

Makine Öğrenmesi Algoritmaları

Kurala dayalı çıkarımlar yapar.İlişkilendirme kuralı öğrenme,büyük veri tabanlarındaki değişkenler arasındaki ilişkileri keşfetmek için kural tabanlı bir makine öğrenme yöntemidir.Makine öğrenmede veri tabanlarında keşfedilen güçlü kuralların bir miktar “ilginçlik” ölçüsü kullanılarak belirlenmesi amaçlanmıştır.

Makine Öğrenmesinin Algoritmalarını Temel Bileşeni:

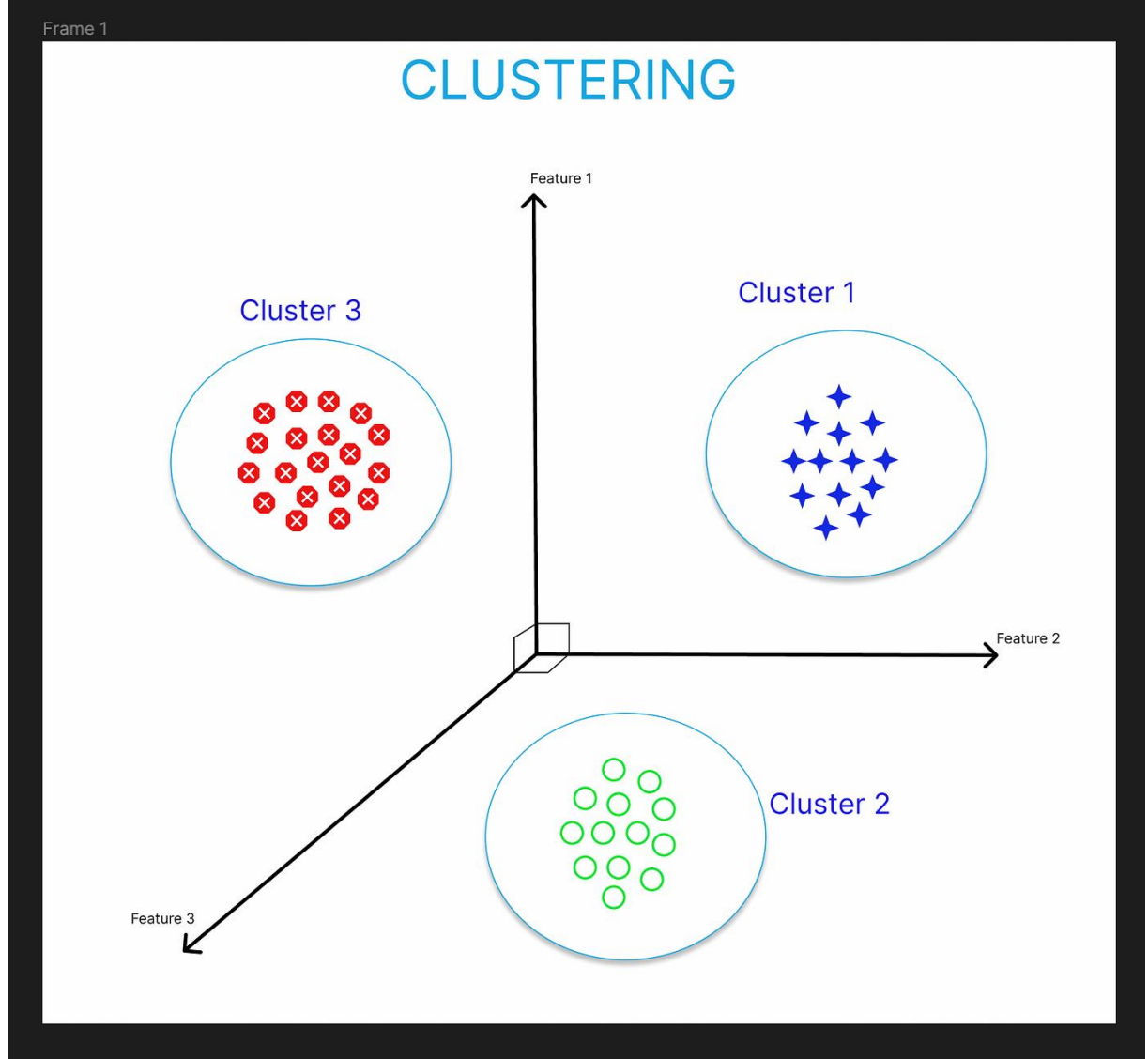
- Hesaplamalar için çeşitli kütüphaneler kullanıldığından gerekli kütüphaneler aktarılır.
- Veri tabanı dosyaları okunur.
- Etiketli verilerin ağırlıklı ortalama,standart sapma gibi istatistiksel analiz öz belirlenir ve yorumlanır.
- Verilerin grafiği çizilir.
- Değerler tahmin edilirken dikkate almak istenilen özellikler ve katsayıları seçilir.
- Bir modelin doğruluğunu kontrol etmek için,veriler eğitim ve test veri setlerine ayrılır.
- Model eğitilir.
- Test veri seti için bir tahmin fonksiyonu bulunur.
- Test verilerinin doğruluğu kontrol edilir:Gerçek değerleri veri setindeki tahmin edilen değerlerle karşılaştırılarak bir modelin doğruluğu kontrol edilebilir.

Makine Öğrenmenin Beş Popüler Algoritması

- Karar ağaçları
- Sinir ağları
- Olasılık ağları
- En yakın komşu
- Vektör makineleri desteklemek

Kümeleme

Kümeleme, denetimsiz öğrenmenin bir yöntemidir ve birçok alanda kullanılan istatistiksel veri analizi için yaygın bir tekniktir. Denetimsiz öğrenme, veri kümesi ile çıktıların olmadığı bir öğrenme metodudur. Veri kümesindeki verileri yorumlayarak ortak noktaları bulmak ve bunları kümeleştirme işlemi yapılarak anlamlı bir veri elde edebilmektedir. Sistem, öğretmenden olmadan öğrenmeye çalışır. Ham verileri organize verilere dönüştüren bir makine öğrenimi türüdür.



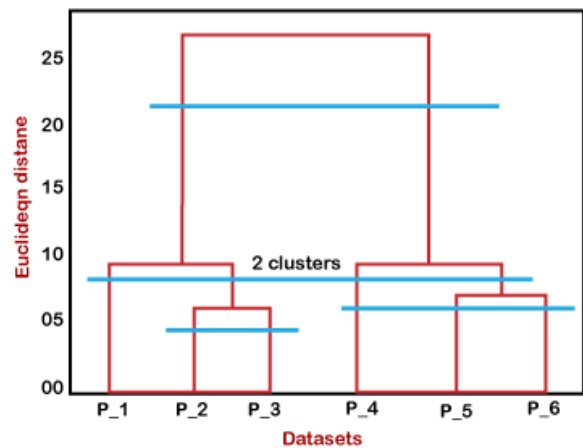
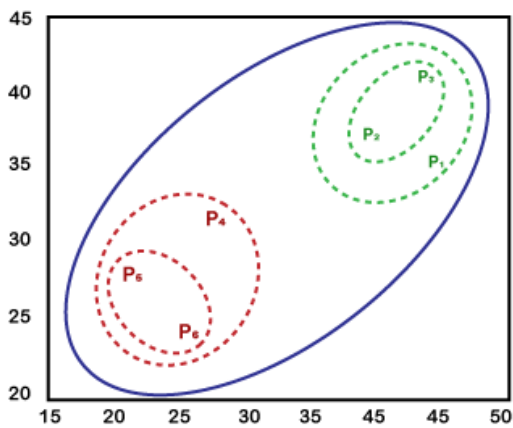
Kümelemenin Uygulama Alanları

Tıp'da elde edilen görüntülemeler üzerindeki farklılıkları analiz ederek değişik nitelikler çıkartabilir.

Suç yerlerinin belirlenmesi: Bir şehirdeki belirli bölgelerde mevcut olan suçlarla ilgili veriler, suç kategorisi, suç alanı ve ikisi arasındaki ilişki

Hiyerarşik Kümeleme

2'ye ayrılır.Yukarıdan aşağıya veya aşağıdan yukarıya.Aşağıdan yukarıya algoritmalar,her veri noktasına başlangıçta tek bir küme olarak ele alır ve ardından tüm kümeler tüm veri noktalarını içeren tek bir kümede birleştirilene kadar küme çiftlerini art arda birleştirir.



K-Means Algoritması:

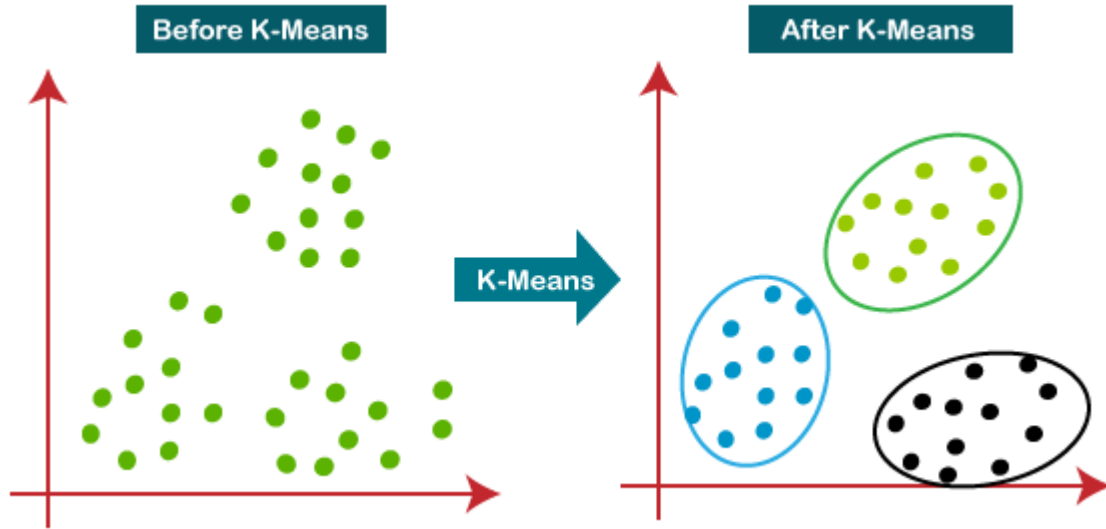
K adet kümeye bölünür.Algoritmanın özü birbirlerine benzerlik gösteren verilerin aynı küme içerisine alınmasına dayanır.Algoritmadaki benzerlik terimi,veriler arasındaki uzaklığa göre belirlenmektedir.Uzaklığın az olması benzerliğin yüksek,çok olması ise düşük olduğu anlamına gelmektedir.

K-means algoritmasının yapısı aşağıdaki gibidir;

- 1-)K adet rastgele küme oluştur.
- 2-)Kare hata oranı hesapla
- 3-)Verilerin kümelerin orta noktalarına olan uzaklıklarını bul
- 4-)Her veri için en yakın kümeyi,o verinin kümesi olarak belirle
- 5-)Yeni yerleşim düzenine göre hata oranı hesapla

6-)Eğer önceki hata oranı ile şimdiki hata oranı eşit değilse
2,3,4,5 ve 6. Adımları tekrarla

7-)Eğer önceki hata oranı ile şimdiki hata oranı eşitse
kümeleme işlemini sonlandır.



Gauss Karışım Modelleri (GMM) Kullanarak Beklenti-

Maksimizasyon(EM) Kümeleme

Gauss Karışım Modelleri (GMM'ler) bize K-Ortalamalarından daha fazla esneklik sağlar. GMM'ler ile veri noktalarının Gauss olarak dağıtıldığını varsayıyoruz; bu, ortalamayı kullanarak döngüsel olduklarını söylemekten daha az kısıtlayıcı bir varsayımdır. Bu şekilde, kümelerin şeklini açıklamak için iki parametremiz var: ortalama ve standart sapma! İki boyutlu bir örnek alırsak, bu, kümelerin her türlü eliptik şekli alabileceği anlamına gelir (çünkü hem x hem de y yönlerinde standart

bir sapmaya sahibiz). Böylece, her Gauss dağılımı tek bir kümeye atanır.

Sınıflandırma

Ayrık ve sürekli özellik değerlerinin bir vektörünü giren ve tek bir ayık değer olan sınıfı çıkaran bir sistemdir. Bir dizi öğenin sınıfını veya kategorisini tahmin etmek için sınıflandırma algoritmaları kullanılır.

1-)K-En Yakın Komşu Algoritması (K-Nearest Neighbour Algorithm)

2-) Lojistik Regresyon (Logistic Regression)

3-) Destek Vektör Makineleri (Support Vector Machines)

4-) Karar Ağaçları (Decision Tree)

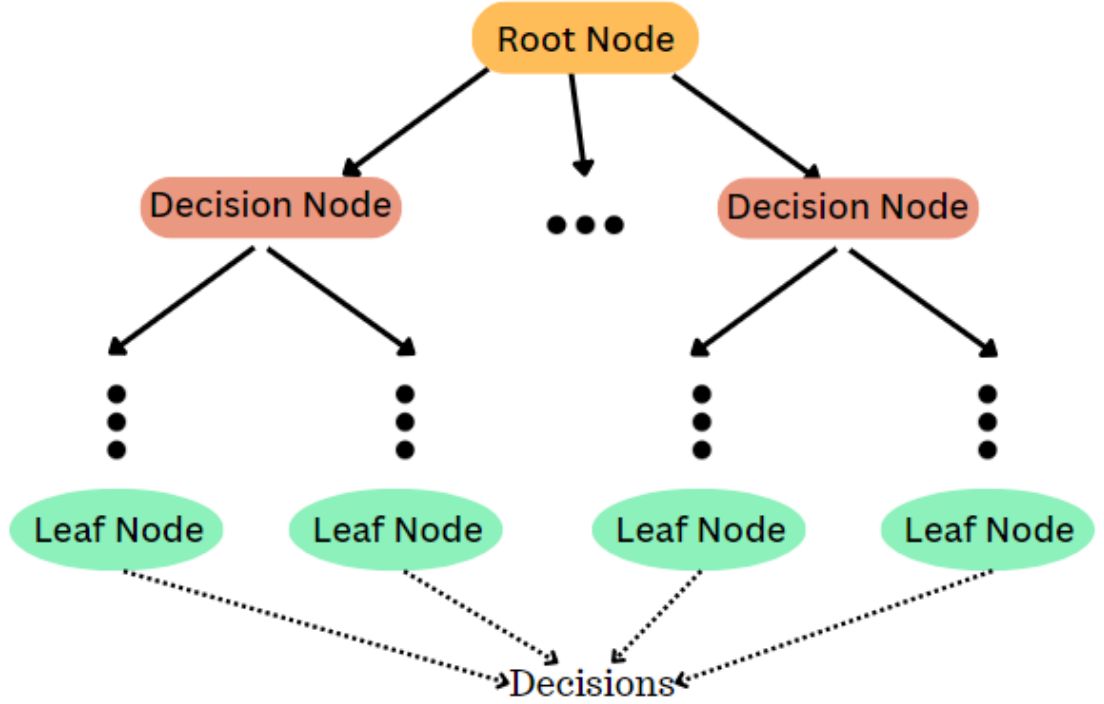
5-) Rasgele Orman Kümeleri (Random Forests)

6-) Yapay Sinir Ağları (Artificial Neural Networks) • Naive Bayes

Karar Ağaçları

Karar ağacı algoritması, denetimli öğrenme kategorisine girer. Hem regresyon hem de sınıflandırma problemlerini çözmek için kullanılırlar. Karar ağacı, her yaprak düğümün bir sınıf etiketine karşılık geldiği ve özniteliklerin ağacın iç düğümünde temsil edildiği sorunu çözmek için ağaç

temsilini kullanır. Karar ağacını kullanarak herhangi bir boole fonksiyonunu ayrık öznitelikler üzerinde temsil edebiliriz.



Karar ağacı kullanılırken yapılan bazı varsayımlar aşağıdadır:

1-) Başlangıçta, tüm eğitim seti kök olarak kabul edilir.

2-)Özellik değerlerinin kategorik olması tercih edilir. Değerler sürekli ise, model oluşturmada önce ayrıklaştırılırlar.

3-)Öznitelik değerleri temelinde, kayıtlar özyinelemeli olarak dağıtılır.

4-)Öznitelikleri kök veya dahili düğüm olarak sıralamak için istatistiksel yöntemler kullanılır

Karar Ağacı Algoritması: Karar ağaçları eğitici öğrenme için çok yaygın bir yöntemdir. Algoritmanın adımları:

- 1) T öğrenme kümesini oluşturulur.
- 2) T kümesindeki örnekleri en iyi ayıran nitelikler belirlenir.
- 3) Seçilen nitelik ile ağacın düğümleri oluşturulur ve herbir düğümden alt düğümler veya ağacın yapraklarını oluşturulur. Alt düğümlere ait alt veri kümesinin örneklerini belirlenir
- 4) 3. adımda oluşturulan her alt veri kümesi için
 - 4.1-) Örneklerin hepsi aynı sınıfa aitse
 - 4.2-) Örnekleri bölecek nitelik kalmamışsa
 - 4.3-) Kalan niteliklerin değerini taşıyan örnek yoksa işlemi sonlandır. Diğer durumda alt veri kümesini ayırmak için 2. adımdan devam edilir.

Ezber (Overfitting: Aşırı Uyum):

Tüm makine öğrenmesi yöntemlerinde verinin ana hatlarının modellenmesi esas alındığı için öğrenme modelinde ezberden (overfitting) kaçınılmalıdır.

Tüm karar ağaçları önlem alınmazsa ezber yapar. Bu yüzden ağaç oluşturulurken veya oluşturulduktan sonra budama yapılmalıdır.

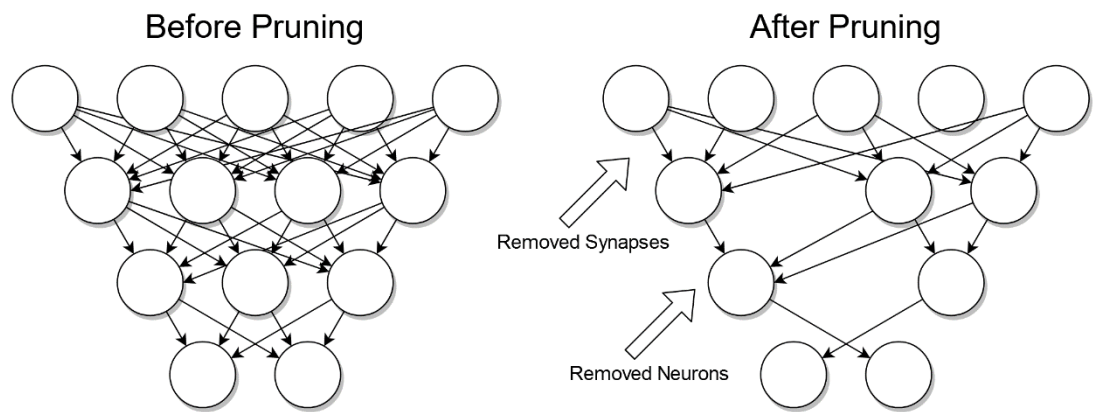
Ağaç Budama

Budama, sınıflandırmaya katkısı olmayan bölümlerin karar ağacından çıkarılması işlemidir. Bu sayede karar ağacı hem sade hem de anlaşılabilir hale gelir. İki çeşit budama yöntemi vardır;

- Ön budama
- Sonradan budama

Ön budama işlemi ağaç oluşturulurken yapılır. Bölünen nitelikler, değerleri belli bir esik değerinin (hata toleransının) üstünde değilse o noktada ağaç bölümleme işlemi durdurulur ve o an elde bulunan kümedeki baskın sınıf etiketi, yaprak olarak oluşturulur.

Sonradan Budama: Sonradan budama işlemi ağaç oluşturulduktan sonra devreye girer. Alt ağaçları silerek yaprak oluşturma, alt ağaçları yükseltme, dal kesme şeklinde yapılabilir.



Karar Ağacında en büyük zorluk, her seviyede kök düğüm için özniteliğin tanımlanmasıdır. Bu işlem öznitelik

seçimi olarak bilinir. İki popüler öznitelik seçim ölçüsü bulunmaktadır:

- 1) Bilgi Kazancı
- 2) Gini İndeksi

Bilgi Kazancı

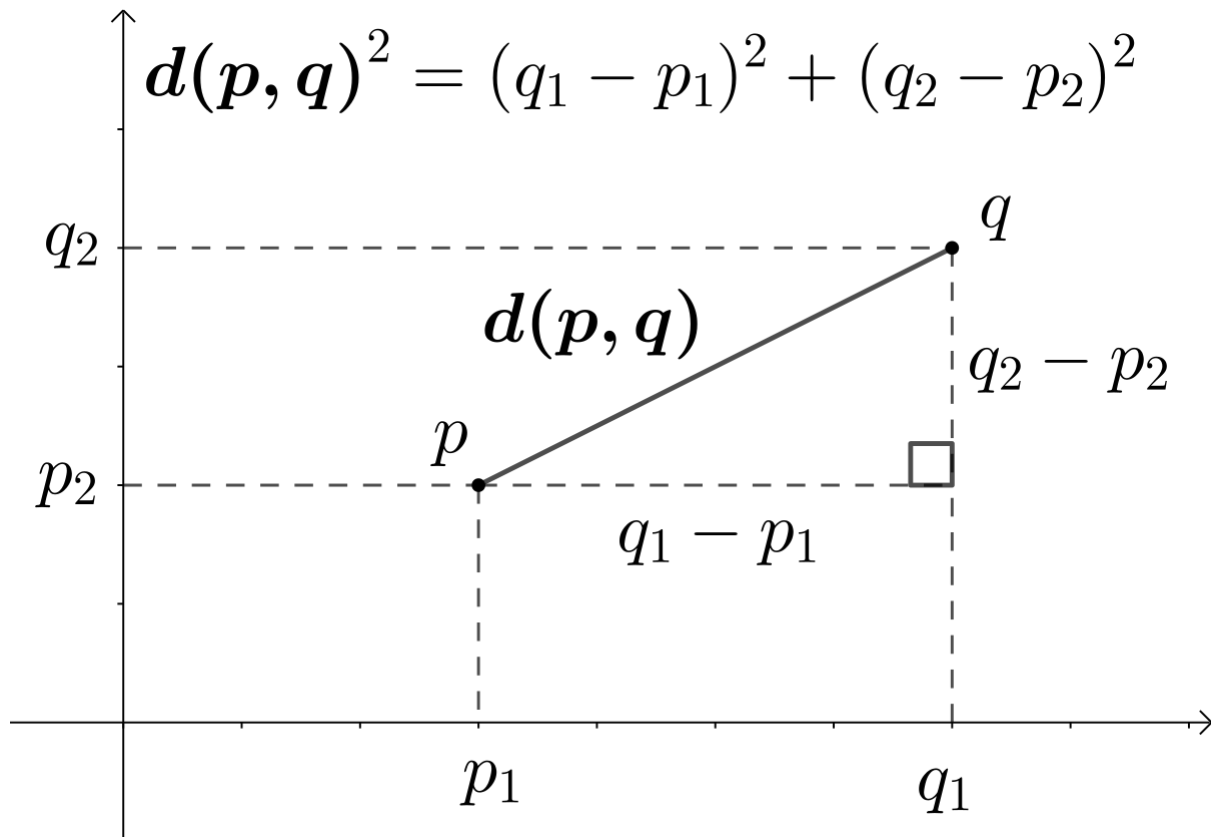
Eğitim örneklerini daha küçük alt kümelere bölmek için karar ağacında bir düğüm kullandığımızda entropi değişir. Bilgi kazancı, entropideki bu değişimin bir ölçüsüdür. Tanım: Diyelim ki S bir örnekler kümesi, A bir nitelik, S_v , S 'nin $A = v$ ile alt kümesi ve Değerler (A), A 'nın tüm olası değerlerinin kümesidir.

K-En Yakın Komşu

KNN algoritması sınıflandırılmak istenen bir veriyi daha önceki verilerle olan yakınlık ilişkisine göre sınıflandıran bir algoritmadır. Algoritma adının içinde bulunduğu “K” algoritmaya dahil edilecek veri kümesindeki veri sayısını ifade etmektedir. Yani algoritmada “k” adet komşu aranır. Bir tahmin yapmak istediğimizde, tüm veri setinde en yakın komşuları arar. 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. İlgili uzaklıklardan en yakın k komşu ele alınır. Öznitelik değerlerine göre k komşu veya komşuların sınıfına atanır. Seçilen sınıf, tahmin edilmesi beklenen gözlem değerinin sınıfı olarak kabul edilir. Yani yeni veri etiketlenmiş (label) olur.

K-NN non-parametric (parametrik olmayan), lazy (tembel) bir öğrenme algoritmasıdır. lazy kavramını anlamaya çalışırsak

“eager learning” aksine “lazy learning”’in bir eğitim aşaması yoktur. Eğitim verilerini öğrenmez, bunun yerine eğitim veri kümesini “ezberler”. 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.

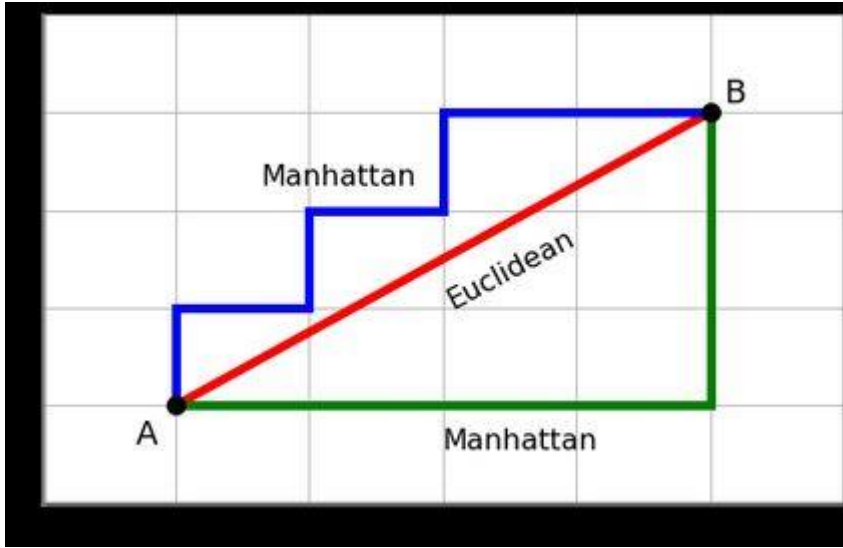


KNN algoritmaları, sınıflandırılacak veri noktasına en yakın komşu olan bir k sayısına karar verir. K değeri 5 ise, o veri noktasına en yakın 5 Komşuyu arayacaktır. Bu örnekte, $k = 4$. KNN en yakın 4 komşuyu bulur. Bu veri noktasının bu komşulara yakın olması nedeniyle sadece bu sınıfa ait olacağı

görülmektedir. K-en yakın komşu sınıflandırıcı algoritmalarının basit versiyonu, en yakın komşu sınıfı olarak hedef etiketini tahmin etmektir. Sınıflandırılacak noktaya en yakın sınıf, Öklid mesafesi kullanılarak hesaplanır.

Manhattan Distance, $|X1-X2| + |Y1-Y2|$

Euclidean Distance, $\sqrt{(x1-x2)^2 + (y1-y2)^2}$



K'nin önemi nedir? K değeri büyüdükçe tahmine duyulan güveni artırır. Öte yandan K çok büyük bir değere sahipse, kararlar çarpık olabilir. K nasıl seçilir? Algoritma, yeni bir veri noktasının diğer tüm eğitim veri noktalarına olan mesafesini hesaplar. Mesafe herhangi bir türde olabilir, örneğin Öklid, Manhattan, vb. Algoritma daha sonra k'ye en yakın veri noktalarını seçer, burada k herhangi bir tam sayı olabilir. Sayısal değerlerin hangi özelliği temsil ettiğine bakılmaksızın, seçimini diğer veri noktalarına yakınlığına göre yapar. Son olarak, veri noktasını benzer veri noktalarının bulunduğu sınıfa atar. Seçilen veri kümesine uyan K değerini seçmek için, KNN algoritması farklı K

değerleri ile defalarca çalıştırılır. Sonra, algoritma, yeni değerler için hassas tahminler yapma yeteneğini korurken karşılaşılan hata sayısını azaltan K 'yi seçilir.

1-) K 'ye karar vermek, K -en yakın Komşular'ın en kritik kısmıdır.

2-) K değeri küçükse, gürültü sonuca daha fazla bağımlı olacaktır. Bu gibi durumlarda modelin aşırı uyumu çok fazladır.

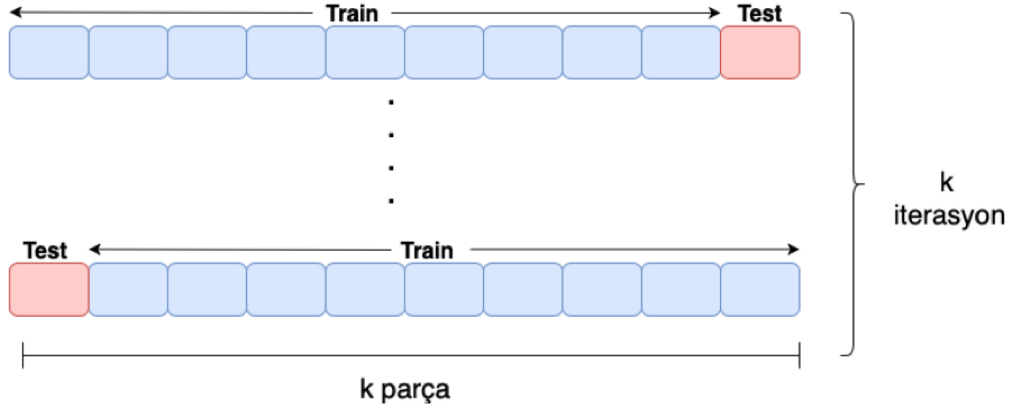
3-) K 'nin değeri ne kadar büyükse, KNN'nin arkasındaki prensibi yok edecektir.

4-) Çapraz doğrulamayı kullanarak K 'nin optimum değeri bulunabilir.

K Katmanlı Çapraz Doğrulama yöntemi; veri kümenizi “ k ” eşit parçaya bölerek, her bir parça için tek tek doğrulama verisi oluşturup işlemde bulunmaktadır. Böylece her veri noktası en az bir kez doğrulama verisi olarak kullanılmış olup, modelin genel performansı daha doğru değerlendirilebilir.

Buradaki bölünme değeri olan “ k ” değeri; genellikle $k=5$ veya $k=10$ olarak seçilmektedir.

Örneğin; veri kümenizi 10 parçaya bölerseniz, her bir parça tek tek doğrulama verisi olarak kullanılır ve geri kalan 9 parça ise eğitim verisi olarak kullanılır. Bu işlem 10 kez tekrar edilir ve her seferinde farklı bir parça doğrulama verisi olarak kullanılır. Performans değerlendirmesi için ise; her seferinde elde edilen doğruluk oranlarının ortalaması alınır.



KNN, eğitim seti yardımıyla veri noktasını belirli bir kategoriye sınıflandırmaya çalıştığımız parametrik olmayan denetimli bir öğrenme tekniğidir. Basit bir deyişle, tüm eğitim durumlarının bilgilerini yakalar ve yeni durumları benzerliğe göre sınıflandırır. Yeni bir örnek (x) için, en benzer K durum (komşular) için tüm eğitim seti aranarak ve bu K durumlar için çıktı değişkeni özetlenerek tahminler yapılır. Sınıflandırmada bu, mod (veya en yaygın) sınıf değeridir.

Adım 1: Uzaklık fonksiyonuna göre Benzerliği hesaplayın Pek çok uzaklık fonksiyonu vardır ama en yaygın olarak kullanılan ölçü Ökliddir. Esas olarak veriler sürekli olduğunda kullanılır. Manhattan mesafesi de sürekli değişkenler için çok yaygındır.

Mesafe ölçümü kullanma fikri, yeni örnek ve eğitim durumları arasındaki mesafeyi (benzerliği) bulmak ve ardından boy ve ağırlık açısından yeni müşteriye en yakın kmüşterileri bulmaktır. 'Monica' adlı yeni müşteri 161cm boyunda ve 61kg

ağırlığındadır. İlk gözlem ile yeni gözlem (monica) arasındaki Öklid uzaklığı aşağıdaki gibidir: $=\text{SQRT}((161-158)^2+(61-58)^2)$
Benzer şekilde, yeni vaka ile tüm eğitim vakalarının mesafesini hesaplayacağız ve mesafe açısından sıralamayı hesaplayacağız. En küçük mesafe değeri 1 olarak sıralanır ve en yakın komşu olarak kabul edilir.

2. Adım : K-En Yakın Komşuları Bulunması K=5 olsun. Ardından algoritma, özellikler açısından Monica'ya en yakın, yani Monica'ya en çok benzeyen 5 müşteriyi arar ve bu 5 müşterinin hangi kategorilerde olduğunu görür. 4 tanesi 'Orta T shirt bedenleri' ve 1 tanesi ise 'Büyük T gömlek bedeni' vardı, o zaman Monica için en iyi tahmininiz 'Orta T gömlek. Aşağıdaki anlık görüntüde gösterilen hesaplama bakın

KNN'nin Artıları ve Eksileri

Artıları:

- Anlaması kolay
- Veriler hakkında varsayım yok
- Hem sınıflandırma hem de regresyona uygulanabilir
- Çok sınıflı problemlerde kolayca çalışır

Eksileri:

- Yoğun Bellek / Hesaplama açısından pahalı
- Veri ölçeğine duyarlı
- Nadir olay (çarpık) hedef değişkeni üzerinde iyi çalışmıyor
- Çok sayıda bağımsız değişken olduğunda mücadele

- Herhangi bir problem için, küçük bir k değeri, tahminlerde büyük bir varyansa yol açacaktır. Alternatif olarak, k değerini büyük bir değere ayarlamak, büyük bir model yanlılığına yol açabilir.

En iyi K değeri nasıl bulunur?

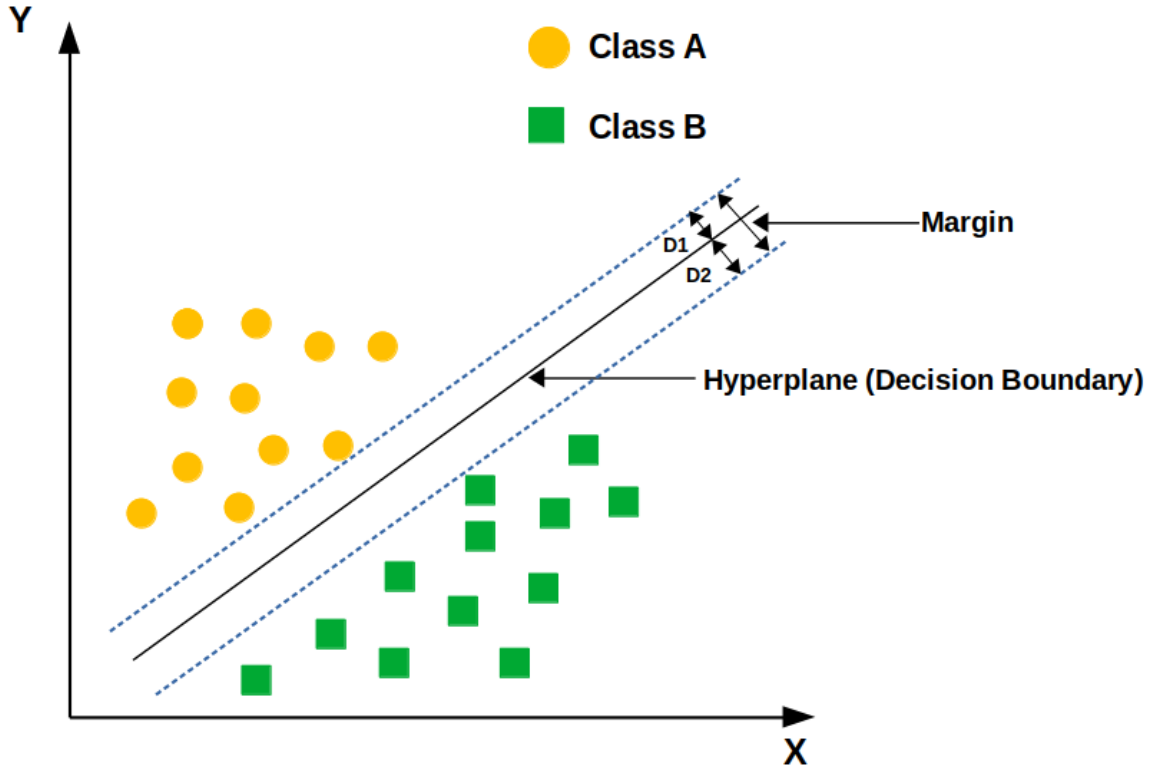
Çapraz doğrulama, optimal K değerini bulmanın akıllı bir yoludur. Model oluşturma sürecinden eğitim setinin bir alt kümesini dışarıda tutarak doğrulama hata oranını tahmin eder. Çapraz doğrulama (diyelim ki 10 kat doğrulama), eğitim setinin rastgele olarak yaklaşık eşit büyüklükte 10 gruba veya katlara bölünmesini içerir. Verilerin %90'ı modeli eğitmek için, kalan %10'u ise onu doğrulamak için kullanılır. Yanlış sınıflandırma oranı daha sonra %10 doğrulama verisi üzerinden hesaplanır. Bu prosedür 10 kez tekrarlanır. Farklı gözlem grupları, 10 defadan her biri bir doğrulama seti olarak ele alınır. Daha sonra ortalaması alınan doğrulama hatasının 10 tahminiyle sonuçlanır.

7.2.3. Destek Vektör Makineleri

Destek Vektör Makinesi, farklı sınıfları ayırmak ve sınır marjını en üst düzeye çıkarmak için karar sınırlarını bulmaktan sorumludur. Farklı sınıfların sınırları arasındaki boşluklar, çizgi ile çizgiye en yakın noktalar arasındaki (dik) mesafelerdir. DVM'de sınıflar arasındaki sınırlar çok önemlidir.

SVM, sınıflandırma için kullanılan denetimli algoritmalar sınıfında yer alır. 2 sınıf bir örnekle başlayalım: Verilen X_1 ve X_2

sınıfları için, 2 sınıfı en iyi, yani minimum hata ile ayıran karar sınırını bulmak istiyoruz. SVM bunu bir "Hiperplane" ile yapar. Şimdi bu hiperdüzlem 2 boyutlu veri olması durumunda tek bir doğru olabilir ve 3 boyutlu veri olması durumunda bir düzlem olabilir.



Destek Vektör Makineleri, hiper düzleme en yakın noktalar olan 'Destek Vektörleri' kavramını kullanır. Yukarıdaki örnekte, kırmızı çizgi, 2 sınıfı (Mavi yıldızlar ve Kırmızı daireler) ayıran karar sınırımızı gösterir ve tireli çizgiler, her iki sınıfın Destek Vektörleri arasında istediğimiz boşluğu, 'Marj'ımızı temsil eder. Sınırlar Önemlidir

Marj, Destek Vektörlerinin (dolayısıyla adı) yardımıyla tanımlanır. Örneğimizde, Sarı yıldızlar ve Sarı daireler, Marjı tanımlayan Destek Vektörleridir. Boşluk ne kadar iyi olursa,

sınıflandırıcı o kadar iyi çalışır. Bu nedenle destek vektörleri sınıflandırıcının geliştirilmesinde önemli bir rol oynamaktadır. Test verilerindeki her yeni veri noktası bu Marj'a göre sınıflandırılacaktır. Sağ tarafındaysa Kırmızı daire, aksi halde Mavi yıldız olarak sınıflandırılır.

Destek vektör ağları olarak da bilinen destek vektör makineleri, sınıflandırma ve regresyon için kullanılan ilgili denetimli öğrenme yöntemleri kümesidir. Her biri iki kategoriden birine ait olarak işaretlenmiş bir dizi eğitim örneği verildiğinde, bir DVM eğitim algoritması yeni bir örneğin bir kategoriye mi yoksa diğerine mi düştüğünü tahmin eden bir model oluşturur. Bir DVM eğitim algoritması olasılık dışı, ikili, doğrusal bir sınıflandırıcıdır, ancak Platt ölçeklendirme gibi yöntemler olasılıklı bir sınıflandırma ayarında DVM'yi kullanmak için mevcuttur. Doğrusal sınıflandırmayı gerçekleştirmenin yanı sıra, DVM'ler, girdilerini yüksek boyutlu özellik alanlarına dolaylı olarak eşleyerek çekirdek hilesi adı verilen kullanarak doğrusal olmayan bir sınıflandırma gerçekleştirebilir.

SVM'nin (Destek Vektör Makinesi) kullanabileceği iki sınıflandırma yöntemi:

- İkili sınıflandırıcıları birleştirme
- Çok sınıflı öğrenmeyi dahil etmek için ikili programı değiştirme

Destek Vektör Makinesi algoritması oluşturulurken,

- Sınıflar arasındaki birbirine en yakın ikili seçilir. Bu noktalara destek vektörleri isimleri veriyoruz.

- Destek vektörlerinden geçecek şekilde doğrular çizilir. Bu doğrulara sınır çizgisi adı verilir.
- Her doğruya eşit uzakta çizilen doğruya karar doğrusu adı verilir. Karar doğrusuna hiper düzlem de denir
- DVM'lerde sınıflar +1 ve -1 olarak etiketlenir. Karar doğrusunun üst kısmında kalan doğru denklemi, $wx+b=1$, altında kalan doğru denklemi ise $wx+b=-1$ olarak belirlenir. Karar doğrusu denklemi ise, $wx +b=0$ olur.

Destek vektör makinesi, makine öğrenmesi sınıflandırma problemlerinde kullanılan gözetimli öğrenme yöntemlerinden birisidir.

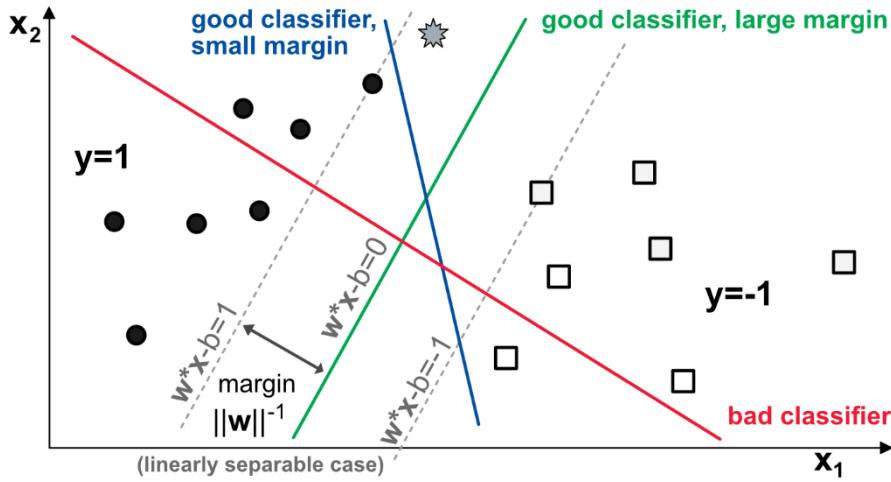
Çalışma Prensipleri Şu Şekildedir;

Bir veri seti içerisinde bulunan iki farklı sınıftan birine ait bir dizi veriler olduğunu düşünelim. SVM algoritması bu iki farklı sınıfa ait olan elemanları bir doğru çizerek birbirinden ayırır. Bu ayırma işlemini sınırdaki elemanlara göre yapar.

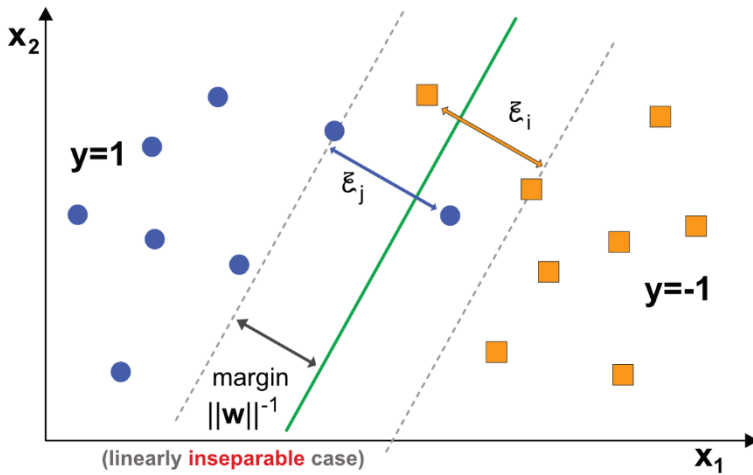
Bu çizilen doğruya hiper düzlem adı verilir. Bu algoritmadaki en önemli nokta hiper düzlemi en kusursuz şekilde belirleyebilmektir.

Doğru Hiper Düzlemi Bulma;

İlk adım doğru hiper düzlemi tanımlamaktır. Aşağıdaki görselde görüldüğü gibi elimizde bulunan üç hiper düzlem arasında sınıfları en iyi ayıran B düzlemidir.



İki sınıfı birbirinden doğru şekilde ayırabilen birden fazla düzlem olsun. Böyle bir durumda düzlemlerin iki sınıfa olan uzaklığına (marjın) bakılır. Marjini iki sınıf içinde en üst düzeyde olan (ortaya en yakın da denilebilir) düzlem en uygun hiper düzlem olarak seçilir.

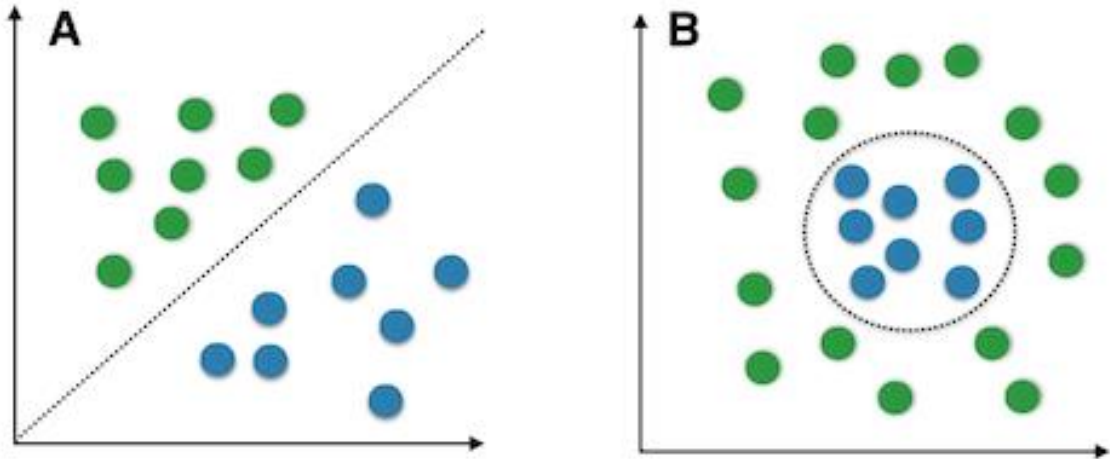


İki veri sınıfından birine ait olan eleman ait olduğu sınıftan uzak bir noktada ise o veri aykırı değer olarak ifade edilir. SVM

algoritması bu gibi aykırı değerleri önemsemeden maksimum marjine göre en uygun hiper düzlemi çizer.

İkili veri setleri her zaman doğrusa bir şekilde ayrılacak duruma sahip olmayabilir. Bu gibi durumda SVM'nin çekirdek numarası adı verilen tekniği kullanılır. Svm çekirdeği düşük boyutlu girdi alanı alan ve onu daha yüksek boyutlu bir alana dönüştüren, yani ayrılamaz problemi ayrılabilir probleme dönüştüren bir fonksiyondur. Kısaca, karmaşık veri dönüşümlerini gerçekleştirerek verilerin tanımlandığı etiketlere ve çıktılara göre ayırma işlemini gerçekleştirir.

Linear vs. nonlinear problems



Naive Bayes

Bayes Ağı, bir dizi değişken arasındaki ilişkilerin olasılıklarını öğrenerek çıkarım yapan bir makine öğrenmesi algoritmasıdır. Düşünce ya da oluşan kanı veya yönlendirilmiş olasılıklı bir modeldir. Örneğin, bir Bayes ağı, hastalıklar ve semptomlar arasındaki olasılık ilişkilerini temsil edebilir. Belirtiler verildiğinde, ağ çeşitli hastalıkların varlığının olasılıklarını hesaplamak için kullanılabilir.

Konuşma sinyalleri veya protein dizileri gibi değişken dizilerini modelleyen Bayes ağlarına dinamik Bayes şebekeleri denir. Belirsizlik altında karar problemlerini temsil edebilen ve çözebilen Bayes ağlarının genellemelerine etki diyagramları denir. Naïve Bayes'te sınıflandırıcı, lojistik regresyon gibi ayırt edici modellere göre daha hızlı birleşir, bu nedenle daha az eğitim verisine ihtiyacınız vardır. Ana avantajı, özellikler arasındaki etkileşimleri öğrenememesidir.

Bayesci mantık programı iki bileşenden oluşur. İlk bileşen mantıklı bir bileşendir; alanın niteliksel yapısını yakalayan bir dizi Bayes Cümlelerinden oluşur. İkinci bileşen nicelikseldir, alanla ilgili nicel bilgileri kodlar.

Yağmur, yağmurlama sisteminin etkinleştirilip etkinleştirilmeyeceğini etkiler ve hem yağmur hem de yağmurlama sistemi çimlerin ıslak olup olmadığını etkiler. Çim ıslak ise yağmu mu yağdı yoksa yağmurlama sistemi mi çalıştı.

Sıra Desen Madenciliği

Sıra desen madenciliđi, belirli bir sırayla gelen istatiksels olarak veri örnekleri arasındaki ilgili örüntüleri bulmaya çalışır.

Müşterilerin teknoloji market alışverişı verilerine göre son 3 ayda sırasıyla önce bilgisayar sonra CD-ROM son olarak dijital kamera satın almaları, tıbbi tedaviler, doğal felaketler(deprem), DNA dizilişı ve gen yapısı sıralı örüntü madenciliđi ile ilgilidir.

Mesela Internet şubesinde yapılan işlemler bir sıraya göre yapıldığı için sıralı örüntü madenciliđi içerisinde yer almaktadır. sıralı örüntü madenciliğinde arka arkaya yapılan işlemler göz önüne alınır.

Rassal Orman Modeli Algoritması

Rastgele ormanlar, denetimli bir öğrenme algoritmasıdır. Hem sınıflandırma hem de regresyon için kullanılabilir. Karar ağaçları: Rastgele orman modelinin yapı taşları oldukları için karar ağaçlarının bilinmesi gerekmektedir. Oldukça sezgisel yaklaşımlar içerir. Çoğu insanın hayatlarının bir noktasında bilerek ya da bilmeyerek bir karar ağacı kullandığına bahse girerim. Bir karar ağacının nasıl çalıştığını bir örnek üzerinden anlamak muhtemelen çok daha kolaydır.

