Makine Öğrenmesi İle Kümeleme Yöntemleri Araştırması

Umut Uzunhasan

2025

İçindekiler

Giriş

Makine öğrenmesi, veriden anlamlı bilgiler çıkararak belirli görevleri otomatikleştirmek ve karmaşık problemlere çözümler üretmek için kullanılan bir yapay zeka dalıdır. Bu alan, genellikle denetimli ve denetimsiz öğrenme olmak üzere iki ana kategoriye ayrılır. Denetimli öğrenme, etiketli verilerle model eğitimi yapmayı içerirken, denetimsiz öğrenme etiketlerin bulunmadığı durumlarda verinin doğal yapısını anlamaya odaklanır. Denetimsiz öğrenmenin temel tekniklerinden biri olan kümeleme, veri setlerini benzer özelliklere sahip alt gruplara ayırmayı amaçlar.

Kümeleme yöntemleri, verinin iç yapısını keşfetmek ve farklı gruplar arasında benzerlikleri anlamak için güçlü araçlar sunar. Bu yöntemlerden biri olan K-Means, veri noktalarını belirli bir sayıda küme merkezine yakın olacak şekilde gruplar. Partitioning Around Medoids (PAM), K-Means'e benzer bir mantıkla çalışsa da, küme merkezleri yerine mevcut veri noktalarını temsilci olarak seçer ve gürültüye karşı daha dayanıklıdır. Hiyerarşik Kümeleme ise, veriyi bir ağaç yapısında temsil eder ve alt kümeler arasındaki ilişkileri keşfetmek için hem birleştirici hem de bölücü yaklaşımlar sunar.

Bu çalışmada, K-Means, PAM ve Hiyerarşik Kümeleme yöntemlerinin teorik temelleri ele alınacak ve bu yöntemlerin bir veri seti üzerindeki uygulamaları incelenecektir. Amaç, her bir yöntemle elde edilen sonuçların karşılaştırılması ve farklı veri yapıları üzerindeki performanslarının analiz edilmesidir. Bu bağlamda, kümeleme yöntemlerinin veri keşfi, segmentasyon ve görselleştirme süreçlerindeki rolü detaylandırılacaktır.

Bölüm 1

Kümeleme Nedir?

Kümeleme, veriyi benzer özellikler taşıyan gruplar halinde organize etmek için kullanılan bir makine öğrenmesi yöntemidir. Bu teknik, denetimsiz öğrenme kategorisinde yer alır ve veri noktaları arasında belirgin bir etiket bulunmadığı durumlarda verinin doğal yapısını anlamayı amaçlar. Kümeleme, pazarlama, biyoinformatik, görüntü işleme ve sosyal ağ analizi gibi pek çok alanda yaygın olarak kullanılır. Örneğin, müşteri segmentasyonu için farklı alışveriş alışkanlıklarına sahip kullanıcıları gruplandırmak, kümeleme uygulamalarına tipik bir örnektir.

Kümeleme, veriyi anlamlandırarak daha iyi analiz yapmamızı sağlar. Bu yöntem, verilerin yapısal özelliklerini ortaya koyar, anomalileri tespit etmeye yardımcı olur ve yeni bilgiler keşfetmek için bir temel sunar. İşlevselliği sayesinde veri keşfi, modelleme ve görselleştirme gibi süreçlerde önemli bir rol oynar. Kümeleme analizi, özellikle büyük ve karmaşık veri setlerinde içsel ilişkileri anlamak için etkili bir araçtır.

1.1 Kümeleme Yöntemleri

Kümeleme yöntemleri arasında yaygın olarak kullanılan birkaç teknik bulunmaktadır.

- **K-Means Kümeleme:** Veri noktalarını, belirlenen bir küme sayısına göre merkezlere yakınlık esasına dayanarak gruplandırır. Hızlı ve etkili bir yöntemdir, ancak başlangıç merkezlerine duyarlıdır.
- **Partitioning Around Medoids (PAM):** Her bir kümenin merkezini veri noktalarından seçerek, K-Means'e kıyasla daha gürültüye dayanıklıdır. Özellikle küçük veri setlerinde etkili sonuçlar sunar.
- **Hiyerarşik Kümeleme:** Veriyi bir ağaç yapısı üzerinde hiyerarşik olarak gruplandırır. Alt kümeler arasındaki ilişkileri keşfetmek için birleştirici (agglomeratif) ve bölücü (divisive) yaklaşımlar kullanır. Bu yöntem, küme sayısının önceden belirlenmesini gerektirmez ve sonuçları dendrogram ile görselleştirilir.

Bu yöntemler, veri türüne, büyüklüğüne ve çözülmek istenen probleme bağlı olarak seçilebilir ve uygulanabilir. Kümeleme analizi, farklı veri yapılarının anlaşılmasına katkıda bulunarak daha bilinçli kararlar alınmasını sağlar.

Bölüm 2

Kümeleme Mesafe Ölçüleri

Kümeleme analizi, verileri benzerliklerine göre gruplara ayırma işlemidir. Bu işlemde, veriler arasındaki uzaklıkları hesaplamak için çeşitli mesafe ölçüleri kullanılır. Bu bölümde, kümeleme analizinde sıklıkla kullanılan elbow, manhattan, kosinüs, Pearson korelasyonu, Spearman korelasyonu ve Kendall korelasyonu mesafe ölçülerini detaylı olarak inceleyeceğiz.

2.1 Öklid Mesafesi (Euclidean Distance))

Öklid mesafesi, en basit ve en yaygın kullanılan mesafe ölçülerinden biridir. Düz bir çizgi üzerindeki iki nokta arasındaki en kısa mesafeyi temsil eder. Günlük hayatta, bir haritadaki iki şehir arasındaki kuş uçumu mesafesi gibi düşünebiliriz.

Formülü:

$$d(x.y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

2.1.1 Avantajları ve Dezavantajları

Avantajları:

- Hesaplaması basittir.
- Anlaması intuitiftir ve geometriye dayalı bir kavram sunar.

Dezavantajları:

- Tüm boyutlara eşit önem verir. Bu, özelliklerin farklı ölçeklerde olduğu durumlarda sorun yaratabilir.
- Özelliklerin ölçeklendirilmemesi durumunda çarpık sonuçlara yol açabilir.

2.1.2 Kullanım Alanları:

- Sayısal verilerin analizinde yaygın olarak kullanılır.
- Geometrik problemler ve uzayda konum belirleme gibi durumlarda tercih edilir.
- Makine öğrenmesinde, kümeleme algoritmaları (örneğin, k-means) ve yakınlık hesaplamalarında kullanılır.

2.2 Manhattan Mesafesi (Manhattan Distance)

Manhattan mesafesi, iki nokta arasındaki dikey ve yatay hareketlerin toplamına dayanır. Genellikle şehir blokları üzerindeki hareketi modellemek için kullanılır. Bu metrikte, köşegen hareketleri dikkate alınmaz.

Formülü:

$$d(x.y) = \sum_{i=1}^{n} |x_i - y_i|$$

2.2.1 Avantajları ve Dezavantajları

Avantajları:

- Hesaplaması basittir.
- Öklid mesafesine göre daha dayanıklı (robust) olabilir; özellikle çok boyutlu verilerde.

Dezavantajları:

- Köşegen hareketleri dikkate almaz, bu da bazı durumlarda gerçek mesafeyi tam anlamıyla temsil etmeyebilir.

2.2.2 Kullanım Alanları:

- Taksilerin rotaları gibi dikey ve yatay hareketlerle sınırlı sistemlerde kullanılır.
- Görüntü işlemede, pikseller arasındaki mesafeyi hesaplamak için kullanılabilecek bir alternatiftir.

2.3 Kosinüs Benzerliği (Cosine Similarity)

Kosinüs benzerliği, iki vektör arasındaki açının kosinüsünü ifade eder. Bu metrik, vektörlerin büyüklüklerinden bağımsız olarak sadece yönleriyle ilgilenir.

Formülü:

$$\text{similarity} = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\|\|\vec{B}\|}$$

2.3.1 Avantajları ve Dezavantajları

Avantajları:

- Vektörlerin uzunluklarından etkilenmez, sadece yön bilgisiyle ilgilenir.
- Yüksek boyutlu verilerle çalışırken iyi performans gösterir.

Dezavantajları:

- Uzunluk bilgisi dikkate alınmaz
- Sadece açı üzerinden benzerlik ölçer, büyüklük farklarını yansıtmaz.

2.3.2 Kullanım Alanları:

- Metin madenciliği ve bilgi getirme (information retrieval) sistemlerinde belgeler arasındaki benzerlikleri ölçmek için yaygın olarak kullanılır.
- Yüksek boyutlu vektör uzayında temsil edilen verilerde tercih edilir.

2.4 Pearson Korelasyon Katsayısı (Pearson Correlation Coefficient)

Pearson Korelasyon Katsayısı, iki değişken arasındaki doğrusal ilişkinin gücünü ve yönünü ölçen bir istatistiksel metriktir. Bu katsayı, iki değişkenin birlikte nasıl değiştiğini ve aralarındaki ilişkinin pozitif mi yoksa negatif mi olduğunu ifade eder.

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

2.4.1 Avantajları ve Dezavantajları

Avantajları

- İki değişken arasındaki doğrusal ilişkinin gücünü ve yönünü açık bir şekilde ifade eder.
- Hesaplaması basit ve yaygın olarak kullanılır.
- Verilerdeki bağımsız değişken ve bağımlı değişken arasındaki ilişkiyi ölçmek için uygundur.

Dezavantajları

- Sadece doğrusal ilişkileri ölçer; doğrusal olmayan ilişkileri dikkate almaz.
- Aykırı değerler (*outliers*), katsayının değerini ciddi şekilde etkileyebilir.
- Değişkenlerin normal dağılıma yakın olması varsayımına dayanır.

2.4.2 Kullanım Alanları

- **Psikoloji ve Sosyoloji:** İnsan davranışları arasındaki ilişkileri analiz etmek.
- **Finans:** Hisse senedi fiyatları arasındaki korelasyonu ölçmek.
- **Biyoloji:** İki biyolojik değişkenin ilişkisini incelemek (örneğin, yaş ve sağlık durumu).
- **Makine Öğrenimi:** Özelliklerin (*features*) birbirleriyle olan ilişkilerini değerlendirmek.

2.5 Hangi Yöntem Uygulanmalı?

Yöntem seçimi, problemin doğasına, verinin yapısına ve analizin hedeflerine bağlıdır. Farklı mesafe ölçümleri ve benzerlik yöntemleri, çeşitli durumlarda farklı avantajlar sağlayabilir. İşte yöntem seçimi için dikkat edilmesi gereken bazı kriterler:

2.5.1 Veri Türü ve Doğası

- **Eğer veriler geometrik bir uzayda ise:**
  - **Öklid Mesafesi (Euclidean Distance):** Sürekli ve geometrik anlam taşıyan veriler için tercih edilir. Örneğin, iki nokta arasındaki fiziksel mesafeyi ölçmek için uygundur.
  - **Manhattan Mesafesi (Manhattan Distance):** Yol uzunluğu gibi doğrusal ve kısıtlanmış yolların olduğu durumlarda daha uygundur.
- **Eğer veriler yön ve benzerlik ile ilgiliyse:**
  - **Kosinüs Benzerliği (Cosine Similarity):** Metin madenciliği, bilgi getirme ve yüksek boyutlu vektörlerle çalışırken tercih edilir. Özellikle büyüklük yerine yön önemli olduğunda uygundur.
  - **Pearson Korelasyon Katsayısı:** İki değişken arasındaki doğrusal ilişkiyi ölçmek için kullanılabilir. Özellikle sürekli veriler ve istatistiksel analizlerde yaygındır.

2.5.2 Kullanım Alanına Göre Seçim

- **Metin Madenciliği:** Kosinüs Benzerliği sıklıkla tercih edilir, çünkü metin vektörlerinin yönleri arasındaki ilişki önemlidir.
- **Coğrafi Problemler:** Öklid veya Manhattan Mesafesi uygun bir seçimdir.
- **İstatistiksel Analizler:** Pearson Korelasyon Katsayısı, doğrusal ilişkiler için güçlü bir araçtır.

Bölüm 3

Veri Seti Tanıtımı

3.1 Başlangıç Veri Seti

	CustomerID	Gender	Age	Annual.Income..k..	Spending.Score..1.100.
1	1	Male	19	15	39
2	2	Male	21	15	81
3	3	Female	20	16	6
4	4	Female	23	16	77
5	5	Female	31	17	40
6	6	Female	22	17	76
7	7	Female	35	18	6
8	8	Female	23	18	94
9	9	Male	64	19	3
10	10	Female	30	19	72

Resim 3.1: Başlangıç Veri Seti

Mall Customers veri seti(Resim: 3.1), bir alışveriş merkezinde müşterilere ait demografik ve davranışsal verileri içermektedir. Kümeleme işlemlerinde Müşteri Segmentasyonu amaçlanmaktadır. Veri Seti toplamda 200 gözlemden ve aşağıdaki değişkenlerden olumşaktadır.

- **CustomerID:** Her bir müşteriye atanmış benzersiz bir kimlik numarası
- **Gender:** Müşterinin cinsiyeti(Kategorik: *Male* veya *Female*)
- **Age:** Müşterinin yaşı (Sayısal)
- **Annual Income (k$):** Müşterinin yıllık geliri(Bin dolar cinsinden, sayısal)
- **Spending score (1-100):** Müşterinin harcama alışkanlıklarını temsil eden bir puan(1 ile 100 arasında, Sayısal)

Bu çalışmada sayısal değişkenlere odaklanmak için kategorik değişkenler çıkarılmış ve veri setindeki eksik değerler temizlenmiştir(Resim: 3.2). Ayrıca, Analize hazır hale getirmek için veri seti standartlaştırılmıştır(Resim: 3.2).

```
data <- data[,-c(1,2)]
data <- na.omit(data)
st.data <- scale(data)
```

Resim 3.2: Veri temizleme Kodu

3.2 Final Veri Seti

Gerekli işlemler yapıldıktan sonra final veri seti(Resim: 3.3) hazırlanmıştır.

	Age	Annual.Income..k..	Spending.Score..1.100.
1	-1.42100291	-1.73464625	-0.433713114
2	-1.27782881	-1.73464625	1.192711064
3	-1.34941586	-1.69657236	-1.711617825
4	-1.13465471	-1.69657236	1.037813523
5	-0.56195833	-1.65849848	-0.394988729
6	-1.20624176	-1.65849848	0.999089138
7	-0.27561014	-1.62042459	-1.711617825
8	-1.13465471	-1.62042459	1.696128071
9	1.80041426	-1.58235070	-1.827790981
10	-0.63354538	-1.58235070	0.844191597

Resim 3.3: Final Veri Seti

Bölüm 4

Kümeleme İçin Optimal Küme Sayısının Belirlenmesi

4.1 Optimal Küme Sayısı Belirlenmesi Ne İşe Yarar ?

Kümeleme analizi, veri noktalarının benzerliklerine göre gruplara ayrılmasını sağlar. Ancak bu işlemin etkili ve anlamlı olabilmesi için uygun küme sayısının belirlenmesi gereklidir. Uygun bir küme sayısı seçilmediğinde şu sorunlar ortaya çıkabilir:

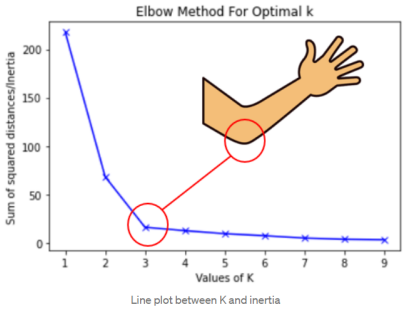
- **Yanlış Sonuçlar:** Verilerin yanlış kümelenmesi, analiz sonuçlarını güvenilmez hale getirebilir. Yanıltıcı gruplar, yanlış çıkarımlar yapılmasına yol açabilir.
- **Verimsiz Kaynak Kullanımı:** Çok fazla küme seçilmesi, analiz sürecini gereksiz yere karmaşıklaştırabilir ve kaynak israfına yol açabilir.
- **Model Performansında Düşüş:** Tahmin ve sınıflandırma modellerinin doğruluğu azalabilir, bu da pratik uygulamalarda yetersiz sonuçlara neden olabilir.

Diğer yandan, doğru küme sayısının belirlenmesi, verilerin daha iyi anlamlandırılmasını sağlar ve analiz sonuçlarının güvenirliğini artırır. Ayrıca, kaynakların daha etkin kullanılmasına ve model performansının iyileştirilmesine katkı sağlar.

Optimal küme sayısını belirlemek için istatistiksel analizden yararlanılır. Bu analizler, kümeleme algoritmalarını veri setine uygun hale getirmek için rehberlik eder

4.2 Elbow (Dirsek) Yöntemi

Elbow yöntemi, toplam hata kareleri (WCSS) değerlerinin küme sayısına bağlı olarak değişimini inceleyen bir yaklaşımdır. Hataların kümülatif olarak azaldığı bir grafikte, "dirsek" olarak adlandırılan nokta, optimal küme sayısını ifade eder.



Resim 4.1: Elbow (Dirsek) Yöntemi

4.2.1 Elbow Yöntemi Algoritması

1. Her bir veri noktası, belirlenen kümesine atanır ve kendi kümesi içindeki diğer noktalara olan mesafeleri hesaplanır.
2. Mesafelerin toplamı (WCSS) bulunur. Küme sayısı arttıkça bu değer azalır çünkü daha fazla küme, noktaların kendi gruplarına daha yakın olmasını sağlar.
3. Grafikte belirgin bir yavaşlama veya eğim kırılması olan nokta, optimal küme sayısını gösterir.

4.2.2 Öne Çıkan Özellikler ve Sınırlamalar

Öne Çıkan Özellikler:

- Görsellik ve basitlik açısından oldukça kullanışlıdır. Analizi görsel olarak destekler.
- Hızlı ve kolay bir şekilde uygulanabilir

Sınırlamaları:

- Çok büyük veri setlerinde dirsek noktasını tespit etmek zor olabilir.
- Dirsek noktası her zaman net bir şekilde belirgin olmayabilir.

4.3 Silhouette Yöntemi

Silhouette yöntemi, her bir veri noktasının kendi kümesine olan yakınlığı ile diğer kümelere olan uzaklığı arasında bir karşılaştırma yapar. Bu yöntemde, -1 ile +1 arasında bir Silhouette skoru hesaplanır.

4.3.1 Silhouette Yöntemi Algoritması

1. Bir noktanın kendi kümesine olan ortalama mesafesi hesaplanır.
2. Aynı noktanın en yakın diğer kümeye olan mesafesi bulunur.
3. Bu iki değer kullanılarak Silhouette skoru şu şekilde hesaplanır:
    - Skor +1'e yaklaştıkça nokta doğru kümeye yerleşmiştir.
    - 0'a yakın skor, noktanın iki küme arasında olduğunu gösterir.
    - -1, noktanın yanlış kümeye yerleştiğini ifade eder.
4. Tüm veri noktaları için ortalama Silhouette skoru hesaplanır ve en yüksek değer optimal küme sayısını işaret eder.

4.3.2 Öne Çıkan Özellikler ve Sınırlamalar

Öne Çıkan Özellikler:

- Kümelerin hem içsel homojenliğini hem de birbirinden ayrışmasını ölçer.
- Daha iyi bir kümelenme değerlendirmesi sunar.

Sınırlamaları:

- Hesaplama süreci, büyük veri setlerinde zaman alıcı olabilir.
- Özellikle yüksek boyutlu verilerde skorlar karmaşıklaşabilir.

4.4 Gap İstatistiği

Gap istatistiği, gerçek veri setini rastgele oluşturulmuş bir veri seti ile karşılaştırır. Bu yöntem, küme yapısının ne kadar anlamlı olduğunu ölçmek için kullanılır.

4.4.1 Gap İstatistiği Algoritması

1. Gerçek veri setine benzer bir rastgele veri seti oluşturulur.
2. Rastgele ve gerçek veri setlerinde kümeleme işlemi gerçekleştirilir.
3. İki veri seti arasındaki fark hesaplanır ve bu farkın maksimum olduğu küme sayısı optimal olarak seçilir.

4.4.2 Öne Çıkan Özellikler ve Sınırlamalar

Öne Çıkan Özellikler:

- Küme yapısının istatistiksel anlamlılığını değerlendirir.
- Farklı veri türleri için esnek bir uygulama sunar.

Sınırlamaları:

- Daha fazla hesaplama gücü ve zaman gerektirir.
- Küçük veri setlerinde, farkları anlamak zor olabilir.

4.5 R Programlama Dili ile Optimal Küme Sayısının Belirlenmesi

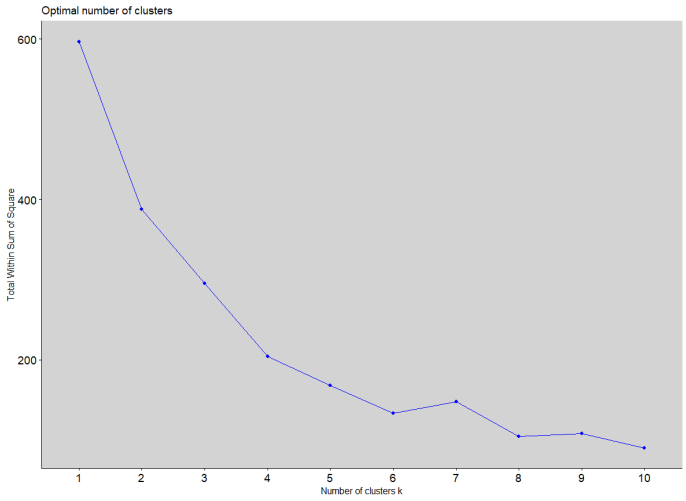
R programlama dilinde, bu yöntemler kullanılarak optimal küme sayısı belirlenebilir.

4.5.1 Elbow Yöntemi

```
fviz_nbclust(st.data,kmeans,"wss",linecolor = "blue")
```

Resim 4.2: Elbow Yöntemi Uygulama Kodları

Gerekli kod(Resim:4.2) kullanılarak Elbow Yöntemi ile optimal küme belirleme grafiği elde edilir.



Resim 4.3: Elbow Yöntemi

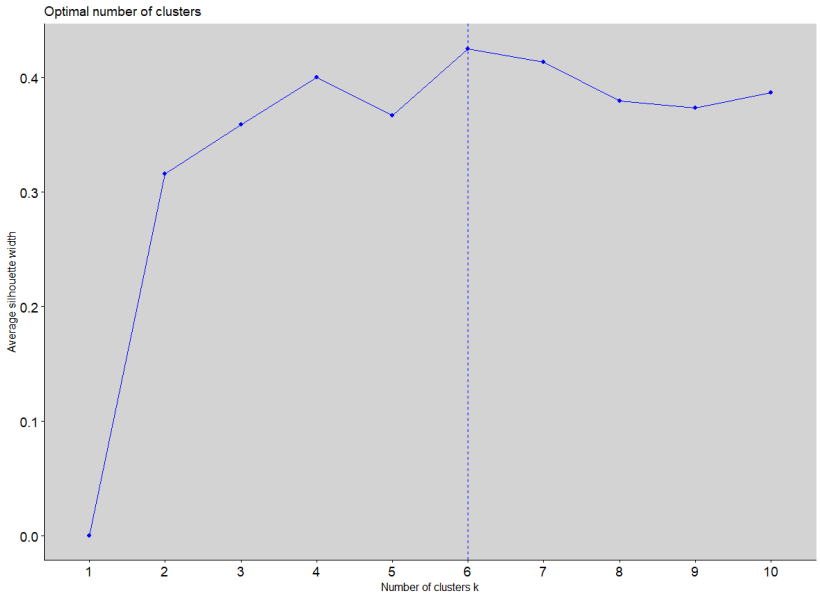
Grafiğe bakıldığında 2,4 ve 6 değerlerinde dirsek noktaları olabileceği görülüyor.

4.5.2 Silhouette Yöntemi

```
fviz_nbclust(st.data,pam,"silhouette")
```

Resim 4.4: Silhouette Yöntemi Uygulama Kodları

Gerekli kod(Resim:4.4) kullanılarak Elbow Yöntemi ile optimal küme belirleme grafiği elde edilir.



Resim 4.5: Silhouette Yöntemi Grafiği

Grafiğe bakıldığında 4,6,7 değerlerinde en yüksek skor görülüyor.

4.5.3 Sonuç

İki yöntem de incelendiğinde 4 ve 6 küme sayılarının bu veri seti için daha uygun olacağı öngörülebilir.

Bölüm 5

K-means

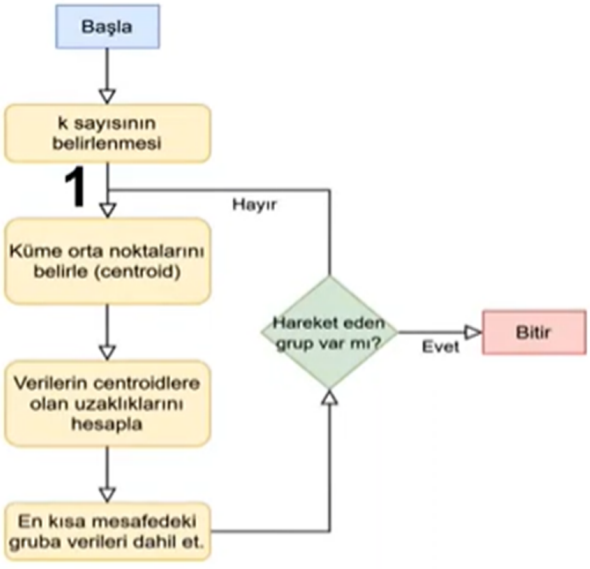
K-Means, büyük veri kümelerini daha küçük ve anlamlı gruplara (kümelere) ayırmayı hedefleyen, makine öğrenmesi alanında sıkça kullanılan bir kümeleme algoritmasıdır. Bu algoritma, özellikle veri noktalarının benzerliklerine göre gruplandırılması gerektiğinde tercih edilir. K-Means, karmaşık veri yapılarındaki gizli desenleri ve ilişkileri ortaya çıkarmak için etkili bir araçtır.

Bu yöntem, özellikle şu durumlarda kullanılır:

- **Veri Keşfi:** Büyük ve karmaşık veri setlerindeki örüntülerin analiz edilmesi.
- **Özellik Gruplandırma:** Verilerin ortak özelliklerine göre segmentlere ayrılması.
- **Ön İşleme:** Daha sonraki analizler için veri setinin basitleştirilmesi.

5.1 K-Means Algoritması

K-Means algoritması, iteratif bir süreçle çalışır ve küme merkezlerini sürekli güncelleyerek en uygun gruplamayı oluşturmayı hedefler. Bu süreç 4 temel adımdan oluşur:



Resim 5.1: K-means Akış Şemaso

Adım Adım Algoritma

1. Başlangıç Küme Merkezlerinin Belirlenmesi: Algoritma, kullanıcının belirlediği K sayıda küme için rastgele merkez noktaları seçer. Bu noktalar, başlangıçta veri kümesindeki gerçek veri noktalarıdır.

2. Veri Noktalarının Kümelere Atanması: Her bir veri noktası, kendisine en yakın olan küme merkezine atanır. Bu yakınlık genellikle Öklid Mesafesi kullanılarak hesaplanır:

$$d(x.y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

3. Küme Merkezlerinin Güncellenmesi: Her bir küme için yeni bir merkez noktası hesaplanır. Yeni küme merkezi, o kümede yer alan veri noktalarının aritmetik ortalaması olarak belirlenir:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

4. İterasyonların Devam Etmesi: Veri noktalarının kümelere atanması ve küme merkezlerinin güncellenmesi işlemleri, küme merkezlerinin konumları sabitlenene kadar veya belirli bir iterasyon sayısına ulaşılana kadar tekrarlanır.

5.2 K-Means Yönteminin Dikkat Çeken Özellikleri ve Kısıtları

Öne Çıkan Özellikler

- K-Means algoritmasının basit yapısı, hızlı bir şekilde uygulanmasını sağlar. Özellikle büyük veri setlerinde bile etkili sonuçlar üretebilir.
- Farklı veri türleri ve uygulama alanlarında kullanılabilir.
- Elde edilen kümeler genellikle sezgisel olarak yorumlanabilir. Özellikle veri görselleştirme ile desteklendiğinde, sonuçlar oldukça açıklayıcıdır.

Kısıtları

- Algoritma başlangıç noktalarına bağlıdır ve farklı başlangıç noktaları farklı sonuçlar üretebilir.
- K sayısının önceden belirlenmesi gerekir. Ancak bu değer genellikle veri setinin özelliklerine bağlıdır.
- Aykırı değerler, küme merkezlerini olumsuz etkileyebilir ve sonuçların anlamlılığını azaltabilir.

5.3 K-Means Yönteminin Uygulanması: R Programlama Dili

kmeans()(Resim: 5.2) kodu kullanılarak belirli bir küme sayısı için kümeleme işlemi gerçekleştirilir.

```
km.res <- kmeans(st.data,2)
km.res <- kmeans(st.data,4)
km.res2 <- kmeans(st.data,6)
```

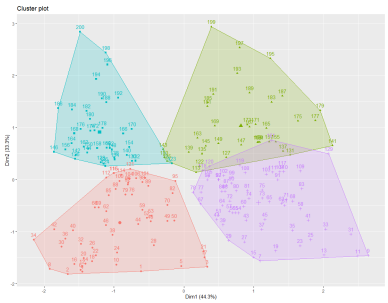
Resim 5.2: K-Means Kümeleme Yöntemi Uygulama kodları

fviz_cluster()(Resim: 5.3) kodu ile elde edilen kümeleme sonuçları grafik haline getirilir

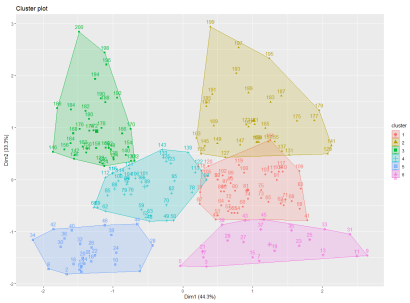
```
fviz_cluster(km.res,data=data)
fviz_cluster(km.res1,data=data)
fviz_cluster(km.res2,data=data)
```

Resim 5.3: Grafik oluşturma kodu

Resim 5.4: 2 Kümeye ayrılmış



Resim 5.5: 4 Kümeye ayrılmış



Resim 5.6: 6 Kümeye ayrılmış

Resim 5.7: Küme sayılarına göre K-Means grafikleri

5.3.1 Sonuç

Verinin doğal yapısını en iyi yansıtan kümeleme genellikle orta düzey bir ayrım sağlar. Görselde 4 kümeye ayrılmış hali (Resim:5.5), hem kümeler arası ayrışma hem de içsel tutarlılık açısından en dengeli sonuçları vermektedir.

Ancak nihai karar, kümeleme sonucunun analizin hedefiyle ne kadar uyumlu olduğuna bağlıdır. Örneğin:

- Daha basit bir genelleme gerekiyorsa 2 küme (Resim: 5.4),
- Daha ayrıntılı alt gruplar gerekiyorsa 6 küme (Resim: 5.6) tercih edilebilir

Bölüm 6

PAM (Partitioning Around Medoids) Kümeleme Yöntemi

PAM, K-Means algoritmasına benzer bir şekilde veri setini önceden belirlenmiş sayıda kümeye ayırmayı amaçlayan bir kümeleme algoritmasıdır. Ancak PAM, K-Means'ten farklı olarak her kümenin merkezini (medoid) bir veri noktası olarak seçer. Bu yaklaşım, PAM'i özellikle aykırı değerlere ve karmaşık veri yapısına karşı daha dayanıklı hale getirir.

Bir medoid, o kümede bulunan veri noktalarına en yakın olan gerçek bir veri noktasıdır. Bu özelliği sayesinde PAM, özellikle K-Means'in başarısız olduğu durumlarda (örneğin, aykırı değerlerin olduğu veri kümelerinde) etkili bir alternatif sunar.

6.1 PAM algoritması

PAM algoritması, aşağıdaki adımları izleyerek kümeleme işlemini gerçekleştirir:

1. Başlangıç Medoidlerinin Belirlenmesi: K sayıda küme için başlangıç medoidleri rastgele seçilir. Medoidler, veri setindeki gerçek veri noktalarından seçilir.
2. Veri Noktalarının Kümelere Atanması: Her bir veri noktası, kendisine en yakın olan medoide atanır. Bu işlem, bir uzaklık metriği (örneğin, Manhattan veya Öklid mesafesi) kullanılarak gerçekleştirilir.
3. Medoidlerin Güncellenmesi: Her bir küme için yeni bir medoid seçilir. Yeni medoid, küme içindeki veri noktaları arasındaki toplam mesafeyi en aza indiren veri noktasıdır.
4. İyileştirme Adımı: Algoritma, mevcut medoidlerle veri noktalarının toplam mesafesini minimize etmeye çalışır. Eğer bir değişiklik daha iyi bir çözüm sağlarsa, medoidler güncellenir ve kümeler yeniden atanır.
5. Sonuçların Değerlendirilmesi: Medoidler sabit kaldığında veya belirli bir iterasyon sayısına ulaşıldığında, algoritma sonlandırılır ve kümeler belirlenmiş olur.

6.2 PAM Algoritmasının Uygulama Alanları

PAM algoritması, geniş bir uygulama yelpazesi sunar ve özellikle aşağıdaki alanlarda etkili bir şekilde kullanılır:

Müşteri Segmentasyonu:
Pazarlama alanında, müşteri davranışlarının sınıflandırılması ve segmentler oluşturulması için kullanılabilir. Örneğin, müşterilerin harcama alışkanlıklarına veya ürün tercihine göre gruplandırılması.

Tıbbi Veri Analizi:
Sağlık verilerinde hastaların benzer semptomlara veya tedavi süreçlerine göre gruplandırılmasında kullanılır. PAM, burada aykırı değerlerin etkisini azaltarak daha sağlam sonuçlar üretebilir.

Metin Madenciliği:
Metin belgelerinin özelliklerine (örneğin kelime sıklıkları veya semantik ilişkiler) göre kümelenmesi için kullanılabilir.

Genomik Analiz:
Genetik verilerin benzerliklere göre gruplandırılmasında etkili bir yöntemdir. Özellikle DNA dizilimleri arasındaki mesafelerin ölçümünde tercih edilir.

Finansal Analiz:
Finansal risk değerlendirmesi, yatırım modellerinin sınıflandırılması gibi işlemlerde kullanılabilir.

6.3 PAM Yönteminin Dikkat Çeken Özellikleri ve Kısıtları

Öne Çıkan Özellikler

- PAM, uç değerlerden daha az etkilenir çünkü medoidler gerçek veri noktalarıdır ve dışsal gürültüye daha dayanıklıdır.
- PAM herhangi bir mesafe metriğiyle çalışabilir; bu, algoritmayı çeşitli veri türleri için uygun hale getirir.

Kısıtları

- PAM, k-means'e göre daha fazla hesaplama gerektirir, çünkü tüm olası medoid kombinasyonlarını göz önünde bulundurur. Bu durum, büyük veri kümelerinde uygulamayı zorlaştırabilir.
- Rastgele başlangıç noktaları, sonuçların değişkenliğine neden olabilir.

6.4 PAM Yönteminin Uygulanması: R Programlama Dili

pam()(Resim: 6.1) kodu kullanılarak belirli bir küme sayısı için kümeleme işlemi gerçekleştirilir.

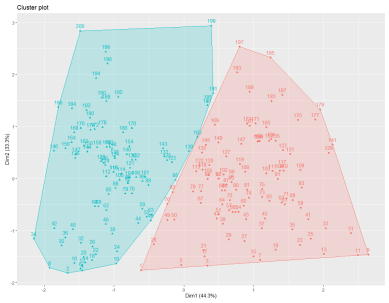
```
pamk_2 <- pam(st.data,2)
pamk_4 <- pam(st.data,4)
pamk_6 <- pam(st.data,6)
```

Resim 6.1: PAM Kümeleme Yöntemi Uygulama kodları

fviz_cluster()(Resim: 6.2) kodu ile elde edilen kümeleme sonuçları grafik haline getirilir

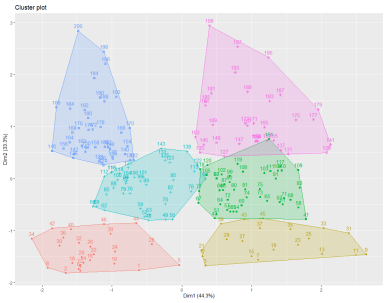
```
fviz_cluster(pamk_2,data=data)
fviz_cluster(pamk_4,data=data)
fviz_cluster(pamk_6,data=data)
```

Resim 6.2: Grafik oluşturma kodu



Resim 6.3: 2 Kümeye ayrılmış

Resim 6.4: 4 Kümeye ayrılmış



Resim 6.5: 6 Kümeye ayrılmış

Resim 6.6: Küme sayılarına göre PAM grafikleri

6.4.1 Sonuç

Görsellerden yola çıkarak 4 kümeleme sonucu (Resim: 6.4), diğerlerine kıyasla daha dengeli bir yapı sergiliyor. İki kümeleme (Resim: 6.3) veriyi fazla genel bir şekilde ayırarak alt grupları yakalayamazken, altı kümeleme (Resim: 6.5) aşırı detaylandırma yaparak bazı kümeler arasında belirgin farkların kaybolmasına neden olabilir. Dört kümeleme, veri setindeki farklı alt yapıları temsil etmek için yeterince ayrıntı sunarken, aynı zamanda kümeler arasında net bir ayrışma sağlayarak veriyi anlamlı bir şekilde özetliyor. Kümeler hem içsel olarak tutarlı hem de birbirinden ayrışmış görünüyor, bu da daha dengeli bir kümeleme yapısına işaret ediyor. Özellikle veri yapısının hem ana hatlarını hem de alt gruplarını anlamlandırmak isteyen analizler için bu sonuç daha uygun bir denge sunuyor.

Bölüm 7

Hiyerarşik Kümeleme Yöntemi

Hiyerarşik kümeleme, verileri çok seviyeli bir yapıda gruplayarak çalışan bir kümeleme algoritmasıdır. Bu yöntem, verileri önceden belirlenmiş bir küme sayısına bölmeksizin çalıştığı için, verideki yapıyı doğrudan ortaya çıkarma avantajına sahiptir. Hiyerarşik kümeleme, özellikle veri yapısının çok detaylı incelenmesi gerektiği durumlarda etkili bir yöntemdir.

7.1 Hiyerarşik Kümeleme Çeşitleri

Hiyerarşik kümeleme iki temel yaklaşımla uygulanabilir:

1. **Birleştirici (Agglomerative) Yöntem:**
   - Bu yaklaşımda, her bir veri noktaları öncelikle kendi başına bir küme olarak kabul edilir.
   - Daha sonra, en yakın küme çiftleri birleştirilerek daha büyük küme yapıları oluşturulur.
   - Bu işlem, tüm veri noktaları birleşip tek bir küme haline gelene kadar devam eder.
2. **Bölücü (Divisive) Yöntem:**
   - Bu yöntem, tüm veri setini tek bir büyük küme olarak kabul ederek başlar.
   - Daha sonra, küme en uzak veya en az benzer alt küme yapılarına ayrılır.
   - Bu bölme işlemi, tüm veri noktaları kendi başlarına birer küme olana kadar devam eder.

7.2 Bağlantı Yöntemleri

Bağlantı yöntemi, hangi küme çiftlerinin birleştirileceğini veya ayrılacağını belirleyen bir kriterdir. Hiyerarşik kümeleme sürecinde yaygın olarak kullanılan bağlantı yöntemleri şunlardır:

7.2.1 Tek Bağlantı Yöntemi

Tek bağlantı yöntemi, en yakın iki veri noktası arasındaki mesafeyi esas alarak kümeleme işlemini gerçekleştirir. Bu yaklaşım, kümeler arasındaki en kısa mesafeyi dikkate aldığı için uzun ve zincir benzeri küme yapıları oluşturabilir. Ancak, gürültülü verilere karşı hassasiyet gösterebilir ve bu durum, kümelerin doğal yapısının bozulmasına neden olabilir.

7.2.2 Tam Bağlantı Yöntemi

Tam bağlantı yöntemi, kümeler arasındaki en uzak mesafeyi dikkate alarak işlem yapar. Bu yöntem, daha kompakt ve sıkı küme yapıları oluşturmayı hedefler. Gürültülü verilere karşı daha dayanıklı olması, bu yöntemi daha sağlam bir seçenek haline getirir. Ancak, çok büyük veri setlerinde hesaplama maliyeti artabilir.

7.2.3 Ortalama Bağlantı Yöntemi

Ortalama bağlantı yöntemi, küme içindeki tüm veri noktalarının ortalama uzaklıklarına dayanır. Bu yöntem, dengeli bir küme yapısı sağlar ve aşırı büyük veya aşırı küçük kümelerin oluşmasını engeller. Ortalamalara dayalı bir yaklaşım olduğu için, verilerdeki uç değerlere karşı diğer yöntemlere kıyasla daha az duyarlıdır.

7.2.4 Ward Yöntemi

Ward yöntemi, toplam varyansı minimize etmeye çalışarak daha homojen ve kompakt küme yapıları oluşturmayı hedefler. Bu yöntem, kümelerin iç yapısını optimize ederek daha anlamlı sonuçlar elde edilmesini sağlar. Hiyerarşik kümeleme süreçlerinde sıklıkla kullanılan ve popüler bir yöntemdir. Özellikle, veri analizinde homojenliği sağlama ihtiyacı olduğunda Ward yöntemi ön plana çıkarır.

7.3 Hiyerarşik Kümeleme Yöntemi Algoritması

Hiyerarşik kümeleme algoritması şu temel adımlardan oluşur:

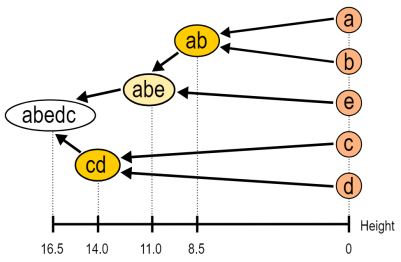
1. Başlangıç: Her veri noktası kendi başına bir küme olarak başlar.
2. Mesafe Matrisi Hesaplama: Veri noktaları arasındaki tüm çiftlerin mesafesi hesaplanır. Örneğin, Öklidyen mesafesi kullanılabilir.
3. Kümeleri Birleştir: İki kümeyi, aralarındaki mesafe en kısa olan kümeleri birleştir. Bu, en yakın iki kümeyi seçmek anlamına gelir.
4. Mesafe Matrisi Güncelle: Yeni küme ile diğer kümeler arasındaki mesafeyi güncelle. Bu adımda farklı bağlantı yöntemleri kullanılabilir (tek, tam, ortalama bağlantı gibi).
5. Dendrogram Güncelle: Yeni kümenin oluşumunu ve ilgili mesafeyi dendrogramda göster.
6. Tekrarla: Tüm veri noktaları tek bir küme içinde toplanana kadar 3. adımdan itibaren tekrarla.
7. Sonuç: Dendrogram tamamlandığında, belirli bir kesme noktasına kadar olan dallanmalar belirli sayıda küme oluşturur.

7.4 Dendrogram

Dendrogram Nedir?

Dendrogram, hiyerarşik kümeleme analizinde kullanılan ve verilerin kümeleme yapısını görselleştiren bir ağaç şemasıdır. Bu grafik, nesneler arasındaki benzerlik veya farklılıkları hiyerarşik bir düzende gösterir. Dendrogramda, her bir yaprak verisetindeki bireysel nesneleri temsil ederken dallar, bu nesnelerin benzerliklerine göre oluşturulan grupları ifade eder. Dikey eksende bağlantı yüksekliği, iki kümenin birbirinden ne kadar farklı olduğunu gösterir ve bu genellikle bir mesafe metriğiyle hesaplanır.

Dendrogram, özellikle biyoinformatikte genetik analizlerde, müşteri segmentasyonunda, doküman sınıflandırmasında ve görüntü işlemelerinde yaygın olarak kullanılır. Bu araç, verisetindeki yapıyı anlamaya ve hangi kümeleme seviyesinin en uygun olduğunu belirlemeye yardımcı olur.



Resim 7.1: Dendrogram Yapısı

7.5 Hiyerarşik Kümeleme Yönteminin Uygulanması: R Programlama Dili

- *agnes()* kodu(Resim:7.2) kullanılarak Birleştirici Kümeleme Yöntemi ile kümeleme işlemi yaptırılır.
- *diana()* kodu(Resim:7.2) kullanılarak ise Bölücü Kümeleme Yöntemi ile kümeleme işlemi yaptırılır.

```
agnes. <- agnes(st.data,metric = "euclidean", method = "ward")
diana. <- diana(st.data,metric = "euclidean")
```

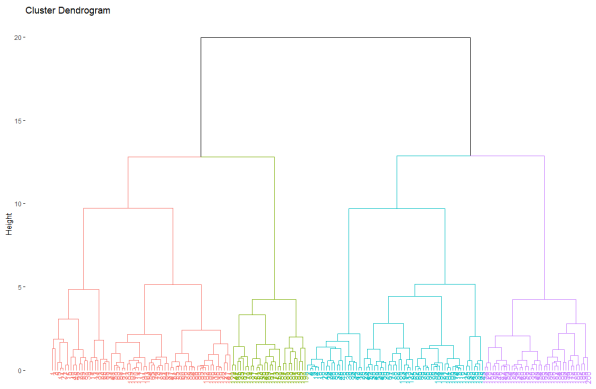
Resim 7.2: Hiyerarşik Kümeleme Yöntemi Kodları

7.5.1 Dendrogram Üzerinde Görselleştirme

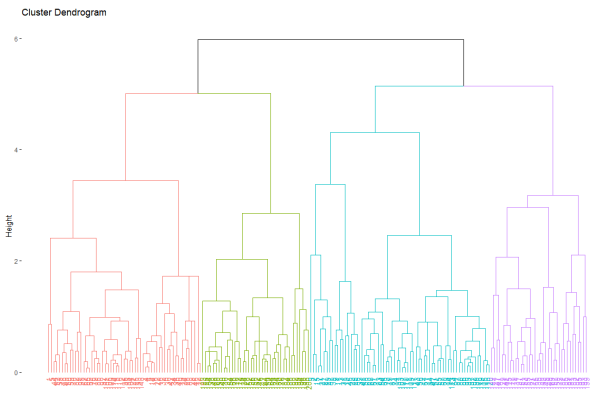
Ardından dendrogram üzerinde görselleştirme kodları(Resim: 7.3) ile dendrogramlar oluşturulur.

```
fviz_dend(agnes.,k=4)
fviz_dend(diana.,k=4)
```

Resim 7.3: Dendrogram Oluşturma Kodları



Resim 7.4: Birleştirici Yöntem



Resim 7.5: Bölücü Yöntemi

Resim 7.6: Dendrogram Grafikleri

7.5.2 Kümeleme Grafiği Üzerinde Görselleştirme

Kümeleme grafiği üzerinde görselleştirmek için önce *cuttree()* kodunu kullanarak ayırmasını istediğimiz küme sayısını ve veriyi belirterek o küme sayısı için kümeleme işlemini yaptırıyoruz.

```
agnes.cl <- cutree(agnes., k=4)
diana.cl <- cutree(diana., k=4)
```

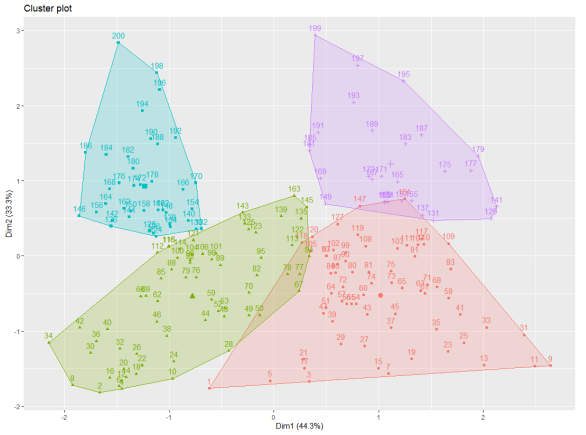
Resim 7.7: Hiyerarşik Kümeleme yönteminde K kümeye ayırma kodları

Ardından gerekli kodları(Bölüm: 7.8) kullanarak kümeleme grafiği üzerinde görselleştiriyoruz.

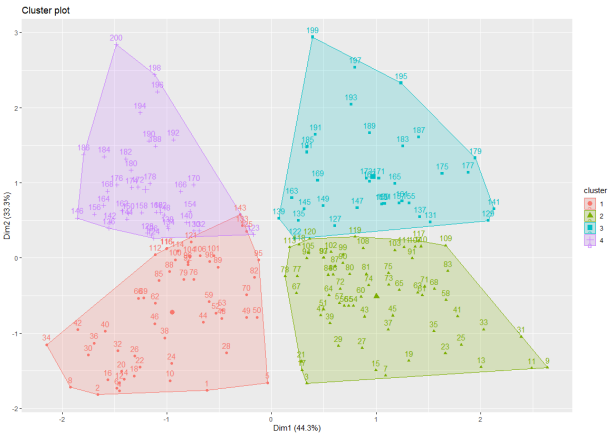
```
fviz_cluster(list(data=st.data,cluster=agnes.cl),
             show.clust.cent=T)

fviz_cluster(list(data=st.data,cluster=diana.cl),
             show.clust.cent=T)
```

Resim 7.8: Hiyerarşik kümeleme yöntemilye kümeleme grafiği oluşturma kodları



Resim 7.9: Birleştirici Yöntem

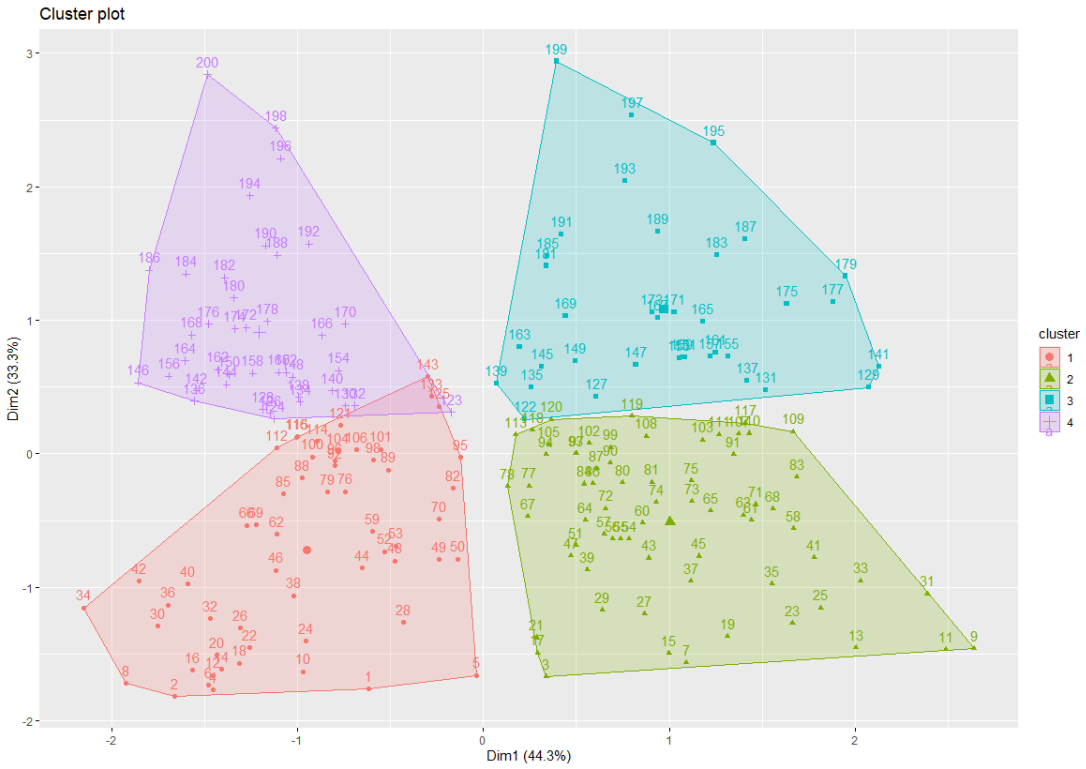


Resim 7.10: Bölücü Yöntemi

Resim 7.11: Hiyerarşik Kümeleme Yöntemi Grafikleri

7.6 Sonuç

Bu iki grafik, birleştirici yöntem (Resim: 7.4) ve bölücü yöntem (Resim: 7.5) kullanılarak yapılan kümeleme sonuçlarını göstermektedir. Birleştirici yöntem ile oluşturulan kümeler daha organik ve birbirine geçişli görünürken, bölücü yöntem ile oluşturulan kümeler daha keskin sınırlarla belirlenmiştir. Özellikle, bazı noktaların hangi kümeye ait olduğu konusunda yöntemler arasında farklılıklar bulunmaktadır. Bu durum, iki yöntemin farklı algoritmik stratejilerinden kaynaklanmaktadır. Bölücü yöntem, başlangıçta büyük bir grubu ayırarak ilerlediği için daha belirgin sınırlar oluşturabilirken, birleştirici yöntem küçük grupları birleştirerek daha esnek kümeler ortaya koyar. Bölücü Yöntemi ile oluşturulan bu çalışma için daha verimli bi kümeleme işlemi olmuştur.



Bölüm 8

Kümeleme Yöntemlerini Ve Sonuçlarını Karşılaştırma

8.1 Kümeleme Yöntemleri Karşılaştırmaları

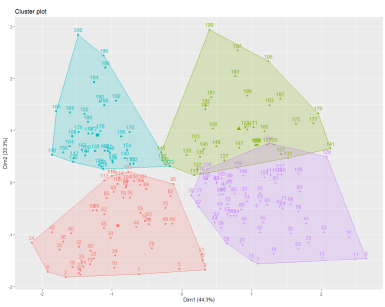
Kümeleme analizinde kullanılan yöntemler, veri setinin yapısına ve analiz amacına göre farklı avantajlar sunar. Aşağıdaki tabloda, **K-Means**, **PAM** ve **Hiyerarşik Kümeleme** yöntemlerinin temel özellikleri karşılaştırılmıştır.

Özellik	**K-Means**	**PAM**	**Hiyerarşik Kümeleme**
Kümeleme Türü	Ayrık	Ayrık	Hiyerarşik
Merkez Temsilcisi	Küme merkezi (ortalama)	Medoid (gerçek bir veri noktası)	Ağaç yapısı (dendrogram)
Hız	Hızlı	Orta	Yavaş
Dayanıklılık	Gürültü ve uç değerlere duyarlı	Gürültü ve uç değerlere dayanıklı	Gürültüye orta derecede dayanıklı
Veri Türü	Sayısal	Sayısal ve kategorik	Sayısal ve kategorik
Küme Sayısı Belirleme	Önceden belirlenir	Önceden belirlenir	Sonradan dendrogram ile karar verilir
Uygulama Alanları	Büyük veri kümeleri	Küçük veri kümeleri	Küçük-orta veri kümeleri

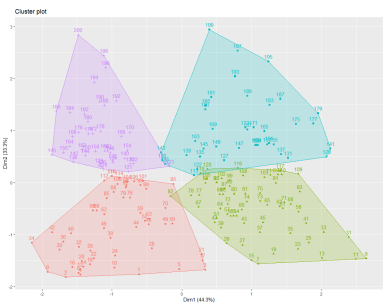
Tablo 8.1: K-Means, PAM ve Hiyerarşik Kümeleme Karşılaştırması

8.2 Kümeleme Sonuçlarını Karşılaştırma

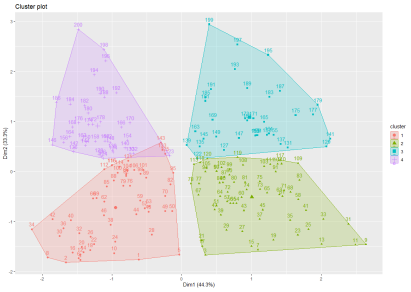
8.2.1 Final Grafikleri



Resim 8.1: K-Means



Resim 8.2: PAM



Resim 8.3: Bölücü Hiyerarşik Kümeleme

Resim 8.4: Kümeleme Yöntemleri Final Grafikleri

Grafiklere Dayalı Yorum:

Müşteri segmentasyonu amaclı veriyi farklı kümeleme yöntemleri ile kümeledikten sonra 4 küme için oluşan grafiklere bakıldığında bu veri seti için PAM kümeleme yönteminin daha başarılı olduğu görülmektedir. Diğer kümeleme yöntemlerinde daha fazla küme çakışmaları varken en net şekilde PAM yöntemi sonucu elde edilen kümeleme en başarılı kümeleme olduğu görülmektedir.

8.3 Sayısal Veriler İle Karşılaştırma

***clValid()* Fonksiyonu**

Ana fonksiyon olan clValid(), birden fazla kümeleme algoritmasını aynı anda değerlendirmek için kullanılır.

Fonksiyon, farklı algoritmaların performansını tablolar veya grafiklerle sunar. Genelde aşağıdaki ölçütleri verir:

- **İçsel Ölçütler (Internal):** Küme içi sıkılık (cohesion) ve küme arası ayrıklık (separation).
- **Dışsal Ölçütler (External):** Kümeleme sonuçlarını bilinen etiketlerle karşılaştırır.
- **Stabilite Ölçütleri (Stability)**: Algoritmanın farklı alt veri setlerinde tutarlı sonuçlar verip vermediğini değerlendirir.

8.3.1 İçsel Ölçütler

İçsel ölçütler (internal validation metrics), kümeleme analizi sonuçlarını değerlendirirken kullanılan ve yalnızca veri setinin kendisine (etiket bilgisi olmadan) dayanan ölçütlerdir. Bu ölçütler, kümelerin birbirine ne kadar uyumlu ve ayrışmış olduğunu anlamaya çalışır. İçsel ölçütler, modelin başarısını objektif bir şekilde değerlendirmenize yardımcı olur ve doğru küme sayısını belirlemeye destek sağlar.

İçsel Ölçütlerin Temel Amaçları

- **Sıkılık (Cohesion):** Aynı kümede bulunan veri noktalarının birbirine ne kadar yakın olduğunu ölçer. İdeal bir durumda, aynı kümedeki noktalar birbirine çok yakın olmalıdır.
- **Ayrıklık (Separation):** Farklı kümeler arasındaki mesafeyi ölçer. İdeal bir durumda, farklı kümeler birbirinden olabildiğince uzak olmalıdır.

8.3.2 R Üzerinde İçsel Ölçüt Hesaplaması

Gerekli kodlar(Resim: 8.5) hesaplamalar elde edilir ardından çıktı(Resim: 8.6) elde edilir

```
clmethods <- c("hierarchical","kmeans","pam")
intern <- clValid(st.data, nClust = 4,
                  clMethods = clmethods, validation = "internal")
summary(intern)
```

Resim 8.5: R Üzerinde İçsel Ölçüt Hesaplama Kodları

```
Clustering Methods:
 hierarchical kmeans pam

Cluster sizes:
 4

Validation Measures:
                                  4

hierarchical Connectivity  16.0488
             Dunn           0.1007
             Silhouette     0.3839
kmeans       Connectivity  34.4337
             Dunn           0.0593
             Silhouette     0.4040
pam          Connectivity  31.8710
             Dunn           0.0551
             Silhouette     0.4004

Optimal Scores:

             Score   Method       Clusters
Connectivity 16.0488 hierarchical 4
Dunn          0.1007 hierarchical 4
Silhouette    0.4040 kmeans       4
```

Resim 8.6: R Üzerinde İçsel Ölçüt Hesaplama Çıktısı

Yorum:

Elde edilen sonuçlara göre, kümeleme yöntemleri arasında Connectivity ve Dunn ölçütlerinde en iyi performansı Hierarchical kümeleme göstermiştir. Bu, kümeler arasındaki ayrışmanın daha net olduğunu ve aynı küme içindeki veri noktalarının daha sıkı bir şekilde gruplandığını göstermektedir. Öte yandan, Silhouette skorunda en iyi sonucu K-Means kümeleme almıştır, bu da bireysel veri noktalarının ait olduğu kümelere daha iyi uyum sağladığını ifade eder. PAM kümeleme, tüm ölçütlerde orta düzeyde bir performans sergileyerek, Silhouette skoru açısından K-Means'e yakın bir sonuç elde etmiştir.

8.3.3 Stabilite Ölçütleri

Stabilite ölçütleri, kümeleme sonuçlarının tutarlılığını ve güvenilirliğini değerlendiren metriklerdir. Bu ölçütler, özellikle veri setinde küçük değişiklikler yapıldığında (örneğin, bazı veri noktalarının çıkarılması veya alt veri setleriyle analiz yapılması gibi) kümeleme algoritmasının ne kadar tutarlı sonuçlar ürettiğini inceler. Stabilite ölçütleri, kümeleme analizinde modelin dayanıklılığını anlamak ve en iyi yöntemi seçmek için önemli bir rol oynar.

Stabilite Ölçütlerinin Temel Amaçları

- **Tekrar Üretilebilirlik (Reproducibility)**: Veri setinde küçük değişiklikler yapıldığında (örneğin, rastgele bir alt küme alındığında), aynı algoritmanın benzer kümeler üretip üretmediğini ölçer.
- **Hassasiyet (Sensitivity):** Veri setindeki küçük değişimlerin veya rastgeleliğin kümeleme sonuçlarını ne ölçüde etkilediğini değerlendirir.
- **Kararlılık (Consistency):** Farklı alt veri setleri kullanıldığında kümeleme yapısının değişmeden ne kadar korunabildiğini inceler.

8.3.4 R Üzerinde Stabilite Ölçütleri Hesaplaması

Gerekli kodlar(Resim: 8.7) hesaplamalar elde edilir ardından çıktı(Resim: 8.8) elde edilir

```
stab<-clValid(st.data,nClust=4,clMethods=clmethods,
              validation="stability")
summary(stab)
```

Resim 8.7: R Üzerinde Stabilite Ölçütleri Hesaplama Kodları

```
Validation Measures:
                         4

hierarchical APN  0.2506
             AD   1.7034
             ADM  0.8859
             FOM  0.9549
kmeans       APN  0.3127
             AD   1.6195
             ADM  0.8218
             FOM  0.9484
pam          APN  0.2872
             AD   1.5905
             ADM  0.7664
             FOM  0.9350

Optimal Scores:

    Score  Method       Clusters
APN 0.2506 hierarchical 4
AD  1.5905 pam          4
ADM 0.7664 pam          4
FOM 0.9350 pam          4
```

Resim 8.8: R Üzerinde Stabilite Ölçütleri Hesaplama Çıktısı

Elde edilen sonuçlara göre, **APN** ölçütünde en düşük skor 0.2506 ile **Hiyerarşik Kümeleme** yöntemine aittir ve bu, veri setindeki küçük değişikliklere karşı en stabil sonuçları verdiğini gösterir. **AD** ölçütünde ise en düşük değer 1.5905 ile **PAM kümeleme** yöntemi tarafından elde edilmiştir, bu da veri noktalarının alt kümelerdeki mesafeleri açısından en kararlı yöntemin PAM olduğunu belirtir. **ADM** ölçütünde 0.7664 ile yine **PAM kümeleme** en iyi performansı göstermiştir, bu durum alt veri setlerine göre gruplandırmanın daha tutarlı olduğunu ifade eder. Son olarak, **FOM** ölçütünde en düşük değer 0.9350 ile **PAM kümeleme** yöntemine aittir, bu da çıkarılan veri noktalarının kümelere daha iyi tahmin edildiğini göstermektedir. Genel olarak, PAM kümeleme stabilite açısından öne çıkarken, Hierarchical kümeleme APN ölçütünde daha iyi bir stabilite sağlamıştır.

Bölüm 9

Sonuç

Bu çalışmada, müşteri segmentasyonu amacıyla K-Means, PAM ve Hiyerarşik Kümeleme yöntemleri uygulanarak elde edilen sonuçlar karşılaştırılmıştır. Her bir yöntemin avantajları, sınırlıkları ve öne çıkan performans özellikleri detaylı bir şekilde analiz edilmiştir. Çalışmanın öne çıkan bulguları şu şekilde özetlenebilir:

K-Means Kümeleme:

K-Means yöntemi, basit yapısı ve hızlı uygulanabilirliği ile dikkat çekmektedir. Ancak, bu yöntemin başlangıç merkezlerine duyarlı olması ve aykırı değerlerden etkilenmesi, kümeleme sonuçlarının bazı durumlarda tutarsız olmasına neden olmaktadır. Elde edilen sonuçlar, özellikle 4 küme durumunda, kümeleme yapısının genel olarak dengeli olduğunu göstermiştir.

PAM (Partitioning Around Medoids) Kümeleme:

PAM algoritması, medoid bazlı yaklaşımı sayesinde aykırı değerlere karşı daha dayanıklıdır. Bu özelliği, özellikle verinin doğal yapısını ortaya çıkarmak ve segmentasyon işlemleri için önemlidir. 4 küme seçeneğiyle elde edilen sonuçlar, kümeleme arasındaki farklılıkları net bir şekilde gözlemleme imkânı sunmuş ve PAM yönteminin bu veri seti için en başarılı sonucu verdiğini göstermiştir.

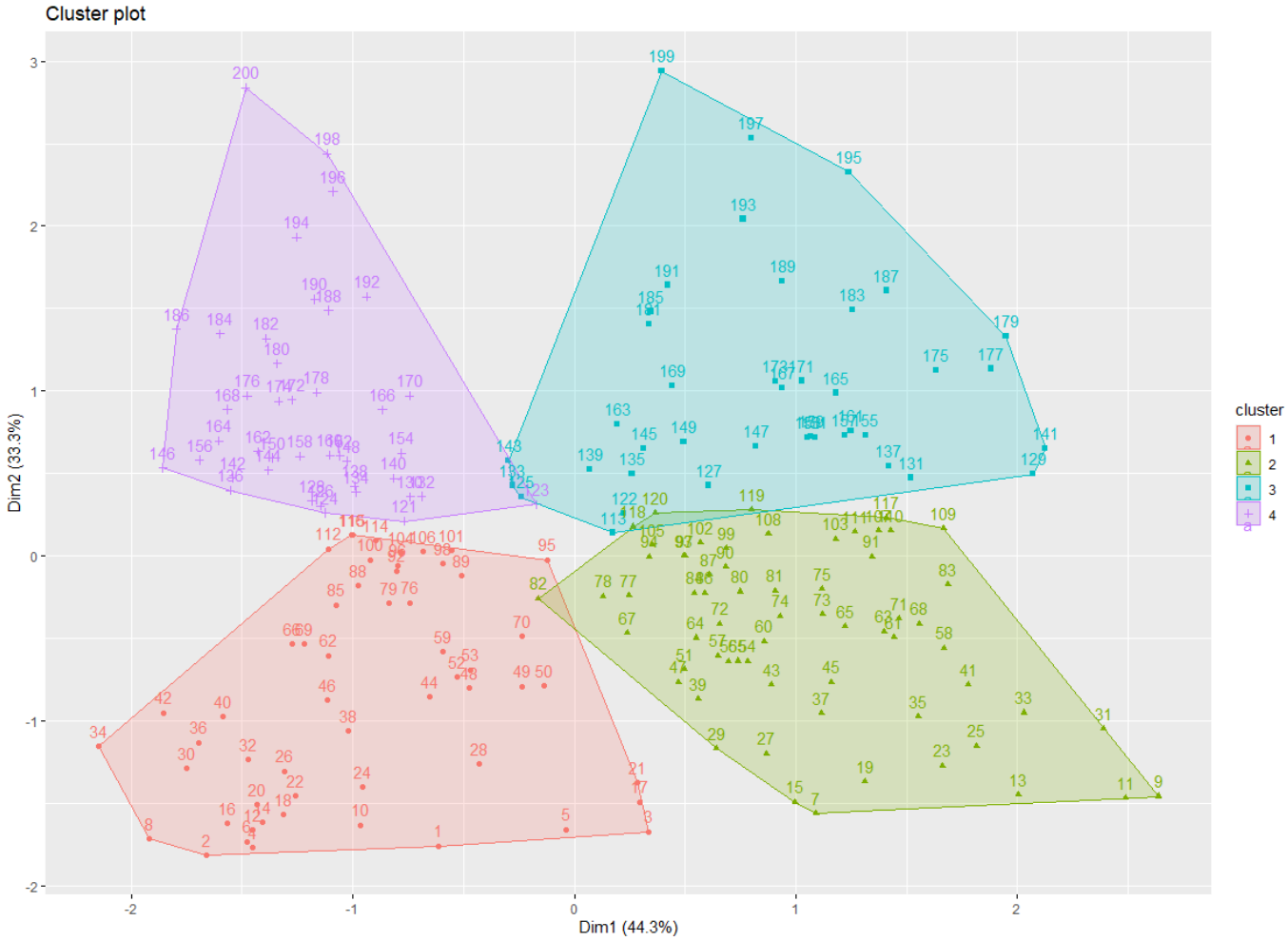
Hiyerarşik Kümeleme:

Hiyerarşik kümeleme yöntemi, dendrogram gibi görselleştirme aracıyla kümeleme yapısının daha detaylı bir şekilde analiz edilmesine olanak tanımıştır. Bu yöntem, veri setindeki hiyerarşik yapıyı görmek isteyen kullanıcılar için faydalı olmuş, ancak hesaplama maliyetlerinin yüksek olması ve çok sayıda veriyle uygulandığında performansın düşmesi dezavantaj olarak dikkat çekmiştir.

Genel Değerlendirme:

Tüm yöntemler karşılaştırıldığında, PAM(Resim: 9.1) yöntemi bu veri seti üzerinde en başarılı sonuçları sağlamıştır. K-Means yöntemi, hızlı ve kolay uygulanabilirliğiyle dikkat çekse de, özellikle aykırı değerlerin etkisi nedeniyle bazı kümelerde dengesiz sonuçlar vermiştir. Hiyerarşik kümeleme ise detaylı analiz isteyen kullanıcılar için uygun bir alternatif sunmuş, ancak hesaplama maliyeti yüksek olmuştur.

Sonuç olarak, çalışma, farklı kümeleme yöntemlerinin çeşitli veri yapıları üzerindeki etkilerini anlamak ve segmentasyon işlemleri için en uygun yöntemi belirlemek adına önemli katkılar sağlamıştır. Bu kapsamda, PAM yöntemi, sağladığı dengeli ve tutarlı sonuçlarla öne çıkmıştır.



Resim 9.1: En Başarılı Kümeleme Grafiği

Resim Listesi

Tablo Listesi