KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MADENCİLİĞİNDE K-MEANS ALGORİTMASI VE TIP ALANINDA UYGULANMASI

YÜKSEK LİSANS

Bilgisayar Müh. Esra Dinçer

Anabilim Dalı: Bilgisayar Mühendisliği

Danışman: Yrd. Doç. Dr. Nevcihan DURU

KOCAELİ, 2006

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MADENCİLİĞİNDE K-MEANS ALGORİTMASI VE TIP ALANINDA UYGULANMASI

YÜKSEK LİSANS TEZİ

Bilgisayar Müh. Esra DİNÇER

Tezin Enstitüye Verildiği Tarih: 14 Haziran 2006

Tezin Savunulduğu Tarih: 12 Temmuz 2006

Tez Danışmanı	Üye	Üye
Yrd.Doç.Dr. Nevcihan DURU	Prof.Dr. Kadir ERKAN	Prof.Dr. Fuat İNCE
(.....)	(.....)	(.....)

KOCAELİ, 2006

ÖNSÖZ ve TEŞEKKÜR

Günümüzde başta iş dünyası olmak üzere birçok farklı alanda kullanılan veri madenciliği dünyayı değiştirecek 10 teknoloji arasında gösterilmiştir. Gelecekte daha çok önem kazanacak olan veri madenciliği üzerinde yapılan çalışmalara her geçen gün yenileri eklenmektedir. Veri madenciliğinde yeni gelişen teknolojilerin birçoğu henüz tıp alanında kullanılan yazılımlara dahil edilmemiştir. Tıp alanında geçmiş teşhis ve tedavi kayıtları, gelecek çalışmalara ışık tutmaktadır. Bu doğrultuda geçmiş kayıtların bilgisayar programları tarafından analiz edilmesi etkili tedaviyi destekleyici bir unsur oluşturmaktadır. Bu tezde tıp alanında geçmiş kayıtları kullanmanın önemi dikkate alınarak gırtlak kanseri ameliyat verileri için bir analiz aracı geliştirilmiştir.

Bu tez çalışmasında bana destek veren, teşvik eden ve yanlışlarımı düzelten tez danışmanım Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü Öğretim Üyesi Yrd. Doç. Dr. Nevcihan Duru'ya teşekkür ederim.
Ayrıca çalışmamın uygulamasında kullanılmak üzere gırtlak kanseri ameliyat verilerini veren Kocaeli Üniversitesi Tıp Fakültesi Hastanesi, Kulak, Burun ve Boğaz Bölümüne teşekkür ederim.

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

SEMBOLLER

C_j	: *j* doğal sayı olmak üzere *j*. Kümeyi belirtir
Eps	: bir veri nesnesi merkezli dairenin yarıçapı (komşuluk yarıçapı)
k	: küme sayısı
O	: hesaplanabilir karmaşıklık (computational complexity)
MinPts	: bir veri nesnesinin Eps komşuluğundaki nokta sayısı (kendisi hariç)
w	: çevresinde küme oluşturmak için seçilen nokta (küme prototipi)

Kısaltmalar

AGNES	:	AGglomerative NESting
BIRCH	:	Balanced Iterative Reducing and Clustering Using Hierarchies
CF	:	Clustering Feature
CLARA	:	Clustering LARge Applications
CLARANS	:	CLustering Algorithm based on RANdomized Search
CLIQUE	:	Clustering High-Dimensional Space
CRM	:	Customer Relations Management
CURE	:	Clustering Using REpresentatives
DBSCAN	:	Density Based Spatial Clustering of Applications with Noise
DENCLUE	:	Density Based Clustering
DIANA	:	DIvisive ANAlysis
DMQL	:	Data Mining Query Language
I/O	:	Input/Output
JDBC	:	Java Database Connection
KDD	:	Knowledge Discovery in Databases
LS	:	Linear Sum
MIT	:	Massachusets Institue of Technology
ODBC	:	Open Database Connection
OLAP	:	Online Analytical Processing
OPTICS	:	Ordering Points to identify the Clustering Structure
OLE-DB	:	Object Linking and Embedding for Databases
PAM	:	Partitioning Around Medoids
ROCK	:	Robust Clustering Algorithm
SOM	:	Self Organizing Maps
SQL	:	Structured Query Language
STING	:	Statistical Information Grid
VLDB	:	Very Large Data Bases
VTBK	:	Veri Tabanlarında Bilgi Keşfi

VERİ MADENCİLİĞİNDE K-MEANS ALGORİTMASI VE TIP ALANINDA UYGULANMASI

Esra DİNÇER

Anahtar Kelimeler: Veri Madenciliği, K-means Algoritması, Tıpta Veri Madenciliği, Tıp Bilişimi, Kanser

Özet : Veri madenciliği, veri yığınlarından anlamlı bilgiler elde etme işlemidir. Çeşitli yöntem ve teknikler aracılığı ile veri kaynakları analiz edilerek taşıdıkları bilgi keşfedilmeye çalışılır. Başta pazarlama, bankacılık ve sigortacılık olmak üzere bir çok alanda etkin şekilde kullanılan veri madenciliği tıp alanında bilgilerin analizi ve yorumlanması aşamalarında kullanılmaktadır.

Bu çalışmanın amacı, veri madenciliğinde bir kümeleme tekniği olan k-means algoritmasını incelemek ve bu algoritmayı kullanarak geliştirilen bir yazılım aracılığı ile gırtlak kanseri ameliyat verilerinin analizini yapmaktır. Uygulamanın tıp doktorlarının kullanımına uygun şekilde verileri çeşitli açılardan analiz etmesi hedeflenmiştir.

K-means algoritması aracılığı ile veriler kümelenmiş ve içlerindeki yoğunlaşmalar grafik üzerinde gösterilmiştir. Algoritmanın başlıca tercih nedenleri; küme sayısının parametrik olması, sonuçların hem grafik olarak hem de yazı ve rakamlarla kolayca ifade edilebilmesi, uygulanmasının kolay olması ve hızlı çalışmasıdır.

Çalışmada kullanılan gırtlak kanseri ameliyat verileri, Kocaeli Üniversitesi Tıp Fakültesi Hastanesi, Kulak, Burun ve Boğaz Bölümünden alınmıştır. Geliştirilen yazılım, tıp doktorlarına geçmiş kayıtları analiz ederek, ileriye dönük tahminde bulunabilmeyi kolaylaştırmaktadır ve karar almada yardımcı olabilecek bir analiz aracıdır. Bu yazılım ile geçmiş verileri analiz ederken değişken parametreler kullanılarak değerlendirme yapılabilir, tüm durumlar için mevcut ve gelecek vakalarla ilgili tahminde bulunulabilir, mevcut ve gelecek vakalar için ameliyat sonrasında tümörün nüks etme olasılığı ve hastanın hayatta kalma olasılığı değerlendirilebilir, doğru öngörülen ameliyat öncesi evreler görüntülenerek incelenebilir ve bu şekilde ameliyat öncesi tahmin başarısı değerlendirilebilir, başarılı ameliyat bilgileri izlenerek, gelecek ameliyat tercihlerinde fikir alınabilir. Ayrıca yazılım, araştırma, denetim ve eğitim etkinliklerinde de kullanılabilir.

THE K-MEANS ALGORITHM IN DATA MINING AND AN APPLICATION IN MEDICINE

Esra DİNÇER

Keywords: Data Mining, K-means Algorithm, Data Mining in Medicine, Medical Informatics, Cancer

Abstract: Data mining is extracting knowledge from large amounts of data. Its techniques and methods try to discover the knowledge by analyzing these resources. It has been used in many areas like marketing, banking and insurance, and used to analyze data and data interpretation in medicine.

The objective of this study was to examine the k-means algorithm which is one of the clustering techniques in data mining, and to analyze the laryngeal cancer operations data by using the software application which was included in this algorithm.

The data was clustered and the intensities in the data set was pointed out in charts by using this algorithm. The reasons for choosing the algorithm: The number of clusters is an input parameter, it is easy to display the clustering result both graphically and in words and figures, it is easy to implement and runs fast.

The medical data set was obtained from Kocaeli University Hospital Ear Nose and Throat Department. The developed software enables the users to analyze the past records by entering variable parameters, to predict for current and future cases, to study about the possibility of the tumor relapse and of the patient's survival after the operation, to consider true estimates of pre-operation stages of the cases,and to track which operations have been successful, and in this way the success of the operations can be studied and decisions made for the future. The software can be used to support research, to enhance supervision, and to aid in teaching activities.

1. GİRİŞ

İnsanoğlu hayatta karşılaştığı zorlukları yenebilmek için önceki tecrübelerine, bilgi birikimlerine ihtiyaç duyar. Bu yüzden tarih boyunca bunları saklayacak ve gerektiğinde kullanmayı sağlayacak teknikler geliştirmeye çalışmıştır. İlk çağlarda mağara duvarlarına resim şeklinde kaydettiği verileri, ilerleyen çağlarda kağıdın icadı ile birlikte kitaplara dökmüştür.

Geleneksel veri kaydetme aracı olan kağıdın yerini gün geçtikçe hızlanan ve ucuzlayan elektronik kayıt ortamlarına bırakması ile birlikte, yeryüzünde çok büyük veri yığınları oluşmaya başlamıştır. Yaşadığımız her saniye bu yığınlara yenileri eklenmektedir.

Veri kendi başına bir değer ifade etmez, bir amaca yönelik olarak işlendiğinde bilgiyi oluşturur. Veriyi bilgiye çevirme sürecine veri analizi denir. Bilgi, bir soruya yanıt vermek için veriden çıkarılan sonuçlardır. Yakın geleceğin, geçmişten çok fazla farklı olmayacağı varsayıldığında, geçmiş veriden çıkarılmış olan kurallar gelecekte de geçerli olacak ve ilerisi için doğru tahmin yapmayı sağlayacaktır.

Yeraltında değerli maden arama işlemine kavram olarak benzerliği nedeniyle, bilgi yığınları içinde büyük miktarda veri içinden, gelecekle ilgili tahmin yapmayı sağlayacak bağıntı ve kuralların bilgisayar programları kullanarak aranması, Veri Madenciliği olarak adlandırılmaktadır [1]. Diğer bir deyişle veri madenciliği, büyük veri yığınlarından anlamlı bilgiler elde etmek için, bilgisayar destekli bir bilgi çözümleme işlemidir. Kendiliğinden oluşan kümelenmeler, örüntüler, birliktelikler ve istisnalar veritabanlarındaki bilgi kaynaklarıdır. Veri madenciliği yöntem ve teknikleri bu kaynakları analiz ederek taşıdıkları bilgiyi keşfetmeye çalışırlar.

Günümüzde başta iş dünyası olmak üzere, birçok farklı alanda kullanılan veri madenciliği, MIT (Massachusets Institue of Technology) tarafından 2001 yılında yayınlanan bildirgeye göre dünyayı değiştirecek 10 teknoloji arasında gösterilmiştir [2]. Gelecekte daha çok önem kazanacak olan veri madenciliği üzerinde yapılan

çalışmalara her geçen gün yenileri eklenmektedir. Günümüzde, tıptan uzay bilimlerine kadar birçok farklı sektörde kullanılan veri madenciliğinin kullanım alanlarına her gün yenileri eklendiği düşünülürse, konunun önemi daha iyi anlaşılır.

Veri madenciliğinde yeni gelişen teknolojilerin birçoğu henüz tıp alanında kullanılan yazılımlara dahil edilmemiştir. Bilgisayar aracılığı ile bilgilerin analizine duyulan gereksinim, bir çok alanda olduğu gibi tıp alanında da ortaya çıkmıştır. Tıp alanında bilgisayarlardan (idari ve finansal konuların dışında) veri toplama ve yorumlama aşamalarında yaygın bir şekilde yararlanılmaktadır. Oluşturulan çok çeşitli düzendeki tıbbi bilgi, bilgisayar ortamında işlenmekte ve saklanmaktadır. Veritabanı analizleri ve karar desteği için veri madenciliği gerekli bir araç haline gelmiştir [3].

İnsanların deneyimlerden sonuç çıkartma yeteneği, geçmişten uygun örneklerin tanınması yeteneğine bağlıdır. Hastalıklara teşhis koyan bir doktor, öncelikle deneyimlerinden benzer vakaları tanımlar ve ardından bu vakaların bilgilerini eldeki probleme uygular. Bilinen vaka kayıtlarının tutulduğu veritabanı, sınıflandırılmış kayıtlar içinden yeni vakaya benzeyenleri bulmak için taranır. Mevcut hasta için en etkili tedavi, muhtemelen benzer hastaların sonuçlarından elde edilen bilgilerle yapılan tedavidir [1].

Ancak klasik hastane bilgi sistemleri daha çok idari ve finansal konulara ağırlık verecek şekilde tasarlanmaktadır. Hastaların ayrıntılı teşhis ve tedavi bilgileri ancak gereksinim duyulması halinde ve kimi zaman doktorların bireysel çabaları ile toplanıp kaydedilmektedir. Bu da teşhis ve tedaviyle ilgili veri madenciliği çalışması yapılmasının önünde bir zorluk oluşturmaktadır. Çoğu kez, çalışmanın yönlendirilmesi için konuyla ilgili bir tıp doktorunun bulunması ve destek alınması gerekebilir. İlgili tıp doktorunun bilgisayar uygulamaları konusunda bilgi sahibi olmaması çalışmayı olumsuz yönde etkilemektedir.

Tıbbi veritabanlarında veri madenciliği, diğer disiplinlerden çok farklı değildir. Ancak tıp alanındaki verilerde belirli bir standardın olmayışı ve mevcut standartlar arasında tam bir uyumun olmaması nedeniyle, bu alanda bir veritabanı oluşturmak ve bu veritabanını işlemek zor bir işlemdir. Diğer taraftan, tıbbi verilerin kendine has bazı özellikleri bulunmaktadır. Temel veri yapıları, diğer birçok alanla

karşılaştırıldığında, matematiksel olarak karakterize edilmeye pek uygun değildir. Bilginin düzenlenebilmesi için kümeleme veya dizi çözümlemeleri gibi karşılaştırılabilir yapılar yoktur. Hekimler, görüntü, sinyal veya diğer klinik bilgilerle ilgili yorumlarını, standartlaştırılması çok güç olan serbest metinler olarak yazmaktadır. Örneğin aynı hastalığın açıklamasında farklı isimler kullanılmaktadır. Tıbbi kavramlar arasındaki ilişkileri açıklamak için de farklı dilbilgisi yapıları kullanılmaktadır [4]. Bu şekilde oluşturulan bilgilerin bilgisayar programları aracılığı ile işlenmesi özel çalışma gerektirmektedir.

Tıbbi veritabanlarında değişik amaçlarla yapılmış çok sayıda veri madenciliği çalışması bulunmaktadır. Bunlardan biri, göğüs kanseri ve cilt lezyonları verilerini sınıflandırmak için k-nearest, bayesian, karar ağacı ve Dempster-Shafer teorisi yöntemlerini birbirleriyle karşılaştıran çalışmadır. Sonuçta bu tip tıbbi verileri sınıflandırmada Dempster-Shafer birleştirme kuralı teorisinin daha verimli olduğu gözlemlenmiştir. Çalışmanın deneysel değerlendirmesine dayanarak, k-nearest, bayesian, karar ağacı ve Dempster-Shafer teorisi yöntemlerinin kullanılan veritabanına göre farklı performans gösterdiği ileri sürülmüştür. Dempster-Shafer birleştirme kuralı teorisinin farklı veritabanlarında daha doğru sınıflandırma yaptığı sonucuna varılmıştır [5].

Veri madenciliği algoritmalarını sınamak için bir standart haline gelen PIDD (Pima Indian Diabet Database) diabet veritabanını kullanan Breault, diabet hastalığı konusunda yapılan tahminlerinin doğruluğunu göstermek için kaba kümeler (rough sets) yöntemini kullanmıştır. Kaba kümeler yöntemi, tıbbi verilerde sıkça kullanılan bir veri madenciliği tahmin aracıdır. Yöntem, akademik ortamda geliştirilen Rosetta yazılımı ile verilere uygulanmıştır. PIDD veritabanı üzerinde daha önceden uygulanan algoritmaların %66-81 arasında elde ettiği tahmin başarısı, kaba kümeler yöntemi ile %82'ye ulaşmıştır [6].

Tıbbi veritabanı üzerinde yazılım uygulaması geliştiren çalışmalara literatürde sık rastlanmamaktadır. Bu çalışmalara bir örnek olarak; Geliştirilen DMAP (Data Minig with Apriori) isimli bir yazılım aracılığı ile Apriori algoritmasını kullanarak diabet hastalarının sosyal durumlarını ortaya çıkarmıştır. Yazılım diabet veritabanından okunan verileri parametrik olarak algoritmaya uygulamaktadır. Genellikle pazar

sepet analizinde (market basket analyzing) kullanılan Apriori algoritması, bu çalışmada başarıyla tıbbi verilere uygulanmıştır [7].

Diğer bir örnek çalışmada, gen ifade profillerinin kümelenmesi için EXCAVATOR (EXpression data Clustering Analysis and VisualizATiOn Resource) isimli yazılım geliştirilmiştir. Büyük miktarda gen ifade verisi, genlerin fonksiyonel açıdan incelenebilmesi için mikro diziler kullanılarak oluşturulmaktadır. Gen ifade verisinin kümelenmesi, biyolojik süreçteki genler arası fonksiyonel ilişkilerin incelenmesi açısından yarar sağlamaktadır. Kümeleme, gen verilerini minimum örten ağaç (min. spanning tree) gibi göstermek için graflar oluşturarak gerçekleştirilmiştir. Bu sayede çok boyutlu veri seti, önemli verilerde kayıp olmadan ağaç yapısına indirgenmiştir. Böylece karmaşık gen verisi, kümeleme açısından daha kolay işlenir hale getirilmiştir [8].

Başka bir çalışmada, teşhis koymada kullanılan testlerin en etkili şekilde belirlenmesini sağlayan öğrenme tabanlı bir program geliştirilmiştir. Hasta bilgilerini ve tavsiye edilen test bilgilerini kullanarak, teşhis performansını uygun ölçümleme ile eniyileme (optimize) yapmaktadır. Tıbbi veritabanlarından kural formunda teşhis bilgilerini okumak için veri madenciliğinin kaba kümeler (rough sets) yöntemi kullanılmıştır. Markov karar süreçleri (desicion process) ve destekli öğrenme (reinforcement learning) yöntemleri bir arada kullanılarak, uygun test stratejilerinin elde edilmesi kolaylaştırılmıştır. Teşhis amacıyla kullanılan testlerin en uygun şekilde belirlenmesi tıp doktorlarının karşılaştığı karmaşık konulardan biridir. Geliştirilen yöntem aracılığı ile teşhis sürecinin geliştirilebileceği düşünülmektedir [9].

Veri madenciliğinde eldeki veri türüne ve elde edilen sonuçların kullanım amacına göre farklı bir çok teknik bulunur. Bu tez çalışmasında kümeleme tekniği kullanılmıştır. Bu teknikte veriler dağılımlarına göre irdelenerek doğal sınıflandırmalar oluşturulur. Kümeleme işleminde temel prensip, sınıf içi benzerliği maksimum, sınıflar arası benzerliği minimum yapmaktır. Sınıfların her birine "küme" adı verilir. Bir kümeleme yönteminin kalitesi bu prensibi sağlaması ile doğru orantılıdır. Veri madenciliğinde aralarında benzerlik olan birçok kümeleme tekniği bulunmaktadır. Bunlardan birisi bölümleme tekniğidir. Teknikte, veritabanındaki her

bir eleman bir farklılık fonksiyonuna göre her biri küme olarak adlandırılan k adet bölümden birine dahil edilir.

K-means algoritması en iyi bilinen ve yaygın kullanılan bir kümeleme algoritması ve bölümleme tekniğidir. İlk olarak J. MacQueen tarafından 1967 yılında tanıtılmıştır [10]. Bu yöntem yıllardır bilimsel ve endüstriyel uygulamalarda en yoğun kullanılan kümeleme algoritması haline gelmiştir. Veriler özelliklerine göre k adet kümeye ayrılarak kümelenir. Bu işlem, verilerin en yakın veya benzer oldukları küme merkezleri (centroid) etrafına yerleştirilmesi ile gerçekleştirilir. Çalışma yönteminde, öklit bağıntısı temel alınarak kümeleme yapılır. Algoritmanın başında k sayısı giriş parametresi olarak verilir. Eğer küme sayısı belirli değil ise deneme yoluyla en uygun sayı bulunur.

K-means algoritması aşağıdaki özellikleri nedeniyle tercih edilmiştir:

1. Küme sayısının okunan bir parametre olması analizi esnek hale getirmektedir.
2. Algoritmanın uygulanması kolaydır ve hızlı çalışmaktadır.
3. Değişik dağılımlarda başarılı sonuçlar alınabilmektedir.
4. Kategorik verilerle çalışacak şekilde adapte edilebilmektedir.
5. Kümeleme sonuçları hem grafik olarak hem de yazı ve rakamlarla kolayca ifade edilebilmektedir.

Tıp alanında k-means algoritmasını kullanan pek çok çalışma gerçekleştirilmiştir: Evans ve meslektaşları ilaçların olumsuz etkileri konusunda risk faktörlerini araştıran çalışmalarında, her bir ilacı kategorize ederken k-means algoritmasından yararlanmışlardır. İlaçların sınıflandırılması yoluyla, bazı risk faktörlerinin bütün olumsuz etkilerle ve ilaçların tedavi sınıflarıyla tutarlı olduğu bazılarının ise ait olduğu sınıfa özel faktörler olduğu izlenmiştir. Çalışmada yüksek risk etkenlerinin hastanın cinsiyet, yaş, kilo gibi karekteristik özellikleri ile ilacın doz ve kullanım şekline bağlı olarak izlenmesi gerektiği sonucuna varılmıştır [11].

Diş hekimliğinde ameliyat sonrası akut ağrıların analizi için yapılan bir çalışmada, akut ağrı çeken hasta gruplarının özellikleri k-means algoritması ile ortaya çıkarılmıştır. Çalışmada hastaların psikolojik ve sosyal durumlarının diş ameliyatları sonrası görülen akut ağrı sorunlarına etkisi araştırılmıştır. Ağrı sorunu yaşayan 438

hastanın doldurduğu anket formundaki bilgiler uygun formata dönüştürülerek k-means algoritması ile kümelenmiştir. Kümeleme analizi sonucunda endişeli, üzüntülü veya depresyondaki hastalarda ağrı şiddetinin yüksek olduğu, bayanların ağrıdan daha fazla yakındığı görülmüştür [12].

Psikiyatri alanında gerçekleştirlen bir çalışmada, antisosyal kişilik bozukluğu gösteren adli suçluların bilgileri ile kümeleme analizi yapılmıştır. Analiz sonucunda iki farklı tipte antisosyal kişilik bozukluğu tespit edilmiştir. Bunlardan birincisi güçlü suç eğilimi gösteren grup, ikinci psikopatik kişilikli grup olarak belirlenmiştir. İki suçlu grubun profilleri arasındaki farklılıklar kümelerin analiziyle ortaya çıkarılmıştır. Kümeler k-means algoritması kullanılarak ve küme sayısı verilmeden oluşturulmuştur [13].

Aynı alanda diğer bir örnek, intihara teşebbüs eden ve ağır şekilde yaralanan hastaların klinik profillerinin çıkarılmasıdır. Çalışmada intihara teşebbüs eden kişiler arasında farklı klinik profil grupların görülmesi amaçlanmıştır. 121 hastadan toplanan bilgilere, k-means algoritması ile kümeleme analizi yapılmıştır. Analiz sonucunda üç küme ortaya çıkmıştır. Çoğunluğunu kadınların oluşturduğu ilk kümede, ilaç alarak kendini zehirleyen, intihar nedeni az ancak intihara çok istekli 43 hasta tespit edilmiştir. Çoğunluğu erkeklerden oluşan ikinci kümede, böcek ilacı içen, orta seviyede intihar isteği olan 53 hasta görülmüştür. Yine çoğunu erkeklerin oluşturduğu üçüncü kümede, yüksek düzeyde intihar isteği olan ve kendisine vahşi şekilde zarar veren 17 kişi bulunmuştur. Bu çalışmada ortaya çıkan kümelerin dışında farklı profiller de bulunabilmektedir [14].

Moleküler biyoloji ve genetik alanında yapılan çalışmalarda da k-means algoritması kullanarak yapılan kümeleme örneklerine sık rastlanmaktadır. Örneğin gen ifadelerinin profilinde, patolojik tipi ve sınıfı bilinmeyen moleküler alttipleri ortaya çıkartmak için k-means algoritması kullanılmıştır. Veri içindeki moleküler alttip kümelerini ortaya çıkartmak amacıyla uygulanan k-means işlemi sonucunda üç farklı küme tespit edilmiştir. Kümeler arası tip ve sınıflar doğru bir şekilde tanımlanarak genler ayırt edilmiştir [15].

Bu alanında diğer bir örnek, ikili ağaç yapısındaki vektör niceleme yaklaşımı ile, mikrodizilerdeki verilerin kümelenmesi ve görüntüye dönüştürülmesinde k-means'den yararlanılmıştır. Mikrodiziler genlerin birbiriyle etkileşimini incelemek için kullanılan bir araçtır. Çalışma içinde ağaç yapısındaki vektör niceleme ve k-means kümeleme birleştirilerek hibrit bir yöntem geliştirilmiştir. Bu yaklaşım, veri önişleme ve normalleştirmeye karşı daha az duyarlı olmuştur ve klinik açıdan uygun kümeleri üç büyük veri tabanına yerleştirilmiştir [16].

Genetik alanında yapılan bir çalışmada ön implantasyon süresinde hayvan embriyolarının gen profilleri cDNA mikrodizi ile analiz edilmiştir. Genlerin kümelenmesinde k-means algoritması kullanılmıştır. Çalışmada hayvan embriyolarının gen biçimleri 7, 14, 21 ve 28 gün sonrası olmak üzere farklı evrelerde cDNA mikrodizisi kullanılarak incelenmiştir. Ön implantasyon süresinde farklı biçimlenen genlerin örüntülerini tespit etmek için veriler kümelenmiştir. Gen değişimleri ve örüntü biçimlerinin kategorize edilmesiyle evrelerdeki değişimler analiz edilmiştir [17].

K-means algoritmasından tıpta görüntü işleme konusunda da yararlanılmaktadır. İstemik beyin dokusu uygulaması kapsamında geliştirilen yarı otomatik görüntü işleme sistemi bu tip bir çalışmaya örnek oluşturmaktadır. Sistem, mikro derecedeki görüntüler üzerinden kıvrım tabanlı görüntü sınıflaması yaparak, bunlar üzerinden ilgili sayısal verileri çıkarmakta ve bu verilere karşılık gelen makro derecede görüntüyü oluşturmaktadır. Çeşitli görüntü işleme teknikleri ile yoğunluk dağılımları elde edilmiştir. Görüntü sınıflandırma k-means kümeleme yöntemi kullanılarak, ile hesaplama işlemlerinin kolaylaştırılması sağlanmıştır [18].

Bu tezde tıp alanında geçmiş kayıtları kullanmanın önemi dikkate alınarak gırtlak kanseri ameliyat verileri üzerinde bir analiz aracı geliştirilmiştir. Gırtlak kanseri, Kulak Burun Boğaz Hekimliğinde en sık görülen kanser türlerinden biridir. Gırtlak (larenks), boğazın hemen altında ses tellerinin bulunduğu bir organdır ve gıda alımı sırasında besinlerin nefes borusuna kaçmasını engeller. Gırtlak kanseri gırtlağın herhangi bir kısmında gelişebilir ve çoğu zaman ses kısıklığı ile erken bulgu verir. Genellikle 50-60 yaş grubundaki erkeklerde sık görülür. Sigara, bu kanser türü için en önemli risk etkenidir. Yoğun alkol kullanımı da riski arttırır. Tedavi tümörün

türü, yeri ve evreye göre belirlenir. En önemli tedavi şekli cerrahidir ve bunun yanında ışın tedavisi kullanılabilir.

Bu tez çalışmasının amacı:

1. Veri madenciliği, kümeleme teknikleri ve k-means algoritmasıyla ilgili literatür çalışması yapılması, k-means algoritmasıyla yapılmış çalışmaların incelenmesi ve konuyla ilgili yazı hazırlanması,
2. Geleneksel kümeleme teknikleri içerisinde yer alan k-means algoritmasını gırtlak kanseri ameliyat bilgilerini içeren veritabanı üzerinde çalıştırıp algoritmayı incelemek ve performansını değerlendirmek,
3. K-means algoritmasını kullanan bir yazılım uygulaması geliştirerek, gırtlak kanseri ameliyat bilgilerini içeren veritabanını, tıp doktorlarının kullanımına uygun şekilde çeşitli açılardan analiz etmektir.

Çalışmanın literatür taramasında kaynak olarak internette yer alan çeşitli bilimsel makalelerden, konu ile ilgili kitaplardan ve sempozyum bildirilerinden yararlanılmıştır. Çalışmada kullanılan gırtlak kanseri ameliyat verileri Kocaeli Üniversitesi Tıp Fakültesi Hastanesi, Kulak, Burun ve Boğaz Bölümünden alınmıştır.

Gırtlak kanseri ameliyat verileri üzerinde, k-means algoritması kullanarak yapılan başka bir veri madenciliği çalışmasına literatür taramasında rastlanmamıştır. Veri madenciliği çalışmalarında genellikle veriler SPSS ve MATLAB gibi paket programlar aracılığı ile analiz edilmektedir. Bu çalışmada geliştirilen yazılım, paket programlardaki kısıtları içermemektedir, kullanıcı açısından öğrenme süresi çok kısadır ve kullanılması kolaydır.

Bu tez çalışması altı bölümden oluşmaktadır. Tezin ikinci bölümünde veri madenciliğine giriş yapılarak genel tanımlara yer verilmiştir. Veri madenciliğinin uygulama alanları ve diğer disiplinlerle ilişkilerinin açıklanmasının ardından veri ambarı konusu anlatılmıştır. Veri madenciliğinde bilgi keşfi kavramı ve aşamaları sıralanmış ve veri madenciliğinde kullanılan tekniklere ana hatlarıyla değinilmiştir.

Üçüncü bölümde veri madenciliği tekniklerinden kümeleme analizi detaylı olarak incelenmiştir. Kümeleme analizinin tanımı, özellikleri ve kümeleme analizinde kullanılan veri türlerine yer verilmiştir. Kümeleme analizinin teknikleri ayrıntılı olarak açıklanırken, her bir kümeleme tekniğini kullanan algoritmanın teorik yapısı ve çalışma şekli hakkında bilgiler verilmiştir.

Dördüncü bölümde uygulamanın analizinde kullanılan k-means algoritması çeşitli açılardan ayrıntılı olarak incelenmiştir. K-means hakkında genel bilgilerin verilmesinin ardından, algoritmanın adımları ve yapısı üzerinde durulmuştur. Algoritma programlamaya uygulanırken hangi adımların izleneceği anlatılmış ve k sayısının nasıl hesaplandığı gösterilmiştir. Son olarak algoritmanın aritmetik ve geometrik hesaplanma yöntemleri örneklerle açıklanarak, matematiksel yorumu yapılmıştır.

Uygulamanın anlatıldığı beşinci bölümde, ilk olarak uygulamanın amaçları ve sağlayacağı yararlardan bahsedilmiştir. Ardından bu çalışmaya konu olan gırtlak kanseri ve tedavisi hakkında genel bilgilere yer verilmiştir. Gırtlak kanseri ameliyat verilerini içeren veritabanı üzerinde uygulama öncesinde yapılan önişleme çalışmaları anlatılmıştır. Verilerin analizinde izlenen yol ve analiz sonuçlarının gösterim şekilleri açıklanmıştır. Uygulamada geliştirilen ekranların nasıl çalıştığı ayrıntılı bir şekilde anlatılmıştır ve beraberinde programların önem taşıyan kod parçalarına yer verilmiştir.

Tezin sonuçlar ve öneriler bölümünde yapılan çalışma özetlenerek, k-means algoritmasının seçilme nedenleri ve sağladığı avantajlar anlatılmıştır. K-means algoritması kullanılarak geliştirilen yazılımın sağladığı yararlar sıralanmıştır. Çalışmada karşılaşılan zorluklara yer verilerek, gelecek çalışmalar için önerilerde bulunulmuştur.

2. VERİ MADENCİLİĞİ

2.1. Veri Madenciliği Tanımı

Teknoloji devrimi ile birlikte verilerin dijital ortamda saklanmaya başlanması nedeniyle, yeryüzündeki bilgi miktarının sürekli arttığı günümüzde veri tabanlarının sayısı da benzer, hatta daha yüksek oranda artmaktadır. Daha yüksek kapasite ve işlem gücüne sahip donanımların geliştirilmesi ile birlikte veri saklama hem daha kolay, hem de daha güvenli hale gelmiştir.

Veri tabanı sistemlerinin artan kullanımı ve veri depolama ünitelerinin hacimlerindeki olağanüstü artış geleneksel sorgulama ve raporlama araçlarının dev veri yığınları karşısında etkisiz kalmasına yol açmıştır. Bunun sonucunda veri tabanlarında bilgi keşfi (VTBK) (KDD-Knowledge Discovery in Databases) adı altında yeni arayışlar ortaya çıkmıştır.

VTBK süreci içerisinde büyük önemi bulunan modelin kurulması ve değerlendirilmesi aşamalarına genel olarak veri madenciliği adı verilmektedir. Bu önemden ötürü birçok kaynakta VTBK ile veri madenciliği eş anlamlı olarak kullanılmaktadır [19].

Veri madenciliği ile ilgili birçok farklı tanım bulunmakla beraber en fazla kabul gören tanımlar aşağıda belirtilmektedir;

- Veri madenciliği, büyük veritabanlarından, çok net olmayan, üstü kapalı, önceden bilinmeyen ancak potansiyel olarak kullanışlı olabilecek bilginin çıkarılmasıdır [20].

- Veri madenciliği, yapay zekadan örüntü tanımaya, istatistikten veritabanı teknolojilerine kadar uzanan disiplinler arası bir uygulama alanıdır.

- Veri Madenciliği, büyük veri tabanlarında örüntülerin, birlikteliklerin, anormalliklerin, ve çeşitli yapıların yarı otomatik bir sistem ile keşfidir.

2.2. Veri Madenciliği Uygulama Alanları

Veri madenciliği her geçen gün yeni ve farklı alanlarda kullanılmaya başlamakla birlikte günümüzde yaygın olarak kullanıldığı alanlar birkaç kategoride toplanabilir:

a) Pazarlama:
Müşterilerin satın alma örüntüleri, demografik bilgileri, kampanya ürünleri belirleme, mevcut müşterileri kaybetmeden yeni müşteriler kazanma, pazar sepeti analizi (Market Basket Analysis), müşteri ilişkileri yönetimi (CRM – Customer Relations Management) ve satış tahmini alanları en yaygın veri madenciliği uygulama alanlarıdır.

b) Banka ve Sigortacılık:
Farklı finansal göstergeler arasında korelasyon tespiti, kredi kartı dolandırıcılıklarının tespiti, kredi taleplerinin değerlendirilmesi, kredi kartı harcamalarına göre müşteri profili belirlenmesinde, sigorta dolandırıcılıklarının tespitinde, yeni poliçe talep edecek müşterilerin tahmininde yoğun olarak kullanılmaktadır.

c) Biyoloji, Tıp ve Genetik:
Bitki türleri ıslahı, gen haritasının analizi ve genetik hastalıkların tespiti, kanserli hücrelerin tespiti, yeni virüs türlerinin keşfi ve sınıflandırılması, fizyolojik parametrelerin analizi ve değerlendirilmesinde kullanılmaktadır.

d) Kimya
Yeni kimyasal moleküllerin keşfi ve sınıflandırılması, yeni ilaç türlerinin keşfinde kullanılmaktadır.

e) Yüzey Analizi ve Coğrafi Bilgi Sistemleri

Bölgelerin coğrafi özelliklerine göre sınıflandırılması, kentlerde yerleşim yerleri belirleme, kentlerde suç oranı, zenginlik-yoksulluk, köken belirleme, kentlere yerleştirilecek posta kutusu, otomatik para makinaları, otobüs durakları gibi hizmetlerin konumlarının tespitinde kullanılmaktadır.

f) Görüntü Tanıma ve Robot Görüş Sistemleri

Çeşitli algılayıcılar aracılığı ile tespit edilen görüntülerden yola çıkarak engel tanıma, yol tanıma, yüz tanıma, parmak izi tanıma gibi tekniklerde kullanılmaktadır.

h) Uzay Bilimleri ve Teknolojisi

Gezegen yüzey şekillerinin ve gezegen yerleşimleri, yeni galaksiler keşfi, yıldızların konumlarına göre gruplandırılmasında kullanılmaktadır.

i) Meteoroloji ve Atmosfer Bilimleri

Bölgesel iklim, yağış haritaları oluşturma, hava tahminleri, ozon tabası deliklerinin tespiti, çeşitli okyanus hareketlerinin belirlenmesinde kullanılmaktadır.

j) Sosyal bilimler ve Davranış bilimleri

Kamuoyu yoklamaları inceleme, genel eğilim belirleme, seçim öngörüleri oluşturmada kullanılmaktadır.

k) Metin Madenciliği (Text Mining)

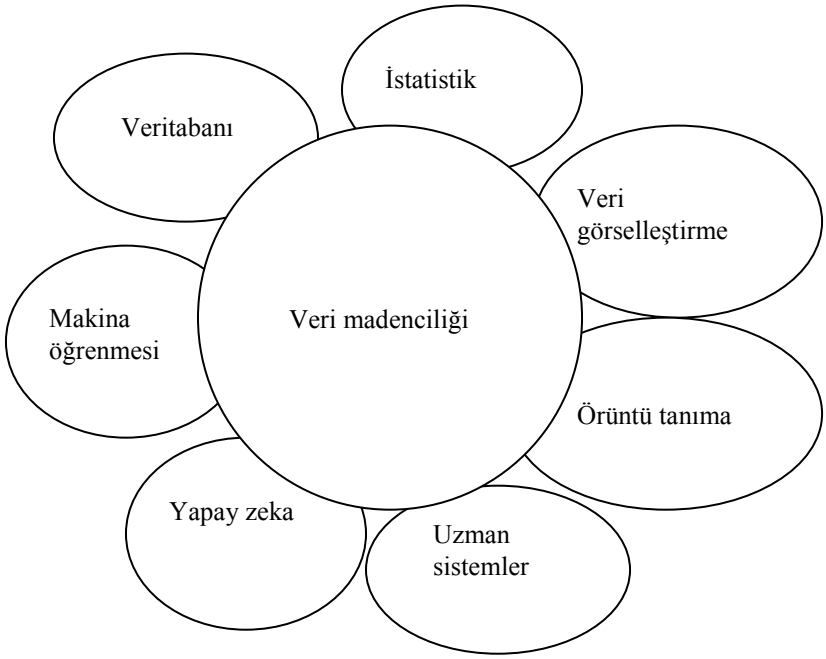
Çok büyük ve anlamsız metin yığınları arasından anlamlı ilişkiler elde etmekte kullanılmaktadır.

l) Internet madenciliği (Web Mining)

İnternet üzerindeki veriler hem hacim hem de karmaşıklık olarak hızla artmaktadır. Sadece düz metin ve resimden başka akan (streaming) ve sayısal veriler de web verileri arasında yer almaktadır. İnternetin belirli kategorilere ayrılarak veriye ulaşım süresinin azaltılması web madenciliğinin temel hedefidir.

2.3. Veri Madenciliği Ve Diğer Disiplinler

Veri madenciliği, makina öğrenmesi, örüntü tanıma, veritabanı teknolojileri, istatistik, yapay zeka, uzman sistemler, veri görselleştirme (data visualization) alanlarının bir kesişim noktası olarak doğmuş ve bu bağlamda gelişmesini sürdürmektedir [20]. Bu yapı temel olarak Şekil 2.1'de görüldüğü gibi sembolize edilebilir.



Şekil 2.1 Veri madenciliğinin diğer disiplinlerle ilişkisi [21].

Makina öğrenmesi, örüntü tanıma ve istatistik alanları, veri madenciliğinde örüntü keşfetme aşamasında, yapay zeka teknolojileri, bulunan örüntüleri yorumlama aşamasında, veritabanı teknolojileri eldeki verileri depolama, süzme, temizleme, sorgulama işlemi aşamasında, veri görselleştirme ise, raporlama ve insan beyni için anlamlı sembollere çevirme aşamasında yardımcı olmaktadır.

2.4. Veri Ambarı

2.4.1. Veri ambarı tanımı

Veri ambarı; karar verme sürecinde kullanılan, konu tabanlı, birleştirilmiş, zamana bağımlı, verilerin sabit olduğu veri topluluğudur [22]. Veri topluluklarının veri ambarı olarak adlandırılabilmesi için taşıması gereken bu dört özelliği kısaca açıklamak gerekirse:

a) Konu tabanlı: Veri ambarları, satış verileri, müşteri bilgileri gibi belirli bir konuda veriler içerir.

b) Birleştirilmiş (Integrated): Veri ambarı birçok farklı kaynaktan gelen bilgilerin toplanması ile kurulur. Örneğin bir veri ambarı içinde ilişkisel veritabanları, düz metin dosyaları, işlemsel veritabanları bulunabilir.

c) Zamana bağımlı: Veri ambarlarında bilgiler periyodik aralıklarla eklenir. Veri ambarındaki her bir anahtar yapı tarihsel olarak dizilmiş olmalıdır. Örneğin günlere göre son beş yılın satış rakamlar.

d) Sabit: Veri ambarında veriler işlevsel veritabanlarında olduğu gibi sürekli güncellenmez. Veri ambarına eklendiği andan itibaren sabit olarak kaydedilir.

Veritabanı ile veri ambarı arasındaki başlıca farklar kısaca aşağıdaki gibi açıklanabilir:
Veri ambarı bir işletmenin günlük kullanımda veri depoladığı işlevsel (operational) veritabanından ayrı tutulur. Bu yüzden veri ambarındaki bilgiler güncel değildir. Belirli zaman aralıklarında işlevsel veritabanlarındaki bilgiler güncel önemlerini yitirdiklerinde veri ambarına gönderilirler. Veritabanları okuma/yazma amaçlı, veri ambarları ise sadece okuma amaçlı kullanılırlar. Veritabanları günlük giriş-çıkış işlemleri için kullanılırken veri ambarı uzun süreli veri analizi ve geleceğe yönelik öngörüler elde etme amaçlı kullanılır.

2.4.2. Veri ambarlarının kullanım nedenleri

Veri ambarları bir karar verme mekanizması veya diğer adıyla karar destek sistemi olarak kullanılmaktadır. Veri Ambarı üzerinde veri madenciliği, çok boyutlu veri analizi (Online Analytical Processing - OLAP), müşteri ilişkileri yönetimi (CRM), istatistiksel analiz ve raporlama işlemleri gerçekleştirilir[23]. Bu işlemlerin tamamına yakını, işlevsel veritabanları üzerinde de gerçekleştirilebilmesine rağmen, veri ambarı kurma ve kullanmanın temel nedeni her iki sistem için de yüksek performans elde etme isteğidir.

İşlevsel veritabanları, sıralama, arama ve hazır sorguları çalıştırma işlemleri için, veri ambarları ise özetleme, özel veri organizasyonu ve çabuk erişim için optimize edilirler. Veri ambarı kurulmadığı durumlarda işlevsel veritabanı performansı önemli ölçüde düşerken, karar destek işlemleri doğruluktan uzaklaşmaktadır. Ayrıca, karar verme işlemleri tarihsel veriler gerektirdiği için veri ambarı karar destek sistemleri için vazgeçilmez bir unsurdur.

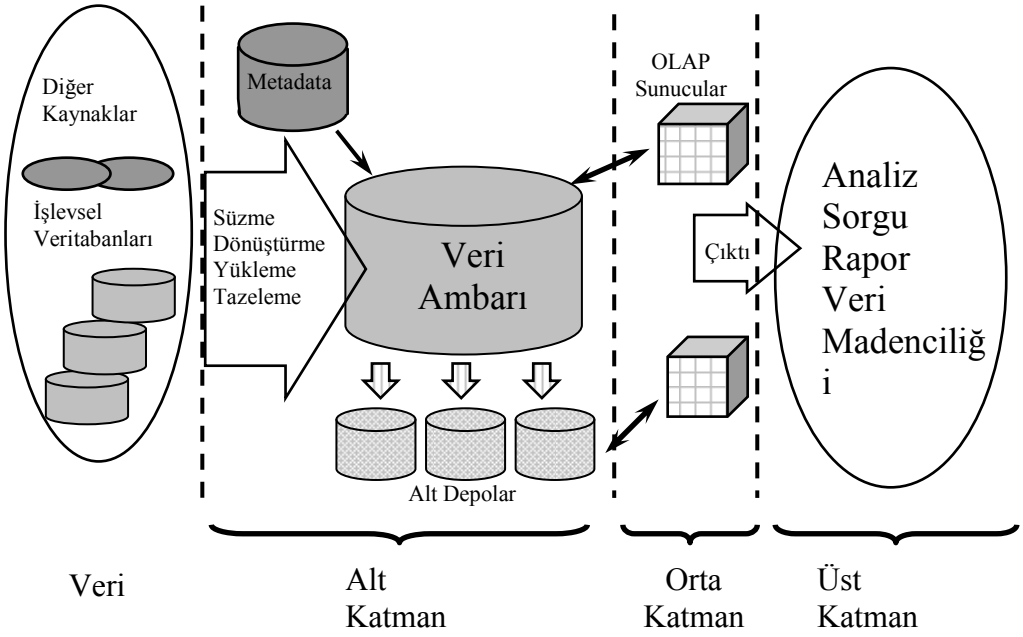
2.4.3. Veri ambarı mimarisi

Veri ambarı mimarisi J.Han'ın yaklaşımına göre 3 katmanlı bir yapıdan oluşmaktadır.

Şekil 2.2'de görülen bu katmanlar şunlardır:

a) Alt katman

Veri ambarı veritabanı sunucusudur ve genellikle ilişkisel bir veritabanı sisteminden oluşur.İşlevsel veritabanlarında veya dış kaynaklardan gelen veriler uygulama program arayüzleri (geçit) tarafından seçilir. Geçit programları bir veritabanı yönetim sistemi ile desteklenir. Bu sayede istemci programların sunucu tarafına SQL kodu şeklinde sorgu gönderebilmesine olanak sağlanır. Geçit programlarının en bilinenleri Microsoft firmasının ODBC (Open Database Connection) ve OLE-DB (Object Linking and Embedding for Databases) ve Sun Microsystems firmasının JDBC (Java Database Connection) adlı ürünleridir.



Şekil 2.2 Veri ambarını oluşturan katmanlar [23].

b) Orta katman

Bir OLAP sunucudur. Bu katmanda bir alt katmandan gelen veriler OLAP sunucular tarafından çok boyutlu analiz yöntemleri kullanılarak raporlama, analiz ve veri madenciliği işlemleri için anlamlı veriler haline getirilir.

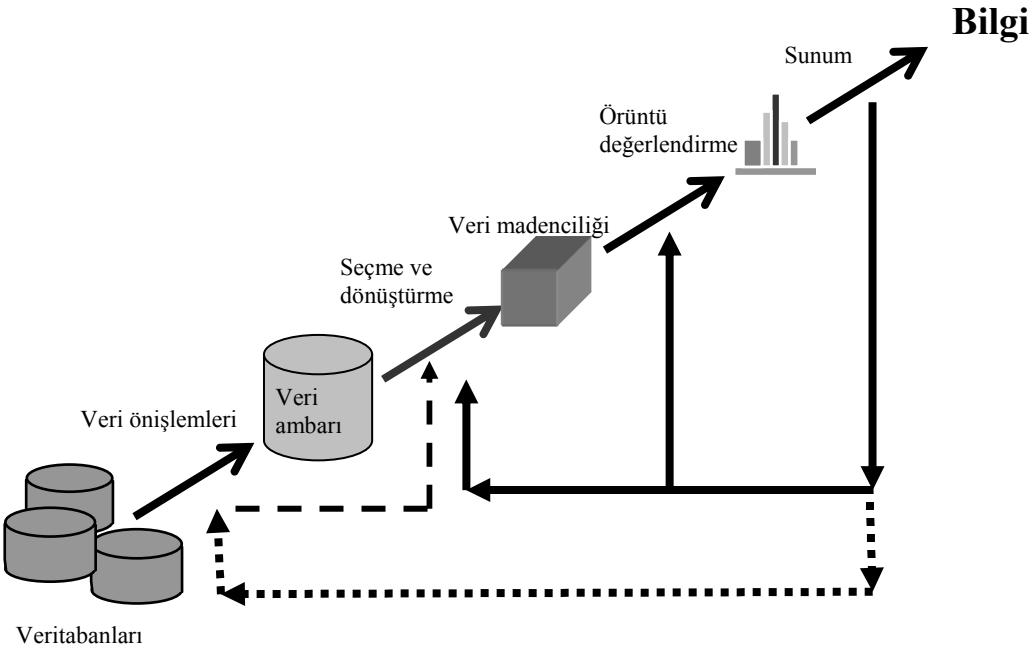
c) Üst katman

Bu katman istemciden oluşur. Bu katmanda sorgula ve raporlama araçları, analiz araçları ve veri madenciliği araçları içerir.

Veri ambarlarında yer alan bilgiler, bilgilerin kullanılacağı alanlara göre ayrı alt depolara dağıtılabilirler. Çoğunlukla işletmelerin içindeki departmanların kullanımına göre bölümlenen veri ambarlarında alt depolar (data-mart) oluşturulur.

2.5. Veritabanlarında Bilgi Keşfi Aşamaları

Veri madenciliği, veritabanlarında bilgi keşfi (VTBK) (KDD–Knowledge Discovery in Databases) işleminin temel bileşenlerinden biridir. Bununla beraber VTBK sadece veri madenciliğinden ibaret değildir. Şekil 2.3'te görüldüğü gibi VTBK süreci 5 aşamadan oluşmaktadır [23].



Şekil 2.3 Veri tabanlarında bilgi keşfi aşamaları [23].

VTBK sürecini oluşturan aşamalar:

a) Veri Önişlemleri (Data Preprocessing):

Bu aşamada öncelikle veriler içindeki gürültüler, tutarsızlık ve düzensizlikler giderilir. Bu işleme veri temizleme (Data Cleaning) denir. İkinci aşamada veri birleştirme (Data integration) işlemi uygulanır. Bu aşamada çeşitli kaynaklardan gelen verilerin tek bir veri ambarında toplanabilmesi için gerekli genelleme, normalizasyon ve uyumluluk işlemleri yapılır.

b) Veri Seçme ve Dönüştürme (Data Selection):

Bu aşamada, veri madenciliğinin sağlıklı yapılabilmesi için veriler üzerinde önişlemler yapılır. Bu önişlemler:

- Veri madenciliği konusu ile ilgili bilgi seçimi.
- Madencilik yapılacak veri türünün belirlenmesi.
- Veriler arasında hiyerarşik yapı ve genellemelerin belirlenmesi.
- Veri madenciliği sonunda bulunacak bilgi için yenilik ve ilginçlik ölçümü yöntemlerinin belirlenmesi.
- Veri madenciliği sonunda bulunacak veri için sunum ve görselleştirme araçlarının belirlenmesi.

Tüm bu önişlemleri gerçekleyebilmek için bir veri madenciliği sorgulama dili (Data Mining Query Language- DMQL) kullanılır. Bu konuda henüz standart bir DMQL dili örneği oluşmamıştır. J.Han böyle bir dil önermiş ve kitabında tüm detayları ve yazım yapısını açıklamıştır.

c) Veri Madenciliği:
İnsanoğlu için anlamlı veri örüntüleri ortaya çıkarmak için çeşitli algoritmaların kullanıldığı aşamadır. İlerleyen sayfalarda bu işlem detaylı olarak anlatılmıştır.

d) Örüntü Değerlendirme(Pattern Evaluation):
İkinci aşamada belirlenen ilginçlik (interestingness) ölçüm yöntemleri kullanılarak veri madenciliği ile bulunan verilerin ne kadar ilginç ve yararlı olduğu tespit edilir.

e) Bilgi Sunumu(Knowledge Presentation):
Çeşitli görselleştirme ve raporlaştırma araçları kullanılarak bulunmuş olan veriler ilgili kullanıcılara sunulur.

VTBK süreci defalarca tekrar ve aşamalar arası atlamalar ve ileri geri hareketler içerebilmektedir. Günümüzde çoğunlukla veri madenciliği aşamasına odaklanılmakta, fakat diğer tüm aşamalar VTBK işleminin bütünlüğü açısından en az veri madenciliği kadar önemlidir [20].

2.6. Veri madenciliği Teknikleri

Veri madenciliği teknikleri eldeki veri türüne ve elde edilen sonuçların kullanım amacına göre farklılıklar gösterir. Temelde veri madenciliği iki kategoride incelenir [23]:

- Tanımlayıcı (Descriptive)
- Öngörüsel (Predictive)

Tanımlayıcı veri madenciliği, veritabanındaki verinin genel karakterini, mevcut durumu ortaya çıkarmaya yönelik yöntemleri ön plana çıkarır. Öngörüsel veri madenciliği ise verileri geleceğe yönelik tahminler yapma, sonuç çıkarma amaçlı işlemlerde kullanır.

Veri madenciliği teknikleri kullanıldıkları veri yapılarına ve keşfedebildikleri örüntü biçimlerine göre kategorilere ayrılır. Birçok kaynak veri madenciliği teknikleri için farklı gruplandırmalar yapmıştır. Bunlardan en yaygın kabul göreni J.Han'ın ortaya sürdüğü kategorilerdir. J.Han kategorilerini kullanan kaynaklar bile, hangi algoritmanın hangi kategoriye ait olduğu konusunda net görüş birliğine sahip değildir. Bu kategorileri aşağıdaki gibidir:

- Tanımlama ve Ayrımlama (Characterization and Discrimination)
- Birliktelik Analizi (Association Analysis)
- Sınıflandırma ve Öngörü (Classification and Prediction)
- Kümeleme Analizi (Cluster Analysis)
- Sıradışılık (istisna) Analizi (Outlier Analysis)
- Evrimsel Analiz (Evolution Analysis)

Bu tezde, aşağıda öğeleri detaylı olarak anlatılan J.Han kategorilerine yer verilmiştir.

2.6.1. Tanımlama ve Ayrımlama

Veriler gösterdikleri ortak özelliklere göre genelleştirilmiş sınıflara ayrılabilirler. Bir firma müşteri portföyünü alışveriş ortalaması belirli bir miktardan daha yüksek olan müşterileri "zengin", diğerlerini ise "orta halli" ya da "fakir" olarak tanımlayabilir. Bu tür genellemeler veri kümesinin elemanlarının ortak özellikleri ya da veri kümesinin diğer veri kümleri ile olan farklılıklarını yansıtacak şekilde yapılabilmektedir.

a) Tanımlama (Characterization)

Bir veri kümesinin elemanlarının genel özelliklerini özetlemek amaçlı kullanılır. Örneğin bir alışveriş merkezinde bu yıl satışı oranı %25'in üzerinde artan mallar ifadesi bir Tanımlama işlemidir.

b) Ayrımlama (Discrimination)

Bir veri kümesinin diğer bir veri kümesinden farklarını ortaya çıkarma işlemidir. Örneğin bu yıl satış oranı %10 artan mallar ile satış oranı %15 azalan malların karşılaştırılması Ayrımlama tabanlı veri madenciliğidir.

Her iki tür veri madenciliği yöntemi birbirine çok benzer yöntemler kullanırlar. Ayrıca her iki yöntemle elde edilen sonuçlar pasta grafiği, sütun grafiği, eğriler ve çok boyutlu küpler ile sunulurlar.

2.6.2. Birliktelik analizi

Birliktelik analizi bir veri kümesinde kendiliğinden, sıklıkla gerçekleşen, birlikte ya da aynı süre içinde alınma, yapılma, oluşma gibi etkileri keşfetme temeline dayanır. Bu yöntem bankacılık işlemlerinin analizinde ya da pazar sepeti analizi yönteminde yaygın olarak kullanılır. Pazar sepeti analizi, bir alışveriş sırasında veya birbirini izleyen alışverişlerde müşterinin hangi mal veya hizmetleri satın alma eğiliminde olduğunun belirlenmesiyle müşteriye daha fazla ürün satılması yollarından biridir [19]. Pazar sepeti analizi ile örneğin müşteriler bira satın aldığında %75 ihtimalle cips de alırlar şeklinde bir ilişki ortaya çıkarılabilir. Bunun sonucunda bira ile cips yan yana raflara yerleştirilebilir veya bira alanlar cips aldığında cips fiyatında indirim yapılacak şekilde kampanyalar oluşturularak satışlar arttırılabilir.

Birliktelik analizi yalnızca mal ve hizmetlerin birlikte satın alınması için değil aynı zamanda hangi koşulları sağlayan müşterilerin hangi ürünleri alacağı hakkında da çözümler getirmektedir. Örneğin bir banka kredi kartı kayıtları incelendiğinde yaşları 20 ile 29 arasında değişen müşterilerden, gelirleri 700 milyon ile 900 milyon TL arasında değişen müşterilerin bilgisayar satın aldıkları görülmüştür. Bu kural, birliktelik analizi yönteminde şöyle ifade edilir:

Yaş(X , “20…29”) ^ Gelir(X , “700…900”) → alır(X , “bilgisayar”)

2.6.3. Sınıflandırma ve öngörü

Sınıflandırma işlemi insan düşünce yapısına en uygun veri madenciliği yöntemidir. İnsanoğlu çevresindeki nesneleri ve olayları daha iyi anlamak ve başkalarına anlatabilmek için hemen her şeyi sınıflandırma eğilimindedir. Örneğin, insanları davranışlarına göre, hayvanları türlerine göre, evleri görünüşlerine göre sınıflandırmaktadır.

Veri madenciliğinde sınıflandırma, eldeki mevcut verileri önceden belirlenen bir özelliğe göre sınıflara ayırmak ve yeni eklenecek verilerin hangi sınıfa dahil olacağını tayin etme işlemdir. Diğer bir deyişle, yeni karşılaşılan bir girdinin hangi sınıfa dahil olacağına karar verme işlemidir.

Sınıflandırma işlemine, bankaların kredi başvurularını düşük, orta ve yüksek riskli olarak sınıflandırması, bir okulda yeni gelen öğrencilerin hangi sınıfta eğitim görmesi gerektiğinin belirlenmesi örnek olarak verilebilir.

Öngörü işlemi sınıflandırma işlemine çok benzer. Ancak öngörü işleminde sınıflandırma, gelecek için tahmin edilen belirli bir davranışa ya da belirli bir değere göre yapılır. Öngörü işleminde yapılan sınıflandırmanın doğru olup olmadığını test etmenin tek yolu "bekle ve gör" prensibidir [23].

Öngörü işlemine örnek olarak deprem tahmini, bir turizm şirketi müşterilerinden hangilerinin bu yaz yurtdışında tatil yapmak isteyeceğinin belirlenmesi verilebilir.

Sınıflandırma ve Öngörü işleminde Karar Ağaçları (Decision Tree), Yapay Sinir Ağları (Neural Networks), K-en yakın komşu (K-Nearest Neighbour), Genetik algoritmalar, Naive Bayesian sınıflama, Bellek Tabanlı Nedenleme (Memory Based Reasoning) yöntemleri kullanılır.

2.6.4. Kümeleme analizi

Kümeleme işlemi sınıflandırma ve öngörü işleminin aksine, veri kümesini önceden sınıflara ayırmaz, bunun yerine veriler dağılımlarına göre irdelenerek doğal sınıflandırmalar oluşturur. Kümeleme işleminin sınıflandırma işleminden en önemli farkı önceden belirlenmiş sınıflar ya da sınıf tanımları (etiketleri) olmamasıdır. Bu yüzden kümeleme işlemi gözetimsiz (unsupervised) veri madenciliği yöntemidir. Kümeleme işlemi sonunda elde edilen kümeler kullanılan yöntemin giriş parametrelerine bağımlı olsa da, giriş parametrelerinden bağımsız kümeleme teknikleri geliştirme çalışmaları sürmektedir [24].

Kümeleme işleminde temel prensip, sınıf içi benzerliği maksimum, sınıflar arası benzerliği minimum yapmaktır [23]. Bir kümeleme yönteminin kalitesi bu prensibi sağlaması ile doğru orantılıdır.

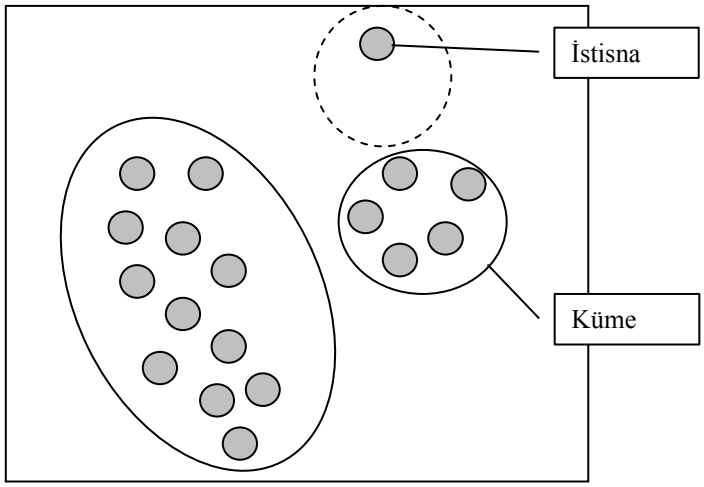
Kümeleme analizi sadece veri madenciliğinde değil, örüntü tanıma, görüntü işleme, coğrafi bilgi sistemleri gibi birçok alanda yoğun olarak kullanılmaktadır. Tez konusu bir kümeleme algoritması olduğu için 3.bölümde kümeleme analizi detaylı olarak açıklanmıştır.

2.6.5. Sıradışılık analizi

Bir veri kümesinde verilerin genel davranışından veya veri dağılım modelinden farklılık gösteren nesnelere sıradışı (Outlier) denir. Birçok veri madenciliği yöntemi istisnaları gürültü veya aşırı durumlar olarak görür, bu yüzden dikkate almaz. Fakat bazı durumlarda istisna noktalar diğerlerine göre çok daha fazla bilgi içerir. Örneğin kredi kartı veya sigorta sahtekarlıklarının tespitinde, tıp biliminde yeni bir hastalığın başlangıcını tespit etmede istisnalar analiz edilir. İstisna analizinde iki yöntem söz konusudur [23]:

a) İstatistik tabanlı yöntem:
Dağılım analizi ya da standart sapma hesabı gibi istatistik yöntemlerle istisna olabilecek noktalar tespit edilir, fakat çok büyük veri yığınlarında yoğun hesaplama gücü gerektirdikleri için performansları sınırlıdır.

b) Yoğunluk tabanlı yöntem:
Bu yöntemde her noktanın çevresindeki komşuları ile olan yakınlığı hesaplanır. Yakınlık hesaplamada genelde Öklit uzaklığı kullanılsa da veri türüne göre yakınlık hesaplama yöntemi farklılık gösterebilir. Bu yöntemin temel prensibi "yeterince komşusu olmayan noktaları" tespit etmektir. Bu durum Şekil 2.4'te görülmektedir.



Şekil 2.4 İstisna ve küme oluşumları

İstisna analizi aynı zamanda bir kümeleme metodudur. Bölüm 3'de kümeleme metodu olarak istisna analizi detaylarıyla açıklanmıştır.

2.6.6. Evrimsel analiz

Evrimsel analiz, zamanla davranışları değişen nesnelerin düzenlilik (regularity) ya da eğilimlerini (trends) ortaya çıkarmayı amaçlar [23]. Evrimsel analiz tanımlama, ayrımlama, birliktelik analizi, sınıflama ve kümeleme metodlarını içerse de asıl amacı verinin zaman ile olan ilişkisini ortaya çıkarmaktır. Bunun için zaman serileri (time series), ardışıklık ve periyodiklik örüntüsü bulma, benzerlik analizi gibi metodları kullanır.

Evrimsel analiz J. Han tarafından veri madenciliği kategorileri içine dahil edilse de birçok kaynakta bağımsız bir kategori olarak yer almaktadır. Evrimsel analizin kullandığı her bir yöntem evrimsel analiz adı altında değil, kendi başına bağımsız bir teknik olarak kabul görmektedir.

3. KÜMELEME ANALİZİ

3.1. Kümeleme Analizi Tanımı

Kümeleme analizi, bir veri kümesindeki bilgileri belirli yakınlık kriterlerine göre gruplara ayırma işlemidir. Bu grupların her birine "küme" adı verilir. Kümeleme analizine kısaca "kümeleme" adı verilir. Kümeleme işleminde küme içindeki elemanların benzerliği fazla, kümeler arası benzerlik ise az olmalıdır.

Kümeleme, gözetimsiz sınıflama (unsupervised classification) yöntemidir [24]. Gözetimli sınıflandırma işleminde veriler önceden sınıflandırılmış örüntülerdir. Burada temel amaç, yeni gelecek ve henüz hangi sınıfta olduğu bilinmeyen verilerin var olan sınıflardan en uygun olanına yerleştirilmesidir. Gözetimsiz sınıflamada ise amaç, başlangıçta verilen ve henüz sınıflandırılmamış bir küme veriyi anlamlı alt kümeler oluşturacak şekilde öbeklemektir. Kümeleme işlemi tamamen gelen verinin özelliklerine göre yapılır.

Kümeleme analizi istatistik, biyoloji, uzaysal veri madenciliği ve makina öğrenmesi, örüntü tanıma ve resim tanıma alanlarında kullanılmaktadır. İstatistik dünyasında k-means ve k-medoids kümeleme yöntemlerini kullanan S-Plus, SPSS ve SAS gibi paket programlar yoğunlukla kullanılmaktadır [24]. Biyolojide genetik yapıların sınıflandırılması ve yeni yapıların keşfinde, uzaysal/düzlemsel (spatial) veri madenciliğinde coğrafi konuma göre yerleşim yerlerine götürülecek mal ve hizmetler için ideal yerler belirlemede, yapay zeka alanında makina öğrenmesi için gözetimsiz öğrenme (unsupervised learning) metodu olarak kullanılmaktadır.

3.2. Kümeleme Analizinin Özellikleri

İyi bir kümeleme analizi yöntemi şu özelliklere sahip olmalıdır [23]:

- Ölçeklenebilir olmalıdır. Birkaç yüz kayıttan oluşan veri kümesine de milyonlarca kayıt içeren kümeye de uygulanabilmelidir.

- Farklı veri türleri ile kullanılabilmelidir. Hem sayısal hem kategorik veriler içeren veritabanlarında kullanılabilmelidir.
- Düzgün şekilli olmayan kümeleri de bulabilmelidir.
- En az sayıda giriş değişkeni gerektirmelidir. Bir yöntem ne kadar az giriş değişkeni gerektiriyorsa o ölçüde kullanıcının kararlarından bağımsızdır.
- Gürültü içeren veriler ile de kullanılabilmelidir.
- Veri kümesindeki kayıtların sıralanmasından bağımsız olmalıdır. Kümenin hangi elemanından başlanırsa başlansın sonuç değişmemelidir.
- Çok boyutlu veritabanlarına uygulanabilmelidir.
- Veri kümesinin sahip olduğu sınırlıkları dikkate alabilmelidir.
- Kolay yorumlanabilir sonuçlar üretebilmeli ve işlevsel olmalıdır.

Bu özellikler ideal bir kümeleme algoritmasının nitelikleridir. Mevcut algoritmaların hiç biri bu özelliklerin tamamına sahip değildir. Kümeleme analizi gelişmekte olan bir araştırma konusudur ve ilerleyen yıllarda ideale yakın yöntemlerin geliştirileceği umulmaktadır.

3.3. Kümeleme Analizi Veri Türleri

Veri madenciliğinin birçok alanında olduğu gibi Kümeleme Analizinde de veri yapısı matris formundadır. Matris formu bilgisayar ortamında hesaplama yapabilmek için en uygun veri yapısı olarak kendini kanıtlamıştır.
Kümeleme işleminde kullanılan matrisler iki temel gruba ayrılır [23]:

a) Veri Matrisi (data matrix):
Bu matris n adet nesne için p adet özelliğin tanımlandığı satırların birleşmesinden oluşan (n x p) boyutundadır. Örneğin bir şehirdeki insanların yaş, boy, ağırlık, cinsiyet, mahalle gibi özellikleri alt alta yazıldığında Denklem 3.1'deki gibi bir matris oluşur. Burada her bir sütun bir niteliği, her bir satır ise niteliklerin değerlerini içermektedir.

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix} \tag{3.1}$$

b) Farklılık matrisi (Dissimilarity matrix):
Nesnelerin diğer nesneler ile olan uzaklık bilgilerinin tutulduğu n x n boyutunda olan matristir. Bu matrisin genel ifadesi Denklem 3.2'de görülmektedir.

$$\begin{bmatrix} 0 & d(1,2) & d(1,3) & & d(1,n) \\ d(2,1) & 0 & d(2,3) & & d(2,n) \\ d(3,1) & d(3,2) & 0 & & \\ : & : & : & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix} \tag{3.2}$$

Nesneler arasındaki uzaklık fonksiyonu değişme özelliğine sahip olduğu için, diğer bir ifade ile :

$$d(i,j) = d(j,i) \tag{3.3}$$

olduğu için farklılık matrisinin asal köşegenin altında kalan değerler ile üstünde kalan değerler simetriktir. Bu yüzden farklılık matrisine tek yönlü (one-mode) matris denir ve yalnızca asal köşegen ve altında kalan elemanları içerir. Veri matrisinin böyle bir özelliği bulunmadığı için iki yönlü (two mode) matris denir.

Veri madenciliğinde çoğunlukla farklılık matrisi kullanılır. Farklılık matrisi elemanlarını bulabilmek için elemanlar arası farklar hesaplanabilmelidir. İlerleyen sayfalarda kümeleme işleminde kullanılan veri türleri arasında fark hesaplama teknikleri açıklanmıştır.

3.3.1. Aralık ölçekli değişkenler

Aralık ölçekli değişkenler (Interval Scaled Variables) doğrusal bir ölçek üzerinde temsil edilebilen değişkenlerdir. En sık kullanılan ağırlık ölçekli değişkenler boy, ağırlık, genişlik, uzunluk ve hava sıcaklığı verileridir. Aralık ölçekli değişkenler ile işlem yapılırken dikkat edilmesi gereken en önemli nokta öncelikle verilerin standartlaştırılmasıdır. Bir niteliği tanımlayan tüm ölçüm değerleri aynı tür ölçüm birimi ile temsil edilmelidir. Örneğin uzunluk ölçüm verileri üzerinde işlem yapılıyorsa verilerin bir bölümünün milimetre, diğer bölümünün santimetre, desimetre, metre gibi farklı ölçeklerde olması kümeleme işleminin başarısız olmasına yol açacaktır.

Aralık ölçekli veriler için uzaklık ya da komşuluk mesafesi hesaplamada üç çeşit uzaklık formülü kullanılır [23] :

a) Öklit uzaklığı (Euclidian Distance):

En sık kullanılan yöntemdir. İki ya da daha çok boyutlu düzlemde kolaylıkla kullanılabilir. Boyut sayısı arttıkça hesaplama süresi de artmaktadır.

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \ldots + |x_{i_p} - x_{j_p}|^2)} \qquad (3.4)$$

Formülde i = (x_{i1}, x_{i2}, …, x_{ip}) ve j = (x_{j1}, x_{j2}, …, x_{jp}) ifadeleri p boyutlu veri nesnelerini temsil etmektedir.

b) Manhattan uzaklığı (Manhattan Distance):
p boyutlu uzayda herhangi iki noktanın karşılıklı her bir koordinat değerinin farkı alınarak bulunur.

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{ip} - x_{jp}| \qquad (3.5)$$

c) Minkowski uzaklığı (Minkowski Distance):

Öklit ve Manhattan uzaklığının genelleştirilmiş hali olarak ifade edilebilir.

$$d(i,j) = (|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + ... + |x_{ip} - x_{jp}|^q)^{1/q} \quad (3.6)$$

q bir pozitif tam sayı olmak üzere q=1 için bu ifade Manhattan uzaklığını, q=2 için ise Öklit uzaklığını belirtir. q değişkeninin değeri arttırıldıkça daha hassas uzaklık ölçüm ifadeleri elde edilir.

3.3.2. İkili değişkenler

İkili değişkenler (Binary Variables) yalnızca "var" ya da "yok", diğer bir deyişle 0 ya da 1 değerini alabilen değişkenlerdir. Bu tür değişkenler tanımladıkları niteliğin ne kadar olduğunu değil, olup olmadığını belirtirler. Örneğin bir hastanede hasta kayıtları veritabanında hastanın sigara kullanımı ile ilgili bilgiler ikili değişken sınıfına girer, çünkü sigara kullanımı ifadesi yalnızca evet ya da hayır değeri alabilmektedir.

İkili değişkenler içeren kayıtlar arasında uzaklık hesabı için Şekil 3.1' deki gibi bir tablo geliştirilmiştir.

		Nesne j		
		1	0	*toplam*
Nesne i	1	a	b	$a+b$
	0	c	d	$c+d$
	toplam	$a+c$	$b+d$	p

Şekil 3.5 İkili değişkenler arası uzaklık hesabı tablosu [23].

Bu tabloyu oluşturmak için ikili değişkenler içeren i ve j nesneleri seçilir. Her iki nesnede de aynı anda 1 değerini almış olan özelliklerin sayısı a, nesne i'de 1 ve nesne j'de 0 değerini almış olan değişkenlerin sayısı b şeklinde devam ederek a,b,c,d sayıları bulunur. Bu sayılar kullanılarak aşağıda verilen formül ile i ve j nesneleri arası uzaklık hesaplanır.

$$d(i,j) = \frac{b+c}{a+b+c+d} \tag{3.7}$$

3.3.3. Nominal, ordinal ve oran değişkenleri

3.3.3.1. Nominal değişkenler

Nominal değişkenler ikili değişkenlerin genelleştirilmiş şekli olarak ifade edilebilir. Örneğin otomobil satışı yapan bir firma müşterilerine sarı, mavi, kırmızı, siyah seçeneklerini sunduğu düşünülürse otomobil rengi değişkeni Nominal değişkenler sınıfına girer. İkili değişkenler yalnızca iki farklı değer alabilmelerine karşın Nominal değişkenler ikiden fazla, fakat sonlu sayıda değer alabilen değişkenlerdir.
Bu tür değişkenler arasında uzaklık hesabı için;

$$d(i,j) = \frac{p-m}{p} \tag{3.8}$$

formülü kullanılır. Formülde m değişkeni i ve j değişkenlerinde aynı anda yani değeri almış olan özellik sayısı, p değişkeni ise i ve j nesnelerinin sahip olduğu toplam özellik sayısını belirtir.

3.3.3.2. Ordinal değişkenler

Bu değişkenler de Nominal değişkenlerde olduğu gibi sonlu sayıda farklı durum içerirler fakat Ordinal değişkenler anlamlı bir sıralama takip ederler. Sıralamada daha üstte olan değişken bir alttakinden daha değerlidir. Örneğin, yarışmalarda elde edilen madalyalar Ordinal değişken türüne girer, çünkü en çok başarı gösteren yarışmacı

altın, onu takip eden gümüş ve madalya almaya layık en son kişi ise bronz madalya alır.

Ordinal değişkenler arası uzaklık tespiti için, farklı yöntemler geliştirilmiş olsa da en kolay kullanılabilecek yöntem, ordinal değişkenin alabileceği değerleri [0-1] aralığında sayı değerler alabilecek şekilde standartlaştırıp aralık ölçekli değişkenlerde kullanılan mesafe yöntemlerini kullanmaktır [23].

3.3.3.3. Oran ölçekli değişkenler

Oran ölçekli değişkenler doğrusal olmayan ölçek üzerinde yapılan ölçümlerin sonuçlarıdır. En bilinen oran ölçekli değişkenler bakteri popülasyonlarının büyüme grafiği ve bir radyoaktif elementin yarı ömrünün ölçüm sonuçlarıdır. Oran ölçekli değişkenlerin genel yapısı aşağıdaki gibidir:

$$Ae^{Bt} \text{ ya da } Ae^{-Bt} \tag{3.9}$$

Denklem 3.9'daki ifadelerde A ve B pozitif sabitlerdir.
Oran ölçekli değişkenlerde uzaklık hesaplama için üç farklı görüş ortaya atılmaktadır.

1) Bu tür değişkenleri aralık ölçekli değişken gibi düşünüp işlem yapılabilir. Bu yöntem büyük hata paylarına neden olmaktadır, çünkü ölçülen aralık doğrusal değildir.
2) Oran ölçekli değişkenlere Logaritmik dönüşüm uygulanabilir.
   $y_{if} = \log(x_{if})$ (3.10)
   Bu durumda y_{if} değeri doğrusal ölçekli hale geldiği için aralık ölçekli değişken olarak işlem yapılabilir.
3) Oran ölçekli değişkenler sürekli ordinal değişken olarak düşünülüp Ordinal değişkenlerde uzaklık hesaplama yöntemleri kullanılabilir[23].

3.3.4. Karışık tür değişkenler

Karışık tür değişkenler şimdiye kadar açıklanan veri türlerinden iki ya da daha fazlasını içeren değişkenlerdir. Karışık tür değişkenlerin hesaplanmasında iki temel yaklaşım bulunmaktadır.

1) Tüm değişkenleri türlerine göre gruplandırıp, her gruba kendi içinde işlem yapılabilir. Bu yaklaşım karmaşık olduğu kadar yoğun işlem gücü de gerektirdiği için tercih edilmemektedir.
2) Bütün veri türleri için genel bir uzaklık hesaplama yöntemi kullanmaktır. J.Han bu tür bir formül sunmuştur [23]. Bu formül tez konusu ile doğrudan ilgili olmadığı için burada açıklanmayacaktır.

3.4. Kümeleme Metodları

Veri madenciliğinde birçok kümeleme metodu bulunmaktadır. Kümeleme metodu seçimi kullanılacak veri türüne ve uygulamanın amacına göre farklılık gösterir. Kümeleme metodları arasında benzerlik fazladır. Bu nedenle bilimsel literatürde en çok kabul gören metodlar bu bölümde açıklananmıştır.

3.4.1. Bölümleme metodları

Bölümleme metodları (partitioning methods), n adet nesneden oluşan veritabanını, giriş parametresi olarak belirlenen k adet bölüme ($k <= n$) ayırma temeline dayanır. Veritabanındaki her bir eleman bir farklılık fonksiyonuna (dissimilarity function) göre k adet bölümden birine dahil edilir. Bu bölümlerden her biri bir küme olarak adlandırılır.

Bölümleme metodları k sayısı doğru tahmin edilebilirse benzer şekilli dışbükey kümeleri bulmakta oldukça başarılı sonuçlar vermektedir. Eğer k sayısı hakkında önceden bir fikir belirlenemezse algoritmayı farklı k değerleri için tekrar tekrar uygulayarak en uygun k değeri bulunabilir.

Bölümleme metodlarının genel problemi k giriş parametresine bağımlı olmaları ve düzgün şekilli olmayan kümeleri bulamamalarıdır [24]. Bölümleme metodları k-means, k-medoids ve CLARA-CLARANS olarak bilinen algoritmaları kullanır [23].

Bu tez çalışmasının uygulama kısmında kullanılan k-means algoritması, J. MacQueen [10] tarafından 1967 yılında tanıtılmıştır. Çalışma yönteminde, Öklit uzaklığı temel alınarak kümeleme yapılır. Bu yöntem yıllardır bilimsel ve endüstriyel uygulamalarda en yoğun kullanılan kümeleme algoritması haline gelmiştir. Verilen nesneleri nitelik veya özelliklerine göre k adet sınıfa ayırmak amacıyla kullanılır. Sınıflandırma, verilerin en yakın veya benzer oldukları küme merkezleri (centroid) etrafına yerleştirilmesi ile gerçekleştirilir. K-means algoritması dördüncü bölümde ayrıntılarıyla açıklanmıştır.

3.4.1.1. K-medoids algoritması

K-medoids algoritması k-means algoritmasının gürültü ve istisna verilere aşırı duyarlılığını gidermek amacıyla Kaufman ve Rousseeuw tarafından 1987 yılında geliştirilmiştir [25].

K-medoids algoritması kümeyi temsil edecek noktayı bulmak için küme elemanlarının ortalamasını almak yerine kümenin en merkez noktasındaki elemanı yeni küme merkezi olarak alır. Böylece istisna verilerin küme merkezini kenarlara doğru kaydırması problemi giderilmiş olur.

K-medoids algoritmasının birçok farklı türevi bulunmaktadır. PAM (Partitioning Around Medoids) ilk ortaya atılan K-medoids algoritmasıdır. PAM, öncelikle k-means algoritmasında olduğu gibi rastgele seçtiği k adet sayıyı küme merkezi olarak alır. Kümeye her yeni eleman katıldığında kümenin elemanlarını deneyerek kümenin gelişmesine en fazla katkıda bulunabilecek noktayı tespit edince bulduğu noktayı yeni merkez, eski merkezi ise sıradan küme elemanı olacak şekilde yer değiştirme (swap) işlemi yapar.

PAM küçük veritabanlarında çok iyi sonuçlar vermesine rağmen hesaplanabilir karmaşıklığı yüksek olduğu için çok eleman içeren veritabanlarında zayıf performans gösterir. Büyük veritabanları için CLARA ve CLARANS algoritmaları geliştirilmiştir. PAM algoritmasının karmaşıklığı $O(k(n-k)^2)$ dir.

3.4.1.2. CLARA ve CLARANS algoritmaları

PAM, K-medoids algoritmalarının başarısını kanıtlamasına rağmen büyük veritabanlarında başarılı olamayınca Kaufman ve Rousseeuw tarafından CLARA (Clustering LARge Applications) 1990 yılında ortaya atılmıştır [25].

CLARA, veritabanının tümünü almak yerine küçük bir örneklem (sample) kümesini temsilcisi olarak alıp örneklem üzerinde PAM algoritmasını uygular. Veritabanında birden çok örneklem seçerek en iyi sonuç veren örneklemden elde ettiği PAM sonucunu çıktı olarak verir.

CLARA'nın avantajı PAM'dan daha büyük veri yığınlarına uygulanabilmesi, dezavantajı ise performansının örneklemin boyutuna göre değişmesi ve örneklem seçimi yeterince bağımsız değilse seçilen örneklem veritabanını yeterince temsil edemeyeceği için yanlış sonuçlara ulaşılmasıdır. CLARA algoritmasının karmaşıklığı s örneklem boyutu olmak üzere $O(ks^2 + k(n-k))$ dir.

CLARANS (CLustering Algorithm based on RANdomized Search) algoritması CLARA'nın sonuçlarını örneklem seçimine bağlı olmaktan kurtarmak amacı ile Ng ve J.Han tarafından 1994 yılında VLDB'94 konferansında bilim dünyasına sunulmuştur.

CLARANS örneklem seçimindeki önyargıyı gidermek için sabit bir örneklem yerine her aşamada değişen örneklem kavramını ortaya atmıştır. Rastgele seçilen noktalar çevresi dikkate alınarak bir örneklem oluşturulur. Bu örneklemde k-medoids algoritması ile bulunan merkez noktalar temsil noktası olarak alınır. Bu işlemden sonra, başka rastgele noktalar bulunur ve bu noktalar çevresindeki noktalara yine k-medoids uygulanır ve yeni merkez noktalar bulunarak bu şekilde tüm veritabanının

örneklemi tarafsızca oluşturulur. Böylece örneklem seçimi önyargıdan bağımsız hale gelir.

CLARANS algoritması CLARA 'ya göre oldukça iyi sonuçlar verir. Ayrıca istisna bölgeleri bulma yeteneğine de sahiptir. Fakat hesaplanabilir karmaşıklığı n nesne sayısı olmak üzere $O(n^2)$ olduğu için veri sayısı arttıkça gerektirdiği hesaplama gücü üstel olarak artar.Ayrıca veritabanı ana bellekte değilse giriş/çıkış (I/O) işlemi oldukça fazla artacağı için genel performansı düşer.

3.4.2. Hiyerarşik metodlar

Hiyerarşik metodlar nesneleri Dendrogram adı verilen ağaç yapısı şeklinde gruplandırma temeline dayanır. Bu yapının inşa edilme yönüne göre Hiyerarşik metodlar iki bölümde incelenir [23]:

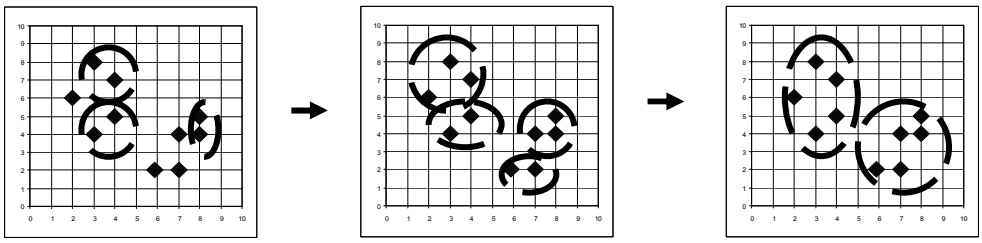
- Birleştirici (agglomerative) kümeleme
- Ayrıştırıcı (divisive) kümeleme

Hiyerarşik metodlar giriş parametresi olarak bulunacak küme sayısını belirten *k* değerine ihtiyaç duymazlar, fakat ağaç yapısı oluşturma işlemini ne zaman durdurulacağını belirten eşik değeri parametresine ihtiyaç duyarlar.

3.4.2.1. Birleştirici ve ayrıştırıcı algoritmalar

a) Birleştirici kümeleme AGNES (AGglomerative NESting)

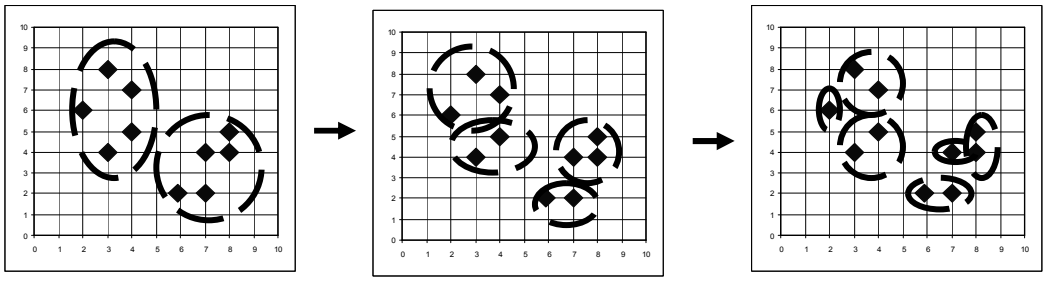
Kaufman ve Rousseeuw tarafından 1990 yılında sunulmuştur [25]. Aşağıdan yukarı inşa yapısı izler. Başlangıçta her bir nesneyi bağımsız bir küme olarak görür ve her adımda bu kümelerden benzer özellik gösterenleri birleştirir. Birleştirme işlemi bir sonlandırma koşulu sağlanana kadar sürer. Herhangi bir sonlandırma koşulu verilmezse bütün nesneler tek bir küme olur. Bu durum Şekil 3.2'de görülmektedir.



Şekil 3.6 Birleştirici Hiyerarşik Algoritmalar, AGNES. [23]

b) Ayrıştırıcı kümeleme DIANA(DIvisive ANAlysis)

DIANA algoritması da Kaufman ve Rousseeuw tarafından 1990 yılında sunulmuştur [25]. AGNES algoritmasından farkı, yukarıdan aşağı inşa yapısı kullanmasıdır.Bu yüzden, başlangıçta veri nesnelerinin tümünü tek bir küme olarak görür ve her adımda kendi içinde benzerlik oranı yüksek olan nesneleri bir araya getirerek büyük kümeyi önce ikiye sonra bölünenleri tekrar ikiye bölerek sonlandırma koşulu sağlanana kadar bölme işlemini sürdürür. Bir sonlandırma koşulu sağlanmaz ise her bir nesne ayrı bir küme olana kadar işlem sürer. İşlemin ilerleyişi Şekil 3.3'te görülmektedir.



Şekil 3.7 Ayrıştırıcı Hiyerarşik Algoritmalar, DIANA. [23]

Birleştirici ve ayrıştırıcı Hiyerarşik algoritmalar S-Plus Ticari İstatistik ve Veri Madenciliği yazılımında kullanılmaktadır.

3.4.2.2. BIRCH

BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies) algoritması SIGMOD 96 konferansında Zhang, Ramakrishnan ve Livny tarafından sunulmuştur. BIRCH iki önemli kavram üzerine kurulmuştur: Küme niteleyici (clustering feature) ve Küme niteleyici ağacı (clustering feature tree). Küme niteleyici küçük gruplar

halindeki veri nesnelerinden oluşturulan alt kümeleri ana bellekte temsil edecek olan üç adet parametreden oluşan yapıdır.

$$CF = (N, \overrightarrow{LS}, SS) \tag{3.11}$$

Formülde N altkümedeki nesne sayısı,

$\overrightarrow{LS}$ altkümedeki N noktanın doğrusal toplamı(linear sum),

$$LS: \sum_{i=1}^{N} = \overrightarrow{X_i} \tag{3.12}$$

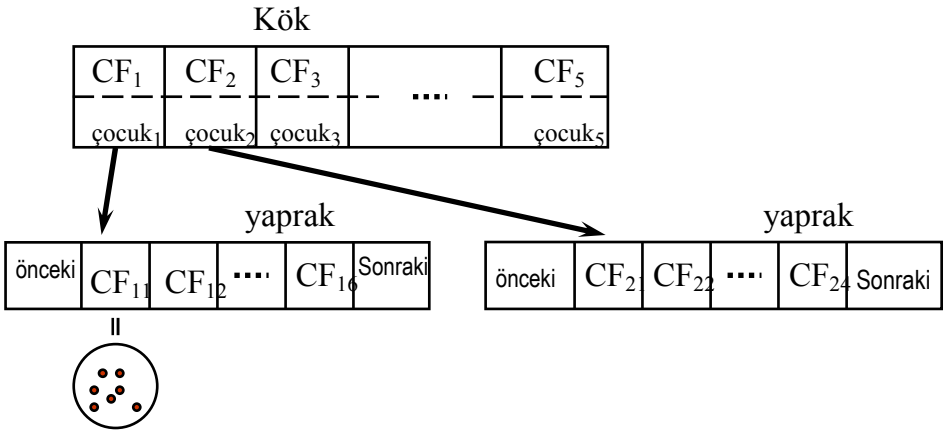
SS ise altkümedeki N noktanın karelerinin doğrusal toplamı,

$$LS: \sum_{i=1}^{N} = \overrightarrow{X_i}^2 \tag{3.13}$$

BIRCH algoritmasın çalışması iki aşamada gerçekleşir:

1) Birinci aşamada bütün veritabanı taranarak birleştirici hiyerarşik algoritmalardaki gibi önceden belirlenen N sayısı kadar veri içeren küçük alt kümeler oluşturulur. Bu alt kümelerin her biri için CF değeri hesaplanır. Bu CF değerleri veritabanına oranla çok daha az yer kapladığı için ana bellekte tutulur. Bulunan CF değerleri kullanılarak CF ağaçları oluşturulur. CF ağaçları için iki parametre söz konusudur: Dallanma katsayısı (branching factor) ağaçta en fazla kaç adet yaprak olacağını belirtir, eşik (treshold) değeri ise yapraklarda oluşturulacak küme sayısının çapının en fazla ne kadar olacağını belirler. CF ağacı yapısı Şekil 3.4'te görülmektedir.

2) Ana bellekte oluşturulan CF ağacı gerçek veritabanının küme yapısını yansıttığı için CF nesneleri üzerinde herhangi bir bölümleme ya da hiyerarşik algoritma kullanılarak kümeleme işlemi kolaylıkla gerçekleştirilebilir.



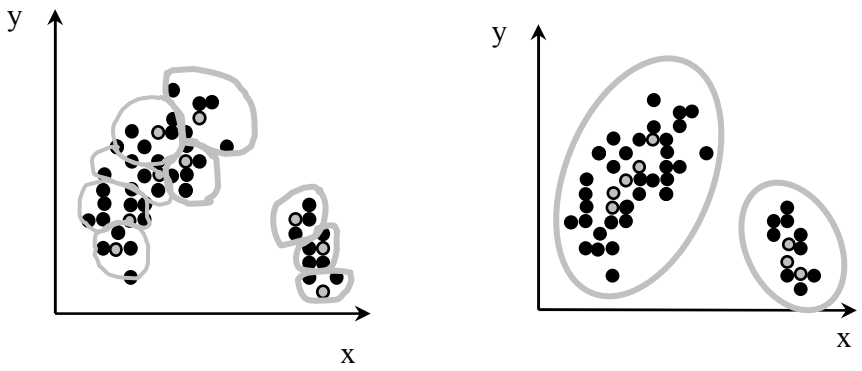
Şekil 3.8 BIRCH algoritması için oluşturulan CF ağacı.[23]

BIRCH tek başına bir kümeleme algoritması değil, verinin ana belleğe sığmayacak kadar büyük boyutlarda olduğu durumlarda veritabanının ana belleğe sığacak bir modelini oluşturmaya imkan veren bir algoritmadır. Bu yüzden BIRCH algoritması diğer kümeleme yöntemleri için ön işlem aşaması olarak da kullanılır.

BIRCH algoritmasının avantajları: veritabanı yapısını ana belleğe sığdırdığı için I/O miktarını azaltarak performansı arttırır, tek bir tarama ile veritabanının modelini oluşturabilir, hesaplanabilir karmaşıklığı O(n) olduğu için çok fazla işlem gücü gerektirmez. Dezavantajları: yalnızca sayısal verilerde kullanılabilir, verilerin okunma sırasına duyarlıdır, tüm hiyerarşik algoritmalarda olduğu gibi sadece dairesel kümeleri bulabilir [24].

3.4.2.3. CURE (Clustering Using REpresentatives)

Hiyerarşik metodların dairesel olmayan kümeleri bulma konusundaki zayıflıklarını gidermek üzere Guha, Rastogi ve Shim tarafından SIGMOD 1998 konferansında CURE algoritması sunulmuştur. CURE algoritması BIRCH algoritmasında olduğu gibi ölçeklenebilirliği arttırmak için, rastgele örneklem alarak veritabanını modelleme ilkesine dayanır.



(a) (b)

Şekil 3.9 CURE algoritmasının çalışma şekli [23].

CURE algoritmasının çalışma şekli altı adımda açıklanabilir:

a) Veritabanı nesnelerinden rastgele noktalar seçilerek bir örneklem kümesi oluşturulur.
b) Veritabanı örneklem kümesinin elemanlarının sayısı kadar bölüme ayrılır. Şekil 3.5 (a)'da görülen açık gri noktalar örneklem kümesinin elemanlarını temsil etmektedir.
c) Veritabanındaki her bir bölüm üzerinde kümeleme işlemi yapılır ve örneklem kümesinin her elemanı bir kümenin merkezi olacak şekilde alt kümeler oluşturulur. Şekil 3.5(a)'da kapalı eğriler kümeleri temsil etmektedir.
d) Kümeleme işlemi sonunda, yeterli büyüklüğe erişmemiş altkümeler istisna olarak adlandırılır ve devre dışı bırakılır.
e) Alt kümelerin oluşturacağı büyük kümeyi bulmak için, alt kümelerin yalnızca merkez noktaları dikkate alınarak kümeleme işlemi uygulanır. İşlem sonunda daha büyük ve küresel olmayan kümeler elde edilir. Bu durum Şekil 3.4 (b)'de görülmektedir.
f) Kümeleme işlemi yeterli olmazsa, yeni bulunan büyük kümelerin merkezleri yeni örneklem noktaları olacak şekilde CURE algoritması tekrar uygulanarak daha büyük kümeler elde edilir.

CURE algoritmasının dezavantajı, kategorik veriler içeren veritabanlarında kullanılamamasıdır. CURE algoritmasının kategorik veriler için tasarlanmış şekli ROCK (Robust Clustering Algorithm) olarak adlandırılır. CURE ile benzer mantığa sahip olduğu için ayrıntılı olarak değinilmeyecektir [23].

3.4.2.4. CHAMELEON

CHAMELEON (Hierarchical Clustering Using Dynamic Modeling) Karypis, Han ve Kumar tarafından 1999 yılında sunulmuştur [23]. CURE ve ROCK algoritmalarının geliştirilmiş bir modeli olarak ortaya çıkmıştır. CHAMELEON, dinamik modelleme yapısını kullanır. Küme oluşturma işlemi sırasında komşu birleştirme işlemleri değişmez değildir, nesneler sürekli olarak kümeler arasında değiştirilerek en uygun olduğu kümeye dahil edilmeye çalışılır.

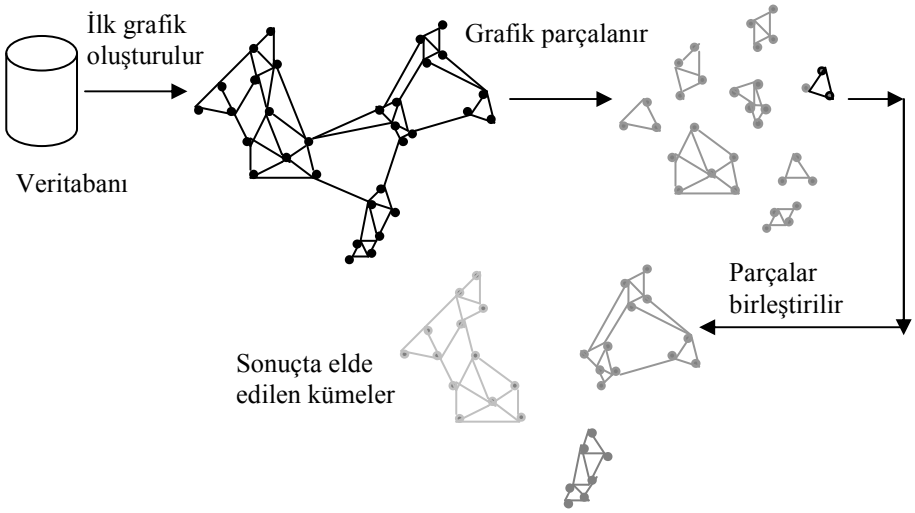
CHAMELEON algoritmasının çalışma yapısı Şekil 3.6 da görülmektedir. Veritabanından alınan verilerle öncelikle k- en yakın komşu (k-nearest neighbour) algoritması kullanılarak ilk kümeleme işlemi yapılır. Bu işleme seyrek grafik (sparse graph) oluşturma adı verilir. K-en yakın komşu algoritması kümeleri belirlemede öklit mesafesini kullanır. Elde edilen kümeleme yapısı tekrar bölünerek ufak alt kümelere ayrılır. Alt kümelerin yeni kümeler oluşturacak şekilde kümelenmesi iki kriter dikkate alınarak yapılır:

a) Bağıl Bağlanabilirlik (Relative Interconnectivity): İki kümenin birbirine ne kadar benzer olduğunun ölçüsüdür. Matematiksel yöntemlerle hesaplanır.

b) Bağıl Yakınlık (Relative Closeness): İki kümenin birbirine olan yakınlık mesafesidir.

Bu iki parametre kullanılarak en uygun altkümeler birleştirilerek kümelenmeler bulunur. Bulunan kümelenmeler mutlak değildir, bir altküme için en uygun küme tespit edilene kadar birleşme ayrılma işlemi sürer. En uygun kümeler bulunduğunda dinamik birleşme ve ayrılma işlemleri sona erer.

CURE algoritmasının CURE ya da DBSCAN algoritmasından daha iyi sonuçlar verdiği ispatlanmıştır, fakat hesaplanabilir karmaşıklığı $O(n^2)$ olduğu için yüksek hesaplama gücü gerektirir.



Şekil 3.10 CHAMELEON algoritması çalışma yapısı [23].

3.4.3. Yoğunluk tabanlı metodlar

Yoğunluk tabanlı metodlar, nesnelerin doğal dağılımını bir yoğunluk fonksiyonu aracılığı ile tespit ederek bir eşik yoğunluğunu aşan bölgeleri küme olarak adlandırırlar. Yoğunluk tabanlı algoritmalar düzgün şekilli olmayan kümeleri bulma başarısı, gürültü ve istisnalardan etkilenmeme ve tek tarama ile sonuca ulaşma avantajları ile en başarılı kümeleme metodları arasındadır.

3.4.3.1. DBSCAN (Density Based Spatial Clustering of Applications with Noise)

Ester, Kriegel, Sander ve Xu tarafından KDD'96 konferansında sunulmuştur [26]. Nesnelerin komşuları ile olan mesafelerini hesaplayarak belirli bir bölgede önceden belirlenmiş eşik değerden daha fazla nesne bulunan alanları gruplandırarak kümeleme işlemini gerçekleştirir. DBSCAN algoritması veri madenciliğine birçok yeni terim ve yaklaşım getirmiştir.

DBSCAN algoritması için önemli tanımlar:

a) Çekirdek Nesne(core object):

Bir veri nesnesi ε-komşuluğunda önceden belirlenen bir eşik değerden(MinPts) daha çok nokta içeriyorsa bu nesne çekirdek nesnedir.

b) Eps:

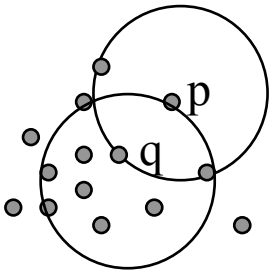
Bir veri nesnesinin komşularını belirlemek için gerekli olan yakınlık mesafesidir.

c) MinPts:

Bir bölgenin yoğun olarak adlandırılabilmesi için Eps komşuluğunda bulunması gereken en az komşu sayısıdır.

d) Doğrudan Yoğunluk Erişilebilir Nokta (Direct Density Reachable point):

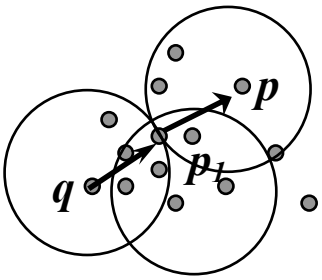
p noktası q nun Eps komşuluğunda ise ve q noktası p ye göre çekirdek nesne ise p noktası q'ya göre doğrudan yoğunluk erişilebilir noktadır. Şekil 3.7'de p ile q noktaları doğrudan yoğunluk erişilebilir noktalardır.



Şekil 3.11 Doğrudan yoğunluk erişilebilir noktalar [26].

e) Yoğunluk Erişilebilir Nokta (Density Reachable point):

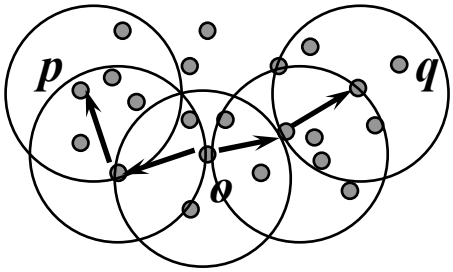
p_{i+1} noktası p ye göre doğrudan yoğunluk erişilebilir ise $p_1,p_2,p_3,\ldots p_n$ doğrudan yoğunluk erişilebilir noktalar olmak üzere $p_1=p$ ve $p_n=q$ ise q noktası Eps ve MinPts değerlerine göre p noktasına yoğunluk erişilebilirdir. Şekil 3.8'de p ile q noktaları yoğunluk erişilebilir noktalardır.



Şekil 3.12 Yoğunluk erişilebilir noktalar [26].

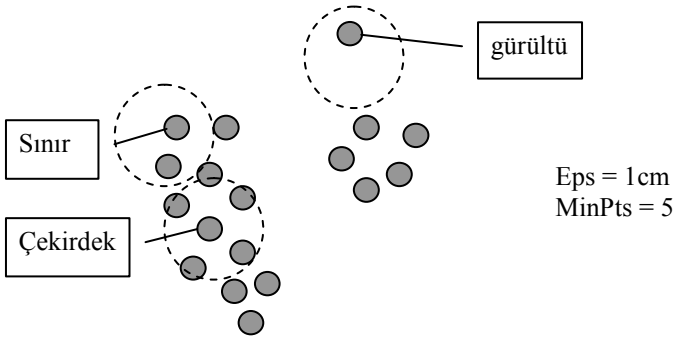
f) Yoğunluk bağlı noktalar (Density connected) :

Herhangi p ve q noktalarının her ikisi de bir o noktasına yoğunluk erişilebilir durumda ise, p ve q noktaları Eps ve MinPts değerine göre yoğunluk bağlı noktalardır. Bu durum Şekil 3.9'da görülmektedir.



Şekil 3.13 Yoğunluk bağlı noktalar [26].

DBSCAN algoritmasının çalışması için MinPts ve Eps parametreleri bildirilmelidir. Algoritma öncelikle rastgele bir p noktası seçer. p noktasına MinPts ve Eps değerlerine göre yoğunluk erişilebilir olan tüm noktaları bulur, eğer p çekirdek nokta koşulunu sağlıyor ise yeni bir küme keşfedilmiş olur. p noktasına yoğunluk erişilebilir olan tüm noktalara teker teker alarak aynı işlem uygulanır, eğer herhangi bir nokta çekirdek nokta koşulunu sağlamıyorsa bu nokta kümenin sınır noktasıdır. İncelenen tüm noktalardan hiçbiri çekirdek nokta koşulunu sağlamadığı zaman kümenin sınırları belirlenmiş olur. Algoritma yeni bir rastgele nokta seçerek aynı işlemleri tekrar eder. Eğer rastgele seçilen nokta çekirdek nokta koşulunu sağlamıyorsa bu nokta gürültü ya da istisna olarak tanımlanır. Eps=1 ve MinPts=5 için DBSCAN algoritmasının çalışma yapısı Şekil 3.10'da görülmektedir.



Şekil 3.14 DBSCAN algoritması çalışma yapısı [23].

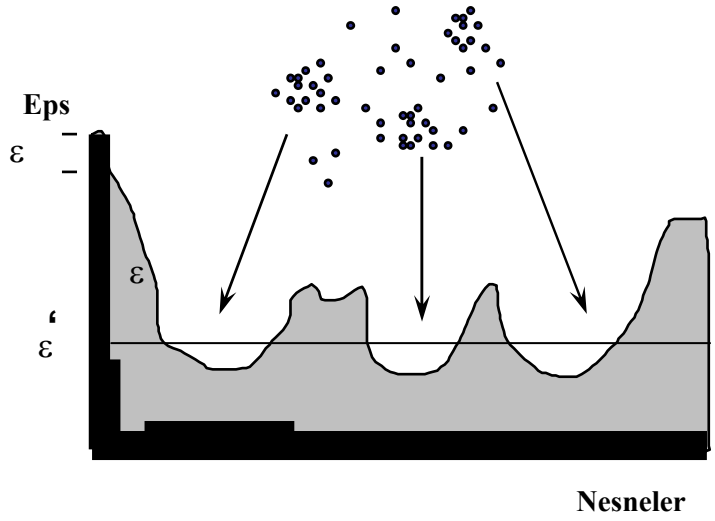
DBSCAN algoritmasının komşu tespiti işlemi için bir sıralama metodu kullanılırsa hesaplanabilir karmaşıklığı O(n log*n*), kullanılmazsa O(n^2) olmaktadır.

3.4.3.2. OPTICS (Ordering Points to Identify the Clustering Structure)

Ankerst, Breunig, Kriegel, ve Sander tarafından SIGMOD'99 konferansında sunulmuştur [27]. DBSCAN algoritmasının geliştirilmiş hali olarak tanımlanabilir. DBSCAN algoritmasının zayıflığı olarak tanımlanabilen Eps ve MinPts değerlerine bağımlığı azaltmak için veri nesnelerini Eps değerine göre bir grafik üzerine yerleştirip MinPts değerine gerek kalmadan grafik üzerinden kümeleri bulmayı sağlar.

OPTICS sadece Eps değerini giriş parametresi olarak aldığı için DBSCAN algoritmasına göre daha bağımsız sonuçlar üretebilmektedir. Ayrıca farklı Eps değerlerine sahip kümelenmeleri tespit etmek için veritabanını tekrar taramaya gerek duymaz, tek bir tarama ile elde edilen grafik tüm analiz işlemleri için kullanılabilir. Bu avantajına rağmen OPTICS kendi başına bir kümeleme algoritması değil, bir kümeleme görselleştirme aracıdır. Veri kümesini insan gözünün analiz edebileceği anlamlı şekiller haline getirir.

OPTICS algoritması öncelikle rastgele bir nokta seçer. Seçilen noktanın Eps komşuluğunda bulunan en yakın komşusu ile seçilen nokta arasındaki uzaklığı bir çubuk grafiğinde bir sütun olarak temsil eder. Aynı işlemi uzaklık sırasına göre Eps komşuluktaki tüm nesneler için gerçekleştirir. Komşu kalmayınca yeni bir rastgele nokta seçip bu noktanın komşuları için aynı işlemi uygular.
Şekil 3.11'de OPTICS tarafından oluşturulmuş grafik görülmektedir. Grafikte istenilen Eps değerinin olduğu noktadan yatay bir çizgi çizildiğinde altında kalan alandaki vadiler istenilen Eps değeri için elde edilen kümelerdir. Şekil 3.11'de yatay çizginin altına bakılırsa üç adet küme olduğu açıkça görülür.



Şekil 3.15 OPTICS algoritması çalışma yapısı [23].

OPTICS, çok boyutlu veriler için daha farklı görselleştirme teknikleri sunmaktadır. OPTICS algoritması DBSCAN ile benzer temellere dayandıkları için ürettikleri sonuçlar benzerdir ve hesaplanabilir karmaşıklıkları aynıdır.

3.4.3.3. DENCLUE (Density Based Clustering)

Hinneburg ve Keim tarafından KDD 98 konferansında sunulmuştur [28]. Kümelenmeleri belirlemek için yoğunluk dağılım fonksiyonlarından yararlanır. Bu yüzden DENCLUE algoritması sağlam matematiksel temellere dayanmaktadır. DBSCAN algoritmasına göre 45 kata varan oranda daha hızlı olduğu deneysel olarak kanıtlanmıştır. Fakat çok fazla giriş parametresi gerektirmektedir.

Denclue algoritması üç temel kavram üzerine kuruludur:

1) Her bir veri nesnesinin diğerleri üzerindeki etkisi bir fonksiyon kullanılarak matematiksel olarak modellenebilir. Bu fonksiyon etkileme fonksiyonu (influence function) olarak adlandırılır.
2) Veri uzayının genel yoğunluk fonksiyonu (overall density function), her bir veri noktasının etkileme fonksiyonları toplanarak bulunabilir.

3) Veri uzayının genel yoğunluk fonksiyonunun yerel maksimum noktaları incelenerek kümeleme merkezleri bulunabilir. Yerel maksimum noktanın bulunduğu eğrinin yamaçları ise kümelenme alanlarını gösterir.

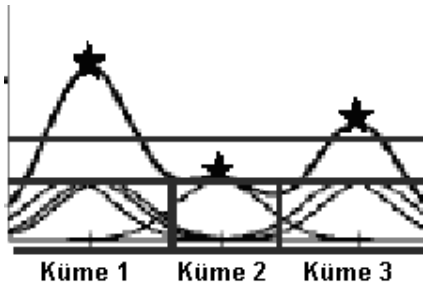
DENCLUE algoritmasında etkileme fonksiyonu olarak genelde kare dalga etkileme fonksiyonu;

$$f_{kare}(x,y) = \begin{cases} 0 & \text{eğer } d(x,y) > \sigma \\ 1 & \text{diğer durumda} \end{cases} \quad (3.14)$$

veya gauss (gaussian) etkileme fonksiyonu;

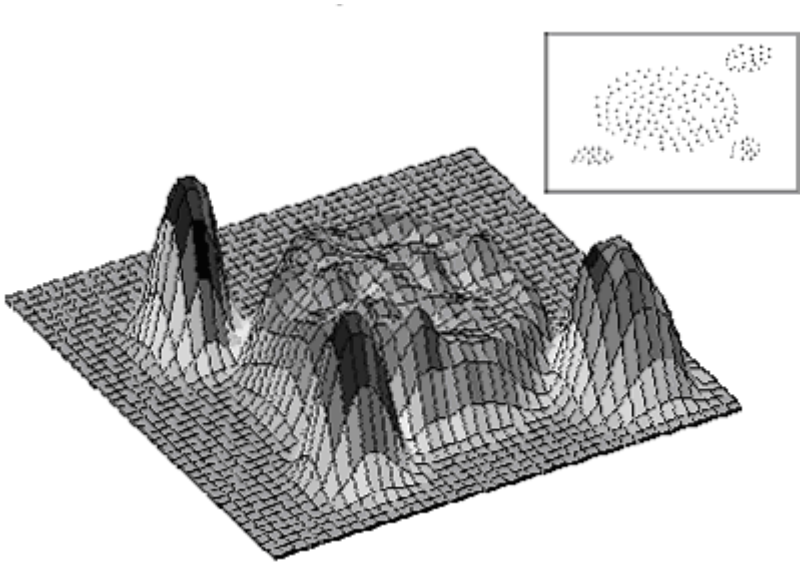
$$f_{Gaussian}(x,y) = e^{-\frac{d(x,y)^2}{2\sigma^2}} \quad (3.15)$$

kullanılmaktadır. Genel yoğunluk fonksiyonu grafiği Şekil 3.12'de görülmektedir. Grafikte (*) olan noktalar yerel maksimum noktalardır. Bu noktaların her biri veri uzayında bir kümelenme merkezidir. Bu noktaların bulunduğu eğrilerin yamaç bölümleri kümelenme bölgelerini gösterir. Kümeler Şekil 3.12 de gösterilmiştir.



Şekil 3.16 Genel yoğunluk fonksiyonu [28].

DENCLUE algoritması matematiksel fonksiyonlarla ifade edildiği için kümelenme yapılarının görselleştirilmesinde de kolaylık sağlar. Şekil 3.13 de iki boyutlu veri kümesi ve DENCLUE algoritması kullanılarak görselleştirilmiş şekli görülmektedir. Şeklin çizilmesinde Gauss etkileme fonksiyonu kullanılmıştır.



Şekil 3.17 İki boyutlu veri kümesi için Gauss etkileme fonksiyonu [28].

DENCLUE algoritması, yoğun miktarda gürültü içeren veritabanlarında dahi başarılı sonuçlar vermektedir. Ayrıca, çok boyutlu ve karmaşık veritabanlarını matematiksel modelleme olanağı sağlamaktadır. Bu avantajlarına rağmen yoğunluk parametresi ve eşik parametresi seçimine karşı çok duyarlıdır. Uygun olmayan parametre seçimi çok farklı sonuçlara sebep olabilmektedir [23].

3.4.4. Izgara tabanlı metodlar

Izgara tabanlı metodlar (Grid Based Methods), veri uzayını incelemek için sonlu sayıda kare şeklinde hücrelerden oluşan ızgara yapıları kullanırlar. Kullandıkları ızgara yapısından dolayı veritabanındaki nesne sayısından bağımsızdırlar. Performanslarını etkileyen tek unsur kullandıkları kare sayısıdır, kare sayısı arttıkça hesaplama zamanı artacağından performans düşer. Izgara tabanlı yöntemlerin en önemli avantajları işlem yükü az olduğu için hızlı ve çabuk sonuca ulaşabilmeleridir [23].

3.4.4.1. STING (Statistical Information Grid)

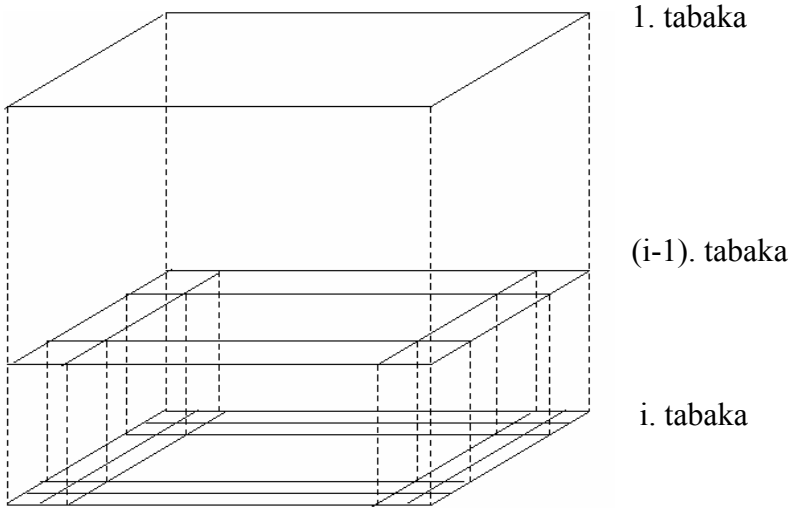
STING (Statistical Information Grid) metodu Wang, Yang ve Muntz tarafından 1997 yılında sunulmuştur [29]. Veri uzayını kare şeklinde hücrelere bölme temeline

dayanır. Veri uzayındaki her hücrenin içindeki verilerin ortalama, minimum, maksimum, değişim (variance) ve dağılım türü bilgilerini önceden hesaplar ve ana bellekte tutar. Daha sonraki işlemleri bu değerleri kullanarak gerçekleştirir.

Şekil 3.14'te STING algoritması için kullanılan ızgara yapısı görülmektedir. Izgara yapısı katmanlar şeklindedir. En üstte olan ve en az sayıda kareden oluşan tabakaya 1. tabaka adı verilir. Her bir tabaka bir öncekinden daha çok kare içererek i. tabakada eşik değeri olarak belirtilen sayıda kare hücre bulunur.

Izgara yapısı sayesinde en alt tabakadaki karelerin ortalama, maksimum, minimum, değişim değerleri kullanılarak daha üst tabakaların bilgileri kolayca hesaplanabilir.

Kümeleme işlemine 1. tabakadan başlanır, her bir kare içindeki değerler istenilen eşik değerleri tutuyorsa bu kareler bir alt katmana geçirilir. Eşik değeri tutmayan kareler bir alt tabakaya geçirilmez. 2. tabaka daha yüksek kare sayısına sahip olduğundan eşik değeri tespiti daha hassas yapılabilir. 2. tabakada eşik değerin altında kalan alanlar bir alt tabakaya geçirilmez. Bu şekilde ilerleyerek alt tabakalara inildikçe kümelenme alanları daha hassas olarak ortaya çıkar.



Şekil 3.18 STING algoritması Izgara yapısı [29].

STING algoritmasının hesaplanabilir karmaşıklığı O(n) dir. Bu yüzden küme bulma kalitesi ve performansı yüksektir. Algoritmanın etkinliği veri uzayının bölündüğü kare sayısı ile doğru orantılıdır. STING algoritması kare şekilli hücreler kullandığı

için bulduğu tüm kümelerin sınırları yatay ya da dikeydir, çapraz (diagonal) sınırları tespit edemediğinden bir miktar hesaplama hatası söz konusudur. Kare sayısı arttırıldıkça hesaplama hatası azalır.

3.4.4.2. WaveCluster Metodu

Sheikholeslami, Chatterjee ve Zhang tarafından VLDB'98 konferansında sunulmuştur [30]. WaveCluster metodu da ızgara tabanlı kümeleme metodu olduğu için veri uzayını çok boyutlu ızgara yapısına yerleştirir. Veri kümesindeki yoğunlukları bulmak için ızgara içindeki hücrelere Wavelet dönüşümü (Wavelet Transform) adı verilen bir yöntemi uygular.

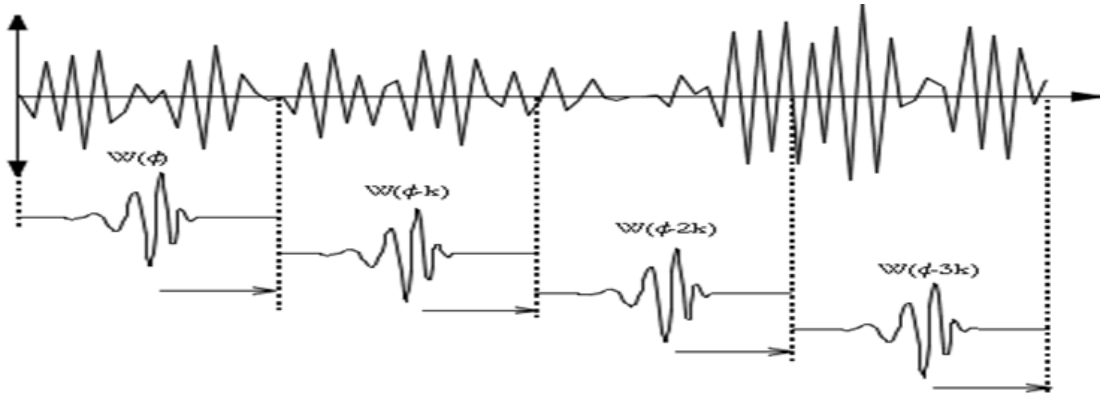
Wavelet dönüşümü, sinyali farklı frekans bandlarına ayırarak inceleyen bir sinyal işleme tekniğidir [23]. Wavelet dönüşümünün matematiksel açıklaması tez konusu dışında olduğu için açıklanmayacaktır.

WaveCluster metodu için iki adet giriş parametresi gereklidir:

- Kullanılacak ızgaradaki hücre sayısı,
- Wavelet dönüşümü için kullanılacak dalga türü ve kaç defa wavelet dönüşümünün uygulanacağını belirten değer.

Wavelet dönüşümün bir sinyale eş periyotlarda tekrarlı olarak uygulanması Şekil 3.15' te görülmektedir. WaveCluster metodu bu tür bir dönüşüm işlemini ızgara içindeki hücrelere uygular.

WaveCluster metodu gözetimsiz (unsupervised) kümeleme algoritmasıdır, Wavelet dönüşüm tekniği uygulanırken kümelenme alanları kendiliğinden ortaya çıkar. İstisna ve gürültü verilerin süzülmesinde oldukça etkili bir yöntem olduğu için resim işleme sistemlerinde kullanılmaktadır. Hesaplanabilir karmaşıklığı O(n) olduğu için hızlı ve çabuk sonuç verir. Tüm bu avantajlarına rağmen çok boyutlu veritabanlarında yeterli başarı gösteremediği gözlenmiştir [23].



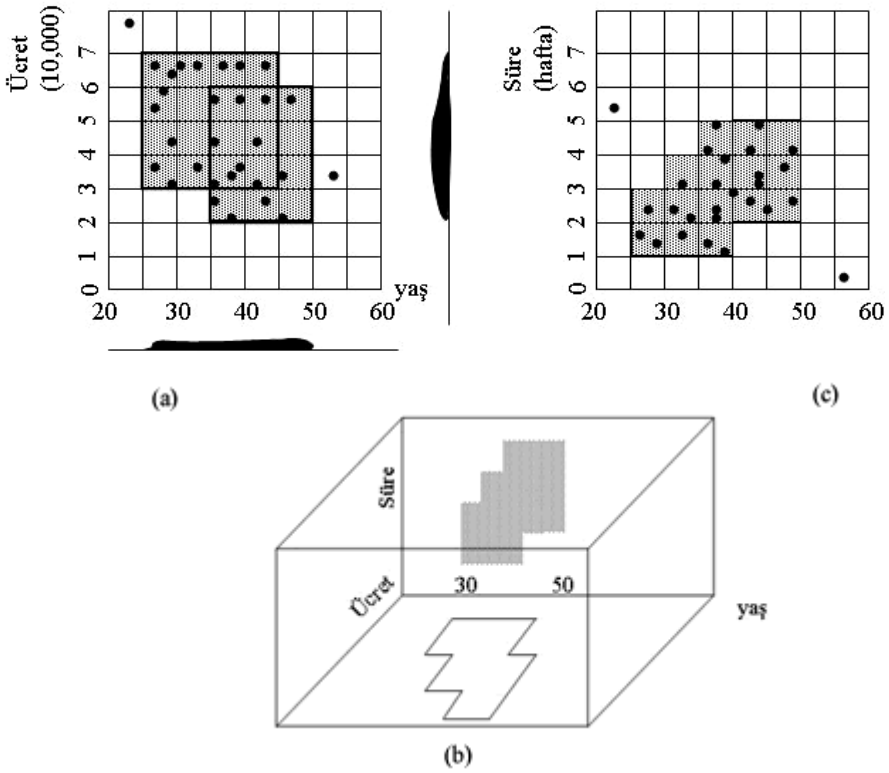
Şekil 3.19 WaveCluster metodunda kullanılan Wavelet dönüşümü [31]

3.4.4.3. CLIQUE

Agrawal, Gehrke, Gunopulos ve Raghavan tarafından SIGMOD'98 konferansında sunulmuştur [32]. CLIQUE (Clustering High-Dimensional Space) algoritması yoğunluk tabanlı ve ızgara tabanlı metodları tek bir algoritma altında toplamak amacıyla geliştirilmiştir. Çok boyutlu veritabanlarında iyi sonuçlar vermektedir.

CLIQUE algoritmasının temel prensibi şöyledir: Eğer k-boyutlu bir birimde yoğun alanlar var ise bu birimin (k-1) boyutlu izdüşümlerinde de aynı yoğun alanlar vardır. Bu yüzden k boyutlu birimdeki yoğun alanları tespit edebilmek için (k-1) boyutlu birimleri incelemek yeterlidir.

CLIQUE algoritması öncelikle n-boyutlu veri uzayını bir ızgara yapısı içine yerleştirir. İkinci aşamada, tüm nesnelerin n-boyutlu veri uzayının her bir boyutundaki bileşenlerini alır. Veri uzayının her bir boyutu Şekil 3.16 (a) ve (c)'de görüldüğü gibi 1 boyutlu bir çizgidir. Her bir boyut üzerinde verilerin yoğunlaştığı noktalar Şekil 3.16 (a)'da sağda ve alt taraftaki çizgilerde görüldüğü gibi belirlenir ve işaretlenir. Tüm boyutlardaki işaretli noktaların kesiştiği bölgeler Şekil 3.16 (b)'deki gibi birleştirildiğinde n-boyutlu düzlemdeki kümelenme bölgeleri elde edilir.



Şekil 3.20 CLIQUE algoritması çalışma yapısı [23].

CLIQUE, verilerin sıralanmasından bağımsız kümeleme işlemi gerçekleştirir, çok boyutlu veri uzayında kümelenmeleri tespit etmede yüksek performans gösterir.

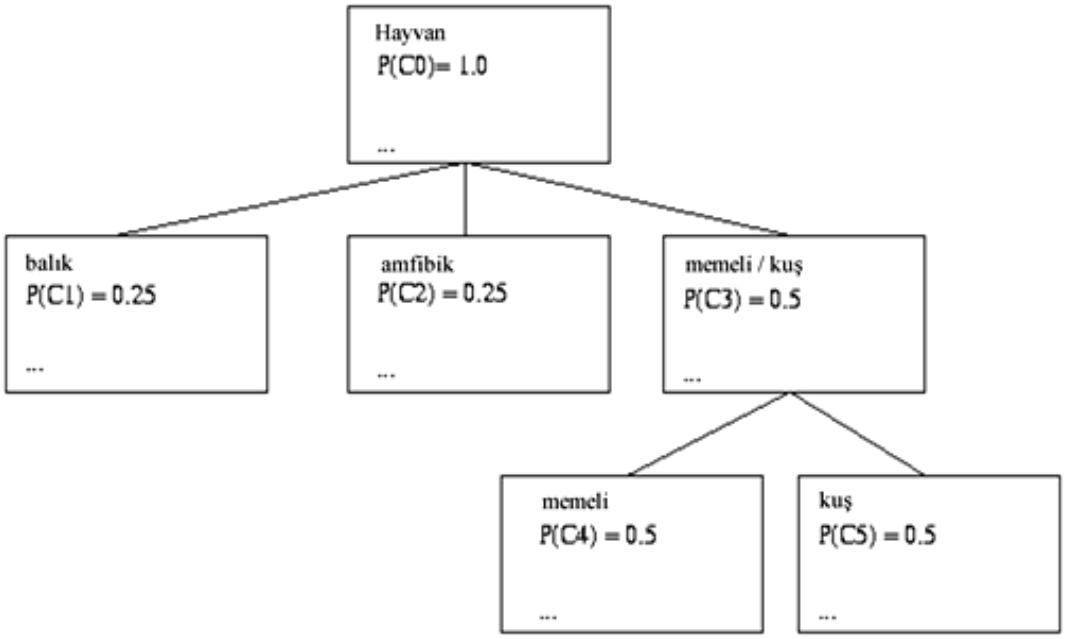
3.4.5. Model tabanlı metodlar

Model tabanlı metodlar eldeki verileri bir matematiksel model ile ifade etmeye çalışırlar. Bu metodlar verilerin belirli bazı olasılık teorilerinin karışımından oluşan bir mantık ile veri uzayına yerleştiklerini farzederler. Model tabanlı metodlar iki temel yaklaşımı kullanırlar: İstatistik yaklaşım ve yapay zeka yaklaşımı.

3.4.5.1. İstatistik yaklaşım

İstatistik yaklaşım kümeleme ve sınıflandırma yöntemlerinin her ikisini de kullanır. İstatistik yaklaşım diğer tüm kümeleme modellerinde olduğu gibi sadece kümelenmeleri ortaya çıkarmakla kalmaz, bunun yanında kümelerin genel karakterleri ile ilgili bilgiler de verir. Bu işleme kavramsal kümeleme denir [23].

COBWEB modeli istatistiksel yaklaşımı kullanan en tanınmış yöntemdir [33]. Bu yöntem Şekil 3.17'da görüldüğü gibi sınıflandırma ağacına benzer hiyerarşik kümeleme yapısı oluşturarak çalışır. Her bir ağaç yaprağında bulunan özelliğin olasılığı 0 ile 1 arası değerler ile gösterilmiştir. Örneğin şekilde bir hayvanın balık olma ihtimali %25, amfibik olma ihtimali 25 ve memeli/kuş olma ihtimali %50 olarak görülmektedir.



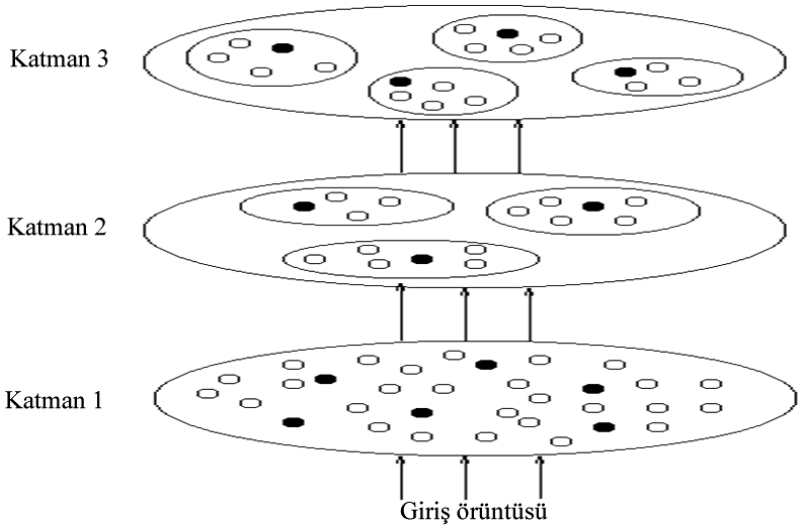
Şekil 3.21 İstatistik yaklaşım olan COBWEB modeli [33]

3.4.5.2. Yapay zeka yaklaşımı

Yapay zeka yaklaşımında her bir küme bir örnek gibi temsil edilir. Veritabanına yeni eklenen nesneler belirli bir uzaklık ölçümü sonucunda hangi örneğe benziyorlarsa o kümeye dahil edilirler [23].

Yapay zeka yaklaşımında iki metod ön plana çıkmaktadır; yarışmacı öğrenme (competitive learning) ve kendi kendini düzenleyen haritalar (self organizing maps-SOM). Şekil 3.18'de görüldüğü gibi, yarışmacı öğrenmede nesneleri işleyen nöronlar "kazanan hepsini alır" mantığı ile kümeyi temsil edebilmek için savaş verirler. Şekil 3.18'de siyah renkle gösterilen nöronlar savaşta galip gelen nöronları göstermektedir. Bu nöronlar kendi ağırlıklarını kümenin genel davranışı olarak bir üst katmana aktarmaya hak kazanırlar. Her katmanda savaş tekrar başlar ve bir kazanan

belirlenene kadar sürer. Üst katmanlara doğru ilerledikçe kümeleme işleminin hassaslığı artar [34].



Şekil 3.22 Yarışmacı öğrenme modeli [34]

Kendi kendine öğrenen haritalar metodu (SOM) diğer bir yapay zeka metodudur. SOM metodu hem bir kümeleme metodu hem de görselleştirme tekniği de sunar. SOM'un teorik boyutu oldukça fazla olduğu için bu tezde açıklanmayacaktır.

3.4.6. İstisna analizi

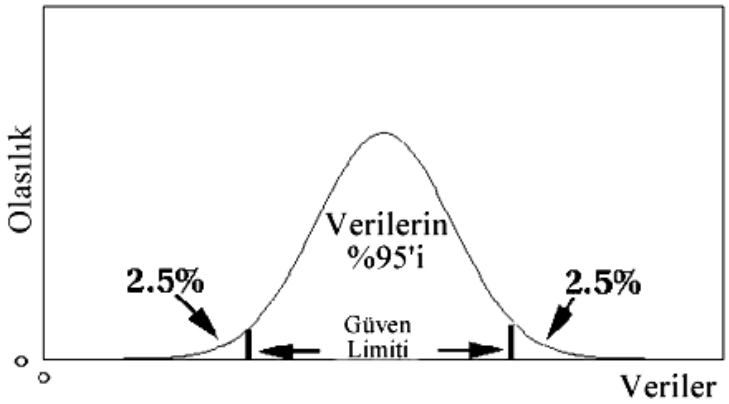
Bir veri kümesinin genel davranışından oldukça farklı özellikler gösteren üyelerine istisna (outlier) denir [23]. İstisnalar ölçme hatalarından, çalışma aksaklıklarından, yazılım hatalarından ya da veri içindeki doğal aşırılıklardan kaynaklanabilir. Örneğin bir müşteri otomasyon yazılımı müşteri yaşı girilmediği zaman varsayılan olarak 999 rakamını giriyorsa, kümeleme işlemi öncesinde bunun farkına varılmazsa istisna veri olarak görülür.

Bu bölüme kadar bahsedilen tüm veri madenciliği algoritmaları istisna verileri anlamlı kümelerden süzme amacını taşırlar. Fakat bazı durumlarda istisna veriler, küme içindeki verilerden çok daha fazla anlam ifade ederler. Örneğin müşterilerin kredi kartı harcamaları incelenirken genel harcama alışkanlıklarının dışına taşan noktalar bir kredi kartı usulsüzlüğünün ipuçlarını taşıyor olabilir.

İstisna analizi verilerin görselleştirilmesi sayesinde insan gözüyle tespit edilebilir. Fakat insan gözü her durumda istisna verileri açıkça göremeyebilir. Bu yüzden istisna tespiti için insan gözü yerine bu iş için geliştirilmiş metodlar kullanılır. İstisna analizi metodları istatistik tabanlı, uzaklık tabanlı ve sapma (deviation) tabanlı olarak ifade edilir [23].

3.4.6.1. İstatistik tabanlı istisna analizi

İstatistik tabanlı yöntem, veri kümesinin bir dağılım ya da olasılık modeli ile ifade edilebileceğini farzeder ve bu modele düzensizlik (discordancy) testi uygulayarak istisna verileri tespit etmeye çalışır. Düzensizlik testi için, veri kümesinin ortalama ve varyans değerleri gibi parametrelerin bilinmesi gereklidir.



Şekil 3.23 İstatistik tabanlı İstisna analizi yöntemi [23]

İstatistik tabanlı yöntem ile elde edilen bir grafik Şekil 3.19'da görülmektedir. Elde edilen dağılım grafiğinde %95'lik bölüme güven limiti adı verilir. Bu bölümdeki noktalar normal davranış gösteren verilerdir. Alt ve üst bölümlerdeki %2,5'lik alanlar ise istisna bölgeleridir.

İstatistik yöntemin iki önemli olumsuzluğu bulunur. Veri kümesinin sadece bir özelliğini inceleyebilir ve çoğunlukla veri kümesinin dağılım modeli belirlenemediği için istatistik yöntem uygulanamaz.

3.4.6.2. Uzaklık tabanlı istisna analizi

Uzaklık tabanlı yöntemler veri analizinde dağılım grafiği yerine nesneler arası uzaklıkları dikkate alarak istisna tespiti yaparlar. Bu yöntemin temel prensibi yeterince komşusu olmayan noktaları bulmaktır. Verilerin genel dağılım bölgesinden daha uzak noktalardaki nesneler istisna değere sahip nesnelerdir, genel dağılım bölgesine olan uzaklığı belirli bir eşik değeri aşan noktalar bu yöntem tarafından istisna olarak kabul edilir.

Uzaklık tabanlı istisna analizi için birçok algoritma geliştirilmiştir. Bunlardan en bilinenleri dizin tabanlı algoritmalar, iç içe döngü algoritmaları ve hücre tabanlı algoritmalardır. Uzaklık tabanlı yöntemler, istatistik yöntemler kadar hesaplama gücü gerektirmezler ve daha anlamlı sonuçlar üretirler [23].

3.4.6.3. Sapma tabanlı istisna analizi

Sapma tabanlı yöntemler istisna noktaları tespit etmek yerine istisna olmayan, genel eğilime uyan verilerin karakteristiğini çıkarmaya çalışır. Bu karakteristik yapıdan sapma gösteren noktaların istisna olduğu sonucuna varır.

Sapma tabanlı yöntemlerin en bilineni OLAP küpü tekniğidir. Günümüzde birçok veritabanı yönetim sisteminin içinde entegre olarak çalışan OLAP analiz birimleri bulunmaktadır [23].

4. K-MEANS ALGORİTMASI

4.1 Genel Bilgiler

En iyi bilinen ve yaygın kullanılan algoritmalardan biri olan k-means, verileri sınıflandıran bir kümeleme algoritmasıdır. Verilen nesneleri nitelik veya özelliklerine göre k adet sınıfa ayırmak amacıyla kullanılır. Sınıflandırma, verilerin en yakın veya benzer oldukları küme merkezleri (centroid) etrafına yerleştirilmesi ile gerçekleştirilir.

Bu çalışmada kullanılan k-means algoritması, J. MacQueen [10] tarafından 1967 yılında tanıtılmıştır. Çalışma yönteminde, Öklit uzaklığı temel alınarak kümeleme yapılmaktadır. Bu yöntem yıllardır bilimsel ve endüstriyel uygulamalarda en yoğun kullanılan kümeleme algoritması haline gelmiştir.

Algoritmaya k-means adı verilmesinin nedeni, algoritmanın çalışmasından önce sabit bir küme sayısına ihtiyaç duyulmasıdır. Küme sayısı k ile gösterilir ve elemanlarının birbirlerine olan yakınlıklarına göre oluşacak grup sayısını ifade eder. Buna göre k önceden bilinen ve kümeleme işlemi bitene kadar değeri değişmeyen sabit bir pozitif tamsayıdır.

Çok yaygın kullanımı olan bu algoritmanın aşağıda belirtildiği gibi birtakım zayıf yanları da bulunmaktadır:

• Algoritmanın başında giriş parametresi olarak bir k sayısına ihtiyacı vardır. Elde edilecek olan sonuçlar k sayısına göre değişkenlik gösterebilir. Eğer küme sayısı belirli değil ise deneme yoluyla en uygun sayı bulunur.

• Aşırı gürültü ve istisna veriler algoritmayla hesaplanan ortalamayı değiştirdiği için k-means algoritması gürültü ve istisnaya karşı çok duyarlıdır. Algoritma uygulanmadan önce veriler gürültü veya istisnadan temizlenebilir.

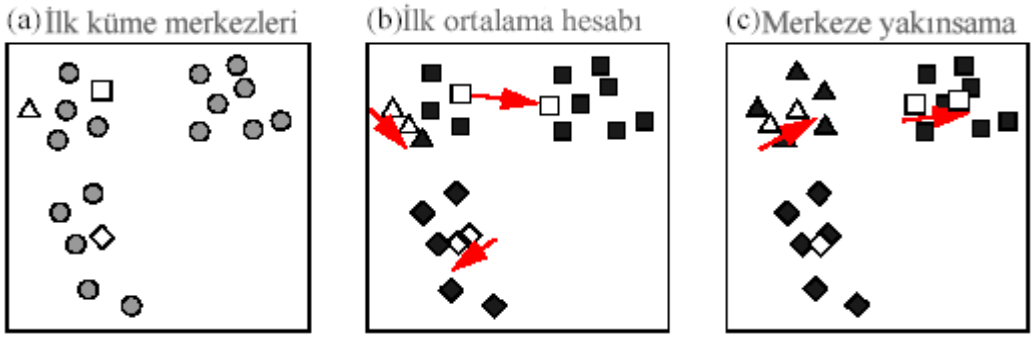
• Çakışan kümelerde iyi sonuç vermez.

• Her eleman aynı anda verilen bir kümenin içindedir veya dışındadır.

• K-means algoritması sadece sayısal veriler ile kullanılabilmektedir. Kategorik verilerin kümelenmesi için k-means algoritması bir çözüm sunmaz [24].

Açıklamada kolaylık olması açısından, bu bölümde algoritma iki boyutlu diyagramlar kullanılarak örneklenmiştir. Ancak uygulamada çok boyutlu elemanlarla, diğer bir deyişle çok eleman vektörü ile çalışılabilmektedir. Boyut sayısının artması algoritmada değişiklik yapılmasına neden olmamaktadır [1].

K-means algoritmasının pek çok çeşidi bulunmaktadır. Bir çok ticari yazılım paketi kümeleme işleminde bu algoritmanın çeşitlerini kullanmaktadır. Başlangıç küme merkezlerinin seçim şekli, kayıtları kümelerle ilişkilendirirken uzaklık yerine olasılık yoğunluğunun kullanılması gibi yaklaşımlar bu çeşitliliği oluşturmaktadır [1].
K-means algoritmasının çalışma şekline bir örnek şekil 4.1'de görülmektedir. Bu örnekte k = 3 olarak seçilmiş ve beyaz ◇△□ simgeleri şekil 4.1 (a)'da rastgele seçilen küme merkezlerini temsil etmektedir. Şekil 4.1 (b)'de geri kalan noktalar (◆▲■) aynı şekilli ve beyaz renk olan küme merkezlerine dahil edilerek ilk kümeler oluşturulur. Bu işlem sonunda küme merkezleri her kümedeki elemanların ortalaması dikkate alınarak tekrar hesaplanır. Değişen küme merkezleri şekil 4.1 (b)'de oklar ile gösterilmiştir. Şekil 4.1 (c)'de aynı işlem tekrar edildiğinde küme merkezlerinin değişimi görülmektedir. Bu şekilde başlangıç durumunda rastgele seçilen küme merkezleri, sürekli yinelemeler ile gerçek kümelenme alanlarının ortasına doğru yaklaşır. Bu işleme merkeze yakınsama (convergence) denir. Merkeze yakınsama minimum seviyeye geldiğinde veya durduğunda kümeleme işlemi sona erer.



Şekil.4.1 K-means kümeleme algoritması [35].

4.2 K-means Algoritmasının Adımları

Algoritmanın ilk adımında öncelikle küme merkezlerini veya diğer bir deyişle küme ortalamasını temsil edecek k adet eleman belirlenir. MacQueen algoritmasında küme merkezleri ilk k adet elemandan seçilir. Ancak elemanların değerleri birbirine çok yakın ise, seçim rastgele yapılır veya birbirinden uzak elemanlar seçilir. Bu noktaların her biri prototip olarak adlandırılır. Belirlenen bu elemanlar tek elemanlı başlangıç kümeleridir ve ilk küme merkezlerini oluştururlar. Kümenin ağırlıklı ortalama değerine sahip olan ya da bu değere en yakın olan elemanı küme merkezi olarak adlandırılır.

İkinci adımda, okunan elemanlar kendilerine en yakın k adet küme merkezinden birine dahil edilir. Elemanların küme merkezine olan yakınlık derecesini bulmak amacıyla çeşitli geometrik yöntemler kullanılır. Bunlardan bir tanesi, kümeler arasındaki sınırları belirleyerek, elemanların hangi küme merkezine daha yakın olduklarını tespit eder. Bunun için önce iki küme merkezi bir doğruyla birleştirilir. Bu doğrunun orta noktasından geçen ve doğruyu dik kesen başka bir doğru daha geçirildiğinde, bu doğru iki kümenin sınırı olarak kabul edilir. Bulunan sınır çizgisi dikkate alınarak, elemanların hangi kümeye dahil edileceği ortaya çıkar.

Noktalar arası uzaklığın hesaplanmasında en çok kullanılan yöntem Öklit bağıntısıdır. İki boyutlu bilgilerde, iki küme merkezinin birleştirilmesinde doğru kullanılırken, boyut sayısı arttığında doğru yerine düzlem kullanılır. Çok boyutlu bilgilerde çok boyulu düzlemler kullanılır. Algoritmanın geometrik gösteriminde küme sınırları yukarıda anlatıldığı şekilde belirlenmektedir. Bilgisayar programları ile geliştirilen k-means algoritmalarında ise, düzlemler yerine noktalar arasındaki uzaklıklar hesaplanarak, noktaların merkeze yakınlığı dikkate alınmaktadır [1].

Üçüncü adımda her bir kümeye eklenen yeni eleman ile, küme elemanlarının ağırlıklı ortalaması tekrar hesaplanarak yeni bir küme merkezi bulunur. Ağırlıklı ortalama kümenin her bir boyutundaki bütün elemanların ortalama değerlerinin alınması ile hesaplanır. Algoritmanın başında seçilen elemanlar küme merkezini oluştururken, ikinci döngü sonucunda bulunan yeni küme merkezleri artık bir küme elemanı değil,

sadece bir ortalama değerdir. Bundan sonraki seçim işlemlerinde küme merkezini bu yeni eleman temsil eder. Her bir döngüde elemanlar farklı bir kümeye dahil edilebilirler.

Kümeleme işlemi, tüm elemanların tekrar aynı veya farklı bir kümeye dahil edilmesiyle devam eder. Elemanların bir kümeye dahil edilmesi ve küme merkezlerinin tekrar hesaplanması işlemlerine ait döngü, küme sınırlarının değişimi bitene kadar devam eder. K-means algoritması ile uygulamalarda genellikle birkaç düzine döngü sonrası kararlı bir küme grubu ortaya çıkar.

Şekil.4.2 K-means algoritmasının adımları.

K-means algoritması bilgisayar programına uygulanırken aşağıdaki adımlar izlenir:

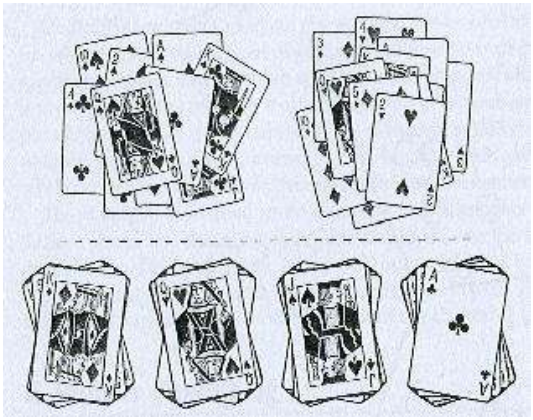
1. Küme sayısı (k) okunur. Bu değer algoritmaya dışardan verilir.
2. k adet rastgele küme merkezi belirlenir.İlk k eleman merkez olabilir.
3. Elemanların merkezleree yakınlıkları hesaplanır

4. Elemanlar yakın oldukları merkezlere göre kümelenir
5. Kümelerin ortalamaları hesaplanarak yeni küme merkezleri belirlenir
6. Kümelenecek başka eleman var mı? Hayır ise(3) e git.Evet ise dur.

Algoritmanın akış diyagramı şekil 4.2'de görülmektedir.

4.3 K Sayısının Kümelemeye Etkisi

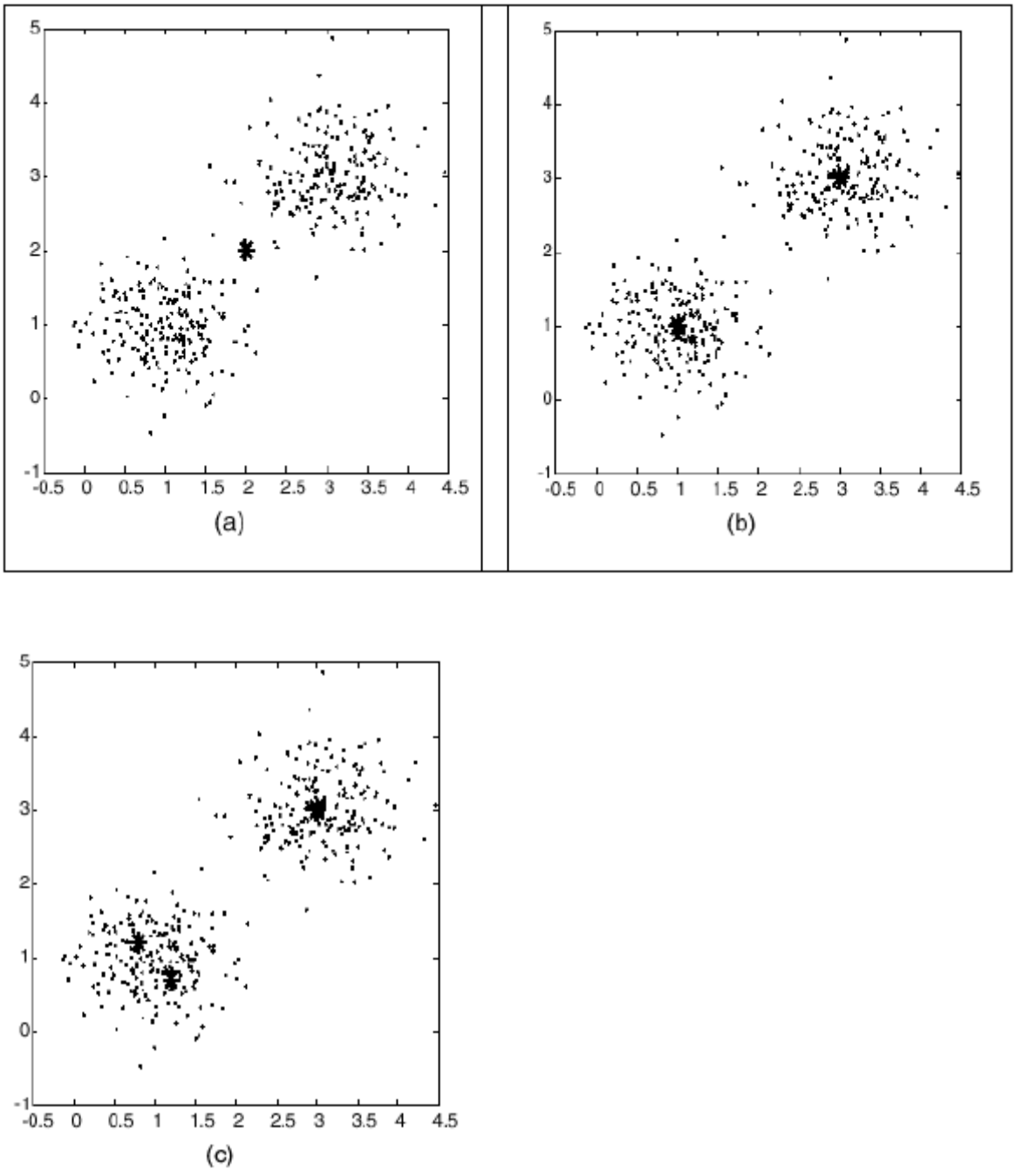
Kümeleme algoritmalarında, elemanların birbirlerine olan yakınlıklarına göre oluşturulan kümelerin sayısı k ile gösterilir. K işlem öncesinde bilinen ve kümeleme işlemi bitene kadar değeri değişmeyen sabit bir pozitif tamsayıdır. Şekil 4.3'de bir deste oyun kağıdının k=2 ve k=4 için kümeleme sonuçları gösterilmektedir. Şekilden görüleceği gibi k'nın farklı değerler alması, her biri geçerli olan çok farklı kümeler oluşmasını sağlamaktadır. Hangisinin daha etkili olduğu, hangi kümelemenin kullanılacağına bağlıdır.



Şekil.4.3 Oyun kağıtlarının k=2 ve k=4 için kümelenmesi [1]

K-means ve benzeri kümeleme algoritmaları k sayısının belirlenmesi konusunda bir çözüm sunmazlar. Ancak birçok durumda, özel bir k değerinin belirlenmesi gerekli olmaz. Analiz aşamasında k değerinin tespiti için ön çalışma yapılır. Tahmini bir değer kullanılarak kümeleme algoritması çalıştırılır ve alınan sonuçlar değerlendirilir. Değerlendirme sonucunda beklenen kümeleme görülmez ise, başka bir k değeri kullanılarak tekrar kümeleme algoritması çalıştırılır veya veriler üzerinde

değişiklik yapılabilir. Algoritmanın her çalıştırılması sonrasında, ortaya çıkan kümelerin etkinliğini hesaplamak için, küme içindeki kayıtların arasındaki ortalama uzaklık ile kümeler arası ortalama uzaklık karşılaştırılır. Hesaplamada başka yöntemler de kullanılabilir. Bu yöntemler algoritmaya dahil edilebilir. Ancak ele alınan uygulama açısından sonucun yararlılığının belirlenmesi için kümeler mutlaka daha öznel temelde değerlendirilmelidir.



Şekil.4.4 Küme sayısına göre K-means algoritmasının sonuçları [36].

Aynı veri grubuna ait farklı k değerleri ile k-means algoritması uygulandığında ortaya çıkan sonuç şekil 4.4'de yer almaktadır. Şekil 4.4 (a)'da k=1 için bütün elemanlar tek bir küme oluşturmuştur. Daha gerçekçi bir kümeleme, şekil 4.4 (b)'de k=2 için ortaya çıkmıştır. Şekil 4.4 (c)'de k=3 için birbirine daha yakın elemanların yer aldığı grupta üçüncü bir küme oluşmuştur.

Bazı uygulamalarda k-means algoritmasının çalıştırılması sonucunda, verilerin büyük çoğunluğunun aynı kümeye dahil edildikleri görülür. Büyük kümenin çevresinde bir kaç küçük küme de yer alır. Bunun nedeni verilerin büyük çoğunluğunun birbirine yakın özellikler taşıması ve az sayıda verinin farklı özellikte olmasıdır. Bu tipteki uygulamalara örnek olarak, sahtekarlık tespiti ve üretim hataları verilebilir. Her iki uygulamada da önemli sayıda veri, istenen özellikleri taşımaktadır. Gürültü/istisna adı verilen az sayıda veri, istenilen özelliklerin dışında kaldıklarıiçin büyük kümenin dışında konumlanmışlardır. Bu konuya ikinci bölümde, sıradışılık analizinde değinilmiştir.

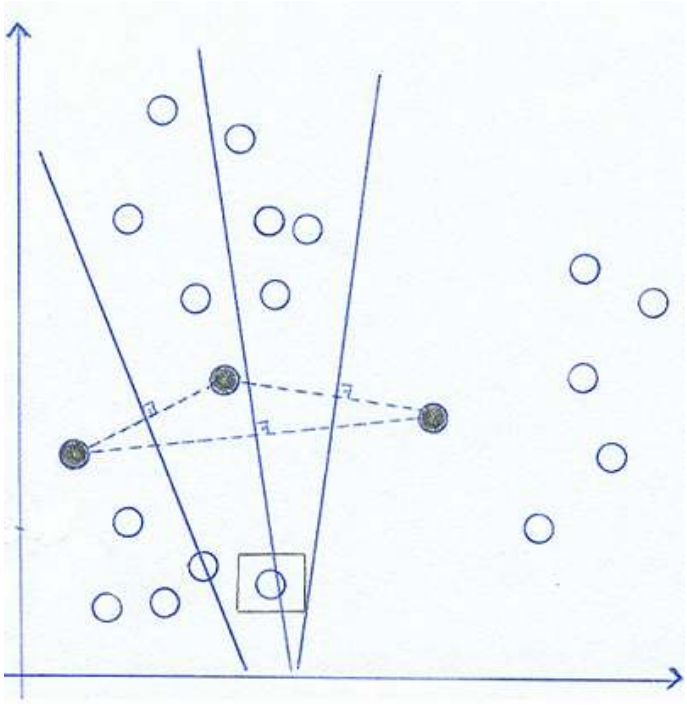
K-means algoritması ile verileri kümelere ayırmak için geometrik veya aritmetik hesaplama yöntemleri kullanılmaktadır. Bu yöntemler aşağıda örneklerle tanıtılmıştır.

4.3.1 Geometrik hesaplama

Bu yöntemde, kümelenecek veriler koordinat sisteminde birer nokta olarak ele alınır. Noktalar arası uzaklığın hesaplanmasında en çok kullanılan yöntem Öklit bağıntısıdır. Bu bağıntı üçüncü bölümde anlatılmıştır. Küme sınırlarının belirlenmesi için iki boyutlu sistemde doğru, boyut sayısı arttığında ise doğru yerine düzlem kullanılır. Çok boyutlu bilgilerde çok boyulu düzlemler kullanılır. Geometrik hesaplama yöntemini bir örnek üzerinden açıklamak için, 20 elemanın üç kümeye bölünmesi aşağıda şekillerle anlatılmıştır.

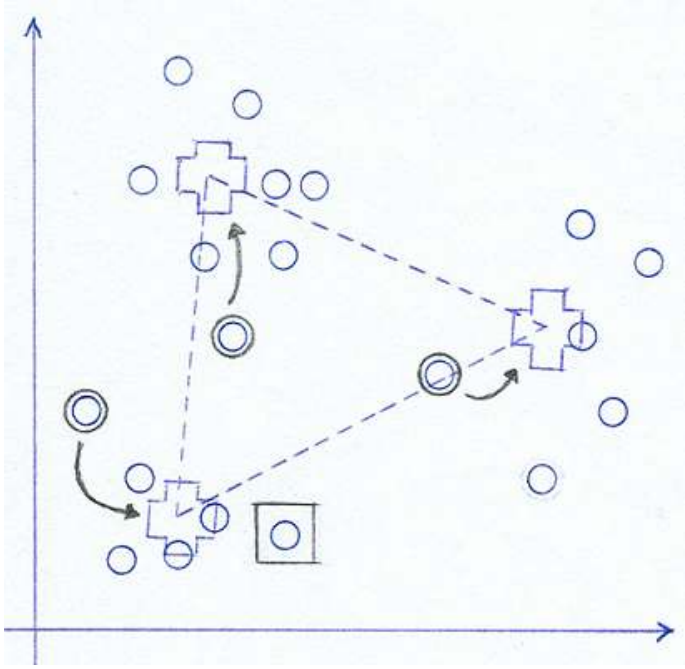
İlk adımda başlangıç küme merkezlerini temsil edecek 3 eleman seçilir. Şekil 4.5'de içi dolu çember şekliyle gösterilen noktalar ilk küme merkezleridir. İkinci adımda, diğer bütün elemanları yakın oldukları bir küme merkezlerine dahil etmek için her

bir kümenin sınırı belirlenir. Küme merkezlerinden eşit uzaklıkta bulunan noktalar küme sınırını oluşturur. Bu noktaları belirlemek için küme merkezleri birer doğruyla birleştirilir. Doğruların orta noktalarından geçen ve doğruları dik kesen başka doğrular çizildiğinde, kümelerin sınırları ortaya çıkarılır. Şekil 4.5'de ilk küme merkezleri noktalı çizgiler ile birleştirilmiştir. Bunları dik kesen kalın çizgiler küme sınırlarıdır. Kümelerin sınır çizgilerine bakılarak diğer elemanların hangi kümeye dahil edileceği görülür.



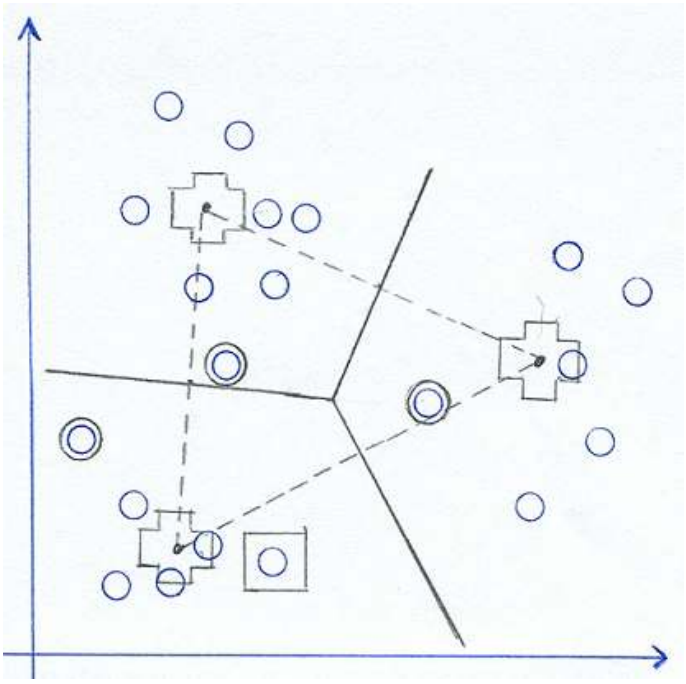
Şekil.4.5 Geometrik hesaplama yöntemiyle ilk kümelerin belirlenmesi [1]

Üçüncü adımda, her bir kümedeki elemanların ortalama değerleri alınarak küme merkezleri yeniden hesaplanır. Şekil 4.6'da yeni küme merkezleri artı sembolünün içinde yer almaktadır. Önceki küme merkezlerinin çevresine çember çizilerek, merkezlerin nereden nereye değiştiği gösterilmiştir. Şekil 4.5'de kare içine alınarak işaretlenen eleman, ilk döngü sonrasında ikinci kümeye dahil olmuştur. İkinci döngü sonrasında ise şekil 4.6'da görüldüğü gibi birinci kümeye dahil edilmiştir.



Şekil.4.6 Noktaların kümelere dahil edilmesi sonrasında yeni küme merkezleri [1]

Döngü içinde ikinci adım tekrarlandıktan sonra, bütün elemanlar en yakın oldukları küme merkezlerine yeniden dahil edilirler. Her döngüde küme merkezlerinin değişimiyle küme sınırları da değişmektedir. Şekil 4.7'de yeni küme sınırları gösterilmektedir. İşlem döngüsü, küme sınırlarının değişimi durana kadar sürer.



Şekil.4.7 Her döngü sonrasında küme sınırları değişmektedir [1].

4.3.2 Aritmetik hesaplama

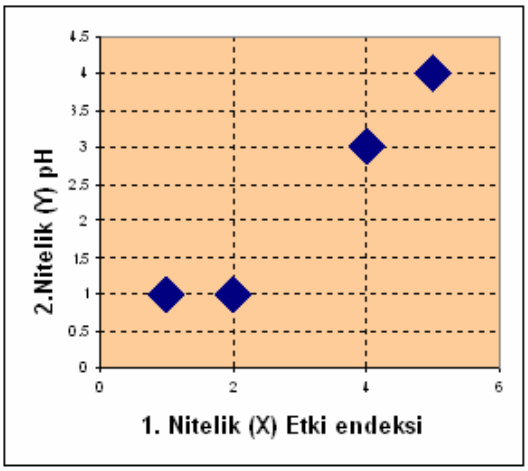
Bilgisayar programları ile geliştirilen k-means algoritmalarında aritmetik hesaplama yöntemlere başvurulmaktadır. Aritmetik hesaplamada, doğrular veya düzlemler yerine eleman değerleri arasındaki uzaklıklar hesaplanarak, elemanların merkezlere yakınlığı bulunmaktadır [1].

Aritmetik hesaplama yöntemini bir örnek üzerinden açıklamak için, tablo 4.1'de görülen veriler kullanılmıştır. Tabloda 4 çeşit ilaca ait veriler yer almaktadır. Her bir ilaç nesne olarak, ilaçların etki endeksi ve ph değeri nitelik değerleri olarak ele alınmıştır. Bu örnekte veriler iki kümeye (k=2) ayrılacaktır.

Tablo 4.1: Kümeleme için kullanılanacak veriler [37]

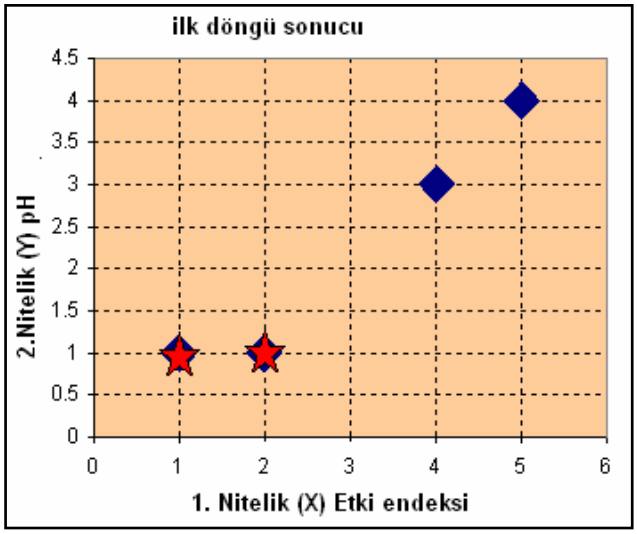
Nesne	1.Nitelik (x) etki endeksi	2.Nitelik (y) pH
İlaç A	1	1
İlaç B	2	1
İlaç C	4	3
İlaç D	5	4

Aritmetik hesaplamada, tablodaki her bir ilaca (nesneye) ait nitelikler, nitelik uzayında (x,y) ile bir koordinat noktasını göstermektedir. Her bir nitelik bir koordinat boyutuna karşılık gelmektedir. İki nitelik iki boyutlu Buna göre, A ilacı (1,1), B ilacı (2,1), C ilacı (4,3) ve D ilacı (5,4) noktası ile temsil edilmektedir. İlaçların iki boyutlu koordinat sistemindeki görüntüsü şekil 4.8'de yer almaktadır.



Şekil.4.8 Aritmetik hesaplamaya uygulanacak veriler [37]

Birinci adımda, ilk küme merkezleri olarak, (0,0) noktasına yakınlıkları nedeniyle A ve B ilaçları seçilmiştir. Küme merkezlerinin koordinatları (1,1) ve (2,1)'dir. İlk küme merkezleri şekil 4.9'da gösterilmiştir.



Şekil.4.9 Aritmetik hesaplamada şeçilen ilk küme merkezleri [37]

İkinci adımda, seçilen ilk küme merkezleri ile diğer elemanlar arasındaki uzaklık hesaplanır. Öklit bağıntısı kullanılarak hesaplanan uzaklık matrisindeki her sütun bir ilacı (nesneyi) temsil etmektedir. Uzaklık matrisinin ilk satırında nesnelerin ilk küme merkezine olan uzaklıkları, ikinci satırında aynı şekilde, nesnelerin ikinci küme merkezine olan uzaklıkları yer alır. Örneğin ilk satırda A ilacının koordinat

noktası ile, B, C ve D ilaçlarının koordinat noktaları arasında hesaplanan uzaklık değerleri yer almaktadır. İkinci satırda B ilacının koordinat noktası ile, A, C ve D ilaçlarının ilaçlarının koordinat noktaları arasında hesaplanan değerler yer almaktadır. Hesaplanan ilk uzaklık matrisi D aşağıdaki gibidir;

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) \\ \mathbf{c}_2 = (2,1) \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

Küme merkezlerinin kendilerine uzaklıkları olmadığından, matrisin (1,1) ve (2,1) indisli değerleri sıfırdır. Öklit bağıntısını kullanarak hesaplamaya örnek olarak, C noktasının A ve B merkezlerine olan uzaklığı şöyle hesaplanır; C noktası (4,3) ile ilk küme merkezi (1,1) noktası arasındaki uzaklık $\sqrt{(4-1)^2+(3-1)^2} = 3.61$, ikinci küme merkezi (2,1) noktası arasındaki uzaklık ise $\sqrt{(4-2)^2+(3-1)^2} = 2.83$ olur.

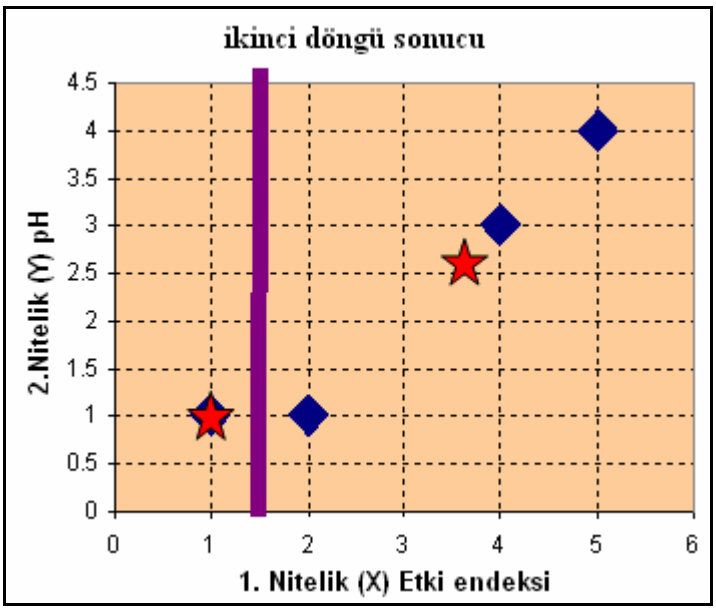
Üçüncü adımda, uzaklık matrisinde bulunan değerlere göre nesneler bir kümeye dahil edilir. Her nesne için matrisin satırları arasındaki değerlerin en küçüğü bulunur. Örneğin D nesnesi için birinci ve ikinci satır değerleri 5 ve 4.24'dür. En küçük değer 4.24 olduğu için D nesnesi ikinci kümeye dahil edilir. Aynı şekilde, C nesnesinin satır değerleri 3.62 ve 2.83'dür. En küçük değer 2.83 olduğu için C nesnesi ikinci kümeye dahil edilir. A ve B nesneleri, bu aşamada merkezleri oldukları kümelerin içinde kalırlar. Nesnelerin dahil edildikleri kümeler, ayrı bir küme matrisinde gösterilir. Aşağıdaki G isimli küme matrisinde, ilk satır birinci kümeyi, ikinci satır da ikinci kümeyi göstermektedir. Nesneler kümeye dahil ise 1, değil ise 0 değerini alırlar. Matristen de görüldüğü gibi, A nesnesi birinci kümenin hem merkezi hem de tek elemanıdır. B nesnesi ise ikinci kümenin merkezidir ve bu kümede C ve D nesneleri diğer küme elemanlarıdır.

$$\mathbf{G}^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

Küme matrisinin oluşturulmasıyla ilk döngü sona erer. İkinci döngü yeni küme merkezlerinin belirlenmesi işlemi ile başlar. Birinci kümede tek eleman olduğu için küme merkezinin değeri (1,1) değişmez. İkinci kümenin yeni küme merkezini hesaplamak için her bir nesnenin x ve y değerlerinin aritmetik ortalaması alınır. Elde edilen ortalama x ve y değeri, yeni küme merkezinin koordinat değeridir. İkinci kümenin üç elemanı ile ortalama x ve y değerleri aşağıdaki gibi hesaplanır;

$$c_2 = (\frac{2+4+5}{3}, \frac{1+3+4}{3}) = (\frac{11}{3}, \frac{8}{3})$$



Şekil.4.10 Aritmetik hesaplamada ikinci döngüde oluşan küme merkezleri [37]

İkinci döngüde hesaplanan yeni küme merkezleri ile ortaya çıkan kümeler şekil 4.10'da gösterilmektedir. Şekilde de görüldüğü gibi ikinci kümenin merkezi bir nesne değil bir noktadır. Yeni küme merkezleri oluşturulduktan sonra, diğer nesnelerin yeni merkezlere olan uzaklıkları tekrar hesaplanır. Hesaplama birinci döngünün ikinci adımında olduğu gibi Öklit bağıntısını kullanarak yapılır. İkinci döngüde oluşan uzaklık matrisi aşağıdaki gibidir;

$$\mathbf{D}^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) \\ \mathbf{c}_2 = (\frac{11}{3}, \frac{8}{3}) \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

Uzaklık matrisinde bulunan en küçük değerlere göre nesneler bir kümeye dahil edilir. Bunun için ilk döngüde olduğu gibi her nesne için matrisin birinci ve ikinci satırlarındaki değerlerin en küçük olanı bulunur. En küçük uzaklığa göre her bir nesne en yakın olduğu kümeye dahil edilir. Küme matrisinin oluşturulmasıyla, yeni kümelenme ortaya çıkar. Bu sefer A ve B nesneleri birinci kümenin elemanları haline gelirler, C ve D nesneleri ise ikinci kümeyi oluştururlar. G küme matrisi şöyledir;

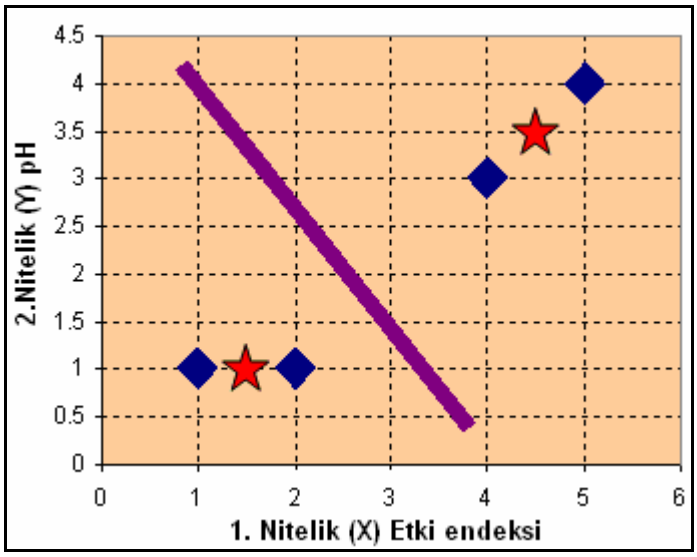
$$\mathbf{G}^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

Üçüncü döngü yeni küme merkezlerinin belirlenmesi işlemi ile başlar. Yeni merkezlerin koordinatlarını hesaplamak için, her bir kümeyi oluşturan nesnelerin x ve y değerlerinin aritmetik ortalaması alınır. Birinci ve ikinci kümelerin yeni küme merkezleri şöyledir;

$$c_1 = (\frac{1+2}{2}, \frac{1+1}{2}) = (1\frac{1}{2}, 1) \qquad c_2 = (\frac{4+5}{2}, \frac{3+4}{2}) = (4\frac{1}{2}, 3\frac{1}{2})$$

Bu döngüde her iki küme merkezi de nesne değil, nokta olmuştur. Şekil 4.11 üçüncü döngüde ortaya çıkan kümelenmeyi göstermektedir.



Şekil.4.11 Aritmetik hesaplamada üçüncü döngüde oluşan küme merkezleri [37]

Küme nesnelerinin yeni küme merkezlerine olan uzaklıkları tekrar hesaplanır. Oluşan uzaklık matrisi şöyledir;

$$\mathbf{D}^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1\frac{1}{2}, 1) \\ \mathbf{c}_2 = (4\frac{1}{2}, 3\frac{1}{2}) \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

Uzaklık matrisinden, en küçük uzaklığa göre her bir nesne en yakın olduğu kümeye dahil edilir. Böylece ortaya çıkan küme matrisi şöyledir;

$$\mathbf{G}^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

Küme matrisinden görüldüğü gibi A ve B nesneleri birinci, C ve D nesneleri ise ikinci kümenin elemanları olarak kalmışlardır. İkinci ve üçüncü döngü sonucunda elde edilen küme matrisleri birbirinin aynı olmuş ve üçüncü döngü sonucunda kümeleme değişmemiştir. Bu sonuçla, nesnelerin artık kümeler arasında hareket

etmeyeceği anlaşılmaktadır. Böylede k-means ile kümeleme işlemi kararlı bir noktaya gelmiştir. Yeni bir döngüye ihtiyaç bulunmamaktadır. İlaçların kümeleme sonucu tablo 4.2'de görülmektedir.

Tablo 4.2: Kümeleme sonucunda oluşan gruplama [37]

Nesne	1.Nitelik (X) etki endeksi	2.Nitelik (Y) pH	Sonuç kümeleme
İlaç A	1	1	1
İlaç B	2	1	1
İlaç C	4	3	2
İlaç D	5	4	2

4.4 K-means algoritmasının matematiksel yorumlanışı

Geometrik ve aritmetik hesaplama yöntemlerinin matematiksel temelleri aynıdır. Bu kısımda algoritmanın matematiksel ifadeleri, programlama yapısının içinde gösterilerek anlatılmıştır.

Kümeleme başlangıcında k adet küme merkezi (w_1,w_2,w_3 .. w_k) ve her bir küme (i_1,i_1,i_3 .. i_n) olmak üzere

$$w_i = i_l, j \in \{1,...,k\}, l \in \{1,...,n\} \tag{4.1}$$

durumundadır. Cj ifadesi j. elemanı temsil etmek üzere, kümeleme işleminin kalitesi denklem 4.2'deki aşağıdaki hata fonksiyonu ile ifade edilir [24]:

$$E = \sum_{j=1}^{k} \sum_{i_l \in C_j} \left| i_l - w_j \right|^2 \tag{4.2}$$

K-means algoritmasının en büyük problemi uygun k değerini tespit edememesidir. Bu yüzden, en uygun kümelenmeleri bulabilmek için farklı k değerleri ile birçok deneme yapmak gerekmektedir.

d boyutlu veritabanında n adet örüntü (küme) olması durumunda, k-means algoritmasının hesaplanabilir karmaşıklığı üç parça halinde incelenir:

1) Matematiksel yorumlanışta birinci döngü için gereken süre O(nkd) dir.
2) İkinci döngüde küme merkezlerini, diğer bir ifade ile küme ortalamalarını hesaplamak için gereken süre O(nd) dir.
3) Hata fonksiyonunu hesaplamak için gereken süre de O(nd) dir.

K-means algoritmasının matematiksel yorumlanışı şöyledir [38];

function K-means()

(1) k adet küme merkezi (w_1,w_2,w_3 .. w_k) için

$w_i = i_l, j \in \{1,...,k\}, l \in \{1,...,n\}$ olacak şekilde ilk değeri belirle

(2) Her C_j kümeye w_j değerini atamak için aşağıdaki döngüyü uygula:

(3) Repeat

for each (giriş vektörü i_l, where $l \in \{1,...,n\}$),

do

$$|i_l - w_{j*}| <= |i_l - w_j|, j \in \{1,..,k\}$$

olmak üzere w_{j*} küme merkezini C_{j*} kümesine en yakın olan i_l değerini C_{j*} kümesine ekle

for each (C_j kümesi, where $j \in \{1,...,k\}$),

do

w_j küme merkezini C_j kümesindeki tüm elemanların ortalaması olacak şekilde güncelle. Bu durumda:

$$w_j = \sum_{i_l \in C_j} \frac{i_l}{|C_j|}$$

olmalıdır.

Hata fonksiyonunu hesaplamak için aşağıdaki denklemi uygula:

$$E = \sum_{j=1}^{k} \sum_{i_l \in C_j} \left| i_l - w_j \right|^2$$

(4) Until E değeri hissedilir değişiklik göstermeme durumu veya küme üyelikleri değişmeme durumu.

5. TIBBİ VERİLERLE VERİ MADENCİLİĞİ UYGULAMASI

5.1 Giriş

İnsanların deneyimlerden sonuç çıkartma yeteneği, geçmişten uygun örneklerin tanınması yeteneğine bağlıdır. Hastalıklara teşhis koyan bir doktor, öncelikle deneyimlerinden benzer vakaları tanımlar ve ardından bu vakaların bilgilerini eldeki probleme uygular. Bilinen vaka kayıtlarının tutulduğu veritabanı, sınıflandırılmış kayıtlar içinden yeni vakaya benzeyenleri bulmak için taranır. Mevcut hasta için en etkili tedavi, muhtemelen benzer hastaların sonuçlarından elde edilen bilgilerle yapılan tedavidir [1].

Bu tezde tıp alanında geçmiş kayıtları kullanmanın önemi dikkate alınarak gırtlak kanseri ameliyat verileri için bir analiz aracı geliştirilmiştir. Bu çalışmanın amacı, veri madenciliğinde bir kümeleme tekniği olan k-means algoritmasını incelemek ve bu algoritmayı kullanarak geliştirilen bir yazılım aracılığı ile gırtlak kanseri ameliyat verilerinin analizini yapmaktır. Uygulamanın tıp doktorlarının kullanımına uygun şekilde verileri çeşitli açılardan analiz etmesi hedeflenmiştir.

Ancak klasik hastane bilgi sistemleri daha çok idari ve finansal konulara ağırlık verecek şekilde tasarlanmaktadır. Hastaların ayrıntılı teşhis ve tedavi bilgileri ancak gereksinim duyulması halinde ve kimi zaman doktorların bireysel çabaları ile toplanıp kaydedilmektedir. Bu çalışmada kullanmak amacıyla tıbbi veri araştırılırken gırtlak kanseri ameliyat verileri bulunmuştur. Veriler çalışma için özel bir seçim değildir.

Bu bölümde, gırtlak kanseri ameliyatları ile ilgili verilerin tutulduğu veritabanı kullanılarak geliştirilen yazılım uygulaması anlatılmaktadır. Yazılım, veritabanından okunan verileri 4 farklı açıdan analiz etmektedir. Dördüncü bölümde açıklanan k-means kümeleme algoritması analizde kullanılmış ve algoritmanın gösterdiği davranışlar ile performansı incelenmiştir.

5.2 Hastalık Hakkında Genel Bilgiler

Kanser, anormal hücrelerin kontrolsüz çoğalması ve yayılması olarak bilinen bir grup hastalığa verilen isimdir. Gırtlak kanseri ise, Kulak Burun Boğaz Hekimliğinde en sık görülen kanser türlerinden biridir. Gırtlak (larenks), boğazın hemen altında ses tellerinin bulunduğu bir organdır ve gıda alımı sırasında besinlerin nefes borusuna kaçmasını engeller. Gırtlak kanseri gırtlağın herhangi bir kısmında gelişebilir ve çoğu zaman ses kısıklığı ile erken bulgu verir. Genellikle 50-60 yaş grubundaki erkeklerde sık görülür. Sigara bu kanser türü için en önemli risk etkenidir. Yoğun alkol kullanımı da riski arttırır. Tedavi tümörün türü, yeri ve evreye göre belirlenir. En önemli tedavi şekli cerrahidir ve bunun yanında ışın tedavisi kullanılabilir.

TNM sınıflaması, malignant(kötü huylu) tümörler için standart bir kanser evrelendirme sistemidir. Bir çok tümörün kendi TNM sınıflaması bulunur. Sınıflandırmanın genel şekli şöyledir: Zorunlu parametreler T, N ve M'dir. T: Tümör büyüklüğünü temsil eder ve 0 ile 4 arası değer alır. N: Node değeri, tümörün lenflere yayılımını ifade eder ve 0 ile 3 arası değer alır. M: Metastas, 0 veya 1 değeri alır ve diğer organlara yayılımı ifade eder [22]. Gırtlak kanseri vakalarında dikkate alınmadığından metastas parametresi veritabanında tutulmamaktadır.

Tedavide patolojik ve diğer görüntü verilerine dayanarak tümör hakkında tahmini bilgi edinilir ve kanserin evresi öngörülür. Bu bilgiler ışığında yapılacak ameliyat şekline karar verilir. Hastanın yaşı da göz önünde tutulan bir etkendir. Ameliyat sırasında nadiren de olsa öngörülen bilgilerden farklı bir görüntüyle karşılaşılabilir. Tümör büyüklüğü öngörülenden farklı olabilir ve kanser evresi tahmin edilenden daha yüksek olabilir. Bu durum ameliyat sırasında uygulanan tekniğin değiştirilmesini gerektirir. Ameliyatla alınan tümör daha sonradan nüks edebilir. Hastaların durumu daha sonraki yıllarda da izlenir ve tümör 5 yıl içinde nüks etmez ise hastanın sorundan kurtulduğu kabul edilir.

5.3 Veritabanının Çalışma için Hazırlanması

Bu çalışmada kullanılan gırtlak kanseri ameliyat verileri Kocaeli Üniversitesi Tıp Fakültesi Hastanesi, Kulak, Burun ve Boğaz Bölümünden alınmıştır. Veritabanı 1995 yılından bu yana toplanan 400 kayıt içermektedir.

Diğer bir çok veritabanında olduğu gibi tıp veritabanlarının da program aracılığı ile kullanılması için bozuk verilerden temizlenmesi ve düzenlenmesi gerekir. Hatalı girilmiş bilgilerin olması sık rastlanır bir durumdur [39]. Çalışmada kullanılan veritabanında rastlanan hatalar şunlardır;
Boş bırakılmış alanlar, birkaç bilginin birleştirilerek aynı alana yazılması, bir durumun değişik kayıtlarda birkaç farklı isimle yer alması, benzer şekilde aynı anlama gelen bilgilerin farklı formatlarda yazılması, yanlış yazılmış tıbbi terimler.

Bu çalışmada kullanılan gırtlak kanseri ameliyat bilgilerinin tutulduğu veritabanı öncelikle bozuk verilerden temizlenmiş ve program aracılığı ile kullanılabilir şekilde düzenlenmiştir. Veriler Excel tablosu şeklinde alınmış ve Access veritabanına aktarılmıştır. Düzenlemeler için Access sorgu nesnesi kullanılmıştır. Veritabanında aşağıdaki işlemler yapılmıştır:

Hasta ismi ve adresi gibi analiz için gerekli olmayan alanlar çıkartılmıştır. Boş (null) değer içeren sayısal alanlara 0 değeri atanmıştır. K-means algoritmasında karaker alanlar kullanılamadığından, patoloji, survive ve operasyon alanları sayısal karşılıklarına çevrilmiştir. Patoloji sonucu yassı epitel hücre (tabloda “Yeh Ca” olarak kısaltılmıştır) ise bu alanın değeri 1, diğer sonuçlar için 0 yapılarak veritabanına yeni eklenen patoloji_kodu alanına yerleştirilmiştir. Nüks bilgisi, ölüm nedeni ve tarihini içeren survive alanından nüks ve hayatta bilgileri alınarak veritabanına yeni eklenen aynı isimli alanlara yerleştirilmişlerdir. Nüks varsa nüks isimli alanın değeri 1, yoksa 0 yapılmıştır. Benzer şekilde, hasta hayatta ise hayatta isimli alanın değeri 1, aksi durumda 0 yapılmıştır.

Çalışmada kullanılmak amacıyla düzenlenen veritabanı Tablo 5.1'de yer almaktadır.

Tablo 5.1: Çalışmada kullanılan gırtlak kanseri ameliyat bilgileri veritabanı

Yas	Patoloji	preop_T	preop_N	preop_evre	postop_T	postop_N	postop_evre	Operasyon	Operasyon_tarihi	nuks	hayatta	patoloji_kodu
45	Epidermoic	T4	N1	4				TL	05.10.2000	0	1	0
40	YEH Ca	T2	N0	2				SCL	02.11.2000	0	1	1
51	YEH Ca	T3	N0	3	T4	N2	4	TL	21.11.2000	1	0	1
70	YEH Ca	T4	N2	4				TL	28.11.2000	0	0	1
43	YEH Ca	T1	N1	3	T2	N2	4	SGL	08.12.2000	0	1	1
39	YEH Ca	T4	N2	4				TL	04.01.2001	1	0	1
60	YEH Ca	T2	N0	2				TL	11.01.2001	0	0	1
53	YEH Ca	T2	N3	4				SGL	01.01.2016	1	0	1
50	YEH Ca	T2	N2	4				SGL	01.02.2001	0	1	1
52	İnsitu Ca	T1	N0	1				FLL	08.02.2001	0	1	0

Veritabanındaki alanların açıklamaları:

Yaş: Hastanın yaşı.

Patoloji: Ameliyat öncesinde tespit edilen patolojik tetkik sonucu. Yeh yassı epitel hücre anlamına gelmektedir.

Preop_T: Teşhis sırasında tespit edilen tümör büyüklüğü.

Preop_N: Teşhis sırasında tespit edilen node (lenf) büyüklüğü (Bkz. 5.2).

Preop_evre: Preop_T ve Preop_N değerlerinden hesaplanan kanser evresi.

Postop_T: Ameliyat sırasında görülen tümör büyüklüğü.

Postop_N: Ameliyat sırasında görülen node (lenf) büyüklüğü (Bkz. 5.2).

Postop_evre: Postop_T ve Postop_N değerlerinden hesaplanan kanser evresi.

Operasyon: Yapılan ameliyatın ismi. Veritabanında yedi çeşit ameliyat bulunmaktadır.

Operasyon_tarihi: Ameliyatın yapıldığı tarih.

Nüks: Ameliyat sonrası tümör nüks ederse bu alanın değeri 1 olur. Aksi durumda 0'dır.

Hayatta: Ameliyat sonrası hasta hayatta ise bu alanın değeri 1 olur. Aksi durumda 0'dır.

Patoloji_kodu: Ameliyat öncesinde tespit edilen patolojik tetkik sonucu "Yeh Ca" ise bu alanın değeri 1 olur. Diğer bütün sonuçlar için 0'dır.

5.4 K-means Algoritmasının Tercih Nedenleri

Bu çalışmada kullanılan k-means algoritması, aşağıdaki özellikleri nedeniyle tercih edilmiştir:

1. Küme sayısı olan k değeri parametre olarak algoritmaya dışardan verilmektedir. Verilerin kaç kümeye ayrılacağının net olarak bilinmediği durumlarda farklı değerler vererek sonuçları izlemek mümkün olmaktadır. Bu olanak uygulamanın analizini esnek hale getirmektedir.
2. Algoritmanın uygulanması kolaydır ve hızlı çalışmaktadır [40]. Veriler matristen okunarak algoritmaya verilmektedir. İterasyon sayısı, veri sayına oranla oldukça azdır ve bu nedenle algoritma hızlı çalışır.
3. Değişik dağılımlarda başarılı sonuçlar alınabilmektedir. Veriler birbirine çok yakın veya çok uzak değilse, kümeleme işlemleri başarıyla gerçekleştirilmektedir.
4. Kategorik verilerle çalışacak şekilde adapte edilebilmektedir. Algoritma rakamsal verilerle çalışmaktadır. Ancak kategorik veriler uygun rakamsal karşılıklarına çevrilerek algoritma tarafından işlenebilir.
5. Kümeleme sonuçları hem grafik olarak hem de yazı ve rakamlarla kolayca ifade edilebilmektedir. Algoritma sonucunda kümelenen veriler matrise yerleştirilmektedir. Buradan grafiğe kolayca aktarılabilmekte, ve her küme farklı renkle gösterilebilmektedir. Kümelerin eleman sayıları ve diğer verilere olan oranları kolayca hesaplanıp, yazılı olarak ifade edilebilmektedir.

Algoritmanın bazı dezavantajları da bulunmaktadır. Çok boyutlu verilerde kötü sonuçlar verebilir ve kümeleme yüksek iterasyon gerektirebilir. Ancak gırtlak kanseri ameliyatlarının verileriyle çalışılırken bu dezavantajlar ortaya çıkmamıştır. Çünkü veriler çok boyutlu değildir ve noktalar düzgün dağıldığından kümeleme yüksek iterasyon gerektirmemiştir.

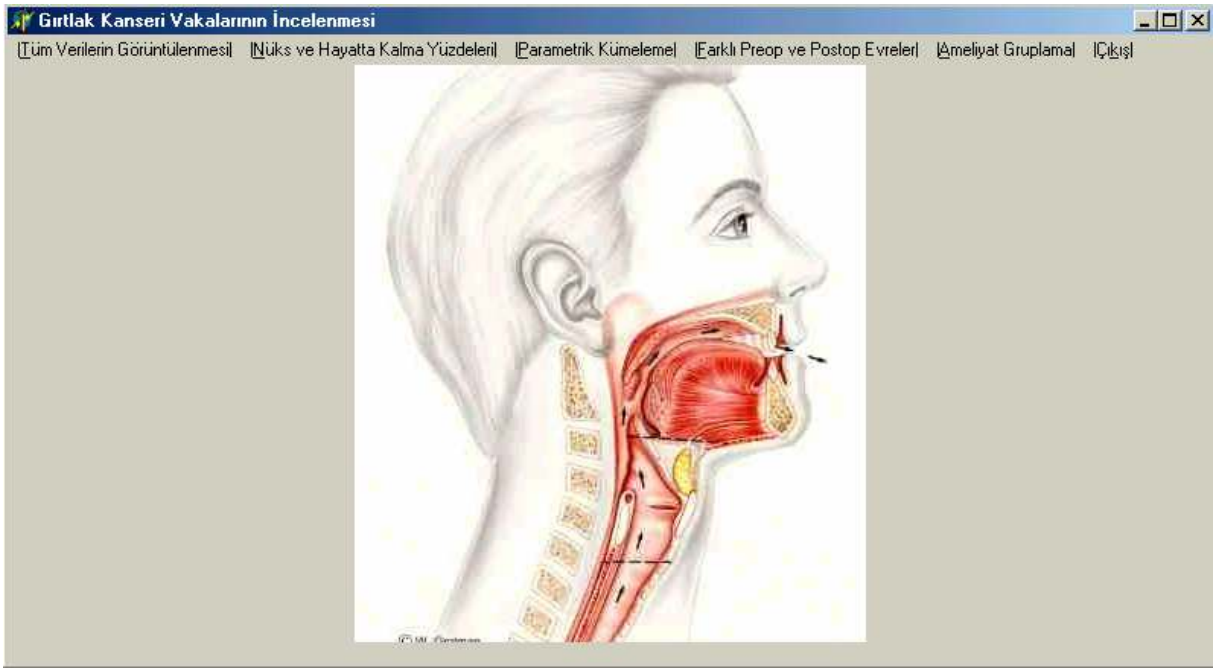
K-means algoritmasının Delphi ile yazılan kodu Ek-a'da yer almaktadır.

5.5 Geliştirilen Uygulama ile Verilerin Analizi

Bu çalışmada geliştirilen uygulama, Borland Delphi 6.0 kullanılarak Windows XP işletim sistemi üzerinde geliştirilmiştir. Veriler Access veritabanında tutulmuştur. K-means algoritması da Delphi ile kodlanmıştır.

Uygulama aracılığı ile veriler dört farklı açıdan analiz edilmiştir. Her bir analiz ayrı bir arayüz ile görüntülenmiştir. Arayüz ekranlarına şekil 5.1'de görülen bir anamenü üzerinden erişilmektedir. Oluşturulan arayüzler şöyledir:

1) Nüks ve Hayatta KalmaYüzdeleri: Gelecek vakaların tahminlerinde kullanılmak amacıyla, geçmiş vakaların tümör nüks etme yüzdeleri ve hayatta kalma yüzdeleri incelenir. Sadece bu arayüzde k-means algoritması kullanılmamıştır.
2) Parametrik Kümeleme: Veri tablosunun seçilen iki alanı arasındaki etkileşim izlenir.
3) Farklı Preop ve Postop Evreler: Doğru öngörülen ve öngörülemeyen ameliyat öncesi evreler incelenerek ameliyat öncesi tahmin başarısı değerlendirilir.
4) Ameliyat Gruplama: Başarılı ameliyatlar izlenerek, gelecek ameliyat tercihlerine karar verilir.



Şekil 5.1: Geliştirilen programın anamenüsü

Analiz yapılan arayüzlere ilave olarak, bütün veritabanının filtre edilmeden görüntülenmesi için Tüm Verilerin Görüntülenmesi seçeneği altında bir liste ekranı hazırlanmıştır.

5.6 Uygulama Arayüzleri

Analiz yapılan arayüzler benzer görüntülerde tasarlanmıştır. Ortak özellikler şöyledir; arayüz ekranının üst kısmında analizi yapılan veriler listelenmektedir. Parametre girişi yapılmayan arayüzlerde, seçenek çalıştırıldığında, veriler hesapanarak görntülenmektedir.

Kümelenmiş veriler ve içlerindeki yoğunlaşmalar orta kısımda grafik üzerinde gösterilmiştir. Arayüzlerde k-means algoritması ile oluşturulacak küme sayısı, kullanıcı tarafından 2 ile 9 arasında bir değer ile belirlenebilmektedir. Sınama sonuçlarına göre, 9 kümeden daha fazlası verimli olmamaktadır. Kullanıcı "K-means çalıştır" tuşunu tıklayarak kümeleme sonucunu grafik üzerinde görüntüleyebilmektedir. Her bir küme ayrı bir renk ile gösterilerek birbirinden ayrıştırılır. Bu şekilde kümelenmiş veriler, kolay analiz olanağı sağlamaktadır.

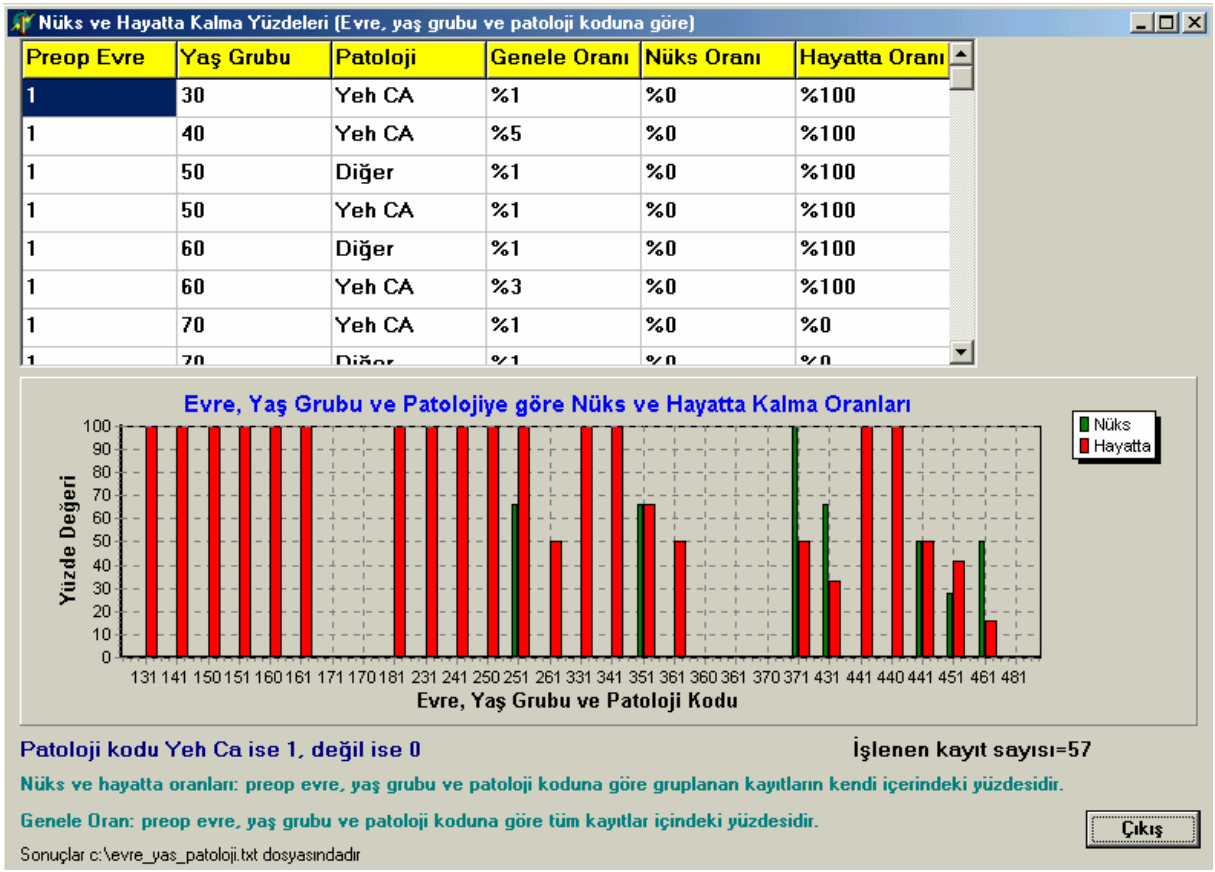
Kümelenmiş sonuçların kısa açıklaması arayüz ekranlarının altındaki bir pencereden görüntülenir. Açıklama, her bir kümedeki kayıt sayısını ve bu sayının işlenen tüm kayıtlara oranını içerir. Her bir açıklama satırı, grafikteki kümesi ile aynı renktedir. Bu kısımda ayrıca kümeleme sonucunun yazılı ifadeleri ile arayüze ilişkin mesajlar ve açıklamalar bulunmaktadır.

Her arayüzün çalıştırılması sırasında hesaplanan bilgiler ayrıca bir metin bir dosyasına yazdırılmaktadır. Bu dosyada ekranda görüntülenen rakamsal bilgiler ve bunlarla ilişkili diğer ayrıntı bilgiler yer almaktadır. Dosyalar grafik içermemektedir. Uygulamanın arayüzleri aşağıda ayrıntılarıyla açıklanmıştır.

5.6.1 Nüks ve hayatta kalma yüzdeleri

Bu arayüzde diğer üç arayüz ekranından farklı olarak veri kümeleme yerine sınıflama yapılmıştır. Analizi yapılacak kayıtların algoritmaya uygulanamaması nedeniyle, burada k-means algoritması kullanılmamıştır.

Bu arayüzün amacı, mevcut ve gelecek vakalar için, ameliyat sonrasında tümörün nüks etme olasılığının ve hastanın hayatta kalma olasılığının tahmin edilmesine destek sağlamaktır. Bunun için geçmiş vaka verileri sınıflandırılarak, tümör nüks etme yüzdeleri ve hayatta kalma yüzdeleri görüntülenir. Veriler, hasta yaş grubu, ameliyat öncesi evresi ve patolojik bilgiye göre sınıflanmıştır. Yüzdeler her sınıf için ayrı ayrı hesaplanmıştır.



Şekil 5.2: Nüks ve hayatta kalma yüzdeleri arayüzü

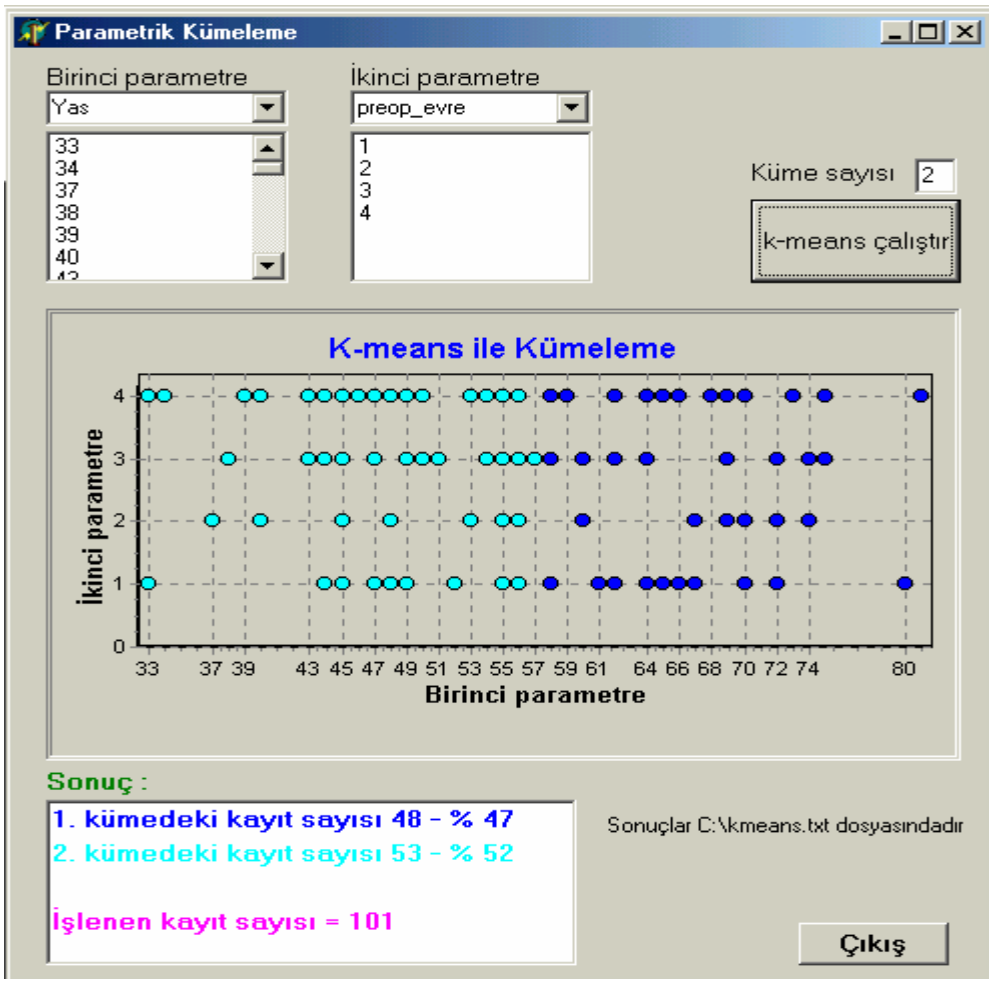
Arayüz görüntülendiğinde hesaplanan sonuçlar listelenir. Her bir sınıftaki kayıt sayısının bütün kayıtlara oranı hesaplanarak "Genele Oranı" adı altında ayrı bir sütunda listelenmiştir. Grafikte çubuk (bar) seri tipi kullanılmıştır. Şekil 5.2'de görüldüğü gibi her bir çubuk bir sınıfı temsil etmektedir. Grafikte görüntülenen yeşil renkli seri, her bir sınıfın tümör nüks yüzdesini, kırmızı renkli seri ise hayatta kalma yüzdesini göstermektedir.

Sınıflar hasta yaş grubu, ameliyat öncesi evresi ve patolojik bilgilerin birleştirilmesinden oluşturulmuştur. Yaş grubunda onlar basamağı aynı olan yaşlar gruplanmıştır. Örneğin 50-59 arası 50 grubu, 60-69 arası 60 grubu gibi. Birleştirilen alanların bilgileri sayısal karşılıklarına dönüştürülerek, her bir sınıf üç basamaklı bir sayı ile temsil edilmiştir. Sayının yüzler basamağını evre kodu, onlar basamağını yaş grubu bilgisinin onlar basamağı, birler basamağını değeri 0 veya 1 olan patoloji kodu oluşturmaktadır. Program kodunda sayının hesaplanmasında izlenen yol şöyledir: (evre kodu)*100+(yaş grubunun onlar basamağı)*10+ (patoloji kodu). Örneğin evre kodu 1, yaş grubu 70 ve patolojik tetkik sonucu "Yeh Ca" (Bkz. 5.3) ise bu sınıfın kodu 171'dir. Evre kodu 2, yaş grubu 60 ve patolojik tetkik sonucu "Diğer" ise bu sınıfın kodu 260'dır.

Örneğin Şekil 5.2'de görülen ilk satırda, birinci evrede olan ve patolojik tetkik sonucu "YehCa" (yassı epitel hücre, bkz. 5.2) olan 30 yaş grubundaki vakalar yer almaktadır. Bu gruptaki vakalar sayıca bütün kayıtların yüzde birini oluşturmaktadır. Vakaların hiç birinde tümör nüksü kaydedilmemiş ve vakaların tamamı hayattadır. Bu grup 131 kodu ile grafikte yer almıştır. Nüks oranı 0 olduğundan yeşil seri görüntülenmemektedir. Hayatta oranı 100, kırmızı seri ile gösterilmiştir. Yedinci satırda, birinci evrede olan ve patolojik tetkik sonucu "YehCa" olan 70 yaş grubundaki vakaların nüks ve hayatta bilgileri veritabanında yer almadığı için oranları 0'dır.

5.6.2 Parametrik kümeleme

Bu arayüzde geriye dönük inceleme kolaylığı ve ileriye dönük karar verme desteği sağlanması amaçlanmıştır. Doktor değişken parametreler kullanarak değerlendirme yapabilir, geçmiş verileri analiz edebilir, mevcut ve gelecek vakalar için tahminde bulunabilir. Veritabanından iki alan seçilerek, bunların değerleri arasındaki etkileşim izlenmektedir. Örneğin yaş ile tümör boyutu veya yaş ile uygulanmış ameliyatlar gibi. Seçilen veriler k-means algoritması ile kümelenerek, veri içindeki yoğunlaşmalar kullanıcıya görüntülenebilmektedir. Geliştirilen uygulama içinde, kullanıcının alanlar seçerek işlem yapabildiği tek arayüzdür. Eğer kullanıcı birbiriyle ilişkisiz alanlar seçerse, analiz sonucu anlamsız olabilir.



Şekil 5.3: Parametrik kümeleme arayüzü

Arayüz görüntülendiğinde şekil 5.3'de görüldüğü gibi, veritabanındaki bütün alanların isimleri üstte yer alan iki açılan kutu (combo box) içinde listelenir. Kullanıcı, alan isimlerinin üstünü tıklayarak herhangi iki tanesini seçer. Seçim sonrası alanların değerleri, her bir değerden bir tane olacak şekilde liste kutusunda (list box) listelenir. Seçilen alanların grafik üzerinde kümelenmiş görüntüsünün oluşturulması için, kullanıcı küme sayısını girerek "K-means çalıştır" tuşunu tıklar. Grafik üzerindeki her bir seri seçilen bir alanı temsil eder.

Şekil 5.3'deki örnek görüntüde, yaş ve ameliyat öncesi evre alanları seçilmiş, ilgili veriler k-means algoritması ile iki kümeye ayrılmıştır. Bu seçimle, ameliyat öncesi öngörülen evrelerin yaşlarla ilişkisi incelenmektedir. Kayıt sayıları birbirine yakın iki kümeden birincisi 33 ile 57 yaş arası, ikincincisi ise 57 ile 83 yaş arasındaki vakaları içermektedir. Birinci kümede, 43 yaş sonrası dördüncü evredeki vakaların çokluğu dikkat çekmektedir. Dördüncü evreyi, yoğunluk açısından üçüncü ve birinci evreler izlemektedir. İkinci kümede ise 57 yaş ile 70 yaş arası dördünce evrede, 61 ve 67 yaş arasında birinci evrede yoğunluk izlenmektedir.

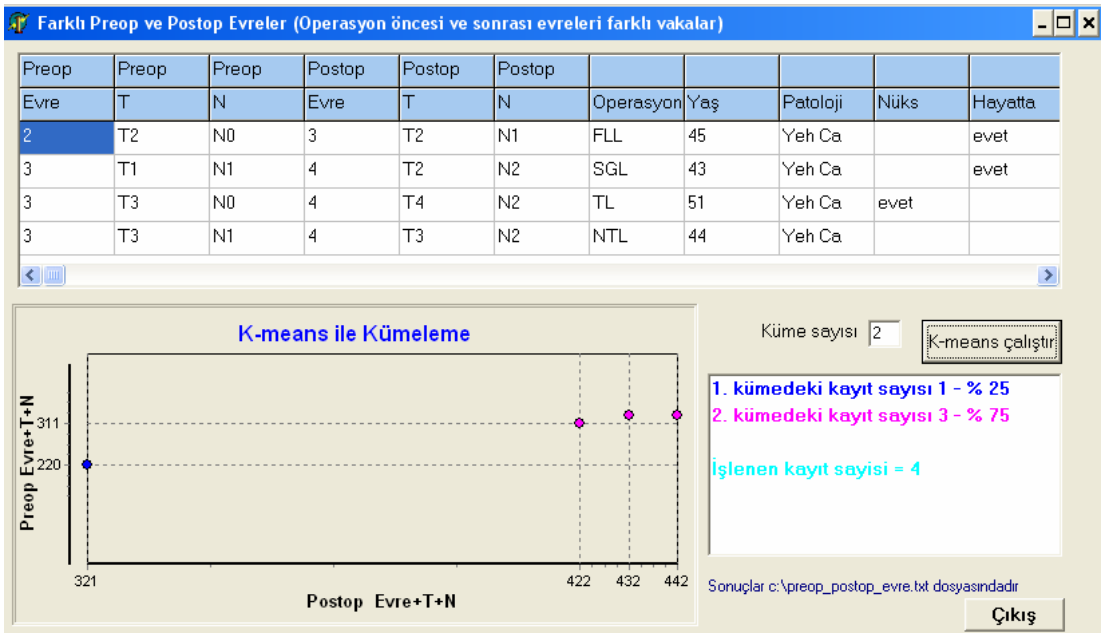
Program kodunda, ameliyat ismi ve patoloji kodu gibi bütün karakter alanlar k-means algoritmasıyla işlenebilmek için sayısal karşılıklarına dönüştürülmüştür.

5.6.3 Farklı preop ve postop evreler

Bu arayüzde doğru öngörülen ve öngörülemeyen ameliyat öncesi evrelerin görüntülenerek incelenmesi ve bu şekilde ameliyat öncesi tahmin başarısının değerlendirilmesi amaçlanmıştır. Kanser ameliyatlarında, nadiren de olsa ameliyat öncesinde tahmin edilen evre, ameliyat sırasında görülen evreden farklı olmaktadır. Bu da ameliyat öncesi evreye göre yapılan hazırlığın, ameliyat sırasında değiştirilmesi ve yeni kararlar alınması anlamına gelmektedir. Gerçek hayatta az rastlanır bir durum olduğundan, veritabanında bu inceleme için listelenen kayıt sayısı azdır. Şekil 5.4'deki örnekte sadece 4 kayıt görüntülenmektedir.

Burada evre değerleri, ameliyat öncesinde ve sonrasında biribirinden farklı kayıtlar listelenmektedir. Koşulu sağlayan her bir vaka için listelenen bilgiler şunlardır:

Ameliyat öncesi ve sonrasına ait evre değeri, tümör ve node (Bkz. 5.2) kodları, uygulanan ameliyatın kısa adı, hastanın yaşı, patoloji kodu, nüks ve hayatta bilgileri. Grafikte ameliyat öncesi ve sonrasına ait evre, tümör ve node değerleri karşılaştırılmaktadır.



Şekil 5.4: Farklı preop ve postop evreler arayüzü

Program kodunda, evre, tümör ve node değerlerinin k-means algoritmasında kullanılabilmesi için alanlar birleştirilerek tek bir sayı haline getirilmiştir. Birleştirmede izlenen yol şöyledir: (evre kodu)*100+(tümör değeri)*10+ (node değeri). Ameliyat öncesi ve sonrası için bu şekilde oluşturulan sayılar k-means algoritmasının her bir boyutunu oluşturur. Şekil 5.4'de görülen örnekte, ilk kaydın preop evresi 2, tümör değeri 2 ve node değeri 0'dır. Bu değerler grafikte 220 sayısına karşılık gelmektedir. Aynı şekilde postop evresi 3, tümör değeri 2 ve node değeri 1'dir. Bu da grafikte 321 sayısına karşılık gelmektedir. Şekil 5.4'de listelenen 4 kayıt iki kümeye ayrılmıştır. Postop evresi 4 olan kayıtlarla preop evresi 3 olan kayıtlar üç elemanlı büyük kümeyi oluşturmuşlardır. Grafikten görüldüğü gibi, ameliyat öncesinde üçüncü evrede olduğu öngörülen vakalar, ameliyat sırasında dördüncü evrede bulunmuştur. Öngörü farkı, en çok üçüncü evrede, patolojisi "yehca" (Bkz. 5.2) olan 40-50 yaş arası vakalarda ortaya çıkmıştır.

5.6.4 Ameliyat gruplama

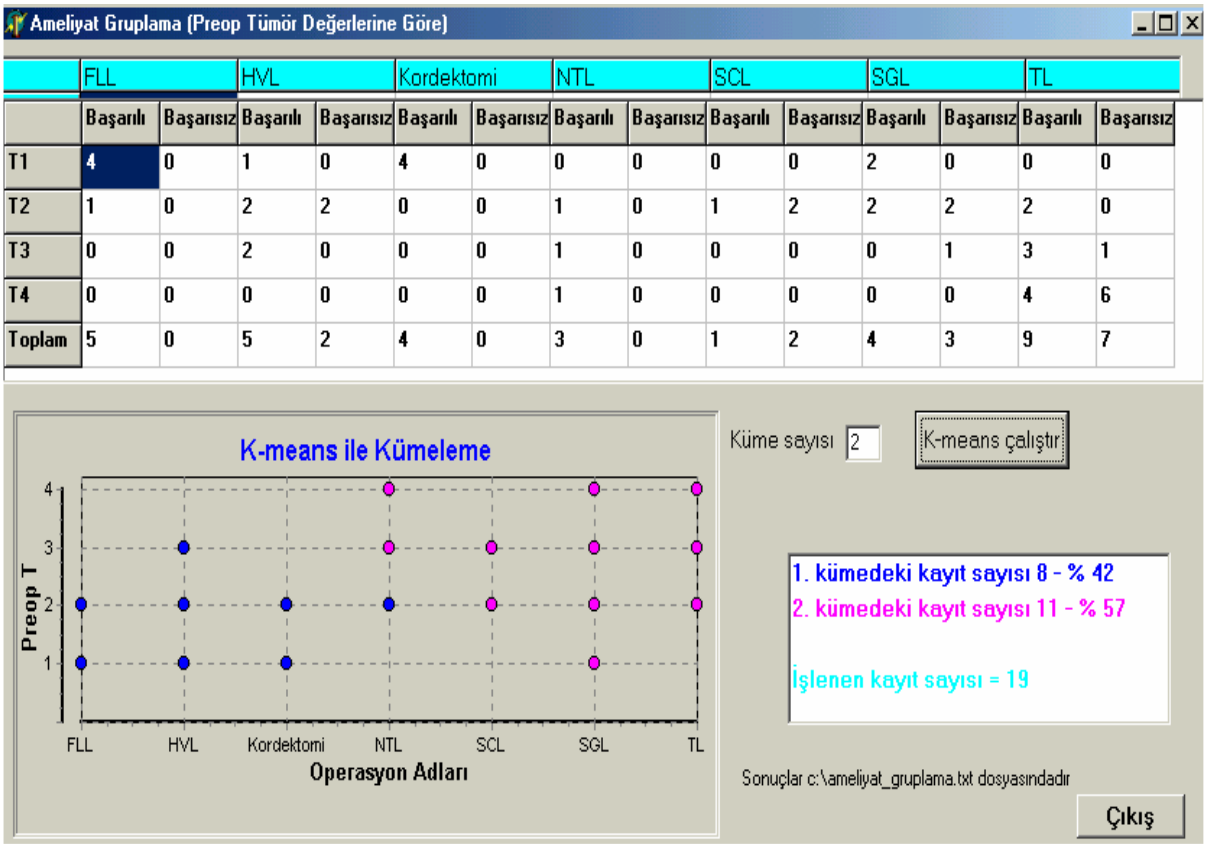
Bu arayüzde başarılı ameliyatların izlenerek, gelecek ameliyat tercihlerine destek sağlanması amaçlanmıştır. Burada gırtlak kanseri ameliyatları, vakalara göre kümelenmiştir. Vakalar ameliyat öncesi tümör değerleri ile temsil edilmektedir. Kümeleme sırasında aynı tümör değerine sahip vakaların sayıları belirlenmiştir.

Aynı tümör değerine sahip her vaka için başarılı ve başarısız ameliyatların sayıları hesaplanmıştır. Hesaplama yöntemi şöyledir; eğer ameliyat sonrasında tümör nüks etmiş ise ameliyat başarısız kabul edilir, eğer tümör nüks etmemiş ve hasta hayatta ise ameliyat başarılı kabul edilir. Böylece arayüzde, gerçekleştirilen ameliyatlar görüntülenerek, hangi tümör değerlerine hangi ameliyatların ne çoklukta yapıldığı ve başarılı/ başarısız sayıları izlenir. Başarılı ve başarısız ameliyatların sayıları grafiğe yansıtılmamıştır.

Arayüz vasıtasıyla kümelenmiş ameliyatlar listelenir. Birçok ameliyatın ismi uzun olduğu için isimler yerine kısaltmalar kullanılmıştır. Şekil 5.5'de görülen tabloda ameliyat öncesi tümör değerlerine karşılık olarak, başarılı ve başarısızlık hesaplaması her bir ameliyat için ayrı ayrı listelenmektedir. Şekildeki örnekte FLL kodlu ameliyat, T1 için 4 kez, T2 için 1 kez başarıyla yapılmıştır. Bu ameliyatta hiç başarısız vaka kaydedilmemiştir. Son satırda bütün ameliyatların başarılı ve başarısız sayıları toplanmıştır. Örnekte, FLL kodlu ameliyatın başarı sayısı 5, başarısız sayısı 0'dır. Bu tabloda, veritabanında nüks ve hayatta alanları dolu olan vakalar hesaplamaya dahil edilmiştir. Her kayıt için bu alanlar dolu olmadığından, burada listelenen kayıt sayısı, veritabanındaki toplam kayıt sayısından daha azdır.

Şekil 5.5'deki grafikte ameliyatlar ile ameliyat öncesi tümör değerleri karşılaştırılmaktadır. Burada tümör tiplerine uygulanan ameliyat çeşitleri kümelenmiştir. Kümeleme için veritabanından okunan vakaların nüks ve hayatta alanları dikkate alınmamıştır. Grafikte gösterilmediğinden tümör tiplerine uygulanan ameliyat sayıları hesaplanmamıştır. Bu nedenle tabloda listelenen ile grafikte ele alınan kayıt sayıları farklıdır. Tabloda 45 kayıt için listeleme yapılırken, grafikte 19 kayıt kümelenmiştir. Şekildeki örnekte, ameliyatlar iki kümeye ayrılmıştır. Sayıca az

olan ilk kümede, daha çok 1 ve 2. derece tümörlere uygulanan üç ameliyat yer almıştır. Büyük kümede ise, 2. derecenin üstündeki tümörlere uygulanan ameliyatlar bulunmaktadır. 2. derece tümörlere bütün ameliyat çeşitlerinin uygulandığı, SGL kodlu ameliyatın bütün tümör türlerine uygulanabildiği görülmektedir. Diğer taraftan 2., 3. ve 4. derecelere uygulanan NTL ve TL kodlu ameliyatlar, başarı rakamlarına göre incelendiğinde, TL'nin daha riskli bir teknik olduğu göze çarpmaktadır. Buna rağmen TL, sayıca NTL'den fazla gerçekleştirilmiştir.



Şekil 5.5: Ameliyat gruplama arayüzü

Program kodunda ameliyatlar k-means algoritmasıyla işlenebilmek için sayısal karşılıklarına dönüştürülmüştür. Bunun için ameliyat isimlerine alfabetik sırada birden yediye kadar sıra numarası verilmiştir.

5.6.5 Tüm Verilerin Görüntülenmesi

Bu arayüz gırtlak kanseri ameliyatlarına ilişkin tüm verilerin görüntülenmesi amacıyla hazırlanmıştır. Verilerin yer aldığı Access veritabanındaki bütün alanlar listelenmiştir. Şekil 5.6'da görüldüğü gibi veriler üzerinde herhangi bir işlem yapılmamıştır.

Yas	Patoloji	preop_T	preop_N	preop_evre	postop_T	postop_N	postop_evre	Operasyon	Operasyon_tarihi	nuks	hayatta	patoloji_kodu
65	YEH Ca	T1	N0	1				FLL	03.01.1996	0	0	1
56	YEH Ca	T4	N2	4				TL	13.12.1995	1	0	1
67	YEH Ca	T2	N0	2				SGL	28.03.1996	0	0	1
45	YEH Ca	T2	N0	2	T2	N1	3	FLL	14.02.1996	0	1	1
58	YEH Ca	T4	N2	4				NTL	20.05.1997	0	1	1
49	YEH Ca	T4	N2	4				TL	03.06.1997	0	1	1
45	YEH Ca	T3	N0	3				TL	03.03.1997	0	1	1
33	epidermoid Ca	T4	N2	4				TL	02.09.1997	0	0	0
48	YEH Ca	T1	N0	1				SGL	17.09.1997	0	1	1
67	YEH Ca	T1	N0	1				Kordektomi	03.11.1997	0	0	1
74	YEH Ca	T2	N0	2				HVL	19.01.1998	0	0	1
62	Ağır displazi	T1	N0	1				Kordektomi	11.03.1997	0	1	0
44	YEH Ca	T3	N1	3	T3	N2	4	NTL	23.02.1998	0	0	1
60	YEH Ca	T3	N0	3				TL	18.12.1997	0	1	1
61	Ağır displazi	T1	N0	1				Kordektomi	10.05.1996	0	0	0

Şekil 5.6: Tüm verilerin görüntülendiği arayüz

6. SONUÇLAR VE ÖNERİLER

Bu bölümde, tez çalışması hakkında bilgi verilerek ve gerçekleştirilen uygulama ile elde edilen sonuçlar özetlenmiştir. Tıp alanında veri madenciliği uygulamalarının ilerlemesinin önündeki engeller anlatılmış ve bölümün sonunda çalışmanın geliştirilebilmesi için ileride yapılabilecek öneriler sıralanmıştır.

Günümüzde başta iş dünyası olmak üzere birçok farklı alanda kullanılan veri madenciliği MIT, (Massachusets Institue of Technology) tarafından 2001 yılında yayınlanan bildirgeye göre dünyayı değiştirecek 10 teknoloji arasında gösterilmiştir. Gelecekte daha çok önem kazanacak olan veri madenciliği üzerinde yapılan çalışmalara her geçen gün yenileri eklenmektedir. Veri madenciliğinde yeni gelişen teknolojilerin birçoğu henüz tıp alanında kullanılan yazılımlara dahil edilmemiştir. Hastalıklara teşhis koyan bir doktor, öncelikle deneyimlerden benzer vakaları tanımlar ve ardından bu vakaların bilgilerini eldeki probleme uygular. Bilinen vaka kayıtlarının tutulduğu veritabanı, sınıflandırılmış kayıtlar içinden yeni vakaya benzeyenleri bulmak için taranır. Mevcut hasta için en etkili tedavi, muhtemelen benzer hastaların sonuçlarından elde edilen bilgilerle yapılan tedavidir. Bu tezde tıp alanında geçmiş kayıtları kullanmanın önemi dikkate alınarak gırtlak kanseri ameliyat verileri için bir analiz aracı geliştirilmiştir.

Çalışmanın amacı, veri madenciliğinde bir kümeleme tekniği olan k-means algoritmasını incelemek ve bu algoritmayı kullanarak gırtlak kanseri ameliyat verileri üzerine geliştirilen bir yazılım aracılığıyla, verilerin analizini yapmaktır.. Uygulamanın tıp doktorlarının kullanımına uygun şekilde verileri çeşitli açılardan analiz etmesi hedeflenmiştir. Çalışmada kullanılan gırtlak kanseri ameliyat verileri, Kocaeli Üniversitesi Tıp Fakültesi Hastanesi, Kulak, Burun ve Boğaz Bölümünden alınmıştır. Veriler dört farklı açıdan analiz edilmiş ve her bir analiz ayrı bir ekranda görüntülenmiştir.

Bu çalışmada kullanılan k-means algoritması aracılığı ile veriler kümelenmiştir. Küme sayısı, kullanıcı tarafından 2 ile 9 arasında bir değer olarak belirlenebilmektedir. Sınama sonuçlarına göre, 9 kümeden daha fazlası verimli olmamaktadır. Kümeleme sonucu grafik üzerinde gösterilerek veri içindeki yoğunlaşmaların ortaya çıkması sağlanmıştır. K-means algoritması aşağıdaki özellikleri nedeniyle tercih edilmiştir:

1. Küme sayısının okunan bir parametre olması analizi esnek hale getirmektedir.
2. Algoritmanın uygulanması kolaydır ve hızlı çalışmaktadır.
3. Değişik dağılımlarda başarılı sonuçlar alınabilmektedir.
4. Kategorik verilerle çalışacak şekilde adapte edilebilmektedir.
5. Kümeleme sonuçları hem grafik olarak hem de yazı ve rakamlarla kolayca ifade edilebilmektedir.

Algoritmanın bazı dezavatajları da bulunmaktadır. Çok boyutlu verilerde kötü sonuçlar verebilir ve kümeleme yüksek iterasyon gerektirebilir. Ancak gırtlak kanseri ameliyatlarının verileriyle çalışılırken bu dezavatajlar ortaya çıkmamıştır. Çünkü veriler çok boyutlu değildir ve noktalar düzgün dağıldığından kümeleme yüksek iterasyon gerektirmemiştir.

Yazılım aracılığı ile yapılan analizler sonucunda aşağıdaki yararlar elde edilebilir;

1. Geçmiş verileri analiz ederken değişken parametreler kullanılarak değerlendirme yapılabilir, vakalar için tahminde bulunulabilir,
2. Mevcut ve gelecek vakalar için ameliyat sonrasında tümörün nüks etme olasılığı ve hastanın hayatta kalma olasılığı değerlendirilebilir,
3. Doğru öngörülen ve öngörülemeyen ameliyat öncesi evreler görüntülenerek incelenebilir ve bu şekilde ameliyat öncesi tahmin başarısı değerlendirilebilir,
4. Başarılı ameliyat bilgileri izlenerek, gelecek ameliyat tercihlerinde fikir alınabilir.

Geliştirilen yazılım, tıp doktorlarının geçmiş kayıtları analiz ederek, ileriye dönük tahminde bulunabilmelerini kolaylaştırmaktadır. Karar almada yardımcı olabilecek bir araçtır. Ayrıca araştırma, denetim ve eğitim etkinliklerinde de kullanılabilir. Mevcut durumda, hastanede geçmiş kayıtları analiz etmek için hasta dosyaları tek tek taranarak araştırma yapılmaktadır.

Tez için yapılan literatür taramasında, gırtlak kanseri ameliyat verileri üzerinde, k-means algoritması kullanarak yapılan başka bir veri madenciliği çalışmasına rastlanmamıştır. Veri madenciliği çalışmalarında genellikle veriler SPSS ve MATLAB gibi paket programlar aracılığı ile analiz edilmektedir. Ancak paket programlar çeşitli kısıtlar içermektedir. Kullanıcı açısından bu çalışmada geliştirilen yazılımın öğrenme süresi çok kısadır ve kullanılması kolaydır.

Tıp alanında veri madenciliği uygulamalarının ilerlemesinin önünde bazı engeller bulunmaktadır. Öncelikle hasta teşhis ve tedavi bilgileri çoğunlukla klasik hastane bilgi sistemlerinde tutulmadığından, çalışmalarda kullanılacak verilerin bulunmasında zorluk yaşanmaktadır. İstenen verilere bulmak zor olduğundan, bireysel çabalarla ve belirli konular için toplanmış verilere ulaşmak gerekmektedir. Verilerin elde edilmesi, çalışmanın başlaması için her durumda yeterli olmamaktadır. Tıp alanındaki verilerde belli bir standardın olmayışı verilerin işlenmesini zorlaştırmaktadır. Çoğu kez, çalışmanın yönlendirilmesi için konuyla ilgili bir tıp doktorunun bulunması ve destek alınması gerekebilir. İlgili tıp doktorunun bilgisayar uygulamaları konusunda bilgi sahibi olmaması çalışmayı olumsuz yönde etkilemektedir. Bu zorlukların en aza indirilmesi, tıp alanında veri madenciliği uygulamalarının artmasını sağlayacaktır.

Geliştirilen yazılımın gerçek veritabanı üzerinde kullanılması, uygulamanın etkinliğini görmeyi sağlamıştır. Yazılıma eklemeler yaparak, Kocaeli Üniversitesi Tıp Fakültesi Hastanesi, Kulak, Burun ve Boğaz Bölümünde kullanılması planlanmaktadır.

Tıbbi veritabanları üzerinde yazılım uygulaması geliştiren çalışmaların azlığı dikkat çekmektedir. Bu tez çalışması, yakın gelecekte tıp alanında geliştirilecek ve veri madenciliği algoritmalarını içerecek yazılımlar için bilgilendirici bir kaynak olabilir. Bu çalışma ile, veri madenciliğindeki kümeleme algoritmalarının çeşitli tıbbi verilerin değerlendirilmesi ve analizine katkısı gözlenebilir. Bu tip yazılımların başka klinikler için de geliştirilmesinin tıbbi çalışmalarda yararlı olabileceği düşünülmektedir. Analizlerde k-means dışında başka teknikler kullanılarak, sonuçlar karşılaştırılabilir.

KAYNAKLAR

[1] Berry, M.; Linoff, D., "Data Mining Techniques", ***Wiley Publishing Inc.,***2004

[2] www.eecs.mit.edu.tr **(ziyaret tarihi 18 mayıs 2005)**

[3] Piatetsky-Shapiro G., Frawley W. J., "Knowledge Discovery in Databases", ***AAAI/MIT Press***, 1991.

[4] Baykal N., Veri Tabanı ve Veri Madenciliği konulu sunum, ***Tıp Bilişimi Güz Okulu,*** Ekim, 2003, www.turkmia.org

[5]. Aslandogan, A., Mahajani, G., Taylor, S., "Evidence Combination in Medical Data Mining," ***International Conference on Information Technology: Coding and Computing (ITCC'04),*** Volume 2, itcc, p. 465, (2004).

[6] Breault, J. L., "Data Mining Diabetic Databases: Are Rough Sets a Useful Addition", ***Computing Science and Statistics***, Volume l, 34, (2001).

[7]. Duru, N., "An Application of Apriori Algorithm on a Diabetic Database", ***Springer-Verlag*** Berlin Heidelberg, LNAI 3681, pp. 398.404, (2005).

[8]. Xu D., Olman V., Wang L., Xu Y., "a computer program for efficiently mining gene expression data", ***Nucleic Acids Research [NLM - MEDLINE].,*** Vol. 31, Iss. 19; p. 5582, (Oct 1 2003).

[9] Fakih S., Das T., "A methodology for learning efficient approaches to medical diagnosis", ***IEEE***, 10(2):220-8, (2006 Apr).

[10] MacQueen, J., "Some methods for classification and analysis of multi-variate observations". In: Proc. of the Fifth Berkeley Symp. ***on Math., Statistics and Probability***, LeCam, L.M., and Neyman, J., (eds.), Berkeley: U. California Press, 281. 1967,

[11] Evans, S., Lloyd, J., Stoddard, G., Nekeber, J., Samone, M., "Risk Factors For Adverse Drug Events", ***The Annals of Pharmacotherapy,*** Vol. 39, No. 7, pp. 1161-1168, (2005).

[12] Vickers E., Boocock H., Harris R., Bradshow J., "Analysis of the acute postoperative pain experience following oral surgery", ***Australian Dental Journal,*** 51(1):69-77, (2006 Mar).

[13] Morana H., Camara F., Arboleda-Florez J., "Cluster analysis of a forensic population with antisocial personality disorder", ***Forensic Science International***, (2006, jan 23).

[14] Rapeli C., Botega N., "Clinical profiles of serious suicide attempters consecutively admitted to a university-based hospital", ***Revista Brasilileira de Psiquiatria***, 27(4):285-9. Epub (2005 Dec 12).

[15] Shai, R., Shi, T., Kremen, T., Horvath, S., Liau, l., Cloughesy, T., Mischel P., Nelsın, S., "Gene Expression Profiling Identifies Molecular Subtypes Of Gliomas To Cluster Molecular Subtypes", ***Oncogene. Basingstoke***, Vol.22,Iss.31;pg.4918 (2003).

[16] Sultan, M., Wigle, D., Cumbaa,C., Maziarz,M., "Binary Tree-Structured Vector Quantization Approach", ***Bioinformatics***, Oxford:.Vol.18, Iss. 1; pg. 111. (Jul,2002)

[17] Ushizawa, K., Herath C., Kaneyama K., "cDNA microarray analysis of bovine embryo gene expression profiles during the pre-implantation period", ***Biology and Endocrinology***, 2:77, doi:10.1186/1477-7827-2-77 (2004)

[18] Wu C., Zhao W., Lin B., Ginsberg M., "Semi-automated image processing system for micro- to macro-scale analysis of immunohistopathology: application to ischemic brain tissue.**"**, ***Computer Methods and Programs in Biomedicine.***, 78(1):75-86. (Apr 2005)

[19] Akpınar, H.: "Veri Tabanlarında Bilgi Keşfi ve Veri madenciliği", ***İ.Ü. İşletme Fakültesi Dergisi*****,** Sayı :1, 1 – 22. (Nisan 2000)

[20] Fayyad, U.M.; Piatesky-Shapiro, G.; Smyth, P.; Uthurusamy, R., "Advances in data mining and Knowledge Discovery", ***AAAI Pres***, USA (1994).

[21] Hinneburg, Alexander; Keim, Daniel A..: "Clustering Techniques for Large Data Sets From the Past to the Future", Powerpoint Slayt, www.ece.northwestern.edu/~harsha/Clustering/keim_slides.pdf

[22] http://en.wikipedia.org **(ziyaret tarihi 18 mayıs 2005)**

[23] Han, J.; Kamber, M., "Data Mining Concepts and Techniques", ***Morgan Kaufmann Publishers Inc.,*** (Ağustos 2001)

[24] Berkhin, Pavel.: "Survey of Clustering Data Mining Techniques", ***Accrue Software Inc.***, San Jose, California, USA (2002).

[25] Kaufman, L.; Rosseeauw, P.J., "Finding Groups in Data: An Introduction to Cluster Analysis.", ***John Wiley and Sons Inc.***, New York, USA (1990)

[26] Ester, M.; Kriegel, H. P.; Sander, J.; Xu, X., "A density based algorithm for discovering clusters in large spatial databases." ***Int. Conference of Knowledge Discovery and Data Mining (KDD'96)***, Portland, USA 226-231. (1996)

[27] Ankerst, M.; Breunig, M.; Kriegel, H.P.; Sander, J., "OPTICS: Ordering points to identify the clustering structure." ***ACM SIGMOD Int. Conf. Management of Data (SIGMOD'99)***, Philadelphia, Pennsylvania USA 49-60. (Haziran 1999).

[28] Hinneburg A.; Keim D. A.: "An Efficient Approach to Clustering in Large Multimedia Databases with Noise", ***Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD'98),*** New York, USA, 58-65. (Ağustos 1998)

[29] Wang, W.; Yang, J.; Muntz, R.: "STING: A statistical information grid approach to spatial data mining.", ***Int. Conference of Very Large Data Bases (VLDB'99).,*** Atina, Yunanistan. 186-195. (Ağustos 1997)

[30] Sheikholeslami, G.; Chatterjee, S.; Zhang, A., "WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases." ***Proc. 24th Int. Conf. on Very Large Data Bases***, New York, USA, 428-439. (Ağustos 1998)

[31] Vidakovic, B.;Mueller, P.: "wavelets for kıds: A Tutorial Introduction By Brani Vidakovic and Peter Mueller", ***Institute of Statistics and Decision Sciences,*** Duke University, Durham, (1991).

[32] Aggrawal, R.; Gehrke J.; Gunopulos, D.; Raghavan, P., " Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications" , ***Proc. ACM SIGMOD Int. Conf. on Managment of Data,*** Seattle, USA 94-105. (Haziran 1998)

[33] FISHER, D.: "Knowledge acquisition via incremental conceptual clustering", ***In Proc. 1987 AAAI Conf.,*** Seattle, USA 139-172. (Temmuz 1987)

[34] Rumelhart, D.E.; Zipser, D.: "Feature discovery by competitive learning." ***Cognitive Science,*** Vol: 9, 75-112. (1985)

[35] Böhm, C., "Powerful Database Support For High Performance Data Mining.", Doktora Tezi, ***Ludwig-Maximilian Üniversitesi Enformatik ve Matematik Bölümü***. Münih,Almanya, (2001).

[36] Sturn A,, "Cluster Analysis for Large Scale Gene Expression Studies", Master Tezi, ***The Institute for Genomic Research (TIGR),*** USA, (2000).

[37] http://people.revoledu.com/kardi/tutorial/kMean **(ziyaret tarihi 18 mayıs 2005)**

[38] Alsabti, K.; Ranka, S.; Singh, V., “An efficient K-means Clustering algorithm.” ***Syracusa University, University of Florida, Information Technology Laboratory of Hitachi America Ltd*** . Florida, USA, (2000).

[39] Kantardzic M., “Data Mining, Concepts, Models, Methods, and Algorithms”, ***IEEE Press***,2001.

[40] Roiger, R.J., Geatz M. W.,”Data Mining, A Tutorial-based Primer”, Third Edition, ***Addison Wesley***, (2003).

EK-A K-MEANS ALGORİTMASININ PROGRAM KODU

```
// k-means algoritmasını çalıştıran alt yordam
procedure Tform5.Button2Click(Sender: TObject);
var

i,j,n,l,m :       Integer;   { dongu degiskenleri }
k           :     Integer ; { k sayisi, gruplama sayisi }
boyut     :       Integer ; { koordinat sisteminin boyutu }
kayit_sayisi :    Integer ; { kumelenecek eleman sayisi }
sayac :           Integer ;
iterasyon :       Integer  ; { iterasyon sayisi }
t :               Real;   { noktalar arasi uzakligin hesabinda kull. }
Adr :             Integer;
ek :              Real;  { min. uzakligin hesaplanmasinda kull. }
son :             Boolean;

merkezler :       Array [1..2,1..2] of Real;  { [boyut sayisi] [k sayisi] }
uzk     :         Array [1..2,1..150] of Real;    { [k sayisi] [eleman sayisi] }
grupla1 :         Array [1..2,1..150] of Integer; { [k sayisi] [eleman sayisi] }
grupla2 :         Array [1..2,1..150] of Integer;
renkler :         Array [1..150] of Integer; {[eleman sayisi] }
veriler :         Array [1..2,1..150] of Integer; { [boyut sayisi] [eleman sayisi] }
const renk:       Array[1..9] of TColor =
(clBlue,clAqua,clLime,clFuchsia,clBlack,clMaroon,clYellow,clRed,clGreen);

dosya:TextFile ;

begin
//  işlem sonuçlarının yazılacağı text dosya
AssignFile(dosya,'c:\kmeans.txt');
Rewrite(dosya);
writeln(dosya,'k-means hesaplaması');

// k sayısı kayıt sayısından büyükse hata!
 if kayit_sayisi <= k
     then begin
           showmessage ('k sayısı toplam kayıt sayısından küçük veya eşit olmalı!');
           exit;
      end;

  for i:=1 to adoquery2.recordcount do
    begin
       writeln(dosya,i,'=',adoquery2.FieldValues[parametre1],'
         ',adoquery2.FieldValues[parametre2]);
```

```
        veriler [1,i] := adoquery2.FieldValues[parametre1];
        veriler [2,i] := adoquery2.FieldValues[parametre2];
        adoquery2.Next;
      end;

boyut       := 2; { koordinat sisteminin boyutu }

{*************************************************************}

writeln(dosya);
writeln(dosya,'k sayisi        = ',k);
writeln(dosya,'koordinat sayisi= ',boyut);
writeln(dosya,'kayıt sayisi    = ',kayit_sayisi);
writeln(dosya);

son := false;
iterasyon := 0;

writeln(dosya,'okunan veriler:');
{ okunan verilerin yazdırılması }
for j:=1 to kayit_sayisi do
   begin
      write(dosya,j,' => ');
      for m:=1 to boyut do   write(dosya,veriler [m,j],'   ');
      writeln(dosya);
   end;

{ kümeleme matrisinin sifirlanmasi }
for j:=1 to k do
         for m:=1 to kayit_sayisi do
                  begin grupla1[j,m] := 0; grupla2[j,m] := 0; end;

{ ilk merkezlerin matrise atanmasi, verilerin ilk k tanesi aliniyor }
writeln (dosya,'ilk merkezler...:');
for i:=1 to boyut do
 begin
    for j:=1 to k do
      begin merkezler [i,j] := veriler [i,j] ;
            write (dosya,merkezler[i,j], ' ');
      end;
      writeln (dosya);
 end;

while (not son) do
         begin

         { uzaklik matrisinin hesaplanmasi }
               for l:=1 to k do
                  begin
```

```
                    for n:=1 to kayit_sayisi do
                     begin  t := 0;
                            for m:=1 to boyut do
                            begin
                                    t := t + ((merkezler[m,l] - veriler[m,n]) *
                                    (merkezler[m,l] - veriler[m,n])) ;
                            end;
                            uzk [l,n] := sqrt(t);
                    end;
            end;

    { uzaklik matrisinin yazdirilmasi }
    writeln (dosya,'uzaklik matrisi..:');
    for i:=1 to k do
    begin
            for j:=1 to kayit_sayisi do
                  write (dosya,uzk[i,j],' ');
            writeln (dosya);
    end;

    { min. uzakligin hesaplanmasi }
    for i:=1 to kayit_sayisi do
    begin
            ek := uzk[1,i] ;
            adr := 1 ;

            for j:=2 to k do
            begin
                    if (ek > uzk[j,i])
                    then begin
                            ek := uzk[j,i] ;
                            adr := j ;
                          end;
            end;
            grupla1 [adr,i] := 1 ;
     end;

     iterasyon := iterasyon + 1 ;

{ kümeleme matrisinin yazdirilmasi }
    writeln (dosya, 'kumeleme matrisi..:'  , iterasyon);
    for i:=1 to k do
    begin
            for j:=1 to kayit_sayisi do
               write (dosya,grupla1[i,j],' ');
            writeln (dosya);
    end;

    { uzaklik hesaplamasi bitti }
```

```
        { onceki ve sonraki kumeleme matrislerinin karsilastirilmasi }
        n := 0;
        for j:=1 to k do
                for m:=1 to kayit_sayisi do
                        if (grupla1 [j,m] <> grupla2 [j,m])
                        then begin n := 1; break; end;

        { kumeleme matrisleri farkliysa islemler suruyor }
        if (n = 1)
        then begin
                { onceki kumeleme matrisi saklaniyor }
                for j:=1 to k do
                        for m:=1 to kayit_sayisi do
                        begin   grupla2 [j,m] := grupla1 [j,m];   end;

        { kumeleme matrisinden yeni kume merkezlerinin bulunmasi }
        for j:=1 to k do
                for m:=1 to boyut do merkezler[j,m] := 0;

        for j:=1 to k do
        begin
                sayac := 0 ;
                for i:=1 to kayit_sayisi do
                begin
                        if (grupla1 [j,i] = 1)
                        then begin
                                sayac := sayac + 1;
                                for n:=1 to boyut do
                                  merkezler[n,j] := merkezler[n,j] + veriler[n,i] ;
                        end;
                end;
                for n:=1 to boyut do  merkezler[n,j] := merkezler[n,j] ;
        end;

        writeln (dosya,'yeni merkezler....:') ;
        for i:=1 to k do
        begin
                for j:=1 to boyut do  write (dosya,merkezler[i,j],' ');
                writeln (dosya);
        end;

        { kumeleme matrisinin sifirlanmasi }
        for j:=1 to k do
                for m:=1 to kayit_sayisi do   grupla1[j,m] := 0;

{ yeni merkezlerin hesaplamasi bitti }
  end
else    son := true;
```

```
end;

{ sonuc yazdirma }
writeln (dosya, 'SONUC' );
writeln (dosya,'iterasyon sayısı=',iterasyon); writeln (dosya);

for j:=1 to k do
begin
      write (dosya, j  , '. kümedeki kayitlar ' );
      sayac:=0;
         for m:=1 to kayit_sayisi do
         begin
                  if (grupla1 [j,m] = 1)
                    then begin
                            write (dosya, m  ,' ');
                            sayac := sayac + 1;
                           renkler[m]:= j;
                     end;
         end;
         writeln (dosya, ' % ' , trunc((sayac * 100)/kayit_sayisi));
         listbox3.items.Add(inttostr(j)+'. kümedeki kayıt sayısı '+inttostr(sayac)+' - %
         '+inttostr(trunc((sayac * 100)/kayit_sayisi)));
end;

writeln (dosya); writeln (dosya,'Renk dizisi'); writeln (dosya);

for m:=1 to kayit_sayisi do  write (dosya,renkler[m],' ');

closeFile(dosya);

{veriler matrisinin grafiğe aktarılması }
chart1.Series[0].Clear;

for m:=1 to kayit_sayisi do

         chart1.Series[0].AddXY(veriler[1,m],veriler[2,m],inttostr(veriler[1,m]),renk[r
         enkler[m]]);

chart1.refresh;

listbox3.items.Add(' ');
listbox3.items.Add('İşlenen kayıt sayısı = '+inttostr(kayit_sayisi));

end;
```

ÖZGEÇMİŞ

1966 yılında İstanbul'da doğdu. İlk, orta ve lise öğrenimini İstanbul'da tamamladı. 1983 yılında girdiği Yıldız Üniversitesi Mühendislik Fakültesi Bilgisayar Bilimleri Mühendisliği Bölümü'nden 1988 yılında Bilgisayar Mühendisi olarak mezun oldu. 1989-2002 yılları arasında, özel sektörde yazılım alanında çeşitli görevlerde çalıştı. 2004 yılından beri Maltepe Üniversitesi Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü'nde Öğretim Görevlisi olarak görev yapmaktadır.