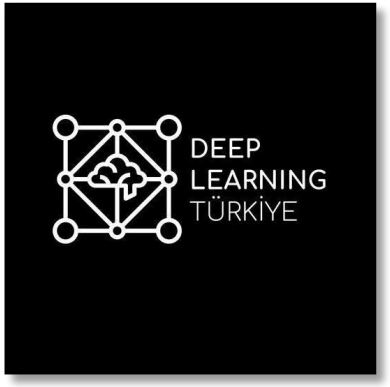
Sınıflandırma Modelleri

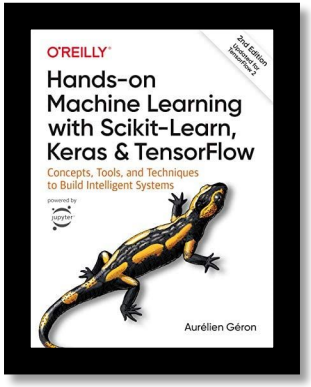
Kursa Başlamadan Önce

- Bu kurs tamamlandığı takdirde giriş düzeyi yapay zeka algoritmaları ve veri analizine temel oluşturabilecek genel bilgileri edinmiş olacaksınız
- Spesifik konular anlaşılması zor ve kişide ders esnasında mantığın oturması kolay olmayacağından bol bol bireysel pratik gerekmektedir
- Slaytlar yazılara boğulmadan görsellerle anlatılacaktır. Bu yüzden ders esnasında not tutulması **son derece** önemlidir
- Konu başlıkları temel düzey algoritmalar için yeterli olduğundan başlıklar araştırılmalı, bol bol uygulama ve teorik bilgiler içeren sitelerde araştırma yapılmalıdır

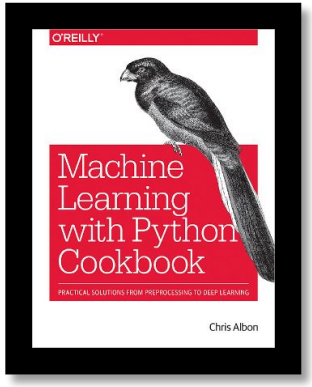
Kaynaklar



Deep Learning Türkiye



Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow
Aurelien Geron



Machine Learning with Python Cookbook
Chris Albon

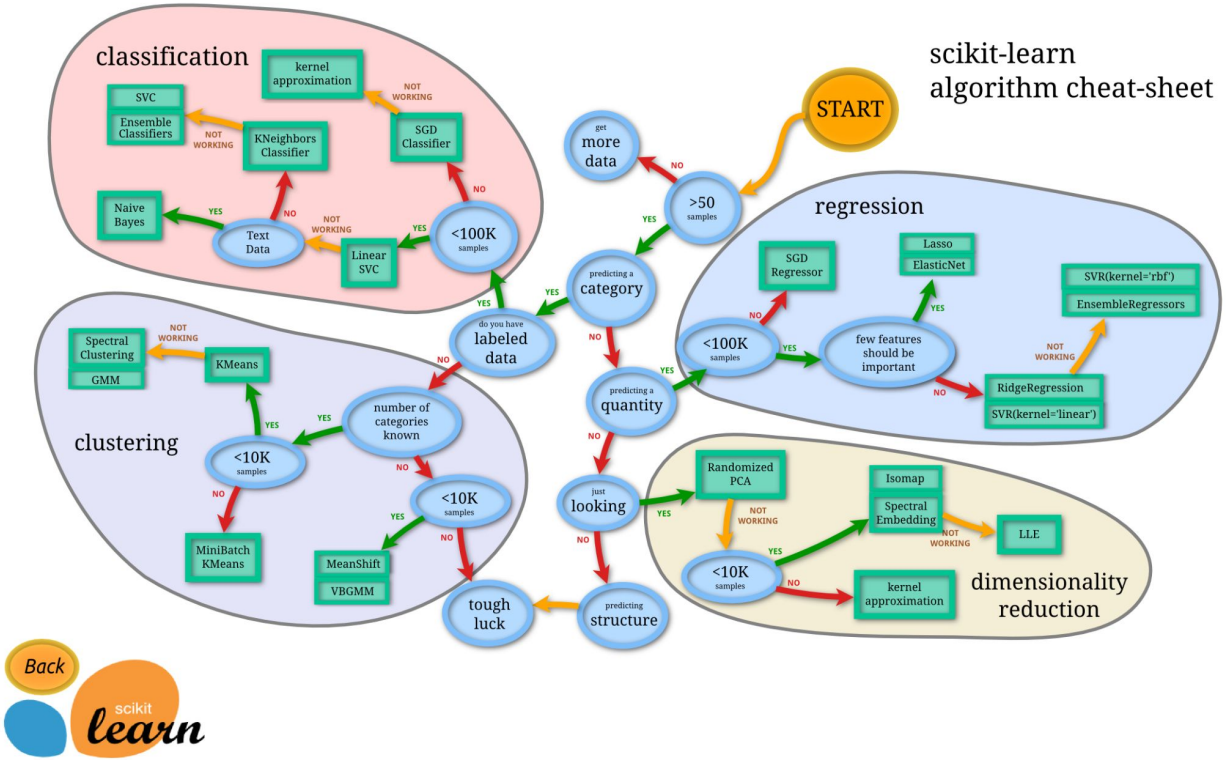
Fethi Tekyaygil

Twitter: fethidev
LinkedIn: fethitekyaygil
GitHub: TekyaygilFethi

Bugün Neler Öğreneceğiz?

1) Sınıflandırma nedir?
2) Lojistik Regresyon ile sınıflandırma nasıl yapılır?
3) Aktivasyon fonksiyonu nedir?
4) Aktivasyon fonksiyonunu neden kullanırız?
5) Sınıflandırma modelimizin performansını nasıl ölçeriz?
6) Cross Validation
7) Hata Metrikleri ölçerken dikkat edilmesi gerekenler, Accuracy, Confusion Matrix vs.
8) Classification Threshold kullanımı ve önemi nedir?
9) ROC ve AUC'un performans ölçümündeki yeri nedir?
10) KNN(K-Nearest Neighbors), SVM(Support Vector Machine) nedir?

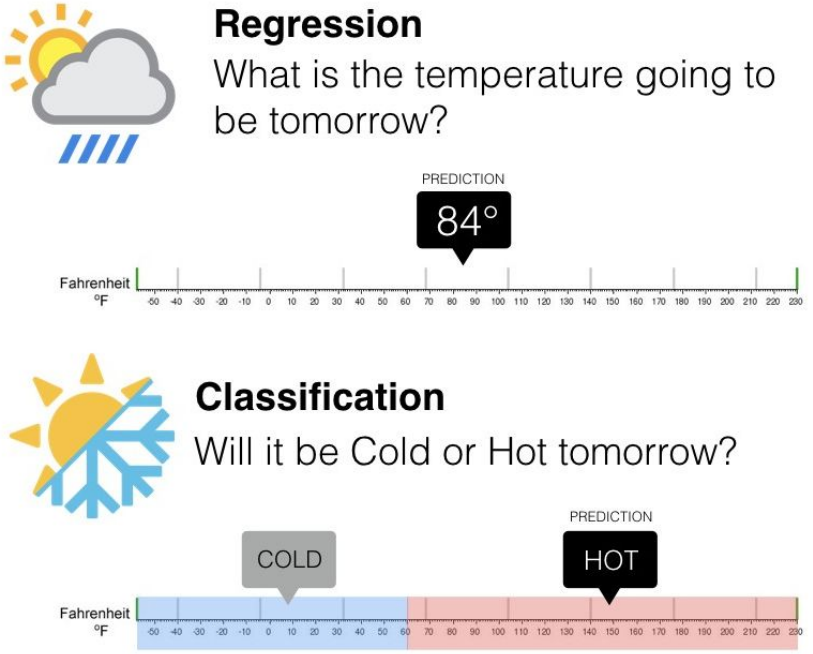
Bugün Neler Öğreneceğiz?



Sınıflandırma Nedir?

Bilgisayarın kendisine verilen etiketli verilerden öğrenerek, yeni gözlemleri sınıflara ayırmasını sağlayan bir denetimli öğrenme algoritmasıdır. Yalnızca iki sınıf sonucu olan ikili sınıflandırıcılar veya ikiden fazla çok sınıflı sınıflandırıcılar olabilir.

Süreç, verilen veri noktalarının sınıfını tahmin etmekle ilgilidir. Tahmin sonucu bulunacak bu sınıflara hedef veya etiket denir.



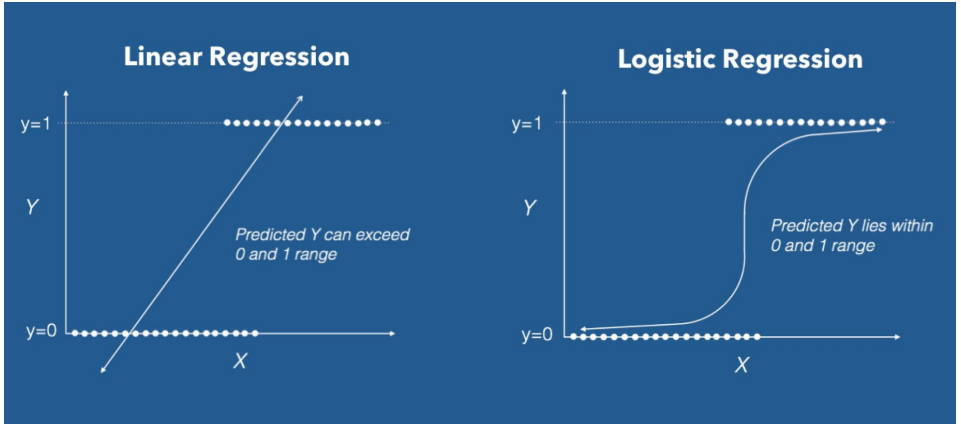
Lojistik Regresyon

Lojistik regresyon, bağımlı değişkenin kategorik bir değişken olduğu, sınıflandırma işlemi yapan bir regresyon yöntemidir. Amaç bağımlı ve bağımsız değişkenler arasında doğrusal bir model kurmaktır.

- Bağımlı değişkenin 2 farklı değer alabilmesi durumunda çalışır
- Lojistik regresyon, tam olarak 0 veya 1'i tahmin etmek yerine, 0 ile 1 arasında, özel bir olasılık değeri üretir.

- Evet/Hayır
- Spam/Not Spam

$$\phi = p(y = 1|x; \theta) = \frac{1}{1 + \exp(-\theta^T x)} = g(\theta^T x)$$



Lojistik Regresyon

- Lineer Regresyon'da olduğu gibi özniteliklerin ağırlıklı toplamını alınır ancak sonucu direk vermek yerine sonuca Sigmoid aktivasyon fonksiyonu uygulanır
- İkili sınıflandırma için kullanılır
- Çıktımız 0 ve 1 arasında bir olasılık değeri olur
- "S" karakteristiğinde bir fonksiyondur

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

$$t = \beta_0 + \beta_1 x \qquad \text{t=A+Bx}$$

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m = \beta_0 + \sum_{i=1}^{m} \beta_i x_i$$

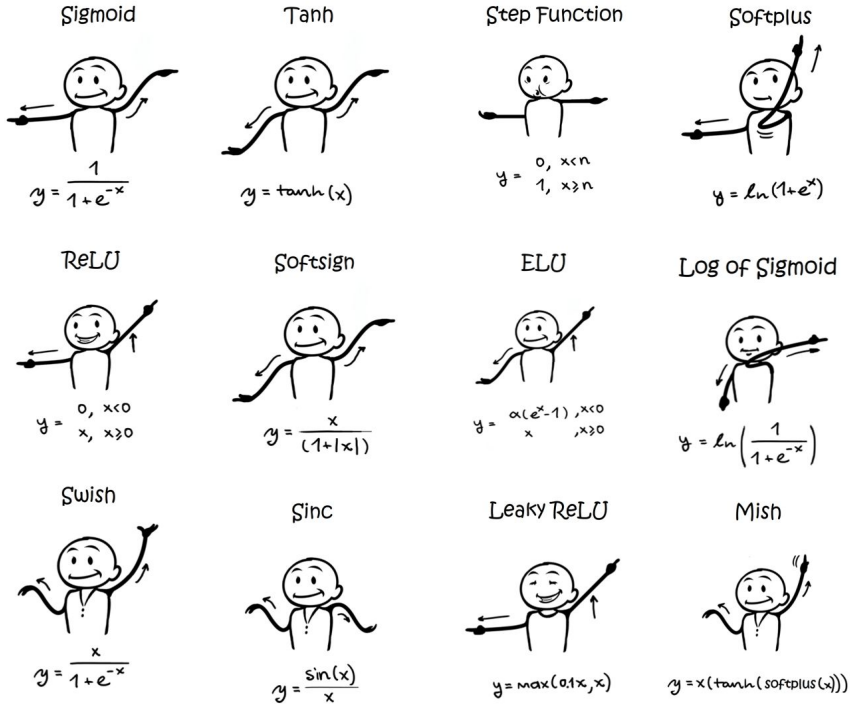
Aktivasyon Fonksiyonu

Aktivasyon Fonksiyonu Nedir?

Aktivasyon fonksiyonları modelin çıktısı üzerinde işlem yaparak çıktının nasıl olacağını belirleyen fonksiyonlardır.

Çıktı, sınıflandırma problemleri gibi belli bir değer kümesi ile sınırlandırmak istenirse lineer regresyon formülü kullanılamaz.

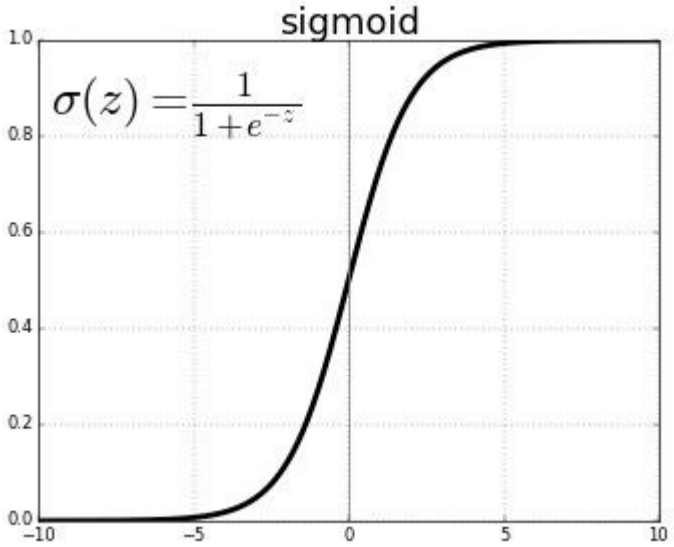
Lineer regresyon formülünde bağımlı değişken bağımsız değişkenlerin ağırlıklı toplamı iken, ikili sınıflandırma probleminde bağımlı değişken 1 ve 0 arasına sıkıştırılması gerektiğinden çıktıya özel bir aktivasyon fonksiyonu uygulanmalıdır.



Sigmoid Function

Sigmoid fonksiyonu, lineer regresyon formülünün çıktısını 0 ve 1 arasına sıkıştırarak bir bir olasılık değeri veren aktivasyon fonksiyonudur. Bu da çıktı değerinin 0 ve 1 arasında olduğu anlamına gelir.

- Sigmoid aktivasyon fonksiyonu kullanıldığında bağımsız değişkenlerdeki küçük değişimler bağımlı değişken üzerinde oransal olarak ciddi değişikliğe neden olacaktır. Bağımsız değişkenlerde ne denli değişiklik olursa olsun bağımlı değişken her zaman 0 ve 1 arasında olacaktır.



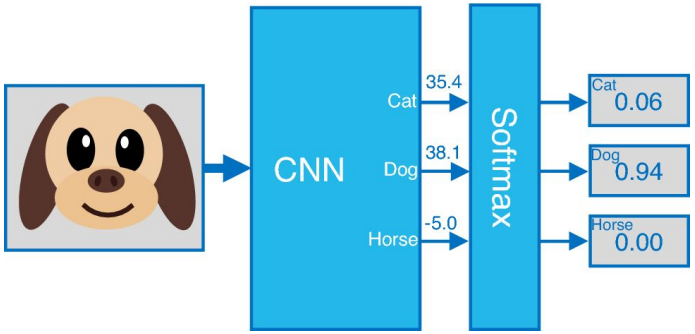
İkiden Fazla Sınıfımız Olduğunda Ne Yapılmalıdır?

Softmax Aktivasyon Fonksiyonu

Softmax, sigmoid fonksiyonunun aksine modelimizin ikiden fazla sınıf için olasılık çıktıları üretmesini sağlayan aktivasyon fonksiyonudur.

Derin Öğrenme modellerinde Softmax aktivasyon fonksiyonu son katmanda kullanıldığında her sınıf için ayrı bir olasılık değeri döndürülür. Bu olasılık değerlerinin toplamı 1'e eşittir.

Örneğin; yandaki modelde Cat, Dog ve Horse sınıfları için bir olasılık değeri üretilmiştir. Bu değerlere baktığınızda modeli, girdiğimizin %94 oranında Dog, %6 oranında Cat ve %0 oranında Horse olduğunu tahmin etmiştir.



$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

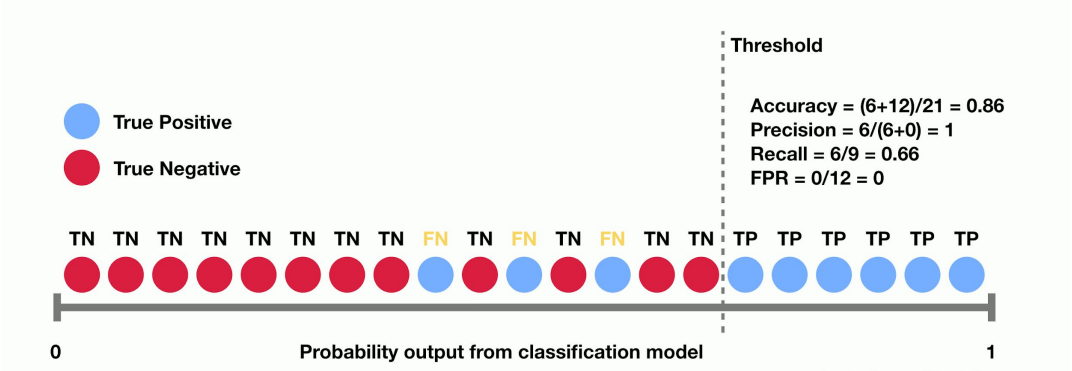
Sınıflandırma Modelinin Performansını Ölçmek

Sınıflandırma Eşikleri (Thresholding)

Eşik, lojistik regresyon sonucunda tahmin edilen olasılık değerlerinden hangilerinin doğru kabul edileceğini belirleyen değerdir.

Örneğin; Maillerin spam olup olmama durumuna göre sınıflandıran bir modele 0.5 eşiği verilirse:

- Spam olasılığı 0.5 değerine eşit veya üstünde olan mailler spam olarak sınıflandırılır
- Spam olasılığı 0.5 değerinin altında olan mailler spam değil olarak sınıflandırılır



True vs False & Positive vs Negative

Sınıflandırma modelinin performansının ölçülebilmesi için bazı verilerin kullanılması gereklidir. Bu veriler True Positive, True Negative, True Positive ve False Positive değerleridir.

Positive vs Negative

True ve False değerlerimiz temelde etiket sınıflarımızdır.

Örneğin; bir e-mail spam sınıflandırma modelinde:

- “Spam” = Positive
- “Not Spam” = Negative

olarak varsayılabilir.

True vs False

Positive ve Negative değerlerimiz modelimizin tahmin sonuçlarıyla alakalıdır.

Örneğin; model gerçekte “Spam” (Positive) olan bir maili “Not Spam” (Negative) olarak tahmin ettiğinde bu tahmin False Negative bir tahmin olur.

Bunun nedeni modelimizin Negative bir tahmin gerçekleştirmesi ve bu tahminin de yanlış (False) olmasıdır.

Karmaşıklık Matrisi

Sınıflandırma modelinin performansını değerlendirmek için kullanılan gerçek sınıf değeri ile tahmin edilen sınıf değerinin karşılaştırıldığı bir matristir.

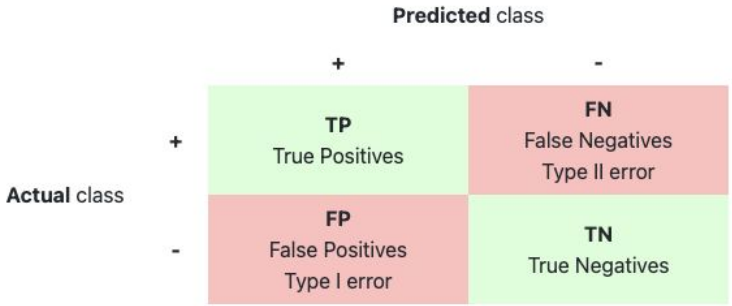
Positive: "Spam" **Negative:** "Spam Değil"

True Positive: Modelin pozitif sınıfı doğru bir şekilde tahminlediği bir sonuçtur **Gerçek:** E-mail Spam **Model Tahmini:** E-mail Spam	**False Positive:** Modelin pozitif sınıfı doğru bir şekilde tahminleyemediği bir sonuçtur **Gerçek:** E-mail Spam Değil **Model Tahmini:** E-mail Spam
False Negative: Modelin negatif sınıfı doğru bir şekilde tahminlediği bir sonuçtur **Gerçek:** E-mail Spam Değil **Model Tahmini:** E-mail Spam	**True Negative:** Modelin negatif sınıfı doğru bir şekilde tahminlediği bir sonuçtur **Gerçek:** E-mail Spam Değil **Model Tahmini:** E-mail Spam Değil

Accuracy

Accuracy metriği, doğru tahmin edilen tahmin sayısının tüm tahminlerin sayısına bölünmesiyle bulunur.

Dengesiz dağılımlı veri setleri için tek başına yeterli olmamasından dolayı başka sınıflandırma metriklerine ihtiyaç duyulmuştur.



Metric	Formula	Interpretation
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	Overall performance of model
Precision	$\frac{TP}{TP + FP}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{TP}{TP + FN}$	Coverage of actual positive sample
Specificity	$\frac{TN}{TN + FP}$	Coverage of actual negative sample
F1 score	$\frac{2TP}{2TP + FP + FN}$	Hybrid metric useful for unbalanced classes

Neden Accuracy Tek Başına Yeterli Değil?

Toplam Mail Sayısı: 100

Gerçek Spam E- Mail Sayısı: 98

Gerçek Spam Olmayan E-Mail Sayısı: 2

Varsayım: Modelin tüm mailler için Spam tahmini gerçekleştirmesi

True Positive 98	**False Positive** 2
False Negative 0	**True Negative** 0

Precision & Recall

Precision ve Recall sınıflandırma modelinin performansının ölçülmesini sağlayan iki metriktir.

Precision: *"Pozitif tahminlerin ne kadarı doğruydu?"* sorusuna cevap arayan bir metriktir.

Bu soruya cevap vermek için aşağıdaki Precision formülünü kullanabiliriz:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall: "*Gerçekte pozitif olan sonuçların kaçı doğru tahmin edildi?"* sorusuna cevap arayan bir metriktir.

Bu soruya cevap vermek için aşağıdaki Recall formülünü kullanabiliriz:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Precision & Recall Hesaplaması

True Positive 176	**False Positive** 97
False Negative 64	**True Negative** 70

Precision = 176 / (176 + 97) => 176 / 273 => 0.64

Recall = 176 / (176 + 64) => 176 + 240 => 0.73

Precision ve Recall Ne Zaman Kullanılmalı?

Precision: Herhangi bir False Positive değeri elde etmek göze alınamadığı zaman, Precision'a öncelik verilir. Başka bir deyişle herhangi bir yanlış algılama göze alınamadığında, yüksek precision değeri aranır.

Örneğin; Bir zombi kıyametinde, güvenli bölgenize mümkün olduğunca çok sayıda sağlıklı insanı (Negative) kabul etmeye çalışırsınız, ancak gerçekten bir zombiyi (Positive) yanlışlıkla güvenli bölgeye geçirmek istemezsiniz (False Positive). Yönteminiz bazı sağlıklı insanların yanlışlıkla güvenli bölgeye girmemesine neden oluyorsa, bu kabul edilebilir bir yöntemdir.

Recall: Herhangi bir False Negative değeri elde etmek göze alınamadığı zaman, Recall'a öncelik verilir. Başka bir deyişle herhangi bir algılama kaçırma göze alınamadığında, yüksek recall değeri aranır.

Örneğin; Sağlıklı bir kişiyi(Negative) kanserli(Positive) olarak sınıflandırmak (False Positive) ve daha fazla tıbbi test yaptırmakta sorun yoktur, ancak bir kanser hastasını (Positive) sağlıklı (False Negative) olarak sınıflandırmak kesinlikle doğru değildir.

Karmaşıklık Matrisi - Precision

Positive: “Zombi Değil” **Negative:** “Zombi”

True Positive **Gerçek:** Zombi Değil **Model Tahmini:** Zombi Değil	**False Positive** **Gerçek:** Zombi **Model Tahmini:** Zombi Değil
False Negative **Gerçek:** Zombi Değil **Model Tahmini:** Zombi	**True Negative** **Gerçek:** Zombi **Model Tahmini:** Zombi

Karmaşıklık Matrisi - Recall

Positive: “Kanser” **Negative:** “Kanser Değil”

True Positive **Gerçek:** Kanser **Model Tahmini:** Kanser	**False Positive** **Gerçek:** Kanser Değil **Model Tahmini:** Kanser
False Negative **Gerçek:** Kanser **Model Tahmini:** Kanser Değil	**True Negative** **Gerçek:** Kanser Değil **Model Tahmini:** Kanser Değil

F1 Skoru

Precision ve Recall bazı durumlarda birbirileri yerine tercih edilebilecek yöntem olsa da her zaman modelde ikisinden hangisinin daha iyi sonuç vereceği bilinmeyebilir. F1 Skoru metriği Precision ve Recall metriklerinin harmonik ortalamasına eşit olduğundan bu iki metriği birleştirir.

Harmonik ortalama olmasının sebebi eğer aritmetik ortalama alınıyor olsaydı precision veya recall değerlerinden biri 0 ise, bu değerlerin ortalamaya katkısı olmayacaktı. Bunun yanı sıra harmonik ortalamada precision ve recall değerleri, F1 Skorunun yükselmesinde eşit rol oynamaktadır.

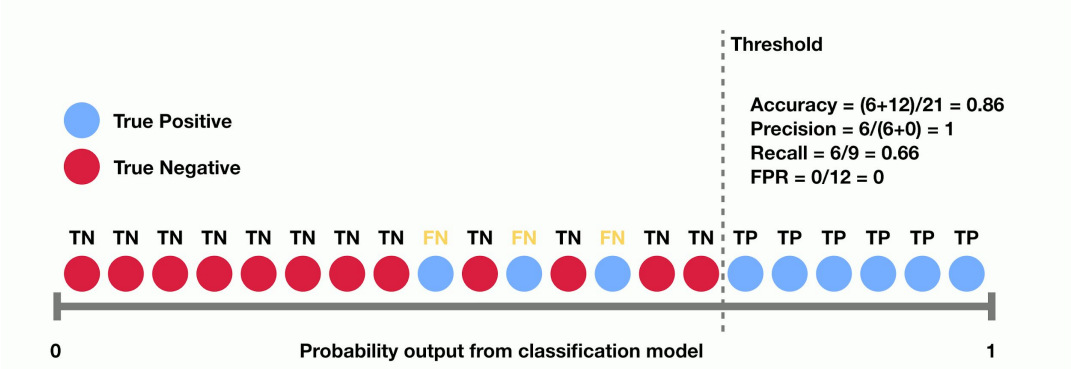
$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Sınıflandırma Eşikleri, ROC Eğrisi & AUC

Sınıflandırma eşiği değiştikçe Karmaşıklık Matrisi değerleriniz de eşiğe bağlı olarak değişecektir. Modelin performansı eşiğe bağlı olarak değişeceğinden belli eşikler için belirli performans sonuçları bulunmaktadır.. Modelin performansını tam kapsamıyla ölçmek için ROC Eğrisine ve AUC'a bakılması gerekmektedir.

ROC eğrisi: Tüm sınıflandırma eşiklerinde bir sınıflandırma modelinin performansını gösteren bir grafik

AUC: "ROC Eğrisinin Altındaki Alan" anlamına gelir. AUC, (0,0) ile (1,1) arasındaki tüm ROC eğrisinin altındaki tüm iki boyutlu alanı ölçer



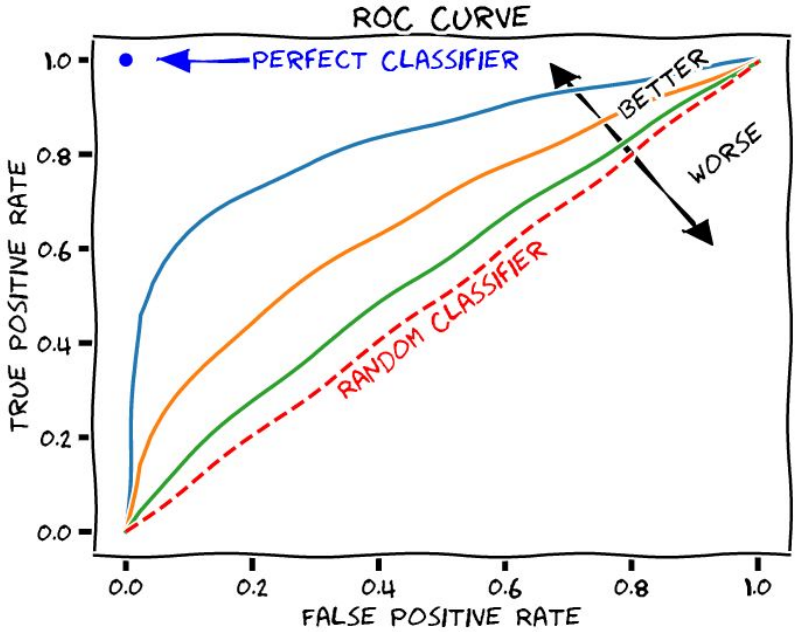
Sınıflandırma Eşikleri ve ROC Eğrisi & AUC

ROC Eğrisi'nin iki parametresi vardır:

1. True Positive Rate (TPR): Recall değeridir
2. False Positive Rate(FPR): "Negatif verilerin kaç tanesi yanlış tahmin edilmiş?" sorusuna cevap verir.

Bir ROC eğrisi, farklı sınıflandırma eşiklerinde TPR'ye karşı FPR'yi çizer. Sınıflandırma eşiğinin düşürülmesi, daha fazla öğeyi pozitif olarak sınıflandırır, böylece hem False Pozitifleri hem de True Pozitifleri artırır.

Bir ROC eğrisindeki noktaları hesaplamak için, farklı sınıflandırma eşikleri ile bir lojistik regresyon modelini birçok kez değerlendirilmesi gerekir, ancak bu yöntem verimsiz olacaktır. Bu işi AUC kullanarak daha efektif şekilde gerçekleştirebiliriz.

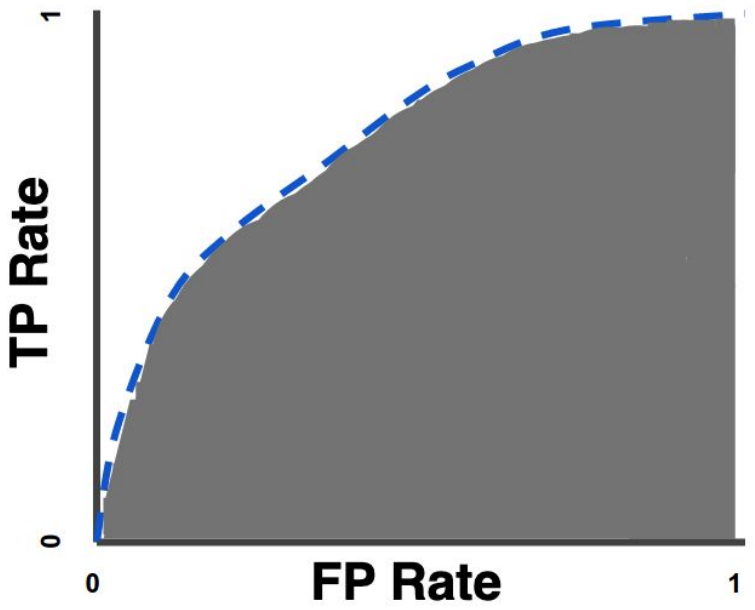


Sınıflandırma Eşikleri ve ROC Eğrisi & AUC

AUC tüm olası sınıflandırma eşiklerinde toplu bir performans ölçüsü sağlar. Başka bir deyişle AUC değerimiz modelimizin sınıfları ne kadar iyi sınıflandırdığını belirler.

AUC değeri 0 ve 1 arasındadır.

- AUC = 0 durumunda model her sınıflandırmayı yanlış gerçekleştirmektedir
- AUC = 1 durumunda model her sınıflandırmayı doğru gerçekleştirmektedir



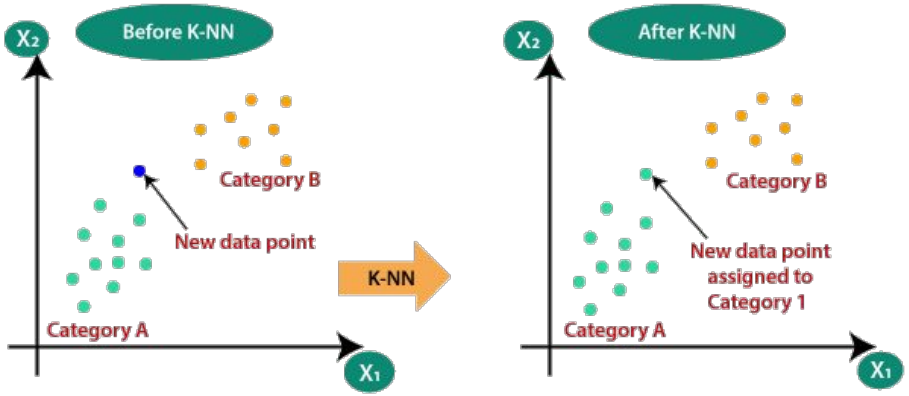
Sık Kullanılan Sınıflandırma Algoritmaları

K-Nearest Neighbors

Birbirine benzer özelliklerin mesafe olarak birbirilerine yakın olduğunu varsayan Denetimli Öğrenme algoritmasıdır.

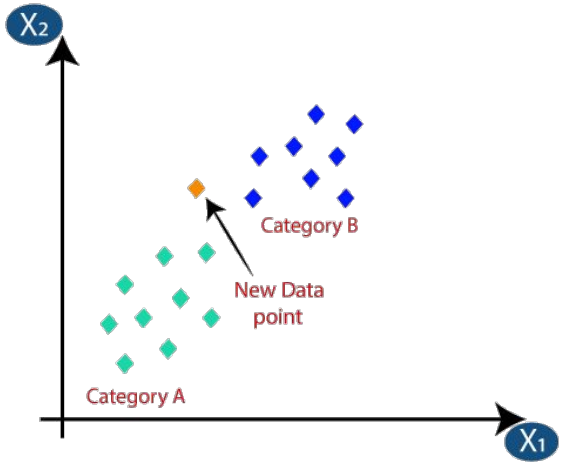
Bir gözlemin, en yakın komşularına bakarak en büyük orana sahip olan sınıfa ait olduğu tahmin eder.

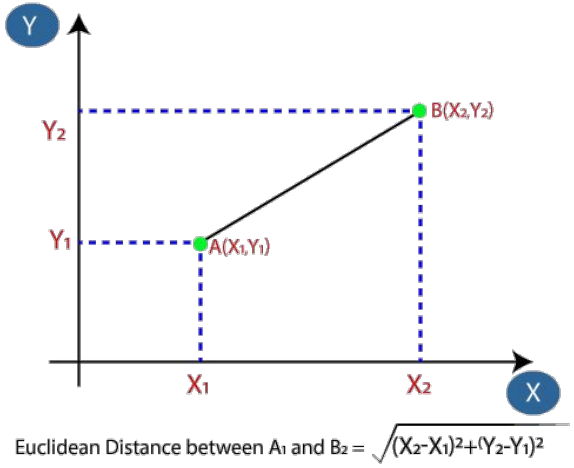
Bir örnek kümesindeki verilerin dağılımından yararlanılarak sınıflandırılmasında kullanılmaktadır. Yeni bir veri noktasının sınıfını tahmin etmek için o noktanın mevcut veri noktalarına uzaklığı hesaplanıp, k sayıda yakın komşuluğuna bakılır.



K-Nearest Neighbors

Tahmin edilmesi istenen veri girilir ve k parametresi ile kaç komşu noktaya bakılmak istendiği belirlenir. Sonrasında veriler arasındaki Öklid mesafesi belirlenerek en yakındaki noktaların sınıflarına bakılır. Yeni veri noktası bu sınıf olarak tahminlenir.

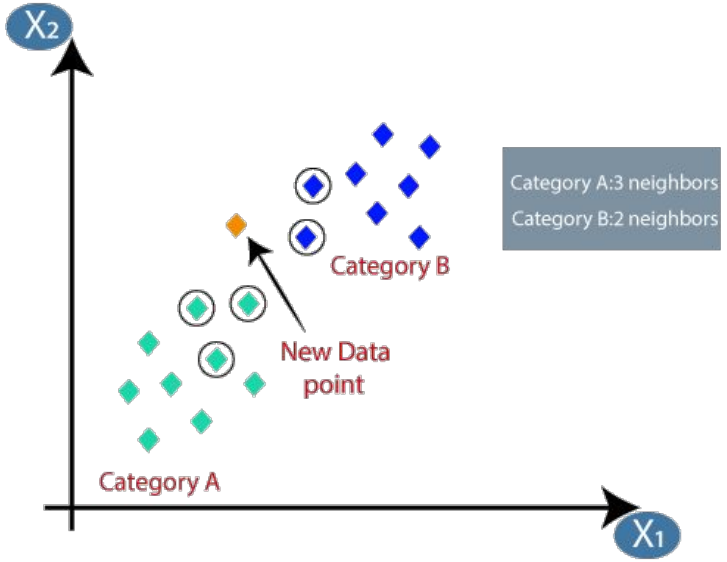




K-Nearest Neighbors

Örneğin, yandaki grafikte turuncu veri noktası model tarafaından sınıflandırılmak istenen veri noktası olsun. k değerinin 5 verildiği varsayılarak, Öklid uzaklığı en yakın 5 en yakın veri noktasına bakılır.

Category A sınıfından 3, Category B sınıfından 2 veri noktası komşu olduğundan veri noktası Category A sınıfı olarak sınıflandırılır.



K-Nearest Neighbors

Mesafe Ölçümleri

Problemin içeriğine bağlı olarak, farklı uzaklık ölçütleri kullanılabilir.

- Minkowski mesafesi, C değişkenine bağlı bir uzaklık hesaplamak isteniyorsa Minkowski yöntemi kullanılır.
- C = 1 olduğunda formül Manhattan mesafesini verir
- C = 2 olduğunda, Öklid mesafesini verir
- Hamming mesafesi iki adet ikili dizi arasındaki benzerliği verir

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

Manhattan distance

$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

Hamming distance

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Euclidean distance

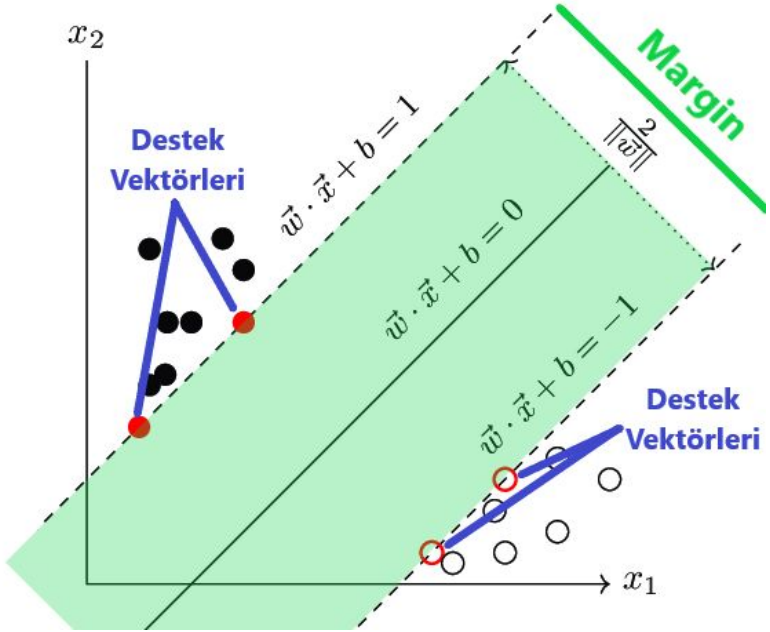
$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^c \right)^{\frac{1}{c}}$$

Minkowski distance

Support Vector Machine (SVM)

SVM, doğrusal veya doğrusal olmayan sınıflandırma, regresyon ve hatta aykırı değer tespiti yapabilen güçlü ve çok yönlü bir Makine Öğrenimi modelidir.

- SVM'ler, özellikle karmaşık küçük veya orta ölçekli veri kümelerinin sınıflandırılması için çok uygundur
- Düzleme yeni veri noktalarının eklenmesi Decision Boundary çizgisini ve destek Vektörlerini etkilemez

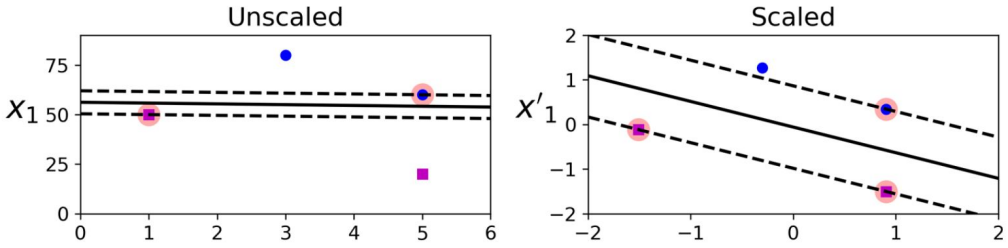


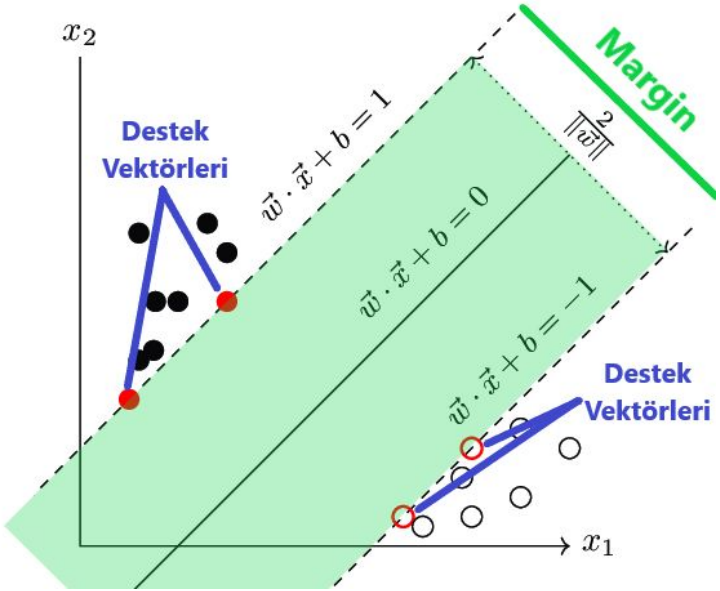
Support Vector Machine (SVM)

Yandaki örnekte ortadaki çizgi, sınıfları ayıran Decision Boundary iken bu çizginin +-1 taraflarında bulunan kesikli çizgiler Destek Vektörleridir.

SVM, iki sınıf verileri arasındaki en geniş sokak olarak düşünülebilir.

SVMler özellik ölçeklemesine duyarlıdır



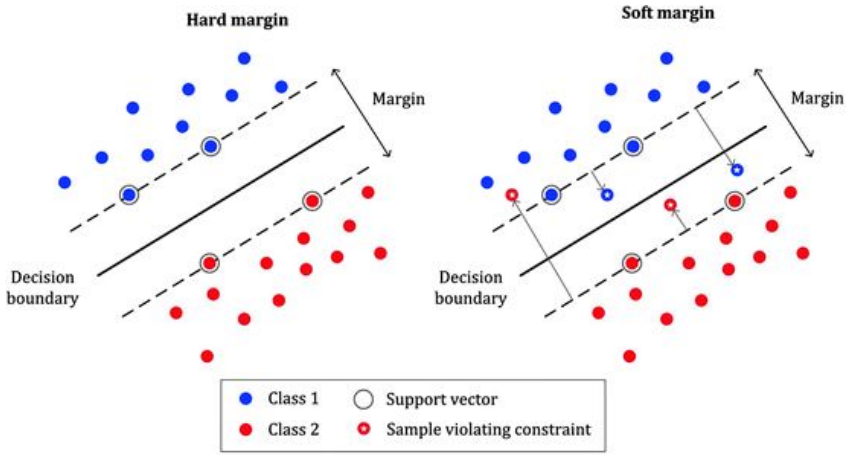


Hard Margin ve Soft Margin

Decision Boundary çizgisi ile ona en yakın veri noktasının arasındaki uzaklık Margin'i belirler. Margin, veri noktasına ulaşmadan önce sınırın artırılabileceği genişliktir. SVM modelinin aykırı verilere bu kadar çok uyum sağlamaması için bazı yanlış sınıflandırmalara izin verilmesi gerekmektedir.

Hard Margin, verilerin margin sınırları içerisine girmemesidir. Başka bir deyişle modelin veriye fazla uyum sağlaması ve yanlış sınıfladırmayz izin verilmemesi anlamına gelir.

Soft Margin, Soft Margin, bazı verilerin Destek Vektörleri sınırlarına girmesidir. Başka bir deyişle sınıflandırmada bazı yanlış sınıflandırmaların tolere edilmesidir.



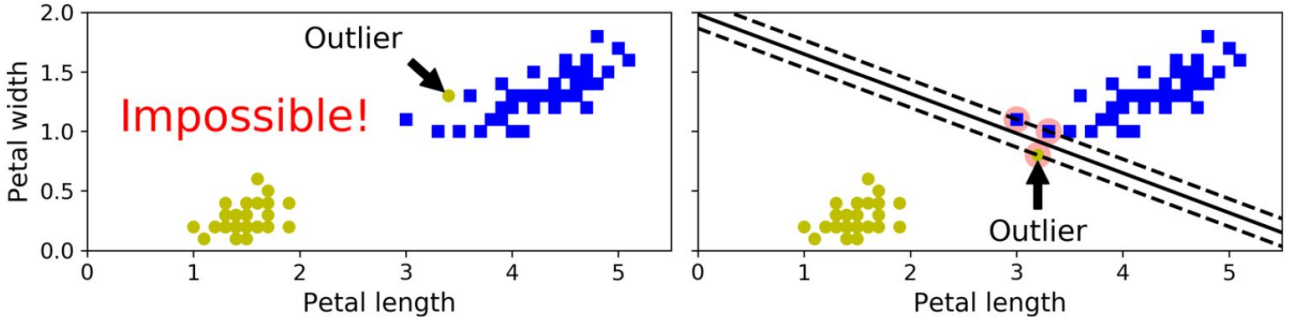
Hard Margin & Soft Margin

Hard Margin: Hard Margin, veriler doğrusal olarak ayrılabiliyorsa çalışır.

Modelimiz yanlış sınıflandırmaları tolere etmiyorsa sınır gözlemler ile eşik arasındaki fark hard margin olur.

Soft Margin: Hard Margin'de oluşan aykırı değer probleminden kaçınmak için daha esnek bir model kullanılması gerekmektedir. Bu esneklik bazı yanlış sınıflandırmalara izin verilmesidir.

Modelimiz bu yanlış sınıflandırmaları tolere ediyorsa, sınır gözlemler ile eşik arasındaki fark soft margin olur.



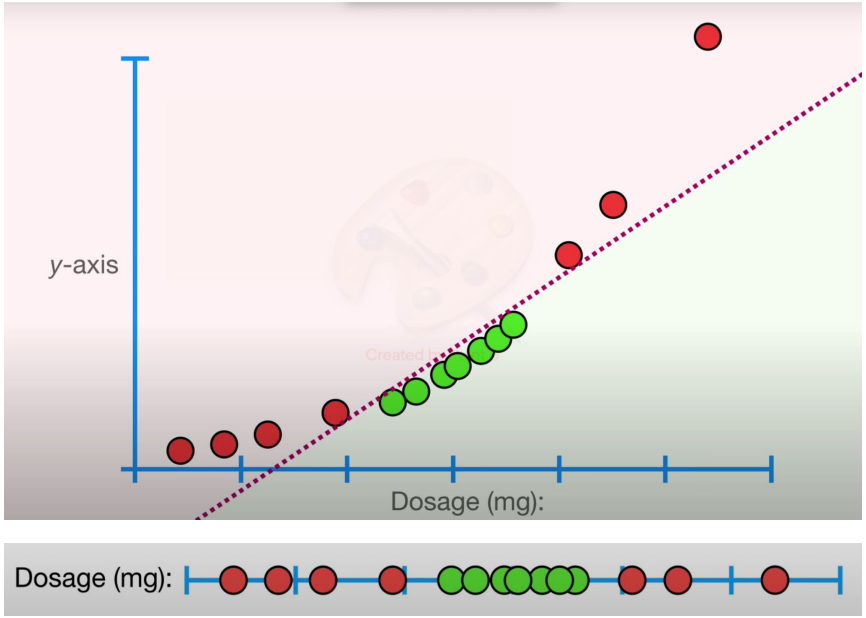
SVM Çekirdek Hilesi

SVM verileri doğrusal olarak sınıflandırmaya çalışır ancak SVM doğrusal dağılmayan verilerde performanslı tahmin gerçekleştiremeyecektir.. Böyle bir durumda Kernel Trick(Çekirdek Hilesi) kullanılır.

Polynomial Kernel

Boyut arttırımı ile yeni bir boyut oluşturarak doğrusal sınıflandırma sağlanır.

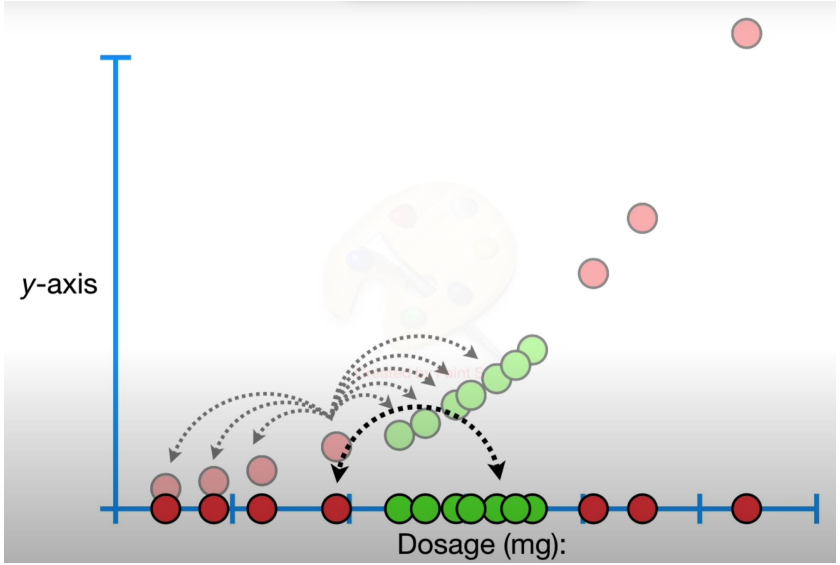
Verinin hangi boyutta SVM tarafından en iyi ayrılacağına Cross Validation ile karar verilebilir.



SVM Çekirdek Hilesi

Veriler yüksek dereceli polinom denklemiyle nonlineer hale dönüştürüldükçe performans kaybı yaşanmaya başlanır.

Çekirdek hilesi, çok yüksek dereceli polinomlarla bile, onları eklemek zorunda kalmadan birçok polinom özelliği eklenmiş gibi aynı sonucu elde etmeyi mümkün kılar. Bu nedenle, aslında herhangi bir özellik eklenmediği için özellik sayısında patlama olmaz.



Colab Zamanı!