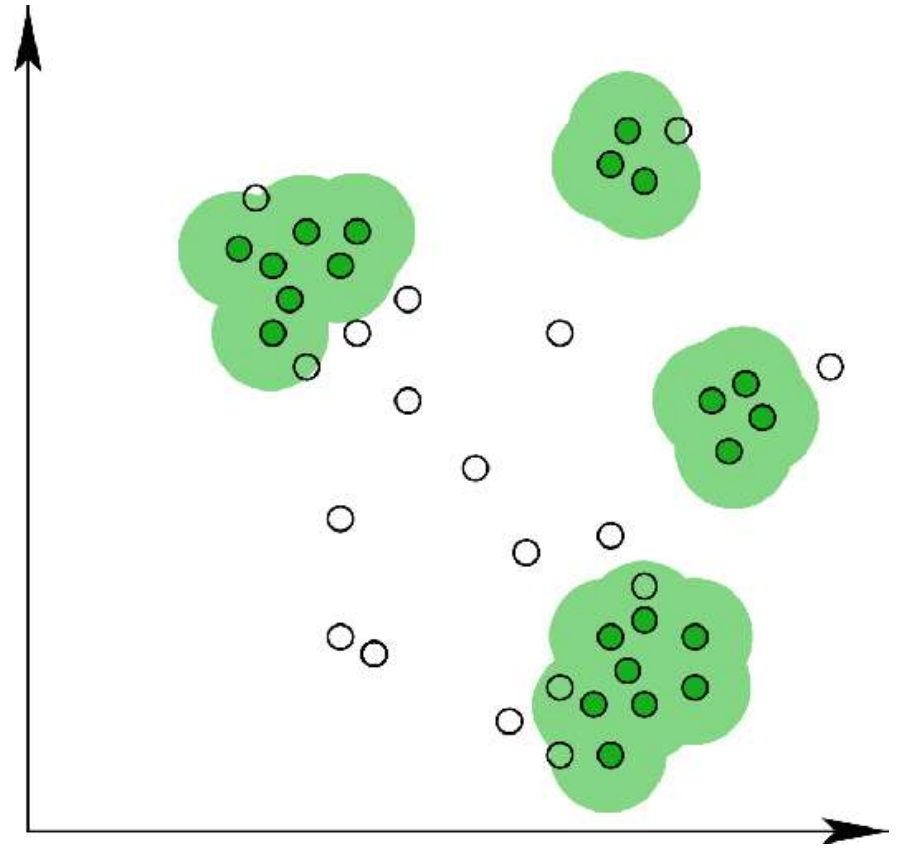
- linear: $a = f(n) = n$
- threshold: $a = \begin{cases} 1 & \text{if } n \geq 0 \\ 0 & \text{if } n < 0 \end{cases}$
(hardlimiting)
- sigmoid: $a = 1/(1+e^{-n})$
- ...



3 – Kümeleme

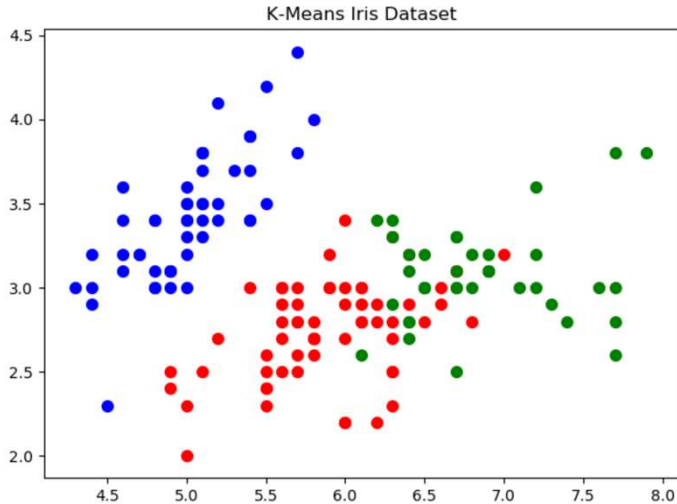


- Özetlemek veya verileri yapılandırmak için kümeleme algoritmaları kullanılır.
 - K-Means





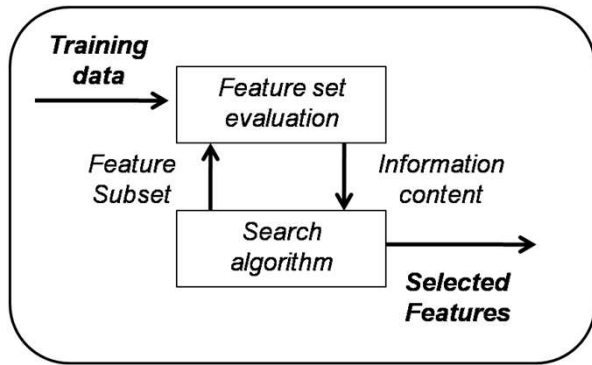
K-Means Clustering (K-Ortalama Kümeleme)



- N adet veri nesnesinden oluşan bir veri kümesini **giriş parametresi** olarak verilen **K adet kümeye bölümlenektir**. Amaç, gerçekleştirilen bölümlenme işlemi sonunda elde edilen kümelerin, **küme içi benzerliklerinin maksimum ve kümeler arası benzerliklerinin ise minimum olmasını sağlamaktır**.
- Öncelikle her kümenin merkez noktasını veya ortalamasını temsil etmek üzere **K adet nesne rastgele seçilir**. Kalan diğer nesneler, kümelerin ortalama değerlerine olan uzaklıkları dikkate alınarak en benzer oldukları **kümelere** dahil edilir. Daha sonra, her bir kümenin ortalama değeri hesaplanarak yeni **küme merkezleri belirlenir** ve tekrar nesnelerin **merkeze uzaklıkları incelenir**. Herhangi bir değişim olmayıncaya kadar **algoritma tekrarlamaya** devam eder.
 - Algoritma temel olarak 4 aşamadan oluşur:
 - 1. Küme merkezlerinin belirlenmesi
 - 2. Merkez dışındaki verilerin **mesafelerine** göre kümelendirilmesi
 - 3. Yapılan kümelendirmeye göre **yeni merkezlerin belirlenmesi** (veya eski merkezlerin yeni merkeze kaydırılması)
 - 4. Kararlı hale (**stable state**) gelinene kadar 2. ve 3. adımların **tekrarlanması**.



Öznitelik Seçim Algoritmaları



- Cfs Subset Öznitelik Seçim Algoritması
- Correlation Attribute Öznitelik Seçim Algoritması
- One-R Öznitelik Seçim Algoritması
- Gain Ratio Öznitelik Seçim Algoritması
- Info Gain Öznitelik Seçim Algoritması
- ReliefF Öznitelik Seçim Algoritması

- Makine öğrenmesi algoritmaları sınıflandırma yaparken en önemli noktalardan biri **hangi değişkenlerin etkili olduğunu öğrenmektir**.
- Sınıflandırma algoritmalarında kullanılan değişken sayısının olabildiğince azaltılması ve sınıflandırma sonuçlarında bunu yaparken önemli bir düşüş olmaması son derece önemlidir.
- Veri setindeki bazı özniteliklerde önemli bilgi bulunmaz. Öznitelik seçimi, veri setindeki özniteliklerden **yeni bir öznitelik alt kümesi oluşturarak düzenlenir**.
- Öznitelik seçimi yapılmasındaki önemli noktalardan biri de, örnek sayısının düşük, nitelik sayısının ise fazla olduğu durumlarda makine öğrenmesi algoritmalarının işlem yapmasının zor olmasıdır.
- Öznitelik seçimi; filtre modeli, sarmal modeli ve gömülü modeli algoritmalarıdır.
 - Filtre modeli algoritmanın olumlu yönü sınıflandırma algoritmasından **bağımsız ve hızlı** olarak işlem yapmasıdır. Olumsuz yönü ise diğer yöntemlere göre **başarı oranının daha düşük seviyede olmasıdır**.
 - Sarmal modeli algoritmanın olumlu yönü **başarı oranını yüksek tutması ve uygulamasının kolay olmasıdır**. İşlem yükünün fazlalığından dolayı **yavaş çalışması** ise olumsuz yönüdür.
 - Gömülü modeli algoritmanın olumlu yönü, **başarı oranını yüksek tutarak işlem yükünü düşürmesidir**. Olumsuz yönü ise karar ağaçlarından **bağımsız olarak kullanılamamalarıdır**.



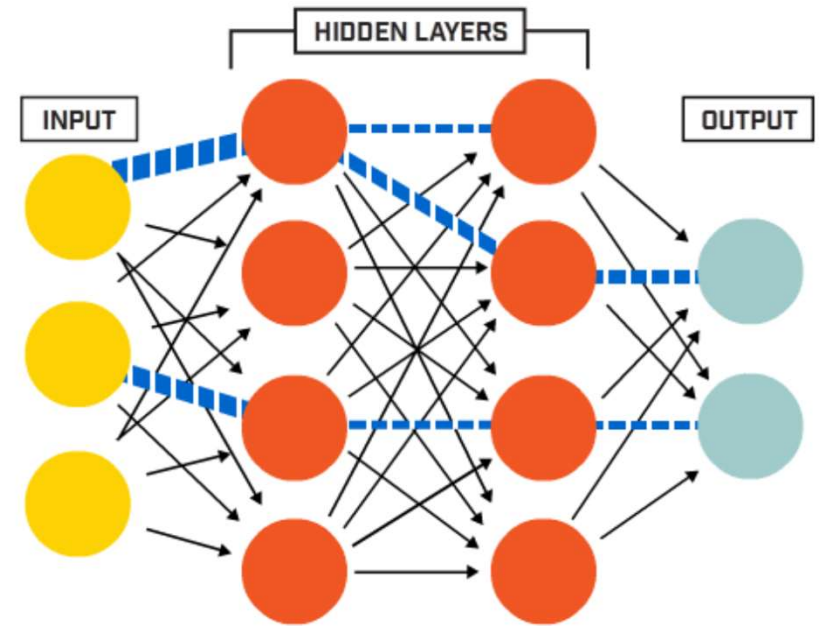
Sınıflandırmalar Algoritmaları için Performans Ölçüm Kriterleri

- doğru pozitif (DP) oranı,
- yanlış pozitif (YP) oranı,
- doğru negatif (DN) oranı,
- Yanlış negatif (YN) oranı,
- hassasiyet (precision),
- f-ölçütü (f-measure),
- alıcı işlem karakteristiği eğrisi (ROC: Receiver Operating Characteristic),
- kappa (κ) istatistiği,
- ortalama mutlak hata (MAE: Mean Absolute Error),
- kök ortalama kare hata (RMSE: Root Mean Square Error),
- Matthews korelasyon katsayısı (MCC: Matthews Correlation Coefficient) olarak sıralanır.
- Doğru Pozitif Oranı (Duyarlılık)
- Yanlış Pozitif Oranı
- Precision (Hassasiyet)
- F-Ölçütü
- Alıcı İşlem Karakteristiği Eğrisi (ROC: Receiver Operating Characteristic)
- Kappa (κ) İstatistiği
- Ortalama Mutlak Hata (MAE: Mean Absolute Error)
- Kök Ortalama Kare Hata (RMSE: Root Mean Square Error)
- Matthews Korelasyon Katsayısı (MCC: Matthews Correlation Coefficient)



Deep learning (Derin Öğrenme)

- Derin öğrenme modeli, verinin yapısına göre **hangi parametrelere ne ağırlık verileceğini kendisi keşfetmektedir.**
- Derin öğrenme yöntemi insan beyninin öğrenme ve karar verme yöntemlerinden esinlenerek geliştiriliyor, yani **nöronların hareketleri taklit ediliyor.**
- Bu da «Yapay Sinir Ağları» (**Artificial Neural Networks**) sayesinde oluyor.
- Derin öğrenme (deep learning) makine öğrenmesi için kullanılan farklı yöntemlerden biridir.



Kuantum Makine Öğrenmesi (Quantum Machine Learning)



Type of Algorithm			
		classical	quantum
Type of Data	classical	CC	CQ
	quantum	QC	QQ

- Son yıllarda veri analizi uygulamaları ve akıllı makineler ilgili araştırmalar güçlü bir şekilde yeniden ortaya çıkmıştır.
- Bu güçlenen ilgi kısmen klasik hesaplama yöntemlerindeki gelişmeler ve kısmen de **Kuantum Hesaplama (QC - Quantum Computing)** ve ilgili kuantum teknolojileri tarafından sunulan muazzam paralellik potansiyelinden kaynaklanmaktadır.
- Hesaplama yöntemlerindeki bu gelişmeler, Makine Öğrenmenin (ML – Machine Learning), veri güdümlü öğrenme ve kuantum destekli hesaplama yöntemleri, hizmet odaklı tamamen akıllı bir iletişim ağının isteklerini gerçekleştirmede güçlü bir potansiyele sahiptir.
- Ortaya çıkan insan ve makine arası bağlantıyı artırma paradigmasında, ağ düğümü sayısı ve veri trafiğinde önemli bir artış beklenmektedir.
- **Makine Öğrenmesi (ML)** ve **Quantum Hesaplama (QC)** yöntemleri hacimli verilerin verimli bir şekilde işlenmesine yönelik olarak, **Quantum ML (QML)** teknolojilerini sağlayan yeni bir çerçeve sunacaktır.

SEVDANUR GENÇ

SGENC@KASTAMONU.EDU.TR



/SEVDANURGENC