

01.12.2022

Destek Vektör Makineleri Rapor

Recep Onur Okan 20360859027

DESTEK VEKTÖR MAKİNELERİ

Makine Öğrenmesi

Bilgisayarlı görme, örüntü tanıma, sınıflandırma, doğal dil işleme ve biyoinformatik gibi birçok farklı alanda kullanılan Makine Öğrenmesi, giriş verileri ile çıkış verileri arasında doğrusal bir formül ile ifade edilemeyen problemleri çözebilmek için geliştirilmiş güncel esnek hesaplama sistemlerine verilen genel addır.

Çok az sayıda veri ya da yanlış veri ile geliştirilen Makine Öğrenme modellerinin doğruluğu, yeterli miktarda ve doğru veri ile geliştirilen modellere göre oldukça düşüktür. Bu nedenle Makine Öğrenmesi sonucunda elde edilen bilgiler ve başarımlar oranı tamamen giriş verilerine bağlıdır.

Makine Öğrenme modelleri denetimli, denetimsiz ve destekleyici olmak üzere üçe ayrılmasına rağmen, literatürde genel olarak denetimli ve denetimsiz olarak sunulmaktadır.

Makine Öğrenmesi		
Denetimli		Denetimsiz
Sınıflandırma	Regresyon	Kümeleme
Destek Vektör Makineleri	Doğrusal Regresyon	K-Ortalamalar
Karar Ağacı	Temel Bileşen Regresyonu	YSA
Naive Bayes	Rastgele Orman	Hiyerarşik Kümeleme
K- En Yakın Komşu	YSA	Temel Bileşenler Analizi
...

Şekil 2: Makine Öğrenme Modelleri

Destek Vektör Makineleri (Support Vector Machine) genellikle sınıflandırma problemlerinde kullanılan denetimli 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.

Eğer çözülmesi gereken sorun, denetlenen ikili sınıflandırmalardan biri ise SVM(Support Vector Machine) kullanılabilir. Yani, görünmeyen yeni nesneleri(kategorisi bilinmeyen), özelliklerine ve zaten kategorize edilmiş bir dizi bilinen örneğe göre iki ayrı gruba ayırmak istediğimizde SVM algoritmalarını kullanabiliriz.

Böyle bir sisteme iyi bir örnek, daha önce pozitif veya negatif olarak sınıflandırılmış diğer belgelere dayanarak bir dizi yeni belgeyi pozitif veya negatif olarak gruplarına göre sınıflandırmaktır. Yada benzer şekilde, mail kutusuna gelen e-postaları, zaten insanlar tarafından spam veya spam olmayan olarak işaretlenmiş büyük bir belge topluluğuna göre spam veya spam olmayan olarak sınıflandırabiliriz. SVM'ler bu tür durumlar için oldukça kullanışlı algoritmalarlardır.

Bir Destek Vektör Makinesi, her boyutu belirli bir nesnenin “özellik”ini temsil eden sonlu boyutlu bir vektör alanı olan bir özellik alanı oluşturarak durumu modeller. Spam mail veya belge sınıflandırması bağlamında, her “özellik” belirli bir kelimenin yaygınlığı veya o kelimenin ne kadar önemli olduğu ile ilgilidir.

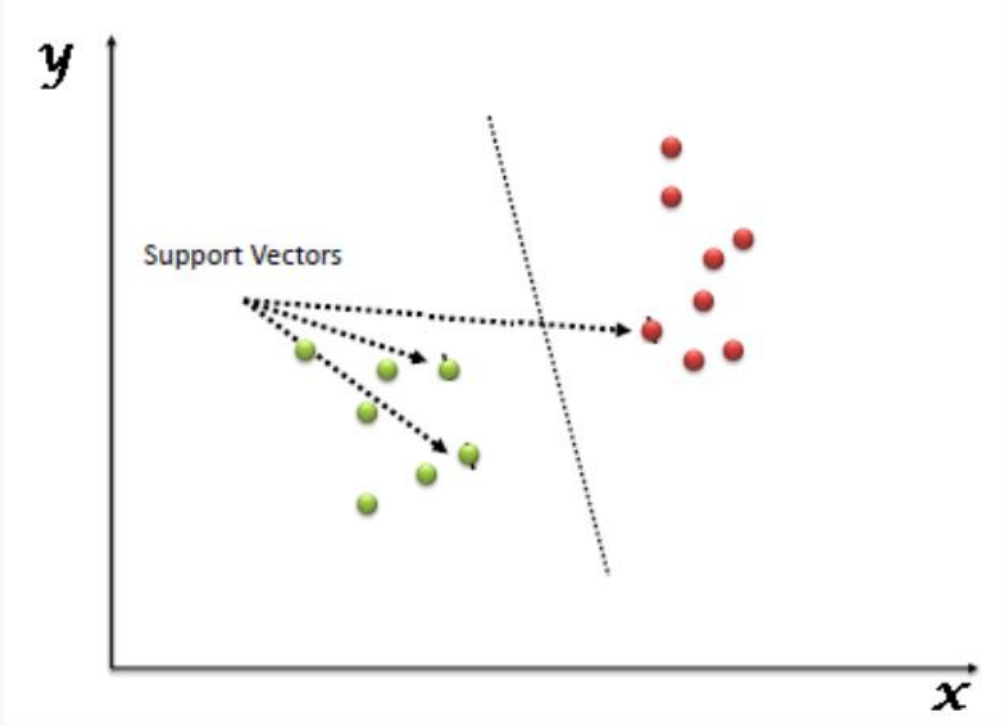
SVM'nin amacı, görünmeyen yeni nesneleri belirli bir kategoriye atayan bir model oluşturmaktır. Bunu, özellik alanının iki kategoriye doğrusal bir bölümünü oluşturarak yapmaya çalışır.Yeni görünmeyen nesnelerdeki özelliklere (örn. Belgeler / e-postalar) dayanarak, bir kategoriye (ör. Spam veya spam olmayan) çoklu ayırma düzleminin (hyperplane) “üstüne” veya “altına” yerleştirir. Bu, olasılık dışı doğrusal bir sınıflandırıcıya bir örnek olur. Olasılıksızdır, çünkü yeni nesnelerdeki özellikler, özellik uzayındaki yerini tam olarak belirler.

SVM'ler Neden Kullanılır;

Destek vektör makineleri (SVM), sınıflandırma ve regresyon analizi için kullanılan verileri analiz eden ilişkili öğrenme algoritmalarına sahip denetimli öğrenme modelleridir.

Eldeki verilerden yeni kalıpları tanımlamak için büyük miktarda veriyi analiz ederler. SVM'ler, aşağıdaki resimde gösterildiği gibi bir veri kümesini en iyi iki sınıfa ayıran bir hiper düzlem bulma amacı ile oluşturulur.

Kullanım alanları olarak ise yüz tanıma,biyoinformatik,metin sınıflandırma gibi birçok alanda kullanılır.



Destek Vektörleri;

- Destek Vektörler sadece bireysel gözlemin koordinatlarıdır.
- Destek Vektör Makinesi, iki sınıfı (düzlem / çizgi) en iyi bir şekilde ayıran sınır çizgileridir.
- Destek vektörleri karar yüzeyine (veya hiper düzleme) en yakın veri noktalarıdır.
- Sınıflandırılması en zor veri noktalarıdır.
- Karar yüzeyinin optimum konumu üzerinde doğrudan etkisi vardır.
- Optimal hiper düzlemin en düşük “kapasiteye” sahip fonksiyon sınıfından kaynaklandığını gösterebiliriz.
- Destek vektörleri hiper düzlemin en yakın veri noktalarıdır, bu sebeple bir veri kümesini bölen hiper düzlemin konumunu değiştirecek noktalardır. Bu nedenle, bir veri kümesinin kritik unsurları olarak kabul edilebilirler.

Hiper düzlem nedir?

Basit bir örnek olarak, yalnızca iki özelliğe sahip bir sınıflandırma görevi için, bir hiper düzlemi bir veri kümesini doğrusal olarak ayıran ve sınıflandıran bir çizgi olarak düşünebiliriz. Şöyle de düşünebiliriz, veri noktalarımız hiper düzlemden ne kadar uzak olursa, doğru sınıflandırıldıklarından o kadar emin oluruz. Bu nedenle, veri noktalarımızın doğrulara hiper düzlemden olabildiğince uzakta olmasını istiyoruz.

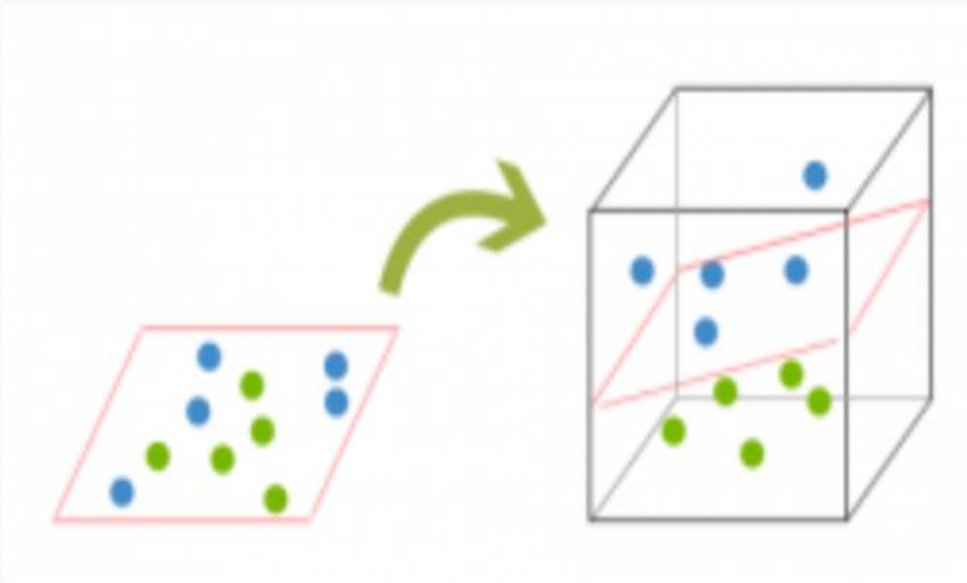
Dolayısıyla, yeni test verileri eklendiğinde, iniş yapan hiper düzlemin herhangi bir tarafı, ona atadığımız sınıfa karar verecektir.

Doğru düzlemi (hyperplane) nasıl bulabiliriz?

Veriler içindeki iki sınıfı en iyi şekilde nasıl birbirinden ayırabiliriz? Düzlem ve her iki kümeden en yakın veri noktası arasındaki mesafe kenar boşluğu olarak bilinir. Amaç, hiper düzlem ile eğitim setindeki herhangi bir nokta arasında mümkün olan en yüksek marja sahip bir hiper düzlem seçmek ve yeni verilerin doğru bir şekilde sınıflandırılma şansını arttırmaktır. Ayrılan bölge içinde asla veri noktası olmaması gerekmektedir.

Eğer net bir hiper düzlem olmazsa ne olur?

Veriler, genellikle yukarıdaki grafikte gösterilen basit örneğimiz kadar temiz değildir. Bir veri kümesi genellikle doğrusal olarak ayrılamayan bir veri kümesini temsil eden karışık elemanlar kümesine benzeyecektir.



Yukarıdaki gibi bir veri kümesini sınıflandırmak için, verilerin 2Boyutlu görünümünden 3B görünümüne geçmek gerekir. Bunu daha basitleştirilmiş bir örnekle açıklayabiliriz. Yukarıdaki iki renkli top setimizin bir kağıda konulduğunu ve bu sayfanın aniden kaldırıldığını ve topları havaya fırlatıldığını düşünün. Toplar havadayken, onları ayırmak için elimizdeki kayıt tabakayı kullanalım. Yani topların bu ‘kaldırılması’ işlemi verilerin daha yüksek bir boyuta eşlenmesini temsil eder. Bu işleme ise *çekirdekleme* (kernel trick) işlemi denilmektedir.

Şimdi üçüncü boyutta olduğumuz için, hiper düzlemimiz artık bir çizgi olamaz. Ve ayırma düzlemi artık yukarıdaki örnekte gösterildiği gibi bir düzlem olmalıdır. Burada bize verilen bilgi, verilerin ayrılması için bir hiper düzlem oluşturulana kadar verilerin daha yüksek boyutlara eşlenmeye devam edileceğidir.

Şimdi bu işlemin nasıl çalıştığına bir bakalım! Ve doğru bir hiper düzlemi nasıl tanımlayabiliriz inceleyelim.

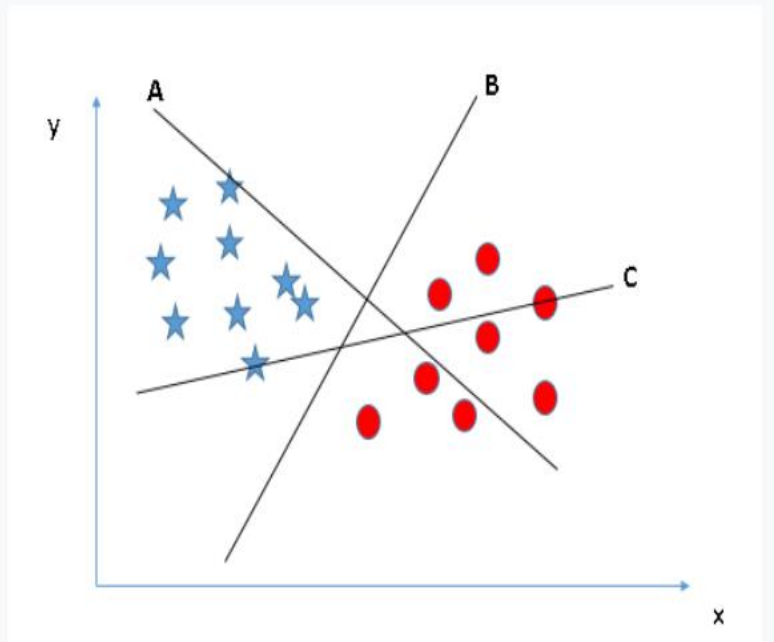
Doğru hiper düzlemi tanımlamak için; “İki sınıfı en iyi şekilde ayıran hiper-düzlem seçilecektir.”(POLYNOMIAL KERNEL)

Yapacağımız işlemleri birkaç farklı yöntem ile örneklendirelim;

Doğru hiper düzlemi tanımlayın (Yöntem-1):

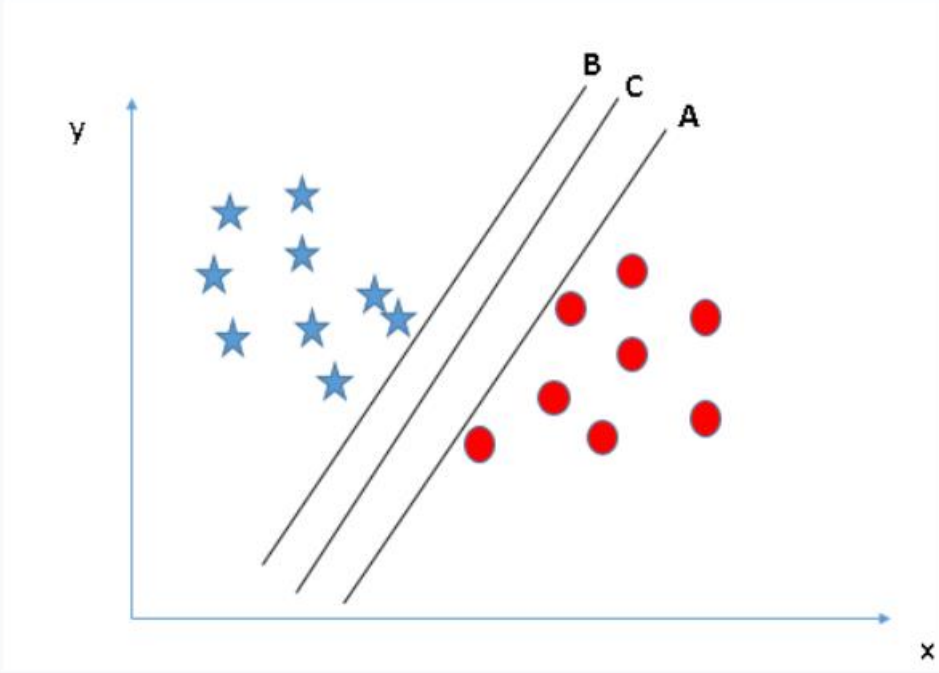
Burada üç hiper düzlemimiz olsun (A, B ve C). Şimdi, yıldızı ve daireyi sınıflandırmak için doğru düzlemi tanımlayalım.

Bu örnekte “B” düzlemi ya da ayırıcı bizim sınıflandırma işlemini mükemmel bir şekilde yerine getirmiştir diyebiliriz.

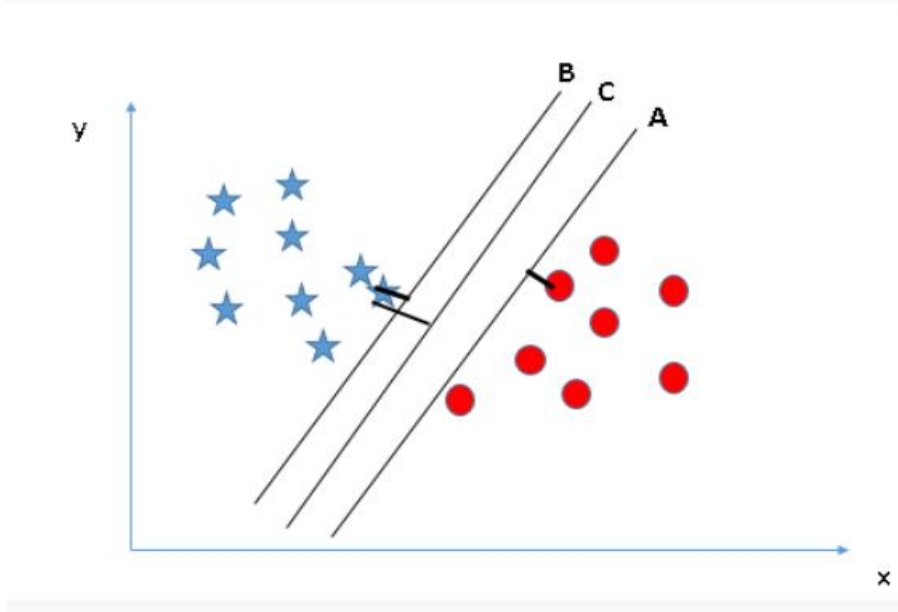


Doğru hiper düzlemi tanımlayın (Yöntem-2):

Burada üç hiper düzlemimiz var (A, B ve C) ve hepsi sınıfları iyi bir şekilde ayırıyor. Şimdi, doğru hiper düzlemi nasıl tanımlayabiliriz?



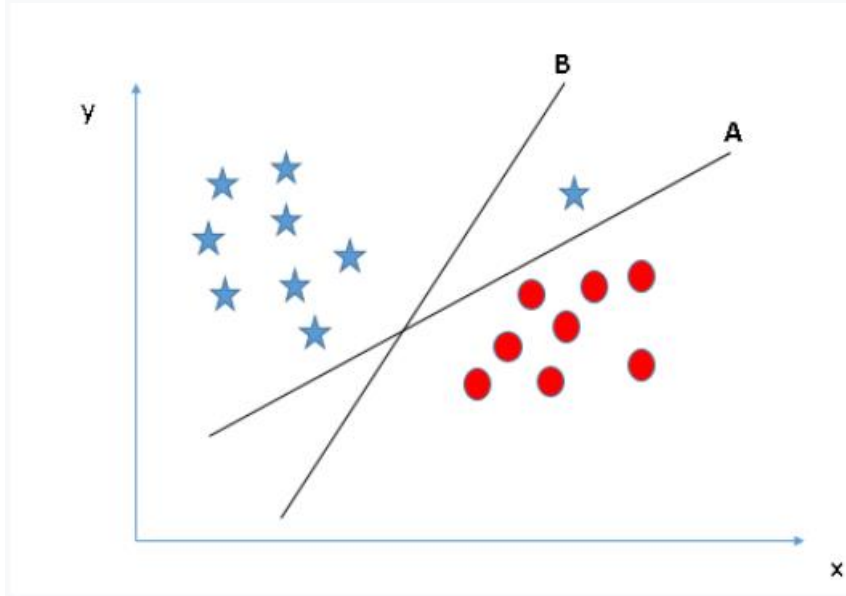
Burada, en yakın veri noktası (sınıflardan biri) ve hiper düzlem arasındaki mesafeleri en üst düzeye çıkarmak doğru hiper düzlemi belirlememize yardımcı olacaktır. Bu mesafeye **Marj mesafesi** denir. Aşağıdaki resme bakalım:



Hiper düzlem C marjının hem A hem de B'ye göre yüksek olduğunu görebiliriz. Bu nedenle, doğru hiper düzlemi C olarak kabul ediyoruz. Yüksek marjlı hiper düzlemi seçmenin bir başka önemli nedeni de sağlamlıktır. Düşük marjlı bir hiper düzlem seçersek, yanlış sınıflandırma olasılığı yüksek olacaktır.

3. Doğru hiper düzlemi tanımlayalım (Yöntem-3)

Bazılarınız hiper-düzlem B'yi A'ya göre daha yüksek marja sahip olduğu için seçmiş olabilir ancak burada hiper düzlem B de bir sınıflandırma hatası olduğunu gözlemliyoruz ve A ayracı farklı türlerin hepsini doğru bir şekilde sınıflandırdığı için doğru hiper düzlem A'dır diyebiliriz.



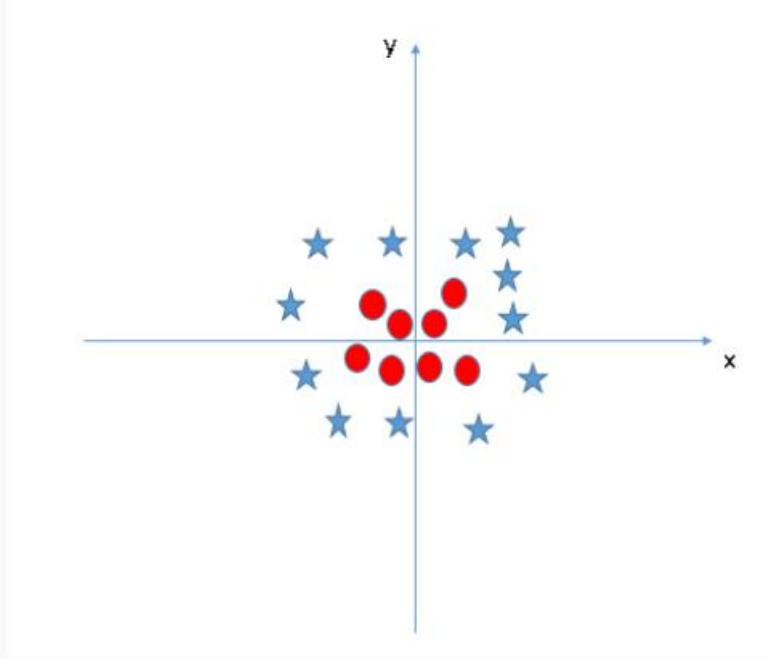
İki sınıfı sınıflandırabilir miyiz?(Yöntem-4)

Yukarıdaki resimde yıldızlardan biri diğer (daire) sınıfın alanında bir aykırı değer olarak yer aldığı için iki sınıfı düz bir çizgi kullanarak ayıramıyoruz. Diğer uçtaki bir yıldız, yıldız sınıfı için bir aykırı değer gibidir. SVM, aykırı değerleri yok sayma ve maksimum kenar boşluğuna sahip hiper düzlemi bulma özelliğine sahiptir. Bu nedenle, SVM'nin aykırı değerlere karşı sağlam olduğunu söyleyebiliriz.

5.Sınıflara ayrılacak hiper düzlemin bulunma işlemi (Yöntem-5)

Aşağıdaki senaryoda, iki sınıf arasında doğrusal ayrım olamaz, bu yüzden SVM bu iki sınıfı nasıl sınıflandırabilir? Şimdiye kadar sadece doğrusal hiper düzlemi inceledik.

SVM bu sorunu da çözebilir. Bu durumu çözmek için ek özellik uygulanır. Burada yeni bir $z = x^2 + y^2$ özelliği ekleyeceğiz. Gelin şimdi veri noktalarını x ve z eksenine çizelim:



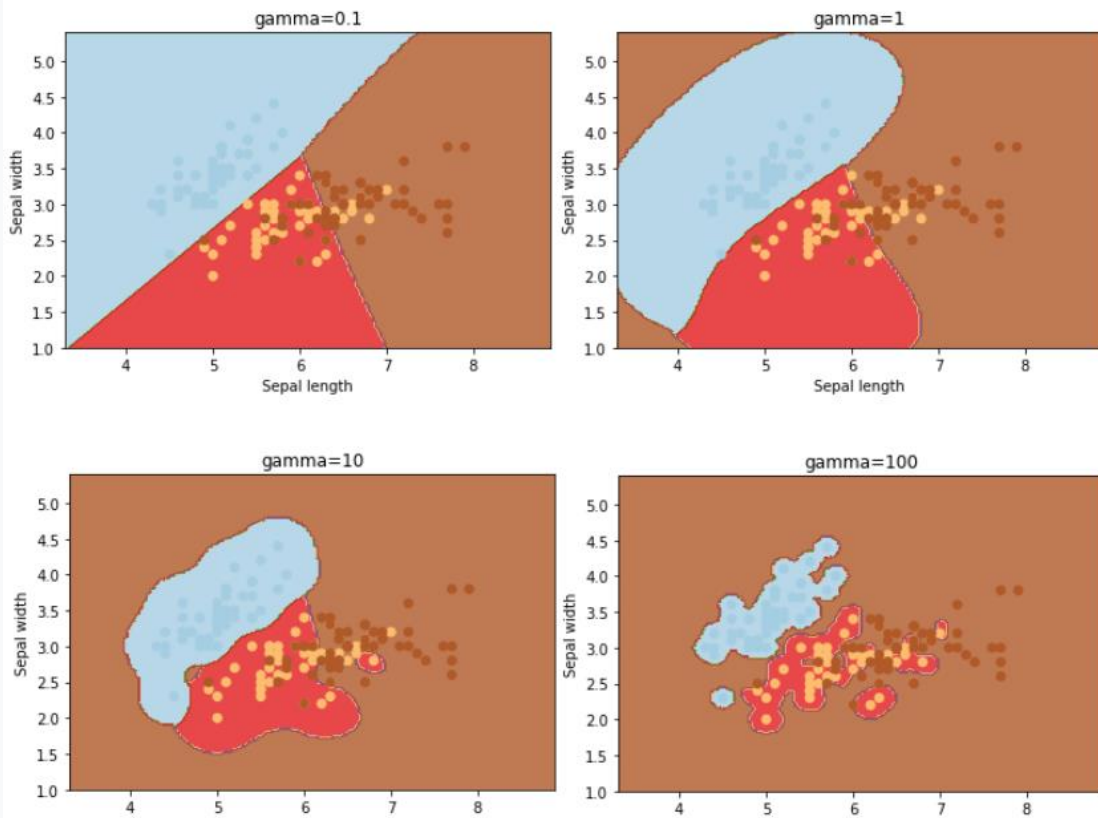
Yukarıdaki grafikte dikkat edilmesi gereken noktalar:

- Z için tüm değerler her zaman pozitif olur çünkü z hem x hem de y'nin karesel toplamıdır
- Orijinal grafikte, kırmızı daireler x ve y eksenlerinin orijinine yakın görünür ve z değerinin düşük olmasına ve başlangıç noktasından nispeten daha yüksek z değerine sahip yıldızla yol açar.
- SVM'de, bu iki sınıf arasında doğrusal bir hiper düzlemin olması kolaydır. Ancak, ortaya çıkan bir başka önemli soru, bir hiper düzlem için bu özelliği manuel olarak eklememiz gerektiğidir.
- İşte burada bize yardımcı olan SVM'nin çekirdek *hilesi adı* verilen bir tekniği vardır. Bunlar, düşük boyutlu girdi alanını alan ve daha yüksek boyutlu bir alana dönüştüren işlevlerdir(bir nevi dönüşüm işlemi), yani ayrılmaz problemi ayrılabilir probleme dönüştürür, bu fonksiyonel işlemlere ise çekirdek işlemleri denir.

- Çoğunlukla doğrusal olmayan ayırma probleminde çekirdek işlemi faydalıdır. Basitçe söylemek gerekirse, son derece karmaşık veri dönüşümleri yapar, ardından verileri tanımladığınız etiketlere veya çıktılarına göre ayırma işlemini öğrenir.

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.

FORMÜL

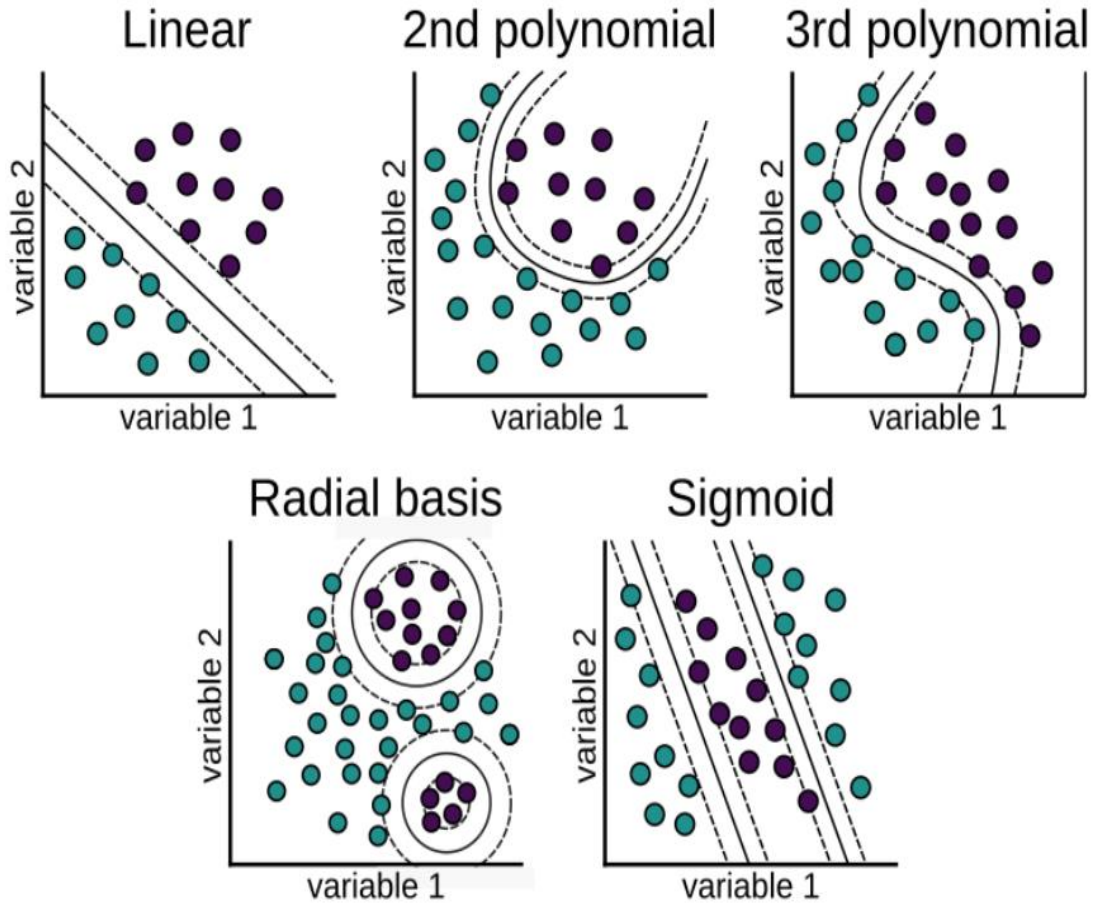
$$K(x, y) = e^{\gamma |x-y|^2}$$

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 azaltılması 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'e Polynomial Kernel denir.
- 7-) Model overfit olursa derecesi düşürülür, underfit olursa derece yükseltilir. Coefo 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.



KAYNAKÇA

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

<https://www.datascienceearth.com/algorithmdestek-vektor-makinelerisupport-vector-machinesr-kod-ornekli/>

https://tr.wikipedia.org/wiki/Destek_vekt%C3%B6r_makinesi