

T.C.
MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK ANABİLİM DALI

VERİ MADENCİLİĞİNDE BAZI KÜMELEME
ALGORİTMALARININ KARŞILAŞTIRILMASI

YÜKSEK LİSANS

ALİ BURAK SAYILAN

AĞUSTOS 2019
MUĞLA

T.C.
MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK ANABİLİM DALI

VERİ MADENCİLİĞİNDE BAZI KÜMELEME
ALGORİTMALARININ KARŞILAŞTIRILMASI

YÜKSEK LİSANS

ALİ BURAK SAYILAN

AĞUSTOS 2019
MUĞLA

MUGLA SITKI KOÇMAN ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

TEZ ONAYI

ALİ BURAK SAYILAN tarafından hazırlanan **VERİ MADENCİLİĞİNDE BAZI KÜMELEME ALGORİTMALARIN KARŞILAŞTIRILMASI** başlıklı tezinin, 29/08/2019 tarihinde aşağıdaki jüri tarafından İstatistik Anabilim Dalı'nda yüksek lisans derecesi için gerekli şartları sağladığı oybirliği/oyçokluğu ile kabul edilmiştir.

TEZ SINAV JURİSİ

Prof. Dr. Erol EĞRİOĞLU (Jüri Başkanı)
İstatistik Anabilim Dalı,
Giresun Üniversitesi, Giresun

İmza:



Prof. Dr. Dursun AYDIN (Danışman)
İstatistik Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

İmza:



Doç. Dr. Nevin Güler DİNÇER (Üye)
İstatistik Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

İmza:



ANA BİLİM DALI BAŞKANLIĞI ONAYI

Prof. Dr. Dursun AYDIN
İstatistik Anabilim Dalı Başkanı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

İmza:



Prof. Dr. Dursun AYDIN
Danışman, İstatistik Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

İmza:



Savunma Tarihi: 29/08/2019

Tez çalışmalarım sırasında elde ettiğim ve sunduğum tüm sonuç, doküman, bilgi ve belgelerin tarafımdan bizzat ve bu tez çalışması kapsamında elde edildiğini; akademik ve bilimsel etik kurallarına uygun olduğunu beyan ederim. Ayrıca, akademik ve bilimsel etik kuralları gereği bu tez çalışması sırasında elde edilmemiş başkalarına ait tüm orijinal bilgi ve sonuçlara atıf yapıldığını da beyan ederim.

Ali Burak Sayılan

29/08/2019



ÖZET
VERİ MADENCİLİĞİNDE BAZI KÜMELEME ALGORİTMALARININ
KARŞILAŞTIRILMASI

Ali Burak SAYILAN

Yüksek Lisans Tezi

Fen Bilimleri Enstitüsü

İstatistik Anabilim Dalı

Danışman: Prof. Dr. Dursun AYDIN

Ağustos 2019, 111 sayfa

Kümeleme çözümlemesi, gruplanmamış verileri benzerliklerine göre gruplar veya kümeler altında toplayan çok değişkenli bir istatistiksel yöntemdir. Kümeleme veri madenciliğinde de yaygın olarak kullanılan bir tekniktir. Bu çalışmada amaç, kümeleme çözümlemesinde kullanılan bazı algoritmaların farklı karakteristik özelliklere sahip veri setleri üzerinde uygulanması ve elde edilen sonuçlar üzerinden algoritmaların performanslarının karşılaştırılmasıdır. Çalışma kapsamında AGNES, DIANA, k-ortalamar, k-medoids, fuzzy c-ortalama, CLARA, DBSCAN, CURE, PAM, DENCLUE, STING, CLIQUE kümeleme algoritmaları incelenmiştir. Seçilen algoritmaların uygulanmasında R istatistiksel programlama dili kullanılmıştır.

Anahtar Sözcükler: Kümeleme, Algoritma, AGNES, k-ortalama, CLARA, DBSCAN

ABSTRACT
COMPARISON OF SOME CLUSTERING ALGORITHMS IN DATA MINING

Ali Burak SAYILAN

Master of Science (M.Sc.)

Graduate School of Natural and Applied Sciences

Department of Statistics

Supervisor: Prof. Dr. Dursun AYDIN

August 2019, 111 pages

Clustering analysis is a multivariate statistical technique that groups or aggregates the ungrouped data by their similarities. Clustering is also a commonly used technique in data mining. The aim of this study is implementation of some clustering algorithms on the data sets which have different characteristics and comparing the algorithm performances by attained results. AGNES, DIANA, k-means, k-medoids, fuzzy c-means, CLARA, DBSCAN, CURE, PAM, DENCLUE, STING, CLIQUE clustering algorithms were researched in the study. R statistical programming language was used for the implementation of the selected algorithms.

Keywords: Clustering, Algorithm, AGNES, k-means, CLARA, DBSCAN



Sevgili Anne ve Babama

ÖNSÖZ

Danışmanım Sayın Prof. Dr. Dursun AYDIN 'a çalışmalarım süresince yönlendiriciliği, katkıları, yardımları, bilimsel çalışmanın ve düşünmenin temellerini bana öğrettiği için teşekkür ederim.

Bu çalışmamın temelini lisans eğitimimde bana öğreten, bu konuyla tanışmama ve sevmeme vesile olan, lisans eğitimim boyunca da yardımlarını esirgemeyen Dokuz Eylül Üniversitesi hocalarımdan Sayın Dr. Öğr. Üye. Engin YILDIZTEPE' ye şükranlarımı sunarım.

Yardımlarından dolayı Dokuz Eylül Üniversitesi İstatistik Bölümünden sınıf arkadaşlarım olan Berkay SARIKAYA ve Rıdvan KOÇ' a, ayrıca desteklerinden dolayı da nişanlım Figen TEKER' e teşekkür ederim.

Son olarak tüm hayatım boyunca maddi, manevi yardım ve desteklerini hiç esirgemeyen, bugünlere gelmemde en büyük pay sahipleri olan sevgili babam Atacan SAYILAN ve sevgili annem Demet SAYILAN' a minnettarlığımı sunmayı borç bilirim.

İÇİNDEKİLER

ÖNSÖZ.....	vii
İÇİNDEKİLER	viii
ÇİZELGELER DİZİNİ	xi
ŞEKİLLER DİZİNİ	xiii
KISALTMALAR DİZİNİ	xv
1. GİRİŞ	1
2. VERİ MADENCİLİĞİ	3
2.1. Veri Madenciliğinin Tanımı	3
2.2. Veri Madenciliği Tarihi	4
2.3. Veri Madenciliği Uygulama Alanları	4
2.4. Veri Madenciliğinin Süreçleri	5
2.5. Veri Madenciliği Modelleri	6
2.5.1. Tahmin edici modeller	7
2.5.1.1. Sınıflama.....	7
2.5.1.2. Regresyon	8
2.5.2. Tanımlayıcı modeller	8
2.5.2.1. Kümeleme	8
2.5.2.2. Birliktelik kuralları.....	9
3. KÜMELEME ANALİZİ	10
3.1. Kümeleme Analizine Giriş	10
3.2. Kümeleme Sürecinin Adımları	12
3.3. Kümeleme Analizi Uygulamaları	13
3.4. Küme Tanımları.....	14
3.4.1. İyi ayrılmış kümeler	14
3.4.2. Merkeze dayalı kümeler	14
3.4.3. En yakın komşu kümeler	15
3.4.4. Yoğunluğa dayalı kümeler	15
3.4.5. Benzerliğe dayalı kümeler	16
4. STANDARTLAŞTIRMA VE UZAKLIK ÖLÇÜLERİ	17

4.1. Verilerin Standartlaştırılması.....	17
4.2. Uzaklık ölçüleri	19
4.2.1. Öklid uzaklık ölçüsü	19
4.2.2. Manhattan uzaklık ölçüsü	20
4.2.3. Pearson uzaklık ölçüsü.....	21
4.2.4. Minkowski uzaklık ölçüsü	22
4.2.5. Mahalanobis uzaklık ölçüsü.....	22
4.2.6. Hotelling T^2 uzaklık ölçüsü	23
5. KÜMELEME YÖNTEMLERİ VE ALGORİTMALARI.....	24
5.1. Bölünmeli Kümeleme Yöntemi.....	25
5.1.1. K-ortalamlar (K-means) algoritması	26
5.1.2. K-medoids algoritması.....	29
5.1.3. PAM algoritması.....	29
5.1.4. CLARA algoritması	31
5.2. Hiyerarşik Kümeleme Yöntemi.....	32
5.2.1. AGNES algoritması	34
5.2.2. DIANA algoritması.....	35
5.2.3. CURE algoritması.....	36
5.3. Yoğunluğa Dayalı Kümeleme Yöntemi	38
5.3.1. DBSCAN algoritması	39
5.3.2. DENCLUE algoritması.....	42
5.4. Bulanık Kümeleme Yöntemi	43
5.4.1. Fuzzy c-means algoritması	45
5.5. Grid Tabanlı Kümeleme Yöntemi	47
5.5.1. STING algoritması.....	47
5.5.2. CLIQUE algoritması.....	49
6. UYGULAMA.....	50
6.1. Giriş	50
6.2. İki Değişkenli Sentetik Veri Setleri Uygulaması.....	50
6.2.1. Aggregation veri seti.....	51
6.2.2. Compound veri seti	56
6.2.3. Sentetik 1 veri seti.....	61

6.2.4. Flame veri seti	65
6.2.5. Pathbased 2 spiral veri seti.....	69
6.2.6. Jain veri seti	73
6.2.7. Pathbased 1 veri seti.....	77
6.2.8. D31 veri seti	81
6.2.9. R15 veri seti	83
6.3. Çok Değişkenli Gerçek Veri Seti Uygulaması	85
6.3.1. Seeds veri seti	85
6.3.2. Materyal ve yöntem	85
6.3.2.1. Değişken ölçümlerinin elde edilmesi.....	85
6.3.2.2. Temel bileşen analizi	85
6.3.3. Kümeleme sonuçları	87
6.4. Yöntemlerin Performansları Arasındaki Farkların Anlamlılığı.....	95
7. SONUÇ.....	97
KAYNAKLAR	99
EKLER.....	102
Ek A. R Kodları	102
Ek B. Gerçek Veri Seti	106
ÖZGEÇMİŞ.....	111

ÇİZELGELER DİZİNİ

Çizelge 5.1. Fuzzy c-means örnek kümesindeki nesnelerin üyelik değerleri	46
Çizelge 6.1. Aggregation veri seti için algoritmaların doğruluk oranları	51
Çizelge 6.2. Aggregation veri seti için DBSCAN algoritmasının karışıklık matrisi .	53
Çizelge 6.3. Aggregation veri seti için AGNES algoritmasının karışıklık matrisi	53
Çizelge 6.4. Aggregation veri seti için PAM algoritmasının karışıklık matrisi.....	54
Çizelge 6.5. Aggregation veri seti için k-ortalamalar algoritmasının karışıklık matrisi.....	54
Çizelge 6.6. Aggregation veri seti için DIANA algoritmasının karışıklık matrisi.....	55
Çizelge 6.7. Aggregation veri seti için CLARA algoritmasının karışıklık matrisi	55
Çizelge 6.8. Compound veri seti için algoritmaların doğruluk oranları	56
Çizelge 6.9. Compound veri seti için DBSCAN algoritmasının karışıklık matrisi ...	58
Çizelge 6.10. Compound veri seti için k-ortalamalar algoritmasının karışıklık matrisi.....	58
Çizelge 6.11. Compound veri seti için DIANA algoritmasının karışıklık matrisi.....	59
Çizelge 6.12. Compound veri seti için AGNES algoritmasının karışıklık matrisi	59
Çizelge 6.13. Compound veri seti için CLARA algoritmasının karışıklık matrisi	60
Çizelge 6.14. Compound veri seti için PAM algoritmasının karışıklık matrisi	60
Çizelge 6.15. Sentetik 1 veri seti için algoritmaların doğruluk oranları	61
Çizelge 6.16. Sentetik 1 veri seti için DIANA, AGNES ve CLARA algoritmalarının karışıklık matrisleri.....	63
Çizelge 6.17. Sentetik 1 veri seti için PAM algoritmasının karışıklık matrisi.....	63
Çizelge 6.18. Sentetik 1 veri seti için DBSCAN algoritmasının karışıklık matrisi ...	64
Çizelge 6.19. Sentetik 1 veri seti için k-ortalamalar algoritmasının karışıklık matrisi.....	64
Çizelge 6.20. Flame veri seti için algoritmaların doğruluk oranları	65
Çizelge 6.21. Flame veri seti için DBSCAN algoritmasının karışıklık matrisi	67
Çizelge 6.22. Flame veri seti için CLARA algoritmasının karışıklık matrisi.....	67
Çizelge 6.23. Flame veri seti için PAM algoritmasının karışıklık matrisi.....	67
Çizelge 6.24. Flame veri seti için k-ortalamalar algoritmasının karışıklık matrisi....	68
Çizelge 6.25. Flame veri seti için AGNES algoritmasının karışıklık matrisi	68
Çizelge 6.26. Flame veri seti için DIANA algoritmasının karışıklık matrisi.....	68
Çizelge 6.27. Pathbased 2 spiral veri seti için algoritmaların doğruluk oranları	69

Çizelge 6.28. Pathbased 2 spiral veri seti için DBSCAN ve AGNES algoritmalarının karışıklık matrisleri.....	71
Çizelge 6.29. Pathbased 2 spiral veri seti için CLARA algoritmasının karışıklık matrisi.....	71
Çizelge 6.30. Pathbased 2 spiral veri seti için PAM algoritmasının karışıklık matrisi.....	71
Çizelge 6.31. Pathbased 2 spiral veri seti için k-ortalamalar algoritmasının karışıklık matrisi.....	72
Çizelge 6.32. Pathbased 2 spiral veri seti için DIANA algoritmasının karışıklık matrisi.....	72
Çizelge 6.33. Jain veri seti için algoritmaların doğruluk oranları.....	73
Çizelge 6.34. Jain veri seti için DBSCAN algoritmasının karışıklık matrisi.....	75
Çizelge 6.35. Jain veri seti için DIANA algoritmasının karışıklık matrisi.....	75
Çizelge 6.36. Jain veri seti için CLARA algoritmasının karışıklık matrisi.....	75
Çizelge 6.37. Jain veri seti için AGNES algoritmasının karışıklık matrisi.....	76
Çizelge 6.38. Jain veri seti için k-ortalamalar algoritmasının karışıklık matrisi.....	76
Çizelge 6.39. Jain veri seti için PAM algoritmasının karışıklık matrisi.....	76
Çizelge 6.40. Pathbased 1 veri seti için algoritmaların doğruluk oranları.....	77
Çizelge 6.41. Pathbased 1 veri seti için DBSCAN algoritmasının karışıklık matrisi.....	79
Çizelge 6.42. Pathbased 1 veri seti için DIANA algoritmasının karışıklık matrisi ...	79
Çizelge 6.43. Pathbased 1 veri seti için CLARA algoritmasının karışıklık matrisi...	79
Çizelge 6.44. Pathbased 1 veri seti için k-ortalamalar algoritmasının karışıklık matrisi.....	80
Çizelge 6.45. Pathbased 1 veri seti için PAM algoritmasının karışıklık matrisi.....	80
Çizelge 6.46. Pathbased 1 veri seti için AGNES algoritmasının karışıklık matrisi...	80
Çizelge 6.47. D31 veri setinin karışıklık matrisleri sonuçları.....	81
Çizelge 6.48. R15 veri setinin karışıklık matrisleri sonuçları.....	83
Çizelge 6.49. Seeds veri seti için CLARA algoritmasının karışıklık matrisi.....	89
Çizelge 6.50. Seeds veri seti için DIANA algoritmasının karışıklık matrisi.....	90
Çizelge 6.51. Seeds veri seti için k-ortalamalar algoritmasının karışıklık matrisi.....	91
Çizelge 6.52. Seeds veri seti için fuzzy c-means algoritmasının karışıklık matrisi...	92
Çizelge 6.53. Seeds veri seti için PAM algoritmasının karışıklık matrisi.....	93
Çizelge 6.54. Seeds veri seti için AGNES algoritmasının karışıklık matrisi.....	94
Çizelge 6.55. Aggregation veri seti için algoritmaların doğruluk oranları.....	52

ŞEKİLLER DİZİNİ

Şekil 2.1. Veri madenciliğinin süreçleri.....	5
Şekil 2.2. Veri madenciliği modelleri	7
Şekil 3.1. Kümelere ayrılmış veriler	10
Şekil 3.2. Kümeleme sürecinin adımları	12
Şekil 3.3. İyi ayrılmış kümeler	14
Şekil 3.4. Merkeze dayalı kümeler.....	14
Şekil 3.5. En yakın komşu kümeler	15
Şekil 3.6. Yoğunluğa dayalı kümeler	15
Şekil 4.1. İki birim arasındaki uzaklık	19
Şekil 4.2. Manhattan ve Öklid uzaklık ölçüleri	20
Şekil 5.1. Kümeleme yöntemleri ve algoritmaları	24
Şekil 5.2. Bölünmeli yöntem akış diyagramı	25
Şekil 5.3. K-ortalamlar algoritması işlem basamakları	28
Şekil 5.4. PAM metoduyla yapılmış kümeleme örneği	30
Şekil 5.5. Hiyerarşik kümeleme yöntemi dendrogramı	32
Şekil 5.6. Birleştirici hiyerarşik kümeleme.....	33
Şekil 5.7. Ayrıştırıcı hiyerarşik kümeleme	33
Şekil 5.8. CURE algoritmasının genel çalışma şekli	37
Şekil 5.9. Yoğunluk tabanlı veri	40
Şekil 5.10. DBSCAN algoritması	41
Şekil 5.11. Farklı şekil ve büyüklükte kümelere oluşan veri seti.....	42
Şekil 5.12. Üyelik fonksiyonu.....	44
Şekil 5.13. Fuzzy c-means örnek kümeleme.....	45
Şekil 5.14. STING algoritmasının hiyerarşik yapısı	48
Şekil 6.1. Aggregation veri seti.....	51
Şekil 6.2. Aggregation veri seti için kümeleme sonuçları	52
Şekil 6.3. Compound veri seti.....	56
Şekil 6.4. Compound veri seti için kümeleme sonuçları.....	57
Şekil 6.5. Sentetik 1 veri seti.....	61
Şekil 6.6. Sentetik 1 veri seti için kümeleme sonuçları	62
Şekil 6.7. Flame veri seti.....	65

Şekil 6.8. Flame veri seti için kümeleme sonuçları	66
Şekil 6.9. Pathbased 2 spiral veri seti.....	69
Şekil 6.10. Pathbased 2 spiral veri seti için kümeleme sonuçları	70
Şekil 6.11. Jain veri seti	73
Şekil 6.12. Jain 2 spiral veri seti için kümeleme sonuçları	74
Şekil 6.13. Pathbased 1 veri seti.....	77
Şekil 6.14. Pathbased 1 veri seti için kümeleme sonuçları	78
Şekil 6.15. D31 veri seti.....	81
Şekil 6.16. D31 veri seti için kümeleme sonuçları.....	82
Şekil 6.17. R15 veri seti	83
Şekil 6.18. R15 veri seti için kümeleme sonuçları.....	84
Şekil 6.19. PCA (Principal Component Analysis).....	86
Şekil 6.20. Seeds veri setinde PCA uygulaması	87
Şekil 6.21. Seeds veri seti	88
Şekil 6.22. Seeds veri seti için CLARA algoritmasının kümeleme sonuçları	89
Şekil 6.23. Seeds veri seti için DIANA algoritmasının kümeleme sonuçları	90
Şekil 6.24. Seeds veri seti için k-ortalamlar algoritmasının kümeleme sonuçları ...	91
Şekil 6.25. Seeds veri seti için fuzzy c-means algoritmasının kümeleme sonuçları..	92
Şekil 6.26. Seeds veri seti için PAM algoritmasının kümeleme sonuçları	93
Şekil 6.27. Seeds veri seti için AGNES algoritmasının kümeleme sonuçları.....	94

KISALTMALAR DİZİNİ

PAM	Partitioning Around Medoids
CLARA	Clustering for Large Applications
CLARANS	Clustering for Large Applications Based Upon Randomized Search
AGNES	Agglomerative Nesting
DIANA	Divisive Analysis
DBSCAN	Density Based Spatial Clustering of Applications with Noise
CURE	Clustering Using Representatives
DENCLUE	Density based Clustering
STING	Statistical Information Grid
OPTICS	Ordering Points To Identify the Clustering Structure
PCA	Principal Component Analysis

1. GİRİŞ

Günümüzde artan veri sayısı, iletişim ve bilgisayar teknolojilerindeki gelişmeler çok daha fazla verinin kısa sürelerde depolanıp işlenmesine ve bilgiye dönüştürülmesine imkânı sunmaktadır. Teknolojide gerçekleşen tüm bu gelişmeler ve veri havuzlarındaki hızlı artışlar, toplanan bu verilerin nasıl bilgiye dönüştürülüp ne şekilde faydalanılacağını ve tüm bu verilerden ne şekilde anlamlar çıkarmamız gerektiği sorununı doğurmuştur. Son yıllarda yoğun olarak kullanılmaya başlayan veri madenciliği, tüm bu sorunların çözümlerine ulaşmamıza yardımcı olan bir bilim dalı olarak karşımıza çıkmaya başlamıştır.

Veri madenciliği, çok büyük hacimli verilerin içerisinde bazı faydalı kararlar alabilmek için uygulanabilir ve anlamlı bilgilerin çıkarılmasıdır. Tek başına çözüm olmayan veri madenciliği, mevcut problemlerin çözümü ve geleceğe yönelik tahminler yapabilmek adına gerekli olabilecek bilgileri çıkarmamızı sağlayan önemli bir araç niteliğindedir. Elde edilen bu bilgiler önceden bilinmeyen fakat potansiyel olarak kullanışlı bilgilerdir.

Veri madenciliği, verileri baştan aşağıya taramakla başlayıp, yapılan analizler ardından ortaya çıkarılan sonuçları konunun uzmanları tarafından değerlendirilmesi ile sonlanır. Veri madenciliğinin alt yapısını büyük çoğunlukla veri tabanı uygulamaları ve istatistik bilimi oluşturur. Veri madenciliğinin istatistik biliminin yanı sıra ilgilendiği başlıca diğer disiplinler ise; veri tabanı teknolojisi, desen tanıma, makine öğrenmesi, görselleştirme ve kümelemedir.

Kümeleme yöntemi, veri madenciliğinde yaygın olarak kullanılan çok değişkenli bir istatistiksel yöntemdir. Hedef değişken belirlenmesine gerek duymadığı için denetimsiz (unsupervised) yöntemlerdendir. Kümeleme analizinin başlıca görevi, belli özellikleri barındıran verilerin benzerliklerini bulmak ve bu benzerlikler temelinde verileri doğru bir şekilde kategorize etmektir. Analiz bu yönüyle çok değişkenli analiz tekniklerinden biri sayılan diskriminant analiziyle benzerlik göstermektedir. Ancak aralarındaki en büyük fark, diskriminant analizinin aksine

kümeleme analizinde birimlerin anlık durumları gözlenmesi nedeniyle geleceğe dair çıkarımlar yapmak mümkün değildir. Kümeleme analizini kullanma amaçları genel itibarıyla şöyle sıralanabilir;

1. Bir sınıflama sağlamak,
2. Hipotez testi yapmak ya da bir veri seti içerisinde diğer yöntemlerle oluşturulan kategorileri ortaya çıkarmak,
3. Doğru grupların belirtilmesi,
4. Model uydurmayı basitleştirmek,
5. Grup ön tahminlerini yapmak,
6. Verilerin yapılarının belirlenmesi,
7. Verilerin indirgenmesi,
8. Aykırı değerlerin (outliers) bulunması.

Kümeleme algoritmalarının, yetenek ve kullanılabilirlik bakımından karşılaştırması birçok literatür çalışması içerisinde yer almaktadır. Bu çalışmalardan bazıları bir yöntemi esas alarak diğer algoritmalarla uygulamalı kıyaslamalar üzerinden yöntemi açıklamaya çalışırken, bazıları da uygulama yapmadan algoritmaların avantaj ve dezavantajlarını sıralayarak birden çok kümeleme algoritması karşılaştırmasına yer vermiştir.

Bu çalışmada, veri madenciliğinde yaygın olarak kullanılan her bir kümeleme yönteminden en az bir algoritmayı ele alarak incelenmesi ve farklı özellikteki hem sentetik hem de gerçek verileri barındıran veri setleri üzerindeki performanslarının karşılaştırılması hedeflenmiştir. Çalışma yedi bölümden oluşmaktadır. İkinci, üçüncü, dördüncü ve beşinci bölümlerde sırasıyla veri madenciliği, kümeleme analizi, uzaklık ölçüleri ve kümeleme algoritmaları konularına giriş yapılarak tanımlamalar yapılmıştır. Altıncı bölümde yer alan uygulamada, dokuz yapay veri seti ve bir gerçek veri seti kullanılarak seçilen kümeleme algoritmaları denenmiştir. Yedinci ve son bölümde ise sonuçlar tartışılmıştır.

2. VERİ MADENCİLİĞİ

2.1. Veri Madenciliğinin Tanımı

Veri madenciliği; her geçen gün artan kullanışlı veya kullanışsız veriler, dijital veri elde etme ve depolama teknolojisindeki gelişmeler, ileriye dönük araştırmalar yapmak için geçmiş verilerin elde tutulmasına olan mecburiyet gibi birçok nedenden dolayı son yıllarda büyük önem kazanmış ve sıklıkla adı geçen bir bilim dalı haline dönüşmüştür. Bu güne kadar farklı alanlara ait çeşitli kaynaklarda pek çok farklı tanımlamayla açıklanmaya çalışan veri madenciliğinin literatürdeki tanımlardan bazılarına bakmak gerekirse;

- Hand (1998) veri madenciliği tanımını, istatistik, veri tabanı teknolojileri, makine öğrenimi, örüntü tanıma ile etkileşimi olan yeni bir bilim dalı ve büyük veritabanlarında öncesinde tahminde bulunulamayan durumların ikincil analizi olarak yapmıştır.
- Önceden bilinmeyen, fakat yarar sağlama potansiyeli yüksek olan verinin keşfedilmesine veri madenciliği denir (Frawley ve Piatetsky-Shapiro, 1991).
- Doğan ve Türkoğlu' da (2007) veri madenciliğini, geniş veri yığınları içerisinden geleceğe dair öngörüler yapabilmemizi sağlayan bağlantıların, bilgisayar programları kullanılmasıyla aranması işidir olarak tanımlamışlardır.
- Büyük miktardaki veriden anlamlı örüntü ve kurallar çıkarılması amacıyla analiz edilmesine veri madenciliği denir (Berry ve Linoff, 2000).
- Jacobs' a (1999) göre ise veri madenciliği, başlangıçtaki datanın kendi başına çıkaramadığı bilgiyi sunmayı sağlayan, veri analizi süreçleridir.

2.2. Veri Madenciliği Tarihi

Matematikçilerin 1950'lerde, bilgisayar bilimleri, mantık disiplinlerindeki yapay zekâ ve makine öğrenimi konularında çalışmaya başlaması veri madenciliğine giden yolun taşlarının döşenmeye başlandığı yıllardır (Adriaans ve Zantinge, 1997). Veri tabanı ve bilgi teknolojileri 1960'lı yıllardan bu yana basit dosyalama işlemlerinden geniş güçlü veri tabanları sistemine doğru sistemli bir şekilde gelişmiştir. Bu gelişim 1970'li yıllardan sonra ilişkisel veri tabanlarının ortaya çıkmasına sebep olmuştur. Veri tabanlarındaki tüm bu gelişimler 1980'li yılların sonlarına doğru veri madenciliğinin ortaya çıkmasını sağlamıştır (Han ve Kamber, 2000). 1990'lı yıllarda veri tabanlarında yaşanan köklü değişiklikler, özellikle de ilişkisel veri tabanlarının hızlı bir şekilde gelişmeye başlaması sayesinde, yeni oluşmaya başlayan ve kullanım alanlarının çok kısıtlı olduğu veri ambarları ve veri madenciliği kavramlarının içleri dolmaya ve de kullanım alanları artmaya başlamıştır. 2000'li yıllara gelindiğinde ise artık veri madenciliği uygulamaları, bilgiyi ve teknolojiyi içinde barındıran tüm alanlarda kullanıldığı hatta bazı alanlarda vazgeçilemez ve hayati bir öneme sahip olduğu, tanınmanın ötesinde gerekliliğinin vurgulandığı, her geçen gün önemini giderek arttırdığı yıllar olmuştur.

2.3. Veri Madenciliği Uygulama Alanları

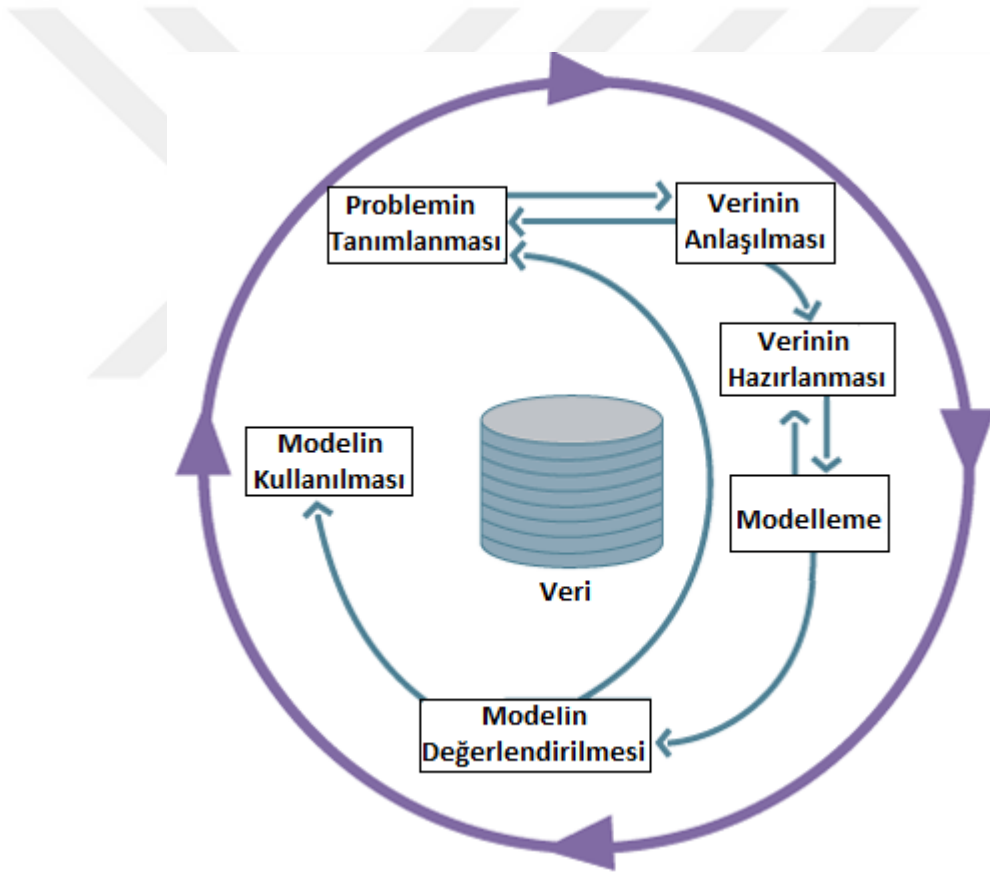
Çok büyük hacimli veriler bulunduran ve karar verilme sürecine ihtiyaç olan her durumda veri madenciliği uygulanabilir. Kaynaklara bakıldığında en fazla bankacılık, finans, telekomünikasyon, pazarlama, sağlık, genetik ve biyoloji sektörleri veri madenciliğini uygulayan sektörler olarak öne çıkmaktadır. Müşteri ilişkileri yönetimi, firma satışları, eğitim, uzay bilimleri teknolojileri, meteoroloji, coğrafi bilgi sistemleri, kimya sektörleri de her aşamasında veri madenciliğini çokça kullanan diğer sektörlerdir. Aynı zamanda suçlarla mücadele alanlarında gizli dinleme ve vergi kaçakçılıklarının ortaya çıkartılması, sigortacılıkta riskli müşteri örüntülerini belirlenmesi gibi tüm bu alanlar ve bunun gibi birçok alan içinde özellikli durumlarda da sıklıkla kullanılmaktadır (Savaş vd., 2012; Baykal, 2006).

2.4. Veri Madenciliğinin Süreçleri

Veri madenciliği genel olarak bir süreçten ibarettir. Veri setleri içerisinde bulunan ham bilginin işlenip işe yarayan birer bilgiye dönüşmesini sağlayan tüm adımlar birbirlerine sıkı sıkıya bağlı olan her bir dişli, süreç ise bütünü oluşturan çark gibidir.

Aynı zamanda veri madenciliği, bilginin keşfi sürecinde örüntüleri ayrıştırıp süzüp daha sonra hazır duruma getirerek bilginin keşfi sürecinin de önemli bir basamağını oluşturur.

Veri madenciliği sürecini oluşturan adımlar genel hatlarıyla (Şekil 2.1.)’de verilmiştir.



Şekil 2.1. Veri madenciliğinin süreçleri

Problemin tanımlanması: Veri madenciliğinin ilk adımı olarak, cevabı aranan problem net bir biçimde ifade edilerek tanımlanan problemin çözümü bir amaç olarak ortaya konulmalıdır. Aynı zamanda hatalı tahminler sonucu doğacak olan

maliyetler ve başarılı tahminlerin sağlayacağı faydalara ilişkin çıkarımlar bu adım içerisinde değinilmelidir (Akpınar, 2000).

Verilerin anlaşılması: Bu adımda, mevcut problemin çözümü için güvenilir kaynaklardan elde edilen verilerin, hatalı değerleri ve aralarındaki tutarsızlıklarını anlamamız ayrıca bu verilerin ne anlam ifade ettiklerinin anlaşılması verilerin hazırlanması aşamasının doğru yapılabilmesi için oldukça önemlidir.

Verilerin hazırlanması: Farklı kaynaklardan sağlanan tüm verilerin birleştirilmesi, hatalı verilerin çıkarılıp eksik verilerin giderilmesi, verilerin uygun formlara dönüştürülmesi işlemlerini içeren veri madenciliğinin bu adımında veriler, işlenmeden önceki son şeklini alırlar.

Modelleme: Verilerin bilgiye dönüştürülmesi için birtakım tekniklerin kullanıldığı için veri madenciliğinin en önemli adımıdır. Uygun tekniğin seçimi veri seti tipine göre yapıldığı gibi bazen de veri seti tiplerinin dönüştürülmesi söz konusu olduğundan modelleme ve verilerin hazırlanması adımları geri dönüşümlü adımlardır (Albayrak ve Yılmaz, 2009)

Modelin değerlendirilmesi: Teknik değerlendirmeleri içeren bu süreç, uygulanan tekniklerin sonuçlarının belli kriterlere göre değerlendirildiği bir veri madenciliği sürecidir.

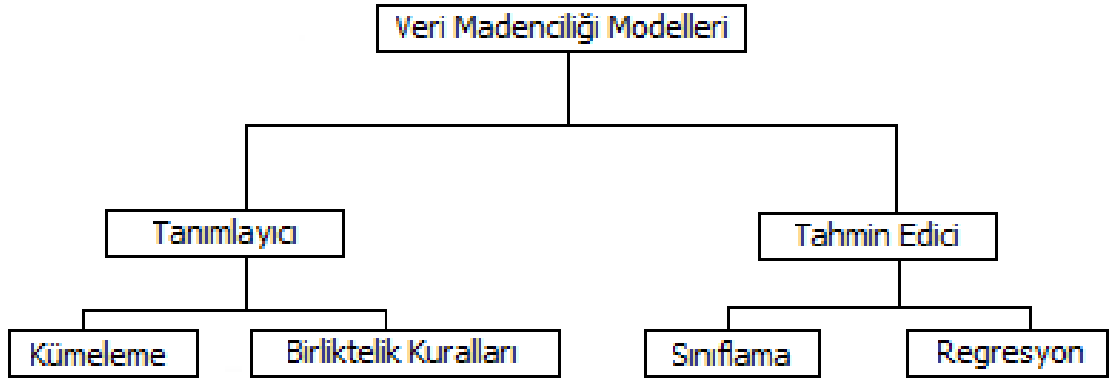
Modelin kullanılması: Değerlendirildikten sonra kurulan ve geçerliliği kabul gören model, doğrudan bir uygulama olabileceği gibi, farklı bir uygulamanın aşaması olarak da kullanılabilir.

2.5. Veri Madenciliği Modelleri

Veri madenciliğinde kullanılan birçok metodun yanında gün geçtikçe yeni modeller ve algoritmalar keşfedilmektedir. Bu modellerden bazıları yıllardır uygulanan klasik istatistiksel yöntemlerdir. Bu klasik yöntemler dışındaki yöntemler de yine istatistik temelli olmakla beraber yapay zekâ ve makine öğrenimi disiplinlerini içeren yeni nesil yöntemlerdir.

Veri madenciliği modellerini, işlevlerine göre tanımlayıcı (descriptive) ve tahmin edici (predictive) olmak üzere iki temel başlık altında ele almak mümkündür (Şekil

2.2.). Birliktelelik kuralları ve kümeleme modelleri tanımlayıcı; regresyon ve sınıflama modelleri tahmin edici modellerdir.



Şekil 2.2. Veri madenciliği modelleri

2.5.1. Tahmin edici modeller

Tahmin edici modellerde, sonuçları bilinen verileri kullanarak elde edilen modeller yardımıyla sonuçları bilinmeyen veri kümeleri hakkında tahminlerde bulunulması amaçlanır. Örnek olarak; kredi talebinde bulunan bir banka müşterisi hakkında, önceki kredileriyle ilgili verilerle kurulacak olan modelle, bankanın müşterisine vereceği kredinin geri dönüp dönemeyeceği tahmininde bulunmasını bir tahmin edici modeldir (Özekeş, 2003).

2.5.1.1. Sınıflama

Sınıflama, veri serinde yer alan verilerin ortak özelliklerine göre, daha önceden tanımlanmış sınıflara ayrıştırılmaları için uygulanan bir modeldir. Veri setinin özelliğine göre oluşturulacak olan algoritmalar kullanılarak veriler sınıflara ayrıştırılır.

Sınıflama modelinde kullanılan başlıca teknikler şunlardır:

1. Karar ağaçları
2. Yapay sinir ağları
3. K- en yakın komşu algoritması

2.5.1.2. Regresyon

Regresyon, probleme göre bir ya da birden çok olabilen ve bağımsız olan değişkenlerden oluşan girdilerle, sonuç değerini veren ve bağımlı değişken olan çıktıları ilişkilendirmeyi amaçlayan modeldir. Sonuç değeri genellikle bir güven aralığı içerisinde açıklanır (Çelik, 2013).

Regresyon modelleri, doğrusal (basit veya çoklu) ve lojistik regresyon olmak üzere ikiye ayrılır. Doğrusal regresyon, normal dağılım göstermesi gereken sürekli veya kesikli olan nicel verilerin ilişkilerini açıklarken lojistik regresyon ise normal dağılımın aranmadığı kategorik (evet/hayır, var/yok, 0/1) verilerin, neden sonuç ilişkisini ortaya koyar ve sonuç değerini olasılık olarak elde eder.

2.5.2. Tanımlayıcı modeller

Tanımlayıcı modeller, karar vermeye rehberlik etmesi için eldeki verilerin özelliklerini tanımlamaya yarayan modelleridir. Örneğin; aynı meslek grubuna mensup insanların sosyal hayatlarındaki etkinliklerinin birbirleriyle benzerlik gösterdiği görülürken, başka bir meslek grubundaki insanlarla ise farklılıklar göstermesinin tespiti bir tanımlayıcı model örneği olarak gösterilebilir.

2.5.2.1. Kümeleme

Kümeleme işlemi nesnelerin kendilerini veya başka nesneler ile aralarındaki ilişkileri anlatan bilgiler aracılığıyla nesneleri gruplandırma işlemidir. Kümelemenin amacı; grup içinde yer alan nesneleri kendi içlerinde mümkün olduğunca benzer, diğer gruplarda yer alan nesnelerden de mümkün olduğunca farklı olacak şekilde gruplara ayırabilmektir (Zaiane vd., 2002).

Kümeleme algoritmalarının sınıflama algoritmalarından en önemli farkı, kümelemenin önceden tanımlı verilere dayalı olmamasıdır. Sınıflamada tanımlı veriler ve bu verilerin önceden almış olduğu değerlerin üzerine bir temel model oluştururken, kümelemede önceden tanımlanmış veri ve örnek mevcut değildir. Veriler kendi aralarında benzerliklerine göre gruplanırlar. Benzerlikleri tanımlayan değer ve özellikler model kurulurken araştırmacı tarafından öngörülür (Argüden ve Erşahin, 2008).

2.5.2.2. Birliktelik kuralları

Birliktelik kuralları, büyük veri setleri içerisinde veriler arası ilişkileri bir kural halinde açıklayan, olayları beraber gerçekleşmesi ihtimallerini geçmiş verileri analiz ederek geleceğe yönelik tahminlerde bulunan bir modeldir. Bu modelin en yaygın uygulaması, perakende satış sektöründe müşterilerin satın alma eğilimlerini tanımlayan pazar sepeti uygulamasıdır. Örnek olarak: diş macunu ve cilt kremi alan bir müşterinin yüksek bir ihtimalle parfüm alması ya da ekmek ve süt alan bir müşterinin yüksek bir ihtimalle yumurta alması birliktelik kuralları modeli örneğidir.

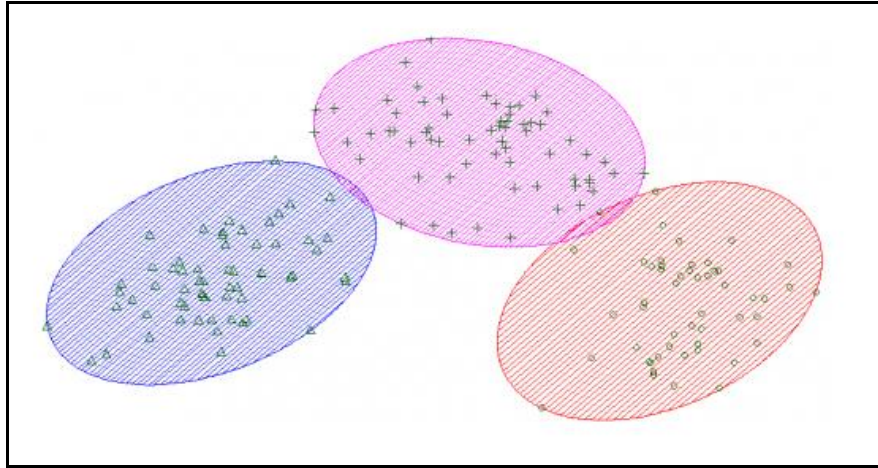
Birliktelik kuralları algoritmalarından ise en yaygın olarak kullanılanı apriori algoritmasıdır.



3. KÜMELEME ANALİZİ

3.1. Kümeleme Analizine Giriş

İstatistik uygulamaları çok fazla veriye ihtiyaç duymakta ve bu durum istatistiksel tekniklerin kullanılmasında zorluklar yaratmaktadır. Dolayısıyla tekniklerin kullanılabilirliklerini artırmak için verileri gruplandırmaya ihtiyaç duyulur. Bu noktada devreye giren kümeleme analizleri son yıllarda bilim ve iş alanlarında oldukça sık kullanılan çok değişkenli veri analizi yöntemlerinin başında gelen bir analiz yöntemidir. Kümeleme analizindeki amaç, veri setleri içerisinde yer alan her bir veri noktasının diğer veri noktalarıyla olan benzerliklerine ya da farklılıklarına göre alt kümeler oluşturmaktır. Bir başka ifadeyle; verilerin aralarındaki benzerlikler dikkate alınarak benzer verileri aynı küme içerisinde toplayarak farklı kümeler oluşturmayı sağlamaktır.



Şekil 3.1. Kümelere ayrılmış veriler

Kümeleme analizi, çeşitli problemler karşısında cevap bulmaya yarayan önemli bir tekniktir. Araştırmacı için, veri yığınları arasından özetleyici bilgiler çıkmasına yardımcı olan kümeleme analizi aynı zamanda bölünen gruplarla çalışan araştırmacılar, hem maliyet hem zaman hem de iş yükü bakımından ciddi kazanımlar elde etmiş olurlar.

Kümeleme adına yapılmış literatürdeki tanımlara bakmak gerekirse;

- Kümeleme analizi, asıl amacı nesnelerin barındırdığı olduğu karakteristik özelliklerine bakarak gruplandırma yapan çok değişkenli analiz tekniğidir. (Akın, 2008).
- Kümeleme analizi, bir araştırma içerisinde incelenmekte olan verilerin aralarındaki benzerliklere göre belli kümeler içerisine toplayarak sınıflandırmayı, verilerin ortak özelliklerini belirlemek ve bu kümelerin genel tanımlamalarını yapabilmeyi sağlayan yöntemidir (Sharma, 1996).

Buradaki tanımlara göre her bir küme, içerlerinde barındırdığı nesneleri en iyi ifade edebilecek şekilde düzenlenmelidir. Kümeleme eğer gerçekten başarılı bir biçimde sonuçlandırılmışsa, geometrik yapılacak bir çizimle nesnelerin kümeleri içerisinde birbirleriyle çok yakın, kümelerin ise birbirlerinden uzak oldukları görülecektir (Hair vd., 1995). Başlangıçta küme sayısı ya da kümelerin yapılarına dair bir varsayım ortaya koymaz. Kümeleme analizleri numerik, kategorik ya da hem numerik hem de kategorik veriler üzerinde uygulanabilmektedir.

Kümeleme analizi, nesneleri p değişkenine göre hesaplanmış birtakım benzerlik ölçülerini kullanarak birbirlerine benzer kümeler oluşturmak amacıyla yapılır. Bu amaçları dört grup olarak aşağıdaki gibi toplayabiliriz (Güler, 2006):

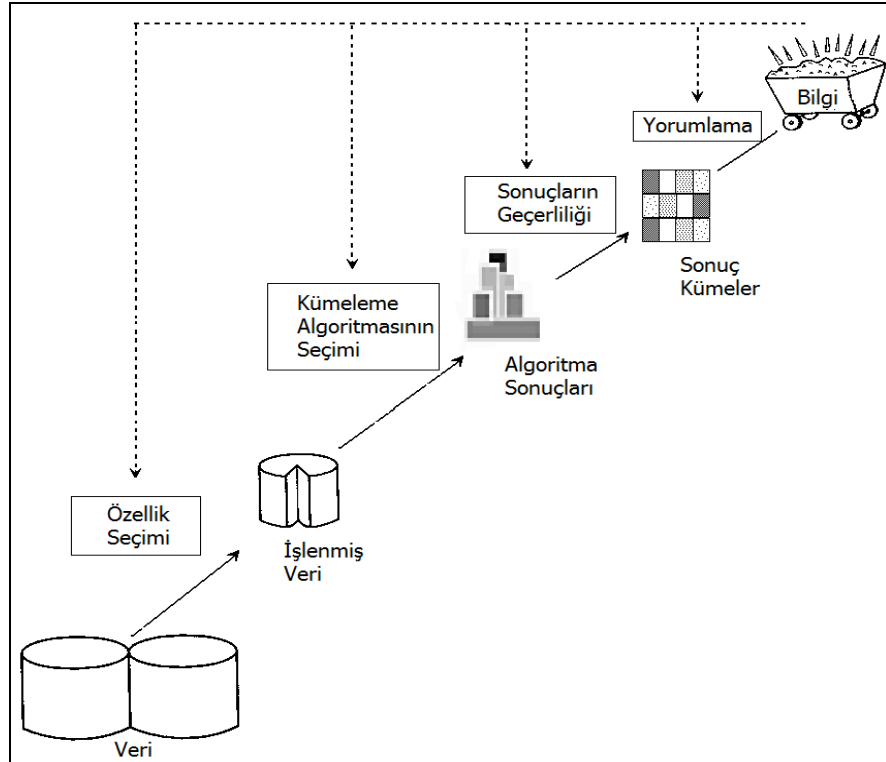
- n sayıda nesneyi, veriyi, birimi, p değişkenine göre belirlenen özelliklere göre kendi içlerinde en homojen, kendi aralarında ise en farklı şekilde alt kümeler oluşturmak.
- p sayıda değişkeni, n sayıda nesnede belirlenen değerler esas alınarak ortak özellikleri ortaya çıkardığını varsayan kümeler oluşturmak ve ortak özellikleri ortaya koymak.
- Hem değişkenleri hem nesneleri beraber ele alarak ortak özelliklere sahip alt kümeler oluşturmak.
- Nesnelerin, p değişkene göre belirlenen değerlere göre tipolojik ve biyolojik sınıflandırılmalarını yapmak.

Kümeleme analizinde başlangıçta küme sayısı ya da kümelerin yapılarına dair bir varsayım ortaya koyulmaz. Genel hatlarıyla uygulama aşamaları aşağıdaki gibi verilebilir:

- Değişken veya nesnelerin doğal kümelerine dair kesinlik olmayan yığınlardan elde edilen n sayıda nesnenin p sayıda değişkenine göre gözlemlerin oluşturulması.
- Değişken veya nesnelerin aralarındaki benzerlikleri ve farklılıkları ortaya koyan bir benzerlik ölçüsü yardımıyla birbirleri ile olan aralarındaki uzaklıkların hesaplanması.
- Uygun olan kümeleme yöntemi uygulanarak benzerlik ve farklılıklara göre değişken veya nesnelerin makul sayıda kümeye bölünmesi.
- Oluşturulan kümelerin yorumlanması ve kümeleme yöntemine göre kurulmuş hipotezlerin sağlamlasının analitik yöntemlerle yapılması (Şahin, 2002).

3.2. Kümeleme Sürecinin Adımları

Kümeleme işleminin süreci, kümeleme yapılırken kullanılacak olan yöntem ve değerlere göre veri yığınlarının farklı şekilde bölünmelerine yol açabilir. Bu nedenle analize başlamadan önce bazı işlemlerin yapılması gerekmektedir (Şekil 3.2.).



Şekil 3.2. Kümeleme sürecinin adımları

Özellik seçimi: İncelenen veri seti hakkında elde edilebilecek en çok bilgiyi içeren kümelemeyle ilgili özellikler doğru seçilmelidir. Bu nedenle veri, kümelemenin adımlarından önce işlenmesi gerekebilir.

Kümeleme algoritmasının seçimi: Elimizdeki verilerin en iyi şekilde kümelenebilmesi esasına dayanan algoritmanın seçilmesidir. Bu adımda nesnelerin birbirleri ile benzerliklerini ölçen uzaklık ölçüleri ve kümeleme kriterleri gibi değerler algoritma seçimi için önemli parametrelerdir.

Sonuçların geçerliliği: Kümeleme analizinde kullanılacak olan algoritmalar, önceliği bilinmeyen kümeler üzerinde çalışacağı için genellikle sonuç kümelenebilmesinin üzerinde birtakım değerlendirmeler yapılması gerekecektir.

Yorumlama: Uygulama alanında doğru sonuca varabilmek için kümeleme sonuçlarının başka deneysel kanıtlarla birlikte ele alınarak değerlendirilmesidir (Güler, 2006).

3.3. Kümeleme Analizi Uygulamaları

Kümeleme analizi uygulamaları, iş ve bilim dallarındaki birçok alanda etkili bir biçimde kullanılmaktadır.

Öncelikle kullanılabileceği basit uygulamaları ele alırsak kümeleme analizi; elde edilmiş bilgi içerikli verinin işlenip sıkıştırılması, verileri ilgilendiren hipotezlerin oluşturulması, oluşturulan hipotezin doğruluğunun sağlanması için hipotez testleri yapılması ve küme içindeki nesnelerin özelliklerini belirten küme oluşumu için gruplara dayalı öngörüler yapması gibi basit uygulamalarda oldukça sık kullanılan metotlardır.

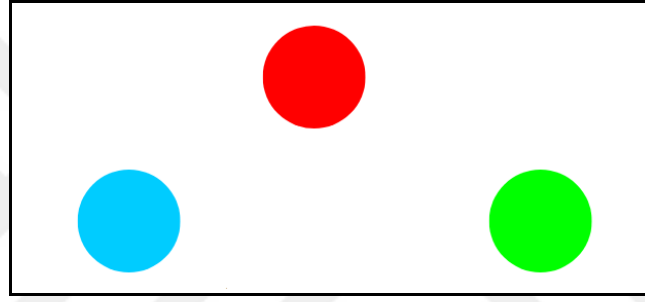
Daha geniş bir bakış açısı ile uygulama alanlarını ele alırsak bu alanlar içerisinde ön plana çıkanlar; örüntü tanımlama, veri analizleri ve veri madenciliği, web madenciliği, resim tanımlama, istatistiki araştırmalar, makine öğrenme, şehir planlamaları, coğrafi analiz (depremler, meteorolojik veriler), uzaysal veri tabanı uygulaması, internet uygulamaları, müşterilerle ilişki yönetimi, sağlık ve biyolojiyi kapsayan araştırmalardır (Jain vd., 1999; Han ve Kamber, 2001; Berkhin, 2002; Bilgin, 2003).

3.4. Küme Tanımları

Kümeleme analizi, farklı küme tanımlarına dayanır. Beş temel küme tanımı vardır (Zaiane vd., 1998).

3.4.1. İyi ayrılmış kümeler

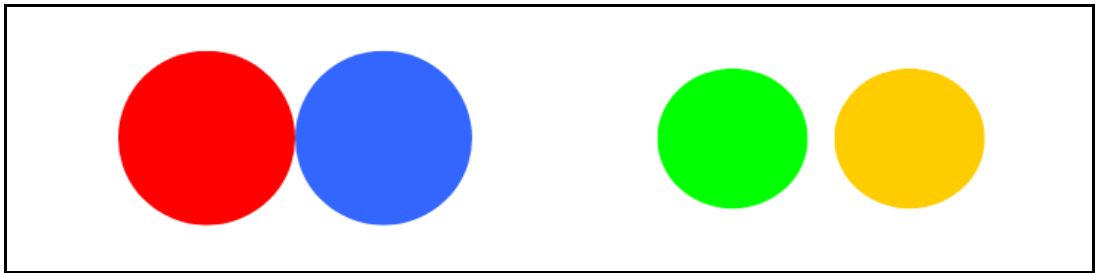
Bir küme içindeki herhangi bir nokta, yine o küme içindeki diğer tüm noktalara, küme dışındaki herhangi bir noktadan daha yakındır veya benzerdir. Kümeler arası farklılıklar veya uzaklıklar bariz olduğu için kümeleme analizi yapılması oldukça kolaydır.



Şekil 3.3. İyi ayrılmış kümeler

3.4.2. Merkeze dayalı kümeler

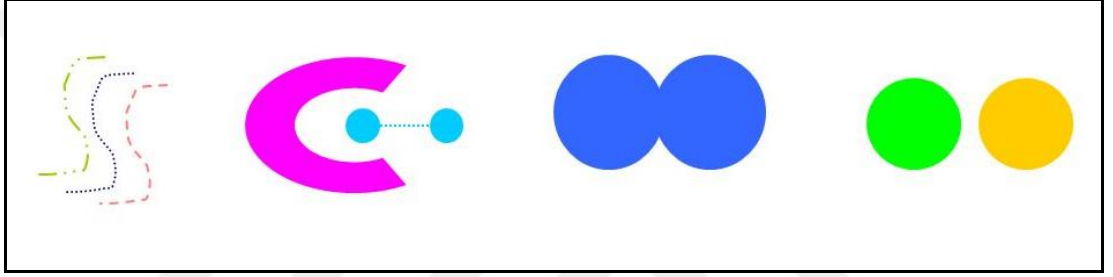
Bir küme içindeki her nokta, o kümenin merkezine, diğer kümelerin merkezlerine olduklarından daha yakındır veya benzerdir. Kümenin merkezi, merkez nokta, centroid veya medoid olarak adlandırılan temsili bir noktadır. Bu temsili noktalar kümeleme analizinde belli matematiksel hesaplamalar sonucunda bulunur ve küme merkezleri olarak kabul edilir. Bir noktanın hangi kümeye ait olacağı, o noktanın temsili noktalara uzaklıklarına veya benzerliklerine göre belirlenir.



Şekil 3.4. Merkeze dayalı kümeler

3.4.3. En yakın komşu kümeler

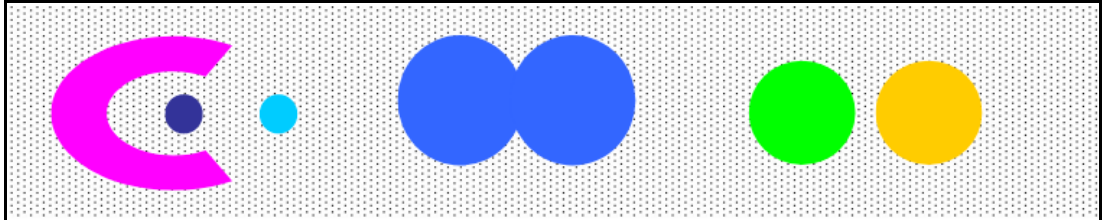
Bir küme içindeki herhangi bir nokta, küme içindeki diğer bir noktaya ya da noktalara başka bir kümedeki noktalardan daha yakındır veya benzerdir. Bu tip kümelerin ele alındığı kümeleme analizleri yapılırken genellikle birleştirme ya da bölme yaklaşımları kullanılır. Birleştirme yaklaşımında, noktaların her biri başlangıçta tek başlarına bir kümeyken daha sonra birbirlerine benzer noktalar ya da gruplar birleştirilir. Bölme yaklaşımında ise tüm noktalar başlangıçta tek bir kümeye aitken daha sonra birbirlerine benzer noktalar ya da gruplar daha küçük alt kümelere ayrıştırılır.



Şekil 3.5. En yakın komşu kümeler

3.4.4. Yoğunluğa dayalı kümeler

Bir küme, noktaların yoğunluklarına göre diğerlerinden ayrılır. Yüksek yoğunluklu alanlar, düşük yoğunluklu alanlar tarafından ayrılır. Veri setinde gürültü ve aykırı gözlem olması durumunda bu tanım önem kazanır. Bu tip kümelerin yer aldığı kümeleme analizinde, komşuluk içindeki yoğunluk belli bir seviyeyi aşana dek kümeler genişlemeye devam etmektedir.



Şekil 3.6. Yoğunluğa dayalı kümeler

3.4.5. Benzerliğe dayalı kümeler

Bir küme, birbirine benzer noktalardan oluşur. Kendi dışındaki kümelerin noktaları, kendi noktalarına benzemez. Bu tanım özellikle noktaların oluşturdukları yoğunluk ya da şekle göre kümeleri farklı olarak tanımlar. Tanımın temeli bir noktanın, kendisiyle aynı kümeye ait diğer nesneler ile maksimum benzerlikte olurken diğer kümelere ait noktalarla ise minimum benzerlikte olmasıdır.



4. STANDARTLAŞTIRMA VE UZAKLIK ÖLÇÜLERİ

4.1. Verilerin Standartlaştırılması

Standartlaştırılma, değişkenlerin her birinden ortalamalarının farkını alarak bu farkı standart sapmaya bölme işlemidir. Bu şekilde elimizdeki her bir ham veri standart verilerden oluşturularak, ölçü birimleri farkından kaynaklanan sorun giderilmiş olacaktır. Bu haliyle veriler normal dağılımın genel şeklini almış olurlar. Başlangıçta elimizde bulunan verileri normalleştirilmesi işleminde Öklid dağılımı ölçümünü kullanmak en faydalı tekniktir. Süreç içerisinde başlangıç verilerinden her biri, 1 değerindeki standart sapmaya ve 0 değerindeki aritmetik ortalamaya sahip bir dağılıma diğer bir ifadeyle de standartlaştırılan değerler skoruna dönüşür. Standartlaştırılan değerlerin değişkenlerle karşılaştırılması çok daha basit olacaktır. Veriler tek ölçek (0 ortalamalı, 1 standart sapmalı) üzerindedirler (Tatlıdil, 2002).

Standartlaştırılmış değişkenlerin kullanımı sadece değişkenlere bağlı olmayan ölçek farklarına bağlı etkileri gerçekten elimine eder. Bununla beraber, araştırmacı sonucu düşünmeden standartlaştırılmış değerleri daima uygulamamalıdır. Standartlaştırılmamış değişkenlere karşı standartlaştırılmış değişkenlerin kullanımı kümeleme çözümünde tamamen kabul görmüş değildir. Eğer değişkenlerin tek bir ölçekte gösterilmesiyle değişkenlerin aralarında tabii bir ilişkisel durum mevcutsa, standartlaştırılma işlemi pek doğru olmayacaktır. Deneysel ve kavramsal etkileşimlerin her biri standartlaştırılma kararı verilmesinde önemli birer kriter olduğu için hiçbir zaman düşünülmeden yapılmamalıdır. Değişkenlerin değerinin standartlaşması değişik şekillerde yapılmakla birlikte genellikle aşağıda bahsedilen teknikler kullanılmaktadır (Servi, 2008).

Z skoruna dönüştürülme: Normal dağılımların sahip olduğu orantılı ya da aralıklı ölçeklerle ölçülmüş verilere uygulanır. En çok kullanılan tekniklerin başında gelir. Z skorlarına dönüştürme tekniğinin genel formülasyonu şu şekildedir;

$$Z_i = \frac{X_i - \bar{x}}{S} \quad (4.1)$$

-1 ≤ x ≤ 1 aralığında dönüştürme: Değerler arasında pozitif ve negatif değerlerin yer aldığı durumlarda, heterojen şekilde yapılan değerlerle aşırı uçlardaki değerlerin yer aldığı durumda kullanımı tercih edilmesi ön görülen tekniktir. Teknikte değişkenlerin her bir değeri maksimum değere oranlanır. $X_{\max}$ dizi içerisindeki en büyük değerdir;

$$x_i = \frac{X_i}{X_{\max}} \text{ şeklindeki formülle dönüşüm yapılır.} \quad (4.2)$$

0 ≤ x ≤ 1 aralığında dönüştürme: Heterojen halde ve/veya aşırı uçtaki değerlere sahip durumlarda, değerler 0 ile +1 arasında değiştirilerek uygulanan tekniğin genel formülasyonu,

$$X_i = \frac{X_i - X_{\min}}{R} \quad (4.3)$$

Burada R; değişim aralığı (Range) olup; $R = X_{\max} - X_{\min}$ olarak belirlenmektedir. Yani değişkenin minimum değere göre farkları değişim aralığının bir yüzdesi olarak elde edilmektedir.

Maksimum değeri 1 yapacak olan dönüştürme: Dizi içerisindeki değerlerden maksimum olanın değeri 1 olması isteniyor ise uygulanacak olan tekniktir.

$$x_i = \frac{X_i}{X_{\max}} \text{ ile uygulanır.} \quad (4.4)$$

Dönüşüm ortalamasını 1 yapacak olan dönüştürme: Dizi ortalaması +1 yapmak istediğimiz durumda uygulanacak olan tekniktir. Dönüştürme;

$$x_i = \frac{X_i - \bar{X}}{X} \text{ ile yapılır.} \quad (4.5)$$

Dönüşüm dizisinin standart sapmasını 1 yapacak olan dönüştürme: Yeni dizinin standart sapmasının 1 olmasını istediğimiz durumlarda uygulanacak olan tekniktir. Dönüştürme;

$$x_i = \frac{X_i}{S} \text{ ile yapılır.} \quad (4.6)$$

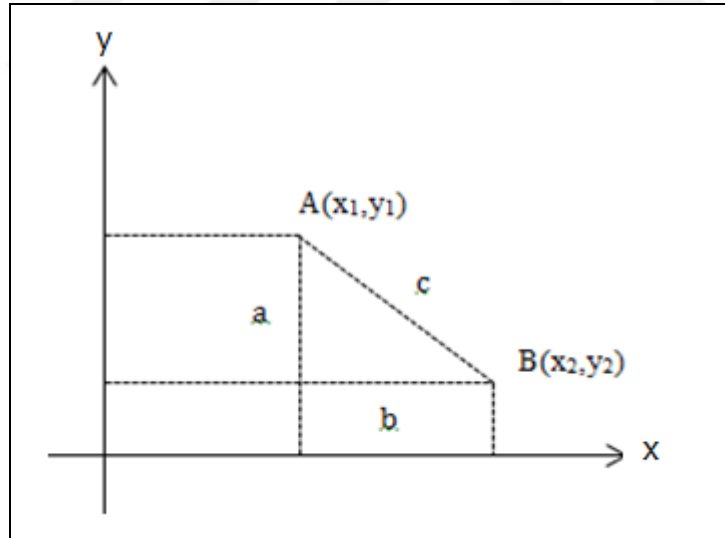
4.2. Uzaklık ölçüleri

Kümeleme analizinde en temel varsayımlardan biri de uzaklık değerlerinin normalliğidir. Kümeleme, uzaklık ve benzerlik gibi kavramlarla ilişkili olduğu için uzaklık ölçülerinden bahsedilmesi gerekmektedir.

Belirli özelliklere sahip nesnelerin, aralarındaki uzaklıkları hesaplamaya yarayan farklı teknikler kullanılmaktadır.

4.2.1. Öklid uzaklık ölçüsü

Öklid uzaklık ölçü birimi (Şekil 4.1.)’deki Pisagor teoreminin uygulaması niteliğinde olmakla birlikte en çok tercih edilen ölçü birimidir.



Şekil 4.1. İki birim arasındaki uzaklık

Öklid ölçüsü genellikle kümeleme alanında daha çok kullanılır. Özellikle nesneler 100’den fazla olduğu durumlarda tercih edilen bir ölçüdür. Çok sık kullanılan ölçü birimi olmasına rağmen değişkenlerin ölçü birimlerine aşırı duyarlı olması en büyük dezavantajıdır. Kısaca, ayırt edici nitelikteki bir değişkenin etkisiyle nesneler arası uzaklıklar, kümeler arası uzaklıklara oranla çok daha fazla olabilir ve bu da

kümeleme analizinin açıklanabilirliğini olumsuz etkiler. O değişkene ait ölçü birimine olan bağımlılık giderilebilse de bu durum başka bazı sorunlara yol açabilir.

Öklid uzaklık ölçüsüyle iki birimin aralarındaki uzaklık,

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{ip} - x_{jp})^2} \quad (4.7)$$

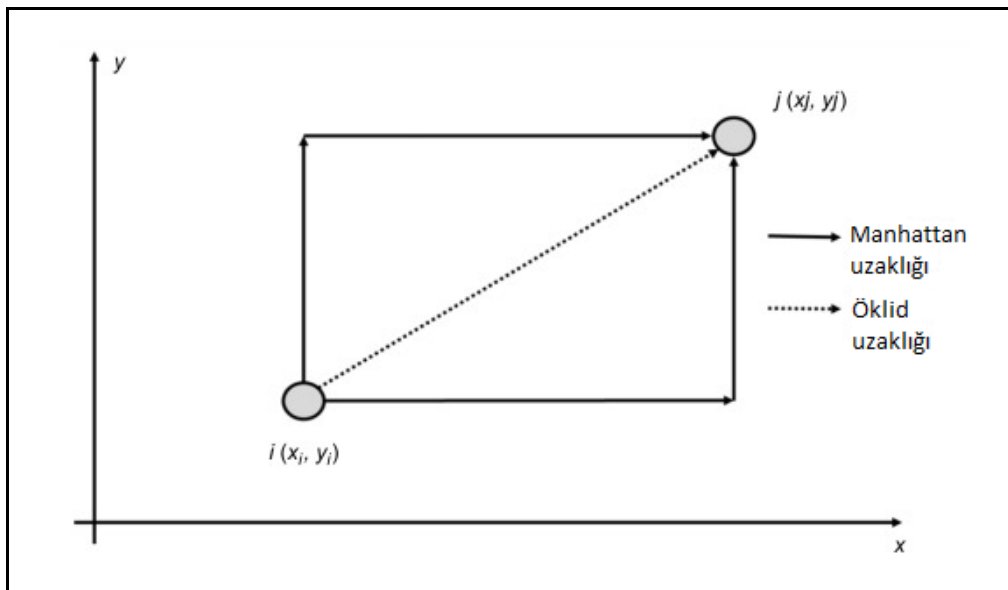
Formülüyle hesaplanmaktadır.

Kümeleme analizlerinde kullanılacak olan değişkenler belli bir önem derecesine göre ağırlıklandırılmışlarsa Öklid uzaklık ölçüsünün formülü aşağıdaki halini alır (Tatlídil, 2002).

$$d(i, j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \dots + w_p(x_{ip} - x_{jp})^2} \quad (4.8)$$

4.2.2. Manhattan uzaklık ölçüsü

Manhattan uzaklık ölçüsünde birimlerin aralarındaki mutlak uzaklık kullanılır. Basitçe ifade etmek gerekirse, bir noktanın üç birim aşağısında ve dört birim solunda bulunan noktaya olan uzaklığını Öklid uzaklık ölçüsü beş birim olarak hesaplarken manhattan uzaklık ölçüsüne göre bu iki noktanın birbirlerine olan uzaklıkları yedi birimdir.



Şekil 4.2. Manhattan ve Öklid uzaklık ölçüleri

Manhattan uzaklık ölçüsüyle iki birimin aralarındaki uzaklık,

$$d(i, j) = (|x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|) \quad (4.9)$$

formülüyle hesaplanmaktadır.

Kümeleme analizlerinde kullanılan değişkenler belli bir önem derecesine göre ağırlıklandırılmışlarsa, Manhattan uzaklık ölçüsünün formülü aşağıdaki halini alır.

$$d(i, j) = (w_1|x_{i1} - x_{j1}| + w_2|x_{i2} - x_{j2}| + \dots + w_p|x_{ip} - x_{jp}|) \quad (4.10)$$

Manhattan uzaklık ölçüsüne, “cityblock uzaklık ölçüsü” adı da verilmiştir.

4.2.3. Pearson uzaklık ölçüsü

Pearson uzaklık ölçüsüyle iki birimin aralarındaki uzaklık,

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 / S_1^2 + (x_{i2} - x_{j2})^2 / S_2^2 + \dots + (x_{ip} - x_{jp})^2 / S_p^2} \quad (4.11)$$

formülüyle hesaplanmaktadır.

Formül içinde yer alan S_p , uzaklığın hesaplandığı değişkenin varyansıdır. Fakat farklı grupların hakkında önceden bilgi sahibi olunmadığından, uzaklık hesaplanmalarında S değerlerini kullanmak doğru olmayacaktır. Bu yüzden Pearson uzaklık ölçüsünün yerine genelde Öklid uzaklık ölçüsü kullanılır.

Kümeleme analizlerinde kullanılan değişkenler belli bir önem derecesine göre ağırlıklandırılmışlar ise, Pearson uzaklık ölçüsünün formülü aşağıdaki halini alır.

$$d(i, j) = \sqrt{w_i(x_{i1} - x_{j1})^2 / S_1^2 + w_2(x_{i2} - x_{j2})^2 / S_2^2 + \dots + w_p(x_{ip} - x_{jp})^2 / S_p^2} \quad (4.12)$$

Pearson uzaklık ölçüsüne, “karesel Pearson uzaklık” ve “standardize öklid uzaklığı” adları da verilmiştir.

4.2.4. Minkowski uzaklık ölçüsü

Genel bir formül olan Minkowski uzaklık ölçüsünün içinde kullanılan m değerinin aldığı farklı değerler, yeni formüllerin oluşmasına sebep olur. Minkowski uzaklık ölçüsüyle iki birimin aralarındaki uzaklık,

$$d(i, j) = \left[|x_{i1} - x_{j1}|^m + |x_{i2} - x_{j2}|^m + \dots + |x_{ip} - x_{jp}|^m \right]^{1/m} \quad (4.13)$$

formülüyle hesaplanmaktadır.

Kümeleme analizinde kullanılan değişkenler belli bir önem derecesine göre ağırlıklandırılmışlarsa Minkowski uzaklık ölçüsünün formülü aşağıdaki halini alır,

$$d(i, j) = \left[w_1 |x_{i1} - x_{j1}|^m + w_2 |x_{i2} - x_{j2}|^m + \dots + w_p |x_{ip} - x_{jp}|^m \right]^{1/m} \quad (4.14)$$

Minkowski uzaklık ölçüsü formülünde kullanılan m değerleri, küçük ve büyük farklara verilecek olan ağırlıkları değiştirir. Yani m = 1 değerini almışsa, formül, Manhattan uzaklık ölçüsü formülüne, m = 2 değerini almışsa, formül Öklid uzaklık ölçüsünün formülüne dönüşmektedir.

4.2.5. Mahalanobis uzaklık ölçüsü

Mahalanobis uzaklık ölçüsü, iki nesnenin arasındaki uzaklığı hesaplamak için nesnelerden hesaplanan varyans-kovaryans matrisini de hesaba katar. Böylece sadece iki nesnenin arasındaki durumu değil, diğer nesnelerin durumunu da hesaba katmış olur.

Mahalanobis uzaklık ölçüsüyle iki birimin aralarındaki uzaklık,

$$d^2(i, j) = (x_i - x_j)^T S^{-1} (x_i - x_j) \quad (4.15)$$

formülü ile hesaplanır. Bu formülde kullanılan S örnek ya da küme içi kovaryans matrisidir.

4.2.6. Hotelling T² uzaklık ölçüsü

Hotelling T² uzaklık ölçüsü iki küme ve ya grubun ortalama vektörleri karşılaştırmalarında kullanılan bir tekniktir. Özetle, bir kümenin ortalamasının diğer bir kümenin ortalamasına olan uzaklık karşılaştırılır.

Hotelling T² uzaklık ölçüsüyle iki birimin aralarındaki uzaklık,

$$d(i, j) = \frac{n_1 - n_2}{n} (x_i - x_j)^T S^{-1} (x_i - x_j) \quad (4.16)$$

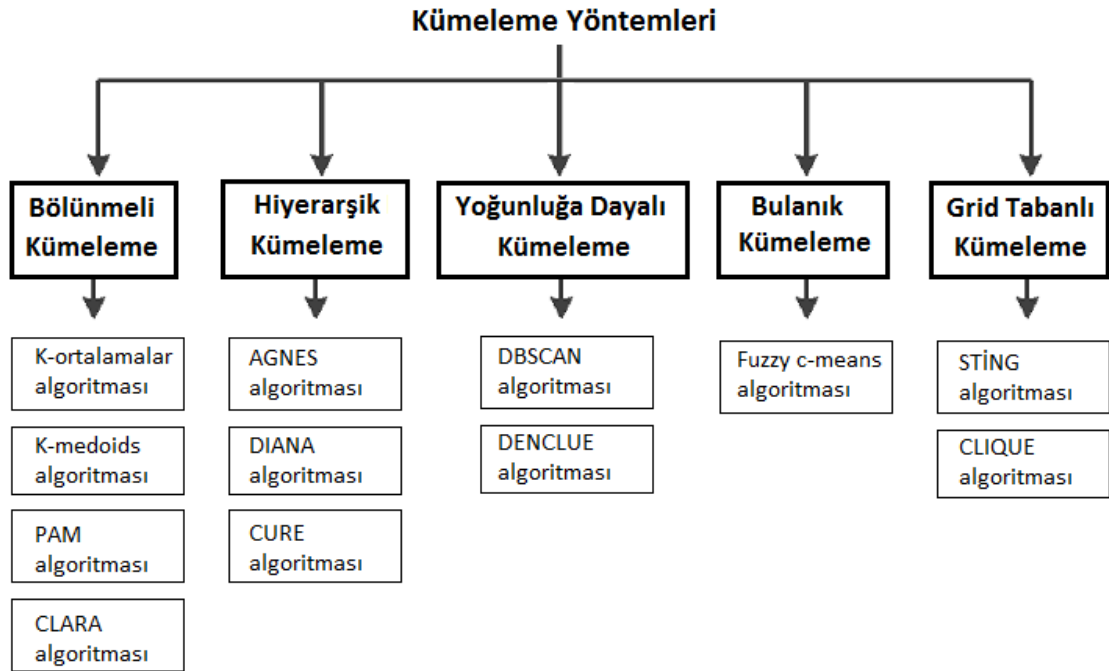
formülü ile hesaplanır.

5. KÜMELEME YÖNTEMLERİ VE ALGORİTMALARI

Kümeleme analizi birçok uygulama için kullanışlı bir araçtır. Farklı bilim dalları bünyesinde yer alan birçok araştırmacı kümeleme problemi ile karşılaşmaktadır. Bu farklı bilim dallarındaki farklı kavramları birleştirmek oldukça zor bir işlemdir. Bu yüzden kavramlar ve araştırmalar arasındaki çeşitlilik, birçok kümeleme yönteminin oluşmasına neden olmuştur.

Kümeleme işlemi gerçekleştirebilmek için çok sayıda kümeleme algoritması geliştirilmiştir. Bu algoritmaların seçimi verinin tipi, yoğunluğu, büyüklüğü gibi bazı özelliklerine doğrudan bağlıdır. Doğru kümeleme analizini bulmak istiyorsak aynı veri üzerinde birçok algoritmayı denemek ve sonuçlarını görmek gerekir (Han & Kamber, 2001).

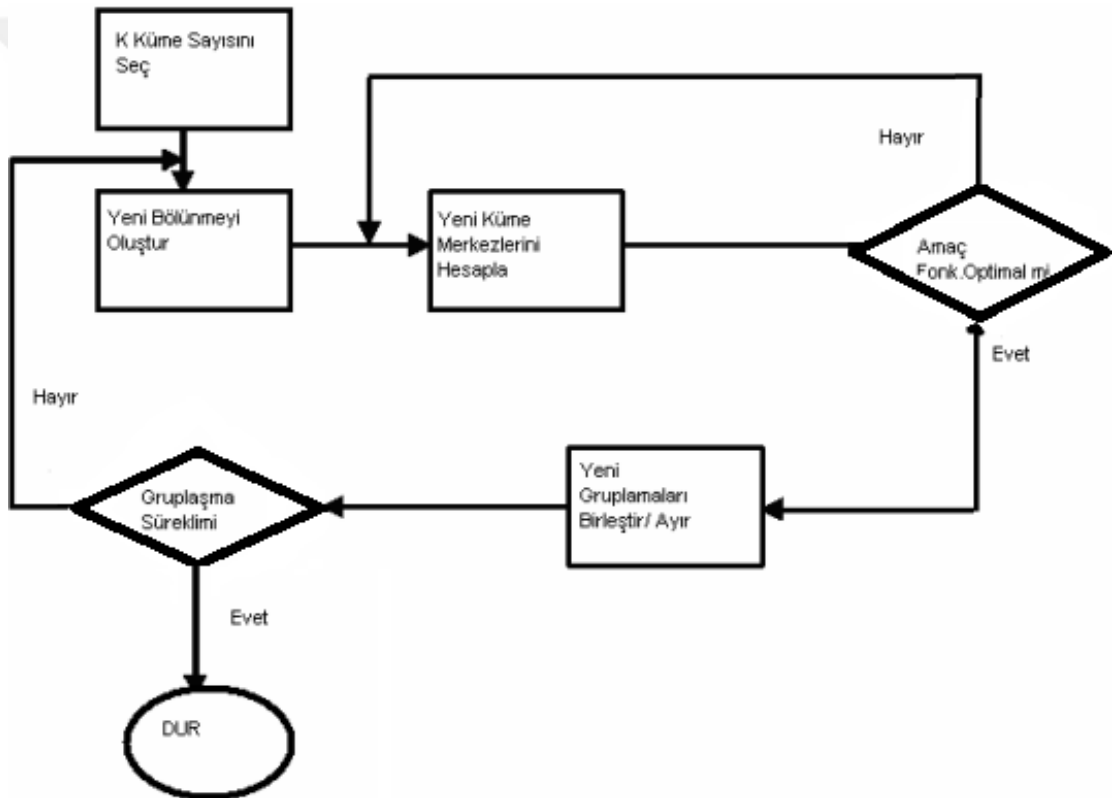
Literatürde birçok kümeleme yöntemi olmasına rağmen veri madenciliğinde kümeleme yöntemlerini başlıca beş bölüme ayırmak mümkündür (Şekil 5.1.).



Şekil 5.1. Kümeleme yöntemleri ve algoritmaları

5.1. Bölünmeli Kümeleme Yöntemi

Bölünmeli kümeleme yöntemlerinin temeli, n tane nesneyi içeren veri setinde başlangıç bölünmesinden başlayıp nesneleri yinelemeli ve her seferinde kümeleme kriteri azalan bir şekilde kümelere atamaktır. Nesneler değişik kıstaslar ele alınarak mantıksal gruplara ayrılıp analiz edilir. En fazla kriter fonksiyonunu optimize edecek şekilde bölünme sayısını bulmayı dener ve bu optimizasyonu döngüsel bir prosedürle yapar. Merkez tabanlı kümelerin tespitinde çok başarılıdır. Yapılacak olan gruplandırma analiz kalitesini direkt etkiler. Bölünmeli yöntem akış diyagramı aşağıdaki (Şekil 5.2.)’de görüldüğü gibidir.



Şekil 5.2. Bölünmeli yöntem akış diyagramı (Güler, 2006)

Bölünmeli kümeleme yapabilmek için gerekli bazı kriterler vardır. Bu kriterler şu şekildedir;

- Başlangıç koşulu
- Küme temsili yapısı
- Atama fonksiyonu

- En iyilik kriteri
- Ayırma ve birleştirme koşulları (Güler, 2006)

Bölünmeli kümelemede eğer veri seri hakkında ön bilgi mevcutsa, bu bilgi en iyi başlangıç koşulunu bulabilmek için kullanılır. Fakat özellikleri önceden elimizde olmayan veri setleri için bu yöntem, birçok kez uygulanır ve en çok oluşan küme şeması doğru bölünmüş kabul edilir. Başlangıç koşulu olarak gereken varsayımlardan en önemlisi (k) küme sayısıdır.

Küme temsil yapıları, kümeye dahil nesneleri karakterize eden geometrik şekillerdir (Michalski vd., 1981). Küme temsil yapıları içerisinde; küme merkezi, uç değerler, normal dağılım, sınıflama şeması gibi ifadeler yer verir. Oluşan küme temsil yapısı o küme içindeki nesneleri en iyi temsil eden yapıdır.

Küme için en iyilik kriteri, nesnelerin ne kadar doğru kümelendiğini belirleyen bir ölçüttür. Asıl amaç kümeyi temsil eden noktalar ile küme içerisinde yer alan diğer noktalar arasındaki uzaklığın minimize edilmesidir. Aynı zamanda yinelemenin devamlılığını belirler. Küme algoritmaları için son derece önemli olan bu kriter, yanlış seçimi sonrasında doğru olmayan kümeler elde edilmesine sebep olur.

Bölünmeli kümeleme, algoritmaya bağlı olarak orta noktalar ya da medoidler gibi kavramları temsil noktaları olarak kabul ettiği için genellikle sayısal olan veri setleri için uygulanmaktadır. Bölünmeli kümeleme yaparken en dikkat edilmesi gereken noktalardan biri de bu yöntemin uç, aykırı ya da gürültülü olarak tabir edilen değere karşı aşırı duyarlı olmasıdır.

Bölünmeli yöntem ile kümeleme yapan birçok kümeleme algoritması vardır. Bu algoritmalar başlıcaları k-ortalamlar, fuzzy c-means, k-medoids, CLARA ve CLARANS'dır (Işık, 2006).

5.1.1. K-ortalamlar (K-means) algoritması

K-ortalamlar (k-means) algoritması 1967'de Mac Queen tarafından önerilmiştir. Kümeleme problemini çözen en basit algoritmalarından biridir. K-ortalamlar algoritmasına her bir veri sadece tek bir kümeye ait olabilir. Merkez noktaların kümeyi temsil etmesine dayalı bir metottur. Verileri belirlenen küme sayıları esas alınarak, her bir küme için belirlenen ortalama değere göre kümelene yapar. K-

ortalamalar algoritması mevcut verileri k adet kümeye ve küme ortalamalarına göre kümeleme yapar. k küme sayısının adetini kullanıcı belirler. Özetle n tane nesneyi küme içindeki benzerlikleri maksimum, kümelerin aralarındaki benzerlikler minimum olacak şekilde k adet kümeye böler. Algoritmada oluşturulacak küme sayısı (k) önceden belirlenir. Algoritmanın ilk adımında küme merkezleri belirlenir. İlk küme merkezlerinin bulunması için uygulanan çeşitli yöntemler mevcuttur:

1. Merkez noktaların rastgele seçilmesi (random),
2. Doğrusal olarak seçilmesi (linear),
3. Kübik olarak seçilmesi (cube),
4. İstatistik olarak seçilmesi (statistics).

En yaygın olarak kullanılan yöntem k adet küme merkezinin rastgele olarak seçilmesidir. Algoritmanın ikinci adımında veri setindeki her nokta kendine en yakın merkez noktayla birleşir. Bu işlem sonunda k adet küme oluşmuş ve her bir veri noktası bu kümelerden birine yerleşmiş olur. Oluşan kümelerin ağırlıklı ortalama değeri hesaplanarak bu değere sahip veya bu değere en yakın nokta kümenin yeni merkez noktası olarak atanır. Merkez noktalar değişeceğinden veri setindeki noktaların merkez noktalara uzaklıkları da değişir. Bu nedenle noktaların kümelere yerleştirilmesi ve bu yerleşime göre oluşan kümelerin merkez noktalarının bulunması işlemi tekrarlanır. Bu işlem, bir kümedeki veri noktalarının kümenin merkez noktasına uzaklıkları toplamının değerindeki düşüş minimum oluncaya kadar devam eder. Burada amaç, küme merkezlerinin birbirinden maksimum uzaklıkta olması ve küme içindeki nesnelerin mümkün olduğunca birbirine yakın olmasıdır.

K-ortalamalar algoritması uygulama basamakları:

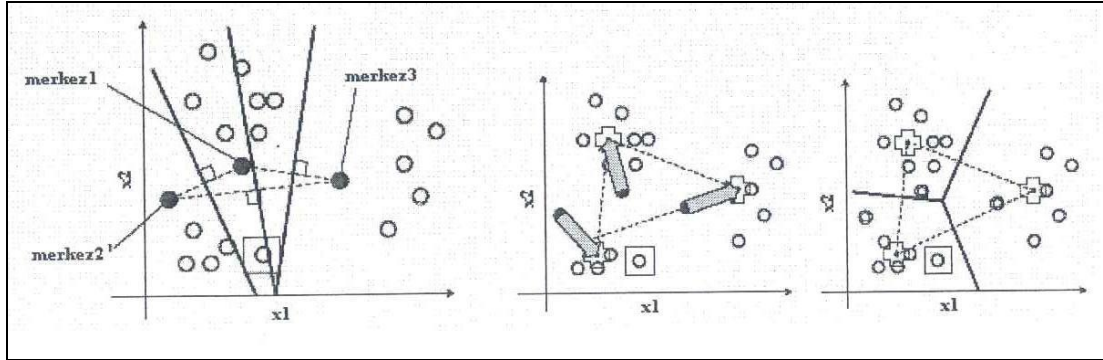
Adım 1: İlk önce kümelerin merkezleri belirlenir.

Adım 2: Her bir nesnenin belirlenen merkez noktalara olan uzakları hesaplanır ve bu sonuçlara göre her bir nesne k tane küme içinden kendisine en yakın kümeye atanır.

Adım 3: Elde edilen kümelerin yeni merkezleri, o küme içindeki bütün nesnelerin ortalamasının değeri ile değiştirilir.

Adım 4: Merkez noktalar değişmeyene dek ikinci ve üçüncü adımlar tekrarlanır.

K-ortalamlar algoritması işlem basamakları (Şekil 5.3.)’de görülmektedir.



Şekil 5.3. K-ortalamlar algoritması işlem basamakları

K-ortalamlar algoritması, küresel kümelerin bulunmasında başarılı bir algoritmadır. K-ortalamlar algoritmasında, tekrar sayısı (t), veri noktaları sayısı (n) ve küme sayısı (k) ise $(t < n, k)$ olmalıdır.

K-ortalamlar algoritmasının avantajları şöyle sıralanabilir;

1. Uygulanabilirliği kolaydır.
2. Büyük veri setlerinde de uygulanabilir.

K-ortalamlar algoritmasının pek çok alanda uygulanmasını sağlayan avantajlarının yanında birtakım dezavantajları da bulunmaktadır. Bunlar;

1. K-ortalamlar algoritması ile elde edilen kümeler, küme sayısı, ilk küme merkezlerinin seçim yöntemi, veri noktalarının sıralaması ve verinin geometrik özelliklerine bağlıdır.
2. K-ortalamlar algoritmasında k değeri önceden tespit edilemediğinden başarılı bir kümeleme elde etmek için deneme yanılma yönteminin kullanılması gerekmektedir.
3. Uç değerlere ve gürültülü verilere karşı duyarlı bir algoritmadır.
4. Ortalama değeri hesaplanamayan kategorik verilerde kullanılamaz.
5. Konveks olmayan kümelerin bulunmasında uygun değildir.

5.1.2. K-medoids algoritması

1987 yılında Kaufman ve Rousseeuw tarafından sunulan k-medoids algoritmasında küme merkezleri k-ortalamlar algoritmasından farklı olarak kümedeki noktaların ağırlıklı ortalama değeri hesaplanarak değil kümenin en merkez noktasında yerleşmiş nokta bulunarak belirlenir (Işık, 2006). Bu işlem sıra dışı noktaların küme merkezini etkilemesini engellemektedir. Merkez noktaların bulunmasında ortalama değer bulunması gibi matematiksel işlemlere gerek olmadığından k-medoids algoritması kategorik verilerde kullanılabilir.

K-ortalamlar algoritmasından farklı olan k-medoids algoritmasında küme merkezleri, kümedeki bireylerin ortalama değeri ile değil, küme merkezine en yakın birey (medoids) bulunarak belirlenir. Böylece gürültülü ve aykırı değerlere karşı k-ortalamlar tekniğine oranla daha az hassastır.

K-medoids algoritması uygulama adımları:

- Adım 1: k küme sayısı belirlenir.
- Adım 2: Başlangıç medoidleri olarak k bireyleri seçilir.
- Adım 3: En yakın medoide sahip küme içerisine, diğer bireyler atanır.
- Adım 4: Amaç fonksiyonu hesaplanır. (Hata kareler kriteri: en yakın medoidler için tüm bireylerin uzaklıkları toplamı)
- Adım 5: Tesadüfi şekilde bir y noktası seçilir.
- Adım 6: Eğer x ile y'nin yer değiştirmesi amaç fonksiyonunu minimize ediyorsa x ile y'nin yerleri değiştirilir.
- Adım 7: Değişiklik olmayana dek Adım 3 ile Adım 6 arasındaki adımlar tekrarlanır (Akın, 2008).

5.1.3. PAM algoritması

K-medoids algoritmasının farklı türevleri vardır ve bu türevler arasında ilk ortaya atılmış olan K-Medoids algoritması PAM (Partitioning Around Medoids) algoritmasıdır. PAM algoritmasında küme sayısı (k) kadar merkez nokta (medoid) rastgele olarak seçilir. Kümeye katılan her yeni bireyin ardından, küme bireyleri

PAM algoritmasının karmaşıklığı $O[k(n - k)^2]$ 'dir. Burada, n: nesne sayısı, k: küme sayısını göstermektedir.



a: p bireyi şu anda O_j merkez noktasına bağlıdır. Eğer O_j , $O_{rastgele}$ ile yer değiştirirse bunun sonucunda p bireyi O_i merkez noktası daha yakın olursa, p O_i 'ye geçer.

- b: p bireyi şu anda O_j merkez noktasına bağlıdır. Eğer O_j , $O_{rastgele}$ ile yer değiştirir ve p bireyi $O_{rastgele}$ 'ye daha yakın ise, p $O_{rastgele}$ 'ye geçer.
- c: p bireyi şu anda O_i merkez noktasına bağlıdır. Eğer O_j , $O_{rastgele}$ ile yer değiştirir ve p bireyi hala $O_{rastgele}$ 'ye en yakın ise, p yine O_i 'ye bağlı kalır.
- d: p bireyi şu anda O_i merkez noktalarına bağlıdır. Eğer O_j , $O_{rastgele}$ ile yer değiştirir ve p $O_{rastgele}$ 'ye en yakın ise, p bireyi $O_{rastgele}$ 'ye geçer.

5.1.4. CLARA algoritması

PAM algoritması sadece küçük veri setleri üzerinde çalışabilmekte büyük veri setlerinde istenilen hassasiyeti gösterememektedir. PAM algoritmasının bu dezavantajını ortadan kaldırmak için 1990 yılında Kaufman ve Rousseeuw tarafından CLARA (Clustering for Large Applications) algoritmasını geliştirilmiştir.

CLARA algoritması örnekleme yaparak geliştirilmiş bir tekniktir. CLARA algoritması, veri setinden başlangıçta birden fazla örneklem çeker ve bu her bir örnekleme PAM algoritmasını uygulayarak medoidleri bulmaya çalışır. Eğer veri setinden örneklem rastgele seçildiyse belirlenen medoid tüm veri setini iyi bir şekilde temsil edecektir. Böylece CLARA algoritması, elde edebileceğimiz en iyi kümeleri çıktı olarak verecektir. Yapılan araştırmalar, bir veri setinden çekilen 5 örneklemin her birinde toplam nesne sayısının $40+2k$ 'sı kadar nesne bulunduğunda elde edilen sonuçların anlamlı olduğunu göstermiştir. CLARA algoritmasının adımları aşağıdaki gibidir:

Adım 1: $i=1,2,3,4,5$ için aşağıdaki adımları tekrarlayın.

Adım 2: Başlangıçtaki veri setinden rastgele $40+2k$ 'lık bir örneklem çekilir ve bu örneklemin k medoidini saptamak için PAM algoritması kullanılır.

Adım 3: Orjinal veri seti içerisindeki nesnelerin her biri için o nesneye en fazla benzerlik gösteren k medoidler belirlenir.

Adım 4: Önceki adımda elde ettiğimiz her bir kümenin birbirlerine olan ortalama uzaklıkları hesaplanır. Bulunan değerler mevcut olan minimum değerlerden daha küçük ise yeni minimum değerlerimiz olarak atanırlar

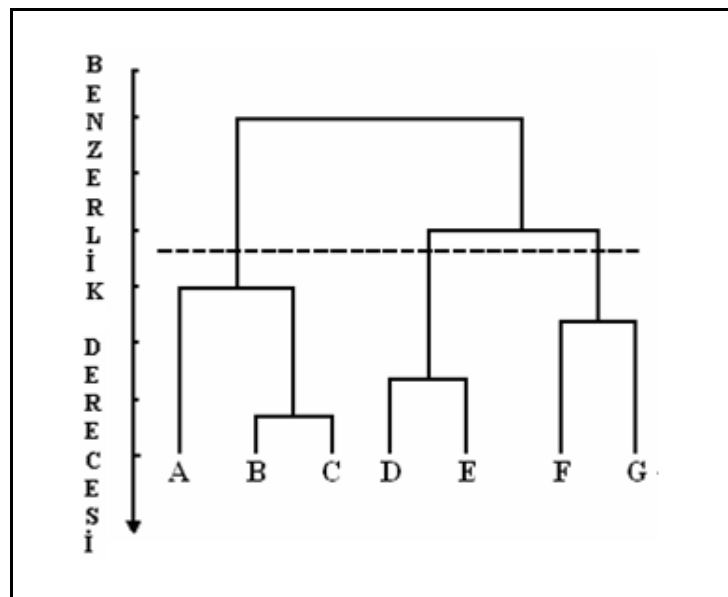
ve Adım 2' de elde ettiğimiz k medoidleri en iyi medoid kümesi olarak kabul edilir.

Adım 5: Daha sonraki iterasyona başlayabilmek için Adım 1'e geri dönülür.

CLARA algoritmasının büyük veri setlerinde kullanılması en büyük avantajı olmasına rağmen algortmadan iyi bir sonuç alabilmek, seçilecek örneklem büyüklüğüyle tamamen doğru orantılıdır. Bunun nedeni algoritma küme merkezlerini k medoidler olarak bu örneklem içerisinden belirlemektedir. Bu sebeple k medoid olan bir nesnenin örnekleme seçilmemesi durumunda algoritma doğru bir kümeleme yapamayacaktır. CLARA algoritması için her bir tekrarlamanın hesaplama karmaşıklığı $O[ks^2 + k(n-k)]$ ' dir. Burada s:örneklemin büyüklüğü, k: küme sayısı ve n: toplam nesne sayısıdır.

5.2. Hiyerarşik Kümeleme Yöntemi

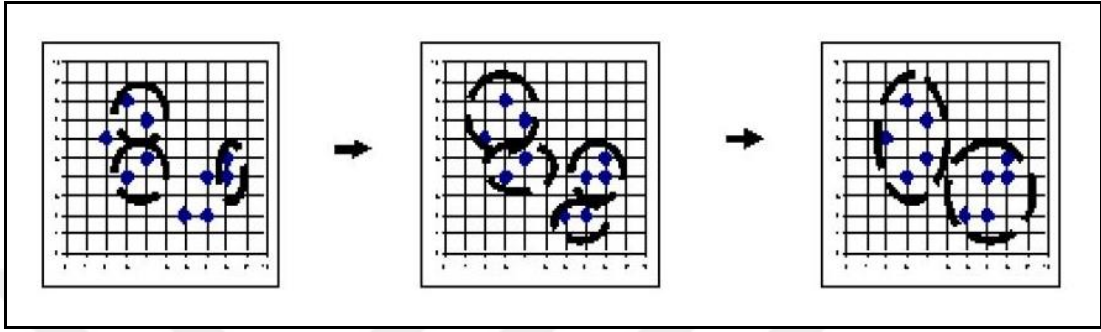
Analiz edilmeden önce nesneler, dendogram ya da ağaç yapısı olarak da bilinen hiyerarşik yapılarına göre düzenlenmektedir. Kümelerin her biri tek bir ebeveyne bağlı verilerin kardeş ve çocuk küme bölünmesini barındırır. Bu şekildeki bir yapı, başka seviyedeki veri yığınlarının da dikkatlice analiz edilebilmelerini sağlar (Jain vd., 1999).



Şekil 5.5. Hiyerarşik kümeleme yöntemi dendrogramı

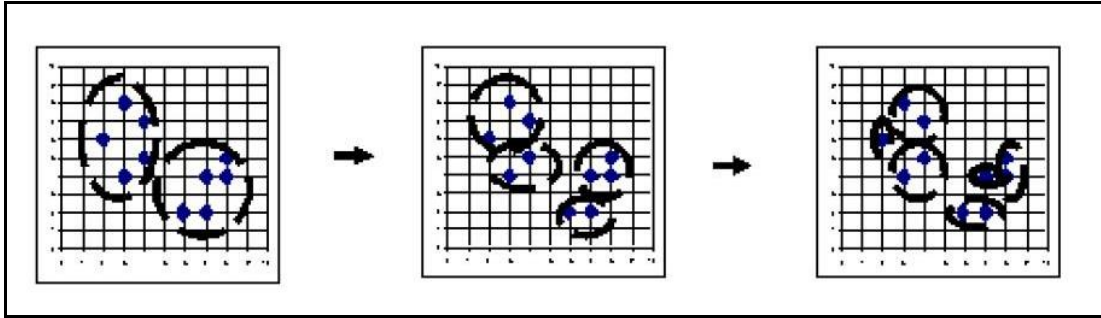
Veriyi hiyerarşik bir yapıya çevirmek için değişik yöntemler uygulanır. Temel olarak iki tipte sınıflandırılabilen bu yöntemler aşağıdaki gibidir:

- Birleştirme yaklaşımı (Agglomerative): Nesnelerin her biri başlangıçta tek başlarına bir gruba temsil ederken daha sonra birbirlerine benzer nesneler ya da gruplar birleştirilir.



Şekil 5.6. Birleştirici hiyerarşik kümeleme

- Bölme yaklaşımı (Divisive): Tüm nesneler başlangıçta tek bir kümeye aitken her bir öteleme işleminde kümeyi uzaklık ve benzerliklere göre daha küçük alt kümelere ayrıştırılır.



Şekil 5.7. Ayrıştırıcı hiyerarşik kümeleme

Hiyerarşik kümeleme yönteminin avantajları ve dezavantajları aşağıdaki gibi sıralanabilir;

Avantajları:

1. Belirli bir küme sayısına bağlı değildir yani giriş parametresi olarak istenen küme sayısını belirten k değerinin verilmesine gerek yoktur. Tersine

dendrogramın uygun bir seviyede kesilmesiyle istenen sayıda küme elde edilebilir.

2. Aykırı, uç, gürültülü değer olarak nitelendiren değerleri içeren veri setlerinde, bölünmeli kümeleme metotlarına oranla çok daha iyi sonuçlar vermektedir.
3. Hızlıdır.
4. Benzerlik ve uzaklık seçimi ve idare edilmesinin kolaydır.
5. Her tür veri tipine uygulanabilmektedir.
6. Sunumu kolaydır.
7. Bölünmeli kümeleme metotlarına oranla daha gerçek sonuçlar vermektedir.
8. Biyolojik bilimlerde sıkça kullanılan âlem, filum, cins, tür gibi taksonomilere benzerliği nedeniyle bu alanda etkili bir biçimde kullanılabilir.

Dezavantajları:

1. Çok büyük veri setlerinde başarılı sonuçlar elde edilmez.
2. Birleştirme ve ayrıştırma işlemleri gerçekleştikten sonra değiştirilip geri alınamamaktadır.
3. Bitirme kriterlerinin belirsiz olması.
4. Aykırı değerlerden her ne kadar bölünmeli yöntemlere nazaran daha az etkilenseler de bazen etkisi çok fazla olabilir.
5. Lineer olmayan zaman karmaşıklığının büyük problemler teşkil etmesi.

Başlıca hiyerarşik kümeleme algoritmaları olarak Birch, CURE, AGNES, DIANA ve Chamelon'u sayabiliriz.

5.2.1. AGNES algoritması

Toplayıcı kümeleme algoritması olarak bilinen AGNES (AGglomerative NESTing) 1990 yılında Kaufman ve Rousseeuw tarafından sunulmuştur. Algoritmada temel mantık oldukça basittir. Her bir nesnenin tek başına oluşturduğu n birimi kümeyi, hiyerarşik şekilde sırayla $n, n-1, n-2, \dots, n-r, \dots, 4, 3, 2, 1$ birim kümede birleştirmeyi amaçlar. İnşa yapısı aşağıdan yukarıya doğru izlenen bir yapıdadır.

AGNES algoritmasının adımları (Han & Kamber, 2001):

Adım 1: Başlangıçta her bir nesne ayrı bir kümeyi temsil ettiği kabul edilir.

Adım 2: Sonraki her adımda bu atomik kümeler benzerlik gösteren bir başka kümeyle birleştirilir.

Adım 3: Her birleşmenin ardından toplam küme sayımız bir azalır.

Adım 4: Küme sayısında istenilen sayı elde edildiğinde ya da en yakın iki kümenin aralarındaki uzaklık belirlenen eşik değere ulaştığı zaman birleşme sona erer.

Adım 5: Her bir nesne tek bir küme içerisinde toplanana kadar ya da belirlenen bir sonlandırma koşuluna ulaşılanaya kadar bu işlem devam eder.

5.2.2. DIANA algoritması

Ayırıcı kümeleme algoritması olarak bilinen DIANA (DIvisive ANAlysis) algoritması da yine 1990 yılında Kaufman ve Rousseeuw tarafından sunulmuştur. n sayıda nesneyi tek bir kümenin elemanları olduğu kabul ederek, bu n sayıda nesneyi hiyerarşik bir şekilde $1, 2, 3, 4, \dots, n-r, \dots, n-2, n-1, n$ kümeyle ayırmayı amaçlar. Bu teknikte kümeleme süreci toplayıcı kümeleme tekniğinin tersine yukarıdan aşağıya doğru bir süreçtir.

DIANA algoritmasının adımları:

Adım 1: Başlangıçta her bir nesne tek bir kümenin elemanları olarak kabul edilir.

Adım 2: Sonraki her adımda birbirlerine benzerlik oranları en fazla olan nesneler yeni bir küme oluşturularak büyük küme ikiye bölünür.

Adım 3: Her ayırıştırmanın ardından toplam küme sayımız bir artar.

Adım 4: Küme sayısında istenilen sayı elde edildiğinde ya da en yakın iki kümenin aralarındaki uzaklık belirlenen eşik değere ulaştığı zaman ayırıştırma işlemi sona erer.

Adım 5: Her bir nesne tek başına bir küme olana kadar ya da belirlenen bir sonlandırma koşuluna ulaşılanaya kadar bu işlem devam eder.

5.2.3. CURE algoritması

CURE algoritması (Clustering Using REpresentatives) ilk olarak SIGMOD 1998 konferansında Guha, Rastogi ve Shim tarafından sunulmuştur. Algoritma diğer kümeleme tekniklerinin dezavantajlarını ortadan kaldırmak için oluşturulmuştur (Guha ve ark. , 1998) CURE algoritması kümeyi temsil etmek için bir tane merkez nokta kullanmak yerine kümeyi sabit sayıda temsil edecek birçok noktayı seçmeyi tercih eder. İki küme arasındaki benzerlik, farklı kümelerdeki temsili noktaların en yakın çiftinin benzerliğine göre ölçülür. Birleştirilen kümeler için yeni temsili noktalar, bütün veri setinden dağılmış noktalardan sabit bir sayı seçilerek ve daraltma faktörü diye adlandırılan ve " α " ile sembolize edilen bir parametreye göre kümenin merkezine doğru daralarak (shrinking) belirlenir (Sia ve Lazarescu, 2005). Algoritmanın her aşamasında, temsili noktaların en yakın iki çiftine sahip iki küme birleştirilmektedir. Ayrıca birleştirme işlemi yapılırken birleştirilen çiftlerdeki her bir noktanın farklı kümelerden olduğuna dikkat edilmelidir. Ancak, seçilen temsili noktalar bazen kümeleri tanımlamak için yetersiz olmakta ve bu durum daraltma aşamasında bazı noktalar çıkartılmaya çalışıldığında daha da iyi ortaya çıkmaktadır.

Kümelerin her biri için birçok temsili noktası saptayan CURE algoritması merkeze dayalı kümeleme tekniklerinin aksine, küresel olmayan şekillere sahip ve büyüklük açısından birbirlerinden farklı büyüklükte olan kümelerin tanımlanmasında da uygulanmaktadır. Ayrıca algoritmadaki daraltma faktörünün yani α 'nın farklı değerleri, kümelerdeki şekillere, büyüklüklere ve verideki gürültüye bağlıdır. Bu nedenle kümelerin daraltılması uç değerlerin etkilerinin azalmasına neden olacağından algoritma aykırı değerlere karşı da çok kullanışlıdır. Algoritma önce tesadüfi bir örnek seçerek önce bölünür ve her bir bölme kısmen kümelenir. Bu kısmi kümeler tekrar kümelenecek istenen asıl kümeler elde edilir. Küme kalitesine çok fazla önem veren CURE algoritması büyük veri tabanları için çok kullanışlıdır (Karypis & Kumar, 1999).

CURE algoritmasının uygulama basamakları aşağıdaki gibidir:

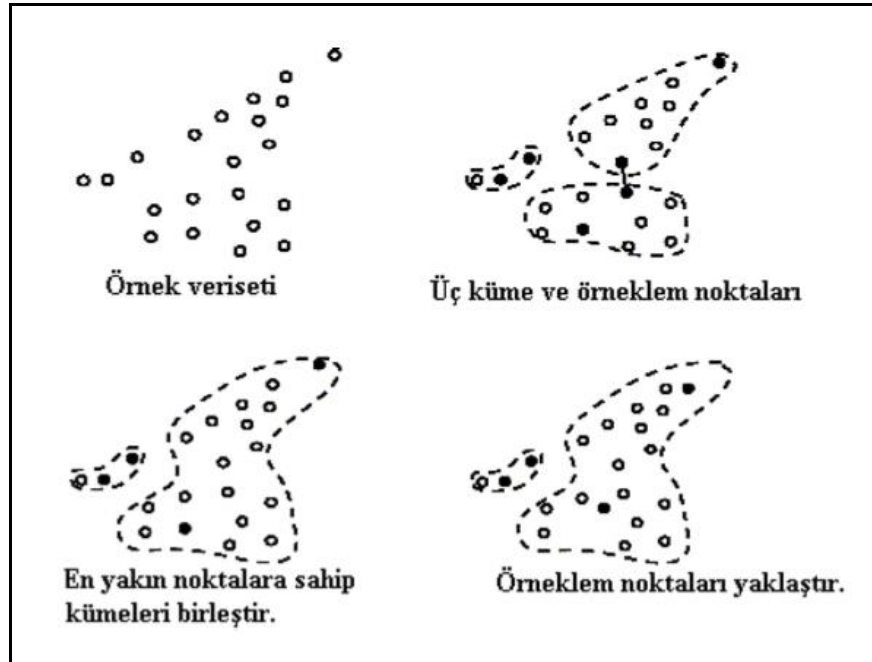
- Adım 1: Başlangıçtaki data setinden rassal bir örneklemin çekilmesi,
- Adım 2: Seçilen örneklemin bölmeler setine ayrılması,
- Adım 3: Bölmelerin her birinin kısmi olarak kümeleneceği.

Adım 4: Aykırı değerlerin elimine edilmesi, eğer çok yavaş büyümekte olan bir küme var ise o kümenin çıkartılması,

Adım 5: Kısmi oluşturulan kümelerin baştan kümelenmesi,

Adım 6: İlgili küme etiketleriyle kümelerin işaretlenmesi.

CURE algoritmasının genel çalışma şekli (Şekil 5.8.)’de görülmektedir. Başlangıçta her bir veri noktası farklı birer küme olduğu kabul görür ve algoritma her başarılı adımda birbirine en yakın örneklem nokta çiftini içeren iki kümeyi birleştirir. Örneklem noktalar, daraltma katsayısı (α) kullanılarak merkeze doğru yaklaştırılır. Birleştirilme adımları istenilen küme sayısı (k), elde edilinceye dek sürer. Bir küme çifti arasındaki uzaklığın hesaplanması için her bir küme için c adet örneklem nokta saklanır. Bu örneklem noktaları bulabilmek için öncelikle küme içinden c adet nokta seçilir ve daha sonra bu noktalar daraltma katsayısına bağlı olarak kümenin ortalama değerine doğru yaklaştırılır. Dolayısıyla, iki kümenin aralarındaki uzaklık, bu kümelerin örneklem noktaları arasındaki uzaklığa eşittir. Bir kümenin diğer kümelere uzaklığını hesaplamak için tüm veri noktaları yerine sadece örneklem noktalar kullanılır. CURE algoritmasında uygulanan kümeler arası uzaklık formülü genellikle öklid uzaklık formülüdür.



Şekil 5.8. CURE algoritması temel çalışma şekli

CURE algoritmasıyla gerçekleştirilen deneysel çalışmalardan varılabilecek sonuçlar aşağıda ortaya konulmuştur:

- CURE algoritması farklı şekillere sahip kümeleri bulabilmekte,
- CURE aykırı değerlere karşı çok duyarlı olmamakta ve bu tip değerleri başarıyla atmakta,
- Örnekleme ve bölümlleme işlemlerini küme kalitesinden ödün vermeyerek data setinin boyutunu azaltmakta,
- Kümeler şekilsiz olduklarında bile, disk üzerinde son etiketlenme işlemini doğru olarak yapmakta.

CURE algoritması avantajları ve dezavantajları şöyle sıralanabilir (Han & Kamber, 2001);

Avantajları:

1. Küresel olmayan şekillere ve farklı boyutlara sahip kümeleri bulur,
2. Aykırı değerlere karşı duyarsızdır,
3. Rastgele örnekleme ve bölümleyici metotlarla beraber kullanıldıklarında çok büyük veri setlerinde ölçeklenebilir zaman karmaşıklığını azaltır.

Dezavantajları:

1. Kategorik verilere sahip veri setlerinde kullanılmaz sadece sayısal veri içeren setlerde etkilidir,
2. Seçilecek olan parametrelere çok duyarlıdır.

5.3. Yoğunluğa Dayalı Kümeleme Yöntemi

Kümeleme yöntemlerinin birçoğu nesneler arasındaki farklılıkları esas alarak kümeleme yaparken çıkan sonuçlar genellikle küreseldir. Bu yöntemler değişik şekillerde olan kümeleri tespit etmede yetersiz kalmaktadır. Yoğunluğa dayalı yöntemler ise nesne yoğunluklarını esas alarak kümeleme yapar. Komşuluk içindeki yoğunluk belli bir seviyeyi aşana dek kümeler genişlemeye devam etmektedir.

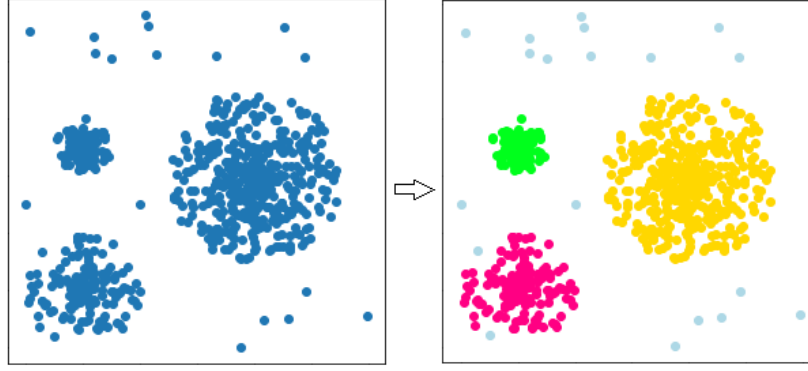
Yoğunluğa dayalı yöntemlerin en önemli özelliklerinden biri gürültülü ve sıra dışı verilerden etkilenme oranları çok azdır.

Yoğunluğa dayalı kümeleme yöntemleri arasında en çok kullanılan algoritmalara örnek olarak DBSCAN, DENCLUE ve OPTICS algoritmaları gösterilebilir ve bu algoritmalar arasında da en yaygın kullanılan DBSCAN algoritmasıdır.

5.3.1. DBSCAN algoritması

Yoğunluk bazlı algoritmaların temelini oluşturan DBSCAN algoritması, ilk olarak Martin Ester, Hans-Peter Kriegel, JörgSader ve XiaoweiXu tarafından 1996 yılında Amerika’da yapılan Knowledge Discoveryand Data Mining konferansında sunulmuştur. DBSCAN kelimesi “Density Based Spatial Clustering of Applications with Noise” sözcüklerinin baş harflerinin bir araya gelmesiyle oluşturulmuştur. DBSCAN, büyük veri setleri ve aykırı değerler içeren veri setleri için oldukça uygun bir algoritmadır. Ayrıca farklı şekil ve büyüklüklere sahip kümeleri tanımlamada da oldukça sık kullanılır.

Kümeler, yüksek yoğunluklu verilerin olduğu alanlar olarak tanımlanır, düşük yoğunluklu verilerin olduğu alanlar aykırı değerleri ya da gürültülü noktalarını gösterir. Bu algoritmada kümeler oluşturulurken verilerin yoğunlukları dikkate alınır. Diğer kümeleme algoritmalarından bir başka farkı da “pointdensity” kavramıyla verileri ilişkilendirmesidir. Nesnelerin yoğun bir şekilde toplandığı bölgeler point density bölgelerdir. Bir başka deyişle veri setinin tamamına bakıldığında görülebilen alt kümeler olarak da tanımlanabilir. Örneğin (Şekil 5.9.)’da 3 adet küme olduğu rahatlıkla söylenebilir.



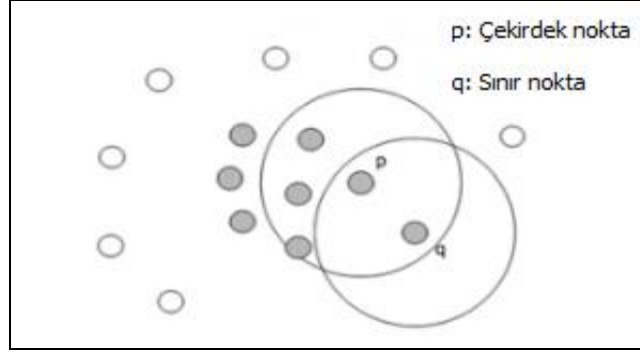
Şekil 5.9. Yoğunluk tabanlı veri

Veri yoğunluklarının hızlı biçimde değişiklik gösterdiği alanlar kümelerin oluşturulmasını sağlayan alanlardır. Bu alanların dışındaki nesneler gürültü “noise” diye adlandırılır. Bu noktalar hiçbir kümeye dâhil edilmezler ve (Şekil 5.10.)’da sağ tarafında kalmış olan açık renkli noktalar buna örnek olarak verilebilir.

DBSCAN algoritmasına ait ϵ ve MinPts olarak adlandırılan iki adet girdi parametresi vardır. Burada (ϵ) epsilon: Yarıçap yani küme içindeki tüm verinin epsilon yarıçapındaki komşuluğu anlamına gelmektedir. MinPts, kümelerin çevrelerinde olan minimum veri sayısı olarak tanımlanır. Yukarıda tanımlanan bu giriş parametrelerine karşı algoritmanın duyarlı olması DBSCAN algoritmasının en önemli dezavantajlarından biri olarak görülmektedir. Algoritmanın adımlarının daha iyi anlaşılması için aşağıdaki kavramlar incelenebilir:

Bir kümenin nesneleri eğer her birinin ϵ yarıçaplı alanında en az MinPts ile belirlenen kadar veri yer alıyorsa bir kümedeki veriler yoğunluklarına göre ayrıldığından bu veriler yoğun veriler olarak tanımlanmaktadır. Verilen bir verinin ϵ yarıçapı ile komşuluğu; verinin ϵ komşuluğu (neighborhood) olarak adlandırılmaktadır.

Eğer bir nesnenin ϵ komşuluğu en az MinPts sayıda nesne içeriyor ise, bu nokta çekirdek nokta (p) (core object) olarak adlandırılmaktadır. Verilen bir veri setinde, eğer p noktası, q noktasının (sınır nokta) ϵ komşuluğu içerisindeyse q noktası p noktasının doğrudan yoğunluğa katılan nesnesi (directly density reachable) olarak tanımlanmaktadır ve burada q noktası çekirdek nesneyi temsil etmektedir.



Şekil 5.10. DBSCAN algoritması

Epsilon ve MinPts parametre seçimleri küme çıktısını şekillendirmesi noktasında çok önemlidir. Epsilon parametresi seçiminde aşağıdaki hususlara dikkat edilmelidir:

Veri normalizasyonu özellikle 2 den daha çok boyutlu veriler için normalizasyon yapılması daha fazla önem taşımaktadır. Veri setlerinin yapılarıyla ilgili kapsamlı bilgi elde edilmiş olur. Bu durumda noktaların arasındaki uzaklıkların biliniyor olması epsilon saptaması noktasında önemli bir bulgudur.

Algoritmanın adımları (Han &Kamber, 2001) :

Adım 1: Data setindeki tüm verinin ϵ komşuluğunu kontrol et.

Adım 2: Bu alan içerisinde MinPts ile belirlenmiş sayıdan daha çok veri varsa bu noktaya çekirdek nokta adını ver.

Adım 3: Tüm kümeleri, çekirdek noktalarından direkt yoğunluğun içerisine katılarak toplanmış noktaların yardımı ile çekirdek veri etrafında büyüt.

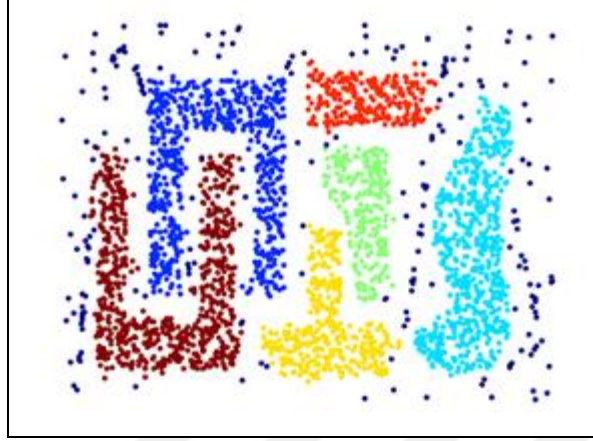
Adım 4: Verilerin kümelere katılma işlemleri sonlandığında yani artık hiç bir veri herhangi bir küme içerisine dâhil olmadığında algoritmayı sonlandır.

Adım 5: Eğer son durumda herhangi bir kümeye dâhil olamamış bir veri var ise bu verileri gürültü diye nitelendir.

DBSCAN algoritmasının avantajları şöyle sıralanabilir:

1. DBSCAN kümenin yoğunluk-tabanlı tanımını kullandığından gürültüye dayanıklıdır.

2. Farklı şekilde ve büyüklükte kümeleri ele alabilir. Bu yüzden (Şekil 5.11.)’de gösterildiği gibi DBSCAN algoritması K-ortalama ile bulunamayan birçok kümeyi bulabilir.



Şekil 5.11. Farklı şekil ve büyüklükte kümelerden oluşan veri seti

DBSCAN algoritmasının dezavantajları ise şunlardır:

1. Kümeler arasındaki yoğunluk farkı büyükse DBSCAN algoritması problem yaşayacaktır.
2. Yüksek boyutta verilerde DBSCAN algoritması problem yaşayacaktır, bunun nedeni böyle veriler için yoğunluk belirlemek oldukça zordur.
3. DBSCAN algoritmasında yakın komşulukların aranması tüm çiftlerin yakınlıklarının hesaplanmasını gerektiriyorsa, bu durum genelde yüksek boyutlu verilerde oluşur ve maliyeti yüksek olacaktır.

5.3.2. DENCLUE algoritması

Bir diğer yoğunluğa dayalı kümeleme algoritması olan DENCLUE (DENsity based CLUstering) algoritması ilk olarak Alexander Hinneburg ve Daniel A. Keim tarafından KDD’98 konferansında sunulmuştur. DENCLUE algoritması farklı etki (influence) fonksiyonunu ve yoğunluk (density) fonksiyonunu kullandığı için diğer tüm yoğunluğa dayalı algoritmaları da kapsadığından dolayı bu yöntem altındaki algoritmaların genelleştirilmiş hali sayılabilir.

DENCLUE algoritması, amacı veri setlerinin dağılımlarını bir fonksiyon gibi tanımlamak olan Kernel yoğunluk tahminine dayanır. Bu yöntemde, her verinin toplam yoğunluk fonksiyonuna katkısına etki fonksiyonu adı verilir ve bu şekilde toplam yoğunluk fonksiyonu her bir veriyle ilişkilenen etki fonksiyonları toplamından oluşur. Çoğunlukla Öklid uzaklığına benzer şekilde simetrik olan etki fonksiyonunun değeri veriye olan uzaklığı azaldıkça artmaktadır (Akin, 2008).

DENCLUE algoritmasının adımları;

Adım 1: Yoğunluk fonksiyonu hesaplanmasında kullanılmak üzere veri uzayının uygun kısımlarına ait bir harita çıkartılır.

Adım 2: Verilerin her birinin kendi çevrelerince etki fonksiyonları bulunur.

Adım 3: Etki fonksiyonları toplanarak veri setinin genel yoğunluğu hesaplanır.

Adım 4: Toplam yoğunluk fonksiyonun yerel maksimumu olan yoğunluk çekicileri sayesinde kümeler oluşturulur.

DENCLUE algoritmasının avantajlarını aşağıdaki gibi sıralanabilir:

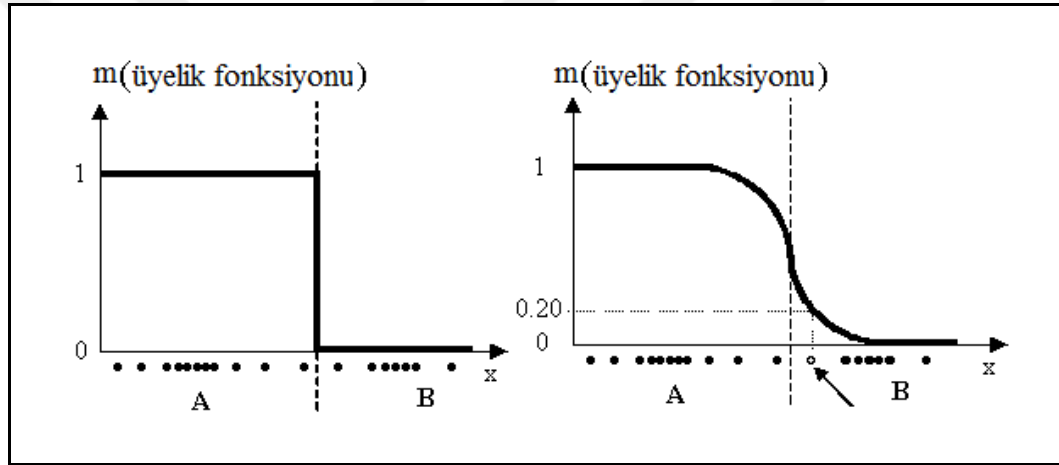
1. Çok benzeri olan DBSCAN algoritmasından 45 kat daha hızlı çalışmaktadır.
2. Oldukça güvenilir bir matematiksel temel üstüne kuruludur.
3. Diğer küme algoritmalarını kapsayan genelleştirilmiş hali gibidir.
4. Aykırı değerleri içeren veri setleri için uygundur.
5. Çok boyutlu veri seti içerisinde keyfi şekillere sahip kümeleri matematiksel olarak tanımlayabilir (Qiang vd., 2005)

DENCLUE algoritmasının dezavantajının ise, iki parametreye dayanması ve kümeleme başarısının bu parametrelere sıkı sıkıya bağlı olması olarak söyleyebiliriz.

5.4. Bulanık Kümeleme Yöntemi

Bulanık mantık, sonucu önceden oluşmuş bir mantıktır. Klasik mantık önermeler sadece iki değer alırlar ancak gerçek hayatta veriler doğru-yanlış, sıcak-soğuk, uzun-kısa vb. gibi sadece iki değer ile ifade edilemeyebilirler. Bu gibi nedenlerden dolayı geliştirilen bulanık mantık, 1965 yılında Azerbaycanlı Prof. Dr. Lotfi Zadeh tarafından ortaya atılmış olan bulanık küme teorisine dayanır. Bulanık mantıkta

klasik yaklaşımın farklı olarak sadece iki değer yoktur ve bir önerme ‘yanlış’, ‘doğru’, ‘yaklaşık olarak yanlış’, ‘yaklaşık olarak doğru’, ‘çok yanlış’, ‘çok doğru’, ‘çok çok yanlış’, ‘çok çok doğru’ gibi çeşitli değerler ile ifade edilebilir. Klasik kesin kümelemelerde bir veri bir kümenin ya elemanıdır ve bu 1 ile ifade edilir ya da elemanı değildir ve bu da 0 ile ifade edilir. Ancak bulanık kümeleme teorisinde de veriler, gerçek hayattaki gibi sadece iki değer yani sadece 0 ile 1 değerlerini almazlar. Bulanık kümeleme yönteminde veriler 2’den fazla değer alabilir. (Şekil 5.12.)’de görüldüğü gibi $[0, 1]$ değerleri aralığında değişen çeşitli üyelik derecelerine sahip olabilirler. Küme merkezine yaklaştıkça nesnelerin üyeliği 1’e, küme merkezine uzaklaştıkça 0’a yaklaşır. Bu üyelik derecelerinin toplamı daima 1’e eşittir.



Şekil 5.12. Üyelik fonksiyonu

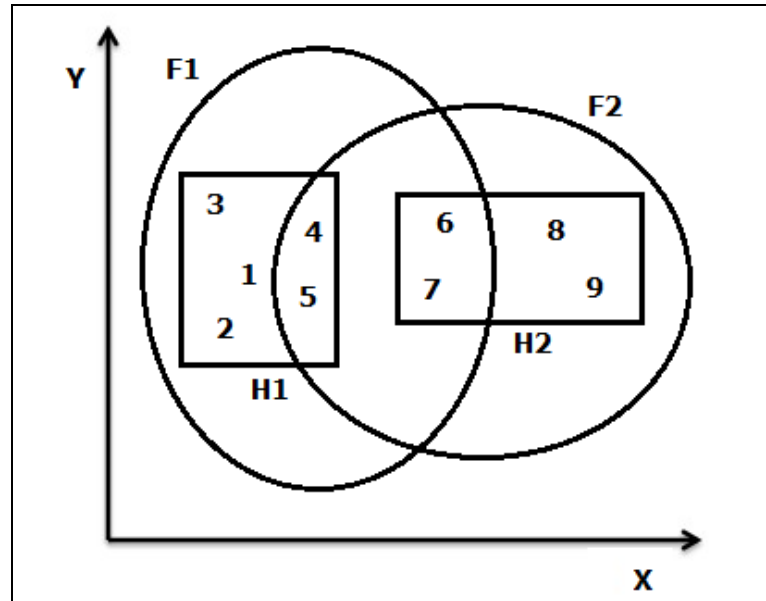
Klasik kesin kümeleme mantığına göre bir nesne sadece bir kümenin elemanı olabilirken, bulanık küme teorisine göre bir nesne aynı anda birden çok kümenin elemanı olabilmektedir. Bulanık kümeleme yöntemi, kümelerin birbirlerinden bariz bir şekilde ayıramadığı durumlarda kullanılır. Dolayısıyla klasik kesin kümeleme, bulanık kümelemenin daha sınırlı olan durumu niteliğindedir.

Bulanık mantık yaklaşımıyla en çok kullanılan kümeleme algoritması fuzzy c-means (bulanık c-ortalama) algoritmasıdır.

5.4.1. Fuzzy c-means algoritması

Fuzzy c-means (FCM) metodu, bulanık bölünmeli kümeleme yöntemlerinden arasında en bilinen ve en çok uygulanan tekniktir. Fuzzy c-means algoritması nesnenin iki ya da daha çok kümeye dâhil olabilmesine imkân tanır. Bulanık mantık prensibi gereği her bir nesne, kümelerin her birine arasında değişen birer üyelik değeri ile aittir. Bir nesnenin bütün kümelere olan üyelik değerlerinin toplamı “1” olmalıdır. Nesne kümelerin merkezlerinden hangisine daha yakınsa o kümeye ait olma üyelik değeri diğer kümelere ait olma üyelik değerinden daha büyük olacaktır. Bulanık kümeleme algoritmalarının çoğu amaç fonksiyonu tabanlıdır. Amaç fonksiyonun belirlenen minimum ilerleme değerine yaklaşmasıyla kümeleme işlemi tamamlanır.

Temelinde k-ortalamlar algoritmasına çok benzeyen fuzzy c-means algoritmasının k-ortalamalardan en önemli farkı (Şekil 5.13.)’de de görüldüğü gibi nesnelerin sadece tek bir kümeye ait olma zorunluluğunda olmamasıdır. H1 ve H2 dikdörtgenleri k-ortalamlar tarafından oluşturulmuş klasik kümeleri, F1 ve F2 elipsleri ise fuzzy c-means ile oluşturulmuş bulanık kümeleri temsil etmektedir. Nesnelerin bulanık kümelerdeki üyelik değerleri ise aşağıda tablo şeklinde verilmiştir (Çizelge 5.1.).



Şekil 5.13. Fuzzy c-means örnek kümeleme

Çizelge 5.1. Fuzzy c-means örnek kümelemesindeki nesnelerin üyelik değerleri

Nesneler	F1	F2
1	1	0
2	1	0
3	1	0
4	0,8	0,2
5	0,7	0,3
6	0,2	0,8
7	0,2	0,8
8	0	1
9	0	1

Fuzzy c-means algoritmasının adımları:

Adım 1: Başlangıç Değerlerini Girmek: Küme sayısı c , bulanıklık indeksi m , üyelik derecesi matrisi V ya da U küme prototipleri rastgele üretilir.

Adım 2: U küme prototipleri rastgele üretildiği varsayılır ise bu değerler kullanılarak üyelik derecesi matrisi hesaplanır.

Adım 3: U küme prototipleri güncellenir.

Adım 4: $|U(t) - U(t-1)| < \alpha$ ise algoritma durur aksi halde Adım 2'ye geri dönülür.

(α =işlem bitirme kriteri)

Fuzzy c-means algoritması uygulandıktan sonra bir nesnenin hangi kümeye dâhil olacağına karar vermek için üyelik derecelerine bakılır. Nesne, kümelerden hangisine olan üyelik değeri en büyük ise o kümeye dâhil olur. Ancak her bir nesne geri kalan kümelere de belli bir üyelikle girerler.

Fuzzy c-means algoritmalarının sonuçları başlangıçta üretilen rastgele değerlere oldukça bağlıdır ve bu nedenle rastgelelikten kaynaklanacak problemlerin ortadan kaldırılması için birtakım algoritmalar geliştirilmiştir.

Fuzzy c-means algoritmasının en temel özelliği olan üyelik matrisinin kümeleme üzerinde çok olumlu etkisi vardır. Bu matris belirsiz durumları tanımlamayı kolaylaştırmaktadır. Ayrıca üyelik dereceleri düşük olduğundan aykırı değerlerin

etkisi az olur. Çok esnek bir yapıdadır. Diğer bölünmeli algoritmalarla karşılaştırıldığında örtüşen kümeler bulma kabiliyeti diğerlerine kıyasla daha fazladır. Bütün bu saydığımız avantajların yanında fuzzy c-means algoritmasının bazı dezavantajları da mevcuttur. Üyelik fonksiyonları işlemsel karmaşıklığı fazlaştırdığı için zaman bakımından oldukça maliyetli sayılabilecek bir bölünmeli kümeleme yöntemidir. $n \times k$ boyutundaki üyelik matrisi nedeniyle algoritma çıkışı çok fazla değer içerir. Büyük miktarlardaki çıktı sonucun yorumlanmasını zorlaştırır, bu nedenle her nesne en fazla üyeliği olan kümeye dâhil edilerek kümeleme sonucu keskin kümelemeye dönüştürülür.

5.5. Grid Tabanlı Kümeleme Yöntemi

Grid yapı oluşacak şekilde nesneler uzayı sonlu sayıda hücreye bölünür. Kümeleme işlemlerinin tamamı bu grid yapı üzerinde yapılır. En önemli avantajı hızlı tamamlanabilmesi ve nesnelerin sayısından bağımsız olabilmesidir. Büyük veri setleri için oldukça kullanışlıdır. Aynı zamanda sıra dışı uç değerler barındıran veri setlerinde de verimli sonuçlar vermektedir. Grid tabanlı teknikle çalışan algoritmaların etkilerinin, önceden belirlenen grid yapıların büyüklüklerinden ve belli hücrelere ait eşik değerlerinden fazla şekilde etkilenmesi bu tekniğin en önemli dezavantajı olarak görülmektedir.

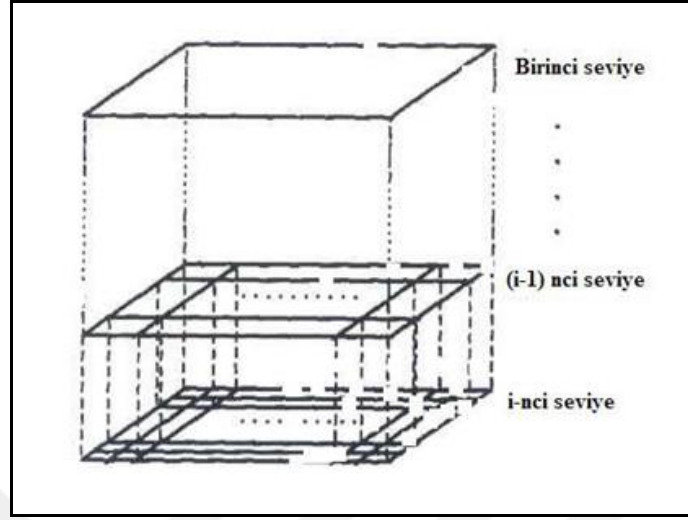
Grid bazlı yöntemlere örnek olarak STING, CLIQUE ve WaveCluster algoritmaları gösterilebilir.

5.5.1. STING algoritması

W. Wang, J. Yang ve R. Muntz tarafından 1997 yılında sunulan STING (STatistical INformation Grid) algoritması uzaysal bölgeyi dikdörtgen hücreler oluşacak şekilde böler ve hücre içindeki nesnelerin sayısal değerlerinin istatistiksel parametrelerini hesaplar.

Hiyerarşik yapının temelinde, genelde farklı seviyelerde hücreler bulunur (Şekil 5.14). Alt seviyede yer alan bir hücrenin oluşturulması, bir üst seviyesindeki hücrenin bölünmesiyle gerçekleşmektedir. En üst seviyede yer alan hücre veri

uzayına karşılık gelir. Alt seviye hücrelerin büyüklükleri nesnenin yoğunluğuna bağlıdır (Wang vd., 1997).



Şekil 5.14. STING algoritmasının hiyerarşik yapısı

STING algoritmasının adımları (Akın, 2008);

- Adım 1: Başlangıç seviyesi belirlenir.
- Adım 2: Başlangıç seviyesinde yer alan tüm hücrelerin, güven aralığı ve olasılıkları gibi istatistiki bilgiler hücreyle ilişkilendirilir.
- Adım 3: Başlangıç seviyesi eğer alt seviyeyse adımlar devam edilir, aksi halde üst seviyedeki ilişkili hücreler alt hücelere bölünmesi için ikinci adıma tekrar dönülür.
- Adım 4: Tüm ilişkili hücreler için, geçerli hücreyle kalan tüm hücrelerin uzaklıkları incelenir.
- Adım 5: Birbirleriyle ilişkisiz hücre kalmayncaya dek dördüncü adım tekrarlanır ve artık yalnızca tek bir küme ortaya çıkar.
- Adım 6: Tüm ilişkili hücreler atanıncaya kadar dördüncü adım ile beşinci adım tekrarlanır.

STING algoritmasının avantajları şöyle sıralanabilir;

1. Hücrelerde saklanan istatistiksel bilgiler, sorgulama olmaksızın bağımsız olarak grid hücresinde yer alan verilerin özetleyici bilgilerini temsil eder.

2. Grid yapı, kademeli güncellemeyle paralel işlemciliğin yapılmasında kolaylık sağlamaktadır.
3. En büyük avantajı olarak da istatistiki bilgilerin elde edilmesinde, veri setinin bir kere okunması algoritmanın etkin olmasına neden olmaktadır.

5.5.2. CLIQUE algoritması

Grid tabanlı ve yoğunluğa dayalı yöntemleri bir arada birleştiren CLIQUE (Clustering High-Dimensional Space) algoritması ilk olarak SIGMOD 1998 konferasında R. Agrawal, J. Gehrke, D. Gunopulos ve P. Raghavan tarafından sunulmuştur.

CLIQUE algoritması, verilerin dengesiz dağıldığı büyük veri tabanlarında yoğun ve aykırı alanları tespit ederek genel bir dağılım ortaya çıkartır. Önce kümeleri içinde barındıran alt uzayları, daha sonra da kümeleri tanımlar. Çok yoğun kümelere sahip alt uzayları otomatik olarak bulur. Veri uzayını hücelere böler ve yoğun noktaların tahminine göre hücrelerdeki noktaların sayısını hesaplar. Algoritmanın uygulaması oldukça basit olsa da elde ettiği sonuçlar karmaşık ve güvenilirliği şüpheli olabilmektedir (Akın, 2008).

6. UYGULAMA

6.1. Giriş

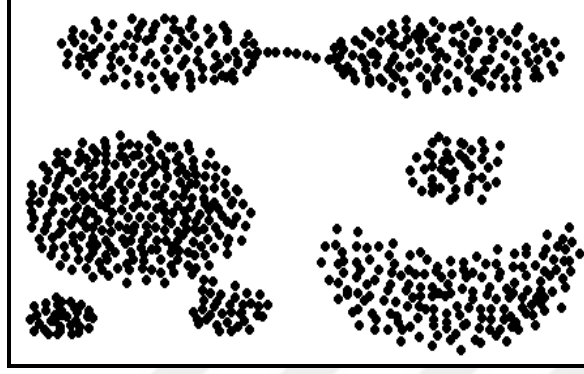
Bu bölümde, üçüncü bölümde açıklanan kümeleme analizi tekniklerini ve dördüncü bölümde açıklanan hiyerarşik kümeleme yöntemlerinden AGNES, DIANA; bölünmeli kümeleme yöntemlerinden k-ortalamalar, CLARA, PAM; yoğunluğa dayalı kümeleme yöntemlerinden DBSCAN; bulanık kümeleme yöntemlerinden fuzzy c-means kümeleme algoritmaları kullanarak dokuz sentetik veri seti ve Ek B.'de verilen gerçek veri seti üzerinde uygulamalar yapılmıştır. Uygulama R programlama dili kullanılarak gerçekleştirilmiştir. Çalışmada kullanılan her bir algoritma için yazılan kodlar Ek A.'da verilmiştir. Uygulama sonuçları R'da grafiksel olarak gösterilmiştir. Veri setleri için her bir algoritmanın sonuçlarını istatistiksel karşılıklarını görmek için karışıklık matrisleri oluşturulmuş ve bu matrislerin köşegen toplamalarının, tüm matrise bölümünden elde edilen algoritmaların doğruluk oranları tespit edilmiş. R grafikleriyle de görsel olarak algoritmalar arasındaki kümeleme farklılıkları ortaya konmuştur.

6.2. İki Değişkenli Sentetik Veri Setleri Uygulaması

Dokuz sentetik veri seti üzerinde AGNES, DIANA, k-ortalamalar, CLARA, PAM, DBSCAN kümeleme algoritmaları uygulanmıştır. Kullanılan veri setleri <http://cs.joensuu.fi/sipu/datasets/> adresinden temin edilmiştir. Veri setleri iki değişkenden ve kümeleme analizi örneklerinde kullanılmaya uygun tasarlanan, gözle kolayca ayırt edebilmemize olanak sağlayan farklı şekil ve büyüklüklere sahip kümelerden oluşmaktadır. Ayrıca tüm veri setlerinde yer alan kümeler başlangıçta bellidir ve algoritmaların oluşturduğu kümeler ile farklılıkları karşılık matrislerinde verilmiştir.

6.2.1. Aggregation veri seti

Farklı büyüklük ve şekildeki yedi küme, iki değişken ve 788 satırdan oluşan veri bir setidir.



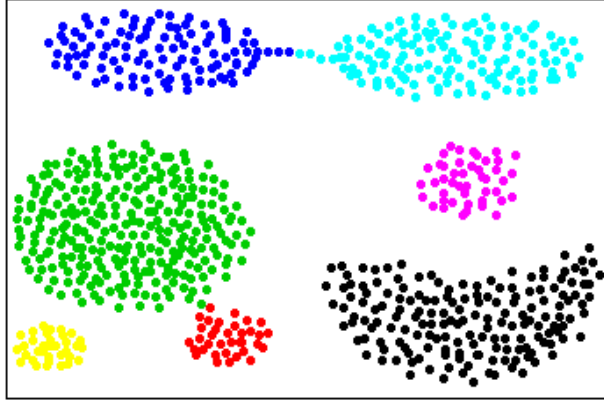
Şekil 6.1. Aggregation veri seti

Aggregation veri seti farklı büyüklük ve şekilde kümeler içerdiği için bölünmeli ve hiyerarşik bölünme algoritmaları kümeleri doğru tespit edememişlerdir. Bu veri seti için en uygun kümeleme yöntemi olan yoğunluğa dayalı kümeleme yöntemlerinden DBSCAN algoritması en başarılı sonucu vermiştir.

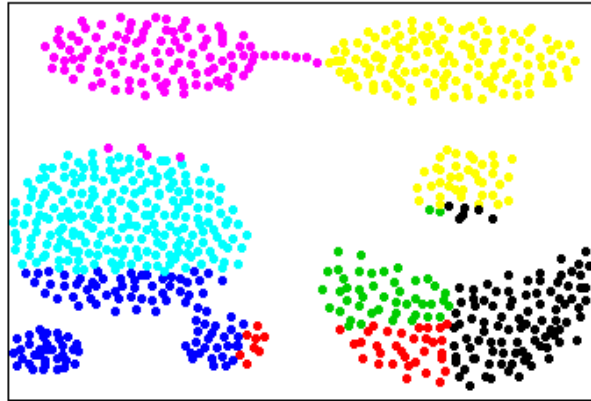
Algoritmaların bu veri seti üzerinde ulaştıkları doğruluk oranlarına aşağıdaki tabloda yer verilmiştir (Çizelge 6.1.). Aggregation veri seti için elde edilen tüm algoritma sonuçları grafiksel olarak gösterilmiştir (Şekil 6.2.).

Çizelge 6.1. Aggregation veri seti için algoritmaların doğruluk oranları

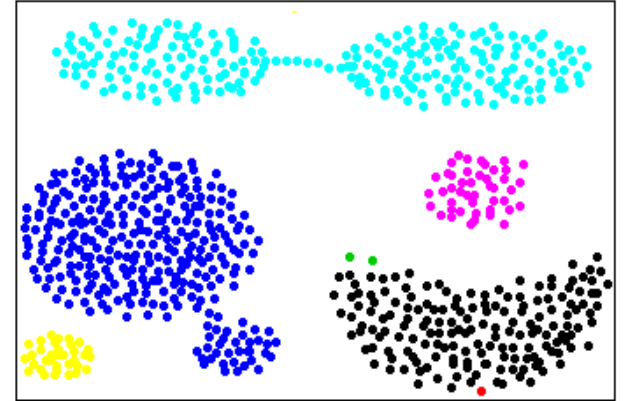
Algoritmalar	Doğru kümelenen nokta sayısı	Doğruluk oranı
DBSCAN	786	%99,74
AGNES	649	%82,36
PAM	613	%77,79
K-ortalamalar	610	%77,41
DIANA	573	%72,71
CLARA	568	%72,08



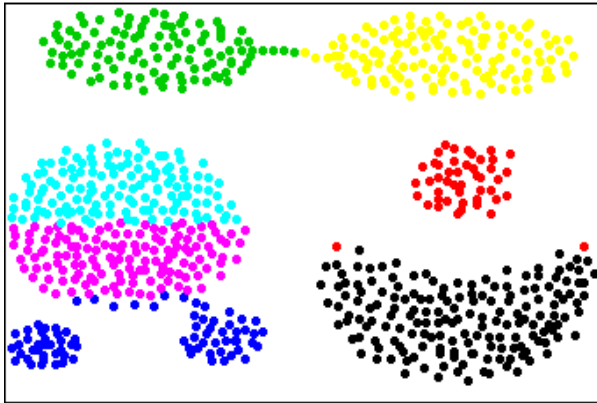
DBSCAN algoritması



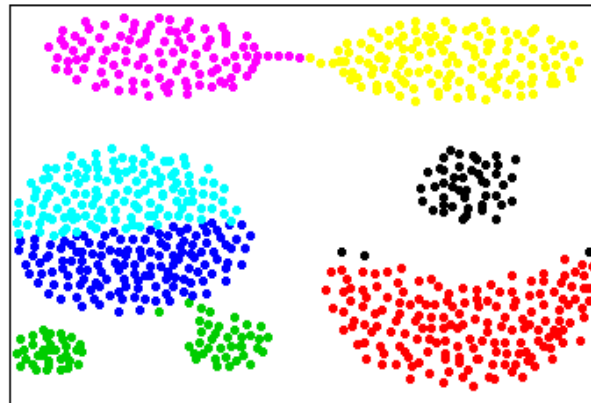
DIANA algoritması



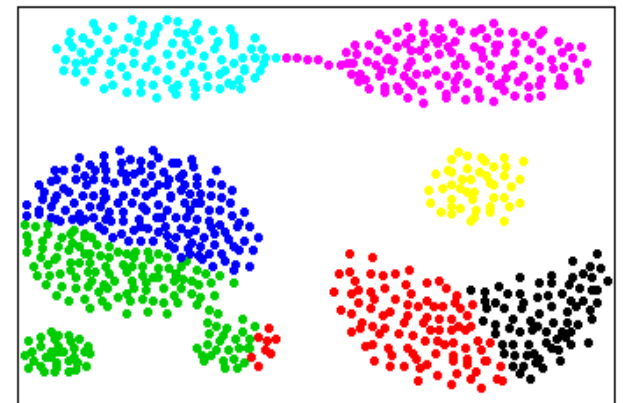
AGNES algoritması



K-ortalamlar algoritması



PAM algoritması



CLARA algoritması

Şekil 6.2. Aggregation veri seti için kümeleme sonuçları

İstatistiksel olarak da bu sonuçları göstermek için kullanılan karışıklık matrisi sonuçları aşağıdaki gibidir:

(Çizelge 6.2.)’deki karışıklık matrisinde köşegenler DBSCAN algoritmasının doğru olarak belirlediği küme sayılarını göstermektedir. Bu veri seti için DBSCAN algoritmasının doğruluk oranı $786/788 = 0.9974$ (%99,74) olarak hesaplanmıştır

Çizelge 6.2. Aggregation veri seti için DBSCAN algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme	Orijinal 6. küme	Orijinal 7. küme
Tahminlenen 1. küme	45	0	0	0	0	0	0
Tahminlenen 2. küme	0	170	0	0	0	0	0
Tahminlenen 3. küme	0	0	102	0	0	1	0
Tahminlenen 4. küme	0	0	0	272	0	0	0
Tahminlenen 5. küme	0	0	0	0	34	0	0
Tahminlenen 6. küme	0	0	0	0	0	129	0
Tahminlenen 7. küme	0	0	0	1	0	0	34

Aggregation veri seti için AGNES algoritmasının doğruluk oranı $649/788 = 0,8236$ (%82,36) olarak hesaplanmıştır.

Çizelge 6.3. Aggregation veri seti için AGNES algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme	Orijinal 6. küme	Orijinal 7. küme
Tahminlenen 1. küme	45	0	0	0	0	0	0
Tahminlenen 2. küme	0	167	0	0	0	0	0
Tahminlenen 3. küme	0	2	102	0	0	0	0
Tahminlenen 4. küme	0	0	0	273	0	0	34
Tahminlenen 5. küme	0	0	0	0	34	0	0
Tahminlenen 6. küme	0	0	102	0	0	130	0
Tahminlenen 7. küme	0	1	0	0	0	0	0

Aggregation veri seti için PAM algoritmasının doğruluk oranı $613/788 = 0,7779$ (%77,79) olarak hesaplanmıştır.

Çizelge 6.4. Aggregation veri seti için PAM algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme	Orijinal 6. küme	Orijinal 7. küme
Tahminlenen 1. küme	45	3	0	0	0	0	0
Tahminlenen 2. küme	0	167	0	0	0	0	0
Tahminlenen 3. küme	0	0	102	0	0	2	0
Tahminlenen 4. küme	0	0	0	137	0	0	0
Tahminlenen 5. küme	0	0	0	132	0	0	0
Tahminlenen 6. küme	0	0	0	0	0	128	0
Tahminlenen 7. küme	0	0	0	4	34	0	34

Aggregation veri seti için k-ortalamlar algoritmasının doğruluk oranı $610/788 = 0,7741$ (%77,41) olarak hesaplanmıştır.

Çizelge 6.5. Aggregation veri seti için k-ortalamlar algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme	Orijinal 6. küme	Orijinal 7. küme
Tahminlenen 1. küme	45	2	0	0	0	0	0
Tahminlenen 2. küme	0	168	0	0	0	0	0
Tahminlenen 3. küme	0	0	102	0	0	2	0
Tahminlenen 4. küme	0	0	0	133	0	0	0
Tahminlenen 5. küme	0	0	0	131	0	0	0
Tahminlenen 6. küme	0	0	0	0	0	128	0
Tahminlenen 7. küme	0	0	0	9	34	0	34

Aggregation veri seti için DIANA algoritmasının doğruluk oranı $573/788 = 0,7271$ (%72,71) olarak hesaplanmıştır.

Çizelge 6.6. Aggregation veri seti için DIANA algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme	Orijinal 6. küme	Orijinal 7. küme
Tahminlenen 1. küme	2	44	0	0	0	0	0
Tahminlenen 2. küme	6	94	0	0	0	0	0
Tahminlenen 3. küme	0	0	102	4	0	4	0
Tahminlenen 4. küme	0	0	0	215	0	0	0
Tahminlenen 5. küme	0	0	0	54	34	0	26
Tahminlenen 6. küme	37	0	0	0	0	126	0
Tahminlenen 7. küme	0	32	0	0	0	0	8

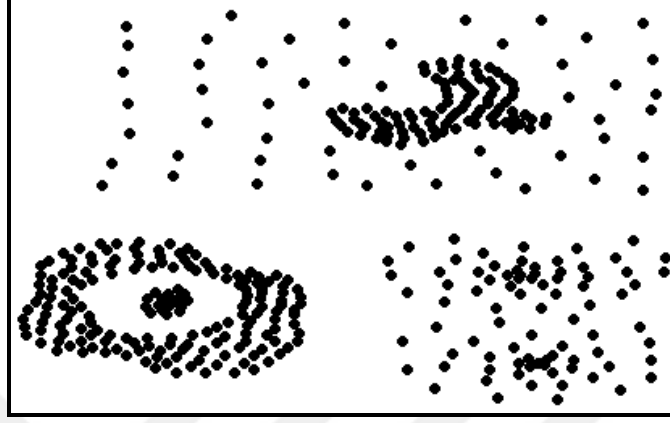
Aggregation veri seti için CLARA algoritmasının doğruluk oranı $568/788 = 0,7208$ (%72,08) olarak hesaplanmıştır.

Çizelge 6.7. Aggregation veri seti için CLARA algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme	Orijinal 6. küme	Orijinal 7. küme
Tahminlenen 1. küme	45	2	0	0	0	0	0
Tahminlenen 2. küme	0	94	0	0	0	0	8
Tahminlenen 3. küme	0	0	101	0	0	0	0
Tahminlenen 4. küme	0	0	0	164	0	0	0
Tahminlenen 5. küme	0	0	0	109	34	0	26
Tahminlenen 6. küme	0	0	1	0	0	130	0
Tahminlenen 7. küme	0	76	0	0	0	0	0

6.2.2. Compound veri seti

Farklı şekil, büyüklük ve yoğunluktaki altı küme, iki değişken ve 399 satırdan oluşan bir veri setidir.



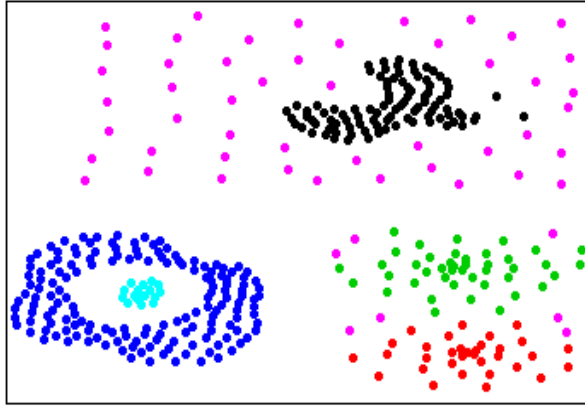
Şekil 6.3. Compound veri seti

Compound veri seti farklı büyüklük, şekil ve yoğunluktaki kümeler içerdiği için yoğunluğa dayalı kümeleme yöntemlerinden olan DBSCAN algoritması başarılı şekilde sonuç vermiştir. Fakat bu veri setinde bölünmeli ve hiyerarşik bölümlemeli algoritmalar kümeleri doğru tespit edememişlerdir.

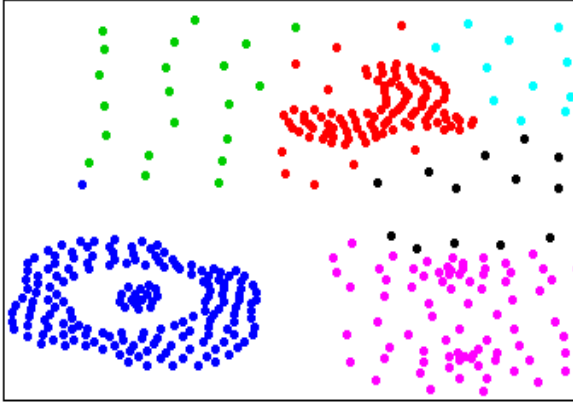
Algoritmaların bu veri seti üzerinde ulaştıkları doğruluk oranlarına aşağıdaki tabloda yer verilmiştir (Çizelge 6.8.). Compound veri seti için elde edilen tüm algoritma sonuçları grafiksel olarak gösterilmiştir (Şekil 6.4.).

Çizelge 6.8. Compound veri seti için algoritmaların doğruluk oranları

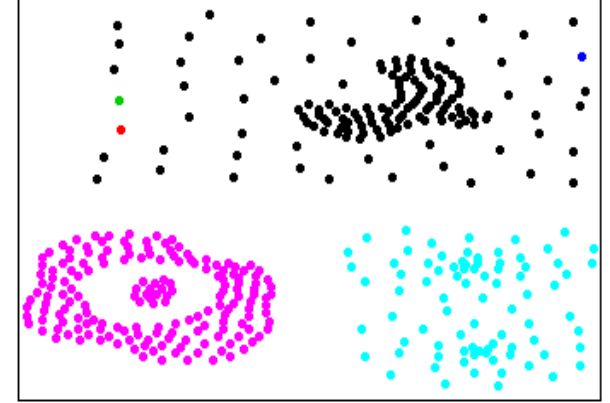
Algoritmalar	Doğru kümelene nokta sayısı	Doğruluk oranı
DBSCAN	389	%97,49
K-ortalamalar	305	%76,44
DIANA	298	%74,68
AGNES	296	%74,18
CLARA	271	%67,91
PAM	255	%63,90



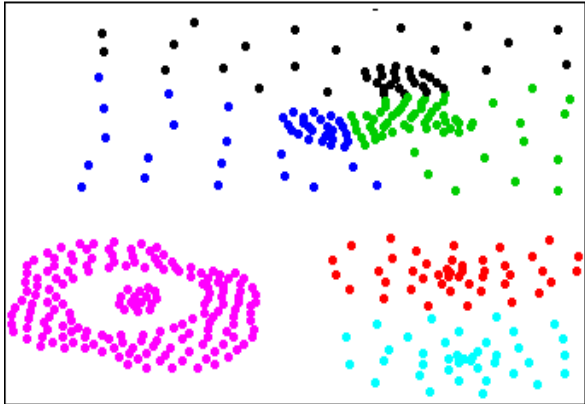
DBSCAN algoritması



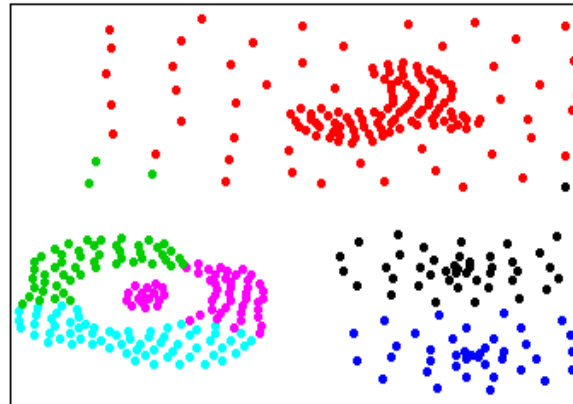
DIANA algoritması



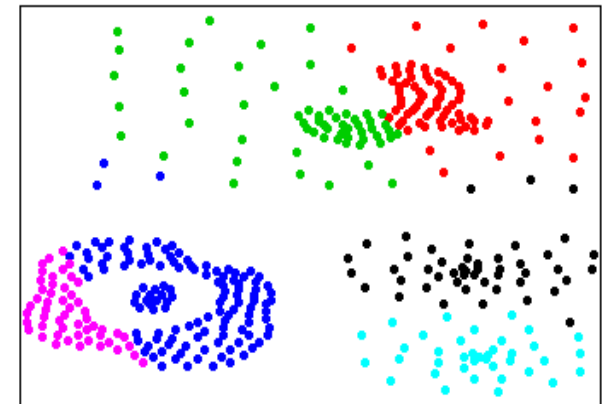
AGNES algoritması



K-ortalamalar algoritması



PAM algoritması



CLARA algoritması

Şekil 6.4. Compound veri seti için kümeleme sonuçları

İstatistiksel olarak da bu sonuçları göstermek için kullanılan karışıklık matrisi sonuçları aşağıdaki gibidir:

Compound veri seti için DBSCAN algoritmasının doğruluk oranı $389/399 = 0,9749$ (%97,49) olarak hesaplanmıştır.

Çizelge 6.9. Compound veri seti için DBSCAN algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme	Orijinal 6. küme
Tahminlenen 1. küme	48	0	4	3	0	0
Tahminlenen 2. küme	2	92	0	0	0	0
Tahminlenen 3. küme	0	0	33	0	0	0
Tahminlenen 4. küme	0	0	1	42	0	0
Tahminlenen 5. küme	0	0	0	0	158	0
Tahminlenen 6. küme	0	0	0	0	0	16

Compound veri seti için k-ortalamlar algoritmasının doğruluk oranı $305/399 = 0,7644$ (%76,44) olarak hesaplanmıştır.

Çizelge 6.10. Compound veri seti için k-ortalamlar algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme	Orijinal 6. küme
Tahminlenen 1. küme	19	22	0	5	0	0
Tahminlenen 2. küme	13	46	0	0	0	0
Tahminlenen 3. küme	0	0	38	1	0	0
Tahminlenen 4. küme	0	0	0	44	0	0
Tahminlenen 5. küme	1	0	0	0	158	16
Tahminlenen 6. küme	18	24	0	0	0	0

Compound veri seti için DIANA algoritmasının doğruluk oranı $298/399 = 0,7468$ (%74,68) olarak hesaplanmıştır.

Çizelge 6.11. Compound veri seti için DIANA algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme	Orijinal 6. küme
Tahminlenen 1. küme	8	0	0	5	0	0
Tahminlenen 2. küme	9	92	0	0	0	0
Tahminlenen 3. küme	21	0	0	0	0	0
Tahminlenen 4. küme	0	0	38	40	0	0
Tahminlenen 5. küme	1	0	0	0	158	16
Tahminlenen 6. küme	11	0	0	0	0	0

Compound veri seti için AGNES algoritmasının doğruluk oranı $296/399 = 0,7418$ (%74,18) olarak hesaplanmıştır.

Çizelge 6.12. Compound veri seti için AGNES algoritmalarının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme	Orijinal 6. küme
Tahminlenen 1. küme	1	0	0	0	0	0
Tahminlenen 2. küme	47	92	0	0	0	0
Tahminlenen 3. küme	1	0	0	0	0	0
Tahminlenen 4. küme	0	0	38	45	0	0
Tahminlenen 5. küme	0	0	0	0	158	16
Tahminlenen 6. küme	1	0	0	0	0	0

Compound veri seti için CLARA algoritmasının doğruluk oranı $271/399 = 0,6791$ (%67,91) olarak hesaplanmıştır.

Çizelge 6.13. Compound veri seti için CLARA algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme	Orijinal 6. küme
Tahminlenen 1. küme	26	34	0	0	0	0
Tahminlenen 2. küme	18	58	0	0	0	0
Tahminlenen 3. küme	0	0	37	1	0	0
Tahminlenen 4. küme	3	0	1	44	0	0
Tahminlenen 5. küme	3	0	0	0	106	16
Tahminlenen 6. küme	0	0	0	0	52	0

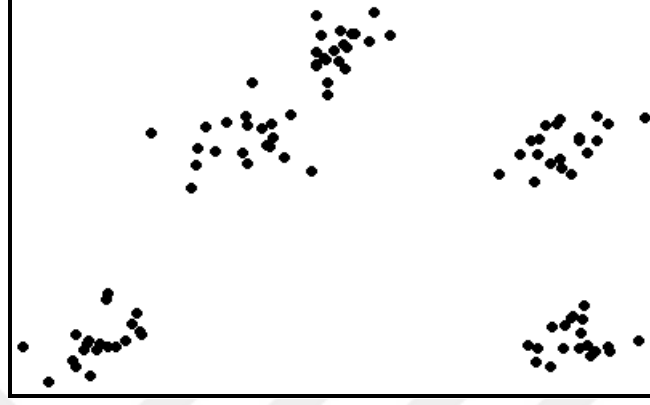
Compound veri seti için PAM algoritmasının doğruluk oranı $255/399 = 0,6390$ (%63,90) olarak hesaplanmıştır.

Çizelge 6.14. Compound veri seti için PAM algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme	Orijinal 6. küme
Tahminlenen 1. küme	0	0	0	0	51	0
Tahminlenen 2. küme	46	92	0	0	0	0
Tahminlenen 3. küme	0	0	38	1	0	0
Tahminlenen 4. küme	1	0	0	44	0	0
Tahminlenen 5. küme	3	0	0	0	65	0
Tahminlenen 6. küme	0	0	0	0	42	16

6.2.3. Sentetik 1 veri seti

Konveks olmayan eşit büyüklükte beş küme, iki değişken ve 100 satırdan oluşan bir veri seti.



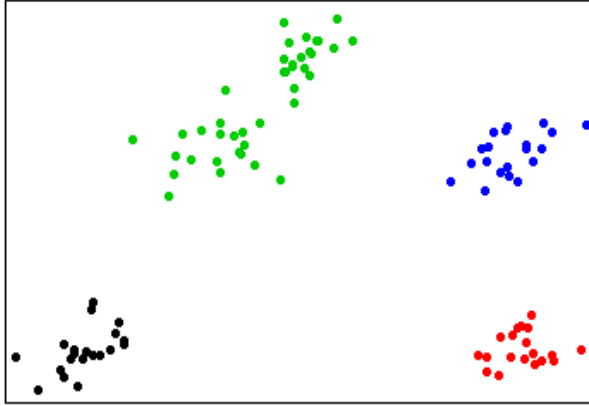
Şekil 6.5. Sentetik 1 veri seti

Uygulamada kullanılan algoritmalar, kümeler arası mesafenin, dolayısıyla da farklılıkların diğer veri setlerine göre nispeten daha fazla olan bu veri seti üzerinde başarılı sonuçlar elde edebilmişlerdir.

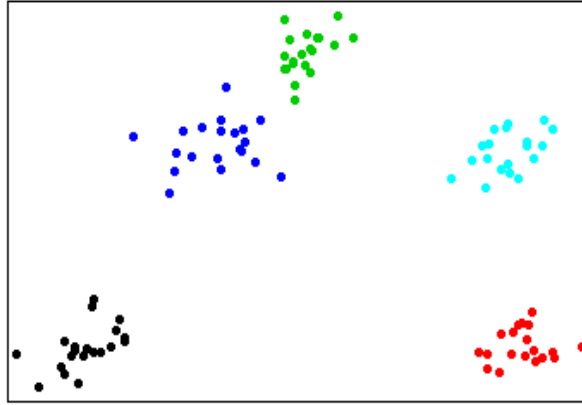
Algoritmaların bu veri seti üzerinde ulaştıkları doğruluk oranlarına aşağıdaki tabloda yer verilmiştir (Çizelge 6.15.). Sentetik 1 veri seti için elde edilen tüm algoritma sonuçları grafiksel olarak gösterilmiştir (Şekil 6.6.).

Çizelge 6.15. Sentetik 1 veri seti için algoritmaların doğruluk oranları

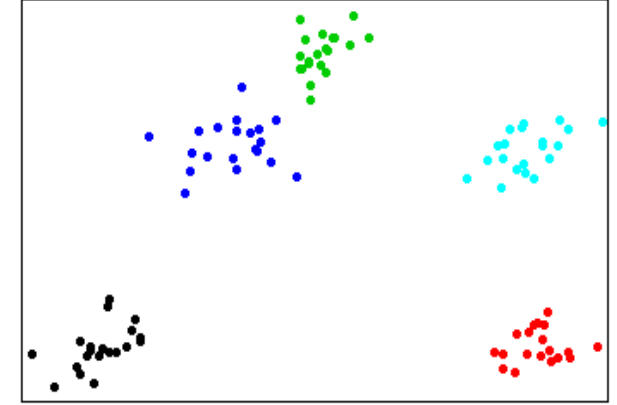
Algoritmalar	Doğru kümelenen nokta sayısı	Doğruluk oranı
DIANA	100	%100
CLARA	100	%100
AGNES	100	%100
PAM	99	%99
DBSCAN	80	%80
K-ortalamalar	74	%74



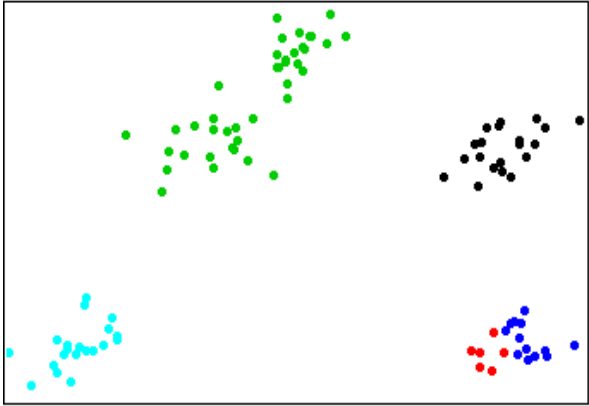
DBSCAN algoritması



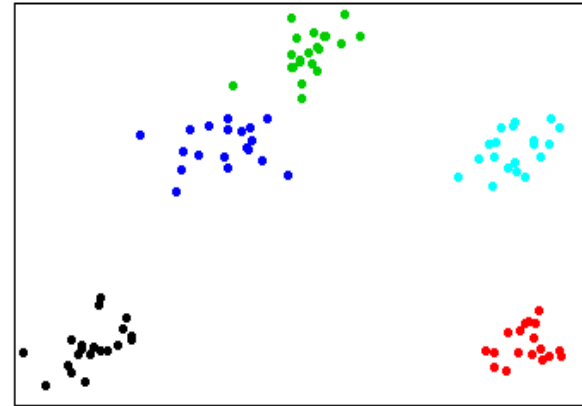
DIANA algoritması



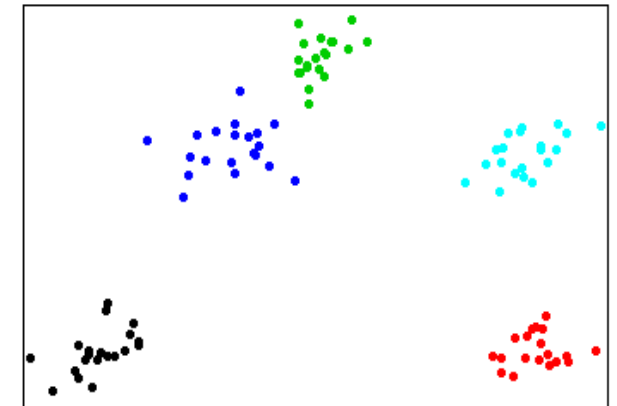
AGNES algoritması



K-ortalamlar algoritması



PAM algoritması



CLARA algoritması

Şekil 6.6. Sentetik 1 veri seti için kümeleme sonuçları

İstatistiksel olarak da bu sonuçları göstermek için kullanılan karışıklık matrisi sonuçları aşağıdaki gibidir:

Sentetik 1 veri seti için DIANA, AGNES ve CLARA algoritmalarının doğruluk oranları $100/100=1$ (%100) olarak hesaplanmıştır.

Çizelge 6.16. Sentetik 1 veri seti için DIANA, AGNES ve CLARA algoritmalarının karışıklık matrisleri

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme
Tahminlenen 1. küme	20	0	0	0	0
Tahminlenen 2. küme	0	20	0	0	0
Tahminlenen 3. küme	0	0	20	0	0
Tahminlenen 4. küme	0	0	0	20	0
Tahminlenen 5. küme	0	0	0	0	20

Sentetik 1 veri seti için PAM algoritmasının doğruluk oranı $99/100=0,99$ (%99) olarak hesaplanmıştır.

Çizelge 6.17. Sentetik 1 veri seti için PAM algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme
Tahminlenen 1. küme	20	0	0	0	0
Tahminlenen 2. küme	0	20	0	0	0
Tahminlenen 3. küme	0	0	20	1	0
Tahminlenen 4. küme	0	0	0	19	0
Tahminlenen 5. küme	0	0	0	0	20

Sentetik 1 veri seti için DBSCAN algoritmasının doğruluk oranı $80/100=0,8$ (%80) olarak hesaplanmıştır.

Çizelge 6.18. Sentetik 1 veri seti için DBSCAN algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme
Tahminlenen 1. küme	45	0	0	0	0
Tahminlenen 2. küme	0	170	0	0	0
Tahminlenen 3. küme	0	0	102	0	0
Tahminlenen 4. küme	0	0	0	272	0
Tahminlenen 5. küme	0	0	0	0	34

Sentetik 1 veri seti için k-ortalamalar algoritmasının doğruluk oranı $74/100 = 0,74$ (%74) olarak hesaplanmıştır.

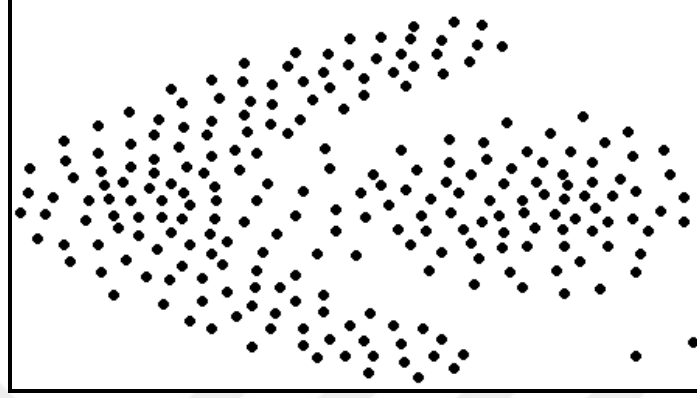
Çizelge 6.19. Sentetik 1 veri seti için k-ortalamalar algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme	Orijinal 4. küme	Orijinal 5. küme
Tahminlenen 1. küme	20	0	0	0	0
Tahminlenen 2. küme	0	14	0	0	0
Tahminlenen 3. küme	0	0	20	20	0
Tahminlenen 4. küme	0	6	0	0	0
Tahminlenen 5. küme	0	0	0	0	20

(Şekil 6.6.)’da sentetik 1 veri seti için elde edilen tüm algoritma sonuçları görülmektedir.

6.2.4. Flame veri seti

Farklı yoğunlukta ve şekilde konveks olan iki küme, iki değişken ve 240 satırdan oluşan bir veri setidir.



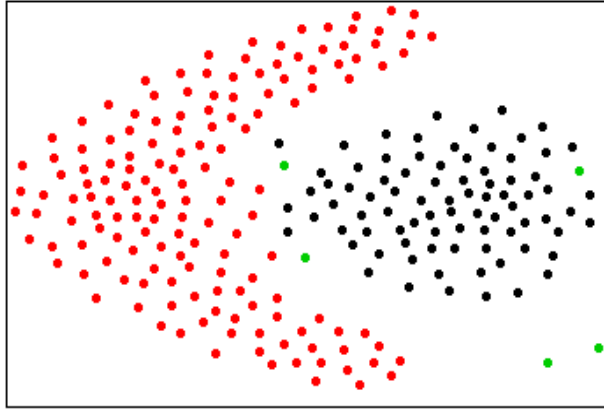
Şekil 6.7. Flame veri seti

Flame veri seti farklı yoğunlukta veriler içerdiği için yoğunluğa dayalı kümeleme yöntemlerinden olan DBSCAN algoritması bu veri setinde en başarılı sonucu verdiği görülmektedir.

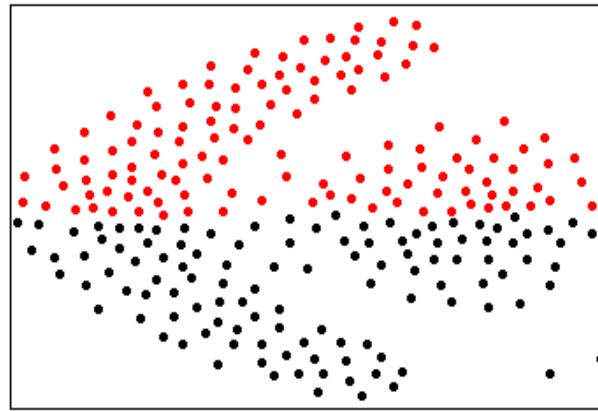
Algoritmaların bu veri seti üzerinde ulaştıkları doğruluk oranlarına aşağıdaki tabloda yer verilmiştir (Çizelge 6.20.). Flame veri seti için elde edilen tüm algoritma sonuçları grafiksel olarak gösterilmiştir (Şekil 6.8.).

Çizelge 6.20. Flame veri seti için algoritmaların doğruluk oranları

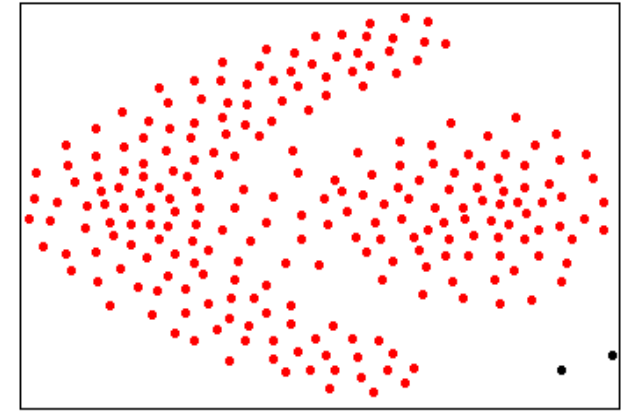
Algoritmalar	Doğru kümelenen nokta sayısı	Doğruluk oranı
DBSCAN	235	%97,91
CLARA	207	%86,25
PAM	204	%85
K-ortalamalar	201	%83,75
AGNES	155	%64,58
DIANA	126	%52,50



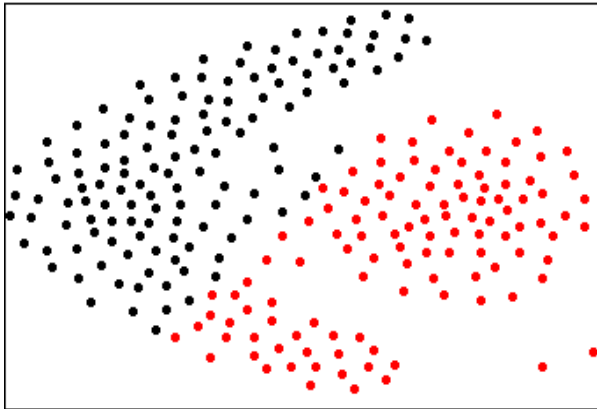
DBSCAN algoritması



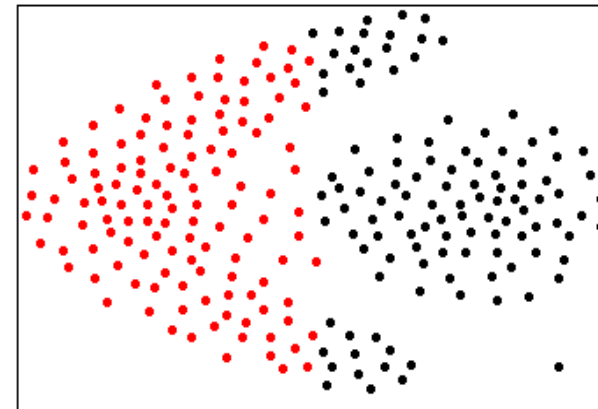
DIANA algoritması



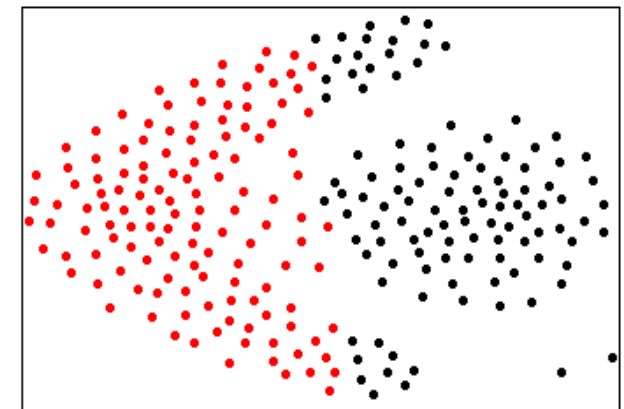
AGNES algoritması



K-ortalamlar algoritması



PAM algoritması



CLARA algoritması

Şekil 6.8. Flame veri seti için kümeleme sonuçları

İstatistiksel olarak da bu sonuçları göstermek için kullanılan karışıklık matrisi sonuçları aşağıdaki gibidir:

Flame veri seti için DBSCAN algoritmasının doğruluk oranı $235/240 = 0,9791$ (%97,91) olarak hesaplanmıştır.

Çizelge 6.21. Flame veri seti için DBSCAN algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme
Tahminlenen 1. küme	84	2
Tahminlenen 2. küme	3	151

Flame veri seti için CLARA algoritmasının doğruluk oranı $207/240 = 0,8625$ (%89,25) olarak hesaplanmıştır.

Çizelge 6.22. Flame veri seti için CLARA algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme
Tahminlenen 1. küme	82	28
Tahminlenen 2. küme	5	125

Flame veri seti için PAM algoritmasının doğruluk oranı $204/240 = 0,8500$ (%85,00) olarak hesaplanmıştır.

Çizelge 6.23. Flame veri seti için PAM algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme
Tahminlenen 1. küme	83	32
Tahminlenen 2. küme	4	121

Flame veri seti için k-ortalamlar algoritmasının doğruluk oranı $201/240 = 0,8375$ (%83,75) olarak hesaplanmıştır.

Çizelge 6.24. Flame veri seti için k-ortalamlar algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme
Tahminlenen 1. küme	81	33
Tahminlenen 2. küme	6	120

Flame veri seti için AGNES algoritmasının doğruluk oranı $155/240 = 0,6458$ (%64,58) olarak hesaplanmıştır.

Çizelge 6.25. Flame veri seti için AGNES algoritmalarının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme
Tahminlenen 1. küme	2	0
Tahminlenen 2. küme	85	153

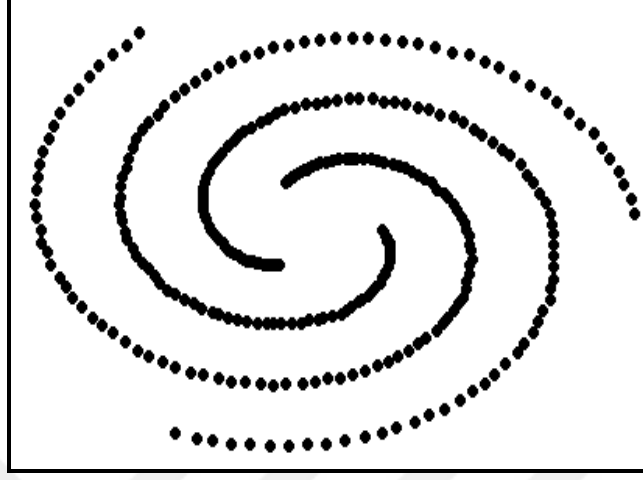
Flame veri seti için DIANA algoritmasının doğruluk oranı $126/240 = 0,5250$ (%52,50) olarak hesaplanmıştır.

Çizelge 6.26. Flame veri seti için DIANA algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme
Tahminlenen 1. küme	43	70
Tahminlenen 2. küme	44	83

6.2.5. Pathbased 2 spiral veri seti

Spiral şekilli 3 küme, iki değişken ve 312 satırdan oluşan bir setidir.



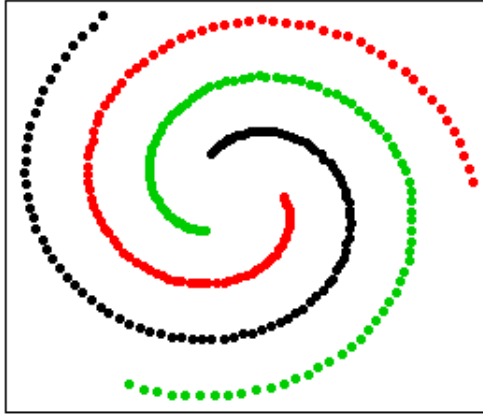
Şekil 6.9. Pathbased 2 spiral veri seti

Yoğunluk tabanlı DBSCAN ve hiyerarşik kümeleme yöntemlerinden AGNES algoritmalarının bu veri seti için %100 başarılı sonuçlar verdiği görülmektedir. Özellikle küme merkezi belirleyerek kümeleme yapan bölünmeli kümeleme yöntemleri gibi öncelik kıstası noktaların birbirine olan uzaklıkları olan algoritmalar, bu veri setinde yanlış kümeleme yapılacağı ön görülmelidir. Bu yüzden diğer algoritmaların ise başarı oranları oldukça düşük çıkmıştır.

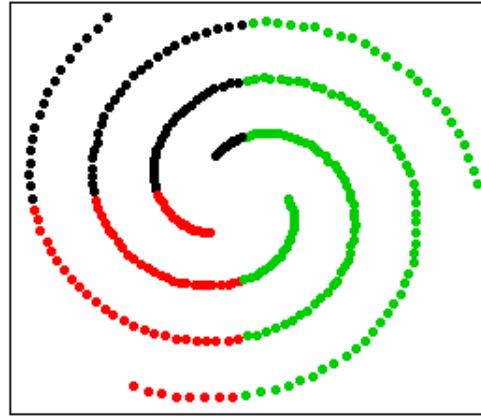
Flame veri seti için elde edilen tüm algoritma sonuçları grafiksel olarak gösterilmiştir (Şekil 6.10.).

Çizelge 6.27. Pathbased 2 spiral veri seti için algoritmaların doğruluk oranları

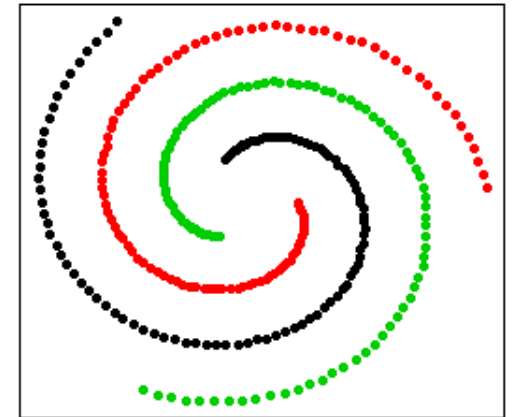
Algoritmalar	Doğru kümelenen nokta sayısı	Doğruluk oranı
AGNES	312	%100
DBSCAN	312	%100
CLARA	113	%36,21
K-ortalamalar	106	%33,97
PAM	106	%33,97
DIANA	105	%33,65



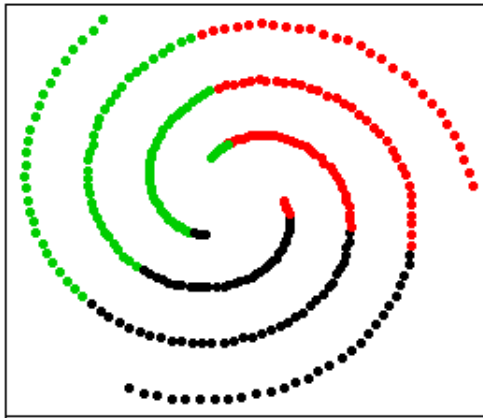
DBSCAN algoritması



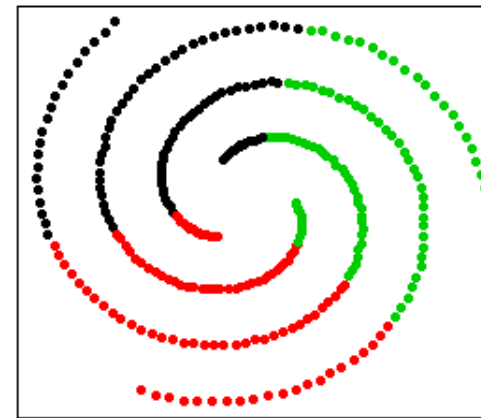
DIANA algoritması



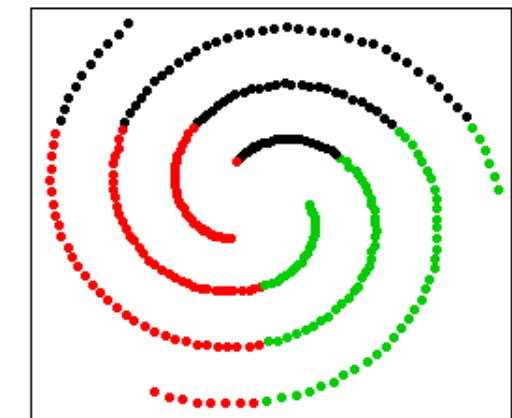
AGNES algoritması



K-ortalamalar algoritması



PAM algoritması



CLARA algoritması

Şekil 6.10. Pathbased 2 spiral veri seti için kümeleme sonuçları

İstatistiksel olarak da bu sonuçları göstermek için kullanılan karışıklık matrisi sonuçları aşağıdaki gibidir:

Pathbased 2 spiral veri seti için DBSCAN ve AGNES algoritmalarının doğruluk oranları $312/312=1$ (%100) olarak hesaplanmıştır.

Çizelge 6.28. Pathbased 2 spiral veri seti için DBSCAN ve AGNES algoritmalarının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme
Tahminlenen 1. küme	101	0	0
Tahminlenen 2. küme	0	105	0
Tahminlenen 3. küme	0	0	106

Pathbased 2 spiral veri seti için CLARA algoritmasının doğruluk oranı $113/312=0,3621$ (%36,21) olarak hesaplanmıştır.

Çizelge 6.29. Pathbased 2 spiral veri seti için CLARA algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme
Tahminlenen 1. küme	35	27	39
Tahminlenen 2. küme	35	44	33
Tahminlenen 3. küme	31	34	34

Pathbased 2 spiral veri seti için PAM algoritmasının doğruluk oranı $106/312=0,3397$ (%33,97) olarak hesaplanmıştır.

Çizelge 6.30. Pathbased 2 spiral veri seti için PAM algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme
Tahminlenen 1. küme	34	33	38
Tahminlenen 2. küme	35	39	35
Tahminlenen 3. küme	32	33	33

Pathbased 2 spiral veri seti için k-ortalamlar algoritmasının doğruluk oranı $106/312 = 0,3397$ (%33,97) olarak hesaplanmıştır.

Çizelge 6.31. Pathbased 2 spiral veri seti için k-ortalamlar algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme
Tahminlenen 1. küme	35	36	35
Tahminlenen 2. küme	33	33	33
Tahminlenen 3. küme	33	36	38

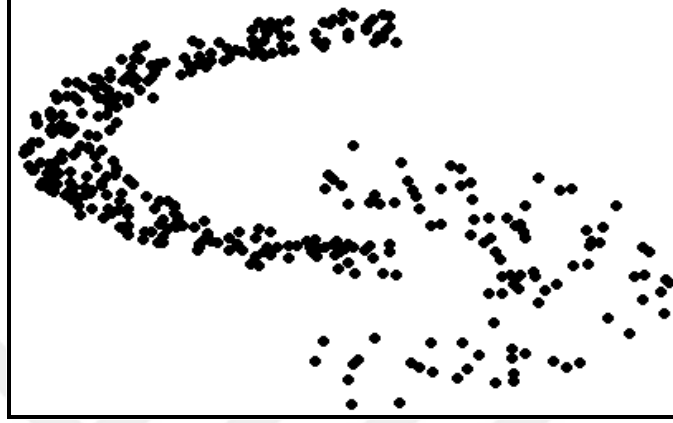
Pathbased 2 spiral veri seti için DIANA algoritmasının doğruluk oranı $105/312 = 0,3365$ (%33,65) olarak hesaplanmıştır.

Çizelge 6.32. Pathbased 2 spiral veri seti için DIANA algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme
Tahminlenen 1. küme	26	25	32
Tahminlenen 2. küme	26	30	25
Tahminlenen 3. küme	49	50	49

6.2.6. Jain veri seti

Farklı yoğunlukta ve şekilde konveks olan iki küme, iki değişken ve 373 satırdan oluşan bir veri setidir.



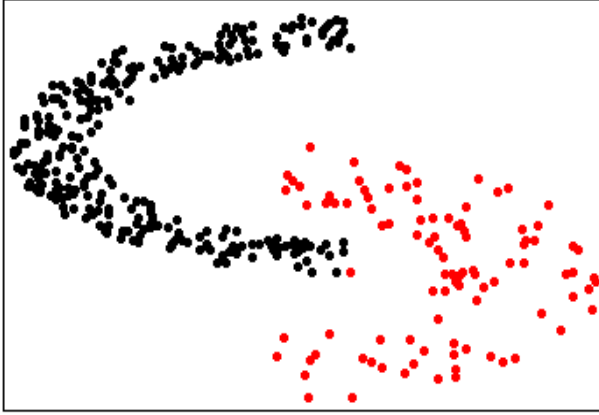
Şekil 6.11. Jain veri seti

Jain veri setinden elde edilen grafikler incelendiğinde, veri setinin yoğunluk tabanlı olmasından dolayı DBSCAN algoritmasının oldukça başarılı, diğer algoritmaların ise başarısız oldukları görülmektedir.

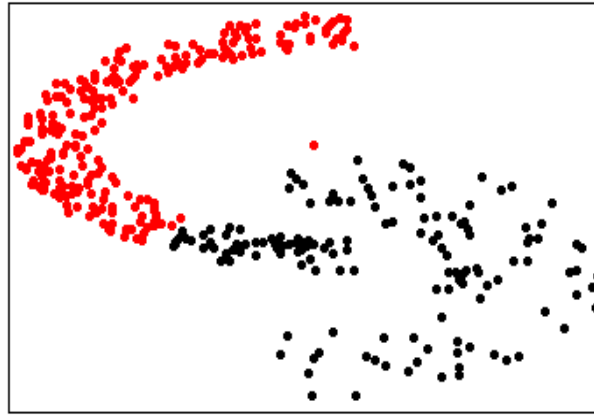
Algoritmaların bu veri seti üzerinde ulaştıkları doğruluk oranlarına aşağıdaki tabloda yer verilmiştir (Çizelge 6.33.). Jain veri seti için elde edilen tüm algoritma sonuçları grafiksel olarak gösterilmiştir (Şekil 6.12.).

Çizelge 6.33. Jain veri seti için algoritmaların doğruluk oranları

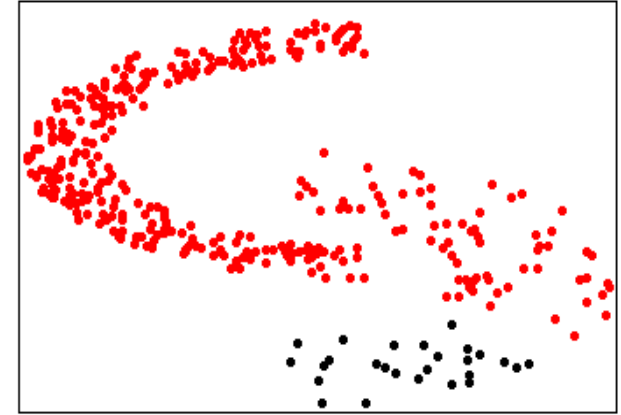
Algoritmalar	Doğru kümelenen nokta sayısı	Doğruluk oranı
PAM	372	%99,73
CLARA	320	%85,70
DBSCAN	304	%81,50
K-ortalamlar	302	%80,96
DIANA	292	%78,28
AGNES	282	%75,60



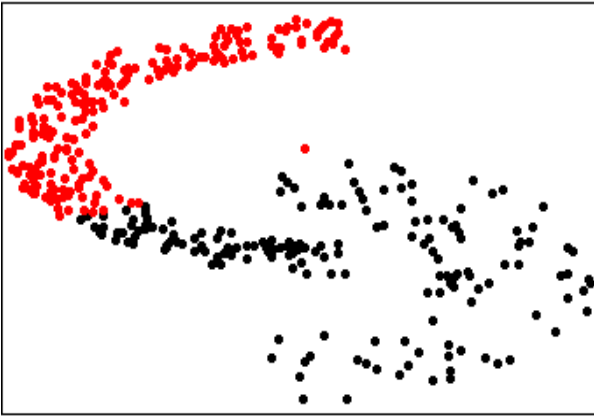
DBSCAN algoritması



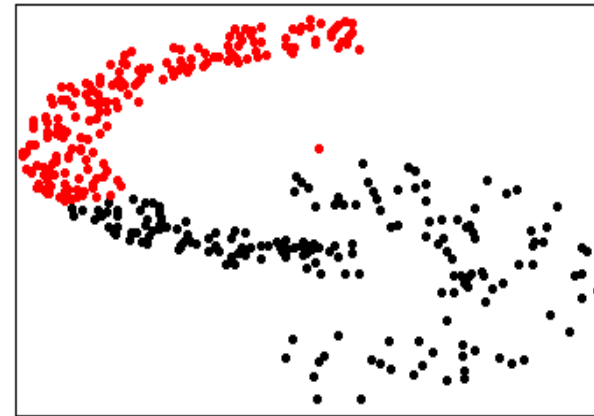
DIANA algoritması



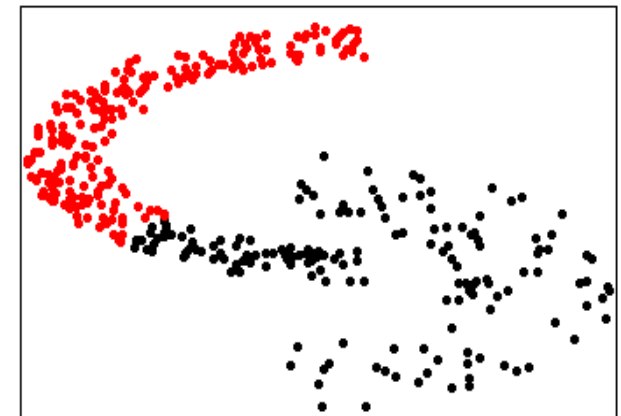
AGNES algoritması



K-ortalamalar algoritması



PAM algoritması



CLARA algoritması

Şekil 6.12. Jain veri seti için kümeleme sonuçları

İstatistiksel olarak da bu sonuçları göstermek için kullanılan karışıklık matrisi sonuçları aşağıdaki gibidir:

Jain veri seti için DBSCAN algoritmasının doğruluk oranı $372/373 = 0,9973$ (%99,73) olarak hesaplanmıştır.

Çizelge 6.34. Jain veri seti için DBSCAN algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme
Tahminlenen 1. küme	275	0
Tahminlenen 2. küme	1	97

Jain veri seti için DIANA algoritmasının doğruluk oranı $320/373 = 0,8579$ (%85,79) olarak hesaplanmıştır.

Çizelge 6.35. Jain veri seti için DIANA algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme
Tahminlenen 1. küme	224	1
Tahminlenen 2. küme	52	96

Jain veri seti için CLARA algoritmasının doğruluk oranı $304/373 = 0,8150$ (%81,50) olarak hesaplanmıştır.

Çizelge 6.36. Jain veri seti için CLARA algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme
Tahminlenen 1. küme	207	0
Tahminlenen 2. küme	69	97

Jain veri seti için AGNES algoritmasının doğruluk oranı $302/373 = 0,8096$ (%80,96) olarak hesaplanmıştır.

Çizelge 6.37. Jain veri seti için AGNES algoritmalarının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme
Tahminlenen 1. küme	276	71
Tahminlenen 2. küme	0	26

Jain veri seti için k-ortalamlar algoritmasının doğruluk oranı $292/373 = 0,7828$ (%78,28) olarak hesaplanmıştır.

Çizelge 6.38. Jain veri seti için k-ortalamlar algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme
Tahminlenen 1. küme	196	1
Tahminlenen 2. küme	80	96

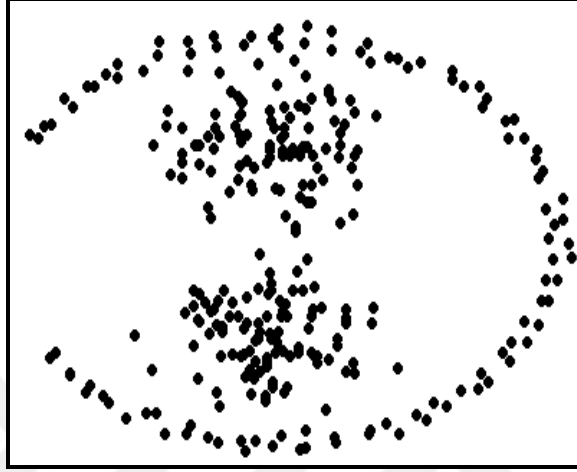
Jain veri seti için PAM algoritmasının doğruluk oranı $282/373 = 0,7560$ (%75,60) olarak hesaplanmıştır.

Çizelge 6.39. Jain veri seti için PAM algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme
Tahminlenen 1. küme	186	1
Tahminlenen 2. küme	90	96

6.2.7. Pathbased 1 veri seti

Farklı şekil ve yoğunlukta üç küme, iki değişken ve 300 satırdan oluşan bir veri setidir.



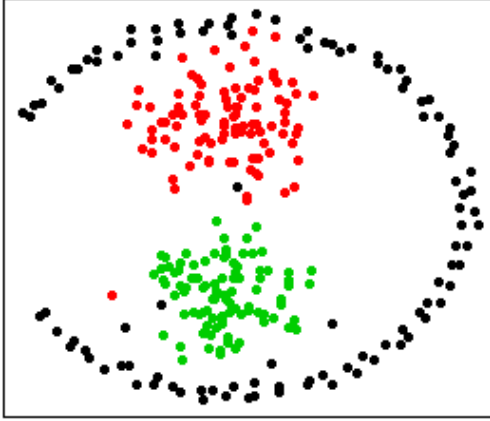
Şekil 6.13. Pathbased 1 veri seti

Pathbased 1 veri setinden elde edilen grafikler incelendiğinde DBSCAN algoritmasının en başarılı, AGNES algoritmasının en başarısız, diğer algoritmaların da birbirine yakın sonuçlar verdiği görülmektedir.

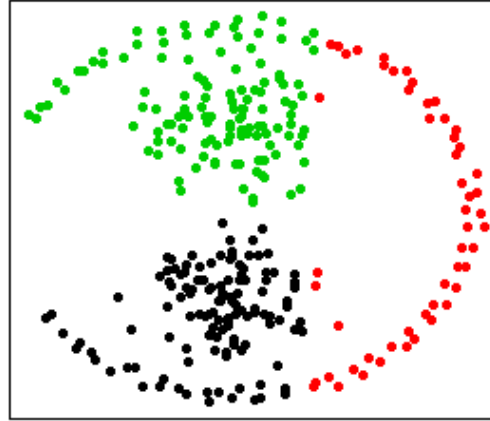
Algoritmaların bu veri seti üzerinde ulaştıkları doğruluk oranlarına aşağıdaki tabloda yer verilmiştir (Çizelge 6.40.). Pathbased 1 veri seti için elde edilen tüm algoritma sonuçları grafiksel olarak gösterilmiştir (Şekil 6.14.).

Çizelge 6.40. Pathbased 1 veri seti için algoritmaların doğruluk oranları

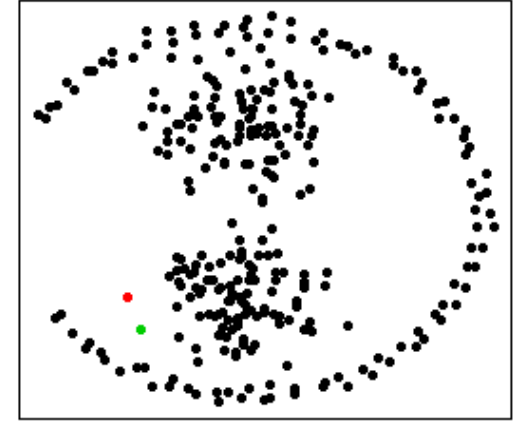
Algoritmalar	Doğru kümelenen nokta sayısı	Doğruluk oranı
DBSCAN	290	%96,66
DIANA	234	%78
CLARA	225	%75
K-ortalamalar	223	%74,33
PAM	222	%74
AGNES	111	%37



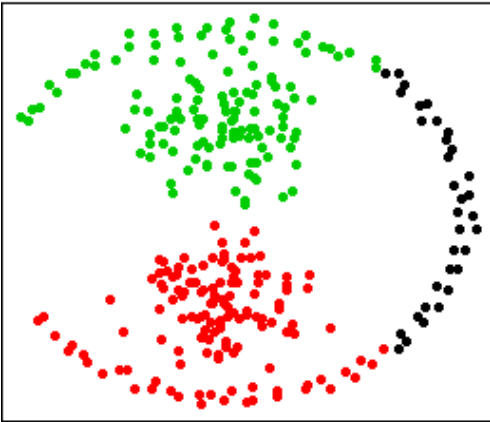
DBSCAN algoritması



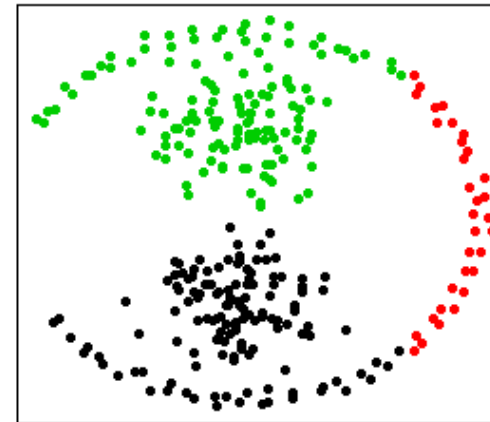
DIANA algoritması



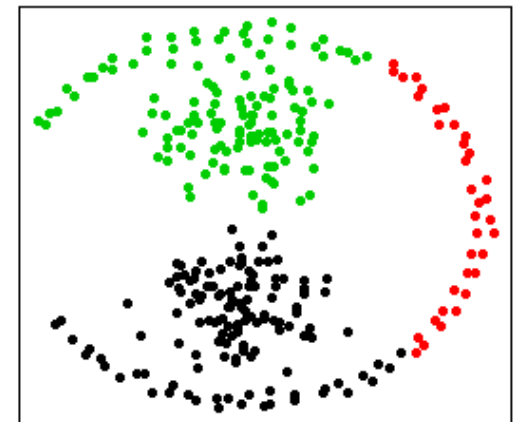
AGNES algoritması



K-ortalamalar algoritması



PAM algoritması



CLARA algoritması

Şekil 6.14. Pathbased 1 veri seti için kümeleme sonuçları

İstatistiksel olarak da bu sonuçları göstermek için kullanılan karışıklık matrisi sonuçları aşağıdaki gibidir:

Pathbased 1 veri seti için DBSCAN algoritmasının doğruluk oranı $290/300 = 0,9666$ (%96,66) olarak hesaplanmıştır.

Çizelge 6.41. Pathbased 1 veri seti için DBSCAN algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme
Tahminlenen 1. küme	104	3	1
Tahminlenen 2. küme	6	94	0
Tahminlenen 3. küme	0	0	92

Pathbased 1 veri seti için DIANA algoritmasının doğruluk oranı $234/300 = 0,7800$ (%78) olarak hesaplanmıştır.

Çizelge 6.42. Pathbased 1 veri seti için DIANA algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme
Tahminlenen 1. küme	48	3	1
Tahminlenen 2. küme	27	94	0
Tahminlenen 3. küme	35	0	92

Pathbased 1 veri seti için CLARA algoritmasının doğruluk oranı $225/300 = 0,7500$ (%75) olarak hesaplanmıştır.

Çizelge 6.43. Pathbased 1 veri seti için CLARA algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme
Tahminlenen 1. küme	35	0	0
Tahminlenen 2. küme	36	97	0
Tahminlenen 3. küme	35	0	93

Pathbased 1 veri seti için k-ortalamlar algoritmasının doğruluk oranı $223/300=0,7433$ (%74,33) olarak hesaplanmıştır.

Çizelge 6.44. Pathbased 1 veri seti için k-ortalamlar algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme
Tahminlenen 1. küme	33	0	0
Tahminlenen 2. küme	42	97	0
Tahminlenen 3. küme	35	0	93

Pathbased 1 veri seti için PAM algoritmasının doğruluk oranı $222/300=0,7400$ (%74) olarak hesaplanmıştır.

Çizelge 6.45. Pathbased 1 veri seti için PAM algoritmasının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme
Tahminlenen 1. küme	32	0	0
Tahminlenen 2. küme	42	97	0
Tahminlenen 3. küme	36	0	93

Pathbased 1 veri seti için AGNES algoritmasının doğruluk oranı $111/300=0,3700$ (%37) olarak hesaplanmıştır.

Çizelge 6.46. Pathbased 1 veri seti için AGNES algoritmalarının karışıklık matrisi

Kümeler	Orijinal 1. küme	Orijinal 2. küme	Orijinal 3. küme
Tahminlenen 1. küme	110	95	93
Tahminlenen 2. küme	0	1	0
Tahminlenen 3. küme	0	1	0

6.2.8. D31 veri seti

Eşit büyüklükte konveks olmayan 31 küme, iki değişken ve 3100 satırdan oluşan bir veri setidir.



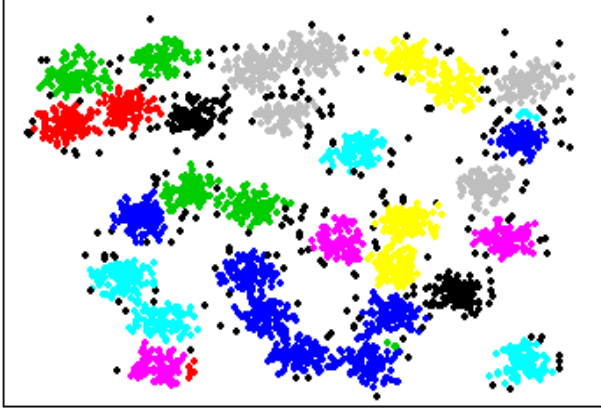
Şekil 6.15. D31 veri seti

D31 veri setinden elde edilen grafikler (Şekil 6.16.) incelendiğinde, CLARA algoritması ise konveks olmayan çok büyük veri setlerinde başarılı sonuçlar verdiğinden bu veri setini en iyi çözümleyen algoritma olmuştur. PAM algoritması büyük veri setlerinde meydana gelen hesaplama karmaşıklığı yüzünden bu veri setinde çalışmamaktadır.

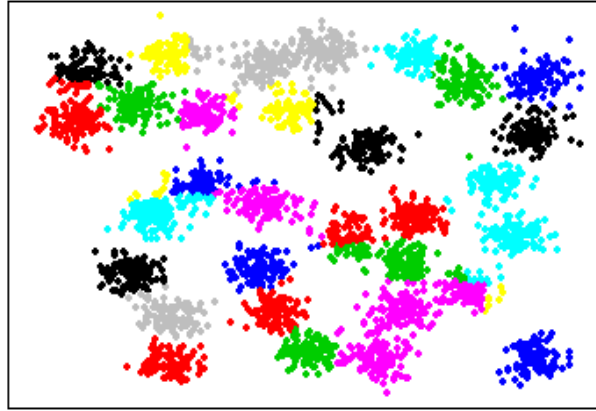
Algoritmaların bu veri seti üzerinde ulaştıkları doğruluk oranlarına aşağıdaki tabloda yer verilmiştir (Çizelge 6.47.). D31 veri seti kullanılan diğer veri setlerine göre çok daha büyük bir veri setidir. 31 kümeden oluşan bu veri setinin karışıklık matrisleri 31×31 boyutunda olması nedeniyle burada yer verilmemiştir.

Çizelge 6.47. D31 veri seti için algoritmaların doğruluk oranları

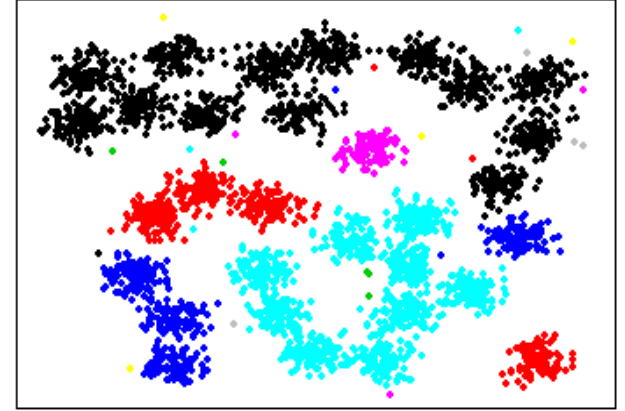
Algoritmalar	Doğru kümelenen nokta sayısı	Doğruluk oranı
CLARA	2969	%95,77
K-ortalamalar	2708	%87,35
DIANA	2691	%86,80
DBSCAN	2448	%78,96
AGNES	1031	%33,25
PAM	0	%0



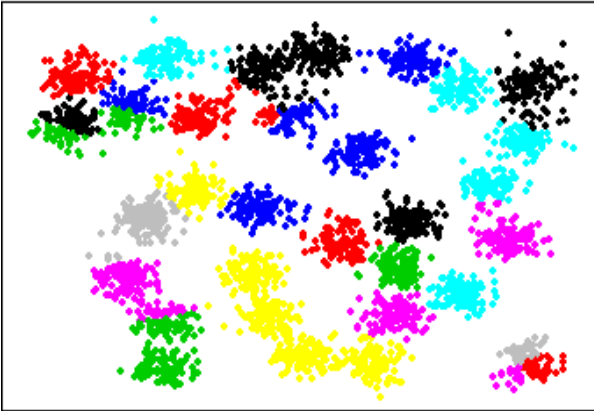
DBSCAN algoritması



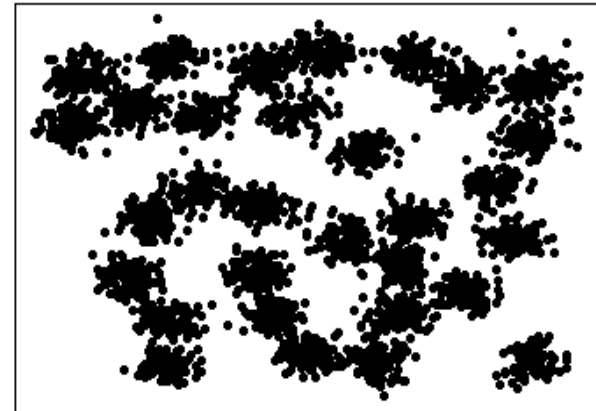
DIANA algoritması



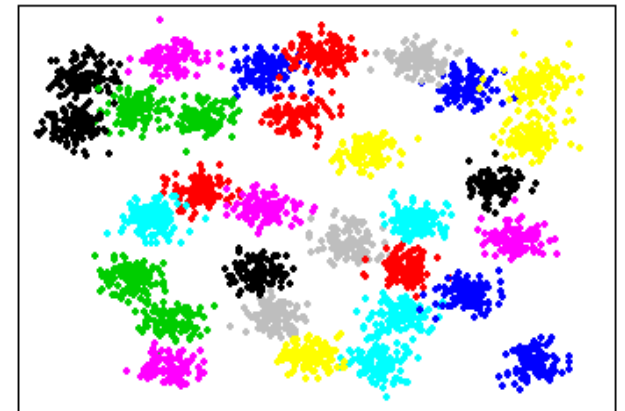
AGNES algoritması



K-ortalamalar algoritması



PAM algoritması

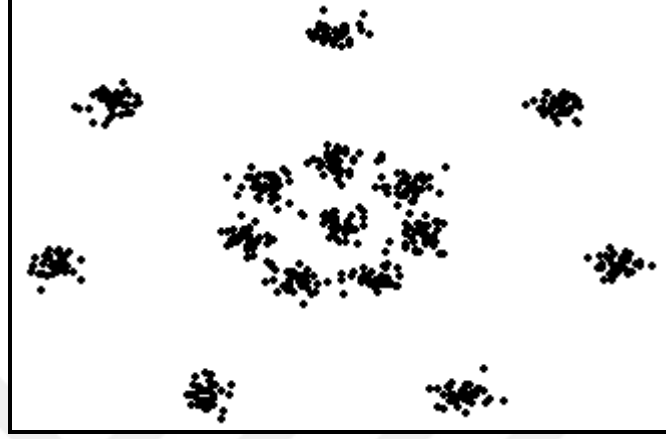


CLARA algoritması

Şekil 6.16. D31 veri seti için kümeleme sonuçlar

6.2.9. R15 veri seti

Eşit büyüklükte konveks olmayan on beş küme, iki değişken ve 600 satırdan oluşan bir veri setidir.



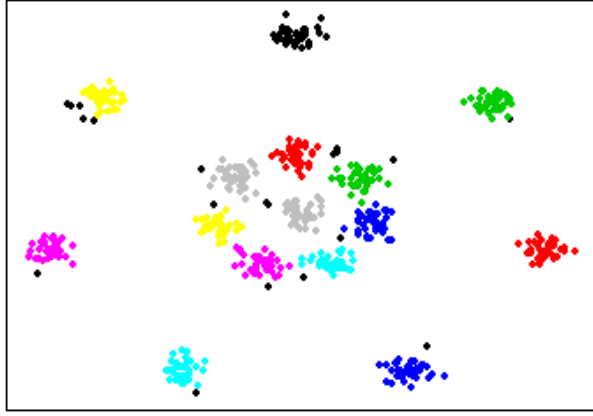
Şekil 6.17. R15 veri seti

R15 veri setinden elde edilen grafikler (Şekil 6.18.) incelendiğinde, eşit büyüklük ve yoğunluktaki konveks olmayan şekildeki kümeler içeren bu veri setinde bölünmeli kümeleme yöntemi algoritmalarından olan PAM ve CLARA algoritması en başarılı sonuçları vermişlerdir.

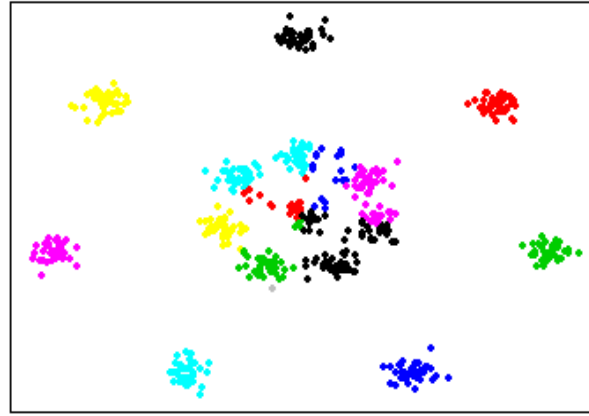
Algoritmaların bu veri seti üzerinde ulaştıkları doğruluk oranlarına aşağıdaki tabloda yer verilmiştir (Çizelge 6.48.). R15 veri seti, 15 kümeden oluşan bir veri setidir ve dolayısıyla karışıklık matrislerinin 15×15 boyutunda olması nedeniyle burada yer verilmemiştir.

Çizelge 6.48. R15 veri seti için algoritmaların doğruluk oranları

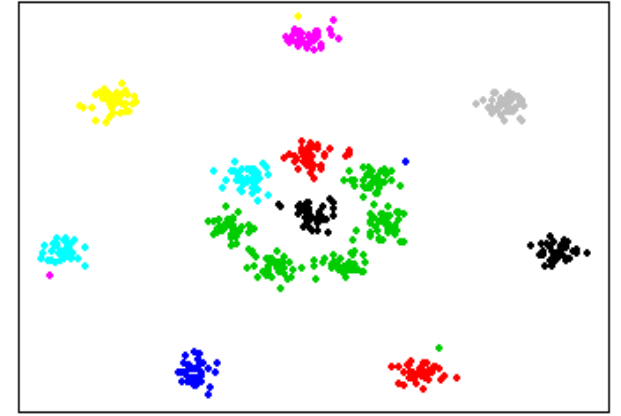
Algoritmalar	Doğru kümelene nokta sayısı	Doğruluk oranı
PAM	598	%99,66
CLARA	597	%99,50
DBSCAN	566	%94,33
K-ortalamlar	485	%80,83
DIANA	463	%77,16
AGNES	436	%72,66



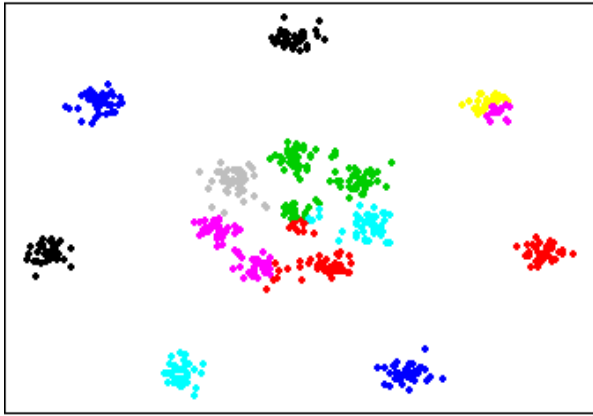
DBSCAN algoritması



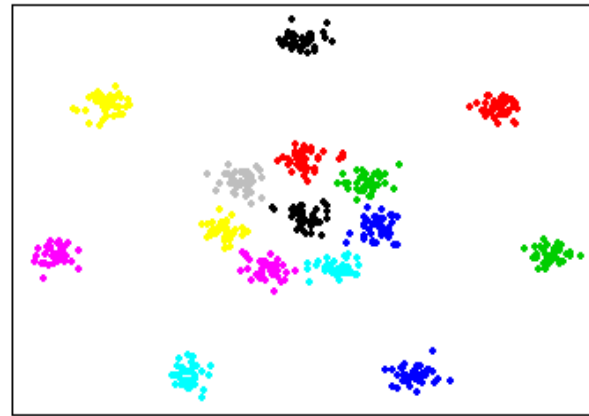
DIANA algoritması



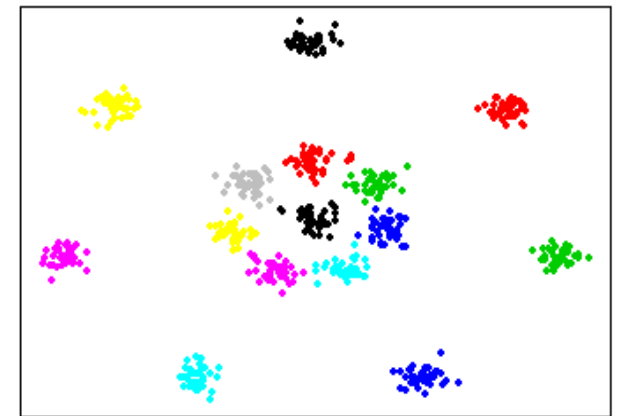
AGNES algoritması



K-ortalamalar algoritması



PAM algoritması



CLARA algoritması

Şekil 6.18. R15 veri seti için kümeleme sonuçları

6.3. Çok Değişkenli Gerçek Veri Seti Uygulaması

6.3.1. Seeds veri seti

Uygulamada kullanılan gerçek veri seti, “seeds”, Kama, Rosa ve Kanada buğday türlerine ait olan tohumların geometrik özelliklerin ölçümlerinden oluşmaktadır. Ek B.’de yer verilen seeds gerçek veri seti; eşit büyüklükte üç kümeden, yedi değişkenden ve 210 satırdan oluşmaktadır. Bu veri seti <http://archive.ics.uci.edu/ml/datasets/seeds> adresinden temin edilmiştir.

6.3.2. Materyal ve yöntem

6.3.2.1. Değişken ölçümlerinin elde edilmesi

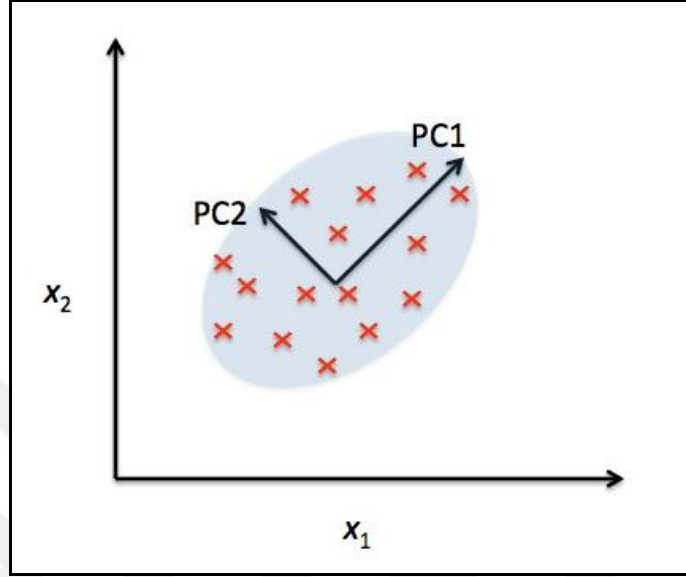
Üzerinde çalışılan tohum grubu 3 farklı buğday çeşidinden oluşmaktadır. Kama, Rosa ve Kanada buğday türlerinin her birinden deneyde kullanılmak üzere rastsal olarak 70’er tane seçilmiştir. Tohumların içyapısının ve yedi farklı değişken ölçümünün elde edilmesinde yüksek kaliteli görünümlü, yumuşak x-ışınları tekniği kullanılarak elde edilmiştir. Bu teknik zararsız ve mikroskobik tarama ya da lazer teknolojisi gibi diğer görüntüleme tekniklerine göre oldukça az maliyetlidir. Elde edilen görüntüler 13x18cm’lik KODAK plakalarına kaydedilmiştir.

Çalışmalar deneysel araştırma alanlarında yetiştirilip hasat edilen buğdaylar ile Lublin’deki Polonya Bilim Akademisi Tarımsal Fizik Enstitüsü’nde gerçekleştirilmiştir. Veriler, sınıflandırma ve küme analizi çalışmalarında kullanılmak üzere hazırlanılmıştır. Ölçümü yapılan değişkenler; tohumun alanı (mm^2), tohumun çevre uzunluğu (mm), tohumun sıklığı (c), tohumun uzunluğu (mm), tohumun genişliği (mm), tohumun simetri katsayısı, tohum kanalının uzunluğu (mm) olmak üzere yedi tanedir.

6.3.2.2. Temel bileşen analizi

Orijinal ismi PCA (Principal Component Analysis) olan temel bileşen analizi, çok değişkenli bir veri setinde yer alan bilgileri daha az değişken ile bilgi kayıpları minimum olacak şekilde açıklayan bir tekniktir (Şekil 6.19.). Verinin orijinal

değişkenlerinin temel bileşenleri olarak adlandırılan yeni değişkenler, veriye ait ilk baştaki bilgilerin tamamını açıklama kabiliyeti olan ve bir nevi orijinal değişkenlerin basitleştirilmiş halleridir. İlk temel bileşen, en büyük varyanslı değişken seçildikten sonra oluşan yeni değerler arasında varyanslar yine büyükten küçüğe doğru sıralanır.

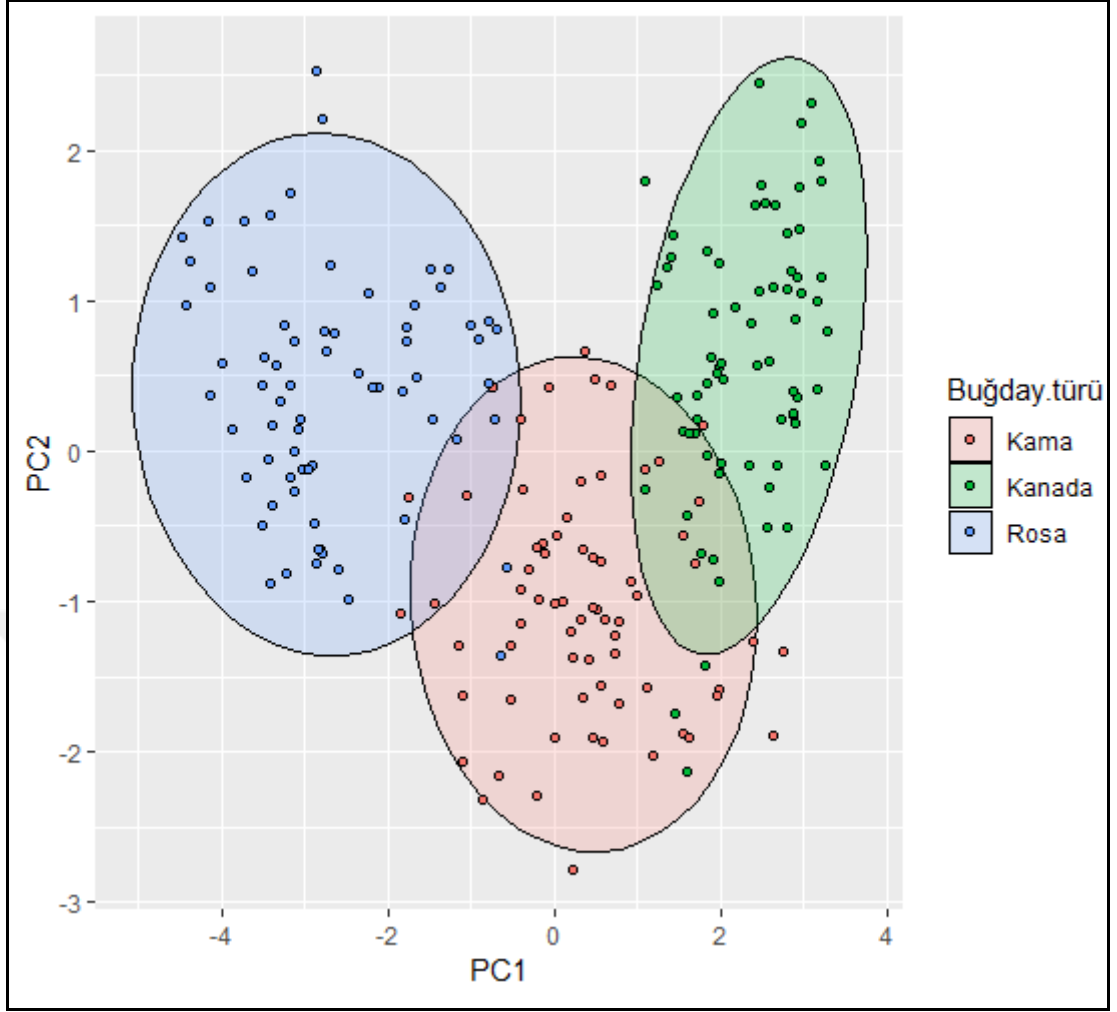


Şekil 6.19. PCA (Principal Component Analysis)

Boyutsallığı azaltmaya yarayan en yaygın tekniklerden biri olan temel bileşen analizinin önemli özellikleri:

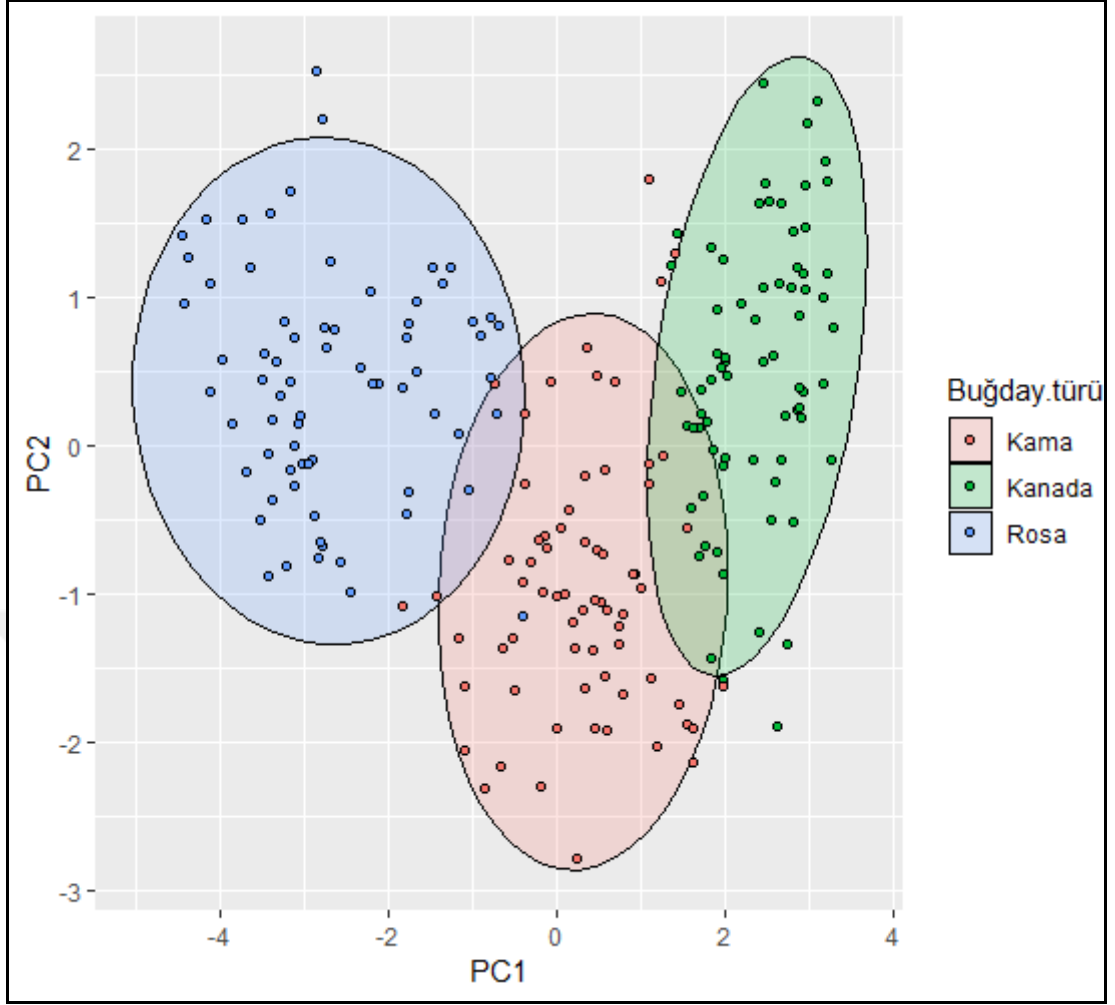
1. Veri desenlerini tanımlamaya yardımcı olur.
2. Karmaşık veri setlerini basitleştirir.
3. Verilerin sonuca olan ağırlıklarını tespit eder.
4. Aykırı değerlere karşı oldukça duyarsızdır.

Uygulamanın bu bölümünde yer verilen seeds gerçek veri setinde, tezde bahsedilen teknikler uygulanarak kümelenme analizi yapılmıştır. Fakat yedi farklı değişkene sahip bu veri setinde veri desenlerini oluşturulabilmesi ve kümeleri iki boyutlu bir düzeyde gözlemleyerek kümeleme algoritmalarının performanslarının karşılaştırılmasının yapılabilmesi için ayrıca temel bileşen analizi uygulanmıştır (Şekil 6.20.). R programlama dilinde uygulanan bu tekniğe ait kodlara Ek A.'da yer verilmiştir.



Şekil 6.21. Seeds veri seti

Veri seti üzerinde AGNES, DIANA, CLARA, PAM, fuzzy c-means, k-ortalamlar, DBSCAN algoritmaları çalıştırılarak kümeler elde edilmiştir. Yoğunluk tabanlı algoritmalarından DBSCAN algoritması iki değişkenden daha fazla değişken içeren veri setlerinde çalışmadığından bu veri setinde sonuç elde edilememiştir. Diğer algoritmaların sonuçları şekiller ve karışıklık matrisleriyle birlikte aşağıda performans sıralarına göre gösterilmiştir.

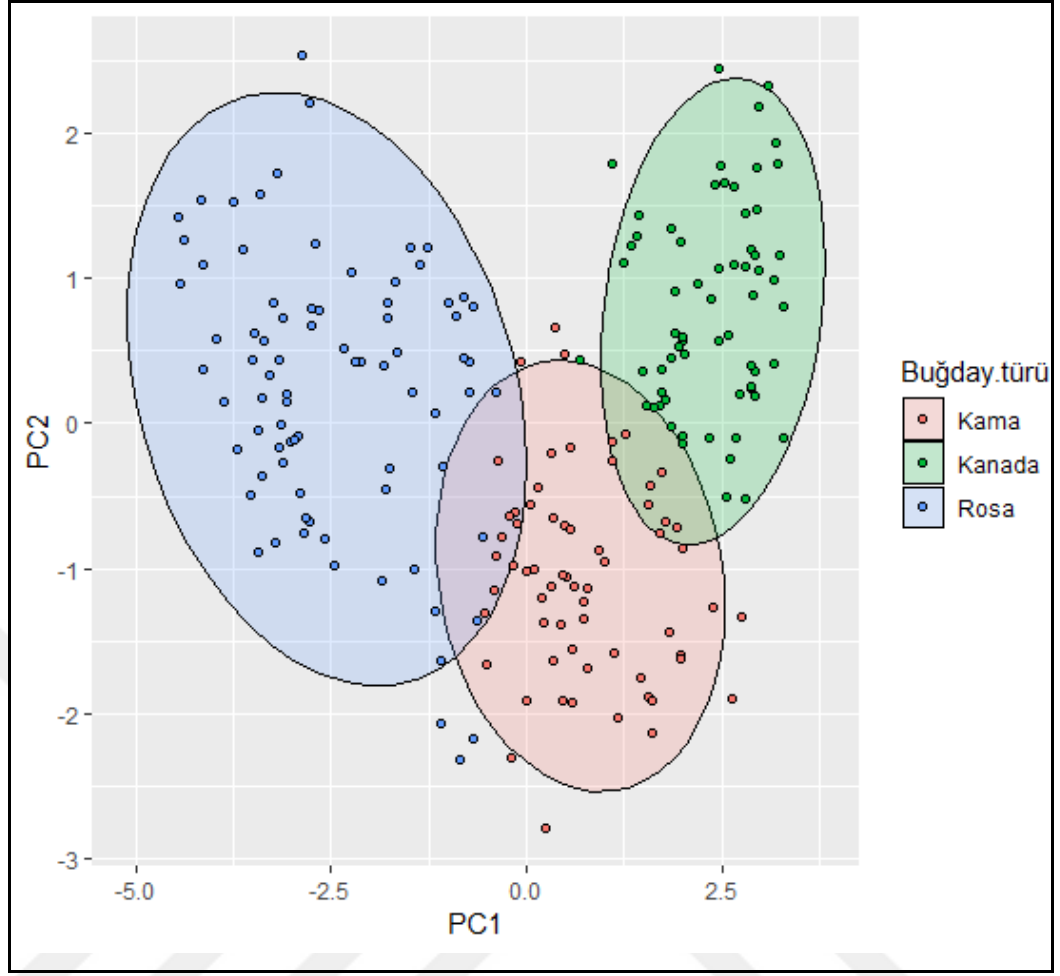


Şekil 6.22. Seeds veri seti için CLARA algoritmasının kümeleme sonucu

Bölünmeli kümeleme algoritmalarından olan ve büyük veri setlerinde etkin çalışan CLARA algoritması $192/210 = 0,9142$ (%91,42) doğruluk oranı ile en iyi sonucu elde eden algoritma olmuştur.

Çizelge 6.49. Seeds veri seti için CLARA algoritmasının karışıklık matrisi

Kümeler	Orijinal Kama türü	Orijinal Rosa türü	Orijinal Kanada türü
Tahminlenen Kama türü	60	2	6
Tahminlenen Rosa türü	3	68	0
Tahminlenen Kanada türü	7	0	64

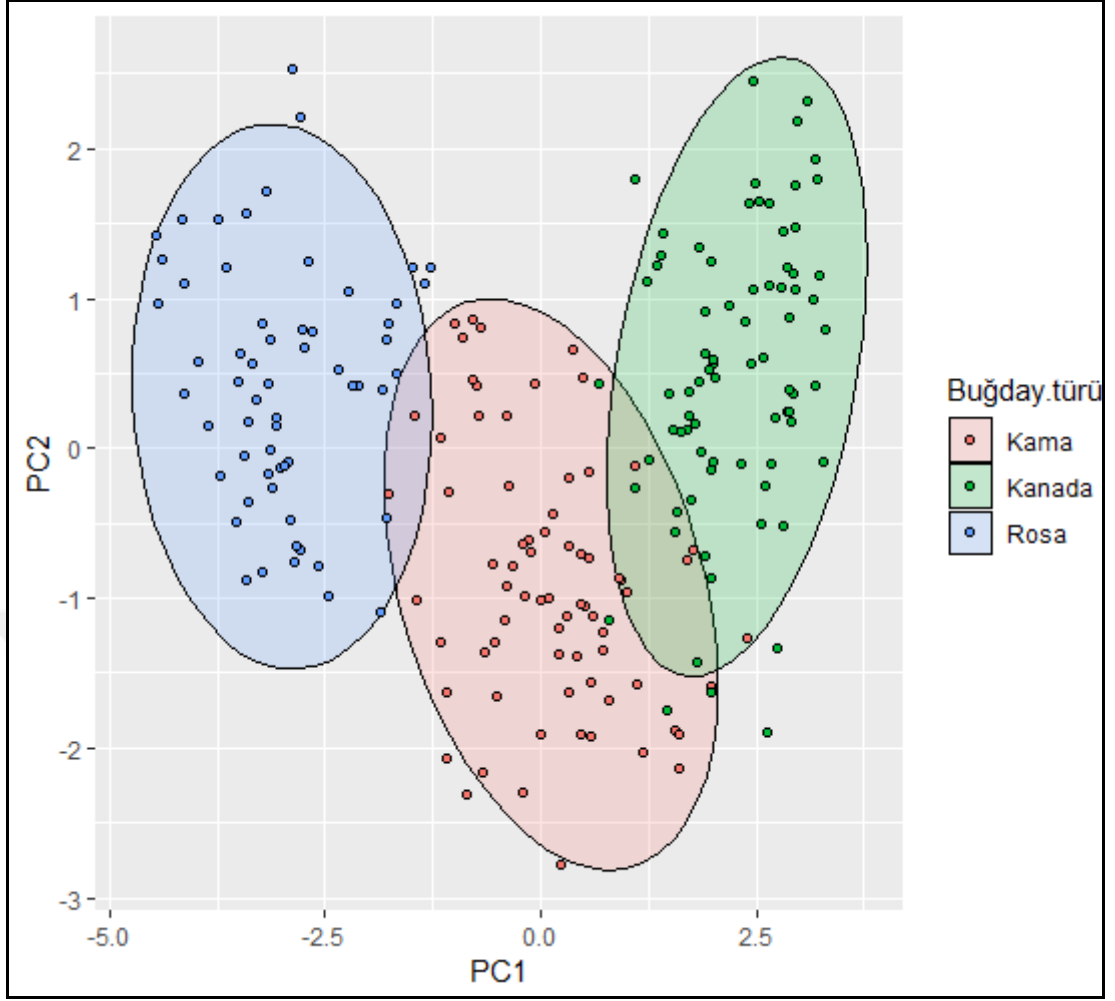


Şekil 6.23. Seeds veri seti için DIANA algoritmasının kümeleme sonucu

Hiyerarşik kümeleme yönteminin ayırıcı mantığıyla çalışan DIANA algoritması $189/210 = 0,90$ (%90,00) doğruluk oranı ile en başarılı ikinci algoritma olmayı başarmıştır.

Çizelge 6.50. Seeds veri seti için DIANA algoritmasının karışıklık matrisi

Kümeler	Orijinal Kama türü	Orijinal Rosa türü	Orijinal Kanada türü
Tahminlenen Kama türü	57	0	8
Tahminlenen Rosa türü	11	70	0
Tahminlenen Kanada türü	2	0	62

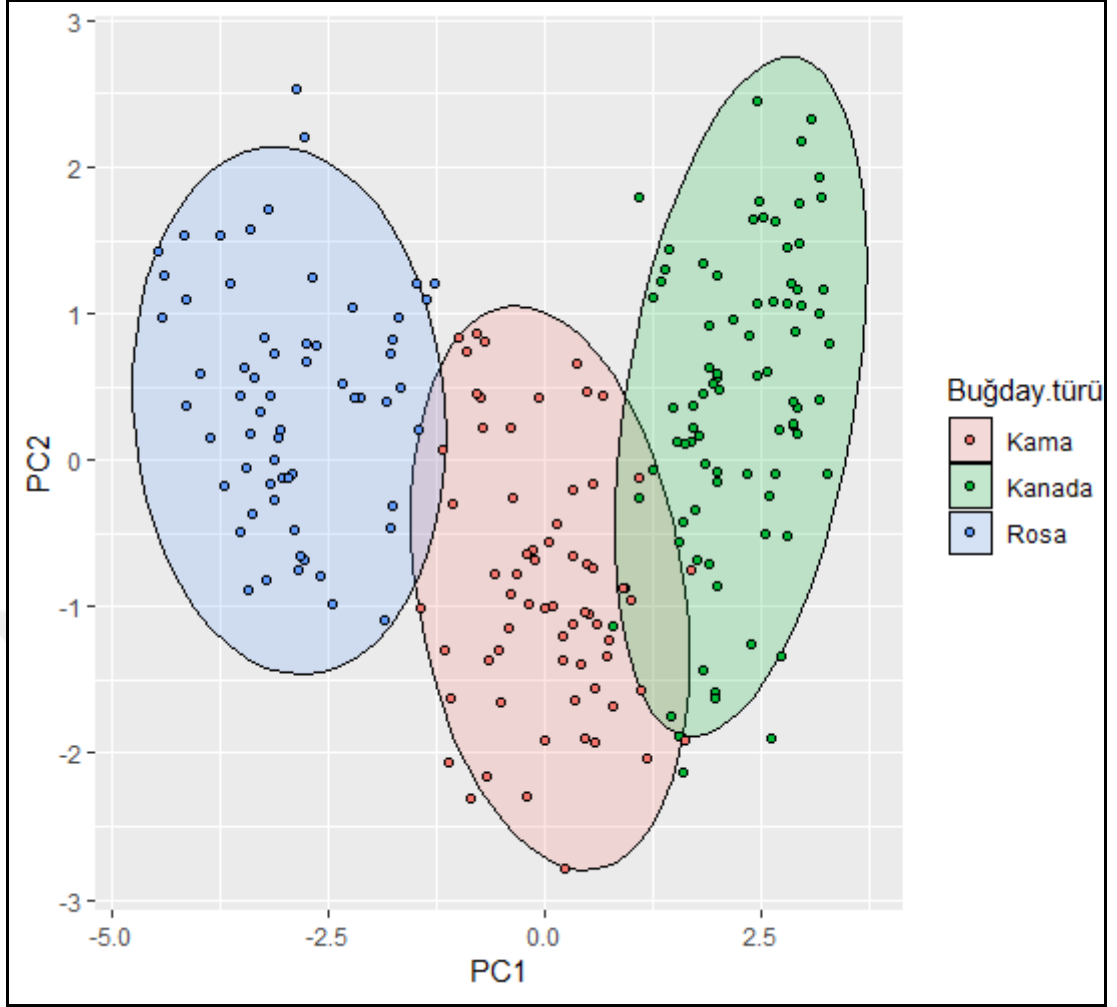


Şekil 6.24. Seeds veri seti için k-ortalamlar algoritmasının kümeleme sonucu

Bölümlü kümeleme yöntemlerinden olan ve literatürde en bilindik kümeleme algoritması olan k-ortalamlar algoritması $188/210 = 0,8952$ (%89,52) doğruluk oranı ile sonuç elde edebilmiştir.

Çizelge 6.51. Seeds veri seti için k-ortalamlar algoritmasının karışıklık matrisi

Kümeler	Orijinal Kama türü	Orijinal Rosa türü	Orijinal Kanada türü
Tahminlenen Kama türü	60	10	2
Tahminlenen Rosa türü	1	60	0
Tahminlenen Kanada türü	9	0	68

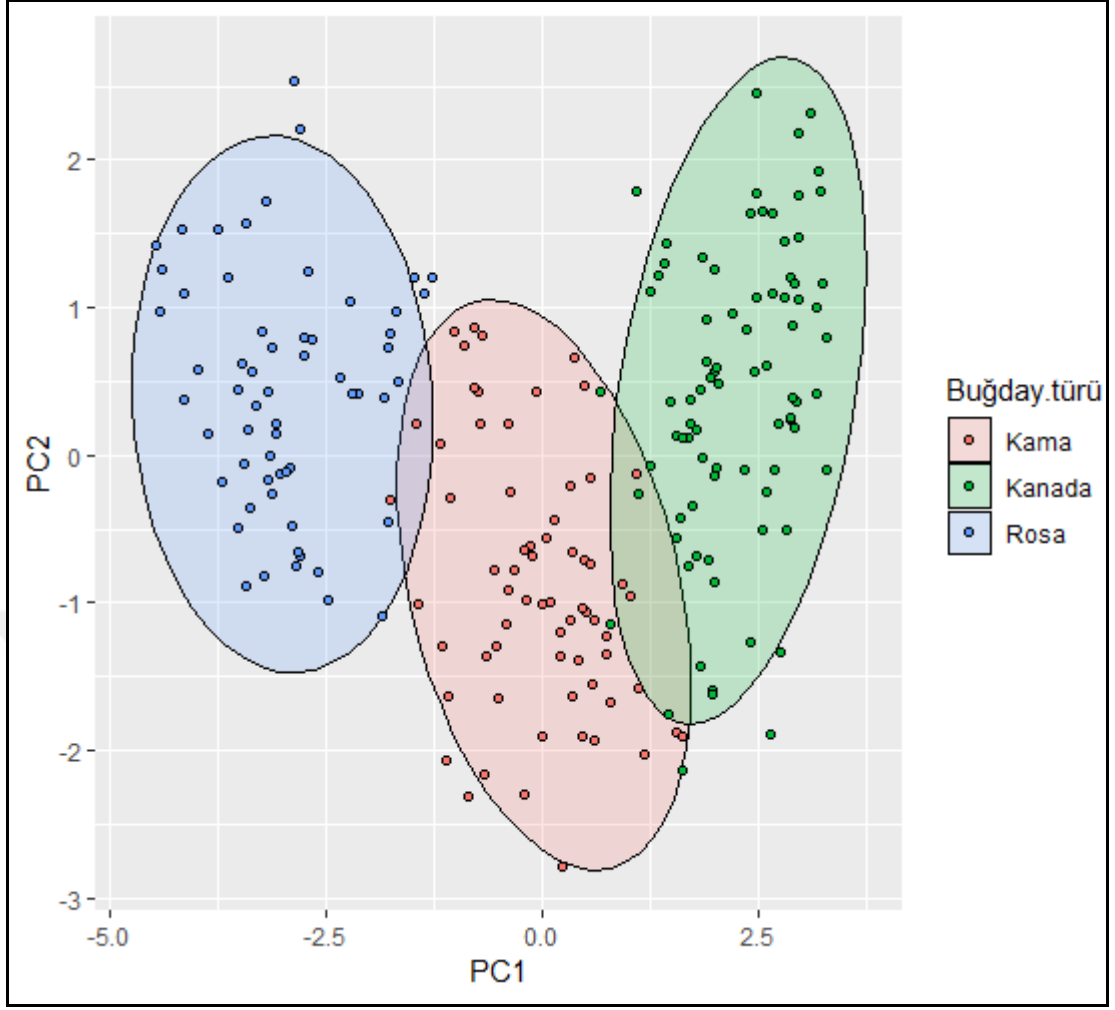


Şekil 6.25 Seeds veri seti için fuzzy c-means algoritmasının kümeleme sonucu

Bulanık kümeleme yöntemi olan fuzzy c-means algoritması $188/210 = 0,8952$ (%89,52) doğruluk oranı ile k-ortalamalar algoritmasıyla aynı oranda başarılı olmuştur. Bu iki algoritmanın doğruluk oranları aynı olsa da küme sonuçları birebir aynı değildir.

Çizelge 6.52. Seeds veri seti için fuzzy c-means algoritmasının karışıklık matrisi

Kümeler	Orijinal Kama türü	Orijinal Rosa türü	Orijinal Kanada türü
Tahminlenen Kama türü	57	9	0
Tahminlenen Rosa türü	2	61	0
Tahminlenen Kanada türü	11	0	70

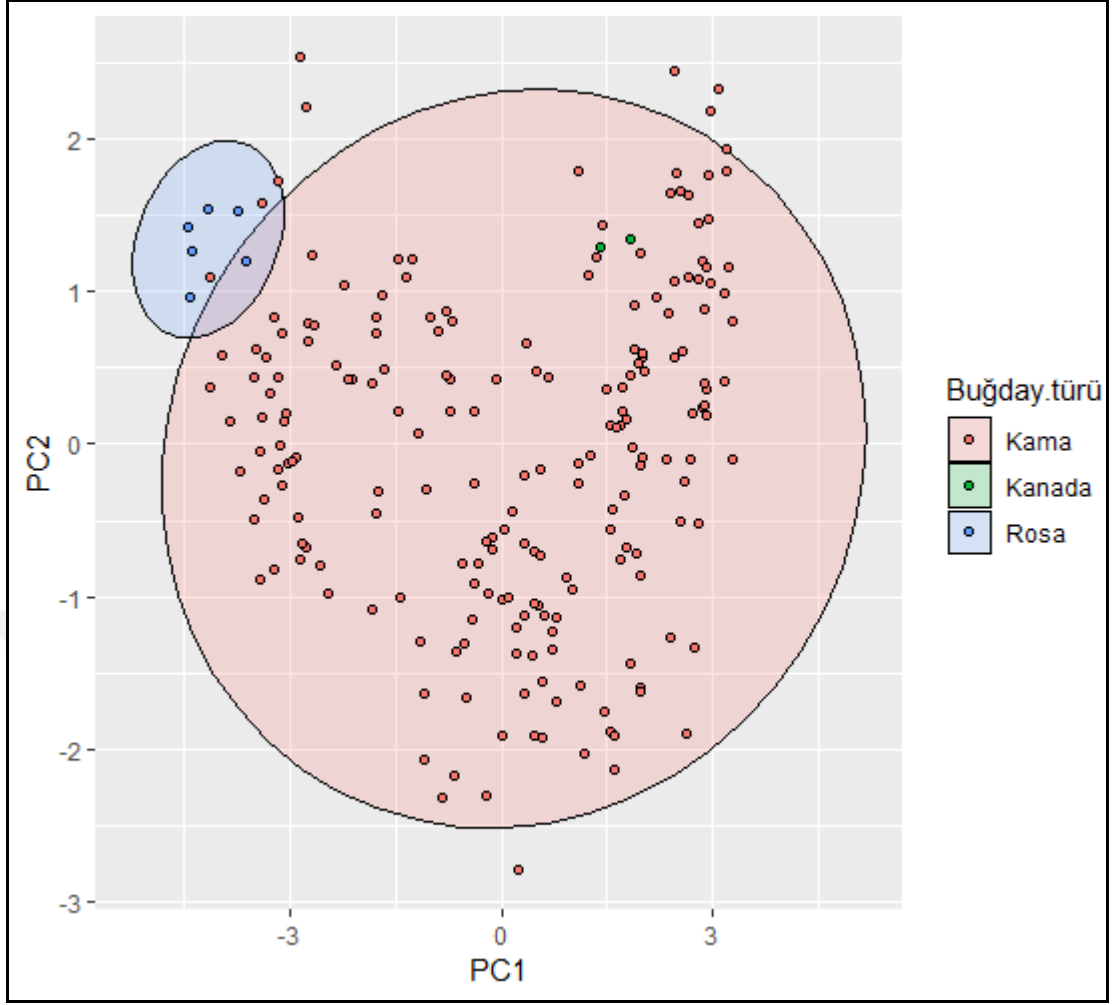


Şekil 6.26. Seeds veri seti için PAM algoritmasının kümeleme sonucu

Bir diğer bölünmeli kümeleme yöntemlerinden biri olan PAM algoritması $187/210 = 0,8904$ (%89,04) doğruluk oranı ile uygulamada yer alan algoritmalar içinde nispeten daha başarısız bir sonuç elde etmiştir.

Çizelge 6.53. Seeds veri seti için PAM algoritmasının karışıklık matrisi

Kümeler	Orijinal Kama türü	Orijinal Rosa türü	Orijinal Kanada türü
Tahminlenen Kama türü	57	10	0
Tahminlenen Rosa türü	1	60	0
Tahminlenen Kanada türü	12	0	70



Şekil 6.27. Seeds veri seti için AGNES algoritmasının kümeleme sonucu

Hiyerarşik algoritmalarından birleştirici mantığıyla çalışan AGNES algoritması büyük veri setlerinde başarılı sonuçlar elde edemediğinden $78/210 = 0,3714$ (%37,14) doğruluk oranı ile en başarısız algoritma olmuştur.

Çizelge 6.54. Seeds veri seti için AGNES algoritmasının karışıklık matrisi

Kümeler	Orijinal Kama türü	Orijinal Rosa türü	Orijinal Kanada türü
Tahminlenen Kama türü	70	64	68
Tahminlenen Rosa türü	0	6	0
Tahminlenen Kanada türü	0	0	2

Seeds gerçek veri seti için, uygulamada kullanılan algoritmaların kümeleme performanslarını gösteren doğru kümelenen nokta sayıları ve doğruluk oranlarını tek bir tabloda incelemek gerekirse;

Çizelge 6.55. Seeds gerçek veri seti için algoritmaların doğruluk oranları

Algoritmalar	Doğru kümelenen nokta sayısı	Doğruluk oranı
CLARA	192	%91,42
DIANA	189	%90
K-ortalamalar	188	%89,52
Fuzzy c-means	188	%89,52
PAM	187	%89,04
AGNES	78	%37,14

6.4. Yöntemlerin Performansları Arasındaki Farkların Anlamlılığı

Uygulamada bölünmeli, hiyerarşik, yoğunluğa dayalı ve bulanık kümeleme yöntemleri uygulanmış ve bu yöntemlerin performanslarını değerlendirebileceğimiz değişken olan doğruluk oranları; normal dağılım göstermedikleri, veri sayısının yeterli olmaması ve birden çok yöntemi aynı anda kıyaslama gibi varsayımlara sahip olduğu için tek yönlü ANOVA'nın non-parametrik karşılığı olan Kruskal Wallis-H testi ile anlamlı farklılıklara sahip olup olmadıkları test edilmiştir.

Test edilecek olan H_0 ve H_a hipotezleri:

H_0 : Tezde kullanılan yöntemlerin doğru kümeleme performansları arasında istatistiksel açıdan fark vardır.

H_a : Tezde kullanılan yöntemlerin doğru kümeleme performansları arasında istatistiksel açıdan fark yoktur.

Yukarıda kurulan hipotezler Kruskal Wallis-H testi ile test edildiğinde elde edilen p değeri 0,04003 olarak hesaplanmıştır. Bu testte 0,05 anlamlılık düzeyinde dikkate alındığında elde edilen p değeri 0,05 değerinden küçük olduğundan H_0 hipotezi kabul edilir ve kullanılan yöntemlerin performansları arasında anlamlı bir fark olduğu söylenebilir. Buna göre tezde kullanılan yöntemlerin bireyleri doğru

kümeleme oranlarının (yani doğru sınıflandırma performansları) istatistiksel açıdan farklı oldukları söylenebilir.



7. SONUÇ

Kavram olarak ilk defa 1939 yılında dile getirilen kümeleme analizi veri madenciliğinde yaygın olarak kullanılan tekniklerden biridir. Kümeleme analizleri, bir araştırmanın incelenen verilerini, aralarında olan benzerliklere göre belli gruplara dahil ederek sınıflandırmaya, nesnelerin benzer özelliklerini ortaya çıkarmaya ve bu özellikler hakkında tanımlar yapabilmeye olanak sağlayan yöntemlerdir. Kümeleme işlemi sonunda ortaya çıkan başarılı kümelenen nesneler arasındaki benzerlik maksimum değerde olurken farklı kümelerdeki nesneler arasındaki farklılık maksimum olur. Kümeleme yöntemleri, verilerin aralarındaki farklılıkları ele alarak kümeleme analizi yapar ve genellikle ortaya küresel sonuçlar çıkarır. Bu yöntemler farklı şekillere sahip kümelerin tespiti durumlarında yetersiz kalmaktadır. Yoğunluk bazlı yöntemler ise veri yoğunluklarına göre kümeleme yapar.

Bu çalışmada, bölümlemeli yöntemlerden k-ortalamar, k-medoids, PAM ve CLARA; hiyerarşik yöntemlerden AGNES, DIANA ve CURE; yoğunluk tabanlı yöntemlerden DBSCAN ve DENCLUE; bulanık kümeleme yöntemlerinden fuzzy c-means; grid tabanlı kümeleme yöntemlerinden STING ve CLIQUE algoritmaları açıklanmıştır. Bu algoritmalarından AGNES, DIANA, CLARA, PAM, Fuzzy c-means, k-ortalamar, DBSCAN algoritmalarının R programlama dili ile uygulamaları yapılmıştır. Elde edilen sonuçlara göre; Aggregation veri setinde DBSCAN, Sentetik 1 veri setinde PAM, Clara, AGNES, Pathbased 1 veri setinde DBSCAN, Jain veri setinde DBSCAN, Pathbased 2 spiral veri setinde DBSCAN ve AGNES, Flame veri setinde DBSCAN, R15 veri setinde PAM ve CLARA, Compound veri setinde DBSCAN, D31 veri setinde PAM ve CLARA algoritmaları en iyi sonuçları vermişlerdir.

Uygulamada kullanılan veri setlerinden elde edilen sonuçlar değerlendirildiğinde, farklı şekil ve büyüklükte kümelerin söz konusu olduğu sentetik veri setlerinde DBSCAN algoritmasının başarılı sonuçlar verdiği görülmüştür. Bununla birlikte, küresel ve benzer büyüklükte kümelerin olduğu veri setlerinde hiyerarşik ve bölümlemeli algoritmalar kümeleri doğru tespit edebilmişlerdir. Bölünmeli küme

algoritmalarının aykırı değerlere karşı duyarlı olduğu ve hiyerarşik küme algoritmalarının bu durumda daha iyi sonuç verdiği görülmüştür. Değişken sayısının yedi olduğu ve bu nedenle DBSCAN algoritmasının kullanılamadığı gerçek veri setinde ise AGNES algoritmasının oldukça başarısız bir sonuç elde etmiş ayrıca CLARA, k-ortalamalar, DIANA, fuzzy c-means, PAM algoritmaları da birbirlerine yakın sonuçlar elde etmiş olsalar da CLARA algoritması aralarında en iyi sonucu veren algoritma olmuştur.

Literatürde yer alan kümeleme algoritmaları karşılaştırmalarında, algoritmaların başarısının uygulanan veri setlerinin özelliklerine göre değişiklik gösterdiği kabul edilmiştir. Aynı zamanda k-ortalamalar algoritmasının hemen hemen her tür veri seti tipinde kabul edilebilen sonuçlar verdiği görülmüştür.

Bu çalışmadan elde edilen sonuçlara göre algoritma seçiminde; üzerinde çalışılan veri setinin büyüklüğü, şekli, dağılımı, yoğunluğu, değişken sayısı, aykırı değerler barındırması gibi birçok özelliğinin dikkate alınması ve kümeleme analizine başlarken öncelikle veri setinin özelliklerinin iyi tanımlanması gerektiği söylenebilir. Ayrıca bir algoritmanın “en iyi” olarak nitelendirilemeyeceği, algoritmaların performanslarının veri setlerinin özelliklerine göre değiştiği söylenebilir.

KAYNAKLAR

- Adriaans, P. ve Zantinge, D. (1997) *Data Mining, Addison Wesley Longman Publishing*, Boston, 176s.
- Akın, K.Y. (2008) *Veri Madenciliğinde Kümeleme Algoritmaları ve Kümeleme Analizi*, Doktora Tezi, Marmara Üniversitesi, İstanbul, 174s.
- Akpınar, H. (2000) Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği, *İ.Ü. İşletme Fakültesi Dergisi*, 29 (1), 1-22.
- Albayrak, A.S. ve Yılmaz, Ş.K. (2009) Veri Madenciliği: Karar Ağaçları Algoritmaları ve İMKB Verileri Üzerine Bir Uygulama, *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 14 (1), 31-52.
- Argüden, Y. ve Erşahin, B. (2008) *Veri Madenciliği Veriden Bilgiye Masraftan Değere*, Alkim Kağıt San. ve Tic. A.Ş., İstanbul, 72 s.
- Baykal, A. (2006) Veri Madenciliği Uygulama Alanları, *D. Ü. Ziya Gökalp Eğitim Fakültesi Dergisi*, 7, 95-107.
- Berkhin, P. (2002) *Survey of Clustering Data Mining Techniques*, Accrue Software Inc, California, 268s.
- Berry, M. ve Linoff, G. (2000) *The Art and Science of Customer Relationship Management*, Wiley Computer Publishing, U.S.A., 512s.
- Bilgin, T. (2003) *Veri Madenciliğinde Kümeleme Analizi Yöntemi Uygulaması*, Yüksek Lisans Tezi, Marmara Üniversitesi, İstanbul, 148s.
- Çelik G. (2013) *Meslek Yüksekokulu Öğrencilerinin Başarı Durumlarını Etkileyen Faktörlerin Veri Madenciliği Kümeleme Teknikleri Kullanılarak Analizi: Ağrı Meslek Yüksekokulu Örneği*, Yüksek Lisans Tezi, Atatürk Üniversitesi, Erzurum, 89s.
- Dinçer, E. (2006) *Veri Madenciliğinde K-Means Algoritması ve Tıp Alanında Uygulanması*, Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli, 101s.

- Doğan, Ş. ve Türkoğlu, İ. (2007) Hypothyroidi and Hyperthyroidi Detection from Thyroid Hormone Parameters by Using Decision Trees, *Doğu Anadolu Bölgesi Araştırmaları Dergisi*, 5: 163-169.
- Güler N. (2006) *Bulanık Kümeleme Analizi ve Bulanık Modellemeye Uygulamaları*, Yüksek Lisans Tezi, Muğla Sıtkı Koçman Üniversitesi, Muğla, 157s.
- Han, J. ve Kamber, M. (2001) *Data Mining Concepts and Techniques*, Morgan Kauffmann Publishers, U.S.A., 702s
- Hand, D.J. (1998) *Data Mining: Statistics and More*, *The American Statistician*, 52: 112-118.
- Hair, J.F., Anderson, R.E. ve Tatham, R. (1998) *Multivariate Data Analysis*, Prentice Hall Inc, New Jersey, 248s.
- Işık, M. (2006) *Bölünmeli Kümeleme Yöntemleri ile Veri Madenciliği Uygulamaları*, Yüksek Lisans Tezi, Marmara Üniversitesi, İstanbul, 76s.
- Jacobs, P. (1999) *Data Mining: What General Managers Need to Know*, Harvard Management Update, 4: 8-9s
- Jain, A.K., Murty M.N. ve Flynn P. J. (1999) *Data Clustering: A Review*, *CSUR*, 31: 264-323.
- Kaufman, L. ve Rousseuw, P. (1990) *Finding Groups Data : An Introduction to Cluster Analysis*, John Wiley and Sons, New York, 342s.
- Karypis, G. ve Kumar, V. (1999) *CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling*, Technical Report, Department of Computer Science and Engineering University of Minnesota, Minnesota, 8: 68-75.
- Michalski, R.S., Stepp, R.E. ve Diday, E. (1981) *A recent advance in data analysis: Clustering objects into classes characterized by conjunctive concepts*. *In Progress in pattern recognition*, North-Holland Publishing Company, 33-56
- Özekes, S. (2003) *Data Mining Models and Application Areas*, *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 21: 65-82.

- Qiang, Z., Zheng Z., Wei, S.Z. ve Daley E. (2005) WINP: A Window-based Incremental and Paralel Clustering Algorithm for Very Large Databases, *17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*, Hong Kong, China, 169-176
- Savaş, S., Topaloğlu, N. ve Yılmaz, M. (2012) Veri Madenciliği ve Türkiye'deki Uygulama Örnekleri, *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 11: 1-23.
- Servi, T. (2009) *Çok Değişkenli Karma Dağılım Modeline Dayalı Kümeleme Analizi*, Doktora Tezi, Çukurova Üniversitesi, Adana, 242s.
- Sharma, S. (1996) *Applied Multivariate Techniques*, John Wiley & Sons, New York, 493s.
- Sia, W. ve Lazarescu, M.M. (2005) Clustering Large Dynamic Datasets Using Exemplar Points, *The IASTED Conference on Artificial Intelligence and Applications*, Australia, 163-173.
- Şahin, M. ve Hamarat, B. (2002) G-10 Avrupa Birliği ve OECD Ülkelerinin Sosyo-Ekonomik Benzerliklerinin Fuzzy Kümeleme Analizi ile Belirlenmesi, VI *International Conference in Economics*, Ankara, 11-14.
- Tatlıdil, H. (2002) *Uygulamalı Çok Değişkenli İstatistiksel Analiz*, Ziraat Matbaacılık, Ankara, 330s.
- Wang, W., Yang, J. ve Muntz, R. (1997) STING: A Statistical Information Grid Approach To Spatial Data Mining, *International Conference of Very Large Databases (VLDB'99)*, Greece, 186-195.
- Zaiane, R.O., Foss, A., Chi-Hoon L. ve Wang W. (2002) On Data Clustering Analysis: Scalability, Constraints and Validation, *Sixth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'02)*, Taiwan, 28-39.

EKLER

Ek A. R Kodları

- Kodlar sadece aggregation veri seti için verilmiştir. Diğer veri setleri için de uygulanabilir.
- Veri setinin bilgisayarda C:\ konumuna kayıtlı olduğu varsayılmıştır.

DBSCAN algoritması

```
library(fpc)
x=read.table(file='C:\\aggregation.txt',dec=',',sep='\t')
x1<-data.frame(x$V1,x$V2)
ds<- dbscan(x1, 1.384,7,showplot=1)
x2<-predict(ds,x1)
x.c<-data.frame(x1,x2)
write.table(x.c,file='k.a.txt',dec=',',sep='\t',row.names=F)
```

K-ortalama algoritması

```
library(cluster)
x=read.table('C:\\aggregation.txt',dec=',',sep=';')
x1=data.frame(x$V1,x$V2)
km=kmeans(x1,6)
show(km)
plot(x1,col=km$cluster)
x.c<-data.frame(x1,km$cluster)
write.table(x.c,file='k.a5.txt',dec=',',sep='\t',row.names=F)
```

Fuzzy c-ortalama algoritması

```
library(e1071)
x=read.table('C:\\aggregation.txt',dec='.',sep='\t')
x1=data.frame(x$V1,x$V2)
cm=cmeans(x1,6,20,verbose=TRUE,m=2,method="cmeans")
print(cm)
show(cm)
plot(x1,col=cm$cluster)
x.c<-data.frame(x1,cm$cluster)
write.table(x.c,file='k.b.txt',dec=',',sep='\t',row.names=F)
```

PAM algoritması

```
library(cluster)
x=read.table(file='c:\\aggregation.txt',dec='.',sep='\t')
x1<-data.frame(x$V1,x$V2)
d.x<- daisy(x1)
pamv<- pam(d.x, 7, diss = TRUE)
clusplot(pamv, shade = TRUE)
x.c<-data.frame(x1,pamv$clustering)
write.table(x.c,file='k.a.txt',dec=',',sep='\t',row.names=F)
```

AGNES algoritması

```
library(cluster)
x=read.table(file='c:\\aggregation.txt',dec='.',sep='\t')
x1<-data.frame(x$V1,x$V2)
agn1 <- agnes(x1, metric="euclidean", method ="single")
k.x<-data.frame(x1,cutree(agn1,k=7))
write.table(k.x,file='k.a.txt',sep='\t',dec=',',row.names=F)
print(agn1)
plot(agn1)
```

DIANA algoritması

```
library(cluster)
x=read.table(file='c:\\aggeration.txt',dec='.',sep='\t')
x1<-data.frame(x$V1,x$V2)
dv <- diana(x1,metric = "manhattan", stand = TRUE)
k.x<-data.frame(x1,cutree(dv,k=7))
write.table(k.x,file='k.a.txt',sep='\t',dec=',',row.names=F)
print(dv)
plot(dv)
```

CLARA algoritması

```
library(cluster)
x=read.table(file='c:\\aggeration.txt',dec='.',sep='\t')
x1<-data.frame(x$V1,x$V2)
clarax <- clara(x1, 7)
clusplot(clarax, shade = TRUE)
x.c<-data.frame(x1,clarax$clustering)
write.table(x.c,file='k.a.txt',dec=',',sep='\t',row.names=F)
```

PCA

```
library("ggplot2")
seeds=read.table(file='c:\\seeds.txt',dec='.',sep='\t')
a<-prcomp(seeds[,-8],scale. = TRUE)
summary(a)
plot(a,type="l")
biplot(a,scale = 0)
str(a)
a$x
b<-cbind(seeds,a$x[,1:2])
head(b)
```

```
ggplot(b,aes(PC1,PC2,col=Buğday.Türü,fill=Buğday.Türü))+  
  stat_ellipse(geom="polygon",col="black",alpha=0.1)+  
  geom_point(shape=21,col="black")
```



Ek B. Gerçek Veri Seti

Tohumun alanı	Tohumun çevre uzunluğu	Tohumun sıklığı	Tohumun uzunluğu	Tohumun genişliği	Tohumun simetri katsayısı	Tohum kanalının Uzunluğu	Buğday Türü
15.26	14.84	0.871	5.763	3.312	2.221	5.22	Kama
14.88	14.57	0.8811	5.554	3.333	1.018	4.956	Kama
14.29	14.09	0.905	5.291	3.337	2.699	4.825	Kama
13.84	13.94	0.8955	5.324	3.379	2.259	4.805	Kama
16.14	14.99	0.9034	5.658	3.562	1.355	5.175	Kama
14.38	14.21	0.8951	5.386	3.312	2.462	4.956	Kama
14.69	14.49	0.8799	5.563	3.259	3.586	5.219	Kama
14.11	14.1	0.8911	5.42	3.302	2.7	5	Kama
16.63	15.46	0.8747	6.053	3.465	2.04	5.877	Kama
16.44	15.25	0.888	5.884	3.505	1.969	5.533	Kama
15.26	14.85	0.8696	5.714	3.242	4.543	5.314	Kama
14.03	14.16	0.8796	5.438	3.201	1.717	5.001	Kama
13.89	14.02	0.888	5.439	3.199	3.986	4.738	Kama
13.78	14.06	0.8758	5.479	3.156	3.136	4.872	Kama
13.74	14.05	0.8744	5.482	3.114	2.932	4.825	Kama
14.59	14.28	0.8993	5.351	3.333	4.185	4.781	Kama
13.99	13.83	0.9183	5.119	3.383	5.234	4.781	Kama
15.69	14.75	0.9058	5.527	3.514	1.599	5.046	Kama
14.7	14.21	0.9153	5.205	3.466	1.767	4.649	Kama
12.72	13.57	0.8686	5.226	3.049	4.102	4.914	Kama
14.16	14.4	0.8584	5.658	3.129	3.072	5.176	Kama
14.11	14.26	0.8722	5.52	3.168	2.688	5.219	Kama
15.88	14.9	0.8988	5.618	3.507	0.7651	5.091	Kama
12.08	13.23	0.8664	5.099	2.936	1.415	4.961	Kama
15.01	14.76	0.8657	5.789	3.245	1.791	5.001	Kama
16.19	15.16	0.8849	5.833	3.421	0.903	5.307	Kama
13.02	13.76	0.8641	5.395	3.026	3.373	4.825	Kama
12.74	13.67	0.8564	5.395	2.956	2.504	4.869	Kama
14.11	14.18	0.882	5.541	3.221	2.754	5.038	Kama
13.45	14.02	0.8604	5.516	3.065	3.531	5.097	Kama
13.16	13.82	0.8662	5.454	2.975	0.8551	5.056	Kama
15.49	14.94	0.8724	5.757	3.371	3.412	5.228	Kama
14.09	14.41	0.8529	5.717	3.186	3.92	5.299	Kama
13.94	14.17	0.8728	5.585	3.15	2.124	5.012	Kama
15.05	14.68	0.8779	5.712	3.328	2.129	5.36	Kama
16.12	15	0.9	5.709	3.485	2.27	5.443	Kama
16.2	15.27	0.8734	5.826	3.464	2.823	5.527	Kama
17.08	15.38	0.9079	5.832	3.683	2.956	5.484	Kama
14.8	14.52	0.8823	5.656	3.288	3.112	5.309	Kama
14.28	14.17	0.8944	5.397	3.298	6.685	5.001	Kama
13.54	13.85	0.8871	5.348	3.156	2.587	5.178	Kama
13.5	13.85	0.8852	5.351	3.158	2.249	5.176	Kama
13.16	13.55	0.9009	5.138	3.201	2.461	4.783	Kama
15.5	14.86	0.882	5.877	3.396	4.711	5.528	Kama
15.11	14.54	0.8986	5.579	3.462	3.128	5.18	Kama
13.8	14.04	0.8794	5.376	3.155	1.56	4.961	Kama

15.36	14.76	0.8861	5.701	3.393	1.367	5.132	Kama
14.99	14.56	0.8883	5.57	3.377	2.958	5.175	Kama
14.79	14.52	0.8819	5.545	3.291	2.704	5.111	Kama
14.86	14.67	0.8676	5.678	3.258	2.129	5.351	Kama
14.43	14.4	0.8751	5.585	3.272	3.975	5.144	Kama
15.78	14.91	0.8923	5.674	3.434	5.593	5.136	Kama
14.49	14.61	0.8538	5.715	3.113	4.116	5.396	Kama
14.33	14.28	0.8831	5.504	3.199	3.328	5.224	Kama
14.52	14.6	0.8557	5.741	3.113	1.481	5.487	Kama
15.03	14.77	0.8658	5.702	3.212	1.933	5.439	Kama
14.46	14.35	0.8818	5.388	3.377	2.802	5.044	Kama
14.92	14.43	0.9006	5.384	3.412	1.142	5.088	Kama
15.38	14.77	0.8857	5.662	3.419	1.999	5.222	Kama
12.11	13.47	0.8392	5.159	3.032	1.502	4.519	Kama
11.42	12.86	0.8683	5.008	2.85	2.7	4.607	Kama
11.23	12.63	0.884	4.902	2.879	2.269	4.703	Kama
12.36	13.19	0.8923	5.076	3.042	3.22	4.605	Kama
13.22	13.84	0.868	5.395	3.07	4.157	5.088	Kama
12.78	13.57	0.8716	5.262	3.026	1.176	4.782	Kama
12.88	13.5	0.8879	5.139	3.119	2.352	4.607	Kama
14.34	14.37	0.8726	5.63	3.19	1.313	5.15	Kama
14.01	14.29	0.8625	5.609	3.158	2.217	5.132	Kama
14.37	14.39	0.8726	5.569	3.153	1.464	5.3	Kama
12.73	13.75	0.8458	5.412	2.882	3.533	5.067	Kama
17.63	15.98	0.8673	6.191	3.561	4.076	6.06	Rosa
16.84	15.67	0.8623	5.998	3.484	4.675	5.877	Rosa
17.26	15.73	0.8763	5.978	3.594	4.539	5.791	Rosa
19.11	16.26	0.9081	6.154	3.93	2.936	6.079	Rosa
16.82	15.51	0.8786	6.017	3.486	4.004	5.841	Rosa
16.77	15.62	0.8638	5.927	3.438	4.92	5.795	Rosa
17.32	15.91	0.8599	6.064	3.403	3.824	5.922	Rosa
20.71	17.23	0.8763	6.579	3.814	4.451	6.451	Rosa
18.94	16.49	0.875	6.445	3.639	5.064	6.362	Rosa
17.12	15.55	0.8892	5.85	3.566	2.858	5.746	Rosa
16.53	15.34	0.8823	5.875	3.467	5.532	5.88	Rosa
18.72	16.19	0.8977	6.006	3.857	5.324	5.879	Rosa
20.2	16.89	0.8894	6.285	3.864	5.173	6.187	Rosa
19.57	16.74	0.8779	6.384	3.772	1.472	6.273	Rosa
19.51	16.71	0.878	6.366	3.801	2.962	6.185	Rosa
18.27	16.09	0.887	6.173	3.651	2.443	6.197	Rosa
18.88	16.26	0.8969	6.084	3.764	1.649	6.109	Rosa
18.98	16.66	0.859	6.549	3.67	3.691	6.498	Rosa
21.18	17.21	0.8989	6.573	4.033	5.78	6.231	Rosa
20.88	17.05	0.9031	6.45	4.032	5.016	6.321	Rosa
20.1	16.99	0.8746	6.581	3.785	1.955	6.449	Rosa
18.76	16.2	0.8984	6.172	3.796	3.12	6.053	Rosa
18.81	16.29	0.8906	6.272	3.693	3.237	6.053	Rosa
18.59	16.05	0.9066	6.037	3.86	6.001	5.877	Rosa
18.36	16.52	0.8452	6.666	3.485	4.933	6.448	Rosa
16.87	15.65	0.8648	6.139	3.463	3.696	5.967	Rosa
19.31	16.59	0.8815	6.341	3.81	3.477	6.238	Rosa
18.98	16.57	0.8687	6.449	3.552	2.144	6.453	Rosa

18.17	16.26	0.8637	6.271	3.512	2.853	6.273	Rosa
18.72	16.34	0.881	6.219	3.684	2.188	6.097	Rosa
16.41	15.25	0.8866	5.718	3.525	4.217	5.618	Rosa
17.99	15.86	0.8992	5.89	3.694	2.068	5.837	Rosa
19.46	16.5	0.8985	6.113	3.892	4.308	6.009	Rosa
19.18	16.63	0.8717	6.369	3.681	3.357	6.229	Rosa
18.95	16.42	0.8829	6.248	3.755	3.368	6.148	Rosa
18.83	16.29	0.8917	6.037	3.786	2.553	5.879	Rosa
18.85	16.17	0.9056	6.152	3.806	2.843	6.2	Rosa
17.63	15.86	0.88	6.033	3.573	3.747	5.929	Rosa
19.94	16.92	0.8752	6.675	3.763	3.252	6.55	Rosa
18.55	16.22	0.8865	6.153	3.674	1.738	5.894	Rosa
18.45	16.12	0.8921	6.107	3.769	2.235	5.794	Rosa
19.38	16.72	0.8716	6.303	3.791	3.678	5.965	Rosa
19.13	16.31	0.9035	6.183	3.902	2.109	5.924	Rosa
19.14	16.61	0.8722	6.259	3.737	6.682	6.053	Rosa
20.97	17.25	0.8859	6.563	3.991	4.677	6.316	Rosa
19.06	16.45	0.8854	6.416	3.719	2.248	6.163	Rosa
18.96	16.2	0.9077	6.051	3.897	4.334	5.75	Rosa
19.15	16.45	0.889	6.245	3.815	3.084	6.185	Rosa
18.89	16.23	0.9008	6.227	3.769	3.639	5.966	Rosa
20.03	16.9	0.8811	6.493	3.857	3.063	6.32	Rosa
20.24	16.91	0.8897	6.315	3.962	5.901	6.188	Rosa
18.14	16.12	0.8772	6.059	3.563	3.619	6.011	Rosa
16.17	15.38	0.8588	5.762	3.387	4.286	5.703	Rosa
18.43	15.97	0.9077	5.98	3.771	2.984	5.905	Rosa
15.99	14.89	0.9064	5.363	3.582	3.336	5.144	Rosa
18.75	16.18	0.8999	6.111	3.869	4.188	5.992	Rosa
18.65	16.41	0.8698	6.285	3.594	4.391	6.102	Rosa
17.98	15.85	0.8993	5.979	3.687	2.257	5.919	Rosa
20.16	17.03	0.8735	6.513	3.773	1.91	6.185	Rosa
17.55	15.66	0.8991	5.791	3.69	5.366	5.661	Rosa
18.3	15.89	0.9108	5.979	3.755	2.837	5.962	Rosa
18.94	16.32	0.8942	6.144	3.825	2.908	5.949	Rosa
15.38	14.9	0.8706	5.884	3.268	4.462	5.795	Rosa
16.16	15.33	0.8644	5.845	3.395	4.266	5.795	Rosa
15.56	14.89	0.8823	5.776	3.408	4.972	5.847	Rosa
15.38	14.66	0.899	5.477	3.465	3.6	5.439	Rosa
17.36	15.76	0.8785	6.145	3.574	3.526	5.971	Rosa
15.57	15.15	0.8527	5.92	3.231	2.64	5.879	Rosa
15.6	15.11	0.858	5.832	3.286	2.725	5.752	Rosa
16.23	15.18	0.885	5.872	3.472	3.769	5.922	Rosa
13.07	13.92	0.848	5.472	2.994	5.304	5.395	Kanada
13.32	13.94	0.8613	5.541	3.073	7.035	5.44	Kanada
13.34	13.95	0.862	5.389	3.074	5.995	5.307	Kanada
12.22	13.32	0.8652	5.224	2.967	5.469	5.221	Kanada
11.82	13.4	0.8274	5.314	2.777	4.471	5.178	Kanada
11.21	13.13	0.8167	5.279	2.687	6.169	5.275	Kanada
11.43	13.13	0.8335	5.176	2.719	2.221	5.132	Kanada
12.49	13.46	0.8658	5.267	2.967	4.421	5.002	Kanada
12.7	13.71	0.8491	5.386	2.911	3.26	5.316	Kanada
10.79	12.93	0.8107	5.317	2.648	5.462	5.194	Kanada

11.83	13.23	0.8496	5.263	2.84	5.195	5.307	Kanada
12.01	13.52	0.8249	5.405	2.776	6.992	5.27	Kanada
12.26	13.6	0.8333	5.408	2.833	4.756	5.36	Kanada
11.18	13.04	0.8266	5.22	2.693	3.332	5.001	Kanada
11.36	13.05	0.8382	5.175	2.755	4.048	5.263	Kanada
11.19	13.05	0.8253	5.25	2.675	5.813	5.219	Kanada
11.34	12.87	0.8596	5.053	2.849	3.347	5.003	Kanada
12.13	13.73	0.8081	5.394	2.745	4.825	5.22	Kanada
11.75	13.52	0.8082	5.444	2.678	4.378	5.31	Kanada
11.49	13.22	0.8263	5.304	2.695	5.388	5.31	Kanada
12.54	13.67	0.8425	5.451	2.879	3.082	5.491	Kanada
12.02	13.33	0.8503	5.35	2.81	4.271	5.308	Kanada
12.05	13.41	0.8416	5.267	2.847	4.988	5.046	Kanada
12.55	13.57	0.8558	5.333	2.968	4.419	5.176	Kanada
11.14	12.79	0.8558	5.011	2.794	6.388	5.049	Kanada
12.1	13.15	0.8793	5.105	2.941	2.201	5.056	Kanada
12.44	13.59	0.8462	5.319	2.897	4.924	5.27	Kanada
12.15	13.45	0.8443	5.417	2.837	3.638	5.338	Kanada
11.35	13.12	0.8291	5.176	2.668	4.337	5.132	Kanada
11.24	13	0.8359	5.09	2.715	3.521	5.088	Kanada
11.02	13	0.8189	5.325	2.701	6.735	5.163	Kanada
11.55	13.1	0.8455	5.167	2.845	6.715	4.956	Kanada
11.27	12.97	0.8419	5.088	2.763	4.309	5	Kanada
11.4	13.08	0.8375	5.136	2.763	5.588	5.089	Kanada
10.83	12.96	0.8099	5.278	2.641	5.182	5.185	Kanada
10.8	12.57	0.859	4.981	2.821	4.773	5.063	Kanada
11.26	13.01	0.8355	5.186	2.71	5.335	5.092	Kanada
10.74	12.73	0.8329	5.145	2.642	4.702	4.963	Kanada
11.48	13.05	0.8473	5.18	2.758	5.876	5.002	Kanada
12.21	13.47	0.8453	5.357	2.893	1.661	5.178	Kanada
11.41	12.95	0.856	5.09	2.775	4.957	4.825	Kanada
12.46	13.41	0.8706	5.236	3.017	4.987	5.147	Kanada
12.19	13.36	0.8579	5.24	2.909	4.857	5.158	Kanada
11.65	13.07	0.8575	5.108	2.85	5.209	5.135	Kanada
12.89	13.77	0.8541	5.495	3.026	6.185	5.316	Kanada
11.56	13.31	0.8198	5.363	2.683	4.062	5.182	Kanada
11.81	13.45	0.8198	5.413	2.716	4.898	5.352	Kanada
10.91	12.8	0.8372	5.088	2.675	4.179	4.956	Kanada
11.23	12.82	0.8594	5.089	2.821	7.524	4.957	Kanada
10.59	12.41	0.8648	4.899	2.787	4.975	4.794	Kanada
10.93	12.8	0.839	5.046	2.717	5.398	5.045	Kanada
11.27	12.86	0.8563	5.091	2.804	3.985	5.001	Kanada
11.87	13.02	0.8795	5.132	2.953	3.597	5.132	Kanada
10.82	12.83	0.8256	5.18	2.63	4.853	5.089	Kanada
12.11	13.27	0.8639	5.236	2.975	4.132	5.012	Kanada
12.8	13.47	0.886	5.16	3.126	4.873	4.914	Kanada
12.79	13.53	0.8786	5.224	3.054	5.483	4.958	Kanada
13.37	13.78	0.8849	5.32	3.128	4.67	5.091	Kanada
12.62	13.67	0.8481	5.41	2.911	3.306	5.231	Kanada
12.76	13.38	0.8964	5.073	3.155	2.828	4.83	Kanada
12.38	13.44	0.8609	5.219	2.989	5.472	5.045	Kanada
12.67	13.32	0.8977	4.984	3.135	2.3	4.745	Kanada

11.18	12.72	0.868	5.009	2.81	4.051	4.828	Kanada
12.7	13.41	0.8874	5.183	3.091	8.456	5	Kanada
12.37	13.47	0.8567	5.204	2.96	3.919	5.001	Kanada
12.19	13.2	0.8783	5.137	2.981	3.631	4.87	Kanada
11.23	12.88	0.8511	5.14	2.795	4.325	5.003	Kanada
13.2	13.66	0.8883	5.236	3.232	8.315	5.056	Kanada
11.84	13.21	0.8521	5.175	2.836	3.598	5.044	Kanada
12.3	13.34	0.8684	5.243	2.974	5.637	5.063	Kanada



ÖZGEÇMİŞ

Kişisel Bilgiler

Ad Soyad : Ali Burak Sayılan
Uyruk : T.C.
Doğum Yeri ve Tarihi : Kars 09/06/1989
Medeni Hali : Bekar
Telefon : 0 555 393 78 62
E-posta : aliburaksayilam@hotmail.com.tr

Eğitim

Alınan Derece	Aldığı Kurum/Üniversite	Mezuniyet Yılı
Lise	Muğla Anadolu Lisesi	2007
Lisans	Dokuz Eylül Üniversitesi	2015
Yüksek Lisans	Muğla Sıtkı Koçman Üniversitesi	2019

İş Tecrübesi

Yıl	Yer	Pozisyon/Görev
2018-	T.C. Maliye Bakanlığı Mali Suçları Araştırma Kurulu Başkanlığı	İstatistikçi

Yabancı Dil

Dil (İngilizce)	Başlangıç	Orta	İleri
Yazma		X	
Konuşma		X	
Anlama		X	
Okuma		X	