T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

MAKİNE ÖĞRENME ALGORİTMALARI İLE WEB SİTELERİ TIKLAMALARININ ANALİZİ

Yüksek Lisans Tezi

Tezi Hazırlayan: **Tevfik ÇOBAN**

İstanbul, 2011

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

MAKİNE ÖĞRENME ALGORİTMALARI İLE WEB SİTELERİ TIKLAMALARININ ANALİZİ

Yüksek Lisans Tezi

Tezi Hazırlayan**:**

Tevfik ÇOBAN

Öğrenci No:
080820005

Danışman:
Yrd.Doç.Dr. Zeynep ALTAN

İstanbul, 2011

YEMİN METNİ

Yüksek lisans tezi olarak sunduğum "Makine Öğrenme Algoritmaları ile Web Siteleri Tıklamalarının Analizi" başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullanıldıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım. 13.06.2011

Tevfik ÇOBAN

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZ SAVUNMA SINAVI SONUÇ TUTANAĞI

Beykent Üniversitesi
Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Aşağıda tez adı belirtilen yüksek lisans öğrencisi 0806200005 no'lu Tevfik ÇOBAN'ın 01/08/2011 tarihinde yapılan tez savunma sınavı[1] sonucunda 60 dakika süreyle sunduğu ve savunduğu tezi hakkında[2] oybirliğiyle/~~oyçokluğuyla,~~ Kabul/~~Red/Düzeltme~~(.....ay içinde) kararı verilmiştir.

Bilgilerinize saygılarımızla arz ederiz.

Anabilim Dalı : Bilgisayar Mühendisliği
Programı : Bilgisayar Mühendisliği
Tez Başlığı[3] : Makine Öğrenme Algoritmaları ile Web Siteleri Tıklamalarının Analizi

---

Tez Sınav Jürisi	**Öğretim Üyesi**	**İmza**
Danışman	: Yrd. Doç. Dr. Zeynep ALTAN	
Üye	: Yrd. Doç. Dr. Gökhan SİLAHTAROĞLU	
Üye	: Prof. Dr. Esat HAMZAOĞLU	

[1] Jüri üyeleri söz konusu tezin kendilerine teslim edildiği tarihten itibaren en geç bir ay içinde toplanarak öğrenciyi tez savunma sınavına alır. Belirlenen günde yapılamayan jüri toplantısı, katılanların hazırladığı bir tutanakla enstitü yönetimine bildirilir. Bu durumda jüri en geç onbeş gün içinde toplanarak adayı tez savunma sınavına alır. Tez savunma sınav süresi en az 45 dakikadır. Yüksek lisans tez savunma sınavı, tez çalışmasının sunulması ve bunu izleyen soru-yanıt bölümlerinden oluşur ve dinleyiciye açıktır. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-3)

[2] Tez sınavının tamamlanmasından sonra jüri, tez hakkında "kabul", "düzeltme" veya "red" kararı verir. Jüri başkanı, jüri üyelerince imzalanmış sınav tutanağını, tez sınavını izleyen üç gün içinde ilgili enstitü yönetimine teslim eder. Tezi başarısız bulunan öğrencinin Enstitü ile ilişiği kesilir. Tezi hakkında düzeltme kararı verilen öğrenci en geç üç ay içinde gerekli düzeltmeleri yaparak ve yönetmelikte belirtilen usullere uygun olarak tezini aynı jüri önünde yeniden savunur. Bu savunma sınavında da tezi kabul edilmeyen öğrencinin enstitü ile ilişiği kesilir. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-4)

[3] İleride doğabilecek aksaklıkların engellenmesi için tezin başlığının yazılması gerekmektedir.

MAKİNE ÖĞRENME ALGORİTMALARI İLE WEB SİTELERİ TIKLAMALARININ ANALİZİ

Tezi Hazırlayan: Tevfik ÇOBAN

Özet

Bu tez çalışmasında, Türkiye'de yaygın olarak tıklanan web sitelerinin istatistiksel verileri kullanılarak, makine öğrenme algoritmaları ile analizi yapılmıştır. Elde edilen veriler üzerinde makine öğrenmesinin başarısının nasıl gerçekleştiği, bu veriler arasında web sitesi trafiğindeki en belirleyici parametreler tespit edilmiştir. Bu tespitler, hem gözetimli öğrenme algoritmalarından Naive Bayes, Bayes Ağı, K En Yakın Komşu, Destek Vektör Makinesi, ID3 ve C4.5 algoritmaları ile hem de gözetimsiz öğrenme algoritmalarından K-means ve Hiyerarşik Kümeleme algoritmaları ile gerçekleştirilmiştir. Eğitim-test, çapraz doğrulama gibi farklı seçeneklerle ayrıntılı olarak incelenen bu algoritmaların birbirine göre başarı ve performans kıyaslaması yapılarak web siteleri tıklamaları analizi üzerindeki uygun ve uygun olmayan algoritmalar belirlenmiştir.

Ayrıca, gözetimsiz öğrenme algoritmaları kullanılarak web sitelerinin kümelendirilmesi gerçekleştirilmiştir. Web sitelerinin türü ve özelliklerinin, ziyaretçilerin tıklama üzerine davranışlarının nasıl değiştiği üzerinde yorumlar ve değerlendirmelere yer verilmiştir.

Anahtar Kelimeler: Makine öğrenmesi, Gözetimli öğrenme, Gözetimsiz öğrenme, Kümeleme, Sınıflandırma, Web siteleri tıklama analizi, Web sitesi istatistikleri

ANALYSIS OF WEBSITE CLICKS WITH MACHINE LEARNING ALGORITHMS

Presented by: Tevfik ÇOBAN

Abstract

In the work presented, statistical data of web sites which are the most common clicked in Turkey are analyzed with machine learning algorithms. For website traffic, the most decisive parameters of this statistical data are identified. Both some supervised learning algorithms like Naive Bayes, Bayesian Network, K Nearest Neighborhood, Support Vector Machines, ID3, C4.5 algorithms and some unsupervised learning algorithms like K-Means, Hierarchical Clustering algorithms are used for these determinations. These algorithms are investigated with different options like training –test, cross validation and performance and success of these algorithms are compared to each other, for the websites clicks analysis appropriate and inappropriate algorithms are selected.

Also, using unsupervised learning algorithms websites are clustered. This study include reviews and assessments about effect of type and characteristics of websites on visitor's click behavior.

Keywords: Machine Learning, Supervised Learning, Unsupervised Learning, Clustering, Classification, Click Analysis of Websites, Website Statistics

TEŞEKKÜR

Tüm tez çalışması süresince disiplinli ve özverili yaklaşımı ile beni yönlendiren, alternatif fikirleri ve etkin önerileri ile çalışmanın sonlandırılabilmesinde büyük katkısı olan ve kaynak elde etme konusunda her an yardımcı olan, tez danışmanım Yrd.Doç.Dr. Zeynep ALTAN'a, tüm desteğinden dolayı çok teşekkür ederim.

İÇİNDEKİLER **Sayfa No.**

TABLOLAR LİSTESİ

Sayfa

ŞEKİLLER LİSTESİ

Sayfa

LİSTELER

Sayfa

KISALTMALAR

Web : World Wide Web (Dünya Çapında Ağ)

E-ticaret : Elektronik ticaret (İnternet vb. bilişim ağları üzerinden yapılan tüm bilgi, hizmet, para vs. gibi ekonomik değerlerin değişimi süreci)

WEKA : Waikato Environment for Knowledge Analysis (Veri madenciliği-Makine öğrenmesi alanında kullanılan bir yazılım.)

1. GİRİŞ

1.1. Makine Öğrenmesi

İnsan doğumundan itibaren, yaşamak için çevresine uyum sağlamak ve bu uyumu gerçekleştirebilmek için de sürekli olarak öğrenmek zorundadır. Bu nedenle yaşamını sürdürebilmek için gereksinim duyduğu bilgi, beceri, tutum ve davranışlarının çoğunu öğrenerek kazanmaktadır.

Bir bilgi ve becerinin, öğrenme sayılması için davranışta değişiklik yapması ve bu değişikliğin uzun süreli olması gerekmektedir. Aynı zamanda ortaya çıkan davranış değişikliğinin yaşantı ürünü olması da gerekmektedir. Yeni öğrenmeler ile kişinin kapasitesi gelişir, önceden yapamadığı bir şeyi yapabilir hale gelir. Daha geniş anlamda, öğrenme sonucu, birey içinde bulunduğu çevreye, ortama, evrene yeni bir anlam yükler ve konumunu yeniden tanımlar.

Geçmişten bu yana bilgisayarların insanlardaki gibi öğrenme, kavrama ve akıl yürütme yapıp yapamayacağı araştırılmıştır. Örneğin, borsa analizinde döviz, altın ya da hisse senetlerinin gelecek zamanlarda değerlerinin ne olacağı, bilgisayar sistemlerinin insan sesini, yazıya nasıl dökebileceği ya da hırsızlığa karşı konulmuş olan bir güvenlik kamerasının bu durumu nasıl anlayabileceği gibi konular araştırma ve merak konusu olmuştur. Örneklere bakıldığında, bu sistemlerde geçmiş bilgiler ile gelecekteki bir durum için tahmini bir bilgi yani tecrübe oluşturulmak istendiği anlaşılmaktadır. Ancak yapılan araştırma ve incelemeler, bilgisayar sistemlerinin, insan beyninin pek çok özelliğini başarıyla gerçekleştirmesine rağmen öğrenme ve tecrübe kazanma konusunda yetersiz olduğunu ortaya koymuştur.

Bir bilgisayar sistemi belirli bir konuda çok özel problemleri sorunsuz çözebilmesine rağmen, aynı probleme benzer başka bir problemle karşılaştığında bir önceki çözümden elde ettiği bilgi ve tecrübeleri kullanmamakta ve çözüm için yaptığı işleri yeni baştan tekrarlamaktadır. Aynı problemle çok kere karşılaştığında da her defasında aynı çabayı harcayarak aynı sonuca ulaşmaktadır. Bu problem, yapay zekâ araştırmacılarının odak noktası olmuş, baştan beri ulaşmak istenilen ideal, insan gibi davranan, geçmiş tecrübelerini kullanabilen, idrak edip karar verme yeteneğine sahip sistemleri üretmek olmuştur. Bu sorunu aşmanın yolu, bilgisayar

sisteminin öğrenmeyi gerçekleştirmesidir. Öğrenen bir bilgisayar sistemi, oluşan durumlara göre hem kendini yenileyebilecek hem de benzer problemleri önceki duruma göre daha etkin ve hızlı bir şekilde çözebilecektir[1].

Yukarıda anlatılan problemlere çözüm aranması makine öğrenmesi alanının doğmasına ve gelişmesine neden olmuştur. Bu açıdan makine öğrenmesi, örnek verileri kullanarak, ya da geçmişteki deneyimlerden yararlanarak, tümevarım yöntemiyle tanımlayıcı ya da tahminsel çıkarımlar yapacak şekilde bilgisayar programlamak olarak tanımlanmaktadır[2]. Diğer tanımıyla makine öğrenmesi, bilgisayarların veritabanlarına dayalı öğrenimini olanaklı kılan algoritmaların tasarım ve geliştirme süreçlerini inceleyen bir bilim dalıdır.

Makine öğrenmesi araştırmalarının ağırlık verildiği konu, bilgisayarlara karmaşık örüntüleri algılama ve veriye dayalı akılcı kararlar verebilme becerisi kazandırmaktır. Makinelere bu becerileri kazandırabilmek için birçok algoritma geliştirilmiştir. Ancak geçmiş bilgilere ait veri kümelerinin sonlu oluşu ve geleceğin tam olarak kestirilememesi nedeniyle öğrenmeyi sağlayacak algoritmaların başarısına ait kesin bir güvence verilememektedir. Bunun yerine, algoritma başarısının olasılıksal sınırları öngörülmeye çalışılmaktadır. Makine öğrenme algoritmalarının çoğu sayısal verilerle çalışmaktadır. Bu nedenle işlenecek olan veriler sayısallaştırılır. Örneğin veri metin ise, metindeki harfler, heceler ve kelimeler genelde frekanslarına göre kodlanarak sayılara çevrilir.

Makine öğrenmesinin başlıca uygulamaları makine algılaması, bilgisayarlı görme, doğal dil işleme, sözdizimsel örüntü tanıma, arama motorları, tıbbi tanı, biyoinformatik, kredi kartı dolandırıcılığı denetimi, borsa yatırımları ve çözümlemesi, DNA dizilerinin sınıflandırılması, konuşma ve el yazısı tanıma, desen tanıma, nesne tanıma, oyun oynama, yazılım mühendisliği, uyarlamalı web siteleri ve robot gezisidir. Bu teknolojilerinin analiz alanında, son yıllardaki en önemli gelişmelerden birisi karar destek sistemlerinin ihtiyacı olan bilgiyi üretmeye aday bir sistem olarak ortaya çıkan bilim dalı da veri madenciliğidir. Veri madenciliği, tanım olarak, büyük miktarda ve oldukça hızlı toplanan verilerin, çeşitli analizler sonucunda anlamlı bilgilere dönüştürülmesi noktasında devreye giren süreçtir. Veri madenciliği çalışmalarındaki gerçek amaç, gözden kaçan veya insan faktörü ile

tespitinin mümkün olmadığı durumların tespitini sağlamak ve geçmişe bakarak geleceğin tahminini gerçekleştirmektir.

Veri madenciliği; veri tabanları, istatistik ve makine öğrenmesi konularının kavramlarına dayanır ve onların tekniklerini kullanır. Şöyle ki, makine öğrenmesi yöntemleri, veri madenciliği algoritmalarında kullanılan yöntemlerin çekirdeğini oluşturur. Örneğin, makine öğreniminde kullanılan bir karar ağacı, kural çıkartımı pek çok veri madenciliği algoritmasında kullanılmaktadır[3]. Dolayısı ile makine öğrenmesi; istatistik, olasılık kuramı, veri madenciliği, örüntü tanıma, yapay zekâ, uyarlamalı denetim ve kuramsal bilgisayar bilimi arasında yakın bir bağ vardır. Bu disiplinler, yarım ya da tam otomatik süreçleri gerektirecek veri içindeki kuralları ve kullanılabilecek bilgileri bulmayı amaçlar.

1.2. Web Sitesi ve İstatistikleri

Web siteleri, teknolojinin hızla geliştiği günümüzde bilgi erişimi ve paylaşımı, ticaret, pazarlama, reklamcılık, eğlence ve sosyal iletişim gibi amaçlarla kullanılması bakımından iyi bir araç konumundadırlar.

Kurumsal açıdan bakıldığında, web siteleri ile geniş bir kitleye hiçbir aracı kullanmadan ulaşmak, kurum imajını güçlendirmek, kurum ile ilgili gelişmeleri ve anında duyurabilmek için kaçınılmaz hale gelmiştir. Bunun yanında, müşteri ile karşılıklı etkileşime olanak sağlayarak, satışları artırma, yeni pazarlar ve müşteriler elde etmede, kurum giderlerini azaltmada büyük yararlar sağlamaktadır. Örneğin, web sitesi aracılığı ile internet bankacılığı hizmeti açan bir banka, müşterisinin bankaya gitmeden birçok işlem yapabilmesini sağlayıp, müşterisini bünyesinde tutmayı, internet bankacılığının kullanımının artmasıyla daha fazla kar yapmayı ve personel giderlerinin azaltmayı başarabilir. Bu da web sitesinin önemini ortaya koyan bir nedendir.

Web sitelerinin alışveriş sitesi olarak hizmete olanak vermesi, artan kredi kartı kullanımı ve kargolama, postalama sektörlerindeki gelişmelerle elektronik ticarete yoğun bir yönelim söz konusudur. e-ticaret alanında faaliyet gösteren bir web sitesi şirketinin en çok önem vereceği konulardan biri, web sitesinin ziyaretçiler tarafında her zaman popüler olmasıdır. Çünkü bir e-ticaret sitesinin ürün satış miktarının, site popülerliği ile orantılı olması beklenir.

Sosyal paylaşım, arkadaşlık, oyun, eğlence, video, müzik yayınları da web siteleri arasında rağbet gören site kategorilerindendir. Bu web siteleri, doğrudan ürün satış veya pazar amaçlı kurulmamış olsalar bile gerek web sitesi yöneticilerinin maddi beklentilerini karşılaması, gerek web sitesinin yayın hayatına devam etmesi, güncelleme, bakım gibi uğraşlar sonucunda yapılan giderleri karşılaması bakımından da dolaylı yoldan gelir elde etme ihtiyacı söz konusudur. Daha çok arkadaşlık ve dosya paylaşım sitelerinde görülmekte olan ücretli üyelik (aylık - 3 aylık- 1 yıllık gibi) yöntemiyle gelir elde edilebileceği gibi, web sitelerine alınan reklamlar aracılığı ile gelir elde edilebilmektedir.

Web sitesinin popülerliği, sahipleri kadar reklam verenler açısından da dikkate alınabilen bir kriterdir. Çünkü reklam verecek olan bir şirket, reklam maliyetini de düşünerek, maksimum hedef kitleye ulaşıp, karını arttırmak istemektedir. İnternet reklamcılığının da, diğer reklam türlerine göre müşterilere daha fazla bilgi sunabilmesi imkânı bakımından üstünlüğünün olması, web sitelerinde reklamcılığı arttıran bir unsurdur.

Web sitesi istatistikleri ise, bir siteye ziyaretçilerin ilgi ölçütünü, o siteyi kullanma alışkanlıklarını ve ziyaretçi profili hakkında fikir edinebilmek amacıyla kaydı tutulan değerler bütünüdür. Web sunucuları tarafında tutulabilen kayıtlar olduğundan ve web sitelerinin popülaritesini kıyaslamada net ve doğru bir bilgi sunduklarından bu istatistiklerin önemi giderek artmaktadır.

Bir reklam veren, bir web sitesine reklam vermek istediğinde hangi siteyi neye göre değerlendirmelidir? Reklam verilen bir web sitesinin çok müşteri kazandıracağı sadece ziyaretçi sayısına mı bağlıdır, yoksa o web sitesinde çok zaman geçiriliyorsa bu durum yeterli midir? Web istatistikleri değerleri bakımından iyi ya da kötü olan bir web sitesinin bu konumunu en çok belirleyici istatistikî parametre hangisidir? Maddi gelir ya da hobi amaçlı olarak bir kişi, çok ziyaret edilen bir web sitesi isteği varsa, hangi konuya yönelik türde bir web sitesi hazırlamalıdır? gibi sorular web siteleri istatistikleri tutulması, analiz edilmesi, değerlendirilmesi ve yorumlanması sürecini başlatan sorulardan birkaçıdır.

Yukarıdaki beklentiler ışığında, yapılan bu çalışmada Türkiye'de Ocak 2011 tarihinde en çok ziyaretçi almış 1000 adet web sitesinin web istatistiklerine ait analizlerinin makine öğrenme algoritmaları ile nasıl yapılabileceği gösterilmiştir.

Tasarlanan yönteme göre hangi algoritmaların bu analizlerde uygun, hangilerinin uygun olmadığı ve algoritma başarı performanslarının nasıl değiştiği de incelenerek bu alandaki ortaya atılan sorulara cevap verilmeye çalışılmıştır.

Başarılı algoritmaların tespitinden sonra, elde edilen sonuçlarda, web sitesi istatistiklerindeki parametrelerin, birbirlerine göre hangilerinin uygulamalarda daha yararlı olabileceği araştırılmış, önem taşıyan parametreler tespit edilmiştir.

Ayrıca, ziyaretçilerin farklı site türlerine ilgisini ve bu site profillerini ortaya çıkarmak için verilerin kümelenmesi sağlanmış ve buna ilişkin yorum ile değerlendirmelere yer verilmiştir.

Yapılan bu çalışmanın içeriği özetlenecek olunursa, 2. Bölüm'de makine öğrenmesi yöntemlerinden 6 farklı gözetimli öğrenme algoritması ve 2 farklı gözetimsiz öğrenme algoritmaları anlatılarak sayısal örneklerle pekiştirilmiştir. 3. Bölüm'de çalışmada kullanılan verilerin elde edilmesi, analiz edilmesini sağlayan uygulama ve veri dönüşümlerine değinilmiştir. 4.Bölüm'de makine öğrenme algoritmalarının başarı ve performans ölçümünde kullanılacak kavramlar açıklanmıştır. Daha sonra çalışmada kullanılan verilerin makine öğrenmesi algoritmaları ile sınaması gerçekleştirilerek, algoritmaların elde edilen başarı oranları da karşılaştırılarak değerlendirmelere yer verilmiştir. 5. Bölüm'de ise kullanılan yöntemle elde edilen yararlı bilgiler özetlenerek sonuçlar açıklanmıştır.

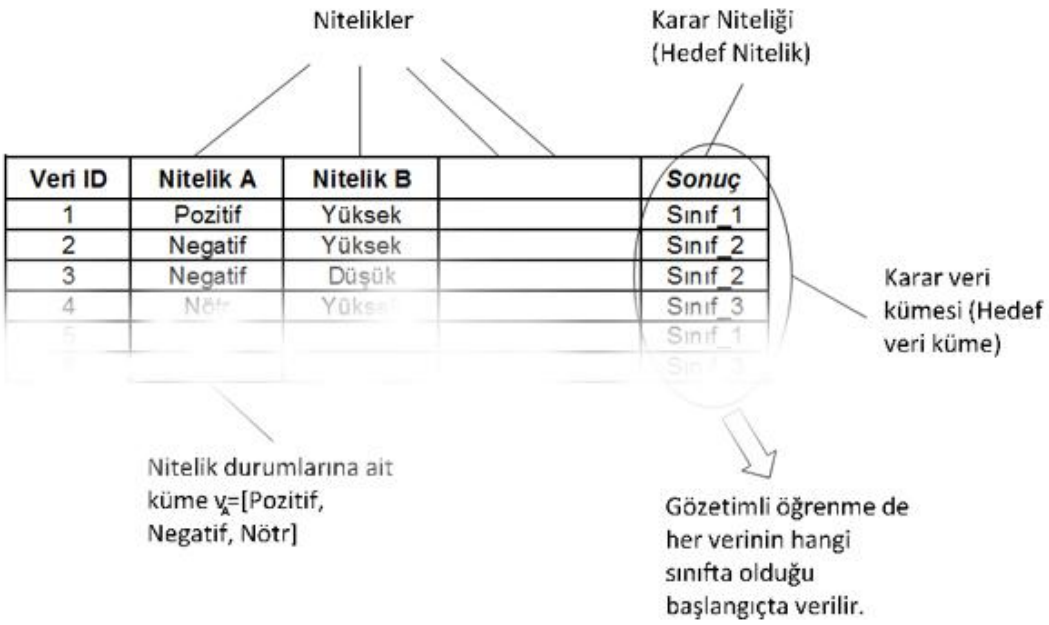
2. MAKİNE ÖĞRENMESİ YÖNTEMLERİ

Bilgisayarların basit bir problemi çözmesi, genellikle gerekli çıktıyı bir girdiler kümesinden türetme yönteminin açıkça tanımlanmasıdır. Sistem tasarımcısının veya programcının görevi; o yöntemi, bilgisayarın istenilen etkiye ulaşmak için takip edeceği direktiflerin sıralamasına çevirmek veya programlamaktır. Ancak bilgisayarları programlamak her türlü probleme çözüm getirmeyebilir. Çünkü girdi verilerine karşılık istenilen çıkışı elde etmenin bilinen bir yöntemi olmayabilir. Gerekli yöntem açıkça ifade edilemediğinden, bu örnekler klasik programlama yaklaşımı ile çözülemez. Bu tür problemlerin çözülebilmesi için alternatif yol deneyimlerden yararlanılmasıdır. Örneğin bir çocuğa birçok hayvan içerisinde hangilerinin kuş olduğunun kendisine söylenerek kuşları öğrenmesi gibi, bilgisayarın da giriş/çıkış işlevselliğini örneklerden öğrenmesidir. Programları sentezlemek için örnek kullanma yaklaşımına *makine öğrenme yöntemi* denir. Öğrenme süreci içinde yer alan giriş/çıkış işlevselliğinin örnekleri ise *eğitim verisi,* bu verilerin oluşturduğu küme ise *eğitim kümesi* olarak adlandırılır [4]. Eğitim kümesinin sınamasının yapılmasında kullanılan verilere *test verisi*, bu verilerin oluşturduğu kümeye ise *test kümesi* adı verilir.

Makine öğrenme yöntemlerinden yaygın olarak kullanılanları gözetimli öğrenme ve gözetimsiz öğrenme olmasına rağmen, yarı gözetimli öğrenme ile destekleyici öğrenme de tercih edilmeye başlamıştır.

2.1. Gözetimli Öğrenme (Supervised Learning)

Öğrenme gerçekleştirecek sisteme, eğitim örnekleri olarak girdi verileri ve bu verilere karşılık gelen her bir çıktı değeri verilmektedir (Şekil.1). Çıktı değerleri verildiği için, her girdi verisinin hangi sınıfa ait olduğu ve toplam kaç ayrı sınıfın olduğu başlangıçta bellidir. Bu yöntemde kullanılan algoritma ile girdi ve çıktı verileri arasındaki ilişkiler ortaya çıkarılmaktadır. Bu çıkarımlar ise girdi verileri ile öğrenme gerçekleştiren sistemin oluşturduğu bir model aracılığı ile olur. Bu model de yeni bir veri gelmesi durumunda test edilebilir. Böylece, oluşturulan gözetimli öğrenme modelinin test başarısı kolayca ölçülebilir. Test işlemi de, elde sahip olunan verilerin büyük kısmının eğitim (genellikle % 70 civarıdır) amaçlı ayrılmasından sonra kalanının ise test verisi olarak kullanılması olabilir.



Şekil. 1. Gözetimli öğrenmede veritabanı genel yapısı

Gözetimli öğrenme modellerinde, öğrenerek yeni bir girdi verisine karşılık gelen sınıf değeri tahmin edilerek, sınıflandırma yapabilmek mümkündür. Sınıflandırma ve regresyon modelleri gözetimli öğrenme yöntemlerine örnek olarak verilebilmektedir.

2.1.1. Bayes Sınıflandırma Algoritmaları (Bayesian Classification Algorithms)

Bilimsel karar yöntemlerinden biri olan Bayes sınıflandırması, istatistik alanında gelişme gösteren bir yaklaşımdır. Bu yaklaşım, iki kaynağı en verimli şekilde birleştirmeye çalışmaktadır. Birinci kaynak, verinin içermiş olduğu objektif bilgidir. Diğeri ise, bir teori, gerçekliği kabul gören bir fikir, önsel bir bilgi veya birçok olasılığı olan sübjektif düşünce ya da bir durum hakkında kişinin kabul etme düzeyidir[5].

Günümüzde bilimsel öğrenme ve karar vermede önemli bir ilgi odağı olan yaklaşım; koşullu olasılık ve toplam olasılık temeline dayanan Bayes teoreminden gelmektedir.

Aynı örnek uzayı içerisinde ortak noktaları bulunan iki olay A ve B olsun. B olayının gerçekleşme olasılığı, eğer A olayının gerçekleşme olasılığına bağlı ise burada koşullu bir olasılıktan söz edilebilir. Yani B olayı, ancak A olayı bilindiğinde gerçekleşebilecektir. Söz konusu olasılık, sembollerle (2.1)'deki gibi gösterilir:

$$P(\mathrm{B}|\mathrm{A}) = \frac{\mathrm{P}(\mathrm{A},\mathrm{B})}{\mathrm{P}(\mathrm{A})} \qquad P(\mathrm{A}) \neq 0 \tag{2.1}$$

Burada;

$\mathrm{P}(\mathrm{A},\mathrm{B})$ $[P(\mathrm{A} \cap \mathrm{B})$ şeklinde de gösterilebilir$]$: A ve B olaylarının birlikte ortaya çıkma olasılığını,

$P(\mathrm{A})$: A olayının ortaya çıkma olasılığını,

$P(\mathrm{B}|\mathrm{A})$: A olayının gerçekleştiği bilindiğinde, B olayının ortaya çıkması olasılığını, yani B olayının A olayı için koşullu olasılığını göstermektedir.

Benzer şekilde,

$$\mathrm{P}(\mathrm{A}|\mathrm{B}) = \frac{\mathrm{P}(\mathrm{A},\mathrm{B})}{\mathrm{P}(\mathrm{B})} \tag{2.2}$$

yazılması mümkündür ($P(\mathrm{B})$: B olayının ortaya çıkma olasılığıdır). Bu durumda, her iki eşitliğin pay kısmında birleşik olasılık mevcuttur ve bu iki eşitlik birleştirildiğinde (2.3)'teki formül elde edilir:

$$P(\mathrm{B}|\mathrm{A}) = \frac{\mathrm{P}(\mathrm{A}|\mathrm{B}).\mathrm{P}(\mathrm{B})}{\mathrm{P}(\mathrm{A})} \qquad \mathrm{P}(\mathrm{A}) \neq 0 \tag{2.3}$$

Bayes sınıflandırması, Bayes teoremini temel alan bir sınıflandırma algoritmasıdır. Bu sınıflandırma, örnek ön bilgiler toplandıktan sonra, karara ilişkin sınıf bilgileri hakkında olasılıksal yorumlar yapılmasına olanak sağlamaktadır. Örneğin, "Bir F olayı bilgisi için, içinde birden fazla sınıf barındıran C olayının hangi sınıfı gerçekleşebilecektir?" sorusuna yanıt bulunmak istensin. Burada, sınıfı temsil eden her i için, $P(\mathrm{C_i}|\mathrm{F})$ olasılığının bulunması gerekmektedir. Bunun için Bayes teoreminden yararlanılırsa, genel formda;

$$P(\mathrm{C}|\mathrm{F}) = \frac{\mathrm{P}(\mathrm{F}|\mathrm{C}).\mathrm{P}(\mathrm{C})}{\mathrm{P}(\mathrm{F})} \tag{2.4}$$

yazılabilir. Böylece sınıflandırılacak olan bir veri için, hangi sınıflara hangi olasılık dâhilinde katılabileceği fikir edinilir. Bayes sınıflandırma, Naive Bayes ve Bayes ağı algoritması olmak üzere ikiye ayrılır.

2.1.1.1. Naive Bayes Algoritması (Naive Bayes Algorithm)

Naive Bayes algoritması, Bayes teoremine dayanan ve matematiksel istatistik tekniklerin kullanıldığı bir sınıflandırma yöntemidir. Amacı, birbirinden bağımsız ve farklı niteliklerin belirlediği sonuçlar ile bu niteliklere ait yeni girdi verilerini kullanarak, bu verinin hedef (karar) kümesinde hangi sınıfa ait olduğunu belirlemektir. Karar vermede kullanılan kriter ise hedef kümede yer alan sınıfların, sınıflandırılacak veriye göre koşullu olasılıklarının değeridir. En yüksek koşullu olasılığı sağlayan sınıf, yeni verinin sınıfı olarak belirlenmektedir.

Naive Bayes sınıflandırma algoritmasında, girdi verisinde birbirinden bağımsız özellikleri olan her bir parametrenin sonuca olan etkisi olasılıksal olarak hesaplanmaktadır. Bundan dolayı, veri sınıflandırması aslında sınırları koşullu olasılık değerleri ile belirli olan bir tahmindir.

F=$\{F_1, F_2, \ldots., F_m\}$ birbirinden bağımsız m adet niteliğe ait sınıf değerlerinden oluşan veri örneği ve $C_1, C_2, \ldots., C_n$ bağımlı sınıf değişkeni olarak düşünülürse, Bayes teoremi yardımıyla;

$$P(C_i|F) = \frac{P(F|C_i)\,P(C_i)}{P(F)} \tag{2.5}$$

elde edilir. Burada her $P(C_i|F)$ için $P(F)$ sabit kalacağından ve sınıf seçiminde maksimum benzerlik dikkate alınarak büyüklük karşılaştırması yapılacağından dolayı paydayı, yani $P(F)$ değerini, hesaplara katmaya gerek kalmamaktadır [6]. Çünkü F verisi sabit bir veri olup, sınıfı belirlenmesi istenen F verisi için (2.5)' deki pay değerinde en büyük değeri sağlayan C_i sınıfı seçilerek, veri örneği sınıfı belirlenmiş olur.

$$\begin{aligned} P(C_1|F) \overset{?}{>} P(C_2|F) \overset{?}{>} \ldots.. \overset{?}{>} P(C_n|F) \quad = \\ P(F|C_1)P(C_1) \overset{?}{>} P(F|C_2)P(C_2) \overset{?}{>} \ldots.. \overset{?}{>} P(F|C_n)P(C_n) \end{aligned} \tag{2.6}$$

Pay değeri olan $P(F|C_i)P(C_i)$ değeri genişletilecek olursa;

$$P(F|C_i)P(C_i) = P(C_i).\ \prod_{k=1}^{n} P(F_k|C_i) = P(C_i).\,P(F_1|C_i).\,P(F_2|C_i) \ldots.P(F_n|C_i\ ) \tag{2.7}$$

olarak elde edilir.

F verilerine ait her C_i sınıfı için hesaplanan bu değerlerden büyük olan *i* sınıfı seçileceğinden dolayı Naive Bayes algoritması için sonuç ifadesi;

$$arg\ max_c \{\ P(F|C_i)P(C_i)\ \} \quad (2.8)$$

olur. Bu ifade "*maximum a posteriori classification (MAP)*" olarak bilinir[203].

Naive Bayes algoritmasının uygulanacağı bir problem ele alınsın. Problem, küçük ölçekli bir sigorta şirketinin farklı kriterlere sahip müşterileri için bazı kasko sigortası bedellerini belirlemek istemesidir. Bu bedeller, sigorta şirketini belirli bir mali dengede tutabilmek amacıyla ne çok yüksek ne de çok düşük olmamalıdır. Bunun nedeni sigorta bedelleri çok yüksek olduğunda, şirketin müşteri kaybına uğraması, çok düşük olduğunda ise müşterilerin, dolayısı ile araç hasar bedellerinin fazla olması nedeniyle sigorta bedellerinin bu giderleri karşılayamamasıdır. Bu nedenle de müşterilerine ait yaş/yerleşim/araç üretim bilgileri kullanılarak daha önce başka bir sigorta şirketinden hizmet almış olan veya ilk defa sigorta yaptırmak isteyen farklı kriterlere sahip bir müşteriye ait sigorta bedelinin ne olacağı belirlenmek istenmektedir. Sigorta bedeli, trafik kazası geçirme riskine göre artmaktadır.

Tablo.1.'deki istatistikî bilgiler bir sigorta şirketinde bulunmaktadır. Söz konusu sigorta şirketinden hizmet almak isteyen, *şehir*de araç kullanan, *genç* yeni bir müşterinin aracı *yerli* olduğu bilindiğinde, tablodaki bilgilere göre bu müşteri için trafik kazası riskinin çok olup olmadığı öğrenilmek istenmektedir.

Tablo.1. Sigorta şirketi müşterilerinin profili

Müşteri No	Yaş	Yerleşim	Araç Üretim	Trafik Kazası Riski
001	Genç	Büyükşehir	Yerli	Çok
002	Genç	Büyükşehir	Yerli	Az
003	Genç	Büyükşehir	Yerli	Çok
004	Yaşlı	Büyükşehir	Yerli	Az
005	Yaşlı	Büyükşehir	İthal	Çok
006	Yaşlı	Şehir	İthal	Az
007	Yaşlı	Şehir	İthal	Çok
008	Yaşlı	Şehir	Yerli	Az
009	Genç	Şehir	İthal	Az
010	Genç	Büyükşehir	İthal	Çok

Sınıflandırılması istenen veri ;

$F = \{F_1 = (Yaş: Genç),\ F_2 = (Yerleşim: Şehir),\ F_3 = (Araç\ Üretim: Yerli)\ \}$ dir.

C_i bağımlı değişken sınıfında *Çok* ve *Az* olmak üzere iki farklı veri olduğundan C sınıfı ikiye ayrılır. $C_1 = Çok$, $C_2 = Az$ olarak seçilebilir. Problemdeki mevcut verilerle hesaplanan koşullu olasılık değerleri Tablo.2.'de verilmiştir.

Tablo.2. Müşteri profilleri veritabanından çıkarılan koşullu olasılık tablosu

C	F=Genç	F=Şehir	F=Yerli
C=C_1=Çok , $P(C_1) = \frac{5}{10} = \frac{1}{2}$	$P(F_1\|C_1) = \frac{3}{5}$	$P(F_2\|C_1) = \frac{1}{5}$	$P(F_3\|C_1) = \frac{2}{5}$
C=C_2=Az , $P(C_2) = \frac{5}{10} = \frac{1}{2}$	$P(F_1\|C_2) = \frac{2}{5}$	$P(F_2\|C_2) = \frac{3}{5}$	$P(F_3\|C_2) = \frac{3}{5}$

(2.7) deki ifadeyi her C sınıfı için hesaplamak gerekmektedir. Dolayısı ile $P(F|C_1)P(C_1)$ ve $P(F|C_2)P(C_2)$ değerleri hesaplanır.

$C = C_1 = Çok$ için (2.7) deki ifadeden

$P(F|C_1)P(C_1) = P(C_1).P(F_1|C_1).P(F_2|C_1)P(F_3|C_1)$

$P(F|C_1)P(C_1) = \frac{1}{2}.\frac{3}{5}.\frac{1}{5}.\frac{2}{5}$ = 0,024 olarak bulunur.

$C = C_2 = Az$ için (2.7) deki ifadeden

$P(F|C_2)P(C_2) = P(C_2).P(F_1|C_2).P(F_2|C_2)P(F_3|C_2)$

$P(F|C_2)P(C_2) = \frac{1}{2}.\frac{2}{5}.\frac{3}{5}.\frac{3}{5}$ = 0,072 olarak bulunur.

$$arg\,max_c\{\,0{,}024\,;\,0{,}072\} = 0{,}072$$

olduğundan dolayı verilen örnek için trafik kazası riskine ait sınıfta "*Az*" durumu seçilir. Bu sonuç farklı kriterler söz konusu olduğunda da, sigorta şirketine müşterisinin aracına ait sigorta bedelinin en doğrusunu belirlemede büyük kolaylık sağlayacaktır. Şöyle ki, elde edilen bu bilgi ile trafik kazası riski "*Az*" olduğundan, bu nitelikteki müşteriye ait sigorta bedeli düşük tutulacaktır.

Naive Bayes algoritması, mevcut verileri kullanarak her yeni bir duruma ait sonuç için olasılıksal yöntemlerle tahmin yaptığı için, mevcut verilerin fazlalılığı sonucun doğru tahmin edilmesi ihtimalini arttırmaktadır.

2.1.1.2. Bayes Ağı Algoritması (Bayesian Network Algorithm)

Bayes ağı, çeşitli olaylarda değişkenler arasındaki sebepleri ve etkileri göstermek için Bayes Teoreminin kullanıldığı, yönlü, çevrimsiz grafiklerden oluşan modeldir[7]. Örneğin hava durumu, alarm cihazının çalışması, hastalık nedenleri araştırmasında kullanılabilmektedir. Bir kanser hastalığına dair araştırmada, sigara kullanımı, genetik, radyasyon, beslenme bozuklukları gibi etkilerin kanser hastalığının nedenlerine etkilerini bir arada olasılıklar dâhilinde incelemek mümkün olmaktadır.

Bayes ağı, değişkenler arasındaki ilişki hakkındaki geçmiş bilgisini ne kadar iyi bilirse o kadar iyi olasılıksal ilişkiler kurar. Ancak, Bayes ağı koşullu olasılık tablolarındaki bilinen verilerin tam olarak doğru olmadığı, yani her zaman geçerli olmadığı durumlarda da doğru sonuçlar üretebilmektedir. Bu nedenle Bayes ağları, diğer yapay zekâ algoritmalarına alternatif duruma gelmektedirler.

Bayes ağı otomatikleşmiş karar mekanizmasına gerek duyulan çok çeşitli alanlarda artarak kullanılmaktadır.

Bir Bayes Ağı;

- Değişkenlere ait bir dizi,
- Değişkenleri birbirine bağlayan grafiksel yapı ve
- Bir dizi koşullu olasılıktan oluşur.

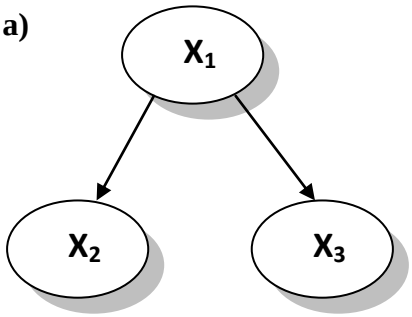
Bayes ağı, uygulamalarda grafiksel olarak tasarlanırken her bir değişken *düğüm*lerle gösterilir. Her düğüm, durumlara ya da her değişken için olası değerler kümesine sahiptir. Düğümler, etkileşimin yönünü tanımlayan bir yay ile nedenselliği göstermek üzere birbirine bağlanır. Bu yaylar *kenar* olarak tanımlanır ve bir düğümün sebep-sonuç ilişkisi içinde, diğer düğümü nasıl etkilediğini ifade eder ve koşullu bağıntılara karşılık gelirler.

Bayes ağında (2.9.)'da verilen zincir kuralı geçerlidir. Bu kural, bir Bayes ağı için bileşik olasılık dağılımına ait fonksiyonun bulunmasında kullanılmaktadır.

$$P(X_1, X_2, \dots., X_n) = P(X_n|X_1, \dots, X_{n-1})P(X_1, .., X_{n-1})$$
$$= P(X_n|X_1, \dots, X_{n-1})P(X_{n-1}|X_1, \dots, X_{n-2})P(X_1, .., X_{n-2}) = \prod_{i=1}^{n} P(X_i|X_1, .., X_{i-1}) \quad (2.9)$$

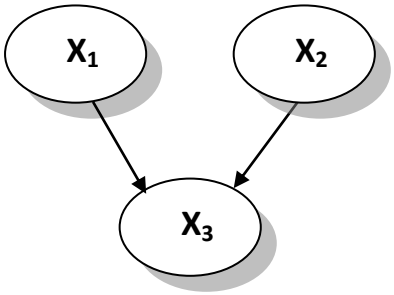
Şekil.2.'de farklı durumlara ait tasarlanmış, örnek Bayes ağları ve bileşik olasılık dağılımları verilmektedir.

a)



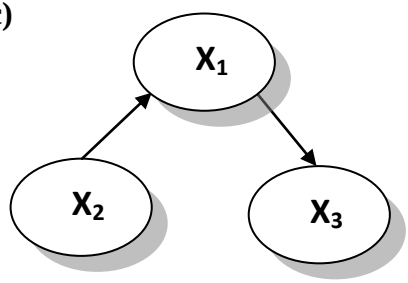
$P(X_1, X_2, X_3)$=$P(X_2|X_1).P(X_3|X_1).P(X_1)$

b)



$P(X_1, X_2, X_3)$=$P(X_3|X_1, X_2).P(X_1).P(X_2)$

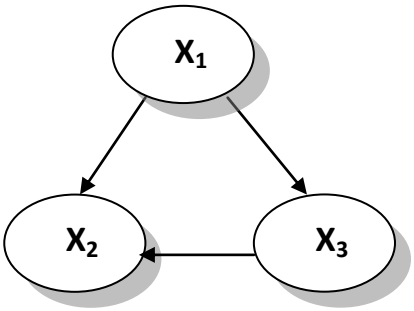
c)



$P(X_1, X_2, X_3)$=$P(X_3|X_1).P(X_2, X_1)$

$P(X_1, X_2, X_3)$=$P(X_3|X_1).P(X_1|X_2).P(X_2)$

d)



$P(X_1, X_2, X_3)$=$P(X_3, X_2|X_1).P(X_1)$

$P(X_1, X_2, X_3)$=$P(X_2|X_3, X_1).P(X_3|X_1).P(X_1)$

Şekil.2. Farklı yapıdaki basit Bayes ağları ve birleşik olasılık dağılımları

Şekil.2.a.'da X_1'den X_2'ye bir kenarın var olduğu görülmektedir. Bu durumda X_1, X_2'nin *ebeveyn*i olur ve X_1'in X_2 üzerinde doğrudan bir etkisi vardır. X_2'nin bir ebeveyni olduğundan, kendisinin yerel olasılık dağılımı da *koşullu* olur. i=1, 2, 3,…..n olmak üzere her X_i'ye ait ebeveynler kümesi *ebeveyn(X_i)* olarak gösterilirse Bayes ağları için genelleştirilebilecek bileşik olasılık dağılımı ifadesi (2.10)'daki hali alır:

$$P(X_1 = x_1, \dots., X_n = x_n) = \prod_{i=1}^{n} P(X_i = x_i \mid ebeveyn(X_i))$$

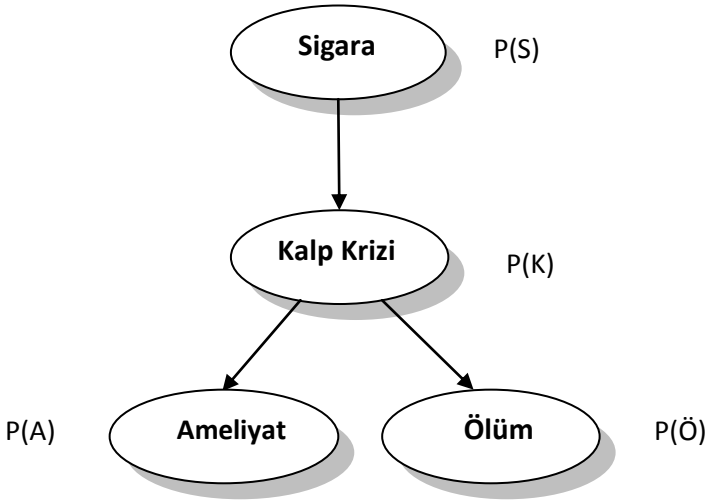
(2.10)

Herhangi bir koşullu olasılık ise (2.11)'deki gibi hesaplanır:

$$P(X_k|X_m) = \frac{P(X_k, X_m)}{P(X_m)} = \frac{\sum_{\substack{X_k \text{ ve } X_m' \text{ i içeren} \\ X \text{ girdileri}}} P(X)}{\sum_{\substack{X_m' \text{ i içeren} \\ X \text{ girdileri}}} P(X)}$$

(2.11)

Bu algoritmaya örnek olarak, Şekil.3.'de kalp krizi sonunda ameliyat geçirmiş ya da hayatını kaybetmiş olan kişilere ait olasılıkları belirlenmiş olaylarda, sigara kullanımının etkisini incelemek üzere oluşturulan basit bir Bayes ağı verilmektedir. Bu olaylara ilişkin olasılıklar da Tablo.3.'de verilmektedir.



Şekil.3. Sigara ve kalp krizine ilişkin basit bir Bayes ağı

Tablo.3.'deki veriler doğrultusunda, ameliyat ve ölümü gerçekleşen kişilerin sigara içme olasılığı öğrenilmek istenmektedir. Bir başka deyişle P(S:1/A:1,Ö:1) değeri bulunmalıdır. Olasılık tablolarında yer alan ameliyat, ölüm, kalp krizi gerçekleşmesi ve sigara kullanılması durumları 1 olarak, tam tersi ise 0 olarak gösterilmiştir.

Tablo.3. Değişkenlere ait olasılık değerleri

a)

P(S)	**Sigara**	
	0	**1**
	0,59	0,41

b)

P(K/S)		**Kalp Krizi**	
		0	**1**
Sigara	**0**	0,91	0,09
	1	0,69	0,31

c)

P(A/K)		**Ameliyat**	
		0	**1**
Kalp Krizi	**0**	0,72	0,28
	1	0,45	0,55

d)

P(Ö/K)		**Ölüm**	
		0	**1**
Kalp Krizi	**0**	0,19	0,81
	1	0,42	0,58

Şekil.3.'deki Bayes ağındaki değişkenler arasındaki ilişki şu şekilde yorumlanabilir: Sigara değişkeni, kalp krizi değişkeninin ebeveynidir. Kalp krizi değişkeni ise hem ameliyat hem de ölüm değişkeninin ebeveynidir. Ölüm, ameliyat ve sigara değişkenleri ise birbirinden bağımsızdır.

Koşullu bir olasılık hesaplamak için öncelikle birleşik olasılık dağılımı hesaplanmalıdır. Yukarıdaki Bayes ağı için bu dağılım aşağıdaki gibi olur:

$$P(S, K, A, \text{Ö}) = P(A|K).P(\text{Ö}|K).P(K|S).P(S)$$

$$P(S: 1 \mid A: 1, \text{Ö}: 1) = \frac{P(S: 1, K, A: 1, \text{Ö}: 1)}{P(S, K, A: 1, \text{Ö}: 1)}$$

P(S:1,K,A:1,Ö:1) = P(A:1 | K:0). P(Ö:1 | K:0). P(K:0 | S:1). P(S:1) + P(A:1 | K:1). P(Ö:1 | K:1). P(K:1 | S:1). P(S:1) = 0,28.0,81.0,69.0,41+ 0,55.0,58.0,31.0,41=0,1046

P(S,K,A:1,Ö:1) = P(A:1 | K:0). P(Ö:1 | K:0). P(K:0 | S:1). P(S:1) +P(A:1 | K:1).P(Ö:1 | K:1).P(K:1 | S:1).P(S:1) + P(A:1 | K:0).P(Ö:1 | K:0).P(K:0 | S:0). P(S:0) + P(A:1 | K:1). P(Ö:1 | K:1). P(K:1 | S:0). P(S:0)= 0,28.0,81.0,69.0,41+ 0,55.0,58.0,31.0,41 +0,28.0,81.0,91.0,59 + 0,55.0,58.0,09.0,59= 0,0641+0,0405+0,1217+0,0169 = 0,2432

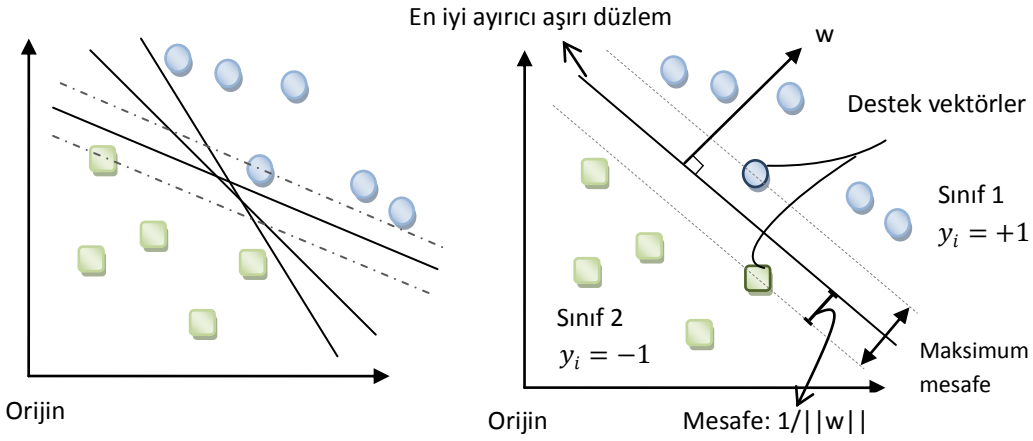
$$P(S: 1 \mid A: 1, \text{Ö}: 1) = \frac{P(S: 1, K, A: 1, \text{Ö}: 1)}{P(S, K, A: 1, \text{Ö}: 1)} = \frac{0{,}1046}{0{,}2432} = 0{,}43$$

Ameliyat olup, ölen kişilerin sigara içme olasılığı % 43 olarak elde edilmiş olunur.

2.1.2.Destek Vektör Makineleri Algoritması (Support Vector Machines Algorithm)

Destek Vektör Makineleri temeli 1960'lara kadar dayanan ve V.N. Vapnik tarafından geliştirilen bir istatistiksel öğrenme teorisi alanında ortaya çıkmış bir öğrenme yöntemidir[8]. Destek Vektör Makineleri algoritması 1995 yılında, sınıflandırma ve doğrusal olmayan fonksiyon yaklaşımı problemlerinin çözümü için V.N. Vapnik tarafından önerilmiş olup, günümüzde el yazısı tanıma, ses tanıma, meme kanseri tahmini, biyoinformatik ve uzaysal veri analizi gibi birçok alanda sıkça kullanılmaktadır[9].

Destek Vektör Makineleri, doğrusal olmayan örnek uzayını, örneklerin doğrusal olarak ayrılabileceği bir yüksek boyuta aktararak, farklı örnekler arasındaki maksimum sınırın bulunması esasına dayanır. Bu alanda karşılaşılan problemlerin büyük çoğunluğu, birçok farklı bileşenden oluşan problemlerdir ve doğrusal olarak ayrılmış bir yapı halinde değildirler. Doğrusal olarak ayrılmış olan veriler arasında maksimum sınırın direkt olarak bulunması basit olmasına rağmen, doğrusal olarak ayrılamayan veriler öncelikle doğrusal olarak ayrılabilecekleri farklı bir uzaya aktarılmaktadır.



a) **b)**

Şekil.4. Doğrusal yapılabilen ayrım a) İki sınıflı verileri ayıran bazı aşırı düzlemler b) En iyi ayırıcı aşırı düzlem ve sınırları

Doğrusal olarak ayrılabilme durumunda, bu iki değerli veriler bir aşırı düzlem ile ayrılabilmektedir. Sınıflandırmayı sağlayan bu düzleme *ayırıcı aşırı düzlem* adı verilir. Şekil.4.'de bu duruma ait bir örnek verilmiştir. Farklı iki sınıfta yer alan verileri birbirinden ayırmak için birçok ayırıcı aşırı düzlem kullanılabilir ancak öğrenme hatasının asgari olması gerektiği için bu düzlemlerden uygun olanı seçilmelidir. Bu nedenle amaç, veri kümesindeki en uç verilerin arasındaki mesafenin maksimum olduğu optimum bir aşırı düzlemle ayırıp, aynı sınıfa ait bütün verileri düzlemin aynı tarafında bırakmaktır. Düzlemin optimum olmasıyla, genelleme yeteneği de maksimum düzeyde olmaktadır. Burada sınıflara ait düzlemler üzerinde yer alan en uçtaki verilere *destek vektör* adı verilir. Ayırıcı aşırı düzlem, bu düzlemleri ortalamaktadır [10].

Doğrusal olarak ayrılabilme durumunda $x_i \in R^d$ özellikler vektörü ve her biri y_i={-1,+1} ile gösterilen sınıflardan birine ait olmak üzere n adet eğitim verisi $\{x_i, y_i\}$ şeklinde ifade edilsin. Bu durumda, ayırıcı aşırı düzlem için fonksiyon aşağıdaki gibidir:

$$\text{f(x)} = w^T.x + b = \sum_{i=1}^{n} w_i x_i + b \tag{2.12}$$

Burada w, aşırı düzlemin normalini, b sabiti ise sapma değerini ifade eder.

|b|/||w|| aşırı düzlemin orijine olan uzaklığı, x aşırı düzlem üzerinde olan herhangi bir nokta olmak üzere aşırı düzlem üzerindeki noktalar cinsinden $w^T.x +$

$b = 0$ koşulunu sağlamaktadır. Dolayısı ile ayırıcı aşırı düzlem verileri iki sınıfa böleceğinden

$$\text{Sınıf 1 :} \quad f(x) > 0 \text{ için } y_i = +1;$$

$$\text{Sınıf 2 :} \quad f(x) < 0 \text{ için } y_i = -1 \tag{2.13}$$

yazılması mümkündür.

Ayırıcı aşırı düzlem, y_i={-1,+1} sınıflarına ait olan veri kümesine optimal olarak eşit uzaklıkta seçilmektedir. Bu ayırıcı aşırı düzlem, verileri iki kısma ayırmaktadır. Bu düzlemin üst tarafında kalan verilerin y_i= +1 sınıfına, altında kalan verilerin ise y_i= -1 ait olduğu kabul edilirse;

$$y_i = +1 \text{ için } w^T.x + b \geq +1$$

$$y_i = -1 \text{ için } w^T.x + b \leq -1 \tag{2.14}$$

yazılabilir. (2.14)'teki iki eşitlik birleştirildiğinde aşağıdaki ifade elde edilmektedir:

$$y_i(w^T.x + b) \geq 1 \qquad (i=1,2,\ldots.n) \tag{2.15}$$

Eğitim kümesi verileri, $\{x_i, y_i\}$ formunda girdi olarak bilinmektedir. Destek vektörler arasındaki mesafenin maksimum olduğu, en uygun ayırıcı aşırı düzlemin belirlenebilmesi için yukarıdaki eşitlikte w ve b değerlerinin bulunması gerekmektedir.

Herhangi bir x_i veri noktasının aşırı düzleme olan uzaklığını aşağıdaki gibidir:

$$d_i = \frac{|w^T.x + b|}{||w||} \tag{2.16}$$

Yukarıdaki iki denklem birleştirildiğinde aşağıdaki ifade elde edilir:

$$y_i d_i \geq \frac{1}{||w||} \tag{2.17}$$

Optimal bir ayırıcı aşırı düzlemin destek vektörler ile arasındaki mesafesinin maksimum olmasının gerekliliğine değinilmişti. Bundan dolayı ayırıcı aşırı düzlemle en yakın x_i noktası arasındaki mesafenin maksimum olması gerekmektedir. (2.17)'deki $\frac{1}{||w||}$ ifadesi (w,b) aşırı düzlemi ile bir x_i veri noktası arasındaki uzaklığın bir alt sınırıdır. Eğer veri noktaları kümesi içinde, bu alt sınırı eşitlikle sağlayan bir x_i noktası bulunursa bu, ayırıcı düzleme en yakın noktanın bulunduğu anlamına gelir. Bu durumda, en yakın x_i noktası ile ayırıcı düzlem arasındaki

mesafenin maksimum olabileceği değer $\frac{1}{||w||}$'e karşılık geldiğinden dolayı $||w||$ ifadesinin minimize edilmesi gerekmektedir. Bu da ikinci dereceden optimizasyon teknikleri ile yapılabilmektedir.

$$\left|\left|w\right|\right|^2 = \sum_{i=1}^{n} w_i^2 = ww = w^T w \quad \text{(Öklid Formu)}$$

(2.18)

İkinci dereceye yükseltildikten sonra (2.18)'deki $\frac{w^T W}{2}$ ifadesinin minimize edilmesi gerekir. Burada ½ çarpanı işlemleri kolaylaştırmak için eklenir. Bu ifadede dual optimizasyonu çözmek için *Lagrange fonksiyonları* ve *Karush-Kuhn-Tucker koşulları*ndan yararlanılır[6].

Lagrange fonksiyonu (2.19)'da verilmiştir:

$$L(w, b, a) = \frac{1}{2}(w^T . w) - \sum_{i=1}^{n} a_i [y_i (w^T . x_i \ + b) - 1]$$

(2.19)

Bu fonksiyonun sırasıyla w ve b' ye göre kısmi türevleri alınırsa aşağıdaki ifadeler elde edilir:

$$\sum_{i=1}^{n} y_i a_i x_i \ = w$$

(2.20)

$$\sum_{i=0}^{n} y_i a_i = 0$$

(2.21)

Karush-Kuhn-Tucker koşulları olarak bilinen (2.20) ve (2.21)'deki eşitlikleri kullanılarak Lagrange açılımı yapılırsa optimizasyon problemi (2.22)'deki hali ile maksimize edilir:

$$L(a) = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j a_i a_j x_i^T \ x_j$$

(2.22)

(2.22)'deki $a_{1,} a_2, \ldots \ldots a_n$ katsayıları Lagrange çarpanları olarak isimlendirilir ve bu denklemin bu çarpanlara göre kısmi türevlerinin alınıp sıfıra eşitlenmesi sonucunda elde edilirler. Lagrange çarpanları elde edildikten sonra w^* değeri (2.23)'deki denklem aracılığı gibi hesaplanabilir.

$$w^* = \sum_{x_i \epsilon\ SV} y_i a_i x_i \quad (i = 1,2,\dots n)$$

(2.23)

Böylece bulunan w^* değeri ile $||w||$ ifadesi minimize edilmiş olur. b sabiti ise (2.24)'deki gibi hesaplanır:

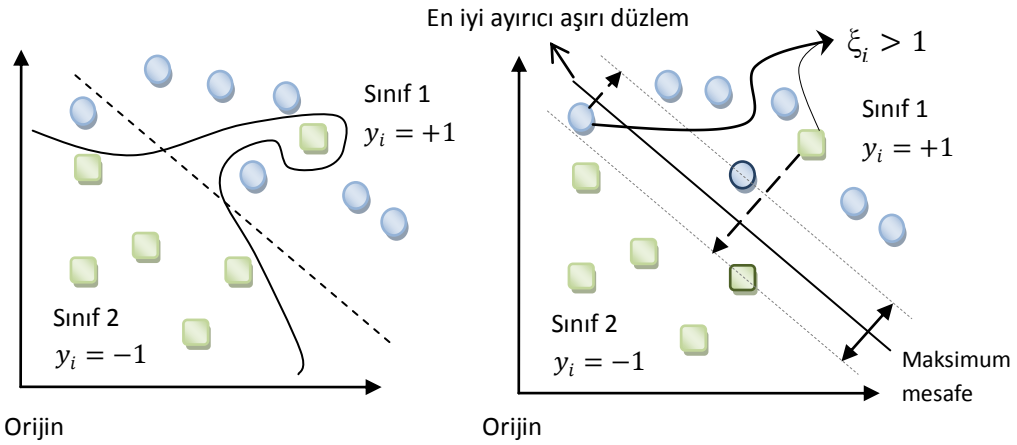
$$b = \frac{1}{n_{sv}} . \left(\sum_{s=1}^{n_{sv}} (\frac{1}{y_s} - x_s^T . w^*) \right) \quad (s = 1,2,..n_{sv})$$

(2.24)

Burada *sv* destek vektörleri, n_{sv} ise destek vektörlerin sayısını göstermektedir[203].

Bu değerlerin kullanılması ile x, sınıfı belirli olmayan bir girdi verisi olmak üzere sınıflandırma problemi, w^* .x +b denkleminin işaretini bulmaya indirgenir. Sonuç olarak sınıflandırma için kullanılacak karar fonksiyonu **sgn(*f(x)*)=*sgn*(w^* *.x+b)** olur.

Veri kümelerinin doğrusal olarak sınıflandırılması her zaman mümkün olamamaktadır. Şekil.5.a.'da buna benzer bir örnek verilmiştir. Bu durumda farklı sınıflara ait verilerin ayrımı, minimum hata ile doğrusal olarak yapılmaya çalışılır. Bu da, aynı sınıfa ait verilerin mümkün olduğunca en fazlasını doğrusal ayırıcı aşırı düzlemin aynı tarafında bırakarak sağlanmaktadır. Şekil.5.b.'de normalde doğrusal ayrımı yapılamayan verilerin belirli bir hata oranı ile doğrusal olarak yapıldığı ayrım gösterilmiştir.



Şekil.5. a)Verilerin doğrusal ayrılamama durumu b) Verilerin gevşek değişkenler ile birbirinden ayrılması

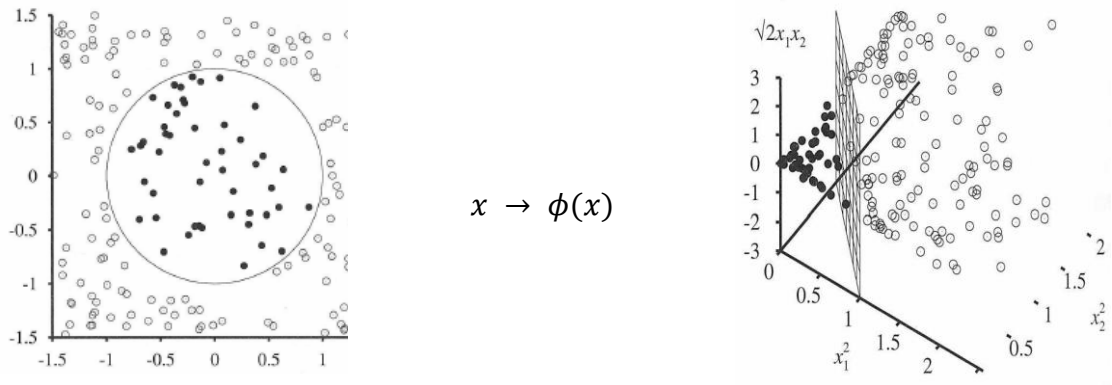
Verilerin sınıflandırılması doğrusal ayrım adımlarına gevşek değişkenlerin (slack variables) eklenmesi yolu ile yapılır. Gevşek değişkenler ξ ile gösterilmektedir.

(2.15)'deki denkleme bu değişkenler eklenir ve (2.25)'deki hale getirilir:

$$y_i(w^T.x + b) \geq 1\text{- } \xi_i \quad (i=1,2,\ldots.n) , \qquad \xi_i \geq 0 \tag{2.25}$$

ξ_i=0 doğru sınıflandırılmış veriler için, $1 > \xi_i > 0$ koşulu doğru sınıflandırılmış ancak ayırıcı aşırı düzlemin ortalamış olduğu maksimal bölge içinde kalan veriler için, $\xi_i > 1$ koşulu ise ayırıcı aşırı düzlemin diğer tarafında kalan yani yanlış sınıflandırılmış veriler için geçerlidir.

Verilerin doğrusal olarak sınıflandırılmasının mümkün olmadığı durumlarda, verileri bir üst boyuta taşıdıktan sonra doğrusal olarak ayırmak mümkün olabilmektedir. Bu, doğrusal olmayan destek vektör makineleri ile sağlanmaktadır. Bu amaçla *çekirdek fonksiyonları*ndan geçirilen giriş verileri, üst boyutlu uzaylara düşürülür ve sınıflandırma burada yapılır.



a.) 2 boyutta doğrusal olarak ayrılamayan veriler

b.) 3 boyuta taşındıktan sonra doğrusal olarak ayrılabilen veriler

Şekil.6. Doğrusal olarak ayrılamayan verilerin üst boyutta ayrılması

Doğrusal olarak ayrılamayan verilerde (Şekil.6.a.), ϕ çekirdek fonksiyonu kullanılarak n boyutlu bir veri kümesini m>n olmak üzere m boyutlu yeni bir veri kümesine dönüştürülerek, yüksek boyutta sınıflandırma işlemi gerçekleştirilir. Şekil.6.b.'de, 2 boyutlu uzayda doğrusal ayrılamayan veri kümelerinin 3 boyutlu

uzaya taşındıktan sonra doğrusal ayrılabileceği gösterilmiştir. Çekirdek fonksiyonu bu gibi problemlere çözüm getirdiğinden, destek vektör makineleri algoritmalarında önemli bir rol oynar.

Lagrange fonksiyonu, çekirdek fonksiyonu kullanılarak dönüşümü yapıldıktan sonra (2.26)'daki hali alır:

$$x \rightarrow \phi(x)$$

$$L(a) = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j a_i a_j \ \phi(x_i)^T \phi(x_j) \tag{2.26}$$

Buradaki $\phi(x_i)^T \phi(x_j)$ çarpımı $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ çekirdek fonksiyonu olarak tanımlanır ve (2.27)'deki gibi de gösterilebilir.

$$L(a) = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j a_i a_j \ K(x_i, x_j) \tag{2.27}$$

Aşağıda bazı çekirdek fonksiyonları görülmektedir:

- *Doğrusal Çekirdek Fonksiyonu :* $K(x_i, x_j) = x_i^T x_j$

- *Polinom Çekirdek Fonksiyonu :* $K(x_i, x_j, c, p) = (c + x_i^T x_j)^p$

Veriler üst boyuta taşındıktan sonraki durumda sınıflandırma için karar fonksiyonu ise **sgn(*f(x)*)=*sgn(w. ϕ(x)+b)*** olarak yazılabilir.

Destek Vektör Makineleri temelde iki sınıfa ait verileri sınıflandırmak için tasarlanmış olsalar da, ikiden fazla sınıfa ait verileri sınıflandırmak içinde kullanılabilirler. Çok sınıflı Destek Vektör Makineleri adı verilen bu sınıflandırıcıların kullandığı yaklaşımlardan en önemli ikisi, bire-bir ve bire-tüm yöntemidir.Bire-bir yönteminde, her sınıfa ait veri kümesi, diğer sınıflara ait veri kümeleri ile ayrı ayrı eğitilir ve n adet farklı sınıf olması durumunda $\frac{n(n-1)}{2}$ tane ikili sınıflandırıcı oluşturulur. Bire-tüm yönteminde ise her sınıfa ait veri kümesi, diğer veriler tek bir sınıfa aitmiş gibi kabul edilerek eğitilir ve n adet farklı sınıf olması durumunda n adet ikili sınıflandırıcı oluşturulur. Her iki yöntemde eğitim işlemleri sonucu test aşamasında gelen örneğin hangi sınıfa ait olduğu, bütün eğitim işlemleri sonucu bulunan destek vektörler kıyaslanarak bulunur. Bu nedenle Destek Vektör Makineleri,

çoklu sınıflandırma içeren problemlerini, iki sınıfa ait sınıflandırma problemlerinin toplamı olarak ele almaktadır [11].

Destek Vektör Makineleri algoritması örneği için Tablo.4.'de koordinatları ve sınıfı verilmiş veriler bulunmaktadır. Verileri birbirinden ayıran fonksiyonu ve $x_4 = (3, -1)$ olan yeni bir verinin hangi sınıfa ait olduğu bulunmak istenmektedir.

Tablo.4. Koordinat sisteminde bulunan veriler

Veri	Koordinat	Sınıf
$\boldsymbol{x_1}$	(3,1)	+
$\boldsymbol{x_2}$	(1,1)	+
$\boldsymbol{x_3}$	(2,3)	-

Destek vektör makinelerinde sınıflandırma için kullanılan karar fonksiyonu aşağıdaki biçimdedir:

$$sgn(f(x))=sgn(w^* .x+b)$$

Veriler, 2 sınıftan (+ ve -) oluşmaktadır. Öncelikle verilerin y_i={-1,+1} değeri karşılığı seçilerek belirlenir.

Sınıf (+) $\Rightarrow y_1 = +1$

Sınıf (+) $\Rightarrow y_2 = +1$

Sınıf (-) $\Rightarrow y_3 = -1$ olduğu kabul edilsin.

Eğitim kümesine ait veriler, matrisel formda aşağıdaki şekilde gösterilir:

$$\boldsymbol{x_1} = \begin{bmatrix} \mathbf{3} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{x_2} = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{x_3} = \begin{bmatrix} \mathbf{2} \\ \mathbf{3} \end{bmatrix}$$

Fonksiyonun bulunabilmesi için ayırıcı aşırı düzlemin normali (w) ve b sabitinin bulunması gerekmektedir. Bunun için önce Lagrange fonksiyonu hesaplanır:

$$L(a) = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j a_i a_j x_i^T x_j$$

n=3 adet veri için x ve y değerleri yerine konularsa denklem aşağıdaki hali alır:

$$L(a) = a_1 + a_2 + a_3 - \frac{1}{2}(10{a_1}^2 + 2{a_2}^2 + 13{a_3}^2 + 8a_1a_2 - 10a_2a_3 - 18a_1a_3)$$

$$\sum_{i=0}^{n} y_i a_i = 0$$

eşitliği de kullanılarak

$y_1 a_1 + y_2 a_2 + y_3 a_3 = 1.a_1 + 1.a_2 + (-1).a_3 = 0$

$a_3 = a_1 + a_2$ elde edilir. Bu ifade $L(a)$'da yerine konulursa;

$L(a) = 2a_1 + 2a_2 - \frac{5}{2}{a_1}^2 - \frac{5}{2}{a_2}^2 - 3a_1 a_2$ elde edilir. Bu denklemin a_1 ve a_2' ye göre kısmi türevleri alınıp sıfıra eşitlemesi ile $a_1 = \frac{1}{4}$, $a_2 = \frac{1}{4}$ bulunur.

$$a_3 = a_1 + a_2 = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$$

Lagrange çarpanlarından hiçbiri sıfıra eşit çıkmadığı için bu çarpanlara bağlı x verilerinin hepsinin destek vektör olduğunu ($n_{sv} = 3$) göstermektedir[6].

$$w^* = \sum_{x_i \epsilon\ SV} y_i a_i x_i$$

eşitliği kullanılarak $w^* = y_1 a_1 x_1 + y_2 a_2 x_2 + y_3 a_3 x_3$ yazılabilir. Bu ifade de değerler yerine konulursa

$\mathrm{w}^* = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$ sonucu elde edilir. Sapma değeri b ise aşağıdaki gibi hesaplanır:

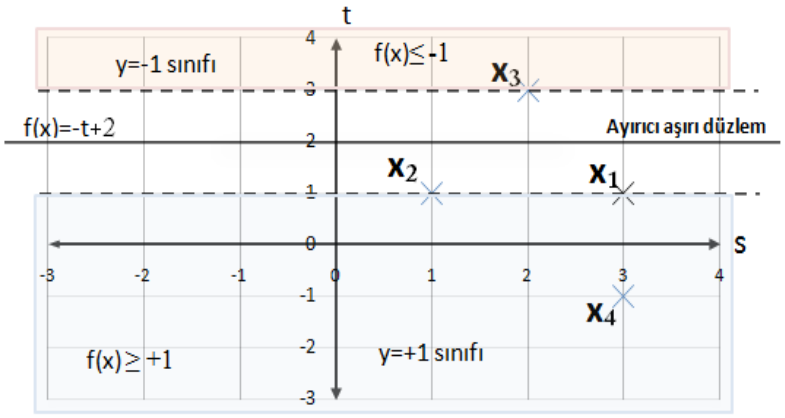
$$b = \frac{1}{n_{sv}} . \left(\sum_{s=1}^{n_{sv}} (\frac{1}{y_s} - x_s^T . w^*) \right) = \frac{1}{3} \left(\left(\frac{1}{y_1} - x_1^T . w^* \right) + \left(\frac{1}{y_2} - x_2^T . w^* \right) + \left(\frac{1}{y_3} - x_3^T . w^* \right) \right) = 2$$

bulunur. Fonksiyon ise *f(x)*= w^* .*x*+*b*=$\begin{bmatrix} 0 \\ -1 \end{bmatrix} x + 2$ olur.

Fonksiyonun bulunması ile herhangi bir (s, t) koordinatlarında olduğu varsayılan yeni bir x örneği sınıflandırması için, kullanılacak ayırıcı aşırı düzlem fonksiyonu **x=(s,t)** olmak üzere

f(x)= $\begin{bmatrix} 0 \\ -1 \end{bmatrix}$ x + 2 = $\begin{bmatrix} 0 \\ -1 \end{bmatrix}$ [s t] + 2 = -t + 2 şeklinde olur. f(x)=0 eşitliğinden t=2 bulunarak ayırıcı aşırı düzlem çizilebilir. Buna ilişkin grafik Şekil.7.'de verilmektedir. Burada ayırıcı aşırı düzlem üzerindeki noktalar cinsinden f(x)= w^* .x+b=0 eşitliğini sağladığı görülmektedir.

$\mathrm{x}_4 = (3, -1)$ verisinin ait olduğu sınıfın belirlenmesi için değerler, karar fonksiyonunda yerine konulur. Burada s=3, t=-1 ve f(x)=-t+2 olduğundan, $\mathrm{sgn}(-(-1) + 2) > 0$ olduğu için f(x)> 0 koşulu gerçekleşir ve y_i= +1 sınıfına denk gelen (+) sınıfı seçilmiş olur (Şekil.7.).



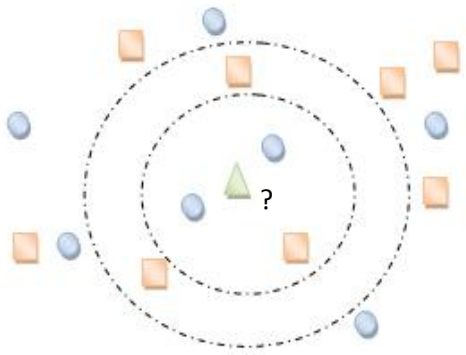
Şekil.7. Verilerin ayırıcı aşırı düzlem ile birbirinden ayılması

2.1.3.K En Yakın Komşu Algoritması (K Nearest Neighborhood- KNN Algoritm)

K en yakın komşu algoritması, sınıflandırılması istenen bir verinin, en yakın k komşuluktaki verileri baz alarak sınıflandırılmasının bir sonucu olan gözetimli öğrenme algoritmasıdır. Bu algoritma coğrafi ve uydu bilgi sistemleri, elektrokardiyografi (EKG), bilgisayar ve ağ güvenliği gibi çeşitli alanlarda kullanılabilmektedir.

Bu algoritmada sınıflandırılması istenen verinin, eğitim kümesindeki verilere uzaklıkları ayrı ayrı hesaplanır. Veriler arasındaki uzaklıklar ölçülürken en çok Öklid uzaklık formülü kullanılmaktadır[12].

Bu uzaklıklarda, en küçük uzaklıklar dikkate alınarak, sınıflandırılması istenen veriye en yakın k tane eğitim noktası bulunur. Sınıflandırma ise, bu k tane verinin en fazla olanına göre yapılır.



Şekil.8. Sınıfı belirlenecek olan bir verinin sınıflara göre konumu

Bu algoritmada, k parametresi, verilen bir noktaya en yakın komşu verilerin sayısını belirler ve sınıfın belirlenmesinde önemli bir rol oynar. Şekil.8.'de sınıflandırılması istenen üçgen şeklindeki veri göz önüne alındığında, karelere ait olan sınıfa mı, yoksa dairelere ait olan sınıfa mı dahil olacağını belirleyen k parametresinin değeri olacaktır. Örneğin, k=3 için daireye en yakın 3 örnekten 2'si daire, 1'i kare (2>1) olduğundan dolayı sınıflandırılması istenen üçgen, daire sınıfına dahil edilebilir. k=5 olduğunda ise daireye en yakın 5 komşu örnekten 3'ü kare ve 2'si dairedir. Bu durumda kare sınıfı örnekleri sayısı, daire örneklerin sayısından fazla (3>2) olduğundan dolayı üçgen, kare sınıfına dahil edilebilir.

Algoritmanın çalışma ilkesi aşağıdaki verilen adımlarla özetlenebilir:

1. Veriler arasındaki mesafeyi bulabilmek için uygun uzaklık formülü seçilir.
2. k değeri belirlenir.
3. Sınıfı belirlenmek istenen nokta ile elde mevcut olan diğer tüm noktalar arasındaki uzaklıklar ayrı ayrı hesaplanır.
4. Hesaplanan bu uzaklıklar kullanılarak küçükten büyüğe doğru sıralanır ve sınıflandırılması istenen noktaya en yakın k tanesi belirlenir.
5. Sınıflandırılması istenen nokta, seçilen bu k tane noktada en çok tekrar eden sınıfa dahil edilir ve veriye ait sınıf belirlenmiş olunur[12].

K en yakın komşu algoritmasına örnek olarak Tablo.5.'de, $X = (x_1, x_2)$ şeklinde verilmiş değer çiftleri ve buna karşılık gelen sonuçlara ait sınıflar verilmiştir. Bu verilere göre k en yakın komşu algoritması kullanılarak k=3 olmak üzere $Y = (3,1)$ noktasına karşılık gelen sonuç öğrenilmek istenmektedir.

Tablo.5. Örnek veriler

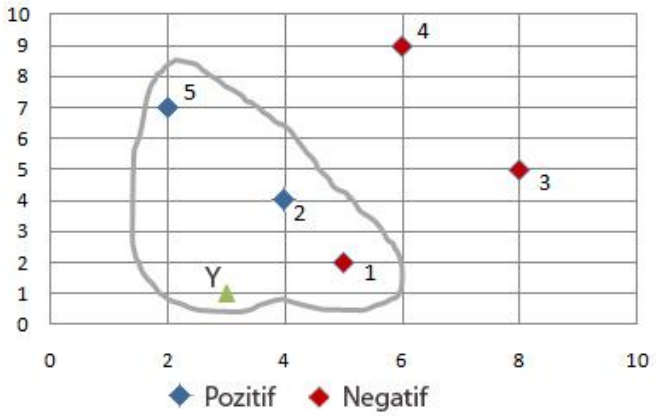
Sıra No	x_1	x_2	Karar
1	5	2	NEGATİF
2	4	4	POZİTİF
3	8	5	NEGATİF
4	6	9	NEGATİF
5	2	7	POZİTİF

Y noktasının her gözlem değeri ile arasındaki uzaklıklar Öklid formülü kullanılarak hesaplanır ve gözlem değerlerinin bu noktaya yakınlıkları belirlenir (Tablo.6).

Tablo.6. Y noktasının diğer noktalara uzaklıkları

Sıra No	x_1	x_2	Öklid Mesafesi (d)	Karar	Y noktasına göre yakınlık sıralaması
1	5	2	$\sqrt{(5-3)^2+(2-1)^2}=\sqrt{5}$	NEGATİF	1
2	4	4	$\sqrt{(4-3)^2+(4-1)^2}=\sqrt{10}$	POZİTİF	2
3	8	5	$\sqrt{(8-3)^2+(5-1)^2}=\sqrt{41}$	NEGATİF	4
4	6	9	$\sqrt{(6-3)^2+(9-1)^2}=\sqrt{73}$	NEGATİF	5
5	2	7	$\sqrt{(2-3)^2+(7-1)^2}=\sqrt{37}$	POZİTİF	3

k=3 olarak seçildiği için sınıfı belirlenmek istenen Y noktasına en yakın (minimum d mesafesindeki) 3 komşu seçilir. Bu noktalar 1, 2 ve 5 nolu noktalardır. Bunlara ait *Karar* niteliğinde en çok hangi değerin olduğu belirlenir. Bu 3 noktaya karşılık gelen *Karar* niteliğinde 2 adet *Pozitif* ve 1 adet *Negatif* değer bulunduğu için $Y = (3,1)$ noktasına karşılık gelen sınıf *Pozitif* olarak belirlenmiş olur.



Şekil.9. Y noktasının k=3 için en yakın komşuları

2.1.4. Karar Ağaçları ile Sınıflandırma Algoritmaları

Karar ağaçları, sınıfları bilinen örnek verilerden tümevarım yöntemiyle öğrenme gerçekleştirebilen ve ağaç diyagramı şeklinde gösterilebilen bir karar yapısı çeşididir [13]. Yöntem, karışık bir sınıflandırma problemini, çok aşamalı bir hale getirerek basit bir karar verme işlemi gerçekleştirdiğinden dolayı yaygınlıkla kullanılmaktadır. Diğer sınıflandırma algoritmalarıyla kıyaslandığında karar ağaçlarının yapılandırılması ve anlaşılması daha kolaydır[14].

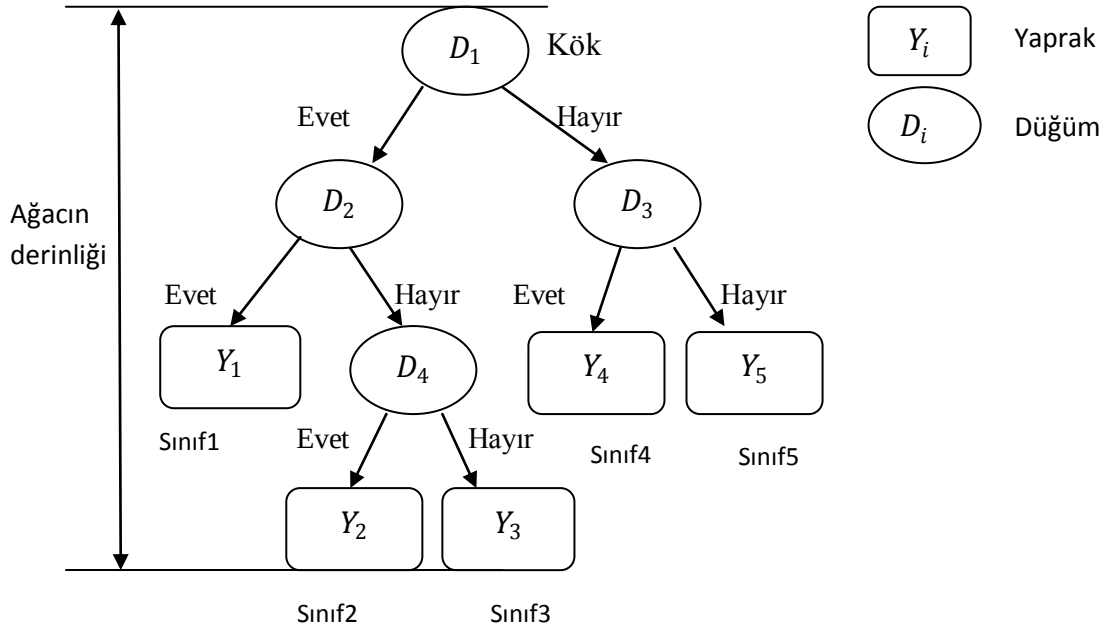
Karar ağaçları yöntemi, belirli bir sınıfın muhtemel üyesi olacak elemanların belirlenmesi, risk durumlarına göre üyelerin kategorilere ayrılması, gelecekteki

olayların tahmin edilebilmesi için kurallar oluşturulması ve veri kümesinden faydalı olacakların seçilmesi gibi temeli olan alanlarda kullanılmaktadır.

Karar ağaçları kullanılarak verinin sınıflandırılması, iki basamakta gerçekleşmektedir. Öncelikle mevcut veriler eğitim ve test kümesi olarak ikiye ayrılır ve sırasıyla şu yol izlenir [15]:

- Eğitim amacıyla kullanılacak veriler, temiz ve tutarlı bir hale dönüştürülüp karar ağacı algoritması öğrenme aşamasından geçirilir. Daha sonra algoritmanın tahminsel amaçlar için kullanabileceği bir karar ağacı modeli oluşturulur.
- İkinci aşamada, oluşturulan model test verileri ile test edilir. Böylelikle, bu karar mekanizmasına ne kadar güvenileceği bilgisi elde edilmiş olur. Daha sonra, elde edilen kurallar kullanılarak yeni verilerde tahmin edilmesi beklenen nitelikler tahmin edilebilir.

4 düğüm ve 5 yapraklı örnek bir karar ağacı Şekil.10.'da verilmiştir. Bu şekilden görüleceği gibi karar ağacının temel yapısı düğüm, dal ve yaprak olarak adlandırılan üç temel kısımdan oluşur. Bu ağaç yapısında, her bir nitelik bir düğüm $(D_1, D_2, D_3, D_4, \ldots)$ tarafından temsil edilir. Dallar ve yapraklar ağaç yapısının diğer elemanlarıdır. Ağaçta, en son kısımlar *yaprak*, en üstteki düğüm ise *kök düğüm* ya da kısaca *kök* olarak adlandırılır. Kök ve yapraklar arasında kalan kısımlar ise *dal* olarak ifade edilir[16].



Şekil.10. Örnek bir karar ağacı yapısı

Tablo.7. Karar Ağacından elde edilen kurallar tablosu

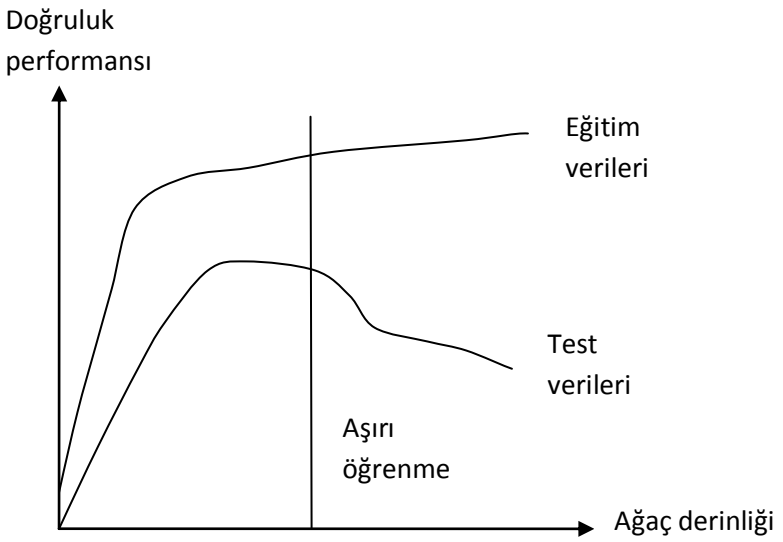
Kurallar	D_1 niteliği	D_2 niteliği	D_3 niteliği	D_4 niteliği	Sonuç
Kural 1	Evet	Evet			Sınıf 1
Kural 2	Evet	Hayır		Evet	Sınıf 2
Kural 3	Evet	Hayır		Hayır	Sınıf 3
Kural 4	Hayır		Evet		Sınıf 4
Kural 5	Hayır		Hayır		Sınıf 5

Bu sınıflandırma yönteminde amaç, ağaç oluşturulurken kullanılan eğitim verilerine ait nitelik bilgilerinden yararlanarak verilere ilişkin bir dizi sorular sorulması ve elde edilen cevaplar doğrultusunda hareket ederek en kısa sürede sonuca gidilmesidir. Bu şekilde karar ağacı sorulara aldığı cevapları toplayarak karar kuralları oluşturur(Tablo.7.). Her yaprak için, çıkarılan kurallar sonucunda öğrenme gerçekleştirilmiş olur.

Karar ağaçlarının oluşumu, yineleme ilkesine dayanır. Başlangıcında, her eğitim verisi, karar ağacının köküne (D_1) yerleştirilir ve nitelikle etiketlenmiş olan kökten çıkacak yanıta göre, bu düğümden ayrılan en az 2 dal ile dallanma (bölümleme) gerçekleştirilir. Her dallanma işlemiyle, hedef niteliği temsil eden sonuç niteliğinin üyeleri bir diğeriyle çok daha benzer hale gelmektedir[17]. Kökten sonraki ilk düğümde, en yüksek ayırımı yapan niteliğe (D_2 veya D_3) dayalı test sorgulaması yapıldıktan sonra örnek, sırayla yönlendirildiği düğüme (D_4 ve varsa diğerleri) ya da sınıf etiketlerinin temsil ettiği yapraklara iletilir. Sırayla, her düğümde sınanan örnek, son yapı olan bir yaprağa ulaşana kadar sorgulamaya devam edilir. Kökten her bir yaprağa giden tek bir yol veya tek bir karar kuralı vardır. Bu kurallarda öğrenmeyi sağlamaktadır. Öğrenme sürecinde, her yeni örnek için ayırımı sağlayan nitelik, farklı düğümlerin ve sınama sorgulamalarının oluşmasını sağlar. Bu nedenle, karar ağacı dinamik bir öğrenme yöntemidir [18]. Öğrenme gerçekleştikten sonra, elde bulunan tüm veriler ya da veri setine yeni eklenen bir veri bu algoritmadan geçirilerek sınıflanma yapılmış olunur.

Az sayıda veriden oluşan bir karar ağacında çok sayıda gerçekleşen dallanma ile çok geniş ve karmaşık bir ağaç yapısı ortaya çıkabilmektedir. Derinliği aşırı sayıda düğümden oluşan bir karar ağacında bazı verilere ulaşabilmek için veri kümesinin çok fazla sorgulamadan geçmesi gerekir ki bu da istenen bir durum

değildir. Aşırı öğrenme (*Overfitting*) adı verilen bu durumda bazı verilere, az sorgulama ile ulaşılabilecek iken veri kümesinin gürültü içermesi nedeniyle daha zor ulaşılır. Aşırı öğrenmede, eğitim verilerine ait hata oranı düşük, test verilerine ait hata oranı büyüktür[19].



Şekil.11. Karar ağaçları derinlik ve doğruluk performansı ilişkisi

Algoritmalar ağaç oluştururken, eğitim kümesi ile çalışırken bu kümedeki en ufak detayları bile yorumlar. O veriye özgü durumları kurallaştırır. Bu nedenle gereğinden fazla büyümeye başlayan ağaç, bu sırada test kümesinin doğruluğunu azaltırken hata oranını da arttırmaktadır. Ancak ağacın, eğitim kümesinin tüm verilerini tutarlı kılmaktan daha çok, test kümesine doğru cevap vermesi gerekmektedir. Bu nedenle, karar ağacında budama yapılarak söz konusu hatanın azaltılması amaçlanır. Şekil.11.'deki grafikten görüleceği gibi karar ağacının belli bir büyüklüğünden sonra (test verileri için) doğruluğu azalmaktadır. İşte bu noktada karar ağacının başarısının arttırılması için budama işleminin gerekliliği ortaya çıkmaktadır. Budama, sınıflandırmayı etkilemeyen ve sonuca bir katkısı olmayan dalların ağaçtan çıkartılması işlemidir. Budama ile gereksiz bilgilerin sonuçtan çıkarılarak karar ağacının karmaşıklığını azaltılması ve algoritma hızının arttırılması sağlanmaktadır.

Karar ağaçlarında budama iki şekilde yapılabilmektedir:

- Ön budama: Bu yöntemde aşırı dallanmayı engellemek için önceden bir eşik değeri belirlenir. Bir düğümde, dallanma sonucunda ayrılan sınıflar bu eşik değerinin altında ise dallanmaya son verilir ve bu aşamadaki baskın sınıf

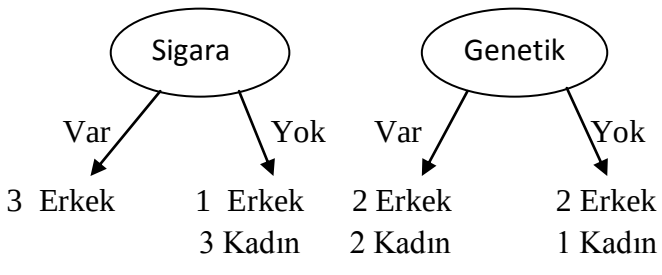
yaprak olarak sonlandırılır. Böylece önemsiz dallara ayrılacak alt ağaçların oluşması önceden engellenir. Eşik değerinin iyi belirlenmemesi bazı sorunlara yol açmaktadır. Bu değer yüksek tutulduğunda, çok genel kurallardan oluşan bir karar ağacı oluşur ve verileri iyi ayrıştıramaz, düşük tutulduğunda ise, gereğinden fazla ayrıştırma yaptığı için karar ağacı çok özele inebilir.

- Son budama: Son budama, ön budamanın aksine karar ağacı tamamen oluşturulduktan sonra uygulanan bir yöntemdir. Bu yöntem 3 farklı şekilde uygulanabilmektedir: Alt ağaçlar silinip yerine yaprak konulması ve alt ağaçların yükseltilmesi işlemi (bir üstteki düğümü silip alt ağacı yükseltme) şeklinde yapılabilmektedir. Ayrıca karar ağacı üzerinde önemsiz derecede bir ağırlığı olan alt ağacın direkt olarak ağaçtan çıkarılması olarak bilinen dal kesme yöntemi mevcuttur.

Daha öncede bahsedildiği gibi karar ağaçlarında esas olan, bir veri kümesini niteliklerine göre sorgulamalar yaptırarak en hızlı şekilde sonuca ulaşmaktır. Bir veri kümesi için, yine aynı niteliklerle çok farklı şekillerde karar ağacı kurulması mümkündür. Ancak bir veri kümesi için ideal tek bir karar ağacı vardır. Bundan kasıt, ideal olan karar ağacı, öğrenme kümesi dışındaki verilerde de aynı kuralları oluşturur ya da az hata payıyla aynı hipotez sonuçlarını ortaya çıkartmaktadır [20]. İdeal olan karar ağaçlarının kurulmasında dikkat edilmesi gereken en önemli aşama en ayırıcı niteliklerin tespitidir. Yani en homojen sonuçların oluşmasını sağlayan nitelikler öncelikli düğümleri oluşturmalıdır. Tablo.8.'deki bir veri kümesi için tek bir dallanma iki farklı niteliğe göre gösterilmiştir:

Tablo.8. Hastaların etkilendiği faktörlere ilişkin istatistikî veri kümesi

Sigara	Genetik	Cinsiyet
Var	Yok	Erkek
Var	Yok	Erkek
Var	Var	Erkek
Yok	Var	Erkek
Yok	Var	Kadın
Yok	Var	Kadın
Yok	Yok	Kadın



Şekil.12. Farklı dallanmalara göre oluşan yapraklar

Yukarıdaki örnekte sigara niteliği, genetik niteliğine göre daha ayırıcı bir özellik olmuştur. Çünkü hedef niteliği temsil eden cinsiyete ilişkin sınıflar baz alındığında, aynı sınıfta olan veriler birbiri ile daha çok yan yana yer almıştır. Kısacası sigara niteliği, genetik niteliğine göre cinsiyeti daha iyi temsil etmektedir. Şöyle ki; sigara niteliğine ait düğümde, genetik niteliği dikkate alınmadan bile "*Sigara içen hastaların hepsi erkektir*" sonucu çıkarılabilmektedir. Genetik niteliğinin yer aldığı düğümde ise ayrım kötü olduğu için çok net bir sonuç çıkarılamamaktadır. Bu da bu düğümün kalitesinin düşük olduğunun göstergesidir.

Karar ağaçlarının oluşturulmasındaki en önemli adım ağaçtaki dallanmanın hangi kriterlere göre yapılacağı ya da hangi nitelik değerlerine göre ağaç yapısının oluşturulacağıdır. Bunun için pek çok algoritma geliştirilmiştir. Bu algoritmalar, karar ağacı oluştururken, kök, düğüm seçimi ve dallanma kriterleri bakımından birbirlerinden ayrılmaktadır. Bu çalışmada, entropiye dayalı algoritmalardan ID3 ve C4.5 algoritmalarına değinilecektir.

Karar ağaçları, sürekli nitelik değerlerinde kullanılabilmelerine rağmen, bu değerleri tahmin etmede ise çok başarılı değildir. Diğer bir dezavantajı ise sınıf sayısı fazla ve öğrenme kümesi örnekleri sayısı az olduğunda model oluşturma çok başarılı olmamasıdır. Bunun yanında yorumlanması kolaydır ve anlaşılabilir kurallar oluşturmaktadır[19]. Oluşturulan karar ağacının her yaprağı için kökten başlayarak, her düğüme uğrayarak If (Eğer)–Then (İse)–Else (Değilse) yapısı formunda karar kuralları yazılabilmektedir.

2.1.4.1. ID3 Algoritması (Iterative Dichotomiser 3 Algorithm)

ID3 algoritması, kökten başlayarak, bir karar ağacının her dallanma durumunda veri kümesinin en ayırıcı niteliğini bularak, adım adım sabit bir veri kümesi için karar ağacının oluşturulmasını sağlayan bir algoritmadır ve ilk olarak J. Ross tarafından 1986 yılında geliştirilmiştir.

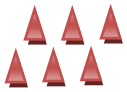
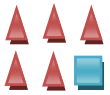
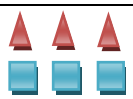
ID3 algoritmasında, hedef kümenin en ayırıcı niteliklerini belirlemek üzere entropi kurallarını temel alan bilgi teorisi kullanılmaktadır. Entropi, bir sistemdeki belirsizlik olarak tanımlanabilir. Tek hedef nitelikli karar ağaçlarında ID3 algoritması, bilgi kazancı yaklaşımını kullanmaktadır[16].

S, hedef niteliği temsil eden bir veri kümesi olmak üzere, bu kümede yer alan her farklı sınıftan verinin gerçekleşme olasılıkları $P=\{p_1,p_2,p_3,....p_n\}$ olmak üzere, S kümesine ait entropi (2.28)'deki gibi hesaplanır:

$$Entropi(S) = -\sum_{i=1}^{n} p_i log_2(p_i) \tag{2.28}$$

Entropi, beklenmeyen bir durumun ortaya çıkma olasılığıdır ve belirsizliğin artması ile entropi değeri de artmaktadır. Bir kümeye ait entropinin minimum değeri 0, maksimum değeri ise 1 olabilir. Bu değerler, 2 sınıf ve 6 örnek veriden oluşan bir küme için Tablo.9.'da irdelenmiştir.

Tablo.9. Farklı sınıflara ait veri sayısı ve entropi ilişkisi

Örnek küme (S)	Entropi (S)	Açıklama
	0	**Homojen olma durumu**: Tüm veriler aynı sınıfta olduğundan belirsizlik yoktur. Çünkü bir verinin, başka bir sınıftan olması mümkün değildir.
	0<Entropi(S)<1	**Örneklerin sınıflara sayıca rastgele dağılması**: Farklı sınıftan birisi diğerlerine göre sayıca baskındır. Bu nedenle bu verinin gerçekleşme olasılığı fazladır ve belirsizlik arada bir değerdedir.
	1	**Bütün örneklerin sınıflara eşit dağılma durumu**: Farklı sınıflardaki verinin gerçekleşme olasılığı eşittir ve baskınlık söz konusu olmadığından belirsizlik maksimumdur.

Bir karar ağacında her karar aşamasında yani dallanma sırasında, en uygun ağacın kurulması için algoritmanın *bilgi kazancı* ya da kısaca *kazanç* adı verilen bir veriye ihtiyacı vardır.

S hedef niteliği ve A bunun haricinde kalan diğer niteliklerinden birisi olmak üzere; S hedef niteliğinin, A niteliğinin değerine bağlı olarak $S=\{S_1, S_2, \ldots, S_n\}$ şeklinde alt kümelere ayrılması mümkündür. Bu durumda, A niteliğinin S veri kümesindeki bilgi kazancı (2.29)'daki gibi olur:

$$Kazanç(S,A) = Entropi(S) - \sum_{v \in Values\,(A)} \frac{|S_v|}{|S|}.Entropi(S_v) \tag{2.29}$$

Values(A), A niteliğinin alabileceği değerleri, S_v ise A'nın *v* olduğu durumda S'nin altkümesini ifade etmektedir.

Algoritma her nitelik için bilgi kazancını hesaplar ve en yüksek bilgi kazancı değerine sahip nitelik belirlenir. Daha sonra niteliğin adıyla adlandırılan bir düğüm oluşturulur. Bu düğümün her değeri için bir dal oluşturulur ve dal tarafından kendisine yeni bir özellik kazandırılmış olan hedef niteliğe ait alt kümeler de yeni oluşturulan alt düğüme yerleştirilir. Bu düğümde de nitelik değerlerine bağlı olarak dallanma meydana getirilir. Ancak, hedef alt kümesi göz önüne alınacağından dolayı entropi ve kazançlar yeniden hesaplanır. Bu da algoritmanın kendini yenileme özelliğinin var olduğunu göstermektedir. Karar ağacındaki dallanmalar, bir düğümdeki bütün örnekler tek bir sınıfa ait olana kadar algoritma her bir alt düğüme tekrar tekrar uygulanır.

ID3 algoritması örneği için bir hastane veritabanın da damar tıkanıklığı için ön kontrol yaptıran kişilerin adları, laboratuvar sonuçları ve rahatsızlığın tespit edilip edilmediğini belirten test sonucuna ilişkin bilgiler Tablo.10.'da verilmiştir.

Tablo.10. Hastalara ait testlerin sonuçları

No.	Kolesterol	Kan şekeri	Tansiyon	Kilo	Test sonucu
01	Düşük	Normal	Düşük	Şişman	Pozitif
02	Normal	Normal	Yüksek	Şişman	Negatif
03	Normal	Düşük	Normal	Normal	Negatif
04	Düşük	Düşük	Normal	Şişman	Pozitif
05	Düşük	Yüksek	Normal	Normal	Negatif
06	Normal	Yüksek	Yüksek	Şişman	Negatif
07	Düşük	Düşük	Düşük	Normal	Negatif
08	Yüksek	Normal	Yüksek	Şişman	Pozitif

Rahatsızlıkla ilişkili en önemli faktör ile ilgili bir araştırma yapılması istenmektedir. Bu veri kümesi kullanılarak, niteliklere ait hangi durumlarda rahatsızlığın görüldüğü ve hangi durumlarda rahatsızlığın görülmediği ortaya çıkarılmak istenmektedir. Bir başka deyişle istenen, sonuçlara ilişkin kuralların elde edilmesidir. Kurallar bulunduğunda, rahatsızlığın kimlerde en fazla risk oluşturabileceği irdelenecek ve sonuçlar tedavide yarar sağlayacaktır.

Problem için karar ağacı oluşturulacaktır ve öncelikle kök düğümü temsil edecek niteliğin bulunması gerekmektedir. Hedef niteliği (S) *test sonucu*dur ve sınıflandırılacak verilere ait sonuçları içermektedir. Nitelikleri ise *kolesterol*, *kan*

şekeri, tansiyon ve *kilo*dan oluşmaktadır. Daha önceden de bahsedildiği gibi düğümler oluşturulurken en çok bilgi kazancını sağlayan nitelikler seçilmektedir. Kazanç formülünde yer alan S hedef niteliğine ait entropi hesaplanacak olursa:

S=[3+, 5-] :

$$Entropi(S) = -\sum_{i=1}^{n} p_i log_2(p_i) = -\left(\frac{3}{8}\right) log_2\left(\frac{3}{8}\right) - \left(\frac{5}{8}\right) log_2\left(\frac{5}{8}\right) = 0.95443$$

olarak hesaplanır.

A niteliği: *Kolesterol* *v*=[Düşük, Normal, Yüksek]

v: *Düşük* $\Rightarrow S_{Düşük}$=[2+,2-] $\Rightarrow Entropi(S_{Düşük})$=$-\left(\frac{2}{4}\right) log_2\left(\frac{2}{4}\right) - \left(\frac{2}{4}\right) log_2\left(\frac{2}{4}\right)$= 1

v: *Normal* $\Rightarrow S_{Normal}$ =[0+,3-] $\Rightarrow Entropi(S_{Normal})$=$-\left(\frac{0}{3}\right) log_2\left(\frac{0}{3}\right) - \left(\frac{3}{3}\right) log_2\left(\frac{3}{3}\right)$= 0

v: *Yüksek* $\Rightarrow S_{Yüksek}$ =[1+,0-] $\Rightarrow Entropi(S_{Yüksek})$=$-\left(\frac{1}{1}\right) log_2\left(\frac{1}{1}\right) - \left(\frac{0}{1}\right) log_2\left(\frac{0}{1}\right)$= 0

Kolesterol için kazanç değeri hesaplanır:

Kazanç(*S, Kolesterol*)= $0.95443 - (\frac{4}{8}.1 + \frac{3}{8}.0 + \frac{1}{8}.0)$= 0,45443

A niteliği: *Kan şekeri* *v*=[Düşük, Normal, Yüksek]

v: *Düşük* $\Rightarrow S_{Düşük}$=[1+,2-] $\Rightarrow Entropi(S_{Düşük})$=$-\left(\frac{1}{3}\right) log_2\left(\frac{1}{3}\right) - \left(\frac{2}{3}\right) log_2\left(\frac{2}{3}\right)$= 0.91829

v:*Normal*$\Rightarrow S_{Normal}$ =[2+,1-]$\Rightarrow Entropi(S_{Normal})$=$-\left(\frac{2}{3}\right) log_2\left(\frac{2}{3}\right) - \left(\frac{1}{3}\right) log_2\left(\frac{1}{3}\right)$= 0.91829

v: *Yüksek* $\Rightarrow S_{Yüksek}$ = [0+,2-] $\Rightarrow Entropi(S_{Yüksek})$=$-\left(\frac{0}{2}\right) log_2\left(\frac{0}{2}\right) - \left(\frac{2}{2}\right) log_2\left(\frac{2}{2}\right)$= 0

Kan şekeri için kazanç değeri hesaplanır:

Kazanç(*S, Kan şekeri*)= $0.95443 - (\frac{3}{8}.0{,}91829 + \frac{3}{8}.0{,}91829 + \frac{2}{8}.0)$= 0,26571

A niteliği: *Tansiyon* *v*=[Düşük, Normal, Yüksek]

v: *Düşük* $\Rightarrow S_{Düşük}$=[1+,1-] $\Rightarrow Entropi(S_{Düşük})$=$-\left(\frac{1}{2}\right) log_2\left(\frac{1}{2}\right) - \left(\frac{1}{2}\right) log_2\left(\frac{1}{2}\right)$= 1

v:*Normal*$\Rightarrow S_{Normal}$ =[1+,2-] $\Rightarrow Entropi(S_{Normal})$=$-\left(\frac{1}{3}\right) log_2\left(\frac{1}{3}\right) - \left(\frac{2}{3}\right) log_2\left(\frac{2}{3}\right)$= 0.91829

v: *Yüksek* $\Rightarrow S_{Yüksek}$ =[1+,2-] $\Rightarrow Entropi(S_{Yüksek})$=$-\left(\frac{1}{3}\right) log_2\left(\frac{1}{3}\right) - \left(\frac{2}{3}\right) log_2\left(\frac{2}{3}\right)$= 0.91829

Tansiyon için kazanç değeri hesaplanır:

Kazanç(*S, Tansiyon*)= $0.95443 - (\frac{2}{8}.1 + \frac{3}{8}.0{,}91829 + \frac{3}{8}.0{,}91829)$ = 0,01571

A niteliği: *Kilo* *v*=[Normal, Şişman]

v: *Normal* $\Rightarrow S_{Normal}$ =[0+,3-] $\Rightarrow Entropi(S_{Normal})$=$-\left(\frac{0}{3}\right)log_2\left(\frac{0}{3}\right) - \left(\frac{3}{3}\right)log_2\left(\frac{3}{3}\right)$= 0

v:*Şişman*$\Rightarrow S_{Şişman}$ =[3+,2-]$\Rightarrow Entropi(S_{Şişman})$=$-\left(\frac{3}{5}\right)log_2\left(\frac{3}{5}\right) - \left(\frac{2}{5}\right)log_2\left(\frac{2}{5}\right)$= 0.97095

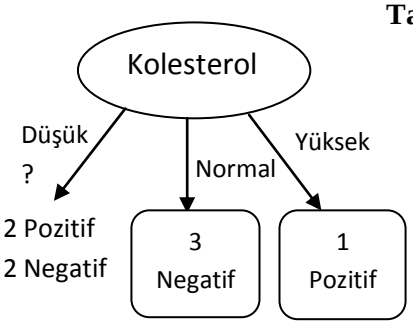
Kilo için kazanç değeri hesaplanır:

Kazanç(*S, Kilo*)= $0.95443 - (\frac{3}{8}.0 + \frac{5}{8}.0{,}97095)$ = 0,3475

Tablo.11. Nitelikler ve kazanç değerleri

Nitelikler	Kazanç
Kolesterol	0,45443
Kan şekeri	0,26571
Tansiyon	0,01571
Kilo	0,3475

Tablo.11.'den görüleceği gibi nitelikler arasında en yüksek kazanç değerini *kolesterol* sağlamıştır. Bir başka deyişle, *kolesterol* bu veriler arasında *test sonuçları* hedef niteliğine göre en ayırıcı niteliktir. Bu nedenle ilk düğüm, *kolesterol* olarak belirlenir.



Şekil.13. Kök düğümün belirlenmesi sonucu oluşan karar ağacı

Tablo.12. Kök düğümün farklı sınıfları içeren durumu

Kolesterol	**Kan şekeri**	**Tansiyon**	**Kilo**	**Test sonucu**
Düşük	Normal	Düşük	Şişman	Pozitif
Düşük	Düşük	Normal	Şişman	Pozitif
Düşük	Yüksek	Normal	Normal	Negatif
Düşük	Düşük	Düşük	Normal	Negatif

Kök düğüm olarak belirlenen *kolesterol* niteliği her durumuna bağlı olarak dallara ayrılır (Şekil.13). Hedef kümenin her bir verisi, dalları temsil eden durumlara göre sınıflandırılır. Görüldüğü gibi *kolesterol*ün normal ya da yüksek olduğu durumlarda veriler aynı sınıftan olduğu için başka bir niteliğin dallanma yapmasına gerek yoktur. Bu nedenle bu durumlar için yaprak oluşturulur. Ancak *kolesterol*ün düşük olduğu durum için farklı sınıflara ait veriler bulunmaktadır. Bu nedenle, diğer

niteliklerden birini ya da birkaçını kullanarak bir düğüm noktası oluşturup yeni bir dallanma yaratarak verilerin sınıflara ayrılması gerekmektedir.

Bu noktada veri tabanı; *kolesterol*=düşük durumu için dikkate alınır (Tablo.12). Öncelikle oluşan yeni hedef kümeye göre entropi hesaplanmalıdır:

S=[2+, 2-] :

$$Entropi(S) = -\sum_{i=1}^{n} p_i log_2(p_i) = -\left(\frac{2}{4}\right) log_2\left(\frac{2}{4}\right) - \left(\frac{2}{4}\right) log_2\left(\frac{2}{4}\right) = 1$$

Farklı sınıflardaki veriler eşit sayıda olduğu için belirsizlik maksimuma ulaşmıştır.

A niteliği: *Kan şekeri* *v*=[Düşük, Normal, Yüksek]

v: *Düşük* $\Rightarrow S_{\text{Düşük}}$=[1+,1-] $\Rightarrow Entropi(S_{\text{Düşük}})$=$-\left(\frac{1}{2}\right) log_2\left(\frac{1}{2}\right) - \left(\frac{1}{2}\right) log_2\left(\frac{1}{2}\right)$=1

v: *Normal* $\Rightarrow S_{Normal}$ = [1+,0-] $\Rightarrow Entropi(S_{Normal})$=$-\left(\frac{1}{1}\right) log_2\left(\frac{1}{1}\right) - \left(\frac{0}{1}\right) log_2\left(\frac{0}{1}\right)$=0

v: *Yüksek* $\Rightarrow S_{Yüksek}$ = [0+,1-] $\Rightarrow Entropi(S_{Yüksek})$=$-\left(\frac{0}{1}\right) log_2\left(\frac{0}{1}\right) - \left(\frac{1}{1}\right) log_2\left(\frac{1}{1}\right)$ = 0

Kan şekeri için kazanç değeri hesaplanır:

Kazanç(*S, Kan şekeri*)= $1 - (\frac{2}{4}.1 + \frac{1}{4}.0 + \frac{1}{4}.0)$ = 0,5

A niteliği: *Tansiyon* *v*=[Düşük, Normal]

v: *Düşük* $\Rightarrow S_{\text{Düşük}}$=[1+,1-] $\Rightarrow Entropi(S_{\text{Düşük}})$=$-\left(\frac{1}{2}\right) log_2\left(\frac{1}{2}\right) - \left(\frac{1}{2}\right) log_2\left(\frac{1}{2}\right)$ = 1

v: *Normal* $\Rightarrow S_{Normal}$ =[1+,1-] $\Rightarrow Entropi(S_{Normal})$=$-\left(\frac{1}{2}\right) log_2\left(\frac{1}{2}\right) - \left(\frac{1}{2}\right) log_2\left(\frac{1}{2}\right)$ = 1

Tansiyon için kazanç değeri hesaplanır:

Kazanç(*S, Tansiyon*)= $1 - (\frac{2}{4}.1 + \frac{2}{4}.1)$=0

A niteliği: *Kilo* *v*=[Normal, Şişman]

v: *Normal* $\Rightarrow S_{Normal}$ =[0+,2-] $\Rightarrow Entropi(S_{Normal})$=$-\left(\frac{0}{2}\right) log_2\left(\frac{0}{2}\right) - \left(\frac{2}{2}\right) log_2\left(\frac{2}{2}\right)$ = 0

v: *Şişman* $\Rightarrow S_{Şişman}$ =[2+,0-] $\Rightarrow Entropi(S_{Şişman})$=$-\left(\frac{2}{2}\right) log_2\left(\frac{2}{2}\right) - \left(\frac{0}{2}\right) log_2\left(\frac{0}{2}\right)$ = 0

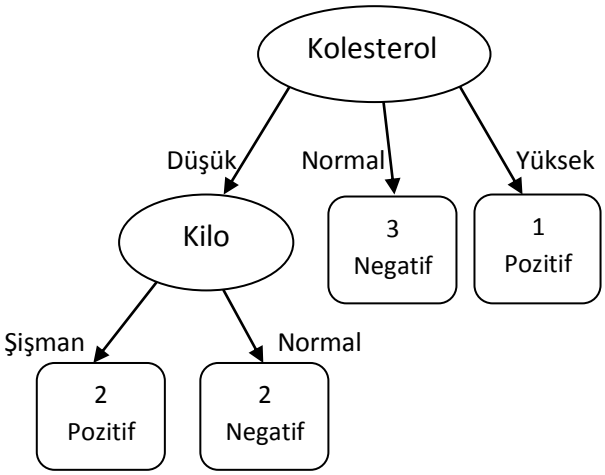
Kilo için kazanç değeri hesaplanır:

Kazanç(*S, Kilo*)= $1 - (\frac{2}{4}.0 + \frac{2}{4}.0)$ =1

Tablo.13. Nitelikler ve kazanç değerleri

Nitelikler	Kazanç
Kan şekeri	0,5
Tansiyon	0
Kilo	1

Bu aşama için en yüksek kazanç sağlayan nitelik olan *kilo* niteliği, yeni düğüme ait nitelik olarak belirlenir ve karar ağacı aşağıdaki son hali alır:



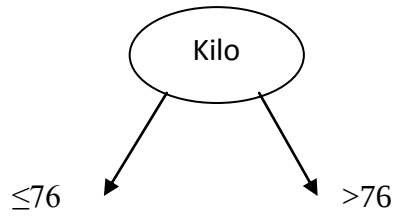
Şekil.14. Karar ağacının son hali

Karar ağacı oluşturulduktan sonra kural oluşturulması ve yorumlanması kolaydır. Bu veri kümesi için hedef kümeyi sınıflara en iyi ayıran nitelik *kolesterol*, daha sonra *kilo* olmuştur (Şekil.14.). Başka bir deyişle, damar tıkanıklığı, en çok *kolesterol*, daha sonra *kilo* ile özdeşleştirilebilir. Bu sonucun bulunması, kuşkusuz tanıda kolaylık sağlamaktadır. Bu da karar ağaçlarının uygulamalarda verimli kullanılabileceğinin en basit göstergesidir. Karar ağaçlarının asıl yararı sonuçlara ilişkindir. Burada 4 adet yaprak 4 ayrı kuralı ifade etmektedir. Kurallar, programlama dillerindeki if-then-else yapısı formunda, kökten her yaprağa giden yol için çıkarılabilir:

If kolesterol=yüksek then test sonucu=pozitif
If kolesterol=normal then test sonucu=negatif
If kolesterol=düşük and kilo=şişman then test sonucu=pozitif
If kolesterol= düşük and kilo=normal then test sonucu=negatif

2.1.4.2. C4.5 Algoritması (C4.5 Algorithm)

ID3 algoritması, karar ağaçları oluşturmada kullanılmasına rağmen bazı noktalarda yetersiz kalan bir algoritmadır. Bu eksiklerin giderilmesi için C4.5 algoritması geliştirilmiştir. ID3 algoritmasının yetersiz olduğu noktalardan birisi niteliklerin, kategorik veriler yerine, birbirleri ile çoğunlukla eşit olmayan sayısal değerlerden oluşmasında görülmektedir[21]. Örneğin Tablo.11.' deki örnek verilerde *kilo* niteliği {Normal, Şişman} sınıfları yerine {67, 69, 74, 78, 81, 89} gibi sayısal değerlerden oluşuyor olsa idi, bu değerlerin her biri için bir dal oluşturulacak ve karar ağacının gereksiz yere çok büyümesine ve dallanmasına yol açmaktadır. Çok yakın değerler için dahi farklı yapraklar oluşturan bu durum, veriler için istenilen sınıflandırma işlemini yapılmasını engellemektedir. C4.5 algoritması, bu soruna çözüm olarak söz konusu niteliğe ait verilerde en büyük bilgi kazancını sağlayacak eşik değer (e) yaklaşımı getirmektedir. Sürekli değişkenler içerisinde, uygun eşik değeri bulunduktan sonra ikili ya da daha çok bölünme ile veri kümesi bölünebilir. Bu yaklaşımla veriler, küçükten büyüğe doğru sıralanır. İkili bölünme gerçekleştirildiğinde $\{v_1, v_2, \dots, v_m\}$ şeklinde sıralanan veriler $\{v_1, v_2, \dots, v_i\}$ ve $\{v_{i+1}, v_{i+2}, \dots, v_m\}$ şeklinde iki parçaya ayrılır. Eşik değer ise (e)= $\frac{v_i+v_{i+1}}{2}$ şeklinde bulunur. Sayısal verilerden oluşan bu küme, eşik değerden küçük ve büyük veriler diye iki alt kümeye ayrılmaktadır. Buna alternatif olarak birkaç farklı olasılık söz konusu olduğunda çoklu bölünme kullanılabilir. Nitelik üzerinde m-1 adet ayrım mümkündür ve her bir aralık arasında $\frac{v_i+v_{i+1}}{2}$ eşiği temsil eden orta nokta olarak belirlenebilir. Bütün bu ayırımlar en uygun bölünmeyi elde etmek için sistemli bir şekilde incelenmelidir [16]. Örnek olarak {67, 69, 74, 78, 81, 89} kümesi için ikili bölünme gerçekleştirilirse, küme {67, 69, 74} ve {78, 81, 89} şeklinde iki parçaya ayrılır. $Eşik\ değer(e) = \frac{74+78}{2}$=76 alınabilir ve niteliğe ait karar düğümü dalları eşik değer göz önüne alınarak Şekil.15.'deki gibi belirlenir.



Şekil.15. Sürekli değişken değerlere sahip bir niteliğin dallanması

C4.5 algoritmasının ID3 algoritmasına göre başka bir üstünlüğü ise kayıp verilerle de çalışabilmesidir. ID3 algoritmasında bazı niteliklere ait verilerde, eksiklikler varsa bu durum entropi, kazanç gibi değerlerin hesaplanmasında sorun oluşturmaktadır. Algoritma, bu durumda eksik veriye bağlı diğer tüm değerleri göz önüne almadan entropi ve kazanç değerlerini hesaplar. Ancak bulunan kazanç değerleri bir düzeltme faktörü yardımıyla yeniden hesaplamaktadır[6].

Eksik verisi olan bir A niteliğinin, eksik verilere bağlı diğer nitelik verileri de çıkarılarak hesaplanan $Kazanç_{ilk}(A)$ olmak üzere, düzeltilmiş olan kazanç (2.30)' daki şekilde hesaplanır:

$$Kazanç_{son}(A) = Kazanç_{ilk}(A).\ (\frac{Niteliklerindeki\ tüm\ değerleri\ bilinen\ örnek\ sayısı}{Tüm\ örneklerin\ sayısı})$$

(2.30)

Kategorik (nominal) verilerden oluşan bir nitelikte, değerlere ait çeşitlilik arttıkça, bilgi kazancı gereksiz bir artış göstermektedir. Bu şekilde verilerden oluşan niteliklerin sınıflandırmaya bir katkısı olmadığı gibi, bilgi kazancı yüksek olan niteliklerin bulunmasını da engellemektedir. Örneğin Tablo.11.'de hasta numaralarının nitelik olarak kabul edildiği düşünüldüğünde, bilgi kazancı yaklaşımı ile hasta numaraları en yüksek kazanç sağlayan nitelik olmaktadır. Bu durumda, karar ağacında sınıflandırma hasta numaralarına göre yapılmış olacaktır. Buda istenmeyen bir durumdur. C4.5 algoritması, *bölünme bilgisi* kavramı ile değer çeşitliliği fazla olan özelliklerin bilgi kazancını azaltarak algoritmanın gereksiz bazı çıkarımlar yapılmasını engellemektedir[22]. C4.5 algoritması ile oluşturulacak karar ağacı için en ayırıcı nitelikler, ID3 algoritmasından farklı olarak maksimum kazanç oranına göre belirlenmektedir.

Hedef niteliği S olan bir veritabanında, bir A niteliği, değerlerin aynı olduğu durumlara bağlı olarak $A_1, A_2, \dots, A_k$ gibi alt kümelere ayrılırsa, A niteliğinin bölünme bilgisi ve kazanç oranı aşağıdaki şekilde hesaplanmaktadır:

$$Bölünme\ bilgisi(A) = -\sum_{i=1}^{k} \frac{|A_i|}{|A|}\ log_2(\frac{|A_i|}{|A|})$$

(2.31)

$$Kazanç\ oranı(S, A) = \frac{Kazanç(S, A)}{Bölünme\ bilgisi(A)}$$

(2.32)

C4.5 algoritmasına örnek olarak Tablo.11.'deki bilgiler kullanılarak karar ağacı oluşturulmak istenilirse en ayırıcı niteliklerin bulunması için öncelikle 4 niteliğe ait bölünme bilgisi değerlerinin belirlenmesi gerekmektedir.

A niteliği: *Kolesterol* v=[Düşük, Normal, Yüksek]=[4,3,1]

$$Bölünme\ bilgisi(Kolesterol) = -\left(\frac{4}{8}\right) log_2\left(\frac{4}{8}\right) - \left(\frac{3}{8}\right) log_2\left(\frac{3}{8}\right) - \left(\frac{1}{8}\right) log_2\left(\frac{1}{8}\right) = 1{,}4056$$

$$Kazanç\ oranı(S, Kolesterol) = \frac{0{,}45443}{1{,}4056} = 0{,}3233$$

A niteliği: *Kan şekeri* v=[Düşük, Normal, Yüksek]=[3,3,2]

$$Bölünme\ bilgisi(Kan\ şekeri) = -\left(\frac{3}{8}\right) log_2\left(\frac{3}{8}\right) - \left(\frac{3}{8}\right) log_2\left(\frac{3}{8}\right) - \left(\frac{2}{8}\right) log_2\left(\frac{2}{8}\right) = 1{,}5612$$

$$Kazanç\ oranı(S, Kan\ şekeri) = \frac{0{,}26571}{1{,}5612} = 0{,}1702$$

A niteliği: *Tansiyon* *v*=[Düşük, Normal, Yüksek]=[2,3,3]

$$Bölünme\ bilgisi(Tansiyon) = -\left(\frac{2}{8}\right) log_2\left(\frac{2}{8}\right) - \left(\frac{3}{8}\right) log_2\left(\frac{3}{8}\right) - \left(\frac{3}{8}\right) log_2\left(\frac{3}{8}\right) = 1{,}5612$$

$$Kazanç\ oranı(S, Tansiyon) = \frac{0{,}01571}{1{,}5612} = 0{,}01$$

A niteliği: *Kilo v*=[Normal, Şişman]=[3,5]

$$Bölünme\ bilgisi(Kilo) = -\left(\frac{3}{8}\right) log_2\left(\frac{3}{8}\right) - \left(\frac{5}{8}\right) log_2\left(\frac{5}{8}\right) = 0{,}95439$$

$$Kazanç\ oranı(S, Kilo) = \frac{0{,}3475}{0{,}95439} = 0{,}36411$$

Tablo.14. Nitelikler ve kazanç oranları

Nitelikler	Kazanç oranı
Kolesterol	0,3233
Kan şekeri	0,1702
Tansiyon	0,01
Kilo	0,36411

Tablo.14.'den görüleceği gibi en yüksek kazanç oranını sağlayan *kilo* niteliği ilk aşamada kök düğüm olarak belirlenir. Bundan sonraki homojen olmayan dallanmalarda da yine en büyük kazanç oranını sağlayan nitelikler, düğümlere etiketlenir ve karar ağacı oluşturulmuş olunur.

2.2. Gözetimsiz Öğrenme (Unsupervised learning)

Gözetimsiz öğrenme, girdi verilerinin en uygun gösterim şeklini belirleyen, bir girdi kümesi modellemesinde kullanılan öğrenme türüdür. Bu öğrenmede, girdi verilerine karşılık gelen herhangi bir çıktı verisi (sınıf ya da etiket bilgileri) yoktur. Hiçbir girdinin hangi sınıfta olduğu bilinmemektedir. Bu bakımdan sistem, çıktı verilerini kullanmadan sadece girdi bilgilerini kullanarak öğrenme gerçekleştirmektedir. Gözetimsiz öğrenme yöntemlerinde, sistemin öğrenmesine yardımcı olan herhangi bir denetçi yoktur ve verilerdeki parametreler arasındaki ilişkiyi sistemin kendi kendine öğrenmesi beklenir. Sistem, sınıfları bilinmeyen girdi kümesini parçalara ayırarak girdi verilerinin gruplandırılmasını ya da kümelenmesini sağlamaktadır. Bu nedenle kümeleme, en temel gözetimsiz öğrenme yöntemidir [23]. Girdi verilerinin, etiket ya da sınıf bilgileri bilinmemesine rağmen, çıktı verilerinde gruplandırılma yapılarak anlamlı kümeler oluşturulabilir. Şöyle ki, öğrenme işlemi tamamlandıktan sonra kullanıcı tarafından bu grupların ne anlama geldiğini gösteren etiketlendirme yapılması kümeleme yorumunu kolaylaştırmaktadır[24]. Bu tip öğrenme yöntemleri, yoğunluk tahmini, dağılımın desteklenmesini öğrenmek vb. durumları içerir.

Bu çalışmada, gözetimsiz öğrenme algoritmalarına örnek olarak K-means, Hiyerarşik Kümeleme yöntemleri açıklanmaktadır.

2.2.1. K –Means Algoritması (K-Means Algorithm)

K-means algoritması, elde mevcut bulunan verileri, kullanıcı tarafından belirlenen k parametresi kadar kümeye ayıran, gerçeklemesi kolay gözetimsiz öğrenme algoritmalarından biridir. Bu algoritma, benzer özellik gösteren verilerin, bir arada kümelenmesi esasına dayanır. Algoritmadaki amaç, oluşturulan k adet kümenin, kendi içlerinde benzerliklerinin maksimum, birbirleri arasındaki benzerliklerinin ise minimum olmasını sağlamaktır[25].

K -means yöntemi, sadece kümedeki verilerin ortalamasının tanımlanabildiği durumlarda kullanılabilen bir yöntemdir. Küme merkezleri, kümeleri oluşturan sayısal verilerin aritmetik ortalamasıdır.

Bu algoritmada benzerlik kavramı veriler arasındaki uzaklıklarla tespit edilir ve mesafenin az olması benzerliğin büyük olduğu anlamına gelir. Veriler ile küme merkezleri arasındaki mesafeyi hesaplamak için farklı uzaklık formülleri kullanılabilir:

n-boyutlu bir uzayda $A = (x_1, .., x_n)$ ve $B = (y_1, .., y_n)$ gibi iki nokta için Öklid uzaklık formülü (2.33)'de verilmektedir.

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} \tag{2.33}$$

Yaygın olarak Öklid uzaklık formülü kullanılsa da bunun yerine, Manhattan (City-block), Chebychev ya da Minkowski uzaklık formülü de kullanılabilir.

K-means algoritmasının çalışma ilkesi aşağıdaki verilen adımlarla özetlenebilir:

1. Verilerin ayrılacağı k küme sayısı kullanıcı tarafından belirlenir.
2. Veriler arasından k adet veri, küme merkezi olarak rastgele seçilir.
3. Diğer verilerin her birinin, küme merkezlerine olan uzaklığı çözüm için kullanılacak olan bir uzaklık formülü ile tek tek hesaplanır. Bu değerlere göre her veri kendisine en yakın merkezi bulunan kümeye atanır.
4. Bu atamalar sonucunda kümelerde bulunan verilerin ortalaması hesaplanarak yeni küme merkezleri hesaplanır.
5. Küme elemanlarında dolayısı ile küme merkezlerinde bir değişiklik yoksa algoritma sonlandırılır, varsa değişiklik olmayana kadar 3. ve 4. adımlar tekrarlanır.

Buradan da anlaşılacağı gibi oluşturulan küme elamanları sürekli küme değiştirebilmektedir ve algoritmadaki döngü, her verinin uygun kümeye atanmasına kadar devam ettirilmektedir.

K-means algoritmasının en büyük eksikliği ideal k değerini tespit edememesi ve kullanıcıya bırakmasıdır. Bu nedenle başarılı bir kümeleme elde etmek için farklı k değerleri için deneme yanılma yönteminin uygulanması gerekmektedir. Diğer bir eksiklik ise, algoritmanın uçtaki verilere karşı olan duyarlılığıdır. Değeri bakımından

çok büyük olan bir nesne, dâhil olacağı kümenin ortalamasını ve merkez noktasını büyük bir derecede değiştirebilir. Bu değişiklik, kümenin hassasiyetini bozabilir.

K-means algoritmasına örnek olarak Tablo.15.'de $X = (x_1, x_2)$ şeklinde verilmiş değer çiftleri, bu algoritma kullanılarak k=2 kümeye ayrılmak istenmektedir.

Tablo.15. Değerler tablosu

Sıra No	x_1	x_2
1	4	3
2	5	4
3	6	6
4	7	5
5	8	6

Veriler arasından rastgele k=2 nokta küme merkezi olarak seçilir. Bu noktalardan ilk ikisi $M_A = (4,3)$ ve $M_B = (5,4)$ küme merkezi kabul edilsin.

Her verinin 2 küme merkezine olan uzaklıkları hesaplanır. Uzaklık hesaplaması için Öklid uzaklık formülü kullanılabilir. Her veri, kendisine merkezi yakın olan kümeye atanır (Tablo.16).

Tablo.16. Verileri kümelere 1. atama durumu

Sıra	x_1	x_2	Öklid Mesafesi d=(X, A)	Öklid Mesafesi d=(X, B)	Küme
1	4	3	$\sqrt{(4-4)^2+(3-3)^2} = 0$	$\sqrt{(4-5)^2+(3-4)^2} = \sqrt{2}$	A
2	5	4	$\sqrt{(5-4)^2+(4-3)^2} = \sqrt{2}$	$\sqrt{(5-5)^2+(4-4)^2} = 0$	B
3	6	6	$\sqrt{(6-4)^2+(6-3)^2} = \sqrt{13}$	$\sqrt{(6-5)^2+(6-4)^2} = \sqrt{5}$	B
4	7	5	$\sqrt{(7-4)^2+(5-3)^2} = \sqrt{13}$	$\sqrt{(7-5)^2+(5-4)^2} = \sqrt{5}$	B
5	8	6	$\sqrt{(8-4)^2+(6-3)^2} = 5$	$\sqrt{(8-5)^2+(6-4)^2} = \sqrt{13}$	B

Oluşan 2 küme de aşağıdaki hali alır:

A={(4,3)} ve B={(5,4) , (6,6) , (7,5) , (8,6)}

Yeni küme merkezleri hesaplanacak olursa;

$M_A = (4,3)$ ve $M_B = \left(\frac{5+6+7+8}{4}, \frac{4+6+5+6}{4}\right)$ = (6.5 , 5.25) elde edilir.

Her verinin yeni küme merkezlerine olan uzaklıkları hesaplanıp, kendisine daha yakın olan kümeye ataması yapılır (Tablo.17.).

Tablo.17. Verileri kümelere 2. atama durumu

Sıra	x_1	x_2	Öklid Mesafesi d=(X, A)	Öklid Mesafesi d=(X, B)	Küme
1	4	3	$\sqrt{(4-4)^2+(3-3)^2}=0$	$\sqrt{(4-6.5)^2+(3-5.25)^2} \cong \sqrt{11.3}$	A
2	5	4	$\sqrt{(5-4)^2+(4-3)^2}=\sqrt{2}$	$\sqrt{(5-6.5)^2+(4-5.25)^2} \cong \sqrt{3.8}$	A
3	6	6	$\sqrt{(6-4)^2+(6-3)^2}=\sqrt{13}$	$\sqrt{(6-6.5)^2+(6-5.25)^2} \cong \sqrt{0.8}$	B
4	7	5	$\sqrt{(7-4)^2+(5-3)^2}=\sqrt{13}$	$\sqrt{(7-6.5)^2+(5-5.25)^2} \cong \sqrt{0.3}$	B
5	8	6	$\sqrt{(8-4)^2+(6-3)^2}=5$	$\sqrt{(8-6.5)^2+(6-5.25)^2} \cong \sqrt{2.8}$	B

2 kümenin yeni hali aşağıdaki gibi olur:

A={(4,3) , (5,4)} ve B={(6,6) , (7,5) , (8,6)}

Küme merkezleri yeniden hesaplanacak olursa;

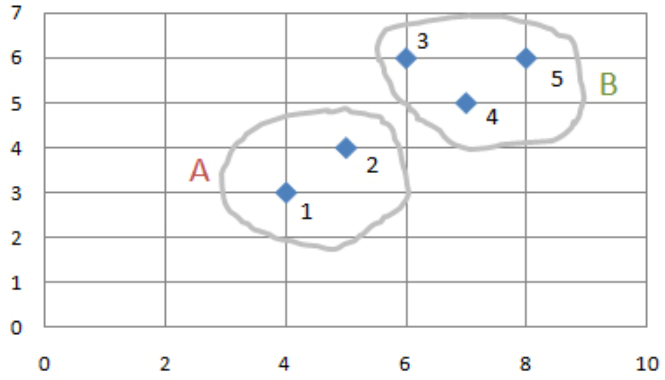
$M_A = \left(\frac{4+5}{2}, \frac{3+4}{2}\right)$ = (4.5 ,3.5) ve $M_B = \left(\frac{6+7+8}{3}, \frac{6+5+6}{3}\right)$ = (7 , 5.67) elde edilir.

2 numaralı noktanın, bir önceki duruma göre B kümesinden A kümesine geçiş yaptığı görülmüştür. Küme elemanlarında değişiklik olduğundan verilerin kendisine yakın olan küme atamalarının yapılması için Öklid uzaklıkları tekrar hesaplanır (Tablo.18).

Tablo.18. Verileri kümelere 3. atama durumu

Sıra	x_1	x_2	Öklid Mesafesi d=(X, A)	Öklid Mesafesi d=(X, B)	Küme
1	4	3	$\sqrt{(4-4.5)^2+(3-3.5)^2}=\sqrt{0.5}$	$\sqrt{(4-7)^2+(3-5.67)^2} \cong \sqrt{16.1}$	A
2	5	4	$\sqrt{(5-4.5)^2+(4-3.5)^2}=\sqrt{0.5}$	$\sqrt{(5-7)^2+(4-5.67)^2} \cong \sqrt{6.8}$	A
3	6	6	$\sqrt{(6-4.5)^2+(6-3.5)^2}=\sqrt{8.5}$	$\sqrt{(6-7)^2+(6-5.67)^2} \cong \sqrt{1.1}$	B
4	7	5	$\sqrt{(7-4.5)^2+(5-3.5)^2}=\sqrt{8.5}$	$\sqrt{(7-7)^2+(5-5.67)^2}=0.67$	B
5	8	6	$\sqrt{(8-4.5)^2+(6-3.5)^2}=\sqrt{18.5}$	$\sqrt{(8-7)^2+(6-5.67)^2} \cong \sqrt{1.1}$	B

Verilerin kendisine yakın olan kümeye ataması sonucunda hiçbir veride küme değişikliği olmadığından dolayı algoritma sonlandırılır ve verilerin kümelenmesi sonucu oluşan durum görsel olarak Şekil.16.'da verilmiştir.



Şekil.16. Verilerin K-means algoritmasına göre 2 kümeye ayrılması

2.2.2. Hiyerarşik Kümeleme Algoritmaları (Hierarchical Clustering Algorithm)

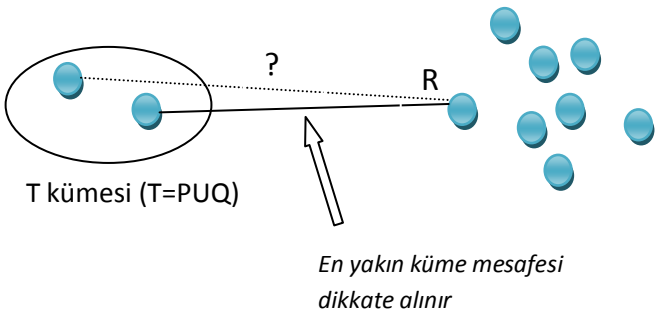
Hiyerarşik kümeleme algoritmaları, veriler arasındaki uzaklık bilgilerinden yararlanarak birleşme ya da bölünme kurallarının çıkarılmasını sağlayan algoritmalardır. Algoritma, başlangıçta bir veritabanındaki verilerin her birini bir küme olarak kabul eder ve aşama aşama birleştirerek belirli bir sıra ile tek bir küme elde edilmesini (birleştirici hiyerarşik kümeleme) sağlar. Bu işlemin tersi olarak da, verileri tek bir küme kabul ederek, yine belirli bir sıra ile bölünebilmesini (ayrıştırıcı hiyerarşik kümeleme) sağlar. Buradaki önemli nokta, hangi verinin hangi kümeye kaçıncı sırada dahil edildiğinin yada ayrıldığının bulunmasıdır. İşleyişin daha kolay anlaşılabilmesi için dendogram adı verilen ağaç grafiğinden yararlanılır[26]. Dendogram, hiyerarşik kümeleme tekniğiyle elde edilen sonuçların görselleştirilmesini sağlamaktadır.

Hiyerarşik kümeleme algoritmaları, benzerlik ve uzaklık ölçümü kullandıkları için gerçeklemesi kolay ve farklı veri tiplerinde de uygulama alanı geniş olan algoritmalardır[27]. Bunun yanı sıra, algoritmalar hızlıdır ve giriş parametresi olarak küme sayısının verilmesine gerek yoktur ki özellikle ayrıştırıcı hiyerarşik kümeleme algoritmaları için küme sayısının verilmesi dezavantaj oluşturmaktadır. Bu algoritmalarda, kümeleme işlemi yapıldıktan sonra birleştirme ve ayrıştırma işlemleri kontrol edilmemektedir.

2.2.2.1. En Yakın Komşu Algoritması (Single Linkage Clustering Algorithm)

En yakın komşu algoritması, hiyerarşik kümeleme analizleri içerisinde en yaygın olarak kullanılanlardandır. n verisi bulunan bir küme için, algoritma işleyişi aşağıdaki şekildedir:

1. Veri setindeki her bir veri ayrı bir küme olarak kabul edilir.
2. Her bir kümenin, diğerlerine olan uzaklıklarından oluşan n*n kare matrisi oluşturulur.
3. Oluşturulan matriste en yakın küme çiftleri tespit edilir ve birleştirilir.
4. *Kümeler arasındaki uzaklık tespiti:* Birleştirilen küme ile diğer kümelerin arasındaki uzaklıklardan oluşan kare matris, birleştirilen kümelere göre yeniden düzenlenir. Bu durumdaki uzaklıklar, birleştirilen kümeler ile diğer kümelerin birbirine en yakın noktalarının mesafesidir.
5. Üçüncü ve dördüncü adımlar n-1 kez tekrar edilir.



Şekil.17. En yakın komşu algoritmasında kümeler arası uzaklık tespiti

Algoritmanın üçüncü adımında en yakın kümeler Şekil.17.'den görüleceği gibi P ve Q kümesi olarak tespit edilsin. T kümesinin de P kümesi ile Q kümesinin birleştirildikten sonraki oluşan yeni küme olduğu kabul edilirse, herhangi bir R kümesi ile T kümesi arasındaki uzaklık, T ile P kümesine göre mi yoksa T ile Q kümesine göre mi hesaplanması gerekmektedir sorusuna yanıt şu şekildedir: Uzaklık d_{TR} ile gösterildiğinde, T kümesi ile R kümesinin birbirine en yakın olan iki değeri arasındaki uzaklık $d_{TR}=\min(d_{PR}, d_{QR})$ olur[28].

En yakın komşu algoritmasına örnek olarak kümelenmesi istenilen 6 şehir ve bu şehirlerin birbirleri arasındaki uzaklıklar km cinsinden Tablo.19.a.'da verilmiştir.

Tablo.19.a. Şehirler ve şehirlerarası uzaklıklar

Şehirler	A	B	C	D	E	F
A	0	-	-	-	-	-
B	662	0	-	-	-	-
C	877	295	0	-	-	-
D	255	468	754	0	-	-
E	412	268	564	219	0	-
F	996	400	138	869	669	0

Tabloda 6 şehrin birbirine göre uzaklık matrisi verilmiştir. Her bir şehrin, bir kümeyi temsil ettiği varsayılarak öncelikle en yakın iki küme tespit edilir. C ve F kümesi 138 değeri ile en yakın iki kümedir. Tablo, C ve F kümeleri birleştirilerek yeniden düzenlenir. Algoritma 4. adımına göre dikkat edilmesi gereken husus, uzaklıkların birleştirilen kümeler ile diğer kümelerin birbirine en yakın noktalarının uzaklıkları olduğudur. Birleştirilen küme C-F kümesidir ve diğer kümeler ile arasındaki uzaklık, bu kümeyi oluşturan C ya da F kümesinden hangisi yakınsa o kümeye göre olan uzaklıktır. Örneğin, A kümesi ile C-F kümesi arasındaki uzaklıklar |AC|=877 ve |AF|=996 dir. A kümesi ile C kümesi daha yakın olduğundan dolayı bu uzaklık, A ve C-F kümesi arası uzaklık olarak belirlenir. Diğer uzaklıklar da bu şekilde belirlenir ve Tablo.19.b.'deki durum elde edilir.

Tablo.19.b.

Şehirler	A	B	C-F	D	E
A	0	-	-	-	-
B	662	0	-	-	-
C-F	877	295	0	-	-
D	255	468	754	0	-
E	412	268	564	219	0

Tablo.19.c.

Şehirler	A	B	C-F	D-E
A	0	-	-	-
B	662	0	-	-
C-F	877	295	0	-
D-E	255	268	564	0

Algoritmanın 3. ve 4. adımına tekrar dönülür. En yakın kümeler 219 ile D ve E kümeleridir. Bu kümeler birleştirilerek matris yeniden güncellenir ve Tablo.19.c.'deki durum elde edilir.

Güncellenmiş değerlere göre en yakın kümeler 255 ile A ve D-E kümeleridir. Dolayısı ile bu kümeler birleştirilerek matriste güncelleme yapılarak Tablo.19.d.'deki duruma gelinir.

Tablo.19.d.

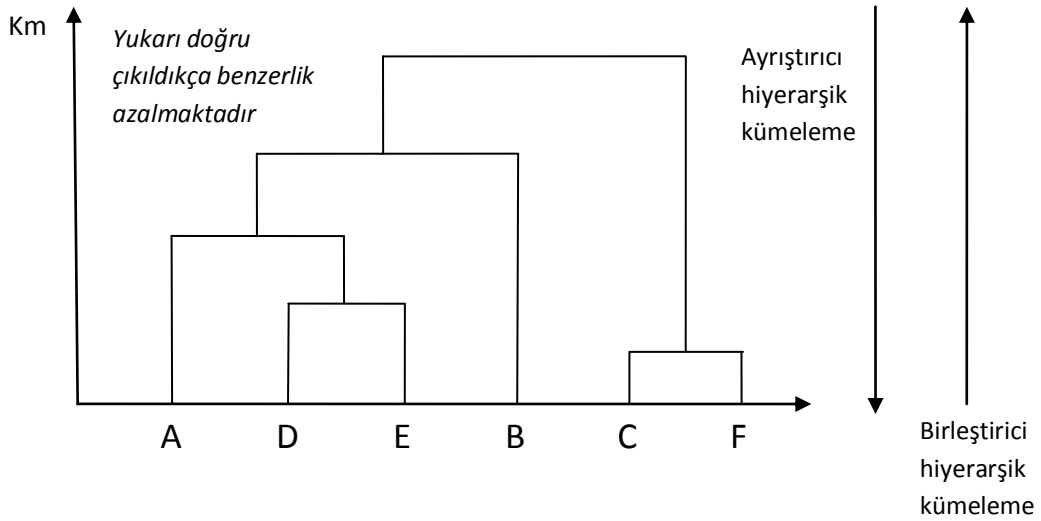
Şehirler	A-D-E	B	C-F
A-D-E	0	-	-
B	268	0	-
C-F	564	295	0

Tablo.19.e.

Şehirler	A-B-D-E	C-F
A-B-D-E	0	-
C-F	295	0

Bu değerlere göre en yakın iki küme 268 değeri ile A-D-E ve B kümesidir. Bu kümeler birleştirilerek Tablo.19.e.'deki durum elde edilir ve yeni uzaklıklar belirlenir. Bu iki kümede birleştirilirse 6 küme aşama aşama birleştirilerek tek bir küme (A-B-C-D-E-F kümesi) elde edilmiş olunur ve algoritma sonlandırılır.

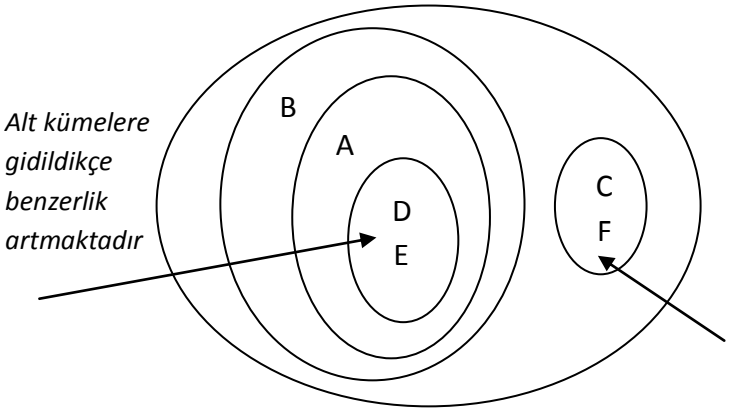
Aşağıda benzerlik ölçütü olarak aralarındaki uzaklık alınan 6 şehrin benzerlik hiyerarşisi dendogramda verilmiştir. Görüldüğü gibi, ilk benzeşmeyi gösteren küme çifti, yani birbirine en çok benzeyen C ve F şehirleridir. Daha sonra bu benzeşmeden bir miktar azını D ve E şehirleri göstermiştir. Ardından bu kümeye gittikçe daha az benzeyen sırasıyla A ve B kümeleri dahil olmuştur. Şekil.18.'deki verilen dendogram ile kümelerin belirli bir sıra ile benzeşme sıralaması da görülmektedir.



Şekil.18. Şehirlerarası mesafe için ortaya çıkan dendogram

Hiyerarşik yapı oluşturulma ve kullanılma amacına göre iki yönteme ayrılmaktadır. Yapı oluşturulurken, dendogramın tepe noktasından birimlere doğru iniliyor ise bu yönteme ayrıştırıcı hiyerarşik kümeleme yöntemi adı verilir. Bu durumun tersinde ise, yani kümeleme işlemi yapılırken her bir birim ayrı bir küme

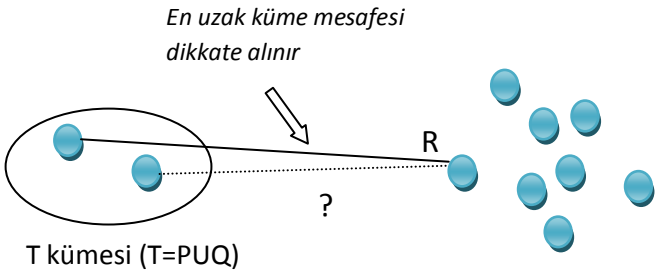
olarak düşünülüp ana küme elde ediliyorsa bu yönteme birleştirici hiyerarşik kümeleme yöntemi adı verilir.



Şekil.19. Verilerin küme olarak gösterimi

2.1.2.2. En Uzak Komşu Algoritması (Complete-linkage Clustering Algorithm)

En uzak komşu algoritması, işleyişi ve kolaylığı bakımından en yakın komşu algoritmasına benzemektedir. Tek farkı, algoritmanın dördüncü adımındaki birleştirilmiş bir küme ile başka bir küme arasındaki uzaklığın belirlenmesidir. Bu algoritmada, en yakın mesafe yerine en uzak mesafe dikkate alınır.



Şekil.20. En uzak komşu algoritmasında kümeler arası uzaklık tespiti

T kümesi, P kümesi ile Q kümesi birleştirildikten sonra oluşan yeni küme olsun. Herhangi bir R kümesi ile T kümesi arasındaki uzaklık şu şekildedir: Uzaklık d_{TR} ile gösterildiğinde, T kümesi ile R kümesinin birbirine en uzak olan iki değeri arasındaki uzaklık $d_{TR}=\max(d_{PR}, d_{QR})$ olur (Şekil.20.) [29].

$X = (x_1, x_2)$ şeklinde Tablo.20.a.'da verilmiş değer çiftleri en uzak komşu algoritması kullanılarak örnek bir hiyerarşik kümeleme analizi aşağıda yapılmaktadır.

Tablo.20.a. Örnek Veriler

	x_1	x_2
A	3	5
B	7	4
C	2	7
D	6	5
E	4	1

Tablo.20.b. Uzaklık matrisi

	A	**B**	**C**	**D**	**E**
A	0	-	-	-	-
B	4.12	0	-	-	-
C	2.23	5.83	0	-	-
D	3	1.41	4.47	0	-
E	4.12	4.24	6.32	4.47	0

Başlangıçta her veri çifti bir küme olarak kabul edilir. Her kümenin birbiri ile arasındaki uzaklık değerlerinden oluşan matris oluşturulur. Kümeler arasındaki uzaklık ölçümü için çeşitli formüller vardır. Burada Öklid uzaklık formülü kullanılacaktır. A kümesi ile B kümesi arasındaki uzaklık;

$d_{AB}=\sqrt{(7-3)^2+(4-5)^2}=4.12$'dir.

Benzer şekilde her kümenin diğer kümeler ile arasındaki uzaklıklar hesaplandığında Tablo.20.b. elde edilir.

Öncelikle, en yakın iki küme araştırılır. Bu kümeler, 1.41 değeri ile B ve D kümesidir. Bu iki küme birleştirilir ve uzaklık matrisi güncellenir (Tablo.21.a). Birleştirilen B-D kümesi ile diğer kümelerin arasındaki uzaklığın maksimum değeri iki küme arasındaki uzaklık olarak belirlenir. Örnek olarak; A ve B-D kümesi arasındaki uzaklık $d_{A,B\text{-}D}$=max(4.12, 3)=4,12 olarak belirlenmiş olur.

Tablo.21. Uzaklık matrisleri

a.

	A	**B-D**	**C**	**E**
A	0	-	-	-
B-D	4.12	0	-	-
C	2.23	5.83	0	-
E	4.12	4.47	6.32	0

b.

	A-C	**B-D**	**E**
A-C	0	-	-
B-D	5.83	0	-
E	6.32	4.47	0

c.

	A-C	**B-D-E**
A-C	0	-
B-D-E	6.32	0

Tablo.21.a.'da en yakın iki küme 2.23 değeri ile A ve C kümesi olarak belirlenir. Bu kümeler birleştirildikten sonra uzaklık matrisi yeniden güncellenir ve Tablo.21.b. elde edilir. Bu aşamada, tekrar en yakın iki küme tespit edildikten sonra birleştirilir. Bu iki küme, 4.47 uzaklık değeri ile E ve B-D kümeleridir. Bu kümelerde birleştirildikten sonra uzaklık matrisinin son hali Tablo.21.c.'deki gibi olur. Bu durumdaki mevcut 2 kümede birleştirilir ve tek bir küme elde edilerek, algoritma sonlandırılır.

Hiyerarşik kümeleme analizi sonuçları dendogramla gösterilebileceği gibi şu şekilde ifade edilebilir:

Sonuç küme ={{A,C},{{B,D},E}}

2.3. Yarı Gözetimli Öğrenme (Semi-Supervised Learning)

Gözetimli ve gözetimsiz öğrenme algoritmaları genellikle yalnız başlarına kullanılsa da, birlikte kullanıldığı alanlarda mevcuttur. Gözetimli öğrenmede, sınıfları belirli olan girdi verileri ile çıktı verileri arasında bir fonksiyon üretilmekte ve girdi-çıktı ilişkisi ortaya çıkarılmaktadır. Gözetimsiz öğrenmede ise sınıfları belli olmayan girdi verilerinin benzerliklerine göre ayrı ayrı kümelenebilmesi mümkündür. Yarı gözetimli öğrenme her iki yönteminde bu avantajının kullanıldığı bir öğrenme türüdür. Etiketlendirilmemiş çok veriyi, etiketlemek hem maliyetli hem de zaman gerektiren bir işlem olabilir. Bu gibi durumlarda etiketlenmemiş çok verinin yanında, yakın ya da daha az sayıda etiketlenmiş veride sisteme girdi olarak verilirse, tüm girdinin etiketlenmesini olanaklı kılar. Yarı gözetimli öğrenme yöntemleri, biyoinformatik, metin işleme, video indeksleme gibi alanlarda kullanılmaktadır[30].

2.4. Destekleyici Öğrenme (Reinforcement Learning)

Destekleyici öğrenme, gözetimli öğrenmenin farklı bir türüdür. Öğrenme gerçekleştirilirken, sisteme girdi kümesi verilir, ancak çıktı kümesi verilmez. Bunun yanında sistemin öğrenmeyi gerçekleştirmesinde sisteme destek sağlayan bir yapı mevcuttur. Sistem, girdi verilerine karşılık olarak bir çıkış verisi üretir. Sisteme, ürettiği çıkışın doğru olup olmamasına göre destekçi pozitif ya da negatif yönde bir geri dönüş sağlanır. İstenilen (doğru) sonuç üretildiğinde sisteme pozitif destek sağlanır. Hangi durumlarda hangi eylemin gerçekleşeceğinin bilgisi bu şekilde elde edilmiş olur. Yöntem, bu yönüyle hayvanların öğrenme şeklini model alan bir yöntemdir. Sistem, geri besleme ile gelen sinyaller doğrultusunda, girdi verileriyle hem öğrenerek hem de sonuç çıkararak öğrenmeye devam etmektedir. Yani çıkış değerleri verilmediği için doğrudan öğrenme yerine girişe göre çıkan bilgiler ile değerlendirilerek öğrenme sağlanır. Bu bakımdan geri besleme şeklinde gelen bu sinyaller öğretici değil değerlendiricidir[31]. Bu öğrenme türüne LVQ ağı, Q-Learning algoritması ve Temporal difference learning algoritması örnek olarak verilebilir. Destekleyici öğrenme, satranç, tavla gibi oyunlar ile robot kontrolü gibi alanlarda tercih edilmektedir.

3. YÖNTEM

3.1. Web Siteleri Tıklamaları Analizi

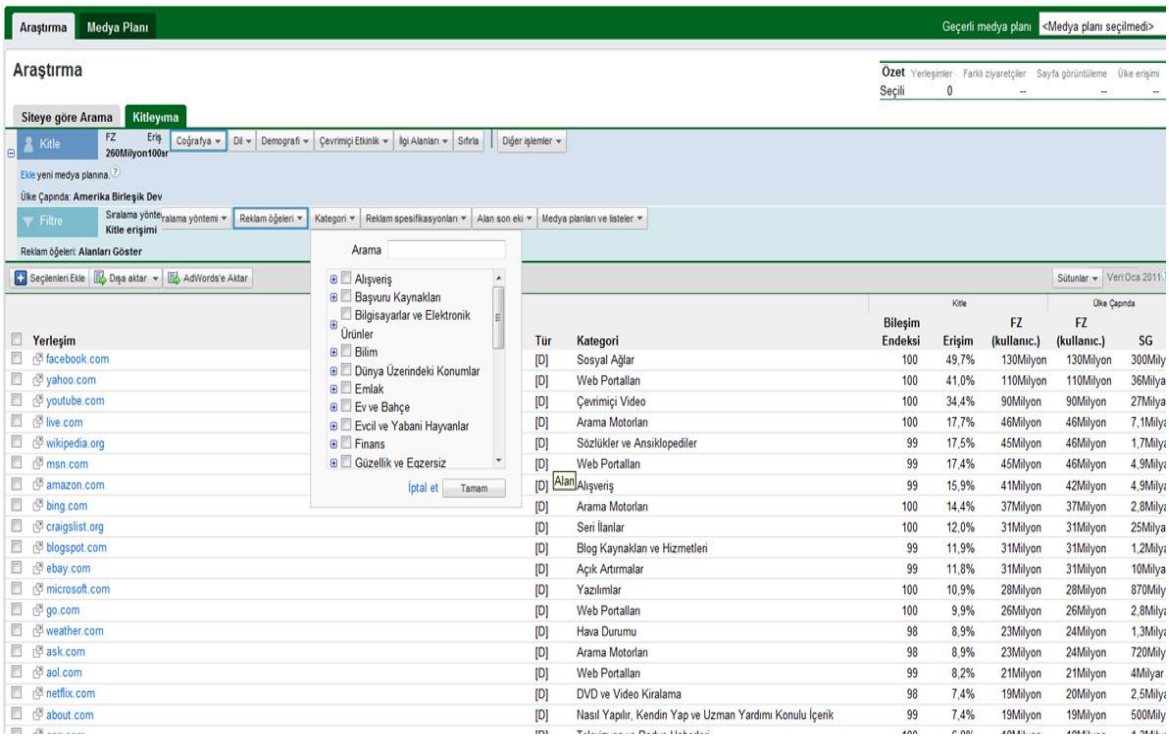
Bu uygulamada, web trafiği analizlerinde kullanılan bazı parametrelerin, makine öğrenmesi algoritmaları ve WEKA yazılımı yardımı ile birbirleri arasındaki ilişkiler ortaya çıkarılmış ve makine öğrenmesi algoritmalarının sonuçları, başarısı ve performansları değerlendirilmiştir. Parametreler arasında ilişkilerin belirlenmesi için Google hizmeti olan DoubleClick Ad Planner verilerinden yararlanılmıştır. 2011 yılının (01-31) Ocak ayındaki Türkiye'deki web siteleri trafiğinde yer alan ilk 1000 web sitesi parametreleri kullanılmış ve çıkarımlar bulunmaya çalışılmıştır. DoubleClick Ad Planner'in verileri incelendiğinde 1 aylık bu dönemde Türkiye'de gerçekleşen bu istatistikler ile gerçek farklı ziyaretçi sayısı yaklaşık 21 milyon kişidir.

Verilerin elde edildiği DoubleClick Ad Planner, reklam verenler ve reklam yayıncılarının, reklamcılıkla ilgili daha bilinçli kararlar verebilmesine katkıda bulunan ve ticari olarak hedef kitlenin ziyaret etme olasılığı yüksek web sitelerini saptanmasına yardımcı olabilen ücretsiz bir medya planlama aracıdır [32]. Diğer bir yönüyle, hem kişisel web siteleri hem de kurumsal siteler olsun, hem de reklam alan veya reklam almayan web siteleri hakkında detaylı bilgiler verebilen ve sunduğu bilgilerin incelenip analiz edilmesini sağlayan bir Google servisidir.

DoubleClick Ad Planner verileri aylık olarak yayınlanmaktadır ve bu veriler yaklaşık değerlerdir. Verilerin elde edilmesi, milyonlarca arama sorgusu ve site ziyaretlerinin otomatik analizine dayanmaktadır. Çeşitli kaynaklardan aktarılan veriler, milyonlarca kullanıcıdan toplanır ve bilgisayar algoritmalarıyla desteklenir. Veriler tam olarak kesin değerler olmasa da, yönlendirici bir şekilde kullanmak ve geçerli planlama önerileri elde etmek açısından yeterince kesindir[33]. DoubleClick Ad Planner tarafından yayınlanan listede, sadece yetişkinlere yönelik siteler, herkesin görebileceği içeriğe sahip olmayan veya düzgün yüklenmeyen alanlar ile Google'a ait web siteleri bulunmamaktadır.

3.2. Çalışmada Kullanılan Verilerin Elde Edilmesi

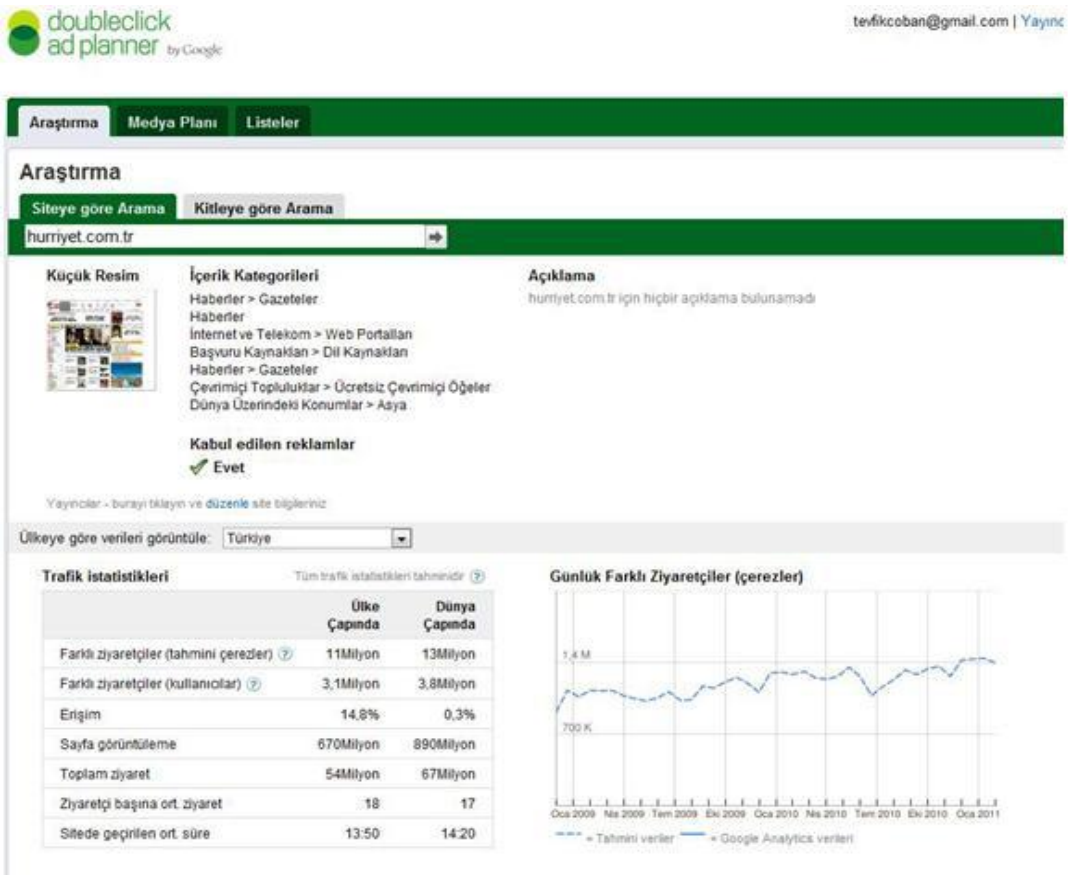
DoubleClick Ad Planner'da 'Araştırma' üst menüsü yardımıyla çeşitli filtrelerde kullanarak istenilen düzeyde web sitesi verilerine ulaşılabilmektedir. Şekil.21.'de DoubleClick Ad Planner'e ait 'Kitleye göre arama' ara yüzü ile web sitelerine ait verilerin yer aldığı liste verilmiştir.



Şekil.21. DoubleClick Ad Planner'in web istatistikleri listesi

Kitle seçeneklerinde yer alan 'Coğrafya' menüsünde aralarında Türkiye'nin de bulunduğu 58 ayrı ülke ya da tüm ülkeler veri edilmek istenilen yer olarak seçilebilmektedir. Çalışmada, Türkiye'ye ait veriler kullanılarak Türkiye'deki internet kullanıcıları ve web sitesi trafiği dikkate alınmıştır. Filtre seçeneklerinde yer alan 'reklam kabul ediyor' filtresi işaretlenmediğinden tüm web sitelerini inceleme imkânı olmuştur.

Listede yer alan 1000 web sitesi için ayrı ayrı site profili incelenerek gerekli veriler için kayıt tutularak veritabanı oluşturulmuştur. Şekil.22.'de bir web sitesi (hurriyet.com.tr) için site profili görüntüsü verilmiştir.



Şekil.22. DoubleClick Ad Planner'de bir web sitesi profili

Türkiye'deki ziyaretçi sayısı en fazla olan ilk 1000 siteye ait 6 adet farklı niteliğe ait veriler DoubleClick Ad Planner ile elde edilmiştir. Bu nitelikler Tablo.22.'de verilmektedir.

Tablo.22. Google Ad Planner ile elde edilen web sitesi verilerine ait nitelikler

	Web sitesi adı
1	Site kategorisi
2	Erişim (%)
3	Farklı ziyaretçiler (tahmini çerezler) sayısı
4	Sayfa görüntüleme sayısı
5	Sitede geçirilen ortalama süre
6	Ortalama ziyaret sayısı

Site kategorisi, web sitesinin yönelik olduğu konu ya da verdiği hizmetlere dönük olarak sitenin ait olduğu türü belirtmektedir.

Erişim (%), belirli bir ay boyunca, belirlenen ülke ya da bölge için tahmini toplam internet kullanıcısı yüzdesini ifade eder.

Farklı ziyaretçiler (tahmini çerezler), belirli bir ay boyunca bir sitedeki, DoubleClick Ad Planner'ın algoritmaları tarafından belirlenen yaklaşık çerez sayısıdır. Farklı ziyaretçiler (tahmini çerezler) değeri, gerçek farklı ziyaretçi sayısı değildir ancak web sitesi tıklama analizlerinde kıyaslama yapılarak kullanılacağından yararlı olabilecek bir parametredir. Gerçek farklı ziyaretçi değeri olmamasının birkaç sebebi vardır. Çünkü çerezler kendiliğinden geçici süreli olabilir veya kullanıcı çerezi kendisi silebilir. Bu durumda aynı IP adresinden web sitesine giren bir kullanıcı tekrar çerez almış olur. Ya da kullanıcılar, birden fazla web tarayıcısı kullanıyorsa her web tarayıcısı için ayrı ayrı çerez alabilir.

Sayfa görüntüleme sayısı, tüm kullanıcıların belirli bir ay boyunca, bir sitedeki sayfaları toplam görüntüleme sayısıdır.

Sitede geçirilen ortalama süre, ortalama olarak her ziyaretçinin sitede geçirdiği süredir.

Ortalama ziyaret sayısı, web sitesi ziyaretçileri başına düşen ziyaret sayısını ifade etmektedir. Burada dikkat edilmesi gereken nokta, eğer bir kullanıcı sitede 30 dakika veya daha fazla hiçbir işlem yapmazsa, sonraki etkinlikler yeni bir ziyaret olarak değerlendirilmesidir[32].

3.3. Çalışmada Kullanılan Verilerin Dönüşümü

Çalışmanın altı farklı niteliğine ilişkin sürekli verilerdeki değişimler üç farklı sınıfta, kategorik verilerdeki benzerlikler ise dönüştürme işlemleri yapılarak 20 farklı sınıfta gruplanmıştır.

Site kategorisi: DoubleClick Ad Planner ile elde edilen 1000 web sitesi için toplamda 200'ün üzerinde site kategorisi bulunmaktadır. Hem sınıflandırma problemlerinin yaşanmaması açısından hem de birbiri ile çok yakın ve ilişkili kategorilerin farklı bir sınıf gibi davranmasını engellemek amacı ile gruplanarak 20 ana web sitesi türü elde edilmiştir. Bu site türleri ve temsil edilen kategoriler Tablo.23.'de verilmektedir.

Tablo.23. Web sitesi kategorilerinin veri dönüşümü

No	Kategoriler (Veri dönüşümünden önce)	Web Sitesi Türü (Veri dönüşümünden sonra)	Veri Sayısı
1	Bilim Kurumları, Devlet, Devlet Arşivleri, Hukuk, Hukuk ve Devlet Hizmetleri, Kamu Güvenliği, Kamu Maliyesi, Yönetim, Yüksekokullar ve Üniversiteler, Eyalet Yönetimi ve Yerel Yönetim	Resmi Kurum	43
2	Web Portalları , Arama Motorları *(*gerçekte arama motoru da içeren bazı internet portalları, Arama Motoru kategorisi adıyla kayıtlı olduklarından İnternet Portalına dahil edilmiştir.)*	İnternet Portalı	26
3	Açık artırmalar, Alışveriş, Alışveriş Portalları ve Arama Motorları, Bilet Satışları, Cinsel Geliştirme Ürün ve Operasyonları, DVD ve Video Alışverişi, Fiyat Karşılaştırmaları, İthalat ve İhracat, Kuponlar ve İndirim Teklifleri, Pazarlama Hizmetleri, Ticari ve Endüstriyel Hizmetler, Tüketici Elektronik Ürünleri, Tüketici İlişkileri, Tüketici İlişkileri ve Ürün İncelemeleri, Toptancılar ve Tasfiye Edilen Mal Satıcıları	e-Ticaret	71
4	Araç Alım-Satım, Emlak, Gayrimenkul İlanları, Konut ve Arazi Geliştirme, Otomobiller ve Araçlar,Seri İlanlar, İş İlanları,İstihdam ve Personel Alımı,İşletme İlanları ve Kişisel İlanlar,İnsan Kaynakları	İlanlar	32
5	Basit Oyunlar, Bilgisayar ve Video Oyunları, Çevrimiçi Oyunlar, Devasa Çok Oyunculu Oyunlar, Kağıt Oyunları, Nişan Oyunları, Oyuncak Bebek Giydirme ve Kız Oyunları, Oyunlar, Spor Oyunları, Sürüş ve Yarış Oyunları,Mizah	Oyun	90
6	Ağ Güvenliği, Ağ Oluşturma, Bilgisayar Güvenliği, Donanım, İnternet ve Telekom, Mobil Uygulamalar ve Eklentiler, Mühendislik ve Teknoloji, Proxy ve Filtre Kullanımı, Teknoloji Haberleri, Web Tasarımı ve Geliştirme, Windows OS, Web Hizmetleri, Bilgisayarlar ve Elektronik Ürünler	Bilişim - Teknoloji	30
7	Eğitim, İlk ve Ortaöğretim (K-12), Okul Öncesi Eğitim, Test Hazırlığı	Eğitim	30
8	Demografi, Kişi Arama, Sosyal Ağlar, Blog Kaynakları ve Hizmetleri	Sosyal Paylaşım	24
9	Amerikan Futbolu, Basketbol, Spor ve maç sonuçları, Futbol, Spor, Spor Haberleri	Spor	29
10	Bankacılık, Emtia ve Vadeli İşlemler Ticareti, Finans, Kredi Kartları, Kredi ve Borç, Muhasebe ve Teftiş, Para Birimleri ve Döviz, Ticaret Hizmetleri ve Ödeme Sistemleri, Yatırım, Bordro Hizmetleri	Ekonomi - Bankacılık	32
11	Forum ve Sohbet Sağlayıcıları	Forumlar	38
12	Dosya Paylaşımı ve Barındırma, Fotoğraf ve Görüntü Paylaşımı, Fotoğraf ve Video Yazılımları, İnternet İstemcileri ve Tarayıcıları, İnternet Yazılımları, Java, Ortam Yürütücüleri, Ücretsiz Çevrimiçi Öğeler, Ücretsiz ve Paylaşımlı Yazılımlar, Virüsten Koruma Yazılımları ve Kötü Amaçlı Yazılımlar, VOIP ve İnternet Telefonculuğu, Yazılım Yardımcı Programları, Yazılımlar, Yüzey Temaları ve Duvar Kağıtları, Küçük Resim ve Animasyonlu GIF'ler	Dosya Paylaşımı	71
13	Cinsel Eğitim ve Danışmanlık, Ereksiyon Sorunları, Kilo Verme, Sağlık, Sağlık Haberleri, Sağlık Sigortası, Sağlık Sorunları, Tıbbi Müdahaleler, Yüz ve Vücut Bakım	Sağlık	18
14	Arkadaş İlanları ve Kişisel İlanlar, Çevrimiçi Topluluklar, E-Posta ve Mesajlaşma, Romantik İlişkiler	Arkadaşlık - Sohbet	25
15	Başvuru Kaynakları, Bilim, Çeviri Araçları ve Kaynakları, Dil Kaynakları, Din ve İnanç, Etnik Gruplar ve Kimlik Grupları,	Bilgi Kaynağı	63

	Geçmiş, Google Haritalar, İlaçlar, İslam, Kitaplar ve Edebiyat, Okült ve Doğaüstü Olaylar, Sözlükler ve Ansiklopediler, Hava Durumu, Emeklilik, Vize ve Göçmenlik,İleri Yaştakiler ve Emeklilik,Kariyer Kaynakları ve Planlama,Sosyal Hizmetler		
16	Dedikodu ve Magazin Haberleri, Dünya Haberleri, Gazeteler, Haberler, Siyaset, Televizyon ve Radyo Haberleri, Yerel Haberler	Gazete ve Haber	67
17	Çizgi Filmler, Film Referansları, Filmler, Radyo, Televizyon Kanalları, Televizyon Pembe Dizileri, Televizyon Suç ve Adalet Dizileri, TV Dizileri, TV Şovları ve Programları, TV ve Video	Tv - Sinema	86
18	Çevrimiçi Video, Fotoğraf ve Video Paylaşımı, Müzik Yayınları ve İndirilebilir Öğeler, Video Paylaşımı, Müzik ve Ses	Video ve Müzik Paylaşımı	77
19	Araç Ruhsatı ve Tescili, Araç Tekerlek ve Lastikleri, Arama Motoru Optimizasyonu ve Pazarlaması, Ayakkabılar, Beslenme, Cep Telefonları, Dizüstü Bilgisayarlar, Eczacılık, Ev Aletleri, Ev Mobilyaları, Fiat, Ford, Hastaneler ve Tedavi Merkezleri, Hava Yolculuğu, İnşaat Malzemeleri ve Gereçleri, İSS'ler, Kablo ve Uydu Sağlayıcıları, Kuryeler ve Posta Hizmetleri, Mac OS, Makyaj ve Kozmetik Ürünleri, Mobil ve Kablosuz, Multimedya Yazılımları,Nissan-Infiniti, Oteller ve Konaklama Hizmetleri, Perakende Ticaret, Raylı Taşımacılık, Reklamcılık ve Pazarlama, Renault-Samsung, Restoranlar,Servis Sağlayıcılar, Taşımacılık ve Lojistik, Toptancılar ve Büyük Mağazalar, Toyota, Vauxhall-Opel, Volkswagen, Web Barındırma Hizmeti ve Alan Adı Tescili	İşletmeler	69
20	Astroloji ve Kehanet, Aşçılık ve Yemek Tarifleri, Dans Müziği ve Elektronik Müzik, El Sanatları, Fotografik ve Dijital Sanatlar, Güzellik ve Egzersiz, Hamilelik ve Annelik, Ebeveynlik, Kadınlara Özgü İlgi Alanları, Kendi Kendine Yardım ve Motivasyon, Kulüpler ve Gece Hayatı, Nasıl Yapılır, Kendin Yap ve Uzman Yardımı Konulu İçerik, Sanat ve Eğlence,Seyahat, Stok Fotoğrafçılığı, Şans Oyunları, Şarkı Sözleri ve Notaları, Şiir, Turistik Yerler, Ünlüler ve Eğlence Haberleri, Yiyecek ve İçecek, Sanat ve Eğlence	Kişisel ilgi-beceri alanları	79

Bu veri dönüştürme işlemi ile çok sayıdaki kategorileri olan web siteleri, yönelik olduğu konu ve işlevleri dikkate alınarak benzerliklerine göre birleştirilmiştir. Bu nitelik, WEKA'da 20 farklı sınıf bazında nominal veri olarak değerlendirilecektir.

Erişim: Bu nitelik bir yüzde ifadesidir ve sürekli değişken veri özelliğindedir. Değerlendirmeye alınan tarih aralığı içinde (Ocak-2011) Türkiye'deki gerçek farklı ziyaretçilerin değerinin (21 Milyon kişi) yüzde olarak kaçına erişildiğini göstermektedir. Örneğin, % 26'lık bir oranla ülkedeki, internet kullanıcılarının % 26'sına ulaşılabiliyor anlamına gelir. (Türkiye'deki Ocak 2011 tarihli istatistikler dikkate alındığında % 26'lık erişim oranına sahip bir web sitesinin gerçek farklı ziyaretçi sayısı 0,26. 21 Milyon= 5,46 Milyon kişi olarak elde edilebilir.)

Erişim niteliğine ait veri dönüşümünden önceki basit istatistik Tablo.24.'de verilmektedir:

Tablo.24. Tüm web sitelerine ait erişim istatistiği

Erişim	
Maksimum değer	74,8
Minimum değer	0,4
Ortalama değer	1,70
Standart sapma	3,64

Erişim niteliğine ait veriler, Tablo.25.'de verilen aralıklarla 3 sınıfa ayrılmıştır.

Tablo.25. Erişim niteliğinin veri dönüşümü

Erişim (veri dönüşümünden önce)	Erişim (veri dönüşümünden sonra)	Veri sayısı
Erişim ≥ 3	3+	107
3 > Erişim ≥ 1	1-3	326
1 > Erişim	1-	567

Farklı ziyaretçiler (tahmini çerezler): Bu niteliğe ait verilerde işlem kolaylığı olması açısından sıfırlar (6 adet sıfır rakamı) atılmıştır. Değerler milyon seviyesinde olup, sürekli değişken verilerdir.

Farklı ziyaretçiler (tahmini çerezler) niteliğine ait veri dönüşümünden önceki basit istatistik Tablo.26.'da verilmektedir:

Tablo.26. Tüm web sitelerine ait farklı ziyaretçiler (tahmini çerezler) istatistiği

Farklı ziyaretçiler (tahmini çerezler)	
Maksimum değer	60
Minimum değer	0,24
Ortalama değer	1,32
Standart sapma	2,90

Farklı ziyaretçiler (tahmini çerezler) niteliğine ait veriler, Tablo.27.'de verilen aralıklarla 3 sınıfa ayrılmıştır.

Tablo.27. Farklı ziyaretçiler (tahmini çerezler) niteliğinin veri dönüşümü

Farklı ziyaretçiler (tahmini çerezler) (veri dönüşümünden önce)	Farklı ziyaretçiler (tahmini çerezler) (veri dönüşümünden sonra)	Veri sayısı
Farklı ziyaretçiler ≥ 3	3+	13
3 > Farklı ziyaretçiler ≥ 1	1-3	53
1 > Farklı ziyaretçiler	1-	934

Sayfa görüntüleme sayısı: Bu niteliğe ait verilerde de işlem kolaylığı olması için sıfırlar (6 adet sıfır rakamı) atılmıştır. Değerler milyon seviyesinde olup, sürekli değişken verilerdir.

Sayfa görüntüleme sayısı niteliğine ait veri dönüşümünden önceki basit istatistik Tablo.28.'de verilmektedir:

Tablo.28. Tüm web sitelerine ait sayfa görüntüleme sayısı istatistiği

Sayfa görüntüleme sayısı	
Maksimum değer	40000
Minimum değer	0,42
Ortalama değer	59,80
Standart sapma	1266,97

Sayfa görüntüleme sayısı niteliğine ait veriler, Tablo.29'daki verilen aralıklarla 3 sınıfa ayrılmıştır.

Tablo.29. Sayfa görüntüleme sayısı niteliğinin veri dönüşümü

Sayfa görüntüleme sayısı (veri dönüşümünden önce)	Sayfa görüntüleme sayısı (veri dönüşümünden sonra)	Veri sayısı
Sayfa görüntüleme sayısı ≥ 100	100+	31
100 > Sayfa görüntüleme sayısı ≥ 50	50-100	368
50 > Sayfa görüntüleme sayısı	50-	601

Sitede geçirilen ortalama süre: Bu çalışmada, sınıflandırma algoritmaları için seçilen 2 hedef nitelikten birisidir ve çalışmanın bundan sonraki bölümünde kısaca "ortalama süre" olarak geçecektir. DoubleClick Ad Planner'da bu niteliğe ait veriler, dakika ve saniye cinsinden verilmektedir. İki farklı birime ait olan bu değer WEKA'da işlem yapılabilmesi için tek birime (dakika) dönüştürülmüştür. Örneğin Şekil.22.'de ortalama sürenin 13:50 olduğu görülmektedir. 13 dakika 50 saniye

anlamına gelen bu veri 13+50/60=13,83 dakika olarak tek birime dönüştürülmüştür. Bu niteliğe ait veriler sürekli değişken verilerdir.

Ortalama süre niteliğine ait veri dönüşümünden önceki basit istatistik Tablo.30.'de verilmektedir:

Tablo.30. Tüm web sitelerine ait ortalama süre istatistiği

Ortalama süre	
Maksimum değer	30
Minimum değer	0,55
Ortalama değer	6,86
Standart sapma	4,53

Ortalama süre niteliğine ait veriler, Tablo.31.'de verilen aralıklarla 3 sınıfa ayrılmıştır.

Tablo.31. Ortalama süre niteliğinin veri dönüşümü

Ortalama süre (veri dönüşümünden önce)	**Ortalama süre (veri dönüşümünden sonra)**	**Veri sayısı**
Ortalama süre ≥ 10	10+	184
10 > Ortalama süre ≥ 5	5-10	403
5 > Ortalama süre	5-	413

Ortalama ziyaret sayısı: Bu çalışmada WEKA'da sınıflandırma algoritmaları için seçilen diğer bir hedef niteliktir. DoubleClick Ad Planner'deki bu niteliğe ait veriler sürekli değişken değerlerdir. Bu niteliğe ait veriler, ortalama değer dikkate alınarak aşağıdaki aralıklarla bölünerek diğer nitelikler gibi nominal değerlere dönüştürülmüştür.

Ortalama ziyaret sayısı niteliğinin veri dönüşümünden önceki verilerine ait basit istatistik Tablo.32.'de verilmektedir:

Tablo.32. Tüm web sitelerine ait ortalama ziyaret sayısı istatistiği

Ortalama ziyaret sayısı	
Maksimum değer	58
Minimum değer	3,3
Ortalama değer	6,32
Standart sapma	3,49

Ortalama ziyaret sayısı niteliğine ait veriler, Tablo.33.'de verilen aralıklarla sınıfa ayrılmıştır.

Tablo.33. Ortalama ziyaret sayısı niteliğinin veri dönüşümü

Ortalama ziyaret sayısı (veri dönüşümünden önce)	Ortalama ziyaret sayısı (veri dönüşümünden sonra)	Veri sayısı
Ort.ziyaret sayısı ≥ 10	10+	102
10 > Ort.ziyaret sayısı ≥ 5	5-10	467
5 > Ort.ziyaret sayısı	5-	431

3.4. Çalışmada Kullanılan Veri Analiz Aracı: WEKA

Çalışmada verilerin algoritmalarla incelenmesi ve analiz edilmesi WEKA (Waikato Environment for Knowledge Analysis) yazılımı ile yapılmıştır. WEKA, makine öğrenmesi ve veri madenciliği alanlarında kullanılan, açık kaynak kodlu Java tabanlı bir yazılımdır. Yazılım, Yeni Zelanda'da bulunan Waikato Üniversitesi'nde geliştirilmiştir. İçinde pek çok makine öğrenme algoritması barındıran WEKA yazılımı ile farklı algoritmalarla analiz yapılarak yorumlamak mümkündür.

Şekil. 23. WEKA Genel Kullanıcı Ara yüzü

WEKA GUI (General Users Interface) Genel Kullanıcı Arayüzünden 4 uygulamaya erişmek mümkündür. Bu uygulamalar WEKA 3.6.4 versiyonu için Şekil.23.'de görüleceği gibi *Explorer, Experimenter, KnowledgeFlow, Simple CLI* olarak sıralanır.

Explorer, verilerin yazılıma tanıtılması, işlenmesi ve analizlerin yapılmasını sağlayan paneldir. Bu panel 6 arayüzden oluşmaktadır.

Preprocess (Önişleme) arayüzünde, WEKA'da verilerin yüklenmesi ve veri önişlemesi yapılmaktadır. *Explorer* uygulamasında, veriler yüklenmeden *Preprocess* ara yüzü dışında hiçbir arayüzü kullanmak mümkün değildir. Dolayısı ile kural veya bilgi çıkarımı yapılması için öncelikle bu arayüzden verilerin yüklenmesi gerekmektedir. Veriler bir dosya olarak yüklenebileceği gibi, bir veritabanından da çekilebilmektedir. Doğrudan yüklemek istenilen veri dosyası .arff, .arff.gz, .names, .data, .csv, .libsvm, .dat, .bsi, .xrff, .xrff.gz uzantılarından birine ait olması gerekse de yaygınlıkla kullanılan dosya türü .arff ve .csv'dir. WEKA'nın tanımlayabildiği veriler sürekli değişken veya nominal olabilmektedir. Aşırı sayıda bulunan farklı sayısal veriler, nominal değerlerden farklı olarak sürekli değişken değerler olarak kabul edilip, veri dosyasında bir değişken kümesi olarak tanımlanmazlar. Nominal veriler ise belirli sayısal değerler (1, -5, 4.03 vb.) olabileceği gibi diğer türde verileri (büyük, az, 20-35, 110+ vb.) ifade eder ve bu değişkenler bir küme içerisinde önceden tanımlanırlar.

Liste.1. CSV dosyası örneği

```
yaş, öğrenim durumu, pozisyon, yıllık izin
26, ortaöğretim, memur, yok
41, ortaöğretim, şef, var
33, lisans, kimyager, var
27, lisans, mühendis, var
30, önlisans, laborant, yok
23, lisans, memur, yok
50, önlisans, memur, var
```

Liste.1.'den görüleceği gibi .csv dosya formatında yalnız veriler bulunmaktadır. İlk satır virgüllerle ayrılarak nitelikleri belirtir. Diğer satırlar ise örnek verileri nitelik sıralamasıyla verir.

.arff dosyası kullanılmak istenirse verilerin yanında bazı tanımlamaların kullanılması gerekmektedir.

Liste.2. ARFF dosyası örneği

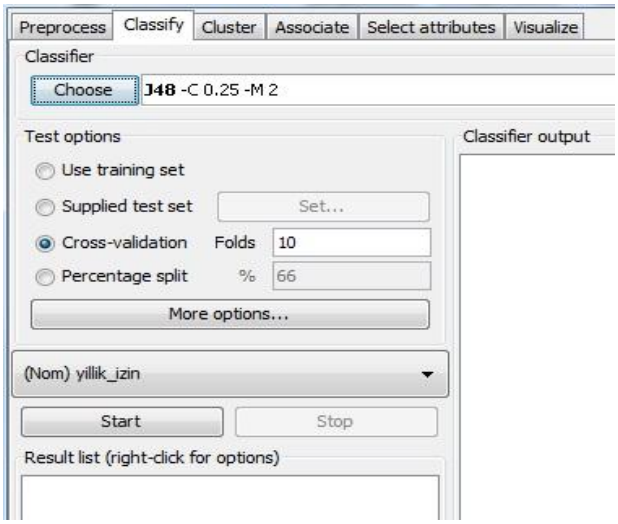
```
@relation personel

@attribute yas integer
@attribute ogrenim_durumu {ortaogretim, onlisans, lisans}
@attribute pozisyon {memur, sef, kimyager, laborant, muhendis}
@attribute yillik_izin {var, yok}

@data
26, ortaogretim, memur, yok
41, ortaogretim, sef, var
33, lisans, kimyager, var
27, lisans, muhendis, var
30, onlisans, laborant, yok
23, lisans, memur, yok
50, onlisans, memur, var
```

Liste.2.'den görüldüğü gibi @relation veritabanı adını, @attribute nitelikleri, @data ise verileri tanımlamak için kullanılır. Nitelik türü olarak sürekli değişkenlerde **real**, **numeric** veya **integer** kullanılmaktadır. Nominal değerlere sahip nitelikler ise { } içinde virgüllerle ayrılarak tanımlanırlar. @data bölümündeki her bir örnek veri, nitelik sıralamasında olması gerekmektedir.

WEKA *Explorer* panelindeki *Classify* (Sınıflandırma) arayüzü, WEKA'da sınıflandırma işlemlerinin yapıldığı bölümdür. Yüklenen veriler için *Classifier* altında *Choose* tıklanarak ilgili klasör içinde bulunan kullanılmak istenen sınıflandırma algoritmaları seçilebilir. Burada istenirse veriler için test ayarları yapmak da mümkündür. *Test options* (test seçenekleri) 'nde, eğer test yapılmayacaksa *Use training set* (eğitim kümesi kullan) seçilir. Test için başka bir veri kümesi kullanılacaksa *Supplied test set* seçilerek diğer veri kümesi yazılıma yüklenir. n fold *Cross-validation* (n katlı çapraz doğrulama) seçeneği ile veri kümesi istenilen (n) sayıda gruba bölünür ve bu gruplardan bir tanesi test için diğerleri (n-1 tanesi) ise eğitim kümesi olarak kullanılır. Bu işlem sırayla n defa tekrar edilerek, bütün verilerin test edilmesiyle tamamlanır. Varsayılan olarak bu sayı WEKA'da 10'dur.



Şekil.24. WEKA'da sınıflandırma ara yüzündeki test seçenekleri

Şekil.24.'de görüldüğü gibi, test ayarlarında veri kümesinin bir kısmını eğitim kümesinden ayırıp test amaçlı olarak da kullanmak mümkündür. *Percentage split* seçeneği ile veri kümesinin yüzde kaçının test amaçlı olarak kullanabileceği belirlenebilmektedir. Belirlenen yüzdeye karşılık gelen küme eğitim kümesi, geriye

kalanlar ise test verisi olarak kullanılmaktadır. Bu oran WEKA'da varsayılan olarak %66'dır. Bu da veri kümesinin %34'ünün test amacı ile kullanılacağını göstermektedir.

Cluster (Kümeleme) arayüzü, kümeleme işlemlerinin gerçekleştirildiği paneldir. Sınıflandırma arayüzüne benzer şekilde burada da uygulanmak istenen kümeleme algoritma seçilerek verilerin kümelenmesi sağlanarak analiz yapılması mümkündür.

Associate (Birliktelik) arayüzü aracılığı ile veriler arasındaki ilişkiler ortaya çıkarılır. Veriler üzerinde birliktelik kuralları uygulanarak yararlı sonuçlar bulunmaya çalışılır.

Select Attributes (Nitelikleri seç) arayüzü veri kümesi üzerinde yapılan seçme ve işleme özelliklerini ayarlamaya yarar[34].

Visualize (Görselleştirme) arayüzü, veri kümesinin görselleştirilmesini sağlayan paneldir. Veri kümesine ait niteliklerin birbirleri arasındaki ilişkiler grafik aracılığı ile gösterilir. Grafikler ve verileri temsil eden noktaların büyüklüğü ayarlanabilmektedir.

3.5. Verilerin WEKA'da Simgelenişi

Veri dönüştürme işlemleri sonrasında analizde kullanılacak 1000 web sitesinin sahip olduğu 6 nitelik ve bu niteliklere bağlı veriler Bölüm 4.2.'de açıklanacak kurallar dahilinde eğitim ve test kümeleri için .arff dosya formlarına getirilerek WEKA yazılımına yüklenmiştir.

Tablo.34. DoubleClick Ad Planner ile elde edilen örnek veriler

	Web Sitesi	**Site kategorisi**	**Erişim (%)**	**Farklı ziyaretçiler (çerezler) (milyon)**	**Sayfa görüntüleme sayısı (milyon)**	**Sitede geçirilen ort. süre (dakika)**	**Ort. Ziyaret Sayısı**
1	*facebook.com*	Sosyal Ağlar	74.8	60	40000	30	58
2	*live.com*	Arama Motorları	47.2	37	1100	7.5	24
3	*msn.com*	Web Portalları	23.8	20	250	5	14
4	*mynet.com*	Web Portalları	23.6	19	1000	11.16	18
5	*meb.gov.tr*	Devlet	23.6	19	1100	11.83	16

DoubleClick Ad Planer ile elde edilen verilerden ilk 5'i dönüşüm işlemleri öncesindeki formuyla Tablo.34.'de verilmektedir.

Veri dönüştürme işlemleri sonrasında, .arff dosya türüne dönüştürülmüş hali ile dosya içeriğinin ilk kısmından bir bölüm Liste.3.'de verilmektedir.

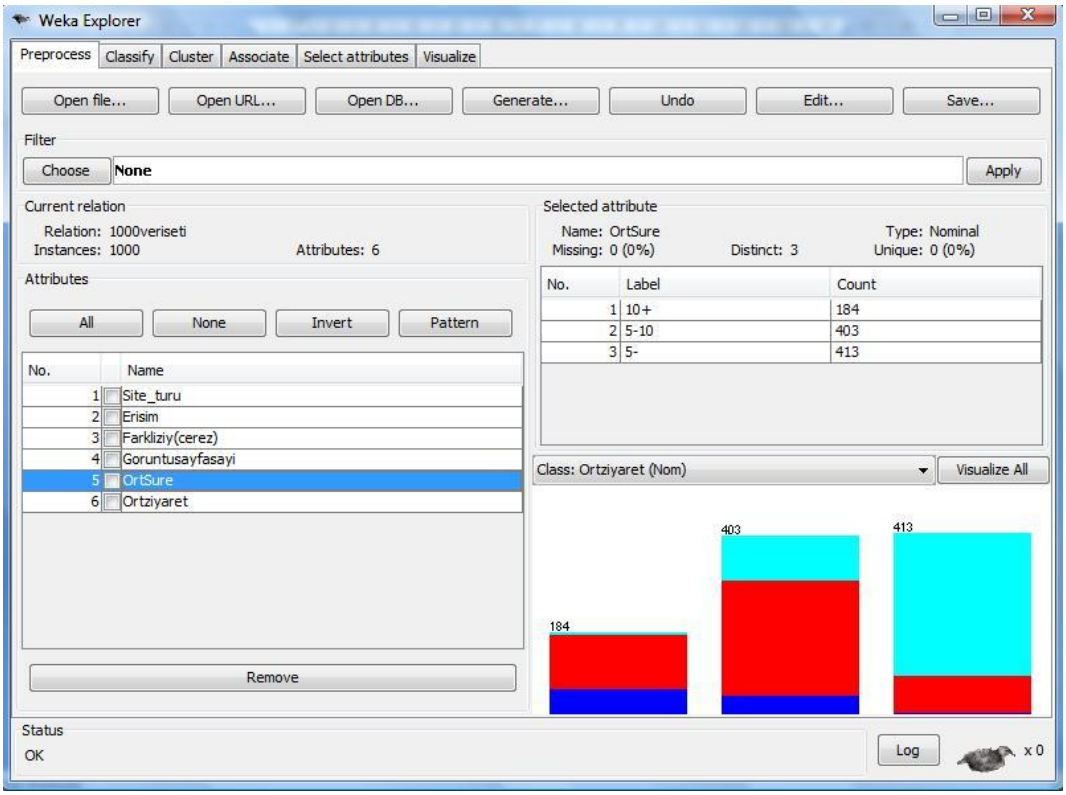
Liste.3. Dönüşümü yapılan verilerin .arff dosya verisi şeklindeki formu

```
@relation veriseti

@attribute Site_turu
{Sosyal_paylasim,Internet_portali,Resmi_kurum,Video_ve_muzik_paylasi
mi,Isletmeler,Ilanlar,Bilgi_kaynagi,Gazete_ve_haber,Oyun,e-
Ticaret,Bilisim_teknoloji,Kisisel_ilgi_beceri_alanlari,Dosya_paylasimi,For
umlar,Ekonomi_bankacilik,Arkadaslik_sohbet,Tv_sinema,Spor,Saglik,Egiti
m}
@attribute Erisim{3+,1-3,1-}
@attribute Farkliziy(cerez){3+,1-3,1-}
@attribute Goruntusayfasayi{100+,50-100,50-}
@attribute OrtSure{10+,5-10,5-}
@attribute Ortziyaret{10+,5-10,5-}

@data
Sosyal_paylasim,3+,3+,100+,10+,10+
Internet_portali,3+,3+,100+,5-10,10+
Internet_portali,3+,3+,100+,5-10,10+
Internet_portali,3+,3+,100+,10+,10+
Resmi_kurum,3+,3+,100+,10+,10+
Sosyal_paylasim,3+,3+,100+,5-10,5-10
Sosyal_paylasim,3+,3+,50-100,5-10,5-10
Video_ve_muzik_paylasimi,3+,3+,100+,5-10,5-10
.
.
```

1000 adet olan veri örneğinin, WEKA'ya yüklendikten sonraki ekran görüntüsü Şekil.25.'de verilmektedir.



Şekil.25. Verilerin WEKA'ya yüklendikten sonraki ekran görüntüsü

4. BULGULAR VE YORUMLAR

4.1. Makine Öğrenmesi Algoritmaları Performans Değerlendirme Ölçütleri

Makine öğrenmesi algoritmaları, bir uygulama üzerinde sınandığı zaman hangi oranda başarı elde edildiği bilinmesi istenir. Değerlendirme ve algoritmaların karşılaştırılması için birçok kavramdan yararlanılabilinir. Bu kavramlardan en çok kullanılanlar, doğruluk oranı, keskinlik, duyarlılık ve F-ölçütü'dür[35]. Bu ölçütler 0 ile 1 arasında değişkenlik gösterir ve değerlerinin yüksek olması başarı oranı ile doğru orantılıdır.

Herhangi bir sınıflandırma modelinde, test sonucunda bir veriye ait olan sınıf yanlış ya da doğru tahmin edilir. Bu tür problemlerde birden fazla sınıf olduğu için, bu durumu sayısal olarak özetleyen karmaşıklık matrislerinden (Confusion Matrix) faydalanılır (Tablo.35).

Tablo.35. Karmaşıklık matrisi genel formu

Karmaşıklık matrisi		Tahmini Sınıf	
		Pozitif	Negatif
Gerçek sınıf	Pozitif	TP	FN
	Negatif	FP	TN

Sınıflandırma işlemi sonucunda sınıflara göre tahmini yapılan veriler, gerçek sınıflarına göre yorumlandığında 4 durumdan birisine ait olmaktadır (Tablo.35). Bu durumlar aşağıda açıklanmaktadır:

TP (True Positive): Gerçekte sınıfı pozitif olan bir veri test ile sınıfı pozitif olarak tahmin edilmektedir.

FN(False Negative): Gerçekte sınıfı pozitif olan bir veri test ile sınıfı negatif olarak tahmin edilmektedir.

FP(False Positive): Gerçekte sınıfı negatif olan bir veri test ile sınıfı pozitif olarak tahmin edilmektedir.

TN(True Negative): Gerçekte sınıfı negatif olan bir veri test ile sınıfı negatif olarak tahmin edilmektedir.

Makine öğrenme algoritmalarında performans değerlendirmesinde kullanılan kavramlardan en sık kullanılanları aşağıda açıklanmaktadır.

Doğruluk:

Doğruluk oranı, test işlemindeki sınıflandırması doğru yapılan veri sayısının, kullanılan tüm veri sayısına oranı olarak tanımlanabilir. Sınıflandırması doğru yapılan verilerin (TP+TN), karmaşıklık matrisinin köşegeni üzerinde yer aldığı görülebilmektedir (Tablo.35.).

$$Doğruluk\ oranı = \frac{TP + TN}{TP + FN + FP + TN} \tag{4.1}$$

Örneğin doğruluk oranı 0,62 olan bir test işleminde, test için kullanılan tüm verilerin % 62'sinin sınıflandırması doğru yapılmıştır.

Doğruluk oranı ile doğrudan ilişkili bir tanımlamada hata oranı için yapılabilir. Hata oranı da sınıflandırması hatalı yapılmış veriler için söz konusu bir kavramdır ve

$$Hata\ oranı = 1 - \ Doğruluk\ oranı \tag{4.2}$$

formülü yardımıyla bulunabilmektedir.

Keskinlik:

Keskinlik (precision), gerçekte pozitif olup ve pozitif olarak tahmin edilen veri sayısının, pozitif tahmin edilen tüm veri sayısına oranıdır. Bir bakıma da belirli bir sınıfa yönelik olarak yapılan tüm tahminlerdeki doğru tahmin oranı diye tanımlanabilir.

$$Keskinlik = \frac{TP}{TP + FP} \tag{4.3}$$

Duyarlılık:

Gerçekte pozitif olup ve pozitif olarak tahmin edilen veri sayısının, gerçekte pozitif olan tüm veri sayısına oranı duyarlılık (recall) olarak tanımlanır. Başka deyişle, belirli bir sınıfa ait gerçek verilerdeki doğru tahmin oranıdır.

$$Duyarlılık = \frac{TP}{TP + FN}$$

(4.4)

F ölçütü:

F ölçütü, duyarlılık ve keskinlik değerinin harmonik ortalamasını ifade eden bir kavramdır. Çoğu zaman verileri, duyarlılık ve keskinlik değerleri ile ayrı ayrı değerlendirmek yerine, bu iki verinin hesaplara katıldığı F ölçütü ile değerlendirmek daha doğru olmaktadır.

$$F\ Ölçütü = \frac{2.Duyarlılık.Kesinlik}{Duyarlılık + Kesinlik}$$

(4.5)

TP oranı (TP rate):

Duyarlılık ile aynı değeri ifade etmektedir.

FP oranı (FP rate):

Gerçekte negatif sınıfa ait olup ve pozitif olarak tahmin edilen veri sayısının, gerçekte negatif olan tüm veri sayısına oranıdır.

$$FP\ oranı = \frac{FP}{FP + TN}$$

(4.6)

4.2. Makine Öğrenme Algoritmaları ile Analiz Sonuçları

4.2.1. Gözetimli Öğrenme Algoritmalarının Sonuçları

Gözetimli öğrenme algoritmaları analizinde, sınıflandırma kuralları çıkarılarak sınıf tahmini yapıldığından dolayı, yapılan tahminin doğru sınıflandırılması yapılıp yapılmadığının kontrolü için test işlemi ile sağlanır.

Bu çalışmada, gözetimli öğrenme algoritmalarında bir hedef nitelik yerine iki hedef nitelik incelenerek, hem algoritmaların farklı hedef nitelikler ile başarı durumu incelenmiş, hem de seçilen hedef nitelikler ile web siteleri tıklamaları analizinde ziyaretçilerin davranış ve tutumlarını yorumlayabilmek için sonuçlar elde edilmiştir.

Gözetimli öğrenme algoritmalarında başarı ve performans karşılaştırma işlemi için test işleminde n=10 için çapraz doğrulama (*Cross Validation*) yöntemi seçilip, her iki hedef nitelik için keskinlik, duyarlılık, F-ölçütü, doğruluk oranı değerleri tespit edilmiştir (WEKA'dan alınan bu değerler niteliklerin farklı sınıflarına ait değerlerin ağırlıklı ortalamalarıdır).

Çapraz doğrulama test işleminden sonra algoritmaların farklı eğitim ve test kümelerinde farklı veri sayıları ile performansları incelenmiştir(Tablo.36.). Bunun için değişik durumlarda veri kümesi, eğitim ve test kümesi olarak rastgele verilerden seçilerek iki kısma ayrılıp, farklı dosya olarak kaydedilmiştir. WEKA yazılımına yüklenen bu dosyalar ile farklı veriler ve veri sayılarına göre algoritma başarısının nasıl değiştiği veya sonuçların tutarlı olup olmadığı incelenmiştir.

Tablo.36. Eğitim ve test veri kümelerine ait veri sayıları

	Eğitim kümesi veri sayısı (Tüm verilere göre yüzdesi)	Test kümesi veri sayısı (Tüm verilere göre yüzdesi)
1.Durum	700 (%70)	300 (%30)
2. Durum	600 (%60)	400 (%40)
3. Durum	500 (%50)	500 (%50)

Sınıflandırma algoritmaları için yapılan tutarlılık analizi için 1000 adet örnekten 1. Durum için % 70'i yani 700 tanesi eğitim amaçlı kullanılmak üzere .arff dosyası haline getirilmiştir. Kalan 300 örnek ise başka bir .arff uzantılı dosya haline getirilerek eğitimi gerçekleşen verilerin testi için kullanılmıştır. 2. ve 3. durumda da veriler yine rastgele alınarak küme sayıları değiştirilip sonuçlar incelenmiştir. WEKA'da eğitim için test seçeneklerinde (test options) *Use training set* seçeneği ile seçilmiş ve buna ilişkin performans değerleri verilmiştir. Ardından *Supplied test set* seçeneği ile test için kullanılacak geriye kalan örneklerden oluşan test dosyası yazılıma yüklenerek, eğitimi gerçekleştirmiş algoritmaların yeni bir veri geldiğindeki tahmin ve sınıflama başarısı ölçülmüştür. Hedef nitelik olarak ortalama süre ve ortalama ziyaret sayısı olmak üzere 2 farklı nitelik için farklı durumdaki performans ölçütlerine ait değerler kaydedilmiştir.

Gözetimli öğrenme algoritmaların sonuçlarında, öncelikle algoritmaların sınıflandırma sonuçlarına ilişkin karmaşıklık matrisleri, sonrasında çapraz doğrulama

sonuçları, en son olarak da yukarıda bölümde belirtildiği üzere, eğitim ve test kümelerinin ayrı ayrı yapılan test sonuçları verilmiştir.

4.2.1.1. Çapraz Doğrulama Testi Karmaşıklık Matrisleri

Tablo.37. Naive Bayes algoritması için oluşan karmaşıklık matrisi

Çapraz Doğrulama - *Cross Validation* (n=10) [1000 veri]	**Hedef Nitelik**	
	Ortalama süre	**Ortalama ziyaret sayısı**
Naive Bayes algoritması	=== Confusion Matrix === a b c <-- classified as 92 83 9 | a = 10+ 52 241 110 | b = 5-10 13 83 317 | c = 5-	=== Confusion Matrix === a b c <-- classified as 30 69 3 | a = 10+ 19 354 94 | b = 5-10 0 100 331| c = 5-

Tablo.37.'deki Naive Bayes algoritması sonuçlarında yer alan karmaşıklık matrisindeki veriler tanımlanacak olursa;

Hedef nitelik, ortalama süre olduğu durumda 3 sınıftan [10+, 5-10, 5-] oluşmaktadır.

a=10+ b=5-10 ve c= 5- sınıflarını temsil etmektedir.

a	**b**	**c**	
92	83	9	**a=10+**
52	241	110	**b=5-10**
13	83	317	**c=5-**

a=10+ olan veri sayısı, ilk satırda yer alan veri sayısının toplamı yani 92+83+9=184'dir. Bu 184 veriden 92'si 10+, 83'ü 5-10 ve 9'u 5- olarak tahmin edilmiştir.

Tüm verilere (3 sınıfa) ait doğruluk oranı hesaplanırsa doğru sınıflandırılan veri sayısı TP+TN=92+241+317=650 ve yanlış sınıflandırılan veri sayısı FP+FN=83+9+52+110+13+83=350 olarak bulunur.

$$Doğruluk\ oranı = \frac{TP + TN}{TP + FN + FP + TN} = \frac{650}{650 + 350} = \frac{650}{1000} = 0{,}65$$

$$Hata\ oranı = 1 - \ Doğruluk\ oranı = 1 - 0{,}65 = 0{,}35$$

a=10+ sınıfı örnek olarak alındığında ise;

TP=92

FN=83+9=92

FP=52+13=65

TN=241+110+83+317=751'dir. Bu durumda a=10+ sınıfı için

$$Keskinlik = \frac{TP}{TP + FP} = \frac{92}{92 + 65} = 0{,}586$$

$$Duyarlılık = \frac{TP}{TP + FN} = \frac{92}{92 + 92} = 0{,}5$$

$$F\ Ölçütü = \frac{2\ .\ 0{,}586\ .0{,}5}{0{,}586 + 0{,}5} = 0{,}540$$

$$TP\ Oranı = Duyarlılık = 0{,}5$$

$$FP\ oranı = \frac{FP}{FP + TN} = \frac{65}{65 + 751} = 0{,}08$$

olur.

Her sınıf için ayrı hesaplanan bu değerler, sınıflara ait veri sayısı ile çarpılıp toplandıktan sonra tüm veri sayısına bölündüğünde niteliği temsil edecek olan ağırlıklı ortalamaları (Weighted Avg.) elde edilebilmektedir.

Ortalama ziyaret sayısı açısından bakılacak olunursa, a=10+ sınıfı için; keskinlik $_{a=10+}$= 30/(30+19+0)=0.612 ve duyarlılık$_{a=10+}$=30/(30+69+3)=0.294 olur. Tüm verilere ilişkin doğruluk oranı ise (30+354+331)/1000= % 71.5 olarak bulunabilir. b=5-10 ve c=5- sınıfları için verilerin çoğu doğru sınıfa atanırken (354 ve 331 adet), a=10+ sınıfı için verilerin çoğunluğu (69 adet) b=5-10 sınıfına atanmıştır.

Tablo.38. Bayes Ağı algoritması için oluşan karmaşıklık matrisi

Çapraz Doğrulama - *Cross Validation* (n=10) [1000 veri]	**Hedef Nitelik**	
	Ortalama süre	**Ortalama ziyaret sayısı**
Bayes Ağı algoritması	=== Confusion Matrix === a b c <-- classified as 92 84 8 | a = 10+ 52 240 111 | b = 5-10 13 83 317 | c = 5-	=== Confusion Matrix === a b c <-- classified as 32 67 3 | a = 10+ 18 357 92 | b = 5-10 0 100 331| c = 5-

Tablo.38.'den görüleceği gibi Bayes ağı algoritmasında ortalama süre hedef niteliği ile gerçekte 184 tane a=10+ verisinin 92'si 10+ olarak tahmin edilmiştir. 403 tane b=5-10 verisinin 240'ı 5-10, 413 tane c=5- verisinin ise 317'si 5- olarak doğru tahmin edilmiştir. c=5- sınıfı için, TP=317, FN=13+83=96, FP=111+8=119, TN=92+84+52+240=468'dir. Bu sınıf için duyarlılık hesaplanacak olunursa; duyarlılık$_{c=5-}$=TP/(TP+FN) =317/(317+96)=317/413=0.767 olarak bulunur.

Tablo.39. Destek Vektör Makinesi algoritması için oluşan karmaşıklık matrisi

Çapraz Doğrulama – *Cross Validation* (n=10) [1000 veri]	**Hedef nitelik**	
	Ortalama süre	**Ortalama ziyaret sayısı**
Destek Vektör Makinesi algoritması	=== Confusion Matrix === a b c <-- classified as 86 89 9 | a = 10+ 52 244 107 | b = 5-10 9 80 324 | c = 5-	=== Confusion Matrix === a b c <-- classified as 35 62 5 | a = 10+ 16 376 75 | b = 5-10 1 120 310 | c = 5-

Tablo.39.'dan Destek Vektör Makinesi algoritmasında ortalama ziyaret sayısı hedef niteliğinde a=10+ sınıfı için keskinlik hesaplanacak olunursa, TP=35, FN=62+5=67, FP=16+1=17, TN= 376+75+120+310=881'dir. Keskinlik$_{a=10+}$= TP/(TP+FP)= 35/(35+17)= 0.673 olarak bulunur. b=5-10 sınıfı için ise TP=376, FN=16+75=91, FP=62+120=182 ve TN=35+5+1+310=351'dir. Dolayısı ile keskinlik$_{b=5-10}$=376/(376+182)=0.673 bulunur. Benzer şekilde keskinlik$_{c=5-}$ =310/(310+80)=0.794 bulunur. Böylece ortalama ziyaret sayısında keskinlik için ağırlıklı ortalama keskinlik$_{ort}$=(0.673. 102+ 0.673.467 +0.794.431)/1000=0.726 olarak bulunur.

Tablo.40. K En Yakın Komşu algoritması için oluşan karmaşıklık matrisi

Çapraz Doğrulama - *Cross Validation* (n=10) [1000 veri]	**Hedef Nitelik**	
	Ortalama süre	**Ortalama ziyaret sayısı**
K en yakın komşu algoritması	=== Confusion Matrix === a b c <-- classified as 104 72 8 | a = 10+ 79 229 95 | b = 5-10 15 89 309 | c = 5-	=== Confusion Matrix === a b c <-- classified as 39 56 7 | a = 10+ 30 340 97 | b = 5-10 4 109 318| c = 5-

Tablo.40.'dan K en yakın komşu algoritmasında ortalama süre hedef niteliği ve a=10+ sınıfı için F-ölçütü hesaplaması yapılacak olunursa a=10+ sınıfı için, TP=104, FN=72+8=80, FP=79+15=94,TN=229+95+89+309=722'dir. Duyarlılık$_{a=10+}$ =TP/(TP+FN)=104/(104+80)=0.565 ve keskinlik$_{a=10+}$ =TP/(TP+FP)= 104/(104+94) =0.525 olur. F-ölçütü bu iki değerin harmonik ortalaması olduğundan, F-ölçütü$_{a=10+}$=2.0.565.0.525/(0565+0525)= 0.544 olarak elde edilir.

Tablo.41. ID3 algoritması için oluşan karmaşıklık matrisi

Çapraz Doğrulama - *Cross Validation* (n=10) [1000 veri]	**Hedef Nitelik**	
	Ortalama süre	**Ortalama ziyaret sayısı**
ID3 algoritması	=== Confusion Matrix === a b c <-- classified as 107 64 9 | a = 10+ 77 220 92 | b = 5-10 20 81 310| c = 5-	=== Confusion Matrix === a b c <-- classified as 50 45 4 | a = 10+ 33 332 90 | b = 5-10 6 113 310| c = 5-

Tablo.41.'den ID3 algoritmasının FN oranının hesaplaması için ortalama süre hedef niteliğinde b=5-10 sınıfı göz önüne alındığında, TP=220, FN=77+92=169, FP=64+81=145, TN=107+9+20+310=446'dır. FN oranı$_{b=5-10}$=FP/(FP+TN)= 145/(145+446)=145/591 =0.245 olarak elde edilir. TP oranı ise TP oranı$_{b=5-10}$=TP/(TP+FN)=220/(220+169)= 220/389=0.566 olarak bulunur.

Tablo.42. C4.5 algoritması için oluşan karmaşıklık matrisi

Çapraz Doğrulama - *Cross Validation* (n=10) [1000 veri]	**Hedef Nitelik**	
	Ortalama süre	**Ortalama ziyaret sayısı**
C4.5 algoritması	=== Confusion Matrix === a b c <-- classified as 100 74 10 | a = 10+ 48 237 118 | b = 5-10 15 56 342 | c = 5-	=== Confusion Matrix === a b c <-- classified as 17 81 4 | a = 10+ 19 364 84 | b = 5-10 0 111 320| c = 5-

Tablo.42.'den C4.5 algoritmasında ortalama süre hedef niteliği için doğruluk ve hata oranı hesaplanacak olunursa, doğruluk =(TP+TN)/(TP+TN+FP+FN) =(100+237+342)/1000 =% 67.9 ve hata oranı ise 1-% 67.9=% 32.1 olarak elde edilir.

4.2.1.2. Çapraz Doğrulama Testi Performans Sonuçları ve Karşılaştırması

1000 veri üzerinde iki farklı hedef nitelik için elde edilen çapraz doğrulama test sonuçları Tablo.43.'de verilmiştir.

Tablo.43. Gözetimli öğrenme algoritmaları çapraz doğrulama testi sonuçları

Çapraz Doğrulama - *Cross Validation* (n=10) [1000 veri]	**Hedef Nitelik**									
	Ortalama süre					**Ortalama ziyaret sayısı**				
	Keskinlik	**Duyarlılık**	**F -Ölçütü**	**Doğru-Yanlış veri sayısı**	**Doğruluk (%)**	**Keskinlik**	**Duyarlılık**	**F- Ölçütü**	**Doğru-Yanlış veri sayısı**	**Doğruluk (%)**
Naive Bayes	0.647	0.65	0.648	650-350	**65**	0.712	0.715	0.707	715-285	**71.5**
Bayes Ağı	0.646	0.649	0.647	649-351	**64.9**	0.718	0.72	0.712	720-280	**72**
Destek Vektör Makinesi	0.65	0.654	0.65	654-346	**65.4**	0.726	0.721	0.714	721-279	**72.1**
K En Yakın Komşu	0.643	0.642	0.642	642-358	**64.2**	0.694	0.697	0.694	697-303	**69.7**
ID3	0.652	0.65	0.65	637-343	**63.7**	0.705	0.704	0.704	692-291	**69.2**
C4.5	0.674	0.679	0.674	679-321	**67.9**	0.692	0.701	0.686	701-299	**70.1**

Gözetimli öğrenme algoritmalarında, testi yapılan 6 algoritma için sonuçların birbirine genel olarak yakın çıktığı gözlenmiştir. Ortalama süre hedef niteliği ile en yüksek doğruluk oranı % 67.9 ile C4.5 karar ağacı algoritmasında görülmüştür. C4.5 algoritmasında keskinlik, duyarlılık ve F-ölçütü değerlerinin hepsi de, diğer algoritmalara göre üstünlük sağlamıştır. Aynı hedef nitelik için ikinci en yüksek doğruluk oranı % 65.4 ile Destek Vektör Makinesinde elde edilmiştir. Destek Vektör

Makinesi, aynı başarısını keskinlikte koruyamamıştır. Şöyle ki keskinlikte % 65 değeri ile % 65.2 başarı gösteren ID3 algoritmasından sonra gelmektedir. Duyarlılıkta ise % 65.4 oranı ile ikinci sırada yer almış, F-ölçütünde ise % 65 oran ile ikinci sırayı ID3 algoritması ile paylaşmıştır. Doğruluk oranını en düşük % 63.7 ile ID3 algoritması göstermiştir.

Ortalama ziyaret sayısı hedef nitelik olması durumunda ise Destek Vektör Makinesi tüm performans kriterlerin de üstünlük sağlamıştır. % 72.1 doğruluk oranı sağlayan bu algoritma, keskinlik, duyarlılık ve F-ölçütünde sırasıyla % 72.6, %72.1 ve % 71.4 değerlerini sağlamıştır. Hemen ardından ikinci en yüksek performansı ise % 72 doğruluk oranı ise Bayes Ağı göstermiştir. Bayes Ağı diğer tüm kriterlerde de Destek Vektör Makinesi değerlerine yaklaşmıştır. Doğruluk oranındaki en düşük değer ID3 algoritmasında diğer kriterlerde ise C4.5 veya K en yakın komşu algoritmasında gözlenmiştir.

Ortalama ziyaret sayısı hedef nitelik seçildiğinde, tüm algoritmalarda diğer hedef niteliğe göre ortalama olarak % 5 oranında başarı performansının arttığı görülmüştür. İki farklı hedef nitelik arasında en büyük fark Bayes ağında gözlenmiştir. Doğruluk oranında % 7.1 bir fark mevcuttur. En yakın değerlerin görüldüğü algoritma ise C4.5 algoritmasıdır. Doğruluk oranları arasındaki fark sadece % 2.2 dir.

4.2.1.3. Eğitim ve Test Kümeleri Performans Sonuçları

Naive Bayes Algoritması ile Analiz Sonuçları

WEKA'daki NaiveBayes kütüphanesi kullanılarak elde edilen sonuçların özeti Tablo.44.'de verilmektedir.

Naive Bayes algoritması sonuçlarında hedef niteliğin ortalama süre olması durumunda 700 eğitim verisi ile doğruluk oranı % 68.71 iken, 300 veri ile test işleminde doğruluk oranı yaklaşık % 4 lük bir hata ile % 64.66 değerini yakalamıştır.

Hedef niteliğin ortalama ziyaret sayısı olması durumunda da test performansı ile eğitim performansı birbirine daha çok yaklaşmıştır.

Tablo.44. Naive Bayes algoritması eğitim/test sonuçları

Naive Bayes	Hedef Nitelik									
	Ortalama süre					Ortalama ziyaret sayısı				
	Keskinlik	Duyarlılık	F -Ölçütü	Doğru-Yanlış veri sayısı	Doğruluk (%)	Keskinlik	Duyarlılık	F- Ölçütü	Doğru-Yanlış veri sayısı	Doğruluk (%)
Eğitim kümesi-(1) (700 veri)	0.682	0.687	0.683	481-219	**68.71**	0.721	0.726	0.72	508-192	**72.57**
Test kümesi-(1) (300 veri)	0.643	0.647	0.644	194-106	**64.66**	0.703	0.703	0.694	211-89	**70.33**
Eğitim kümesi-(2) (500 veri)	0.663	0.664	0.663	332-168	**66.4**	0.719	0.72	0.716	360-140	**72**
Test kümesi-(2) (500 veri)	0.65	0.658	0.653	329-171	**65.8**	0.712	0.712	0.699	356-144	**71.2**

Eğitim veri sayısı artışının öğrenmeye katkısı için ise 400/600/800 veri ile yapılan analizde, algoritma performansının neredeyse hiç değişmediği gözlenmiştir.

Bayes Ağı Algoritması ile Analiz Sonuçları

Bayes ağı algoritması ile analiz yapabilmek için WEKA'da Bayes.Net kütüphanesi seçilerek öğrenme ve test işlemi yapılmıştır. Sonuçlar Tablo.45.'de verilmektedir.

Tablo.45. Bayes Ağı algoritması eğitim/test sonuçları

Bayes Ağı	Hedef Nitelik									
	Ortalama süre					Ortalama ziyaret sayısı				
	Keskinlik	Duyarlılık	F -Ölçütü	Doğru-Yanlış veri sayısı	Doğruluk (%)	Keskinlik	Duyarlılık	F- Ölçütü	Doğru-Yanlış veri sayısı	Doğruluk (%)
Eğitim kümesi-(1) (700 veri)	0.684	0.689	0.685	482-218	**68.85**	0.721	0.726	0.72	508-192	**72.57**
Test kümesi-(1) (300 veri)	0.646	0.65	0.647	195-105	**65**	0.696	0.7	0.693	210-90	**70**
Eğitim kümesi-(2) (600 veri)	0.666	0.667	0.666	400-200	**66.66**	0.731	0.728	0.721	437-163	**72.83**
Test kümesi-(2) (400 veri)	0.655	0.65	0.651	260-140	**65**	0.715	0.723	0.709	289-111	**72.25**

Bayes ağı sonuçları incelendiğinde eğitim ve test performanslarının birbirine yakın olduğu gözlenmiştir. Hedef niteliğin ortalama süre olması durumunda 700 eğitim verisi ile keskinlik ve duyarlılık değerleri % 68.4 ve % 68.9 iken 300 veri ile test işlemi sonucunda bu değerler yaklaşık % 4 hata ile % 64.6 ve % 65 olmuştur.

Hedef niteliğin ortalama ziyaret sayısı olması ile eğitim ve test işlemleri arasındaki hatanın küçüldüğü gözlenmiştir. Bu fark ilk durumdaki kümeler için yaklaşık % 2.5 iken ikinci durumdaki kümeler için % 1 dolaylarındadır.

Eğitim veri sayısı artışının öğrenmeye katkısı için ise 400/600/800 veri ile yapılan analizde, algoritma performansının çok az (% 1-2 arası) etkilendiği gözlenmiştir.

Destek Vektör Makinesi Algoritması ile Analiz Sonuçları

WEKA'daki SMO kütüphanesi kullanılarak elde edilen Destek Vektör makinesi algoritmasına ait öğrenme ve test sonuçları Tablo.46.'da verilmektedir.

Tablo.46. Destek Vektör Makinesi algoritması eğitim/test sonuçları

Destek Vektör Mak.	Hedef Nitelik									
	Ortalama süre					Ortalama ziyaret sayısı				
	Keskinlik	Duyarlılık	F -Ölçütü	Doğru-Yanlış veri sayısı	Doğruluk (%)	Keskinlik	Duyarlılık	F- Ölçütü	Doğru-Yanlış veri sayısı	Doğruluk (%)
Eğitim kümesi-(1) (700 veri)	0.705	0.707	0.705	495-205	**70.71**	0.763	0.747	0.741	523-177	**74.71**
Test kümesi-(1) (300 veri)	0.652	0.65	0.651	195-105	**65**	0.718	0.713	0.702	214-86	**71.33**
Eğitim kümesi-(3) (500 veri)	0.703	0.702	0.701	351-149	**70.2**	0.759	0.748	0.745	374-126	**74.8**
Test kümesi-(3) (500 veri)	0.687	0.688	0.685	344-156	**68.8**	0.743	0.732	0.719	366-134	**73.2**

Destek Vektör Makinesi algoritması ile her iki hedef nitelikle eğitim ve test performansları birbirine çok yakın olmasa da yakın olduğu söylenebilir. 700 eğitim/300 test verisi ile doğruluk oranı ve F-ölçütü ise % 5 dolaylarında düşme gözlenmiştir.

Ortalama ziyaret sayısı için ise söz konusu değişim daha azdır. 700 veri ile eğitim işlemindeki doğruluk oranı % 74.71 iken 300 veri ile test işleminde % 71.33 oranına ulaşılmıştır.

Eğitim veri sayısı artışının öğrenmeye katkısı için ise 400/600/800 veri ile yapılan analizde, algoritma performansı, ortalama süre hedef niteliği için % 5-6 oranında azalmış, ortalama ziyaret sayısı için ise pek değişmemiştir.

K- En Yakın Komşu Algoritması ile Analiz Sonuçları

Bu algoritmanın uygulaması için WEKA'da Ibk kütüphanesi seçilmiştir. Teste ilişkin sonuçlar Tablo.47.'de verilmektedir:

Tablo.47. K En Yakın Komşu algoritması eğitim/test sonuçları

K en yakın komşu	Hedef Nitelik									
	Ortalama süre					Ortalama ziyaret sayısı				
	Keskinlik	Duyarlılık	F -Ölçütü	Doğru-Yanlış veri sayısı	Doğruluk (%)	Keskinlik	Duyarlılık	F- Ölçütü	Doğru-Yanlış veri sayısı	Doğruluk (%)
Eğitim kümesi-(1) (700 veri)	0.795	0.791	0.792	554-146	**79.14**	0.84	0.836	0.836	585-115	**83.57**
Test kümesi-(1) (300 veri)	0.661	0.66	0.661	198-102	**66**	0.733	0.737	0.731	221-79	**73.66**
Eğitim kümesi-(2) (600 veri)	0.798	0.797	0.797	478-122	**79.66**	0.825	0.822	0.821	493-107	**82.16**
Test kümesi-(2) (400 veri)	0.645	0.635	0.639	254-146	**63.5**	0.714	0.723	0.712	289-111	**72.25**

K en yakın komşu algoritmasın sonuçları analiz edildiğinde eğitim ve test işlemlerinde başarı oranları arasındaki farkın fazla olduğu gözlenmiştir. Hedef niteliğin ortalama süre olması ile eğitimde doğruluk % 79 seviyesinden test işleminde % 63-66 seviyelerine inmektedir.

Hedef niteliğin ortalama ziyaret sayısı olması ile eğitim ve test performansı arasındaki hata azalmakta, ancak bu performanslar yakın olamamaktadır. Bu hata miktarı doğruluk oranı için yaklaşık % 10 dolayındadır.

Eğitim veri sayısı artışının öğrenmeye katkısı için ise 400/600/800 veri ile yapılan analizde, algoritma performansı sırasıyla önce % 5 sonra %1 dolaylarında azalmasına sebep olmuştur.

ID3 Algoritması ile Analiz Sonuçları

WEKA'da veriler üzerinde ID3 algoritmasının uygulanabilmesi için ID3 kütüphanesi seçilmiştir. Algoritmanın eğitim/test sonuçları Tablo.48.'de verilmektedir:

Tablo.48. ID3 algoritması eğitim/test sonuçları

ID3	Hedef Nitelik									
	Ortalama süre					Ortalama ziyaret sayısı				
	Keskinlik	Duyarlılık	F -Ölçütü	Doğru-Yanlış veri sayısı	Doğruluk (%)	Keskinlik	Duyarlılık	F- Ölçütü	Doğru-Yanlış veri sayısı	Doğruluk (%)
Eğitim kümesi-(1) (700 veri)	0.795	0.791	0.792	554-146	**79.14**	0.84	0.836	0.836	585-115	**83.57**
Test kümesi-(1) (300 veri)	0.65	0.651	0.651	190-102-(8)	**63.33**	0.743	0.745	0.744	213-73-(14)	**71**
Eğitim kümesi-(3) (500 veri)	0.813	0.808	0.808	404-96	**80.8**	0.851	0.85	0.85	425-75	**85**
Test kümesi-(3) (500 veri)	0.639	0.637	0.637	310-177-(13)	**62**	0.699	0.698	696	338-146-(16)	**67.6**

ID3 algoritmasında hedef niteliğin ortalama süre olması ile 700 eğitim verisi ve 300 test verisi ile yapılan analizde % 16-18 dolaylarında performans düşüklüğü gözlenmiştir.

Ortalama ziyaret sayısının hedef nitelik olması durumunda eğitim ve test performansının ortalama süreye göre her ikisinde bir artış olmuş ancak aradaki hata farkı pek değişmemiştir.

Eğitim verilerinin sayısı ile performans arasındaki ilişkiye bakılacak olunursa, ortalama süre hedef niteliği için 400/600/800 eğitim verisi ile doğruluk oranlarında yaklaşık önce % 4 sonra % 1 azalmaya sebep olmuştur. Dolayısı ile

eğitim verilerinin artışı ile öğrenmede % 5 dolaylarında performans düşmesine sebep olmuştur. Diğer hedef nitelikte hemen hemen aynı durum gerçekleşmiştir.

C4.5 Algoritması ile Analiz Sonuçları

Karar ağaçları algoritmalarından C4.5 algoritması ile J.48 kütüphanesi kullanılarak WEKA'da elde edilen eğitim/test sonuçları Tablo.49'da verilmektedir.

Tablo.49. C4.5 algoritması eğitim/test sonuçları

C4.5	Hedef Nitelik									
	Ortalama süre					Ortalama ziyaret sayısı				
	Keskinlik	Duyarlılık	F -Ölçütü	Doğru-Yanlış veri sayısı	Doğruluk (%)	Keskinlik	Duyarlılık	F- Ölçütü	Doğru-Yanlış veri sayısı	Doğruluk (%)
Eğitim kümesi-(1) (700 veri)	0.731	0.734	0.73	514-186	**73.42**	0.645	0.71	0.673	497-203	**71**
Test kümesi-(1) (300 veri)	0.68	0.68	0.677	204-96	**68**	0.635	0.697	0.662	209-91	**69.66**
Eğitim kümesi-(2) (600 veri)	0.702	0.703	0.697	422-178	**70.33**	0.642	0.698	0.665	419-181	**69.83**
Test kümesi-(2) (400 veri)	0.664	0.668	0.664	267-133	**66.75**	0.642	0.718	0.676	287-113	**71.75**

C4.5 algoritması ile hedef niteliğin ortalama süre olması durumunda 700 eğitim verisi ile keskinlik %73.1, duyarlılık %73.4 ve doğruluk % 73.42 olmuştur. 300 veri ile yapılan test işleminde ise bu değerler yaklaşık % 5 hata payıyla her üç değerde de % 68 gibi yakın bir oranı yakalamıştır. 600 eğitim ve 400 test verisiyle yapılan durumda da aradaki yaklaşık aynı hata payıyla performans gözlenmiştir.

Ortalama ziyaret sayısının hedef nitelik olması durumunda ise test performansı eğitim performansına daha yakındır. Aradaki fark % 2 dolaylarındadır.

Eğitim verisi sayısı ile doğruluk oranı kıyaslandığında ilk hedef nitelik için, bu değerler 400/600/800 eğitim verisi için % 5 dolaylarında performans artışını sağlamıştır. Ortalama ziyaret sayısı hedef niteliği için ise algoritma performansını pek etkilememiştir.

4.2.1.4. Eğitim ve Test Kümeleri Performans Karşılaştırması

Bölüm 4.2.1.3.'de eğitim ve test performansı ayrı ayrı verilen gözetimli öğrenme algoritmalarının 700 eğitim/300 test verisi sonuçlarında yer alan doğruluk oranları ve bu oranlar arasındaki farklar, daha rahat görülebilmesi için karşılaştırmalı olarak Tablo.50.'de verilmektedir.

Tablo.50. Eğitim/test sonuçları açısından algoritmaların karşılaştırılması

700 Eğitim / 300 Test Verisi	Hedef Nitelik					
	Ortalama süre			Ortalama ziyaret sayısı		
Gözetimli Öğrenme Algoritmaları	Doğruluk oranı (%) (Eğitim kümesi)	Doğruluk oranı (%) (Test kümesi)	Kümeler arasındaki doğruluk oranı farkı (%)	Doğruluk oranı (%) (Eğitim kümesi)	Doğruluk oranı (%) (Test kümesi)	Kümeler arasındaki doğruluk oranı farkı (%)
Naive Bayes	68.71	64.66	4.05	72.57	70.33	2.24
Bayes Ağı	68.85	65	3.85	72.57	70	2.57
Destek Vektör Makinesi	70.71	65	5.71	74.71	71.33	3.38
K En Yakın Komşu	79.14	66	13.14	83.57	73.66	9.91
ID3	79.14	63.33	15.81	83.57	71	12.57
C4.5	73.42	68	5.42	71	69.66	1.34

Tablo.50.'den görüleceği gibi hedef niteliğin ortalama süre seçilmesi durumunda eğitim ve test doğruluğu en yakın gerçekleşen algoritma Bayes ağı olmuştur. Naive Bayes, C4.5 ve Destek vektör makinesi algoritmaları da yakın çıkan algoritmalar arasındadır.

Ortalama ziyaret sayısı hedef niteliği ile ise eğitim ve test doğruluğu en yakın çıkan algoritma C4.5 algoritması olmuştur. Diğer hedef nitelik gibi Naive Bayes, Bayes ağı ve Destek vektör makinesi algoritmaları değerleri yine birbirine yakın çıktığı görülmektedir.

Başarısı en düşük algoritmalar ise her iki hedef nitelik içinde K en yakın komşu ve ID3 algoritması olmuştur. Test doğrulukları, öğrenme doğruluklarına göre % 9.91 ila % 15.81 arasında düşme göstermiştir.

4.2.1.5. Gözetimli Öğrenme Algoritmaları Sonuçlarının Değerlendirilmesi

n=10 için çapraz doğrulama sonuçları ile yapılan test sonuçlarında algoritma başarıları arasında büyük fark olmamakla birlikte, ortalama süre hedef niteliği ile en iyi performans C4.5 algoritmasında, ortalama ziyaret sayısı hedef niteliği ile ise Destek Vektör Makinesinde elde edilmiştir. Her iki hedef nitelikte de en kötü performansı diğerlerine göre yaklaşık % 3'lük küçük bir başarı eksikliği ile ID3 algoritması göstermiştir.

Veri kümesinin eğitim ve test kümesi olarak birbirinden ayrılarak yapılan analizde ise K en yakın komşu ve ID3 algoritmalarının başarı performansı oldukça düşmektedir. Eğitim-test kümeleri arasındaki doğruluk oranı farkı diğer algoritmalarda % 5 dolaylarına kadarken, bu iki algoritmada % 15 dolaylarına kadar çıkmaktadır. Bu da ID3 ve K en yakın komşu algoritmasının sınıfı tahmin edilmesi istenen yeni bir veri için, iyi bir tahmin yapamayabileceğinin göstergesidir. Bu bakımdan, bu iki algoritmanın web tıklama alanındaki analizlerde kullanılması dezavantajlı olabilecektir.

İncelenen hedef nitelikler açısından oluşan genel bir durum ise, ortalama süre niteliği ile elde edilen performansın, ortalama ziyaret sayısı hedef niteliği performansının hep altında kaldığıdır. Bu da ziyaret sayılarının, sürelere göre bu analizler için daha elverişli olduğunu göstermektedir.

Ortalama süre hedef niteliği için en başarılı algoritma C4.5 algoritması olmuştur. Bu algoritmaya ait WEKA'da elde edilen sonuçlarının büyük bir bölümü Liste.4.de verilmiştir.

Ortalama süre niteliği ile C4.5 algoritması ile oluşturulan karar ağacı incelendiğinde ağacın kök düğümünü ortalama ziyaret sayısı temsil etmektedir. Bu da en yüksek kazanç oranını ortalama ziyaret sayısının sağladığını göstermektedir. Öte yandan, veri kümesi ortalama süre dikkate alınarak sınıflara ayrılmak istendiğinde bu niteliği en iyi temsil edecek nitelik ortalama ziyaret sayısıdır denilebilir.

Liste.4. Ortalama süre hedef niteliği için WEKA'da C4.5 algoritması çıktısı

```
Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     1000veriseti
Instances:    1000
Test mode:    10-fold cross-validation
=== Classifier model (full training set) ===
J48 pruned tree
------------------
Ortziyaret = 10+
|   Site_turu = Sosyal_paylasim: 10+ (5.0/1.0)
|   Site_turu = Internet_portali
|   |   Erisim = 3+: 5-10 (7.0/2.0)
|   |   Erisim = 1-3: 5- (1.0)
|   |   Erisim = 1-: 10+ (2.0)
|   Site_turu = Resmi_kurum: 5-10 (8.0/2.0)
|   Site_turu = Video_ve_muzik_paylasimi: 10+ (1.0)
|   Site_turu = Isletmeler: 10+ (2.0)
|   Site_turu = Ilanlar: 10+ (1.0)
|   Site_turu = Bilgi_kaynagi: 10+ (3.0/1.0)
|   Site_turu = Gazete_ve_haber: 10+ (16.0/4.0)
|   Site_turu = Oyun: 10+ (12.0/2.0)
|   Site_turu = e-Ticaret: 10+ (1.0)
|   Site_turu = Bilisim_teknoloji: 10+ (0.0)
|   Site_turu = Kisisel_ilgi_beceri_alanlari: 10+ (5.0)
|   Site_turu = Dosya_paylasimi: 5-10 (4.0/1.0)
|   Site_turu = Forumlar: 5-10 (1.0)
|   Site_turu = Ekonomi_bankacilik: 10+ (5.0/1.0)
|   Site_turu = Arkadaslik_sohbet: 10+ (5.0/1.0)
|   Site_turu = Tv_sinema: 5-10 (8.0/2.0)
|   Site_turu = Spor: 5-10 (15.0/4.0)
|   Site_turu = Saglik: 10+ (0.0)
|   Site_turu = Egitim: 10+ (0.0)
Ortziyaret = 5-10
|   Site_turu = Sosyal_paylasim
|   |   Farkliziy(cerez) = 3+: 5-10 (4.0/1.0)
|   |   Farkliziy(cerez) = 1-3: 5- (3.0/1.0)
|   |   Farkliziy(cerez) = 1-
|   |   |   Goruntusayfasayi = 100+: 10+ (0.0)
|   |   |   Goruntusayfasayi = 50-100: 5- (3.0)
…..
….
…
|
Ortziyaret = 5-: 5- (431.0/107.0)
```

C4.5 karar ağacı ile ortalama süre hedef niteliği ile kök düğüme atanan ortalama ziyaret sayısı arasındaki veri dağılımına ait istatistiki veri dağılımı Tablo.51.'de özetlenmiştir.

Tablo.51. Kök düğümde gerçekleşen dallanmaya göre verilerin dağılımı

Kök düğüm	Oluşan yapraklar	Oran
Ortalama ziyaret sayısı=10+ :102	Ortalama süre=10+ : 58	%5.8
	Ortalama süre=5-10: 43	%4.3
	Ortalama süre=5- : 1	% 0.1
Ortalama ziyaret sayısı=5-10: 467	Ortalama süre=10+ : 87	%8.7
	Ortalama süre=5-10: 331	%33.1
	Ortalama süre=5- : 49	%4.9
Ortalama ziyaret sayısı=5- : 431	Ortalama süre=10+ : 0	% 0
	Ortalama süre=5-10: 0	% 0
	Ortalama süre=5- :431	% 43.1

Ortalama ziyaret sayısı 10+ ve 5-10 sınıfları için oluşturulan dallara bakıldığında, hemen altında oluşan düğüm site türü niteliğidir. Bu da, ortalama süre ile ilişkili ikinci niteliğin site türü olduğunu göstermektedir. Başka bir deyişle, ortalama süreyi belirleyen ikinci etken site türüdür. Ortalama ziyaret sayısı 5- sınıfına ait dal için ise herhangi bir alt düğüm oluşmamıştır. Oluşan (ortalama süre=5-) yaprakta 431 veriden 324 tanesi doğru olarak sınıflandırılmıştır. Yani yaprağa gelen 5'in altındaki ziyaret sayılarının % 75'inin ortalama süre değeri de 5 dk.'ın altındadır.

Ortalama ziyaret sayısı hedef niteliği için en başarılı algoritma Destek vektör makinesi algoritması olmuştur. Bu algoritmaya ait WEKA'da elde edilen sonuçlardan bir bölüm Liste.5'de verilmiştir.

Liste.5. Ortalama ziyaret sayısı hedef niteliği için WEKA'da Destek Vektör Makinesi algoritması çıktısı

```
Scheme:       weka.classifiers.functions.SMO
Relation:     1000veriseti
Instances:    1000
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

SMO
Kernel used:
  Linear Kernel: K(x,y) = <x,y>
Classifier for classes: 10+, 5-10

BinarySMO
Machine linear: showing attribute weights, not support vectors.
```

```
      0.0589 * (normalized) Site_turu=Sosyal_paylasim
+     -1.9405 * (normalized) Site_turu=Internet_portali
+      0.0584 * (normalized) Site_turu=Resmi_kurum
+      1      * (normalized) Site_turu=Video_ve_muzik_paylasimi
+      0.0581 * (normalized) Site_turu=Isletmeler
+      1      * (normalized) Site_turu=Ilanlar
+      0.059  * (normalized) Site_turu=Bilgi_kaynagi
+      0.0579 * (normalized) Site_turu=Gazete_ve_haber
+      0.0591 * (normalized) Site_turu=Oyun
+      1      * (normalized) Site_turu=e-Ticaret
+      0.061  * (normalized) Site_turu=Bilisim_teknoloji
+      0.0581 * (normalized) Site_turu=Kisisel_ilgi_beceri_alanlari
+      0.0585 * (normalized) Site_turu=Dosya_paylasimi
+      0.0581 * (normalized) Site_turu=Forumlar
+      0.0582 * (normalized) Site_turu=Ekonomi_bankacilik
+      0.0584 * (normalized) Site_turu=Arkadaslik_sohbet
+      0.0594 * (normalized) Site_turu=Tv_sinema
+     -1.9415 * (normalized) Site_turu=Spor
+      0.0599 * (normalized) Site_turu=Saglik
+      0.0592 * (normalized) Site_turu=Egitim
+      0.0012 * (normalized) Erisim=3+
+     -0.0006 * (normalized) Erisim=1-3
+     -0.0006 * (normalized) Erisim=1-
+     -0.0016 * (normalized) Farkliziy(cerez)=3+
+      0.0005 * (normalized) Farkliziy(cerez)=1-3
+      0.0011 * (normalized) Farkliziy(cerez)=1-
+     -1.3312 * (normalized) Goruntusayfasayi=100+
+      0.6654 * (normalized) Goruntusayfasayi=50-100
+      0.6658 * (normalized) Goruntusayfasayi=50-
+     -0.6673 * (normalized) OrtSure=10+
+     -0.666  * (normalized) OrtSure=5-10
+      1.3332 * (normalized) OrtSure=5-
+      0.9419

Number of kernel evaluations: 97777 (82.057% cached)

Classifier for classes: 10+, 5-

BinarySMO
Machine linear: showing attribute weights, not support vectors.

      -1      * (normalized) Site_turu=Sosyal_paylasim
+     -0.7714 * (normalized) Site_turu=Internet_portali
+     -1.9142 * (normalized) Site_turu=Resmi_kurum
+      1.2254 * (normalized) Site_turu=Video_ve_muzik_paylasimi
+      1.2248 * (normalized) Site_turu=Isletmeler
+      0.0854 * (normalized) Site_turu=Ilanlar
+      0.0831 * (normalized) Site_turu=Bilgi_kaynagi
+      0      * (normalized) Site_turu=Gazete_ve_haber
…
```

Binary yani ikili ve lineer çekirdek fonksiyonlu sınıflandırıcı olarak çalıştırılan Destek Vektör Makinesi algoritması, WEKA'da ortalama ziyaret sayısına ait üç sınıfın (5-,5-10 ve 10+) ikili fonksiyonlarını çıkartmıştır. Liste.5.'de verilen çıktı sonuçlarının başında verilen ortalama ziyaret sayısı=[10+,5-10] sınıflandırıcısında en çok etkili olan nitelikler ve sınıflar incelendiğinde, sırasıyla

1.9415 çarpanıyla Site_turu=Spor, 1.9405 çarpanıyla Site_turu=Internet_portali, ve 1.3332 çarpanıyla OrtSure=5- değerleri yer almaktadır.

4.2.2. Gözetimsiz Öğrenme Algoritmalarının Sonuçları

Kümeleme analizinde, gözetimsiz öğrenme algoritmaları kullanılarak benzer özellikleri gösteren verileri aynı kümede toplanır ve verilere ilişkin bir fikir elde edilmeye çalışılır. Gözetimsiz öğrenme algoritmaları, gözetimli öğrenme algoritmalarının tersine modeli denetimsiz olarak öğrendikleri için niteliklerden herhangi birisi hedef nitelik değildir ve verilerin gözetimli öğrenmedeki gibi önceden belirlenen sınıflara atanması söz konusu değildir. Böyle bir atama olmadığı içinde doğruluk vb. oranlar elde edilememektedir. Bu da verilerin doğru ayrım yapılıp yapılmadığının tespitini zorlaştırmaktadır. Ancak, kümelemede temel amaç küme içi benzerliklerinin maksimum ve kümeler arası benzerliklerin minimum olduğu küme sayısını bulmaktır. Bunun için farklı küme sayıları için oluşan kümelerdeki verileri inceleyerek karar vermek gerekir.

Bu çalışmada, web sitesi istatistikleri üzerinde uygulaması yapılacak gözetimsiz öğrenme algoritmalarının hem küme sayısını belirlemede, hem de algoritmaların kıyaslaması WEKA'da ki test seçenekleri ile belirlenmiştir.1000 adet veriden 700'ü öğrenme ve geri kalan 300'ü test için kullanılacak olup, farklı küme sayıları için oluşan kümelerdeki verilerin yüzde oranları kayıt edilmiştir. WEKA'da eğitim için *Use training set* seçeneği, test için ise *Supplied test set* seçeneği kullanılmıştır. Kayıt edilen küme verilerine ait yüzde oranları büyükten küçüğe sıralamış şekilde verilmiştir. Daha sonra test işleminde oluşan kümelerindeki bu yüzde oranlarının, eğitim kümelerinde oluşan oranlara ne kadar yaklaşabildiğini bulabilmek için bağıl hata oranları bulunmuş ve küme sayılarına göre ortalaması hesaplanmıştır. Kümeleme kalitesinin ölçümünde bu hata ortalaması baz alınacaktır. Algoritmalarda küme sayısı sınırları k=2 ve k=6 arası seçilmiş ve analiz edilmiştir. En az hata oranını veren algoritma ve küme sayısı daha sonradan tüm verilerin kümelenmesi işleminde kullanılmıştır.

4.2.2.1. Eğitim ve Test Kümeleri Sonuçları

K-Means Algoritması Sonuçları

WEKA'daki SimpleKMeans kütüphanesi kullanılarak elde edilen K-Means algoritması ilişkin kümeleme sonuçları Tablo.52.'de verilmektedir.

Tablo.52. K-means algoritması sonuçlarına ait oluşan küme yüzdeleri

K- means Algoritması	k=2		k=3		k=4		k=5		k=6	
	Küme veri sayıları (%)	Kümelerin hata ortalaması (%)	Küme veri sayıları (%)	Kümelerin hata ortalaması (%)	Küme veri sayıları (%)	Kümelerin hata ortalaması (%)	Küme veri sayıları (%)	Kümelerin hata ortalaması (%)	Küme veri sayıları (%)	Kümelerin hata ortalaması (%)
Eğitim kümesi (700 veri)	63 37	**0**	37 35 28	**8.46**	36 34 16 14	**2.52**	27 20 19 19 15	**4.17**	28 22 15 13 12 10	**2.67**
Test kümesi (300 veri)	63 37		35 33 32		36 33 16 15		28 20 19 17 16		28 22 15 14 11 10	

K-means algoritması ile eğitim ve test için kullanılan verilerle oluşan küme verisi sayıları en yakın k=2 için % 0 hata ortalamasıyla olmuştur. k=2'den sonra en yakın oran k=4 küme için olmuştur. Buradaki hata ortalaması ise % 2.52 olmuştur. En fazla hata ortalaması ise % 8.46 ile k=3 için olmuştur. Buda verilerin 3 kümeye ayrılmasının elverişsiz olduğunu göstermektedir.

Hiyerarşik Kümeleme Algoritması sonuçları

WEKA'daki HierarchicalClusterer kütüphanesi kullanılarak elde edilen Hiyerarşik kümeleme algoritması ilişkin sonuçlar Tablo.53.'de verilmektedir. Hiyerarşik kümelemelerden bağlantı tipi olarak *Complete* seçilerek En uzak komşu algoritması analiz edilmiştir.

Hiyerarşik kümeleme algoritmaları ile eğitim ve test verileri ile oluşan kümeler arasındaki hata ortalamasının en az olduğu durum k=3 için % 3.2 ile olduğu gözlenmiştir. Bunun ardından en az hata ortalaması k=4 için gerçekleşmiştir. k=2 ve k=5 için hata ortalaması ise k=4'teki duruma yakındır. Hata ortalamasının en yüksek olduğu küme sayısı ise 6 olarak gerçekleşmiştir.

Tablo.53. Hiyerarşik Kümeleme algoritması sonuçlarına ait oluşan küme yüzdeleri

Hiyerarşik Kümeleme Algoritması (En uzak komşu algoritması)	k=2		k=3		k=4		k=5		k=6	
	Küme veri sayıları (%)	Kümelerin hata ortalaması (%)	Küme veri sayıları (%)	Kümelerin hata ortalaması (%)	Küme veri sayıları (%)	Kümelerin hata ortalaması (%)	Küme veri sayıları (%)	Kümelerin hata ortalaması (%)	Küme veri sayıları (%)	Kümelerin hata ortalaması (%)
Eğitim kümesi (700 veri)	88 12	**4.73**	78 12 10	**3.20**	66 12 12 10	**4.16**	36 30 12 12 10	**4.55**	36 29 12 12 10 1	**19.90**
Test kümesi (300 veri)	87 13		77 13 10		66 13 11 10		35 31 13 11 10		35 29 13 11 10 2	

Hiyerarşik kümeleme algoritmaları temelde birleştirici ve ayrıştırıcı olarak ikiye ayrılabilir. Burada kurulan modelin analizinde kümeleme mantığı, ayrıştırıcı özelliğe uygundur. Çünkü büyük tek bir küme topluluğu gibi elde edilen 1000 adet web sitesi istatistik verisi, K-means algoritmasında olduğu gibi, belirlenen k küme sayısına kadar bölümlenmektedir.

4.2.2.2. Kümeleme Sonuçlarının Karşılaştırması

Tablo.54.Gözetimsiz öğrenme algoritmalarının 700/300 veri için karşılaştırılması

700 Eğitim / 300 Test verisi	k küme sayılarına göre kümeler arası hata ortalaması (%)				
Gözetimsiz Öğrenme Algoritmaları	**k=2**	**k=3**	**k=4**	**k=5**	**k=6**
K-means Algoritması	0	8.46	2.52	4.17	2.67
Hiyerarşik Kümeleme Algoritması	4.73	3.20	4.16	4.55	19.90

Tablo.54. incelendiğinde hata ortalamasının % 0 olduğu k=2 küme sayısı için test işleminde en isabetli kümeleme K-means algoritması ile gerçekleştiği

görülmektedir. Bu bakımdan K-means algoritmasının, Hiyerarşik kümeleme algoritmasına göre üstünlük sağladığı söylenebilir.

Küme sayısının belirlenmesi açısından WEKA ile değerlendirmeye alınabilecek diğer bir kriter ise hata karelerinin toplamı (Sum of Squared Error/SSE) değeridir[25] ve (4.7)'deki gibi hesaplanır:

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, m_i)^2 \tag{4.7}$$

k küme sayısını, C herhangi bir kümeyi, m küme merkezini, d uzaklığı, x ise i. kümeye ait herhangi bir veriyi ifade etmektedir. SSE değerinin minimize edilmesi, küme kalitesini arttırmaktadır. SSE, küme sayısı artınca azalan bir değerdir ve küme sayısının veri sayısına yaklaştığında ya da eşit olduğunda sıfıra eşit olabilmektedir. Bu durum da yüzlerce küme oluşur ki bu da istenilmeyen bir durumdur. Bu nedenle küme sayısının belirlenmesinde başka bir yaklaşımlar izlenmiştir.

Tablo.55.'de görüleceği gibi 1000 veri için 2 ve 6 arasındaki küme sayıları için SSE değerleri kayıt edilmiş ve küme sayıları arttırılarak SSE değerindeki azalmanın büyüklüğü incelenmiştir. Bu tablo incelendiğinde 1000 veri için en uygun küme sayısının 2 olduğu görülmektedir. Çünkü SSE, yani hata karelerinin toplamı % 40.49 gibi büyük bir oranda azalış göstermiştir. Hata oranın çok azalması da küme kalitesinin arttığını göstermektedir. Burada belirlenen en uygun küme sayısı **(k=2)**, Tablo.54.'deki 700/300 veri ile yapılan inceleme sonucu ile de örtüşmektedir.

Tablo.55. K-means algoritmasında farklı küme sayıları için ölçülen SSE değeri

	Küme sayısı (k)	**Hata kareleri toplamı (SSE)**	**SSE'deki önceki duruma göre azalma**	**SSE'deki önceki duruma göre azalma (%)**
K- means Algoritması (1000 veri)	**k=1**	3951	-	-
	k=2	2351	1600	40.49
	k=3	2182	169	7.18
	k=4	1840	342	15.67
	k=5	1815	25	1.35
	k=6	1804	11	0.60

4.2.2.3. Gözetimsiz Öğrenme Algoritma Sonuçlarının Değerlendirilmesi

En uygun k sayısı tespit edildikten sonra çalışmada kullanılan veriler (1000 veri) WEKA'da k=2 kümeye ayrılmıştır. Yazılı sonuçlar Liste.6.'da grafiksel sonuçları ise Şekil.26. ve 27.'de verilmektedir.

Liste.6. Verilerin K-means algoritması ile WEKA'da kümeleme çıktısı

```
kMeans
======
Number of iterations: 3
Within cluster sum of squared errors: 2351.0
Missing values globally replaced with mean/mode
Cluster centroids:
                                        Cluster#
Attribute               Full Data           0                1
                          (1000)          (618)            (382)
=============================================================
Site_turu                 Oyun            Oyun    Kisisel_ilgi_beceri_alanlari
Erisim                    1-              1-3              1-
Farkliziy(cerez)          1-              1-               1-
Goruntusayfasayi          50-             50-              50-100
OrtSure                   5-              5-10             5-
Ortziyaret                5-10            5-10             5-

Clustered Instances

0      618 ( 62%)
1      382 ( 38%)
```

K-means algoritması ile en ideal şekilde 2 kümeye ayrılan verilerin her nitelik için küme merkezini temsil eden sınıflar ve bu sınıfların kendi nitelikleri içindeki veri sayısına bağlı yüzde oranları Tablo.56.'da verilmektedir.

Tablo.56. Küme merkezleri ve veri yüzdeleri

Nitelikler	Küme0		Küme1	
	Sınıf	Oran (%)	Sınıf	Oran (%)
Site türü (20 sınıf)	Oyun	10.51	Kişisel ilgi...	10.13
Erişim (3 sınıf)	1-3	38.09	1-	70.63
Farklı ziyaretçiler (3 sınıf)	1-	58.09	1-	89.52
Görüntülenen sf. say. (3 sınıf)	50-	75.77	50-100	58.98
Ortalama süre (3 sınıf)	5-10	50.17	5-	67.99
Ortalama ziyaret sayısı (3 sınıf)	5-10	63.11	5-	73.34

K-means algoritması ile iki kümeye ayrılan 1000 adet verinin 618 tanesi Küme0 olarak mavi renkle, kalan 382 tanesi ise Küme1 olarak kırmızı renkle simgelenmiştir.

Küme0 ve Küme1 nitelik sınıflarına göre analiz edildiğinde, Küme0'ın erişim, ortalama süre ve ortalama ziyaret sayısında sayısal olarak üstünlüğü mevcuttur. Farklı ziyaretçiler niteliğine ait olan sınıflar ise eşittir. Bu durum, web siteleri açından istenen bir şey olduğu için, web siteleri tıklama analizi yapanların beklentileri karşılayacak olan kümenin, daha çok Küme0 olduğu sonucunu ortaya çıkarır.

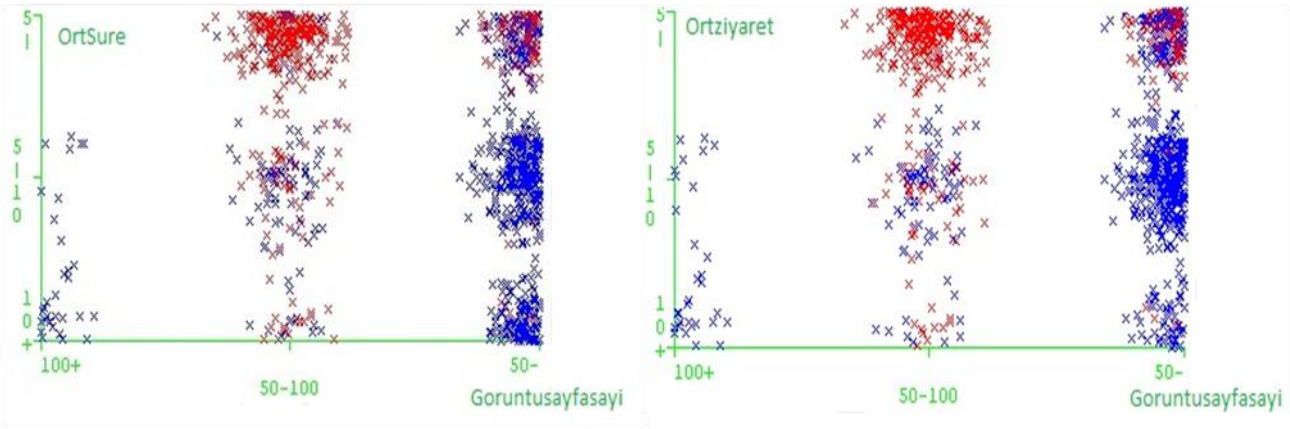
Küme0'a ait küme merkezlerinde, site türünün oyun olduğu görülmektedir. Oyun ve bu kümede yer alan benzeri türdeki web siteleri, Küme1 ile karşılaştırıldığında erişim yüzdesi bakımından diğer kümeye üstünlük sağlamıştır. Buda daha geniş bir kitlenin ilgi alanına girmek demektir. Ancak farklı ziyaretçiler (tahmini çerezler) sayısında bir üstünlük mevcut değildir. Daha geniş bir kitleye erişim sağlanmasına rağmen farklı ziyaretçiler sayısının daha çok olmaması, bu türdeki web sitelerinin genel olarak ziyaretçilerinin aynı olduğu ve bu web sitelerinin yeni ziyaretçi kazanmada pek başarılı olmadığı söylenebilir.

Görüntülenen sayfa sayısına bakıldığında, Küme0, Küme1'in gerisinde kalmaktadır. Bu da Küme0 içinde yer alan bu web sitelerinde ziyaretçilerin aradıkları şeyleri kolaylıkla bulup, onları inceledikleri ve başka sayfalara yönelmeyi daha az tercih ettiklerini göstermektedir.

Ortalama süre açısından, Küme0, hem bütün verilere ait kümeye hem de Küme1'e üstünlük sağlamış, ancak maksimum noktalara (10+ sınıfına) ulaşamamıştır. Küme0'ı temsil eden oyun türündeki web sitesi ziyaretçilerinin yukarıda bahsedildiği gibi aradıkları şeyleri kolaylıkla bulmasının yanında ortalama sürenin daha fazla olması, bu kümedeki web sitelerinin, diğer kümedekilere göre içerik bakımından daha çok ilgi, beğeni çektiğini ve vakit geçirildiğini göstermektedir.

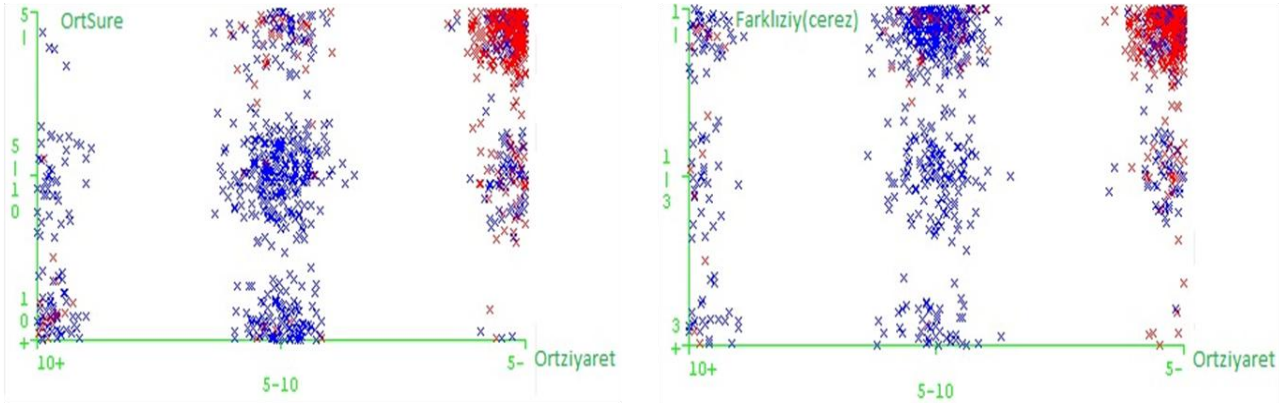
Ortalama ziyaret sayısına bakıldığında Küme0'ın Küme1'e göre ziyaretçi başına daha fazla ziyaret edildiği görülür. Gerçek ziyaretçi sayısı ile orantılı olan erişim değerine ait sınıfın, Küme0'da Küme1'e göre daha fazla olmasına rağmen, ortalama ziyaret sayısında da üstünlük kurması, web sitelerinin ziyaretçilerinin siteye geri döndüklerini ve çok ziyaret ettiklerini açıkça ortaya koymaktadır. Küme1'de ise bunun tam tersi olarak erişim sayısı daha az olduğundan, ortalama ziyaret sayısının daha yüksek beklenmesine rağmen bu gerçekleşmemiştir. Ortalama ziyaret sayısı

5'ten az çıktığından Küme1'de yer alan web sitelerine ziyaretçilerin pek bağlı kalmadıkları söylenebilir.



Şekil.26.a.)-b.) Web sitelerinin farklı niteliklere göre dağılımları

Şekil.26.a.'da web sitelerinin ortalama süre-görüntülenen sayfa sayısı niteliklerine göre dağılımlarına bakıldığında kümeleri ayıran nitelikler arasında ortalama sürenin diğer niteliğe göre az farkla baskın olduğu söylenebilmektedir. Küme1'in ortalama süre=5- hattında, Küme0'ın ise görüntülenen sayfa sayısı=50- hattında, ortalama süre=5-'ye kadar yoğunlaştığı görülmektedir. Benzer bir grafik ortalama ziyaret sayısı-görüntülenen sayfa sayısı grafiğinde oluşmuştur. Verilerin ait oldukları kümeyi belirleyen etken, görüntülenen sayfa sayısından daha çok ortalama ziyaret sayısı niteliği olmuştur. Veri yoğunluğu açısından bakıldığında elde edilen bir sonuç ise görüntülenen sayfa sayısı 50 milyonu aştığında ortalama sürenin ve ortalama ziyaret sayısının çok artış gösteremediği ve 5'in altında yoğunlaştığıdır.



Şekil.27.a.)-b.) Web sitelerinin farklı niteliklere göre dağılımları

Şekil.27.a.'da ortalama süre niteliği ve ortalama ziyaret sayısının oluşturduğu küme dağılımlarında kümelerin ayrımında ortalama ziyaret sayısının, ortalama süreye göre daha etkin olduğu söylenebilmektedir. Küme0 ortalama ziyaret sayısı için 5-10 sınıfında, ortalama sürenin 5-10 ve 10+ sınıfında yoğunlaşırken, Küme1 ise her iki niteliğinde 5-10 sınıfında yoğunlaşmıştır.

Şekil.27.b.)'de ortalama ziyaret sayısının farklı ziyaretçiler (tahmini çerezler) niteliğine göre kümelerin ayrılmasında baskın olduğu görülmektedir. Bu ayrımın, daha çok ortalama ziyaret sayısının 5- sınıfından 5-10 sınıfına geçerken olduğu söylenebilmektedir. Verilerin dağılımda ise farklı ziyaretçiler (tahmini çerezler) niteliği 1 milyonun altında iken, verilerin önemli yoğunluğu koruduğu görülmektedir.

1000 veri 5 (rastgele seçildi) site türü olarak düşünüldüğünde bu sitelerin tıklama analizindeki özellikleri nedir? gibi bir soruya, tüm veriler k=5 kümeye ayrılarak cevap vermek mümkün olur. Bu durumda Tablo.57.'deki veriler ile aşağıdaki çıkarımlar elde edilir.

Tablo.57. Tüm verilerin 5 kümeye ayrılması

Site türü / Nitelikler	**Kişisel ilgi ve beceri alanları**	**Gazete ve Haber**	**Oyun**	**e-Ticaret**	**Forumlar**
Erişim (%)	1-	1-3	1-	3+	1-
Farklı Ziyaretçiler (tahmini çerezler)	1-	1-3	1-	3+	1-
Görüntülenen Sayfa Sayısı	50-100	50-	50-	50-	50-100
Ortalama Süre (dk)	5-	5-10	5-10	10+	5-
Ortalama ziyaret sayısı	5-	5-10	5-10	5-10	5-

Veriler arasındaki oluşan 5 grupta ilk web sitesi türü kişisel ilgi ve beceri alanlarıdır. Bu web sitesi türü, niteliklerden birisi hariç diğerlerin tümünde en asgari değerdedir. Buda kişisel ilgi ve beceri türündeki web sitelerinin popüler olmadığını göstermektedir. Ortalama ziyaret sayısının 5'in altında kalması, bu tür web sitelerinin sık ziyaret edilmediği göstermektedir. Bu tür web siteleri, gerek internette gerekse

başka yayın araçlarında reklam vererek kendilerine ziyaretçi trafiği sağlamalıdır. Ortalama sürenin de yetersizliğini gidermek için mümkünse site içeriğine video türünde dosyalar eklenerek, ziyaretçilerin web sitesinden hemen ayrılmamaları sağlanabilir.

Gazete ve haber siteleri ile oyun türündeki web siteleri, ilk site türüne nispeten iyi konumdadır. Gazete ve haber siteleri ortalama bir erişim, süre ve ziyaret sayısını sağlayabilmiştir. Ancak oyun türündeki web sitelerinde ziyaretçi sayısında sıkıntı olduğu söylenebilir. Bu da bu web sitelerinin her yaş grubunda yönelik olmadığından dolayı dezavantaja sebebiyet verdiğini düşündürmektedir. Bu nedenle her yaş grubuna hitap edebilecek oyun vs. içerik konulması önerilebilir. Her iki türdeki web sitelerinde de görüntüleme sayfa sayısının az olması, sayfaların içeriğinin ziyaretçiye yeterli gelmesinden ya da site içeriğin plansızlığından kaynaklanmaktadır. Bu durum, ayrı ayrı web sayfalarına alınan reklamlar olduğunda reklam gelirlerinin azalmasına sebep olabilir. Görüntülenen sayfa sayısı azlığı için, menü başlıklarının arttırılarak her sayfada kolaylıkla erişilebilen bir noktaya konulması bu sorunu ortadan kaldırabilir.

E-Ticaret siteleri, ilk iki niteliği bakıldığında, geniş bir kitlece ilgi gören popüler web sitelerindedir. Ortalama ziyaret sayısı 5-10 değeri ile ziyaretçilerini kaybetmediği söylenebilir. Görüntülenen sayfa sayısının alt değerde seyretmesi, ziyaretçilerin bu web sitelerinde aradıkları ürünleri kolaylıkla bulabildiklerini düşündürmektedir. Ancak ortalama süre 10+ değeri ile üst değerlerde seyretmesi, ürünlerin satın alınması sürecinin karmaşık ve zahmetli olduğuna işaret etmektedir. Birçok web sitesinde satın alma işlemi için üyelik mecburiyeti vardır. Mümkünse zorunlu üyeliğin kaldırılarak direkt satın aldırmaya olanak verilmesi satın alma işlem karmaşıklığını ve süresini oldukça azaltacak ve şirket karını arttıracaktır.

Forum siteleri, nitelik değerlerine bakıldığında aynen kişisel ilgi ve beceri alanları siteleri değerlerini sağlamış olup, diğer site türlerine oranla ilgi çekmeyen web sitelerindendir. Erişim ve ortalama ziyaret sayısı değerinin düşük olması da ziyaretçilerin ortak beğeni ve ilgi görecek türde web sitesi olmadığını göstermektedir. Bu sorunun giderilmesi içinde, içeriğin farklı kesimlere hitap edecek şekilde zenginleştirilmesi ve ziyaret trafiğini arttırmak için web sitesine link içeren reklamlarında farklı yerlere verilmesi önerilmektedir.

5. SONUÇ

Bu çalışmada, makine öğrenme algoritmaları ile Türkiye'deki web siteleri istatistikleri üzerinde detaylı bir inceleme ve analiz yapılmıştır. Gözetimli öğrenme algoritmaları ile ortalama süre ve ortalama ziyaret sayısı parametreleri hedef nitelik seçilmiş olup algoritmaların birbirleri ile performans karşılaştırmaları, sonuçlara ilişkin değerlendirmeler ve yorumlar yapılmıştır.

Gözetimli öğrenme algoritmalarında, çapraz doğrulama yöntemi ile hedef nitelik ortalama süre alındığında C4.5, ortalama ziyaret sayısında da Destek Vektör Makinesi algoritması % 70 civarında bir oranda başarı göstermiştir. Diğer gözetimli öğrenme algoritmalarında da bu başarı oranına yakın performanslar elde edilmiştir. Böylece, bu alanda yapılacak olan çalışmalarda iyi bir başarı elde edilebileceği de gösterilmiştir. Bu algoritmaların yeni verilerdeki tahminleme başarısını gözlemlemek için ayrı ayrı yapılan eğitim ve test işlemlerinde ise C4.5 ve Destek Vektör Makinesi algoritmaları test işleminde de başarısını koruduğu söylenebilmektedir. Diğer algoritmalardan K En Yakın Komşu ve ID3 ise başarılarını % 10 ila % 15 arasında kaybettiği gözlenmiştir. Bu da bu iki algoritmanın web siteleri tıklama analizi için yanıltıcı sonuçlar ortaya koyabileceğini sonucunu çıkarmaktadır.

Hedef nitelikler açısından ortalama ziyaret sayısı algoritma performansı ortalama süre algoritma performansından daha yüksek çıkmaktadır. Ortalama % 5'lik bir fark, süreden ziyade ziyaret sayılarının bu analizlerde daha uygun olduğunu göstermiştir.

Eğitim verilerinin artışının öğrenme performansını nasıl etkilediği ise 400/600/800 veri ölçülmüştür. Algoritmalardaki öğrenme performansı genelde çok küçük miktarlarda değişmiştir. Bazen azalmakla beraber bazen de değişmeyip, nadiren de artmıştır. Çapraz doğrulama yöntemindeki hedef niteliklere göre başarılı iki algoritmaya değinilecek olunursa, sadece C4.5 algoritmasının ortalama süre hedef değişkeni ile % 5-6 oranında öğrenme performansı artmıştır (Ortalama ziyaret sayısı hedef niteliğinde ise neredeyse hiç değişmemiştir). Destek Vektör Makinesinin ortalama ziyaret sayısı hedef niteliğinde ki öğrenme performansı ise veri artışı ile pek değişmemiştir. (Ortalama süre için ise % 5-6 oranında performans düşüşü olmuştur.).

Gözetimli öğrenme algoritmalarında ortalama süre hedef niteliğinde en yüksek başarıyı gösteren C4.5 algoritması sonuçlarında kök düğüm ortalama ziyaret

sayısı olmuştur. Buda ortalama süre ilgili en ilgili niteliğin ortalama ziyaret sayısı olduğunu göstermektedir. En çok vakit geçirilen web sitelerine (10+ sınıfı için) bakıldığında oluşan yapraklarda genellikle 10+ sınıfına ait yaprakların oluştuğu görülmektedir. Bu da ziyaret sayısı artışı ile sitede geçirilen sürenin artışının orantılı değiştiğini göstermektedir. Ortalama süreyi belirleyen ikinci nitelik ise site türü olmuştur. Oluşan karar ağacı incelendiğinde, gazete ve haber, oyun, spor siteleri çok vakit geçirilen web sitesi türleri olduğu görülmektedir.

Gözetimli öğrenme algoritmalarında ortalama ziyaret sayısı hedef niteliğinde en yüksek başarıyı gösteren Destek Vektör Makinesi sonuçlarında 10+ ve 5-10 sınıflayıcısındaki ayrımı yapan faktörlerden bazıları sırasıyla Site_turu=Spor, Site_turu=Internet_portali, OrtSure=5-, Goruntusayfasayi=100+, Site_turu= Video_ve_muzik_paylasimi, Site_turu=Ilanlar ve Site_turu=e-Ticaret olmuştur. Dikkat edilirse site türünün [10+,5-10] aralığında ortalama ziyaret sayısında belirleyici olan nitelik olduğu söylenebilir.

Gözetimsiz öğrenme algoritmalarında eğitim ve test veri kümeleri için K-means ve Hiyerarşik Kümeleme yönteminden En Uzak Komşu Algoritması kümeleme sonuçlarında eğitim ve test verileri için ayrı ayrı oluşan kümelerde yüzdelik veriler karşılaştırılmıştır. En az hata oranı k=2 için K-means algoritmasında gözlenmiştir.

K-means algoritması ile k=2 için yapılan kümelemede veriler 618-382 veri şeklinde dağılmıştır. 618 veri içeren küme niteliklerinde sınıflara bakıldığında, diğer kümeye oranla daha yüksek değerlere sahiptir. Farklı niteliklerin küme ayrımlarındaki etkisine grafikler yardımıyla bakıldığında ise ortalama ziyaret sayısı ve ortalama süre niteliklerinin diğer niteliklere oranla daha belirleyici oldukları görülmüştür. Bu da web sitesi istatistiklerinde bu niteliklerin, verilerin hangi kümeye ait olmasında ağırlığı olduğunu ve öneminin büyük olduğunu göstermektedir. Bu iki nitelik arasında yapılan kümeleme kıyaslamasında ise, verilerin kümelere aitliğine daha çok etkiyi yapan ortalama süreye nazaran, ortalama ziyaret sayısıdır.

Son olarak; tüm veriler K-means ile k=5 için kümelere ayrılarak kümeler arasındaki kıyaslamalar yapılmıştır. Sayısal veriler açısından bakıldığında kişisel ilgi ve beceri alanlarına yönelik web siteleri ile forum siteleri benzer özellik göstermiştir ve bunlar popülerliği olmayan web sitelerindendir. Bu tür web sitelerine, ortalama sürenin artması olumlu olduğundan eğitici açıklayıcı videolar eklenebilir. Gazete ve haber siteleri ise veriler açısından daha iyi bir konumdadır. Görüntülenen sayfa sayısı

bu tür web siteleri için de yetersiz sayılabilir. Bunu gidermek için sitede menüler kolay erişilebilecek şekilde tasarlanabilir. Oyun sitelerinin ziyaret sayısı normal düzeydedir; fakat erişim sıkıntıları nedeni ile belirli bir ziyaretçi kesimine yönelmiştir. Erişim sayısının bu kesim tarafından oldukça fazla tercih edildiği gözlenmiştir. e-Ticaret siteleri ise en popüler sitelerdendir; ancak ortalama süre bu web sitelerinde az olması avantajlı sayıldığından, sürenin kısaltılması için üyeliksiz satış hizmeti sağlanabilir. Forum siteleri ise ilgi çekmeyen web sitelerindendir ve bu tür web sitelerinin genelde görselliğe hitap etmemesi dezavantaj oluşturmaktadır. Ziyaretçi çekilmesi için farklı web sitelerine reklam verilmesi çözüm olabilir.

Yapılan bu çalışma, bundan sonra yapılacak çalışmalar için başlangıç niteliğindedir. Bir arayüz ile geliştirilen sistem dinamik hale getirilebilir. Böylece verilerdeki değişimler sürekli kaydedilerek daha gerçekçi sonuçlar elde edilebilecek ve sonuçlar üzerinde daha anlamlı yorumlar yapılabilecektir. Ayrıca yapılan çalışma bir prototip niteliğinde olup, farklı alanlarda gereksinim duyulan bütün uygulamalara da uyarlanabilecek şekilde geliştirilmiştir.

KAYNAKLAR

[1] Dalyan, T.(2006). *Makine Öğrenmesinde 1R Algoritması ve İkinci Kuralın (2R) Oluşturulması,* Yüksek Lisans Tezi, Kocaeli Üniversitesi.

[2] Alpaydın, E.(2004). *Introduction to Machine Learning*, The MIT Press.

[3] Erdoğan, Ş.Z.(2004). *Veri Madenciliği ve Veri Madenciliğinde Kullanılan K-means Algoritmasının Öğrenci Veritabanında Uygulanması,* Yüksek Lisans Tezi, İstanbul Üniversitesi.

[4] Cristianni, N., Shawe-Taylor, J.(2000). *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods,* UK: Cambridge University Press.

[5] Efron, B.(1986). Why isn't everyone a Bayesian?, *American Statistician*,40,1-11.

[6] Özkan, Y. (2008). *Veri Madenciliği Yöntemleri* (1.Baskı). İstanbul: Papatya Yayıncılık.

[7] Yücebaş, S.C.(2006). *Hipokrat-I:Bayes Ağı Tabanlı Tıbbi Teşhis Destek Sistemi,* Yüksek Lisans Tezi, Başkent Üniversitesi.

[8] Vapnik, V.N.(1995). *The Nature of Statistical Learning Theory* (2nd Ed), New York: Springer-Verlag.

[9] Vapnik,V.N. (1998). *Statistical Learning Theory*, New York: John Wiley & Sons.

[10] Eray, O. (2008). *Destek Vektör Makineleri ile Ses Tanıma Uygulaması*, Yüksek Lisans Tezi. Pamukkale Üniversitesi.

[11] Özbek M.E, Özkurt N., Savacı F.A.(2006). *Dalgacık Tepeleri ve Destek Vektör Makineleri ¨ile Müzik Çalgısı Sınıflandırma* Eleco'2006 Elektrik-Elektronik-Bilgisayar Mühendisliği Sempozyumu, Bursa, 2006

[12] Silahtaroğlu, G. (2008).*Veri Madenciliği* (1.Basım). İstanbul: Papatya Yayıncılık.

[13] SPSS.(1999).*AnwerTree Algorithm Summary*. SPSS White Paper, USA.

[14] Agrawal, R.(1993). Database Mining: A Performance Perspective, *IEEE Transactions on Knowledge and Data Engineering*, 914-925.

[15] Han, J., Kamber, M., (2006). *Data Mining Concepts and Techniques* (2nd Ed), San Francisco, USA: Morgan Kaufmann Publishers.

[16] Quinlan J.R. (1993). *C4.5: Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann.

[17] Sun, J., Li, H.,(2008) Data Mining Method for Listed Companies, Financial Distress Prediction, *Knowledge-Based Systems*, 21, No. 1.

[18] Witten, I.H., Frank, E. (2000). *Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations,* San Fransisco, CA: Morgan Kaufman Publishers, Inc.

[19] Steinbach M., Tan P., Kumar V. (2003). *Introduction to Data Mining* University Of Minnesota: Addison Wesley Longman Publisher.
[20] Moore, A.W. (2001). *Decision Trees*, Carnegie Mellon University, USA.
[21] Dunham M.H. (2003). *Data Mining Introductory and Advanced Topics*. Southern Methodist University. Pearson Education Inc.
[22] Yıldırım, S. (2003). *Tümevarım Öğrenme Tekniklerinin C4.5'in İncelenmesi*, Yüksek Lisans Tezi. İstanbul: İTÜ.
[23] Graepel, T. (1998). *Statistical Physic of Clustering Algorithms*, Thesis. Diplomarbeit Technische Universität, Berlin.
[24] Öztemel, E.(2006). *Yapay Sinir Ağları* (2.Baskı). İstanbul:Papatya Yayıncılık.
[25] Monz, C., *Machine Learning for Data Mining*, Week 6: Clustering.
[26] Bülbül,Ş., Güler, M.F., Kandemir, A.Ş., *Propensity Skor Uygulamalarında Kümeleme Analizinin Test Amaçlı Kullanımı*,172.
[27] Berkhin,P. (2002) *Survey of Clustering Data Mining Techniques*, California,USA : Accrue Software Inc.
[28] Anderberg, M. R. (1973), *Cluster Analysis for Applications*, New York: Academic Press.
[29] Johnson, R. A., Dean W. W. (1999). *Applied Multivariate Statistical Analysis* (Fourth Editon)., New Jersey: Prentice Hall, Upper Saddle River.
[30] http://www.public.asu.edu/~huanliu/dmml_presentation/zhao.pdf (Erişim tarihi: 10.03.2011).
[31] http://www.willamette.edu/~gorr/classes/cs449/Reinforcement/reinforcement0.html (Erişim tarihi: 18.03.2011).
[32] http://www.google.com/support/adplanner/ (Erişim tarihi:10.04.2011).
[33] http://www.google.com/support/adplanner/bin/answer.py?hl=tr&answer=96385 (Erişim tarihi: 11.02.2011)
[34] http://www.bilgisayarkavramlari.com/2009/06/01/WEKA/ (07.01.2011)
[35] Coşkun, C., Baykal, A., Veri Madenciliğinde Sınıflandırma Algoritmalarının Bir Örnek Üzerinde Karşılaştırılması, *Dicle Üniversitesi, Fen Fakültesi Matematik Bölümü*, Diyarbakır.

ÖZGEÇMİŞ

29 Aralık 1982 tarihi, İstanbul İli doğumluyum. İlkokulu Zeytinburnu'nda, Ortaokulu ve Liseyi Küçükçekmece'de tamamladıktan sonra, 1999 yılında Kocaeli Üniversitesi, Mühendislik Fakültesi, Elektrik Mühendisliği Bölümüne kaydoldum. Bu bölümden 2003 yılında mezun olduktan sonra, askerlik görevimi Hv.K.K. 3. Ana Jet Üs'sünde 2005 yılında yedek subay olarak tamamladım. 2005 yılından 2007 yılına kadar özel bir şirkette çalıştım. 2007 yılından beri bir Kamu kurumunda Bilgi İşlem sorumlusu olarak görev yapmaktayım.

Akademik anlamda özel ilgi alanlarım, veri madenciliği, makine öğrenmesi, yapay zekâ, web programcılığı ve programlama dilleridir. Yabancı dilim İngilizcedir.

Tevfik ÇOBAN