



T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**BÜYÜK VERİLER İÇİN KÜMELEME
ALGORİTMALARININ META-ANALİZİ**

Qoriah Indah SUSILOWATI

YÜKSEK LİSANS TEZİ

Bilişim Teknolojileri Mühendisliği Anabilim Dalı

Temmuz - 2025
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Qoriah Indah SUSILOWATI tarafından hazırlanan “**Büyük Veriler İçin Kümeleme Algoritmalarının Meta-Analizi**” adlı tez çalışması **03/07/2025** tarihinde aşağıdaki jüri tarafından oy birliği / ~~oy çokluğu~~ ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilişim Teknolojileri Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Dr. Öğr. Üyesi Levent CİVCİK

.....

Danışman

Prof. Dr. Fatih BAŞÇİFTÇİ

.....

Üye

Prof. Dr. Humar KAHRAMANLI ÖRNEK

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Hasan AYDOĞAN
FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Qoriah Indah SUSILOWATI

Tarih: 03.07.2025

ÖZET

YÜKSEK LİSANS

BÜYÜK VERİLER İÇİN KÜMELEME ALGORİTMALARININ META-ANALİZİ

Qoriah Indah SUSILOWATI

**Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilişim Teknolojileri Mühendisliği Anabilim Dalı**

Danışman: Prof. Dr. Fatih BAŞÇİFTÇİ

2025, 104 Sayfa

Jüri

**Prof. Dr. Fatih BAŞÇİFTÇİ
Prof. Dr. Humar KAHRAMANLI ÖRNEK
Dr. Öğr. Üyesi Levent CİVCİK**

Hızlı teknolojik büyümenin birçok etkisinden biri de veri patlamasıdır. Veriler karmaşıklaştıkça, daha büyük veri kümelerinden en anlamlı içgörülerini çıkarmak için kümelemeyi kullanan bilgi madenciliği yaklaşımları giderek daha popüler hale gelmektedir. Veri karmaşıklığı arttıkça kullanılan bilgi madenciliği yaklaşımlarından biri de kümelemedir. Kümeleme, verileri benzerlik özelliklerine göre gruplayabilen ve çeşitli alanlarda yaygın olarak uygulanan denetimsiz bir öğrenme yöntemidir. Bu araştırma, uluslararası konferans katılımcılarından oluşan gerçek bir veri kümesi kullanarak K-Means, DBSCAN, Mean Shift, Aglomeratif Kümeleme, Spectral Kümeleme ve Deep Embedded Kümeleme (DEC) olmak üzere altı kümeleme algoritmasının meta analizini gerçekleştirmiştir. Bu veri kümesi 22.000'den fazla veriden oluşmaktadır ve altı temel özelliğe dayalı olarak özellik seçimi, temizleme ve puanlama aşamalarından geçirilmiştir. Veri seti, altı temel özelliğe göre özellik seçimi, temizleme ve puanlama işlemlerinden geçirilmiştir. Silhouette Skoru, Davies-Bouldin Endeksi (DBI) ve Dunn Endeksi kullanılarak yapılan değerlendirme, 17 küme oluşturmaya ve 5401,72 MiB bellek tüketmesine rağmen, Aglomeratif Kümelemenin en iyi sonuçları verdiğini göstermiştir (Silhouette: 0,2645, DBI: 1,5168, Dunn: 0,533). Mean Shift ve DEC, eşit derecede düşük Silhouette Skorları (-0,055) ve DBI (-2,291) ile oldukça iyi performans gösterirken, Dunn Endeksi değerleri çok düşüktür. K-Means, zaman ve bellek açısından daha verimlidir, ancak küme kalitesi daha düşüktür (Silhouette: 0,04319, Dunn: 0,2617). DBSCAN, verimli olmasına rağmen, yüksek gürültü (258 nokta) ve en kötü DBI (9.969) değerleri üretmektedir. Spektral Kümeleme genel olarak en düşük performansı göstermiştir (Silhouette: -0,04485, DBI: 6,52696). Bu çalışmadan, Aglomeratif Kümelemenin en iyi küme kalitesini ürettiği, ancak bellek verimliliği ve aşırı küme sayısı gibi dezavantajları olduğu sonucuna varılabilir. K-Means ve DBSCAN hesaplama verimliliğinde üstündür, ancak küme kalitesinde daha az optimaldir. Spektral Kümeleme, temsili olmayan küme ayrımı ile en kötü performansı göstermiştir. Bu araştırma, büyük veri bağlamında kümelemenin yalnızca bir değerlendirme metriğinden görülemeyeceği ve en iyi yaklaşımın daha optimum sonuçlar elde etmek için dahili değerlendirme matrislerini birleştirmek olduğu sonucuna varmaktadır.

Anahtar Kelimeler: Kümeleme, Büyük Veri, Meta-Analiz, Değerlendirme, K-Means, DBSCAN, DEC, Spektral Kümeleme

ABSTRACT

MS THESIS

META-ANALYSIS OF CLUSTERING ALGORITHMS FOR BIG DATA

Qoriah Indah SUSILOWATI

**The Graduate School of Natural and Asslied Science of Selçuk University
The Degree of Master of Science in Information Technologies Engineering**

Advisor: Prof. Dr. Fatih BAŞÇİFTÇİ

2025, 104 Pages

Jury

Prof. Dr. Fatih BAŞÇİFTÇİ

Prof. Dr. Humar KAHRAMANLI ÖRNEK

Assist. Prof. Dr. Levent CİVCİK

One of the many impacts of rapid technological growth is the explosion of data. As data grows more complex, knowledge mining approaches that use clustering to extract the most meaningful insights from larger data sets are becoming increasingly popular. One of the knowledge mining approaches used as data complexity increases is clustering. Clustering is an unsupervised learning method that can group data based on similarity features and has been widely implemented in various fields. This research conducted a meta-analysis of six clustering algorithms K-Means, DBSCAN, Mean Shift, Agglomerative Clustering using a real dataset of more than 22,000 international conference participants. The dataset was processed through feature selection, cleaning, and scoring based on six key attributes. Evaluation using Silhouette Score, Davies-Bouldin Index (DBI), and Dunn Index showed that Agglomerative Clustering gave the best results (Silhouette: 0.2645, DBI: 1.5168, Dunn: 0.533), despite producing 17 clusters and consuming 5401.72 MiB of memory. Mean Shift and DEC perform quite well with equally low Silhouette Scores (-0.055) and DBI (-2.291), but very low Dunn's Index values. K-Means is more efficient in terms of time and memory but has lower cluster quality (Silhouette: 0.04319, Dunn: 0.2617). DBSCAN, although efficient, produces high noise (258 points) and the worst DBI (9,969). Spectral Clustering had the poorest performance overall (Silhouette: -0.04485, DBI: 6.52696). From this study it can be concluded that Agglomerative Clustering produces the best cluster quality, but still has the disadvantages of memory efficiency and excessive number of clusters. K-Means and DBSCAN excel in computational efficiency, but are less optimal in cluster quality. Spectral Clustering showed the worst performance with unrepresentative cluster separation. This research concludes that in the context of big data clustering cannot only be seen from one evaluation metric and the best approach is to combine internal evaluation matrices to get more optimal results.

Keywords: Clustering, Big Data, Meta-Analysis, Evaluation, K-Means, DBSCAN, DEC, Spectral Clustering

ÖNSÖZ

Üstümde emeği geçen başta ailem olmak üzere herkese ve bu tez çalışmasında her türlü yardımı, bilgiyi ve ilgiyi gösteren danışmanım Prof. Dr. Fatih BAŞÇİFTÇİ'ye teşekkür ederim. Desteklerinden ötürü şirketime ve iş arkadaşlarıma da teşekkür ederim.

Qoriah Indah SUSILOWATI
KONYA-2025



İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	ix
1. GİRİŞ.....	1
1.1. Tezin Amacı.....	6
1.2. Tezin Önemi	6
1.3. Tezin Organizasyonu	7
2. KAYNAK ARAŞTIRMASI	8
2.1. Meta Analizi Çalışmalar	8
2.2. Büyük Veri Çalışmaları	9
2.3. Çoklu Kümeleme Algoritmaları Kullanan Çalışmalar	11
2.3.1. K-Means Algoritmasını Kullanan Çalışmalar	15
2.3.2. DBSCAN Algoritmasını Kullanan Çalışmalar	17
2.3.3. Hiyerarşi Algoritmasını Kullanan Çalışmalar	19
3. MATERYAL VE YÖNTEM.....	21
3.1. Büyük Veriler	21
3.2. Kümeleme	23
3.2.1. K-Means Kümeleme	26
3.2.2. DBSCAN Kümeleme.....	30
3.2.3. Aglomeratif Kümeleme	32
3.2.4. Spektral Kümeleme.....	35
3.2.5. Deep Embedded Kümeleme	38
3.2.6. Mean Shift Kümeleme	41
3.3. Normalleştirme	44
3.3.1. Z-Skor Normalizasyonu (Standardizasyon).....	45
3.3.2. One-Hot Kodlama.....	45
4. UYGULAMA	47
4.1. Veri Seti Üzerinde Yapılan Çalışmalar	47
4.1.1. Veri Toplama	47

4.1.2. Veri Analizi.....	48
4.2. Veri Seçimi	49
4.2.1. Veri Özellikleri	50
4.3. Meta Analizi	56
4.3.1. K-Means Kümelemesi Çalışma Analizi	56
4.3.2 DBSCAN Çalışma Analizi	58
4.3.3 Mean Shift Kümelemesi Çalışma Analizi	59
4.3.4 Deep Embedded Kümelemesi Çalışma Analizi.....	60
4.3.5 Aglomeratif Kümelemesi Çalışma Analizi.....	62
4.3.6 Spektral Kümelemesi Çalışma Analizi	63
5. ARAŞTIRMA BULGULARI VE TARTIŞMA	65
5.1. K-Means Bulguları	66
5.2. DBSCAN Algoritmaları Bulguları	69
5.3. Mean Shift Algoritmaları Bulguları.....	72
5.4. Deep Embedded Algoritmaları Bulguları	74
5.5. Aglomeratif Algoritmaları Bulguları	76
5.6. Spektral Algoritmaları Bulguları	78
6. SONUÇLAR VE ÖNERİLER	81
6.1. Sonuç	81
6.2. Öneriler	85
KAYNAKLAR	87

SİMGELER VE KISALTMALAR

Simgeler

ϵ	: Epsilon
x	: Yatay (dikey olmayan) eksendir
y	: Dikey eksendir
s	: seconds (saniye)
MiB	: Mebibyte bilgisayar bilimlerinde veri boyutunu belirtmek için kullanılır.

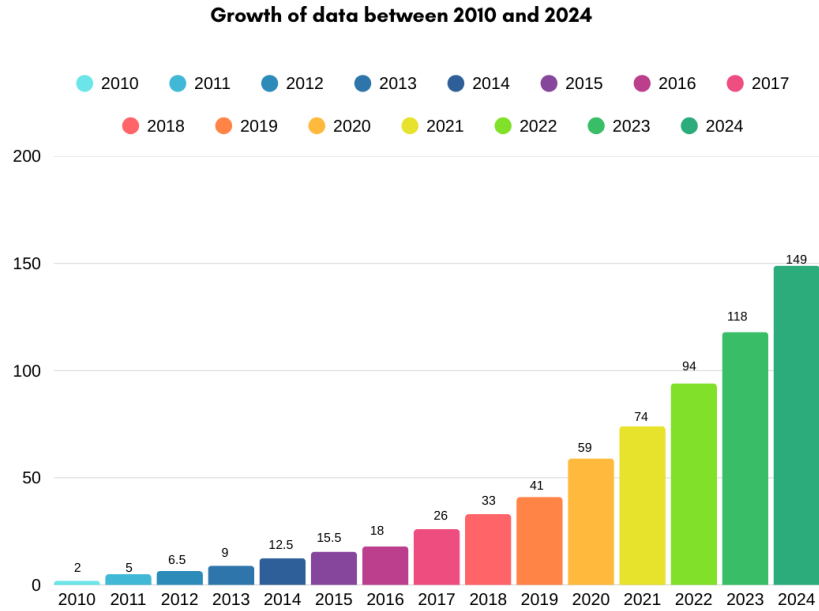
Kısaltmalar

ICT	: Information and Communications Technology (Bilgi ve İletişim Teknolojisi)
IoT	: Internet of Things (Nesnelerin İnterneti)
IDC	: International Data Corporation (Uluslararası Veri Kuruluşu)
RFM	: Recency, Frequency, Monetary (Yakınlık, Sıklık, Parasal Değer)
CLV	: Customer Lifetime Value (Müşteri Yaşam Boyu Değeri)
DBSCAN	: Density-Based Spatial Clustering of Applications with Noise (Gürültülü Uygulamaların Yoğunluk Tabanlı Mekansal Kümelenmesi)
STING	: Statistical Information Grid based Clustering (İstatistiksel Bilgi Izgarası Tabanlı Kümeleme)
GMM	: Gaussian Mixture Models (Gauss Karışım Modelleri)
DEC	: Deep Embedded Clustering (Derin Gömülü Kümeleme)
OPTICS	: Ordering Points to Identify the Clustering Structure (Kümeleme Yapısını Tanımlamak için Noktaları Sıralama)
HCA	: Hierarchial Cluster Analysis (Hiyerarşik Küme Analizi)
MDBCA	: Make Density Based Clustering Algorithm (Yoğunluk Tabanlı Kümeleme Algoritması Yapın)
FPC	: Fuzzy Partition Clustering
ELKI	: Environment for Developing KDD-Asslications Sussorted by Index-Structures (Dizin Yapıları ile Desteklenen KDD Uygulamaları Geliştirmek için Ortam)
WEKA	: Waikato Environment for Knowledge Analysis (Waikato Bilgi Analizi Ortamı)

BIRCH	: Balanced Iterative Reducing and Clustering using Hierarchies (Hiyerarşiler Kullanarak Dengeli Yinelemeli İndirgeme ve Kümeleme)
LiDAR	: Light Detection and Ranging (Işık Algılama ve Uzaklık Ölçümü)
RFM	: Recency, Frequency, Monetary (Tekrarlılık, Frekans, Parasal)
CLV	: Customer Lifetime Value (Müşteri Yaşam Boyu Değeri)
AMD	: Adaptive Multi Density (Adaptif Çoklu Yoğunluk)
KDD	: Knowledge Discovery in Databases (Veritabanlarında Bilgi Keşfi)
SGDs	: Sustainable Development Goals (Sürdürülebilir Kalkınma Hedefleri)
KLD	: Kullback-Leibler Divergence (Kullback-Leibler Iraksaklığı)
DBI	: Indeks Davies Bouldin (Dizin Davies Bouldin)
PCA	: Principal Component Analysis (Temel Bileşen Analizi)
t-SNE	: t-distributed Stochastic Neighbor Embedding (t-dağılımlı Stokastik Komşu Gömmme)

1. GİRİŞ

Teknolojik gelişmeler insanlara bilgi edinme konusunda yeni bir aşama açmıştır. *Information and Communications Technology* (ICT) insanları yeni bir yaşama, yani bilgi çağına doğru bir toplumsal dönüşüm biçimine sokmuştur. Bilgi teknolojilerindeki hızlı gelişmelerden biri de insan faaliyetlerinin veriden ayrılmaz hale gelmesine yol açmıştır. Veri, bir zamandaki gerçek bir olayı tanımlayan bir olgudur (Lubis, 2016). Günümüzde insan faaliyetleri veriden ayrılmamaktadır. İstatistik 2024'e göre her gün 3,5 kentilyon bayt veri oluşturulmaktadır (Wise, 2023). Her birey bile her saniye veri oluşturabilir ve gönderebilir. Bu veriler, onu kullananlar için katma değer ve bilgi sağlayacak enformasyona dönüşmektedir. İşlenen ya da bilgiye dönüştürülen bir veri, kısa ve uzun vadede karar vermede kullanılabilir. Şekil 1.1'de 2010 yılından 2024 yılına kadar verilerin büyümesi ile ilgili grafik verilmiştir.



Şekil 1.1. 2010'dan 2024'e kadar verilerin büyümesi

Yukarıdaki Şekil 1.1'de 2010'dan 2024'e kadar küresel veri hacmindeki artışı göstermektedir. 2010'da 2 zetabayt olan hacim, 2024'te katlanarak 149 zetabayta çıkmıştır. Dönemin başında veri büyümesi nispeten istikrarlı bir seyir izlemiş, 2010'da 2 zetabayttan 2015'te 15 zetabayta yükselmiştir. Ancak 2016'dan itibaren önemli ölçüde artarak 2020'de 59 zetabayta ulaşmıştır. Büyüme, 2020 ile 2024 yılları arasında veri hacimlerinin neredeyse üç katına çıkmasıyla devam etmiş ve 2024 yılında 149 zetabayt

sınırını aşmıştır. Bu artış, dijital teknolojilerin gelişmesiyle ortaya çıkan veri patlamasını, internet kullanıcılarındaki artışı ve *Internet of Things* (IoT) cihazlarının yaygınlaşmasını yansıtmaktadır. Ayrıca bulut hizmetleri, sosyal medya ve video içeriği ile video akışının yükselişi gibi faktörler de bu ciddi artışı tetikliyor. Üretilen veriler sadece bireysel veriler değil, aynı zamanda stratejik karar alma için büyük veri analitiğine dayanan devlet ve iş verileridir. Bu eğilim, verinin her sektörde değerli bir varlık haline geldiği yeni bir döneme işaret etmektedir.

Bir önceki şekilde görülen artan veri hacmiyle birlikte, verinin son derece bol ve karmaşık hale geldiği bir döneme girilmiştir. Verinin büyümesi, zetabayt veya daha fazla olarak ölçülen çok büyük ve çeşitli veri setlerini ifade eden Büyük Veri kavramının önünü açmıştır. Büyük verinin gelişimi, dünyadaki internet kullanıcılarının sayısındaki artışla doğru orantılıdır. Ayrıca, büyük veri genellikle bir kuruluşun veya bireyin çalışma programını desteklemek için kullanılır. Çünkü büyük verinin faydalarından biri de veri veya bilginin paydaşlar için dijital okuryazarlık olarak kullanılabilmesidir.

Büyük veri kavramı ilk olarak 2000 yılında bir endüstri analisti olan Doug Laney tarafından ortaya atılmıştır (Tysara, 2022). O dönemde Doug Laney, verinin büyük ölçekte kullanılmasını istemektedir. Buradan itibaren büyük veri ismi çeşitli ihtiyaçlarda kullanılmaya başlanmıştır. McKinsey Global'in teori üzerinde yazdığı büyük veri analizi kitabında, büyük verinin çok büyük (hacim), dağıtık (hız), çeşitli (çeşitlilik) veya sınırsız (sonsuz) veri olduğu, bu nedenle yeni değer sağlayabilecek içgörü veya bilgi elde etmek için teknik mimarı ve yenilikçi analiz yöntemlerinin kullanılmasını gerektirdiği görüşü yer almaktadır (Cholissodin ve Riyandani, 2016). Bir başka görüşe göre büyük veri; hacmi, hızı veya amaç çeşitliliği nedeniyle geleneksel veri tabanları veya basit veri işleme araçları kullanılarak depolanması, yönetilmesi ve analiz edilmesi zor olan veri kümesidir (Santosa, 2007).

Büyük veri günümüzde başlı başına bir fenomendir. Bunun nedeni, insanların söylediği, yaptığı veya gözlemlediği her şeyden veri oluşturulabilmesi ve bunun daha fazla veri üretecek olmasıdır (Hoy, 2014). Bazı taraflar, büyük hacimlerde, çeşitlilikte, yüksek karmaşıklıkta ve yüksek depolama oranlarında mevcut olan verileri işleyebilir ve kullanabilir ve yüksek ekleme oranları veri sahiplerine muazzam faydalar sağlayabilir. Veriden değer elde etmek, onu işlemek için alternatif yollar seçmeyi gerektirir (Dumbill, 2013).

İnternet kullanımındaki artış, ağ üzerindeki verinin sosyal medya sensörleri, videolar, gözetim ve akıllı şebekeler gibi çeşitli kaynaklardan oluşturulan trilyonlarca

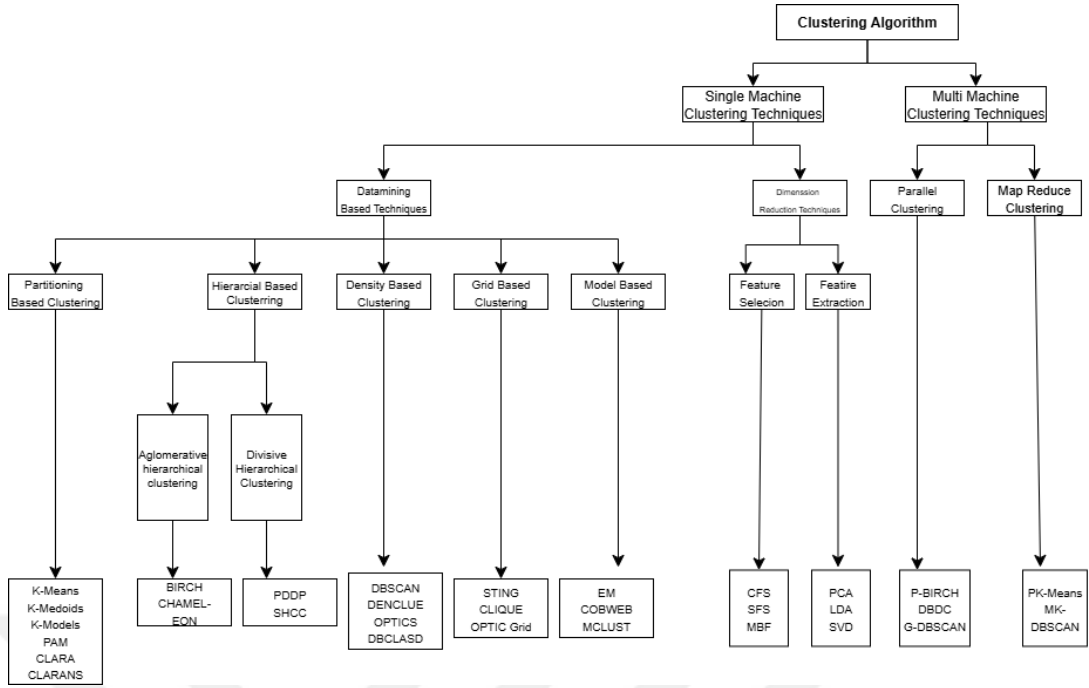
baytlık veriye kadar büyümesine neden olmaktadır. Bu tanım, büyük veri teriminin geldiği yerdir. Büyük veri, hacmi, hızı veya çeşitli hedefleri geleneksel veri tabanları ve veri işleme araçları kullanılarak depolanması, yönetilmesi ve analiz edilmesi zor olan bir dizi veridir (Santosa, 2007).

Büyük veri kullanımı, farklı ve kaliteli ürün ve hizmetlerin sağlanmasında büyük ölçekte veri kullanan bankacılık şirketleri gibi günlük yaşamda yaygın olarak kullanılmaktadır. Son yirmi yılda muazzam bir veri büyümesi yaşanmıştır. Bir araştırma şirketi olan *International Data Corporation* (IDC) tarafından hazırlanan bir rapora göre, 2012-2020 yılları arasında dijital evrendeki bilgi miktarı 35 trilyon gigabayt artacaktır. Bu miktar fiziksel evrendeki yıldızların sayısına eşittir (Bhadani ve Jothimani, 2016). Bu gelişme, dünyadaki internet kullanıcı sayısındaki artışla doğru orantılıdır.

Büyük veri, yapılandırılmış, yarı yapılandırılmış ve yapılandırılmamış verilerden oluşur. Bunların her biri farklı kümeleme yaklaşımlarına sahiptir ve her birinden değerli içgörüler elde etmek için farklı kümeleme yaklaşımları gerektirir. Temel sorun, toplanan verilerin nasıl işleneceği ve verilerin makul bir sürede yeni içgörüler elde etmek için nasıl elde edileceğidir. Bir dizi veriden yeni içgörülerin çıkarılmasında kullanılan yöntemlerden biri kümelemedir. Kümeleme, verilerin belirli benzerliklere veya özelliklere göre gruplandırılmasını sağlayan bir veri madenciliği tekniğidir, böylece büyük verilerde gizli olan örüntüler veya yapılar tespit edilebilir.

Kümeleme, denetimsiz öğrenmeye dahil olan bir veri madenciliği işleme tekniğidir. Sınıflandırmada olduğu gibi etiketler veya öğretmenler olmadan öğrenmeyi gerçekleştirebilir (Arifah ve ark., 2023). Kümeleme yöntemi, verileri birkaç gruba ayırır, böylece aynı özelliklere sahip veriler aynı kümede toplanır (Priyatman ve ark., 2019). Kümeleme teknikleri, bilgi indekslemeye kümeleme ekleyerek büyük miktarda veri ve aykırı değerlerle ilgili sorunların üstesinden gelebilir.

Kümeleme algoritmaları, verileri anlaşılması ve analiz edilmesi daha kolay gruplar halinde düzenleyerek eldeki verilere ilişkin daha derin içgörüler elde edilmesini sağlayabilir. Büyük verinin karmaşıklığıyla birlikte, verileri işlemek için doğru kümeleme algoritmasını seçmek giderek daha önemli hale gelmektedir. Kaynakların çeşitliliği, büyük veri boyutu ve farklı veri yapıları, doğru ve ilgili sonuçlar sağlayabilecek bir algoritma seçiminde kendi zorluklarını yaratacaktır. Kümeleme tekniklerinin bir örneği Şekil 1.2’de gösterilmektedir (Djouzi ve Beghdad Bey, 2019).



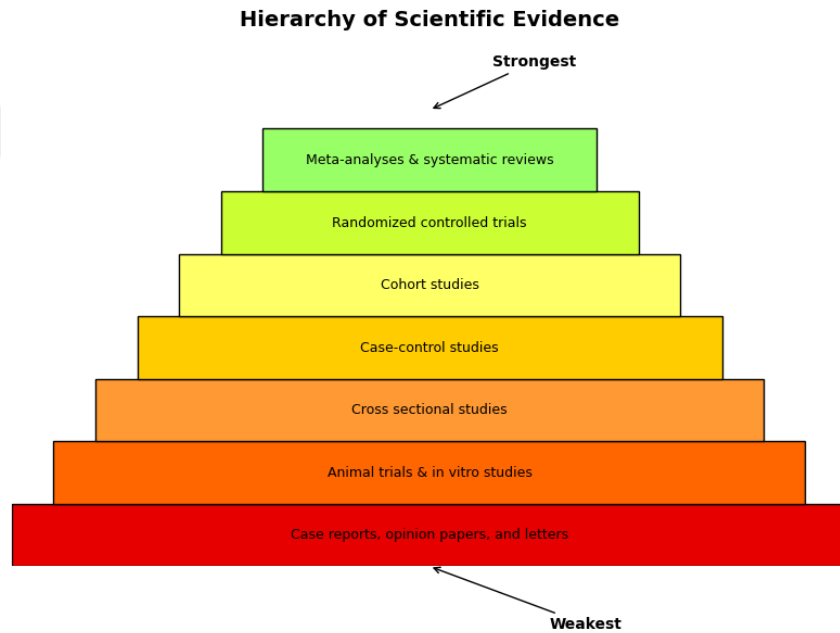
Şekil 1.2. Büyük Veri kümeleme Algoritmalarının Kategorilendirilmesi.

Yukarıdaki şekil, tek makineli kümeleme teknikleri ve çok makineli kümeleme teknikleri olmak üzere ikiye ayrılan kümelemenin bir gruplandırmasıdır. Kümeleme algoritmaları, önceden belirlenmiş küme sayısına göre veriyi bölen K-Means gibi parçalama algoritmaları, Agglomerative (sözlük karşılığı yığınsal) veya bölme yöntemi ile kademeli küme yapısı oluşturan hiyerarşik algoritmalar, verinin yoğunluğuna dayalı kümeleri tanımlayan *Density-Based Spatial Clustering of Asslications with Noise* (DBSCAN) gibi yoğunluk tabanlı algoritmalar, eri alanını ızgara hücrelerine bölen ve hücre yoğunluğuna göre verileri gruplandıran *Statistical Information Grid based Clustering* (STING) gibi ızgara tabanlı algoritmalar, ve verilerin belirli bir modelden üretildiğini varsayan ve en uygun model parametrelerini arayan *Gaussian Mixture Models* (GMM) gibi model tabanlı algoritmalar gibi ana kategorilere ayrılır.

Kümeleme algoritmalarının çeşitliliği, boyut, dağılım ve istenen küme sayısı gibi farklı özellikler kullanılan kümelemenin etkinliğini kesinlikle etkilediğinden, en iyi algoritmanın seçiminde kesinlikle yeni zorluklar ortaya çıkarmaktadır. Belirsizlik, beklenen verilerin yanlış analiz sonuçlarına neden olabileceği durumlarda optimal olmayan sonuçlara yol açabilir. Bu nedenle, bir meta-analiz yaparak her bir algoritmanın en iyi performansı sağladığını anlamak için çeşitli kümeleme algoritmalarının derinlemesine bir değerlendirmesine ihtiyaç vardır.

Kümeleme işlemi, haberleri birkaç parçaya ayırmayı amaçlar. Böylece bir karışımda veriler eşit olmayan parçalardaki nesneleri kullanarak birçok benzerliğe ve gerçek farklılıklara sahip olur. Bu algoritmanın amacı verileri gruplara ayırmaktır. Bu algoritma, sınıf etiketleri olmayan veri biçiminde girdi kabul eder. Bu, girdiyi $(x1, y1)$, $(x2, y2)$, ..., (xi, yi) vektörleri şeklinde kabul eden denetimli öğrenmeden farklıdır; burada xi bir eğitim verisinden alınan veriyi ve yi de xi için sınıf etiketidir (Nanthiya ve ark., 2010).

Meta-analiz, bir etki alanındaki çeşitli yaklaşımların performansını değerlendirmek için yürütülen bir araştırma metodolojisidir. Çeşitli durumlarda veya veri kümelerinde kümeleme algoritmalarının performansındaki ortak kalıpları bulmak için mevcut araştırma sonuçlarını birleştirir. Bu araştırmanın odak noktası, güç analizi alanındaki gerçek koşulları temsil ettiği düşünülen uluslararası bir konferans veri deneyinden alınan büyük verilerin kümelenmesinde algoritmaların performansını değerlendirmektir. Şekil 1.3 bilimsel kanıt hiyerarşisini göstermektedir (Vikipedi, 2023).



Şekil 1.3. Büyük Veri kümeleme Algoritmalarının Kategorilendirilmesi.

Şekil 1.3, en güçlü ve güvenilir kanıtların sistematik incelemeler ve meta analizlerden, en zayıf kanıtların ise uzman görüşlerinden geldiğini göstermektedir. Bu piramit, araştırmacıların, sağlık hizmeti uygulayıcılarının ve politika yapıcıların klinik kararlar alırken veya politika önerilerinde bulunurken bilimsel kanıtların kalitesini değerlendirmelerine yardımcı olmaktadır. Dolayısıyla bu araştırma, büyük veri bağlamında uygulanan çeşitli kümeleme algoritmalarının değerlendirilmesinde

kullanılacak meta-analizlere odaklanmaktadır. Bu araştırma, önceki çeşitli çalışmaların sonuçlarını değerlendirerek ve karşılaştırarak, çeşitli büyük veri uygulamalarında hangi kümeleme algoritmalarının en etkili şekilde kullanıldığına ilişkin daha net ve kanıta dayalı rehberlik sağlamayı amaçlamaktadır.

1.1. Tezin Amacı

Son on yıldaki muazzam veri artışı, büyük veri, karmaşık bilgilerin düzenlenmesi ve analiz edilmesinde önemli zorluklar ortaya çıkarmaktadır. Kümeleme, büyük miktarda veriyi benzerliklerine göre gruplandırarak düzenleme ve anlama konusunda çözümler sunan önemli veri çıkarma tekniklerinden biridir. Bununla birlikte, mevcut kümeleme algoritmalarının sayısı, karmaşık büyük veri bağlamında hangi algoritmanın en etkili şekilde kullanılacağına ilişkin temel soruları gündeme getirmektedir. Bu tezin amacı, önceki çalışmaların meta-analiz yaklaşımını kullanarak kümeleme algoritmalarının kapsamlı bir değerlendirmesini ve karşılaştırmasını yapmaktır. Ayrıca, meta-analiz sonuçlarında büyük veriler için belirli bir kümeleme algoritmasının tercih edildiğini gösteren eğilimler veya kalıplar olup olmadığını, algoritma yapılandırma parametrelerinin büyük veriler üzerinde kümeleme sonuçlarını nasıl etkilediğini, kümeleme algoritmaları farklı alanlardan veya veri kaynaklarından gelen büyük veri kümelerine uygulandığında önemli performans farklılıkları olup olmadığını ve büyük veriler için kümeleme algoritmalarının kullanımında dikkate alınması gereken ölçeklenebilirlik, hesaplama verimliliği veya bellek yönetimi gibi özel hususlar olup olmadığı da araştırılmaktadır. Son olarak, büyük veri zorluklarını ele almak için kümeleme algoritmalarının geliştirilmesinde geleneksel ve yenilikçi yaklaşımları da karşılaştıracaktır.

1.2. Tezin Önemi

Bu tez, hızlı teknolojik ilerlemeler ve dijitalleşmenin yol açtığı büyük veri analizindeki önemli bir konuyu ele almaktadır. Büyük veri, artık geleneksel teknolojiler tarafından işlenemeyen büyük veya karmaşık veri kümesidir (Hurwitz ve ark., 2013). Veri hacmi katlanarak arttıkça, etkili ve verimli analiz tekniklerine duyulan ihtiyaç da aciliyet kazanmaktadır. Kümeleme tekniklerinden biri olarak kümeleme algoritmalarının bolluğu büyük bir zorluk olacaktır. Bu nedenle, bu tez, özellikle Büyük Veri bağlamında çeşitli kümeleme algoritmalarının sistematik bir değerlendirmesini ve karşılaştırmasını sağlamak için acil hale gelmektedir. Meta-analizin kullanılması, her bir algoritmanın

performansı hakkında daha derin bir içgörü sağlamayı ve araştırmacılara ve uygulayıcılara ihtiyaçları için en uygun yöntemi seçmelerinde yardımcı olmayı amaçlamaktadır. Bu tezin önemi, Büyük Verinin karmaşıklığıyla başa çıkmak için güvenilir bir rehber sağlamanın yanı sıra hızla büyüyen veri çağında bilim ve teknolojinin gelişimine katkı sağlamaktır.

1.3. Tezin Organizasyonu

Büyük Veriler İçin Kümeleme Algoritmalarının Meta-Analizi başlıklı tezde elde edilen veriler ve bilgiler yazıya dökülmüştür. Bu tezde yapılan çalışmalar aşağıda 6 bölümde açıklanmıştır.

İlk bölüm konuya genel bir bakış sağlayan bir giriş olup, bu çalışmanın amacı ve önemi vurgulanmaktadır.

İkinci bölümde, konuyla ilgili benzer araştırmaların ele alındığı bir literatür taraması, büyük veri, kümeleme ve kullanılacak algoritmalar hakkında bilgiler yer almaktadır. Aynı zamanda kullanılan algoritmalarla ilgili daha önce yapılmış araştırmaların bilgileri ve özellikleri sunulmuştur.

Üçüncü bölümde, bu araştırmada kullanılacak olan araştırma yöntemi ile ilgili genel bilgiler sunulacak, ayrıca kullanıcının tam bir tanımı, global tasarımın detaylı bir tanımı, prosedür tasarımı, kod tasarımı ve veritabanı tasarımı yazılacaktır.

Dördüncü bölümde, uygulama tasarımı ve analizi ile ilgili bilgiler ve araştırma sırasında elde edilen bulgular yer almaktadır,

Beşinci bölüm kapanış, yani sonuçlar ve çeşitli önerilerle ilgili olarak açıklanacaktır.

Son bölüm, bu araştırma sürecinde daktilo edilmiş ve kullanılmış olan tüm yazılı veya basılı kaynakları içeren kaynakça ve eklerdir. Hazırlık ve tasarım sürecinde kullanılan dokümanları içermektedir.

2. KAYNAK ARAŞTIRMASI

Bu tezde, genel olarak büyük veride kullanılan kümeleme araştırmaları ve örnek olay olarak eğitim verilerini kullanan araştırmalar ele alınmıştır. Bu literatür çalışması, yetkin bir sistem geliştirme çabası içinde yürütülecek araştırma yönteminin bir aşaması olarak gerçekleştirilmektedir. Aşağıda, bu tezin konusu ile ilişkisi olan araştırmalar yer almaktadır.

2.1. Meta Analizi Çalışmalar

Çoğaltay ve Karadağ, 2015'in çalışmasında, bir bütün olarak daha fazla ve güvenilir sonuçlar elde etmek için çeşitli çalışmaların sonuçlarını birleştirmede istatistiksel bir teknik olan meta-analiz açıklanmaktadır. Bu çalışma, meta-analizin bireysel çalışmalar için çok yararlı olduğunu, çünkü örneklem büyüklüğü veya metodoloji ile sınırlı olabileceğini belirtmektedir. Bu çalışmada ayrıca analiz için doğru modelin seçilmesi açıklanmakta ve çalışmalar arasındaki bulgulardaki farklılıkların heterojenliğini anlamının önemi vurgulanmaktadır (Çoğaltay ve Karadağ, 2015).

Espen 2014 yılında, etki büyüklüklerinin sistematik analizinden yoksun olan araştırmacılara ve öğrencilere meta-analiz yaklaşımını tavsiye etmiştir. Araştırmacı, tezlerde meta-analiz kullanımının tezlerin yayınlanma şansını artırabileceğini belirtmiştir (Sjoberg, 2014).

Andy ve Raphael 2010 yılında meta-analiz yapma konusunda bir eğitim yazısı kaleme almıştır. Bu çalışmada, meta-analiz yaparken atılması gereken adımların makalelerin toplanması, dahil etme kriterlerinin belirlenmesi, etki büyüklüklerinin hesaplanması, uygun analiz yöntemlerinin seçilmesi ve Vevea ve Woods modeli kullanılarak yayın yanlılığının etkilerinin değerlendirilmesi olduğu belirtilmiştir. Bu çalışma ayrıca, etki büyüklüklerinin dağılımı, değişkenlik, popülasyon etki tahminleri ve yayın yanlılığının etkisi de dahil olmak üzere nihai sonuçların açık bir şekilde yazılması gerektiğini önermektedir (Field ve Gillett, 2010).

Dr. Stefan 2015 yılında düzenlenen bir seminerde meta-analizin istatistiksel araçlardan daha fazlasını içerdiğini belirtmiştir. Daha fazla analiz yapmadan önce güçlü bir veri tabanı sağlamak için atılması gereken birkaç önemli adım vardır. Bu araştırma aynı zamanda önyargıdan kaçınmak ve doğru ve güvenilir sonuçlar üretmede istatistiksel temeli ve uygulama alanını anlamak gerektiğini belirtmektedir (Strohmeier ve Duveneck, 2015).

2020 yılında Sonal Jain meta-analiz kullanarak yaptığı çalışmada, dikkatle yürütülen meta-analiz tekniklerinin literatürdeki karmaşık ve tartışmalı konuların niceliksel ve ayrıntılı bir şekilde gözden geçirilmesine yardımcı olabileceğini belirtmiştir. Klinik araştırmalarda bu araca güvenilse de doğru ve güvenilir sonuçlar üretmek için derin bir istatistik anlayışına ihtiyaç olduğunu da sözlerine eklemiştir (Jain, 2020).

Lorry ve Jerry 2008 yılında meta-analiz tekniklerine ilişkin metodolojik literatürün gözden geçirmeye dayalı bir analizini yapmıştır. Makalede, son yıllarda insan boyutu üzerine yapılan araştırmaların memnuniyet, motivasyon, kalabalık ve normlar gibi kavramların anlaşılmasını derinleştirdiği belirtilmektedir. Bu araştırma, benzer değişkenler ve çalışma yöntemleriyle üretilen çok sayıda veri kümesiyle, meta-analiz gibi bütünleştirici analitik yaklaşımların giderek daha önemli hale geldiğini savunmaktadır. Meta-analiz, nicel araştırmaların sentezi için titiz bir metodoloji sunmaktadır, ancak bu literatürdeki kullanımı hala sınırlıdır. Meta-analiz, büyük çaba, potansiyel hatalar ve önyargı gibi zorluklara rağmen, araştırmacıların süreçte uzmanlıklarını kullanmalarına olanak tanır. Ancak dikkatli bir planlama ve yorumlama ile bu sınırlamaların üstesinden gelinir. Yazarlar ayrıca, meta-analiz sonuçlarının daha doğru ve temsili olmasını sağlamak için anlamlı olsun ya da olmasın tüm çalışmalar için etki büyüklüklerinin hesaplanmasının önemini vurgulamaktadır (Shelby ve Vaske, 2008).

2.2. Büyük Veri Çalışmaları

Djouzi ve Beghdad-Bey tarafından 2019 yılında yazılan makalede, büyük veride kullanılan kümeleme algoritmaları incelenmiştir. Djouzi ve Beghdad-Bey tarafından 2019 yılında yazılan makalede, büyük veride kullanılan kümeleme algoritmaları incelenmiştir. Bu araştırma, geleneksel tek makine tekniklerinin toplanan büyük miktarda veriyi yönetmek için yeterince güçlü olmayacağı sonucuna varan bu işlem sırasında kullanılan makine sayısına dayalı algoritmalara bir sınıflandırma yaklaşımı önermektedir. Bu araştırma aynı zamanda yüksek hesaplama hızı ve paralel paradigması sayesinde MapReduce'un paralel algoritmaların uygulanması için iyi bir platform olarak kullanılmasını önermektedir (Djouzi ve Beghdad-Bey, 2019).

Rahmat ve arkadaşları tarafından 2021 yılında yükseköğretimde büyük verinin uygulanması üzerine bir araştırma yapılmıştır. Bu çalışmada, öğrencileri toplam 395 veri ile 5 kümeye ayırmak için devamsızlık, 1. sınıf, 2. sınıf, 3. sınıf olmak üzere 4 değişken kullanılmıştır. 1. küme, en büyük ortalama değere sahip olan ve küçük bir öğrenci grubuna sahip olan bir kümedir. Bunun yanı sıra, bu çalışma, araştırmada değerin en üst

düzeye çıkarılabilmesi için doğrudan her üniversitenin akademik bilgi sisteminden alınan veri kümelerinin kullanılmasını da önermektedir (Purnomo ve ark, 2021).

2014 yılında Nada ve Ahmda, veri patlaması nedeniyle büyük veri analitiği ile ilgili araştırmaları gözden geçirmiştir. Bu araştırma, büyük verinin iş dünyasında değişim yaratma ve büyük veride saklı içgörülerini ortaya çıkararak karar alma süreçlerini iyileştirme potansiyelini incelemektedir. Bu makalede büyük veri analitiği kavramı, özellikleri ve gerekli veri depolama, yönetim ve işleme teknikleri tartışılmaktadır. Büyük veri, daha iyi kararları desteklemek için sağlık ve perakende gibi çeşitli sektörlerde uygulanabilir. Zorluklara rağmen, büyük veri iyi yönetilirse parlak bir geleceğe sahiptir ve gelecekteki araştırmalar bu zorlukların üstesinden gelmeye yardımcı olabilir. Büyük veri analitiği, çeşitli alanlarda karar vericilere önemli faydalar sağlama potansiyeline sahiptir (Elgendy ve Elragal, 2014).

2018 yılında Youssra ve Sara büyük veri üzerine bir literatür araştırması gerçekleştirmiş ve sonuçları büyük verinin kişisel ve profesyonel hedeflerin gerçekleştirilmesinde teknoloji kullanıcıları tarafından üretilen sayısal verileri ifade ettiğini açıklamıştır. Büyük veri analizi gizli kalıpları ortaya çıkarabilir. Pazar eğilimleri, müşteri tercihleri ve karar vermede diğer yararlı bilgiler. İş Zekâsı, veri analizi yaparken karar verme kısmını halledebilir (Riahi ve Riahi, 2018).

2022 Hemn, büyük veri uygulamaları ve teknolojilerinin genellikle zaman alıcı karmaşıklığını ve muazzam hesaplama karmaşıklığını tartışan ve ilgili zorlukları tanımlayan büyük veri üzerine bir anket gerçekleştirdi. Bu makale, büyük veride kullanılan teknikleri yayın yılı, performans ölçütleri ve kullanılan yöntemler gibi faktörlere göre analiz etmekte, avantaj ve dezavantajlarını değerlendirmektedir. Buna ek olarak, büyük veri fırsatlarını ve uygulamalarını gözden geçirmekte ve Hadoop, MapReduce ve Spark gibi büyük veri işleme çerçevelerine genel bir bakış sunmaktadır. Ancak bu araştırma, derin öğrenme, makine öğrenimi ve veri görselleştirme gibi gelecekteki araştırmalarda ele alınan tekniklere değinmemektedir (Abdalla, 2022).

Semarang Bilgisayar Bilimi ve Teknolojisi Üniversitesi Rektörü Dr. Joseph Teguh Santoso tarafından 2020 yılında yazılan Büyük Veri Analizi kitabı 12 bölümden oluşmaktadır. Bölüm I'de, bu kitap büyük veri analizinin veri temizleme, veri toplama, işleme ve verilerin büyük ölçekte görselleştirilmesinden başlayarak çeşitli adımları olduğunu açıklamaktadır. Ayrıca bu kitap, veri toplama, veri depolama, toplu analiz, gerçek zamanlı analiz, etkileşimli veritabanı sorgulama ve farklı bulut hizmetlerinde

uygulanabilecek sunum gibi Büyük Veri çerçeve kategorilerini kapsayan dört analiz modeli önermektedir (Santoso, 2020).

2023 Showimy, yaygın olarak kullanılan büyük veri kullanımını etkileyen faktörler üzerine bir çalışma yürütmüştür. Bu çalışma, küçük ve orta ölçekli işletmelerde büyük veri analistliği üzerinde en etkili faktörler olan 13 faktörü çıkarmıştır. Bu faktörler arasında üst yönetim desteği, eğitim, göreceli avantaj, bilgi teknolojisi altyapısı, güvenlik, uyumluluk, karmaşıklık, uyarlanabilirlik, hükümet politikası, yetkinlik, iş birliği, dijital dönüşüm araçları ve karar kalitesi ve karar kalitesi yer almaktadır (Aldossarive ark., 2023).

Qasim ve Syed tarafından yürütülen araştırma, küçük ve orta ölçekli işletmelerde büyük veri analizi uygulaması ile kurumsal performans arasındaki ilişkiyi anlamayı amaçlamaktadır. Araştırmanın sonuçları, büyük veri analizi uygulamasının Kobi'lerde kurumsal performans üzerinde olumlu ve önemli bir etkiye sahip olduğunu göstermektedir. Bulgular, özellikle gelişmekte olan ülkelerdeki Kobi'lerde üst yönetimin karar alma süreçlerini iyileştirmesi ve gelişmiş kurumsal performans yoluyla rekabet avantajı elde etmesi için önemli stratejik ve pratik çıkarımlar sağlamaktadır (Shabbir ve Gardezi, 2020).

Davide ve arkadaşları tarafından 2024 yılında gerçekleştirilen ve büyük verinin 15 yılını inceleyen araştırma, Büyük Verinin geleceğinin teknolojik ilerlemeleri güçlü bir etik duruşla birleştireceğini, inovasyon ve sorumluluğun el ele gittiği teknolojik ve etik açıdan gelişmiş bir ortam yaratacağını göstermektedir. Çalışma, Büyük Verinin çeşitli alanlarda önemli bir oyuncu haline geldiğini, teknolojik değişimlere uyum sağladığını ve değişen iş ihtiyaçlarını karşıladığını göstermektedir (Tosi ve ark., 2024).

2.3. Çoklu Kümeleme Algoritmaları Kullanan Çalışmalar

El Mehdi, 2019 yılında K-Means ve DBSCAN algoritmalarını kullanarak parmak izi segmentasyonunu iyileştirme üzerine bir araştırma yapmıştır. Çalışmada kullanılan K-Means ve DBSCAN algoritmaları, parmak izi segmentasyon görüntülerini iyileştirmeye yönelik olup, önerilen yöntemin ortalama segmentasyon hata oranı açısından mevcut bazı yeni tekniklerden daha etkili olduğunu önemli ölçüde ortaya koymaktadır (Cherrat ve ark., 2019).

Erwin ve ark tarafından 2023 yılında yapılan çalışmada, toplam 2114 veri ve 11 öz nitelik ile çeşitli kümeleme teknikleri kullanılarak öğrenci kabul verileri kullanılarak araştırma yapılmıştır. Bu çalışmada, K-Means, hiyerarşik ve DBSCAN kümeleme

algoritmalarının kullanılmasının neredeyse aynı sayıda küme ile sonuçlandığı sonucuna varılmıştır. Buna ek olarak, kümeleme büyük veri kümelerindeki örüntüleri ve benzerlikleri tanımlar ve daha sonra faktörleri ve diğer veri kaynaklarını ekleyerek öğrenci dersleri ve ödevleriyle ilgili kararlar gibi daha ayrıntılı kararlar verebilir (Cahapin ve ark., 2023).

Turgay Tığas Bilgin tarafından 2005 yılında yapılan araştırmada DBSCAN, *Ordering Points to Identify the Clustering Structure* (OPTICS) ve K-Means algoritmalarının uygulamaları karşılaştırılmıştır. Turgay, küresel olmayan küme şekillerinde DBSCAN ve OPTICS'in K-Means'e kıyasla maksimum sonuç verdiğini belirtmiştir. Buna ek olarak, K-Means algoritması uygulanırken küme sayısının başlatılması gerekir. Bu, küme sayısı doğru tahmin edilmediğinde yanlış çıktı verecektir (Tugay Bilgin ve Çamurcu, 2005).

Peerzada tarafından yürütülen araştırma, genel küme kalitesini sıralayan parametreleri değerlendirmek ve dört kümelemenin performansını karşılaştırmak için farklı veri kümeleri kullanarak K-Means, DBSCAN, *Hierarchical Cluster Analysis* (HCA) ve *Make Density Based Clustering Algorithm* (MDBCA) olmak üzere 4 ana kümeleme algoritması üzerindeki çalışmaların karşılaştırmalı bir analizini yapmıştır. Bu çalışma, dört algoritmanın kümelendiğinde bazı verilerde belirsizlikler (gürültü) olduğunu, yani K-Means'in minimum sürede kümeler oluşturduğunu, MDBCA'nın küme oluşturmada Basit K-means ile neredeyse aynı performansı gösterdiğini, HCA'nın gürültülü verilere karşı daha hassas olduğunu ve zaman karmaşıklığı ile çok fazla değişkenlik gösterdiğini ve DBSCAN'ın yüksek varyans yoğunluğuna sahip veriler için uygun olmadığını açıklamaktadır. Zaman karmaşıklığı ve kullanılan veri kümesi açısından K-Means diğer 3 algoritmadan daha iyi sonuçlar vermektedir. Bu çalışma ayrıca, genel küme kalitesini sağlayan bir parametre olmadığını gösteren verileri de değerlendirmiştir (Ahmad ve Dang, 2015).

2015 Özge Pasin ve Handan Ankaralı, farklı kümeleme algoritmalarının sonuçlarının gerçek veri setlerine uygulanarak karşılaştırılabileceğini belirtmiştir. Bu çalışmanın değerlendirmesi, Kassa katsayısının istatistiği gösterdiğini ve derecesinin düşük olduğunu belirtmektedir. Veri seti açısından en uygun ve en hızlı K-Means algoritması, kesişen Framingham risk grupları arasında grup dağılımı açısından doğru sonuçlarla doğru orantılıdır. Kullanılan verilerle elde edilen diğer sonuçlar da veri dağılımının ve veri setinin özelliklerinin değerlendirilmesini sağlamaktadır. Kümeleme

algoritması sağlık sektöründe hastalıkların planlanması ve teşhisi için bir tanı aracı olarak kullanılabilir (Pasin ve Ankarali, 2015).

Agustin ve ark. pansiyon verilerinin K-Means yöntemi ile gruplandırılması sonucunda grup 1’de 24 veri, grup 2’de 145 veri ve grup 3’te 131 veri elde edildiğini belirtmiştir. DBSCAN kullanıldığında ise eps değerinin 0,1 ve min Pts değerinin 2 olduğunu ve 292 veri ile 1. grup, 2 veri ile 2. grup ve 4 veri ile 3. grup ve gürültü verisi haline gelen 2 veri ürettiğini belirtmiştir. *Silhouette* indeksi hesaplamasının sonuçları, K-Means yönteminin 0,463 ve DBSCAN yönteminin 0,281 değerini ürettiğini göstermektedir, böylece bu çalışmada DBSCAN yönteminin pansiyon verilerini gruplamada daha iyi olduğu sonucuna varılabilir (Budiman ve ark., 2016).

Michael Hahsler ve Mattew Piekenbrock, yoğunluk tabanlı kümeleme algoritması DBSCAN ve artımlı sıralama algoritması veya OPTICS’in eksiksiz ve hızlı bir uygulamasını belirtmektedir. Diğer uygulamalarla karşılaştırıldığında DBSCAN, hesaplamayı hızlandırmada C++ ve gelişmiş veri yapılarının açık kaynak kullanımını sunar. Bu uygulamanın önemli bir avantajı, orijinal yayınından bu yana eklenen birkaç önemli ilerleme ile güncel olmasıdır, bunlar arasında anomali düzeltmeleri ve OPTICS için dendrogram (hiyerarşik ağaç diyagramı) çıkarma yöntemleri yer alır. *Fuzzy Partition Clustering* (FPC), *Environment for Developing KDD-Asslications Sussorted by Index-Structures* (ELKI), *Waikato Environment for Knowledge Analysis* (WEKA), PyClustering, SciKit-Learn ve SPMF gibi diğer kütüphanelerle karşılaştırıldığında DBSCAN ve OPTICS uygulamaları ile yapılan deneyler, DBSCAN’ın çok verimli bir uygulama sağladığını göstermektedir (Hahsler ve ark., 2019).

Xu Gao ve ark 2022 yılında K-Means kümeleme algoritmasının basitlik ve verimli çalışma avantajları nedeniyle yaygın olarak kullanıldığını açıklamıştır. Bununla birlikte, algoritmadaki kümeleme merkezlerinin eksikliği genellikle bazı ayırık noktaların yanlış kategorize edilmesine yol açmaktadır. Bu araştırma, DBSCAN ile birleştirilmiş K-Means algoritmasını geliştirmeyi amaçlamaktadır. Bu çalışmanın sonuçları, DBSCAN ve K-Means algoritmalarının bir kombinasyonunu kullanarak kümeleme sonuçlarının, kümeleme sonuçlarının doğruluğunda ve etkileyen faktörlerin hızında ve kümeleme etkisinde net bir iyileşme ile sonuçlandığını göstermektedir (Gao, 2022).

Nafuri ve ark, araştırmasında B40 öğrenci verilerine dayalı olarak K-Means, *Balanced Iterative Reducing and Clustering using Hierarchies* (BIRCH) ve DBSCAN adlı denetimsiz modelleri kullanmıştır. Sonuçlar, B modelinde optimize edilmiş K-Means’in tüm modeller arasında en yüksek performansı elde ettiğini göstermiştir. KmoB,

performanslarına göre beş grup B40 öğrencisi üretir, böylece üniversitede okulu bırakma oranını azaltmada hükümete yardımcı olabilir ve öğrencilerin sosyo-ekonomik durumunu kesinlikle iyileştirecektir (Nafuri ve ark., 2022).

Songfeng Sun 2023 yılında, kullanılan iki algoritmanın avantaj ve dezavantajları ile birlikte iki algoritmayı analiz etmeyi ve iki algoritma arasındaki farkları test etmeyi amaçlayan ve aynı 3 durumda kullanan bir çalışma yapmıştır. Bu çalışmanın sonuçları, DBSCAN'ın belirsizlik/gürültü noktalarını tanımlayabildiğini, küme verilerindeki varyasyonları işleyebildiğini, K-Means'in ise verimli, hızlı ve basit bir işlem algoritması olduğunu ancak küme türlerini tanımlayamadığını ve belirsizlik/gürültü noktalarına duyarlı olmadığını belirtmektedir (Sun ve ark., 2023).

C. Kondal Raj araştırmasında gen ifadesi verileri için çeşitli kümeleme yaklaşımlarını incelemeyi amaçlamaktadır. Bu çalışmanın sonuçları, veri noktası sayısına ve her bir algoritmanın hesaplama süresine dayanmaktadır. Veri noktalarının sayısı, veri noktalarının keyfi şeklinin dağılımına göre algoritma tarafından kümelendir. Algoritmanın performansı, toplamda 125.000 noktadan oluşan bir girdi ile farklı veri noktaları dikkate alınarak çeşitli yürütmelerde analiz edilmiştir. Bu araştırmanın sonucu, DBSCAN algoritmasının K-Means ve K-Medoids algoritmalarına kıyasla hesaplama süresi performansının daha düşük olduğu, ancak verimlilik açısından DBSCAN algoritmasının özellikle büyük ölçekli veri tabanlarında rastgele kümelere sahip veriler için diğer iki algoritmadan daha iyi olduğudur (C. Kondal raj, 2017).

Erwin Lanceta C ve ark araştırmasında, yeni bir öğrenci kabul veri setine dayalı olarak öğrencileri kümelemek için veri madenciliğinin kullanımını sunmaktadır. Optimum küme sayısı Silhoutte yöntemi kullanılarak belirlenmiş ve bu da k-ortalamalar ve hiyerarşik kümeleme algoritmaları kullanılarak gerçek kümelemede kullanılan optimum $k=2$ sayısı ile sonuçlanmıştır. Gürültü kümeleme (DBSCAN) aynı veri kümesinde kullanılmış olup, bu araştırma, öğrenci kayıt verilerinin en az 5 yıllık bir veri kümesi kullanılarak, öğrenci tutma ve okul sınavlarına geçişi önerecek diğer algoritmalar kullanılarak çoğaltılabilir (Cahapin ve ark., 2023).

2022 yılında Ziqing Wang ve arkadaşları, geleneksel DBSCAN algoritmasının çok yoğunluklu (multi-density) veri setlerinde iyi kümeleme sonuçları veremediğini belirtmiştir. Bunun üzerine Wang, geleneksel DBSCAN algoritmasının bir varyasyonu olan Adaptive Multi-Density DBSCAN (AMD-DBSCAN) algoritmasını geliştirmiştir. Yapılan araştırmaya göre, AMD-DBSCAN'ın sunduğu adaptasyon yöntemi, algoritmanın temel parametreleri olan Eps ve MinPts'yi belirlemede önemli bir rol

oynamaktadır. Deney sonuçları, AMD-DBSCAN'ın algoritmanın daha düşük karmaşıklığı sayesinde ortalama yürütme süresini %75 oranında azalttığını göstermiştir. Bunun yanı sıra, AMD-DBSCAN, yoğunluğu büyük ölçüde değişen çok yoğunluklu veri setlerinde mevcut en iyi tasarımlara kıyasla ortalama doğruluğu %24,7 oranında artırmıştır. Bu iyileştirmeler, tek yoğunluklu (single-density) senaryolarda performans kaybı olmaksızın elde edilmiştir.

2023 yılında Mafalda Ines Oliveira ve Andre R. S. Marçal, güvenli sürüşü sağlamak amacıyla otonom sürüş sistemlerinde çoklu nesne tespiti gerçekleştirmiştir. Elde edilen sonuçlar, K-Means algoritmasının iç doğrulama endekslerinden siluet, C-Index ve Calinski-Harabasz'ta DBSCAN'a kıyasla daha iyi performans gösterdiğini; öte yandan, DBSCAN algoritmasının Dunn ve Davies-Bouldin endekslerinde K-Means'ten daha iyi performans sergilediğini ortaya koymuştur. Dış doğrulama endeksi ise, DBSCAN'ın yoğunluk tabanlı kümeleme yaklaşımının *Light Detection and Ranging* (LiDAR) segmentasyonu için uygun olduğunu destekleyen en iyi sonuçları verdiğini göstermektedir (Oliveira ve Marcal, 2023).

2.3.1. K-Means Algoritmasını Kullanan Çalışmalar

2010 yılında Oyelade ve ark., öğrenci performansına dayalı öğrenci puanları grupları elde etmek için okul çocuklarının verilerini kullanarak K-Means üzerine bir araştırma yapmıştır. Bu çalışmada K-Means algoritmasının kullanılması, öğretmenler tarafından gelecekteki öğrenme planlarının yapılmasında kullanılabilecek her öğrencinin performansını izlemeye iyi ölçütlerle sonuçlar sağlar (Oyelade ve ark., 2010).

Januardi Nasir tarafından yürütülen araştırma, kaç kitap ödünç alındığını bilmedikleri için her zaman zorluk yaşayan kütüphane sorunlarının üstesinden gelmek için K-Means kullanarak kümeleme araştırması yaptı. Elde edilen araştırma sonuçlarına göre 2. Küme en çok ödünç kitap alınan kümedir. Bu araştırma aynı zamanda K-Means yönteminin kütüphaneye sıklıkla ödünç alınan kitapların belirlenmesinde yardımcı olabileceğini belirtmektedir (Nasir, 2020).

K-Means uygulaması, Gözde ve ark tarafından 2023 yılında *Recency, Frequency, Monetary* (RFM) modeli kullanılarak müşteri bölümlendirmesinin belirlenmesinde kullanılmıştır. Bu çalışmada, K-Means, müşterilerin benzerlik özelliklerine göre kümelendiği ev uygulamasının RFM yönlerini temsil eden özellikleri çıkararak RFM modeline uygulanmıştır. Bu Kümeleme işlemi, siluet (*Silhouette*) puanı için istenen eşiği karşılamış ve ortaya çıkan kümeler, müşterilerin işletme için değerini değerlendiren

Customer Lifetime Value (CLV) metriğine göre sıralanmış ve adlandırılmıştır (Aslantaş ve ark., 2023).

Ahmad ve ark, K-Means kullanarak Endonezya illerini nüfus yoğunluğu, okullaşma, insani gelişme endeksi ve açık işsizlik oranına göre bölümlere ayırmıştır. Analiz sonucunda Güney Sumatra, Lampung, DKI Jakarta, Orta Java ve Batı Kalimantan merkezli beş ana küme belirlenmiştir (Ahmar ve ark., 2018).

2022’de yazılan bir araştırma raporunda, Alzahar Üniversitesi, tanıtım segmentasyonunu yönetmek için kümeler oluşturmada K-Means kümelemesini kullanarak öğrenci kabul sayısını artırmaya çalışmıştır. Bu çalışmada, ana küme, minimum küme ve uyarlanabilir küme olmak üzere 3 küme üretilmiştir (Arsyad ve ark, 2022).

Event in 2021, satış seviyesine göre ölçüm ve gruplandırma yaparak kümeleme yöntemiyle ürün bulunabilirliğinin sağlanması ve son kullanma tarihi geçmiş ürünlerden kaynaklanan kayıpların önlenmesinde otel müşteri memnuniyeti yönetimi üzerine bir araştırma yapmıştır. Bu araştırma, Newton otel yönetimine yönetimini optimize etmede bilgi sağlayan 3 küme ile sonuçlanmıştır (Event, 2021).

2023 yılında Ling Zhang, büyük ölçekli verilerin (Büyük veri) kümelenmesinde K-Means algoritmasını optimize etmiştir. Bu araştırma, MapReduce’un K-Means algoritması üzerinde kullanılmasının onu büyük veri kapsamında daha etkili hale getirebileceğini kanıtlamaktadır. Bu yöntemin, özellikle daha fazla sayıda özellik ile verileri daha iyi kümelemede başarılı olduğu kanıtlanmıştır. Map Reduce, paralel programlamanın uygulama detaylarını basitleştirerek bilgisayar kümeleri üzerinde kolay ve güvenilir büyük veri işlemeyi mümkün kılar. Bu küme analizi, istatistik, görüntü analizi, doku algılama ve biyotıp gibi çeşitli uygulamalardaki dağılım modellerini, veri değişikliklerini ve gelişim eğilimlerini ortaya çıkarabilir (Zhang, 2023).

Onofrio ve arkadaşları tarafından 2017 yılında yapılan bir çalışmada, öğrencilerin özellikle termal bilim alanında tanımlayıcı modeller ile açıklayıcı modelleri ayırt etmekte zorlandıkları görülmüştür. Bu çalışma, Fenynman’ın bütünsel yaklaşımının öğrencilerin açıklayıcı becerilerini ve termal bilimi geliştirmelerine etkili bir şekilde yardımcı olduğu sonucuna varmıştır. Öğrencilerin akıl yürütme biçimlerinde K-Means analizi ile elde edilen değişimler, 20 saatin öğrencilerin bilimsel süreci açıklayan fan açıklayıcı modelleri anlama ve uygulama becerilerini geliştirmede önemli bir rol oynadığını göstermiştir (Battaglia ve ark, 2017).

Yihan Hu'nun 2023 yılındaki bilimsel yayınında, büyük verileri işlemek için K-Means algoritmasında küme merkezlerini başlatma verimliliğini artırmaya yönelik bir araştırma yapılmıştır. Bu araştırma, veri dağılımına dayalı olarak başlangıç merkezlerini seçmek için bölgeler arası veri değerleri için dinamik bir küme algoritması kullanan yeni bir yöntem önermektedir. Bu, küme merkezlerinin yüksek yoğunluklu alanlarda bulunmasını sağlayarak aşırı yoğunlaşmayı önler. Bu araştırma aynı zamanda yöntemi geleneksel K-Means ve K-Means++ ile karşılaştıran simülasyon deneyleri de gerçekleştirmiştir. Deney sonuçları, K-Means++'dan 0,1249 ve K-Means'den 0,4903 daha yüksek Siluet Skoru ile müşteri pazar analizi verilerinin kümelenebilmesinde daha iyi performans göstermektedir. Bu, geleneksel yöntemlere göre önemli bir avantaj göstermektedir (Hu, 2023).

2.3.2. DBSCAN Algoritmasını Kullanan Çalışmalar

Dwina ve ark tarafından 2022 yılında yapılan araştırma, burs almaya hak kazanan öğrencilerin seçimine ilişkin bir araştırma gerçekleştirmiştir. Bu çalışmada kümelemede not ortalaması kriteri, akademik olmayan başarı, bakmakla yükümlü olunan kişi sayısı ve ebeveynlerin geliri olmak üzere 4 parametre kullanılmıştır. Bu araştırma, farklı yoğunluklarda daha doğru kümeleme yapabilen DBSCAN algoritmasını uygulamaktadır. Bu araştırma tarafından üretilen çıktı, D, F, G, H, I ve J alternatifleri olmak üzere I. kümede vakıf bursu almaya hak kazanan öğrencilerdir (Indini ve ark., 2022).

Hui ve ark tarafından 2021 yılında yazılan araştırma, öğrencileri sınıflandırmak ve farklı öğrenci türlerine uygun öğretim stratejileri geliştirmek için DBSCAN kullanarak araştırma yaptı. İlk olarak, kariyer yönelimi, öz denetim endeksi ve bilgi düzeyi dahil olmak üzere öğrenci özelliklerini veri olarak çıkarın. Ardından, öğrencileri kümelemek için DBSCAN algoritmasını kullanmıştır. Öğrencilerin öğrenme durumlarını anlamak için DBSCAN algoritması DBSCAN kümeleme algoritmasının kullanılmasının öğretim faaliyetleri için belirli bir yol gösterici öneme sahip olduğu doğrulanmıştır (Du ve ark, 2021).

2022 yılında Doruva ve arkadaşları, yeni çalışanların şirkete adaptasyon sürecini dijitalleştirmek için DBSCAN uygulamıştır. Temel amaç, makine öğrenimi algoritmalarını, özellikle de modelin kalitesini artırmak için uzmanların varlığını gerektirmeyen denetimsiz bir yöntem olan DBSCAN kümeleme yöntemini kullanarak bu süreci daha çekici hale getirmektir. Verileri yönetmek için ilgili veri tabanı yapıları ve web sayfaları geliştirdik ve proje uygulaması için finansal analizler sunduk. Bu

yaklaşımın etkili, uygulaması kolay ve belirli bir konu alanına bağlı kalmadan geniş bir şirket yelpazesinde kullanılabilecek kadar esnek olduğu kanıtlanmıştır (Durova ve ark., 2022).

2019 yılında Jennifer ve Heinrich, klasik DBSCAN algoritmasının geliştirilmiş bir versiyonu olan DBSCAN++ ile klasik yoğunluk tabanlı kümeleme gerçekleştirmiştir. Bu araştırma, geleneksel DBSCAN'ın her veri noktasında ampirik yoğunlukların hesaplanmasını gerektirdiğini ve bunun da ikinci dereceden zaman karmaşıklığına yol açtığını ve DBSCAN++'ın yalnızca seçilen bir nokta alt kümesi için yoğunlukları hesaplayarak bu yükü azaltabileceğini belirtmektedir. Bu araştırma, DBSCAN++'ın istatistiksel tutarlılığı koruduğunu, hesaplama maliyeti ile tahmin doğruluğu arasında bir denge sunduğunu ve böylece küme düzeyinde tahmin için minimum optimum seviyeye ulaştığını kanıtlamaktadır (Jang ve Jiang, 2018).

Gloru, 2012 yılında DBSCAN algoritmasını büyük ölçekli veri birikimi oluşturma teknikleri üzerinde geliştirmiştir ve bu geliştirme çeşitli alanlarla ilişkilidir. Bu araştırma, DBSCAN kümeleme algoritmasını iyileştirmek için bir yaklaşım düşünülmesini önermektedir. Kümeleme işlemi gerçekleştirilmeden önce, veri seti eşik değerine göre bölünmüştür. Ayrıca, önerilen algoritma sentetik veri setlerinde test edilmiştir. Testler, düşük boyutlu veri setlerinden yüksek boyutlu veri setlerine kadar tüm veri seti türlerini dikkate alarak gerçekleştirilmiştir. Öklid Mesafesi'nin (*Euclidean Distance*) düşük boyutlu veri setlerinde iyi çalıştığı, ancak yüksek boyutlu veri setlerinde iyi performans göstermediği bulunmuştur. Ayrıca, Euclidean mesafesi daha fazla küme oluşturmaya rağmen, hatalı küme sayısının Manhattan mesafesi (diğer bir ifadeyle *Taxicab Distance* veya *City Block Distance*, iki nokta arasındaki mesafeyi ızgara üzerindeki eksen boyunca koordinatların mutlak farklarını toplayarak hesaplayan bir ölçüm) kullanıldığında daha az olduğu tespit edilmiştir (Trivusi, 2022). Gürültü oranı, Manhattan mesafesi kullanıldığında Euclidean'a göre daha yüksek kalmaktadır. Ancak, Euclidean mesafesi kullanıldığında algoritmanın küme oluşturma süresi, Manhattan mesafesine kıyasla daha uzundur. Araştırmacılar, giriş parametreleri olan Eps ve MinPts'yi belirlemek için bazı yaklaşımlar geliştirerek bu değerlerin daha doğru bir şekilde hesaplanabileceğini umut etmektedir. Ayrıca, bazı çekirdek noktaların kaybolması nesnelerin kaybına neden olabileceğinden, bu sorunun da çözülmesi gerektiği belirtilmiştir (Shah, 2012).

2020 yılında Mingkung ve Huncheng, negatif veri tabanına dayalı gizlilik korumasında DBSCAN kümelemesi gerçekleştirmiştir. Bu araştırma, yoğunluk kümeleme analizi temelinde veri gizliliğini korumak için tasarlanmıştır. Araştırmacılar,

orijinal verileri sert negatif verilere dönüştürmek ve veri gizliliği korumasını gerçekleştirmek için K-Hidden negatif veri oluşturma algoritmasını kullanmaktadır. Ardından, negatif veri kümelerinin yoğunluk kümelemesini gerçekleştirmek için DBSCAN algoritmasındaki iki nesne arasındaki mesafeyi hesaplamak için negatif verilerin Öklid mesafe formülü kullanılır. Deneysel sonuçlar, gizliliği koruyan DBSCAN algoritmamızın yüksek kümeleme doğruluğuna ulaşabildiğini ve uygun parametrelerin seçilmesi temelinde iyi kümeleme etkisi elde edebildiğini göstermektedir. Orijinal algoritma ile karşılaştırıldığında, kümeleme doğruluğu yalnızca birkaç yüzde puanı daha düşüktür ve bu da orijinal algoritmaya çok benzerdir. Gelecekteki çalışmalarda, negatif veri tabanı üzerinde diğer mesafe ölçüm formüllerini zorlayacağız, gizlilik koruma DBSCAN algoritmamızı geliştireceğiz ve gizlilik koruma DBSCAN algoritmasının kümeleme verimliliğini ve doğruluğunu artırmak için negatif veri tabanı üzerinde DBSCAN algoritmasındaki Eps ve MinPts parametrelerinin otomatik seçimi incelenmiştir (M. Zhang ve Liao, 2020).

2.3.3. Hiyerarşi Algoritmasını Kullanan Çalışmalar

2012 yılında Fionn ve Pedro, hiyerarşik algoritması ile ilgili bir tartışma yürütmüş ve 60'lı ve 70'li yıllarda ortaya çıkmış olmasına rağmen kümelemedeki hiyerarşik algoritmanın çeşitli alanlardaki çeşitli sorunların ve yeni zorlukların üstesinden gelmede hala geçerli olduğunu kanıtlamıştır. Hiyerarşik yaklaşımın sonraki algoritmalar için temel teknikler sağladığı belirtilmiştir (Murtagh ve Contreras, 2012).

Nisha ve Puneet tarafından 2015 yılında yapılan araştırmada hiyerarşik kümeleme kullanılmıştır. Bu çalışma, daha iyi kümelerin test edilmesinde uyum ölçümleri, siluet indeksleri ve geçen süre olmak üzere üç parametre kullanmaktadır. Bu çalışma, kümelemede hiyerarşik algoritma kullanımının, belirli bir veri kümesinden kümelerin değerlendirilmesinde aktif bir yöntem olduğunu, çünkü bu yöntemin bir dizi kümeyi bir ağaç şeklinde birleştirebileceğini belirtmektedir. Bu çalışmanın sonuçları, yüksek uyum, sıfır ve pozitif siluet indeks değerleri ve düşük geçen süreye sahip kümelerin her zaman iyi küme kalitesi ürettiğini göstermektedir. Elde edilen sonuçlar bu üç parametre arasında bir ilişki olduğunu göstermektedir. Bununla birlikte, bu araştırmada benimsenen yaklaşım, daha genel sonuçlara varmak için çeşitli nesne yönelimli sistemler için kullanılabilir (Nisha ve Kaur, 2015).

2019 yılında Andrea ve arkadaşları, Doğu Kalimantan'daki ilçe/shehirlerin kümeleme sürecinde en iyi benzerlik ölçümünün, birimlerin otokorelasyon değerlerine

dayalı mesafeyi hesaplayan autocorrelation based distance olduğunu, en uygun algoritmanın ise kümeler arasındaki ortalama mesafeyi temel alan average linkage yöntemi olduğunu belirlemiştir. Kümeleme sonuçlarının geçerliliği, dendrogramdaki mesafelerin veri yapısını ne kadar iyi temsil ettiğini ölçen cophenetic korelasyon katsayısı ile değerlendirilmiş ve 0,99 gibi yüksek bir değer elde edilmiştir. Ayrıca, temsil için belirlenen 2 kümenin netliğini ölçen siluet katsayısı 0,68 olarak bulunmuş, bu da kümelerin iyi ayrıştığını göstermiştir (Dani ve ark., 2019).

Xingcheng ve arkadaşları, 2023 yılında kümeleme ve son gelişmeler üzerine kapsamlı bir araştırma yapmıştır. Bu araştırma, bölücü ve yığılmalı kümeleme olmak üzere iki hiyerarşik kümeleme yöntemi olduğunu ve her ikisinin de temel sorun özelliklerine sahip olduğunu göstermektedir. Bu çalışma, bölücü kümelemedeki ana sorunun, benzerliğe dayalı sonraki ayırma prosedüründe kümelerin nasıl seçileceği ve seçilen kümelerin nasıl bölüneceği olduğunu söylemektedir. Yığılmalı kümelemede ise, sonraki birleştirme için en benzer iki küme tarafından kullanılan benzerlik ölçüsüdür. Aglomeratif (aşamalı) kümelemede, sonraki birleştirme için en benzer iki küme tarafından kullanılan benzerlik ölçüsüdür. Her iki algoritma da çıktı olarak bir veri dendrogramı (öbek ağacı) üretir. Kümeleme sonuçları, kümelemede kullanılan farklılık veya benzerlik ölçüsüne ve veri türüne büyük ölçüde bağlıdır (Ran ve ark., 2023).

Bu araştırma, 2021-2023 dönemi için Güneydoğu Asya'daki deprem verilerini *Agglomerative Hierarchical Clustering* (AHC) kullanarak kümelemektedir. Kümeleme süreci, yeni öznitelikler oluşturmak ve deprem türlerini tanımlamak amacıyla her aşamada üç küme üretmektedir. Sonuçlar, Bölge 0'ın yüksek deprem sıklığına sahip bir bölge olduğunu göstermektedir. Bu bulgular, deprem tahmini ve daha iyi afet azaltma stratejilerinin geliştirilmesi de dahil olmak üzere daha ileri araştırmalar için kullanılabilir (Arifin, 2023).

3. MATERYAL VE YÖNTEM

Kümeleme algoritmalarının performansı hakkında bilgi edinmek için kümeleme algoritmalarının büyük veri üzerinde kullanımı bu tezin konusudur. Bu tezde, büyük veriye uygulanan çeşitli kümeleme algoritmaları, avantaj ve dezavantajlarını belirlemek için analiz edilmiştir. Algoritmaların performansı, veri boyutu, boyutluluğu ve dağılımı gibi faktörler göz önünde bulundurularak değerlendirilmektedir. Araştırma süreci ayrıca çeşitli alanlardaki algoritma tercihlerini, yapılandırma parametrelerinin etkisini ve büyük veri bağlamında hesaplama verimliliği ve bellek yönetimini analiz etmeyi de içermektedir.

Bu tez iki ana noktaya odaklanmaktadır. Birincisi, kümeleme algoritmalarının işleyişini ve performans farklılıklarını büyük veri bağlamında değerlendirmek. İkinci olarak, çeşitli alanlarda kullanılan algoritmaların karşılaştırmalı bir analizinin yapılması ve ilgili literatürdeki en son yöntemlerin gözden geçirilmesi.

3.1. Büyük Veriler

Büyük veri olgusu günümüzde benzersizdir çünkü insanların yaptığı, söylediği veya gözlemlediği her şey veri oluşturmaktadır (Hoy, 2014). Şu anda, büyük veri teknolojisi hem büyük hem de küçük şirketlerin ve hükümetlerin her alanında yaygın olarak kullanılmaktadır (Sirait, 2016).

Büyük veri terimi 2005 yılından sonra O'Reilly Media tarafından ortaya atılmıştır. Büyük veri, gerçek dünyayı, insanları ve sanal dünyayı bütünleştiren bir sistem olarak tanımlanmaktadır (Philip Chen ve Zhang, 2014). Gartner BT Sözlüğü'nde büyük veri, yüksek hacimli, yüksek hızlı bilgi varlığı ve yenilikçi ve uygun maliyetli bilgi işleme ve etkinleştirmenin çeşitli biçimleridir (Gartner, 2024).

Yukarıdaki tanımların bazılarında büyük verinin, belirli amaçlar için değerli bilgiler üretmek üzere toplanan ve işlenen büyük miktarda yapılandırılmış, yarı yapılandırılmış ve yapılandırılmamış verilerden oluşan bir dijital veri kümesi olduğu sonucuna varılabilir. Nesnelerin interneti aracılığıyla sanal dünyaya yansıyan sosyal gerçeklikle ilgili gerçek dünya, teknoloji, bilgisayarlar, yapay zekâ ve mobil internet aracılığıyla sanal dünya tarafından üretilen büyük veriyi üretmektedir. Bu durum büyük veriyi iki kategoriye ayırmaktadır: gerçek dünyadan kaynaklanan veri; bu veri doğa, iklim, haritalar ve diğerleri gibi dolaşımda olan her türlü veriyi yakalayan teknoloji aracılığıyla elde edilmektedir. İkincisi ise insan kaynaklı veriler olup bu veriler sosyal

medya, internet ve diğer teknolojiler gibi nesnelerin internetinden elde edilmektedir (Salahudin, 2019). DOMO'nun yakın tarihli bir raporu, popüler çevrimiçi platformlarda her dakika üretilen veri miktarını aşağıdaki gibi tahmin etmektedir (Domo, 2024):

- 1) E-posta kullanıcıları 231,4 Milyon mesaj gönderdi.
- 2) Google kullanıcıları 5,9 Milyon arama yaptı.
- 3) YouTube kullanıcıları 500 saatlik yeni video içeriği yükledi.
- 4) Amazon kullanıcıları 443.000 dolarlık işlem yaptı.
- 5) Instagram kullanıcıları 66.000 fotoğraf paylaştı.
- 6) X kullanıcıları 347.200 tweet paylaştı.
- 7) Kripto kullanıcıları 90,2 milyon dolarlık kripto para birimi satın aldı.

Yukarıdaki veriler, dijital çağda bilginin hızlı büyümesini gösteren çeşitli çevrimiçi platformlardan her dakika üretilmektedir. Böylesine büyük bir hacimde, depolamayı yönetmek ve giderek karmaşılaşan verileri analiz etmek için yapılması gereken zorluklar kesinlikle olacaktır. Büyük verinin çeşitli şekillerde tanımlanmasında, büyük veri olduğu söylenebilecek verilerin özellikleri vardır. Başlangıçta, büyük veri aşağıdaki boyutlar 3V modeli ile karakterize edilmiştir (Cholissodin ve Riyandani, 2016):

- 1) **Hacim:** Hacim, terabaytlardan petabaytlara kadar hızla artabilen üretilen ve toplanan veri miktarını ifade eder. Büyük veri terimi, geleneksel veritabanı ve veri işleme mimarileri kullanılarak depolanması, yönetilmesi ve işlenmesi zor olan büyük ölçekli verileri ifade eder.
- 2) **Hız:** Büyük verideki hız, verilerin gerçek zamanlı olarak oluşturulma hızını ifade eder. Veri hızı, verinin ne kadar hızlı üretildiğini ifade eder.
- 3) **Çeşitlilik:** Büyük veri modeli, çeşitli kaynaklardan toplanan yapılandırılmış, yapılandırılmamış ve yarı yapılandırılmış verilerden oluşur. Şuanda veriler e-postalar, PDF'ler, fotoğraflar, videolar, sesler ve diğerleri gibi çeşitli biçimlerde gelmektedir. Bununla birlikte, teknolojinin ve veri analizinin gelişmesiyle birlikte, veri özellikleri aşağıdaki eklemelerle 5V haline gelmiştir:
- 4) **Doğruluk:** Bu noktadaki özellikler, verinin geçerliliğine güvenilip güvenilemeyeceği ile ilgilidir. Verilerin çok çeşitli olması, bilgi hakkındaki verilerin doğruluğunun daha az kontrol edilmesi için kesinlikle bir zorluktur. Bunun nedeni, veri hacmi ne kadar büyük olursa, verilerin o kadar az doğru olacaktır.

- 5) **Değer:** Değer, veriden değer çıkarabilen verinin kullanışlılığını ifade eder. Bu veri değeri aynı zamanda verinin doğruluğu veya kesinliği ile de ilgili olacaktır (Santoso, 2020).

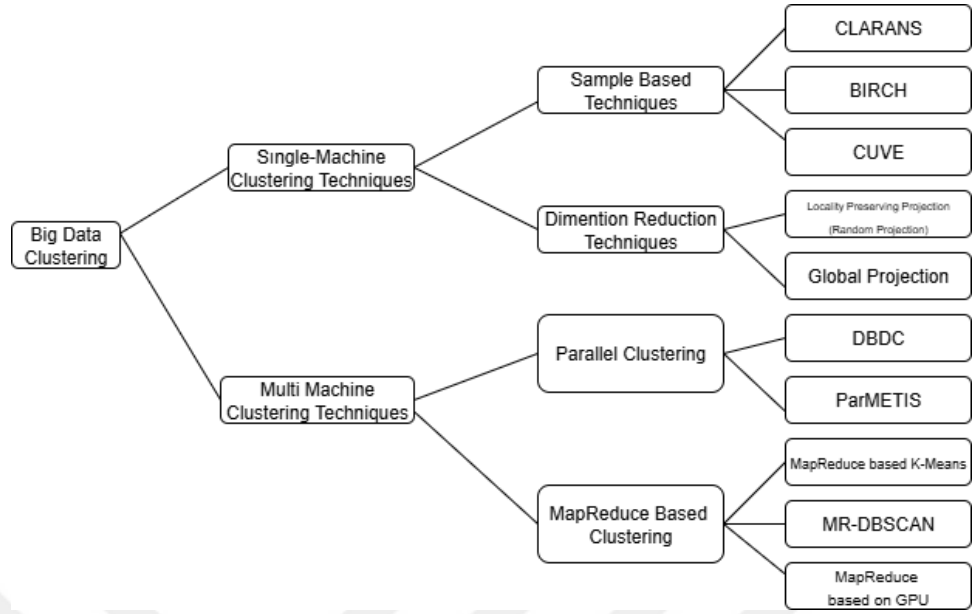
3.2. Kümeleme

Kümeleme, verileri mümkün olan en küçük varyasyona sahip benzer veri gruplarına ve gruplar arasında mümkün olan en büyük veri varyasyonuna bölmeyi amaçlayan veri keşif tekniklerinden biridir (Wierzchoń ve Kłopotek, 2018). Küme, temelde küme analizinin gruplar üreteceği bir grup olarak yorumlanabilir. Bu kümeleme, belirli etiketlenmiş bilgilere sahip olmadığı için üretken bir model olarak kabul edilir (Aggarwal ve Reddy, 2014). Kümeleme aynı zamanda, nesnelerin insanlar, olaylar ve diğerleri olabileceği ve daha sonra kümeler arasında birbirine bağlı birkaç seviyeye sahip olacak şekilde gruplara dağıtıldığı bir dizi veri grubunu kendi özellikleriyle ayırmak için çalışarak segmentasyon olarak da adlandırılır (A ve ark., 2015).

Kümeleme tekniği sınıflandırmadan farklıdır çünkü kümelemede hedef değişken/etiket yoktur. Kümeleme, hedef değerden veya değişkenden tahminler yapmaya çalışmaz, ancak verileri benzer bir gruba sahip olan her gruba böler. İş dünyasında ve araştırmada kümeleme örnekleri şunlardır:

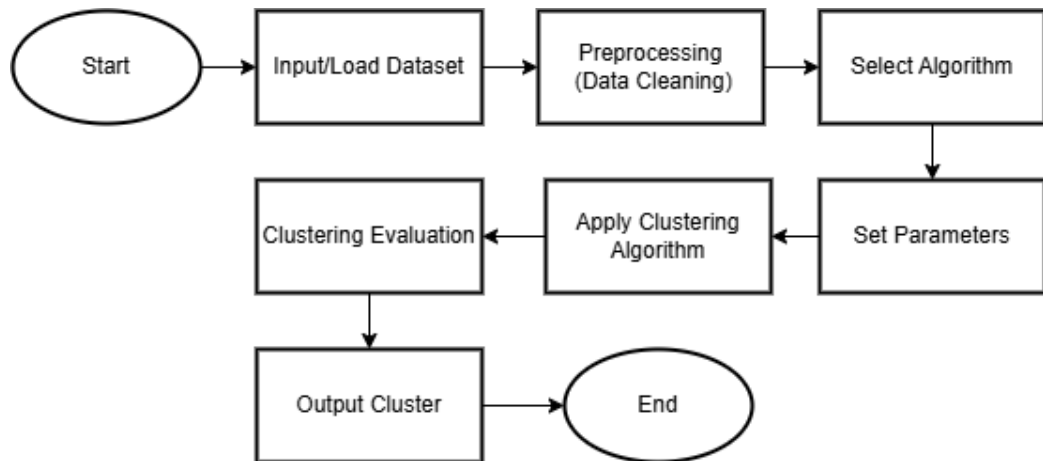
- 1) Ürün önerilerini ve pazarlama kampanyalarını kişiselleştirmek için müşterilerinin alışveriş modellerini anlamada E-Ticaret müşteri kümeleme teknikleri
- 2) Genler ve belirli hastalıklar arasındaki ilişkiyi keşfetmede genetik örnekleri kümelemek için genetik analiz
- 3) Turizm sektöründe müşteri tercihlerine uygun tur paketleri sunmak amacıyla pazar segmentasyonu uygulamaları.

Büyük veride kümeleme teknikleri iki kategoriye ayrılır: tek makineli kümeleme tekniği ve çok makineli kümeleme tekniği (Suganya ve ark., 2018). İki kümeleme türünün hiyerarşisi Şekil 3.1’de gösterilmektedir (Shirkhorshidi ve ark., 2014):



Şekil 3.1. Büyük veri kümeleme algoritmasının görüntü sınıflandırması.

Yukarıdaki hiyerarşi, tek makineli kümeleme algoritmaları ve çok makineli kümeleme algoritmaları olmak üzere iki kategoriye ayrılan büyük veri kümelemesini açıklamaktadır. Son yıllarda çoklu makine kümeleme algoritmaları, tekli makine kümeleme algoritmalarına göre daha fazla ilgi görmektedir. Bunun nedeni, tek makineli algoritmaların yürütme süresinin daha büyük veri boyutlarını işleme yeteneğinin daha az olmasıdır. Şekil 3.2’de gösterildiği gibi, tekli ve çoklu makine algoritmaları farklı tekniklerden oluşmaktadır.



Şekil 3.2. Genel olarak kümeleme akışının akış şeması.

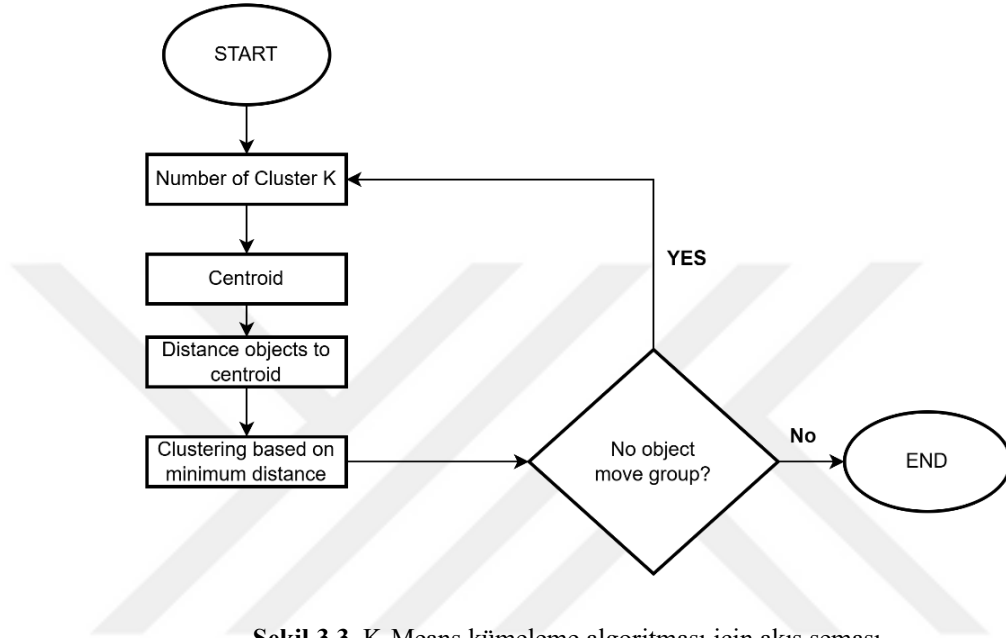
Örnekleme teknikleri veya boyut azaltma teknikleri içeren tek makineli kümeleme ve paralel kümeleme veya MapReduce tabanlı kümeleme içeren çift makineli kümeleme algoritması (L. Zhang, 2023):

- 1) Tek Makine Kümeleme, aşağıda ana hatlarıyla belirtildiği gibi örnekleme ve boyut azaltma yöntemlerini içerir. Bu algoritmalar, kümeleme görevlerinde üstel uzay karmaşıklığının yarattığı zorlukların üstesinden gelmek amacıyla performansı ve ölçeklenebilirliği artırmak için tasarlanmış ilk kümeleme algoritmaları kümesini temsil etmektedir. Bu algoritmalar, kümeleme işlemini veri kümesinin bir alt kümesi üzerinde gerçekleştirdikleri ve daha sonra bulguları tüm veri kümesi üzerinde kümeleme yapmak yerine tüm veri kümesine genelleştirdikleri için örnekleme yöntemleri olarak kategorize edilir. Bu yaklaşım, hesaplama daha küçük veri alt kümeleri üzerinde gerçekleştirildiği için algoritmaları daha verimli hale getirmekte ve böylece işlem süresi ve alanının karmaşıklığını azaltmaktadır.
- 2) Çok makineli kümeleme teknikleri, verilerin birden fazla makine kullanılarak analiz edildiği kümeleme yöntemleridir. Bu, yürütme süresini azaltmaya ve oluşturulan kümelerin hızını artırmaya olanak tanır. Çok makineli kümeleme teknikleri paralel kümeleme ve harita indirgenmiş tabanlı kümeleme olarak ikiye ayrılır. Bu tür bir teknik, verilerin birkaç sunucuya bölünmesine ve isteklerin işlenmesine izin verir (Shirchorshidi ve ark., 2014).

Her iki algoritmanın tek bir makine ve birden fazla makine kullanılarak uygulanmasında, kümelemenin genel akışı Şekil 3.2'deki gibi tanımlanabilir. Süreç veri toplama ile başlar. Daha sonra veriler işlenebilmesi için temizlenmek ve normalize edilmek üzere ön işleme aşamasına girer. Bundan sonra, uygulama seçilen algoritmaya dayalı olarak gerçekleştirilir, bazı algoritmalar kesinlikle küme sayısı gibi başlangıç parametrelerini belirlemek için başlatma gerektirir. Ayrıca, algoritma kümeleme yürütme aşamasında çalıştırılmıştır. Kümeleme sonuçları daha sonra, küme kalitesinin değerlendirme metrikleri kullanılarak kontrol edildiği Sonuç Değerlendirme adımıyla değerlendirilir. Gerekirse, ek Küme Doğrulaması gerçekleştirilebilir. Tüm adımlar tamamlandıktan sonra, kümeleme işleminin nihai sonucu Nihai Çıktı adımıyla oluşturulur ve işlem sona erer.

3.2.1. K-Means Kümeleme

K-Means denetimsiz öğrenmeye aittir (Sun ve ark., 2023). K-Means, veri öğelerini K kümeye gruplayan ve her gözlemi kareler toplamını en aza indiren bir kümeye yerleştiren bir kümeleme yöntemidir (Syakur ve ark., 2018). K-Means Algoritmasının iş akışının bir illüstrasyonu Şekil 3.3'te gösterilmektedir.



Şekil 3.3. K-Means kümeleme algoritması için akış şeması.

Yukarıdaki Şekil 3.3, ilk girdi olarak istenen küme sayısının K belirlenmesiyle başlayan K-Means kümeleme algoritması sürecini göstermektedir. Bundan sonra, her bir küme için ilk merkez noktası, verilerden rastgele bir değer seçilerek başlatılır. Ardından, her bir veri kümesi nesnesinin her bir merkez noktasına göre uzaklığı, metrik olarak Öklid uzaklığı kullanılarak hesaplanır. Bu hesaplama dayanarak, her grup daha sonra en yakın mesafeye sahip merkez noktasına sahip kümeler halinde toplanacaktır. Kümelemeden sonra algoritma, bir önceki iterasyona kıyasla nesne grubunda herhangi bir değişiklik olup olmadığını değerlendirir. Hiçbir nesne küme değiştirmemişse, algoritma yakınsamaya ulaşmıştır ve işlem sona erer. Ancak kümelemede hala değişiklikler olduğunda, her kümedeki nesnelerin ortalama konumu hesaplanarak merkez noktası güncellenir ve iterasyon tekrarlanır. Bu işlem, optimum kümelerle sonuçlanan nesne konumlarını değiştiren hiçbir nesne kalmayana kadar tekrarlanmaktadır. Çizelge 3.1, K-Means algoritması için sözde kod biçimindeki adımların bir açıklamasını sunmaktadır.

Çizelge 3.1. K-Means Algoritması Sözde Kodu.

Input: Dataset X with n data points, Number of clusters K, Max iterations max_iter

1. Initialize Cluster Centers
2. **for** each iteration l **do**
3. Computer r_{nk} :
4. **For** each data point x_n **do**
5. Assign each data point to a cluster:
6. **For** each data point to a cluster:
7. **If** $k = \text{argmin } \|x_n - \mu_k^{l-1}\|$ **then**
8. $r_{nk} = 1$
9. **else**
10. $r_{nk} = 0$
11. **end if**
12. **end for**
13. **end for**
14. **for** each cluster k **do**
15. Update cluster centers as the mean of each cluster:
16. $\mu_k^{l-1} = \frac{\sum r_{nk} x_n}{\sum r_{nk}}$
17. **end for**
18. **end for**

Çizelge 3.1, algoritmanın her bir iterasyonunun Öklid gibi bir uzaklık metriği kullanarak her bir veri noktasının merkeze olan uzaklığını hesaplayacağı ve ardından veri noktasını en yakın merkeze sahip kümeye atayacağı merkezin rastgele başlatılmasıyla başladığını açıklayan k-ortalamlar sözde kodunu göstermektedir. Bundan sonra, her kümedeki veri noktalarının ortalama konumu hesaplanarak merkez noktası güncellenecektir. Bu işlem, veri gruplamasında herhangi bir değişiklik olmayana veya maksimum iterasyon sınırına ulaşılan kadar tekrarlanır. Nihai sonuç, nihai merkez noktası ve verilerin kümelerle optimize edilmiş dağılımıdır. K-Means algoritmasında, k küme sayısının belirlenmesinde doğru sayıya ihtiyaç vardır, çünkü başlangıçtaki küme merkezi değişebilir, böylece seçilen küme sayısı doğru değilse, veri gruplaması kararsız hale gelir (Tzortzis ve Likas, 2008). K-Means'in belirlenmesindeki adımlar aşağıda verilmiştir:

- 1) K küme sayısını ve maksimum iterasyon sayısını belirleyin
- 2) Başlatma işlemini gerçekleştirin K orta nokta kümesi, ardından merkez noktası sayısı özelliğinin denklemi:

$$C_i = \frac{1}{M} \sum_{j=1}^M x_j \quad (3.1)$$

Denklem 3.1, $i=1$ 'den $i=p$ 'ye kadar p boyut kadar yapılır.

- 3) Herhangi bir gözlem verisini en yakın kümeye bağlayın, Öklid mesafe aralığı ölçümleri denklem 2 kullanılarak bulunabilir.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3.2)$$

- 4) Veriler arasındaki farkın her bir grubun merkez noktası ile karşılaştırılmasına dayalı olarak verilerin her bir gruba yeniden tahsis edilmesi (Joshi ve Nalwade, 2013).

$$a_{ij} \begin{cases} 1 & \text{ve} \\ 0 & \text{ve} \end{cases} d = \min \{D(x_i, C_i)\} \quad (3.3)$$

Aksi takdirde, a_{ij} değeri, verilerin bir kümeye ait olup olmadığını gösterir (evet için 1, hayır için 0).

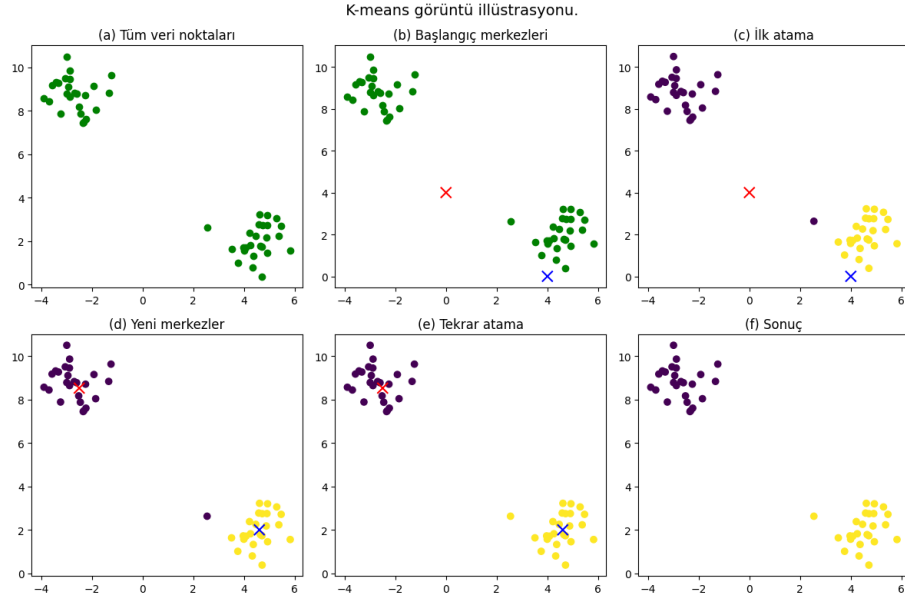
- 5) Küme merkezinin konumunu yeniden hesaplayın: Veriler yeniden tahsis edildikten sonra, merkez noktası konumu yeniden hesaplanır. Bu algorithmada amaç fonksiyonunu hesaplamak için kullanılan denklem şöyledir:

$$J = \sum_{i=1}^n \sum_{l=1}^k a_{il} D(x_i, c_l)^2 \quad (3.4)$$

Burada n veri sayısı, k küme sayısı, a_{il} x_i verisinin c_l kümesine üyelik değeri, $D(x_i, c_l)^2$ x_i verisi ile c_l kümesi arasındaki mesafedir.

- 6) Merkez noktası konumu değişirse veya iterasyon sayısı belirtilen maksimum iterasyondan azsa, işlem adım 3'e döner. Daha fazla değişiklik yoksa veya iterasyon sayısı sınıra ulaşırsa, kümeleme sonucu sonuçlandırılır.

Yukarıdaki adımlar Şekil 3.4'te K-Means ile kümeleme sürecini göstermektedir. Veri noktası başlatma ve rastgele merkez seçimi ile başlayan süreç, merkez konumu sabit olana kadar tekrarlanır ve tüm veriler her bir merkeze en yakın mesafeye göre optimum kümeler halinde gruplandırılır. Şekil 3.4, kümeleme işleminin gerçekleştirilmesinde K-Means'in her bir adımını göstermektedir.



Şekil 3.4. K-means görüntü illüstrasyonu.

Şekil 3.4, K-Means Kümeleme algoritmasındaki iterasyon sürecini göstermektedir. Her panel için aşağıdaki açıklama aşağıdaki gibidir:

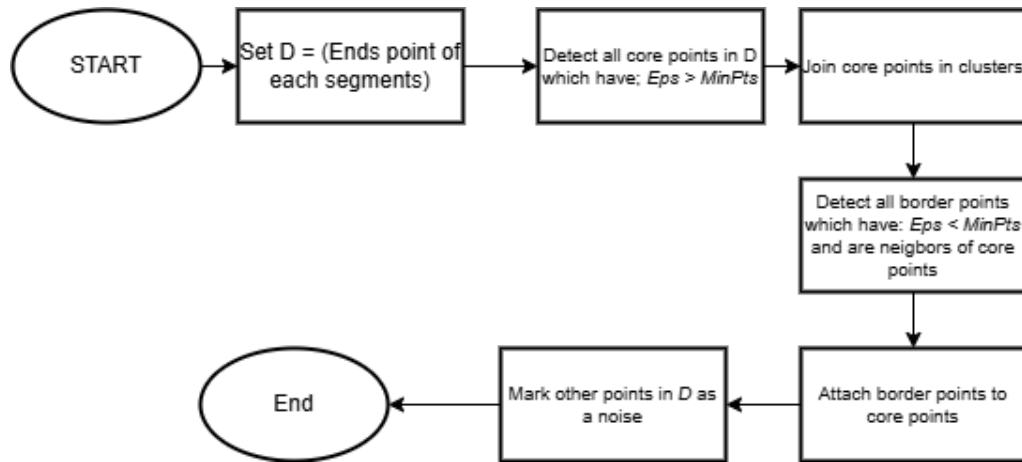
- 1) Panel a, tüm verilerin doğal kümeler halinde gruplanmadığını gösteren tek bir yeşil renkte görüntülenen başlangıç verilerini göstermektedir
- 2) Panel b, veriler arasında kırmızı ve mavi X sembolleriyle simgelenen iki noktanın bulunduğu merkez noktası başlatmayı göstermektedir.
- 3) Panel c, verilerin en yakın merkeze olan uzaklığa göre gruplandırıldığı mesafe tabanlı kümelemeyi göstermektedir. Kırmızı merkeze daha yakın olan veriler kırmızı, mavi merkeze daha yakın olan veriler ise mavi ile işaretlenmiştir.
- 4) Panel d, her bir küme içindeki ortalama konum hesaplanarak güncellenen merkez noktası güncellemesini göstermektedir. Yeni merkez, ilgili kümenin veri merkezine yaklaştırılır.
- 5) Panel e, güncellenmiş merkezoidlere dayalı kümeleme iterasyonlarını göstermektedir. Bazı veriler yeni merkeze daha yakınsa kümeleri taşıyabilir.
- 6) Panel f, iterasyon sürecinin tamamlandığı, verilerin kümelenmesinde daha fazla değişiklik olmadığı ve merkezin son konumunda olduğu yakınsamayı göstermektedir. Sonuç, kırmızı ve mavi renklerle işaretlenmiş iki kararlı kümedir,

3.2.2. DBSCAN Kümeleme

DBSCAN, veri yoğunluğuna dayalı olarak belirlenen bir kümeleme algoritmasıdır. Bu algoritmanın temel nesneleri Eps (ϵ), MinPts, doğrudan yoğunluk erişim noktaları, erişilebilir nokta terimlerine dayalı yoğunluk, yoğunluğa bağlı noktalardır (Tugay Bilgin ve Çamurcu, 2005). Çekirdek veri, eps adı verilen bir yarıçap içinde en az komşu nokta sayısı minimum değerden (minPts) büyük veya eşit olan ve kümenin merkezi olan noktadır. Öte yandan, gürültü verisi, minPts değerinden daha küçük bir eps yarıçapı içinde bir dizi komşu noktaya sahip olduğu için çekirdek nokta olmaya uygun olmayan bir noktadır, bu nedenle çekirdek noktadan eps ile ayrılmış olarak kabul edilir ve kendisi bir çekirdek nokta olarak kabul edilmek için yeterli komşuya sahip değildir. DBSCAN için aşağıdaki denklem kullanılabilir:

$$D(x_i, C_i) = \sqrt{\sum_{j=1}^q (x_{ij} - C_{ij})^2} \quad (3.5)$$

Burada $D(x_i, C_i)$ x_i ile küme merkezi C_i arasındaki mesafedir, j ise 1'den q 'ya kadar değişebilen boyut indeksidir. Bu formül, veri noktasının her bir bileşeni ile küme merkezi arasındaki farkı ölçerek, bunları toplayarak ve ardından sonucun karekökünü alarak mesafeyi hesaplamak için kullanılan bir formüldür. DBSCAN algoritmasını daha iyi kavrayabilmek adına, Şekil 3.5'te algoritmanın kullanım akışı görselleştirilmiştir.



Şekil 3.5. DBSCAN kümeleme algoritma için akış şeması.

Şekil 3.5, veri kümesindeki tüm veri noktalarından oluşan D kümesinin belirlenmesiyle başlayan DBSCAN algoritması sürecini açıklayan bir akış şemasıdır. Bir sonraki adım, yarıçapı ϵ (Eps) içinde MinPts eşğine eşit veya daha büyük sayıda

komşusu olan noktalar olan çekirdek noktalardır. Birbirine yakın olan çekirdek noktalar daha sonra tek bir küme halinde birleştirilir. Birbirine yakın olan çekirdek noktalar daha sonra tek bir kümede birleştirilir. Daha sonra, sınır noktaları, yani komşu sayısı $MinPts$ 'den az olan ancak epsilon yarıçapı içinde olan ve çekirdek noktaların komşuları olan noktalar belirlenir ve en yakın kümeye eklenir. Çekirdek noktalara veya sınır noktalarına ait olmayan noktalar gürültü olarak işaretlenir. Bu süreç, tüm verilerin kümelendiği ya da gürültü olarak işaretlendiği son kümenin oluşturulmasıyla sona erer. Çizelge 3.2, DBSCAN algoritmasının pseudocode adımlarını sunmaktadır.

Çizelge 3.2. DBSCAN Algoritması Sözde Kodu.

Input: DB : Database
Input: ϵ : Radius
Input: $minPts$: Density threshold
Input: $dist$: Distance function
Data: label: Point labels, initially undefined

```

1 foreach point  $p$  in database  $DB$  do
2   if label( $p$ )  $\neq$  undefined then continue
3   Neighbors  $N \leftarrow RANGEQUERY(DB, dist, p, \epsilon)$ 
4   if  $|N| \leftarrow minPts$  then
5     label( $p$ )  $\leftarrow$  Noise
6     continue
7    $c \leftarrow$  next cluster label
8   Label( $p$ )  $\leftarrow c$ 
9   Seed set  $S \leftarrow N \setminus \{p\}$ 
10 foreach  $q$  in  $S$  do
11   If label( $q$ ) = Noise then label( $q$ )  $\leftarrow c$ 
12   If label( $q$ ) undefined then continue
13   Neighbors  $N \leftarrow RangeQuery(DB, dst, q, \epsilon)$ 
14   If  $N \leftarrow minPts$  then
15     label( $q$ )  $\leftarrow c$ 
16 if  $|N| \geq minPts$  then continue
17  $S \leftarrow S \cup N$ 
18 return  $S_4$ 
```

DBSCAN algoritmasını açıklamaktadır. Algoritma, veritabanındaki (DB) her bir veri noktasını kontrol ederek başlar. Her bir ppp noktası için, eğer nokta henüz bir etiket almamışsa, algoritma, mesafe fonksiyonunu kullanarak ϵ yarıçapındaki komşuların sayısını hesaplar. Eğer komşu sayısı minimum eşik değeri olan $minPts$ 'ten azsa, nokta gürültü (*noise*) olarak etiketlenir. Eğer komşu sayısı yeterli ise, yeni bir küme oluşturulur ve noktanın komşuları kümeye eklenir. Algoritma, küme içindeki tüm noktalar işlenene kadar komşuları kontrol ederek kümeyi genişletmeye devam eder.

Sonuç olarak, algoritma tarafından bulunan kümeler ve gürültü olarak tanımlanan veri noktaları elde edilir.

3.2.3. Aglomeratif Kümeleme

Aglomeratif veya AGNES (*Agglomerative Nesting*), benzerliğe dayalı olarak kümelerde gruplamak için kullanılan en yaygın hiyerarşik kümeleme türlerinden biridir. Her bir nesneyle ayrı bir kümede başlayan ve daha sonra giderek daha büyük kümeler oluşturan hiyerarşik bir kümeleme algoritmasıdır. Ancak, yukarıdaki adımları gerçekleştirirken, Tek Bağlantı, Ortalama Bağlantı ve Tam Bağlantı kullanılarak veri noktası benzerlik ölçüsünün küme farklılığının bilinmesi gerekir.

- 1) Tekli bağlantı, iki küme arasındaki en yakın üyeler arasındaki mesafeye göre kümeleri birleştiren bir tekniktir. Bir sonraki adım (UV) ve (UV) arasındaki mesafeyi bulmaktır. Diğer kümeler, örneğin W, böylece aşağıdaki gibi yazılmıştır:

$$d_{(UV)} = \min(d_{uw}, d_{vw}) \quad (3.6)$$

d_{uw} U ve W kümelerinin en yakın komşu mesafesi ve d_{vw} V ve W kümelerinin en yakın komşuları.

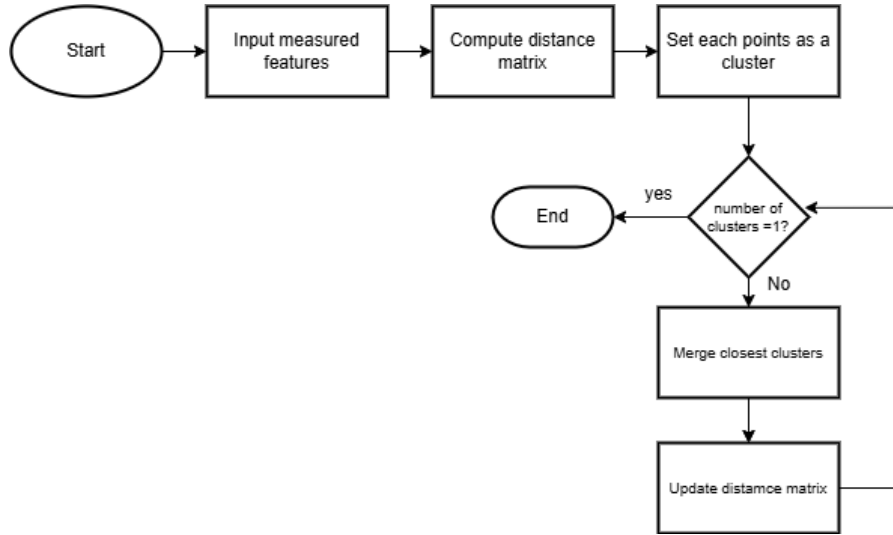
- 2) Ortalama bağlantı, bu tekniğin iki küme arasındaki kümedeki her bir üye çiftinin ortalama uzaklığını birleştireceği yerdir.

$$d_{(UV)W} = \max(d_{uw}, d_{vw}) \quad (3.7)$$

- 3) İki kümedeki en uzak üyeler arasındaki mesafeye göre kümeleri birleştiren bir teknik olan tam bağlantı.

$$d_{(UV)W} = \frac{\max(d_{uw} + d_{vw})}{n_{(UV)}n_W} \quad (3.8)$$

Şekil 3.6'da aglomeratif kümeleme algoritmasının çalışma akış şeması gösterilmiştir.



Şekil 3.6. Aglomeratif kümeleme algoritma için akış şeması.

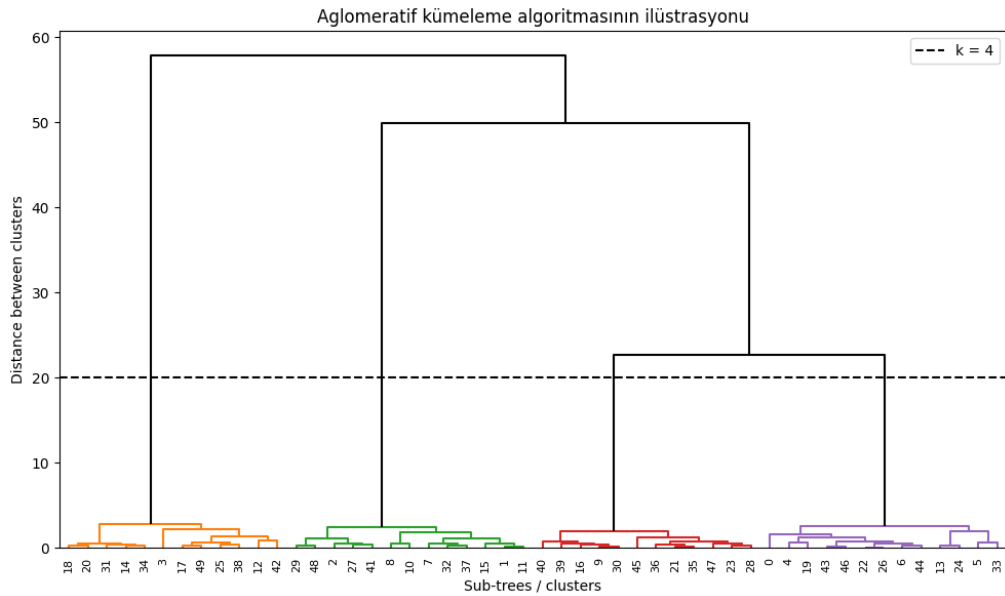
Şekil 3.6, Aglomeratif Kümeleme Algoritmasının akışını açıklamaktadır. Algoritma, kümeleme için kullanılan özelliklerin (veri) giriş olarak verilmesiyle başlamıştır. İlk adım başlar öklid (Euclidean) gibi metrikler kullanılarak veri çiftleri arasındaki mesafe matrisinin hesaplanmasıdır. Her bir veri noktası başlangıçta bireysel bir küme olarak kabul edilir. Eğer tüm veriler birleştirilmediyse, mesafesi en yakın olan iki küme birleştirilir ve bu birleşmeyi yansıtmak için mesafe matrisi güncellenir. Bu adım, tüm veriler tek bir büyük küme oluşturan kadar tekrarlanır. Sonuç olarak, dendrogram olarak görselleştirilebilen bir kümeleme hiyerarşisi elde edilir. Çizelge 3.3, aglomeratif kümeleme algoritmasının adımlarını pseudocode formatında özetlemektedir.

Çizelge 3.3. Aglomeratif Kümeleme Algoritması Sözde Kodu.

Input: Ontologies/Scheme O_1 and O_2 ; Recalculated similarity matrix, M' , O_1 and O_2
Output: Equivalent concept pairs between O_1 and O_2 , current clusters as equivalent concept pairs;

1. **while** True **do**
2. a *singleton* of cluster <- each original concept;
3. find a pair of clusters, (a) and (b), so that their similarity $s[(a), (b)] = \max(s[(m), (n)])$
4. **If** $(s[(a), (b)] < threshold)$ **then**
5. terminate the while loop
6. else
7. merge (a) and (b) into a new cluster (a+b);
8. Update M' by deleting both the row and column corresponding to (a) and (b);
9. **end**
10. **end**

Çizelge 3.3'teki sözde kod, iki ontoloji veya şema O_1 ve O_2 arasındaki kavramları eşleştirmek için kullanılan Aglomeratif algoritmasını açıklamaktadır. Süreç, ontolojideki her bir kavramı tek bir küme olarak ele alarak başlar. Algoritma daha sonra güncellenmiş benzerlik matrisine (M') dayalı olarak en yüksek benzerlik değerine sahip küme çiftlerini arar. Küme çiftleri arasındaki benzerlik değeri önceden tanımlanmış bir eşikten küçükse işlem durdurulur. Ancak, benzerlik değeri kriterleri karşılıyorsa, iki küme yeni bir kümede birleştirilir ve benzerlik matrisi, birleştirilen küme ile ilişkili satır ve sütunlar kaldırılarak güncellenir. Bu işlem, benzerlik eşikini karşılayan başka küme çifti kalmayana kadar tekrarlanır. Sonuç olarak O_1 ve O_2 ontolojileri arasında eşdeğer olduğu düşünülen bir küme çifti elde edilir ve bu da çıktı olarak verilir. Bu algoritma, hesaplanan benzerlik değerine dayalı olarak iki şema arasındaki kavramsal ilişkiyi bulmak için kullanışlıdır. Şekil 3.7, aglomeratif kümeleme algoritmasının işleyişini dendrogram kullanarak görselleştirmektedir. Bu, kümeleme sürecinde grupların nasıl birleştirildiğini veya bölündüğünü gösterir.



Şekil 3.7. Aglomeratif kümeleme algoritmasının ilüstrasyonu

Bu Şekil 3.7, aglomeratif kümeleme için kullanılan ve aralarındaki hiyerarşik ilişkiyi gösteren dendrogramdır. Aşağıda görüntünün açıklaması yer almaktadır:

- 1) Dikey eksen (alt kaleler/kümeler arasındaki mesafe) birleştirilmiş kümeler arasındaki mesafeyi gösterir, bu eksendeki bağlantı çizgisi ne kadar yüksekse, kümeler arasındaki mesafe o kadar büyüktür.

- 2) Yatay eksen (alt kale/küme): Veri nesnelerini veya alt kümeleri görüntüler. Her dal, kümelenmiş bir nesneyi veya nesne kümesini temsil eder.
- 3) Aglomeratif Kümeleme: Kümeleme işlemi, her verinin ayrı bir küme olarak ele alındığı en alttan başlar. Kümeler daha sonra tüm veriler tek bir büyük kümede toplanana kadar en yakın mesafeye göre iteratif olarak birleştirilir.
- 4) Bölmeli Kümeleme: kavramsal olarak yukarıdan aşağıya doğru gösterilir, burada kümeleme işlemi büyük bir kümenin daha küçük alt kümelere bölünmesi olarak görülebilir.
- 5) $k = 4$ 'te Yatay Çizgi: Yatay çizgi, dendrogramın belirli bir seviyede kesilmesini gösterir ve bu da $k = 4$ kümeyle sonuçlanır. Bu durumda, ortaya çıkan kümeler Küme 1, 2, 3 ve 4'tür.

Bu şekilde, verilerin belirli bir mesafe seviyesine göre nasıl birleştirildiği veya ayrıldığı da dahil olmak üzere verilerin hiyerarşik yapısı gösterilmektedir. Dendrogramın belirli bir noktada kesilmesi, kullanıcının istenen küme sayısını (k) belirlemesine olanak tanır.

3.2.4. Spektral Kümeleme

Spektral Algoritma, kümeleme yapmak için verilerden oluşturulan bir yakınlık matrisinin özvektörlerini kullanan çok yönlü bir kümeleme tekniğidir (Hidayati ve Surono, 2021). Spektral kümeleme, basitliği, sezgiselliği ve doğrusal olarak ayrılamayan veri noktalarını kümeleme yeteneği nedeniyle yaygın olarak kullanılmakta ve diğer yöntemlerle karşılaştırılabilir hatta daha iyi hesaplama sonuçları sağlamaktadır (von Luxburg, 2007a).

Spektral kümeleme, kümeleme yapmak için verilerden oluşturulan bir benzerlik matrisinin özvektörlerini kullanan çok yönlü bir kümeleme tekniğidir. Spektral kümeleme teknikleri, daha az boyutta kümeleme yapmadan önce boyutsal azaltma gerçekleştirmek için benzerlik matrisinin spektrumunu (özdeğerlerini) kullanır. Benzerlik matrisi, $A_{ij} \geq 0$ 'ın x_i ve x_j noktaları arasındaki yakınlığı gösterdiği simetrik bir A matrisi olarak tanımlanabilir. Spektral kümelemeye yaygın bir yaklaşım, Laplacian matrisi A 'nın ilgili öz vektörleri üzerinde standart kümeleme yöntemlerini (k -ortalamlar gibi) kullanmaktır. Bu öz vektörler genellikle Laplacian fonksiyonunun en büyük birkaç öz değerine karşılık gelen öz değer vektörleri olarak hesaplanır. Laplacian matrisi şu şekilde tanımlanır.

$$L = D - A \quad (3.9)$$

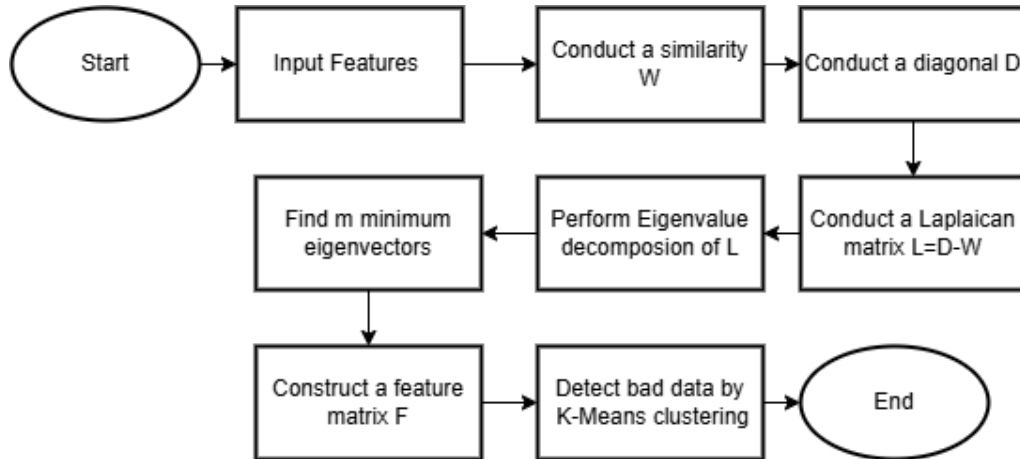
Burada D köşegen matristir:

$$D_{ii} = \sum_{j=1}^n A_{ij} \quad (3.10)$$

Popüler bir Spektral Kümeleme tekniği, Jianbo Shi ve Jitendra Malik tarafından tanımlanan normalleştirilmiş kesme algoritması veya Shi-Malik algoritmasıdır. Bu teknik, Eşitlikte tanımlanan Laplacian matrisinin en küçük özdeğerine karşılık gelen v özvektörüne dayalı olarak noktaları iki kümeye (B_1, B_2) ayırır:

$$L^{norm} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \quad (3.11)$$

Algoritma, $P = D - A$ üzerinde rastgele yürüyerek normalleştirilmiş bitişiklik matrisinin en büyük öz değerine karşılık gelen öz vektörü seçer. Şekil 3.8, spektral kümeleme algoritmasının adımlarını ve işlem akışını görselleştiren bir akış şeması sunmaktadır.



Şekil 3.8. Spektral kümeleme algoritma için akış şeması.

Akış Şeması 3.8, K-Means kümeleme yardımıyla kötü verilerin tespit edilmesinde Spektral Kümelemeyi temsil etmektedir. Süreç veri özelliklerinin girilmesiyle başlar, ardından bir benzerlik matrisi W , bir köşegen matris D ve bir Laplacian matrisi $L = D - WL - D - WL - D - WL - D - W$ oluşturur. Daha sonra, bazı minimum öz vektörleri bulmak için

L üzerinde öz ayrıştırma gerçekleştirilir. Bu özvektörler F özellik matrisini oluşturmak için kullanılır ve daha sonra kötü verileri tespit etmek için K-Means kümeleme kullanılarak kümelendir. Kötü veriler başarılı bir şekilde kümelendiğinde işlem sona erer. Çizelge 3.4, spektral kümeleme algoritmasının temel adımlarını sözde kod biçiminde özetlemektedir.

Çizelge 3.4. Spektral Kümeleme Algoritması Sözde Kodu.

Input: Data set $X = \{x_1, x_2, \dots, x_n\}$

Output: Current clusters as equivalent concept pairs;

```

1. while True do
2.   a singleton of cluster  $\leftarrow$  each original concept;
3.   find a pair of clusters,  $(a)$  and  $(b)$ , so that their similarity  $s[(a), (b)] = \max(s[(m), (n)])$ ;
4.   If  $(s[(a), (b)] < \text{threshold})$  then
5.     terminate the while loop
6.   Else
7.     merge  $(a)$  and  $(b)$  into a new cluster  $(a+b)$ ;
8.     update  $M'$  by deleting both the row and column corresponding to  $(a)$  and  $(b)$ ;
9.   end
11 end

```

Çizelge 3.4'teki sözde kod, verileri benzerliğe dayalı kümeler halinde gruplandırmayı amaçlayan Spektral Kümeleme algoritmasını açıklamaktadır. Süreç, her bir verinin tek bir küme (*singleton*) olarak kabul edilmesiyle başlar. Algoritma, benzerlik matrisine dayalı olarak en yüksek benzerlik değerine sahip küme çiftlerini iteratif olarak arar. Benzerlik değeri eşik değer altındaysa iterasyon durdurulur. Aksi takdirde, iki küme yeni bir kümede birleştirilir ve benzerlik matrisi, birleştirilen kümeden satır ve sütunlar kaldırılarak güncellenir. Bu işlem ilgili tüm kümeler oluşturulana kadar devam eder. Sonuç, verilerde bulunan kavramlar veya kalıplarla eşleşen bir kümeler koleksiyonudur. Spektral kümelemenin gerçekleştirilmesindeki aşağıdaki adımlar aşağıdaki gibidir:

- 1) Her tepe noktası eğitim verilerindeki bir kaydı temsil eder. Kenar ağırlıkları üstel uzaklık kullanılarak hesaplanır:

$$w_{ij} = \exp\left(-\frac{\|s_i - s_j\|^2}{2\sigma^2}\right) \quad (3.12)$$

Sonuç, grafiğin bir temsili olarak bir ağırlık matrisi oluşturur.

- 2) Her bir tepe noktasının derecesi, bağılı kenarların ağırlıkları toplanarak hesaplanır ve çapraz bir derece matrisi elde edilir.
- 3) Laplacian matrisi şu şekilde hesaplanır:

$$L = D - W \quad (3.13)$$

Burada D , derece matrisi ve W , ağırlık matrisidir.

- 4) Laplacian matrisinden k tane birinci eigenvector hesaplanır ve bu, $n \times k$ boyutunda bir k -eigen matrisi oluşturur (n , kayıt sayısıdır).
- 5) K -eigen matrisi normalize edilir ve her eigenvector'ün normalize edilmiş değerlerini temsil eden k sütun üretilir.
- 6) Son olarak, normalizasyon sonuçları K-Means Kümelemeye giriş olarak kullanılarak veriler kümelenir. Kayıtlar, normalizasyon sonuçlarına göre uygun kümeye atanır.

3.2.5. Deep Embedded Kümeleme

Düşük boyutlu özellikleri öğrenmede ve kümeleme hedeflerini optimize etmede derin bir çerçeve sağlayan temsili bir kümeleme algoritmasıdır, ancak yalnızca sayısal verilerle çalışır, bu da pratikte yalnızca sayısal özellikleri değil, aynı zamanda DEC tarafından ele alınamayan kategorik özellikleri de içerir (Lee ve ark., 2023).

DEC'deki kümeleme süreci iki adıma ayrılır: ön eğitim, ağırlıklı yığılmış otomatik kodlayıcı (SAE) modeli, giriş verilerini düşük boyutlu bir uzayda gizli bir vektör olarak gömmek ve ardından geri döndürmek suretiyle kullanılır. Ardından, her gömülü veri ve ilk küme merkezi girdi olarak kullanılarak yapılan ikinci kümeleme işleminde, hangi girdi verisinin hangi kümeye ait olduğunu belirleyen yumuşak atamaya dayalı bir sınıflandırma problemi çözülerek kümeleme amaç fonksiyonu optimize edilir. Algoritma, verilerin kümelenmesini teşvik eden özel bir kayıp fonksiyonu tasarlayarak küme sonuçlarını yinelemeli olarak iyileştirir. Çizelge 3.5, Deep Embedded algoritmasının adımlarını sözde kod formatında özetlemektedir.

Çizelge 3.5. Deep Embeded Algoritması Sözde Kodu.

Input: Dataset (D); number of clusters (K); encoding dimension (d); pretrain epochs ep ; clustering epochs ec ; batch size b ; learning data α
Output: Cluster assignments and trained DEC model

1. Pre-train autoencoder:
 2. Initialize input layer X and build encoder-decoder structure
 3. Train autoencoder on dataset D for ep epochs to reduce reconstruction loss.
 4. Initialize clustering:
 5. Encode dataset D using the trained encoder to obtain latent features Z .
 6. Apply K-Means on Z to initialize K cluster centers μ .
 7. Iteratively refine clustering:
 8. **For** iter = 1 to ec do
 9. Sample a minibatch from dataset D .
 10. Compute soft cluster assignments q using the Student's t-distribution.
 11. Calculate target distribution p from q to emphasize confident assignments.
 12. Compute clustering loss as KL divergence between p and q .
 13. Update encoder parameters and cluster centers using gradient descent.
 14. **end for**
 15. Evaluate clustering results:
 16. Assign data points to clusters based on maximum q values.
 17. Calculate accuracy or other metrics if ground truth labels are available.
-

İşte DEC hesaplama adımları (Lee ve ark., 2023)

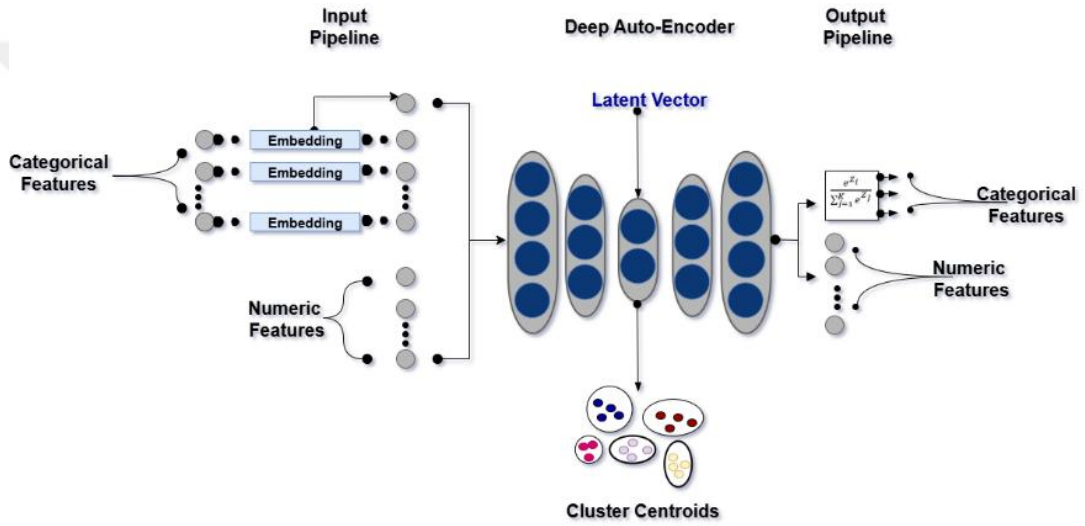
- 1) Otokodlayıcı (*Autoencoder*) ile ön eğitim: Bu aşamada DEC, ön eğitim gerçekleştirmek için bir otokodlayıcı derin sinir ağı kullanır. Bu, ilk verilerin boyutluluğunu veya temsilini azaltmayı amaçlar. Model daha düşük boyutlu bir iç temsili öğrenecektir.
- 2) Küme merkezlerinin başlatılması: Bu aşama, kodlayıcıdan elde edilen gizli temsil verileri üzerindeki merkez konumunu belirlemek için kullanılır.
- 3) Student's t-dağılımı ile yumuşak kümeleme
 - a. DEC, küme üyeliği olasılığını hesaplarken Student dağılımını kullanarak sert atamaları yumuşak atamalara dönüştürür:
 - b. 1'inci verinin j'inci kümeye ait olma olasılığı q_{ij} olarak tanımlanır:

$$q_{ij} = \frac{(1 + ||z_i - \mu_j||/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'} (1 + ||z_i - \mu_{j'}||/\alpha)^{-\frac{\alpha+1}{2}}} \quad (3.14)$$

- 4) Kayıp Fonksiyon: Yumuşak atama olasılığı q_{ij} ile hedef dağılım P_{ij} arasındaki KL sapmasını en aza indirerek gizli temsili ve küme merkezini güncelledikten sonra, hedef dağılım p_{ij} , kümeleme sürecini optimize etmek için küme merkezine daha yakın verileri güçlendirmek üzere tasarlanır.

- 5) Uçtan uca ince ayar, gradyan iniş algoritması kullanılarak oto kodlayıcı ve küme merkezi de dahil olmak üzere tüm modelin aynı anda tek bir işlemle yeniden eğitilmesidir. Bu aşamada, kodlayıcı tarafından üretilen gizli temsil, kümeleme işlemini kolaylaştırmak için geliştirilirken, merkez noktası konumu da gizli temsille daha iyi eşleşecek şekilde aşamalı olarak güncellenir.

Şekil 3.9, Deep Embedded kümeleme algoritmasının katman yapısını görselleştirerek, modelin mimarisini ve katmanlar arasındaki ilişkiyi detaylı bir şekilde göstermektedir.



Şekil 3.9. Deep Embedded Kümeleme Katmanlar görüntü illüstrasyonu.

Şekil 3.9, Derin Kümeleme (*Deep Clustering*) mimarisini üç ana bölüme açıklamaktadır: giriş hattı (*input pipeline*), derin otomatik kodlayıcı (*deep autoencoder*) ve çıktı hattı (*output pipeline*). Bu yapı, kümeleme ile birlikte temsili öğrenmeyi gerçekleştirmek için tasarlanmıştır. Giriş verisi ister kategorik ister sayısal özelliklerden oluşsun, gömme (*embedding*) ve otomatik kodlayıcı (*autoencoder*) aracılığıyla işlenerek gizli bir temsil vektörü (*latent vector*) oluşturulur. Bu gizli alanda, veri dağılımındaki kalıplara dayanarak küme merkezleri hesaplanır ve veriler gruplandırılır. Çıktı hattı, kümeleme sonuçlarını döndürür veya orijinal veriyi yeniden oluşturur kategorik özellikler için softmax, sayısal özellikler için ise doğrudan işlem uygulanır. Bu algoritma, anlamlı veri temsillerinin ve optimum kümelemenin eşzamanlı olarak öğrenilmesini sağlar.

3.2.6. Mean Shift Kümeleme

Mean Shift Kümelemesi, özellik uzayında yoğunluğun gradyanının parametrik olmayan bir tahminini gerektiren veri noktalarının özellik tabanlı bir analizidir (Apriyanty ve ark., 2023). Algoritma, her bir grubun/sınıfın merkez noktasını bulmayı amaçlayan yoğun bir nokta alanı bulmaya çalışan bir kayan pencereye dayanır ve merkez noktası olmaya aday olanları kayan penceredeki noktaların ortalaması olacak şekilde güncelleyerek çalışır.

Pencere daha sonra işlem sonrası aşamada filtrelenerek yakın kopyalar çıkarılır ve nihai merkez noktaları kümesi ve benzer yaştaki gruplar oluşturulur. Mean Shift kümelemesi, amacı kullanılan kayan pencere modelinde ortalama nokta olacak merkez nokta adaylarını güncelleyerek her bölgenin noktalarını bulmak olan merkez noktası tabanlı bir algoritmadır. Bu aday güncelleme, x_i 'nin n boyutundaki komşulukta herhangi bir noktaya karşılık geldiği varsayılarak, *Gaussian Kernal* olduğu varsayılan çekirdek bant genişliği h , tahmin edilen h 'nin gradyanı dikkate alınarak belirlenir ve aşağıdaki gibi varsayılır

$$\nabla f(x) = \left[\frac{1}{n} \sum_{i=1}^n g \left(\left| \frac{x-x_i}{h} \right|^2 \right) \right] \left[\frac{\left[\sum_{i=1}^n x_i g \left(\left| \frac{x-x_i}{h} \right|^2 \right) \right]}{\left[\sum_{i=1}^n g \left(\left| \frac{x-x_i}{h} \right|^2 \right) \right]} - x \right] \quad (3.15)$$

Yukarıdaki denklem, x noktasındaki yoğunluk fonksiyonunun vektör gradyanı, n veri sayısı, g bir gauss fonksiyonu, x veri noktası ve h bant genişliği olmak üzere veri dağılımındaki modu bulmak için kullanılır. g , $K(.)$ çekirdek fonksiyonundan türetilen bir çekirdek fonksiyonudur. Burada n veri miktarı, g gauss fonksiyonu, x veri noktaları ve h bant genişliğidir. İlk terim x noktasındaki tahmini çekirdek yoğunluğudur ve aşağıdaki gibi tanımlanır:

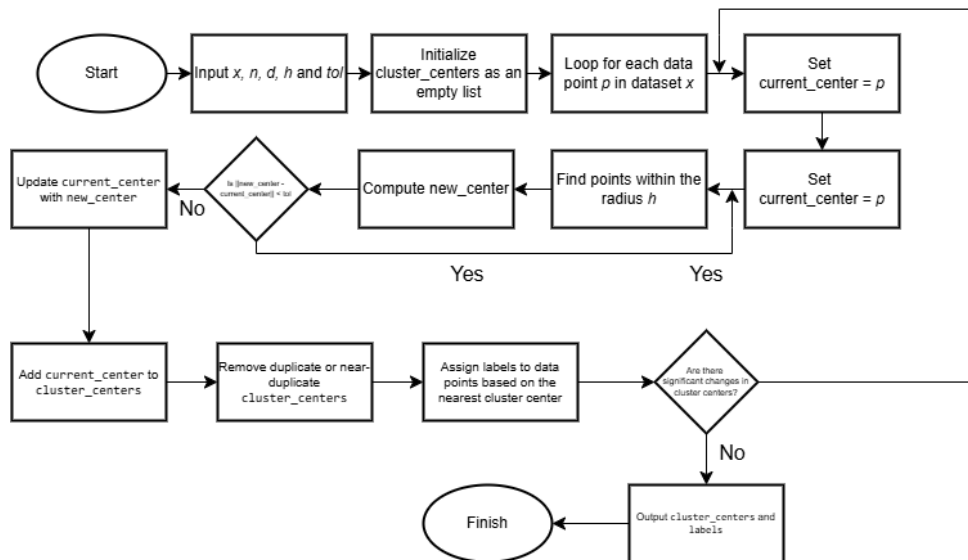
$$f(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (3.16)$$

Yukarıdaki denklem, Kernel Yoğunluk Tahmini (Kernel Density Estimation - KDE) formülünü ifade etmektedir. Bu formül, bir veri kümesinin olasılık dağılımını parametrik olmayan bir şekilde tahmin etmek için kullanılır. $f(x)$ nokta x etrafındaki verilerin yoğunluğunu ifade eder, n veri kümesindeki toplam veri noktası sayısını temsil eder. h bant genişliği parametresini ifade eder ve tahminin pürüzsüzlük seviyesini kontrol eder. Daha büyük bir bant genişliği daha pürüzsüz bir tahmin sağlar, ancak

ayrıntılar kaybolabilir. Daha küçük bir bant genişliği ise daha ayrıntılı bir tahmin verir ancak gürültüye daha duyarlı olabilir. K , Kernel fonksiyonudur ve x noktasına göre veri noktalarına ağırlık verir. Noktaya daha yakın veriler daha yüksek ağırlık alırken, daha uzak veriler daha düşük ağırlık alır. İkinci terim, Mean Shift yöntemini temsil eder ve şu şekilde açıklanır:

$$m_h(x) = \frac{\sum_{i=1}^n x_i g\left(\left|\frac{x-x_i}{h}\right|^2\right)}{\sum_{i=1}^n g\left(\left|\frac{x-x_i}{h}\right|^2\right)} - x \quad (3.17)$$

Yukarıdaki denklem, merkez noktasının (x) konumunu veri dağıtım moduna doğru güncellemek için kullanılır, burada $m_h(x)$, x etrafındaki veri dağılımına dayalı olarak x noktasının kayma yönünü ve mesafesini gösteren bir kaydırma vektörüdür. Bu kaydırma, Mean shift algoritması yinelemesinin merkez noktasının konumunu güncellemek için kullanılır, böylece x kademeli olarak veri dağıtım moduna yaklaşır. $f(x)$ Gradyanı, ortalama kaydırma vektörü veya $m_h(x)$ tarafından verilen faktör ile tahmin edilen çekirdek yoğunluğu ile orantılıdır. Gradyan $\nabla f(x)$, tahmini çekirdek yoğunluğu ile vektör ortalama kayması veya $m_h(x)$ tarafından verilen bir faktörle orantılıdır. Bu, vektör ortalama kaymasının kütle merkezinin maksimum yoğunluğunda bir artışa yol açacağını göstermektedir. Bant genişliği seçimi, normal ölçekli bant genişliği seçimi, yani *rule of-thumb* ve *bandwidth plug in* seçimi kullanılarak yapılır (Martin, 2019). Şekil 3.10, Mean Shift kümeleme algoritmasının işlem akışını görselleştiren bir akış şeması sunmaktadır.



Şekil 3.10. Mean Shift kümeleme algoritma için akış şeması.

Şekil 3.10, veri kümesi parametrelerinin (x, n, d, h, tol) girilmesi ve $cluster_center$ 'ların boş bir liste olarak başlatılmasıyla başlayan mean shift kümeleme adımlarını açıklamaktadır. Her veri noktası, $current_center$ 'ın verilere karşı ayarlandığı ana döngüde kullanılır. Algoritma daha sonra h yarıçapındaki noktaları arar, new_center belirlemek için bunların ortalamasını alır ve $current_center$ ile new_center arasındaki değişimin toleranstan (tol) daha az olup olmadığını kontrol eder. Değilse, $current_center$ güncellenir ve iterasyon devam eder. Yakınsamadan sonra $current_center$ $cluster_centers$ 'a eklenir. Tüm veriler işlendikten sonra, algoritma küme merkezlerindeki yinelemeleri kaldırır ve verileri en yakın küme merkezine göre etiketler ve kümelerde önemli bir değişiklik olup olmadığını kontrol eder. Eğer kararlı ise, algoritma tamamlanır ve sonucu çıktı olarak verir, aksi takdirde iterasyon tekrarlanır. Mean shift kümeleme algoritması için sözde kod aşağıdaki gibidir. Çizelge 3.6, Spektral Kümeleme algoritmasının temel adımlarını sözde kod formatında özetlemektedir.

Çizelge 3.6. Spektral Kümeleme Algoritması Sözde Kodu.

Initialize: Create an empty list 'cluster_centers' to store the resulting cluster centers

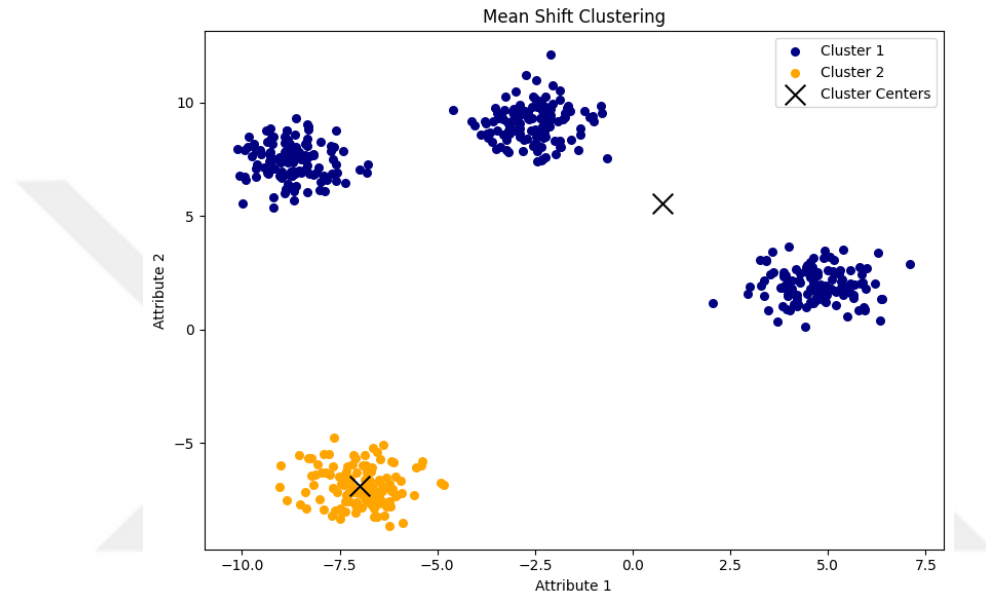
```

1. while True do
2.   for each point p in dataset x do
3.     current_center ← p;
4.     while not converged do
5.       find all points within bandwidth h of current_center;
6.       compute the mean of these points:
           new_center ← mean(points_within_bandwidth);
7.       if ||new_center - current_center|| ≤ tol then
8.         break the inner loop;
9.       else
10.        update current_center \gets new_center;
11.      end
12.    end
13.    add current_center to cluster_centers;
14.  end
15. remove duplicate or near-duplicate cluster centers;
16. assign each point in x to the nearest cluster center to create labels;
17. if no significant changes in cluster_centers then
18.   break the while loop;
19. end

```

Çizelge 3.6, veri yoğunluğuna dayalı olarak küme merkezlerini yinelemeli olarak bulan Mean Shift Kümeleme algoritmasını açıklamaktadır. Algoritma her bir veri noktasını başlangıç merkezi olarak alır, ardından bu noktanın etrafındaki h bant genişliği

yarıçapındaki tüm noktaları bulur. Bu noktaların ortalaması yeni merkez olarak hesaplanır ve bu işlem merkez yakınsayana (önemli ölçüde değişmeye) kadar tekrarlanır. Elde edilen küme merkezleri bir listede saklanır ve kopyalar kaldırılır. Ardından, her veri noktası en yakın küme merkezine göre etiketlenir. Küme merkezlerinde önemli bir değişiklik olmadığında süreç sona erer ve veriler için bir dizi küme merkezi ve etiket elde edilir. Şekil 3.11, varsayılan bant genişliği kullanılarak Mean Shift kümelemesinin sonuçlarını görselleştiren bir ilüstrasyon sunmaktadır.



Şekil 3.11. Varsayılan Bant Genişliği ile Mean Shift Kümelemesinin Gösterimi (Chugh, 2024)

Yukarıdaki resim, iki boyutlu bir veri kümesinin veri yoğunluğuna göre 4 küme halinde gruplandırıldığı Mean Shift Kümelemesinin sonucudur. Her mavi, mor, sarı ve turuncu nokta kendi kümelerinde gruplandırılmış verileri temsil etmektedir. Her kümenin ortasında Mean Shift algoritması tarafından bulunan küme merkezini gösteren bir çarpı işareti bulunmaktadır. Bu merkez, veri dağılımındaki mod veya tepe yoğunluğudur. Algoritma, küme sayısının önceden belirlenmesine gerek kalmadan veri yoğunluğuna bağlı olarak kümelerin doğal olarak oluştuğu yakınsamaya ulaşana kadar her veri noktasını en yüksek yoğunluğa sahip alana doğru kaydırır.

3.3. Normalleştirme

Normalizasyon, ön işleme aşamasında bir ölçekleme tekniği veya haritalama tekniğidir (Fenny, 2019). Bu aşama, kullanılan veri kümesinin aynı kıvrılganlığa sahip değerler konumunda veri oluşturabilmesi için çok önemlidir. Bu araştırmada, sayısal

veriler için Standardizasyon ve kategorik veriler için Tek Sıcak Kodlama olmak üzere 2 tür normalizasyon tekniği kullanılmaktadır.

3.3.1. Z-Skor Normalizasyonu (Standardizasyon)

Standardizasyon olarak da bilinen Z-Skor normalizasyonu, orijinal değerleri ortalaması 0 ve standart sapması 1 olan bir normal dağılım ölçeğine dönüştüren sayısal bir veri dönüştürme yöntemidir. Bu yöntem, veriler farklı birimlere veya özellikler arasında veri ölçeklerine sahip olduğunda kullanılır, böylece algoritmanın daha büyük değerlere karşı tarafsız olmasına yardımcı olur. Z-Skor hesaplaması aşağıdaki şekilde yapılabilir:

$$Z = \frac{X - \mu}{\sigma} \quad (3.18)$$

Burada Z normalleştirilmiş değer, X orijinal veri değeri, σ özelliğin standart sapması ve μ özelliğin ortalamasıdır. Han'a göre standardizasyon, verilerin kümeleme veya sınıflandırma sonuçlarını etkileyebilecek aşırı veya uç değerlere sahip olması durumunda önerilmektedir (Ha et al., 2011).

3.3.2. One-Hot Kodlama

One-Hot Kodlama, kategorik (nominal) verileri ikili sayısal verilere dönüştürme yöntemidir. Her kategori yeni bir sütuna dönüştürülür ve sütunun değeri kategori eşleşiyorsa 1, aksi takdirde 0 olur. Örnek Tablo 3.7, Dini bilim kümeleri, Beşeri bilimler kümesi, Sosyal bilim ailesi, Doğa bilimleri ailesi, Resmi bilim ailesi, Uygulamalı bilim ailesi olmak üzere 6 farklı veri türüne sahip bilimsel kategori sütununu göstermektedir, bu durumda hesaplama aşağıdaki gibidir.

Çizelge 3.7. Bilim Kategorilerinin Kodlanmış Hali

Bilim Türü	A	B	C	D	E	F
Dini bilim kümeleri	1	0	0	0	0	0
Beşeri bilimler kümesi	0	1	0	0	0	0
Sosyal bilim ailesi	0	0	1	0	0	0
Doğa bilimleri ailesi	0	0	0	1	0	0
Resmi bilim ailesi	0	0	0	0	1	0
Uygulamalı bilim ailesi	0	0	0	0	0	1

Bu yöntem önemlidir çünkü makine öğrenimi algoritmaları kategorik verileri doğrudan işleyemez. Sayısal etiketler olarak bırakılırsa (örneğin Erkek=1 ve Kadın=2), algoritmalar kategoriler arasında bir düzen veya mesafe olduğunu yanlış anlayabilir, oysa aslında böyle bir şey yoktur. Geron'a göre, kategoriler arasındaki ilişkinin tarafsızlığını koruduğu için, sıralı olmayan kategoriler için en yaygın ve güvenli dönüştürme yöntemidir (Géron, 2019).



4. UYGULAMA

Bu bölümde, kümeleme algoritmalarının uygulanmasından veri analizi süreci ve kullanılan veri kümesi üzerinde algoritma performansının meta-analizi ayrıntılı olarak açıklamaktadır. Bu bölümde ayrıca, daha ileri analizler için verilerin kalitesini sağlamak amacıyla kullanılan veri setinin temizlenmesi, dönüştürülmesi ve araştırılmasını içeren veri analizi de yapılmıştır.

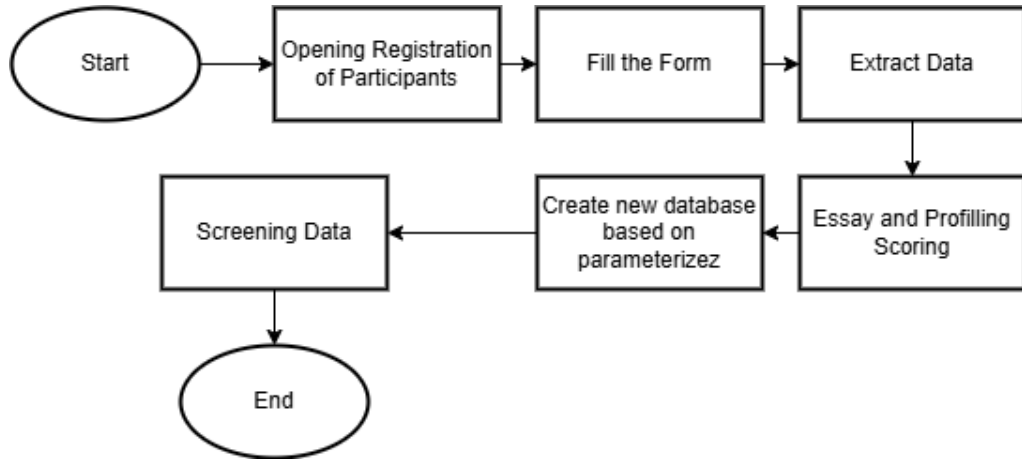
Bu analizi, bir sonraki bölümde açıklanan kümeleme algoritmasının belirli özelliklere dayalı olarak uygulanması takip etmektedir. Daha sonra, algoritmanın performansı Siluet Skoru, Davies-Bouldin Endeksi ve hesaplama süresi analizi kullanılarak değerlendirilmekte ve çeşitli senaryolarda algoritmanın etkinliği ve verimliliği değerlendirilmektedir. Bu araştırmanın sonuçları, değerlendirme sonuçlarına dayalı olarak her bir kümeleme algoritmasının güçlü ve zayıf yönlerini belirleyen bir meta-analizde özetlenmiştir.

4.1. Veri Seti Üzerinde Yapılan Çalışmalar

Yapay zekâ araştırma ve geliştirme çalışmalarında veri kümeleri önemli bir bileşendi çünkü modelin kalitesi kullanılan verilere bağlıydı (Hermawan, 2024). Bu bölümde, araştırmanın yürütülmesinde kullanılan verilerle ilgili ayrıntılı açıklama yapılacaktır. Bu veri seti, eğitim dünyasında yer alan ve çeşitli ülkelerde düzenli olarak sempozyumlar düzenleyen bir kuruluşa ait verilerdir.

4.1.1. Veri Toplama

Bu araştırmayla ilgili olan araç türü, bu araştırmada kullanılan yaklaşıma uygun olduğu için doğrudan uluslararası sempozyum web sitelerinden biri aracılığıyla alınan veri setlerinin alınmasıdır. Veri toplama için kullanılan akış şeması Şekil 4.1'de açıklanmaktadır.



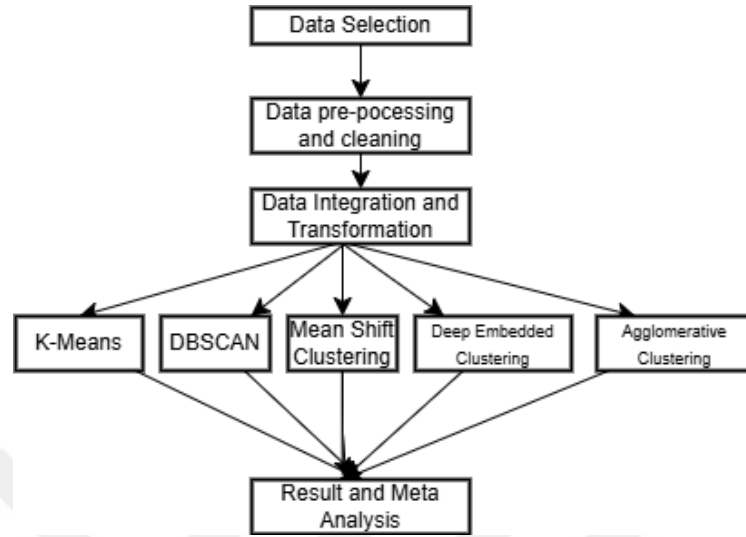
Şekil 4.1. Youth Break the Boundaries sisteminde veri toplama sürecinin akış şeması

Veri toplama süreci, veri setinin bir parçası olan katılımcı kaydının açılmasıyla başlar. Bu süreç katılımcılar tarafından organizatörün bir form içeren resmi web sitesinde gerçekleştirilir. Daha sonra katılımcılar, kişisel veriler ve makalelerle ilgili bilgiler ve makale toplama da dahil olmak üzere konferans etkinliklerinde katılımcıları tanımlamak amacıyla kullanılan diğer bazı bilgiler şeklinde veri bilgilerini doldururlar. Ayrıca sistem, katılımcılar tarafından yönetici web sitesine girilerek yapılan veri çıkarma adı verilen verileri toplar. Bu aşamada, veri çıkarma manuel olarak yapılacak ve veriler, gelen her veriyi değerlendirmek için zaten bir formatın bulunduğu bir excel formatına girilecektir. Ayrıca, konuya uygunluk, yenilik yaratıcılığı, yapı ve sağlanan referanslardan elde edilen bilgiler gibi profil oluşturma parametrelerine dayalı olarak girilen veriler üzerinde bir değerlendirme yapılacaktır. Ayrıca, girilen değerlendirme verileri, diğer parametrelerle birleştirilmek üzere araştırmacılar tarafından oluşturulan yeni bir veri tabanına girilecektir. Son aşamada, toplanan ve ayrıştırılan veriler, analiz ve araştırma için kullanılabilmesi amacıyla verilerin uygunluğunu ve kalitesini sağlamak için taranır.

4.1.2. Veri Analizi

Veri toplama süreci, veri kümesinin bir parçası haline gelen katılımcı kayıtlarının açılmasıyla başlamıştır. Bu çalışmada kullanılan algoritmaların meta analizini gerçekleştirmek için KDD (*Knowlegdes Discovery in Databases*) süreç seviyesi kullanılmıştır. Bu KDD sürecinde veri seçimi gerçekleştirilmekte, ardından veri ön işleme yapılmakta, daha sonra veri entegrasyonu ve veri dönüşümü gerçekleştirilmekte, ardından algoritmalar kullanılarak veri madenciliği hesaplamaları yapılmakta ve son olarak

kullanılan her bir algoritma için analizler yapılarak sonuçlar gruplandırılmakta ve değerlendirilmektedir. Bu araştırmadaki veri analizi akışı Şekil 4.2'de gösterilmektedir.



Şekil 4.2. Araştırmada Kümeleme Algoritması Analizinin Akışı

4.2. Veri Seçimi

Bu araştırmada kullanılan ilk veri kümesi, Youth Break the Boundaries kuruluşunun yönetici web sitesinden alınan ve 11 öznetelikten oluşan uluslararası bir sempozyuma ait bir veri kümesidir. Bilgiler Çizelge 4.1'de gösterilmektedir.

Çizelge 4.1. Uluslararası sempozyum katılımcıları veri seti

Özellikler	Veri Tipi	Açıklama
Email (e-posta)	Varchar	Katılımcının e-posta adı
full_name (tam adı)	Varchar	Kullanıcının tam adı
Gender (cinsiyet)	Enum (Male, Female)	Sempozyum katılımcılarının cinsiyeti
Birthdate (doğum tarihi)	Date	Katılımcının daha sonra yaş verisine dönüştürülecek olan doğum tarihi
Occupation (meslek)	Varchar	Katılımcıların çalışmaları
Major (bölüm)	Varchar	Katılımcının bilimsel ailesi
Experiences (deneyim)	Text	Daha sonra kategorize edilecek kullanıcı deneyimi
Essay topic (Kompozisyon Konusu)	Varchar	Kullanıcının kompozisyon konusu seçimi
Essay (kompozisyon)	Text	Kullanıcı tarafından yazılmış kompozisyon cevapları
Essay Score (kompozisyon puanı)	Float	Katılımcının kompozisyon puanı
Personality Result	Varchar	Kullanıcılardan Gelen Kişilik Sonuçları

Çizelge 4.1'deki veri seti, analiz edilecek birkaç önemli özellikten oluşmaktadır:

- 1) **Demografik Bilgiler:** Bu türde, katılımcının profilini ve profilini sağlamak için Cinsiyet, Yaş ve Doğum Tarihi sütunları alınacaktır.
- 2) **Tema ve deneyim:** Tema, deneyim ve başarı parametrelerinden alınan bu veri türünde, katılımcının deneyim ve başarı düzeyi.
- 3) **Metin:** bu sütun, katılımcının temasına göre fikirleri tanımlayan metinsel verileri içerir.
- 4) **Sayısal:** Bu veri, katılımcıların düşünce kalitesindeki eğilimleri belirlemek için kullanılan katılımcıların kompozisyon değerlendirmesinin puanıdır.

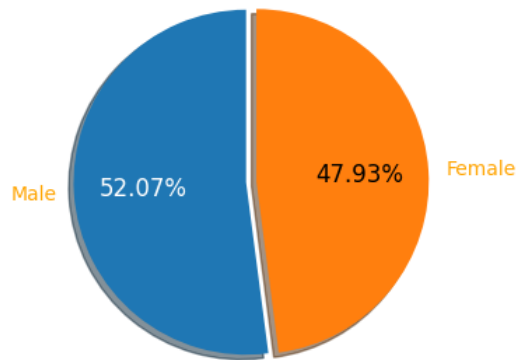
4.2.1. Veri Özellikleri

Çalışmada kullanılan veri satırı sayısı 22.589 satırdır. Bu çalışmada kullanılan veriler demografik özelliklerden mesleki deneyime, tema tercihlerinden makale kalitesine ve kişilik anketi sonuçlarına kadar çeşitli ilgili boyutları içermektedir. Veri setindeki her bir parametre, kümeleme ve meta-analiz sonuçları için daha zengin bir bağlam sağlamak üzere seçilmiştir. Aşağıdaki açıklama, bu çalışmada kullanılan verilerin her bir özelliği hakkında ayrıntılar sunmaktadır:

4.2.1.1. Cinsiyet

Bu parametre, veri setindeki cinsiyet dağılımına bakmak ve cinsiyetin kompozisyon ve kişilik puanları gibi belirli sonuçları nasıl etkilediğini analiz etmek için kullanılır. Bu araştırmada kullanılacak cinsiyete ilişkin veri dağılımı Şekil 4.3'te aşağıdaki gibi gösterilmiştir.

Data Sharing Distribution Based on Gender



Şekil 4.3. Veri setindeki verilerin cinsiyet dağılımı

Yukarıdaki Şekil 4.3, kullanılan toplam 22 bin verinin %52,07'sinin erkek ve %47,93'ünün kadın olduğu cinsiyete dayalı veri bölümü dağılımını göstermektedir. Erkekler ve kadınlar arasındaki oran farkı, tek bir grup arasında baskın olmayacak şekilde nispeten dengelidir. Ayrıca elde edilen sonuçların bir cinsiyete karşı önyargılı olmaması amaçlanmıştır.

4.2.1.2. Deneyim

Kullanılan ikinci parametre deneyimdir. Bu parametre, katılımcıları yaşadıkları deneyime göre gruplandırmak için bu araştırmada çok önemlidir. Elde edilen veriler giriş seviyesi, erken seviye, orta kariyer seviyesi, üst seviye ve yönetici seviyesi olmak üzere 5 kategoride gruplandırılacaktır. Katılımcıların her bir sınıfa ayrılmasında kullanılan aşağıdaki kategoriler Çizelge 4.2'de gösterilmiştir.

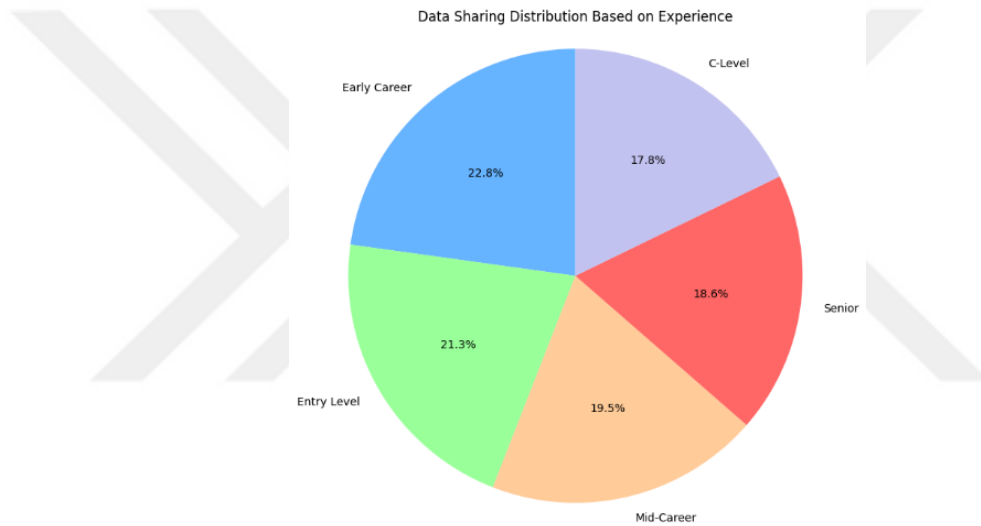
Çizelge 4.2. Deneyime Dayalı Kariyer Kategorileri

Tür	Deneyim Yılları	Mevcut Pozisyon	Eğitim	Yaş Aralığı
Entry Level Deneyimsiz	0-1 yaş	Öğrenci	Yeni mezun olmuş veya kariyerlerinin ilk aşamalarında.	< 20 yaş
Early Career Erken Kariyer	1-3 yaş	Staj, stajyer veya alt düzey pozisyon	Biraz mesleki deneyime sahip diploma sahibi	<24 yaş
Mid-Career (Orta Kariyer)	3-8 yaş	Orta ya da 3 yıldan fazla süredir çalışan biri.	Küçük ekipleri veya projeleri yönetme ve önemli kararlara katkıda bulunma konusunda deneyimli.	28 – 31 Yaş
Senior (Kıdemli / Deneyimli)	8-10	Üst düzey yönetici, bölüm başkanı veya ileri düzey uzman.	Karar almada etkili ve üst düzey stratejik sonuçlardan sorumludur.	>30 yaş
C-Level	>9	Organizasyonda en üst düzey yöneticiye sahip biri	Kariyer basamaklarının en üstünde bulunan kişiler, örneğin İcra Kurulu Başkanı, Finans Direktör veya Operasyon Direktörü.	>27

Yukarıdaki Çizelge 4.2, giriş seviyesine dahil olan katılımcıların profesyonel iş deneyimi olmayan ya da staj veya kuruluşlara katılım gibi çok az deneyime sahip olan katılımcılar olduğunu açıklamaktadır. Bu kariyer seviyesine dahil olan katılımcılar, bir kariyerde temel becerileri oluşturma henüz erken aşamalarında olan katılımcılardır. Bir sonraki grup türü erken kariyerdir ve bu seviyede kariyerlerine başlamış ancak 6 aydan fazla deneyime sahip stajyerler ve hala alt seviyede olan çeşitli pozisyonlar gibi erken profesyonel aşamalarda olan katılımcılar yer almaktadır, daha sonra orta kariyerden gelen katılımcılar yer almaktadır ve bu gruptaki katılımcılar kendi alanlarında 3-8 yıl

katkıda bulunmuş katılımcılardır. Bir sonraki seviye ise üst düzeydir; bu gruptaki katılımcılar uzun süredir mesleklerinde olan ve büyük ekiplere liderlik etme veya önemli stratejik kararlar alma deneyimine sahip katılımcılardır. En üst seviyede, yani yönetici seviyesi C veya direktör bu gruptaki katılımcılar profesyonel alanda 9 yıldan fazla deneyime sahip olan ve genellikle en yüksek yönetici pozisyonlarında bulunan katılımcılardır.

Oluşturulan kategori grupları, katılımcının deneyimine, sosyal faaliyetlerine ve başarılarına göre kategorize edilmek üzere her bir katılımcının verilerine yerleştirilecek kategorilerdir. Şekil 4.4, araştırma verilerinde toplanan katılımcı verilerine genel bakışın bir özetidir.



Şekil 4.4. Deneyim seviyesine göre veri paylaşım diyagramı

Yukarıdaki Şekil 4.4'te yer alan diyagram, çalışmada kullanılan verilerdeki dağılımı göstermektedir. Veri satırında yer alan 5 kategoriden en yüksek yüzde Erken Kariyer'dir, bunun nedeni birçok katılımcının 20'li yaşlarının başında olmasıdır, ardından %22,8 ile Giriş seviyesi, %19,5 ile Kariyer Ortası, %18,6 ile Kıdemli ve %17,8 ile C-Seviyesi gelmektedir. Kullanılan veriler, kümeleme sürecinde oluşacak önyargıyı en aza indirecek şekilde herhangi bir meslek grubuna hakim olmayan verilere sahiptir.

4.2.1.3. Konu

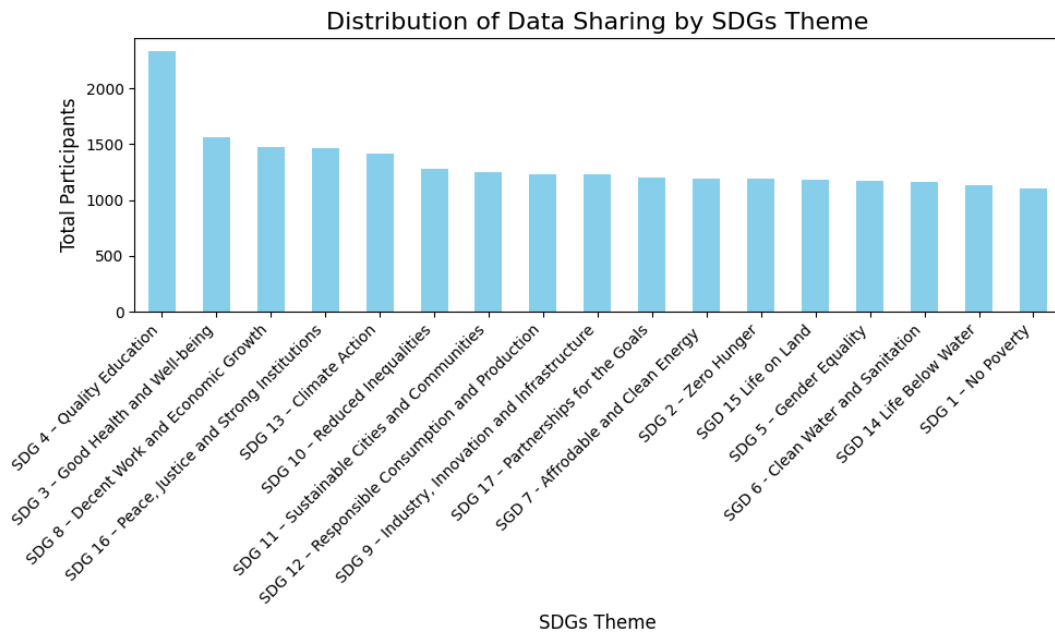
Her konferans, her bir organizatör tarafından belirlenen hedefler doğrultusunda kendi temasına sahiptir. *Youth Break the Boundaries* tarafından düzenlenen sempozyum *Sustainable Development Goals* (SGD'ler) odaklanmıştır. 2015 yılında tüm *United*

Nation üyeleri, 17 SGD belirleyen 2030 Sürdürülebilir Kalkınma Gündemini kabul etmiştir. Bu küresel hedefler "insanlar ve gezegen için barış ve refah" için çaba göstermektedir (United Nation, 2022). Aşağıda Şekil 4.5'te SGD'lerdeki temalara genel bir bakış yer almaktadır



Şekil 4.5. Şekil 17 SGD'ler teması için tema ve simge (Lewis ve Kearney, 2020)

Yukarıdaki temalara dayanarak, Şekil 4.5 araştırma veri setinde bulunan temalara göre araştırma verilerindeki verilerin dağılımını göstermektedir.



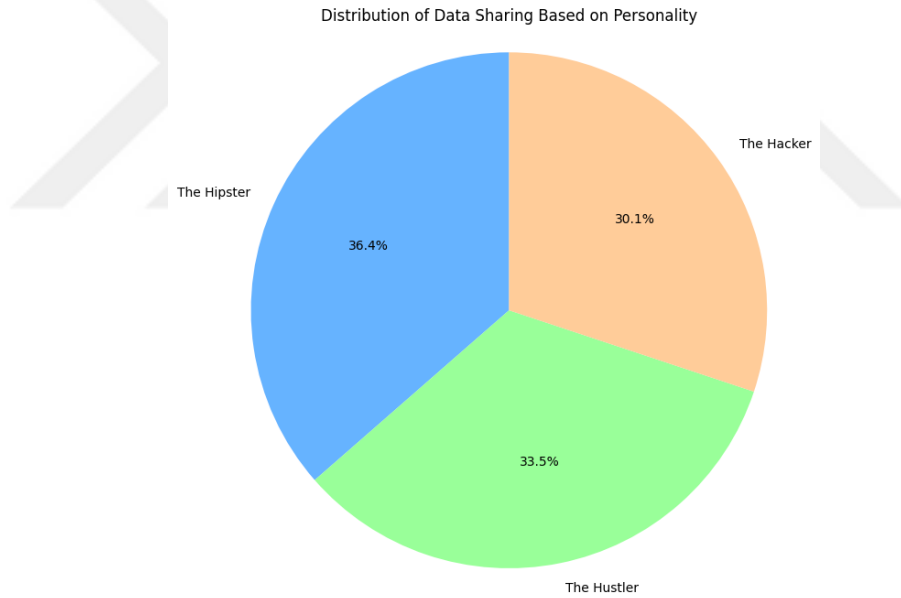
Şekil 4.6. Veri Paylaşımının Temalara Göre Dağılımı

Şekil 4.5, Sürdürülebilir Kalkınma Hedefleri (SGD'ler) temasına göre bireysel katılımcıların dağılımını göstermektedir. Grafikten, SGD's 4 - kaliteli eğitim temasının,

2000 katılımcıyı aştığı için katılımcılar arasında en popüler tema olduğu görülebilir. Temalar açısından grafik henüz adil değildir çünkü SGD 4 teması ile çok daha az sayıda bireye sahip olan SGD 1 ve 14 gibi çarpıcı bir dengesizlik söz konusudur.

4.2.1.4. Kişilik Testi Sonucu

Bu parametre, katılımcıların bir sonuç üretmek için her biri farklı bir ağırlığa sahip 25 soruyu dolduracakları bir anket kullanılarak gerçekleştirilir. Bu üç kişilik Hustler, Hipster ve Hacker'dır. Bu üç terim ilk olarak 2012 yılında eski AKQA Kreatif Direktörü Rei Inamoto tarafından ortaya atılmıştır ve günümüzde daha çok "*The Startup Triangle* (Startup Üçgeni)" olarak bilinmektedir. Bu test, uzmanlar tarafından derlenen ve her biri rastgele düzenlenmiş farklı ağırlıklı cevaplara sahip 25 kadar sorudan oluşan bir anket ile oluşturulmaktadır (Susilowati & Başçiftçi, 2025). Şekil 4.7, kullanılan araştırma verilerindeki veri dağılımını göstermektedir.



Şekil 4.7. Veri Paylaşımının Temalara Göre Dağılımı

Yukarıdaki Şekil 4.7'de, kullanılan veri tabanındaki kişilik kategorilerine göre verilerin dağılımı yer almaktadır. İncelendiğinde bu verilerin oldukça dengeli bir dağılıma sahip olduğu görülmektedir çünkü bu testin sonuçlarına hakim olan tek bir kategori yoktur. Hipster %36,4 ile en yüksek değeri gösterirken, Hustler %33,5 ile ikinci en büyük dağılıma sahiptir ve Hacker toplam katılımcıların %30,1'ini oluşturmaktadır.

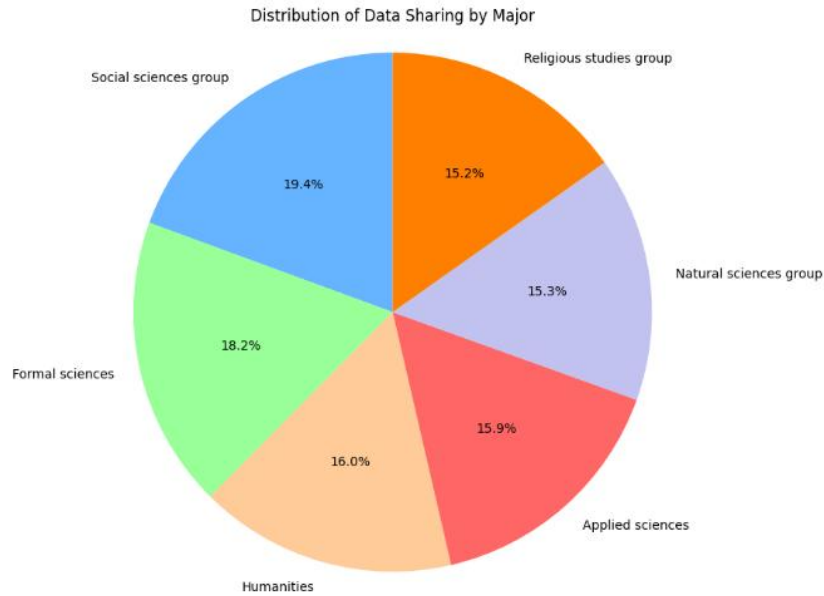
Bu veriler oldukça dengelidir ve daha sonra kümeleme algoritmaları kullanılırken önyargıyı önler.

4.2.1.5. Bilimsel Gruplar

Bu çalışmada kullanılan özellikler, çalışma alanlarını ve disiplinleri belirli kategorilere ayırmak için kullanılmıştır. Endonezya Eğitim ve Kültür Hükümeti'nin 2014 tarihli 124 sayılı yönetmeliğine atıfta bulunarak, bilim ve teknoloji dallarının 6 bilim kümesinde olduğunu belirtmektedir:

- A. Dini bilim kümeleri
- B. Beşeri bilimler kümesi
- C. Sosyal bilim ailesi
- D. Doğa bilimleri ailesi
- E. Resmi bilim ailesi
- F. Uygulamalı bilim ailesi

Bu düzenlemeler temelinde, bu araştırma da verileri bu 6 kategoriye göre sınıflandırmaktadır. Aşağıdaki Şekil 4.8, araştırmadaki verilerin dağılımını göstermektedir.



Şekil 4.8. Veri Paylaşımının Temalara Göre Dağılımı

Çalışmada kullanılan yukarıdaki Şekil 4.8'deki verilere dayanarak, ana verilerin dağılımı da tek bir bilim ailesine hakim değildir. Verilerdeki hakimiyet çok önemlidir, kümeleme yapılırken bundan kaçınılır.

4.2.1.6. Kompozisyon puanları

Her eğitim sempozyumunda, katılımcıların fikirlerini sistematik bir şekilde aktarma becerilerini belirlemek ve test etmek için kullanılan yazı türlerinden biri de denemelerdir. Veri kümesindeki denemelerin özellikleri panelistler tarafından manuel olarak işlenerek çok zayıftan çok iyiye kadar değişen kalitelerine göre değerlendirilir.

Bu kategori, cevapların doğruluğunu ve argümanların gücünü değerlendirir, paragrafların bütünlüğü ve tutarlılığının yanı sıra gramer doğruluğu ve kompozisyon yazma stili gibi ana unsurlara dayanmaktadır. Bu çalışmada kompozisyonlar şu şekilde gruplandırılmıştır

1. Çok zayıf 1-30
2. Zayıf 21-40
3. Orta 41-69
4. İyi 61-80
5. Çok iyi 81-100

Bu kümeleme, panelistlerin denemeleri belirli deneme sınıflarına ayırarak değerlendirmelerine yardımcı olur. Ancak bu araştırma, panelistlerin verdiği orijinal puanları kullanır ve denemelerin kümelenmesini dikkate almaz.

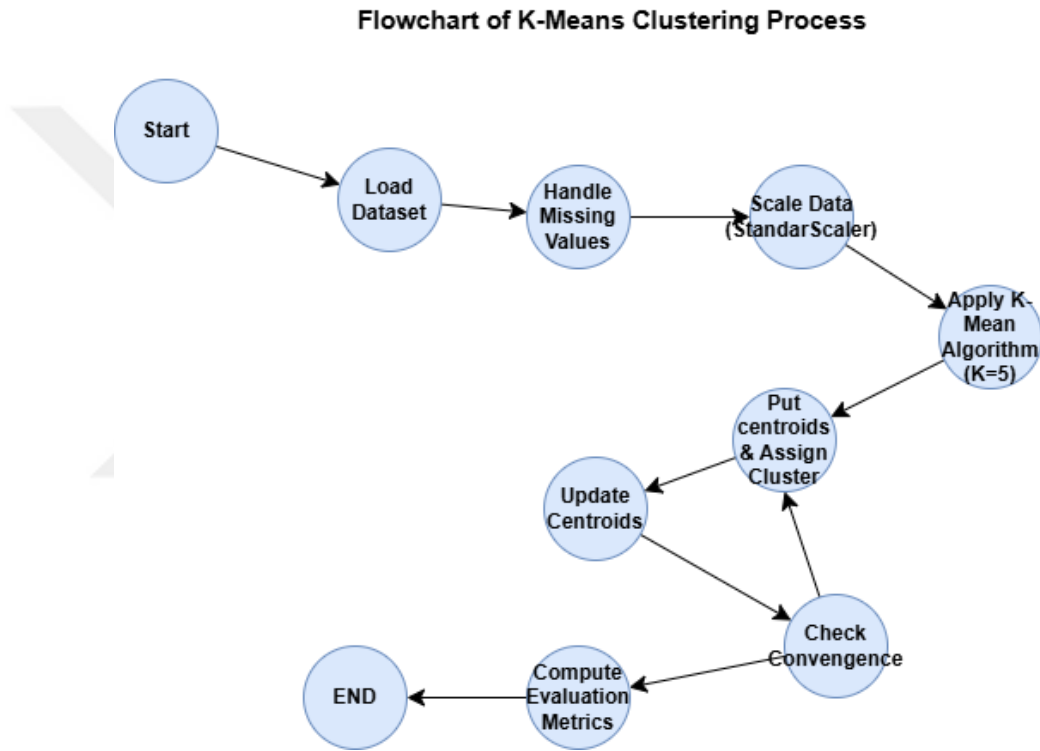
4.3. Meta Analizi

Bu araştırmadaki meta-analiz, büyük veride kullanılan çeşitli performansları incelemek için yapılmıştır. Bu yaklaşım, her algoritmanın eksikliklerini, eksikliklerini ve etkinliğini daha geniş bir şekilde anlamak için her algoritmanın sonuçlarını birleştirmeye hizmet eder.

4.3.1. K-Means Kümelemesi Çalışma Analizi

Bu çalışmada kullanılan küme sayısı, sınıflandırmanın ilk başlangıcı olarak 5'tir. Bu çalışmadaki başlangıç süreci, kümeleme sürecine müdahale etmemek için her kümenin ortalama değerini kullanarak imputasyon tekniği ile önce işlenecek boş değerler içeren verileri kontrol eder. Veriler işlendikten sonra, küme merkezi (*centroid*) rastgele

olarak başlatılır ve Öklid Uzaklığı formülü kullanılarak veri ile küme merkezi arasındaki mesafe hesaplanır. Yineleme işlemi, merkez noktası yakınsak bir duruma ulaşana kadar, yani artık önemli ölçüde değişmeyece ve hiçbir veri başka bir kümeye taşınmayana kadar, her kümedeki verilerin ortalamasına dayalı olarak merkez konumunu güncelleyerek devam eder. Kümeleme sonuçlarının kalitesini değerlendirmek için, her bir veri noktasının diğer kümelere kıyasla ilgili kümede ne kadar iyi olduğunu ölçen Siluet Puanı kullanılır. Aşağıdaki şekil 4.9'da bu çalışmadaki K-Means süreci bir akış şeması ile açıklanmaktadır



Şekil 4.9. K-Means Algoritması ile Kümeleme Sürecinin Akış Şeması

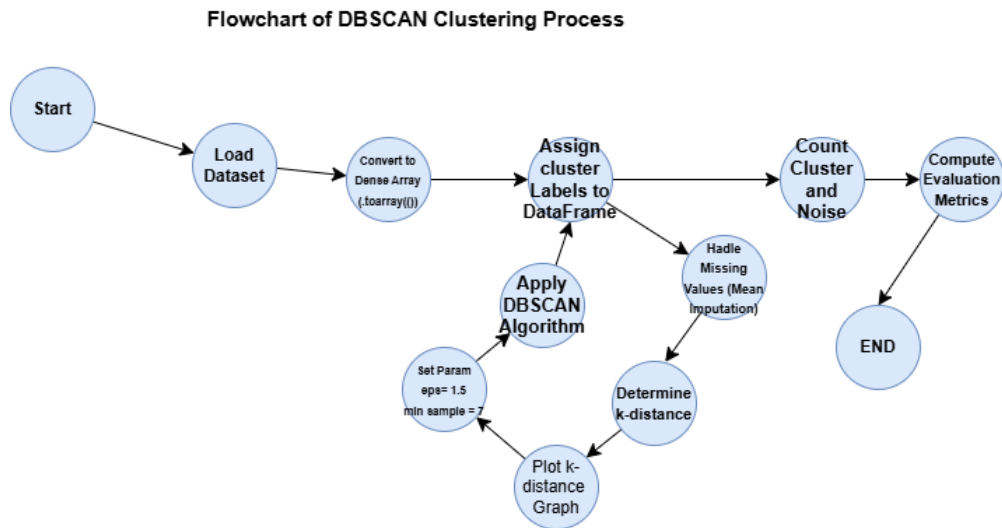
Şekil 4.9, kullanılan veri kümesinin yüklenmesiyle başlayan K-Means Kümeleme algoritmasının uygulanma sürecini göstermektedir. Daha sonra kullanılan veriler, ortalama yöntemiyle eksik değerleri ele alarak boş değerleri ortadan kaldıracaktır. Veriler temizlendikten sonra bir sonraki adım, özellikleri aynı ölçekte olacak şekilde standartlaştırmaktır, ardından küme sayısı 5 olarak başlatılarak K-Means algoritması uygulanacaktır. Daha sonra süreç, yeni merkezin hesaplandığı ve verilerin merkeze yakınlığına göre gruplandırıldığı ve ardından mevcut küme üyelerine dayalı olarak merkez konumunu güncellemek için merkezin güncellendiği yinelemeli döngüye girer.

Yakınsama kontrolü aşamasında merkezin stabilize olup olmadığından emin olunur, eğer stabilize olmadıysa süreç merkez hesaplamasına geri döner, eğer stabilize olduysa süreç değerlendirme aşamasına devam eder. Son aşamada, kümeleme kalitesini değerlendirmek için bir değerlendirme yapılmıştır.

4.3.2 DBSCAN Çalışma Analizi

Bu çalışmada, DBSCAN verileri yoğunluğa dayalı olarak kümelemektedir. Bu algoritma, başlangıçta toplam kümeleri belirleyen K-Mean algoritması ile aynı değildir. DBSCAN'de küme sayısı Minimum Puan (minpts) değerine bağlıdır. Bu çalışmadaki epsilon değeri, mesafenin dramatik bir şekilde artmaya başladığı "*Elbow (dirsek)*" veya keskin virajı tanımlayarak k-mesafe grafiğine bakılarak elde edilir. Bu nokta, DBSCAN için en uygun epsilon değeri için referans olarak kullanılır.

Bu çalışmada, küme olarak kabul edilecek noktalar arasındaki maksimum mesafeyi belirlemek için epsilon (eps) =1.5 ve bir noktanın çekirdek nokta olarak kategorize edilebilmesi için eps yarıçapı içindeki minimum nokta sayısını belirlemek için min_samples=7 kullanılmıştır. Ardından, DBSCAN veri kümesindeki üç tür noktayı tanımlar: eps yarıçapı içinde yeterli komşuya sahip çekirdek noktalar, eps yarıçapı içinde olan ancak küme çekirdeği olarak nitelendirilemeyen sınır noktaları ve herhangi bir kümeye ait olmayan gürültü noktaları. DBSCAN modeli çalıştırıldıktan sonra, sonuçlar daha fazla analiz için veri çerçevesinde küme etiketleri şeklinde saklanır. Bu çalışmada DBSCAN uygulamasına ilişkin akış şeması Şekil 4.10'da gösterilmektedir



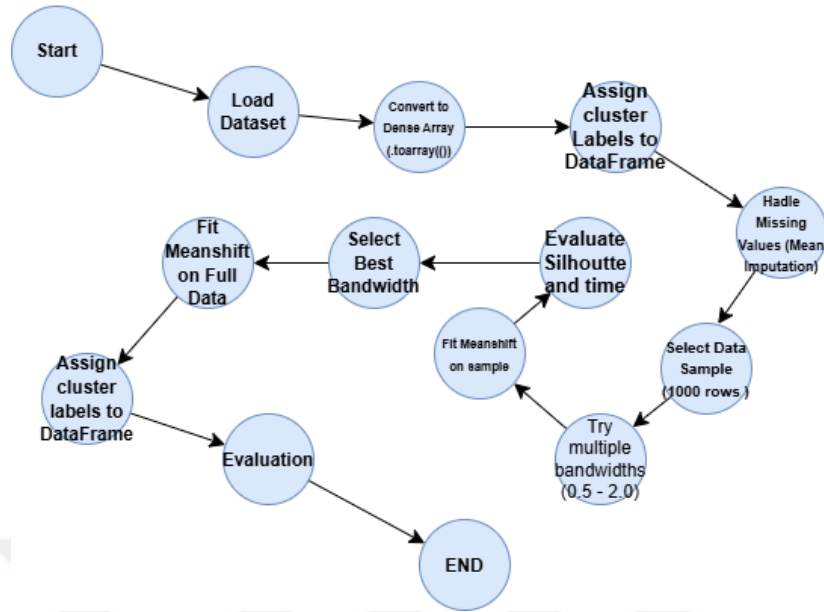
Şekil 4.10 DBSCAN Algoritması ile Kümeleme Sürecinin Akış Şeması

Şekil 4.10'da DBSCAN algoritması kullanılarak yapılan kümeleme süreci, analiz edilecek verilerin sisteme yüklendiği Veri Yükleme (Load Dataset) aşamasıyla başlamaktadır. Ayrıca, hâlâ seyrek matris (sparse matrix) biçiminde olan veriler, algoritmaya uyumlu hale getirilmesi için `.toarray()` yöntemi kullanılarak yoğun bir diziye (dense array) dönüştürülür. Kayıp veriler (missing values) varsa, bunlar ortalama atama (mean imputation) yöntemiyle giderilir. Daha sonra, `eps` ve `min_samples` parametrelerinin belirlenmesi amacıyla `k`-mesafesi (`k`-distance) hesaplanır ve görselleştirilir. Parametreler belirlendikten sonra DBSCAN algoritması çalıştırılır ve sonuçlar bir veri çerçevesi (*DataFrame*) içinde kümeler olarak etiketlenir. Son aşamada, toplam küme sayısı ve gürültü (*noise*) hesaplanır; kümeleme performansı ise Siluet Skoru, Davies-Bouldin Endeksi ve Dunn Endeksi gibi metrikler kullanılarak değerlendirilir.

4.3.3 Mean Shift Kümelemesi Çalışma Analizi

Mean Shift Kümeleme Algoritması, yoğunluğa dayalı veri kümeleme algoritmasıdır. Algoritma uygulanmadan önce veri kümesi yoğun diziye dönüştürülür ve önceki algoritmada kullanılan yöntemle eksik değerler doldurulur. Bu algoritmada etkili olan parametre bant genişliğidir. Bu nedenle, Mean Shift algoritmasını uygulamadan önce, bu çalışmada ilk 1000 veri üzerinde bant genişliği adayları test edilmiştir. Ardından, Mean Shift algoritması, önceki deneylerde ilk 1000 veri üzerinde elde edilen 2 bant genişliği parametresi kullanılarak uygulanmıştır. Bu parametre, küme merkezini (*centroid*) bulmak için arama yarıçapını belirler. Kümeleme işlemi, veri noktalarının konumlarını en yüksek yoğunluklu bölgelere doğru güncelleyerek yinelemeli olarak gerçekleştirilir, konverjans elde edilene kadar, yani merkez noktaların konumlarında artık önemli bir değişiklik olmayana kadar. Kümeleme işlemi tamamlandıktan sonra, `mean_shift.labels_`'den elde edilen küme etiketleri *DataFrame*'e eklenir. Aşağıda, bu çalışmada mean shift kümeleme yöntemini kullanan sürecin akışını gösteren Şekil 4.11 verilmiştir.

Flowchart of Meanshift Clustering Process



Şekil 4.11. Mean Shift Algoritması ile Kümeleme Sürecinin Akış Şeması

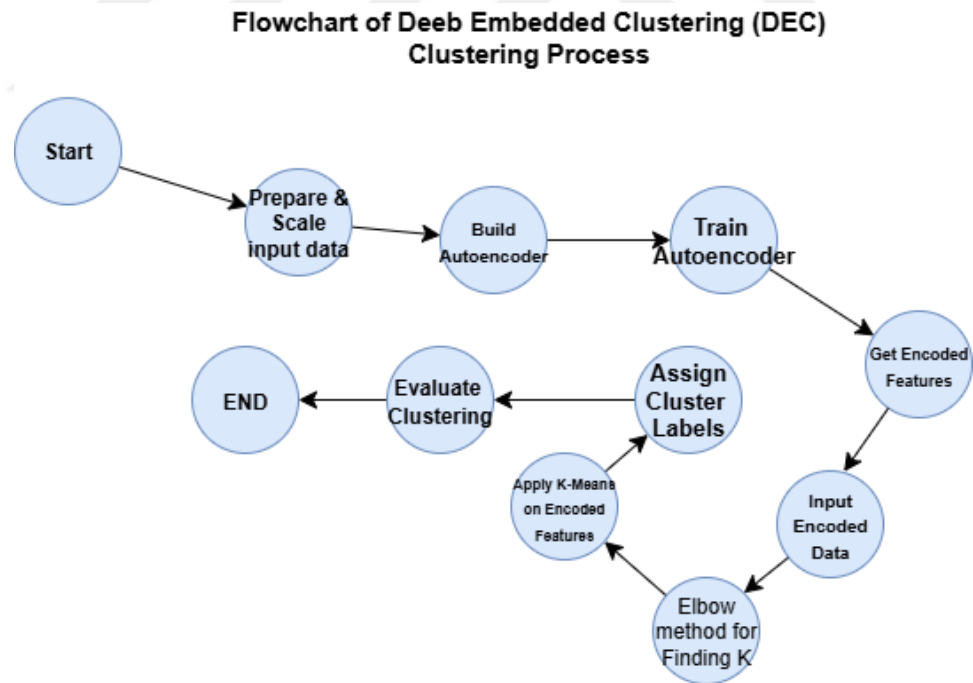
Şekil 4.11'de bu araştırmanın veri yükleme ile başlayıp algoritma değerlendirme sonuçlarına kadar olan akış şeması gösterilmektedir. Süreç, önceden işlenmiş verilerin yüklenmesi ile başlar ve daha sonra `.toarray()` fonksiyonu ile dönüştürülür, böylece algoritma tarafından işlenmeyi amaçlayan seyrekten yoğunla dönüşür. Bir sonraki aşama, verideki boş değerlerin ortalama sütun ile doldurularak daha temiz ve işlemeye hazır hale getirileceği eksik değerlerin hadle edilmesidir. Ayrıca, temizlenen veriler bant genişliği testi için bir deney olarak ilk 1000 veri olarak seçilecektir. Bu aşamada test edilen bant genişliği 0,5 ila 2,0'dır, test edilen her bant genişliğinde, sonuçların Siluet Skoru ile değerlendirilmesi ve işlenmesi için örnek üzerinde Ortalama Kaydırma uydurması gerçekleştirilir. Ayrıca, örnek test sonuçlarından, tüm veri kümesi kullanılarak yeniden uydurulacak nihai bant genişliği seçilecektir. Küme sonuçları daha sonra orijinal veri çerçevesine eklenir ve metrik değerlendirme yapılır.

4.3.4 Deep Embedded Kümelemesi Çalışma Analizi

Algoritma, Python'daki *StandardScaler* kütüphanesini kullanarak verileri katmanlara ayırarak başlar, böylece her özellik aynı ölçeğe sahip olur. Daha sonra veriler, verileri daha kompakt bir temsile sıkıştıran kodlayıcı (*encoder*) ve yeniden yapılandırma hatalarını en aza indirmek için verileri yeniden yapılandıran kod çözücü (*decoder*) olmak

üzere iki ana bölümden oluşan bir otomatik kodlayıcı kullanılarak daha düşük boyutlara dönüştürülür. Bu model, kümeleme işlemi için daha temsili yerleştirmeler (embedding) üretmek amacıyla Adam optimize edici ile derlenmiş ve eğitilmiştir. Eğitildikten sonra, kodlayıcı kısmı çıkarılır ve gömme sonuçları ilk kümelemeyi gerçekleştirmek için K-Means algoritmasında girdi olarak kullanılır.

Bununla birlikte, K-Means doğrusal bir uzayda çalıştığından ve verilerdeki karmaşık yapıları bulmada her zaman optimal olmadığından, kümelerin olasılığını daha iyi dağıtabilen softmax tabanlı bir kümeleme katmanı eklenerek DEC kullanılarak bir ince ayar aşaması gerçekleştirilir. Model, küme temsiliyi iyileştirmek için bir kayıp fonksiyonu olarak Kullback-Leibler Diverjansı kullanılarak derlenir. Yeniden eğitim süreci tamamlandığında, nihai küme etiketlerini belirlemek için tahminler yapılır. Bu yaklaşım, modelin K-Means gibi geleneksel kümeleme yöntemlerine göre daha anlamlı özellikler uyarlamasına olanak tanıyarak verilerdeki karmaşık örüntülerin belirlenmesinde daha etkili olmasını sağlar. Şekil 4.12, bu çalışmada gerçekleştirilen DEC algoritmasının akışını göstermektedir.



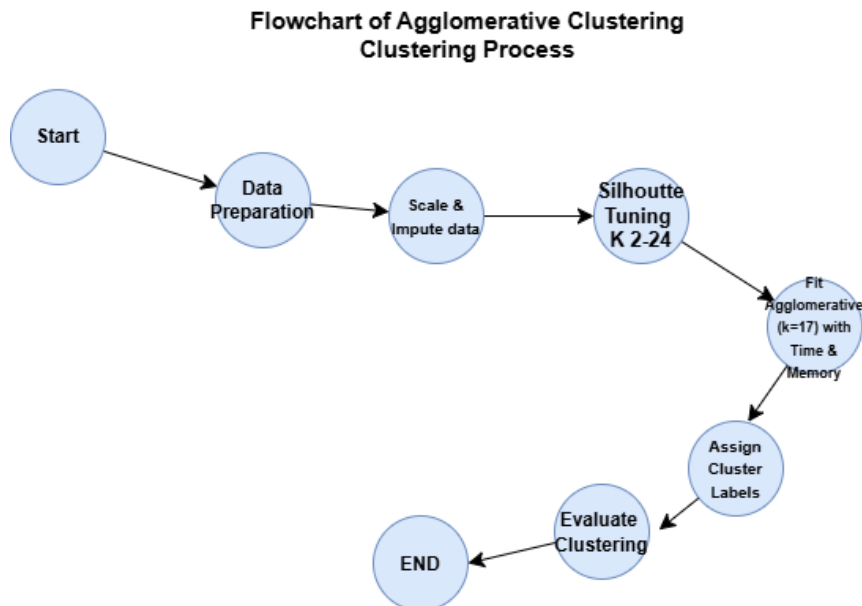
Şekil 4.12. Deep Embedded kümelemesinin akış şeması

Şekil 4.12 yukarıda gösterilen akış şeması, veri hazırlama aşamasıyla başlar. Bu aşamada veriler yüklenir ve temizlenir. Ardından, standartlaştırıcı kullanılarak veriler standartlaştırılır ve işleme hazır hale getirilir. Daha sonra, otomatik kodlayıcıyı eğitme

aşamasına geçilir. Otokodlayıcı, girdi verilerini temel özelliklerine kadar sıkıştırmak (kodlamak) ve ardından bu sıkıştırılmış temsilinden orijinal girdiyi yeniden oluşturmak (kodunu çözmek) için tasarlanmış bir tür sinir ağı mimarisidir (IBM, 2023). Bu kodlayıcının amacı, orijinal verinin önemli özelliklerini koruyan daha düşük boyutlu bir gizli temsil üretmektir. Ardından, model, girdi verisini yeniden yapılandırarak eğitilir. Otokodlayıcı eğitildikten sonra, encoder latent temsil üretir ve bu temsil yeniden girdi olarak kullanılır. Optimum küme sayısını belirlemek için dirsek yöntemi kullanılır, ardından K-Means latent veriler üzerinde çalıştırılır. Küme sonuçları veri çerçevesine her verinin küme etiketi olarak eklenir ve son aşamada küme analizi değerlendirilir.

4.3.5 Aglomeratif Kümelemesi Çalışma Analizi

Aglomeratif kümeleme, alt-üst yaklaşımına dayalı bir hiyerarşik kümeleme yöntemidir. Bu yöntemde, her bir veri başlangıçta tek bir küme olarak ele alınır. Ardından, kümeler yakınlık derecelerine göre aşamalı olarak birleştirilir ve istenen küme sayısına ulaşılan kadar bu işlem tekrarlanır. Bu kümeleme sürecinde, birleştirilen kümeler arasındaki toplam türleri en aza indirmeyi amaçlayan Ward Bağlantı (*ward linkage*) yöntemi kullanılır. Diğer yaklaşımlardan, örneğin Tekli Bağlantı (*Single Linkage*) veya Tam Linkage'den farklı olarak, Ward Bağlantı boyut ve dağılım açısından daha dengeli kümeler oluşturma eğilimindedir. Bu model, veriye imputasyon işlemi uygulanmış veri setine uygulanarak her bir veriye küme etiketi oluşturmak için kullanılır. Aşağıdaki şekil 4.13 aglomeratif çalışmanın akış şemasını göstermektedir.

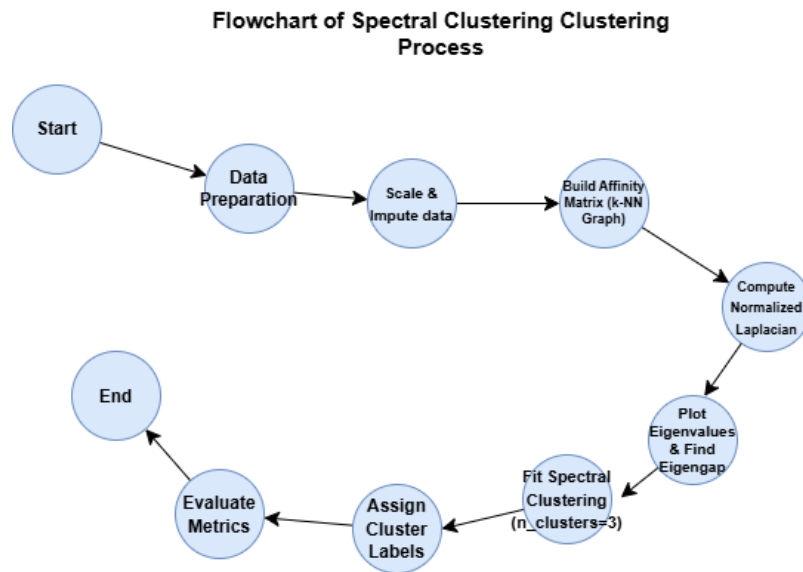


Şekil 4.13. Aglomeratif kümelemesinin akış şeması

Yukarıdaki Şekil 4.13, bu çalışmadaki Aglomeratif algoritmanın akışını göstermektedir. Süreç, veri setinin StandardScaler ile normalleştirileceği ve eksik değerlerin giderileceği veri hazırlama aşamasıyla başlamaktadır. Ardından, optimum küme yapısını bulmak için 2 ila 24 arasındaki k değerlerini analiz etmek üzere Siluet Skoru deneyi gerçekleştirilir. Bundan sonra, Aglomeratif Kümeleme modeli $n_cluster=17$ kümesi ve $linkage='ward'$ ile çalıştırılmıştır çünkü önceki deneysel deneme en uygun değeri üretmiştir. Kümeler oluşturulduktan sonra, her veri bir küme etiketi ile etiketlenir ve ardından son aşamada Aglomeratif algoritma değerlendirilir.

4.3.6 Spektral Kümeleme Çalışma Analizi

Spektral kümeleme için hazır olan verilerde, $n_cluster = 3$ olacak şekilde küme sayısı belirlenir. Bu işlem, $affinity = 'nearest_neighbors'$ parametresi kullanılarak gerçekleştirilir. Bu parametre, veriler arasındaki yakınlığı en yakın komşuluk ilişkisine göre hesaplar. Spektral kümeleme, verileri laplacian grafiye dönüştürerek çalışır. Bu grafikte her veri bir düğüm olarak kabul edilir ve veriler arasındaki ilişkiler yakınlık ağırlıklarına göre belirlenir. Ardından, afinite matrisine öz ayrışım (eigen decomposition) uygulanarak verilerin daha yapılandırılmış bir eigen uzayında temsil edilmesi sağlanır. K-Means algoritması kullanılarak bu eigen uzayında veriler spektral bileşenlerine göre kümelendir. Şekil 4.14, bu çalışmada uygulanan Spektral Kümeleme algoritmasının akış şemasıdır.



Şekil 4.14. Spektral Kümeleme kümelemesinin akış şeması

Şekil 4.14, veri kümesinin StandardScaler kullanılarak normalleştirileceği ve boş değerlerin atandığı veri hazırlama ile başlayan spektral kümeleme aşamalarını göstermektedir., simetrikleştirilen k-en yakın komşu grafiği yaklaşımı kullanılarak bir yakınlık matrisi (affinity matrix) oluşturulur. Bu matristen, spektral temsilin temelini oluşturan normalize edilmiş Laplasyen matrisi hesaplanır. Laplasyen oluşturulduktan sonra, öz ayrışım (eigen decomposition) işlemi uygulanarak özdeğerler elde edilir ve eigengap yöntemi kullanılarak optimal küme sayısı belirlenir. Belirlenen küme sayısına göre spektral kümeleme algoritması uygulanır ve elde edilen sonuçlar veriye küme etiketleri atamak için kullanılır. Son olarak, algoritmanın performansı çeşitli değerlendirme metrikleri ile analiz edilir.



5. ARAŞTIRMA BULGULARI VE TARTIŞMA

Teknoloji hayatın birçok alanında olduğu gibi eğitim alanında da çok ve yaygın bir şekilde kullanılmaktadır. Bu bölümde araştırmanın sonuçları ve bu tez çalışmasında kullanılan her bir algoritmanın detaylı bulguları açıklanmaktadır.

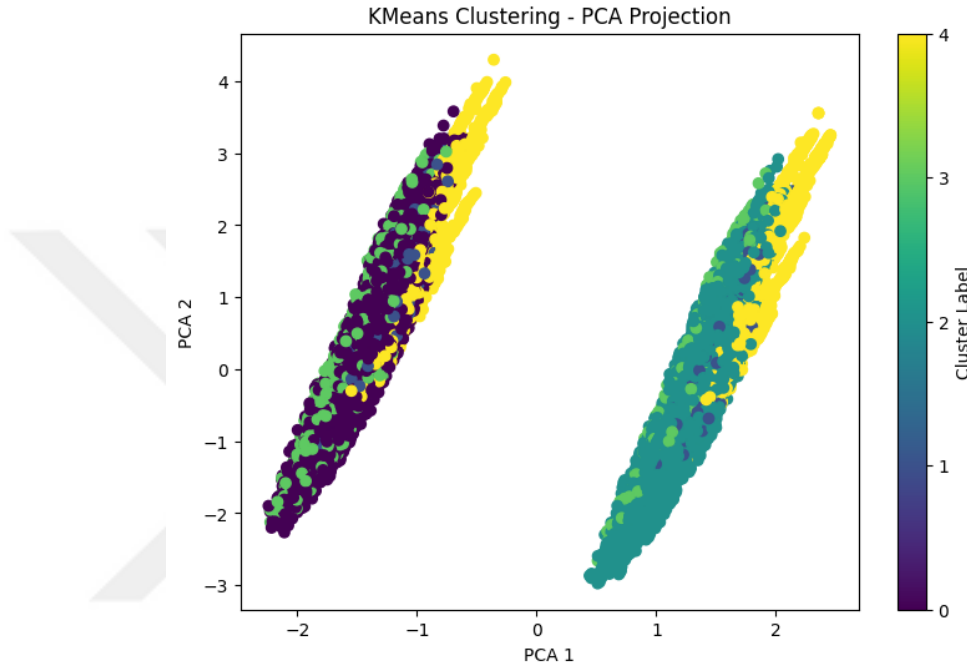
Büyük veri, iş dünyası ve teknolojinin geniş bir alanını kapsayan bir trenddir. Büyük veri, geleneksel teknoloji veya altyapının etkili bir şekilde işlemesi için çok zor olan çeşitli, hızlı değişen ve hacimli verileri içeren teknolojileri ve girişimleri ifade eder.

Bu bölümde, bu araştırma bir sonraki bölümde bahsedilen çeşitli kümeleme algoritmalarını karşılaştırmıştır. Bu araştırmada, algoritmanın büyük veriyi ele alırken ihtiyaç duyduğu yetenekler ölçeklenebilirlik, yürütme süresi ve performanstır. Bu araştırma, her bir algoritmayı değerlendirirken çeşitli analizler kullanmaktadır. Bu araştırmada aşağıdaki değerlendirme kullanılmıştır:

- A. Siluet (*Silhouette*) Skoru, bir nesnenin diğer kümelerle kıyasla kendi kümesine ne kadar benzer olduğunu ölçmek için kullanılır. Bu değer -1 ile +1 arasında değişir; Siluet puanı ne kadar yüksekse, nesne diğer kümelerle kıyasla kendi kümesine o kadar benzerdir.
- B. *Indeks Davies Bouldin* (DBI) teriminin Türkçesi Davies-Bouldin Endeksi veya güvenilirlik indeksi olarak da adlandırılır. DBI, her bir küme için ortalama kümeler arası mesafenin en yakın komşu kümeye oranıdır. DBI bir kümeleme metriği olarak kullanılır çünkü genel doğrulama kümeleme yöntemi, kümeleme sonuçlarının performansını değerlendirmek için iç ve dış olmak üzere iki ana kategori içerir. (Sopyan ve ark., 2022).
- C. *Dunn Index* teriminin Türkçesi Dunn Endeksi, kümelerin yoğunluğunu ve ayrılmasını dikkate alarak kümelemenin geçerliliğini değerlendirmek için kullanılan bir metriktir. (S. Lee, 2025).
- D. Büyük veride hesaplama süresi ve verimlilik önemlidir. Bu durumda saniye birimleri kullanılır.
- E. Ölçeklenebilirlik, algoritmanın büyük miktarda veriyi ne kadar iyi işlediğidir. Bu durumda MiB (Mebibyte) birimleri kullanılarak

5.1. K-Means Bulguları

K-Means algoritmasının sürece uygulanması, yapılan işlemin kümelenmesini belirlemek için kullanılır. Konferans katılımcısı verilerinin işlenmesi, $n_clusters=5$ ile tanımlanan 5 kümeye ayrılacaktır. Aşağıdaki Şekil 5.1, 0,1,2,3 ve 4 kümeli K-Means modeli ile algoritmaya dayalı olarak her bir kümedeki toplam veri dağılımını açıklamaktadır.



Şekil 5.1. K-Means veri dağılım grafiği

Şekil 5.1, K-Means algoritması kullanılarak yapılan kümeleme sonuçlarının PCA (*Principal Component Analysis*) le görselleştirilmesini göstermektedir. PCA, veri keşfi, görselleştirme ve veri işleme uygulamalarında kullanılan doğrusal bir boyut indirgeme tekniğidir (Wikipedia, 2025).

Yukarıdaki grafikte yer alan her bir nokta, K-Means algoritması tarafından başlangıçta 5 küme halinde gruplandırılan veri kümesinden bir veri noktasını temsil etmektedir. 0'dan 4'e kadar olan sınıfları gösteren 5 renk vardır, ancak alanlar kümeler arasında örtüşme gösterir ve gruplar arasındaki sınırların tamamen kesin olmadığını gösterir. Aşağıda bu algoritmadaki renklerin bir açıklaması yer almaktadır:

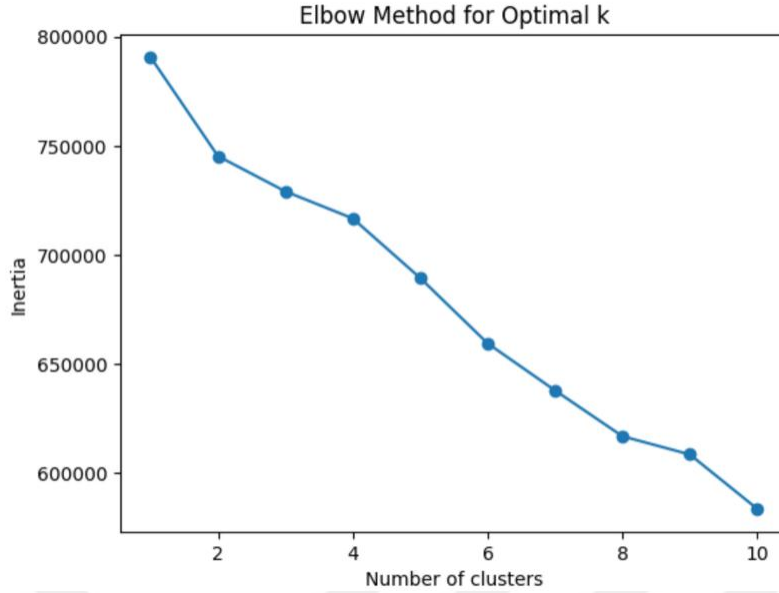
1. Sınıf 0 koyu mor renkli noktalarla temsil edilmektedir. Bu noktaların çoğunlukla grafiğin sol alanında yoğunlaşması, bu kümedeki verilerin oldukça

güçlü bir özellik benzerliğine sahip olduğunu ve diğer kümelerin bazılarının uzamsal olarak benzer olduğunu göstermektedir.

2. Sınıf 1 koyu mavi noktalarla temsil edilmektedir. Bu noktalar çoğunlukla grafiğin sağ tarafına dağılmıştır. Bu sınıf bir ana küme oluşturur, ancak yine de diğer kümelere bitişiktir.
3. Küme 2, turkuaz yeşili ile temsil edilmekte olup bu renk küme 1 ile birleşiyor gibi görünmekte ve küme 1 ile 2 arasında bir örtüşme olduğunu göstermektedir.
4. Açık yeşil renkle temsil edilen 3. küme, iki büyük kümenin kenarında yer almaktadır. Hem solda hem de sağda. Bu nokta genellikle farklı bir karaktere işaret eden kenardadır.
5. Sarı renkle temsil edilen Küme 4, ağırlıklı olarak iki büyük kümenin dış bölgelerinde yer almaktadır. Bu dağılım, ana gruba uymayan azınlık verilerini veya aykırı değerleri yansıtmaktadır.

PCA uzayında hem sağda hem de solda coğrafi olarak iki ana grubun ana hatları. Bu ana grupların her biri renkle ayırt edilen birkaç alt kümeden oluşmaktadır. Renk dağılımı tamamen temiz olmasa da K-Means, PCA uzayında şekil ve noktalara dayalı olarak verilerdeki örüntüleri bulmayı başarmıştır. Bununla birlikte, kümeler arasında örtüşme vardır ve bu da ayrımın tamamen net olmadığını gösterir. Bu durum, verilerin karmaşık veya doğrusal olmayan özelliklere sahip olması ve algoritma tarafından en iyi şekilde ayrılamayan verilerle sonuçlanması halinde ortaya çıkabilir.

Ayrıca, bu çalışmada, optimum küme sayısının belirlenmesinde, küme sayısının artırılmasının artık atalette önemli bir azalma sağlamadığı noktayı belirlemek için Elbow yöntemi kullanılmıştır. Aşağıdaki şekil 5.2, küme sayısı ile K-Means algoritması tarafından üretilen atalet değeri arasındaki ilişkiyi göstermektedir



Şekil 5.2. Optimum k için Dirsek Yöntemi

Şekil 5.2, $k=1$ 'den $k=5$ 'e kadar eylemsizlikte keskin bir düşüş olduğunu ve bu noktadan sonra düşüşün artık şiddetli olmadığını açıklamaktadır. Bu, 5 küme kullanımının model karmaşıklığı ve kümeleme kalitesi arasında en iyi dengeyi sağlayabileceğini göstermektedir. Bu nokta, segmentasyon işleminde optimum küme sayısının belirlenmesinde referans olarak kullanılabilir. Küme sayısına bağlı olarak değerlendirme, kümeleme kalitesinin çeşitli metrikler kullanılarak değerlendirilmesiyle devam eder. Çizelge 5.1, K-Means'in verileri ne kadar iyi kümelediğinin nicel bir tanımını göstermektedir.

Çizelge 5.1. K-Means Algoritması için Değerlendirme Metrikleri Çizelge

Siluet Skoru	0.04319
Davies-Bouldin İndeksi	3.38706
Atalet (Inertia)	689401.894
Time Elapsed	0.4757 seconds
Memory Use	1669.07 MiB
Dunn Index	0.2617
Toplam Küme Sayısı	5

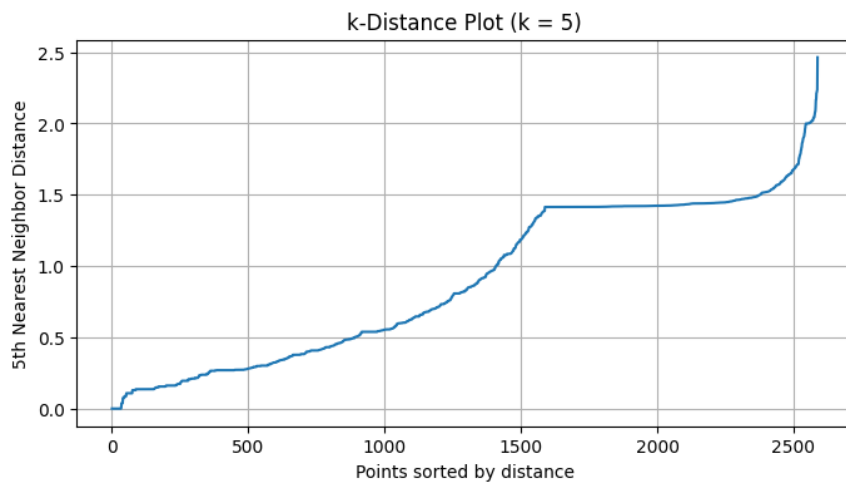
Çizelge 5.1'e göre K-Means algoritmasının performans değerlendirme sonuçları analiz edilmiştir. Silhouette Score değerinin 0.04319 olması, kümeleme örüntüsü oluşmasına rağmen küme ayrımının hala nispeten zayıf olduğunu göstermektedir. K-Means algoritmasında elde edilen Davies Bouldin Endeksi 3.38706'dır ve bu da kümeler

arasındaki benzerlik seviyesinin oldukça yüksek olduğunu, dolayısıyla gruplar arasındaki ayırım kalitesinin optimal olmadığını göstermektedir. Bu arada, Atalet değeri 689401.894 olarak kaydedilmiştir ve bu değer her noktadan küme merkezine olan toplam karesel uzaklığı göstermektedir ve bu değer ne kadar küçükse genellikle daha kompakt bir kümeye işaret etmektedir. Modelin işlem süresi 0.4757 saniye ile nispeten hızlıdır ve 1669.07 MiB bellek kullanımı, kullanılan veri ölçeği için kabul edilebilir verimliliğe işaret etmektedir. Dunn Endeksi değeri olan 0,2617, değer çok yüksek olmasa da kümeler arasındaki mesafenin en büyük küme boyutuna kıyasla nispeten yeterli olduğunu yansıtmaktadır.

5.2. DBSCAN Algoritmaları Bulguları

DBSCAN yoğunluk tabanlı bir kümeleme algoritmasıdır ve başlangıçta küme sayısının belirlenmesini gerektirmez. (Simbolon & Friskila, 2024). DBSCAN kümeleri belirlerken epsilon (ϵ) ve minPts (minPts) parametrelerini kullanır. Bu iki değer kümeleme değerlendirmesinin başarısını belirleyecektir.

Araştırmacılar, bu algoritma üzerinde optimum test yaparken, eps (ϵ) için optimum değeri belirlemek ve minimum nokta değerini (*min_samples*) belirlemek için n'inci kümelenmiş komşu mesafe grafiği yaklaşımını kullandılar. K-Mesafe Grafiği, her bir yiyiğin en yakın komşusuna olan mesafesini gösteren bir grafikdir ve genellikle $k=\text{min_samples}$ olarak başlatılır. Bu çalışmada $k=5$ kullanılmıştır çünkü önceki K-Means çalışması $k=5$ değeri ile başlatılmaktadır. Şekil 5.3, maksimum sonuç sağlamak için eps olarak kullanılmış k-distance grafiğinin sonucudur.



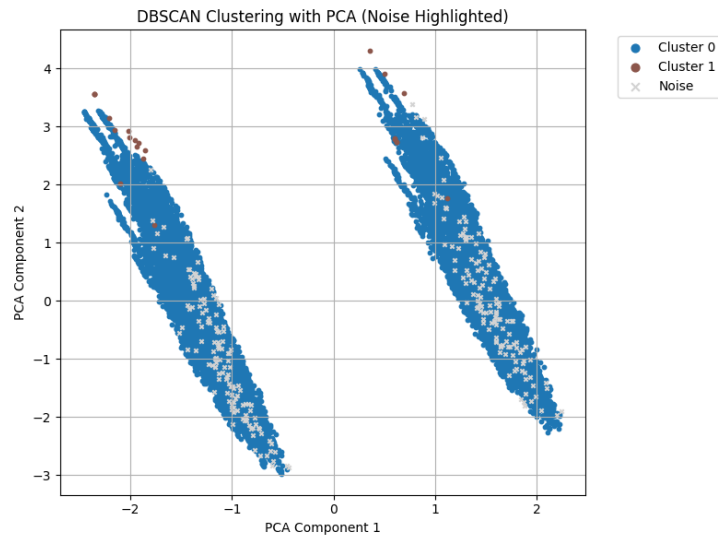
Şekil 5.3. DBSCAN için k-Mesafe Grafiği ($k = 5$)

Yukarıdaki Şekil 5.3 $k=5$ değeri ile k-mesafe grafiğini göstermektedir, 1,5 mesafe değerinde net bir dirsek noktası vardır. Bu nokta, noktalar arasındaki mesafe dağılımında önemli bir değişiklik göstermektedir, bu nedenle 1,5 optimum *eps* değeridir. *min_sample=5* değeri ise genel standartlara uygun olarak korunmuştur. Daha sonra bu değerler DBSCAN algoritmasının uygulanmasında kullanılır. Şekil 5.4 örnek bir uygulamadır.

```
# 3. Fit DBSCAN
dbscan = DBSCAN(eps=1.5, min_samples=7)
dbscan.fit(X_imputed)
labels = dbscan.labels_
```

Şekil 5.4. $\text{eps} = 1.5$ ve $\text{min_samples} = 7$ Parametreleriyle DBSCAN Modelinin Eğitilmesi

Şekil 5.5, PCA grafiği kullanılarak DBSCAN tarafından oluşturulan veri dağılımının sonucudur. Şekil 5.4, optimum kümeleme sonuçları için kullanılan eps ve $\text{min_samples}=5$ parametreleri için kullanılan uygulamadır. Model eğitim süreci `fit()` fonksiyonu ile gerçekleştirilmekte ve küme sınıflandırma sonuçları `labels_` algoritmasından elde edilmektedir, burada her veri noktası kümelerden birine kategorize edilecek ve gürültü olarak kabul edilirse -1 etiketi verilecektir. Şekil 5.5, PCA grafiği kullanılarak DBSCAN tarafından oluşturulan veri dağılımının sonucudur.



Şekil 5.5. PCA grafikleri ile DBSCAN nokta yayılımı

PCA projeksiyonu ile DBSCAN kullanılarak yapılan kümelemenin görselleştirilmesi 2 farklı kümenin (Sınıf 0 ve 1) oluştuğunu göstermektedir. Şekil 5.5'te bulunan renge göre verilerin dağılımı aşağıdaki gibidir Koyu mavi renk (Sınıf 0) grafiğe hakimdir ve dağılımın merkezini oluşturan birbirine yakın diğer veriler arasındaki en büyük kümeyi göstermektedir.

1. Mavi renk (Sınıf 0) grafiğin hem sol hem de sağ büyük kümelerine eşit olarak dağılmıştır ve bunlar aynı zamanda ana kümenin de büyük parçalarıdır. Bu, DBSCAN'ın yoğunluğun yakınlığına dayalı olarak verilerin çoğunu başarıyla kümelediğini göstermektedir.
2. Sınıf 1'i temsil eden kahverengi renk daha az görünürdür ve iki büyük kümenin tepesine dağılmıştır. Bu, DBSCAN'ın farklı yoğunlukta bir bölge olarak tanıdığı ek bir küçük kümeyi gösterir.
3. Şekil 5.5'te X sembolü ile temsil edilen gri renk, herhangi bir küme olarak kabul edilecek kadar diğer noktalara yakın olmayan outlier veya gürültü noktalarını temsil etmektedir.

Çizelge 5.2, DBSCAN algoritmasını niceliksel olarak değerlendirmek için DBSCAN algoritmasının analitik metriklerini detaylandırmaktadır.

Çizelge 5.2. DBSCAN algoritması üzerindeki metrik değerlendirme sonuçları

Siluet Skoru	0.11495
Davies-Bouldin Endeksi	9.9690
Atalet (Inertia)	n/a
Geçen Süre	0.62892
Bellek Kullanımı	1684.203125 MiB
Dunn Endeksi	0.225150
Toplam Küme	2
Gürültü noktası	218

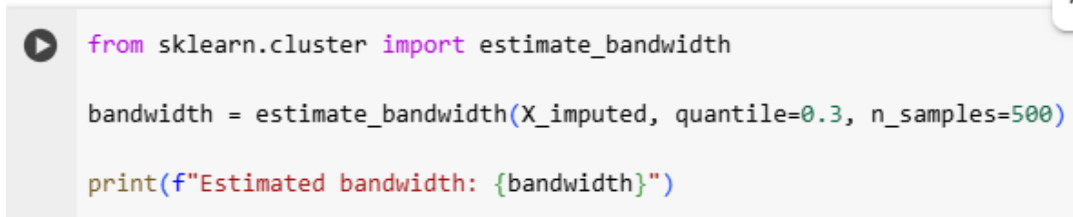
Çizelge 5.2'ye göre, DBSCAN algoritmasının performans değerlendirmesi 258 adet gürültü olarak tanımlanan veri ile 2 küme oluşturmuştur. Siluet Skoru 0,11495, nispeten düşük bir ayrım olduğunu göstermektedir. Bu değer, optimal değere (1'e yakın) göre daha düşük olup, küresel olmayan küme şekilleri veya karmaşık veri dağılımından etkilenebilir. Öte yandan, Davies-Bouldin Endeksi 9,9690 olarak kaydedilmiştir, bu oldukça yüksek bir değerdir. Bu değer, kümelerin kalitesinin nispeten kötü olduğunu ve kümeler arasında çakışma veya merkezi olmayan dağılım olduğunu gösterir. Dunn

İndeksi 0,225150 olarak kaydedilmiştir ve kümeler arasındaki minimum mesafenin en büyük küme çapına kıyasla çok önemli olmadığını gösterir. Genel olarak, düşük Dunn İndeksi değeri kümelerin çok net bir şekilde ayrıldığını gösterir, bu da diğer metriklerin bulguları ile tutarlıdır. Bu algorithmada inertia değeri önemsizdir çünkü DBSCAN, K-Means gibi merkez nokta tabanlı algorithmalar için geçerlidir. Teknik performans açısından, bu algorithma 0,62892 saniye sürmüş ve 1684,29 MiB bellek kullanmıştır, bu da büyük ölçekli veri kümeleri için hala nispeten hızlıdır.

5.3. Mean Shift Algoritmaları Bulguları

Mean-Shift, yüksek yoğunluklu alanların belirlenmesinde pencere kaydırma yöntemini kullanarak verilerin dağılımı yoluyla merkezleri veya sözde "nokta" bulmaya odaklanan bir güç kümelemesidir (Utari ve ark., 2024). DBSCAN'den farklı olarak Mean Shift algorithmasını kullanmak için en önemli parametre bant genişliğinin belirlenmesidir.

Bu araştırma, Scikit-learn kütüphanesinde yer alan bir fonksiyonu kullanmaktadır; bu kütüphane, Mean Shift algorithmasında uygulanmış olan *estimate_bandwidth()* fonksiyonunu kullanmaktadır. Şekil 5.6, kod satırında uygulandığı şekliyle işlevi göstermektedir.



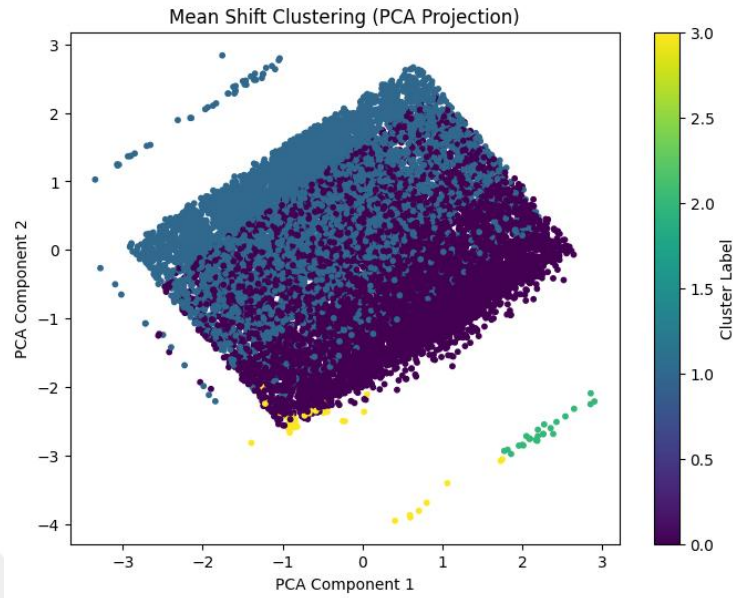
```
from sklearn.cluster import estimate_bandwidth

bandwidth = estimate_bandwidth(X_imputed, quantile=0.3, n_samples=500)

print(f"Estimated bandwidth: {bandwidth}")
```

Şekil 5.6. Python'da *estimate_bandwidth* uygulamasının kod satırı

Şekil 5.6'teki fonksiyon üzerindeki tahmin sonucu 2.63'tür ve bu değer kümeleme işleminde ana parametre olarak kullanılmıştır. Bu çalışmadaki verilerle uygulanan Ortalama Kaydırma algorithması 8 sınıf ile sonuçlanmıştır. Şekil 5, bu çalışmada Ortalama Kaydırma Kümeleme algorithması kullanılarak veri dağılımının görselleştirilmesini göstermektedir.



Şekil 5.7. Mean shift algoritması kullanılarak veri dağılımı üzerinde PCA grafiği

Şekil 5.7, Ortalama Kaydırma algoritmasının bir küme ve birkaç küçük dağınık küme oluşturma eğiliminde olduğunu göstermektedir. Bu, yüksek yoğunluklu alanları arayan Ortalama Kaydırma algoritmasının özellikleriyle uyumludur. PCA görselleştirmesi, sadece 4 küme oluşmasına ve kompozisyonun dengeli olmamasına rağmen bunu göstermeye yardımcı olmaktadır. Bu sonuçların kalitesini değerlendirmek için, Çizelge 5.3'te DBSCAN algoritmasının değerlendirme metrikleri açıklanmaktadır.

Çizelge 5.3. Mean Shift algoritması üzerindeki metrik değerlendirme sonuçları

Siluet Skoru	0.0551
Davies-Bouldin İndeksi	2.2912
Dunn İndeksi	0.02406
Geçen Süre	43.5792 seconds
Bellek Kullanımı	1551.69 MiB
Toplam Küme Sayısı	4

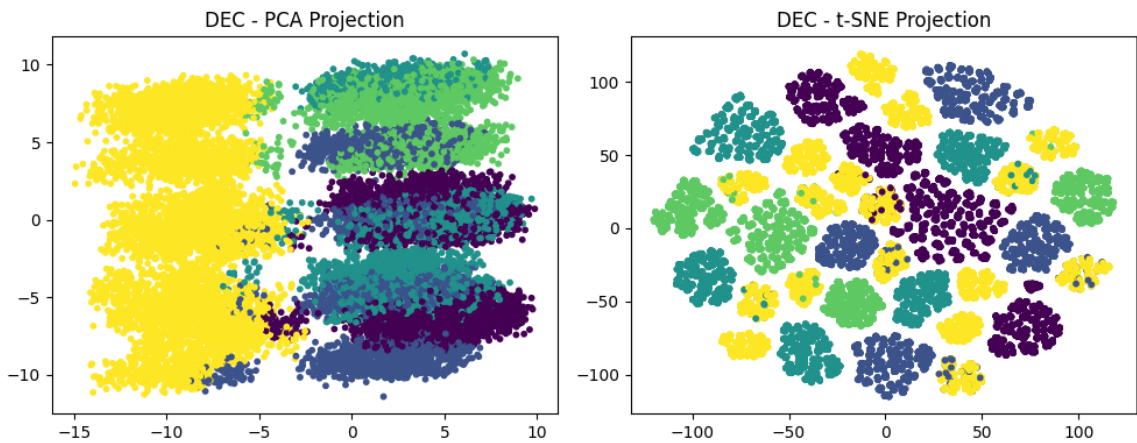
Çizelge 5.3, 0,0551'lik düşük bir Siluet Puanını göstermektedir ve kümeler arasındaki sınırların hala belirsiz olduğunu açıklamaktadır. Birçok veri noktası kümeler arasındaki sınırda veya kümeler içinde net olmayan sınırlarda olabilir. Ardından, 2,2912'lik Davies Bouldin Endeksi değeri oldukça yüksek bir sayı göstermekte ve kümelerin üst üste geldiğini veya çok geniş olduğunu, dolayısıyla küme ayırma kalitesinin optimal olmadığını tanımlamaktadır. Bu durum, kümeler arasında zayıf bir

ayırımı veya büyük ve dağınık bir kümeyi gösteren 0,02406'lık Dunn endeksi tarafından da desteklenmektedir. Yoğunluk tabanlı kümeleme ve büyük veri kümeleri için 43.5792 saniyelik zaman ve 1.5 Giga baytlık bellek normaldir. Bu da algoritmanın kümeleme işlemini tamamlamadaki verimliliğini göstermektedir. Bu kümeleme 4 küme ile sonuçlanmıştır, PCA görüntü 5'te açıkça görülebilen sadece 2 büyük küme varken diğerleri çok küçüktür, ancak bu durumda kümeler dengesiz hale gelmiştir.

5.4. Deep Embedded Algoritmaları Bulguları

Deep Embedded kümeleme, derin sinir ağları kullanarak temsil öğrenme ve kümelemeyi aynı anda birleştiren bir algoritmadır. (Artana ve ark., 2023; Xie ve ark., 2015). DEC yöntemi, göz boyutunu azaltmak için kullanılan kodlayıcının ağırlıklarını elde etmek için gerçekleştirilen otokodlayıcı öğrenme aşaması ve başlangıç merkezini elde etmek için kullanılan K-Means yöntemi olmak üzere iki aşamadan oluşmaktadır. (Pradana, 2020).

Bu çalışmada, çalışmadaki verilerin üzerindeki 2 aşamadan önce model, kayıp değerler olmayacak şekilde tekdüze hale getirilmiştir. Bir sonraki aşama, 10 gizli boyuta sahip tek katmandan oluşan bir otokodlayıcı mimarisi oluşturmaktır. Tek katmanlı kod çözücü kullanımı, orijinal girdiyi yeniden yapılandırmayı amaçlamaktadır. Daha sonra otomatik kodlayıcı, kodlayıcı ekranında sağlam ve bilgilendirici bir gizli temsil elde etmek için 256'lık bir yığın boyutuyla 50 epok boyunca eğitilir ve daha sonra bu model DEC modeli kümeleme işlemi için kullanılacaktır. İkinci aşama, K-Means algoritmasını 3 küme ile gerçekleştirmektir. Şekil 5.8'te DEC yöntemi ile küme dağılımının görselleştirilmesi gösterilmektedir.



Şekil 5.8. Deep Embedded algoritması kullanılarak veri dağılımı üzerinde PCA grafiği

Şekil 5.8'de DEC, PCA ve t-SNE (*t-distributed Stochastic Neighbor Embedding*) olmak üzere iki projeksiyon aracılığıyla gösterilmektedir. t-SNE, yüksek boyutlu verileri anlaşılması kolay iki veya üç boyuta dönüştüren bir görselleştirme tekniğidir (Helmi, 2024). Sol tarafta, PCA kullanılarak yapılan görselleştirme, verilerin doğrusal iki boyutlu bir uzaydaki dağılımını göstermektedir. Her nokta bir veri noktasını temsil etmektedir ve DEC tarafından oluşturulan kümeye göre renklendirilmiştir. Kümeler arasında bir örtüşme vardır, bu da bu doğrusal gösterimde gruplar arasındaki ayrımın tamamen net olmadığını gösterir. Bu arada, sağ taraf aynı sonuçları, veriler arasındaki yerel ayrımı en üst düzeye çıkarmak için tasarlanmış doğrusal olmayan bir teknik olan t-SNE (*t-dağıtılmış Stokastik Komşu Gömme*) kullanarak görselleştirmektedir. t-SNE grafiğinde kümeler daha ayrık görünmekte ve net kümeler oluşturmaktadır; bu da DEC'in küme yapısını daha karmaşık boyutlarda iyi bir şekilde haritalayabildiğini göstermektedir. Dolayısıyla, t-SNE, DEC sonuçlarının küme ayırma kalitesinin PCA'ya göre daha temsili bir resmini sunmaktadır. Çizelge 5.3 DEC algoritmasının metrik değerlendirmesini açıklamaktadır.

Çizelge 5.4. Deep Embedded algoritması üzerindeki metrik değerlendirme sonuçları

Siluet Skoru	0.05511
Davies-Bouldin Endeksi	2.29122
Dunn Endeksi	0.01429
Atalet (<i>Intertia</i>)	6694070.00
Geçen Süre	43.5792 seconds
Kullanılan Bellek	1551.69 MiB
Toplam Kümeler	4

Çizelge 5.4, 0,1477'lik Siluet Skoru değerinin düşük olduğunu ve kümeler arasındaki sınırların net bir şekilde oluşmadığını göstermektedir. Bu durum, kümeler arasında bir örtüşme olduğunu gösteren 2.29122 Davies-Bouldin İndeks değeri ile de desteklenmektedir. Çok düşük olan 0,0041 Dunn Endeksi, kümeler arasındaki mesafenin her bir kümenin iç dağılımına kıyasla nispeten küçük olduğunu göstermektedir. Ayrıca, DEC uygulamasında atalet kullanılarak değerlendirilen her bir veri noktasından ilgili küme merkezlerine olan toplam mesafe miktarının ölçülmesi açısından atalet yalnızca diğer algoritmalarla karşılaştırma yapmak için ek bir gösterge olarak kullanılmaktadır. Bir sonraki metrik 1.5 GB yer kaplayan bellek kullanımıdır ve bu da bu modelin yeterince büyük bir veri kümesi veya karmaşık kodlayıcı mimarisi ile çalıştığını gösterir, bu aynı

zamanda diğer algoritmalarından daha uzun süre gereken zamanla doğru orantılıdır çünkü sırayla otomatik kodlayıcı ve K-Means içerir.

5.5. Aglomeratif Algoritmaları Bulguları

Aglomeratif Kümeleme, aşağıdan yukarıya (bottom up) bir yaklaşımla hiyerarşik kümeleme yöntemine bir örnektir (Artana ve ark., 2023). Optimum küme sayısını belirlerken, araştırmacı k değerini test etmek ve aşağıdaki Şekil 5.9'da görselleştirildiği gibi değerlendirme metriklerini karşılaştırmak için bir iç değerlendirme yöntemi yürütmüştür.

```
from sklearn.metrics import silhouette_score

for k in range(2, 11):
    agg = AgglomerativeClustering(n_clusters=k, linkage='ward')
    labels = agg.fit_predict(X_scaled_imputed)
    sil = silhouette_score(X_scaled_imputed, labels)
    print(f"k = {k} → Silhouette Score = {sil:.4f}")
```

Şekil 5.9. Optimum k değerinin belirlenmesinde iç değerlendirme

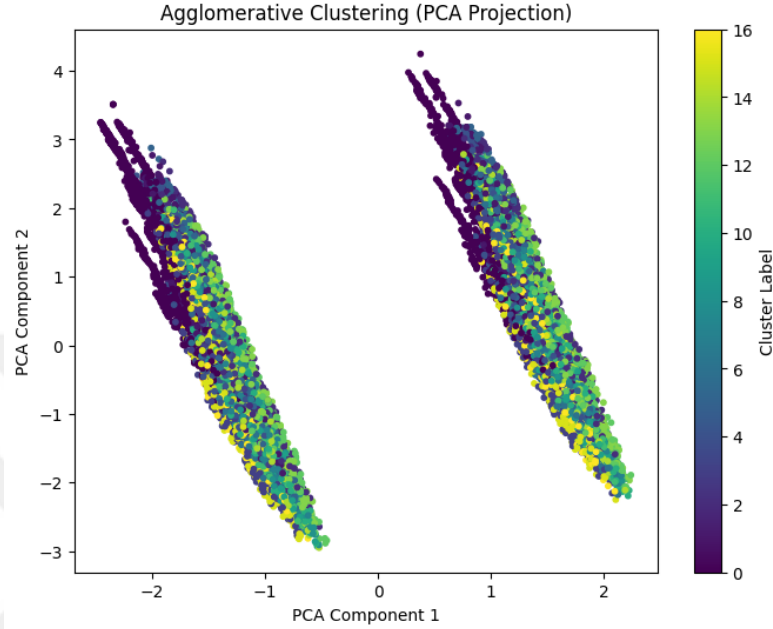
Aglomeratif kümelemede çeşitli K değerleri için Silhouette skor testi sonuçlarının kod satırını gösteren Şekil 5.9'dan, aşağıdaki Çizelge 5.5'te gösterilmiştir.

Çizelge 5.5. Sonuçlar k değerine dayalı siluet skoru

Toplam k	Siluet Skoru	Toplam k	Siluet Skoru
2	-0.0072	14	0.2115
3	0.0102	15	0.2293
4	0.0297	16	0.2471
5	0.0492	17	0.2647
6	0.0682	18	0.2275
7	0.0864	19	0.2182
8	0.1045	20	0.2077
9	0.1225	21	0.1998
10	0.1404	22	0.1871
11	0.1582	23	0.1794
12	0.1758	24	0.1815
13	0.1936		

Çizelge 5.5, Silhoutte değeri ne kadar yüksekse ideal kümeler arasındaki ayrımın o kadar iyi olduğunu göstermektedir çünkü 1'e yakındır. Yukarıdaki verilerden, 8-10'un

iç yapı bakımından birbirinden oldukça farklı doğal kümeler olması muhtemeldir. Bu nedenle, bu araştırmada k değerinin başlatılmasında en yüksek değer, yani $k = 10$ 'un kullanılması kararlaştırılmıştır. Aglomeratif algoritma uygulandıktan sonra veri dağılımı için Şekil 5.10'da PCA grafiği gösterilmiştir.



Şekil 5.10. Aglomeratif algoritması kullanılarak veri dağılımı üzerinde PCA grafiği

Şekil 5.10'deki 10 başlangıç kümeli görselleştirme, verilerin çoğunun yoğun bir doğrusal model oluşturduğunu göstermektedir. Bazı kümeler oldukça iyi dağılmış görünmektedir, ancak bazı alanlarda kümeler arasındaki ayrımın tam olarak optimal olmadığını gösteren renk örtüşmesi vardır. Yine de bu sonuç, algoritmanın basit olmayan mekansal örüntülere sahip verilerin gizli yapısını belirleme yeteneğini göstermektedir. Daha ileri bir değerlendirme için Çizelge 5.6, Aglomeratif algoritma kullanılarak yapılan metrik değerlendirmenin ayrıntılarını göstermektedir.

Çizelge 5.6. Aglomeratif algoritması üzerindeki metrik değerlendirme sonuçları

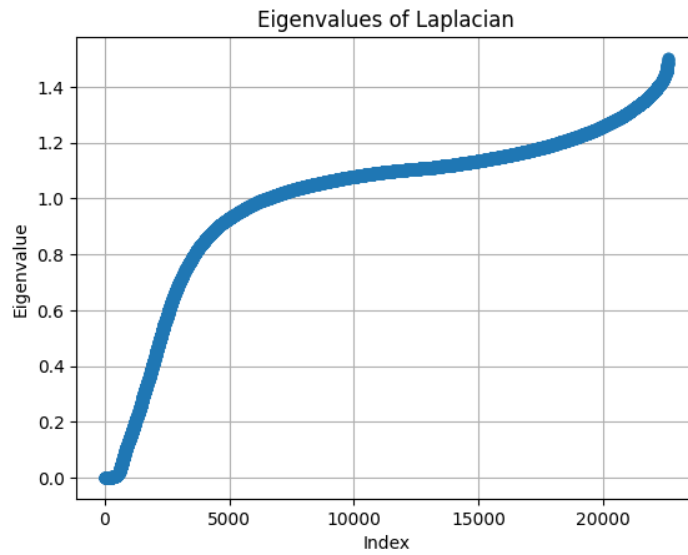
Siluet Skoru	0.2645
Davies-Bouldin Endeksi	1.5168
Dunn Endeksi	0.5330
Geçen Süre	23.6844 seconds
Kullanılan Bellek	5401.72 MiB
Toplam Küme	17

Aglomeratif Kümeleme algoritmasına ilişkin Çizelge 5.6, kümeler arasındaki ayrımın tam olarak optimal olmadığını göstermektedir. 0,1615 Siluet Skoru ve 1,625 Davies-Bouldin İndeksi, kümeler arasında hala örtüşme olduğunu göstermektedir. Bununla birlikte, hızlı işlem süresi (0,349 saniye) algoritmayı büyük verilerde kullanım için verimli kılmaktadır. Düşük Dunn Endeksi (0,0639), kümeler arasındaki mesafenin her bir kümenin iç yayılımına kıyasla hala nispeten küçük olduğunu göstermektedir. Dolayısıyla, Aglomeratif Kümeleme hesaplama açısından iyi çalışmasına rağmen, kümelemenin yapısal kalitesi hala geliştirilebilir.

5.6. Spektral Algoritmaları Bulguları

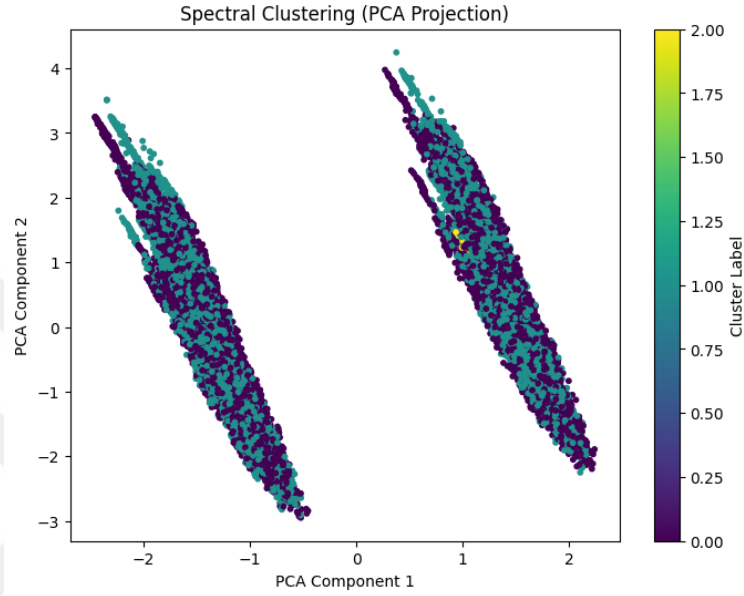
Spektral kümeleme normalleştirilmiş bir kırpma algoritmasıdır. Her bir veri noktasını bir grafik düğümü olarak oluşturur ve ardından problemi grafik bölümleri halinde bir kümeleme problemine dönüştürür (Jordan, 2004). Bu algoritmanın uygulanmasını optimize etmek için bu araştırma, optimum n değerini elde etmek amacıyla k . ve $k+1$ özdeğeri arasındaki farkı bulmak için laplacian grafiğinin özdeğer ayrışımını kullanmaktadır. Doğrusal cebirde, doğrusal bir operatörün özdeğeri, artan sırada düzenlenmiş iki ardışık özdeğer arasındaki farktır (Wikipedia, 2023).

Python kullanarak spektral algoritmayı uygularken, optimum K değerini belirlemek için Laplacian'ın özdeğerlerini elde etmeyi sağlayan bir kütüphane vardır. Şekil 5.9, özdeğerlerin belirlenmesinin sonuçlarının bir grafiğidir.



Şekil 5.11. Laplasyen Matrisinin Özdeğerleri

Şekil 5.11'e göre, başlangıç indeksinde indeks 3 civarında özdeğerde bir artış vardır. Bu, eigengap'e dayalı optimum küme sayısının 3 aralığında olduğunu göstermektedir. Bu nedenle, bu çalışmada Spektral algoritmaya dayalı modeli uygulamak için $n_kümeler=3$ kullanılmıştır. Aşağıdaki şekil 5. Spektral Kümeleme ile oluşturulan PCA grafiği sonuçları.



Şekil 5.12. Spektral Kümeleme Sonuçları (PCA İzdüşümü ile)

Şekil 5.12'deki grafik, Spektral kümelemenin verideki iki ana yapıyı başarılı bir şekilde tanıdığını ve 3 kümeye ayırdığını göstermektedir. Görsel olarak tamamen ayrı 3 grup varmış gibi görünmese de algoritma yerel farklılıkları tanımlamakta ve verileri daha ayrıntılı bir şekilde parçalara ayırmaktadır. Algoritmayı daha iyi tanımlamak için Çizelge 5.3'te spektral algoritmanın metrik değerlendirme sonuçları açıklanmaktadır.

Çizelge 5.7. Spektral algoritması üzerindeki metrik değerlendirme sonuçları

Siluet Skoru	-0.04485
Davies-Bouldin Endeksi	6.52696
Dunn Endeksi	0.25385
Geçen Süre	43.5792 seconds
Kullanılan Bellek	1551.69 MiB
Toplam Küme	3

Spektral Kümeleme değerlendirme sonuçlarına göre, eigengap sezgiselinin sonuçlarına uygun olarak küme sayısı 3'tür. Çizelgedeki Silhouette Score değeri, veri

noktalarının -1 ila 1 aralığında olduğunu göstermektedir ki bu da kümeleme sonuçlarının iyi olmadığının ve küme yapısının verinin doğal örüntüsünü yansıtmayabileceğinin güçlü bir göstergesidir. Bu durum, sonucu 6,52 olan ve yüksek olarak tanımlanan DBI ile doğru orantılıdır ve kümelerin üst üste bindiğini ve kümeler arasındaki ayrımın optimal olmadığını göstermektedir. Ayrıca, Dunn Endeksi değeri 0.253'tür ve bu da kümeler arasında hafif bir ayrıma işaret etmektedir ancak Silhouette ve DBI'daki önceki parametrelerin zayıf sonuçlarını telafi edecek kadar güçlü değildir. Spektral algoritma için 43.5792 saniyelik süre ve 1.5 GB'lık bellek kullanımı, büyük ve karmaşık veriler için hala nispeten standarttır.



6. SONUÇLAR VE ÖNERİLER

Bu bölüm, büyük veri bağlamında çeşitli kümeleme algoritmalarının performansının değerlendirilmesi ve karşılaştırılmasıyla ilgili araştırma sonuçlarından bir seçki sunmaktadır. Bu araştırma, çeşitli değerlendirme ölçütleri kullanan bir meta-analiz yaklaşımı aracılığıyla altı popüler kümeleme algoritmasının güçlü ve zayıf yönlerini derinlemesine anlamaya odaklanmaktadır.

Büyük veri günümüzde çok popüler bir konudur ve Sanayi Devrimi 4.0'ın bir sonucu olarak çeşitli kaynaklardan gelen büyük miktarlardaki verilerin yönetimini, depolanmasını ve analizini kolaylaştıran yeni bir atılım olarak kategorize edilmektedir (Supriyanto ve ark., 2021). Bu başlık altında tez çalışmasından elde edilen sonuçlar detaylı olarak açıklanacak ve sonuçlara dayalı öneriler okuyucu ile paylaşılacaktır.

6.1. Sonuç

Kümeleme, tabanın paylaştığı belirli özelliklerin benzerliğine dayalı olarak verileri gruplar veya kümeler halinde gruplama işlemidir (Ha ve ark., 2011). Bu çalışmada, bu yöntem yüksek boyutlu ve büyük ölçekli verilerde gizli yapıları anlamak amacıyla kullanılmıştır. Kümeleme yöntemi, sınıflandırma gibi sınıf etiketlerine sahip bir yöntem değildir. Ancak, kümelemenin temel amacı, benzer özelliklere sahip verileri kendi içinde homojen, diğer gruplardan ise heterojen olacak şekilde gruplara ayırmaktır (A. K. Jain, 2010).

Kümelemeye duyulan ihtiyaç aynı zamanda çok hızlı gelişen teknoloji ile de doğru orantılıdır, böylece içinden bilgi çıkarma ihtiyacı duyulan dijital veri hacminde artış yaşanmaktadır. Bu büyük hacimli verilerden biri de kümeleme ile işlenebilir. Kümelemenin kendisi pazarlama, biyomedikal sosyal analiz, anomali tespiti ve tavsiye sistemleri gibi çeşitli alanlarda yaygın olarak kullanılmaktadır (3). Bunun nedeni, bu yöntemin etiketlenmemiş verileri işlemede çok esnek olması ve karmaşık örüntüleri tanımlamadaki başarısıdır.

Kümeleme algoritmalarının farklı çalışma şekilleri ve varsayımları vardır. Örneğin, K-Means verileri merkeze olan uzaklığa göre kümeler ve kümelerin merkezi şeklinde ve eşit dağılmış olduğunu varsayar (Ha ve ark., 2011). DBSCAN ise veri yoğunluğunu kullanır ve gürültü ve düzensiz küme şekillerini tespit etmek için uygundur. Aglomeratif Kümeleme aşağıdan yukarıya doğru bir küme hiyerarşisi oluştururken, Spektral Kümeleme bitişiklik matrisinin spektrumuna dayanır (von Luxburg, 2007).

Mean Shift, veri dağılımındaki yoğunluk modlarını izler ve DEC (Deep Embedded Clustering), otokodlayıcı (*autoencoder*) temsil gücünü K-Means'in sınıflandırma yeteneği ile birleştirir (Xie ve ark., 2015).

Bu tez, önceden tanımlanmış özelliklerin benzerliklerine dayalı olarak gruplandırmak için uluslararası konferans katılımcı verilerini kullanarak eğitim verilerini kümelemeye odaklanmaktadır. Veriler özel olduğu için daha önce hiçbir araştırma bu veri setini kullanmamıştır.

Bu tez araştırmasının ilk aşaması olarak, uluslararası konferanslar düzenleyen kurumlardan veri toplanmıştır. Elde edilen veriler ham verilerdir. Çalışma, ilgili bölümlerde tanımlanan kriterlere göre verileri sıralamış, temizlemiş ve puanlamıştır. Sonuç olarak, veri setinden 6 özellik seçilmiştir. Ve yaklaşık 22 bin veri, veri seti olarak kullanıldı.

Bu tezin ikinci aşamasında kümeleme modelleri ele alınmıştır. Ayrıca, hazırlanan verilere uygulanan her algoritmanın özellikleri de tartışılmaktadır. Araştırma için aşağıdaki algoritmalar seçilmiştir K-Means, DBSCAN, Mean Shift, Aglomeratif Kümeleme, Spektral Kümeleme ve Deep Embedded Kümeleme.

Ayrıca bu tezin üçüncü aşamasında elde edilen bulguların sonuçları paylaşılmaktadır. Bu tezin ortamında hazırlanan ve daha önce hiçbir yerde kullanılmamış olan veri kümesi ile elde edilen sonuçlara dayanmaktadır. 6 farklı kümeleme algoritması değerlendirildikten sonra küme kalitesi, hesaplama verimliliği ve veri yapısının haritalanmasındaki algoritmalar açısından çeşitli sonuçlar elde edilmiştir. Çizelge 6.1, kümeleme algoritmalarının ölçüm sonuçlarının özetini göstermektedir.

Çizelge 6.1. Kümeleme Algoritmalarına Ait Ölçüm Sonuçlarının Özeti

Küme Adı	Spektral	Aglomeratif	Mean Shift	DEC	DBSCAN	K-Means
Siluet (Silhouette) Skoru	-0.04485	0.2645	0.0551	0.05511	0.11495	0.04319
Davies-Bouldin İndeksi	6.52696	1.5168	2.2912	2.29122	9.969	3.38706
Dunn İndeksi	0.25385	0.533	0.02406	0.01429	0.22515	0.2617
Atalet (Inertia)				6694070	n/a	689401.894
Geçen Süre (Saniye)	43.5792	23.6844	43.5792	43.5792	0.62892	0.4757
Kullanılan Bellek (MiB)	1551.69	5401.72	1551.69	1551.69	1684.203125	1669.07
Toplam Küme Sayısı	3	17	4	4	2	5

Çizelge 6.1, tüm metriklerde tutarlı bir şekilde üstün olan tek bir algoritma olmadığını açıklamaktadır.

- a) Çizelgedeki çalışmanın sonuçlarına göre K-Means en hızlısıdır ve fazla bellek kullanmaz. Ancak, Silhoutte çekirdeği 0,04319 ile çok düşüktür ve DBI 3,38706 ile oldukça yüksektir ve zayıf kümeleme kalitesini gösterir. Dunn İndeksi (0.2617) oldukça iyidir ve atalet (689.401) hala makul sınırlar içerisinde. Bu çalışmada, K-Means'in karmaşık veya doğrusal olmayan küme şekillerini yakalamada daha az yetenekli olduğu sonucuna varılabilir. Bu tür bir yapıya sahip veri kümeleri için K-Means yalnızca PCA veya ön kümeleme gibi diğer yaklaşımlardan sonra kullanıldığında uygundur. K-Means, en hızlı zaman ve en düşük bellek tüketimi ile hesaplama açısından verimlidir, ancak küme ayırma kalitesinde optimal değildir.
- b) DBSCAN orta düzeyde bir Siluet Puanı değeri (0,11495) ve orta düzeyde bir Dunn Endeksi (0,22515) DBI çok yüksektir (9,969). Bu algoritma oldukça hızlı olmasına rağmen, çünkü tüm algoritmalarla karşılaştırıldığında kullanılan süre en hızlısıdır, yine de kümeye dahil edilmeyen 258 veri vardır ve sadece 2 küme üretir, bu da bu algoritmanın optimizasyon eksikliğini gösterir.
- c) Çizelge 6.2'deki Ortalama Kaydırma algoritması sonuçları, bu algoritmanın düşük Silhoutte Skoru değeri (0,0551) ve küçük Dunn Endeksi (0,02406) ile yoğun ve iyi ayrılmış kümeler üretmede başarısız olduğunu tanımlamaktadır. DBI (2,2912) hala kabul edilebilir ve küme sayısı ideal olmasına rağmen, Ortalama Kaydırma algoritması yüksek boyutlu veriler için hala daha az etkilidir ve bant genişliği parametresinin genişletilmesi gerekir.
- d) DEC algoritması, en düşük Dunn Endeksi (0.01429) ve çok yüksek atalet (6,694,070) ile çok zayıf iç kalite gösterir ve bu da kümelerin geniş bir şekilde yayıldığını gösterir. Küme sayısı (4) ve DBI (2.29122) hala makul olsa da gizli temsil kalitesi optimal değildir. Bu durum, derin eğimli modelin kümeleme için bir özellik temsili bulmakta başarılı olamadığını ve bunun oto kodlayıcı mimarisinin ayarlanması ve daha derin ön eğitimin dikkate alınmasıyla iyileştirilebileceğini göstermektedir.
- e) Bu çalışmada Aglomeratif algoritma, en yüksek Siluet Puanı (0,2645) ve Dunn Endeksi (0,533) ve en düşük DBI (1,5168) ile küme kalitesi açısından en iyi sonuçları vermektedir. Bu algoritma metrik değerlendirme için idealdir, ancak

küme sayısının dendrogram kesme veya diğer dahili doğrulama gibi yöntemlerle çok büyük olmaması için optimize edilmesi gerekir. Spektral algoritma, Spektral Kümelemenin bu veri seti için uygun olmadığını göstermektedir, çünkü veri yapısı Spektral tarafından kullanılan grafik tabanlı yöntemi desteklememektedir. Bu durum, kümelerin örtüştüğünün ve iyi ayrılmadığının güçlü göstergeleri olan negatif Siluet Skoru (-0,04485) ve yüksek DBI (6,52696) ile kanıtlanmaktadır.

Yukarıdaki açıklamadan yola çıkarak, aşağıdaki Çizelge 6.2, Çizelge 6.1'deki değerlendirme sonuçlarına dayalı olarak her bir algoritmanın performansının bir özetini sunmaktadır.

Çizelge 6.2. Kümeleme Algoritmalarının Karşılaştırmalı Değerlendirmesi

Küme adı	Spektral	Aglomeratif	Mean Shift	DEC	DBSCAN	K-Means
Küme kalitesi sonuçları	Kötü	En iyi (tüm yüksek metrikler)	Ortalama	İçsel olarak zayıf	Kararsız	Zayıf kümelenme
Zaman Verimliliği	Uzun süre	Orta	Orta	Uzun süre	Hızlı	Çok hızlı
Efisiensi Memori	Orta	Yüksek	Orta	Orta	Yüksek	Yüksek

Çizelge 6.2, Spektral, Aglomeratif, Ortalama Kaydırma, DEC, DBSCAN ve K-Means olmak üzere altı kümeleme algoritmasından Aglomeratif Kümelemenin en iyi genel sonuçları sağladığını göstermektedir. Bu, 0,2645'lik en yüksek Siluet Skoru değerinden, 1,5168'lik en düşük Davies-Bouldin Endeksinden ve 0,533'lük en yüksek Dunn Endeksinden görülebilir; bu da kümeler arasında iyi bir ayrım ve güçlü bir iç küme yapısı olduğunu gösterir. Bununla birlikte, algoritma 5401,72 MiB'lik bir kullanımla bellek verimliliğinden muzdariptir ve aşırı kümelenmeye yol açabilecek çok fazla küme (17 küme) üretir.

Öte yandan, Mean Shift ve DEC algoritmaları düşük Siluet Skoru (0,0551 ve 0,05511) ve 2,291 Davies-Bouldin Endeksi ile oldukça benzer metrik performans göstermektedir, ancak Dunn Endeksi değeri çok düşüktür ve bu da küme sonuçlarının kompakt olmadığını göstermektedir. K-Means zaman ve bellek açısından verimlidir, ancak eylemsizlik hala makul sınırlar içinde olmasına rağmen 0,04319 Siluet Puanı ve 0,2617 Dunn Endeksi ile kümeleme kalitesi düşüktür.

Bu arada, DBSCAN algoritması çok sayıda gürültü noktası (258) ve yüksek Davies-Bouldin İndeksi (9,969) üreterek oldukça hızlı ve bellek açısından verimli

olmasına rağmen düşük küme kalitesine işaret etmektedir. Spektral Kümeleme, negatif Siluet Skoru (-0,04485) ve yüksek DBI ile en kötü sonuçları kaydetmiştir; bu da kümeler arasındaki ayrımın çok zayıf olduğunu ve sonuçların temsili olmadığını göstermektedir.

Bu değerlendirmeye dayanarak, yüksek kaliteli kümeler üretmek için Aglomeratif Kümeleme önerilebilir, ancak küme sayısının bellek ve yorumlanabilirlik hususları ile dengelenmesi gerekir. Buna karşılık, Spektral ve DBSCAN gibi algoritmalar, düşük kümeleme kalitesi ve yüksek gürültü nedeniyle mevcut veriler ve yapılandırma için daha az uygundur.

Tüm deneyden, büyük veriler için kümeleme algoritmalarının seçiminin yalnızca küme ayrımının doğruluğuna değil, aynı zamanda hesaplama verimliliğine ve veri özellikleriyle uyumluluğa da bağlı olduğu sonucuna varılabilir. Bu nedenle en iyi strateji, en uygun algoritmayı belirlemeden önce dahili metrik analizi, görselleştirme ve veri yapısının anlaşılmasını birleştirmektir. Bu yaklaşım, yalnızca doğru değil, aynı zamanda verimli ve bağlamsal olarak büyük veri analizinin ihtiyaçlarıyla ilgili olan yöntemlerin uygulanmasında daha bilinçli karar verilmesine olanak tanır.

6.2. Öneriler

Bu araştırma, büyük veri bağlamında kümeleme algoritmalarının meta-analizine odaklanmaktadır. Bu çalışma, yalnızca algoritmaların bireysel performanslarını karşılaştırmakla kalmayıp, aynı zamanda veri çeşitliliğini karşılayabilecek standartlaştırılmış sistematik bir değerlendirme çerçevesi geliştirmeyi de amaçlamaktadır. Araştırmada, Siluet Skoru, Davies-Bouldin Endeksi ve Dunn Endeksi gibi iç değerlendirme metriklerine dayanarak çeşitli kümeleme algoritmalarının performansı analiz edilmiştir.

Gelecek araştırmalarda, özellikle DEC, Spectral Clustering ve Mean Shift gibi parametreye duyarlı algoritmalar için Grid Search veya Bayesian Optimization gibi otomatik parametre ayarlama yaklaşımlarının kullanılması önerilmektedir. Özellikle DEC algoritması için, kümeleme performansının artırılması amacıyla encoder mimarisinin geliştirilmesine odaklanılabilir. Bu geliştirmeler arasında daha uyarlanabilir bir kayıp fonksiyonu seçimi, gizli katmanların artırılması, latent kod boyutunun optimize edilmesi ve KL divergence gibi düzenleme ağırlıklarının dinamik olarak ayarlanması yer alabilir.

Bunun yanı sıra, algoritmaların farklı uygulama alanlarına genellenebilirliğini test etmek amacıyla çeşitli veri kümeleri üzerinde daha fazla deney yapılması gerekmektedir.

Örneğin, öneri sistemleri, müşteri segmentasyonu veya sosyal analiz gibi alanlarda uygulanabilirliği değerlendirilebilir. Pratik uygulamalarda, kümeleme sonuçlarının yorumlanabilirliğine de dikkat edilmesi, karar verme süreçlerini desteklemek ve yapay zeka tabanlı ürünlerin daha etkili geliştirilmesini sağlamak açısından büyük önem taşımaktadır. Tüm bu unsurlar göz önünde bulundurularak, bu çalışmanın büyük ve karmaşık verilerin işlenmesinde en uygun kümeleme stratejisini belirlemede değerli bir referans olacağı öngörülmektedir.



KAYNAKLAR

- A, I., D, N., & Sriani. 2015. Penerapan Data Mining dengan Algoritma Fp-Growth untuk Mendukung Strategi Promosi Pendidikan (Studi Kasus Kampus STMIK Triguna Dharma). *Saintikom*, 14(3), 211–266.
- Abdalla, H. B., 2022, A Brief Survey On Big Data: Technologies, Terminologies And Data-Intensive Asslications, In *Journal of Big Data*, vol. 9, no.1, ss. 1-31.
- Aggarwal, C. C., & Reddy, C. K. 2014. *Data Clustering Algorithms and Applications*. CRC Press
- Ahmad, P. H., ve Dang, S., 2015, Performance Evaluation of Clustering Algorithm Using Different Datasets, *Journal of Information Engineering and Asslications*, vol. 5, no.1, ss. 39–46.
- Ahmar, A. S., Napitupulu, D., Rahim, R., Hidayat, R., Sonatha, Y., ve Azmi, M., 2018, Using K-Means Clustering to Cluster Provinces in Indonesia, *Journal of Physics: Conference Series*, 1028(1), ss. 1-6.
- Aldossari, S., Mokhtar, U. A., ve Abdul Ghani, A. T., 2023, Factor Influencing the Adoption of Big Data Analytics: A Systematic Literature and Experts Review, *SAGE Open*, ss. 1-25.
- Apriyanty, D., Mariza Putra, A., Agil Caesar, R., 2023. Implementasi Mean Shift Clustering Dalam Mengelompokkan Pelanggan Retribusi Alat Pemadam Kebakaran Pada Dinas Pemadam Kebakaran dan Penanggulangan Bencana Kota Palembang. In *Jurnal Sistem Informasi* (Vol. 4, Issue 1).
- Arifah, S., Swedia, E. R., ve Septian, M. R. D., 2023, Analisis Perbandingan Algoritma Clustering Dalam Melakukan Segmentasi Warna Pada Citra Jajan Tradisional. *Sebatik* 27(1), 70–76.
- Arifin, A., 2023, Pengelompokan Titik Kejadian Gempa bumi di Wilayah Asia Tenggara Menggunakan Agglomerative Hierarchical Clustering Clustering Earthquake Event Points in the Southeast Asia Region using Agglomerative Hierarchical Clustering, vol 12, no. 3, ss. 900-9014.
- Artana, P. N. P., Mandyartha, E. P., & S, M. H. P. (2023). Penerapan Data Mining Pada Algoritma Hierarchical Clustering. *Jurnal Mahasiswa Teknik Informatika*, 4, ss. 2903–2909.
- Arzinal, D. (2022, April 8). *BRI TECH - Hustler, Hipster, Hacker: Tiga Persona Inovator Produk BRI - BRI TECH*. <https://digital.bri.co.id/article/hustler-hipster-hacker-tiga-persona-inovator-produk-3sy3> [Ziyaret Tarihi: 5 Mayıs 2025].
- Aslantaş, G., Gençgöl, M., Rumelli, M., Özserağ, M., ve Bakırlı, G., 2023, Customer Segmentation Using K-Means Clustering Algorithm and RFM Model, *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, ss. 491–503.

- Battaglia, O. R., Di Paola, B., ve Fazio, C., 2017, K-means clustering to study how student reasoning lines can be modified by a learning activity based on Feynman's unifying approach. *Eurasia Journal of Mathematics, Science and Technology Education*, vol. 13, no.6, 2005–2038.
- Bhadani, A. K., ve Jothimani, D., 2016, *Big data: Challenges, opportunities and realities*, IGI Global.
- Budiman, S. A. D., Safitri, D., ve Ispriyanti, D., 2016, Perbandingan Metode K-Means dan metode DBSCAN Pada Pengelompokan Rumah Kost Mahasiswa dikelurahan Tembalang Semarang, *Jurnal Gaussian*, vol. 5, no.4, ss.757–762.
- Cahapin, E. L., Malabag, B. A., Santiago, C. S., Reyes, J. L., Legaspi, G. S., ve Adrales, K. L., 2023, Clustering of students admission data using k-means, hierarchical, and DBSCAN algorithms, *Bulletin of Electrical Engineering and Informatics*, vol. 12, no.6, ss. 3647–3656.
- Cherrat, E. M., Alaoui, R., ve Bouzahir, H., 2019, Improving of fingerprint segmentation images based on K-means and DBSCAN clustering. *International Journal of Electrical and Computer Engineering*, vol. 9, no. 4, ss.425–2432.
- Cholissodin, I., ve Riyandani, 2016, *Big Data vs Big Information vs Big Knowledge*, ss.1-35.
- Chugh, V. 2024. Mean Shift Clustering: A Comprehensive Guide | DataCamp. <https://www.datacamp.com/tutorial/mean-shift-clustering>[Ziyaret Tarihi: 5 Aralık 2024].
- C.Kondal raj, 2017, *Comparison Of K Means, K Medoids, Dbscan Algorithms Using Dna Microarray Dataset*.
- Çoğaltay, N., ve Karadağ, E., 2015, Introduction to meta-analysis. In *Leadership and Organizational Outcomes: Meta-Analysis of Empirical Studies*, ss. 19–28.
- Dani, A. T. R., Wahyuningsih, S., ve Rizki, N. A., 2019, Penerapan Hierarchical Clustering Metode Agglomerative pada Data Runtun Waktu. *Jambura Journal of Mathematics*, vol.1, no.2, ss.64–78.
- Djouzi, K., ve Beghdad-Bey, K., 2019, A Review of Clustering Algorithms for Big Data, 2019 International Conference on Networking and Advanced Systems (ICNAS), ss. 1–6.
- Domo. 2024. *Data Never Sleeps 10.0* | Domo. <https://www.domo.com/data-never-sleeps#data> [Ziyaret Tarihi: 1 Aralık 2024].
- Du, H., Chen, S., Niu, H., ve Li, Y., 2021, Application of DBSCAN clustering algorithm in evaluating students' learning status, *Proceedings - 2021 17th International Conference on Computational Intelligence and Security CIS 2021*, ss. 372–376.

- Dumbill, E., 2013, Making sense of Big Data: Vol. I. Mary Ann Liebert.
- Durova, M. A., Khodunov, E. A., Anikeeva, A. S., Mishin, A. A., Zein, A. N., ve Borisova, S. V., 2022, Asslication of the dbscan Method to Solve the Problem of New Employees Adaptation, Proceedings of the 2022 4th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2022.
- Elgendy, N., ve Elragal, A., 2022, Big data analytics: A literature review paper. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), ss. 214–227.
- Event., 2021. Analisis Clustering Dengan K-Means Untuk Pengelompokkan Penjualan Produk Pada Hotel Newton, Putera Batam University ss. 1-31.
- Field, A. P., ve Gillett, R., 2010, How to do a meta-analysis. British Journal of Mathematical and Statistical Psychology, 63(3), ss. 665–694.
- Gao, X., 2022, Analysis of Inuencing Factors on Excellent Teachers' Professional Growth based on DB-K-Means Method.
- Gartner. 2024. *What Is Data and Analytics: Everything You Need to Know* | Gartner. <https://www.gartner.com/en/topics/data-and-analytics> [Ziyaret Tarihi: 10 Aralık 2024].
- Ha, J., Kambe, M., & Pe, J., 2011. Data Mining: Concepts and Techniques. In *Data Mining: Concepts and Techniques*.
- Hahsler, M., Piekenbrock, M., ve Doran, D., 2019, DBSCAN: Fast density-based clustering with R, Journal of Statistical Software, ss. 91.
- Helmi, F. 2024. *Coding • Data Mengenal t-SNE: Teknik Visualisasi untuk Menggali Pola Tersembunyi dalam Data*. <https://Purwadhika.Com/>. <https://purwadhika.com/blog/mengenal-t-sne-teknik-visualisasi-untuk-menggali-pola-tersembunyi-dalam-data> [Ziyaret Tarihi: 15 Aralık 2024].
- Hermawan, G. (2024). *Memahami Peran Dataset dalam Penelitian Kecerdasan Buatan: Kualitas, Aksesibilitas, dan Tantangan*. 1.
- Hidayati, Q. R., & Surono, S. 2021. Implementasi Algoritma Spectral Clustering Untuk Analisis Sentimen. *Delta: Jurnal Ilmiah Pendidikan Matematika*, 9(1), 27.
- Hoy, M. B., 2014, Big Data: An Introduction for Librarians. Medical Reference Services Quarterly, 33(3), 320–326.
- Hu, Y., 2023, Customer Market Analysis Based on Interval Value Data Dynamic Clustering Algorithm. International Conference on Integrated Intelligence and Communication Systems, ICIICS 2023.

- Hurwitz, J., Nugent, A., Halper, F ve Kaufman, M., 2013, *Big Data For Dummies*, D. Miller, Ed John Wiley ve Sons, Inc.
- IBM. 2023. *Apa Itu Autoencoder?* | IBM. <https://www.ibm.com/id-id/think/topics/autoencoder> [Ziyaret Tarihi: 15 Mayıs 2025].
- Indini, D. P., Siburian, S. R., ve Utomo, P., 2022, Implementasi Algoritma DBSCAN untuk Clustering Seleksi Penentuan Mahasiswa yang Berhak Menerima Beasiswa Yayasan. *Seminar Nasional Sosial Humaniora Dan Teknologi*, ss. 325–331.
- Jain, A. K., 2010, Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8), ss. 651–666.
- Jain, S., 2020, Meta-Analysis: A Higher Quality of Evidence in Clinical Research Pyramid, Article in *International Journal of Science and Research*.
- Jang, J., ve Jiang, H., 2018, DBSCAN++ Towards fast and scalable density clustering. *Accessibility Forum*.
- Jordan, M, 2004, Learning Spectral Clustering, *In Proceedings of 17th Advances in Neural Information Processing Systems*, 16.
- Joshi, K. D., & Nalwade, P. S. 2013. Modified K-Means for Better Initial Cluster Centres. *International Journal of Computer Science and Mobile Computing*, 2(7), 219–3.
- Lee, S., 2025, *7 Essential Metrics for Dunn Index in Clustering Analysis*. Number Analytics. <https://www.numberanalytics.com/blog/dunn-index-clustering-metrics> [Ziyaret Tarihi: 10 Şubat 2025].
- Lee, Y., Park, C., & Kang, S. ,2023, Deep Embedded Clustering Framework for Mixed Data. *IEEE Access*, 11, ss. 33–40.
- Lewis, A. L., & Kearney, P. (2020). *UN's 17 Sustainable Development Goals (SGDs) | Kearney Group*. <https://kearneygroup.com.au/article/what-are-the-sgds-17-sustainable-development-goals/> [Ziyaret Tarihi: 01 Şubat 2025].
- Lubis, A., 2016., *Basis Data Dasar*. Deepublish.
- Martin, K. (2019). *Implementasi Algoritma Mean Shift Clustering pada Color Quantization*. ss 1-10
- Mohamed Nafuri, A. F., Sani, N. S., Zainudin, N. F. A., Rahman, A. H. A., ve Aliff, M., 2022, Clustering Analysis for Classifying Student Academic Performance in Higher Education, *Asslied Sciences (Switzerland)*, 12(19).
- Murtagh, F., ve Contreras, P., 2012, Algorithms for hierarchical clustering: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no.1, 86–97.

- Nanthiya, G., Punitha S. C. ve Punithavalli, M., 2010, An Analytical Study on Behavior of Clusters Using K-Means, EM and K-Means Algorithm. (IJCSIS) International Journal of Computer Science and Information Security, vol. 7, no.3, ss. 185–190.
- Nasir, J., 2020, Penerapan Data Mining Clustering Dalam Mengelompokkan Buku Dengan Metode K-Means. Jurnal SIMETRIS, vol 11, no.2.
- Nisha ve Kaur, P. J., 2015, Cluster Quality Based Performance Evaluation of Hierarchical Clustering Method. T International Conference on Next Generation Computing Technologies, ss. 649–653.
- Oliveira, M. I., & Marcal, A. R. S. 2023. Clustering LiDAR Data with K-means and DBSCAN. *International Conference on Pattern Recognition Applications and Methods, 1*, 822–831.
- Oyelade, O. J., Oladipupo, O. O., ve Obagbuwa, I. C., 2010, Asslication of k-Means Clustering algorithm for prediction of Students' Academic Performance. IJCSIS) International Journal of Computer Science and Information Security, vol. 71, ss. 292–295.
- Pasin, Ö., ve Ankarali, H., 2015, Usage of Kernel K-Means and DBSCAN cluster algorithms in health studies: An asslication, Clinical Research and Trials, vol.1, no.3, ss. 63–69.
- Philip Chen, C. L., & Zhang, C.-Y. 2014. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, 275, ss 314–347.
- Pradana, Q. C., 2020, *Deep Embedded Clustering untuk Pendeteksian Topik Tweet Berita Berbahasa Indonesia = Deep Embedded Clustering for Topic Detection on Indonesian News Tweet*. Universitas Indonesia.
- Priyatman, H., Sajid, F., ve Haldivany, D., 2019, Klasterisasi Menggunakan Algoritma K-Means Clustering untuk Memprediksi Waktu Kelulusan Mahasiswa. Jurnal Edukasi Dan Penelitian Informatika, vol. 5, no.1, ss.62–66.
- Purnomo, R., Wowon Priatna ve Tri Dharma Putra, 2021, Implementasi Big Data Analytical Untuk Perguruan Tinggi Menggunakan Machine Learning. Journal of Informatic and Information Security, vol. 2, no.1, ss. 77–88.
- Ran, X., Xi, Y., Lu, Y., Wang, X., ve Lu, Z., 2023, Comprehensive survey on hierarchical clustering algorithms and the recent developments. Artificial Intelligence Review, vol. 56, no.8, ss. 8219–8264.
- Riahi, Y., ve Riahi, S., 2018, Big Data and Big Data Analytics: concepts, types and technologies. International Journal of Research and Engineering, vol. 5, no.9, ss. 524–528.
- Salahudin, S. 2019. *Mengenal Big Data (BD) dan Big Data Analysis (BDA)*.

- Santosa, B., 2007, Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis, In Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis. Graha Ilmu.
- Santoso, J. T., 2020, Analisa Big Data, Yayasan Prima Agus Teknik.
- Shabbir, M. Q., ve Gardezi, S. B. W., 2020, Asslication of big data analytics and organizational performance: the mediating role of knowledge management practices. *Journal of Big Data*, vol. 7, no. 1, ss. 1-17.
- Shah, G. H., 2012, An improved DBSCAN, a density based clustering algorithm with parameter selection for high dimensional data sets. 3rd Nirma University International Conference on Engineering (NUICONE) 2012, ss.1-6.
- Shelby, L. B., ve Vaske, J., 2008, Understanding meta-analysis: A review of the methodological literature. In *Leisure Sciences*, Vol. 30, Issue 2, ss. 96–110.
- Shirkhorshidi, A. S., Aghabozorgi, S., Wah, T. Y., & Herawan, T. 2014. Big data clustering: A review. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8583 LNCS(PART 5), ssç 707–720.
- Simbolon, I. N., & Friskila, P. D., 2024, Analisis Dan Evaluasi Algoritma Dbscan (Density-Based Spatial Clustering Of Applications With Noise) Pada Tuberkulosis. *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(3S1).
- Sirait, E. R. E. 2016. Implementasi Teknologi Big Data Di Lembaga Pemerintahan Indonesia. *Jurnal Penelitian Pos Dan Informatika*, 6(2), ssç 113.
- Sjoberg, E. A., 2014, Conducting a meta-analysis for your student dissertation. *PsyPag Quarterly*, vol. 1, ss.18–23.
- Sopyan, Y., Lesmana, A. D., & Juliane, C., 2022, Analisis Algoritma K-Means dan Davies Bouldin Index dalam Mencari Cluster Terbaik Kasus Perceraian di Kabupaten Kuningan. *Building of Informatics, Technology and Science (BITS)*, 4(3).
- Strohmeier, S., ve Duveneck, A.-J., 2015, Introduction to Procedures and Methods of Meta-Analysis. *Seminararbeit Wintersemester 2014/2015*, ss. 1–25.
- Suganya, R., Pavithra, M., & Nandhini, P. (2018). Algorithms and Challenges in Big Data Clustering. *International Journal of Engineering and Techniques*, 4, ss.40–47.
- Sun, S., Lei, K., Xu, Z., Jing, W., ve Sun, G., 2023, Analysis of K-means and K-DBSCAN Commonly Used in Data Mining. *Proceedings - 2023 International Conference on Intelligent Media, Big Data and Knowledge Mining (IMBDKM) 2023*, ss. 37–41.
- Supriyanto, E. E., Susilo Bakti, I., Furqon, M., & Artikel, R., 2021. *The Role Of Big Data In The Implementation Of Distance Learning Info Artikel Abstrak*. 12(1), ss. 61–68.

- Susilowati, Q. I., & Başçiftçi, F. (2025). Startup Success Factors: Classifying 3H (Hustler, Hipster, Hacker) Framework using Simple Additive Weighting. *Journal Homepage: Journal of Electrical Engineering and Computer (JEECOM)*, 7(1). ss 184-194
- Syakur, M. A., Khotimah, B. K., Rochman, E. M. S., & Satoto, B. D. (2018). Integration K-Means Clustering Method and Elbow Method for Identification of the Best Customer Profile Cluster. *IOP Conference Series: Materials Science and Engineering*, 336(1).
- Tosi, D., Kokaj, R., ve Roccetti, M., 2024, 15 years of Big Data: a systematic literature review. *Journal of Big Data*, vol. 11, no.1.
- Trivusi. (2022). *Pengertian dan Jenis-jenis Distance Metric pada Machine Learning* -. <https://www.trivusi.web.id/2022/06/jenis-distance-metric.html> [Ziyaret Tarihi: 5 Şubat 2025].
- Tugay Bilgin, T., ve Çamurcu, Y., 2005, DBSCAN, OPTICS ve K-Means Kümeleme Algoritmalarının Uygulamalı Karşılaştırılması. *Politeknik Dergisi Journal of Polytechnic Cilt*, vol. 8, no. 2, ss. 139–145.
- Tysara, L., 2022, Gürültü (elektronik), Big Data adalah Penyedia Ruang Penyimpanan Tak Terbatas, Asal Usul, dan Contoh Penerapannya. *Liputan 6*. <https://www.liputan6.com/hot/read/4848247/big-data-adalah-penyedia-ruang-penyimpanan-tak-terbatas-asal-usul-dan-contoh-penerapannya> [Ziyaret Tarihi: 5 Haziran 2024].
- Tzortzis, G., & Likas, A. 2008. The Global Kernel k-Means Clustering Algorithm. *IEEE*, ss.1978–1985.
- United Nation. (2022). *THE 17 GOALS | Sustainable Development*. <https://sdgs.un.org/goals> [Ziyaret Tarihi: 25 Şubat 2025].
- Utari, R. F., Insani, F., Agustian, S., & Afriyanti, L., 2024, Pengelompokan Data Pendistribusian Listrik Menggunakan Algoritma Mean Shift. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(3), ss.1015–1023.
- Wikipedi, 2023, Gürültü (elektronik), Meta-analiz. <https://tr.wikipedia.org/wiki/Meta-analiz> [Ziyaret Tarihi: 17 Temmuz 2024].
- Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4), ss. 395–416.
- Wierzchoń, S. T., & Kłopotek, M. A. (2018). *Studies in Big Data 34 Modern Algorithms of Cluster Analysis*, Vol. 34.
- Wikipedia, 2023, *Eigengap*. Wikipedia. <https://en.wikipedia.org/wiki/Eigengap>. [Ziyaret Tarihi: 17 Mart 2025].

- Wikipedia.2025. *Principal component analysis*.
https://en.wikipedia.org/wiki/Principal_component_analysis [Ziyaret Tarihi: 10 Mart 2025].
- Wise, J., 2023, Gürültü (elektronik), How Much Data Is Generated Every Day In 2024? (NewStats),https://earthweb.com/how-much-data-is-created-everyday/#Key_Data_Creation_Statistics_2024(elektronik), [Ziyaret Tarihi: 08 Haziran 2024].
- Xie, J., Girshick, R., & Farhadi, A., 2015, *Unsupervised Deep Embedding for Clustering Analysis*. <http://arxiv.org/abs/1511.06335>
- Zhang, L. 2023 Research on K-means Clustering Algorithm Based on MapReduce Distributed Programming Framework. *Procedia Computer Science*, 228, 262–270.
- Zhang, M., ve Liao, H., 2020, Privacy-Preserving DBSCAN clustering algorithm based on negative database, Institute of Electrical and Electronics Engineering, Inc, vol. 5.