BÜYÜK VERİDE HİYERARŞİK KÜMELEME YÖNTEMLERİNİN KOFENETİK KORELASYON İLE KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

Murat Akşit

Danışman
Doç. Dr. Sinan Saraçlı

İSTATİSTİK ANABİLİM DALI

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANSTEZİ

BÜYÜK VERİDE HİYERARŞİK KÜMELEME YÖNTEMLERİNİN KOFENETİK KORELASYON İLE KARŞILAŞTIRILMASI

Murat AKŞİT

Danışman
Doç. Dr. Sinan SARAÇLI

İSTATİSTİK ANABİLİM DALI

EYLÜL 2020

TEZ ONAY SAYFASI

Murat AKŞİT tarafından hazırlanan "Büyük Veride Hiyerarşik Kümeleme Yöntemlerinin Kofenetik Korelasyon ile Karşılaştırılması" adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 28/09/2020 tarihinde aşağıdaki jüri tarafından **oy birliği** ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **İstatistik Anabilim Dalı'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman: Doç. Dr. Sinan SARAÇLI **İmza**

Başkan: Doç. Dr. İbrahim KILIÇ
Afyon Kocatepe Üniversitesi, Veteriner Fakültesi

Üye: Doç. Dr. Sinan SARAÇLI
Afyon Kocatepe Üniversitesi, Fen Edebiyat Fakültesi

Üye: Dr. Öğr. Üyesi Cengiz GAZELOĞLU
Isparta Süleyman Demirel Üniversitesi, Fen Edebiyat Fakültesi

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
...../...../..... tarih ve
..... sayılı kararıyla onaylanmıştır.

.....
Prof. Dr. İbrahim EROL
Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİM SAYFASI

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

28 / 09 /2020

Murat AKŞİT

ÖZET

Yüksek Lisans Tezi

BÜYÜK VERİDE HİYERARŞİK KÜMELEME
YÖNTEMLERİNİN KOFENETİK
KORELASYON İLE KARŞILAŞTIRILMASI

Murat AKŞİT
Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü
İstatistik Anabilim Dalı
Danışman: Doç. Dr. Sinan SARAÇLI

Bu çalışmada, öncelikle büyük verinin tanımı, büyük verinin bilişenleri, büyük veri analitiği ve büyük veri teknolojileri hakkında teorik-kuramsal bilgilere yer verilmiştir. Bununla birlikte kümeleme analizi, kümeleme yöntemleri, kümeleme yöntemi uzaklık ölçütleri ve Kofenetik korelasyon katsayısı hakkında da teorik-kuramsal bilgiler yer almaktadır. Devamında ise büyük veri teknolojilerini kullanarak büyük veride hiyerarşik kümeleme yöntemleri Kofenetik korelasyon katsayısı karşılaştırılmıştır. Veri analizi için açık kaynaklı büyük veri teknolojilerini içeren Amazon bulut sunucusu kullanılmıştır. Sunucu üzerine Python programlama dili kurulmuş ve analiz sürecinde Python için geliştirilmiş kütüphaneler kullanılmıştır. Çalışmada ABD Ulaştırma Bakanlığı tarafından yayınlanan 2015 Hava Seyahat Tüketici Raporundaki veri seti kullanılmıştır. Çalışmanın sonucuna etki etmeyecek veri setindeki değişkenler, analiz süreçlerini uzatabileceğinden özellik seçim işlemi ile çıkartılmıştır. Sonrasında, boş gözlemler temizlenmiş ve veriler standardize edilmiştir. Ardından, veri seti içerisinden ana kütleye temsilen rastgele seçim yöntemiyle 4 farklı veri seti oluşturulmuştur. Bu veri setlerine kümeleme analizi uygulanmıştır. Yapılan analizler sonucunda tüm veri setlerinde Kofenetik korelasyon katsayısının, ortalama bağlantı kümeleme yönteminde en yüksek değeri sağladığı gözlemlenmiştir.

2020, ix + 50 sayfa

Anahtar Kelimeler: Kofenetik korelasyon, Büyük veri, Kümeleme analizi.

ABSTRACT

M.Sc.Thesis

COMPARISON OF HIERARCHICAL CLUSTER METHODS BY COPHENETIC CORRELATION IN BIG DATA

Murat AKŞİT

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Statistics

Supervisor: Assoc. Prof. Sinan SARAÇLI

In this study, firstly, theoretical information about the definition of big data, components of big data, Big data analytics and big data technologies are included. In addition, theoretical information about cluster analysis, clustering methods, distance measures of clustering method and cophenetic correlation coefficient are given. Afterwards, hierarchical clustering methods in big data using big data technologies were compared with the cophenetic correlation coefficient. Amazon Cloud Server containing open source big data technologies was used for data analysis. Python programming language is installed on this server. Libraries developed for Python were used in the analysis processes. Air Travel Consumer Report in the USA for 2015, which was published as an open access data set, was used. Since the inclusion of variables that do not affect the result analysis may prolong the analysis process, the feature selection process has been performed. The blank observations were then cleared and the data were standardized. Afterwards, 4 different data sets were created by random selection method representing the main population from the data set. Clustering analysis was applied to these data sets. As a result of the analysis, it was observed that the cophenetic correlation coefficient gave the highest result in the Avarage Clustering method in all data sets.

2020, ix + 50 pages

Keywords: Cophenetic correlation, Big data, Cluster analysis.

TEŞEKKÜR

Bu araştırmanın konusu, deneysel çalışmaların yönlendirilmesi, sonuçların değerlendirilmesi ve yazımı aşamasında yapmış olduğu büyük katkılarından dolayı tez danışmanım Sayın Doç. Dr. Sinan SARAÇLI'ya, araştırma ve yazım süresince yardımlarını esirgemeyen her konuda öneri ve eleştirileriyle yardımlarını gördüğüm hocalarıma ve arkadaşlarıma teşekkür ederim.

Bu araştırma boyunca maddi ve manevi desteklerinden dolayı aileme teşekkür ederim.

Murat AKŞİT

Afyonkarahisar 2020

İÇİNDEKİLER DİZİNİ

Sayfa

KISALTMALAR DİZİNİ

Kısaltmalar

AWS	Amazon web hizmetleri (Amazon web servisi)
CPU	Central processing unit (Merkezi işlem birimi)
EB	Ekzabayt
EC2	Amazon elastic compute cloud (Amazon elastik bilişim bulutu)
EMR	Elastic map reduce (Elastik mapreduce)
GB	Gigabayt
HDD	Hard disk drive (Sabit disk sürücü)
HDFS	Hadoop distributed file system (Hadoop dağıtık dosya sistemi)
PB	Petabayt
RAM	Random access memory (Rastgele erişimli bellek)
TB	Terabayt

ŞEKİLLER DİZİNİ

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

1. GİRİŞ

Günümüzde artan veri hacmi ve çeşitliliği nedeniyle günümüzde veriler geleneksel yöntemler ile işlenemeyecek boyutlara ve farklılıklara ulaşmıştır. Büyük veri olarak isimlendirilen veriler, kablosuz sensörler, bloglar, elektronik posta, sosyal medya vb. gibi alıştığımızın dışında geleneksel olmayan yollardan ve tahmin edilenin ötesinde büyük boyutlarda ve birçok farklı kaynaklardan derlenmektedir. Yapılan araştırmalarda, veri türlerinin homojen yapıda ve belirli bir formata sahip olmadığı gözlemlenmiştir. Bu durum, veri bilimiyle uğraşan araştırmacıların karşılaştıkları zorlukların başında gelmektedir. Araştırmacıların karşılaştığı diğer sorunlar ise şöyle sıralanabilir; büyük depolama alanı ihtiyacı ve yüksek donanım özelliğine sahip sunucu ihtiyacıdır. Bu ihtiyaçları karşılamak için, bilgisayarların donanımsal kapasitelerinde artış ve yazılımsal çeşitlilik olmuştur. Bu sayede büyük veri teknolojileri ortaya çıkmıştır. Bu teknolojiler aracılığıyla, büyük miktarda veri gerçek zamanlı ve pratik olarak işlenebilmektedir.

Büyük veri analitiği kavramının ortaya çıkışı 1970'lere dayanmaktadır. 1970'ler öncesi, geleneksel yöntemlerle veri tabanlarında tutulan veriler kolayca analiz edilip kullanılabiliyorken, artan veri üretimi ile geleneksel depolama yöntemleri yetersiz kalmıştır. Bu durumun nedeni, çok sayıdaki verinin artan heterojen yapısı olduğu belirtilmektedir. Geleneksel veri analizi yöntemlerinin ve veri tabanlarının geliştirilmesinin büyük verinin depolanmasının ve analiz edilmesinin ne kadar önemli olduğu anlaşılmıştır (De Witt ve Gray 1992). Büyük veri analitiği alanında yapılan çalışmalarda en çok kullanılan yöntemlerden biri kümeleme analizidir. Kümeleme analizi, oluşturulan kümelerde verinin daha iyi anlaşılmasını sağlamaktadır (Liao ve Tasi 2019). Bu nedenle çalışmamızda kümeleme yöntemi kullanılmıştır. Kümeleme yöntemi literatürde sıklıkla kullanılan yöntem olmasına rağmen büyük veride kümeleme yöntemlerinin Kofenetik korelasyon katsayısı ile karşılaştırıldığı bir çalışmaya rastlanılmamıştır. Bu çalışmamızda büyük veri teknolojilerini kullanarak büyük veride hiyerarşik kümeleme yöntemleri Kofenetik korelasyon katsayısı ile karşılaştırılmıştır. Çalışmamızın önemi düşünüldüğünde hem literatürdeki bu boşluğa katkı hem de uygulamacılara fayda sağlayacaktır.

Bu bağlamda, çalışmanın literatür bölümünde büyük veri kavramı üzerinde durulmuş ve büyük verinin bileşenleri ve veri analitiği aşamaları ele alınmıştır. Daha sonra büyük veri teknolojilerinden Apache Hadoop ve Apache Spark hakkında bilgiler verilmiştir. Kümeleme analizi ve dikkat edilmesi gereken hususlar ele alınarak, uzaklık ölçütleri ve yöntemleri hakkında da bilgiler verilmiştir. Son olarak Kofenetik korelasyon katsayısı ve özellik seçiminden bahsedilmiştir.

Uygulama bölümünde ise büyük veri teknolojilerini kullanarak büyük veride hiyerarşik kümeleme yöntemleri Kofenetik korelasyon katsayısı ile karşılaştırılmıştır. Yapılan analiz sonucunda ortaya çıkan veriler incelenmiştir. Tartışma ve sonuç bölümünde elde edilen sonuçlara yer verilerek, bulgular tartışılmıştır.

2. BÜYÜK VERİ

Büyük veri ile ilgili çalışan bilim insanları bu konuda tek bir ortak tanım olamayacağına, kullanılan alana göre farklı tanımlamalar yapılabileceğine vurgu yapmışlardır. Vinod (2013)'a göre büyük veri, tipik olarak verinin büyüklük olarak Terabit veya Petabitin yüzlerce katı olmasını tanımlayan bir kavramdır. Rubinstein (2013) ise operasyonel ve uygulama bakımından büyük veriyi "işletme, devlet veya organizasyonların farklı dijital veri setlerini bütünleştirerek istatistik ve veri madenciliği teknikleriyle gizli kalmış bilgileri ve sürpriz korelasyonları kullanmaları" olarak tanımlar (Bakırarar 2016, Demirtaş ve Argan 2015).

Günümüzde veri tabanları Terabayt (TB), Petabayt (PB) ve Ekzabayt (EB) gibi terimler kullanılarak tanımlanır (Bakırarar 2016, Altunışık 2015). Tanımlar Çizelge 2.1'de sunulmuştur.

Çizelge 2.1 Büyük veri kapasitelerine ilişkin terimler.

Terim	Boyut	Kapasite
GB (Gigabayt)	1 milyar bayt	1GB=2 saatlik CD kalitesinde ses veya 7 dakikalık HD TV
TB (Terabayt	1 trilyon bayt	1TB=2000 saatlik CD kalitesinde ses veya 5 günlük HD TV
PB (Petabayt)	1 quadrilyon bayt	1PB=7 haftalık HDTV veya 1.5 milyon 64GB'lık iPod
EB (Ekzabayt)	1 quintilyon bayt	1EB=16 aylık HDTV veya 15 milyon 64GB'lık iPod

2.1 Büyük Veri Türleri

Büyük veri türlerini 6 ana başlıkta toplamak mümkündür. Operasyonel veriler, sensörler, makineler, bazı ölçüm aygıtları ve otomasyon süreçlerinden elde edilen verilerdir. Bu veriler müşteri hizmet anlaşmalarının kapsamı, tesis kurulumu ve yönetimi gibi süreçleri yönetme ile ilgili kararlar almak için kullanılabilirler. Yine bu veriler çeşitli internet sitelerindeki müşteri profilini ve hareketlerini inceleyerek daha iyi hizmet sunmaya ve pazarlama stratejilerini müşteri bazlı uygulamaya imkan sağlar. Bilimsel veri; yeni bilgiler elde etmek ve mevcut bilgiyi doğrulamak için kullanılabilir. Örnek vermek

gerekirse yeni hastalıklara ait gen tespiti, hastalık salgınlarının tahmini bilimsel veriye örnek olarak verilebilir. Ağ verileri ise ağ üzerindeki veri alışverişi sayesinde kişiler, firmalar hakkında genel bilgiye sahip olmaya ve davranış tespitine olanak sağlar (Çelik ve Akdamar 2018). Büyük veri türleri Şekil 2.1'de gösterilmiştir.

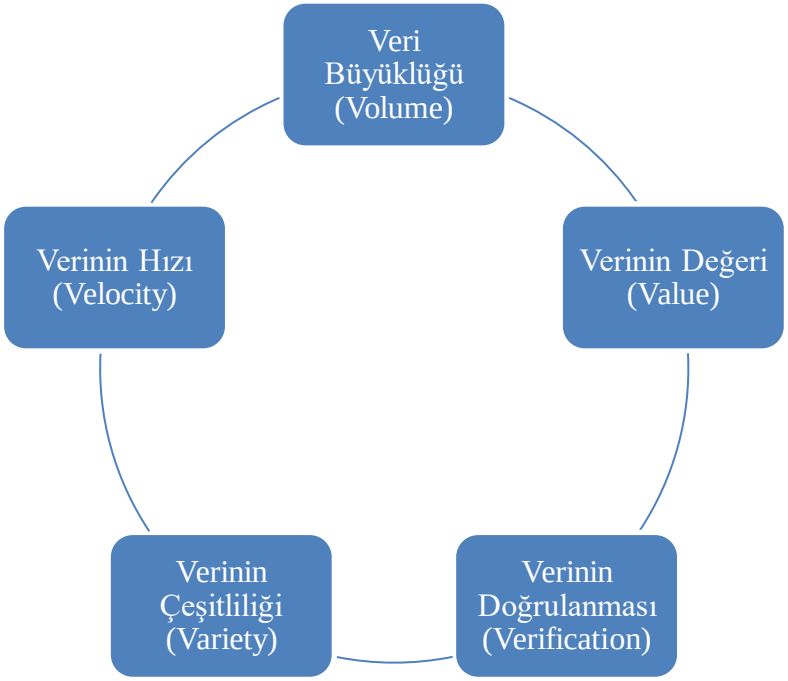
Şekil 2.1 Büyük veri türleri.

2.2 Büyük Verinin Bileşenleri

Büyük veri beş bileşenden oluşmaktadır. Bunlar;

- Veri Büyüklüğü
- Verinin Hızı
- Verinin Çeşitliliği
- Verinin Değeri
- Veriyi Doğrulama (Takçı ve Aydemir 2018).

Büyük veri bileşenleri Şekil 2.2'de özetlenmiştir.



Şekil 2.2 Büyük veri bileşenleri.

2.2.1 Veri Büyüklüğü

Büyük veri sayesinde, küresel anlamdaki veri büyüklüğü de ciddi oranda artırmaktadır. Bunun sebebi yüksek hızda üretilen verilerin çok hızlı artması ile açıklanabilmektedir. Bu artışlar, verinin toplanması, saklanması, ve analiz edilmesi gibi hususlarda firmaların teknolojik yatırım yapması gerekliliğini artırmaktadır (Aslan ve Özerhan 2017, Warren vd. 2015).

2.2.2 Verinin Hızı

Büyük veri üretiminin gittikçe artış göstermesi verinin ihtiyaç duyulan yerdeki işlem hızını artırmakta ve veri çeşitliliğine önemli katkılar sunmaktadır (Özdemir ve Sağıroğlu 2018, Schaeffer ve Olson 2014).

2.2.3 Verinin Çeşitliliği

Birçok kurum ve kuruluşlar tarafında kablosuz sensörler, bloglar, elektronik posta ve sosyal medya verileri vb. veriler gerçek zamanlı verilerin üretilmektedir. Bu veriler gün geçtikçe artmakta ve çeşitlenmektedir (Aktan 2018).

2.2.4 Verinin Değeri

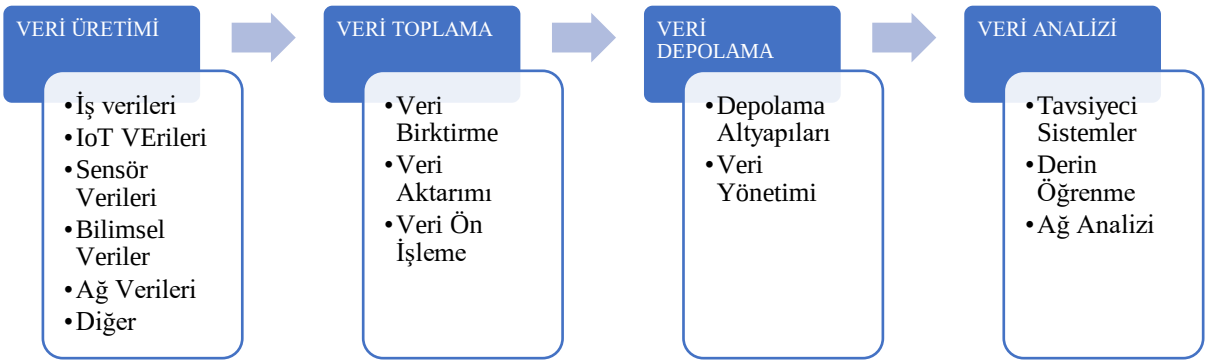
Verinin sağladığı değerler birçok çok alanda doğru ve etkin kararların verilmesinde önemli yere sahiptir. Şirketlerin doğru strateji elde etmelerini sağlayarak şirketlere ciddi katkılar sunmakta ve artı değer yaratmaktadır (Aslan ve Özerhan 2017, Kaya vd. 2017).

2.2.5 Verinin Doğrulanması

Verilerin güvenli bir şekilde üretilmesi önemli bir husustur. Verilerin etik ve güvenli bir şekilde üretilmesi, kaynaklarının doğrulanması ve gizliliği kritik unsurlar arasında yer almaktadır (Altındiş ve Kıran 2018).

2.3 Büyük Veri Yaşam Döngüsü

"Büyük veri yaşam döngüsü" adım adım büyük verinin üretiminden başlayarak tüm adımları özet olarak ifade etmektedir. Tıklamalar, kurum ticari bilgileri, çevrimiçi insanların (mesaj, ses, görüntü vb.) etkileşimi ile elde edilen bilgiler, bilimsel araştırmalar sonucu elde edilen bilgiler büyük veri yaşam döngüsünün ilk aşaması olan veri kaynağı adımını ifade etmektedir. Bu aşamada gerçek zamanlı, çoğunlukla akışkan olan bilgi toplanmaya çalışılmaktadır. Verilerin elde edilmesi adımı verilerin bir veri kümesine depolanmasından önce toplanması, filtrelenmesi ve temizlenmesi süreci olan büyük yaşam döngüsünün ikinci aşamasıdır. Üçüncü aşamada depolanan veri, son aşamada veriler analiz edilmektedir (Cavanillas vd. 2016). Büyük verinin yaşam döngüsü Şekil 2.3'te gösterilmiştir.



Şekil 2.3 Büyük veri yaşam döngüsü.

2.4 Büyük Veri Analitiği

Büyük veri analizinde toplanan verilerin analizinden elde edilen sonuçlar kurumların geleceğine dair karar almasını doğrudan etkilemektedir. Bu bağlamda gerçek zamanlı elde edilen veriler depolanıp, analiz edilip ve sonrasında raporlanması büyük önem arz etmektedir. Büyük veriyi işleyebilme kapasitesine sahip olan makineler ile anlık veri modellemeleri yapılması şirketlerin karar alma mekanizmalarını kolaylaştırmaktadır (Kong vd. 2014).

2.4.1 Açıklayıcı Veri Analizi

Bölümlere ayırma, kümeleme ve sınıflandırma gibi tanımlayıcı analizler, verilerin şekilleri ve kalıpları hakkında bilgi elde etmek için sürecin ilk aşamasında gerçekleştirilen analiz yöntemidir (Hardoon ve Nash 2017).

2.4.2 Tanımlayıcı Veri Analizi

Tanımlayıcı Veri Analizi, veri kümesindeki bilgiler arasındaki ilişkileri anlamak için gerçekleştirilen analiz yöntemidir. **“**Bir müşterinin bir ürünü diğerine tercih etmesi ne ile ilişkilidir?” gibi sorularına cevap aramada kullanılmaktadır (Onay 2020).

2.4.3 Tahmini Veri Analizi

Geçmişin bilgisini kullanarak geleceği anlama aşamasında gerçekleştirilen analiz yöntemidir. Örneğin, “bir sonraki adımın ne olacağı” veya “bir müşterinin bundan sonra ne satın alması muhtemeldir” sorularına cevap aramaktadır (Bilgiç vd. 2019).

2.4.4 Kuralcı Veri Analizi

Kuralcı veri analizi, bu sınıflandırma içindeki en zor analiz türüdür. İstenen bir sonucun meydana gelme olasılığını arttırmak için neler yapılabileceği konusunda fikir vermektedir. “Bir müşterinin B ürününe göre A ürününü seçme olasılığı” nedir gibi sorularına cevap aramaktadır (Cibaroğlu ve Yalçınkaya 2019).

2.5 Büyük Veri Teknolojileri

Büyük veri teknolojileri maliyet, zaman, verim ve kalite arttırma açsından bir çok kolaylık sağlamaktadır. Bu büyük teknolojileri Apache Hadoop ve Apache Spark’dır.

2.5.1 Apache Hadoop

Büyük veri kümelerinin basit programlama modellerini kullanarak dağıtık olarak işlemeye olanak sağlayan bir kütüphanedir. Tek bir bilgisayardan oluşan sunucularda kullanılabileceği gibi, binlerce bilgisayardan oluşan sunucular içinde de kullanılabilecek şekilde geliştirilmiştir. Uygulama katmanlarında meydana gelen hataları belirleyebildiği ve hatalarla başa çıkabildiği için yüksek kullanılabilirlik sağlamaktadır (Takcı ve Aydemir 2018).

Apache Hadoop’un işlevselliği dağıtık hesaplama mimarisi sayesinde gerçekleşmektedir. Dağıtık hesaplama; problemlerin çözümü için, birden çok bilgisayarın tek bir bilgisayar gibi davranması yaklaşımına dayanmaktadır. Bir iş tüm bilgisayarlar arasında iş bölümü yapılarak ele alınır ve tek bir bilgisayar gibi davranılarak iş sonuçlandırılır (Keskin 2018, Gonzolez 2012).

Apache Hadoop temel olarak Hadoop Common, Hadoop Distributed File System (HDFS), Hadoop YARN ve Hadoop MapReduce bileşenlerinden oluşur (Keskin 2018).

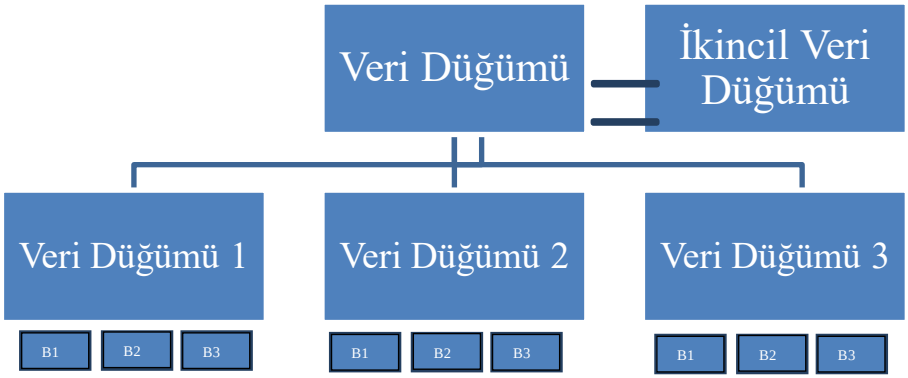
2.5.1.1 Hadoop MapReduce

MapReduce, Apache Hadoop'ta kullanılmak üzere geliştirilmiş bir programlama modelidir. Hadoop MapReduce, büyük, çok setli donanım kümelerinde paralel olarak büyük veri setlerini ölçeklenebilir, güvenilir ve hataya dayanıklı bir şekilde işleyen ve analiz eden uygulamalar geliştirmek için tasarlanmıştır (Kaya ve Aydoğan 2019).

2.5.1.2 Hadoop Distributed File System

HDFS, Güvenilir, ölçeklenebilir ve hataya dayanıklı veri depolama sağlayan, kendi kendini onaran, dağıtılmış bir dosya sistemidir. Depolama kaynaklarını ve hesaplamayı büyük kümelerde gerçekleştirir. HDFS, mimariden bağımsız olarak metin, resim, video vb. Herhangi bir formattaki verileri kabul eder ve yüksek bant genişliği akışı için otomatik olarak optimize etme özelliğine sahiptir (Ghazi ve Gangodkar 2015).

HDFS'nin en büyük avantajı hata toleransıdır. Olası depolama kaynak hataları durumunda bile hizmet vermeye devam eder. Bu sayede kaynak kaybını minimuma indirmektedir (Ghazi ve Gangodkar 2015, Faghri vd. 2013). HDFS'nin çalışma mantığı Şekil 4'de gösterilmiştir.



Şekil 2.4 Hadoop distributed file system (HDFS) mimarisi.

2.5.1.3 Hadoop YARN Framework

YARN'ın temel ilkesi, kaynak yönetimi ve iş planlama işlevlerini ayrı ayrı yönetmektir. Kaynak yöneticisinin iki ana bileşeni vardır. Bunlar zamanlayıcı ve uygulama yöneticisidir. Zamanlayıcı yönetici, kaynakları çalışan çeşitli uygulamalara tahsis eder ve uygulamaların kaynak gereksinimlerine göre zamanlama gerçekleştirir. Uygulama yöneticisi iş gönderimlerini kabul eder ve her iş uygulama yöneticisine tahsis etmede görev alır. Uygulama yöneticisi, uygulamayı yürütmek için kapsayıcılara ayırır ve hata durumunda uygulama ana kapsayıcısını yeniden başlatmak için her uygulamanın uygulamaya özel yöneticisi iletişime geçer (Bhathal ve Singh 2019).

2.5.1.4 HBASE

Hataya dayanıklı ölçeklenebilir bir veritabanı projesidir. HBase, verilere rastgele gerçek zamanlı okuma/yazma erişimi ile HDFS dosya sisteminin üzerine kurulmuştur. Her HBase tablosu, her hücrenin bir zaman damgasına sahip olduğu, sıralar ve sütunlar ile çok boyutlu bir veritabanı olarak saklanmaktadır (Taylor 2010).

2.5.1.5 Pig

Pig derleyicisi Hadoop içinde yürütmek için Harita/Küçültme programları dizileri üreten yüksek seviyeli bir veri akışı dili (Pig/Latin) ve yürütme çerçevesidir. Pig, verilerin toplu işlenmesi için tasarlanmıştır (Yavuz vd. 2012).

2.5.1.6 Hive

SQL tipi sorgulama dili ile özel sorgulama için kullanılan ve daha karmaşık analizler için kullanılan bir veri ambarı projesidir. SQL benzeri bir sorgu dili olan HiveSQL, özetler, raporlar ve analizler oluşturmak için kullanılmaktadır (Gupta ve Gupta 2017).

2.5.1.7 Cascading

Hadoop MapReduce katmanının üstünde bulunan ince, açık kaynaklı bir Java kitaplığıdır. Hadoop kümesinde hataya dayanıklı veri işleme akışlarını tanımlamak ve yürütmek için tasarlanmış bir API projedir. MapReduce'dan daha yüksek bir düzeyde çalışmasına ve karmaşık dağıtılmış süreçleri daha hızlı bir şekilde bir araya getirmesine ve bağımlılıklara göre zamanlamasına olanak tanıyan bir sorgu işleme imkanı sağlamaktadır (Kunal 2016).

2.5.2 Apache Spark

Başlangıçta 2009 yılında UC Berkeley'nin AMPLab'da geliştirilmiş ve 2010 yılında açık kaynaklı bir Apache projesi olarak sunulmuştur. Apache Spark, Hadoop MapReduce'u yapısına bir alternatif olarak geliştirilmiştir. Spark'ın temel özelliği, bir uygulamanın işlem hızını arttıran bellek içi küme işlemidir. Spark, toplu iş uygulamaları, algoritmaları, etkileşimli sorgular ve akış gibi çok çeşitli iş yüklerini kapsayacak şekilde tasarlanmıştır. Bütün bu iş yükünü ilgili bir sistemde desteklemenin yanı sıra, ayrı araçları korumanın yönetim yükünü de azaltmaktadır (Çelik 2017, İnt.Kyn.2).

2.5.2.1 Spark Core

SparkCore, bellek yönetimi, zamanlama süreci, yönetim bileşenleri için dönüşümler, eylem ve paylaşılan değişkenler gibi önemli bir operasyon kolaylığı sağlayarak tüm sistemin temelini oluşturmaktadır (Fikri vd. 2019).

2.5.2.2 Spark SQL

Basit ve kullanışlı ara yüzü ile çeşitli veri kaynakları üzerindeki ilişkisel dönüşümleri SQL sorguları ile işleme, kalıcı bir tablo olarak saklama, sıralama ve bölümlemeye izin veren veri ambarı projesidir (Salloum 2016).

2.5.2.3 Spark Streaming

Bir veri kümesi bellekte saklanabilir, ancak sürekli veri kümesi akışı olduğu durumlarda bellekte saklanabilmesi imkansız bir işlemdir ve beklenmedik veri kayıpları neden olabilmektedir. SparkStreaming, sürekli veri akışı için verilerin canlı veri işleme imkanı sunmaktadır. Sürekli ve hataya dayanıklı canlı veri işleme süreci için iki denetim noktası mevcuttur. Yapılandırmaya, işlemlere ve zamanlanmış görevlere dayanan öğeler için meta veri kontrol noktası ve veri kümeleri için veri kontrol noktası bulunmaktadır (Li vd. 2019).

2.5.2.4 Machine Learning Library

Sınıflandırma, regresyon ve kümeleme gibi yaygın öğrenme algoritmaları ve istatistik araçları içeren kütüphanedir. Bu kütüphane özellikle büyük ölçekli ortamlarda süreçleri hızlandırmak ve basitleştirmek için tasarlanmıştır (Gil 2017, Meng vd. 2016).

2.5.2.5 Spark GraphX

Spark'daki grafik işleme sistemidir. Kullanıcılar hem grafikleri hem de koleksiyonları dönüşümlü olarak görüntüleyebilme, dönüştürebilme ve birleştirebilme imkanı sunmaktadır (Gil 2017, Malewicz vd. 2010).

3.KÜMELEME ANALİZİ

Kümeleme analizi, değişkenleri benzerliklerine göre gruplandırmak ve aynı gruba ait nesneler hakkında bu gruplar aracılığıyla özet bilgi elde etmek için en önemli veri madenciliği süreçlerinden biridir. Bu nedenle, başlangıç aşaması sonucunda kaç kümenin oluşturulacağı ve bu kümelenme sürecini hangi niteliklerin etkileyeceği bilinmemektedir (Yılmaz ve Patır 2011).

Kümeleme analizi, verilerin mevcut olduğu her yerde kullanılabilir. Bununla birlikte, yaygın olarak kullanım alanlarından bazıları şu şekilde sıralanabilir

- Müşteri Davranış Analizi
- Web Pazarlama İşlemleri
- Metin Analizi
- Yazılım Geliştirme

3.1 Kümeleme Analizinde Dikkat Edilmesi Gereken Hususlar

Veri kümesi için iyi bir kümeleme yapmak üzere uygun algoritmanın seçilmesi ile doğrudan ilgilidir. Benzerlik ölçütleri ve kümeleme metotları genellikle veri kümesinin yapısına uygun kümeleme tasarımını tanımlamak için oldukça hızlı ve verimli çalışan kümeleme algoritmasını anlamaya çalışırlar (Doğan 2002).

Benzerlik ölçütü seçimi: İki veri noktasının ne kadar benzer olduğunu ölçer. Çoğu durumda, veri noktalarının tüm nitelikleri yakınlık ölçüsünün hesaplanmasına eşit katkıda bulunur. Veri noktalarının hiçbir özelliği diğerleri üzerinde baskın değildir (Karakoç 2019).

Kümeleme metodu seçimi: Bu adımda, sabit bir işlevle veya başka tür kurallarla ifade edilebilen kümeleme ölçütünü tanımlamak gerekmektedir. Veri kümesinde oluşması beklenen tüm küme türleri dikkate alınarak işleme alınmalıdır. Böylece, veri kümesine doğru bölünmeyi sağlayan en iyi kümeleme kriteri belirlenmesinde yardımcı olmaktadır (Karakoç 2019).

3.2 Uzaklık Ölçütleri

İki birim arasındaki uzaklık, bu iki birimin üçüncü bir birime olan uzaklıkları toplamından küçüktür veya bu toplama eşittir (Kazaz 2019).

- Pozitiflik

$d(i, j) \geq 0$

- Yansıtma

$d(i, j) = 0\ i \Leftrightarrow j$

- Simetri

$d(i, j) = d(j, i)$

- Üçgen eşitsizliği

$d(i, j) \leq d(i,k) + d(k, j)$

3.2.1 Öklid Uzaklığı

Kümeleme yöntemlerinde kullanılan popüler ve klasik benzerlik ölçülerinden biridir. Öklid mesafesi, iki nokta veya vektör arasındaki mesafe olarak tanımlanmaktadır (Kumar ve Toshniwal 2016). Öklid Uzaklığı Eşitlik 3.1 verildiği gibi hesaplanmaktadır.

(T_1) ve (T_2) = İki nokta veya vektör

$$d_{Euclidean}(T_1, T_2) = \sum_{j=1}^{n} \sqrt{(T_{1j} - T_{2j})^2} \qquad (3.1)$$

3.2.2 Canberra Uzaklığı

Negatif olmayan değerler alan ve sıfıra yakın değerdeki küçük noktalar için duyarlı bir uzaklık ölçütüdür (Kazaz 2019, Ziviani vd. 2004). Canberra Uzaklığı Eşitlik 3.2 verildiği gibi hesaplanmaktadır.

(x_i) ve (x_j) = İki nokta veya vektör

$$d_{Canberra}(x_i, x_j) = \sum_{l=1}^{d} \frac{\mathrm{I}x_{il} - x_{jl}\mathrm{I}}{\mathrm{I}x_{il}\mathrm{I} + \mathrm{I}x_{jl}\mathrm{I}} \quad (3.2)$$

3.2.3 Manhattan Uzaklığı

İki nokta arasındaki Manhattan mesafesi, koordinatlarının mutlak farklılıklarının toplamı olarak ifade edilmektedir (Kumar vd. 2014). Manhattan Uzaklığı Eşitlik 3.3 verildiği gibi hesaplanmaktadır.

(x_i) ve (x_j) = İki nokta veya vektör

$$d_{Manhattan}(\chi_i, \chi_j) = \sum_{l=1}^{d} \mathrm{I}\chi_{il} - \chi_{jl}\mathrm{I} \quad (3.3)$$

3.2.4 Minkowski Uzaklığı

Minkowski mesafesi, hem Öklid mesafesinin hem de Manhattan mesafesinin genelleştirilmesi olarak kabul edilebilecek vektör uzayında bir metrik olarak tanımlanmaktadır (Kumar ve Toshniwal 2016). Minkowski Uzaklığı eşi Eşitlik 3.4 verildiği gibi hesaplanmaktadır.

(T_1) ve (T_2) = İki nokta veya vektör

$$d_{Minkowski}(T_1, T_2) = (\sum_{i=1}^{n} \mathrm{I}T_{1i} - T_{2i}\mathrm{I}^2)_p^1 \quad (3.4)$$

3.2.5 Spearman Uzaklığı

Spearman uzaklığı Öklid mesafesinin karesi alınarak hesaplanan bir ölçüm metodudur (Jaskowiak vd. 2014). Spearman Uzaklığı Eşitlik 3.5 verildiği gibi hesaplanmaktadır.

(T_1) ve (T_2) = İki nokta veya vektör

$$d_{Spearman}(T_1, T_2) = \sum_{j=1}^{n} (T_{1j} - T_{2j})^2 \quad (3.5)$$

3.2.6 Pearson Uzaklığı

Pearson mesafesi ölçüsü Pearson korelasyon katsayısından türetilmiştir. Korelasyon katsayısı, iki veri noktası arasındaki doğrusal bağımlılık derecesini ölçmek için kullanılır. Korelasyon temelli mesafe ölçüsü matematiksel olarak formüle edilmiştir (Xu ve Wunsch 2005). Pearson Uzaklığı Eşitlik 3.6 verildiği gibi hesaplanmaktadır.

(x_i) ve $\left(x_j\right)$ = İki nokta veya vektör

$$PE\left(\chi_i, \chi_j\right) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{y})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (3.6)$$

3.2.7 Kendall Uzaklığı

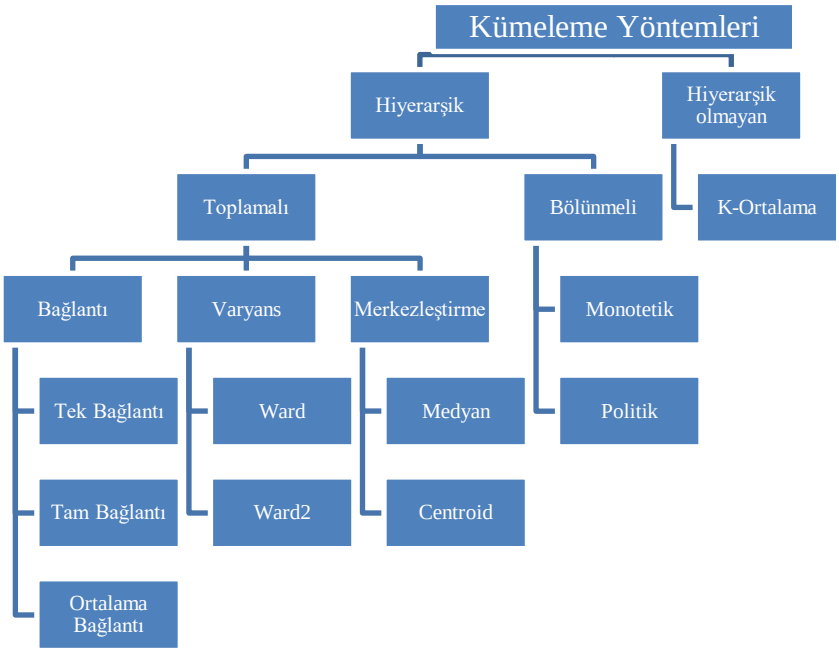
Sıra tabanlı bir korelasyon katsayısıdır. X ve Y 'deki değer çiftlerinin sayısını hesaplar. Bu farklı normalleştirmeden Kendall, yalnızca değerlendirme altındaki dizilerin nötr çiftleri olmadığında maksimum değerlerini elde dikkate almaktadır (Jaskowiak vd. 2014). Kendall Uzaklığı Eşitlik 3.7 verildiği gibi hesaplanmaktadır.

(x_i) ve $\left(x_j\right)$ = İki nokta veya vektör

$$KE\left(x_i, y_j\right) = \frac{P_+ - P_-}{\frac{n(n-1)}{2}} \quad (3.7)$$

3.3 Kümeleme Yöntemleri

Kümeleme yöntemleri Hiyerarşik ve Hiyerarşik Olmayan şeklinde iki başlıkta incelenmektedir. Başlıklar ise Şekil 3.1'de gösterildiği gibi sınıflandırılmıştır (Kazaz 2019, Akın 2008).



Şekil 3.1 Genel Kümeleme analizi sınıflaması.

3.3.1 Hiyerarşik Olmayan Kümeleme Yöntemi

Hiyerarşik olmayan kümeleme yöntemi, önceden belirlenmiş sayıda kümeden veri elde etmeyi denemektedir. Her nesne en az bir nesne içermeli ve her nesne tam olarak bir gruba ait olmalıdır şeklinde açıklamaktadır. Nesnelerin k tarafından sabitlendiği ve kullanıcı tarafından k'nin verildiği k gruplarına sınıflandırılması, verilerde bulunan 'doğal' grupları ortaya çıkarmıştır (Fırat vd. 2013).

Hiyerarşik kümeleme yöntemlerinden farklı olarak, hiyerarşik olmayan kümeleme yöntemleri verilerin tek bir bölümünü oluşturmaktadır. Hiyerarşik yöntemler genellikle yakınlık matrisini kullanırken, hiyerarşik olmayan yöntemler ise desen matrisini kullanmaktadır (Sakarya 2007, Johnson ve Wichern 1988).

3.3.2 Hiyerarşik Kümeleme Yöntemi

Verilerin uzaklık veya benzerlik matrislerindeki aralarındaki ilişkiyi hesaplayarak küme oluşturmaktadır. Özellikle verilerin 250'den az olduğu durumlar küçük örneklemler için tercih edilmektedir (Kazaz 2019).

3.3.2.1 Tek Bağlantılı Kümeleme Yöntemi (TEBKY)

İki küme arasındaki minimum uzaklık olarak tanımlanmıştır. Küme yapısını dikkate almaz. En yakın komşuluk olarak da adlandırılmaktadır (Derya 2019, Murtagh ve Contreras 2017).

İki küme C_1 ve $C_2 \cup C_3$ arasındaki minimum uzaklık Eşitlik 3.8 verildiği gibi hesaplanmaktadır.
d=İki küme arasındaki uzaklık

$$d(C_1, C_2 \cup C_3) = min[d(C_1, C_2), (C_1, C_3)] \quad (3.8)$$

3.3.2.2 Tam Bağlantılı Kümeleme Yöntemi (TABKY)

İki küme arasındaki maksimum uzaklık olarak tanımlanmıştır. Tek bağlantı yöntemi gibi küme yapısını dikkate almaz. En uzak komşuluk olarak da adlandırılmaktadır (Fırat 1997, Everitt 2011). İki küme C_1 ve $C_2 \cup C_3$ kümeleri arasındaki maksimum uzaklık Eşitlik 3.9 verildiği gibi hesaplanmaktadır.
d=İki küme arasındaki uzaklık

$$d(C_1, C_2 \cup C_3) = max[d(C_1, C_2), (C_1, C_3)] \quad (3.9)$$

3.3.2.3 Ortalama Bağlantı Kümeleme Yöntemi (OBKY)

İki küme arasındaki mesafe, her gruptan bir örnekten oluşan tüm veri çiftleri arasındaki mesafenin ortalamasıdır. Ortalama bağlantı yaklaşımı kullanarak ağırlıksız çift grup

yöntemi olarak da kabul edilmektedir (Carvalho 2019, Everitt 2011). Ortalama bağlantı kümeleme yöntemi Eşitlik 3.10 verildiği gibi hesaplanmaktadıe.

Buradaki n_1 , n_2 ve n_3sırasıyla C_1 ve C_2kümelerindeki örnek veri çiftleridir.
d=İki küme arasındaki uzaklık
n=veri sayısı

$$d(C_1, C_2 \cup C_3) = \frac{n_2.\mathrm{d}(C_1,C_2) + n_3.\mathrm{d}(C_1,C_3)}{n_2+n_3} \quad (3.10)$$

3.3.2.4 Ward Kümeleme Yöntemi

Ward yöntemi, küme içi varyansı en aza indirerek yeni kümeler elde etmektedir. Bu kümeler içerisinden hata kare değerinin düşük olan kümeyi seçmektedir (Çelik 2013, Aldenderfer ve Blashfield 1984). Ward kümeleme yöntemi Eşitlik 3.11 verildiği gibi hesaplanmaktadır.

d=İki küme arasındaki uzaklık
x= i'inci gözlem
n=veri sayısı

$$d = \sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n} \quad (3.11)$$

3.3.2.5 Ward D2 Kümeleme Yöntemi

Kareler ölçütünün hata toplamından kaynaklandığından dolayı Ward'ın aksine, Öklid mesafelerini hesaplamaktadır (Carvalho 2019, Everitt 2011). Ward D2 kümeleme yöntemi Eşitlik 3.12 verildiği gibi hesaplanmaktadır.

(i) ve (i') = İki nokta

$$d(i, i') = \Sigma_j (x_{ij} - x_{i'j})^2 \quad (3.12)$$

3.3.2.6 Centroid Kümeleme Yöntemi

Bu yöntem, noktaların Öklid uzayında temsil edilebileceğini varsayarak, kümelerin merkez uzaklığını hesaplamaktadır. Her küme, merkez olarak adlandırılan örnek ortalaması ile temsil edilmektedir. (Fırat 1997, Everitt 2011). Centroid kümeleme yöntemi Eşitlik 3.13 verildiği gibi hesaplanmaktadır.

d=İki küme arasındaki uzaklık
n=veri sayısı

$$d(C_1, C_2 \cup C_3) = \frac{n_2}{n_{2+}n_3} d(C_1, C_2) + \frac{n_3}{n_{2+}n_3} d(C_1, C_3) - \frac{n_2 n_3}{(n_2+n_3)^2} d(C_2, C_3) \quad (3.13)$$

3.3.2.7 Medyan Kümeleme Yöntemi

Birleştirilecek iki kümenin boyutları çok farklıysa, yeni kümenin merkezi daha büyük kümeninkine çok yakın olacağından bu kümele yöntem dezavantaj oluşturabilmektedir. Bu nedenle Gower tarafından medyan yöntem olarak adlandırılan alternatif bir yöntem geliştirmiştir. Bu yöntem hem benzerlik hem de mesafe ölçümleri için uygun hale getirilebilmektedir (Carvalho 2019).

3.3.2.8 Mcquitty Kümeleme Yöntemi

Benzerlik analizi, hem ayrık hem de sürekli verilere uygulanabilmektedir. haliyle, dezavantajlarına sahiptir. Bununla birlikte, karmaşık ve zahmetli olmasından kaynaklı tutarsızlıklara yol açabilmektedir. Bu tür problemleri çözmek hem ayrık hem de sürekli verilere uygulanabilir karşılıklı çiftler tarafından benzerlik analizi olarak adlandırılan iki küme toplamının yarısı dikkate alınarak çok basit bir hiyerarşik analiz yöntemi geliştirilmiştir (Kayaalp vd. 2000, Mcquitty 1966). Mcquitty kümeleme yöntemi Eşitlik 3.14 verildiği gibi hesaplanmaktadır.
d=İki küme arasındaki uzaklık

$$d\left(x_i, x_j\right) = \frac{d_{x_i} + d_{x_j}}{2} \quad (3.14)$$

4. KOFENETİK KORELASYON KATSAYISI

Kofenetik korelasyon katsayısı, ham veri uzaklıkları ile kullanılan uzaklık ölçütleri arasındaki uyumu değerlendirmek için hesaplanan bir katsayıdır (Ponde 2016, Choi vd. 2010). Hem veri seti sınıflandırmasının uygun bir uzaklık ölçütünü hem de çeşitli kümelenme tekniklerinin verimliliğini değerlendirmek için yaygın olarak tercih edilmektedir (Carvalho 2019, Saraçlı vd. 2013). Kofenetik korelasyon katsayısının yüksek olması, veri seti için en doğru kümeleme ve uzaklık ölçütü olduğunu göstermektedir (Ponde 2016, Choi vd. 2010). Kofenetik korelasyon katsayısı eşitlik 4.1 verildiği gibi hesaplanmaktadır.

x(i, j) = |Xi – Xj |= Öklid mesafesi

t(i, j) = |Ti – Tj |= Dendrogram mesafesi

$$c = \frac{\sum_{i<j}(x(i,j)-x)(t(i,j)-t}{\sqrt{\sum_{i<j}[x(i,j)-x)^2]\sum_{i<j}[t(i,j)-t)^2]}} \quad (4.1)$$

Yapılan çalışmalar incelendiğinde, 3 farklı kaza alanından toplanan 45 farklı seramik parçasına uygulanabilecek en doğru kümeleme yöntemi belirlenilmeye çalışılmıştır. Analiz sonucunda Kofenetik korelasyon katsayısının, ortalama bağlantı yönteminde en yüksek değeri elde ettiği gözlemlenmiştir (Carvalho 2019). Farklı bir çalışmada 26 ilçe yaşanan 1560 kaza için aynı yöntem ile en doğru kümeleme yöntemi belirlenilmeye çalışılmıştır. Kofenetik korelasyon katsayısının, ortalama bağlantı yönteminde en yüksek değeri elde ettiği gözlemlenmiştir (Kumar 2016). Farklı bir çalışmada 211 güvenlik tasarım deseni için en doğru kümeleme yöntemi belirlenilmeye çalışılmıştır. Kofenetik korelasyon katsayısının, ortalama bağlantı yönteminde en yüksek değeri elde ettiği gözlemlenmiştir (Ponde 2016). Farklı bir çalışmada 17 sarımsak çeşidi için aynı yöntem uygulanarak en doğru kümeleme yöntemi belirlenilmeye çalışılmıştır. Kofenetik korelasyon katsayısının, ortalama bağlantı yönteminde en yüksek değeri elde ettiği gözlemlenmiştir (Silva 2013). Literatürdeki farklı açıdan incelenen çalışmada ise değişken sayısı ve gözlem sayısına göre farklı veri setleri oluşturmuştur. Bu ger veri seti için Kofenetik korelasyon katsayısı ile de en iyi kümeleme yöntemi bulmaya çalışılmıştır. Tüm veri setlerinde Kofenetik korelasyon katsayısı ortalama bağlantı yönteminde en yüksek değerler elde edildiği gözlemlenmiştir (Saracli 2013).

5. ÖZELLİK SEÇİMİ

Özellik seçimi veri setinden n adet özellik arasından k adet özelliği seçerek veri setini temsil edebilecek en iyi alt kümenin seçimi olarak tanımlanmaktadır (Budak 2018, Forman 2003). Özellik seçimi, analize başlamadan önce veri setindeki sonuca etki etmeyecek değişkenlerin belirlenmesinde kullanılmaktadır. Bu yöntem büyük veri ve veri madenciliği süreçlerinde ilk ve önemli adımların başında gelmektedir (Guyon ve Elisseeff 2003).

Özellik seçiminde kullanılan yöntemler ise;

- Filtreleme Yöntemleri
- Sarmal Yöntemler
- Gömülü Yöntemler

olmak üzere genel olarak üç grupta toplanmaktadır (Rong vd. 2019).

5.1 Filtreleme yöntemi

Filtreleme yöntemi, büyük veri ve veri madenciliği süreçlerinde en çok tercih edilen özellik seçimi yöntemidir. Bu yöntemde uzaklık, bilgi, bağımlılık ve ilişki gibi istatistiksel metotlara dayalı özellik seçimi yapılmaktadır. En çok Korelasyon-bazlı öznitelik seçme yöntemi kullanılmaktadır (Gümüşçü vd. 2016).

5.1.1 Korelasyon Tabanlı Özellik Seçimi

Korelasyon tabanlı özellik seçimi veri setinin içerisinden en yüksek korelasyon katsayısına sahip ve birbirinden farklı öznitelikler içeren alt kümeleri bulma esasına göre seçim yapmaktadır (Emhan ve Akın 2019). Korelasyon tabanlı özellik seçim yöntemi Eşitlik 5.1 verildiği gibi hesaplanmaktadır.

k= alt kümedeki özellik sayısı,
$\overline{r}_{ci}$= Y ile özellik arasındaki ortalama korelasyonu,
$\overline{r}_{ii}$= Özelliklerin birbirleri arasındaki ortalama iç korelasyonunu

$$M_s = \frac{k\bar{r}_{ci}}{\sqrt{k+k(k-1)\bar{r}_{ii}}} \quad (5.1)$$

5.1.2 Bilgi Kazancı Özellik Seçimi

Bilgi kazancı skoru Entropi modeli kullanılarak, X'in özelliklerinin yardımı ile Y özelliğini tanımlamak için hesaplanmaktadır. Bilgi kazancı simetrik bir ölçüt olup, X ve Y'nin gözlemlendikten sonraki bilgileri birbirine eşittir. Bu seçim ile daha fazla bilgi elde edilebildiği gibi, bu bilgiler ön yargı olarakta kullanılabilmektedir. Bu da yöntemin zayıf yönünü oluşturmaktadır (Budak 2018, Holmes ve Nevill-Manning 1995). Eşitlik 5.2'de Y için simetrik ölçüt hesaplaması, Eşitlik 5.3'de X için simetrik ölçüt hesaplaması ve Eşitlik 5.3'de bilgi kazancı formülüze edilmiştir.

$$H(Y) = \sum_{y\epsilon Y} p(y)\log_2(p(y)) \quad (5.2)$$

$$H(Y \setminus X) = \sum_{x\epsilon X} p(x) \sum_{y\epsilon Y} p(y \setminus x)\log_2(p(y \setminus x)) \quad (5.3)$$

$$\text{Bilgi Kazancı} = \text{H(Y)} - \text{H(Y} \setminus \text{X)} \quad (5.4)$$

5.1.3 Kazanç Oranı Özellik Seçimi

Bilgi kazancı yöntemi özellik seçiminde çok sapmalar meydana geldiği için, sapmayı azaltmak için kazanç oranı yöntemi geliştirilmiştir. Bu yöntem sapmayı azaltmak için bölünme bilgisini kullanmaktadır (Rong vd. 2019). Kazanç oranı 0-1 aralığında bir değer almaktadır. Eşitlik 5.5'de bölünme bilgisi ve Eşitlik 5.6'da kazanç oranı seçim yöntemi formülüze edilmiştir.

$$\text{Bölünme Bilgisi} = \sum_{\text{i=1}}^{\text{v}} \left(\frac{|\text{s}_\text{i}|}{|\text{s}|}\right) \log_2 \left(\frac{|\text{s}_\text{i}|}{|\text{s}|}\right) \quad (5.5)$$

$$\text{Kazanç Oranı} = \text{Bilgi Kazancı} \setminus \text{Bölünme Bilgisi} \quad (5.6)$$

5.1.4 Simetrik Belirsizlik Katsayısı

Bilgi kazancının olumsuz yanını iyileştirmek amacıyla Y ve X'in entropi değerlerinin toplamına bölük simetrik belirsizlik katsayısı hesaplanmaktadır. Simetrik belirsizlik katsayısı 0-1 aralığında değer almaktadır. Simetrik belirsizlik katsayısı 1'e eşit ise X bilgisinin Y bilgisini tahmin edebileceği anlamına gelmektedir. Keza Simetrik belirsizlik katsayısı 0'a eşit olduğunda ise Y ile X arasında hiçbir ilişkisinin olmadığı anlamına gelmektedir (Budak 2018, Forman 2003). Simetrik belirsizlik katsayısı Eşitlik 5.7'de verildiği gibi hesaplanmaktadır.

$$\text{Simetrik Belirsizlik Katsayısı} = 2\frac{\text{Bilgi Kazancı}}{\text{H(Y)+H(X)}} \tag{5.7}$$

5.1.5 Gini Katsayısı Yöntemi

Gini katsayısı, kazanım oranı ve bilgi kazancı yöntemleri alternatif olarak geliştirilmiştir. Bu yöntem diğer yöntemlerden farklı olarak entropi değerini kullanmadan özellik seçimi yapmaktadır. İlk olarak bir etiket değeri ve her bir öznitelik içinde gini katsayısı belirlemektedir. Ardından her bir öznitelik için ayrı ayrı gini katsayısı hesaplanmaktadır (Kaynar vd. 2018). Eşitlik 5.8'de etiket değeri, Eşitlik 5.9'da ise her bir öznitelik için gini katsayısının hesaplanması formülüze edilmiştir.

$$Gini = \prod_{i=1}^{n} p(sınıf = i) \tag{5.8}$$

$$\sum_{i=1}^{n} p(değer = i)\, x \prod_{j=1}^{m} \frac{N(değer=i\ \&\ sınıf=j)}{N(değer=i)} \tag{5.9}$$

5.1.6 Fisher Skoru

Fisher Skor yöntemi, her bir sınıf için ortalama ve standart sapma değerlerini kullanarak bir skor hesaplar. Daha sonra bu skorlar büyükten küçüğe doğru sıralanır ve ardından en üst sıradan başlanılarak özellik seçimi işlemi gerçekleştirmektedir. Eşitlik 5.10'da fisher skor hesaplanması formülüze edilmiştir. Formülde $\mu_i{}^+ - \mu_i{}^-$değerleri sınıfların aritmetik ortalamalarını, $\sigma_i{}^+ - \sigma_i{}^-$ değerleri sınıflara ait standart sapma değerlerini göstermektedir (Yöntem ve Adem 2019, Ferreira ve Figueiredo 2012).

$$F_{x_i} = \frac{|\mu_i{}^+ - \mu_i{}^-|}{|\sigma_i{}^+ - \sigma_i{}^-|} \tag{5.10}$$

5.2 Sarmal Yöntem

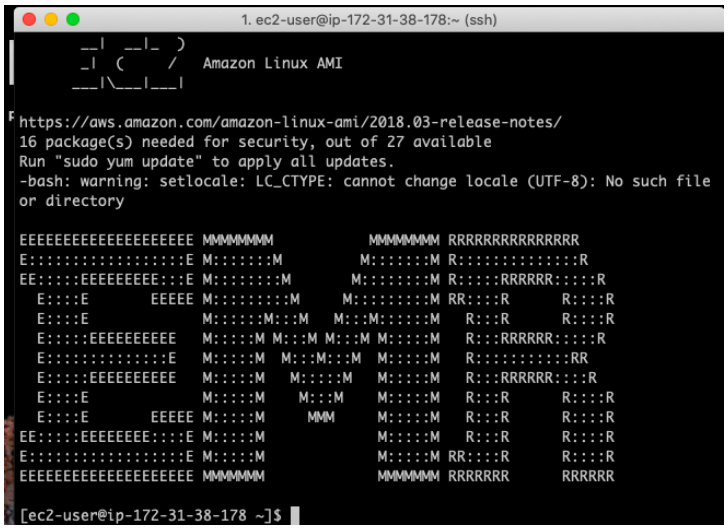
Sarmal yöntemler, istatistiksel yöntemlerden farklı olarak sınıflandırma algoritmasına ihtiyaç duymaktadırlar. Her işlem sırasında sınıflandırıcıya ihtiyaç duymalarından dolayı işlemler diğer özellik seçim yöntemlerine göre daha uzun sürmektedir. Bu durumda performans açısından olumlu sonuçlar vermesine karşın hız ve maliyetlerin açısından zayıf kalabilmektedir (Kaya 2014).

5.3 Gömülü Yöntem

Gömülü yöntemler, sınıflandırma ve özellik seçme işlemlerini bir arada gerçekleştirerek sarmal yöntemlerden ayrışmaktadır. Bilinen en basit sınıflandırıcılardan karar ağaçları yöntemidir. Gömülü yöntemler, hız açısından filtreleme yöntemlere göre yavaş, sarmal yöntemlere göre göre hızlı sonuçlar verebilmektedir (Guyon ve Elisseeff 2003).

6. MATERYAL ve METOT

Bu çalışmada kümeleme analizi gerçekleştirmek için Amazon tarafından sunulan bulut sunucu hizmeti kullanılmıştır. Amazon bulut sunucu hizmeti kurumsal uygulamaları, büyük veri projeleri ve mobil uygulamalara birçok geliştirmenin bulut altyapısında geliştirilmesine imkan sağlayan bir web hizmetleri koleksiyonudur (Kokkinos vd. 2014). Bu bulut hizmet üzerine Amazon elastik bilgi işlem bulutu aktif edilmiştir. Amazon elastik bilgi işlem bulutu ise sanal makine başlatmak ve yönetmek için mekanizmalar sağlayan, belirli bir işletim sistemi, belirli hesaplama, depolama ve ağ özelliklerine sahip bir bulut bilgi işlem ortamıdır (Kokkinos vd. 2014). Bir diğer kullanılan bulut bilgi işlem ortamı Amazon Elastic Map Reduce (EMR) bu sunucu üzerine kurulmuştur. Amazon Elastic Map Reduce (EMR) servisi, Amazon tarafından geliştirilmiş Hadoop, Spark gibi açık kaynaklı büyük veri teknolojilerini içeren ve hızlı bir şekilde verileri işlemek ve yönetmek kullanılan veri işleme platformudur. Sunucu üzerine kurulmuş olan Amazon Elastic Map Reduce (EMR) servisi Şekil 6.1'de sunulmuştur.



Şekil 6.1 Amazon sunucuya kurulmuş EMR.

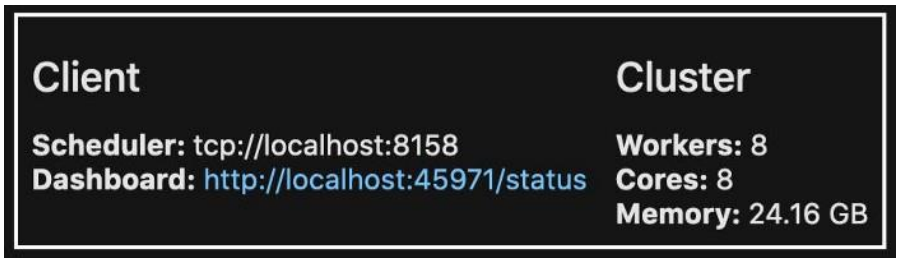
Bu çalışmada, amazon bulut sunucu hizmetlerinden "f1.4xlarge" paketine sahip "Amazon EC2" elastik bilgi işlem bulutu kullanılmıştır. Amazon bulut sunucu özellikleri Çizelge 6.1'de sunulmuştur.

Çizelge 6.1 Amazon bulut sunucu özellikleri.

Sunucu Özellikleri
Hızlandırılmış “f1.4xlarge” Amazon EC2 Bulut Sunucusu, Linux işletim sistemi, 16vCPU, 244 GB RAM, 940 GB SSD HDD

Kümeleme analizlerini gerçekleştirmek için Python programlama dili tercih edilmiştir. Python, 1990 yılında Guido Van Rossum tarafından geliştirilen açık kaynaklı ve fonksiyonel bir programlama dilidir. Büyük veri analizi, veri madenciliği, görüntü işleme gibi bir çok alanda kullanım kolaylığı sunan bir çok kütüphanesi bulunmaktadır. Bu kütüphaneler dünyanın farklı yerlerinden gönüllü geliştirici tarafından geliştirilmekte ve ücretsiz bir şekilde kullanıma sunulmaktadır. Aynı zamanda basit, sade ve anlaşılır ara yüz desteği sunmaktadır. Birçok özelliği ile Python, dünyadaki en popüler programlama dillerinden biri arasında gösterilmektedir (Korkmaz 2020, Severance 2015).

Kümeleme analizinde kullanılan Python kütüphanelerinden ilki Python Dask kütüphanesi kullanılmıştır. Dask, ana belleğe sığmayan veri kümelerinde paralel sunucu olarak çalışabilen üst düzey Array, Bag ve DataFrame koleksiyonları sağlayan Python kütüphanesidir (Rocklin 2015). Dask yardımı ile 8 çekirdekli ve 24 GB RAM olan özelliğe sahip paralel 8 bir sunucu oluşturulmuştur. Sunucu özellikleri Şekil 6.2’de sunulmuştur.



Şekil 6.2 8 Çekirdekli ve 24 GB RAM özelliğe sahip paralel 8 sanal sunucu.

Kofenetik Korelasyon katsayısını hesaplamak için geliştirilmiş Python dili için geliştirilmiş "Sicikit-Learn" kütüphanesi kullanılmıştır. Sicikit-Learn; doğrusal regresyon, lojistik regresyon, karar ağaçları vb. bir çok veri madenciliği süreçlerinde kullanılan temel yöntemleri içeren bu Python kütüphanesidir (Sönmez ve Zengin 2019). "Sicikit-Learn" paketinin desteklediği kümeleme yöntemler; "TEBKY", "TABKY", "OBKY", "Ward" uzaklık ölçütleri ise "canbera", "minkowski" ve "Öklid" olduğu için bu çalışmada Kofenetik Korelasyon katsayıları bu yöntemler için hesaplanmıştır.

Bu çalışmada veri seti olarak, ABD Ulaştırma Bakanlığı tarafından yayınlanan 2015 Hava Seyahat Tüketici Raporundaki veri seti kullanılmıştır. Bu veri seti ücretsiz ve açık erişim olarak yayımlanmıştır (İnt.Kyn.3). Veri seti 5.819.079 satır 31 sütundan oluşmaktadır. Veri setine ilişkin detaylı açıklama Çizelge 6.2'de sunulmuştur.

Çizelge 6.2 Değişkenlere ilişkin bilgiler.

Değişken	**Değişken (Türkçe Açıklaması)**	**Birimi**
YEAR	Yıl	Yıl
MONTH	Ay	Ay
DAY	Gün	Gün
DAY_OF_WEEK	Haftanın Günü	Gün
AIRLINE	Havayolu	Metin
FLIGHT_NUMBER	Uçuş Numarası	Numara
TAIL_NUMBER	Kuyruk Numarası	Numara
ORIGIN_AIRPORT	Kalkış Havalimanı	Metin
DESTINATION_AIRPORT	Varış Havalimanı	Metin
SCHEDULED_DEPARTURE	Programlı Kalkış Saati	Saat
DEPARTURE_TIME	Kalkış Saati	Saat
DEPARTURE_DELAY	Kalkış Gecikmesi	Saat
TAXI_OUT	Taksi Çıkışı	Dakika
WHEELS_OFF	Tekerleklerin Kapama Süresi	Dakika
SCHEDULED_TIME	Planlanmış Kalkış Zaman	Dakika
ELAPSED_TIME	Uçuş Zamanı	Dakika

Çizelge 6.2 (Devamı) Değişkenlere ilişkin bilgiler.

Değişken	Değişken (Türkçe Açıklaması)	Birimi
TAXI_IN	Taksi Girişi Zamanı	Dakika
AIR_TIME	Tekerleklerin Kapama İle Açılma Zamanı Arasında Geçen Zaman	Dakika
DISTANCE	Mesafe	Kilometre
WHEELS_ON	Tekerlekleri Kapama	Saat
SCHEDULED_ARRIVAL	Programlı Varış	Saat
ARRIVAL_TIME	Varış Zamanı	Saat
ARRIVAL_DELAY	Gecikme Zamanı	Saat
DIVERTED	Yönlendirme Durumu	Metin
CANCELLED	İptal Durumu	Metin
CANCELLATION_REASON	İptal Nedeni	Metin
AIR_SYSTEM_DELAY	Hava Sistemi Nedeniyle Gecikme	Dakika
SECURITY_DELAY	Güvenlik Nedeniyle Gecikme	Dakika
AIRLINE_DELAY	Havayolunda Kaynaklı Gecikme	Dakika
LATE_AIRCRAFT_DELAY	Piste Geç Gelme Süresi	Dakika
WEATHER_DELAY	Hava Durumu	Dakika

Veri setinde 12 tane havayolu şirketi bulunmaktadır. Bu havayollarının isimleri Çizelge 6.3'de sunulmuştur.

Çizelge 6.3 Havayolu şirketlerine ilişkin bilgiler.

Kısaltma	Açıklaması
UA	United Airlines
AA	American Airlines
US	US Airlines
F9	Frontier Airlines
B6	JetBlue Airlines
OO	Skywest Airlines
AS	Alaska Airlines
NK	Spirit Airlines

Çizelge 6.3 (Devamı) Havayolu şirketlerine ilişkin bilgiler.

Kısaltma	Açıklaması
WN	Southwest Airlines
DL	Delta Airlines
EV	Atlantic Southeast Airlines
HA	Hawaiian Airlines

Veri setinde 323 tane havalimanı bulunmaktadır. Çizelge 6.4'de alfabetik kodlama sırasına göre ilk 10 havalimanı sunulmuştur.

Çizelge 6.4 Havalimanı açıklamalarına ilişkin bilgiler.

Kısaltma	Açıklaması
ABE	Lehigh Valley International Airport
ABI	Abilene Regional Airport
ABQ	Albuquerque International Sunport
ABR	Aberdeen Regional Airport
ABY	Southwest Georgia Regional Airport
ACK	Nantucket Memorial Airport
ACT	Waco Regional Airport
ACV	Arcata Airport
ACY	Atlantic City International Airport
ADK	Adak Airport
ABE	Lehigh Valley International Airport
ABI	Abilene Regional Airport
ABQ	Albuquerque International Sunport
ABR	Aberdeen Regional Airport

Sunucu ve Python kütüphane kurulum süreçlerinin tamamlanmasından sonra kümeleme analizine geçilmiştir. Kümeleme analizine başlamadan önce ise veri setindeki sonuca etki etmeyecek değişkenleri belirlemek için özellik seçimi işlemi yapılmıştır. Bu özellik seçimi, sonuca etki etmeyecek değişkenler veri setinden çıkartıldığı için daha fazla

gözlem içeren veri seti ile çalışma imkanı sağlamıştır. Burada filtreleme yöntemlerinden Korelasyon tabanlı özellik seçimi tercih edilmiştir. Özellik seçiminde kullanılan Python kodu şekil 6'da sunulmuştur.

```
correlated_features = set()
correlation_matrix = onbin_data.corr()
```

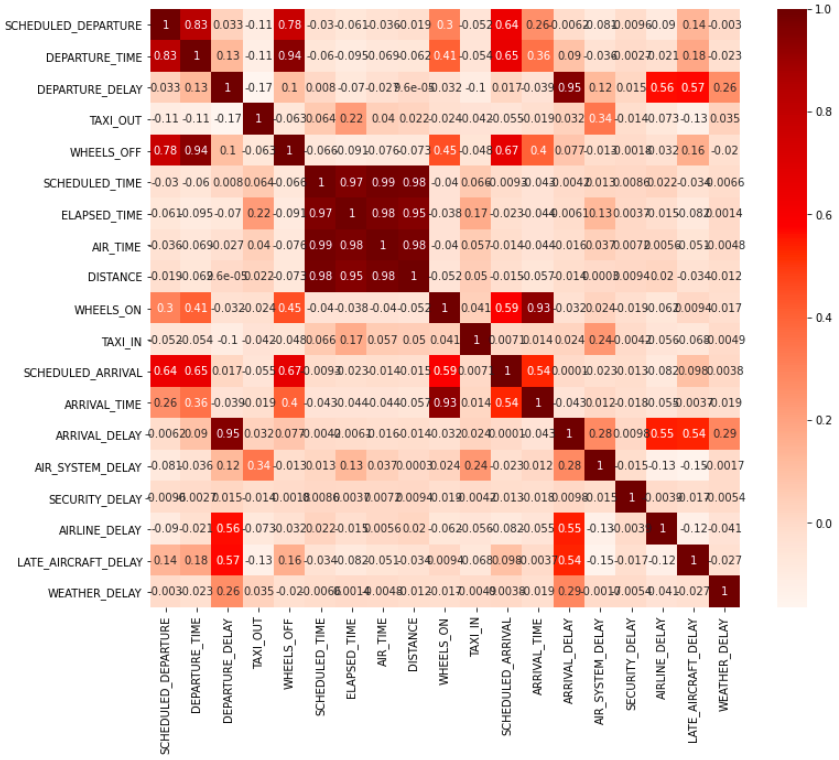
```
for i in range(len(correlation_matrix .columns)):
    for j in range(i):
        if abs(correlation_matrix.iloc[i, j]) > 0.8:
            colname = correlation_matrix.columns[i]
            correlated_features.add(colname)
```

Şekil 6.3 Özellik seçiminde kullanılan Python kodu.

Aralarındaki korelasyon katsayısı 0,8'den küçük olan değişkenler veri setinden çıkartılmıştır. Veri setinde kalan değişkenler Çizelge 6.5'de, Bu değişkenlerin birbirleri ile arasındaki korelasyon grafiği ise Şekil 6.4'de sunulmuştur.

Çizelge 6.5 Özellik seçime ilişkin sonuçlar.

Değişken	**Değişken (Türkçe Açıklaması)**	**Birimi**
TAXI_IN	Taksiye Girişi Süresi	Dakika
TAXI_OUT	Taksi Çıkış Süresi	Dakika
WHEELS_OFF	Tekerlekler Kapama Süresi	Dakika
AIR_TIME	Tekerleklerin Kapama İle Açılma Zamanı Arasında Geçen Zaman	Dakika
DISTANCE	Mesafe	Kilometre
ARRIVAL_DELAY	Toplam Gecikme Süresi	Dakika
ELAPSED_TIME	Uçuş Süresi	Dakika



Şekil 6.4 Değişkenlerin birbirleri ile arasındaki korelasyon grafiği

Özellik seçimi sonucunda veri setinden diğer değişkenler çıkartılarak yeni bir veri seti oluşturulmuştur. Bu veri setinin çok değişkenli normallik varsayımları sağlanmıştır. Daha sonra değişkenlerin birimleri farklı olduğundan değişkenler standardize edilmiştir. Değişkenler standardize etme kullanılan Python kodu şekil 6.5'de sunulmuştur.

```
def normalize_col(col):
  return (col-col.min())/(col.max()-col.min())

features_df = df
normalized_data = features_df.apply(lambda x: normalize_col(x), axis=1)
```

Şekil 6.5 Değişkenler standardize etme kullanılan Python kodu.

Daha sonra bu veri setini temsil edecek ve belleğe sığabilecek özellikte rastgele seçme yöntemi ile 4 farklı veri seti oluşturulmuştur. Tüm veri setlerinde gözlem sayısı rastgele seçilmiştir. Oluşturulan 1.veri seti toplam veri setinden çıkartılmış şekilde 2.veri seti oluşturulmuştur. Bu yöntem diğer veri setlerinin oluşturulmasında da kullanılmıştır. Bu sayede farklı veri setleri oluşturulmuştur. Seçilen kümelere ait gözlem ve değişken sayıları Çizelge 6.6'da sunulmuştur.

Çizelge 6.6 Seçilen 4 kümeye ait gözlem ve değişken sayıları.

Küme	Gözlem Sayısı	Değişkenler	Havayolu Şirketleri
1.Veri Seti	10,859	*Taksiye Girişi Süresi *Taksi Çıkış Süresi *Tekerlekler Kapama Süresi Tekerleklerin Kapama İle Açılma Zamanı Arasında Geçen Zaman *Mesafe *Toplam Gecikme Süresi *Uçuş Süresi	*United Airlines *American Airlines *US Airlines *Frontier Airlines *JetBlue Airlines *Skywest Airlines *Alaska Airlines *Spirit Airlines *Southwest Airlines *Delta Airlines *Atlantic Southeast Airlines *Hawaiian Airlines
2. Veri Seti	51,428		
3. Veri Seti	72,553		
4. Veri Seti	108,568		

Tüm bu işlemler tamamlandıktan sonra kümeleme analiz sürecine geçilmiştir. İlk olarak 1. veri setinin Kofenetik korelasyon katsayıları hesaplanmıştır. Kofenetik korelasyon katsayısının en yüksek değere ulaştığı kümeleme yöntemi ve uzaklık ölçütü belirlenmiştir. Bu işlemler diğer veri setlerinde uygulanmıştır.

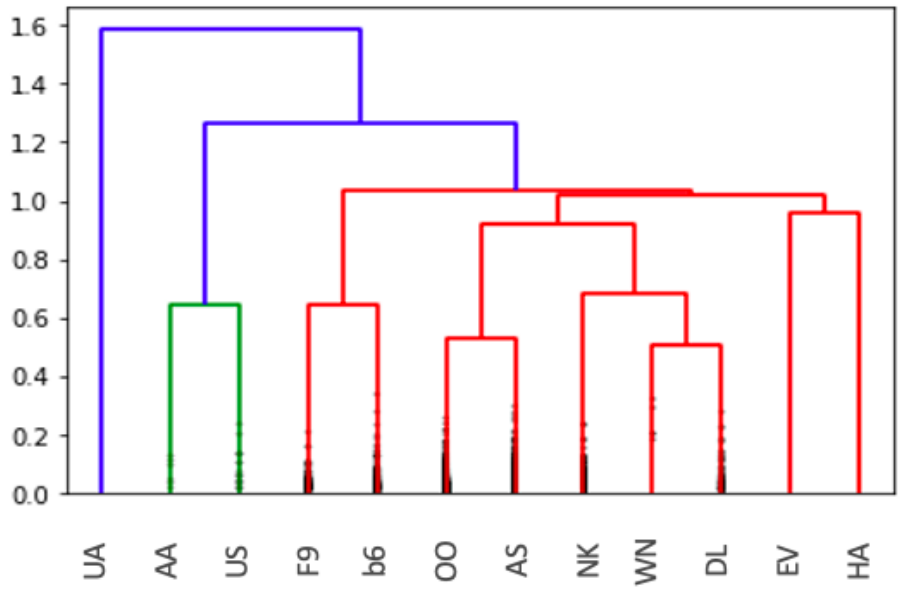
7. BULGULAR

1.veri setinde Kofenetik Korelasyon katsayısı; kümeleme yöntemi OBKY, uzaklık ölçütünde ise Öklid olduğu durumda en yüksek sonucu verdiği gözlemlenmiştir. Sonuçlar Çizelge 7.1'de gösterilmiştir.

Çizelge 7.1 1.Veri setindeki Kofenetik korelasyon katsayıları.

Uzaklık Ölçütleri	Kümeleme Yöntemi	Kofenetik Korelasyon
Öklid	TEBKY	0,577
Öklid	TABKY	0,698
Öklid	**OBKY**	**0,783**
Öklid	Centroid	0,757
Öklid	Ward	0,480
Canberra	TEBKY	0,608
Canberra	TABKY	0,575
Canberra	OBKY	0,773
Minkowski	TEBKY	0,577
Minkowski	TABKY	0,698
Euclidean	OBKY	0,577

Kümeleme yöntemi OBKY, uzaklık ölçütü Öklid olduğu durumun Dendrogram grafiği incelendiğinde 11 birim uzaklık değeri ile Havayolu şirketleri 3 kümeye ayrıldığı görülmektedir. Bu kümeler incelediğinde, UA (United Airlines) tek başına bir kümede, AA (American Airlines) ve US (US Airways) birlikte bir kümede, F9 (Frontier Airlines), B6 (JetBlue Airlines), OO (Skywest Airlines), AS (Alaska Airlines), NK (Spirit Airlines), WN (Southwest Airlines), DL (Delta Airlines), EV (Atlantic Southeast Airlines) ve HA (Hawaiian Airlines) havayolu şirketlerinin diğer kümede yer almaktadır. Kümeleme yöntemi OBKY, uzaklık ölçütü Öklid olduğu durumun Dendrogram grafiği Şekil 7.1'de sunulmuştur.



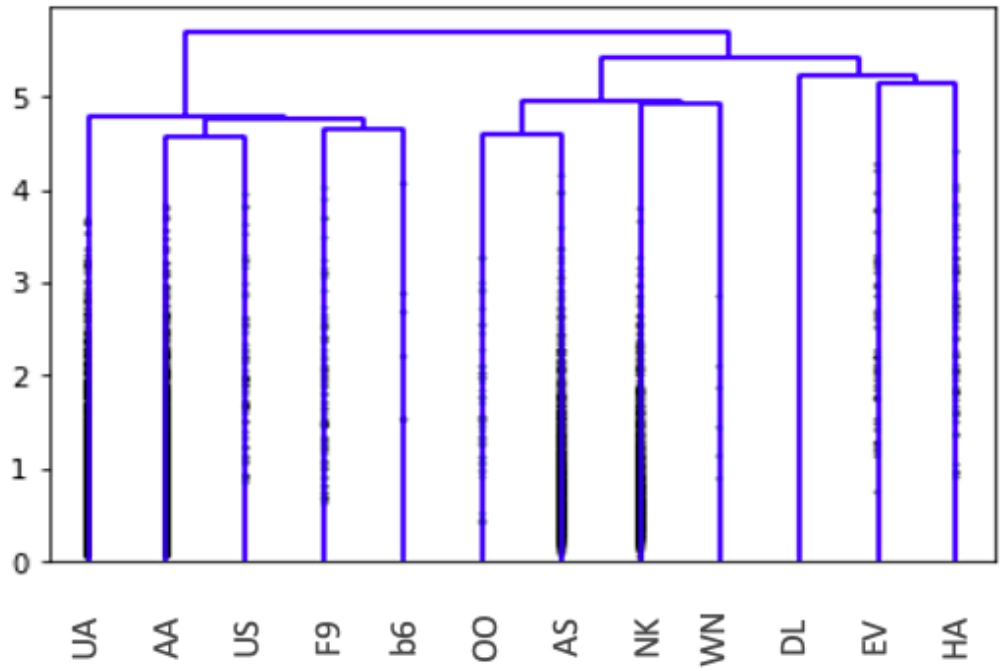
Şekil 7.1 Kümeleme yöntemi OBKY, uzaklık ölçütü Öklid olduğu durumdaki Dendrogram grafiği.

2.veri setinde Kofenetik Korelasyon katsayısı; kümeleme yöntemi OBKY, uzaklık ölçütünde ise Canberra olduğu durumda en yüksek sonucu verdiği gözlemlenmiştir. Sonuçlar Çizelge 7.2'de sunulmuştur.

Çizelge 7.2 2.Veri setindeki Kofenetik korelasyon katsayıları.

Uzaklık Ölçütleri	Kümeleme Yöntemi	Kofenetik Korelasyon
Öklid	TEBKY	0,524
Öklid	TABKY	0,644
Öklid	OBKY	0,753
Öklid	Centroid	0,750
Öklid	Ward	0,574
Canberra	TEBKY	0,597
Canberra	TABKY	0,588
Canberra	**OBKY**	**0,764**
Minkowski	TEBKY	0,524
Minkowski	TABKY	0,644
Minkowski	OBKY	0,751

Kümeleme yöntemi OBKY, uzaklık ölçütü Öklid olduğu durumun Dendrogram grafiği incelendiğinde 6 birim uzaklık değeri ile Havayolu şirketleri 2 kümeye ayrıldığı görülmektedir. Bu kümeler incelediğinde, UA (United Airlines), AA (American Airlines), US (US Airways), F9 (Frontier Airlines) ve B6 (JetBlue Airlines) birlikte bir küme de, OO (Skywest Airlines), AS (Alaska Airlines), NK (Spirit Airlines), WN (Southwest Airlines), DL (Delta Airlines), EV (Atlantic Southeast Airlines) ve HA (Hawaiian Airlines) havayolu şirketlerinin diğer kümede yer almaktadır.. Kümeleme yöntemi OBKY, uzaklık ölçütü Canberra olduğu durumun Dendrogram grafiği Şekil 7.2'de sunulmuştur.



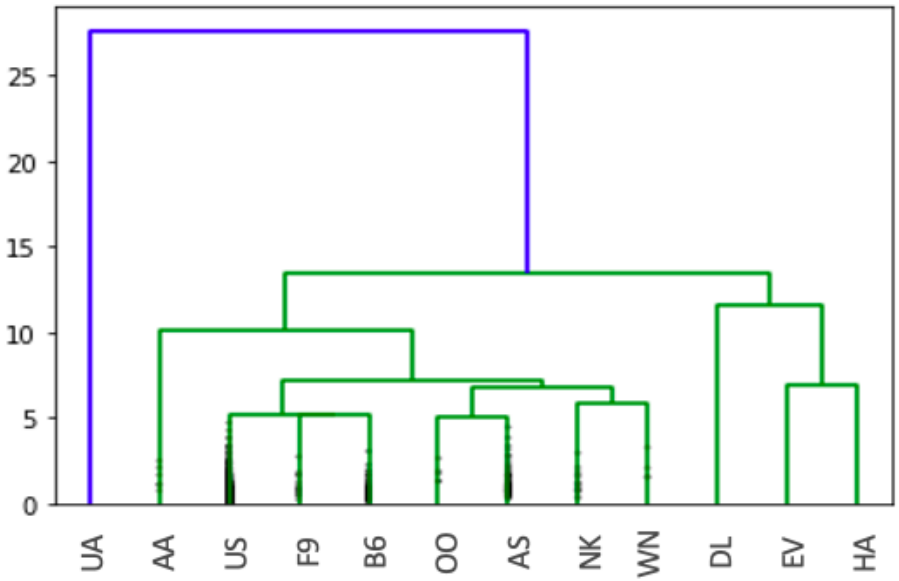
Şekil 7.2 Kümeleme yöntemi OBKY, uzaklık ölçütü Canberra olduğu durumdaki Dendrogram grafiği.

3.veri setinde Kofenetik Korelasyon katsayısı; kümeleme yöntemi OBKY, uzaklık ölçütünde ise Öklid olduğu durumda en yüksek sonucu verdiği gözlemlenmiştir. Sonuçlar Çizelge 7.3'de sunulmuştur.

Çizelge 7.3 3.Veri setindeki Kofenetik korelasyon katsayıları.

Uzaklık Ölçütleri	Kümeleme Yöntemi	Kofenetik Korelasyon
Öklid	TEBKY	0,510
Öklid	TABKY	0,671
Öklid	**OBKY**	**0,774**
Öklid	Centroid	0,765
Öklid	Ward	0,542
Canberra	TEBKY	0,612
Canberra	TABKY	0,554
Canberra	OBKY	0,768
Minkowski	TEBKY	0,510
Minkowski	TABKY	0,671
Minkowski	OBKY	0,771

Kümeleme yöntemi OBKY, uzaklık ölçütü Öklid olduğu durumun Dendrogram grafiği incelendiğinde 15 birim uzaklık değeri ile Havayolu şirketleri 2 kümeye ayrıldığı görülmektedir. Bu kümeler incelediğinde, UA (United Airlines) tek başına bir kümede, diğer hava yolu şirketleri AA (American Airlines) ve US (US Airways) birlikte bir kümede, F9 (Frontier Airlines), B6 (JetBlue Airlines), OO (Skywest Airlines), AS (Alaska Airlines), NK (Spirit Airlines), WN (Southwest Airlines), DL (Delta Airlines), EV (Atlantic Southeast Airlines) ve HA (Hawaiian Airlines) tek kümede yer almaktadır. Kümeleme yöntemi OBKY, uzaklık ölçütü Öklid olduğu durumun Dendrogram grafiği Şekil 7.3'de sunulmuştur.



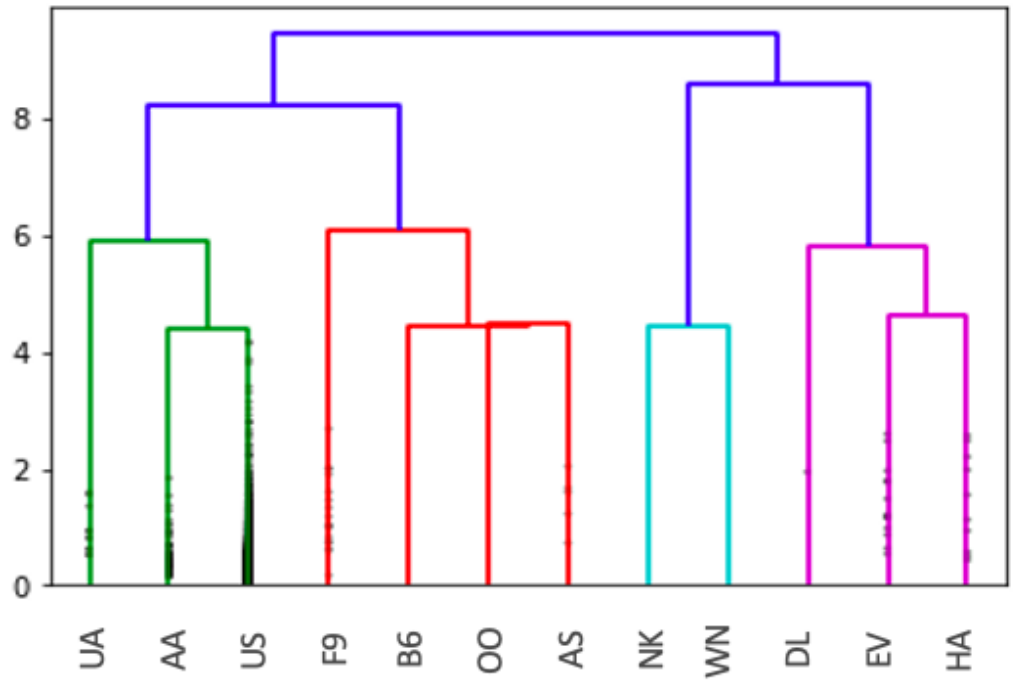
Şekil 7.3 Kümeleme yöntemi OBKY, uzaklık ölçütü Öklid olduğu durumdaki Dendrogram grafiği.

4.veri setinde Kofenetik Korelasyon katsayısı; kümeleme yöntemi OBKY, uzaklık ölçütünde ise Centroid olduğu durumda en yüksek sonucu verdiği gözlemlenmiştir. Sonuçlar Çizelge 7.4'de sunulmuştur.

Çizelge 7.4 4.Veri setindeki Kofenetik korelasyon katsayıları.

Uzaklık Ölçütleri	Kümeleme Yöntemi	Kofenetik Korelasyon
Öklid	TEBKY	0,492
Öklid	TABKY	0,717
Öklid	OBKY	0,760
Öklid	**Centroid**	**0,779**
Öklid	Ward	0,465
Canberra	TEBKY	0,579
Canberra	TABKY	0,555
Canberra	OBKY	0,768
Minkowski	TEBKY	0,492
Minkowski	TABKY	0,717
Minkowski	OBKY	0,750

Dendrogram grafiği incelediğinde, Kümeleme yöntemi Centroid, uzaklık ölçütü Öklid olduğu durumda 8 birim uzaklık değeri ile 4 kümeye ayrıldığı görülmektedir. Bu kümeler incelediğinde UA (United Airlines), AA (American Airlines) ve US (US Airlines) birlikte bir kümede, F9 (Frontier Airlines) B6 (JetBlue Airlines), OO (Skywest Airlines Inc.) ve AS (Alaska Airlines) birlikte bir kümede, NK (Spirit Airlines) ve WN (Southwest Airlines) birlikte bir kümede ve DL (Delta Airlines), EV (Atlantic Southeast Airlines) ve HA (Hawaiian Airlines) bir kümede yer almaktadır. Kümeleme yöntemi Centroid, uzaklık ölçütü Öklid olduğu durumun Dendrogram grafiği Şekil 7.4'de sunulmuştur.



Şekil 7.4 Kümeleme yöntemi Centroid, uzaklık ölçütü Öklid olduğu durumdaki Dendrogram grafiği.

8. TARTIŞMA ve SONUÇ

Bu çalışmada büyük veri teknolojilerini kullanarak büyük veride hiyerarşik kümeleme yöntemleri Kofenetik korelasyon katsayısı ile karşılaştırılmıştır.

Amazon tarafından sunulan bulut sunucu hizmetlerinden elastik bilgi işlem bulut sunucusu kurulmuştur. Bu sunucu üzerine büyük veri işlemeyi kolaylaştırmak amacıyla Hadoop, Spark gibi açık kaynaklı büyük veri teknolojilerini içeren Amazon Elastic Map Reduce (EMR) servisi aktif edilmiştir.

Kümeleme analizinde Python için geliştirilmiş kütüphaneler kullanılmıştır. Bunlardan birincisi Dask kütüphanesidir. Dask ana belleğe sığmayan veri kümelerinde paralel olarak çalışabilen sanal sunucu oluşturmak için kullanılmıştır. Diğeri ise Sicikit-Learn kütüphanesidir. Sicikit-Learn kütüphanesi kümeleme analizi gerçekleştirme ve Kofenetik Korelasyon katsayılarını hesaplamak için kullanılmıştır.

Veri seti olarak, ABD Ulaştırma Bakanlığı tarafından yayınlanan 2015 Hava Seyahat Tüketici Raporundaki veri seti kullanılmıştır. Bu veri setinin çok değişkenli normallik varsayımları sağlanmıştır. Değişkenlerin birimleri farklı olduğundan değişkenler standardize edilmiştir.

Kümeleme analizine başlamadan önce ise veri setindeki sonuca etki etmeyecek değişkenleri belirlemek için özellik seçimi işlemi yapılmıştır. Yöntem filtreleme özellik seçiminin alt yöntemi olan Korelasyon tabanlı özellik seçimi kullanılmıştır. Bu noktada korelasyon katsayısı 0,8'den küçük olan değişkenler veri setinden çıkartılmıştır.

Daha sonra veri seti içerisinden ana kütleye temsilen rastgele seçim yöntemiyle 4 farklı veri seti oluşturulmuştur. Her veri setinde uzaklık ölçütleri hesaplanarak kümeleme metotları uygulanmıştır. En son aşama ise Kofenetik korelasyon katsayısının en yüksek değeri verdiği kümeleme yöntemi ve uzaklık ölçütü belirlenmiştir.

Çalışma sonucunda 1.veri setinde kümeleme yöntemi ortalama bağlantı kümeleme, uzaklık ölçütü ise Öklid olduğu durumda Kofenetik korelasyon katsayısı en yüksek sonucu verdiği gözlemlenmiştir. 2.veri setinde kümeleme yöntemi ortalama bağlantı kümeleme, uzaklık ölçütü ise Canberra olduğu durumda Kofenetik korelasyon katsayısı en iyi sonucu verdiği gözlemlenmiştir. 3.veri setinde kümeleme yöntemi ortalama bağlantı kümeleme, uzaklık ölçütü ise Öklid olduğu durumda Kofenetik korelasyon katsayısı en iyi sonucu verdiği gözlemlenmiştir. 4.veri setinde ise kümeleme yöntemi Centroid, uzaklık ölçütü ise Öklid olduğu durumda Kofenetik korelasyon katsayısı en iyi sonucu verdiği gözlemlenmiştir. Çalışma sonucunda Kofenetik korelasyon Katsayısının ortalama bağlantı kümeleme yönteminde en yüksek sonucu verdiği gözlemlenmiştir.

Daha önce bu konuda yapılan çalışmalar incelendiğinde (Silva ve Dias 2013, Carvalho vd. 2019, Kumar ve Toshniwal 2016, Ponde ve Shirwaikar 2016, Saraçli vd. 2013) Kofenetik korelasyon katsayısının ortalama bağlantı yönteminde en yüksek sonucu verdiği gözlemlenmiştir. Önceki yapılmış çalışmaların ışığında tasarlanan bu çalışmanın büyük veri teknolojilerini kullanarak, büyük veride en iyi kümeleme yöntemini belirlemeye yönelik olması ve sonuçları itibariyle literatürü destekleyici olması nedeniyle literatüre katkı sağlayacağı öngörülmektedir.

Uygulayıcıların büyük veri setinde kümeleme analizi yapmaları halinde karşılaştıkları temel sorun olan donanımsal yetersizliğin Amazon EMR, Python ve Dask ile aşılmasının mümkün olduğu anlaşılmıştır ve bu yöntem önerilmektedir.

Yüksek miktarda verilerin işlenmesinde Özellik Seçimi kullanılması halinde sonucu etkilemeyecek değişkenler çıkartılarak daha hızlı ve daha fazla gözlem yoluyla çalışma imkânı sağlanacağı için bu yöntemin kullanılması önerilmektedir.

Çalışmadan elde edilen bulgular doğrultusunda, farklı büyük veri (pazarlama, e-ticaret vb.) setlerinde hem akademisyenler hem de sektör uygulayıcıları tarafından ortalama bağlantı yönteminin kullanılması önerilmektedir. Gelecekteki çalışmalarında farklı sektörleri kapsaması ve farklı büyük veri tipleri bu yöntemin kullanmaları önerilmektedir.

9. KAYNAKLAR

Akın Y K, 2008, Veri Madenciliğinde Kümeleme Algoritmalari ve Kümeleme Analizi, Marmara Ünversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 164s, İstanbul.

Aktan E, 2018, Büyük Veri: Uygulama Alanları, Analitiği ve Güvenlik Boyutu, Bilgi Yönetimi , 1(1), 1–22.

Aldenderfer M S, Blashfield R K, 1984, Cluster Analysis: Quantitative Applications in the Social Sciences, 43p, Beverly Hills.

Altunışık R, 2015, Büyük Veri: Fırsatlar Kaynağı mı Yoksa Yeni Sorunlar Yumağı mı?, Yıldız Social Science Review, 1, 45–76.

Altındiş S, Kıran M İ, 2018, Sağlık Hizmetlerinde Büyük Veri. Ömer Halisdemir Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 11(2), 257–271.

Aslan Ü, Özerhan Y, 2017, Big Data, Muhasebe ve Muhasebe Mesleği, Muhasebe Bilim Dünyası Dergisi, 19(4), 862–883.

Bakırarar B, 2016, Sağlık Alanında Büyük Veri ve Veri Madenciliği Yöntemlerinin Kullanımı, Ankara Üniversitesi, Sağlık Bilimleri Enstitüsü, Yüksek Lisans Tezi, 72s, Ankara.

Bekar E T, Nyqvist P, Skoogh A, 2020, An Intelligent Approach for Data Pre-Processing and Analysis in Predictive Maintenance with an Industrial Case Study, Advances ın Mechanical Engineering, 12(5), 1–14.

Bhathal G S, Singh A, 2019, Big data: Hadoop Framework Vulnera-Bilities, Security İssues and Attacks, Elsevier, 100002, 1–8.

Bilgiç E , Türkmenoğlu M , Bozoğlu B G, 2019, İş Analitiği ve Değer Zinciri: Detaylı ve Sistematik Bir Literatür Taraması, Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 54, 1–24.

Budak H, 2018, Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 22, 21–31.

Cavanillas J M, Curry E, Wahlster W, 2016, New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe. Springer Open, 312p, Switzerland.

Carvalho P R, Munita C S, Lapolli A L, 2019, Validity Studies Among Hierarchical Methods of Cluster Analysis Using Cophenetic Correlation Coefficient, Brazilian Journal of Radiation Sciences, 7, 1–14.

Choi S, Cha S, Tappert C, 2010, A Survey of Binary Similarity and Distance Measures. Journal of Systemics, Cybernetics and Informatics, 8(1), 43–48.

Cibaroğlu M, Yalçınkaya B, 2019, Belge ve Arşiv Yönetimi Süreçlerinde Büyük Veri Analitiği ve Yapay Zeka Uygulamaları, Bilgi Yönetimi, 2(1), 44-58.

Çelik S, 2017, Büyük Veri Teknolojilerinin İşletmeler İçin Önemi, Social Sciences Studies Journal, 3(6), 873–883.

Çelik S, Akdamar E, 2018, Büyük Veri ve Veri Görselleştirme, Akademik Bakış Uluslararası Hakemli Sosyal Bilimler Dergisi, 65, 253–264.

Çelik Ş, 2013, Kümeleme Analizi ile Sağlık Göstergelerine Göre Türkiye'deki İllerin Sınıflandırılması, Doğuş Üniversitesi Dergisi, 14(2), 175–194.

Demirtaş B, Argan M, 2015, Büyük Veri ve Pazarlamadaki Dönüşüm: Kuramsal Bir Yaklaşım, Pazarlama ve Pazarlama Araştırmaları Dergisi, 15, 1–21.

Derya B, 2019, Farklı Bağlantı Yöntemleri ile Hiyerarşik Kümeleme Topluluğu, Selçuk Üniversitesi Mühendislik, Bilim ve Teknoloji Dergisi, 7(1), 154–164.

De Witt D J, Gray J, 1992, Parallel Database Systems: The Future of High Performance Database Processing,, Communications of the ACM, 35(6), 85–98.

Doğan İ, 2002, Selectionby Cluster Analysis, TurkishJournal of Veterinary and Animal Sciences, 26(1), 47–53.

Emhan Ö, Akın M, 2019, Filtreleme Tabanlı Öznitelik Seçme Yöntemlerinin Anomali Tabanlı Ağ Saldırısı Tespit Sistemlerine Etkisi, Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10(2), 549–559.

Everitt B, Landau S, Leese M, Stahl D, 2011, Cluster Analysis, Wiley, 346p, Chichester.

Faghri F, Bazarbayev S, Overholt M, Farivar R, Campbell R H, Sanders W H, 2013, Failure Scenario As A Service (Fsaas) for Hadoop Clusters. In: Proceedings Of The Workshop on Secure and Dependable Middleware for Cloud Monitoring and Management ACM, 1–11, Oct, China.

Ferreira A J, Figueiredo M A T, 2012, Efficient Feature Selection Filters for High Dimensional Data, Pattern Recognit Lett, 33(13), 1794–1804.

Forman G, 2003, An Extensive Empirical Study of Feature Selection Metrics for Text Classification, Journal of Machine Learning Research, 3, 1289–1305.

Fırat M, Dursun Ö, Aydoğdu M , Dikbaş F, 2013, Hiyerarşik Olmayan Kümeleme Yöntemi ile Türkiye Akarsularındaki Askı Maddesi Konsantrasyonu ve Miktarının Sınıflandırılması, Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 2(1), 61–67.

Fırat S Ü, 1997, Kümeleme Analizi: İstihdamın Sektörel Yapısı Açısından Avrupa Ülkelerinin Karşılaştırılması, İstanbul Üniversitesi Sosyal Bilimler Dergisi, 2(3), 50–59.

Fikri N, Rida M, Abghour N, Moussaid K, Omri A E, 2019, An adaptive and real-time based architecture for financial data integration, Journal of Big Data, 6(97), 2-25.

Gil G D, Gallego S R, García S, Herrera F, 2017, A Comparison on Scalability For Batch Big Data Processing on Apache Spark and Apache Flink, Big Data Analytics, 2, 1–11.

Ghazi M R, Gangodkar D, 2015, Hadoop, MapReduce and HDFS, A Developers Perspective Procedia Computer Science, 48, 45–50.

Gonzolez J, 2012, Parallel and Distributed Systems for Probabilistic Reasoning, Carnegie Mellon University, Machine Learning Department School of Computer Science, Doctoral Thesis, 181p, Pittsburgh.

Gupta A, Gupta M K, 2017, HIVE- Processing Structured Data in HADOOP, International Journal of Scientific & Engineering Research, 8(6), 45–48.

Guyon I, Elisseeff A, 2003, An Introduction to Variable and Feature Selection, The Journal of Machine Learning Research, 3, 1157–1182.

Gümüşçü A, Aydilek İ B, Taşaltın R, 2016, 3 Farklı Filtre Modelli Öznitelik Seçme Algoritmalarının Kombine Edilerek İyileştirilmesi, Afyon Kocatepe University Journal of Science and Engineering, 16, 31–35.

Holmes G, Nevill-Manning C, 1995, Feature Selection via The Discovery Of Simple Classification Rules, to AppearIi Proceedings of Symposium on Intelligent Data Analysis (IDA– 95), 17–19.

Jaskowiak P A , Campello R G B , Costa I V, on the selection of Appropriate Distances for Gene Expression Data Clustering, Twelfth Asia Pacific Bioinformatics Conference (APBC 2014), 2014, 17–19, Jan, Shanghai, 2–17.

Johnson R A, Wichern D W W, 1998, Applied Multivariate Statistical Analysis, Prentice Hall, 767p, New Jersey.

Karakoç Ö, 2019, Evaluation of Development Levels of the Provinces With Grey Cluster Analysis Based on Human Development Index, Marmara University, Institute For Graduate Studies in Pure and Applied Sciences, Master Thesis, 70s, İstanbul.

Kaya İ, Ateş S, Akbulut D, Köksal A, 2017, Büyük Veri, Veri Analitiği ve Veri Analizi Işığında Muhasebe Eğitimi: Ders İçerikleri Üzerine Bir Araştırma, 36. Muhasebe Eğitimi Sempozyumu Kitabı, Matsis Matbaa, İstanbul.

Kaya M, Aydoğan T, 2019, Hadoop Map/Reduce Yöntemi ile Klasik Veri Okuma Tekniği Arasında Bir Performans Karşılaştırılması, Uluslararası Teknolojik Bilimler Dergisi, 10(3), 10–19

Kaya M, 2014, Gen İfade Verilerinde Öznitelik Seçimi ve Sınıflandırma, Gazi Ünversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 97s, Ankara.

Kayaalp T, Yazgan E, Şahinler S, 2000, Aşamalı Kümeleme Analizi Yöntemlerinin Karşılaştırmalı Olarak İncelenmesi, İstatistik Araştırma Sempozyumu, 27–29 Kasım, Ankara, 154–163.

Kaynar O, Arslan H, Görmez Y, Işık Y E, 2018, Makine Öğrenmesi ve Öznitelik Seçim Yöntemleriyle Saldırı Tespiti, Bilişim Teknolojileri Dergisi, 11(2), 175–185.

Kazaz N M E, 2019, Veri Madenciliğinde Kümeleme Analizi Yöntemlerinin İncelenmesi Ve Sağlik Bilimleri Alanindaki Uygulamalari, İstanbul Ünversitesi, Sağlık Bilimleri Enstitüsü, Yüksek Lisans Tezi, 59s, İstanbul.

Keskin M, 2018, Büyük Veride Makine Öğrenmesi Uygulaması, Yıldız Teknik Ünversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 84s, İstanbul.

Kocatürk A, Turfan D, Altunkaynak B, Gen Açıklama Verilerinde Özellik Seçimi için Kullanılan Filtreleme Yöntemleri, 2019 IMCOFE, 2019, 24–26 April, Antalya, Turkey, 68–75.

Korkmaz M, 2020, Python Programlama Dili ile Termoelastik Gerilme Analizi, Gazi Ünversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 196s, Ankara.

Kokkinos P, Varvarigou T A, Kretsis A, Soumplis P, Varvarigos E A, 2015, SuMo:Analysis and Optimization of Amazon EC2 Instances, J Grid Computing, 13, 255–274.

Kong W, Wu Q, Li L, Qiao F, Intelligent Data Analysis and its Challenges in Big Data Environment, 2014 IEEE International Conference on System Science and Engineering (ICSSE), 2014, 11–13, July, Shanghai, 108–113.

Kumar R, 2015, Future For Scientific Computing Using Python, International Journal of Engineering Technologies and Management Research, 2(1), 30–41.

Kumar V, Chhabra J K, Kumar D, 2014, Performance Evaluation of Distance Metrics in the Clustering Algorithms, InfoComp, 13(1), 38–51.

Kumar C, Toshniwal D, 2016, Analysis Of Hourly Road Accident Counts Using Hierarchical Clustering and Cophenetic Correlation Coefficient (CPCC), Journal Big Data, 3(13), 2–11.

Kunal K, 2016, Analysing Cascading over MapReduce, Research Journal of Computerand Information Technology Sciences, 4(9), 1–4.

Li W, Niu D, Liu Y, Liu S, Li B, 2019, Wide-Area SparkStreaming: Automated Routing and Batch Sizing, in IEEE Transactions on Paralleland Distributed Systems, 30(6), 1434–1448.

Liao S H, Tasi Y S, 2019, Big data Analysis on the Business Process and Management Forthestore Layout and Bundling Sales, Business Process Management Journal, 25(7), 1783–1801.

Malewicz G, Austern M H, Bik A J C, Dehnert J C, Horn I, Leiser N, Czajkowski G, Pregel: A System For Large-Scale Graph Processing, SIGMOD '10: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, 2010, 6–10, June, New York, 135–46.

Mcquitty L L, 1966, Similarity Analysis by Reciprocal Pairs For Discrete And Continuous Data, Educationaland Psychological Measurement, 26, 825–831.

Meng X, Bradley J, Yavuz B, Sparks E, Venkataraman S, Liu D, Freeman J, Tsa D B, Amde M, Owen S, Xin D, Xin R, Franklin M J, Zadeh R, Zaharia M, Talwalkar A, 2016, Machine learning in Apache Spark. Journal of Maching Learning Research, 17(34), 1–7.

Murtagh F, Contreras P, 2017, Algorithms For Hierarchical Clustering: An Overview II, Wıres Data Miningand Knowledge Discovery, 7(6), 1–16.

Narayanan V, 2014, Using Big-Data Analytics to Manage Data Deluge and Unlock Realtime Business Insights, Journal of Equipment Lease Financing, 32, 1–7.

Narula S, Jain A, Prachi, Cloud Computing Security: Amazon Web Service, 2015, Fifth International Conference on Advanced Computing & Communication Technologies, 21-21, Haryana, 501–505.

Onay A, 2020, Büyük Veri Çağında İç Denetimin Dönüşümü, Muhasebe Bilim Dünyası Dergisi, 22(1), 127–163.

Orçanlı K, 2019, Kalite Kontrol Grafiklerinde R Programlama Dilinin Kullanımı ile İlgili İçerik Analizi, OPUS Uluslararası Toplum Araştırmaları Dergisi, 13(19), 1390–1429.

Özdemir İ, Sağıroğlu Ş, 2018, Denetimlerde Büyük Veri Kullanımı ve Üzerine Bir Değerlendirme, Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım Ve Teknoloji, 6(2), 470–480.

Ponde P, Shirwaikar S, Gore S, 2016, Hierarchical Cluster Analysis on Security Design Patterns, Associationfor Computing Machinery, 92, 1–6.

Racine S J, 2012, Rstudio: A Platform-Independent ide For R and Sweave, Journal of Applied Econometrics, 27, 167–172.

Rençber S , Özdemir A, 2019, Almanya ve Türkiye'deki Büyük Veri Uzmanlarının Eğitim ve Yeteneklerinin Karşılaştırılması: Linked-in Veri Madenciliği Uygulaması, Veri Bilimi Dergisi, 2(1), 35–43.

Rocklin M, 2015, Dask: Parallel Computation with Blocked Algorithms and Task Scheduling, 14th Python in Science Conference, 126–132.

Rong M, Gong D, Gao X, 2019, Feature Selection and its Use in Big Data: Challenges, Methods, and Trends, in IEEE Access, 7, 19709–19725.

Sakarya B, 2007, From Delphi To Scenario by Using Cluster Analysis: Turkish Foresight Case, Middle East Technical University, Graduate School of Social Science, Master Thesis, 50s, Ankara.

Salloum S, Dautov R, Chen X, Peng P X, Huang J Z, 2016, Big data analytics on Apache Spark. International Journal of Data Science and Analytics, 1(3), 145–64.

Saraçlı S, Dogan N, Dogan I, 2013, Comparison of Hierarchical Cluster Analysis Methodsby Cophenetic Correlation. JournalI Nequalities and Applications, 203, 1–8.

Schaeffer D M, Olson P C, 2014, Big Data Options For Small and Medium Enterprises: Review of Business Information Systems, Clute Institute, 41–46.

Shi-Nash A, Hardoon D R, 2016, Data Analytics and Predictive Analytics in the Era Of Big Data. in: Geng H, Editor. Internet of Things and Data Analytics Handbook, Wiley, 800p, Hoboken.

Silva A R, Dias C T S, 2013, A Cophenetic Correlation Coefficient For Tocher'smethod. Pesquisa Agropecuária Brasileira, 48(6), 589–596.

Severance C, 2015, Guido Van Rossum: the Early Years of Python, IEEE in Computer, 48(2), 7–9.

Sönmez O, Zengin K, 2019, Yiyecek ve İçecek İşletmelerinde Talep Tahmini: Yapay Sinir Ağları ve Regresyon Yöntemleriyle Bir Karşılaştırma, European Journal of Science and Technology, Special Issue, 302–308.

Takcı H, Aydemir N, 2018, Büyük Veri Yaklaşımıyla Birden Çok Bilgi Erişim Merkezinin Kolektif Kullanımı, Bilişim Teknolojileri Dergisi, 11(2), 123–129.

Taylor R C, 2010, An Overview of the Hadoop/Mapreduce/Hbase Framework and its Current Applications in Bioinformatics, BMC Bioinformatics, 11, 2–6.

Ulaş M, Karabay B, 2020, Terör Saldırılarını İçeren Büyük Verinin Makine Öğrenmesi Teknikleri ile Analizi, Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 32(1), 267–277.

Warren J D, Moffit J R K, Byrnes P, 2015, How Big Data Will Change Accounting, Accounting Horizons, 29(2), 397–407.

Xu R, Wunsch D, 2005, Survey of Clustering Algorithm’s, IEEE Transactions on Neural Networks, 16(3), 120–127.

Yavuz G, Aytekin S, Akçay M, 2012, Apache Hadoop ve Dağıtık Sistemler Üzerindeki Rolü, Dumlupınar Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 27, 43–54.

Yılmaz Ş, Patır S, 2011, Kümeleme Analizi ve Pazarlamada Kullanımı, Akademik Yaklaşımlar Dergisi , 2 (1), 91–113.

Yöntem M K, Adem K, 2019, Otomatik Düşüncelere Makine Öğrenme Yöntemlerinin Uygulanması ile Aleksitimi Düzeyinin Tahmini, Psikiyatride Güncel Yaklaşımlar, 11, 64–78.

Ziviani A, Fdida S, Ezende J F, Duarte M B, 2004, Toward a Measurement Based Geographic Location Service, Lecture Notes in Computer Science, 47, 43-52.

İnternet Kaynakları

1- http://databricks.com/spark/about/20160102/, 10.05.2017

2- http://tutorialspoint.com/apache_spark/ 10.05.2017

3- http://kaggle.com/usdot/flight-delays - airports.csv/, 16.05.2020

ÖZGEÇMİŞ

Adı Soyadı : Murat AKŞİT

Doğum Yeri ve Tarihi : Aydın / 04.06.1992

Yabancı Dili : İngilizce

İletişim (Telefon/e-posta) : 5397780446/murat@bigcatresearch.com

Eğitim Durumu (Kurum ve Yıl)

Lise : Muhsin Kalkan Anadolu Ticaret Lisesi (2006–2010)

Lisans : Afyon Kocatepe Üniversitesi, İstatistik Böl., (2011– 2015)

Yüksek Lisans : Afyon Kocatepe Üniversitesi, Fen Bilimleri Ens., İstatistik ABD, (2017– Devam Ediyor)

Çalıştığı Kurum/Kurumlar ve Yıl

: Big Cat Research(2017– Devam Ediyor)