

- Makine Öğrenimi Öncesinde, Kural Tabanlı Sistemler vardı.

### Kural (Rule Base) Temelli Yaklaşım:

1. 100 metre ilerle
2. Sola dön
3. 25 metre ilerle
4. (sonraki adım için kuralları tekrar tanımlamak gerek)

### Makine Öğrenimi Yaklaşımı:

1. Veri (Değişen Kısım)
  2. Topa olan mesafeyi hesapla – En az mesafeye doğru etrafında dön – Bu yönde Doğru ilerle (Değişmez Kısım)
- Veriler, Kurallar -> Klasik Programlama -> Cevaplar
  - Veriler, Cevaplar -> Makine Öğrenimi -> Kurallar

### Makineler Öğrenebilir Mi?

- Makineler mevcut verilere bakarak onlardan öğrenebilir. Mevcut veriler arasındaki örüntüyü matematiksel algoritmalarla keşfedip, modelleyerek yeni gelen veriler için bir öngörü oluşturabilirler. Böylece subjektif değerlendirmeler dışında matematiksel algoritmalarla dayalı bir öngörü elde edebiliriz.
- “Bak elimde şu veriler var, sen bunlara bir bak, anlamlı ilişkileri, örüntüleri şu algoritmalarla dayanarak anla(öğren) ve bir yerde sakla(model). Elime yeni veri geldikçe bu verinin ne demek olduğunu, nasıl bir sonuç çıkartacağını sana soracağım.”
- Elimizdeki bu modele göre elimizde olmayan yeni veriler hakkında tahminlerde bulunabiliriz.

### Öğrenme Tipleri

1. Denetimli Öğrenme:
  - a. İşlenmemiş Veriler
  - b. Süpervizör
  - c. Eğitim Veri Seti/Bilinen Çıktılar -> Algoritma -> Hesaplatma
  - d. Çıktılar
2. Denetimsiz Öğrenme:
  - a. Eğitim Veri Seti olmadan makinenin anlamlı sonuçları kendisi çıkartması gerekmektedir.
  - b. İşlenmemiş Veriler

- c. Eğitim Veri Seti Yok/Bilinen Çıktılar-Cevaplar Yok -> Algoritma -> Hesaplama
  - d. Çıktılar
3. Yarı Denetimli Öğrenme:

## Öğrenme ve Problem Tipleri:

En yaygın problem tipleri; sınıflandırma, regresyon ve kümelenmedir.

**Sınıflandırma:** İçinde kedi, köpek, sincap, at gibi hayvan resimlerinin bulunduğu bir veri setindeki hayvanları doğru sınıflara ayırabilmek bir sınıflandırma problemidir.

**Regresyon:** ‘Değerler’ bulmaya yönelik problemlerin çözümü için kullanılan teknikleri içerir. Mesela Youtube hesabımıza yeni atacağımız videonun kaç ‘like’ alabileceği, kaç ‘izlenme’ yapabileceği vs. gibi nicelikleri bulmaya, tahmin etmeye yönelik çalışmalardır. Süreklilik gösterirler ve sayısal bir değer üretirler.

Sınıflandırma ve Regresyon yaklaşımları ‘denetimli öğrenme’ tekniği ve algoritmalarıyla çözülebilen problemlerdir.

**Kümelenme:** Verilerin hangi alanlarda toplandığı, kümelendiğine dair yapılan çalışmalardır.

Kümelenme sorunları ise, ‘denetimsiz öğrenme’ kategorisinde yer alır.

**Pekiştirmeli Öğrenme:** Çevre ile etkileşim alan, anlık gelişmelere göre öğrenen ve çözümler üreten bir algoritma tipidir.

Örneğin oyun oynayan yapay zeka, pekiştirmeli öğrenme algoritması kullanır ve oyundaki hamleleri ve kuralları öğrenerek tepkiler verir.

## En Yaygın Kullanılan Algoritma Teknikleri:

### Lineer Regresyon:

Regresyonda iki temel değişken tipi vardır, bağımlı ve bağımsız değişkenler.

Bağımsız değişkenin değeri değiştikçe, bağımlı değişkenin değerleri değişmektedir.

Diğer bir değişle, değerler arasındaki ilişkiyi matematiksel olarak bulmaya yönelik bir çalışmadır.

Örneğin kullanım şekli:

Bağımlı/Bağımsız ilişkideki verilerimizi **dağılım grafiği (scatter plot)** a alırız.

Elimizdeki bu verilerin arasına öyle bir düz çizgi çizelim ki, noktaların bu çizgiye olan uzaklıklarının toplam değeri en az olsun.

**İdeal Çizgi (Hatasız Çizgi):** Veri noktalarının herhangi birisinin çizgiyle arasında mesafe(hata) yoktur.

Ama gerçek hayattaki veriler böyle bir dağılım göstermezler.

Bu yüzden, öyle bir çizgi çizmeliyiz ki, bu noktalara olan uzaklıklarının toplamı (**hata toplamı**) en az olsun.

En az hata toplamı olan çizgi, bizim '**modelimizdir**'.

Makine Öğrenimi/Derin Öğrenmede amacımız ideal modeller oluşturmaktır.

## **Modellerin Değerlendirilmesi: Uydurma, Aşırı Uydurma, Uyduramama**

Uydurma: Fitting

Aşırı Uydurma: Over-fitting

Uyduramama: Under-fitting

Aşırı Uydurma: Model, eğitim verimizi ezberlemiştir. Dolayısıyla yeni veriler için istikrarsız tahminler üretir.

Uyduramama: Gerçek değerden çok uzaktır, aşırı genelleme yapmıştır. Yani hata toplamı çok yüksektir.

## **Lineer Regresyona Yakından Bakış:**

x,y grafiğinde, eğrinin y eksenini kestiği noktaya b dersek

$y = ax + b$  olmaktadır.

x(bağımsız değişken)

y(bağımlı değişken)

b(kesen/intercept)

Doğrudan x değiştirilmez, x'e yeni değerler verilir. Değiştirilebilir değerler a ve b parametreleridir. Yani bu eğriyi farklı pozisyonlarda çizebilmemiz için ancak 'a' ve 'b' değerleri ile oynayabiliriz.

$x=0$  iken,  $b$ 'nin aldığı değere '**bias**' denmektedir. Diğer bir ifadeyle eğrinin eğimi değişmez ama  $y$ 'den geçtiği nokta  $b$  değeri ile yukarı/aşağı olarak değişir.

' $b$ ' değeri aslında modelimiz değişmemekle birlikte eğrimizi noktalar arasında en iyi konumda yerleştirebilmemizi sağlayan bir sabittir.

' $a$ ' değerine çarpan denir ve **eğimi belirler**. ' $a$ ' değeri artarsa eğim artar, azalırsa eğim azalır.

Özetle, lineer regresyon formülünde ayarlamaya yapabileceğimiz değerler ' $a$ ' ve ' $b$ 'dir.

' $x$ ' zaten girdimiz, ' $y$ ' ise bu değerlerin çarpım ve toplamına göre bir değerden oluşur.

$y(\text{bağımlı değişken}) = a(\text{çarpan}) \times (\text{bağımsız Değişken}) + b(\text{kesen/bias})$

### Model Oluşturma Döngüsü:

Tahmin edilecek yeni verilerin bazıları geleceğe yönelik olabilirken bazen de modeli oluşturan verilerden eksik olanlarını **tahmin (Predict)** etmeye yönelik olabilir. Geleceğe yönelik tahminlere **öngörü (Forecast)** de diyebiliriz.

Mevcut verilere dayalı bir algoritma uygulanıp, model oluşturuluyor, daha sonra bu modele dayalı tahminler yapılıyor, model test ediliyor ve gerekirse güncelleme yapılıyor. Yeni verileri geldikçe model güncelleniyor.

### Bias-Varyans Kavramları:

**Bias**, modeli optimize etmek için dışarıdan eklenen bir değer, sabit olarak düşünebiliriz.

Model optimizasyonunda temel amaç, bias değerinin en az olmasıdır.

Çünkü bias bir anlamda varsayımdır. Modelin bias değeri ne kadar yüksek olursa, çıktılarda o kadar **varsayımlara** dayanır.

**Varyans**, modelin **daha önce görmediği örneklerle** ne kadar iyi uyduğunu yansıtır.

Varyansın mümkün olduğunca az olmasını hedeflemekteyiz.

Model oluşturmada bias/varyans arasındaki ilişkide **en ideal durum düşük bias/varyans** yakalamaktır.

Hangi algoritmayla model geliştirilirse geliştirilsin bu iki durum hedeflenir.

### Çoklu Doğrusal Regresyon:

Modeli birden fazla özellik(min. 2) etkilemesi durumudur.

Lineer Regresyon formülünün genel mantığı değişmez, 2 bağımsız değişken gelmesi halinde,  $y = ax + cz + b$  haline dönüşür.

‘x’ ve ‘z’ bağımsız değişkenlerdir. Yani özellik sayısı 2’ye çıkartılmıştır. ‘z’ bağımsız değişkeninin aynen ‘x’de olduğu gibi çarpanı vardır.

‘a’, ve ‘c’ çarpanları optimize edilerek mevcut verilere dayalı bir model oluşturulacaktır.

**Dikkat edilmesi gereken nokta**, artık 3 parametremiz(x,y,z) olduğu için 3 boyutlu uzayda düşüneceğiz fakat hala 3 boyutlu uzayda bir doğru grafiğidir. ‘x’, ‘z’ üslü ifade olmadığı için doğru grafiği şeklindedir. Eğerki, ‘x’ veya ‘z’ üslü ifadeler olsaydı bir eğri(polinom) oluşurdu.

Bundan dolayı sistemde ne kadar bağımsız değişken olursa olsun, **elde edilecek model doğrusal bir modeldir.**

Sisteme dilediğimiz kadar bağımsız değişken yani özellik ekleyebiliriz. Bunun bir sınırı yoktur. Oluşacak modelde her bir özelliğin modele etkisi farklı ağırlıklarda olacaktır veya bazılarının hiç olmayacaktır. Bunu anlamamanın yöntemleri, hesaplamaları vardır.

Yukarda bahsettiğim gibi, kaç boyutlu olursa olsun sonuçta oluşacak model lineer, yani doğrusaldır. (3, 4, 5 vs. boyutlu olacağının bir önemi yok. Grafik çizilmese de matematiksel hesaplaması yapılabilir)

## Polinom Regresyon:

2. dereceden polinom:  $y = ax^2 + cx + b$

3. dereceden polinom:  $y = ax^3 + cx^2 + dx + b$

Makine Öğrenimi modellemesinde amaç, dağılım grafiğindeki bağımsız değişkenlerin **saçılımına göre en uygun eğriyi oluşturacak denklemini** bulabilmektir. Bu bizim **modelimiz** olur.

Kullanılan araçlar, programsal kütüphaneler ve bunlara yapılan hesaplamalar bu denklem veya denklemlere ait, mevcut verilere göre **çarpan ve üst değerlerini bulmaya yöneliktir.**

## Karar Ağaçları:

Karar ağaçları, makine öğreniminde diğer yaygın kullanılan algoritmalarından biridir.

Her bir düğüm için bir özellik koşuluna sahiptir ve akış koşullarla, dallanarak devam eder. Ta ki koşullar bitene kadar, nihai nokta olan uçlarda **‘yapraklar’** vardır ve bunlar da **‘sonuçlardır’**.

Karar ağaçları hem regresyon hem de sınıflandırma için kullanılan bir algoritmadır.

Sonuçların alındığı en uç noktalar(yapraklar), değer olarak bir sınıf etiketi döndürebileceği gibi, regresyonda olduğu gibi sayısal bir değer de döndürebilir.

## Birden Fazla Özelliğe Dayalı Karar Ağaçları ile Regresyon:

Birden fazla bağımsız değişkene sahip karar ağaçlarının çalışma biçimidir.

## Ortalama Karesel Hata:

Gerçek değerlerin oluşturduğumuz doğruya olan uzaklıklarını hesaplayıp buna 'hata' diyorduk, yani oluşturduğumuz model ile gerçek değerler arasındaki farkı buluyorduk. Daha sonra modelin toplam hatasının ne olduğunu anlamak için bu değerleri topluyorduk.

Toplam hatayı, oluşturduğumuz diğer çizgilere(modellere) ait hata toplamaları ile kıyaslayıp en az hataya sahip olan modeli bulmaya çalışıyorduk.

Hata toplamaları modelin doğruluğu konusunda fikir verirler.

Bu mantık bütün değer döndüren regresyon modelleri için geçerlidir.

Hataların değerlendirilmesinde en yaygın kullanılan metot Ortalama Karesel Hata (Mean Squared Error) metodudur.

Bu metodun toplam hata metodundan farkı, hataların karesinin alınması ve daha sonra bunların ortalamasının hesaplanmasıdır (veri sayısına bölümdür).

Negatif hataları burada önce karesini alarak çözümlerken, hata toplamaları modelinde mutlak değerini alarak çözümlüyoruz.

Eğrinin eğimi azaldıkça (tepe/dip noktasına yaklaştıkça, buralarda eğim 0) hata miktarı düşmektedir.

Bu, çok katmanlı derin ağlarda, ağ içindeki ileri/geri yayılım hata oranlarına dayalı ağırlıklar hesaplanırken kullanılan bir yöntemdir.

## Karar Ağacımıza Ortalama Karesel Hatanın (MSE) Uygulanması:

## Destek Vektör Makineleri (Support Vector Machines):

Denetimli Öğrenmenin, regresyon kısmında destek vektör makineleri algoritması kullanılabilmektedir.

Bu algoritma yaygın olarak sınıflandırma için kullanılır fakat regresyon için de kullanılabilmekteyiz.

Destek Vektör Makinelerinin temel mantığı, en fazla veriyi olabilecek en dar aralıkta kapsayabilmektir.

Bazı aykırı değerler olduğunda o verileri de **kapsayacak aralıkta çizgiler (karar sınırları)** çizmek çok iyi bir yaklaşım olmayabilir. Bazı aykırı veriler dışarda kalacakken, olabilecek en fazla veriyi alabilecek fakat en az aralıkta çizgiler oluşturmak, SVM regresyonu için ideal bir modeldir.