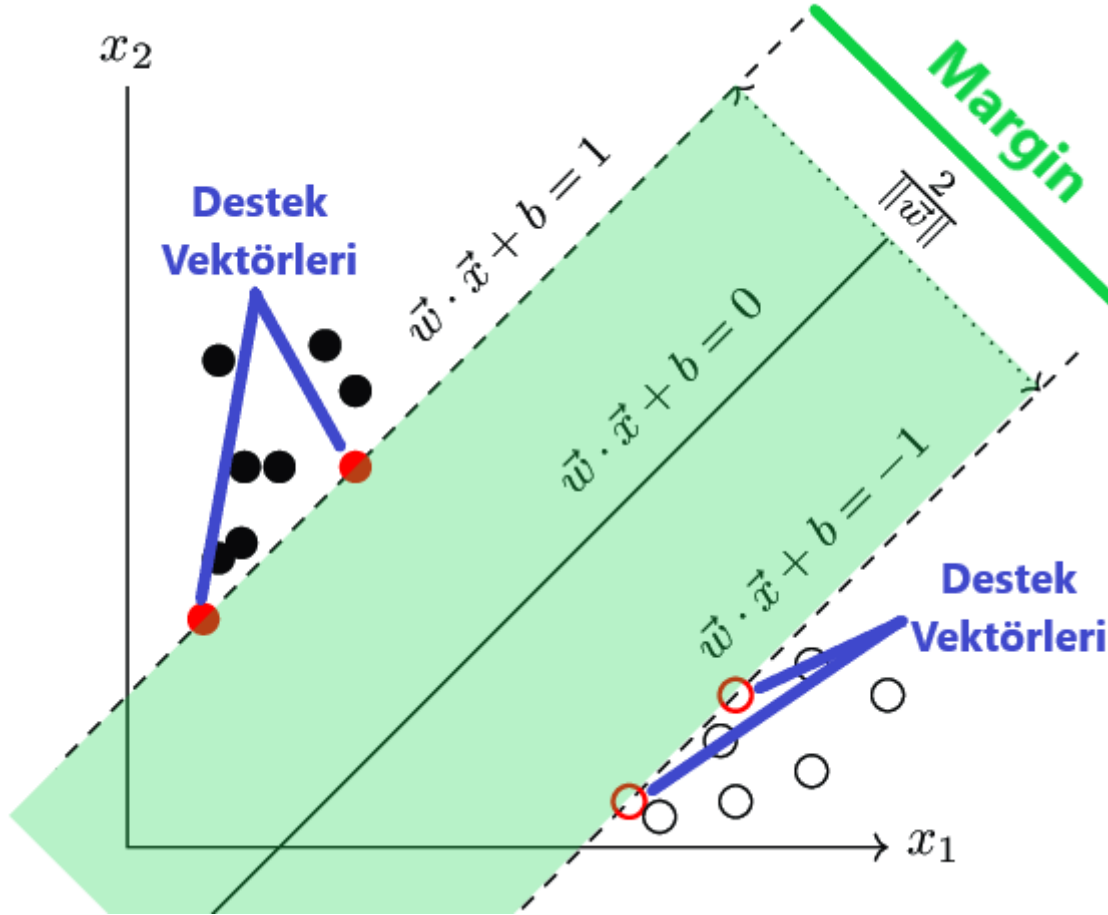


Destek Vektör Makineleri (Support Vector Machine)

genellikle sınıflandırma problemlerinde kullanılan gözetimli öğrenme yöntemlerinden biridir. Bir düzlem üzerine yerleştirilmiş noktaları ayırmak için bir doğru çizer. Bu doğrunun, iki sınıfının noktaları için de maksimum uzaklıkta olmasını amaçlar. Karmaşık ama küçük ve orta ölçekteki veri setleri için uygundur.

Daha açıklayıcı olması için görsel üzerinde tekrar inceleyelim.



Tabloda siyahlar ve beyazlar olmak üzere iki farklı sınıf var. Sınıflandırma problemlerindeki asıl amacımız gelecek verinin hangi sınıfta yer alacağını karar vermektir. Bu sınıflandırmayı yapabilmek için iki sınıfı ayıran bir doğru çizilir ve bu doğrunun ± 1 'i arasında kalan yeşil bölgeye Margin adı verilir. Margin ne kadar geniş ise iki veya daha fazla sınıf o kadar iyi ayrıştırılır.

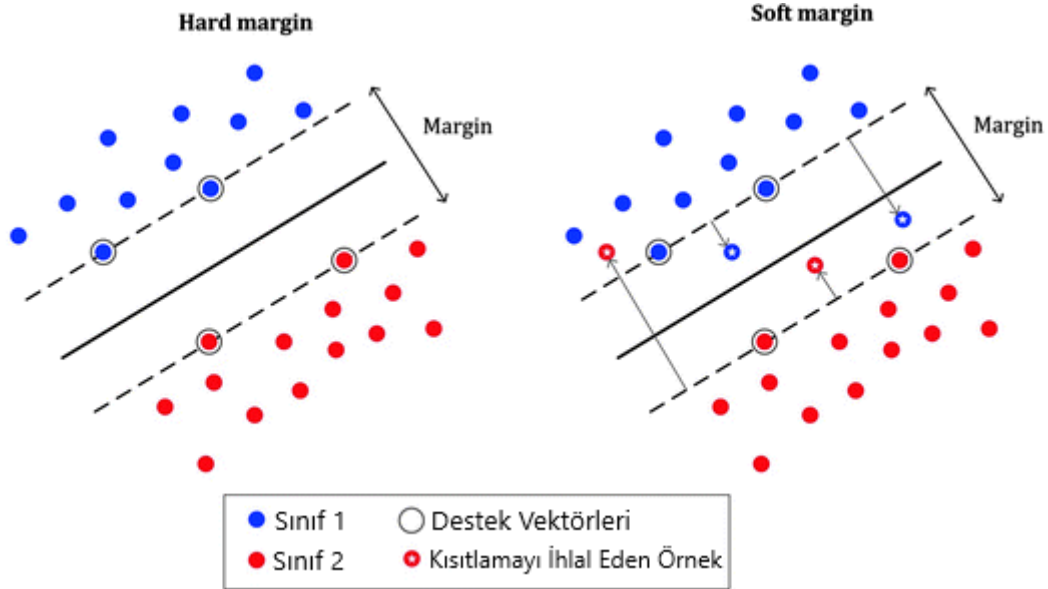
Formüle bakacak olursak:

$$\hat{y} = \begin{cases} 0 & \text{if } \mathbf{w}^T \cdot \mathbf{x} + b < 0, \\ 1 & \text{if } \mathbf{w}^T \cdot \mathbf{x} + b \geq 0 \end{cases}$$

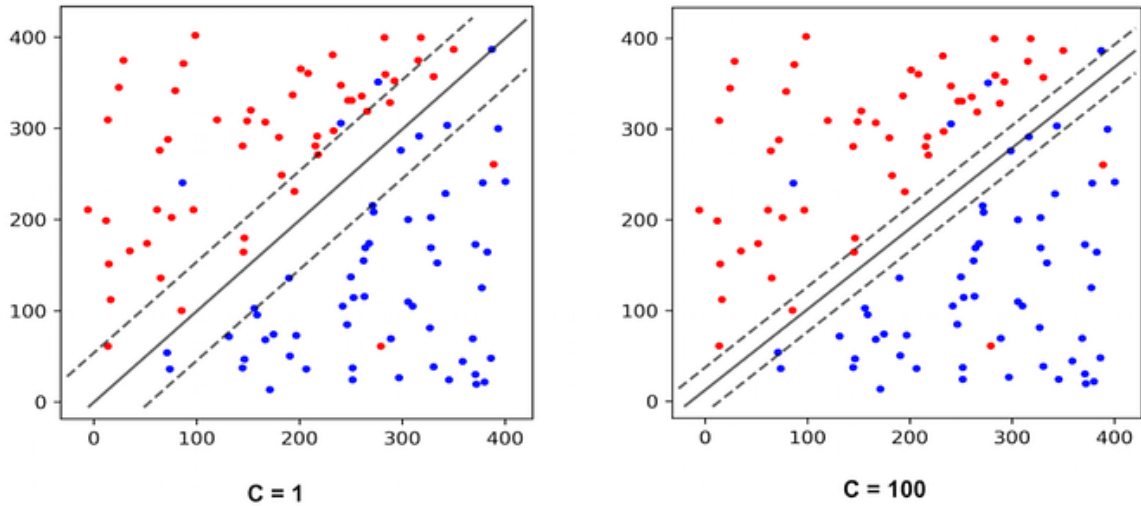
Aslında deęiřen pek bir řey yok. w ; aęırlık vektörü (θ_1), x ; girdi vektörü, b ; sapmadır (θ_0). Yeni bir deęer iin ıkan sonu 0'dan kkse, beyaz noktalara daha yakın olacaktır. Tam tersi, ıkan sonu 0'a eřit veya bkse, bu durumda siyah noktalara daha yakın olacaktır.

Hard Margin vs Soft Margin

Marginimiz her zaman bu řekilde olmayabilir. Bazen rneklerimiz Margin blgesine girebilir. Buna Soft Margin denir. Hard Margin, verimiz doęrusal olarak ayrılabilirse alıřır ve aykırı deęerlere karřı ok duyarlıdır. Bu yzden bazı durumlarda Soft Margin'i tercih etmemiz gerekebilir.



İkisi arasındaki dengeyi SVM ierisindeki C hiperparametresi ile kontrol edebiliriz. C ne kadar bkse Margin o kadar dardır.



Ayrıca model overfit olursa C 'yi azaltmamız gerekir.

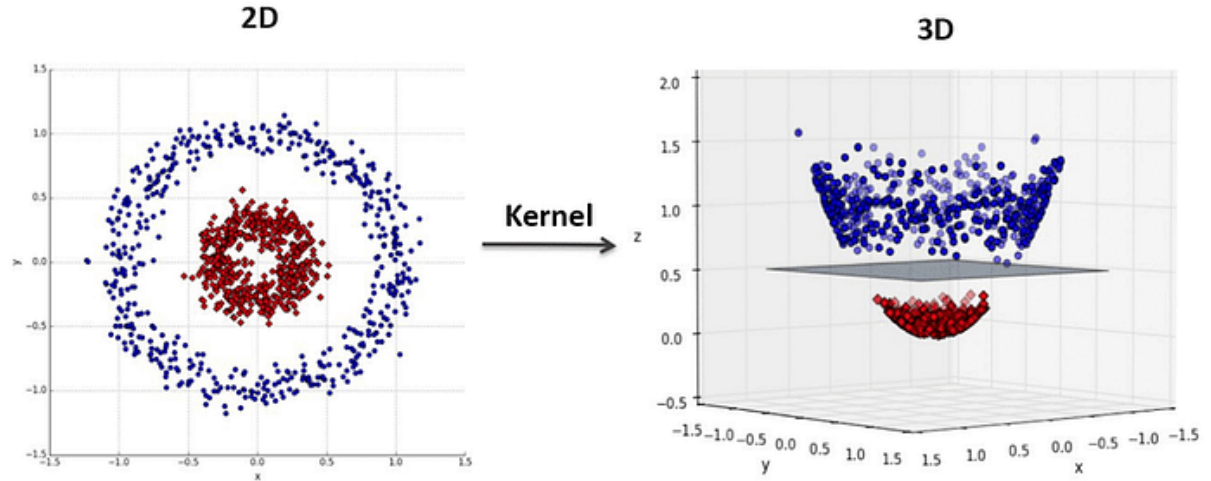
Kernel Trick

Dřk boyutlar karmařık veri setlerini aıklamada yeterli olmayabilir. Boyutu arttırsak iřlemler artacağı iin ok uzun srer. İřte Kernel Trick burada devreye giriyor. Elimizdeki koordinatları belirli Kernel Fonksiyonları ile arparak ok daha anlamlı hale getirebiliyoruz.

Bu yöntemlerden en çok kullanılan ikisini detaylı olarak açıklayacağım.

1-) Polynomial Kernel:

Bu yöntemde problemimizi çözmek için 2 boyuttan çıkıp 3 veya daha fazlası boyutta işlem yapıyormuş gibi hareket ediyoruz.

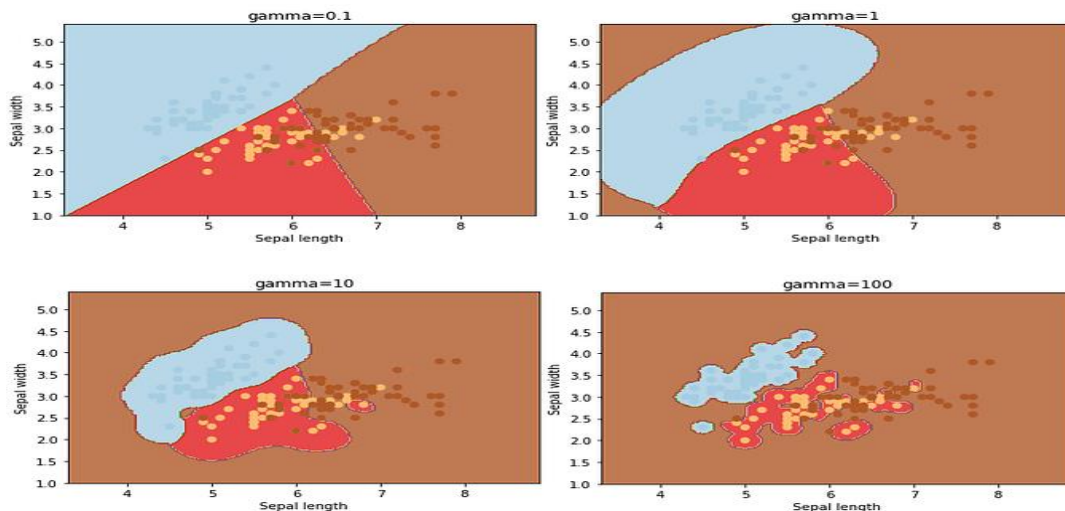


Soldaki (2 boyut) dağılımı bir doğru ile sınıflandıramayız. Bunun için bu gibi problemlerde Polynomial Kernel'i kullanabiliriz. 3. boyutta işlem yaparken sınıflara ayırmak için doğru yerine bir düzlem kullanılırız ve çok daha düzgün bir şekilde sınıflandırabiliriz.

Modelimiz overfit olmuşsa derecesini düşürmeniz, underfit olmuşsa derecesini yükseltmeniz gerekir. Ayrıca coef0 hiperparametresiyle modelinizin yüksek dereceli denklemlerden ne kadar etkileneceğini ayarlayabilirsiniz (sadece 'poly' ve 'sigmoid' kernelda etkili olur).

2-) Gaussian RBF (Radial Basis Function) Kernel:

Anlaması biraz güç olabilir ama sonsuz boyuttaki Destek Vektör Makinelerini bulur ve her bir noktanın belirli bir noktaya ne kadar benzediğini normal dağılım ile hesaplar, ona göre sınıflandırır. Dağılımın genişliğini gamma hiperparametresi ile kontrol ederiz. **Gamma** ne kadar küçükse dağılım o kadar geniş olur. C hiperparametresindeki gibi, model overfit olmuşsa gamma değerini düşürmemiz, model underfit olmuşsa gamma değerini yükseltmemiz gerekir.



Veri setiniz aşırı büyük değilse genellikle RBF Kernel tercih edilir.

ÖZET:

1-) Destek Vektör Makineleri (SVM), düzlem üzerindeki noktaların bir doğru veya hiper düzlem ile ayrıştırılması ve sınıflandırılmasıdır.

2-) Küçük veya orta büyüklükteki veri setleri için uygundur. [Scale'e duyarlıdır](#). Scale edilmesi gerekir.

3-) [Hard Margin ve Soft Margin](#) arasındaki dengeyi C ile kontrol edebiliriz. [C büyüdükçe Margin daralır](#).

4-) Model overfit olursa C'nin azlatılması gerekir.

5-) 2 boyutta açıklanamayan değişimleri boyut arttırarak çözüyormuş gibi yapılan hilelere Kernel Trick denir.

6-) 2 boyutta açıklayamadığımız veri setimizi daha fazla boyutta açıklamak için kullanılan Kernel Trick metoduna [Polynomial Kernel](#) denir.

7-) Model overfit olursa derecesi düşürülür, underfit olursa derece yükseltilir. Coef0 hiperparametresi ile yüksek dereceli denklemlerden ne kadar etkileneceğini ayarlayabilirsiniz.

8-) Her bir noktanın belirli bir noktaya ne kadar benzediğini normal dağılım ile hesaplayan, ona göre sınıflandıran Kernel Trick metoduna RBF Kernel denir.

9-) [Dağılım genişliğini kontrol ettiğimiz gamma değeri ne kadar küçükse dağılım o kadar geniş olur](#). Model overfit olmuşsa gamma değerini düşürmemiz, model underfit olmuşsa gamma değerini yükseltmemiz gerekir.

10-)

