



# Makine Öğrenmesine Giriş

# Aykırı Değerler<sub>(outliers)</sub>

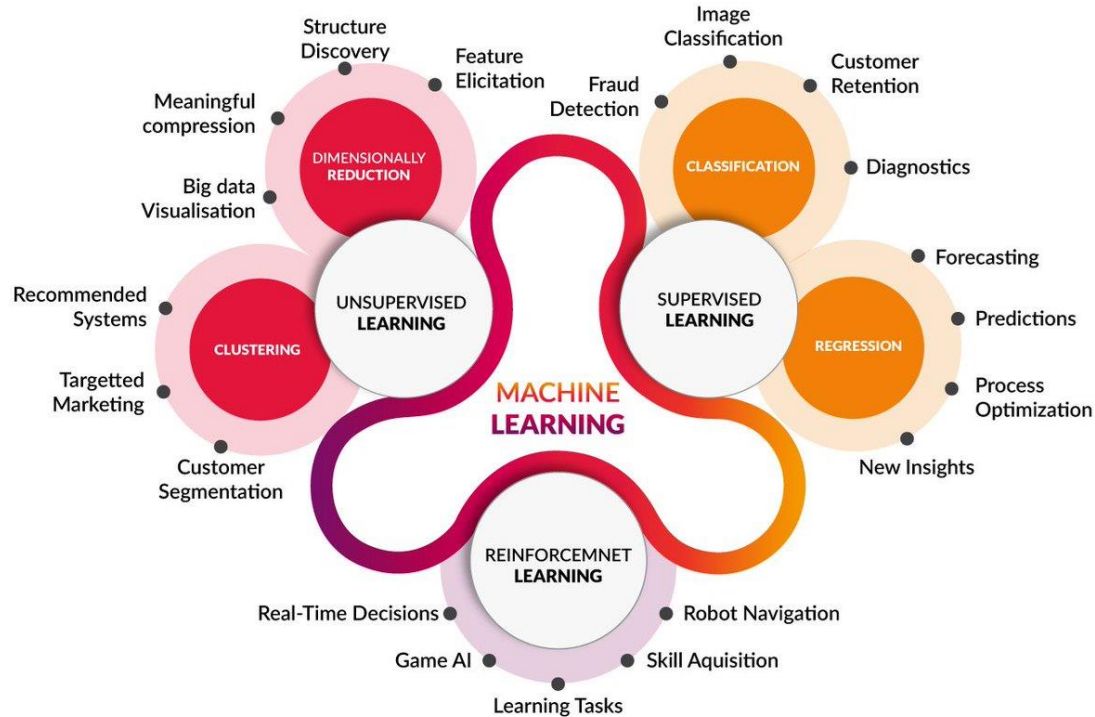
- Aykırı değer tespiti için kullanılabilecek yöntemler::

<https://archive.siam.org/meetings/sdm10/tutorial3.pdf>




- Aykırı değerler ile baş etmek için kullanılabilecek yöntemler:

- <https://hwbddocuments.env.nm.gov/Los%20Alamos%20National%20Labs/TA%2054/11587.pdf>
- <https://www.analyticsvidhya.com/blog/2021/05/detecting-and-treating-outliers-treating-the-odd-one-out/>

# Öğrenme Türleri



# Denetimli Öğrenme

- ML modelinin etiketlenmiş eğitim verileriyle eğitildiği öğrenme türüdür
- Sisteme örnek verileri nasıl kategorize edeceğini söylemeniz gerekir. Örneğin:
  - **Renk , Ağırlık , Etiket**
    - Kırmızı , 200g , Elma 
    - Turuncu , 300g , Portakal 
    - Yeşil , 150g , Elma 
- 2 girdi (renk ve ağırlık) göz önüne alındığında, sisteme her durumda (portakal veya elma) beklenen çıktı etiketinin ne olduğunu söyleriz - (denetim).
- ML sistemi daha sonra bu verileri gelecekteki görünmeyen girdileri tahmin etmek için kullanmalıdır.
- Şu anda en çok çalışılan alandır.

# Denetimli Öğrenme (Bias Varyans İlişkisi)

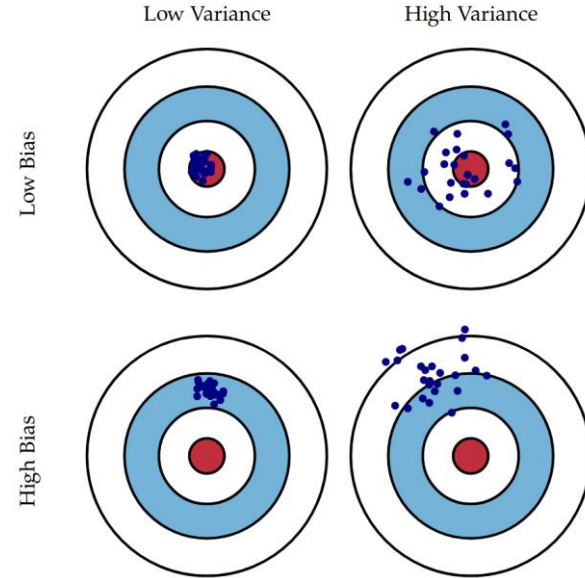
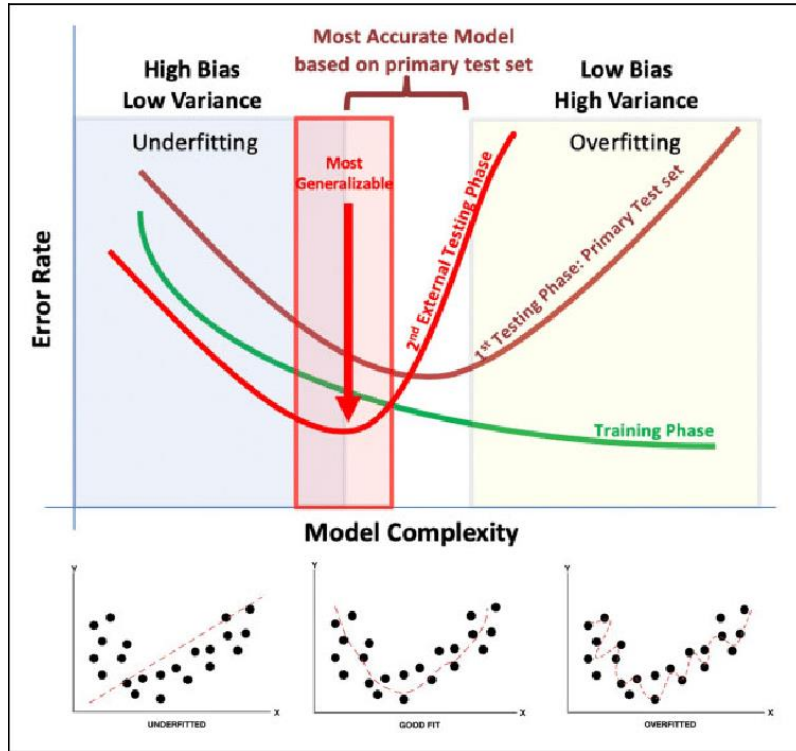
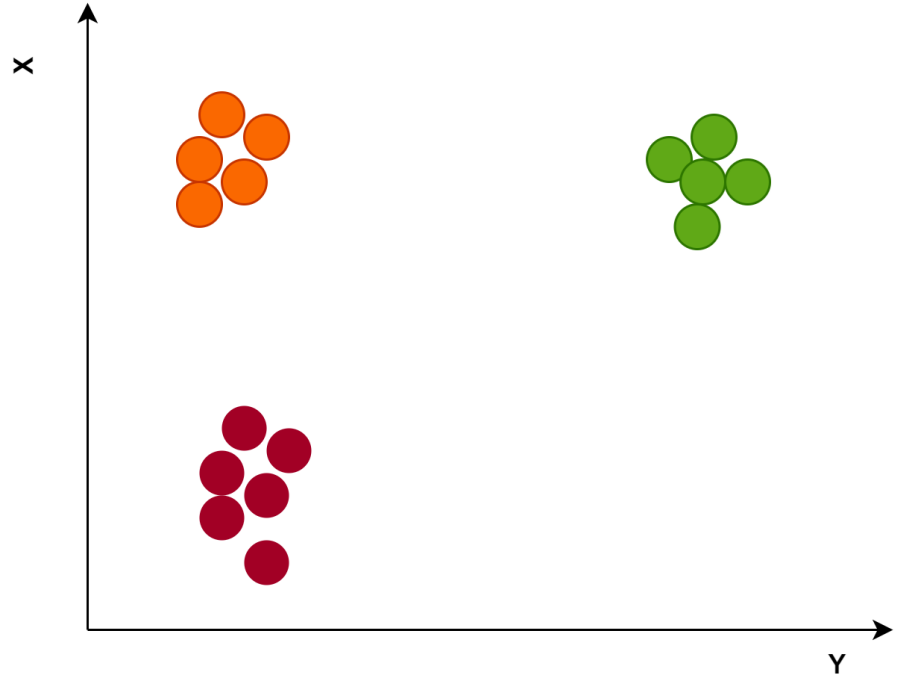
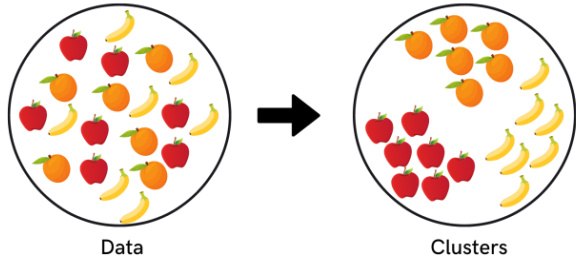
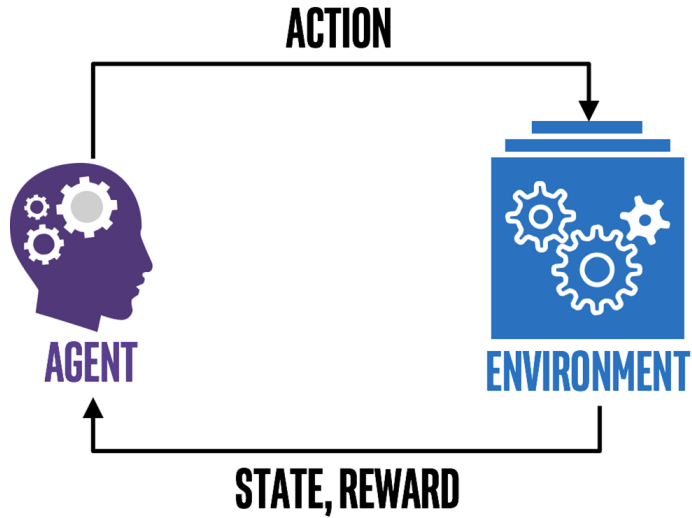


Fig. 1 Graphical illustration of bias and variance.

# Denetimsiz Öğrenme



# Pekiştirmeli Öğrenme



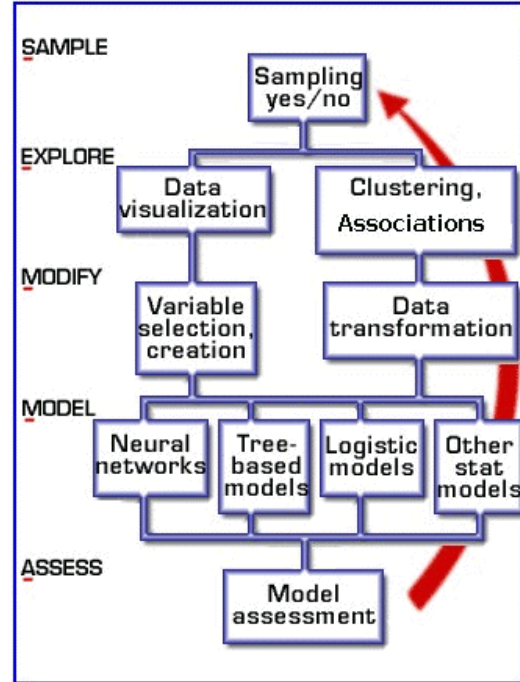
Google Deepmind Atari

# Model Oluşturma Süreci

- **sEMMA** (sample, explore, modify, model, assess) veri madenciliğinin temel sürecini ifade eder.
  1. **Örnekleme (Sample)** -> Büyük bir veri kümesinin **bir bölümünü çıkararak** verilerinizi örnekleyin. Yeni kümeniz, önemli bilgileri içerecek kadar büyük, ancak hızlı bir şekilde manipüle edilebilecek kadar küçük olmalıdır.
  2. **Keşif (Explore)** -> Anlamak ve fikir edinmek için beklenmeyen eğilimleri ve anormallikleri arayarak verilerinizi keşfedin.
  3. **Değiştirme (Modify)** -> Model seçim sürecine odaklanmak için yeni değişkenler oluşturarak, bir değişken kümesi seçerek veya var olan değişkenleri dönüştürerek verilerinizi değiştirin.
  4. **Modelleme (Model)** -> Veri madenciliğinde modelleme teknikleri, sinir ağlarını, ağaç tabanlı modelleri, lojistik modelleri, zaman serisi analizi, hafıza tabanlı akıl yürütme ve temel diğer istatistiksel modelleri içerir.
  5. **Değerlendirme (Assess)** -> Veri madenciliği sürecinden elde edilen bulguların yararlılığını ve güvenilirliğini değerlendirerek verilerinizi değerlendirin ve ne kadar iyi performans gösterdiğini tahmin edin



# Model Oluşturma Süreci



# Makine Öğrenmesi Algoritmaları

## Regresyon

Sayısal / Kategorik değerlerin tahmin edilmesi  
(ör: evin fiyatı)

## Sınıflandırma

n etiketten biri...  
(ör: kedi, köpek, insan)

## Kümeleme

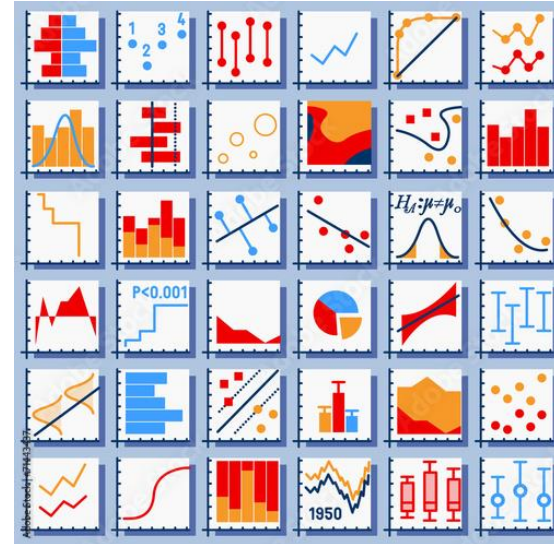
Birbirine benzer örneklerin gruplanması (ör: Amazon'daki ilgili ürünler)

## Boyut İndirgeme

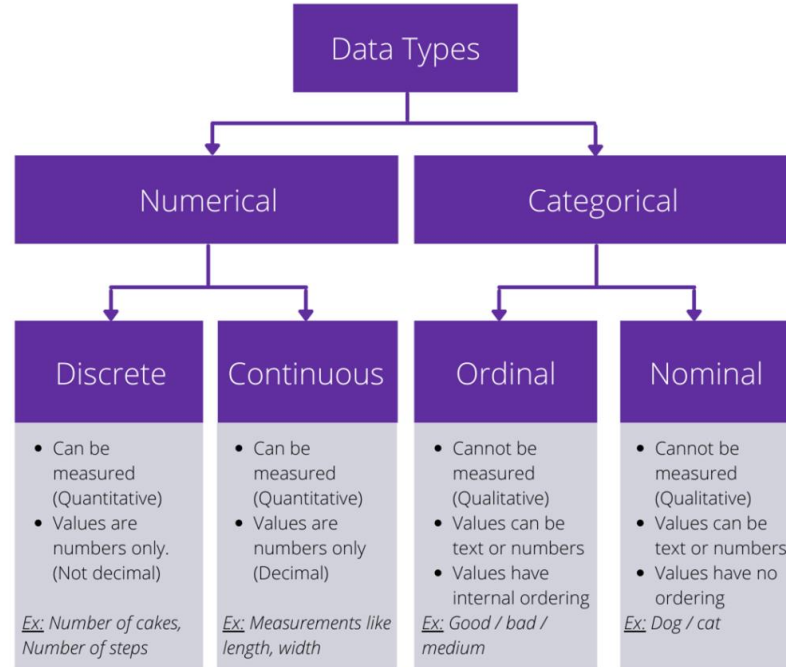
Veri boyutunun indirgenmesi  
(ör: Özellik seçimi)

# Makine Öğrenmesi Algoritmaları

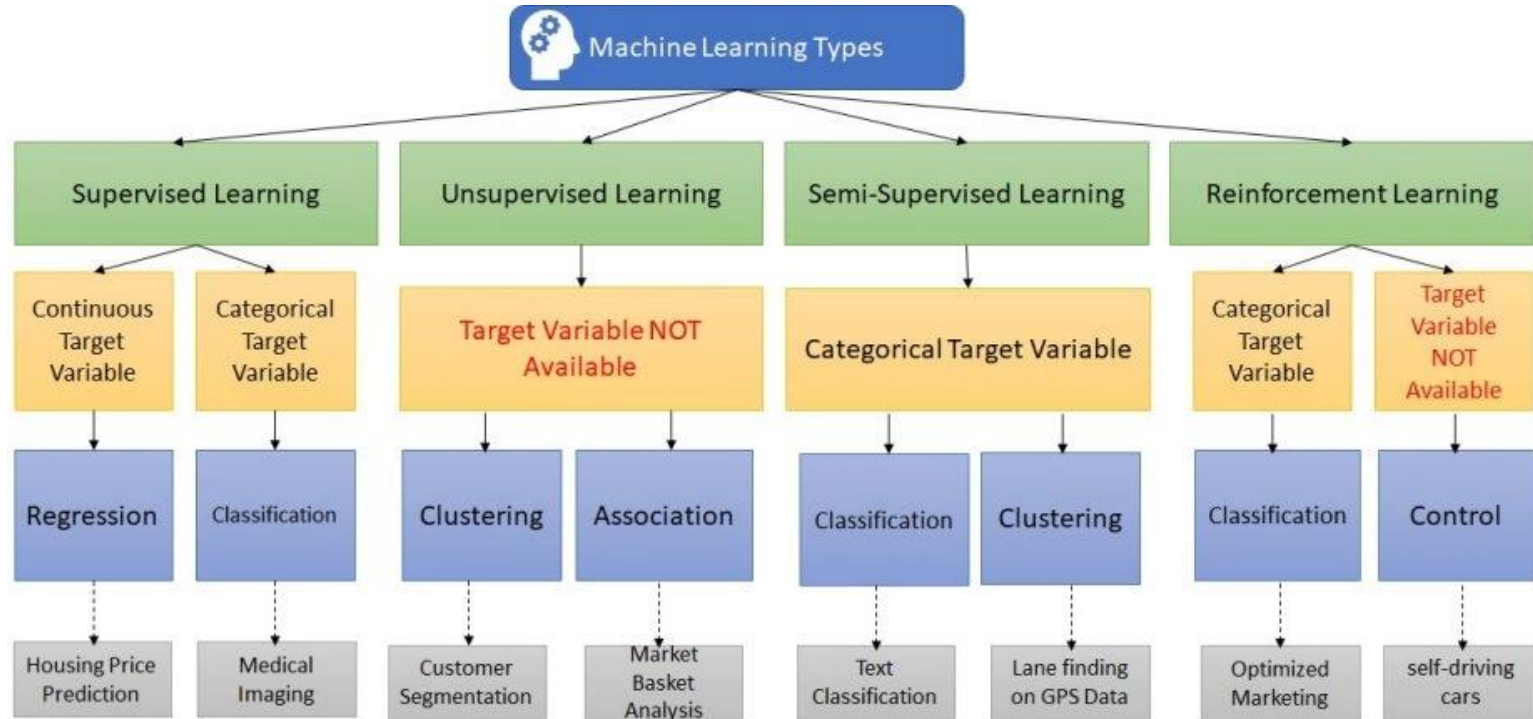
- Regresyon Algoritmaları
- Örnek Tabanlı Algoritmalar
- Karar Ağaçları
- Bayes Algoritmaları
- Kümeleme Algoritmaları
- Birliktelik Kuralları
- Yapay Sinir Ağları
- Derin Öğrenme Algoritmaları



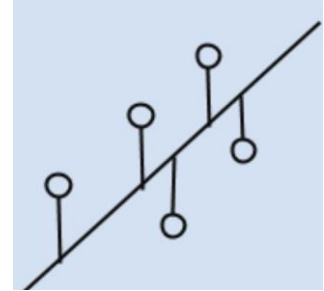
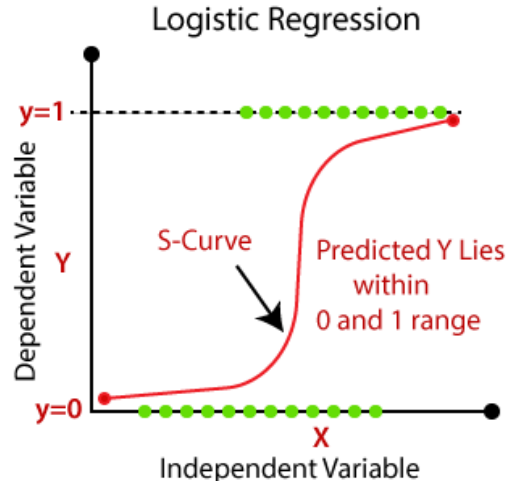
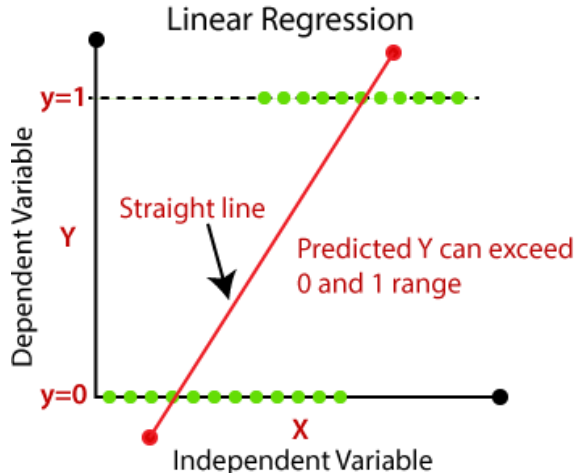
# Makine Öğrenmesi Veri Türleri



# Makine Öğrenmesi Çıktı Türleri



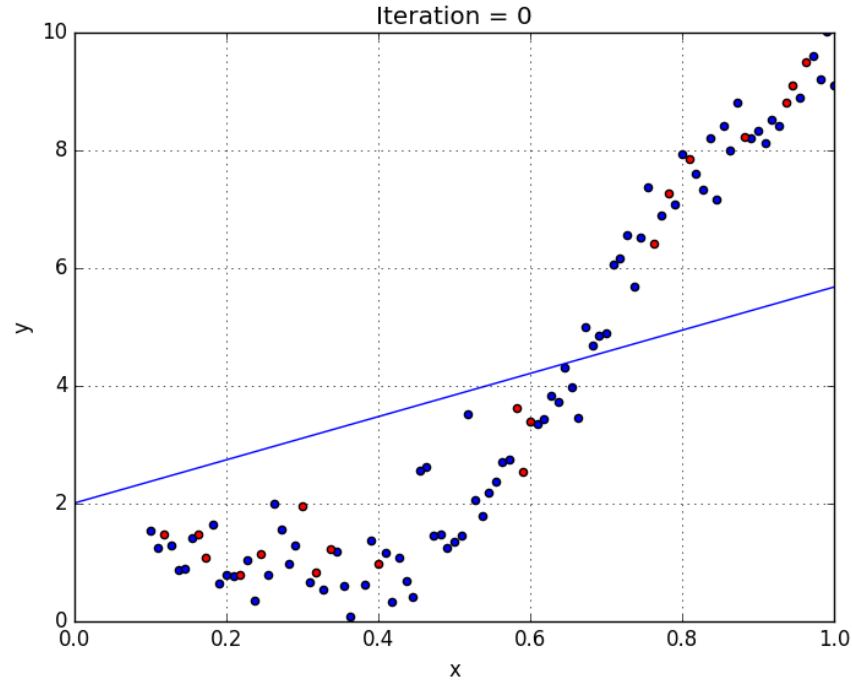
# Regresyon ile Öğrenme



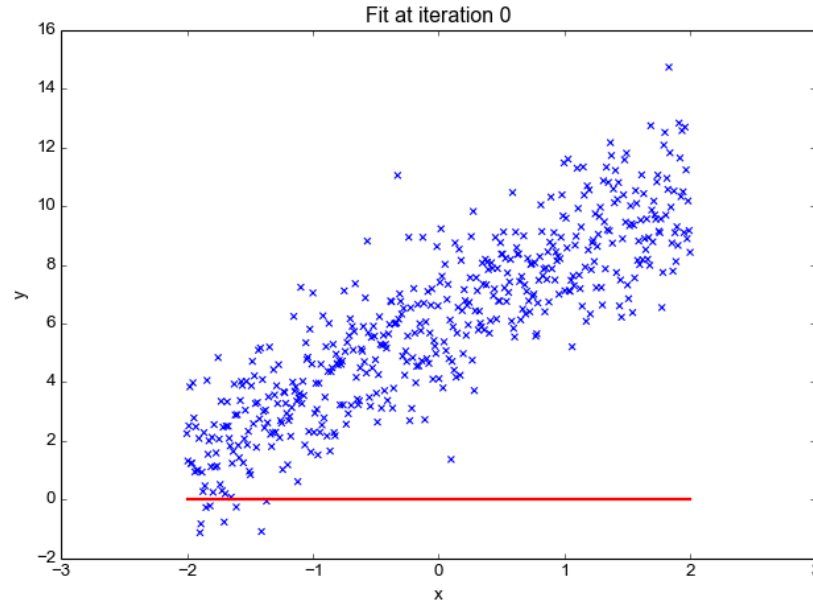
## Regresyon Algoritmaları

1. Linear Regression.
  2. Logistic Regression
  3. Ridge Regression.
  4. Lasso Regression.
- ⋮

# Regresyon ile Öğrenme



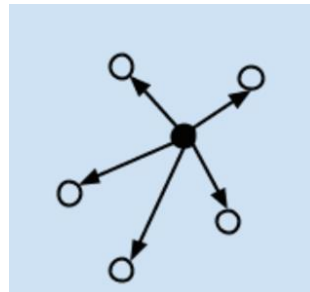
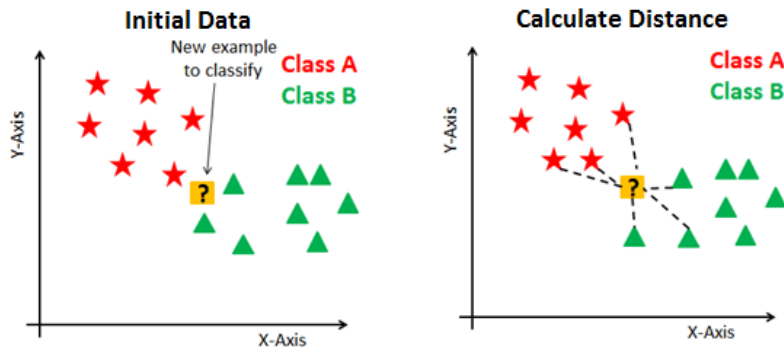
# Sürekli Çıktı (Continuous Output)



Lineer regresyon

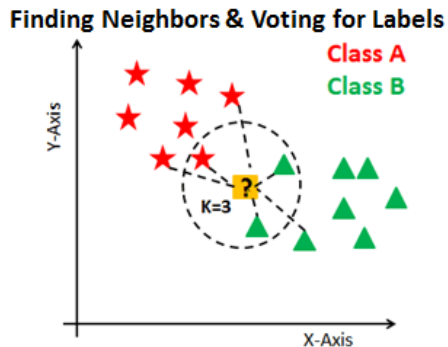


# Örnek Tabanlı Öğrenme

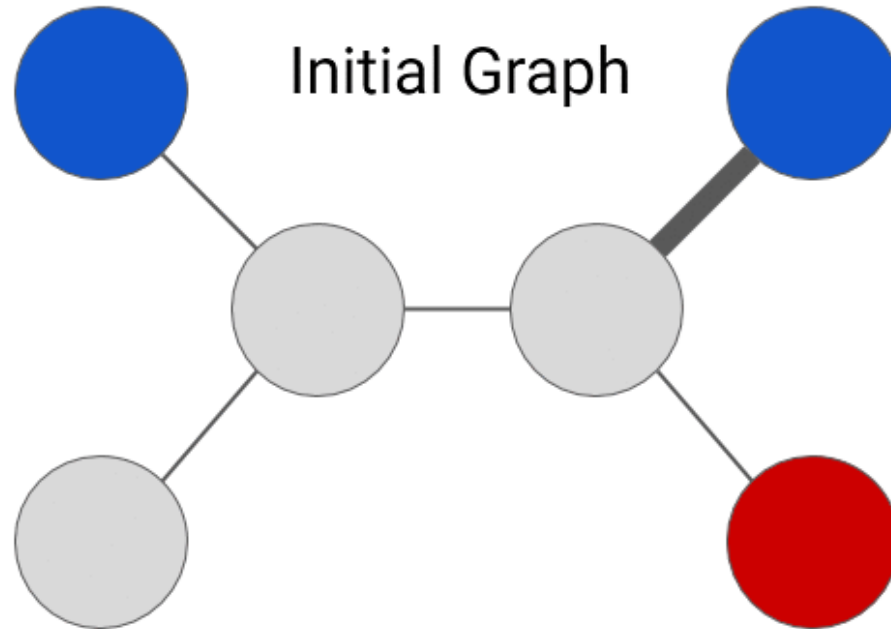


## Örnek Tabanlı Algoritmalar

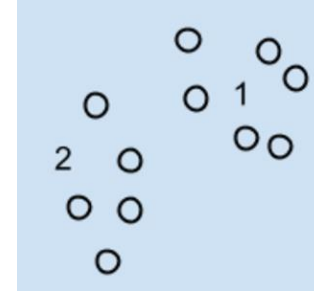
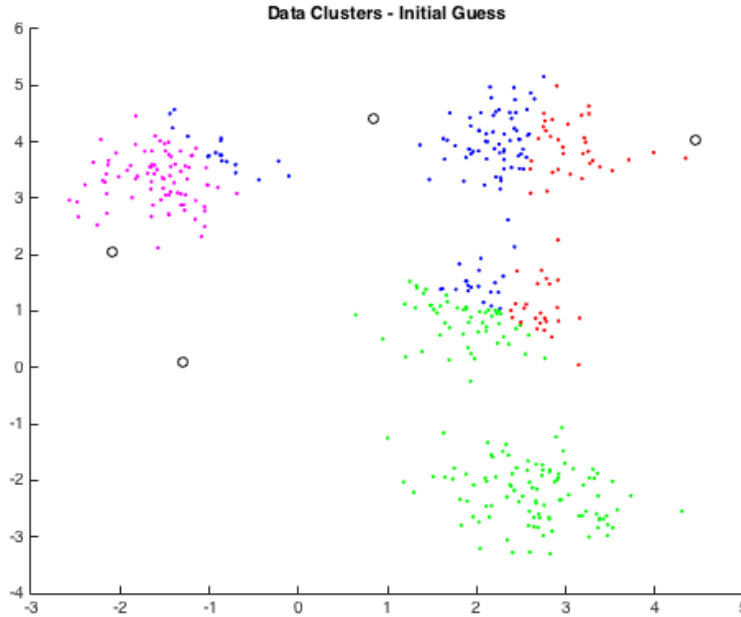
1. k-Nearest Neighbor (kNN)
2. Self-Organizing Map (SOM)
3. Support Vector Machines (SVM)
- ⋮



# Örnek Tabanlı Öğrenme



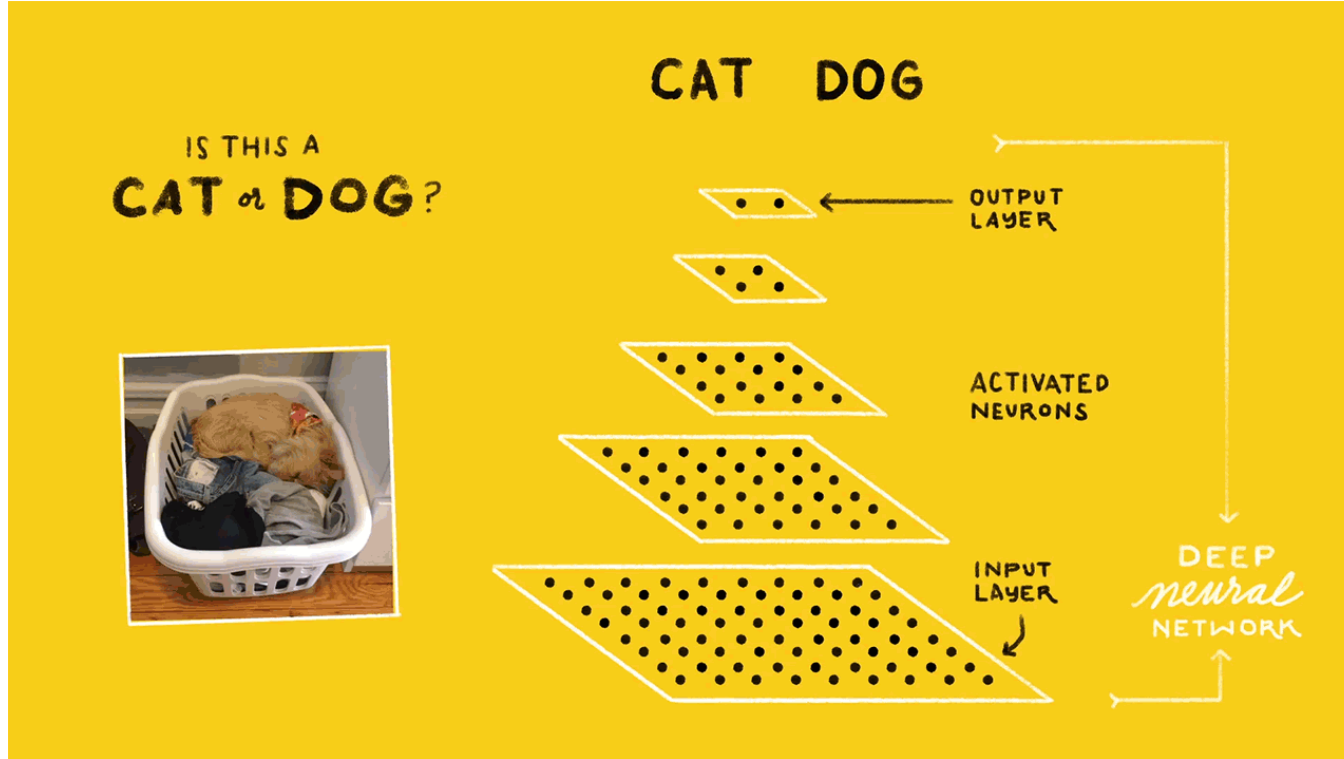
# Kümeleme



## Kümeleme Algoritmaları

1. k-Means
  2. k-Medians
  3. Expectation Maximisation (EM)
  4. Hierarchical Clustering
- ⋮

# Sınıflandırma



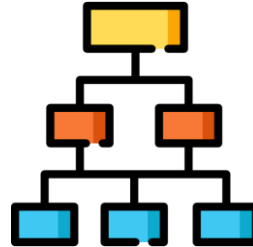
# İkili Sınıflandırma

- **İkili sınıflandırma**, sınıflandırmanın en yaygın türüdür.
- Çoğu ikili sınıflandırma probleminde, bir sınıf normal durumu, diğeri ise anormal durumu temsil eder.
- İkili sınıflandırma kullanımına ilişkin bazı örnekler:
  - E-postanın spam olup olmadığını tespit etmek için
  - Bir hastanın belirli bir hastalığı olup olmadığını belirlemek için
  - Kredi kartı dolandırıcılık işlemi algılama

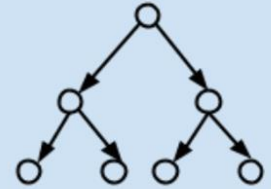
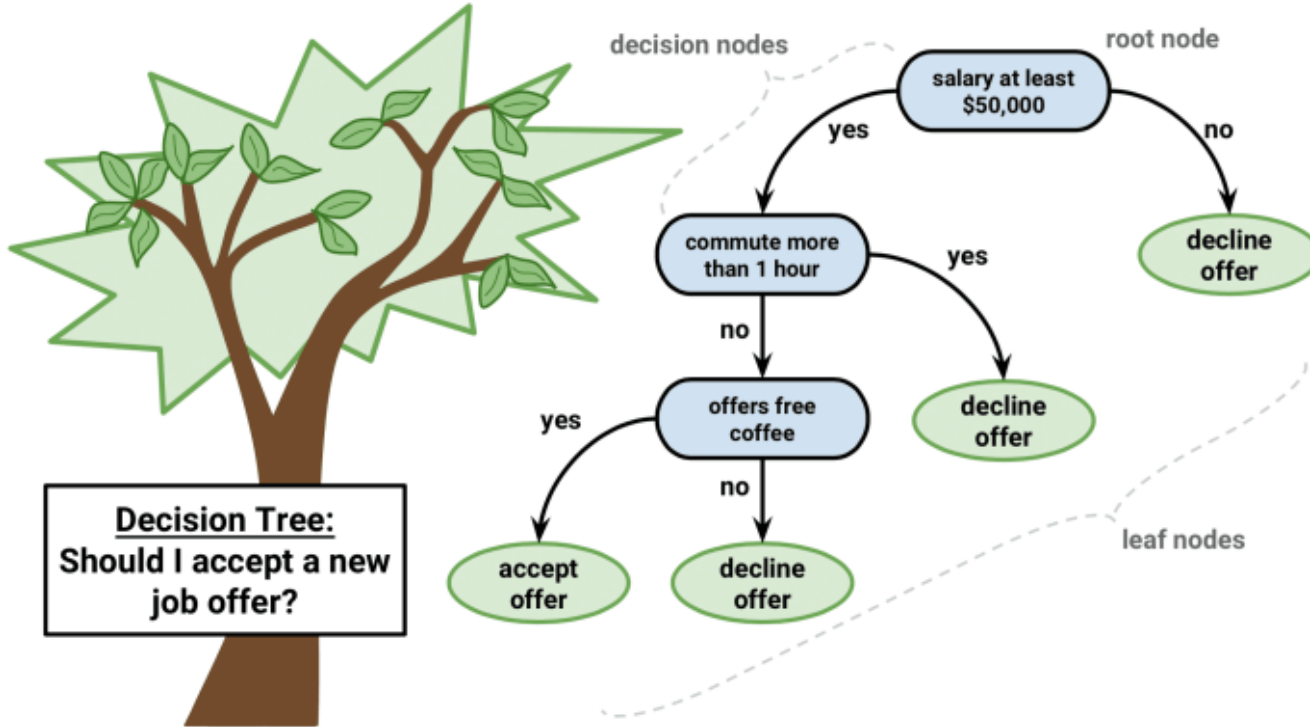


# Çok Seviyeli Sınıflandırma

- **Çok seviyeli veya nominal sınıflandırma**, ikiden fazla sınıfa sahip olması dışında ikili sınıflandırmaya çok benzer.
- Çok seviyeli sınıflandırma, **bir örnekte yalnızca bir sınıfın** kullanılmasını gerektirir.
- Çok seviyeli sınıflandırma kullanımına ilişkin bazı örnekler:
  - Haberlerin farklı kategorilerde sınıflandırılması,
  - Kitapları konularına göre sınıflandırmak
  - Fotoğraflardaki hayvan türlerinin sınıflandırılması



# Karar Ağaçları



## Karar Ağacı Algoritmaları

1. CART
2. ID3
3. C4.5
- ⋮

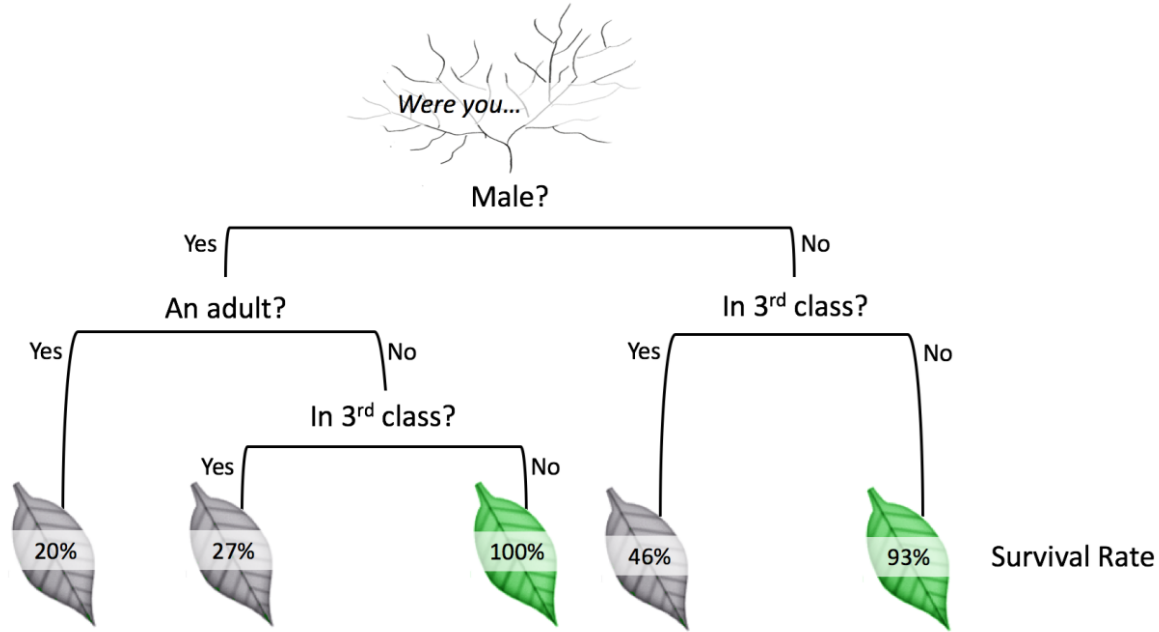
# Olasılık Tahmini

- Bu, ML sisteminin çıktısının, 0 ile 1 arasında bir ondalık sayı olduğu anlamına gelir; bu, verilen girdinin istenen bir çıktı olduğunu düşündüğümüz olasılığı belirtir ( $0,751 == \%75,1$ )
- Örnek: aşağıdaki top havuzunda kaç tane sarı top olduğunu tahmin eden bir ML sistemi.  $\%54,3$  olasılıkla “90” çıkarsa, bu bilgi ve verilen doğruluk oranına dayanarak tahminde bulunabiliriz.





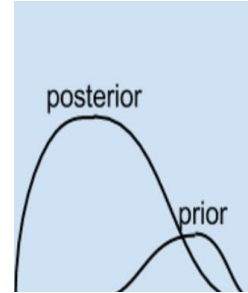
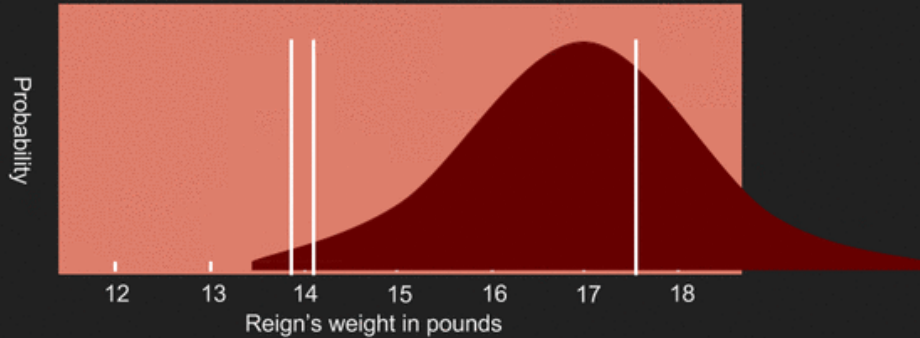
# Karar Ağaçları



# Bayes

$$P(w \mid m) = P(m \mid w)$$

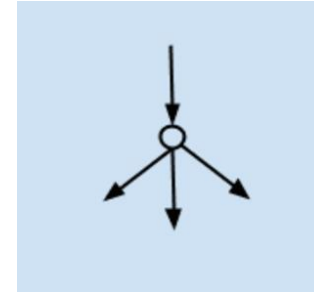
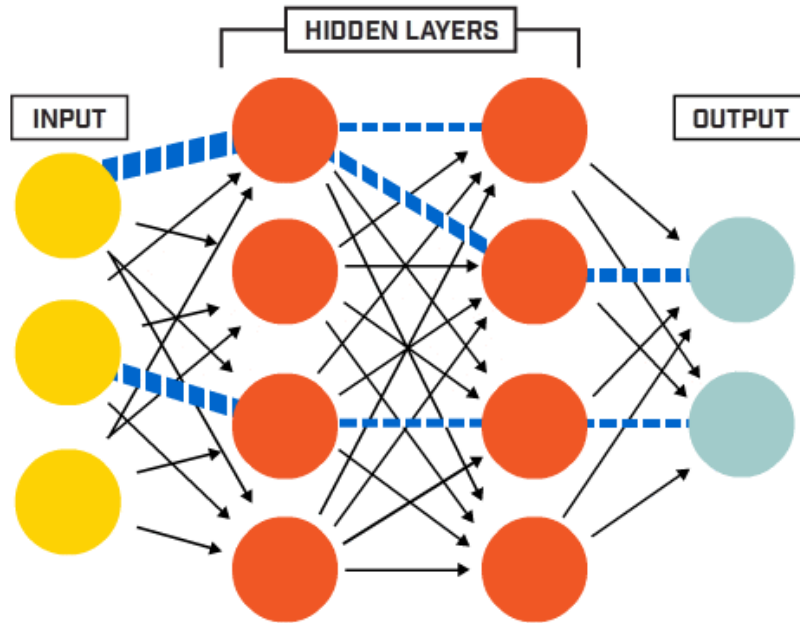
$$P(w = 17 \mid m = [13.9, 14.1, 17.5]) = P(m = [13.9, 14.1, 17.5] \mid w = 17)$$



## Bayes Algoritmaları

1. Naive Bayes
2. Gaussian Naive Bayes
- ⋮

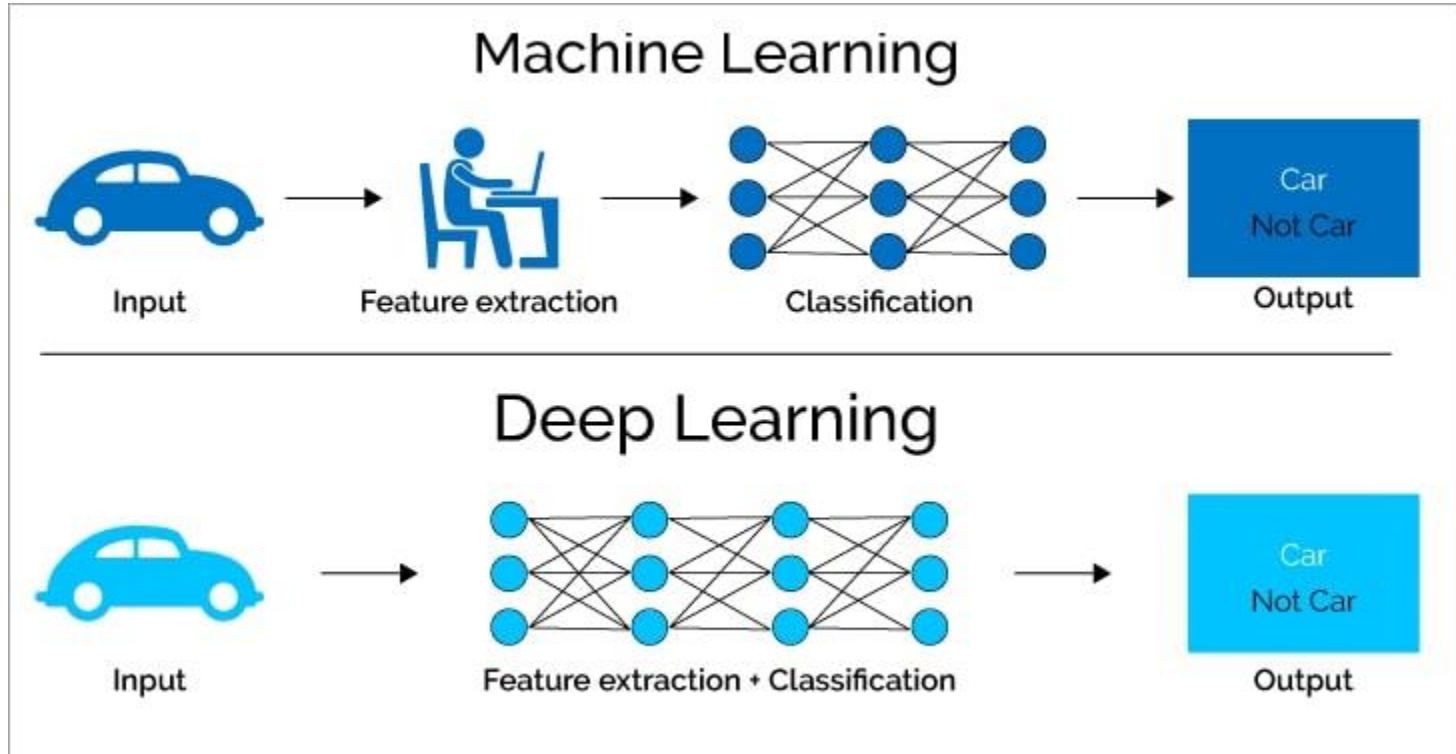
# Yapay Sinir Ağları



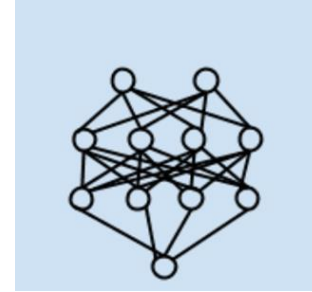
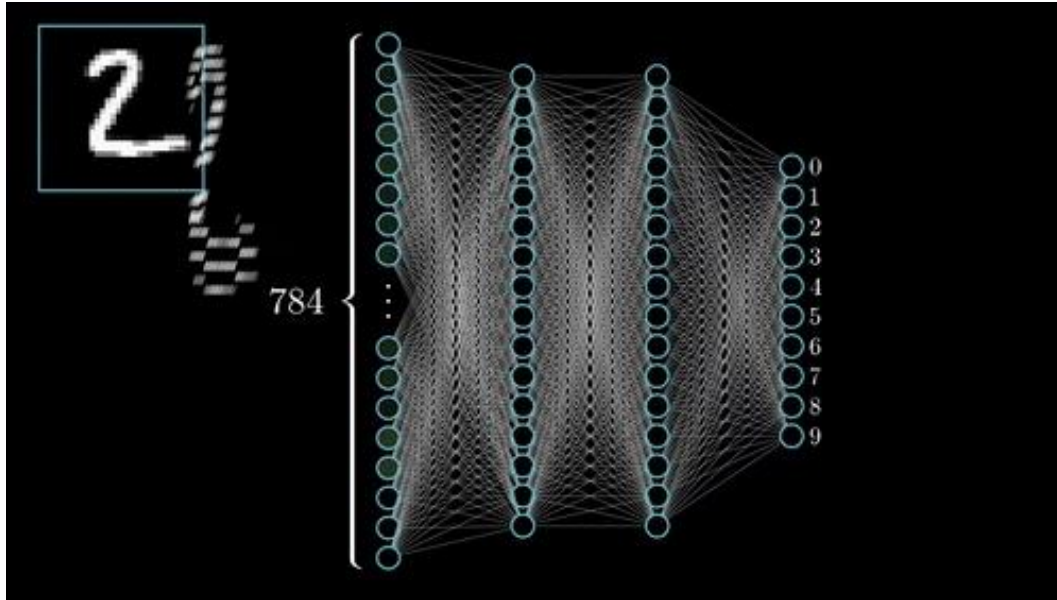
## YPA Algoritmaları

1. Perceptron
  2. Multilayer Perceptrons (MLP)
- ⋮

# Derin Öğrenme



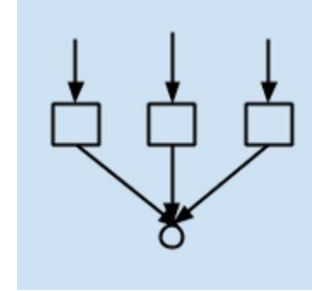
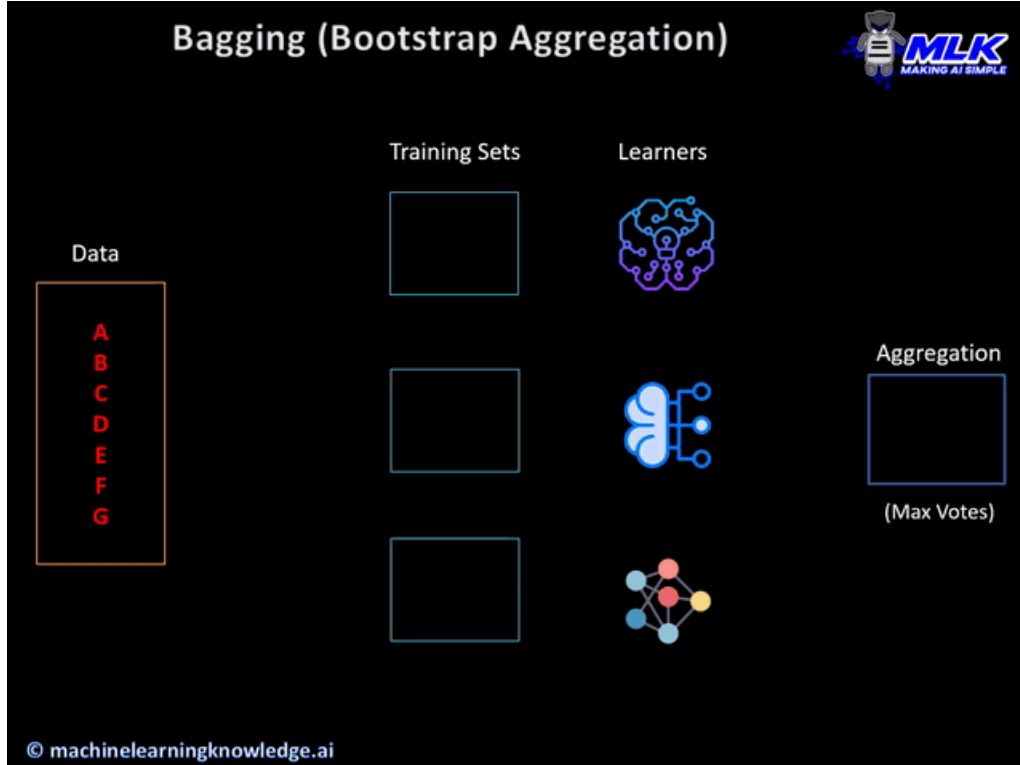
# Derin Ağlar



## Derin Öğrenme Algoritmaları

1. Convolutional Neural Network (CNN)
2. Recurrent Neural Networks (RNNs)
- ⋮

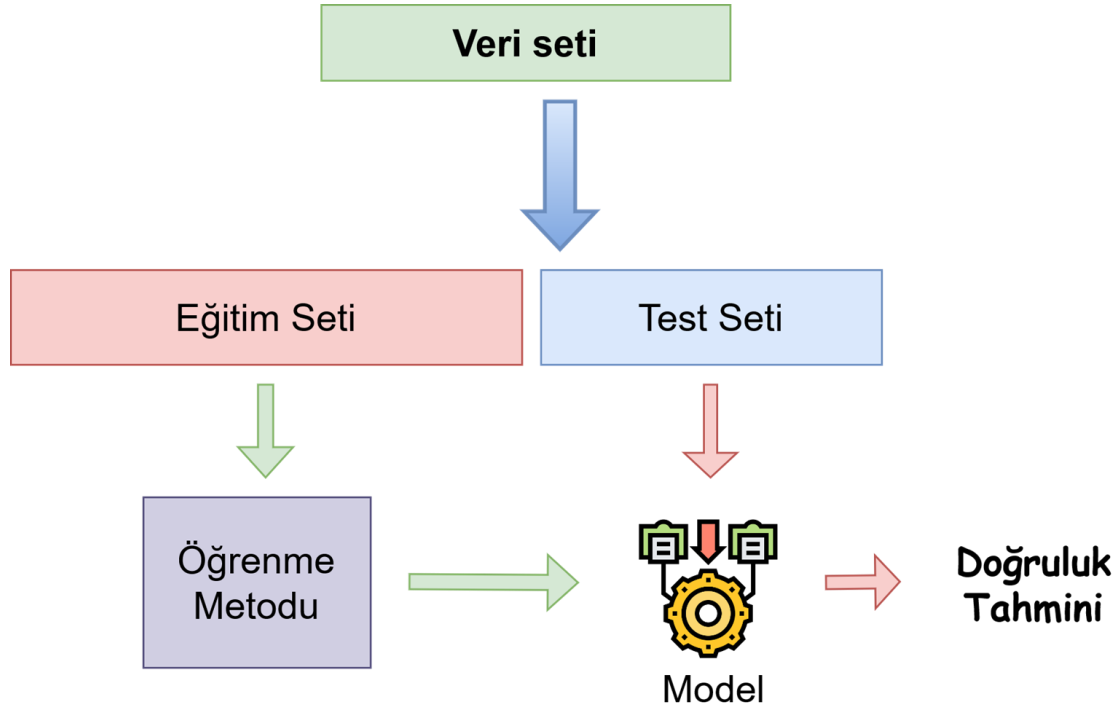
# Topluluk Yöntemleri



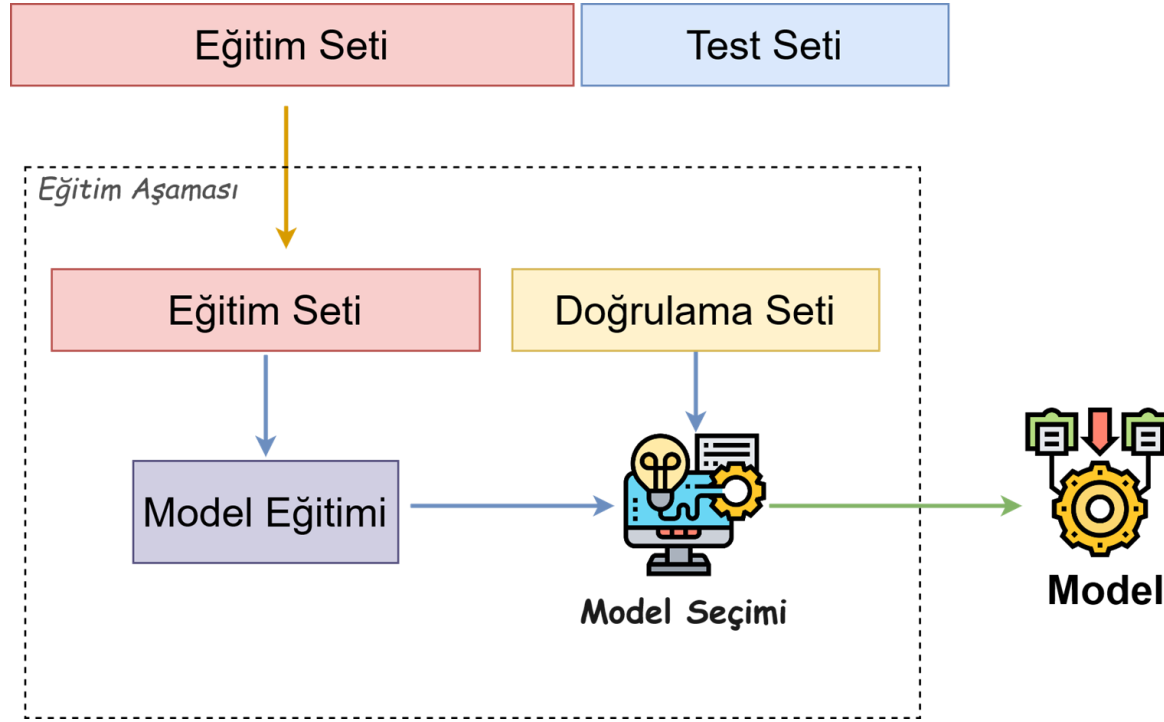
## Topluluk Yöntemleri

1. AdaBoost
2. Gradient Boosting Machines (GBM)
- ⋮

# Eğitim ve Test



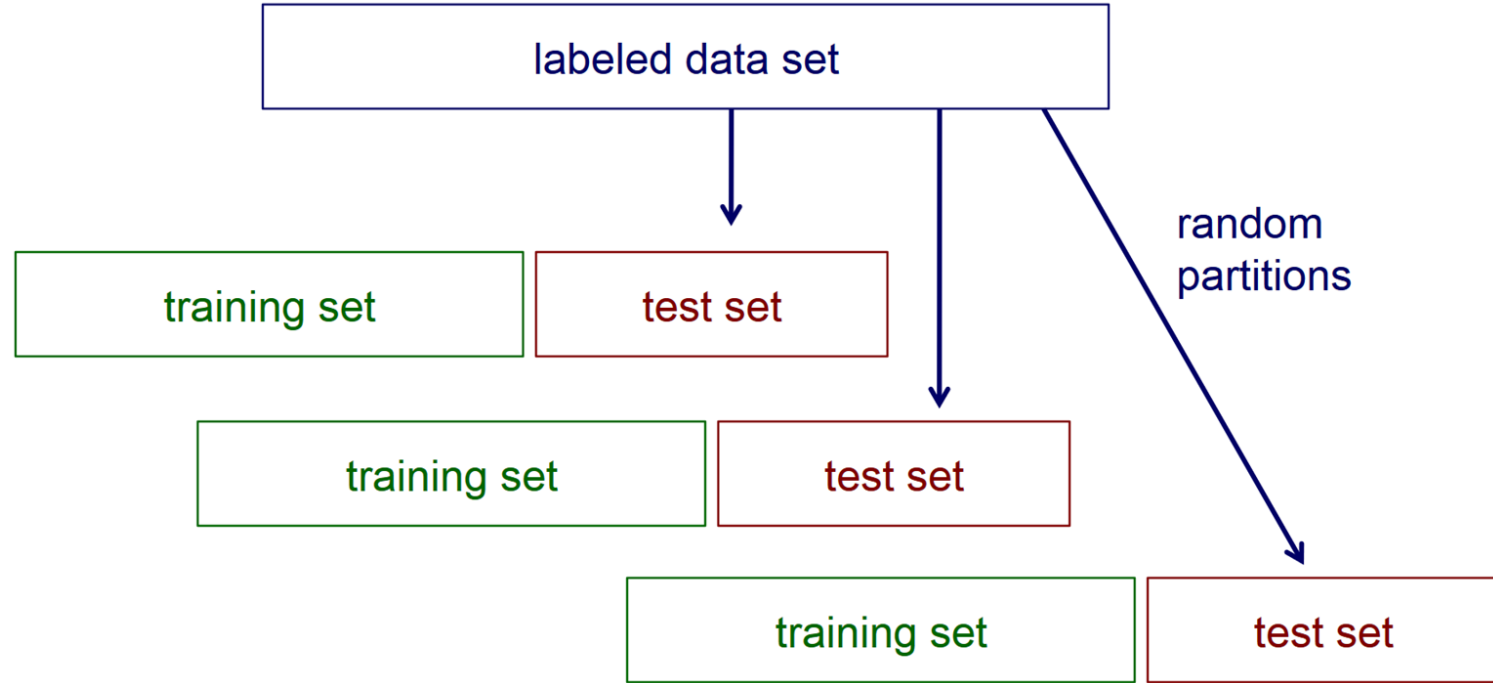
# Doğrulama Seti





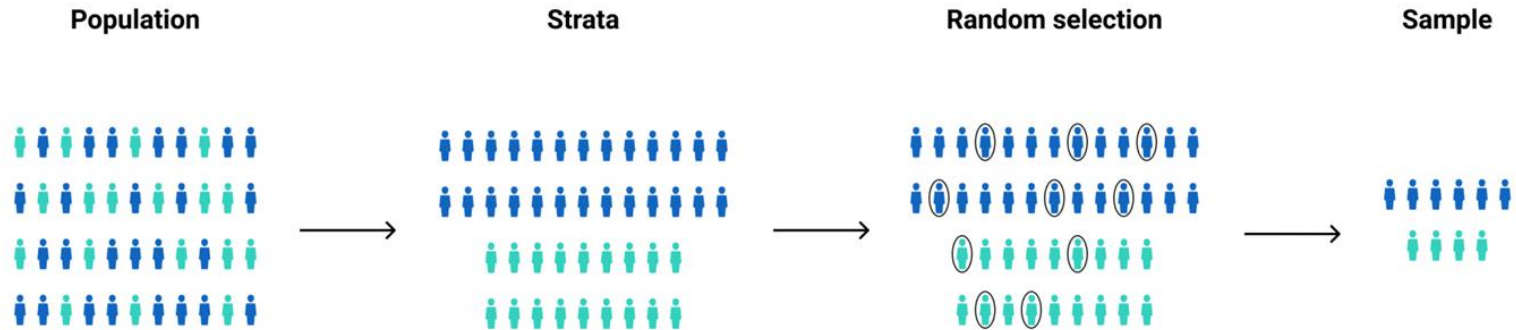
# Rastgele Yeniden Örnekleme

Random Resampling



# Tabakalı Örnekleme

Stratified sampling



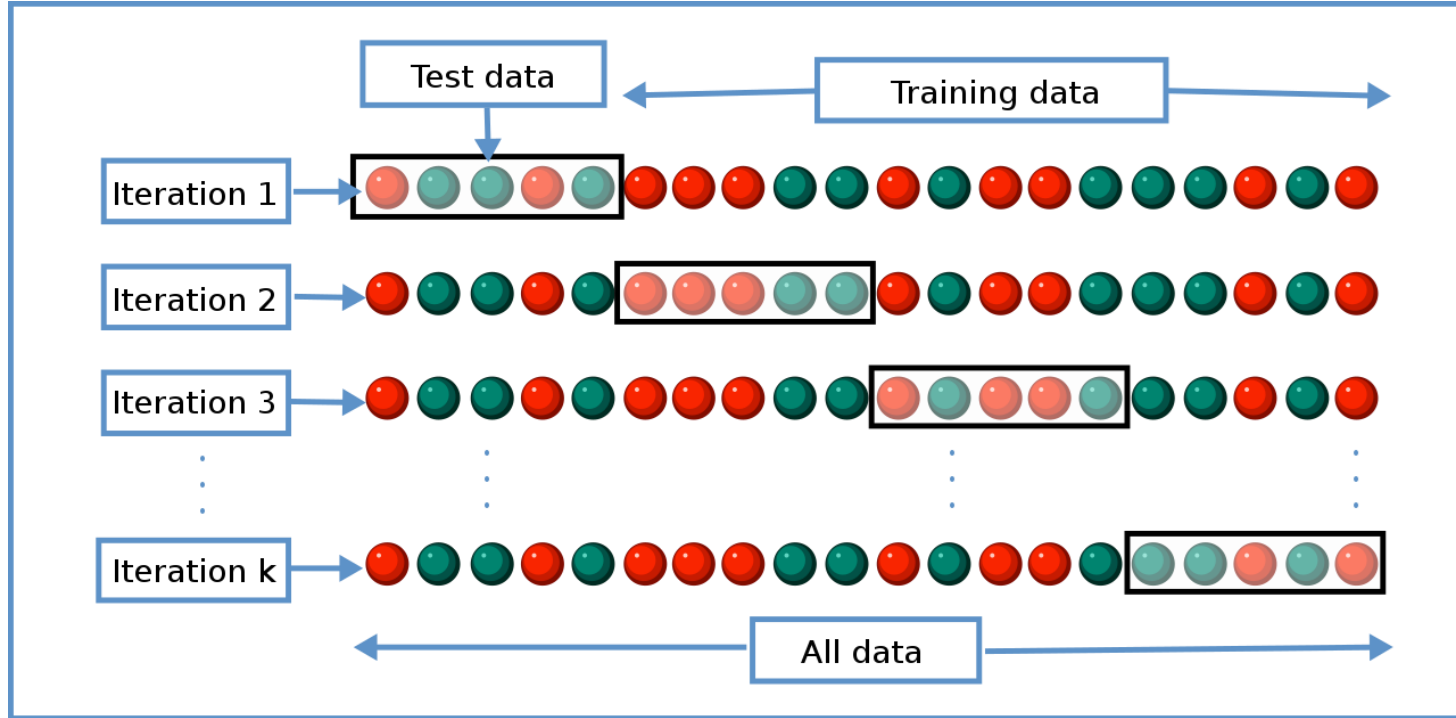
Stratified sampling

# Çapraz Doğrulama

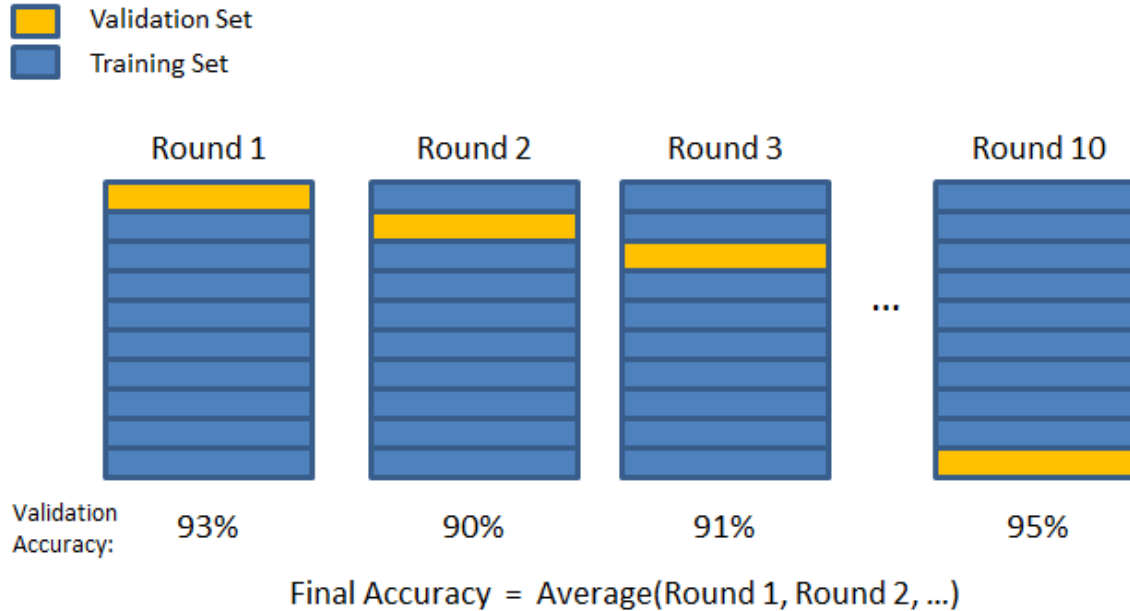
- Eğitim ve test setleri oluşturmak için **yeterli veriye** sahip olmayabiliriz.
- **Daha büyük bir test seti** bize **daha güvenilir** bir doğruluk tahmini verir. (yani daha düşük bir varyans)
- **Daha büyük bir eğitim seti** ise öğrenme süreci için veri çeşitliliğini arttıracığından **yanılgıyı (bias) düşürür**.
- Tek bir eğitim seti, **belirli bir eğitim örneğine yönelik doğruluğun** ne kadar hassas (**sensitivity**) olduğunu bize göstermez.
- Çapraz doğrulama kullanarak verilerimizi **daha verimli** kullanabiliriz.



# Çapraz Doğrulama



# Çapraz Doğrulama



# İç İçe (Nested) Çapraz Doğrulama

## Nested Cross Validation Methodology

