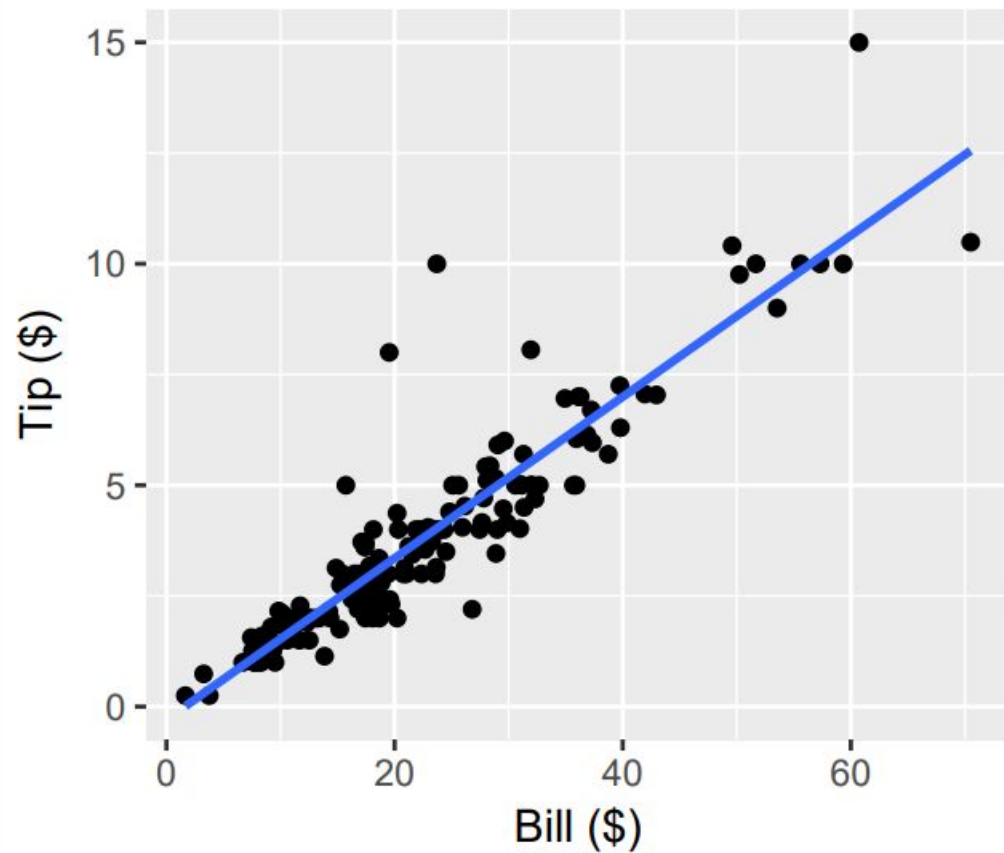
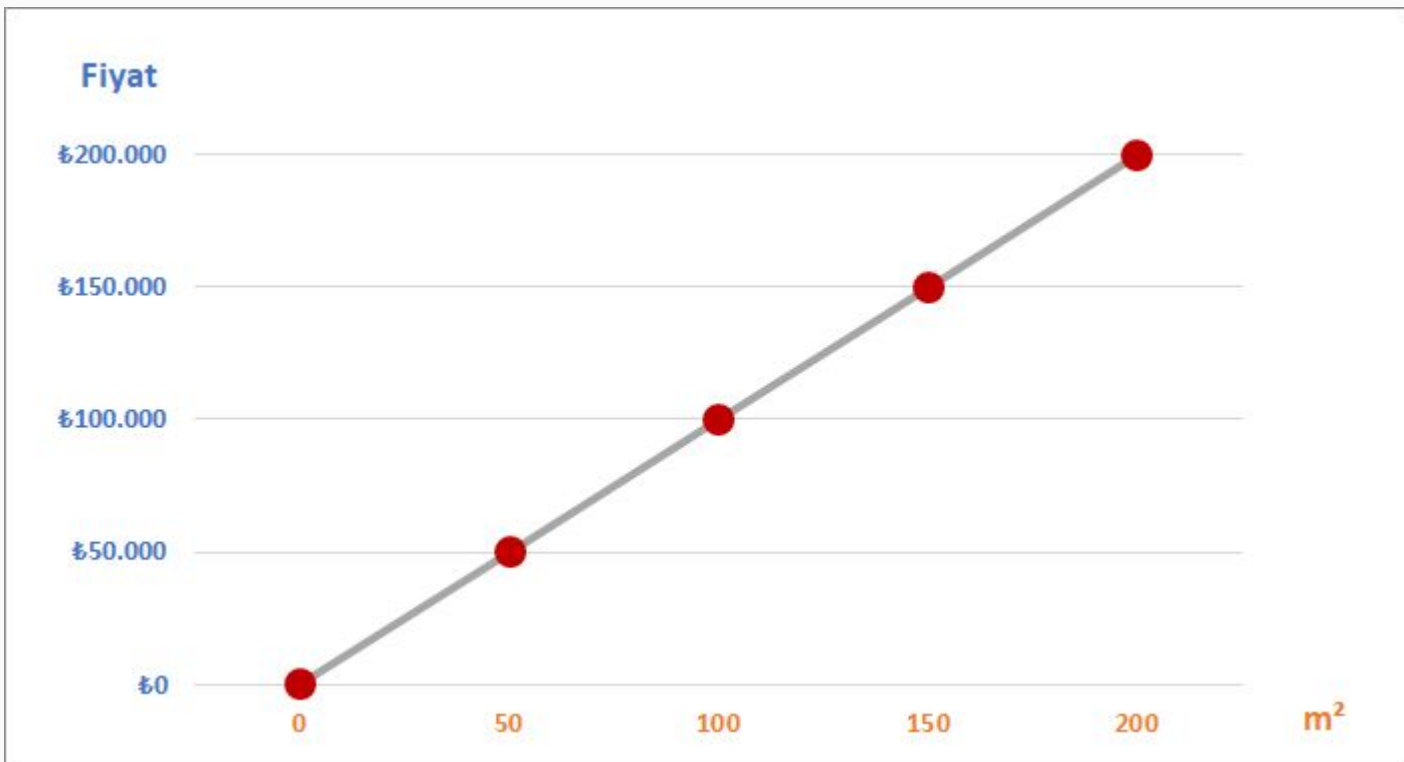


ML Modelleri

Lineer Regresyon

İki veya daha fazla değişken arasındaki ilişkinin bir doğru ile gösterilmesine Lineer Regresyon denir.





Fiyat

₺200.000

₺150.000

₺100.000

₺50.000

₺0

0

50

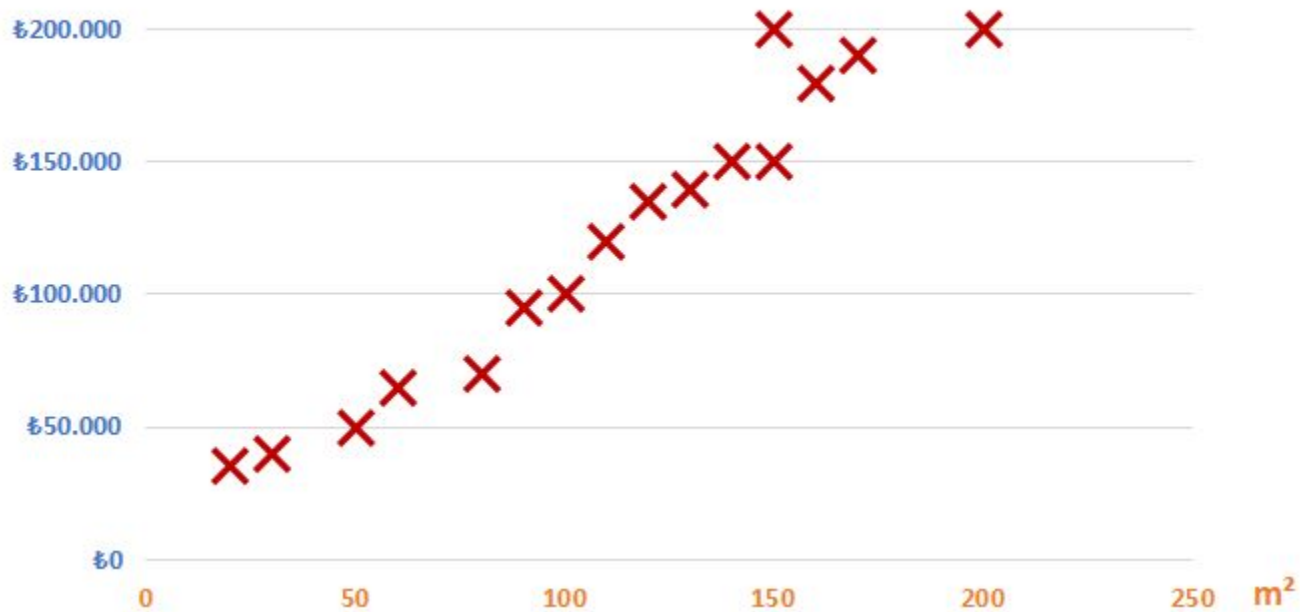
100

150

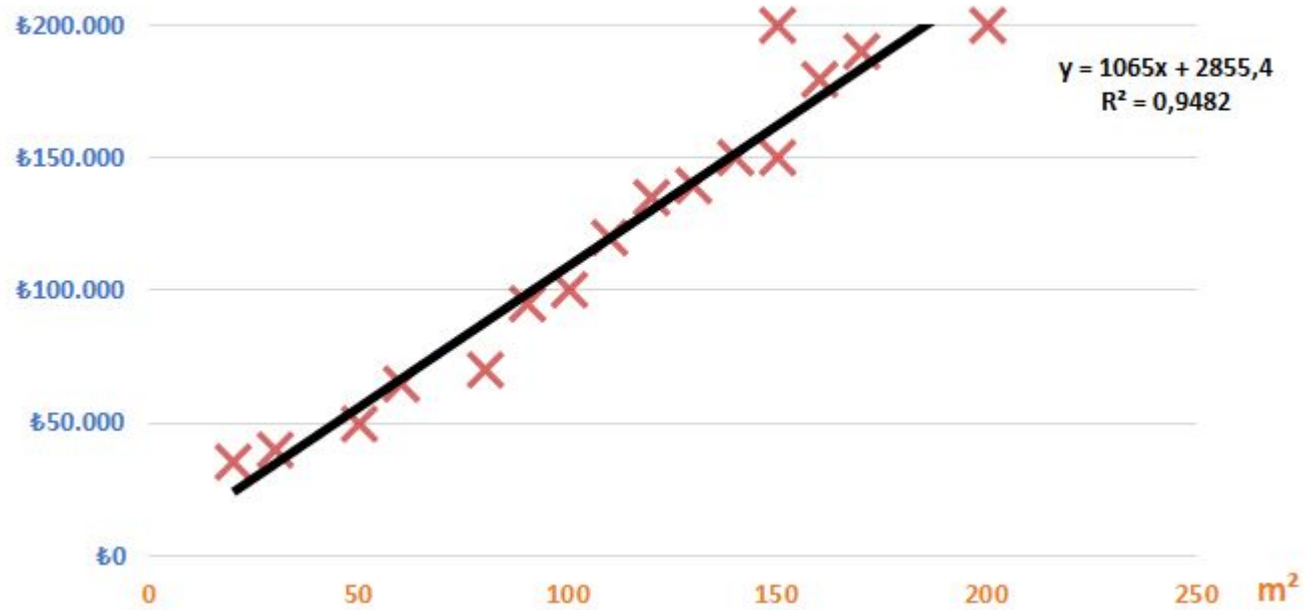
200

250

m²



Fiyat



$$\hat{Y} = bx + \alpha$$

Tahmin edilen y değeri

Doğrunun eğimi

Tahmin etmek için verilen x değeri

x=0 iken y'nin değeri

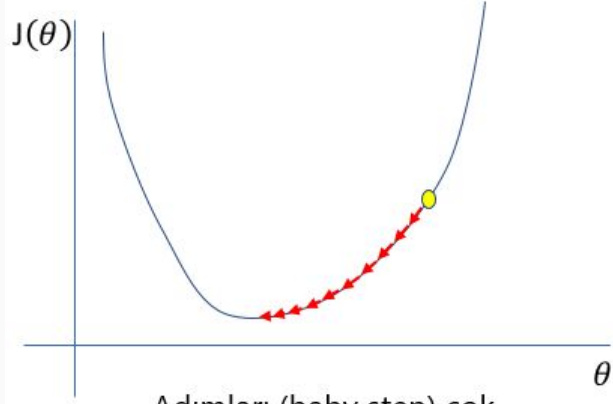
Ortalama Hata Karesi (Mean squared error)

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n \underbrace{(y_i}_{\text{Gerçek y değeri}} - \underbrace{\hat{y}_i)}_{\text{Tahmin edilen y değeri}})^2$$

n = veri setinin büyüklüğü

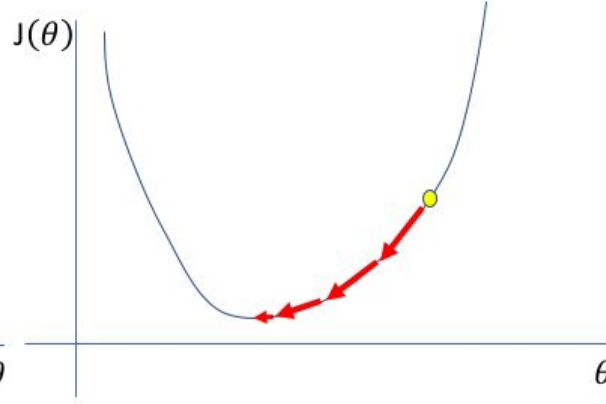
Doğrunun, veriye ne kadar yakın olduğunu ölçer.

Çok düşük learning rate



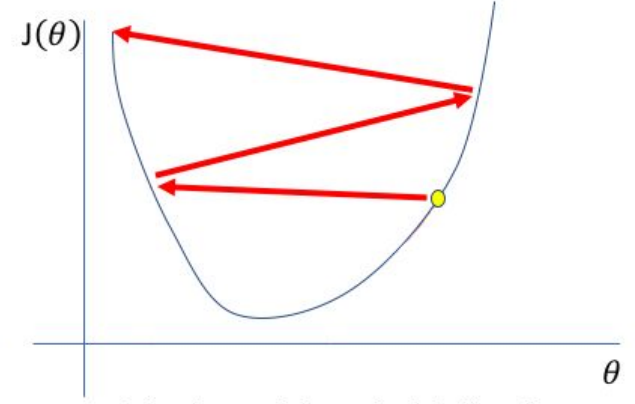
Adımları (baby step) çok ufak olduğundan global minimum (optimum) noktasına ulaşması çok uzun sürüyor.

Olması gereken learning rate



Kısa bir şekilde global minimum (optimum) noktasına ulaşılır.

Çok büyük learning rate



Adımları çok büyük olduğundan hata değeri azalmak yerine artar.

Lojistik Regresyon

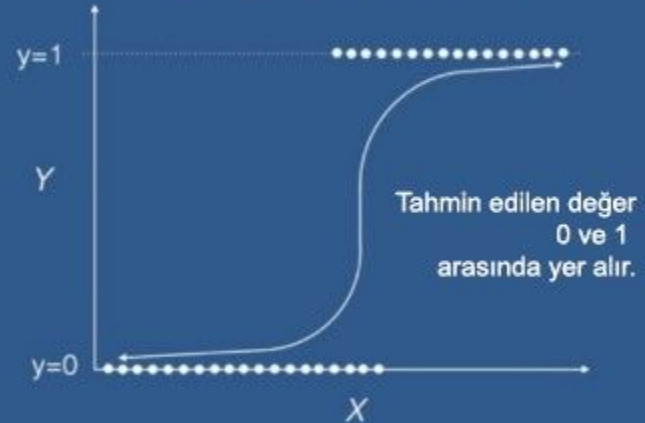
Lojistik regresyonun, lineer regresyon ile arasındaki en büyük farkı iki sınıfı birbirinden ayıracak çizgiyi nasıl uyguladığıdır (fit).

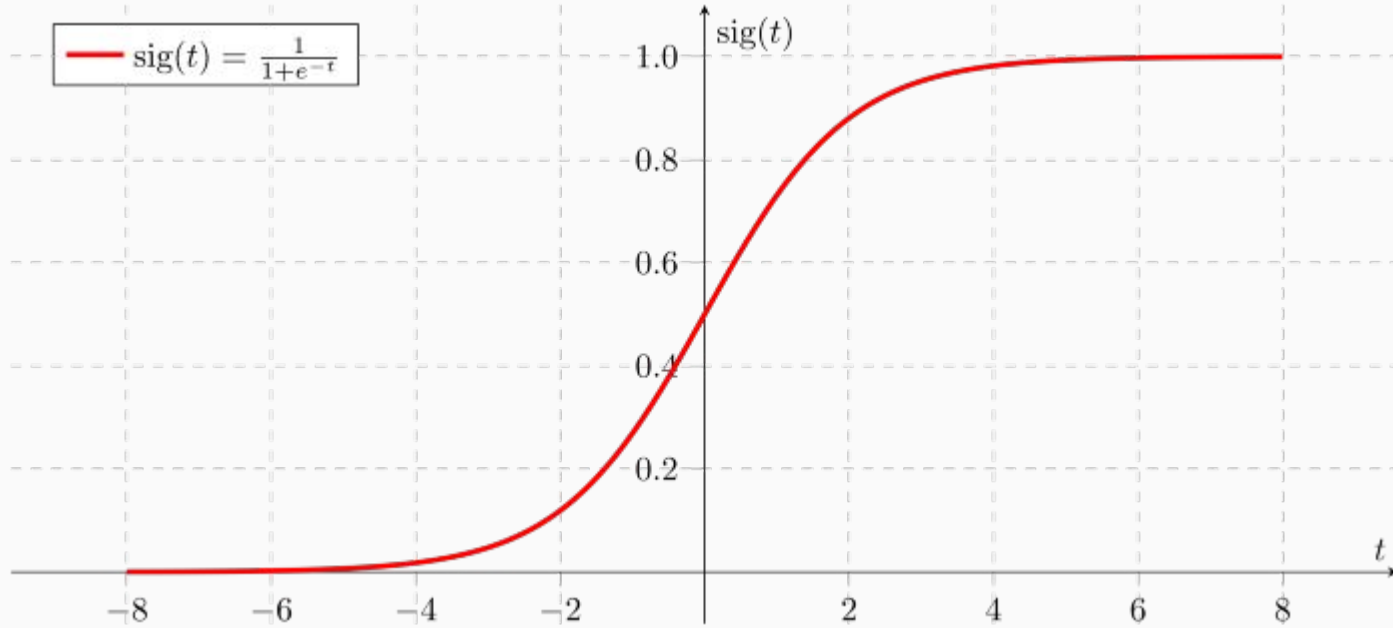
Lineer regresyon, optimum çizgiyi çizmek için “En Küçük Kareler Yöntemi” (Least Squares) kullanırken, lojistik regresyon “Maksimum Olabilirlik” (Maximum Likelihood) kullanır.

Linear Regression



Logistic Regression





Lojistik regresyon, sınıflandırma yapmak için Sigmoid (Lojistik) Fonksiyonu kullanır. Sigmoid fonksiyonu “S” şeklinde bir eğridir. Lojistik fonksiyon ($f(x)$), $-\infty/+\infty$ aralığındaki tüm değerleri girdi olarak kabul edebilir. Çıktı olarak ise 0 ile 1 arasında değerler almaktadır.

Sigmoid fonksiyonu basitçe, verilerimizi 0 ve 1 arasına sıkıştırmak için kullanılan fonksiyondur. Bu fonksiyon sayesinde sınıflandırma yapabiliriz. Derin Öğrenme içerisinde aktivasyon fonksiyonları altında da sıkça kullanılır.

$$g(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(\theta^T x)}}$$

Avantajları:

- Lojistik regresyonun uygulanması, yorumlanması kolaydır.
- Veri seti doğrusal olarak ayrılabiliriyorsa oldukça iyi performans gösterir.
- Overfitting'e daha az meyillidir ama büyük veri setlerinde overfit olabilir.

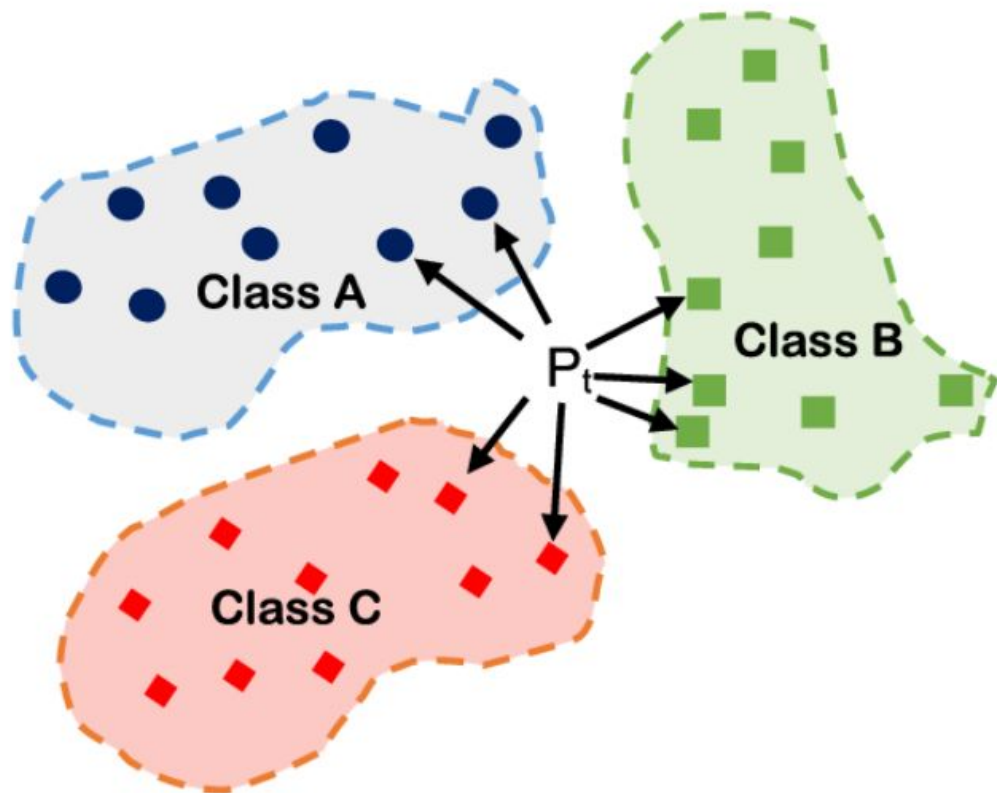
Dezavantajları:

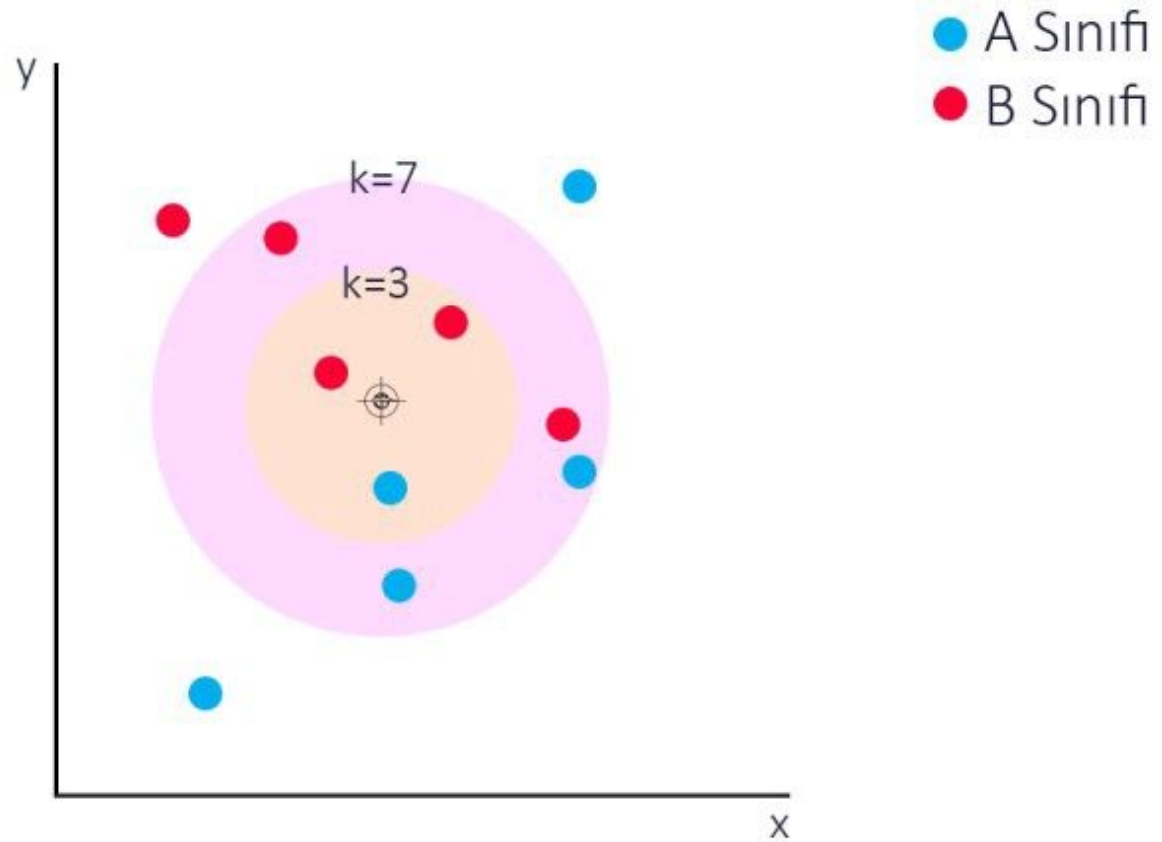
- Gözlem sayısı özellik sayısından azsa, Lojistik Regresyon kullanılmamalıdır, aksi takdirde overfit olabilir.
- Lojistik regresyonun ayırım yapabilmesi için veri setinin doğrusal olarak ayrılabiliriyor olması lazım.

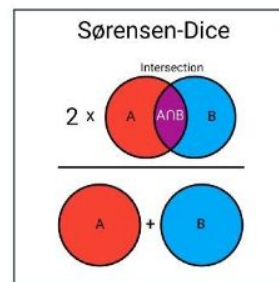
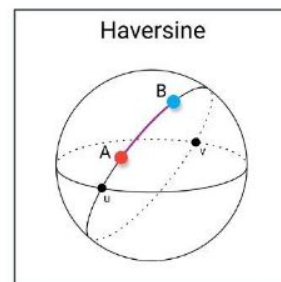
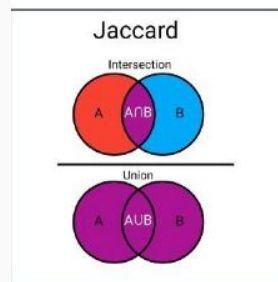
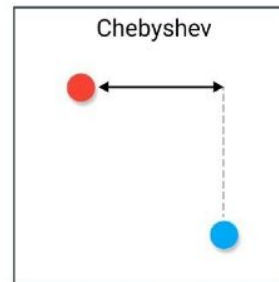
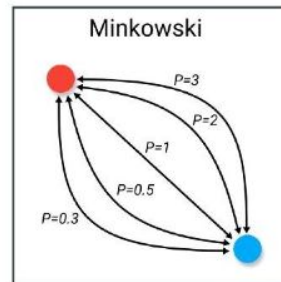
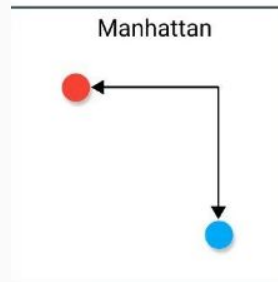
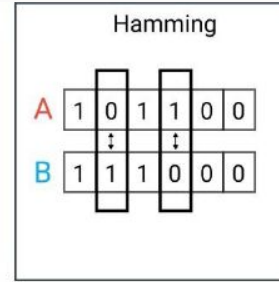
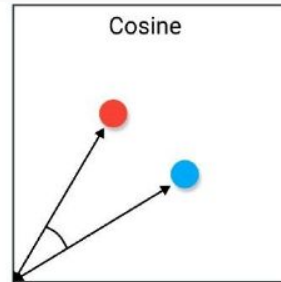
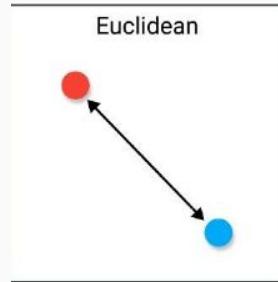
KNN

KNN algoritması, son derece basit, uygulaması kolay olan bir denetimli (supervised) makine öğrenmesi algoritmasıdır. KNN algoritması hem regresyon hem de classification için kullanılır.

Eğitim verilerini öğrenmez, bunun yerine eğitim veri kümesini “ezberler”. Bir tahmin yapmak istediğimizde, tüm veri setinde en yakın komşuları arar.



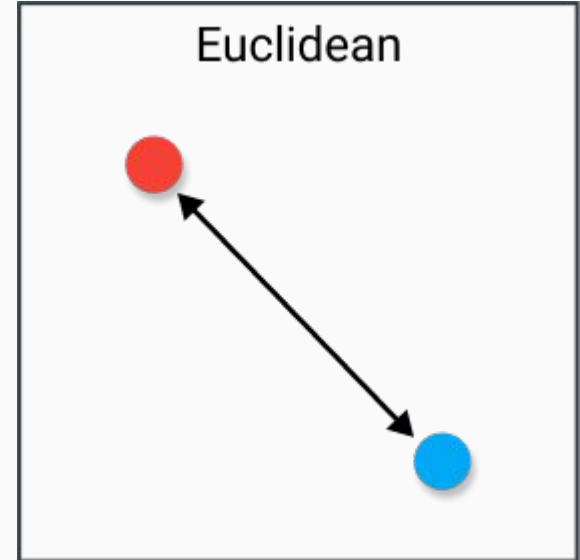




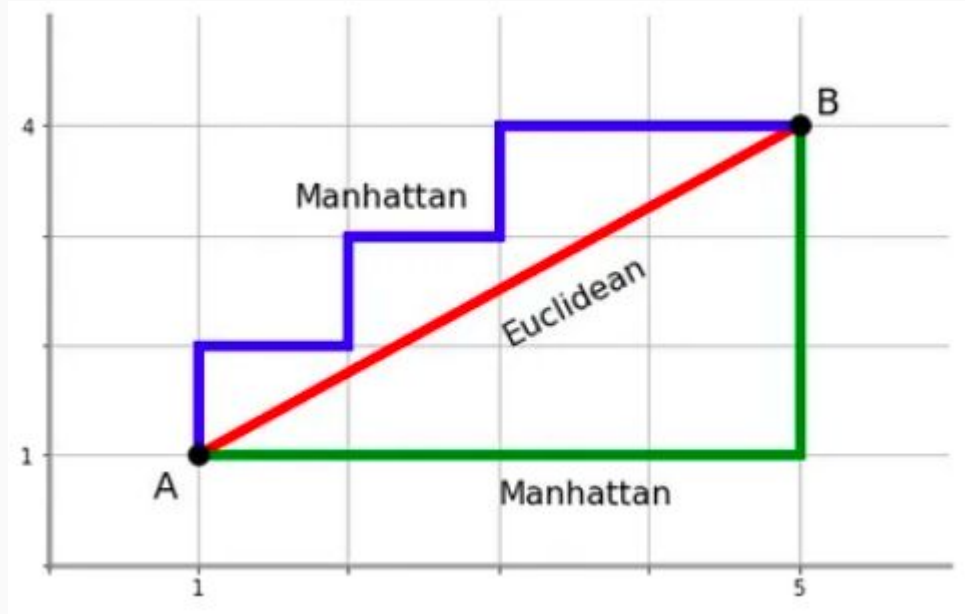
Öklid mesafesi (Euclidean distance) : Matematikte pisagor bağlantısı kullanılarak bulunan iki nokta arasındaki mesafe ölçüm birimidir. Buna göre iki boyutlu düzlemde iki nokta arasındaki mesafe basitçe iki noktanın x ve y koordinatlarının ayrı ayrı farklarının hipotenüs'üne eşittir.

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

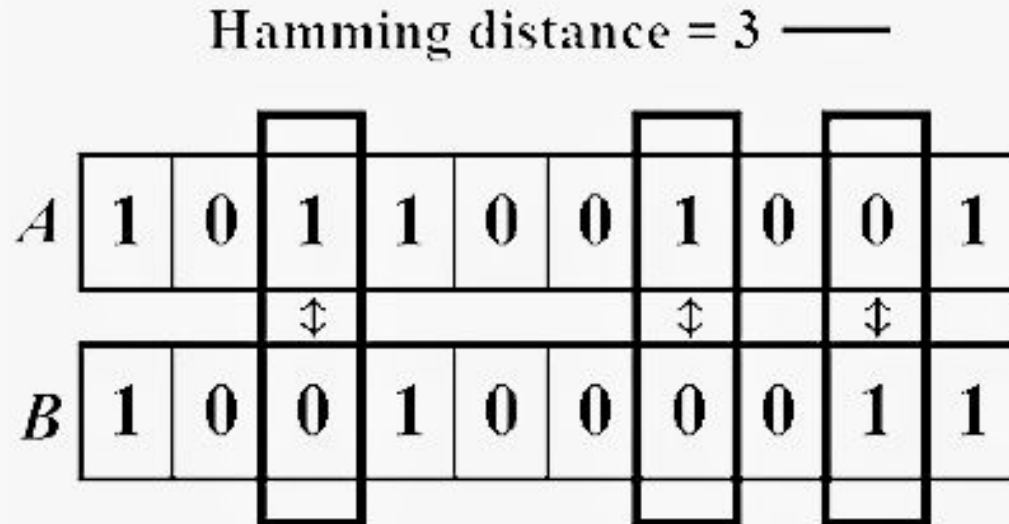
Öklid mesafesi



Manhattan mesafesi : Manhattan mesafesi, iki vektörün mutlak olarak farklarının toplamıdır. Yani iki nokta $(X1, Y1)$ ve $(X2, Y2)$ olarak verilirse Manhattan mesafesi $|X1-X2| + |Y1 -Y2|$ olarak hesaplanır.



Hamming Mesafesi : Hamming mesafesi, iki ikili veri dizisini karşılaştırmak için bir ölçümdür. Eşit uzunluktaki iki ikili diziyi karşılaştırırken, Hamming mesafesi, iki bitin farklı olduğu bit konumlarının sayısıdır. Hamming uzaklığı yöntemi tüm verilere bakar ve kaç özelliğin farklı olduğunun sonucunu verir.



- Algoritmanın çalışmasında bir K değeri belirlenir. Bu K değerinin anlamı bakılacak eleman sayısıdır.
- Bir değer geldiğinde en yakın K kadar eleman alınarak gelen değer arasındaki uzaklık hesaplanır.
- Uzaklık hesaplama işleminde genelde Öklid fonksiyonu kullanılır. Öklid fonksiyonuna alternatif olarak Manhattan, Minkowski ve Hamming fonksiyonları da kullanılabilir.
- Uzaklık hesaplandıktan sonra sıralanır ve gelen değer uygun olan sınıfa atanır.

Avantajları:

- Algoritma basit ve uygulanması kolaydır.
- Bir model oluşturmaya, birkaç parametreyi ayarlamaya veya ek varsayımlar yapmaya gerek yoktur.
- Algoritma çok yönlüdür. Sınıflandırma, regresyon ve arama için kullanılabilir.

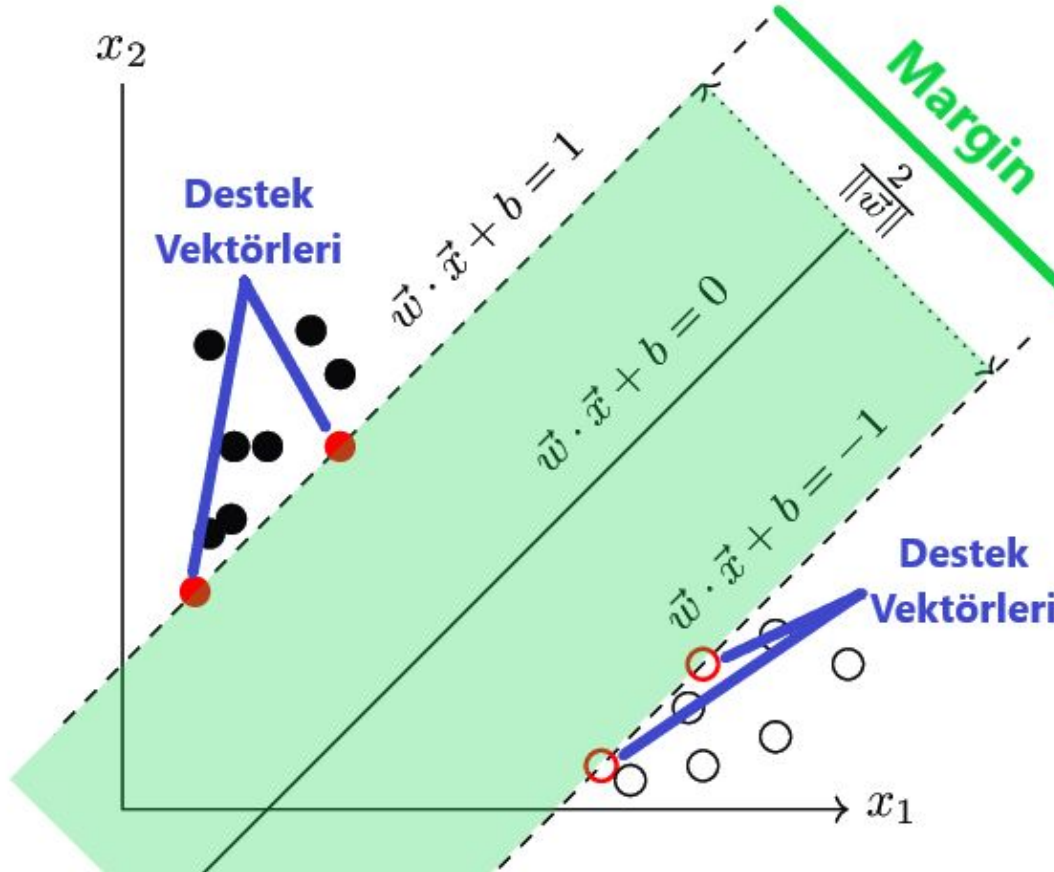
Dezavantajları:

- Örneklerin ve/veya tahmin edicilerin/bağımsız değişkenlerin sayısı arttıkça algoritma önemli ölçüde yavaşlar. Hesaplama yönünden pahalıdır (Computationally expensive)
- Yüksek memory ihtiyacı. Çünkü bütün train datasını hafızada tutar
- Örneklem sayısı artınca, tahmin uzun sürebilir.

SVM

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.

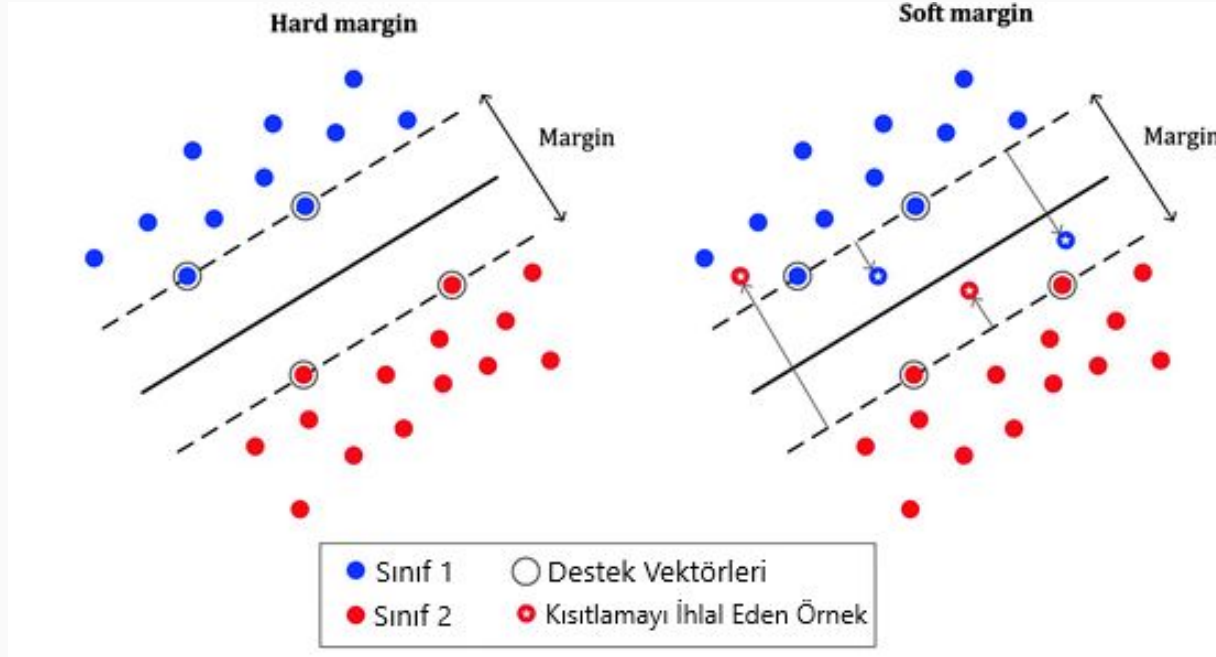
Support Vector Machine Logistic Regression ile benzer bir sınıflandırma algoritmasıdır. Her ikisi de iki sınıfı ayıran en iyi çizgiyi bulmaya çalışırlar. Algoritma çizilecek doğrunun iki sınıfında elemanlarına en uzak yerden geçecek şekilde ayarlanmasını sağlar. Hiçbir parametre almayan (nonparametric) bir sınıflayıcıdır. SVM aynı zamanda doğrusal ve doğrusal olmayan verileride sınıflandırabilir ancak genellikle verileri doğrusal olarak sınıflandırmaya çalışır.



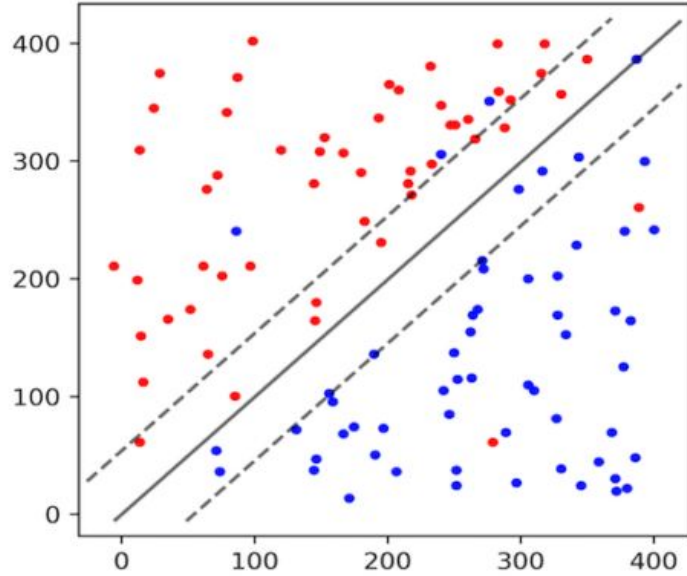
Tabloda siyahlar ve beyazlar olmak üzere iki farklı sınıf var. Sınıflandırma problemlerindeki asıl amacımız gelecek verinin hangi sınıfta yer alacağını karar vermektir. Bu sınıflandırmayı yapabilmek için iki sınıfı ayıran bir doğru çizilir ve bu doğrunun ± 1 'i arasında kalan yeşil bölgeye Margin adı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

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

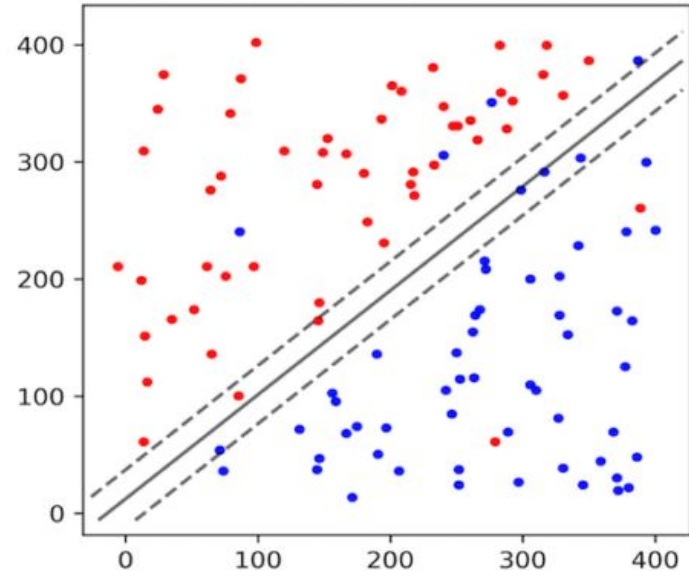
Hard Margin vs Soft Margin



Bazen örneklerimiz Margin bölgesine girebilir. Buna Soft Margin denir. Hard Margin, verimiz doğrusal olarak ayrılabilirse çalışır ve aykırı değerlere karşı çok duyarlıdır. Bu yüzden bazı durumlarda Soft Margin'i tercih etmemiz gerekebilir.



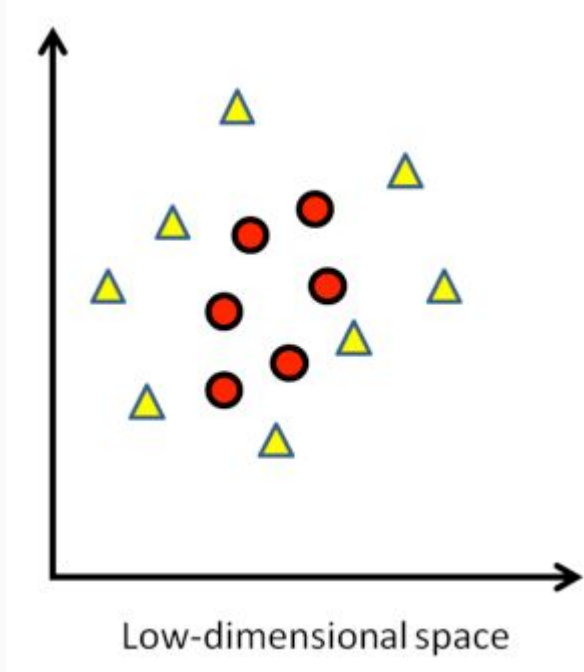
C = 1



C = 100

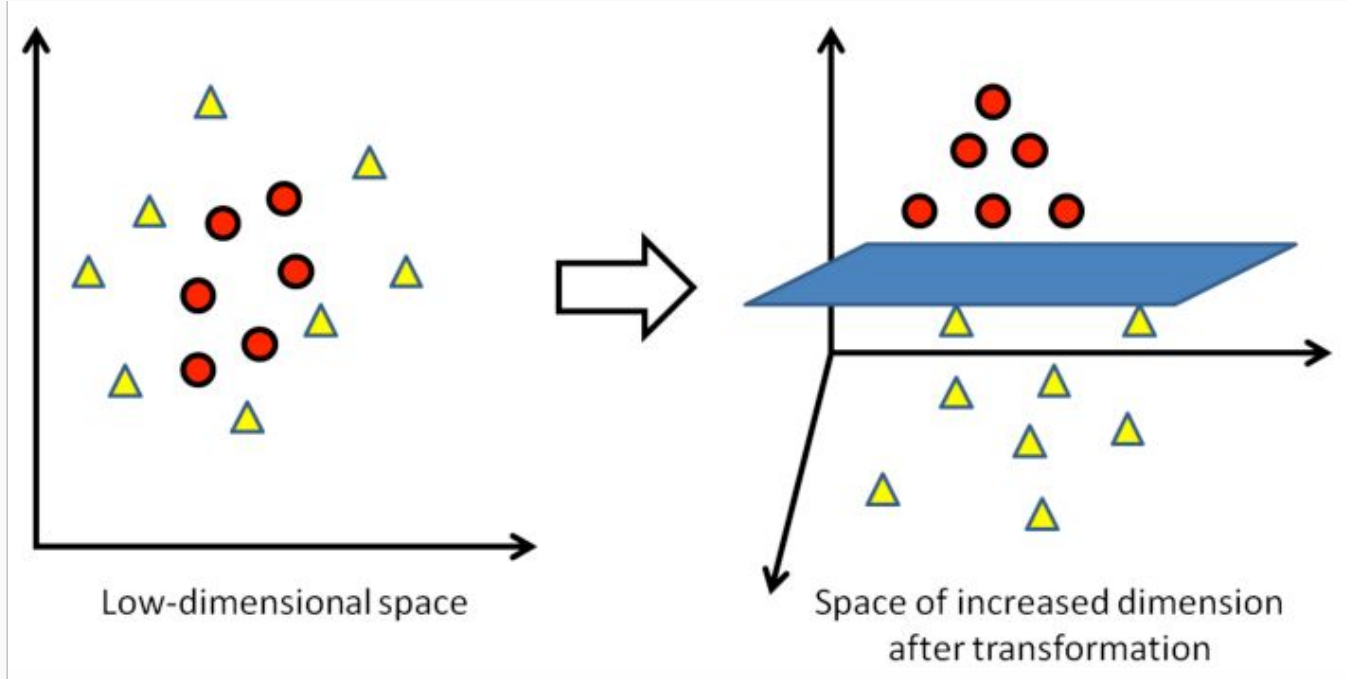
İkisi arasındaki dengeyi SVM içerisindeki C hiperparametresi ile kontrol edebiliriz. C ne kadar büyükse Margin o kadar dardır.

Kernel Trick (Çekirdek Hilesi)

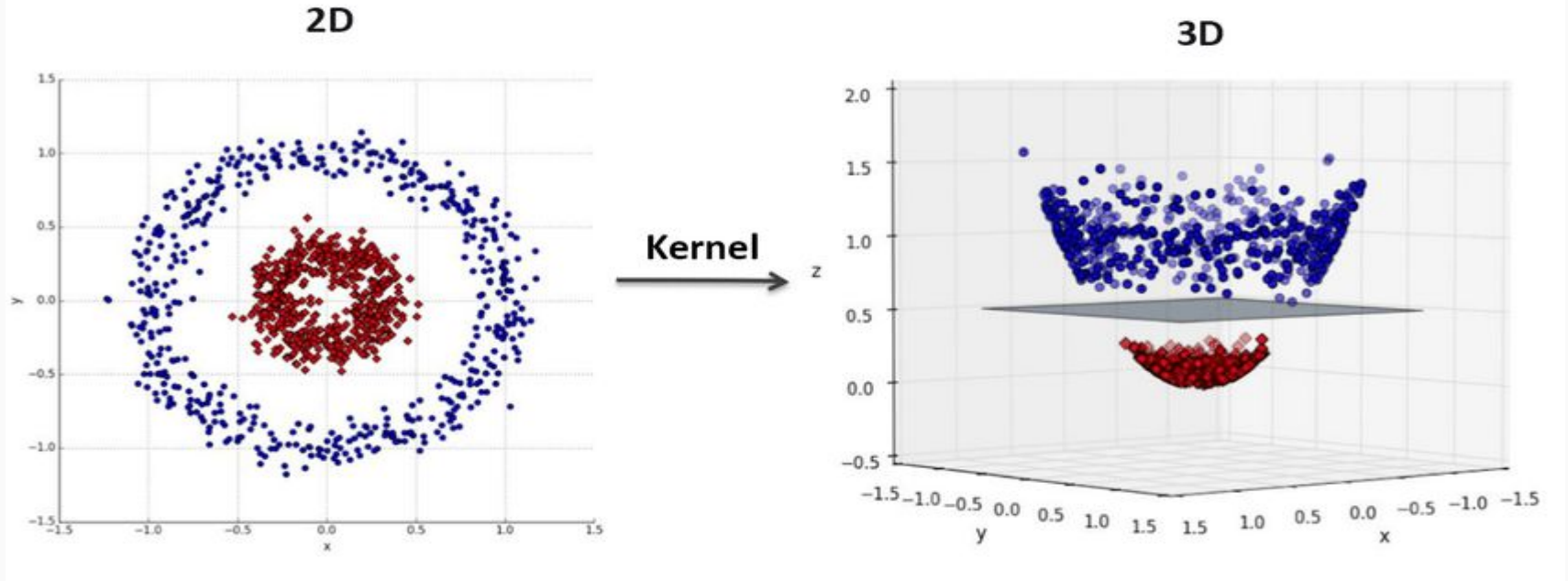


SVM'in verileri doğrusal olarak sınıflandırmaya çalışır ancak bazı durumlarda bu mümkün olmaz. Bu durumdan kurtulmak için çekirdek hilesine (Kernel Trick) başvururuz. Yeni bir boyut oluşturabilirsek doğrusal olarak sınıflandırmamız mümkün olabilir. Örneğin sağdaki şekilde grafikte kırmızı noktaları biraz yukarı kaldırıp (z ekseni) 3. bir boyut oluşturabilirsek SVM ile doğrusal bir çizgi oluşturabiliriz.

Kernel Trick (Çekirdek Hilesi)

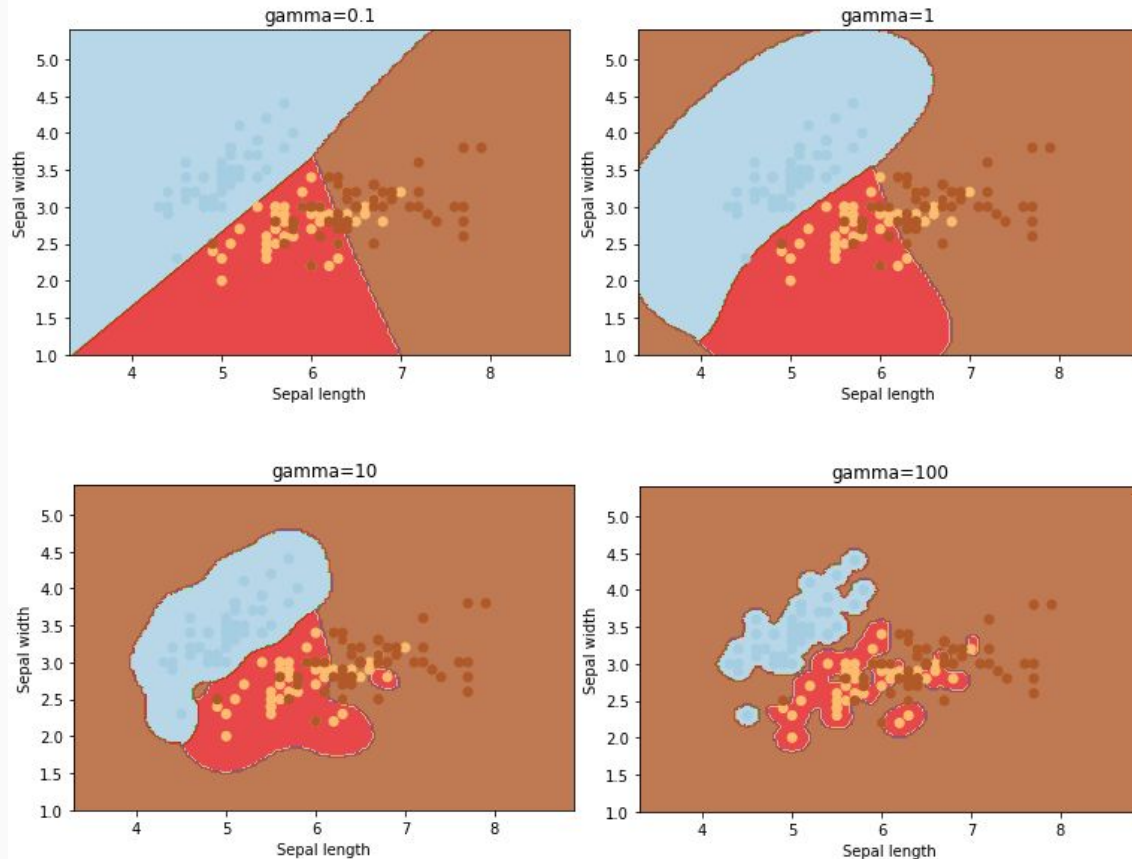


Polynomial Kernel



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.

Gaussian RBF (Radial Basis Function) Kernel



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.

Destek Vektör Makinesinin Avantajları

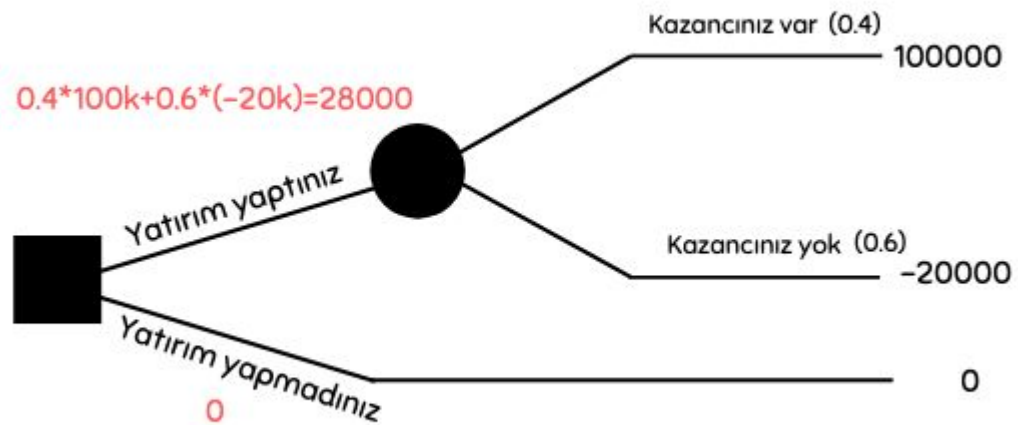
- Birden fazla bağımsız değişkenle çalışabilirler.
- Uyum sorunu diğer yöntemlere göre daha azdır.
- Karmaşık karar sınıflarını modelleyebilmektedir.
- Doğrusal olan ve doğrusal olmayan verilere uyarlanabilir.

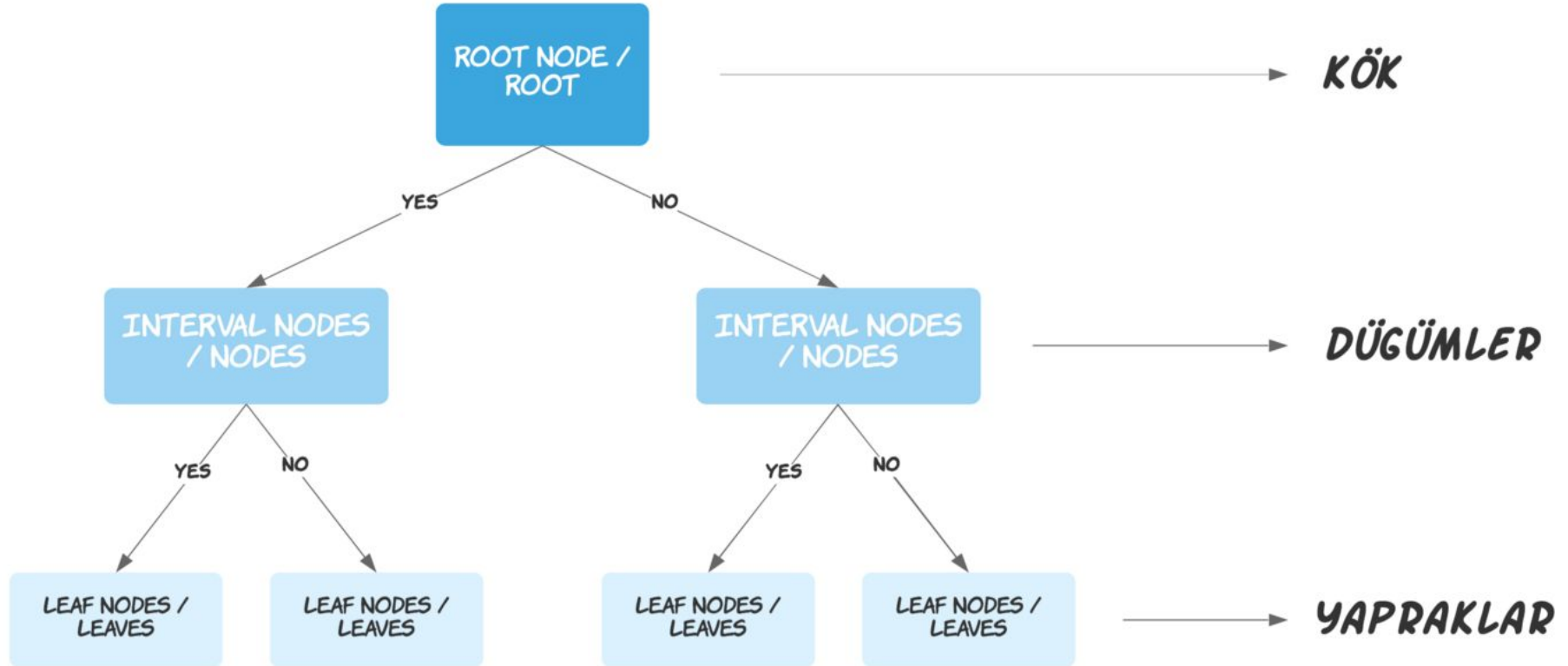
Decision Tree

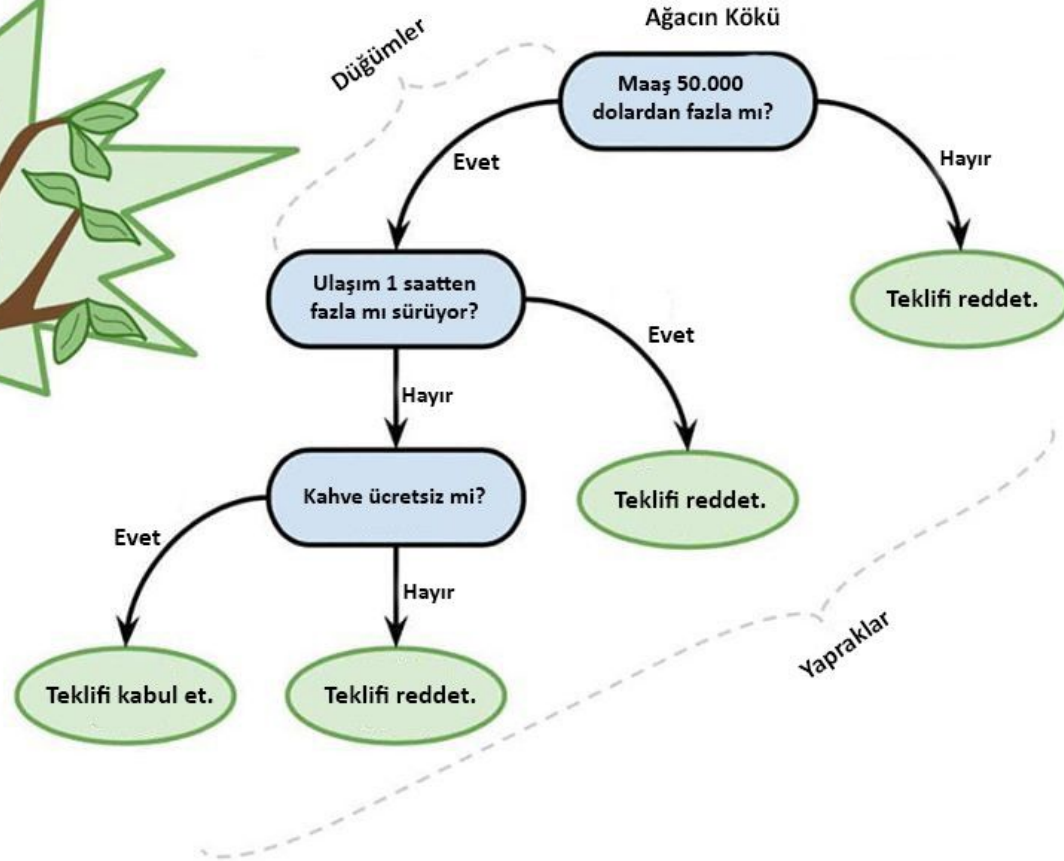
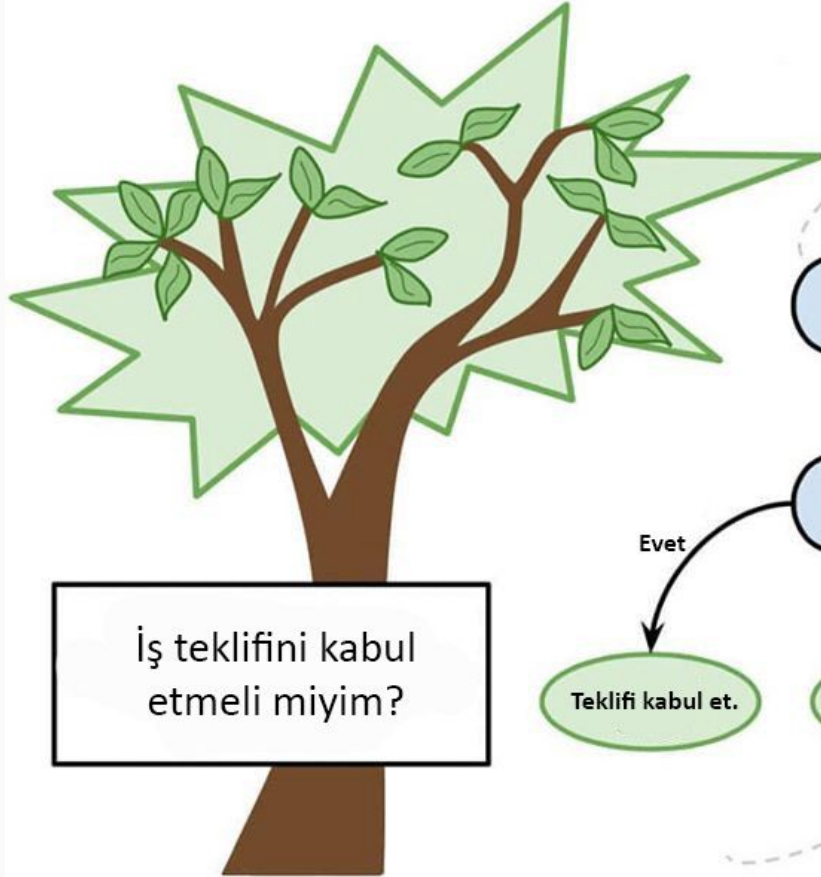
Karar ağaçları, Sınıflandırma ve Regresyon problemlerinde kullanılan, ağaç tabanlı algorithmadan biridir. Karmaşık veri setlerinde kullanılabilir.

Decision Tree

Amacı veri özelliklerinden basit kurallar çıkarıp bu kuralları öğrenerek bir değişkenin değerini tahmin eden modeli oluşturmaktır. Algoritma eksik değerleri desteklemez makineyi eğitmeden önce eksik değerleri (missing value) çözmemiz gerekir.







Karar Ağaçlarının Avantajları:

- Anlaması ve yorumlaması kolaydır. Kullanılan ağaç yapılar görselleştirilebilir.
- Az oranda bir veri hazırlığına ihtiyaç duyar. Fakat unutulmamalıdır ki bu model kayıp değerleri desteklememektedir.
- Kullanılan ağacın maliyeti, ağacı eğitmek için kullanılan veri noktalarının sayısıyla logaritmiktir.
- Hem sayısal hem de kategorik verileri işleyebilir.
- Çok çıktılı problemleri ele alabilmektedirler.
- İstatistiksel testler kullanılarak bir modelin doğrulanması mümkündür.
- Karar ağaçları, parametrik olmayan bir yöntem olarak düşünülebilir. Yani uzay dağılımı ve sınıflandırma yapısı hakkında bir yaklaşıma sahip değildir.

Karar Ağaçlarının Dezavantajları:

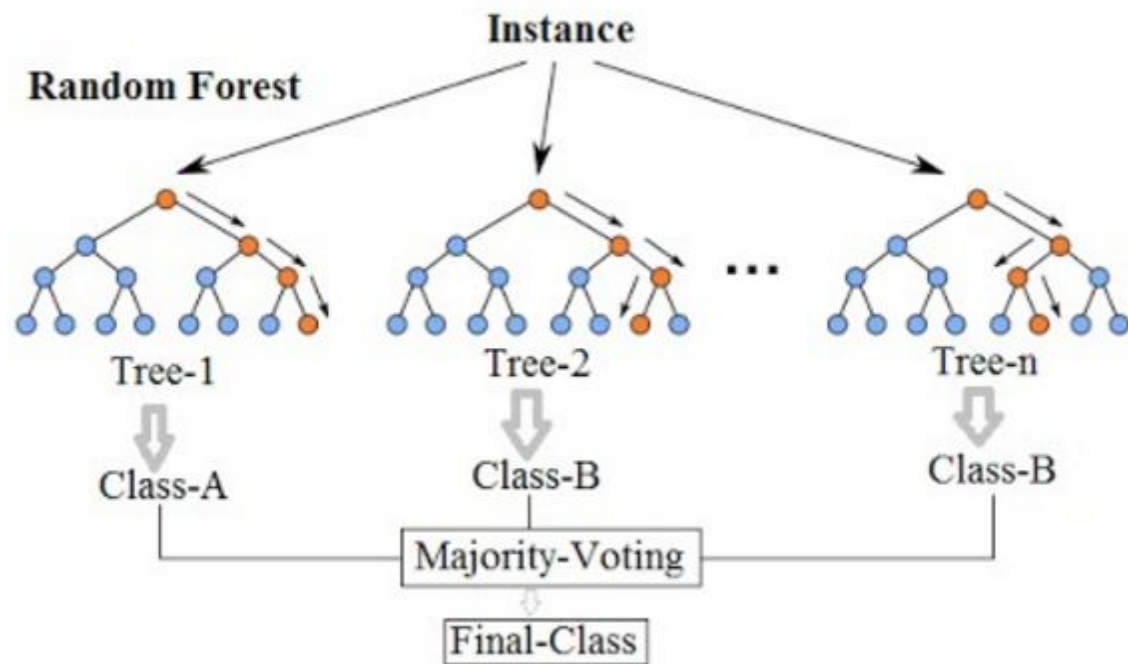
- Veriyi iyi bir şekilde açıklamayan aşırı karmaşık ağaçlar üretilebilir. Bu durumda ağaç dallanması takip edilemeyebilir.
- Ezbere öğrenme yaşanabilir (“over-fitting”). Bu problemin çözümü için model parametrelere kısıtlamalar ve budama gibi yöntemler kullanılabilir. Budama işlemi, az sayıda nesneyi barındıran yaprak düğümlerin karar ağacı grafiğinden atılmasını ifade etmektedir

- 1-) Sınıflandırma ve Regresyon problemlerinde çok çıktılı bir şekilde çalışabilir.
- 2-) Karmaşık veri setlerinde kullanılabilir.
- 3-) Scale etmeye ve çok fazla Veri Ön İşleme'ye gerek duymaz.
- 5-) Karar Ağacı, önceden tanımlanmış olan max_depth hiperparametresine ulaştığında veya daha saf bir alt küme elde edemediğinde durur.
- 6-) Gini ve Entropi arasında çok büyük bir fark yoktur. Entropi daha dengeli bir ağaç çıkarmaya meyilli iken Gini frekansı fazla olan sınıfı ayırtırmaya meyillidir.
- 7-) Model overfit (aşırı uyum) olmuşsa genellikle önce max_depth hiperparametresi düşürülür.

Random Forest

Rassal ormanlar birden fazla karar ağacının birlikte (ensemble) çalışmasından oluşur. Birbiri arasında korelasyon olmayan modellerin bir araya gelip çalışması, tek başına çalışan herhangi bir modele göre daha iyi performans gösterir, ensemble learning buna dayanır. Birbirlerinin arasında korelasyon olmaması, her ağacın kendi hatalarını diğerlerinden sakınmasına yardımcı olur. Korele olmayan sonuçların daha iyi performans sağlamasını karar teorisinden bir örnekle açıklayalım.

Random Forest Simplified



Teşekkürler!