Makine Öğrenmesinin Matematiksel Yöntemleri

Hazırlayan: HASAN ÇELİK

Öğrenci No : 090180305

Teslim Tarihi: 08.01.2022

Danışman : PROF. DR. ELİF ÖZKARA CANFES

Contents

List of Figures

Makine Öğrenmesinin Matematiksel Yöntemleri

Hasan Çelik

10 Ocak 2023

1 Tasarımın Tanımı ve Amacı

Bu çalışmanın amacı makine öğrenmesindeki sınıflandırma algoritmalarını ve tekniklerinin matematiksel olarak araştırıp algoritmaların hangi veriler üzerinde daha iyi çalıştığını anlamaktır.

2 Tasarımın Kapsamı ve Kullanım Alanları

Tasarım sırasında öncelikle makine öğrenmesinin temel prensipleri ve makine öğreniminin alt dalı olan sınıflandırma problemleri açıklanmıştır. Sınıflandırma problemlerinde kullanılan yaygın makine öğrenimi algoritmaların çalışma prensipleri matematiksel formüllerle tanımlanmıştır. Algoritmaların daha doğru ve performanslı çalışması için kullalınabilecek teknikler araştırılmış ve makine öğrenimi modellerin performansını ve kullanılabilirliğini test etmek için gereken istatistiksel yöntemlerden bahsedilmiştir.

3 Yapılan Çalışmalar

3.1 Makine Öğrenimi Nedir?

Makine öğrenimi insan öğrenme sürecini otomatize edip yeni bilgiler edinmek için bilgisayarları kullanan bir sistemdir ve bilgisayar performansını ve doğruluğunu geliştirmek için üzerinde çalışılan bir konudur. Bilgisayar bilgi edinmek için girdileri kullanır ve girdilerin yapısını farklı tekniklerle işleyerek çıktı üretir. Makine öğrenimi bu girdilere göre denetimli ve denetimsiz öğrenme olarak ikiye ayrılır.
Denetimli Öğrenme,girdilerin bir hedef değişkene sahip olduğu makine öğrenmesi türüdür. Algoritmalar hedef değişken dışındaki değişkenleri kullanarak hedef değişkeni tahmin etmeyi amaçlar. Sınıflandırma ve regresyon modelleri denetimli öğrenme problemleridir. Bu araştırmada sınıflandırma problemleri üzerinde çalışacağız.

3.2 Sınıflandırma Problemi ve Algoritmaları

Sınıflandırma, bir modelin girdi verilerine bir sınıf etiketi atamak üzere eğitildiği bir gözetimli makine öğrenimi tekniğidir. Öğrenim sürecinde algoritmalar girdi olarak veriler ile birlikte bu verilerin sahip olduğu sınıf etiketlerini alır. Girdi olarak alınan veri kümesi bağımlı değişken sınıf etiketlerini ise bağımsız değişken olarak düşündüğümüzde, algortimalar bağımsız değişkenlere göre bağımlı değişkeni tahmin etmek için her bağımsız değişkene ağırlık vererek sınıf etiketine karar verir. Algoritmaların yaklaşımlarına göre yapılan sınıflandırmalar farklılık gösterir. Çalışmanın devamında sınıflandırma problemleri için uygulanan farklı algoritmaların çalışma prensiplerini inceleyerek sınıflandırmanın nasıl yapıldığını araştıracağız. Birçok sınıflandırma algoritması olmasına rağmen yaygın olarak kullanılan algoritmalar üzerinde çalışacağız. Bu algoritmalar:

- Lojistik regresyon
- Destek vektör makinaları
- K en yakın komşu algoritması
- Karar ağaçları

3.3 Lojistik Regresyon

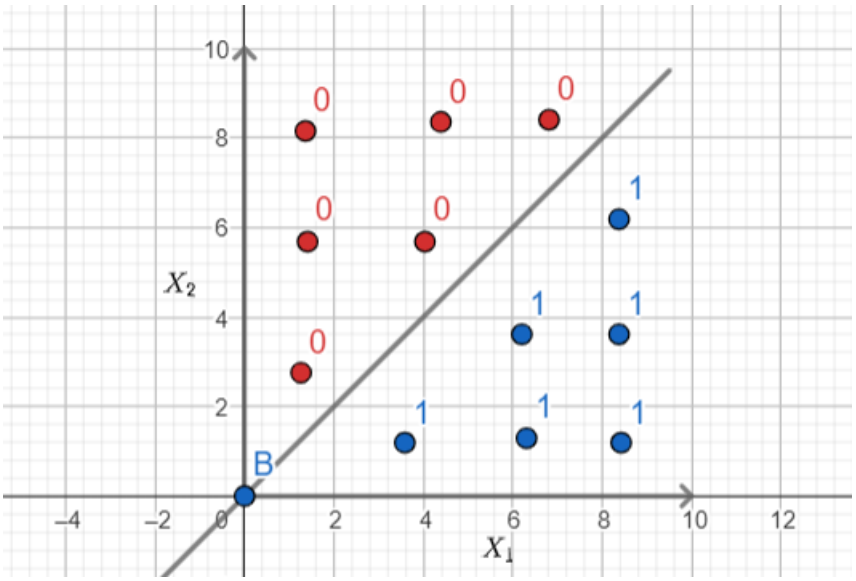
Lojistik regresyonu anlamak için öncelikle bir hiper düzmlem ile ayrılan verilerin nasıl sınıflandırıldığına bakalım. $\mathbf{X}_i$'lerin bağımlı değişken, $\mathbf{Y}_i$'lerin ise hedef değişken olduğu $(\mathbf{X}_i, \mathbf{Y}_i)$ veri noktaları için $\mathbf{X} \in R^n$,$\mathbf{Y} \in \{1, 0\}$ olsun. Bu veri noktalarını normali $\mathbf{n}$ olan ve yer değiştirmesi $\mathbf{b}$ olan $\mathbf{nX} + \mathbf{b}$ hiperdüzlemi ile kestiğimizde:

$$\begin{aligned} nX + b > 0, Y = 1 \\ nX + b < 0, Y = 0 \end{aligned} \tag{1}$$

Sınıflandırması yapılırsa,

$$n = \begin{bmatrix} 1 & -1 \end{bmatrix}$$

için sınıflandırma aşağıdaki gibi olur



Şekil 1: Hiper düzlem ile sınıflandırma

Lojistik regresyon böyle bir sınıflandırma yapmak yerine olasılık oranı yaklaşımını kullanır. Bir olayın gerçekleşme olasılığı p olduğunda gerçekleşmeme olasılığı $1 - p$ dir ve olasılık oranını $p/1 - p$ olur. Varsayılan olarak,

$$\begin{aligned} P > 0.5, y = 1 \\ P < 0.5, y = 0 \end{aligned} \tag{2}$$

sınıflandırmasını yapacağımızı düşünelim. 1 denkelmindeki hiperdüzlemi olasılık oranına eşitlediğimizde aşağıdaki denklemi elde ederiz.

$$nx + b = \frac{p}{1 - p} \tag{3}$$

(3) denkleminde eşitliğin sol tarafından elde edilen değer arttıkça p olasılığının arttığı , değer azaldıkça p olasılığının azaldığı görülür. Denklem (1)' deki sınıflandırma mantığına yaklaşmış olsak da denklem(3) de eşitliğin sağ tarafı 0 ile $+\infty$ arasında değerler alır. (3) Denkleminin sağ tarafına log dönüşümü yapıldığında

$$nx + b = \log\left(\frac{p}{1 - p}\right) \tag{4}$$

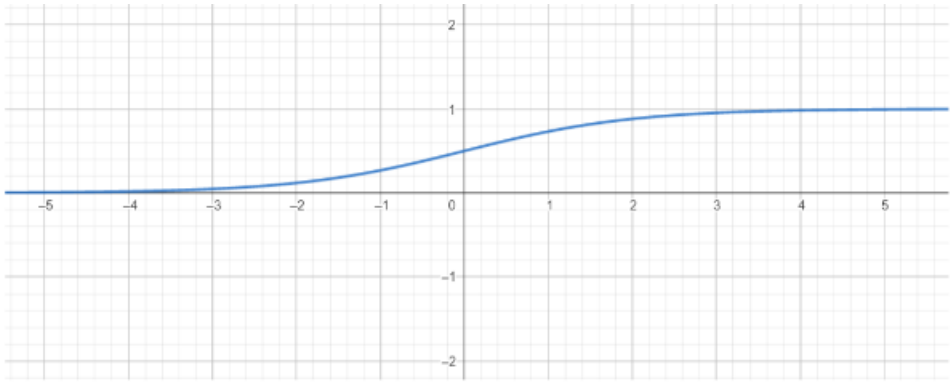
denklemin sağ taraf değerleri (-∞,+∞) aralığına genişletilmiş olur. Artık (4) ve (2) eşitiklerini kullanarak elde ettiğimiz p değerleri ile sınıflandırma yapabilir.(4) denkleminde sağ taraftaki p değeri yalnız bırakılırsa:

$$y = \log\left(\frac{p}{1-p}\right)$$
$$p = \frac{e^y}{1+e^y} = \frac{1}{1+e^{-y}} \tag{5}$$

denklemi elde ediliriz. Denklemin sağ tarfındaki fonksiyon sigmoid fonksiyondur ve $(-\infty\ ,+\infty)$ aralığındaki y değerlerini, (0,1) aralığındaki p değerlerine dönüştürmemizi sağlar. Sigmoid fonksiyonu veri noktalarını ayırdığımız hiperdüzleme uygulanırsa:

$$p = \frac{e^{nx+b}}{1+e^{nx+b}} = \frac{1}{1+e^{-nx-b}} \tag{6}$$

eşitliğini elde edilir ve grafiği aşağıdaki gibi olur:



Şekil 2: sigmoid Fonksiyon

Lojistik Regresyon, denklem (6)'yı kullanılarak bir veri noktası için $nx+b$ değerine karşılık gelen p değeri hesaplanır ve denklem (2) deki p olasılık değerine göre sınıflandırma yapar.Grafiğe baktığımızda $nx+b$ nin 0 değeri için p değeri 0.5 e karşılık gelir yani herahngi bir sınıflandırma yapamayız. Bu değer aynı zamanda (1) denkleminde veri noktasının hiperdüzelmin üzerinde olma durumudur. Diğer değerlere bakıldığında lojistik regresyonun 0 ın etrafındaki küçük bir değişim için veri noktasının o yöndeki sınıfta olma olasığını çok hızlı arttırdığını görülür.

Herhangi bir verinin doğru sınıfta olma olasılığı aşağıdaki gibi tanımlanır:

$$P(x,y) = \left(\frac{1}{1+e^{-nx-b}}\right)^y \left(\frac{1}{1+e^{nx+b}}\right)^{1-y}. \tag{7}$$

Öyleyse bütün veriler için toplam olasılık değeri

$$P(T) = \prod_{i=1}^{n} P(x_i, y_i) \tag{8}$$

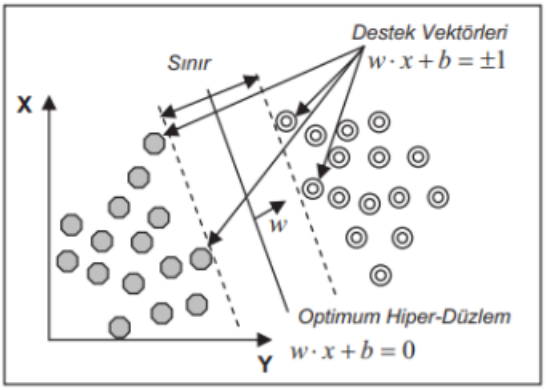
olur. n ve b parametrelerini belirlemek için (8) denklemi maksimize edilir. Denklem (7) deki ifadenin logaritmasını alarak işlemleri kolaylaştırırız ve minimize edilmek istenilen hata fonksiyonu

$$\mathrm{LE}(n,b) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i \log\left(1+e^{-nx_i-b}\right) + (1-y_i)\log\left(1+e^{nx_i+b}\right)\right) \tag{9}$$

şeklini alır. Lojistik regresyon maksimum olabilirlik yöntemi ile (n,b) parametrelerine karar verir.

3.4 Destek Vektör Makineleri(SVM)

Destek vektör makineleri sınıflandırma problemlerinde kullanılan denetimli makine öğrenmesi algoritmalarından biridir. Destek vektör makineleri sınıflandırma yaparken iki sınıfı birbirinden ayıracak bir hiperdüzlem bulur. İki sınıfı birbirinden ayıran birçok hiperdüzlem bulunabilir ama SVM algoritması en geniş sınır aralıklı hiperdüzlemi bulmayı amaçlar. Aralığı maksimuma çıkararak en uygun ayrımı yapan hiperdüzleme optimum hiperdüzlem ve bu sınır aralığını oluşturan noktalar ise destek vektörleri olarak adlandırılır. Doğrusal olarak ayrılabilen iki sınıflı bir sınıflandırma probleminde DVM'nin eğitimi için



Şekil 3: Destek vektör makineleri ile sınıflandırma

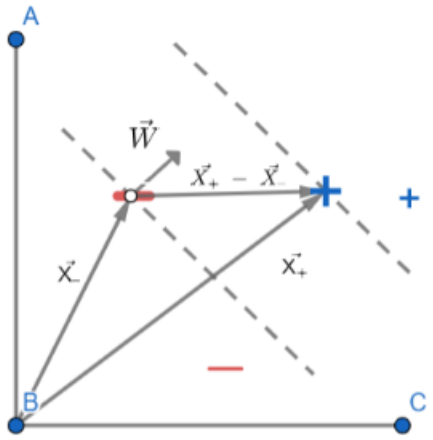
k sayıda örnekten oluşan eğitim verisini olduğu kabul edilirse, $i = 1, 2...k$ için optimum hiperdüzleme ait eşitsizlikler aşağıdaki gibi olur:

$$\begin{aligned} W \cdot X_i + b \geq +1, y = +1 \\ W \cdot X_i + b \leq -1, y = -1 \end{aligned} \tag{10}$$

Bu ifadeler aşağıdaki gibi tek bir ifade halinde yazılabilir.

$$y_i(W \cdot X_i + b) \geq 1 \tag{11}$$

SVM algoritması sınıflandırma yapmak için sınır aralığı maksimum olan hiperdüzlemi bulmayı amaçladığı için sınır aralığını W ve b cinsinden ifade edersek maksimizasyon problemini elde ederiz. Pozitif taraftaki destek vektörünü belirleyen veköre X_+ neagtif yöndeki vektöre X_- dersek. Bu iki vektörün farkı aldındığında $X_+ - X_-$ vektörü elde edilir.



Şekil 4: Destek vektörleri ve sınır aralığı

Böylece $X_+ - X_-$ vektörü W yönündeki birim vektör ile çarpıldığında sınır aralığının uzunluğu ifade edilmiş olur.

$$d = (X_+ - X_-) * \frac{\overrightarrow{W}}{||\overrightarrow{W}||} \tag{12}$$

(13) denklemini düzenlenirse

$$d = \frac{(X_+ * \overrightarrow{w} - X_- * \overrightarrow{W})}{||\overrightarrow{W}||} \tag{13}$$

eşitliğini elde edilir. Denklem (11) den destek vektörleri üzerindeki noktalar için

$$\begin{aligned} W \cdot X_+ + b &= +1 \\ W \cdot X_- + b &= -1 \end{aligned} \tag{14}$$

olduğunundan $W \cdot X_+$ ve $W \cdot X_-$ ifadelerini yalnız bırakırsak

$W \cdot X_+ = +1 - b$
$W \cdot X_- = -1 - b$

eşitlikleri elde edilir. Bu eşitlikler denklem (14) de yerine yazıldığında

$$d = \frac{((1-b) - (-1-b))}{||\overrightarrow{W}||} = \frac{2}{||\overrightarrow{W}||} \tag{15}$$

denklemi elde edilir. Buradan sınır aralığını maksimize etmek için hiper düzlemin normalinin normunu minimize edilmesi gerektiği sonucuna ulaşılır. Minimize edilmek istenilen fonksiyon matematiksel kolaylık için

$$\frac{1}{2} ||W||^2 \tag{16}$$

şeklinde belirlenebilir.

Denklem (17) elde edilen sonuç, denklem (12) de tanımladığımız destek vektörleri tarafından sınırlandırıldığı için minimizasyon yaparken Lagrange çarpanlarından yararlanılır. Lagrange çarpanları minimizasyon problemini çift(dual) probleme dönüştürerek problemin daha kolay çözülmesini sağlar ve eşitlik aşağıdaki gibi olur(AyhanErdoğmuş,2014):

$$L(w, b, a) = \frac{1}{2} |w|^2 - \sum_i \alpha_i (y_i (w^* x_i + b) - 1). \tag{17}$$

Lagrange fonksiyonunun w ve b'ye göre kısmi türevleri alınarak aşağıdaki Karush-Kuhn-Tucker koşulları elde edilir:

$$\frac{\partial}{\partial w} L_p = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \tag{18}$$

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{19}$$

$$\frac{\partial}{\partial b} L_p = -\sum_{i=1}^{n} \alpha_i y_i = 0. \tag{20}$$

(20) ve (21) Denklemlerindekini asal(primal) denklem olan (18) denkleminde yerine yazıldığında

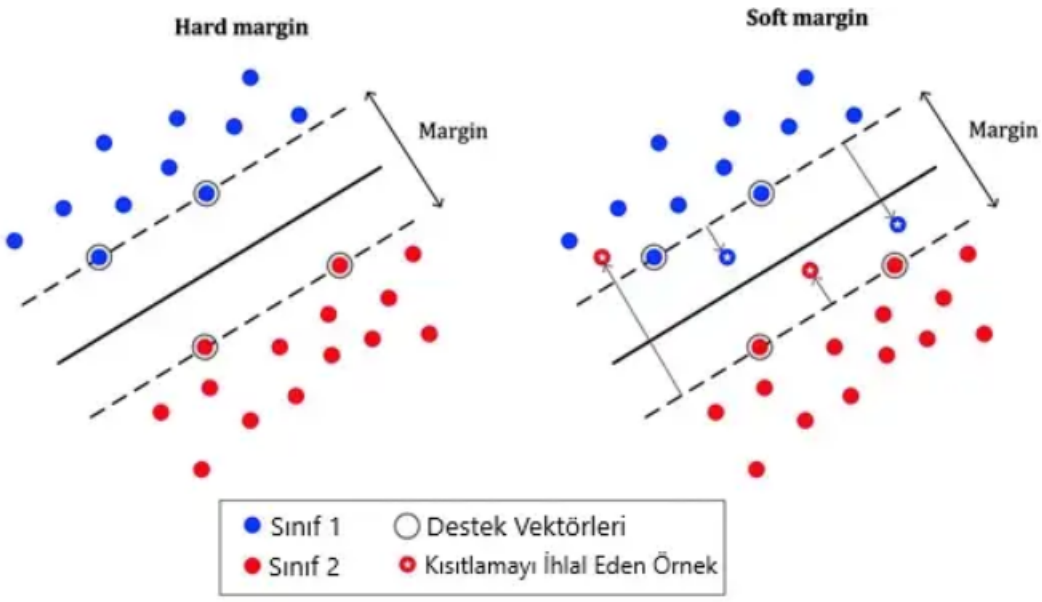
$$L(w, b, a) = \frac{1}{2} \left(\sum_i (\alpha_i y_i x_i) \sum_j \alpha_i y_i x_j \right) - \left(\sum_i \alpha_i y_i x_i \sum_j \alpha_i y_j x_j \right) - \sum_i \alpha_i y_i + \sum_i \alpha_i \tag{21}$$

$$L(w, b, a) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j \tag{22}$$

bulunur.

3.4.1 Belirli bir hata ile SVM (Soft Margin)

Bazı veri kümelerinde bütün veri noktalarını tam olarak sınıflandıracak lineer hiperdüzlemi bulmak mümkün olmayabilir. Bu durumda SVM algoritması bazı veri noktaları için hata toleransına izin verir.



Şekil 5: Destek vektör makineleri hata toleransı

Bu sayede hiperdüzlemin sınır aralığı arttırılarak model daha genel hale getirilebilir ve yeni veriler için model daha iyi performans gösterebilir.
Tolerans gösterilen veri noktaları denklem (11) de tanımladığımız denklemleri sağlamadığından eşitliklerin sağlanması için bir hata miktarı tanımlanması gerekir. Epsilon değeri, tolerans gösterilen verilerin

sınıflarını sınırlandıran destek vektörlerine olan uzaklığı olarak tanımlanırsa, hata miktarı bu epsilon değeri olur ve denklem aşağıdaki hale gelir

$$\begin{aligned} W \cdot X_i + b &\geq +1 - \epsilon, y = +1 \\ W \cdot X_i + b &\leq -1 + \epsilon, y = -1 \\ y_i(\langle W, X_i \rangle + b) &\geq 1 - \Xi_i \\ \xi_i &\geq 0. \end{aligned} \tag{23}$$

SVM algoritmasının yanlış sınıflandırma ihtimalini düşürmek için minimazsyon problemi

$$\frac{1}{2} ||W||^2 + C \sum_i \epsilon_i \tag{24}$$

olarak tanımlanır. Epsilon değeri bir hata terimidir. Hata terimi için bir fonskiyon tanımlanır ve denklem (24)'de yerine yazılır.
Hata fonksiyonu

$$L_h = \max(0, 1 - y(\langle w, x \rangle + b)) \tag{25}$$

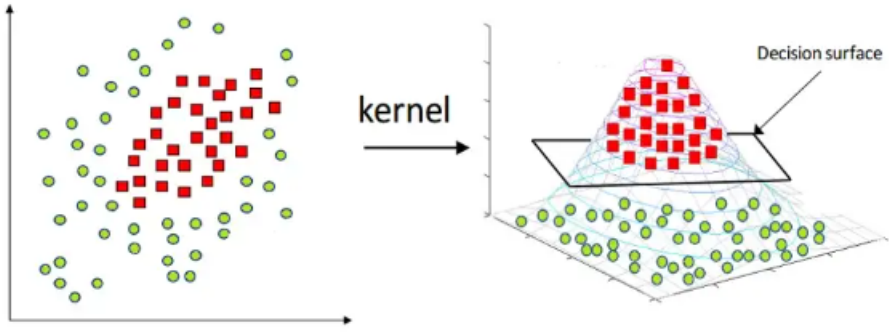
olarak ifade edilebilir ve (25) deki minizasyon problemini

$$\frac{1}{2} ||W||^2 + L_h \tag{26}$$

şeklinde tekrar yazılır.

3.4.2 Çekirdek Hilesi

SVM ile lineer olarak ayrılamayan veri kümlerini sınıflandırmak için kullanabilecek diğer yöntem çekirdek hilesidir. Bu yöntem veri noktalarını daha yüksek boyutlarda temsil ederek sınıf ayrımını yapabilmeyi amaçlar. Örnek:



Şekil 6: Çekirdek hilesi

Yukarıda görüldüğü gibi lineer olarak ayrılamayan örnek veri kümesine bir çekirdek fonksiyonu uygulandığında örnek veri kümesi daha yüksek bir boyutta temsil edilir ve sınıf ayrımı yapılabilir.

Denklem (22)ye bakıldığında amaç fonksiyonunda iç çarpım xi ve xj örnekleri arasında gerçekleşir. Bu nedenle, xi'yi temsil eden bir dizi özellik $\phi(x_i)$ düşünürsek, DVM'deki değişiklik iç çarpımı değiştirmek olacaktır. $\phi(.)$ tanımını yapmak ve x_i ve x_j örnekleri arasındaki iç çarpımı hesaplamak yerine, çekirdek fonksiyonu x_i ve x_j arasında bir benzerlik fonksiyonu olan $k(x_i, x_j)$ olarak tanımlanır.(Deisenroth et al.,2020)

$$k(x_i, x_j) = \phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j}) \tag{27}$$

Kullanılabilecek birçok çekirdek fonksiyonu olmasına rağmen yaygın olarak kullanılanlar aşağıda verildiği gibidir.

Lineer çekirdek fonksiyonu:

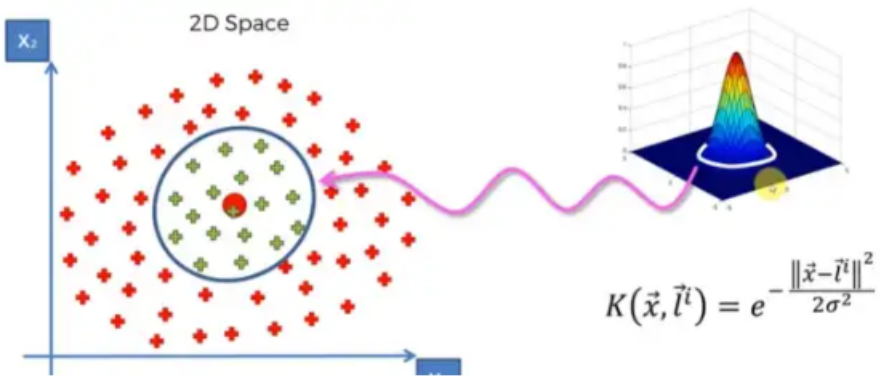
$$k(x_i, x_j) = x_i^T x_j + c \tag{28}$$

Gaussian RBF çekirdek fonksiyonu:

$$k(x_i, x_j) = exp(-\frac{||x_i - x_j||^2}{2\sigma^2}) \tag{29}$$

Sigmoid çekirdek fonksiyonu :

$$k(x_i, x_j) = tanh(ax_i^T x_j + c) \tag{30}$$

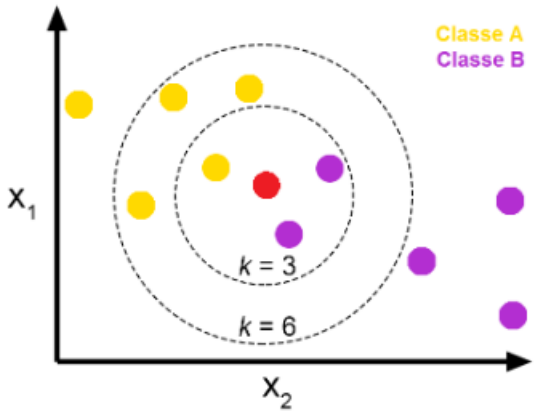


Şekil 7: rbf çekirdek dönüşümü

3.5 K En Yakın Komşu Algoritması(KNN)

Sınıflandırma problemlerinde K-En Yakın Komşu algoritması, veri kümesinde verilen bir değer için en yakın komşularını bulmak ve bu komşuların sınıf etiketlerine göre sınıflandırmak için kullanılan bir yöntemdir. Algoritma birbirine yakın veri noktalarının sınıf etiketlerinin benzer olacağı yaklaşımı ile sınıflandırma yapar. KNN ile sınıflandırma yapılırken verilen bir örneğe en yakın 'K' verinin sınıf etiketlerinden maximum sayıda olan etiket örneğin sınıfı etiketi olarak belirlenir. Burda belirlediğimiz K değerlerine göre sınıflandırmalar farklılık gösterir.

Örnek:



Şekil 8: K en yakın komşu algortması ile sınıflandırma

k=3 değeri için Euclid mesafesine göre kırmızı veri noktasına en yakın 3 verinin 2 tanesi B, 1 tanesi A sınıfdan olduğu için bu veriye B sınıf etiketi atanır. K=6 değeri için belirtilen verinin en yakın 6 komşusundan 4'ü A 2'si B sınıf etiketine sahip olduğundan veriye B sınıf etiketi atanır.
Birbirine uzaklıklık olarak yakın olan veri noktalarının sınıflarının benzer olacağını düşünüyorsak burda bahsedilen uzaklığın bir benzerlik kavramı olduğu söylenebilir. O zaman farklı uzaklık ölçüleri kullanarak bu benzerlik farklı yaklaşımlarla ölçülebilir. Kullanabilecek bazı uzaklık tanımları aşağıda verildiği gibidir.

Euclid Mesafesi :

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

Manhattan Mesafesi :

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

Minkowski mesafesi :

$$d = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p}$$

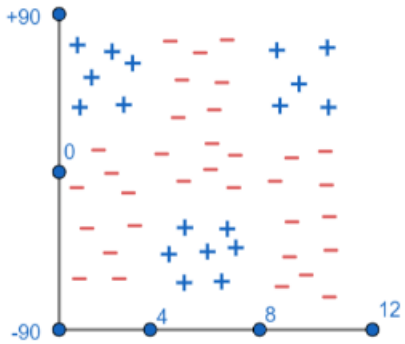
Kosinüs Mesafesi:

$$d = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \cdot ||\mathbf{y}||}$$

Özellikle ikili sınıflandırma problemlerinde K değeri bir çift sayı olarak belirlendiğinde bazı verilerin en yakın komşuların sınıf etiketlerinin sayısı birbirine eşit olabilir. Bu durumda algoritmanın bu veri noktalarını yanlış sınıflandırma ihtimali yüksektir. Bu yüzden ikili sınıflandırmada problemlerinde K değerini tek sayı olarak belirlemek daha uygun olur.

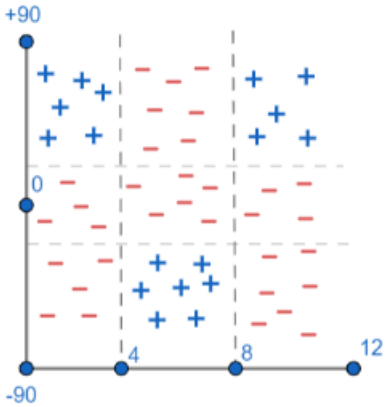
3.6 Karar Ağaçları

Karar ağaçları karmaşık veri kümelerini sınıflandırmak için belirli karar kurallarıyla daha küçük veri kümelerine ayrırarak sınıflandırma yapar.
Örnek:
X_1 değişkeninin ayları , X_2 değişkeninin enlemleri belirttiğini varsayılsın ve y=0,1 sınıf etiketleri kayak yapılabilmesi durumu olsun. Grafikte 0 sınıfını - , 1 sınıfını + değerler olduğunu kabul edelim. Kuzey yarım kürede sonbahar ve kış mevsimleri yılın ilk ve son aylarında, Güney yarım kürede ise yılın orta aylarında görüldüğü için aşağıdaki gibi bir veri kümesi elde edilir.



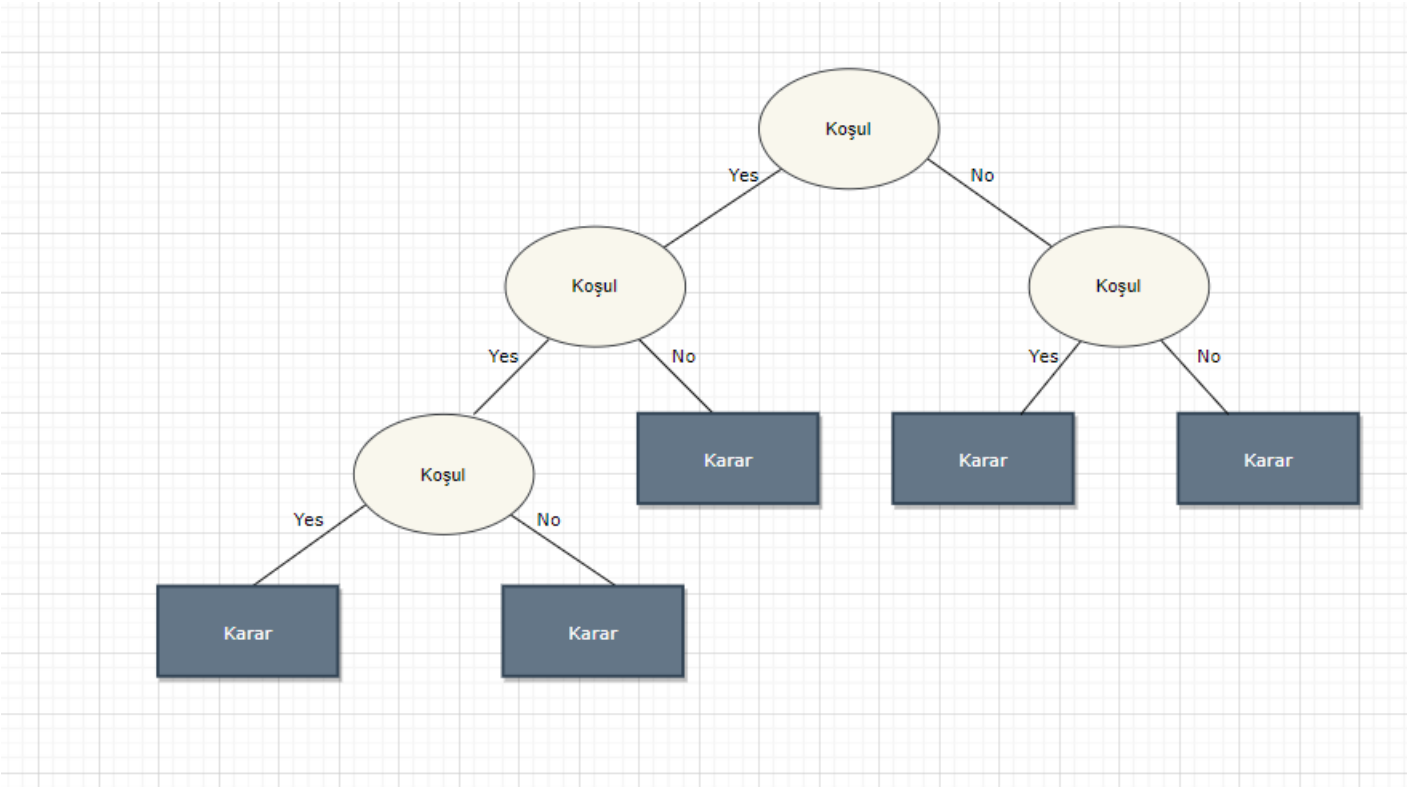
Şekil 9: Kayak yapılabilme için örnek veri kümesi

Bu verileri sınıflandırabilmek için enlemler ce yılın bazı ayları için karar sınırları oluşturabilir. -15 ve +15 enlem derecelerine ve 4. ve 8. aylara sınırlar çizdildiğinde aşağıdaki bölünmüş veri kümesi elde edilir.



Şekil 10: Kayak yapılabilme örnek veri kümesi için karar sınırları

Bu yapı artık bir model olarak kullanılırsa yeni bir veri noktasının bulunduğu bölgedeki sınıf etiketlerine bakılarak sınıflandırma yapılabilir.
Karar ağaçlarının tanımında bahsedildiği gibi bu sınıflandırma süreci bir dizi karar kuralını takip ederek gerçekleşir ve ağaç yapısı aşağıdaki gibi olur:



Şekil 11: Karar ağacı

Karar ağaçlarında,verileri bölmek için oluşturulan karar yapıları sınıf etiketlerinin düzenine göre oluşturulur. Daha doğru sınıflandırma yapılabilmesi için aynı bölümde olan verilerin sınıf etiketlerinin düzenli yani tekdüze olması hedeflenir.Karar ağaçları verilerin düzenli olarak bölebilmek için bilgi kazancından faydalanır.

Bilgi kazancını tanımlamak için önce entropi kavramının anlaşılması gerekir. Shanon'un teorisine göre bir olayın gerçekleşmesinde rastgelelik ne kadar fazla ise taşıdığı bilgi o kadar fazladır.Entropi ise rastgeleliğin beklenen değeridir. Buradaki rastgelelik değeri gerçekleşme sıklığı ile ilişkilidir ve

$$I = \log_2\left(\frac{1}{P(X)}\right) \tag{31}$$

formülü ile haseplanır. Rastgeleliğin beklenen değeri hesaplandığında entropi formülü

$$H(X) = E(\log_2\left(\frac{1}{P(X)}\right)) = -\sum(\log_2(P(X)) \cdot P(X)) \tag{32}$$

olarak elde edilir. Aşağıdaki $S1$ ve $S2$ dizileri üzerinde için bu kavramlar incelenirse:

$S1$= [0,1,0,1,0,1,0,1,0,1]
$S2$= [1,1,1,1,1,1,1,1,1,1]

Yukarıda belirttiğimiz teoriye göre $S1$ sınıfı $S2$ sınıfına göre daha fazla rastgelelik içerdiği için S1 sınıfı daha çok bilgi taşıması beklenir. Entropi değerleri incelendiğinde

$$\begin{aligned} H(S1) &= -0.5 \cdot \log(0.5) - 0.5 \cdot \log(0.5) = 1 \\ H(S2) &= -1 \cdot \log(1) - 0 \cdot \log(0) = 0 \end{aligned} \tag{33}$$

veri sıklığı daha daha düzensiz olan $S1$ dizisinin entropisinin ve taşıdığı bilginin daha yüksek olduğu söylenebilir.

Karar ağaçları verileri hangi sütun özelliğine göre böleceğine karar vermek için her sütun için bilgi kazancını hesaplar. Hedef değişkenin entropi değeri $H(T)$, veri kümesinin a sütun özelliğine göre bölündüğünde ağırlıklı entropisi $H(T|a)$ olduğunda bilgi kazancı aşağıdaki gibi olur(Shalev-Shwartz Ben-David,2014).

$$IG(T,a) = H(T) - H(T|a) \tag{34}$$

Herhangi a sütunundaki tekil değerler $(u_1, u_2...u_i)$ ve sütunun bü değerlere göre bölünmesiyle elde edilen parçalar $(S_1, S_2...s_i)$ olduğunda ağırlıklı entropi değeri

$$H(T|a) = \sum_{i=1}^{n} \left(\frac{|s_i|}{T} \cdot H(s_i) \right) \tag{35}$$

olarak tanımlanır. Bilgi kazancı ise

$$IG(T,a) = H(T) - \sum_{i=1}^{n} \left(\frac{|s_i|}{T} \cdot H(s_i) \right) \tag{36}$$

şeklinde tekrar ifade edilebilir. Karar ağaçları veri kümesini hangi sütun özelliğine göre böleceğine karar vermek için bütün sütunlar için bilgi kazancını hesaplar ve bilgi kazancı maksimum olan sütuna göre verisetini böler. Bölünme sonucunda geriye kalan verileri üzerinde aynı işlemlerin uygulanmasıyla ağaç yapısı dallanmaya devam eder.
Örnek:

	c1	c2	Hedef Değişken
0	1	0	A
1	0	1	B
2	1	0	A
3	1	0	B
4	1	1	A
5	0	0	B
6	0	1	A
7	1	1	A
8	0	0	B
9	0	1	B

Şekil 12: Karar ağaçları için uygulama veri kümesi

Yukaridaki veri kümesi üzerinden bilgi kazancı kavramını incelenirse denklem (28)' den

$H(T) = -(0.5 \log_2(0.5) + 0.5 \log_2(0.5)) = 1$

elde edilir. c1 sütunu için ağırlıklı entropiyi hesaplamak için c1 sütununa göre ayrılmala incelenirse:

```
df[df["c1"]==1]
```

	c1	c2	Hedef Değişken
0	1	0	A
2	1	0	A
3	1	0	B
4	1	1	A
7	1	1	A

```
df[df["c1"]==0]
```

	c1	c2	Hedef Değişken
1	0	1	B
5	0	0	B
6	0	1	A
8	0	0	B
9	0	1	B

Şekil 13: c1 sütun değerlerine göre ayrılma

c1 sütunundaki 1 değeri için 4 A sınıfından, 1 B sınıfından veri olduğu için entropi değeri

$-(\frac{4}{5}\log_2(\frac{4}{5}) + \frac{1}{5}\log_2(\frac{1}{5})) = 0.72$

olur.0 değeri için 1 A sınıfından , 4 B sınıfından veri olduğu için entropi değeri

$-\left(\frac{1}{5}\log_2\frac{1}{5} + \frac{4}{5}\log_2\frac{4}{5}\right) = 0.72$

olur. c1 sütunu için denklem (31) kullanılarak bilgi kazancı hesaplanırsa

$1 - (1/2 * 0.72) + (1/2 * 0.72) = 0.28$

sonucuna ulaşılır.c2 sütunu için bilgi kazancı hesaplanırsa

```
df[df["c2"]==1]
```

	c1	c2	Hedef Değişken
1	0	1	B
4	1	1	A
6	0	1	A
7	1	1	A
9	0	1	B

```
df[df["c2"]==0]
```

	c1	c2	Hedef Değişken
0	1	0	A
2	1	0	A
3	1	0	B
5	0	0	B
8	0	0	B

Şekil 14: c2 sütun değerlerine göre ayrılma

c2 sütunundaki 1 değeri için 3 A sınıfından, 2 B sınıfından veri olduğu için entropi değeri

$$-\left(\frac{3}{5}\log_2\frac{3}{5}+\frac{2}{5}\log_2\frac{2}{5}\right)=0.97$$

olur.0 değeri için 2 A sınıfından , 3 B sınıfından veri olduğu için entropi değeri

$$-\left(\frac{2}{5}\log_2\frac{2}{5}+\frac{3}{5}\log_2\frac{3}{5}\right)=0.97$$

olur ve c2 sütunu için denklem (31) kullanılarak bilgi kazancı hesaplanırsa

$1-(1/2*0.72)+(1/2*0.72)=0.03$ sonucuna ulaşılır.

Bilgi kazancı c1 sütunu için daha yüksek olduğu için veri kümesi öncelikli olarak c1 sütununa göre bölünür. Ayrıca veri kümesine bakıldığında c1 sütununda sınıf etiketlerinin daha homojen dağıldığı farkedilir yani karar ağaçları bilgi kazanımı kullanılarak sezgisel olarak doğru şekilde verileri bölebilir . Karar ağaçlarında verilerin sürekli olarak bölünmesi sonucunda model eğitim veri kümesini aşırı öğrenecektir ve yeni veriler üzerinde yanlı tahminler yapacaktır. Bu sebepten bölünme işlemini bir bölünme sayısından sonra durdurmamız gerekir.

Kaynaklar

(1) Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). *Mathematics for Machine Learning*. *Mathematics for Machine Learning*. Cambridge University Press.

(2) Ayhan, S., & Erdoğmuş, Ş. (2014). Destek Vektör Makineleriyle Sınıflandırma Problemlerinin Çözümü İçin Çekirdek Fonksiyonu Seçimi Sevgi AYHAN Şenol ERDOĞMUŞ. *ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ İİBF DERGİSİ*, *9*(1), 175–198.

(3) Shalev-Shwartz, S., & Ben-David, S. (2013). *Understanding machine learning: From theory to algorithms*. *Understanding Machine Learning: From Theory to Algorithms* (Vol. 9781107057135, pp. 1–397). Cambridge University Press.