KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

MERKEZ TABANLI KÜMELEME ALGORİTMALARININ KARŞILAŞTIRILMASI

YÜKSEK LİSANS

Bilgisayar Müh. Aysel BİLGİN

Anabilim Dalı: Bilgisayar Mühendisliği

Danışman: Yrd. Doç. Dr. Nevcihan DURU

KOCAELİ, 2008

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

MERKEZ TABANLI KÜMELEME ALGORİTMALARININ KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

Aysel BİLGİN

Tezin Enstitüye Verildiği Tarih: 13 Haziran 2008

Tezin Savunulduğu Tarih: 03 Temmuz 2008

Tez Danışmanı	Üye	Üye
Yrd.Doç.Dr. Nevcihan DURU	Prof.Dr. Kadir ERKAN	Yrd.Doç.Dr. Songül ALBAYRAK
(.....)	(.....)	(.....)

KOCAELİ, 2008

ÖNSÖZ ve TEŞEKKÜR

Veri madenciliği alanında kullanılan birçok modelleme tekniği vardır. Bu tekniklerden biri olan kümeleme veri kümesini doğal kümelere ayırma işlemi olarak tanımlanabilir. Kümeleme genellikle diğer modelleme teknikleri için bir ön adım olarak kullanılmaktadır. Günümüzün ihtiyaçlarının sürekli artması, eski kümeleme algoritmalarının yeni versiyonlarının çıkmasına ve daha yeni algoritmaların üretilmesine neden olmaktadır. Bu tezde kümeleme algoritmalarının bir çeşidi olan merkez tabanlı kümeleme algoritmaları üzerinde durulmuş ve bu algoritmalar daha önceden geçerliliği kanıtlanmış veritabanları üzerinde uygulanarak belirlenen kıstaslar doğrultusunda karşılaştırılmıştır.

Bu tez çalışması sırasında daha yakından tanıma fırsatı bulduğum, tanıdıkça daha fazla saygı duyduğum tez danışmanım Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü Öğretim Üyesi Yrd. Doç. Dr. Nevcihan Duru'ya emeklerinden dolayı teşekkür ederim. Ayrıca tezim sırasında moralimin hep üst seviyede olmasını sağlayan, yaptığım her işte arkamda olduğunu hissettiren ve bana inanan ailem ve nişanlıma da çok teşekkür ederim.

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

SEMBOLLER

c^j	: j. küme
ε	: bir veri nesnesi merkezli dairenin yarıçapı
ξ	: DENCLUDE algoritmasında kullanılan gürültü eşiği
k	: oluşturulacak küme sayısı
MinPts	: bir veri nesnesinin ε komşuluğundaki nokta sayısı
O	: hesaplanabilir karmaşıklık
σ	: DENCLUDE algoritmasında kullanılan yoğunluk parametresi.
x^i	: veri kümesi içindeki i. eleman

Kısaltmalar

AGNES	: AGglomerative NESting
BIRCH	: Balanced Iterative Reducing and Clustering Using Hierarchies
CF	: Clustering Feature Tree
CHAELEON	: A Hierarchical Clustering Algorithm Using Dynmaic Modeling
CLARA	: Clustering LARge Applications
CLARANS	: CLustering Algorithm based on RANdomized Search
CLIQUE	: Clustering High-Dimensional Space
CU	: Category Utility
CURE	: Clustering Using REpresentatives
DBSCAN	: Density Based Spatial Clustering of Applications with Noise
DENCLUE	: Clustering Based On Density Distribution Functions
DIANA	: DIvisive ANAlysis
EM	: Gaussian Expectation Maximization
FKM	: Fuzzy K-Means
HA	: Harmonic Average
H1	: Hybrid 1
H2	: Hybrid 2
I/O	: Input/Output
KDD	: Knowledge Discovery in Databases
KM	: K-Means
KHM	: K-Harmonic Means
OPTICS	: Ordering Points To Identify the Clustering Structure
PAM	: Partitioning Around Medoids
ROCK	: Robust Clustering Algorithm
STING	: Statistical Information Grid
TS	: Tabu Search
TabuKHM	: Tabu K-Haronic Means

MERKEZ TABANLI KÜMELEME ALGORİTMALARININ KARŞILAŞTIRILMASI

Aysel BİLGİN

Anahtar Kelimeler: Kümeleme Analizi, Kümeleme Metotları, Merkez Tabanlı Kümeleme Algoritmaları, K-Ortalama Algoritması, Bulanık K-Ortalama Algoritması, K-Harmonik Ortalama Algoritması, Hibrit 1, Hibrit 2.

Özet: Kümeleme, Öklit veya Manhattan uzaklığı gibi bir benzerlik ölçümüne dayalı olarak veriyi doğal gruplara ayırma işlemidir. Kümelemede amaç, grup içindeki nesnelerin benzer olması ve bu nesnelerin diğer gruplar içindeki nesnelerden farklı ve başka olmasıdır. Kümelemenin biyoloji, iklim, eğitim, arkeoloji, örüntü tanımlama, tıp, psikoloji ve ilaçlar, elektronik bankacılık, görüntü işleme, astronomi, istatistik ve mühendislik gibi alanlar ile yakından ilişki olması onun daha da gelişmesini sağlamıştır. Kümelenecek olan verinin yapısına bağlı olarak farklı özelliklere sahip birçok kümeleme metodu ortaya çıkmıştır. Kümeleme metotlarından en popüler olanlardan biri bölümlemeli kümeleme metotlarının bir sınıfı olan merkez tabanlı kümeleme algoritmalardır. Merkez tabanlı kümeleme algoritmaları içinde en temel olan K-ortalama kümeleme algoritmasıdır. Diğer merkez tabanlı kümeleme algoritmaları, beklenen eniyileme algoritması ve K-ortalama algoritmasından türetilmiş olan, Bulanık K-Ortalama ve K-Harmonik Ortalama algoritmalarıdır. Merkez tabanlı kümeleme algoritmalarının her birinin kendine ait bir amaç fonksiyonu bulunmaktadır. Bu algoritmaların amacı, kendi amaç fonksiyonlarını en aza indirmektir. Bu çalışma da K-Ortalama, Bulanık K-Ortalama, K-Harmonik Ortalama algoritmaları ve K-Ortalama ve K-Harmonik Ortalama algoritmalarının özelliklerini içeren Hibrit 1 ve Hibrit 2 algoritmaları farklı veri kümeleri üzerinde uygulanmış ve performans değeri ve işlemci zamanına göre karşılaştırılmıştır. Çalışmada kullanılan veriler UCI veri deposundan alınmıştır. Bu çalışma ile merkez tabanlı kümeleme algoritmalarından biri ile kümeleme işlemi yapılacağı zaman ilgili veri kümesi için hangi algoritmanın daha uygun olduğuna karar vermede uzman kişiye yardımcı olmak hedeflenmiştir.

THE COMPARISON A CENTER-BASED CLUSTERING ALGORITHMS

Aysel BİLGİN

Keywords: Clustering Analysis, Clustering Methods, Center-Based Clustering Algorithms, K-means Algorithm, Fuzzy K-Means Algorithm, K-Harmonic Means Algorithm, Hybrid 1, Hybrid 2.

Abstract: Data clustering is the process of identifying clusters based on some similarity measure like Euclidean, Manhattan distance. The goal of clustering is that patterns within a cluster are similar and different from the patterns in other clusters. The close relationship between data clustering and biology, climate, education, archeology, pattern recognition, medical, psychology and medicine, banking, signal processing, astronomy, statistic, engineering, has caused to improve it. Many clustering methods have appeared based on the structure of data that will be clustered. One popular class of data clustering algorithms is the center-based clustering algorithms. The main algorithm in the center-based clustering algorithms is K-means clustering algorithm. The other center based clustering algorithms, which was developed from k-means and Expectation-maximization, are fuzzy k-means and k-harmonic means algorithm. They each have their own objective function and they try to mininize its own objective function. In this study k-means, fuzzy k-means, k-harmonic means algorithms and two algorithms are named Hybrid 1 and Hybrid 2 that combine features of k-means and k-harmonic means algorithms have been run on different kind of data sets and compared according to their performance value and CPU time. Data that used in this study have been taken from UCI warehouse. The purpose of this study is to help experts making decision about suitable algorithm for relevant data set when they will make a clustering with one of these center-based clustering algorithms.

1. GİRİŞ

Bilişim teknolojilerinde takip etmekte zorlandığımız gelişmeler yaşanmaktadır. Teknolojideki gelişim, bilgisayar teknolojisine de paralel olarak yansımaktadır. Bilgisayarların hesaplama güçleri ve disklerin kapasiteleri artarken fiyatlar azalmakta ve büyük miktardaki veri doğrudan sayısal olarak toplanıp saklanabilmekte ve daha kısa süre de işlenebilmektedir. Sürekli olarak artan veri yığınları belli bir amaca yönelik olarak işlenip bilgiye dönüştürülmediği sürece bizim için değersizdir. Veri işlenip bilgiye dönüştürüldüğünde bizim için bir anlam ifade etmeye başlar. Büyük veri yığınlarından yararlı bilgiye erişim ihtiyarcını karşılamak için veri madenciliği çözüm olarak sunulmuş ve giderek önemi artan bir araştırma alanı haline gelmiştir.

Veri madenciliği veri içinden ilginç, üstü kapalı ve anlamlı örüntüleri otomatik veya yarı otomatik olarak bulma işlemidir [1]. Veri içindeki örüntüler insan yaşamında önemli bir yere sahiptir. Bu örüntüler kullanım amaçlarına göre sürekli insanlar tarafından araştırılmaktadır. Avcılar hayvanların göç etmesindeki örüntüleri, çiftçiler ürün yetiştirmedeki örüntüleri, politikacılar oyların dağılımındaki örüntüleri ve sevgililer eşlerine karşılık vermedeki örüntüleri ileride kullanılmak üzere araştırmaktadırlar [2]. Veri içinden ilginç örüntüler elde edilmesinde kullanılan veri madenciliği; analiz etmek için veriyi seçme, veriyi hazırlama, veriyi birleştirme, veriyi dönüştürme, veri madenciliği algoritmalarına başvurma ve sonra sonuçları yorumlama ve değerlendirme şeklindeki birkaç adımdan meydana gelir [1].Veri madenciliği araçları veriden örüntüleri bulur ve bunlardan bağlantı ve kuralları çıkarır. Çıkarılan bilgi sonra veritabanları arasında veya veri kayıtları arasındaki bağlantıları tanımlayan tahmin ve sınıflandırma modelleri için kullanılır. Bu örüntüler ve kurallar, karar vermede ve bu kararların etkilerini tahmin etmede rehberlik edebilirler.

Veri madenciliği deyimi literatüre yerleşmeden önce bilim adamları tarafından veri madenciliğine eş değer birçok adlandırmalar ortaya atılmıştır. Ancak bunlardan en

fazla rağbet göre veritabanlarında bilgi keşfi (KDD-Knowledge Discovery in Databases) terimi olmuştur. Bazı bilim adamları veri madenciliği ile KDD' nin aynı olduğunu, bazıları da KDD' nin bir süreç olduğunu ve veri madenciliğinin de bu süreç içindeki bir adım olduğu görüşünü benimsemişlerdir. KDD ve veri madenciliği arasındaki farkı göstermek Fayyad, Shapiro ve Smyth' ın makalesinin ana konusu olmuştur. Veri madenciliği, veriden örüntülerin çıkarılması için belirli algoritmaların uygulanması anlamına gelmektedir. KDD süreci ise veri hazırlama, veri seçme, veri temizleme ve veri madenciliği sonucu çıkan sonuçların yorumlanması gibi ek adımlarla birlikte veriden türetilen yararlı bilginin elde edilmesi demektir. KDD işlemi kullanıcı tarafından verilen kararlarla, etkileşimli ve tekrar eden birçok adımı içeren bir işlemdir. Bu makalede KDD işlemi 9 adım içermektedir. Birinci adımda; müşteri bakış açısına göre KDD işleminin amacı tanımlanır, bu işlem sonucunda istenen öncelikli bilgi ve uygulama alanı anlaşılmaya çalışılır. İkinci adımda; üzerinde işlem yapılacak veri kümesi yaratılır. Üçüncü adımda; veritabanları içindeki veriler üzerinde işlem yapılmanda önce önişlemlere tabi tutulması gerekir. Verinin gürültülü, eksik, tutarsız verilerden arındırılması bu aşamada yapılır. Dördüncü adımda çok büyük miktardaki verinin analiz edilmesi zor olacağından verinin bütünlüğü bozmayacak şekilde tamamını temsil edecek veri alınır ve analiz edilir. Bu işleme veri indirgenmesi (Data Reduction) denmektedir. Ayrıca bu aşamada boyut azalımı ve dönüştürme metotları ile kullanılacak olan niteliklerin efektif sayısı bulunur. Beşinci aşama da; birinci aşamadaki KDD işleminin hedefleri veri madenciliği metodu ile eşleştirilir. Altıncı aşamada model ve hipotez seçilir. Veri madenciliği algoritmalarının seçimi ve veri örüntülerinin araştırılmasında kullanılacak olan modelin seçimi yapılır. Hangi parametrelerin ve modellerin uygun olduğuna karar verilir. Yedinci adım seçilen veri madenciliği algoritmasının uygulandığı aşamadır. Sekizinci aşama veri madenciliği algoritmalarının uygulanması ile açığa çıkan örüntülerin yorumlandığı aşamadır. Dokuzuncu aşama keşfedilen bilginin kullanıldığı aşamadır. Bu bilgi direkt olarak kullanılabildiği gibi rapor üretmek için kullanılabilir. Bu bilgi gelecekteki etkisini görmek üzere başka bir sistem içine aktarılabilir [3].

Veri madenciliği istatistik, dilbilim, veritabanları ve yapay zekâ gibi birçok bilim dalının katkıları ile gelişen ve gelişmeye devam eden çok disiplinli bir daldır.

İstatistik alanında regresyon, faktör, kümeleme, ayırma (discriminant) ve zaman serileri analizleri; yapay zekâ alanında makine öğrenimi, yapay sinir ağları, genetik algoritmalar, zeki ajan sistemleri (intelligent agent systems), bayes ağları, örüntü tanıma (pattern recognition) modelleri veri madenciliğine önemli katkılarda bulunmaktadır. Bilgisayar dilbilimi (computer linguistik) alanında ise web madenciliği (web usage mining), metin madenciliği ve vaka temelli çıkarım (case based reasoning) veri madenciliğinde önemli rol oynayan alanlardır [4].

Veri madenciliği ile yakından ilişkili olan iki disiplin; bilgi çıkarımı (information retrieval) ve metin madenciliğidir. Bilgi çıkarımı ve veri madenciliği teknikleri arasında birbirini tamamlayan bir ilişki vardır. Bilgi çıkarımı yıllardır veri tabanı sistemleri ile paralel olarak geliştirilmektedir. Yapısal veriler üzerinde sorgu ve bilgi işleme üzerine odaklanan veritabanı sistemlerinin aksine bilgi çıkarımı organizasyon ile ilgili olup, metin tabanlı dokümanlardan bilginin çıkarılmasıdır. Bir bilgi çıkarımı problemi anahtar kelimeler veya örnek dokümanlar vb. kullanıcı girişlerine bağlı olarak ilişkili dokümanların bulunmasıdır. Bilgi çıkarımı sistemleri, çevrim içi kütüphane katalog sistemleri ve çevrim içi doküman yönetim sistemlerini içerir. Veri madenciliğinde kullanılan birçok teknik bilgi çıkarımından gelir fakat veri madenciliği bilgi çıkarımının ötesine geçmektedir. Veri madenciliği diğer taraftan depo içinde var olan veriye erişmek ile ilgilenmez. Bunun yerine, bize yeni şeyler söyleyecek veri içinde açık olmayan örüntülerle ilgilenir. Bilgi çıkarımı teknikleri metin tabanlı koleksiyonlara uygulanır. Veri madenciliği teknikleri; geçici veri ve karmaşık veri, meta veri, internet tabanlı içerik ve veritabanları gibi metin dokümanlarına uygulanabilir [1].

Geleneksel bilgi çıkarım teknikleri, metin verilerinin büyük boyutlarda artışı karşısında etkisiz kalmaktadır. Çoğunlukla, elde edilebilir dokümanın yalnızca küçük bir kısmı verilen kullanıcı ile ilişkili olmaktadır. Dokümanların içerisinde ne bulunabileceğini bilmeden verilerin çözümlenmesi ve kullanışlı bilginin çıkarılması için etkili sorgular oluşturmak oldukça zor olmaktadır. Kullanıcılar, farklı dokümanları karşılaştırmak, önemlerine göre derecelendirmek ve ilişki kurmak veya çoklu dokümanlar arasından örnekleri ve eğilimleri bulmak için bazı araçlara ihtiyaç duymaktadır. Metin madenciliği kavramı burada açığa çıkmaktadır. Metin

madenciliği metinleri bilgiye dönüştürme işlemi olarak tanımlanabilmektedir. Metin madenciliğinde metin dokümanları sayısal simgelere dönüştürülerek standart veri madenciliği metotlarının kullanılması sağlanır. Metin madenciliği ile birbirine benzer olan dokümanlar otomatik olarak çıkarılan anahtar sözcüklerin tekrar sayısına göre belirlenir [1].

Veri madenciliği ile ilişkili olan diğer bir disiplinde web madenciliğidir. Web madenciliği, veri madenciliği tekniklerinin web üzerinde uygulanmasıyla web' te bulunan veriden faydalı bilginin keşfedilmesi ve yorumlanması şeklinde tanımlanabilir. İnternette çok büyük veri olmasına rağmen bunlar son derece dağınık ve düzensiz yapıda bulunmaktadır. İnternette bilginin artması ve web sitelerinin etkinleştirilmesi ihtiyacı web madenciliğinin ortaya çıkmasına neden olmuştur. Web içerik madenciliği ve web kullanım madenciliği olmak üzere web madenciliği 2' ye ayrılmıştır. Web dokümanları içerisinde metin, resim, ses, görüntü, meta veri bulunmaktadır. Web içerik madenciliğinin amacı bu dokümanlar içerisinden bilginin bulunması veya filtrelenmesidir. Bu konuda Ajan temelli yaklaşım (agent based approach) ve Veritabanı yaklaşımı (database approach) olmak üzere iki yöntem vardır. Web kullanım madenciliği kullanıcıların web' de dolaşırken yaptıkları erişim hareketlerince oluşturulan veriden bilgi üretmeyi hedefler. Kullanıcı kayıt bilgileri veya geçmiş bilgileri, oturum ve hareket bilgileri, site yapısı ve içeriği kullanıcının veya sitenin karakterini çıkarmamıza yardımcı olan veri kümelerini barındırmaktadır. Web madenciliği sayesinde; kullanıcıların şekilleri çıkarılabilir ve zaman içindeki değişimleri takip edilebilir. Ayrıca sitedeki beğenilen ya da beğenilmeyen köşeler tespit edilebilir, sistemimizin güvenliğinin az olduğu noktalar belirlenebilir ve saldırı ve sahtekârlık kalıpları belirlenebilir, kullanıcı ve ziyaret davranışlarının modellenmesi, kullanıcıların gezinti şekli/hızı sitenin içerik, yapılandırma ve alt-yapı açısından performansı hakkında bir fikir edinmemizi sağlayabilir [5].

Veri madenciliği farklı birçok alanda kullanıldığından veri madenciliği ile birçok görüş ortaya atılmıştır. Veri madenciliğinin, veritabanı, makine öğrenimi ve istatistik olmak üzere 3 farklı bakış açısı vardır. Veritabanı bakış açısında "verimlilik" ön plandadır, çünkü bu bakış açısı tüm keşfetme işlemi ve büyük miktarda veri ile uğraşır. Makine öğrenimi bakış açısında ise "yararlılık" ön plandadır, çünkü bu bakış

açısı veri analizinde deneye dayalı çalışmadan etkilenir, fakat her zaman kullanışlı olmayabilir. İstatistik bakış açısında ise "geçerlilik (doğruluk) " ön plandadır, çünkü bu bakış açısı madencilik metotlarının arkasındaki matematiksel geçerliliği önemser [6].

Veri madenciliği veri içinde açık olmayan ilginç örüntüleri bulmada kullanılan bir araçtır fakat veri madenciliği sihirli bir değnek değildir. Veri madenciliği veritabanı içinde neler olduğunu izlemez ve veritabanı içinde ilginç örüntüler gördüğünde bizim dikkatimizi çekmek için maille bu durumu bize iletmez. Hiçbir veri madenciliği algoritması incelenmesi gereken işin ve verilerin özelliklerinin bilinmemesi durumunda fayda sağlamaz. Veri madenciliği veri içindeki bağıntıları ve örüntüleri bularak iş analistine yardımcı olur. Eğer iş analisti ilgili iş ve veri özelliklerinin derinlemesine öğrenmemiş ve anlamamış ise veri madenciliği işlemi sonucunda elde edilen örüntülerden yararlanamaz. Veri madenciliği, kuruluşa bu örüntülerin değerini ve nasıl yararlı olabileceğini söylemez. Veri madenciliği yetenekli iş analistleri ve müdürlerin yerini alamaz fakat onların yapacağı işi daha da geliştirmek için oldukça güçlü yeni araçları onlara verir [7] .

Veri madenciliği birçok alanda yaygın olarak kullanılmaktadır. Bu alanlardan biri de tıptır. Tıpta veri madenciliği, belirli bir hastalığa sahip hastaların özelliklerinin ve hastaların ilgili hastalıktan kurtulma şansının belirlenmesi, hastaların ilgili hastalığın tedavisine yanıt verip vermeme durumlarının tahmini, kullanılan ilaçların yan etkilerinin ve hastaların hastanede kalış sürelerinin tespiti gibi yararlı bilgilerin elde edilmesinde kullanılmaktadır. Diyabet tıp alanında veri madenciliği teknolojileri için birçok nedenden dolayı uygun bir hastalıktır. Birinci olarak veri kümesi mevcuttur. İkinci olarak diyabet büyük miktarda paranın harcanmasına neden olan genel bir hastalıktır. Üçüncü olarak diyabet körlük, böbrek iflası, uzuv kesilmesi, dolaşım sistemi hastalıkları ve bunlardan dolayı meydana gelen erken ölüm gibi yan etkilere neden olan bir hastalıktır. Sonuç olarak doktorlar hastalığın seyrini mümkün olduğunca düzeltmenin yolarını bilmek isteyeceklerdir. PIDD(Pima Indian Diabet Database) diyabet veritabanı veri madenciliği algoritmalarını test etmek için standart haline gelmiştir. Diyabet hastalığı ile ilgili tahminlerin doğruluğunu göstermek için tıbbi verilerde sıkça kullanılan veri tahmin aracı olan kaba kümeler(rough sets)

yöntemi kullanılmıştır. Rosetta yazılımı ile verilere uygulanan bu yöntemle, başka algoritmalar kullanılarak %66–81 arasında elde edilen tahmin başarısı %82'ye ulaşmıştır [8] .

Veri madenciliği alanında yapılmış çalışmalardan biri de IBM Almaden araştırma merkezinde yapılan sorgu veri madenciliği (quest data mining) projesidir. Bu proje ile geniş veritabanlarındaki yararlı örüntüleri keşfetmek için yeni teknolojiler geliştirilmiştir. Bu teknolojiler birliktelik kuralları, sınıflandırma, apriori algoritması zaman serilerini kümeleme vs. için madencilik içerir. IBM, veri madenciliği ürünü olan IBM Akıllı Madenci sayesinde bu teknolojilere ulaşabilmeyi sağlıyor. Sorgu sisteminin amacı; hızlı, ölçeklenebilir algoritmaları geliştirmek ve uygulamaları daha kısa yoldan çalıştıran basit veri madenciliği işlemlerini tanımlamaktır. Algoritmalardan örüntü varlığını onaylamak yerine, büyük veritabanlarındaki örüntüyü keşfetmesi, aynı türdeki örüntülerin keşfedildiğini garanti eden bir tamamlama özelliğine sahip olması, geniş gerçek veritabanlarında yüksek performansa sahip olması beklenmiştir. Sorgu sisteminde; madencilik algoritmaları veri kaynağına yakın olan sunucu üzerinde çalışır. Farklı istemci makine veya aynı çalışma istasyonu üzerinde çalışabilen GUI sayesinde kullanıcılar sistemle etkileşim halindedir. Kullanıcı opsiyonel olarak herhangi madencilik işleminin sonuçlarını, tercih ettiği bir yazılıma API sayesinde aktarabilmektedir. Sorgu mimarisinin ilginç yapısı onun I/O (Input/Output) mimarisinde gizlidir. Veri giriş API içine konulan veri depolarındaki detaylardan algoritma kodunu ayırarak, girdiye bütün ulaşımlar için tanımlanan standart bir arayüz vardır. Bu sayede sorgu sistemine yeni veri depo çeşitlerini eklemek kolay olur. Sorgu sistemi AIX ve MVS platformlarının her ikisinde de çalışmaktadır [9].

Veri madenciliği alanın yapılan diğer çalışma da dinamik veri madenciliğidir. Dinamik veri madenciliği, veri içinden daha fazla bilgi kazanılmasını sağlar. Veri madenciliği bilgilerinin sonuçlarının doğruluğu, performans ve sonuçların yorumlanması, veritabanı güncellemelerinin etkin yönetimi ile ilişkili problemlerin çözümünde dinamik veri madenciliği kullanılmaktadır. Dinamik veri madenciliği uygulamasında; önceki veri madenciliği işlemlerinde elde edilen bilgiler dinamik olarak güncellenir. Uzun süren işlemler ardı ardına gelen bölümler kümesini

oluşturacak şekilde bölünür. Uygulama da Apriori benzeri bir yaklaşım kullanılmıştır. Bu uygulama da önceki bölümlerde keşfedilen veri madenciliği kuralları ile birlikte geçerli bölüm süresince var olan güncellemeleri kullanarak geçerli veri madenciliği kuralları keşfedilmiştir [10].

Veritabanları, veri ambarları, uzaysal veri, çoklum ortam verisi, internet tabanlı veri ve karmaşık nesneleri içeren veri depolarına veri madenciliği teknikleri uygulanarak yararlı bilgi çıkarılmaya çalışılmaktadır. Veri madenciliği teknikleri için farklı birçok gruplandırma yapılmıştır. Bunlardan Han' in ileriye sürdüğü kategoriler; ayrımlama ve tanımlama, birliktelik analizi, sınıflandırma ve öngörü, kümeleme analizi, sıra dışılık analizi ve gelişimsel analiz kategorileridir. Berry ve Linoff' un ileriye sürüdüğü kategoriler ise sınıflandırma, kestirim, tahmin, benzer gruplama, kümeleme ve tanımlama ve profil oluşturmadır. Bu tezde, Berry ve Linoff' un sunduğu kategoriler ikinci bölümde ayrıntılı olarak ele alınmıştır.

Veri madenciliği tekniklerinden olan sınıflandırma bir çeşit örüntü tanımlama işlemidir. Nesnelerin veya bir şeylerin sınıflanması bilginin özündedir. Nesneleri sınıflandırmak ve tanımlamak için bazı modern lineer olmayan metotlar incelenmiştir. Bu metotlar kümeleme, gelişmiş kümeleme, olasılık ve yapay sinir ağlarıdır. Örüntü tanımlama; sınıflandırma ve tanımlama olmak üzere 2 tane işlem içerir. Sınıflama; bir nesne topluluğundan alınan örnek sınıf olarak adlandırılan gruplara bölünme işlemidir. Tanımlama; aynı popülasyondan verilen bilinmeyen bir nesnenin tanımlanan sınıflardan bir tanesine ait olduğunun tanımlanmasıdır. Tanımlama bazen tanımlama ve kimlik olarak ayrılır. Bu durum özel bir nesnenin tanımlandığı anlamına gelir. Sınıflandırma ve tanımlama terimleri literatürde yer değiştirerek kullanılır. Bir sınıflandırma işlemi; popülasyonu temsil eden nesneler örneğini inceler. Sınıflar arasındaki nesnelerin benzerliği ve sınıflar içindeki nesnelerin benzerliğine göre örneği alt sınıflara parçalar. Bu tip işlem; eğitilmemiş öğrenme olarak adlandırılır. Bir tanımlayıcı, aynı topluluktan herhangi bilinmeyen bir nesneye sınıf etiketi atamak için eğitilebilir. Eğitilme işlemi; tanımlayıcının eğitilmesi veya eğitilmiş öğrenme olarak bilinir. Eğitilmiş tanımlayıcı; çevrimiçi olarak örüntü tanımlamayı gerçekleştirebilir [11]. Birçok sınıflandırma modeli vardır. Nöron ağları, genetik algoritmalar, Bayes metotları, istatistiksel metotlar ve

karar ağaçları bu sınıflandırma modellerine örnek olarak verilebilir. Bu modellerden karar ağaçları veri madenciliğinde önemli bir yere sahiptir. Karar ağaçlarını oluşturma analist tarafından verilmesi gereken herhangi bir giriş değişkeni gerektirmez. Çok geniş eğitim veritabanlarından karar ağaçlarını oluşturmak için hızlı ve ölçeklenebilir algoritmalar kullanılabilir. Karar ağaçlarının tahmini doğruluğu diğer sınıflandırma modellerinden daha yüksek ya da eşittir [12].

Veri madenciliği tekniklerinden bir diğeri olan kümeleme birçok algoritma için temel basamak özelliğini taşımaktadır. Kümeleme bazı araştırmacılara göre heterojen grupları parçalayıp daha homojen olan alt gruplara dönüştürmek bazılarına da göre de veri kümesinden gruplar bulmak olarak tanımlanmaktadır. Kümeleme sınıflandırmaya benzemektedir. Fakat ondan farklı olarak kümeleme de önceden tanımlanmış bir sınıf yoktur. Veriler kendi aralarındaki benzerliklere göre gruplandırılır. Bu nedenle kümeleme denetimsiz sınıflandırma (unsupervised learning) olarak adlandırılır [13]. Bir kümeleme işleminin başarılı bir şekilde tamamlanması için kümeleme işlemine başlanmadan önce kümelenecek verinin analiz edilmesi, kümelenecek veri parçalarının ve değişkenlerin seçilmesi, benzersizlik ölçümlerinin ve kümelemenin amaç fonksiyonunun belirlenmesi, eksik veri durumunda izlenecek stratejinin belirlenmesi, kümelemede kullanılacak algoritmanın ve küme sayılarının seçilmesi gerekmektedir. Bu adımların baştan aşağıya uygulanması ile başarılı kümeleme sonuçları elde edilir [14].

Kümeleme birçok algoritma için temel basamak özelliği taşımaktadır. Kümelemenin mühendislik, tıp, eğitim gibi birçok alanda yaygın olarak kullanılması onun daha gelişmesine neden olmuştur. Birçok alanda yaygın olarak kullanıldığın algoritmaların eksik kaldığı noktaları görmek daha da kolaylaşmış ve ihtiyaçlar doğrultusun sürekli yeni algoritmalar ortaya çıkmıştır ve çıkmaya da devam etmektedir. Kümeleme algoritmalarının çeşitliliği başlangıç prensipleri ve modellerinin çeşitliliğinden kaynaklanmaktadır. Birçok başlangıç prensibi olmasının nedeni kümelemenin bakan göze göre değişiyor olmasıdır. Başlangıç prensipleri araştırmacıların inandığı küme tanımının matematiksel formülüdür. Bir kümeyi neyin oluşturduğu ve iyi bir kümelemeyi neyin oluşturduğu sübjektiftir [15].

Veri madenciliğinde veri türüne ve kullanım amacına göre kullanılan birçok kümeleme algoritması vardır. Bunlardan biri de merkez tabanlı kümeleme algoritmalarıdır (A Center-Based Clustering Algorithms). Merkez tabanlı kümeleme algoritmaları içinde en temel olanı, K-ortalama (KM-K-Means) kümeleme algoritmasıdır. Diğer merkez tabanlı kümeleme algoritmaları, beklenen eniyileme (EM-Gaussian Expectation Maximization) ve KM kümeleme algoritmasından türetilmiş, Bulanık K-ortalama (FKM-Fuzzy K-Means) ve K-harmonik ortalama (KHM-K-Harmonic Means) algoritmalarıdır. Merkez tabanlı kümeleme algoritmalarının her birinin kendine ait bir amaç fonksiyonu bulunmaktadır. Bu algoritmaların amacı, kendi amaç fonksiyonlarını en aza indirmektir.

Merkez tabanlı kümeleme algoritmalarından olan KM algoritması ilk defa MacQueen tarafından 1967' de tanıtılmıştır. KM algoritması n adet eleman oluşan veri kümesini giriş parametresi olarak verilen k adet kümeye böler. KM algoritmasının ilk adımı, küme merkezlerini temsil edecek k tane elemanın belirlenmesi ile başlar. Bunlar, veri kümesinin ilk k adet elemanı olabileceği gibi, veri kümesi içinden rasgele seçilerek oluşturulmuş olan k tane eleman da olabilir. Belirlenen bu elemanlar ilk küme merkezlerini oluştururlar. İkinci adımda, veri kümesi içindeki her bir elemanın seçilen merkezlere olan uzaklığı Öklit uzaklık formülü kullanılarak hesaplanır. Elde edilen sonuçlara göre her bir eleman k adet kümeden kendisine en yakın olan kümeye dâhil edilir. Üçüncü adımda, yeni küme merkezleri ilgili küme içindeki elemanların ortalaması alınarak hesaplanır. Dördüncü adımda, belirlenen durdurma kıstası sağlanmamışsa ikinci adıma dönülerek işlemler tekrar edilir. Merkez tabanlı kümeleme algoritmalarında kullanılan farklı birçok durdurma kıstası vardır. Bu kıstaslar; belirlenen maksimum iterasyon sayısına ulaşılması, yeni kümelere verilerin minimal düzeyde atanması ve hiç atanmaması, toplam hatanın karesinin en küçük olması gibi kıstaslar olabilir. KM algoritmasının birçok avantajı vardır. Bunlar; hızlı çalışması, uygulanmasının kolay olması, geniş veritabanları içinde kullanışlı olmasını sağlayan zaman karmaşıklığının O(N) olması, verilerin sırasına bağımlı olmamasıdır. Ayrıca KM algoritması kategorik verilerle değil de sayısal verilerle çalışmaktadır [16]. KM algoritmasının birçok dezavantajı da vardı. Bunlardan en önemlileri; küme sayısı k' ya önceden karar verilmesi, algoritmanın başlangıçta seçilen merkezlere duyarlı olması ve hatta başarısız

başlangıç noktalarının seçilmesi ile boş kümelerin oluşması, veriye bağımlı olması, sıra dışılıklara duyarlı olması, kategorik niteliklerle çalışmaması ve algoritmanın yerel bir en küçük değerde tuzağa düşmesidir. Algoritma başlangıçta seçilen merkez noktalarına o kadar duyarlıdır ki kümeleme işlemi sonucunda toplam hatanın karesi fonksiyonun yerel bir en küçük değerinin elde edilebilir fakat iyi bir kümele için bizim ihtiyarcımız olan tümel en küçük değeridir [17]. KM algoritmasının dezavantajlarından biri olan k' ya önceden karar verilmesinde kümelemenin özünde yatan mantıkta yaralanılmaktadır. Küme içi uzaklık ve kümeler arası uzaklık ölçümlerinden yaralanarak en uygun k sayısına karar verilebilir. Küme içi uzaklık, her bir küme merkezi ile küme merkezine ait olan noktalar arasındaki uzaklıkların toplamının aritmetik ortalaması alınarak hesaplanır. Daha sonra kümeler arası uzaklıklar hesaplanır ve bu değerin minimumu alınır. Bu iki değer birbirlerine oranlanır. Elde edilen sonuç ne kadar küçük ise o kadar iyi bir kümeleme olmuş demektir [18].

Dezavantajları kadar avantajları olan KM algoritması çok yaygın olarak kullanılan bir algoritmadır. KM algoritmasının birçok uygulama alanı vardır. Bunlardan biri de tıptır. Tıp alanında gırtlak kanseri verilerinin analiz edilmesinde KM algoritması kullanılmış ve hastalığa ilişkin kararlar verilmesinde yardımcı olabilecek sonuçlar elde edilmiştir [19]. KM algoritması iklim alanında da uygulanmıştır. KM algoritması ile aylık yağış toplamları kullanılarak Türkiye' nin ana yağış bölgeleri belirlenmeye çalışılmıştır. 1977–2006 yılları için 148 noktada KM algoritması ile yağış verileri sınıflandırmaya tabi tutulmuş benzer özellikler gösteren istasyonlara ait olan yağış bölgeleri tespit edilmiştir [20].

Merkez tabanlı kümeleme algoritmalarının bir diğeri olan FKM algoritması, Dunn tarafından 1973 yılında önerilmiş ve Bezdek tarafından 1981'de geliştirilmiş olan bir algoritmadır. FKM algoritmasının KM algoritmasına göre avantajı her bir elemanın her bir kümeye belirli bir üyelik derecesi ile dâhil olmasıdır. Bu da veri kümesi içinde kümeler arasında çakışmanın olduğu gerçek uygulamalar için daha elverişli bir durumdur. FKM algoritmasının ilk adımında eleman sayısından küçük olacak şekilde k küme sayısı($1<k<n$) ve bulanıklılık katsayısı($r>1$) seçilir. Bulanıklılık katsayısının artması algoritmayı daha bulanık yapar. Bulanıklık katsayısının 1 olarak

seçilmesi onu KM algoritması gibi yapar [21]. İkinci adımda rasgele üyeliklerle üyelik matrisi belirlenir. Üçüncü adımda, üyelik matrisine göre merkezler hesaplanır. Dördüncü adımda, yeni merkezlere göre yeni üyelikler hesaplanır. Beşinci adımda, yeni üyelik matrisi ve eski üyelik matrisi arasındaki fark önemli ölçüde değişmiş ise üçüncü adıma dönülerek işlemler tekrar edilir. FKM algoritması yumuşak bir üyelik fonksiyonuna ve sabit bir ağırlık fonksiyonuna sahiptir. FKM algoritması KM algoritmasında daha iyi performansa sahiptir fakat KM de olduğu gibi k küme sayısının başlangıçta kullanıcı tarafından belirtilmesine ihtiyaç duyar. Ayrıca FKM algoritması da yerel bir en küçük değere yakınsar [22].

Diğer bir merkez tabanlı kümeleme algoritması olan KHM, KM ve EM algoritmasının başlangıçta verilen merkezlere olan duyarlılığından dolayı alternatif bir algoritma olarak ortaya atılmıştır. KHM algoritması harmonik ortalama (HA-Harmonic Average) fonksiyonunu kullanmaktadır. KHM bir veri noktasından tüm merkezlere olan karesel uzaklığın harmonik ortalamalarının bütün veri noktaları üzerindeki toplamını hesaplayan bir performans fonksiyonuna sahiptir. KM ve EM' den farklı olarak başlangıç nokta seçimine karşı duyarsızdır. KHM' de, KM' de kullanılan minimum fonksiyonu harmonik ortalama ile değiştirilerek farklı bir yaklaşım ele alınmıştır. HA, MIN' a benzerdir. Fakat daha yumuşaktır. KHM optimizasyonu daha kolay yapabilmeyi sağlar. Başlangıçta seçilen noktalar yerel minimumdan uzak olduğunda KHM, KM' den daha hızlı yakınsar. Fakat başlangıçta verilen noktalar yerel bir en küçük değere yakınsa KM çok hızlı yakınsar [23]. KHM yumuşak üyelik ve değişen bir ağırlık fonksiyonun sahiptir. KHM küme merkezlerine yardım etmek için bütün merkezlerden uzakta olan veri noktaları için yüksek ağırlıklar atar [24]. KHM ilk adımında, küme merkezlerini temsil edecek k tane elemanın belirlenmesi ile başlar. Belirlen bu elemanlar ilk küme merkezlerini oluştururlar. İkinci adımda merkezlere göre üyelik değerleri hesaplanır. Üçüncü adımda her bir elemanın ağırlık fonksiyonu hesaplanır. Dördüncü adımda yeni üyelik ve ağırlık değerlerine göre yeni merkezler hesaplanır. Beşinci adımda, herhangi bir durma kıstası sağlanmamış ise ikinci adıma dönülerek işlemler tekrar edilir. Altıncı adımda, ilgili durma kıstası sağlamış ise üyelik değerleri durulaştırılarak her bir veri noktasının hangi kümeye ait olduğu belirlenir. KHM algoritması başlangıçta seçilen merkez noktalarına diğer merkez tabanlı kümeleme algoritmalarına göre oldukça az

duyarlı olduğundan sahip olduğu yumuşak üyelik fonksiyonu ve değişen ağırlık fonksiyonu araştırmacıların ilgi alanı haline gelmiştir. KHM' in üyelik ve ağırlık fonksiyonun etkilerini araştırmak amacıyla Hibrit 1(H1-Hybrid 1) ve Hibrit 2 (H2-Hybrid 2) adında 2 yeni algoritma oluşturulmuştur. H1, KM' in katı üyelik fonksiyonu kullanır. Böylece her veri noktası hangi merkeze daha yakın ise sadece o merkezin bulunduğu küme içinde yer alabilir. Buna rağmen H1 algoritması KHM' in değişen ağırlık fonksiyonunu kullanır. Ağırlık fonksiyonu her bir merkezden uzakta olan noktalara daha fazla ağırlık verir. H1 algoritması katı üyelik fonksiyonuna sahip olsa bile ağırlıklardan dolayı KM' den çok daha hızlı yakınsar. H2 algoritması KHM' in yumuşak üyelik fonksiyonunu ve KM' in sabit ağırlık fonksiyonu kullanır. Bu yönüyle H2 algoritması KHM' e benzer [25].

Merkez tabanlı kümeleme algoritmaları olan KM, FKM, KHM, H1 ve H2 gibi kümeleme algoritmalarının kaliteli kümeler bulmada ne kadar etkili olduğunu görebilmek amacıyla bu algoritmalar performans bakımından karşılaştırılmıştır. Burada yapılan karşılaştırma, düşük boyutlarda yüksek kaliteli kümeler bulmada KHM' in ne kadar başarılı olduğu göstermiştir. H2, KHM gibi iyi sonuçlar vermiştir. Fakat H1 ise KM' den iyi sonuçlar vermesine rağmen H2 ve KHM' den daha iyi sonuçlar vermemiştir. KM' den aldığı katı üyelik böyle bir sonucun elde edilmesinde etkili olmuştur. Fakat KM' den iyi sonuç almasını da değişen ağırlık fonksiyonları sağlamıştır. Yüksek boyutlarda kümeleme önemli bir problemdir. Buna rağmen son araştırmalar boyut azaltma tekniklerinin tercih edilebileceğini ve sonrada KHM gibi düşük boyutlu kümeleme algoritmalarının kullanılabileceğini göstermiştir [26] .

Kümeleme alanında yapılan karşılaştırmalardan biri de Demiralay ve Çamurcu tarafından yapılmıştır. Bu çalışma da CURE, AGNES ve KM algoritmaları sentetik veri kümeleri üzerinde uygulanarak elde edilen sonuçlara göre karşılaştırılmışlardır. Karşılaştırma sonuçlarına göre, CURE ve AGNES algoritmalarının küresel kümeleri bulma da KM algoritmasından daha başarılı olduğu, KM algoritmasının küresel kümeleri bulabildiği fakat büyük boyutlu küresel kümelerin bulunmasında başarısız olduğu görülmüştür. Ayrıca CURE algoritmasının küresel ve şekilsiz kümeleri bulma da oldukça başarılı olduğu, AGNES algoritmasının ise küresel olmayan kümelerde kötü sonuçlar verdiği saptanmıştır [27].

Merkez tabanlı kümeleme algoritması olan KHM algoritmasını daha da geliştirmek için tabu arama tekniğinden yararlanılmış ve KHM ve tabu arama tekniğinin birleşiminden oluşan tabu k-harmonik ortalama (TabuKHM-Tabu K-Harmonic Means) algoritması ortaya atılmıştır. Tabu arama algoritması(TS-Tabu Search), optimizasyon problemlerinin çözümü için geliştirilmiş yinelemeli bir araştırma algoritmasıdır. TS yerel veya komşuluk arama prosedürünü kullanır. Yinelemeli olarak S çözümüne ve ona komşu olan S' çözümüne durdurma kıstasını sağlayıncaya kadar hareket eder. TS en önemli yönü tabu listesi(Tabu list) adında bir hafıza yapısına sahip olmasıdır. TS ismini de bu tabu listesinden alır. En basit anlamıyla tabu listesi yakın geçmişteki ziyaret edilmiş olan çözümleri içerir (n kadar hareketten önce). n tabu listesindeki uzunluktur. Yapılan çalışma sonucunda; KM ve FKM' in başlangıç koşullarına duyarlı olduğu, KHM ve TabuKHM' in KM ve FKM' den daha iyi sonuçlar verdiği, KHM' in işlemci zamanının KM ve FKM' den daha fazla olduğunu ve TabuKHM' in kullanıldığında da işlemci zamanının önemli ölçüde azaldığı görülmüştür[28].

KM algoritmasında ve diğer merkez tabanlı kümeleme algoritmalarında kullanılan birçok başlangıç yöntemi vardır. Bu yöntemler aracılığıyla kümelemede kullanılacak olan ilk küme merkezleri oluşturulur. Bu başlangıç yöntemlerinden Macqueen, rasgele (forgy) ve rasgele bölümleme (random partition) yöntemleri en çok tercih edilen başlangıç yöntemleridir. Bu yöntemlerden Macqueen yönteminde, veri kümesi içindeki ilk k tane veri noktası başlangıç küme merkezi olacak şekilde seçilir. Bu yöntemde önemli olan seçilen merkez noktalarının birbirini takip eder bir sırada seçilmesidir. Diğer bir yöntem olan rasgele yönteminde, veri kümesi içinden rasgele olarak k tane veri noktası seçilir. Rasgele bölümleme yönteminde ise, veri kümesi rasgele olarak seçilmiş k adet küme parçasına bölünür. Hangi parçanın hangi küme ile ilişkili olduğu belli değildir. Bu parçaların her biri rasgele seçilmiş olan k kümeden biri ile ilişkilendirilir ve her bir küme merkezi kendisi ile ilişkili olan parça içindeki veri noktalarının aritmetik ortalaması alınarak hesaplanır [29]. Bu yöntemler ile oluşturulan küme merkezleri, hem ilk küme merkezlerini hem de tek elemanlı ilk kümeleri oluştururlar. Bu yöntemlerden rasgele ve rasgele bölümleme yöntemleri, veri noktalarının sırasından bağımsız olarak başlangıç noktalarını oluştururlar. Macqueen yöntemi ise veri noktalarının sırasına bağımlı olarak başlangıç noktalarını

oluşturur. Bazı algoritmalar başlangıç yöntemleri kullanılarak elde edilen başlangıç noktalarından çok fazla etkilenmektedir. Bu başlangıç noktalarının kümelemeye etkisi kümeleme sonucu oluşan kümeler üzerinde görülebilmektedir. Algoritmaların başlangıca karşı olan duyarlılığından dolayı sürekli yeni başlangıç yöntemleri araştırmacılar tarafından oluşturulmaya çalışılmıştır. Bu yöntemlerden biri de Daoud tarafından geliştirilmiş olan başlangıç yöntemidir. Bu yöntemde ilk önce d boyutlu olan veri kümesindeki, her bir boyut ya da sütun içindeki verinin varyansı hesaplanır. Maksimum varyansa sahip olan sütun bulunur ve bu sütun herhangi bir sıralama ile sıralanır. k tane alt küme içine maksimum varyansa sahip olan sütunun veri noktaları bölünür. k burada istenen küme sayısını ifade eder. Her bir kümenin ortalaması alınır. Daha sonra küme merkezlerini oluşturmada her bir ortalamanın veri noktaları ile ilişkisi kullanılır [30].

Kümeleme üzerine yapılan birçok çalışma, sayısal veri üzerine odaklanır. Sayısal verinin kalıtsal geometrik özellikleri veri noktaları arasında uzaklık fonksiyonlarını tanımlamak için kullanılabilir. Çok geniş veritabanlarını kümelemek için hesapsal maliyet önceki algoritmaların birçoğunu kabul edilemez bir duruma getirmiştir. Son zamanlarda kategorik verinin kümelenmesi problemi ilgi çekmeye başlamıştır. Yapay zekâ da geliştirilen kavramsal kümeleme algoritmalarına karşı sayısal kümeleme algoritmaları değerlendirilmiştir. Sayısal teknikler bazı benzerlik ölçümlerine göre homojen kümelere karar vermeye odaklanır. Ancak bu işlemi düşük seviye kümelerin tanımlanmasını sağlayarak yapar. Kavramsal uygulamalar sınıfların daha yüksek seviye tanımları ile ilgilidir. Ralambondrainy, hibrit sayısal-sembolik bir metot sunmuştur. Bu metot; küme tanımı için tamamlayıcı kavramsal bir algoritmadır ve kümeye karar vermek için KM algoritmasının daha gelişmiş versiyonunu içerir. Kategorik nitelikler birçok kategoriye sahipse, Ralambondrainly'nin uygulamasına bağlı olarak kategorik nitelikleri ikili niteliklere çevirmek, bu sunulan tekniği hesaplama ve alan maliyetinin artışı ile karşı karşıya bırakmıştır. Küme ortalamalarını gösteren 0 ve 1 arasındaki gerçek değerler kümelerin özelliklerini ifade etmez. Huang, veri madenciliğinde geniş kategorik veri kümelerinin kümeleme probleminin üstesinden gelmek için k-modes algoritmasını alternatif olarak sunmuştur. K-modes algoritması; kümeleme maliyet fonksiyonunu minimuma indirgemek için kümelemede modelleri güncelleştiren sıklık tabanlı bir

metottur. K-modes algoritması kümeler için ortalamalar yerine modelleri, kategorik nesneler için Öklit uzaklık ölçüsü yerine basit eşleştirme benzersizlik ölçümünü kullanarak KM algoritmasını daha da geliştirmiştir. Bu sayede kategorik nesneleri için KM algoritmasını uygulamaya kalktığımızda karşılaştığımız küme merkezlerinin oluşturulması ve küme merkezleri ve nesneler arasında benzersizliğin hesaplanması problemleri de tamamen çözülmüştür [31].

Huang, karışık sayısal ve kategorik nitelikler tarafından tanımlanan kümeleme nesneleri için k-modes algoritması ve KM algoritmasını, k-prototip algoritması ile sonuçlanacak şekilde birleştirmiştir. Bu algoritma; kategorik nesneler için temsilci (represantative) denen küme merkezlerinin yeni bir gösterimini tanıtarak k-modes algoritması içindeki güçlükleri yok etmeye çalışmaktadır. Kategorik nesnelerin ortamında aritmetik işlemler olmadığından kümeler için ortalamaların yerine temsilcileri tanımlamada bulanıklık gösterimine başvurulmuştur. Bu gösterim ile KM algoritmasındaki bölümleme problemine benzer olarak kategorik nesnelerin kümeleme problemi formülleştirilmektedir. Kategorik nesnelerde aritmetik işlemler olmadığından, kartezyen ürünü ve birleşme operasyonları küme merkezlerini bulmak için kullanılır. KM algoritmasındaki küme merkezlerini hesaplamada kullanılan denklemdeki ekleme ve çarpma işlemi, birleşme ve kartezyen ürünü ile değiştirilmektedir. Kategorik nesnelerin kümeleri için temsilci oluşturmadaki değişiklik yüzünden, kategorik nesne ile kümenin temsilcileri arasındaki benzersizlik basit eşlemeye dayanarak tanımlanmaktadır. Kategorik nesneler arasındaki basit eşleme benzersizlik ölçümü, Öklit uzaklık ölçüsünün karesinin kategorik tersidir. Algoritmanın kümeleme performansı ölçmek için soybean disease ve nursery olmak üzere 2 veritabanı kullanılmıştır. Soybean veritabanında k-temsilci algoritması ile gerçekleştirilen uygulama sonunda 1000 kümeleme sonucu üretilmiştir. Kümeleme doğruluğu r>0,87' nin iyi bir kümeleme sonucu olduğu varsayılırsa, 686 iyi kümeleme sonucu üretilmiştir. Bu da k-temsilci algoritması çalıştırıldığında iyi sonuç elde etmek için %68,6 şansımız olduğunu göstermiştir. K-modes algoritması nursery veritabanına uygulandığında, kümeleme doğruluğu r>0,86 olduğu varsayıldığında 100 test içinde 691 iyi kümeleme sonucu elde edilmiştir. Bu da nursery veritabanı için k-temsilci algoritması çalıştırıldığında iyi kümeleme sonucu elde etmek için %69,1 şansımız olduğunu göstermiştir [31] .

KM kümeleme algoritmasının daha etkili çalışması için sürekli yeni algoritmalar geliştirilmektedir. Bu algoritmalardan biri de verimli kolaylaştırılmış KM algoritmasıdır. Her bir iterasyonda, KM algoritması veri noktası ve merkezler arasındaki uzaklığı hesaplar. Bu işlem büyük veritabanları için çok pahalıya mal olur. Bu maliyeti düşürebilmek amacıyla bu algoritmada KM' in önceki iterasyonlarından yararlanılmıştır. Her bir veri noktası için, en yakın kümeye olan uzaklığı bir değişkende tutulmuştur. Bir sonraki iterasyonda her bir veri için önceki en yakın kümeye olan uzaklığı hesaplanmıştır. Yeni uzaklık önceki uzaklığa eşit veya ondan daha az ise, bu noktanın bu küme içinde olduğu anlaşılır ve bu noktanın diğer küme merkezlerine olan uzaklığının hesaplanmasına gerek kalmaz. Bu da k–1 küme merkezlerine olan uzaklıkları hesaplamak için gerekli olan zamanı korur. Bu algoritma toplam çalışma zamanı ve kümelerin kalitesi bakımından CLARA ve orijinal KM algoritmaları ile karşılaştırılmıştır ve ikisinden de daha iyi sonuçlar verdiğin gözlenmiştir [32].

Birçok kümeleme metodu önceden tanımlı kümeleme sayısına veya kesin benzerlik eşik değerine ihtiyaç duyar. Auto-K adında eşik değerine ihtiyaç duyan yeni ve basit bir algoritma geliştirilmiştir. Auto-K, otomatik olarak veri kümesinden uygun küme sayısını seçer. Auto-K' da, k başlangıç küme merkezlerinin sayısı, toplam veri kümesinin sayısından küçüktür. Auto-K' da ilk önce k başlangıç küme merkezler seçilmiştir($1 \leq j \leq k$) . KM' te olduğu gibi k ile veri kümesi kümelenmiştir ve C kümeleme sonuçları elde edilmiştir. Sonra da sırasıyla, her bir C_i kümesi için küme içi benzerlik, olası kümeleme sonuçlarının her biri için küme içi benzerlik, olası kümeleme sonuçlarının her biri için kümeler arası benzerlik ve kümeleme uygunluğu hesaplanmıştır. Bu işlemler tüm veri noktaları için tekrarlanmıştır. Döngü sonunda da, kümeleme uygunluğu değerinin maksimumu alınmıştır ve buna göre en iyi k başlangıç küme merkezleri seçilmiştir. Auto-K' nın hesapsal karmaşıklığı KM metodundan daha yüksektir. KM' in karmaşıklığı O(N)' dir ve Auto-K' nın ise $O(N^2)$' dir. Auto-K' nın hesapsal yüksek karmaşıklığı k için birkaç daha olası adayın seçilmesiyle azaltılabilir. Auto-K ile ilgili diğer bir problemde geniş kümeleme maliyetidir [33].

KM' in hesaplama ölçümünü kötü yapması, küme sayılarının kullanıcı tarafından girilmesi ve aramanın yerel bir en küçük değere eğilimli olmasından dolayı k değerini hızlı bir şekilde tahmin eden X-Means adında bir algoritma geliştirilmiştir. X-Means KM'in her çalıştırılmasından sonra harekete geçer. O anki merkezin alt kümesinin hangisinin veriye daha iyi uyması için bölünmesi gerektiğinin yerel kararını alır. Bölünme kararı Bayes bilgi kıstasına göre yapılır. Algoritma verilen aralığın en düşük sınırına eşit olan k ile başlar ve en üst sınıra varıncaya kadar onların ihtiyaç duyduğu yere merkezleri ekleyerek devam eder. Deneysel sonuçlar, sentetik ve gerçek veriler üzerinde bu algoritmanın KM' ten daha hızlı ve daha iyi çalıştığını göstermiştir [34.

Kümeleme işleminin daha efektif sonuçlar verebilmesi için Barbakh ve Fyfe tarafından bir çalışma yapılmıştır. Bu çalışmada kümeleme algoritmalarının performanslarını ölçmek için farklı performans fonksiyonlarının üzerinde durulmuştur. Başlangıç koşullarına bağlılık göstermeyen 2 yeni algoritma türetilmiştir. Algoritmalar bütün seçilen merkezlerin aynı başlangıç noktasına sahip olması ve küme merkezlerinin verilerden çok uzakta olması şeklindeki iki kıstasa göre karşılaştırılmıştır. Karşılaştırma da KM, KHM ve yeni geliştirilen iki algoritma kullanılmış ve yeni geliştirilen algoritmaları hepsinden daha iyi sonuçlar verdikleri gözlenmiştir [35].

Bir kümeleme işleminde, uygun kümeleme algoritmasının seçilip veri kümesi üzerinde bu algoritmanın uygulanıp kümelerin oluşturulması yeterli değildir. Oluşan kümelerin bir şekilde doğal kümelere benzer olup olmadığının kontrolünün yapılması gerekmektedir. Burada devreye kümeleme geçerlilik teknikleri girmektedir. Kümeleme geçerlilik teknikleri kümeleme işlemi sonucu oluşan kümelerin değerlendirilmesinde kullanılmaktadır. En iyi geçerlilik ölçümüne sahip olan veri bölümlemesini seçmek ve farklı giriş değişkenleri için algoritmayı çalıştırmak uygun küme sayısına karar vermede kullanılan gene yaklaşımlardır. Oluşan kümelerin kalitesini ölçmede kullanılan iki kıstas vardır. Bunlardan biri kümelerin yoğunluğudur (compactness). Bir küme içindeki örüntüler, aynı küme içindeki örüntülere benzer olmalı fakat farklı küme içindeki örüntülerden de farklı olmalıdır. Küme içindeki örüntülerin varyansı kümenin yoğunluğu hakkında bilgi verir. Küme

kalitesini ölçmede kullanılan diğer kıstas ise ayrıklıktır (seperation). Oluşan kümeler birbirinden çok iyi şekilde ayrılmış olmalıdır. Küme merkezleri arasındaki Öklit uzaklığı, kümelerin ne kadar birbirinden ayrı olduğuna dair bilgi verir. Bu iki kıstas küme geçerlilik tekniklerinin temelini oluşturmaktadır. Kümeleme geçerlilik teknikleri geçerlilik indekslerine sahiptir. Dunn indeksi, Davies and Bouldin indeksi, Turi indeksi, Silhouette indeksi gibi indeksler küme geçerliliğinin ölçme de kullanılan indekslerdir. Bu indekslerin amacı kümeler arası uzaklıkları maksimuma çıkarmak ve küme içi uzaklıkları minimuma düşürmektedir [36].

Çalışmanın literatür taramasında kaynak olarak internette yer alan çeşitli bilimsel makalelerden ve konu ile ilgili kitaplardan yararlanılmıştır. Çalışmada kullanılan süsen çiçeği (iris), cam (glass identification), diyabet (Pima Indians Diabetes) ve mamografi (mammographic) veritabanları UCI veri deposundan (UCI Machine Learning Repository) alınmıştır. Bu tez çalışmasında karşılaştırma işlemi yapılacağından geçerliliği kanıtlanmış UCI veri deposundan alınan veritabanları tercih edilmiştir. Veritabanlarının diğer veri depolarından değil de UCI veri deposundan alınmasının nedenleri arasında, veritabanların içindeki verilere ait nitelik ve nitelik değerlerinin çok iyi açıklanması, niteliklere ait istatistiksel sonuçların verilmesi ve genelde verilerin eksik veriler içermemesi verilebilir.

Merkez tabanlı kümeleme algoritmalarının karşılaştırılmasına yönelik yapılmış birçok veri madenciliği çalışmasına literatür taramasında rastlanmıştır. Veri madenciliği çalışmalarında genellikle veriler SPSS ve MATLAB gibi paket programlar aracılığı ile analiz edilmektedir. Bu çalışmada geliştirilen uygulamanın, kullanıcı tarafından kullanılması çok kolaydır ve paket programlardaki kısıtları içermemektedir.

Bu tez çalışması altı bölümden oluşmaktadır. Tezin ikinci bölümünde veri madenciliği hakkında genel bilgiler verilerek giriş yapılmış ve ardından veri madenciliği ile ilgili en fazla kullanılan tanımlara yer verilmiştir. Veri madenciliği tanımlarından sonra, veri madenciliğinin tarihi gelişimi, veri madenciliğinin uygulama alanları ve veri ambarı kavramına değinilmiştir. Ardından veritabanlarında bilgi keşfi ve adımları ayrı iki model üzerinden ayrıntılarıyla açıklanmış ve bilimsel

metot adımları ile veritabanlarındaki bilgi keşfi adımları karşılaştırmalı olarak ele alınıp incelenmiştir. Son olarak veri madenciliği modelleme teknikleri sırayla ele alınıp açıklanmıştır.

Üçüncü bölümde veri madenciliği tekniklerinden kümeleme analizi detaylı olarak incelenmiştir. Kümeleme analizinin tanımı, özellikleri ve kümeleme analizinde kullanılan veri türlerine yer verilmiştir. Kümeleme işleminin adımları sırayla ele alınıp incelenmiştir. Daha sonra da kümeleme algoritmalarının çok fazla olmasının nedenleri irdelenmiştir. Kümeleme analizinin teknikleri ayrıntılı olarak açıklanırken, her bir kümeleme tekniğini içinde bulunan algoritmaların teorik yapısı ve çalışma şekli hakkında bilgiler verilmiştir. Son olarak kümeleme analizinin kullanıldığı alanlara değinilmiştir.

Dördüncü bölümde merkez tabanlı kümeleme yapısı hakkında bilgiler verilerek giriş yapılmış ve ardından merkez tabanlı kümeleme algoritmalarında kullanılan başlangıç yöntemlerine yer verilmiştir. Daha sonra da temel merkez tabanlı kümeleme algoritmaları olan K-ortalama, Bulanık k-ortalama, K-harmonik ortalama algoritmaları sırasıyla örneklerle ele alınıp açıklanmıştır. Ayrıca K-harmonik ortalama ve K-ortalama algoritmalarının özelliklerini barındıran Hibrit 1 ve Hibrit 2 adındaki 2 algoritmada sırasıyla açıklanmıştır.

Uygulamanın anlatıldığı beşinci bölümde, ilk başta UCI veri deposundan alınan süsen çiçeği, cam, diyabet ve mamografi veritabanları ayrıntılarıyla ele alınıp açıklanmıştır. Ardında geliştirilen uygulamaya ait olan arayüzler ayrıntılı şekilde anlatılıp şekilsel olarak tanıtılmıştır. Daha sonra geliştirilen uygulama ile verilerin karşılaştırılacağı kıstaslara ilişkin bilgilere yer verilmiştir. Ardından bu veritabanları üzerinde merkez tabanlı kümeleme algoritmaları olan k-ortalama, bulanık k-ortalama, k-harmonik ortalama, hibrit 1 ve hibrit 2 algoritmaları uygulanmış ve belirtilen kıstaslar doğrultusunda bu algoritmalar birbirleri ile karşılaştırılmıştır

Tezin sonuçlar bölümünde merkez tabanlı kümeleme algoritmalarının belirlenen kümeleme analizi kıstaslarına göre karşılaştırılması sonucu elde edilen bilgilere yer verilmiştir.

2. VERİ MADENCİLİĞİ

2.1 Giriş

Bu bölümde veri madenciliği hakkında genel bilgiler verilerek giriş yapılmış ve ardından veri madenciliği ile ilgili en fazla kullanılan tanımlara yer verilmiştir. Veri madenciliği tanımlarından sonra, veri madenciliğinin geçmişten günümüze kadar olan tarihi gelişimi, veri madenciliğinin uygulama alanları ve veri ambarı kavramına değinilmiştir. Ardından veritabanlarında bilgi keşfi ve adımları ayrı iki model üzerinden ayrıntılarıyla açıklanmış ve bilimsel metot adımları ile veritabanlarındaki bilgi keşfi adımları karşılaştırmalı olarak ele alınıp incelenmiştir. Son olarak veri madenciliği modelleme teknikleri sırayla ele alınıp açıklanmıştır.

2.2 Veri Madenciliği Nedir?

Veri madenciliği, anlamlı örüntüler ve kurallar bulabilmek için büyük miktardaki verinin incelenmesi ve araştırılmasıdır. Veri madenciğinin amacı; veritabanları içindeki gizli bilgiyi keşfetmektir. Veri madenciliği; makine öğrenmesi, istatiksel analiz, modelleme teknikleri ve veritabanı teknojilerini kullanarak veri içinde güçlükle farkedilen örüntüleri ve ilişkileri bulur ve gelecek sonuçların tahminine olanak sağlayan kuralları çıkartır. Veri madenciliği deyimi yanlış kullanılan bir deyim olabileceğinden bilim adamları tarafından buna eş değer başka adlandırmalarda literatüre geçmiştir. Bunlar; veritabanlarında bilgi keşfi (KDD-Knowledge Discovery in Databases), bilgi çıkarımı (Knowledge Extraction), veri ve örüntü analizi (data/pattern anaysis), veri tarama (data dredging), bilgi keşfi (knowledge discovery), veri arkeolojisi (data archaeology), veri avcılığı (data fishing), bilgi üretimi (knowledge creation) ve bilgi hasadıdır (information harvesting). Bu adlandırmalardan veri madenciliği deyimi yerine en çok kullanılanı KDD' dır. Fakat KDD, veriler arasından yararlı bilgileri keşfetme sürecidir ve veri madenciliği KDD sürecinin önemli bir adımıdır. KDD süreci ise veri hazırlama, veri

seçme, veri temizleme ve veri madenciliği sonucu çıkan sonuçların yorumlanması gibi ek adımlarla birlikte veriden türetilen yararlı bilginin elde edilmesi demektir. KDD sürecinin adımları ile birlikte aşağıda ayrıntıları ile açıklanmıştır.

Veri madenciliği ile ilgili birçok tanım yapılmıştır. Bunlardan birkaçı aşağıda verilmiştir:

a) Veri madenciliği veya KDD, veriden üstü kapalı, ilginç, daha önceden bilinmeyen ve potansiyel olarak kullanışlı bilginin çıkarılmasıdır [37].

b) Veri madenciliği, geçerli tahminler yapmak için veri analiz araçlarını kullanarak veri içindeki bilgiyi ve örüntüleri keşfetmek işlemidir [7].

c) Veritabanı içindeki veriden bilgi çıkartmak ve otomatik olarak analiz etmek için bir ya da fazla bilgisayar öğrenme tekniğini çalıştırma işlemi olarak tanımlanmaktadır [38].

d) Veri madenciliği, çoğunlukla temizlenmiş, dönüştürülmüş veriyi giriş verisi olarak alan, algoritmaları kullanarak veri üzerinde araştırmalar yapan ve KDD' nin değerlendirme ve yorumlama adımına örüntüleri çıkış verisi sunan KDD' nin bir adımı olarak tanımlanmaktadır [39].

e) Veri madenciliği, veri sahibine daha anlaşılır ve daha yararlı olacak şekilde veriyi özetletmek ve fark edilemeyen ilişkileri bulmak için veri kümesinin analiz edilmesidir. Veri madenciliği yoluyla türetilen ilişkiler ve özetler genellikle model veya örüntü olarak ifade edilir [40].

İlk veri madenciliği algoritması ticari uygulamalarla birlikte icat edilmiştir. Ticari veri madencileri; istatistik, bilgisayar bilimi, makina öğrenme araştırmalarından ödünç aldıkları teknikleri kullanmışlardır. Belli bir duruma uygulamak için, hangi tekniğin seçileceği; veri madenciliği görevinin özelliğine, eldeki verinin özelliğine, veri madencilerinin tercihlerine ve becerilerine dayanmaktadır [41].

Veri madenciliği büyük oranda modelleri inşa etmekle ilgilenir. Model; giriş topluluğunu belli bir hedefe veya sonuca bağlayan bir algoritma veya kurallar topluluğu olarak tanımlanmaktadır. Regresyon, yapay sinir ağları, karar ağaçları (Decision Trees) ve diğer veri madenciliği teknikleri model oluşturmak için kullanılan tekniklerdir. Doğru koşullar altında, bir model ile belli bir konunun nasıl sonuçlanacağı hakkında bir açıklama sağlanabilmektedir. Modeller ayrıca sayı (score) üretmek içinde kullanılmaktadırlar. Sayı bir modelin bulgularını tek bir rakamla ifade etme yoludur. Sayı, müşterileri en sadık müşteriden en az sadığa veya cevaplama olasılığı en çok olandan en aza olacak şekilde bir listede sıralamak için kullanılabilir.

2.3 Veri Madenciliğinin Tarihsel Gelişimi

Veri madenciliğinin yapıtaşları yapay zeka ve makine öğrenmesini oluşturmak için matematikçilerin, mantıkçıların ve bilgisayar bilginlerinin bir araya geldiği 1950' li yıllarına dayanır.

1960' larda yapay zekâ ve istatistik uzmanları regresyon analizi, maksimum olasılık tahmini, yapay sinir ağları ve lineer sınıflandırma modelleri gibi yeni algoritmalar geliştirmişlerdir. Veri madenciliği sözü bu dönemde ortaya çıkmıştır. Fakat bu dönemde veri madenciliği; veriden zorla ilerleyen ve bulunan örüntülerin istatiksel değerinin olmadığı bir uygulama olarak tanımlanmıştır.

1960' larda IR alanı; kümeleme teknikleri ve benzerlik ölçümlerine kendi katılımını sağlanmıştır ve yine bu dönemlerde bu teknikler metin dokümanlarına uygulanmıştır. Daha sonra bu teknikler; büyük dağıtılmış veri kümeleri ve veritabanlarındaki veriler madenlenirken kullanılmıştır. Veritabanı sistemleri, yapısal verinin sorgulanması ve işlenmesi ile ilgilenir. IR büyük sayıdaki metine dayalı dokümanlardaki bilginin organizasyonu ve yeniden düzeltilmesiyle ilgilenir. 1960' ların sonunda veritabanı sistemleri ve IR paralel olarak gelişmekteydiler.

1971' de Gerard Salton büyük bir yankı uyandıran "Akıllı bilgi çıkarımı sistemleri" üzerindeki çalışmasını yayımlamıştır. Cebire dayalı vektör uzay modelini(VSM)

kullanan IR için yeni bir yaklaşım sunmuştur. VSM modelleri veri madenciliği araçlarında anahtar bir içerik olduğunu kanıtlayacaktı. 1970, 1980 ve 1990 yılları boyunca, AI (yapay zekâ), IR, istatistik ve veritabanı sistemleri gibi disiplerin birleşmesi ve hızlı mikrobilgisayar sistemlerinin varlığı veri analizinde ve çıkarımında yeni yollar açmıştır. Bu zaman esnasında yeni programlama dilleri geliştirilmiş ve genetik algoritmalar, EM algoritmaları, KM kümeleme ve karar ağacı algoritmaları gibi hesaplama teknikleri geliştirilmiştir. İlerleyen zamanlarlarda bu tekniklere yenileri de eklenmiştir.

1990' ların başlangıcında KDD terimi oluşmuş ve ilk KDD çalışması yapılmıştır. Muazzam bir verinin olması, bu muazzam bilgiyle baş edebilecek yeni tekniklere ihtiyarcı doğurmuştur.

1990' larda veri ambarında (data warehouse) gelişmeler görülmüştür. Veri ambarı tek bir şemadan oluşan büyük miktardaki veriyi tanımlamak için kullanılan bir terimdir ve işlemsel veritabanı verisinin yoğunlaştırılması ile oluşturulmuştur. Veri ambarının gelişmesiyle birlikte çevrimiçi analitik işleme (OLAP), karar destek sistemleri, veri dönüştürme ve birliktelik kural algoritmaları da ortaya çıkmıştır.

1990' larda veri madenciliği; bilgisayar disk depolama maliyetinin azalması, işlem gücünün yükselmesi ve veri madenciliğinin yararlarının açık olarak görülmesiyle ilginç yeni bir teknoloji olmaktan çıkmış ve günlük iş hayatının bir parçası olmuştur. İş dünyası müşterilerin hayat döngülerinin tüm aşamalarını yönetmek için veri madenciliğini kullanmaya başlamışlardır. Örneğin; yeni müşteriler kazanmak, var olan müşterilerden daha fazla kazanç sağlamak ve iyi müşterilerin elde tutulması.

Veri madenciliği birçok değişik sektör ve endüstri alanlarında kullanılmaktadır. Bu alanların başında; satışlar, tıp, bilim, mali, eczacılık, pazarlama, internete dayalı şirket ve hükümetler gelmektedir. Mayıs 2004 yılında Federal veri madenciliği raporu yayınlanmıştır. Amerikan Genel Muhasebe Ofisi, çeşitli federal ajanlarda 199 adet veri madenciliği projesinin yürülükte veya planlanmakta olduğunu rapor etmiştir ve bu rapor Matrix ve NSA gibi istihbarat örgütü Ulusal Güvenlik Kurumunun projelerini içermemekteydi [1].

2.4 Veri Madenciliğinin Uygulama Alanları

Veri madenciliğinin birçok uygulama alanı vardır. Bu uygulama alanlarının başlıcaları aşağıda sıralanmıştır:

a) Pazarlama: Müşterilerin satın alma örüntülerini belirlemede, müşterilerin demografik karakterleri arasındaki ilişkileri bulmada, pazar sepeti analizinde, müşteri ilişkileri yönetiminde yaygın olarak kullanılmaktadır.

b) Banka, Finans ve Sigortacılık: Kredi kartı ve sigorta dolandırıcılıklarının tespitinde, çapraz satış, risk derecelendirme, mevcut müşteriyi elde tutma, yeni müşteriler kazanma, maliyetleri azaltma, kayıp ve kaçakları engelleme, müşteri memnuniyetini sağlamada, kredi kartı taleplerinin belirlenmesinde, kredi kartı harcamalarına bağlı olarak müşteri profilinin belirlenmesinde yaygın olarak kullanılmaktadır.

c) Haberleşme: Telekom sektöründe en önemli sorun müşteri kaybıdır. Örneğin; Amerika'nın en büyük kablosuz iletişim sağlayıcısı olan Verizon kaybetme olasılığı yüksek olan müşterilerini ve müşteri kaybına neden olan faktörleri belirleme amaçlı bir veri madenciliği çalışması yapmıştır.

d) Metin Madenciliği: Çok büyük ve anlamsız metin yığınları arasından anlamlı ilişkiler elde etmekte kullanılmaktadır.

e) Biyoloji, Tıp ve Genetik: Bitki türleri ıslahı, gen haritasının analizi ve genetik hastalıkların tespitinde kullanılmaktadır. Ayrıca kanserli hücrelerin tespiti, yeni virüs türlerinin keşfi ve sınıflandırılması, ameliyatlarda yüksek risk faktörünün sınanması, hasta verilerinin yaş, cinsiyet, ırk ve tedavi yöntemi gibi faktörlere göre sınanması, hasta sağlığı açısından geriye dönük faktörterin sınanması, tedavi yönteminin geliştirilmesinde ve fizyolojik parametrelerin değerlendirilmesinde ve analizinde yaygın olarak kullanılmaktadır. Tıp alanında dünya çapıda çok sayıda başarılı uygulama örneği mevcuttur. Örneğin; San Francisco Kalp Enstitüsü hasta sonuçlarının iyileştirilmesi, hastanın hastanede kalma süresinin azaltılması amacıyla

bir çalışma başlatmıştır. Böylelikle kurum bünyesinde toplanan verilerden hastanın geçmişine ait veriler, laboratuar verileri, kollestrol verileri, diğer medikal verileri bilgiye dönüştürmüştür.

f) Devlet Uygulamaları: Kaynakların doğru olarak kullanımı sağlama ve planlama, rastlantısal olayların çözümüne dair izleri keşfetme ve olası güvenlik sorunlarını eş zamanlı olarak tespit edebilme ve çözüm üretebilme, vergi ile ilgili yolsuzlukları belirleme, sağlık ödemeleri, suistimal ve israfları belirleme ve milyonlarca dolarlık zararı engelleme, emniyet birimleri için suç istatistiklerine dair çevrimiçi raporlama, hangi profildeki insanların ne tür suçlara meyilli olduklarını belirleme, kamu güvenliğini sağlamak amacıyla güvenlik problemlerini önceden tahmin etmede ve eş zamanlı suç engelleme politikaları oluşturmada veri madenciliğinden yararlanılmaktadır.

g) Kimya: Kimyasal moleküllerin keşfi ve sınıflandırılmasında, yeni ilaç türlerinin keşfinde kullanılmaktadır.

h) Yüzey Analizi ve Cografi Bilgi Sistemleri: Bölgelerin coğrafi özelliklerine göre sınıflandırılması, kentlerde yerleşim yerleri belirleme, kentlerde suç oranı, zenginlik, yoksulluk, köken belirleme, kentlerde yerleştirilecek posta kutusu, otomatik para makinaları, otobüs durakları gibi hizmetlerin konumlarının espitinde kullanılmaktadır.

i) Uzay Bilimleri ve Teknolojisi: Gezegen yüzey şekilleri ve gezegen yerleşimleri, yeni galaksiler keşfi, yıldızların konumlarına göre gruplandırılmasında kullanılmaktadır.

j) Görüntü Tanımı ve Robot Görüş Sistemleri: Çeşitli sensörler aracılığıyla tespit edilen görüntülerden yola çıkarak engel tanıma, yol tanıma, yüz tanıma, parmak izi tanıma gibi tekniklerde kullanılmaktadır.

k) Uzay Bilimleri ve Teknolojisi: Gezegen yüzey şekillerinin ve gezegen yerleşimleri, yeni galaksiler keşfi, yıldızların konumalarına göre gruplandırılmasında kullanılmaktadır.

l) Meteoroloji ve Atmosfer Bilimleri: Bölgesel iklim, yağış haritaları oluşturma, hava tahminleri, ozon tabakası deliklerinin tespiti, çeşitli okyanus hareketlerinin belirlenmesinde kullanılmaktadır.

m) Sosyal Bilimler ve Davranış Bilimleri: Kamuoyu yoklamalarını inceleme, genel eğilim belirleme, seçim öngörüleri oluşturmada kullanılmaktadır.

n) Metin Madenciliği: Çok büyük ve anlamsız metin yığınları arasından anlamlı ilişkiler elde etmekte kullanılmaktadır.

o) İş ve Elektronik Ticaret Verileri: Geri ofis, ön ofis ve ağ uygulamaları iş süreçleri sırasında geniş çaplarda veri üretirler. Bu veriyi karar verme mekanizmalarında efektif olarak kullanmak, ilgili ticari kuruluşun temel yapıtaşlarından olmalıdır.

p) Bilimsel, Mühendislik ve Sağlık Bakım Verileri: Günümüzde bilimsel veriler, iş sahası verilerinden daha karmaşık hale gelmişlerdir. Buna ek olarak, bilim adamları ve mühendisler uygulama sahası bilgilerini kullanarak simülasyon ve sistem kullanımının arttırılması hedefindedirler.

r) Web Verileri: Veri madenciliği tekniklerinin web üzerinde uygulanmasıyla web'de bulunan veriden faydalı bilginin keşfedilmesi ve yorumlanması için web madenciliği ortaya çıkmıştır. İnternette bilginin artması ve web sitelerinin etkinleştirilmesi ihtiyacı web madenciliğinin daha da popüler olmasını sağlamıştır.

2.5 Veri Ambarı

Bir veri ambarı, bir işletmenin kullanmakta olduğu veritabanlarından ayrı olarak tutulan ve işletmenin değişik bölümleri tarafından toplanan bilgilerin, ileride değerlendirilmek üzere arka plandaki sistemde birleştirilmesinden oluşan geniş

ölçekli veri deposudur. Veri ambarı, özneye dayalı, bütünleşmiş, zaman dilimli ve yöneticinin karar verme işleminde yardımcı olacak biçimde toplanmış olan değişmeyen veriler topluluğu şeklinde de tanımlanmaktadır. Bu özellikleri yakından inceleyecek olursak;

a) Özneye Dayalı: Bir veri ambarı, tüketici, tedarikçi firma, ürün ve satış gibi önemli özneler etrafında kurulur. Veri ambarı bir işletmeye ait olan gündelik işlere yoğunlaşmak yerine karar destek sürecinde karar vericilere yardımcı olabilecek veriye ait modelleme ve analizler üzerine yoğunlaşır [37].

b) Tümleşik(Integrated): Veri ambarları ilişkisel veritabanları, düz metin dosyaları ve çevrimiçi işlem kayıtları gibi çeşitli farklı türde veri kaynaklarının birleştirilmesinden oluşur [37].

c) Zaman dilimli: Veri ambarı içine veriler belirli zaman dilimleri çerçevesinde eklenir [37].

d) Değişmeyen: Veri ambarına eklenen veriler işlemsel veritabanları olduğu gibi sürekli güncellenmezler [37].

2.6 Veritabanlarında Bilgi Keşif Adımları

Veri madenciliği, KDD işleminin adımlarından biridir. KDD işleminin adımları farklı kişiler tarafından farklı modellerle ifade edilmiştir. Han' ın sunduğu modelde KDD işlemi yedi adımda meydana gelmektedir. Bu adımlar sırasıyla veri temizleme, veri birleştirme, veri seçme, veri dönüştürme, veri madenciliği, örüntü değerlendirme ve bilgi sunumudur. KDD sürecindeki adımlar Şekil 2.1' de görsel olarak ifade edilmiştir. Han' ın sunduğu modeldeki KDD sürecinde yer alan adımlar açıklamaları ile aşağıda belirtilmiştir [37]:

1) Veri Temizleme (Data Cleaning): Gerçek hayat kullanılan veritabanları içindeki veriler kirlenmeye, eksik olmaya ve tutarsız olmaya eğilimlidirler. Bu nedenle verilerin kullanılmadan önce bazı ön işlemlerden geçmeleri gerekir. Ön işlemlerden

geçen veriler üzerinde veri madenciliği algoritmalarının uygulanması ile daha kalite sonuçlar elde edilir. Bu ön işlemlerden biri veri temizlemedir. Veri temizleme ile veritabanlarındaki eksik, tutarsız ve gürültü verilerin giderilir.

2) Veri Birleştirme (Data Integration): Veri temizlemeden sonra veri birleştirme işlemi uygulanır. Veri birleştirme, çeşitli kaynaklardan gelen verilerin tek bir veri ambarı altında toplanmasıdır.

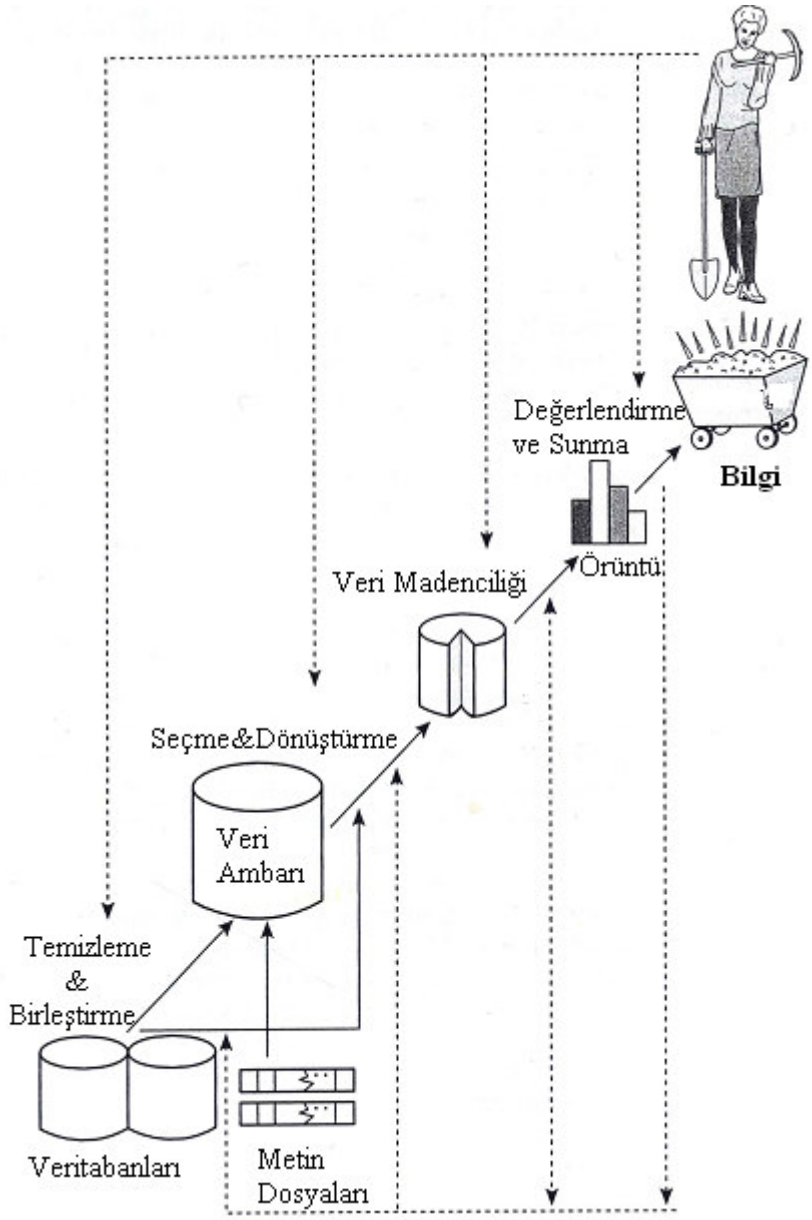
3) Veri Seçme (Data Selection): Veritabanlarından üzerinden işlem yapılacak olan veri seçilir ve veri türleri bu aşamada belirlenir.

4) Veri Dönüştürme (Data Transformation): Bu aşamada veriler veri madenciliği algoritmalarının uygulanabilmesi için uygun bir forma dönüştürülür. Veri dönüştürme işlemi veri düzeltme, birleştirme, genelleştirme ve normalleştirme gibi işlemlerin bir ya da birkaçını içerebilir.

5) Veri Madenciliği (Data Mining): Bu aşamada, anlamlı örüntüler elde edebilmek için veri üzerinde veri madenciliği algoritmaları uygulanır. Sınıflandırma, kümeleme algoritmaları gibi veri madenciliği algoritmaları kullanılarak yararlı bilgi keşfedilmesi sağlanır.

6) Örüntü Değerlendirme (Pattern Evaluation): Elde edilmiş olan bilginin basitlik, geçerlilik, yararlılık ve yenilik gibi bazı ölçüm değerlerine göre değerlendirildiği aşamadır.

7) Bilgi Sunumu (Knowledge Presentation): Bu aşamada, çeşitli görselleştirme ve bilgi sunum araçları kullanılarak elde edilmiş olan bilginin kullanıcıya sunumu gerçekleştirilir.



Şekil 2.1: Han' a göre KDD işleminin adımları [37].

Bir diğer KDD modeli Roiger ve Geatz tarafından sunulmuştur. Bu modeldeki KDD işlemi de 7 adımdan meydana gelmektedir. Her iki modelde birbirine benzemesine rağmen adımlar birbirinden farklıdır. Aşağıda Roiger ve Geatz' ın sunduğu modeldeki adımlar Şekil 2.2' de gösterilmiş ve aşağıda da ayrıntılarıyla açıklanmıştır[38]:

1) Hedefleri Tanımlama (Goal Identification): Bu adımda problem ve ulaşılmak istenen hedefler tanımlanır. Kullanılacak olan veri madenciliği aracının seçimi yapılır.

2) Hedef Veri Kümesini Oluşturma (Creating a target data set): Bu aşamada, bilgi keşfi araçları ve bir ya da daha fazla uzmanın yardımı ile analiz edilecek veri kümesi seçilir.

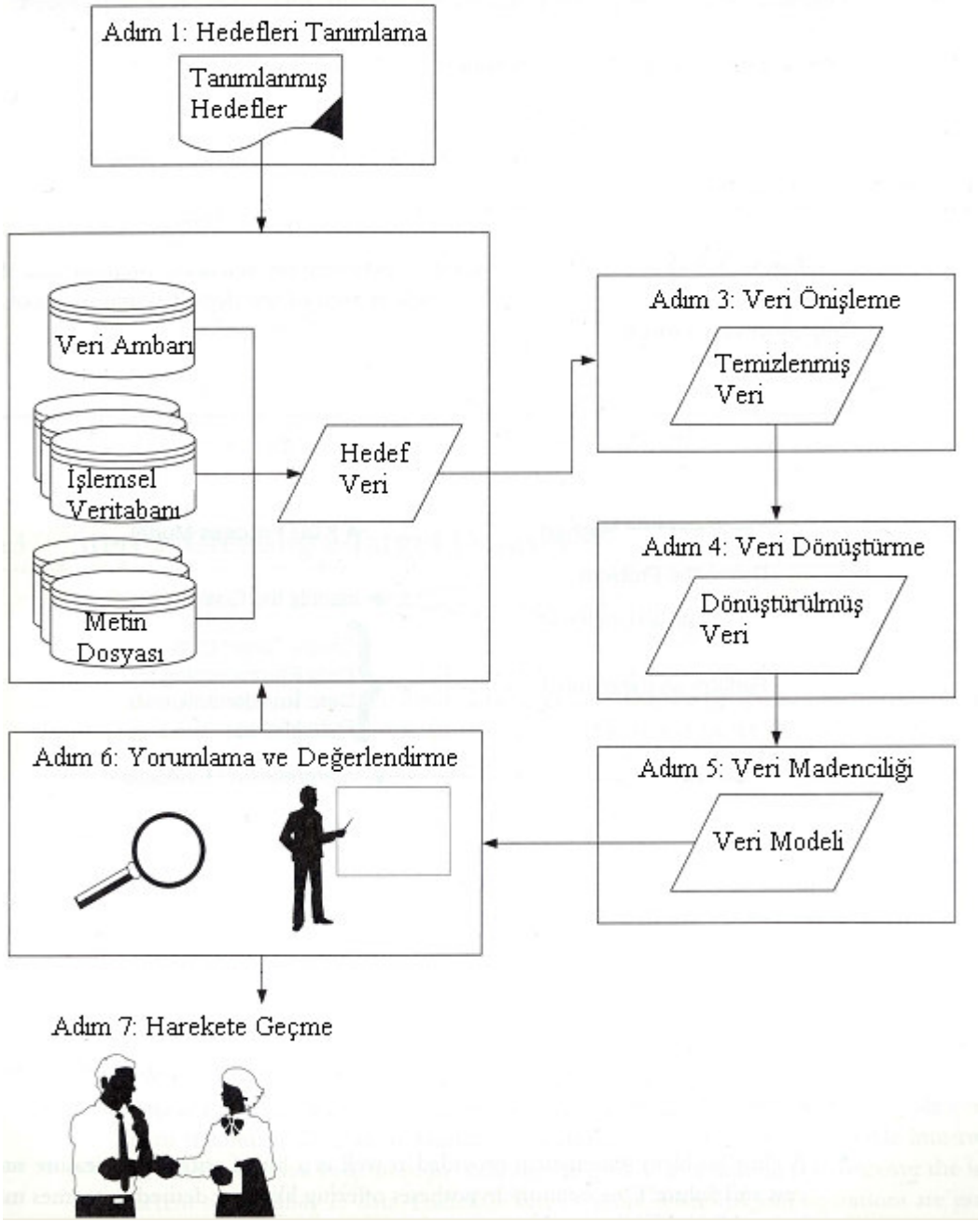
3) Veri Önişleme (Data Preprocessing): Veritabanındaki veya veri ambarlarındaki eksik, tutarsız ve gürültülü veriler giderilir. Bu işlem veri temizleme olarak adlandırılmaktadır.

4) Veri Dönüştürme (Data Transformation): Bu aşamada veriler veri madenciliği algoritmalarının uygulanabilmesi için uygun bir forma dönüştürülür. Oluşturulan hedef veri içine nitelikler ve örnekler eklenir ya da çıkarılır. Bunun nedeni ise bazı veri madenciliği algoritmalarının birden fazla nitelik içeren veri kümelerini analiz edememesi bazılarının da örneklerin çok olması durumunda çalışma esnasında sorunlar yaşamasıdır. Bazı algoritmalar kategorik veriler üzerinde çalışamadığından ilgili verilerin sayısal forma dönüştürülmesi gerekir. Bu aşamada, kullanılan veri madenciliği algoritmasına bağlı olarak veri tipinin uygun forma dönüştürülmesi sağlanır.

5) Veri Madenciliği (Data Mining): Bu aşamada, anlamlı örüntüler elde edebilmek için veri üzerinde veri madenciliği algoritmaları uygulanır.

6) Yorumlama ve Değerlendirme (Interpretation and Evaluation): Yararlı ve ilginç örüntülerin keşfedilip keşfedilmediğine karar vermek için beşinci adımdan elde edilen çıkış incelenir. İncelenme sonucunda, yeni nitelikler ve örneklerin kullanılarak bir önceki adımların tekrar edilip edilmemesi ile ilgili kararlar alınır.

7) Harekete Geçme (Taking Action): Keşfedilen bilgi yararlı ise, bilgi uygun problemlere direkt olarak uygulanır.



Şekil 2.2: Roiger ve Geatz göre KDD işleminin adımları [38].

KDD veri madenciliğine bilimsel metot uygulaması olarak tanımlanabilir. Bilimsel metot 4 adımlı bir işlem olarak tanımlanmaktadır. Bu adımlar çözülecek problemi tanımlamak, hipotezi formülleştirmek, hipotezi çürütmek ya da doğruluğunu kanıtlamak için bir veya birden fazla deney yapmak, çıkan sonuçları düzenlemek ve yararlı sonuçlar elde etmek olarak sıralanabilir. Tablo 2.1' de bilimsel metot ve KDD işleminin birbirine denk düşen adımlarının karşılaştırılması yer almaktadır.

Tablo 2.1: Bilimsel Metot ve KDD işleminin karşılaştırılması [38].

Bilimsel Metot	**KDD İşlemi**
Problemi tanımlamak	Hedefi tanımlamak
Hipotezi formülleştirmek	Hedefi tanımlamak
Deney yapmak	Hedef veriyi yaratmak Veri önişleme Veri dönüştürme Veri madenciliği
Sonuçları düzenlemek	Yorumlama/değerlendirme
Kullanıma hazır hale gelen sonuçları elde etmek	Harekete geçme

2.7 Veri Madenciliği Modelleme Teknikleri

Veri madenciliği alanında kullanılan birçok model vardır. Bu modellerden bir tanesi Han' a göre iki kategori altında toplanan tanımlayıcı (descriptive) ve öngörüsel (predictive) modellerdir. Tanımlayıcı veri madenciliği, veritabanı içindeki verinin genel özelliklerinin ortaya çıkarır. Öngörüsel veri madenciliği, gelecekle ilgili tahminlerde bulunmak için geçerli olan veri üzerinde sonuçlar çıkarmaya çalışır. Han' ın modelini kullananlar bile hangi kategorinin hangi modellin altında olduğunun kararını verememişlerdir. Bu kategoriler tanımlama ve ayrımlama (characterization and discrimination), birliktelik analizi (association rules), sınıflandırma ve öngörü (classification and prediction), kümeleme analizi (cluster analysis), sıradışılık analizi (outlier analysis) ve gelişimsel analizdir (evolution analysis). Önerilen bir diğer veri madenciliği modellerinden biri de Berry ve Linoff tarafında ileriye sürülmüştür. Buna göre veri madenciliği yönetilmiş (directed) ve yönetilmemiş (undirected) olmak üzere iki kategoriye ayrılmıştır. Yöneltilmiş veri madenciliği, gelir gibi belirli hedef alanlarını açıklamaya, sınıflamaya ve belirli hedef değişkenlerinin değerlerini bulmaya çalışır. Yöneltilmemiş veri madenciliği, belirli hedef alanları veya önceden tanımlanmış sınıf topluluklarını kullanmadan kayıtlar arasındaki benzerlikleri veya örüntüleri bulmaya çalışır. Veri madenciği tekniklerinden olan sınıflandırma, kestirim ve tahmin yönetilmiş veri madenciliğine

örnektir. Benzer gruplama ve kümeleme teknikleri yönetilmemiş veri madenciliğine örnektir. Tanımlama ve belgeleme tekniği ise yönetilmiş veya yönetilmemiş veri madenciliğinden ikisinden birine girmektedir [41]. Bu tezde, aşağıda detaylı olarak anlatılan Berry ve Linoff kategorilerine yer verilmiştir.

2.7.1 Sınıflandırma (Classification)

Sınıflandırma veri madenciliğinde en yaygın olarak kullanılan tekniklerden biridir. İnsan doğası gereği dünyayı anlamak ve iletişim kurmak için çevresindeki hemen herşeyi sınıflandırmaktadır. İnsanlar, yaşayan varlıkları filum, tür ve cinse; maddeleri elementlere; köpekleri cinslere; insanları ırklara ayırmışlardır. Sınıflandırma yeni sunulan nesnenin özelliklerini incelemeyi ve bu nesneyi önceden tanımlanmış sınıflar kümesine atamayı içerir. Sınıflandırılacak nesneler veritabanı tablosundaki veya bir dosyadaki kayıtlardan alınmaktadır. Sınıflandırma, sınıf koduyla yeni kolonların eklenmesinden oluşur [41].

Sınıflandırma iki adımdan oluşmaktadır. Birinci adımda sınıflar iyi bir şekilde tanımlanır ve tahmin için kullanılacak bir model üretilir. İkinci adımda bu model sınıflandırılmamış veriye uygulanarak sınıflar tahmin edilir. Tahmin edilen değerler sayısal ya da kategorik değerler olabilirler. Ama sınıflandırma işlemi gerçekleştirilirken üzerinde işlem yapılacak olan eğitim verisi değerleri sayısal değerler olmalıdır [41].

Sınıflandırma işlemine; kredi başvurularının düşük, orta ve yüksek riskli olarak sınıflandırılması, web sayfasında gösterilecek içeriklerin belirlenmesi, hangi telefon numaralarının faks makinelerine karşılık geldiğinin belirlenmesi, dolandırıcı sigorta vaadlerinin belirlenmesi, kredi kartı sahtekârlıklarının tespit edilmesi ve endüstri kodlarının tayini ve serbest metin iş tanımına göre iş atamaları örnek olarak verilebilir. Karar ağaçları, yapay sinir ağları (artifical neural networks), k-en yakın komşu (k-nearest neighbour), genetik algoritmalar (genetic algorithms), katı küme yaklaşımı (rough set approach), bulanık küme yaklaşımı (fuzzy set approach) teknikleri başlıca sınıflama teknikleridir.

2.7.2 Kestirim (Estimation)

Sınıflandırma ile kestirim birbirine çok benzemektedir. Sınıflandırama, evet veya hayır; kızamık, kızamıkçık veya suçiçeği gibi ayrık sonuçlarla uğraşmaktadır. Kestirim ise sürekli değerlendirilmiş sonuçlarla uğraşır. Kestrim, bilinmeyen sürekli değişkenler(gelir, yükseklik, kredi kartı dengesi) için bir değer bulur. Sınıflandırma ile kestirim arasındaki belirgin fark; sınıflandırma sınıfsal değerlerle ilgilenirken, kestirim rakamsal değerlerle ilgilenir. Regresyon modelleri ve nöron ağları kestirim işlemine iyi uyarlar [41].

Uygulamada, kestirim sınıflama işi için kullanılır. Örneğin; kayak ayakkabısı üreticisi bir firmaya reklam alanı satmak isteyen bir kredi kartı şirketi, tüm kart sahiplerini kayakçılar ve kayakçı olmayanlar diye iki sınıfa ayıran bir sınıflama modeli oluşturabilir. Başka bir yaklaşım ise her kart sahibine bir kaymaya eğilim skoru atayan bir model olusturmaktır. Bu değer 0' dan 1' e kadar bir değer olabilir ve bu da kart sahibinin bir kayakçı olma olasılığının tahminidir. Sınıflama görevi bir eşik skorun belirlenmesidir. Eşik değerinin üstünde veya eşik değerine eşit bir skora sahip herkes kayakçı olarak sınıflandırılır. Eşik değerin altındakiler kayakçı olmayanlar olarak sınıflandırılır [41].

Kestirim yaklaşımının, bireysel kayıtların tahmine göre sıralanabildiğinden büyük bir avantajı vardır. Bunun önemini görmek için kayak ayakkabısı üretici firmanın reklam için 500.000 adetlik bir bütçe ayırdığını hayal edelim. Eğer sınıflandırma yaklaşımı kullanırsak ve kredi kartı müşterilerinden 1.5 milyon kayakçı tanımlandığını varsayarsak, algoritmamız havuzdan (1.5 milyon kişiden) raslantısal olarak seçilen 500.000 kişinin kredi kartı hesap zarfına bu reklamı koyacaktır. Kestirime göre ise her kart sahibini bir kaymaya eğilim skoru olacak ve reklâm kayakçı olma olasılığı en fazla olan 500.000 adaya gönderilecektir [41].

Kestirim işlemine; bir ailedeki çocuk sayısının tahmini, bir ailenin toplam gelirinin tahmini, bir müşterinin yaşam kalitesinin tahmini, bir kişinin dengeli aktarım talebine cevap verip vermeyeceğinin olasılık tahmini örnek olarak verilebilir.

2.7.3 Tahmin (Prediction)

Tahmin, kestirim ve sınıflandırma ile aynıdır. Tahminde kayıtların sınıflanması, gelecek bir davranışı veya gelecek bir değeri tahmin etmeye göre yapılır ve bu yönüyle kestirimden ve sınıflandırmadan ayrılır. Tahmin tekniğinde, sınıflandırmanın doğruluğunu test etmenin tek yolu bekleyip görmektir [41].

Sınıflandırma ve kestirim için kullanılan tüm teknikler, eğitim örnekleri kullanarak tahmine uygun hale getirilebilir ve tahminde kullanılabilir. Tarihsel veri, günümüzde gözlenen davranışı açıklayan model oluşturmak için kullanılır. Tarihsel veriden oluşturulan model, günümüz verisine uygulandığında gelecekte gözlenecek davranışın tahmini yaratılmış olur. Bu durumda, mevcut durum; daha önce olmuş olayları, yeni durum ise gelecekte olacak olayları tanımlar [41].

Tahmin metoduna; kredi kartı müşterisi hesap transfer teklifini kabul ettiğinde transfer edilecek hesabın büyüklüğünün tahmini, 6 ay içinde hangi müşterinin ayrılacağının tahmini, hangi telefon abonesinin üç hatlı arama, sesli mesaj gibi ücret ilavesi gerektiren servisler sipariş edeceğinin tahmini gibi örnekler verilebilir [41].

2.7.4 Benzer Gruplama (Affinity Grouping)

Benzer gruplamasının görevi; hangi nesnelerin birlikte olduklarını ve birbirlerini nasıl etkilediklerini bulmaktır. Bir süpermarkette alışveriş arabasında hangi malzemelerin beraber alındığını belirlemek benzer gruplamaya örnek olarak verilebilir. Market zincirleri benzer gruplamayı; bir dükkânın rafını veya bir katologtaki nesneleri düzenlemek için kullanırlar. Müşteriler tarafından daha çok birlikte alınan ürünler aynı yere yerleştirilmektedir ve bu da müşterileri bir ürünü aldığında onunla ilişkili diğer ürünüde almak için teşvik etmektedir [41].

Benzer gruplama, ayrıca karşıt-satış fırsatlarını tanımlamak ve ürünleri ve servisleri gruplayarak göze hoş gelen çekici paketler oluşturmak için kullanılabilir. Cep partner, cep ögrenci gibi servis paketleri örnek olarak verilebilir [41].

Benzer gruplama, veriden kurallar üretmek için basit yaklaşımlardan biridir. Eğer iki nesne yeterince birlikte alınıyorsa bunlar için birliktelik kuralları(association rules) oluşturabilir. Birliktelik kuralına örnek verecek olursak; kedi maması alan kişiler P1 olasılığı ile kedi atık kovasıda alırlar, kedi atık kovası alan kişiler P2 olasılığı ile kedi mamasıda alırlar.

2.7.5 Kümeleme (Clustering)

Kümeleme, heterojen olan büyük bir grubu daha homojen alt gruplara veya kümelere ayırma işlemidir. Kümeleme işleminde amaç; küme içi benzerliğin maksimum ve kümeler arası benzerliğin ise minimum yapmaktır. Küme içindeki benzerliğin mükemmelliği ve kümeler arasındaki farklılığın mükemmelliği kümenin daha iyi ve daha açık olmasını sağlar. Kümeleme, denetimsiz sınıflandırma yönetimidir. Kümeleyi, sınıflandırmadan ayıran en önemli özellik kümelemenin önceden tanımlanmış sınıflara bağlı olmamasıdır. Denetimli sınıflandırma yönteminde ise yeni, sınıflandırılmamış veri önceden oluşturulan var olan sınıflardan uygun olanına yerleştirilmektedir. Kümeleme de önceden tanımlanmış bir sınıf yoktur. Kayıtlar kendi aralarındaki benzerliklere göre gruplandırılırlar. Kümeleme ile ortaya çıkan kümelerin ne anlam taşıdığını belirlemek kullanıcıya bağlıdır.

Kümeleme genellikle veri madenciliği ve modelleme tekniklerinde başlangıç olarak kullanılır. Market bölümlenmesinde kümeleme ilk adım olabilir. ” Hangi tür promosyonlara müşteriler en iyi cevabı verir ya da hangi tür promosyonlar müsterileri alışverişe yönlendirir? “ sorusuyla uğraşmak yerine, önce müşteriler benzer alışveriş alışkanlıklarına göre alt kümelere bölünür ve ondan sonra hangi tür promosyonların her bir küme için en iyi sonuç sağladığı sorusu sorulur [41].

2.7.6 Tanımlama ve profil oluşturma (description and profiling)

Veri madenciliğinin amacı; bazen verinin üretmiş olduğu insanlar, ürünler ve işlemler hakkındaki bilgimizi ve anlayışımızı arttırmak için karmaşık bir veri tabanında neyin olup bittiğini tanımlamaktır. Yeterince iyi tanımlanmış bir davranış kendisi için iyi bir açıklama sunar. İyi bir tanımlama, bir açıklamayı nereden

aramaya başlayacağımızı gösterir. Amerikan politikasındaki ünlü cinsiyet ayrılığı "kadınlar demokratları erkeklerden daha çok destekliyorlar" olayı basit bir tanımlamamın nasıl ilgileri üstüne topladığını göstermektedir. Bu tanımlamadan dolayı gazeteciler, sosyologlar, ekonomistler ve politika bilimcileri bu konu hakkından daha ileri çalışmalar yapmışlardır.

3. KÜMELEME ANALİZİ

3.1 Giriş

Bu bölümde ilk başta kümeleme analizi ne olduğuna dair genel bilgiler verilerek giriş yapılmış ardından kümeleme analizi terimi yerine bugüne kadar kullanılmış diğer adlandırmalara yer verilişti r. Ayrıca kümeleme işleminin kullanım alanlarına değinilmiş ve örnekler verilerek kümeleme işlemi açıklanmıştır. Daha sonra kümeleme analizinin sahip olması gereken belli başlı özelliklere ve kümeleme analizi veri türlerine değinilmiştir. Ardından kümeleme işleminin adımları sırayla ele alınıp açıklanmış ve birçok kümeleme algoritmasının ortaya çıkmasının nedenleri üzerinde durulmuştur. Daha sonra kulanı amacına ve kullanılan veri türüne göre kümeleme metotları sınıflandırılmış ve her bir kümeleme metodu altındaki algoritmalar sırasıyla ela alınıp ayrıntılarıyla anlatılmıştır. Son olarak kümeleme analizinin kullanım alanları belirtilerek her biri ayrıntılı bir şekilde açıklanmıştır.

3.2 Kümeleme Analizi Nedir?

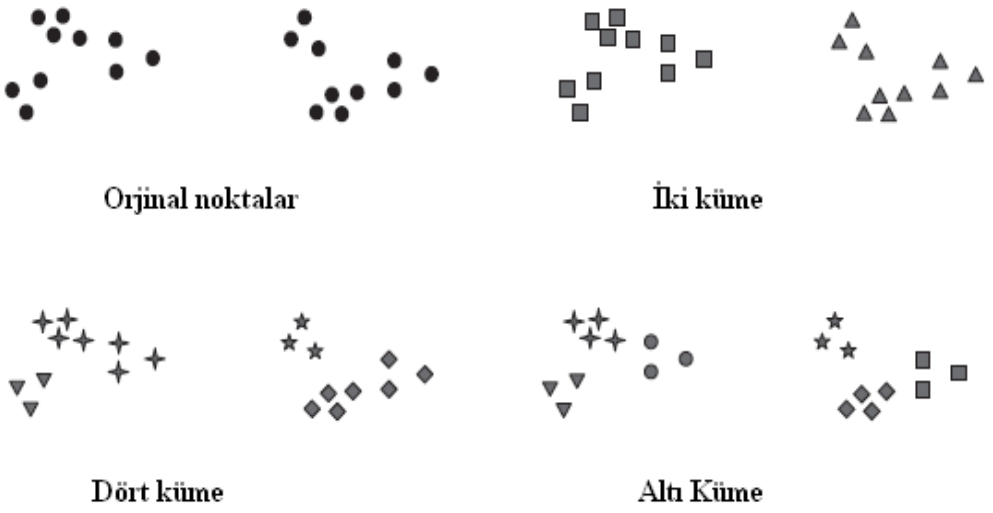
Kümeleme analizi, yakınlıkları ve nesneleri tanımlayan veri içinde bulunanan bilgiye dayanarak veri nesnelerini gruplara ayırma işlemidir. Bu grupların her birine küme denir. Kümeleme analizi literatürde kümeleme adıyla da geçmektedir. Kümelemede amaç; grup içindeki nesnelerin benzer olması ve bu nesnelerin diğer gruplar içindeki nesnelerden farklı ve başka olmasıdır. Grup içindeki benzerliğin (similarity) mükemmelliği ve gruplar arasındaki farklılığın (dissimilarity) mükemmelliği kümenin daha iyi ve daha açık olmasını sağlar [42].

Kümeleme analizinin psikoloji, biyoloji, istatistik, tıp ve mühendislik gibi bilim dalları ile ilişkili olması onun daha da gelişmesini sağlamıştır. Kümeleme analizinin doğal olarak birçok farklı adı ortaya çıkmıştır. Kullanılan adların başlıcaları; sayısal taksonomi (numerical taxonomy), otomatik sınıflandırma (automatic classification),

tipolojik analiz (typological analysis), denetimsiz sınıflandırma (unsupervised classification), veri parçalama (veri segmentation), kümeleme ve veri bölme (data partition)' dır. Bu adlandırmalardan en çok kullanılan kümeleme ve kümeleme analizi olmuştur [42].

İnsanlar yaratışları gereği karmaşık konuları anlamaya çalışırken daha basit anlaşılacak şekilde konuları küçük parçalara ayırmaya çalışırlar. Birisinden orman içindeki ağaçların renklerini tanımlaması istendiğinde, cevap büyük olasılıkla yapraklarını döken ve dökmeyen ağaçlar arasında veya yaz, kış, sonbahar ve ilkbahar arasında olma durumuna göre değişiklik gösterecektir. Benzer renklere sahip ağaçların kümelerini oluşturmak için en iyi kullanılan faktörler, yaş ve yükselikten daha çok orman ile ilişkili yüzlerce değişken, mevsim ve yaprak çeşitleridir. İnsanlar bu faktörleri tahmin etmek için ormanlar hakkında yeterince bilgiye sahiptirler. Yapraklarını döken ağaçlar kışın yapraklara sahip değildir. Bu yüzden bu ağaçlar kışın kahverengi olmaya eğilimlidir. Yapraklarını döken ağaçların yaprakları tipik olarak sonbaharda kırmızı, turuncu ve sarı olacak şekilde renk değiştirirler. Bu bilgilere dayalı olarak ağaçlar renklerine göre gruplandırılabilmektedirler. Çok küçük çoçuklar bile sürekli bilinçaltı kümeleme şemalarını geliştirerek kedi ve köpekleri, hayvan ve bitkileri nasıl ayıracağını öğrenirler [41].

Kümeleme analizi, veri nesnelerini gruplara ayırmada kullanılan diğer tekniklerle ilişkilidir. Kümeleme sınıflandırmanın bir çeşidi olarak düşünülebilir. Kümeleme, denetimsiz sınıflandırma yöntemidir. Denetimli sınıflandırma yönteminde veriler önceden sınıflandırılmıştır. Bu yöntemde yeni ve hangi sınıfta olacağı bilinmeyen veri var olan sınıflardan uygun olanına yerleştirilmektedir. Denetimsiz sınıflandırma yönteminde yeni ve hangi sınıfta olacağı bilinmeyen veri herhangi bir özelliğe dayalı olmadan sadece eldeki veriler kullanılarak anlamlı alt kümeler oluşturacak şekilde gruplandırılmaktadır. Şekilde 3.1' de 20 tane nokta vardır ve kümeler içine onları bölmenin 3 farklı yolu vardır. İşaretlerin şekilleri küme üyeliklerini belirtir. Sırasıyla veri kümesi 2, 4 ve 6 parçaya bölünmüştür. Bu şekil, kümeleme tanımının kesin olmadığını ve en iyi tanımın verinin doğasında ve istenen sonuçlara bağlı olduğunu gösterir.

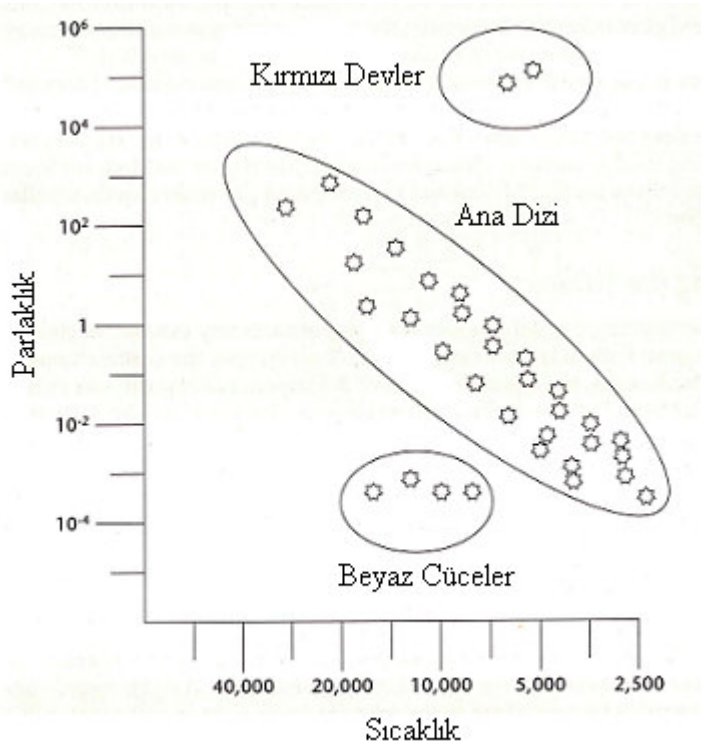


Şekil 3.1: Aynı noktalar kümesini kümelemenin farklı yolları [42].

Veri kümeleme çok hızlı bir gelişim içindedir. Uygulama alanları hızlı bir şekilde artmaktadır. Analiz edilecek veri gün geçtikçe sürekli arttığı için çok kullanılacak bir yöntemdir. Kümeleme ile seyrek ve yoğun alanlar tanımlanabilir ve sonuçta veri nitelikleri arasındaki ilginç ilişkiler ve dağınık örüntüler keşfedilebilir. Kümeleme örüntü tanımlama (pattern recognition), veri analizi (data analysis), görüntü işleme (image processing), astronomi, kıyafet tasarımı ve pazar araştırması (market research), tıp, iklim gibi birçok alanda kullanılmaktadır. Bu alanlardan astronomi ve kıyafet tasarımıyla ilgili örnekler aşağıda belirtilmiştir.

20. yüzyılın başlarında, astronomlar yıldızların parlaklıklarıyla sıcaklıklıkları arasındaki ilişkiyi açıklamaya çalışıyorlardı. Şekil 15.1' de dikey ölçümün güneşin parlaklığını, yatay ölçümün ise Kelvin derecesinde yüzey sıcaklığını göstermekte olduğu bir şekil çizmişlerdir. Enjar Hertzsprung ve Norris Russell adındaki 2 astronot yıldızların dağılım çiziminde 3 kümeye düştüğünü gözlemlemişlerdir. Bu gözlem daha ileriki çalışmaların yapılamasına ön ayak olmuş ve bu 3 kümenin yıldızların hayat döngülerinin farklı aşamalarını gösterdiği ortaya çıkmıştır. Parlaklık ve sıcaklık arasındaki ilişki her küme içinde sabittir. Fakat kümeler arasındaki ilişki farklıdır. Çünkü ışığı ve ısıyı üreten işlemler farklıdır. Yıldızların %80' i hidrojeni helyuma çevirerek nükleer füzyonla enerji üretirler. Bu yıldızlar ana dizi(Main Sequence) adında olan yıldızlardır. Bu tüm yıldızların aktif hayatlarını nasıl

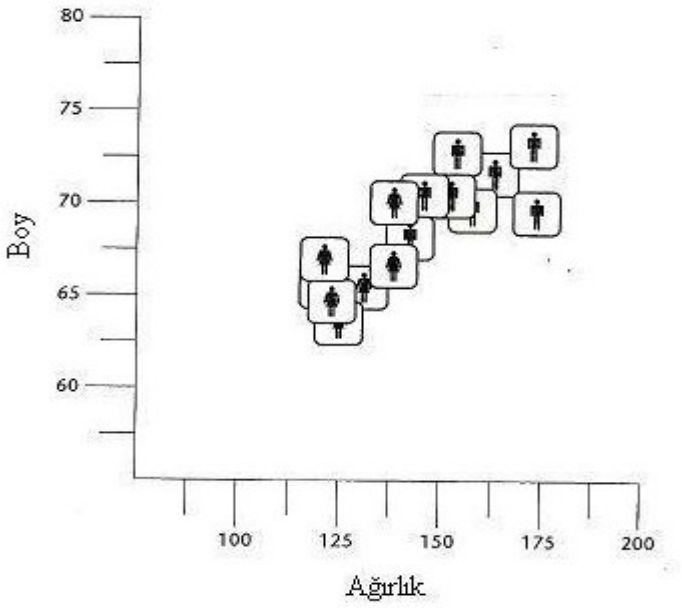
geçirdiklerini gösterir. Birkaç milyar sonra hidrojen tükenir. Kütlesine bağlı olarak daha sonra yıldız helyum füzyonuna başlar ya da füzyon durur. Füzyon durursa yıldızın çekirdeği çöker ve bu esnada büyük miktarda ısı açığa çıkar. Aynı zamanda dış yüzeyindeki gaz tabaka genişler ve kırmızı devler (Red Giants) oluşur. En sonunda, dış yüzeyindeki gaz tamamen kaybolur ve geride kalan çekirdek tamamen soğumaya başlar. Bu şekilde oluşan yıldıza da beyaz cüce (White Dwarf) denir. Google' da "Hertzsprung-Russell Diyagramı" deyimiyle yapılacak bir arama, küme bulmaya yönelik olarak yapılan birçok geçerli araştırmanın linkini getirecektir. Günümüzde bile HR(Hertzsprung-Russell) diyagramına dayalı kümeler, ana dizi öncesi yıldız evrimini anlamak ve kahverengi cüceleri (Brown Dwarfs) yakalamak için kullanılırlar. Kahverengi cüceler, nükleer füzyonu başlatmak için yeterince kütlesi olmayan yıldız benzeri nesnelerdir. HR diyagramı kümeleye iyi bir örnektir. Çünkü sadece iki değişken kullanılarak kümeleri iyi bir şekilde göstermektedir. HR diyagramı Şekil 3.2' de verilmiştir [41].



Şekil 3.2: Hertzsprung-Russell Diyagramı [41].

Şekil 3.3 insanların vücut şekilleri hakkında kabaca bir fikir vermektedir. Amaç bu insanları giydirmek olduğunda birkaç ölçüm daha gerekmektir. 1990 yıllarında

Amerikan ordusu, kadın askerlerinin üniformalarının yeniden tasarlanması için bir çalışma yürütmüştür. Ordunun amacı kayıt defterlerinden tutulması gereken farklı bedenlerdeki uniforma sayısını azaltmaktı. Bunu yaparken her askere iyi uyan uniformaların kalitesininde bozulmaması gerekiyordu. Daha önce kadın kıyafeti almış herkes, gereğinden fazla gruplama sisteminin(tek beden, çift beden, artı beden, minyon ve birçok beden) var olduğunu farkındadır. Bu sistemlerden hiçbiri Amerikan ordusunun ihtiyaçlarına göre dizayn edilmemiştir. Cornel üniversitesinde araştırmacı olan Susan Ashdown ve Beatrix Paal ordudaki kadınların vücut şekillerine göre yeni bedenler kümesi dizayn ettiler. Geleneksel kıyafet sisteminden farklı olarak, tüm boyutların birlikte artmadığı bir düzenleme getirmişlerdir. Belirli vücut yapılarına uyan bedenleri getirmişlerdir. Her vücut tipi, vücut ölçüm veritabanındaki kayıtlar kümesine karşılık gelmektedir. Bir küme kısa bacaklı, ince belli, büyük göğüslü, geniş omuzlu ve ince boğazlı özelliklerden oluşuyordu. Başka kümelerde farklı özelliklerden oluşmaktaydı. Veritabanı neredeyse 3.000 kadının her biri için 100 ölçümden daha fazlasını içeriyordu. Uygulanan kümeleme tekniği KM' tir. Sonunda kümeleri tanımlamak için 10' den fazla ölçüm gerekmekteydi. Değişkenlerin bu kadar küçük sayıda bulunması kümeleme işleminin diğer bir yararını da göstermektedir. Kümelemenin gerçek yaşamda uygulandığını ve iyi sonuçlar alınıldığını bu iki örnek açık bir şekilde belirtmektedir [41].



Şekil 3.3: Gençler grubunun ağırlık ve boyları [41].

3.3 Kümeleme Analizi Özellikleri

Kümeleme analizinin belli başlı özellikleri aşağıda açıklaması ile birlikte verilmiştir [37].

a) Ölçeklenebilir olmalıdır. Kümeleme algoritmaları 200' den az veri nesnelerini içeren küçük veri kümelerinde iyi bir şekilde çalışırken, büyük veri kümeleri üzerinde çok iyi bir şekilde çalışmayabilir. Bu gibi durumlarda ölçeklendirme algoritmalarına ihtiyaç vardır.

b) Farklı nesne tiplerine göre çalışabilmelidir. Günümüzde birçok kümeleme algoritması sayısal veriler üzerinde çalışması üzerine geliştirilmiştir. Ancak sayısal olmayan, kategorik ve ikili(binary) veriler üzerinde çalışacak algoritmalara ihtiyaç gün geçtikçe artmaktadır.

c) Düzgün şekilli olmayan kümeler de bulabilmelidir. Birçok kümeleme algoritması Manhattan ve Öklit uzaklık ölçümlerine göre kümelere karar vermektedir. Uzaklık ölçümlerine dayalı olan algoritmalar benzer boyut ve yoğunlukta olan küresel kümeler bulmaya eğilimlidirler. Buna rağmen kümeler herhangi bir şekilde olabilirler. Düzgün şekillerde olmayan kümeleri bulabilen algoritmaları geliştirmek önemlidir.

d) En az miktarda giriş değişkeni gerektirmelidir. Birçok kümeleme algoritması kullanıcı girişlerine ihtiyaç duyar. Kümeleme sonucuda bu parametrelere karşı hassastır ve bunlara göre değişiklik gösterir. Algoritma sonucu bu parametrelere bu kadar bağımlı olmamalıdır. Bu durum, parametreyi girecek kullanıcılar için büyük bir sıkıntıdır ve kümeleme analizinin sonucunu kontrol etmeyi zorlaştırır.

e) Gürültü içeren verileri de kullanılabilmelidir. Gerçek hayattta kullanılan birçok veritabanı eksik, tanımlanmamış ve aykırı veriler içerir. Kümeleme algoritmaları bu tür verilere karşı oldukça duyarlıdır ve bu tür veriler zayıf kalitede kümeler üretilmesine sebep olabilirler.

f) Verilen parametrelerin sırasına duyarsız olmalıdır. Bazı kümeleme algoritmalar giriş verisinin sırasına duyarlıdır. Bazı algoritmalarda girilen parametrelerinin sırası değiştiğinde algoritma sonucu oluşacak olan kümeler bundan etkilenir. İstenmeyen durumların oluşmaması için, algoritmada girilen parametrelerin sırasının önemli olmaması gerekir.

g) Çok boyutlu veritabanları ile çalışabilmelidir. Veritabanı veya veri ambarları birçok boyut ve nitelik içerebilirler. Birçok kümeleme algoritması düşük boyutlu veriyi kullanmakta iyidir. İnsan gözü en çok 3 boyutlu veriyi anlayabilcek yapıdadır. Fakat kümeleme algoritması daha fazla boyutta çalışabilmelidir.

h) Veri kümesinin sahip olduğu kısıtlamalar dikkate alınmalıdır. Gerçek dünya uygulamaları çeşitli kısıtlamalar altında kümeleme işlemini yapılabilmesine ihtiyaç duyar. Örneğin; belirli sayıda yeni otomatik para dağıtma makineleri için yerleri seçememiz gerektiğini düşünelim. Bu yerlere karar vermek için, şehrin nehirleri, karayolları ve her bölgenin müşteri gereksinimleri gibi kısıtlamaları dikkate almak gereklidir. Burada yapılması gereken, belirtilen kısıtlamaları tatmin eden iyi bir kümeleme yaparak verinin gruplarını bulmaktır.

i) Kolay yorumlanabilen ve kullanılabilen sonuçlar üretebilmelidir.

Mevcut kümeleme algoritmaları ideal bir kümeleme algoritmasından istenen bu özelliklerin tamamına sahiptir değildir. Kümeleme analizi ile ilgili çalışmalar devam etmektedir ve bu özelliklerin olabildiğince tamamını içinde barındırabilecek algoritmaların geliştirileceği umulmaktadır.

3.4 Kümeleme Analizi Veri Türleri

Günümüzde kümeleme algoritmaları 2 tür veri yapısıyla çalışırlar [37] :

a) Veri Matrisi (Data Matrix): Bu tip veri yapısında n tane nesne, p tane değişken olur. Örneğin nesneler insanlar, evler ve ağaçları temsil ediyorsa, değişkenler; bir

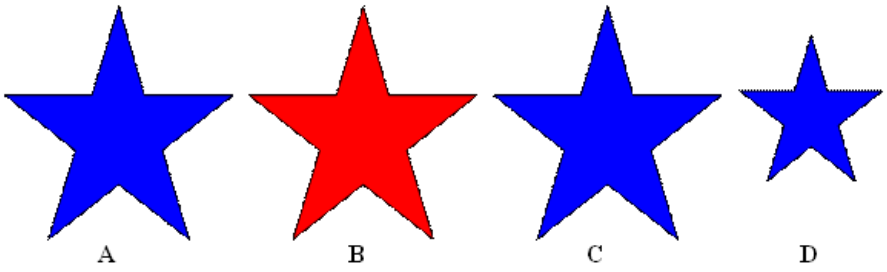
insanın ağırlık, boy ve yaşını temsil etmektedir. Bu matris Denklem 3.1' te gösterilmiştir.

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

b) Benzersizlik Matrisi (Dissimilarity Matrix): Nesnelerin diğer nesnelerle olan uzaklık bilgilerinin tutulduğu nxn boyutunda bir matristir. Bu matrisin genel ifadesi Denklem 3.2' te gösterilmiştir.

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

Veri madenciliği uygulamalarında çoğunlukla benzersizlik matrisi kullanılır. Benzersizlik matrisi tek modlu (one-two) matris olarak bilinir. Çünkü nesneler arası uzaklık fonksiyonu değişme özelliğine sahip olduğundan benzerlik matrisinin asal köşegeni altında ve üstünde kalan değerler simetriktir. Veri matrisi ise bu değişme özelliğine sahip olmadığından iki modlu matris(two mode) olarak bilinir[37]. Birçok algoritma benzersizlik matrisi ile çalışır. Eğer veriler ilk yapıdaki gibiyse yani veri matrisi ile ifade edilmişse, önce benzersizlik matrisine çevrilir ve daha sonra işletilir.



Şekil 3.4: Boyut ve renk özelliklerine göre yıldız şekilleri arasındaki benzerlik [43].

Benzerliğin ölçülmesi oldukça zordur. Benzerlik, iki nesne ya da özellik arasındaki güçlü ilişkiyi yansıtır. Benzemezlik ise birkaç özelliğe dayalı olarak iki nesne arasındaki farkın hesaplanmasıdır. Benzemezliğin ölçülmesi daha kolay olduğundan ilk önce benzemezlik hesaplanır ardından elde edilen değer benzerlik değerine dönüştürülür. Şekil 3.4' teki yıldızların hangilerinin birbirine benzediğini ya da benzemediğinin açık bir şekilde görebilmekteyiz. A' daki yıldız C' deki yıldıza benzemektedir. A, B ve C' deki yıldızlar aynı boyuta sahiptir. A, C ve D' deki yıldızlar aynı renge sahiptir. Boyut ve renk özelliklerinin kullanarak hangi yıldızların benzediğinin bulabiliriz. Uzaklık, iki nesne arasındaki benzemezliği ölçer. Benzerlik veya benzemezliği hesapladığımızda elde edebileceğimiz bilgiler [43];

- Bir veri kümesi içindeki bir nesneden diğeri ayırt edilebilir.
- Bir veri kümesi içindeki nesneler benzerlik ve benzemezliğe dayalı olarak gruplandırılabilir.
- Nesneler gruplandırıldıktan sonra, her bir grubun karakteristik özelliği ve kümelerin davranışı kolay bir şekilde anlaşılabilir.
- Gruplama bilginin daha etkili bir şekilde hazırlanması ve çıkarılmasını sağlayabilir.
- Yeni bir nesneyi uygun kümeye yerleştirebilir ve nesnenin davranışı tahmin edebilir.
- Veri kümesi içindeki yapı keşfedilebilir.
- Veri daha da yalınlaştırılabilir.

Uzaklık ölçütünün sağlaması gereken özellikler vardır. Bunlar aşağıda belirtilmiştir:

- $d(i,j) \geq 0$: Uzaklık, negative bir tam sayı değildir.
- $d(i,i) = 0$: Bir nesnenin kendine uzaklığı 0'dır.
- $d(i,j) = d(j,i)$: Uzaklık, simetrik bir fonksiyondur.
- $d(i,j) \leq d(i,h) + d(h,j)$: i nesnesinden j nesnesine direkt olarak gitmek, herhangi başka bir h nesnesi üzerinden dolambaçlı olarak gitmeye benzer. Buna üçgen eşitsizliği denmektedir.

Uzaklık ölçütü, yukarıdaki dört durumu sağlarsa metrik (metric) adıyla da çağrılmaktadır. Üçgen eşitsizliğinden dolayı tüm ölçülen uzaklıklar metrik değildir fakat tüm metrikler uzaklıktır [43].

3.4.1 Aralık ölçekli değişkenler (interval-scaled variables)

.

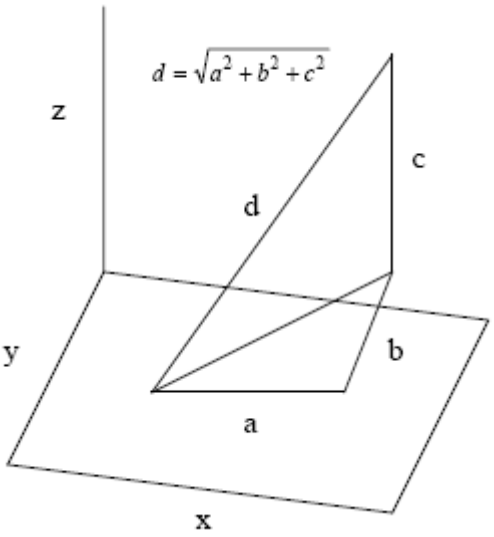
Tam olarak kesin belirlenmiş değerlerden çok, belli bir aralık şeklinde belirlenen verilerde geçerlidir. En sık kullanılan ağırlık ölçekli değişkenler boy, ağırlık, genişlik ve uzunluk verileridir. Ölçümde kullanılan birim çok önemlidir. Birimin değişmesi, analizin sonucunu etkiler. Sonucun kafa karıştırıcı olmaması için analize giren verilerin de standart olması gerekir (verilerin bir kısmı kg, diğerleri gr olmamalıdır). Standartlaştırmadan sonra benzersizlik matrisi ile analiz yapılır [37].

Aralık ölçekli veriler için uzaklık ölçümlerini hesaplamada üç çeşit formül kullanılır [37]:

a) Öklid Uzaklığı (Euclidean Distance): Öklid uzaklığı formülü ile standartlaştırılmış verilerle değil, işlenmemiş verilerle hesaplama yapılır. Öklid uzaklıkları kümeleme analizine sıradışı olabilecek yeni nesnelerin eklenmesinden etkilenmez. Ancak boyutlar arasındaki ölçek farklılıkları öklid uzaklıklarını önemli ölçüde etkilemektedir. Örneğin boyutlardan biri santimetre ile ölçülen bir uzunluğa ayarlı ise ve sonra biz bunu milimetreye çevirirsek, öklid uzaklığı bundan önemli ölçüde etkilenir. Kümeleme analizinin sonuçları çok farklı olabilir [44]. Denklem 3.3' te Öklit uzaklığını formülü gösterilmektedir.

$$\mathrm{d}(i,j)=\sqrt{\left(\left|x_{i1}-x_{j1}\right|^2+\left|x_{i2}-x_{j2}\right|^2+\ldots+\left|x_{ip}-x_{jp}\right|^2\right)} \tag{3.3}$$

Formüldeki i ve j nesneleri p boyutlu veri nesneleridir.



Şekil 3.5: Öklit uzaklığının şekilsel olarak gösterimi [44].

Öklit uzaklığının hesaplanmasına bir örnek verecek olursak; maliyet(cost), zaman(time), ağırlık(weight) ve harekete geçirici(incentive) şeklinde 4 tane özelliğe sahip olan A ve B nesnelerini düşünelim. Tablo 3.1' de A ve B nesnelerinin belirtilen özeliklere ait değerleri verilmiştir [43].

Tablo 3.1: A ve B nesnelerinin belirtilen özelliklerine göre değeri [43].

	Maliyet	Zaman	Ağırlık	Harekete Geçirici
A nesnesi	0	3	4	5
B nesnesi	7	6	3	-1

A nesnesi (0, 3, 4, 5) koordinatlarına, B nesnesi (7, 6, 3, -1) koordinatlarına sahiptir. A ve B nesneleri arasındaki Öklit uzaklığı sonucu(d_{BA}) aşağıda gösterilmiştir:

$$d_{BA} = \sqrt{(0-7)^2 + (3-6)^2 + (4-3)^2 + (5+1)^2} = 9.747$$

b) City-Block (Manhattan) Uzaklığı(City-Block (Manhattan Distance)): Manhattan uzaklığı boyutlar arasındaki farka eşittir. Bu ölçüt kullanıldığında farkın karesi alınmadığı için aykırı değerlerin etkisi azalır. Manhatan uzaklığının formülü 3.4' te gösterilmektedir.

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + .. + |x_{ip} - x_{jp}| \quad (3.4)$$

Şekil 3.6: Manhattan uzaklığının şekilsel olarak gösterimi [44].

Tablo 3.2' de A ve B nesnelerinin belirtilen özeliklere ait değerlerini kullanarak A ve B nesneleri arasındaki Manhattan uzaklığını hesaplarsak sonuç aşağıdaki gibi bulunur:

$$d_{BA} = |0-7| + |3-6| + |4-3| + |5+1| = 17$$

c) Minkowski Uzaklığı (Minkowski Distance): Manhattan uzaklığı ve öklid uzaklığının genelleştirilmiş hali olarak ifade edilmektedir. Minkowski uzaklığını formülü 3.5' te gösterilmektedir.

$$d(i,j) = \sqrt[q]{\left(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + ... + |x_{ip} - x_{jp}|^q\right)} \quad (3.5)$$

q pozitif bir tam sayıdır. q=1 olduğunda d(i,j) Manhattan uzaklığına, q=2 olduğunda d(i,j) öklid uzaklığı eşit olur.

Tablo 3.2' de A ve B nesnelerinin belirtilen özeliklere ait değerlerini kullanarak A ve B nesneleri arasındaki Minkowski uzaklığını hesaplarsak sonuç aşağıdaki gibi bulunur:

$$d_{BA} = \left(|0-7|^3 + |3-6|^3 + |4-3|^3 + |5+1|^3\right)^{\frac{1}{3}} = \sqrt[3]{587} = 8.373$$

3.4.2 İkili değişkenler (binary variables)

Bir ikili değişkenin 0 ve 1 olmak üzere 2 durumu vardır. 0 yok, 1 var anlamında kullanılır. Aralık ölçeklinin tersine, kesin ve net sonuçların olduğu analizlerde kullanılır. Örneğin; bir yolcunun sigara içip içmediğine yönelik sorulan bir sorunun karşılığı; eğer içiyorsa 1, içmiyorsa 0' dır. Örnekte de görüldüğü gibi cevap olarak bir aralık çıkmamakta ve kesin bir cevap alınmaktadır [37]. İkili değişkenler verisi için olasılık tablosu Tablo 3.2' de gösterilmiştir:

Tablo 3.2: İkili değişkenler için olasılık tablosu [37].

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

Simetrik ikili değişkenler için uzaklık ölçüsü:

$$\mathrm{d}(i,j)=\frac{b+c}{a+b+c+d} \tag{3.6}$$

Asimetrik ikili değişkenler için uzaklık ölçüsü:

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

Jaccard katsayısı, asimetrik ikili değişkenler için benzerlik ölçüsüdür.

$$\mathrm{sim}_{Jaccard}(i,j)=\frac{a}{a+b+c} \tag{3.8}$$

a, Ortak olan 1'lerin sayısını belirtmektedir. b, ilk nesne için 1, ikinci nesne için 0 olanların sayısını belirtmektedir. c, ilk nesne için 0, ikinci nesne için 1 olanların sayısını ifade etmektedir. d, ortak olan 0'ların sayısını ifade etmektedir.

Örneğin ad, cinsiyet, ateş, öksürük, test–1, test–2, test–3, test–4 niteliklerini içeren bir Tablo 3.3' de olduğu gibi bir hasta kayıt tablosu olduğunu düşünelim. Cinsiyet, simetrik bir niteliktir. Kalan diğer nitelikler ise asimetrik niteliklerdir. Y ve P değerleri 1 olarak, N değerleri de 0 olarak ayarlanır.

Tablo 3.3: Hasta kayıt tablosu [42].

İsim	Cinsiyet	Ateş	Öksürük	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	Y	N	N	N	N
…	…	…	…	…	…	…	…
….	…	…	…	…	…	…	…

$$d(Jack, Mary) = \frac{0+1}{2+0+1} = 0.33 \longrightarrow \text{En çok benzer}$$

$$d(Jack, Jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(Jim, Mary) = \frac{1+2}{1+1+2} = 0.75$$

Bu ölçümler Jim ve Mary'nin aynı hastalığa sahip olmayacaklarını gösterir. Çünkü 3 çift arasındaki benzemezlik değeri en yüksek olan çift Jim ve Mary çiftidir.

3.4.3 Nominal, ordinal ve oran değişkenleri (nominal, ordinal and ratio-scaled variables)

Nominal değişkenler, ikili değişkenlere çok benzeyen değişkenlerdir. Ordinal değişkenler ise nominal değişkenlere benzemekle birlikte sıranın önemli olduğu değişkenlerdir. Oran ölçekli değişkenler ise üstel olarak artan verilerde kullanılırlar. Bu değişkenler aşağıda sırayla açıklanmıştır:

a) Nominal değişkenler

Nominal değişkenler, ikili değişkenlere benzeyen ve çok sayıda seçeneği olan değişkenlerdir. Örneğin renk değişkeni nominal bir değişkense kırmızı, yeşil, mavi, pembe ve sarı durumlarına sahip olduğunu düşünebiliriz. Nominal değişkenin durumlarının sayısı M olsun. Durumlar; 1, 2, ..., M gibi tamsayı kümesi, sembol ve harflerle ifade edilebilir. Tamsayılar özel bir sıralama olmadan veriyi kontrol etmek için kullanılır. Örneğin map_color nominal değişkenini oluşturmak için, yukarıda listenen her bir renk için bir ikili değişkeni yaratılabilir. Sarı rengine sahip bir nesne için, sarı değişkeni 1' e ayarlanır, kalan 4 değişkende 0' a ayarlanır. Nominal değişkenler olarak tanımlanan nesneler arasında farklılığın hesaplanması için aşağıdaki formül kullanılır [37]:

$$\mathrm{d}(i, j) = \frac{p - m}{p} \tag{3.9}$$

Formüldeki p değişkeni i ve j nesnelerinin sahip olduğu toplam özellik sayısını, m değişkeni i ve j değişkenlerinde aynı anda yer almış olan özellik sayısını ifade eder.

b) Ordinal değişkenler

Ordinal değişkenler, nominal değişkenlerde olduğu gibi sonlu sayıda farklı durum içerir. Nominal değişkenlerden farklı olarak ordinal değişkenlerde sıra önemlidir. Örneğin yarışmalarda en yüksek dereceye sahip olan yarışmacıya altın, daha sonrakine gümüş ve üçüncü olan yarışmacıyada bronz madalya verilir. Ordinal

değişkenler arası farklılığı hesaplamak için, ordinal değişkenlerin alabileceği değerleri [0-1] aralığında sayı değerleri alabilecek şekilde standartlaştırıp aralık ölçekli değişkenlerde kullanılan mesafe yöntemleri kullanılır.

c) Oran ölçekli değişkenler

Üstel olarak artan verilerin benzerliğinin bulunmasında kullanılır. Oran ölçekli değişkenlere bakteri popülasyonlarında büyüme ve radyoaktif elementin yarı ömrünün ölçüm sonuçları örnek olarak verilebilir. Oran ölçekli değişkenlerin genel yapısı aşağıdaki gibidir:

$$Ae^{Bt} \text{ veya } Ae^{-Bt} \tag{3.10}$$

Denklemdeki A ve B pozitif sabitlerdir. Oran ölçekli değişkenlerde nesneler arasındaki farklılığı hesaplamak için üç farklı metot vardır:

1) Bu yöntemde; oran ölçekli değişkenler aralık ölçekli değişkenler gibi davranırlar. Bu yöntem iyi bir seçim değildir. Çünkü ölçülen aralık doğrusal olmadığından ölçümün hatalı olması olasıdır.

2) Oran ölçekli değişkenlere logaritmik ölçümler uygulanabilir.

$$y_{if} = \log(x_{if}) \tag{3.11}$$

Bu formül kullanıldığında elde edilen y^{if} değeri ile aralık ölçekli değişken olarak işlem yapılabilir.

3) Bu metotta, oran ölçekli değişkenler sürekli ordinal değişkenler olarak düşünür ve ordinal değişkenlerdeki uzaklık hesaplamaları kullanılır. Son 2 metot çok etkilidir. Buna rağmen kullanılan metodun seçimi verilen uygulamaya bağlı olabilir.

3.4.4 Karışık tür değişkenler

Karışık tür değişkenler ikili, aralık ölçekli, oran ölçekli, ordinal değişkenler gibi veri türlerinden 2 ya da daha fazlasına içeren değişkenlerdir. Birçok gerçek veritabanında değişik tipte veriler bulunur. Bu karışık verilerin hepsinin bir arada analiz edilmesi gerekir[37]. Karışık tür değişkenlerin hesaplanmasında 2 temel yaklaşım kulllanılmaktadır:

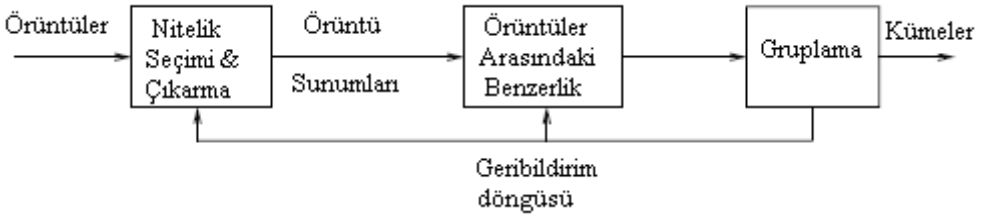
1) Tüm değişkenler çeşitlerine göre gruplandırılır. Her bir grup için ayrı bir kümeleme analizi işlemi gerçekleştirilir. Bu kümeleme analizleri birbiri ile uyumlu sonuçlar üretirse bu yaklaşımı kullanmak uygun olur. Ancak gerçek uygulamalarda her bir değişken çeşidinin ayrı kümeleme analizinin uyumlu sonuçlar üretebilmesi olanaklı değildir. Ayrıca bu yaklaşım karışık olduğu kadar yoğun işlem gücü gerektirir [37].

2) Tüm değişken çeşitleri birlikte işleme tabi tutulur. Bu teknikte; farklı değişkenler tek bir benzersizlik matrisi içine yerleştirilir. Veri kümesi p tane karışık türdeki değişkenleri içerir. J.Han bu yaklaşımla ile ilgili bir formül sunmuştur. Fakat tez konusu ilgili olmadığından burada bu formüle yer verilmemiştir [37].

3.5 Kümeleme İşleminin Adımları

Kümeleme işlemi 5 adımdan oluşmaktadır. Bu adımlar;

- Örüntü Sunumu(Pattern Representation).
- Örüntü yakınlık ölçümüne uygun veri alanının tanımlanması.
- Kümeleme(Clustering) veya gruplama(Grouping).
- Veri çıkarılması(Data Abstraction).
- Çıkışın Değerlendirilmesi.



Şekil 3.7: Kümeleme işleminin adımları [17].

Şekil 3.7 ilk 3 basamağın tipik sıralanmasını gösterir. Ayrıca gruplama işlemi sonuçlarının sonraki özellik çıkarımını ve benzerlik hesaplamasını etkileyebildiği geribildirim yolunuda içerir [17].

Örüntü Sunumu (Pattern Represantation): Sınıfların sayısı, mevcut örüntülerin sayısı ve var olan niteliklerin sayı, tipi ve ölçüsünü kümeleme algoritmasına gönderir. Bu bilgilerin bazıları kullanıcı tarafından kontrol edilemeyebilir. Örüntü sunumu, nitelik çıkarma ve seçimini isteğe bağlı olarak içerir. Nitelik seçimi; kümelemede kullanmak için orjinal niteliklerin en etkili alt gruplarını tanımlama işlemidir. Nitelik çıkarma; yeni belirgin nitelikler üretmek için giriş niteliklerinin bir ya da daha fazla dönüşümlerinin kullanılmasıdır. Bu tekniklerin ikisinden biri ya da her ikisi kümelemede kullanmak üzere uygun nitelikler kümesini sağlamak için kullanılabilir.
Örüntü Yakınlığı (Pattern Proximity): Örüntü yakınlığı genellikle örüntü çiftlerinde tanımlanan uzaklık fonksiyonu ile ölçülür. Çeşitli uzaklık ölçümleri değişik alanlarda kullanılır. Öklid uzaklık ölçümü gibi basit uzaklık ölçümleri 2 örüntü arasındaki farklılığı yansıtmak için sık sık kullanılır.

Gruplama (Grouping): Gruplama işlemi değişik yollarla yapılabilmektedir. Kümelemenin çıkışı katı (hard) veya bulanık (fuzzy) olabilir. Katı kümeleme algoritmalarında her bir veri noktası sadece bir tane kümeye atanır. Bulanık kümeleme de ise her veri noktası bazı üyelik dereceleri ile her bir kümeye atanır.

Veri Çıkarma (Data Abstraction): Bu işlem isteğe bağlı olarak gerçekleştirilir. Veri kümesinin temsilinin basit ve öz olarak çıkartılması işlemidir. Bu basitlik otomatik analizden veya insan düzenlemesinden gelir. Otomatik analiz sayesinde bir makine ileri işlemleri otomatik olarak gerçekleştirebilir. Basitlik insan düzenlemesiyle

gerçekleştirilirse elde edilen sunumların kavranması kolay olur. Kısaca kümelemede veri çıkarımı her kümenin yoğun ve öz olarak tanımlanmasıdır.

Küme Geçerliliği (Cluster Validity): Bu işlem isteğe bağlı olarak gerçekleştirilir. Küme geçerliliği kümeleme işlemi sonucunda oluşan çıkışların değerlendirilmesidir. Bu değerlendirme işlemi için özel kıstaslar kullanılır. Ancak bu kıstasların sonuçları genelde öznel olur.

3.6 Birçok Kümeleme Algoritmasının Ortaya Çıkmasının Nedenleri

Küme kavramı tam olarak tanımlanamadığından, birçok kümele algoritması ortaya çıkmıştır. Kümelemenin doğasında; doğrulamaktan daha çok keşif vardır. 1964' te Bonner küme için evrensel bir tanımın olamıyacağını söylemiştir. Daha yakın geçmişimizde, aykırı değerler ve kümelerin kişinin bakış açısına bağlı olduğunu keşfetmişlerdir. Bir insanın gürültüsü başka bir insanın sinyali olabilir. Bu da farklılığa katkıda bulunan bir elementtir. Araştırmacılar birçok başlangıç prensibi ve modeller önermişlerdir. Bunların optimizasyon problemi sadece çok sayıdaki algoritma tarafından yaklaşık olarak çözülebilmektedir [15].
Kümeleme algoritmalarının çeşitliliği başlangıç prensipleri ve modellerinin çeşitliliğinden kaynaklanır. Her başlangıç prensibi için birçok algoritma ve başlangıç prensibi olduğundan dolayı birçok kümeleme algoritması vardır. Birçok başlangıç prensibi olmasının nedeni kümelemenin bakan göze göre değişiyor olmasıdır. Başlangıç prensipleri araştırmacıların inandığı küme tanımının matematiksel formülüdür. Bir kümeyi neyin oluşturduğu ve iyi bir kümelemeyi neyin oluşturduğu sübjektiftir. Çünkü bunlar araştırmacının geçmişinden ve uygulamasından etkilenir. Bu makalede kümeleme için bazı öneriler sunulmuştur [15]:

- Kümelemenin büyük bir bölümünün bakan göz tarafından etkilendiğini unutmamak gerekir.
- Farklı araştırmacılar değişik matematiksel formüllerle iyi kümelerin ne olduğunu belirtmişlerdir. Farklılıkta bir zenginlik vardır.
- Kümeleme algoritmalarının derlenmesi ve incelenmesinin kategorilere ayrılması başlangıç prensiplerinden daha çok modellere dayanır. Kategorilere ayırma

işlemi modellerin ve başlangıç prensiplerinin var olmayışından etkilenmemelidir. Kümeleme algoritmaları arasındaki en güçlü farklılık; matematiksel ve yapısal modellerin benimsenmesidir.

- Kümeleme optimizasyon problemine çevrilir. Optimizasyon probleminin hesapsal karmaşıklığı tipik olarak kontrol edilmesi zor ve yaklaşık algoritmalar tarafından çözülebilmesidir.
- İki kümeleme algoritması arasında birinci seviye karşılaştırma; aynı performans fonksiyonu tarafından ölçülen çözümün kalitesine göredir.
- Diğerlerinden daha az maliyetli olan performans fonksiyonları vardır. Daha az maliyetli ve pahalı performans fonksiyonlarına göre daha iyi sonuçlar veren performans fonksiyonlarını aramak doğaldır. Başlangıç prensipleri arasındaki ilişkinin araştırılması gerekir.
- Araştırmacıların yeni veya geliştirilmiş bir kümeleme algoritması önerirken kendi modellerini ve başlangıç temellerinin ne olduğunu açıkça belirtmeleri gerekir. Önceki ve oluşturulan metotların karşılaştırılmasını ve gelecekteki araştırmaları kolaylaştırır.
- Küme geçerliliğinin indeksi; başlangıç prensiplerinin matematiksel olarak formüle dönüştürülmesidir. Algoritmaları karşılaştırmak içerik hakkında bazı kavramları sağlar ve hangi algoritmanın daha iyi çalıştığı hakkında bilgi sağlar. Fakat bu bir algoritmanın diğerinden daha geçerli sonuçlar ürettiği anlamına gelmez. Geçerlilik yapının var olduğu veri kümesine bağlıdır. Bir kümede yapısı olmayan 2 algoritma geçersiz sonuçlar üretecektir.
- Kümele kalitesi; ilişkili ölçüler ve geçerlilik kıstası ile gösterilir. Yapının ne kadarı algoritma tarafından düzeltilebilir. Bu görüşe göre, eğitilmiş öğrenme için veri kümeleri kümeleme algoritmalarını değerlendirmek için kullanılabilir. Bilinen etiketler ve kümeleme arasındaki zıtlık ölçülür.
- Eğer veri kümesinin yapısı tamamen farklı bir aileden olan model tarafından sunulabiliyorsa, evrensel modeller için düzenlenen bir algoritmanın bir şansı yoktur (Örnek K-Means dış bükey olmayan kümeleri bulamaz.).

3.7 Kümeleme Metotları

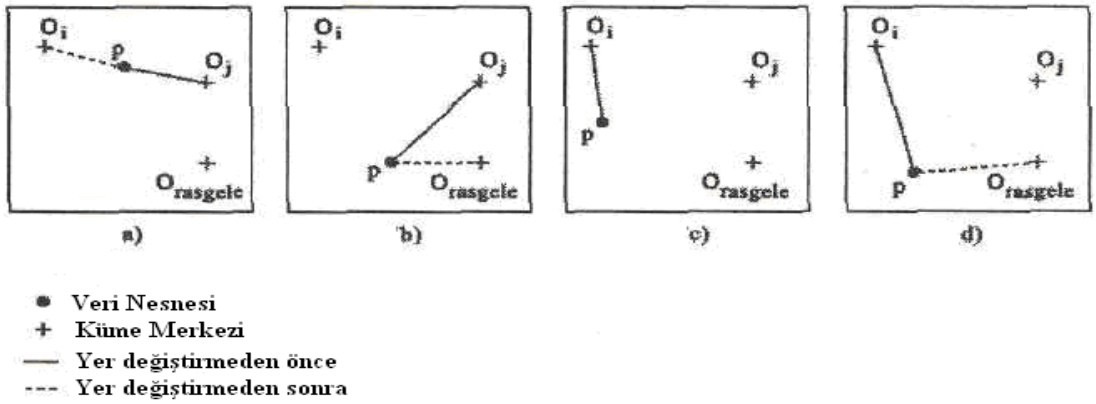
Veri madenciliğinde uygulanan pek çok kümeleme metodu bulunmaktadır. Kullanıcak veri türüne ve uygulamanın amacına göre uygun metotlar seçilir. Başlıca kümeleme metotları ve bu metotlarda uygulanan algoritmalar bu bölümde açıklanmıştır.

3.7.1 Bölümleme metotları (partitioning methods)

Bölümleme metotlar (partitioning methods), n adet nesneden oluşan veritabanını, giriş parametresi olarak belirlenen k adet bölüme ($k \leq n$) ayırma esasına dayanır. Bu ayrılan her bölüm kümeyi ifade eder.

3.7.1.1 K-medoids algoritması

Çok yüksek değerdeki nesneler küme dağılımını olumsuz yönde etkiler. KM algoritmasında değeri çok büyük olan nesne, dahil olacağı kümenin ortalamasını ve merkez noktasını büyük derecede değiştirebilir. Bu sorunu gidermek amacıyla ortaya çıkan k-medoids algoritması ortaya atılmıştır. K-medoids algoritması, kümeyi temsil edecek noktayı bulmak için küme elemanlarının ortalamasını almak yerine medoid' i kullanır. Medoid, küme içinde en merkeze yerleşmiş olan nesnedir. K-medoids kümeleme algoritmasının başlıca stratejisi ilk olarak n adet nesne içinde, merkezi temsili bir medoid olan k adet küme bulmaktır. Geriye kalan nesneler kendilerine en yakın olan medoide bulunduğu k adet kümeye yerleştirilirler. Bu stratejide iteratif olarak medoid olmayan nesnelerden biri ile medoid olan nesnelerden biri yer değiştirir. Bu şekilde kümenin ortasına en yakın olan nesne bulunmaya çalışılır. Bu işlem en verimli medoid bulunana kadar devam eder. Sonuçlanan kümelemenin kalitesi, nesne ve nesnenin bulunduğu kümenin medoid' i arasındaki ortalama benzersizliğini (average dissimilarity) ölçen maliyet fonksiyonu kullanılarak tahmin edilir.



Şekil 3.8: K-medoids yöntemi ile kümeleme örneği [37].

Şekil 3.8' te O_i ve O_j iki ayrı kümenin medoidlerini, $O_{rastgele}$ rastgele seçilen ve medoid adayı olan bir nesneyi, p ise medoid olamayan bir nesneyi temsil etmektedir. Şekil 3.8 $O_{rastgele}$ ' nin, şu anda medoid olan O_j ' nin yerine geçip, yeni medoid olup olmayacağını dört durumu göz önüne alınarak karar verilmektedir[37].

a) p nesnesi şu anda O_j medoidine bağlıdır (O_j medoidinin bulunduğu kümededir). Eğer O_j, $O_{rastgele}$ ile yer değiştirir ve p O_i' ye en yakınsa($i \neq j$), p nesnesi O_i' ye geçer.

b) p nesnesi şu anda O_j medoidine bağlıdır. Eğer O_j, $O_{rastgele}$ ile yer değistirir ve p $O_{rastgele}$ ' ye en yakınsa, p nesnesi $O_{rastgele}$' ye geçer.

c) p nesnesi şu anda O_i medoidine bağlıdır($i \neq j$). Eğer O_j, $O_{rastgele}$ ile yer değistirir ve p hala $O_{rastgele}$' ye en yakınsa, p nesnesi yine O_i' ye bağlı kalır.

d) p nesnesi şu anda O_i medoidine bağlıdır($i \neq j$). Eğer O_j, $O_{rastgele}$ ile yer değiştirir ve p $O_{rastgele}$ ' ye en yakınsa, p nesnesi $O_{rastgele}$' ye geçer.

K-medoids algoritmasının birçok farklı türevi vardır. PAM (Partitioning Around Medoids) ilk olarak ortaya atılan K-medoids algoritmasıdır. PAM öncelikle rastgele seçtiği k adet nesneyi başlangıç medoid' leri olarak ele alır. Kümeye her yeni eleman

katıldığında kümenin elemanlarını deneyerek kümenin gelişmesine en fazla katkıda bulanan noktayı bulunca bu noktayı yeni medoid, eski medoid de normal bir nokta olacak şekilde yer değiştirir. PAM küçük veri kümeleri için iyi sonuçlar verirken büyük veri kümeleri için hesaplanabilir karmaşıklığı yüksek olduğundan iyi sonuçlar vermez. Büyük veritabanları için CLARANS algoritması geliştirilmiştir [37].

K-medoids algoritması gürültü ve aykırı değerlere KM algoritmasından daha dayanıklıdır. K-medoids algoritmasının çalışması KM algoritmasından daha maliyetlidir. Her iki algoritma da kümelerin sayısını ifade eden k değerini kullanıcının belirtmesine ihtiyaç duyar [37].

3.7.1.2 Beklenen eniyileme (gaussian expectation maximization)

Beklenen eniyileme (EM-Expectation Maximization) algoritması yaygın olarak kullanılan bölümleme algoritmalarının bir çeşidi olan merkez tabanlı kümeleme algoritmalarından biridir. EM algoritması, özellikle olasılık modellerindeki parametrelerin maksimum olasılık tahminlerini bulmak için istatistikte kullanılır. EM algoritması bilinmeyen verinin varlığında parametrelerin tahmininde kullanılır. EM algoritmasının amaç fonksiyonu aşağıda verilmiştir [26]:

$$\text{GEM(X,C)} = -\sum_{i=1}^{n} \log\left(\sum_{j=1}^{k} p\left(x_i \middle| c_j\right) p\left(c_j\right) \right) \tag{3.12}$$

Denklemdeki X, veri noktalarını içinde barındıran kümedir. C, merkez noktalarını içinde barındıran kümedir. n, veri kümesi içindeki noktaların sayısıdır. K, oluşturulacak olan küme sayısıdır. $P(x_i \mid c_j)$, c_j' nin olasılığıdır. $P(c_j)$, önceki c_j merkezinin olasılığıdır. EM algoritması aşağıdaki gibi yumuşak üyelik fonksiyonuna sahiptir [26]:

$$m_{GEM}\left(c_j \middle| x_i\right) = \frac{p\left(x_i \middle| c_j\right) p\left(c_j\right)}{p\left(x_i\right)} \tag{3.13}$$

EM algoritması aşağıdaki gibi sabit ağırlık fonksiyonuna sahiptir. Ağırlık fonksiyonu, bütün veri noktalarına eşit önemi verir. Ağırlık fonksiyonu $w_{GEM}(x_i)$ ile $p(x_i)$ aynı değildir [26].

$$w_{GEM}(x_i) = 1 \tag{3.14}$$

EM algoritması yinemeli bir algoritma olup iki aşamadan oluşmaktadır. Algoritma başlangıç parametrelerinin tahmini ile başlar. Sonra beklenti adımına (expectation step), bilinen veri değerlerinin bilinmeyen veriden beklenen değerlerin hesaplanmasında başvurulur. Daha sonra eniyileme (maximization step) adımı geçilir. Bu adımda, verinin bilinen ve beklenen değerleri yeni tahmini parametrelerin yaratılmasında kullanılır. Bu iki adım yakınsama gerçekleşinceye kadar devam eder. EM algoritması parametrelerin başlangıç tahminlerine duyarlıdır. EM algoritmasında kümelerin sayısının kullanıcı tarafından belirtilmesine ihtiyaç duyulmaktadır [45].

3.7.1.3 CLARA ve CLARANS algoritmaları

Küçük ölçekli veritabanlarında PAM adındaki k-medoids algoritması kullanılmaktadır. Fakat büyük veritabanlarında bu algoritmanın performansı iyi değildir. Bu nedenle büyük ölçekli veritabanları için CLARA (Clustering LARge Applications) algoritması geliştirilmiştir. CLARA algoritması örnek tabanlı bir metottur(sampling-based method). CLARA algoritması bütün veri kümesini almak yerine, veri kümesinin küçük bir kısmını verinin temsili olarak seçer. CLARA veritabanında birden çok örnek(sample) seçer. Her bir örnek üzerine PAM uygular ve en iyi sonucu veren örnekten elde ettiği PAM sonucunu çıktı olarak verir [37].

CLARA geniş veri kümeleri üzerinde PAM'den daha iyi çalışır. CLARA'nın her bir iterasyonda karmaşıklığı $O(ks^2+k(n-k))$' dır. s örnek boyutunu, k kümelerin sayısını ve n nesnelerin toplam sayısını ifade eder. CLARA metodunun etkisi ve kalitesi, boyuta ve rasgele seçilen verilerin ne kadar iyi seçildiğine bağlıdır. PAM verilen veri kümesi arasındaki en iyi k medoidlerinin araştırırken CLARA veri kümesinin seçilen örneği arasındaki en iyi k medoidlerini araştırır. Herhangi örneklenmiş medoid en iyi

k medoidleri arasında değilse CLARA en iyi kümelemeyi bulamaz. Örneğin Oi nesnesi en iyi medoidlerin içindeki medoidlerden biri ise ve örnekleme boyunca bu medoid seçilmezse CLARA en iyi kümelemeyi asla bulamayacaktır [37].

CLARA' nın ölçeklenirliğini ve kalitesini geliştirmek için CLARANS (CLustering Algorithm based on RANdomized Search) adındaki k-medoids algoritması ortaya atılmıştır. CLARANS PAM ile örnekleme tekniğini birleştirir. CLARA aramanın her bir evresinde sabit bir örnek kullanırken CLARANS her aşamada değişen örnekleri kullanılır [37].

CLARANS, CLARA ve PAM' den daha etkili olarak çalışmaktadır. CLARANS aykırı değerleri bulmayı sağlar. CLARANS algoritmasının hesapsal karmaşıklığı $O(n^2)$' dir. n nesnelerin sayısını ifade ettiğinde, veri sayısı arttıkça hesaplama gücü üstel olarak artar [37].

3.7.2 Hiyerarşik metotları (hierarchical methods)

Hiyerarşik kümeleme metotları veri nesnelerini dendogram adı verilen ağaç yapısı içerisine gruplandırmaya çalışır. Bu ağaç, yapraklardan gövdeye doğru veya gövdeden yapraklara doğru kurulabilir. Hiyerarşik kümelemeye toplayıcı ve bölücü kümeleme (Agglomerative and Divisive Hierarchical Clustering), BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies), CURE (Clsutering Using REpresentatives) ve CHAMELEON (A Hierarchical Clustering Algorithm Using Dynmaic Modeling) algoritmaları örnek olarak verilebilir. Toplayıcı hiyerarşik kümelemede hiyerarşik ayrışma aşağıdan yukarıya doğru olur. Bu nedenle aşağıdan-yukarıya yaklaşım olarak geçmektedir. Bölücü hiyerarşik kümelemede hiyerarşik ayrışma yukarıdan aşağıya doğru olmaktadır. Bölücü hiyerarşik kümeleme yukarıdan-aşağıya yaklaşım olarak geçmektedir.

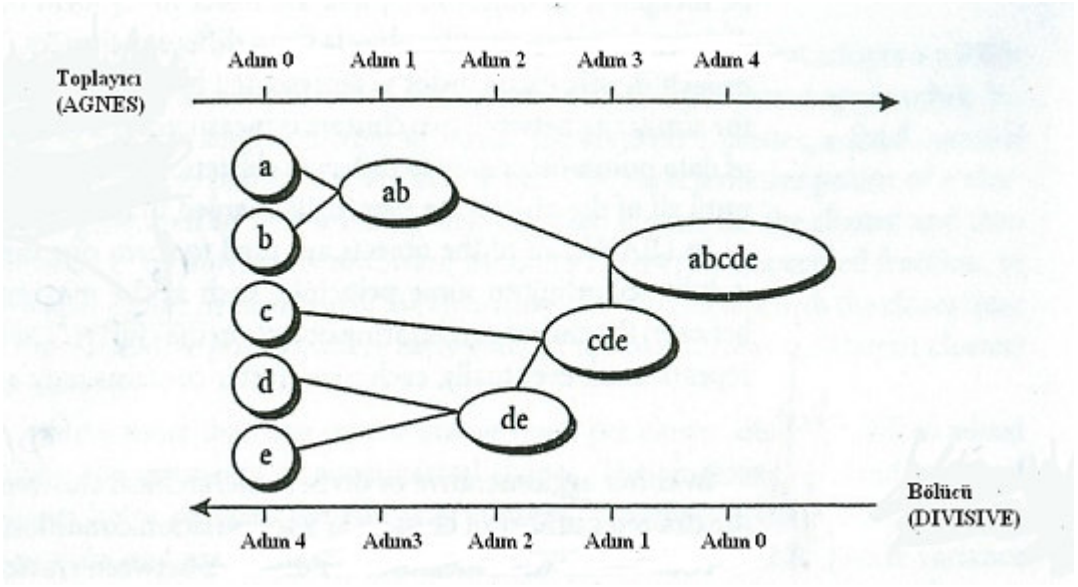
3.7.2.1 Toplayıcı ve bölücü algoritmalar

Her bir nesnenin ilk başata farklı bir küme oluşturmasıyla başlayıp sonlandırma kıstası sağlanıncaya kadar kümelerin birleştirme işlemine devam eden toplayıcı

hiyerarşik kümeleme yaklaşımı vardır. Ayrıca bütün nesnelerin aynı kümede olması ile başlayıp sonlandırma koşulu sağlanıncaya kadar kümeleri bölen bölücü hiyerarşik kümeleme yaklaşımı vardır. Bu belirtilen iki yaklaşım aşağıda sırasıyla açıklanmıştır:

3.7.2.1.1 Toplayıcı hiyerarşik kümeleme

Aşağıdan-yukarıya yaklaşım hiyerarşik kümeleme; kendi kümesi içindeki her bir nesnenin yerini değiştirmesiyle başlar ve daha sonra atomik kümeleri, tüm nesneler tek bir küme içinde toplanıncaya kadar veya belirli bir son koşulu sağlanana kadar, benzerliklere dayalı olarak birbirini takip edecek şekilde daha geniş olan kümelere birleştirir. n nesne için, n–1 birleştirme yapılır. Hiyerarşik algoritmalarla bir birleştirme yapıldığında geriye dönüş yoktur. Toplayıcı hiyerarşik kümeleme algoritmaları az hesaplama maliyetine sahip olmasına rağmen yanlış bir birleştirme yapılması sorunlara yol açar. Bu yüzden birleştirme noktalarının dikkatlice seçilmesi gerekir. Çoğu hiyerarşik kümeleme metodları bu kategoride yer almaktadır. Örnek AGNES (AGglomerative NESting) olarak adlandırılan Kaufman tarafından ortaya atılan metod bu yaklaşımı kullanır. Şekil 3.9' da gösterildiği gibi {a, b, c, d, e} nesneleri AGNES metodu ile başlangıçta her nesne kendisine ait olan bir küme içinde olacak yerleştirilir. Daha sonra kümeler diğer kümeler içindeki en yakın nesneyle arasındaki minimum öklit uzaklığına göre adım adım birleştirilmiştir.



Şekil 3.9: Veri nesneleri üzerinde toplayıcı ve bölücü hiyerarşik kümeleme [37].

3.7.2.1.2 Bölücü hiyerarşik kümeleme

Yukarıdan-aşağıya yaklaşım hiyerarşik kümeleme; başlangıçta tüm nesneleri tek bir küme içinde saklar, her bir nesne kendi içinde bir küme oluşana kadar veya istenen sayıda küme elde edilmesi ya da iki en yakın küme arasındaki uzaklığın eşik değerin üstünde olması gibi belirli bir son koşul sağlanana kadar daha küçük parçalara kümeyi böler. Yüksek seviyede bölümleme için doğru bir seçim gerektiği durumlarda uygulanması zor olduğundan bölücü metod kullanılmaz. Örnek: DIANA (DIvisive ANAlysis) Kaufman tarafından ortaya çıkarılan bölücü hiyerarşik kümeleme metodudur. Bu metodta; Şekil 3.9' da görüldüğü gibi ilk olarak bütün nesneler tek bir kümeye yerleştirilir. Küme, küme içindeki en yakın komşu nesneler arasındaki maksimum öklit uzaklığı gibi bazı kriterler sağlanıncaya kadar bölünür. Küme bölme işlemi, her bir yeni küme sadece tek bir nesne içerinceye kadar devam eder. Toplayıcı ve bölücü hiyerarşik kümeleme yaklaşımlarının her ikisinden de kullanıcı sonlandırma kıstası olarak kümelerin sayısı belirtilebilir. İşlem istenilen küme sayısına ulaştığı zaman hiyerarşik kümeleme işlemi sonlanır. Kümeler arasındaki uzaklık için kullanılan 4 ölçüm aşağıda gösterilmiştir. Bu ölçümlerdeki $|p - p'|$, p ve p' noktaları arasındaki uzaklık, m_i C_i kümesinin ortalamasını, n_i C_i kümesi içindeki nesnelerin sayısını ifade eder [37].

Minimum uzaklık(Minimum Distance): $d_{min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} |p - p'|$

Maksimum uzaklık(Maximum Distance): $d_{max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} |p - p'|$

Ortalama uzaklık(Mean Distance): $d_{mean}(C_i, C_j) = |m_i - m_j|$

Ortalama uzaklık(Average distance): $d_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i} \sum_{p' \in C_j} |p - p'|$

Hiyerarşik kümeleme yaklaşımı basit olsa bile bölme ve birleştirme noktalarının dikkatlice seçilmesi gerekir. Birleştirme ve bölme noktaları doğru bir şekilde seçilmezse düşük kalitede kümeler oluşabilir.

3.7.2.3 BIRCH algoritması

Birleştirilmiş hiyerarşik kümeleme metodu olan BIRCH Zhang tarafından geliştirilmiştir. Bu metod; kümeleme özelliği ve kümeleme özelliği ağacı (CF-Clustering Feature Tree) olmak üzere 2 kavrama dayanmaktadır. BIRCH yeni nesneleri artımlı ve dinamik olarak kümelede etkilidir.

CF ağacı hiyerarşik kümeleme için kümeleme özelliklerini depolayan yükseklik dengeli ağaçtır. CF ağacı dallanma çarpanı ve eşik değeri olmak üzere 2 parametreye sahiptir. Dallanma çarpanı parametresi her bir yapraksız düğüm için çocuklarının maksimum sayısını ifade eder. Eşik değeri parametresi yaprak düğümlerde depolanan alt kümelerin maksimum çapını ifade eder. Bu iki parametre sonuçlanan ağacın boyutunu etkileyebilir [37]. BIRCH algoritması 2 aşamaya sahiptir. Birinci aşamada, bir başlangıç belleğindeki CF ağacını inşa etmek için veritabanını taranır. Çok seviyeli bir sıkıştırma olarak görülebilir ve verinin doğasında olan kümeleme yapısını korumaya çalışır. İkinci aşama da CF ağacının yaprak düğümlerini kümeleme için isteğe göre bir kümeleme algoritması uygulanır.

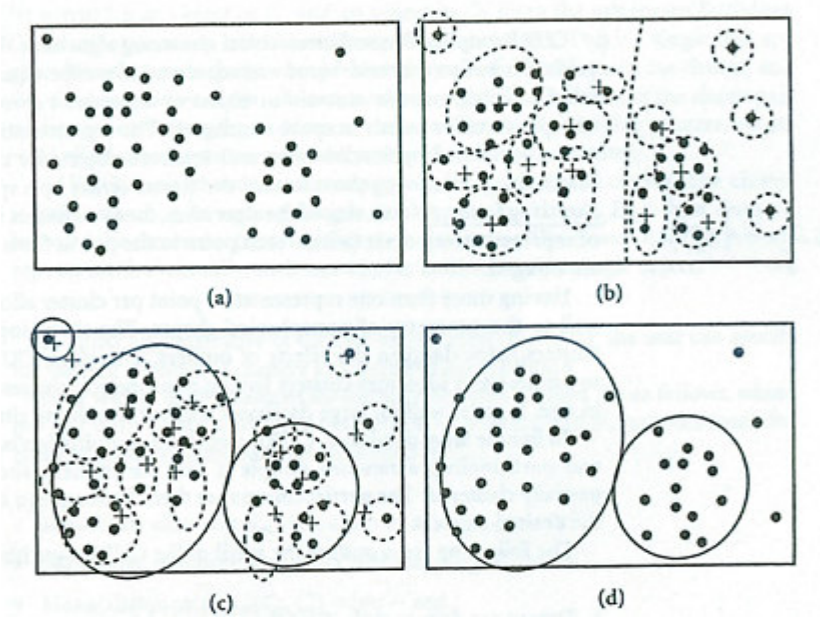
Birinci aşamada CF ağacı nesneler eklenerek dinamik olarak inşa edilir. Bu metod artımlı bir metodtur. Nesne en yakın yaprak girişine eklenir. Eğer alt kümenin çap değeri yaprak düğüme yapılan eklemeden sonra eşik değerinden daha büyük olursa, yaprak düğümü ve belki de diğer düğümler bölünür. Yeni nesneyi ekleme işleminden sonra ağacın köküne doğru bilgi geçirilir. Eşik değerinin değişmesiyle ağacında boyutu değişebilir. Eğer CF ağacının saklanması için gerekli olan bellek boyutu ana belleğin boyutundan daha büyükse daha küçük bir eşik değeri belirlenir ve CF ağacı yeniden inşa edilir. Eski ağacın yaprak düğümlerinden yeni bir ağaç inşa edilir. Böylece ağacın yeniden yapılanma işlemi tüm noktaların okunmasına gerek kalmaksızın yapılır. Bu B+ ağaçlarının oluşturulmasındaki ekleme ve bölme işlemine benzer. Bu yüzden ağaç inşa etmek için veri yalnızca bir kere okunur. Bazı metodlar CF ağacının kalitesini geliştirmek için veri de ek taramalar gerçekleştirebilir. CF ağacı inşa edildikten sonra tipik bir bölümleme algoritması gibi herhangi bir kümeleme algoritması CF ağacı üzerinde 2. aşama için kullanılabilir [37].

BIRCH mevcut kaynaklarla en iyi kümeleri üretmeye çalışır. Ana belleğin limitiyle, I/O işlemleri için gerekli olan zamanı minimize etmek önemli bir husustur. BIRCH çok aşamalı kümeleme tekniğini kullanır. Veri kümesini bir kere tarama iyi bir kümeleme için kazanç sağlarken bir ya da daha fazla ek tarama kaliteyi geliştirmek için kullanılabilir. Algoritmanın hesapsal karmaşıklığı O(n) ' dir. n kümelenmiş nesnelerin sayısını ifade etmektedir. Deneyimler algoritmanın doğrusal ölçülebilirliğinin nokta sayısına uyulmasına ve verinin kümelenmesindeki kalitesine bağlı olduğunu göstermiştir. CF ağacındaki her bir düğüm, boyutuna göre girişlerin sınırlı sayıda bir kısmını tutabilir. Bir CF ağacı düğümü doğal kümeleme için asla uygun olmaz. Eğer kümeler küresel şekilde değilse BIRCH iyi bir şekilde kümelemeyi gerçekleştiremez. Çünkü BIRCH bir kümenin sınırını kontrol etmek için yarıçap veya çap fikrini kullanır [37].

3.7.2.4 CURE algoritması

Guha, Rastogi ve Shim tarafından ilk olarak SIGMOD 1998 konferansında sunulan CURE algoritması birleştirici bir kümeleme metodudur. Birçok kümeleme algoritması ya küresel şekilli ve eşit büyüklükteki kümelerde etkili çalışır ya da uzakta bulunan noktaların varlığında etkili çalışamazlar. CURE, centroid tabanlı ve temsilci-nesne tabanlı kümeleme yaklaşımlarını birleştirir. Kümeyi belirtmek için bir tek centroid ya da temsilci kullanmak yerine, küme sabit sayıda temsilci ile gösterilir. Kümenin temsilci elemanlarını belirtmek için önce iyi saçılmış nesneler belirlenir ve küme merkezine belirli bir oranda yaklaştırılır. Birçok temsilci nokta olması küresel şekilli olmayan noktalar için daha uygun kümeler oluşturulmasını sağlar. Daraltma işlemi ise dışarda kalan noktaların etkisini azaltmakta etkilidir. Büyük veritabanları ile başedebilmek için CURE rasgele örnekleme ve bölümleme kullanır. Önce rasgele örnek bölümlenir, daha sonra da bu bölümler kümelendirilir.

CURE algoritmasında ilk önce rasgele bir örnek kümesi S seçilir. S kümesi bölümlere ayrılır. Her bölüm kısmen kümelenir. Rasgele örnekleme yapılarak dışta kalan noktalar elenir. Kısmi kümeler, yeniden kümelenir. Her yeni kümenin temsilci noktası belirli oranda küme merkezine doğru kaydırılır. Bu noktalar kümenin şeklini temsil eder.



Şekil 3.10: CURE Algoritmasının işleyişi [37].

CURE yüksek kaliteli kümeler oluşturur. Oluşan kümeler karmaşık şekilli ve farklı büyüklükte olabilir. CURE algoritması veritabanını sadece bir kez tarar, yani karmaşıklığı O(n)'dir. CURE algoritmasının başarısı başlangıç parametrelerine büyük ölçüde bağlıdır. Bu nedenle en iyi kümeleme sonucunun bulunabilmesi için algoritmanın aynı veri seti üzerinde birkaç kez tekrarlanması gerekebilmektedir.

CURE algoritması sınıflandırlımış niteliklerle çalışamaz, bunun yerine ROCK (Robust Clustering Algorithm) algoritması kullanılır. ROCK algoritması iki kümenin birbirine benzerliğini, iki küme arasındaki toplam ara bağlantı (interconnectivity) miktarını hesaplayarak ölçer. İki küme arasındaki ara bağlantı miktarı kümeler arasındaki çapraz bağlantı miktarıdır. Her bir bağlantı ise iki noktanın ortak komşularının sayısıdır. Yani kümelerin benzerliği, farklı kümelerdeki noktaların ortak komşularının sayısı ile belirlenir.

3.7.2.5 CHAMELEON algoritması

Chameleon algoritması dinamik modellemeyi hiyerarşik kümelemeye uygular. Kümeleme işleminde eğer iki küme arasındaki ara bağlantı ve yakınlık değerleri kümelerin kendi içlerindeki ara bağlantı ve yakınlık değerleri ile yüksek oranda ilişkili ise bu kümeler birleştirilir. Bu birleştirme işlemi doğal ve homojen kümelerin

ortaya çıkarılmasını sağlar ve benzerlik fonksiyonunun tanımlanabildiği her veri tipi için uygulanabilir. Chameleon, hem CURE hem de ROCK algoritmalarının eksik taraflarını kapatır. CURE algoritması iki küme arasındaki toplam ara bağlantı miktarını göz ardı ederken, ROCK algoritması da iki kümenin birbirine ne kadar yakın olduğunu göz ardı eder.

Chameleon algoritması önce graf bölümleme algoritması kullanarak veriyi çok sayıda küçük alt kümeye ayırır. Daha sonra bu küçük alt kümeleri birleştirerek özel kümeleri oluşturur. Birbirine en çok benzeyen alt kümeleri belirlerken hem alt kümelerin kendi içlerindeki ara bağlantı ve yakınlık değerlerini hem de alt kümelerin kendi aralarındaki alt bağlantı ve yakınlık değerlerini kullanır. Böylece algoritma kendini verinin yapısına göre ayarlar.

3.7.3 Yoğunluk tabanlı metotlar (density-based methods)

Yoğunluğa dayalı metotlar düzensiz(arbitrary) şekildeki kümeleri incelemek için geliştirilmiştir. Bu metotlar kümeleri, veri uzayındaki düşük yoğunluklu(gürütülü) alanlarca ayrılmış yoğun nesne alanları olarak görürler. DBSCAN (Density-Based Spatial Clustering of Applications with Noise), OPTICS (Ordering Points To Identify the Clustering Structure) ve DENCLUE (Clustering Based On Density Distribution Functions) algoritmalardır.

3.7.3.1 DBSCAN algoritması

DBSCAN yeterince yoğun görünen alanlardan kümeler oluşturur ve düzensiz kümeleri gürültülü uzaysal veri tabanlarında inceler. DBSCAN bir öbeği yoğunlukça bağlanmış noktaların en büyük kümesi olarak tanımlar.

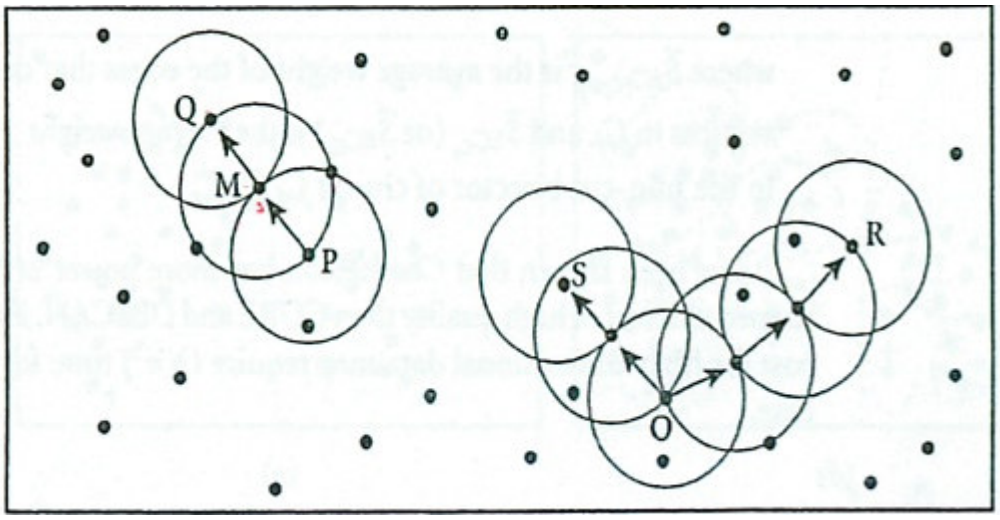
Yoğunluk tabanlı kümeleme bir takım yeni terimler gerektirmiştir.

- Bir nesnenin ϵ çapı içindeki komşuluğuna o nesnenin ϵ-komşuluğu adı verilir.
- Bir nesnenin ϵ-komşuluğu en azından asgari sayı olarak kabul edilen MinPts adet nesne içeriyorsa bu nesneye çekirdek nesne denir.

● Verilen bir D nesne kümesi içindeki p nesnesi çekirdek nesne olan q'nun ϵ-komşuluğunda ise q'dan doğrudan yoğunlukça erişilebilirdir.

● Verilen bir D nesne kümesi içindeki p nesnesi $p^1 = q \Lambda p^n = p$ olmak üzere $p^1,\ldots,p^n$ nesne zinciri içindeki p^{i+1} nesnesi $(1 \leq i \leq n , p^i \in D)$ p^i'den ϵ ve MinPts'ye göre doğrudan yoğunlukça erişilebilir olduğu durumlarda ϵ ve MinPts'ye göre q'dan yoğunlukça erişilebilirdir. Bu özellik çekirdek nesneler dışında asimetriktir.

● D nesne kümesi içinde p ve q'nun, ϵ ve MinPts'ye göre doğrudan erişilebilir olduğu bir o nesnesi varsa p, q'ya ϵ ve MinPts'ye göre yoğunlukça bağlıdır. Bu özellik simetriktir. Örneğin MinPts = 3 olan ϵ çaplı birer çemberden oluşan kümeler Şekil 3.11' daki gibi olduğunu düşünelim.



Şekil 3.11: Yoğunluk tabanlı kümelemede yoğunluk erişilebilirliği [37].

● M, P, O ve R ϵ komşuluklarında en azından üçer nokta bulundurduğu için çekirdek nesnelerdir.

● Q, M' den; M, Q' dan doğrudan erişilebilirdir. Bu hükümlerin tersleri de doğrudur.

● Q, P'den dolaylı olarak yoğunlukça erişilebilirdir çünkü Q, M'den ve M de P'den doğrudan yoğunlukça erişilebilirdir. Ancak Q bir çekirdek nesne olmadığı için bu önermenin tersi geçerli değildir.

● O, R, S birbirleriyle yoğunlukça bağlıdır.

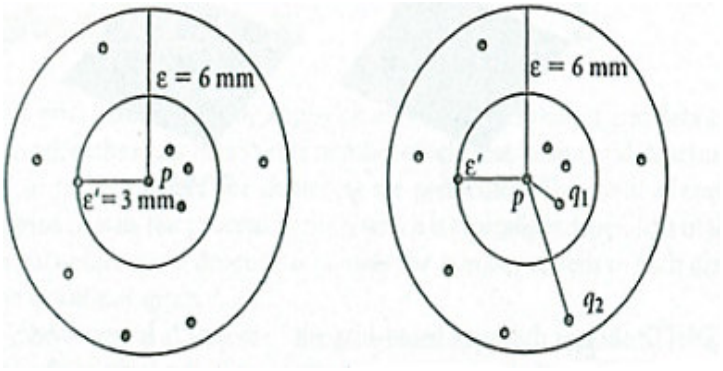
DBSCAN veri tabanındaki her bir noktanın ϵ-komşuluğunda MinPts sayıda nesne bulunup bulunmadığına bakarak kümeler oluşturur. Eğer veri tabanında uzaysal

indisleme kullanılıyorsa DBSCAN' in karmaşıklığı n veri tabanındaki nesne sayısı olmak üzere O(nlogn) olur. Diğer durumlarda karşmaşıklık $O(n^2)$' dir.

3.7.3.2 OPTICS algoritması

DBSCAN, nesneleri MinPts ve ε gibi verilen giriş parametrelerine göre kümelendirmesine rağmen kabullenilebilir kümelerin keşfinde kullanılacak olan parametreleri seçme sorumluluğunu kullanıcıya bırakmaktadır. Bu sorun diğer kümeleme algoritmalarının da sorunudur. Birçok algoritma giriş parametrelerine karşı duyarlıdır. Farklı giriş parametreleri veri kümesinin çok farklı kümelenmesine neden olabilmektedir. Bu sorunun üstesinden gelmek için OPTICS adı verilen kümeleme analiz yöntemi ortaya konmuştur. OPTICS, dış destekli bir veri kümeleme yapmak yerine etkileşimli ve otomatik bir küme analizi için artımlı bir küme sıralama yapar.

DBSCAN' de elde edilen kümelemelerde MinPts sayısı sabit olduğunda yüksek yoğunluklu kümelerin düşük yoğunluklu kümeler tarafından tamamen kapsandığı görülür. Nesneler farklı bir sıra ile tarandığında ardı ardına farklı kümeler oluşturulabilmektedir. Her bir nesne için nesnenin çekirdek uzaklığı (core-distance) ve erişilebilirlik uzaklığı (reachability-distance) olmak üzere 2 tane değerin tutulması gerekir. p nesnesinin çekirdek uzaklığı, p' yi çekirdek yapan en küçük ε' değeridir. q' nun p' ye göre erişilebilirlik uzaklığı, p' nin çekirdek uzaklığı ile p ile q' nun Öklit uzaklıklarından daha büyük olanıdır. Şekil 3.12' da çekirdek uzaklığı ve erişilebilirlik uzaklığı kavramları net bir şekilde gösterilmiştir. $\varepsilon = 6$ mm ve MinPts=5 olduğunu düşünelim. p nesnesinin çekirdek uzaklığı dördüncü en yakın veri noktası ve p nesnesi arasındaki uzaklıktır(ε') . q1' in p nesnesine göre erişilebilirlik p'uzaklığı, p' den q1' e Öklit uzaklığı büyük olduğundan p' nin çekirdek uzaklığıdır. q2' nin p' ye göre erişilebilirlik uzaklığı p' den q2' ye Öklit uzaklığıdır. Çünkü bu değer p'nin çekirdek uzaklığından daha büyüktür.



Şekil 3.12: Çekirdek uzaklığı ve erişilebilirlik uzaklığı [37].

OPTICS algoritması veritabanındaki nesnelerin sıralamasını oluşturur ve her bir nesnenin çekirdek uzaklığı depolar. Ayrıca erişilebilirlik uzaklığını da depolar. Algoritmanın amacı bu sıralı bilgiye dayanarak kümeleri çıkarmaktır. Bu bilgiler ε' den küçük her ε' uzaklığından bir küme oluşturmak için yeterlidir.

3.7.3.3 DENCLUE algoritması

DENCLUE algoritması yoğunluk dağılım fonksiyonlarına dayalı olarak kümeleme yapan bir metottur.

DENCLUE algoritması aşağıdaki temellere dayanır:

- Her bir veri noktasının baskınlığı, baskınlık fonksiyonu (influence function) adı verilen ve veri noktasının komşuluğu üzerindeki etkisini tanımlayan matematiksel bir model kullanılarak modellenir.
- Veri alanının toplam yoğunluğu tüm veri noktalarının baskınlık fonksiyonlarının toplamı alınarak düzlemsel olarak modellenebilir.
- Kümeler toplam yoğunluk fonksiyonunun yerel maksimum noktaları olan yoğunluk çıkarıcılar (density attractors) ile matematiksel olarak karar verilebilir.

X ve y d boyutlu F^d uzayında birer nesne olsun. Y' nin x üzerindeki baskınlık fonksiyonu olan $f_B^y : F^d \longrightarrow R_0^+ :$

$$f_B^y(x) = f_B(x, y) \tag{3.15}$$

Baskınlık fonksiyonu, komşuluktaki iki nesne arasındaki uzaklığa bakılarak karar verilebilen herhangi düzensiz bir fonksiyon olabilir. Öklit uzaklık fonksiyonu gibi uzaklık fonksiyonu d(x,y), simetrik bir fonksiyon olmalıdır. Kare dalga baskınlık fonksiyonu aşağıdaki gibi hesaplanır:

$$f_{Square}(x,y)= 0 \text{ if } d(x,y)>\sigma, \text{ diğer } 1. \tag{3.16}$$

Gaussian baskınlık fonksiyonu aşağıdaki gibidir:

$$f_{Gauss}(x,y)=e^{-\frac{d(x,y)^2}{2\sigma^2}} \tag{3.17}$$

Toplam yoğunluk fonksiyonu ise tüm veri noktalarının baskınlık fonksiyonlarının toplamıdır:

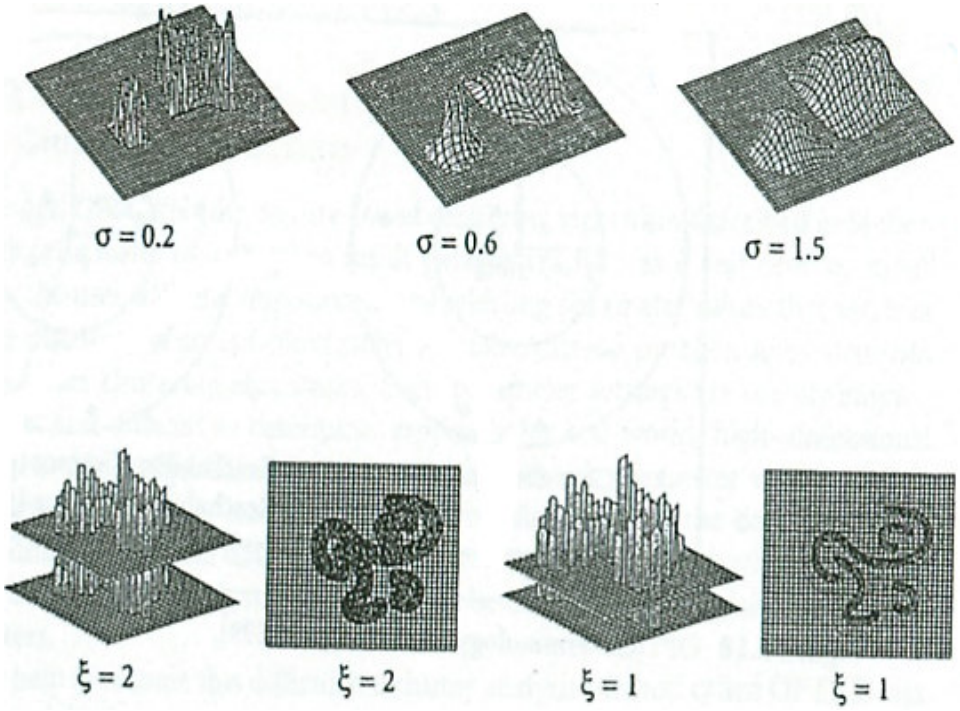
$$f_B^D(x)=\sum_{i=1}^{n} f_B^{x_i}(x) \tag{3.18}$$

Şekil 3.13, 2 boyutlu bir veri kümesi, Gauss yoğunluk fonksiyonu ve yoğunluk çıkartıcıyı göstermektedir.

Şekil 3.13: 2 boyutlu veriler için olası yoğunluk fonksiyonu [37].

Bu modelleme ile merkez tanımlı kümeler(center-defined cluster) ve düzensiz şekilli kümelerde (arbitrary-shape cluster) tanımlanabilmektedir. Bir x^* yoğunluk çıkartıcısı için merkez tanımlı bir küme x^* tarafından yoğunluğu çıkartılmış ve x^*'daki yoğunluk fonksiyonu bir ξ eşiğinden daha az olan bir C alt kümesidir. Düzensiz

şekilli küme ise C' lerin, her biri ξ eşiğini aşan yoğunluk fonksiyonu çıkışları olan yoğunlukları çıkartılmış, bir bölgeden diğerine bir P yolu bulunduran ve bu yol üzerindeki her bir noktanın yoğunluk fonksiyonu sonucu ξ eşiğini karşılayan kümesidir. Şekil 3.14 merkez tabanlı ve düzensiz şekilli kümelerin örneklerini göstermektedir.



Şekil 3.14: Merkez tabanlı ve düzensiz şekilli kümelerin örnekleri [37].

DENCLUE algoritmasının diğer kümeleme algoritmalarına göre birçok avantajı vardır. Bunlar;

- Kesin matematiksel bir yapıdır ve diğer kümeleme metotlarını genelleştirir.
- Gürültüsü fazla olan veri kümelerini iyi kümeleme özelliğine sahiptir.
- Çok boyutlu veri kümelerinde, düzensiz şekilli kümelerin ufak bir matematiksel modelini sunar.
- Izgara hücreleri kullanır fakat yalnızca veri içeren hücreler hakkında bilgileri tutar. Bu hücreleri bir ağaç yapısında tutar. Bu yönüyle diğer algoritmalardan daha hızlı çalışır.

- Bu metotta gürültü eşiği ξ ve yoğunluk parametresi σ dikkatli bir şekilde seçilmelidir. Bu parametreler kümeleme sonuçlarının kaliteli olmasında önemli derece de etkilidir.

3.7.4 Grid-tabanlı metotlar (grid-based methods)

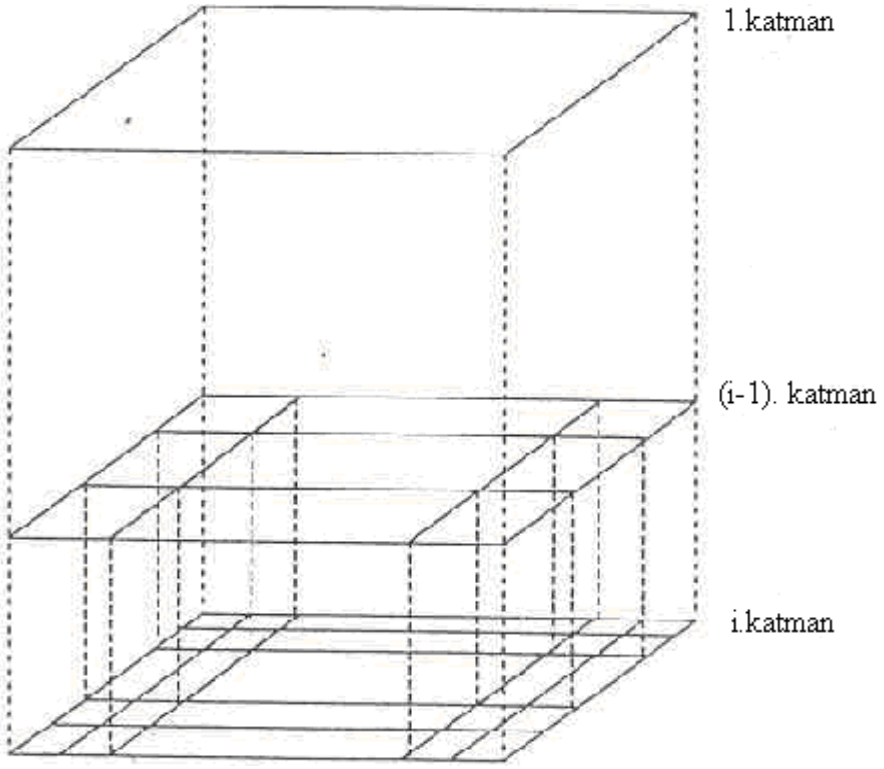
Grid-tabanlı kümeleme yaklaşımı çok çözümlü grid veri yapısını kullanır. Kümeleme yapılacak alanın sonlu sayıda hücrelere bölünmesini sağlar. Grid-temelli kümeleme yaklaşımının ana avantajı birbirinden bağımsız sayıda veri nesnelerinde hızlı işlem zamanıdır. STING (Statistical Information Grid), WaveCluster (Clustering Usign Wavelet Transformation) ve CLIQUE (Clustering High-Dimensional Space) metotları grid-tabanlı metotlardır. STING grid hücreleri içine depolanan istatistiksel bilgiyi araştırır. WaveCluster metodu, wavelet dönüşüm metodunu kullanarak nesneleri kümeler. CLIQUE metodu ise yüksek boyulu veri alanlarını kümelemek için grid ve yoğunluk tabanlı yaklaşımı temsil eder. Burada STING algoritması üzerinde durulacaktır.

3.7.4.1 STING algoritması

STING uzayı dikdörtgensel hücrelere bölen bir tekniktir. Bu hücreler hiyerarşik yapıdadır. Üst seviyedeki her bir hücre bir sonraki alt seviye de parçalanmış hücrelerden oluşur. Her bir grid hücresindeki niteliklerle ilişkili istatistiksel bilgi işlenir veya depolanır [37].

Şekil 3.15' de STING kümelemenin hiyerarşik yapısını göstermektedir. Üst seviyedeki hücre istatistiksel parametreleri, alt seviyedeki hücre istatistiksel parametrelerden kolayca hesaplanabilir. Bu parametreler; nitelik bağımsız parametre, sayı (count), nitelik-bağımlı parametreler (attribute-dependent parameters), ortalama (m-mean), standart sapma (s-standart deviation), minimum (min-minimum), maksimum (max-maximum) ve normal, üstel, hiçbiri gibi dağılım tipleridir. Veri, veri tabanına kaydedilirken alt seviye hücrelerdeki count, m, s, min ve max parametreleri direkt olarak hesaplanır. Dağılım değeri dağılım tipi biliniyorsa kullanıcı tarafından hesaplanabilir veya X^2 testi gibi hipotez testinden elde edilebilir.

Üst seviyedeki hücrenin dağılım tipi, aynı alt seviyedeki hücrelerin birleştirilerek eşik filtreleme işleminden geçirilerek bulunabilir. Eğer alt seviye hücrelerdeki dağılım birbiri ile uyuşmuyorsa, eşik testi başarısız olur ve üst seviye hücredeki dağılım tipi hiçbiri olarak ayarlanır [37].



Şekil 3.15: STING kümeleme için hiyerarşik yapısı [37].

"İstatistiksel bilgi sorgu cevabı için nasıl bir yarar sağlar?" ilk olarak sorgu cevap işleminin başlayacağı hiyerarşik yapı içindeki katmana karar verilir. Bu katman genellikle küçük sayıdaki hücreleri içerir. Geçerli katman içindeki her bir hücre için hücrenin verilen sorguya ilgisine göre güven aralığı (confidence interval) hesaplanır. İlgisiz olan hücreler ileriki adımlar için silinir. Bu işlem en alt seviyeye varana kadar devam eder. Sorgu şartı sağlanırsa, hücrelerin ilgili bölgeleri döndürülür. İlgili veri, sorgunun gereklerini yerine getirene kadar yeniden düzeltilir ve işlenir.

STING metodunun sağladığı yararlar aşağıda maddeler halinde anlatılmıştır [37]:

- Grid-tabanlı hesaplama sorgu bağımsızdır. Her bir hücre içinde depolanan istatistiksel bilgi grid hücre içindeki özet bilgileri içerir.
- Grid yapısı, paralel işleme ve güncelleştirmelere uygundur.

- Metodun verimi asıl avantajıdır. STING hücrelerin istatistiksel parametrelerinin hesaplamak için veritabanına bir kere gider. Kümeleri oluşturma zaman karmaşıklığı O(n)' dir. Burada n nesnelerin toplam sayınsı ifade etmektedir. Hiyerarşik yapıyı oluşturduktan sonra, sorgu işleme zamanı O(g) olur. Buradaki g, en alt seviyedeki grid hücrelerinin toplam sayısıdır.

STING metodunun kalitesi, grid yapısının en alt katmanındaki taneciğe bağlıdır. Tanecikler hassas ise işlem maliyeti artar. Ayrıca grid yapısının en alt katmanın kalın olması kümeleme analiz kalitesini azaltabilir. STING ana hücrenin oluşumu için çocuk ve komşularını ile olan ilişkilerini düşünmez. Kümeleme sınırları düşey veya yataydır, diyagonal değildir [37].

3.7.4.2 WaveCluster algoritması

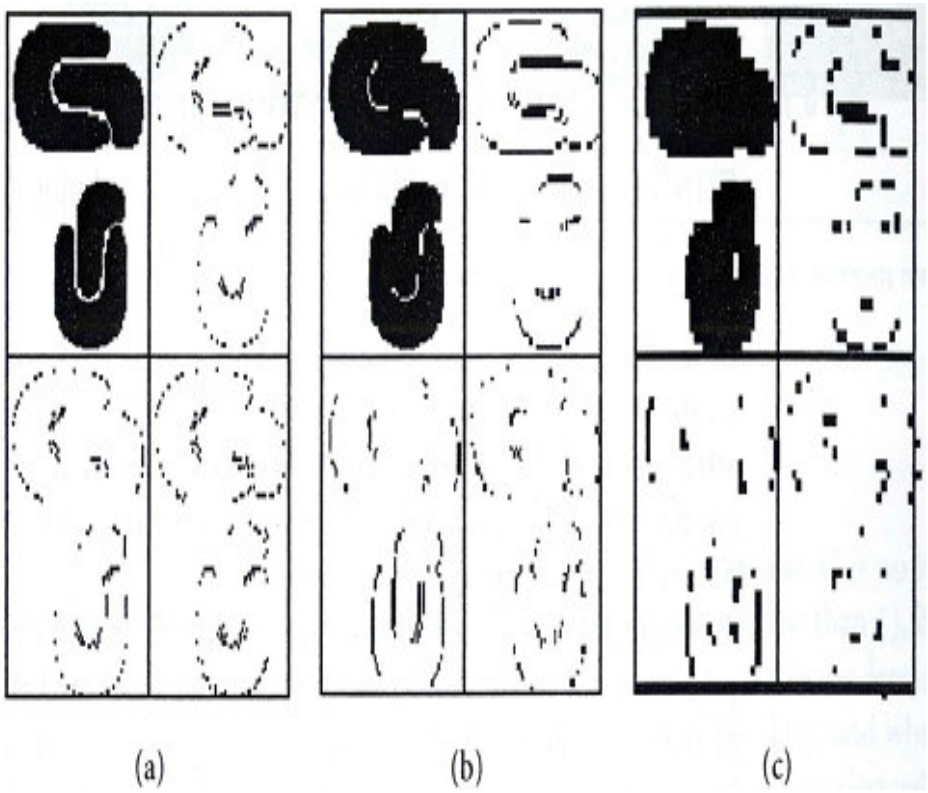
WaveCluster algoritması, çoklu çözüm kümeleme algoritmasıdır. İlk olarak veri uzayını çok boyutlu grid yapısına dönüştürür. Wavelet dönüşümünü kullanarak yoğun bölgeleri bularak orijinal nitelik uzayında dönüşüm yapar [37].

Wavelet dönüşümü, işareti alt frekans bantlarına ayrıştıran işaret işleme tekniğidir. Wavelet modeli, n defa dönüşüm yaparak bir boyutlu sinyali n boyutlu işaretlere dönüştürürebilir. Wavelet dönüşümüne başvurularak, farklı çözüm seviyelerindeki nesneler arasındaki göreceli uzaklığı saklamak için veri dönüştürülür. Bu doğal kümelerin daha fazla ayırt edilebilir olmasını sağlar. Yeni alan içindeki yoğun bölgeleri arayarak kümeler tanımlanabilir [37].

Wavelet dönüşümünün kümelemede birçok yararı vardır. Denetimsiz kümeleme(unsupervisesd clustering) sağlar. Nokta kümelerin olduğu bölgeleri vurgulayarak şapka şeklinde filtreler kullanır. Aynı zamanda zayıf bilgileri küme sınırlarının dışına atar. Veri kümelerini otomatik olarak belirler ve onların dışındaki bölgeleri temizler. Wavelet dönüşümü sınır dışında kalan verileri otomatik olarak temizler. Wavelet dönüşümünün çok çözümlü özelliği, kümelerdeki farklı seviyelerdeki doğruluğu bulmada yardım edebilmektedir. Şekil 3.16' de 2 boyutlu nitelik uzayındaki bir örneği gösterir. Resim içindeki her bir nokta uzaysal veri

kümesindeki bir nesnenin nitelik veya özellik değerlerini gösterir. Şekil 3.17' de farklı çözünürlükteki Wavelet dönüşüm sonuçları gösterilmektedir. Her bir seviye alt dört banda ayrılmıştır. Sol üst band her veri noktası üzerindeki ortalama komşuluğu, sağ üst band verinin yatay kenarlarını, sol alt band verinin dikey kenarlarını ve sağ alt band köşeleri vurgular [37].

Şekil 3.16: 2 boyutlu nitelik uzayındaki bir örnek [37].



Şekil 3.17: Farklı çözünürlükteki Wavelet dönüşüm sonuçları [37].

Wavelet tabanlı kümeleme çok hızlıdır ve hesaplama karmaşıklığı O(n)' dir. n veritabanındaki nesne sayısıdır. WaveCluster grid tabanlı ve yoğunluk tabanlı bir algoritmadır. WaveCluster iyi bir kümeleme gereksinimlerinin çoğunu karşılar.

Geniş veri kümelerinde verimlidir, düzensiz şekilli kümeleri keşfeder, sıra dışılıkları başarılı bir şekilde tutar, giriş sırasına duyarlı değildir, küme sayısı gibi giriş parametrelerinin belirlenmesine gerek duymaz. Deneme çalışmalarında, WaveCluster' ın BIRCH, CLARANS ve DBSCAN' den verim ve kümeleme kalitesi olarak daha yüksek performansa sahip olduğu bulunmuştur [37].

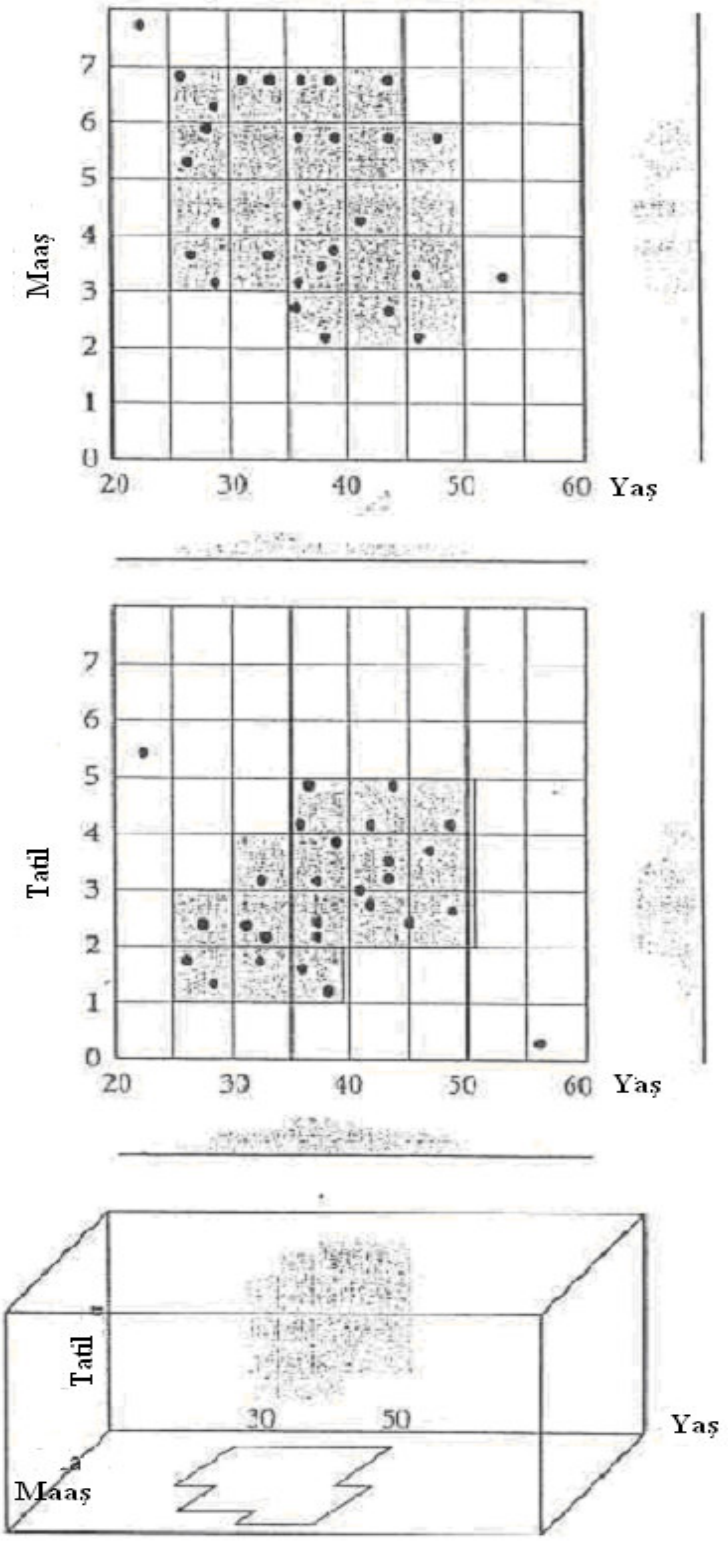
3.7.4.3 CLIQUE algoritması

CLIQUE yoğunluk tabanlı ve grid tabanlı kümelerin birleşmesinden oluşan bir algoritmadır. Büyük veritabanlarında yüksek boyutlu veri kümelemede yararlı olan bir algoritmadır. Dağınık örüntülü veri kümelerinde seyrek ve kalabalık olan alanları tanımlar [37].

CLIQUE iki adımda çalışır. Birinci adımda CLIQUE, üst üste çakışmayacak şekilde veri uzayını n boyutlu dikdörtgen şeklinde parçalara ayırır. Şekil 3.18' te yoğun dikdörtgen parçaları, yaşa göre maaş ve tatil boyutları olarak gösterilmiştir. Alt alanlar ile yüksek boyutlu yoğun parçaların kesişmesi durumunda aday arama alanı(candidate search space) oluşmaktadır. Aday arama alanının tanımlanması Apriori özelliğine dayanmaktadır. Apriori özelliği arama alanı içindeki öncelikli bilgiyi kullanır. Bu şekilde alan budanarak parçalara bölünür. CLIQUE algoritması için uyarlanan özellik, şu şekilde ifade edilebilir: Eğer k boyutlu parça yoğun ise k–1 boyutlu parçaya bakılır. k–1 boyutlu parçada yoğunluk yok ise k boyutlu parçada da yoğunluk yoktur. Sonuç olarak, k–1 boyutlu uzayda bulunan yoğun parçalardan k boyutlu uzayda bulunan olası ve aday yoğunluklu parçalar yaratılabilir. Genelde sonuçlanan alan orijinal alandan daha küçük olacaktır. Bu yoğunluk parçaları kümelere karar verme için kullanılır. İkinci adımda, CLIQUE her bir küme için minimal bir tanımlama yapar. Her bir küme için, bağlantılı yoğun parçaların (connected dense units) kümesini kapsayan maksimum bölgeye karar verilir. Her bir kümenin minimal kapsamına karar verir [37].

CLIQUE otomatik olarak yüksek boyutlu alt alanları ve her alt alan yüksek yoğunluktaki kümeleri bulur. Algoritma giriş satırlarının sırasına duyarlıdır ve herhangi bir geleneksel veri dağılımını tahmin edemez. Giriş boyutuna göre lineer

ölçeklenir ve verideki boyut sayısı arttıkça iyi ölçekleme özelliğine sahiptir. Algoritmanın basitliği pahasına kümeleme sonucunun doğruluğu azalabilir.



Şekil 3.18: CLIQUE algoritmasının işleyişi [37].

3.7.5 Model tabanlı kümeleme metotlar (model-based clustering methods)

Model-tabanlı kümeleme metotları, verilen veri ve bazı matematiksel modeller arasında uygunluğu optimize etmeye çalışır. Model-tabanlı kümeleme metotlarının istatistiksel yaklaşım (Statistical approach) ve nöron ağları yaklaşımı (Nueral Network Approach) olmak üzere iki tane yaklaşımı vardır. Nöron ağları yaklaşımında kümeleme, her kümeyi bir örnekleyici (exemplar) olarak gösterir. Örnekleyici bir kümenin prototipi olarak davranır ve nesne veya özel veri örneği ile bir ilişkiye sahip olması gerekmez. Uzaklık ölçümüne dayalı olarak örnekleyici çok fazla benzeyen bir kümeyi yeni nesneler dağıtabilir. Kümeye atanan nesnelerin nitelikleri küme örnekleyicisinin niteliklerinden tahmin edilebilir. Burada İstatistiksel yaklaşım üzerinde durulmuştur [37].

3.7.5.1 İstatistiksel yaklaşım

Kavramsal kümeleme (Conceptual clustering), verilen etiketlenmemiş nesne kümeleri ve nesnelerdeki sınıf şemaları üretilmesi ile oluşan makine öğrenimli bir formdur. Benzer nesne gruplarını tanımlayan geleneksel kümelemenin (conventional clustering) tersine, kavramsal kümeleme sınıf veya kavramı gösteren her bir grup için karakteristik tanımlamaları bulur. Kavramsal kümeleme iki adımdan oluşur. Birinci olarak kümeleme ardından nitelendirme (characterization) yapılır. Kümeleme kalitesi nesneler için fonksiyon değildir. Bunun yanında, kümeleme kalitesi kavramların genellik ve basitlik gibi etkenlerini de içerir. Tüm kavramsal kümeleme metotları kavramlara ve kümelere karar vermede olasılık ölçümlerinin kullanan istatistiksel yaklaşımı benimsemektedirler. COBWEB basit ve popüler olan artışsal kavramsal kümeleme metodudur. Giriş değişkenleri kategorik nitelik-değer çiftleri tarafından tanımlanır. COBWEB sınıflandırma ağacı içinde hiyerarşik kümeleme oluşturur [37]. COSWEB, ağacın oluşturulmasına rehberlik etmek için kategori yararı (CU-Category Utility) adında heuristik bir değerlendirme ölçümü kullanır.

$$\frac{\sum_{k=1}^{n} P(C_k)\left[\sum_i \sum_j P(A_i = V_{ij} | C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2\right]}{n} \tag{3.19}$$

n düğüm sayısı, $[C_1, C_2, \ldots, C_n]$ konsept veya kategori, CU verilen bölemeden doğru olarak tahmin edilebilen beklenen sayıda nitelik değerinde yükselmedir. CU, sınıf benzerlik ve sınıflar arası benzersizlik hakkında bilgi verir.

COSWEB artışsal olarak nesneleri sınıflandırma ağacının içine dâhil eder. Yeni bir nesne verildiğinde, COSWEB nesneyi sınıflandırma ağacı içinde nereye ekleyeceğine belli aşamalardan geçerek karar verir. COSWEB en iyi düğümü bulana kadar ağaçta aşağıya doğru ilerler ve yol boyunca da değerleri günceller. Geçici olan nesneyi tüm düğümlere yerleştirir ve yerleştirilen bölüm için CU değerini hesaplar. En büyük CU değerinin olduğu yer nesnenin yeridir. Eğer verilen nesne ağaçta bulunan her bir konsepte uzaksa ve yeni bir düğüm yaratmak daha iyi ise COSWEB yeni bir düğüm oluşturmak için CU' yu ölçer. Bu değer var olan düğümler ile karşılaştırılır. En yüksek CU değeri ile beraber yeni bir sınıf yaratılır veya var olan sınıfın içine yerleştirilir. COSWEB bölüm içindeki sınıfların sayısını otomatik olarak ayarlama yeteneğine sahiptir. Nesnelerin giriş sırasına göre iki operatör yüksek hassasiyet gösterir. COSWEB bu giriş sırasına olan duyarlılığı azaltmak için iki ek operatöre sahiptir. Bunlar birleştirme ve ayırmadır. Nesne dâhil edildiğinde, iki en iyi düğüm tek bir sınıf içine yerleştirilir. Sonra COSWEB, en iyi olan düğümü ayrıştırır. Bu kararlar CU değerine bakılarak alınır. Birleştirme ve ayrıştırma operatörleri COSWEB' in direk olarak arama yapmasını izin verir.

COSWEB' in sınır değerleri vardır. Birincisi, ayrık nitelikler üzerindeki olasılık dağılımları istatistiksel olarak birbirinden bağımsız sayılır. Buna rağmen bu varsayım her zaman doğru değildir. Nitelikler arasında karşılıklı bir ilişki vardır. Kümelerin olasılık dağılımlarının temsili, kümelerin depolanması ve güncellenmesini pahalı kılar. Sınıflandırma ağacı dengeli bir ağaç olmadığından zaman ve alan karmaşıklığı giriş verilerine bağlı olarak düşebilir.

3.7.6 Sıradışılık analizi (outlier analysis)

Çoğu kez elde edilen veri nesneleri genel davranışa veya verinin modeline uymayabilir. Bu veri nesneleri, geri kalan veri kümesinden tamamen farklı veya uyumsuz olabilir, buna sıra dışılık (Outliers) denir.

Sıradışılık ölçüm veya çalışma hatalarıyla oluşabilir. Sıradışılığa bir örnek verecek olursak; bir şirketin çalışkan olan müdürünün maaşının firma içinde bulunan diğer çalışanların maaşları arasında sıra dışı bir veri olarak görülebilir [37].

Çoğu veri madenciliği algoritması sıradışılıkların etkisini minimize edebilir ya da tamamen yok edebilir. Fakat bu sıradışılıkların yok edilmesi işlemi, bir insanın gürültüsü başka bir insanın sinyali olabileceğinden gizli ve önemli bilgini kayı ile sonuçlanabilir. Sıradışılık, telekomünikasyon servisi ve kredi kartlarının usulsüz olarak kullanımı gibi sahtecilik tespitinde kullanılabilmektedir [37].

Sıradışılık madenciliği (Outlier mining), n verilen veri nesnelerinin sayısı ve k beklenen sıradışılık sayısı olacak şekilde geri kalan veri kümesi içinde tutarsız, seyrek ve farklı olan en büyük k tane nesnenin bulunması olarak tanımlanmaktadır. sıradışılık madenciliğinin iki alt problemi vardır. Birinci problem, verilen veri kümesi içinde ne verisinin tutarsı olduğunu tanımlamak, ikincisi ise sıradışılıkların belirlenmesi için verimli olan bir metot bulmaktır. Sıradışılıkların bulunmasında istatistiksel yaklaşım (statistical approach), uzaklık tabanlı yaklaşım (distance-absed approach) ve sapma tabanlı yaklaşımlardan (deviation-based approach) yararlanılmaktadır [37].

İstatistiksel-tabanlı yaklaşım verilen veri kümesi için dağılım veya olasılık modelini varsayar ve uyumsuzluk testini (discordancy test) uygulayarak sıradışılıkları tanımlar. Testin uygulaması aşamasında, veri kümesi parametreleri hakkında bilgi, dağılım parametreleri hakkında bilgi ve beklenen sıra dışılık sayısı hakkında ön bilgi gerekir. İstatistiksel yaklaşımda, testler tek bir nitelik için yapılmaktadır fakat çoğu veri madenciliği problemi çok boyutlu alanda sıradışılıkların bulunmasını gerektirmektedir. Verinin dağılımının bilinmediği durumlarda istatistiksel yaklaşım sıra dışılıkların bulunmasını garanti etmez [37].

Uzaklık-tabanlı sıradışılık denetimi, istatistiksek yaklaşımın eksiklerinin kapatmak amacıyla ortaya atılmıştır. Uzaklık-tabanlı sıradışıları yeteri kadar komşusu olmayan nesneler olarak düşünülebilir. Komşular verilen nesneye göre uzaklığı tanımlar. İstatistiksel tabanlı metotlarla karşılaştırıldığında, uzaklık tabanlı sıra dışılık

denetimi standart dağılımlar için uyumsuzluk testi fikrini genelleştirilmesidir. Uzaklık-tabanlı sıra dışılık denetimi, aşırı hesaplama gözlenen dağılımın standart dağılıma ve uyuşmazlık testine uyarlanması ile ilgili olduğundan aşırı hesaplamalardan kaçınır. Büyük veri kümeleri için uygundur fakat çok boyutlu veriler için uygun değildir.

Sapma-tabanlı sıradışılık denetimi, istisnai nesneleri tanımlamak için istatistiksel testleri ve uzaklık tabanlı ölçümleri kullanmaz. Bunun yerine grup içindeki nesnelerin başlıca karakteristiklerini sorgulayarak sıradışıları tanımlar. Algoritma analiz için veri kümesinden alt kümeleri sıralı olarak seçer. Her bir alt kümes için, sıralı olarak önceki alt küme ile o anki alt küme arasındaki benzemezlik farkına karar verir.

3.8 Kümeleme Analizinin Kullanıldığı Alanlar

Kümeleme analizi birçok farklı alanda kullanılmaktadır. Kümeleme analizinin kullanıldığı belli başlı alanlar aşağıda belirtilmiştir:

• Biyoloji: Yaşayan varlıkların taksonomosini oluşturmak için biyologlar uzun yıllar harcamışlardır. Bunlar; alem, filum, sınıf, tür, aile gibi kümelerdir. Son zamanlarda biyologlar var olan genetik bilginin analizi için kümeleyi uygulamışlardır. Örneğin; kümeleme benzer fonksiyonlara sahip gen gruplarını bulmak için kullanılmaktadır.

• Bilgi Çıkarma (Information Retrieval): Web milyonlarca web sayfasından oluşur ve arama motoruna yapılacak bir sorgu binlerce sayfa ile geri dönebilir. Kümeleme bu arama sonuçlarını küçük sayıdaki kümelere gruplamak için kullanılabilir. Örneğin; film için sorgu yaptığımızda web sayfalarını eleştiriler, fragman, yıldızlar ve tiyatrolar kategorilerine gruplayabilir. Her kategori(küme) hiyerarşik bir yapı oluşturarak alt kategorilere bölünebilir. Bu yapı kullanıcıya daha ileri keşifler için yardımcı olur.

• İklim (Climate): Kümeleme analizi kutup bölgelerinin atmosfer basınçlarını ve kara iklimleri üzerinde belirgin etkiye sahip okyanus alanlarının örüntülerini bulmak için kullanılmaktadır.

• Psikoloji ve İlaçlar (Psychology and Medicine): Bir hastalık veya bir durumun birçok varyasyonu vardır ve kümeleme analizi bu farklı alt kategorileri tanımlamak için kullanılabilir. Örneğin kümeleme analizi değişik şekillerdeki depresyon tiplerini tanımlamak için kullanılmaktadır. Bir hastalığın geçici dağılımı veya uzaysal dağılımındaki örüntüleri bulmak için kullanılır.

• İş (Business): Kuruluşlar, müşterilerilerinin ve potansiyel müşterilerinin hakkında büyük miktarda bilgi toplarlar. Kümeleme ilave analizler yapmak ve piyasa faaliyetleri için müşterileri küçük gruplara bölmede kullanılabilir.

• Özetleme (Summarization): Regreasyon ve PCA gibi birçok veri analizi teknikleri zaman ve alan karmaşıklığına sahiptir ve bu gibi teknikler geniş veri kümeleri için uygun değildir. Giriş veri kümesine bu algoritmaları uygulamak yerine sadece küme prototiplerini içeren azaltılmış veri kümesine başvurulabilir. Analizin çeşidine dayalı olarak prototiplerin sayısı ve veriyi temsil eden prototipin doğruluğu, anlalizi tüm veri üzerine uygulayarak elde edilecek sonuçlarla prototip üzerine uygulanarak elde edilecek sonuçlarla karşılaştırılabilir.

• Sıkıştırma (Compression): Küme prototipleri veri sıkıştırmak için kullanılabilir. Özellikle, her bir küme için prototipleri içeren bir tablo yaratılır. Örneğin her bir prototipe tablo için pozisyonunu belirten bir tamsayı değeri atanır. Her bir nesne onun kümesi ile ilişkili olan prototipin indeksi ile gösterilir. Sıkıştırmanın bu çeşidi vektör niceleme(vector quantization) olarak bilinir ve resim, ses ve video verileri gibi veri nesnelerinin birçoğunun birbirine çok benzediği, az veri kayıplarının kabuledilebilir olduğu ve veri büyüklüğünde önemli azalmaların tercih edildiği durumlarda genellikle uygulanır.

• En Yakın Komşulukları Etkili Bir Şekilde Bulma (Efficiently Finding Nearest Neighbors): En yakın komşulukları bulmak, tüm noktalar arasındaki mesafenin

hesaplanmasını gerektirebilir. Genellikle kümeler ve kümelerin prototipleri daha verimli bir şekilde bulunabilir. Nesneler onların küme prototiplerine nispeten yakınsa, nesnelerin en yakın komşuluklarını bulmak için gerekli olan uzaklık hesaplamlarının sayısını azaltmak için protipler kullanılabilir. Sezgisel olarak 2 küme prototipi birbirinden uzakta ise, ilişkili olan kümeler arasındaki nesneler birbirinin en yakın komşuları olmaz. Nesnelerin en yakın komşuluklarını bulmak için sadece yakın kümeler içindeki nesnelere uzaklığı hesaplamak gerekir. İki kümenin yakınlığı onların prototipleri arasındaki uzaklıkla ölçülür.

4. MERKEZ TABANLI KÜMELEME

4.1 Giriş

Bu bölümde merkez tabanlı kümeleme yapısı hakkında bilgiler verilerek giriş yapılmış ve merkez tabanlı kümelemede kullanılan başlangıç yöntemleri sırayla tanıtılarak ayrıntılı bir şekilde anlatılmıştır. Ardından K-ortalama (k-means), Bulanık k-ortalama (fuzzy k-means), K-harmonik ortalama (k-harmonic means) algoritmaları sırasıyla örneklerle ele alınıp açıklanmıştır. Ayrıca K-ortalama ve K-harmonik ortalama algoritmalarının özelliklerini barındıran Hibrit 1 (Hybrid 1) ve Hibrit 2 (Hybrid 2) adındaki 2 algoritmada sırasıyla açıklanmıştır.

4.2 Merkez Tabanlı Kümeleme

Bölümlemeli kümeleme metotları, veri kümesini giriş parametresi olarak belirlenen k adet küme içine böler. Bu algoritmalar karesel hata fonksiyonu gibi belirli bir kümeleme fonksiyonu kriterini minimize etmeye çalışırlar ve sonuç olarak optimizasyon problemi olarak davranırlar. Bölümlemeli kümeleme metotlarının en popüler olan sınıfı merkez tabanlı kümeleme algoritmalarıdır. Merkez tabanlı kümelemeye örnek tabanlı kümeleme (prototype based clustering) ve amaç fonksiyonu tabanlı kümeleme de denmektedir. Merkez tabanlı kümeleme algoritmaları çok iyi çalışır fakat bu algoritmalar tümel en küçük değerden uzakta olan yerel bir en küçük değerde yakınsayabilirler. Kötü yerel bir en küçük değerde yakınsamak başlangıçta seçilen noktalara duyarlılıktan kaynaklanmaktadır ve bu durum veri kümelemenin ana problemidir.

K-ortalama (K-Means), beklenen eniyileme (Expectation Maximization), bulanık k-ortalama (Fuzzy K-Means) ve k-harmonik ortalama (K-Harmonic Means) algoritmaları merkez tabanlı kümeleme algoritmalarıdır. Bu algoritmaların her birinin kendine ait amaç fonksiyonu (objective function) vardır.

Merkez tabanlı kümeleme algoritmaları yerel bir en küçük değerde yakınsayan yinelemeli algoritmalardır (iterative algorithms). Yinelemeli algoritmalarının adımlarını inceleyecek olursak [22];

1) k küme merkezini k adet rasgele seçilen veri olacak şekilde belirle.

2) Her bir veri noktası x_i' nin, her bir merkez c_j içindeki üyelik değeri $m(c_j|x_i)$ ve ağırlığı $w(x_i)$ hesapla.

3) c_j küme merkezlerini yeni üyelik ve ağırlık değerlerine göre tekrar hesapla.

$$c_j = \frac{\sum_{i=1}^{n} m(c_j|x_i) w(x_i) x_i}{\sum_{i=1}^{n} m(c_j|x_i) w(x_i)} \tag{4.1}$$

4) Durdurma kıstası sağlanıncaya kadar 2. ve 3. adımları tekrar et.

Yukarıdaki algoritmada, $X=\{x_1,\ldots,x_n\}$ d boyutlu n veri noktasından oluşan veri kümesini ifade etmektedir. k giriş parametresi oluşturulması istenen küme sayısını ve $C=\{c_1,\ldots,c_k\}$ d boyutlu k merkezlerinden oluşan kümeyi ifade etmektedir. $m(c_j|x_i)$ üyelik fonksiyonu, $m(c_j|x_i)\geq 0$ ve $\sum_{j=1}^{k} m(c_j|x_i)=1$ kısıtlamaları ile x_i noktasının c_j merkezine olan üyelik değerini tanımlar. Ağırlık fonksiyonu $w(x_i)$, $w(x_i)>0$ kısıtlaması ile bir sonraki iterasyonda merkez parametrelerinin hesaplanmasında x_i veri noktasının ne kadar etkili olduğunu tanımlar.

Yinelemeli kümeleme algoritmalarında kullanılan durdurma kıstasları aşağıda verilmiştir [22]:

- Maksimum iterasyon sayısı aşıldığında,
- Merkez değerleri içindeki değişim kullanıcı tarafından belirlenen değerden küçük olduğunda,
- Hatanın karesinin toplamı yeterince küçük olduğunda algoritmanın çalışması durur.

Bazı algoritmalar katı üyelik fonksiyonu kullanırlar. Katı üyelik fonksiyonunda, veri kümesi içindeki bir nesne sadece bir kümeye ait olabilir ($m(c_j|x_i) \in \{0,1\}$). Kazanan hepsini alır mantığı üzerine kuruludur. Yumuşak üyelik fonksiyonunda veri kümesi içindeki bir nesnenin tüm merkezlere üyeliği vardır ($0 \leq m(c_j|x_i) \leq 1$) [41].

4.3 Merkez Tabanlı Kümelemede Kullanılan Başlangıç Yöntemleri

Kümeleme işlemine başlanmadan önce başlangıç yöntemleri kullanılarak başlangıç noktaları oluşturulur. Oluşturulan bu başlangıç noktalarına bazı algoritmalar aşırı derece de duyarlıdır. Seçilen başlangıç noktalarının etkisi kümeleme işlemi sonucu oluşan kümelere direkt olarak yansımaktadır. Kümelemedeki bu başlangıca karşı duyarlılıktan dolayı araştırmacılar birçok başlangıç yöntemleri geliştirmişler ve bu problemi olabildiğince ortadan kaldırmaya hedeflemişlerdir. Bu başlangıç yöntemlerinden en fazla kullanılanlar aşağıda belirtilmiştir [29]:

- MacQueen Yöntemi: Veri kümesinin ilk k tane veri noktası ilk küme merkezleri olarak alınır ve bu merkezler algoritmalar tarafından kümeleme işleminde kullanılır. Bu yöntemde önemli alınan başlangıç merkezlerinin birbirinin takip eden bir sırada olmasıdır.
- Rasgele (Forgy) Yöntemi: Veri kümesi içinden rasgele olarak k tane veri noktası seçilir. k sayısı kullanıcı tarafından girilen oluşturulacak küme sayısını ifade eden kümeleme işlemi boyunca kullanılan sabit bir sayıdır. Seçilen k tane veri noktası ilk küme merkezlerini ifade eder ve bu noktalar tek elemanlı olan ilk kümeleri oluştururlar.
- Rasgele Bölümlemeli (Random Partition) Yöntemi: Bu yöntem veri kümesinin rasgele olarak seçilmiş olan k tane küme parçasına bölünmesi mantığına dayanır. Bu yöntemde ilk önce veri kümesi k adet parçaya ayrılır. Hangi parçanın hangi küme ile ilişkili olduğu belli değildir. Bu parçaların her biri rasgele seçilmiş olan k kümeden biri ile ilişkilendirilir ve her bir küme merkezi kendisi ile ilişkili olan parça içindeki veri noktalarının aritmetik ortalaması alınarak hesaplanır. Bu hesaplanan küme merkezleri tek elemanlı olan ilk kümeleri oluştururlar.

4.4 Merkez Tabanlı Kümeleme Algoritmaları

K-ortalama, beklenen eniyileme, bulanık k-ortalama ve k-harmonik ortalama algoritmaları merkez tabanlı kümeleme algoritmalarıdır. Bu tezde k-ortalama, bulanık k-ortalama ve k-harmonik ortalama algoritmaları üzerinde durulmuş ve bu algoritmalar üzerinde karşılaştırma işlemi yapılmıştır. Belirtilen merkez tabanlı kümeleme algoritmaların yapısı ve işleyişi sırasıyla ayrıntılı bir şekilde aşağıda ele alınmıştır.

4.4.1 K-ortalama algoritması (k-means algorithm)

K-ortalama algoritması, kümeleme algoritmaları içinde en eski ve yaygın olarak kullanılan algoritmalardan biridir. K-ortalama algoritması ilk defa MacQueen tarafından 1967' de tanıtılmıştır. K-ortalama algoritması, veri nesnelerini nitelik ve özelliklerine göre k adet kümeye ayıran bir algoritmadır. Küme sayısını gösteren k değeri, kümeleme işleminden önce kullanıcı tarafından belirlenen pozitif bir tamsayısıdır. K-ortalama algoritması literatürdeki kaynaklarda İngilizce kısaltmasıyla kullanıldığı için bundan sonraki kısımlarda k-ortalama algoritması KM kısaltmasıyla kullanılacaktır.

KM algoritması oluşan kümelerin kalitesinin ölçümünde kümeleme kriteri olarak hatanın karesi kriterini kullanır. KM algoritmasıyla kümeleme, veri noktası ve ilişkili küme merkezi arasındaki uzaklıkların karelerinin toplamının minimize edilmesiyle yapılır. k adet küme içinde toplam N tane kayıt içeren veri kümesi üzerindeki K kümelenmesi sonucu oluşan toplam hatanın karesi aşağıdaki gibi hesaplanır [50]:

$$\text{Perf}_{KM}(X,C)=\sum_{i=1}^{N}\min\left\|x_i - c_j\right\|^2 \qquad j=\{1,..,k\} \tag{4.2}$$

Bu denklemdeki x_i, i. veri noktasını, c_j ise j.merkezi ifade etmektedir. k oluşturulacak küme sayısını, n veri kümesi içindeki veri noktalarının sayısını ifade etmektedir.

KM algoritmasının adımları sırasıyla aşağıda verilmiştir:

1) k adet küme merkezlerinin ilk değerleri belirlenir(k<n). İlk k adet eleman küme merkezi olabileceği gibi k adet rasgele seçilmiş elemanda küme merkezi olabilir.

2) Her bir veri noktası, kendisine en yakın olan merkezin bulunduğu kümeye atanır. Örneğin x_1 veri noktası, c_1 merkezine en yakın ise $m_{KM}(c_1|x_1)=1$ olurken, x_1 noktasının diğer tüm merkezlere üyelikleri 0 olur. Aşağıdaki denklem ile bu durum ifade edilmiştir.

$$m_{KM}(c_l|x_i)=1;\ \text{if}\ \ l=\arg\min_j \left\|x_i - c_j\right\|^2,\ 0;\ \text{diğer.} \qquad j=\{1,2,..,k\} \tag{4.3}$$

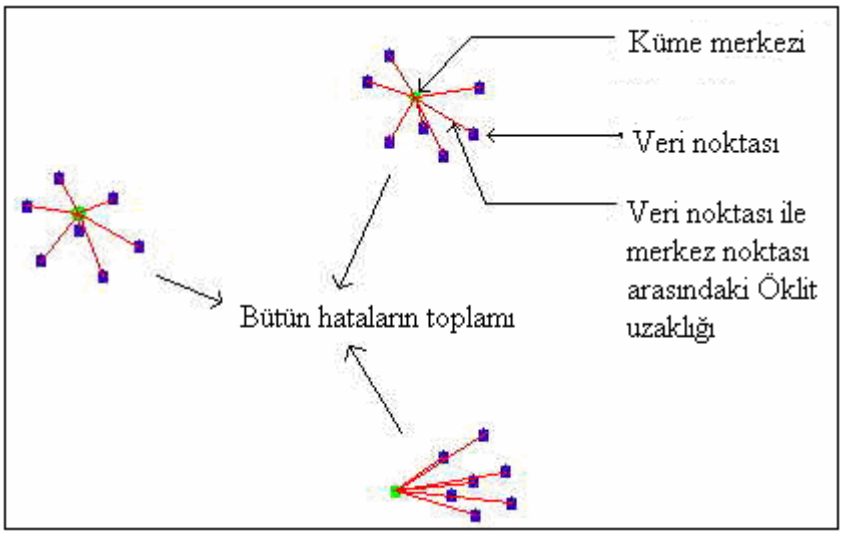
3) Küme üyelik değerlerini kullanarak yeni küme merkezleri aşağıdaki formüle göre hesaplanır. Formül de görüldüğü gibi her bir kümenin merkezi, küme içindeki elemanların aritmetik ortalaması alınarak hesaplanır.

$$c_k = \frac{\sum_{i=1}^{N} \left(m_{KM}\ (c_k|x_i)\right) * x_i}{\sum_{i=1}^{N} \left(m_{KM}\ (c_k|x_i)\right)} \tag{4.4}$$

4) Herhangi bir yakınsama koşulu sağlanmamışsa adım 2' ye tekrar dönülür. Yakınsama koşuluk olarak merkezler değişmemesi, toplam hatanın karesi kıstasının en küçük olması verilebilir.

KM algoritmasında ilk başta k tane küme merkezi seçilir. k adet küme merkezi, veri kümesinin ilk k adet elemanı olabileceği gibi, veri kümesi içinden rasgele seçilerek oluşturulmuş olan k tane elemanda olabilir. Bu şekilde ilk küme merkezleri belirlenir. İlk küme merkezlerinin belirlenmesinden sonra veri kümesi içindeki her bir veri noktasının k adet küme merkezine olan öklit uzaklığı hesaplanır. Her bir veri noktası hesaplanan öklit uzaklığı değerine göre hangi küme merkezine daha yakınsa o kümeye dâhil edilir. Bir veri noktası ile ilgili merkez arasındaki öklit uzaklık değeri ne kadar küçük ise veri noktası ilgili merkez noktasına o kadar yakın demektir ve

dolayısıyla veri noktası o merkezin bulunduğu kümeye dâhil edilir. Veri kümesi içindeki tüm veri noktaları k adet küme merkezlerinin bulunduğu kümelerden birine dâhil edildikten sonra her bir kümenin merkezi, kümeye atanan noktaların aritmetik ortalaması alınarak hesaplanır. Ayrıca Şekil 4.1' de görüldüğü gibi her bir küme merkezi ve ilgili küme içindeki her bir veri noktası arasındaki öklit uzaklık değerlerinin toplamı hesaplanarak hatanın karesinin toplamı elde edilir. Her bir iterasyonda hesaplanan hatanın karesinin toplamı kümelerin kalitesi hakkında kullanıcıya bilgi verir. Kümeye noktaları atama ve merkezleri güncelleme işlemleri, kümeler içindeki noktalar değişmediği sürece ya da merkezler aynı kaldığı sürece devam eder. KM algoritmasında yakınsamanın çoğu erken adımlarda gerçekleşir.

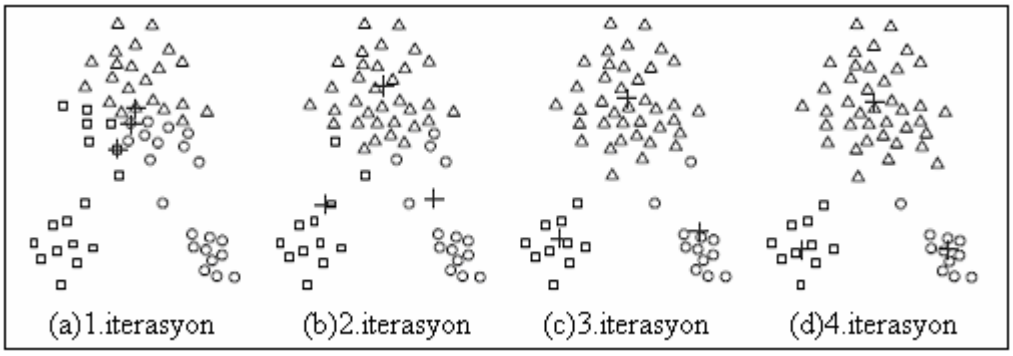


Şekil 4.1: KM algoritmasının işleyişi ile bütün hataların toplamının elde edilmesi [46].

KM algoritması katı üyelik fonksiyonuna sahiptir. Her eleman aynı anda verilen bir kümenin içindedir veya dışındadır ($m(c_j|x_i) \in \{0,1\}$). KM algoritması bütün veri noktalarına eşit derece önem veren sabit ağırlık fonksiyonuna sahiptir [41].

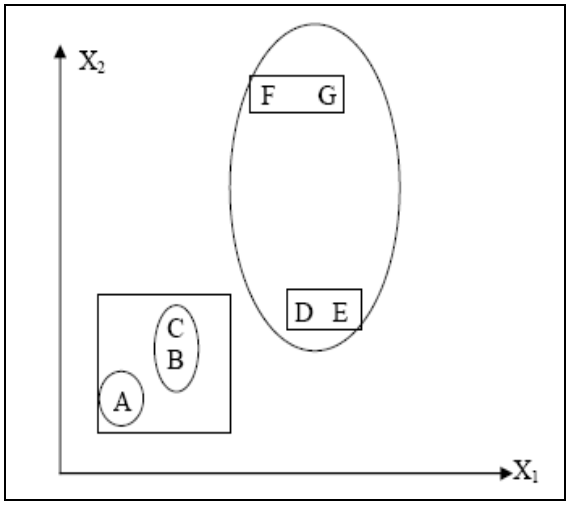
Şekil 4.2' de KM algoritmasının işleyişi adım adım gösterilmiştir. Bu örnekte algoritma 3 başlangıç merkezi ile çalışmaya başlamış ve 4 atama ve güncelleme adımından sonra kümelerin son haline ulaşılmıştır. Şekil 4.1' deki '+' sembolü merkezi ifade etmektedir. Aynı kümeye ait olan noktalar aynı şekillerle gösterilmiştir. Şekil 4.2 (a)' da ilk adımda, noktalar başlangıç merkezlerine atanmıştır. Noktalar kendilerine en yakın olan merkeze atandıktan sonra, merkezler

noktaların aritmetik ortalaması alınarak güncellenirler. İkinci adımda noktalar kendilerine en yakın, güncelleştirilmiş olan merkezlere atanırlar ve bu atama işlemlerinden sonra merkezler tekrar güncellenir. Şekilde 4.2 (b), (c), (d)' de gösterildiği gibi merkezlerin 2 tanesi şekillerin altındaki 2 küçük noktalar gruplarına ilerlemişlerdir. KM algoritması kümelere atanan noktalar değişmediği için Şekil 4.2 (d)' de sonlanmıştır [42].



Şekil 4.2: KM algoritmasının işleyişi [42].

KM algoritması başlangıçta seçilen küme merkezlerine oldukça duyarlıdır. Küme merkezlerinin değerlerinin uygun seçilmemesi durumunda algoritma toplam karenin hatası kriterinin yerel en küçük değerini bulmaktadır. Bu kaliteli olmayan kümelerin oluşması anlamına gelmektedir. Şekil 4.3' te 2 boyutlu olarak A, B, C, D, E, F, G veri noktaları verilmiştir. Başlangıç küme merkezi noktaları olarak A, B ve C noktaları seçildiğinde, KM algoritmasının uygulanması sonucu {A}, {B, C} ve {D, E, F, G} kümeleri üretilmiştir ve bu kümeler elips şekillerle gösterilmiştir. Bu veri kümesinin en iyi bölümlenmesi {A, B, C}, {D, E} ve {F, G} şeklindedir. Şekil 4.3' te veri kümesinin en iyi bölümlenmesi dikdörtgen şekilleriyle gösterilmiştir. {A}, {B, C} ve {D, E, F, G} kümelerinin oluştuğu kümelemeden sonra elde edilen toplam hatanın karesi fonksiyonun değeri, {A, B, C}, {D, E} ve {F, G} kümelerinin oluştuğu kümelemeden sonra elde edilen toplam hatanın karesi fonksiyonun değerinden daha büyüktür. İlk kümeleme işlemi sonucunda toplam hatanın karesi fonksiyonun yerel bir en küçük değeri elde edilmiştir. Fakat iyi kümeleme için tümel en küçük değerinin elde edilmesi gerekmektedir. Tümel en küçük değerinin elde edildiği {A, B, C}, {D, E} ve {F, G} kümelerinin oluşması için başlangıç noktası olarak A, D ve F noktalarının seçilmesi gerekmektedir [17].



Şekil 4.3: KM algoritmasının başlangıçta seçilen küme merkezlerine duyarlı olması [17].

Doğru küme sayısını seçmek KM algoritmasında önemli bir problemdir. Bazen uygun olan k değeri deneme yanılma yolu ile belirlenirken bazen de kümeleme problemi k değerine karar verir. Tüketicilerin kümelenmesi örneğinde kısıtlayıcı mevcut olan satıcıların sayısı olabilir [47]. k değeri seçilmesinde, satıcıların sayısından daha küçük olması kısıdı kullanılabilir. Farklı k sayısının seçilmesi farklı kümelerin oluşmasına neden olmaktadır. Oluşan kümelerden hangisinin daha iyi olduğu, hangi kümelerin kullanacağına bağlıdır [47].

K algoritmasının birçok avantajı ve dezavantajı vardır. Avantajları:

- Anlaşılması ve uygulanması basit algoritmadır.
- Basit ve hızlı bir algoritma olduğunda geniş veritabanları üzerinde etkili bir şekilde çalışır.
- Kategorik veriler sayısal verilere dönüştürülerek KM algoritması ile çalışacak durumu getirilebilmektedir.
- Veriye bağlı olarak hiyerarşik kümelemeden çok daha hızlı olabilir.
- Zaman karmaşıklığı O(n)' dir [17].

Dezavantajları [48]:

- Algoritma veri bağımlıdır.
- Veri miktarı az ise, ilk gruplama kümeyi belirgin şekilde belirler.

- Küme sayısı k' ya önceden karar verilmelidir. Eğer küme sayısı belirli değilse deneme yanılma yoluyla en uygun küme sayısı bulunur.
- İlk duruma duyarlıdır. Bu da algoritmanın alt en iyi(suboptimal) çözümlerde yakınsamasına neden olmaktadır.
- Farklı ilk durum koşulları farklı kümeler oluşturabilir.
- Hangi niteliğin gruplama işlemine daha fazla katılacağını bilemeyiz. Çünkü her özelliğin eşit ağırlıkta olduğunu varsayılır.
- Aritmetik ortalamanın güçsüzlüğü, aykırı değerlere karşı dayanıksız olmasıdır. Merkezden çok veri, merkezi gerçek merkezden uzaklaştırabilir. Bu yüzden veriler gürültü ve istisnadan temizlenmelidir.
- Çakışan kümelerde iyi sonuçlar vermez [41].
- Her eleman aynı anda verilen bir kümenin içindedir veya dışındadır [41].
- Sadece sayısal veriler için kullanılabilmektedir.
- Sonuçlar, mesafeye dayandığından dairesel küme şeklindedir.

KM algoritması verileri kümelere ayırmak için aritmetik ve geometrik hesaplama tekniklerinin kullanmaktadır. Bu teknikler aşağıda ayrıntıları ile açıklanmıştır.

4.4.1.1 Aritmetik hesaplama

KM algoritmasında, her bir küme içindeki noktaların aritmetik ortalaması alınarak yeni küme merkezleri bulunmaktadır. Aritmetik ortalama, bir küme içindeki noktaların toplamına noktaların sayısının bölünmesiyle elde edilir. Aritmetik hesaplamanın daha iyi anlaşılması için bir örnekle açıklama yoluna gidilmiştir. Aşağıdaki Tablo 4.1' de 2 tane niteliğe sahip olan 4 tane nesne verilmiştir. pH değeri ve ağırlık indeksi nitelikler olarak ele alınmıştır. Tabloda her bir ilaç nesnesinin ağırlık indeksi ve pH değeri verilmiştir. Ağırlık indeksi x koordinat sisteminde, pH değeri de y koordinat sisteminde gösterilmektedir. Örnekte amaç, verilen ilaç nesnelerini niteliklere dayalı olarak 2 kümeye ayırmaktır [48].

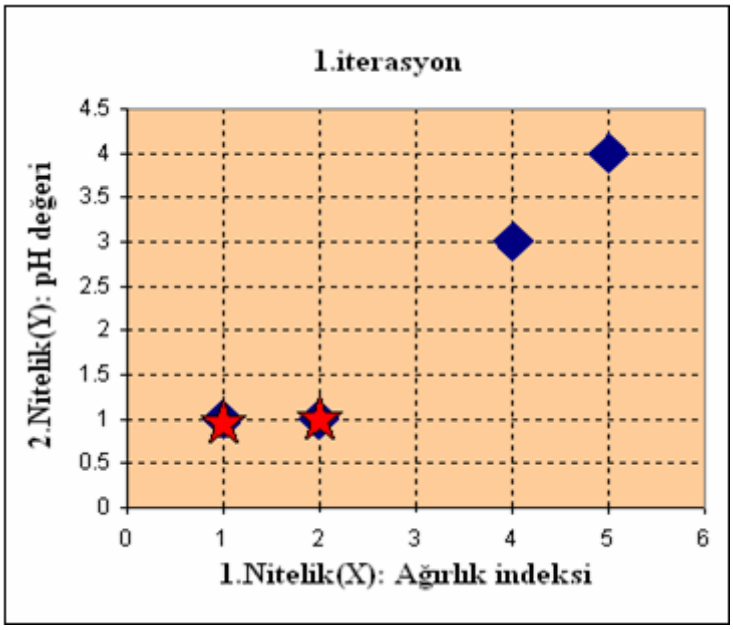
Tablo 4.1: Kümelemede kullanılacak ilaç nesneleri ve nitelik değerleri[48].

Nesne	1. Nitelik(X): Ağırlık indeksi	2.Nitelik(Y) pH değeri
A İlacı	1	1
B İlacı	2	1
C İlacı	4	3
D İlacı	5	4

Her bir ilaç nesnesi koordinat sisteminde pH değeri ve ağırlık indeksi nitelikleri(X,Y) ile tek bir nokta olarak gösterilir. A ilacı (1,1), B ilacı (2,1), C ilacı (4,3) ve D ilacı (5,4) noktaları Şekil 4.4' te koordinat sisteminde gösterilmiştir.

Şekil.4.4: İlaç nesnelerinin koordinat sisteminde gösterilişi [48].

Bu örnekte A ve B ilacı başlangıç merkezi olarak kullanılmıştır. c_1, birinci küme merkezi için başlangıç merkezi olarak A noktası seçilmiş ve c_1=(1,1) olmuştur. c_2, ikinci küme merkezi için başlangıç merkezi olarak B noktası seçilmiş ve c_2=(2,1) olmuştur. Şekil 4.5' da ilk küme merkezleri yıldız şekilleriyle gösterilmiştir.



Şekil.4.5: İlk küme merkezlerinin gösterilmesi [48].

Her bir nesne ve küme merkezleri arasındaki uzaklıklar Öklit uzaklık formülü kullanılarak hesaplanır ve 1.iterasyon için aşağıdaki uzaklık matrisi(D^0) elde edilmiştir:

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

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

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

Uzaklık matrisi içindeki her bir kolon bir nesneyi ifade etmektedir. Uzaklık matrisinin ilk satırı her bir nesnenin birinci merkeze olan uzaklık değerini, ikinci satırı her bir nesnenin ikinci merkeze olan uzaklık değerini göstermektedir. A ve B noktaları ilk merkezler olduğundan, A noktası (1,1) ile birinci küme merkezi (1,1) noktası arasındaki uzaklık değeri $\sqrt{(1-1)^2+(1-1)^2}=0$, ikinci küme merkezine olan uzaklık değeri $\sqrt{(1-2)^2+(1-1)^2}=1$' dir. A noktası ilk küme merkezi olduğundan, küme merkezinin kendisine olan uzaklık değeri 0 olur. Aynı durum küme merkezi olan B noktası içinde geçerlidir. C noktası (4,3) ile birinci küme merkezi (1,1)

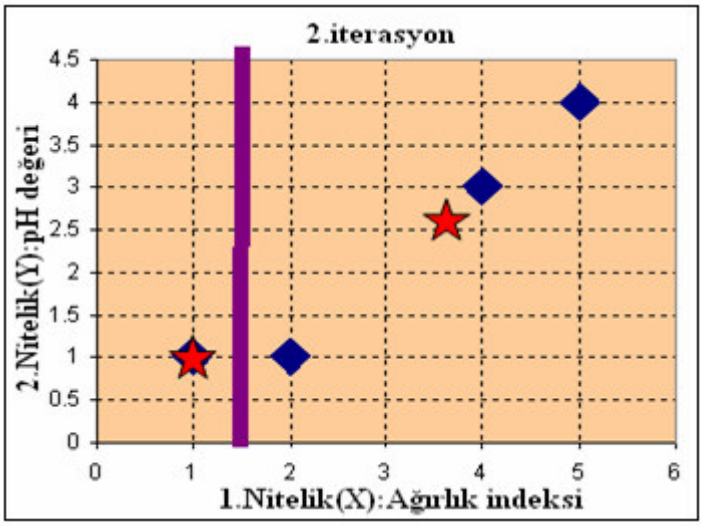
noktası arasındaki uzaklık değeri $\sqrt{(4-1)^2+(3-1)^2}=3.61$' dir. İkinci küme merkezi (2,1) noktası arasındaki uzaklık değeri $\sqrt{(4-2)^2+(3-1)^2}=2.83$' dür.

Her bir ilaç nesnesi, uzaklık matrisinde bulunan her bir merkeze uzaklık değerleri dikkate alınarak kümelere atanır. A ilacının, birinci merkeze uzaklık değeri olan 0, ikinci merkeze uzaklık değeri 1' den daha küçüktür. Dolayısıyla A ilacı birinci küme merkezinin bulunduğu kümeye dâhil edilir. C ilacının birinci merkeze uzaklık değeri 3.61, ikinci merkeze uzaklığı 2.83' tür. C ilacının ikinci merkeze uzaklığı birinci merkeze uzaklığından daha küçük olduğundan C ilacı ikinci küme merkezinin bulunduğu kümeye dâhil edilir. A ilacı birinci kümeye, B ilacı ikinci kümeye, C ilacı ikinci kümeye ve D ilacı ikinci kümeye dâhil edilir. Bir ilacın hangi kümeye dâhil olduğunu G küme matrisi ile tutarız. G matrisinin satırları kümeleri, sütunları ilaçları göstermektedir. G matrisi üzerinde nesneler, dâhil edildikleri kümelerde 1, dâhil edilmediklerinde ise 0 ile gösterilir. Örneğin A nesnesi birinci kümeye dâhil edildiği için G matrisinin ilk satırı 1, ikinci satırı ise 0 değerini almaktadır. Aynı şekilde C ilacı ikinci kümeye dâhil edildiği için G matrisinin birinci satırı 0, ikinci satırı 1 değerini almaktadır. G matrisi üzerindeki değere bakarak hangi ilacın hangi küme içinde olduğu anlaşılır. G matrisi bir ilacın kümelere üyeliğini göstermektedir.

$$\mathbf{G}^0=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 1 & 1\end{bmatrix}$$
$$A \quad B \quad C \quad D$$

İkinci iterasyonda her bir kümenin üyelerini bildiğimizden, her bir kümenin yeni merkezleri hesaplanır. Birinci küme içinde sadece bir tane üye vardır. Birinci kümenin merkezi c_1=(1,1) olarak kalır. İkinci kümede 3 tane üye vardır. İkinci kümenin içindeki nesnelerin x ve y değerlerinin aritmetik ortalaması aşağıdaki gibi alınarak yeni küme merkezinin koordinat değeri oluşturulur.

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



Şekil.4.6: İkinci iterasyonda oluşan küme merkezleri [48].

Şekil 4.6' da ikinci iterasyon sonucu oluşan küme merkezler görülmektedir. Şekilde görüldüğü gibi birinci kümenin merkezi bir ilaç nesnesi iken, ikinci kümenin merkezi ilaç nesnelerinden farklı olan bir noktadır. Yeni oluşan küme merkezlerine tüm nesnelerin uzaklıkları Öklit uzaklığına göre tekrar hesaplanır. İkinci iterasyon sonucunda oluşan uzaklık matrisi aşağıda verilmiştir.

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

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

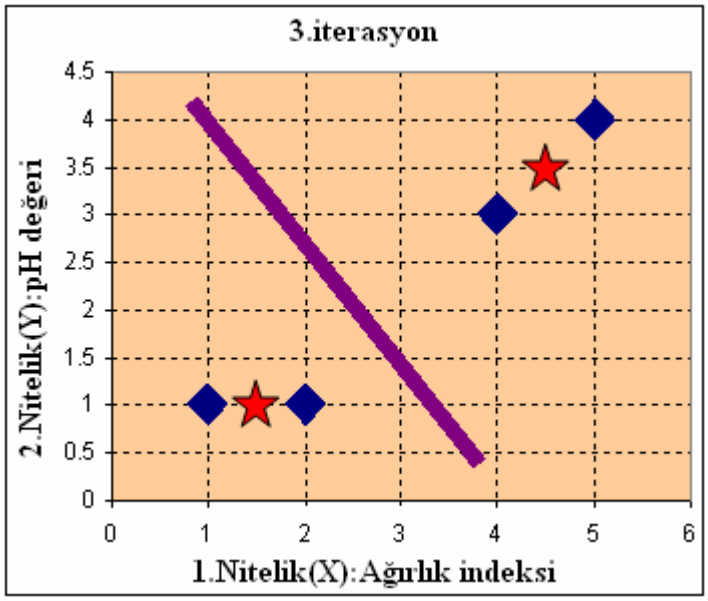
$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

Yeni uzaklık matrisindeki değerleri dikkate alarak ilaç nesneleri kümelere yerleştirilir. Birinci iterasyonda olduğu gibi uzaklık matrisinin birinci ve ikinci satırlarındaki değerler her ilaç nesnesi için karşılaştırılarak hangi kümeye dâhil oldukları tespit edilir. Yeni uzaklık matrisine göre, birinci iterasyondaki G^0 küme matrisindeki değerlerden, ikinci iterasyondaki küme matrisi G^1 üzerinde sadece B ilaç nesnesi üzerinde değişiklik olmuştur. B ilaç nesnesi birinci küme içine dâhil olmuştur. Diğer ilaç nesneleri aynı kümelerinde kalmıştır. Buna göre G küme matrisi aşağıdaki gibidir:

$$\mathbf{G}^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$
$$A \quad B \quad C \quad D$$

Üçüncü iterasyon, G^1 küme matrisindeki üyelikler dikkate alınarak yeni küme merkezlerin hesaplaması ile başlar. Birinci ve ikinci kümenin yeni merkez değerleri aşağıda verilmiştir. Kümelerin merkezlerinin koordinat değerleri küme içindeki nesnelerin x ve y değerlerinin aritmetik ortalaması alınarak hesaplanır. Şekil 4.7 üçüncü döngü sonucu oluşan küme merkezlerini göstermektedir.

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



Şekil.4.7: Üçüncü iterasyonda oluşan küme merkezleri [48].

İlaç nesnelerinin yeni küme merkezlerine olan uzaklıkları tekrar hesaplanır. Elde edilen uzaklık matrisi D^2 aşağıda verilmiştir.

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

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

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

Her bir ilaç nesnesi için uzaklık matrisinin birinci ve ikinci satır değerleri karşılaştırılarak ilaç nesnesinin hangi kümeye dâhil olduğu bulunur. Buna göre oluşturulmuş olan küme matrisi G^2' nin değerleri aşağıda verilmiştir.

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

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

İkinci ve üçüncü iterasyon sonucu oluşan küme matrisleri birbirlerine eşittir(G^1=G^2). Bu da nesnelerin artık herhangi bir küme içine ilerlemeyeceği anlamına gelmektedir. Daha fazla iterasyona ihtiyaç olmadığından kümeleme işlemi sona ermiştir. Kümeleme sonucu oluşan kümeler ve içeriğindeki ilaç nesneleri Tablo 2'de gösterilmiştir.

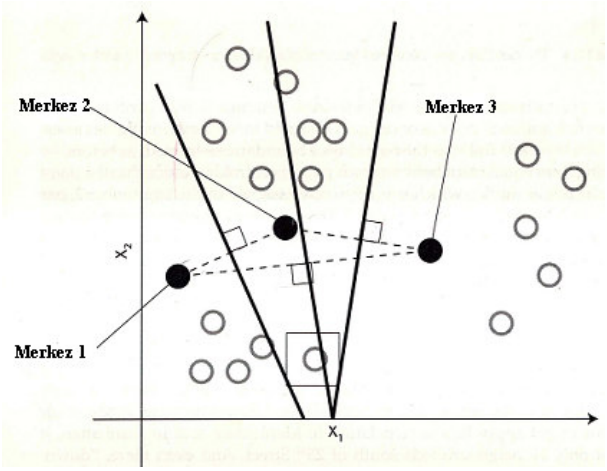
Tablo 4.2: Kümeleme sonucu oluşan kümeler ve içeriğindeki ilaç nesneleri [48].

Nesne	1.Nitelik(X): Ağırlık İndeksi	2.Nitelik(Y): Ph Değeri	Kümeleme Sonucu
A ilacı	1	1	1
B ilacı	2	1	1
C ilacı	4	3	2
D ilacı	5	4	2

4.4.1.2 Geometrik hesaplama

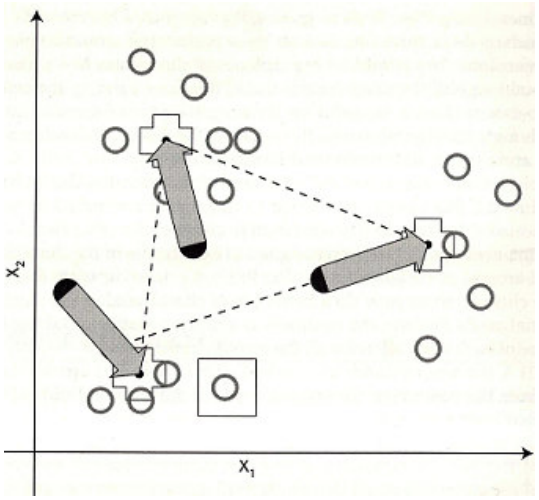
Bu yöntemde verileri kümelere ayırmak için küme sınırları kullanılmaktadır. İki boyutlu sistemde küme sınırları doğru olurken, üç boyutlu sistemde küme sınırları düzlem olmaktadır. N boyutlu sistemde, küme sınırları çok boyutlu düzlemeler

(hyperplanes) olacaktır. Geometrik ortalamanın daha anlaşılması için bir örnek ile açıklama yoluna gidilmiştir. Bu örnekte veri kümesi içindeki kayıt sayısı 20, k küme sayısı ise 3 olarak ayarlanmıştır. İlk adımda KM algoritması rasgele seçilmiş olan k veri noktasını küme merkezi olacak şekilde ayarlar. Merkezlerin her biri tek bir veri noktasını içeren embriyonik kümelerdir. İkinci adımda, her bir veri noktası en yakın küme merkezine atanır. Bunu yapmanın bir yolu da Şekil 4.9' da geometrik olarak gösterildiği gibi kümelerin sınırları bulmaktır. Şekil 4.9' da gösterildiği gibi içi dolu olan çemberler başlangıç küme merkezlerini, içi boş olan çemberler ise veri noktalarını göstermektedir. Başlangıç küme merkezleri başlangıç küme sınırlarına karar verilmesini sağlar. Küme sınırlarını belirlemek için merkezler kesikli çizgilerle birleştirilir. Bu kesikli çizgileri dik kesen kalın çizgilerde küme sınırlarını oluşturmaktadır. Şekil 4.8' da bu kalın çizgileri kullanarak hangi veri noktasının hangi küme merkezine dâhil olacağı açık bir şekilde görülmektedir. Gerçek küme sınırlarını bulmak geometrik olarak kümeleme işlemini göstermek için yararlıdır. Fakat algoritma genellikle her bir veri noktasının her bir merkeze olan uzaklığını hesaplar ve bu adımda, en yakın küme merkezinin bulunduğu kümeye veri noktasını atar [41].



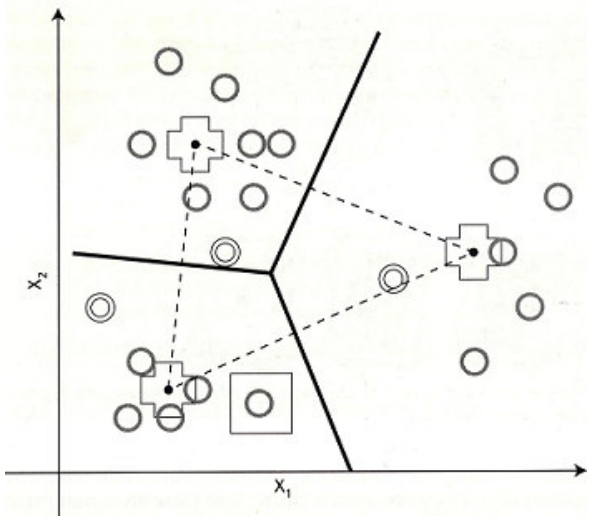
Şekil 4.8: Başlangıç merkezleri başlangıç küme sınırlarına karar verir [41].

Üçüncü adımda, kümelerin yeni merkezi kümeye atanan noktaların ortalaması alınarak hesaplanır. Şekil 4.9' da oluşan yeni merkezler çarpı ile gösterilmiştir. Oklar da ilk küme merkezlerinden yeni oluşan küme merkezlerine doğru gerçekleşen ilerlemeyi göstermektedir [41].



Şekil 4.9: Merkezler her bir kümeye atanan noktaların ortalaması alınarak hesaplanır [41].

Her bir veri noktası en yakın merkezin bulunduğu kümeye atanarak adım 2 tekrar edilir. Şekil 4.10' de küme merkezlerine göre belirlenen yeni küme sınırları gösterilmiştir. Şekil 4.10' da görüldüğü gibi ilk döngü sonucunda ikinci kümeye dâhil edilen kare içindeki veri noktası, ikinci döngü sonucunda sınırların değişimiyle birlikte birinci kümeye dâhil edilmiştir. Ayrıca Şekil 4.10' da döngü sonucunda küme merkezlerinin değişiminin takip edilmesi için önceki küme merkezleri çember içine alınarak gösterilmiştir. Kümelere veri noktalarını atama ve merkezleri tekrar hesaplama işlemleri küme sınırları değişimi duruncaya kadar devam eder [41].



Şekil 4.10: Her bir iterasyonda küme sınırları değişmektedir [41].

4.4.1.3 Optimizasyon problemi olarak KM' in incelenmesi

KM algoritmasını optimizasyon problemi olarak incelenemeden önce optimizasyondan özet olarak bahsetmek yararlı olacaktır. Optimizasyon problemi, belirli sınırlamalar çerçevesinde bilinmeyen parametre değerlerinin bulunması problemi olarak ifade edilebilir. Optimizasyon işleminde ilk olarak parametre kümesi belirlenir. Ardından en küçük yapılacak bir maliyet fonksiyonu veya en büyük yapılacak bir kâr fonksiyonu belirlenir. Ayrıca parametrelerin alamayacağı değerleri belirlemek içinde sınırlama(constraints) fonksiyonları tanımlanır. Problem için belirlenen bu sınırlamalar dâhilinde mümkün olan çözümlerin oluşturduğu bölge uygun(feasible) bölge olarak adlandırılır. Optimum çözüm, en küçük yapılacak problem durumuna uygun bölgedeki en düşük maliyet fonksiyonu değeri, en büyük yapılacak problem durumunda da en yüksek kâr fonksiyonu değerine sahip olan çözümdür [49].

KM algoritması karesel hata fonksiyonu gibi belirli bir kümeleme fonksiyonu kriterini minimize etmeye çalışır ve sonuç olarak optimizasyon problemi olarak davranır. Bu problemi çözmenin bir yolu, tümel en küçük değeri bulmaktadır. Bu değeri bulmak için veri kümesini kümelere bölmenin olası tüm yolları araştırılır. Toplam karesel hata fonksiyonu değerini minimize eden amaç fonksiyonunun tatmin eden en iyi çözümünün bulunduğu kümelenme seçilir. KM algoritmasında uzaklık fonksiyonu olarak Öklit uzaklık fonksiyonu kullanıldığında, kümenin toplam karesel hata fonksiyonu değerini minimize eden en iyi merkez değeri, küme içindeki noktaların aritmetik ortalamalarını alınması ile elde edilir. Bir boyutlu veri için toplam karesel hata fonksiyonu denklemi aşağıda verilmiştir [42]:

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} (c_i - x)^2 \tag{4.5}$$

Yukarıdaki denklemdeki C_i, i. kümedir. x, C_i kümesi içindeki bir veri noktasıdır. c_i i. kümenin ortalamasıdır. k. merkez c_k elde etmek için denklem (4,1) 0' a eşitlenip minimize edilirse yani (4.6), (4.7), (4.8) işlemler uygulanırsa aşağıdaki (4.9)' daki denklem elde edilir [42]:

$$\frac{\partial SSE}{\partial c_k} = \frac{\partial}{\partial c_k} \sum_{i=1}^{K} \sum_{x \in C_i} (c_i - x)^2 \tag{4.6}$$

$$= \sum_{i=1}^{K} \sum_{x \in C_i} \frac{\partial}{\partial c_k} (c_i - x)^2 \tag{4.7}$$

$$= \sum_{x \in C_k} 2 * (c_k - x_k) = 0 \tag{4.8}$$

$$\sum_{x \in C_k} 2 * (c_k - x_k) = 0 \Rightarrow \sum_{x \in C_i} x_k \Rightarrow \frac{1}{m_k} \sum_{x \in C_k} x_k \tag{4.9}$$

Denklemin çözümünden anlaşıldığı gibi bir küme içindeki toplam karesel hatayı minimize eden en iyi merkez, küme içindeki noktaların ortalamasıdır. KM algoritmasında uzaklık fonksiyonu olarak Manhattan uzaklık fonksiyonu kullanıldığında, kümenin toplam mutlak hata(sum of absolute errors)fonksiyonu değerini minimize eden en iyi merkez değeri, küme içindeki noktaların ortancasıdır (median) [42].

4.4.1.4 KM algoritmasında dikkat edilmesi gereken noktalar

KM algoritmasının ilerleyişi sırasında oluşan boş kümeler oluştuğunda, işlemden sonra toplam karesel hatayı azaltmak gerektiğinde ve aykırı değerlerin olması durumunda yapılması gerekenler aşağıda maddeler halinde belirtilmiştir:

1) Boş kümeleri yönetmek: KM algoritması ile ilgili problemlerden bir tanesi atama adımı boyunca bir kümeye herhangi bir nokta atanmazsa boş kümeler elde edilir. Eğer bu durum gerçekleşirse değişik bir merkez seçme stratejisi gereklidir. Çünkü hatanın karesi gereğinden fazla büyük olacaktır. Boş kümeler için diğer bir yaklaşımda o anki merkezden en uzakta olan noktayı seçmektir. Herhangi bir aksilik olmazsa bu işlem toplam karesel hataya en çok katkı da bulunan noktayı yok eder. Diğer bir yöntem; en yüksek karesel hataya sahip kümeden yer değiştirecek olan

merkezi seçmektir. Bu işlem kümeyi ikiye bölecektir ve kümenin toplam karesel hatası azalacaktır. Birkaç boş küme varsa bu işlem birkaç kez tekrarlanır [42].

2) Aykırı değerler: Karesel hata kıstası kullanıldığından, aykırı değerler buluna kümeleri aşırı derece de etkileyebilir. Özellikle aykırı değerler var olduğunda sonuçlanan küme merkezleri kümeyi iyi bir şekilde temsil etmez ve toplam karesel hata daha yüksek çıkar. Bu nedenle aykırı değerleri keşfetmek ve işlem yapmadan önce yok etmek kullanışlıdır. Farklı aykırı değerlerin silinmemesi gereken belirli kümeleme uygulamaları da vardır. Kümeleme veri sıkıştırma için kullanıldığında her nokta kümelenmelidir. Bazı durumlarda örneğin finansal analiz, açık, belirli aykırı değerlerde ve nadir karlı müşterilerde en ilginç noktalardan biri olabilir. Eğer kümeleme yapmadan önce aykırı değerleri eleyen bir yaklaşım kullanırsak iyi kümeleme yapmayacak noktaların seçiminden kaçınılmış olunur. Alternatif olarak aykırı değerler kümeleme yapıldıktan sonra yapılan ek bir adımla tanımlanabilirler. Örneğin her noktanın katılım yaptığı toplam karesel hata değerini takip etmeye devam edebilir ve toplam karesel hataya çok aşırı yüksek katılım yapan noktalar elenebilir. Ayrıca küçük kümeler elenebilir. Çünkü bu kümeler genellikle aykırı değerler gruplarını temsil ederler [42].

3) İşlemden sonra toplam karesel hatayı azaltmak: SSE' yi azaltmak için bir yolda daha fazla sayıda küme bulmaktadır. Birçok durumda SSE' yi geliştirmek istenir ama küme sayılarının artması istenmez. Bu durumla sık sık karşılaşılır. Çünkü KM algoritması yerel optimum da yakınsar. Düşük SSE' ye sahip kümeler üretmek ve kümelemeden sonra oluşan kümeleri onarmak için değişik teknikler kullanılır. Bu strateji, toplam SSE her bir küme ile katılan SSE' nin basitçe toplamı olduğundan tek tek kümeler üzerine odaklanır. Olası karışıklıkları önlemek için toplam SSE ve küme SSE değerleri kullanılmaktadır. Kümeleri birleştirme ve ayırma gibi kümeler üzerinde değişik işlemleri yerine getirerek toplam SSE değiştirilebilir. Genellikle kullanılan yaklaşımlardan biri değimli olarak küme bölme ve birleştirme evrelerini kullanmaktır. Bölme everesi boyunca kümeler bölünür, birleştirme evresi boyunca kümeler birleştirilir. Bu yolla yerel SSE minimumdan kaçınmak ve istenen sayıdaki küme sayısı ile kümeleme çözümü üretmek olasıdır [42].

Kümelerin sayısını arttırarak toplam SSE' yi azaltan iki strateji vardır:

- Kümeyi bölmek: En büyük SSE' ye sahip olan küme genellikle seçilir. Fakat belirli bir özellik için en yüksek standart sapması olan küme de bölünebilir.
- Yeni bir küme merkezi seçmek: Genellikle herhangi küme merkezine en uzakta olan nokta seçilir. Her noktanın SSE' lere katılımı takip edilerek bu nokta kolayca seçilebilir. Başka bir yaklaşımda en yüksek SSE' ye sahip olan noktalardan veya bütün noktalardan bir tane seçmektir.

Toplam SSE' deki artışı en aza indirmeye denerken kümelerin sayısını azaltan iki strateji aşağıdaki gibidir[42]:

- Kümeyi dağıtmak: Kümeye karşılık gelen merkez kaldırılır ve noktalar diğer kümelere yeniden atanır. İdeal olarak dağıtılan küme toplam SSE değerini en az arttıran küme olmalıdır.
- İki kümenin birleştirilmesi: Merkezleri birbirine yakın iki seçilir. Ayrıca toplam SSE' da en küçük artışa neden olan iki kümede seçilebilir.

4.4.1.5 KM algoritmasının uygulandığı örnekler

KM algoritması basit ve hızlı bir algoritma olduğundan dolayı farklı birçok alanda kullanılmıştır. KM algoritmasının uygulandığı birkaç örnek aşağıda verilmiştir:

a) Eğitim: Öğrencilerin üniversiteye giriş sınav sonuçları ve başarıları arasındaki ilişkiyi incelemek için kümeleme analizi ve KM algoritmasından yararlanılmıştır. Maltepe Üniversitesinin öğrencilerinden toplanan veri kümesi içinde 722 tane öğrenci kaydı vardı. Veritabanı KM algoritmasının uygulanacağı hale dönüştürüldükten sonra KM algoritması uygulanmış ve sonuçta 5 kümede başarılı bir kümelenin elde edildiğine karar verilmiştir. Birinci kümeye içindeki öğrenciler daha fazla başarılı iken, beşinci küme içindeki öğrenciler daha başarılı olarak tespit edilmiştir. Birinci küme içindeki öğrencilerin çoğu fen ve güzel sanatlar fakültesindeki öğrencilerdir. Bu fakültelerdeki öğrenciler yüksek başarı derecelerine ve burslara sahip olan öğrencilerdir. Beşinci küme içindeki öğrenciler iletişim ve hukuk gibi fakültelerdendir. Bu küme içindeki öğrenciler üniversite giriş sınavında düşük derecelere ve düşük sonuçlara sahip olan öğrencilerdir [51].

b) Elektronik Bankacılık: Thai ticari bankaları her geçen günkü gelişmelere ayak uydurmak için bankacılık hizmetlerini değiştirmektedirler. Bankalar, müşterilerini kaybetmemek ve rekabet alanında diğerlerine göre avantaj elde etmek için sürekli bankacılık servislerin geliştirmeye çalışmaktadırlar. Müşteri davranışını analiz etmek için bankacılık işlemlerinden oluşturulan veri kümesi üzerinde KM algoritması uygulanmış ve müşteri kullanımına dayalı olarak bankacılık işlemleri 5 küme içinde sınıflandırılmıştır. Bu veri kümesi içinde zaman, tarih, erişim kanalı, dil ve işlem nitelikleri bulunmaktadır. Bulunan birinci küme içinde müşteriler, 6.00 ve 17.59 saatleri arasında ayın üçüncü ve dördüncü haftalarında bankacılık işlemleri için kişisel bilgisayarları(PC-Personel Computer) kullanmışlardır. Müşterilerin sadece İngilizce kullandığı kümedir. Bu küme içindeki müşterilerin gerçekleştirdiği ana işlem fatura ödemeleridir. İkinci küme içindeki müşteriler, ayın birinci ve ikinci haftalarında kişisel bilgisayarlarını kullanmışlardır. Müşteriler 6.0 ve 17.59 saatleri arasında kişisel bilgisayarları ile bankacılık işlemlerini gerçekleştirmişlerdir. Kullanılan dil Thai dili olmuştur ve yapılan işlemler, kredi kartı hakkında bilgi edinme işlemleri olmuştur. Üçüncü küme en fazla müşterinin bulunduğu kümedir. Doğal olarak bankacılık işlemlerinin çoğu bu küme içindedir. Müşteriler bankacılık işlemleri için kişisel bilgisyarları kullanmıştır. Dil olarak Thai dilini kullanılmıştır. Müşterilerin 6.00 ve 17.59 saatleri arasında ayın üçüncü ve dördüncü haftalarında bankacılık işlemelerini gerçekleştirmişlerdir. Ankacılık işlemi olarak fatura ödemeleri yapılmıştır. Dördüncü küme en küçük olan kümedir. Müşteriler 12.00 ve 17.59 saatleri arasında bankacılık işlemleri için ATM' leri kullanmışlardır. Kullanılan dil Thai dilidir ve işlem olarak fatura ödemesi yapılmıştır. Beşinci küme, en geniş ikinci kümedir. Müşteriler ayın birinci ve ikinci haftalarında 6.00 ve 17.59 saatleri arasında bankacılık işlemleri için kişisel bilgisayarlarını kullanmışlardır. Kullanılan dil Thai dilidir ve yapılan işlem fatura ödemesidir. Bu kümeler, müşterilerin davranışları hakkında yararlı olabilecek bilgileri bankaya sağlamaktadır [52].

c) Arkeoloji: Güney Levant' ta eski tunç çağına ilişkin istatistiksel ve uzaysal çıkarımı("Spatial and Statistical Inference of late Bronze Age Polities in the Southern Levant") kâğıdından alınan örneğe göre, İsrail' deki arkeolojik yerler KM algoritması uygulanarak kümelenmiştir. Bu kümelemenin amacı, çıkan kümelere

dayalı olarak İsrail'i in tarihi hakkında sonuçlar çıkarmaktır. Küme işlemi sonucu 24 kümede karar kılınmıştır. Kümelerin karışık hesapsal tekniklerle dikkatli bir şekilde seçilmiştir [47].

d) Tıp: Gırtlak kanseri verilerinin analiz edilmesinde veri madenciliği algoritmalarından KM algoritmasının kullanılmıştır. Yapılan uygulama geçmiş verileri analiz ederken değişken parametreler kullanılarak değerlendirme yapılabilmesi ve vakalar için tahminde bulunulabilmesi, mevcut ve gelecek vakalar için ameliyat sonrasında tümörün nüks etme olasılığı ve hastanın hayatta kalma olasılığının değerlendirilebilmesine ilişkin bilgiler sağlamaktadır. Ayrıca doğru öngörülen ve öngörülemeyen ameliyat öncesi evrelerin görüntülenerek incelenebilmesi ve bu şekilde ameliyat öncesi tahmin başarısının değerlendirilmesi, başarılı ameliyat bilgilerinin izlenerek bu bilgiler ışığında gelecek ameliyat tercihlerinde fikir alınabilmesi de bu uygulama aracılığı ile elde edilmiş bilgiler arasındadır [19].

e) İklim: KM algoritması iklim alanında da uygulanmıştır. KM algoritması ile aylık yağış toplamları kullanılarak Türkiye' nin ana yağış bölgeleri belirlenmeye çalışılmıştır. 1977–2006 yılları için 148 noktada KM algoritması ile yağış verileri sınıflandırmaya tabi tutulmuş benzer özellikler gösteren istasyonlara ait olan yağış bölgeleri tespit edilmiştir. KM algoritması benzer özellikleri gösteren istasyonların aynı küme içinde toplanmasını ve tüm istasyonların k kümeye ayrılmasını sağlamıştır. En uygun olan küme sayısının belirlenmesi için en düşük negatif siluet sayısı ve maksimum ortalama siluet değeri dikkate alınmıştır. Buna göre yapılan kümeleme analizi sonucunda 1977–2006 yılları arasında en uygun yağış bölgesi sayısının(küme sayısı) 6 olarak tespit edilmiştir [20].

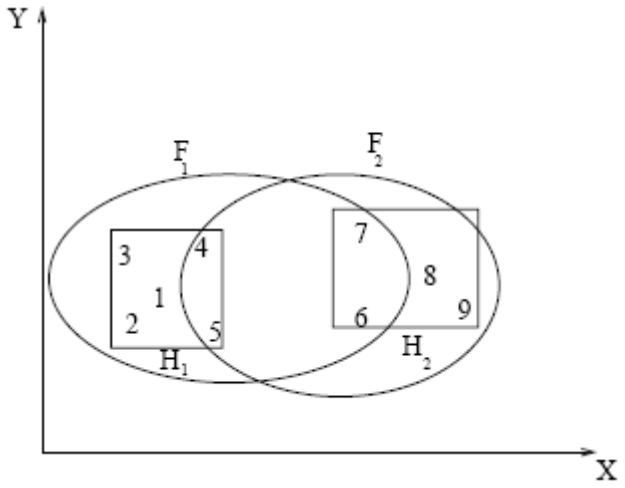
4.4.2 Bulanık k-ortalama algoritması (fuzzy k-means)

KM algoritmasının eksiklerini gidermek amacıyla ortaya atılan bulanık k-ortalama algoritmasının temelinde bulanık mantık vardır. Bulanık k-ortalama algoritmasını derinlemesine girmeden önce bulanık küme kavramından ve bulanık mantıktan bahsetmemiz gerekmektedir. Bulanık mantık yaklaşımı 1965 yılında ilk defa Zadeh

tarafından ortaya atılmıştır. Zadeh insan düşüncesinin büyük çoğunluğunun klasik yaklaşım da belirtildiği gibi kesin olmadığını bulanık olduğu belirtmiştir. 0 ve 1 ile temsil edilen klasik mantık insan düşüncesinin ifade edilmesinde yetersiz kalmıştır [53].Bulanık mantıkta uzun-kısa, çok kısa, çok uzun, az sıcak, çok sıcak, çok soğuk, az soğuk, çok çok doğru, yaklaşık olarak doğru gibi dilsel ifadeler vardır. Bu ifadelerin bilgisayara aktarılmasında bulanık kümeler kullanılır. Klasik kümelerin genelleştirilmiş şekli olan bulanık kümeler, farklı üyelik derecelerine sahip elemanlardan oluşur. Klasik küme yaklaşımında bir eleman kümenin elemanıdır veya değildir. Klasik kümelerde elemanların üyelik değerlerleri 0 ya da 1'dir. Eğer bir elemanın üyelik değeri 1 ise kümenin elemanı, 0 ise kümenin elemanı değildir. Bulanık küme yaklaşımında bir eleman kümeye kısmen ait olabilir. Bir bulanık kümedeki üyelik derecesi $\mu_A(x)$ üyelik fonksiyonu ile ifade edilir($0 \leq \mu_A \leq 1$). Üyelik fonksiyonuna ait ara değerler üyelik derecesi olarak adlandırılır. Üyelik derecesinin 0 olması, kümenin üyesi olmama;1 olması ise kümenin tam üyesi olma anlamına gelmektedir [54]. Bulanık mantık eksik, yanlış girilen, belirsiz, çelişkili olan bilgilere göre de işlem yapabilmektedir.

Bulanık kümeleme de veri kümesi içindeki her biri, belirli bir kümeye belirli bir üyelik derecesi ile dâhil olur. Bulanık kümelemede üyelik fonksiyonu [0,1] aralığında bir değer almaktadır. Bulanık kümeleme yaklaşımını Şekil 4.11' de gerçekleştirilen kümelerle açıklanmaya çalışılmıştır. Şekil 4.11' de dikdörtgenler içindeki H_1={1, 2, 3, 4, 5} ve H_2={6, 7, 8, 9} kümeleri katı kümelerdir. Bulanık kümeleme algoritması F_1 ve F_2 adında elips şeklinde gösterilen 2 bulanık küme üretir. Veri noktaları [0,1] aralığında her bir küme için üyelik değerlerine sahiptir. Her bir küme içindeki (i, μ_i) sıralı çifti, i. veri noktasını ve μ_i kümesine veri noktasının üyelik değerini ifade etmektedir [17].

F_1 bulanık kümesi için {(1,0.9), (2,0.8), (3,0.7), (4,0.6), (5,0.55), (6,0.2), (7,0.2), (8,0.0), (9,0.0)} şeklinde tanımlanabilir. F_2 bulanık kümesi için değerler; {(1,0.0), (2,0.0), (3,0.0), (4,0.1), (5,0.22), (6,0.4), (7,0.49), (8,1.0), (9,0.9)} şeklinde tanımlanabilir [17].



Şekil 4.11: Bulanık kümeler [17].

Üyelik derecesi ne kadar büyük ise veri noktasının kümeye atanma olasılığı o kadar yüksek demektir. Üyelik değerine belli bir eşik değeri uygulayarak bulanık kümelemeden katı kümelemeye geçilebilir.

En popüler bulanık kümeleme algoritması Bulanık k-ortalama algoritmasıdır. Bulanık k-ortalama algoritması Dunn tarafından 1973 yılında önerilmiş ve Bezdek tarafından 1981' de geliştirilmiş olan bir algoritmadır. Bulanık k-ortalama algoritması literatürde İngilizce kısaltmasıyla yaygın olarak kullanıldığından tezin bundan sonraki kısımlarında FKM kısaltmasıyla kullanılacaktır. FKM algoritması, KM algoritmasına alternatif olarak ortaya atılmış bir algoritmadır. Her bir veri noktasını kendisine en yakın olan merkeze atayan KM algoritmasından farklı olarak FKM algoritması veri noktasına tüm merkezlerin kısmi olarak sahip olmasına izin verir. KM algoritmasında üyelik fonksiyonu 0 ve 1 değerlerini almakta iken FKM algoritmasında üyelik fonksiyonu 0–1 arasında değerler almaktadır ve küme merkezi veri noktasının kendisi olmadığı sürece hiçbir zaman 1.0 değerini almamaktadır. FKM algoritmasında kümelerin her biri bulanık olarak tanımlanır. FKM algoritması yumuşak üyelik fonksiyonunu ve sabit bir ağırlık fonksiyonuna($w(x_i)=1$) sahiptir. FKM algoritması aşağıdaki amaç fonksiyonunun minimize edilmesine dayanır [55]:

$$\text{Perf}_{FKM}(X,C)=\sum_{i=1}^{N}\sum_{j=1}^{k}\left(m\left(c_j|x_i\right)\right)^r d_{ij}^2 \qquad (4.10)$$

Denklemdeki N, verilerin sayısını, k küme sayısını ifade eder. d_{ij}, i. veri noktası ile j. merkez arasındaki Öklit uzaklığı ifade eder. Denklemdeki x_i, veri kümesi içindeki i. veri noktasını ve c_j ise j. küme merkezin ifade eder. $m(c_j|x_i)$, x_i noktasının j. kümeye üyeliğini ifade eder ($m(c_j|x_i) \in [0,1]$). $m(c_j|x_i)$ üyelik fonksiyonun sahip olduğu kısıtlamalar aşağıda verilmiştir.

$$\sum_{j=1}^{k} m(c_j|x_i) = 1 \qquad \text{i=1,2,…,N} \tag{4.11}$$

$$\sum_{i=1}^{N} m(c_j|x_i) > 0 \qquad \text{j=1,2,..,k} \tag{4.12}$$

r, bulanıklılık katsayısıdır. r parametresi $r \geq 1$ kısıtlayıcısına sahiptir. r' nin değerinin büyük olması metodu daha da bulanık yapar. r=1 olması katı üyeliğe eşit olduğu anlamına gelir [55].

FKM algoritmasının adımları sırasıyla açıklanmıştır [56]:

1) k küme sayısını(1<k<N) ve r>1 değeri seçilir.
2) Rasgele üyelikler ile ilk üyelik matrisini($M^{(0)}$) belirlenir.
3) $M^{(t-1)}$ ve aşağıdaki denklem kullanılarak $C^{(t)}$ merkez hesaplanır.

$$c_j = \frac{\sum_{i=1}^{N} m(c_j|x_i)^r x_i}{\sum_{i=1}^{N} m(c_j|x_i)^r} \qquad \text{j=1,2,…,k} \tag{4.13}$$

4) $C^{(t)}$ ve aşağıdaki denklem kullanılarak $M^{(t)}$ hesaplanır.

$$m_{ij} = \frac{1}{\sum_{l=1}^{k} \left(\frac{\|x_i - c_j\|}{\|x_i - c_l\|} \right)^{\frac{2}{r-1}}} \tag{4.14}$$

5) $\left\|M^{(t)} - M^{(t-1)}\right\| < \varepsilon$ ise algoritmanın çalışması durur aksi halde adım 3' e dönülerek işlemler tekrar edilir.

FKM algoritmasının adımları içindeki t iterasyon sayısını, ε parametresi sonlandırma kıstasını ifade eder. ε parametresi, 0 ile 1 arasında bir değer alır.

FKM algoritması yumuşak üyelik fonksiyonu ve sabit ağırlık fonksiyonu sahiptir. r değeri 1' e yaklaştıkça algoritma KM algoritması gibi davranmaya başlar merkezler veri noktalarını daha az paylaşır. FKM algoritması çalıştırıldığında $\left\|M^{(t)} - M^{(t-1)}\right\| < \varepsilon$ kıstasını sağlayan çözüm, en iyi çözüm olmayabilir. FKM algoritması başlangıç seçimlerine dayalı olarak yerel bir en küçük değerde yakınsayabilir. Eğer bir çözüm en iyi çözüm ise, kümelerin merkezi başlangıç üyeliklerine rağmen daima aynı kalır. Ayrıca FKM algoritmasında k başlangıç değeri KM' de olduğu gibi kullanıcı tarafından belirlenmelidir. FKM algoritması veri içindeki belirsizliğin varlığından daha az etkilenir ve KM algoritmasından daha iyi performansa sahiptir [56].

4.4.3 K-harmonik ortalama algoritması (k-harmonik means algorithm)

K-harmonik ortalama algoritması her bir veri noktasından merkezlere uzaklıkların harmonik ortalamalarını kullanan merkez tabanlı yinelemeli kümeleme algoritmasıdır. KM algoritmasına alternatif olarak ortaya atılan algoritmalardan farklı olarak KM algoritmasına yeni bir başlatma modeli getirmemiştir. KM algoritmasının performans fonksiyonundaki en küçük belirleme olayına harmonik ortalama kavramını getirmiştir. K-harmonik ortalama algoritması literatürde İngilizce adıyla yaygın olarak kullanıldığından tezin bundan sonraki kısımlarında KHM kısaltmasıyla kullanılacaktır. KHM algoritmasının getirdiği yenilikleri anlamak için harmonik ortalama (HM-Harmonic Averages) kavramını açıklamak gereklidir [23].

HA, uzun zamandan beri bilinen bir ortalamadır. k tane sayının harmonik ortalaması $\{a_1,...,a_k\}$:

$$\mathrm{HA}\left(\{a_i | i = 1,\ldots,K\}\right) = \frac{K}{\sum_{i=1}^{K} \frac{1}{a_i}} \tag{4.15}$$

şeklinde tanımlanmaktadır. k tane sayının aritmetik ortalaması, veri kümesi içindeki sayıların terslerinin aritmetik ortalamalarının tersinin alınması ile hesaplanır. Denklemdeki a_k değerinden birinin küçük olması durumunda HA' da küçük çıkmaktadır. Bu nedenle harmonik ortalama aritmetik ortalamadan daha çok en küçük fonksiyonuna benzemektedir. a_k değerlerinin hepsini eşit olması durumunda aritmetik ortalama ve HA sonucu aynı değer çıkmaktadır. a_k değerlerinin farklı olması durumunda aritmetik ortalama sonucu elde edilen değer, harmonik ortalama sonucu elde edilen değerden daha büyük olmaktadır. HA için bir örnek verirsek bir sürücü otoyolda aracını sürerken şunlara dikkat eder. İlk önce 60 mil/sa, sonra 70 mil/sa ve sonra da 75 mil/sa hızı ile gitmiştir. İlk nokta ile son nokta arasındaki ortalama hızını bulmak için HA kullanılır.İlk nokta ve son nokta arasındaki toplam mil sayısı 3A' dır.İki nokta arasındaki mesafeyi geçmek için alınan süre A/60, A/70 ve A/75' dir. İlk ve son noktalar arasındaki toplam süre A/60+A/70+A/75' dir.

Ortalama Hız (60, 70 ve 75'in harmonik ortalama)= Toplam mil/Toplam süre

= (A+A+A)/(A/60+A/70+A/75)

= 3/(1/60+1/70+1/75)=67.7

C=k tane küme merkezinden oluşan küme olduğunda ve X veri noktalarını barındıran veri kümesi olduğunda, kümeleme algoritmasının performans değeri aşağıdaki gibidir:

$$\mathrm{Perf}(X, C) = \sum_{x \in X} d(x, C) \tag{4.16}$$

Performans fonksiyonun değeri, bütün veri elemanlarının ait oldukları küme merkezlerinden uzaklıklarının toplamı sonucu elde edilir. Denklemdeki d(x,C) ise x noktasının C küme merkezine öklit uzaklığını ifade eder.

$$d(x,C)_{KM} = \text{MIN}\left\{\|x-c\|^2 \,|\, c \in C\right\} \tag{4.17}$$

$$d(x,C)_{KHM} = \text{HA}\left\{\|x-c\|^2 \,|\, c \in C\right\} = \frac{|C|}{\sum_{c \in C} \frac{1}{\|x-c\|^2}} \tag{4.18}$$

şeklinde ifade edilebilir.

KM algoritmasının performans fonksiyonu aşağıda verilmiştir:

$$\text{Perf}_{KM}\left(\{x_i\}_{i=1}^{N}, \{c_l\}_{l=1}^{K}\right) = \sum_{l=1}^{K} \sum_{x \in S_l} \|x - c_j\|^2 \tag{4.19}$$

Performans fonksiyonundaki S_l, C içindeki diğer tüm merkezlerden c_l' ye daha yakın olan x' lerin alt kümesidir. S_l, X' in alt kümesidir($S_l \in X = \{x_i\}_{i=1}^{N}$). c_l küme merkezi, $C=\{c_l\}_{l=1}^{K}$ kümesi içindeki merkezlerden biridir. Yukarıdaki performans fonksiyonundaki ikili toplama bütün x değerleri üzerinde tekli toplamaya dönüştürülebilir ve toplamalar altındaki kareli uzaklık MIN() fonksiyonu ile ifade edilebilir. KM algoritmasının performans fonksiyonu buna göre yeniden düzenlendiğinde;

$$\text{Perf}_{KM}\left(\{x_i\}_{i=1}^{N}, \{c_l\}_{l=1}^{K}\right) = \sum_{i=1}^{N} MIN\left\{\|x_i - c_j\|^2 \,|\, l = 1,...,K\right\} \tag{4.20}$$

Yukarıdaki KM' nin performans fonksiyonunda MIN() ile HA() ile yer değiştirdiğinde;

$$\text{Perf}_{KHM}\left(\{x_i\}_{i=1}^{N}, \{c_l\}_{l=1}^{K}\right) = \sum_{i=1}^{N} HA\left\{\|x_i - c_j\|^2 \,|\, l = 1,...,K\right\} = \sum_{i=1}^{N} \frac{K}{\sum_{l=1}^{K} \frac{1}{\|x_i - c_l\|^2}} \tag{4.21}$$

Yukarıdaki performans fonksiyonunda dıştaki toplamının içindeki değer, $\left\{\left\|x-c_l\right\|^2 \middle| l=1,...,k\right\}$ şeklindeki k tane kareli uzaklığın harmonik ortalamasıdır. Yukarıdaki KHM' in performans fonksiyonu genel uzaklık fonksiyonu d(x,m) kullanılarak elde edilmiştir. d(x,m) için istenen ağırlık fonksiyonuna sahip olmayan KHM algoritması için performans fonksiyonu yukarıdaki gibidir. KHM $_p$ içindeki d(x,m) gibi öklit uzaklığının p. kuvveti uygulanarak istenen ağırlık fonksiyonu türetilebilir. KHM $_p$' in performans fonksiyonu aşağıdaki gibidir [24]:

$$\text{Perf}_{KHM\ p}(X,C)=\sum_{i=1}^{N} HA\left\{\left\|x_i-c_l\right\|^p \middle| l=1,....,K\right\}=\sum_{i=1}^{N}\frac{K}{\sum_{l=1}^{K}\frac{1}{\left\|x_i-c_l\right\|^p}} \tag{4.22}$$

Yukarıdaki performans fonksiyonundaki p parametresi, girdi parametresidir. Genellikle $p\geq 2$ olarak alınmaktadır. Lineer olmayan fonksiyonların yerel en iyi değerini bulabilmek için birçok farklı optimizasyon algoritması vardır. Başlangıç seçimlerine çok fazla duyarlı olamayan KHM' in performans fonksiyonu için optimizasyon algoritması türetmek için, KHM' in performans fonksiyonunun merkez pozisyonlarına göre kısmi türevini almamız gereklidir. KHM 'in performans fonksiyonunun c_k' ya göre kısmi türevini alıp 0' a eşitlersek;

$$\frac{\partial Perf_{KHM}(X,C)}{\partial c_k}=-K\sum_{i=1}^{N}\frac{p(x_i-c_k)}{d_{i,k}^{p+2}\left(\sum_{l=1}^{K}\frac{1}{d_{i,l}^{p}}\right)^2}=\vec{0} \tag{4.23}$$

Denklemdeki 0 üzerine konulan ok, 0 vektörü olduğunu göstermek için kullanılır. c_k merkez pozisyonları da aynı zamanda vektördür. Bu denklemden c_k' ları çekildiğinde merkezleri hesaplamamızda yardımcı olacak aşağıdaki formül elde edilir:

$$c_k = \frac{\sum_{i=1}^{N} \frac{1}{d_{i,k}^{p+2}\left(\sum_{l=1}^{K} \frac{1}{d_{i,l}^{p}}\right)^2} x_i}{\sum_{i=1}^{N} \frac{1}{d_{i,k}^{p+2}\left(\sum_{l=1}^{K} \frac{1}{d_{i,l}^{p}}\right)^2}} \tag{4.24}$$

Her bir iterasyonda merkezlerin hesaplanmasında bu formül hesaplanır. KHM_p algoritması merkezlerin başlangıç pozisyonları ile başlar ve her bir merkez ile veri noktası arasındaki öklit uzaklığı hesaplanır ($d_{i,l} = \|x_i - ml\|$). Daha sonra da merkezlerin yeni pozisyonları yukarıdaki denklem kullanılarak hesaplanır. Yukarıdaki formül aşağıdaki küçük parçalara ayrılarak merkezlerin daha kolay bir şekilde hesaplanması sağlanabilir.

$$a_i = \frac{1}{\left(\sum_{l=1}^{K} \frac{1}{d_{i,l}^{p}}\right)^2} \tag{4.25}$$

$$q_{i,k} = \frac{a_i}{d_{i,k}^{p+2}} \tag{4.26}$$

$$q_k = \sum_{i=1}^{N} q_{i,k} \tag{4.27}$$

$$p_{i,k} = \frac{q_{i,k}}{q_k} \tag{4.28}$$

$$c_k = \sum_{i=1}^{N} p_{i,k} x_i \tag{4.29}$$

$q_{i,k}$'ın hesaplanması aşağıdaki şekilde yapılabilir:

$$q_{i,k} = \frac{d_{i,\min}^{2p}}{d_{i,l}^{p+2}\left[1+\sum_{l\neq\min}\left(\frac{d_{i,\min}}{d_{i,l}}\right)^{p}\right]^{2}} = \frac{d_{i,\min}^{p-2}\left(\frac{d_{i,\min}}{d_{i,k}}\right)^{p+2}}{\left[1+\sum_{l\neq\min}\left(\frac{d_{i,\min}}{d_{i,l}}\right)^{p}\right]^{2}} \tag{4.30}$$

KHM algoritması yumuşak üyelik fonksiyonuna sahiptir. KHM' in üyelik fonksiyonu aşağıda verilmiştir:

$$m_{KHM}(c_j|x_i) = \frac{\left\|x_i - c_j\right\|^{-p-2}}{\sum_{j=1}^{K}\left\|x_i - c_j\right\|^{-p-2}} \tag{4.31}$$

KHM algoritmasının üyelik fonksiyonun değeri veri noktası merkezin kendisine eşit ise ilgili olan veri noktasının o merkeze üyeliği 1 olarak alınırken, diğer merkezlere üyelik değeri 0 olarak alınır. KHM algoritması değişen ağırlık fonksiyonuna sahiptir. KHM' in ağırlık fonksiyonu aşağıda verilmiştir:

$$w_{KHM}(x_i) = \frac{\sum_{j=1}^{K}\left\|x_i - c_j\right\|^{-p-2}}{\left(\sum_{j=1}^{K}\left\|x_i - c_j\right\|^{-p}\right)^{2}} \tag{4.32}$$

KHM algoritmasının ağırlık fonksiyonu, her bir merkezden çok uzakta olan noktalara yüksek ağırlıklar verir. Bu şekilde veriyi ayırarak merkezlere yardım eder.Ağırlık fonksiyonu $w(x_i)>0$' dan olmalıdır. Ağırlık fonksiyonu $w(x_i)$, x_i veri noktasının bir sonraki iterasyonda merkezlerin hesaplanmasını nasıl etkileyeceğini göstermektedir.

KHM algoritmasının adımları aşağıda belirtilmiştir:

1) K küme sayısı belirlenir. Veri kümesi içinden k adet rasgele seçilmiş eleman küme merkezi olarak belirlenir.
2) Her bir veri noktasının merkezlere göre küme üyelik değerleri hesaplanır. Üyelik değerlerini hesaplamak için (4.31)' daki formül kullanılır.

3) Her bir elemanın ağırlığı hesaplanır. Her bir elemanın ağırlığını hesaplamak içinde (4.32)' deki formül kullanılır.
4) Yeni hesaplanan üyelik ve ağırlık değerlerine göre yeni merkezler hesaplanır. Hesaplan bu yeni merkezlere göre amaç fonksiyonu hesaplanır. Merkezleri hesaplamak için (4.24)' daki formül ya da bu formülün parçalanmış hali olan (4.25), (4,26), (4,27), (4,28), (4,29)' deki formüller sırasıyla kullanılır.
5) Eğer bir durdurma kıstası sağlanmamışsa adım 2' ye dönülerek işlemler tekrar edilir. Merkezlerin değişmemesi, üyelik değerlerinin değişmemesi gibi kıstaslar durdurma kıstasları olarak verilebilir.
6) Durdurma kıstası sağlamışsa üyelik değerleri durulaştırılarak her bir veri noktasının hangi kümeye ait olduğu belirlenir.

KHM algoritması, KM algoritmasına göre başlangıçta seçilen veri noktalarına daha az duyarlıdır.

4.4.4 Yeni kümeleme algoritmaları

KHM algoritmasının başlangıç durumundan çok fazla etkilenmemesi, KM, FKM ve EM merkez tabanlı kümeleme algoritmalarına göre performansının daha iyi olması bilim adamlarının dikkatini çekmiştir. Bilim adamları KHM algoritmasının diğerlerine göre daha iyi sonuçlar vermesinin nedenlerini araştırmaya başlamışlardır. KHM algoritmasını diğer merkez tabanlı kümeleme algoritmaları arasında farklı kılan yumuşak üyelik fonksiyonu ve değişen ağırlık fonksiyonuna sahip olmasıdır. KHM' in sahip olduğu üyelik fonksiyonu ve ağırlık fonksiyonundan hangisinin bu algoritmayı diğerlerine daha iyi yaptığını anlamak için Hybrid 1 ve Hybrid 2 adında iki tane algoritma yaratılmıştır. Bu algoritmalar aşağıdaki ayrıntılı bir şekilde açıklanmıştır [22].

4.4.4.1 Hibrit 1 (hybrid 1)

Hibrit 1, KM algoritmasının katı üyelik fonksiyonunu kullanır. Her bir veri noktası sadece kendisine en yakın olan küme merkezinin bulunduğu kümeye dâhil olur. Bir veri noktası kümenin içindedir ya da dışındadır. Eğer bir veri noktasının üyelik

değeri 1 ise ilgili kümenin üyesi, 0 ise ilgili kümenin üyesi değildir. Hibrit 1, KHM algoritmasının ağırlık fonksiyonunu kullanır. Bu ağırlık fonksiyonu, her bir merkezden uzakta olan noktalara daha fazla ağırlık verir. Bu algoritma KM' e göre ağırlıklardan dolayı çok daha hızlı yakınsamaktadır. Fakat katı üyelik fonksiyonuna sahip olduğundan KHM kadar iyi performansa sahip değildir [22].

Algoritmanın adımlarını aşağıda verilmiştir:

1) K küme sayısı belirlenir. Veri kümesi içinden k adet rasgele seçilmiş eleman küme merkezi olarak belirlenir.
2) Her bir veri noktası, kendisine en yakın olan merkezin bulunduğu kümeye atanır. Örneğin x_1 veri noktası, c_j merkezine en yakın ise $m_{KM}(c_1|x_1)=1$ olurken, x_1 noktasının diğer tüm merkezlere üyelikleri 0 olur. Veri noktalarının merkezlere göre üyeliklerinin tespitinde (4.3)' teki denklem kullanılır.
3) Her bir elemanın ağırlığı hesaplanır. Her bir elemanın ağırlığını hesaplamak içinde (4.32)' deki formül kullanılır.
4) Yeni hesaplanan üyelik ve ağırlık değerlerine göre yeni merkezler hesaplanır. Merkezlerin hesaplanmasında (4.1)' deki denklemden yararlanılır.
5) Eğer bir durdurma kıstası sağlanmamışsa adım 2' ye dönülerek işlemler tekrar edilir. Merkezlerin değişmemesi, üyelik değerlerinin değişmemesi gibi kıstaslar durdurma kıstasları olarak verilebilir.

4.4.4.2 Hibrit 2 (hybrid 2)

Hibrit 2, KHM algoritmasının yumuşak üyelik fonksiyonunu kullanır. Her bir veri noktasının merkezlere kısmen üyeliği vardır. Üyelik fonksiyonun değeri veri noktası merkezin kendisine eşit ise o merkeze üyeliği 1 olarak alınırken, diğer merkezlere üyelik değeri 0 olarak alınır. Hibrit 2, KM algoritmasının sabit ağırlık fonksiyonunu kullanır. Bu algoritma KM' e göre sahip olduğu üyelik fonksiyonundan dolayı daha iyi performans göstermektedir [22].

Algoritmanın adımlarını aşağıda verilmiştir:

1) K küme sayısı belirlenir. Veri kümesi içinden k adet rasgele seçilmiş eleman küme merkezi olarak belirlenir.

2) Her bir veri noktasının merkezlere göre küme üyelik değerleri hesaplanır. Üyelik değerlerini hesaplamak için (4.31)' daki formül kullanılır.
3) Yeni hesaplanan üyelik değerlerine göre yeni merkezler hesaplanır. Merkezlerin hesaplanmasında (4.1)' deki denklemden yararlanılır.
4) Eğer bir durdurma kıstası sağlanmamışsa adım 2' ye dönülerek işlemler tekrar edilir. Merkezlerin değişmemesi, üyelik değerlerinin değişmemesi gibi kıstaslar durdurma kıstasları olarak verilebilir.
5) Durdurma kıstası sağlamışsa üyelik değerleri durulaştırılarak her bir veri noktasının hangi kümeye ait olduğu belirlenir.

5. MERKEZ TABANLI KÜMELEME ALGORİTMALARININ KARŞILAŞTIRILMASI

5.1 Giriş

Bu bölümde ilk başta UCI veri deposundan alınan dört veritabanına ilişkin karakteristik özelliklere yer verilmiş ardından da veritabanları içindeki nitelikler istatistiksel olarak ele alınıp incelenmiştir. Kümelemede kullanılacak veritabanları ile ilgili açıklamalardan sonra merkez tabanlı kümeleme algoritmaları olan k-ortalama, bulanık k-ortalama, k-harmonik ortalama, hibrit 1 ve hibrit 2 algoritmalarının performans ve işlemci zamanı bakımından karşılaştırılacağı kıstaslara değinilmiştir. Daha sonra geliştirilen uygulamaya ait olan arayüzler sırasıyla tanıtılmıştır. Ardından merkez tabanlı kümeleme algoritmaları belirtilen kıstaslar doğrultusunda sırasıyla karşılaştırılmış ve her bir kıstasa ilişkin karşılaştırma sonuçları tablolar ile görsel olarak ifade edilmiştir.

5.2 Karşılaştırmada Kullanılan Veritabanları

Bu çalışmada UCI veri deposundan alınan her biri farklı sayıda kayıt ve nitelik içeren 4 tane veritabanından yararlanılmıştır. Bunlar; süsen çiçeği (iris), cam (glass identification), Pima Hindistan diyabet hastalıklarını içeren veritabanı ve mamografi (Mammographic_Masses) veritabanlarıdır. Bu çalışmada merkez tabanlı kümeleme algoritmaları karşılaştırılmaktadır. Karşılaştırma işlemi yapıldığı için geçerliliği ve doğruluğu kanıtlanmış veritabanlarına ihtiyarcımız vardı. Bu nedenle bu çalışma da UCI veri deposundan seçtiğimiz 4 veritabanından yararlanılmıştır [58]. Bu veritabanları sayıca ve içeriğindeki veri türleri bakımından birbirinden farklı veritabanlarıdır. Bu veritabanlarının karakteristik özellikleri Tablo 5.1' de belirtilmiştir. Süsen çiçeği, cam, diyabet ve mamografi veritabanlarının her biri aşağıda ayrıntılı bir şekilde açıklanmıştır.

Tablo 5.1: UCI veri deposundan alınan 4 veritabanının karakteristik bilgileri [58].

Veritabanı	**Verilerin Sayısı**	**Sınıf Sayısı**	**Her Bir Sınıftaki Verilerin Sayısı**	**Niteliklerin Sayısı**	**Yıl**
Süsen Çiçeği	150	3	(50, 50, 50)	4	1988
Cam	214	7	(70, 17, 76, 0, 13, 9, 29)	10	1987
Diyabet	768	2	(500, 268)	8	1990
Mamografi	961	2	(516, 445)	6	2007

Bu çalışmada kullanılan UCI veri deposundan alınan veritabanları öncelikle bozuk verilerden arındırılmış ve program aracılığı ile kullanılabilir hale getirilmiştir. Veritabanları UCI veri deposundan metin dosyası formatında alınmış ve Access veritabanına aktarılmıştır. Düzenlemeler için Access sorgu nesnesi kullanılmıştır. Veritabanında aşağıdaki işlemler yapılmıştır:

Kayıt sırası gibi karşılaştırma işleminde gerekli olmayan alanlar çıkartılmıştır. Boş (null) değer içeren sayısal alanlara 0 değeri atanmıştır.

Bu tez çalışmasında süsen çiçeği, cam, diyabet ve mamografi veritabanları içindeki veriler üzerinde Min-Max normalleştirme tekniği kullanılmıştır. Min-Max normalleştirmesi ile veriler, yeni veri aralığına doğrusal dönüşüm ile dönüştürülmüşlerdir. Bu veri aralığı genellikle [0,1] aralığındadır. Min-Max normalleştirmesinde kullanılan formül aşağıda verilmiştir:

$$v' = \frac{v - \min_A}{\max_A - \min_A} \tag{5.1}$$

Yukarıdaki formülde kullanılan v' değişkeni yeni veriyi, v değişkeni eski veriyi temsil etmektedir. $\min_A$ değişkeni A niteliğine ait olan değerler içindeki minimum

değeri, $\max_A$ değişkeni ise A niteliği ait olan değerler içindeki maksimum değeri temsil etmektedir. Merkez tabanlı kümeleme algoritmaları hesaplamalarında bu tez çalışmasında Öklit ve Manhattan uzaklık ölçümlerinden yararlanılmaktadır. Bu uzaklık ölçümlerinden Öklit uzaklık ölçümü büyük değerlere sahip olan niteliklerden çok fazla etkilenmektedir. Her niteliğin uzaklık ölçümlerinin hesaplanmasına önemli ölçüde katılması için veriler üzerinde Min-Max normalleştirmesi kullanılmıştır.

5.2.1 Süsen Çiçeği Veritabanı

Süsen çiçeği (iris) veritabanı Fisher tarafından tanıtılan çok değişkenli popüler bir veritabanıdır. Süsen çiçeği (Iris) veritabanı 150 tane kayıt ve 4 tane nitelik değeri içermektedir. Veritabanındaki nitelikler çanak yaprak uzunluğu, çanak yaprak genişliği, taç yaprak uzunluğu, taç yaprak genişliği olmak üzere 4 tanedir. Süsen çiçeği veritabanı Tablo 5.2' de gösterilmiştir. Süsen çiçeği veritabanı 3 sınıf içerir. Her bir sınıfta 50 tane örnek vardır. Her bir sınıf süsen çiçeği bitkisinin bir çeşidini ifade eder. Süsen bitkisinin çeşitleri sırası ile Setosa, Versicolor ve Virginica'dır. Üç sınıfın her biri için dağılım %33' tür. Şekil 5.1' de süsen çiçeğinin çeşitleri görsel olarak aşağıda gösterilmiştir:

Şekil 5.1: Süsen çiçeğinin soldan sağa Setosa, Virginica ve Versicolor çeşitleri.

Süsen çiçeği veritabanı UCI veri deposundan metin dosyası formatın alınmış uygulamada kullanılmak üzere Access veritabanına aktarılmıştır Süsen çiçeği veritabanı UCI veri deposundan alındığında eksik nitelik değeri içermemekteydi. Bu nedenle süsen çiçeği veritabanı üzerinde düzeltmeler yapılmamıştır.

Tablo 5.2: Süsen çiçeğine ait bilgileri içeren veritabanı.

Çanak Yaprak Uzunluk	Çanak Yaprak Genişlik	Taç Yaprak Uzunluk	Taç Yaprak Genişlik	Bitki Türü
5,1	3,5	1,4	0,2	Setosa
4,9	3	1,4	0,2	Setosa
4,7	3,2	1,3	0,2	Setosa
4,6	3,1	1,5	0,2	Setosa
5	3,6	1,4	0,2	Setosa
5,4	3,9	1,7	0,4	Setosa
4,6	3,4	1,4	0,3	Setosa
5	3,4	1,5	0,2	Setosa
4,4	2,9	1,4	0,2	Setosa
4,9	3,1	1,5	0,1	Setosa
5,4	3,7	1,5	0,2	Setosa
4,8	3,4	1,6	0,2	Setosa
4,8	3	1,4	0,1	Setosa
4,3	3	1,1	0,1	Setosa
5,8	4	1,2	0,2	Setosa

Süsen çiçeğine ait olan nitelikler istatistiksel olarak analiz edilmiş ve her bir niteliğe ait olan minimum, maksimum, ortalama, varyans ve standart sapma değerleri hesaplanmıştır. Minimum değer, bir veri kümesi içindeki en küçük değere eşdeğerdir. Maksimum değer de bir veri kümesi içindeki en büyük değere eşdeğerdir. Ortalama değer, bir veri kümesi içindeki değerlerin toplamının veri sayına bölünmesi ile elde edilir. Aynı zamanda ortalama değer aritmetik ortalama değer anlamına da gelmektedir. Varyans değeri ise bir veri kümesindeki değerlerin aritmetik ortalamadan sapmalarının kareler ortalaması şeklinde hesaplanır. Varyans değerinin hesaplanmasına ilişkin formül aşağıda verilmiştir. Bu formüldeki N veri kümesindeki kayıtların sayısını, x_i veri kümesi içindeki i. kaydı, μ veri kümesinin aritmetik ortalama değerini ve σ^2 ise varyans değerinin ifade etmektedir.

$$\sigma^2 = \frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N} \tag{5.2}$$

Standart sapma ise varyans değerinin karekökünün alınması ile elde edilir. Standart sapma denklemi aşağıda verilmiştir:

$$\sigma = \sqrt{\sigma^2} = \frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N} \tag{5.3}$$

Bir veri kümesinin sadece bir tane aritmetik ortalama değeri vardır. Aritmetik ortalama değeri bir veri setindeki aşırı değerlerden kolay bir şekilde etkilenmektedir. Aritmetik ortalama bir dağılımın orta noktasını göstermektedir. Fakat aritmetik ortalama bir dağılımın yaygınlığı hakkında bilgi vermez. Varyans değeri, standart sapma değeri bir dağılımın yaygınlığı hakkında bilgi veren ölçülerden sadece birkaçıdır. Varyans değerini yorumlamak oldukça güçtür. Çünkü varyansın birimi varyans değerini oluşturan verilerin ölçü biriminin karesidir. Bu nedenle varyansın karekökü alınır ve verilerin ölçü birimi ile aynı ölçüye sahip olan standart sapma değeri elde edilir. Standart sapma değeri bir veri kümesi içindeki değerlerin ortalama değere ne kadar uzaklıkta olduğunu gösterir. Standart sapma arttıkça bir dağılımın yaygınlığı artar. Süsen çiçeği bitkisinin nitelikleri ve istatistiksel değerleri Tablo 5.3' de gösterilmiştir [57] .

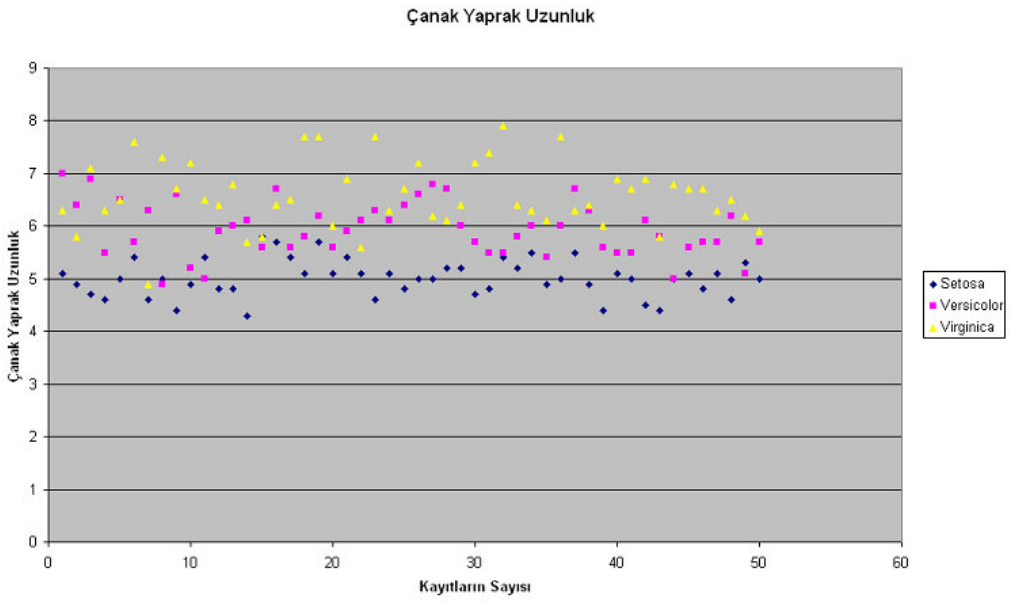
Tablo 5.3: Süsen çiçeğine ait niteliklerin istatistiksel analiz değerleri.

Nitelikler	**Minimum Değer**	**Maksimum Değer**	**Ortalama Değer**	**Varyans Değeri**	**Standart Sapma Değeri**
Çanak Yaprak Uzunluk	4.3	7.9	5.843	0.681	0.825
Çanak Yaprak Genişlik	2	4.4	3.054	0.187	0.432
Taç Yaprak Uzunluk	1	6.9	3.759	3.092	1.759
Taç Yaprak Genişlik	0.1	2.5	1.199	0.579	0.761

Tablo 5.3' teki değerler incelendiğinde taç yaprak uzunluk nitelik değeri 3.759 ortalama değere sahipken, 3.092 varyans değeri ve 1.759 standart sapma değerine sahiptir. Varyans değeri ve standart sapma değeri diğer niteliklere göre oldukça büyüktür. Bu değerlerin oldukça büyük olması dağılımın geniş olduğunu gösterir.

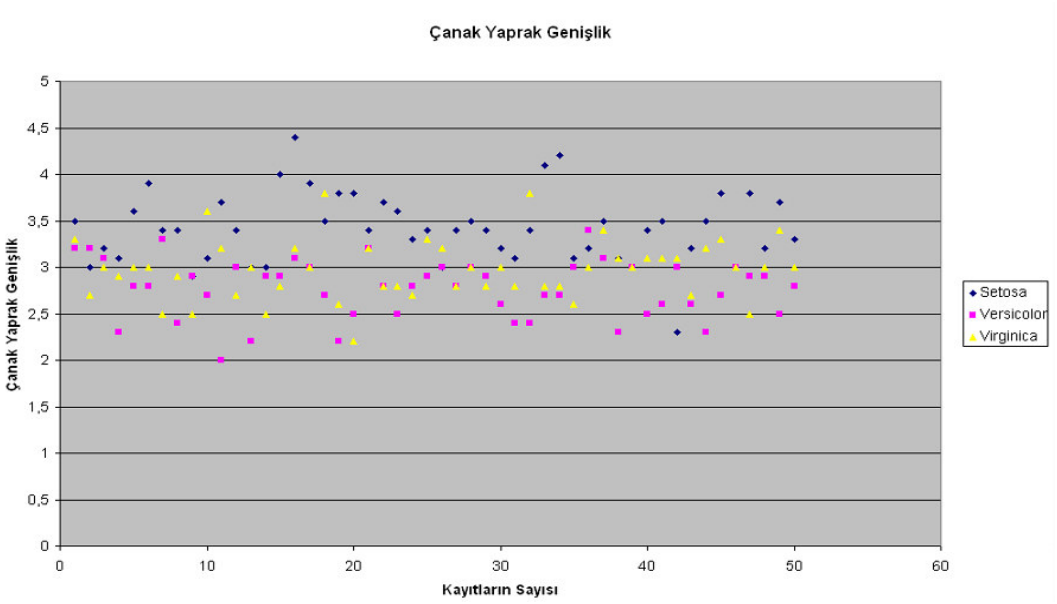
Bu değerlerin büyük olması Şekil 5.4' teki kümelerin küçük bir çakışma ile çok iyi bir şekilde birbirinden ayrılmasının nedeni açıklar. Çanak yaparak genişlik niteliğinin ortalama değeri 3.054, varyans değeri 0.187 ve standart sapma değeri 0.432' dir. Varyans değeri ve standart sapma değeri diğer niteliklerin varyans ve sapma değerlerine göre oldukça düşüktür. Bu değerlerin düşük olması dağılımın yaygınlığının küçük yani ortalamadan sapmaların küçük olduğunu gösterir. Dağılımın küçük olması nedeniyle Şekil 5.3' teki veri noktalarının özellikle ortalama etrafında birbiri ile çakıştığı görülebilmektedir.

Süsen çiçeğinin çeşitlerinin her bir nitelik değerleri için şekilsel olarak karşılaştırılması Şekil 5.2, Şekil 5.3, Şekil 5.4 ve Şekil 5.5' te yapılmıştır. Şekil 5.2' de süsen çiçeği çanak yaprak uzunluk nitelik değerine göre Setosa, Versicolor ve Virginica kümelerine ayrılmıştır. Şekil 5.1' de görüldüğü gibi Setosa, Versicolor ve Virginica kümeleri içindeki bazı veri noktaları birbirleri ile çakışmaktadır. Şekil 5.2' e bakarak süsen çiçeğinin türleri ile ilgili bazı gözlemler yapmak mümkündür. Virginica türündeki süsen çiçekleri diğer türler ile karşılaştırıldığında en uzun çanak yapraklara sahiptir. Versicolor türündeki süsen çiçekleri diğer türlere göre orta uzunlukta çanak yapraklara sahiptir. Setosa türündeki süsen çiçekleri diğer türlere göre en kısa çanak yapraklara sahiptir.



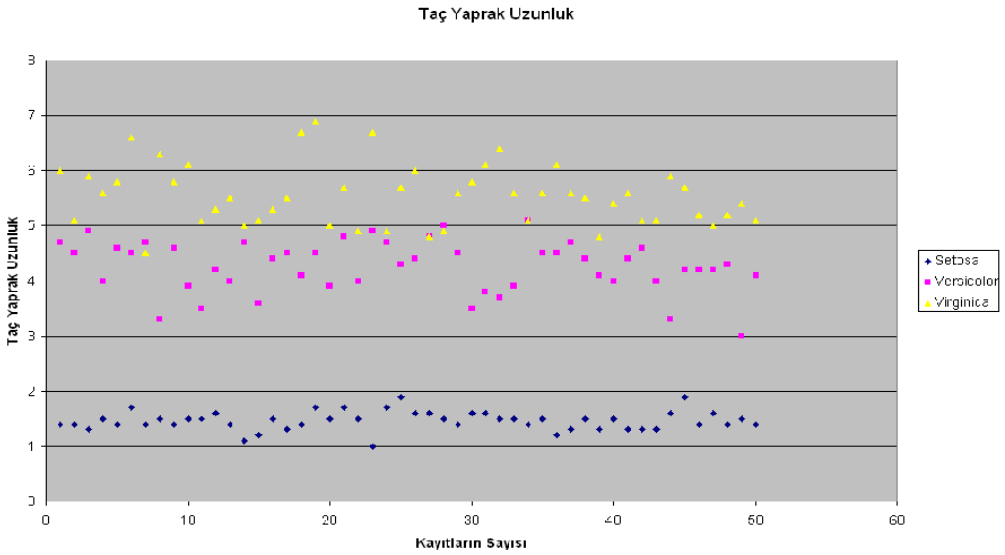
Şekil 5.2: Süsen çiçeğinin çeşitlerinin çanak yaprak uzunluk değerleri.

Şekil 5.3’ de süsen çiçeği çanak yaprak genişlik nitelik değerine göre Setosa, Versicolor ve Virginica kümelerine ayrılmıştır. Şekil 5.3’ de görüldüğü gibi Setosa, Versicolor ve Virginica kümeleri içindeki bazı veri noktaları birbirleri ile çakışmaktadır. Şekilde de görüldüğü gibi Setosa türündeki süsen çiçekleri diğer türler ile karşılaştırıldığında en geniş çanak yapraklara sahiptir. Virginica türündeki süsen çiçekleri diğer türlere göre orta genişlikte çanak yapraklara sahiptir. Versicolor türündeki süsen çiçekleri diğerlerine göre daha dar çanak yapraklara sahiptir. Fakat şekilde de görüldüğü gibi Versicolor ve Virginica türleri çanak yaprak genişlik nitelik değerine göre birbirlerinden tam olarak ayırt edilememektedir. Çanak yaprak genişlik niteliği süsen çiçeğinin türlerinin ayırt edilebilmesinde belirgin bir etkiye sahip değildir.



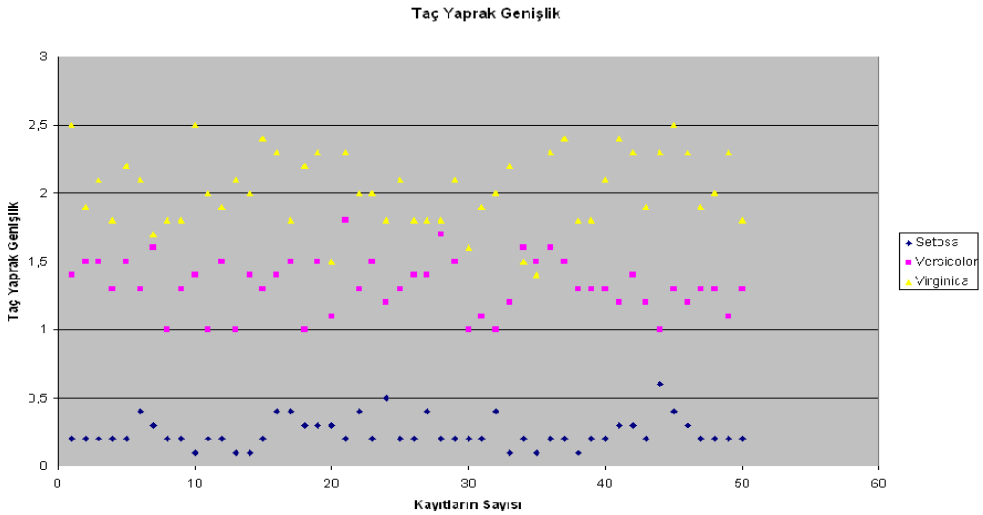
Şekil 5.3: Süsen çiçeğinin çeşitlerinin çanak yaprak genişlik değerleri.

Şekil 5.4’ de süsen çiçeği taç yaprak uzunluk nitelik değerine göre Setosa, Versicolor ve Virginica kümelerine ayrılmıştır. Şekil 5.4’ de görüldüğü gibi Setosa, Versicolor ve Virginica kümeleri çok az veri noktasının çakışması dışında çok belirgin şekilde birbirinden ayrılmıştır. Şekil 5.4’ de görüldüğü gibi Virginica türündeki süsen çiçekleri diğer türler ile karşılaştırıldığında en uzun taç yapraklara sahiptir. Versicolor türündeki süsen çiçekleri orta uzunlukta taç yapraklara sahiptir. Setosa türündeki süsen çiçekleri diğer türler ile karşılaştırıldığında en kısa taç yapraklara sahiptir.



Şekil 5.4: Süsen çiçeğinin çeşitlerinin taç yaprak uzunluk değerleri.

Şekil 5.5' de süsen çiçeği taç yaprak genişlik nitelik değerine göre Setosa, Versicolor ve Virginica kümelerine ayrılmıştır. Şekil 5.5' de görüldüğü gibi Setosa, Versicolor ve Virginica kümeleri içindeki veri noktaları birbirleri ile çakışmamaktadır. Şekil 5.5' de görüldüğü gibi Virginica türündeki süsen çiçekleri diğer türler ile karşılaştırıldığında en geniş taç yapraklara sahiptir. Versicolor türündeki süsen çiçekleri orta genişlikte taç yapraklara sahiptir. Setosa türündeki süsen çiçekleri diğer türler ile karşılaştırıldığında en dar taç yapraklara sahiptir.



Şekil 5.5: Süsen çiçeğinin çeşitlerinin taç yaprak genişlik değerleri.

Sonuç olarak yukarıdaki dört şekilden yola çıkaracak süsen çiçeğini türlerine ayırt etmede yardımcı olabilecek bazı tahminlerde bulunabilir. Eğer bir süsen çiçeğinin çanak yaprak uzunluğu 6–8 arasında, taç yaprak uzunluğu 4.8–7 arasında, taç yaprak genişliği de 1.5–2.5 arasından ise bu süsen çiçeği büyük olasılıkla Virginica türündedir. Bir süsen çiçeğinin çanak yaprak uzunluğu 4.5–5.5 arasında, taç yapak uzunluğu 1–2 arasında ve taç yaprak genişliği 0.1–0.5 arasında ise bu süsen çiçeği büyük ihtimalle Setosa türündedir. Eğer bir süsen çiçeği Setosa ve Virginica türleri arasına düşüyorsa bu süsen çiçeği büyük olasılıkla Versicolor türündedir. Süsen çiçeğinin türünün tahmin edilmesinde çanak yaprak genişlik niteliği kullanılmamıştır. Çünkü Setosa, Virginica ve Versicolor türlerinin çanak yaprak genişlik değerleri birbiri içine girmiştir ve türler için bu nitelik değerleri belirgin değildir.

5.2.2 Cam Veritabanı

İnsanoğlu tarafından yaklaşık 5000 yıldır yapılan cam, kumun eritilip şekillendirilip soğutulmasıyla ortaya çıkmaktadır. Uygulama da kullanılacak olan cam veritabanı German tarafından oluşturulmuş ve Spiehler tarafından kullanımına sunulmuştur. Cam veritabanı 214 tane kayıt ve 9 tane nitelik içeren 9 boyutlu bir veritabanıdır. Cam veritabanı Tablo 5.4' de gösterilmiştir.

Tablo 5.4: Cam veritabanı içindeki nitelik değerleri.

Kırılma İndisi	Sodyum	Magnezyum	Alüminyum	Silisyum	Potasyum	Kalsiyum	Baryum	Demir	Cam çeşidi
1,52101	13,64	4,49	1,1	71,78	0,06	8,75	0	0	Bina penceresi düz cam
1,51761	13,89	3,6	1,36	72,73	0,48	7,83	0	0	Bina penceresi düz cam
1,51618	13,53	3,55	1,54	72,99	0,39	7,78	0	0	Bina penceresi düz cam
1,51766	13,21	3,69	1,29	72,61	0,57	8,22	0	0	Bina penceresi düz cam
1,51742	13,27	3,62	1,24	73,08	0,55	8,07	0	0	Bina penceresi düz cam
1,51596	12,79	3,61	1,62	72,97	0,64	8,07	0	0,26	Bina penceresi düz cam
1,51743	13,3	3,6	1,14	73,09	0,58	8,17	0	0	Bina penceresi düz cam
1,51756	13,15	3,61	1,05	73,24	0,57	8,24	0	0	Bina penceresi düz cam
1,51918	14,04	3,58	1,37	72,08	0,56	8,3	0	0	Bina penceresi düz cam
1,51755	13	3,6	1,36	72,99	0,57	8,4	0	0,11	Bina penceresi düz cam
1,51571	12,72	3,46	1,56	73,2	0,67	8,09	0	0,24	Bina penceresi düz cam
1,51763	12,8	3,66	1,27	73,01	0,6	8,56	0	0	Bina penceresi düz cam
1,51589	12,88	3,43	1,4	73,28	0,69	8,05	0	0,24	Bina penceresi düz cam
1,51748	12,86	3,56	1,27	73,21	0,54	8,38	0	0,17	Bina penceresi düz cam
1,51763	12,61	3,59	1,31	73,29	0,58	8,5	0	0	Bina penceresi düz cam
1,51761	12,81	3,54	1,23	73,24	0,58	8,39	0	0	Bina penceresi düz cam
1,51784	12,68	3,67	1,16	73,11	0,61	8,7	0	0	Bina penceresi düz cam

Cam veritabanı UCI veri deposundan metin dosyası formatın alınmış uygulamada kullanılmak üzere Access veritabanına aktarılmıştır. Cam veritabanı UCI veri deposundan alındığında eksik nitelik değeri içermemekteydi. Cam veritabanı ilk

alındığında kayıtların sıra numarası da veritabanında bulunmaktaydı. Fakat kayıtların sırasına ihtiyarcımız olmadığından ilgili sütun yani nitelik değeri kaldırılmıştır. Bu değişiklik dışında veritabanı üzerinde herhangi bir düzenleme yapılmamıştır. Cam veritabanındaki nitelikler sırası ile kırılma indisi, sodyum(Na), magnezyum(Mg), Alüminyum(Al), Silisyum(Si), Potasyum(K), Kalsiyum(Ca), Baryum(Ba) ve Demir(Fe)' dir.

Cam üretiminde düz (float) cam önemli bir yere sahiptir. Düz cam, cam eriyiğinin erimiş kalay üzerinde yüzdürülmesi (floating) yoluyla elde edilir. Bu işlem camın iki yüzünün birbirine paralel ve hatasız olmasını sağlar. Belirtilen nitelik değerleri göz önüne alındığında cam veritabanındaki camlar 7 çeşide ayrılmıştır. Birinci sınıfta bina penceresi düz cam olan 70 kayıt, ikinci sınıfta bina penceresi düz cam olmayan 76 kayıt, üçüncü sınıfta araç penceresi düz cam olan 17 kayıt, dördüncü sınıfta araç penceresi düz cam olmayan 0 kayıt içermektedir. Bu ilk 4 sınıf pencere camlarının çeşitlerini içermektedir. Beşinci, altıncı ve yedinci cam çeşitleri pencere camı olmayan cam çeşitleridir. Beşinci sınıf şişe, kap gibi eşyalarda kullanılan cam, altıncı sınıf tabak, çanak gibi sofra takımlarında kullanılan cam ve yedinci sınıf farlarda kullanılan cam çeşitlerini içermektedir. Bu sınıflarda sırasıyla bulunan kayıt sayısı 13, 9 ve 29' dur. Birinci sınıfın dağılımı %32.71, ikinci sınıfın dağılımı %35.51, üçüncü sınıfın dağılımı %7.94, dördüncü sınıfın dağılımı %0, beşinci sınıfın dağılımı %6.08, altıncı sınıfın dağılımı %4.21 ve yedinci sınıfın dağılımı ise %15.55' dir.

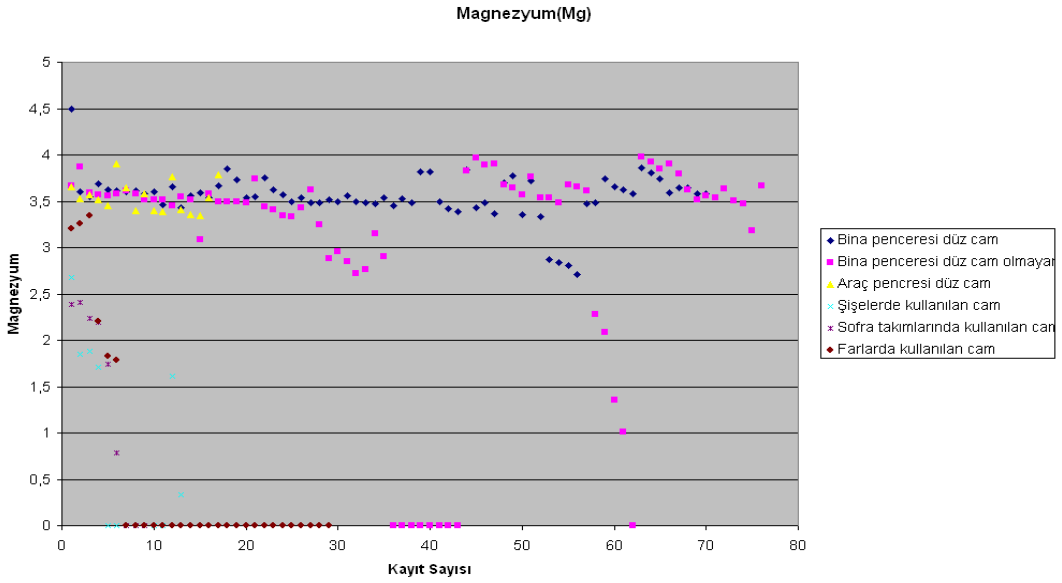
Cam veritabanındaki nitelikler ve niteliklere ait olan istatistiksel değerler Tablo 5.5' de gösterilmiştir. Tablo 5.5' te görüldüğü gibi magnezyum niteliğinin varyans değeri ve standart sapma değeri diğer niteliklerin değerlerine göre oldukça yüksektir. Magnezyumun varyans değeri 2.071 ve standart sapma değeri 1.439' dur. Bu değerlerin oldukça büyük olması dağılımın geniş olduğunu gösterir. Ardından en yüksek varyans ve standart sapma değerine sahip olan ikinci nitelik kalsiyumdur. Kalsiyumun varyans değeri 2.016 ve standart sapma değeri 1.419' dur. Bu değerlerin büyük olması ortalamadan sapmaların ne kadar büyük olduğunu gösterir. Kırılma indisinin varyans değeri 0.000009 ve standart sapma değeri 0.003' tür ve diğer niteliklerin varyans ve standart sapma değerinden daha düşüktür. Bu değerlerin küçük olması dağılımın küçük olduğunu göstermektedir. Magnezyum ve kalsiyum

niteliklerinin değerleri ortalama sapmadan diğer nitelik değerlerine göre daha büyük olsa bile yine de üst üste binen veri noktaları vardır. Camların sınıflarına ayırt edilmesinde bu değerler yeterince etkili değildir. Bu veritabanı içindeki verilerin beklenen sınıflara ayrılmaması şaşırtıcı bir durum değildir. Çünkü sınıfların ayırt edilmesinde kullanılacak olan niteliklerin değerleri birbirine çok yakın ve ayırt edici değildir.

Tablo 5.5: Cam veritabanına ait niteliklerin istatistiksel analiz değerleri.

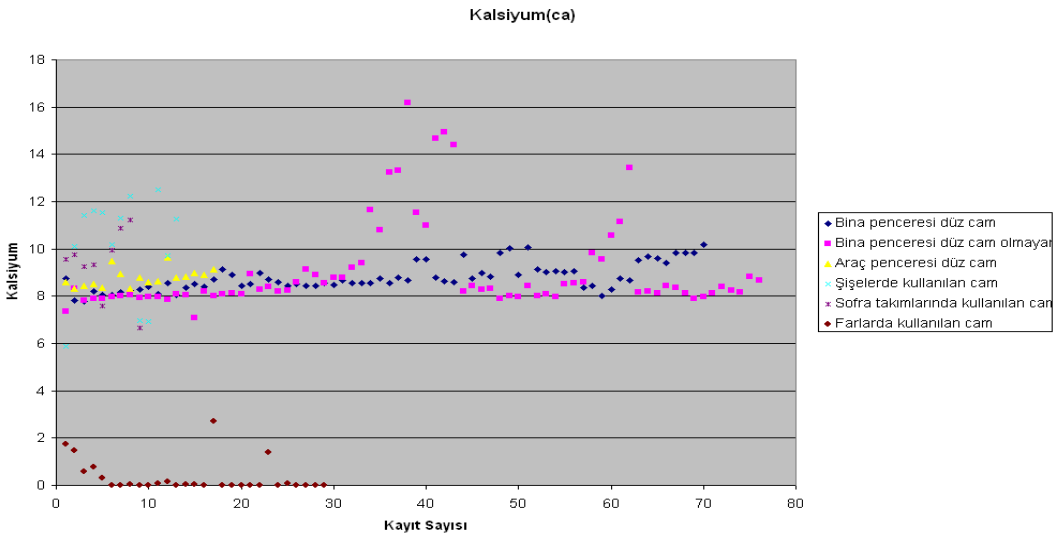
Nitelikler	**Minimum Değer**	**Maksimum Değer**	**Ortalama Değer**	**Varyans Değeri**	**Standart Sapma Değeri**
Kırılma İndisi	1.511	1.534	1.518	0.000009	0.003
Sodyum	10.73	17.38	13.408	0.664	0.815
Magnezyum	0	4.49	2.685	2.071	1.439
Alüminyum	0.29	3.5	1.445	0.248	0.498
Silisyum	69.81	75.41	72.651	0.597	0.773
Potasyum	0	6.21	0.497	0.423	0.651
Kalsiyum	5.43	16.19	8.957	2.016	1.419
Baryum	0	3.15	0.175	0.246	0.496
Demir	0	0.51	0.057	0.009	0.097

Cam çeşitlerinin her bir nitelik değerleri için şekilsel olarak karşılaştırılması Şekil 5.6, Şekil 5.7 ve Şekil 5.8' de yapılmıştır. Şekil 5.6' de camlar magnezyum nitelik değerine göre bina penceresi düz cam, bina penceresi düz cam olmayan, araç penceresi düz cam, şişelerde kullanılan cam, sofra takımlarında kullanılan cam ve farlarda kullanılan cam olacak şekilde 6 sınıfa ayrılmıştır. Şekil 5.6' de görüldüğü bazı veri noktaları birbirleri ile çakışmaktadır. Her bir sınıf içindeki veri noktalarının magnezyum değerleri birbirine çok yakın ya da aynıdır. Bu nedenle sınıfların belirlenmesinde magnezyum niteliğinin standart sapma ve varyans değeri büyük olmasına rağmen tam olarak etkili değildir.



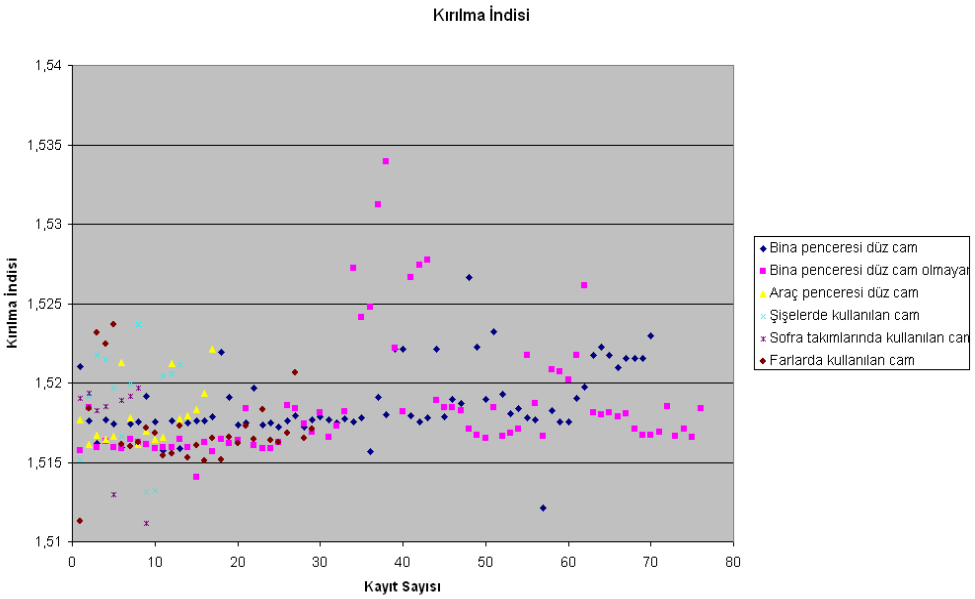
Şekil 5.6: Cam çeşitlerine ait olan magnezyum değerleri.

Şekil 5.7' de görüldüğü bazı veri noktaları birbirleri ile çakışmaktadır. En yüksek kalsiyum değerine bina penceresi düz cam olmayan sınıf içindeki veri noktaları sahiptir. Kalsiyum değerinin özellikle 8-10 olduğu durumlarda bina penceresi düz cam, araç penceresi düz cam ve bina penceresi düz cam olmayan sınıflarına ait olan veri noktaları bulunmaktadır. Bu veri noktaların hangi sınıf içinde olduğu ayırt etmek zordur. Buna karşın 0–2 değer aralığında farlarda kullanılan cam sınıfına ait olan veri noktaları belirgin şekilde ayırt edilebilmektedir.



Şekil 5.7: Cam çeşitlerine ait olan kalsiyum değerleri.

Şekil 5.8' de görüldüğü bazı veri noktaları birbirleri ile çakışmaktadır. En yüksek kırılma indisi değerine bina penceresi düz cam olmayan sınıf içindeki veri noktaları sahiptir. Her bir sınıf içindeki veri noktalarının kırılma indisi değerleri birbirine çok yakın ya da aynıdır. Bu nedenle sınıfların belirlenmesinde kırılma indisi niteliği tam olarak etkili değildir.



Şekil 5.8: Cam çeşitlerine ait olan kırılma indisi değerleri.

5.2.3 Diyabet Veritabanı

UCI veri deposundan alınan diyabet veritabanını incelemeden önce diyabet hastalığı hakkında genel bilgi vermemiz gereklidir. Diyabet, diğer adıyla şeker hastalığı karbonhidratlar, protein ve yağ mekanizmasını ilgilendiren bir metabolizma hastalığıdır. Diyabet, pankreas adı verilen salgı bezinin yeterli düzeyde insülin hormonu üretmemesi ya da üretilen insülin hormonunun etkili bir şekilde kullanılmaması durumunda ortaya çıkar. Besinlerden kana geçen şeker, insülin hormonu aracılığıyla hücreler geçer. Hücreler şekeri enerji için kullanırlar. Şekerin ihtiyaçtan fazlası sonra kullanılmak üzere karaciğerde depolanır. Yeteri düzeyde insülin hormonu üretilemezse, besinlerle alınan şeker hücrelere geçemez ve kandaki şeker yoğunluğu artar. Kandaki şeker miktarının artması zehir etkisi yaratarak tüm hücrelerin tahrip olmasına neden olur. Bu nedenle diyabet hastalarının kendilerine çok fazla dikkat etmeleri gerekmektedir.

Uygulamada kullanılacak olan diyabet veritabanı 768 kayıt ve 8 tane nitelik değeri içermektedir. Pima Hintlilerine ait olan diyabet veritabanında hastaların hepsi en az 21 yaşına olan kadınlardır. Diyabet veritabanına it olan değerler Tablo 5.6' de gösterilmiştir.

Tablo 5.6: Diyabet veritabanı içindeki nitelik değerleri.

Hamilelik Sayısı	Plazma Glukoz Değerleri	Diyastolik Kan Basıncı	Tricep Deri Katlanma Kalınlığı	2 Saatlik Serum İnsülin	Vücut Kitle Oranı
6	148	72	35	0	33,6
1	85	66	29	0	26,6
8	183	64	0	0	23,3
1	89	66	23	94	28,1
0	137	40	35	168	43,1
5	116	74	0	0	25,6
3	78	50	32	88	31
10	115	0	0	0	35,3
2	197	70	45	543	30,5
8	125	96	0	0	0
4	110	92	0	0	37,6
10	168	74	0	0	38
10	139	80	0	0	27,1
1	189	60	23	846	30,1
5	166	72	19	175	25,8
7	100	0	0	0	30

Diyabet veritabanı UCI veri deposundan metin dosyası formatın alınmış uygulamada kullanılmak üzere Access veritabanına aktarılmıştır. Diyabet veritabanı UCI veri deposundan alındığında eksik nitelik değeri içermemekteydi. Bu nedenle veritabanı üzerinde herhangi bir değişiklik yapılmamıştır. Diyabet veritabanındaki nitelikler sırası ile hamilelik sayısı, plazma glukoz değerleri (Plasma glucose concentration), diyastolik kan basıncı (küçük kan basıncı), tricep deri katlanma kalınlığı, 2 saatlik serum insülin, vücut kitle oranı (body mass index), ailedeki şeker hastalığı fonksiyonu (diabetes pedigree function) ve yaş nitelikleridir. Diyabet veritabanı belirtilen nitelik değerleri göz alındığında 2 sınıfa ayrılmıştır. Sınıf nitelik değeri 0 ise diyabet hastalığı olmadığı, 1 ise diyabet hastalığı olduğu anlamına gelmektedir. Birinci sınıfta 500, ikinci sınıfta 268 örnek vardır. Birinci sınıf diyabet hastalığı olmayanları ifade ederken, ikinci sınıf diyabet hastalığı olanları ifade etmektedir.

Diyabet veritabanındaki nitelikler ve niteliklere ait olan istatistiksel değerler Tablo 5.7' de gösterilmiştir. Tablo 5.7' te görüldüğü gibi 2 saatlik serum insülin niteliğinin varyans değeri ve standart sapma değeri diğer niteliklerin değerlerine göre oldukça yüksektir. 2 Saatlik serum insülin varyans değeri 13263.887 ve standart sapma değeri 115.169' dur. Bu değerlerin oldukça büyük olması dağılımın geniş olduğunu gösterir.

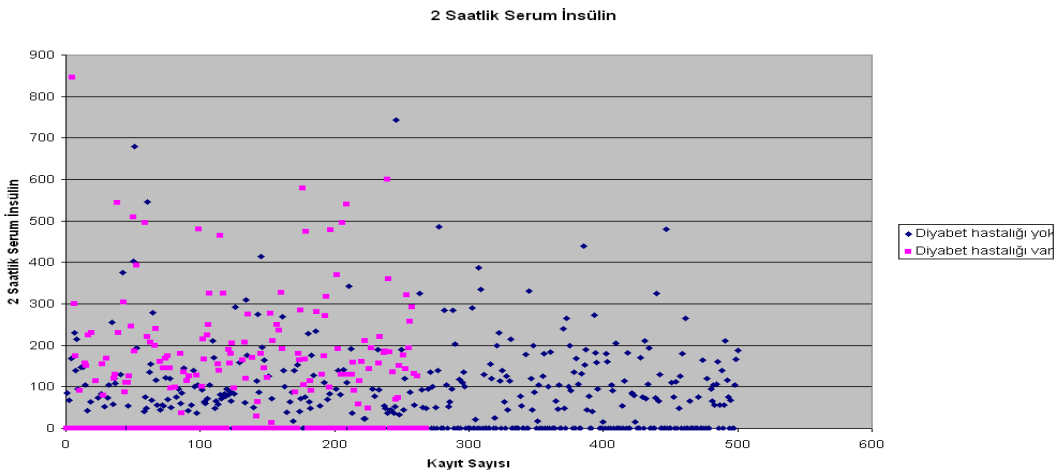
Ailedeki şeker hastalığı oranının varyans değeri 0.109 ve standart sapma değeri 0.331'dir ve diğer niteliklerin varyans ve standart sapma değerinden daha düşüktür. Bu değerlerin küçük olması dağılımın küçük olduğunu göstermektedir. 2 saatlik serum insülin niteliklerinin değerleri ortalama sapmadan diğer nitelik değerlerine göre daha büyük olsa bile yine de üst üste binen veri noktaları vardır. Ayrıca beklenen iki sınıf içinde yer alacak olan veri noktalarının değerleri birbirine çok yakındır veya aynıdır. Diyabet veritabanına ait olan sınıflarına ayırt edilmesinde bu değerler yeterince etkili değildir. Çünkü sınıfların ayırt edilmesinde kullanılacak olan niteliklerin değerleri birbirine çok yakın ve ayır edici değildir.

Tablo 5.7: Diyabet veritabanına ait niteliklerin istatistiksel analiz değerleri.

Nitelikler	**Minimum Değer**	**Maksimum Değer**	**Ortalama Değer**	**Varyans Değeri**	**Standart Sapma Değeri**
Hamilelik Sayısı	0	17	3.845	11.339	3.367
Plazma Glukoz Değerleri	0	199	120.895	1020.917	31.952
Diyastolik Kan Basıncı	0	122	69.106	374.159	19.343
Tricep Deri Katlanma Kalınlığı	0	99	20.537	254.142	15.942
2 Saatlik Serum İnsülin	0	846	79.799	13263.887	115.169
Vücut Kitle Oranı	0	67.1	31.993	62.079	7.879
Ailedeki Şeker Hastalığı Fonksiyonu	0.078	2.42	0.472	0.1096	0.3311
Yaş	21	81	33.241	138.123	11.753

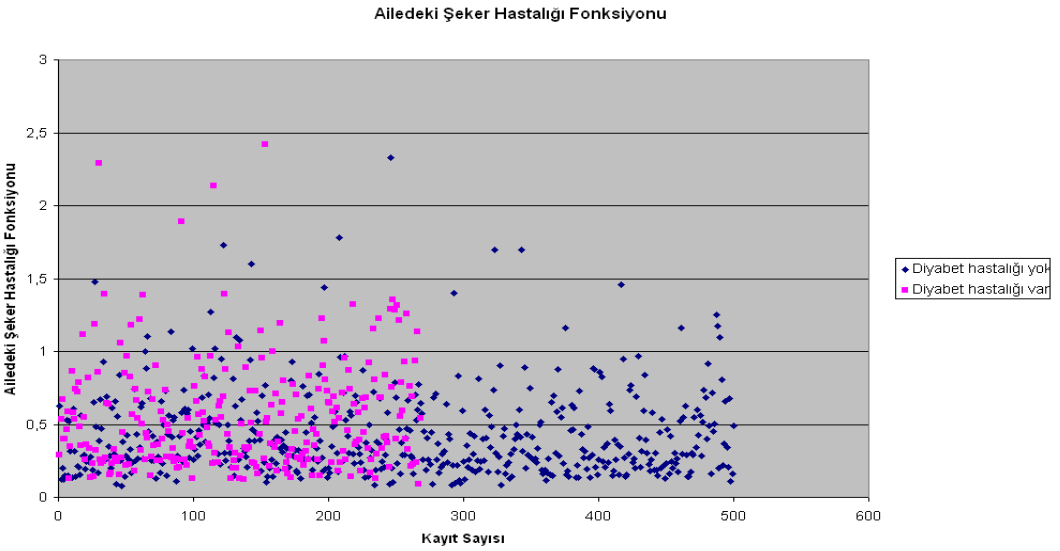
Diyabet hastalığının olup olmama durumunun 2 saatlik serum insülin ve ailedeki şeker hastalığı fonksiyonu nitelik değerleri için şekilsel olarak karşılaştırılması Şekil 5.9 ve Şekil 5.10' de yapılmıştır. Şekil 5.9' da 2 saatlik serum insülin nitelik değerleri diyabet hastalığının olup olmama durumunda birbirine yakın değerlere sahiptir. 2 saatlik serum insülin niteliğin varyans değeri ve standart sapma değeri

diğer nitelik değerlerine göre oldukça büyüktür. Buna rağmen diyabet hastalığının olup olmama durumunu ayırt etmede bu nitelik değeri etkin bir role sahip değildir.



Şekil 5.9: Diyabet hastalığının olup olmama durumuna etki eden 2 saatlik serum insülin niteliğine ait olan değerler.

Şekil 5.10' da ailedeki şeker hastalığı fonksiyonu nitelik ait olan değerler diyabet hastalığının olup olmama durumunda birbirine yakın değerlere sahiptir. Bu niteliğe ait olan varyans ve standart sapma değerleri de diğer niteliklere göre daha küçüktür. Bu nedenle ailedeki şeker hastalığı fonksiyonu niteliği diyabet hastalığının olup olmama durumunu ayırt etmede etkin bir role sahip değildir.



Şekil 5.10: Diyabet hastalığının olup olmama durumuna etki eden ailedeki şeker hastalığı fonksiyonu niteliğine ait olan değerler.

5.2.4 Mamografi Veritabanı

UCI veri deposundan alınan mamografi veritabanı incelemeden önce mamografi hakkında genel bilgiler vermemiz gereklidir. Eksik yanlarının olmasına rağmen mamografi, düşük doz radyasyon kullanılarak meme kanseri tanısında etkili olan en iyi görüntüleme yöntemidir. Mamografi tarama ve tanısal amaçlı olarak uygulanmaktadır. Mamografinin tarama amaçlı olarak uygulanması meme hastalıklarının erken teşhis edilmesinde gereklidir. Klinik şikâyeti olan hastaların doktorun muayenesi sırasında ele gelen lezyonlarının karakterlerinin saptanması için mamografinin tanısal amaçlı olarak uygulanması gereklidir. Mamografi tanı konulan hastalarda tedavi planlaması için ve tedavi sonrası takip içinde kullanılmaktadır. Mamografi' nin kanser tanısında duyarlılığı %83' tür. Mamografide kanser tanısı uyandıran lezyonlara yapılan biyopsilerin %14–36' sı pozitif çıkmaktadır. Mamografi değerlendirilmesi ile yapılan biyopsilerin, pozitif tahmin değerlerinin düşük çıkması %70 dolayında gereksiz biyopsinin(mamografide kötü huylu olduğu düşünülen tümörlerin iyi olduğunun açığa çıkması gibi) yapılmasına neden olmuştur. Yüksek rakamlardaki gereksiz meme biyopsilerinin sayısını azaltmak için bilgisayar yardımlı tanı sistemlerine başvuruldu. Bu sistemler mamografi ile elde edilen şüpheli lezyonlarda biyopsi yapma kararını verme de veya biyopsi yapmak yerine kitlenin takibi kararını vermede yardımcı olmuştur. Uygulamada kullanılacak olan mamografi veritabanı 961 kayıt ve 5 tane nitelik değeri içermektedir. Mamografi veritabanına ait olan değerler Tablo 5.8' de gösterilmiştir.

Tablo 5.8: Mamografi veritabanı içindeki nitelik değerleri.

Radyolog Değerlendirmesi	Hastanın Yaşı	Kitlenin Şekli	Kitlenin Sınırları	Kitledeki Yoğunluk	Önem Derecesi
5	67	3	5	3	1
4	43	1	1	5	1
5	58	4	5	3	1
4	28	1	1	3	0
5	74	1	5	5	1
4	65	1	6	3	0
4	70	5	6	3	0
5	42	1	6	3	0

Mamografi veritabanı UCI veri deposundan metin dosyası formatın alınmış uygulamada kullanılmak üzere Access veritabanına aktarılmıştır. Mamografi veritabanındaki nitelikler sırası ile radyolog değerlendirmesi, hastanın yaşı, kitlenin

şekli (shape), kitlenin sınırları (margin) ve kitledeki yoğunluktur (density). Mamografi veritabanındaki radyolog değerlendirmesi 1 ile 5 arasında eğer almaktadır. Bu değerler radyologun ilgili örneği değerlendirme sonucunu ifade etmektedir (Örneğin 1 değerinin kesinlikle iyi huylu, 5 değerinin büyük olasılıkla kötü huylu kanser olması gibi). Hastanın yaşı niteliği hastanın yaşını ifade etmektedir. Kitlenin şekli 1 ile 4 arasında değer almaktadır. Kitlenin şekli niteliğinin 1 olması yuvarlak, 2 olması oval, 3 olması yuvarlak çıkıntılı (lobular) ve 4 olması ise düzensiz bir kitle şekli olduğunu gösterir. Kitle sınırları niteliği 1 ile 5 arasında değer almaktadır. Kitle sınırları niteliğinin değerinin 1 olması yuvarlak, 2 olması mikrolobullu, 3 olması belirsiz, 4 olması tam tanımlanamayan ve 5 olması ise sivri uçlu çıkıntı gösteren bir kitle sınırına sahip olduğunu gösterir. Kitle yoğunluk niteliği değeri 1 ile 4 arasında değer alır. Kitle yoğunluk değerinin 1 olması yüksek, 2 olması eşit, 3 olması az ve 4 olması ise yağ içeren yoğunluk değerine sahip olduğunu gösterir. Bu veritabanı hastanın yaşı ve radyolog değerlendirmesinden incelenen kitlenin ciddiyetini(iyi huylu veya kötü huylu) tahmin etmede kullanılabilir. Her bir örneğe 1 (kesinlikle iyi huylu) ile 5 (yüksek olasılıkla kötü huylu) arasında değişen radyolog değerlendirmesi doktorlar tarafından atanır.

Mamografi veritabanı belirtilen nitelik değerleri göz alındığında 2 sınıfa ayrılmıştır. Hastalık önem derecesinin 0 olması iyi huylu, 1 olması kötü huylu bir meme kanseri olduğunu belirtir. Birinci sınıfta 516, ikinci sınıfta 445 örnek vardır. Birinci sınıf iyi huylu meme kanseri saptanan örnekleri ifade ederken, ikinci sınıf kötü huylu meme kanseri saptanan örnekleri ifade etmektedir.

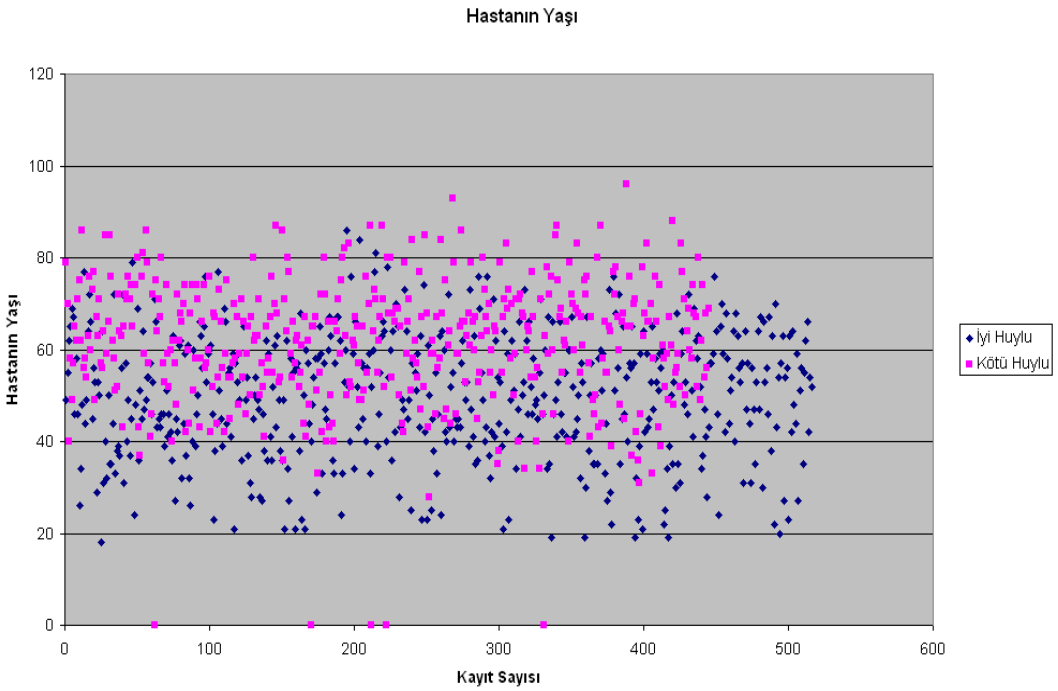
Mamografi veritabanındaki nitelikler ve niteliklere ait olan istatistiksel değerler Tablo 5.9' de gösterilmiştir. Tablo 5.9' te görüldüğü gibi hastanın yaşı niteliğinin varyans değeri ve standart sapma değeri diğer niteliklerin değerlerine göre oldukça yüksektir. Hastanın yaşı varyans değeri 224.301 ve standart sapma değeri 14.977' dur. Bu değerlerin oldukça büyük olması dağılımın geniş olduğunu gösterir. Bi-rads değerlendirmesinin varyans değeri 0.508 ve standart sapma değeri 0.713'dür ve diğer niteliklerin varyans ve standart sapma değerinden daha düşüktür. Bu değerlerin küçük olması dağılımın küçük olduğunu göstermektedir. Hastanın yaşı niteliğinin değerleri ortalama sapmadan diğer nitelik değerlerine göre daha büyük olsa bile yine

de üst üste binen veri noktaları vardır. Ayrıca beklenen iki sınıf içinde yer alacak olan veri noktalarının değerleri birbirine çok yakındır veya aynıdır. Mamografi veritabanına ait olan sınıflarının ayırt edilmesinde bu değerler yeterince etkili değildir. Çünkü sınıfların ayırt edilmesinde kullanılacak olan niteliklerin değerleri birbirine çok yakın ve ayır edici değildir.

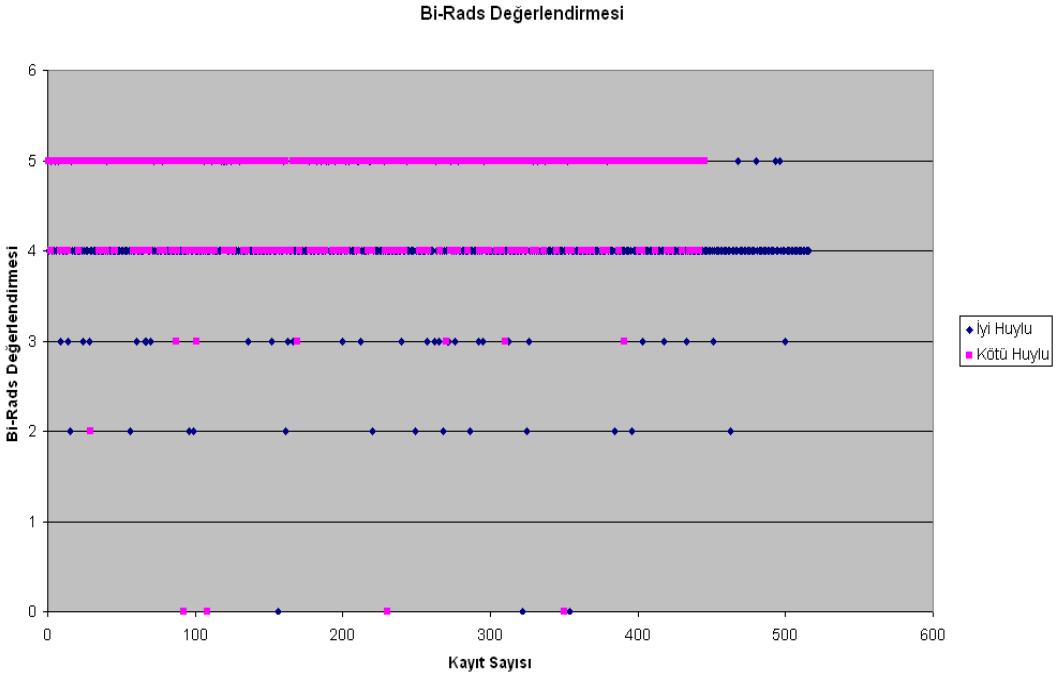
Tablo 5.9: Mamografi veritabanına ait niteliklerin istatistiksel analiz değerleri.

Nitelikler	**Minimum Değer**	**Maksimum Değer**	**Ortalama Değer**	**Varyans Değeri**	**Standart Sapma Değeri**
Bi-Rads Değerlendirmesi	0	5	4.276	0.508	0.713
Hastanın Yaşı	0	96	55.199	224.301	14.977
Kitlenin Şekli	0	4	2.634	1.724	1.313
Kitlenin Sınırları	0	5	2.657	2.699	1.643
Kitledeki Yoğunluk	0	4	2.681	0.750	0.866

Mamografi veritabanındaki veriler doğrultusunda iyi huylu ve kötü huylu bir kitlenin varlığının teşhisinde hastanın yaşı ve bi-rads değerlendirmesi nitelik değerlerinin şekilsel olarak karşılaştırılması Şekil 5.11 ve Şekil 5.12' de yapılmıştır. Şekil 5.11' da hastanın yaşı nitelik değerleri incelendiğinde kötü huylu bir kitlenin olma olasılığının yaşın artmasıyla daha da arttığı görülmüştür. 40–60 yaş arasında kitlenin iyi huylu ya da kötü huylu çıkma olasılığının birbirine yakın olduğu görülmektedir. 20–40 yaş arasında tek tük kötü huylu kitle çıkma olasılığına karşın genelde iyi huylu kitlenin çıktığı görülmektedir. Şekil 5.12' deki bi-rads değerlendirmesi değerleri incelendiğinde 5 ve 4 bi-rads değerlendirmesi değerlerinde kitlenin iyi huylu ve kötü huylu çıkma olasılığının birbirine yakın olduğu görülmektedir. Bi-rads değerlendirmesinin 2 ve 3 olduğu durumlarda ise genelde iyi huylu kitlenin çıkma olasılığının diğerine göre daha yüksek olduğu görülmektedir.



Şekil 5.11: Mamografi veritabanındaki veriler doğrultusunda bir kitlenin iyi huylu ve kötü huylu olup olmama durumuna etki eden hastanın yaşı niteliğine ait olan değerler.



Şekil 5.12: Mamografi veritabanındaki veriler doğrultusunda bir kitlenin iyi huylu ve kötü huylu olup olmama durumuna etki eden bi-rads değerlendirmesi niteliğine ait olan değerler.

5.3 Geliştirilen Uygulama ile Verilerin Analizi

Yapılan uygulama Pentium 4, 2.8 GHz işlemci ve 1 GB Ram' e sahip olan masa üstü bilgisayarda gerçekleştirilmiştir. Gerçekleştirilen uygulama Borland Delphi 6.0 kullanılarak Windows XP işletim sistemi üzerinde geliştirilmiştir. Veriler Access veritabanında tutulmuştur. Merkez tabanlı kümeleme algoritmaları da Delphi ile kodlanmıştır.

Uygulama aracılığı ile merkez tabanlı kümeleme algoritmaları yedi farklı açıdan birbirleri ile karşılaştırılmıştır. Bu 6 farklı nokta aşağıda belirtilmiştir:

1) Başlangıç Durumuna Duyarlılık: Merkez tabanlı kümeleme algoritmaları başlangıç koşullarına duyarlı olduğundan algoritmalar bu kıstas temel alınarak karşılaştırılmıştır. Uygulamada karşılaştırma amaçlı olarak 3 farklı başlangıç yöntemi kullanılmıştır. Bu başlangıç yöntemleri MacQueen yöntemi, rasgele (Forgy) yöntemi ve rasgele bölümleme (random partition) yöntemidir. Bu yöntemler en popüler başlangıç koşulu oluşturma yöntemleridir. Bu yöntemler kullanılarak algoritmalar performans ve işlemci zamanı bakımından karşılaştırılmıştır.

2) K Küme Sayısının Kümelemeye Etkisi: Merkez tabanlı kümeleme algoritmalarının hepsi başlangıçta belirlenen k sayısına ihtiyaç duyar. k sayısı oluşturulacak olan küme sayısını ifade eder. Bu uygulama ile k sayısının değişiminden merkez tabanlı kümeleme algoritmalarının kümeleme sonuçlarının nasıl etkilendiği araştırılmıştır.

3) Verinin Boyutunun Az ya da Çok Olması: Bazı algoritmalar verinin boyutunun az veya çok olması durumuna göre farklı sonuçlar üretebilmektedir. Bu uygulama ile merkez tabalı kümeleme algoritmalarının verinin boyutuna bağlı olarak nasıl sonuçlar verdikleri analiz edilmiştir.

4) Aykırı Değerlerin Kümelemeye Etkisi: Uygulama ile merkez tabanlı kümeleme algoritmalarının aykırı değerler karşısında sergiledikleri davranışları incelenmiş ve birbirleri ile karşılaştırılmıştır.

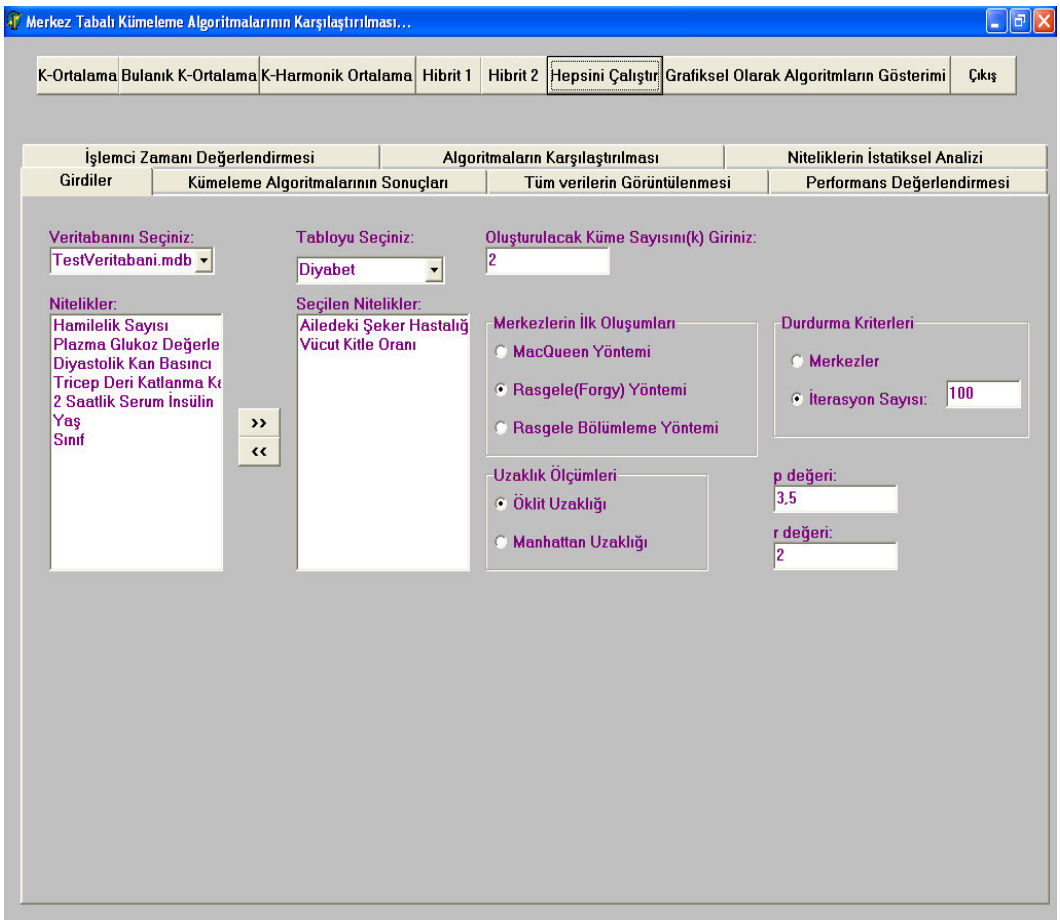
5) Algoritmaların Toplam Karesel Hata Değerleri ve İşlemci Zamanına Göre Karşılaştırılması: Bir veri kümesi üzerinde merkez tabanlı kümeleme algoritmaları toplam karesel hata ve işlemci zamanı değerlerine göre karşılaştırılmışlardır.

6) Algoritmaların Yakınsama Durumuna Göre Karşılaştırılması: Bir veri kümesi üzerinde merkez tabanlı kümeleme algoritmaları yakınsama durumuna göre karşılaştırılmıştır. En uygun toplam karesel hata değerine algoritmaların hangisinin daha hızlı yakınsadığını tespit etmek amacıyla bu karşılaştırma yapılmıştır. Bu karşılaştırma işlemi için uygulama arayüzünündeki girdiler sekmesindeki durdurma kıstaslarından merkezler durdurma kıstası seçilmiştir.

5.4 Uygulamaya Ait Arayüzler ile İlgili Açıklamalar

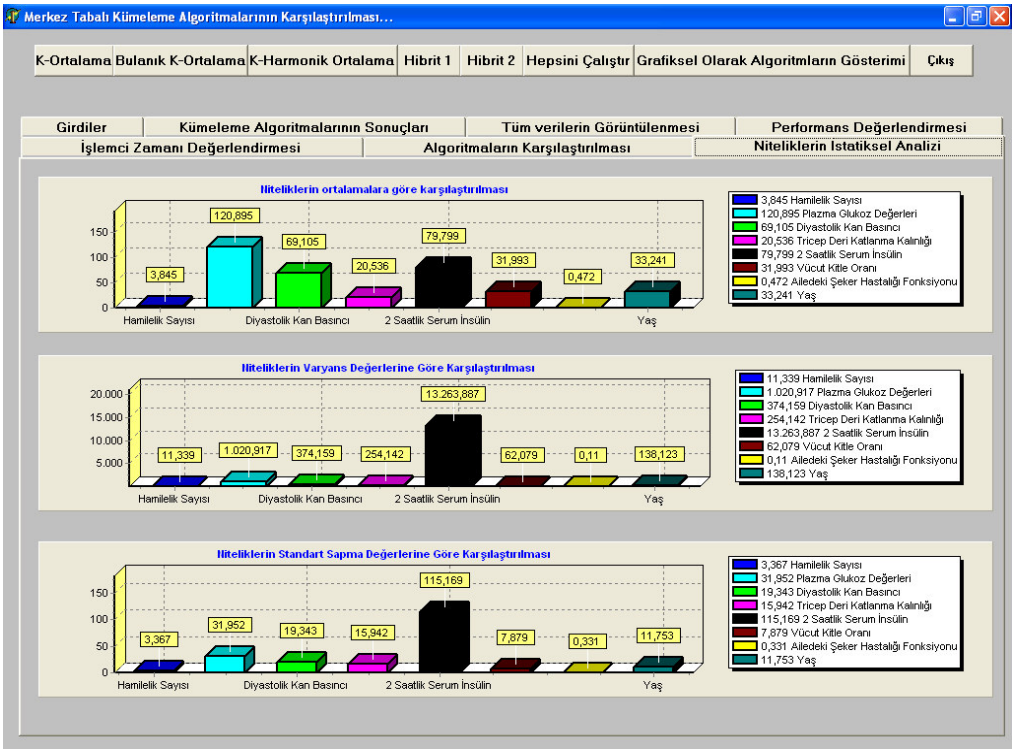
Uygulamaya ait olan ana arayüzdeki girdiler sekmesi, merkez tabanlı kümeleme algoritmalarının çalıştırılmasından önce girilmesi gereken bilgilerin bulunduğu sekmedir. Girdiler sekmesinde ilk önce üzerinde işlem yapılacak olan veritabanı seçilir. Daha sonra seçilen veritabanı içindeki tablolardan seçim yapılır. Buradaki tabloların her biri UCI veri deposundan alınan veritabanlarına denk düşmektedir. Seçilen tablonun içeriğindeki nitelikler yani sütunlar ana arayüzdeki nitelikler kısmına dolar. Ardından bu niteliklerden seçilenler, seçilen nitelikler kısmına aktarılır. Girdiler sekmesinde kullanıcı tarafından oluşturulacak olan küme sayısını ifade eden k sayısının girilmesi gerekmektedir. Bu k sayısı 2 ile 20 arasında bir değer almaktadır. Girdiler sekmesinde ilk merkezlerin nasıl oluşturulacağının belirlenmesi için kullanıcıya 3 seçenek sunulmuştur. Bunlardan Macqueen yöntemi seçildiğinde ilk merkezler, veri kümesinin baştan ilk k adet elemanının merkez olarak seçilmesi ile belirlenir. İkinci seçenek olan rasgele yöntemi seçilirse ilk merkezler veri kümesinin rasgele seçilmiş k adet elemanı olmaktadır. Bir diğer seçenek olan rasgele bölümleme yöntemi seçilmiş ise, veri kümesi k adet parçaya ayrılır ve bu parçaların her biri rasgele seçilmiş olan k adet merkezinden biri ile ilişkilendirilir ve her bir merkez kendisi ile ilişkili olan parça içindeki veri noktalarının aritmetik ortalaması alınarak hesaplanır. Girdiler sekmesindeki uzaklık ölçümlerinin seçimi kullanıcıya bırakılmıştır. Kulacı veri noktaları ve merkez noktaları arasındaki uzaklığı bulmada Öklit ya da Manhattan uzaklık ölçümünden yararlanabilmektedir. Belirlenen girdiler

doğrultusunda çalıştırılacak olan merkez tabanlı kümeleme algoritmalarının işleyişinin durdurulabilmesi için kullanıcı seçimine bağlı olan iki durdurma kıstası verilmiştir. Bu durdurma kıstasları; o anki iterasyondaki merkez ile bir önceki iterasyondaki merkezin eşit olması ya da iterasyon sayısının kullanıcı tarafından belirlenen sınıra ulaşması şeklindedir. Ayrıca bulanık k-ortalama algoritmasını için kullanıcı tarafın r değerinin, k-harmonik ortalama, hibrit 1 ve hibrit2 algoritmaları için de p değerinin girilmesi gerekmektedir. Girdiler sekmesine gerekli bilgiler girildikten sonra merkez tabanlı kümeleme algoritmaları olan k-ortalama, bulanık k-ortalama, k-harmonik ortalama, hibrit 1 ve hibrit 2 tuşlarına basılarak algoritmalar çalıştırılabilir. Ayrıca algoritmaları ayrı ayrı çalıştırmak yerine "Hepsini çalıştır" tuşuna basılarak tüm algoritmalar tek bir tuşla da çalıştırılabilir. Uygulamaya ait ana arayüz Şekil 5.13' te verilmiştir.



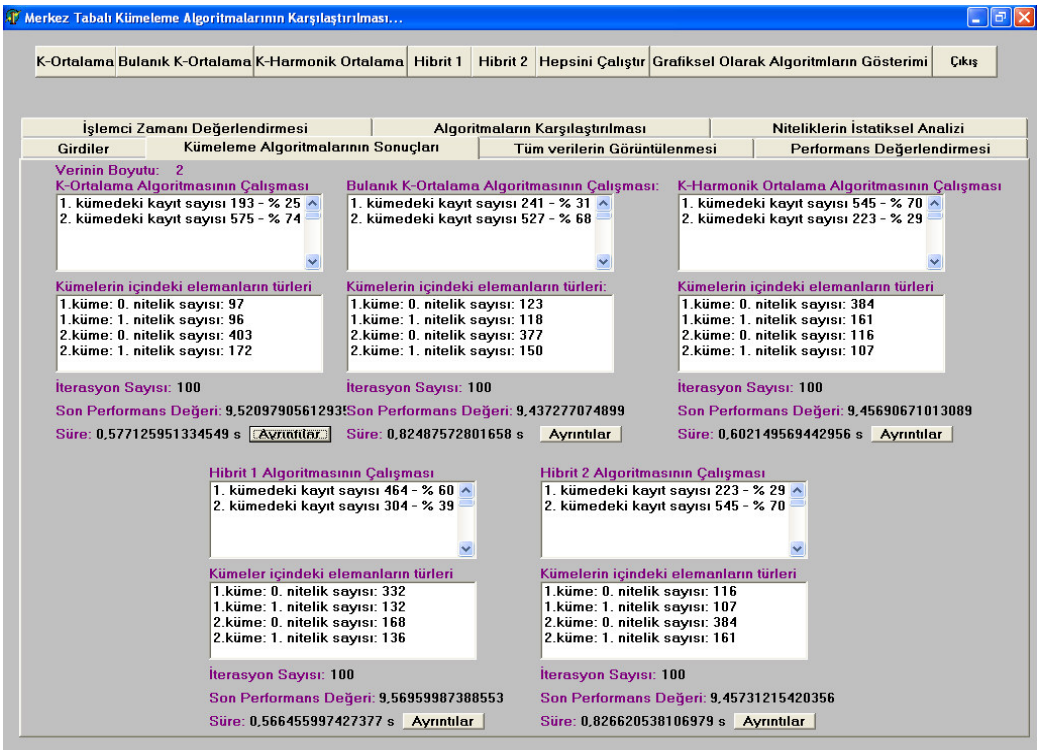
Şekil 5.13: Uygulamaya ait olan ana arayüz.

Merkez tabanlı kümeleme algoritmalarını ilgili veritabanı üzerinde çalıştırmadan önce kullanılacak niteliklere karar vermek için "Niteliklerin istatistiksel analizi" sekmesini tıklayıp incelemek gerekmektedir. Şekil 5.14' te görüldüğü gibi diyabet veritabanı içindeki niteliklerin ortalama, varyans ve standart sapma değerleri karşılaştırmalı olarak verilmiştir. Bu nitelik değerleri bize dağılımın yaygınlığı hakkında bilgi vermektedir.



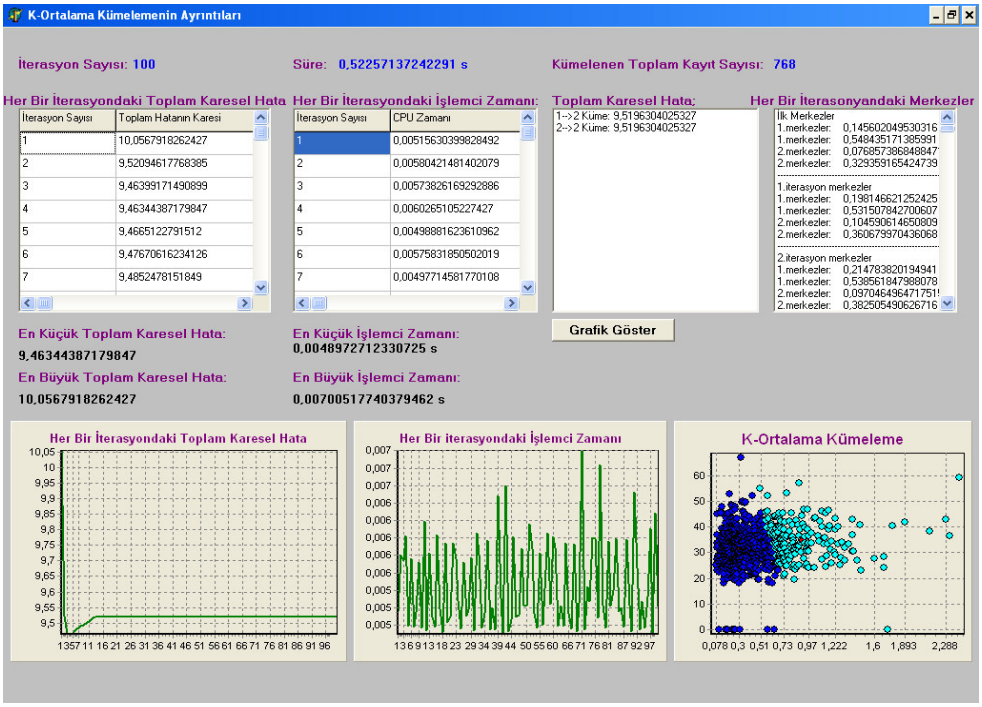
Şekil 5.14: Niteliklerin istatistiksel analizi.

Girdiler sekmesine girilen bilgiler doğrultusunda merkez tabanlı kümeleme algoritmalarının adları ile başlayan tuşların her birine basılması ile ya da "hepsini çalıştır" tuşuyla algoritmaların çalıştırılmasıyla elde edilen kümeleme sonuçları "Kümeleme Algoritmalarının Sonuçları" sekmesinden takip edilebilir. Bu sekmede her bir algoritmanın çalıştırılmasından sonra elde elden sonuçlara ilişkin bilgiler yer almaktadır. Her bir algoritmanın adının altında oluşan kümeler, kümelerde bulunan veri noktalarının sayısı, algoritmanın kümeleme işlemini gerçekleştirdiği iterasyon sayısı, algoritmanın yaptığı kümelemeye ilişkin son performans değeri ve algoritmanın kümeleme işlemini gerçekleştirdiği toplam süre bilgileri yer almaktadır. Şekil 5.15' da kümeleme sonuçlarının gösterildiği arayüz görülmektedir.



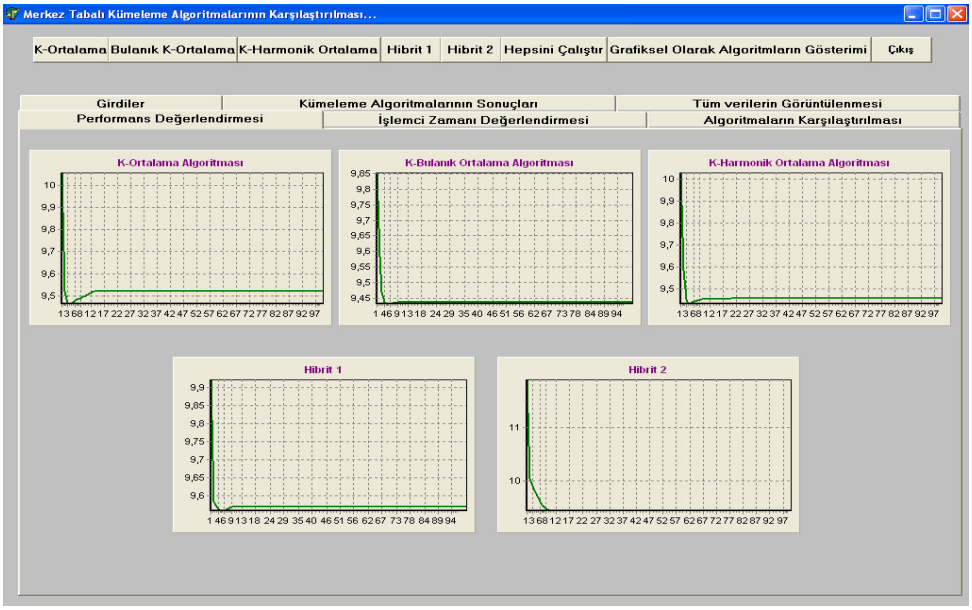
Şekil 5.15: Kümeleme sonuçları arayüzü.

Kümeleme sonuçları arayüzündeki her bir algoritmanın altında bulunan ayrıntılar butonuna basılmasıyla, yapılan kümelemeye ilişkin daha ayrıntılı sonuçlar görülebilmektedir. Ayrıntılar butonuna basılmasıyla elde edilen arayüzde ilgili algoritmanın yaptığı kümelemeye ilişkin iterasyon sayısı, toplam süre ve kümelenen toplam kayıt sayısı bilgileri görülebilmektedir. Ayrıca her bir iterasyondaki toplam karesel hata değerlerine, her bir iterasyondaki milisaniye cinsindeki işlemci zamanı değerlerine, algoritmanın sürekli çalıştırıldığında elde edilen toplam karesel hata değerlerine, her bir iterasyondaki merkez noktaları değerlerine bu ekran üzerinden ulaşılabilmektedir. Aynı ekranda her bir iterasyondaki toplam karesel hata değerlerinin ve işlemci zamanı değerlerinin değişimi grafikler üzerinden takip edilebilmektedir. Yapılan kümeleme işlemi sonucu oluşan kümelerdeki dağılım da veri kümesinin iki boyutlu olması durumunda ekrandaki üçüncü grafik üzerinden takip edilebilmektedir. Kümeleme sonucu oluşan her bir küme ayrı bir renk ile gösterilerek birbirinden ayrıştırılır. Merkez tabanlı kümeleme algoritmalarından biri olan k-ortalama kümeleme algoritmasının kümeleme sonuçları Şekil 5.16' de ayrıntılı bir şekilde yer almaktadır. Her bir merkez tabanlı kümeleme algoritmalarının kümeleme sonuçlarının takibi için aşağıdaki gibi bir arayüz vardır.



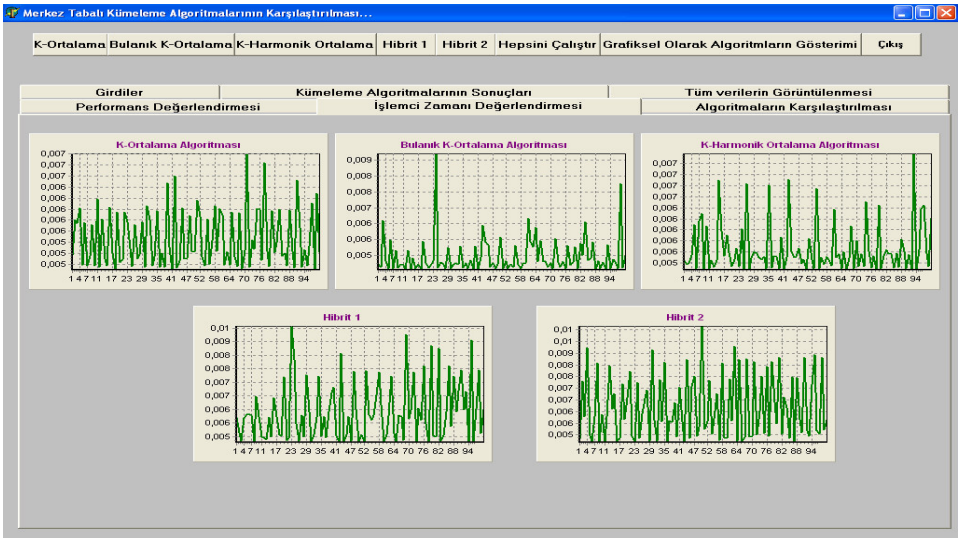
Şekil 5.16: Merkez tabanlı kümeleme algoritmalarından bir olan k-ortalama algoritmasına ilişkin ayrıntılı kümeleme sonuçları.

Ana arayüz üzerindeki "Performans Değerlendirmesi" sekmesinde işleme tabi tutulan tüm merkez tabanlı kümeleme algoritmaları performans açısından karşılaştırılır. Merkez tabanlı kümeleme algoritmalarının her birinin kendine ait bir amaç fonksiyonu vardır. Her bir algoritma kendi amaç fonksiyonu temel alınarak birbiri ile karşılaştırılamaz. Bu nedenle ilgili algoritmaların karşılaştırılmasında kullanılacak ortak bir amaç fonksiyonuna ihtiyaç vardır. Bu amaç fonksiyonun değerlerine göre algoritmalar birbirleri ile performans açısından karşılaştırılırlar. Bu uygulamada, K-ortalama algoritmasının amaç fonksiyonu bilindik ve uygulanması daha kolay olduğundan, algoritmaların performanslarının karşılaştırılmasında K-ortalama algoritmasının amaç fonksiyonunun karekökü kullanılmıştır. Şekil 5.17' de merkez tabanlı kümeleme algoritmaları grafikler kullanılarak performans açısından karşılaştırılmıştır.



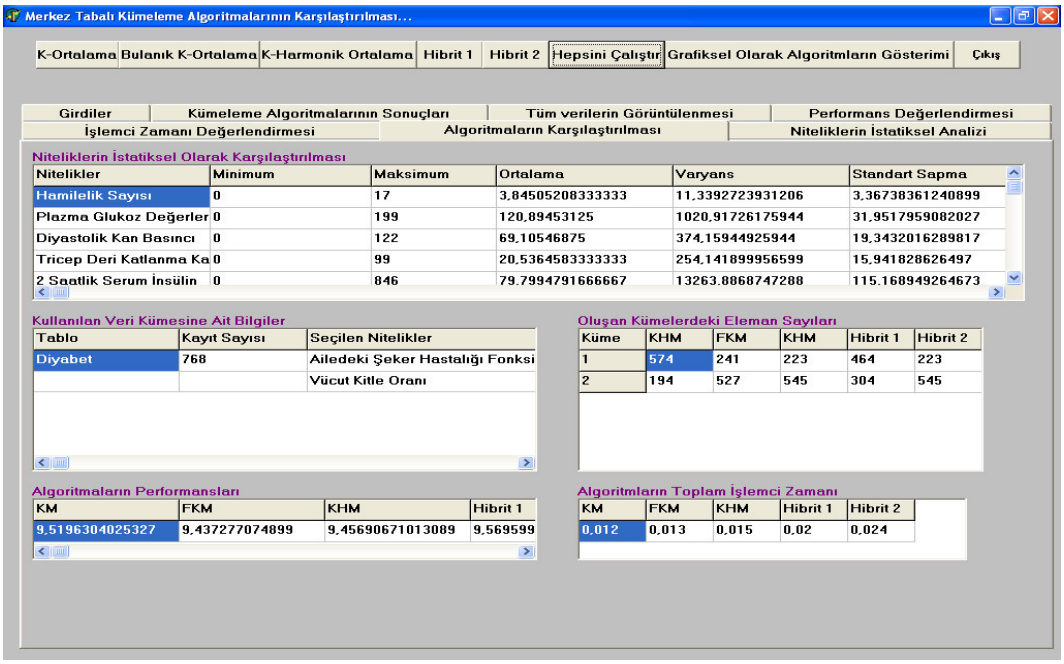
Şekil 5.17: Merkez tabanlı kümeleme algoritmaların performans değerlerine göre karşılaştırılması.

Ana arayüz üzerindeki "İşlemci Zamanı Değerlendirmesi" sekmesinde, işleme tabi tutulan tüm merkez tabanlı kümeleme algoritmalarının işlemci zamanına göre karşılaştırma sonuçları grafikler ile gösterilmektedir. Şekil 5.18' da merkez tabanlı kümeleme algoritmalarının işlemci zamanına göre karşılaştırılma sonuçları görülebilmektedir. Karşılaştırmaların grafikler aracılığı ile yapılması hangi algoritmanın daha iyi olduğu konusunda kullanıcıya yardımcı olmaktadır.



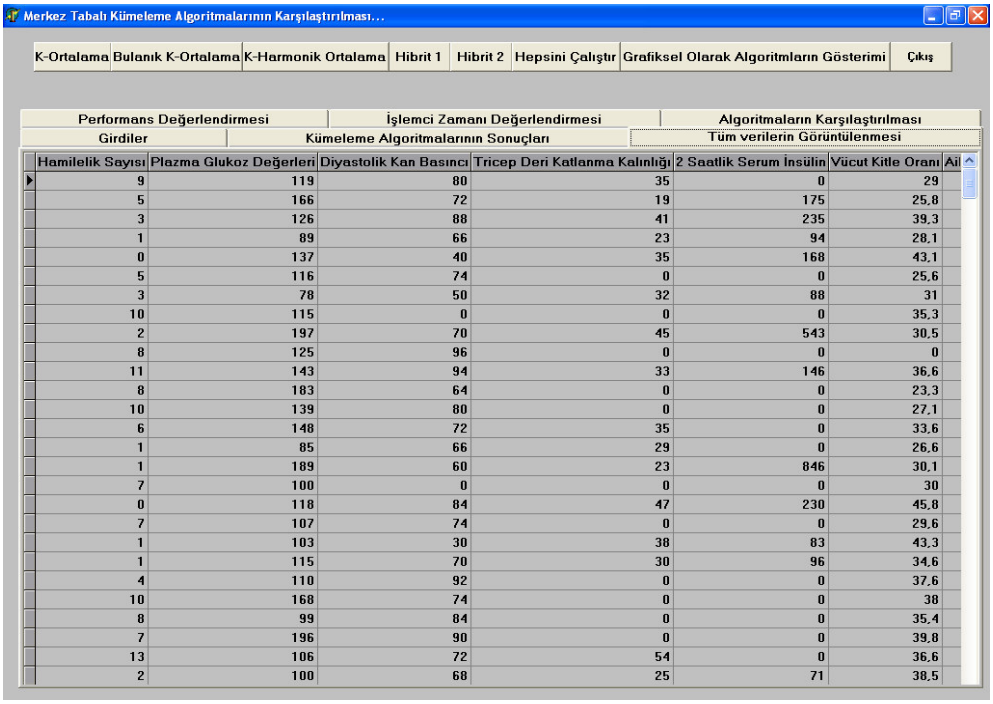
Şekil 5.18: Merkez tabanlı kümeleme algoritmaların işlemci zamanı değerlerine göre karşılaştırılması.

Ana arayüzdeki sekmelerden biri olan "Algoritmaların Karşılaştırılması" sekmesinde işleme tabi tutulan algoritmalar oluşan kümelerdeki eleman sayıları, son performans değerleri, toplam işlemci zamanlarının tüm işlemci zamanlarına oranı ve küme sayısının geçerliliğine göre karşılaştırılmaktadır. Küme sayısının geçerliliğinin karşılaştırılmasında kullanılan birçok teknik vardır. Bu teknikler kümelemenin temel mantığına dayalı olduğundan uygulamada çalıştırılan algoritmaların küme sayısının geçerliliği bu tekniklere göre karşılaştırılmamıştır. Bunun yerine algoritmalar kümelemenin temel mantığına göre karşılaştırılmıştır. Her veri noktasının kendisine en yakın merkeze uzaklıklarının toplamının, merkezler arasındaki minimum uzaklıkların toplamına bölünmesiyle elde edilen sonuçlara göre algoritmalar karşılaştırılmıştır. Bu küme içi benzerliğin maksimum, kümeler arası benzerliğin minimum olması mantığa dayalı olarak yapılan bir karşılaştırmadır. Bu arayüzde ayrıca kümeleme işlemine tabi tutulan veri kümesi, veri kümesinin eleman sayısı ve kullanılan nitelik değerleri ayrıntılı bir şekilde gösterilmektedir. Bu arayüzde ilgili veritabanı içindeki nitelik değerlerinin her birine ait olan minimum, maksimum, ortalama, varyans ve standart sapma değerleri de verilmekte ve niteliklerin istatistiksel olarak karşılaştırılmaktadır. Şekil 5.19' da algoritmaların yukarıda anlatılan kıstaslarına göre karşılaştırma sonuçları ayrıntılı bir şekilde görülmektedir.



Şekil 5.19: Merkez tabanlı kümeleme algoritmalarının, oluşan kümelerdeki eleman sayıları, son performans değerleri ve işlemci zamanına göre karşılaştırılması.

İşleme tabi tutulan veri kümesine "Tüm Verilerin Görüntülenmesi" sekmesinden ulaşılabilmektedir. Arayüz aracılığı ile tüm verilerin görüntülenmesi kullanıcıya karşılaştırma yaparken verileri inceleme fırsatı sunmaktadır. Şekil 5.20' de kümeleme işlemine tabi tutulan veri kümesine ilişkin veriler görüntülenmektedir.



Hamilelik Sayısı	Plazma Glukoz Değerleri	Diyastolik Kan Basıncı	Tricep Deri Katlanma Kalınlığı	2 Saatlik Serum İnsülin	Vücut Kitle Oranı	Aİ
9	119	80	35	0	29	
5	166	72	19	175	25,8	
3	126	88	41	235	39,3	
1	89	66	23	94	28,1	
0	137	40	35	168	43,1	
5	116	74	0	0	25,6	
3	78	50	32	88	31	
10	115	0	0	0	35,3	
2	197	70	45	543	30,5	
8	125	96	0	0	0	
11	143	94	33	146	36,6	
8	183	64	0	0	23,3	
10	139	80	0	0	27,1	
6	148	72	35	0	33,6	
1	85	66	29	0	26,6	
1	189	60	23	846	30,1	
7	100	0	0	0	30	
0	118	84	47	230	45,8	
7	107	74	0	0	29,6	
1	103	30	38	83	43,3	
1	115	70	30	96	34,6	
4	110	92	0	0	37,6	
10	168	74	0	0	38	
8	99	84	0	0	35,4	
7	196	90	0	0	39,8	
13	106	72	54	0	36,6	
2	100	68	25	71	38,5	

Şekil 5.20: Tüm verilerin görüntülendiği arayüz.

Her algoritmanın çalıştırılması sırasında hesaplanan bilgiler kendi adını taşıyan bir metin dosyasına yazdırılmaktadır. Bu dosyada ekranda görüntülenen rakamsal bilgiler ve bunlarla ilişkili diğer bilgiler yer almaktadır. Dosyalar aracılığı ile algoritmaların kümeleme işlemine ilişkin adımları takip edilebilmektedir.

5.5 Merkez tabanlı Kümeleme Algoritmalarının Karşılaştırılması

Kümeleme birçok alanda bir ön işlem olarak kullanılmaktadır. Kümeleme işlemine başlanmadan önce kullanılacak olan algoritmanın özelliklerinin çok iyi bilinmesi ve uygulanacak veriye uygun olup olmadığının kararının uzman tarafından verilmesi gerekmektedir. Uygun olmayan kümeleme algoritmaları ile elde edilen sonuçlar uzmanın işini yaramaz. Bu nedenlerden dolayı kümeleme algoritmalarının popüler bir sınıfı olan merkez tabanlı kümeleme algoritmalarının karşılaştırılması üzerine bir

tez çalışması yapılmıştır. Bu yapılan çalışma ile merkez tabanlı kümeleme algoritmalarının davranışlarının iyi bir şekilde analiz edilmesi ve bu algoritmaları kullanacak olan uzmanların algoritmaların avantajlarını ve dezavantajlarını bilerek bu algoritmaları tercih etmesi amaçlanmıştır. Merkez tabanlı kümele algoritmalarının işleyişine dair çok az Türkçe doküman bulunmaktadır. Bu tez çalışması ile doküman eksiliği büyük ölçüde giderilmiştir.

Uygulamada merkez tabanlı kümeleme algoritmalarından KM, FKM, KHM, Hibrit1 ve Hibrit 2 algoritmaları belirlenen kıstaslar doğrultusunda karşılaştırılmıştır. Bu kıstaslar ve ilgili kıstasa ilişkin karşılaştırmalar aşağıda sırayla verilmiştir.

1) Başlangıç Durumuna Duyarlılık: Kümeleme algoritmalarından bazıları başlangıç koşullarına duyarlı olduğundan belirtilen merkez tabanlı kümeleme algoritmaları belirlenen üç başlangıç koşulunu temel alarak karşılaştırılmıştır. Bu karşılaştırma ile hangi algoritmanın başlangıçtaki koşullar karşısında daha az etkilendiği bulunmaya çalışılmıştır. Başlangıç koşullarından diğer merkez tabanlı kümeleme algoritmalarına göre daha az etkilenen algoritma bizim iyi bir algoritma demektir. Bu çalışma da merkez tabanlı kümeleme algoritmalarının başlangıç değerlerine duyarlı olup olmadığını analizi yapılırken 4 veritabanından yararlanılmıştır. Dört veritabanı üzerinde farklı başlangıç yöntemleri kullanılmış ve merkez tabanlı kümeleme algoritmalarının başlangıç değerleri karşısında nasıl davrandığı ele alınmıştır. Oluşan kümeler her bir algoritmanın kendi adını taşıyan metin dosyası incelenerek takip edilebilmektedir. Başlangıç değerlerine göre yapılan karşılaştırmalarda iterasyon sayısı 100, k değeri 3, FKM algoritması için r değeri 2 ve KHM, Hibrit 1, Hibrit 2 için p değeri ise 3.5 olarak alınmıştır.

Başlangıçtaki değerlere duyarlılık kıstasına ilişkin ilk karşılaştırma işlemi 150 kayıt ve 4 nitelik değerine sahip olan süsen çiçeği veritabanı üzerinde yapılmıştır. Bu veritabanına ait olan 4 nitelik kümeleme işlemine dahil olmuştur. Tablo 5.10' da süsen çiçeği veritabanı üzerinde üç başlangıç yöntemi temel alınarak merkez tabanlı kümeleme algoritmalarının performans ve kümeler içindeki niteliklerin eleman sayılarına göre karşılaştırmalar yapılmıştır. Yapılan karışlaştırma sonuçlara göre süsen çiçeği veritabanı üzerinde üç başlangıç yönteminden elde elden başlangıç

değerleri kullanıldığında KHM, Hibrit 2 ve FKM' nin diğerlerine göre daha başarılı olduğu görülmüştür. KHM algoritmasının üç başlangıç yöntemini kullanması sonucu elde edilen kümeler biribirinin aynısıdır. Hibrit 2 algoritmasının üç başlangıç yöntemi sonucu elde edilen merkez noktalarını kullanması sonucu elde edilen kümeler birbirinin aynısıdır. Aynı durum FKM algoritması içinde geçerlidir. Rasgele yöntemi sonucu elde edilen merkez noktalarının değişmesine karşın FKM algoritması sonucu oluşan kümeler, diğer yöntemler kullanılarak elde edilen kümelerin benzeri çıkmıştır. Üç algoritma başlangıç yöntemlerinden etkilenmemiştir. KM ve Hibrit 1 algoritması başlangıç yöntemleri sonucu elde edilen merkez noktalarından etkilenmişlerdir. KM algoritması ve Macqueen ve rasgele bölümleme yöntemlerinin kullanılması sonucu oluşan kümeler birbirine benzerdir. Rasgele yönteminin kullanılması sonucu oluşan kümelerden ise sadece biri, diğer yöntemlerle elde dilen kümelere benzerken, diğer iki küme ise Macqueen ve rasgele bölümleme yöntemi sonucu elde edilen kümelere benzememektedir. Hibrit 1 algoritması başlangıç seçilen yöntem sonucu elde edilen merkez noktalarına duyarlıdır. Hibrit 1 ve rasgele ve rasgele bölümleme yöntemlerinin kullanılması sonucu oluşan kümeler birbirine benzemektedir. Bu benzerlik rasgele başlangıç yöntemi sonucu elde edilen merkez noktalarının değişimi ile yerini farklılığa bırakabilir. Macqueen başlangıç yönteminin kullanılması sonucu elde edilen kümelerin diğer iki başlangıç yöntemi ile elde edilen kümelerden farklı olduğu görülmüştür. Sonuç olarak süsen çiçeği veritabanı üzerinde üç başlangıç yönteminin kullanılması sonucu KHM, Hibrit 2 ve FKM' nin başlangıçtaki değerlere çok fazla duyarlı olmadığı hatta neredeyse hiç etkilenmediği, KM ve Hibrit 1' in ise başlangıçtaki değerlere duyarlı olduğu görülmüştür. KM ve Hibrit 1 algoritmaları başlangıç merkezlerinin değişimine paralel olarak sayıca büyükleri ve içerikleri farklı kümeler üretmişlerdir. Her bir algoritmanın kendi adıyla başlayan metin dosyalarının incelenmesi sonucu algoritmalarla ilgili bu karara varılmıştır. Toplam karesel hata değerleri incelendiğinde KHM, FKM ve Hibrit 2'nin iyi değerlere sahip olduğu görülmüştür. Buna karşın KM ve Hibrit 1' in ise toplam karesel hata değeri daha büyük çıkıştır. Toplam karesel hata değerinin büyük çıkması iyi bir kümelemenin yapıldığının değil kötü bir kümelenin yapıldığının belirtisidir.

Tablo 5.10: MacQueen, rasgele ve rasgele bölümleme yöntemlerinin süsen çiçeği veritabanı üzerinde uygunlanması.

Veritabanı	**Başlangıç Yöntemi**	**Algoritma**	**Toplam Karesel Hata Aralığı**	**Kümeler içindeki niteliklerin eleman sayıları**
Süsen Çiçeği	**MacQueen Yöntemi**	**KM**	10.05–5.41	**1.küme**–0,3,36 **2.küme**–0,47,14 **3.küme**–50,0,0
		FKM	7.92–5.41	**1.küme**–50,0,0 **2.küme**–0,4,38 **3.küme**–0,46,12
		KHM	10.05–5.41	**1.küme**–50,0,0 **2.küme**–0,4,36 **3.küme**–0,46,14
		Hibrit 1	10.05–5.99	**1.küme**–28,0,0 **2.küme**–0,50,50 **3.küme**–22,0,0
		Hibrit 2	10.05–5.41	**1.küme**–50,0,0 **2.küme**–0,4,36 **3.küme**–0,46,14
	Rasgele Yöntemi	**KM**	9.93–5.43	**1.küme**–0,10,42 **2.küme**–50,0,0 **3.küme**–0,40,8
		FKM	6.75–5.41	**1.küme**–50,0,0 **2.küme**–0,46,12 **3.küme**–0,4,38
		KHM	8.51–5.41	**1.küme**–0,46,14 **2.küme**–0,4,36 **3.küme**–50,0,0
		Hibrit 1	8.16–5.45	**1.küme**–0,36,3 **2.küme**–50,0,0 **3.küme**–0,14,47

Tablo 5.10 "(DEVAM)": Macqueen, rastgele ve rastgele bölümleme yöntemlerinin süsen çiçeği veritabanı üzerinde uygulanması.

Süsen Çiçeği	**Rastgele Yöntemi**	**Hibrit 2**	9.26–5.41	**1.küme**–0,46,14 **2.küme**–0,4,36 **3.küme**–50,0,0
	Rastgele Bölümleme Yöntemi	**KM**	5.43–5.41	**1.küme**–0,47,14 **2.küme**–50,0,0 **3.küme**–0,3,36
		FKM	5.42–5.41	**1.küme**–0,46,12 **2.küme**–0,4,38 **3.küme**–50,0,0
		KHM	5.43–5.41	**1.küme**–0,4,36 **2.küme**–0,46,14 **3.küme**–50,0,0
		Hibrit 1	5.45–5.43	**1.küme**–0,14,47 **2.küme**–0,36,3 **3.küme**–50,0,0
		Hibrit 2	5.43–5.41	**1.küme**–50,0,0 **2.küme**–0,4,36 **3.küme**–0,46,14

Başlangıç yöntemlerinin hangi algoritmalar üzerinde daha etkili olduğuna dair yapılan diğer bir karşılaştırma işlemi mamografi veritabanı üzerinde yapılmıştır. Tablo 5.11' de mamografi veritabanı üzerinde üç başlangıç yöntemine ilişkin olarak merkez tabanlı kümeleme algoritmalarının toplam karesel hata değerleri ve kümeler içindeki niteliklerin eleman sayıları verilmiştir. Her bir algoritmanın çalıştırılmasından sonra kendi adıyla kaydedilen metin dosyaları incelendiğinde mamografi veritabanı üzerinde yapılan karşılaştırma işleminde KHM, FKM ve Hibrit 2 algoritmalarının başlangıç yöntemlerinden etkilenmediği diğerlerinin ise etkilendiği görülmüştür. KHM algoritması ile üç başlangıç yönteminin kullanılması sonucu oluşan kümeler birbirinin aynısı çıkmıştır. Aynı durum FKM ve Hibrit 2 içinde geçerlidir. KM algoritması ve Hibrit 1 algoritmaları üzerinde yapılan karşılaştırma sonuçları incelediğinde bu iki algoritmanın başlangıç noktalarından etkilendiği saptanmıştır. KM ve Hibrit 1' de kullanılan üç farklı başlangıç yöntemi

sonucu elde edilen kümeler birbirinden farklı çıkmıştır. Algoritmalar toplam karesel hata değerlerine göre karşılaştırıldığında KHM ve Hibrit 2' nin düşük toplam karesel hata değerlerine sahip oldukları görülmektedir. Düşük toplam karesel hata değeri algoritmaların yapılan kümeleme işleminde ne kadar başarılı olduklarını gösteren bir değerdir. Buna göre mamografi veritabanı üzerinde yapılan karşılaştırmalar sonucunda KHM, Hibrit 2 ve FKM' in başarılı kümelemeler yaptıkları tespit edilmiştir. Fakat rastgele yöntemi sonucu üretilen farklı başlangıç noktalarının kullanılması sonucu bu benzerlik yerini farklılığa bırakabilir. KM algoritması ve Macqueen ve rastgele başlangıç yöntemi sonucu elde edilen kümeler sayıca ve içerik olarak birbirine benzemektedir. Fakat KM algoritması ve rastgele bölümleme sonucu elde edilen kümeler ise diğer iki yöntemle elde edilen kümelere benzememektedir. Hibrit 1 ve rastgele ve rastgele bölümleme sonucu elde edilen kümeler içerik ve sayıca birbirinin aynısıdır. Hibrit 1 ve Macqueen başlangıç yöntemi sonucu elde edilen kümeler ise diğer iki yöntemle elde edilen kümelere benzememektedir.

Tablo 5.11: MacQueen, rasgele ve rasgele bölümleme yöntemlerinin Mamografi veritabanı üzerinde uygulanması

Veritabanı	**Başlangıç Yöntemi**	**Algoritma**	**Toplam Karesel Hata Aralığı**	**Kümeler içindeki eleman sayıları**
Mamografi	**MacQueen Yöntemi**	**KM**	22.20–17.11	**1.küme**–55,43 **2.küme**–357,54 **3.küme**–104,348
		FKM	18.32–17.09	**1.küme**–89,20 **2.küme**–307,59 **3.küme**–120,366
		KHM	22.20–16.92	**1.küme**–61,20 **2.küme**–332,59 **3.küme**–123,366
		Hibrit 1	22.20–17.05	**1.küme**–72,63 **2.küme**–360,53 **3.küme**–84,329

Tablo 5.11 "(DEVAM)": MacQueen, rasgele ve rasgele bölümleme yöntemlerinin Mamografi veritabanı üzerinde uygulanması.

		Hibrit 2	22.20–16.99	**1.küme**–62,21 **2.küme**–331,59 **3.küme**–123,365
	Rasgele Yöntemi	**KM**	22.28–16.89	**1.küme**–55,43 **2.küme**–357,54 **3.küme**–104,348
		FKM	17.51–17.09	**1.küme**–89,20 **2.küme**–120,366 **3.küme**–307,59
		KHM	19.59–16.91	**1.küme**–61,20 **2.küme**–123,366 **3.küme**–332,59
		Hibrit 1	19.54–17.32	**1.küme**–162,31 **2.küme**–126,372 **3.küme**–228,42
		Hibrit 2	21.73–16.99	**1.küme**–123,365 **2.küme**–62,21 **3.küme**–331,59
	Rasgele Bölümleme Yöntemi	**KM**	22.07–16.88	**1.küme**–61,20 **2.küme**–123,366 **3.küme**–332,59
		FKM	21.40–17.09	**1.küme**–89,20 **2.küme**–120,366 **3.küme**–307,59
		KHM	22.07–16.92	**1.küme**–123,366 **2.küme**–332,59 **3.küme**–61,20
		Hibrit 1	22.07–17.31	**1.küme**–126,372 **2.küme**–162,31 **3.küme**–228,42
		Hibrit 2	22.07–16.99	**1.küme**–331,59 **2.küme**–123,365 **3.küme**–62,21

Tüm veritabanları üzerinde yapılan karşılaştırmalar sonucunda KHM ve Hibrit 2' nin başlangıç noktalarına duyarlı olmadığı saptanmıştır. FKM' in de başlangıç noktalarına çok fazla duyarlı olmadığı saptanmıştır. KM ve Hibrit 1 algoritmalarının başlangıç noktalarına duyarlı olduğu tespit edilmiştir. Buna göre başlangıç noktalarına karşı duyarlılığın olmamasının istendiği kümeleme işlemlerinde KHM ve Hibrit 2' nin kullanılması yerinde olur. Başlangıç noktalarına karşı duyarlılığın hassas olmadığı durumlarda ise FKM, KM ve Hibrit 1 algoritmaları tercih edilebilir.

2) K Küme Sayısının Kümelemeye Etkisi: Merkez tabanlı kümeleme algoritmaları başlangıçta kullanıcı tarafında belirlenen k sayısına ihtiyaç duyar. k sayısı oluşturulacak olan küme sayısını ifade etmektedir. Bu uygulama ile k sayısının değişiminden merkez tabanlı kümeleme algoritmalarının kümeleme sonuçlarının nasıl etkilendiği araştırılmıştır. Algoritmalar toplam karesel hata ve işlemci zamanı bakımında karşılaştırılmışlardır. Bu karşılaştırma işlemleri süsen çiçeği ve mamografi veritabanı üzerinde yapılmıştır. Özellikle bu iki veritabanının seçilmemesinin nedeni süsen çiçeği veritabanının en az diğerinin ise en fazla kayıt sayısına sahip olmasıdır.

İlk karşılaştırma yapılan veritabanı süsen çiçeği veritabanıdır. Karşılaştırma işleminde k değeri 3, başlangıç yöntemi olarak Macqueen başlangıç yöntemi ve uzaklık ölçümü olarak Öklit uzaklığı alınmıştır. Macqueen başlangıç yönteminin seçilmesinin nedeni ilk k tane kaydın başlangıç noktası olarak seçilmesiyle oluşan belirgin merkez noktaları ile karşılaştırma işleminin yapılabilmesidir. Diğer başlangıç yöntemlerinde her algoritmanın çalıştırılmasıyla farklı merkez noktaları seçilebilmekte dolayısıyla bazı algoritmalar için sonuçlar farklı çıkmaktadır. Bu durumun önüne geçmek ve sadece k değerlerinin değişiminden merkez tabanlı kümeleme algoritmalarının nasıl etkilendiğini görmek için Macqueen başlangıç yöntemi tercih edilmiştir. FKM algoritması için r değeri 2, KHM, Hibrit 1 ve Hibrit 2 için p değeri de 3.5 olarak alınmıştır. Bu veritabanına ilişkin sonuçlar Tablo 5.12' da gösterilmiştir. Tablo 5.12' daki değerler incelendiğinde KM ve Hibrit 1' in diğerlerine göre daha hızlı çalıştığı görülmektedir. Toplam karesel hata değerlerine göre algoritmalar incelendiklerinde ise durum değişmektedir. Toplam karesel hata değeri en düşükten en büyüğe doğru olan algoritmaların KHM, Hibrit 2, FKM, KM

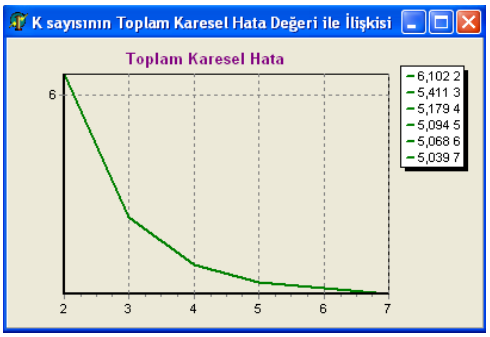
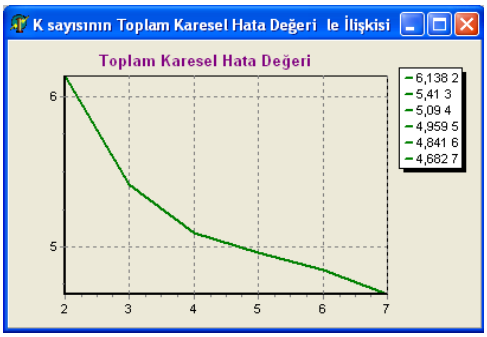
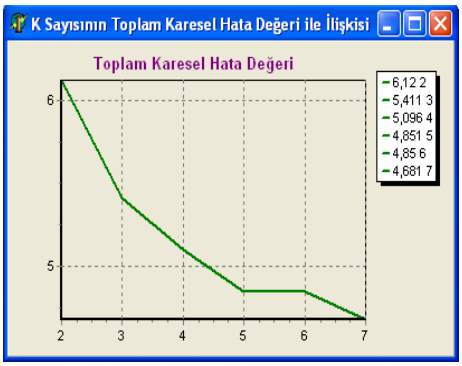
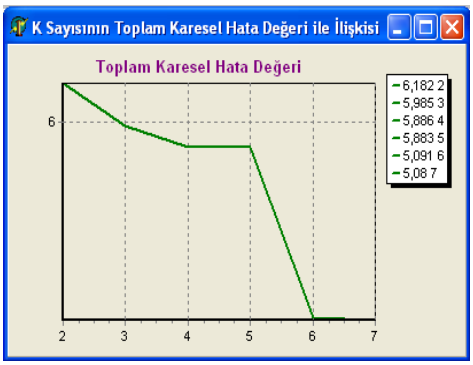
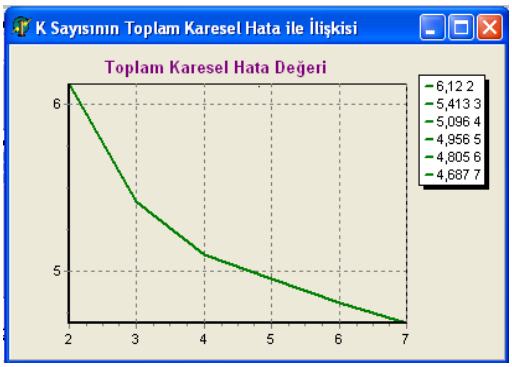
ve Hibrit 1 olduğu görülmüştür. Toplam karesel hata değerinin düşük olması ilgili algoritmanın performansının iyi olduğunu ve iyi bir kümeleme yaptığını göstermektedir.

k sayısının artışı ile orantılı olarak toplam karesel hata değeri de azalmaktadır. Hangi k değerinin daha iyi olduğunun karar vermede toplam karesel hata değeri dikkate alınır. Bir k değerinden diğer k+1 değerine geçerken bu iki k değerlerine ilişkin toplam karesel hata değerlerinin farkı alınır. Eğer fark diğer k değerleri arasındaki fark değerinden daha büyük ise k+1 değerinin en uygun k değeri olduğuna karar verilir. Tablo 5.12' de k sayısının ile toplam karesel hata değeri arasındaki ilişki görsel olarak ifade edilmektedir. Tablo 5.14 incelendiğinde KM, FKM, KHM, Hibrit 2 için en uygun k değerinin 3 olduğu, Hibrit 1 içinse 6 olduğu saptanmıştır. KM, FKM, KHM ve Hibrit 2'nin k=2' deki toplam karesel hata değerinden, k=3'teki toplam karesel hata değerinin çıkardığımızda elde edilen fark diğer k değerlerinden elde edilen farktan daha büyük olduğundan en uygun k değerinin bu algoritmalar için 3 olduğuna karar verilmiştir. Hibrit 1' in k=5' teki toplam karesel hata değerinden k=6'daki toplam karesel hata değeri çıkarıldığında elde edilen fark, diğer k değerlerine göre elde edilen farktan daha büyük olduğundan bu algoritma için en uygun k değerinin 6 olduğuna karar verilmiştir. k sayısının her bir algoritma için en uygun değeri görsel olarak Tablo 5.13' te de görülmektedir. K sayısının kümeleme üzerine önemli bir etkisi vardır. Verilen k sayısına göre ne kadar küme olacağı belirlenmekte ve bu k değeri kadar başlangıç yöntemleri ile ilk merkezler oluşturulmaktadır. Bu ilk merkezler ile başlayarak kümeleme algoritmaları kullanılarak kümeler oluşturulmaktadır. k sayısını olması gerekenden daha büyük verirsek veriler daha fazla kümeye ayrılır ve bazen de verilerin olması gereken daha fazla kümeye ayrılması tam olarak istenen sonuç olmayabilir ve doğal kümelerin oluşmasını verilen k sayısına bağlı olarak zorlaşabilir. Bu nedenle en uygun k değerinin belirlenmesi iyi bir kümele için gereklidir. Örneğin; k değeri 3 olması gereken bir kümelemede k sayısını 5 vermemiz durumda kümeler 5 küme içinde paylaştıracaktır ve doğal olarak bazı veri noktaları olmaması gereken kümeler içine girebilecektir.

Tablo 5.12: Süsen çiçeği üzerinde k sayısının son toplam karesel hata ve işlemci zamanı üzerindeki etkisi.

Veritabanı	K sayısı	Algoritma	Toplam Karesel Hata	Toplam İşlemci Zamanı(s)
Süsen Çiçeği	2	KM	6.10	0.1676
		FKM	6.14	0.1688
		KHM	6.12	0.1879
		Hibrit 1	6.18	0.1697
		Hibrit 2	6.12	0.1516
	3	KM	5.41	0.2379
		FKM	5.41	0.2370
		KHM	5.41	0.3274
		Hibrit 1	5.99	0.2181
		Hibrit 2	5.41	0.2425
	4	KM	5.18	0.2852
		FKM	5.09	0.5213
		KHM	5.09	0.3018
		Hibrit 1	5.89	0.5825
		Hibrit 2	5.09	0.4396
	5	KM	5.09	0.3461
		FKM	4.96	0.4432
		KHM	4.85	0.4465
		Hibrit 1	5.88	0.4988
		Hibrit 2	4.96	0.3596
	6	KM	5.07	0.4104
		FKM	4.84	0.6273
		KHM	4.85	0.5629
		Hibrit 1	5.09	0.6728
		Hibrit 2	4.81	0.4266
	7	KM	5.04	0.7059
		FKM	4.68	0.5886
		KHM	4.68	0.5931
		Hibrit 1	5.08	0.5361
		Hibrit 2	4.69	0.5286

Tablo 5.13: Süsen çiçeği üzerinde k sayısının toplam karesel hata değeri ile ilişkisi.

KM	FKM
	
KHM	**Hibrit 1**
	
Hibrit 2	
	

K sayısının merkez tabanlı kümeleme algoritmalarının toplam karesel hata değeri ve işlemci zamanı üzerindeki etkisini görebilmek amacıyla üzerinde karşılaştırma işlemi

yaptığımız ikinci veritabanı mamografi veritabanıdır. Mamografi veritabanına ait olan sonuçlar Tablo 5.14' de görülmektedir. Tablo 5.14' deki değerler incelendiğinde KM' in diğerlerine göre daha hızlı çalıştığı görülmektedir. Toplam karesel hata değerlerine göre algoritmalar incelendiklerinde ise durum değişmektedir. Toplam karesel hata değeri en düşükten en büyüğe doğru olan algoritmaların KHM, Hibrit 2, FKM, KM ve Hibrit 1 olduğu görülmüştür. Seçilecek olan algoritmada kriter olarak hız ön planda ise KM' nin, performans ön planda ise toplam karesel hata değeri düşük olan algoritmaların seçilmesi tavsiye edilir. Tablo 5.16' teki sonuçlar incelendiğinde KM, FKM, KHM, Hibrit 2 için en uygun k değerinin 3 olduğu, Hibrit 1 içinse 4 olduğu saptanmıştır. KM, FKM, KHM ve Hibrit 2'nin k=2' deki toplam karesel hata değerinden, k=3' teki toplam karesel hata değerinin çıkardığımızda elde edilen fark diğer k değerlerinden elde edilen farktan daha büyük olduğundan en uygun k değerinin bu algoritmalar için 3 olduğuna karar verilmiştir. Hibrit 1' in k=3' teki toplam karesel hata değerinden k=4'daki toplam karesel hata değeri çıkarıldığında elde edilen fark, diğer k değerlerine göre elde edilen farktan daha büyük olduğundan bu algoritma için en uygun k değerinin 4 olduğuna karar verilmiştir. Tablo 5.15' daki merkez tabanlı kümeleme algoritmalarına ait olan k değerine tekabül eden toplam karesel hata değerleri de incelendiğinde KM, FKM, KHM ve Hibrit 2 için k=3 olduğu noktanın, Hibrit 1 içinse k=4 olduğu noktanın görsel olarak en fazla değişimin yaşandığı nokta olduğu tespit edilmiştir.

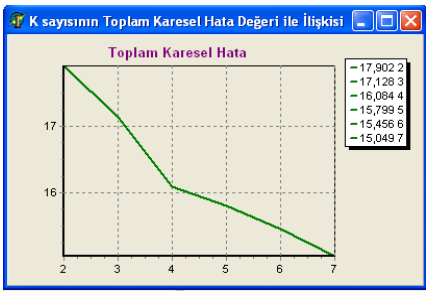
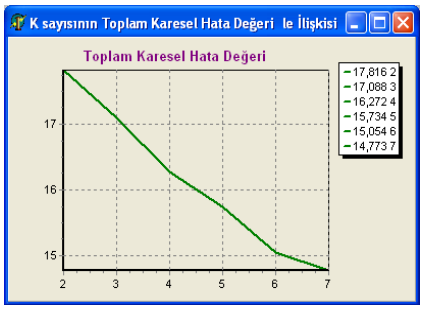
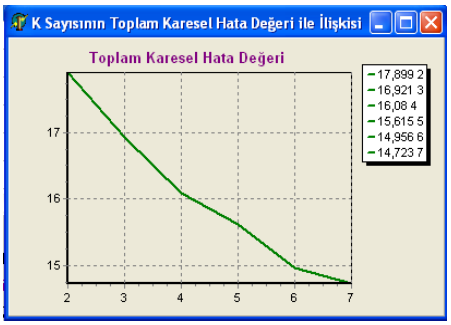
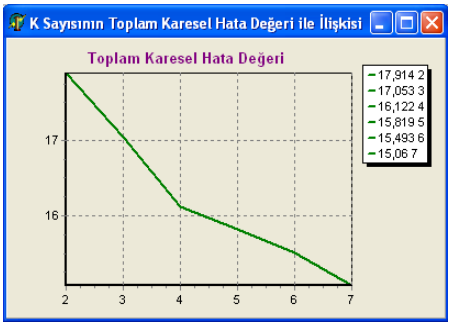
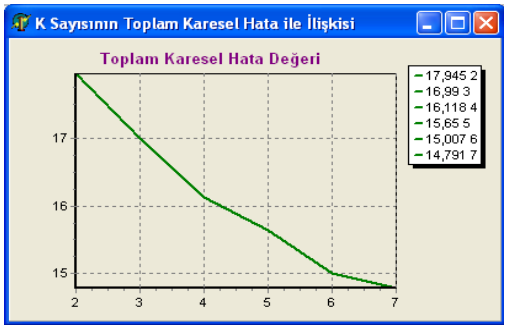
Tablo 5.14: Mamografi veritabanı üzeride k sayısının son toplam karesel hata ve işlemci zamanı üzerindeki etkisi.

Veritabanı	**K sayısı**	**Algoritma**	**Toplam Karesel Hata**	**Toplam İşlemci Zamanı(s)**
Mamografi	**2**	**KM**	17.90	0.7507
		FKM	17.82	0.7301
		KHM	17.89	0.7714
		Hibrit 1	17.91	0.8043
		Hibrit 2	17.95	0.8381
		KM	12.13	1.1084

Tablo 5.14 “(DEVAM)”: Mamografi veritabanı üzerinde k sayısının son toplam karesel hata ve işlemci zamanı üzerindeki etkisi.

	3	**FKM**	17.09	1.2756
		KHM	16.92	1.1946
		Hibrit 1	17.05	1.2479
		Hibrit 2	16.99	1.1790
	4	**KM**	16.08	1.4663
		FKM	16.27	1.7144
		KHM	16.08	1.8604
		Hibrit 1	16.12	1.7461
		Hibrit 2	16.12	1.7789
	5	**KM**	15.79	1.9284
		FKM	15.74	2.2347
		KHM	15.62	2.1169
		Hibrit 1	15.82	2.0546
		Hibrit 2	15.65	1.9436
	6	**KM**	15.46	2.4985
		FKM	15.05	2.5427
		KHM	14.96	2.3143
		Hibrit 1	15.49	2.3168
		Hibrit 2	15.01	2.4859
	7	**KM**	15.05	2.7659
		FKM	14.77	2.0765
		KHM	14.72	2.9787
		Hibrit 1	15.06	3.0491
		Hibrit 2	14.79	3.0424

Tablo 5.15: Mamografi veritabanı üzerinde k sayısının toplam karesel hata değeri ile ilişkisi.

KM	FKM
	
KHM	**Hibrit 1**
	
Hibrit 2	
	

Diğer veritabanlarında yapılan karşılaştırmalarda bu iki veritabanındakilere benzer sonuçlar çıkmıştır. Bu nedenle diğer kümelere ilişkin sonuçları vermeye gerek duyulmamıştır. Bu yapılan çalışma ile küme sayısının merkez tabanlı kümeleme algoritmalarının sonuçları üzerindeki etkisi net bir şekilde görülmüştür. Verilen k

sayısına göre ne kadar küme olacağı belirlenmekte ve bu k değeri kadar başlangıç yöntemleri ile ilk merkezler oluşturulmaktadır. Bu ilk merkezler ile başlayarak kümeleme algoritmaları kullanılarak kümeler oluşturulmaktadır. Doğal olarak k sayısı yapılan kümeleme üzerinde önemli bir etkisi vardır.

3) Verinin Boyutunun Az ya da Çok Olması: Bu uygulamada merkez tabanlı kümeleme algoritmaları aynı k değeri ve veri kümesinin farklı boyutları ile çalıştırılmış ve oluşan kümelerin bu değişiminden ne kadar etkilendiği incelenmiştir. Oluşan kümeler sayıca ve içerik bakımından birbirine benzer olup olmadıklarına göre karşılaştırılmışlardır. Ayrıca algoritma bazında oluşan kümeler içindeki niteliklerin eleman sayıları da incelenmiş ve toplam karesel hata değerlerine göre karşılaştırma işlemi yapılmıştır. Bu karşılaştırma işlemi tüm veritabanları üzerinde yapılmıştır. Fakat tezde sadece süsen çiçeği ve mamografi veritabanları üzerinde yapılan karşılaştırma sonuçlarına yer verilmiştir. Karşılaştırma işlemlerinde FKM için r değeri 2, KHM, Hibrit 1 ve Hibrit 2 için p değeri 3.5, iterasyon değeri de 100 ve k değeri de 3 olarak alınmıştır. Rasgele ve rasgele bölümleme başlangıç yöntemlerinde başlangıç merkezleri rasgele oluşturulduğu için farklı sayıda elemanlara sahip kümelerin oluşması durumunda bunun başlangıç yönteminde mi yoksa veri boyutunun değişiminden mi kaynadığını anlamamız zor olacağından karşılaştırma işlemlerinde Macqueen başlangıç yöntemi kullanılmıştır. Macqueen başlangıç yönteminde veri kümesi içindeki ilk k tane eleman sırası ile merkez olarak ele alınmaktadır. Dolayısıyla algoritmaların her çalıştırılışında aynı merkezler alındığı için algoritmaların boyut artışından nasıl etkilendiklerini kolay bir şekilde anlayabiliriz.

İlk karşılaştırma yapılan veritabanı süsen çiçeği veritabanıdır. Süsen çiçeği veritabanındaki nitelik sayısı 4 olduğundan en fazla 4 boyutuna kadar karşılaştırma yapılabilmiştir. Tablo 5.16 incelediğinde merkez tabanlı kümeleme algoritmalarının boyut artışından etkilendiği görülmüştür. Boyut artıkça toplam karesel hata değeri yükselmiş ve oluşan kümeler içindeki eleman sayıları değişmiştir. Toplam karesel hata değerinin artması kümelemenin gittikçe daha da kötüleştiğinin göstermektedir. Toplam karesel hata değeri ne kadar küçük ise o kadar iyi bir kümeleme yapılmış demektir.

Tablo 5.16: Süsen çiçeği veritabanı üzerindeki boyut artışının toplam karesel hata değeri ve eleman sayıları üzerindeki etkisi.

Veritabanı: Süsen Çiçeği **K:** 3			
Boyut	**Algoritma**	**Toplam Karesel Hata**	**Kümeler içindeki niteliklerin eleman sayısı**
2	KM	4.60	**1.küme**–0,13,30 **2.küme**–1,37,20 **3.küme**–49,0,0
	FKM	4.58	**1.küme**–49,0,0 **2.küme**–0,13,34 **3.küme**–1,37,16
	KHM	4.59	**1.küme**–0,13,34 **2.küme**–1,37,16 **3.küme**–49,0,0
	Hibrit 1	4.96	**1.küme**–32,0,0 **2.küme**–0,44,49 **3.küme**–19,6,1
	Hibrit 2	4.59	**1.küme**–0,13,34 **2.küme**–1,37,16 **3.küme**–49,0,0
3	KM	4.98	**1.küme**–0,10,35 **2.küme**–0,40,15 **3.küme**–50,0,0
	FKM	4.97	**1.küme**–50,0,0 **2.küme**–0,10,36 **3.küme**–0,40,14
	KHM	4.97	**1.küme**–50,0,0 **2.küme**–0,10,35 **3.küme**–0,40,15
	Hibrit 1	5.41	**1.küme**–32,0,0 **2.küme**–0,47,50 **3.küme**–18,3,0

Tablo 5.16 "DEVAM": Süsen çiçeği veritabanı üzerindeki boyut artışının toplam karesel hata değeri ve eleman sayıları üzerindeki etkisi.

3	Hibrit 2	4.97	**1.küme**–50,0,0 **2.küme**–0,10,36 **3.küme**–0,40,14
4	KM	5.41	**1.küme**–0,3,36 **2.küme**–0,47,14 **3.küme**–50,0,0
	FKM	5.41	**1.küme**–50,0,0 **2.küme**–0,4,38 **3.küme**–0,46,12
	KHM	5.41	**1.küme**–50,0,0 **2.küme**–0,4,36 **3.küme**–0,46,14
	Hibrit 1	5.99	**1.küme**–28,0,0 **2.küme**–0,50,50 **3.küme**–22,0,0
	Hibrit 2	5.41	**1.küme**–50,0,0 **2.küme**–0,4,36 **3.küme**–0,46,14

Tablo 5.16' de KM algoritmasının toplam karesel hata değerinin boyut sayısı arttıkça yükseldiği görülmüştür. Toplam karesel hata değerinin yüksek olması iyi bir kümelemenin yapılmadığı anlamına gelmektedir. Boyut sayısının artışı ile hem toplam karesel hata değeri değişmiş hem de her bir boyutta oluşan kümelerin içerikleri farklı çıkmıştır. 2, 3 ve 4 boyutuna göre çalıştırılan KM algoritması sonucunda sadece süsen çiçeğinin Setosa türünün net bir şekilde tespit edildiği, Versicolor ve Virginica türünün ise birbirine karıştığı görülmüştür. Setosa türü 3. ve 4. boyutlarda elde edilen kümelerde daha net bir şekilde görülmektedir. Buna karşın 4. boyutta elde edilen kümelerde Versicolor ve Virginica türünün birbirine karıştığı tamamen belli olmuştur. Bu karşılaştırma işlemi sonucunda KM algoritmasının boyut artışından etkilendiği ortaya çıkmıştır. Karşılaştırma da kullanılan diğer algoritma olan FKM algoritması da boyut artışından etkilenmiştir. FKM algoritmasının boyut artışı ile toplam karesel hata değeri gittikçe artmış ve oluşan kümeler içindeki eleman sayıları da değişmiştir. FKM algoritmasında da Setosa türü 3. ve 4. boyutta

oluşturulan kümelerde net bir şekilde tespit edilmiştir. Diğer türler ise birbirine karışmıştır. KHM algoritmasında da boyut değişimi ile oluşan kümelerin sayıca büyüklükleri de değişmiştir. KHM algoritması ile 3. ve 4. boyutlarda oluşturulan kümeler Setosa türü net bir şekilde kümelendiği tespit edilmiştir. Fakat Versicolor ve Virginica türleri yine net bir şekilde tespit edilememiştir. Bu iki birbirine çok yakın nitelik değerlerine sahip olduklarından algoritmalar bu iki türü ayrı küme yerleştirememiştir. KHM algoritmasının bu karşılaştırma ile boyut artışından etkilendiği görülmüştür. Hibrit 1 algoritması ve Hibrit algoritması da boyut değişiminden etkilenmiş ve boyut değişimi direkt oluşan kümeler ve toplam karesel hata değeri üzerine yansımıştır. Hibrit 1 algoritmasında diğer algoritmalardan farklı olarak Versicolor ve Virginica türünü ek olarak Setosa türü de hiçbir boyutta net olarak tespit edilememiştir. Hibrit 2 algoritmasında ise Setosa türü 3. ve 4. boyutta oluşturulan kümelerde net olarak tespit edilmiş fakat diğer türler yine belirgin bir şekilde tespit edilememiştir. Merkez tabanlı kümeleme algoritmalarının kendi adıyla başlayan metin dosyaları her bir boyut için ayrı ayrı incelediğinde bu algoritmaların kümeleme sonuçlarının boyut değişiminde ne kadar etkilendiği daha net bir şekilde görülebilmektedir.

İkinci karşılaştırma yaptığımız veritabanı mamografi veritabanıdır. Mamografi veritabanı üzerinde farklı boyutlarda merkez tabanlı kümeleme algoritmalarının işletilmesine dair sonuçlar Tablo 5.17' de gösterilmiştir. Mamografi veritabanı üzerinde uygulanan merkez tabanlı kümeleme algoritmalarının toplam karesel hata değerleri ve kümeler içindeki elemanların sayıları ve içerikleri incelendiğinde algoritmaların boyut artışından etkilendikleri görülebilmektedir. Merkez tabanlı kümeleme algoritmalarının hepsi boyut artışından etkilenmişlerdir. Bazıları daha fazla bazıları da az etkilenmişlerdir. Tablo 5.17' de görüldüğü gibi boyut arttıkça algoritmaların toplam karesel hata değerleri de artmış ve kümeler sayıca büyüklükleri ve içerikleri de değişmiştir. Toplam karesel hata değerinin artması algoritmaların iyi bir kümeleme yapamadığının göstergesidir. Merkez tabanlı kümeleme algoritmalarından en azından bu veritabanı için KM' nin diğerlerine göre daha az boyut artışından etkilendiği görülmüştür. Hatta 4. ve 5. boyutta kümelerin nerdeyse sayıca büyükleri eşit ve içindeki elemanlar da birbirine benzer çıkmıştır.

Diğer algoritmaların oluştukları kümeler arasında da benzerlikler vardır. Fakat bu benzerlikler birebir benzerlikler değildir.

Tablo 5.17: Mamografi veritabanı üzerindeki boyut artışının toplam karesel hata değeri ve eleman sayıları üzerindeki etkisi.

Veritabanı: Mamografi **K:** 3			
Boyut	**Algoritma**	**Toplam Karesel Hata**	**Kümeler içindeki niteliklerin eleman sayısı**
2	KM	4.60	**1.küme**–142,227 **2.küme**–343,63 **3.küme**–31,155
	FKM	4.58	**1.küme**–35,277 **2.küme**–241,62 **3.küme**–240,106
	KHM	4.59	**1.küme**–241,106 **2.küme**–239,60 **3.küme**–36,279
	Hibrit 1	4.96	**1.küme**–217,98 **2.küme**–262,58 **3.küme**–37,289
	Hibrit 2	4.59	**1.küme**–36,279 **2.küme**–238,60 **3.küme**–242,106
3	KM	4.98	**1.küme**–217,50 **2.küme**–205,50 **3.küme**–94,345
	FKM	4.97	**1.küme**–216,49 **2.küme**–205,50 **3.küme**–95,346
	KHM	4.97	**1.küme**–216,50 **2.küme**–205,50 **3.küme**–95,345
	Hibrit 1	5.41	**1.küme**–53,49 **2.küme**–381,84 **3.küme**–82,312

Tablo 5.17 “(DEVAM)”: Mamografi veritabanı üzerindeki boyut artışının toplam karesel hata değeri ve eleman sayıları üzerindeki etkisi

3	Hibrit 2	4.97	**1.küme**–216,50 **2.küme**–205,50 **3.küme**–95,345
4	KM	5.41	**1.küme**–53,41 **2.küme**–357,54 **3.küme**–106,350
	FKM	5.41	**1.küme**–71,64 **2.küme**–360,53 **3.küme**–85,328
	KHM	5.41	**1.küme**–59,42 **2.küme**–357,54 **3.küme**–100,349
	Hibrit 1	5.99	**1.küme**–73,63 **2.küme**–359,53 **3.küme**–84,329
	Hibrit 2	5.41	**1.küme**–56,42 **2.küme**–357,54 **3.küme**–103,349
5	KM	5.41	**1.küme**–55,43 **2.küme**–357,54 **3.küme**–104,348
	FKM	5.41	**1.küme**–89,20 **2.küme**–307,59 **3.küme**–120,366
	KHM	5.41	**1.küme**–61,20 **2.küme**–332,59 **3.küme**–123,366
	Hibrit 1	5.99	**1.küme**–72,63 **2.küme**–360,53 **3.küme**–84,329
	Hibrit 2	5.41	**1.küme**–62,21 **2.küme**–331,59 **3.küme**–123,365

Diğer veritabanlarında da bu iki veritabanındakilere benzer sonuçlar çıkmıştır. Bu nedenle diğer kümelere ilişkin sonuçları vermeye gerek duyulmamıştır. Merkez tabanlı kümeleme algoritmalarının bu karşılaştırma işlemi ile boyut artışından önemli ölçüde etkilendikleri saptanmıştır.

4) Aykırı Değerleri Kümelemeye Etkisi: Uygulama ile merkez tabanlı kümeleme algoritmalarının aykırı değerler karşısında sergiledikleri davranışları incelenmiş ve birbirleri ile karşılaştırılmıştır. Oluşan kümeler sayıca aynı büyükte olup olmama, kümelerin içeriklerinin değişip değişmeme durumuna ve toplam karesel hata değerlerine göre incelenmiştir. Bu karşılaştırma işlemi için veritabanları içindeki kayıtlardan birinin değerleri ile oynanıp diğer kayıtlardan olabildiğince sıra dışı bir kayıt olması sağlanmıştır. Üzerinde oynanmış kayıtları içermeyen veritabanı ile kayıtlarından biri sıra dışı yapılmış aynı veritabanı karşılaştırılarak aykırı değerlerin kümeleme üzerindeki etkisi ölçülmeye çalışılmıştır.

Karşılaştırma da başlangıç yöntemi olarak Macqueen başlangıç yöntemi kullanılmıştır. Bunun nedeni başlangıç noktalarının değişiminden kümelerin etkilenmesini önlemek ve kümelerin sadece sıra dışı bir kaydın olması durumunda etkilenilip etkilenmediğini araştırmaktır. Rasgele ve rasgele bölümleme başlangıç yöntemlerinde başlangıç merkezleri rasgele oluşturulduğu için farklı sayıda elemanlara sahip kümelerin oluşması durumunda bunun başlangıç yönteminde mi yoksa kayıtların bir tanesinin sıra dışı olmasından mı kaynadığını anlamamız zor olacağından karşılaştırma da Macqueen başlangıç yöntemi kullanılmıştır. Karşılaştırma işlemlerinde FKM için r değeri 2, KHM, Hibrit 1 ve Hibrit 2 için p değeri 3.5, iterasyon değeri de 100 ve k değeri de 3 olarak alınmıştır. Karşılaştırma işleminde kullanılacak olan ilk veritabanı süsen çiçeği veritabanıdır. Tablo 5.24' de süsen çiçeği veritabanı üzerindeki Öklit ve Manhattan uzaklık ölçümlerinin kategorisi içindeki topla karesel hata değerleri, işlemci zamanı ve kümelerdeki eleman sayıları verilmiştir. Tablo 5.24' de süsen çiçeği veritabanı içinde sıra dışı veri bulunmayan veritabanıdır. Tablo 5.25' de ise veritabanı içindeki diğer kayıtlardan oldukça sıra dışı olan bir kaydın bulunduğu süsen çiçeği veritabanı üzerinde merkez tabanlı kümeleme algoritmalarının çalıştırılması sonucu oluşan toplam karesel hata değerleri, işlemci zamanı ve oluşan kümeler içindeki eleman sayılarına dair sonuçlar

yer almaktadır. Süsen çiçeği veritabanı içindeki 26. kaydın değerleri (5,3,1.6,0.2) olan değerleri (11.2, 4.5, 7.2, 3) ile değiştirilerek sıra dışı bir kayıt oluşturulmuş ve bu yeni veritabanı da "Süsen_Çiçeği_Outlier" veritabanı olarak karşılaştırma işleminde kullanılmak üzere kaydedilmiştir. Tablo 5.18' deki sonuçlar Öklit ve Manhattan uzaklık kategorisi altında ele alınıştır. Bunun nedeni ise iki uzaklık ölçümünün sıra dışılıklar karşısında verdikleri tepkilerinde bu tez kapsamında incelenmek istenmesidir. Sonuçta merkez tabanlı kümeleme algoritmaları bu uzaklık ölçülerini hesaplamalarda kullandıkları için bu uzaklık ölçümlerinin sıra dışılıklardan etkilenmesi direkt olarak oluşacak kümeler üzerine yansıyacaktır. Tablo 5.18 ve Tablo 5.19' deki sonuçlar karşılaştırıldığında tek bir sıra dışı verinin kümeleme sonuçlarını nasıl değiştirdiği topla karesel hata değerleri, işlemci zamanı ve kümeler içindeki sayılara bakılarak net bir şekilde anlaşılabilmektedir. Fakat daha ayrıntılı olarak sıra dışılıkların etkisi incelenmek istenirse algoritmaların kendi adıyla algoritmanın işletilmesinden sonra kaydedilen metin dosyalarına başvurulmalıdır. Sıra dışı değer durumunda Öklit ve Manhattan uzaklık ölçütlerinin ikisi de bundan önemli ölçüde etkilenmiş ve oluşan kümelerin sayıca büyüklükleri değiştiği gibi içeriğindeki elemanlarda değişmiştir.

Tablo 5.18: Süsen çiçeği veritabanı üzerindeki toplam karesel hata değerleri ve kümeler içindeki eleman sayıları.

Uzaklık Ölçütü	**Algoritma**	**Toplam Karesel Hata**	**Kümeler içindeki niteliklerin eleman sayısı**
Öklit	KM	5.41	**1.küme**–0,3,36 **2.küme**–0,47,14 **3.küme**–50,0,0
	FKM	5.41	**1.küme**–50,0,0 **2.küme**–0,4,38 **3.küme**–0,46,12
	KHM	5.41	**1.küme**–50,0,0 **2.küme**–0,4,36 **3.küme**–0,46,14

Tablo 5.18 "(DEVAM)": Süsen çiçeği veritabanı üzerindeki toplam karesel hata değerleri ve kümeler içindeki eleman sayıları.

	Hibrit 1	5.99	**1.küme**–28,0,0 **2.küme**–0,50,50 **3.küme**–22,0,0
	Hibrit 2	5.41	**1.küme**–50,0,0 **2.küme**–0,14,36 **3.küme**–0,46,14
Manhattan	KM	6.99	**1.küme**–0,46,15 **2.küme**–0,4,35 **3.küme**–0,50,0
	FKM	6.97	**1.küme**–50,0,0 **2.küme**–0,3,35 **3.küme**–0,47,15
	KHM	6.98	**1.küme**–50,0,0 **2.küme**–0,4,35 **3.küme**–0,46,15
	Hibrit 1	7.91	**1.küme**–28,0,0 **2.küme**–0,50,50 **3.küme**–22,0,0
	Hibrit 2	6.98	**1.küme**–50,0,0 **2.küme**–0,4,35 **3.küme**–0,46,15

Tablo 5.19: Sıra dışı değer içeren süsen çiçeği veritabanı üzerindeki toplam karesel hata değerleri ve kümeler içindeki eleman sayıları.

Uzaklık Ölçütü	**Algoritma**	**Toplam Karesel Hata**	**Kümeler içindeki niteliklerin eleman sayısı**
Öklit	KM	5.39	**1.küme**–14,0,0 **2.küme**–1,50,50 **3.küme**–35,0,0
	FKM	4.94	**1.küme**–49,0,0 **2.küme**–1,3,39 **3.küme**–0,47,11

Tablo 5.19 "(DEVAM)": Sıra dışı değer içeren süsen çiçeği veritabanı üzerindeki toplam karesel hata değerleri ve kümeler içindeki eleman sayıları.

	KHM	4.95	**1.küme**–49,0,0 **2.küme**–0,47,12 **3.küme**–1,3,38
	Hibrit 1	5.45	**1.küme**–31,0,0 **2.küme**–1,50,50 **3.küme**–18,0,0
	Hibrit 2	4.95	**1.küme**–49,0,0 **2.küme**–1,2,38 **3.küme**–0,48,12
Manhattan	KM	6.40	**1.küme**–49,0,0 **2.küme**–1,3,34 **3.küme**–0,49,16
	FKM	6.36	**1.küme**–49,0,0 **2.küme**–1,2,36 **3.küme**–0,48,14
	KHM	6.37	**1.küme**–49,0,0 **2.küme**–0,48,14 **3.küme**–1,2,36
	Hibrit 1	7.19	**1.küme**–31,0,0 **2.küme**–1,50,50 **3.küme**–18,0,0
	Hibrit 2	6.37	**1.küme**–49,0,0 **2.küme**–0,48,15 **3.küme**–1,2,35

Tablo 5.18 ve Tablo 5.19' sonuçlar ve algoritmaların kendi adıyla başlayan metin dosyaları incelendiğinde aykırı değerlerden merkez tabanlı kümeleme algoritmalarını etkilendiği saptanmıştır. Süsen çiçeği veritabanı ile içinde bir tane sıra dışı değer olan süsen çiçeği veritabanının öklit uzaklık ölçümü bazındaki sonuçları incelediğinde KM algoritması sonucu oluşan kümelerin sayıca ve içerik olarak birbirine eşit olmadığı belirlenmiştir. Tek bir aykırı değer olsa bile farklı bir kümeleme ortaya çıkmıştır. Oluşan kümelerden bir tanesi 26 verisi ile başlamaktadır. Sonuç olarak KM

algoritmasının öklit uzaklı bazındaki sonuçları incelediğinde oluşan kümelerin çok farklı olduğu görülmüştür. Diğer bir algoritma olan FKM algoritması sonucu oluşan kümelerde sayıca birbirine yakı olsa bile içerik olarak oldukça farklı çıkmıştır. KM algoritması sonucu oluşan kümelerdeki gibi çok büyük değişiklik olmasa bile oluşan kümeler farklı çıkmış ve sıra dışı değer içeren süsen çiçeği veritabanına ait olan kümelerden biri 26 verisi ile başlamıştır. KHM algoritması sonucu oluşan kümelerde sayıca birbirine yakın olsa içerikleri çok fazla olmamakla birlikte farklı çıkmıştır. Ayrıca diğer algoritmalar olduğu gibi 26 verisi ile başlayan bir küme oluşmuştur. Hibrit 1 ve Hibrit 2 sonucu oluşan kümelerde de ufak tefek değişiklikler vardır. Oluşan kümelerden biri 26 verisi ile başlamıştır. Süsen çiçeği veritabanı ile içinde bir tane sıra dışı değer olan süsen çiçeği veritabanının Manhattan uzaklık ölçümü bazındaki sonuçları incelediğinde KM algoritması sonucu oluşan kümelerin çok büyük bir değişiklik olmasa bile değiştiği saptanmıştır. Sıra dışı değer içeren süsen çiçeği veritabanına KM algoritmasının uygulanması sonucu oluşan kümelerden biri 26 verisi ile başlamaktadır. Fakat bu oluşan kümeler Öklit uzaklık ölçümü kullanılarak oluşturulmuş olan kümelerden daha az değişiklik içermektedir. Buradan Manhattan uzaklık ölçümünün aykırı değerler karşısında daha dayanıklı olduğu anlaşılmıştır. FKM ve KHM algoritmalarına ait olan sonuçlar incelendiğinde de kümelerde çok az değişiklik olduğu ve 26 ile başlayan bir kümenin oluştuğu saptanmıştır. Yine oluşan kümelerde çok az değişiklik olması Manhattan uzaklık ölçümünden kaynaklanmaktadır. Hibrit 1 ve Hibrit 2 algoritmaları sonucu oluşan kümelerde çok fazla bir değişiklik olmadığı 26 ile başlayan bir küme oluştuğu görülmüştür. Sonuç olarak süsen çiçeği ve sıra dışı değer içeren süsen çiçeği veritabanları üzerinde uzaklık ölçümleri de kullanılarak merkez tabanlı kümeleme algoritmaları işletilmiş ve oluşan kümelerde aykırı değerlerden dolayı değişiklikler olduğu saptanmış ancak Manhattan uzaklık ölçümünün kullanıldığında Öklit uzaklık ölçümüne göre aykırı değerlerden daha az etkilenildiği keşfedilmiştir.

Karşılaştırma da kullanılan ikinci veritabanı mamografi veritabanı olmuştur. Tablo 5.20' de mamografi veritabanı üzerindeki Öklit ve Manhattan uzaklık ölçümlerinin kategorisi içindeki topla karesel hata değerleri, işlemci zamanı ve kümelerdeki eleman sayıları verilmiştir. Tablo 5.20' deki mamografi veritabanı içinde sıra dışı veri bulunmayan veritabanıdır. Tablo 5.21' de ise veritabanı içindeki sıra dışı bir

kayıt bulunmaktadır. Ttabloda da bu veritabanı üzerinde merkez tabanlı kümeleme algoritmalarının çalıştırılması sonucu oluşan toplam karesel hata değerleri ve oluşan kümeler içindeki eleman sayılarına dair sonuçlar yer almaktadır.

Mamografi veritabanı içindeki 803. kaydın değerleri (4, 31, 2, 1, 3, iyi huylu) olan değerleri (1, 31, 1, 1, 1, iyi huylu) ile değiştirilerek sıra dışı bir kayıt oluşturulmuş ve bu yeni veritabanı da "Mammografi_Outlier" veritabanı olarak karşılaştırma işleminde kullanılmak üzere kaydedilmiştir. Tablo 5.26 ve Tablo 5.27' deki sonuçlar karşılaştırıldığında tek bir sıra dışı verinin kümeleme sonuçlarını nasıl değiştirdiği topla karesel hata değerleri, işlemci zamanı ve kümeler içindeki sayılara bakılarak net bir şekilde anlaşılabilmektedir. Fakat daha ayrıntılı olarak sıra dışılıkların etkisi incelenmek istenirse algoritmaların kendi adıyla algoritmanın işletilmesinden sonra kaydedilen metin dosyalarına başvurulmalıdır. Sıra dışı değer durumunda Öklit ve Manhattan uzaklık ölçütlerinin ikisi de bundan önemli ölçüde etkilenmiş ve oluşan kümelerin sayıca büyüklükleri değiştiği gibi içeriğindeki elemanlarda değişmiştir.

Tablo 5.20: Mamografi veritabanı üzerindeki toplam karesel hata değerleri ve kümeler içindeki eleman sayıları.

Uzaklık Ölçütü	**Algoritma**	**Toplam Karesel Hata**	**Kümeler içindeki niteliklerin eleman sayısı**
Öklit	KM	17.128	**1.küme**–55,43 **2.küme**–357,54 **3.küme**–104,348
	FKM	17.088	**1.küme**–89,20 **2.küme**–307,59 **3.küme**–120,366
	KHM	16.921	**1.küme**–61,20 **2.küme**–332,59 **3.küme**–123,366
	Hibrit 1	17.053	**1.küme**–72,63 **2.küme**–360,53 **3.küme**–84,329

Tablo 5.20 ”(DEVAM)”: Mamografi veritabanı üzerindeki toplam karesel hata değerleri ve kümeler içindeki eleman sayıları.

	Hibrit 2	16.989	**1.küme**–62,21 **2.küme**–331,59 **3.küme**–123,365
Manhattan	KM	22.527	**1.küme**–53,41 **2.küme**–357,50 **3.küme**–106,354
	FKM	22.014	**1.küme**–73,17 **2.küme**–323,49 **3.küme**–120,379
	KHM	22.427	**1.küme**–64,45 **2.küme**–357,49 **3.küme**–95,351
	Hibrit 1	21.751	**1.küme**–72,51 **2.küme**–357,52 **3.küme**–87,342
	Hibrit 2	22.580	**1.küme**–60,45 **2.küme**–357,49 **3.küme**–99,351

Tablo 5.21: Sıra dışı değer içeren mamografi veritabanı üzerindeki toplam karesel hata değerleri ve kümeler içindeki eleman sayıları.

Uzaklık Ölçütü	**Algoritma**	**Toplam Karesel Hata**	**Kümeler içindeki niteliklerin eleman sayısı**
Öklit	KM	17.145	**1.küme**–55,43 **2.küme**–357,54 **3.küme**–104,348
	FKM	17.081	**1.küme**–90,20 **2.küme**–306,59 **3.küme**–120,366
	KHM	16.934	**1.küme**–62,20 **2.küme**–331,59 **3.küme**–123,366

Tablo 5.21"(DEVAM)": Sıra dışı değer içeren mamografi veritabanı üzerindeki toplam karesel hata değerleri ve kümeler içindeki eleman sayıları.

	Hibrit 1	17.069	**1.küme**–72,63 **2.küme**–359,53 **3.küme**–85,329
	Hibrit 2	17.003	**1.küme**–63,20 **2.küme**–330,59 **3.küme**–123,366
Manhattan	KM	22.562	**1.küme**–53,41 **2.küme**–357,50 **3.küme**–106,354
	FKM	22.037	**1.küme**–74,17 **2.küme**–322,49 **3.küme**–120,379
	KHM	22.469	**1.küme**–64,43 **2.küme**–357,49 **3.küme**–95,353
	Hibrit 1	21.775	**1.küme**–72,51 **2.küme**–357,52 **3.küme**–84,342
	Hibrit 2	22.622	**1.küme**–60,45 **2.küme**–357,49 **3.küme**–99,351

Mamografi veritabanı ve sıra dışı değer içeren mamografi veritabanlarına ait öklit bazındaki sonuçlar incelendiğinde ufak tefek deşikliklere rağmen her bir algoritmaya ait olan toplam karesel hata değerlerinin çok fazla değişmediği görülmüştür. Ayrıca her bir algoritmanın kendi adıyla başlayan dosyalar incelendiğinde ufak tefek değişikliklerle küme içeriklerinin de birbirine yakın olduğu görülmüştür. Eğer kullanılan kaydın sıra dışılığı artarsa oluşan kümelerdeki benzerlik yerini farklılığa bırakabilir. Mamografi veritabanı ve sıra dışı kaydı içeren mamografi veritabanı ait Manhattan uzaklık ölçümü bazındaki sonuçlar incelendiğinde ufak tefek değişikliklere rağmen her bir algoritmaya ait olan toplam karesel hata değerlerinin çok fazla değişmediği görülmüştür. Mamografi ve sıra dışı değer içeren Mamografi veritabanı üzerinde merkez tabanlı kümeleme algoritmalarının uygulanması sonucu

oluşan kümelerin içerikleri ve sayıları açısından da bir farklılık yoktur. Sonuç olarak merkez tabanlı kümeleme algoritmaları aykırı değerlerin işleme dâhil edilmesiyle farklı sonuçlar üretebilmektedir.

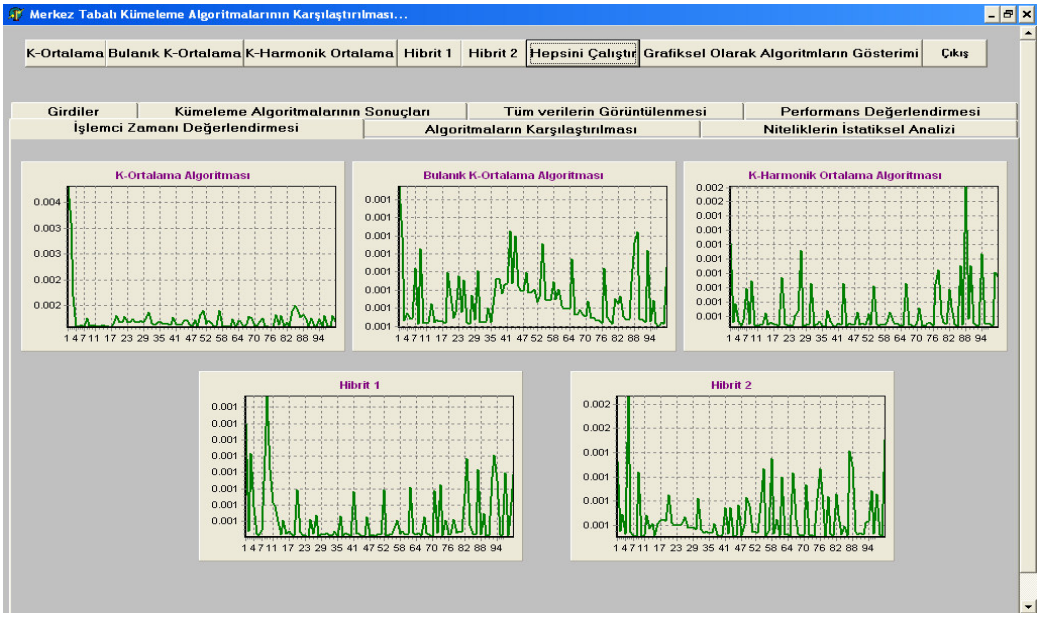
5) Algoritmaların Performans ve İşlemci Zamanı Açısından Karşılaştırılması:

Uygulama ile merkez tabanlı kümeleme algoritmaları süsen çiçeği ve mamografi veritabanları üzerinde topla karesel hata değeri ve işlemci zamanı açısından karşılaştırılmışlardır. Karşılaştırma işlemlerinde FKM için r değeri 2, KHM, Hibrit 1 ve Hibrit 2 için p değeri 3.5, iterasyon değeri 100, k değeri 3 ve başlangıç yöntemi de rasgele başlangıç yöntemi olarak alınmıştır. Süsen çiçeği veritabanı ve mamografi veritabanı üzerinde yapılan karşılaştırma sonuçları Tablo 5.22 ve Tablo 5.23' de verilmiştir. Veritabanları üzerindeki sonuçlar incelendiğinde KHM, FKM ve Hibrit 2 algoritmalarının toplam karesel hata değerlerinin diğerlerine göre daha düşük olduğu dolayısıyla performanslarının da diğerlerine göre daha fazla olduğu görülmektedir. İşlemci zamanına göre en hızlı olan algoritmanın ise KM ve Hibrit 1 olduğu anlaşılmıştır. Bu sonuçlar rasgele başlangıç yöntemi kullanıldığından genellikle değişmektedir. Süsen çiçeği veritabanı üzerinde uygulanan merkez tabanlı kümeleme algoritmaları ait olan toplam karesel hata değeri ve işlemci zamanına ilişkin sonuçlar Şekil 5.21 ve Şekil 5.22' te görsel olarak sunuluştur. Aynı şekilde mamografi veritabanına ait olan topla karesel hata değerleri ve işlemci zamanı sonuçları Şekil 5.23 ve Şekil 5.24' da görsel olarak sunulmuştur.

Tablo 5.22: Süsen çiçeği veritabanı üzerinde merkez tabanlı kümeleme algoritmaların toplam karesel hata ve işlemci zamanına göre karşılaştırılması.

Algoritma	**Toplam Karesel Hata**	**İşlemci Zamanı(s)**
KM	5.43	0.119
FKM	5.41	0.121
KHM	5.41	0.124
Hibrit 1	5.45	0.119
Hibrit 2	5.41	0.145

Şekil 5.21: Süsen veritabanı üzerine uygulanan algoritmaların toplam karesel hata değerlerinin görsel olarak sunumu.

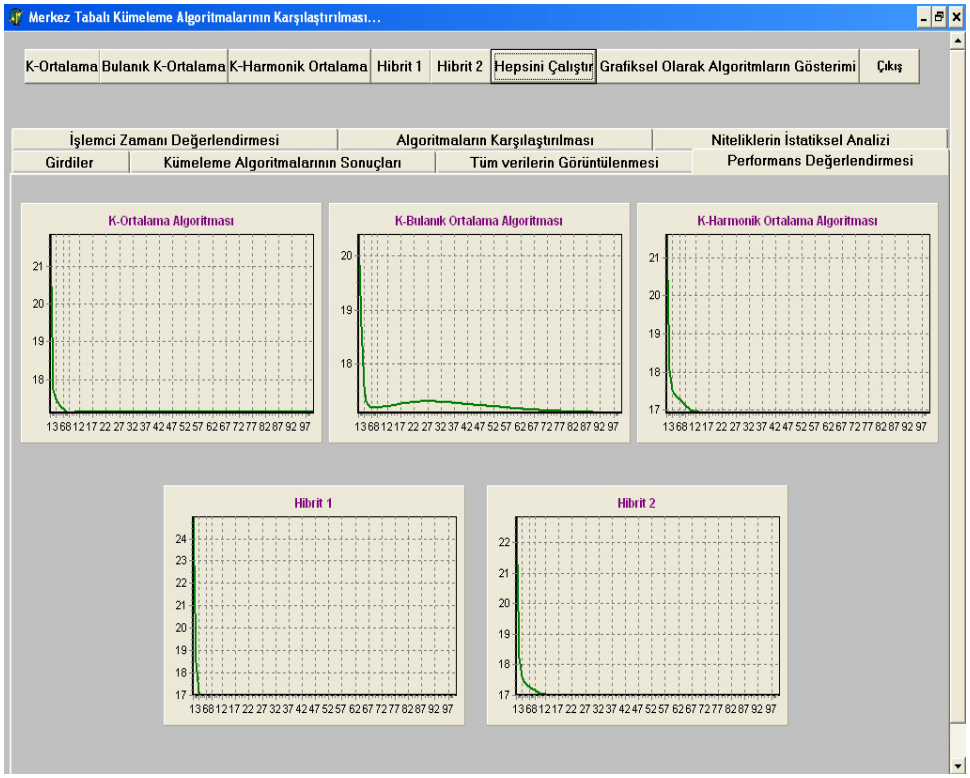


Şekil 5.22: Süsen veritabanı üzerine uygulanan algoritmaların işlemci zamanı değerlerinin görsel olarak sunumu.

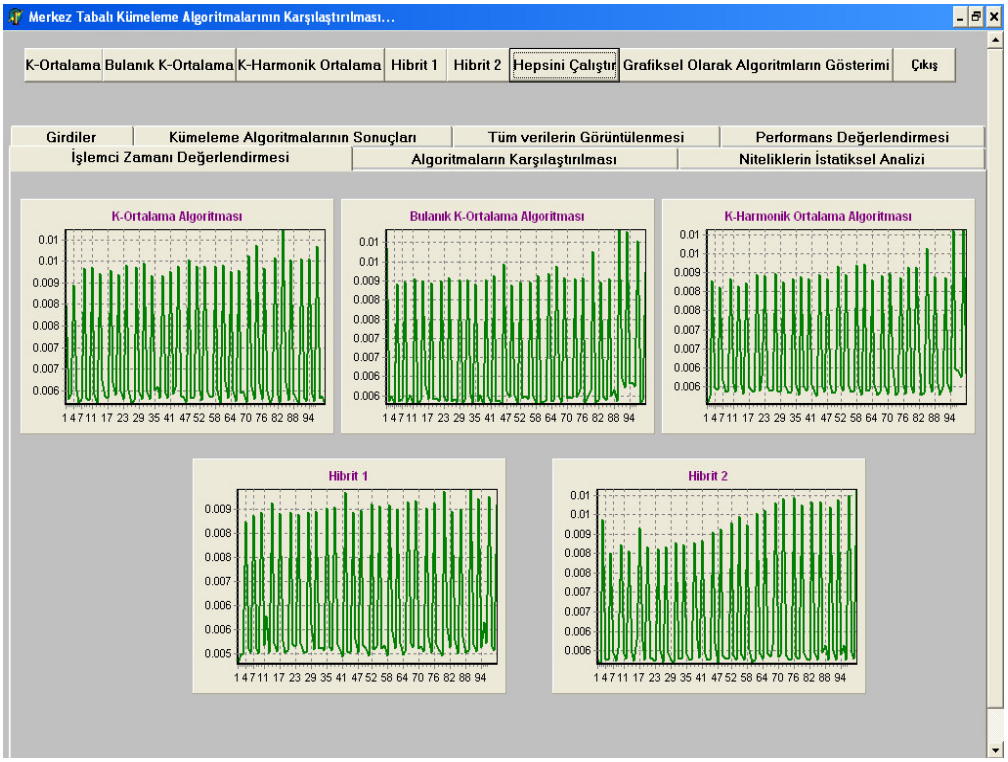
Mamografi veritabanına ait olan sonuçlar incelendiğinde en hızlı çalışan algoritmanın KM olduğu, toplam karesel hata değeri en düşük olan algoritmaların ise KHM, Hibrit 1 ve Hibrit 2 olduğu tespit edilmiştir. Ancak bu karşılaştırma işleminde rasgele başlangıç yöntemi kullanıldığından farklı başlangıç noktalarını seçilmesiyle Hibrit 1'in toplam karesel hata değerinde bir yükseliş olduğu da tespit edilmiştir.

Tablo 5.23: Mamografi veritabanına üzerinde uygulanan merkez tabanlı kümeleme algoritmaların toplam karesel hata ve işlemci zamanına göre karşılaştırılması.

Algoritma	Toplam Karesel Hata	İşlemci Zamanı(s)
KM	17.13	0.618
FKM	17.09	0.922
KHM	16.92	0.636
Hibrit 1	16.99	0.858
Hibrit 2	16.99	1.015



Şekil 5.23: Mamografi veritabanı üzerine uygulanan algoritmaların toplam karesel hata değerlerinin görsel olarak sunumu.



Şekil 5.24: Mamografi veritabanı üzerine uygulanan algoritmaların işlemci zamanı değerlerinin görsel olarak sunumu.

6) Algoritmaların Yakınsama Durumuna Göre Karşılaştırılması: Merkez tabanlı kümeleme algoritmaları yakınsama durumuna göre karşılaştırılmıştır. En uygun toplam karesel hata değerine algoritmaların hangisinin daha hızlı yakınsadığını tespit etmek amacıyla bu karşılaştırma yapılmıştır. Uygulama üzerindeki girdiler sekmesindeki durdurma kıstaslarından merkezler durdurma kıstası temel alınarak algoritmalar karşılaştırılmıştır. Süsen çiçeği veritabanı ve mamografi veritabanı üzerinde yapılan karşılaştırma sonuçları Tablo 5.24 ve Tablo 5.25' de verilmiştir.

Uygulama ile merkez tabanlı kümeleme algoritmaları süsen çiçeği ve mamografi veritabanları üzerinde topla karesel hata değeri ve iterasyon sayısı bakımından karşılaştırılmışlardır. Karşılaştırma işlemlerinde FKM için r değeri 2, KHM, Hibrit 1 ve Hibrit 2 için p değeri 3.5 ve başlangıç yöntemi de Macqueen başlangıç yöntemi olarak alınmıştır. Süsen çiçeği veritabanı ve mamografi veritabanı üzerinde yapılan karşılaştırma sonuçları Tablo 5.24 ve Tablo 5.25' de verilmiştir

Tablo 5.24: Süsen çiçeği veritabanına üzerinde uygulanan merkez tabanlı kümeleme algoritmaların toplam karesel hata ve iterasyon sayısı bakımından karşılaştırılması.

Veritabanı: Süsen çiçeği.			
K Sayısı	**Algoritmalar**	**İterasyon Sayısı**	**Toplam Karesel Hata Değeri**
2	**KM**	10	6.10
	FKM	8	6.14
	KHM	6	6.12
	H1	9	6.18
	H2	7	6.12
3	**KM**	5	5.41
	FKM	11	5.41
	KHM	15	5.41
	H1	7	5.99
	H2	16	5.41
4	**KM**	7	5.18
	FKM	19	5.09
	KHM	15	5.09
	H1	13	5.89
	H2	17	5.09
5	**KM**	6	5.09
	FKM	28	4.96
	KHM	17	4.85
	H1	6	5.88
	H2	32	4.96
6	**KM**	9	5.07
	FKM	63	4.85
	KHM	27	4.85
	H1	11	5.09
	H2	30	4.81
7	**KM**	9	5.04
	FKM	35	4.68
	KHM	33	4.68
	H1	8	5.08
	H2	26	4.69

Tablo 5.25: Mamografi veritabanına üzerinde uygulanan merkez tabanlı kümeleme algoritmaların toplam karesel hata ve iterasyon sayısı bakımından karşılaştırılması.

Veritabanı: Mamografi			
K Sayısı	**Algoritmalar**	**İterasyon Sayısı**	**Toplam Karesel Hata Değeri**
2	**KM**	7	17.90
	FKM	7	17.82
	KHM	7	17.89
	H1	9	17.91
	H2	6	17.95
3	**KM**	6	17.13
	FKM	84	17.10
	KHM	36	16.92
	H1	7	17.05
	H2	36	16.99
4	**KM**	9	16.084
	FKM	23	16.279
	KHM	18	16.081
	H1	8	16.122
	H2	18	16.118
5	**KM**	9	15.79
	FKM	20	15.74
	KHM	19	15.62
	H1	16	15.82
	H2	15	15.65
6	**KM**	9	15.46
	FKM	36	15.04
	KHM	29	14.96
	H1	16	15.49
	H2	34	15.01
7	**KM**	18	15.05
	FKM	37	14.78
	KHM	40	14.73
	H1	25	15.06
	H2	28	14.79

Tablo 24 ve Tablo 25' daki sonuçlar incelendiğinde merkez tabanlı kümeleme algoritmalarından KM algoritmasının diğer algoritmalara göre daha az iterasyonda yakınsadığı görülmektedir. KM algoritmasının yakınsaması diğerlerine göre daha hızlı olmasına rağmen kümeleme sonucunda toplam karesel hata değeri yüksek çıkabilir. Toplam karesel hata değerinin yüksek olması iyi bir kümelemenin olmadığı anlamına gelir. KM algoritmasının hızlı yakınsamasına rağmen iyi bir kümeleme yapabilmesi için başlangıç noktalarının düzgün verilmesi gerekir. Başlangıç noktalarının da düzgün verilmesi ile KM algoritması hem iyi bir kümeleme yapar hem de hızlı yakınsadığından işlemci zamanı da düşük olur. Tablolar incelendiğinde diğer algoritmaların iterasyon sayısı sürekli değişmektedir. Bazen algoritmalar en yüksek bazen de en düşük iterasyon değerine sahip olmaktadır. FKM algoritması genelde en yüksek iterasyon sayısına sahip olmaktadır. Onu genelde yüksekten küçüğe doğru KHM, Hibrit 2 ve Hibrit 1' e ait olan iterasyon değerleri takip etmektedir.

6. SONUÇLAR

Kümeleme, heterojen olan büyük bir grubu homojen olan alt gruplara ya da kümelere ayırma işlemidir. Kümeleme de amaç küme içi benzerliğin maksimum kümeler arasındaki benzerliğin ise minimum olmasıdır. Bir kümelemede olması gereken belli başlı özellikler vardır. Bunlar; ölçeklenebilir olmalı, farklı nesne tipleri ile çalışabilmeli, düzgün şekilli olmayan kümeleri de bulabilmeli, en az miktarda giriş değişkeni gerektirmeli, gürültü içeren verileri de kullanabilmeli, çok boyutlu veritabanları ile çalışabilmeli ve kolay yorumlanabilen sonuçlar üretebilmelidir.

Kümeleme işlemine başlanmadan önce kullanılacak olan algoritmanın özelliklerinin çok iyi bilinmesi ve uygulanacak veriye uygun olup olmadığının kararının uzman tarafından verilmesi gerekmektedir. Bu nedenlerle kümeleme algoritmalarının popüler bir sınıfı olan merkez tabanlı kümeleme algoritmalarının karşılaştırılması üzerine bir tez çalışması yapılmıştır. Bu yapılan çalışma ile merkez tabanlı kümeleme algoritmalarının davranışlarının iyi bir şekilde analiz edilmesi ve bu algoritmaları kullanacak olan uzmanların algoritmaların avantajlarını ve dezavantajlarını bilerek bu algoritmaları tercih etmesi amaçlanmıştır.

Bu çalışmada merkez tabanlı kümeleme algoritmaları olan KM, FKM ve KHM algoritmaları ve KM ve KHM' nin özelliklerini içinde barındıran Hibrit 1 ve Hibrit 2 adındaki algoritmalar bir kümeleme analizinde olması gereken kıstaslar doğrultusunda karşılaştırılmışlardır. Algoritmalar başlangıç durumuna duyarlılık, k küme sayısının kümelemeye etkisi, verinin boyutunun az ya da çok olması, aykırı değerlerin kümelemeye etkisi ve algoritmaların topla karesel hata ve işlemci zamanı ve yakınsama durumu kıstaslarına göre karşılaştırılmışlardır.

İlk kıstasa göre karşılaştırma yapıldığında KHM, Hibrit 2' nin Macqueen, rasgele ve rasgele bölümleme başlangıç yöntemleri ile seçilen merkez noktalarından etkilenmedikleri tespit edilmiştir. FKM algoritması da KHM ve Hibrit 2 gibi

başlangıçta seçilen noktalara çok fazla duyarlı değildir fakat bazen bu noktalardan da etkilenip bu noktaların etkisinde olan kümeler oluşturabilmektedir. Hibrit 1 ve KM algoritmaları ise başlangıçta seçilen merkez noktaları çerçevesinde kümeler oluşturmaktadırlar. Farklı başlangıç noktalarının verilmesi durumunda farklı kümeler oluşturmaktadırlar. Dolayısıyla KHM, Hibrit 2 ve FKM başlangıçta seçilen noktalardan etkilenmezken, Hibrit 1 ve KM ise başlangıçta seçilen noktalardan etkilenmektedir.

İkinci karşılaştırma kıstası başlangıçta kullanıcı tarafından karar verilen k küme sayısıdır. Bu uygulama ile k sayısının değişiminden merkez tabanlı kümeleme algoritmalarının kümeleme sonuçlarının nasıl etkilendiği araştırılmıştır. k sayısı oluşan kümeler üzerinde doğrudan etkilidir. Çünkü verilerin kaç tane kümeye ayrılacağını bu k sayısı belirler. Fakat yapılan kümelemenin iyi bir sonuca sahip olması için en uygun k sayısına karar verilmesi gerekir. k sayısının artışı ile toplam karesel hata değeri de azalmaktadır. Hangi k değerinin daha iyi olduğuna karar vermede toplam karesel hata değeri dikkate alınır. Bir k değerinden diğer k+1 değerine geçerken bu iki k değerlerine ilişkin toplam karesel hata değerlerinin farkı alınır. Eğer fark diğer k değerleri arasındaki fark değerinden daha büyük ise k+1 değerinin en uygun k değeri olduğuna karar verilir. Yapılan karşılaştırma işleminde merkez tabanlı kümeleme algoritmaları için en uygun k değerinin genellikle 3 olduğuna karar verilmiştir.

Üçüncü karşılaştırma kıstasına göre merkez tabanlı kümeleme algoritmaları karşılaştırıldığında veri boyutunun az ya da çok olmasından etkilendikleri saptanmıştır. Veri boyutunun çok olması durumunda daha fazla nitelik değeri kümeleme işlemine dâhil olacağından doğal olmayan kümeler üretilebilmektedir. Boyut sayısının artması ile birlikte algoritmaların toplam karesel hata değerleri de artmış ve kümelerin sayıca büyüklükleri ve içerikleri eşit çıkmamıştır. Yapılan karşılaştırma işlemi sonucu özellikle KHM, Hibrit 2 ve FKM boyut artışından olumsuz yönde etkilendiği, KM ve Hibrit 1' in ise boyut artışından diğerleri kadar etkilenmediği saptanmıştır.

Dördüncü karşılaştırma kıstasına göre algoritmaların aykırı değerler karşısında nasıl bir performansa sahip oldukları incelenmiş ve birbirleri ile karşılaştırılmıştır. Oluşan kümeler sayıca aynı büyükte olup olmama durumuna ve toplam karesel hata değerlerine göre incelenmiştir. Bu karşılaştırma işlemi için veritabanları içindeki kayıtlardan birinin değerleri ile oynanıp diğer kayıtlardan olabildiğince sıra dışı bir kayıt olması sağlanmıştır. Üzerinde oynanmış kayıtları içermeyen veritabanı ile kayıtlarından biri sıradışı yapılmış aynı veritabanı karşılaştırılarak aykırı değerlerin kümeleme üzerindeki etkisi ölçülmeye çalışılmıştır. Karşılaştırma da başlangıç yöntemi olarak Macqueen başlangıç yöntemi kullanılmıştır. Bunun nedeni başlangıç noktalarının değişiminden kümelerin etkilenmesini önlemek ve kümelerin sadece sıra dışı bir kaydın olması durumunda etkilenilip etkilenmediğini araştırmaktır. Rasgele ve rasgele bölümleme başlangıç yöntemlerinde başlangıç merkezleri rasgele oluşturulduğu için farklı sayıda elemanlara sahip kümelerin oluşması durumunda bunun başlangıç yönteminde mi yoksa kayıtların bir tanesinin sıra dışı olmasından mı kaynadığını anlamamız zor olacağından karşılaştırma da Macqueen başlangıç yöntemi kullanılmıştır. Yapılan karşılaştırma sonuçları Öklit ve Manhattan uzaklık ölçümü kategorisi altında ele alınmıştır. Bunun nedeni ise iki uzaklık ölçümünün sıra dışılıklar karşısında verdikleri tepkilerinde bu tez kapsamında incelenmek istenmesidir. Sonuçta merkez tabanlı kümeleme algoritmaları bu uzaklık ölçülerini hesaplamalarda kullandıkları için bu uzaklık ölçümlerinin sıra dışılıklardan etkilenmesi direkt olarak oluşacak kümeler üzerine yansıyacaktır. Yapılan karşılaştırma sonuçları incelendiğinde merkez tabanlı kümeleme algoritmalarının aykırı değer bir tane olsa bile bundan etkilendikleri saptanmıştır.

Beşinci karşılaştırma kıstasına göre algoritmalar toplam karesel hata değerine ve işlemci zamanına göre karşılaştırılmışlar ve sonuç olarak toplam karesel hata değeri en düşük çıkan algoritmalar KHM, Hibrit 2 ve FKM oluştur. Bu onların iyi bir kümeleme yaptıklarını göstermektedir. İşlemci zamanına göre en hızlı çalışan algoritma KM olmuştur.

Altıncı karşılaştırma kıstası olan yakınsama durumuna göre de algoritmalar karşılaştırılmıştır. Bu karşılaştırma işlemi sonucunda KM algoritmasının diğer algoritmalara göre çok hızlı yakınsadığı fakat her zaman en iyi kümelemeyi

yapamadığı tespit edilmiştir. Bunun nedeni ise başlangıçta seçilen noktalara duyarlı olmasıdır. FKM algoritması karşılaştırma sonuçları değerlendirildiğinde genelde en yüksek iterasyon sayısına sahip olduğundan en yavaş yakınsayan algoritma olmuştur. Onu genelde yüksekten küçüğe doğru KHM, Hibrit 2 ve Hibrit 1' e ait olan iterasyon değerleri takip etmektedir. Bu verilen kıstaslardaki sonuçlar dikkate alınarak kullanıcı amacına göre en uygun olan merkez tabanlı kümeleme algoritmasını seçilebilir.

KAYNAKLAR

[1] Ayre, L., B., "Data Mining for Information Professionals", (2006).

[2] Witten, I., H., Frank, E., "Data Mining Practical Machine Learning Tools and Techniques", Second Edition, Cerra, D., ***Morgan Kaufmann Publishers,*** (2005).

[3] Fayyad, U., Shapiro, G., P., Smyth, P., "From Data Mining to Knowledge Discovery in Databases", ***American Association for Artifical Intelligence***,(1996)

[4] Akpınar, H., "Business Intelligence&Data Mining", (2004).

[5] Cooley, R., Srivastava, J., "Web Mining: Information and Pattern Discovery on the World Wide Web".

[6] Zhou, Z., H., "Three Perspectives of Data Mining".

[7] "Introduction to Data Mining and Knowledge Discovery", Third Edition, ***Two Crows Corporation,*** (2005).

[8] Breault, J. L., "Data Mining Diabetic Databases: Are Rough Sets a Useful Addition".

[9] Agrawal, R., Mehta, M., Shafer, J., Srikant, R., Arning, A., Bollinger, T, "The Quest Data Mining System"

[10] Raghavan, V., Hafez, A., "Dynamic Data Mining".

[11] Looney, C., G., "Pattern Recognation", ***CRC Press LLC***, (2003).

[12] Ganti, V., Gehrke, J., Ramakrishnan, "Mining Very Large Databases", ***IEEE***, 1999.

[13] Andreescu, A., "Forecasting Corporate Earnings: A Data Mining Approach", M. Sc. Thesis in Accounting, ***The Swedish School of Economics and Business Administration***, (2004)

[14] Ayramo., S., Karkkainen., T., "Introduction to partitioning-based clustering methods with a robust example", ***Reports of the Department of mathematical Information Technology Series C. Software and Computational Engineering, No. C. 1, Finland***, (2006).

[15] Castro, V., E., "Why so many clustering algorithms-A Position Paper", Volume 4,65-75.

[16] Berkhin, P., "Survey of Clustering Data Mining Techniques", ***Accrue Software Inc***, (2002).

[17] Jain, A., K., Murty, M., N., Flynn, P., J., "Data Clustering: A Review", ***ACM Computing Surveys***, Vol. 31, No. 3, (1999).

[18] Ray., S., Turi, R., H., "Determination of Number of Clusters in K-Means Clustering and Application in Colour Image Segmentation"

[19] Dinçer, E., "Veri Madenciliğinde K-Means algoritması ve tıp alanında uygulanması", Yüksek Lisans Tezi, ***Kocaeli Üniversitesi Fen Bilimleri Enstitüsü***, İzmit, (2006).

[20] Sönmez, İ., Kömüşçü, A., Ü., "K-Ortalamaları Kümeleme Yöntemi İle Türkiye Yağış Bölgelerinin Yeniden Tanımlanması ve Alt-Periyodlardaki Değişimleri", ***"1. Türkiye İklim Değişikliği Kongresi TİKDEK 2007"***, İTÜ, İstanbul, 11-13 Nisan (2007).

[21] Juan, O., Keriven, R., Postelnicu, G., "Stochastic Motion and the Level Set Method in Computer Vision: Stoshactic Active Contours" , ***CERTIS, 04-41***, ***France***, (2004).

[22] Orman, M., G., H., "Particle Swarm Optimazation Methods for Pattern Recognation and Image Processing", Philosophiae Doctor, ***University of Pretoria***, (2005).

[23] Zhang, B., Hsu, M., Dayal, U., "K-Harmonic Means-A Data Clustering Algorithm", ***Hewlett-Packard Company***, (1999).

[24] Zhang, B., "Generalized K-Harmonic Means-Boosting in Supervised Learning", ***Hewlett-Packard Company***, (2000).

[25] Zhang, B., "Dependence of Clustering Algorithm Performance on Clusteredness of Data", ***Hewlett-Packard Research Laboratories.***

[26] Hamerly, G., Elkan, C., "Alternatives to the k-means algorithm that find better clusterings".

[27] Demiralay, M., Çamurcu, A., Y., "Cure, Agnes, ve K-Means Algoritmalarındaki Kümeleme Yeteneklerinin Karşılaştırılması", ***İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi***, 8, 1-18, (2005).

[28] Güngör, Z., Ünler, A., "K-Harmonic Means Data Clustering with Tabu-Searh Method", ***Proceedings of 5th International Symposium on Intelligent Manufacturing Systems***, Sakarya University, Sakarya, 346-360, 29-31 May (2006).

[29] Cano, J., R., Cordon, O., Herrera, F., Sanchez, L., "A greedy randomized adaptive search procedure applied to the clustering problem as an initialization process using K-Means as a local search procedure", **Journal of Intelligent@Fuzzy Systems**, 12, 235-242, (2002).

[30] Belal, M., Daoud, A., "A New Algorithm for Cluster Initialization", Proceedings of World Academy of Science, Engineering And Technology, ***World Enformatika Society***, (2005).

[31] San, O., M., Huynh, V., N., Nakamori, Y., "An Alternative Extension Of The K-Means Algorithm For Clustering Categorical Data", ***Int. J. Appl. Math. Comput. Sci***., Vol. 14, No. 2, 241-247, (2004).

[32] Fahim, A., M., Salem, A., M., Torkey, F., A., Ramadan, M., A., "An efficient enhanced k-means clustering algorithm",1626-1633, (2006).

[33] Han, X., Zhao, T., "Auto-K Dynamic Clustering Algorithm", ***Asian Journal of Information Technology***, 448-451, (2005).

[34] Pelleg, D., Moore, A., "X-Means: Extending K-Means with Efficient Estimation of the Number of Clusters".

[35] Barbakh, W., Fyfe, C., "Performance Functions and Clustering Algorithms"

[36] Toledo, M., D., G., "A Comparison in Cluster Validation Techniques", Master of Science, ***University of Puerto Rico Mayagüez Campus***, (2005)

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

[38] Roiger, R., J., Geatz, M., W., "Data Mining-A Tutorial-Based Primer", ***Addison Wesley***, (2003).

[39] Firestone, J., M., "Data Mining and KDD: A Shifting Mosaic", ***Executive Information Systems Inc.***, (1997).

[40] Hand, D., Manila, H., Smyth, P., "Principles of Data Mining", ***MIT Press***, (2001).

[41] Berry, M., J., A., Linoff, G., S., "Data Mining Techniques for Marketing, Sales, and Customer Relationship Management", Second Edition, ***Wiley Publisching***, (2004).

[42] Tan, P., N., Steinbech, M., Kumar, V., "Introduction To Data Mining", ***Addison-Wesley***, (2006).

[43]http://people.revoledu.com/kardi/tutorial/Similarity/index.html, **(Ziyaret Tarihi: 20 Nisan 2007).**

[44] Wagholikar, A., S., "Acquisition of Fuzzy Measures in Multicriteria Decision Making Using Similarity-based Reasoning", PhD Thesis, ***Griffith University***, Australia, (2007)

[45] Aydoğan, F., "E-Ticarette Veri Madenciliği Yaklaşımlarıyla Müşteriye Hizmet Sunan Akıllı Modüllerin Tasarımı ve Gerçekleştirimi", Yüksek Lisans Tezi, Hacettepe Üniversitesi, ***Hacettepe Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, (2003).

[46]http://www.server.bcc.ac.uk/oncology/MicroCore/HTML_resource/KMeans_Algo2.htm/, **(Ziyaret Tarihi: 2 Nisan 2007).**

[47] Blei, D., "Review of Clustering and K-Means", (2007).

[48] http://people.revoledu.com/kardi/tutorial/kMean/NumericalExample.htm (**Ziyaret Tarihi: 10 Mayıs 2007**)

[49] Karaboğa, D., "Yapay Zeka Optimizasyon Algoritmaları", 1. Baskı, ***Atlas Yayın Dağıtım***, (2004)

[50] Kanungo, T., Mount, D., M, Netanyahu, N., S., Piatko, C., Silveran, R., Wu, A. Y., "The Analysis of a Simple k-eans Clustering Algorithm", ***ACM***, 2006

[51] Erdoğan, Ş., Z., Timor, M., "Data Mining Application in A student Database", *Journal of Aeronautics and space Technologies*, Vol. 2, No. 2, 53–57, (2005).

[52] Wiwattanacharoenchai, S., Srivihok, A., "Data Mining of Electronic Banking in Thailand: Usage Behavior Analysis by Using K-Means Algorithm"

[53] Elmas, Ç., "Bulanık Mantık Denetleyiciler(Kuram, Uygulama, Sinirsel Bulanık Mantık)", 1. Baskı, ***Seçkin yayıncılık San. Ve Tic. A.Ş.***, (2003)

[54] Duru, N., "Bulanık Mantık Temelli Gürültü Azaltma Sistemi", Doktora Tezi, ***Kocaeli Üniversitesi Fen Bilimleri Enstitüsü***, İzmit, (1997).

[55] Duda, R., O., Hork, P., O., Stork, D., G., "Pattern Classification", Second Edition.

[56] http://www.usyd.edu.au/su/agric/acpa/fkme/FkME.html#Algorithm, **(Ziyaret Tarihi: 23 Nisan 2007).**

[57] Hoey, P., S., "Statistical Analysis of the Iris Flower Dataset", (2003).

[58] http://archive.ics.uci.edu/ml/datasets.html, **(Ziyaret Tarihi: 10 Ocak 2007).**

ÖZGEÇMİŞ

1983 yılında Bulgaristan' da doğdu. İlk, orta ve lise öğrenimini Bursa' da tamamladı. 2001 yılında girdiği Kocaeli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü' den 2005 yılında Bilgisayar Mühendisi olarak mezun oldu. 2005 yılından beri Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Bölümü' nde Yüksek Lisans' a devam etmektedir.