

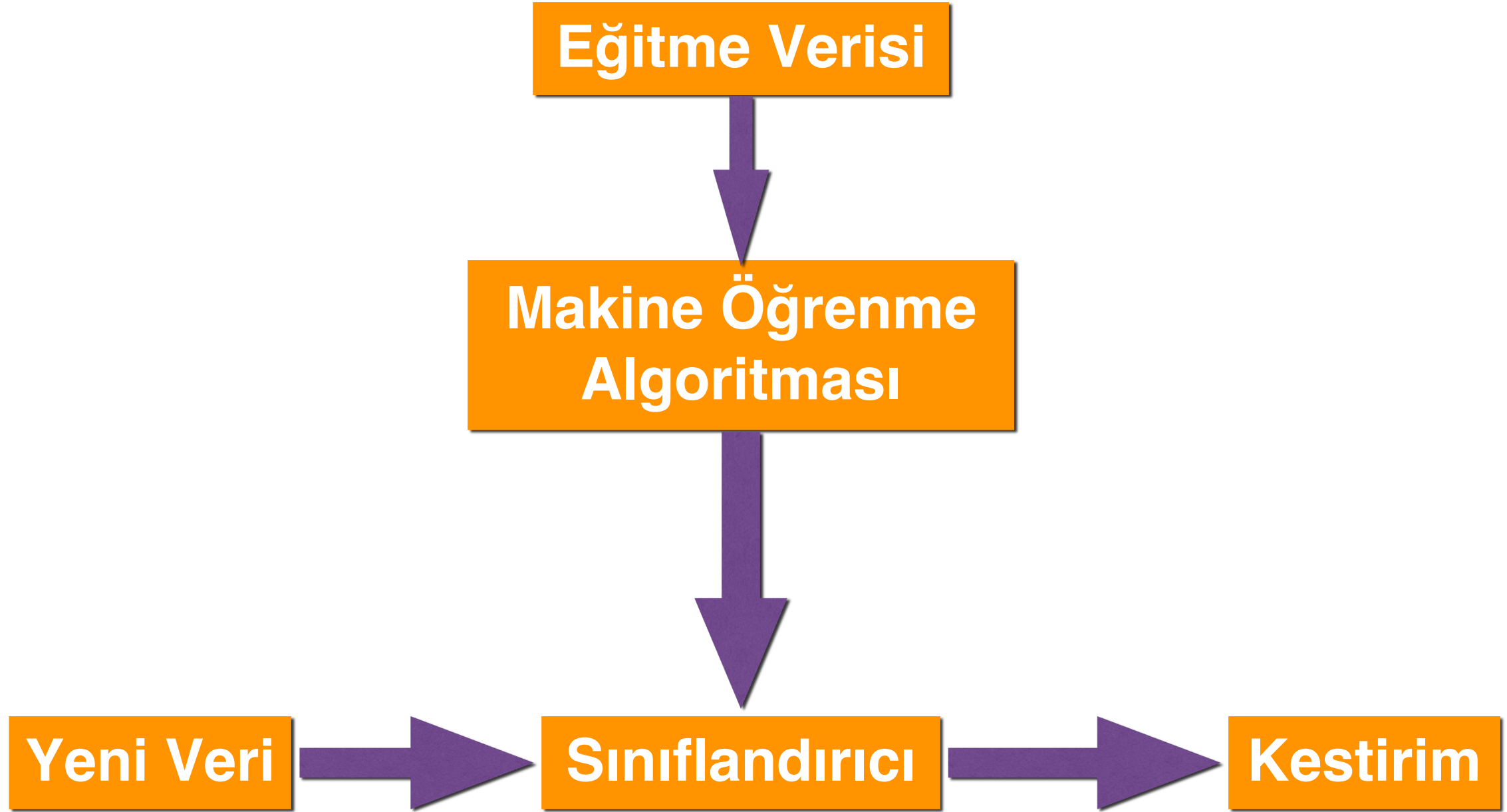


Mekatronik Mühendisliği Uygulamalarında Yapay Zekâ

Sınıflandırma Algoritmaları

Doç.Dr. Erhan AKDOĞAN





Sınıflandırma Kavramı:

- **Danışmanlı öğrenme olarak da adlandırılır.**
- **Etiketlenmiş veriler (eğitim verisi) üzerinden sistemin bir model oluşturması ve sonra etiketsiz (test kümesi) verilerinin sınıflarına göre etiketlenmesi işlemidir.**

Sınıflandırma Algoritmaları:

Advantages and Disadvantages of Classification Algorithm

Sr. No	Algorithm	Features	Limitations
1	C4.5 Algorithm	<ul style="list-style-type: none"> -Build Models can be easily interpreted. -Easy to implement -Can use both discrete and continuous values -Deals with noise. 	<ul style="list-style-type: none"> -Small variation in data can lead to different decision trees. -Does not work very well on a small training data set. -Overfitting.
2	ID3 Algorithm	<ul style="list-style-type: none"> -It produces the more accuracy result than the C4.5 algorithm. -Detection rate is increase and space consumption is reduced. 	<ul style="list-style-type: none"> -Requires large searching time. -Sometimes it may generate very long rules which are very hard to prune. -Requires large amount of memory to store tree.
3	K-Nearestneighbor Algorithm	<ul style="list-style-type: none"> -Classes need not be linearly separable. -Zero cost of the learning process. -Sometimes it is Robust with regard to noisy training data. -Well suited for multimodal classes. 	<ul style="list-style-type: none"> -Time to find the nearest Neighbours in a large training data set can be excessive. -It is Sensitive to noisy or irrelevant attributes. -Performance of algorithm depends on the number of dimensions used.
4	Naive Bayes Algorithm	<ul style="list-style-type: none"> -Simple to implement. -Great Computational efficiency and classification rate. -It predicts accurate results for most of the classification and prediction problems. 	<ul style="list-style-type: none"> -The precision of algorithm decreases if the amount of data is less. -For obtaining good results it requires a very large number of records.
5	Support vector-machine Algorithm	<ul style="list-style-type: none"> -High accuracy. -Work well even if data is not linearly separable in the base feature space. 	<ul style="list-style-type: none"> -Speed and size requirement both in training and testing is more. -High complexity and extensive memory requirements for classification in many cases.
6	Artificial Neural Network Algorithm	<ul style="list-style-type: none"> -It is easy to use, with few parameters to adjust. -A neural network learns and reprogramming is not needed. -Easy to implement. -Applicable to a wide range of problems in real life. 	<ul style="list-style-type: none"> -Requires high processing time if neural network is large. -Difficult to know how many neurons and layers are necessary. -Learning can be slow.

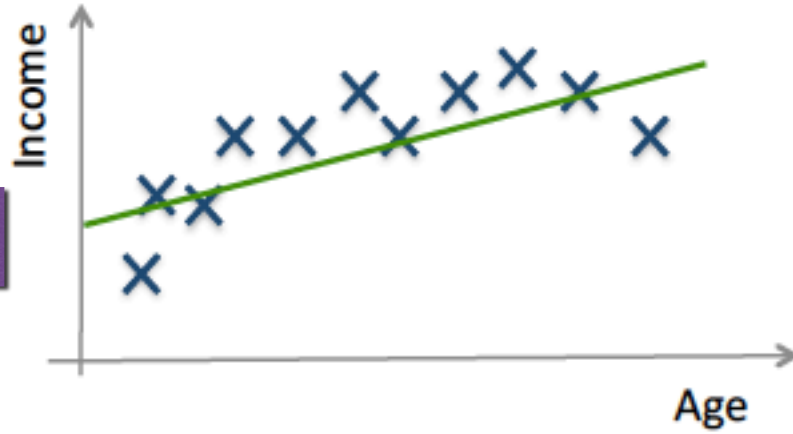
Sınıflandırma Algoritmaları:

Advantages and Disadvantages of Classification Algorithm

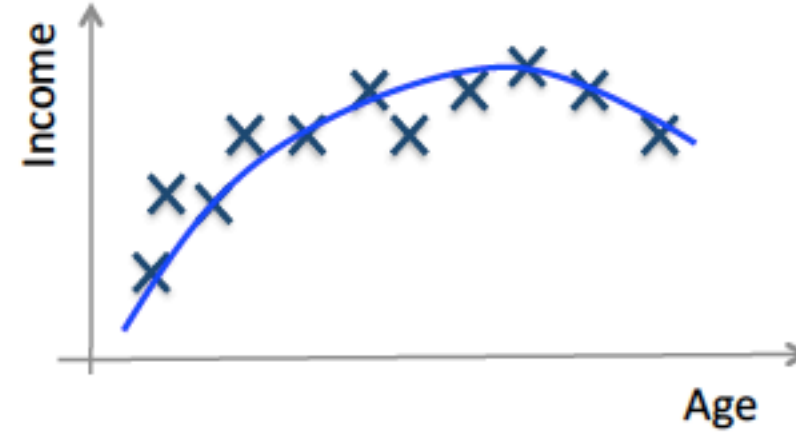
Sr. No	Algorithm	Features	Limitations
★	1 C4.5 Algorithm	<ul style="list-style-type: none"> -Build Models can be easily interpreted. -Easy to implement -Can use both discrete and continuous values -Deals with noise. 	<ul style="list-style-type: none"> -Small variation in data can lead to different decision trees. -Does not work very well on a small training data set. -Overfitting.
	2 ID3 Algorithm	<ul style="list-style-type: none"> -It produces the more accuracy result than the C4.5 algorithm. -Detection rate is increase and space consumption is reduced. 	<ul style="list-style-type: none"> -Requires large searching time. -Sometimes it may generate very long rules which are very hard to prune. -Requires large amount of memory to store tree.
★	3 K-Nearestneighbor Algorithm	<ul style="list-style-type: none"> -Classes need not be linearly separable. -Zero cost of the learning process. -Sometimes it is Robust with regard to noisy training data. -Well suited for multimodal classes. 	<ul style="list-style-type: none"> -Time to find the nearest Neighbours in a large training data set can be excessive. -It is Sensitive to noisy or irrelevant attributes. -Performance of algorithm depends on the number of dimensions used.
★	4 Naive Bayes Algorithm	<ul style="list-style-type: none"> -Simple to implement. -Great Computational efficiency and classification rate. -It predicts accurate results for most of the classification and prediction problems. 	<ul style="list-style-type: none"> -The precision of algorithm decreases if the amount of data is less. -For obtaining good results it requires a very large number of records.
★	5 Support vector-machine Algorithm	<ul style="list-style-type: none"> -High accuracy. -Work well even if data is not linearly separable in the base feature space. 	<ul style="list-style-type: none"> -Speed and size requirement both in training and testing is more. -High complexity and extensive memory requirements for classification in many cases.
✓	6 Artificial Neural Network Algorithm	<ul style="list-style-type: none"> -It is easy to use, with few parameters to adjust. -A neural network learns and reprogramming is not needed. -Easy to implement. -Applicable to a wide range of problems in real life. 	<ul style="list-style-type: none"> -Requires high processing time if neural network is large. -Difficult to know how many neurons and layers are necessary. -Learning can be slow.

Overfitting-Underfitting Kavramı:

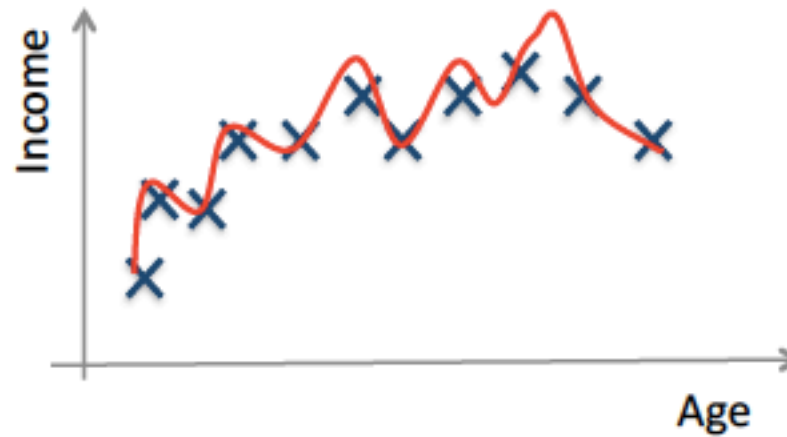
Yetersiz Uyum



High bias (underfitting)



Just right!



High variance (overfitting)

Aşırı Uyum

1. Destek Vektör Makineleri (Support Vector Machine-SVM)

- **Vapnik tarafından geliştirilmiştir.**
- **istatiksel öğrenme teorisine dayanır.**
- **Yüksek boyutlu bir uzay haritalandırılır.**
- **İlk olarak optik karakter okumada kullanıldı.**

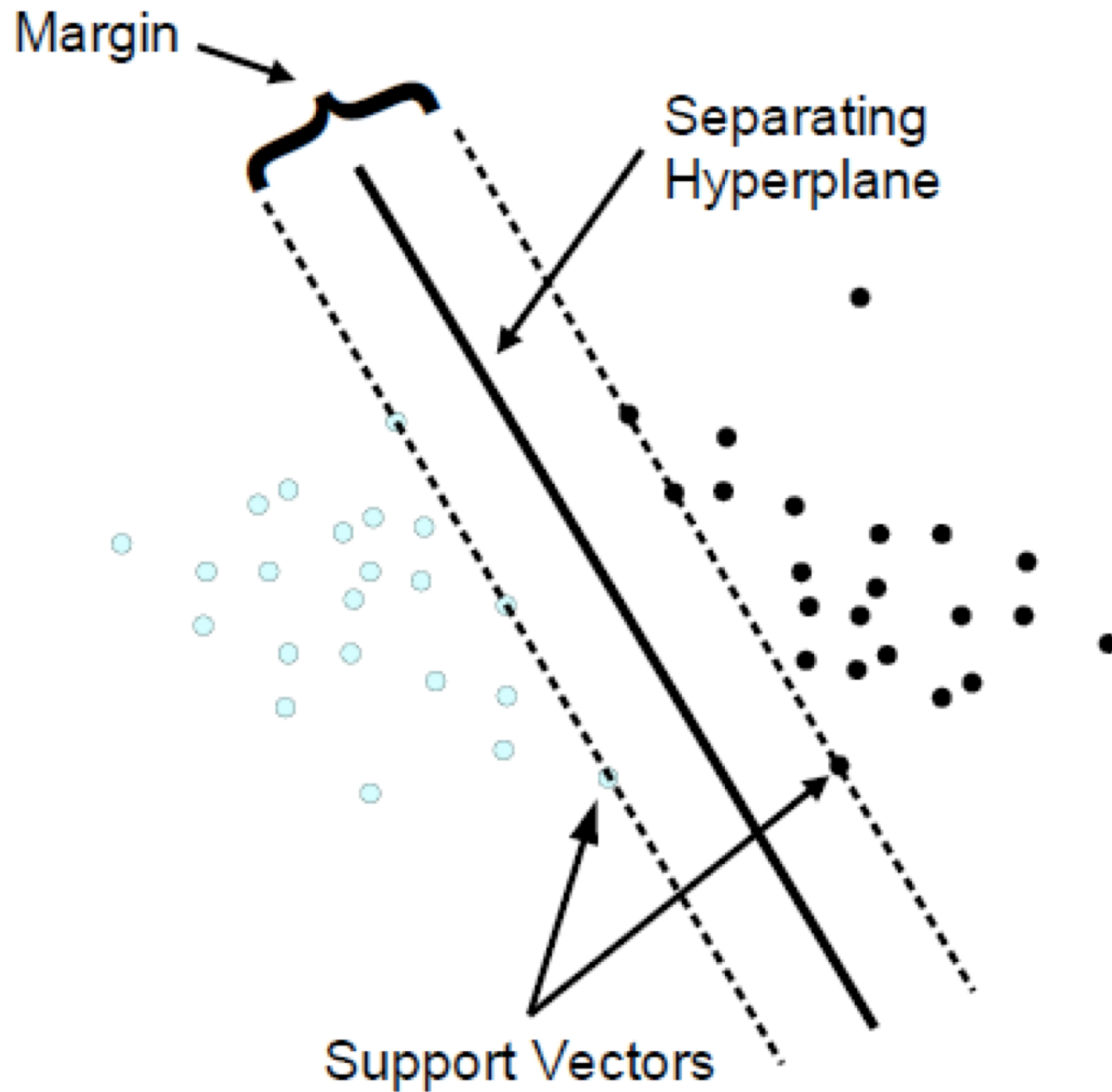
Avantajları:

- **Karmaşık karar sınırlarını modeller.**
- **Çok sayıda bağımsız değişken ile çalışabilir.**
- **Doğrusal ve doğrusal olmayan verilere uygulanabilir.**
- **Aşırı uyum sorunu azdır (göreceli olarak).**

Amaç:

- Sınıf sayısına göre sınıf üyeleri arasında bu sınıfları birbirinden ayıran sonsuz sayıdaki doğru içinden marjini en yüksek doğruyu seçerek sınıflandırmayı gerçekleştirmek.
- Oluşan marjin doğrusu sınıf üyelerinin seçilen doğruya en yakın olan üyelerine paralel olmalıdır. Çizilen doğrulara **hiperdüzlem** denir.

Hiperdüzlem (hyperplane)



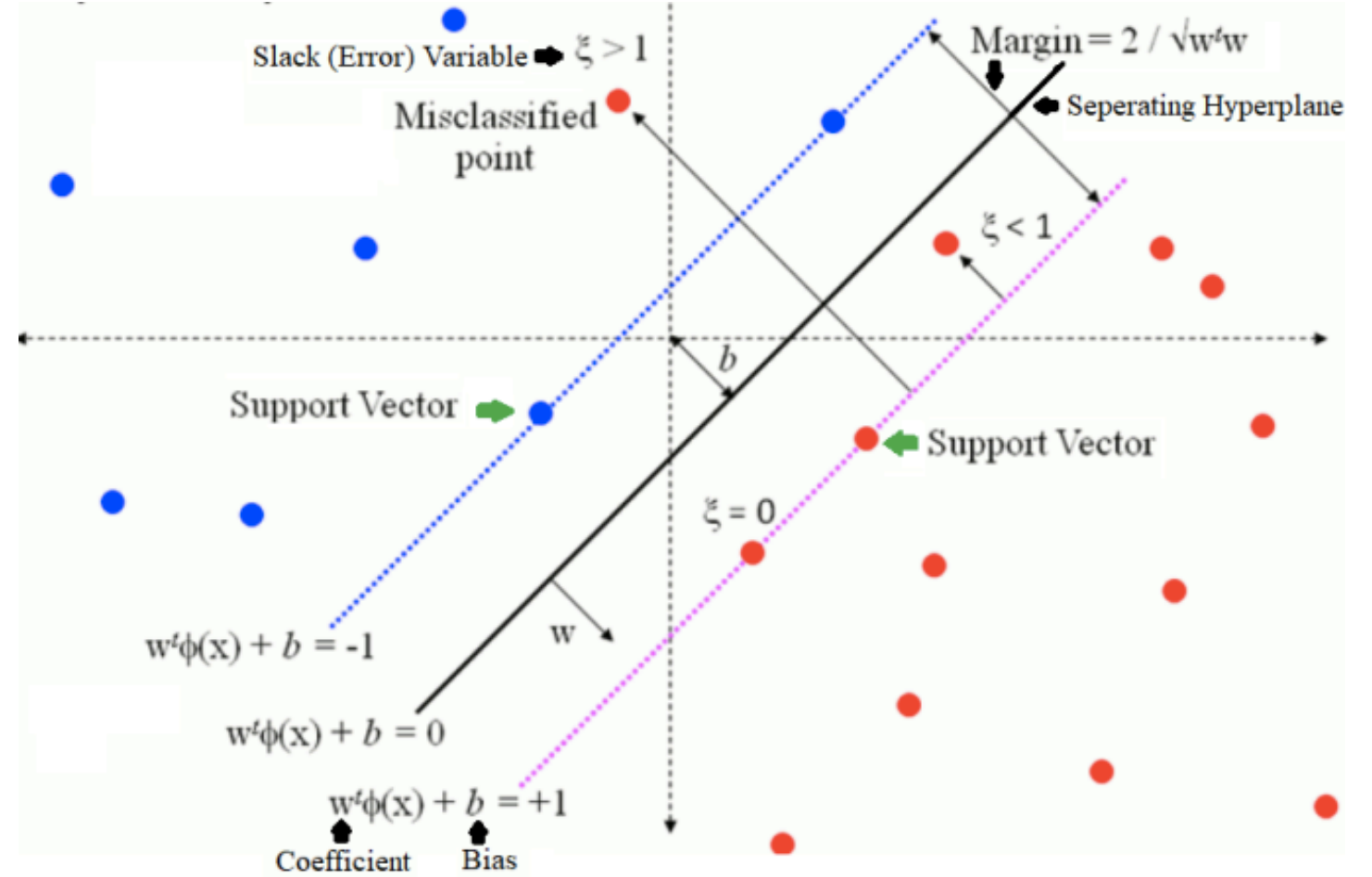
Marjin ve Destek Vektörleri

Sınıfları birbirinden ayırmak için hiperdüzlemler kullanılır.

Oluşturulan bu düzlemlerden marjini en büyük olan seçilerek işleme devam edilir.

En büyük marjine sahip hiperdüzlemi seçebilmek için her iki sınıfa yakın noktalardan geçen paralel doğrular çizilebilir. Paralel olarak çizilen ve destek vektörleri olarak.

Marjin genişliği $2/w$ ile verilir. O nedenle hedef w değerinin minimize etmektir.



- **Sınıflandırma “iki sınıf” temellidir.**
- **Daha çok sınıf var ise ikili sınıflamalar birleştirilir.**
- **Bunun için farklı yöntemler mevcuttur. Bire karşı bir(BKB) ve bire karşı diğer(BKD) bilinen yöntemler arasındadır.**

- **BKD de eğitim verisi ikili SVM ile ayrılabilir şekilde m adet sınıflandırıcıya bölünür. Oluşturulan k adet sınıflandırıcıların her biri, bir sınıfı kendisi dışındaki diğer sınıflardan ayırmak için kullanılır.**
- **Bu işlem elimizdeki tüm sınıflandırıcılar için tekrar edilir. Son aşamada elde edilen sonuçlar birleştirilir.**

- **m sınıflı bir sınıflandırma problemini çözmek için $M=m(m-1)/2$ adet farklı sınıflandırıcı oluşturulur.**
- **Bu sınıflar x. ve y. sınıfa ait veriler ile eğitilerek işleme başlanır.**
- **Eğitim örneklerini sınıflandırabilmek için yarışma yöntemi kullanılır.**
- **Eğitim bittiğinde n eğitim örneği x. sınıfa ait ise x sınıfının puanı 1 artırılır, değilse y. sınıfın puanı 1 artırılır.**
- **İşlem bittiğinde n örneği en yüksek puanı elde eden sınıfa eklenir.**

- **Doğrusal olarak ayrılamayan veri kümelerinde sınıflandırma yapılabilmesi için ilgili veriler daha yüksek boyutta bir uzaya taşınarak, sınıflandırma işlemi, oluşturulan bu yeni uzayda gerçekleştirilebilir.**

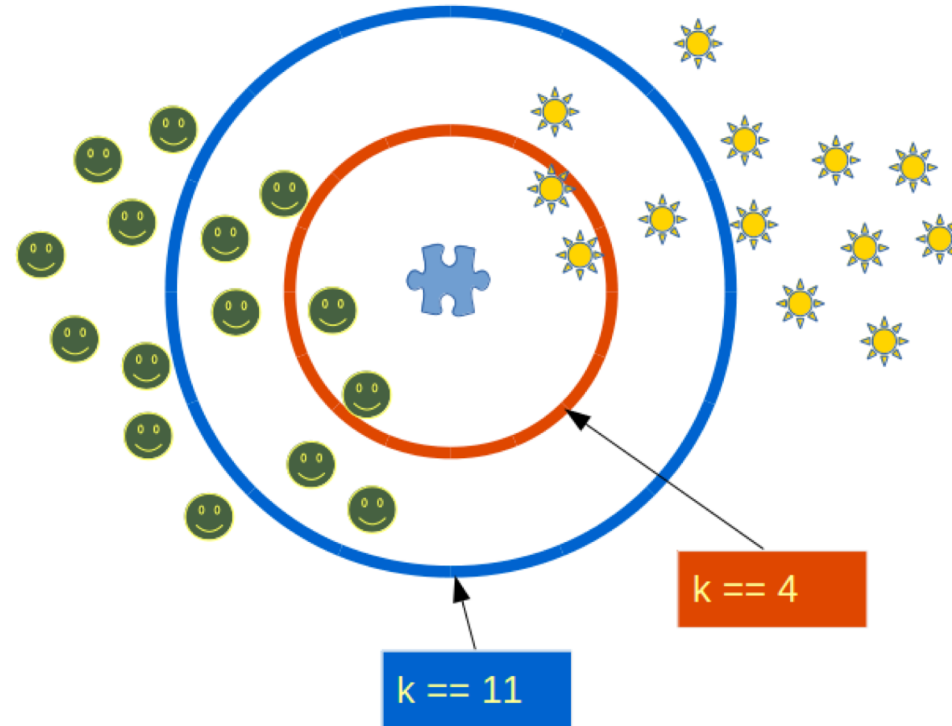
2. Yapay Sinir Ağları

- **Perceptron**
- **MLP**
- **RBF**

3. k-En Yakın Komşu (k-nearest neighbour)

- Seçilen bir özelliğin kendisine en yakın olan özellikle arasındaki yakınlığı kullanarak sınıflandırma gerçekleştirilir.
- k parametresi komşu sayısını temsil etmektedir.

🧩 == 😊 or 🧩 == ☀️ ?



- **Örüntü tanıma**
- **Veri Madenciliği**
- **Metin Sınıflandırma**
- **Görüntü İşleme**

Avantajları / Dezavantajları:

- **Uygulanması kolaydır.**
- **Gürültülü verilerde etkili çalışabilir.**
- **Eğitim setinin büyüklüğü fazla ise daha etkin sonuçlar üretir.**
- . . .
- **Algoritmanın başlangıcında k parametresine ihtiyaç duyar.**
- **En iyi sonucun elde edilebilmesi için hangi uzaklık ölçüsünün seçileceği belirgin değildir.**
- **Hesaplama maliyeti yüksektir.**

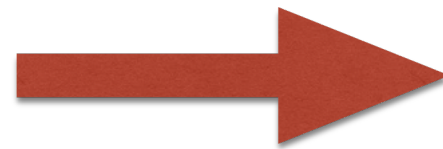
En yakın veri için uzaklık hesaplama yöntemleri:

- **Öklid**
- **Manhattan**
- **Minkowski**

3.1 Öklid Mesafesi:

- En yaygın kullanılan teknik.
- İki nokta arasındaki doğrusal mesafe olarak açıklanabilir.

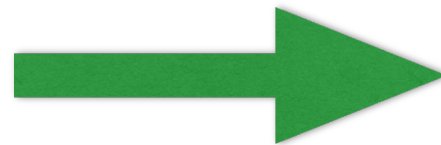
$\{x_1, x_2, \dots, x_n\}$



Herhangi iki nokta

$\{y_1, y_2, \dots, y_n\}$

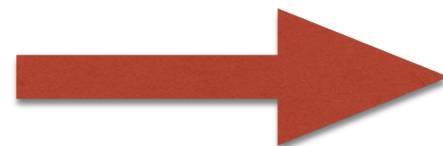
Öklid Mesafesi



$$d = \sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

3.2. Manhattan Mesafesi:

- **N boyutlu iki nokta arasındaki farkın mutlak değerlerinin toplamını ifade eder.**

$$\{x_1, x_2, \dots, x_n\}$$


Herhangi iki nokta

$$\{y_1, y_2, \dots, y_n\}$$

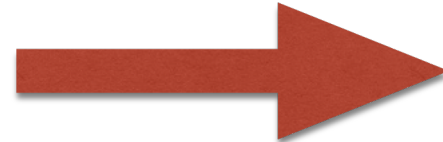
Manhattan Mesafesi



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

3.3. Minkowski Mesafesi:

- Öklid ve Manhattan Mesafelerinin genelleştirilmiş halidir.

$$\{x_1, x_2, \dots, x_n\}$$


Herhangi iki nokta

$$\{y_1, y_2, \dots, y_n\}$$

Minkowski Mesafesi



$$d = \left(\sqrt[p]{\sum_{i=1}^k (|x_i - y_i|)^p} \right)^{\frac{1}{p}}$$

p=1 Manhattan
p=2 Öklid

En yakın mesafeye karar verme:

- **k parametresine göre en yakın örnekler seçildikten sonra test örneğinin hangi sınıfa ait olduğuna karar vermek gerekir. Bunu iki şekilde yapmak mümkündür.**
 - 1. Basit Oylama:** Yapılan mesafe değerlendirmesi sonucunda seçilen komşu örnekleri içinde sayısı en fazla olan sınıfa atama yapılır.
 - 2. Ağırlıklı Oylama:** Yapılan mesafe değerlendirmesi sonucunda yakın komşuların uzak komşulara göre daha ağırlıklı etki göstermesi prensibine dayanmaktadır. Dolayısı ile sınıflandırmada yakın komşular uzak komşulardan daha etkindir.

- **K-en yakın komşu algoritması sınıfı bilinmeyen gözlem değeri için k gözlem içindeki en fazla tekrar eden sınıfın seçilmesi esasına dayanmaktadır.**
- **Ancak seçilen bu sınıf sadece k komşunun göz önüne alınması nedeniyle her zaman uygun olmayabilir.**
- **Bu son aşamada k komşu arasında en çok tekrarlanan sınıfı seçme yöntemi yerine ağırlıklı oylama (weighted voting) uygulanabilir.**
- **Söz konusu ağırlıklı oylama yöntemi gözlem değerleri için aşağıdaki bağıntıya göre ağırlıklı uzaklıkların hesaplanmasına dayanır.**

$$d(i, j)' = \frac{1}{d(i, j)^2}$$

$d(i, j)$: i. ve j. veriler arasındaki Öklid mesafesidir.

- Her bir sınıf değeri için bu uzaklıkların toplamı hesaplanarak ağırlıklı oylama değeri elde edilir.
- Hesaplama sonucunda ağırlığı en büyük olan sınıfa atama işlemi gerçekleştirilir. Bir başka deyişle en büyük ağırlıklı oylama değerine sahip olan sınıf değeri yeni gözlemin ait olduğu sınıf olarak kabul edilir.

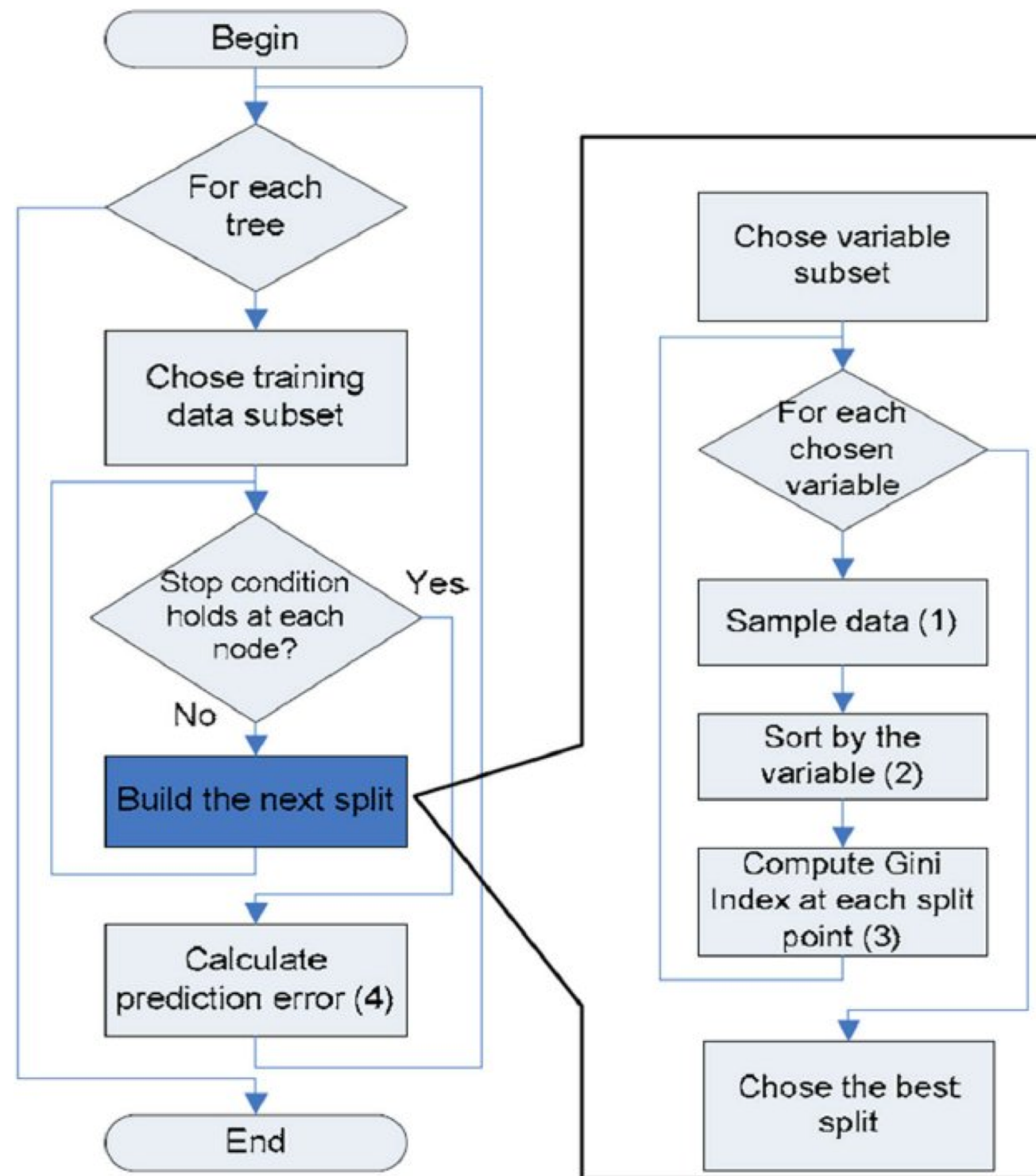
4. Rasgele Orman (Random Forest):

- Topluluk sınıflandırma yöntemlerinin çalışma ilkesi, *bir sınıflandırıcının yerine birden çok sınıflandırıcının ürettiği sonuçlar kullanılarak yeni gelen bir verinin sınıflandırılmaya çalışılmasıdır.*
- En yaygın topluluk sınıflandırma yöntemlerinden biri rasgele ormandır.
- Mevcut durumlardan rasgele oluşturulan örnekleri kullanarak oluşturulan çoklu ağaçlardan oluşan bir sınıflayıcı türüdür.
- Bir örneği sınıflandırmak için ormandaki her ağaca giriş vektörü verilmekte ve her ağaç için bir sonuç üretilmektedir. Rasgele Orman algoritması verilen sonuçlardan en çok oyu alan sınıf seçer.

Rasgele Orman Algoritması:

- Eğitim setindeki veriler yardımı ile ağaçları oluşturabilmek için kullanıcının belirlediği ağaç sayısına (N) ve her düğümde kullanılması düşünülen özellik sayısına m diyelim.
- **InBag (IB)** adı verilen eğitim veri kümesinin $2/3$ 'ü ön yükleme verisi olarak kullanılmak üzere ayrılır.
- Ardından **Out of Bag (OOB)** adı verilen, eğitim veri kümesinden kalan $1/3$ 'lük bölüm, sistemi test etmek için ayrılır.
- Yapılan bu işlemlerin ardından her bir ön yükleme verisi yardımı ile bir ağaç modeli oluşturulur.
- Her bir düğüm noktasında m özellikleri, tüm özellikler arasında rasgele olacak şekilde seçilir.
- Seçilen özellikler arasından en iyi dal bulunmaya çalışılır.
- Özellik sayınının seçimi düşük korelasyon elde etmek için hayati öneme sahiptir.

Rasgele Orman Algoritması:



Avantajları / Dezavantajları:

- Çok hızlı çalışır.
- Aşırı uyuma karşı dayanıklıdır.
- İstenildiği kadar ağaç ile çalışmaya imkan verir.
- Yüksek boyutlu veri kümelerinde kullanılabilir.
- Eksik veri problemine karşı tutarlı sonuçlar üretir.
- Hızlandırma yöntemi çok yavaş çalışır.
- Gürültülü verilerden etkilenir.

5. Naive Bayes (NB):

- Olasılık kuramı içinde incelenen bir konudur.
- Rastlantısal değişken için olasılık dağılım içerisinde şartlı olasılıklar arasındaki ilişkiyi göstermeye çalışır.
- Olasılık teorisi içinde incelenen bir olay olarak B olayına şartlı bir A olayı (yani B olayı bilindiği halde A olayı) için olasılık değeri , A olayına şartlı olarak B olayı (yani A olayı bilindiği halde B olayı) için olasılık değerinden farklıdır.
- Ancak bu iki birbirine ters şartlılık arasında belirli bir ilişki vardır ve buna **Bayes Teoremi** denir.

- Ulaşılmak istenilen hedef değişken ile elimizdeki bağımsız değişkenler arasındaki ilişkiyi tespit etmeye çalışan tahmin edici ve tanımlayıcı bir sınıflandırma algoritmasıdır.
- Bu algoritmada, modelin öğrenilmesi sayesinde her çıktının öğrenme kümesinde kaç kere meydana geldiğinin hesaplanmasını ve hesaplanan bu değerlerin öncelikli olasılık olarak adlandırılmasını sağlar.
- Bir diğer özelliği de her **bağımsız değişken/bağımlı değişken** kombinasyonunun oluşma sıklığını da bulmaya çalışmasıdır. Bu sıklıkları tahmin etmek için öncelikli olasılıklar birleştirilir.
- Büyük boyutlu verilerle çalışmada iyi sonuçlar üretemeyebilir.

- **X ve Y rasgele iki olay olsun.**

$$P(X | Y) = \frac{P(X | Y)P(X)}{P(Y)}$$

- **P(X): X olayının olasılığı (bağımsız)**
- **P(Y): Y olayının olasılığı (bağımsız)**

$P(X | Y)$, Y 'nin olduğu bilindiğinde X'in olma olasılığı

$P(Y | X)$, X 'in olduğu bilindiğinde Y'in olma olasılığı

- Yukarıdaki eşitlikten yararlanarak $P(X | Y)$ maksimum durumlar için hesaplanarak algoritma çalıştırılmış olur. $P(Y)$ bütün sınıflar için sabit olduğundan:

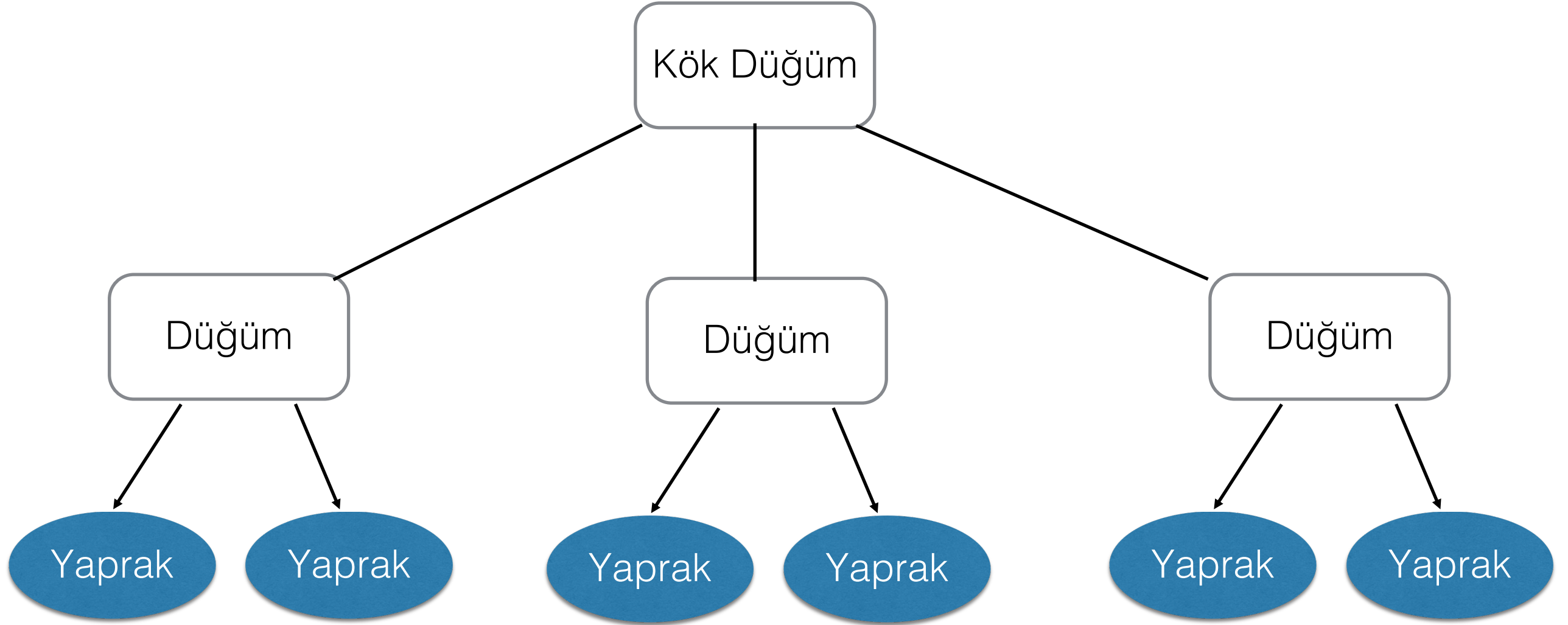
$$P(X | Y) = P(Y | X)P(X)$$

olasılığı için en büyük değer aranır.

6. Karar Ağaçları

- Çok sayıda veri içeren bir veri tabanını bazı teknikler kullanarak alt bölümlere ayırmak için geliştirilmiştir.
- Sınıf seçenekleri ile olasılıklara bağlı durumları belirli bir düzen içinde sıralar.
- Kategorik ve nümerik veriler üzerinde çalışabilir.
- İki aşamada çalışır: öğrenme, uygulama.

Karar Ağacı Yapısı:



- **Kayıt kök düğümünden başlar ve ara düğümlerden hangi yöne dallanacağı belirlenir.**
- **Her bir sınıf ağaçta tek yaprak olarak gösterilir.**
- **Bu yüzden bir sınıfa giden yol yalnızca bir tane olmalıdır.**
- **Yapraklar arasında herhangi kısa bir yol yoktur.**
- **Dallanma işlemi yaprak düğüme ulaşıncaya kadar devam eder.**

Avantajları / Dezavantajları:

- **Büyük boyutlu veriler ile çalışabilir.**
- **İlişkilerin yön ve önemini ayırt etmede kolaylık sağlar.**
- **Sürekli ve kategorik(ayrık) verilerle çalışabilir.**
- **Hesaplama maliyeti düşüktür.**
- **Modelin yorumlanması ve anlaşılması kolaydır.**
- **Kayıp veya eksik değerler algoritmanın çalışmasına etki etmez.**
- **Sürekli değişkenlerde çalışma performansı düşüktür. Çok sınıflı problemlerde başarısı düşüktür**
- **Eğitim verisi kısıtlı ise başarısı düşüktür.**
- **Ağaç oluşturma ve budama karmaşıklığı fazladır.**
- **En uygun ağacı bulmak bazen imkansız olabilir.**

- **CART - Classification and Regression Trees**
- **CHAID- Chi Squared Automatic Interaction detector**
- **ID3**
- **C4.5-C.5.0**

Kök düğümünde yer alacak özelliğın belirlenmesi:

- **DKM Kriteri**
- **Bilgi kazancı (IG)**
- **Gini Index (GI)**
- **Kazanç Oranı (GR)**

DKM Kriteri

$$M_y = d_{i,j}$$

Seçilen özelliğin $y=d_i$ durumunda iken M veri kümesinin alt kümeleri

i : seçilen özelliğin durum sayısı

j : sınıflarına ayrılacak özelliğin sınıf sayısı

p_1 : seçilen özellikte d_i durumu için birinci sınıfın gerçekleşme olasılığı

p_2 : ikinci sınıfın gerçekleşme olasılığı

$$DKM(M_{d_i}) = 2\sqrt{p_1 - p_2}$$

- **Veri kümesindeki yanlış sınıflandırma olasılığı. Gini indexi en küçük olan özellik ile işleme devam edilir.**

M_{d_i} : Seçilen özelliğin $y=d_i$ durumunda M veri kümesinin alt kümelerini gösterir.

i seçilen özelliğin durum sayısı

j sınıflarına ayrılacak özelliğin sınıf sayısı

p_j seçilen özellikte d_i durumu için j sınıfının gerçekleşme olasılığı

y özelliğinin d_i durumu için gini indeksi:

$$G(M_{d_i}) = 1 - \sum_j^t p_j^2$$

En küçük gini index değerine sahip özellik köke yerleştirilir.

Kazanç Oranı:

- **Bilgi kazancının normalleştirilmiş halidir.**
- **y seçilen özelliği ve M veri kümesini ifade etmek üzere**

$$GR(y, M) = \frac{IG(y, M)}{entropi(y, M)}$$

Budama:

- Oluşturulan ağaç aşırı büyük ise budama yapılır.
- Ağaç büyüklüğünün nedenleri:
 - Veri setindeki niteliklerin sayısı
 - Sınıf sayısı
 - Veri setinin gürültülü olması
 - Veri setinin ilgili olmaması
- Basit veri kümesinin işlenmesi sonucu ağaç yapısı büyük ise bu duruma **şişme** denir.
- Budama tipleri: ön budama ve geç budama
- Ağaç oluşturulur iken beraberinde budama yapılması **ön budama** işlemidir. Örneklerin az bir miktarının kullanılması ile yapılacak sınıflandırmalarda hata oranı yüksek olacağından eğitim kümesinin belli bir oranından az olan örneklem kümesi için budama işlemi sonlandırılır. O ana kadar ki en fazla elemana sahip olan sınıfın etiketi ilgili düğüme atanır.

- **Geç budamada** eğitim kümesinin her özelliği incelenerek sınıflar oluşturulur.
- Eğitim kümesinin bir kısmı budama işleminde kullanılmak üzere ayrılır.
- Ardından belli bir alt ağacın yerine ilgili alt ağaçta en fazla bulunan sınıfın etiketi yaprak şeklinde atanır.
- Oluşturulan yapraktan sonra sistem test için ayrılan veri ile sinanır.
- Sinama sonucunda hata oranı bir önceki durumdan daha fazla değilse budama gerçekleştirilir.
- *Ön budama* hızlı ve pratik iken *geç budama* doğruluğu daha yüksek bir ağaç oluşturabilmektedir.

6.1. CHAID (Chi-Squared Automatic Interaction Detector) Algoritması:

- Özelliklere bölebilmek için ki-kare testini kullanır.
- Dalların sayısı iki ile sınıf kategorisi arasında değişebilir.
- Kategorik ve nümerik değerler ile çalışabilir.
- Her düğümü ikiden fazla alt gruba ayırabilir.
- Çoklu ağaçlar üretir.
- Bağımlı değişken için tüm bağımsız değişkenler ile çapraz tablolar oluşturulur ve ki-kare değerleri hesaplanır.
- Ki-kare değeri en büyük olan kök düğüme yerleştirilir.
- Beklenen değer şu eşitlikten hesaplanır.

$$B_{ij} = \frac{(T_i)(T_j)}{n}$$

T_i: i. satır toplamı
T_j: j. sütun toplamı
n: genel toplam

- **Ardından ki-kare değerleri şu eşitlikten hesaplanır:**
- **G_{ij} : i. satır ve j.sütündeki gözlem**
- **B_{ij} : i. satır ve j.sütündeki beklenen değer**

$$\chi^2 = \sum_i^r \sum_j^c \frac{(G_{ij} - B_{ij})^2}{B_{ij}}$$

6.2. ID3 Algoritması:

- **Bilgi kazancında kullanılan entropi değeri üzerinden algoritma geliştirilir.**
- **Örnek:**

6.3. C4.5 ve C5.0 Algoritması:

C4.5:

- Veriyi rekürsif (özyinelemeli) olacak şekilde alt kümelere böler.
- Oluşturulan alt kümeleri kullanarak bir sınıflandırma ağacı oluşturur.
- Algoritma sürekli özellikler kullanır.
- Budamaya imkan verir.

C5.0 :

- %90 oranında daha az bellek kullanır. 5-240 arası kez daha hızlı çalışır.
- Sonuca varabilmek için daha küçük karar ağaçları oluşturabilir.
- Verinin ayrıştırılmasında daha başarılıdır.

- Hem sayısal hem de kategorik veriler ile çalışır.
- Dallanma kriteri olarak Gini İndexinden yararlanır ve ikili ağaçlar üretir.
- İki çocuk düğümü oluşturup bütün açıklayıcı özellikleri kullanarak veriyi alt kümelere bölmek üzere tasarlanmıştır.
- Bölme işlemi gerçekleştirilirken özellikler kategorik ise **Gini yada Twoing** yöntemi sürekli ise **en küçük kareler hata** yöntemi kullanılır.

- **Twoing İndexi aşağıdaki bağıntıdan hesaplanır.**

$$\phi(c | x) = 2P_{sol}P_{sag} \sum_{i=1}^m | P(i | x_{sol}) - P(i | x_{sag}) |$$

x: dallanmanın yapılacağı düğüm

c: aday

P_{sol}: ilgili verinin solda olma olasılığı

P_{sag}: ilgili verinin sağda olma olasılığı

$P(i | x_{sol})$ i. sınıf değerinin solda olma olasılığı

$P(i | x_{sag})$ i. sınıf değerinin sağda olma olasılığı



Referans:
Makine Öğrenmesi
Papatya Yayıncılık
Dr. Metin Bilgin

