BİL 475 Örüntü Tanıma

Hafta-2:

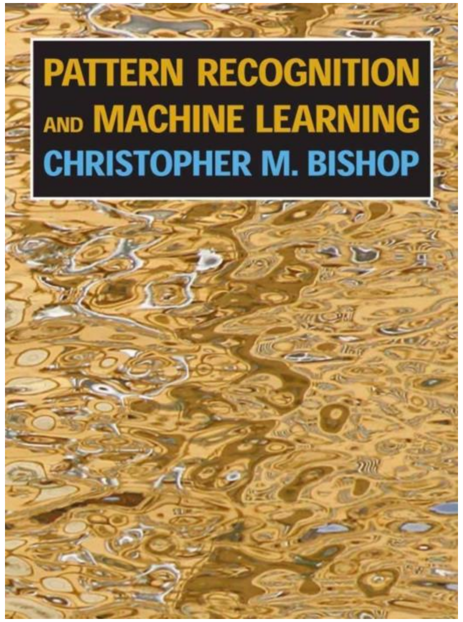
Vektör ve Matris İşlemleri

K-En Yakın Komşu Algoritması

Ana Kaynaklar:

Pattern Recognition and Machine Learning by Christopher M. Bishop

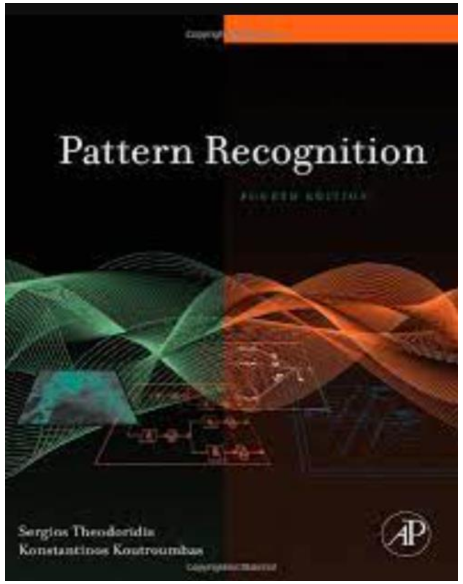
Pattern Recognition, Sergios Theodoridis



Yardımcı Kaynaklar:

https://www.youtube.com/watch?v=esTIhqAFKu4&list=PLcXJymqaE9PPGGtFsTNoDWKl-VNVX5d6b&index=2&ab_channel=CBCSLteaching

https://hci.iwr.uni-heidelberg.de/people/fhamprec



Pattern Recognition Video Lectures (Summer Term 2012)

Full playlist here.

1 Introduction
- 1.1 Applications of Pattern Recognition
- 1.2 k-Nearest Neighbors Classification
- 1.3 Probability Theory
- 1.4 Statistical Decision Theory

2 Correlation measures, Gaussian Models
- 2.1 Pearson Correlation
- 2.2 Alternative Correl. Measures
- 2.3 Gaussian Graphical Models
- 2.4 Discriminant Analysis

3 Dimensionality Reduction
- 3.1 Regularized LDA/QDA
- 3.2 Principal Component Analysis (PCA)
- 3.3 Bilinear Decompositions

4 Neural Networks
- 4.1 History of Neural Networks
- 4.2 Perceptrons

Öznitelik Kavramı

Sayısal Veri;

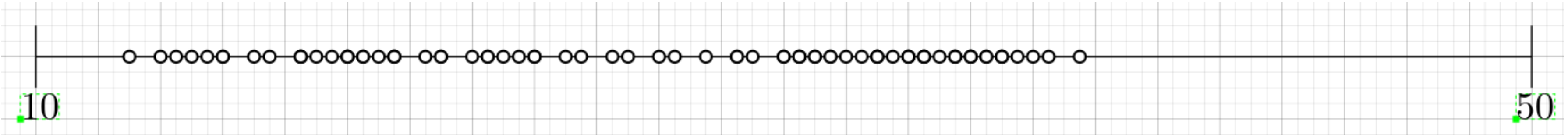
Olguların Sayılar Alemindeki Karşılığı

Öznitelik Kavramı

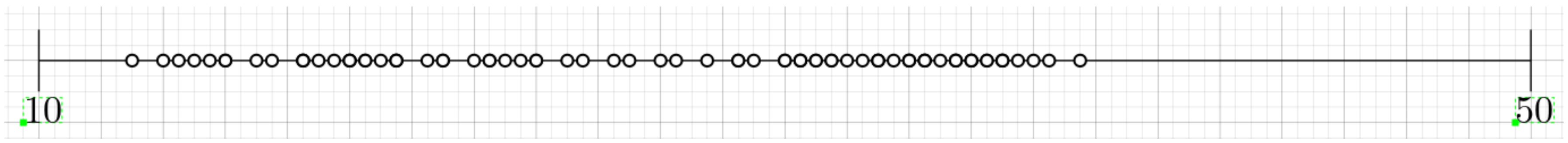
Sayısal Veri Analizi;

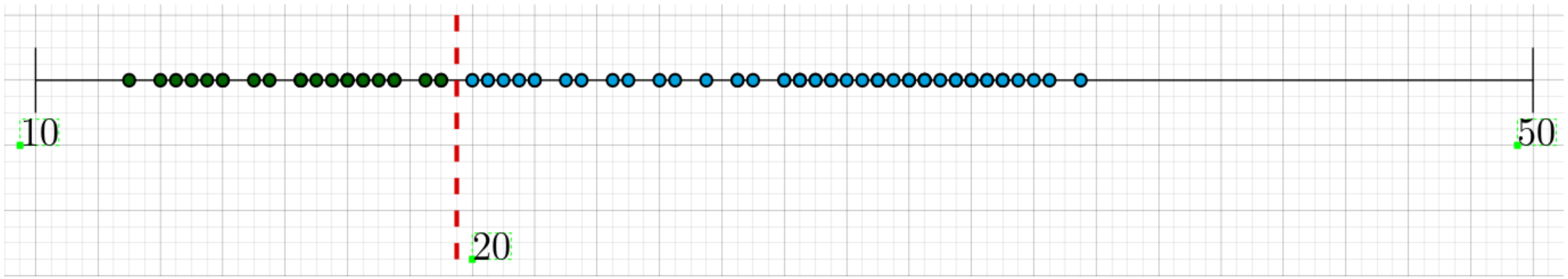
Analitik Sistemde
Bir Düzen (Pattern) Arayışıdır.

Öznitelik Kavramı



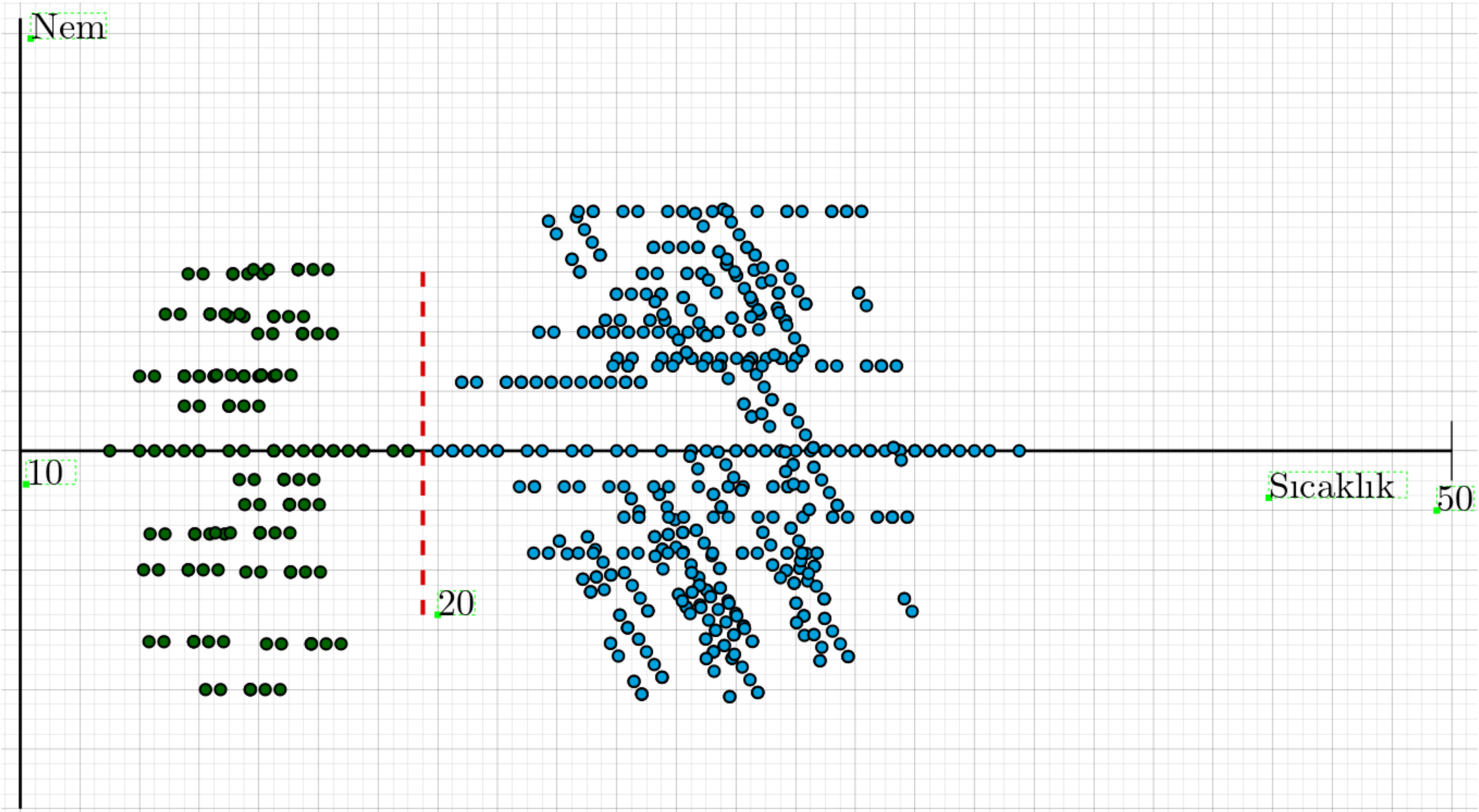
Öznitelik Kavramı



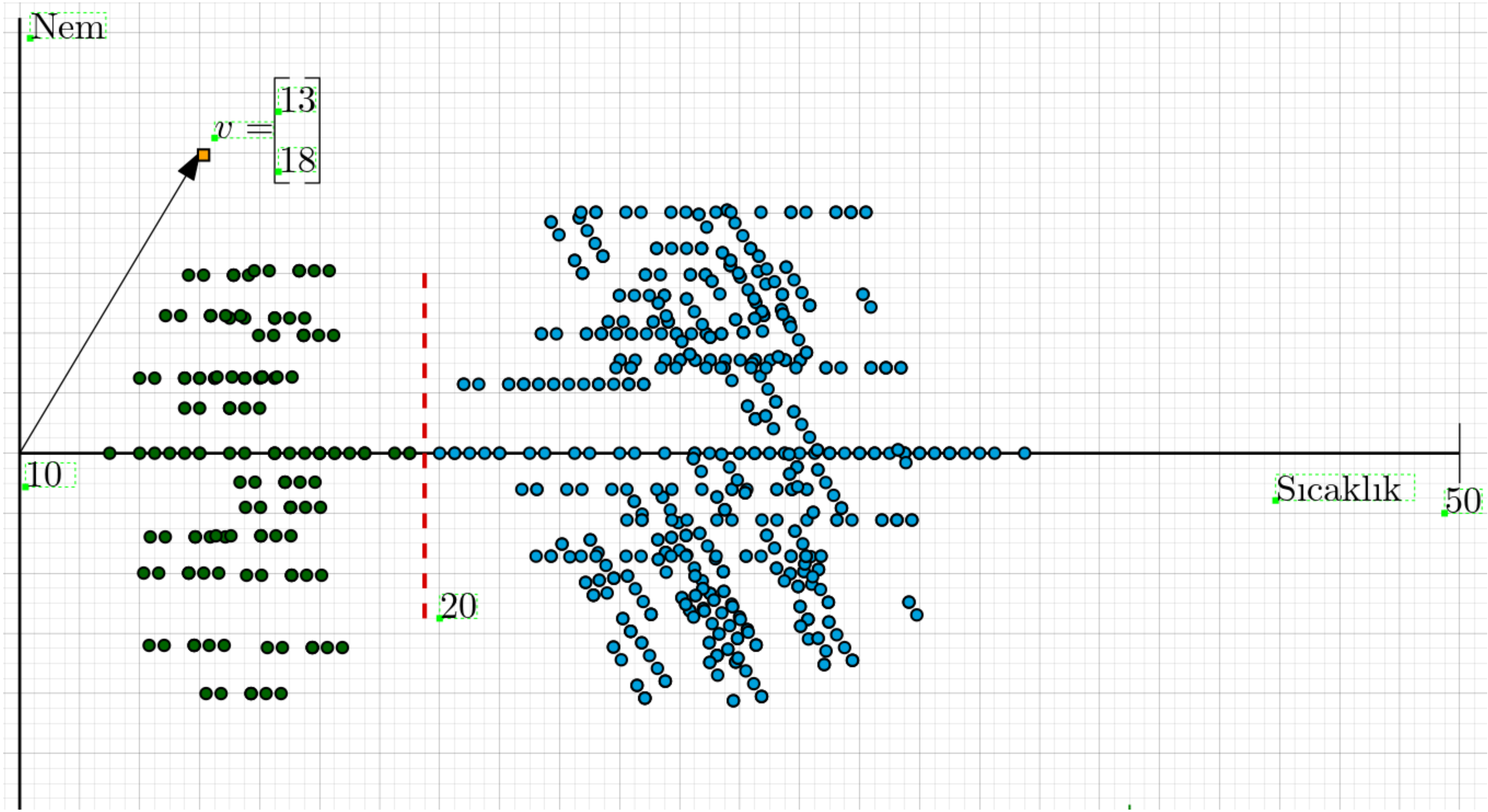


$$IF\ (x \geq 20\ )$$

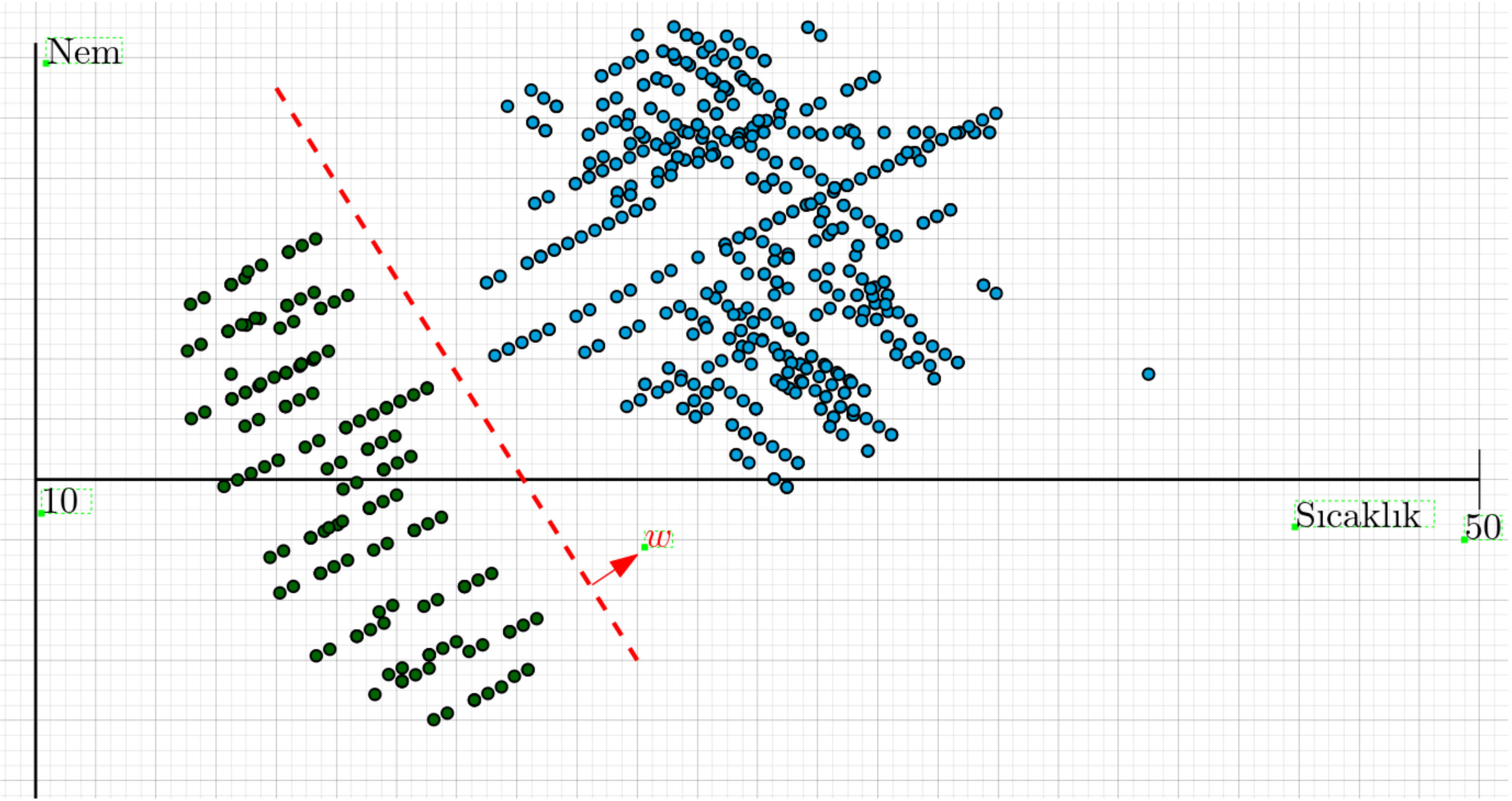
Öznitelik Kavramı



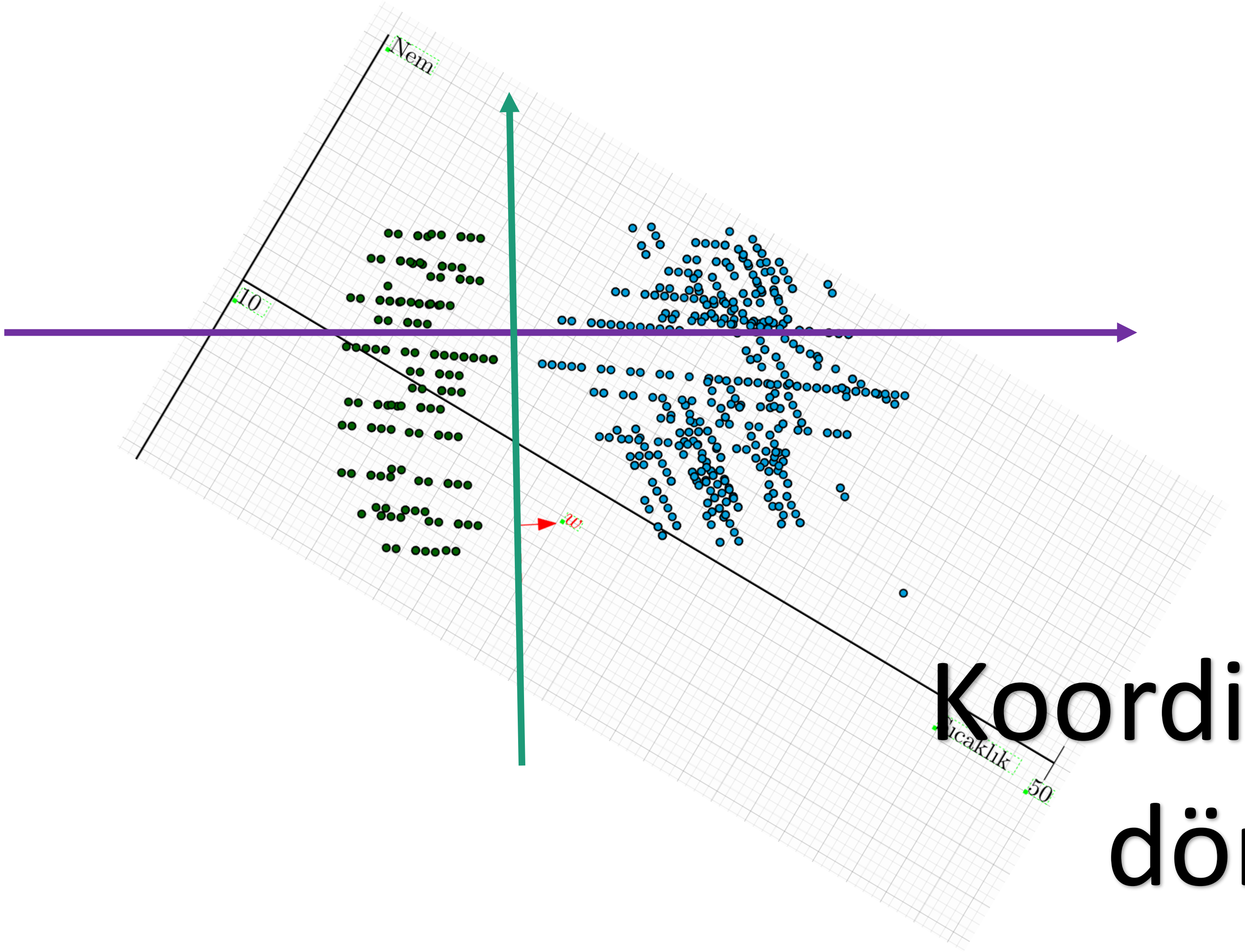
Öznitelik Kavramı



Öznitelik Kavramı



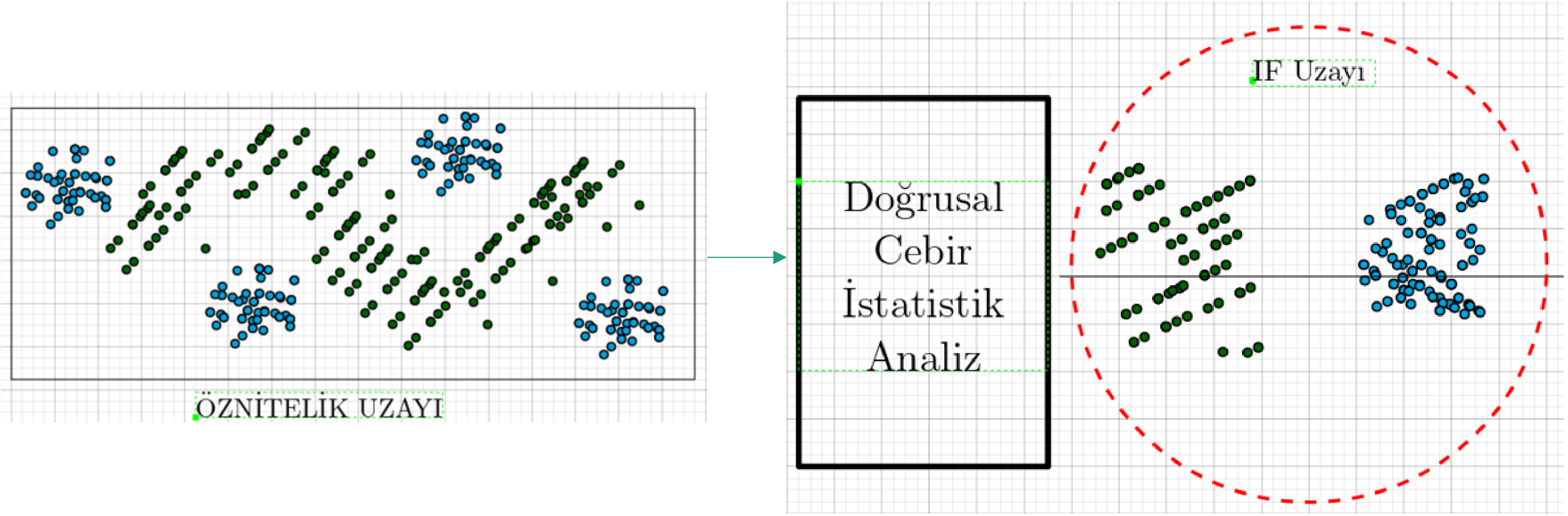
Öznitelik Kavramı



Koordinat sistemi dönüşümü

Öznitelik Kavramı

Öznitelik Kavramı

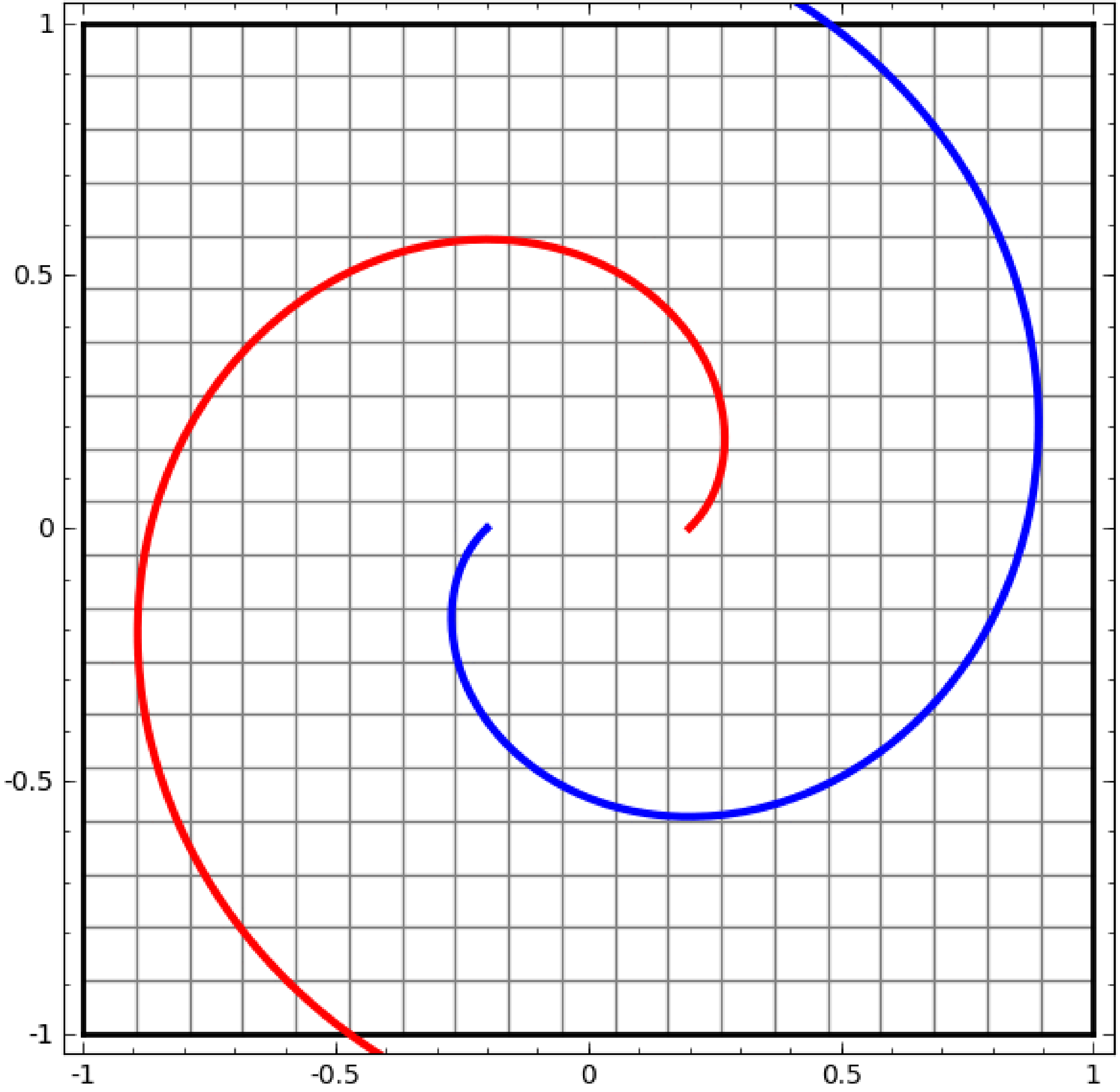


Öznitelik Kavramı

Makine Öğrenmesinin Amacı

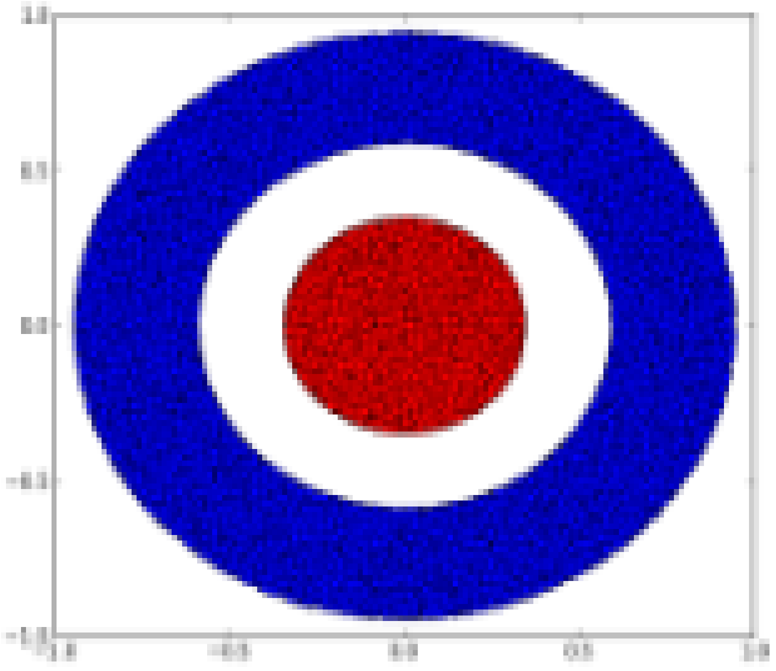
Doğrusal ayrıştırılabilir bir uzaya

koordinat dönüşümü yapmak



http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

Öznitelik Kavramı



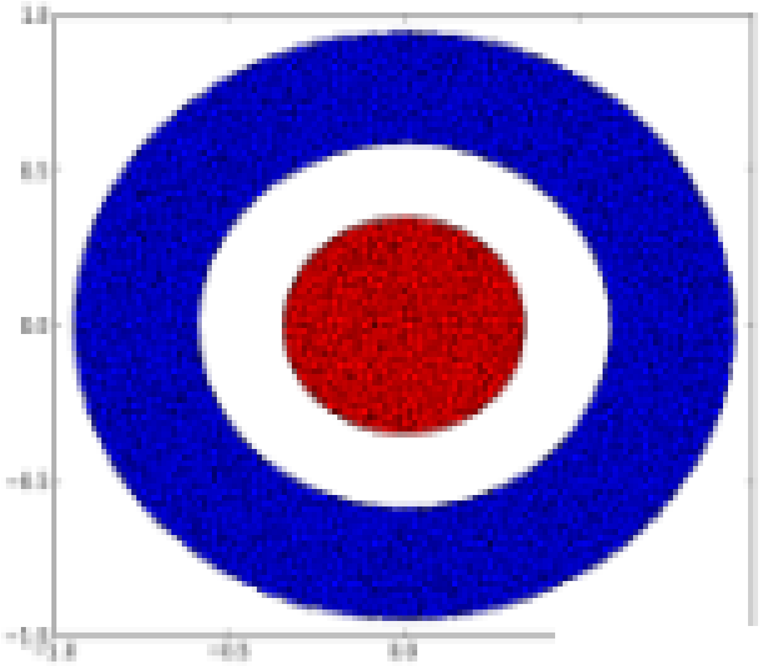
Makine Öğrenmesinin Amacı

Doğrusal ayrıştırılabilir bir uzaya

koordinat dönüşümü yapmak

http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

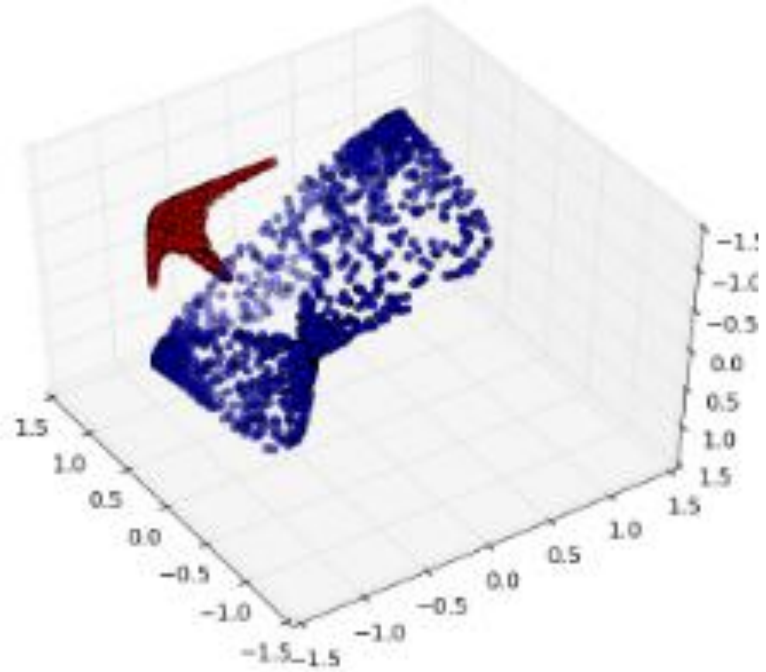
Öznitelik Kavramı



Makine Öğrenmesinin Amacı

Doğrusal ayrıştırılabilir bir uzaya

koordinat dönüşümü yapmak

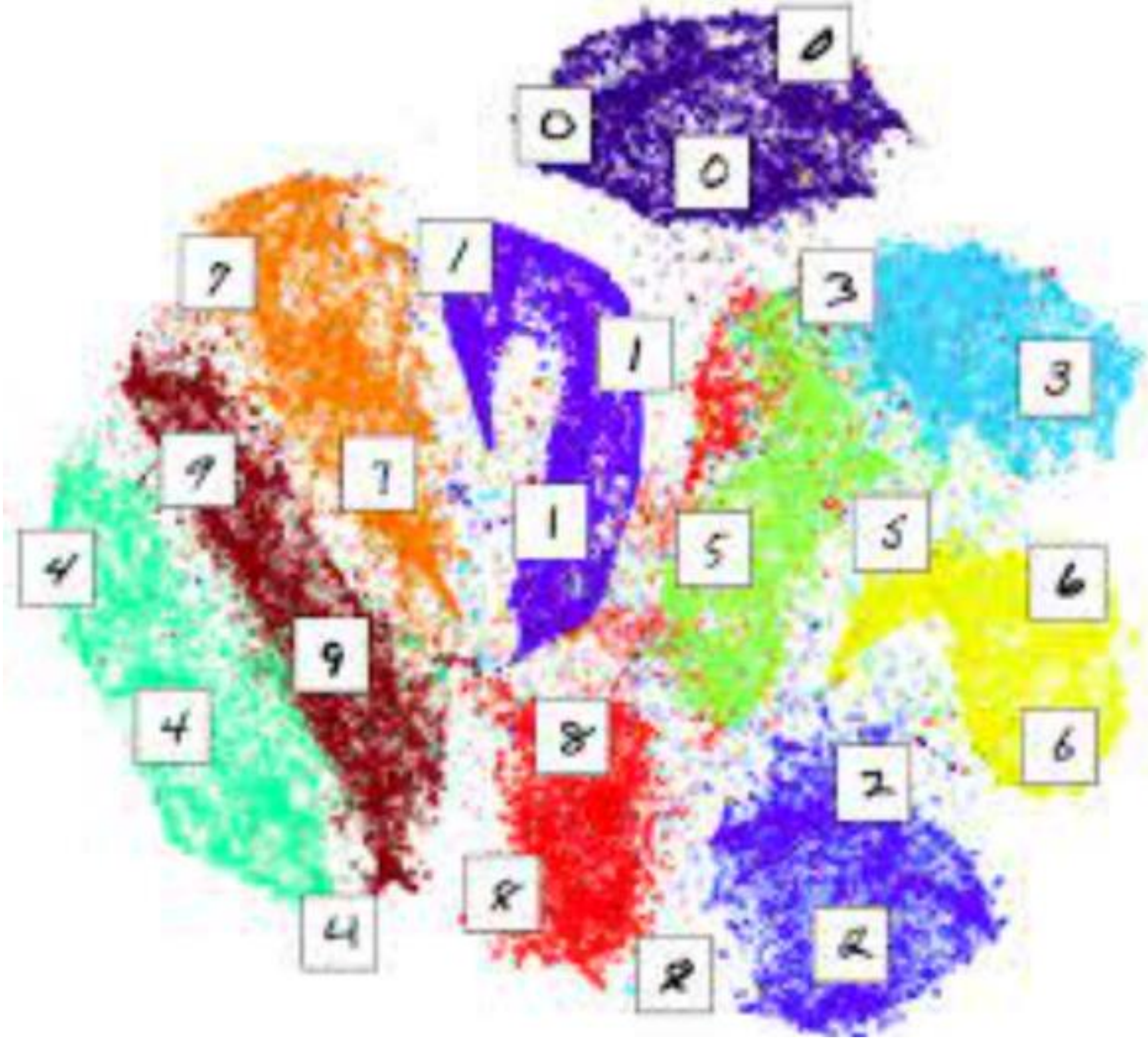


---

http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

Öznitelik Kavramı

https://github.com/scienceai/tsne-js



https://nlml.github.io/in-raw-numpy/in-raw-numpy-t-sne/

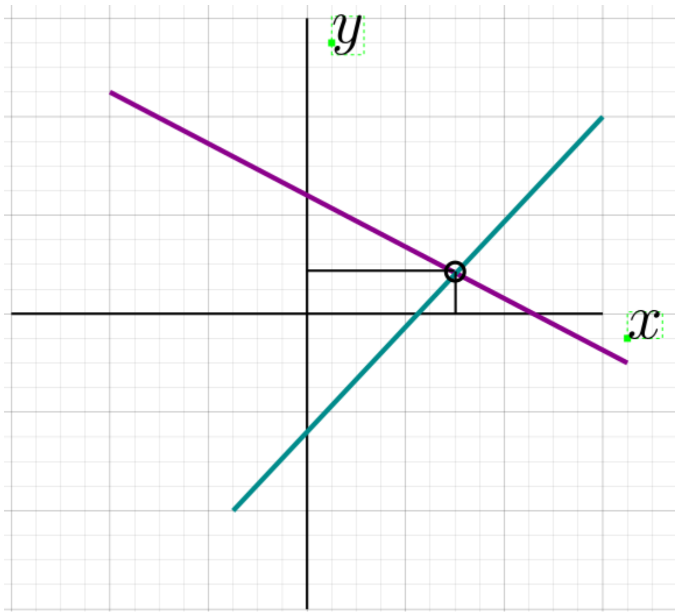
Sayılar, Vektörler ve Matrisler

Fiziksel Olgulardan Matematiğe Geçiş

$$3x + 7 = 11$$

$$3x + 7y = 11$$
$$2x + 3y = 1$$

$$3x + 7y - z = 11$$
$$2x + 3y + z = 1$$
$$2x + 2y + z = -5$$



$N \times N$ için makul değil!

Sayılar, Vektörler ve Matrisler

Fiziksel Olgulardan Matematiğe Geçiş

$$3x + 7 = 11$$

$$3x + 7y = 11$$
$$2x + 3y = 1$$

$$3x + 7y - z = 11$$
$$2x + 3y + z = 1$$
$$2x + 2y + z = -5$$

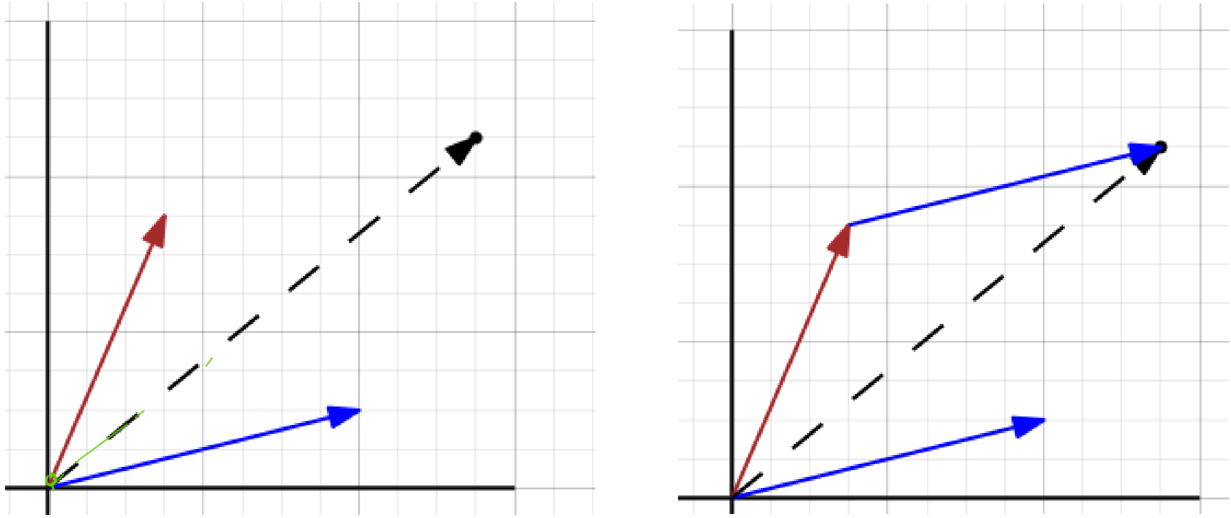
Daha sistemli (algoritmik) bir yöntem ??

$$\begin{bmatrix} 3 & 7 & -1 \\ 2 & 3 & 1 \\ 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 11 \\ 1 \\ -5 \end{bmatrix}$$

$$\begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix} x + \begin{bmatrix} 7 \\ 3 \\ 2 \end{bmatrix} y + \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix} z = \begin{bmatrix} 11 \\ 1 \\ -5 \end{bmatrix}$$

Sayılar, Vektörler ve Matrisler

Fiziksel Olgulardan Matematiğe Geçiş



Baz vektörleri ile diğer vektörleri ifade edebilme (Taylor , Fourier)

$$\begin{bmatrix}3\\2\\2\end{bmatrix}x+\begin{bmatrix}7\\3\\2\end{bmatrix}y+\begin{bmatrix}3\\2\\2\end{bmatrix}z=\begin{bmatrix}11\\1\\-5\end{bmatrix}$$

Sayılar, Vektörler ve Matrisler

Vektör İşlemleri

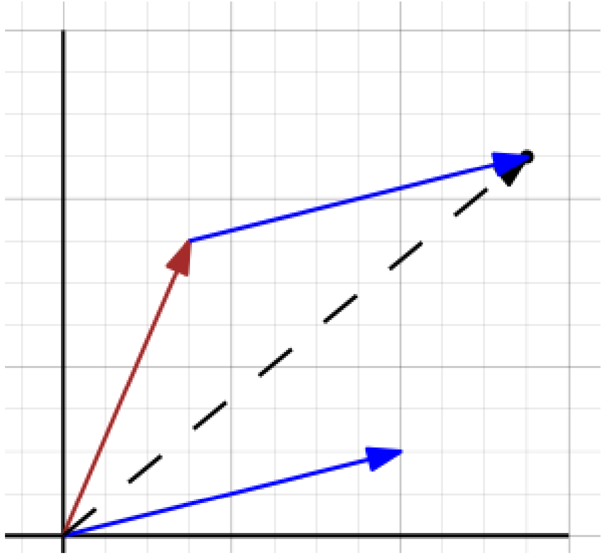
Toplama

İç çarpım (Bra-ket)

Transpoze

Sayılar, Vektörler ve Matrisler

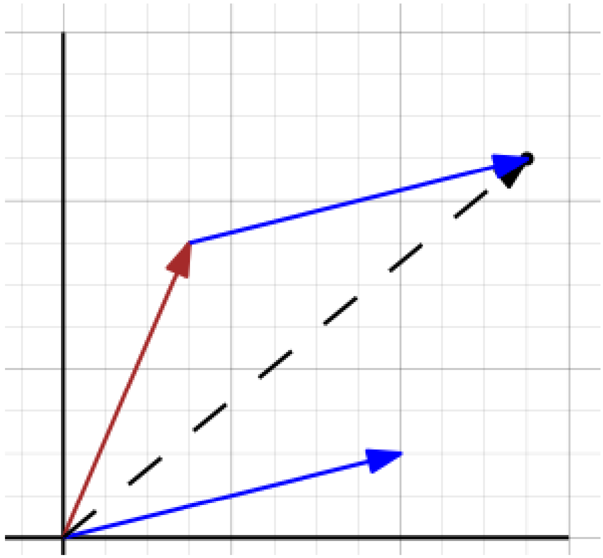
Vektör İşlemleri



Doğrusal Dönüşüm
(Linear Transformation)

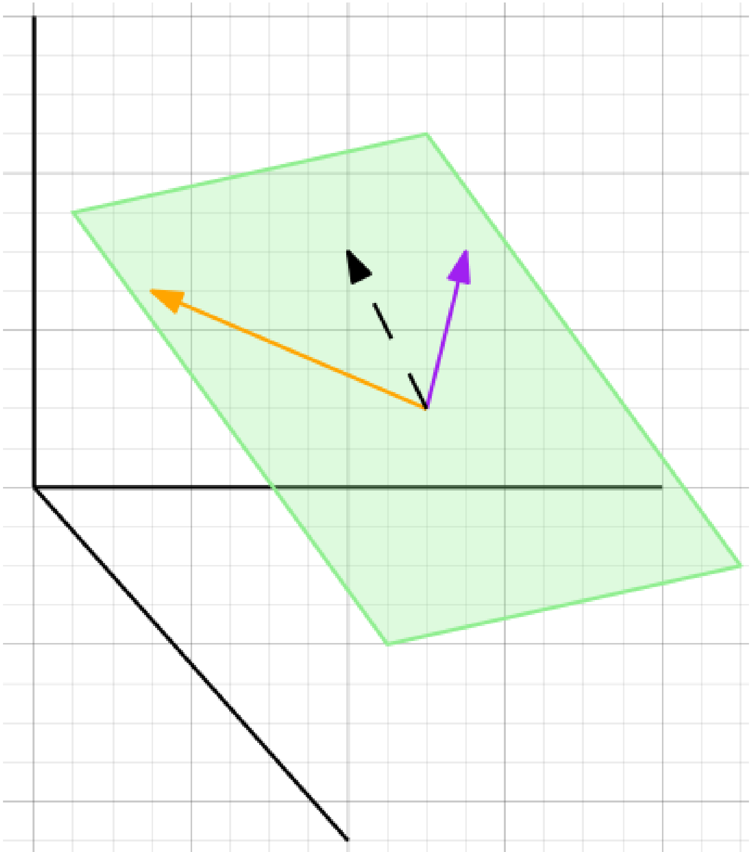
Sayılar, Vektörler ve Matrisler

Vektör İşlemleri



Doğrusal Dönüşüm
(Linear Transformation)

Uzayın Germe: İstanbul'da Yemek Kültürü



Sayılar, Vektörler ve Matrisler

Vektör İşlemleri

```
clear all, close all; clc

v1 = [1,2,3]';
v2 = [2,2,3]';
v3 = [1,1,1]';

V = [v1,v2,v3]

rank(V)
```

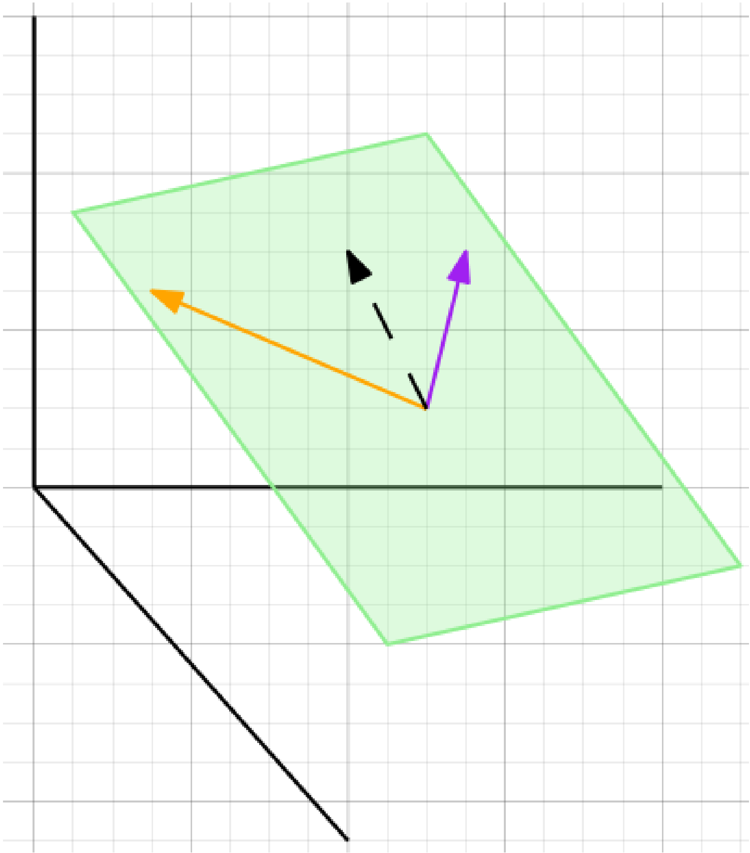
```
V =

     1     2     1
     2     2     1
     3     3     1


ans =

     3
```

Uzayın Germe: İstanbul'da Yemek Kültürü



Sayılar, Vektörler ve Matrisler

Vektör İşlemleri

```
clear all, close all; clc

v1 = [1,2,3]';
v2 = [2,2,3]';
v3 = randn(1)*v1 + randn(1)*v2;

V = [v1,v2,v3]

rank(V)
```

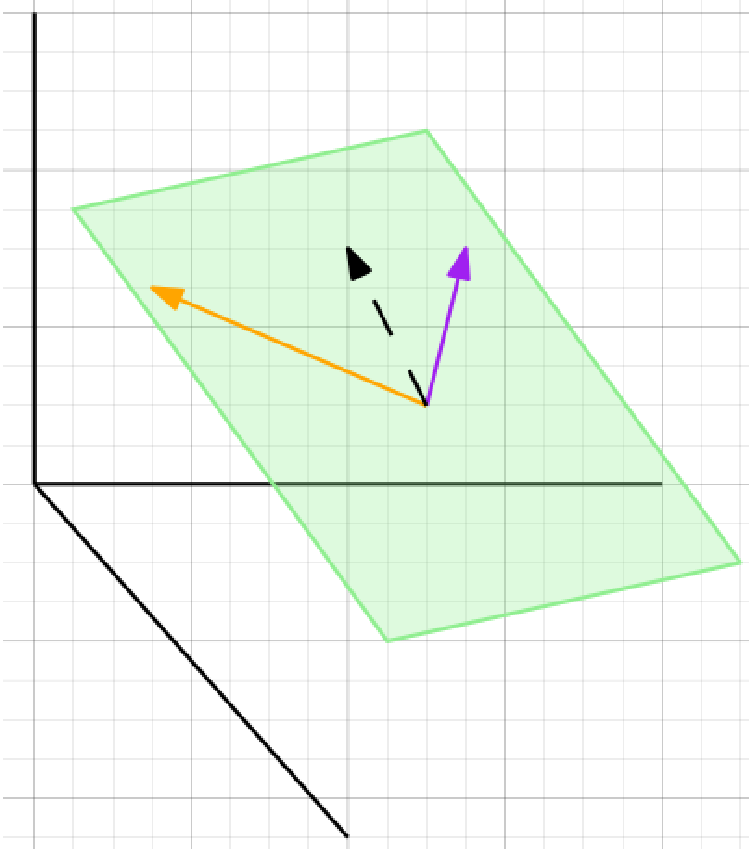
```
V =

    1.0000    2.0000   -3.7624
    2.0000    2.0000   -4.0244
    3.0000    3.0000   -6.0366


ans =

     2
```

Uzayın Germe: İstanbul'da Yemek Kültürü

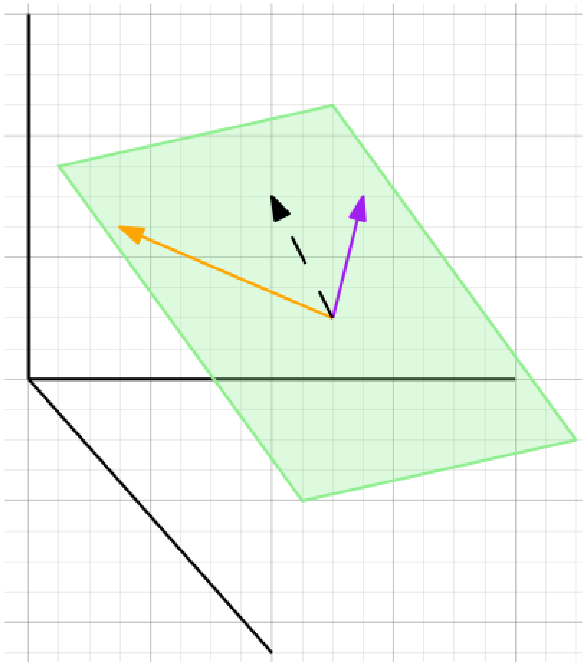


Sayılar, Vektörler ve Matrisler

Vektör İşlemleri

Uzayın Germe ve Doğrusal Bağımsızlık

Baz vektörleri,
gerdikleri uzay içinde
herhangi bir noktayı oluşturabilirler.

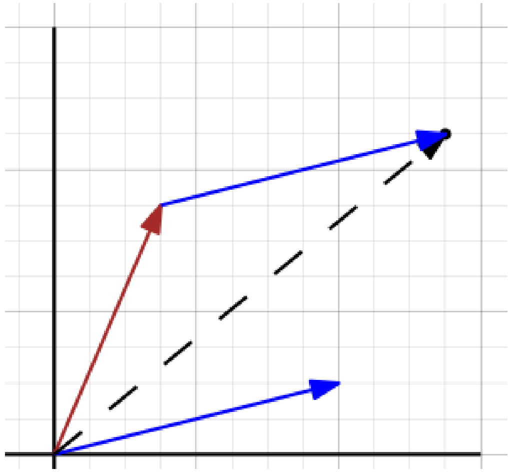


Bu kural N boyutlu uzaylar için geçerlidir. !!

Göremesek bile sayılar bize yol gösterici
Örnek: Norm

Sayılar, Vektörler ve Matrisler

Vektör İşlemleri

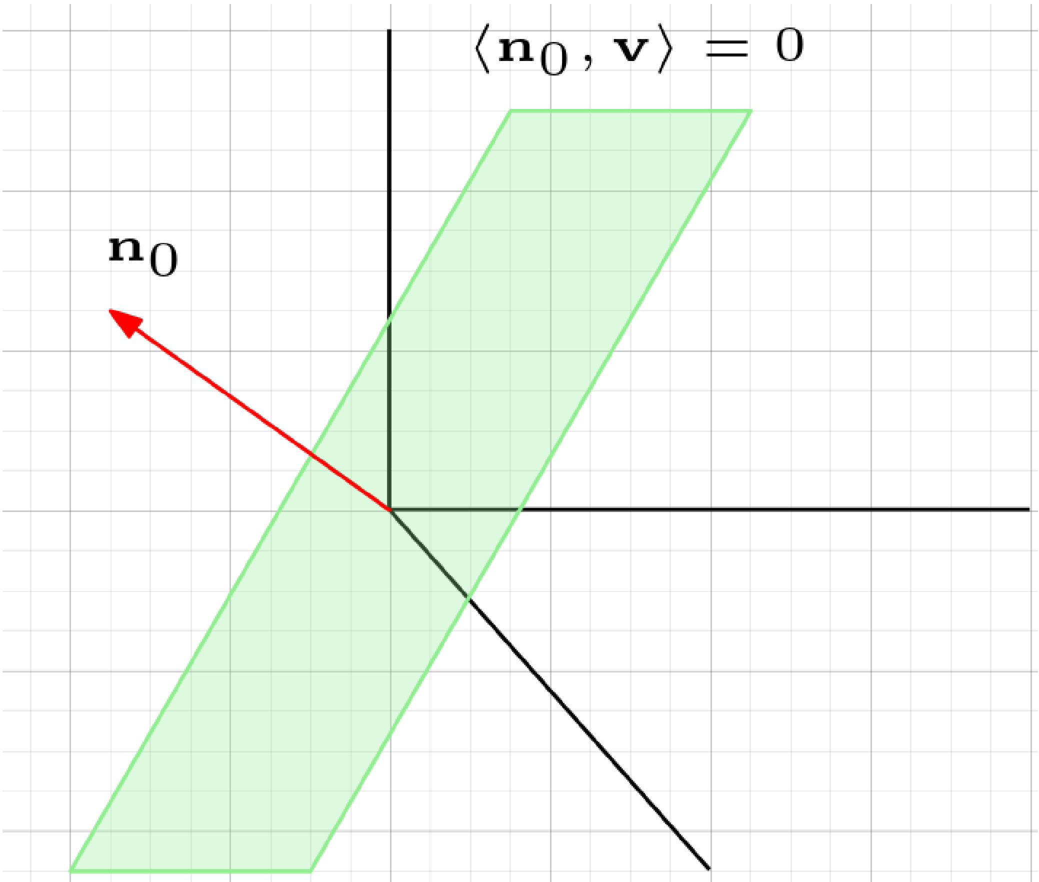


Determinant !!

$$\det(\boldsymbol{A}) = ??$$

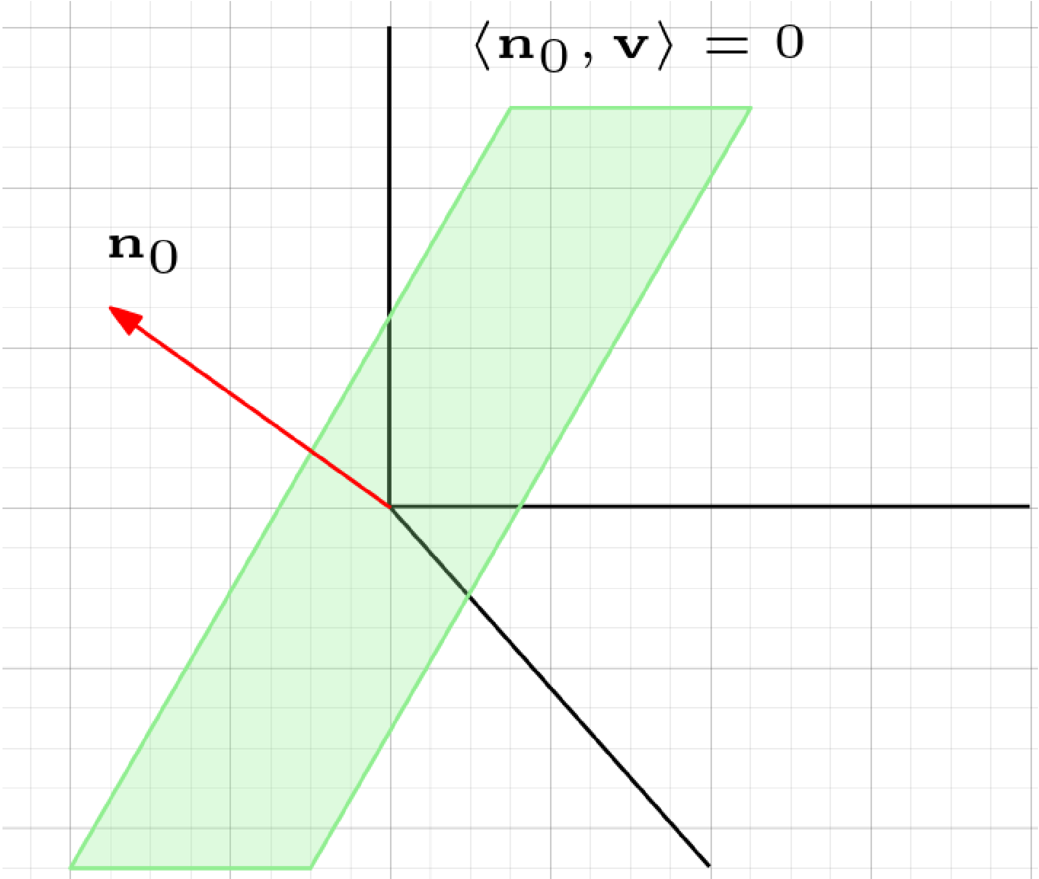
Sayılar, Vektörler ve Matrisler

- Düzlem Denklemi (İpucu: İki vektör dikse iç çarpım 0’dır):



Sayılar, Vektörler ve Matrisler

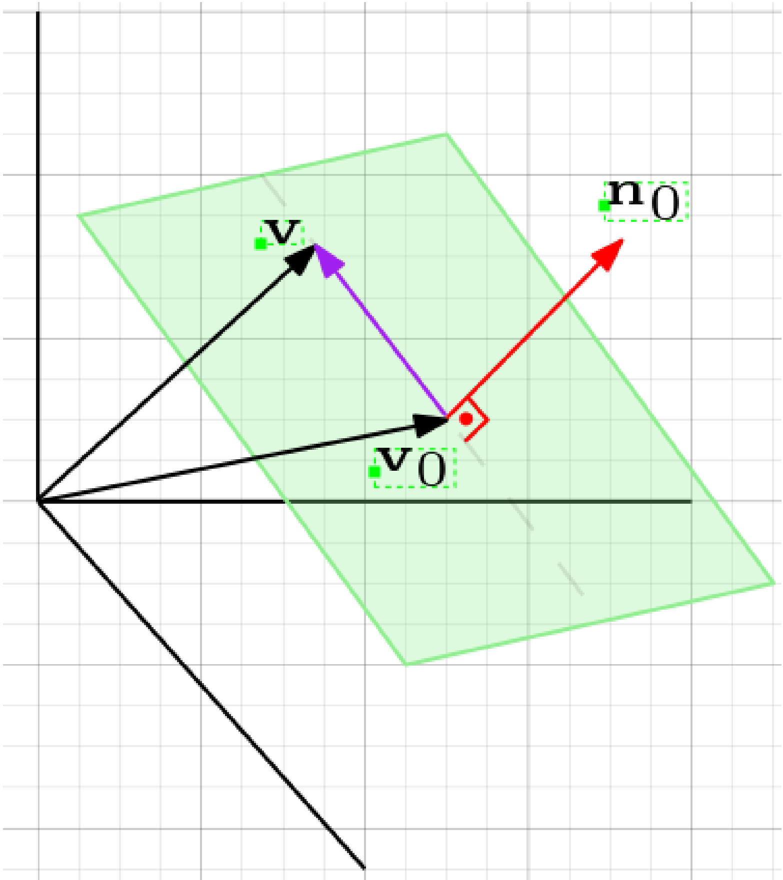
- Düzlem Denklemi (İpucu: İki vektör dikse iç çarpım 0’dır):



Uzayı İkiye Böler !!!

Sayılar, Vektörler ve Matrisler

- Düzlem Denklemi (İpucu: İki vektör dikse iç çarpım 0'dır):

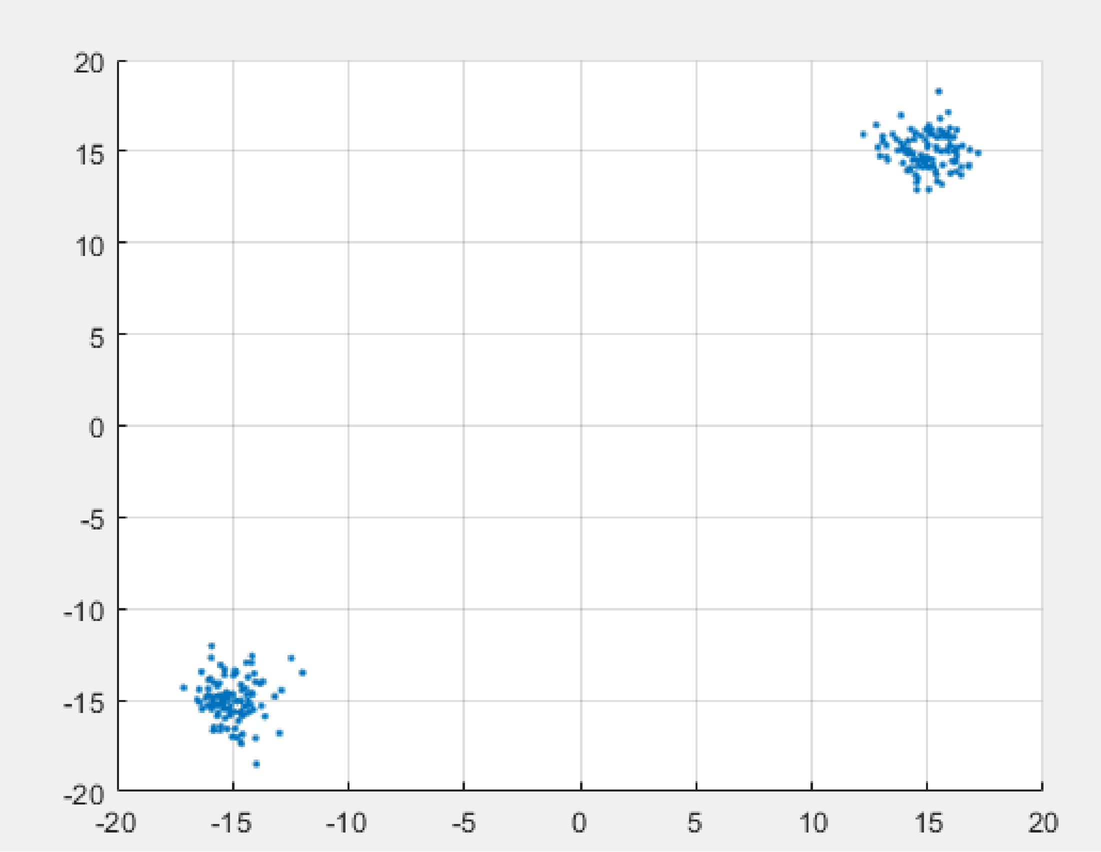


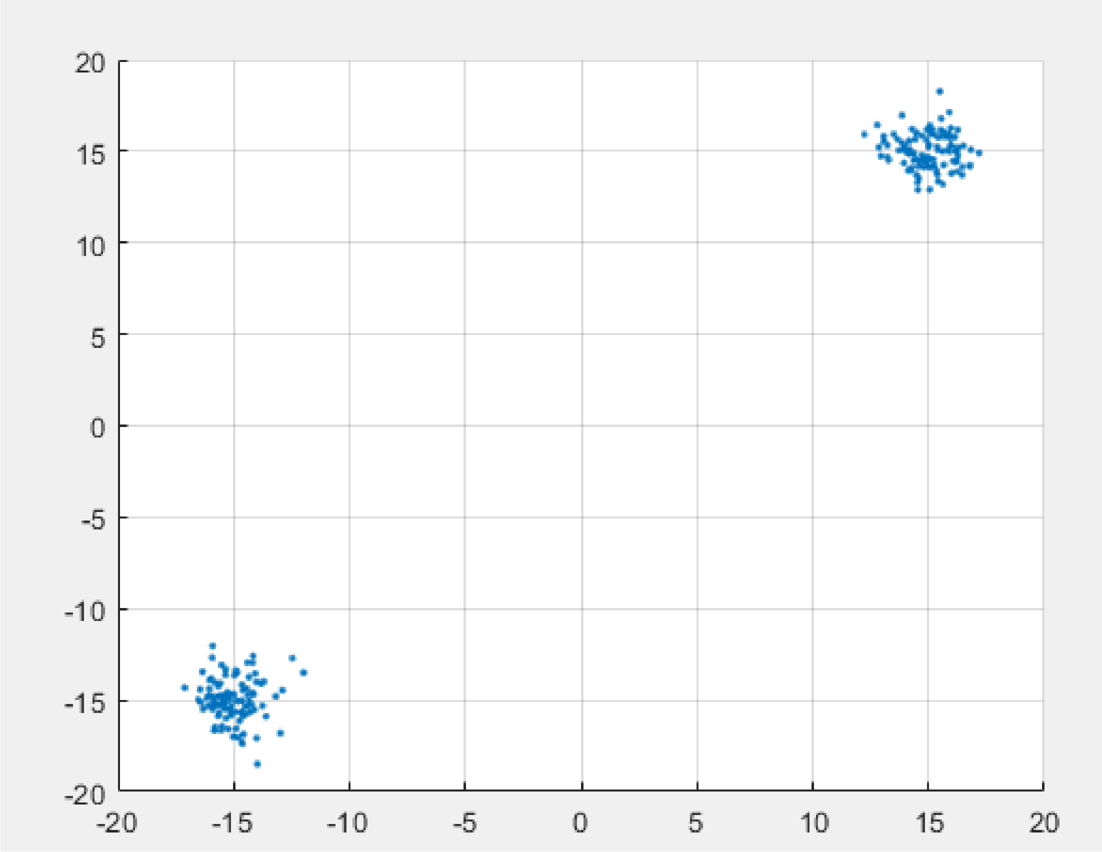
$$\langle \boldsymbol{n}_0, (\boldsymbol{v} - \boldsymbol{v_0}) \rangle = 0$$

Koordinat sistemini kaydırma

Sayılar, Vektörler ve Matrisler

$$\langle \boldsymbol{n}_0, (\boldsymbol{v} - \boldsymbol{v_0}) \rangle = 0$$





Sayılar, Vektörler ve Matrisler

$$\langle \boldsymbol{n}_0, (\boldsymbol{v} - \boldsymbol{v_0}) \rangle = 0$$

S1A) Aşağıdaki noktaların konumlarını belirleyiniz . (**P.15**)

$Düzlem: \; p(\boldsymbol{x}, \boldsymbol{n_0}, \boldsymbol{v_0}) = \boldsymbol{n_0^T}.(\boldsymbol{x_0} - \boldsymbol{v_0}) \quad ; \quad \boldsymbol{n_0} = \begin{bmatrix} 2 \\ -1 \end{bmatrix} \; \boldsymbol{v_0} = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \; Noktalar: \; \boldsymbol{x_0} = \begin{bmatrix} -1 \\ -2 \end{bmatrix} \quad \boldsymbol{x_1} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad \boldsymbol{x_3} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$

$Küre: \; b(\boldsymbol{x}, \boldsymbol{c}_0, r_0) = |\boldsymbol{x} - \boldsymbol{c_0}|^2 - r_0 \quad ; \quad \boldsymbol{c_0} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad r_0 = 7 \quad Noktalar: \; \boldsymbol{x_1} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad \boldsymbol{x_2} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$

Sayılar, Vektörler ve Matrisler

$$\langle \boldsymbol{n}_0, (\boldsymbol{v} - \boldsymbol{v_0}) \rangle = 0$$

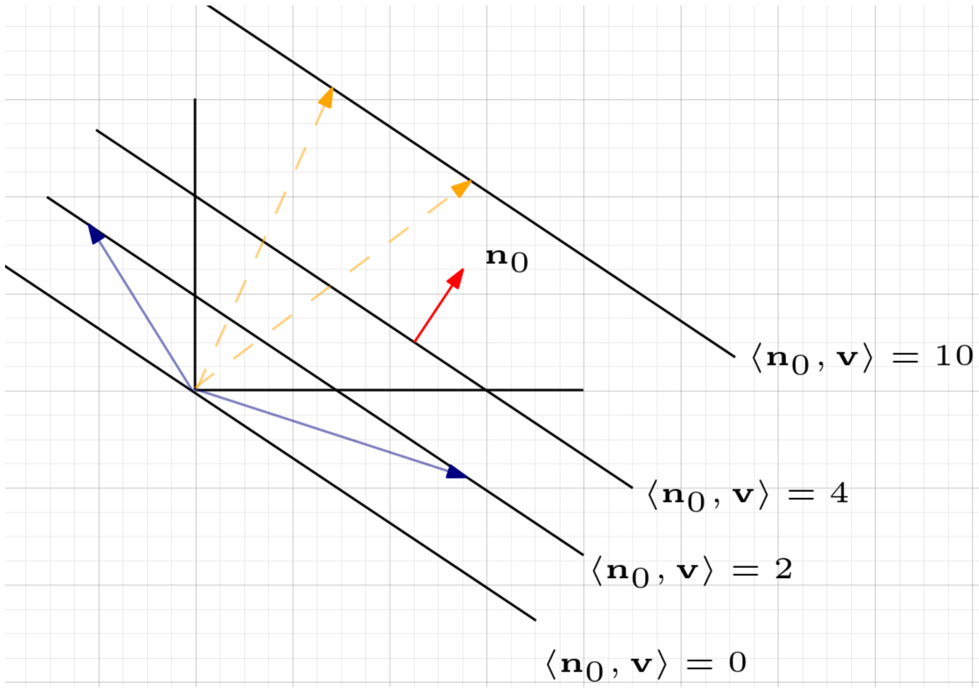
$$\boldsymbol{n_0} = \begin{bmatrix} 2 \\ -1 \end{bmatrix} \quad \boldsymbol{v_0} = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \quad Noktalar: \ \boldsymbol{x_0} = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$$

Sayılar, Vektörler ve Matrisler

$$\langle \boldsymbol{n}_0, (\boldsymbol{v} - \boldsymbol{v_0}) \rangle = 0$$

$$\boldsymbol{n_0^T}.\,(\boldsymbol{v} - \boldsymbol{v_0}) = 0$$

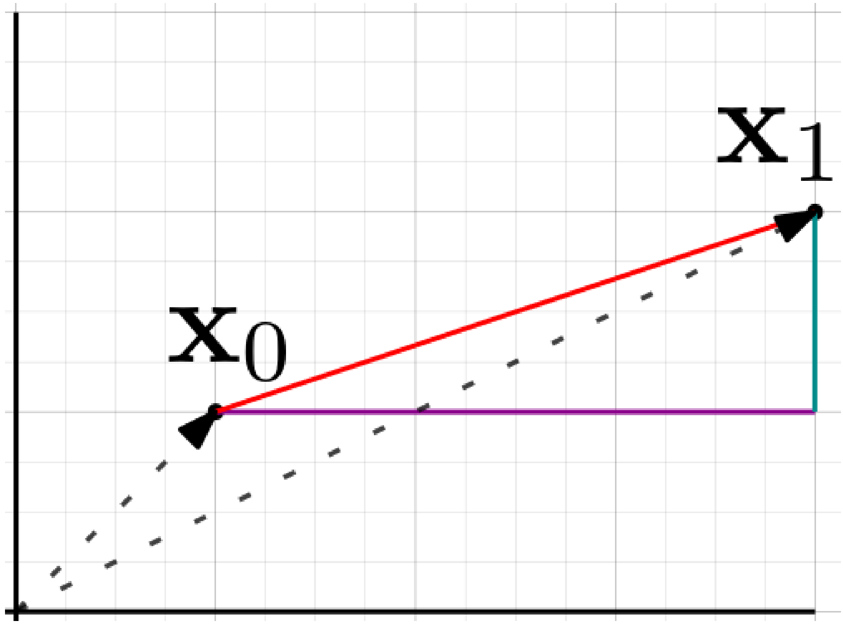
$$\langle \boldsymbol{n_0}\,, \boldsymbol{v} \rangle \ = \langle \boldsymbol{n_0}\,, \boldsymbol{v_0} \rangle$$



Eş değer eğrileri

Sayılar, Vektörler ve Matrisler

NORM KAVRAMI



Sayılar, Vektörler ve Matrisler

NORM KAVRAMI

$$\mathbf{x} \in \Re^{N\times 1} \text{ olsun}$$

$$\ell_1 \text{ -norm} = |\mathbf{x}|_1 \longrightarrow \sum_{i=1}^{N} |\mathbf{x}(i)|$$

$$\ell_2 \text{ -norm} = |\mathbf{x}|_2 \longrightarrow \sqrt{\sum_{i=1}^{N} |\mathbf{x}(i)|^2}$$

$$\ell_p \text{ -norm} = |\mathbf{x}|_p \longrightarrow (\sum_{i=1}^{N} |\mathbf{x}(i)|^p)^{\frac{1}{p}}$$

$$\ell_\infty \text{ -norm} = |\mathbf{x}|_\infty \longrightarrow max\{|\mathbf{x}(i)| : i \in 1, 2, 3, .., N\}$$

Sayılar, Vektörler ve Matrisler

NORM KAVRAMI

$$\mathbf{x} \in \Re^{N\times 1} \text{ olsun}$$

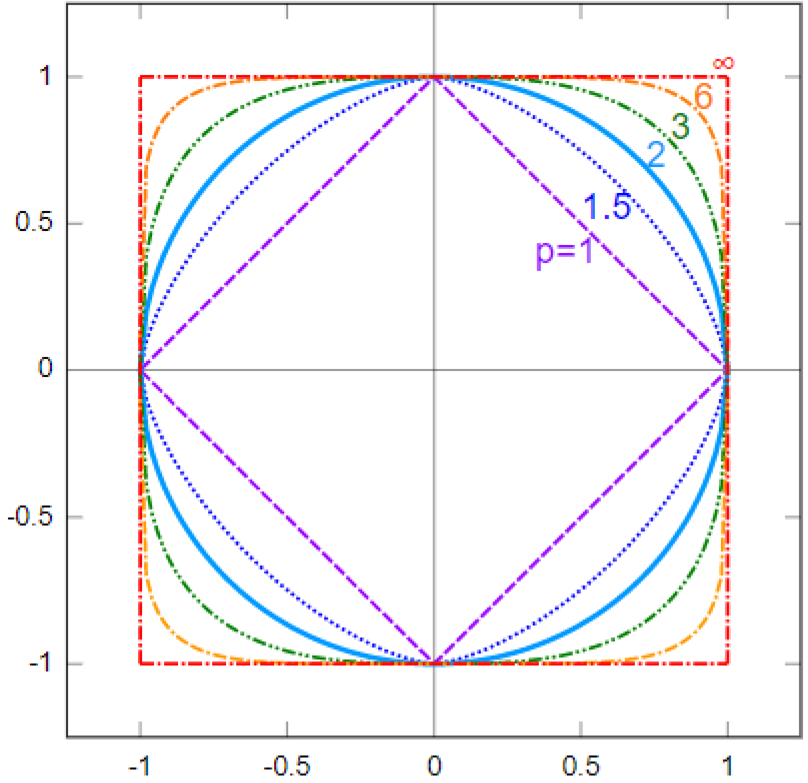
$$\ell_1 \text{ -norm } = |\mathbf{x}|_1 \longrightarrow \sum_{i=1}^{N} |\mathbf{x}(i)|$$

$$\ell_2 \text{ -norm } = |\mathbf{x}|_2 \longrightarrow \sqrt{\sum_{i=1}^{N} |\mathbf{x}(i)|^2}$$

$$\ell_p \text{ -norm } = |\mathbf{x}|_p \longrightarrow \left(\sum_{i=1}^{N} |\mathbf{x}(i)|^p\right)^{\frac{1}{p}}$$

$$\ell_\infty \text{ -norm } = |\mathbf{x}|_\infty \longrightarrow max\{|\mathbf{x}(i)| : i \in 1, 2, 3, .., N\}$$

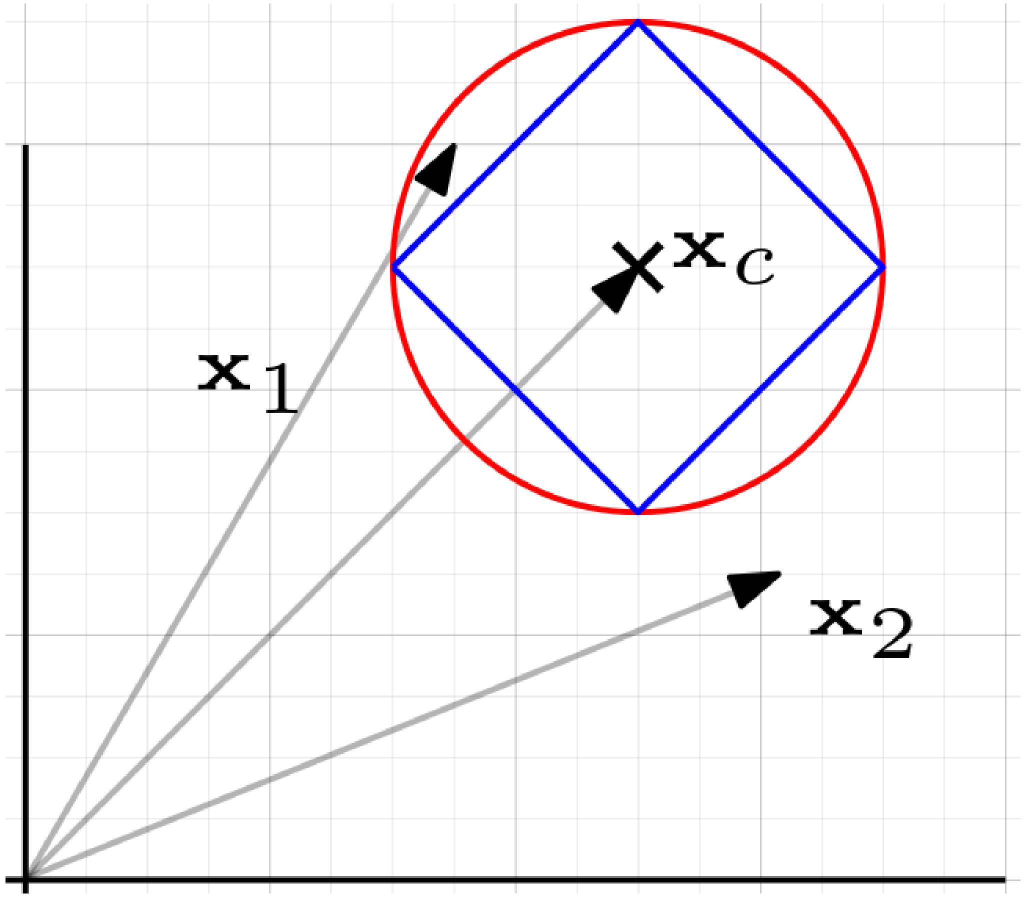
$$|\boldsymbol{x}|_p = 1$$



https://upload.wikimedia.org/wikipedia/commons/d/d4/Vector-p-Norms_qtl1.svg

Sayılar, Vektörler ve Matrisler

$NORM\ KAVRAMI$



$$\boldsymbol{x} \in \Re^{2\times 1} \quad |\boldsymbol{x} - \boldsymbol{x_c}|_p = 1$$

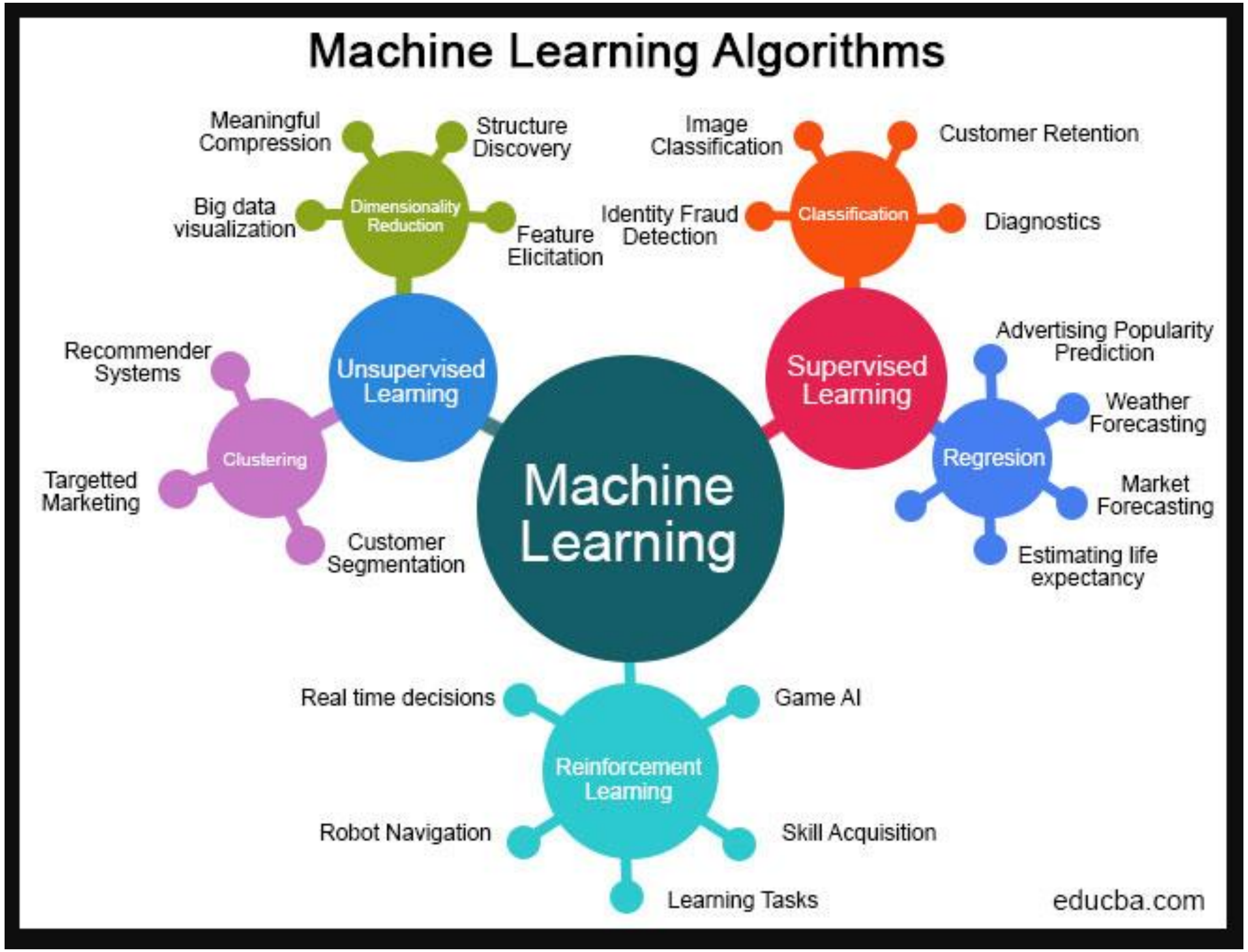
$Verinin\ etrafını\ sarma$

Sayılar, Vektörler ve Matrisler

$\text{Ö}ZET$

Veri Nedir?

Denetimli Öğrenme



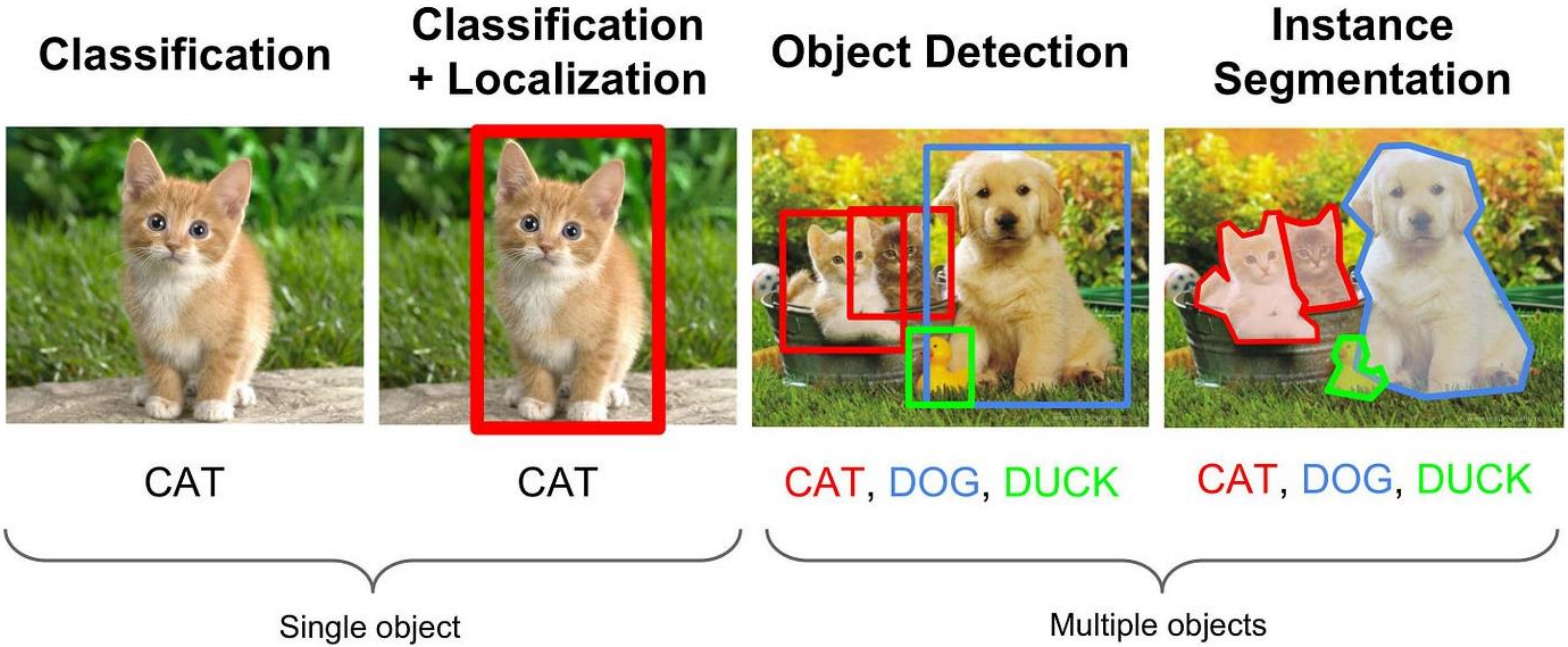
Denetimli Öğrenme

- Denetimli Öğrenme

- Veri

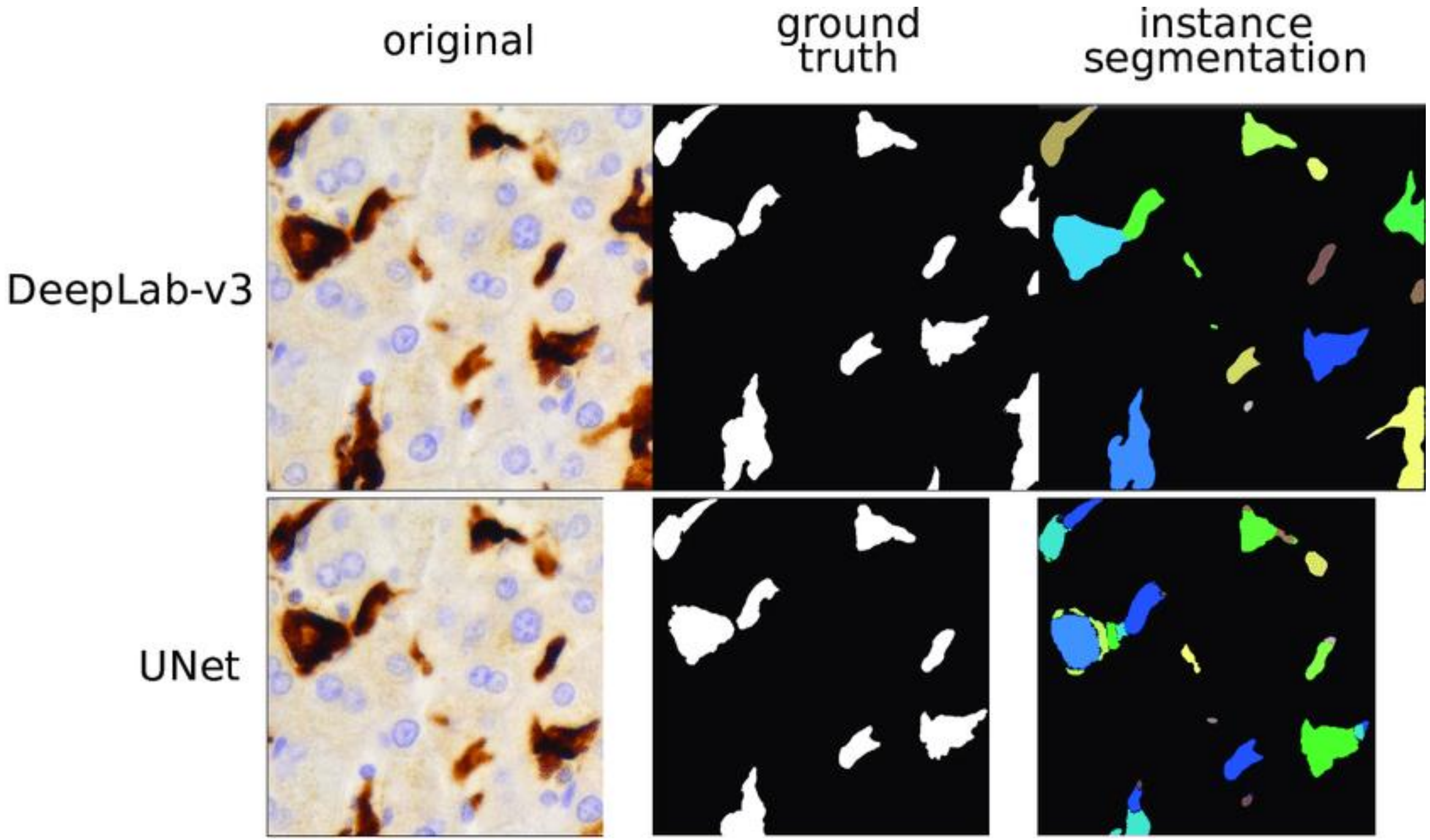
- Veri seti

- Eğitim verisi
- Test verisi



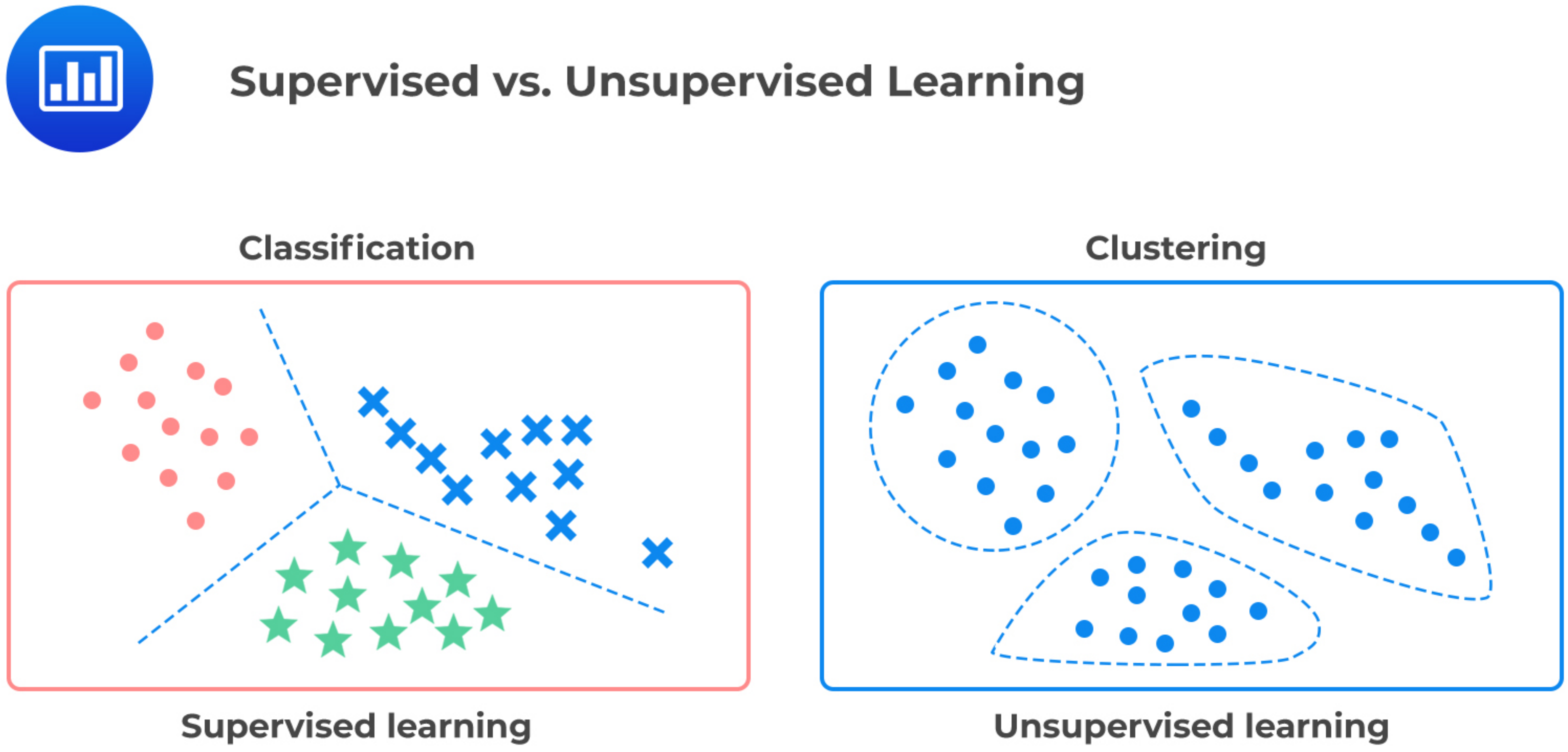
https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852

Denetimli Öğrenme



https://www.researchgate.net/figure/TAM-segmentation-instance-approach-This-figure-compares-the-histopathological-image-and_fig2_352915991

Denetimli Öğrenme (Tersi ??)



https://analystprep.com/study-notes/cfa-level-2/quantitative-method/supervised-machine-learning-unsupervised-machine-learning-deep-learning/

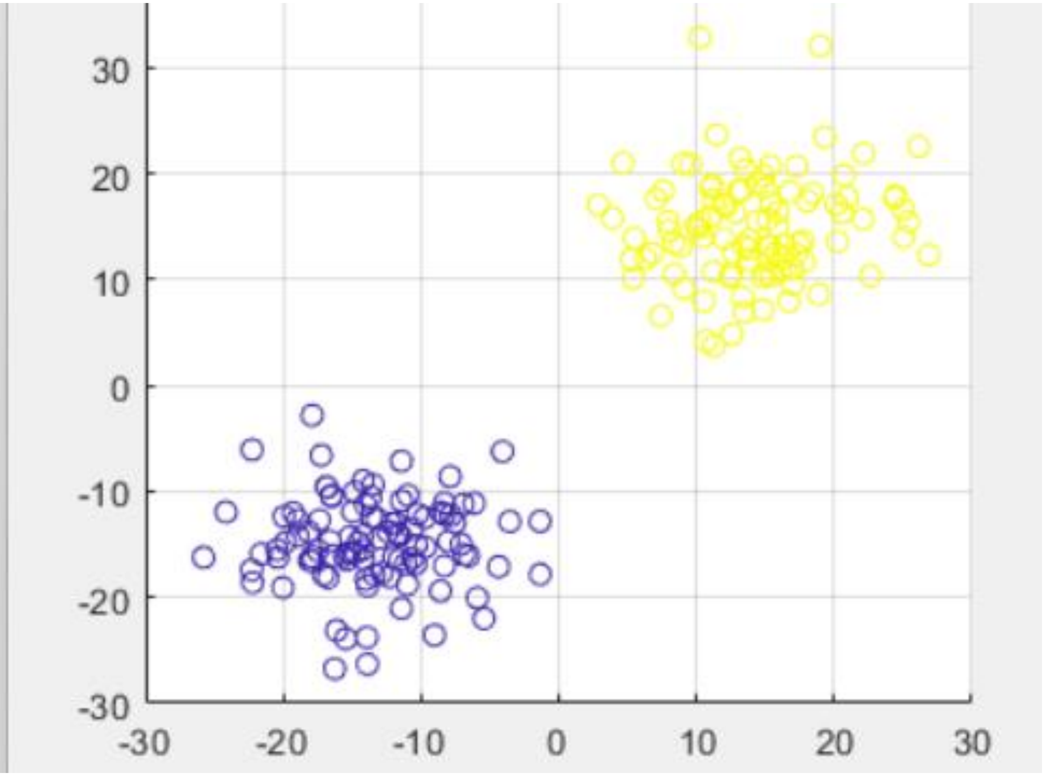
```
clear all, close all; clc


x = -15 + 5*randn(100,2);
y =  15 + 5*randn(100,2);

z = [x ; y];

c = [ones(1,100) , 2*ones(1,100)];

figure
scatter(z(:,1) , z(:,2) ,[] ,c),grid on
```



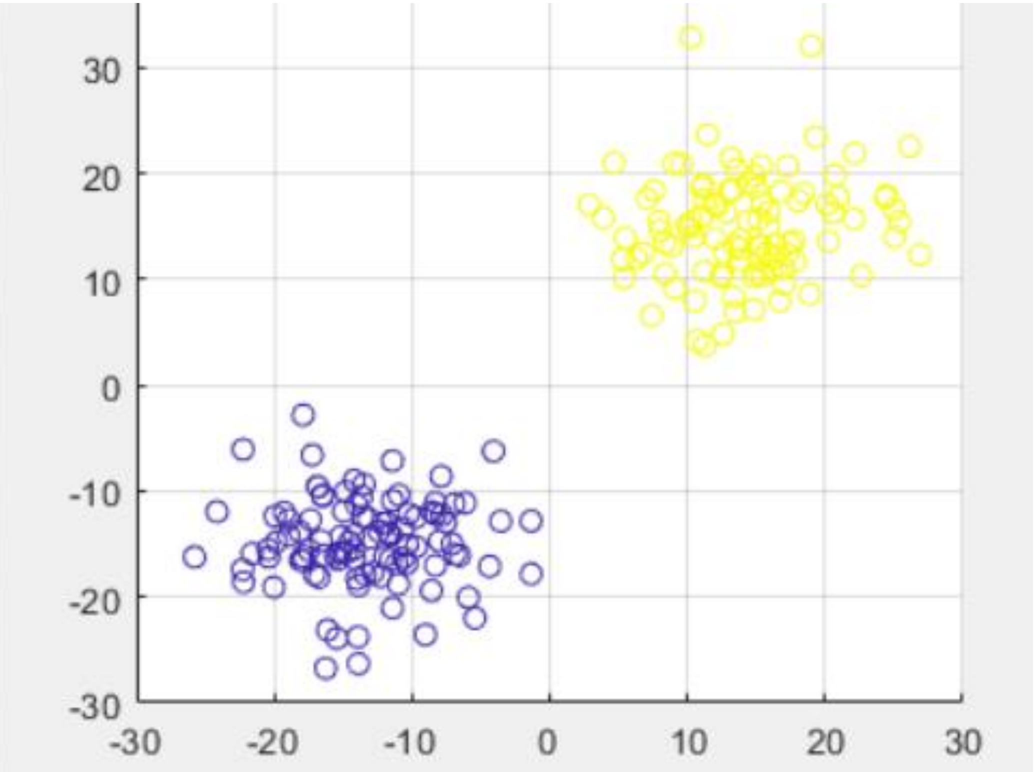
```
clear all, close all; clc


x = -15 + 5*randn(100,2);
y =  15 + 5*randn(100,2);

z = [x ; y];

c = [ones(1,100) , 2*ones(1,100)];

figure
scatter(z(:,1) , z(:,2) ,[] ,c),grid on
```

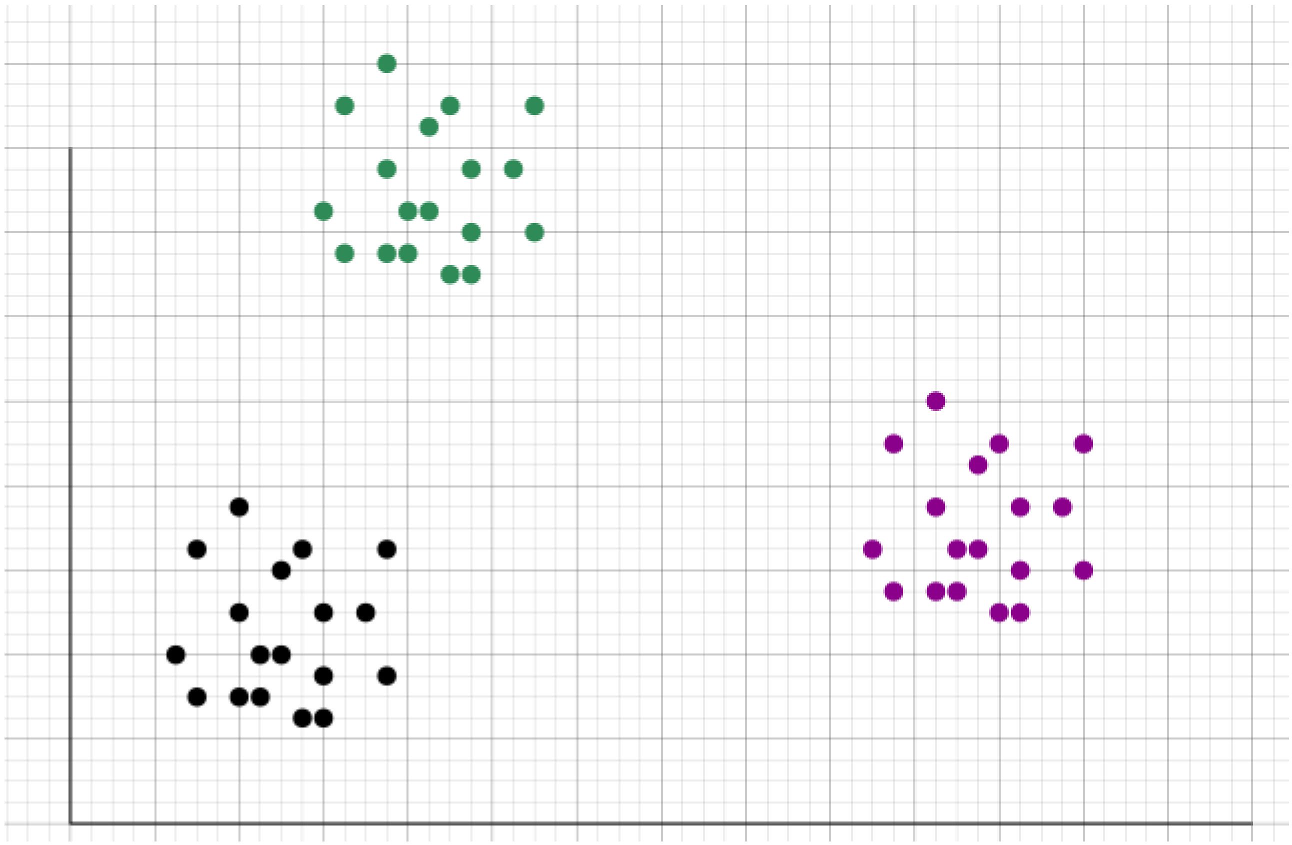


z

200x2 double

	1	2
1	-8.2906	-17.0246
2	-19.9422	-12.3607
3	-5.9103	-20.0348
4	-16.8722	-9.5549
5	-22.2587	-6.0756
6	-18.0934	-16.5188
7	-10.3275	-15.0435
8	-9.7204	-12.4668
9	-14.1989	-8.9837
10	-13.5630	-12.3899
11	-11.8355	-13.0148
12	-22.2952	-17.4141
13	-17.9085	-16.1575
14	-24.1507	-11.9331
15	-17.2455	-6.5857
16	-10.2536	-12.1580

K-En Yakın Komşu Sınıflandırma (k-NN)



K-En Yakın Komşu Sınıflandırma (k-NN)

K-En Yakın Komşu Sınıflandırma (k-NN)

```
X = [1,2        L = [1        v = [6,5];
     4,4             2
     7,2             3
     8,6             3
     11,5];          4];
```

$\color{red}{Metrik}: \ell_2 - Norm$

K-En Yakın Komşu Sınıflandırma (k-NN)

```
X = [1,2
     4,4
     7,2
     8,6
     11,5];

v = [6,5];
```

```
D =
```

-5	-3
-2	-1
1	-3
2	1
5	0

25	9
4	1
1	9
4	1
25	0

34
5
10
5
25

5.8310
2.2361
3.1623
2.2361
5.0000

```
L = [1
     2
     3
     3
     4];
```

K-En Yakın Komşu Sınıflandırma (k-NN)

$$\color{red}{Metrik}: \ell_1 - Norm\ Olsaydı$$

```
D =

-5      -3        5      3        8        L = [1
-2      -1        2      1        3              2
 1      -3        1      3        4              3
 2       1        2      1        3              3
 5       0        5      0        5              4];
```

K-En Yakın Komşu Sınıflandırma (k-NN)

- Euclidian Distance (ℓ_2)
- City Block (Manathan) (ℓ_1)
- Minkowski Dist.
- Chebyshev (ℓ_∞)

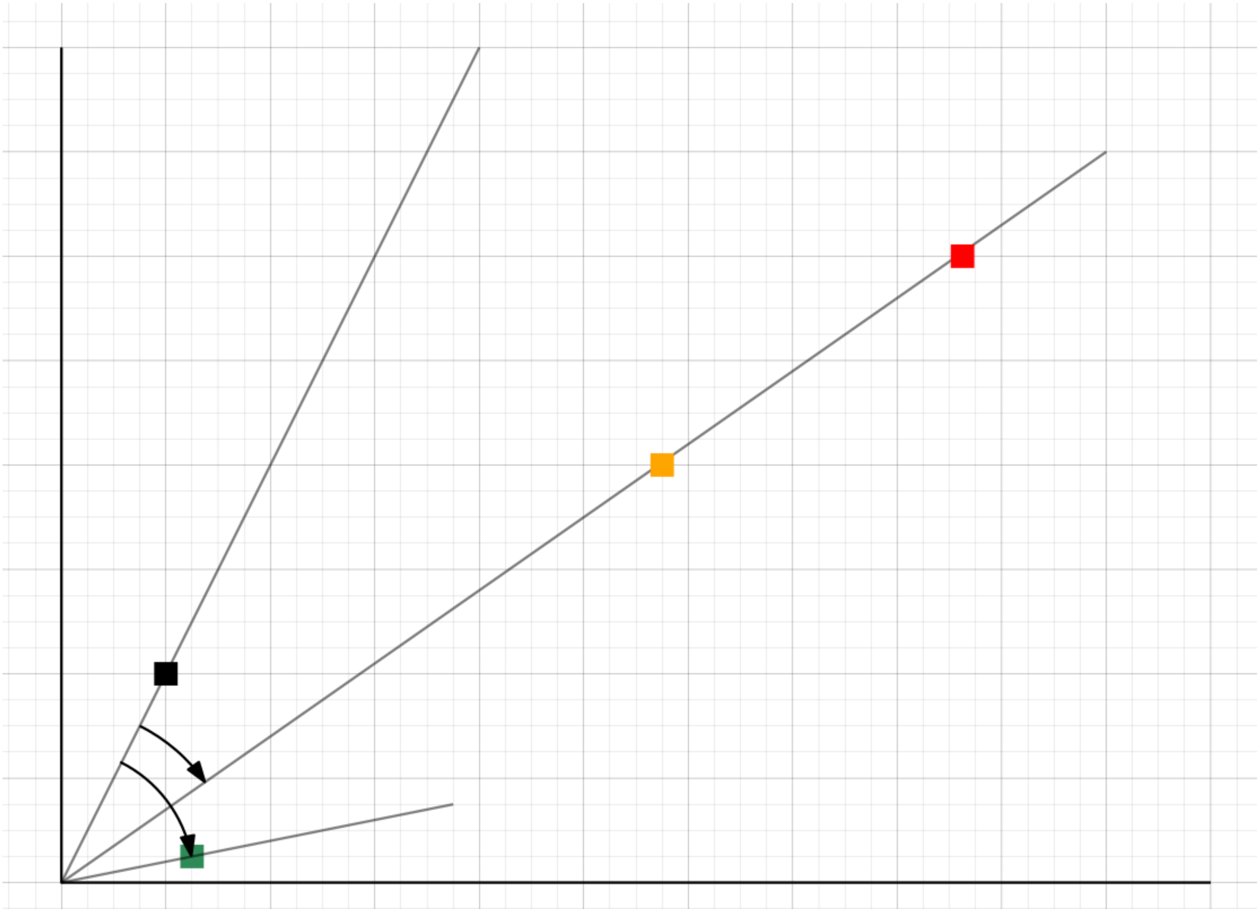
$$\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$$

K-En Yakın Komşu Sınıflandırma (k-NN)

- Euclidian Distance (ℓ_2)
- City Block (Manathan) (ℓ_1)
- Minkowski Dist.
- Chebyshev (ℓ_∞)
- Cosine Similarity

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

K-En Yakın Komşu Sınıflandırma (k-NN)



$$\frac{\langle \boldsymbol{x_1}, \boldsymbol{x_2} \rangle}{|\boldsymbol{x_1}||\boldsymbol{x_2}|}$$

$$\mathbf{u.v} = u_1v_1 + u_2v_2 + u_3v_3$$

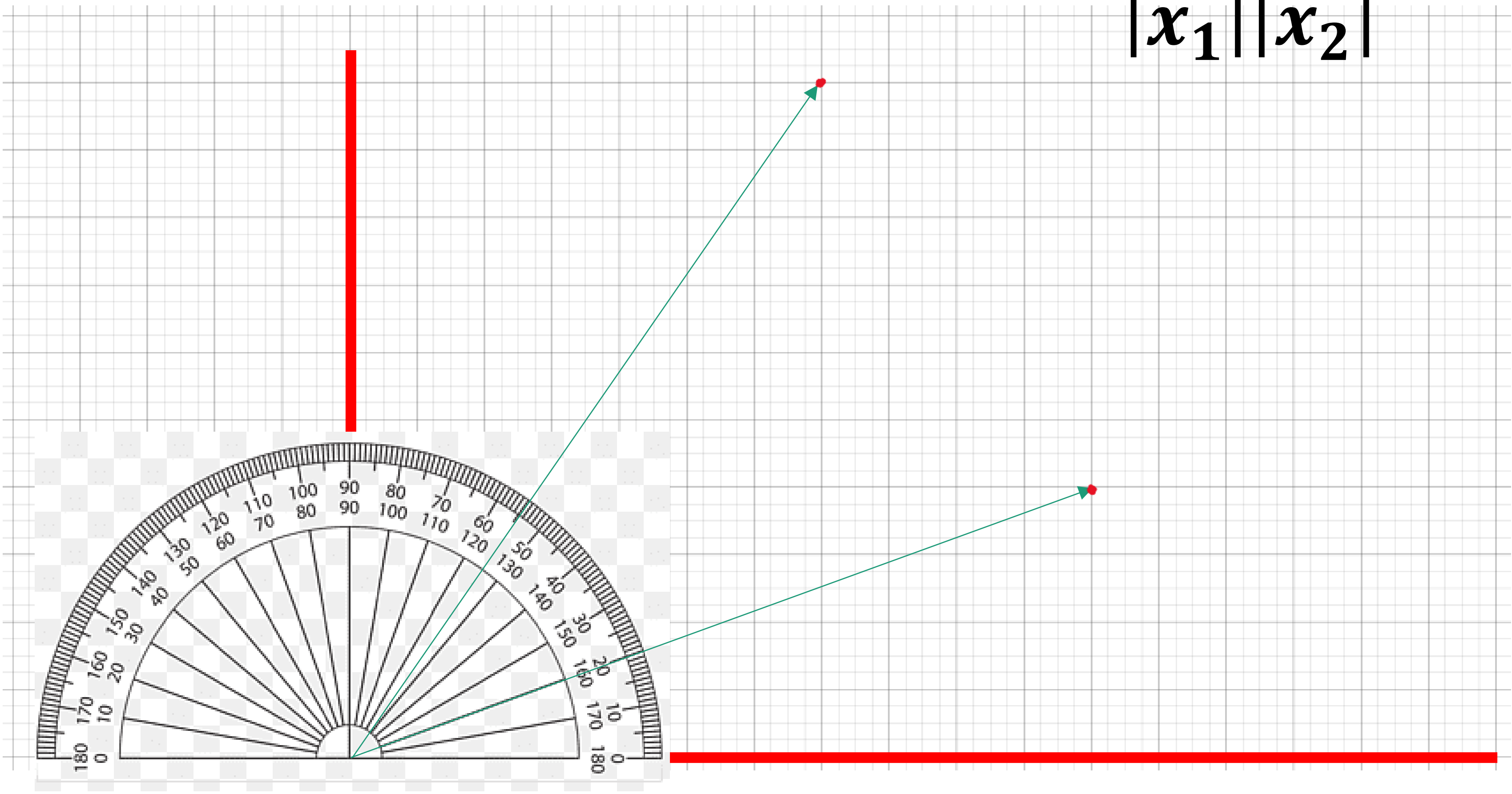
$$\mathbf{u.v} = ||\mathbf{u}||\,||\mathbf{v}|| \cos\theta$$

K-En Yakın Komşu Sınıflandırma (k-NN)

$$\frac{\langle \boldsymbol{x_1}, \boldsymbol{x_2} \rangle}{|\boldsymbol{x_1}||\boldsymbol{x_2}|}$$

$$\mathbf{u.v} = u_1v_1 + u_2v_2 + u_3v_3$$

$$\mathbf{u.v} = ||\mathbf{u}||\,||\mathbf{v}|| \cos\theta$$



K-En Yakın Komşu Sınıflandırma (k-NN)

- Başka hangi mesafe metrikleri olabilir?

K-En Yakın Komşu Sınıflandırma (k-NN)

Algoritma adımları

- Etiketli bir veri seti
- Test örneği
- T-V her örnekle mesafeler hesaplanacak
- Sort (en düşük mesafe , indis)
- K tane en düşük örnek arasında baskın olan sınıfı

- PYTHON

Parameters::

n_neighbors : *int, default=5*

Number of neighbors to use by default for `kneighbors` queries.

weights : *{'uniform', 'distance'} or callable, default='uniform'*

Weight function used in prediction. Possible values:

- 'uniform' : uniform weights. All points in each neighborhood are weighted equally.
- 'distance' : weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.
- [callable] : a user-defined function which accepts an array of distances, and returns an array of the same shape containing the weights.

algorithm : *{'auto', 'ball_tree', 'kd_tree', 'brute'}, default='auto'*

Algorithm used to compute the nearest neighbors:

- 'ball_tree' will use `BallTree`
- 'kd_tree' will use `KDTree`
- 'brute' will use a brute-force search.
- 'auto' will attempt to decide the most appropriate algorithm based on the values passed to `fit` method.

Note: fitting on sparse input will override the setting of this parameter, using brute force.

leaf_size : *int, default=30*

Leaf size passed to BallTree or KDTree. This can affect the speed of the construction and query, as well as the memory required to store the tree. The optimal value depends on the nature of the problem.

p : *int, default=2*

Power parameter for the Minkowski metric. When p = 1, this is equivalent to using manhattan_distance (l1), and euclidean_distance (l2) for p = 2. For arbitrary p, minkowski_distance (l_p) is used.

metric : *str or callable, default='minkowski'*

Metric to use for distance computation. Default is "minkowski", which results in the standard Euclidean distance when p = 2. See the documentation of scipy.spatial.distance and the metrics listed in `distance_metrics` for valid metric values.

If metric is "precomputed", X is assumed to be a distance matrix and must be square during fit. X may be a sparse graph, in which case only "nonzero" elements may be considered neighbors.

If metric is a callable function, it takes two arrays representing 1D vectors as inputs and must return one value indicating the distance between those vectors. This works for Scipy's metrics, but is less efficient than passing the metric name as a string.

sklearn.metrics.pairwise.distance_metrics

```
sklearn.metrics.pairwise.distance_metrics()
```
[source]

Valid metrics for pairwise_distances.

This function simply returns the valid pairwise distance metrics. It exists to allow for a description of the mapping for each of the valid strings.

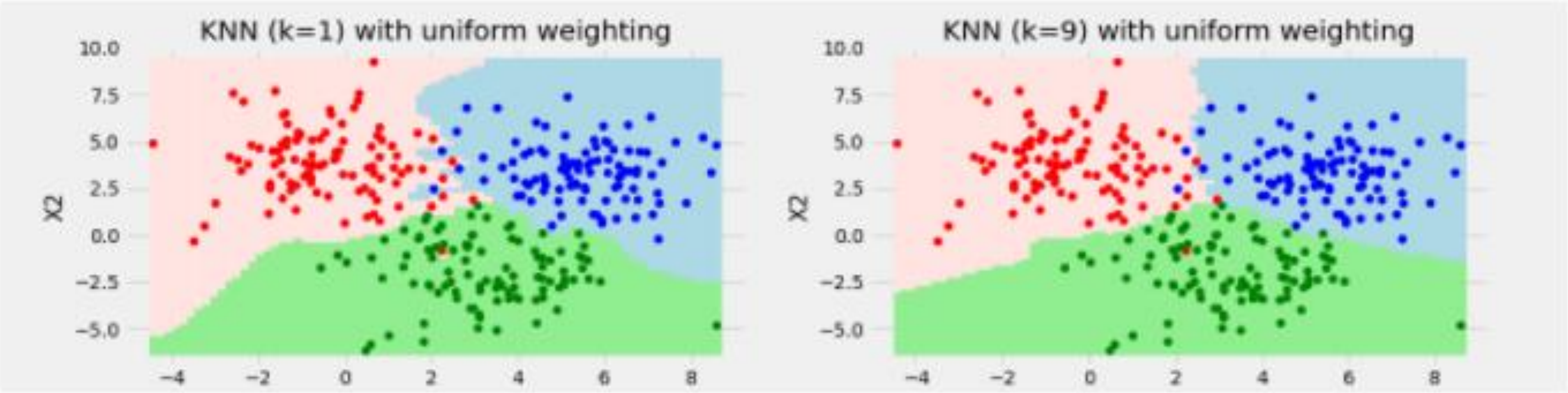
The valid distance metrics, and the function they map to, are:

metric	Function
'cityblock'	metrics.pairwise.manhattan_distances
'cosine'	metrics.pairwise.cosine_distances
'euclidean'	metrics.pairwise.euclidean_distances
'haversine'	metrics.pairwise.haversine_distances
'l1'	metrics.pairwise.manhattan_distances
'l2'	metrics.pairwise.euclidean_distances
'manhattan'	metrics.pairwise.manhattan_distances
'nan_euclidean'	metrics.pairwise.nan_euclidean_distances

K-NN algoritması

- Avantajları
  - Gerçekleme kolaylığı
  - Herhangi bir ön kabule ihtiyacı yoktur.
  - Eğitim yok
  - Yeni örnekler geldiğinde hızlı adaptasyon sağlar
  - Hem sınıflandırma hem de regresyon için kullanılır.
  - Birkaç parametre (k ve norm)
  - Doğrusal olmayan veriler sınıflandırılabilir
- Dezavantajları
  - Yavaş bir algoritmadır (büyük veri)
  - Homojen öznitelikler olması gerekir
  - Aykırı örneklere takılabilir.
  - K sayısının tespiti
  - RAM ihtiyacı

K-NN algoritması



K-NN (k=1) vs. K-NN (k=9) Classifiers (all images are generated by the author)

https://towardsdatascience.com/k-nearest-neighbors-k-nn-explained-8959f97a8632

K-NN algoritması