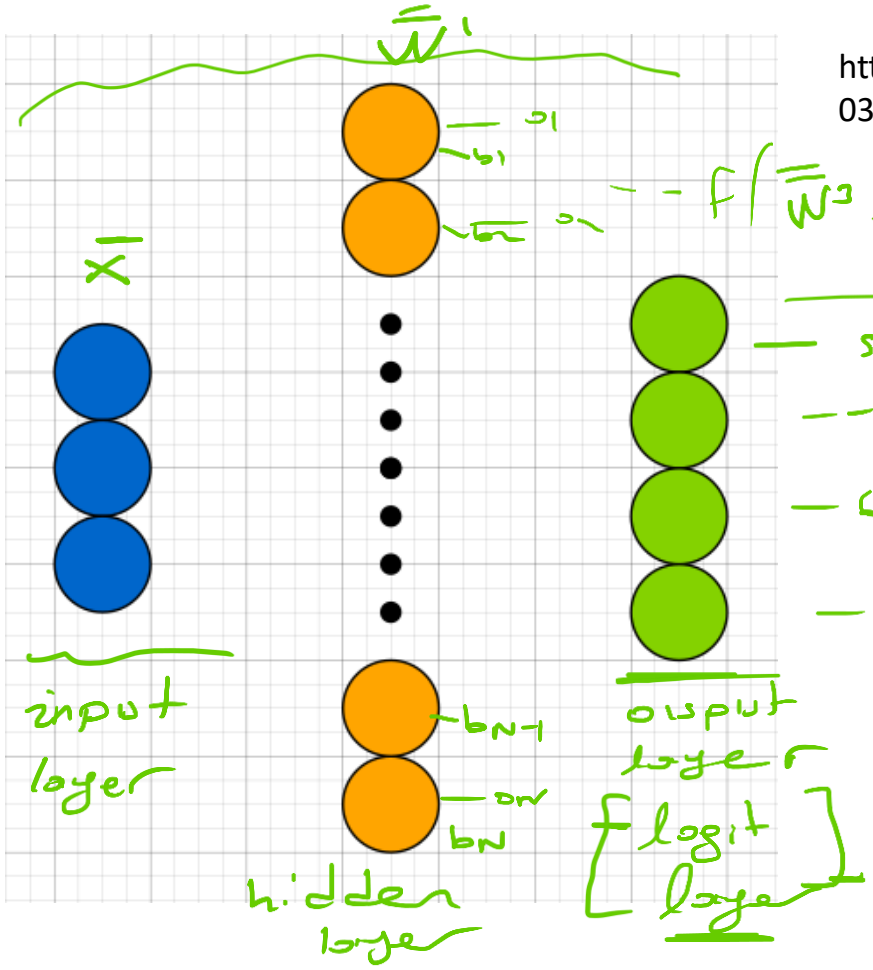
BİL 475 Örüntü Tanıma

Hafta-10:

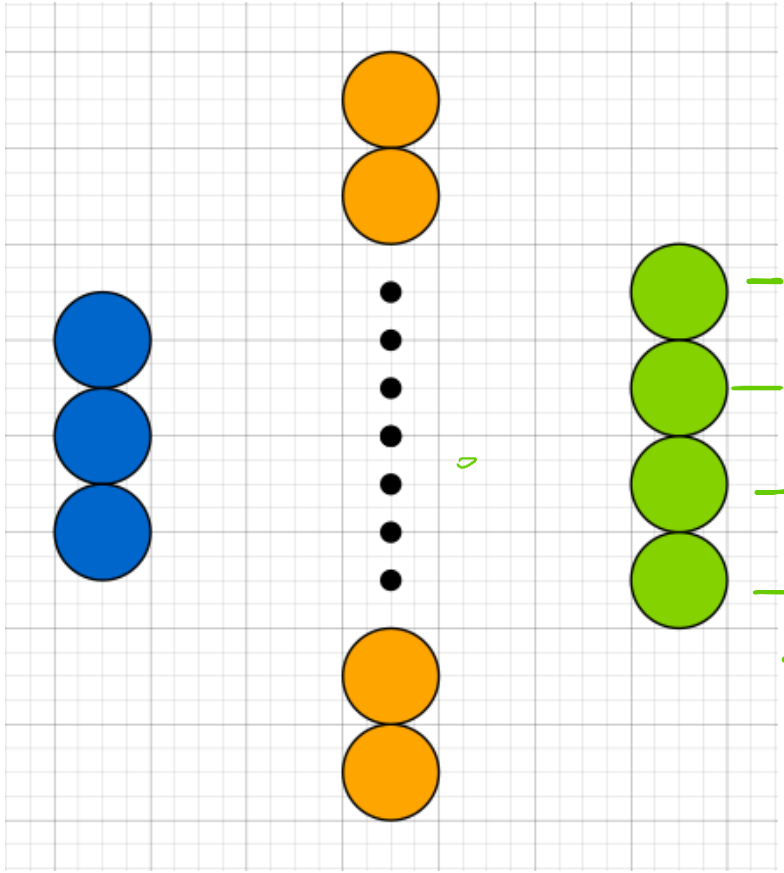
Destek Vektör Makinesi

Multi Layer Perceptron (MLP)



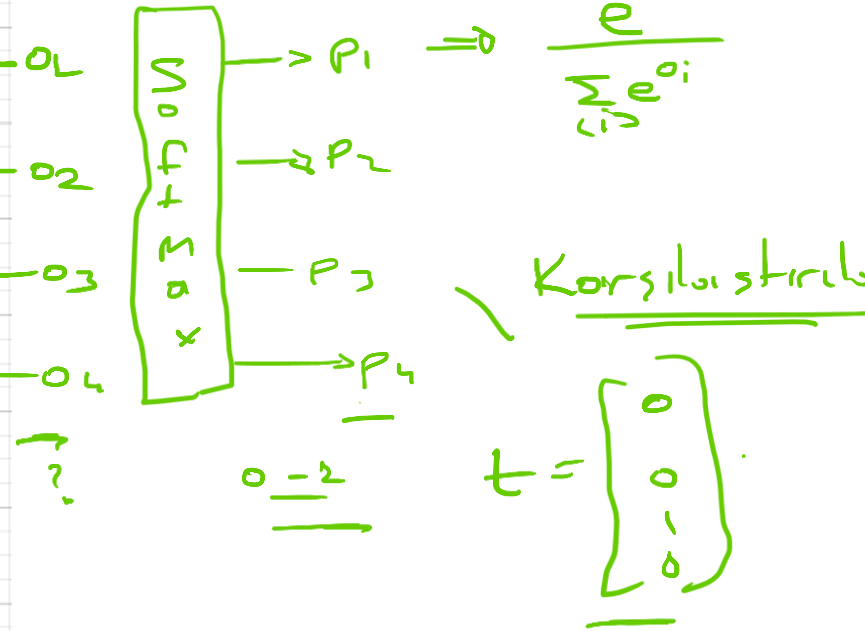
http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

Multi Layer Perceptron (MLP)

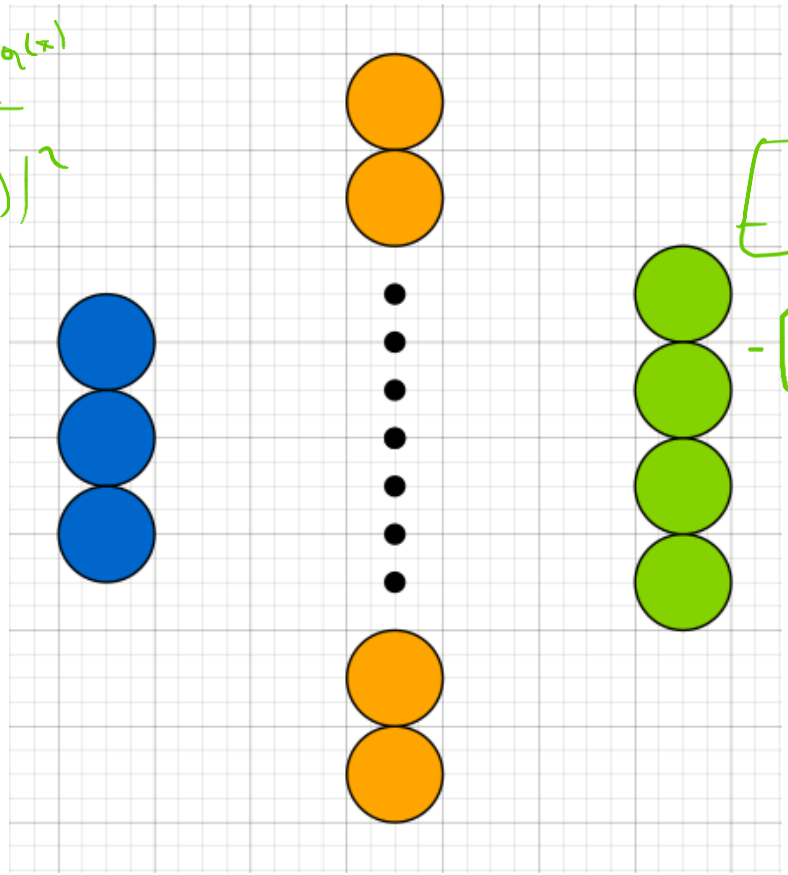


Tahmin – Hedef İlişkisi (Çapraz Entropi)

Softmax akt. fnk.



Multi Layer Perceptron (MLP)



Tahmin – Hedef İlişkisi (Çapraz Entropi)

Kullback-Leibler Divergance (KL-Div)

$$D(p||q) = \sum p_i \log\left(\frac{p_i}{q_i}\right)$$

- p → softmax üzerinden gele
- t → hedef vektörün

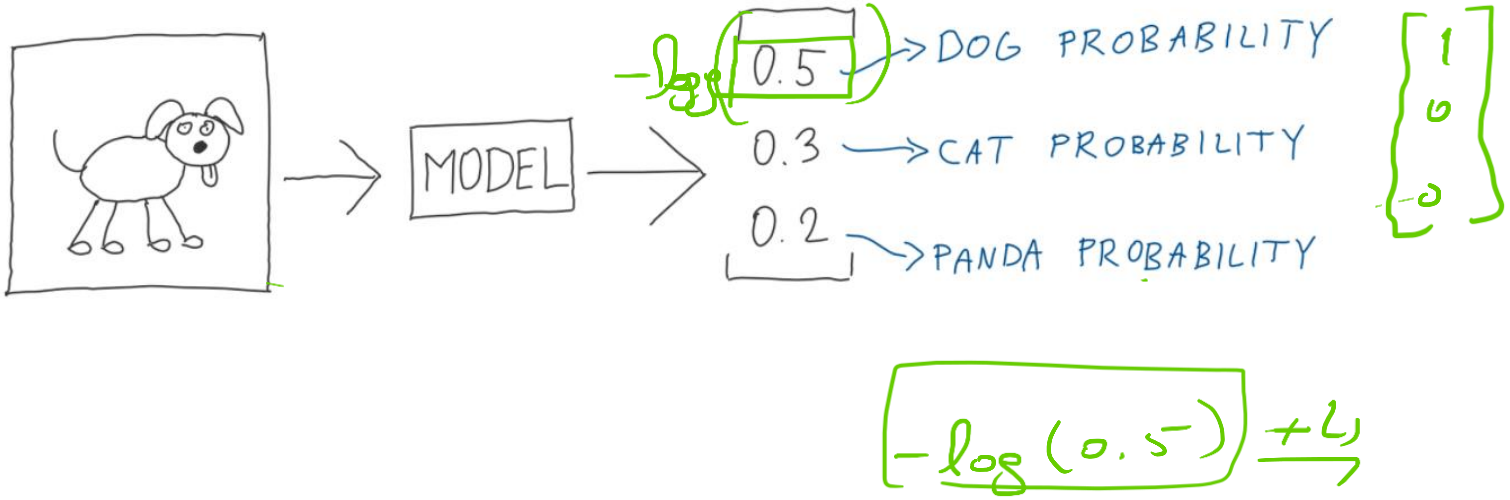
$$D(t||p) = \sum t_i \log\left(\frac{t_i}{p_i}\right)$$

$[0 \quad 0 \quad 1 \quad 0]$

$[p_1 \quad p_2 \quad p_3 \quad p_4]$ > $\pm \log\left(\frac{1}{p_3}\right)$

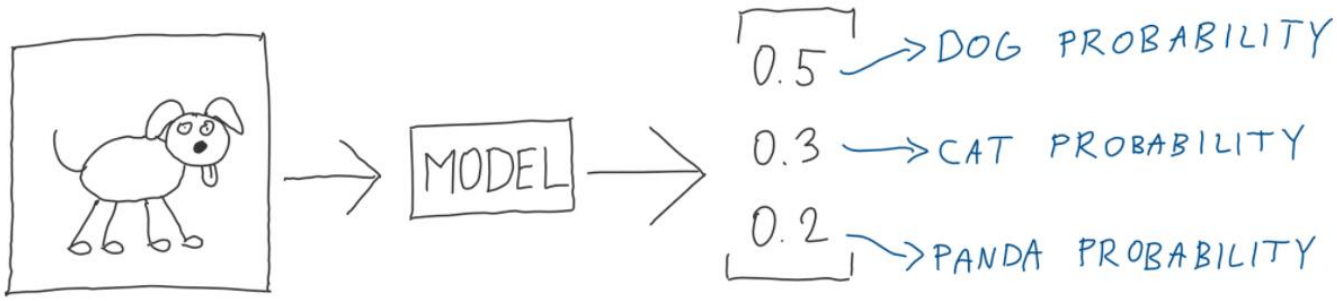
$\log(p_3^{-1}) = -\log(p$

Multi Layer Perceptron (MLP)



https://towardsdatascience.com/cross-entropy-for-classification-d98e7f974451

Multi Layer Perceptron (MLP)

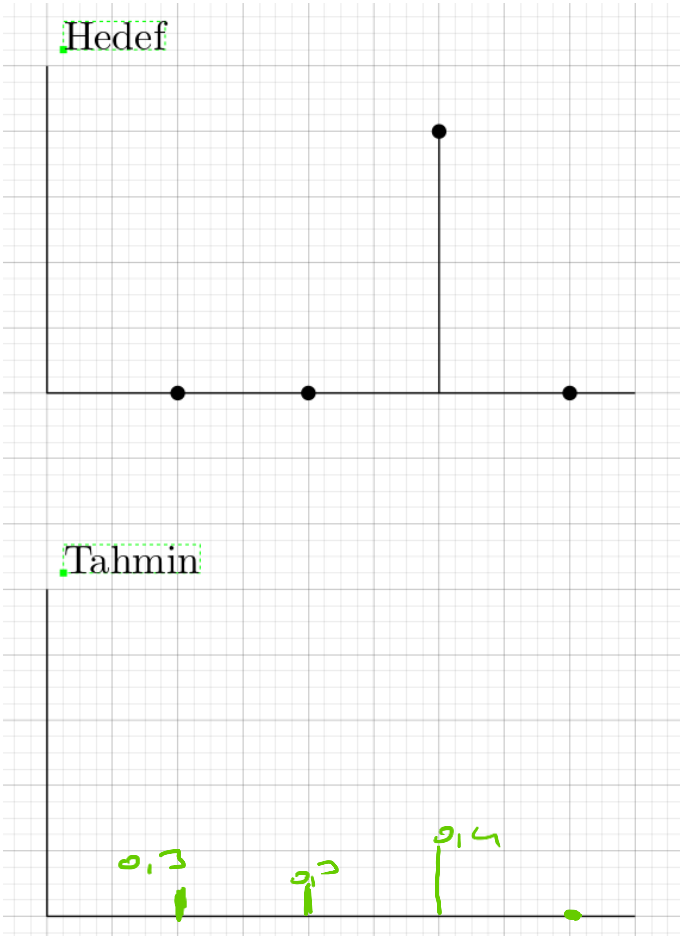


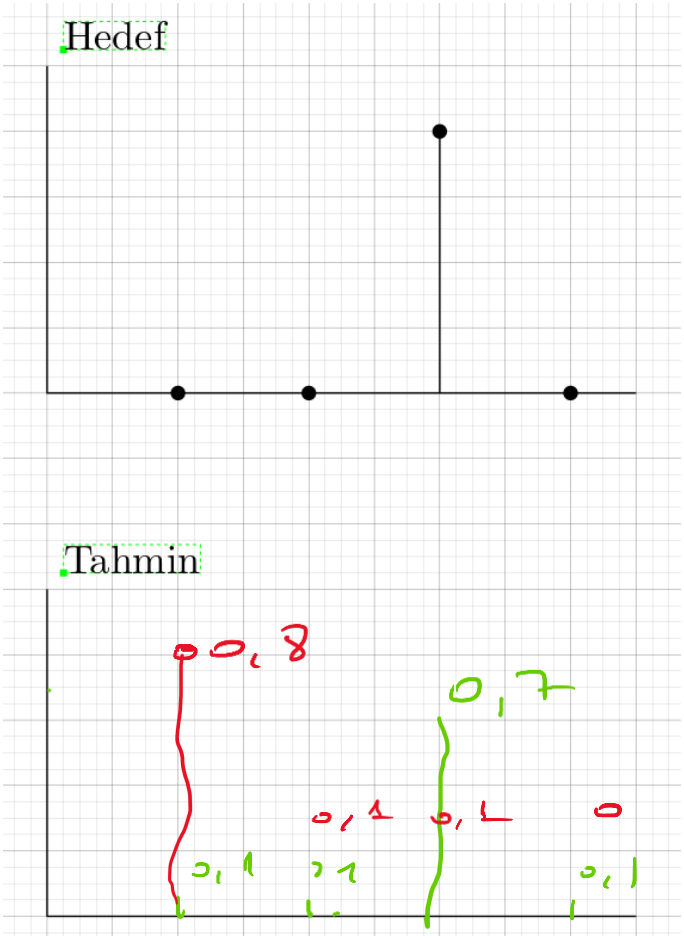
TARGET	PREDICTION
1	0.5
0	0.3
0	0.2

$$\text{Loss For class } X = -\underbrace{p(X)}_{\substack{\text{probability} \\ \text{of class X} \\ \text{in TARGET}}} \cdot \log \underbrace{q(X)}_{\substack{\text{probability} \\ \text{of class X} \\ \text{in PREDICTION}}}$$

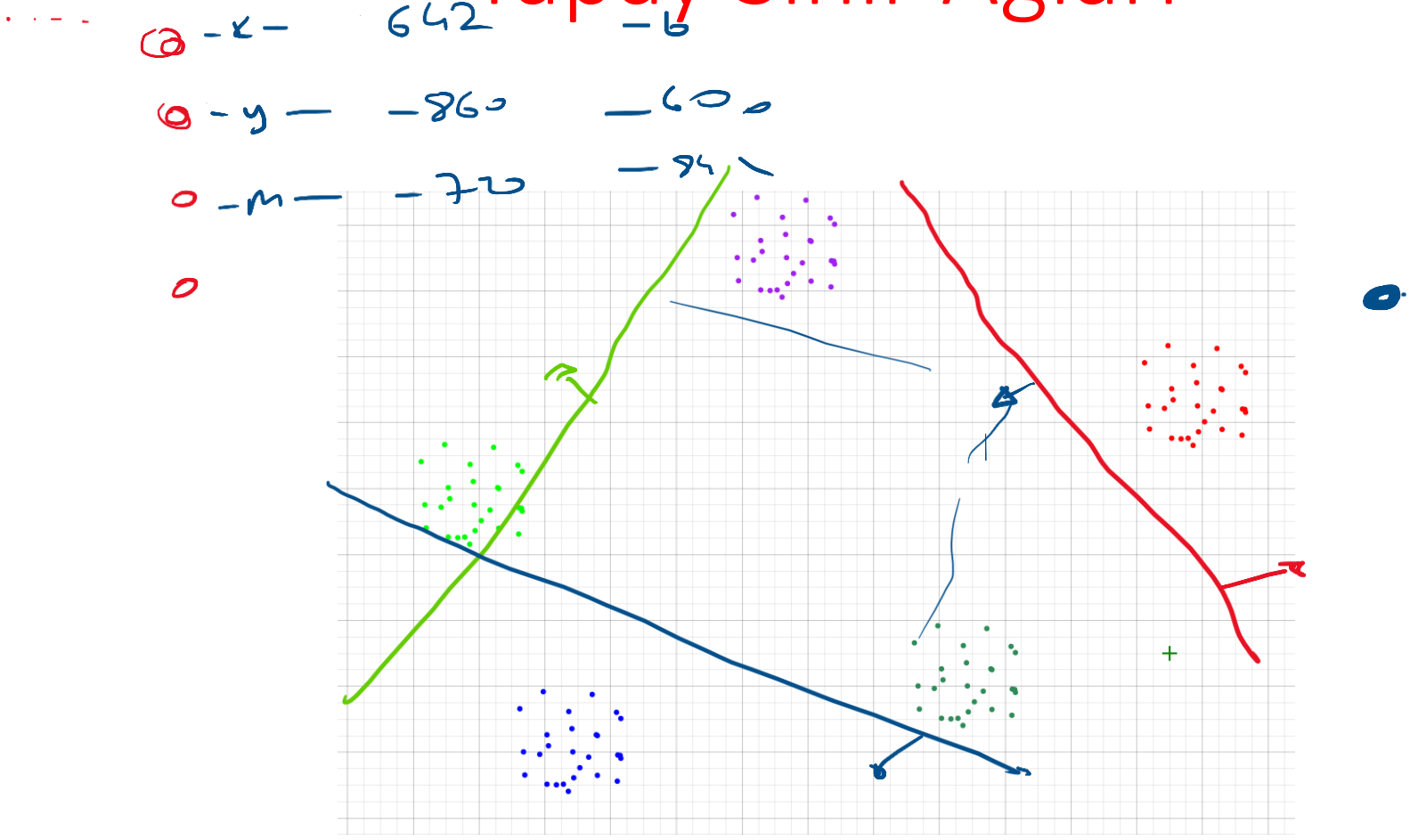
https://towardsdatascience.com/cross-entropy-for-classification-d98e7f974451

Multi Layer Perceptron (MLP)

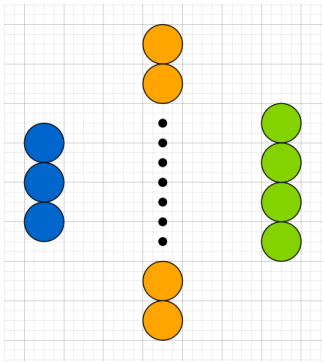




Yapay Sinir Ağları

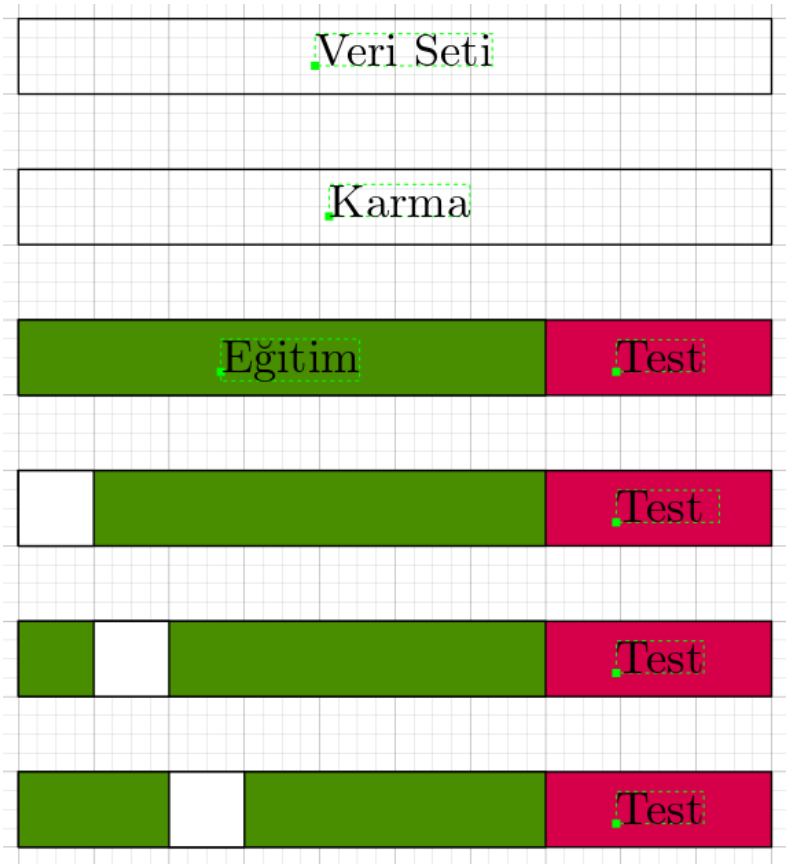


Multi Layer Perceptron (MLP)



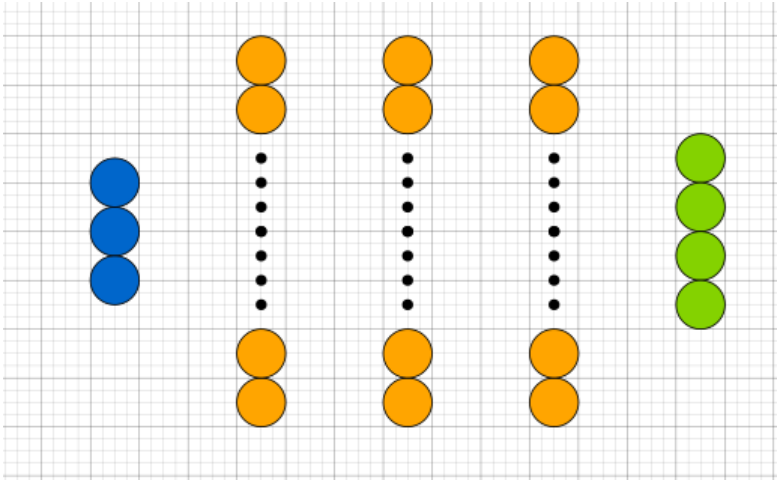
Üst parametre ayarlama
Kaç katman
Kaç nöron
vb.

k-Kat Çapraz Doğrulama (k-Fold Cross Validation)



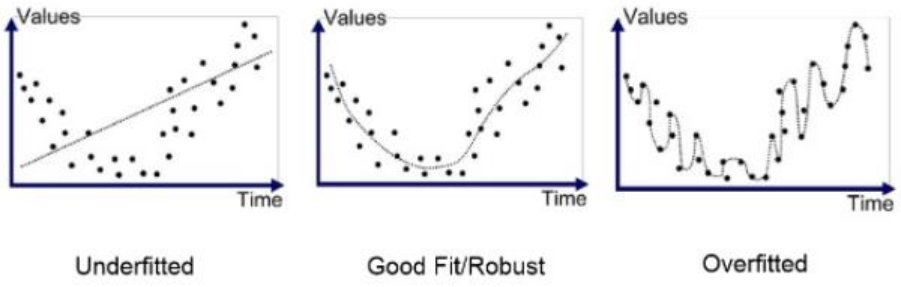
Multi Layer Perceptron (MLP)

Shallow vs Deep

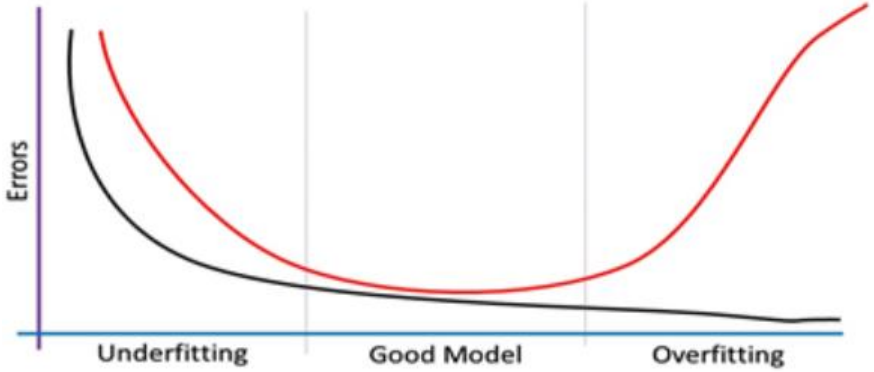


Multi Layer Perceptron (MLP)

Aşırı Öğrenme (Overfitting)



https://medium.com/data-science-tr/overfitting-underfitting-cross-validation-b47dfda0cf4e



https://medium.com/kodluyoruz/makine-%C3%B6%C4%9Frenmesinde-overfitting-underfitting-best-fitting-kavramlar%C4%B1-9f80a5d42719

Multi Layer Perceptron (MLP)

Aşırı Öğrenme (Overfitting)

→ *Daha fazla veri ile eğitim (Training with more data)*: Örnek sayımızı artırmak verimizdeki target ile feature arasında ki ilişkiyi daha rahat anlamamızı sağlamayabilmektedir. Daha fazla veriye sahip olan algoritmanın, daha fazla veri türünü dikkate alarak daha iyi genelleme olasılığı daha yüksektir.

Data augmentation

→ *Regularization*: Düzenleme, modelin daha iyi genelleşmesi için öğrenme algoritmasında küçük değişiklikler yapan bir tekniktir. Bu da modelin görünmeyen verilerdeki performansını artırır. Genellikle L1 ve L2 olmak üzere iki türü bulunmaktadır.

https://medium.com/kodluyoruz/makine-%C3%B6%C4%9Frenmesinde-overfitting-underfitting-best-fitting-kavramlar%C4%B1-9f80a5d42719

Multi Layer Perceptron (MLP)

Aşırı Öğrenme (Overfitting)

→ *Removing Features*: Feature setimizden alakasız feature'ları çıkartarak target-feature ilişkisini daha net bir hale getirebilmekteyiz.

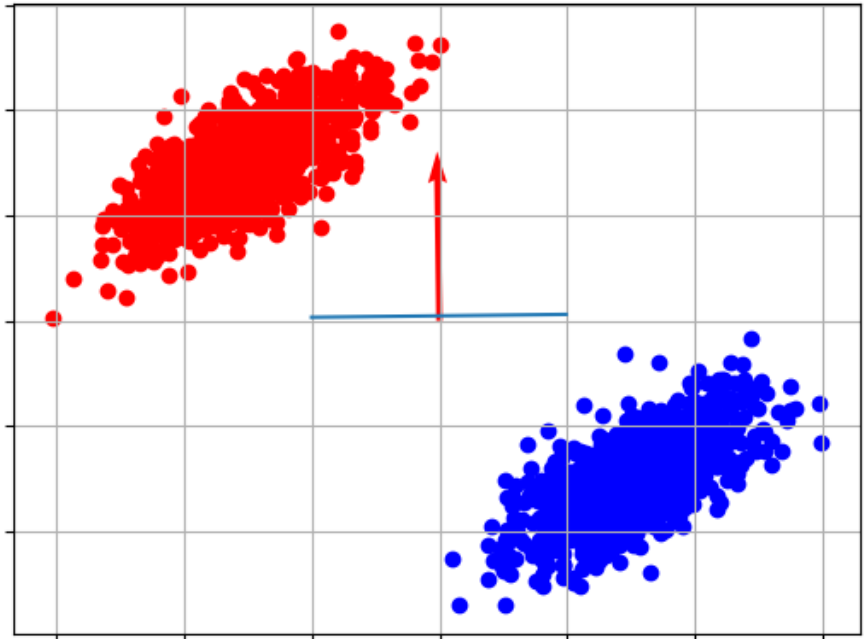
→ *Early stopping:* Çok fazla dönem, eğitim veri kümesinin aşırı sığmasına neden olabilirken, çok azı bir yetersizlik modeliyle sonuçlanabilir. Bir Makine Öğrenimi modelini yinelemeli olarak eğittiğinizde, birkaç yinelemeye kadar modelin performansının arttığını fark edeceksiniz. Belirli bir noktadan sonra, yineleme sayısını artırırsanız, model eğitim veri kümesinde daha iyi performans gösterir, ancak model aşırı yüklenir ve test veri kümelerinde düşük performans gösterir. Bu nedenle, modele fazla uymadan önce modelinizin eğitim tekrarlarını durdurmalısınız.

Nöron sayısını düşürmek

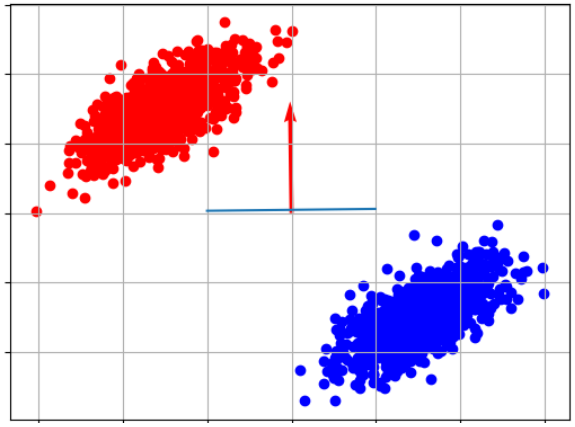
https://medium.com/kodluyoruz/makine-%C3%B6%C4%9Frenmesinde-overfitting-underfitting-best-fitting-kavramlar%C4%B1-9f80a5d42719

Destek Vektör Makinesi
(Support Vector Machines)
(SVM)

YSA’nın Problemi:



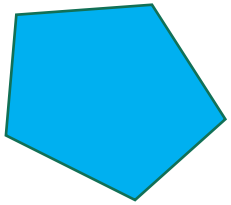
YSA'nın Problemi:

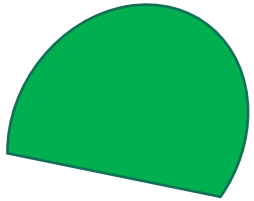


Hyperplane separation theorem[4] — Let *A* and *B* be two disjoint nonempty convex subsets of $\mathbf{R}^n$. Then there exist a nonzero vector *v* and a real number *c* such that

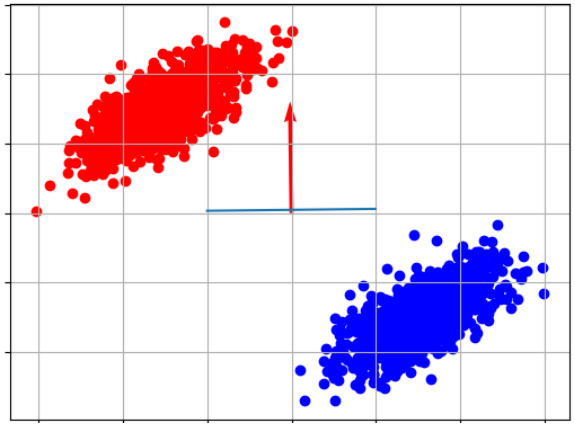
$$\langle x, v\rangle \geq c \text{ and } \langle y, v\rangle \leq c$$

for all *x* in *A* and *y* in *B*; i.e., the hyperplane $\langle \cdot, v\rangle = c$, *v* the normal vector, separates *A* and *B*.





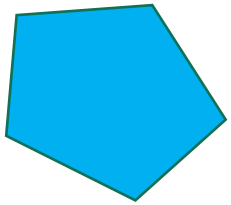
YSA'nın Problemi:

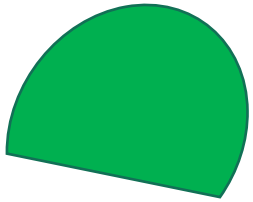


Hyperplane separation theorem[4] — Let *A* and *B* be two disjoint nonempty convex subsets of $\mathbf{R}^n$. Then there exist a nonzero vector *v* and a real number *c* such that

$$\langle x, v\rangle \geq c \text{ and } \langle y, v\rangle \leq c$$

for all *x* in *A* and *y* in *B*; i.e., the hyperplane $\langle \cdot, v\rangle = c$, *v* the normal vector, separates *A* and *B*.

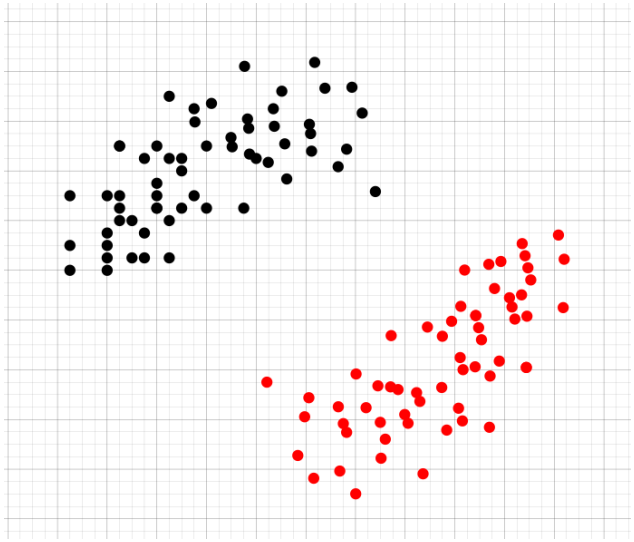


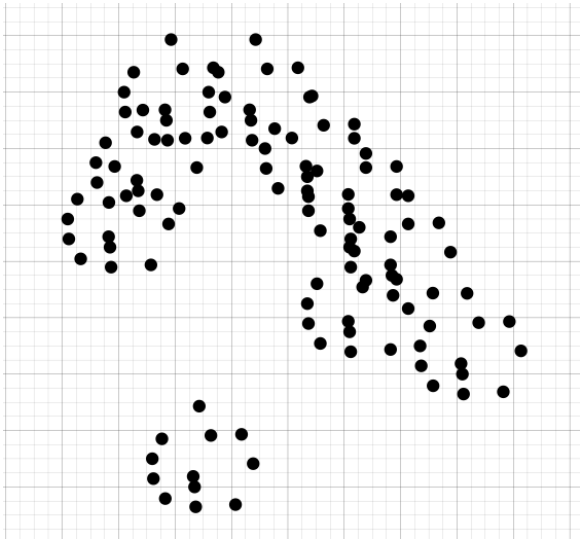


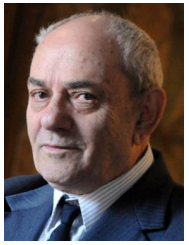
Dışbükey Örtüsü
Convex Hull

http://www.princeton.edu/~aaa/Public/Teaching/ORF523/S16/ORF523_S16_Lec5_gh.pdf

http://www.its.caltech.edu/~kcborder/Notes/SeparatingHyperplane.pdf







Vladimir N. Vapnik

The Nature of Statistical Learning Theory

Second Edition

Introduction: Four Periods in the Research of the Learning Problem

In the history of research of the learning problem one can extract four periods that can be characterized by four bright events:

(i) Constructing the first learning machines,

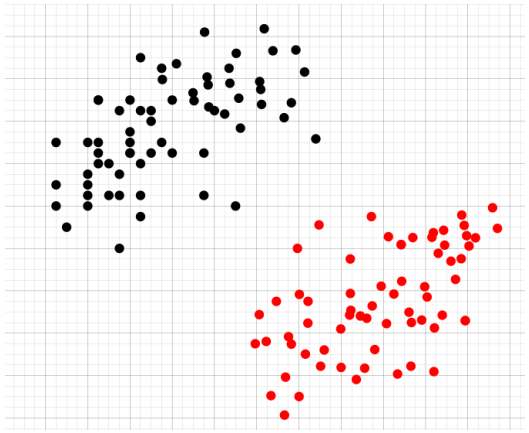
(ii) constructing the fundamentals of the theory,

(iii) constructing neural networks,

(iv) constructing the alternatives to neural networks.

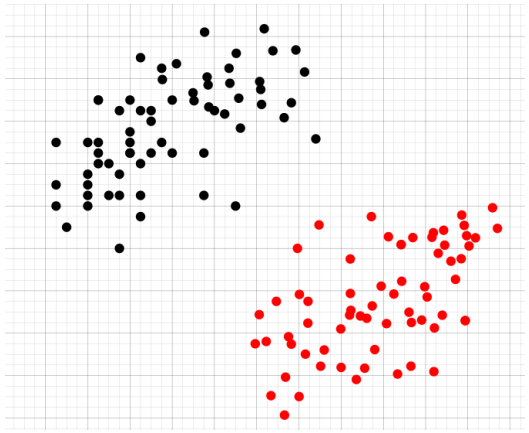
1990 - 95

https://books.google.com.tr/books?hl=en&lr=&id=EqgACAAAQBAJ&oi=fnd&pg=PR7&ots=g4F_gw8V15&sig=mpwjkHiVo8ozjBYYB_jkempAiTU&redir_esc=y#v=onepage&q&f=false



- Genel bir Makine Öğrenmesi Problemi

$$\mathcal{L}(y_i, p_i)$$



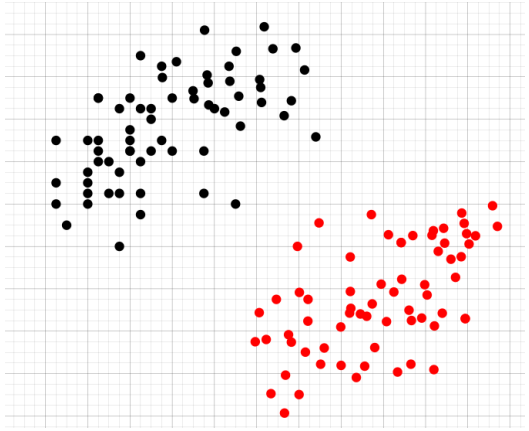
- Genel bir Makine Öğrenmesi Problemi

$$\mathcal{L}(y_i, p_i)$$

$$\sum_{\langle i \rangle} \mathcal{L}(y_i, p_i) \qquad p_i =$$

$f(\boldsymbol{x}_i; \boldsymbol{\Theta}) \longrightarrow$ GD Üzerinden YSA

$$\sum_{\langle i \rangle} \mathcal{L}(y_i, f(\boldsymbol{x}_i; \boldsymbol{\Theta}))$$



- Genel bir Makine Öğrenmesi Problemi

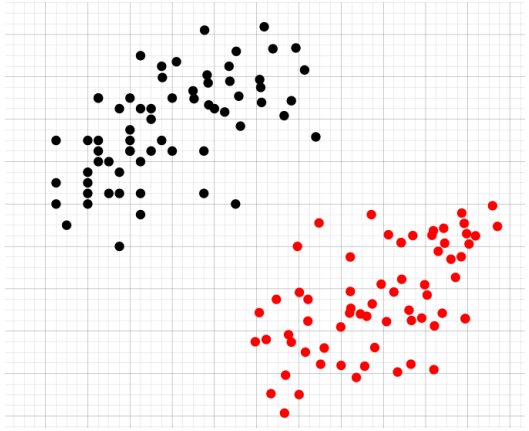
$$\mathcal{L}(y_i, p_i)$$

$$\sum_{\langle i \rangle} \mathcal{L}(y_i, p_i) \qquad p_i =$$

$f(\boldsymbol{x}_i; \boldsymbol{\Theta}) \longrightarrow$ GD Üzerinden YSA

$$\sum_{\langle i \rangle} \mathcal{L}(y_i, f(\boldsymbol{x}_i; \boldsymbol{\Theta}))$$

$f(\boldsymbol{x}_i; \boldsymbol{\Theta}) \longrightarrow$ Convex Opt. SVM



- Genel bir Makine Öğrenmesi Problemi

$$\mathcal{L}(y_i, p_i)$$

$$\sum_{\langle i \rangle} \mathcal{L}(y_i, p_i) \qquad p_i =$$

$f(\boldsymbol{x}_i; \boldsymbol{\Theta}) \longrightarrow$ GD Üzerinden YSA

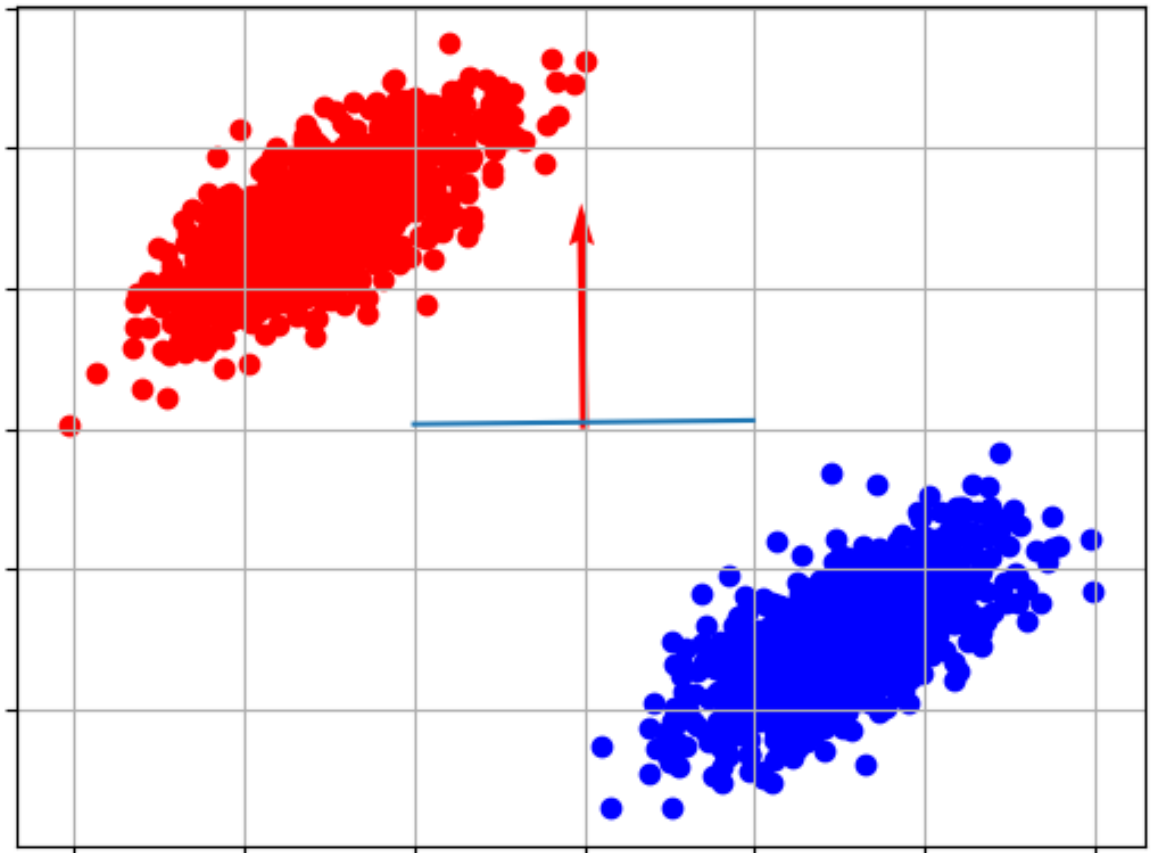
$$\sum_{\langle i \rangle} \mathcal{L}(y_i, f(\boldsymbol{x}_i; \boldsymbol{\Theta}))$$

$f(\boldsymbol{x}_i; \boldsymbol{\Theta}) \longrightarrow$ Convex Opt. SVM

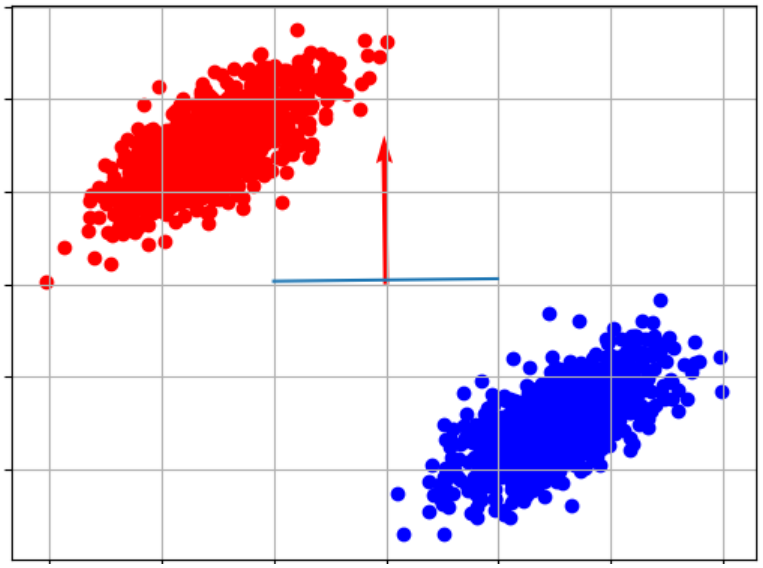
$$f(\boldsymbol{x}_i; \boldsymbol{\Theta}) = sign(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \qquad \boldsymbol{\Theta} = \{\mathbf{w}, \mathrm{b}\}$$

Destek Vektör Makinesi
(Support Vector Machines)
(SVM)

YSA’nın Problemi:



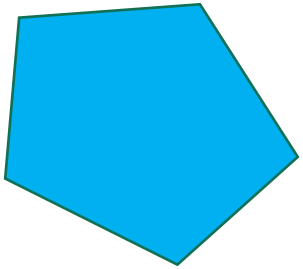
YSA'nın Problemi:

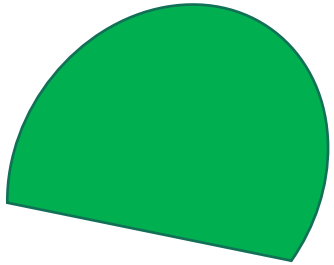


Hyperplane separation theorem[4] — Let *A* and *B* be two disjoint nonempty convex subsets of $\mathbf{R}^n$. Then there exist a nonzero vector *v* and a real number *c* such that

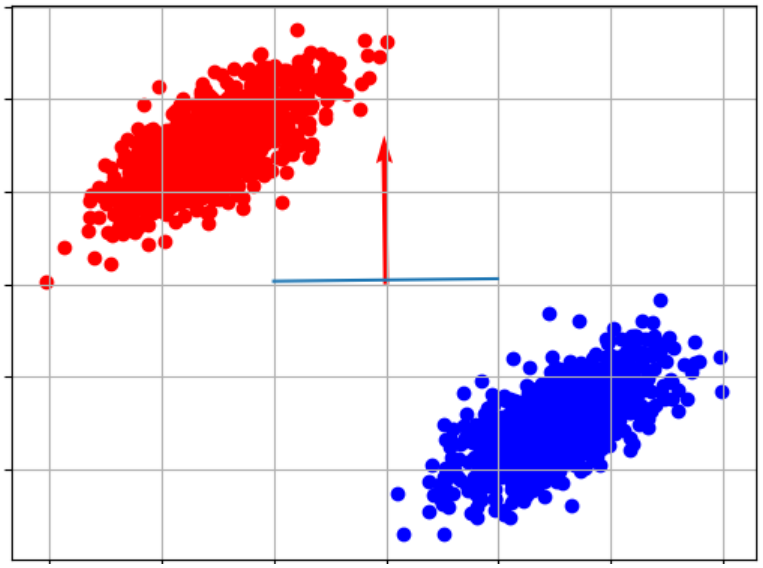
$$\langle x, v \rangle \geq c \text{ and } \langle y, v \rangle \leq c$$

for all *x* in *A* and *y* in *B*; i.e., the hyperplane $\langle \cdot, v \rangle = c$, *v* the normal vector, separates *A* and *B*.





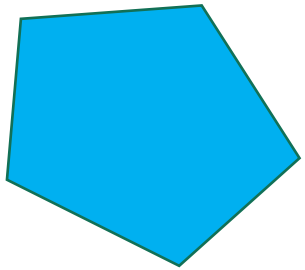
YSA'nın Problemi:



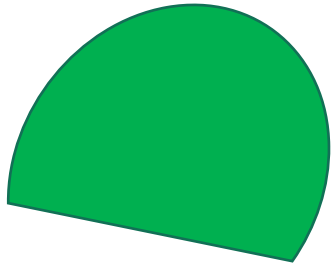
Hyperplane separation theorem[4] — Let *A* and *B* be two disjoint nonempty convex subsets of $\mathbf{R}^n$. Then there exist a nonzero vector *v* and a real number *c* such that

$$\langle x, v\rangle \geq c \text{ and } \langle y, v\rangle \leq c$$

for all *x* in *A* and *y* in *B*; i.e., the hyperplane $\langle \cdot, v\rangle = c$, *v* the normal vector, separates *A* and *B*.

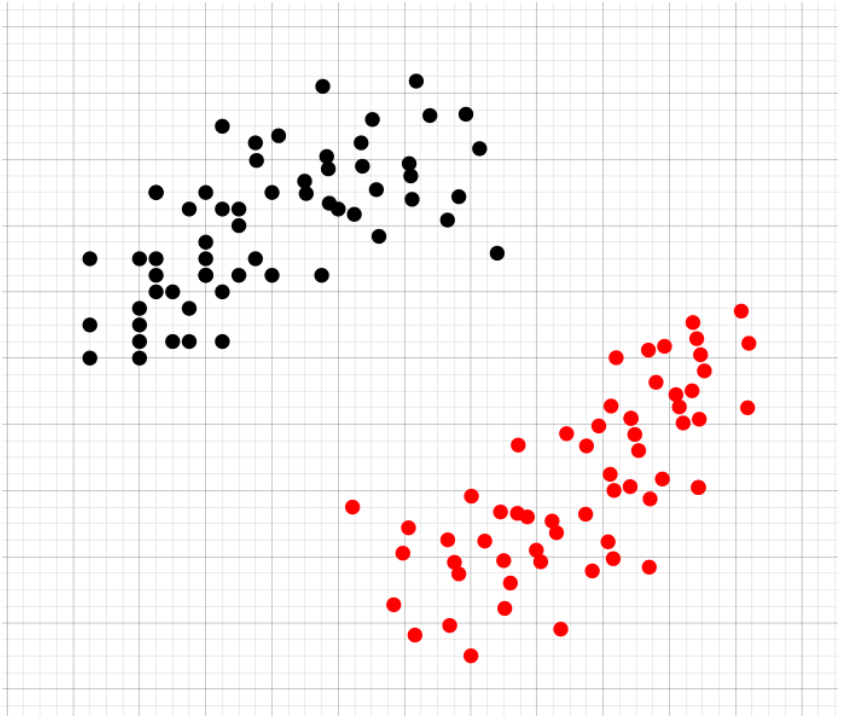


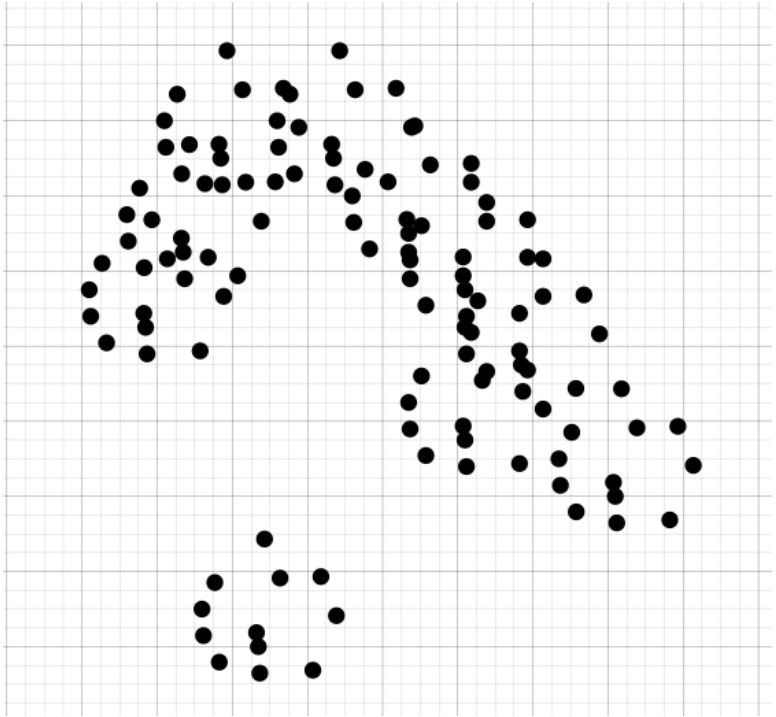
Dışbükey Örtüsü
Convex Hull

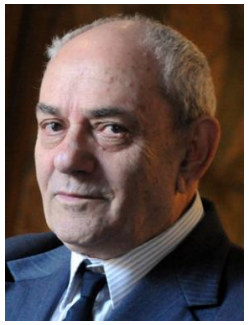


http://www.princeton.edu/~aaa/Public/Teaching/ORF523/S16/ORF523_S16_Lec5_gh.pdf

http://www.its.caltech.edu/~kcborder/Notes/SeparatingHyperplane.pdf







Vladimir N. Vapnik

The Nature of Statistical Learning Theory

Second Edition

Introduction: Four Periods in the Research of the Learning Problem

In the history of research of the learning problem one can extract four periods that can be characterized by four bright events:

(i) Constructing the first learning machines,

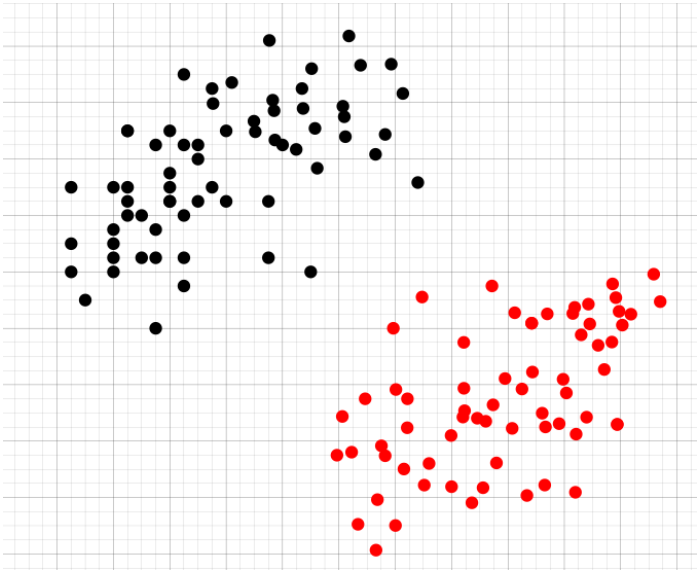
(ii) constructing the fundamentals of the theory,

(iii) constructing neural networks,

(iv) constructing the alternatives to neural networks.

1990 - 95

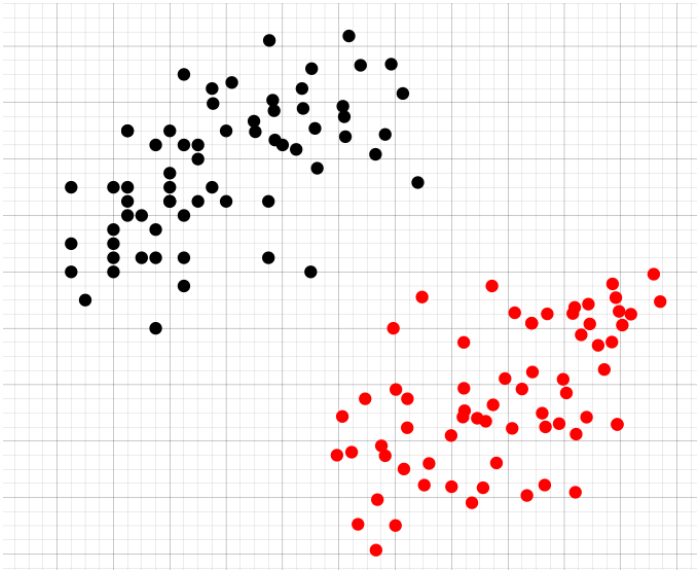
https://books.google.com.tr/books?hl=en&lr=&id=EqgACAAAQBAJ&oi=fnd&pg=PR7&ots=g4F_gw8V15&sig=mpwjkHiVo8ozjBYYB_jkempAiTU&redir_esc=y#v=onepage&q&f=false



- Genel bir Makine Öğrenmesi Problemi

$$\mathcal{L}(y_i, p_i)$$

$$\sum_{\langle i \rangle} \mathcal{L}(y_i, p_i) \qquad p_i =$$



- Genel bir Makine Öğrenmesi Problemi

$$\mathcal{L}(y_i, p_i)$$

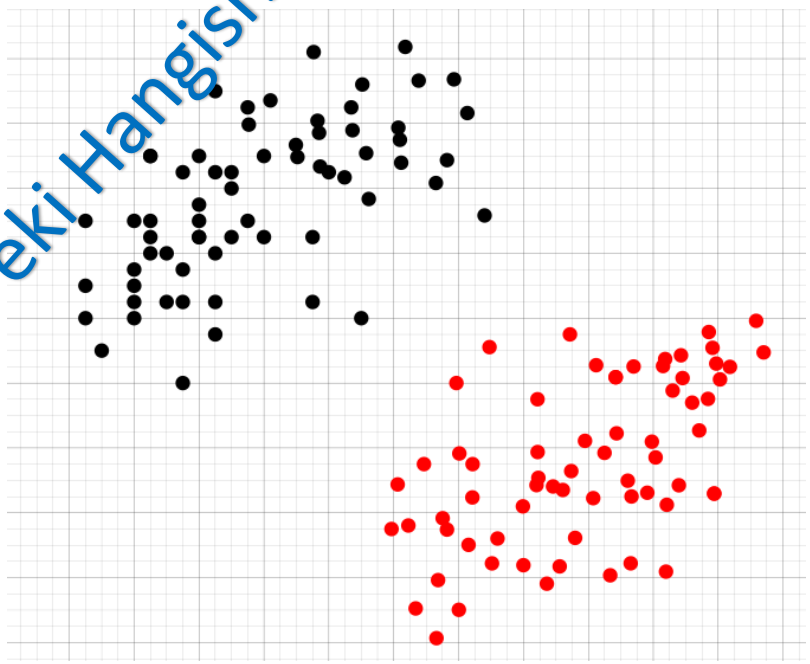
$$\sum_{\langle i \rangle} \mathcal{L}(y_i, p_i) \qquad p_i =$$

$$f(\boldsymbol{x}_i; \boldsymbol{\Theta}) \longrightarrow \text{GD Üzerinden YSA}$$

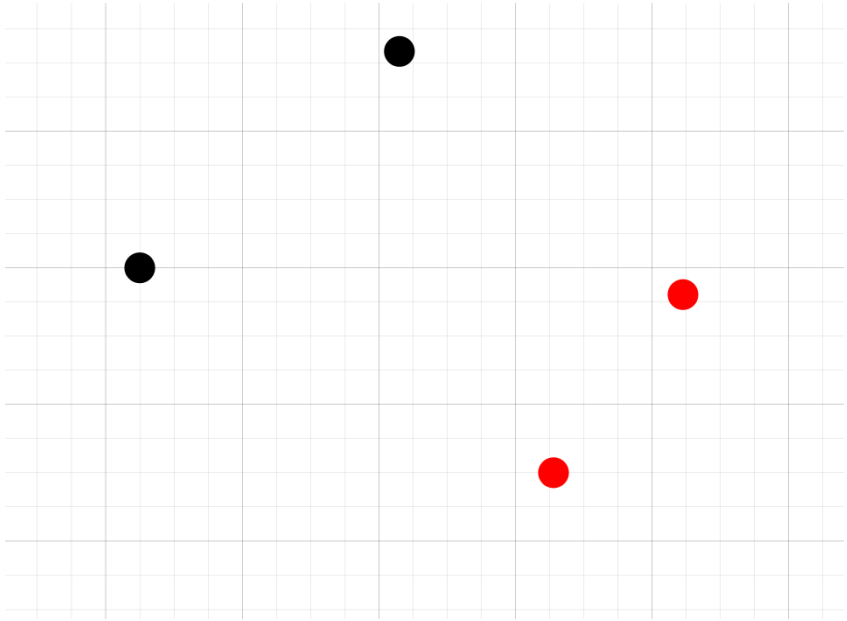
$$\sum_{\langle i \rangle} \mathcal{L}(y_i, f(\boldsymbol{x}_i; \boldsymbol{\Theta}))$$

$$f(\boldsymbol{x}_i; \boldsymbol{\Theta}) \longrightarrow \text{Convex Opt. SVM}$$

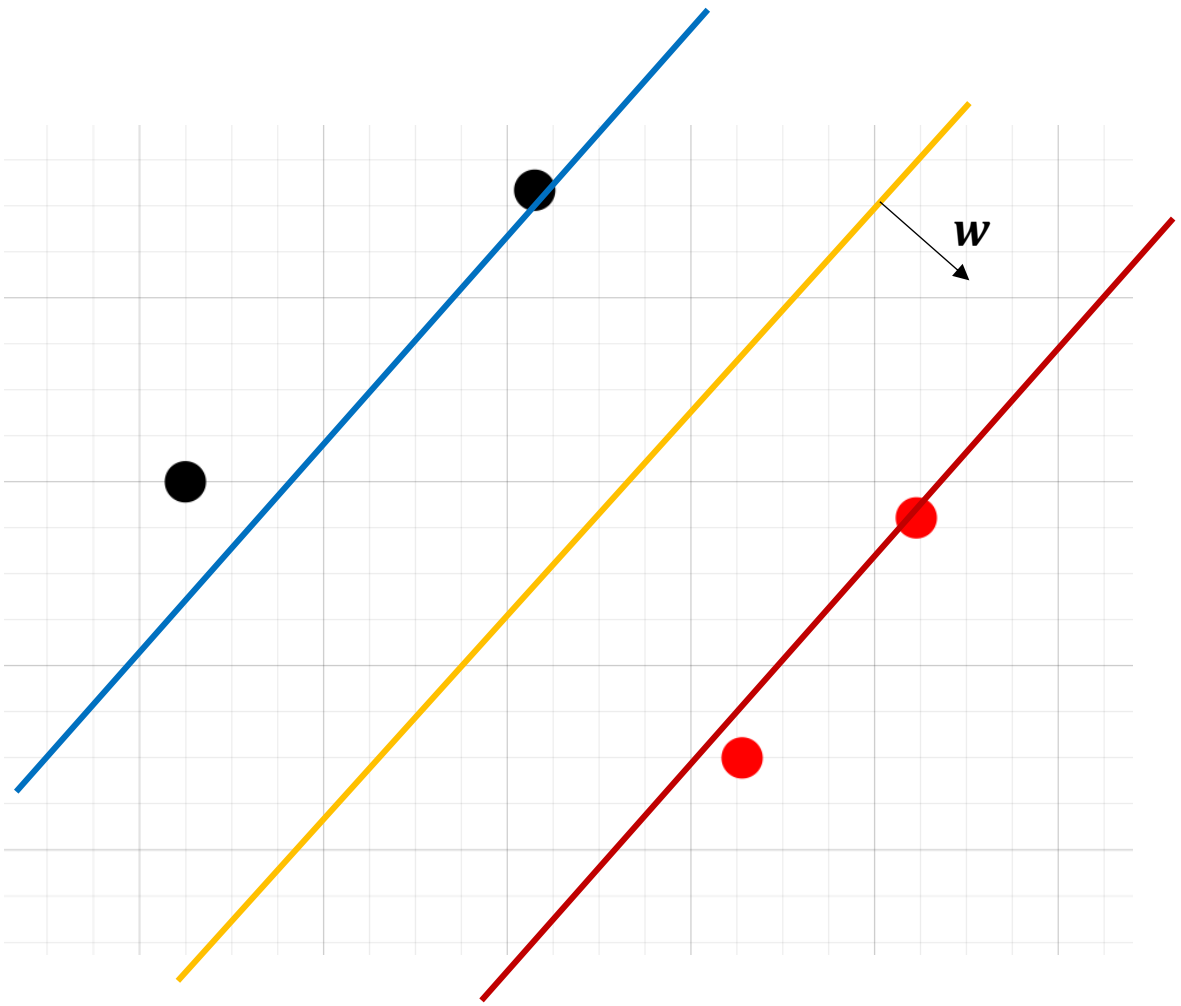
$$f(\boldsymbol{x}_i; \boldsymbol{\Theta}) = sign(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \qquad \boldsymbol{\Theta} = \{\mathbf{w}, \mathrm{b}\}$$



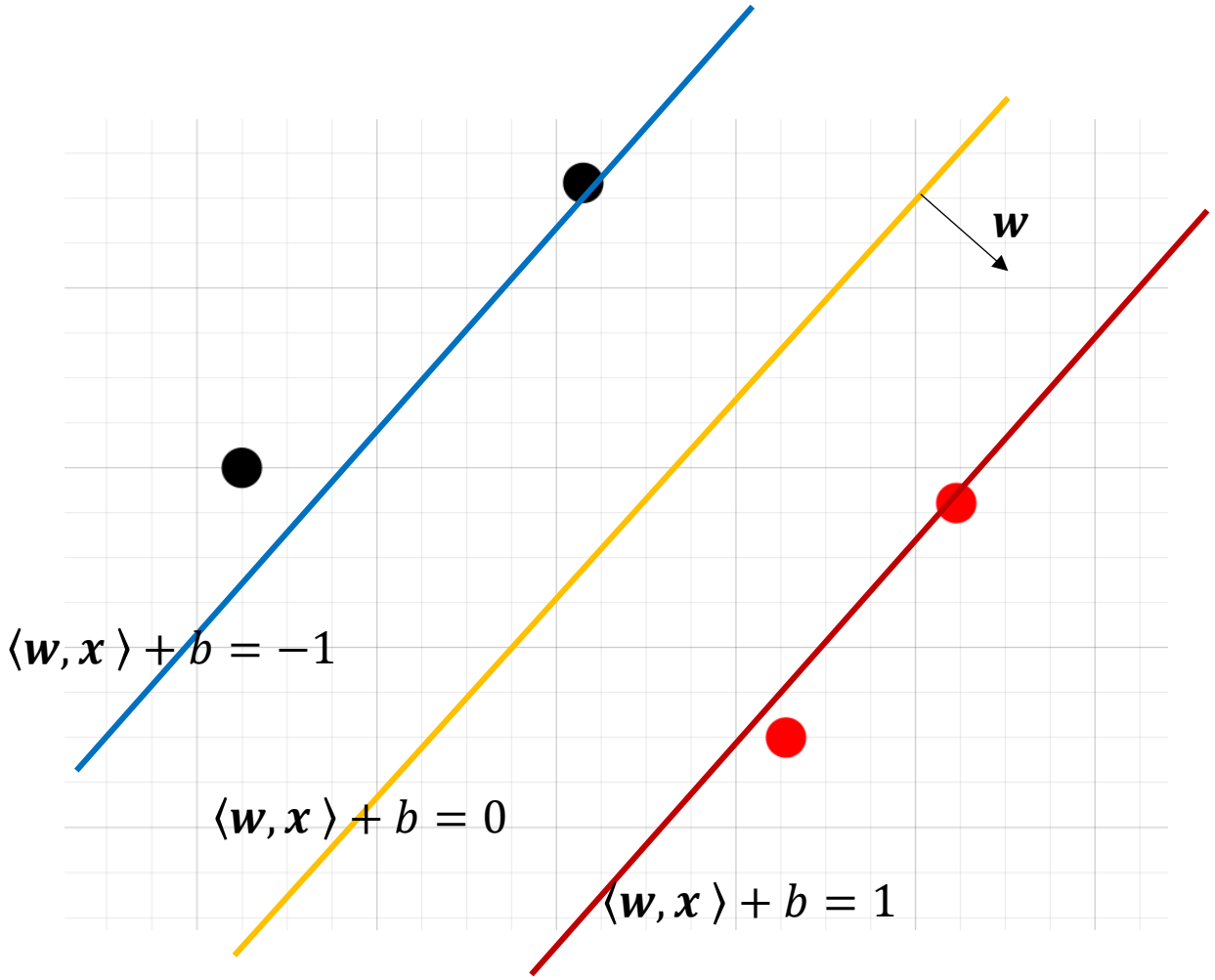
- **SVM:** Maksimum Aralık (Margin)
- (Eşdeğer Eğrileri)

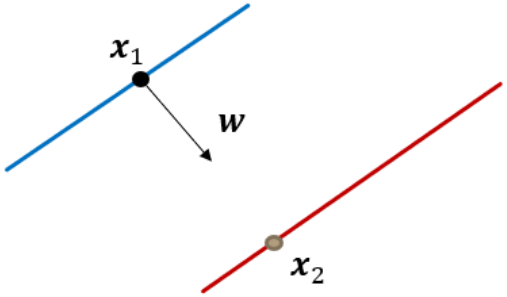


- **SVM:** Maksimum Aralık (Margin)
- (Eşdeğer Eğrileri)



- **SVM:** Maksimum Aralık (Margin)
- (Eşdeğer Eğrileri)





$$x_2 = x_1 + \lambda w$$

$\Rightarrow \mathbf{w}^T\mathbf{x_2} + b = 1$ where $\mathbf{x_2} = \mathbf{x_1} + \lambda\mathbf{w}$

$\Rightarrow \mathbf{w}^T(\mathbf{x_1} + \lambda\mathbf{w}) + b = 1$

$\Rightarrow \mathbf{w}^T\mathbf{x_1} + b + \lambda\mathbf{w}^T\mathbf{w} = 1$ where $\mathbf{w}^T\mathbf{x_1} + b = -1$

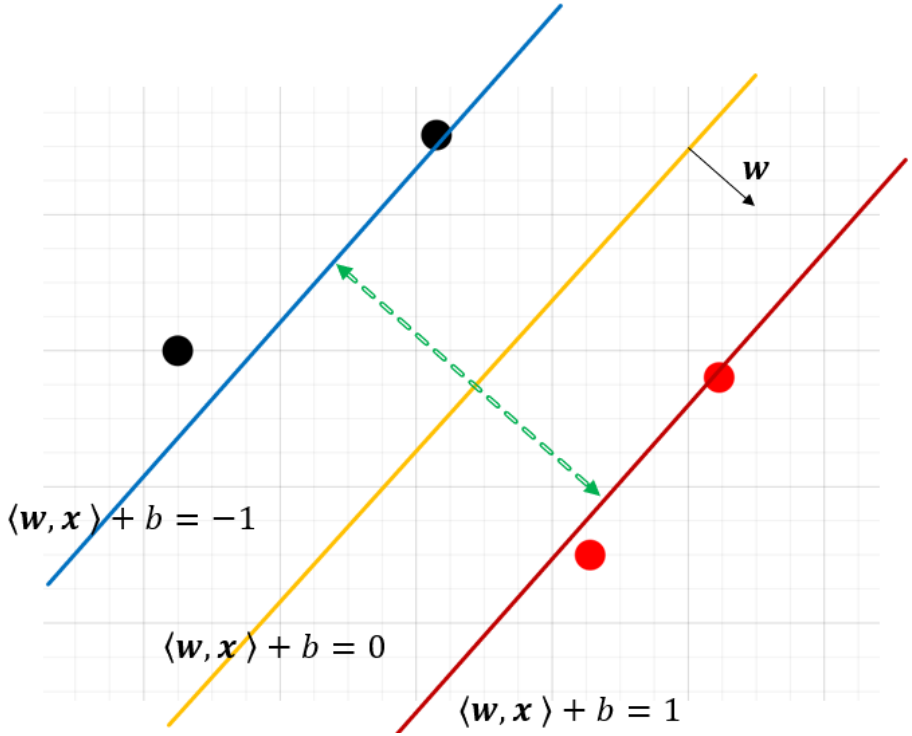
$\Rightarrow -1 + \lambda\mathbf{w}^T\mathbf{w} = 1$

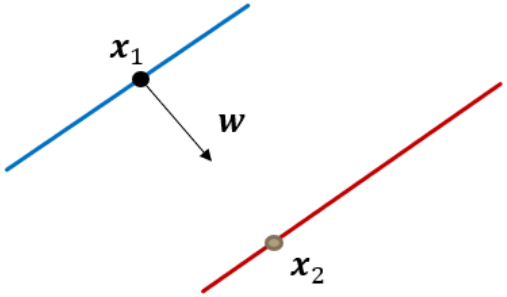
$\Rightarrow \lambda\mathbf{w}^T\mathbf{w} = 2$

$\Rightarrow \lambda = \frac{2}{\mathbf{w}^T\mathbf{w}} = \frac{2}{\|\mathbf{w}\|^2}$

And so, the distance $\lambda\|\mathbf{w}\|$ is $\frac{2}{\|\mathbf{w}\|^2}\|\mathbf{w}\| = \frac{2}{\|\mathbf{w}\|} = \frac{2}{\sqrt{\mathbf{w}^T\mathbf{w}}}$

http://web.mit.edu/zoya/www/SVM.pdf





$$x_2 = x_1 + \lambda w$$

$\Rightarrow \mathbf{w}^T\mathbf{x_2} + b = 1$ where $\mathbf{x_2} = \mathbf{x_1} + \lambda\mathbf{w}$

$\Rightarrow \mathbf{w}^T(\mathbf{x_1} + \lambda\mathbf{w}) + b = 1$

$\Rightarrow \mathbf{w}^T\mathbf{x_1} + b + \lambda\mathbf{w}^T\mathbf{w} = 1$ where $\mathbf{w}^T\mathbf{x_1} + b = -1$

$\Rightarrow -1 + \lambda\mathbf{w}^T\mathbf{w} = 1$

$\Rightarrow \lambda\mathbf{w}^T\mathbf{w} = 2$

$\Rightarrow \lambda = \frac{2}{\mathbf{w}^T\mathbf{w}} = \frac{2}{\|\mathbf{w}\|^2}$

And so, the distance $\lambda\|\mathbf{w}\|$ is $\frac{2}{\|\mathbf{w}\|^2}\|\mathbf{w}\| = \frac{2}{\|\mathbf{w}\|} = \frac{2}{\sqrt{\mathbf{w}^T\mathbf{w}}}$

http://web.mit.edu/zoya/www/SVM.pdf

$$\max \quad margin$$

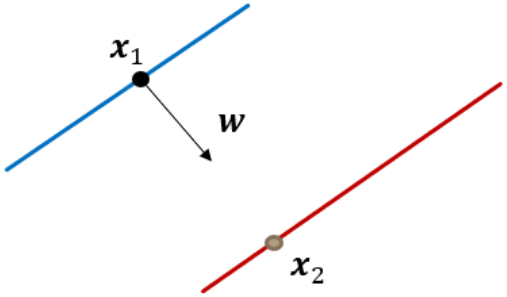
$$max\ \frac{2}{|\boldsymbol{w}|}$$

$$\min_{\boldsymbol{w},b} \frac{1}{2}|\boldsymbol{w}|^2$$

$$subject\ to\ ??$$

$$\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \geq 1 \qquad y_i = 1$$

$$\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \leq -1 \qquad y_i = -1$$



$$x_2 = x_1 + \lambda w$$

$\Rightarrow \mathbf{w}^T\mathbf{x_2} + b = 1$ where $\mathbf{x_2} = \mathbf{x_1} + \lambda\mathbf{w}$

$\Rightarrow \mathbf{w}^T(\mathbf{x_1} + \lambda\mathbf{w}) + b = 1$

$\Rightarrow \mathbf{w}^T\mathbf{x_1} + b + \lambda\mathbf{w}^T\mathbf{w} = 1$ where $\mathbf{w}^T\mathbf{x_1} + b = -1$

$\Rightarrow -1 + \lambda\mathbf{w}^T\mathbf{w} = 1$

$\Rightarrow \lambda\mathbf{w}^T\mathbf{w} = 2$

$\Rightarrow \lambda = \frac{2}{\mathbf{w}^T\mathbf{w}} = \frac{2}{\|\mathbf{w}\|^2}$

And so, the distance $\lambda\|\mathbf{w}\|$ is $\frac{2}{\|\mathbf{w}\|^2}\|\mathbf{w}\| = \frac{2}{\|\mathbf{w}\|} = \frac{2}{\sqrt{\mathbf{w}^T\mathbf{w}}}$

http://web.mit.edu/zoya/www/SVM.pdf

$$\max \quad margin$$

$$max\ \frac{2}{|\boldsymbol{w}|}$$

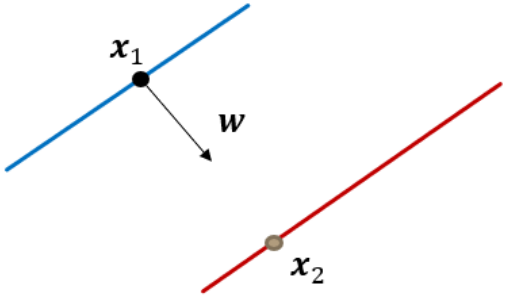
$$\min_{\boldsymbol{w},b} \frac{1}{2}|\boldsymbol{w}|^2$$

$$subject\ to\ ??$$

$$\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \geq 1 \qquad y_i = 1$$

$$\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \leq -1 \qquad y_i = -1$$

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1$$



$$x_2 = x_1 + \lambda w$$

$\Rightarrow \mathbf{w}^T\mathbf{x_2} + b = 1$ where $\mathbf{x_2} = \mathbf{x_1} + \lambda\mathbf{w}$

$\Rightarrow \mathbf{w}^T(\mathbf{x_1} + \lambda\mathbf{w}) + b = 1$

$\Rightarrow \mathbf{w}^T\mathbf{x_1} + b + \lambda\mathbf{w}^T\mathbf{w} = 1$ where $\mathbf{w}^T\mathbf{x_1} + b = -1$

$\Rightarrow -1 + \lambda\mathbf{w}^T\mathbf{w} = 1$

$\Rightarrow \lambda\mathbf{w}^T\mathbf{w} = 2$

$\Rightarrow \lambda = \frac{2}{\mathbf{w}^T\mathbf{w}} = \frac{2}{\|\mathbf{w}\|^2}$

And so, the distance $\lambda\|\mathbf{w}\|$ is $\frac{2}{\|\mathbf{w}\|^2}\|\mathbf{w}\| = \frac{2}{\|\mathbf{w}\|} = \frac{2}{\sqrt{\mathbf{w}^T\mathbf{w}}}$

http://web.mit.edu/zoya/www/SVM.pdf

$$\max \quad margin$$

$$max\ \frac{2}{|\boldsymbol{w}|}$$

$$\min_{\boldsymbol{w},b} \frac{1}{2}|\boldsymbol{w}|^2$$

$$subject\ to\ ??$$

$$\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \geq 1 \qquad y_i = 1$$

$$\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \leq -1 \qquad y_i = -1$$

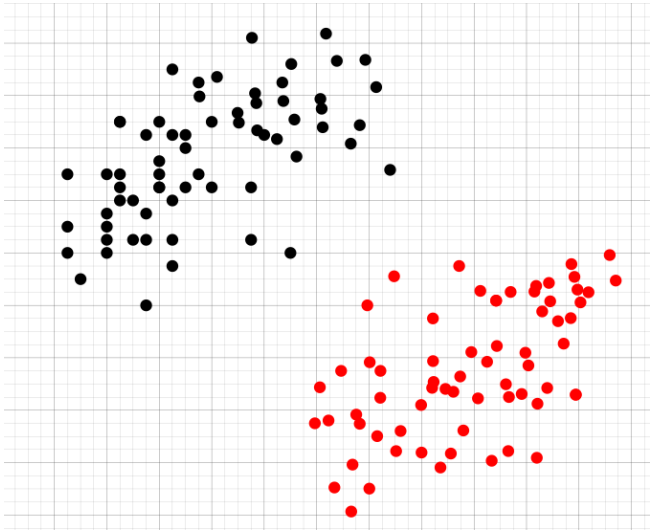
$$\boxed{y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1}$$

Başka örnek?

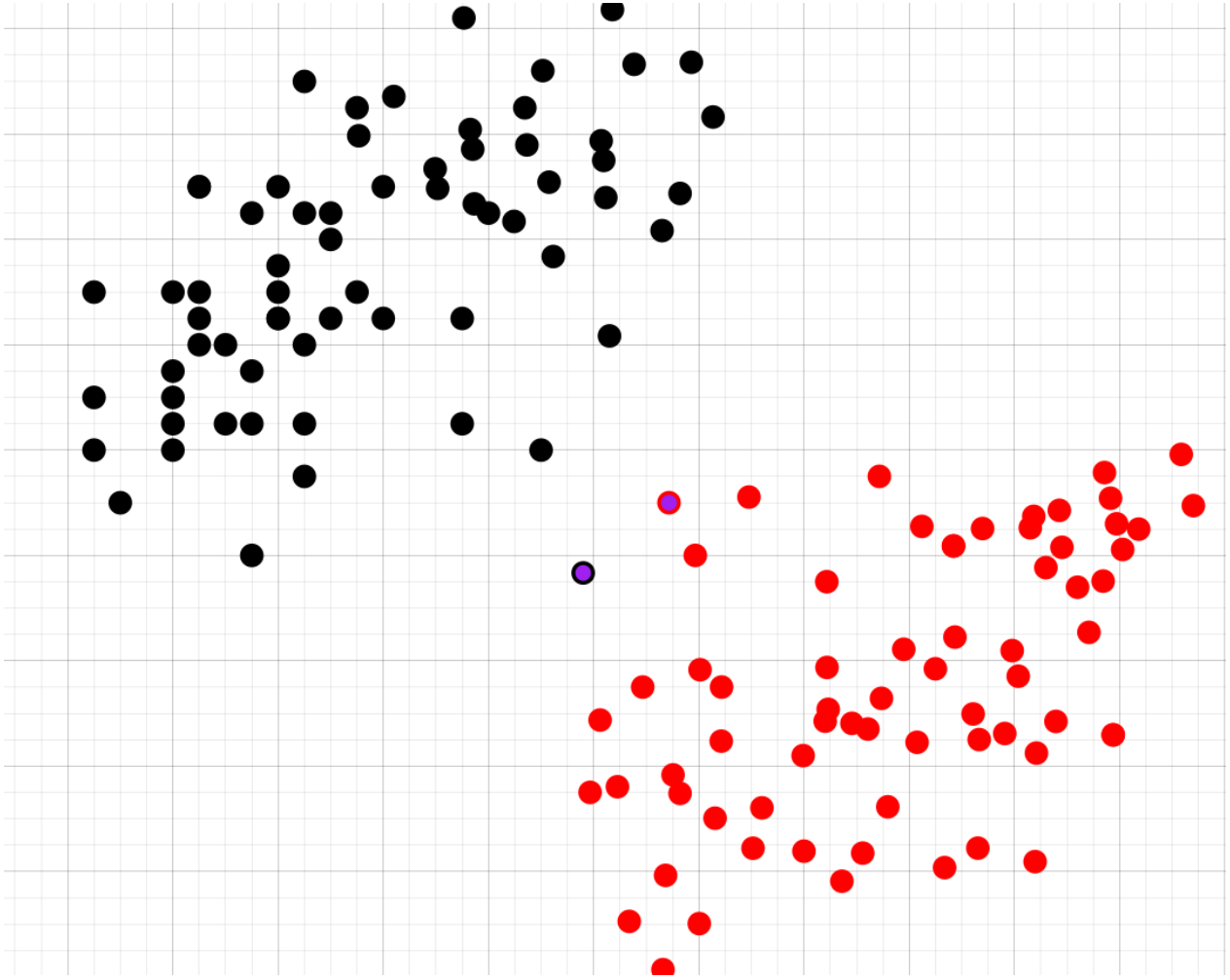
- **SVM:** Maksimum Aralık (Margin) Neden Önemli
- $f(\boldsymbol{x}; \boldsymbol{\Theta})'in\ veriyi\ genel\ olarak$ öğ$renmesi$!!

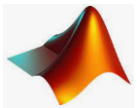
less margin? Do you think such kind of decision boundary will generalize well on unseen data? The answer is: No. The `green` decision boundary has a wider margin that would allow it to generalize well on unseen data. In that sense, soft margin formulation would also help in avoiding the overfitting problem.

https://towardsdatascience.com/support-vector-machines-soft-margin-formulation-and-kernel-trick-4c9729dc8efe



- **SVM:** Maksimum Aralık (Margin) Neden Önemli
- «Hard Margin» Tanımı !!
- Doğrusal olmayan veri örneği





- **SVM:** Maksimum Aralık (Margin) Neden Önemli
- «Soft Margin» Tanımı !!

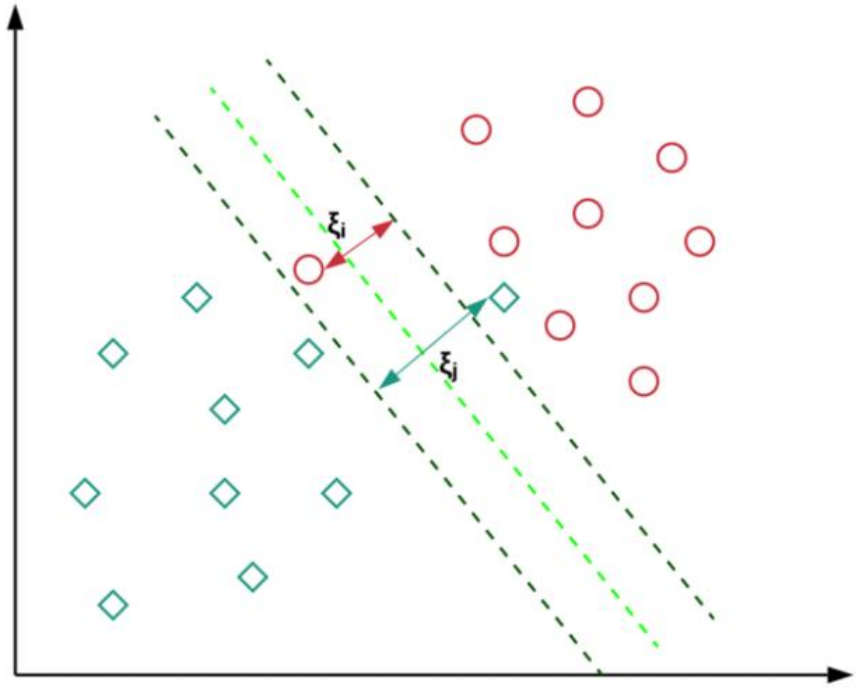
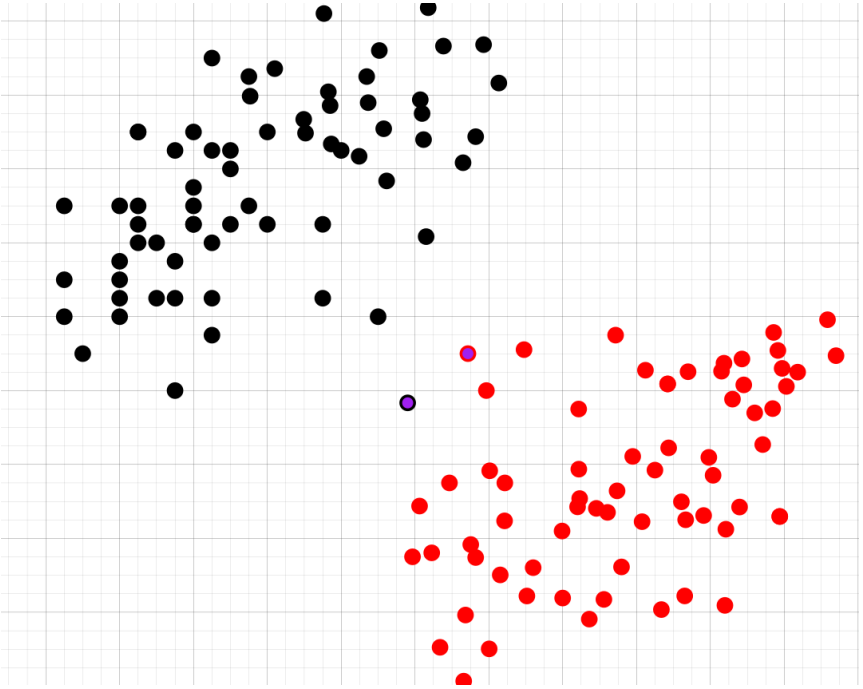


Figure 3: The penalty incurred by data points for being on the wrong side of the decision boundary



$$\min_{\boldsymbol{w},b} \frac{1}{2}|\boldsymbol{w}|^2 \qquad subject\ to$$

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) + \xi_i \geq 1$$

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

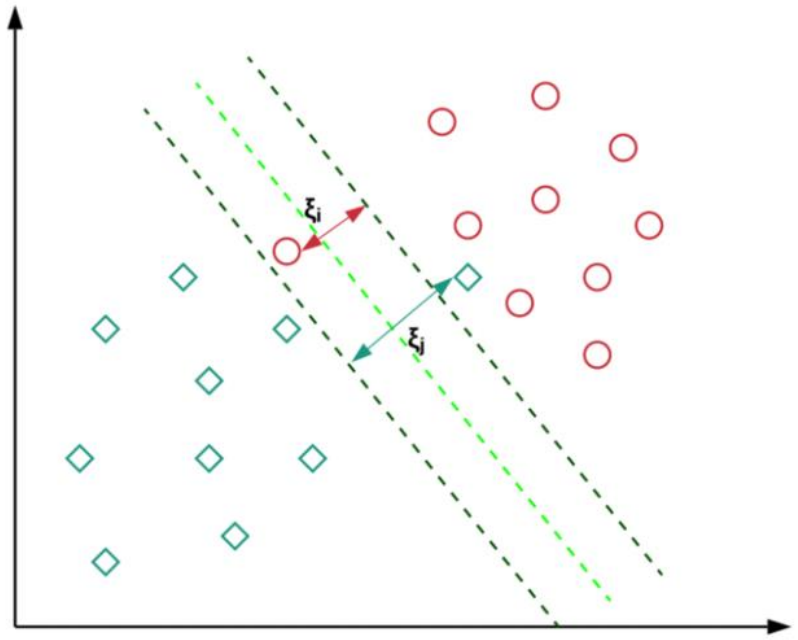


Figure 3: The penalty incurred by data points for being on the wrong side of the decision boundary

$$\min_{\boldsymbol{w},b} \frac{1}{2}|\boldsymbol{w}|^2 \qquad subject\ to$$

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) + \xi_i \geq 1$$

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

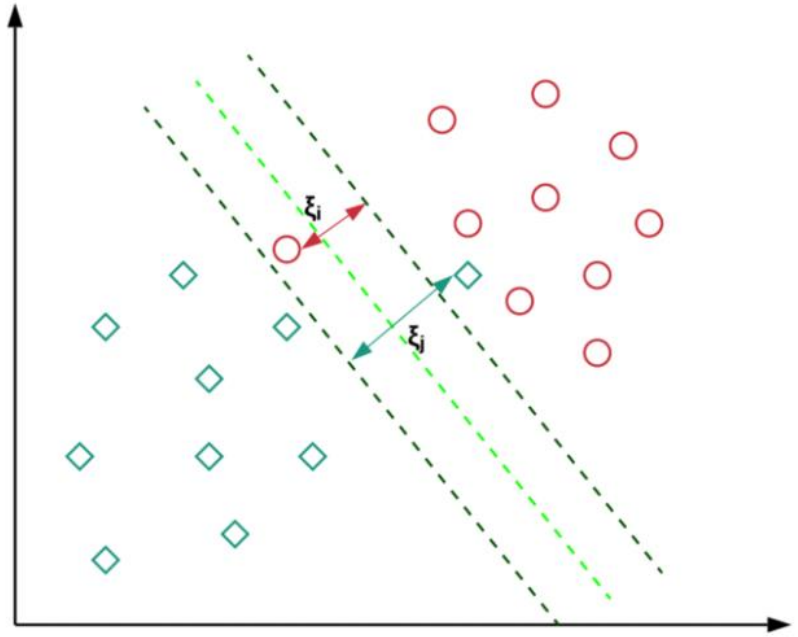
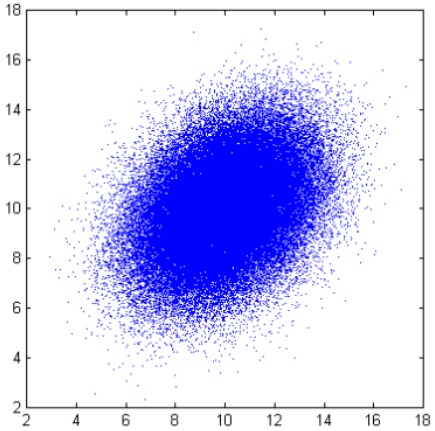
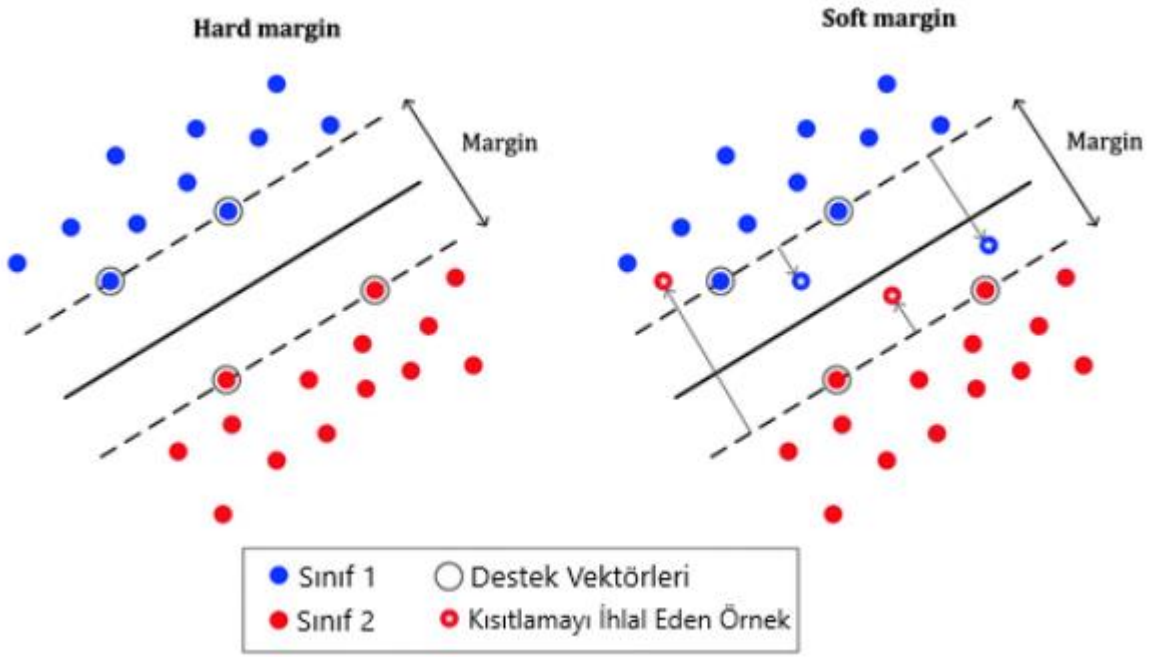


Figure 3: The penalty incurred by data points for being on the wrong side of the decision boundary

$$\min_{\boldsymbol{w},b} \frac{1}{2}|\boldsymbol{w}|^2 + C \times \sum \xi_i$$

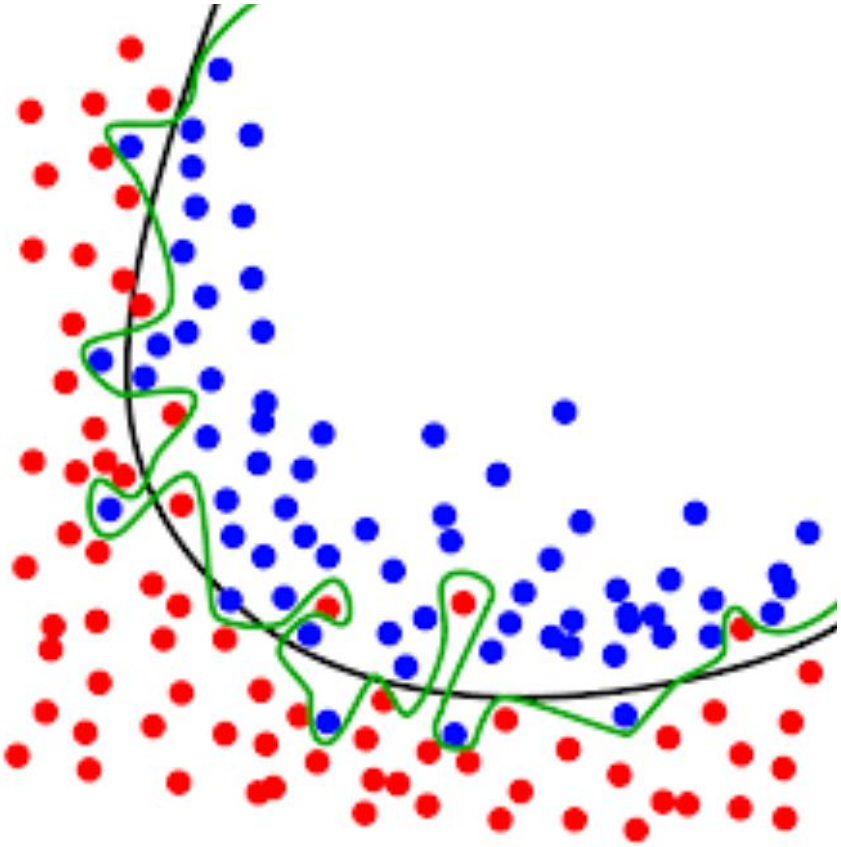
Küre Örneği !!





https://medium.com/deep-learning-turkiye/nedir-bu-destek-vekt%C3%B6r-makineleri-makine-%C3%B6%C4%9Frenmesi-serisi-2-94e576e4223e

Modelin Ezberlemesi «Overfitting»



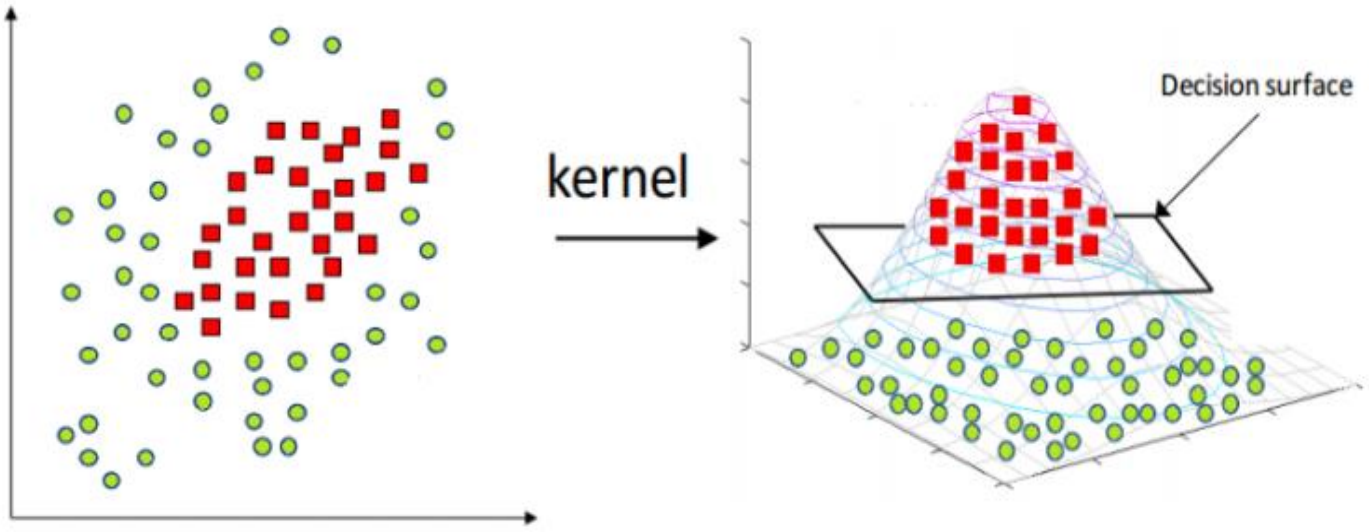
https://machinelearningmedium.com/2017/09/08/overfitting-and-regularization/

SVM vs Doğrusal Olmayan Veriler

YSA’nın taktiği

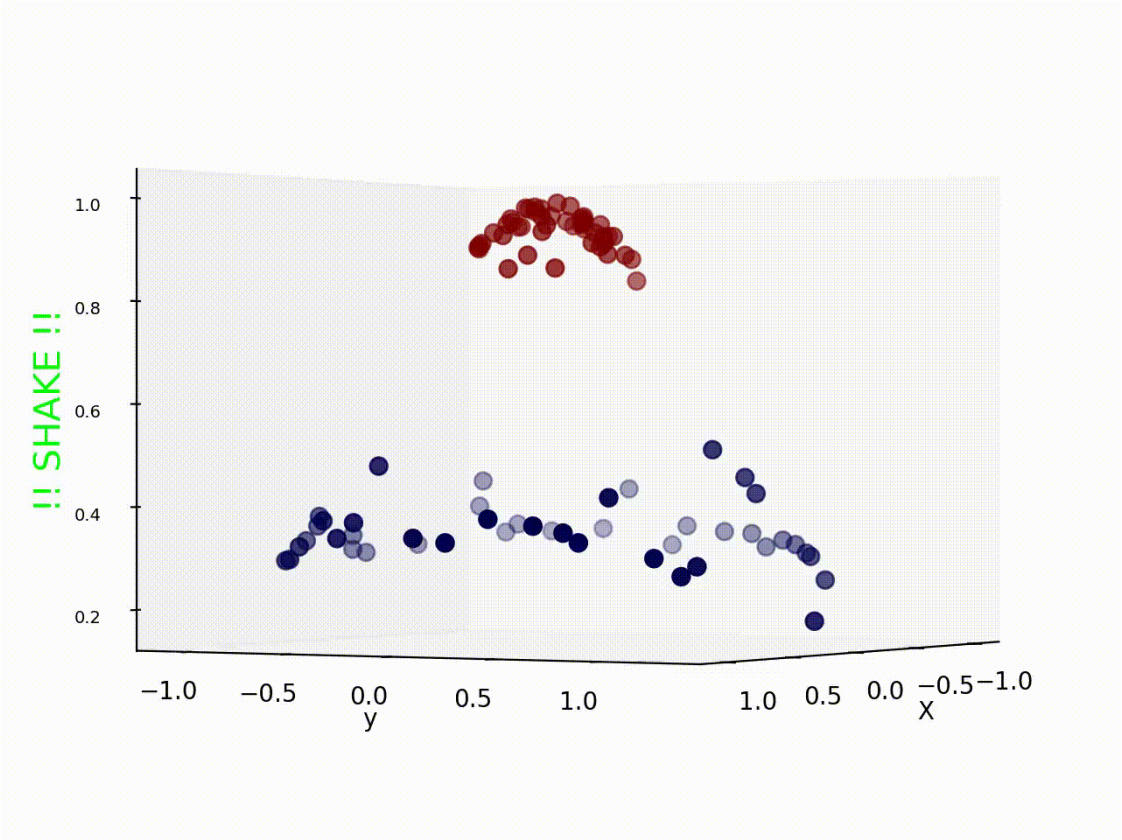
SVM’in taktiği

SVM vs Doğrusal Olmayan Veriler



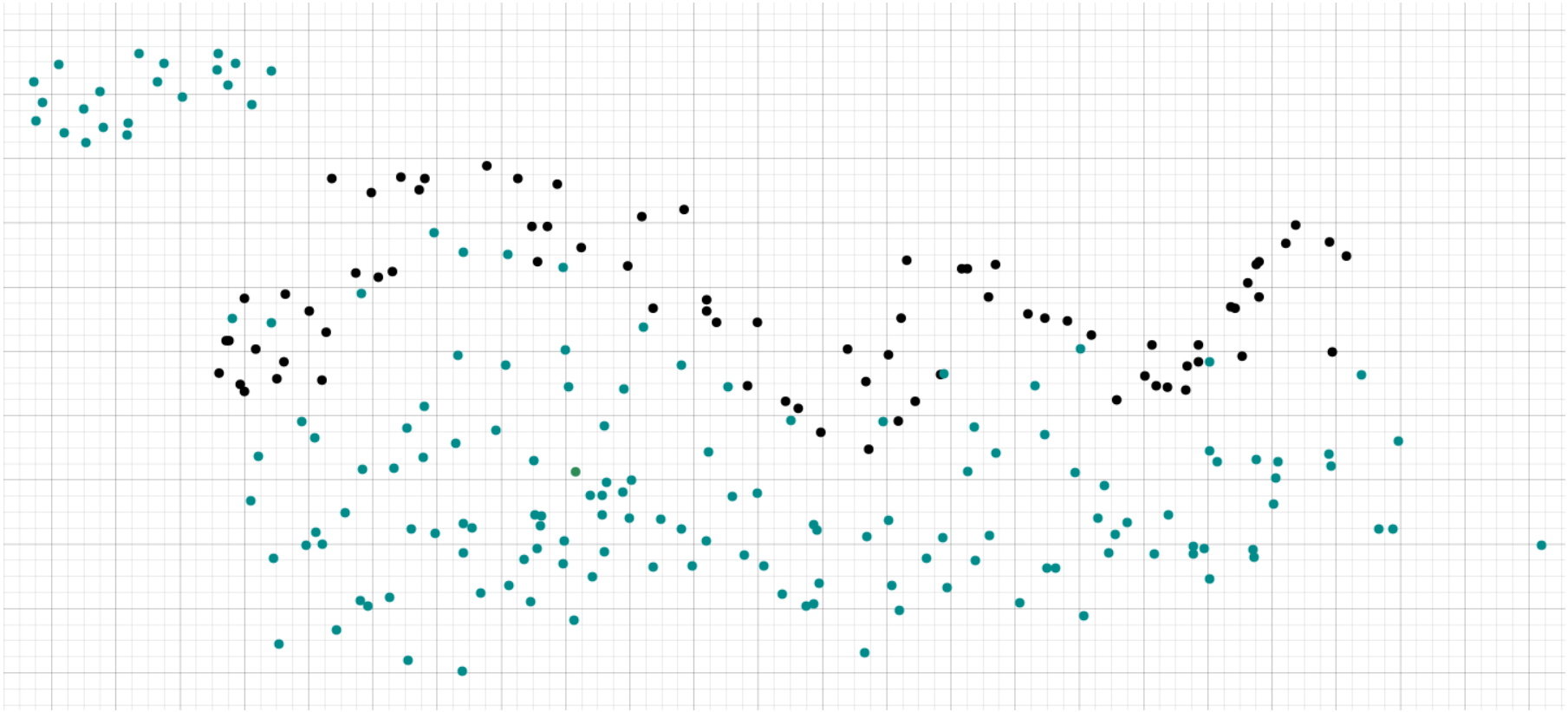
https://towardsdatascience.com/understanding-support-vector-machine-part-2-kernel-trick-mercers-theorem-e1e6848c6c4d

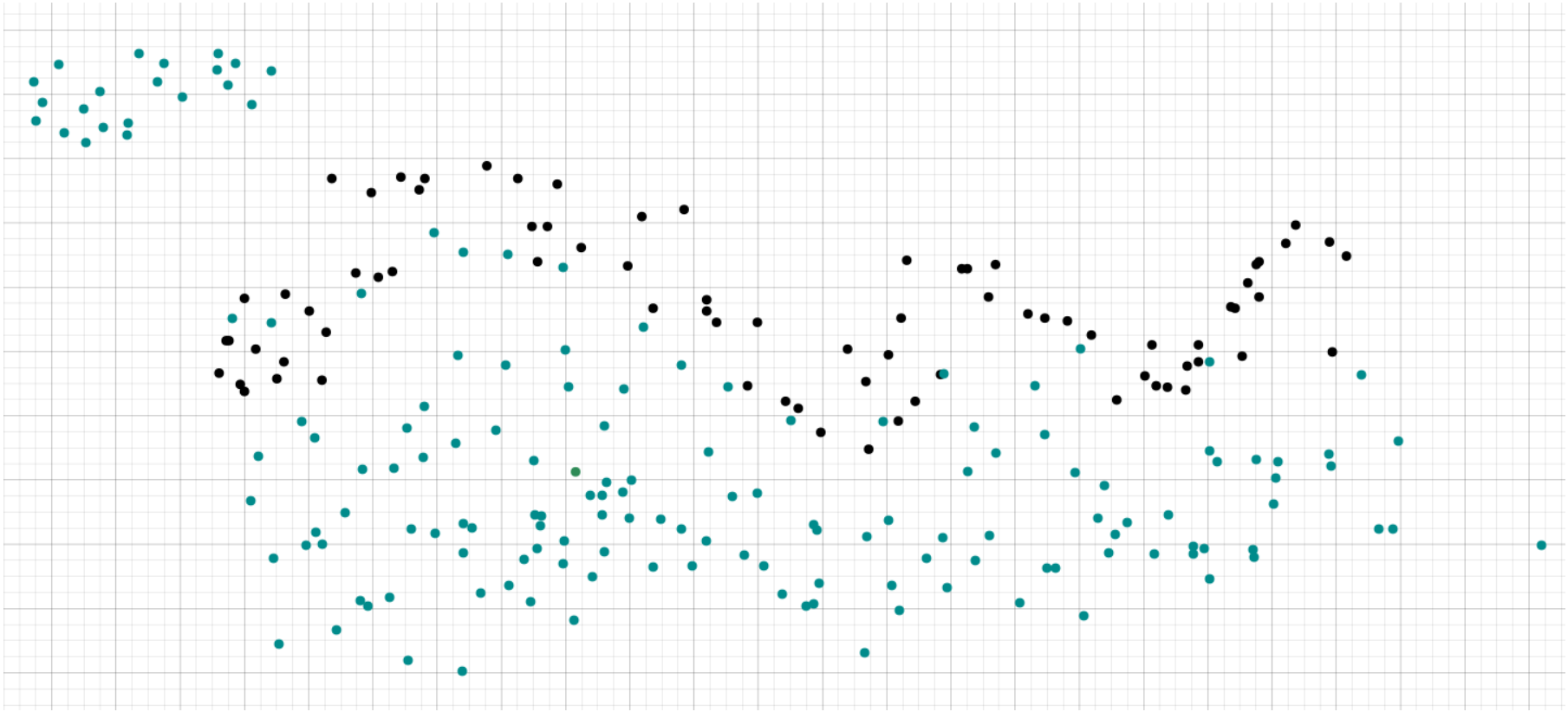
SVM vs Doğrusal Olmayan Veriler



https://towardsdatascience.com/understanding-support-vector-machine-part-2-kernel-trick-mercers-theorem-e1e6848c6c4d

Kernel SVM





Yeni Boyutlar

$$\min_{\boldsymbol{w},b} \frac{1}{2}|\boldsymbol{w}|^2 \qquad s.t.$$

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1$$

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - 1 \geq 0$$

$$\mathcal{L}\{\boldsymbol{w}, b, a_i\} = \min_{\boldsymbol{w},b} \frac{1}{2}|\boldsymbol{w}|^2 + \sum_{\langle i \rangle} \alpha_i [y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - 1]$$

$$\min_{\boldsymbol{w},b} \frac{1}{2}|\boldsymbol{w}|^2 \qquad s.t.$$

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1$$

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - 1 \geq 0$$

$$\mathcal{L}\{\boldsymbol{w}, b, a_i\} = \min_{\boldsymbol{w},b} \frac{1}{2}|\boldsymbol{w}|^2 + \sum_{\langle i \rangle} \alpha_i [y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - 1]$$

(Dual) $$\max_{\vec{\alpha} \geq 0} \min_{\vec{w},b} \frac{1}{2}||\vec{w}||^2 - \sum_j \alpha_j \left[(\vec{w} \cdot \vec{x}_j + b)\, y_j - 1\right]$$

Can solve for optimal **w**, b as function of α:

$$\frac{\partial L}{\partial w} = w - \sum_j \alpha_j y_j x_j \quad \rightarrow \quad \mathbf{w} = \sum_j \alpha_j y_j \mathbf{x}_j$$

$$\frac{\partial L}{\partial b} = -\sum_j \alpha_j y_j \quad \rightarrow \quad \sum_j \alpha_j y_j = 0$$

Substituting these values back in (and simplifying), we obtain:

(Dual) $$\max_{\vec{\alpha} \geq 0,\ \sum_j \alpha_j y_j = 0} \sum_j \alpha_j - \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j \, (\vec{x}_i \cdot \vec{x}_j)$$

So, in dual formulation we will solve for α directly!

- **w** and b are computed from α (if needed)

https://www.robots.ox.ac.uk/~az/lectures/ml/lect3.pdf

Linear SVM dual formulation

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1)$$

Optimality: $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \qquad \sum_{i=1}^{n} \alpha_i \, y_i = 0$

$$\begin{aligned}\mathcal{L}(\alpha) &= \underbrace{\tfrac{1}{2} \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_j \alpha_i y_i y_j \mathbf{x}_j^\top \mathbf{x}_i}_{\mathbf{w}^\top \mathbf{w}} - \sum_{i=1}^{n} \alpha_i y_i \underbrace{\sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j^\top}_{\mathbf{w}^\top} \mathbf{x}_i - b \underbrace{\sum_{i=1}^{n} \alpha_i y_i}_{=0} + \sum_{i=1}^{n} \alpha_i \\ &= -\frac{1}{2} \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_j \alpha_i y_i y_j \mathbf{x}_j^\top \mathbf{x}_i + \sum_{i=1}^{n} \alpha_i\end{aligned}$$

Dual linear SVM is also a quadratic program

$$\text{problem } \mathcal{D} \quad \begin{cases} \min\limits_{\alpha \in \mathbf{R}^n} & \frac{1}{2}\alpha^\top G \alpha - \mathbf{e}^\top \alpha \\ \text{with} & \mathbf{y}^\top \alpha = 0 \\ \text{and} & 0 \leq \alpha_i \qquad\qquad i = 1, n \end{cases}$$

with G a symmetric matrix $n \times n$ such that $G_{ij} = y_i y_j \mathbf{x}_j^\top \mathbf{x}_i$

$$= -\frac{1}{2} \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_j \alpha_i y_i y_j \mathbf{x}_j^\top \mathbf{x}_i + \sum_{i=1}^{n} \alpha_i$$

$$\phi(\mathbf{x}) = (x_1^2, x_1x_2, x_1x_3, x_2x_1, x_2^2, x_2x_3, x_3x_1, x_3x_2, x_3^2)^T$$
$$\phi(\mathbf{y}) = (y_1^2, y_1y_2, y_1y_3, y_2y_1, y_2^2, y_2y_3, y_3y_1, y_3y_2, y_3^2)^T$$

KERNEL TRICK

$$\phi(\mathbf{x})^T\phi(\mathbf{y}) = \sum_{i,j=1}^{3} x_i x_j y_i y_j$$

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y})^2$$

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

https://towardsdatascience.com/understanding-support-vector-machine-part-2-kernel-trick-mercers-theorem-e1e6848c6c4d

https://medium.com/@zxr.nju/what-is-the-kernel-trick-why-is-it-important-98a98db0961d

- More formally, if we have data $\mathbf{x}, \mathbf{z} \in X$ and a map $\phi : X \to \Re^N$ then

$$k(\mathbf{x}, \mathbf{z}) = \left\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \right\rangle$$

  is a kernel function

$$
\begin{aligned}
e^{\frac{-1}{2\sigma^2}(x_i - x_j)^2} &= e^{\frac{-x_i^2 - x_j^2}{2\sigma^2}} \left(1 + \frac{2x_i x_j}{1!} + \frac{(2x_i x_j)^2}{2!} + ...\right) \qquad (1.25) \\
&= e^{\frac{-x_i^2 - x_j^2}{2\sigma^2}} \left(1 \cdot 1 + \sqrt{\frac{2}{1!}} x_i \cdot \sqrt{\frac{2}{1!}} x_j + \sqrt{\frac{(2)^2}{2!}} (x_i)^2 \cdot \sqrt{\frac{(2)^2}{2!}} (x_j)^2 + ...\right) \\
&= \phi(x_i)^T \phi(x_j) \\
\text{where, } \phi(x) &= e^{\frac{-x^2}{2\sigma^2}} \left(1, \sqrt{\frac{2}{1!}} x, \sqrt{\frac{2^2}{2!}} x^2, ...\right)
\end{aligned}
$$

Classifier:

$$f(\mathbf{x}) = \sum_i^N \alpha_i y_i \, \mathbf{x}_i{}^\top \mathbf{x} + b$$

$$\rightarrow f(\mathbf{x}) = \sum_i^N \alpha_i y_i \, \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}) + b$$

Learning:

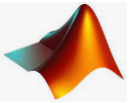
$$\max_{\alpha_i \geq 0} \sum_i \alpha_i - \frac{1}{2} \sum_{jk} \alpha_j \alpha_k y_j y_k \mathbf{x}_j{}^\top \mathbf{x}_k$$

$$\rightarrow \max_{\alpha_i \geq 0} \sum_i \alpha_i - \frac{1}{2} \sum_{jk} \alpha_j \alpha_k y_j y_k \Phi(\mathbf{x}_j)^\top \Phi(\mathbf{x}_k)$$

subject to

$$0 \leq \alpha_i \leq C \text{ for } \forall i, \text{ and } \sum_i \alpha_i y_i = 0$$

$$f(\mathbf{x}) = \sum_i^N \alpha_i y_i \, k(\mathbf{x}_i, \mathbf{x}) + b$$



Polynomial kernel

$$k(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} \cdot \mathbf{x_j} + 1)^d$$

Polynomial kernel equation

Gaussian kernel

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

Gaussian kernel equation

Laplace RBF kernel

$$k(x, y) = \exp\left(-\frac{\|x - y\|}{\sigma}\right)$$

Laplace RBF kernel equation

Hyperbolic tangent kernel

$$k(\mathbf{x_i}, \mathbf{x_j}) = \tanh(\kappa \mathbf{x_i} \cdot \mathbf{x_j} + c)$$

Hyperbolic tangent kernel equation

Bessel function

$$k(x, y) = \frac{J_{v+1}(\sigma\|x - y\|)}{\|x - y\|^{-n(v+1)}}$$

Equation of Bessel function of the first kind kernel

(RBF)

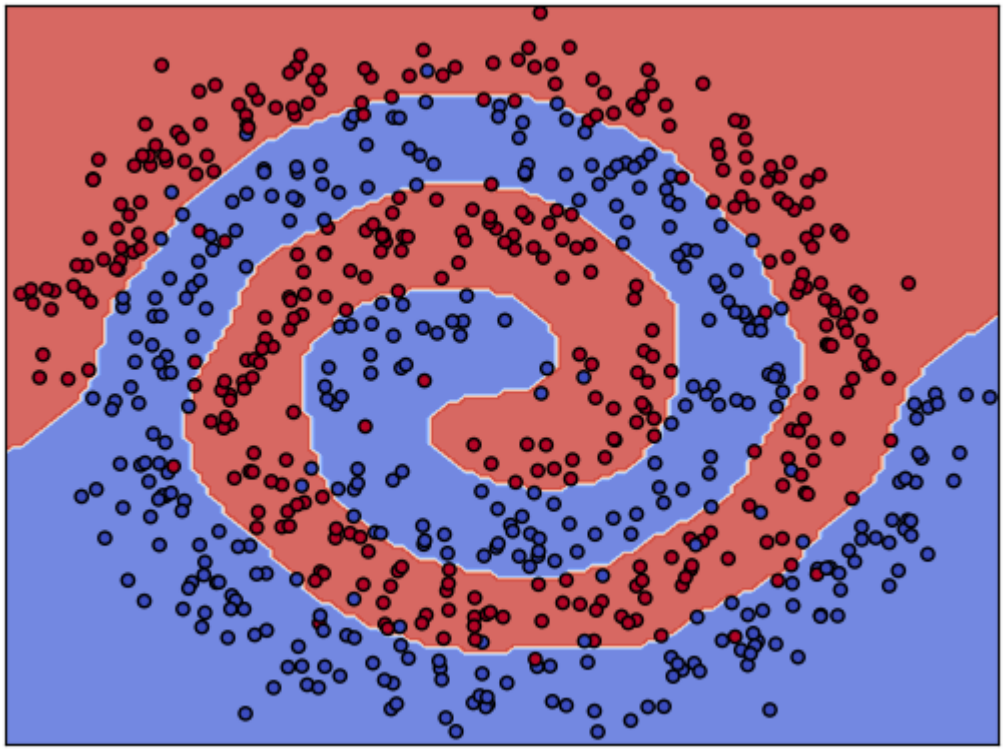
$$k(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\gamma\|\mathbf{x_i} - \mathbf{x_j}\|^2)$$

Gaussian radial basis function (RBF)

Gaussian kernel

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

Gaussian kernel equation

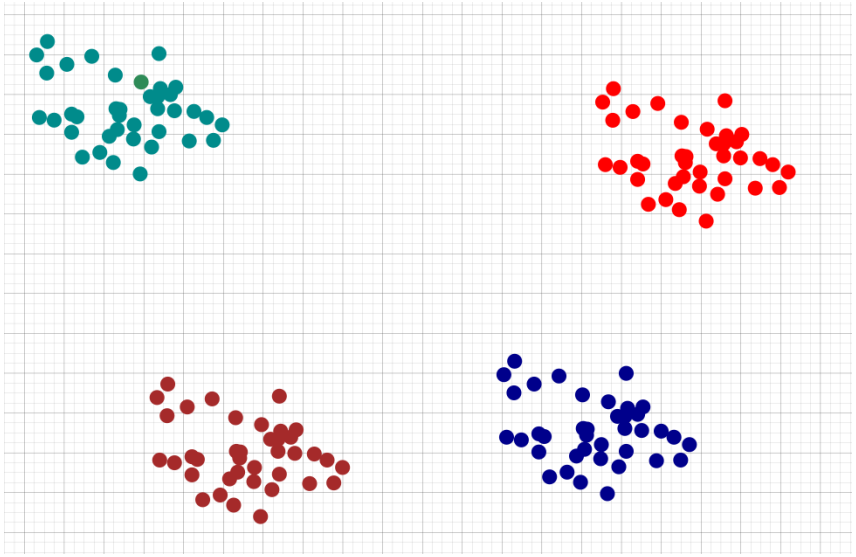


(RBF)

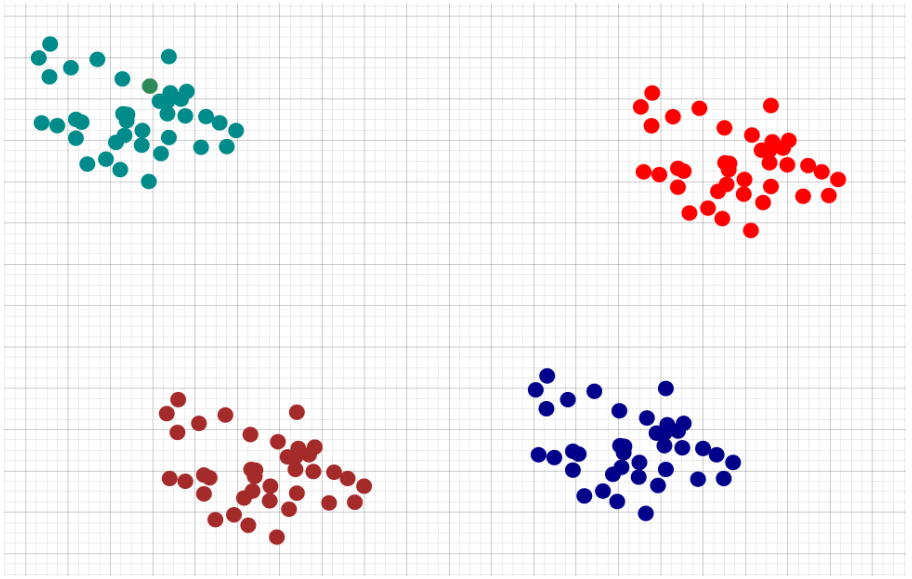
$$= \exp(-\gamma\|\mathbf{x_i} - \mathbf{x_j}\|^2)$$

radial basis function (RBF)

One vs. All (OVA)

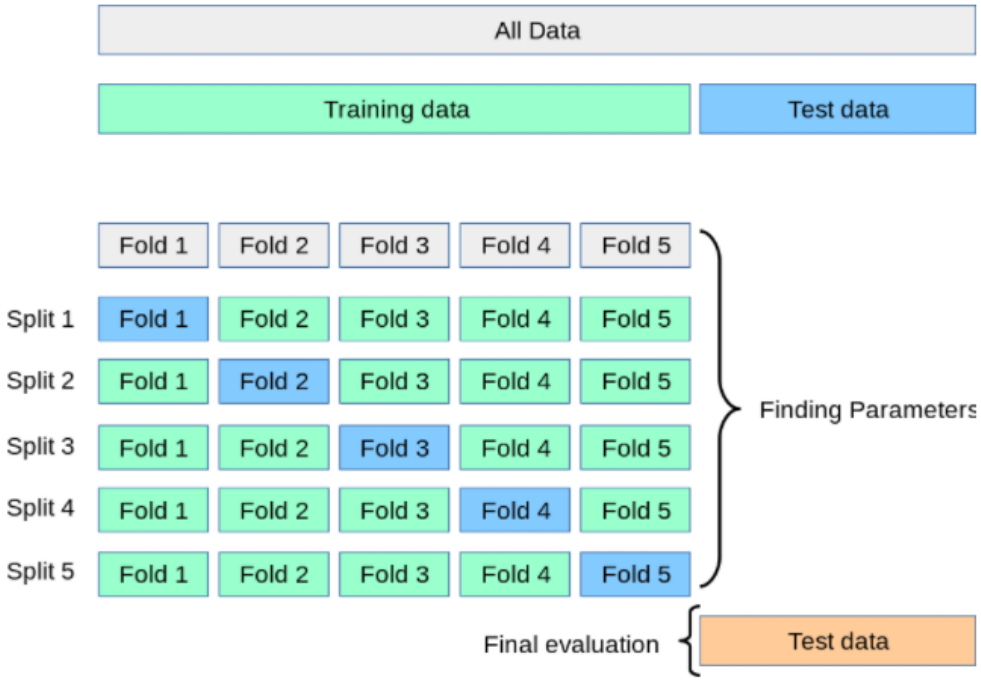


One vs. Ove (OVO)



Üst Parametrelerin Belirlenmesi

An alternative and smarter approach involves estimating the test error rate by holding out a subset of the `training set` from the fitting process. This subset, called the `validation set`, can be used to select the appropriate level of flexibility of our algorithm! There are different validation approaches that are used in practice, and we will be exploring one of the more popular ones called **k-fold cross validation**.



https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

Parameters:

C : *float, default=1.0*

Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty.

kernel : *{'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'}, default='rbf'*

Specifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable. If none is given, 'rbf' will be used. If a callable is given it is used to pre-compute the kernel matrix from data matrices; that matrix should be an array of shape `(n_samples, n_samples)`.

degree : *int, default=3*

Degree of the polynomial kernel function ('poly'). Ignored by all other kernels.

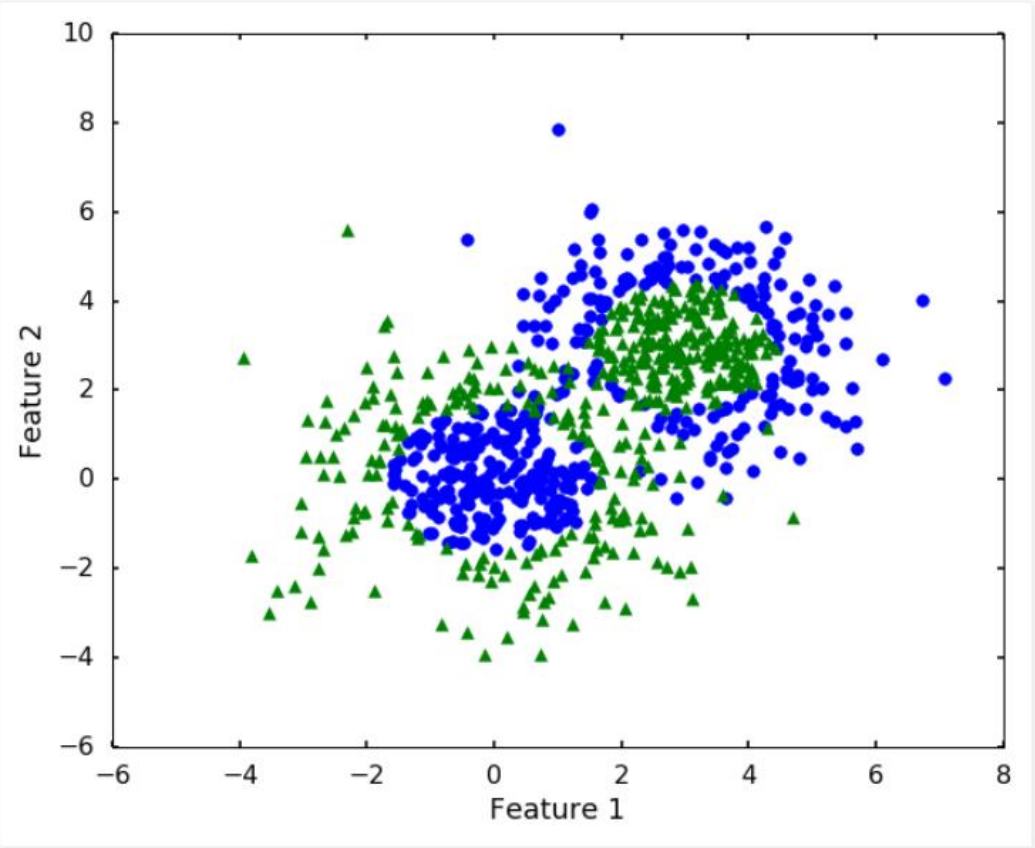
gamma : *{'scale', 'auto'} or float, default='scale'*

Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.

- if `gamma='scale'` (default) is passed then it uses 1 / (n_features * X.var()) as value of gamma,
- if 'auto', uses 1 / n_features.

decision_function_shape : *{'ovo', 'ovr'}, default='ovr'*

Whether to return a one-vs-rest ('ovr') decision function of shape (n_samples, n_classes) as all other classifiers, or the original one-vs-one ('ovo') decision function of libsvm which has shape (n_samples, n_classes * (n_classes - 1) / 2). However, one-vs-one ('ovo') is always used as multi-class strategy. The parameter is ignored for binary classification.

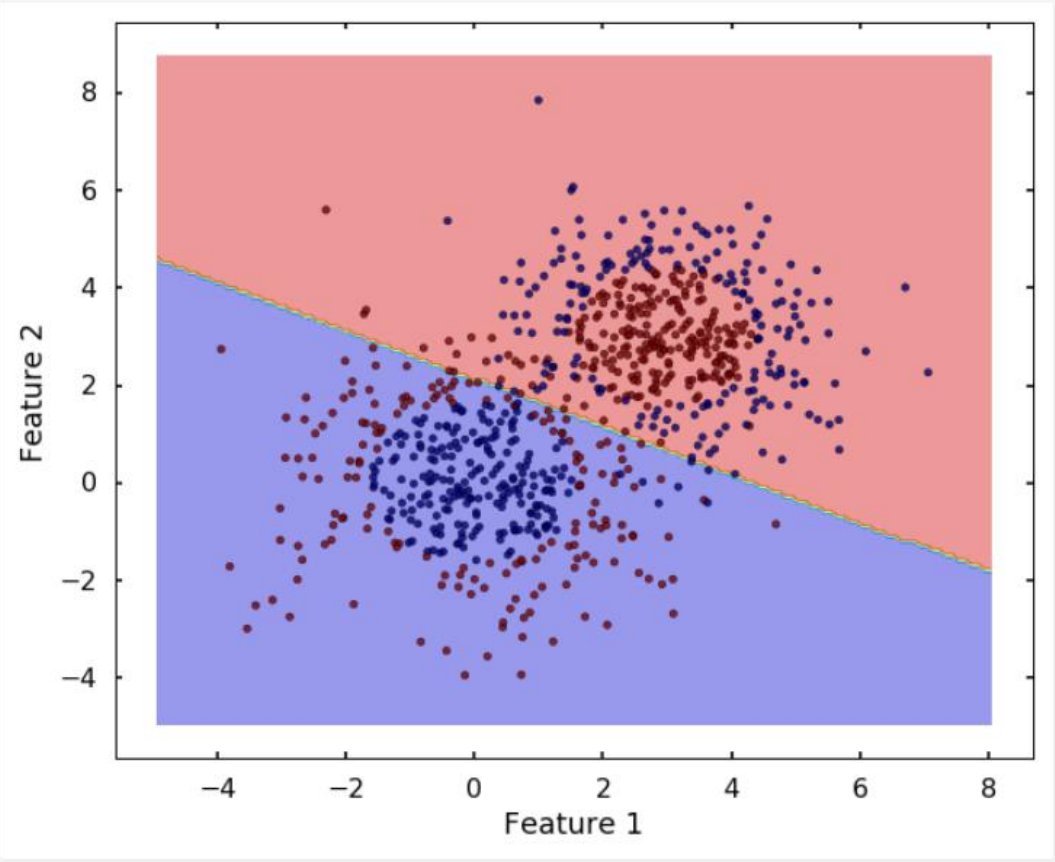



```
clf = svm.SVC(kernel='linear')
clf.fit(X,y)
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma='auto', kernel='linear',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

```
plot_desicion_boundary(X, y, clf)
```

http://qingkaikong.blogspot.com/2016/12/machine-learning-8-support-vector.html

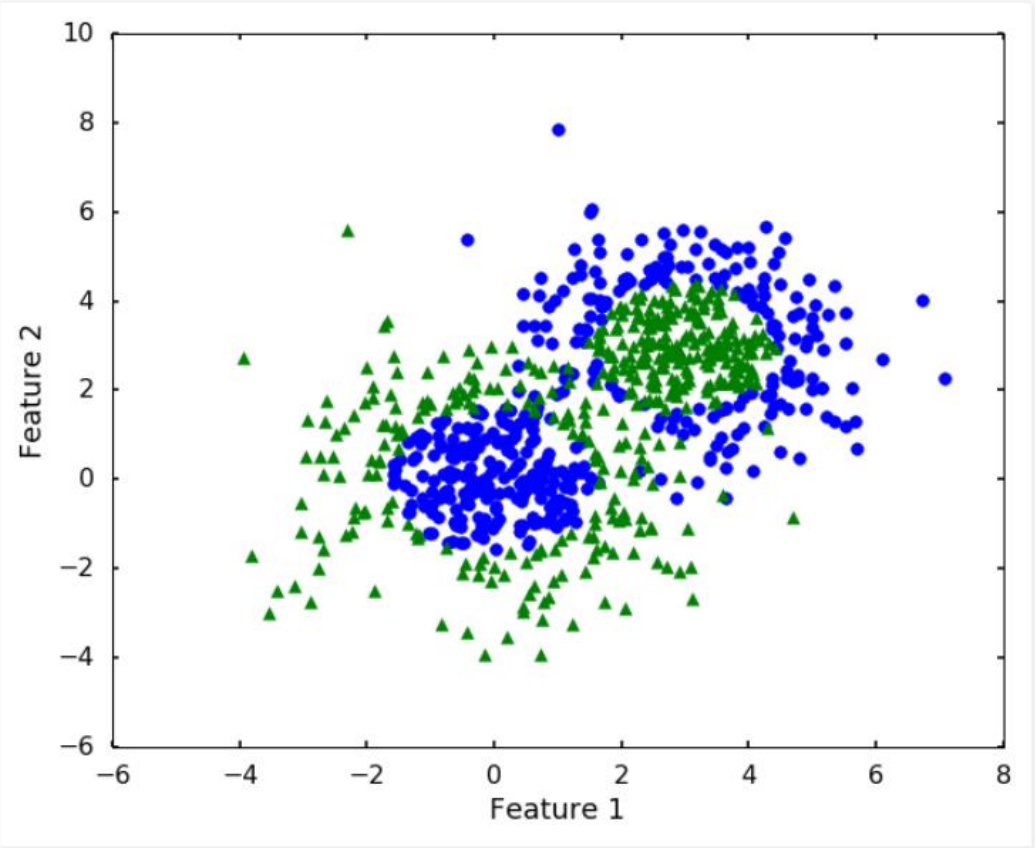



```
clf = svm.SVC(kernel='linear')
clf.fit(X,y)
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma='auto', kernel='linear',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

```
plot_desicion_boundary(X, y, clf)
```

http://qingkaikong.blogspot.com/2016/12/machine-learning-8-support-vector.html

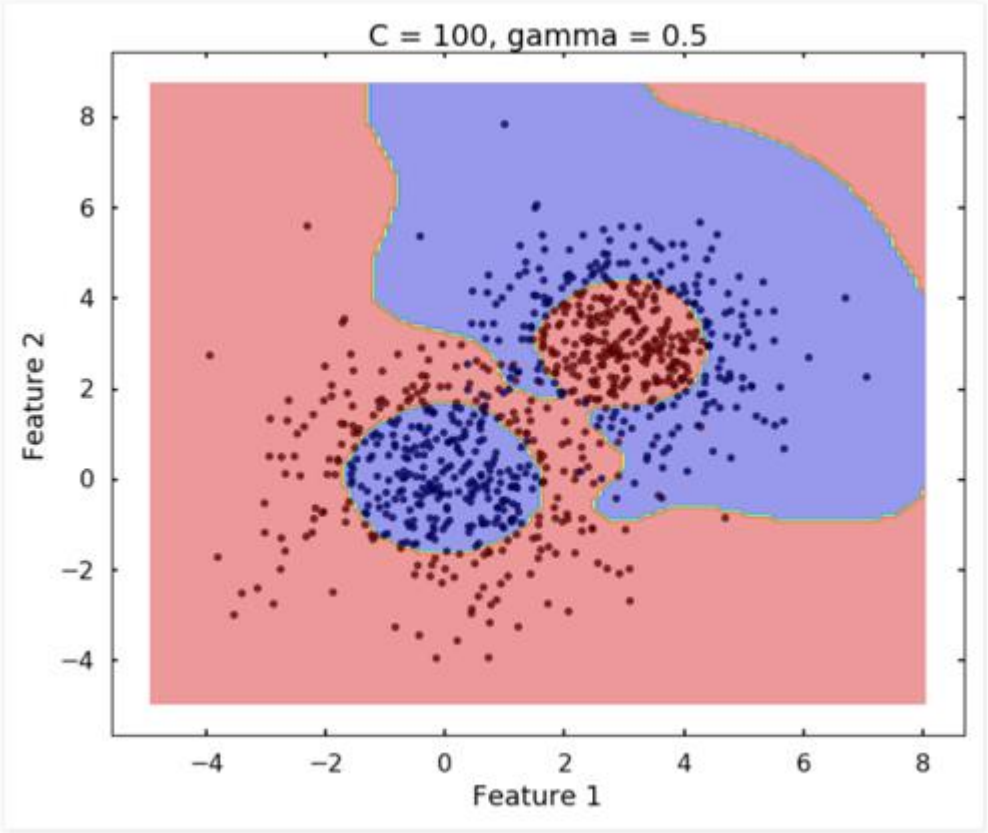



```
clf = svm.SVC(kernel='rbf')
clf.fit(X,y)
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

```
plot_desicion_boundary(X, y, clf)
```

http://qingkaikong.blogspot.com/2016/12/machine-learning-8-support-vector.html

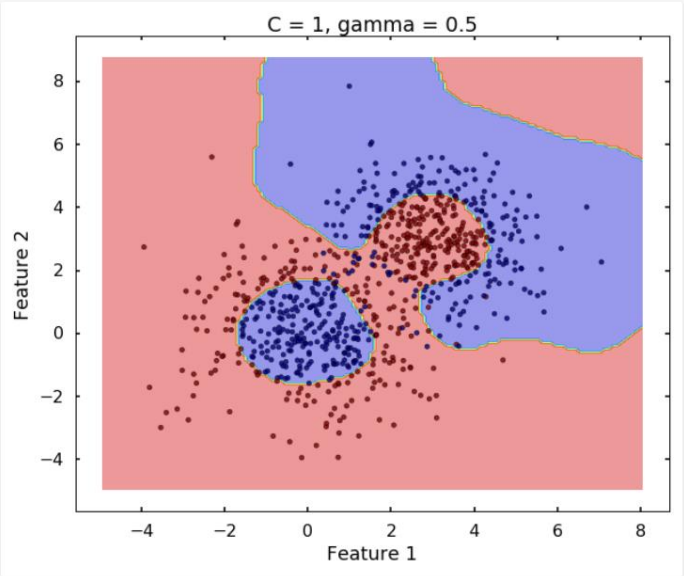


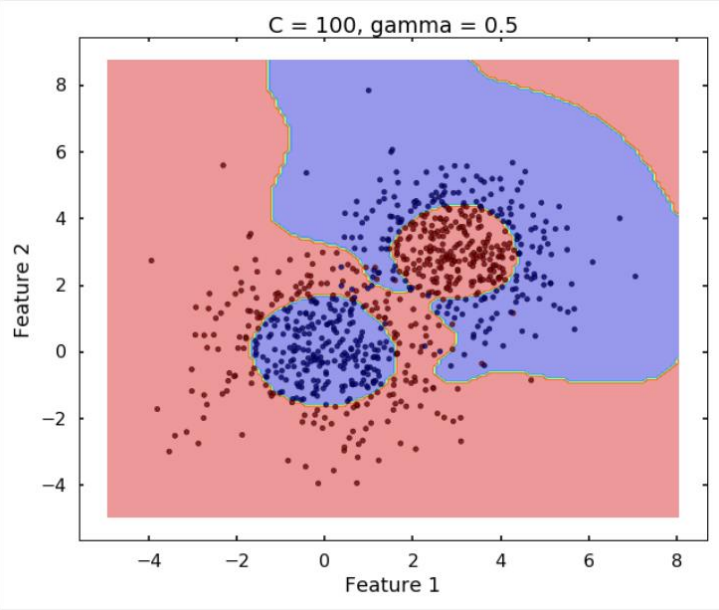

```
clf = svm.SVC(kernel='rbf')
clf.fit(X,y)
```

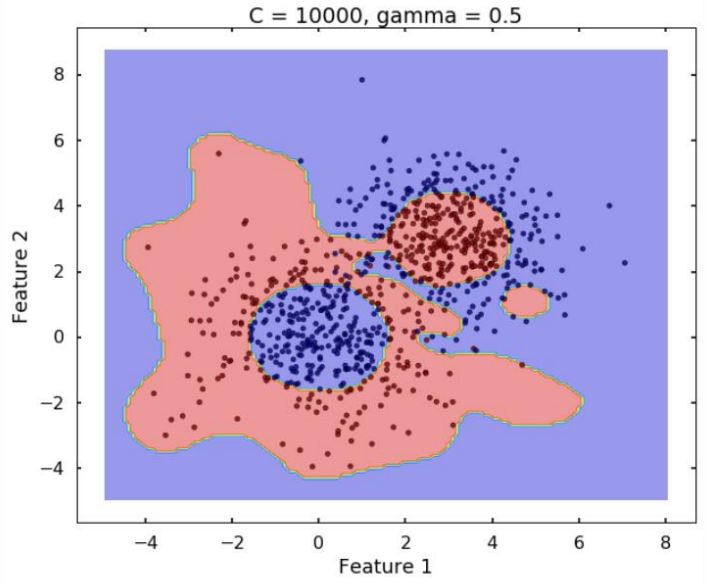
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

```
plot_desicion_boundary(X, y, clf)
```

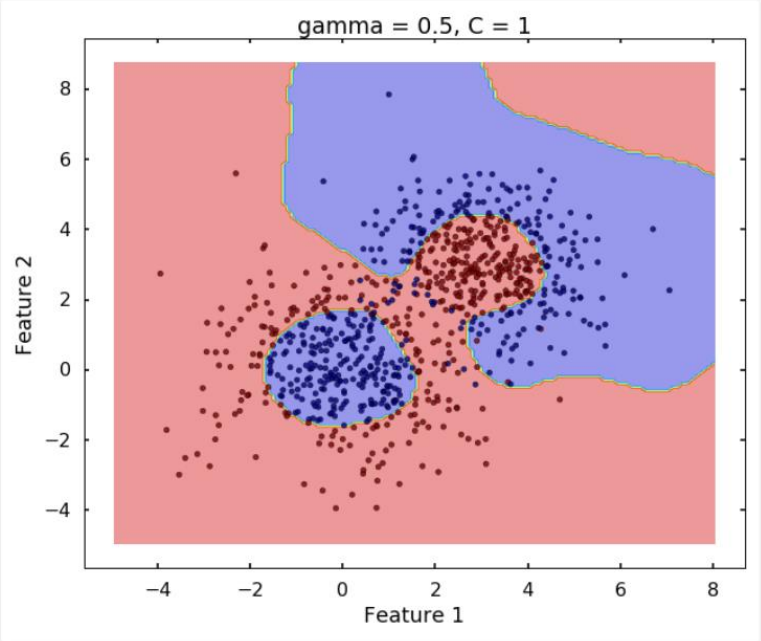
http://qingkaikong.blogspot.com/2016/12/machine-learning-8-support-vector.html

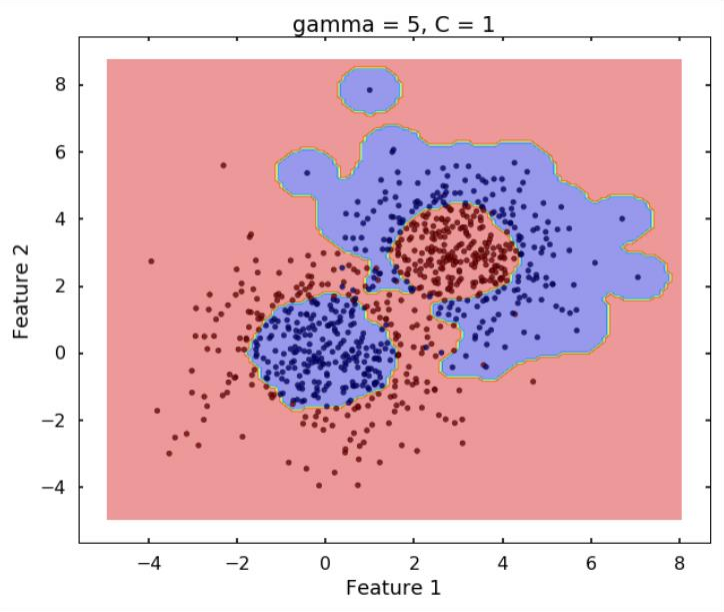


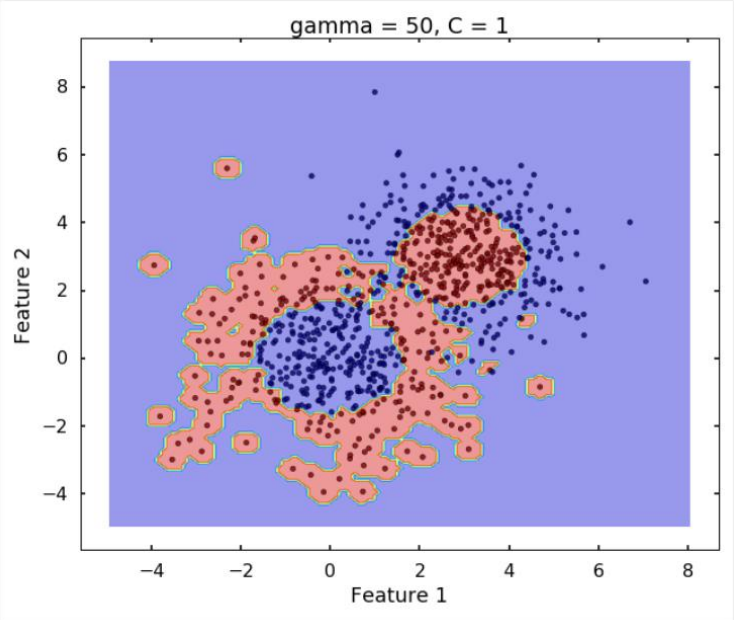




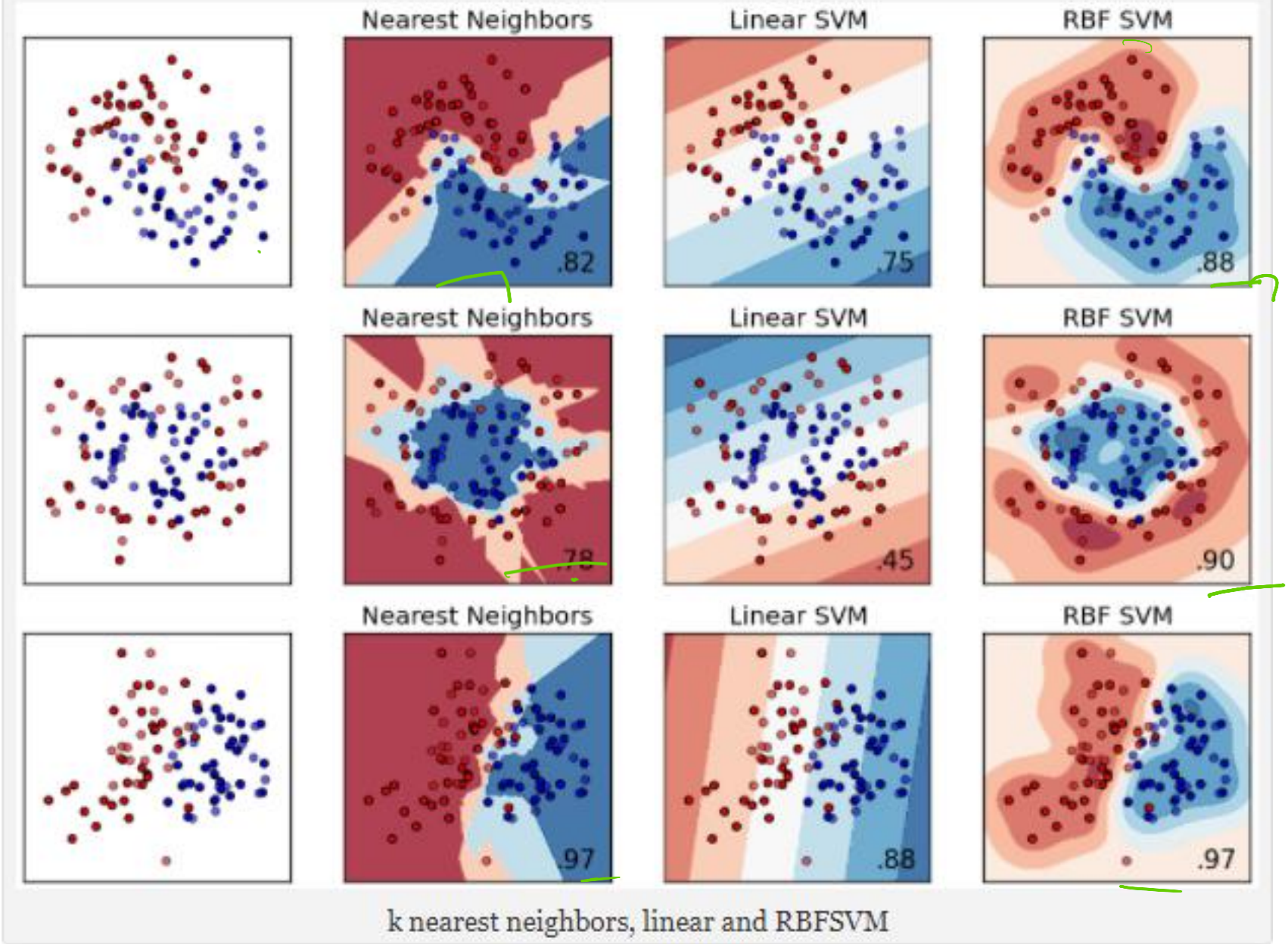
http://qingkaikong.blogspot.com/2016/12/machine-learning-8-support-vector.html





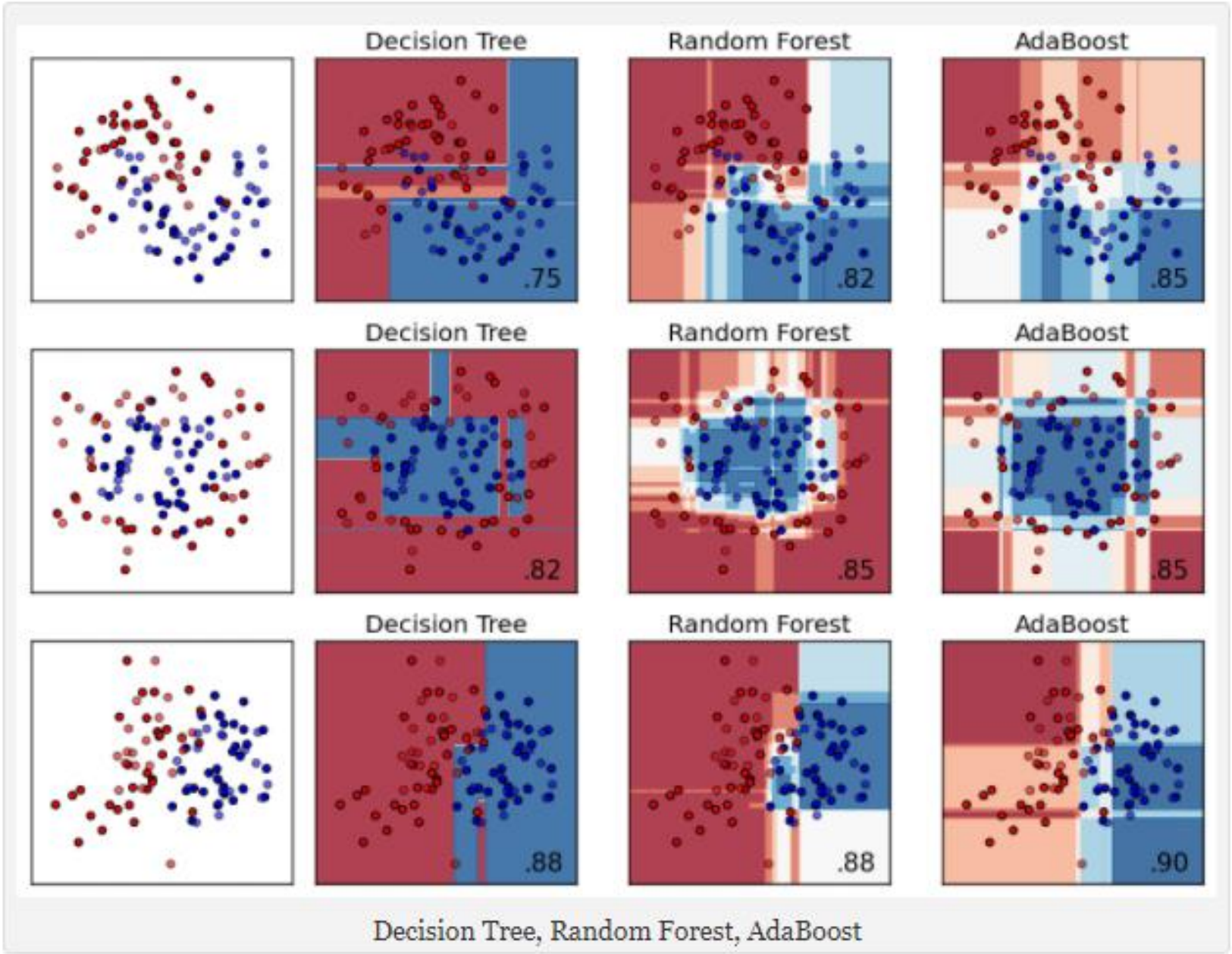


http://qingkaikong.blogspot.com/2016/12/machine-learning-8-support-vector.html



k nearest neighbors, linear and RBFSVM

https://martin-thoma.com/comparing-classifiers/



https://martin-thoma.com/comparing-classifiers/

Kernel

PCA

LDA

REGRESSION

k-NN

ICA

CCA

- Büyük veri
- Homojen olmayan veriler
- Gürültülü veri seti
- Boyut Örnek İkilemi
- Hesaplama yükü

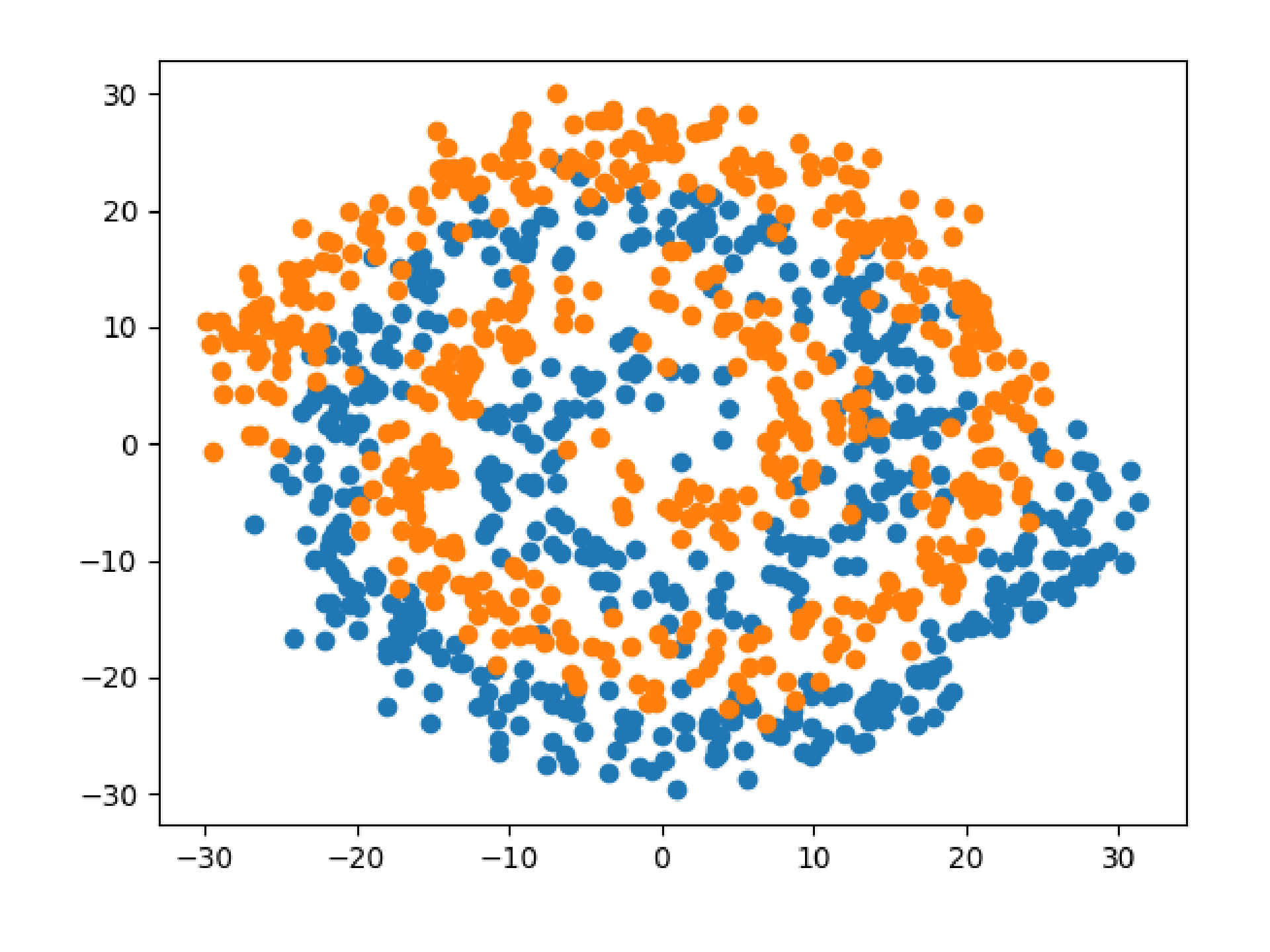
- Büyük veri
- Homojen olmayan veriler
- Gürültülü veri seti
- Boyut Örnek İkilemi
- Hesaplama yükü

- Denetimli
  - Sınıflandırma
    - k-NN
    - Bayes
    - ANN
    - SVM
  - Regresyon
    - Doğrusal
    - ANN
    - SVM
- Denetimsiz

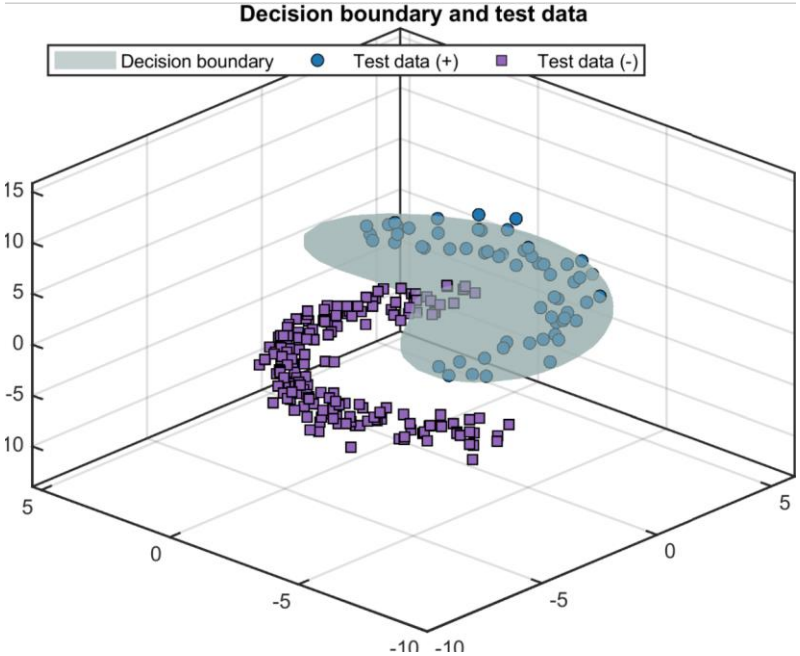
- ACC
- PRE
- RCL
- FSM

- MSE
- MAE
- RME

- Aşırı Öğrenme ve Ezberleme
- Çapraz Doğrulama
- Veri Normalizasyonu


30
20
10
0
−10
−20
−30
−30
−20
−10
0
10
20
30

Anomali Tespiti (SVDD)



https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/0d9fdb82-77ac-4464-b5d8-26c4406a9f84/23303489-d135-4f3c-81b6-e00d6a6ff8a2/images/1620718088.png