

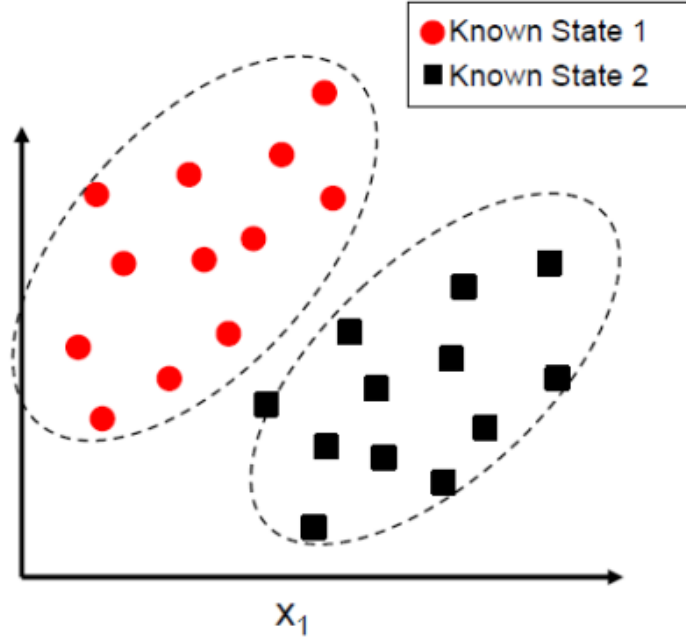
Gömülü Sistemlerde Yapay zeka ve Makina Öğrenmesi

Ali Batur

Makina Öğrenmesi

- Yapay zeka ve makina öğrenmesi çok uzun zamandır üzerinde çalışılan konulardır
- Ancak yüksek performans ve büyük veri ihtiyacı zon zamanlara kadar yayılmasını engelliyordu.
- Son dönemde teknolojide oluşan gelişmeler , üretilen chiplerin performanslarının artıp fiyatlarının düşmesi , performansın ulaşılır hale gelmesi , bu konunun yayılmasına yardımcı olmaya başladı
- Su anda herkesin bir ucundan tutmaya başlaması gereken bir konu haline geldi
- Eskiden yapılamayan uygulamalar yapılabilir hale geldi
- Birazdan göreceğiniz gibi aslında temeli matematiğe dayanıyor
- Matematiği farklı bir yorumla kullanıyorsunuz

Kullanım alanları

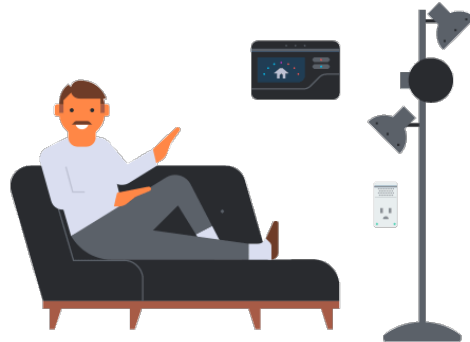


Supervised → you can fully classify

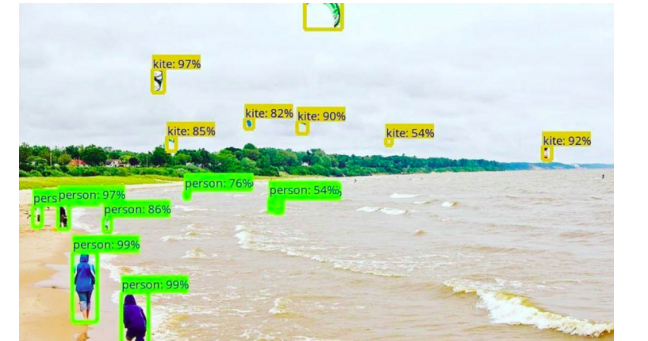
- Kullanım alanları çok çeşitli ve gelişmeye açık. Tamamıyla sizin yaratıcılığınıza kalmış
- Örneğin , çamaşır makinanızın veya değerli bir üretim makinanızın ömrünü , ne zaman tamir ihtiyacı olduğunu , zarara uğratmaması için belli parçaların ne zaman değiştirilmesi gerektiğini önceden tahmin etmek istiyorsanız , parçaya bir ivmeölçer eklersiniz.
- Bu ivme ölçer sürekli ölçüm yapar , ivme değerleri belli bir bölgedeyse sorun yok demektir.
- Bunun dışında bir bölgedeyse parçanın değişmesini önerebilirsiniz.
- Bu da sizin bir makina parçasını bekleyip üretimin durmasına sebep olabilecek arızalardan korur

Kullanım alanları

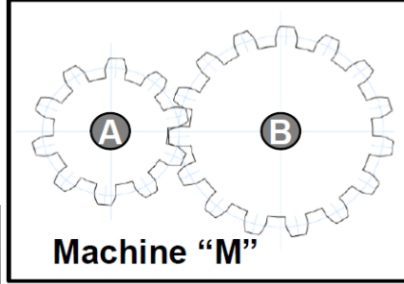
- Bir resimde istenilen objelerin farkedilmesini sağlamak gibi birçok uygulama örneği düşünebilirsiniz
- Ses tanıma , yüz tanıma , obje tanıma , yazı karakteri tanıma gibi birçok örnek verebiliriz
- Obje tanıma sayesinde araçlar , yol kenarındaki objeleri tanıyabilir , Bunların ne tür objeler olduğunu , hareket ihtimalleri olup olmadığını hesaplayıp , sizi uyarabilir
- Ses tanıma sayesinde , evinizdeki herhangi bir sistemi sesle kumanda edebilirsiniz
- Aracınızı sürerken , aracınız ses , yüz ifadenizden , tepki sürenizden, sizin uyku durumunuzu algılayabilir
- Yazı karakteri tanıma sayesinde arşivdeki kitaplarınızı bilgisayar ortamına aktarabilirsiniz veya plaka tanıma gibi özellikleri sistemlerinize ekleyebilirsiniz



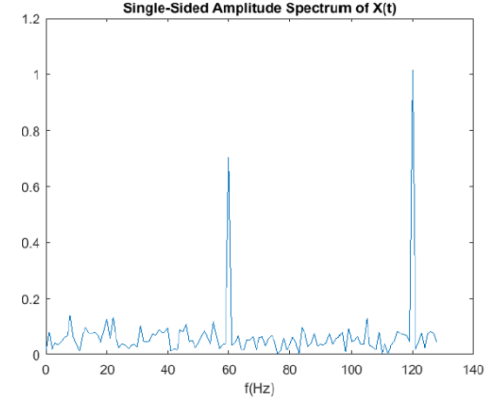
1 2 3 4 5 6 7 8 9



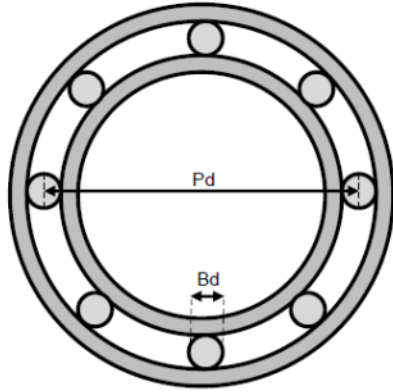
Gömülü sistem örneği



- Şöyle bir makina düşünün.
 - İçinde
 - Dönen bir motor var
 - Bir pompa var
 - Ikisini birleştiren bir bağlantı var
 - Tüm sistem , birçok problem yaşayabilir
 - Rulman problemleri
 - Yük dengesizliği
 - Şaftın yanlış hizalanması
 - Dişli hataları
 - Kayış hataları
 - Rezonans
 - Rulmanların ve dişlilerin dönme hızları ve titreme frekansları bellidir. Eğer bunun dışında bir frekansda bir gürültü duyuyorsanız bu muhtemel bir hatayı gösterir



Rulman Hataları



For ball defects:

$$\text{BSF} = \frac{1}{2} (P_d/B_d) \times S \times [1 - (B_d/P_d \times \cos \theta)^2]$$

For outer trace defects:

$$\text{BPFO} = \frac{1}{2} N_b \times S \times [1 - (B_d/P_d \times \cos \theta)]$$

For inner trace defects:

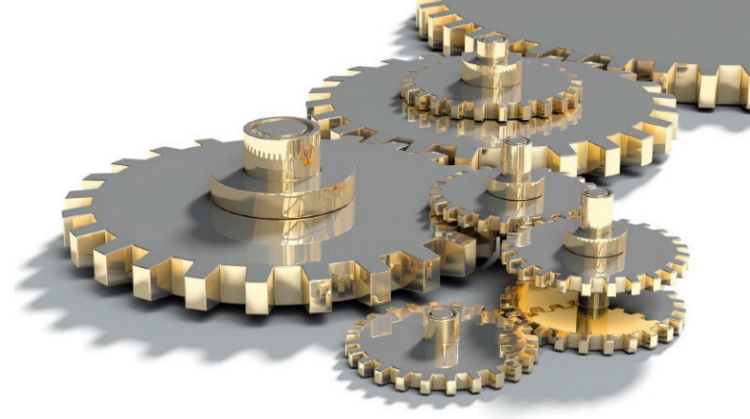
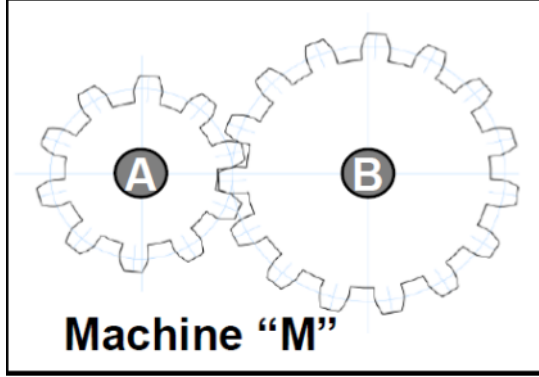
$$\text{BPFI} = \frac{1}{2} N_b \times S \times [1 + (B_d/P_d \times \cos \theta)]$$

P_d = pitch diameter
 B_d = ball diameter
 N_b = number of balls
 S = speed (revolutions/sec)
 θ = contact angle
BSF = Ball Spin Frequency
BPFO = Ball Pass Frequency of Outer Trace
BPFI = Ball Pass Frequency of Inner Trace



- Rulman hatalarının belli bir frekans izleri vardır. Bu frekanslara bakarak , problem olup olmadığı anlaşılabilir

Dişli Hataları



- A dişlisi 10 , B dişlisi 15 dişe sahip
- A dişlisi , saniyede 10 tur dönerse , 10 Hz , B dişlisi 6.67 tur yani 6.67 Hz titreşim üretecektir
- A dişlisindeki bir dişli bozuk ise bu titreşim frekansları kayacak ve size problem konusunda bilgi verecektir.

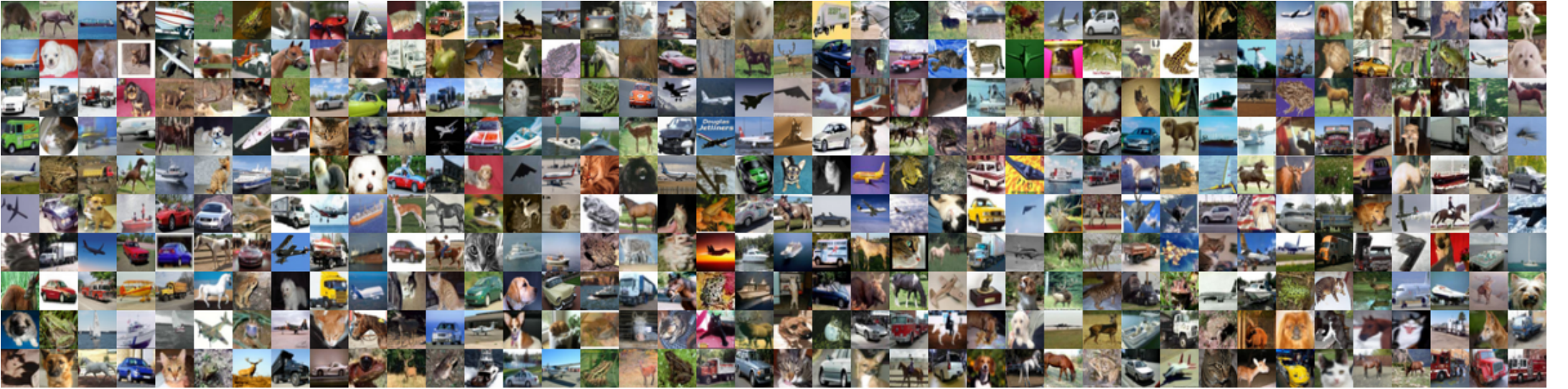
- Makina öğrenmesinde , bir takım veriler büyük veri depolarında toplanır. Depolandıkları yerler genellikle bulutdaki sunuculardır.
- Bu veriler , istatistiksel yöntemlerle oluşturulan algoritmalar yardımıyla sınıflandırılır ve modeller oluşur
- Daha önce karşılaşmadığımız bir veri seti incelendiğinde bu modellerle karşılaştırılır ve karar verilir
- Bu karşılaştırma sırasında inceleme bulutta yapılıyorsa , yani model buluttaki sunucudaysa veri buluta gider , incelenir ve sonuç buluttan uç noktaya yollanır. Burada uç noktada düşük performanslı sadece sensor verisini iletecek , internete bağlanabilir mikrodenetliyeciler kullanılır
- Bazi yöntemlerdeyse model bulutta veya PC de yaratılır ancak modelin bir kopyasında uç noktadadır. Bu sayede incelenecek veri geldiğinde buluta bağlanıp inceleme yapılması gerekmez. Uç nokta kendi kendine sınıflandırmayı , hesaplamayı yaparak sonuca ulaşır
- Sonuç her zaman doğru olacak diye bir şey yoktur. Önemli olan yüksek doğrulukta sonuç verecek modeller oluşturmaktır

- Örneğin bir resmin kuş resmi içerip içermediğini makinaya öğreteceksem
 - Önce içinde kuş resimleri olan birçok görselleri makinaya depolarım. Ne kadar fazla sayıda görsel olursa o kadar iyi olur
 - Sonra içinde kuş resmi olmayan resimleri makinaya depolarım
 - Sinir ağıları kullanılarak modeli oluştururum
 - Daha sonra test görselleriyle , makinanın ne kadar doğrulukta sonuç bulduğuna bakarım
 - Eğer yeterince doğruysa , modeli kullanırım
 - Yeterince doğru sonuç vermiyorsa , daha fazla resimle eğitirim ve modelde kullandığım yöntemi değiştiririm
 - Doğruluk miktarı istediğim düzeye ulaşınca , artık makinamızı kullanıp , sınıflandırma yapabilirim

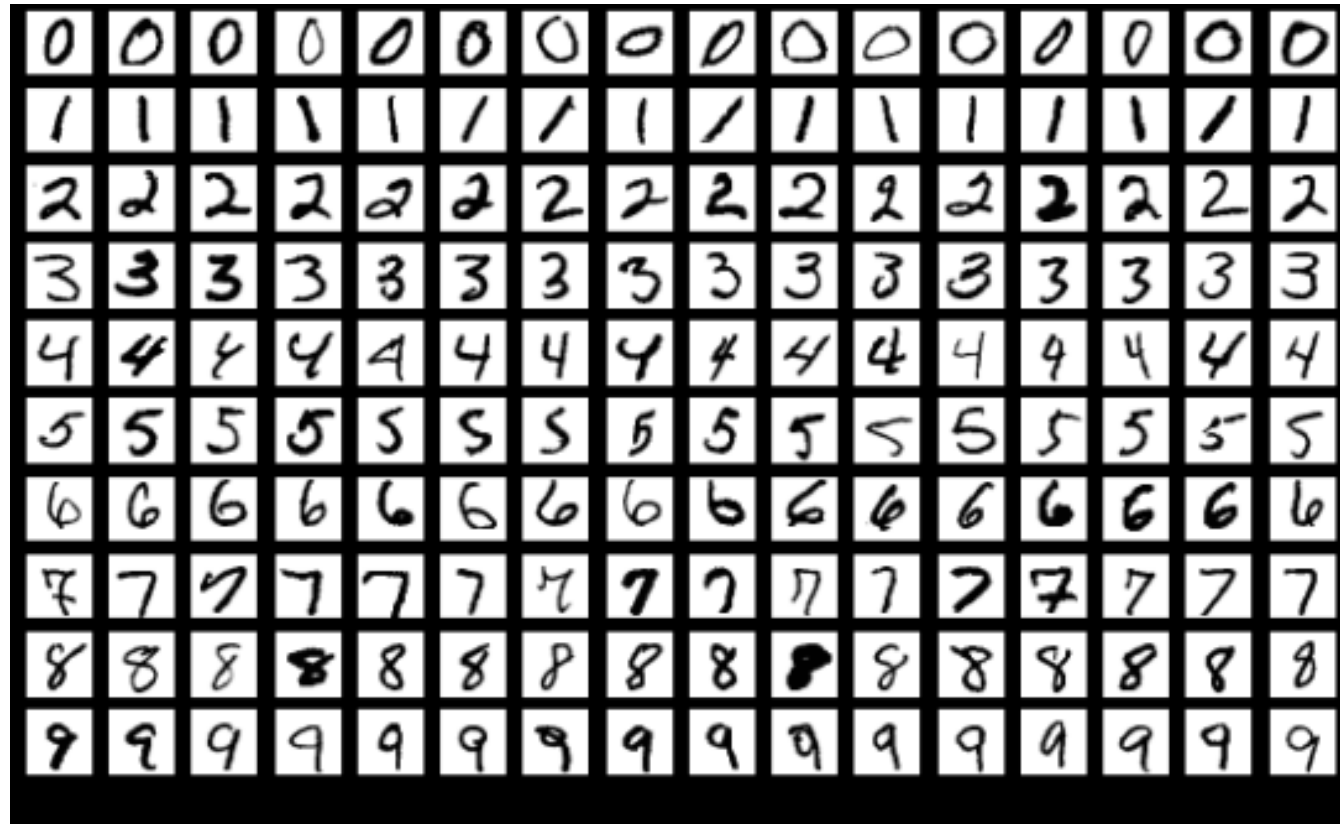
İçinde kuş resmi içeren görseller



İçinde kuş resmi içermeyen görseller



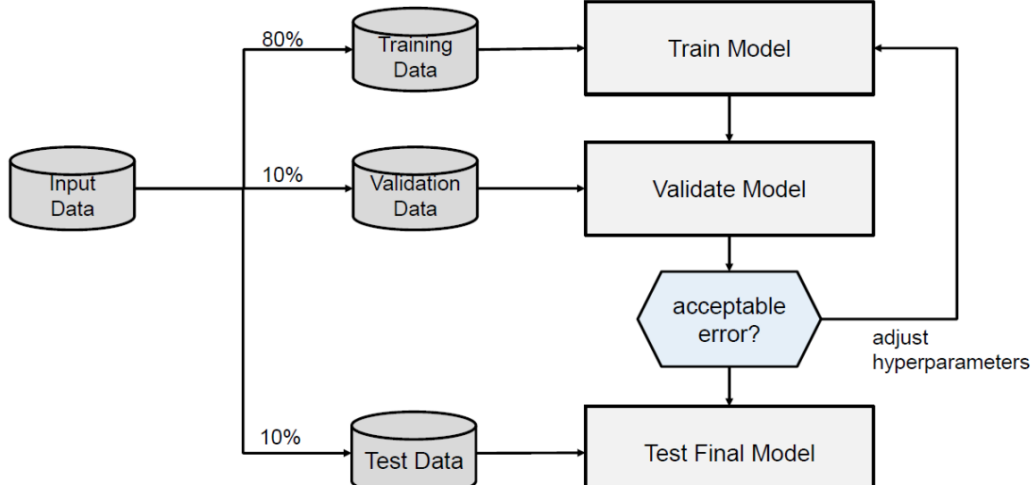
İçinde sayı içeren görseller



Tanımlar

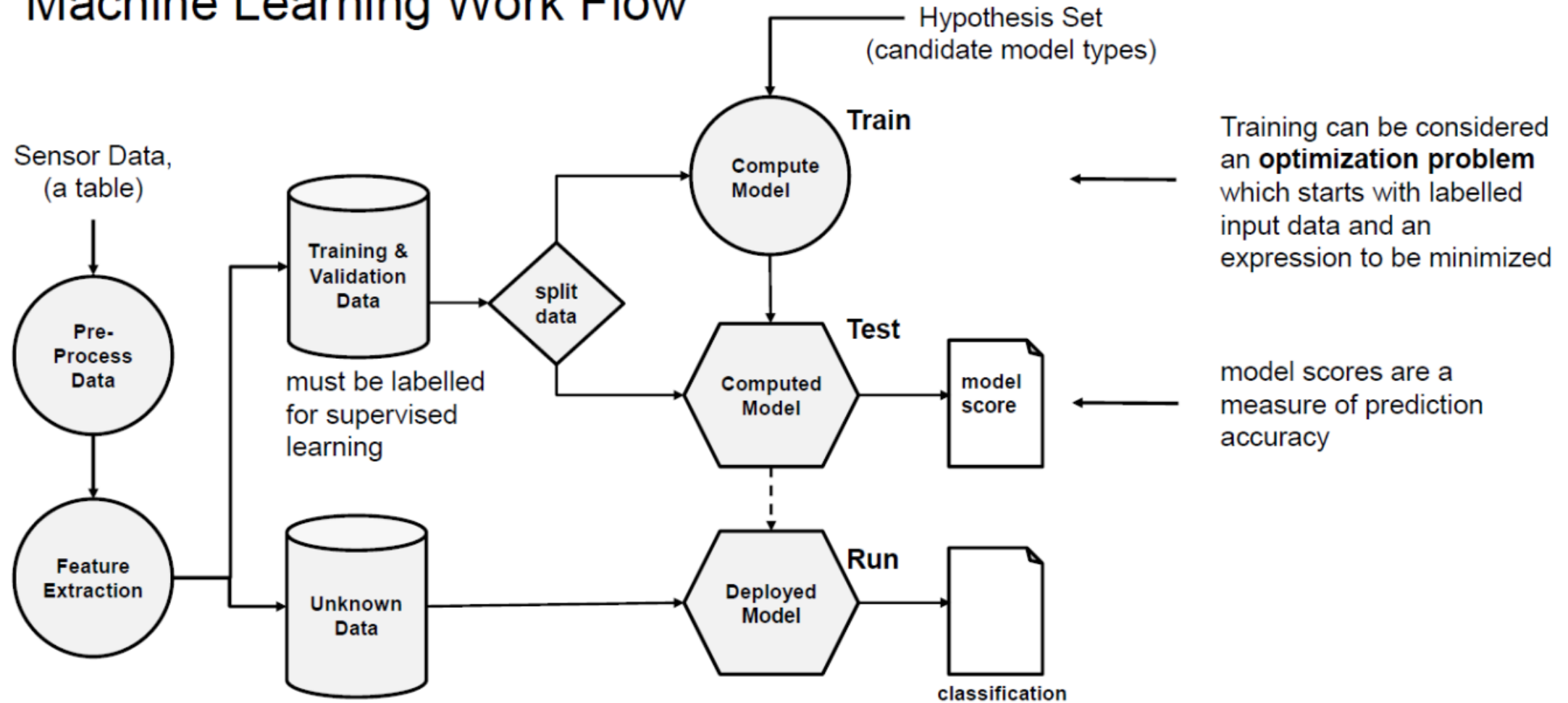
- **Observation (Gözlem) :**
 - Incelediğimiz her bir veri gözlemdir
- **Labels (Etiketler) :**
 - Bu makina öğrenmesi sırasında tahmin etmeye çalıştığımız sonuçlardır.
 - Örneğin
 - Bir sürü resim seti içinde hangisinde araba resmi olduğunu bilmek istiyorsak , araba bir etikettir.
 - E-mailer içinde hangilerinin spam olduğunu tahmin etmek istiyorsak bu bir etikettir.
 - Bir evin olası satış fiyatını tahmin etmek istiyorsak bu bir etikettir.
 - Bir çamaşır makinası motorunun ömrünü tahmin etmek istiyorsak bu bir etikettir
 - Bir motorun parçalarının bozulmaya başladığını tahmin etmek istiyorsak bu bir etikettir
- **Features (Özellikler) :**
 - Gözlemi anlatan özelliklerdir.
 - Örneğin bir emailin uzunluğu , içinde geçen bazı kelimeler , tarihi vs...
 - Bir evin semti , metrekaresi , oda sayısı vs...
 - Çamaşır makinasında , kireç ölçüm sensörü , ivmelenme sensörü vs
 - Makinanızın mümkün olduğu kadar az sayıda özellik ile öğrenmesini sağlayın. Buna Dimension reduction denir. Bu hesaplama zamanını ve depolanan veriyi azaltacaktır , ancak tahminin doğruluğunda azaltabilir. En verimli sayı ve özellikleri bulmak gerekir.

Tanimlar



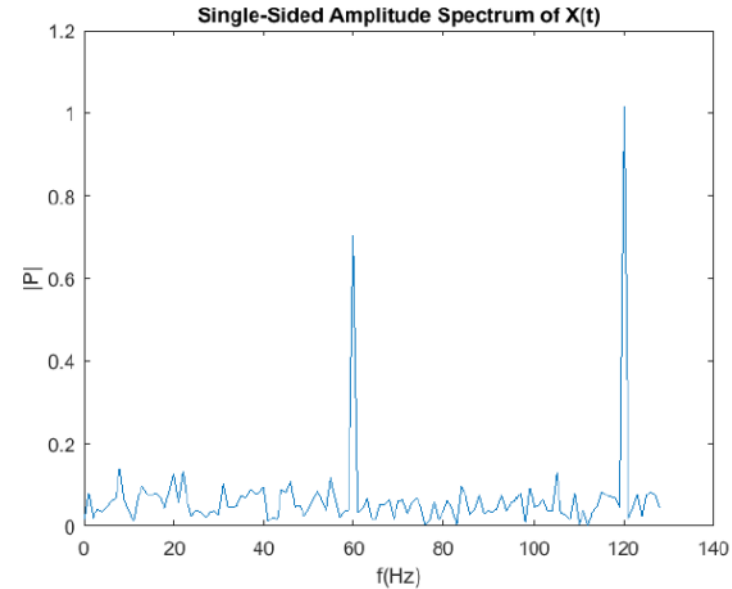
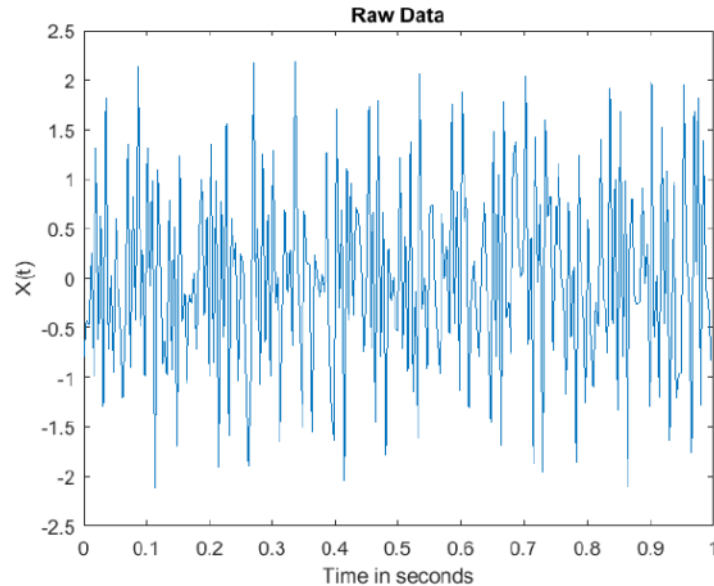
- **Model** : Model , Girdilerle(features) , çıktılar (label) arasında bağlantıyı oluşturan algoritmadır.
- **Training** (Eğitmek) : Girdisi ve çıktısı belli olan verilerle model oluşturulur veya eğitilir. Bu sayede model girdi – çıktı arasında bağlantı kurar
- **Test** (Test etmek) : Model eğitildikten sonra bir test verisiyle test edilir ve doğruluk derecesi bulunur , yeterli gelmez ise model değiştirilir

Machine Learning Work Flow



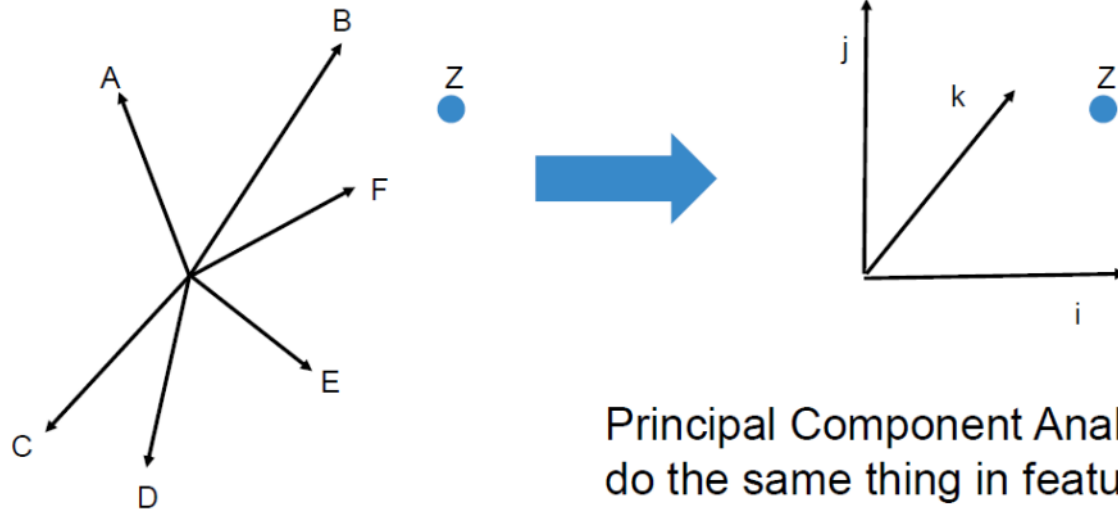
Özellikleri Bulmak (Feature Extraction)

- Makina öğrenmesi , ham veriden özelliklerin bulunması temeline dayanır
- Toplanan bir titreşim verisinden , frekans bilgisini hesaplamak bir ozellik bulma işlemidir. Gelen girdinin faydalı bir özeti cıkar
- Bir diğer özellik , değişim hızını ölçmek olabilir
- Başka bir özellik . Bir resimdeki keskin kenarları bulmak olabilir
- Konuya göre birçok farklı özellik bulunabilir



Boyut küçültülmesi (Dimension Reduction)

- Şekilde görüldüğü gibi 6 çeşit özellik bulup sonuca ulaşabilirsiniz yada biraz daha dikkatli inceleyip , 3 özellekle sonuca ulaşabilirsiniz
- Sayıyı azaltınca hesaplama süresi azalacaktır ama doğruluk yüzdesi azalabilir



Principal Component Analysis can be used to do the same thing in feature space

Confusion Matrix

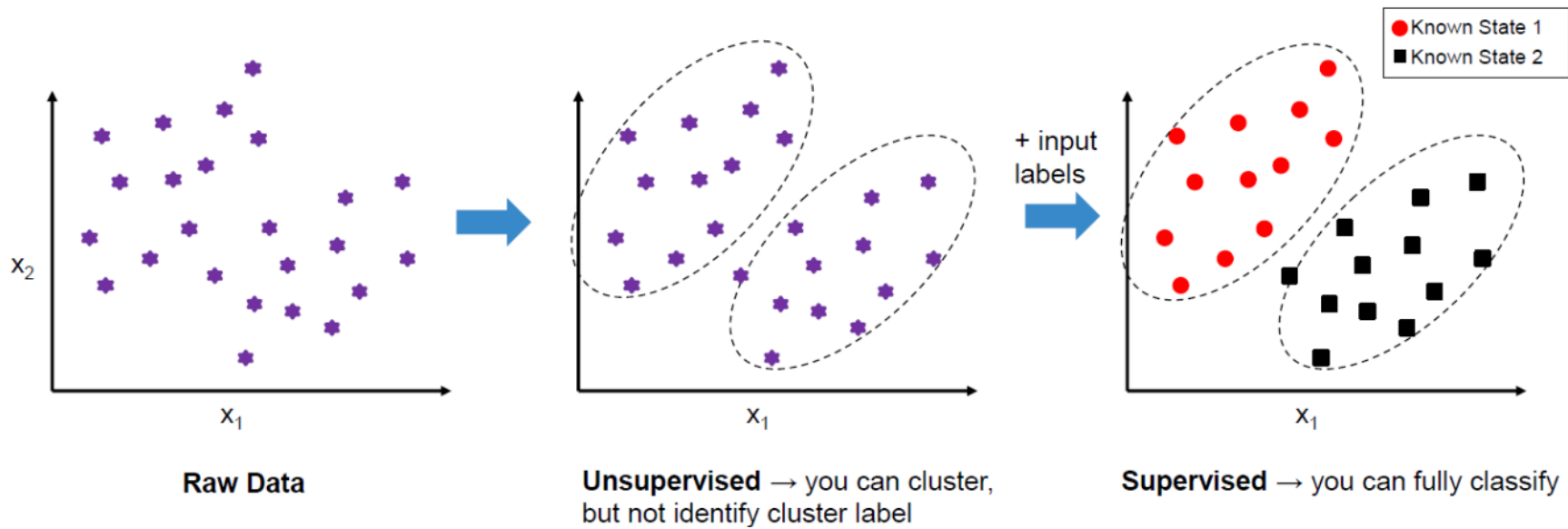
		Predicted Class			
		Class A	Class B	Class C	Class D
Actual Class	Class A	45	5		
	Class B	1	44	5	
	Class C		6	40	4
	Class D	8		3	39

- Doğru bir sistemde , gerçekteki sınıf ile tahmin edilen sınıf aynı olmalıdır
- Yandaki örnekte , bir model , class A sınıfında 50 veriyi incelemiş ve 45 adet doğru tahmin edilerek class A bulunmuş ancak 5 adeti , gerçekte class A olan sınıfı class B olarak tahmin edilmiş.
- Burada , bu doğruluk oranı bizim için yeterli diyebiliriz veya modelimizi daha fazla eğiterek veya değiştirerek daha doğru sonuçlar elde etmeye çalışabiliriz

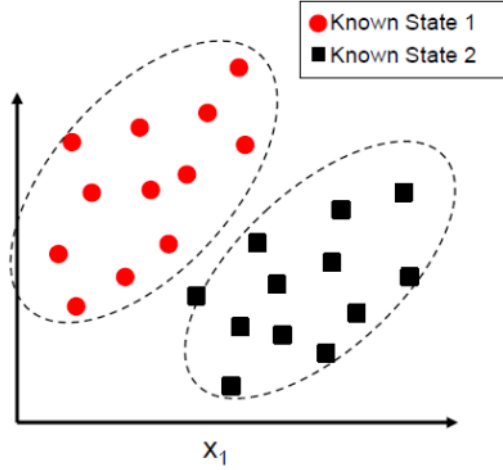
Makina Öğrenmesi Yöntemleri

Makina öğrenmesi yöntemleri

- **Supervised** (Gözetimli)
- **Unsupervised** (Gözetimsiz)



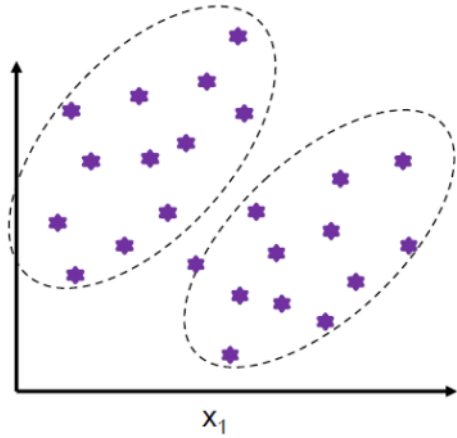
Gözetimli öğrenme



Supervised → you can fully classify

- Supervised Learning (Gözetimli öğrenme) :
 - Gözetimli öğrenmede etiketlenmiş veriler kullanılır. Yani eğitimde kullanılacak veri ve veri kategorileri , sınıfları bellidir.
 - Örneğin resimler içinde arabaları , bisikletleri ve kamyonları ayırmak istiyoruz. Öncelikle sistemde araba etiketiyle , araba resimleriyle bir veri seti oluştururuz. Sonra aynı şekilde bisiklet ve kamyon veri setlerini oluştururuz. Bu veri setlerine eğitim veri setleri (training data) denir
 - Bunun dışında içinde araba resimleri , bisiklet resimleri , kamyon resimleri olan test veri setleri oluştururuz. Bunlarla oluşan modeli test ederiz.

Gözetimsiz öğrenme



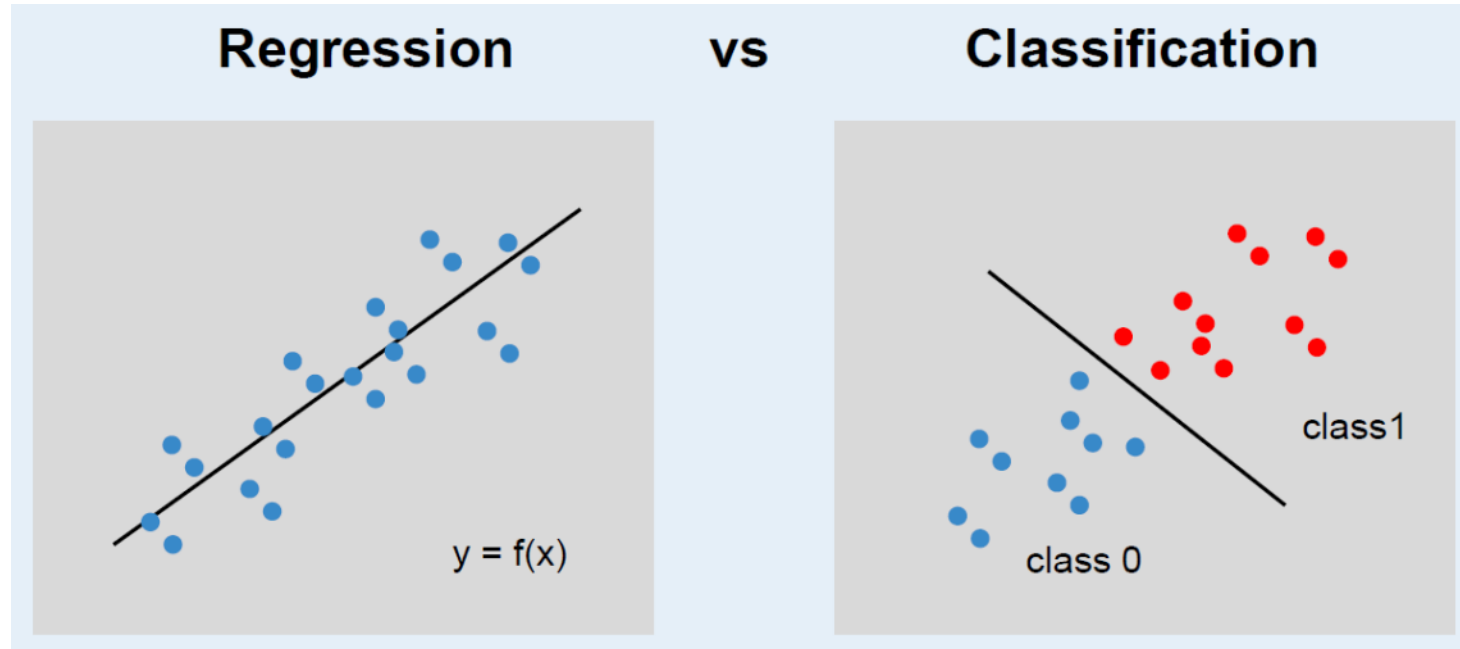
Unsupervised → you can cluster,
but not identify cluster label

- Unsupervised Learning (Gözetimsiz öğrenme) :
 - Gözetimsiz öğrenmede , algoritmalar veri setine bakarak bilgiler arasındaki bağlantıyı bulur , sınıflandırır ama o sınıfın veya sonucun ne olduğunu bilmez.

Makina Öğrenmesi Çeşitleri

Makina öğrenmesi

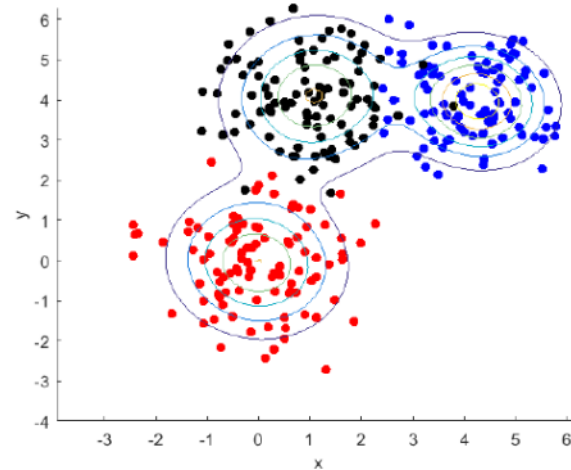
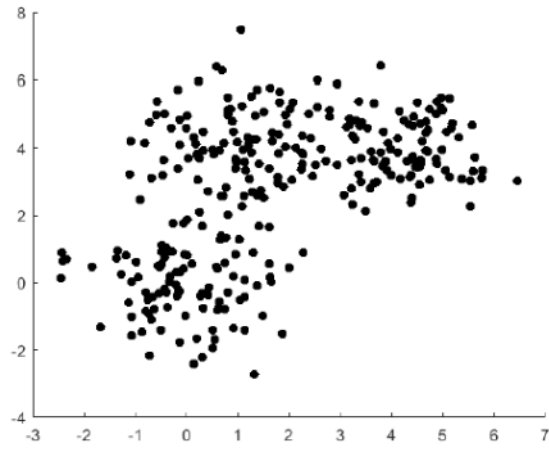
- **Classification** (Üretilen çıktı , Sınıflara ayırmak) Tahmin edilen sonuç arabamı, bisikletmi kamyonmu gibi sınıflar
- **Regression** (Üretilen çıktı , Sürekli sayı) Tahmin edilen sonuç bir evin fiyatı gibi sonuçlar



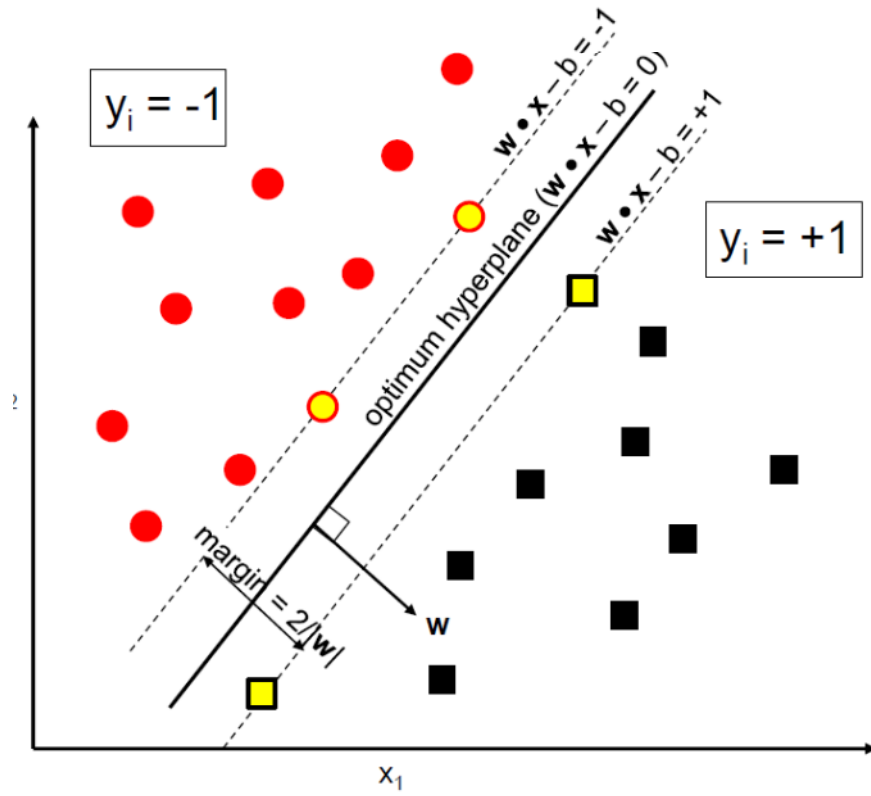
Karışık modeller

- Gaussian Mixture Models
 - Probabilistic
 - Modelinizi n-boyutlu Gaussian Distribution fonksiyonlarından oluşturursunuz

```
x = normrnd(0,1) + normrnd(4,1) + normrnd(1,1)  
y = normrnd(0,1) + normrnd(4,1) + normrnd(4,1)
```



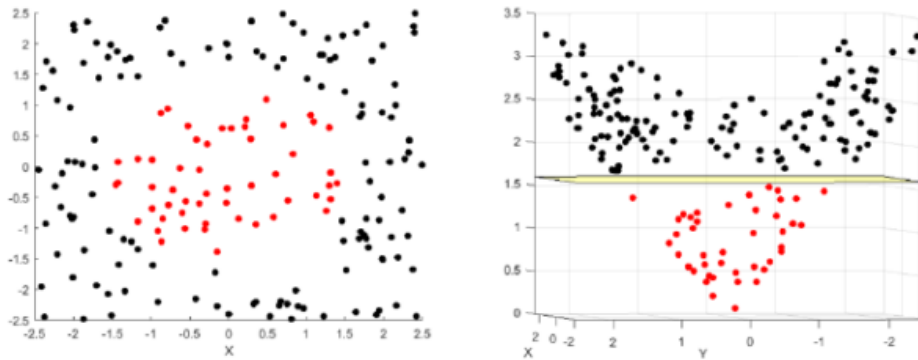
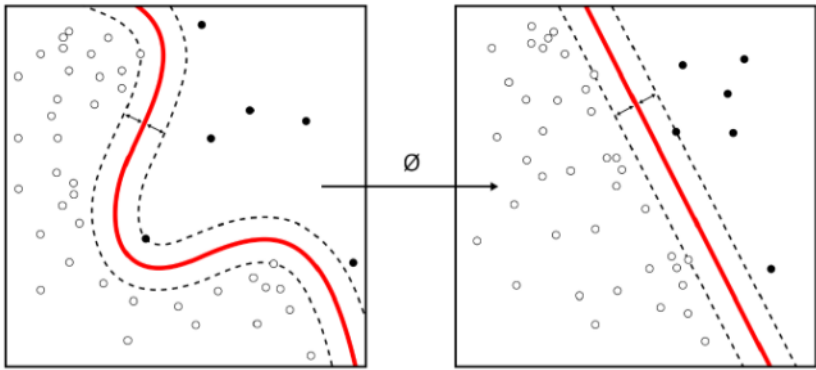
SVN (Support Vector Machines)



- Veri setinde birbirine benzeyen gruplararasına birbirinden en uzak olan noktalardan sınırlar çizmeye yarayan algoritmadır.
- Birçok farklı yüzey çizilebilir ancak asıl amaç , gruplarla arası en çok açık olacak yüzeyi bulmaktır

$$\phi(x_1, x_2) = (z_1, z_2, z_3) = (x_1, x_2, x_1^2 + x_2^2)$$

Kernel Trick

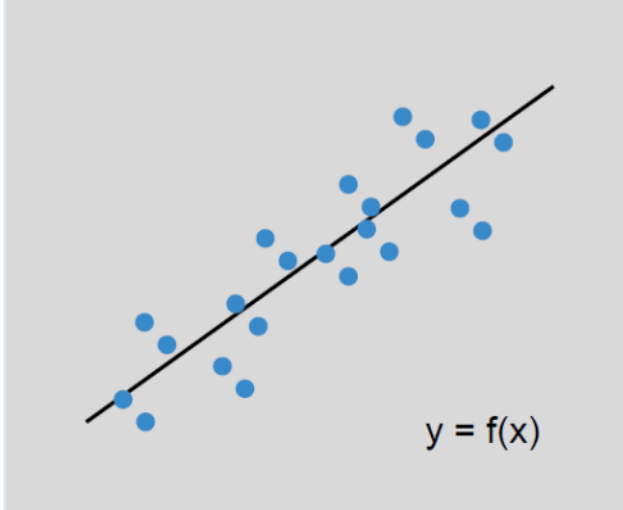


XY data not separable in 2D (left), but completely separable in 3D (right)

$$\phi(x_1, x_2) = (z_1, z_2, z_3) = (x_1, x_2, x_1^2 + x_2^2)$$

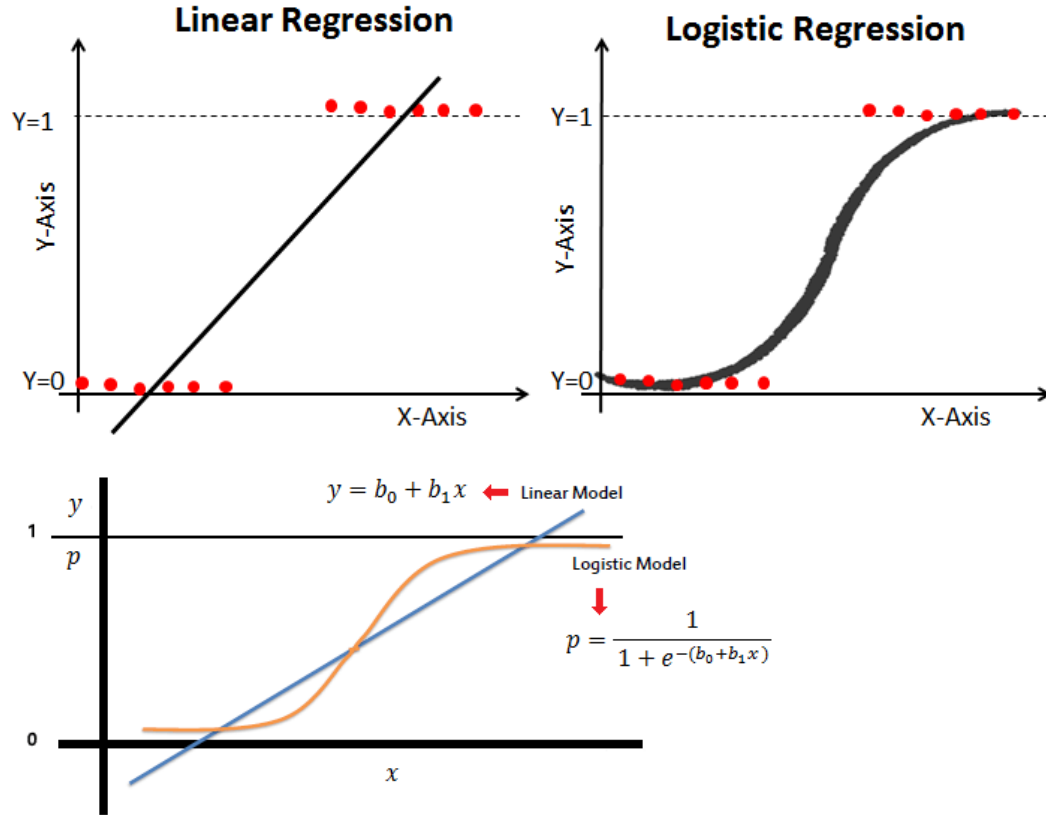
- SVM doğrusal olarak ayrılabilen veri setlerinde kullanılırken , Kernel doğrusal olarak ayrılamayan veri setlerinde kullanılır.
- Buradaki amaç bir fonksiyon yardımıyla üçüncü bir boyut ekleyerek , ayrımın yapılmasını sağlamaktır

Linear Regression



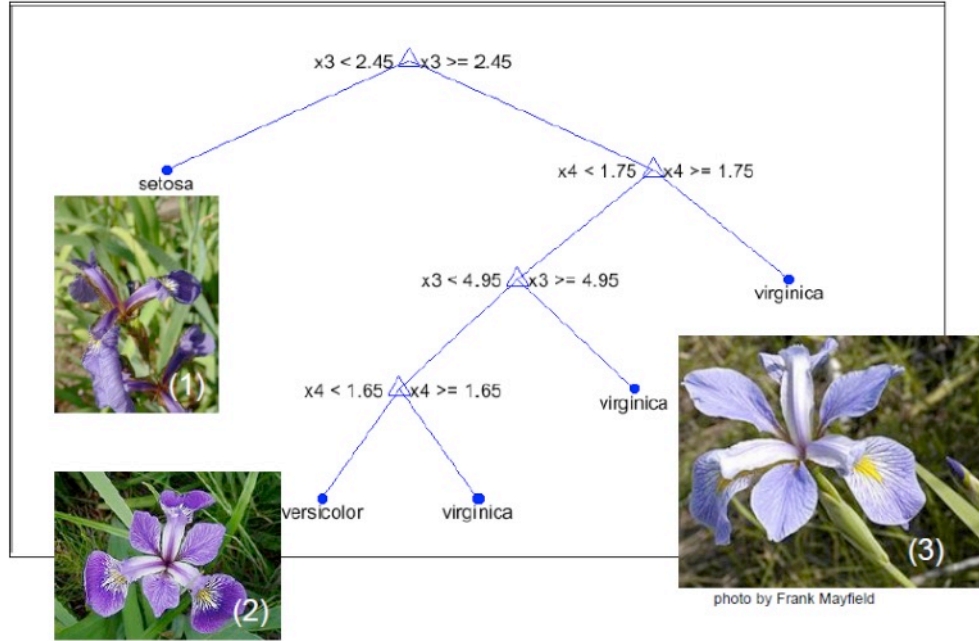
- Sayısal girdi ve çıktılar arasındaki doğrusal ilişkiyi tespit etmeyi sağlar. Düzlemde yayılmış verinin modelini en iyi biçimde doğrusal olarak çıkartmaya çalışan yöntemdir

Logistic Regression



- Bir sonucu belirleyen bir veya daha fazla bağımsız değişken bulunan veri kümesini analiz etmek için düzlemde en iyi eğriyi yakalamaya çalışan istatistiksel bir yöntemdir. Sonuç, ikiye bölünmüş katagorik bir değişkenle ölçülür (sadece iki olası sonuç vardır).
- Linear model $y=b_0 + b_1x$ iken logistic model logaritmiktir

Decision Tree (Karar Ağacı)



- Petal (Taç yaprağı uzunluğuna ve genişliğine göre)
- Sepal (Çanak yaprağı uzunluğuna ve genişliğine göre)
- Fisher's Iris verisine göre çiçeğin hangi çiçek olduğuna karar veriyor. Bu veriyi internette bulabilirsiniz

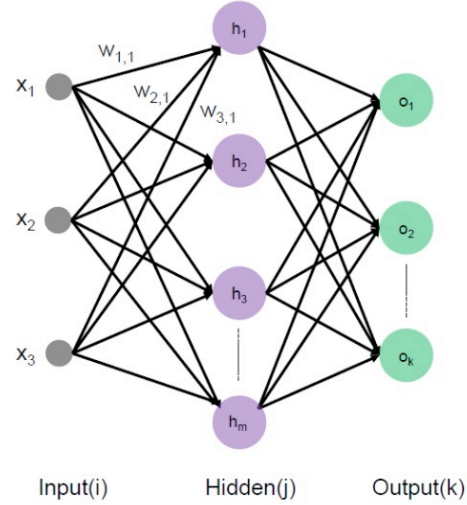
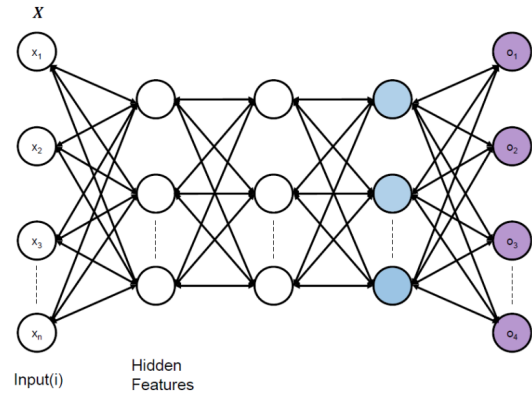
Fisher's Iris Data (https://en.wikipedia.org/wiki/Iris_flower_data_set)				
Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	<i>I. setosa</i>
4.9	3.0	1.4	0.2	<i>I. setosa</i>
4.7	3.2	1.3	0.2	<i>I. setosa</i>
4.6	3.1	1.5	0.2	<i>I. setosa</i>
5.0	3.6	1.4	0.3	<i>I. setosa</i>

Sinir Ağları

For a single neuron in the hidden layer:

$$h_j = f_A \left(\sum_{i=1}^n w_{i,j} x_i + b \right)$$

where f_A is the activation function (often a hyperbolic tangent or ReLU)



- Bir sinir ağında , her bir girdiyi alır , belirlediğimiz ağırlıklarla çarpıp sonuçları toplarız ve düğümleri oluştururuz. Bu düğümler , dikey olarak ara katmanları oluşturur.
- Yine bu ara katmanlardan çıkan sonuçları belirlediğimiz ağırlıklarla çarparak her bir çıktı için toplarız ve sonuca ulaşmaya çalışırız
- Bu sayede birçok karmaşık fonksiyon elde ederiz
- Bu lineer bir işlemdir.
- Bu işlemi , nonliner sistemlerde kullanmak için , aktivasyon fonksiyonlarını kullanırız.

Aktivasyon fonksiyonları

- Aktivasyon fonksiyonları , sinir ağlarının non-linear modelleme yapabilmesi için kullanılır

sigmoid

$$f(x) = \frac{1}{1 + e^{-x}}$$

tanh

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

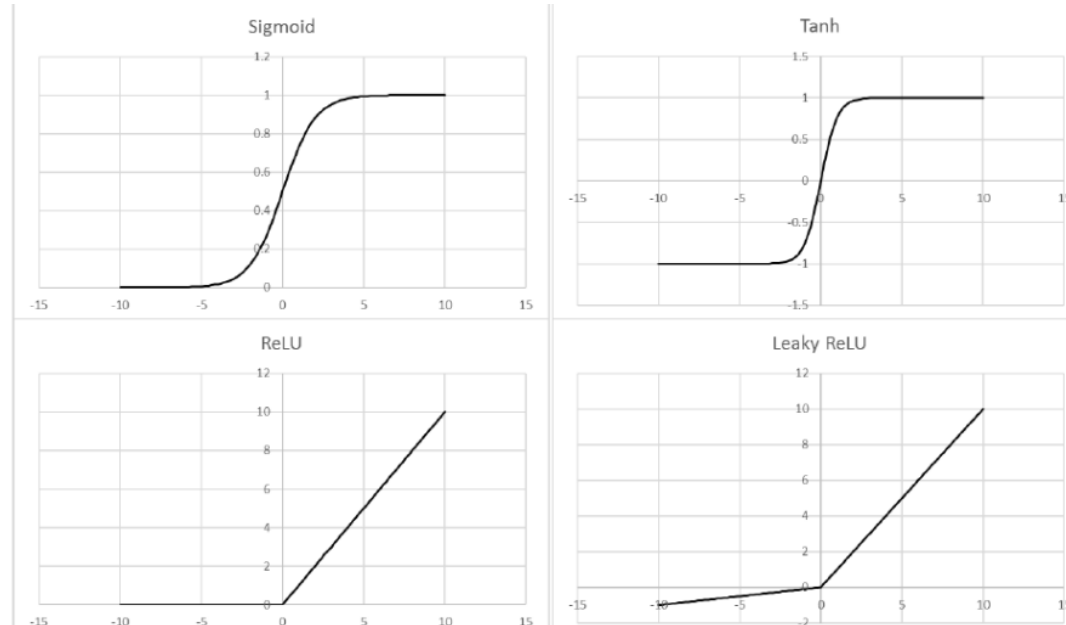
Rectified
linear unit
(ReLU)

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

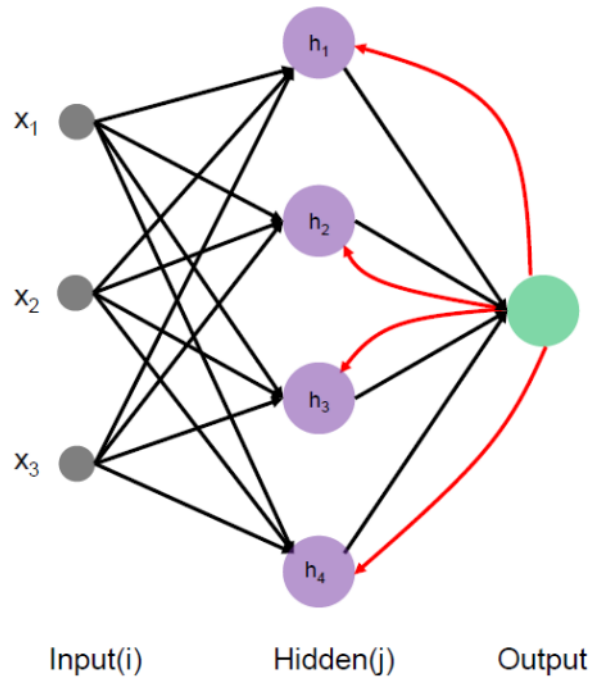
Leaky ReLU

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

where $\sigma \ll 1$

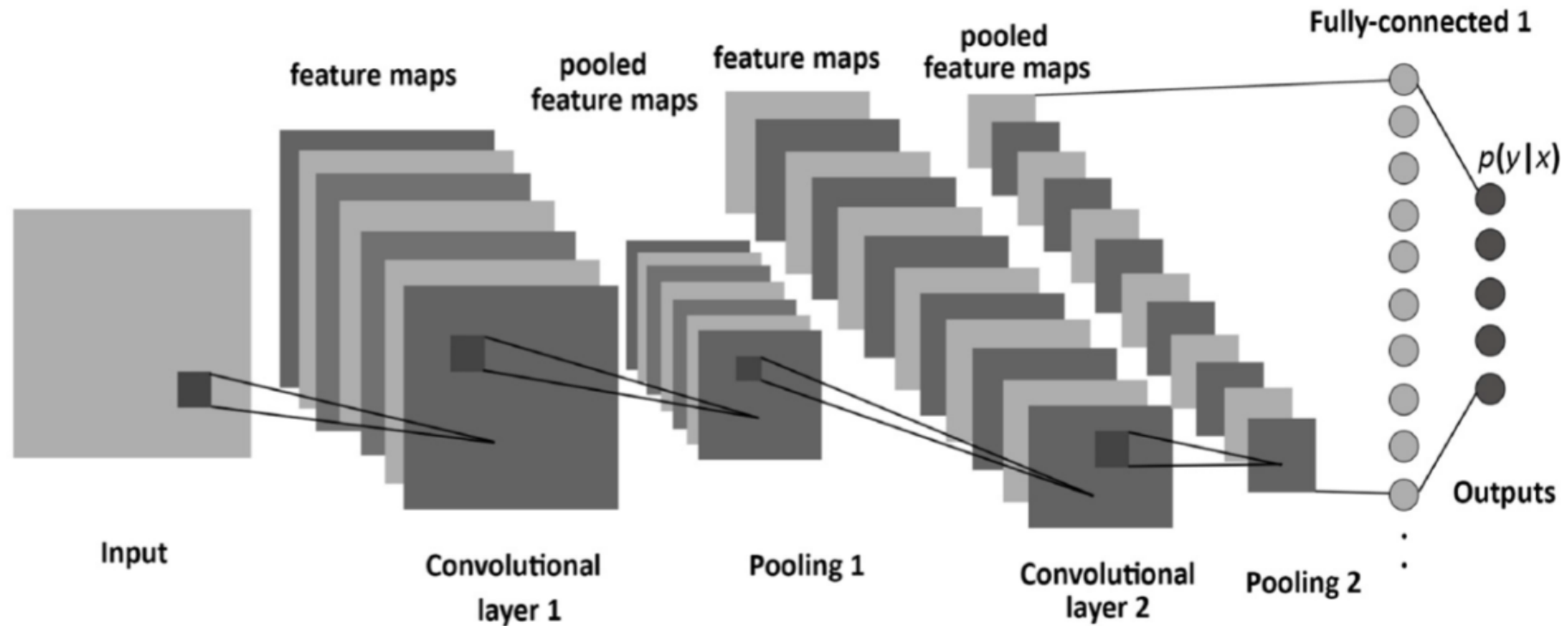


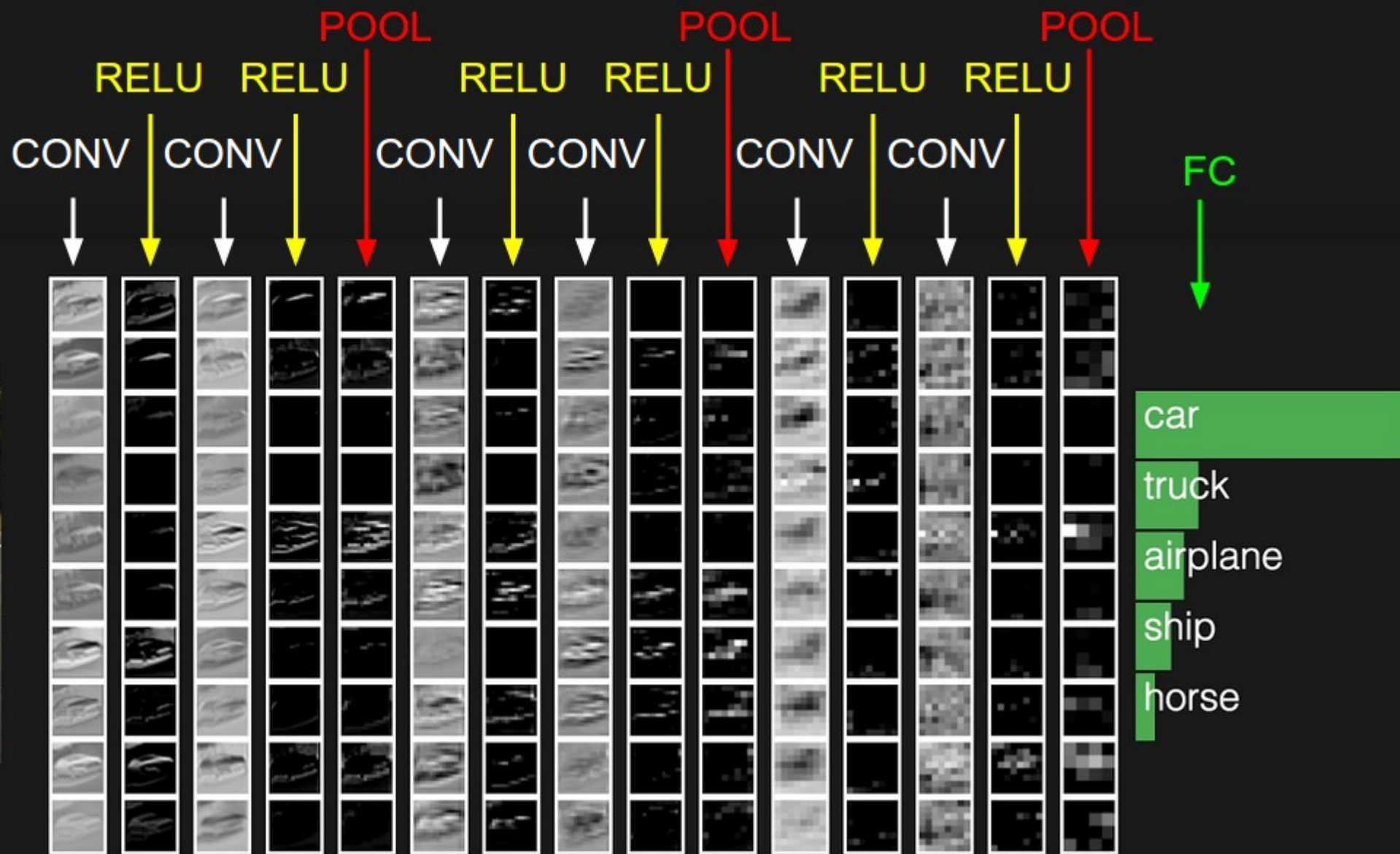
Recurrent Neural networks



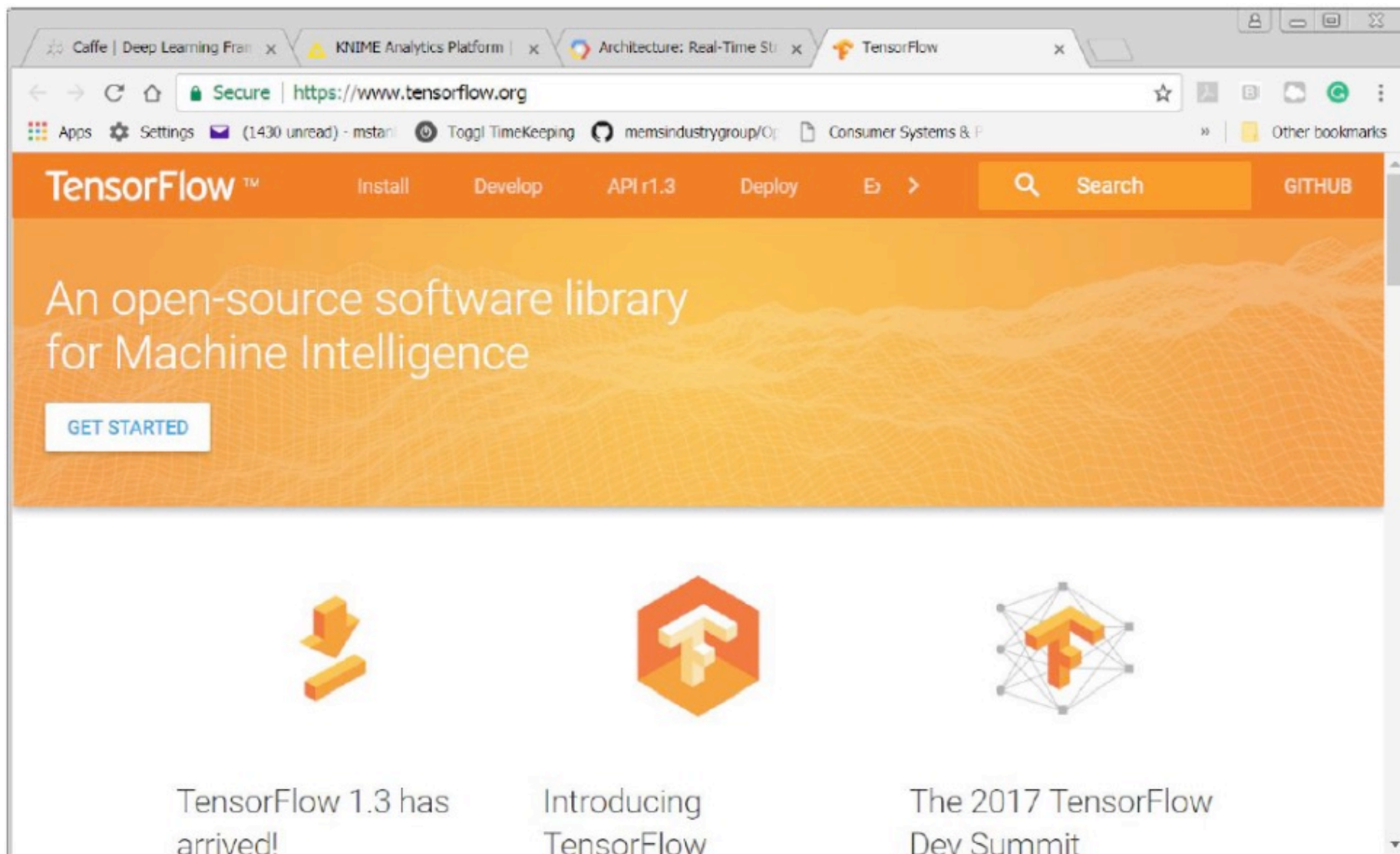
- Geçmişide hesaba katan sinir ağlarıdır
- Daha önce bulunan sonuçlarıda hesaba katar ve ara katman değerlerini bu şekilde değiştirir

Convolutional Neural Networks





- Makina öğrenmesi algoritmamızı tanımlarken nelere dikkat etmeliyiz
 - Eğitim zamanı (training time)
 - Doğruluk , hassasiyet (accuracy)
 - Doğrusallık (linearity)
 - Özellik sayısı (Number of features)
 - Hesaplama gücü (number of CPU and RAM)
 - Tolere edilebilir gecikme (Latency)
 - Kolay uygulanabilirlik
 - Eğitim için gerekli veri miktarı (Amount of data required for training)
 - Etiket bilgisi (Availability of label data)





eIQ Machine Learning Software ▾

[Actions ▾](#)

[Overview](#) [Content](#) [People](#) [Subspaces](#)

 [Log in](#) to follow, share, and participate in this community.

Get started with eIQ on

i.MX Applications Processors

-  [Getting started on eIQ if you have an i.MX Applications Processors](#)
-  [eIQ download information](#)
-  [eIQ sample applications for i.MX Applications Processors](#)
-  [eIQ frequently asked questions \(FAQ\)](#)

Microcontrollers

-  [Getting started on eIQ if you have a microcontroller](#)
-  [eIQ download information](#)
-  [eIQ porting guides for MCUs](#)
-  [eIQ demo input guide for MCUs](#)
-  [eIQ frequently asked questions \(FAQ\)](#)

Recent Content

[Filter by Categories & Tags ▾](#)


eIQ Heterogenous Demo

2 days ago

by Raluca Popa 



al-link.jspa?url=https%3A%2F%2Fwww.nxp.com%2F

Featured Videos



[eIQ\(TM\) Machine Learning Software Environment demo](#)



[AI Public Safety Detection Demo with Arcturus](#)

Teşekkürler