T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

K – ORTALAMALAR ALGORİTMASI İLE İLERİYE DÖNÜK MODELLEMELER

KEMAL KOŞUTA

YÜKSEK LİSANS TEZİ
MATEMATİK MÜHENDİSLİĞİ ANABİLİM DALI
MATEMATİK MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
PROF. DR. AYLA ŞAYLI

İSTANBUL, 2018

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

K - ORTALAMALAR ALGORİTMASI İLE İLERİYE DÖNÜK MODELLEMELER

Kemal KOŞUTA tarafından hazırlanan tez çalışması 31.05.2018 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Prof. Dr. Ayla ŞAYLI

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Prof. Dr. Ayla ŞAYLI

Yıldız Teknik Üniversitesi _____

Prof. Dr. İbrahim Emiroğlu

Yıldız Teknik Üniversitesi _____

Dr. Öğr. Üyesi Mustafa Zahid Gürbüz

Doğuş Üniversitesi _____

ÖNSÖZ

Akademik hayatımın bu ilk tezinde, günümüzde popüler olan ve popülerliği hızla arttığı gözlemlenen, makine öğrenmesine çalışılmıştır. Makine öğrenmesinin denetimsiz öğrenme algoritmaları başlığı altında bulunan ve önemli başlıklarından biri olan kümeleme analizi alanında yaptığımız çalışmamızda doğru küme sayısının belirlenip yorumlanabilmesinin gerçek veri kümesi üzerindeki uygulamasını gerçekleştirdik. Yaptığımız araştırma ve literatür taraması esnasında bilimin birikimli olarak ilerleyen bir süreç olduğunu yaşayarak tecrübe edindik. Bu çalışmamın da yeni çalışmalar yapacak olanlara bir kaynak olabilmesi en büyük temennimizdir.

Gerek yüksek lisans eğitimim de gerekse bu tezi hazırlarken benden yardımlarını esirgemeyen hocam, tez danışmanım Prof. Dr. Ayla ŞAYLI'ya, yüksek lisans eğitimimde ders aldığım tüm hocalarıma ve destekleriyle beni yalnız bırakmayan çok sevdiğim aileme teşekkürü bir borç bilirim.

Mayıs, 2018

Kemal Koşuta

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ

X	Veri kümesi
k	Küme sayısı
n	Gözlem sayısı
m	Doğru eğimi
y`	Min- max normalleştirmesi sonucunda elde edilen değer
x`	Z-skor normalleştirmesi sonucunda elde edilen değer
x	Öznitelik değeri
μ	Aritmetik ortalama
σ	Standart sapma
Q_1	1. kartil değeri
Q_3	3. kartil değeri
Gap(k)	Gap istatistik değeri
d(x, y)	2 nokta arasındaki Öklid uzaklığı
SC	Durma kriteri
D_r	Öklid uzaklığının kareler toplamı
W_k	Ağırlık değeri

KISALTMA LİSTESİ

BSS	Between Cluster Sum of Squares
GAP	Gap Statistic
IQR	Interquartile Range
R	R Programming Language
SQL	Structured Query Language
WSS	Within Cluster Sum of Squares

ŞEKİL LİSTESİ

Sayfa

ÇİZELGE LİSTESİ

Sayfa

ÖZET

K- ORTALAMALAR ALGORİTMASI İLE İLERİYE DÖNÜK MODELLEMELER

Kemal KOŞUTA

Matematik Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Prof. Dr. Ayla Şaylı

Çağımızdaki üretilen verilerin sayısı hızla artmaktadır ve sürekli artmaya devam edecektir. Makine öğrenmesi, popülerliği gün geçtikçe artmakta olan bir araştırma alanıdır. Makine öğrenmesi algoritmaları veriye dayalı modeller kurulmasına olanak sağlar. Elde edilen veriden faydalanıp değerli bilgilerin çıkarımı yapılması oldukça önemlidir. Bu alanın alt başlıkları ise genel olarak, denetimli öğrenme algoritmaları, kümeleme algoritmaları, birliktelik kuralları olarak adlandırılmıştır. Tez kapsamında kümeleme yöntemlerinden faydalanılacaktır. Yapılan çalışma ile veri kümesinin kaç kümeye ayrılması gerektiği ve daha sonra kümelenen veriler ile rezervasyon işlemleri için ne yapabileceğimiz belirlenecektir.

Bu tezin amacı, kümeleme algoritmalarına ile ileriye dönük modellemeler gerçekleştirmektir. Kümeleme algoritmalarından hiyerarşik ve K - Ortalamalar algoritması üzerinde çalışılmıştır. Hiyerarşik kümelemede elde edilen sonuçlar ile uygun olmadığı anlaşıldığından çalışmanın devamı için K-Ortalamalar algoritması ile

detaylı çalışılmasına karar verilmiştir. K-ortalamalar algoritması kullanılarak dinamik bir kümeleme yapılması sağlanmıştır. Aykırı değer analizi, veri dönüştürme, eksik gözlemleri doldurma gibi veri ön hazırlık aşamalarından sonra Gap istatistik değeri ve Elbow yöntemi ile veri kümesinin kaç kümeye ayrılması gerektiğine karar verilmiştir. Bu aşamada belirlenecek olan k değeri ilk aşamada geniş bir aralıkta tutulmaktadır. Yapılan uygulamada k değeri 2 ile 15 arasında seçilip, her bir k değeri için Gap İstatistik değeri ve Dirsek yöntemi hata terimi hesaplandıktan sonra, çizdirilen grafikler yardımıyla seçilmesi gereken doğru k değeri belirlenmiştir. Belirlenen değerler için K-Ortalamalar algoritması ile kümeleme yapılır. Bu yöntemlere göre belirlenen k değerlerinin farklı çıkması durumunda Davies - Bouldin, Dunn, Calinski – Harabasz, Wemmert Gancarski, ve Silhouette'nun kümeleme değerlendirme kriterleri ile bulunan k değerlerinden hangisinin daha doğru olduğu kesin olarak belirlenmiştir. Bu çalışma ile doğru k değerini belirleme yöntemleri ile kümeleme değerlendirme kriterleri birlikte kullanılıp, veri kümesini kaç kümeye ayırmak gerekir sorusuna cevap verilmiştir. Yapılan çalışma sonucu Turizm sektörü üzerinde, gerçek veriler kullanılarak on farklı modelleme gerçekleştirilmiştir. Elde edilen sonuçlar kıyaslanmıştır ve modellemelerden en kayda değer başarıya sahip olan belirlenmiştir

Anahtar Kelimeler: Kümeleme Analizi, K- Ortalamalar Algoritması, Dirsek Yöntemi, Kümeleme Değerlendirme Kriterleri.

ABSTRACT

FORECASTING MODELLINGS BASED ON K-MEANS ALGORITHM

Kemal KOŞUTA

Department of Mathematical Engineering

MSc. Thesis

Adviser: Prof. Dr. Ayla Şaylı

The number of collected data in our epoch is increasing rapidly and will continue to increase continuously. Machine learning, a research field that is growing popularity day by day. Machine learning algorithms allow for the build models based on the data. It is very important to take advantages of the data and to extract valuable knowledges. Sub-headings of this field are generally called supervised learning algorithms, clustering algorithms and association rules. We will use clustering algorithms within the thesis. The study will determine how many clusters should be separated and then what we can do with the clustered data for reservation processes.

The purpose of this thesis is to realize reservation optimization with the prediction of the conversion rate based on clustering algorithms. Hierarchical and K- Means algorithms have been studied. Since it is understood that the results obtained from the hierarchical clustering are not appropriate, therefore it has been decided to study in detail by K-means algorithms for the continuation of the study. A dynamic clustering is

also achieved by using the K-means algorithm. After the data preparation steps such as outlier analysis, data scaling, filling in missing observations, it was decided how many clusters of data should be separated by Gap statistic value and Elbow method. The k to be determined at this stage is kept in a wide range in the first stage. The true k value between 2 to 15 is selected by the plotted graphs is determined by the Gap statistic and the Elbow method. Their errors are calculated. After the determined values, the clustering is done by K-means algorithms. If the true k values determined by these methods are different, the k values found with the clustering validation criteria by Davies - Bouldin, Dunn, Calinski - Harabasz, Wemmert Gancarski and Silhouette are more precisely determined. In this study, the optimal k value determination methods and clustering evaluation criteria are used together, and the answer is given to how many clusters of data should be separated. A conclusion of the study, on the tourism sector, we work on ten forecasting models by the use of the real data for tourism company. The results from the forecasting models are found, compared and then the model with a remarkable success is obtained

Keywords: Cluster Analysis, K-Means Algorithm, Elbow Method, Cluster Validation Techniques.

BÖLÜM 1

GİRİŞ

1.1 Literatür Özeti

İnternet kullanımının son zamanlarda artmasının da etkisiyle, içinde bulunduğumuz çağ, "Veri ve Bilgi" çağı olarak adlandırılmaktadır. Bilişim teknolojilerinin hızla gelişmesiyle ve veri üretim hızının artmasıyla ilk olarak elde edilen verilerin saklanması, ardından işlenmesi ve son olarak da bu verilerden bilgi çıkarımı yapılması son derece önemli bir durum haline gelmiştir.

Veri ve bilgi kavramları arasında farkı özetleyecek olursak, bilgi; verilerin belirli süreçlerden geçip, işlenip kullanılabilir hale gelmesinin ardından belirlenen amaca yönelik oluşturulan çıktıya denir. Bu amaç için bilginin anlamlılık ve kullanılabilirlik açısından ihtiyacı karşıladığından emin olunmalıdır [1].

Veri madenciliği ile ilgili tek, kesin bir tanım bulunmamaktadır. Literatürde karşımıza çıkan veri madenciliği tanımlarından bazıları şu şekildedir.

Bilgi haline getirilmiş verilerden faydalanarak çıkarım yapmayı anlamlı kılan ve ayrıca yeni eğilimleri, örüntüleri ve ilişkileri ortaya çıkarma süreci olarak tanımlanmaktadır [2].

Elde edilen veriler arasındaki ilişkileri ortaya çıkarmak ve ihtiyaç halinde ileriye dönük tahminler yapabilmek için veri yığınlarının içinden değerli olan bilgileri elde etme tekniğidir. Bir işletme açısından bakarsak, bu işletmede üretilen tüm verilerin veri madenciliği yöntemleri kullanılarak şu anda var olan veya gelecekte ortaya çıkma ihtimali bulunan "değerli bilgileri" ortaya çıkarma aşamalarının tamamına denir [3].

Kısaca veri madenciliği özetlemek gerekirse [4], veriden bilgi çıkarımı sağlanarak verinin anlamlandırılıp faydalı hale getirilmesidir. Mevcut verideki eğilim, kurallar ve örüntüleri bulmak amacıyla verinin çok yönlü analiz edilmesi ve keşfedilmesi aşamasıdır. Verilerin analizi sonucunda edinilen anlamlı bilgilere, güvenirliliği olan karar ve sonuçları elde etmek için verinin bilgiye dönüştürülmesidir.

Depolanabilen veri miktarının artması ve saklama maliyetlerinin düşmesi ile veriler dijital ortamda saklanmaya başlamıştır. Bu gelişmeler bilgi teknolojilerindeki kaydedilen ilerlemenin sonucunda ortaya çıkmıştır. Bu gelişmelerin sonucu olarak 1970'li yılların başlarından itibaren veri tabanı yönetim sistemlerinin gelişmesine paralel olarak verilerin işlenip, bilgiye dönüştürülmesi kolaylaştırılmıştır [5].

Veri tabanı yönetim sistemlerinin oluşturulmasından sonra veri madenciliği yöntemleri ortaya çıkmıştır. Bu sistemler aracılığıyla saklanan veriler kolayca bir araya getirilebilmiş ve SQL dili aracılığıyla, veriler etkin bir şekilde sorgulanabilir duruma gelmiştir [6].

1980'li yılların sonlarından itibaren verilerden anlamlı bilgiler çıkarmak amacıyla veri madenciliği teknikleri uygulanmaya başlamıştır. Bu teknikler aktif olarak girdiye sahip olan sistemlere uygulanmak yerine sadece gerekli bilgilerin olduğu veri kümelerine uygulanır [6].

Günümüzde yaşanan yeni teknolojik gelişmeler sayesinde büyük bir ivme kazanan veri madenciliği çalışmaları her alanda uygulanabilen çok disiplinli bir yapı halini almıştır.

Veri madenciliği teknikleri; gen analizleri, kredi kartı sahteciliklerinin yakalanması, pazar ve rekabet analizleri gibi birçok alanda aktif bir şekilde kullanılmaktadır.

Şekil 1.1 de veri madenciliği yöntemlerinin kullanıldığı bazı sektörler ve uygulama alanları gösterilmiştir.

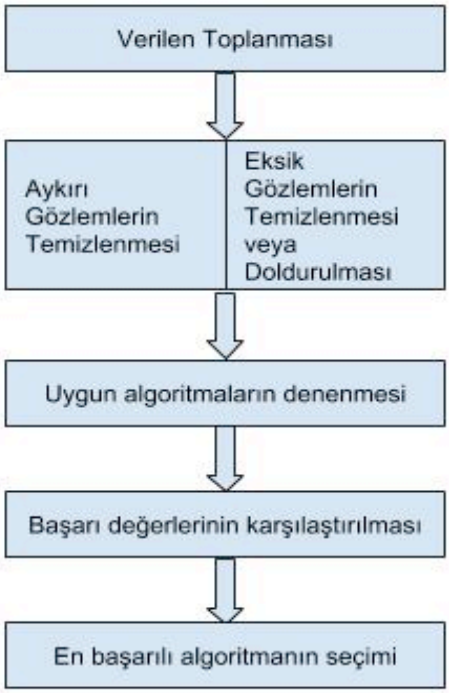
Pazarlama	Müşterilerin gruplanması, Müşteri kayıp analizi
Sağlık Sektörü	Gen analizleri, İlaçlarının yan etkilerinin analizleri
Elektronik Ticaret	Müşteri yorumlarının çözümlenmesi, Müşteri memnuniyetinin araştırılması
Finans Sektörü	Kredi kartı sahteciliklerinin yakalanması, Kredi taleplerinin değerlendirilmesi
Sosyal Medya	Duygu analizleri, Tüketici memnuniyet oranının araştırılması

Şekil 1.1 Veri madenciliği kullanım alanları

Veriler arasındaki etkin ilişkilerin tespit edilmesine yönelik yöntemleri içeren veri madenciliği tekniklerinden olan kümeleme, sınıflandırma ve birliktelik kuralları farklı amaçlar için uygun olan algoritmalar kullanılarak yapılmaktadır.

Veri madenciliği teknikleri her alana uygulanabilmektedir. Veri madenciliği projesinin akışı, verilerin toplanması ile başlayıp, verinin temizlenmesi ile devam etmektedir. Veri temizleme aşamasında en sık kullanılan yöntemler aykırı gözlemlerin belirlenip, mevcut veri kümesinden çıkarılması ve boş değer içeren özelliklerin veri kümesinden kaldırılması veya eksik gözlemlerin doldurulmasıdır. Veri temizleme sürecinin devamında çözülmek istenen problemin sınıflandırma, kümeleme veya birliktelik kuralları alanlarından hangisine uygun ise o alandaki algoritmalar veri kümesine uygulanarak en yüksek başarı elde edilen algoritma seçilir. İlgili problem için hangi tekniğin daha yüksek başarı değeri elde edeceği öngörülemediği için, tek bir algoritma ile çalışmak yerine, mevcut algoritmaların tümü denenir [7].

Şekil 1.2 de veri madenciliği metodolojisi açıklanmıştır.



Şekil1.2 Veri madenciliği metodolojisi

Makine öğrenmesi; verileri kullanarak, veriye dayalı model kurmak ve bu kurulan modele dayanarak tahmin yapılmasına olanak sağlar.

Karar ağaçları, Bayes ağları, Destek vektör makineleri gibi makine öğrenmesi algoritmaları eğitim kümesini kullanarak veriye dayalı model oluşturulduktan sonra başarı ölçütleri ile kıyaslanarak yüksek başarıya sahip olan model tercih edilir.

Makine öğrenmesi 3 aşamada gerçekleşir:

- Verinin temizlenmesi ve görselleştirilmesi
- Öğrenme algoritmalarının uygulanması
- Öğrenme algoritmaların performanslarının değerlendirilmesi [8]

Makine öğrenmesi algoritmalarından daha sağlıklı, tutarlı sonuçlar elde edebilmek ve başarı değerlerinin yükseltilebilmesi için, kullanılacak olan veri kümesine, veri ön hazırlık aşamalarının uygulanması gerekmektedir. Ön hazırlık kısmında ise başlıca aykırı gözlemlerin belirlenip çıkartılması, eksik gözlemlerin doldurulması, verilerin normalleştirilmesi aşamalarından sonra veri kümesi makine öğrenmesi algoritmalarının uygulanmasına hazır hale gelir.

En sık kullanılan öğrenme algoritmaları Denetimli (Supervised) Öğrenme ve Denetimsiz (Unsupervised) Öğrenme yöntemleridir.

Denetimli Öğrenme: Bu öğrenme tipinde, veri kümesinde gözlemlerin ait olduğu sınıf etiketi mevcuttur. Denetimli öğrenme algoritmaları sayesinde bir fonksiyon, ağaç aracılığıyla, gözlemlerin etiketlerinin tahmin edilebilmesine olanak sağlar. Kurulan bu fonksiyon veya ağaç yardımıyla yeni gözlemin sınıfı veya değeri tahmin edilebilir.

Denetimli öğrenmeye örnek olarak, kişilerin kan tahlili sonuçlarının elimizde olduğunu varsayalım. Her bir gözlem, bir hastanın kan değerlerine ait ölçümleri içeriyor ve kişinin hasta olup olmadığının da veri kümesinde mevcut olduğunu düşünelim. Denetimli öğrenme algoritmaları ile kurulan fonksiyon aracılığıyla yeni tahlillerin yorumlanması yapılabilir.

Denetimsiz Öğrenme: Bu öğrenme tipinde, denetimli öğrenme modelindeki gibi bir sınıf etiketi mevcut değildir. Bu tip öğrenme algoritmalarında yapılan iş, gözlemler arasındaki yakınlık, uzaklık durumlarına göre farklı uzaklık ölçütleri kullanılarak kümeleme ile birbirine benzer gözlemlerin kümelenip, gruplara ayrılmasıdır [9]

1.2 Tezin Amacı

Bu tezin amacı, denetimsiz öğrenme tekniklerinden olan kümeleme yöntemlerine dayalı dönüşüm oranı belirlenerek ileriye dönük tahmin modellemeleri gerçekleştirmektir.

Tez kapsamında turizm sektöründeki bir firmanın gerçek verileri kullanılarak modellemeler gerçekleştimektir.

1.3 Hipotez

Bu tezin hipotezi, dönüşüm oranına dayalı rezervasyon modellemeleri için denetimsiz öğrenme tekniklerinden olan kümeleme yöntemlerinın kullanılabilir olup olmadığının belirlenmesidir.

Bu doğrultuda makine öğrenmesinin denetimsiz öğrenme başlığına ait olan kümeleme algoritmalarında, algoritma seçiminden bağımsız olarak ilk iş olarak doğru küme

sayısının belirlenmesi gerekir. Belirlenecek küme sayısı sonuçları doğrudan etkileyeceğinden, bu aşama kritik bir öneme sahiptir. Doğru küme sayısının belirlenmesinde yaygın olarak Elbow yöntemi ve Gap istatistik değeri kullanılır. Tezde yapılan uygulama kapsamında bu iki yöntem ile doğru küme sayısı belirlenecektir. Bu aşamadan sonra kümeleme değerlendirme kriterleri ile belirlenen küme sayılarının doğruluğu sorgulanacaktır. Bu yöntemlere ek olarak, kümeler arası ve küme içi uzaklıklar hesaplanarak farklı bir doğru küme sayısı belirleme yöntemi de incelenecektir. Bu aşamadan sonra kümeleme değerlendirme kriterleri ve hatalar hesaplanarak en az hataya sahip çözümün hangi durumda ortaya çıktığı belirlenecektir.

BÖLÜM 2

VERİ HAZIRLAMA İŞLEMLERİ

2.1 Verinin Dönüştürülmesi

Veri ön hazırlık işlemleri süresince, farklı öznitelikler, farklı ölçekler ile belirtilebilir ve bu yüzden birbirileri ile kıyanlanması doğru değildir. Örnek olarak bir web sitesinin görüntülenme sayısı ile yapılan satış farklı ölçeklerdedir. Web sitesinin görüntülenme sayısı yapılan satış sayısından oldukça büyük olacaktır. Bu durumda gözlemler arasındaki ilişkileri sorgulamak için uzaklık veya benzerlik hesaplandığında, küçük değere sahip özniteliğin önemi, büyük olan özniteliğe göre çok küçük olacaktır ve bu durum sadece büyük değere göre karar vermeye sebep olacak, dolayısıyla küçük gözlemin etkisi ortadan kalkacaktır. Bu sorunu ortadan kaldırmak için verilerin normalize edilmesi gerekmektedir. Normalleştirme teknikleri, gözlemleri daha yakın hale getirmek için gereklidir [10].

Veri madenciliği aşamalarından biri olan veri ön işleme, veri madenciliği sürecinde önemli ve kritik bir adımdır. Bu sebeple bir veri madenciliği projesinin başarısı üzerinde büyük bir etkiye sahiptir. Normalizasyon sayesinde verileri daha iyi anlaşılır ve veri analizi daha doğru ve verimli bir şekilde gerçekleştirilir. Normalizasyon yöntemleri, sayısal sütunlara min-max normalizasyonu, z-skor normalizasyonu gibi matematiksel dönüşümler kullanılarak uygulanır [11].

Veri kümesindeki aykırı değerler modeli tespit etmeyi zorlaştırabilir. Veri kümesindeki öznitelikler farklı ölçeklerde ise verilerin normalize edilip daha iyi bir dağılıma uygun hale getirilmeleri sağlanmalıdır [12].

Veri dönüşümünü sağlayan normalizasyon yöntemleri makine öğrenmesi algoritmalarının doğruluğunu ve verimliliğini arttırır. Eğer analiz edilecek veri kümesi normalize edilmişse, yani [0.0, 1.0] gibi belirli aralıklara ölçeklendirildi ise, bu tür yöntemler daha iyi sonuç verebilir. Min-max normalizasyonu, orijinal veriler üzerinde bir doğrusal dönüşüm gerçekleştirir.

Z-skor normalizasyonunda, belirli özniteliğe ait ortalama ve standart sapmaya dayanarak normalleştirilir. Bu normalizasyon yöntemi o özniteliğin en küçük ve en değerlerinin bilinmediği, sadece ortalama ve standart sapmasının bilindiği durumlarda kullanılır.

2.1.1 Min – Max Normalleştirmesi

Veri kümesinde bulunan nümerik öznitelikler için, genellikle en büyük değere sahip gözlemin 1'e, en küçük gözlemin ise 0'a eşit olacak şekilde lineer doğru denkleminin elde edilip, yeni değerlerin belirlenen lineer doğru denklemi ile hesaplanması yöntemidir. Min - Max normalleştirmesine örnek olarak nümerik bir özniteliğin $X = \{10, -2, -7, 15, 3, 0, 14, 11, 1{,}2{,}7\}$ olduğunu varsayalım. Bu durumda en büyük değer olan 15, kurulacak lineer denklemde 1'e, en küçük değer olan -7 ise 0'a karşılık gelecektir. $\{15, 1\}\{-7, 0\}$ noktaları kullanılarak oluşturulacak lineer denklemin eğimi $m = {}^{1}/_{22}$ olacaktır. Veri kümesinin [0-1] aralığına normalleşmesini sağlayacak denklem $y' = \frac{1}{22}(x + 7)$ olacaktır. y' değeri normalleştirme sonucu elde edilen yeni değerleri göstermektedir. Örnek olarak elde edilen lineer denklem sonucunda 10 değerinin yeni değeri, $y' = \frac{1}{22}(10 + 7)$ işlemi sonucunda 0.7727 olacaktır. Normalleştirme sonucu hesaplanan yeni değerler aşağıdaki gibidir (Çizelge 2.1).

Çizelge 2.1 Min- max normalizasyonu örneği

Gerçek Değer(x)	Normalleştirme Sonucu Elde Edilen Değer($y^{'}$)
10	0.7727
-2	0.2272
-7	0
15	1
3	0.4545
0	0.3181
14	0.9545
11	0.8181
1	0.3636
2	0.4090
7	0.6363

Min – Max normalleştirmesini R dili aracılığıyla ***BBmisc*** kütüphanesinde bulunan ***normalize()*** fonksiyonu aracılığıyla aşağıdaki gösterildiği gibi kullanılması sonucu istenilen sonuçlar elde edilmiş olunur:

normalize(x, method = "range", range = c(0, 1), margin = 1L)

x parametresi dönüşümün uygulanacağı nümerik değişkeni göstermektedir.

method parametresi hangi tip normalleştirmenin uygulanacağını göstermektedir.

range parametresi özniteliğinin hangi değer aralığına indirgeneceğini göstermektedir.

margin parametresi satır veya sütun bazında işlem yapılacağını göstermektedir [13].

2.1.2 Z- Skor Normalleştirmesi

Z skor normalleştirmesi yöntemi, normal dağılan veriler için kullanılan z değerinin hesaplanması ile aynıdır. İlgili özniteliğe ait aritmetik ortalama veya standart sapma değerleri hesaplandıktan sonra, yeni değeri elde etmek için, ilgili özniteliğe ait değerden, özniteliğin aritmetik ortalaması çıkartıldıktan sonra özniteliğin standart sapmasına bölünmesi ile yeni değer elde edilir [14].

$$x^{'} = \frac{x_i - \mu_i}{\sigma_i} \tag{2.1}$$

$x^{'} = Z\ skor\ yöntemi\ ile\ normalleştirilmiş\ veriyi$,

$x_i = Özniteliğin\ i.değerine\ ait\ nümerik\ değeri,$

$\mu_i = Özniteliğin\ aritmetik\ ortalamasını,$

$\sigma_i = Özniteliğin\ standart\ sapmasını\ ifade\ etmektedir.$

Bir önceki başlıkta normalleştirdiğimiz örnek veri kümesini Z skor yöntemi ile normalleştirdiğimizde, belirtilen $X = \{10, -2, -7, 15, 3, 0, 14, 11, 1{,}2{,}7\}$ veri kümesi için sonuçlar Çizelge 2.2 de gösterilmiştir.

Çizelge 2.2 Z- skor normalizasyonu yeni değerleri

Gerçek Değer(x)	Normalleştirme Sonucu Elde Edilen Değer($y^{'}$)
10	0.7251
-2	-0.9840
-7	-1.6962
15	1.4372
3	-0.2719
0	-0.6992

Çizelge 2.2 Z- skor normalizasyonu yeni değerleri(devamı)

14	1.2948
11	0.8675
1	-0.5567
2	-0.4143
7	0.2978

Z- Skor normalleştirmesini R dili aracılığıyla ***base*** kütüphanesinde bulunan ***scale()*** fonksiyonu aracılığıyla aşağıdaki gösterildiği gibi kullanılması sonucu istenilen sonuçlar elde edilmiş olunur:

scale(x, center = TRUE, scale = TRUE)

x parametresi dönüşümün uygulanacağı nümerik değişkeni göstermektedir.

center = TRUE parametresi öznitelikten aritmetik ortalamanın çıkartılacağını göstermektedir.

scale = TRUE parametresi öznitelikten aritmetik ortalamanın çıkartılmasından sonra standart sapmaya bölüneceğini göstermektedir.

2.2 Aykırı Değer Analizi

Aykırı değerler, veri kümesinin genel özelliklerinden beklenenden farklı değere sahip gözlemlerdir. Genellikle, veri kümesindeki aykırı olmayan gözlemlere kıyasla farklı bir süreç sonucunda üretilmişlerdir. Sebep ne olursa olsun, istatistiksel aykırı değerler bir deneyin sonuçlarını derinden etkileyebilir ve benzer popülasyonların farklı ya da farklı popülasyonların benzer görünmesini sağlar [15].

Bir veri kümesinin geri kalanıyla veya bir alt kümesi ile tutarsız olan ve veri kümesinden belirgin şekilde sapan değerlere aykırı değer denir. Aykırı değerler, her uygulama alanına ait veri kümelerinde bulunabilir. Kredi kartı sahtekarlığının tespiti, klinik deneyler, veri temizleme, siber saldırılar, hava durumu tahmini, coğrafi bilgi sistemleri ve diğer veri madenciliği uygulamalarında aykırı değerlerin analiz edilmesi önerilmektedir. Aykırı değerler beklenmedik bilginin ortaya çıkmasını sağlamaktadır.

Aykırı değerlerin ortaya çıkma sebepleri arasında, veri kümesinin doğal değişkenliği, yapılabilecek ölçüm hatası veya kullanıcılar tarafından yapılan kayıt altına alma sırasında oluşabilecek hatalar şeklinde sıralanabilir. Aykırı değerler ile alakalı bir diğer problem ise tespit edilen aykırı değerlere ne şekilde bir işlem uygulanacağıdır. Buradaki yaklaşımlar, genellikle aykırı değerlerin veri kümesinden çıkartılması veya aykırı değerlerin yerine ortalama veya medyan değerinin yansıtılmasıdır [16].

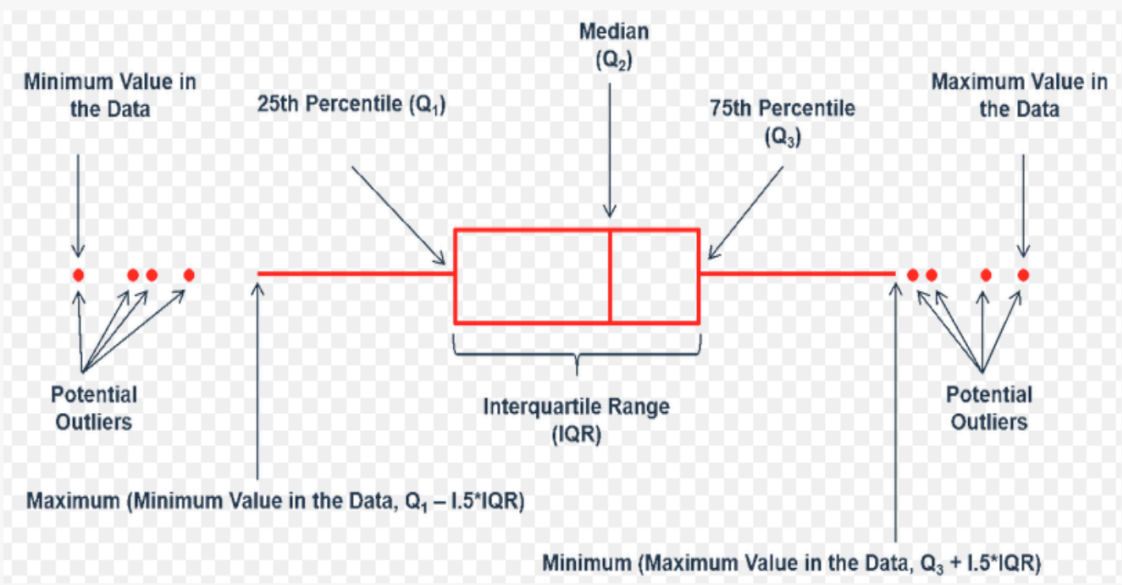
Aykırı değerler, veri kümesinin bütününü bozabileceği gibi, oldukça değerli bir bilgi değeri de taşır. Örneğin kullanıcıların kredi kartı harcamaları dikkate alındığında, kullanıcının önceki harcamalarına kıyasla çok yüksek bir harcama yapılırsa, buradaki oluşacak aykırı değer, bir dolandırıcılığı önleyecek özelliğe sahiptir.

2.2.1 Box - Plot Yöntemi

Sürekli değişkenler için uygulanabilen bu yöntemde, aykırı değerlerin tespit edileceği öznitelik belirlendikten sonra aşağıdaki adımlar takip edilir:

1.İlgili özniteliğe ait kartil değerlerini hesapla
2.3.kartil ve 1. kartil arasındaki farkı hesapla
3.2.adımda hesaplanan değeri 1.5 ile çarp
4.1. Kartil değerinden 3. adımda hesaplanan değeri çıkar ve alt sınır belirle
5.3.Kartil değerine 3. adımda hesaplanan değeri ekle ve üst sınır belirle
6.4. ve 5. adımda belirlenen sınırlar arasında kalmayan değerler aykırı değerdir [17].

Adımlar ile belirtilen kutu grafiği Şekil 2.1 de verilmiştir.

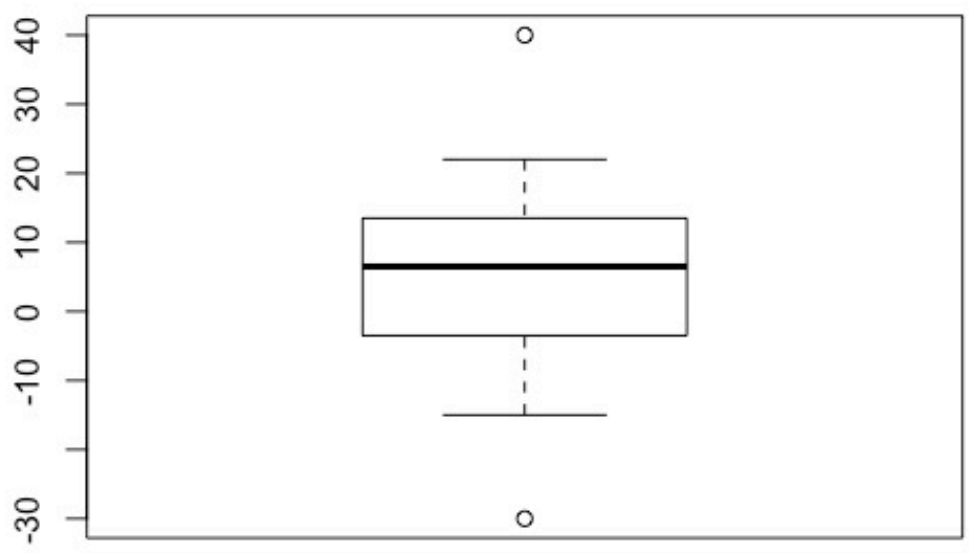


Şekil 2.1 Box – plot yöntemi

Bir örnek ile açıklamak istersek, bir özniteliğe ait değerlerin X ile verildiğini kabul edelim.

$X = \{-30, -15, -7, 0, 1, 3, 10, 11, 12, 15, 22, 40\}$ şeklinde olsun. X özniteliğine ait kartil değerleri, $Q_1 = -1.75, Q_2 = 6.50, Q_3 = 12.75$ olarak hesaplanır. Q_3- Q_1 $= 14.50$ olarak hesaplanır. Daha sonra bulunan bu değer 1.5 ile çarpılıp 21.75 değeri elde edilir. Q_1-21.75 $= -23.50$ olacak şekilde hesaplandıktan sonra alt sınır belirlenir. $Q_3 + 21.75 = 34.50$ değeri hesaplanarak üst sınır değeri hesaplanır. Bu sınırlar dışında kalan -30 ve 40 değerleri aykırı değerlerdir.

Yukarıda verilen örneğin kutu grafiği Şekil 2.2 deki gibidir.



Şekil2.2 Box – plot sonuç grafiği

2.2.2 Ortalama ve Standart Sapma Kullanılarak Aykırı Değer Analizi

Sürekli değişkenler için uygulanabilen bu yöntemde, aykırı değerlerin tespit edileceği öznitelik belirlendikten sonra aşağıdaki adımlar takip edilir:

1. İlgili özniteliğe ait ortalama ve standart sapma değerlerini hesapla
2. Ortalamadan standart sapmanın 2 katını çıkart ve alt sınırı belirle
3. Ortalamaya standart sapmanın 2 katını ekle ve üst sınırı belirle
4. Eğer değer 2. ve 3. adımda bulunan aralıklar arasında değilse aykırı değerdir.

$$Aykırı\ Değer\ \text{İçermeyen}\ Bölge = Ortalama \mp 2\ x\ Standart\ Sapma \quad (2.2)$$

Bir örnek ile açıklamak istersek, bir özniteliğe ait değerlerin X ile verildiğini kabul edelim.

$X = \{-30, -15, -7, 0, 1, 3, 10, 11, 12, 15, 22, 40\}$ şeklinde olsun. Verilen örneğin ortalaması 5.16, standart sapma değeri 17.93 tür. Bu durumda aykırı değer içermeyen bölgenin alt sınırı, -30.70 üst sınırı ise 41.03 olacaktır. Bir önceki yöntemde -30 ve 40 değerleri aykırı değerler iken, bu yönteme göre veri kümesinde aykırı gözlem bulunmamaktadır.

R dili aracılığıyla yazılan kullanıcı tanımlı fonksiyon ile bu yönteme göre aykırı değerlerin tespit edilmesi mümkündür.

```
AykiriDeger <- function(x){
 ortalama = mean(x)
 standartSapma = sd(x)
 altSinir <- ortalama - 2 * standart_sapma
 ustSinir <- ortalama + 2 * standart_sapma

 for (i in 1:length(x)) {
  if(x[i] >= altSinir & x[i] <= ustSinir)
  {
   cat(paste(x[i], "aykiri deger degildir", collapse=" "), "\n")
  }
  else
  {
   cat(paste(x[i], "aykiri degerdir", collapse=" "), "\n")
  }

 }
 }
```

2.3 Uzaklık Ölçütleri

Makine öğrenmesi algoritmalarının sonuçlarını etkileyen temel unsurlardan biri de hangi uzaklık ölçü biriminin kullanıldığıdır. Seçilecek uzaklık ölçü biriminin genellikle sonucu değiştirmeye etkisi vardır. Sayısal veriler için uzaklık ölçüsü birimlerinin hesaplanması, kategorik değişkenler ile kıyaslandığında daha kolaydır.

İki veri arasındaki uzaklığı veya farklılığı ölçmek, veri madenciliği ve makine öğrenmesi alanlarından olan kümeleme, sınıflandırma, tavsiye sistemleri ve aykırı değer analizi konularında kritik öneme sahiptir. Uzaklık hesabı bu algoritmalar için bir ön işlem niteliğinde olduğundan istenilen uzaklık birimi seçilebilir. Seçilecek uzaklık biriminin modelin performansı üzerinde önemli bir etkiye sahiptir [18].

Tamamen sayısal değerlerden oluşan veri kümeleri için, uzaklık hesabı kategorik veriler içeren veri kümelerine kıyasla uygulanması daha kolaydır. Bu başlık altında, en çok kullanılan uzaklık birimlerinden olan Öklid, Manhattan, Canberra, Maksimum Değer Uzaklığı, Minkowski ve Mahalanobis uzaklıkları ele alınmıştır.

Aşağıdaki tabloda tüm uzaklık ölçülerinin formülleri ve R dili aracılığıyla nasıl hesaplandığı gösterilmiştir (Çizelge 2.3). **Stats** paketi içinde bulunan dist() ve mahalanobis() fonksiyonları kullanılmıştır.

S, Mahalanobis uzaklığı hesaplamada kullanılan kovaryans matrisini göstermektedir.

Çizelge 2.3 Uzaklık ölçütleri

Uzaklık Ölçüsü	Formül	R Dilinde Hesaplanması
Öklid Uzaklığı	$d = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}$	dist(rbind(A, B), method = "euclidean")
Manhattan Uzaklığı	$d = \sum_{i=1}^{n} \lvert A_i - B_i \rvert$	dist(rbind(A, B), method = "manhattan")
Canberra Uzaklığı	$d = \sum_{i=1}^{n} \frac{\lvert A_i - B_i \rvert}{\lvert A_i \rvert + \lvert B_i \rvert}$	dist(rbind(A, B), method = "canberra")
Maksimum Değer Uzaklığı	$d = max \left\{ \begin{matrix} \lvert A_1 - B_2 \rvert, \\ \lvert A_2 - B_2 \rvert, \\ \dots, \\ \lvert A_n - B_n \rvert \end{matrix} \right\}$	dist(rbind(A, B), method = "maximum")
Minkowski Uzaklığı	$d = \left[\sum_{i=1}^{n} (\lvert A_i - B_i \rvert)^p \right]^{1/p}$	dist(rbind(A, B), method = "minkowski", p = 3)

Çizelge 2.3 Uzaklık ölçütleri(devamı)

Mahalanobis Uzaklığı	$d = \sqrt{(A\text{-}B)^T S^{-1} (A\text{-}B)}$	mahalanobis(A, B, S)

BÖLÜM 3

DOĞRU KÜME SAYISININ BELİRLENMESİ

Kümeleme algoritmaları; benzer karakteristik özelliklere sahip verilerin, birbirinden ayrılarak ilgili kümeler aracılığıyla veriyi temsil etmesini sağlayan ve bu ayrılma ile verilerin kümelenmiş halde iken daha değerli olarak değerlendirilip kullanmasına olanak tanıyan bir süreçtir [19]. Kümeleme algoritmalarının kullanım alanı oldukça geniştir. Bu alanlardan bazıları ise; Sağlık, E-ticaret, Eğitim ve Bankacılık – Finans sektörleridir.

K- ortalamalar algoritması en popüler kümeleme algoritmalarının başında gelir. Bu algoritmanın uygulanması için k değerinin algoritma uygulanmadan önce belirlenmesi gerekmektedir. Belirli bir veri kümesi için uygun sayıda kümenin bulunması genellikle "doğru" kümelenmeyi neyin oluşturduğu kararının verilebilmesi için doğru k değerinin belirlemeye yardımcı algoritmaları kullanarak gerçekleştirilen bir süreçtir [20].

Bir kümeleme algoritmasının performansı ve başarısı, seçilen k değerinden etkilenebilir. Bu nedenle, önceden tanımlanmış bir k kullanmak yerine, belirli bir aralık için tüm k değerleri ile kümeleme yapılıp sonuçlarının kıyaslanması gerekir. Veri kümesinin belirgin özelliklerini yansıtabilmesi için bahsedilen k aralığının yüksek tutulması önemlidir. Doğru k değerini belirlemek için seçilecek aralığı belirlerken seçilen değerler kümeleme işleminin değerlendirilebilmesi için veri kümesindeki kayıtların sayısından önemli ölçüde daha küçük olmalıdır.

3.1 Elbow (Dirsek) Yöntemi

Dirsek Yöntemi, kümelerin sayısının bir fonksiyonu olarak açıklanan varyans yüzdesine bakılan bir yöntemdir. Bu yöntem, bir dizi kümeyi seçmesi fikrine dayanmaktadır, böylece başka bir kümenin eklenmesi, verilerin daha iyi modellenmesini sağlamaz.

Kümeler tarafından açıklanan varyans yüzdesi sayı kümelerine karşı çizilir. İlk kümeler diğer kümelere istinaden daha fazla bilgi katacak olmalarına rağmen bir noktada marjinal kazanç dramatik bir şekilde düşecek ve grafikte bir açı verecektir. Doğru "k" değeri yani küme sayısı bu noktada seçilir. Yukarıdaki bahsedilenler grafiğe döküldüğünde bir dirsek şekline benzeyecektir [21].

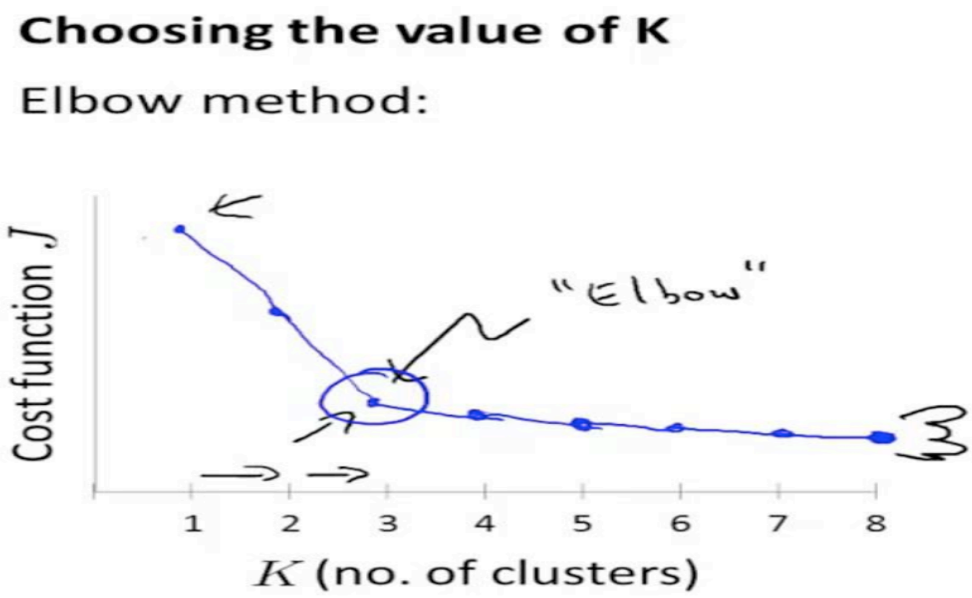
k = 2 değeri ile başlamak ve her adımda k değerini 1 arttırmaya devam etmek ve kümelerin ve çalıştırma ile birlikte gelen hatanın hesaplanmasına yardımcı olur. K için bazı değerlerde hata önemli ölçüde düşer ve bu adımdan daha fazla arttırdığımızda grafik yatay düzleme paralele yakın hale gelecektir. Bundan önceki adım olan paralelliğe geçmeden önceki k değeri, aranan k değeridir [22].

Dirsek yönteminin çalışma prensibi sonucu, bulunan k değerinden sonra kümelerin sayısının arttırılması durumunda, yeni kümelenme mevcut olanın bir kısmına çok yakındır.

Aşağıdaki Şekil 3.1 de, bükülme k = 1 den k=2 ye ve k=2den k=3 e kadar hızla düşer ve daha sonra grafik k=3 bükülme noktasına ulaşır ve bundan sonra bükülme çok daha yavaş bir şekilde ilerler.

Şekil 3.1' e bakarak doğru k değerinin 3 olduğunu söyleyebiliriz.

Bükülme k=3 noktasına kadar hızla azalırken, bu veri kümesi için gerekli olan küme sayısı k=3 değerinden sonrası için çok yavaş bir şekilde azalır.



Şekil 3.1 Elbow yöntemi açıklaması

Dirsek yöntemi algoritmasının K-ortalamalar Algoritmasına uygulanması aşağıdaki sıra ile gerçekleşir:

1.k = 1 ile başlat.

2.Algoritmayı çalıştır.

3.Hata terimini hesapla.

4.k değerini arttır.

5.k değeri ve hata terimine göre grafiği çiz.

6.Eğer grafikte bir noktada fonksiyon şiddetli düşüş yaşar ve dirsek şekline benzerse bu noktadaki k değerini doğru k değeri olarak seç.

7.Algoritmayı sonlandır [23] [39].

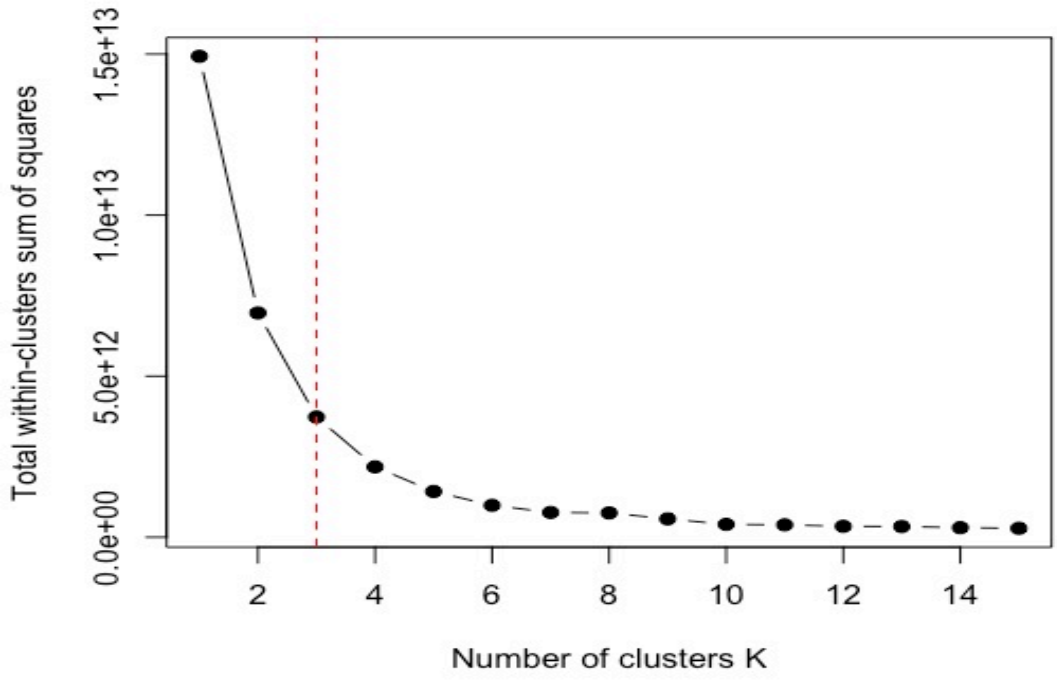
Dirsek Yönteminin veri kümesine uygulanarak doğru k değerlerinin seçilmesi için aynı veri kümesi üzerinde 3 farklı veri ön hazırlık aşaması uygulanmıştır.

1.Herhangi bir önişlem uygulanmamış veri kümesi

2.Min – Max normalizasyonu uygulanarak indirgenmiş veri kümesi

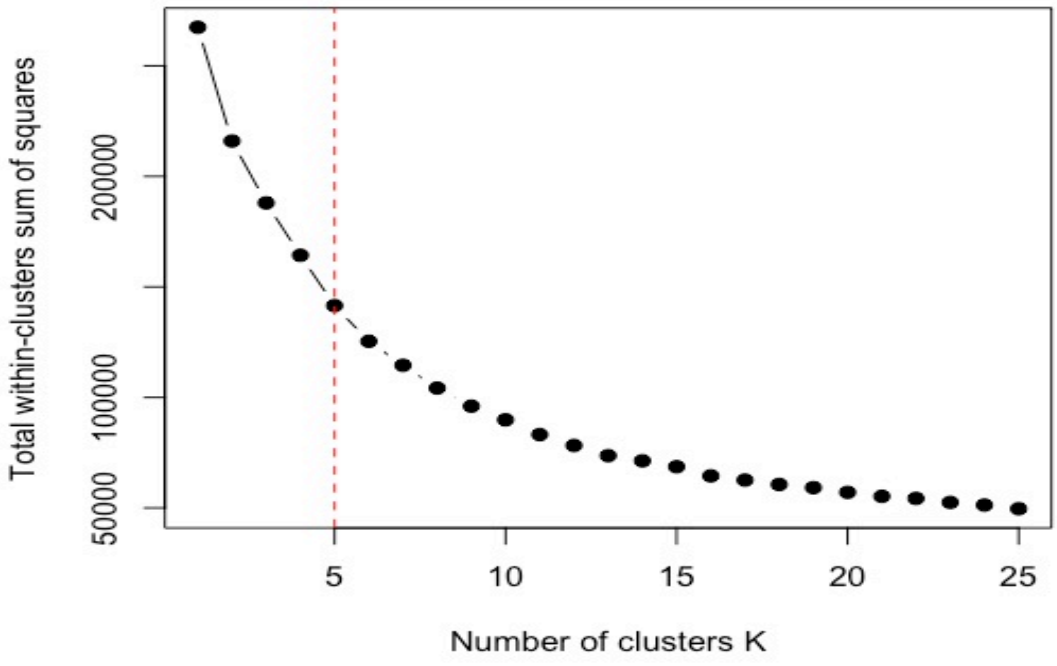
3.Z score yöntemi ile indirgenmiş veri kümesi

Aşağıdaki görselde (Şekil 3.2) 1. durum için ortaya çıkan dirsek yöntemi görseli incelendiğinde doğru k değerinin 3 olduğu anlaşılmaktadır.



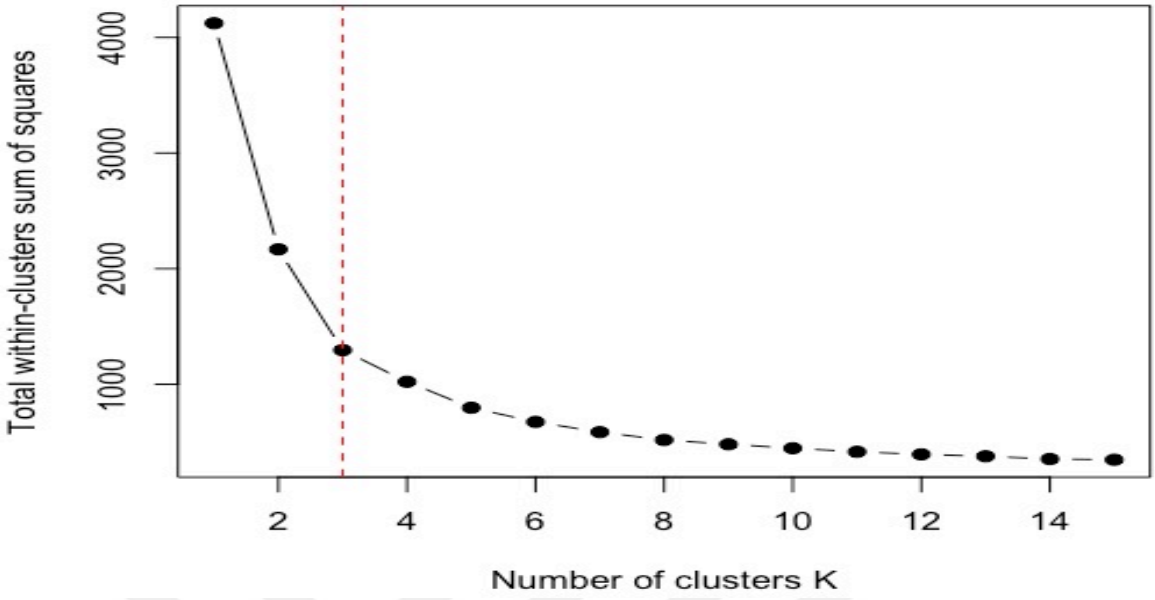
Şekil 3.2 Önişlem uygulanmamış veri kümesi için elbow yöntemi grafiği

Aşağıdaki görselde (Şekil 3.3) 2. durum için ortaya çıkan dirsek yöntemi görseli incelendiğinde doğru k değerinin 5 olduğu anlaşılmaktadır.



Şekil 3.3 Min – max normalizasyonu uygulanan veri kümesi için elbow yöntemi grafiği

Aşağıdaki görselde (Şekil 3.4) 3. durum için ortaya çıkan dirsek yöntemi görseli incelendiğinde doğru k değerinin 3 olduğu anlaşılmaktadır.



Şekil 3.4 Z- skor normalizasyonu uygulanan veri kümesi için elbow yöntemi grafiği

3.2 Gap İstatistik Değeri

Veri kümesi $\{x_{ij}\}, i = 1,2,\dots,n\,, j = 1,2,\dots,p,$ n gözlem sayısı, p öznitelik sayısı olacak şekilde tanımlanır. $d_{ii'}$ değeri de, i ve i' noktalarının öklid uzaklığının karesi olarak tanımlanır.

Veri kümesini k kümeye ayırdığımız varsayalım. $C_1, C_2, \dots, C_k$ olacak şekilde ve $r \leq k$ şeklinde alınan r değeri için C_r bu indisteki gözlem sayısı $n_r = |C_r|$ olduğunu varsayalım. $d_{ii'}$ değerinin formülü aşağıda belirtilmiştir [24]:

$$d_{ii'} = (x_{i1}\text{-}x_{i'1})^2 + (x_{i1}\text{-}x_{i'1})^2 + \cdots + (x_{ip}\text{-}x_{i'p})^2 \tag{3.1}$$

D_r , $r.$ kümenin elemanlarının d cinsinden toplamıdır:

$$D_r = \sum_{i,i' \in C_r} d_{ii'} \tag{3.2}$$

W_k ağırlık değeri r. kümenin küme içi uzaklıkları toplamının ortalamalarının yarısıdır. Formülü aşağıdaki gibi verilmiştir:

$$w_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r \tag{3.3}$$

Gap istatistik katsayısının formülü aşağıda verilmiştir:

$$Gap_n(k) = E_n^*\{\log(W_k)\} - \log(W_k) \tag{3.4}$$

Kümeleme seçilecek k sayısını bulmak için kullanılacak formül aşağıda verilmiştir:

$$k = en\ küçük\ k \mid Gap(k) \geq (Gap(k+1) - s_{k+1}) \tag{3.5}$$

Gap istatistiğinin hesaplanması adımları şu şekildedir.

Adım 1: Veri kümesi belirlenen üst sınır değerine kadar kümelenir. K değeri belirlenen üst sınır değerdir[25].

W_k $buradaki\ k = 1, 2, \ldots, K$

Adım 2: B adet uniform referans veri kümesi üretilir.

$W_{kb}^*, b = 1, 2, \ldots, B$, $k = 1, 2, \ldots, K$ olacak şekilde aşağıdaki formül ile tahmini gap istatistik değeri hesaplanır.

$$Gap(k) = (1/B) \sum_b \left(\log\left(W_{kb}^*\right) - \log(W_k)\right) \tag{3.6}$$

Adım 3: $\bar{l} = (1/B) \sum_b \log(W_{kb})$ olarak alınıp standart sapma hesaplanır.

$$sd_k = (1/B) \sum_b \left\{log\left(W_{kb}^*\right) - \bar{l}\right\}^{2\,1/2} \tag{3.7}$$

ve $s_k = sd_k \sqrt{1 + 1/B}$ olsun. Son olarak küme sayısını seçmek için aşağıdaki formül uygulanır.

$$\hat{k} = en\ k\text{üçü}k\ k \mid Gap(k)\ \geq (Gap(k+1) - s_{k+1}) \quad (3.8)$$

$\hat{k}$ değeri, veri kümesinin kaç kümeye ayrılması gerektiğini göstermektedir.

3.3 R Dili Aracılığıyla Gap İstatistik Değerinin Hesaplanması

R dili aracılığıyla, *cluster* kütüphanesi altında bulunan clusGap fonksiyonu yardımı ile hesaplanabilir. Örnek kullanımı aşağıdaki gibidir.

library("cluster")

clusGap(data, FUN = , nstart =, K.max =, B =)

İlk parametrede kümeleme uygulanacak veri kümesinin belirtilmesi gerekmektedir.

FUN parametresinde hangi kümeleme yönteminin kullanılacağının belirtilmesi gerekmektedir.

nstart parametresi uygulanacak kümeleme algoritmasında kaç farklı başlangıç değeri seçileceğini belirtmektedir.

K.max parametresi en fazla seçilecek K değerini göstermektedir.

$\boldsymbol{B}$ parametresi kaç adet uniform veri kümesi üretileceğini göstermektedir[26].

3 farklı veri kümesi için Gap istatistiği sonuçları aşağıdaki gibidir.

- İlk durum olan ön işlem uygulanmamış veri kümesi için Gap istatistiği sonuçları Çizelge 3.1 de verilmiştir.

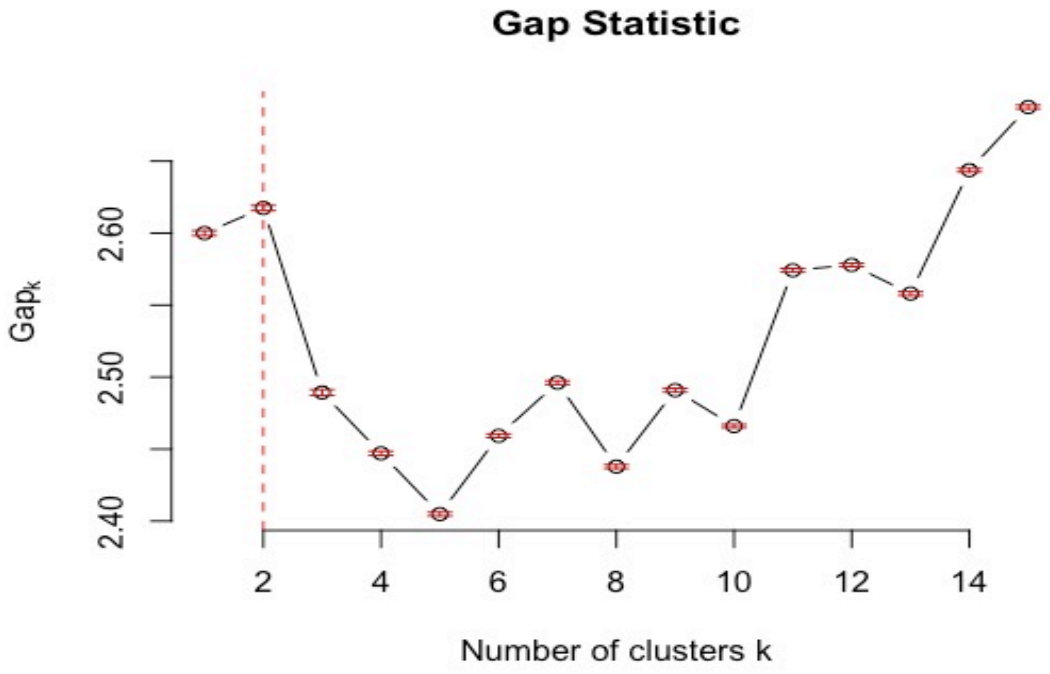
Çizelge 3.1 Önişlem uygulanmamış veri kümesi gap sonuçları

k değerleri	$\log(W_k)$	$E_n^*\{\log(W_k)\}$	s_{k+1}	$Gap(k)$
2	18.34718	20.96462	0.001784429	2.617442
3	18.09940	20.58870	0.002053424	2.489300
4	17.88872	20.33577	0.001455019	2.447046
5	17.74493	20.14980	0.001310191	2.404869

Çizelge 3.1 Önişlem uygulanmamış veri kümesi gap sonuçları(devamı)

6	17.54818	20.00739	0.001283584	2.459213
7	17.39755	19.89368	0.001318532	2.496132
8	17.36395	19.80180	0.001365975	2.437852
9	17.23399	19.72490	0.001347888	2.490905
10	17.19508	19.66105	0.001156714	2.465973
11	17.03233	19.60650	0.001226769	2.574168
12	16.98171	19.55959	0.001220068	2.577883
13	16.96124	19.51925	0.001386740	2.558013
14	16.83987	19.48355	0.001227263	2.643679
15	16.76523	19.45277	0.001330285	2.687539

Tabloda verilen sonuçların grafiği Şekil 3.5 de çizilmiştir.



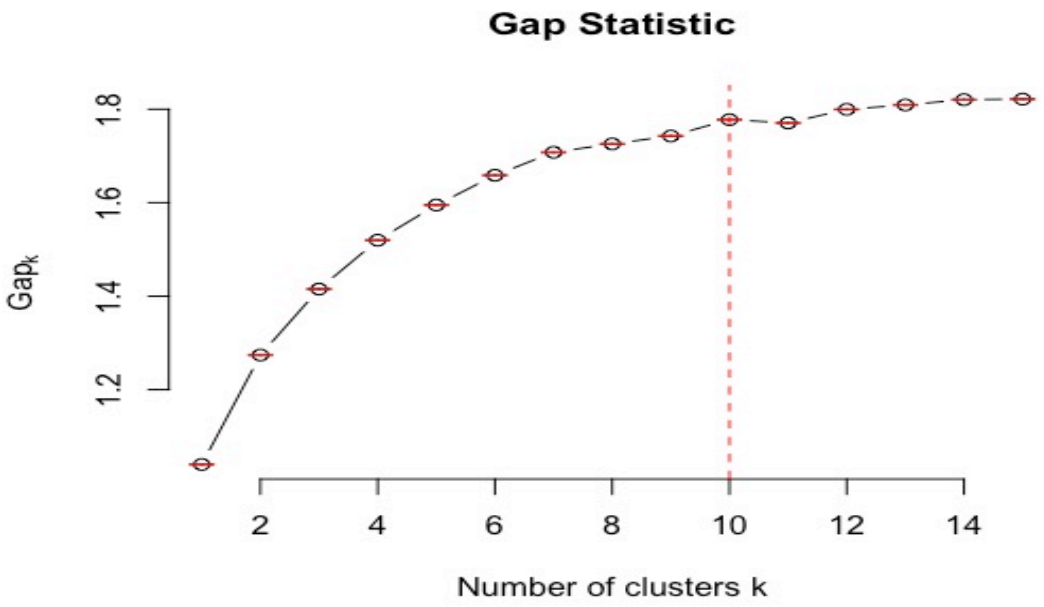
Şekil 3.5 Önişlem uygulanmamış veri kümesi için gap grafiği

- İkinci durum olan Min – Max normalizasyonu uygulanan veri kümesi için sonuçlar Çizelge 3.2 de verilmiştir.

Çizelge 3.2 Min – max normalizasyonu uygulanan veri kümesi için gap sonuçları

k değerleri	$\log(W_k)$	$E_n^*\{\log(W_k)\}$	s_{k+1}	$Gap(k)$
2	8.052164	9.326232	0.0009777676	1.274069
3	7.851997	9.267196	0.0010810789	1.415199
4	7.695051	9.214723	0.0008759583	1.519672
5	7.580239	9.175175	0.0008733676	1.594936
6	7.482339	9.140927	0.0008823177	1.658588
7	7.399324	9.107104	0.0010986491	1.707780
8	7.344768	9.070502	0.0008775755	1.725734
9	7.302254	9.044905	0.0010040728	1.742651
10	7.222122	8.999745	0.0009122253	1.777622
11	7.248949	9.019546	0.0008498276	1.770597
12	7.179430	8.979070	0.0008221472	1.799640
13	7.149448	8.958545	0.0009592573	1.809097
14	7.116731	8.937148	0.0009185710	1.820417
15	7.093160	8.914647	0.0010041580	1.821488

Tabloda verilen sonuçların grafiği aşağıdaki gibidir (Şekil 3.6).



Şekil 3.6 Min- max normalizasyonu uygulanmış veri kümesi için gap grafiği

- Üçüncü durum olan Z- skor yöntemi uygulanan veri kümesi için sonuçlar Çizelge 3.3 de verilmiştir.

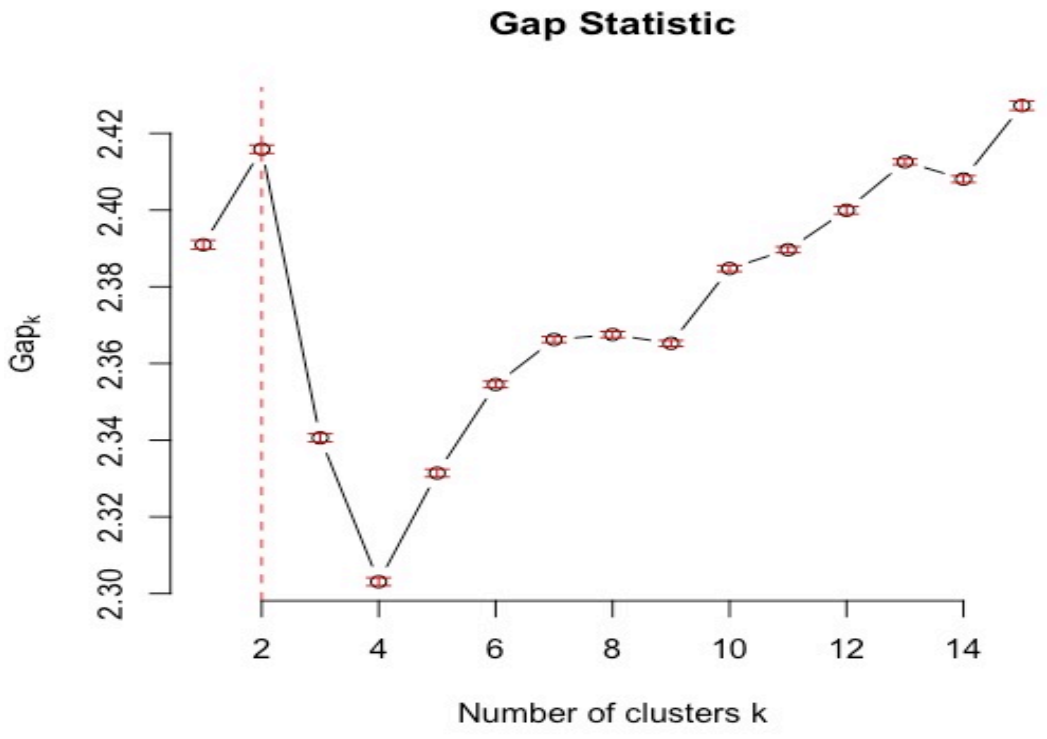
Çizelge 3.3 Z-skor normalizasyonu uygulanan veri kümesi için gap sonuçları

k değerleri	$\log(W_k)$	$E_n^*\{\log(W_k)\}$	s_{k+1}	$Gap(k)$
2	10.331159	12.72212	0.0010958631	2.415874
3	10.278249	12.61885	0.0010542370	2.340602
4	10.231419	12.53449	0.0010363844	2.303074
5	10.155257	12.48667	0.0009951614	2.331411
6	10.085631	12.44015	0.0008447852	2.354524
7	10.038183	12.40442	0.0007757365	2.366241
8	10.000973	12.36851	0.0008051943	2.367537
9	9.979842	12.34509	0.0007627627	2.365244
10	9.937564	12.32232	0.0007933022	2.384760

Çizelge 3.3 Z-skor normalizasyonu uygulanan veri kümesi için gap sonuçları(devamı)

11	9.914400	12.30406	0.0007855680	2.389659
12	9.886277	12.28623	0.0009793675	2.399954
13	9.858441	12.27104	0.0007336939	2.412596
14	9.848398	12.25649	0.0008783632	2.408091
15	9.815283	12.24253	0.0012157915	2.427244

Tabloda verilen sonuçların grafiği aşağıdaki gibidir (Şekil 3.7).



Şekil 3.7 Z-skor normalizasyonu uygulanmış veri kümesi için gap grafiği

BÖLÜM 4

KÜMELEME ALGORİTMALARI

4.1 K- Ortalamalar Algoritması

En yaygın kullanılan denetimsiz öğrenme yöntemlerinden biridir. K-ortalamalar algoritması her gözlemin sadece bir kümeye ait olabilmesine izin verir. K-ortalamalar algoritmasının genel mantığı n adet gözlemden oluşan bir veri kümesini, algoritma uygulanmaya başlamadan önce belirlenen k adet kümeye ayırmaktır. Amaç, algoritma ile elde edilen kümeleme işlemi sonunda elde edilen kümelerin, küme içi benzerliklerinin maksimum ve kümeler arası benzerliklerinin minimum olmasını sağlamaktır [19].

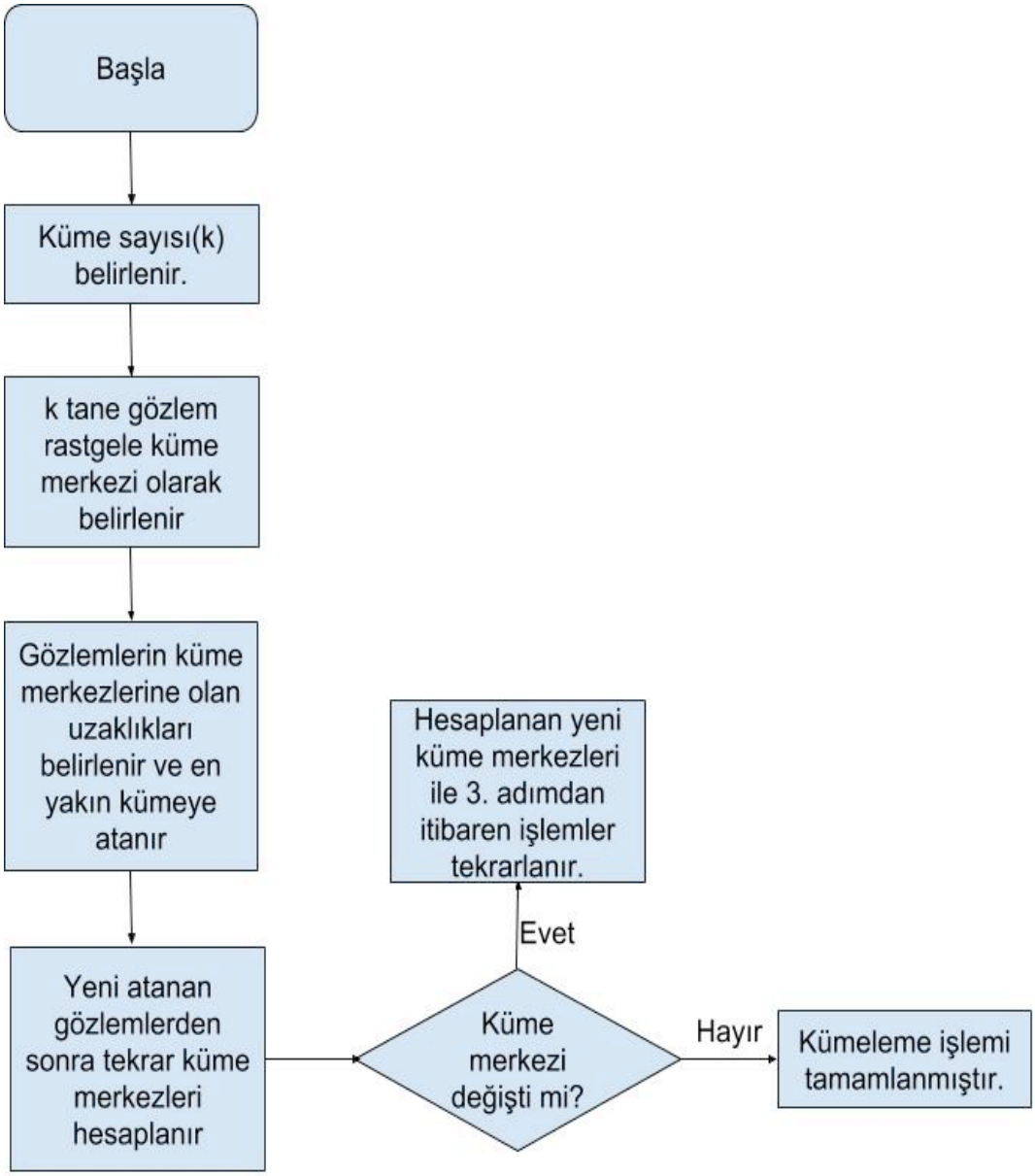
K-Ortalamalar algoritması ilk olarak rastgele seçilen k adet merkez noktayla başlar. Veri kümesindeki her nokta kendisine en yakın küme merkezine ait olan kümeye dahil edilir. Küme merkezlerinin değeri o kümede bulunan noktaların aritmetik ortalaması alınarak hesaplanır. Bu işlem merkezlerin değerleri değişmeyinceye kadar devam eder [27].

K-Ortalamalar algoritmasının gerçekleştirilmesi aşağıdaki 5 adımı takip ederek elde edilir [28]:

1. Önceden belirlenmiş olan küme sayısı olan k değeri seçilir.
2. Veri kümesinden k adet gözlem rastgele seçilir ve küme merkezleri olarak kabul edilir.
3. Kalan gözlemlerin en yakın olduğu küme merkezleri belirlenir ve ilgili gözlem o kümeye atanır.
4. Her küme için atanan gözlemler ile küme merkezi hesaplanır.

5. Eğer kümelerin yeni merkez noktaları bir önceki merkez noktaları ile aynı ise kümeleme tamamlanır. Değil ise hesaplanan yeni küme merkezleri ile 3. adımdan itibaren işlemler tekrarlanır.

K- Ortalamalar algoritmasının akış diyagramı Şekil 4.1 de verilmiştir.



Şekil 4.1 K-Ortalamalar algoritması

K-Ortalamalar algoritmasının nasıl hesaplandığını göstermek için aşağıdaki tabloda belirtilen veri kümesi kullanılmıştır (Çizelge 4.1).

Çizelge 4.1 K-ortalamalar algoritması örneği

Gözlem Numarası	1.Öznitelik	2.Öznitelik
1	3.00	3.50
2	3.20	3.30
3	3.50	3.80
4	7.00	7.01
5	7.10	6.70
6	7.15	7.30
7	7.75	7.20
8	4.00	8.00

1. Adım

Yukarıdaki tabloda verilmiş örnek veri kümesini 2 kümeye ayırmak için başlangıç küme merkezleri olarak 2 ve 3. gözlemleri seçelim. Bu durumda $\mu_1 = (3.50, 3.80)$, $\mu_2 = (7{,}00, 7.01)$ olur (Çizelge 4.2).

Çizelge 4.2 K-ortalamalar algoritması küme merkezleri

Küme	Gözlem Numarası	Küme Merkezi
1. Küme	3	$\mu_1: (3.50, 3.80)$
2. Küme	4	$\mu_2: (7.00, 7.01)$

Bütün gözlemler ve küme merkezleri arasındaki Öklid uzaklığı hesaplanır. Sonuçlar aşağıdaki tabloda gösterilmiştir (Çizelge 4.3).

Çizelge 4.3 K-ortalamalar algoritması örneği 2. adım

Gözlem Numarası	1. Merkeze uzaklık	2. Merkeze Uzaklık	Küme No
1	0.5830952	5.321663	1
2	0.5830952	5.310753	1
3	0.0000000	4.749116	1
4	4.749116	0.0000000	2
5	4.62277	0.325730	2
6	5.056926	0.326497	2
7	5.442656	0.773692	2
8	4.229657	3.159130	2

1. kümeye ait gözlemlerin aritmetik ortalaması tekrar hesaplanıp, yeni küme merkezi bulunmuştur.

$$\mu_1 = \left(\frac{(3.00 + 3.20 + 3.50\,, 3.50 + 3.30 + 3.80)}{3.00}\right) = (3.23,3.53\,)$$

$$\mu_2 = \left(\frac{(7.00 + 7.10 + 7.15 + 7.75 + 4.00, 7.01 + 6.70 + 7.30 + 7.20 + 8)}{5}\right)$$
$$= (6.60,7.24\,)$$

Yeni küme merkezleri, $\mu_1 = (3.233333,3.533333\,)$ ve $\mu_2 = (6.6,7.242\,)$ olarak hesaplanmıştır.

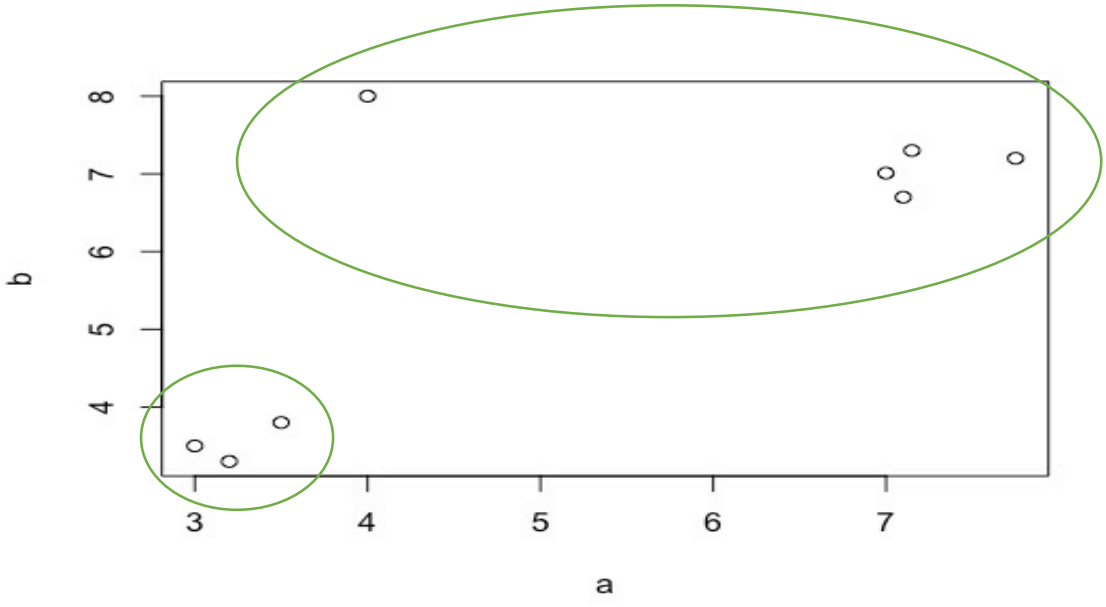
2. Adım

Hesaplanan yeni küme merkezleri ile gözlemlerin küme merkezine olan uzaklıkları aşağıdaki tabloda gösterilmiştir (Çizelge 4.4).

Çizelge 4.4 K-ortalamalar algoritması örneği 3. adım

Gözlem Numarası	1. Merkeze uzaklık	2. Merkeze Uzaklık	Küme No
1	0.23570	5.19255	1
2	0.23570	5.20570	1
3	0.37712	4.63221	1
4	5.12591	0.46241	2
5	4.99789	0.73740	2
6	5.43397	0.55305	2
7	5.81762	1.15077	2
8	4.53199	2.70824	2

1. adım ve 2.adımdaki küme durumları değişmediği için, algoritma sonlandırılır. Bu durumda 1, 2, 3 numaralı gözlemler 1. kümeye, 3,4,5,6,7,8 numaralı gözlemler 2. Küm eye ait olur (Şekil 4.2).



Şekil 4.2 Örnek kümeleme sonuçları

4.2 Hiyerarşik Kümeleme Algoritmaları

Hiyerarşik kümeleme, ayrıştırıcı ve birleştirici kümeleme yöntemleri olmak üzere 2 başlıkta incelenir. Ayrıştırıcı hiyerarşik kümeleme yönteminde başlangıçta tüm gözlemler tek bir kümeye ait olduğu kabul edilir ve ayrıştırma algoritmaları ile birden fazla kümeye ayrılır. Birleştirici hiyerarşik kümeleme yöntemlerinde ise ilk durumda her bir gözlemin bir küme olduğu kabul edilir be gözlemler aşağıda belirtilen algoritmalar ile birleştirilir [29].

Hiyerarşik kümelemede ilk olarak gözlemler arası uzaklıklar hesaplanır. Uzaklık matrisi elde edildikten sonra Tek Bağlantılı, Tam Bağlantılı hiyearşik kümeleme algoritmaları kullanılabilir.

4.2.1 Tek Bağlantılı (Single Linkage) Yöntemi

Bu kümeleme yönteminde ilk durumda her gözlem farklı bir kümeye ait olarak değerlendirilir. Daha sonra hesaplanan uzaklık matrisi yardımıyla her adımda en yakın iki gözlem birleştirilir.

Hiyerarşik kümelemenin tek bağlantı yöntemi uygulanırken aşağıdaki adımlar uygulanır. Sırasıyla uygulanacak adımlar aşağıdaki gibidir [30] [31]:

1. n adet gözlem, n tane küme kabul edilip işleme başlanır.
2. Tekli uzaklık matrisi hesaplanır.
3. En yakın iki küme birleştirilir, diğer kümeler aynı bırakılır.
4. Küme sayısı bir azaldığından dolayı, en küçük uzaklıklara göre güncellenen uzaklık matrisi birleştirilen küme için birleşik uzaklık, diğer kümeler için tekli olarak hesaplanır.
5. 3. ve 4. adımlar tüm gözlemler bir küme altında toplanana kadar tekrarlanır.
6. Her birleşmenin gösterildiği dendrogram oluşturulur.
7. Kaç küme elde edilmek isteniyorsa dendrogramın bu sayısı verecek uygun hizasından kesme yapılır.

Örnek

Aşağıdaki tabloda 5 gözlemden oluşan bir veri kümesi verilmiştir. Bu veri kümesini Tek bağlantılı yöntem ile çözümünün nasıl yapılacağı anlatılmıştır (Çizelge 4.5).

Çizelge 4.5 Hiyerarşik kümeleme örneği

Gözlem Numarası	1. Öznitelik Değeri	2. Öznitelik Değeri
1	2	0
2	3	1
3	3	2
4	10	6
5	7	5

İlk olarak uzaklık matrisi hesaplanır. Öklid uzaklığı kullanılarak hesaplanmış uzaklık matrisi aşağıdaki gibidir (Çizelge 4.6). Uzaklık matrisi simetrik olduğundan sadece alt kare matris hesaplanmıştır.

Çizelge 4.6 Uzaklık matrisi

	1	2	3	4	5
1	0.0000				
2	1.4142	0.0000			
3	2.2360	1.0000	0.0000		
4	10.0000	8.6023	8.0622	0.0000	
5	7.0710	5.6568	5.0000	3.1622	0.0000

En küçük uzaklık değeri olarak $d(2,3) = 1$ seçilir. 2. ve 3. Gözlemler birleştirilir

Tek bağlantı yönteminde en küçük uzaklık seçilir. 2 ve 3. gözlemlerin birleştirilmesiyle hesaplamalar ve yeni uzaklık matrisi aşağıdaki gibidir (Çizelge 4.7).

$$d\big((2,3),1\big) = \min\big(d(2,1), d(2,3)\big) = \min(1.4142, 1.0000) = 1.0000$$

$$d\big((2,3),4\big) = \min\big(d(2,4), d(3,4)\big) = \min(10.0000, 8.0622) = 8.0622$$

$$d\big((2,3),5\big) = \min\big(d(2,5), d(3,5)\big) = \min(5.6568, 5.0000) = 5.0000$$

Çizelge 4.7 Güncellenen uzaklık matrisi 1. adım

	1	2-3	4	5
1	0.0000			
2-3	1. 0000	0.0000		
4	10. 0000	8.0622	0.0000	
5	7.0710	5. 0000	3.1622	0.0000

Uzaklık matrisinde en küçük uzaklık 2-3 ve 1 arasında olduğu için 2-3 kümesi ve 1 gözlemi birleştirilir. Hesaplamalar ve yeni uzaklık matrisi aşağıdaki gibidir (Çizelge 4.8).

$$d\big((1,2,3),4\big) = \min(d(1,4), d(2,4), d(3,4)) = \min(10, 8.6023, 8.0622) = 8.0622$$

$$d\big((1,2,3),5\big) = \min(d(1,5), d(2,5), d(3,5)) = \min(7.0710, 5.6568, 5) = 5$$

Çizelge 4.8 Güncellenen uzaklık matrisi 2. adım

	1-2-3	4	5
1-2-3	0. 0000		
4	8.0622	0. 0000	
5	5. 0000	3.1622	0. 0000

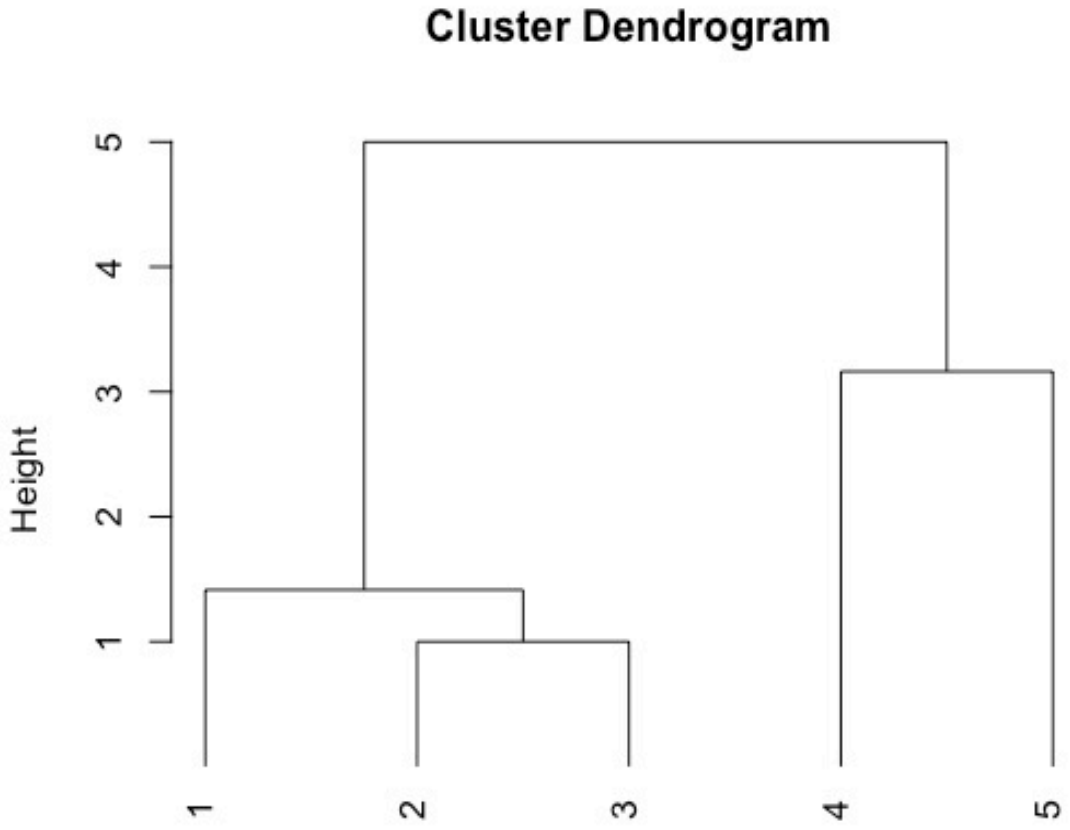
Uzaklık matrisinde en küçük uzaklık 4. ve 5. gözlem arasında olduğu için bu gözlemler birleştirilir. Hesaplamalar ve yeni uzaklık matrisi aşağıdaki gibidir (Çizelge 4.9).

$$d\big((1,2,3),(4,5)\big) = \min(d(1,4), d(2,4), d(3,4), d(1,5), d(2,5), d(3,5))$$
$$= \min(10, 8.6023, 8.0622, 7.0710, 5.6568, 5) = 5$$

Çizelge 4.9 Güncellenen uzaklık matrisi 3. adım

	1-2-3	4-5
1-2-3	0. 0000	
4-5	5. 0000	0. 0000

Son olarak da 1-2-3 ve 4-5 gözlemleri birleştirilir. Bu birleşimi dendrogram ile gösterimi aşağıdaki gibidir (Şekil 4.3).



Şekil 4.3 Hiyerarşik kümeleme dendrogramı

4.2.2 Tam Bağlantılı (Complete Linkage) Yöntemi

Bu kümeleme yönteminde ilk durumda her gözlem farklı bir kümeye ait olarak değerlendirilir. Daha sonra hesaplanan uzaklık matrisi yardımıyla her adımda en uzak iki gözlem birleştirilir.

Hiyerarşik kümelemenin tam bağlantı yöntemi uygulanırken aşağıdaki adımlar uygulanır. Sırasıyla uygulanacak adımlar aşağıdaki gibidir [30][31]:

1. n adet gözlem, n tane küme kabul edilip işleme başlanır.
2. Tekli uzaklık matrisi hesaplanır.
3. En yakın iki küme birleştirilir, diğer kümeler aynı bırakılır.
4. Küme sayısı bir azaldığından dolayı, en büyük uzaklıklara göre güncellenen uzaklık matrisi birleştirilen küme için birleşik uzaklık, diğer kümeler için tekli olarak hesaplanır.
5. 3. ve 4. adımlar tüm gözlemler bir küme altında toplanana kadar tekrarlanır.
6. Her birleşmenin gösterildiği dendrogram oluşturulur.
7. Kaç küme elde edilmek isteniyorsa dendrogramın bu sayısı verecek uygun hizasından kesme yapılır

Örnek

Tam bağlantılı kümeleme yöntemi için aynı adımlar tekrarlanacak olup, uzaklık kıyaslamalarında en küçük uzaklık yerine en büyük uzaklık tercih edilerek yapılır. Bu yöntem ile yapıldığında dendrogram yine aynı şekilde olacaktır.

4.3 K- Ortalamalar ve Hiyerarşik Kümeleme Algoritmalarının R dili ile Uygulaması

Örnek veri kümesi olarak kullanılan IRIS veri dosyası [32] kullanılmıştır. IRIS veri dosyası 150 kayıt ve 5 öznitelikten oluşmaktadır. Beş öznitelikten bir tanesi sınıf etiketini içermektedir. IRIS veri dosyasının örnek olarak seçilmesinin sebebi doğruluğunun bilinmesi ve araştırmacılar tarafından sıkça kullanılmasıdır.

4.3.1 K-Ortalamalar Algoritmasının R Dili ile Uygulanması

clustering.R

Source on Save | Run | Source

```
1  library(stats)
2  library(base)
3
4  data("iris")
5  # gercek sinif degerleri tags degiskenine atanir
6  tags <- iris$Species
7
8  # gercek degerler veri kümesinden cikartilir
9  iris$Species <- NULL
10
11 ## k-ortalamalar algoritmasının k = 3 için uygulanmasi
12
13 iris_kmeans <- kmeans(iris, centers = 3)
14
15 # iris_kmeans liste degiskenin icindeki cluster parametresi kume degerlerini icermektedir
16 iris_kmeans$cluster
17
18 # iris_kmeans liste degiskenin icerdigi parametreler
19 print(iris_kmeans)
20
```

15:63 (Top Level) R Script

Şekil 4.4 K- Ortalamalar algoritmasının R uygulaması

Yukarıdaki kod çalıştırıldığında (Şekil 4.4) k- Ortalamalar algoritması ile 3 kümeye ayrılmış olacaktır. Kümelemenin atandığı değişkeni ekrana yazdırıldığında aşağıdaki çıktı alınacaktır (Şekil 4.5).

Kümelemenin atandığı değişken içinde, küme merkezleri, hangi elemanın hangi kümeye ait olduğu bilgileri yer almaktadır.

Console ~/

```
> print(iris_kmeans)
K-means clustering with 3 clusters of sizes 50, 62, 38

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     5.901613    2.748387     4.393548    1.433871
3     6.850000    3.073684     5.742105    2.071053

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 3 2
 [55] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3
[109] 3 3 3 3 3 2 2 3 3 3 3 2 3 2 3 3 2 2 3 3 3 3 3 2 3 3 3 3 2 3 3 3 2 3 3 2 3 3 2 3 3 2

Within cluster sum of squares by cluster:
[1] 15.15100 39.82097 23.87947
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
>
```

Şekil 4.5 Kümeleme sonuçları

4.3.2 Hiyerarşik Kümeleme Algoritmalarının R Dili ile Uygulaması

clustering.R

Source on Save | Run | Source

```r
library(stats)
library(base)

data("iris")
# gercek sinif degerleri tags degiskenine atanir
tags <- iris$Species

# gercek degerler veri kümesinden cikartilir
iris$Species <- NULL

# uzaklik matrisi hesaplanir
dist_matrix <- dist(iris)

# hclust fonksiyonu yardimiyla hiyerarsik kümeleme yapilir
iris_hc_single <- hclust(dist_matrix, method = "single")

# plot fonksiyonu ile dendrogram cizdirilir
plot(iris_hc_single)

# iris_hc_single liste degiskenin cutree fonksiyonu ile kume degerleri elde edilir
clusterCut <- cutree(iris_hc_single, 3)

# iris_kmeans liste degiskenin icerdigi parametreler
print(iris_hc_single)

```

26:1 (Top Level) R Script

Şekil 4.6 Hiyerarşik kümeleme algoritmasının R uygulaması

Yukarıdaki kod çalıştırıldığında (Şekil 4.6) Hiyerarşik kümeleme (Single Linkage) algoritması ile 3 kümeye ayrılmış olacaktır.

clustering.R

Source on Save | Run | Source

```r
library(stats)
library(base)

data("iris")
# gercek sinif degerleri tags degiskenine atanir
tags <- iris$Species

# gercek degerler veri kümesinden cikartilir
iris$Species <- NULL

# uzaklik matrisi hesaplanir
dist_matrix <- dist(iris)

# hclust fonksiyonu yardimiyla hiyerarsik kümeleme yapilir
iris_hc_single <- hclust(dist_matrix, method = "complete")

# plot fonksiyonu ile dendrogram cizdirilir
plot(iris_hc_single)

# iris_hc_single liste degiskenin cutree fonksiyonu ile kume degerleri elde edilir
clusterCut <- cutree(iris_hc_single, 3)

# iris_kmeans liste degiskenin icerdigi parametreler
print(iris_hc_single)

```

15:57 (Top Level) R Script

Şekil 4.7 Hiyerarşik kümeleme sonuçları

Bir önceki sayfadaki kod (Şekil 4.7) çalıştırıldığında Hiyerarşik kümeleme (Complete Linkage) algoritması ile 3 kümeye ayrılmış olacaktır.

BÖLÜM 5

KÜMELEME DEĞERLENDİRME KRİTERLERİ

Kümeleme analizi, veri madenciliği, bilgi bilimi, tarım teknolojisi ve biyomedikal gibi birçok araştırma alanında önemli bir tekniktir. Farklı hesaplama fonksiyonlarına sahip kümeleme algoritmaları farklı çözümler sunarlar. Tüm olası veri kümeleri için algoritma ve hesaplama fonksiyonu için tek bir en iyi seçimi yoktur. Bu nedenle amaç, bir veri kümesi için mümkün olan en iyi kümeleme yöntemini seçmektir. Çoğu kümeleme algoritması için, küme sayısı bir parametre olarak ayarlanır, kümeleme işlemi uygulanmadan önce belirlenmesi gerekir. Bununla birlikte, kümelerin sayısı başlangıçta çoğu veri kümesi için mevcut değildir. Kümelenme sonuçları bir kümeleme algoritması ile elde edildiğinde, bir sonraki önemli adım, kümelenme çözümlerinin, veri kümesi için, genellikle kümelerin sayısı için optimal bir çözüm veya küme yapısını belirlemek üzere değerlendirilmesidir. Bu adım, kümelenme sonuçlarının veya verilen veri kümesine en uygun kümeleme çözümünü bulmayı amaçlayan kümelenme sonucunun değerlendirilmesine bağlıdır. Bu problemlerin hepsi kümeleme geçerlilik analizi başlığı altında incelenir [33].

5.1. Kullanılan Kümeleme Değerlendirme Kriterleri

Bu tez kapsamındaki yapılan uygulamada; Calinski – Harabasz [34], Davies – Bouldin [35], Dunn [36], Silhouette [37] ve Wemmert – Gançarski [38] kriterlerinden faydalanılmıştır. Aşağıdaki tabloda formülleri ve optimal değer için hangi değerin seçilmesi gerektiği verilmiştir (Çizelge 5.1).

Çizelge 5.1 Uzaklıl Ölçütleri

Kriter	Formül	Optimal Değer
Calinski – Harabasz	$\frac{\sum_i n_i \frac{d^2(c_i, c)}{KS-1}}{\sum_i \sum_{x \in c_i} \frac{d^2(x, c_i)}{(n\text{-}KS)}}$	Max
Davies – Bouldin	$\frac{1}{KS}\sum_i max_j, j \neq i \left\{\left[\frac{1}{n_i}\sum_{x \in c_i} d(x, c_i) + \frac{1}{n_j}\sum_{x \in c_j} d(x, c_j)\right] / d(c_i, c_j)\right\}$	Min
Dunn	$min_i \left\{min_j \left(\frac{min_{x \in C_i, y \in C_j} d(x,y)}{max_k \{max_{x,y \in C_k} d(x,y)\}}\right)\right\}$	Max
Silhouette	$\frac{1}{KS}\sum_i \left\{\frac{1}{n_i}\sum_{x \in c_i} \frac{b(x)\text{-}a(x)}{max\lvert b(x), a(x)\rvert}\right\}$ $a(x)$: *ilgili küme içindeki ortalama uzaklık* $b(x)$: *ilgili küme haricindeki kümelere olan ortalama uzaklık değerlerinin minimum değeri*	Max
h	$R(M) = \frac{\lVert M\text{-}G^{\{k\}} \rVert}{min_{k' \neq k} \lVert M\text{-}G^{\{k'\}} \rVert}$ $J_k = max\left\{0, 1 - \frac{1}{n_k}\sum_{i \in c_k} R(M_i)\right\}$ $WG = \frac{1}{KS}\sum_{k=1}^{K} n_k J_k$	Max

n: veri kümesindeki gözlem sayısı, c: veri kümesinin merkezi, KS : küme sayısı, C_i i indisli küme, n_i : C_i kümesindeki gözlem sayısı, c_i : i indisli kümenin merkezi, d(x,y) : x ve y arasındaki uzaklık

Silhouette indeksi: Kümelerin kompaktlığını ve birbirinden ayrılmasını yansıtan bir karma indekstir. Daha büyük bir ortalama indeks, kümelenme sonucunun daha iyi olduğunu gösterir. Bu yüzden en uygun küme sayısı, en büyük ortalama Siluet değerini veren en uygun k değeridir.

Davies-Bouldin indeksi: Her bir küme ve o kümeye en benzer olanı arasındaki ortalama benzerliği hesaplanarak bulunur. Bu indeksin küçük değerleri, kompakt ve birbirinden uzak olan merkezlere sahip kümelere karşılık gelir. Bu nedenle, optimal k değerini en küçük değere sahip olan k belirler.

Calinski-Harabasz indeksi: Küme içi uzaklık ve kümeler arası uzaklıklar hesaplanarak bulunur. Optimal k değerini belirlemek için en yüksek değere sahip olduğu k değeri seçilir.

Dunn index: Kümeler arası uzaklığı en aza indirirken kümeler arası mesafeleri maksimize eder. Bu indeksin büyük değerleri, kompakt ve iyi ayrılmış kümelerin elde edildiğini gösterir, bu nedenle en büyük değere sahip k değeri optimal küme sayısı olarak belirlenir.

Wemmert – Gançarski: Kümeler arasındaki maksimum ve minimum uzaklıklardan faydalanılarak hesaplanır. Bu indeksin büyük değerleri, kompakt ve iyi ayrılmış kümelerin elde edildiğini gösterir, bu nedenle en büyük değere sahip k değeri optimal küme sayısı olarak belirlenir.

5.2 Kriterlerin R Dili ile Hesaplanması

Formülleri ve optimal değeri belirlemek için en küçük veya en büyük değerlerden hangisinin optimal k değerini belirleyeceğini bir önceki başlıkta anlatılmıştır. Örnek veri kümesi olarak popüler olan iris veri kümesi [32] kullanılacaktır.

Iris veri seti 150 gözlem ve 5 öznitelikten oluşmaktadır. Özniteliklerden biri sınıf etiketini belirtmektedir. Sınıf etiketini içeren öznitelikte 3 farklı sınıf mevcuttur. Kümelemede kullanılacak öznitelikler nümerik değerlerden oluşmaktadır. Nümerik değerlere ait en küçük değer, 1. kartil değeri, medyan değeri, ortalama, 3.kartil değeri ve en büyük değeri gösteren özeti aşağıdaki gibidir (Şekil 5.1).

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
Median :5.800   Median :3.000   Median :4.350   Median :1.300
Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

Şekil 5.1 Irıs veri dosyanın R özeti

3 sınıfa ait 50'şer adet gözlem bulunmaktadır. Veri kümesinden sınıf etiketi kaldırılarak K-ortalamalar algoritması ile k = 2,3,4,5 değerleri için kümeleme yapılmıştır. Daha sonra yukarıda açıklanan kümeleme değerlendirme kriterlerinin R dili aracılığıyla hesaplanması yapılmıştır.

```
library(stats)

library(clusterCrit)


df <- as.data.frame(iris)

df$Species <- NULL


vals_Calinski_Harabasz <- vector()

vals_Davies_Bouldin <- vector()

vals_Dunn <- vector()

vals_Silhouette <- vector()

vals_Wemmert_Gancarski <- vector()
```

```
for (k in 2:5) {
  # Perform the kmeans algorithm
  cl <- kmeans(df, k)

  # Compute the Calinski_Harabasz index
  vals_Calinski_Harabasz <-
    c(vals_Calinski_Harabasz, as.numeric(intCriteria(
      as.matrix(df), cl$cluster, "Calinski_Harabasz"
    )))
  vals_Davies_Bouldin <-
    c(vals_Davies_Bouldin, as.numeric(intCriteria(
      as.matrix(df), cl$cluster, "Davies_Bouldin"
    )))
  vals_Dunn <-
    c(vals_Dunn, as.numeric(intCriteria(as.matrix(df), cl$cluster, "Dunn")))
  vals_Silhouette <-
    c(vals_Silhouette, as.numeric(intCriteria(
      as.matrix(df), cl$cluster, "Silhouette"
    )))
```

```
vals_Wemmert_Gancarski <-
  c(vals_Wemmert_Gancarski, as.numeric(intCriteria(
    as.matrix(df), cl$cluster, "Wemmert_Gancarski"
  ))) }
```

Yukarıdaki kodun çalışması bittiğinde kümeleme kriterlerin atandığı değişkenlerin tablosu aşağıdaki gibidir (Çizelge 5.2).

Çizelge 5.2 Kümeleme değerlendirme kriterleri sonuçları

k	Silhouette	Davies-Bouldin	Calinski-Harabasz	Dunn	Wemmert – Gançarski
2	0.701	0.404	513.924	0.076	0.769
3	0.555	0.363	561.627	0.098	0.666
4	0.466	0.931	530.765	0.136	0.620
5	0.364	0.790	320.463	0.037	0.545

Yukarıdaki sonuçlara göre optimal k değerleri aşağıdaki tabloda belirtilmiştir (Çizelge 5.3).

Çizelge 5.3 Kümeleme değerlendirme kriterleri ile doğru küme sayısının belirlenmesi

Kriter	Optimal k
Silhouette	k = 2 (0.701)
Davies-Bouldin	k = 3 (0.363)
Calinski-Harabasz	k = 3 (561.627)

Çizelge 5.3 Kümeleme değerlendirme kriterleri ile doğru küme sayısının belirlenmesi(devamı)

Dunn	k = 4 (0.136)
Wemmert – Gançarski	k = 2 (0.769)

Tablodaki sonuçların yorumu olarak Davies-Bouldin indeksi ve Calinski-Harabasz indeksleri doğru küme sayısını tam olarak belirleyebilmiştir.

BÖLÜM 6

VERİ ANALİZİNE DAYALI İLERİYE DÖNÜK TAHMİN MODELLEMESİ

6.1 Dönüşüm Oranı Hesaplanması

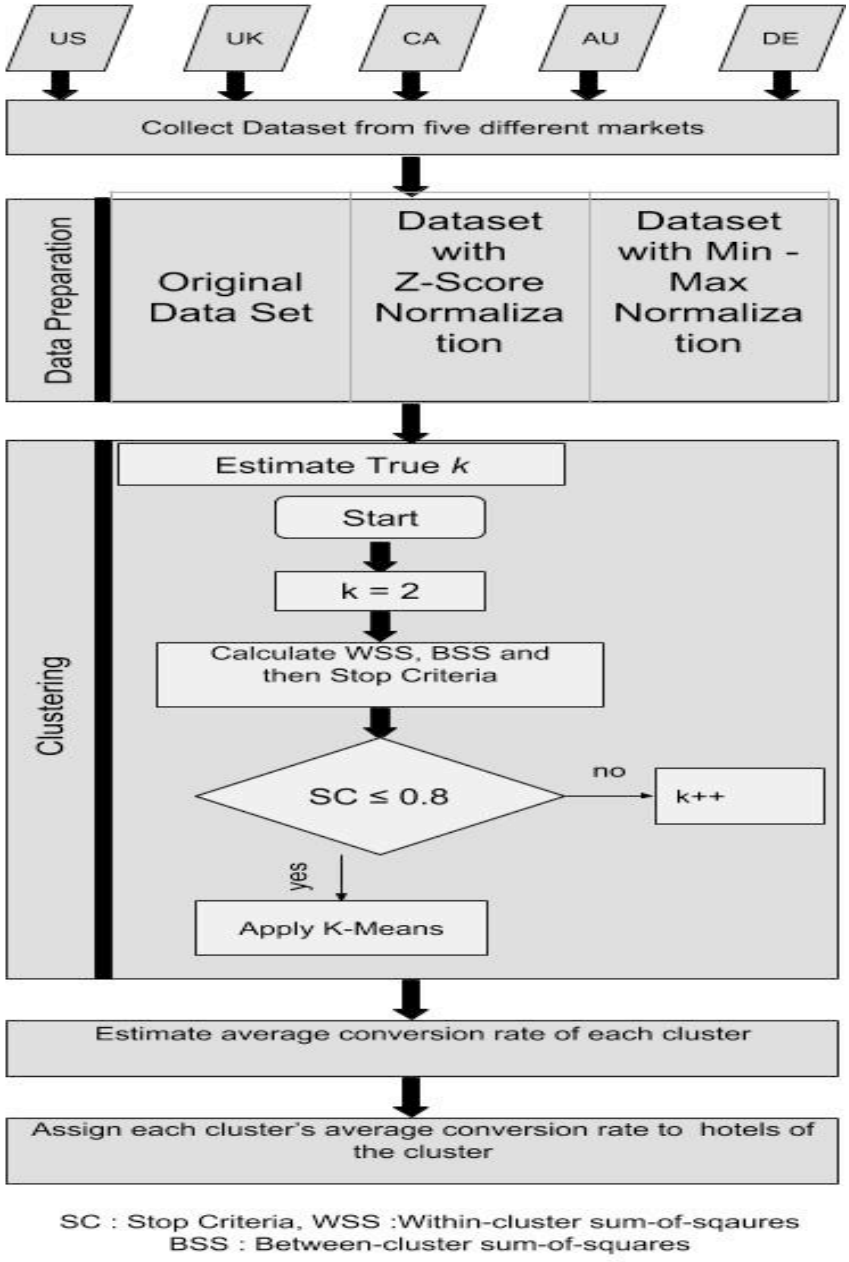
Yapılan tez kapsamında amaç E-ticarette turizm sektörü için hotellerin dönüşüm oranının tahmin edilerek oluşacak rezervasyon maliyetini azaltmaktır. İnternet üzerinden hotel satışı yapan sitelerin büyük çoğunluğu web trafiklerini para ödeyerek Trivago.com, Skyscanner.com, Kayak.com gibi meta kanallardan almaktadırlar. Burada oluşacak maliyeti azaltmak adına daha önce rezervasyonu olmayıp, müsaitliği mevcut olan hotellerin hem klasik veri madenciliği yöntemleri ile hem de hotel kümelemesi yapılarak dönüşüm oranları tahmin edilmiştir. Dönüşüm oranın formülü aşağıda verilmiştir.

$$Dönüşüm\ Oranı = \frac{Toplam\ Rezervasyon\ Sayısı}{Toplam\ Tık(Click)\ Sayısı} \tag{6.1}$$

Almanya, İngiltere, Amerika Birleşik Devletleri, Kanada ve Avustralya olmak üzere 5 farklı ülkeden hotel ve ülke bazında veriler toplanıp, veri kümesi elde edilmiştir. Tez kapsamında üzerinde çalışılan veri kümesi üzerindeki en önemli problem çok az sayıda hotelin rezervasyonun olması. Bu sebeple veri kümesinin büyük çoğunluğun dönüşüm oranı 0 olarak hesaplanmaktadır. Bu durumda bir hotelin rezervasyon maliyeti hesaplanamamaktadır. Maliyeti hesaplanamayan hotelin zarara yol açma durumu söz konusudur. Bu yüzden bir hotelin rezervasyon maliyetinin hesaplanması son derece önemlidir. Örnek rezervasyon maliyeti hesabı şu şekildedir. Bir hotelin dönüşüm oranının %2.5 olduğunu varsayalım. Bu durumda hotel ortalama 40 tıklanmada 1 rezervasyon almaktadır. Bir tık maliyetinin 30 sent olduğunu kabul edilirse ortalama 12 dolarlık bir rezervasyon maliyeti ortaya çıkacaktır.

6.2 K- Ortalamalar Algoritmasına Dayalı Dönüşüm Oranı Tahmini ile Rezervasyon Optimizasyonu

Yapılan uygulama ile klasik veri madenciliği yaklaşımlarından olan, öznitelikteki eksik gözlemleri o özniteliğin ortalaması ile doldurmak yerine, hotelleri kümeleyip o kümedeki hotellerin ortalama dönüşüm oranı hesaplanarak doldurulmuştur. Bildiride ele alınan yöntem kümeler arası uzaklığın toplam uzaklığa oranı ile doğru küme sayısı belirleme yöntemi olarak kullanılmıştır. Bu yöntemin akışı aşağıdaki gibidir (Şekil 6.1).



Şekil 6.1 Doğru küme sayısını belirleme adımları

R dili ile K –Ortalamalar Algoritması uygulandığında kümeleme sonuçlarının bulunduğu listenin içinde between_SS / total_SS değeri mevcuttur. Doğru küme sayısını belirleme kriteri olarak bu oranın 80'den büyük olduğu en küçük k değeri alınır. Bu yöntem ile elde edilen sonuçlar, sonuçlar kısmında açıklanmıştır.

6.3 Doğru Küme Sayısı Belirlenerek K- Ortalamalar Algoritmasına Dayalı Dönüşüm Oranı Tahmini ile Rezervasyon Optimizasyonları

Bu kapsamda ilk olarak doğru küme sayısı olan k değerinin belirlenmesi gerekmektedir. Bu değeri hesaplayabilmek için Elbow ve Gap İstatistik Değeri yöntemleri kullanılmıştır. Bu yöntemler aynı veri kümesinin, iki farklı dönüşüm uygulanmış hali ve dönüşüm uygulanmamış hali için hesaplanmıştır. Belirtilen üç farklı durum için de k değerleri 2 ile 15 aralığında alınıp, optimal k değerleri hesaplanmıştır. Sonuçlar kısmında tablo ve grafik halinde detaylı bir şekilde açıklanmıştır.

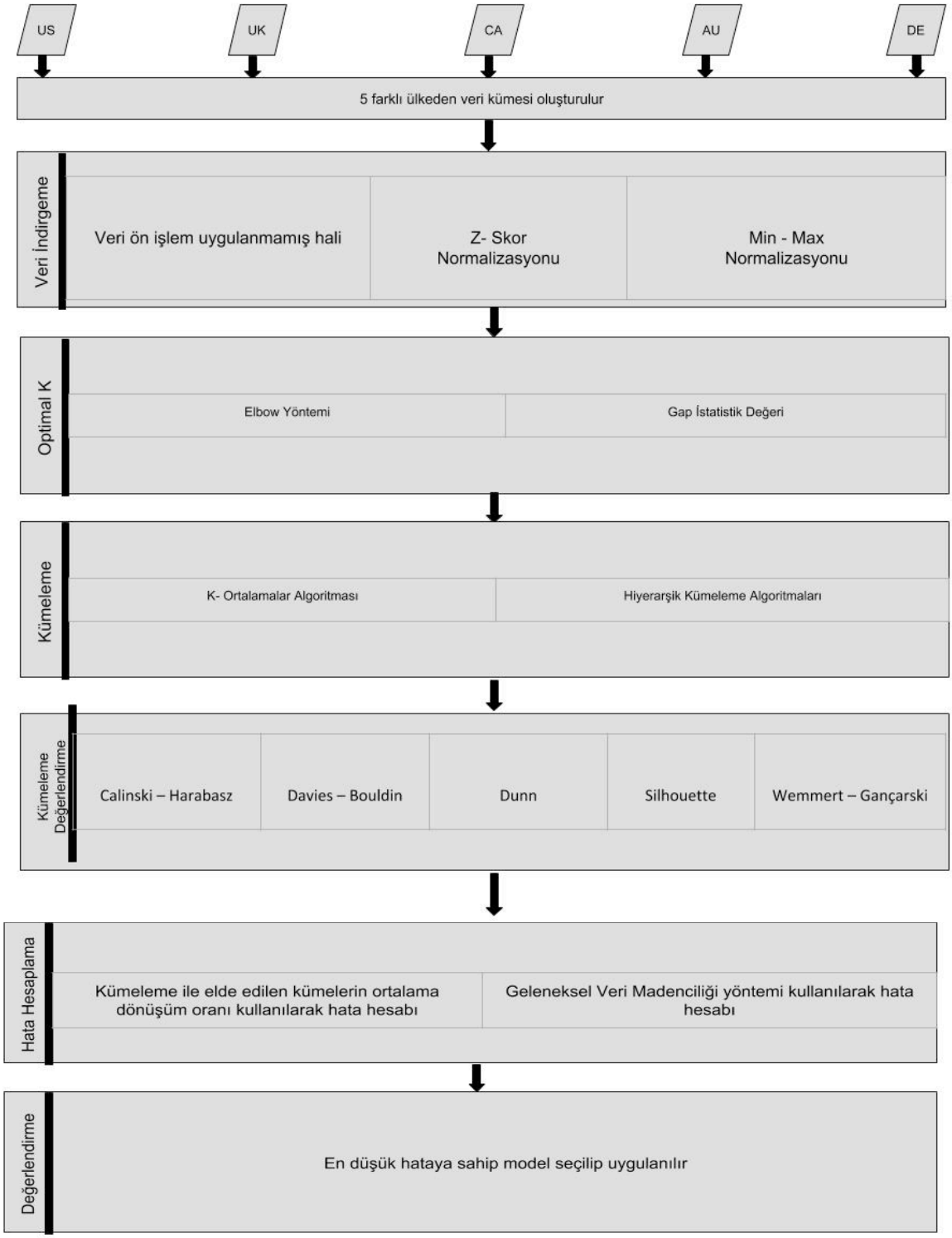
Optimal k değerleri belirlendikten sonra daralan k aralıkları için K- Ortalamalar Algoritması ile kümeleme yapılmıştır. Yapılan kümeleme sonucu elde edilen sonuçlar kullanılarak Calinski – Harabasz, Davies – Bouldin, Dunn, Silhouette ve Wemmert – Gançarski kümeleme değerlendirme kriterleri hesaplanıp kıyaslama yapılmıştır. Bu kriterlere göre 3 farklı veri kümesi için de hangi k değerinin daha geçerli olduğu belirlenmiştir. Sonuçlar kısmında tablo ve grafik halinde detaylı bir şekilde açıklanmıştır.

Bu aşamadan sonra oluşan her durum için Ortalama Karesel Hatanın Karekökü (RMSE) ile hatalar hesaplanmıştır. Ortalama Karesel Hatanın Karekökü (RMSE) formülü aşağıdaki gibidir.

$$\text{Ortalama Karesel Hatanın Karekökü (RMSE)} = \sqrt{\frac{\sum_{i=1}^{n}(Actual_i - Predicted_i)^2}{N}} \quad (6.1)$$

Tüm varyasyonlara göre elde edilen hatalar, klasik veri madenciliği yaklaşımı ile uygulanan duruma göre daha düşüktür.

Detaylı akış şeması aşağıdaki gibidir (Şekil 6.2).



Şekil 6.2 Tez kapsamında yapılan uygulamanın akışı

BÖLÜM 7

SONUÇ VE ÖNERİLER

7.1 Tez Veri Kümesinin Tanıtılması

Veri kümesi hotel ve kullanıcı ülkesi bazında 5 farklı ülke için toplanmıştır. Bu ülkeler, Almanya, İngiltere, Kanada, Amerika Birleşik Devletleri ve Avustralyadır. Kümeleme yapılırken kullanılan öznitelikler ise sırasıyla hotelin kullanıcılar tarafından görüntülenme sayısı, hotelin web sayfasında ortalama kaçıncı sırada gözüktüğü, hotelin ortalama sepet tutarı miktarı, hotelin kullanıcılar tarafından verilen puanlarla oluşan ortalama puanı ve hotelin yıldızıdır. Özniteliklerin hepsi nümerik değerlere sahiptirler.

Optimal küme sayısı üç farklı veri kümesi içinde ilk olarak Elbow yöntemi ve Gap istatistik değeri hesaplanarak optimal küme sayısı belirlenmiştir. Bu sayıyı belirlemek için k değeri tüm durumlarda [2,15] aralığında seçilmiştir. Hiyerarşik kümeleme ile elde edilen küme sonuçları iyi sonuç vermediği için tez kapsamında K- Ortalamalar algoritması tercih edilmiştir. Belirtilen yöntemlerle optimal k değerleri belirlendikten sonra K- Ortalamalar algoritması ile kümeleme işlemi yapılmıştır. Bu adımdan sonra belirlenen k değerleri için Calinski – Harabasz, Davies – Bouldin, Dunn, Silhouette ve Wemmert – Gançarski kümeleme değerlendirme kriterleri kullanılarak ve hatalar hesaplanarak problemin kesim çözümü için kullanılacak yöntem belirlenmiştir.

7.2 Geleneksel Yöntemler ile Yapılan Dönüşüm Oranı Tahmini

Dönüşüm oranı olmayan hoteller için ortalama dönüşüm oranı yansıtılarak ortalama karesel hatanın karekökü (RMSE) değeri hesaplandığında hata oranı 0.04848399 olarak hesaplanmıştır.

7.3 K- Ortalamalar Algoritmasına Dayalı Dönüşüm Oranı Tahmini ile Rezervasyon Optimizasyonu Sonuçları

6.2 de bahsedilen yöntem ile ön işlem uygulanmamış ve iki farklı yöntem ile normalleştirilmiş veri kümeleri için belirlenen doğru küme sayısı değerleri aşağıdaki tabloda verilmiştir (Çizelge 7.1).

Çizelge 7.1 Önerilen yöntem ile elde edilen küme sayıları

	Ön işlem uygulanmamış veri kümesi	Z-Skor normalizasyonu uygulananan veri kümesi	Min-max normalizasyonu uygulananan veri kümesi
Doğru küme sayısı	4	23	5

Yukarıdaki küme sayısı değerlerine göre K- Ortalamalar algoritması uygulandığında hata oranları aşağıdaki gibidir (Çizelge 7.2).

Çizelge 7.2 Önerilen yöntem ile hesaplanan hatalar

	Ön işlem uygulanmamış veri kümesi	Z-Skor normalizasyonu uygulananan veri kümesi	Min-max normalizasyonu uygulananan veri kümesi
Doğru küme sayısı	4	23	5
RMSE	**0.047027580**	0.047107180	0.047110230
MSE	0.002211593	0.002211593	0.002219373

7.4 K – Ortalamalar Algoritması ile Ön İşlem Uygulanmamış Veri Kümesi İçin Elde Edilen Sonuçlar

Ön işlem uygulanmamış veri kümesinin R dili ile alınan özeti aşağıdaki gibidir (Şekil 7.1). Özet içerisinde sırasıyla her öznitelik için en küçük değer, 1. kartil değeri, medyan, ortalama, 3.kartil değeri ve en büyük değer mevcuttur.

```
> summary(df)
   Hotel.Impr      Avg.Hotel.Pos    Avg.Min.Price        Rating          Stars
Min.   :     1   Min.   :  1.00   Min.   :    9.0   Min.   :41.00   Min.   :0.00
1st Qu.:   371   1st Qu.: 13.20   1st Qu.:  237.0   1st Qu.:77.00   1st Qu.:3.00
Median :  1245   Median : 18.60   Median :  398.0   Median :81.00   Median :4.00
Mean   :  5829   Mean   : 21.67   Mean   :  610.5   Mean   :80.45   Mean   :3.43
3rd Qu.:  4414   3rd Qu.: 26.50   3rd Qu.:  709.0   3rd Qu.:85.00   3rd Qu.:4.00
Max.   :552676   Max.   :248.50   Max.   :43587.0   Max.   :96.00   Max.   :5.00
```

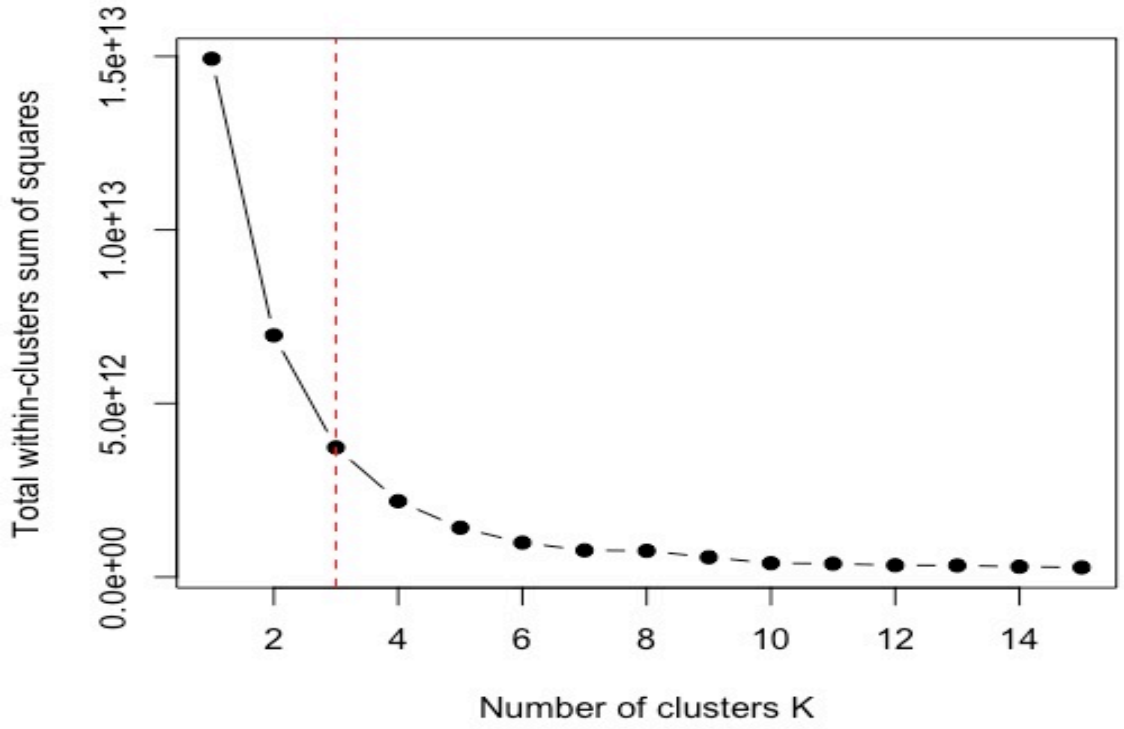
Şekil 7.1 Tez veri kümesinin R özeti

Elbow Yöntemi ve Gap istatistik değeri sonuçları aşağıdaki tabloda verilmiştir (Çizelge 7.3).

Çizelge 7.3 Önişlem uygulanamamış veri kümesi için doğru küme sayısı sonuçları

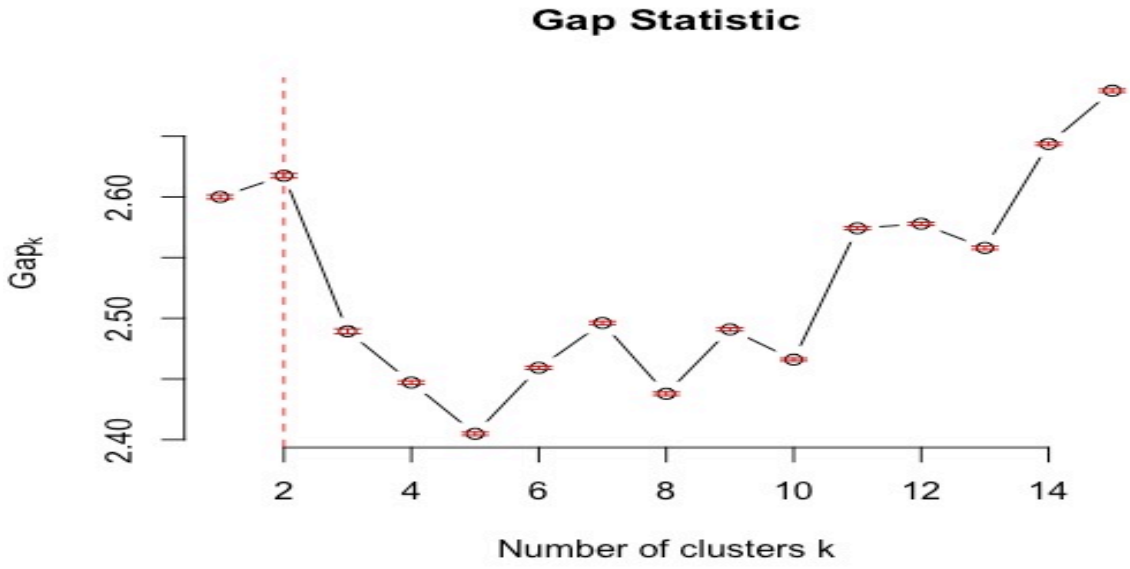
k	Elbow Yöntemi	Gap İstatistik Yöntemi
	WSS	Gap İstatistik Değeri
2	6.96E+12	2.617442
3	3.73E+12	2.4893
4	2.18E+12	2.447046
5	1.42E+12	2.404869
6	9.89E+11	2.459213
7	7.69E+11	2.496132
8	7.54E+11	2.437852
9	5.68E+11	2.490905
10	3.98E+11	2.465973
11	3.83E+11	2.574168
12	3.38E+11	2.577883
13	3.31E+11	2.558013
14	2.97E+11	2.643679
15	2.76E+11	2.687539

Elbow yöntemi ile çizdirilen grafik sonucunda optimal k değerinin 3 olduğu belirlenmiştir. Grafik aşağıdaki gibidir (Şekil 7.2).



Şekil 7.2 Doğru küme sayısı grafiği

Gap istatik değeri ile optimal k değerini seçerken iken ilk en büyük (first max) yöntemi kullanılmıştır. Gap istatistik değeri ile hesaplanan k değerinin 2 olduğu görülmektedir. Grafiği aşağıdaki gibidir (Şekil 7.3).



Şekil 7.3 Doğru küme sayısı grafiği

Bu veri kümesi için k değerleri 2 ve 3 olarak belirlenmiştir. Bundan sonraki yapılacak olan kümeleme değerlendirme kriterleri ve hata hesaplarında bu durum için k = 2 ve k = 3 için sonuçlar elde edilmiştir.

Kümeleme değerlendirme kriterleri ile elde edilen sonuçlar aşağıdaki tabloda verilmiştir. Hesaplanan değerleri göre optimal k seçilmesi gereken değerleri siyah ile gösterilmiştir (Çizelge 7.4).

Çizelge 7.4 Önişlem uygulanmamış veri kümesi için kümeleme değerlendirme kriterleri

k	Calinski – Harabasz	Davies – Bouldin	Dunn	Silhouette	Wemmert – Gançarski
2	61198.98	**0.4908621**	**0.000313599**	**0.6595748**	**0.9315481**
3	**80258.62**	0.537159	0.000083007	0.5759705	0.894332

Yukarıdaki tabloda belirlenen değerleri göre klasik veri madenciliği yöntemleri ile kümeleme sonucu ile elde edilen hata matrisi aşağıdaki gibidir (Çizelge 7.5).

Çizelge 7.5 Önişlem uygulanmamış veri kümesi için hata sonuçları

k	Ortalama Karesel Hatanın Karekökü (RMSE)	Ortalama Karesel Hata(MSE)
2	0.04708016	0.002216541
3	**0.04706028**	0.002214670

7.5 K – Ortalamalar algoritması ile Min – max normalizasyonu uygulanan veri kümesi için elde edilen sonuçlar

Min – Max normalizasyonu uygulanmış veri kümesinin R dili ile alınan özeti aşağıdaki gibidir (Şekil 7.4).

```
> summary(df_normalized_0to1)
   Hotel.Impr        Avg.Hotel.Pos     Avg.Min.Price         Rating          Stars
Min.   :0.0000000   Min.   :0.00000   Min.   :0.000000   Min.   :0.0000   Min.   :0.0000
1st Qu.:0.0006695   1st Qu.:0.04929   1st Qu.:0.005232   1st Qu.:0.6545   1st Qu.:0.6000
Median :0.0022509   Median :0.07111   Median :0.008927   Median :0.7273   Median :0.8000
Mean   :0.0105453   Mean   :0.08350   Mean   :0.013803   Mean   :0.7173   Mean   :0.6859
3rd Qu.:0.0079857   3rd Qu.:0.10303   3rd Qu.:0.016063   3rd Qu.:0.8000   3rd Qu.:0.8000
Max.   :1.0000000   Max.   :1.00000   Max.   :1.000000   Max.   :1.0000   Max.   :1.0000
```

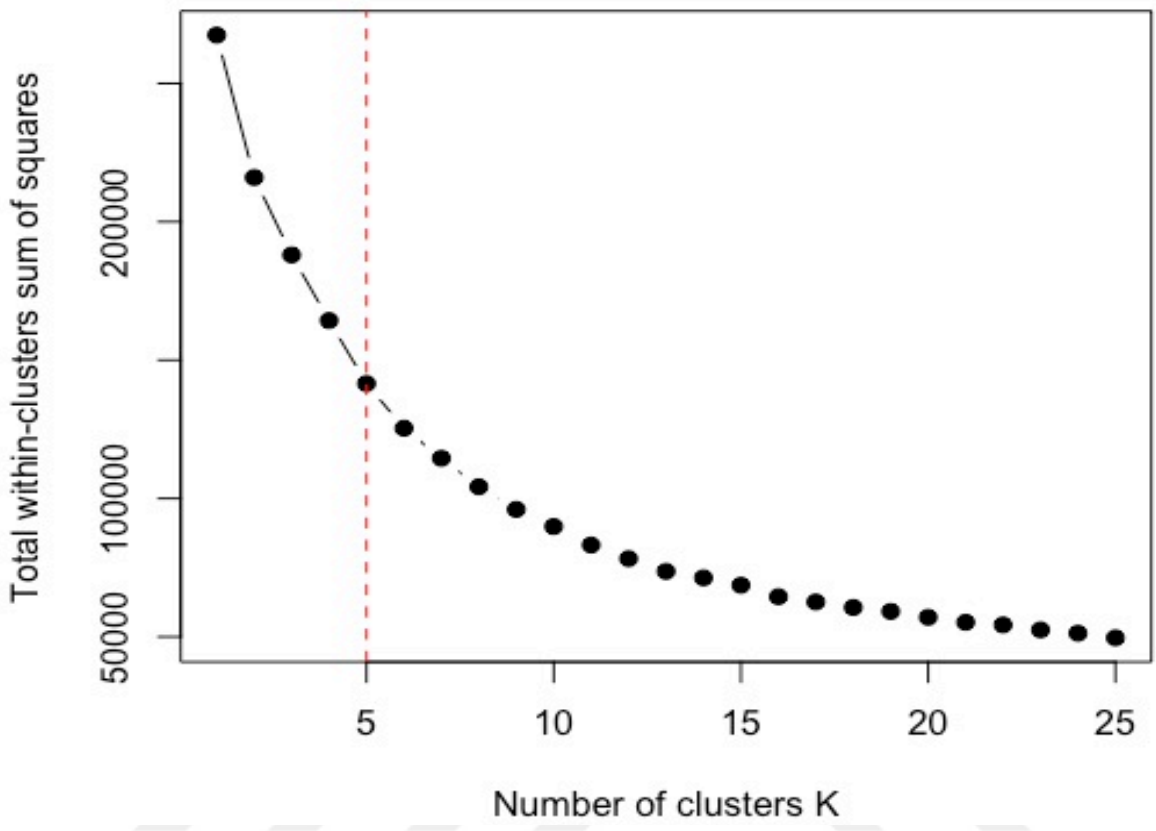
Şekil 7.4 Tez veri kümesi R özeti

Elbow Yöntemi ve Gap istatistik değeri sonuçları aşağıdaki tabloda verilmiştir (Çizelge 7.6).

Çizelge 7.6 Min- max normalizasyonu uygulanan veri kümesi için doğru küme sayısı sonuçları

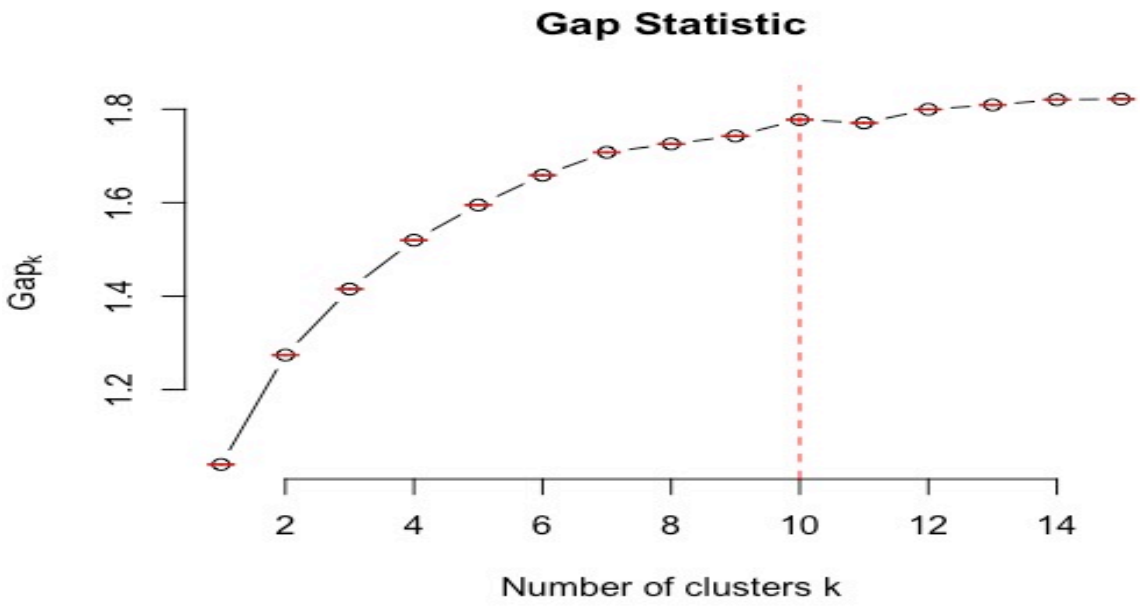
k	Elbow Yöntemi	Gap İstatistik Yöntemi
	WSS	Gap İstatistik Değeri
2	215982.24	1.274069
3	187984.28	1.415199
4	164298.22	1.519672
5	141525.37	1.594936
6	125389.3	1.658588
7	114515.79	1.707780
8	104231.97	1.725734
9	96049.01	1.742651
10	89865.64	1.777622
11	83176.09	1.770597
12	78257.58	1.799640
13	73641.33	1.809097
14	71326.77	1.820417
15	68634.04	1.821488

Elbow yöntemi ile çizdirilen grafik sonucunda optimal k değerinin 5 olduğu belirlenmiştir. Grafik aşağıdaki gibidir (Şekil 7.5).



Şekil 7.5 Doğru küme sayısı grafiği

Gap istatik değeri ile optimal k değerini seçerken iken ilk en büyük (first max) yöntemi kullanılmıştır. Gap istatistik değeri ile hesaplanan k değerinin 10 olduğu görülmektedir. Grafiği aşağıdaki gibidir (Şekil 7.6).



Şekil 7.6 Doğru küme sayısı grafiği

Bu veri kümesi için k değerleri 5 ve 10 olarak belirlenmiştir. Bundan sonraki yapılacak olan kümeleme değerlendirme kriterleri ve hata hesaplarında bu durum için

$k = \{5,6,7,8,9,10\}$ alınarak sonuçlar elde edilmiştir.

Kümeleme değerlendirme kriterleri ile elde edilen sonuçlar aşağıdaki tabloda verilmiştir. Hesaplanan değerleri göre optimal k seçilmesi gereken değerleri siyah ile gösterilmiştir (Çizelge 7.7).

Çizelge 7.7 Min- max normalizasyonu uygulanan veri kümesi kümeleme değerlendirme kriterleri

k	Calinski – Harabasz	Davies – Bouldin	Dunn	Silhouette	Wemmert – Gançarski
5	**55825.15**	0.8423955	**0.005025751**	**0.4542787**	**0.5691994**
6	54708.21	0.7205892	0.001630285	0.4251950	0.5437635
7	53825.95	0.7882648	0.003041979	0.4153195	0.5528490
8	53087.67	0.8290306	0.001090597	0.4084710	0.5565969
9	50647.56	0.8494697	0.001612800	0.3801532	0.5323822
10	49003.42	**0.5378472**	0.001108988	0.3895347	0.5355465

Yukarıdaki tabloda belirlenen değerleri göre klasik veri madenciliği yöntemleri ile kümeleme sonucu ile elde edilen hata matrisi aşağıdaki gibidir (Çizelge 7.8).

Çizelge 7.8 Min- max normalizasyonu uygulanan veri kümesi için hata sonuçları

k	Ortalama Karesel Hatanın Karekökü (RMSE)	Ortalama Karesel Hata(MSE)
5	**0.04710647**	0.002219308
6	0.04710953	0.002219019
7	0.04710819	0.002219182
8	0.04710689	0.002219059
9	0.04710834	0.002219196
10	0.04711214	0.002219554

7.6 K – Ortalamalar Algoritması ile Z - Skor Normalizasyonu Uygulanan Veri Kümesi İçin Elde Edilen Sonuçlar

Z- Skor normalizasyonu uygulanmış veri kümesinin R dili ile alınan özeti aşağıdaki gibidir (Şekil 7.7).

```
> summary(df_center_scale)
   Hotel.Impr       Avg.Hotel.Pos     Avg.Min.Price        Rating            Stars        
 Min.   :-0.34923   Min.   :-1.5361   Min.   :-0.7514   Min.   :-5.88006   Min.   :-2.8483  
 1st Qu.:-0.32706   1st Qu.:-0.6293   1st Qu.:-0.4666   1st Qu.:-0.51458   1st Qu.:-0.3568  
 Median :-0.27469   Median :-0.2279   Median :-0.2655   Median : 0.08158   Median : 0.4737  
 Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000  
 3rd Qu.:-0.08477   3rd Qu.: 0.3593   3rd Qu.: 0.1230   3rd Qu.: 0.67775   3rd Qu.: 0.4737  
 Max.   :32.76806   Max.   :16.8603   Max.   :53.6862   Max.   : 2.31720   Max.   : 1.3042  
```

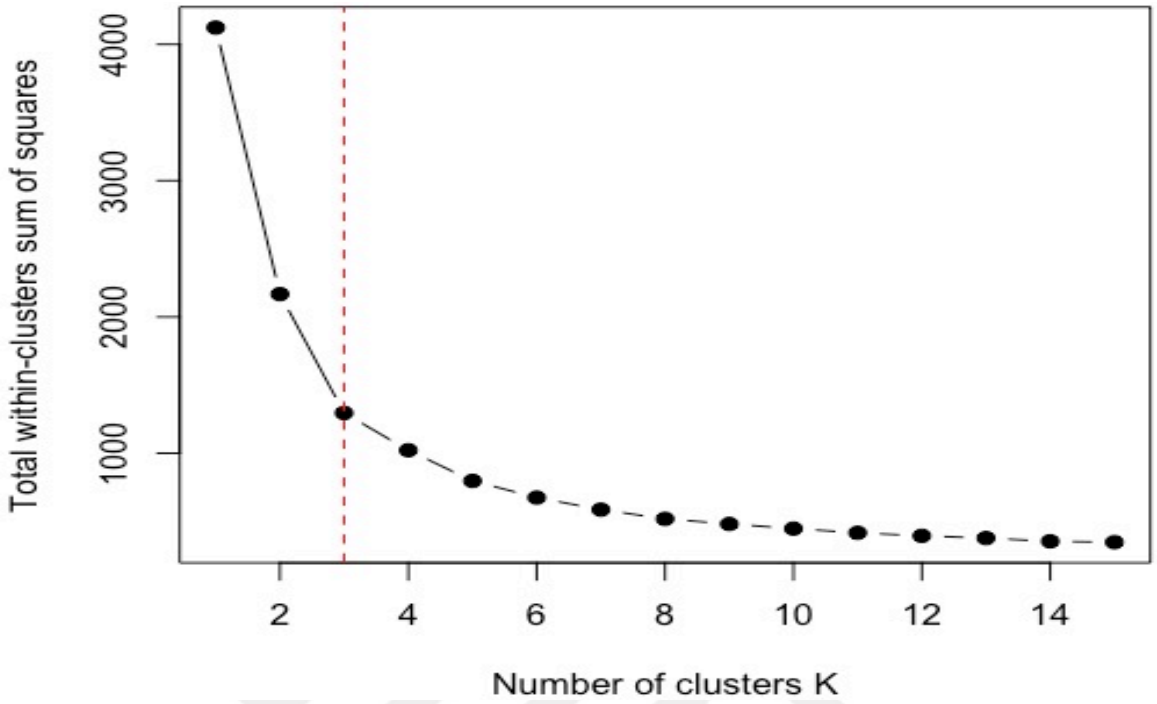
Şekil 7.7 Tez veri kümesi R özeti

Elbow Yöntemi ve Gap istatistik değeri sonuçları aşağıdaki tabloda verilmiştir (Çizelge 7.9).

Çizelge 7.9 Z-skor normalizasyonu uygulanan veri kümesi için doğru küme sayısı sonuçları

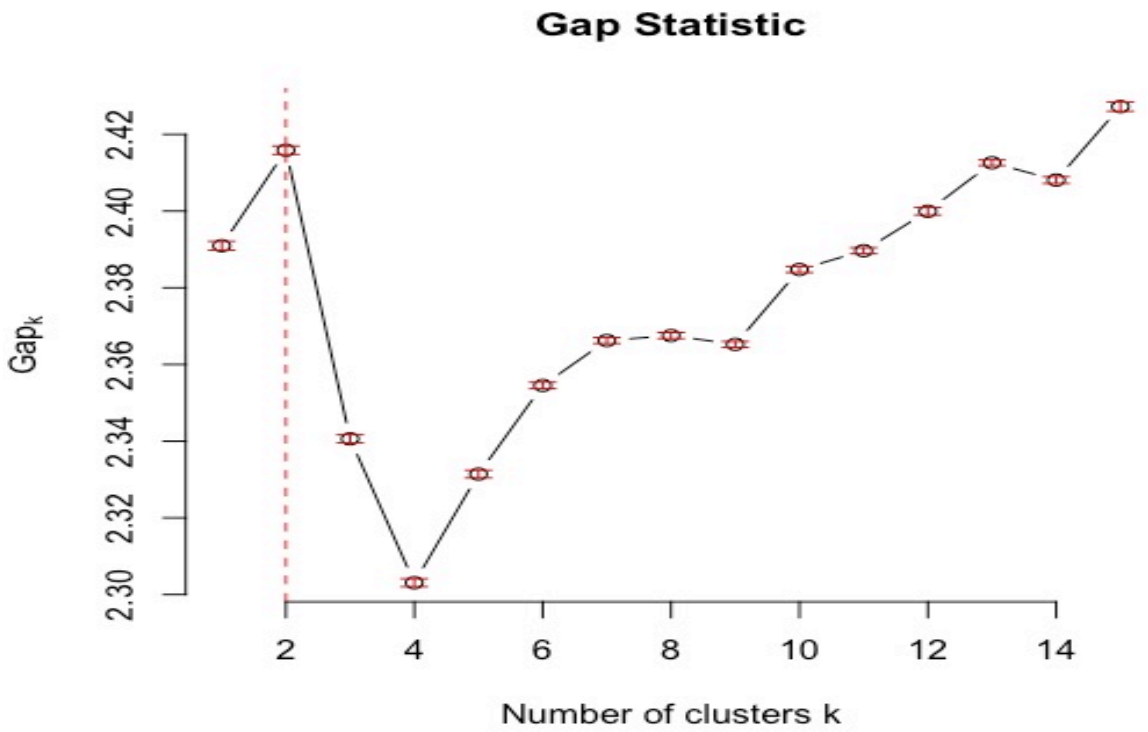
k	Elbow Yöntemi	Gap İstatistik Yöntemi
	WSS	Gap İstatistik Değeri
2	2167.0768	2.415874
3	1294.7466	2.340602
4	1021.7722	2.303074
5	796.6035	2.331411
6	674.211	2.354524
7	585.6983	2.366241
8	518.0442	2.367537
9	480.6834	2.365244
10	446.6645	2.384760
11	415.8023	2.389659
12	393.5827	2.399954
13	376.8359	2.412596
14	353.4884	2.408091
15	346.9353	2.427244

Elbow yöntemi ile çizdirilen grafik sonucunda optimal k değerinin 3 olduğu belirlenmiştir. Grafik aşağıdaki gibidir.



Şekil 7.8 Doğru küme sayısı grafiği

Gap istatik değeri ile optimal k değerini seçerken iken ilk en büyük (first max) yöntemi kullanılmıştır. Gap istatistik değeri ile hesaplanan k değerinin 2 olduğu görülmektedir. Grafiği aşağıdaki gibidir (Şekil 7.8).



Şekil 7.9 Doğru küme sayısı grafiği

Bu veri kümesi için k değerleri 2 ve 3 olarak belirlenmiştir. Bundan sonraki yapılacak olan kümeleme değerlendirme kriterleri ve hata hesaplarında bu durum için k = 2 ve k = 3 için sonuçlar elde edilmiştir.

Kümeleme değerlendirme kriterleri ile elde edilen sonuçlar aşağıdaki tabloda verilmiştir. Hesaplanan değerleri göre optimal k seçilmesi gereken değerleri siyah ile gösterilmiştir (Çizelge 7.10).

Çizelge 7.10 Z-skor normalizasyonu veri kümesi için kümeleme değerlendirme kriterleri

k	Calinski – Harabasz	Davies – Bouldin	Dunn	Silhouette	Wemmert – Gançarski
2	**12740.32**	1.6124222	0.000189631	0.2253273	0.3798828
3	11301.68	**0.7847963**	**0.000340190**	**0.2503610**	**0.3902804**

Yukarıdaki tabloda belirlenen değerleri göre klasik veri madenciliği yöntemleri ile kümeleme sonucu ile elde edilen hata matrisi aşağıdaki gibidir (Çizelge 7.11).

Çizelge 7.11 Z-skor normalizasyonu uygulanan veri kümesi için hata sonuçları

k	Ortalama Karesel Hatanın Karekökü (RMSE)	Ortalama Karesel Hata(MSE)
2	0.04708237	0.002216749
3	**0.04706313**	0.002214939

7.7 Sonuçların Yorumlanması ve Öneriler

Ön işlem uygulanmamış veri kümesi için k=2 ve k=3 değerleri için kümeleme değerlendirme kriterleri ile 5 yöntemden 4 tanesi optimal k değerinin 2 olduğunu göstermektedir (Çizelge 7.4). Yapılan hata hesabı ile bulunan **0.04706028** değeri en düşük hatanın k=3 olduğu durumda gerçekleştiğini göstermektedir (Çizelge 7.5). Bu durum için Calinski – Harabasz kümeleme değerlendirme kriteri ile en az hataya sahip durumun meydana geldiği sonucuna varılmıştır.

Min - Max normalizasyonu uygulanan veri kümesi için $k = \{5,6,7,8,9,10\}$ değerleri alınarak kümeleme değerlendirme kriterleri ile 5 yöntemden 4 tanesi optimal k değerinin 5 olduğunu göstermektedir (Çizelge 7.7). Yapılan hata hesabı ile bulunan **0.04710647** değeri en düşük hatanın k=5 olduğu durumda gerçekleştiğini göstermektedir (Çizelge 7.8). Bu durum için Calinski – Harabasz, Dunn, Silhouette ve Wemmert – Gançarski kümeleme değerlendirme kriterleri ile en az hataya sahip durumun meydana geldiği sonucuna varılmıştır.

Z-score normalizasyonu uygulanmış veri kümesi için k=2 ve k=3 değerleri için kümeleme değerlendirme kriterleri ile 5 yöntemden 4 tanesi optimal k değerinin 3 olduğunu göstermektedir (Çizelge 7.10). Yapılan hata hesabı ile bulunan **0.04706313** değeri en düşük hatanın k =3 olduğu durumda gerçekleştiğini göstermektedir (Çizelge 7.11). Bu durum için Davies - Bouldin, Dunn, Silhouette ve Wemmert – Gançarski kümeleme değerlendirme kriteri ile en az hataya sahip durumun meydana geldiği sonucuna varılmıştır.

Kümeler arası uzaklık ve toplam uzaklık oranı ile belirlenen doğru küme sayısı yöntemi ile önişlem uygulanmamış veri kümesi için k değeri 4, min – max normalizasyonu uygulanan veri kümesi için k değeri 5, z – skor normalizasyonu uygulanan veri kümesi için k değeri 23 olarak hesaplanmıştır (Çizelge 7.1). Yapılan hata hesabı ile en düşük hata **0.047027580** oranı olarak hesaplanmıştır (Çizelge 7.2).

En az hataya sahip durumun herhangi bir ön işlem uygulanmamış veri kümesi için kümeler arası uzaklık ile toplam uzaklık yöntemi kullanılarak belirlenen yöntemle k=4 seçilmesi durumunda ortaya çıkmıştır. Bu duruma ait hata değeri **0. 0.047027580'**dir.

Elbow ve Gap istatistik değeri yardımıyla hesaplanan k değerlerinin hataları kıyaslandığında (Çizelge 7.5 ve Çizelge 7.8 ve Çizelge 7.11) en düşük hata değeri **0.04706028** olarak hesaplanmıştır. Bu durum k=3 değeri için önişlem uygulanmamış veri kümesinde hesaplanmıştır. Hesaplanan tüm durumlar için elde edilen sonuçlar Şekil 7.10 da gösterilmiştir.

	#Durum	Veri Kümesi	Doğru k Belirleme Yöntemi	Doğru *k* Değeri	RMSE	En Düşük Hata
Geleneksel Yöntem	1	Geleneksel Dönüşüm Oranı			0.04848399	*
KÜMELEME	2	Dönüşüm Uygulanmamış	Elbow	3	0.04706028	*
	3	Dönüşüm Uygulanmamış	Gap	2	0.04708016	
	4	Z- Skor Dönüşümü	Elbow	3	0.04708237	
	5	Z- Skor Dönüşümü	Gap	2	0.04706313	*
	6	Min – Max Dönüşümü	Elbow	5	0.04710647	*
	7	Min – Max Dönüşümü	Gap	10	0.04711214	
	8	Dönüşüm Uygulanmamış	Kümeler Arası Uzaklık Oranı	4	0.04702758	*
	9	Z- Skor Dönüşümü	Kümeler Arası Uzaklık Oranı	23	0.047107180	
	10	Min – Max Dönüşümü	Kümeler Arası Uzaklık Oranı	5	0.047110230	

Şekil 7.10 Tüm modellemelerin hata hesabı sonucu

Önerilen yöntem ile minumum hataya sahip olan diğer doğru küme sayı belirlenerek yapılan kümeleme yöntemlerinin sonuçları kıyaslandığında fark **%0.7'**lik bir iyileştirme gerçekleştirilmiştir.

Sonuç olarak, geleneksel yöntem ile yapılan optimizasyonunun (Bölüm 7.2) **0.04848399**'luk hata oranı ile doğru küme saysı belirlenerek yapılan kümeleme yöntemlerindeki minimum hata oranı (Çizelge 7.5) **0.04706028** arasındaki fark **%3.1** dir. Dolayısıyla reservasyon optimizasyonunda kümeleme yöntemleri kullanılabilir ve kullanılan veri kümesinde **% 3.1** kar edilmesini sağlamıştır. En düşük hataya sahip modellemelerin özeti Şekil 7.11 de gösterilmiştir.

Şekil 7.11 En düşük hataya sahip modellerin özeti

KAYNAKLAR

[1] Arabacı, G., (2007). "Veri madenciliğinde appriori, tahminci appriori ve tertius algoritmalarının weka ve yale programları ile karşılaştırılması ve bir uygulama", Yüksek Lisans Tezi, İstanbul Ticaret Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul

[2] Larose, D.T. (2005). Discovering Knowledge in Data: An Introduction to Data Mining. Wiley Interscience A. John Wiley & Sons Inc., Canada: 214

[3] Silahtaroğlu, G., (2016). Veri madenciliği: Kavram ve algoritmaları. 3'üncü Basım. İstanbul: Papatya Yayıncılık.

[4] Ergün, E., (2008). "Ürün kategorileri arasındaki satış ilişkisinin birliktelik kuralları ve kümeleme analizi ile belirlenmesi ve perakende sektöründe bir uygulama", Doktora Tezi, Afyon Kocatepe Üniversitesi, Sosyal Bilimler Enstitüsü, Afyonkarahisar

[5] Han and Kanber(2006). Data Mining:Concepts and Techniques, 2. The Morgan Kaufmann Series in data Management Systems,Jim Gray, Series Editor Morgan Kaufmann Publishers.

[6} Altunkaynak B.,(2017), Veri Madenciliği Yöntemleri ve R Uygulamaları 1'inci Basım:Ankara,Seçkin Yayıncılık

[7] Alpaydin, E(2014). Introduction to machine learning, 2 ed. MIT press: 5-9.

[8] Rajaraman, A., and Ullman, J. D. (2011). Mining of massive datasets. Cambridge University Press: 2-4.

[9] Alpaydin, E.(2014) Introduction to machine learning, 2 ed. MIT press,:11-26.

[10] Al Shalabi, L., Shaaban, Z., & Kasasbeh, B. (2006). Data mining: A preprocessing engine. Journal of Computer Science, *2*(9): 735-739.

[11] Jia, H., Cheung, Y. M., & Liu, J. (2016). A new distance metric for unsupervised learning of categorical data. IEEE transactions on neural networks and learning systems, *27*(5): 1065-1079.

[12] Ilango, V., Subramanian, R., & Vasudevan, V. (2012). A five step procedure for outlier analysis in data mining. European Journal of Scientific Research, 75(3): 327-339.

[13] Cran R Project, BBmisc Documentation, ftp://cran.r-project.org/pub/R/web/packages/BBmisc/BBmisc.pdf, 10 Mart 2017

[14] Cheadle, C., Vawter, M. P., Freed, W. J., & Becker, K. G. (2003). Analysis of microarray data using Z score transformation. The Journal of molecular diagnostics, 5(2): 73-81.

[15] Aggarwal, C. C. (2015). Outlier analysis. In Data mining (pp. 237-263). Springer, Cham.

[16] Aggarwal, C. C. (2013). An introduction to outlier analysis. In Outlier analysis (pp. 1-40). Springer, New York, NY.

[17] Williamson, D. F., Parker, R. A., & Kendrick, J. S. (1989). The box plot: a simple visual method to interpret data. Annals of internal medicine, 110(11): 916-921.

[18] Singh, A., Yadav, A., & Rana, A. (2013). K-means with Three different Distance Metrics. International Journal of Computer Applications, 67(10).

[19] Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8): 651-666.

[20] Pham, D. T., Dimov, S. S., & Nguyen, C. D. (2005). Selection of K in K-means clustering. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 219(1): 103-119.

[21] Bholowalia, P., & Kumar, A. (2014). EBK-means: A clustering technique based on elbow method and k-means in WSN. International Journal of Computer Applications, 105(9).

[22] Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of Cluster in K-Means Clustering. International Journal, 1(6): 90-95.

[23] Arai, K., & Barakbah, A. R. (2007). Hierarchical K-means: an algorithm for centroids initialization for K-means. Reports of the Faculty of Science and Engineering, 36(1): 25-31.

[24] Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63(2): 411-423.

[25] Mohajer, M., Englmeier, K. H., & Schmid, V. J. (2011). A comparison of Gap statistic definitions with and without logarithm function. *arXiv preprint arXiv:1103.4767*.

[26] Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., & Hornik, K. (2014). Package 'cluster'. Version, 1(4): 6-7.

[27] Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. IEEE transactions on pattern analysis and machine intelligence, 24(7): 881-892.

[28] Ray, S., & Turi, R. H. (1999, December). Determination of number of clusters in k-means clustering and application in colour image segmentation. In Proceedings of the 4th international conference on advances in pattern recognition and digital techniques (pp. 137-143).

[29] Uzun, E , Erdoğan, C , Saygılı, A . (2016). Hiyerarşik Kümeleme Modeli Kullanan Web Tabanlı Bir Ödev Değerlendirme Sistemi. Ejovoc (Electronic Journal of Vocational Colleges), 6 (3), 87-98. Retrieved from http://dergipark.gov.tr/ejovoc/issue/36634/417046 [32]Çelebi, M.E., Kingravi, H.A. ve Vela, P.A., (2013). "A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm", Expert Systems with Applications, 40(1): 200–210.

[30] Salvador, S., & Chan, P. (2004, November). Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on (pp. 576-584). IEEE.

[31] Alpaydin, E.(2014) Introduction to machine learning, 2 ed. MIT press: 146-148.

[32] Iris Veri Kümesi, https://archive.ics.uci.edu/ml/datasets/iris, 12 Şubat 2018.

[33] Hämäläinen, J., Jauhiainen, S., & Kärkkäinen, T. (2017). Comparison of Internal Clustering Validation Indices for Prototype-Based Clustering. Algorithms, 10(3): 105.

[34] Dudoit, S. & Fridlyand, J. (2002) A prediction-based resampling method for estimating the number of clusters in a dataset. Genome Biology, 3(7): 0036.1-21.

[35] Dimitriadou, E., Dolnicar, S., & Weingessel, A. (2002) An examination of indexes for determining the Number of Cluster in binary data sets. Psychometrika, 67(1): 137-160.

[36] Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001) On Clustering Validation Techniques. Intelligent Information Systems Journal, 17(2-3): 107-145.

[37] Kaufman, L. & Rousseeuw, P. J. (1990) Finding Groups in Data: An Introduction to Cluster Analysis. New York, John Wiley & Sons.

[38] Forestier, G., Gancarski, P., & Wemmert, C. (2010). Collaborative clustering with background knowledge. Data & Knowledge Engineering, 69(2): 211-228.

[39] Elbow Method (Clustering),

https://en.wikipedia.org/wiki/Elbow_method_(clustering), 22 Ocak 2018.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Kemal Koşuta

Doğum Tarihi ve Yeri : 08.01.1992- İstanbul

Yabancı Dili : İngilizce

E-posta : kosuta@yildiz.edu.tr

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Lisans	Matematik Müh	Yıldız Teknik Üniversitesi	2015
Lise	Fen Bilimleri	Bahçelievler And. Lisesi	2010

İŞ TECRÜBESİ

Yıl	**Firma/Kurum**	**Görevi**
2017-Halen	Yıldız Teknik Üniversitesi	Araştırma Görevlisi
2015-2017	MetGlobal	Veri Bilimi Uzmanı

YAYINLARI

Bildiri

1. Şaylı A., Koşuta K.,"K – Ortalamalar Algoritmasına Dayalı Dönüşüm Oranı Tahmini İle Rezervasyon Optimizasyonu", 37. Yöneylem Araştırması Ve Endüstri Mühendisliği Ulusal Kongresi, İstanbul, Türkiye, 5-7 Temmuz 2017, İstanbul