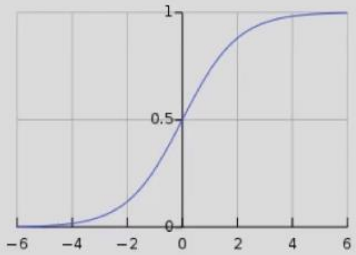
Sınıflandırma Modelleri

- Lojistik Regresyon
- K-En Yakın Komşu (KNN)
- Destek Vektör Regresyonu (SVR)
- Çok Katmanlı Algılayıcılar (ANN)
- Classification and Regression Tress (CART)
- Bagging (Bootstrap Aggregation)
- Naive Bayes
- Random Forests (RF)
- Gradient Boosting Machines (GBM))
- Extreme Gradient Boosting (XGBoost)
- LightGBM

Hedeflenen bağımlı değişken bir kategorik değişkendir
Amaç, yeni bir gözlem değeri geldiğinde onun hangi sınıfa ait olduğunu tahmin etmektir.



Lojistik Regresyon

Amaç sınıflandırma problemi için bağımlı ve bağımsız değişkenler arasındaki ilişkiyi tanımlayan doğrusal bir model kurmaktır.

Tahmin etmek istenilen hedef değişkeni kategorik ise doğrusal regresyon yerine lojistik Regresyon kullanabilir. Kategorik hedef değişkeni bir kişinin hasta olup olmadığı, bir kişiye kredi verilip verilmeyeceği veya bir mailin spam olup olmadığı örneklerinde olduğu gibi ***iki değerden birini (0,1) aldığı durumda binary (ikili) lojistik regresyon kullanılabilir***.

Lojistik Regresyon

- Bağımlı değişken kategoriktir.
- Adını bağımlı değişkene uygulanan logit dönüşümden alır.
- Doğrusal Regresyonda aranan varsayımlar burada aranmadığı için daha esnek kullanılabilirliği vardır.
- Bağımlı değişkenin 1 olarak tanımlanan değerinin gerçekleşme olasılığı hesaplanır. Dolayısıyla bağımlı değişkenin alacağı değer ile ilgilenilmez.
- Lojistik fonksiyonu sayesinde üretilen değerler 0-1 arasında olur.

Lojistik Regresyon

- Doğrusal regresyon çıktısı sürekli bir değer ölçeğidir. Örneğin buna sayılar, kilometreler, fiyat ve ağırlık dahildir.

- Buna karşılık, lojistik regresyon modeli çıktı değeri, sabit bir kategorik olayın meydana gelme olasılığıdır.

Lojistik Regresyon

- Linear regresyon bağımlı değişken ile bağımsız değişkenler arasındaki ilişkiyi tahmin etmek için kullanılırken, lojistik regresyon ise iki sınıflı sınıflandırma problemlerinde olasılıkları modellemek için tercih edilir.
- Linear (doğrusal) regresyon tahminleri nicelik ifade ederken, lojistik regresyon tahminleri iki sınıfın olasılıklarını yansıtır.

Naïve Bayes

Olasılık temelli bir modelleme tekniğidir. Amaç belirli bir örneğin her bir sınıfa ait olma olasılığının koşullu olasılık temelli hesaplanmasıdır.

- E ticaret gibi çok kategorili sınıf yapılarının olduğu ve burada bazı modelleme çalışmalarına gerek duyulduğu durumlarda iyi sonuçlar verebilmektedir.

- İlgilenilen hedef değişkeninin çok sınıflı olduğu ve buna karşılık veri setinde kategorik değişken sayısı sürekli değişken sayısı kadar çoksa bu algoritma başarılı sonuçlar verebilir.

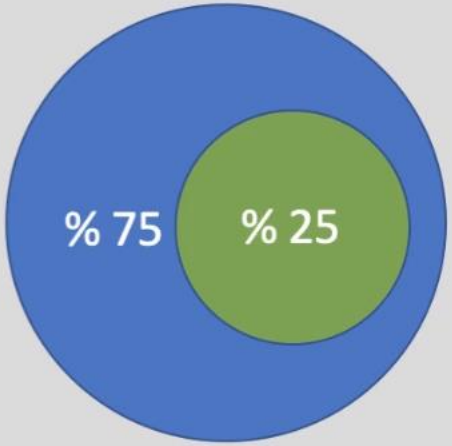
Naïve Bayes

Yeni kredi Başvurusu

Aylık Gelir: 2000 TL

Bu kişi krediyi ödeyebilir mi?

Geçmiş Veriler İncelendiğinde



Ödemiş
Ortalama Aylık Gelir: 6000 TL

Ödeyememiş
Ortalama Aylık Gelir: 1800 TL

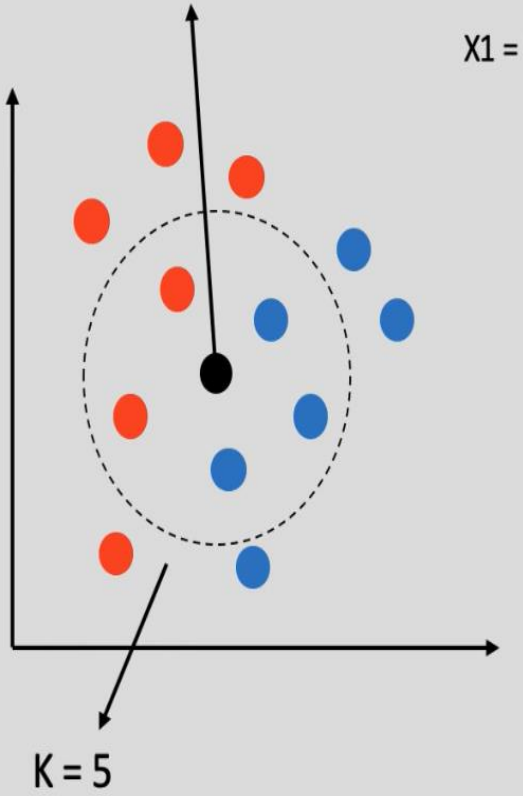
$$P\,(C\,|x_1, \ldots, x_n) = P(C) \sum_{i=1}^{n} P(x_i\,|C)$$

K-En Yakın Komşu (KNN)

Tahminler gözlem benzerliğine göre yapılır.

Bana arkadaşını söyle sana kim olduğunu söyleyeyim.

K-En Yakın Komşu



Y tahmini nedir?

X1 = 50, X2 = 230

Y	X1	X2
1	56	241
0	85	250
.	.	.
.	.	.
0	56	231

$$d(i,j) = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2}$$

En yakın K adet gözlemin y değerlerinin en sık gözlenen frekansı tahmin edilen sınıf olur.

Öklid ya da benzeri bir uzaklık hesabı ile her bir gözleme uzaklık hesaplanır.

K-En Yakın Komşu

KNN Basamakları

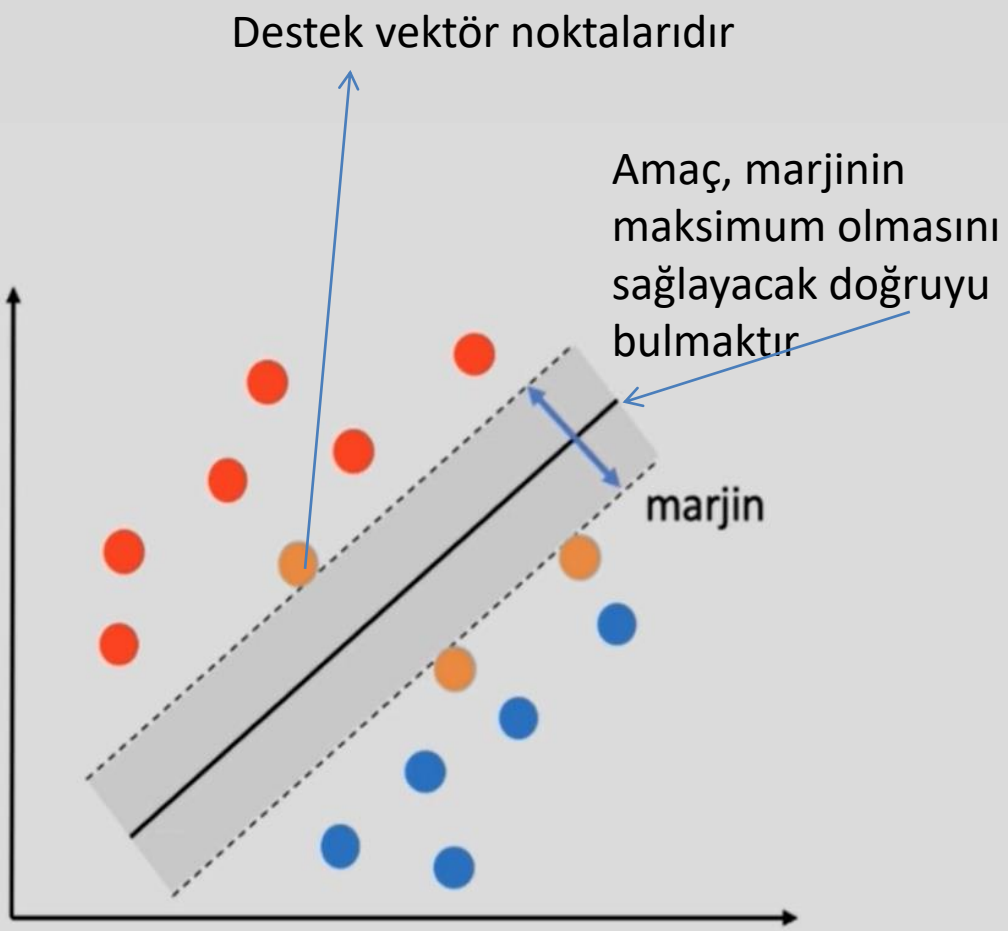
- Komşu sayısını belirle (K)
- Bilinmeyen nokta ile diğer tüm noktalar ile arasındaki uzaklıkları hesapla
- Uzaklıkları sırala ve belirlenen k sayısına göre en yakın olan k gözlemi seç
- Sınıflandırma ise en sık sınıf, regresyon ise ortalama değeri tahmin değeri olarak ver.

(Support Vector Machine)

Destek Vektör Makineleri (SVM)

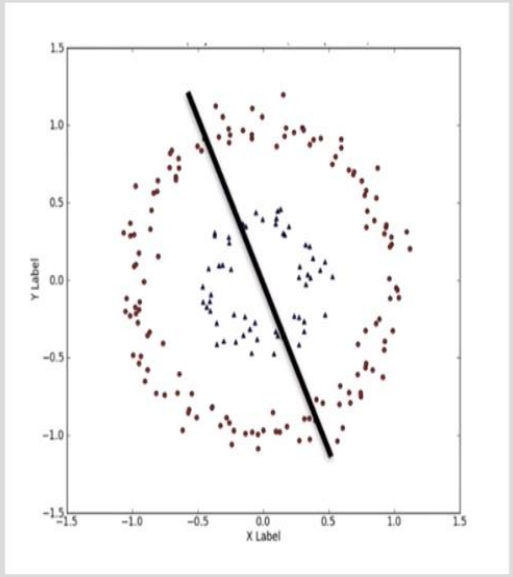
Amaç iki sınıf arasındaki ayrımın optimum olmasını sağlayacak hiper-düzlemi bulmaktır.

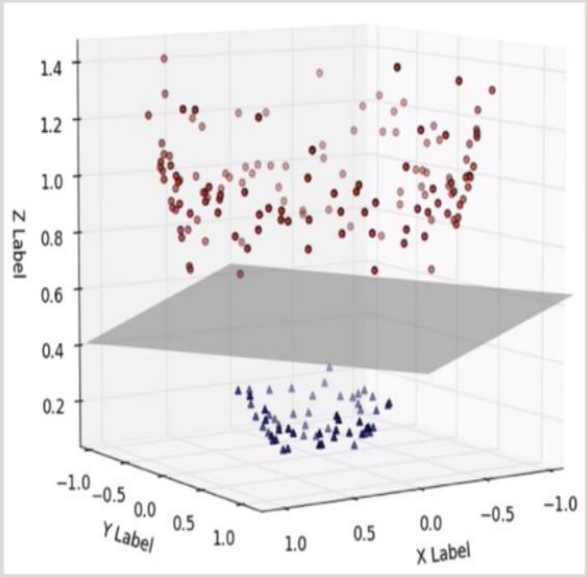
Destek Vektör Makineleri (SVM)



C: Ceza parametresi, aykırı gözlemleri kontrol altında tutup,
oluşturulacak olan ayrımın kontrol edilmesi ile ilgili bir parametredir.

Doğrusal Olmayan SVM



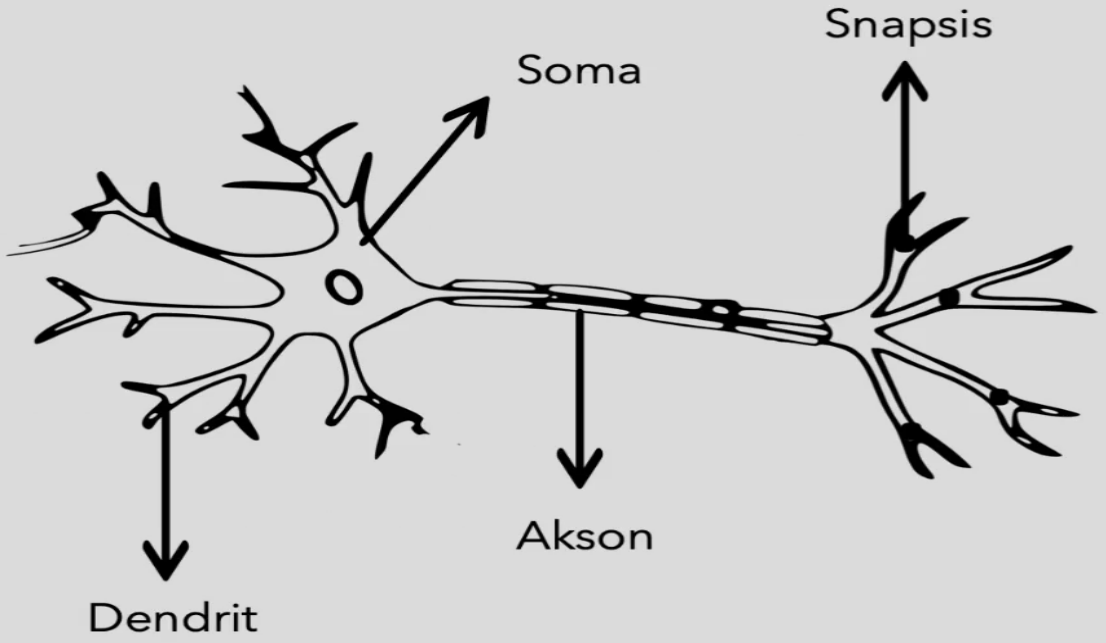


Kernel trick (çekirdek hilesi) ile çok boyutlu uzaya çevrilerek ayrıştırılabilir hale getirilmiştir

Yapay Sinir Ağları

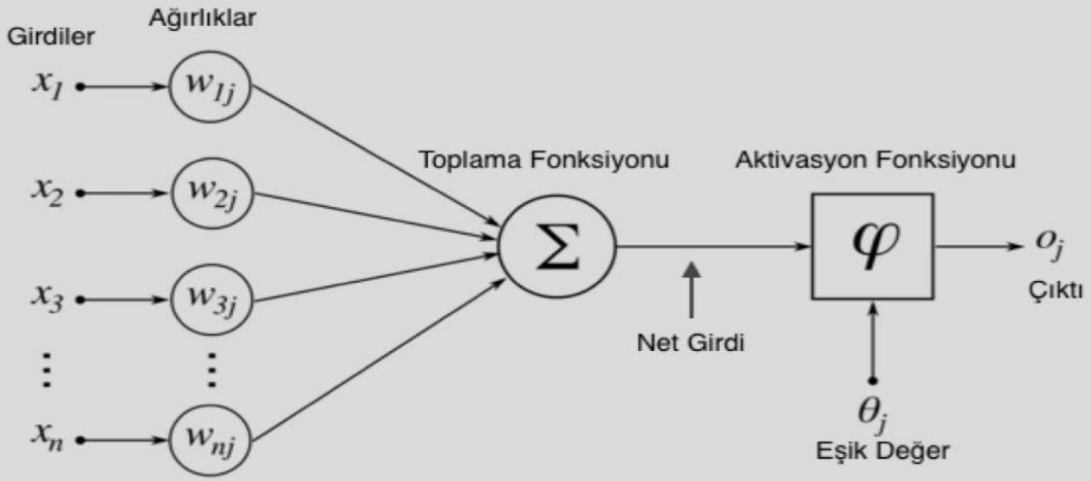
İnsan beyninin bilgi işleme şeklini referans alan sınıflandırma ve regresyon problemleri için kullanılabilen kuvvetli makine öğrenmesi algoritmalarından birisidir.

Sinir Hücresi



- Dendrit'ler alınan sinyalleri çekirdeğe iletmekle görevlidirler.
- Hücre çekirdeği ile dentritler arasında bir iletişim gerçekleşmektedir. Dentritlere göre iletişim farklılık göstermekte ve hücre çekirdeği alınan sinyaller konusunda seçicilik yapabilmektedir. Bu durum dentritlerin hücre çekirdeğine ilettiği bilgilerin hücre çekirdeği tarafından **farklı ağırlıklar ile dikkate alınabileceğini ifade eder**.
- Somaların görevi dentritler tarafından iletilen sinyalleri bir merkezde toplamaktır.
- Aksonun görevi çekirdekten gelen bilgileri alarak diğer hücrelere iletmektir.
- Fakat sinyaller diğer hücrelere aktarılmadan önce sinapsislerde bir ön işleme tabi tutulmaktadır. Sinapsislerin görevi gelen sinyalleri belirli bir eşik değerine denk getirecek şekilde değiştirmek, işlemek ve sonrasında diğer hücrelere aktarmaktır.

Yapay Sinir Hücresi



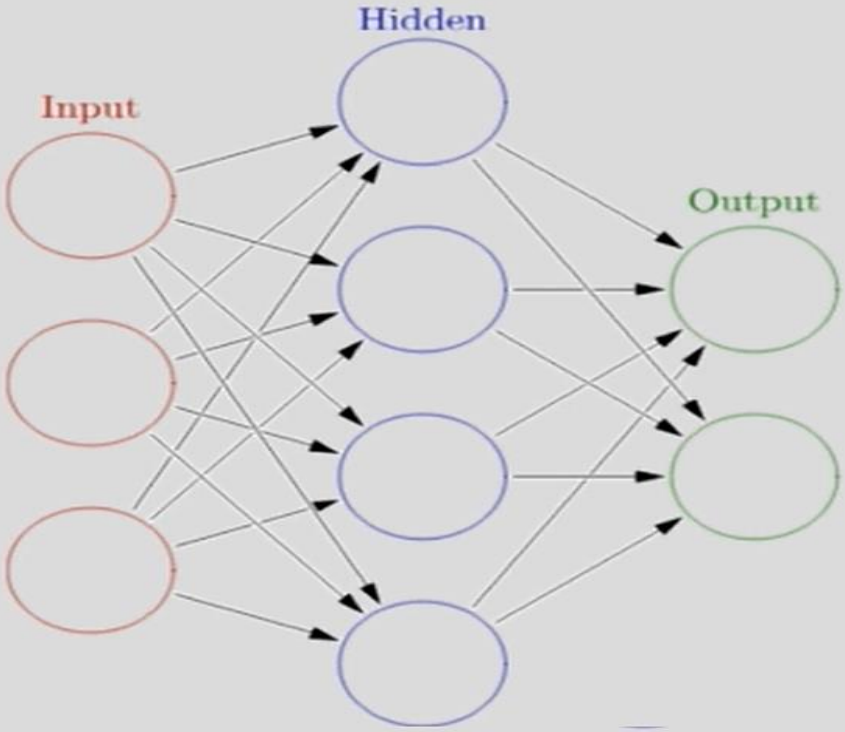
- Girdiler = Dentritler. X1, X2, X3 … Xn →Bağımsız Değişkenlerin Değerleridir
- Toplama Fonksiyonu = Soma
- Dentritler ile soma arasındaki etkileşimde bu etkileşimin ağırlıkları belirleniyor. Yani girdi olarak verilen değişkenlerin değerlerinin çıktıya (bağımlı değişkene) olan etkileri kontrol ediliyor.
- Aktivasyon Fonksiyonu = Sinapsis

Amacımız hatayı minimum yapacak şekilde bir matematiksel formül veya kural seti oluşturmaktır. Amacımız her zaman aynıdır. Bağımlı değişkeni bulmak üzere bu ağırlıkları öyle bir bulmalıyız ki neticesinde minimum hataya sahip olalım.

Başlangıçta rastgele atanan ağırlık değerleri ile ağ çalışmaya başlıyor ve belirli bir iterasyon sayısınca optimum ağırlıklar bulunmaya çalışılıyor. Her iterasyonda bulunan hatalar (gerçek ve tahmin edilen arasındaki) geriye yayılarak optimum değerler bulunmaya çalışılıyor. (Geriye yayılım Algoritması). Bu ağırlıkların değişmesi aslında öğrenme işlemi olmaktadır.

Bu basit sinir hücreleri bir araya gelerek yapay sinir ağlarını oluşturur.

Yapay Sinir Ağı



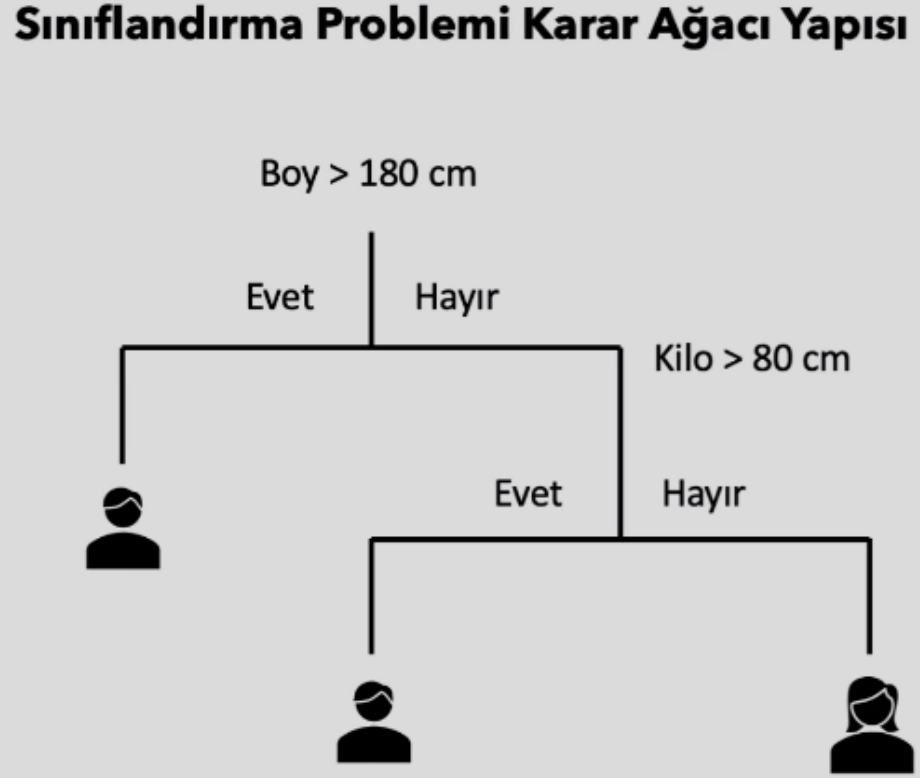
- Ağı oluşturan birimlere nöron adı verilir.
- Ağ Input (Girdi), Hidden (Gizli) ve Output (Çıktı) katmanlarından oluşur.
- Birden fazla gizli katman olabilir. Her katmanda birbirinden farklı sayıda nöron bulunabilir.
- Eğitim sırasında yine girdi değerleri ile çıktı değerleri birlikte sunularak elde edilen tahmin değerleri gerçek değerler ile karşılaştırılır. Bunun sonucunda bir epok yapısıyla hatalar geriye doğru yayılarak ağırlıklar ve katmanların optimum değerleri bulunmaya çalışılarak çıktı değeri sınıflandırma veya regresyon problemi için elde edilmeye çalışılır.
- Yapay Sinir Ağlarına, Çok Katmanlı Algılayıcılar ismi de verilir.
- Çok Katmanlı Algılayıcılarda en sık kullanılan algoritma geriye yayılım algoritmasıdır. Bu algoritmada Delta kuralı uygulanır.

Classification and Regression Trees (CART)

Amaç veri seti içerisindeki karmaşık yapıları basit karar yapılarına dönüştürmektir.

Heterojen veri setleri belirlenmiş bir hedef değişkene göre homojen alt gruplara ayrılır.

Sınıflandırma Ağaçları



Random Forests Sınıflandırma

Temeli birden çok karar ağacının ürettiği tahminlerin bir araya getirilerek değerlendirilmesine dayanır.

Random Forests

- Bagging (Breiman, 1996) ile Random Subspace (Ho, 1998) yöntemlerinin birleşimi ile oluşmuştur.
- Ağaçlar için gözlemler bootstrap rastgele örnek seçim yöntemi ile **değişkenler random subspace yöntemi ile seçilir.**
- Karar ağacının her bir düğümünde en iyi dallara ayırıcı (bilgi kazancı) değişken tüm değişkenler arasından rastgele seçilen daha az sayıdaki değişken arasından seçilir.
- Ağaç oluşturmada veri setinin 2/3'ü kullanılır. Dışarıda kalan veri ağaçların performans değerlendirmesi ve değişken öneminin belirlenmesi için kullanılır.
- Her düğüm noktasında rastgele değişken seçimi yapılır. (regresyon'da p/3, sınıflama'da karekök p)
- Nihai tahmin için ağaçlardan tahmin değerleri talep edilirken her bir ağacın daha önce hesaplanan hata oranları göz önüne alınarak ağaçlara ağırlık verilir.

Gradient Boosting Machines

AdaBoost'un sınıflandırma ve regresyon problemlerine kolayca uyarlanabilen genelleştirilmiş versiyonudur.

Artıklar üzerine tek bir tahminsel model formunda olan modeller serisi kurulur.

Boosting Yöntemlerine Giriş

Zayıf öğrenicileri bir araya getirip güçlü bir öğrenici ortaya çıkarmak fikrine dayanır.

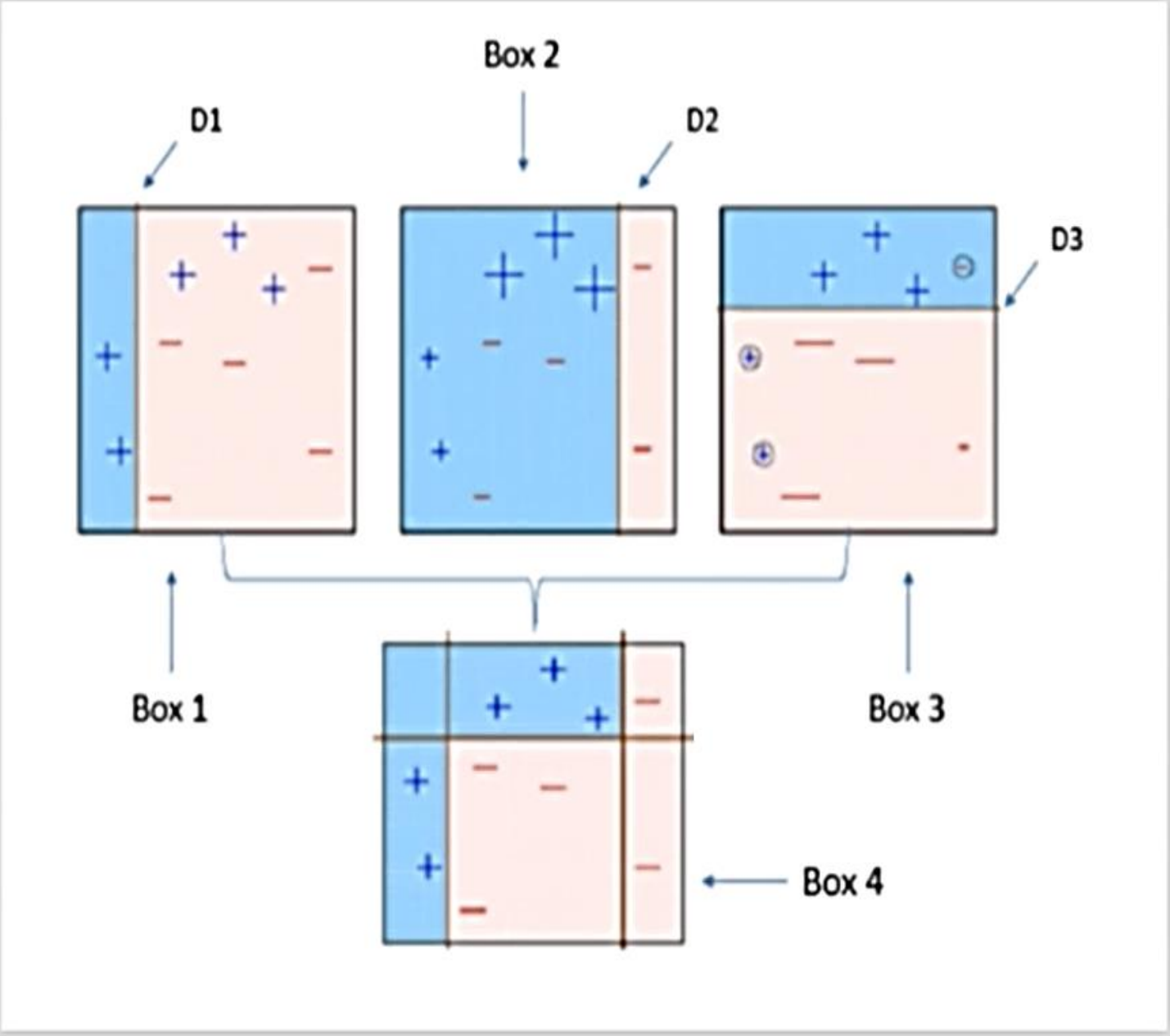
Kearns ve Valiant 1990

Adaptive Boosting (AdaBoost)

Zayıf sınıflandırıcıların bir araya gelerek güçlü bir sınıflandırıcı oluşturması fikrini hayata geçiren algoritmadır.

Schapires ve Freund 1996-1999

otive
sting
oost)

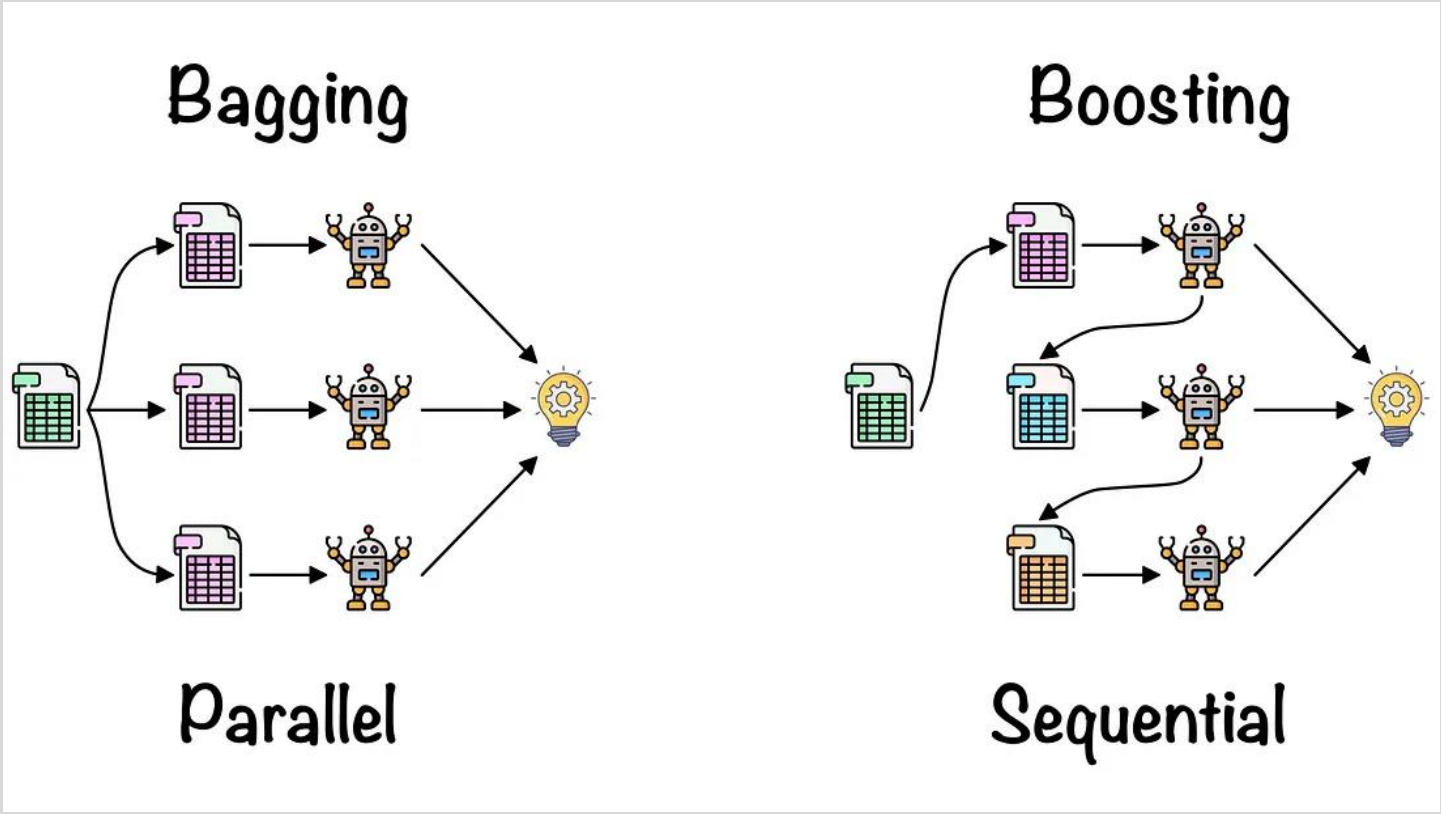


Gradient Boosting Machines

- Gradient boosting tek bir tahminsel model formunda olan modeller serisi oluşturur.
- Seri içerisindeki bir model serideki bir önceki modelin tahmin artıklarının/ hatalarının (residuals) üzerine kurularak (fit) oluşturulur.
- GBM diferansiyellenebilen herhangi bir kayıp fonksiyonunu optimize edebilen Gradient descent algoritmasını kullanmakta.
- GB bir çok temel öğrenici tipi (base learner type) kullanabilir. (Trees, linear terms, splines,...)
- Cost fonksiyonları ve link fonksiyonları modifiye edilebilirdir.
- Boosting + Gradient Descent

Bagging vs. Boosting

- Bagging’te ağaçlar bağımsızdı, Boostingde ise ağaçların birbirine bağımlılığı söz konusudur.



Bagging vs. Boosting

- Boosting algoritmalarında ağaçlar birbirlerine bağlı ve her bir sınıflandırıcı, önceki sınıflandırıcıların başarısı dikkate alınarak eğitilir. Her bir eğitimden sonra ağırlıklar yeniden paylaştırılır. Yanlış sınıflandırılmış verilerin ağırlıkları artırılır. En iyi güçlendirme teknikleri AdaBoost, Gradient Boosting ve XgBoost v.b

- Bagging yöntemine göz attığımızda ise ağaçların birbirlerinden bağımsız olduğunu söyleyebiliriz. Bagging yönteminde kullanılan en popüler makine öğrenmesi tekniği Rastgele Orman'dır. Rastgele ormanda birden fazla karar ağacı kullanılır.

eXtreme Gradient Boosting (XGBoost)

XGBoost, GBM'in hız ve tahmin performansını arttırmak üzere optimize edilmiş; ölçeklenebilir ve farklı platformlara entegre edilebilir halidir.

Tianqi Chen 2014

XGBoost

- R, Python, Hadoop, Scala, Julia ile kullanılabilir.
- Ölçeklenebilirdir.
- Hızlıdır.
- Tahmin başarısı yüksektir.
- Bir çok kaggle yarışmasında başarısını kanıtlamıştır.

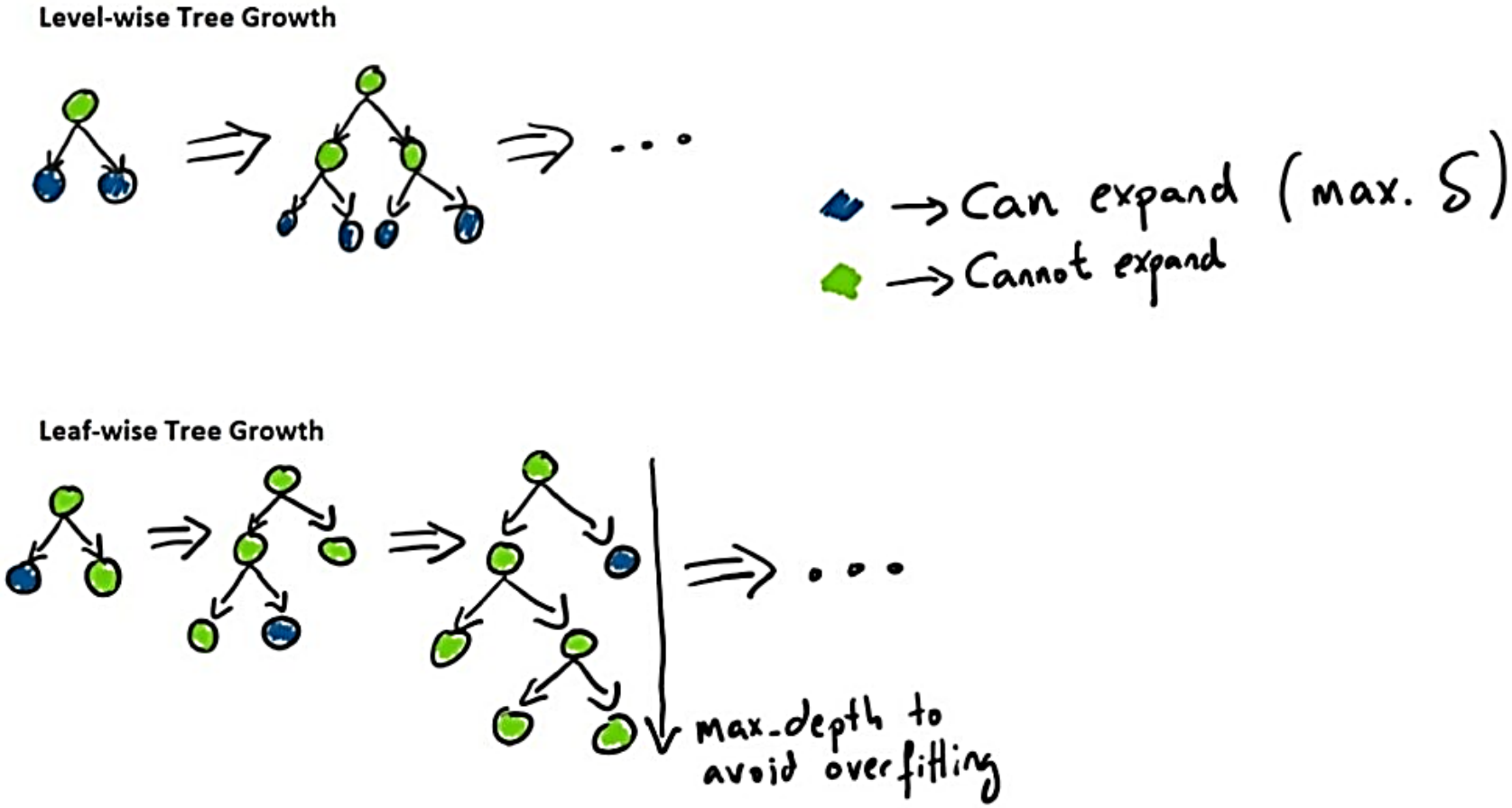
Light GBM

Light GBM, XGBoost'un eğitim süresi performansını arttırmaya yönelik geliştirilen bir diğer GBM türüdür.

Light GBM

- Daha performanslı
- Level-wise büyüme stratejisi yerine Leaf-wise büyüme stratejisi
- Breadth-first search (BFS) yerine depth-first search (DFS)

Xgboost ve light GBM arasındaki en büyük farklılık karar ağacını oluşturma şekilleridir. **XGBoost seviye bazlı büyüme yöntemini (Level-wise tree growth) tercih ederken**, **LightGBM yaprak bazlı büyüme yöntemini (leaf-wise tree growth) tercih ediyor**. Seviye bazlı yaklaşım yatay olarak büyür ve bir alt seviyeye geçmek için mevcut bulunulan seviyenin açılmasını bekler. Yaprak bazlı yaklaşım ise gidebildiği kadar diklemesine ağacı açmaya devam eder, maksimum derinliğe ulaştığında yukarıdan itibaren diğer dalı diklemesine açmaya başlar.



Ağaçların büyüme şekilleri (Felipe Sulser)

LightGBM 10 kat daha hızlı iken, XGBoost ile üretilen modeller %1-2 aralığında daha başarılıdır.

TEŞEKKÜRLER