
GAZİ ÜNİVERSİTESİ
Gazi
1926

KÜÇÜK ÖRNEKLEM ÇAPLI YÜKSEK BOYUTLU VERİLERDE KLASİK VE SAĞLAM KÜMELEME YÖNTEMLERİNİN PERFORMANSLARININ KARŞILAŞTIRILMASI

Gülşah KILIÇ

YÜKSEK LİSANS TEZİ
İSTATİSTİK ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TEMMUZ 2025

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

.

Gülşah KILIÇ
03/07/2025

KÜÇÜK ÖRNEKLEM ÇAPLI YÜKSEK BOYUTLU VERİLERDE KLASİK VE SAĞLAM KÜMELEME YÖNTEMLERİNİN PERFORMANSLARININ KARŞILAŞTIRILMASI

(Yüksek Lisans Tezi)

Gülşah KILIÇ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Temmuz 2025

ÖZET

Günümüzde, özellikle genom verisi gibi Küçük Örneklem Çaplı Yüksek Boyutlu (KÖÇYB) veri setleri üzerine yapılan çalışmalar önem kazanmıştır. Bu tür veri yapısında, değişken sayısının örnek çapından fazla olması (p>>n), analiz süreçlerinde çeşitli zorluklara yol açmakta; özellikle kümeleme analizlerinde uzaklık hesaplamalarının güvenilirliğini azaltarak kümelerin sağlıklı bir şekilde belirlenmesini güçleştirmektedir. Bu çalışmada, yapısal zorluklara ek olarak aykırı gözlemler ve karışma (kontaminasyon) gibi bozulmaların etkisi altında, klasik ve sağlam kümeleme algoritmalarının performansı değerlendirilmiştir. Kümeleme performansı ölçümü, dışsal doğrulama ölçütü olan Ayarlanmış Rand (AR) indeksi ve içsel doğrulama ölçütleri olan Calinski-Harabasz (CH), Silhouette ve Dunn indeksleri aracılığıyla yapılmıştır. Analizler, hem kanserle ilişkili genomik veri setleri hem de farklı aykırı gözlemler ve karışma oranları içeren simülasyonlar aracılığıyla, R programlama dili kullanılarak gerçekleştirilmiştir. Simülasyon çalışması sonucunda, sağlam kümeleme yöntemlerinden kırpılmış k-ortalamalar ve k-medyan algortimalarının KÖÇYB veri yapılarında klasik algoritmalardan daha başarılı olduğu gözlemlenmiştir. Kümeleme algoritmalarının başarısı yanlızca yöntemsel yeterliliğe değil, aynı zamanda veri yapısının özelliklerine de bağlı olması nedeniyle, başarı ölçütlerinin yorumlanmasında veri setinin yüksek boyutluluğu, örnek çapı, içerdiği aykırı gözlemler ve karışma durumu gibi faktörler dikkate alınarak değerlendirilmiştir.

Bilim Kodu	:	20513
Anahtar Kelimeler	:	Küçük örneklem çaplı yüksek boyutlu (KÖÇYB) veri setleri, *k*-ortalamalar, hiyerarşik kümeleme, *k*-medoid, *k*-medyan, kırpılmış *k*-ortalamalar, aykırı gözlemler, karışma, ayarlanmış Rand indeksi, Dunn indeksi, Silhouette indeksi, Calinski-Harabasz indeksi
Sayfa Adedi	:	84
Danışman	:	Prof. Dr. Necla GÜNDÜZ TEKİN

PERFORMANCE COMPARİSONS OF CLASSİCAL AND ROBUST CLUSTERİNG METHODS FOR HİGH DİMENSİONAL AND LOW SAMPLE SİZE DATA

(M. Sc. Thesis)

Gülşah KILIÇ

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

July 2025

ABSTRACT

Nowadays, studies focusing on High-Dimensional Low Sample Size (HDLSS) datasets, particularly genomic data, have gained significant importance. In such data structures, the number of variables exceeds the number of observations (p >> n), which creates various challenges in the analysis process. Especially in clustering analysis, this situation reduces the reliability of distance calculations, making it more difficult to accurately identify cluster structures. In this study, the performance of classical and robust clustering algorithms was evaluated under both structural challenges and the presence of data imperfections such as outliers and contamination. Clustering performance was assessed using the Adjusted Rand Index (AR) as an external validation measure, along with Calinski-Harabasz (CH), Silhouette, and Dunn indices as internal validation metrics. The analyses were performed in the R programming language using both cancer-related genomic datasets and simulations incorporating varying levels of outliers and contamination. The simulation results revealed that robust clustering algorithms, particularly trimmed k-means and k-median, tend to outperform classical methods when applied to HDLSS data structures. Since the success of clustering algorithms depends not only on methodological adequacy but also on the inherent characteristics of the dataset, the evaluation of clustering performance considered critical factors such as high dimensionality, sample size, the presence of outliers, and contamination levels.

Science Code	:	20513
Key Words	:	High-dimensional low sample size (HDLSS) datasets, *k*-means, hierarchical clustering, *k*-medoids, *k*-median, trimmed *k*-means, outliers, contamination, Adjusted Rand index, Dunn index, Silhouette index, Calinski-Harabasz index
Page Number	:	84
Supervisor	:	Prof. Dr. Necla GÜNDÜZ TEKİN

TEŞEKKÜR

Tez çalışmam süresince yalnızca akademik bir danışmandan çok daha fazlası olarak yanımda olan, bilgi ve deneyimiyle gelişimime büyük katkı sağlayan değerli danışmanım Prof. Dr. Necla GÜNDÜZ'e en içten teşekkürlerimi sunarım. Kendisinin bilgiye olan tutkusu, öğrenciye yönelik özverili yaklaşımı ve ilham verici duruşu hem akademik hem de kişisel yolculuğumda bana rehberlik etmiştir. Onun rehberliğinde çalışmak, bu sürecin en değerli kazanımlarından biri olmuştur.

Bu süreçte her daim yanımda olan ve sevgileriyle bana güç veren aileme; kıymetli babam Günaydın KILIÇ'a, sevgili annem Güllüzar KILIÇ'a ve canım kardeşim Başak Bengü KILIÇ'a teşekkür ederim. Özellikle çocukluğumdan beri bana hayal kurmayı öğreten, "daha iyisi mümkün" diyerek hep ileriyi hedeflememi sağlayan anneme minnettarım. Küçük kardeşim olmasına rağmen kimi zaman ablalık yapan Başak Bengü'nün sabrı ve desteği bana her zaman güç vermiştir. Emek vermeyi, çok çalışmayı ve yılmadan ilerlemeyi bize aşılayan anne ve babama en derin şükranlarımı sunarım.

Bu yolculuk boyunca yanımda olan, destekleriyle moral kaynağım olan arkadaşlarıma gönülden teşekkür ederim.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ ix

ŞEKİLLERİN LİSTESİ x

SİMGELER VE KISALTMALAR xi

1. GİRİŞ 1

2. LİTERATÜR TARAMASI 5

3. KÜMELEME ALGORİTMALARI 9

3.1. Klasik Kümeleme Algoritmaları 11

3.1.1. *k*-ortalamalar (*k*-means) algoritması 11

3.1.2. Hiyerarşik kümeleme algoritması 13

3.2. Sağlam (Robust) Kümeleme Algoritmaları 16

3.2.1. *k*-medoids kümeleme algoritması 18

3.2.2. *k*-medyan (*k*-median) kümeleme algoritması 19

3.2.3. Kırpılmış *k*-ortalamalar (trimmed *k*-means) kümeleme algoritması 21

3.3. Temel Bileşenler Analizi ile Küme Yapısını Görselleştirme 23

3.4. Kümelemede Performans Değerlendirme İndeksleri 24

3.4.1. Dışsal doğrulama indekslerinin performans karşılaştırılması 25

3.4.2. İçsel doğrulama indekslerinin performans karşılaştırılması 28

4. KÜÇÜK ÖRNEKLEM ÇAPLI YÜKSEK BOYUTLU (KÖÇYB) VERİ SETLERİNDE KARŞILAŞILAN ZORLUKLAR 33

4.1. Küçük Örneklem Çaplı Yüksek Boyutlu (KÖÇYB) Verilerde Sağlamlık İhtiyacı ve Aykırılık Etkisi 34

Sayfa

5. GENOMİK VERİ SETLERİ İLE UYGULAMA 35

5.1. Veri Setlerine Genel Bakış 35

5.2. Kümeleme Algoritmalarında R Paketleri ve Fonksiyonların Kullanımı 38

5.3. Veri Setlerine Kümeleme Yöntemlerinin Uygulanması ve Doğrulama İndeksleri 40

6. SİMÜLASYON ÇALIŞMASI 45

7. SONUÇ VE ÖNERİLER 57

KAYNAKLAR 59

EKLER 65

ÖZGEÇMİŞ 84

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 3.1. *k*-ortalamalar kümeleme algoritması 13

Çizelge 3.2. Hiyerarşik birleştirici kümeleme algoritması 15

Çizelge 3.3. Uzaklık ölçütlerinin açıklamaları ve formülleri 16

Çizelge 3.4. *k*-medoids algoritması..... 18

Çizelge 3.5. *k*-medyan kümeleme algoritması..... 20

Çizelge 3.6. Kırpılmış *k*-ortalamalar kümeleme algoritması..... 23

Çizelge 3.7. Kümeleme ve gerçek sınıf etiketlerine göre gözlem çiftlerinin doğruluk sınıflandırması (TP, TN, FP, FN)..... 26

Çizelge 5.1. Uygulamada kullanılan veri setlerinin özellikleri..... 37

Çizelge 5.2. Kümeleme algoritmalarının içsel ve dışsal doğrulama indekslerine göre sonuçları 41

Çizelge 6.1. *k*-ortalamalar kümeleme algortiması için performans değerlendirme indeksleri (AR, CH, Silhouette ve Dunn indeksleri) sonuçları..... 47

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 3.1. *k*-medyan algoritması için iterasyon sayısına karşılık ceza skorları (162.760, 106.886, 73.207) 21

Şekil 6.1. Simülasyon senaryolarında *k*-ortalamalar yöntemine ilişkin AR skorlarının karışma ve n/p oranlarına göre görselleştirilmesi 48

Şekil 6.2. Simülasyon senaryolarında *k*-ortalamalar yöntemine ilişkin CH skorlarının karışma ve n/p oranlarına göre görselleştirilmesi 49

Şekil 6.3. Simülasyon senaryolarında *k*-ortalamalar yöntemine ilişkin Silhouette skorlarının karışma ve n/p oranlarına göre görselleştirilmesi..... 50

Şekil 6.4. Simülasyon senaryolarında *k*-ortalamalar yöntemine ilişkin Dunn indekslerinin karışma ve n/p oranlarına göre görselleştirilmesi 51

Şekil 6.5. *k*-ortalamalar kümeleme yönteminin tba düzleminde görselleştirilmesi ve gerçek etiketlerle karşılaştırılması ($n = 20,\ p = 100$ için) 52

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklamalar
(n/p)	Örnek çapının değişken sayısına oranı
$\mathbf{C}^{(t)}$	t'nci iterasyondaki merkez vektörleri kümesi
$\mathbf{n}_{\text{aykırı}}$	Aykırı gözlem sayısı
$S_k^{(t)}$	t'nci iterasyonda k'ncı kümeye atanan gözlem kümesi
c_k	k'ncı kümenin merkezi
$p_{\text{in}}(x)$	Normal (bozulmamış) verinin yoğunluk fonksiyonu
p_{out}	Karışma verisinin yoğunluk fonksiyonu
$u_{ik}^{(t)}$	i'nci gözlemin k'ncı kümeye aitliğini gösteren gösterge
Δ_I	Küme içindeki en büyük uzaklık
δ_{ij}	Ceza terimi
$\|\cdot\|$	Vektör normu
A	Küme birleştirme sırası listesi
I	Aktif kümeleri gösteren vektör
k	Küme sayısı
$\mathbf{n} \to \infty$	Örnek çapının sonsuza gitmesi durumu
n	Örnek çapı (gözlem sayısı)
p	Değişken sayısı (boyut)
X	Gözlem matrisi
$\mathbf{SIM}(i, m, j)$	Yeni küme ile diğer kümeler arasındaki uzaklığı güncelleyen fonksiyon
U	Üyelik matrisi
$\mathbf{p} \gg \mathbf{n}$	Değişken sayısının, örnek çapından büyük olması
t	İterasyon sayısı
$d(x_i, \mathbf{c}_k)$	Gözlem ile küme merkezi arasındaki uzaklık
$a(x_i)$	Gözlemin kendi kümesindeki ortalama uzaklığı
$b(x_i)$	Gözlemin en yakın diğer kümeye ortalama uzaklığı

Simgeler	Açıklamalar
$\boldsymbol{n > p}$	Örnek çapının değişken sayısından büyük olması
$\boldsymbol{p(x \mid \epsilon)}$	Karışma oranı bilinirken gözlemin koşullu yoğunluğu
$\boldsymbol{s(x_i)}$	Silhouette skoru/indeksi
$\boldsymbol{x_{ij}}$	i'nci gözlem için j'nci özelliğin değeri
$\boldsymbol{\alpha}$	Kırpma Oranı
$\boldsymbol{\varepsilon}$	Karışma oranı (epsilon)
$\boldsymbol{\delta(\mathrm{U}_I, \mathrm{U}_J)}$	İki küme arasındaki en küçük uzaklık

Kısaltmalar	Açıklamalar
ALL	Akut Lenfoblast Lösemi
AML	Akut Miyeloid Lösemi
AR	Ayarlanmış Rand İndeksi
CH	Calinski-Harabasz
CH	Calinski-Harabasz indeksi
CLASSIT	İstatistiksel Kavramsal Kümeleme
CLIQUE	Otomatik Alt Alan Seçimli Izgara
COBWEB	Kavramsal Kümeleme Algoritması
DBCLASD	Yoğunluk Tabanlı Kümeleme ve Veri Keşfi
DBSCAN	Gürültülü Verilerle Yoğunluk Tabanlı Kümeleme
DENCLUE	Yoğunluk Fonksiyonlarına Dayalı Kümeleme
EM	Beklenti-Maksimizasyon Algoritması
FCM	Bulanık *c*-ortalamalar
FN	Yanlış Negatif
FP	Yanlış Pozitif
GMM	Gauss Karışım Modelleri
KDKT	Küme Dışı Kareler Toplamı
KİKT	Küme İçi Kareler Toplamı
KÖÇYB	Küçük Örneklem Çaplı Yüksek Boyutlu
MFOM	Mutlak Farkların Ortalama Mesafesi
NMI	Normalize Edilmiş Karşılıklı Bilgi

Kısaltmalar	Açıklamalar
OPTICS	Sıralı Yoğunluk Tabanlı Kümeleme
OPTI-GRID	Optimal Izgara Tabanlı Kümeleme
Rİ	Rand İndeksi
RP	Radikal prostatektomi
SOMs	Öz-Düzenlemeli Haritalar
STING	İstatistiksel Bilgi Izgara Tabanlı Kümeleme
TBA	Temel Bileşen Analizi
TN	Doğru Negatif
TP	Doğru Pozitif
UVK	Uzaklık Vektör Kümeleme
VOK	Varyans Oran Kriteri

1. GİRİŞ

İstatistiksel yöntemlerin büyük bir kısmı, örneklem çapının (n) artmasıyla daha güvenilir ve anlamlı sonuçlar sunacak şekilde tasarlanmıştır. Bu tür yöntemler, örneklem çapı ($n \to \infty$) arttıkça, tahminlerin tutarlılık ve normal dağılıma yakınsama gibi özellikler sergiler. Bu bağlamda Merkezi Limit Teoremi en temel örneklerden biridir ve örneklem çapı arttıkça ortalamanın normal dağılıma yaklaşacağını ifade eder (Casella ve Berger, 2002; Shen, Shen, Zhu ve Marron, 2016).

Klasik istatistiksel analizlerde, özellikle değişken sayısının (ya da bir diğer deyişle boyut sayısının) (p) örnek çapından, yani örnek çapından (gözlem sayısından) az olduğu durumlarda ($n > p$), aşağıdaki teorik özelliklerin belirli koşullar altında sağlanması beklenir:

- Örneklem büyüklüğü arttıkça tahmin edicilerin tutarlılığı (consistency)
- Merkezi Limit Teoremi'nin uygulanabilirliği
- Kovaryans matrisinin tam ranklı (full-rank) ve tersinin alınabiliyor olması
- Tahmin edicilerin asymptotik normal dağılıma yakınsaması
- Parametre tahminlerinde yansızlık/sapmasızlık ve verimlilik gibi özelliklerin sağlanması

2000'li yılların başından itibaren, genomik, mikrodizi (microarray), tıp, görüntü işleme ve sosyal ağ analizi gibi birçok alanda, gözlem sayısının sabit ya da sınırlı kalmasına karşın değişken sayısının çok yüksek olduğu veri yapıları ortaya çıkmıştır. Mahalanobis'in yüksek boyutlu veri analizine yönelik fikirleri, Rao (1973) tarafından istatistiksel çerçevede genişletilmiş ve boyut artışının analiz sonuçları üzerindeki etkileri detaylı biçimde ortaya konmuştur. İzleyen dönemlerde, özellikle boyut (değişken) artışının istatistiksel tahmin üzerindeki etkisi Portnoy (1984, 1988) tarafından incelenmiştir; boyut sayısının örnek çapına göre yavaşça artması durumunda bile, klasik tahmin edicilerin güvenilirliğinin nasıl etkilendiği ve önemli sapmalar meydana gelebileceği gösterilmiştir.

İstatistiksel analizler için boyut (değişken) sayısı p'nin çok büyük, buna karşılık örneklem capı n'nin sabit veya küçük olduğu durumları kapsayan Küçük Örneklem Çaplı Yüksek Boyutlu (KÖÇYB) (High Dimensional Low Sample Size-HDLSS) veri yapısı ortaya çıkmıştır ve bu tür verilerinin analizinde yaşanan yüksek boyutlu veriyle çalışmanın

zorlukları ile ilgili olarak terminoloji ilk olarak Hall, Marron ve Neeman (2005) tanımlanmıştır. Bu durum klasik yöntemlerin varsayımlarını ihlal etmekte ve analiz sürecini zorlaştırmaktadır (Fan ve Lv, 2008). Çok sayıda değişken içeren (yüksek boyutlu) bu tür veri setleri için, klasik varsayımların geçerliliğinin sorgulaması kaçınılmaz olmuştur.

KÖÇYB veri yapılarında, yani $p \gg n$ durumlarında, klasik istatistiksel yöntemler genellikle yetersiz kalmaktadır. Bu başarısızlığın temel nedenlerinden biri, tahminlerde kullanılan varyans-kovaryans matrisinin en fazla $(n-1)$ ranklı olmasından kaynaklanır. Bir başka ifade ile varyans-kovaryans matrisi tekil (singular) hale gelir ve tersi alınamaz, dolayısıyla $p \gg n$ durumlarında bazı yöntemlerin uygulanması matematiksel olarak mümkün olmaz. Ayrıca, bu tür yapılarda Merkezi Limit Teoremi gibi temel asimptotik sonuçların geçerliliği zayıflar ve bu da klasik sınıflandırma ve kümeleme algoritmalarının KÖÇYB ortamında düşük performans göstermesine yol açar (Hall, Marron ve Neeman, 2005). Sonuç olarak, parametre tahminleri kararsız hale gelir ve modeller sıklıkla aşırı uyum (overfitting) eğilimi gösterir.

Bu çalışmada, KÖÇYB sahip veri yapılarında ortaya çıkan analiz zorlukları dikkate alınarak, klasik kümeleme algoritmaları (k-ortalamalar ve hiyerarşik aglomeratif yöntemler) ile sağlam kümeleme yaklaşımları (k-medoids, k-medyan, kırpılmış k-ortalamalar) karşılaştırmalı olarak gerçek veri setleri ve geniş kapsamlı bir simülasyon çalışması ile incelenmiştir. Kümeleme algortimalarının değerlendirilmesinde performans değerlendirme indeksleri olarak dışsal doğrulama indeksi olarak Ayarlanmış Rand indeksi **(**Adjusted Rand Index (AR)), indeksi, içsel doğrulama indeksleri olarak tanımlanan Calinski-Harabasz (CH), Silhouette ve Dunn indeksleri kullanılmıştır.

Tez çalışmasının ikinci bölümde KÖÇYB veri yapılarında kullanılan kümeleme algoritmalarına ilişkin literatür taramasına yer verilmiştir.

Üçüncü bölümde klasik ve sağlam kümeleme algoritmalarının yapıları açıklanmış, algoritmaların kümeleme performanslarını değerlendirmek amacıyla kullanılan performans indeksleri tanıtılmıştır.

Dördüncü bölümde, KÖÇYB veri yapısının temel özellikleri tanıtılmakta ve bu yapıların neden özel olarak ele alınması gerektiği açıklanmaktadır. Ayrıca, KÖÇYB veri setlerinde

karşılaşılan matematiksel ve istatistiksel zorluklar, küresellik varsayımının bozulması, çok boyutluluk, aykırı gözlemler ve başka dağılımdan gözlem karışması (karışma durumu/kontaminasyon) gibi sorunlara değinilmektedir.

Beşinci bölümde, ikisi $n > p$ yapısında, altı tanesi $p \gg n$ yapısında, yani KÖÇYB veri yapısında toplam sekiz veri seti üzerinde gerçekleştirilen klasik ve sağlam kümeleme algoritmalarının uygulamaları ve performans değerlendirme indeksleri bakımından sonuçları yer almaktadır. Böylelikle, her bir algoritmanın $n > p$ ve $p \gg n$ yapılarına karşı ne derece etkili olduğu belirlenmiş; kümeleme yapılarının ayrışabilirliği, sınıf etiketleriyle uyumu ve yapısal bütünlüğü karşılaştırılmıştır.

Altıncı bölümde, her bir kümeleme algoritmasının temel varsayımlarına ve çalışma prensiplerine uygun biçimde yapılandırılmış bir simülasyon tasarımı geliştirilmiştir. Simülasyon çalışmasında, örnek çapının değişken sayısından çok daha fazla olduğu ($n \gg p$), yani n/p oranının 1'in üzerine çıkması durumunu, örnek çapının değişken sayısından çok küçük olduğu ($p \gg n$), yani n/p oranının 1'in altında olması durumlarını ve aykırı gözlemler ile başka dağılımlardan gözlem karışıması durumlarını içeren senaryolar altında gerçekleştirilmiştir. Veriler başlangıçta normal dağılımdan üretilmiş, ardından belirlenen oranlarda (%20) aykırı gözlemler eklenmiş ve kontaminasyon oranları (0, 0.05, 0.10, …, 0.45) uygulanmıştır. Bu koşullar altında, kümeleme algoritmalarının performansları değerlendirilmiştir. KÖÇYB veri yapılarında elde edilen kümeleme sonuçlarının yorumlanabilirliğini artırmak amacıyla, kümeler Temel Bileşen Analizi (TBA) kullanılarak iki boyutlu uzaya indirgenmiş ve görselleştirilmiştir. Bu yaklaşım, kümeler arası ilişkileri ve iç yapıları daha açık biçimde ortaya koymuştur.

Yedinci ve son bölümde sonuçlar özetlenmektedir.

2. LİTERATÜR TARAMASI

k-ortalamalar algoritması, denetimsiz öğrenme alanında yaygın olarak kullanılan kümeleme yöntemlerinden biridir. Literatüre ilk olarak 20. yüzyıl ortalarında geliştirilmiş ve kümeleme analizinin temel taşlarından biri haline gelmiştir. Bu algoritmanın temelinde, gözlemlerin birbirine uzaklıklarına göre belirli sayıda (k) küme etrafında toplanması ve her bir gözlemin merkezine (centroid) en yakın kümeye atanması yatmaktadır.

Algoritmanın gelişimi iki temel çalışmaya dayanmaktadır. İlk olarak Bell Telephone Laboratories'de iç iletişim sistemlerine yönelik araştırmalar sırasında ortaya atılmış, ancak bu çalışma 1982 yılına kadar yayımlanmadığı için literatüre geç yansımıştır (Lloyd, 1982). Buna karşılık, MacQueen (1967) algoritmayı ilk kez akademik literatüre kazandırarak iteratif yapısını ve küme merkezlerinin güncellenme sürecini ayrıntılı biçimde açıklamıştır. MacQueen'in bu katkısı, *k*-ortalamalar algoritmasının istatistiksel analiz, makine öğrenmesi ve veri madenciliği gibi pek çok alanda yaygınlaşmasının önünü açmıştır.

Hiyerarşik kümeleme algoritmaları, çok değişkenli istatistik ve biyolojik sınıflandırma (taksonomi) alanlarında 20. yüzyılın ortalarında literatüre girmiştir. Bu alandaki en erken sistematik çalışmalardan biri, Sokal ve Michener (1958) tarafından gerçekleştirilmiştir. Araştırmacılar, "numerical taxonomy" (sayısal taksonomi) yaklaşımı kapsamında, uzaklık matrislerine dayalı olarak hiyerarşik aglomeratif kümeleme yöntemlerini geliştirmiştir. Bu yöntemler, özellikle biyolojide türlerin sınıflandırılmasında kullanılmıştır. Daha sonra Johnson (1967), hiyerarşik kümeleme algoritmalarını istatistiksel bağlamda ele almış ve farklı bağlantı (linkage) yöntemlerini tanımlamıştır. Tek bağlantı, tam bağlantı, ortalama bağlantı (single linkage, complete linkage ve average linkage) gibi bağlantı stratejileri, Johnson'un çalışmasıyla sistematik biçimde karşılaştırılmıştır. Bu gelişmeler, hiyerarşik kümeleme algoritmalarının modern istatistik ve makine öğrenmesi alanlarında yaygınlaşmasının önünü açmıştır.

Yoğunluk tabanlı kümeleme algoritmaları arasında en çok bilinen ve kullanılan yöntemlerden biri olan DBSCAN (Gürültülü Verilerle Yoğunluk Tabanlı Kümeleme) algoritması, ilk kez Ester, Kriegel, Sander ve Xu (1996) tarafından literatüre kazandırmıştır. Söz konusu algoritma, büyük hacimli uzamsal veri tabanlarında yoğunluk temelli küme yapılarının keşfi ve aykırı gözlemlerin belirlenmesi amacıyla geliştirilmiştir.

Son yıllarda, bu yönteme benzer yoğunluk veya uzaklık tabanlı algoritmaların yanı sıra, yapay zekâ destekli kümeleme teknikleri de geliştirilmiş ve farklı alanlara özgü çözümler sunulmuştur (Karim ve diğerleri 2021).

Kümeleme algoritmalarının genel prensiplerini, sınıflandırmalarını, güçlü ve zayıf yönlerini inceleyen Xu ve Wunsch (2005) çalışması, alandaki temel başvuru kaynaklarından biri olarak öne çıkmaktadır. Bu çalışmada, farklı veri yapılarında klasik ve sağlam kümeleme algoritmaların çeşitli başarı düzeyleri sergilediği, kümeleme değerlendirmesinin ise genellikle öznel ölçütlere dayandığını vurgulamıştır. Özellikle yüksek boyutlu veri setlerinde, uzaklık ölçütlerinin seçimi ve kümeleme kalitesini değerlendirme kriterlerinin belirlenmesi büyük önem taşımaktadır (Friedman ve Rubin, 1967).

Klasik kümeleme algoritmaları, genellikle düşük boyutlu ve büyük örneklem çaplı ($n >> p$) veri setleri için geliştirilmiştir. Ancak KÖÇYB veri yapılarında karşılaşılan yapısal zorluklar sebebi ile bu yöntemler, aykırı gözlemlere ve başka dağılımdan gözlem karışmasına (karışma durumuna/kontaminasyona) karşı oldukça hassas olup ciddi performans kayıpları yaşayabilmektedir. Özellikle aykırı gözlemler ve kontaminasyon, algoritmaların güvenilir kümeler oluşturmasını zorlaştırmaktadır (Tibshirani, 2002; Vanden Branden ve Hubert, 2005). Hall, Marron ve Neeman (2005), Öklidyen uzaklığının/mesafenin yüksek boyutlu uzaylarda yetersiz kaldığını göstererek, farklı uzaklık ölçütlerinin *k*-ortalamalar algoritması üzerindeki etkilerine dikkat çekmiştir. Sarkar ve Ghosh (2020) klasik uzaklık ölçütlerinin zayıflıklarını aşmak üzere Mutlak Farkların Ortalama Mesafesi (MFOM) adında yeni bir metrik önermiştir. Benzer şekilde, Terada (2013) tarafından geliştirilen Uzaklık Vektör Kümeleme (UVK) yöntemi, gözlem-temelli değil, uzaklık vektörleri üzerinden değerlendirme yaparak KÖÇYB yapılarında daha başarılı sonuçlar elde etmiştir.

Yüksek boyutlu yapılarda yalnızca uzaklık ölçütleri değil, aynı zamanda algoritmaların dayanıklılığı da önem arz etmektedir. Gündüz ve Fokoué (2015), klasik sınıflandırma yöntemlerinin KÖÇYB veri setlerinde gösterdiği yetersizlikleri hem teorik hem de simülasyonlar yoluyla kapsamlı şekilde incelemiş; aykırı gözlemlere ve gürültüye karşı dayanıklı sağlam sınıflandırma yaklaşımlarına odaklanmıştır. Çalışmalarında, sağlam sınıflandırma yöntemlerini örtük (örneğin, düzenlileştirme temelli) ve açık (örneğin, kırma ya da ağırlıklandırma) stratejiler şeklinde sınıflandırarak, bu yaklaşımların farklı veri

koşullarındaki avantaj ve sınırlılıklarını metodolojik bir bütünlük içinde ele almışlardır. Bu katkılar, KÖÇYB veri yapılarında uygulanabilir sağlam yöntemlerin seçiminde yol gösterici niteliktedir.

Bu literatür ışığında, klasik ve sağlam sınıflandırma algoritmalarının KÖÇYB veri yapılarındaki performanslarının detaylı şekilde karşılaştırıldığı çalışmalar, kümeleme algoritmaları için de benzer kapsamlı analizlere duyulan ihtiyacı ortaya koymaktadır.

Sağlam kümeleme yöntemleri üzerine yapılan araştırmalar, özellikle kontaminasyon içeren veri setlerinde kırpılmış *k*-ortalamalar gibi yöntemlerin, klasik algoritmalara kıyasla daha tutarlı sonuçlar verdiğini göstermektedir (García-Escudero ve diğerleri, 2010). Bu yöntemler, gözlemlerin belirli bir oranının sistematik biçimde dışlanması esasına dayanmaktadır. Cuesta-Albertos ve diğerleri (1997) tarafından önerilen kırpılmış *k*-ortalamalar algoritması, bu alandaki öncü sağlam yaklaşımlardan biri olup özellikle aykırı gözlemlerin etkisini azaltmada etkili olmuştur. Ancak sabit bir kırpma (trimming) oranına dayanması, her veri yapısına uyum sağlamasını zorlaştırmakta ve yöntemin esnekliğini sınırlamaktadır.

KÖÇYB veri yapılarında boyut indirgeme tekniklerinin kümeleme algoritmaları üzerindeki etkileri de literatürde dikkat çekmiştir. Peters (2023), boyut indirgeme işlemi uygulanmış ve uygulanmamış veri setlerinde klasik ve KÖÇYB 'e özgü kümeleme algoritmalarını hem içsel hem de dışsal doğrulama indeksleriyle değerlendirmiştir. Bulgular, boyut indirgeme uygulandığında daha başarılı kümeleme sonuçları elde edildiğini göstermiştir. Ancak bu avantajlara rağmen, boyut indirgeme yöntemlerinin bazı teorik kısıtları olduğu da ifade edilmektedir. Weeraratne ve diğerleri (2024), $n < p$ koşulunda temel bileşen analizi (TBA) gibi yöntemlerin modelin aşırı uyum göstermesine yol açtığını ve kovaryans yapısındaki zayıf bağlantılar dikkate alınmadan tüm bileşenlere eşit önem vermesinin yüksek boyutlu ortamlarda performans kaybına neden olduğunu belirtmişlerdir.

Aşırı uyum, bir modelin öğrenme verisini "ezberlemesi", ancak yeni ve daha önce görülmemiş verilere genelleme yapamaması durumu olarak tanımlanır (Hastie, Tibshirani ve Friedman, 2009). Özellikle $p >> n$ koşullundaki kovaryans matrisleri, TBA'nın istikrarını azaltmakta ve güvenilir bileşenler elde edilmesini zorlaştırmaktadır.

Bellas ve diğerleri (2012) ise doğrudan beklenti-maksimizasyon algoritması (Expectation-Maximization (EM)) tabanlı yöntemleri kullanmamakla birlikte, sağlam kümeleme yaklaşımlarında kırpma (trimming) tabanlı stratejilerin, aykırı gözlemleri sistematik olarak dışlayarak özellikle yüksek boyutlu verilerde performans kaybını azaltabileceğini belirtmişlerdir. Bu stratejiler, artan boyutlarla birlikte ortaya çıkan ve modelin başarısını olumsuz etkileyen gereksiz değişkenler ve ölçüm hataları gibi zorluklara karşı duyarlılığı azaltmak için önemli bir alternatif sunmaktadır.

3. KÜMELEME ALGORİTMALARI

Kümeleme algoritmaları, benzer özelliklere sahip gözlemleri aynı kümede toplayarak, veri yapısındaki gizli ve anlamlı örüntüleri ortaya çıkarmayı amaçlayan denetimsiz öğrenme yaklaşımıdır. Özellikle sınıf etiketlerinin bulunmadığı durumlarda, veri yapısını keşfetmek için yaygın olarak kullanılmaktadır. İdeal bir kümeleme yapısında, küme içindeki nesneler birbirine benzer (homojen), kümeler arası nesneler ise birbirinden farklı (heterojen) olmalıdır (Alpar, 2021).

Eşitlik 3.1'de X, p boyutlu ve n örneklem çapından oluşan gözlem matrisidir. Genellikle p boyutunda bir veri matrisi olarak ifade edilir. Matrisin elemanı x_{ij}, i'nci gözlem için j'nci özelliğin değerini göstermektedir. Yani satırlar gözlemleri, sütunlar ise özellikleri/değişkenleri temsil etmektedir.

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{p2} & \cdots & x_{np} \end{bmatrix} \tag{3.1}$$

Literatürde çok sayıda kümeleme algoritması bulunmaktadır. Bu yöntemlerin birden fazla örtüşen özelliğe sahip olması, algoritmaların kesin ve genel geçer bir biçimde sınıflandırılmasını güçleştirmektedir. Ancak Garima (2021), literatürün daha anlaşılır hâle gelmesi açısından sistematik bir sınıflandırmanın önemine dikkat çekmiş ve kümeleme yöntemlerin daha anlaşılır sunulabilmesi adına algoritmaları beş ana grupta sınıflandırmıştır:

1. Bölümleme Kümeleme Yöntemleri: *k*-ortalamalar, *k*-medoids, *k*-medyan, Kırpılmış (trimmed) *k*-ortalamalar, FCM (Bulanık *c*-ortalamalar- Fuzzy c −means)
2. Hiyerarşik Kümeleme Yöntemleri: Birleştirici (Aglomerative), Ayrıştırıcı (Divisive)
3. Yoğunluk Tabanlı Kümeleme Yöntemleri: DBSCAN, OPTICS, DBCLANSD, DENCLUE
4. Izgara Tabanlı Kümeleme Yöntemleri: STING, CLIQUE, OPTI-GRID
5. Model Tabanlı Kümeleme Yöntemleri: COBWEB, CLASSIT, SOMs, GMM

Bu tez çalışmanın kapsamı ise yalnızca bölümleme (*k*-ortalamalar, *k*-medoids, *k*-medyan, Kırpılmış (trimmed) *k*-ortalamalar) ve hiyerarşik kümeleme (Birleştirici (Aglomerative)) algoritmalarıyla sınırlandırılmıştır.

Bölümleme temelli kümeleme algoritmaları, verileri hiyerarşik bir yapı oluşturmaksızın, önceden belirlenen sayıda *(k)* kümeye ayırmayı ve her gözlemi yalnızca bir kümeye atamayı hedefler. Küme sayısı olan *k*, çoğunlukla analizden önce belirlenir. Bu değerin belirlenmesinde elbow yöntemi, siluet analizi ve gap istatistiği gibi istatistiksel yaklaşımlar kullanılabilir. Ayrıca, uygulama alanına özgü bilgi ve deneyim de bu süreçte yol gösterici olabilir (Tibshirani, Walther ve Hastie, 2002). Bu tez çalışmasında, küme sayısı önceden belirlenmiş olup, küme sayısını belirlemeye yönelik herhangi bir yöntem uygulanmamıştır.

Hiyerarşik kümeleme algoritmaları, verileri birleştirici veya ayrıştırıcı yaklaşımlar kullanarak iç içe geçmiş kümeler hâlinde organize eder. Birleştirici (aglomeratif) yöntemlerde her gözlem, başlangıçta ayrı bir küme olarak kabul edilir ve benzerlik/uzaklık ölçütlerine göre kademeli olarak diğer kümelerle birleştirilir. Ayrıştırıcı (divisive) yöntemlerde ise süreç, tüm verilerin tek bir küme olarak değerlendirilmesiyle başlar ve veriler adım adım alt kümelere ayrılır. Her iki yaklaşım da kümeler arasındaki yapısal ilişkileri ortaya koyar ve bu yapı, dendrogram adı verilen ağaç benzeri bir grafikle görselleştirilir (Kaufman ve Rousseeuw, 1990).

Bu kümeleme algoritmaları, düşük boyutlu veriler için geliştirilmiş olup ($n \gg p$), KÖÇYB veri yapılarında sınırlı etkililik gösterebilmektedir. Özellikle boyut sayısının artması, hesaplama karmaşıklığını artırarak bu yöntemlerin performansını düşürebilir. Örneğin *k*-ortalamalar algoritması her kümenin merkezini p boyutlu örneklem ortalama vektörüne göre belirleyip gözlemleri bu merkezlere olan Öklidyen uzaklıklara göre kümelendirir. Ancak bu yaklaşım, aykırı gözlemlere karşı oldukça duyarlıdır. Aykırı gözlemler, küme merkezlerini kaydırarak hatalı kümelenmelere neden olabilir. Tan, Steinbach ve Kumar (2006), yapmış oldukları çalışmada, aykırı gözlem etkisinin, özellikle KÖÇYB veri yapılarında daha belirgin olarak ortaya çıktığını ve bu tür veri setlerinde boyut (değişken) sayısının örnek çapına göre fazla olması, uzaklık hesaplarını kararsız hale getirdiğini ve aykırı gözlemlerin etkisi daha baskın olduğunu söylemektedirler. Bu doğrultuda, klasik yöntemlerin zaaflarını aşmak amacıyla daha sağlam/dayanıklı yapıda kümeleme algoritmaları kullanılabilir.

Böylece, klasik algoritmaların aksine, veri setindeki bozulmalardan daha az etkilenerek istikrarlı sonuçlar elde edilebilir.

Literatürde, genellikle yalnızca klasik ya da yalnızca sağlam kümeleme yöntemlerine odaklanılmış ve sınırlı sayıda algoritma karşılaştırılmıştır (García-Escudero ve diğerleri, 2008; Maronna ve Zamar, 2002; Brodinová ve diğerleri, 2019). KÖÇYB veri yapıları özelinde hem klasik hem de sağlam algoritmaların sistematik biçimde değerlendirildiği kapsamlı çalışmalar oldukça sınırlıdır (Müller ve diğerleri, 2008; Bouveyron ve Brunet-Saumard, 2014).

Bu amaçla bu tez çalışmasında, ele alınan bölümleme ve hiyerarşik kümeleme yöntemleri, analiz teknikleri açısından klasik (*k*-ortalamalar, hiyerarşik birleştirici (aglomeratif) kümeleme) ve sağlam (*k*-medoids, *k*-medyan, kırpılmış (trimmed) *k*-ortalamalar) kümeleme yöntemleri olmak üzere iki grupta değerlendirilmektedir.

3.1. Klasik Kümeleme Algoritmaları

Bu çalışmada, *k*-ortalamalar ve hiyerarşik aglomeratif kümeleme algoritmaları, klasik yöntemler olarak değerlendirilmiştir. *k*-ortalamalar algoritması, özellikle büyük veri setlerinde düşük hesaplama maliyeti ve hızlı çözümler sunma kapasitesi nedeniyle öne çıkmaktadır. Öte yandan, hiyerarşik kümeleme yöntemi ise ön bilgi gerektirmemesi, dendrogram gibi görsel araçlarla sonuçların kolay yorumlanabilmesi ve esnek yapısı ile dikkat çekmektedir. Her iki yöntemin de uygulama kolaylığı, hesaplama verimliliği, yorumlanabilirlik düzeyi ve literatürde yaygın olarak kullanılıyor olmaları, bu çalışma kapsamında klasik kümeleme yöntemleri olarak tercih edilmesinde etkili olmuştur.

3.1.1. *k*-ortalamalar (*k*-means) algoritması

k-ortalamalar kümeleme algoritması, en yaygın kullanılan denetimsiz öğrenme algoritmalarından birisi olmasına rağmen, en büyük sınırlılığı, küme sayısının önceden belirlenmesi gerekliliğidir. *k*-ortalamalar kümeleme algoritmasında veri kümesi, önceden belirlenmiş *k* adet kümeye ayrılır. Küme sayısı, kullanıcı tarafından belirlenebilir ya da bazı sezgisel yaklaşımlar uygulanarak seçilebilir. Algoritma, küme merkezlerini rastgele başlatarak her bir veri noktasını $x = \{x_1,\ x_2, \dots, x_n\}$ merkezler kümesi $c = \{c_1,\ c_2, \dots, c_n\}$

içerisindeki en yakın merkeze atar. Bu işlem genellikle Öklid uzaklığı kullanılarak yapılır (Ali, ve diğerleri, 2022). Veri noktaları arasındaki uzaklığı ölçmek için başka yöntemler olsa da en yaygın kullanılan uzaklık ölçme yöntemi Öklid uzaklığıdır. İki nokta arasındaki arasındaki Öklid uzaklığı Eşitlik (3.2)'deki gibi tanımlanır:

$$\text{Öklid uzaklığı} = d(x, c) = \sqrt{\sum_{i=1}^{n} (x_i - c_i)^2} \tag{3.2}$$

Tüm veri noktaları kümelere atandıktan sonra, her küme için yeni merkez değerleri hesaplanır ve ardından veri noktaları bu güncellenmiş merkezlere göre yeniden atanır. Bu işlem, merkez değerlerinde bir değişiklik kalmayıncaya kadar tekrarlanır.

Algoritmanın amacı, tüm veri noktalarının kendi küme merkezlerine olan uzaklıklarının karelerinin toplamını minimize etmektir. Bu toplam, Küme İçi Kareler Toplamı (KİKT) olarak adlandırılır ve her bir küme için, o kümeye ait tüm veri noktalarının kendi küme merkezlerine olan kareli uzaklıklarının toplanmasıyla hesaplanır. Ardından, tüm kümeler için elde edilen bu değerler birleştirilerek toplam hata değeri elde edilir. Matematiksel olarak *k* adet küme için eşitlik (3.3)'teki gibi ifade edilir (Ali, ve diğerleri, 2022):

$$K\text{İ}KT = \sum_{i=1}^{k} \sum_{x \in G_i} d(x, c_i), \tag{3.3}$$

Burada c_i, i'nci kümenin merkezini, G_i ise merkez noktası c_i olan kümeye ait veri noktaları kümesini ifade eder. Kare alma işlemi, yalnızca negatif değerleri ortadan kaldırmakla kalmaz, aynı zamanda büyük sapmaları daha fazla cezalandırarak kümelerin merkez etrafında yoğunlaşmasını sağlar. Bu nedenle, KİKT değeri ne kadar küçükse, kümeler o kadar homojen ve net ayrılmış kabul edilir. *k*-ortalamalar algoritması da bu değeri minimize etmeye çalışır. Bu yaklaşım, kümelerin küresel şekilli ve sıkı gruplar olarak oluşmasını hedefler. Böylece her küme küresel şekilli gruplar olarak kabul edilir. Algoritmanın genel adımları Çizelge 3.1'de verilmiştir.

Çizelge 3.1. *k*-ortalamalar kümeleme algoritması

1. Başlangıçta
- Veri setinde n adet gözlem ve kullanıcı tarafından belirlenmiş k adet küme sayısı olsun.
- k Adet küme merkezi rastgele seçilir: $c_1^{(1)},\ c_2^{(1)},\ ..\ , c_j^{(1)},\ ..\ , c_k^{(1)}$
- Bu merkezler, ilk iterasyon $(k = 1)$ için başlangıç noktalarıdır.

2. Gözlemlerin Kümelere Atanması
- Her bir gözlem x, Öklid uzaklığına göre en yakın merkeze atanır:

$$x \in G_j^{(k)} \text{ eğer } \left\|x - c_j^{(k)}\right\|^2 < \left\|x - c_i^{(k)}\right\|^2 \text{ tüm } i \neq j$$

Burada $G_j^{(k)}$, k. iterasyondaki j. kümedir.

3. Yeni Merkezlerin Güncellenmesi
- Her küme için yeni merkez, o kümeye ait tüm gözlemlerin ortalaması alınarak hesaplanır:

$$c_j^{(k+1)} = \frac{1}{N_j} \sum_{x \in G_j} x, \qquad j = 1,2, \dots, k$$

Burada N_j, $G_j^{(k)}$ kümesindeki veri noktasını/gözlem sayısı ifade eder.

4. Yakınsama Kontrolü
5. Eğer tüm merkezler değişmemişse, yani

$$c_j^{(k+1)} = c_j^{(k)} \text{ Tüm } j = 1,2, \dots, k$$

Koşulu sağlanıyorsa, algoritma sonlandırılır. Aksi takdirde, 1'nci adıma geri dönülerek işlem tekrarlanır.

Not: $\|\cdot\|$ norm sembolü, vektör normunu (uzaklığını) temsil eder. *k*-ortalamalar algortiması için norm olarak Öklid uzaklığı (L2 normu: Kare farkların toplamı) kullanılmıştır.

3.1.2. Hiyerarşik kümeleme algoritması

Hiyerarşik kümeleme algoritmaları, veriyi iç içe geçmiş kümeler hâlinde organize ederek, veriler arasındaki benzerlikleri/uzaklıkları dikkate alan bir yapı içerisinde kümeler oluşturmayı amaçlar. Bu yöntem, özellikle sınıf etiketlerinin bulunmadığı durumlarda veri yapısını keşfetmek için sıkça tercih edilir. Oluşan hiyerarşik yapı, dendrogram adı verilen ağaç diyagramlarıyla görselleştirilir ve bu sayede farklı kümeleme seviyeleri analiz edilebilir (Kaufman ve Rousseeuw, 1990). Hiyerarşik yöntemler iki ana kategoriye ayrılır:

- *Birleştirici (Agglomerative):* Her gözlem başlangıçta ayrı bir küme olarak kabul edilir; gözlemler ya da kümeler, aralarındaki benzerliklere/uzaklıklara göre aşamalı olarak birleştirilir (Hastie, Tibshirani ve Friedman, 2009).
- *Ayrıştırıcı (Divisive):* Tüm gözlemler başlangıçta tek bir büyük küme altında toplanır; ardından bu küme, belirli uzaklık ölçütlerine göre alt kümelere ayrılır.

Kümelerin nasıl birleştirileceği ya da ayrıştırılacağına, genellikle bağlantı (linkage) kuralları ile karar verilir. En yaygın kullanılan bağlantı türleri şunlardır:

- *Tek bağlantı (Single linkage):* İki küme arasındaki en kısa uzaklık/mesafe,
- *Tam bağlantı (Complete linkage):* En uzun uzaklık/mesafe,
- *Ortalama bağlantı (Average linkage):* Gözlemler arası ortalama uzaklık/mesafe,
- *Ward's yöntemi:* Gruplar arası varyansı minimize edecek şekilde birleştirme (Ward, 1963).

Hiyerarşik kümeleme yöntemlerinin avantajlarından biri, küme sayısının önceden belirlenmesini gerektirmemesidir. Bununla birlikte, işlem bir kez yapıldığında (birleştirme veya ayrıştırma), bu adımlar geri alınamaz; dolayısıyla algoritma esneklikten yoksundur. Ayrıca, büyük veri setlerinde hesaplama maliyeti yükselebilir. Küme ilişkilerinin dendrogram üzerinden ayrıntılı olarak incelenebilmesi açısından oldukça açıklayıcı bir kümeleme algortimasıdır (Tan, Steinbach ve Kumar, 2006).

Dendrogram, gözlemlerin birleşme veya ayrılma noktalarını grafiksel olarak gösterir ve veri içerisindeki alt yapıların sezgisel olarak gözlenmesine olanak tanır. Ayrıca, kullanılan bağlantı ölçütüne bağlı olarak farklı dendrogramlar üretilebilir, bu da yöntemin veri değişimlerine duyarlılığını artırır (Hastie, Tibshirani ve Friedman, 2009).

k-ortalamalar gibi bölümleme kümeleme algoritmaları, önceden belirlenmiş küme sayısına ve başlangıç koşullarına bağlı olarak sonuç üretirken, hiyerarşik kümeleme algoritmaları bu tür bir ön bilgiye ihtiyaç duymaz. Bunun yerine, gözlemler arası benzerlik veya uzaklık ölçütlerine dayanarak, veri seti üzerinde çok seviyeli bir kümeleme yapısı oluşturur. Bu yapı, gözlemlerin birbirleriyle olan ilişkilerine göre şekillenen, iç içe geçmiş kümeleri yansıtan hiyerarşik bir düzen ortaya koyar.

Hiyerarşik birleştirici kümeleme algoritması, her bir gözlemi başlangıçta ayrı bir küme olarak kabul eden ve benzer kümeleleri adım adım birleştirerek hiyerarşik bir yapı oluşturan denetimsiz öğrenme yöntemlerinden biridir. Bu algoritma, özellikle küçük ve orta büyüklükteki veri setleri üzerinde etkili sonuçlar verir. Hiyerarşik birleştirici kümeleme algoritması Çizelge 3.2'de yer almaktadır.

Çizelge 3.2. Hiyerarşik birleştirici kümeleme algoritması

1. Başlangıçta
- Veri setinde n adet gözlem olsun. Her bir gözlem başlangıçta bir küme olarak kabul edilir, dolayısıyla ilk durumda n tane küme vardır.
- Tüm gözlem çiftleri arasındaki uzaklıklar (veya benzerlikler) kullanılarak bir uzaklık matrisi $C \in \mathbb{R}^{n \times n}$ oluşturulur. Her bir hücre $C[i][j], i$ ve j kümeleri arasındaki uzaklık değerini temsil eder.
- Kümelerin birleştirilme sırasını kaydetmek üzere boş bir liste tanımlanır: $A = \emptyset$.
- Kümelerin aktiflik durumunu izlemek için gösterge vektörü $I \in \{0,1\}^n$ tanımlanır. Başlangıçta $I[i] = 1$ tüm i için, yani tüm kümeler aktiftir.

2. Döngüsel Birleştirme Süreci
- Bu aşama, $n - 1$ adımda gerçekleştirilir. Her bir adımda aşağıdaki işlemler tekrarlanır:

a) En Yakın (Benzer) Küme Çifti Seçimi:
- $I[i] = 1$ ve $I[m] = 1, i \neq m$ koşullarını sağlayan aktif ve farklı kümeler arasından en düşük uzaklığa (veya en yüksek benzerliğe) sahip çift $\langle i, m \rangle$ belirlenir:

$$\langle i, m \rangle = \arg \min_{(i,m): i \neq m, I[i]=1, I[m]=1} C[i][m]$$

b) A Listesine Kayıt: Belirlenen çift, birleştirme sırasını temsil eden listeye eklenir:

$$A \leftarrow A \cup \{(i, m)\}$$

c) Yeni Kümenin Uzaklıklarının Güncellenmesi: i ve m kümeleri birleştirildikten sonra, oluşan yeni kümenin diğer aktif kümelerle olan uzaklıkları, kullanılan bağlantı (linkage) yöntemine bağlı olarak güncellenir. Uzaklık matrisi şu şekilde güncellenir:

$$C[i][j] \leftarrow \text{SIM}(i, m, j), C[j][i] \leftarrow \text{SIM}(i, m, j), \forall j \neq i, j \neq m, I[j] = 1$$

- Bağlantı fonksiyonu $\text{SIM}(i, m, j)$, farklı bağlantı kurallarına göre tanımlanabilir.

d) Kümenin Pasifleştirilmesi: Birleştirme işleminde kullanılan kümelerden biri (genellikle m) pasif hâle getirilir:

$$I[m] \leftarrow 0$$

3. Çıktı: Tüm $n - 1$ birleştirme işlemi tamamlandığında, liste A algoritmanın çıktısı olarak döndürülür. Bu liste, kümelerin hangi sırayla birleştirildiğini gösteren hiyerarşik yapıyı tanımlar. Elde edilen bu yapı genellikle bir dendrogram (ağaç diyagramı) ile görsel olarak temsil edilir.

Hiyerarşik kümeleme algoritmaları, gözlemler arasındaki uzaklık veya farklılıkları cbelirlemek amacıyla çeşitli uzaklı uzaklık ölçütlerinden yararlanır. Bu ölçütler, hangi gözlem çiftlerinin veya kümelerin birleştirileceğine karar verme sürecinde belirleyici bir rol oynar. Seçilen uzaklık metriği, elde edilen küme yapısının doğrudan şeklini ve kalitesini etkilediğinden, yöntem tercihi titizlikle yapılmalıdır (Hastie, Tibshirani ve Friedman, 2009). Literatürde yaygın olarak kullanılan başlıca uzaklık ölçütleri ve bu ölçütlere ilişkin açıklamalar Çizelge 3.3'te verilmektedir.

Çizelge 3.3. Uzaklık ölçütlerinin açıklamaları ve formülleri

Uzaklık Ölçümü	Açıklama	Formül
Öklid Uzaklığı	İki veri noktası arasındaki doğrudan, düz çizgi uzaklığını ifade eder.	$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
Manhattan Uzaklığı	İki nokta arasındaki farkın mutlak değerlerinin toplamını ifade eder.	$\sum_{i=1}^{n}\lvert x_i - y_i\rvert$
Minkowski Uzaklığı	Öklid ve Manhattan uzaklıklarının genelleştirilmiş halidir. p parametresiyle kontrol edilir, ölçümün hassasiyetini belirler.	$\left(\sum_{i=1}^{n}\lvert x_i - y_i\rvert^p\right)^{1/p}$
Kosinüs Benzerliği	Veri noktaları arasındaki açısal benzerliği ölçer ve genellikle metin madenciliği veya yüksek boyutlu seyrek (sparsity) veri kümelerinde kullanılır.	$\frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\cdot\sqrt{\sum_{i=1}^{n} y_i^2}}$
Jaccard Benzerliği	İki kümenin ortak elemanlarının, tüm elemanlarına oranını ölçer.	$\frac{\lvert A \cap B\rvert}{\lvert A \cup B\rvert}$

Hiyerarşik kümeleme analizinde ise hem simülasyon hem de genomik veri setleri üzerinde tam bağlantı yöntemi tercih edilmiştir. Bu yöntem, iki küme arasındaki uzaklığı, kümeler arasında bulunan en uzak iki gözlem arasındaki uzaklık/mesafeye göre tanımlar. Bu yaklaşım, küme içi tutarlılığı korurken kümeler arası ayrımı da en üst düzeye çıkarmayı hedefler. Özellikle kümelerin net biçimde ayrılmasının amaçlandığı durumlar için güçlü bir alternatif sunmaktadır. Ancak, bu yöntem tüm gözlem çiftleri arasındaki uzaklıkları gerektirdiğinden, analiz öncesinde bir uzaklık matrisinin oluşturulması zorunludur.

3.2. Sağlam (Robust) Kümeleme Algoritmaları

Gerçek veri kümeleri, genellikle beklenmeyen sapmalar, aykırı gözlemler ya da başka dağılımlardan gözlemler içerebilir. Bu tür unsurlar, kümeleme analizinin doğruluğunu ve güvenilirliğini ciddi biçimde zedeleyebilir. Oysa kümeleme yöntemlerinin temel amacı, benzer özelliklere sahip gözlemleri bir araya getirerek yapısal olarak anlamlı gruplar elde etmektir (Everitt, 1977; Gordon, 1981; Kaufman ve Rousseeuw, 1990). Ancak, klasik kümeleme algoritmaları, bu amaca ulaşırken sağlamlık (robustluk) açısından sınırlı performans gösterir; yani başka dağılımlardan gözlem karışma durumuna veya aykırı gözlemlere karşı duyarlıdırlar.

Bu tür sorunlar, yalnızca birkaç aykırı gözlemin bile küme merkezlerini ciddi ölçüde saptırmasına ve sonuç olarak yanlış gruplamaların yapılmasına neden olabilir. Gerçekte var olmayan kümelerin oluşması, mevcut kümeler arasındaki sınırların bulanıklaşması ya da farklı yapıya sahip gözlemlerin aynı gruba atanması gibi sorunlar sıklıkla ortaya çıkabilir (Liu, Motoda ve Yu, 2002). Bu nedenle, özellikle aykırılıklar içeren veya yapısal heterojenlik gösteren veri setlerinde, klasik yöntemlerle yapılan kümeleme güvenilir sonuçlar veremeyebilir (Croux, Filzmoser ve Fritz, 2011).

Sağlam (robust) kümeleme algoritmaları bu sınırlılıkları aşmak için geliştirilmiştir. Aykırı gözlemler ile ve başka dağılımdan gözlem karışması durumuna karşı (karışma/kontaminasyon) daha dirençli olan bu yöntemler, veri yapısını daha doğru şekilde yansıtan küme yapıları elde etmeyi amaçlar. Sağlam istatistik, özellikle aykırı gözlemlerin birlikte gruplaştığı durumlarda kümeleme analizine büyük katkı sağlar; çünkü bu tür kümelenmiş aykırılıklar klasik yöntemlerin performansını ciddi şekilde düşürebilir (Rocke ve Woodruff,1996).

Bu tür yapılar, analiz üzerinde en fazla bozucu etkiye sahip olabileceğinden, aykırı gözlemlerin kümelenerek değerlendirilmesi önem kazanır. Bazı araştırmacılar, küçük aykırı gözlem gruplarını veya izole aykırıları ayrı kümeler olarak değerlendirmeyi önermektedir (Cuesta-Albertos ve diğerleri, 1997; Hubert, Rousseeuw ve Vanden Branden, 2005; Kaufman ve Rousseeuw, 1990). Bu yaklaşım, özellikle aykırı yapıların açıkça diğer gözlemlerden ayrıldığı durumlarda anlamlı olabilir. Ancak her veri setinde bu stratejiyi uygulamak uygun olmayabilir. Araştırmacı, veri toplama sürecine bağlı olarak kaç grup aradığını önceden biliyor olabilir, fakat aykırı gözlemlerin varlığını ön görmeyebilir. Bu nedenle, sağlam kümeleme yöntemleri yanlızca aykırı gözlemleri tespit etmekle kalmaz aynı zamanda doğru gruplamayı yapabilmek açısından etkili bir yaklaşımdır.

Bu doğrultuda, sağlam istatistiksel yaklaşımları temel alan kümeleme algoritmaları ön plana çıkmaktadır. Bu algoritmalar, verideki aykırı gözlemlerin ya da küçük sapmaların kümeleme sonuçlarını bozmasını engellemek amacıyla tasarlanmıştır. Özellikle *k*-medoids, *k*-medyan ve kırpılmış (trimmed) *k*-ortalamalar algoritmaları, sağlamlık özellikleri sayesinde, veri kümesindeki aykırı gözlemlere ve kontaminasyona karşı klasik yöntemlere kıyasla daha dayanıklı yöntemler olarak ele alınmıştır.

3.2.1. *k*-medoids kümeleme algoritması

k-medoids algoritması, her kümenin merkezini, veriye ait gerçek gözlemler arasından seçerek tanımlayan bir bölümleme yöntemidir. Bu sayede, her bir küme için veri kümesindeki en "temsil edici" gözlem belirlenir ve bu noktaya medoid adı verilir (Kaufman ve Rousseeuw, 1990). *k*-ortalamalar algoritmasının aksine, merkez noktası olarak ortalamalar yerine gerçek gözlemlerin kullanılması, küme merkezlerinin aykırı ya da karışma dağılımından gelen gözlemlere kaymasını engeller. Bu durum, özellikle aykırı değer içeren veri yapılarında algoritmanın daha sağlam sonuçlar üretmesini sağlar. Böylece, hesaplamalarda bu tür değerlerin etkisi azaltılarak gözlemler arasındaki yapısal benzerliklere odaklanmak mümkün hâle gelir (Jain ve Dubes, 1988).

k-medoids yöntemi, özellikle küçük örneklem büyüklüğüne, aykırı gözlemlere ve karışma durumuna sahip KÖÇYB veri yapılarında daha kararlı sonuçlar üretebilme potansiyeline sahiptir (Yang, Yang ve Li, 2020). Ancak, hesaplama maliyeti yüksek olduğundan büyük veri setlerinde uygulanabilirliği sınırlı olabilir. Ayrıca, *k*-ortalamalar kümeleme yönteminde olduğu gibi, küme sayısı olan k değeri önceden belirlenmelidir *k*-medoids algoritmasının adım adım uygulaması, Çizelge 3.4'te yer almaktadır.

Çizelge 3.4. *k*-medoids algoritması

1. Başlangıçta veri kümesi:
- $\mathrm{X} = \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^p$: n adet p-boyutlu uzayda yer alan n gözlemden oluşur.
- k : oluşturulacak küme sayısı
- $\mathrm{C} = \{c_1, c_2, \dots, c_k\}$: her iterasyonda seçilen k adet medoid gözlem (küme merkezleri), doğrudan gözlem noktalarından seçilir.
- $\mathrm{U} = \{u_{ik}\}$: Küme üyelik göstergeleri matrisi; her gözlemin hangi kümeye ait olduğunu belirtir.

2. Başlangıç küme merkezlerinin (medoidlerin) seçilmesi:
Veri kümesinden rastgele k adet gözlem seçilerek başlangıç medoid seti oluşturulur:
$$C^{(0)} = \left\{c_1^{(0)}, c_2^{(0)}, \dots, c_k^{(0)}\right\}$$
3. Gözlemlerin Medoidlere Atanması:
Her bir gözlem x_i, kendisine en yakın mevcut medoidler $C^{(t-1)}$ arasından kendisine en yakın olan c_i ile eşleştirilir ve o medoidin (merkezin) kümesine dahil edilir. Böylece her gözlem en yakın medoidin kümesine atanır.
4. Medoid olmayan bir gözlem x_r, yani mevcut medoidler kümesi $C^{(t-1)}$'nin dışında kalan veri kümesinden $\left(X \setminus C^{(t-1)}\right)$rastgele seçilir. Bu gözlem, potansiyel bir yeni medoid adayıdır. Seçilen bu x_r gözlemiyle, mevcut medoidlerden biri örneğin m_i yer değiştirilerek önerilen yeni medoid C'oluşturulur. Yani c_i çıkarılır, yerine c_r dahil edilir.
5. Toplam Uzaklık (Maliyet) Hesaplaması: Yeni medoid seti C' için uzaklık hesaplanır.
$$S = \sum_{\mathrm{r}=1}^{n} \min_{m \in M'} d(x_r, m)$$

Çizelge 3.4. (devam) *k*-medoids algoritması

Burada dx_r, m, x_r ile medoid m arasındaki uzaklıktır. 6. Eğer yeni toplam uzaklık skoru, önceki toplam uzaklık skorundan küçük ise: $C(t) \leftarrow C'$ Aksi halde: $C(t) \leftarrow C(t-1)$ 7. Yakınsama Kontrolü: Eğer medoidler değişmemişse $C(t)=C(t-1)$ ise algoritma sonlanır. Medoidlerde değişiklik varsa $t \leftarrow t+1$ olarak güncellenir ve 3. Adıma dönülerek süreç tekrarlanır. Not: *k*-medoids algoritması, küme merkezlerini gerçek gözlem noktaları arasından seçerek, *k*-ortalamalara göre aykırılıklara daha dayanıklı (robust) bir yaklaşım sunar. (Kaur, Kaur, ve Singh, 2014).

3.2.2. *k*-medyan (*k*-median) kümeleme algoritması

k-medyan (*k*-median) kümeleme algoritması, MacQueen (1967) tarafından önerilmiş ve Kaufman ile Rousseeuw (1990) tarafından geliştirilerek klasik *k*-ortalamalar algoritmasının daha sağlam bir versiyonu haline getirilmiştir. Özellikle aykırı gözlemler veya kontaminasyon içeren veri setlerinde, küme yapısını koruma açısından daha güvenilir sonuç verir. Bu yöntem, küme merkezlerini belirlerken aritmetik ortalama yerine her bir özelliğe ait medyan değerlerini kullanır. Böylece birkaç aykırı gözlem nedeniyle küme merkezlerinin kayması önlenir. (Godichon-Baggioni ve Surendran, 2022).

k-medyan algoritmasında, küme merkezleri her bir özelliğe ait medyan değerlerinden oluşan çok boyutlu bir vektörle tanımlanır. Bu medyan vektörü, klasik anlamda geometrik medyan değil, her değişkenin sırasıyla eksene dayalı medyanı (her değişken (boyut) için ayrı ayrı medyan değerlerini hesaplar ve bu değerlerden oluşan bir vektör) alınarak oluşturulur. Bu yapı, özellikle yüksek boyutlu verilerde yorumlamayı kolaylaştırırken, algoritmanın aykırı gözlemlere karşı daha sağlam sonuçlar üretmesini sağlar.

k-medyan algoritması, Manhattan uzaklığıdır (L1 normu) temelinde çalışan ve toplam uzaklığı minimize etmeyi amaçlayan bir optimizasyon yaklaşımıdır. Bu norm, iki nokta arasındaki uzaklığı her boyutta mutlak farkların toplamı olarak tanımlar. Algoritma her iterasyonda veri noktalarını en yakın medyan merkeze atar ve ardından her kümenin merkezi, ilgili boyutlardaki koordinatların medyanı alınarak güncellenir (Godichon-Baggioni ve Surendran, 2022).

$$\min_{\{c_1,\ldots,c_k\}} \frac{1}{n} \sum_{i=1}^{n} \min_{j=1,\ldots,k} \left\| x_i - c_j \right\|_1 \tag{3.5}$$

k-medyan algoritmasının adım adım uygulaması, Çizelge 3.5'te yer almaktadır.

Çizelge 3.5. *k*-medyan kümeleme algoritması

1. Başlangıçta
- Veri seti: $X = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^{n\,x\,p}$, n adet p-boyutlu gözlemlerden oluşur.
- Küme sayısı k
- Ceza terimi: δ_{ij}, gözlem i ile özellik j arasındaki uzaklık ölçümünde kullanılan ceza (penalty) değeridir.
- İlk iterasyon $t = 0$ olarak ayarlanır. Küme merkezleri $C^{(0)} = \{{c_1}^0, {c_2}^0 \dots, {c_k}^0\}$, veri seti X içinden rastgele seçilen k gözlem ile başlatılır.

2. Küme Atama: Her gözlemx_i, Manhattan uzaklığı kullanılarak en yakın küme merkezine atanır. Bu atama, aşağıdaki kurala göre atanır.

$$u_{ik}{}^{(t)}: \begin{cases} 1, & eğer\ \mathrm{k} = \arg\min \left\{ \sum_{j\in J} \left| x_{ij} - c_{kj}^{(t-1)} \right| + \sum_{j\in J_0} \delta_{ij} : k = 1, \dots, k, \quad \delta_{ij} (i \in I, j \in J) \right\} \\ 0, & aksi\ halde \end{cases}$$

Burada $u_{ik}{}^{(t)}$, gözlem i'nin k'inci kümeye ait olup olmadığını gösterir.

3. Merkez Güncelleme: Atamalar sabit kabul edilerek, her küme için yeni merkez $V_k^{(t)}$güncellenir. Her kümenin merkezi, kümeye ait gözlemlerin her bir özelliği için eksene dayalı medyan değerinden oluşur:

$$c_{kj}^{(t)} = Medyan\{x_{ij} : i \in {S_k}^{(t)}\}, j = 1,2, \dots, p$$

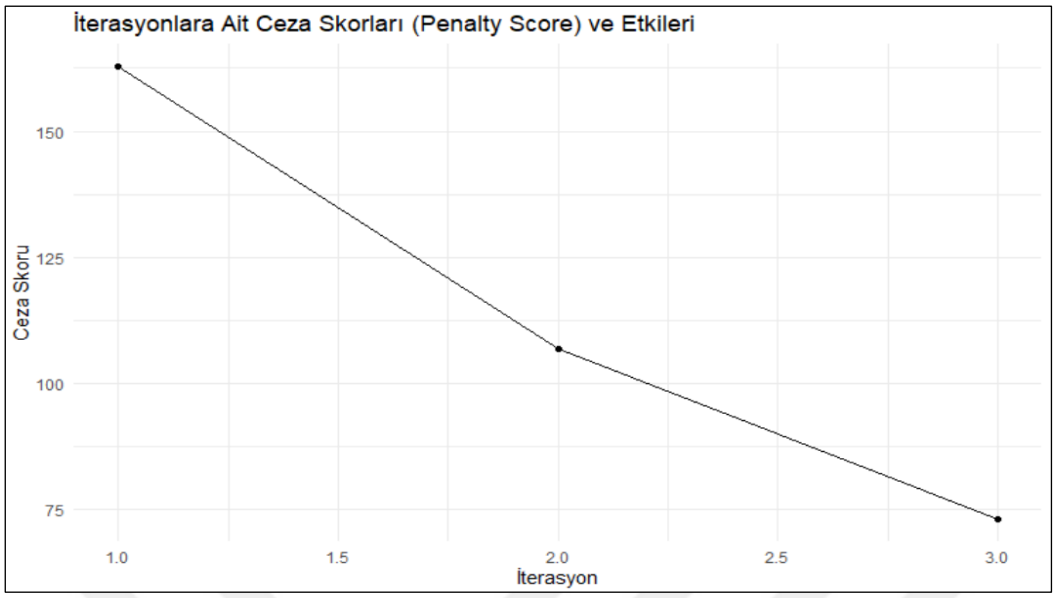
Burada ${S_k}^{(t)}$, t-inci iterasyonda k'inci kümeye atanan gözlemlerdir.

4. Durdurma Kriteri: Eğer üyelik matrisi önceki iterasyonla aynıysa, yani: $u_{ik}^{(t)} = u_{ik}^{(t-1)}$ tüm $i = 1, \dots, n, k = 1, \dots, K$ için sağlanıyorsa algoritma durur. Aksi takdirde adımlar tekrarlanır. (Li, Zhang ve Zhou, 2016).

Karışma durumlarında, algoritmanın küme atama adımında kullanılan uzaklık fonksiyonuna genellikle bir ceza terimi de eklenir. Aşağıda verilen formülde, her gözlem noktası için küme ataması yapılırken uzaklık dışında ek bir bozulma terimi (penalty) dikkate alınır:

$$\arg\min \left\{ \sum_{j\in J} \left| x_{ij} - c_{kj}^{(t-1)} \right| + \sum_{j\in J_0} \delta_{ij} : k = 1, \dots, k \right\} \tag{3.6}$$

Eşitlik (3.6)'da yer alan ceza terimi, özellikle karışma dağılımından gelen bozulmuş gözlemlerin kümeleme üzerindeki etkisini azaltmak amacıyla kullanılmaktadır. Bu yapıda, her gözlemin küme merkezine olan uzaklığına ek olarak, belirli değişkenler için tanımlanan ceza puanları (δ_{ij}) da dikkate alınır. Ceza puanı yüksek olan bir gözlemin toplam uzaklığı (uzaklık + ceza) daha büyük olacağından, algoritma bu gözlemi diğer kümelere göre daha uzak olarak değerlendirir ve o kümeye atamaktan kaçınabilir. Böylece, kontamine veya aykırı gözlemler, küme merkezlerini bozarak algoritmanın genel doğruluğunu düşürecek şekilde kümelere dahil edilmez. Bu yaklaşım sayesinde daha sağlam ve güvenilir bir kümeleme yapısı elde edilir (Li, Zhang, ve Zhou, 2016).



Şekil 3.1. *k*-medyan algoritması için iterasyon sayısına karşılık ceza skorları (162.760, 106.886, 73.207)

Ceza skorlarına ilişkin sonuçlar Çizelge 3.7'de yer almaktadır. İlk iterasyonda ceza skorlarının yüksek çıkması, başlangıçta aykırı gözlemlerin ve kontaminasyonun algoritma üzerindeki etkisinin belirgin olduğunu göstermektedir. Ancak, sonraki iterasyonlarda bu skorların giderek düşmesi, uygulanan ceza mekanizmasının bu olumsuz etkileri başarıyla bastırdığını ve algoritmanın daha güvenilir (aykırı gözlemlerden uzaklaştırılmış) küme merkezleri oluşturduğunu ortaya koymaktadır.

Ceza skorlarının iterasyon boyunca düşmesi, Gündüz ve Fokoué (2015) tarafından da vurgulandığı üzere, yüksek boyutlu ve farklı dağılımlardandan gelen gözlemlerin karıştığı veri yapılarında, varyan artırmasının modele belirsizlik yarattığı ve bu belirsizliği azaltmasında sağlam yaklaşımların ya da ceza terimli-düzenleyici (penalty-based) yöntemlerin kritik rol oynadığını göstermektedir. Nitekim ceza terimleri, algoritmanın hem karışık yapılar hem de aykırı gözlemler karşısında daha dayanıklı çalışmasını sağlamakta, hem de kümelerin doğruluğunu artırmaya yönelik bir düzenleme mekanizması işlevi görmektedir.

3.2.3. Kırpılmış *k*-ortalamalar (trimmed *k*-means) kümeleme algoritması

Kırpılmış *k*-ortalamalar (trimmed *k*-means) algoritması, aykırı gözlemlerin etkisini azaltarak daha sağlam kümeleme sonuçları elde etmeyi amaçlayan önemli bir yöntemdir. Bu

algoritma, klasik *k*-ortalamalar yöntemine dayanmaktadır ancak her iterasyonda, küme merkezlerine en uzak % α oranındaki gözlemler veri setinden çıkarılarak işlem yapılır. Böylece, aykırı gözlemlerin küme merkezlerini ve genel küme yapısını bozması engellenir. Kalan gözlemlerle küme merkezleri yeniden hesaplanır ve bu süreç merkezler kararlı hale gelene kadar devam eder. Bu yaklaşım, Cuesta-Albertos, Gordaliza ve Matrán (1997) tarfından önerilmiştir ve iteratif yapısı sayesinde sağlamlık sağlar.

Klasik *k*-ortalamalar algoritması, ideal koşullar altında başarılı sonuçlar verirken, aykırı gözlemlere karşı oldukça duyarlıdır. Özellikle veri yapısına uymayan veya farklı gözlemler, gerçekten bir kümeye mi ait oldukları yoksa aykırı grup mu oluşturdukları çoğu zaman belirsizdir (García-Escudero, Gordaliza, Matrán ve Mayo-Iscar, 2008). Bu durum, küme merkezlerinin sapmasına, kümelerin bozulmasına ve hatta yanlış küme sayısı tahminlerine yol açabilir. Alternatif olarak geliştirilen k-medoids gibi yöntemler, aykırı gözlemlere karşı daha dayanıklı olmakla birlikte, yüksek oranda aykırılık içeren veri setlerinde yeterince etkili olmayabilirler. Bu bağlamda, García-Escudero ve Gordaliza (1999) tarafından önerilen genelleştirilmiş kırpılmış *k*-ortalamalar algortiması, daha esnek ve teorik açıdan güçlü çözümler sunar. Bu yöntem, sabit bir α oranına dayanmaktan öte, farklı aykırı gözlem belirleme stratejileri, ağırlıklandırma teknikleri ve parametrik olmayan modeller içerebilir. Ancak, bu gelişmiş yapı uygulamada karmaşıklık yaratır ve parametrelerin dikkatli seçilmesini gerektirir.

Literatürde kırpılmış *k*-ortalamalar algoritmasının birçok farklı versiyonu bulunmaktadır. Bu versiyonlar, modelleme düzeyindeki farklılıklar, çeşitli kırpma stratejileri ve algoritmanın yapısal özelliklerinde değişiklikler içermektedir. Ancak bu tez çalışmasında, yorumlanabilirliği yüksek, uygulaması nispeten kolay ve parametrik karmaşıklığı düşük olması nedeniyle klasik kırpılmış *k*-ortalamalar algoritması tercih edilmiştir. Kırpılmış *k*-ortalamalar algoritması Çizelge 3.6'da yer almaktadır.

Çizelge 3.6. Kırpılmış k-ortalamalar kümeleme algoritması

1. X: Veri kümesi, yani n örnekten oluşan veri matrisi
- $X = [x\_1, x\_2, \dots, x\_n]^T \in \mathbb{R}^{nxp}$: n adet p-boyutlu gözlemlerden oluşan veri matrisi
- k: Küme sayısı
- $\alpha \in (0,1)$): Kırpma oranı; toplam verinin $\%\alpha$'lık kısmı aykırı kabul edilerek kırpılır.
- Rastgele k adet gözlem seçilerek başlangıç küme merkezleri belirlenir:

$$C^{(0)} = \left\{c_1^{(0)}, c_2^{(0)}, \dots, c_k^{(0)}\right\}$$

2. Küme Atama (Atama Adımı): Her gözlem x_i, önceki iterasyondaki küme merkezlerine olan uzaklıklar (Öklid normu) kullanılarak en yakın küme merkezine atanır:

$$u_{ik}^{(t)} = \begin{cases} 1 & \text{eğer } k = \arg\min_j \left\|x_i - c_j^{(t-1)}\right\|^2 \\ 0 & \text{aksi halde} \end{cases}$$

Burada $u_{ik}^{(t)}$, i-inci gözlemin k-inci kümeye aitliğini gösteren gösterge değişkendir.

3. Kırpma (Trimleme):
- Her gözlem için, ait olduğu küme merkezine olan uzaklıklar hesaplanır.
- Bu uzaklıklar sıralanır ve toplam $n\ x\ \alpha$ kadar gözlem, en uzak olanlar olarak belirlenir.
- Bu en uzak gözlemler analizden çıkarılır, yani kümeleme işleminin dışında bırakılır.

4. Merkez Güncelleme: Kırpılmış veri setinde kalan gözlemler kullanılarak her kümenin yeni merkezi hesaplanır:

$$c_k^{(t)} = \frac{1}{\left|S_k^{(t)}\right|} \sum_{x_i \in S_k^{(t)}} x_i \quad \text{(yalnızca kırpılmamış veriler için)}$$

Burada $S_k^{(t)}$ t-inci iterasyonda k-inci kümeye atanmış ve kırpılmamış gözlemler kümesidir.

5. Yakınsama Kontrolü: Eğer tüm kümeler için merkezlerdeki değişim çok küçük veya sıfır ise $\left\|c_k^{(t)} - c_k^{(t-1)}\right\| < \varepsilon$ tüm k için çok küçükse (veya sıfırsa), algoritma sonlandırılır. Aksi halde 2. adıma dönülür.

3.3. Temel Bileşenler Analizi ile Küme Yapısını Görselleştirme

Bu çalışmada, KÖÇYB veri setlerinin yalnızca görselleştirilmesinde Temel Bileşenler Analizi (TBA) kullanılmıştır. TBA, çok değişkenli yüksek boyutlu veri setlerinde değişkenler arasındaki ilişkileri analiz ederek, bilgi kaybını en aza indirecek biçimde veriyi daha az boyutla temsil etmeyi amaçlayan yaygın tekniklerden biridir. İlk olarak Pearson (1901) tarafından ortaya atılan ve Hotelling (1933) tarafından geliştirilen bu yöntem, bilgisayar teknolojilerinin ilerlemesiyle geniş bir uygulama alanı kazanmış ve günümüzde neredeyse tüm istatistik yazılımlarında standart bir analiz aracı hâline gelmiştir.

TBA, orijinal değişkenlerin doğrusal kombinasyonlarıyla oluşturulan ve birbiriyle ilişkisiz (bağımsız) yeni değişkenler, yani temel bileşenler aracılığıyla gerçekleştirilir. Bu bileşenler, veri setindeki toplam varyansı en iyi şekilde açıklayacak şekilde sıralanır ve genellikle ilk birkaç bileşen, değişkenliğin büyük bir kısmını temsil ederek boyut indirgemeyi mümkün kılar. Bu yaklaşım, aynı zamanda çoklu doğrusal bağlantı (multicollinearity) sorununu ortadan kaldırır (Mishra ve diğerleri, 2017).

TBA sonucunda elde edilen birinci temel bileşen (TB1), veri setindeki toplam varyansın en büyük kısmını açıklayan ekseni temsil ederken, ikinci temel bileşen (TB2) ise TB1'e dik olacak şekilde konumlanır ve toplam varyansın ikinci büyük kısmını açıklar. Bu iki bileşen genellikle verinin iki boyutlu düzlemdeki projeksiyonu için yeterli olup, kümeler arasındaki ayrımın görselleştirilmesinde oldukça etkilidir (Mishra ve diğerleri, 2017).

Bu çalışmada TBA, kümeler oluşturulup etiketler belirlendikten sonra, verilerin iki boyutlu bir düzleme indirgenerek görselleştirilmesi amacıyla kullanılmış; böylece özellikle KÖÇYB veri yapılarında kümelerin dağılımlarının analizi ve farklı kümeleme yöntemlerinin karşılaştırılması sağlanmıştır.

3.4. Kümelemede Performans Değerlendirme İndeksleri

Kümeleme analizi, denetimli sınıflandırma yöntemlerinden farklı olarak önceden tanımlanmış etiket bilgisine ihtiyaç duymadan, etiketsiz veri üzerinde desen ve yapı keşfi gerçekleştiren bir denetimsiz öğrenme yöntemidir (Han, Kamber ve Pei, 2011). Ancak çoğu durumda veri üzerindeki gerçek sınıf yapısı bilinmediğinden, kümeleme sonuçlarının güvenilirliğinin ve kalitesinin nesnel biçimde değerlendirilmesi gerekmektedir. Bu amaçla, kümeleme başarısı genellikle içsel ve dışsal geçerlik indeksleri aracılığıyla ölçülmektedir (Guyon, Luxburg ve Williamson, 2012).

Bu bağlamda, KÖÇYB veri setleri üzerinde uygulanan klasik ve sağlam kümeleme yöntemlerinin performansı, içsel ve dışsal doğrulama indeksleri kullanılarak analiz edilmiştir. Dışsal doğrulama indeksleri, kümeleme algoritmalarının tahmin ettiği kümelerin doğruluğunu, gerçek sınıf etiketleriyle karşılaştırarak ölçerken; içsel doğrulama indeksleri yalnızca elde edilen kümeleme sonuçlarınına dayanarak, kümeler içindeki benzerlik (tutarlılık) ve kümeler arasındaki farklılığın (ayrılabilirliğin) ne ölçüde sağladığını analiz eder (Chen, Lin ve Huang, 2024).

Bu çalışmada uygulamada kullanılan sekiz veri setinde sınıf etiketlerinin önceden tanımlı olması, algoritma doğruluğunun dışsal ve içsel doğrulama indeksleriyle nesnel biçimde değerlendirilmesine olanak tanımıştır. Literatürde yaygın kabul gören bu yaklaşım, Ullmann, Hennig ve Boulesteix (2021) tarafından kapsamlı biçimde ele alınmış; çalışmada hem içsel hem de dışsal geçerlik ölçütleri incelenmiş ve özellikle dışsal doğrulamanın

kümeleme sonuçlarının güvenilirliğini artırmadaki önemi vurgulanmıştır. Bunun yanı sıra, Zerabi ve Meshoul (2017) dışsal indekslerin büyük veri bağlamındaki uygulanabilirliğine dikkat çekmiştir. Bu tür indeksler, kümeleme algoritmalarının başarımını karşılaştırmalı ve nesnel biçimde değerlendirme olanağı sunar.

3.4.1. Dışsal doğrulama indekslerinin performans karşılaştırılması

Dışsal doğrulama indeksleri**,** kümeleme algoritmalarının elde ettiği sonuçlar ile gözlemler için önceden bilinen sınıf etiketleri arasındaki uyumu değerlendirmek amacıyla kullanılan ölçütlerdir. Bu tür indeksler, kümeleme analizinin doğruluğunu ölçmek için yalnızca algoritma çıktısına değil, aynı zamanda veri setinde mevcut olan gerçek etiketlere de dayanır. Bu nedenle dışsal doğrulama, algoritma çıktılarının nesnel biçimde değerlendirilmesini sağlar ve özellikle farklı algoritmaların karşılaştırmalı analizlerinde önemli bir rol oynar. En sık kullanılan dışsal doğrulama indeksleri arasında Ayarlanmış Rand indeksi **(**Adjusted Rand Index (AR)), Normalize Edilmiş Karşılıklı Bilgi **(**Normalized Mutual Information (NMI)) ve F-skoru **(**F-measure) yer almaktadır. Bu ölçütler, kümelerin gerçek sınıf yapılarıyla ne ölçüde örtüştüğünü sayısal olarak ifade ederek, kümeleme sonuçlarının güvenilirliğini artırmaya katkıda bulunur.

Dışsal küme doğrulaması kapsamında, Steinley (2004) AR kümeleme performansını değerlendirmede sahip olduğu avantajlara dikkat çekmektedir ve bu indeksin yaygın olarak kullanılan diğer dışsal doğrulama indekslerine kıyasla daha üstün sonuçlar verdiğini ortaya koymuştur. Bu nedenle çalışmada dışsal doğrulama için yalnızca AR indeksi kullanılmıştır.

Ayarlanmış rand (AR) indeksi

Bir veri kümesindeki gözlemler $X = \{x_1, x_2, \dots, x_n\}$ ile gösterilsin. Bu gözlemler üzerinde uygulanan kümeleme algoritması sonucunda elde edilen kümeler $U = \{U_1, U_2, \dots, U_R\}$ olsun. Aynı gözlemlerin, önceden bilinen gerçek sınıf etiketlerine göre ait oldukları kümeler ise $V = \{V_1, V_2, \dots, V_C\}$ ile ifade edilmektedir.

Bu durumda Rand, (1971) kümeleme yöntemleri ile gerçek etiketler arasındaki ilişkiyi ölçerken nesnelerin nasıl gruplandığına odaklanır. Bu sebeple ilk sezgisel doğrulama indeksi olan Rand İndeksi'dir (Rİ) geliştirilmiştir. Rİ doğru şekilde kümelenmiş gözlem çiftlerinin

tüm gözlem çiftlerine oranını ölçer. Bu gruplama, gözlemlerin bir çapraz tablo aracılığıyla nasıl sınıflandırıldığını gösterir.

Burada, n sayıda gözlem için $\binom{n}{2}'li$ farklı çift vardır ve bu çiftler dört farklı şekilde sınıflandırılır. Her bir gözlem çifti (x_i, x_j) için olası durumlar Çizelge 3.7'deki gibi tanımlanabilir:

Çizelge 3.7. Kümeleme ve gerçek sınıf etiketlerine göre gözlem çiftlerinin doğruluk sınıflandırması (TP, TN, FP, FN)

		KÜMELEME SONUCU	
		AYNI KÜME	FARKLI KÜME
GERÇEK SINIF	AYNI SINIF	(i) Doğru Pozitif (TP) (Aynı sınıfa ait gözlemler aynı kümeye atanmış)	(ii) Yanlış Negatif (FN) (Aynı sınıfa ait gözlemler farklı kümelere atanmış)
	FARKLI SINIF	(iii) Yanlış Pozitif (FP) (Farklı sınıfa ait gözlemler aynı kümeye atanmış)	(iv) Doğru Negatif (TN) (Farklı sınıfa ait gözlemler farklı kümelere atanmış)

Çizelge 3.7'deki hücreler, kümeleme sonuçları ile gözlemlerin gerçek sınıf etiketleri arasındaki ilişkiye göre iki temel gruba ayrılabilir:

- Tam eşleşen çiftler (yani kümeleme algoritması ile gerçek etiketlerin örtüştüğü durumlar) Doğru Pozitif (TP) ve Doğru Negatif (TN).
- Eşleşmeyen veya çelişen çiftler (yani kümeleme sonucu ile gerçek etiketlerin farklılık gösterdiği durumlar) Yanlış Pozitif (FP) ve Yanlış Negatif (FN).

Bu çerçevede, $A = \mathrm{TP} + \mathrm{TN}$ eşleşen çiftlerin sayısını, $D = \mathrm{FP} + \mathrm{FN}$ eşleşmeyen çiftlerin sayısını temsil eder. Toplam gözlem çifti sayısı ise şu şekilde verilir: A+ D = $\binom{n}{2}$ ve burada $\binom{n}{2}$, n gözlemden oluşturulabilecek tüm gözlem çiftlerini ifade eder ve Rİ Eşitlik (3.7) ile tanımlanır:

$$R\dot{I} = \frac{i+iv}{i+ii+iii+iv} \qquad (3.7)$$

Ancak Rİ, rastlantısal eşleşmeleri dikkate almadığından, tesadüfen oluşabilecek doğru eşleşmeleri de başarı olarak değerlendirir. Bu nedenle, yüksek bir Rİ değeri her zaman anlamlı bir kümeleme başarısı anlamına gelmeyebilir.

Bu sorunu gidermek amacıyla, şansa bağlı eşleşmeleri de dikkate alan Ayarlanmış Rand (AR) indeksi geliştirilmiştir. AR, yalnızca doğru sınıflandırılmış gözlem çiftlerini değil, bu eşleşmelerin rastlantısal olma olasılığını da göz önünde bulundurarak değerlendirme yapar. Böylece kümeleme sonuçlarının doğruluğu daha güvenilir biçimde ölçülür ve şansın etkisi ortadan kaldırılarak daha objektif bir uyum ölçütü sunulur. AR'ın bu biçimi Hubert ve Arabie (1985) tarafından önerilmiştir.

Eşitlik 3.8 ile verilen AR indeksi, kümeleme sonuçları ile gözlemlerin gerçek sınıf etiketleri arasındaki uyumu değerlendirmek amacıyla kullanılan dışsal bir geçerlik ölçütüdür. Bu indeks, -1 ile 1 arasında değer alır ve bu değerlerin büyüklüğü, kümelenen verinin gerçek sınıf yapısıyla ne ölçüde örtüştüğünü gösterir.

$$AR = \frac{\binom{n}{2}(i+iv)-[(i+ii)(i+iii)+(iii+iv)(ii+iv)]}{\binom{n}{2}^2-[(i+ii)(i+iii)+(iii+iv)(ii+iv)]} \tag{3.8}$$

AR değerinin 1'e yakın olması, kümeleme sonuçlarının gerçek sınıflarla mükemmel düzeyde örtüştüğünü ve algoritmanın yüksek doğrulukla çalıştığını gösterir. Pozitif ancak 1'den küçük değerler, kümeler ile gerçek sınıflar arasında belli bir uyum olduğunu; yani gözlemlerin önemli bir kısmının doğru sınıflandırıldığını, ancak bazı hataların da bulunduğunu ifade eder.

AR değerinin 0'a yakın olması, algoritmanın performansının rastgele atama düzeyinde kaldığını, yani anlamlı bir kümeleme yapısı oluşturmadığını gösterir. Negatif AR değerleri ise kümeleme ile gerçek sınıflar arasında sistematik bir uyumsuzluk olduğunu ortaya koyar. Bu durumda, aynı sınıftaki gözlemlerin farklı kümelere, farklı sınıftaki gözlemlerin ise aynı kümeye atanması söz konusudur.

Özellikle AR değerinin -1'e yaklaşması, kümeleme yapısının gerçek sınıf yapısıyla neredeyse tamamen ters olduğunu ve algoritmanın sınıflar arasındaki ayrımı doğru yansıtamadığını gösterir. Bu nedenle, AR indeksinin büyüklüğü yalnızca uyumun derecesini değil, aynı zamanda algoritmanın yapısal başarısını da objektif biçimde yansıtır.

3.4.2. İçsel doğrulama indekslerinin performans karşılaştırılması

İçsel doğrulama indeksleri, kümeleme sonuçlarını yalnızca kümelerin kendi iç yapısal özelliklerine dayanarak değerlendirir. Diğer bir deyişle, bu yöntemler, dışsal sınıf etiketlerine ihtiyaç duymadan algoritmanın başarısını ölçmeyi amaçlar. Değerlendirme, kümeler arası ayrışma (heterojenlik) ve küme içi benzerlik (homojenlik) düzeyine yapılır. Böylece, kümelerin birbirinden ne kadar ayrık ve kendi içlerinde ne kadar tutarlı olduğu belirlenmeye çalışılır.

İçsel doğrulama indeksleri çok sayıda olmakla birlikte, en yaygın kullanılanları arasında Silhouette indeksi (Rousseeuw, 1987), Dunn indeksi (Dunn, 1973), Calinski–Harabasz (CH) indeksi (Caliński ve Harabasz, 1974), Davies-Bouldin indeksi ve S_Dbw (Scatter Density between and within - kümeler arasındaki ve içindeki dağılım yoğunluğu) indeksi yer almaktadır. Bu çalışmada ise, literatürde sıkça tercih edilen ve performans değerlendirmesinde güvenilir sonuçlar veren Silhouette, Dunn ve CH indeksleri kullanılmıştır. Söz konusu indeksler, veri setindeki küme yapısının iç tutarlılığını ve kümeler arasındaki ayrımı ölçmek için uygun ve yaygın kabul görmüş ölçütlerdir.

Silhouette indeksi

Silhouette indeksi, her bir gözlemin kendi kümesine olan yakınlığı ile diğer kümelere olan uzaklığını karşılaştırarak, kümelerin ne ölçüde kompakt (sıkı) ve birbirinden ayrışmış olduğunu analiz eden bir ölçüttür (Rousseeuw, 1987).

Bir gözlem kümesi $X = \{x_1, x_2, \dots x_n\}$ ve bu gözlemlerden elde edilen kümeler $U = U_1, U_2, \dots, U_R$ olmak üzere, her bir $x_i \in U_i$ gözlemi için Silhouette skoru hesaplanır. Bu yöntem, yalnızca kümeleme sonucunda oluşan kümeler ve gözlemler arası uzaklık bilgisine dayanır. Silhouette skorlarını oluşturmak için, uygulanan kümeleme tekniği sonucunda aşağıdaki adımlarla hesaplanır.

- $a(x_i)$: Gözlem x_i'nin ait olduğu küme U_i içindeki diğer tüm gözlemlere olan ortalama uzaklığıdır. Bu değer, gözlemin kendi kümesi ile ne kadar iyi örtüştüğünü, yani küme içi tutarlılığı ifade eder.

- $b(x_i)$: Gözlem x_i'nin kendi kümesi dışındaki kümelere olan ortalama uzaklıklarının en küçüğüdür. Her bir dış küme U_j $(i \neq j)$ için ortalama uzaklık hesaplanır ve bu değerler arasından en küçük olanı seçilir. Bu ölçüt, gözlemin en yakın yabancı kümeye olan benzerliğini ve kümeler arası ayrılığı yansıtır:

$$b(\mathrm{x}_i) = \min_{j \neq i} \frac{1}{|\mathrm{U}_j|} \sum_{\mathrm{x}_j \in \mathrm{U}_j} d(\mathrm{x}_i, \mathrm{x}_j) \tag{3.9}$$

Bu iki ölçüte dayanarak, her bir gözlemi için Silhouette skoru $s(x_i)$, Eşitlik 3.10'daki gibi tanımlanır:

$$s(\mathrm{x}_i) = \frac{b(\mathrm{x}_i) - a(\mathrm{x}_i)}{\max\{a(\mathrm{x}_i), b(\mathrm{x}_i)\}}, \text{ eğer } |\mathrm{U}_i| > 1 \tag{3.10}$$

Silhouette skoru, gözlemin ait olduğu kümeye ne kadar iyi yerleştirildiğini ve diğer kümelerden ne ölçüde ayrıldığını gösterir. Yüksek bir $s(\mathrm{x}_i)$ değeri, gözlemin kendi kümesi içinde sıkı bir şekilde gruplandığını ve diğer kümelere olan uzaklığının fazla olduğunu ifade eder. Bu nedenle, Silhouette skoru, kümeler arası ayrışmanın büyük, küme içi benzerliğin ise yüksek olmasını hedefler yer alır ve şu şekilde yorumlanır:

- $s(x_i) \approx 1$: Gözlem kendi kümesine güçlü şekilde bağlıdır; çok iyi kümelenmiştir.
- $s(x_i) \approx 0$: Gözlem, kendi kümesi ile en yakın diğer küme arasında; sınırdadır, net kümelenmemiştir.
- $s(x_i) \approx$ -1: Gözlem en yakın başka bir kümeye, kendi kümesinden daha yakındır; muhtemelen yanlış kümelenmiştir.

Eğer bir küme yalnızca bir gözlem içeriyorsa ($|U_i| = 1$), $a(\mathrm{x}_i)$ tanımsız olur. Bu durumda, ilgili gözlemin Silhouette değeri $s(x_i)$ sıfır olarak atanır. Bu sebeple eğer $|U_i| > 1$ koşulunu sağlıyorsa, bu değer Silhouette skoru kabul edilir. Tüm gözlemler için hesaplanan bireysel Silhouette skorlarının ortalaması, genel Silhouette indeksi olarak adlandırılır ve kümeleme çözümünün bütünsel kalitesini ölçmekte kullanılır.

Dunn indeksi

Diğer bir içsel doğrulama indeksi ise Dunn indeksi'dir. Dunn (1973), tarafından önerilen bu indeks, kümeler arasındaki ayrışmayı ve içsel yoğunluğu değerlendirmeyi amaçlar ve kompakt ve birbirinden iyi ayrılmış kümelerin oluşturulmasını hedefler.

Dunn indeksi, kümeler arasındaki en küçük uzaklığının, aynı küme içindeki en büyük uzaklığa oranı olarak tanımlanır. Bu şekilde, kümeler arasındaki ayrımın ve kümelerin iç yapısının ne kadar yoğun olduğunun gözlemlenmesine olanak tanır. Küme çapı küçükse, gözlemler birbirine yakın (kompakt) olur, büyükse, gözlemler daha geniş bir alanda yayılmış (yayvan) olarak değerlendirilir. Küme içindeki en büyük uzaklık (Δ_I), x_i ve x_j gözlemlerinin arasındaki (Öklidyen) uzaklıkların en büyük değeri olarak tanımlanır.

$$\Delta_I = \max_{\mathrm{x}_i,\mathrm{x}_j \in \mathrm{U}_I} d\left(\mathrm{x}_i, \mathrm{x}_j\right) \tag{3.11}$$

İki farklı küme arasındaki en küçük uzaklık $\delta\left(\mathrm{U}_I, \mathrm{U}_J\right)$, ise bir kümeye ait x_i gözlemi iken diğer kümeye ait x_j gözlemi arasındaki minimum uzaklık olarak tanımlanır:

$$\delta\left(\mathrm{U}_I, \mathrm{U}_J\right) = \min_{\mathrm{x}_i \in \mathrm{U}_i, \mathrm{x}_j \in \mathrm{U}_j} d\left(\mathrm{x}_i, \mathrm{x}_j\right),\ burada\ I \neq J \tag{3.12}$$

Bu durumda, Dunn indeksi Eşitlik 3.13'deki gibi hesaplanır:

$$\text{Dunn indeksi} = \frac{\min_{I,J} \delta\left(U_I, U_J\right)}{\max_I \Delta_I} \tag{3.13}$$

Dunn indeksi için alt sınır 0 olup, daha büyük değerler daha iyi kümelenme performansını gösterir. Yüksek Dunn İndeksi değerleri, kümelerin birbirinden daha iyi ayrıldığını ve her bir küme içindeki gözlemlerin sıkı bir şekilde gruplanmış olduğunu işaret eder.

Calinski-Harabasz (CH) indeksi

Caliński ve Harabasz (1974) tarafından önerilen CH indeksi, Varyans Oranı Kriteri (VOK) olarak da bilinir ve kümelerinin ne kadar iyi ayrıldığını gruplandığını değerlendiren yaygın bir içsel doğrulama yöntemidir. Bu indeks, kümeler arası ayrımı, küme dışı kareler toplamı

(KDKT), ve küme içi dağılım küme içi kareler toplamı (KİKT) karşılaştırarak, kümelerin ne kadar anlamlı bir şekilde ayrıldığını ölçer.

Küme içi kareler toplamı (KİKT), her bir kümedeki gözlemlerin kendi küme merkezine (centroid) olan uzaklıklarının kareleri toplamıdır. Bu değer ve küme içindeki gözlemlerin benzerliğini ve homojenliğini gösterir. KİKT değeri ne kadar düşükse kümeler o kadar iyi tanımlanmış kabul edilir.

Küme dışı kareler toplamı **(**KDKT**)** ise her küme merkezinin, tüm verilerin genel merkezi (grand centroid) ile olan uzaklığının karelerinin toplamıdır. Bu, kümeler arasındaki farklılıkların büyüklüğünü ifade eder. KDKT değeri ne kadar yüksekse, kümeler arasındaki ayrım o kadar belirgindir. CH indeksi, Eşitlik 3.14 verildiği gibi, KDKT ve KİKT oranını hesaplanır:

$$\mathrm{CH} = \mathrm{VOK} = \frac{\frac{\mathrm{KDKT}}{k-1}}{\frac{\mathrm{KİKT}}{n-k}} \tag{3.14}$$

Bu oran, yüksek değerler aldığında, kümeler arasındaki ayrımın daha belirgin olduğunu ve kümelenmenin daha başarılı olduğunu gösterir. Bu, kümeler arasındaki ayrımın küme içindeki varyasyona kıyasla daha büyük olduğu anlamına gelir.

4. KÜÇÜK ÖRNEKLEM ÇAPLI YÜKSEK BOYUTLU (KÖÇYB) VERİ SETLERİNDE KARŞILAŞILAN ZORLUKLAR

Değişken sayısının (p) çok yüksek, örnek çapının (gözlem sayısının) (n) ise küçük olduğu veri setleri, literatürde genellikle "küçük örneklem çaplı yüksek boyutlu" ya da "büyük p, küçük n" şeklinde tanımlanmaktadır. Bu durum, klasik çok değişkenli analiz yöntemlerinin uygulanabilirliğini ve güvenilirliğini kısıtlamaktadır.

Hall, Marron ve Neeman (2005), sabit örnek çapına karşılık artan değişken sayısı durumunu ele aldıkları çalışmalarında, gözlemler arasındaki uzaklıkların benzeştiğini ve verinin düzgün bir simpleksin köşelerine doğru yakınsadığını göstermiştir. Bu yapı, yüksek boyutlu verilerde klasik yöntemlerin güvenilirliğini azaltan temel nedenlerden biridir. Amaratunga ve Cabrera (2016), KÖÇYB veri setlerinde bu tür problemlerin temelinde boyut sayısının örnek çapına göre aşırı büyük olmasının yattığını ve bu nedenle analiz süreçlerinde çeşitli yapısal zorluklar yaşandığını vurgulamaktadır. Bu veri türünde karşılaşılan başlıca zorluklar şunlardır:

1. Kovaryans Temelli Analiz Problemleri: Klasik analizlerin temelinde yer alan kovaryans matrisi, KÖÇYB ($p \gg n$) veri yapısında çoğu zaman tam ranklı değildir (tekildir) ve bu durum, matrisin tersinin alınamamasına ve dolayısıyla istatistiksel çıkarımların geçersizleşmesine neden olur Ahn, Marron, Müller ve Chi (2007) ile Fokoué ve Titterington (2006), bu tür veri yapılarında klasik kovaryans tahminlerinin yanıltıcı sonuçlar verebileceğini vurgulamaktadır. Ayrıca küreselleştirme varsayımının bozulması (homojen varyans ihlali), analizlerin anlamlı sonuçlar üretmesini engeller ve modelin yapısal bütünlüğünü zayıflattığını belirtmiştir (Fan ve Lv, 2009).
2. Çok boyutluluk ("Curse of Dimensionality"): Boyut arttıkça gözlemler arasındaki farklar azalır ve gözlemler birbirine daha çok benzer hale gelir. Bu durum, modelleme sürecinde genelleme hatalarına, bilgi kaybına ve tahmin zorluklarına yol açar.
3. Aykırı Gözlemler ve Kontaminasyon: Yüksek boyutlu veri setlerinde aykırı gözlemlerin tespiti zorlaşır. Bu aykırılıklar, özellikle sınırlı örnek çapı nedeniyle analizden çıkarılamadığında modelin performansını düşürür. Ahn, Lee ve Lee (2018), bu veri yapılarında sağlam yöntemlerin kullanımının hayati öneme sahip olduğunu belirtmiştir.

4.1. Küçük Örneklem Çaplı Yüksek Boyutlu (KÖÇYB) Verilerde Sağlamlık İhtiyacı ve Aykırılık Etkisi

Kısım 4'de ele alınan zorlukların temelinde, analizlerin aykırı gözlemler karşısında kırılgan hale gelmesi yatmaktadır. Bu nedenle, sağlamlık kavramı KÖÇYB veri yapılarında kritik bir rol oynamaktadır. Sağlamlık, istatistiksel analizlerin model varsayımı ihlalleri ve aykırı gözlemler gibi olumsuzluklara karşı dirençli hale getirilmesini amaçlar. Bu kavram ilk kez Box (1953) tarafından ortaya atılmış, Huber (1964) tarafından teorik temelleri geliştirilmiştir.

KÖÇYB veri setleri $(p > n)$, klasik istatistiksel yöntemlerin güvenilirliğini azaltır. Özellikle kovaryans matrisinin tahmini kararsız hale gelir ve analizlerde ciddi sapmalara neden olur (Bühlmann ve Van de Geer, 2011). Boyut sayısı arttıkça gözlemler arasındaki farklar belirginliğini kaybeder; bu durum "çok boyutluluk (curse of dimensionality)" olarak adlandırılır (Aggarwal ve Reddy, 2013).

Ayrıca bu yapılarda yalnızca klasik aykırı gözlemler değil, aynı zamanda veriye dış dağılımlardan gelen gözlemlerin karışması da (kontaminasyon) sıkça görülür. Bu durum genellikle ε oranında farklı bir dağılımdan gelmesi şeklinde modellenir ve Eşitlik (4.1) ile ifade edilir:

$$p(x \mid \epsilon) = (1-\epsilon) \cdot p_{\text{in}}(x) + \epsilon \cdot p_{\text{out}}(x) \tag{4.1}$$

Bu model, verinin bir kısmının bozucu bir yapı taşıdığını ve analiz sonuçlarını saptırabileceğini gösterir.

Kontaminasyon, özellikle kovaryans matrisinin tersinin hesaplanması gibi işlemleri daha da problemli hale getirir. Hennig, Meila, Murtagh ve Rocci (2015), bu tür durumlarda klasik yöntemlerin duyarlılığını vurgulamış ve sağlam yöntemlerin tercih edilmesi gerektiğini belirtmiştir.

Sonuç olarak, KÖÇYB veri ortamlarında aykırı gözlemler ve kontaminasyonun etkisi, analizlerin güvenilirliğini doğrudan tehdit eder. Sağlam yöntemler, bu olumsuzluklara karşı istatistiksel çıkarımların güvenilirliğini koruyan bir araç olarak ön plana çıkar.

5. GENOMİK VERİ SETLERİ İLE UYGULAMA

Bu bölümde, üçüncü bölümde ayrıntıları verilen kümeleme yöntemlerinin (*k*-ortalamalar, hiyerarşik aglomeratif, *k*-medoids, *k*-medyan ve kırpılmış *k*-ortalamalar), hem klasik ($n > p$) hem de KÖÇYB yapısına sahip toplam sekiz farklı veri seti üzerindeki performansları; varsayılan sonuçlarda bütünlük sağlamak amacıyla, dışsal (AR) ve içsel (CH, Silhouette, Dunn) doğrulama indeksleri kullanılarak karşılaştırmalı biçimde analiz edilmiştir.

5.1. Veri Setlerine Genel Bakış

Kullanılan veri setleri farklı örneklem sayısı (n) ve değişken sayısı (p) özelliklerine sahiptir. İlk iki veri seti düşük boyutlu (n ≫ p), kalan altı veri seti ise KÖÇYB yapısında olup, yani p'nin n'den büyük olduğu durumları temsil etmektedir. Bu veri setleri, genomik alanında gen ifade düzeylerine ilişkin olup, her gözlem için sınıf (kanser türü) etiketi içermektedir. Böylece klasik ve KÖÇYB yapılarındaki yöntemlerin performans farkları ortaya konmuştur.

Aşağıda her bir veri setinin genel özellikleri tanıtılmakta ve Çizelge 5.1'de özet bilgileri sunulmaktadır:

i. Diyabet verisi ($n >> p$): Veri seti, üç değişken temel (glukoz toleransı, insülin düzeyleri, SSPG) ve 145 gözlemden oluşmaktadır. Reaven ve Miller (1979), diyabet veri seti üzerine yaptıkları çalışmada, üç temel biyokimyasal değişkeni (örneğin, glukoz toleransı, insülin düzeyleri vb.) dikkate alarak *k*-ortalamalar kümeleme algoritmasını uygulamış ve üç farklı grup tanımlamışlardır: 'normal', 'kimyasal diyabetik' ve 'belirgin diyabetik' bireyler. Bu çalışma, tıbbi veri üzerinde kümeleme analizlerinin nasıl kullanılabileceğine dair önemli bir örnek sunmuş ve daha sonraki biyomedikal araştırmalara yol göstermiştir.
ii. Seramik çömlek verisi ($n >> p$): Çömlek veri seti 27 gözlem, altı değişken (element içeriği) (Si, Al, Fe, Mg, Ca ve Ti) ve kökenlerine göre Attik ve Eritrean olarak sınıflandırılan Yunan seramik parçalarına ilişkin verileri içermektedir (Stern ve Descoeudres, 1977).
iii. Prostat kanseri verisi ($p >> n$): Bu veri seti, prostat kanseriyle ilişkili mikroarray gen ifade düzeylerinden elde edilmiş olup, Stephenson ve arkadaşları, (2005) yürüttüğü daha kapsamlı bir çalışmanın alt kümesini oluşturmaktadır. Veri seti, 1993 ile 1999 yılları

arasında yalnızca radikal prostatektomi (RP) ile tedavi edilen, klinik olarak lokalize prostat karsinomu tanısı almış 79 hastadan elde edilen doku örneklerini içermektedir. Toplamda 500 gen değişkeninden oluşan bu veri setinde, tümör nüksü (tekrarlama) gösteren 37 hastanın doku örnekleri, tümör nüks göstermeyen 42 hastanınki ise birincil prostat tümör örnekleri olarak sınıflandırılmıştır.

iv. Lenfoma verisi ($p >> n$): Bu veri seti, lenfoma hastalığına ilişkin mikroarray gen ifade düzeylerini içermektedir. Veri seti, 180 gözlem (örnek) ve 661 gen değişkeninden oluşmaktadır. Veri, Alizadeh ve diğerleri, (2000) tarafından gerçekleştirilen çalışmadan alınmıştır ve üç farklı lenfoma alt tipini sınıflandırmak amacıyla kullanılmıştır: Diffüz büyük B hücreli lenfoma (DLBCL), foliküler lenfoma (FL) ve kronik lenfositik lösemi (CLL). Çalışma, gen ekspresyon profillerini kullanarak bu alt tiplerin birbirinden ayırt edilebileceğini göstermeyi amaçlamıştır.

v. Akciğer kanseri verisi ($p >> n$): Bu veri seti, akciğer kanseri hastalarına ait mikroarray gen ifade düzeylerini içermekte olup, literatürdeki birçok mevcut akciğer kanseri veri setinden yalnızca biridir. Bu çalışmada kullanılan versiyon, 197 gözlem (hasta örneği) ve 1000 gen değişkeninden oluşmaktadır. Veri seti, Gordon (1981) tarafından yürütülen çalışmada yayımlanmış olup, gen ekspresyon profillerinin akciğer kanseri türlerinin sınıflandırılmasında kullanılabilirliğini ortaya koymayı amaçlamaktadır. Özellikle Akciğer zarından kaynaklanan agresif bir kanser türü ve akciğer dokusundan kaynaklanan ve akciğer kanserinin en yaygın alt türlerinden biri olan kanser örneklerini ayırt edebilen genlerin belirlenmesi hedeflenmiştir.

vi. Kolon kanseri verisi ($p >> n$): Toplam 62 örnekten oluşmakta olup, bunların 40'ı kolon kanseri hastalarına, 22'si ise sağlıklı bireylere aittir. Veri setinde yer alan 2000 gen, tüm örneklerde belirli bir düzeyin üzerinde ekspresyon sergileyen ve biyolojik olarak anlamlı kabul edilen genlerden seçilmiştir. Bu genler, her örnekte minimum ekspresyon düzeylerinin yüksek olduğu genler arasından seçilerek belirlenmiştir. Böylece, düşük veya sıfıra yakın ekspresyon gösteren, dolayısıyla analize anlamlı katkı sağlamayabilecek genler dışlanmıştır. Bu veri seti, kolon kanseri ile sağlıklı durum arasındaki genetik farkların belirlenmesinde yaygın olarak kullanılmaktadır. (Alon ve diğerleri, 1999).

vii. Lösemi verisi ($p >> n$): Toplam 3571 değişken (özellik) ve 72 örnekten oluşmaktadır. Bu lösemi veri setinde, Örneklerden 49'u akut lenfoblast lösemi (ALL), 23'ü ise akut miyeloid lösemi (AML) vakalarını temsil eder. Her örnek, 3571 gen üzerinden ölçülen

gen ekspresyon seviyelerini içerir. Bu veri seti, mikrodizilim analiz yöntemlerinin geliştirilmesinde sıklıkla referans olarak kullanılmaktadır (Golub ve diğerleri, 1999).

viii. Beyin kanseri verisi ($p >> n$): Pomeroy ve diğerleri (2002) tarafından sağlanan bu veri seti, 42 beyin kanseri hastasına ait gen ekspresyon profillerini içerir. Her hasta örneği, 5597 mikrodizilim gen ifade özelliği ile temsil edilmiştir. Veri seti, 5 farklı beyin kanseri türünü kapsamaktadır. Yüksek boyutlu genetik veri analizinde sıkça kullanılan veri setlerinden biridir (Pomeroy ve diğerleri, 2002).

Çizelge 5.1. Uygulamada kullanılan veri setlerinin özellikleri

Veri setleri	n	p	n/p Oranı	Aykırı Gözlem Oranı (%)	Etiket sayısı
Diyabet	145	3	48.3333	0.1517	3
Seramik	27	6	4.5000	0.0185	2
Prostat	79	500	0.2723	0.0498	2
Akciğer kanseri	197	1000	0.1970	0.4798	4
Kolon	62	2000	0.0310	0.5848	2
Lenfoma	180	661	0.2723	0.0498	3
Lösemi	72	3571	0.0201	1.9909	2
Beyin kanseri	42	5597	0.0075	7.0964	5

Çizelge 5.1'de, her bir veri setine ait örnek çapı (n), değişken sayısı (p), n/p oranı, aykırı gözlem oranı (%) ve küme (sınıf) sayısı gibi temel istatistiksel bilgiler özetlenmiştir. Veri setlerinde n/p oranının 1'den büyük olduğu durumlar klasik veri yapısı olarak değerlendirilirken, bu oranın 1'in altında kaldığı durumlar KÖÇYB veri yapılarını göstermektedir.

Her bir veri setine ilişkin aykırı gözlem oranı, veri setinin doğal yapısına bağlı olarak belirlenmiştir. Aykırı gözlemler, çeyrekler arası açıklık (IQR) yöntemine dayalı olarak tespit edilmiştir. Bu kapsamda, her değişken için 1. çeyrek (Q_1) ve 3. çeyrek (Q_3) değerleri hesaplanmış, $IQR = Q_3 - Q_1$ tanımıyla çeyrekler arası açıklık bulunmuştur. Aykırı gözlemler ise $Q_1 - 1.5 \times IQR$ ve $Q_3 + 1.5 \times IQR$ sınırlarının dışında kalan gözlemler olarak tanımlanmıştır. Tespit edilen aykırı gözlem oranları, kırpılmış (trimmed) k −ortalamalar algoritması için kırpma oranını belirlemede temel alınmış ve her veri setine özgü kırpma oranı ile analiz edilmiştir.

Çalışmada kullanılan tüm veri setleri önceden etiketlenmiş sınıflandırma verisi niteliğindedir. Kümeleme algoritmalarının uygulanmasında, küme sayısının belirlenmesi

süreci genellikle ayrı analizler gerektirse de bu çalışmada her bir veri setinin yapısal ve alan bilgisine dayalı olarak sahip olduğu mevcut etiket sayısı, küme sayısı olarak kabul edilmiştir.

5.2. Kümeleme Algoritmalarında R Paketleri ve Fonksiyonların Kullanımı

Bu tez çalışmasında ele alınan kümeleme algoritmaları hem uygulamada hem de simülasyon aşamalarında, R programlama dili ile kümeleme yöntemlerine özgü paketler ile fonksiyonlar aracılığıyla uygulanmıştır. Her algoritmada kullanılan fonksiyonlar, veri yapısı ve algoritmanın özelliklerine uygun şekilde seçilmiştir. Sırasıyla *k*-ortalamalar, hiyerarşik kümeleme, *k*-medoids, *k*-medyan ve kırpılmış *k*-ortalamalar algoritmaları için kullanılan R paketleri ve ilgili fonksiyonlara aşağıda sunulmuştur. Ayrıca performans değerlendirme indeksleri ve küme yapılarını görselleştirmek amacıyla kullanılan TBA yöntemine ilişkin R paket ve fonksiyonları da belirtilmiştir

- *k*-ortalamalar algoritması, R'ın temel paketlerinden biri olan stats paketi içerisinde yer alan kmeans() fonksiyonu kullanılarak gerçekleştirilmiştir. Bu fonksiyon, kullanıcı tarafından belirlenen küme sayısı doğrultusunda 25 farklı rastgele başlangıç noktası ile çalışarak en uygun çözümü seçmekte ve her gözleme bir küme etiketi atamaktadır.
- Hiyerarşik (Birleştirici) kümeleme analizleri ise yine stats paketinde bulunan hclust() fonksiyonu ile gerçekleştirilmiş, kümelerin belirlenmesinde ve her bir gözleme ait küme etiketlerinin atanmasında cutree() fonksiyonundan yararlanılmıştır.
- *k*-medoids kümeleme algoritması, cluster paketinde yer alan pam() fonksiyonu aracılığıyla uygulanmıştır. Fonksiyon, veri matrisi ve küme sayısı parametreleriyle çalışarak her gözlem için küme üyeliği etiketlerini sağlamaktadır.
- *k*-medyan algoritması ise iki farklı bağlamda uygulanmıştır. Gerçek veri uygulamasında, karışma (kontaminasyon) etkisinin olmadığı genomik mikro-dizilim verisi üzerinde doğrudan Kmedians() fonksiyonu kullanılmıştır. Buna karşın, simülasyon çalışmalarında, R ortamında ceza terimli bir *k*-medyan algoritması tanımlı olmadığından, tez kapsamında geliştirilen kullanıcı tanımlı k_median_custom_with_penalty() fonksiyonu kullanılmıştır. Bu fonksiyon, L1 normuna (Manhattan uzaklığına) dayalı olarak çalışmakta; her iterasyonda küme merkezlerini medyan vektörleri ile güncellemekte ve karışma oranına göre ceza skorlarını dikkate alarak sağlam bir kümeleme yapısı sunmaktadır. Simülasyon sürecinde, $t = 3$ iterasyon boyunca ceza skorları hesaplanarak yöntemlerin karşılaştırılması sağlanmıştır.

- Kırpılmış k-ortalamalar algoritması için ise trimcluster paketinde yer alan trimkmeans() fonksiyonu kullanılmıştır. Bu fonksiyon, klasik k-ortalamalar algoritmasının sağlamlaştırılmış bir versiyonu olup, belirli bir kırpma oranına (α) göre aykırı gözlemleri analiz dışında bırakarak daha güvenilir kümeler oluşturmayı hedeflemektedir.
- Kümeleme performans indeksleri ise R paket programında aşağıdaki fonksiyonlar aracılığıyla hesaplanmıştır.
- Ayarlanmış Rand (AR) indeksi: Gerçek sınıf etiketleri ile kümeleme sonuçları arasındaki uyumu değerlendirmek amacıyla, mclust paketinde yer alan adjustedRandIndex() fonksiyonu kullanılmıştır.
- Silhouette indeksi: Her bir gözlemin, ait olduğu küme ile en yakın komşu küme arasındaki uzaklık farkını normalize ederek ölçen Silhouette skoru, cluster paketindeki silhouette() fonksiyonu aracılığıyla hesaplanmıştır. Bu fonksiyon her bir gözlem için:

1. Ait olduğu küme etiketini,
2. En yakın ikinci kümenin etiketini,
3. Silhouette skorunu

içeren bir matris döndürmektedir. Bu matrisin üçüncü sütunundaki değerlerin ortalaması, genel Silhouette indeksini vermekte ve kümeleme yapısının bütünsel kalitesini yansıtmaktadır.

- Dunn indeksi: Küme içi tutarlılığı ve kümeler arası ayrımı birlikte değerlendiren bu indeks, fpc paketinde yer alan cluster.stats() fonksiyonu ile elde edilmiştir.
- Calinski-Harabasz (CH) indeksi: Küme içi varyans ile kümeler arası varyans oranına dayanan Calinski-Harabasz skoru da yine fpc paketinin cluster.stats() fonksiyonu kullanılarak hesaplanmıştır.

Yüksek boyutlu verinin iki boyutlu düzlemde görselleştirilmesi amacıyla, R'ın temel paketlerinden biri olan stats paketinde bulunan prcomp() fonksiyonu kullanılarak TBA uygulanmıştır. Elde edilen temel bileşenler ve kümeleme etiketlerini içeren veri çerçevesi, ggplot2 paketi kullanılarak iki boyutlu düzlemde görselleştirilmiştir.

5.3. Veri Setlerine Kümeleme Yöntemlerinin Uygulanması ve Doğrulama İndeksleri

Bu bölümde, beş farklı kümeleme algoritmasının sekiz farklı veri seti üzerindeki performansları dört farklı doğrulama metriği kullanılarak değerlendirilmiştir. Değerlendirilen algoritmalar; iki klasik yöntem olan *k*-ortalamalar ve hiyerarşik kümeleme ile üç sağlam yöntem olan *k*-medoids, *k*-medyan ve kırpılmış *k*-ortalamalardır.

Algoritmaların her bir veri setindeki başarısı, Ayarlanmış Rand (AR), Calinski-Harabasz (CH), Silhouette ve Dunn indeksleri ile ölçülmüştür. Bu doğrulama indekslerine göre elde edilen karşılaştırmalı sonuçlar Çizelge 5.2'de sunulmuştur. Ayrıca, her algoritmanın veri setlerindeki başarı sıralamaları, Ek-1'de daha ayrıntılı biçimde verilmiştir.

- Klasik ve sağlam kümeleme algoritmalarının genel performansları, farklı veri setleri ve doğrulama ölçütleri üzerinden değerlendirildiğinde aşağıdaki sonuçlara ulaşılmıştır: *k*-ortalamalar algoritması, özellikle CH indeksi bakımından güçlü performans sergilemiştir. Örneğin, Diyabet verisinde CH değeri 233.076'dır. Ancak AR ve Dunn indekslerinde bu başarı genellenememektedir. Akciğer verisinde AR 0.904 gibi yüksek bir değerdeyken, Diyabet verisinde yalnızca 0.380'dir. Silhouette indeksinde ise *k*-ortalamalar genellikle orta düzey performans göstermiştir (Diyabet: 0.582). Bu durum, algoritmanın CH indeksi açısından güçlü ancak diğer indeksler bakımından sınırlı başarı sağladığını göstermektedir. Genel olarak, *k*-ortalamalar algoritması özellikle CH doğrulama metriğinde güçlü sonuçlar üretmektedir; ancak AR, Silhouette ve Dunn indeksleri bakımından tutarlı bir üstünlük sağlamamaktadır. Bu durum, algoritmanın yalnızca belirli yapıdaki veri setlerinde etkili olduğunu göstermektedir.
- Hiyerarşik kümeleme algoritması, özellikle Dunn indeksi bakımından birçok veri setinde yüksek başarı sergilemiştir. Örneğin, Beyin veri setinde elde edilen Dunn değeri 0,646'dır. Bu başarı, Silhouette indeksi tarafından da kısmen desteklenmektedir. Öte yandan, Hiyerarşik kümeleme yönteminin AR ve CH indeksleri açısından tutarlı bir üstünlük göstermemektedir.

Çizelge 5.2. Kümeleme algoritmalarının içsel ve dışsal doğrulama indekslerine göre sonuçları

Algoritmalar	Veri Setleri	AR İndeksi $[-1, 1]$	CH indeksi (>1, sonsuz)	Silhouette İndeksi $[-1, 1]$	Dunn indeksi (>0)
k-ortalamalar	Diyabet	0,380	233,076	0,582	0,054
	Seramik	0,083	13,623	0,344	0,359
	Beyin	0,452	3,80	0,086	0,615
	Kolon	-0,0058	25,810	0,225	0,292
	Lösemi	0,186	7,663	0,079	0,548
	Akciğer	0,904	40,197	0,233	0,462
	Lenfoma	0,324	20,185	0,089	0,383
	Prostat	0,058	63,275	0,337	0,356
Hiyerarşik	Diyabet	0,376	171,142	0,634	0,156
	Seramik	-0,0078	10,222	0,346	0,413
	Beyin	0,227	2,95	0,100	0,646
	Kolon	-0,0057	21,030	0,208	0,321
	Lösemi	0,185	6,592	0,067	0,551
	Akciğer	0,757	38,496	0,214	0,470
	Lenfoma	0,332	17,786	0,087	0,415
	Prostat	0,125	38,959	0,276	0,271
k-medoids	Diyabet	0,382	221,115	0,518	0,047
	Seramik	0,138	12,791	0,325	0,305
	Beyin	0,610	3,51	0,071	0,609
	Kolon	-0,0122	24,589	0,216	0,276
	Lösemi	0,135	6,958	0,073	0,465
	Akciğer	0,935	39,963	0,240	0,463
	Lenfoma	0,340	18,852	0,083	0,325
	Prostat	0,105	62,174	0,335	0,324
k-medyan	Diyabet	0,473	212,392	0,527	0,047
	Seramik	0,0837	13,623	0,344	0,359
	Beyin	0,639	3,48	0,069	0,608
	Kolon	-0,0113	25,628	0,224	0,283
	Lösemi	0,435	7,122	0,077	0,495
	Akciğer	0,338	33,856	0,101	0,345
	Lenfoma	0,542	19,692	0,097	0,400
	Prostat	0,058	63,275	0,337	0,356
Kırpılmış *k*-ortalamalar	Diyabet	0,424	70,070	0,470	0,027
	Seramik	0,0540	8,830	0,261	0,387
	Beyin	0,445	3,22	0,073	0,555
	Kolon	-0,0051	15,678	0,176	0,266
	Lösemi	0,133	4,548	0,063	0,482
	Akciğer	0,736	27,012	0,194	0,254
	Lenfoma	0,348	13,207	0,066	0,282
	Prostat	0,018	17,424	0,255	0,185

- *k*-medoids algoritması özellikle, CH indeksi açısından birçok veri setinde yüksek skorlar üreterek istikrarlı bir başarı sergilemiştir. Örneğin, Diyabet veri setinde elde edilen CH değeri 221,115, bu başarının dikkat çekici bir örneğidir. Ancak bu algoritma AR, Silhouette ve Dunn indeksleri açısından her zaman tutarlı bir üstünlük sağlamamaktadır. Örneğin, KÖÇYB yapısına sahip Akciğer veri setinde AR değeri 0,935 ile oldukça

yüksek bir başarı elde edilirken, Diyabet veri setinde bu değer yalnızca 0,610 olarak hesaplanmıştır.

- *k*-medyan algoritması, özellikle CH indeksi açısından birçok veri setinde yüksek başarı göstermiştir (örneğin, Diyabet: 212,392). AR ve Silhouette indeksleri açısından da algoritma bazı veri setlerinde dikkat çekici performans sergilemiştir. Örneğin, AR değerleri: Beyin (0,639), Lenfoma (0,542), Diyabet (0,473); Silhouette değerleri: Diyabet (0,527), Seramik (0,344), Prostat (0,337). Dunn indeksi bakımından ise genel olarak orta düzeyde bir performans gözlenmektedir. Bu bulgular, *k*-medyan algoritmasının CH ve Silhouette indekslerinde ve belirli veri yapılarında etkili olduğunu (Diyabet $n >> p$); ancak her doğrulama ölçütü için genel ve tutarlı bir üstünlük göstermediğini ortaya koymaktadır.
- Kırpılmış *k*-ortalamalar, genel olarak doğrulama indeksleri açısından diğer yöntemlere kıyasla daha sınırlı başarı sergilemiştir. AR, CH, Silhouette ve Dunn indekslerinde genellikle orta düzeyde değerlere ulaşmıştır. Bununla birlikte, bazı veri setlerinde (örneğin Beyin: Dunn = 0,555, Silhouette = 0,470) öne çıkan değerler elde edilmiştir. Ancak bu başarı, genellenebilir ve tutarlı bir üstünlük düzeyine ulaşmamaktadır. Bu durum, algoritmanın belirli yapıdaki veri setlerinde etkili olabileceğini, ancak her indekste istikrarlı bir başarı sağlamadığını göstermektedir.

Analiz bulguları, her bir veri setinin kendine özgü yapısal özellikleri doğrultusunda en uygun kümeleme algoritmasının farklılık gösterdiğini ortaya koymaktadır.

Diyabet veri seti için *CH ve Silhouette indeksleri* bakımından en iyi performans *k-ortalamalar algoritması* tarafından sağlanırken, AR açısından en başarılı sonuç *k-medyan algortiması* ile elde edilmiştir. Bu durum, farklı indekslerin farklı yönleri ölçtüğü ve tek bir algoritmanın her açıdan üstünlük sağlamasının zor olduğunu göstermektedir.

Beyin veri setinde, *AR* değerleri açısından *k*-medyan algoritması öne çıkarken, *Dunn indeksi* bakımından en yüksek başarı *Hiyerarşik kümeleme algoritması* ile elde edilmiştir. Bu durum, veri setinin yüksek boyutluluk ve dengesiz sınıf dağılımı gibi karmaşık yapısal özellikler taşıyabileceğine işaret etmektedir.

Akciğer veri seti, istisnai bir şekilde tüm doğrulama indeksleri açısından *k*-medoids algoritması ile yüksek başarı sergilemiş ve bu algoritmanın bu özel veri yapısına oldukça uygun olduğunu göstermiştir.

Prostat veri setinde ise *CH* ve *Silhouette* indeksleri açısından *k-ortalamalar* ve *k-medoids* algoritmalarının benzer ve görece yüksek performanslar sergilediği görülmektedir. Bu durum, her iki algoritmanın bu veri setindeki küme yapısını benzer şekilde modelleyebildiğini düşündürmektedir.

Son olarak, Kolon veri seti genel olarak tüm algoritmalarda düşük başarı düzeyi göstermiştir. Ancak göreli olarak, *CH indeksi* bakımından *k-ortalamalar, Dunn indeksi* açısından ise *Hiyerarşik algoritma* daha iyi sonuçlar üretmiştir. Bu bulgu, veri setinin karmaşık, belirsiz ya da kümelenebilirlik açısından zayıf bir yapıya sahip olduğunu düşündürmektedir.

Sonuçlar, farklı doğrulama indekslerinin farklı algoritma-veri seti kombinasyonlarında öne çıktığını ve bu durumun büyük ölçüde veri setinin yapısal özelliklerine bağlı olduğunu göstermektedir. Özellikle, AR indeksine göre bakıldığında, *k-medyan algoritması,* en fazla sayıda veri setinde (Diyabet, Beyin, Lenfoma, Lösemi) başarılı sonuçlar üretmiş ve bu yönüyle öne çıkmıştır. Bu veri setlerinin büyük kısmı, değişken sayısının örnek çapından (gözlem sayısından) fazla olduğu $p > n$ yapısına sahiptir; bu da *k*-medyan'ın yüksek boyutlu, düşük gözlemli veri yapılarında daha dayanıklı çalıştığını göstermektedir.

CH indeksi açısından ise *k-ortalamalar, k-medoids ve k-medyan* algoritmaları birçok veri setinde (örneğin Diyabet, Prostat, Akciğer) yüksek değerler üretmiştir. Bu algoritmaların özellikle değişken sayısının daha az olduğu $n > p$ yapısına sahip Diyabet ve Seramik veri setlerinde güçlü sonuçlar verdiği görülmektedir. CH indeksinin varyans temelli olması, özellikle veri yapısının daha düzenli ve sınıflar arası ayrımın daha net olduğu durumlarda bu algoritmaların avantaj sağladığını göstermektedir.

Silhouette ve Dunn indeksleri, daha çok küme içi tutarlılığı ve kümeler arası ayrımı birlikte dikkate alan indekslerdir. Bu ölçütler bakımından en başarılı sonuçlara genellikle *Hiyerarşik kümeleme algoritması* ile ulaşılmıştır. Özellikle Silhouette indeksinde (Diyabet: 0,634, Seramik: 0,346, Prostat: 0,276) ve Dunn indeksinde (Beyin: 0,646, Lösemi: 0,551, Akciğer:

0,470) öne çıkması, hiyerarşik yaklaşımın özellikle belirgin ayrım barındıran ya da daha az gürültülü veri setlerinde etkili sonuçlar verdiğini göstermektedir.

Bu değerlendirmeler sonrası, "en iyi genel algoritma" seçimi, öncelikli olarak hangi doğrulama metriğinin analizde esas alındığına ve veri setinin yapısal özelliklerine, özellikle (n/p) oranına ($n > p$ mi yoksa $p > n$ mi?) bağlı olarak değişmektedir. Örneğin, AR indeksi açısından *k*-medyan algoritması, Diyabet, Beyin, Lenfoma ve Lösemi veri setlerinde sağladığı yüksek başarılarla öne çıkmakta; bu yönüyle özellikle yüksek boyutlu ($p > n$) veri yapılarında etkili olmaktadır. CH indeksi bakımından ise *k*-ortalamalar, *k*-medoids ve *k*-medyan algoritmaları, Diyabet, Prostat ve Akciğer veri setlerinde dikkat çekici sonuçlar üretmiştir. Silhouette ve Dunn indeksleri açısından en başarılı sonuçlar genellikle Hiyerarşik kümeleme algoritması tarafından elde edilmiş olup, bu durum söz konusu yöntemin yapısal tutarlılık ölçütlerinde güçlü performans sergilediğini göstermektedir. Bu bağlamda, analiz sürecinde kullanılacak doğrulama indekslerinin önceliği ile veri yapısının dikkatli bir şekilde değerlendirilmesi, uygun algoritmanın seçilmesinde kritik bir rol oynamaktadır.

6. SİMÜLASYON ÇALIŞMASI

Simülasyon çalışmasında, KÖÇYB veri yapılarında, *k*-ortalamalar, hiyerarşik, *k*-medoids, *k*-medyan ve kırpılmış *k*-ortalamalar gibi farklı kümeleme algoritmaları; örnek çapı (n), değişken sayısı (p), (n/p) oranları, aykırı gözlem ve karışma oranlarına göre uygulanmıştır. Performans karşılaştırmaları, AR, CH, Silhouette ve Dunn indeksleri kullanılarak yapılmış; tüm analizler R programlama dili (sürüm 4.4.2–4.4.3) aracılığıyla yürütülmüştür.

Simülasyon algoritması aşağıdaki gibidir:

I. k ayrı küme içeren ve her bir gözlemin p boyuttan oluştuğu yapay bir veri seti oluşturulması:

1. Toplam gözlem sayıları $n = (20, 30, 75, 200, 45, 275)$ ve her gözleme için değişken (boyut) sayıları $p = (200, 150, 250, 500, 50, 50)$ olarak belirlenmiştir.
2. Küme sayısı sabit tutulup $k = 3$ olarak seçilmiştir.
3. Her veri setine, toplam örnek çapının $\%20$'si kadar aykırı gözlem eklenmiştir. Kontaminasyon oranı ise $\%0$'dan $\%45$'e kadar, $\%5$'lik artışlarla değişmektedir.
4. Küme merkezleri, her biri p boyutlu olacak şekilde, k adet merkez normal dağılımdan rastgele üretilmiştir.
5. Her kümenin merkezi, kullanılan algoritmanın yapısına uygun biçimde (ortalama, medyan veya medoid) tanımlanmış ve veriler bu merkezler etrafında normal dağılımdan üretilmiştir
6. Veri üretimi öncesinde np boyutunda bir boş veri matrisi ve gözlemlerin ait olduğu küme etiketlerini tutmak üzere uzunluğu n olan boş bit gerçek_etiketler vektörü oluşturulmuştur.
7. Her kümenin kaç gözlem içereceği $\lfloor n/k \rfloor$ olarak belirlenmiş, kalan gözlemler ise sırayla kümelere eklenerek adil dağılım sağlanmıştır.
8. Her küme için belirlenen gözlem sayıları kadar veri, ilgili kümenin merkezi ve sabit bir standart sapma $(sd = 3)$ kullanılarak normal dağılımdan rastgele üretilmiştir. Bu gözlemler veri matrisine eklenmiş ve karşılık gelen doğru sınıf etiketleri de $gerçek\_etiketler$ vektörüne kaydedilmiştir.

II. Bu veri setine aykırı gözlemler ve/veya kontaminasyon (bozulma) eklenmesi

1. Her veri setine aykırı gözlemler, ortalaması 30 ve standart sapması 15 olan normal dağılımdan üretilerek eklenmiştir. Bu gözlemler veri matrisine satır olarak dahil edilmiş ve $gerçek\_etiketler$ vektöründe "0" olarak tanımlanmıştır.
2. Kontaminasyon oranına göre kontamine (bozulacak) olacak örnek çapı belirlenmiş ve kontaminasyon verilerinin eklenmesi için ham veriden çıkarılacak gözlemler rastgele seçilmiştir. Rasgele çıkarılacak ham gözlemler, ortalaması 20 ve standart sapması 10 olan normal dağılımdan gelen yeni değerlerle güncellenmiş; ancak ait oldukları kümenin etiketleri korunarak veri setine entegre edilmiştir.
3. Bu işlem sonucunda $Ham\ veri + Aykırı\ gözlemler + Kontaminasyon$ içeren gözlemlerden oluşan "karma veri seti" elde edilmiştir.
4. Karma veri setindeki tüm değişkenler, farklı ölçeklerin etkisini ortadan kaldırmak için scale() fonksiyonu kullanılarak normalize (standart normal) edilmiştir. Böylece tüm değişkenler analizde eşit katkı sağlar hâle getirilmiştir.

III. Kümeleme Algoritmalarının Uygulanması ve Performans Değerlendirmesi

1. Gerekli kütüphaneler yüklendikten sonra, *k*-ortalamalar, hiyerarşik, *k*-medoids, *k*-medyan, kırpılmış *k*-ortalamalar kümeleme algoritmaları uygun parametrelerle R paket programında yer alan paketler aracılığıyla sadece karma veri seti üzerinde çalıştırılmış ve veri $k = 3$ kümeye ayrılmıştır.
2. Kümeleme algoritmalarının dayanıklılıkları ve kümeleme başarıları AR, CH, Silhouette, Dunn indeksleri ile elde edilmiştir.
3. Elde edilen indeks değerleri grafiklerle görselleştirilmiştir.

IV. Temel Bileşenler Analizi (TBA) ile Görselleştirme ve Kümeleme Sonuçlarının Sunumu

1. Yüksek boyutlu verinin küme yapısının görselleştirilmesi TBA uygulanmıştır. İlk iki bileşen (TB1 ve TB2) görsellik için kullanılmıştır.
2. Görselleştirme için TB1, TB2 ve kümeleme etiketlerini içeren bir veri çerçevesi oluşturulmuş, gerçek sınıf etiketleri de bu çerçeveye eklenmiştir.

3. İki boyutlu düzlemde görselleştirilen küme yapılarında; renkler kümeleri, şekiller ise gerçek etiketleri temsil etmiştir. Böylece, kümelerin ayrışma düzeyi ve kümeleme sonuçlarının gerçek sınıflarla olan uyumu görsel olarak değerlendirilebilmiştir.

KÖÇYB veri setlerinde *k*-ortalamalar kümeleme algoritması için gerçekleştirilen simülasyon çalışmasında; örnek çapı (n), değişken sayısı (p), n/p oranları, aykırı gözlem ve karışma oranlarına göre ayrı ayrı hesaplanan AR, CH, Silhoutte, Dunn indeks değerleri için elde edilen sonuçlar, Çizelge 6.1'de ve bu indekslere ait grafikler ise Şekil 6.1- 6.5 ile verilmiştir.

Hiyerarşik, *k*-medoids, *k*-medyan ve kırpılmış *k*-ortalamalar gibi farklı kümeleme algoritmaları; örnek çapı (n), değişken sayısı (p), n/p oranları, aykırı gözlem ve karışma oranlarına göre ayrı ayrı hesaplanan AR, CH, Silhoutte, Dunn indeks değerleri için elde edilen sonuçlar, EK-2'de Çizelge 2.1-2.4'de AR, CH, Silhoutte, Dunn indeks değerlerine ilişin grafikler ise Şekil 2.1-2.22 ile verilmiştir.

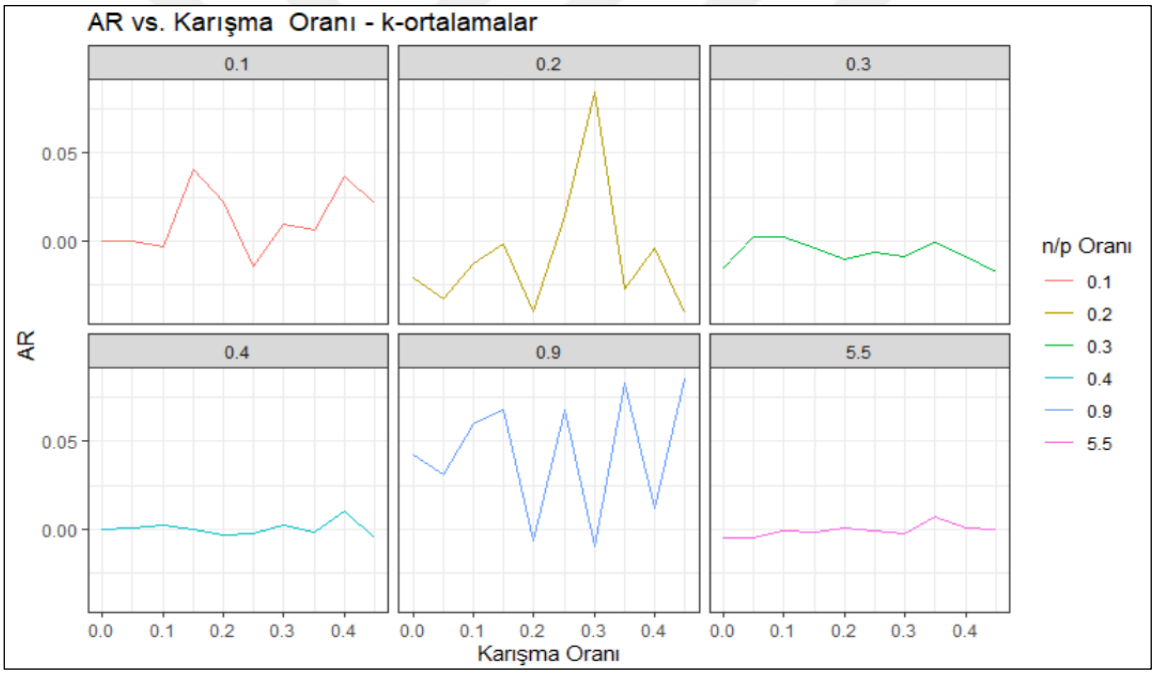
Bu bölümde, simülasyon çalışmasının sonuçları, hem *k*-ortalamalar kümeleme algoritması için hem de ele alınan diğer algoritmlar için burada yorumlanmıştır. Böylece kümeleme sonuçlarının sezgisel değerlendirilmesi sağlanmıştır.

Çizelge 6.1. *k*-ortalamalar kümeleme algoritması için performans değerlendirme indeksleri (AR, CH, Silhouette ve Dunn indeksleri) sonuçları

k-ortalamalar Kümeleme Algoritması												
AR Indeksi, n=20, N(μ, 3²); Aykırı Gözlem =0,20 x n, N(30,15)												
			Karışma Oranları, N(20,10)									
n	p	n/p	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45
20	200	0,1	0,000	0,011	-0,020	-0,034	-0,015	0,022	0,004	-0,056	0,038	-0,071
30	150	0,2	-0,022	0,070	-0,013	0,014	-0,025	0,022	-0,035	-0,001	-0,027	-0,004
75	250	0,3	-0,006	-0,006	-0,014	-0,009	-0,010	0,004	0,011	-0,015	0,001	-0,014
200	500	0,4	0,000	-0,001	-0,002	0,001	-0,003	-0,004	-0,002	-0,006	-0,004	0,026
45	50	0,9	0,016	0,065	0,049	0,027	0,165	0,031	-0,015	-0,026	-0,028	0,001
275	50	5,5	0,004	0,000	0,003	0,000	0,003	-0,001	-0,001	-0,001	-0,004	0,001
CH Indeksi, n=20, N(μ, 3²); Aykırı Gözlem =0,20 x n, N(30,15)												
			Karışma Oranları, N(20,10)									
n	p	n/p	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45
20	200	0,1	0,000	1,814	3,062	5,403	5,268	5,153	6,881	6,805	7,863	8,494
30	150	0,2	1,882	3,916	6,024	7,066	9,004	7,555	11,958	10,071	13,077	11,455
75	250	0,3	4,078	9,921	14,560	19,341	21,629	25,462	26,253	30,331	31,684	32,296
200	500	0,4	0,000	19,125	33,877	48,902	63,417	73,841	82,490	91,040	96,751	98,653
45	50	0,9	2,939	8,614	10,447	12,371	16,199	19,782	16,607	19,759	21,316	19,573
275	50	5,5	5,426	20,493	44,260	65,333	80,267	98,219	106,878	117,201	124,082	128,298

Çizelge 6.1. (devam) k-ortalamalar kümeleme algortiması için performans değerlendirme indeksleri (AR, CH, Silhouette ve Dunn indeksleri) sonuçları

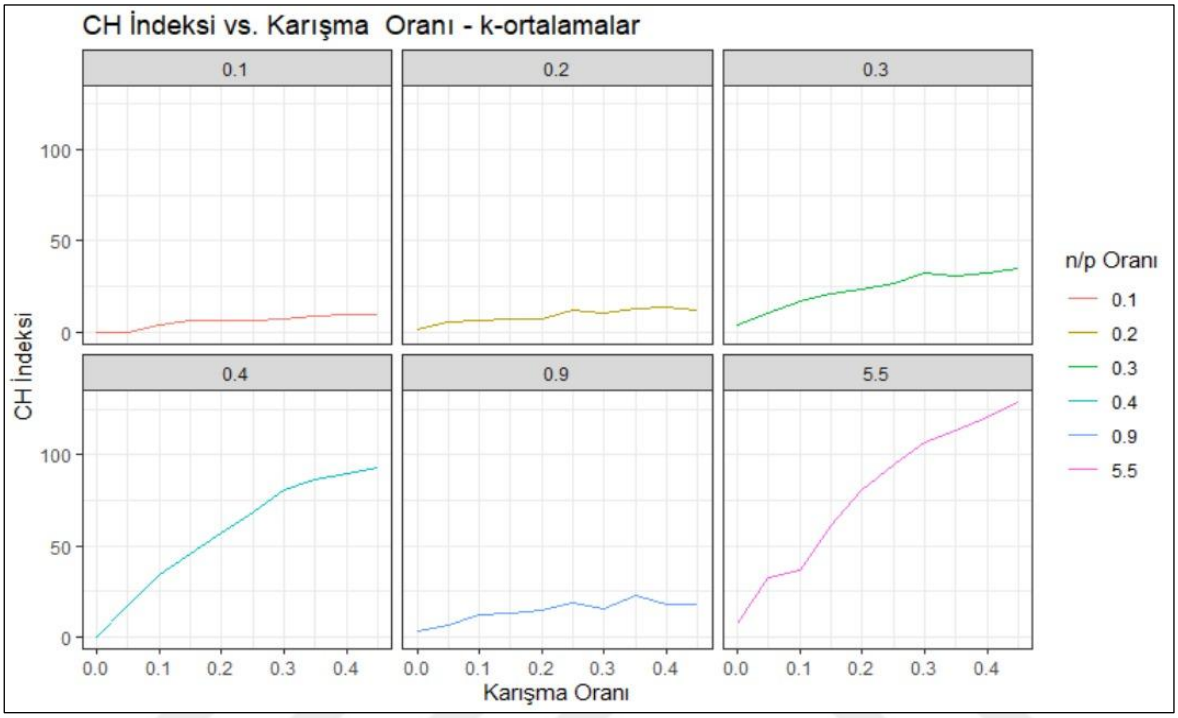
Silhouette Indeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45
20	200	0,1	0,439	0,401	0,354	0,329	0,292	0,312	0,287	0,295	0,289	0,250
30	150	0,2	0,130	0,126	0,339	0,296	0,301	0,324	0,317	0,304	0,327	0,305
75	250	0,3	0,137	0,139	0,141	0,300	0,321	0,305	0,303	0,315	0,313	0,312
200	500	0,4	0,490	0,343	0,340	0,339	0,333	0,333	0,337	0,330	0,328	0,320
45	50	0,9	0,157	0,149	0,156	0,168	0,202	0,218	0,306	0,314	0,321	0,298
275	50	5,5	0,096	0,327	0,325	0,325	0,322	0,323	0,321	0,314	0,321	0,319
Dunn Indeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45
20	200	0,1	0,988	0,935	0,908	0,913	0,801	0,900	0,809	0,858	0,900	0,794
30	150	0,2	0,397	0,396	0,823	0,792	0,780	0,791	0,809	0,784	0,781	0,834
75	250	0,3	0,320	0,349	0,342	0,678	0,711	0,747	0,704	0,673	0,727	0,730
200	500	0,4	0,827	0,796	0,787	0,769	0,764	0,773	0,767	0,773	0,784	0,783
45	50	0,9	0,303	0,304	0,305	0,303	0,313	0,413	0,610	0,629	0,573	0,603
275	50	5,5	0,141	0,480	0,456	0,509	0,519	0,498	0,496	0,510	0,566	0,505



Şekil 6.1. Simülasyon senaryolarında *k*-ortalamalar yöntemine ilişkin AR skorlarının karışma ve n/p oranlarına göre görselleştirilmesi

Şekil 6.1 *k*-ortalamalar kümeleme yöntemi için her panelde sabit bir n/p oranı için farklı karışma oranlarında, tahmin edilen küme etiketlerinin, gerçek etiketlerle ne kadar örtüştüğünü ölçen, AR değerinin değişimi gösterilmektedir. Tüm panellerde AR değerleri 0.05'in altında seyretmektedir. Bu durum, farklı karışma oranları ve KÖÇYB veriler için *k*-ortalamalar yönteminin kümeleri doğru ayıramadığını ve genellikle başarısız olduğunu göstermektedir. Bu yöntem için, $n/p = 0{,}2\ ve\ 0{,}9$ oranlarında AR değerlerinin diğer n/p

oranlarına kıyasla daha dalgalı olduğu ve bu oranlar için istikrarsız çalıştığını, bazı karışma oranlarında (0,2 n/p oranı için 0,30 karışma oranı) şansa bağlı olarak daha iyi sonuçlar verdiği söylenebilir.

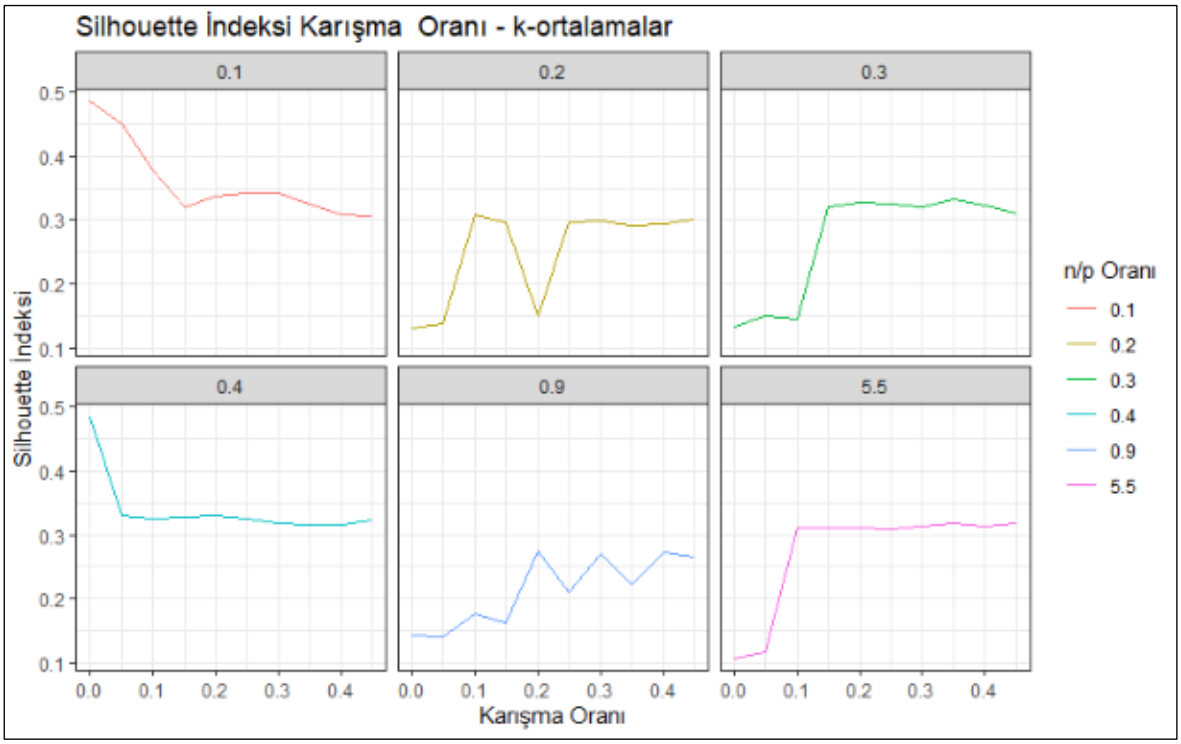


Şekil 6.2. Simülasyon senaryolarında k-ortalamalar yöntemine ilişkin CH skorlarının karışma ve n/p oranlarına göre görselleştirilmesi

Şekil 6.2'de, farklı n/p oranları ve karışma oranları altında k-ortalamalar yöntemi ile elde edilen kümelere ilişkin CH indeks değerleri görselleştirilmiştir. CH indeksi, kümeler arası ayrışmayı kümeler içi sıkılıkla karşılaştırarak kümeleme yapısının iç geçerliliğini değerlendiren bir ölçüttür.

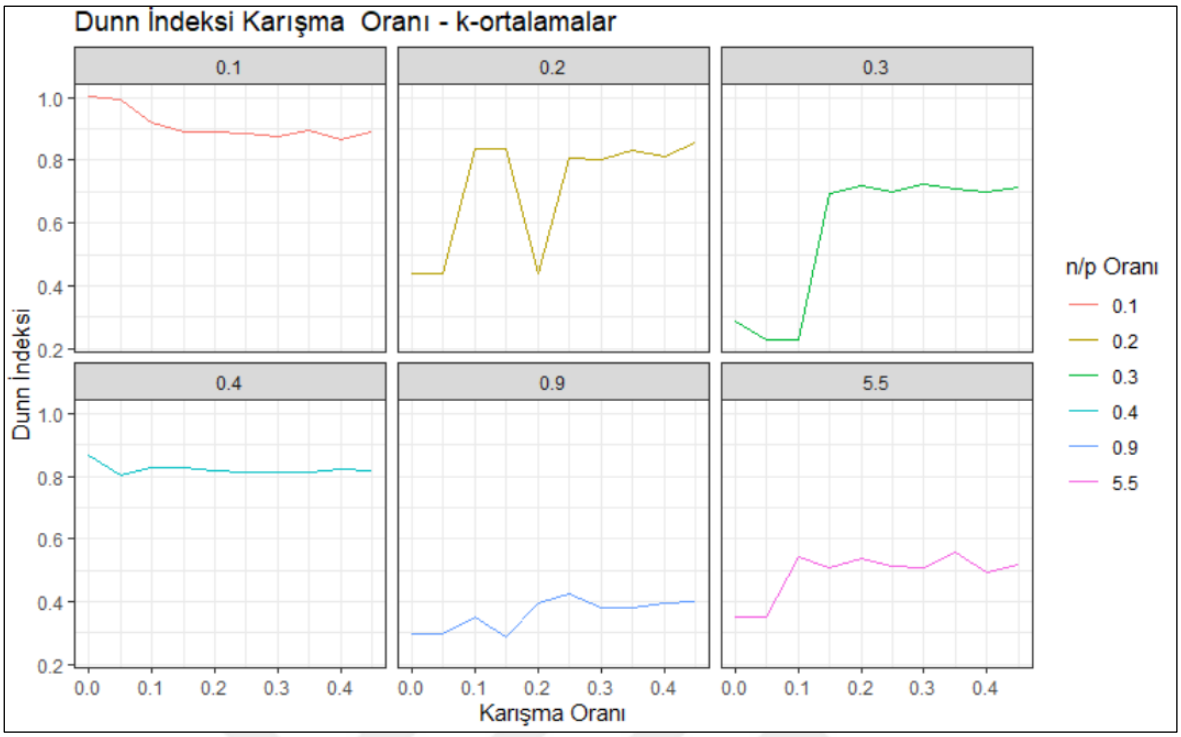
n/p oranı arttıkça CH indeksinde belirgin bir yükselme gözlemlenmektedir. Düşük n/p oranlarında CH değerleri genel olarak düşük seviyelerde kalmakta; karışma oranı artsa bile anlamlı bir artış görülmemektedir. Bu durum, örnek çapının az olduğu durumlarda k-ortalamalar yönteminin kümeler arası ayrımı yeterince başarılı şekilde gerçekleştiremediğini göstermektedir. Buna karşılık, n/p oranı arttıkça CH indekslerinde istikrarlı ve daha belirgin bir artış meydana gelmektedir. Özellikle yüksek n/p oranlarında, karışma oranı arttıkça CH indeksinin neredeyse doğrusal biçimde yükseldiği gözlemlenmektedir. Bu eğilim, örnek sayısının artmasının yöntemin daha ayrışmış kümeler oluşturmasına katkı sağladığını göstermektedir.

Burada dikkat edilmesi gereken önemli bir nokta, CH indeksinin yalnızca iç geçerliliği ölçen bir kriter olmasıdır. CH değerindeki bu artış, kümelerin gerçekte daha anlamlı veya doğru şekilde ayrıldığını garanti etmemektedir. Bu nedenle, yapılan değerlendirmelerin dışsal doğrulama ölçütleriyle desteklenmesi ve diğer içsel değerlendirme indeksleriyle birlikte yorumlanması, yöntemin genel başarısını anlamada kritik öneme sahiptir (Jain ve Dubes, 1988).



Şekil 6.3. Simülasyon senaryolarında k-ortalamalar yöntemine ilişkin Silhouette skorlarının karışma ve n/p oranlarına göre görselleştirilmesi

Karışma oranı arttıkça, gözlemlerin kendi kümesine olan yakınlığı ile diğer kümelere olan uzaklık farkını temel alan, Silhouette indeksinde düşüş eğilimi görülmektedir. Bu durum, veri setine karıştırılan (yani küme yapısını bozan) gözlemlerin kümeleme kalitesini olumsuz etkilediğini göstermektedir. Karışma oranı çok düşükken, algoritma daha başarılı sonuçlar vermekte; ancak belirli bir eşiği geçtikten sonra küme ayrımı zorlaşmakta ve performans azalmaktadır. Bazı durumlarda ani sıçramalar veya dalgalanmalar olsa da genel eğilim, karışma oranı arttıkça performansın azaldığı yönündedir. Bu da göstermektedir ki, k-ortalamalar algoritması, veri kümesine karışma oranı arttıkça oldukça hassas davranmakta ve özellikle yüksek karışma oranlarında güvenilirliğini kaybedebilmektedir.



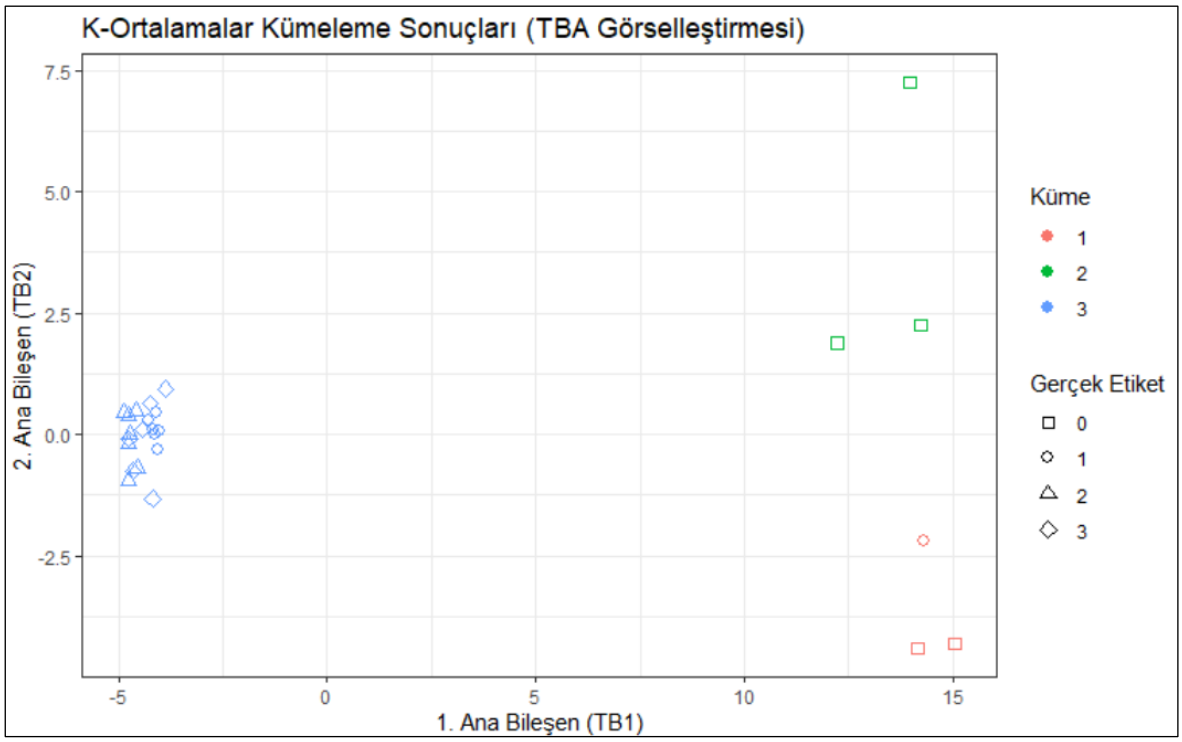
Şekil 6.4. Simülasyon senaryolarında k-ortalamalar yöntemine ilişkin Dunn indekslerinin karışma ve n/p oranlarına göre görselleştirilmesi

Dunn indeksi, kümeler arası uzaklık ile kümeler içi yoğunluğu birlikte dikkate alarak, kümeleme kalitesini değerlendiren önemli bir indekstir. Bu bağlamda, k-ortalamalar algoritmasının farklı karışma ve n/p oranları altındaki performansı incelendiğinde, genel bazı eğilimler ortaya çıkmaktadır.

Genel olarak, n/p oranı düşük olduğunda (örneğin 0,1; 0,2 ve 0,3), Dunn indeksi değerleri yüksek seyretmektedir. Bu durum, k-ortalamalar algoritmasının düşük örnek sayısına karşın yüksek boyutluluk içeren verilerde başarılı bir kümeleme performansı sergileyebildiğini göstermektedir. Özellikle bu oranlarda elde edilen Dunn skorları 0.8'in üzerindedir ve bu da kümeler arası ayrımın güçlü, kümeler içi benzerliğin yüksek olduğuna işaret etmektedir.

Orta düzeydeki n/p oranlarında (0,4 civarında) Dunn indekslerinde hafif bir azalma gözlemlense de genel olarak kümeleme yapısının bozulmadığı ve karışma oranından çok fazla etkilenmediği görülmektedir. Bu durum k-ortalamaların orta düzey boyut/örnek sayısı dengesinde de makul sonuçlar verebildiğini ortaya koymaktadır. Ancak, n/p oranı yüksek seviyelere ulaştığında (örneğin 0,9 ve 5,5), Dunn indeksi belirgin şekilde düşmektedir. Bu, k-ortalamalar algoritmasının yüksek örnek sayısına karşın boyut sayısının düşük kaldığı

durumlarda kümeler arası ayrım gücünü yitirdiğini ve kümelerin daha fazla iç içe geçtiğini göstermektedir. Bu durum, özellikle karışma oranı arttıkça daha da belirginleşmektedir. Böylece, yüksek n/p oranlarında k-ortalamaların Dunn indeksi açısından zayıf bir performans sergilediği sonucuna varılabilir.



Şekil 6.5. k-ortalamalar kümeleme yönteminin tba düzleminde görselleştirilmesi ve gerçek etiketlerle karşılaştırılması ($n = 20,\ p = 100$ için)

Şekil 6.5'te, k-ortalamalar kümeleme algoritmasının çıktıları TBA yardımıyla iki boyutlu bir düzleme indirgenerek görselleştirilmiştir. Bu görsel, algoritmanın veriyi kümelere ayırma başarısını daha net gözlemlemek ve elde edilen kümelerin, gözlemlerin ait olduğu gerçek sınıf etiketleriyle ne derece örtüştüğünü değerlendirmek amacıyla oluşturulmuştur.

TBA ile elde edilen birinci (TB1) ve ikinci (TB2) ana bileşenler, veri setindeki toplam değişkenliğin büyük bir bölümünü temsil etmekte ve kümeler arasındaki genel yapısal ayrışmayı görsel olarak yansıtmaktadır. Görselleştirme incelendiğinde, k-ortalamalar algoritmasının veriyi üç kümeye ayırdığı ve her bir kümenin farklı renklerle temsil edildiği görülmektedir. Aynı zamanda, her bir gözlemin şekli onun ait olduğu gerçek sınıf etiketini göstermektedir. Böylece, algoritmanın yaptığı tahminle gerçek sınıflar arasındaki uyum kolayca takip edilebilmektedir.

Özellikle TB1 ekseni boyunca kümeler arasında belirgin bir ayrışma olduğu dikkat çekmektedir. Sağ üst bölgede konumlanan az sayıda gözlem (etiket 0 ve 1), farklı kümelerde gruplandırılmış ve bu durum, algoritmanın bu gözlemleri ayırt etmede başarılı olduğunu göstermektedir. Buna karşın, sol alt köşede yoğunlaşan ve mavi renkle gösterilen büyük gözlem grubu hem etiket 2 hem de 3'e ait gözlemleri içermektedir. Bu da söz konusu sınıflar arasındaki yapısal uzaklık nedeniyle algoritmanın bu iki grubu ayırt etmekte zorlandığını ve hatalı eşleştirmeler yaptığını göstermektedir.

Bu analiz sonucunda, *k*-ortalamalar algoritmasının ancak kümeler arasındaki farkların açık, net ve belirgin olduğu, yapısal olarak veride kümeler arasında net sınırların bulunduğu, aykırı gözlem içermeyen ve karışma oranı bakımından seviyesi düşük veri setlerinde etkili çalıştığı anlaşılmaktadır. Ancak KÖÇYB yapısındaki ve özellikle karışma oranının veya n/p oranının yüksek olduğu veri senaryolarında, bu yöntemin performansı önemli ölçüde düşmektedir. Bu bağlamda, *k*-ortalamalar kümeleme algoritmasının her veri türü için genellenebilir veya güvenilir bir kümeleme aracı olarak değerlendirilmesi uygun değildir. Bu nedenle, özellikle aykırı gözlemler ve kontaminasyon içeren veri yapılarında, dayanıklılığı yüksek algoritmaların kullanılması daha doğru bir yaklaşım olacaktır. Ayrıca, kümeleme algoritmalarının başarısını değerlendirmek için sadece içsel doğrulama indekslerine değil, aynı zamanda dışsal doğruluk ölçütlerine de yer verilmesi gerektiği; yani içsel yapı ile sınıf uyumunun birlikte değerlendirilmesinin kritik önemde olduğu sonucuna varılmıştır.

EK-2'de verilmiş olan diğer kümeleme algoritmaları için simülasyon sonuçları aşağıdaki gibi özetlenmektedir.

Hiyerarşik kümemele algoritmasının artan karışma oranına karşı tüm n/p oranlarında performansını koruyamadığı görülmektedir. AR, CH, Silhouette ve Dunn indekslerinde gözlenen tutarlı düşüş, yöntemin kontaminasyona karşı duyarlı olduğunu ve kümeler arası ayrımı kaybettiğini göstermektedir. Bu kümemeleme yöntemi için, Şekil 2.1–2.6'daki grafikler ve TBA görselleştirmesi birlikte değerlendirildiğinde, TBA çıktısında, Küme 1'in oldukça sıkı ve doğru bir şekilde ayrıştığı; buna karşın Küme 2 ve 3'ün daha seyrek ve kısmen karışık bir yapı sergilediği gözlemlenmiştir. Bu durum, algoritmanın daha homojen ve yoğun kümeleri başarıyla ayırabildiğini, ancak dengesiz veya sınırlı örnek içeren

kümelerde hatalı sınıflandırmalar yaptığını göstermektedir. Genel olarak yöntem, özellikle aykırı ve karışma içeren senaryolarda sınıf uyumunu sürdürememektedir.

k-medoids algoritması, elde ettiği düşük AR skorlarıyla, kümeleme sonuçlarının gerçek etiketlerle örtüşmesi bakımından zayıf performans sergilemiştir. CH indeksinde yüksek n/p oranlarında artış görülmüş, bu durum örnek çapı ile boyut sayısının birbirine yakın olduğu durumda kümelerin daha belirgin bir yapı sergilediğini göstermektedir. Silhouette skoru, karışma oranı arttıkça düşmüş; küme içi bütünlüğün ve kümeler arası ayrımın bozulduğunu göstermiştir. Dunn indeksi ise daha dalgalı (rastgele) bir yapı sergilemiş, özellikle %0,2 üzerindeki karışma oranlarında kümeler arası ayrım zayıflamıştır. TBA görselleştirmesi, büyük ve homojen kümelerin doğru sınıflandırıldığını, küçük ve dengesiz gruplarda ise hatalı yerleşimler olduğunu ortaya koymuştur.

k-medyan algoritması, tüm n/p oranlarında düşük AR skorlarıyla, kümelerin gerçek sınıflarla uyumunda zayıf performans göstermiştir. CH indeksi özellikle $n/p = 0{,}4\ ve\ 5{,}5$ oranlarında yüksek değerlere ulaşmış, bu durum kümeler arası ayrımın güçlendiğini göstermiştir. Silhouette skorlarında ise karışma oranı arttıkça dikkat çekici sıçramalar gözlemlenmiş, bu da algoritmanın bazı aykırı gözlemlerin ve karışma durumlarının dışlanarak daha ayrışabilir yapılar oluşturabildiğini düşündürmüştür. Dunn indeksi de zaman zaman yüksek değerler alarak, kümeler arası uzaklığın ve ayrıklığın belirli senaryolarda korunduğunu göstermektedir. TBA çıktısı, özellikle Küme 2'nin sıkı ve başarılı şekilde ayrıldığını; buna karşın diğer kümelerin zayıf yapı sergilediğini ortaya koymuştur.

Kırpılmış *k*-ortalamalar algoritmasının performansı, farklı karışma ve kırpma oranları altında AR, CH, Silhouette ve Dunn indeksleri kullanılarak değerlendirilmiştir. AR skorları genel olarak düşük kalmış ve karışma oranı arttıkça daha da azalmıştır. Bu durum, algoritmanın kümeleme sonuçlarının gerçek etiketlerle örtüşmesinde sınırlı başarı sağladığını göstermektedir. CH indeksi ise özellikle yüksek n/p oranlarında belirgin artışlar göstermiştir; bu da kırpma işleminin, özellikle gürültülü gözlemleri dışlayarak kümeler arası ayrımı güçlendirdiğini düşündürmektedir. Silhouette ve Dunn indeksleri zaman zaman dalgalı sonuçlar vermekle birlikte, yüksek n/p oranlarında bu indekslerde görece daha olumlu değerlere ulaşılmıştır. Bu bulgular, algoritmanın belirli veri koşullarında, özellikle yapısal ayrım açısından daha başarılı olabildiğini, ancak genel doğruluk açısından tutarsızlıklar barındırdığını ortaya koymaktadır. TBA görselleştirmesi, algoritmanın

özellikle büyük ve yoğun kümeleri başarılı şekilde ayırabildiğini ortaya koymuştur. Mavi renkli Küme 3'ün sıkı yapısı, bu algoritmanın bazı sınıflarda yapısal tutarlılığı koruduğunu göstermektedir. Öte yandan, kırpma oranı arttıkça CH skorunda düşüş gözlenmiş, bu da aşırı kırpmanın bilgi kaybına yol açarak ayrım gücünü zayıflatabileceğini düşündürmüştür. Genel olarak, kırpılmış *k*-ortalamalar algoritması klasik yöntemlere göre kontaminasyona karşı daha dayanıklı olmakla birlikte, kırpma oranının dengeli belirlenmesi, performans açısından kritik bir unsur olarak öne çıkmaktadır.

7. SONUÇ VE ÖNERİLER

Bu çalışmada, Küçük Örneklem Çaplı Yüksek Boyutlu (KÖÇYB) veri yapılarında, klasik ve sağlam kümeleme algoritmalarının performansları değerlendirilmiştir. Özellikle aykırı gözlemler ve başka dağılımdan gözlem karışması (karışma/kontaminasyon) durumlarının bulunduğu koşullarda, sağlam yöntemlerin klasik algoritmalara kıyasla ne ölçüde daha sağlam/dayanıklı ve başarılı kümeler oluşturduğu performans değerlendirme indeksleri ile incelenmiştir.

İlk olarak, genomik veri setleri üzerinde yapılan uygulamalarda, her iki algoritma türü de Ayarlanmış Rand (AR), Calinski-Harabasz (CH), Silhouette ve Dunn indeksleri kullanılarak karşılaştırılmıştır. Elde edilen sonuçlar, kümeleme algoritmalarının başarısının hem veri setinin yapısına (özellikle n/p oranına) hem de hangi doğrulama metriğine öncelik verildiğine bağlı olarak değiştiğini göstermiştir. Yüksek boyutlu yapılarda k-medyan CH indeksi açısından; k-ortalamalar ve k-medoids ise Silhouette, Dunn indekslerinde öne çıkarken, yapısal tutarlılık ölçütlerinde en başarılı sonuçlar genellikle hiyerarşik algoritmalardan elde edilmiştir. Bu durum, algoritma seçiminde hem veri yapısının hem de değerlendirme ölçütlerinin birlikte dikkate alınması gerektiğini ortaya koymuştur.

Simülasyon çalışmalarında, gerçekçi senaryolar oluşturmak amacıyla başlangıçta normal dağılımdan üretilen yapay veri setlerine aykırı gözlemler ve karışım dağılımları eklenmiştir. Böylece, algoritmalar yalnızca ideal koşullarda değil, aynı zamanda veride aykırı gözlem ve karışma durumlarının bir arada görüldüğü durumlarda da test edilmiştir. Her yöntem, kuramsal varsayımlarına uygun biçimde uygulanmış ve Temel Bileşenler Analizi (TBA) yardımıyla elde edilen küme yapıları görselleştirilmiştir. Bu görselleştirme, sınıf ayrımlarını ve yapısal örüntüleri sezgisel olarak değerlendirme imkânı sunmuştur.

Elde edilen bulgular, kümeleme başarısının aykırı gözlem varlığı, karışma oranı ve n/p oranı gibi veri setine özgü faktörlere oldukça duyarlı olduğunu göstermektedir. Özellikle $n << p$ durumlarında, uzaklık ölçümlerinin güvenilirliği azalmakta, kümeler arasındaki farklar belirginliğini yitirmekte ve bu da klasik yöntemlerin performansını olumsuz etkilemektedir. k-ortalamalar ve hiyerarşik algoritma, yalnızca düşük kontaminasyon ve dengeli n/p oranlarında sınırlı başarı sağlamıştır. Buna karşılık, kırpılmış k-ortalamalar ve k-medyan gibi sağlam yöntemler hem yapısal ayrımı hem de sınıf eşleşmesini daha iyi korumuş ve klasik

algoritmalara göre daha tatmin edici sonuçlar üretmiştir. Ancak bu yöntemlerin başarısı da seçilen parametrelere duyarlıdır; örneğin kırpma oranının fazla belirlenmesi, veri içerisindeki önemli ve yapısal bilgileri taşıyan gözlemlerin de dışlanmasına neden olarak anlamlı bilgilerin kaybına yol açabilirken, az belirlenmesi ise veri içindeki aykırılıkların hala korunmasına yol açar. Her iki durumda da modelin temsil gücü azalır ve kümeleme sonuçlarının doğruluğu olumsuz etkilenir.

Genel olarak, kümeleme performansının veri setinin doğasına bağlı olarak ciddi biçimde değişkenlik gösterdiği sonucuna ulaşılmıştır. Bu nedenle, yalnızca tek bir doğrulama metriğine değil; hem içsel (CH, Silhouette, Dunn) hem de dışsal (AR) ölçütlere başvurulması, daha güvenilir yorumlara ulaşmak açısından önemlidir. Bununla birlikte, yalnızca sınırlı sayıda sağlam algoritmanın değerlendirilmiş olması, bulguların genellenebilirliğini sınırlayabilir. Bu nedenle, yapılacak çalışmalarda DBSCAN gibi sağlam veya yarı-parametrik algoritmaların da benzer veri yapılarında test edilmesi önerilmektedir. Ayrıca, KÖÇYB veri setlerine özgü olarak geliştirilecek algoritmaların, farklı karışma oranları ve aykırı gözlem yapılarına karşı sağlamlıklarının detaylı indelenmesi, bu yöntemlerin pratikteki etkinliğini arttıracaktır. Parametre hassasiyet analizleri ile optimum ayarların belirlenmesi ise genel başarıyı yükseltebilir.

Bu çalışma, yalnızca teorik değil, uygulamalı analizlerle yöntemlerin avantajlarını, sınırlılıklarını ve veri bozulmalarına karşı dirençlerini karşılaştırmalı biçimde ortaya koyarak, KÖÇYB veri analizlerinde sağlam kümeleme algoritmalarının önemini ortaya koymuştur. Ayrıca kullanılan algoritmaların paket ve fonksiyon detaylarının paylaşılmasıyla uygulama süreci şeffaf biçimde sunulmuştur. Sunulan sınırlılıklar ve öneriler doğrultusunda, çalışmanın literatüre anlamlı bir katkı sağlaması beklenmektedir.

KAYNAKLAR

Aggarwal, C. C., and Reddy, C. K. (Eds.). (2013). *Data clustering: Algorithms and applications*. Boca Raton: Chapman and Hall/CRC, 12-35.

Ahn, J., Lee, M. H., and Lee, J. A. (2018). Distance-based outlier detection for high dimension, low sample size data. *Journal of Applied Statistics*, 46(1), 13–29.

Ahn, J., Marron, J. S., Müller, K. M., and Chi, Y.-Y. (2007). The high dimension, low sample size geometric representation holds under mild conditions. *Biometrika*, 94(3), 760–766.

Ali, I., Rehman, A. U., Khan, D. M., Khan, Z., Shafiq, M., and Choi, J.-G. (2022). Model selection using K-means clustering algorithm for the symmetrical segmentation of remote sensing datasets. *Symmetry*, 14(6), 1149.

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A. and Staudt, L. M. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769), 503–511.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 96(12), 6745–6750.

Alpar, R. (2021). *Uygulamalı çok değişkenli istatistiksel yöntemler* (6. Baskı). Ankara: Detay Yayıncılık, 78-85.

Amaratunga, D., and Cabrera, J. (2016). High dimensional data. *Journal of the National Science Foundation of Sri Lanka*, 44(1), 3–9.

Andrews, D. F., and Herzberg, A. M. (1985). *Data: A collection of problems from many fields for the student and research worker* (Ch. 36). New York: Springer-Verlag, 45-52.

Bellas, A., Bouveyron, C., Cottrell, M., and Lacaille, J. (2012). *Robust clustering of high-dimensional data.* In Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, 329-334.

Bouveyron, C., and Brunet-Saumard, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis*, 71, 52–78.

Box, G. E. P. (1953). Non-normality and tests on variances. *Biometrika*, 40(3/4), 318–335.

Brodinová, Š., Gijbels, I., Kalina, J., and Fišerová, E. (2019). Robust clustering methods: A review. *WIREs Computational Statistics*, 11(5), e1462.

Bühlmann, P., and van de Geer, S. (2011). *Statistics for high-dimensional data: Methods, theory and applications*. New York: Springer, 66-77.

Caliński, T., and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics – Theory and Methods*, 3(1), 1–27.

Casella, G., and Berger, R. L. (2002). *Statistical inference* (2nd ed.). Pacific Grove: Duxbury Press, 45-55.

Chen, R., Lin, L., and Huang, X. (2024). A study of changes in college English online phrases supported by clustering algorithm. *Applied Mathematics and Nonlinear Sciences*, 9(1), 122-136.

Croux, C., Filzmoser, P., and Fritz, H. (2011). Robust sparse principal component analysis (Working Paper No. 1113). Catholic University of Leuven, Department of Decision Science and Information Management. *SSRN*, 36.

Cuesta-Albertos, J. A., Gordaliza, A., and Matrán, C. (1997). Trimmed K-means: An attempt to robustify quantizers. *The Annals of Statistics*, 25(2), 553–576.

Cuesta-Albertos, J., Gordaliza, A., and Matrán, C. (1997). Trimmed K-means: An attempt to robustify quantizers. *The Annals of Statistics*, 25(2), 553–576.

Dunn, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3), 32–57.

Ester, M., Kriegel, H. P., Sander, J., and Xu, X. (1996). *A density-based algorithm for discovering clusters in large spatial databases with noise.* In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD), Portland, Oregon, 226–231.

Everitt, B. S. (1977). *Cluster analysis*. London: Heinemann Educational Books, 122-136.

Fan, J., and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5), 849–911.

Fan, J., and Lv, J. (2009). A selective overview of variable selection in high dimensional feature space [Invited review article]. *arXiv*, 4, 78-85.

Fokoué, E., and Titterington, D. M. (2006). Mixtures of factor analysers. *Bayesian Analysis*, 1(1), 179–200.

Friedman, H. P., and Rubin, J. (1967). On some invariant criteria for grouping data. *Journal of the American Statistical Association*, 62(320), 1159–1178.

García-Escudero, L. Á., and Gordaliza, A. (1999). Robustness properties of k means and trimmed k means. *Journal of the American Statistical Association*, 94(447), 956–969.

García-Escudero, L. A., Gordaliza, A., Matrán, C., and Mayo-Iscar, A. (2010). A review of robust clustering methods. *Advances in Data Analysis and Classification*, 4, 89–109.

García-Escudero, L. A., Gordaliza, A., Matrán, C., and Mayo-Iscar, A. (2008). A general trimming approach to robust cluster analysis. *The Annals of Statistics*, 36(3), 1324–1345.

Garima, S. (2021). Clustering algorithms and their taxonomy. *Journal of Data Science Research*, 15(2), 123–138.

Godichon-Baggioni, A., and Surendran, S. (2022). A penalized criterion for selecting the number of clusters for K-medians [Preprint]. *arXiv*, 2, 38-45.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P. and Lander, E. S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439), 531–537.

Gordon, A. D. (1981). *Classification*. London: Chapman and Hall, 22-35.

Guyon, I., von Luxburg, U., and Williamson, R. C. (2012). Clustering: Science or art? In *Proceedings of the ICML 2012 Workshop on Unsupervised and Transfer Learning*. Max Planck Institute for Intelligent Systems, 78-85.

Gündüz, N., and Fokoué, E. (2015). Robust classification of high dimension low sample size data [Preprint]. *arXiv*, 1, 38-48.

Hall, P., Marron, J. S., and Neeman, A. (2005). Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(3), 427–444.

Han, J., Kamber, M., and Pei, J. (2011). *Data mining: Concepts and techniques* (3rd ed.). San Francisco: Morgan Kaufmann, 12-35.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). New York: Springer, 88-95.

Hennig, C., Meila, M., Murtagh, F., and Rocci, R. (Eds.). (2015). *Handbook of cluster analysis* (1st ed.). Boca Raton: Chapman and Hall/CRC, 35-38.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6), 417–441.

Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1), 73–101.

Hubert, L., and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1), 193–218.

Hubert, M., Rousseeuw, P. J., and Vanden Branden, K. (2005). ROBPCA: A new approach to robust principal component analysis. *Technometrics*, 47(1), 64–79.

İnternet: Weeraratne, N., Hunt, L., and Kurz, J. (2024). Challenges of principal component analysis in high-dimensional settings when n < p. Web: https://assets-eu.researchsquare.com/files/rs-4033858/v1_covered_bca5dcbf-27ae-4506-867f-fd89b43d6918.pdf, Son Erişim Tarihi: 22/04/2025.

Jain, A. K., and Dubes, R. C. (1988). *Algorithms for clustering data*. Englewood Cliffs: Prentice-Hall, 85-94.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, 32, 241–254.

Karim, R., Beyan, O., Zappa, A., Costa, I., Rebholz-Schuhmann, D., Cochez, M., and Decker, S. (2021). Deep learning-based clustering approaches for bioinformatics. *Briefings in Bioinformatics*, 22(1), 85-95.

Kaufman, L., and Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. New York: Wiley-Interscience, 65-78.

Kaur, N. K., Kaur, U., and Singh, D. (2014). K-Medoid clustering algorithm – A review. *International Journal of Computer Application and Technology (IJCAT)*, 1(1), 42.

Li, J., Song, S., Zhang, Y., and Zhou, Z. (2016). Robust K-median and K-means clustering algorithms for incomplete data. *Mathematical Problems in Engineering*, 2016, Article ID 4321928.

Liu, H., Motoda, H., and Yu, L. (2002). *Feature selection with selective sampling*. In Proceedings of the Nineteenth International Conference on Machine Learning, Sydney, 395–402.

Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2), 129–137.

MacQueen, J. (1967). *Classification and analysis of multivariate observations*. In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, 281–297.

Maronna, R. A., and Zamar, R. H. (2002). Robust estimates for high-dimensional datasets. *Technometrics*, 44(4), 307–317.

Mishra, S., Sarkar, U., Taraphder, S., Datta, S., Swain, D., Saikhom, R., Panda, S., and Laishram, M. (2017). Principal component analysis. *International Journal of Livestock Research*, 7(5), 1.

Müller, E., Günnemann, S., Assent, I., and Seidl, T. (2008). Evaluating clustering in subspace projections of high dimensional data. *Proceedings of the VLDB Endowment*, 2(1), 1270–1281.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11), 559–572.

Peters, W. J. S. (2023). Comparing clustering methods on (non)dimensionally reduced High Dimensional Low Sample Size data. *Econometrie*, 1, 48-65.

Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E., and Golub, T. R. (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415(6870), 436–442.

Portnoy, S. (1984). Asymptotic behavior of M-estimators of p regression parameters when p^2/n is large. I. Consistency. *The Annals of Statistics*, 12, 1298–1309.

Portnoy, S. (1988). Asymptotic behavior of likelihood methods for exponential families when the number of parameters tends to infinity. *The Annals of Statistics*, 16, 356–366.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336), 846–850.

Rao, C. R. (1973). Mahalanobis era in statistics. *Sankhyā: The Indian Journal of Statistics, Series A*, 35, 12–26.

Reaven, G. M., and Miller, R. G. (1979). An attempt to define the nature of chemical diabetes using a multidimensional analysis. *Diabetologia*, 16, 17–24.

Rocke, D. M., and Woodruff, D. L. (1996). Identification of outliers in multivariate data. *Journal of the American Statistical Association*, 91(435), 1047–1061.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53–65.

Sarkar, S., and Ghosh, A. K. (2020). On perfect clustering of high dimension, low sample size data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(9), 2257–2272.

Shen, D., Shen, H., Zhu, H., and Marron, J. S. (2016). The statistics and mathematics of high dimension low sample size asymptotics. *Statistica Sinica*, 26(4), 1747–1770.

Sokal, R. R., and Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38, 1409–1438.

Steinley, D. (2004). Properties of the Hubert-Arable Adjusted Rand Index. *Psychological Methods*, 9(3), 386–396.

Stephenson, A. J., Smith, A., Kattan, M. W., Satagopan, J., Reuter, V. E., Scardino, P. T., and Gerald, W. L. (2005). Integration of gene expression profiling and clinical variables to predict prostate carcinoma recurrence after radical prostatectomy. *Cancer*, 104(2), 290–298.

Stern, W., and Descoeudres, J.-P. (1977). X-ray fluorescence analysis of Archaic Greek pottery. *Archaeometry*, 19(1), 73–86.

Tan, P. N., Steinbach, M., and Kumar, V. (2006). *Introduction to data mining*. Boston: Pearson Addison Wesley, 86-95.

Terada, Y. (2013). Clustering for high-dimension, low-sample size data using distance vectors [Preprint]. *arXiv*, 1, 22-45.

Tibshirani, R., Walther, G., and Hastie, T. (2002). Estimating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2), 411–423.

Ullmann, T., Hennig, C., and Boulesteix, A.-L. (2021). Validation of cluster analysis results on validation data: A systematic framework. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 11(4), e1444.

Vanden Branden, K., and Hubert, M. (2005). Robust classification in high dimensions based on the SIMCA method. *Chemometrics and Intelligent Laboratory Systems*, 79, 10–21.

Xu, R., and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3), 645–678.

Yang, G., Yang, S., and Li, R. (2020). Feature screening in ultrahigh-dimensional generalized varying-coefficient models. *Statistica Sinica*, 30(2), 1049–1067.

Zerabi, S., and Meshoul, S. (2017). External clustering validation in big data context. In M. Essaaidi and M. Zbakh (Eds.), *Proceedings of the 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*. New York: IEEE, 6.

EKLER

EK-1. Genomik veri setleri üzerinde indekslere göre (AR, CH, Silhouette ve Dunn İndeksleri) Veri Setlerinin Başarı Sıralaması

Çizelge 1.1. *k*-ortalamalar kümeleme algoritmasının AR, CH, Silhouette ve Dunn indekslerine göre veri seti bazlı performansı

Algoritma	Veri	AR	Veri	CH	Veri	Silhouette	Veri	Dunn
k-ortalamalar	Akciğer	0,904	Diyabet	233,076	Diyabet	0,582	Beyin	0,615
	Beyin	0,452	Prostat	63,275	Seramik	0,344	Lösemi	0,548
	Diyabet	0,380	Akciğer	40,197	Prostat	0,337	Akciğer	0,462
	Lenfoma	0,324	Kolon	25,810	Akciğer	0,233	Lenfoma	0,383
	Lösemi	0,186	Lenfoma	20,185	Kolon	0,225	Seramik	0,359
	Seramik	0,083	Seramik	13,623	Lenfoma	0,089	Prostat	0,356
	Prostat	0,058	Lösemi	7,663	Beyin	0,086	Kolon	0,292
	Kolon	-0,0058	Beyin	3,800	Lösemi	0,079	Diyabet	0,054

Çizelge 1.2. Hiyerarşik kümeleme algoritmasının AR, CH, Silhouette ve Dunn indekslerine göre veri seti bazlı performansı

Algoritma	Veri	AR	Veri	CH	Veri	Silhouette	Veri	Dunn
Hiyerarşik	Akciğer	0,757	Diyabet	171,142	Diyabet	0,634	Beyin	0,646
	Diyabet	0,376	Prostat	38,959	Seramik	0,346	Lösemi	0,551
	Lenfoma	0,332	Akciğer	38,496	Prostat	0,276	Akciğer	0,470
	Beyin	0,227	Kolon	21,030	Akciğer	0,214	Lenfoma	0,415
	Lösemi	0,185	Lenfoma	17,786	Kolon	0,208	Seramik	0,413
	Prostat	0,125	Seramik	10,222	Beyin	0,1	Kolon	0,321
	Kolon	-0,0057	Lösemi	6,592	Lenfoma	0,087	Prostat	0,271
	Seramik	-0,0078	Beyin	2,950	Lösemi	0,067	Diyabet	0,156

Çizelge 1.3. *k*-medoids kümeleme algoritmasının AR, CH, Silhouette ve Dunn indekslerine göre veri seti bazlı performansı

Algoritma	Veri	AR	Veri	CH	Veri	Silhouette	Veri	Dunn
k-medoids	Akciğer	0,935	Diyabet	221,115	Diyabet	0,518	Beyin	0,609
	Beyin	0,61	Prostat	62,174	Prostat	0,335	Lösemi	0,465
	Diyabet	0,382	Akciğer	39,963	Seramik	0,325	Akciğer	0,463
	Lenfoma	0,34	Kolon	24,589	Akciğer	0,24	Lenfoma	0,325
	Seramik	0,138	Lenfoma	18,852	Kolon	0,216	Prostat	0,324
	Lösemi	0,135	Seramik	12,791	Lenfoma	0,083	Seramik	0,305
	Prostat	0,105	Lösemi	6,958	Lösemi	0,073	Kolon	0,276
	Kolon	-0,0122	Beyin	3,510	Beyin	0,071	Diyabet	0,047

EK-1. (devam) Genomik veri setleri üzerinde indekslere göre (AR, CH, Silhouette ve Dunn İndeksleri) Veri Setlerinin Başarı Sıralaması

Çizelge 1.4. *k*-medyan kümeleme algoritmasının AR, CH, Silhouette ve Dunn indekslerine göre veri seti bazlı performansı

Algoritma	Veri	AR	Veri	CH	Veri	Silhouette	Veri	Dunn
k-medyan	Beyin	0,639	Diyabet	212,392	Diyabet	0,527	Beyin	0,608
	Lenfoma	0,542	Prostat	63,275	Seramik	0,344	Lösemi	0,495
	Diyabet	0,473	Akciğer	33,856	Prostat	0,337	Lenfoma	0,400
	Lösemi	0,435	Kolon	25,628	Kolon	0,224	Seramik	0,359
	Akciğer	0,338	Lenfoma	19,692	Akciğer	0,101	Prostat	0,356
	Seramik	0,0837	Seramik	13,623	Lenfoma	0,097	Akciğer	0,345
	Prostat	0,058	Lösemi	7,122	Lösemi	0,077	Kolon	0,283
	Kolon	-0,0113	Beyin	3,480	Beyin	0,069	Diyabet	0,047

Çizelge 1.5. Kırpılmış *k*-ortalamalar kümeleme algoritmasının AR, CH, Silhouette ve Dunn indekslerine göre veri seti bazlı performansı

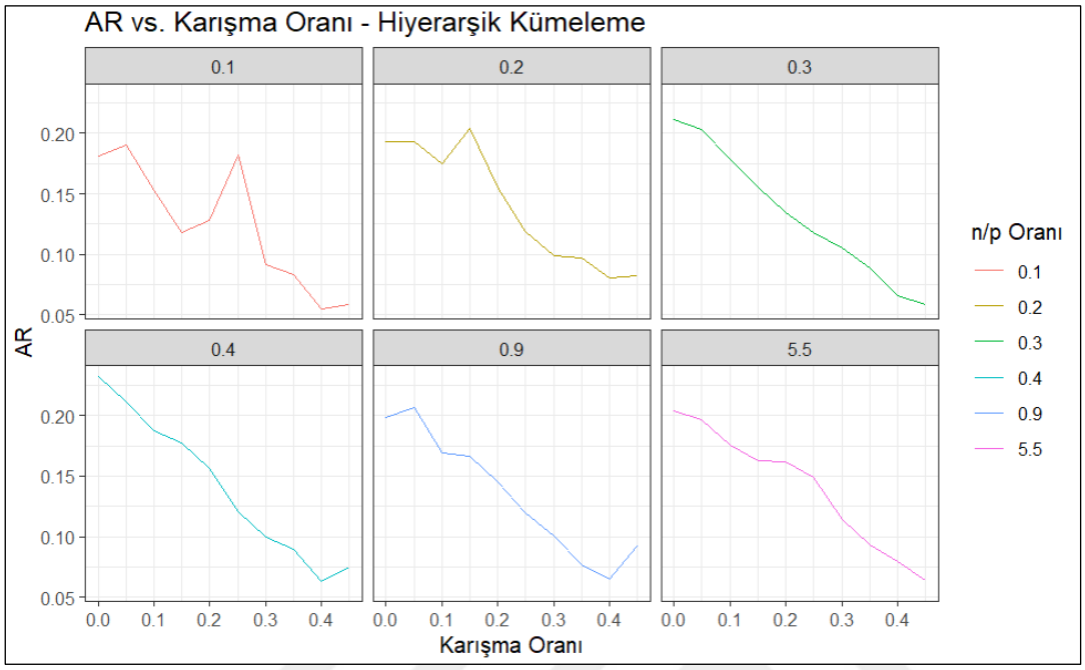
Algoritma	Veri	AR	Veri	CH	Veri	Silhouette	Veri	Dunn
Kırpılmış *k*-ortalamalar	Akciğer	0,736	Diyabet	70,070	Diyabet	0,470	Beyin	0,555
	Beyin	0,445	Akciğer	27,012	Seramik	0,261	Lösemi	0,482
	Diyabet	0,424	Prostat	17,424	Prostat	0,255	Seramik	0,387
	Lenfoma	0,348	Kolon	15,678	Akciğer	0,194	Lenfoma	0,282
	Lösemi	0,133	Lenfoma	13,207	Kolon	0,176	Kolon	0,266
	Seramik	0,054	Seramik	8,830	Beyin	0,073	Akciğer	0,254
	Prostat	0,018	Lösemi	4,548	Lenfoma	0,066	Prostat	0,185
	Kolon	-0,0051	Beyin	3,220	Lösemi	0,063	Diyabet	0,027

EK-2. Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması

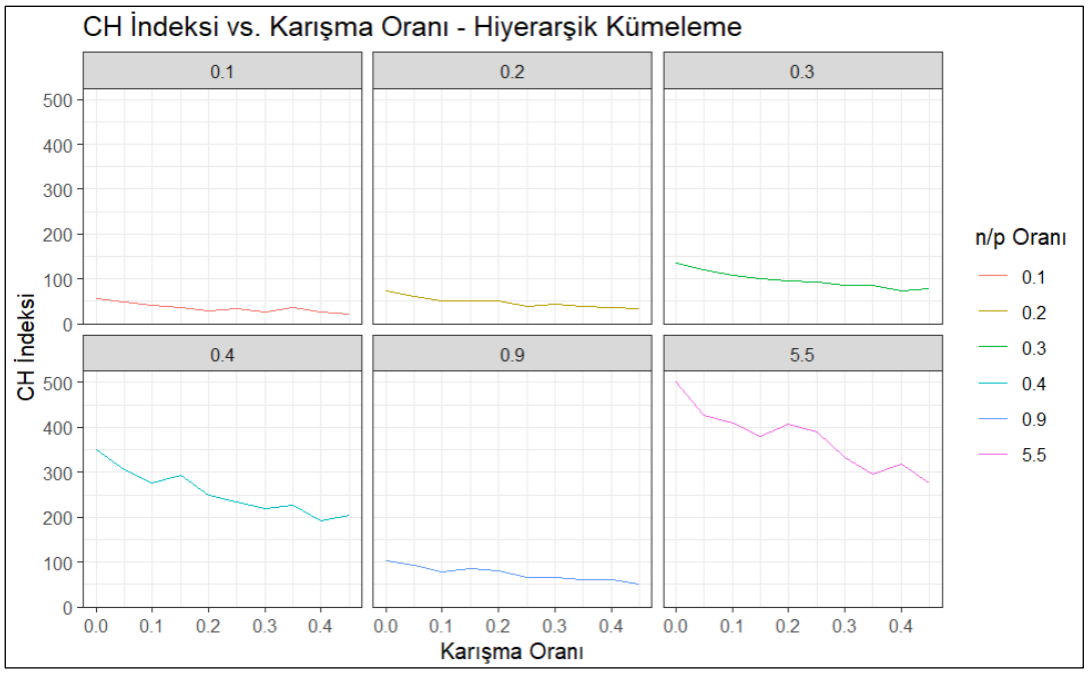
Çizelge 2.1. Hiyerarşik kümeleme algoritması için performans değerlendirme indeksleri (AR, CH, Silhouette ve Dunn indeksleri) sonuçları

Hiyerarşik Kümeleme Algoritması											
AR İndeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,0000	0,0500	0,1000	0,1500	0,2000	0,2500	0,3000	0,3500	0,4000	0,4500
20	200	0,1	0,181	0,19	0,153	0,118	0,128	0,182	0,092	0,083	0,055	0,059
30	150	0,2	0,193	0,193	0,175	0,204	0,156	0,119	0,099	0,097	0,081	0,082
75	250	0,3	0,211	0,203	0,178	0,156	0,135	0,118	0,105	0,089	0,066	0,059
200	500	0,4	0,232	0,211	0,188	0,177	0,156	0,12	0,1	0,089	0,063	0,075
45	50	0,9	0,198	0,207	0,169	0,166	0,146	0,12	0,101	0,077	0,065	0,093
275	50	5,5	0,204	0,197	0,175	0,162	0,162	0,149	0,115	0,093	0,079	0,064
CH Indeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,0000	0,0500	0,1000	0,1500	0,2000	0,2500	0,3000	0,3500	0,4000	0,4500
20	200	0,1	56,08	48,97	40,43	35,81	29,35	32,66	27,23	36,07	27,36	22,6
30	150	0,2	72,3	62,17	51,04	51,77	50,39	39,64	43,97	38,95	36,4	33,99
75	250	0,3	135,9	119,6	108,2	101,1	95,36	92,66	86,47	84,49	72,13	79,18
200	500	0,4	351,4	306	276,6	293,5	248,1	234,9	218,8	227,4	191,2	205,1
45	50	0,9	102,8	93,44	79,08	86,29	81,43	66,52	65,07	60,48	60,14	51,35
275	50	5,5	500,9	426,5	409,6	379,8	407,3	389,2	334	296,5	317,8	275
Silhouette Indeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,0000	0,0500	0,1000	0,1500	0,2000	0,2500	0,3000	0,3500	0,4000	0,4500
20	200	0,1	0,724	0,708	0,644	0,62	0,565	0,576	0,514	0,605	0,517	0,466
30	150	0,2	0,728	0,702	0,618	0,629	0,612	0,556	0,598	0,548	0,528	0,484
75	250	0,3	0,733	0,702	0,66	0,632	0,601	0,582	0,569	0,523	0,519	0,517
200	500	0,4	0,734	0,692	0,653	0,616	0,606	0,566	0,556	0,543	0,516	0,497
45	50	0,9	0,74	0,668	0,648	0,637	0,649	0,61	0,591	0,558	0,527	0,489
275	50	5,5	0,73	0,69	0,653	0,633	0,61	0,588	0,541	0,569	0,524	0,506
Dunn Indeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,0000	0,0500	0,1000	0,1500	0,2000	0,2500	0,3000	0,3500	0,4000	0,4500
20	200	0,1	0,991	0,998	0,917	0,888	0,972	0,888	0,935	1,139	0,938	0,878
30	150	0,2	0,913	0,924	0,772	0,762	0,829	0,902	0,964	0,909	0,903	0,754
75	250	0,3	0,869	0,873	0,867	0,83	0,845	0,83	0,903	0,83	0,899	0,833
200	500	0,4	0,919	0,916	0,914	0,832	0,901	0,859	0,902	0,837	0,894	0,836
45	50	0,9	0,741	0,59	0,665	0,748	0,756	0,753	0,766	0,779	0,699	0,7
275	50	5,5	0,614	0,626	0,563	0,597	0,575	0,602	0,453	0,674	0,548	0,514

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması

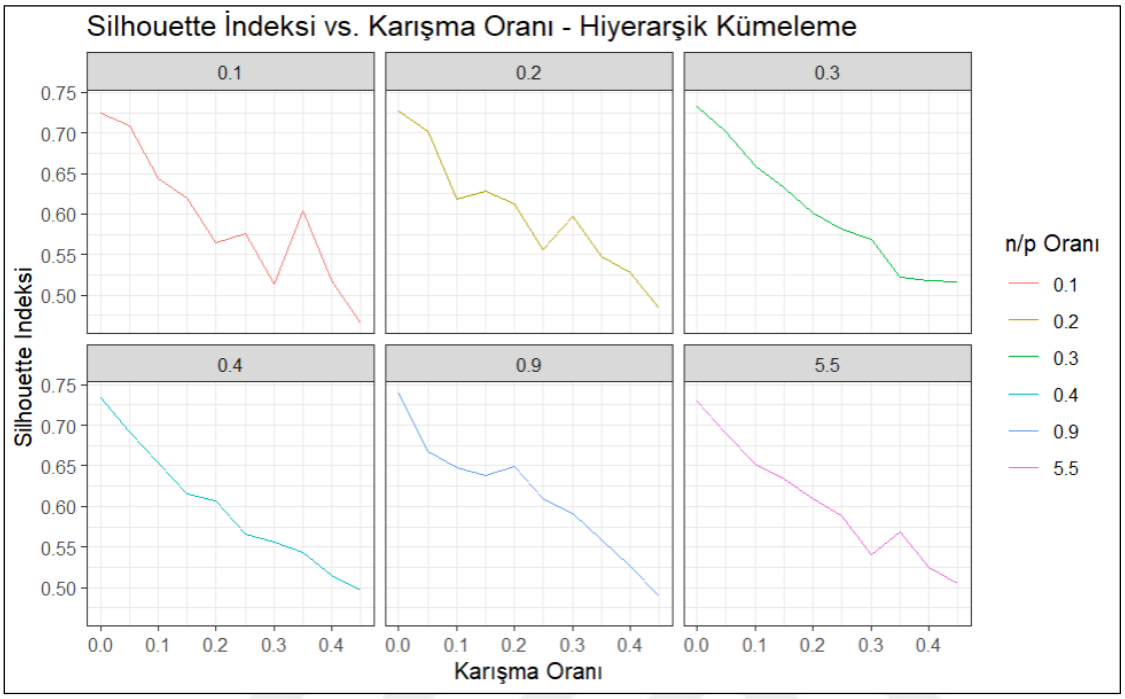


Şekil 2.1. Simülasyon senaryolarında hiyerarşik kümeleme yöntemine ilişkin AR indeksinin karışma ve n/p oranlarına göre görselleştirilmesi

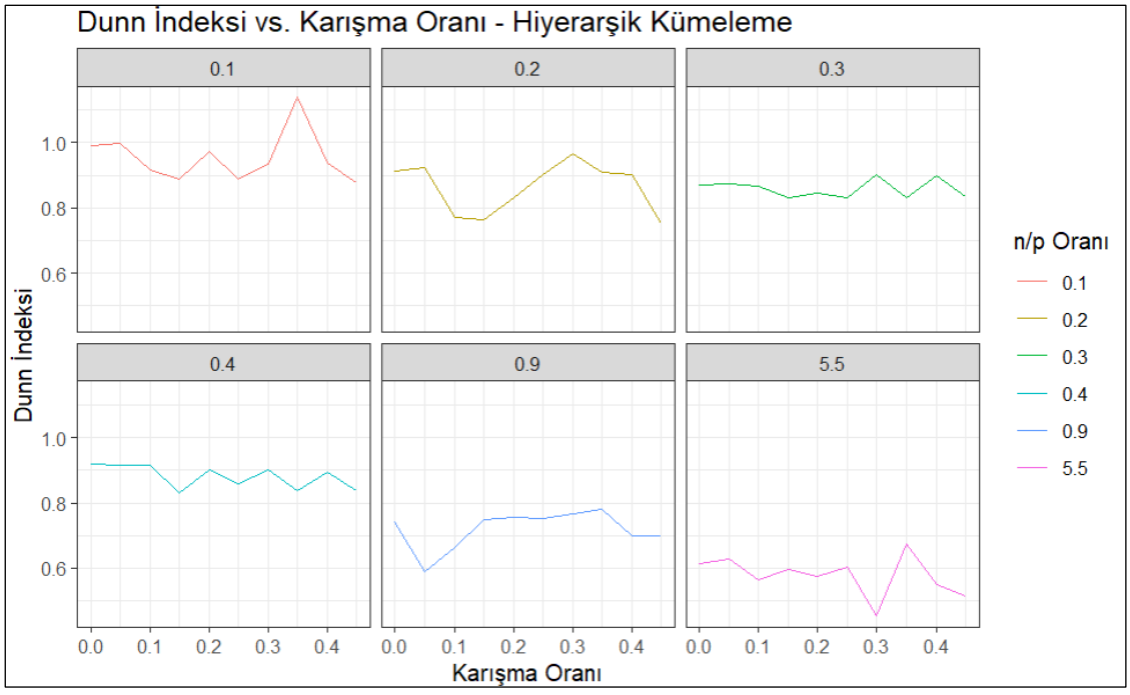


Şekil 2.2. Simülasyon senaryolarında hiyerarşik kümeleme yöntemine ilişkin CH indeksinin karışma ve n/p oranlarına göre görselleştirilmesi

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması

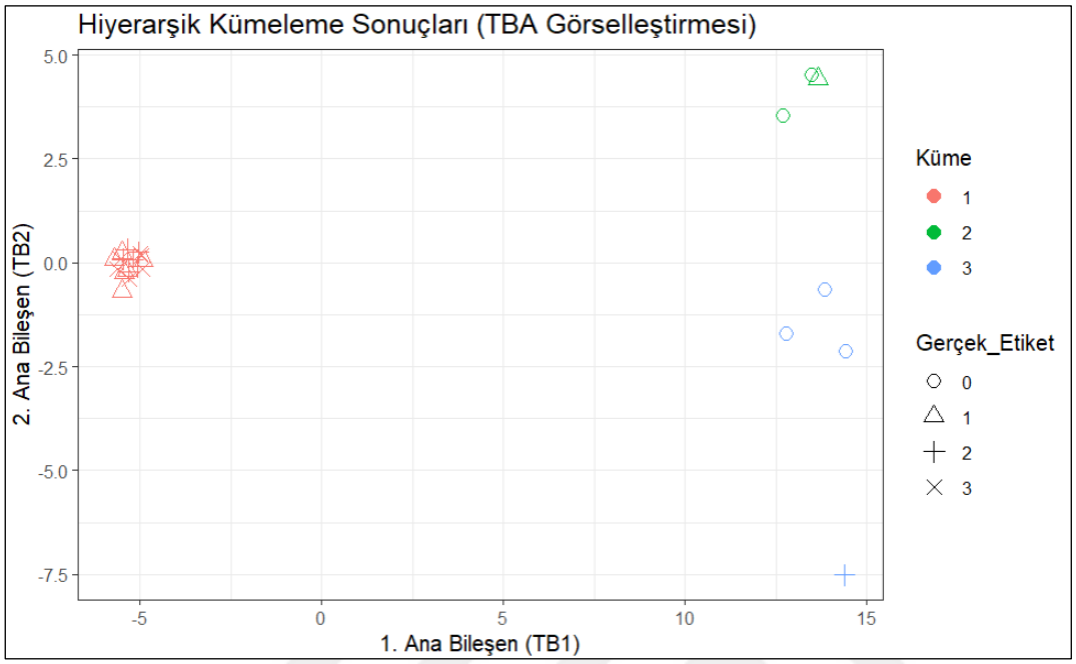


Şekil 2.3. Simülasyon senaryolarında hiyerarşik kümeleme yöntemine ilişkin Silhouette indeksinin karışma ve n/p oranlarına göre görselleştirilmesi

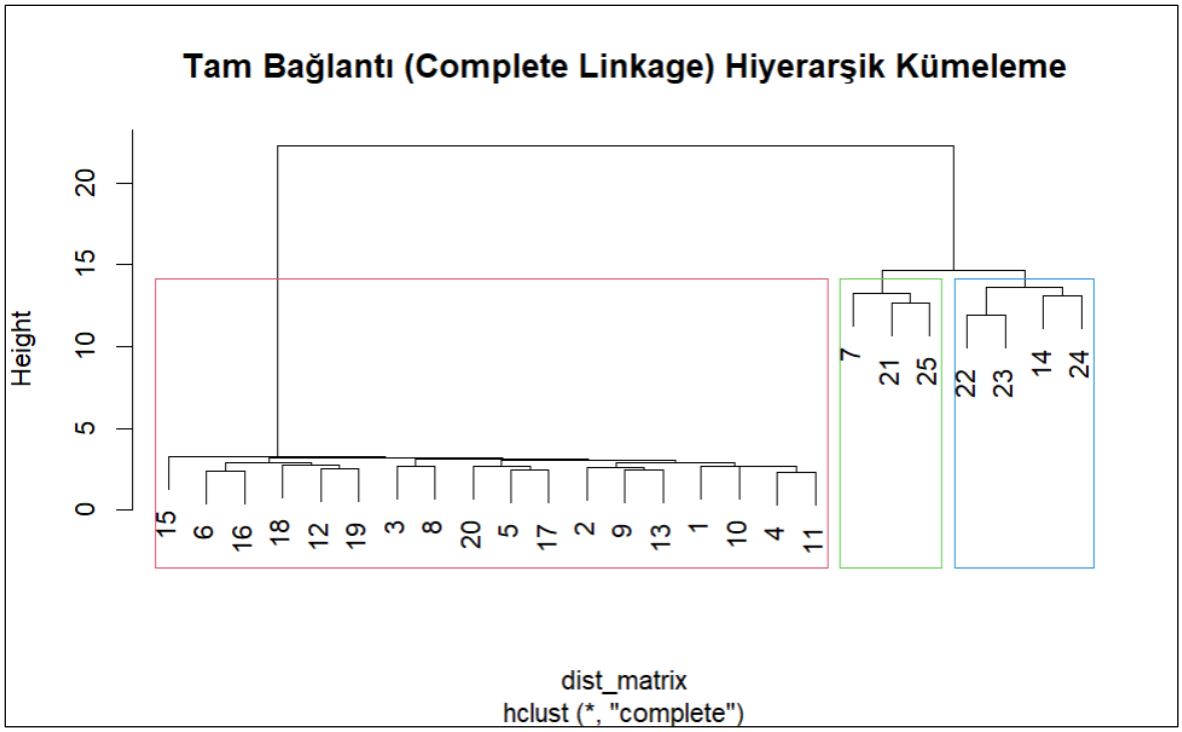


Şekil 2.4. Simülasyon senaryolarında hiyerarşik kümeleme yöntemine ilişkin Dunn indeksinin karışma ve n/p oranlarına göre görselleştirilmesi

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması



Şekil 2.5. Hiyerarşik kümeleme yönteminin TBA düzleminde görselleştirilmesi ve gerçeketiketlerle karşılaştırılması ($n = 20, p = 100$ için)



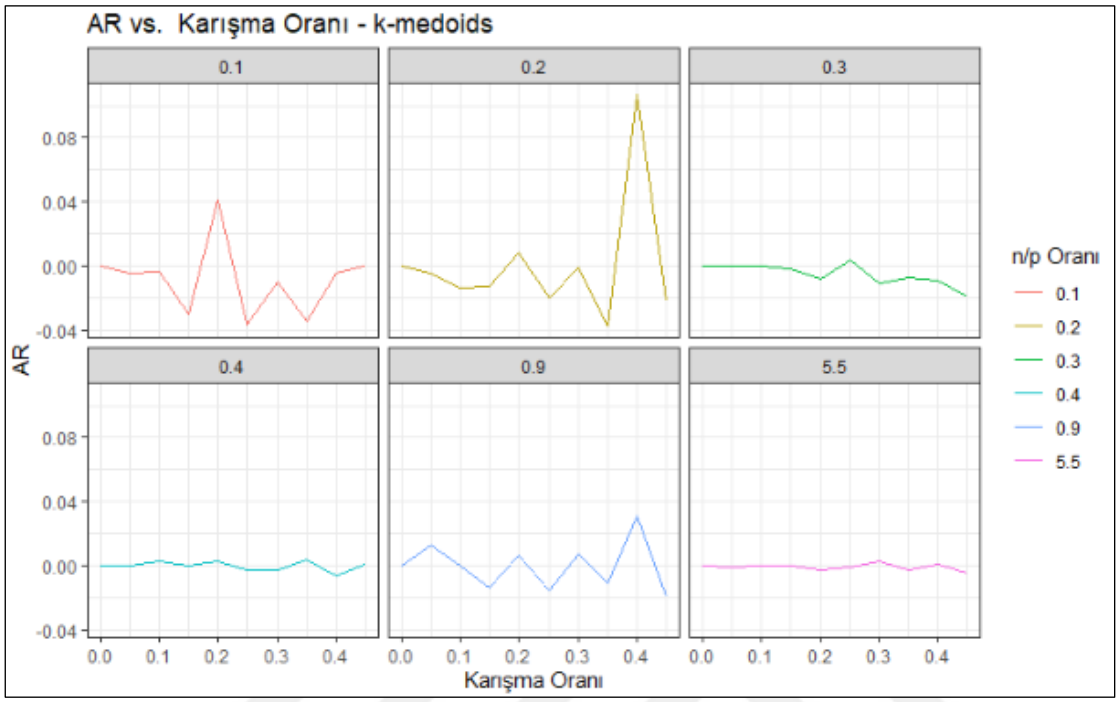
Şekil 2.6. Hiyerarşik kümeleme yöntemine (tam bağlantı) göre dendrogram grafiği (cutree $= k = 3$)

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması

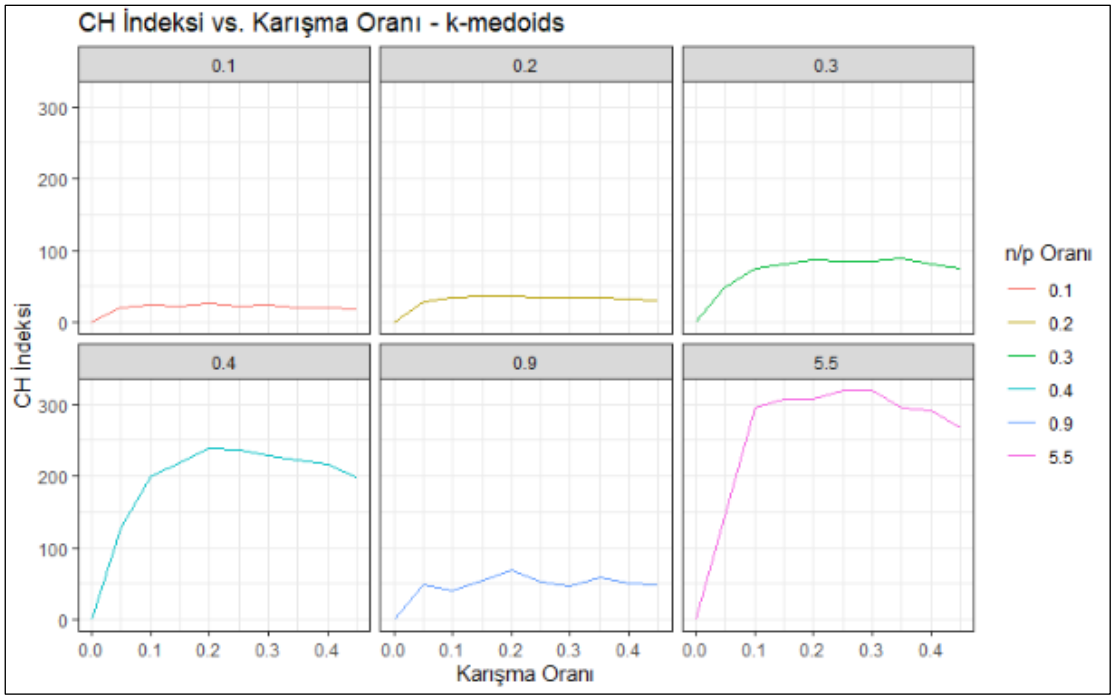
Çizelge 2.2. *k*-medoids kümeleme algoritması için performans değerlendirme indeksleri (AR, CH, Silhouette ve Dunn indeksleri) sonuçları

k-medoids Kümeleme Algoritması											
AR İndeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,000	0,050	0,100	0,150	0,200	0,250	0,300	0,350	0,400	0,450
20	200	0,1	0,000	-0,005	-0,003	-0,030	0,041	-0,036	-0,010	-0,035	-0,004	0,000
30	150	0,2	0,000	-0,004	-0,014	-0,013	0,008	-0,021	-0,001	-0,037	0,107	-0,022
75	250	0,3	0,000	0,000	0,000	-0,002	-0,008	0,004	-0,011	-0,007	-0,009	-0,019
200	500	0,4	0,000	0,000	0,003	0,000	0,003	-0,003	-0,003	0,004	-0,007	0,001
45	50	0,9	0,000	0,013	0,000	-0,013	0,006	-0,015	0,008	-0,011	0,031	-0,019
275	50	5,5	0,000	-0,001	0,000	0,000	-0,002	-0,001	0,002	-0,003	0,001	-0,005
CH İndeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,000	0,050	0,100	0,150	0,200	0,250	0,300	0,350	0,400	0,450
20	200	0,1	0,000	19,827	24,171	22,330	25,692	21,272	24,082	20,724	20,532	17,796
30	150	0,2	0,000	28,484	33,680	36,019	35,890	34,145	33,447	33,824	31,520	30,354
75	250	0,3	0,000	48,888	75,083	81,329	87,343	84,891	85,229	89,413	81,645	74,239
200	500	0,4	0,000	127,812	201,022	218,466	238,485	236,370	229,456	223,204	216,967	198,182
45	50	0,9	0,000	48,220	41,019	54,363	68,652	53,008	47,060	59,715	49,680	49,034
275	50	5,5	0,000	143,926	294,873	307,130	308,149	319,986	319,964	295,047	291,278	266,508
Silhouette İndeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,000	0,050	0,100	0,150	0,200	0,250	0,300	0,350	0,400	0,450
20	200	0,1	0,715	0,664	0,640	0,615	0,597	0,533	0,583	0,519	0,513	0,457
30	150	0,2	0,719	0,629	0,653	0,636	0,561	0,541	0,516	0,533	0,467	0,511
75	250	0,3	0,721	0,673	0,640	0,622	0,598	0,557	0,550	0,449	0,396	0,366
200	500	0,4	0,723	0,685	0,645	0,619	0,560	0,513	0,485	0,439	0,412	0,363
45	50	0,9	0,727	0,677	0,671	0,607	0,538	0,573	0,582	0,473	0,397	0,384
275	50	5,5	0,728	0,676	0,637	0,626	0,597	0,555	0,482	0,445	0,410	0,379
Dunn İndeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,000	0,050	0,100	0,150	0,200	0,250	0,300	0,350	0,400	0,450
20	200	0,1	0,984	0,975	0,938	0,960	0,942	0,935	0,935	0,895	0,873	0,927
30	150	0,2	0,890	0,755	0,889	0,871	0,740	0,773	0,820	0,832	0,821	0,840
75	250	0,3	0,835	0,816	0,842	0,829	0,814	0,809	0,848	0,571	0,537	0,570
200	500	0,4	0,842	0,841	0,814	0,812	0,552	0,541	0,569	0,552	0,543	0,549
45	50	0,9	0,847	0,755	0,776	0,735	0,537	0,716	0,805	0,529	0,504	0,501
275	50	5,5	0,570	0,574	0,528	0,500	0,539	0,510	0,334	0,352	0,326	0,304

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması

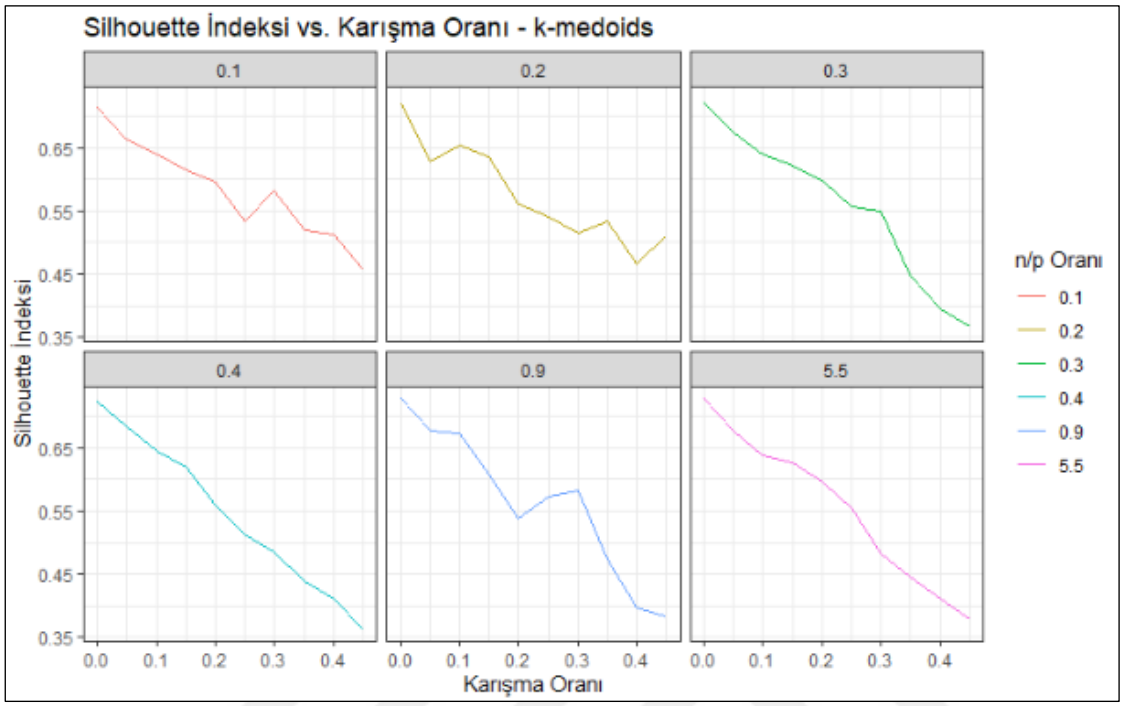


Şekil 2.7. Simülasyon senaryolarında *k*-medoids kümeleme yöntemine ilişkin AR indeksinin karışma ve n/p oranlarına göre görselleştirilmesi

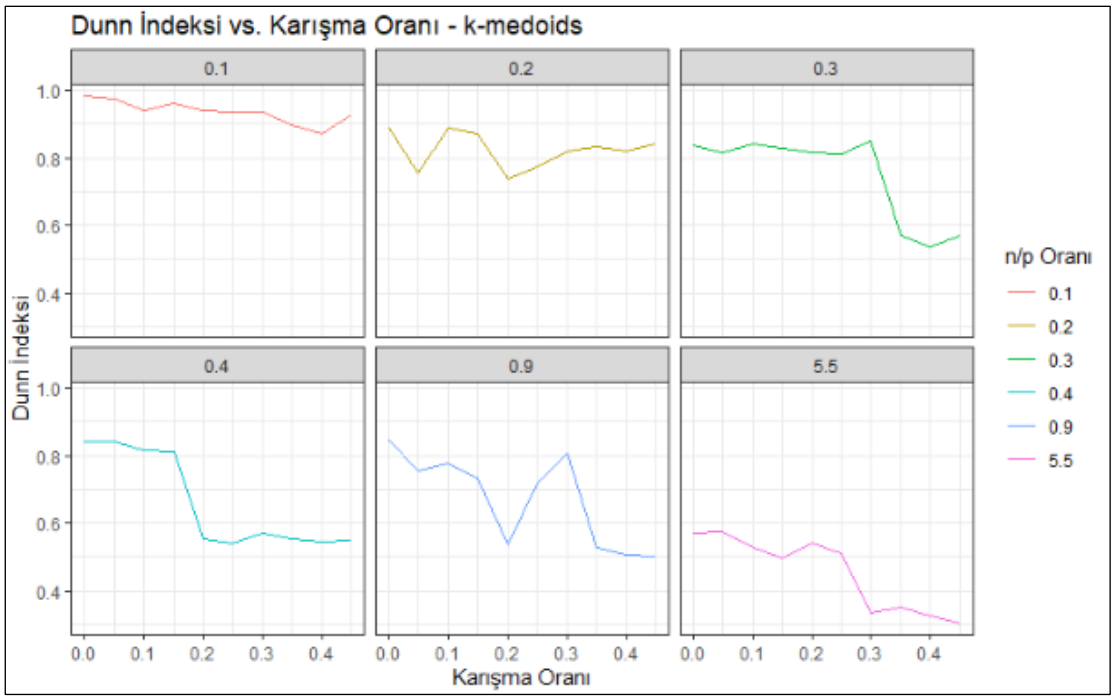


Şekil 2.8. Simülasyon senaryolarında *k*-medoids kümeleme yöntemine ilişkin CH indeksinin karışma ve n/p oranlarına göre görselleştirilmesi

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması

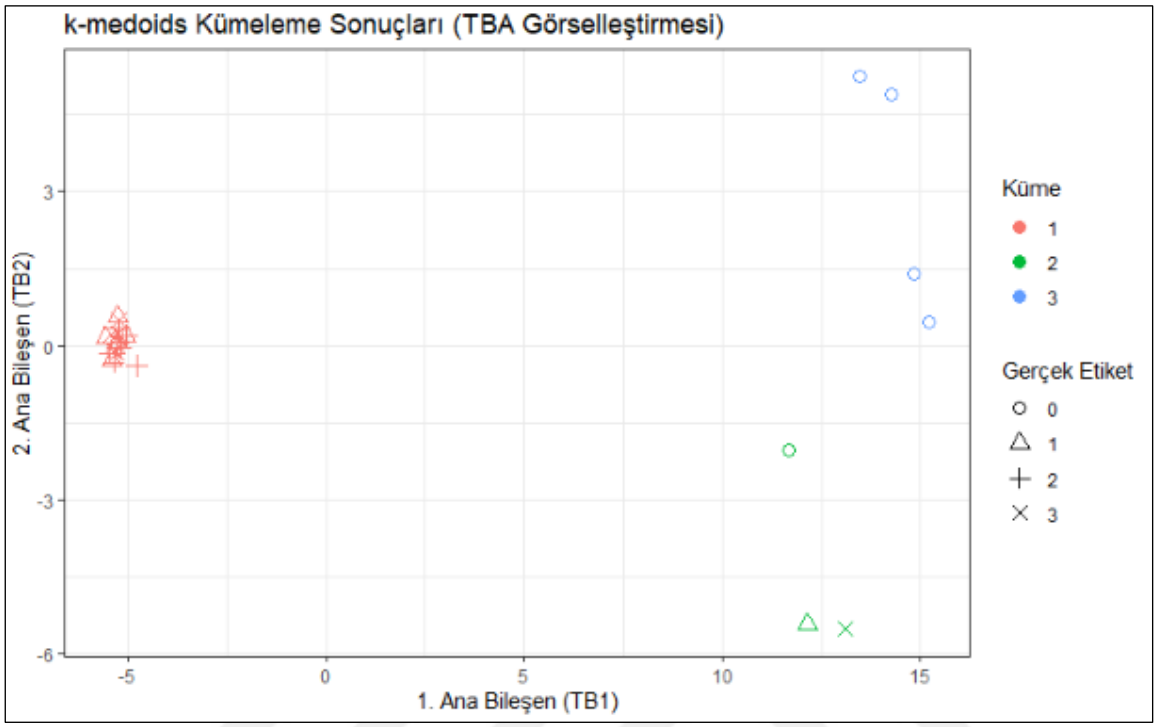


Şekil 2.9. Simülasyon senaryolarında *k*-medoids kümeleme yöntemine ilişkin Silhouette indeksinin karışma ve n/p oranlarına göre görselleştirilmesi



Şekil 2.10. Simülasyon senaryolarında *k*-medoids kümeleme yöntemine ilişkin Dunn indeksinin karışma ve n/p oranlarına göre görselleştirilmesi

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması



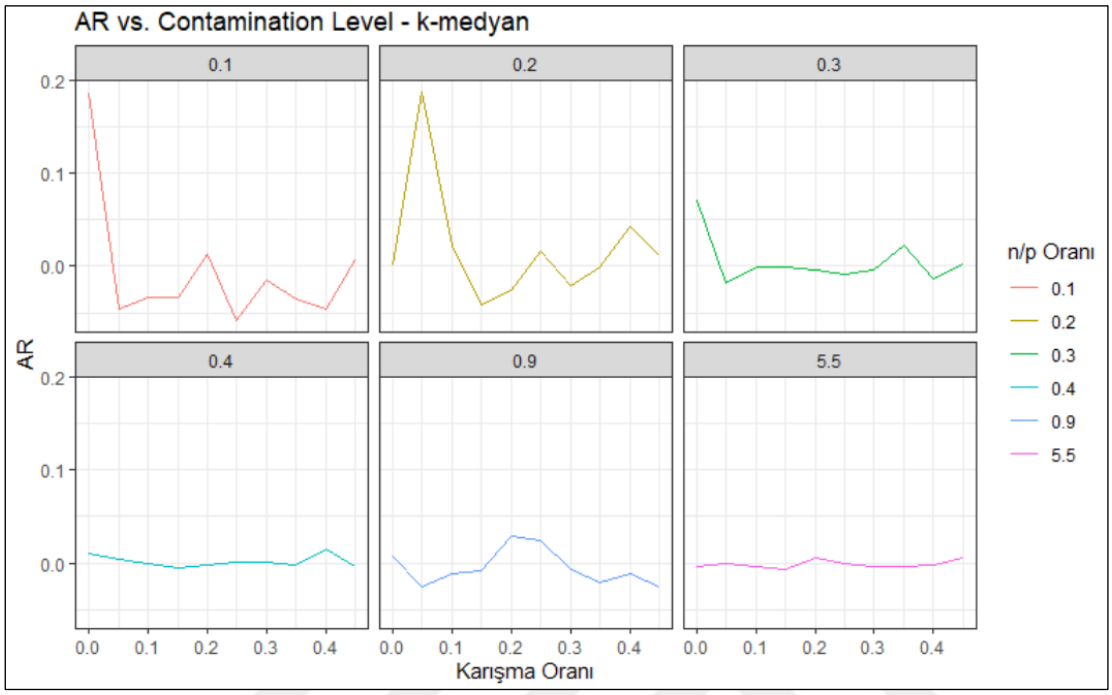
Şekil 2.11. *k*-medoids kümeleme yönteminin TBA düzleminde görselleştirilmesi ve gerçek etiketlerle karşılaştırılması ($n = 20$ $p = 100$için)

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması

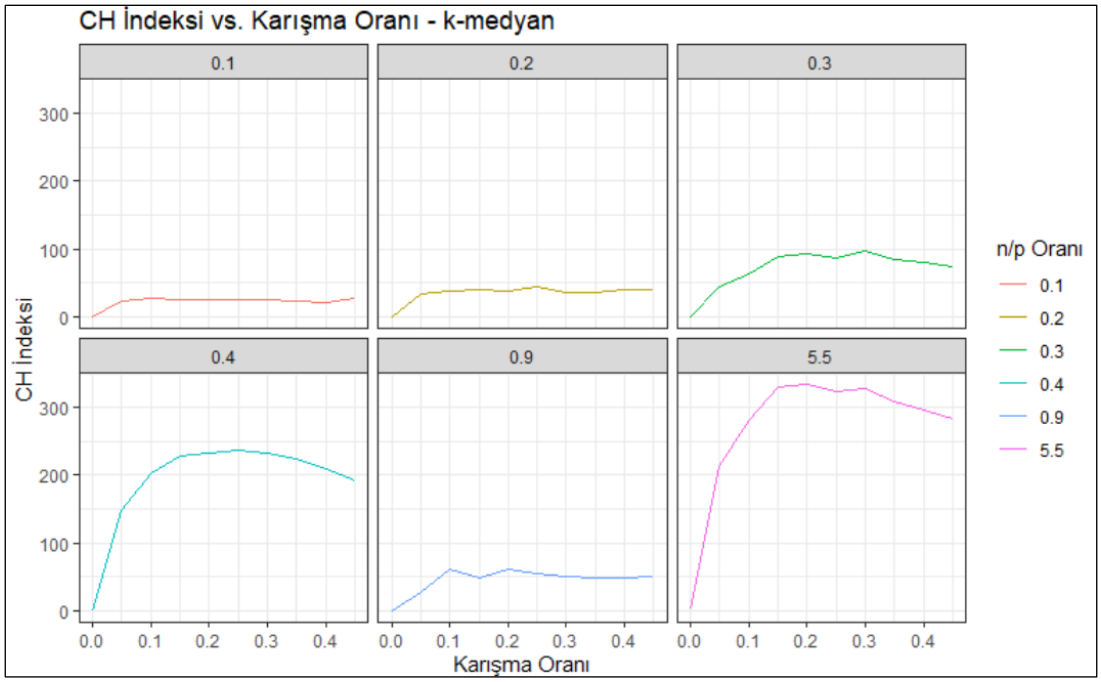
Çizelge 2.3. *k*-medyan kümeleme algoritması için performans değerlendirme indeksleri (AR, CH, Silhouette ve Dunn indeksleri) sonuçları

k-medyan Kümeleme Algoritması											
AR İndeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,000	0,050	0,100	0,150	0,200	0,250	0,300	0,350	0,400	0,450
20	200	0,1	0,186	-0,046	-0,034	-0,034	0,012	-0,058	-0,015	-0,035	-0,047	0,006
30	150	0,2	0,000	0,188	0,023	-0,041	-0,025	0,017	-0,021	-0,001	0,042	0,011
75	250	0,3	0,071	-0,017	0,000	-0,001	-0,004	-0,009	-0,004	0,023	-0,013	0,002
200	500	0,4	0,011	0,005	-0,001	-0,006	-0,002	0,001	0,000	-0,002	0,015	-0,004
45	50	0,9	0,006	-0,025	-0,012	-0,009	0,029	0,024	-0,007	-0,022	-0,012	-0,025
275	50	5,5	-0,004	-0,001	-0,003	-0,006	0,005	0,000	-0,004	-0,004	-0,003	0,006
CH İndeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,000	0,050	0,100	0,150	0,200	0,250	0,300	0,350	0,400	0,450
20	200	0,1	0,658	23,518	26,437	24,575	25,660	25,986	25,420	23,616	21,547	26,791
30	150	0,2	0,000	34,007	37,627	39,222	37,826	44,312	35,425	35,014	39,714	39,733
75	250	0,3	0,673	44,244	62,366	89,543	92,008	87,641	96,341	83,963	80,725	73,763
200	500	0,4	0,660	148,654	202,730	229,057	231,641	236,930	232,637	224,095	208,575	192,279
45	50	0,9	1,078	27,571	61,179	48,920	61,570	54,106	51,909	49,566	48,738	51,881
275	50	5,5	2,462	212,899	280,989	329,812	334,615	322,597	328,155	307,785	295,980	283,831
Silhouette İndeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,000	0,050	0,100	0,150	0,200	0,250	0,300	0,350	0,400	0,450
20	200	0,1	0,092	0,096	0,108	0,598	0,125	0,171	0,631	0,538	0,543	0,449
30	150	0,2	0,736	0,104	0,103	0,120	0,593	0,505	0,574	0,209	0,461	0,442
75	250	0,3	0,076	0,077	0,082	0,122	0,624	0,141	0,478	0,182	0,191	0,531
200	500	0,4	0,076	0,086	0,096	0,112	0,618	0,585	0,483	0,640	0,618	0,514
45	50	0,9	0,100	0,084	0,122	0,124	0,137	0,590	0,586	0,178	0,447	0,411
275	50	5,5	0,089	0,099	0,109	0,128	0,140	0,151	0,582	0,545	0,546	0,503
Dunn İndeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,000	0,050	0,100	0,150	0,200	0,250	0,300	0,350	0,400	0,450
20	200	0,1	0,173	0,175	0,176	0,858	0,170	0,182	1,130	0,886	0,861	0,811
30	150	0,2	0,806	0,166	0,156	0,167	0,828	0,528	0,760	0,161	0,584	0,577
75	250	0,3	0,153	0,153	0,158	0,161	0,801	0,179	0,547	0,155	0,161	0,771
200	500	0,4	0,163	0,163	0,163	0,164	0,811	0,810	0,563	1,018	1,023	0,805
45	50	0,9	0,124	0,125	0,127	0,112	0,136	0,693	0,620	0,118	0,455	0,490
275	50	5,5	0,103	0,101	0,096	0,097	0,099	0,100	0,543	0,394	0,547	0,377

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması

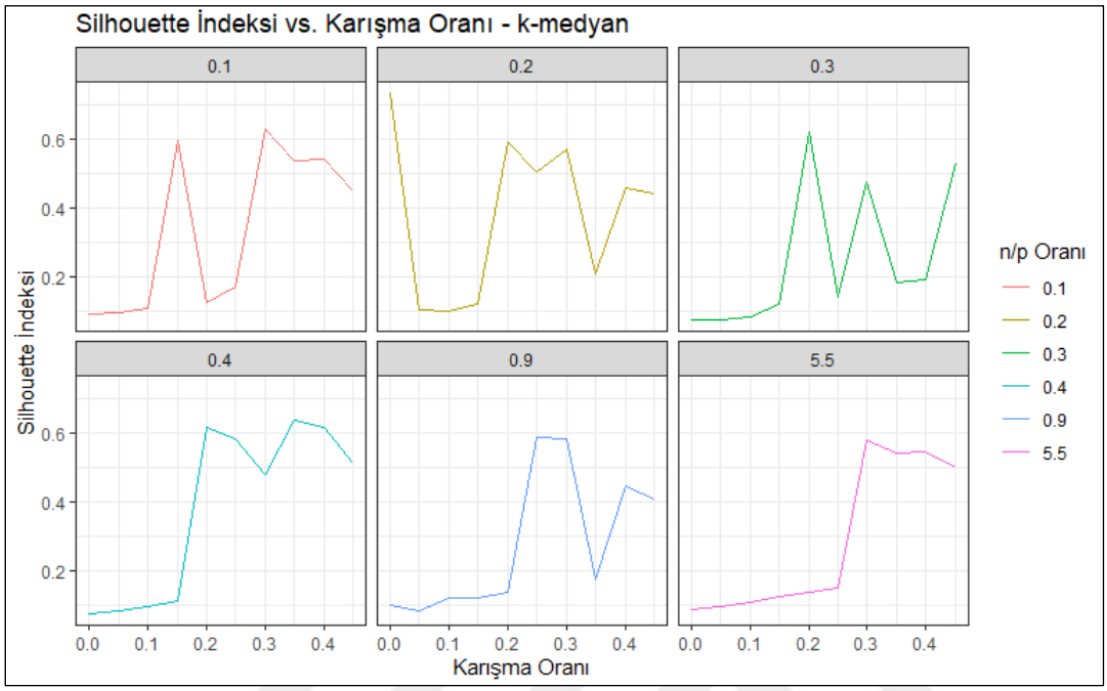


Şekil 2.12. Simülasyon senaryolarında k-medyan kümeleme Yöntemine ilişkin AR indeksinin karışma ve n/p oranlarına göre görselleştirilmesi

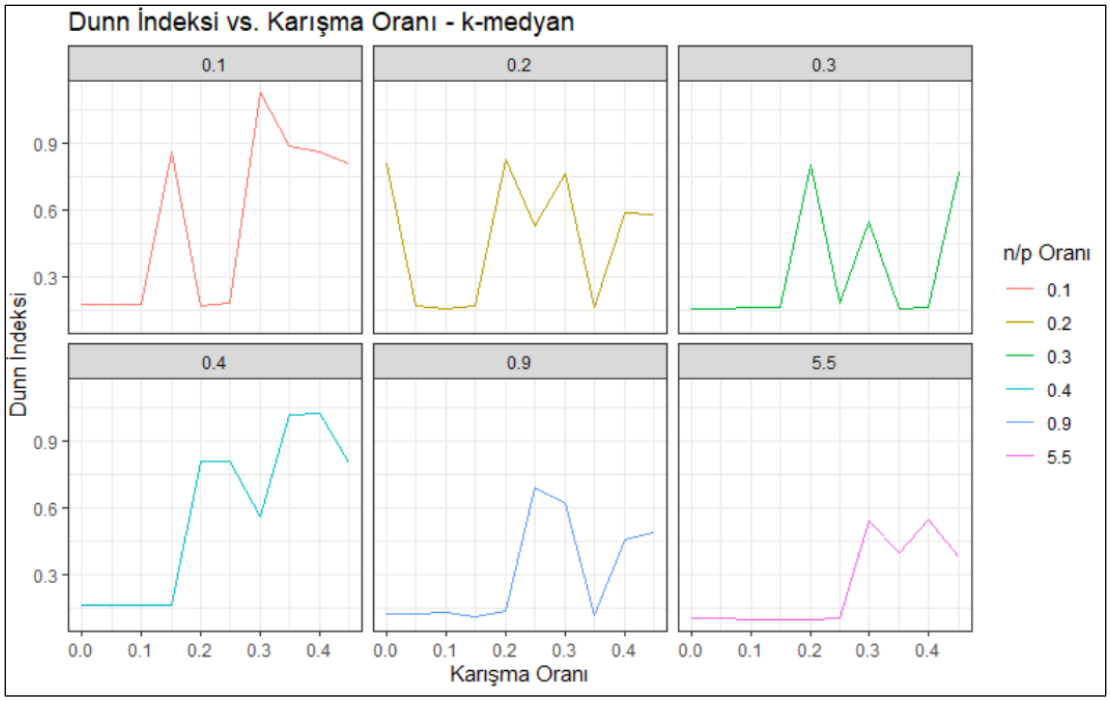


Şekil 2.13. Simülasyon senaryolarında k-medyan kümeleme yöntemine ilişkin CH indeksinin karışma ve n/p oranlarına göre görselleştirilmesi

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması

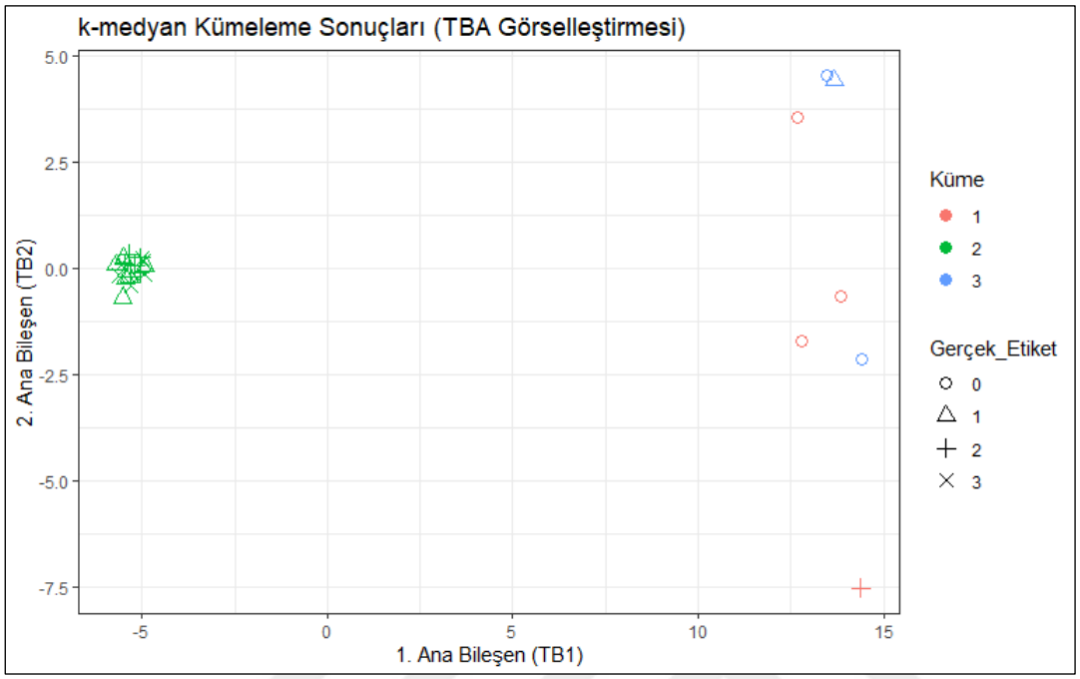


Şekil 2.14. Simülasyon senaryolarında k-medyan kümeleme yöntemine ilişkin Silhouette indeksinin karışma ve n/p oranlarına göre görselleştirilmesi



Şekil 2.15. Simülasyon senaryolarında k-medyan kümeleme yöntemine ilişkin Dunn indeksinin karışma ve n/p oranlarına göre görselleştirilmesi

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması



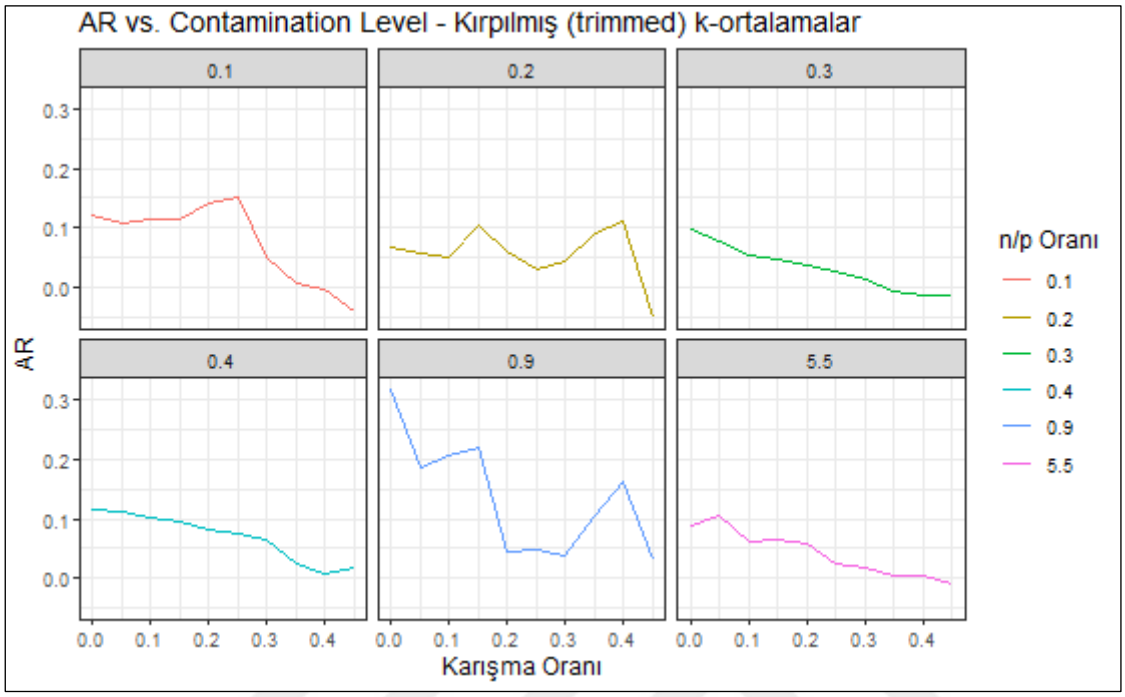
Şekil 2.16. k-medyan kümeleme yönteminin TBA düzleminde görselleştirilmesi ve gerçek etiketlerle karşılaştırılması ($n = 20$ $p = 100$için)

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması

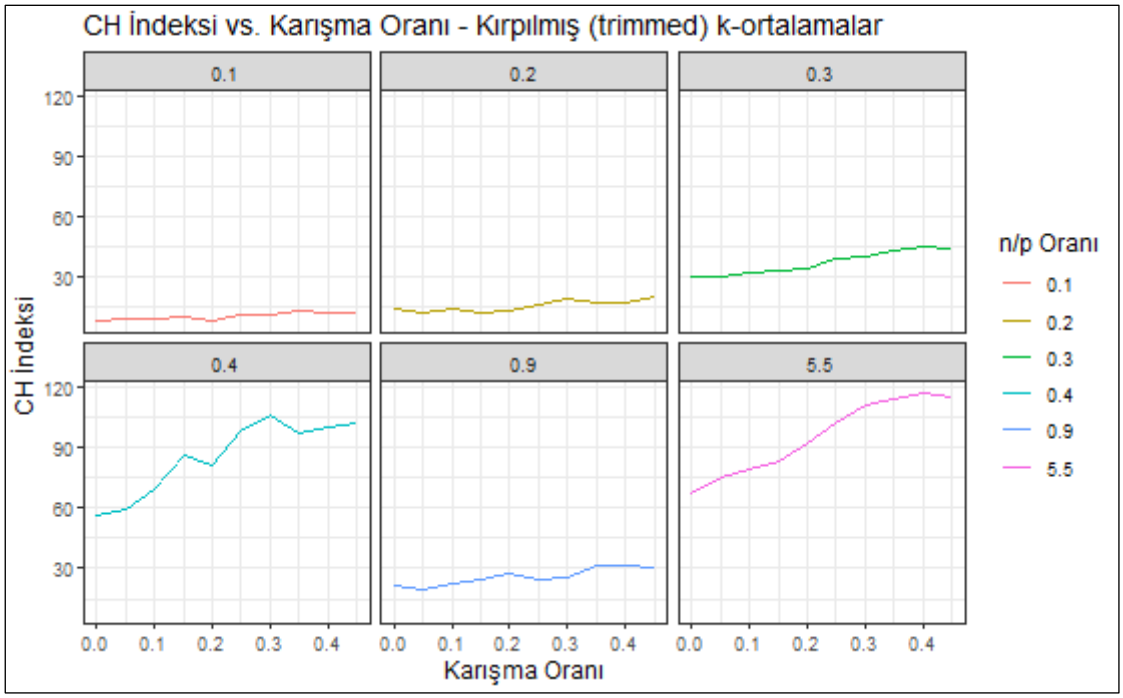
Çizelge 2.4. Kırpılmış *k*-ortalamalar kümeleme algoritması için performans değerlendirme indeksleri (AR, CH, Silhouette ve Dunn indeksleri) sonuçları

Kırpılmış (Trimmed) *k*-ortalamalar											
AR Indeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,000	0,050	0,100	0,150	0,200	0,250	0,300	0,350	0,400	0,450
20	200	0,1	0,122	0,109	0,115	0,114	0,141	0,151	0,051	0,007	-0,002	-0,040
30	150	0,2	0,069	0,057	0,052	0,105	0,060	0,030	0,043	0,090	0,112	-0,049
75	250	0,3	0,099	0,077	0,055	0,048	0,038	0,029	0,015	-0,006	-0,013	-0,011
200	500	0,4	0,116	0,112	0,102	0,095	0,082	0,076	0,064	0,024	0,007	0,018
45	50	0,9	0,317	0,186	0,204	0,219	0,044	0,049	0,038	0,105	0,162	0,033
275	50	5,5	0,089	0,107	0,061	0,065	0,060	0,025	0,019	0,005	0,004	-0,010
CH Indeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,000	0,050	0,100	0,150	0,200	0,250	0,300	0,350	0,400	0,450
20	200	0,1	8,446	9,384	9,676	10,260	8,376	11,808	11,904	12,945	12,873	12,041
30	150	0,2	14,536	12,900	14,466	12,340	13,776	15,960	19,001	17,729	17,359	20,004
75	250	0,3	30,427	30,185	32,081	33,641	34,026	39,668	40,375	42,937	45,002	44,294
200	500	0,4	56,463	59,502	69,593	86,419	81,489	98,625	106,746	96,810	100,546	102,533
45	50	0,9	21,355	19,091	22,881	24,382	27,300	24,270	25,654	31,592	31,294	30,091
275	50	5,5	67,887	74,905	78,863	83,635	92,462	102,520	111,746	114,472	117,595	115,152
Silhouette Indeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,000	0,050	0,100	0,150	0,200	0,250	0,300	0,350	0,400	0,450
20	200	0,1	0,497	0,433	0,389	0,368	0,114	0,356	0,344	0,343	0,337	0,230
30	150	0,2	0,108	0,110	0,128	0,139	0,158	0,184	0,384	0,211	0,356	0,267
75	250	0,3	0,131	0,134	0,137	0,158	0,175	0,181	0,197	0,221	0,247	0,247
200	500	0,4	0,525	0,364	0,365	0,355	0,362	0,349	0,343	0,169	0,192	0,214
45	50	0,9	0,176	0,174	0,174	0,173	0,235	0,229	0,241	0,249	0,250	0,340
275	50	5,5	0,066	0,079	0,094	0,111	0,131	0,154	0,173	0,197	0,210	0,231
Dunn Indeksi, n=20, N(μ, 3^2); Aykırı Gözlem =0,20 x n, N(30,15)											
			Karışma Oranları, N(20,10)								
n	p	n/p	0,000	0,050	0,100	0,150	0,200	0,250	0,300	0,350	0,400	0,450
20	200	0,1	1,222	1,087	0,962	1,222	0,810	1,166	1,186	1,102	1,163	0,869
30	150	0,2	0,365	0,395	0,372	0,443	0,404	0,451	1,052	0,462	0,919	0,516
75	250	0,3	0,316	0,330	0,317	0,349	0,338	0,347	0,356	0,276	0,362	0,423
200	500	0,4	0,876	0,847	0,879	0,839	0,880	0,878	0,862	0,829	0,818	0,834
45	50	0,9	0,381	0,384	0,382	0,377	0,414	0,396	0,463	0,362	0,467	0,705
275	50	5,5	0,204	0,450	0,177	0,198	0,468	0,456	0,466	0,182	0,177	0,478

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması

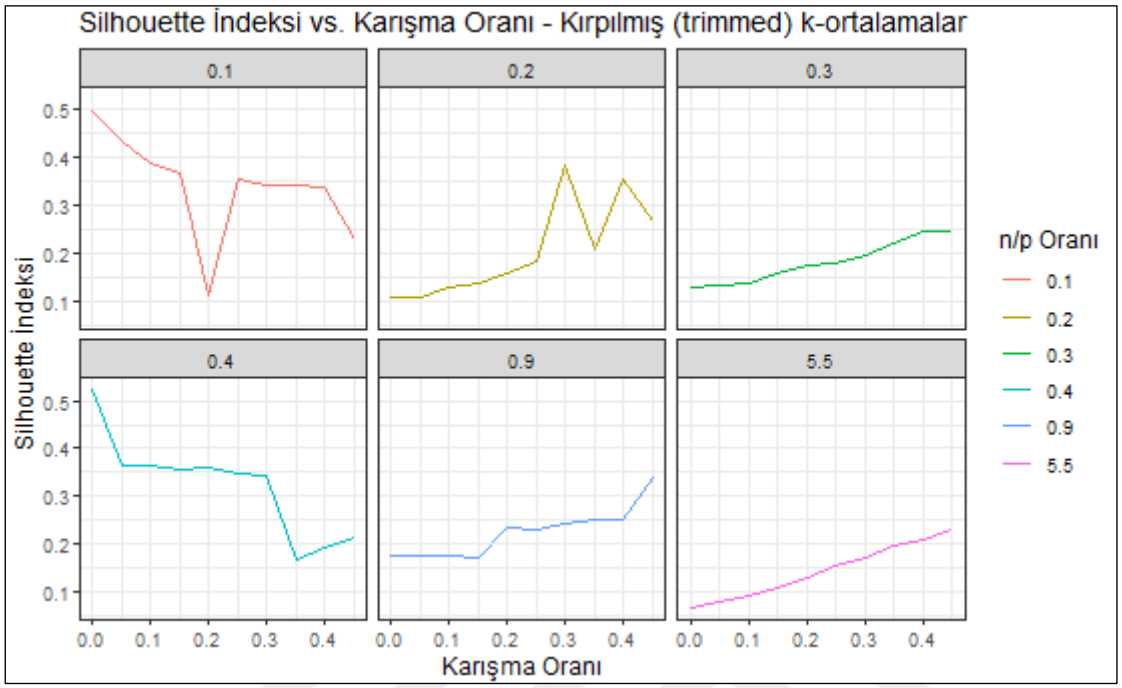


Şekil 2.17. Simülasyon senaryolarında Kırpılmış k-ortalamalar kümeleme yöntemine ilişkin AR indeksinin karışma ve n/p oranlarına göre görselleştirilmesi

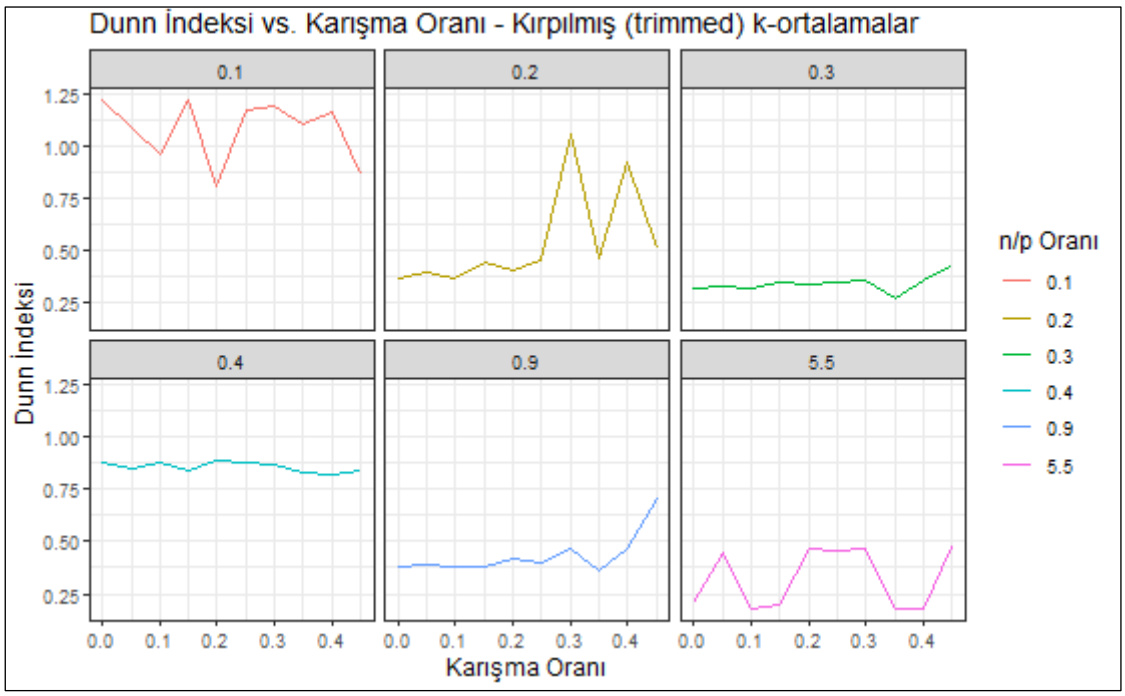


Şekil 2.18. Simülasyon senaryolarında Kırpılmış k-ortalamalar kümeleme yöntemine ilişkin CH indeksinin karışma ve n/p oranlarına göre görselleştirilmesi

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması

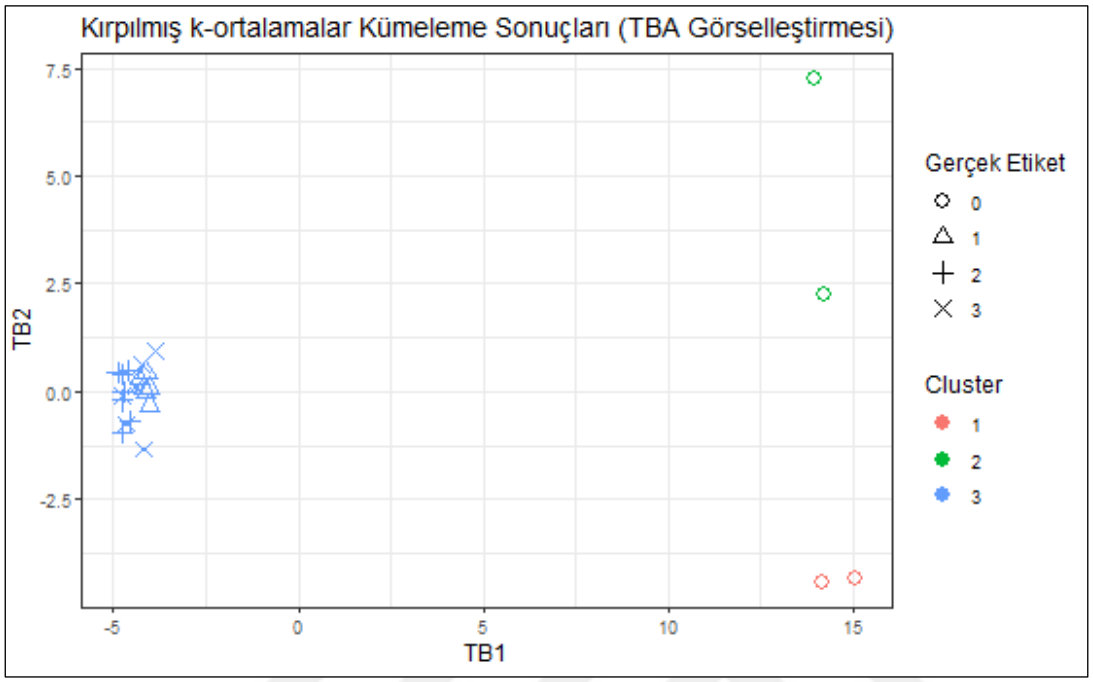


Şekil 2.19. Simülasyon senaryolarında kırpılmış k-ortalamalar kümeleme yöntemine ilişkin Silhouette skorlarının karışma ve n/p oranlarına göre görselleştirilmesi

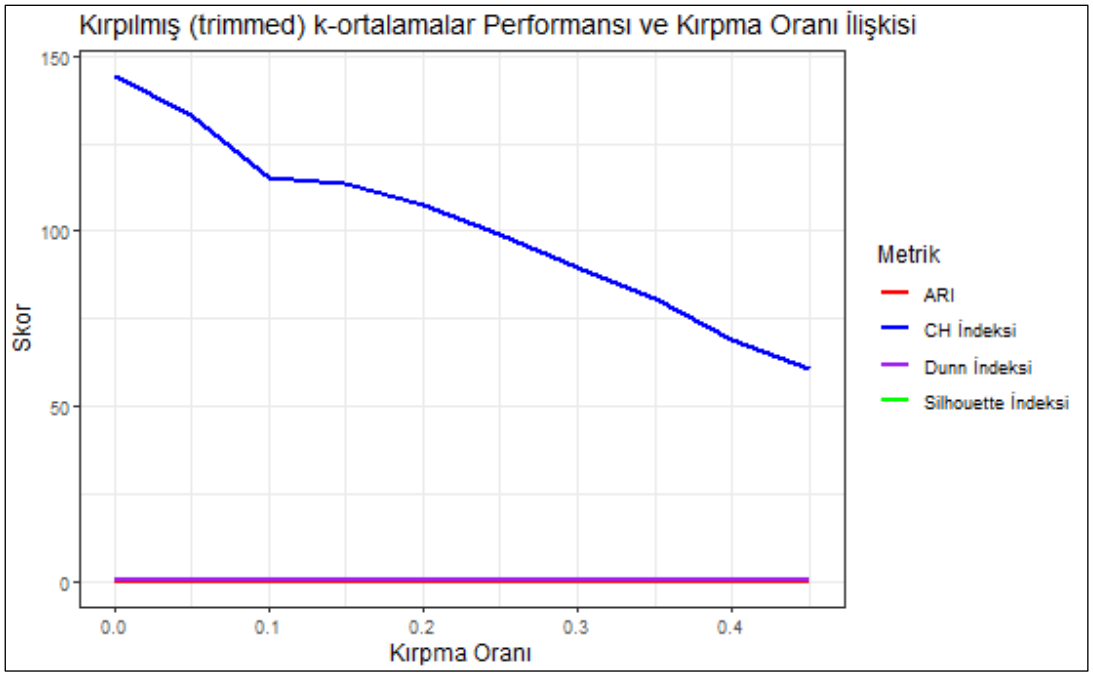


Şekil 2.20. Simülasyon senaryolarında kırpılmış k-ortalamalar kümeleme yöntemine ilişkin Dunn skorlarının karışma ve n/p oranlarına göre görselleştirilmesi

EK-2. (devam) Simülasyon özelinde kümeleme algoritmalarının performans indekslerine göre (AR, CH, Silhouette ve Dunn İndeksleri) başarı sıralaması



Şekil 2.21. Kırpılmış *k*-ortalamalar kümeleme yönteminin TBA düzleminde görselleştirilmesi ve gerçek etiketlerle karşılaştırılması ($n = 20, p = 100$)



Şekil 2.22. Kırpılmış *k*-ortalamalar yönteminin performansı: kırpma oranı ve karışma oranı etkisi

Gazili olmak ayrıcalıktır…