K-MEANS VE HİYERARŞİK KÜMELEME ALGORİTMANIN WEKA VE MATLAB PLATFORMLARINDA KARŞILAŞTIRILMASI

Arş. Gör. Mustafa TAKAOĞLU
İstanbul Aydın Üniversitesi, Türkiye
mustafatakaoglu@aydin.edu.tr
https://orcid.org/0000-0002-1634-2705

Faruk TAKAOĞLU
TÜBİTAK, BİLGEM, Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü,
Türkiye faruk.takaoglu@tubitak.gov.tr
https://orcid.org/0000-0003-0828-2017

ÖZ

Günümüz teknolojik gelişmeleri ve gelinen nokta ele alındığında, veri tabanı yönetimi ve veri madenciliğinin büyük önem kazandığı görülmektedir. Geçmişte veri yığınlarının kapladıkları alan ve depolama masrafları firmalar için gereksiz bir masraf olarak görülmekteydi. Günümüzde ise veri yığınlarının işlenmesi ve yorumlanması sayesinde elde edilen kazanımlar, irili ufaklı tüm resmi ve özel sektör kurum ve kuruluşlarının ilgisini çekmektedir. Bu sebeple günümüzün popüler çalışma alanlarından biri olan veri madenciliğinin önemi artmaktadır. Veri madenciliğinin sıkça kullanılan yöntemlerinden biri olan kümeleme algoritmaları ise bu alanda çalışma yapacak kişilerin bilgi sahibi olması gereken konulardan biridir. Bu çalışmamızda iki kümeleme algoritması incelenmiştir. K-Means kümeleme algoritması ve Hiyerarşik kümeleme algoritması bu doğrultuda ele alınmıştır. Ele alınan bu algoritmalar Kandilli iklim verileri kullanılarak WEKA ve MATLAB platformlarında teste tabi tutulmuştur. WEKA ve MATLAB platformlarındaki bulgulara göre, her iki yöntemin üstün ve kısıt oluşturan yönleri irdelenmiştir.

Anahtar Kelimeler: *WEKA, MATLAB, K-Means, Hiyerarşik Kümeleme*

Geliş Tarihi: 02.01.2019, Kabul Tarihi: 12.05.2019, DOI NO: 10.17932/IAU.IAUD.m.13091352.2019.3/43.303-317
Araştırma Makalesi - Bu makale Turnitin programıyla kontrol edilmiştir.

COMPARING K-MEANS AND HIERARCHICAL CLUSTERING ALGORITHMS ON MATLAB AND WEKA PLATFORMS

ABSTRACT

Database management systems and data mining have an increasing importance owing to the recent technological developments. In the past, data stacks storages and keeping costs were considered as an unnecessary expenditure for every company. But today data mining has a great importance from the point of view of most of the markets. Because of this situation, data analysts gained an important role in the recent years. Clustering algorithms are the mostly used algorithm types by the data analysts. This algorithm types are required to be learned by every analyst. In this article, K-Means clustering algorithm and Hierarchical clustering algorithm were applied on climate data. These algorithms were tested in MATLAB and WEKA platforms. As a conclusion, the advantages and disadvantages of MATLAB and WEKA for data mining were discussed.

Keywords: *WEKA, MATLAB, K-Means, Hierarchical Clustering*

GİRİŞ

Algoritmalar, günümüzde birçok sektörde doğrudan ya da dolaylı olarak kullanılmaktadır. Her sorunun çözümüne getirilen yaklaşımlar farklı olabilmekle birlikte, her yaklaşım doğru sonuca ulaşabilir. Burada algoritmanın kullanılacağı spesifik durum önemlidir. Örneğin sıralanacak bir veri yığını olduğunu düşünelim. Bu veri yığınının büyüklüğünün, kullanılacak sıralama algoritması seçilirken çok önemli olduğunu söyleyebiliriz. Çünkü bazen çok hızlı çalışan karışık algoritmalar küçük verilerde, çok daha basit bir mantıkla çalışan basit algoritmalara kıyasla daha verimsiz çalışır. Bunun nedeni, algoritmanın kullanılacağı doğru durumun saptanmasının büyük önem taşıyor olmasıdır. Ayrıca Algoritmanın kalitesi, çalışma hızı ve kullandığı bellek ile orantılıdır. Yani bir algoritma ne kadar hızlı çalışıyorsa ve ne kadar az bellek kullanıyorsa, o kadar kaliteli, o kadar verimli bir algoritma demektir. Kümeleme algoritmaları ise sınırlı bir obje grubu üzerinde uygulanan bir sınıflandırma yöntemidir (Dubes ve Jain, 1988). Kümeleme sınıflandırmanın özel bir türüdür (Kendall, 1966). Ayrıca kümelemenin "unsupervised", yani eğiticisiz öğrenme prensibine sahip olduğunu da belirtmek gerekir.

Makalemizde kullanılan verilerimiz, Boğaziçi Üniversitesi Kandilli Rasathanesi ve Deprem Araştırma Enstitüsü'nden alınmış olup, 1952-2015 yılları arası 12 aylık ölçümü yapılmış Ortalama Rüzgâr Şiddeti verilerinden oluşmaktadır.

Makalemizde seçilen rüzgâr şiddeti verileri sadece algoritmaların çalışmasını incelemek amacıyla kullanılmış olup, veriler üzerinden herhangi bir yorum çıkarma amacı ile çalışılmamıştır.

K-Means Kümeleme Algoritması

K-Means algoritması, kümeleme algoritmalarının içinde belki de en eski, en çok kullanılan ve bir o kadar da basit bir algoritmadır. "Unsupervised", yani eğiticisiz öğrenme prensibine sahiptir. Avantajları ve dezavantajları vardır, ancak büyük verilerdeki hızlı çalışması sebebiyle tartışılmaz en popüler algoritmalardan biridir. Eski bir algoritma denilmesinin sebebi, ilk kez K-Means isminin 1967 yılında J. B. MacQueen tarafından kullanılmış olmasıdır. K-Means algoritmasının mantığı 1957 yılı Hugo Steinhaus'un yaptığı çalışmalara dayanmaktadır (Steinhaus, 1957).

K-Means algoritmasında, kümelenecek verilerden her biri sadece bir kümenin elemanı olabilir. Bu kümelerin temsil edildiği noktalara ise merkez noktası denir. Dezavantaj olarak söyleyebileceğimiz belki de en önemli husus, algoritmanın kulanılacak verinin bölüneceği küme sayısını, kullanıcının girmesine bağlı olarak belirlemesi durumudur. Bu sebeple doğru küme sayısı belirlenene kadar deneme yanılma yöntemine başvurulması gerekebilir. K-Means işleminin başarıyla tamamlanması için bazen birkaç kez fonksiyonun çağırılması gerekebilir. Çünkü ilk seferde oluşan kümelerin içindeki benzerlik uyumu tutmayabilir. Fonksiyonun birkaç tekrardan sonra, kümelerdeki değişimin durması, elde edilen kümelerde istenilen sonucun alındığı anlamına gelir. Bir diğer dezavantaj da gürültülü verinin kullanımıdır. Kümeleme esnasında benzer veriler seçilirken, verideki gürültü gibi etkenler dikkate alınmaz.

K-Means algoritmasının formülsel ifadesi için aşağıdaki denklemlerden yararlanılır:

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{ci} (||x_i - v_j||)^2$$

Burada; '$||x_i - v_j||$', x ve y arasındaki öklid mesafesi, 'c_i', i^{th} kümesindeki veri noktalarının sayısı, c ise küme merkezlerinin sayısıdır.

K-Means Kümelemenin Algoritmik Adımları:

$X = \{x_1, x_2, x_3, \ldots, x_n\}$ kümesi veri noktalarının, $V = \{v_1, v_2, \ldots\ldots, v_c\}$ ise merkez noktalarının kümesi olsun.

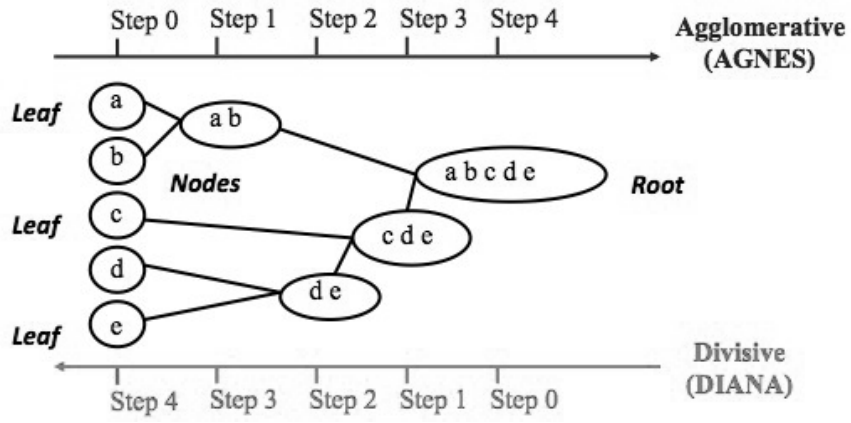
1) Rastgele 'c' küme merkezlerini seç.
2) Her veri ile küme merkezlerinin arasındaki mesafeyi hesapla.
3) Küme merkeziyle arasındaki mesafe, diğer küme merkezleriyle olan mesafeden daha az olan veriyi, yakın olan o küme merkezine ata.
4) Yeni küme merkezini aşağıdaki denklemle yeniden hesapla:

$$v_i = (1 / c_i) \sum_{j=1}^{Ci} x_i$$

5) Her verir noktasıyla, yeni küme merkezleri arasındaki mesafeyi yeniden hesapla.
6) Eğer hiçbir veri noktası atanmadıysa dur, diğer durumda üçüncü adımdan itibaren tekrar et.

Hiyerarşik Kümeleme Algoritması

Hiyerarşik algoritmalar iki başlık altında incelenirler. Bunlar AGNES (Agglomerative Nesting), yani Aglomerativ Kümeleme ve DIANA (Divise Analysis), yani Bölücü Hiyerarşik Kümeleme'dir. AGNES'de aşağıdan yukarıya doğru bir kümeleme mantığı vardır. Verilerin her biri başlangıç aşamasında birer küme olarak kabul edilir ve bunlar arasından en benzer olan ikililer kümelenir. Bu işlem kümelenecek başka bir veri kalmayıncaya kadar devam eder. Sonuç ağacı ise dendrogramda gösterilir. Şekil 1 'de AGNES ve DIANA kümeleme çalışma dendrogramı şematik olarak sunulmaktadır.



Şekil 1: AGNES ve DIANA kümeleme çalışma dendrogramı

DIANA'da ise AGNES'e göre tam ters bir mantık görülür. Yukarıdan aşağıya doğru kümelere böler. Bütün kümelerde tek bir veri kalıncaya kadar bu işlem devam eder. Hiyerarşik kümelemede, kümeler arasındaki benzerlikler ve yakınlık, farklı yöntemlerle belirlenebilir.

Bunlar; tam bağlantılı kümeleme, tek bağlantılı kümeleme, ortalama bağlantılı kümeleme, Ward'ın minimum varyans yöntemidir.

Yukarıdaki yöntemler çalışırken, benzer olarak tüm verileri birer küme olarak kabul eder ve hepsini bir kümenin içindeymiş gibi ele alır. Daha sonra bu kümelerden birbirine en benzemeyen iki veriyi, yani alt kümeyi seçer ve bunların diğer veriler ile aralarındaki mesafeyi ökçerek bir ilişki kurar. Tam bağlantılı kümelemede bu kıyaslama birinci ve ikinci dendrogramda incelediğimiz aşamalarda algoritmamız tüm ikililerin farklılıklarını hesaplar ve bu farklılıkları iki küme arasındaki mesafe olarak kullanır. Tek bağlantılı kümeleme algoritması ise birinci ve ikinci aşamadaki kümelerin ikililerinin farklılıklarını hesaplayıp, bunların en ufak farklılığını bağlantı kriteri olarak belirleyerek çalışır. Ortalama bağlantılı kümeleme algoritması, birinci ve ikinci aşamadaki kümelerin ikililerinin farklılıklarını hesaplayıp, bunların ortalama farklılığını iki küme arasındaki mesafe olarak kabul ederek çalışır. Ward'ın yönteminde ise küme içi varyans minimize edilir. Her adımda mesafesi en az olan iki küme birleştirilir.

WEKA

Günümüzde üzerine birçok araştırma yapılan WEKA, Yeni Zelanda'nın Waikato Üniversitesi tarafınca ücretsiz GNU lisansı ile üretilmiş modüler bir veri madenciliği uygulamasıdır. Bünyesinde birçok method, algoritma, hazır fonksiyon ve kütüphane bulunmaktadır. Modüler özelliği sayesinde yeni geliştirilen ya da standart programla birlikte gelmeyen birçok özellik; fonksiyon, algoritma, method vb. gibi kullanıcının isteği doğrultusunda WEKA platformundan ücretsiz bir şekilde indirilip programa entegre edilerek kullanılabilir.

WEKA programı geliştirilme aşamasında, Java programlama dili kullanılarak üretildiği için kütüphaneleri .jar uzantılıdır. Bu, kullanıcılara büyük kolaylıklar sağlamaktadır. Uzantısının .jar olması, Java ile üretilmiş birçok programın projeye entegre işleminde büyük kolaylıklar sağlar.

Veri madenciliği aşamasında WEKA platformunda; sınıflandırma, kümeleme, ilişkilendirme, veri ön işleme ve görselleme işlemleri kolayca yapılabilmektedir. Bu işlemlerin yapılabilmesi için, kullanılacak verilerin uzantısının arff olması gerekmektedir. Ayrıca farklı uzantılardaki verilerin dönüşümü kolayca yapılabilmektedir.

Programın ihtiyaç duyduğu arff uzantılı dosyanın kullanıcı tarafından hazırlanacağını düşünürsek, kullanılacak dosyayı şu şekilde hazırlamamız gerekir. Öncelikle not defteri programında bir sayfa açıp, @relation yazılıp yanına dosyamızın kullanılacağı iş ile ilgili bir isim verilir. Daha sonra, @ attribute yazılarak yanına özelliğin ifade ettiği kavramın ismi yazılır. Yanına

uzantısı belirtilir. Bu uzantı; numeric, nominal, real, string veya date formatında olabilir. Bu özellikleri tanıttıktan sonra tanıtılan özelliklere değer girilir. Bunu yapabilmek için öncelikle, @data yazılır ve altına aralarında virgül olacak şekilde belirtilen uzantıya uygun veriler girilir. Her satırda girilecek veriler bittiğinde, bir alt satıra geçip, sırayla veri girme işlemi elimizdeki tüm verilerin girilmesi bitene kadar tekrar eder.

Ufak çaplı bir uygulama yapılacağı zaman not defterinde kolayca arff uzantılı dosyamızı hazırlayabiliriz. Bu dosyayı kaydederken formatını arff yaparak kolayca işlemimizi tamamlamış oluruz. Aşağıdaki Çizelge 1'de örnek bir arff dosyası görebilirsiniz.

Tablo 1: Örnek Arff dosyası

@RELATION growth
@ATTRIBUTE year NUMERIC
@ATTRIBUTE exportdollar NUMERIC
@ATTRIBUTE importdollar NUMERIC
@ATTRIBUTE population NUMERIC
@ATTRIBUTE gsyh NUMERIC
@ATTRIBUTE debt NUMERIC
@ATTRIBUTE unemployment NUMERIC
@ATTRIBUTE uneducatedpeople NUMERIC
@ATTRIBUTE growthratio NUMERIC
@DATA
2007,107271749904,170062714501,70586256,8430000000,250300000000,10.3,5347461,4.7
2008,132027195626,201963574109,71517100,9510000000,281000000000,11,4930012,0.7
2009,102142612603,140928421211,72561312,9530000000,269200000000,14,4672257,-4.8
2010,113883219184,185544331852,73722988,10990000000,291900000000,11.9,3825644,9.2
2011,134906868830,240841676274,74724269,12980000000,304200000000,9.8,3171270,8.5

Yukarıdaki Çizelge 1'de gördüğünüz veriler, lisans eğitimimde Türkiye ile ilgili bazı veriler üzerinden, büyüme oranları hakkında forecasting formulünü kullanarak tahmin yaptığım çalışmamın verileridir. Arff dosyası, yukarıdaki kurallara uyarak kolayca hazırlanabilir.

MATLAB

MATLAB, MathWorks tarafından geliştirilen, dördüncü nesil, ücretli bir programlama dilidir. İstatistiksel, matematiksel, simulasyon ve birçok işlemlerin yapılabildiği Matlab, C, C++, Java, Fortran, Python gibi programlama dillerine uyumlu olarak çalışabilmektedir. Bu sebeple geniş bir kullanım alanı vardır. Özellikle endüstriyel ve akademik alanlarda çalışan kişilerin çokça faydalandığı bir uygulamadır.

Çalışmamızda, belirlediğimiz iki kümeleme algoritmasını analiz etmek amacıyla Matlab programı tercih edilmiştir. Kullanımı kolay, anlaşılır ve hızlı çalışan bir uygulama olan MATLAB sayesinde algoritmalarımızın kodlarını kolayca çalıştırıp elde edilen sonuçlar analiz bölümünde verilmiştir. Elde edilen sonuçlar görsellik açısından çok zengin olup, bu görsellerin oluşturulması işlemi sebebiyle programın çalışma süresi biraz uzundur.

ANALİZ

WEKA'da K-Means Kümeleme Algoritması Analizi: Bu bölümde, ele alınan K-Means kümeleme algoritmasının, WEKA'da kullanılması sonucu elde edilen sonuçları açıklanmıştır. K-Means kümeleme algoritmasında kullanılan Kandilli Ortalama Rüzgâr Şiddeti veri kümemiz Giriş bölümünde tanıtılmıştır. Aşağıdaki sonuçlara ulaşılmasında ortak bazı adımlar bulunmaktadır. Bunlar; WEKA platformunun çalıştırılması ve Explorer uygulamasının seçilmesi, kullanılacak verinin açılması ve kümeleme algoritmaları sekmesinin kümeleme algoritmalarından kullanacağımız K-Means algoritmasının seçilip, üzerinde gerekli değişikliklerin yapılması gibi adımlardır.

Çalışmamızda, küme sayısı, k değeri, veri kümemizin işlemlerinde 2›den 10›a kadar alınıp, her k değeri için 1›den 50›ye kadar seed değeri denenerek yapılmıştır. Veriler arası mesafenin hesaplanması için aşağıda açıklanan Öklid fonksiyonu kullanılmıştır.

Öklid mesafesinin bulunmasında aşağıda gösterilen formül kullanılmaktadır. Bu formül, i ve j noktalarının arasındaki uzaklığı bulmamızı sağlar. Burada p değişkeni, p boyutlu bir uzay demektir ve k değişken indeksidir.

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

Ayrıca formulün sağlanması için aşağıdaki denklemlerdeki şartlar aranmaktadır.

$d(i,j) \geq 0$
$d(i,i) = 0$
$d(i,j) = d(j,i)$
$d(i,j) \leq d(i,k) + d(k,j)$

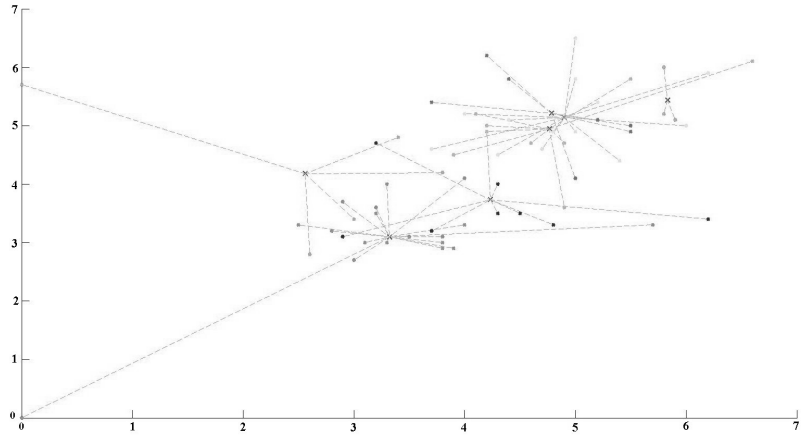
Kandilli Ortalama Rüzgâr Şiddeti Verileri WEKA K-Means Analizi: Bu veri kümesinin K-Means kümeleme algoritmasına sokulması sonucu elde edilen en verimli sonuç aşağıda açıklanmıştır. Veriler arası mesafenin ölçümü için Öklid fonksiyonu kullanılmıştır. Toplamda yapılam 450 denemeden Hata Kare Toplamı en düşük, yani en başarılı deneme seçilmiştir. K değeri 10 ve seed değeri 44 olan kümelemede en verimli sonuca ulaşılmıştır. Algoritma bu sonuca ulaşmak için 6 kez tekrar etmiştir. Hata Kare Toplamı 8.160678975344503 olarak hesaplanmıştır. Aşağıda Çizelge 2'de WEKA'nın çalışması sonucu elde edilen sonuç verilmiştir.

Tablo 2: K Means Kandilli Ortalama Rüzgâr Şiddeti analiz çizelgesi

=== Run information ===		
Scheme: weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 10 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 44		
Relation: ortruzsid		
Instances: 64		
Attributes: 13		
yıl, ocak, şubat, mart, nisan, mayıs, haziran, temmuz, ağustos, eylül, ekim, kasım, aralık		
Test mode: evaluate on training data		
=== Clustering model (full training set) ===		
kMeans		
======		
Number of iterations: 6		
Within cluster sum of squared errors: 8.160678975344503		
Time taken to build model (full training data) : 0 seconds		
=== Model and evaluation on training set ===		
Clustered Instances		
0 7 (11%)	4 1 (2%)	8 14 (22%)
1 4 (6%)	5 4 (6%)	9 8 (13%)
2 6 (9%)	6 11 (17%)	
3 3 (5%)	7 6 (9%)	

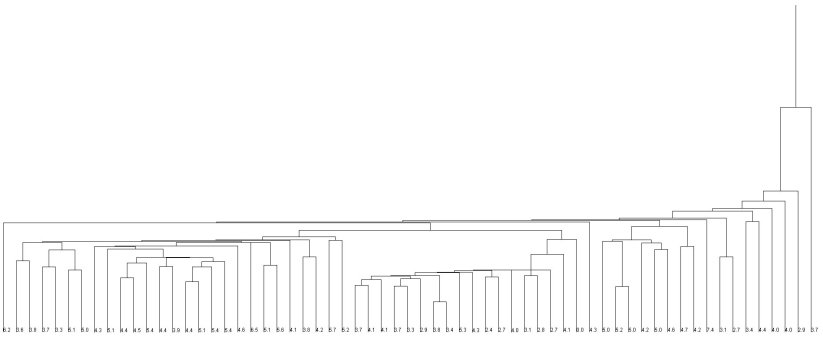
MATLAB'da K-Means Kümeleme Algoritması Analizi: Bu bölümde, ele aldığımız kümeleme algoritması olan K-Means kümeleme algoritmasının MATLAB platformunda kullanılması sonucu elde edilen sonuç açıklanmıştır. Veri olarak Giriş bölümünde belirtilen Kandilli'den alınmış Ortalama Rüzgâr Şiddeti veri kümesi üzerinde çalışılmıştır. MATLAB platformunun en önemli özelliklerinden biri olan görselliği sayesinde elde edilen sonuç görsellik açısından çalışmamıza zenginlik katmış ve algoritmanın nasıl çalıştığını görmemiz açısından büyük katkı sağlamıştır.

Kandilli Ortalama Rüzgâr Şiddeti Verileri MATLAB K-Means Analizi: Kandilli Ortalama Rüzgâr Şiddeti veri kümesinin MATLAB platformunda K-Means kümeleme algoritmasına sokulması sonucu elde edilen en verimli sonuç Şekil 2 'de verilmiştir. Veriler arası mesafenin ölçümü için daha önce açıklanan Öklid fonksiyonu kullanılmıştır. Kümeleme işlemi sonucunda k değeri 7 olan küme en başarılı sonucu vermiştir. Bu sonuca ulaşmak için algoritma 7 kez tekrar etmiştir. İşlemin tamamlanması için 13.52 sn zaman harcanmıştır. Hata Kare Toplamı 12.3285684518 olarak hesaplanmıştır.

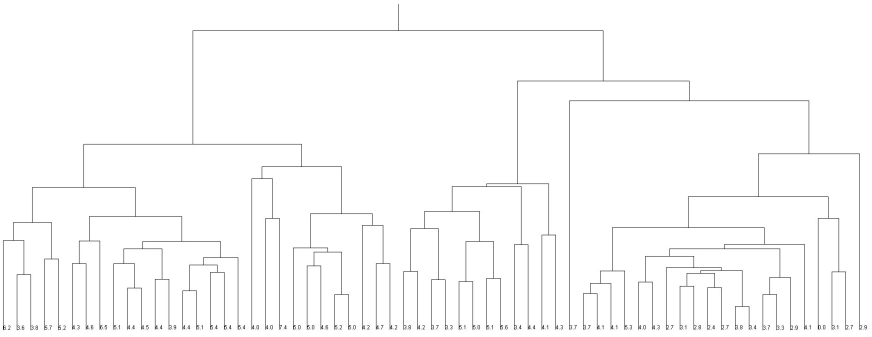


Şekil 2: Kandilli Ortalama Rüzgâr Şiddeti verileri K-Means kümeleme algoritması MATLAB sonucu

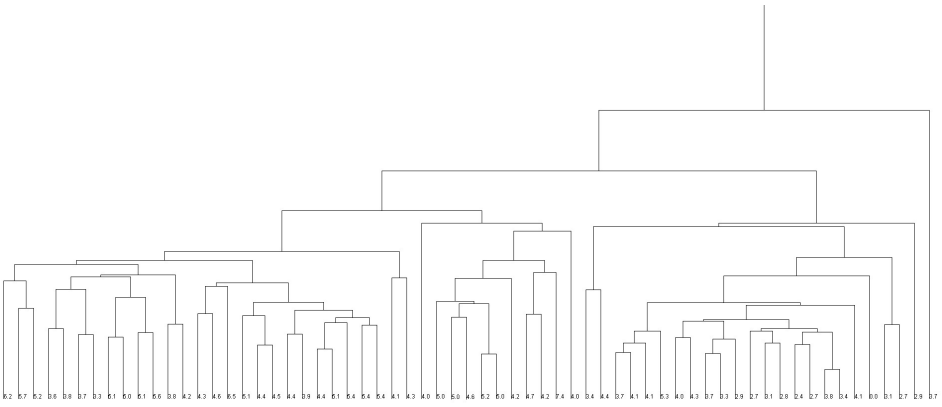
WEKA'da Hiyerarşik Kümeleme Algoritması Analizi*:* Bu bölümde Kandilli Ortalama Rüzgâr Şiddeti veri kümemiz, Hiyerarşik kümeleme algoritmaları olan Tek, Tam, Ward ve Ortalama Hiyerarşik kümeleme algoritmalarına sokulup elde edilen sonuçlar aşağıda gösterilmiştir. Veriler arasındaki mesafe Öklid fonksiyonuyla hesaplanmıştır. Öklid fonksiyonu çalışmamızın önceki bölümlerinde ayrıntılı olarak açıklanmıştır. Küme sayısı tüm verilerde 2 olarak seçilmiştir. Elde edilen dendrogramlar sırasıyla tek, tam, ortalama ve ward, Şekil 3-6 aralığında verilmiştir.



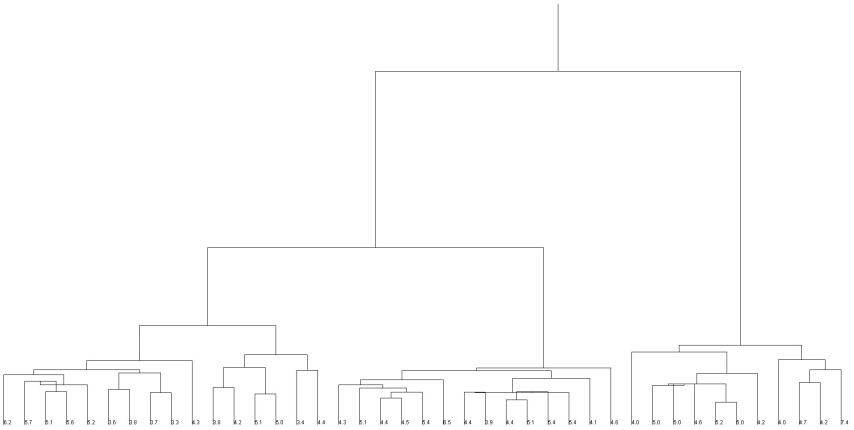
Şekil 3: WEKA'da hiyerarşik kümeleme algoritmasına dayalı olarak Kandilli Ortalama Rüzgâr Şiddeti tek bağlantılı kümeleme dendrogramı



Şekil 4: WEKA'da hiyerarşik kümeleme algoritmasına dayalı olarak Kandilli Ortalama Rüzgâr Şiddeti tam bağlantılı kümeleme dendrogramı

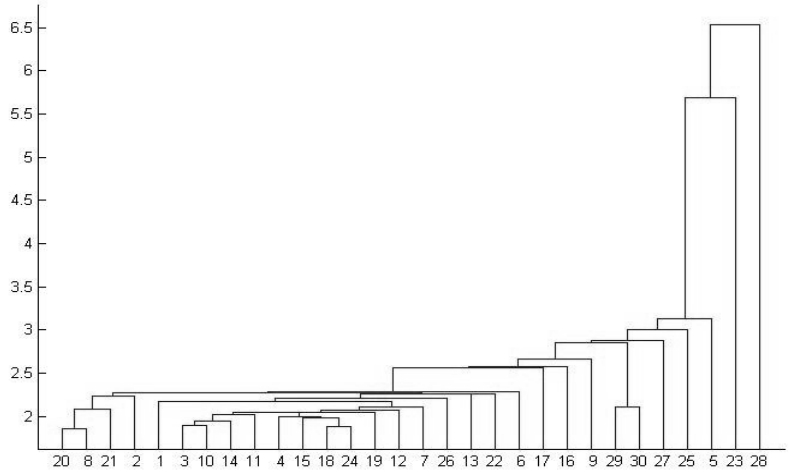


Şekil 5: WEKA'da hiyerarşik kümeleme algoritmasına dayalı olarak Kandilli Ortalama Rüzgâr Şiddeti ortalama bağlantılı kümeleme dendrogramı

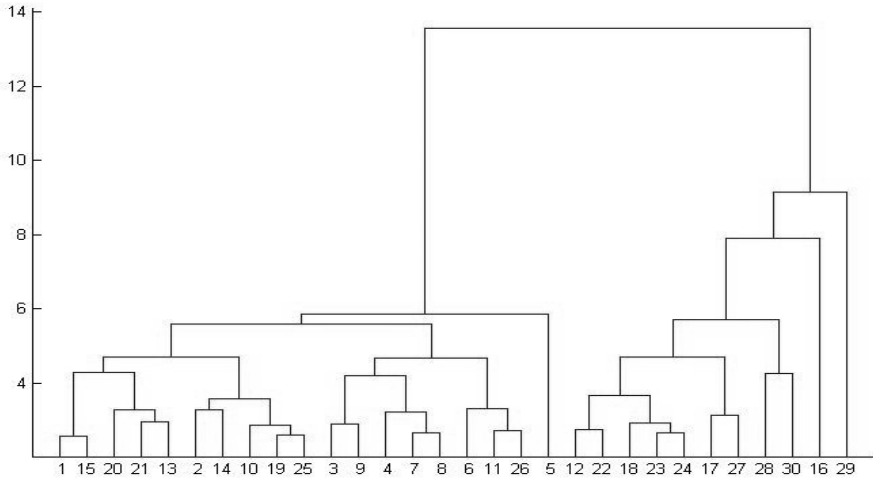


Şekil 6: WEKA'da hiyerarşik kümeleme algoritmasına dayalı olarak Kandilli Ortalama Rüzgâr Şiddeti Ward metodu dendrogramı

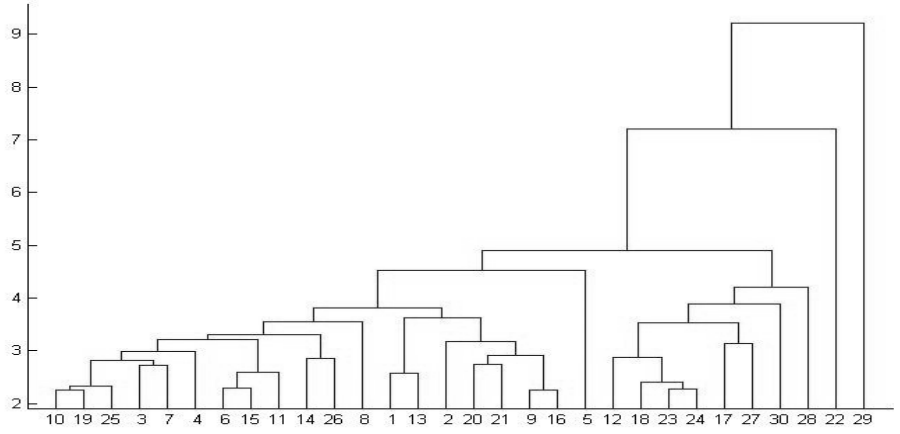
MATLAB'da Hiyerarşik Kümeleme algoritması analizi: Kandilli Ortalama Rüzgâr Şiddeti veri kümemiz, Hiyerarşik kümeleme algoritmaları olan Tek, Tam, Ward ve Ortalama Hiyerarşik kümeleme algoritmalarına MATLAB platformunda sokulup elde edilen sonuçlar aşağıda gösterilmiştir. Veriler arasındaki mesafe Öklid fonksiyonuyla hesaplanmıştır. Öklid fonksiyonu çalışmamızın önceki bölümlerinde ayrıntılı olarak açıklanmıştır. Küme sayısı tüm verilerde 2 olarak seçilmiştir. Elde edilen dendrogramlar sırasıyla Tek, Tam, Ortalama ve Ward, Şekil 7-10 aralığında verilmiştir.



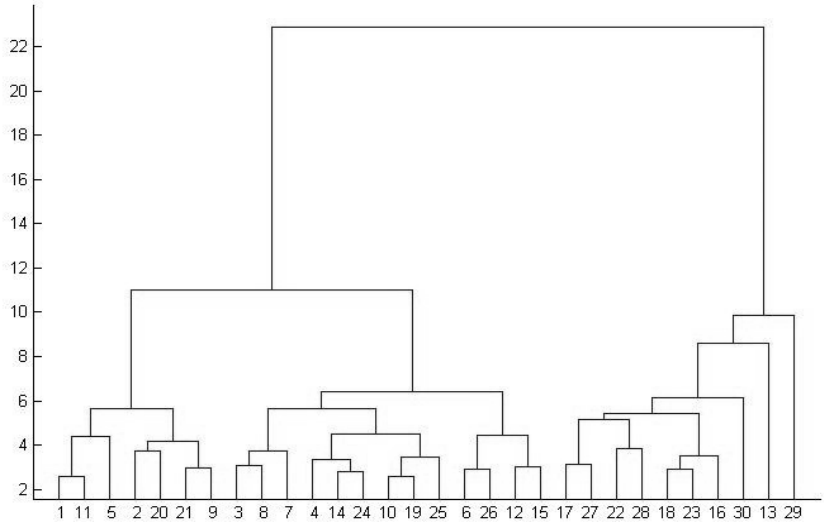
Şekil 7: MATLAB'da hiyerarşik kümeleme algoritmasına dayalı olarak Kandilli Ortalama Rüzgâr Şiddeti Tek bağlantılı kümeleme dendrogramı



Şekil 8: MATLAB'da hiyerarşik kümeleme algoritmasına dayalı olarak Kandilli Ortalama Rüzgâr Şiddeti Tam bağlantılı kümeleme dendrogramı



Şekil 9: MATLAB'da hiyerarşik kümeleme algoritmasına dayalı olarak Kandilli Ortalama Rüzgâr Şiddeti Ort. bağlantılı kümeleme dendrogramı



Şekil 10: MATLAB'da hiyerarşik kümeleme algoritmasına dayalı olarak Kandilli Ortalama Rüzgâr Şiddeti Ward metodu kümeleme dendrogramı

SONUÇ

Makalemizde ele alınan K-Means ve Hiyerarşik kümeleme algoritmaları, WEKA ve MATLAB platformlarında, Boğaziçi Üniversitesi Kandilli Rasathanesi Deprem ve Araştırma Enstitüsü'nden kaydedilen veriler kullanılarak incelenmiştir. K-Means algoritmasının çalışma prensipleri ve temel tanıtımı yapılmıştır. Aynı şekilde Hiyerarşik kümeleme için de aynı bilgiler verilmiştir. Sırasıyla WEKA ve MATLAB platformları tanıtıldıktan sonra paylaşılan ANALİZ bölümünde her iki kümeleme algoritmamız WEKA ve MATLAB platformunda uygulanmıştır.

İlk olarak K-Means kümeleme algoritması ele alındmıştır. WEKA platformunda Kandilli Ortalama Rüzgâr Şiddeti verileri ile uygulanan K-Means algoritmamız toplamda 450 deneme yapılarak incelenmiştir. Veriler arası mesafenin ölçümü için Öklid fonksiyonu kullanılmıştır. Elde edilen sonuçların incelenmesi sonucu K değeri 10 ve seed değeri 44 olan kümelemede en verimli sonuca ulaşılmıştır. Algoritma bu sonuca ulaşmak için 6 kez tekrar etmiştir. İşlemi tamamlaması için 0 sn zaman harcamıştır. Hata Kare Toplamı 8.160678975344503 olarak hesaplanmıştır.

MATLAB platformunda aynı veri kümemiz ile K-Means algoritmamız uygulanmıştır. Veriler arası mesafenin ölçümü için Öklid fonksiyonu kullanılmıştır. Kümeleme işlemi sonucunda k değeri 7 olan küme en başarılı sonucu vermiştir. Bu sonuca ulaşmak için algoritma 7 kez tekrar etmiştir.

İşlemin tamamlanması için 13.52 sn zaman harcanmıştır. Hata Kare Toplamı 12.3285684518 olarak hesaplanmıştır. Bu iki sonuç üzerinden analiz yapıldığında; WEKA platformunun algoritmanın çalışma hızı açısından çok daha hızlı çalıştığı ölçülmüştür. WEKA'nın işlem süresi 0 olarak ölçülmüştür. Bu değer MATLAB platformunda 13.52 sn olmuştur. WEKA'da 10 küme ve algoritmanın 6 tekrar ile çalışması sonucu en başarılı sonuç elde edilmiştir. MATLAB'da bu küme sayısı 7 ve algoritmanın tekrarı 7 olarak çıkmıştır. Bu küme ve algoritmanın tekrar sayısı kesin bir karar sebebi değildir. Ancak elde edilen Hata Kare Toplamı göz önüne alındığında WEKA'nın daha düşük bir değer olan 8.160678975344503 değerini vermesi, WEKA'nın bu veri kümesi üzerinde MATLAB'da elde edilen 12.3285684518 sonucuna göre daha başarılı olduğunu ıspatlamıştır.

Hiyerarşik kümeleme algoritmalarımız, WEKA platformunda Kandilli Ortalama Rüzgâr Şiddeti verilerimiz ile uygulandığında çok hızlı olmak kaydıyla, Analiz bölümünde belirttiğimiz sonuçlar elde edilmiştir. Elde edilen sonuçlar incelendiğinde görsellik açısından yeterli ve algoritmanın çalışma prensibine uygun dendrogramlar çizildiği görülmüştür. Aynı durum MATLAB platformunda yapılan Hiyerarşik kümeleme algoritması denemelerinde de görülmüştür. MATLAB platformunda elde edilen sonuçlar süre açısıdan WEKA'ya nazaran daha yüksek değerler olduğu görülmüştür. Ancak elde edilen sonuçlar karşılaştırıldığında benzer dendrogramlar elde edilmiştir. Buradan da anlaşılacağı üzere, Hiyerarşik kümeleme algoritmalarımız olan Tek, Tam, Ortalama ve Ward yöntemlerinin her iki platformda çalışması sonucu elde edilen sonuçlar büyük benzerlikler göstermiştir.

Algoritmalarımızın bu iki platformda incelenmesi esnasında fark edilen bazı hususlar bulunmaktadır. Bunlar platformların algoritmaların çalıştırılmasına uygunluğu ve kullanım kolaylığı ile ilgili konulardır. Öncelikle WEKA platformu daha önceki bölümlerde belirtildiği üzere ücretsiz bir uygulama olup, açık kaynak kodludur. Yani internet ortamındaki yazılımcılar tarafından sürekli geliştirilmekte ve programla ilgili birçok kaynak kod paylaşılmaktadır. Ancak WEKA platformunda ve kütüphanesinde bulunmayan bir algoritmanın kullanılmak istemesi durumunda birçok uyum problemi yaşanmaktadır. Bu sebeple WEKA platformunun tercih edileceği durumlarda kullanılacak olan algoritmanın, programda bulunmasına özen gösterilmesi tavsiye edilmektedir. Görsellik açısından ise WEKA, MATLAB'da elde edilen sonuçlara nazaran daha zayıf sonuçlar vermiştir. Ancak programın çalışması hızı ve kullanım kolaylığı tartışılmaz bir şekilde MATLAB'dan üstündür.

MATLAB platformu ise ücretli bir program olmasına rağmen, öğrenci ve akademisyenlere sağladığı kolaylıklar ve internet ortamında paylaşılan birçok kaynak kodu ile WEKA'dan aşağı bir platform değildir. Daha önceki bölümlerde tanıtılan bu platform, daha çok istatistiksel işlemler ve simulasyonlarda başarı ile kullanılmaktadır. Bu sebeple elde edilen sonuçlar görsellik açısından çok zengindir. Ancak MATLAB platformunda işlem yapabilmek için belirli bir seviyede yazılım bilgisine sahip olmak gerekmektedir. Platform bünyesinde birçok kütüphane ve fonksiyon bulundurmaktadır. Bu sebeple başarılı sonuçlar elde edebilmek için kodun yazılım aşamasında platform hakkında da bilgi sahibi olunması gerekmektedir. Platform görselliğinin yüksek olması sebebiyle WEKA platformuna nazaran daha uzun sürelerde işlemlerini sonuçlandırmaktadır. Kullanım kolaylığı ve çalışma hızı açısından yukarıda açıklanan sebeplerden ve çalıştığımız algoritma ve verilerimizden dolayı bu çalışmada WEKA platformuna nazaran başarı oranı daha düşük sonuçlar vermiştir. Ancak WEKA platformunda bulunmayan ve entegre edilmesi güç olan birçok yeni nesil algoritmalar, MATLAB platformunda yeterli bir yazılım bilgisiyle kolaylıkla çalıştırılabilmektedir.

KAYNAKÇA

Dubes, C.R., Jain, K.A. *Algorithms for Clustering Data.* Prentice Hall Englewood Cliffs, New Jersey 07632, pp. 55-141, 1988. ISBN 0-13-022278-X.

Kendall, M.G. *Discrimination and Classification. In Multivariate Analysis* (P.R. Krishnaiah, ed.), Academic Press, Inc., New York, pp. 165-185, 1966.

Steinhaus, H. *Sur la division des corps meteriels en parties.* Bull. Acad. Polon Sci. (in French), 4(12):801-804, 1957. MR.0090073.Zbl.0079.16403.