T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI
İŞLETME YÖNETİMİ YÜKSEK LİSANS PROGRAMI

YÜKSEK LİSANS TEZİ

VERİ MADENCİLİĞİ VE HAVACILIK SEKTÖRÜNDE BİR UYGULAMA

EYYÜP BURAK LEVENT
13713017

TEZ DANIŞMANI
YRD. DOÇ. DR. AYŞE DEMİRHAN

İSTANBUL
2016

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI
İŞLETME YÖNETİMİ YÜKSEK LİSANS PROGRAMI

YÜKSEK LİSANS TEZİ

VERİ MADENCİLİĞİ VE HAVACILIK SEKTÖRÜNDE BİR UYGULAMA

EYYÜP BURAK LEVENT
13713017

Tezin Enstitüye Verildiği Tarih : 16.06.2016
Tezin Savunulduğu Tarih : 18.07.2016

Tez Oy birliği / Oy çokluğu ile başarılı bulunmuştur.

Unvan Ad Soyad	İmza
Tez Danışmanı : Yrd. Doç. Dr. Ayşe DEMİRHAN	
Jüri Üyeleri : Prof. Dr. Murat KARAGÖZ	
Yrd. Doç. Dr. Leyla İşbilen YÜCEL	

İSTANBUL
TEMMUZ 2016

ÖZ

VERİ MADENCİLİĞİ VE HAVACILIK SEKTORUNDE BİR UYGULAMA

Eyyüp Burak Levent

Temmuz, 2016

Dijitalleşen dünyada artık her hareketimiz, bilgisayar sistemlerinde bir iz bırakmaktadır. Şirketlerin veritabanları, şirket ile ilgili tüm işlemleri kayıt altına almaktadır. Geriye dönük bilgi erişimi sağlayan bu sistem, tüm bilgileri kayıt etmesiyle birlikte veri büyüklüğü sorununu da beraberinde getirmiştir.

Veri madenciliği işte bu büyük veri yığınından, şirkete kar sağlayacak, anlamlı veri çıkarma analizlerinde devreye girmektedir. Şirket yöneticileri kritik durumlarda karar alırken bazı önemli bilgilere ihtiyaç duymaktadırlar. Veri madenciliği yöntemleri ile aslında büyük verinin içinde gizli olan bu önemli bilgilerin çıkarılmasıyla karar destek sistemlerine zemin oluşturulmaktadır. Böylece yöneticilerin daha anlamlı karar almasına yardımcı olunabilmektedir.

Bu çalışmada veri tabanı ile veri ambarı kavramları açıklanmıştır. Veri madenciliği ile ilgili olan alt kavramların üzerinde durulduktan sonra, veri madenciliği yöntemleri detaylandırılmıştır.

Uygulama aşamasında ise havacılık sektöründe karşılaşılan fazla ikram problemi, veri madenciliği yöntemlerinden olan kümeleme analizi ile incelenerek analiz edilmiştir. Fazla ikram problemi, uçağa binecek yolcu sayısından daha fazla ikram yüklenmesi durumdur. Kümeleme Analizi Yöntemlerinden K-Ortalamalar (K-Means) Algoritması ile bilgiler analiz edilerek bu sorunun ortaya çıkarılmasına yönelik açıklayıcı ve önleyici öneriler sunulmuştur.

Anahtar Kelimeler: Veri Madenciliği, Veri Tabanı, Veri Ambarı, Kümeleme Analizi, K-Ortalamalar Algoritması

ABSTRACT

DATA MINING METHODS AND AN APPLICATION IN AVIATION SECTOR

Eyyüp Burak Levent

July, 2016

In the digitalizing world, every move we made has a footprint in the computer systems. Databases of companies record all data which related with company. It provides access to backward information. However, it costs a problem that makes harder to handle with big data.

Data mining provides profit to company when they need to retrieve valuable data from big raw data. Managers or executives of a company need some important information when they about to make decision in critical situations. Data mining methods provide to executives this valuable information for decision support systems. Hence managers make decision easier.

In this study, database, data warehouse concepts was explained. After explanation of data mining subtitle, data mining methods were detailed.

As an application, in aviation sector, over catering problem was researched and analyzed with cluster analysis method. Over catering problem happens when it was load to aircraft treat more than flight list. With using K-Means Algorithm, data was analyzed and it was given suggestions to prevent this problem.

Keywords: Data Mining, Database, Data Warehouse, Cluster Analysis, K-Means Algorithm

ÖN SÖZ

Çalışmamın temel amacı, veri madenciliği ile temel kavramları açıklamak, veri madenciliği yöntemlerini detaylandırmaktır. Anlatılan bu yöntemlerin nasıl uygulandığını ise bir uygulama üstünde göstererek, veri madenciliği süreci, veri toplamadan, analiz ve sonuç kısmına kadar detaylı bir şekilde anlatılmıştır.

Yüksek lisans öğrenimim boyunca, desteklerini esirgemeyen değerli arkadaşlarıma, öğretim üyelerine, tez aşamasında, konu belirlenmesi, tez yazımı ve uygulamada tecrübeleriyle destekleyen danışman hocam Yrd. Doç. Dr. Ayşe Demirhan'a, tez sürecinde bana her türlü desteği sağlayan sevgili eşim Pınar Levent'e ve aileme teşekkürlerimi sunarım.

İstanbul; Temmuz, 2016 Eyyüp Burak Levent

İÇİNDEKİLER

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

KISALTMALAR

BIRCH : Balanced Iterative Reducing and Clustering using Hierarchies

CLIQUE : Clustering in Quest

CURE : Clustering using Representatives

DBA : Database Administrator

DBSCAN : Density-Based Spatial Clustering of Applications With Noise

DENCLUE : Density-Based Clustering

GB : Giga Byte

IATA : International Air Transport Association

ICAO : International Civil Aviation Organization

KNN : K-Nearest Neighbors

OLAP : Online Analytical Processing

OLTP : Online Transaction Processing

PAM : Partitioning Around Medoids

SPSS : Statistical Package for the Social Sciences

SQL : Structured Query Language

STING : Statistical Information Grid

TB : Tera Byte

1. GİRİŞ

Şirketlerin temel amaçlarından birisi, kârını maksimize edip, maliyetlerini ise düşürmektir. İşletme alanında birçok çalışma bu hedefe hizmet için bulunup literatüre eklenmiştir.

Teknolojinin hızlı gelişmesi, şirketlere süreçlerinin yönetmede birçok yönden kolaylık sağlamaktadır. Bu süreçte şirketler rahatça veri üretip, bunları saklamaktadır. Fakat veri üretim hızının artmasıyla birlikte, şirketler sahip oldukları verileri kullanmakta ve ondan anlamlı başka bilgiler çıkarmakta zorluk yaşamaya başlamıştır. Bu durumu aşmak için veri madenciliği yöntemleri geliştirilmeye başlanmıştır.

Şirket yöneticileri, şirketlerin temel hedefi olan kâr maksimizasyonu için belirli dönemlerde önemli kararlar almak zorundadırlar. Bu kararları alırken ellerinde destekleyici bilgilerin olması gerekmektedir. Sözkonusu bu önemli bilgiler veri madenciliği yöntemleri ile veri ambarlarında duran dağınık verilerden çıkarılarak yöneticilere raporlanmaktadır.

Çalışma, literatür araştırması ve uygulama kısımları olmak üzere iki ana başlık altında toplanabilir. Literatür taramasında, veri tabanı, veri ambarı kavramları açıklanmakta, veri madenciliği yöntemleri iste detaylı bir şekilde ele alınmaktadır.

Uygulama kısmında ise, havacılık sektöründe havayolu şirketlerine ekstra maliyet getiren fazla ikram problemi ele alınmakta, bir x havayolundan alınan veriler ışığında analizler gerçekleştirilerek, mevcut durum incelenmekte ve olası çözüm önerileri sunulmaktadır.

2. VERİTABANI

2.1. Veritabanı Kavramı

Veritabanı, aralarında anlamsal ilişkiler olan bilgilerin bir arada bulunduğu, nerede ve ne amaçla kullanılacağına uygun olarak düzenlenmiş verilerin, mantık çerçevesinde açıklamalarının olduğu veri topluluğu depolarıdır. Gerçek hayatta var olan ve veriler arasında mutlak bağlar bulunan ilişkileri modellemektedir[1].

Günümüz gelişen teknolojileri ile verilerin sistemler üzerinde kayıt altında bulundurulması daha kolay hale gelmiştir. Bununla beraber, verilerin sistemlere kayıtları esnasında oluşabilecek hatalara karşın ek önlemlere başvurulmuştur. Böylece gereksinim duyulan alanlarda ortaya çıktıkça veritabanı sistemleri geliştirilmiş ve ortaya çıkan veriyi yönetebilmek için yönetim sistemleri geliştirilmiştir[2].

Veritabanı sistemlerinin, diğer dosyalama yöntemlerine göre oldukça üstün yanları bulunmaktadır. Bu üstünlüklerden biri; geleneksel dosyalama süreçlerinde kullanılan yöntemlere karşılaştırıldığında veritabanı sisteminde verinin tekrarlanmasını önleyebilmesidir. Ayrıca, verinin tutarlı olmasını sağlaması ve aynı andaki erişimlerde tutarsızlıkların ortaya çıkmasını önlemekle birlikte veriye herkesin ulaşılmasını engellemesi ve bu yolla verinin güvenliğini sağlaması da diğer bir üstünlüğüdür. Muhasebe departmanında çalışan bir personelin diğer personelin özlük bilgilerine ulaşmasının engellenmesi buna örnek olarak verilebilir [3].

[1] Zehra Alakoç Burma, **Veri Tabanı Yönetim Sistemleri**, 2009, 12
[2] Yalçın Özkan, **Veri Madenciliği Yöntemleri**, 2008, 38
[3] age, 39

2.2. Veritabanı Modellerinin Gelişimi

Teknolojiyle beraber gelişen veri sistemlerinin, geçmişten günümüze kadar olan değişimlerini ve gelişmelerini Tablo 1'deki gibi sınıflandırmak mümkündür.

Tablo 1 : Veri Tabanı Modellerinin Gelişimi

KUŞAK	ZAMAN	VERİ MODELİ	ÖRNEKLERİ
İLK	1960-1970	Dosya Sistemi	VMS/VSAM
İKİNCİ	1970	Hiyerarşik Model ve Ağ Modeli	IMS/ADABAS/IDS-II
ÜÇÜNCÜ	1970lerden günümüze	İlişkisel Model	DB2/Oracle/MS SQL
DÖRDÜNCÜ	1980lerden günümüze	İlişki Veri Modeli İlişki Varlık Modeli	Oracle 11g/Versant/DB/DB2
GELECEK KUŞAK	Günümüzden geleceğe	XML Hibrit DBMS	DbXML/Tamino/ DB2 UDB /Oracle 11g/MS SQL Server

Kaynak: Carlos Coronel, Steven Morris, and Peter Rob, Database Systems: Design, Implementation and Management, 2012, 35

1960-1970 dönemini kapsayan İlk kuşak evresinde veri modeli olarak Dosya Sistemi kullanılırken, 1970'den sonraki dönemlerde ise Hiyerarşik Model ve İlişkisel Model modellerinin daha yaygın kullanıldığı görülmektedir.

2.3. Günümüzde Kullanılan Veri Modelleri

Veri modeli anlamsal olarak baktığımızda; bir veritabanı yapısının ana hatlarını ve zeminini oluşturmaktadır. Toplanan veriyi açıklanabilir düzeyde düzenleyebilmek için gerekli olan kavramlar ve kavramlarla beraber kullanılan yapılar ve işlemlerin tümüne veri modeli diyebiliriz. Günümüze kadar geçen zamanda birçok veri modeli oluşturulmuş ve geliştirilmiştir. Bu modeller Tablo 1'de gösterildiği gibi 4 grup altında birleştirilebilir:

- Hiyerarşik Model,

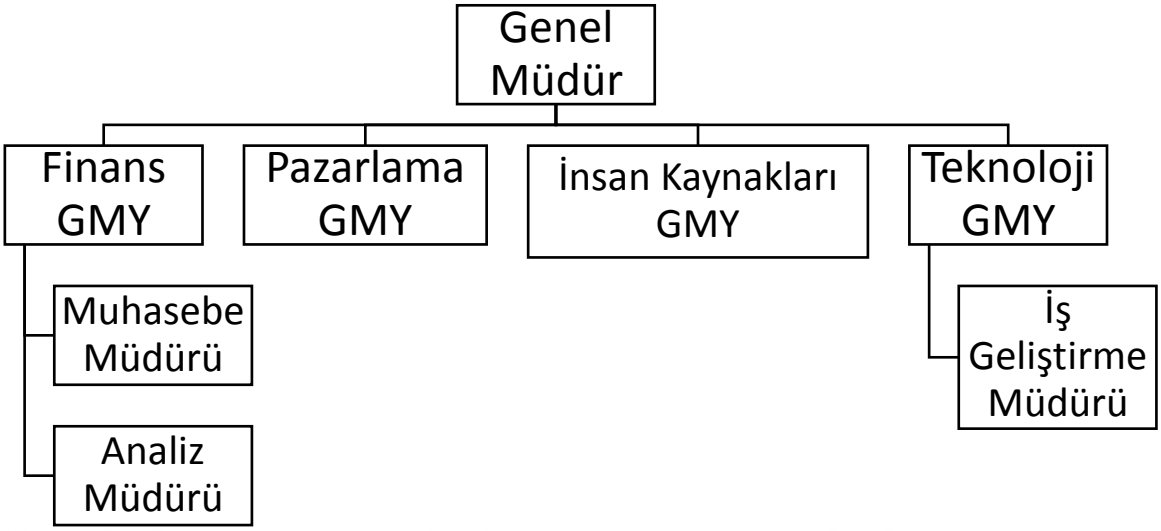
- Ağ Model,
- İlişkisel Model,
- Nesneye Yönelik Veri Modeli.

2.3.1. Hiyerarşik Model

Hiyerarşik veritabanı modeli kısaca 1960'lar ve 1970'lerde geliştirilmiş, dalları ve gövdesiyle sıklıkla bir ağaca benzetilen veritabanı modelidir. Bu veritabanı modelinde ilk göze çarpan temel özellik parent-child ilişkisine sahip olması ve bu özelliğiyle organizasyon şemasını andırmasıdır. Her ana gövde (parent) birden fazla dala(child) sahip olabilmektedir. Sistemde veriler arasındaki bazı ilişkisel hareketler oluşturulurken her dosyanın ana veri alanları belirlenerek bilginin transfer edileceği ana parent arasında ilişkilendirme yapılır. Bununla beraber child tablosuna eklenecek olan verinin eş zamanlı olarak parent tablosunda da aynı veriyi karşılayacak verilerin olması gerekmektedir. Böylece parent ve child dosyaları arasında ilişkilendirilme yapılması sağlanabilmektedir.

Hiyerarşik veri tabanı modelinin bazı kısıtları bulunmaktadır. Çoğu zaman dezavantaj olarak görülebilecek bir kısıtı, aranacak verinin ilk önce parent tablosundan başlayarak aranmasıdır. İstenilen verinin ilk önce ilgili parent dosyasının bulunup, daha sonra o child dosyasındaki verilerin bulunması gerekmektedir[4]. Hiyerarşik veritabanı modelini Şekil 1'deki gibi şekillendirebiliriz,

[4] Emel Seymen Turan, **Bir Telekomünikasyon Firmasında Müşteri Segmentasyonu,** 2010, 6

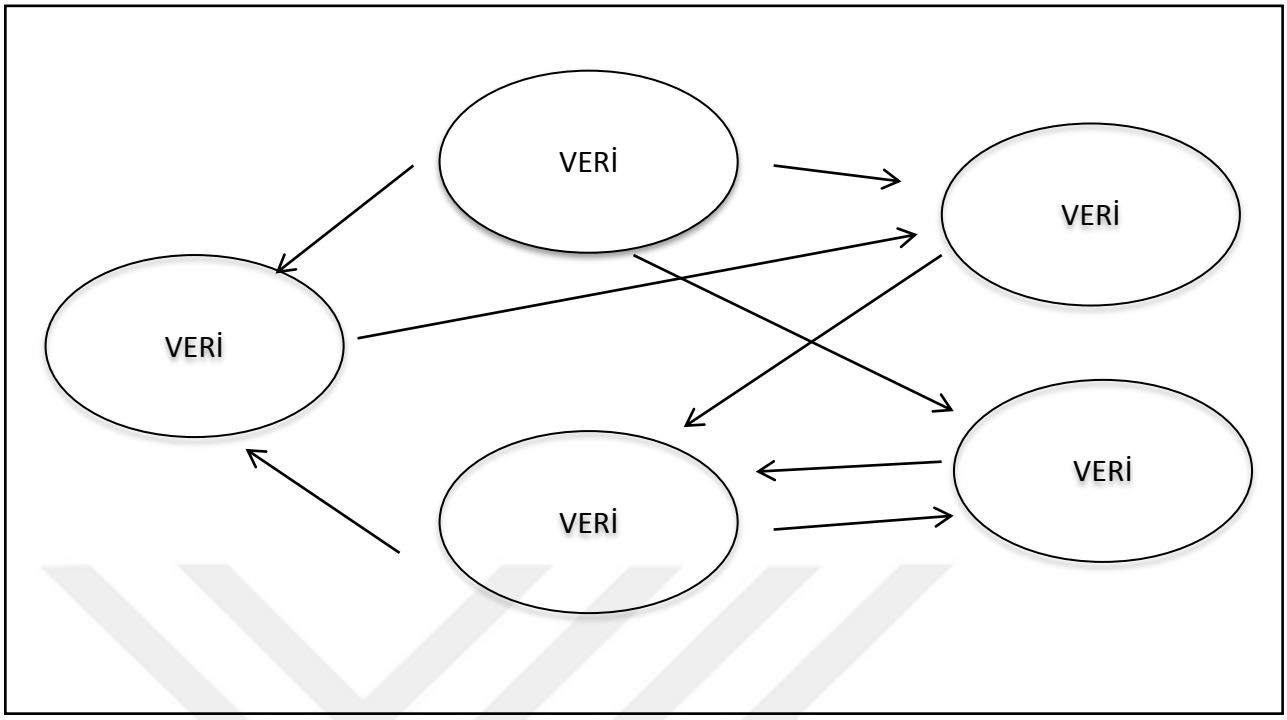


Şekil 1: Hiyerarşik Veritabanı Modeli Örneği

Şekil 1'de bir firmanın organizasyon şeması temsil edilmiştir. Şirket yöneticisi olarak Genel Müdür ve ona bağlı olarak çalışan Finans, Pazarlama, İnsan Kaynakları ve Teknoloji Genel Müdür Yardımcılığı olarak alt dallara ayrılmaktadır.

2.3.2. Ağ Tipi Modeli

Hiyerarşik veri tabanı modelinin ardından daha gelişmiş bir model olan ve genellikle 1970'lerde kullanılan ve diğer adıyla Şebeke Veritabanı Modeli olarak bilinen Ağ Tipi Model gelmektedir. Hiyerarşik veri tabanı modeliyle benzerlikleri bulunsa da en önemli ayırt edici özelliği bir dosyanın birden fazla parent ile ve birden fazla child ile ilişkili olacağı düşünülerek ortak bilgi alanları temelleri üzerine kurulmuş olmasıdır. Bir diğer ifade ile aracı dosyalarla çalışan hiyerarşik modeldeki bu ilişkiyi kaldırarak doğrudan doğruya dosyalar arasında bağlantı kurmayı sağlamıştır ve birbirleriyle direkt ilişkisi bulunan dosyalarda bu model ile çalışıldığı görülmüştür.



Şekil 2 : Ağ Tipi Veri Modeli

Şekil 2'de verilerin Ağ Modelinde nasıl birbirleri ile ilişkili olduğu görselleştirilmiştir.

2.3.3. İlişkisel Veri Modeli

Günümüze yaklaştıkça İlişkisel Veri Modeli hemen hemen her alanda gittikçe artan bir önemde kullanılmaya devam etmektedir. İlişkisel Veri Tabanı Modeli 1970 yılında Dr. Edgar F Codd. tarafından yazılan "A Relational Model of Data for Large Shared Data Banks" adlı makalede ortaya atılmıştır. Kullanım alanı oldukça yaygın olmasına karşın, daha çok ticari veritabanı yönetim sistemlerinde kullanılan bir model olarak karşımıza çıkmaktadır. İlişkisel modelin en temel dayanağı, nesneler arasındaki bağlantının, nesnelerin içerdiği değerlere göre ilişkilendirilmesidir. Varlıklar arasındaki çözülmesi zor ve bir o kadar karmaşık ilişkileri en aza indirmek ve daha basit hale getirebilmek amacıyla oluşturulmuştur ve geliştirilmesi de bu yönde ilerlemiştir[5].

İlişkisel Veri Modelinde veritabanı tablolar halinde bulunmaktadır. Bu tablolar kendine özgü isimlere sahip olup her tablo bir varlığa ya da bir ilişkiye karşılık gelmektedir. Tabloyu oluşturan sütunlar, nitelikleri açıklamakta ve tabloların

[5] Yalçın Özkan, **Veri Madenciliği Yöntemleri**, 2008, 17

oluşturulması esnasında tanımlanırken, satırlarda bahsedilen bu niteliklerin değerlerine karşılık gelmekte ve veri girişinde tanımlanmaktadır. Satırların her biri ‘’kayıt’’ olarak da nitelendirilebilir. [6]

Tablo 2’deki örnekte tabloyu oluşturan satır ve sütunlar gösterilmektedir.

Tablo 2: İlişkisel Veritabanı Modeli Tablosu

NO	**ADI**	**BÖLÜM NO**
1453	SERHAT AKAYDIN	7
1247	AYŞEGÜL UFAK	10
765	DENİZ YAR	8

İlişkisel Veri Tabanı Yönteminde satır(kayıt), sütun(özellik) ların yanısıra gözönünde bulundurulması gereken bir başka kavram da domain(etki alanı)dir. Domain, tablolarda var olan dataların özelliklerine göre alabilecekleri değerleri gösterir. Yukarıdaki Tablo 2 incelediğinde, ad sütunu için eğer metinsel bir kodlama yapılmış ise bu sütuna asla rakam girilmemesi gerekmektedir. Aynı zamanda “No” olarak belirtilmiş ve sayı kodlamasıyla verinin girileceği “No” sütununa metinsel ya da float bir rakam yazılmaması beklenir. Domain doğrudan doğruya tablolarla ilişkilidir[7].

2.3.3.1. İlişkisel Veritabanı Tablo Özellikleri

1. İlişkisel veritabanı tabloları satır ve sütunlardan oluşmaktadır.
2. Satırların sırasının bir önemi olmadığı gibi satırların kendi içlerinde yer değiştirmeleri de veritabanı özelliğini etki altına almaz.

[6] age, 17
[7] Ramez Elmasri, Shamkant B Navathe, **Fundementals of Database Systems**, Fourth Edition, 2003, 125

3. Sütun sırası da aynı satır sırasında olduğu gibi önemsizdir ve yer değiştirilmesi bir hataya sebebiyet vermez.
4. Sütunlara verilen isimler birbirlerinden ayrı olmak zorundadır.
5. Satırlardaki veriler birbirlerinden farklıdır.
6. Farklı isimler verilen sütunlar aynı domaindeki değeri alabilir[8].

2.3.3.2. Veritabanı Şeması

Veritabanı Şemasını, tablolar ve nitelikleri oluşturur. Veri tabanının anlamlı birleştirilmiş haline veritabanı şeması adı verilmektedir. Veritabanı Şemalarını aşağıdaki gibi 2 şekilde sınıflandırmak mümkündür.

1. Fiziksel Şema
2. Kavramsal Şema

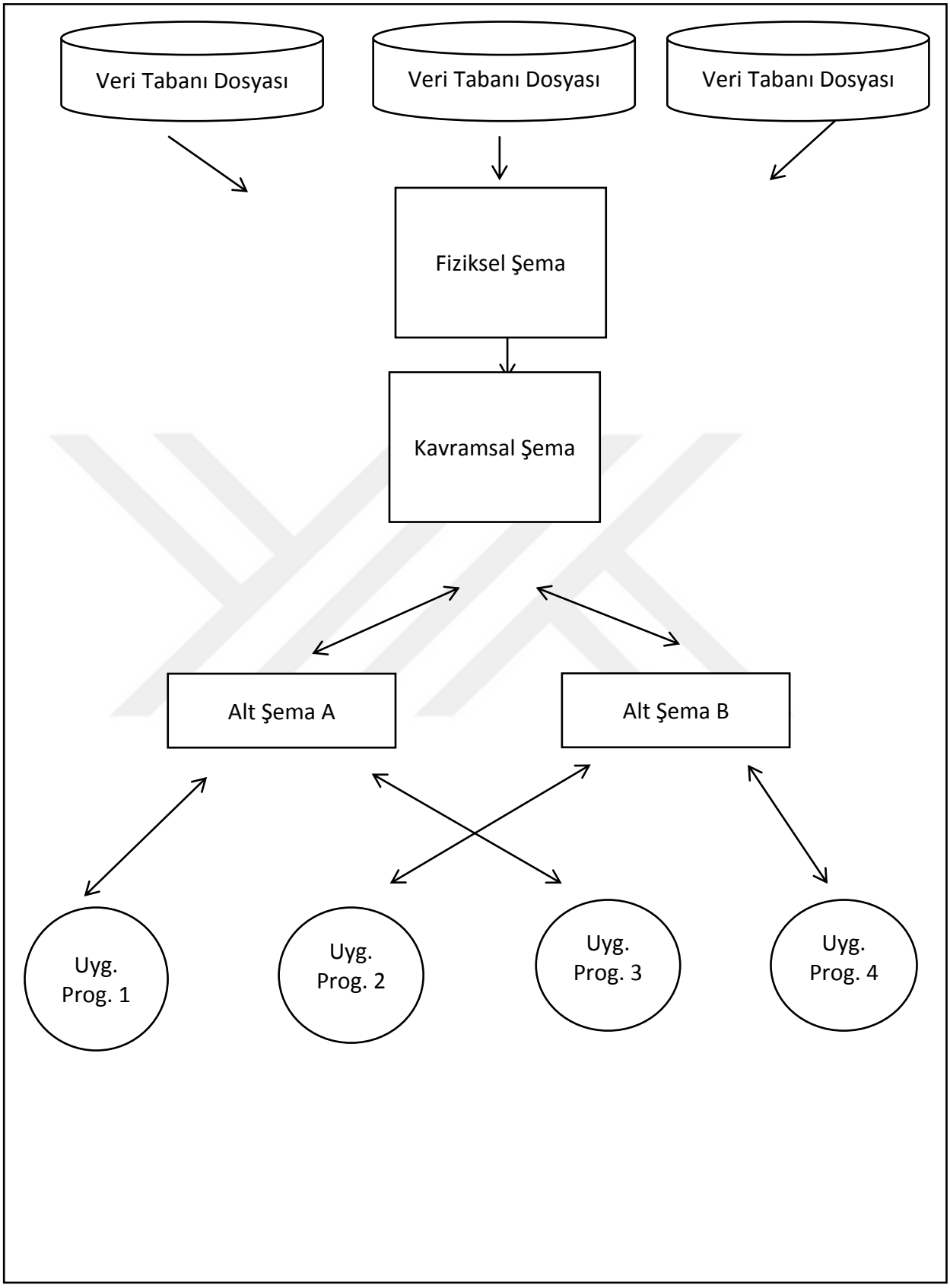
Fiziksel Şemayı bir örnek kapsamında değerlendirecek olursak, bilgisayarda bulunan bir disk dosyası veritabanını oluşturacak, bu dosyanın adres ve özellikleri ile ilgili ifadeler ise fiziksel şemayı oluşturacaktır.

Kavramsal Şema ise mantıksal tasarımdan oluşmaktadır. Her verinin kaydından sonra anlamsal ilişkilerin oluşturulması aşamasında kavramsal şema oluşturulmaktadır. Kavramsal şema içerisinde veri elemanları ve ilişkileri, veri alanları, kaydedilmiş dosyalar ve verilerin türleri yer almaktadır.

Kavramsal Şema oluşturulduktan sonra alt şemalar oluşturulmaya başlanır. Alt şemalara programlar tarafından alt düzey mantıksal görünümde ihtiyaç duyulmaktadır. Geçerli uygulama, veri tabanının tüm alanlarına erişim olanağına ihtiyaç duymadan sadece kendisi ile ilgili alt şemalara ulaşılmasını yeterli görmektedir[9].

Sekil 3'de ise veri tabanı dosyalarına, fiziksel ve kavramsal şemalar aracılığı ile Uygulama Programcılarının erişimi görselleştirilmiştir.

[8] Edgar Frank Codd, **The Relational Model for Database Management**, 2000, 11
[9] Yalçın Özkan, **Veri Madenciliği Yöntemleri**, 2008, 18



Şekil 3: Veritabanı Şeması

2.3.4. Nesneye Yönelik Veri Modeli

1980'li yıllardan itibaren, ilişkisel veri tabanı modelinin çok kullanıldığı yönetimsel ve işletimsel sistemler dışında diğer sistemler için ilişkisel modelin yetersiz kaldığı görülmüş ve Nesneye Yönelik Veritabanı Modeli geliştirilmiştir.

İlişkisel veri tabanının sistemsel yetersizliğiyle zorlu, karmaşık veriler ve veri tabanlarının büyüklüğü üzerine gidilmiş ve sonuç olarak bilgi sistemlerinin yeniden tasarlanması ile sonuçlanmıştır. Yapılan araştırmaların sonuçları ise Nesneye yönelik veri modeline öncülük etmiştir.

Nesneye Yönelik Veritabanı Modeli, ilişkisel veri tabanı modelinden farklı olarak üç boyutlu yapılandırmayla oluşur. Verileri İki boyutlu tablolar haline getiren ilişkisel veri modelinin aksine nesne modeliyle veriler tek boyutlu hale gelmektedir. Tek parça halinde gelen veriler birden fazla verinin sonuçlanması beklenildiğinde yetersiz kalmaktadır.

Nesneye Yönelik Veritabanı Modelinin Özellikleri;

- Bir nesne birçok nesneden oluşabildiği gibi birden çok nesneyle ilişki kurup gönderme yapabilmektedir.
- Nesnelerin her birinin yapılarına ait ve davranışlarının bilgileri kayıtlıdır.
- Nesneler arasındaki iletişimi ileti ile sağlayabilmektedir. İleti aktarma ile nesneler arasındaki iletişim gerçekleştirilebilmektedir.
- Nesneler eğer benzer davranışlara sahip ya da benzer yapılara sahip ise gruplandırılarak sınıflar haline getirilebilmektedir. Sınıf ise bu şekilde nesnelerden oluşmaktadır.
- Benzer davranışlardaki nesnelerden oluşan sınıfların altında ise bir ya da birden fazla alt sınıflarda olabilmektedir[10].

[10] Ferhat Çakır, **Analysis of Some Data Relating to Small and Medium-Sized Enterprises Using Data Mining Methods**, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2012, 12

3. VERİ AMBARI

Veri ambarı, zaman boyutunda gelen verilerin birikimiyle oluşan, işletmenin daha sonra kullanabileceği bir yığındır.

İşletmelerde bilgi sistemleri sadece o departmandaki çalışanlara hizmet vermeyi değil, kurumun yöneticilerine de raporlama yaparak, stratejik karar alma ve performans değerlendirmelerini kolaylaştırmayı amaçlamaktadır. Veritabanı sistemlerinin avantajlarına karşın, bilginin çekilmesi ve karar destek sistemlerine anlamlı bilgi sunma konusunda zorlandığı gözlenmiştir. Yaşanan bu zorluktan dolayı, veriyi farklı bir şekilde saklama ve daha hızlı erişimi sağlayacak sistemler üzerinde çalışılması bir gereklilik haline gelmiştir. Bununla birlikte, gün geçtikçe islenmesi gereken bilgi boyutunun önemli ölçüde artması, bu denli büyük verileri islemek için geleneksel veritabanı yöntemlerinin yetersiz kalmasına yol açmıştır. Veri ambarlarının işte, bu negatif durumları ortadan kaldırmak için ortaya çıktığı görülmüştür.[11]

3.1. Veri Ambarı Kavramı

Veri ambarcılığı, üst seviye yöneticiler için, sistematik olarak düzenlenmiş, anlamlı ve karar almada yardımcı bir mimari sunar. Veri ambarı sistemlerinin günümüzün rekabetçi ve hızlı gelişen dünyasında gittikçe değerlenen bir araç olduğu görülmektedir. Rekabetçi ortamda, veri ambarcılığı mutlaka sahip olunması gereken bir pazarlama silahı olarak görülmekte ve bu nedenle son yıllarda şirketlerin bu sistemleri kurmak için büyük bütçeler ayırmaya başladıkları bilinmektedir.

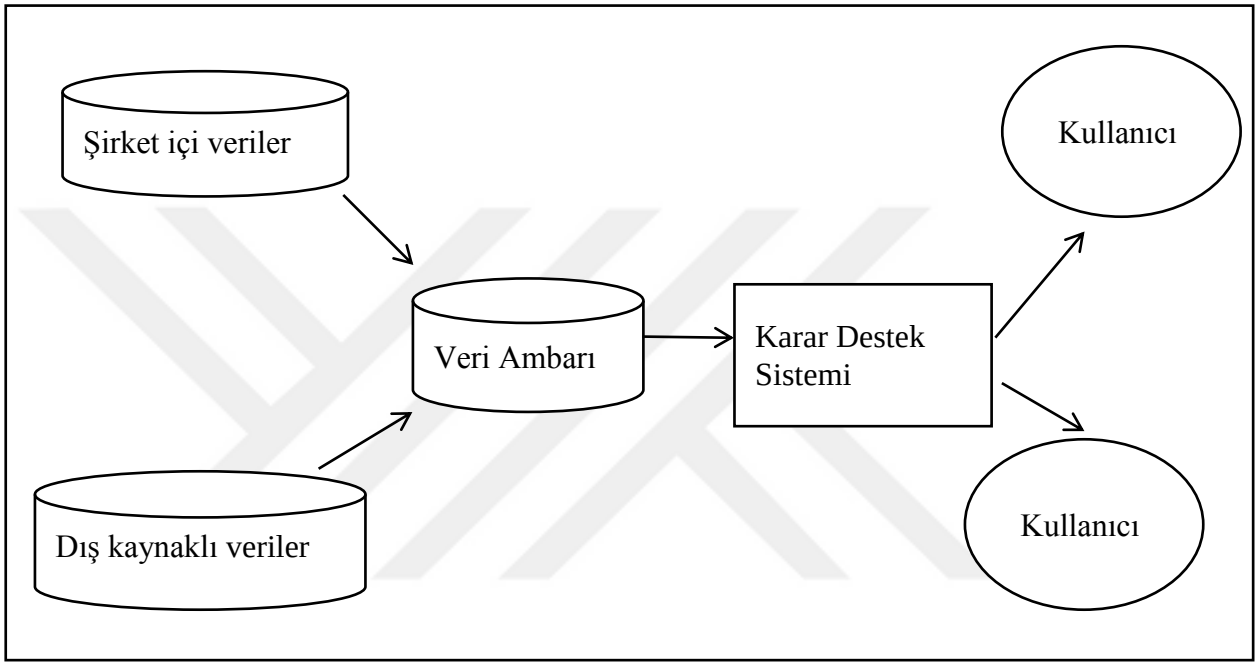
Veri ambarını en basit tanımıyla, karar destek uygulamaları için hazırlanan bir araçtır. Diğer bir deyişle, veri ambarları, karar destek sistemlerine teknik bir altyapı sağlamaktadır.

Veri ambarı, işletmenin veri tabanlarında girmiş olan, yeni, eski tüm veriyi, karar destek amacıyla saklar ve kullanılmasını sağlar. Bu sayede, veritabanında olan ama kullanılamayan veri de artık yararlı bir hale gelmektedir. Bu durum, verisi büyük

[11]Yalçın Özkan, **Veri Madenciliği Yöntemleri**, 2008, 9

olan ve var olan müşteri ihtiyaçlarının eğilimini tahmin etmek isteyen; bankalar, sigorta şirketleri, perakende satış yapan firmalar için çok önemlidir. Günlük işlemler, girdiler, çıktılar, satışlar vb. işlemler an ve an ilgili veri tabanında kayıt altına alınabilmektedir.

Toplamda birbiri ile ilişkili olmayan bu verinin, bütünleştirilip anlamlandırılmasında veri ambarı devreye girmektedir [12].



Şekil 4 : Veri Ambarı, Karar Destek Sistemleri ve Kullanıcılar Arasındaki İlişkiler

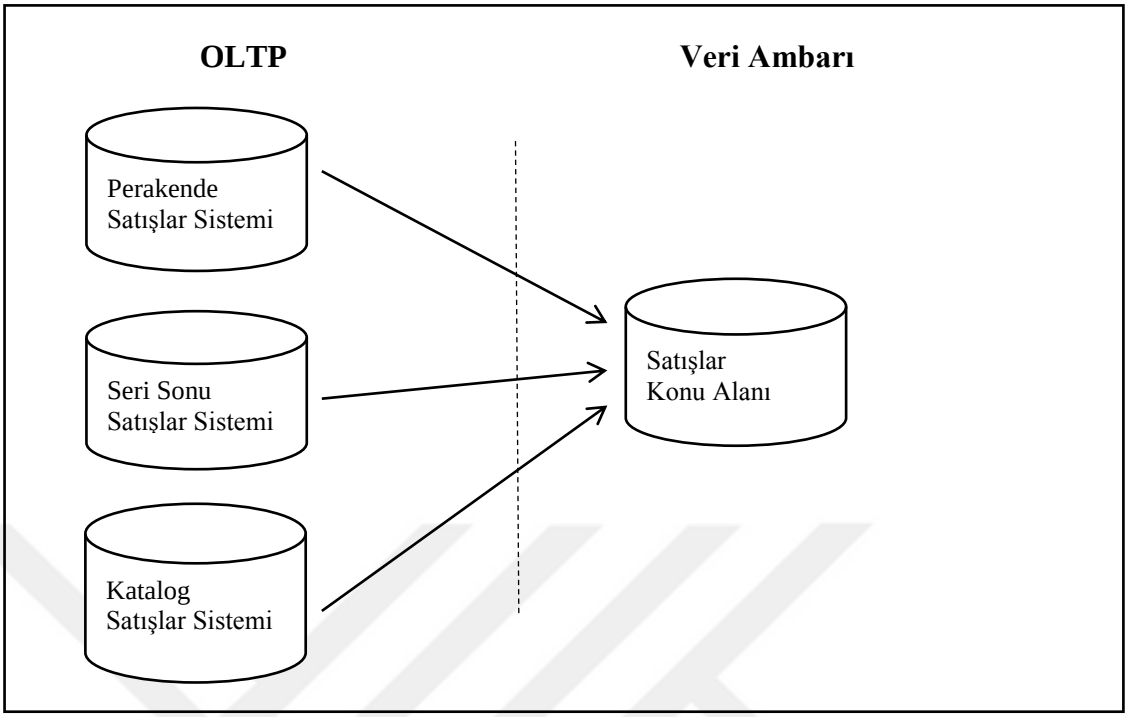
Sekil 4'de şirket içi ve dışı verilerin, veri ambarında depolandıktan sonra karar destek sistemlerine kullanılabilir bilgi olarak yollanarak, şirket yöneticileri tarafından kullanılabilir hale getirilmesi görselleştirilmiştir.

3.2. Veri Ambarının Özellikleri

Veri ambarı, karar destek süreçlerinde, yöneticilere destek vermek üzere, bir zaman boyutu içinde, konuya yönelik, silinmeyen ve bütünleşik olarak verinin depolanmasıyla oluşmaktadır. Bu özellikler aşağıda ayrıntılı olarak açıklanmaktadır.

[12] Jiawei Han, Micheline Kamber, Jian Pei, **Data Mining Concepts and Techniques**, 2012, 126

3.2.1. Verinin Konuya Yönelik Olması (Subject-Oriented)



Şekil 5: Veri Ambarının Konuya Yönelik Olma Özelliği

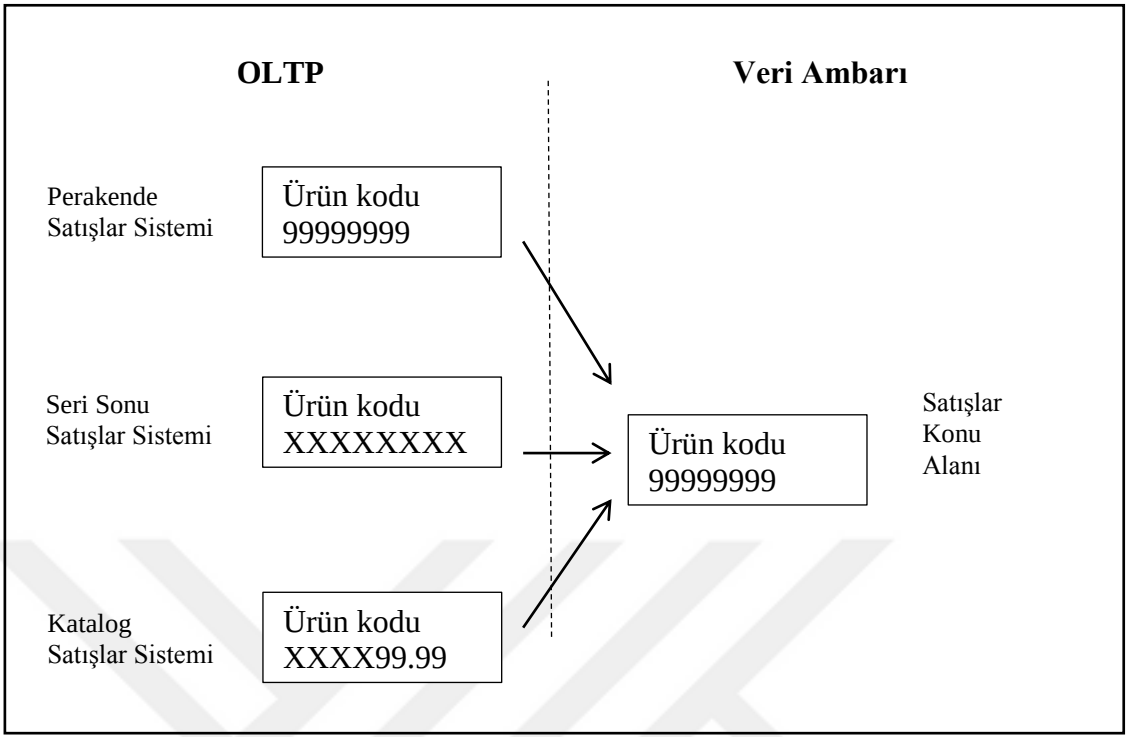
Veri Ambarının en önemli özelliği, işletmenin amaç ve konularına yönelik olmasıdır. Konuya yönelik olma, işletmedeki yüksek seviyeli varlıklara odaklanması anlamına gelmektedir. Bu varlıklar, bir üniversite için öğrenciler, dersler, öğretim görevlileri vb. olabilir.

Klasik işlemsel sistemler, daha çok, işletmedeki süreçlere odaklanmışken, veri ambarı işletmedeki konulara yoğunlaşmıştır. Örnek olarak, işlemsel sistemler, finans, muhasebe, insan kaynakları vb. sistemlere yöneliktir. Buna karşın, veri ambarı, müşteri, satıcı, ürünler gibi konulara yönelik tasarlanmaktadır. Veri ambarındaki verinin tasarımı ilgili konulara göre belirlenmektedir.

İşlemsel sistemlerin süreç bazlı, veri ambarının ise konuya yönelik olması, kişilere verilerin ayrıntıları ile ilgili bilgiler vermektedir. Veri ambarı, karar destek sistemlerinde kullanılamayacak hiçbir veriyi içermez. Bununla birlikte, işlemsel sistemlerin süreç temelli veritabanları işletmenin fonksiyonel bazda ihtiyaç duyabileceği verinin tümünü hemen sağlamalıdır.[13]

[13] age, 127

3.2.2. Verinin Bütünleşik Olması (Integrated)



Şekil 6: Verinin Bütünleştirme Özelliği

Aynı bilgi farklı sistemlerde farklı şekilde kodlanmış olabilir ancak farklı biçimde kodlanmış alanlar ortak kodlama biçimine dönüştürülür.

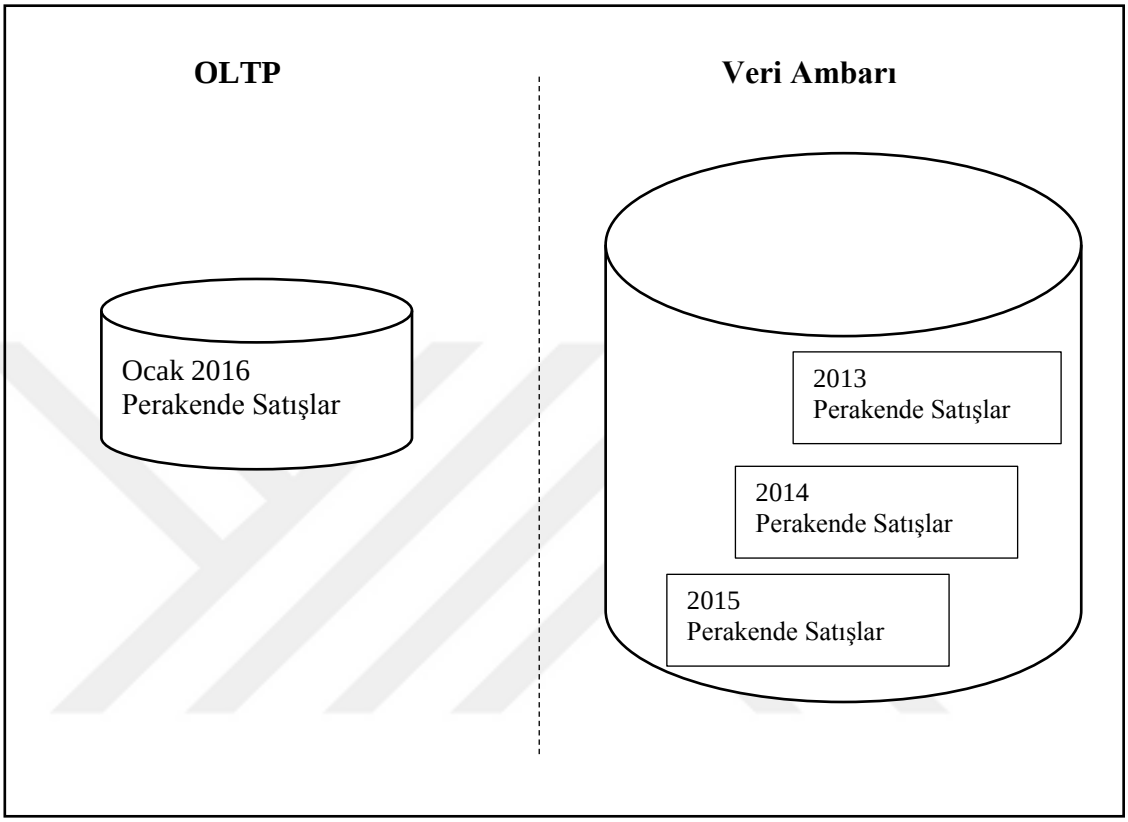
Veri ambarlarındaki verinin en belirgin özelliklerinden birisi de, bütünleşik olmasıdır. Veri ambarı içindeki veri mutlaka bütünleşik olmalıdır. Bütünleşme farklı biçimlerde olabilir. Tasarım aşamasında, verinin kodlanmasında tüm katılımcılar ile ortak bir payda da olunması, ölçü birimlerindeki tutarlılık, sayısal değerlerin ara yüzdeki gösterimdeki anlamlılık, bütünleştirme kavramının temel değerleridir.

İşlemsel sistemler süreç bazlı olduğu için sadece o süreçte çalışan ve o süreçte bu veriyi kullanan çalışanlar için belirli bir anlam çerçevesi belirlenmektedir. Tasarım aşamasında bu kullanıcılar veriyi tanımlarlar. Bu aşamada isimlendirme farklılıkları, kodlama farklılığı, ara yüzdeki farklılıklar gibi birçok farklılık olabilmektedir.

Örneğin, bazı uygulamalarda ağırlık ölçüsü olarak gram, bazılarında kilogram, bazılarında ise pound kullanılmış olabilir. Bu tür verinin, veri ambarına taşınması sırasında, birimlerin ortak bir standartta birbirine dönüştürülmesi gerekir.

Veri bütünleştirmede en çok karşılaşılan bütünleştirme sorunu tarihlendirmenin biçim farklılığıdır. Çünkü veri tabanlarında, farklı tarih biçimleri ile karşılaşmak mümkündür[14].

3.2.3. Verinin Zamana Bağlı Olması (Time-Variant)



Şekil 7: Verinin Zamana Bağlı Olması

Veri tabanında hem o döneme ilişkin verilere hem de daha önceki dönemlere ait verilere de yer verilmektedir. Veri saklamadaki en önemli unsurlardan birisi de, verinin ne zamana ait olduğunun bilinmesidir. Çünkü zamansız tutulan veriler, belirli bir süre sonra anlamsızlaşamaya başlamaktadır. Veri ambarlarında da tüm veri belirli zaman bilgisi içermektedir. Veri ambarındaki verinin bu özelliği, işlemsel sistemlerdekinden farklılık göstermektedir. İşlemsel sistemlerde, verinin kayıt altına alınma zamanı önemlidir.

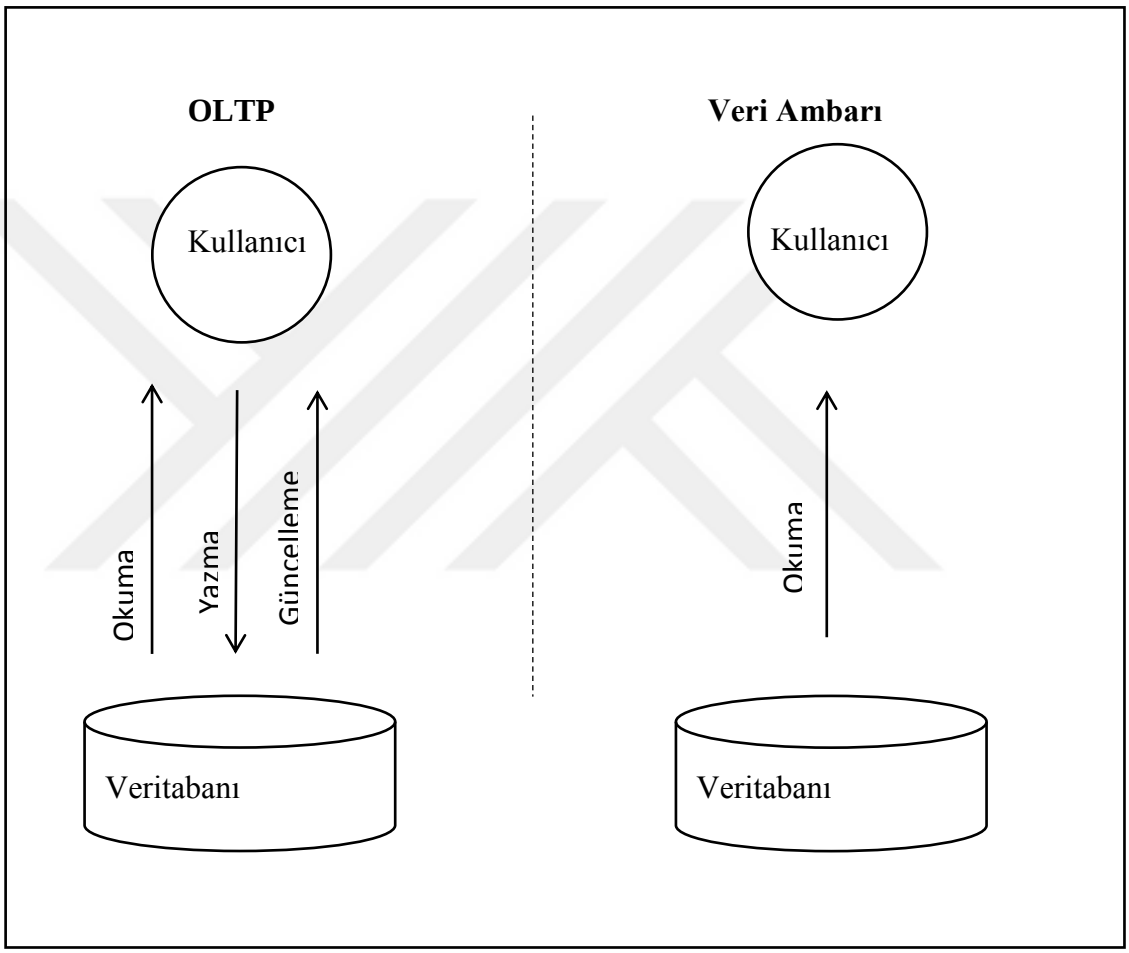
Veri ambarında ise, anlık veriler ile değil, geçmişteki aralıklı değerler ile ilgilenilmektedir. Veri zaman içinde aralıklandırılarak işleme alınmaktadır. Örneğin, mali yıl, ilk çeyrek, kapanış gibi aralıklar göz önüne alınır. Veri ambarında verinin

[14] age, 127

an az 5 yıllık değerleri tutulmalıdır.

İşlemsel sistemlerde, zaman boyutu olmaması sebebiyle, güncelleme ve silme işlemleri yapılabilmektedir. Veri ambarı ise, geçmiş dönemlere ilişkin tüm bilgileri içerdiği için, işlemsel sistemlerden değerler bir kez yansıtıldığında, bir sonraki güncellemeye kadar sabit kalmaktadır.[15]

3.2.4. Verinin Kalıcı Olması (Nonvolatile)



Şekil 8: Verinin Kalıcı Olması

Veri tabanına bilgi kaydedilebilir, okunabilir, güncellenebilir. Veri ambarının diğer bir özelliği ise, veri ambarındaki yazılmış verilerin kalıcı olması, silinemez bir yapıda bulunmasıdır. Veri ambarları, işletmenin yönetim gereksinimlerinde kullanılmak üzere tasarlandığı için, anlık işlemlerin yapılmasına uygun değildir.

İşlemsel sistemlerde, veritabanlarında yeni veri, kolaylıkla okunur, değiştirilir,

[15] Yalçın Özkan**, Veri Madenciliği Yöntemleri**, 2008, 27

silinir, güncellenebilir ve ekleme yapılabilir. Çünkü bu sistemler, işletmede günlük çalışan fonksiyonların devamlılığı için tasarlanmış olup günlük her türlü işlemin yapılmasına uygundur.

Veri ambarında ise iki tür işlem vardır, veriler yüklenir ve veriye erişilir. Bu yönüyle işlemsel sistemlere göre daha sade bir görünüme sahiptir.

4. VERİ TABANI İLE VERİ AMBARI ARASINDAKİ FARKLAR

Ticari veritabanı sistemlerinin bilinilirliği sayesinde, veri ambarının ne olduğunu bu iki sistemi karşılaştırarak anlamak oldukça kolaydır.

Canlı operasyonel veritabanı sistemlerinin en önemli görevi, canlı işlemlerin ve sorguların gerçekleştirilmesidir. Bu sistemlere canlı işlem işleme (Online Transaction Processing - OLTP) sistemi denir. Bu sistem, işletmenin günlük operasyonlarının gerçeklenmesinde kullanılır. Örneğin satın alma, üretim, bordrolama, işe alım ve muhasebe gibi. Buna karşın veri ambarı sistemleri, veri analizi ve karar almada rol oynayan üst düzey çalışanlara hizmet vermektedir. Bu sistemler, veriyi farklı ihtiyaçlar için çeşitli formatlarda organize eder ve sunarlar. Bu sistemler canlı analitik işleme (Online Analytical Processing - OLAP) sistemleri olarak bilinmektedir.[16]

4.1. OLTP ve OLAP Sistemleri Arasındaki Farklar

OLTP ve OLAP sistemleri arasındaki farklar 5 başlık altında toplanabilir. Tablo 3'de ise OLTP ve OLAP arasındaki farklar sıralanmaktadır.

Tablo 3: OLTP ve OLAP Sistemleri Arasındaki Farklar

Özellik	**OLTP**	**OLAP**
Karakteristik	Operasyonel işlemler	Bilgilendirici işlemler
Odak	İşlem	Analiz
Kullanıcı	Memur, Dba, veritabanı uzmanı	Müdür, yönetici, analist

[16] Jiawei Han, 128-129

Fonksiyon	Günlük operasyonlar	Karar destek için uzun dönemli bilgi
Veri tabanı tasarımı	ER, uygulama odaklı	Yıldız/kartanesi, nesne odaklı
Veri	Güncelliği garantili	Tarihsel, zamanla bakım yapılmalı
Özetleme	Detaylı	Öz, konsolide
Görüntüleme	Detaylı	Öz
Erişim	Okuma/Yazma	Okuma
Odak	Data girişi	Bilgi çıkışı
Erişilen kayıtlar	Onlar	Milyonlar
Kullanıcı sayısı	Binler	Yüzler
Veritabanı boyutu	GB	>TB
Öncelik	Yüksek performans, yüksek geçerlilik	Yüksek esneklik,

Kaynak: Jiawei Han, Micheline Kamber, Jian Pei, **Data Mining Concepts and Techniques**, 2012, 130

Tablo 3'e göre; OLTP operasyonel işlemlerde, OLAP'ın ise bilgilendirici işlemlerde öne çıktığı görülmektedir. OLTP veritabanı sistemlerinde, OLAP ise karar destek sistemlerinde kullanılır. OLTP ve OLAP sistemleri arasındaki farklar aşağıdaki gibi detaylandırılmıştır.

4.1.1. Kişiler ve Sistem Oryantasyonu

OLTP sistemi, müşteri odaklıdır ve memurlar, müşteriler ve bilgi işlemi çalışanları tarafından işlemler ve sorgular için kullanılmaktadır. Buna karşılık OLAP sistemi ise pazarlama odaklıdır ve üst düzey yöneticiler ve analistler tarafından veri analizi için

kullanılmaktadır.[17]

4.1.2. Veri İçeriği

Bir OLTP sistemi anlık detaylı veriyi karar almada kullanmak üzere yönetirken, OLAP sistemi ise yüksek miktardaki zaman boyutu olan veriyi, özetleme ve bir araya getirme için saklamakta ve yönetmektedir. Bu şekilde veri, karar almada daha kolay kullanılır hale gelmektedir.[18]

4.1.3. Veritabanı Tasarımı

OLTP sistemi genel olarak varlık bağıntı veri modeline ve uygulama odaklı veritabanı tasarımına uyum sağlar iken, OLAP sistemi ise yıldız veya kar tanesi modeline ve nesne odaklı veritabanı tasarımına uyum sağlamaktadır.[19]

4.1.4. Görüntüleme

OLTP sistemi temel olarak, şirket veya departmanlar içindeki, zamandan bağımsız olan anlık veriye odaklanmaktadır. Buna karşın, OLAP sistemi sıklıkla veritabanı şemalarının çoklu versiyonlarını kapsamaktadır. Ayrıca farklı organizasyonların başlattığı, birçok farklı veri depolarından entegre edilmiş verilerle de ilgilenmektedir. Yüksek veri miktarından dolayı, OLAP verisi çoklu depolama alanlarında depolanabilmektedir.[20]

4.1.5. Erişim Şekli

OLTP sistemlerine erişim genel olarak kısa, bölünmez(atomik) işlemlerle olmaktadır. Söz konusu sistemler eşzamanlı kontrol ve kurtarma mekanizmalarına sahip olmalıdır. Bununla birlikte, OLAP sistemlerine erişim genel olarak sadece okuma işlemleridir, fakat güncel bilgi yerine zamansal veri tutulduğu için, daha karışık sorgulardan oluşabilmektedir.[21]

[17] Micheline Kamber, 131
[18] age, 131
[19] Edgar Frank Codd, 29
[20] Yalçın Özkan, 28
[21] Micheline Kamber, 131

5. VERİ MADENCİLİĞİ

Teknolojinin gelişmesiyle beraber bilgilere ulaşabilme ve bilgileri kaydedebilme olanağı da oldukça artmıştır. Öyle ki günlük hayatımızda hiç farkında olmadan bilgilerimizin kayıt altında tutulduğunun çoğu zaman farkında olmayabiliriz. Yaptığımız alışverişlerdeki satın aldığımız ürünler, markalar, tercih ettiğimiz alışveriş merkezleri, çoğu zaman bu alışveriş merkezindeki mağazaların kameralarındaki görüntülerimiz veya gidilen hastanelerde verilen bilgiler günümüzde saklanıp bir bilgi birikimi elde edilebilmektedir. Bu bilgi birikimi karşımıza bir veritabanını çıkarmaktadır. Bilginin bu kadar değerli olduğu günümüzde, verinin bir madene dönüşmesi işlemi ise yukarıda anlatıldığı gibi oluşmakta ve bu bilgiler anlamlı bir hale gelip o madenden çıkmayı bekleyen birer cevher durumuna gelmektedir.

Veri madenciliği, önceden bilinmeyen anlamlı ve geçerli bilgilere veritabanları aracılığı ile ulaşılması ve işletmelerin hayatta kalabilmeleri için bu bilgilerin kullanılması yöntemidir.[22]

İşletmeler en çok hayatta kalmak ve sürekliliklerini devam ettirmek için ve aynı zamanda bunları yaparken doğru kararlar alabilmek için veri madenciliğini kullanmaktadırlar. Veri madenciliğinin işletmelere sağladığı en önemli özellik ise var olan fakat farkında olunmayan bilgilerin veri ambarından çıkarabilmesidir. Bu özelliği ile özellikle büyük ölçekli işletmelerde üretilen ürünün satış aşamasından sonra analizini yaparak ileride oluşturabileceği kampanyaları hazırlayıp ürünler arasındaki bağlantıları kurabilmektedir. Burada önemli olan işletmenin farkında olmadığı verileri veri madenciliği ile değerli hale getirip işletme için karlılığı artıracak yarar sağlamasıdır. Doğru karar almak için bilgiye dayalı bir sisteme ihtiyaç duyulduğu gibi bu noktada veri madenciliğine de özellikle önem verilmektedir.[23]

[22] Yalçın Özkan, 38
[23] Gökhan Silahtaroğlu, 10

5.1. Veri Madenciliği Tarihi

Hesaplama için kullanılmak amacıyla kullanılan ilk bilgisayarlar 1950'li yıllarda ortaya çıkmıştır. İlerleyen yıllarda ise yaklaşık 1960'larda verilerin saklanması kavramı ortaya çıkmış ve bilgisayarlar bu işlem için kullanılmaya başlanmıştır.1960'ların sonlarına yaklaşıldığında bilim adamları tarafından basit öğrenmeli bilgisayarlar geliştirilmeye başlanmış ve Minksy ve Papert perseptronların basit kuralları öğrenebileceğini kanıtlamışlardır. Bu basit kurallara dayanan sistemler geliştirilip, bu sistemleri uzmanlaştırmışlar ve sonuçta makinelerin öğrenimlerini sağlamışlardır.1970'lerde Veri Tabanı Yönetim Sistemleri'nin(VTYS) kullanılmasıyla beraber 1980'lerde VTYS'nin birçok alanda kullanılmaya ve yaygınlaşmaya başladığı görülmüştür. Bu veritabanlarına SQL veri tabanı sorgulama dilleriyle ulaşılabilmiş ve bu veritabanlarına çok büyük veriler kaydedilmiştir. Verilerin büyümesiyle beraber artık bilgi yoğunluğu arasından en yararlı bilginin nasıl bulunabilineceği aranmaya başlanmış ve bunun üzerinde araştırmalar yoğunlaşmıştır. 1992 yılına gelindiğinde ise artık veri madenciliğine ilişkin ilk yazılım ortaya çıkmıştır. İlerleyen teknoloji ve biriken veri kalabalıklığı ile veri madenciliğine olan ilgi ve ihtiyaç artarak devam etmektedir.[24]

5.2. Veri Madenciliği Kavramı

Verilerin çeşitli yöntemlerle toplanması sonucunda oluşan yığına yapılacak olan sorgulamalar ve bu sorgulardan çıkarılacak analizler veri madenciliği olarak adlandırılmaktadır.

Bir ev dekorasyon marketinde hangi ürünlerin çok sattığını ya da şubelerin satış ciroları karşılaştırıldığında hangisinin daha çok müşteriyi elde tuttuğunun belirlenmesi veri madenciliğinin bir uygulaması olarak görülemez. Bununla beraber yapılan araştırmalar sonucunda regresyon analizi yapılarak bulunmaya çalışılan yaş ile gelir arasındaki ilişkiyi belirlemek veri madenciliği olarak adlandırılabilir.

Veri madenciliği internette gerekli olan bilgiyi aramak yerine birbirlerine benzer verileri gruplandırmaktır. Finans departmanın her sene yaptığı finansal toplantıların analizlerini yapmak yerine şirketin satışlarını inceleyerek veri tabanı yardımıyla

[24] Serkan Savaş, Nurettin Topaloğlu, Mithat Yılmaz, **İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi**, Yıl:11 Sayı: 21 Bahar 2012, 1-23

satışların artışında etken olan müşteri segment profilini ortaya çıkarmak veri madenciliğidir denebilir.[25]

5.3. Veri Madenciliği Uygulama Alanları

Veri madenciliğinin kullanım alanları çeşitli sektörlerde farklılık göstermekle beraber birçok alanda işlevsel olarak yararlanılmaktadır. Veri madenciliği yöntemleri en çok pazarlama, bankacılık, sigortacılık, sağlık sektörü ve satış alanlarında sıklıkla kullanılmaktadır.

Pazarlama departmanlarında veri madenciliğinden ağırlıklı olarak, müşterilerin satın alma örüntülerinin belirlenmesi, müşterilerin demografik özellerinin satın alma kararı üzerindeki etkilerinin ortaya çıkarılması, mevcut halde bulunan müşterinin elde tutularak şirkete yeni müşteri kazanımında bulunulması ya da risk yönetimi ve dolandırıcılıkların ortaya çıkarılması gibi konularda yararlanılmaktadır.

Veri madenciliğinin kullanım alanları ve oranları tablo 4'deki gibidir;

Tablo 4: Veri Madenciliğinin Kullanım Alanları

Kullanım Alanı	Kullanım Oranı %
CRM müşteri Analitiği	32
Bankacılık	24
Direk Pazarlama	16
Kredi Puanlama	15
Telekomünikasyon	14
Dolandırıcılık Tespiti	13
Satış	11
Reklamcılık	10

[25] Burhan Gemici**, Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Ekonometri Anabilim Dalı Ekonometri Programı Yüksek Lisans Tezi,** 2012, 10

Sigortacılık	10
İlaç	7

Kaynak: Yalçın Özkan, Veri Madenciliği Yöntemleri, 2013, 38

Veri madenciliğinin tanımında yer alan, veri madenciliğinin daha önceden bilinmeyen ve tahmin edilmeyen verileri ortaya koyduğuna ilişkin en popüler örnek ise çoğu kişi tarafından bilinen bira-çocuk bezi örneğidir.

Ünlü bir perakende mağazalar zincirinin veri madenciliği yöntemini kullanarak yaptıkları araştırmalar sonucunda, cuma günleri bebek bezi ve bira alımı arasında oldukça kuvvetli bir ilişkinin olduğu ortaya çıkmıştır. Cuma günleri çocukları için alışverişe çıkan babaların aynı zamanda kendilerine de alışveriş yapmakta oldukları görülmüş ve mağazalar zinciri market içerisinde ürün konumlandırmasını buna göre şekillendirmiştir. Bu örnekle veri madenciliğinin bilinmeyeni nasıl ortaya koyduğu çarpıcı bir şekilde ortaya çıkmaktadır.

Bankacılık sektöründe ise Amerikan Bankası kendi ürünlerini kullanan banka müşterilerinin tespitinde veri madenciliği yöntemini kullanmakta ve müşterilerin ihtiyaçlarını karşılamak üzerine tasarladığı ürün ve servisleri sunmaktadır.

Twentieth Century Fox film şirketinde ise veri madenciliğini fatura bilgileri üzerinde kullanarak hangi filmin ya da hangi aktörün hangi bölgelerde daha çok izlendiğini belirleyerek bölgesel bazda bir film gösterim ağı ortaya sunduğu görülmüştür.

Veri madenciliğinin kullanım alanları yukarıda sözü edilen alanlarla sınırlı kalmayarak her sektörde birbirinden bağımsız olarak her konuyla ilgilenmektedir.[26]

5.4. Veri Madenciliğinin İşlevi

Veri madenciliği, hazırlanan veri gruplarından çekilmek istenen bilgilerin alınması amacıyla çok farklı işlevleri görebilmektedir. Veri madenciliğinin işlevlerini aşağıdaki gibi gruplandırabiliriz[27]:

1. Tanımlama,
2. Sınıflandırma,

[26] Silahtaroğlu, 11-12
[27] Erol Tutar, **Veri Madenciliği Yöntemiyle Döviz Kuru Tahmini**, 2011, 6

3. Kümeleme,
4. Tahmin,
5. Birliktelik.

5.4.1. Tanımlama

Bir verideki modeli tanımlamak, muhtemel model ve trendin açıklamalarını içermektedir. Veri madenciliğindeki model olabildiğinde açık ve anlaşılır olmalıdır. Modelin sonuçlarının açıklanması için yön gösteren bir tarifinin olması esastır. Böylece model herkes tarafından rahatça anlaşılabilir duruma gelmektedir.

5.4.2. Sınıflandırma

Sınıflandırma ise daha önceden tanımlanmış olan verinin birçok sınıfa ayrılmasıyla oluşturulur. Sınıflandırılmış ve daha önceki kaynaklarda kullanılan verilerden elde edilen bilgileri kullanarak, daha önce görünmemiş veri kayıtlarının oluşturulmasından sonra sınıflandırılmasıyla elde edilir.

5.4.3. Kümeleme

Verilerin birbirleriyle olan ilişkisine bakılarak benzerlik ya da yoğunluğundan yararlandıktan sonra verilerin doğal gruplandırılma ve eşleştirme işlemine kümeleme denilmektedir. Elde edilen çeşitli kümeler birbirleriyle eşleştikten sonra benzer davranışlar sergileyen veriler alt kümeler oluşturup bölünmektedir. Veri yığınının ilk halinden sonraki kümeleme sonrası olması gereken durumu birbirlerine benzer alt gruplara ayrılmış olarak belirlenmesi beklenmektedir.

5.4.4. Tahmin

Tahminleme, gelecek ile ilgili olan sonuçları içerir. Sınıflandırmaya benzemektedir ve tahmin genellikle iş ve araştırma gereken konularda kullanılmaktadır. Örneğin bir basketbol karşılaşmasında kazananın tahmin yöntemi ve istatistiki verilerden yararlanılarak bilinmesi yöntemi ya da bir şirketin gelecek altı ay için satış tahminlerinin yapılmasında kullanılan yöntemdir.

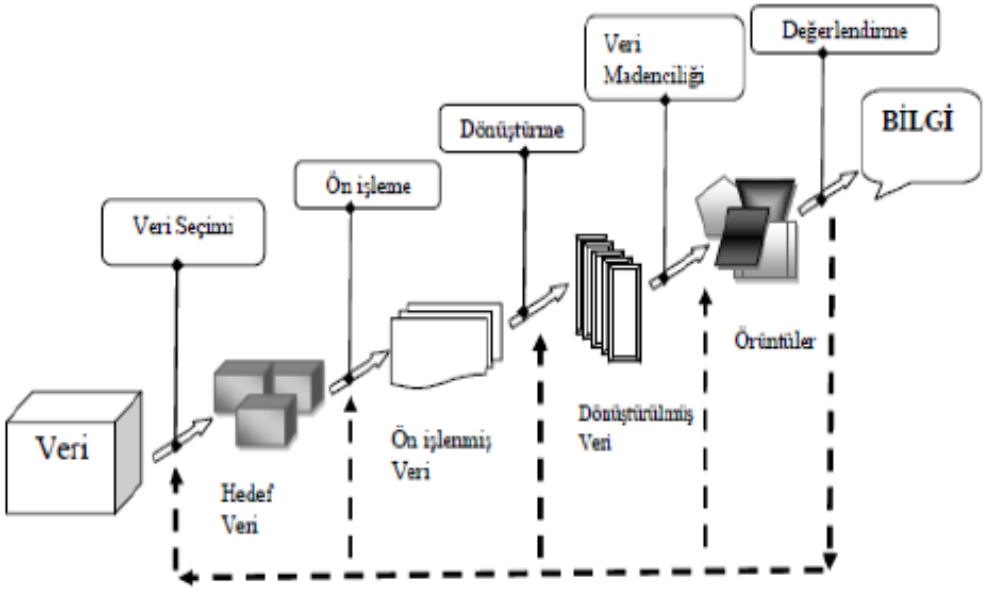
5.4.5. Birliktelik

Çok sayıda olan nitelik arasındaki ilişkinin ortaya konulması ve hangi niteliğin birleşeceğinin bilinmesi işidir. İş yaşamında genellikle ilişki analizi kurulurken karşımıza çıkmaktadır. Örneğin bir market zincirinde satılan ürünlerin hangilerinin

beraber satıldığını ve hangilerinin bir arada satılmaması gerektiğinin bulunması birliktelikle sağlanır.[28]

5.5. Veri Tabanında Bilgi Keşfi Süreci

Veri tabanlarında, veri madenciliğinin bilgi keşfi konusu daha geniş çaplı ele alınmalıdır. Veri analizi yapılırken veri madenciliğinin tekrarlayan yönlerinin olduğu bilinmektedir. Bu süreçte her veri analiziyle beraber yeni bilgiler ve yeni hipotezler ortaya çıkabilir; bu da yeni ve çeşitli soruların belirmesine neden olabilmektedir. Verilerden elde edilecek gizli bilgilerin ortaya çıkması konusunda da veri madenciliğini kullanan araştırmacının da geçmiş tecrübelerinden yararlanılmalıdır. Veri işlenmeleri aşamalardan oluştuğu için bilgi keşfinde bu süreçler arasında geçişler de görülebilmektedir[29].



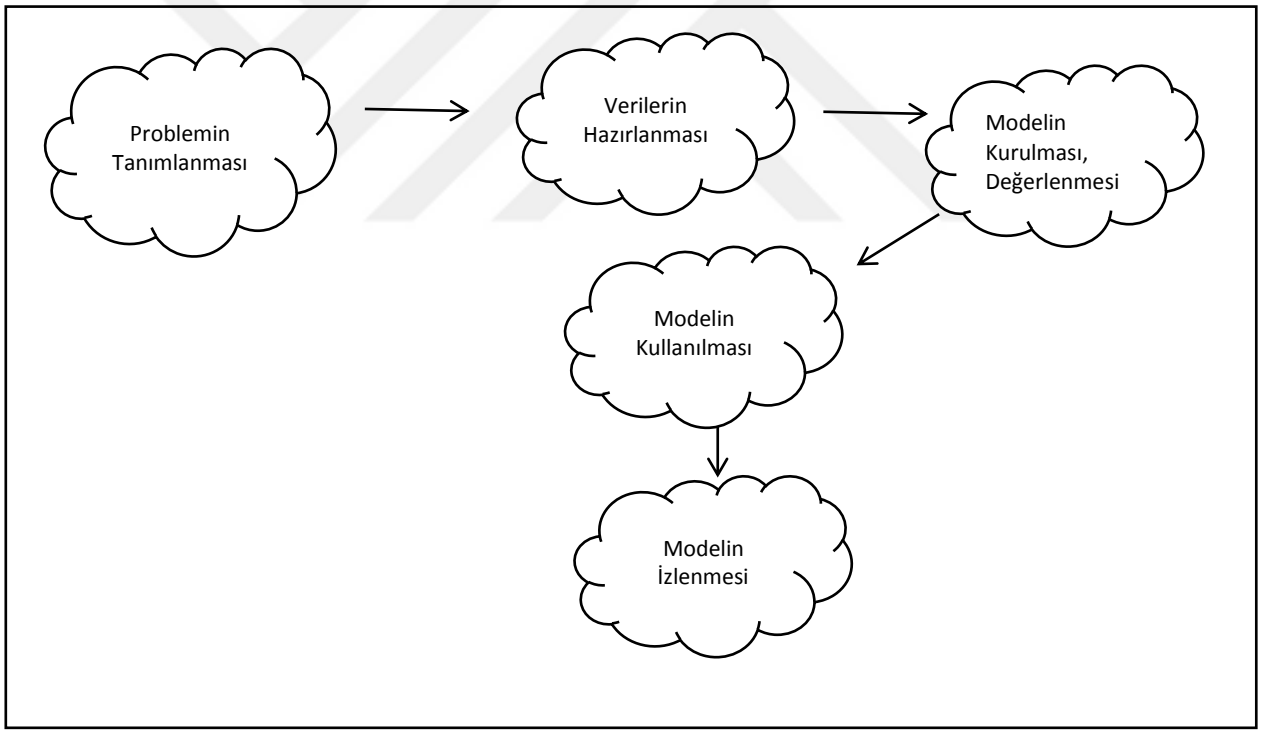
Şekil 9: Veri Madenciliği Bilgi Keşfi Süreci

[28] Erol Tutar, **Veri Madenciliği Yöntemiyle Döviz Kuru Tahmini**, 2011, 6
[29] Yasemin Koldere Akın, **Veri Madenciliğinde Kümeleme Algoritmaları ve Kümeleme Analizi**, Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, 2008, 3

Veri madenciliği bir süreç olarak tanımlandığında sürecin tüm seviyeleri hassasiyetle incelenmelidir. Her bir süreç çıktısı bir diğer sürecin girdisi olarak görülmekte ve buna bağlı olarak her bir süreç bir öncekinin sonucuna bağlı olarak ilerlemektedir[30].

Veri madenciliğinde araştırmacının en çok üzerinde durduğu konu veri hazırlama bölümüdür. Bu bölümü; problemin belirlenmesi, veri analizi ve son olarak çıkan sonuçların yorumlanması izlemektedir

Veri madenciliği organize yürütülmesi gereken bir süreçtir. Kullanılan süreçlerde yapılan bazı hatalar örneğin, verinin sınırlarının belirlenmemesi ya da verinin boyutlarının bilinmemesi araştırmacı için zaman kaybıyla beraber ciddi hatalar ortaya çıkarabilmektedir. Bu hataların ortaya çıkmasını engellemek adına ve süreçlerin standartlarının belirlenmesi amacıyla The Cross- Industry Standard Process for Data Mining (CRISP-DM) konsorsiyumu veri madenciliği süreçlerini standardize ederek belirlemişlerdir[31].



Şekil 10: Veri Madenciliği Süreci

Veri tabanlarında bilgi keşfi sürecini yukarıdaki bölümde sınıflandırdıktan sonra, aşağıdaki gibi ayrıntılı olarak açıklamak mümkündür:

[30] Savaş Serkan, Nurettin Topaloğlu, Yılmaz Mithat, **Veri Madenciliği ve Türkiye'deki Uygulama Örnekleri**, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi Sayı: 21, 2011, 8
[31] Umman Tuğba, Şimşek Gürsoy**, Veri Madenciliği ve Bilgi Keşfi**, 2009, 32

5.5.1. Problemin Tanımlanması

Veri madenciliği projelerinde en önemli konu, mevcut durumun iyi analiz edilmesinin ardından ortaya çıkmış olan problemin doğru tespit edilmesidir. İlk aşamada projenin limitleri en doğru şekilde belirlendikten sonra o proje için amaç ve gereksinimler oluşturulmalıdır. Amaç ve gereksinimlerden sonra olması gereken bir diğer gereksinim ise optimum noktaya ulaşmak için yapılması gereken çalışmalardır.

Bir kitap yayım şirketi için ‘’En çok okunan yazarlar kimlerdir? ‘’ sorusuna verilecek cevap kolay bir SQL sorgusu ile veri tabanından çekilebilmektedir. Fakat üst düzey yöneticiler tarafından gelen ‘’Okurlarımız için en önemli yazarlar kimlerdir?’’ sorusunun cevabı bir önceki sorunun cevabı kadar kolay olmamaktadır. Sorunun cevaplanabilmesi için en önemli parametrenin çok iyi belirlenmesi gerekmektedir. Belirlenen en önemli kavramından sonra bu kavrama bağlı olan alt parametrelerinde oluşturulması gereklidir. Bu parametrelere ulaşılabilecek olan yardımcı veriler belirlenmeli ve buna nasıl ulaşılacağı açıkça belirlenmelidir. Bu şekilde problemin tanımlanması için gerekli adımlar oluşturulmuş olmaktadır[32].

5.5.2. Verinin Hazırlanması

Verinin hazırlanması, veri madenciliğinin en önemli safhalarından biridir. Bu safhada modelin kurulması aşamasında karşımıza çıkacak herhangi bir sorunda sık sık geri dönülerek tekrar gözden geçirilebilir. Veri önişleme aşaması en çok çaba ve zaman gerektiren kısımdır. Çok büyük veri tabanları düşünüldüğünde bu veri tabanının içindeki datalar kullanımı bazen mümkün olmayan verilerden oluşmaktadır. Tutarsızlıklar ve problemlerle karşı karşıya kalınan bu gibi büyük veri tabanlarında operasyonel işlemlere tabii olan veriler veri madenciliği için uygun hale getirilmek üzere çalıştırılmaktadır. Bu işlemlerden sonra artık veri tabanında daha kaliteli ve veri madenciliği için daha verimli bir çalışma alanı oluşturulmuş olunur. Verilerin hazırlanmasında geçen süre veri yığınının büyüklüğü ile eş zamanlı düşünülebilir[33].

Verilerin hazırlanması dört aşamadan oluşmaktadır;

1.Verilerin Toplanması,

2.Veri Birleştirme ve temizleme,

[32] Şenol Gökmen, **Müşteri İlişkileri Yönetiminde Bir Araç Olarak Veri Madenciliği ve Perakende Sektöründe Bir Uygulama**, 2014, 42
[33] Gürsoy, 33

3.Veri İndirgeme,

4.Veri Dönüştürme.

5.5.2.1. Verilerin Toplanması

Veri toplama aşamasında problem için gerekli olan verilerin toplanmasında öncelikle veri kaynaklarının belirlenmesi gerekir. Araştırmacı bu verileri birincil veri kaynaklarından bulabileceği gibi farklı veri tabanlarından da bulabilmektedirler.

Verilerin toplama aşamasında ulaşılan veri tabanındaki verilerin dikkatle incelenmesi gerekir. Veri tabanındaki bilgilerde birçok kayıtın doğru olduğu ve cinsiyet bilgilerinin bulunduğu görülürken birkaç veri de cinsiyet bilgileri boş bırakılmış ya da hiç girilmemiş olabilmektedir. Bu tip verileri kayıp veriler olarak isimlendirilir. Bazı durumlarda ise verilerim uç noktalarda olabileceği gözlenmektedir. Kişinin veri tabanında var olan yaşı 897 olarak gözükmektedir. Bu bilgilere "gürültülü" veri adı verilmektedir. Bu gibi verilerin tespit edildikten sonra verilerin toplanması veri madenciliği için hem zaman tasarrufu hem de kaliteli bir çalışma ortamı oluşturmaktadır [34].

5.5.2.2. Veri Birleştirme ve Temizleme

Toplanan veri kaynaklarının farklı olması ya da çeşitli veri tabanlarının kullanılması sonucu hazırlanan kaynakların birlikte değerlendirmeye alınabilmesi için tek bir türe dönüştürülmesi gerekir. Veri madenciliği için çalışan bir araştırmacı bir veri ambarı oluşturduysa bu ambarın tek bir türde bütünleştirme işleminin yapılmış olması gerekmektedir.

Veri birleştirmeden sonra elde edilen veri tabanı bazen istenilen özellikte olmayabilir. Eksik veriler, birbirleriyle tutarsız ve hatalı verilerin bulunduğu durumlarla karşılaşılabilir. Bu gibi durumlar gürültülü veri olarak adlandırılmaktadır. Eksik veriler yerine doğru veriler ile tamamlanmalıdır.

Eksik veri veya gürültülü veriyle karşılaşıldığında aşağıdaki yöntemler kullanılabilir;

1. Veri kümesi içerisinde var olan eksik veriler mevcut veri tabanından atılabilir,

2. Tüm eksik veriler yerine standart olarak belirlenmiş bir veri kalıbı getirilebilir. Örneğin; ~Bilinmiyor~ değeri kullanılabilir,

[34] Yunus Köse, **Değerli Müşterilerde Ürün Kategorileri Arasındaki Satış İlişkilerinin Veri Madenciliği Yöntemlerinden Birliktelik Kuralları ve Kümeleme Analizi ile Belirlenmesi ve Ulusal bir Perakendecide Örnek Uygulama**, Selçuk Üniversitesi Yüksek Lisans Tezi, 2015, 60

3. Tüm veriler göz önüne alınarak bir ortalama hesaplandıktan sonra eksik veriler yerine bir değer ataması yapılabilir,

4. Eksik değer; karar ağacı ya da regresyon yöntemi kullanılarak tahmin edilmeye çalışılarak eksik değer için bulunan bu verinin kullanılması uygun olmaktadır.[35]

5.5.2.3. Veri İndirgeme

Veriyi çözümleme işlemi veri madenciliğinde bazen çok uzun zaman alabilmektedir. Oldukça büyük olduğu düşünülen bir veri tabanı yapısı varsa ve çözümlemede bir değişikliğe neden olmayacak ise veri sayısında azaltılma yapılabilmektedir.[36]

Tablo 5'de veri indirgeme işlemlerinin çeşitleri gösterilmektedir;

Tablo 5: Veri İndirgeme Yöntemleri

Veriyi Birleştirme veya Veri Küpü
Boyut İndirgeme
Veri Sıkıştırma
Örnekleme
Genelleme

Verileri çok boyutlu veri küpleri haline dönüştürmek, veri indirgeme aşamasında kullanılabilir. Bu durum araştırmacının işini sadece boyutlara göre çözümlemeler yapabilmesi açısından kolaylaştırmaktadır. Oluşturulan veriler arasından tercihler yapılarak veri tabanında kullanmaya gerek olmayan veriler çıkarılarak veri tabanı boyutunun azaltılması sağlanabilmektedir.

[35] Yalçın Özkan, 40
[36] Micheline Kamber, 67

Veri sıkıştırması işleminde ise var olan büyük yapıdaki kümelerin daha az yer işgal etmesinin önüne geçilmesi amacıyla bir sıkıştırılma yapılmaktadır. Örneklemede bu hantal yapı yerine yeni oluşturulmuş daha küçük veri kümeleri kullanılmaktadır.

5.5.2.4. Veri Dönüştürme

Veri madenciliği için yapılan çalışmalarda bazen kullanılan kaynakların çözümlemeye doğrudan katılması doğru olmayabilir. Değişkenler açısından, her değişkenin ortalama ve varyansları birbirlerinden farklı olacağından ortalaması daha büyük olan değişkenlerin diğerleri üzerinde oluşturduğu hâkimiyet daha fazla olmakta ve diğerlerinin rollerini önemli ölçüde azaltmaktadır. Değişkenlerin içerisindeki değerlerin kendi içlerindeki çok büyük ya da tam aksi küçük değerler olması çözümleme esnasında yanlış sonuçlara gidilmesine neden olabilmektedir. Veri dönüştürme işlemini bu gibi durumlarda veriler arasında çözümlemeyi etkilememesi için kullanmak gerekir.

5.5.3. Modelin Kurulması ve Değerlendirilmesi

Olson ve Shi ye göre modelleme aşaması, veri madenciliği yazılımı yardımıyla uygun teknikler kullanarak farklı durumlar için sonuçlar üretilmesi aşamasıdır. Veri madenciliği süreçlerinin tamamlanmasından sonra en iyi modelin seçilmesi gerekmektedir. Modellerin hepsinin kurulup, birbirleriyle karşılaştırılarak en iyi algoritmanın tespit edilmesinden sonra model kurulma aşaması tamamlanmaktadır.

Modelin kurulmasında genel olarak kümeleme analizi ile verinin görselleştirilmesi teknikleri uygulanmaktadır. Model uygulamaları verilerin tiplerine göre değişiklik göstermektedir. Veri tipinde amaç gruplandırmak ise diskriminant analizi kullanılması tercih edilebilir. Amaçlardan bir diğeri tahmin ve eldeki veri sürekli ise regresyon analizi kullanılabilir. Verilerin sınıflandırılmasında ise karar ağacı tekniği tercih edilmektedir[37].

Modellerin kurulması aşamasındaki değişkenlerin arasındaki ilişki düzeyleri karşılaştırılıp model için her zaman daha anlamlı olan değişkenleri seçmek sağlıklı olmaktadır. Model seçiminin tamamlanmasından sonra eğer en doğru modelin

[37] David Olson, Yong Shi, **Introduction To Busıness Data Mining**, 2007, 24

seçildiği biliniyorsa bir sonraki adım olan modelin değerlendirilmesi aşamasına geçilebilmektedir[38].

Modelin değerlendirilmesi; Modelin sonuçlarının değerlendirilmesi, gerçekleşen süreçlerin değerlendirilmesi ve bir sonraki adıma karar verilmesi şeklinde gerçekleşir. Modelin değerlendirme aşamasında; hangi modelin amaçlanan projenin hedeflerine ne doğrultuda gidilmesine olanak sağladığına göre değişmektedir. Model sonucunda amaçlanan hedef başarılamadıysa bunun sonuçlarının da değerlendirilmesi gerekmektedir.

Araştırmacı tarafından sürecin sonucunda işin nasıl devam etmesi gerektiğine karar verilerek; gerektiğinde yeni bir veri madenciliği sürecinin başlayıp başlamayacağı kararına varılır.[39]

5.5.4. Modelin Kullanılması

Model seçilip değerlendirildikten sonra doğrudan kullanılabileceği gibi başka bir çalışmanın bir parçası olarak da kullanılabilmektedir. Söz konusu bu modeller çok çeşitli iş analizlerinde de etkili bir şekilde çalıştırılabilmektedir[40.]

5.5.5. Modelin İzlenmesi

Modelin izlenmesi aşaması ise veri madenciliği sürecinin son aşamasında görülmektedir. Kurulan modelin işlemesinin ardından geçerliliğinin kanıtlanmasıyla beraber kurulan sistemin sürekli izlenmesi gerekmektedir ve gerekli görülen yerlerde müdahale edilip düzeltilme işlemi yapılmalıdır.

[38] Bülent Ordu, **Veri Madenciliğinde Sınıflayıcı Teknikler ile Demir Çelik Sektöründe Bir Uygulama**, 2013, 39
[39] Pete Chapman, Julian Clinton, **Crisp-Dm 1.0 Step-By-Step Data Mining Guide**, 2000, 28
[40] Haldun Akpınar, **Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği**, İstanbul Üniversitesi İşletme Fakültesi Dergisi, 2000, 29

6. VERİ MADENCİLİĞİ TEKNİKLERİ VE MODELLERİ

Veri madenciliğinde modeller tanımlayıcı ve tahmin edici olarak ikiye ayrılmaktadır. Tanımlayıcı modeller; mevcut veriyi analiz etmek ve yorumlamak için kullanılırken, tahmin edici modeller; gelecekteki eğilimi ile sonuçların ne olabileceğine odaklanmaktadır. İşlevsel olarak ise veri madenciliği modellerini üç ana gruba ayırarak incelemek mümkündür:

1. Sınıflama ve Regresyon
2. Kümeleme
3. Birliktelik Kuralları

Bu modellerden kümeleme ve birliktelik kuralları tanımlayıcı, sınıflandırma ve regresyon ise tahmin edici modellerdir[41].

6.1. Sınıflandırma ve Regresyon Teknikleri

Sınıflandırma kavramı, genel olarak veriyi, tanımlanan sınıflara dağıtmak ya da sınıflara ayırmayı ifade etmektedir. Burada eldeki verinin bir sınıfa atanıp atanamamasıyla ilgilenilmektedir. "Nesneler veya durumlar için uygun sınıf tahmin edilmesi olarak da tanımlanabilecek sınıflandırmanın girdileri, her biri bir sınıf etiketi ile etiketlenecek gözlem veya örneklerden oluşan bir eğitim kümesidir. Çıktı ise modelin her bir gözlem için niteliklere dayalı olarak atadığı sınıf etiketidir[42]."

Regresyon ise süreklilik gösteren değerleri tahmin ederken kullanılan bir yöntemdir. Sınıflandırma, bir veriyi sınıflara koyarak tasnif ederken, regresyon veriyi gerçek değerli tahmini bir değişkene atar[43].

Veri madenciliğinde çok sık kullanılan sınıflandırma tekniklerinin birçok kullanım alanı bulunmaktır. Bu yöntemlerin en çok kullanıldığı sektörleri aşağıdaki gibi özetleyebiliriz:

---

[41] age, 3

[42] Aytaç Yıldızeli, Aykut Arıkan, Tolga Çakmak, **Bilgi Çağında Varoluş: "Fırsatlar ve Tehditler" Sempozyumu Yeditepe Üniversitesi,** 2009, 156

[43] Usama Fayyad, Gregory Shapiro, **From Data Mining to Knowledge Discovery in Databases**, American Associatin for Artificial Intelligence, 1996, 37-54

1. **Sağlık**: kişiye özel hastalıkların tespiti ve teşhisinde,

2. **Egitim**: Öğrencilerin özelliklerine göre gruplara ayırıp, sınıflara yerleştirilmesinde,

3. **Hukuk**: Vasiyetin kim tarafından yazılmış olma ihtimaline karar verilmesinde,

4. **Bankacılık**: Musteri davranışlarına göre, belirli profiller çıkartılarak, risk grupları oluşturmada kullanılır[44].

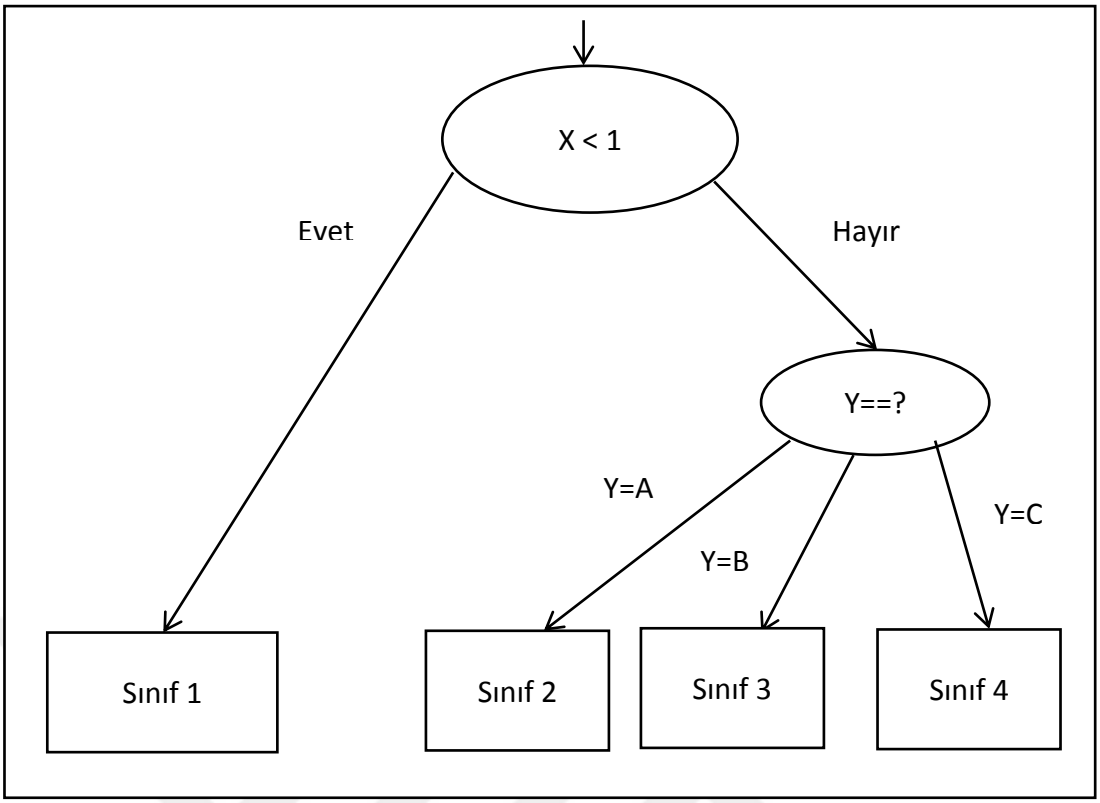
6.1.1. Karar Ağaçları ile Sınıflandırma

Karar ağaçları, veri sınıflandırma yöntemleri arasında en çok kullanılanlardan biridir. Önceden belirlenmiş ve karar almada yardımcı kurallar konularak model oluşturulur ve bu sayede yeni üyeler, modeldeki derecelerine göre farklı sınıflara ayrılır.

Karar ağaçları, veri tabanlarında kolayca modellenebildikleri, düşük maliyetli, kolayca yorumlanabildikleri ve güvenilirliklerinin yüksek olmasından dolayı veri madenciliği teknikleri arasında en çok kullanılan sınıflandırma tekniklerindendir.

Karar ağaçları yapıları gereği akış şemalarına benzemektedir. Her değer bir düğüm tarafından dallanır. Kök, dallar ve yapraklar ağaç yapısının birer parçasıdır. En üst yapı kök, en sondaki yapı yaprak ve bunlar arasında kalan yapılar ise dal olarak adlandırılmaktadır [45].

[44] Daniel T. Larose, **Discovering Knowledge In Data: An Introduction To Data Mining**. New Jersey: A. John Willey&Sons, 2005, 180
[45] John Ross Quinlan, **Induction of Decision Trees**, Machine Learning, 1986, 81-106



Şekil 11: Karar Ağacı Örneği

Yukarıdaki şekil üzerinde X ve Y girişlerinden oluşan bir örnek sınıfının basit karar ağacı gösterilmiştir.

Sınıflandırma yapılırken;

- X < 1 ise sınıf=1,
- X > 1 ve Y=A ise sınıf =2,
- X > 1 ve Y=B ise sınıf =3,
- X > 1 ve Y=C ise sınıf =4

şeklinde sorgular ile sınıflar birbirinden ayrılmaktadır.

Karar ağaçlarına göre sınıflandırma yapılırken birçok algoritma geliştirilmiştir. En çok kullanılanlar arasında ID3, C4.5, CART, SPRINT ve SLIQ bulunmaktadır.

6.1.2. Yapay Sinir Ağları

Yapay sinir ağları, biyolojik sinir ağlarından esinlenerek yapılmış bir veri analiz sistemidir. Yapay sinir ağları, nöronlar, katmanlar, hücreler ve düğümlerden oluşmaktadır. Her nöron bağlantılar ile diğer nöronlara bağlıdır. Nöronlar da ise

karar katmanları bulunmaktadır. Kendine sinyal geldiğinde, içindeki kurallar ile iletim yapıp yapılmayacağına karar verir. Bu süreç ağ bitene kadar devam eder ve sonuçta, veri sınıflandırılmış olur. Genel olarak ise girdi katmanı, gizli katman ve çıktı katmanından oluşmaktadır[46].

Yapay sinir ağları sınıflandırmada kullanıldıkları zaman, iç algoritması bir öğrenme sürecine girer. Literatürde Makine Öğrenmesi (Machine Learning) olarak da bilinen bu öğrenme ile çıktı katmanına ulaşabilmek için verilerin ağırlıkları hesaplanmaya çalışılır. Öğrenme için ayrılan veri kümesini bitirdikten sonra kalan veriler ağırlıklandırılır ve öğrenme sürecinin doğruluğu test edilir. Testin sonucundaki ağırlıklar öğrenme sürecindeki ağırlıklar ile örtüşür ise algoritma öğrenme işlemini tamamlamış olur. Eğer test sonucu ağırlıklar örtüşmez ise ağırlıklar üzerinde düzeltmeler yapılması gerekir. Yapay sinir ağlarının öğrenme süreci zaman alabilmekte, fakat öğrenme işlemi tamamlandığında çok hassas ve doğruluğu yüksek bir şekilde sınıflandırma yapılmış olmaktadır[47].

6.1.3. Genetik Algoritmalar

Genetik algoritmalar, "İşletme problemine tek bir çözüm yerine farklı çözümlerin bir araya gelerek oluşturduğu bir çözüm kümesi üretmeyi esas alan, doğadaki evrimsel süreci bilgisayar ortamında taklit eden bir arama ve eniyileme yöntemidir[48]".

Genetik algoritmalar, özellikle pazarlama araştırmalarında sıklıkla kullanılmaktadır. İlgili pazarlardaki bilgi keşfi aşamasında kullanılarak, araştırmanın devamı için bilgi sağlanır.

6.1.4. Bellek Tabanlı Sınıflandırma

En yaygın olarak kullanılan algoritmalardan birisi *K-Nearest Neighbor* ya da *KNN* olan K-En yakın Komşu algoritmasıdır. Algoritmanın temel mantığı, sınıflandırma yaparken, her bir kaydın diğer kayıtlara uzaklığı ölçülür. Ancak, bir kayıt için diğer kayıtlardan sadece *k* âdeti göz önüne alınır[49].

---

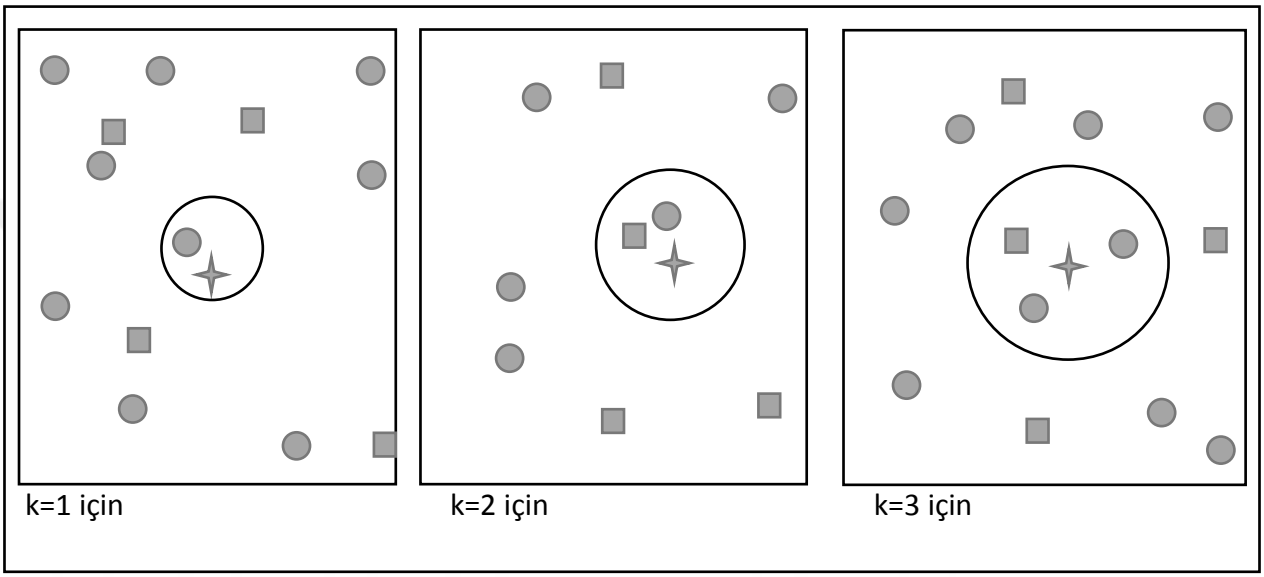
[46] Laurene Fausett, **Fundamentals of Neural Networks: Architectures, Algorithms And Applications,** 1994, 3

[47] Gökhan Silahtaroğlu, **Veri Madenciliği**, 2008, 66

[48] Abdülkadir Özdemir, **Lise Türü Ve Lise Mezuniyet Başarısının, Kazanılan Fakülte İle İlişkisinin Veri Madenciliği Tekniği İle Analizi,** Atatürk Üniversitesi, Sosyal Bilimleri Enstitüsü Dergisi, Cilt 10, Sayı 2, 2007, 446

[49] Beyer Kevin, **When is Nearest Neighbor Meaningfull**, 7. Uluslararası Database Theory Konferansı, İsrail, 1999, 217-235

K değeri kullanıcı tarafından seçilmektedir. K değeri çok küçük seçilirse, sınıf içindeki eleman sayısı azalacağı için sınıf sayısı artacaktır fakat asıl sorun sınıf sayısının artması değil benzer sınıflarda olması gereken elemanların farklı sınıflara konulmasıdır. K değerinin çok büyük seçilmesi durumunda ise, sınıf sayısı azalacak ancak sınıf içindeki homojenlik de azalacağından, farklı sınıflarda olması gereken elemanların aynı sınıfta yer alması sorunu ortaya çıkacaktır[50].



Şekil 12: K Değerleri için En Yakın Komşu Sınıflandırması

Yukarıdaki şekilde örnek k değerleri için en yakın komşular gösterilmiştir. K değeri büyüdükçe sınıf dairesi de büyümektedir. Sınıflandırma için elemanların uzaklıkları hesaplanır. En çok kullanılan uzaklık ölçüsü ise Öklid Ölçüsüdür.

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

Eşitlik 1'de,

- x = $x_1,\ x_2,\ x_3, \dots, x_n$
- y = $y_1,\ y_2,\ y_3, \dots, y_n$

[50]Gökhan Silahtaroğlu, **Veri Madenciliği**, 2008, 65

şeklindedir.

Algoritma adımları aşağıdaki gibi sıralanabilir:

- Noktalar arası uzaklık ölçüm uzayı belirlenmelidir,
- En yakın k adet noktayı belirlenmelidir,
- Belirlenen grubun en çok bulduğu sınıfı belirlenmelidir,
- Bu gruba sınıfın ismini verilmelidir.

6.1.5. İstatistiksel Sınıflandırma

İstatistiksel sınıflandırma, veri madenciliği literatüründe Bayesyen sınıflandırma olarak bilinmektedir. Bayes teoremini temel olan bu sınıflandırma tekniği, elde var olan, sınıflanmış verileri kullanarak yeni bir verinin mevcut sınıflardan herhangi birine girme olasılığını hesaplayan bir yöntemdir[51].

Bayes teoremini aşağıdaki 2 nolu eşitlik ile gösterebiliriz.

$$P(C_1/x_i) = \frac{P(C_1)\,P(x_i/C_1)}{P(x_i)} \qquad (2)$$

2 Nolu Bayesyen algoritmasında, verilen öğrenme kümesinde $P(C_1)$ değerini, her sınıfın verilen öğrenme kümesi içinde bulunma olasılığını hesaplar. x_i ler sayılarak $P(x_i)$ bulunur. Benzer şekilde herbir x_i bulunma olasılığı $P(x_i/C_1)$, C_1 ler içinde x_i lerin sayılmasıyla elde edilir[52].

6.2. Kümeleme Analizi

Kümeleme analizi veya kısaca kümeleme, bütün halinde olan veri objelerinin parçalara ayrılarak, alt kümelere ayrılma işlemidir. Her alt kümedeki objeler birbirine benzer özellik gösterirken, farklı kümelere ayrılmış objeler ise birbirlerine benzememektedir. Kümeleme analizi için birçok yöntem geliştirilmiştir. Aynı veri setine, farklı kümeleme yöntemleri uygulandığında farklı sonuçlar ortaya çıkabilmektedir [53].

Kümeleme analizi, iş analizi, görüntü işleme, internet araştırmaları, biyoloji ve güvenlik gibi alanlarda yaygın bir kullanıma sahiptir. Bankacılık sektöründe

[51] Michael Berry, Murray Browne, **Lecture Notes In Data Mining Singapore: World Scientific Publishing,** 2006, 20
[52] Gökhan Silahtaroğlu, **Veri Madenciliği**, 2008, 61
[53] Micheline Kamber, 444

pazarlama faaliyetlerinden bir örnek vermek gerekirse, banka müşterilerini, müşterilerin davranış biçimine göre sınıflayarak, daha sonra yapacağı ve müşteriline sunacağı ürünlerini daha doğru şekilde doğru müşterilere ulaştırabilir. Bu şekilde her şirketin asıl amacı olan kar maksimizasyonunu sağlamak kolaylaşmaktadır.

6.2.1. Kümeleme Analizi İçin Gereksinimler

Kümelemr algoritmalarının geliştirilmesi için gerekli bazı bazı önemli özellikler bulunmaktadır. Bu gereksinimleri aşağıdaki gibi sıralamak mümkündür:

Ölçeklenebilirlik: Birçok kümeleme algoritması birkaç yüz veri objesini kapsayan küçük veri kümelerinde düzgün bir şekilde çalışmakta, fakat büyük veritabanları milyonlarca hatta milyarlarca veri objelerini kapsamaktadır. Büyük bir veri setinden alınmış bir örnekte kümeleme yapmak bile hatalı sonuçlar verebilmektedir. Bu yüzden yüksek ölçekli kümeleme algoritmaları gereklidir.

Farklı tiplerin özellikleri ile baş çıkabilme: Birçok algoritma sayısal verileri kümelemek için tasarlanmıştır. Fakat bazı uygulamalar binary, nominal, ordinal veya bunların karışımı olan veri tiplerini içerebilmektedir.

Kümelerin şekillerinin keşfi: Birçok kümelemr algoritması kümeleme yaparken Öklid veya Manhattan uzaklık ölümleri ile hesaplama yapmaktadır. Bu tarz algoritmalar küresel kümeler çıkarma eğilimindedir. Fakat ihtiyaçlara göre kümenin şekli küresel olmak zorunda değildir. Bu yüzden kümenin keyfi şekillerini bulacak şekilde algoritma geliştirmek önemlidir.

Giriş parametrelerini bulmak için alan bilgisi ihtiyaçları: Birçok kümeleme algoritması, kullanıcılardan giriş parametrelerinde alan bilgisi sağlamalarını istemektedir. Örneğin analizin kaç kümeden oluşması gerektiği gibi. Dolayısı ile kümeleme analizinin sonuçları bu parametrelere göre hassasiyet göstermektedir. Çoğu zaman ise özellikle çok boyutlu veri kümelerinde çalışırken bu parametreleri bulmak oldukça zordur.

Kirli veriler ile başa çıkabilme: Gerçek dünya verilerinin birçoğu aykırı, bilinmeyen veya hatalı bilgi içerebilmektedir. Örnek olarak, sensor okuma bilgileri, hissetme mekanizmalarından dolayı fazla veri toplar. Kümeleme algoritmaları, bu tarz gürültü içeren verilere karşı daha dayanıklı olmalıdır.

Artımlı kümeleme ve veri girişi sırasına hassas olmama: Birçok uygulamada, verilere güncelleme her an gelebilmektedir. Bazı kümeleme algoritmaları, var olan kümelenmiş yapıya artımlı veri güncellemesini desteklemez, bunun yerine son gelen veriler ile en baştan kümeleme yapılır. Ayrıca veri giriş sırasının değişmesi de kümelemeyi etkileyebilmektedir. Dolayısıyla artımlı kümeleme algoritmaları ve veri giriş sırasından bağımsız çalışan her sırada aynı sonucu veren algoritmalar istenmektedir.

Çok boyutlu veri kümeleme kabiliyeti: Bir veri seti birçok boyut ve özellik içerebilir. Birçok kümeleme algoritmasının düşük boyutlu veri setlerinde başarılı olmasının yanı sıra algoritmanın çok boyutlu veri setlerinde de çalışması beklenmektedir.

Kısıt bazlı kümeleme: Gerçek dünya uygulamaları kümeleme yaparken bazı kısıtlar altında çalışabilir. Örneğin şehirde nereye bankamatik koyulabilir sorusunu araştırırken, şehrin raylı sistemleri, ilçe merkezleri, varsa nehirlerin konumu gibi kısıtlar göz önünde bulundurulmalıdır.

Yorumlanabilirlik ve kullanılabilirlik: Kullanıcılar kümeleme sonuçlarının yorumlanabilir, anlaşılır ve kullanışlı olmasını istemektedir. Bu yüzden kümeleme anlamlı yorumlara bağlı kalmalıdır[54].

6.2.2. Kümeleme Analizi Yöntemleri

Kümeleme analizi kategorisine girebilecek birçok algoritma geliştirilmiştir. Algoritmalar, kümeleme oluşturma biçimine, kullanılan veri setine ve yapılacak çalışmanın amacını göre sınıflandırılabilmektedir. Bu yöntemleri genel anlamda dört ana başlık altında mümkündür. Bunlar:

- Bölümlemeli Yöntemler
- Hiyerarşik Yöntemler
- Yoğunluk Temelli Yöntemler
- Grid Temelli Yöntemler

Algoritmaların genel özelliklerine göre ise aşağıdaki Tablo 6 ile ana hatları çizilebilinir [55].

[54] Jiawei Han, 446-447
[55] age, 450

Tablo 6 : Kümeleme Yöntemlerime Genel Bakış

Yöntem	Genel Karakteristik
Bölümlemeli Yöntemler	• Birbirinden bağımsız küresel kümeler bulur. • Mesafe bazlıdır. • Ortalama veya merkez nokta ile küme merkezini temsil eder. • Küçük ve orta büyüklükte veri setlerinde etkilidir.
Hiyerarşik Yöntemler	• Kümeleme, hiyerarşik ayrışımdır. • Hatalı birleştirmeleri veya ayırmaları düzeltemez. • Mikro kümeleme veya benzeri teknikleri içerebilir.
Yoğunluk Temelli Yöntemler	• Farklı şekillerdeki kümeleri bulabilir. • Kümeler yoğun objelerin bölgesidir ve düşük yoğunluklu bölgeler ile ayrılır. • Küme yoğunluğu: Her nokta kendi "komşuları" içerisinde minimum sayıdaki noktalara sahip olmalıdır. • Aykırı değerleri filtreleyebilir.
Grid Temelli Yöntemler	• Çok çözünürlüklü grid yapısını kullanır. • Hızlı işleme zamanı. Veri objesi sayısından bağımsız, grid boyutuna bağımlı

6.2.2.1. Bölümlemeli Yöntemler

Kümeleme analizinde en çok kullanılan ve temel yöntem olan bölümlemeli yöntemler, veri topluluğunu kendi içinde birbirine benzeyen gruplara veya kümelere ayırmaktadır. Bölümleme işleminde, kaç kümeye ayrılmak istendiği önceden belirlenmiş olup kullanıcı tarafından ilgili algoritmaya parametre olarak verilmelidir.

Bölümlemeli yöntemler, hiyerarşik yöntemlere göre daha hızlı çalışmaktadır. Bunun nedeni ise hiyerarşik yöntemlerde olduğu gibi bir benzerlik matrisi kullanmamalarıdır. Bu sebeple büyük verilerde çalışmaya daha uygundur bölümlemeli algoritmalar. Ancak önceden verilmek zorunda olan sonuç küme sayısı, sonuçlardaki *güvenilirliğini* azaltmaktır. İstenen küme sayısının kesin olarak

bilinmesi durumu haricinde, optimum sonuca ulaşmak için birden fazla kez algoritma tekrarlanır, bu da efor kaynak tüketimini arttırmaktadır [56].

6.2.2.1.1. K-Ortalamalar (K-Means)

K-Means ya da K-Ortalama algoritması, her adım da kümelerin yenilendiği ve en optimum sonuca ulaşana kadar devam eden bir algoritmadır. Bu alandaki öncü algoritmalardandır ve bölümlemeli diğer algoritmalar, K-Means algoritmasından esinlenmiştir ve benzemektedir [57].

K-Means algoritmasındaki k, aranan küme sayısını ifade eder ve önceden belirlenmiş olmalıdır. Algoritma başlangıcında keyfi olarak yine k adet küme merkezi seçilir. Bu noktalar prototip olarak isimlendirilir. Kümeleme analizi başlangıcındaki durum k adet prototip ve her bir küme olmak üzere 3 nolu eşitlikteki gibidir.

$$w_i = i_l, j \in \{1, \dots, k\}, l \in \{1, \dots, n\} \quad (3)$$

K-Means algoritmasının matematiksel ifadesi için kullanılan 4 nolu eşitlikte, C_j ifadesi j. elemanı temsil eder. İşlemin kalitesi ise aşağıdaki hata fonksiyonu ile gösterilir[58].

$$E = \sum_{j=1}^{k} \sum_{i_l \in C_j} \left| i_l - w_j \right|^2 \quad (4)$$

Algoritma girdileri, adımları ve çıktısı aşağıdaki gibi açıklanabilmektedir:

Girdiler:

k: verilen küme sayısı

C: eldeki veriler

Adımlar:

- Keyfi olarak alınan *k* tane elemanın küme merkezi $(m_1, m_2, \dots, m_k)$ olarak belirlenmesi,
- Her bir elamanın en yakın olduğu m_i nin kümesine atanması,

[56] Dunham Margaret H., **Data Mining Introductory and Advanced Topics,** Prentice Hall, Pearson Education Inc., New Jersey, 2003, 8
[57] Silahtaroğlu, 114
[58] Turgay Tugay Bilgin, Yılmaz Çamurcu, **DBSCAN, OPTICS ve K-Means Kümeleme Algoritmalarının Uygulamalı Karşılaştırılması**, Politeknik Dergisi, Cilt: 8 Sayı: 2, 2005, 140

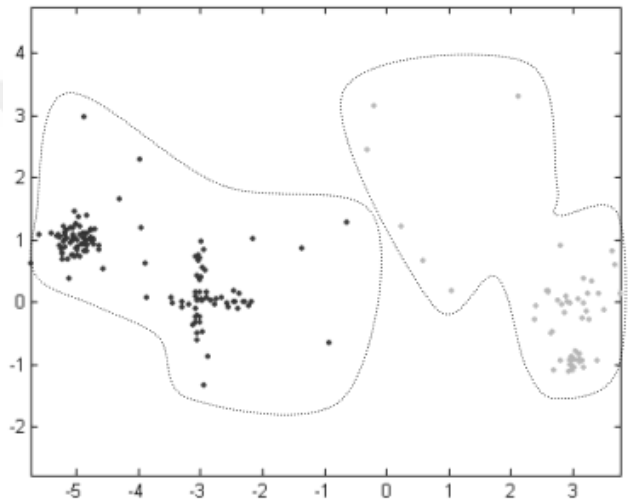
- Kümelere ait $m_1, m_2, \ldots, m_k$ değerlerinin tekrar hesaplanması,
- Küme de değişiklik olmayana kadar ilk adımdan itibaren devam edilmesi. Değişiklik yoksa durulması gerekmektedir.

Çıktılar:

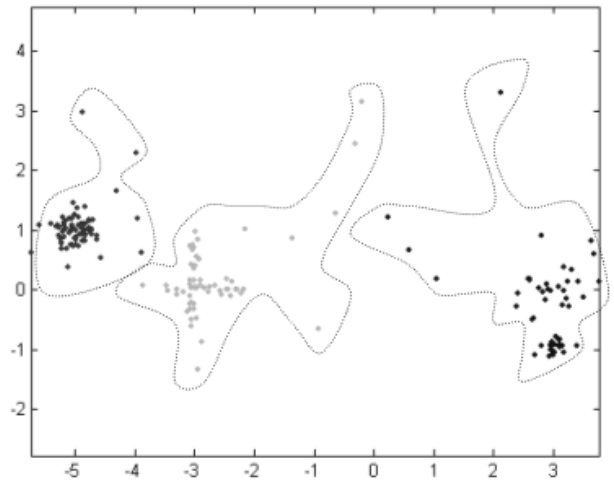
k adet küme[59]

şeklindedir.

Şekil 13, 14, 15, 16, 17 ve 18 ile algoritma giriş parametresi olan k değerleri için kümelerin değişimleri gösterilmiştir.

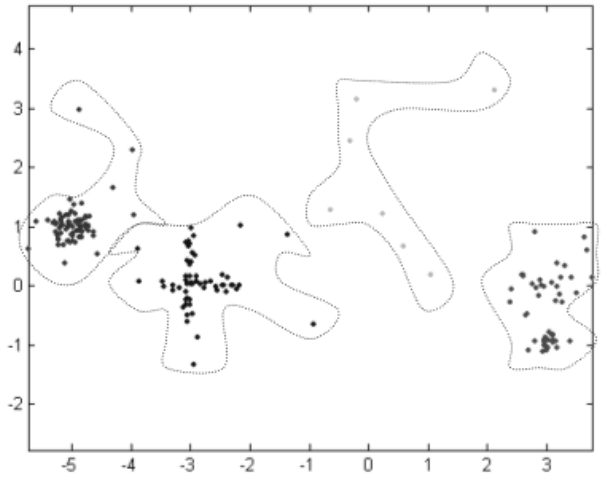


Şekil 13 : K=2 için K-Means Örnek Kümeleri

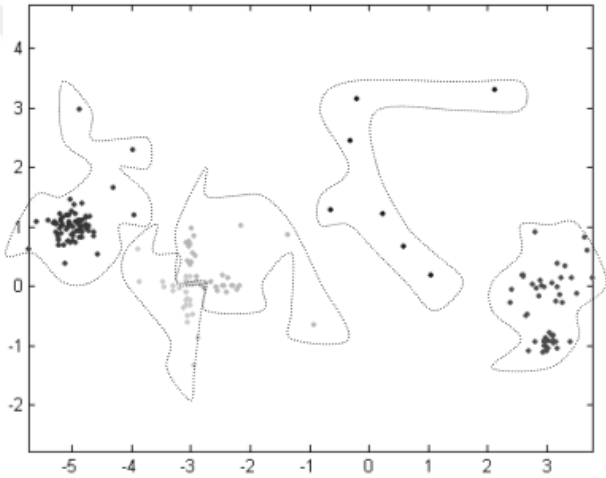


Şekil 14 : K=3 için K-Means Örnek Kümeleri

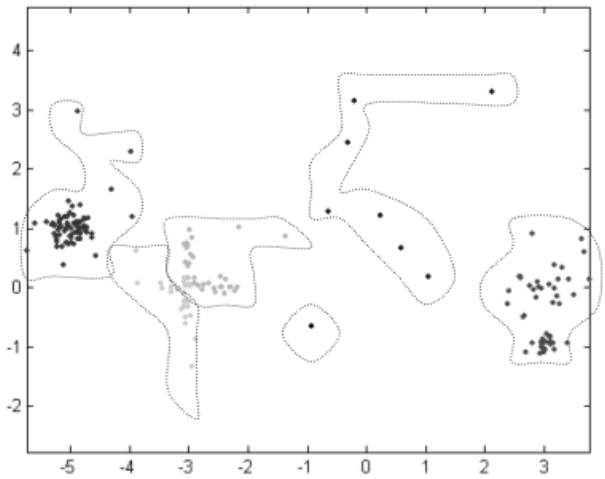
[59] Jiawei Han, Micheline Kamber, Jian Pei, **Data Mining Concepts and Techniques**, 2012, 452



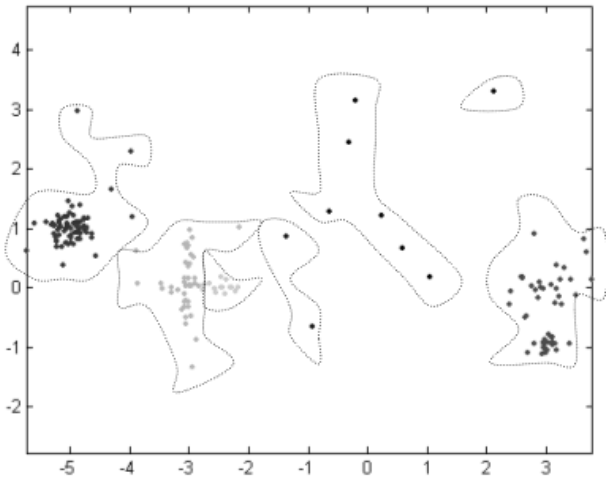
Şekil 15 : K=4 için K-Means Örnek Kümeleri



Şekil 16 : K=5 için K-Means Örnek Kümeleri



Şekil 17 : K=6 için K-Means Örnek Kümeleri



Şekil 18 : K=7 için K-Means Örnek Kümeleri [60]

K-Means algoritmasının en büyük problemlerinden birisi de k değerini kendi hesaplamayıp, kullanıcıdan istemesidir. Uygun sonuç kümelerini bulmak için farklı *k* değerleri için algoritma birden fazla kere gerçeklenmelidir. Kullanıcı girdi verilerinin alan bilgisine hakim değilse bu deneme sayısı arttırılabilir[61].

6.2.2.1.2. Pam Algoritması

Pam algoritması, *Partitioning Around Medoids(Medoidler etrafında bölümleme)* kelimelerinin kısaltılmış şeklidir. Literatürde bu algoritma k-medoids olarak da bilinmektedir.

K-medoids ya da pam algoritması, verilen veri kümesinin elemanlarını, k adet kümeye ayırmak için küme içinden seçilmiş elemanlar merkezinde toplamaktadır. K-means'e çok benzeyen pam algoritmasının, farklı olarak kümenin merkezinde gerçek bir eleman bulunmaktadır.

K-means algoritması aykırı değerlere karşı hassastır. Çünkü aykırı elamanlar, çoğunluktan uzakta yer alarak, ortalamanın dolayısı ile kümenin dramatik bir şekilde bozulmasına neden olmaktadır. Pam algoritmasında, k-means algoritmasındaki bu sorunun ortadan kaldırılması için, kümenin referans noktası olarak kümedeki elemanlarının ortalamalarını almak yerine, kümeyi temsil etmek için gerçek elamanlar seçilmekte ve buna "medoid" denilmektedir.

---

[60] Turgay Tugay Bilgin, Yılmaz Çamurcu, **DBSCAN, OPTICS ve K-Means Kümeleme Algoritmalarının Uygulamalı Karşılaştırılması**, Politeknik Dergisi, Cilt: 8 Sayı: 2, 2005, 140-141

[61] age, 140

Medoid'in formülü ise aşağıdaki 5 nolu eşitlikteki gibidir.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - o_i| \quad (5)$$

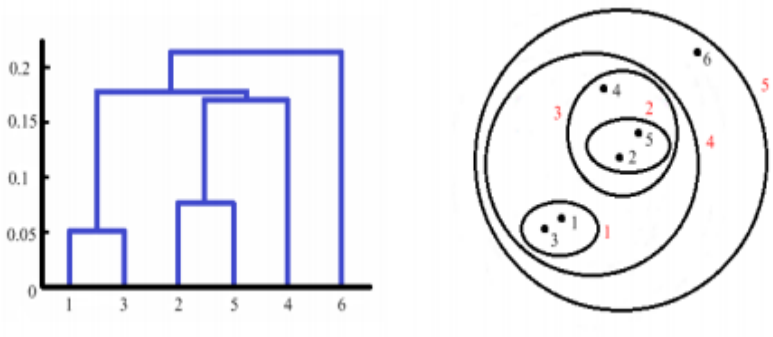
Buradaki p, kümedeki elemanları temsil etmektedir. o_i ise rastgele seçilmiş medoiddir[62].

6.2.2.2. Hiyerarşik Yöntemler

Hiyerarşik kümeleme tekniklerinde noktaların birbirleriyle olan yakınlık ilişkilerine göre ortaya çıkan kümelerden bir hiyerarşik yapı inşa edilir. Bu hiyerarşik yapıyı bir ağaca benzetebiliriz. Hiyerarşik yapının içinde bulunması gereken özellikleri aşağıdaki şekilde sıralamak mümkündür

1. Hiyerarşik bir yöntem kullanıyorsak; birkaç kümeye ayrıştırabilen bir veritabanına sahip olduğumuz anlaşılabilir,
2. Veri tabanını birkaç kümeye ayrıştırmamızda bize yardımcı olan dendogram adı verilen ağaçsı bir yapı bulunmaktadır,
3. Dendogram yapısını yapraklardan gövdeye doğru oluşturabildiğimiz gibi gövdeden yapraklara doğru da oluşturabiliriz bununla beraber oluşturduğumuz bu yapıyı istenilen yerden bölerek kümeleri elde etmek mümkündür.

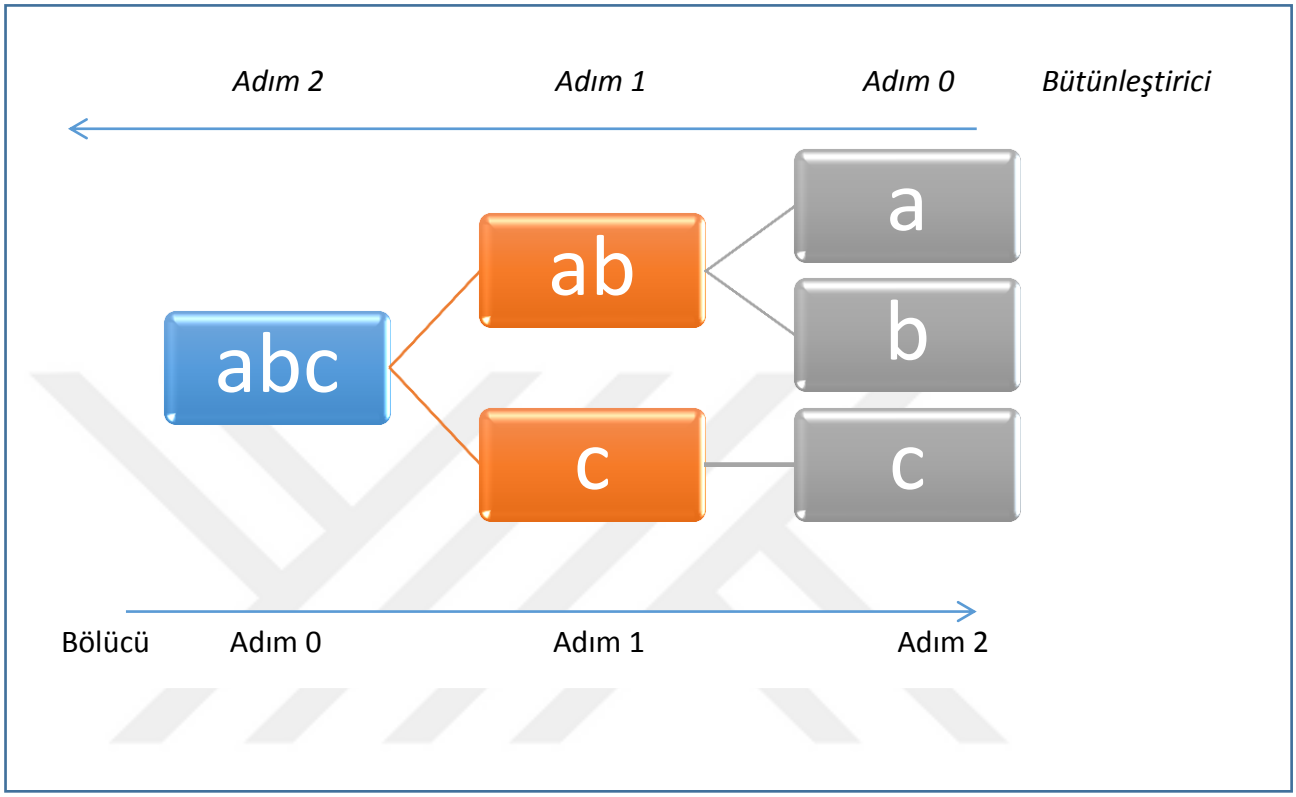
Örnek bir dendogram yapısını aşağıdaki Şekil 19'da gösterebiliriz;



Şekil 19: Dendogram Yapısı

[62] age, 455

Küme ağacında sınıflanan veri nesneleri hiyerarşik küme yöntemlerinde bir ayrışma sırasında yukarıdan aşağıya ya da aşağıdan yukarıya doğru bir bölümlemeye tabi tutmak mümkündür. Bu bölümleme tekniğine bütünleştirici ve bölücü kümeleme analizi denebilmektedir[63].



Şekil 20 : Bütünleştirici ve Bölücü Kümeler

Yukarıda görülen hiyerarşik kümelemedeki bütünleştirici yaklaşıma göre her bir nesne için farklı grup oluşturarak işleme başlanmaktadır. Merkezler arası uzaklık ya da merkezler arası ortalamaya göre gruplar birleştirilir. Seviyenin sonuna gelene kadar işleme devam edildikten sonra artık nesnelerin hepsi tek bir küme oluşturana kadar işleme devam edilir.

Bütünleştirici yaklaşımın tam tersi ise bölücü yaklaşımdır. Aynı kümedeki bütün nesnelerle işleme başlandıktan sonra bu kümenin içindeki nesneleri bir kümeye daha bölmememiz gerekmektedir. Her nesne ayrı bir küme oluşturana kadar bölme işlemine devam edilmektedir[64].

[63] Halil Çağdaş Darakçı, **Kümeleme Analizi Kullanılarak Benzin istasyonlarının Operasyonel Değerlendirilmesi**, 2014, 35-36
[64] age, 37

6.2.2.2.1. Birch Algoritması

Kümeleme analizlerinin denetimsiz yapılarından dolayı dinamik çalışmalarda ortaya çıkan eksiklikler olabilmektedir. Çoğu çalışmaların yoğunlukları eşit değildir bununla beraber daha öncede belirttiğimiz gibi büyük hacimli olan veri tabanlarında uygun sonuçlar üretilemeyebilmektedir.

Büyük veri tabanlarında asıl amaç parçalara bölünerek sonuca ulaşılmaya çalışmaktır. Buradaki en önemli sorun en uygun parçalanmanın nasıl gerçekleşebileceğini bulmaktır. Büyük veritabanlarında ayrımın yapılırken yatay mı yoksa dikey olarak mı ayrımın yapılacağı belirlenmesi gereken en önemli konudur. Birch algoritması ise tüm bu sorunları ve işlem yoğunluğunu ortadan kaldırmaktadır.

Veri tabanlarının değişik şekillerde ayrılmasının sonucunda ortak verilerin paylaşılması durumunda elde edilen sonuçların güvenilirlikleri düşmektedir. Bu gibi durumlarda veritabanı ayrımının nasıl olacağını Birch algoritması kullanarak bir kümeleme analizinin gerçekleştirilmesi uygun olacaktır.

Birch Algoritmasında yer alan kavramları aşağıdaki gibi sıralayabiliriz;

CF: Kümeleme özelliği; Veriler hakkındaki üç özelliğin birleşmesiyle oluşmaktadır.(N, LS, SS)

N: Toplam Veri Sayısı,

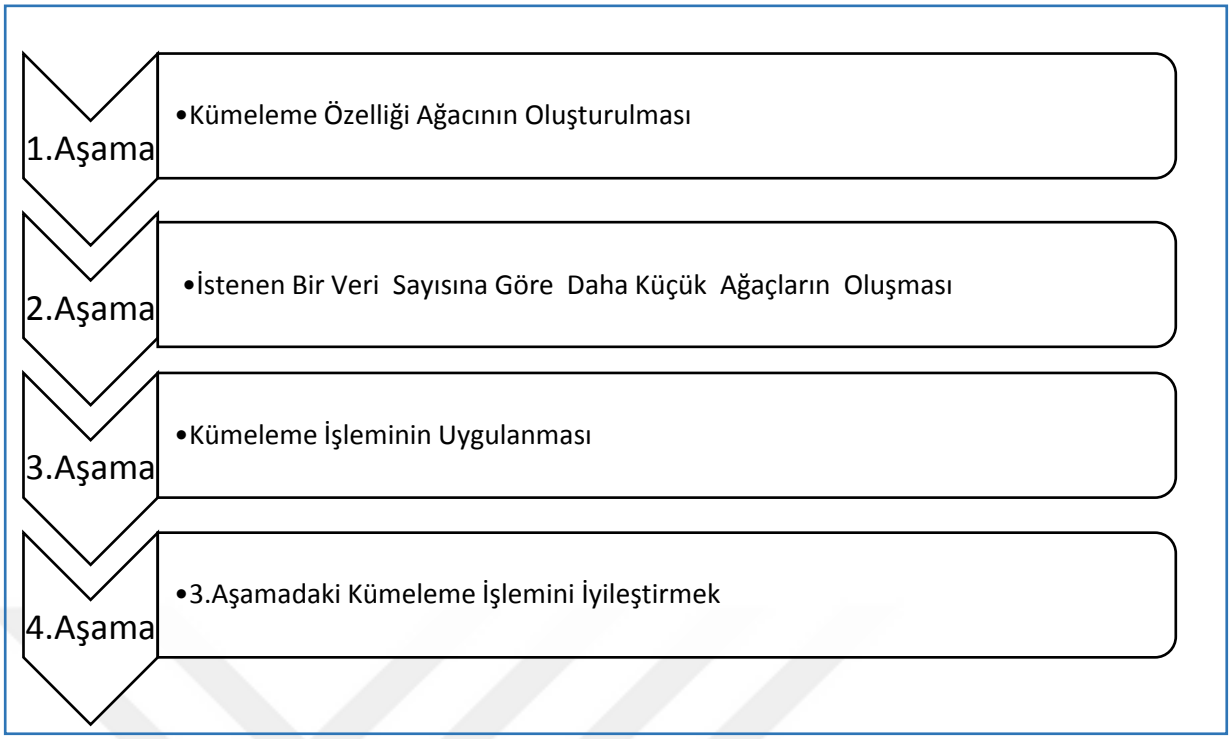
LS: Verilerin Doğrusal Toplamı,

SS: Verilerin Hata Karelar Toplamı,

CF Tree: Kümeleme Özelliği Ağacı; Alt Kümelerde toplanan bilgilerin birleştiği ağaçtır[65].

Birch Algoritmasının aşamaları ise Şekil 21 de gösterildiği gibidir:

[65]Selim Çam, **Veri Madenciliğinde Kümeleme Analizi ve Sağlık Sektöründe Bir Uygulama**, Cumhuriyet Üniversitesi, Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, 2014, 69-70



Şekil 21: BIRCH Algoritması

6.2.2.2.2. Chameleon Algoritması

Karypis, Han, Kumar tarafından geliştirilmiş bir kümeleme analizi algoritmasıdır. 1999 yılında Hiyerarşik Kümeleme tekniklerini kullanan dinamik modelleme yapısıyla oluşturulmuştur[66].

Belirlenen iki kümenin birbirleriyle olan benzerlikleri kendilerinin iç benzerlikleri ve yakınlıklarıyla karşılaştırılmaktadır. Böylece gerekli koşulları sağladıklarında bu iki alt küme birleştirilebilir. Bu şekilde homojen bir yapıda sağlanmış olmaktadır.

Chamelon Algoritmasının üzerinde çalıştığı şebeke diyagramındaki düğümler her bir veriyi temsil etmektedirler bu veriler arasındaki benzerlik ise kenarlarla belirtilir.

Chamelon ölçütü, mesafe yerine benzerliktir. Öncelikle diyagram alt kümelere ayrılmakta, ardından hiyerarşik bir algoritma ile birleştirilmektedir.

Cj ve Ci kümeleri arasındaki bağlantısallık ölçümleri 6 nolu eşitlikten bulunabilir.

[66] Sinan Çetinkaya, Melih Başaraner, **Kentsel Alanlarda Lokal Bina Örüntülerinin Elde Edilmesi: Karşılaştırmalı Bir Çalışma,** TMMOB Coğrafi Bilgi Sistemleri Kongresi, Antalya, 2011, 3

$$RI(C_i, C_j) = \frac{2\ x\ |EC_{(Ci,Cj)}|}{|EC_{Ci}| + |EC_{Cj}|} \quad (6)$$

RI (Relative Inter-Connectivity) : Göreceli Bağlantısallık

RC (Relative Closeness) : Göreceli Yakınlık

$\boldsymbol{EC_{(Ci,Cj)}}$: Ci ve Cj arasındaki mutlak bağlantısallık

Cj ve Ci kümeleri arasındaki yakınlık ölçümleri 7 nolu eşitlikten aşağıdaki bulunabilir:

$$RC(C_i, C_j) = \frac{\bar{S}_{EC_{(C_i,C_j)}}}{\frac{|C_i|}{|C_i + C_{ij}|}\ \bar{S}_{EC_{(C_i)}} + \frac{|C_j|}{|C_j + C_{ij}|}\ \bar{S}_{EC_{(C_j)}}} \quad (7)$$

$\bar{S}_{EC_{(C_i,C_j)}}$: $C_i\ ve\ C_j$ kümelerinin kenarlarının ortalama ağırlığıdır.

Toplaşım algoritmaların özelliklerini taşıyarak kümeleri birleştirir ve yakınlık ile bağlantısallığı kullanarak ortaya çıkabilecek küme kalitesizliklerini de önler[67].

6.2.2.2.3. Cure Algoritması

Veri topluluğu içinde diğer verilerden çok daha uzakta olup sayıları az olsa da kümelerin kalitesini önemli ölçüde etkileyen uç verilerden yola çıkılarak 1998 yılında geliştirilen bu modelin amacı; uç verilerin kümelemenin kalitesini bozmaması için oluşturulmuş bir algoritma olmasıdır.

Cure algoritması girilen her adayı bir kümeymiş gibi düşünmekte, bunların birbirleriyle olan yakınlıklarını hesaba katarak birleştirme ya da ayrıma işlemini gerçekleştirmektedir. Kümelerin her biri için temsil gücü yüksek olan c adet temsilci seçilir. Bu temsilciler kümelerini iyi temsil etmekte ve her biri ***a*** katsayısıyla çarpılarak kümelerin merkezine konumlandırılmaktadır. İki küme arasındaki uzaklık birbirine en yakın temsilciler arasındaki uzaklık kadardır. Kümelerin a katsayısıyla

[67] Gökhan Özkan, **Veri Ambarlama ve Veri Madenciliği "Chameleon Algoritması"**, Gazi Üniversitesi Bilişim Enstitüsü, https://groups.google.com/forum/#!msg/veri_madenciligi/-5y0NfGR8WU/8SXACBgvy24J, 21.05.2016, 3-8

çarpılmaları kümeleri uç verilerin etkisinden de korumaktadır. Aynı zamanda bu a katsayısı kümelerin şekillerini de belirlemektedir. Çarpılan *a* değeri 0 ile 1 değeri arasında yer alımaktadır.

Algoritmanın az yer kaplaması için ana kütleden alınan ufak bir örneklem üzerinde de uygulanan Cure algoritması rastgele yapılan bir örneklemde kümelerin kalitesini artırmaktadır[68].

6.2.2.2.4. Slink Algoritması

En yakın komşu tekniğini kullanan Slink algoritmasında kümeler arasındaki mesafe hesaplanmaktadır. Kümeler arası mesafeyi ise kümelerin birbirlerine en yakın olan elemanlarının birbirlerine olan uzaklıkları olarak belirtmektedir.

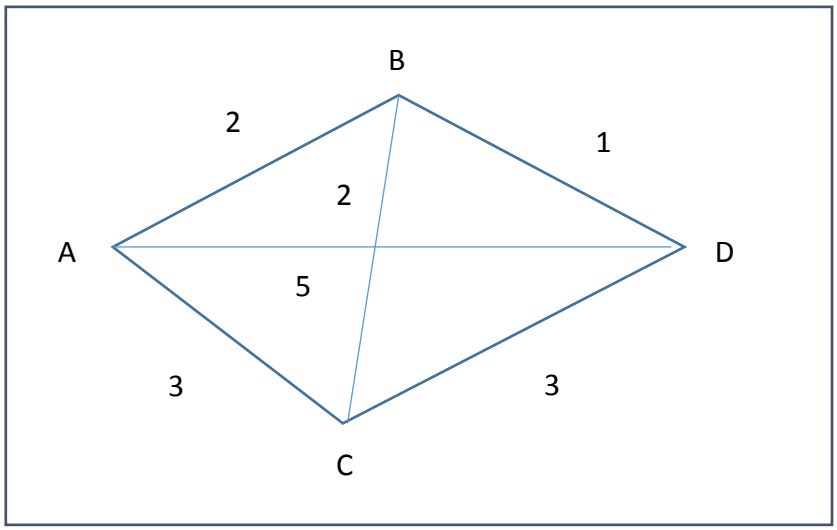
Slink Algoritmasının aşamaları öncelikle mesafe ve benzerlik matrisi çıkarılarak bulunur. Bu matrislerden en küçük maliyetli bir ağaç çıkarılır ve kümeler oluşturulur. Tablo 7 ve Şekil 22 ile bu algoritmayı aşağıdaki gibi açıklayabiliriz.

Tablo 7 : Mesafe Ölçüsü

	A	**B**	**C**	**D**
A	0	2	3	5
B	2	0	2	1
C	3	2	0	3
D	5	1	3	0

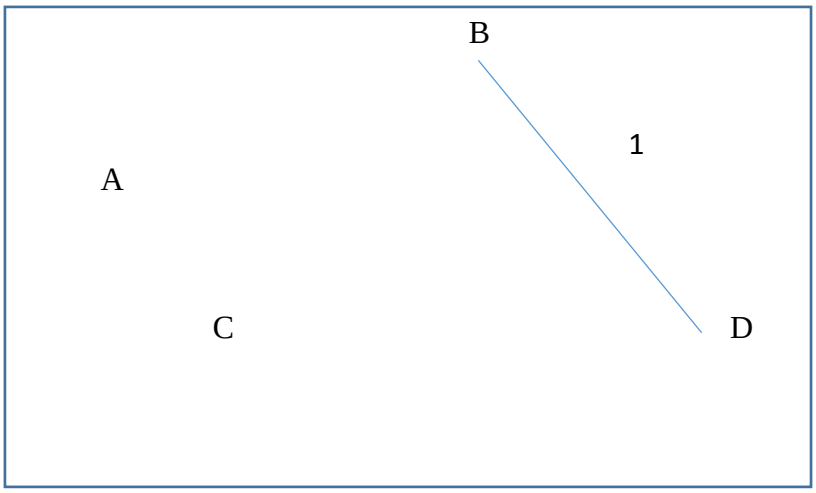
Yukarıda Tablo 7 ile belirtilen mesafe tablosuna göre çizilebilecek şebeke diyagramı ise aşağıdaki olmaktadır;

[68] Silahtaroğlu,109-110



Şekil 22 : Şebeke Diyagramı

Diğer yandan Şekil 23'de ise aşağıdaki gibi eşik değeri 1 iken bu değerden büyük olan bağlarla ilişki koparılmış kümelerin var olduğunu görülmektedir. B, C, D noktaları Şekil 23'de her biri bir küme olarak ele alınmış daha sonra mesafeleri birbirine yakın olan değeler birleştirilerek ayrı kümelerin ortaya çıktığı görülmektedir. Aşağıdaki Şekil 23 den de görülebileceği üzere, 3 ana küme ortaya çıkmıştır. Bunlar; A kümesi, C kümesi ve BD kümesidir[69].



Şekil 23 : Eşik Değeri 1

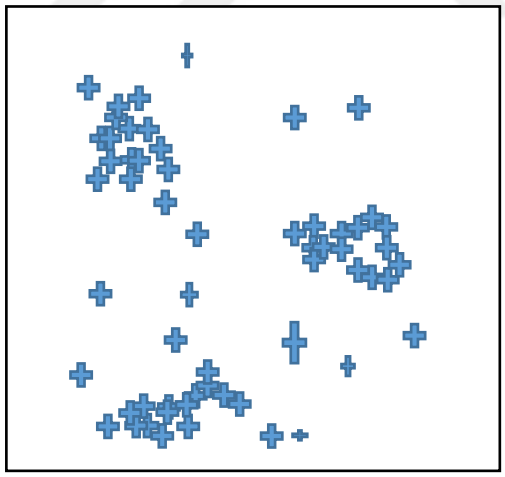
[69] Silahtaroğlu, 107-108

6.2.2.3. Yoğunluk Temelli Yöntemler

Şekilleri çok farklı olan kümeler için yoğunluk temeline dayalı algoritmalar kullanılmaktadır. Yoğunluk temeline dayalı yöntemlerde dağınık olan noktalar arasındaki Euclid mesafesine dayanarak bir kümeleme yapmak kolay olmamaktadır. Bu tür çok dağınık kümelerde uç noktaların (outliers) küme dışına çıkarılması gerekmektedir. Gerçek kümelerin ortaya çıkmasında zorluk yaratacak bunun gibi uç veriler hesaplamalarda hatalara neden olmaktadır[70].

Ancak söz konusu dağınık kümelerde yoğunluğa dayalı olarak kümeleme işlemi gerçekleştirilebilir. Şekil 24'de görüldüğü gibi bir arada oldukça sık ve göze çarpan yoğunlukta bulunan noktalar bir küme olarak algılanmalıdır. Yoğunluklara dikkatlice bakıldığında bazı noktalar bu yoğunluğun sınırlarını belirlerken bazıları ise iç kısmındaki yoğunluğu göz önüne sermektedir[71].

Yoğunluğa dayalı örnek bir veri topluluğunu, aşağıda yer alan Şekil 24'den görmek mümkündür.



Şekil 24 : Örnek Veri Topluluğu

[70] Çamurcu, 140
[71] Silahtaroğlu, 121-122

6.2.2.3.1. Dbscan Algoritması

Martin Ester ve arkadaşları tarafından bulunan Dbscan Algoritması; iki ya da daha çok boyuta sahip veri noktalarının birbirleriyle olan uzaklıklarını ortaya çıkarmaya yaramaktadır. Bu veritabanı, daha çok kümeleme analizine uzaysal bir bakış açısı kazandırmış olup bundan dolayı da genellikle uzaysal verilerin analizinde sıklıkla kullanılmaktadır[72].

Dbscan algoritmasında karşılaşılabilecek başlıca temel terimler aşağıdaki gibidir;

***Eps*:** Kümede var olan iki nokta arasındaki en uzun mesafeyi,

$$N_{Eps}(p) = \{q \in D \mid mes(p,q) \leq Eps\} \quad (8)$$

***MinPts*:** Belirlenen bir kümede olması gereken en az nokta sayısını,

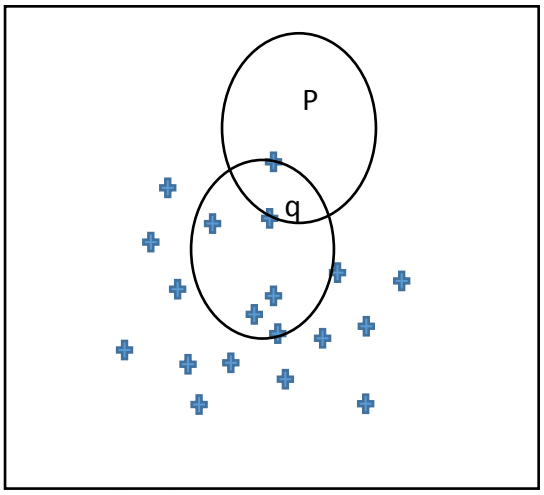
***Yoğunluğa Doğrudan Erişilebilirlik*:** Aşağıda belirtilen 9 ve 10 nolu eşitliklerde p ve q noktaları birbirleri sayesinde yoğunluğa ulaşabilmektedirler.

$$p \in N_{Eps}(q) \quad (9)$$

$$\left|N_{Eps}(q)\right| \geq MinPts \quad (10)$$

9 ve 10 nolu eşitliklere dayanarak p ve q noktaları yoğunluğun içerisinde ise erişilebilirlik simetrik olmaktadır. Seçilen q ve p noktalarından herhangi biri yoğunluğun sınırında bulunuyor ise yoğunluğa erişilebilirlik simetrik değildir. Eğer p noktası sınırda q noktası içerdeyse Şekildeki gibi p noktası, q noktası sayesinde doğrudan yoğunluğa erişmiş olmaktadır.

[72] Turgay Tugay Bilgin, Yılmaz Çamurcu, **DBSCAN, OPTICS ve K-Means Kümeleme Algoritmalarının Uygulamalı Karşılaştırılması**, Politeknik Dergisi, 2005, 141



Şekil 25 : DBSCAN Algoritmasında p ve q Noktaları

Yoğunluğa Erişebilirlik: Eğer $p_{1,}p_2,..p_n$ noktalarından oluşan bir örüntü var ise $p_{1,} = q\ ve\ p_n = p$ öyle ki $p_{i+1}\ p_i$ aracılığıyla yoğunluğa doğrudan ulaşılabilir[73].

Yoğunluk Bağıntısallığı: p ve q noktaları yoğunsallığa herhangi bir üçüncü nokta ile bağlanabilmektedir.

***Küme*:** C noktalardan oluşan bir veritabanı olduğunu varsayalım Eps ve MinPts de olması gereken gereklilikleri sağlayan bir N kümesi C'nin bir alt kümesi olmaktadır.

Gürültü: C veritabanında herhangi bir N kümesine ait olmayan bir noktadır.

Başlangıç aşamasında rastgele bir p noktası seçildikten sonra p noktası üzerinden MinPts ve Eps koşullarını sağlayan tüm noktalar toplanır ve zaten p noktası bir iç nokta ise küme oluşturulmuş bulunmaktadır. Eğer p bir dış nokta ise ve hiçbir şekilde yoğunluğa erişilemiyor ise başka bir p noktası denenmelidir[74].

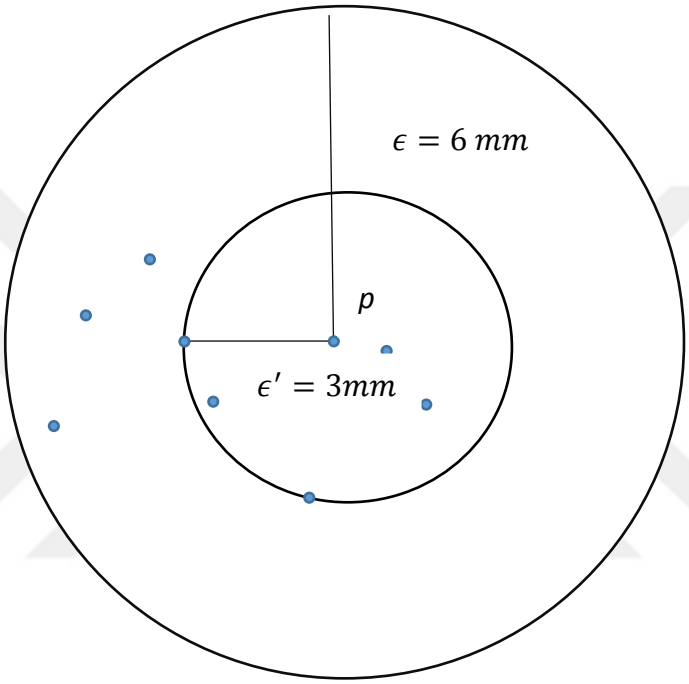
6.2.2.3.2. Optics Algoritması

DBSCAN algoritmasında kullanılan iki önemli değer olan MinPts ve Eps değerleri kümeleme analizini gerçekleştirmek için kullanıcı tarafından girilmesi gereken iki önemli parametredir. Dışardan gelen işlemlere bağlı olan bu algoritmada istenen küme sayısının belirlenmesi, Euclid mesafesinin araştırmacı tarafından daha önceden

[73] Silahtaroğlu, 123
[74] age, 124

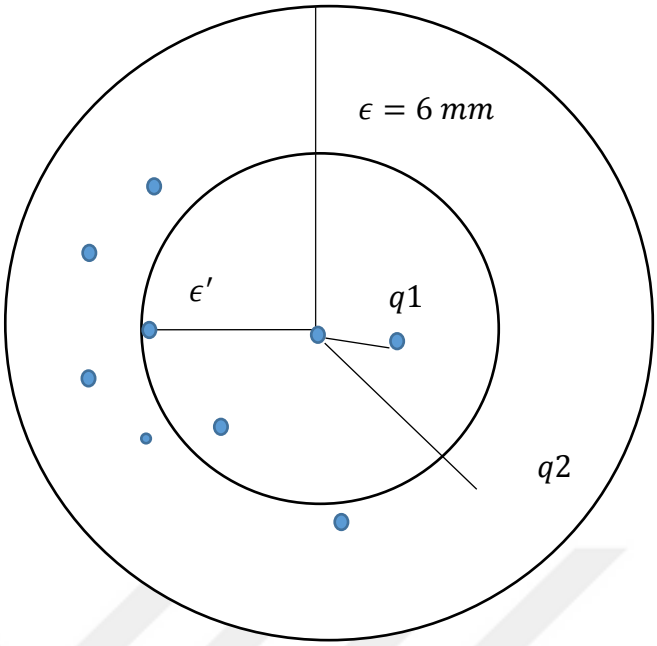
bilinerek girilmesini gerektirmektedir. Parametrelerin bu durumu küme yapısını ve sayısını önemli ölçüde etkilemekte idi. Bunun sonucunda DBSCAN'i geliştiren araştırmacılar bu olumsuzluğu ortadan kaldırabilmek amacıyla Optics algoritmasını tasarlamışlardır[75].

Bu algoritmayı tasarlarken iki ayrı parametre kullanılmıştır. Bu parametreleri gösteren Şekil 26 ve Şekil 27'i aşağıda görmek mümkündür.



Şekil 26 : P'nin İç Mesafesi

[75] Jiavei Han, Micheline Kamber, **Data Mining: Concept and Techniques**, 2006, 420



Şekil 27 : Ulaşılabilirlik Mesafesi (*p, q1*) = $\epsilon' = 3mm$

Ulaşılabilirlik Mesafesi (*p, q2*) = (*p, q2*)

Şekillerde de adı geçen algoritma için gerekli olan parametreler; Çekirdek uzaklığı ve ulaşılabilirlik uzaklığıdır.

Çekirdek uzaklığı (iç mesafe) ve ulaşılabilirlik mesafesi optics algoritması için her nokta üzerinden hesaplanmaktadır. Veritabanını belli bir sıra ve düzen içerisinde bulunduran algoritma zaman karmaşıklığı düşünüldüğünde DBSCAN içe aynı olduğu söylenebilir[76].

6.2.2.3.3. Denclue Algoritması

Kümeleme analizi yoğunluk tabanlı çalışan formüllerden oluşmaktadır. Declue algoritmasının iki aşaması vardır. Bu aşamaları aşağıdaki gibi sıralaybiliriz;

1. **Ön kümeleme;** Belirlenen veri setinin ilgi alanları üzerinde durulur ve bir harita oluşturulur. Yoğunluk fonksiyon hesaplamalarının hızlanması için veri setinin alanını bu haritalarla ulaşabiliriz.
2. **Gerçek Kümeleme**; Yoğunluk oluşturucu ve yoğunluğa göre hareket eden noktaları belirlemektedir.

[76] age, 421

Denclue algoritmasında ilk göze çarpan her veri noktasının matematiksel bir şekilde gösterildiği etki noktaları mevcuttur. Etki noktaları sayesinde çevre üzerindeki etki görülebilmektedir. Aynı zamanda aşağıdaki eşitlik sağlanmış olmalıdır.

Veri Alanının Toplam Yoğunluğu = Tüm Veri Noktalarının Etki Fonksiyonları Toplamı

Denclue Algoritmasını daha önceki algoritmalardan ayıran temel özellikleri şöyle sıralayabiliriz; Diğer algoritmalara kıyasla daha çok matematiksel fonksiyonlara sahiptir. Veri setleri içerisinde yüksek derecede kirlilik olduğunda bile oldukça başarılıdır. Çok boyutlu veri setlerinde sabit şebekeli olmayan kümeleri tanımlamak için matematiksel tanımlamalar kullanmaktadır. Fonksiyonel bilgileri grid hücrelerinde barındıran DBSCAN algoritmasından çok daha hızlıdır.

6.2.2.4. Grid Temelli Yöntemler

Grid Veritabanı bazı kaynaklarda "Izgara Temelli Yöntemler" olarak da kullanılmaktadır. Grid temelli yöntemlerin kullanıldığı veritabanları oldukça büyük veri setleri içerisinde kullanılmaktadır.

Şekilleri dörtgensel numaralandırılmış hücrelere ayrılmış ızgara şeklini anımsatan bir kümeleme yöntemidir[77].

Algoritmalar hazırlanırken belirlenen her input(girdi) için bir değer verilir ve modelleme yapılır. Bu değerlerin toplamı ya da bu değerlerden oluşması beklenen örneklem sonucunda Grid merkezli algoritmanın sonuçlanması beklenmektedir[78].

Bölünmede kullanılan bazı geometrik şekiller kümeleme analizi için sorun oluşturabilmektedir, fakat ızgara temelli kümeleme analizi hücreleri dörtgensel şeklinde bölümlendirdiği için bu sorun ortadan kalkmaktadır. Dörtgensel şekilde oluşturulan bu kutular birleştirildiği zaman ortaya çıkan aşamada kümeleme analizinin gerçekleştiği takip edilebilmektedir[79]. Grid Temelli yöntemlerde sıklıkla kullanılan algoritmalar ise Sting ve Clique algoritmalarıdır.

Grid temelli yöntemlerin başlıca özelliklerini özetleyecek olursak;

---

[77] Iivari Kunttu, Ari Visa, **Classification Method for Defect Images Based on Association and Clustering**, Data Mining and Knowledge Discovery USA, 2003, 22-24

[78] Ömer Uçan, **Dijital Kütüphanelerde Veri Madenciliği Uygulamaları**: Akdeniz Üniversitesi Merkez Kütüphanesi Örneği, 2010, 43

[79] age, 43

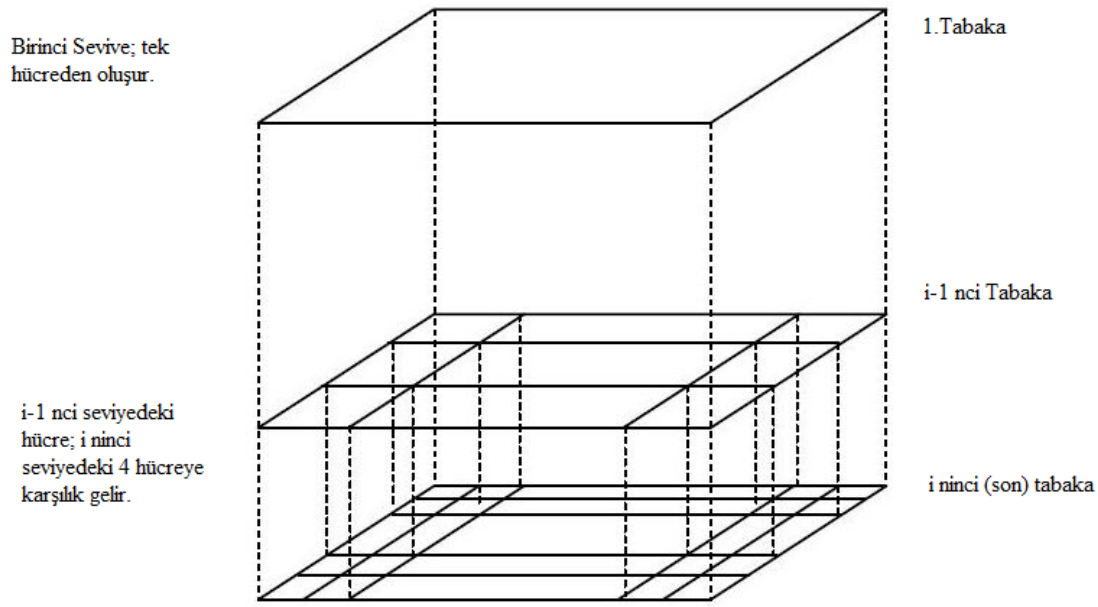
Büyük veri setlerinin işlenmesine kolaylık sağlaması için Grid kümelemesinin o veritabanını bir kez taraması yeterli olmaktadır.

Kullanılan dörtgensel şekiller göz önüne alındığında çok farklı geometrik şekillerin oluşacağı kümeleme olanakları sağlanmaktadır.

Yoğunluğu göz önüne alındığında Grid temelli yöntemde geometrik şekil dahilinde olmayan kümelenmelerde de nokta yoğunluğa bakılarak sınırlar belirlenebilmektedir[80].

6.2.2.4.1. Sting Algoritması

Hiyerarşik bir yapı oluşturan Sting algoritması öncelikle incelenecek bölgenin dikdörtgen hücrelere bölünmesiyle başlamaktadır. Öncelikle hiyerarşik yapıda oluşturulmuş görseli Şekil 28'den incelemek mümkündür.



Şekil 28 : Hiyerarşik Yapı

Şekil 28'de görülen hiyerarşinin temelini Birinci Seviye olarak adlandırdığımızda, bu temel seviyeye bağlı olan alt seviyelere "oğullar" adı verilmektedir. Oğullar ikinci ve üçüncü şekilde sıralanmaktadır. (i+1) nci seviyelerin alanlarının birleşmesinden i

[80] age, 44

hücresi meydana gelmektedir. Her hücre yapraklar hariç 4 tane oğuldan meydana gelmektedir. Şekil 28'den de görüldüğü gibi birinci seviyenin oluşturduğu tek hücre tüm alanı oluşturmaktadır. Hücrelerin şekilsel boyutunun oluşmasında nokta yoğunluğu temel bir etken olup bu noktalar düzinelerce ya da binlerce olabilmektedir. Bu konunun açıklandığı temel makalede Wang ve ark tarafından Stingin açıklandığı kısımlarda dört oğuldan bahsedilmiş ve kullanılmıştır. Oğul sayısı artırılıp azaltılabildiği gibi temel makale göz önüne alındığında genellikle dört oğul kullanılmaktadır[81].

Hücrelerdeki istatistiki veriler aşağıdaki gibi oluşturulmuştur;

n- var olan nokta sayısı,

m- hücrede bulunan tüm sayısal verilerin ortalaması,

s- hücrede bulunan tüm sayısal değerlerin standart sapması,

min- en küçük,

maks- en büyük değerler dağılım tipi,

dağılım- hücre değerlerinin dağılım tipleri[82]

ni göstermektedir.

Alt seviye hücreler için yukarıdaki gibi m, n, s, min, maks gibi parametreler otomatik hesaplanmaktadır. Sonrasında dağılım şeklinin bilinmesi halinde sisteme giriş yapılmakta, bilinmemesi durumunda ise ki-kare testi yardımıyla dağılım şekli belirlenmektedir. Alt seviye hücrelerin parametreleri belirlendiği için üst seviye hücreler bu parametreler yardımıyla belirlenmektedir. Üst seviye hücrelerin parametreleri alt hücreler doğrultusunda belirlendiği gibi dağılımları da alt seviye hücrelerin çoğunluğunun kullandığı dağılım yoluyla belirlenir. Alt seviyelerde dağılımın belirlenmesinde sorunlar yaşanmış ise ve uyuşmazlık devam ediyor ise üst seviye dağılımı hiçbiri olarak belirlenir[83].

Yukarıda belirtilen bu istatistiki bilgilerin kullanımları ise şu şekilde belirlenmektedir. Öncelikle oluşturulan hiyerarşide bir katman seçilerek gerekli sorgulamaya başlanır. Seçilen bu katmanın içerisinde çoğunlukla az sayıda hücreler

[81] Wei Wang, Jiong Yang, Richard Munthz, **A Statistical Information Grid Approach To Spatial Data Mining**, University of California, 1997, 5
[82] age, 5
[83] Silahtaroğlu, 130

yer almaktadır. Seçilen katmanın belirlenen sorgulamanın yüzde kaçını karşılayabileceğini ya da bu sorgulamada ne kadarlık kısmını karşılayamayacağının belirlenmesiyle bir güven aralığı çıkartılır. Güven aralığı belirlendikten sonra her bir hücrenin artık bu güven aralığında uygun ya da uygun olmadığını belirtilir ve uygun olmayan olarak belirlenen bu hücreler sistem dışına çıkarılarak işleme uygun olan hücrelerle devam edilir. Bu işleme, belirlenen en alt seviyeye ulaşana kadar yapmaya devam edilir.

Tüm bu işlemler bittikten sonra artık elimizde uygunluğu kanıtlanmış olan hücrelerle daha önceden belirlenmiş yoğunluk kriterlerine göre bölgelerde seçimler yapılır. Tespit edilen ve yoğunluğa uygun olan alanlar tespit edilmiş olur. Sting algoritması bahsedilen bu işlemi en alt seviyedeki hücre sayısı kadar zamanda tamamlayabilmektedir[84].

6.2.2.4.2. Clique Algoritması

İyi bir kümeleme analizi gerçekleştiren Clique algoritması bu başarasını yüksek yoğunluklu kümelemelerde alt uzaylara inebilmesi sayesinde gerçekleştirebilmektedir. İnputların göz önüne serildiği bu algoritmada herhangi bir dağılım ya da sıralama önemli olmadığı gibi ortaya koyduğu sonuçlar ise özdeş olarak görülmektedir.

Clique genel dağılım modellerini ortaya koyarken algoritmadaki seyrek ve yoğun bölgeleri belirlemektedir. Bu yoğun bölgeler birbirine yakın bölgelerden oluşmaktadır ve bu bölgelere ‘’ birim’’ adı verilmektedir. Clique kümeleme analizinde belirlenen çok boyutlu alanlar düzenli olarak veri noktaları tarafından belirlenmemiştir[85].

Clique kümeleme analizinin ilk aşamasında, n boyutlu bir veri uzayı birbiriyle bağlantılı olmayan ve çakışmayan dörtgensel hücrelere bölünmektedir. Bu hücrelerde ulaşılması gereken yoğun bölgeler aranır. Kümeler ise bu yoğun birimlerden oluşmaktadır. İki birimin birleşmesi için birbirleriyle bağlantılı olması ya da onlara bağlı bir diğer birimin olması gerekmektedir. Bu birimler birleşerek

[84] age, 131
[85] Ömer Uçan, **Dijital Kütüphanelerde Veri Madenciliği Uygulamaları**: Akdeniz Üniversitesi Merkez Kütüphanesi Örneği, 2010, 43

kümeleri oluşturmaktadır. Son adımda ise artık algoritma kümeyi oluşturan en küçük bölgeyi tanımlamış olmaktadır[86].

6.3. Birliktelik kuralları

Beraber gerçekleşen olayları analiz etmek ve bunlardan anlamlı sonuçlar çıkarmak veri madenciliğinin kapsamına girmektedir. Örneğin bir süper markette süt alan müşterilerin %52 sinin mısır gevreği de aldığını söylemek, birlikte gerçekleşen olaylara örnek olarak gösterilebilir. Sonuç olarak, bu tür bir bilgi market için çok önemlidir. İşletme bu bilgiden yola çıkarak, birlikte satın alınan ürünleri yakın raflara koyarak satışlarını artırabilir. Bu sayede satışlarını artırabilir ve satın alma eğilimlerini değerlendirebilir.

Veri madenciliğinde, birlikte gerçekleşen olayları bulan ve analiz eden yöntemler birliktelik kuralları(association rules) adı altında toplanmıştır[87].

Birliktelik kurallarının kullanım alanlarına örnek olarak;

- Süpermarketteki hangi ürünlerin birlikte alındığı,
- İlaçların yan etkilerine karar vermek,
- Borsada işlem gören hisse senetlerinin birliktelikleri,
- Telekomünikasyon ağlarındaki kesinti ve düşüşleri tahminlemek gösterilebilir[88].

6.3.1. Pazar Sepeti Analizi

Pazar sepet analizlerinde, satılan ürünler arasında ilişkileri ortaya çıkarmak için destek ve güven kavramlarından yararlanılmaktadır. Destek ölçütü, bir ilişkinin tüm alışverişlerdeki tekrarlanma sıklığını ifade eder. Güven ölçütü ise, X ürün grubunu alan müşterilerin Y ürün grubunu da alma olasılığını ifade etmektedir.

$$Destek\ (X \rightarrow Y) = \frac{P(X \cap Y)}{N} = \frac{X\ ve\ Y\ içeren\ işlemler}{Tüm\ işlemler} \qquad (11)$$

[86] Silahtaroğlu, 133
[87] Yalçın, 157
[88] Daniel Larose, **Discovering Knowledge In Data: An Introduction To Data Mining**. New Jersey: A. John Willey&Sons, 2005, 180

$$Güven\ (X \rightarrow Y) = \frac{P(X \cap Y)}{P(X)} = \frac{X\ ve\ Y\ içeren\ işlemler}{X'i\ içerem\ işlemler} \quad (12)$$

Yukarıdaki 11 ve 12 nolu eşitlikte anlatılan destek ve güven ölçütlerini bir örnek ile açıklayacak olursak; bir mağazada 10 müşterinin tek seferde yaptığı alışveriş bilgilerine göre birliktelik kuralının şu şekilde elde edildiğini varsayalım:

$Güven\ (Süt, Mısır\ Gevreği \rightarrow Şeker)$

Burada $X = \{Süt, Mısır\ Gevreği\}$ ürünlerinin yanında $Y = \{Şeker\}$ ürününü de satın alma olasılığını göstermektedir. Bu 3 ürünün birlikte satın alınma sayısını 3 varsayalım. Bu durumda destek ölçütü aşağıdaki gibi hesaplanmaktadır:

$$Destek\ (Süt, Mısır\ Gevreği \rightarrow Şeker) = \frac{Süt, Mısır\ Gevreği, Şeker\ sayısı}{Toplam\ Alışveriş}$$

$$= \frac{3}{10} = 0.3 = \%30$$

Süt ve Mısır gevreğinin birlikte satın alınma sayısı 4 ise, güven ölçütü aşağıdaki gibi elde edilir[89].

$$Güven\ (Süt, Mısır\ Gevreği \rightarrow Şeker) = \frac{Süt, Mısır\ Gevreği, Şeker\ sayısı}{Süt, Mısır\ Gevreği\ Sayısı}$$

$$= \frac{3}{4} = 0.75 = \%75$$

6.3.2. Apriori Algoritması

Apriori algoritması, birliktelik analizlerinin uygulanmasında en çok kullanılan algoritmadır. Bu algoritma Agrawal ve Srikant tarafından 1994 yılında geliştirilmiştir.

Büyük kümeler oluşturan algoritmalar eldeki veriyi birden çok kez taramaktadır. İlk taramada, her bir nesnenin destek ölçütü hesaplanır. Başlangıçta girilmiş olan destek ölçütü ile karşılaştırılır ve tek tek tüm nesnelerin genişliğine bakılır. Bu adımdan sonraki adımlarda ise bir önceki adımda geniş olarak belirlenmiş nesnelerden başlar

[89] Yalçın, 157-158

ve kümeler oluşturulmaya başlanır. Bu kümelere "Aday Nesne Kümesi" adı verilir. Taramanın bitiminde ise hangi aday nesne kümesinin gerçekten geniş olduğuna bakılır. Bir nesne kümesinin geniş olmasına karar vermek için, kullanıcı tarafından verilen minimum destek ölçütüne bakılır, eğer hesaplanan değer verilen değerden büyük ise geniş olarak kabul edilmektedir. Bu işlemlere veri tabanı bitene kadar ve başka yeni geniş nesne kümeleri bulunamayana kadar devam edilmelidir.

Algoritmanın adımları aşağıdaki gibidir:

- Veriler ilk taramada, geniş nesne kümelerinin tespiti için tüm nesneler sayılır.
- Bir sonraki tarama, *k.* tarama olduğunu varsayarsak,
- Apriori-gen fonksiyonu ile (k-1). taramada elde edilen, L_{k-1} nesne kümeleri ile, C_k aday nesne kümeleri oluşturulur.
- Veri tabanı taranarak, C_k daki adayların desteği sayılır.
- Hızlı bir sayım için, verilen bir l işleminde, C_k yı oluşturan adayların çok iyi belirlenmesi gerekmektedir[90].

[90] Agrawal Rakes, Srikant Ramakrishnan, **Fast Algorithms for Mining Assosication Rules,** 20. VLDB Konferansı, Şili, 1994, 487-499

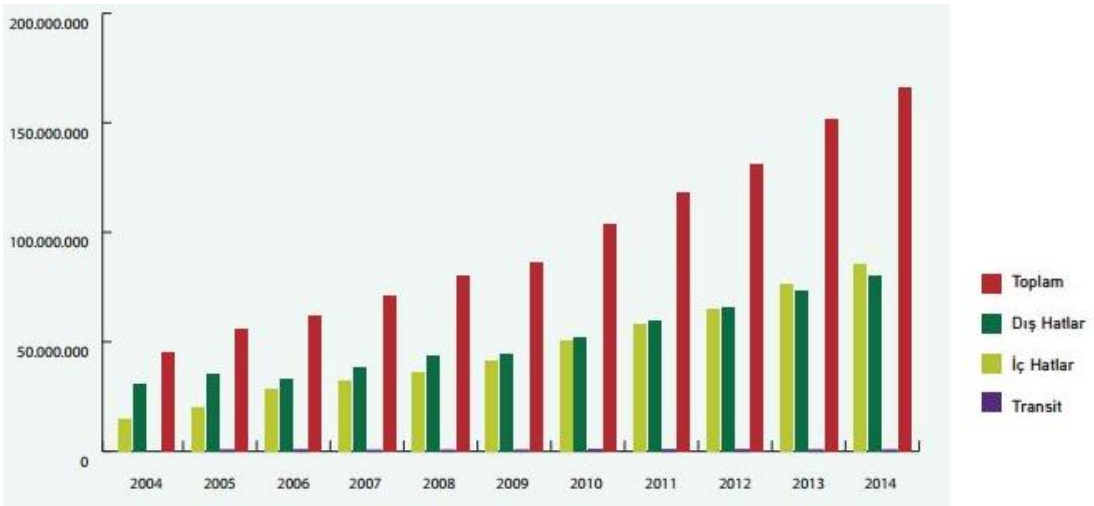
7. HAVACILIK SEKTÖRÜNDE BİR UYGULAMA

Havacılık sektörü, Türkiye'de ve dünyada büyümeye devam eden sektörlerden bir tanesidir. Amacı ulaştırma olduğu için, mesafeleri kısaltmakta, aynı zamanda bilginin de ulaşımını sağlamaktadır.

Tablo 8'de, 2004 ile 2014 arasında, Türkiye'de taşınan toplam yolcu sayıları gösterilmiştir. Artan yıllara göre, yolcu sayıları parabolik artış göstermektedir.

Tablo 8 : 2004-2014 Türkiye'de Havayolu Yolcu Trafiği[91]

Yıllar	İç Hat	Dış Hat	Transit	Toplam
2004	14.460.864	30.596.507		45.057.371
2005	20.529.469	35.042.957	547.046	56.119.472
2006	28.774.857	32.880.802	616.217	62.271.876
2007	31.949.341	38.347.191	418.731	70.715.263
2008	35.832.776	43.605.513	449.091	79.887.380
2009	41.226.959	44.281.549	492.835	86.001.343
2010	50.575.426	52.224.966	736.121	103.536.513
2011	58.258.324	59.362.145	671.531	118.292.000
2012	64.721.316	65.630.304	677.896	131.029.516
2013	76.148.526	73.281.895	565.447	149.995.868
2014	85.607.565	80.360.476	523.047	166.491.088



Şekil 29 : 2004-2014 Türkiye'de Havayolu Yolcu Trafiği Grafiği

[91] Sivil Havacılık Genel Müdürlüğü, 2014 Faaliyet Raporu, http://web.shgm.gov.tr/documents/sivilhavacilik/files/pdf/kurumsal/raporlar/2014faaliyetraporuv2.pdf 23.05.2016

Bu denli büyük bir sektörün, işler bir şekilde kalması için birçok süreci başarı ile götürmesi gerekmektedir. Bunun için tedarik, yöneylem, planlama, satış ve lojistik gibi alanlarda başarılı bir ekip çalışması gerekmektedir.

7.1. Araştırmanın Konusu

Veritabanlarında saklanan ve analiz edilmemiş ham veri, veri madenciliği için hammadde olarak şirket veri merkezlerinde beklemektedir. Şirketler içinde veri madenciliğine başlamak için en önemli sebeplerden bir tanesi, kısıtlı olan kaynakları daha verimli kullanma, maliyetleri azaltıp, karı artırmaktır.

Araştırma bir X havayolu şirketinde, kabin içinde yolculara verilen ikramların, uçağa binen yolcu sayısından fazla yüklenmesi durumunda ortaya çıkan maliyetin araştırılması ve minimize edilmesi için yapılmıştır.

Fazla ikram yüklemesi olarak adı geçen bu sorun, ikram yapan yurtiçi veya yurtdışı tüm havayolu şirketlerinde karşılaşılan önemli bir sorundur.

Bir yolcunun havayolu ekosistemine dahil olması, bilet rezervasyon ve satın alma ile başlamaktadır. Bilet satın alma kanalları, genel anlamda ikiye ayrılır;

- Agentalar üzerinden,
- Şirketin kendi kanalları üzerinden, müşteriler satın alma işlemini gerçekleştirebilir.

Şirketin kendi kanalları ise internette bulunan web sayfası, akıllı telefon uygulamaları, çağrı merkezi ve fiziki ofisleri olarak sıralanabilir.

Müşteri bilet alarak şirket ekosistemine giriş yapmış olmaktadır. Henüz biletleme işleminde iken, kişinin ikram bilgileri hazırlanmaktadır. Kişinin özel tercihi ile farklı menüler seçilebilir, vejetaryen menüler istenebilir.

Uçuş günü ve saati yaklaştığında *check-in* servisleri açılmaya başlar. Bu noktada tercih havayoluna bırakılmıştır. Bazı firmalar uçuştan 72 saat önce açmakla beraber, bazı firmalar uçuştan 24 saat önce check-in servisini açmaktadır. Check-in ile yolcu aslında uçağa bineceğini belirtmekte ve bu işlem sırasında koltuğunu seçmektedir. Böylece yolcu listesi ve uçak oturum planı şekillenmiş olmaktadır.

Araştırma konusu, yolcunun bilet satın almasıyla başlayıp uçağın hedef istasyona inip bagajını almasıyla sona eren uçuş hizmetinin bir parçası olan, uçak içi

ikramların uçağa yolcu sayısından fazla yüklenmesinin nedenlerini ve mümkün çözümlerini, maliyeti düşürmek ve karlılığı artırmak için araştırmaktır.

7.2. Araştırmanın Kısıtları

Araştırmanın en önemli kısıtlarından biri, X havayolunun uçuş bilgilerine ulaşmaktır. Uçuş bilgilerinin kapsadığı detaylar; uçak tipi, kalkış ve iniş istasyon bilgileri uçak yolcu kapasitesi, check-in yapmış yolcular, uçağa binen yolcu sayısı ve yüklenen ikram sayısı olarak sıralanabilir.

Bu tarz geçmiş bilgiler şirket için önem arz etmektir. Hem gizlilik hem güvenlik açısından çok önemli olan bilgiler, 3. kişiler ile paylaşılmamak kaydı ile tarafıma paylaşılmıştır.

İkinci kısıt ise, biletini almış, sonrasında check-in yapmış fakat uçağa binmemiş yolcu olma ihtimalidir. Bu kısıtın hem havaalanı işletmesince hem şirket sistemi ile takibi mümkün değildir. Bu durum uçağa biniş (boarding) sonrası sayımlarda, check-in yapmış yolcu sayıları ile kontrolünden sonra ortaya çıkmaktadır. Bu da ikramların, hali hazırda check-in yapmış yolcu sayısı kadar yüklemesi yapıldıktan sonra meydana gelmektedir.

Biletini almış ve check-in yapmış yolcunun uçağa binmeme nedenleri araştırıldığında ise ortaya çıkan sonuçlar aşağıdaki gibidir;

- Keyfi olarak. Yolcunun kendi isteği ile uçağa binmediği durumlardır.
- Polis gücüyle. Yolcunun kendi isteği dışında, polis tarafından alıkonulduğu durumlardır. Yurt dışına çıkacak yolcunun pasaport kontrol noktasından geçememesi, yurt dışındaki hava meydanlarında bulunup Türkiye'ye dönecek yolcunun yabancı polisler tarafından alıkonulması örnek olarak gösterilebilir.

7.3. Araştırmanın Metodolojisi

7.3.1. Araştırmanın Amacı

Araştırmanın amacı, şirket bünyesinde hali hazırda var olan, uçuş bilgilerine göre fazla yükleme yapılan uçuşların tespiti ve bu uçuşları havaalanlarına göre kümeleyerek raporlamaktır.

7.3.2. Araştırmanın Türü

Bu araştırma yöntem açısından nicel bir araştırmadır. Veri madenciliği literatüründe sık kullanılan veri kümeleme yöntemlerinden K-Means algoritması uygulanmıştır.

K-Means kümeleme algoritmasının başlangıç parametresi olan küme sayısı; 3 ve 4 şeklinde belirlenerek iki ayrı analiz yapılmıştır. K değeri 3 olarak seçildiğinde sözkonusu kümeleri aşağıdaki isimlendirebiliriz;

- Güvenli,
- Orta,
- Riskli.

7.3.3. Araştırmanın Ana Kütlesi

Araştırmada ana kütle olarak Türkiye hava taşımacılığı sektörü alınmıştır. Sektörde birden fazla sayıda ve değişik büyüklüklerde havayolu firması bulunmaktır. Fazla ikram sorunu havayolu şirketinden bağımsız bir problem olduğu için ana kütlemiz Türkiye havacılık sektörü olarak alınmıştır.

7.3.4. Araştırmanın Örneklemi

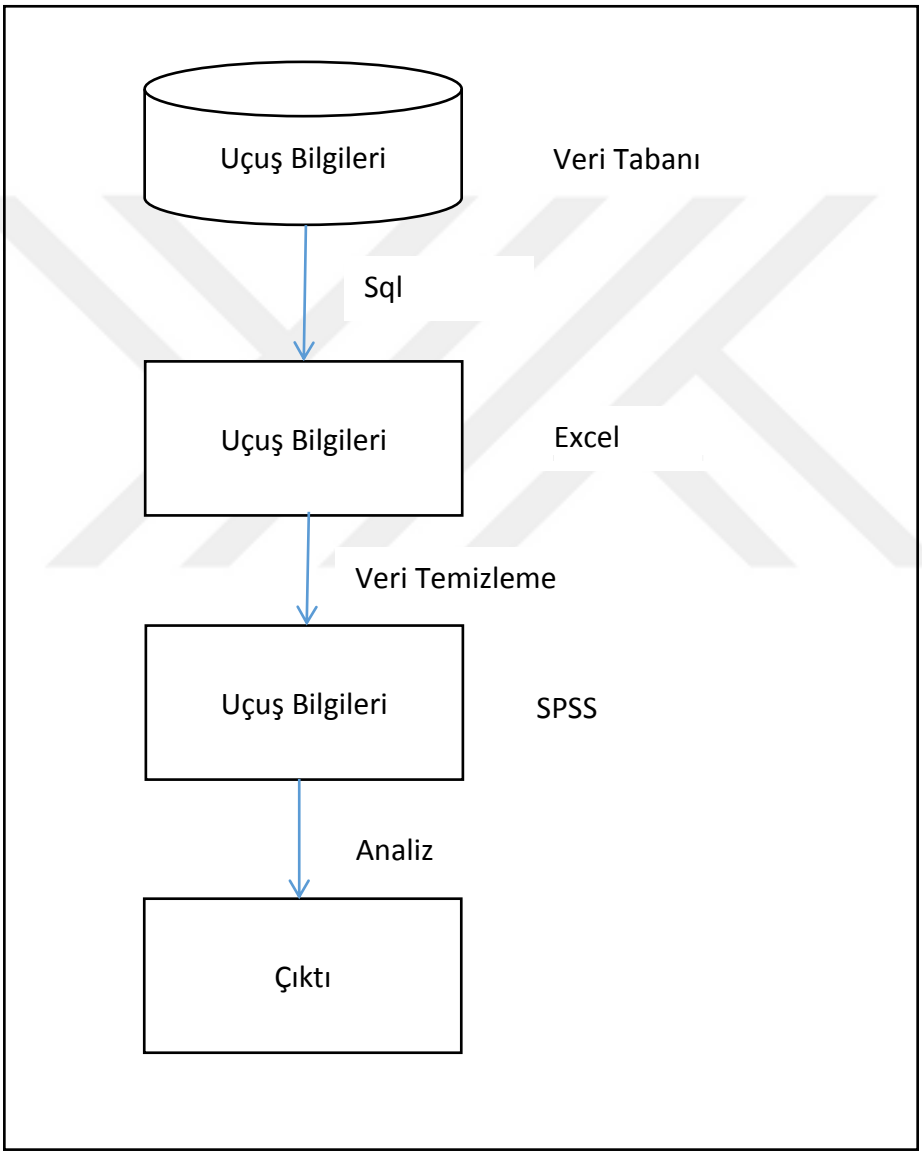
Türkiye havacılık sektöründe faaliyet gösteren ve sektörün büyük bir kısmına tek başına hakim konumda olan bir X havayolu şirketi ile görüşülürek verileri alınmıştır. Yapılan analizlerde sözkonusu firmanın 1 yıllık uçuş verisi kullanılmıştır. Uçuş verisinin kapsadığı bilgiler; kalkış ve varış bilgileri, uçak tipi, yolcu kapasitesi, uçağa biniş yapan yolcu sayısı ve yüklenen ikram sayısı şeklindedir. X havayolu şirketi pazar payı en yuksek olduğu için elde edilen sonuçların sektor için temsili olacağı düşünülmektedir.

7.4. Araştırmanın Değişkenleri

Araştırmadaki en önemli iki değişkenin, uçağa binen yolcu sayısı ve uçağa yüklenen ikram sayısı olduğu bilinmektedir. Buna göre, havalimanları bu iki değişkene göre kümeleme analizi ile gruplandırılmıştır. Elde edilen analiz sonuçları, fazla ikram problemini gidermek için gerekli yorumlar ve önlemler önermek için kullanılacaktır.

7.5. Araştırma Bulguları

Veritabanında bulunan uçuş bilgileri SQL sorguları ile çekilerek ham Excel dokümanı oluşturulmuştur. Sonrasında bu büyük ham dataya veri temizleme adımları uygulanmış ve yemek tipi, son kullanma tarihi, sınıf gibi gereksiz sütunlar veriden çıkarılmıştır. Uç ve sorunlu verilerin temizlenmesinden sonra veri analiz için hazır hale getirilmiştir. Bu durumu gösteren aşamaları Şekil 30'dan incelemek mümkündür.



Şekil 30 : Veri Toplama ve Uygulama Süreci

Araştırmada veri analizi için SPSS 20.0 (Statistical Packages for Social Sciences) paket programı kullanılmıştır.

Sözkonusu X havayolu firmasında 2016 Ocak ayında gerçekleşen, günde 293 iç hat seferine ilişkin olarak toplam 9083 uçuş bilgisi alınmıştır. İç hat seferlerinde uçaklar maksimum 165 yolcu kapasitesi ile seferlerini gerçekleştirmektedir. Bu nedenle uçaklara yüklenebilecek maksimum ikram sayısı da 165 dir.

7.5.1. Küme sayısı 3 için K-Means Analizi Sonuçları

K-Means algoritmasında, başlangıç olarak küme sayısı 3 olarak seçilmiştir. Algoritmanın ilk aşaması olan başlangıç küme merkezleri seçimi Tablo 9'daki gibidir.

Tablo 9 : K=3 için Başlangıç Küme Merkezleri

	Kümeler		
	1	2	3
YOLCU_SAYISI	145.00	145.00	165.00
YUKLENEN_IKRAM	145.00	165.00	165.00

Algoritmanın devamında gerçekleşen küme merkezi değişimleri Tablo 10'da gösterilmiştir. Tablo 10'da görüleceği gibi 9. adımda küme merkezi değişimi durmuştur. Küme merkezlerinin değişimleri 0 değerine ulaşana kadar iterasyonlar devam eder.

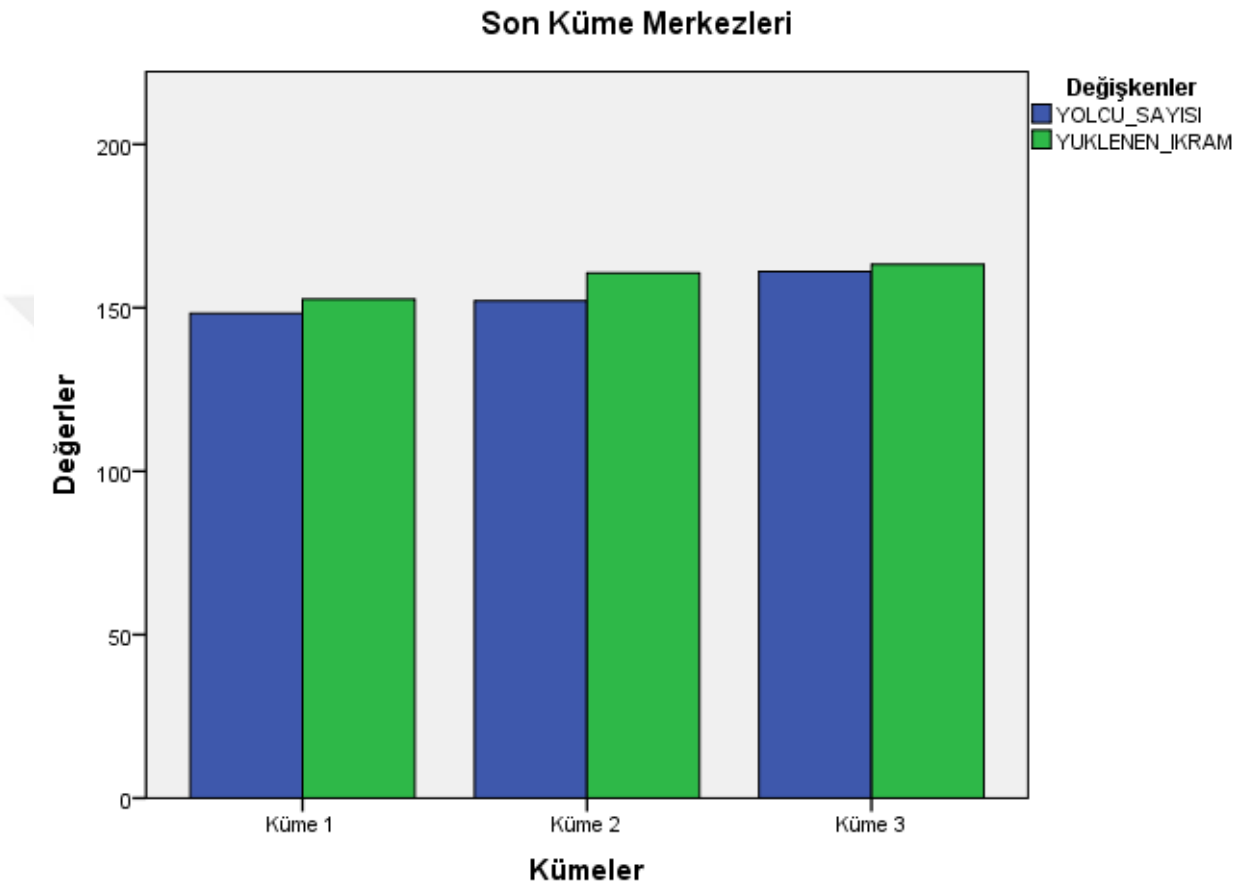
Tablo 10 : K=3 için Küme Merkezlerindeki Değişim

Adım	Küme Merkezlerindeki Değişim		
	1	2	3
1	7.777	7.258	5.075
2	.514	.507	.290
3	.207	.372	.206
4	.106	.207	.134
5	.000	.133	.120
6	.063	.096	.065
7	.065	.037	.000
8	.064	.037	.000
9	.000	.000	.000

Küme merkezlerinin değişimlerinden sonra, son küme merkezleri Tablo 11 ve Şekil 31'de gösterilmiştir.

Tablo 11 : K=3 için Son Küme Merkezleri

	Kümeler		
	1	2	3
YOLCU_SAYISI	148.31	152.12	161.05
YUKLENEN_IKRAM_SAYISI	152.62	160.64	163.25



Şekil 31: K=3 için Son Küme Merkezleri

Burada Tablo 11 analiz edildiğinde kümelerin boyutları ortaya çıkmaktadır. 1. Kümenin merkezi, (152.62-148.31)'den 4.31 çıkmıştır. İkramlar bölünemeyeceği için ortalama 4 fazla ikramlı uçuşlar olarak sıralayabiliriz. 2. Kümedeki durum ise, (160.64-152.12)'den 8.52, ortalama 8 fazla ikramlı uçuşlar olarak sıralanabilir. 3. Kümede ise durum ortalama 2 fazla ikramlı uçuşlar olarak sıralanmaktadır.

Kümelerin sınıflandırma sonuçları aşağıdaki gibidir;

- Küme 1, 4 fazla ikram ile *Orta*,
- Küme 2, 8 fazla ikram ile *Riskli*,
- Küme 3, 2 fazla ikram ile *Güvenli*, şeklinde görülmektedir.

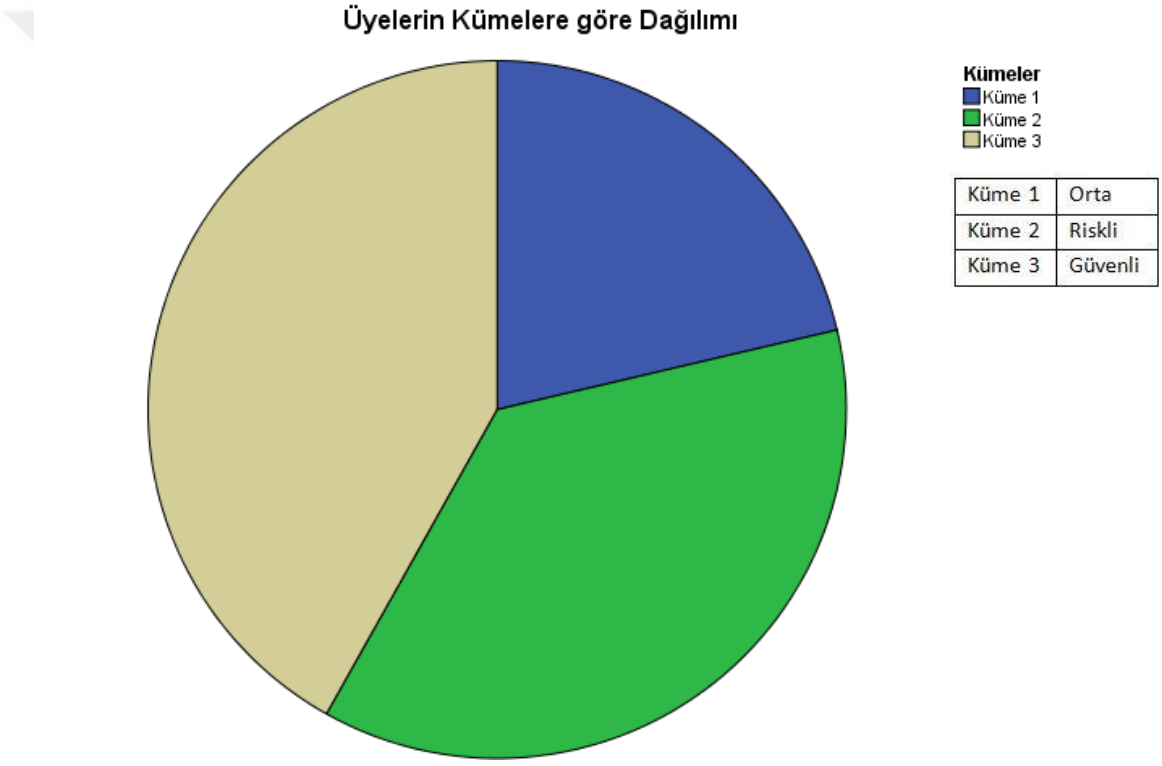
Veri kümemizdeki toplam eleman sayısı 9083 olup, bu elemanların kümele analizi sonrasındaki dağılımı Tablo 12'de gösterilmiştir.

Tablo 12 : K=3 için Üyelerin Kümelere Göre Dağılımı

Kümeler	1	1938.000
	2	3342.000
	3	3803.000
Geçerli		9083.000
Kayıp		.000

Küme 1	Orta
Küme 2	Riskli
Küme 3	Güvenli

Tablo 12 incelendiğinde en çok üyenin Küme 3'de toplandığı görülmektedir. Şekil 32'de de bu durum grafik olarak gösterilmektedir.



Şekil 32 : Üyelerin Kümelere Göre Dağılımı

Şekil 33 : Havalimanlarına Göre Günlük Sefer Sayıları

Bir gün içerisinde toplam olarak 293 tane iç hat seferi yapılmakta olup, sözkonusu bu seferler 43 farklı havalimanı arasında gerçekleştirilmektedir. Havalimanlarına göre sefer dağılımı Ek 2 esas alınarak Şekil 33'de gösterilmiştir. İnceleme yapıldığında günde en çok sefer yapılan ilk beş uçuş için aşağıdaki gibi sıralanmanın olduğu görülmektedir:

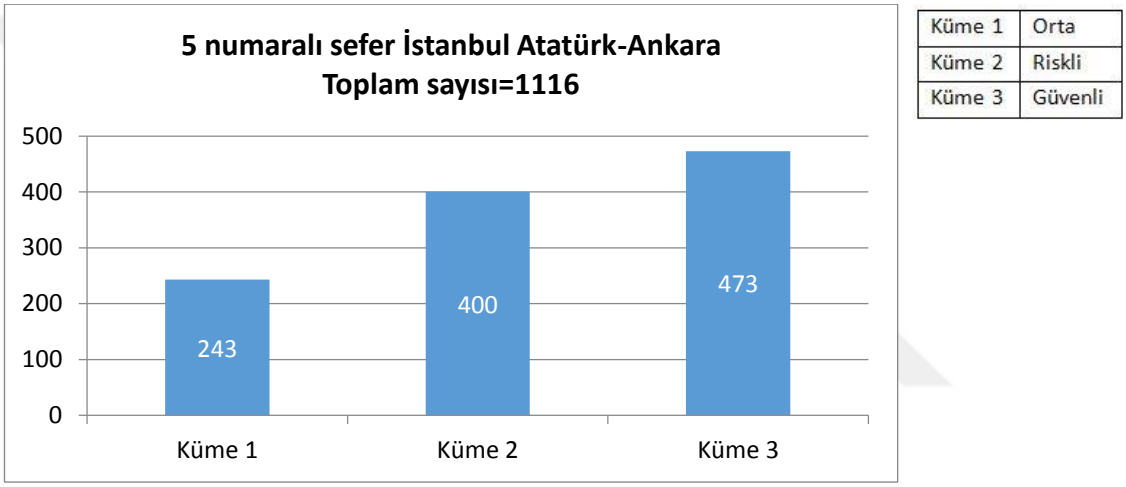
- 5 sefer numaralı İstanbul Atatürk-Ankara, 36 sefer,
- 11 sefer numaralı İstanbul Atatürk-İzmir, 24 sefer,
- 13 sefer numaralı İstanbul Atatürk-Adana, 19 sefer,
- 43 sefer numaralı İstanbul Sabiha Gökçen-Ankara, 19 sefer,
- 41 sefer numaralı İstanbul Sabiha Gökçen-İzmir, 16 sefer.

Kümeleme işlemi yapılırken, mevcut veri tablosuna ek bir sütun olarak analiz sonucunda her üyenin ait olduğu sınıf değeri eklenmiştir. Bu şekilde her bir sefere ait üyelerin hangi kümelere ait olduğu ortaya çıkarılmaktadır. Eldeki veriler günlük 293 sefer, aylık olarak da 9083 sefer olarak sıralanmaktadır, ayrıca her seferin ağırlığı da birbirine eşit değildir. Örneğin 1 numaralı sefer olan İstanbul Atatürk Havalimanı-Nevşehir Havalimanı arasında günlük toplam 4 sefer gerçekleşmekte iken; İstanbul

Atatürk Havalimanı-Kayseri Havalimanı arasında gerçekleşen 2 numaralı seferde ise günde toplam 8 sefer gerçekleşmektedir.

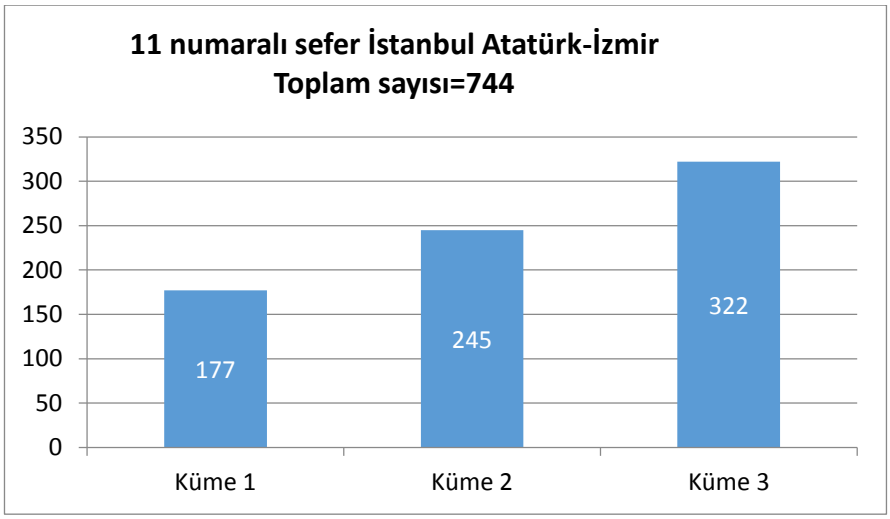
Bu durumda seferleri kendi aralarında, kümeleme analizi neticesinde ortaya çıkan sonuca göre sıralamak ve değerlendirmek için ayrı bir analize ihtiyaç duyulmuştur. Bu analizi gerçekleştirmek içinde Java programlama dilinde Ek 3'deki program yazılarak alınan çıktılar yorumlanmıştır. Bu program ile 9083 satır veri, her bir seferdeki üyelerin, Küme 1, 2 ve 3'e ait olan üyeleri ayrı ayrı bulunmuştur. Sonuç olarak 43 seferin ayrı ayrı kümelere ait olan üyeleri listelenerek Ek 4'de gösterilmektedir.

En çok sefer gerçekleşen 5 hat Şekil 34, 35, 36, 37 ve 38'de sıralanmaktadır.



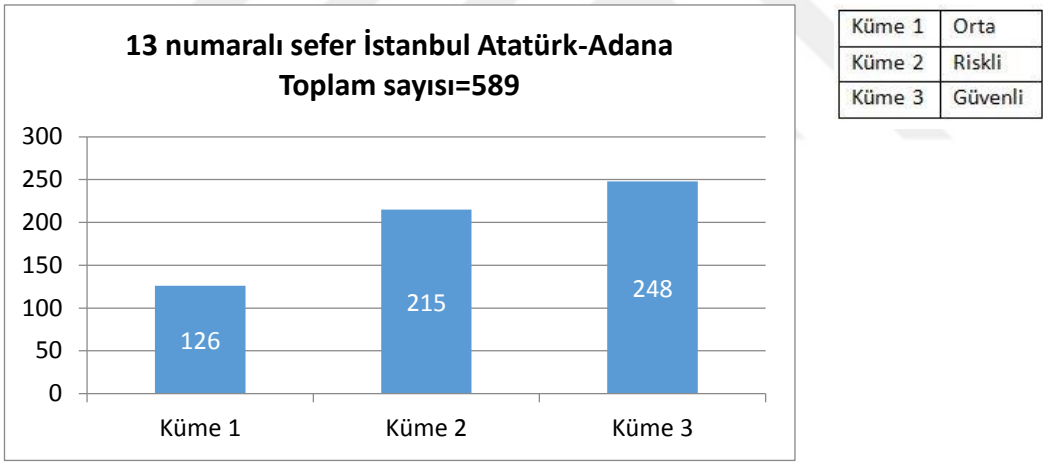
Şekil 34 : 5 Numaralı Sefer İstanbul Atatürk-Ankara

İstanbul Atatürk-Ankara arası gerçekleşen 5 numaralı sefer incelendiğinde; en fazla sayının 473 ile Küme 3'de yani güvenli kümede çıktığı görülmektedir. Bunu izleyen küme ise 400 değeri ile Küme 2 dir. Bu küme de riskli küme kapsamında değerlendirilebilinir.



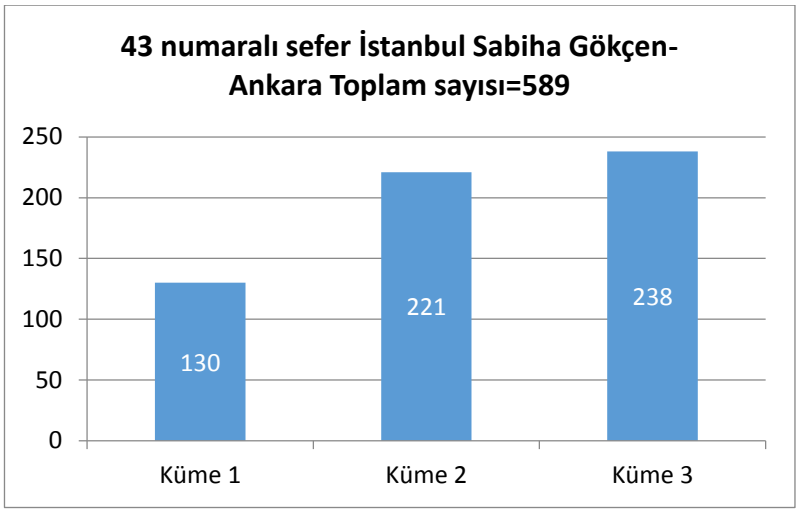
Şekil 35 : 11 Numaralı Sefer İstanbul Atatürk-İzmir

11 numaralı sefer olan İstanbul Atatürk- İzmir arasında gerçekleştirilen uçuşların toplamı 744'dir. Şekil 35'e göre sıralama; 322 değeri ile Küme 3, 245 ile Küme 2, 177 ile Küme 1 den oluşmaktadır.



Şekil 36 : 13 Numaralı Sefer İstanbul Atatürk-Adana

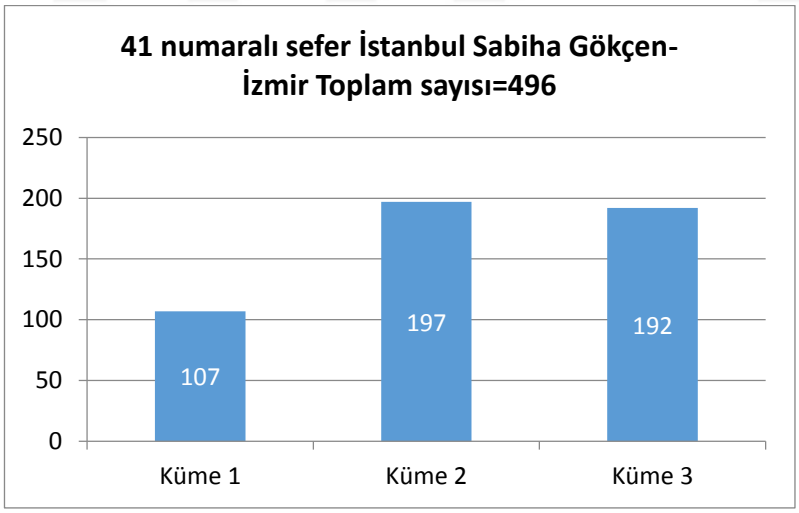
İstanbul Atatürk-Adana arasında toplam 589 sefer gerçekleşmektedir. Bu seferlerin, 248 tanesinin Küme 3'e ait olmak üzere güvenli, 215 tanesinin Küme 2'ye ait olarak riskli ve 126 tanesinin de Küme 1'e dahil olarak orta küme sınıfında kümelendikleri görülmüştür.



Küme 1	Orta
Küme 2	Riskli
Küme 3	Güvenli

Şekil 37 : 43 Numaralı Sefer İstanbul Sabiha Gökçen-Ankara

43 numaralı İstanbul Sabiha Gökçen-Ankara Esenboğa Havalimanı arasında, bir ay içinde 589 sefer gerçekleşmektedir. Bunlar içerisinde birbirine çok yakın değerler olarak görülen 238 ve 221 değerleri sırasıyla Küme 3 ve Küme 2'ye aittir. Son değer olan 130 ise Küme 1 yer alan eleman sayısını göstermektedir.



Küme 1	Orta
Küme 2	Riskli
Küme 3	Güvenli

Şekil 38 : 41 Numaralı Sefer İstanbul Sabiha Gökçen-İzmir

41 numaralı İstanbul Sabiha Gökçen-İzmir Havalimanları arasında gerçekleşen sefer sayısı bir ayda toplam 496 adettir. Dikkat çeken 197 miktarı ile Küme 2 yani riskli küme grubu öne çıkmaktadır. Bu değeri, 192 ile Küme 3'ün, 107 ile de Küme 1'in takip ettiği görülmektedir.

Aşağıdaki Tablo 13'de ANOVA tablosu verilmiştir. Kümeleme analizinde, ANOVA sonuçları, değişkenlerin belirlenen kümeler itibariyle farklılığının incelenmesi amacıyla kullanılmaktadır. Değişkenlerin kümeler itibariyle farklı çıkması beklenen bir durumdur. Kümeleme analizi yapısı gereği, bu farkı kendi içerisinde yaratmakta ve kümeler arasındaki farkı en büyüklemektedir. Buna göre, değişkenlere göre belirlenen kümeler arasında anlamlı bir farklılık bulunduğu görülmektedir.

Tablo 13 : K=3 için Anova Tablosu

	Küme		Hata		F	Sig.
	Ort Kare	df	Ort Kare	df		
YOLCU_SAYISI	126914.779	2	8.107	9080	15654.960	.000
YUKLENEN_IKRAM	73496.110	2	6.249	9080	11761.777	.000

7.5.2. Küme sayısı 4 için K-Means Analizi Sonuçları

Bu adımda uçuş verileri kümeleme analizinde kullanılırken, Küme sayısı 4 olarak belirlenmektedir. K-Means Algoritmasının başlangıç aşaması olan, başlangıç küme merkezi seçimi Tablo 14'deki gibidir

Tablo 14 : K=4 için Başlangıç Küme Merkezleri

	Kümeler			
	1	2	3	4
YOLCU_SAYISI	155.00	145.00	145.00	165.00
YUKLENEN_IKRAM	155.00	165.00	145.00	165.00

Tablo 14'de görüldüğü gibi, K-Means algoritması başlarken, keyfi başlangıç küme merkezleri seçilmektedir.

Algoritmanın devamında gerçekleşen küme merkezi değişimleri Tablo 15'de gösterilmektedir. Tablo 15'den de görülebileceği gibi 13. adımda küme merkezi değişimi durmaktadır.

Tablo 15 : K=4 için Küme Merkezlerindeki Değişim

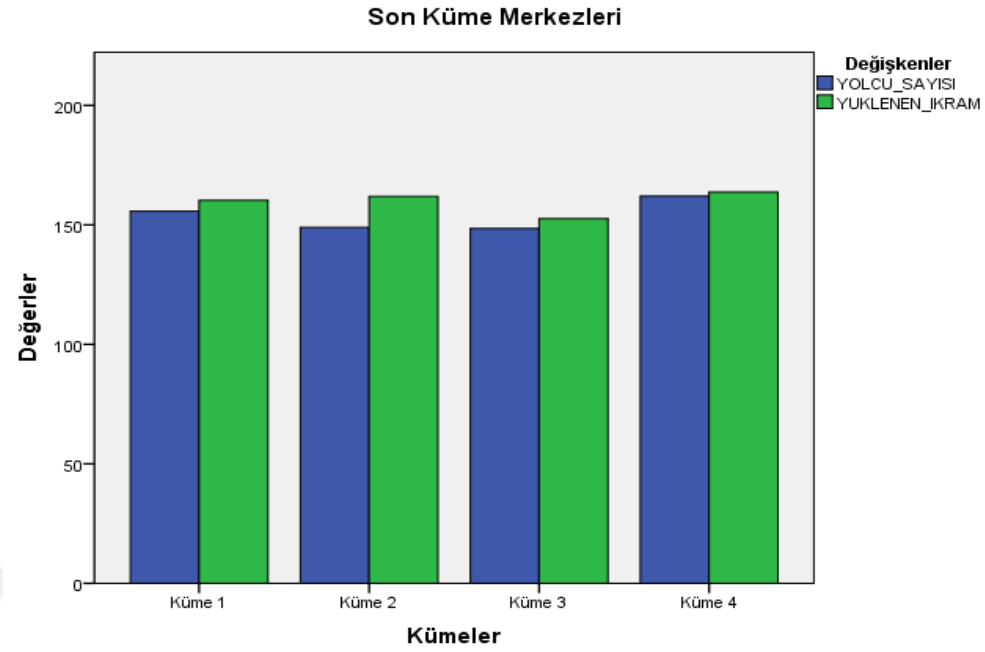
Adım	Küme Merkezlerindeki Değişim			
	1	2	3	4
1	3.399	4.467	4.679	3.487
2	.514	.498	1.948	.644
3	.578	.131	.656	.000
4	.488	.141	.393	.144
5	.375	.040	.131	.190
6	.373	.000	.263	.137
7	.296	.103	.069	.154
8	.122	.095	.000	.071
9	.120	.000	.081	.069
10	.192	.026	.225	.000
11	.121	.053	.071	.075
12	.095	.000	.000	.084
13	.000	.000	.000	.000

Tablo 15'deki adımlar sonrasında, son küme merkezleri Tablo 16 ve Şekil 39'de gösterilmektedir.

Tablo 16 : K=4 için Son Küme Merkezleri

	Kümeler			
	1	2	3	4
YOLCU_SAYISI	155.69	148.97	148.52	162.00
YUKLENEN_IKRAM	160.27	161.85	152.59	163.73

Tablo 16 ve Şekil 39 analiz edildiğinde ise kümelerin özelliklerinin belirginleştiği görülmektedir. 1. Kümenin merkezi, (160.27-155.69)'dan 4.58 çıkmaktadır. 2. Kümede durum, (161.85-148.97)'den 12.88, 3. Kümede ise (152.59-148.52) sonucuna göre 4.07 çıkmaktadır. Son olarak 4. Kümedeki durumun ise (163.73-162) işlemine göre 1.73 çıktığı görülmektedir.



Şekil 39 : K=4 için Son Küme Merkezleri

Kümeler, ikram değerleri tam sayıya yuvarlandıktan sonra aşağıdaki gibi sıralanmaktadır:

- Küme 1, 5 fazla ikramlı uçuşlar,
- Küme 2, 13 fazla ikramlı uçuşlar,
- Küme 3, 4 fazla ikramlı uçuşlar,
- Küme 4, 2 fazla ikramlı uçuşlar olarak görülmektedir.

Tablo 17 : K=4 için Üyelerin Kümelere Göre Dağılımı

Kümeler	1	2594.000
	2	1573.000
	3	1952.000
	4	2964.000
Geçerli		9083.000
Kayıp		.000

Veri kümesinde yer alan toplam 9083 eleman sayısının, kümele analizinden sonraki dağılımı yukarıda yer alan Tablo 17'de gösterilmektedir. Bu tabloya göre en çok elemanın 2964 birimle Küme 4'de, en az elemanın ise 1573 birimle Küme 2'de toplandığı görülmektedir. Tablo 17'den elde edilen sonuçlar Şekil 40 da görselleştirilmiştir.

Şekil 40 : Üyelerin Kümelere Göre Dağılımı

K değeri 4 seçildiğinde; küme değerleri sırasıyla 5, 13, 4 ve 2 olarak bulunmuştur. 3 kümenin değerinin birbirine çok yakın olmasından dolayı ve son değerin 13 ile uç bir değerde olması sebebiyle K=3 için K-Means algoritması optimum değer olarak kabul edilmiştir.

Tablo 18'de Küme sayısı 4 için ANOVA tablosu verilmiştir. Buna göre, değişkenlere göre belirlenen kümeler arasında anlamlı bir farklılık bulunduğu görülmektedir.

Tablo 18 : K=4 için Anova Tablosu

	Küme		Hata		F	Sig.
	Ort Kare	df	Ort Kare	df		
YOLCU_SAYISI	95193.010	3	4.611	9079	20645.342	.000
YUKLENEN_IKRAM	51312.493	3	5.484	9079	9356.026	.000

8. SONUÇ

Bilginin artık çok kolay üretildiği günümüzde, bu üretilen bilgilerin saklanması için de gerekli teknolojiler geliştirilmektedir. Veritabanlarında saklanan bu veriler, verinin üretilme hızıyla paralel bir şekilde büyümeye başlamıştır.

Şirketlerin her yapılan işlemi, her departman için ayrı ayrı tutmak istemesi; veri merkezinde birinden farklı birçok bilginin bir arada tutulduğu ve eldeki bilgi düzeyinin git gide büyüdüğü bir durumla karşılaşılmasına neden olmaktadır. Büyük verilerden anlamlı bilgilerin elde edilmesi süreci veri madenciliği yöntemlerinin uygulanmasını zorunlu hale getirmiştir. Bu nedenle, veri madenciliği bu ham büyük bilgilerden anlamlı bilgiler çıkarmak için sıklıkla uygulanmaktadır.

Karar verme konumunda olan kişiler, şirket ile ilgili stratejik kararları alırken, ellerinde doğru kararı verebilmelerine yardımcı olacak, temel bazı bilgilerin olmasını istemektedirler. Bu bilgilere çoğu zaman kolaylıkla ulaşılamaz. Çünkü basit veri tabanı sorguları bu ihtiyaca cevap vermemektedir. Bu durumda, veri ambarlarında duran ham veriye, veri madenciliği yöntemleri uygulanır ve yöneticinin ihtiyaç duyduğu veri, hali hazırda veri ambarında olan fakat çıkarılmayı bekleyen bilgilerden elde edilmektedir. Böylelikle karar destek sistemleri beslenerek, yöneticiye gerekli olan bilgiler raporlanabilmektedir.

Bu çalışmada uygulama olarak, veri madenciliği yöntemlerinden olan kümeleme analizi uygulanmıştır. Kümele analizi, büyük veri kümesini, birbirine benzeyen üyeleri birleştirerek alt kümelere ayrıma işlemidir. Genel olarak bu işlem üyelerin birbirine olan uzaklıkları kullanılarak yapılmaktadır. Araştırmada, kümeleme analizi yöntemlerinden biri olan K-Means algoritması kullanılmıştır.

Araştırma konusu olarak seçilmiş olan fazla ikram problemi, uçuşlarında ikram yapan tüm havayolları için geçerlidir. Uçağa binen yolcu sayısından daha fazla ikramın uçağa yüklendiği durum olan fazla ikram problemi detaylı olarak incelenmiştir.

Havayolu şirketi ekosistemine bilet alarak giren müşteriler, check-in yaparak, uçağa bineceğini belirtir ve çok doğaldır ki binmek zorunda değildir. Şirketin bu durumu teknik olarak bilme ihtimali maalesef yoktur ve bu yolcular için uçağa ikram yüklememe ihtimalleri de yoktur. Bu durum için getirilebilecek öneri, mobil uygulamaya ve web sitesine, uçağa binmeyecek yolcunun bu durumu bildirebileceği bir servis eklemektir. Önerinin kullanıcı inisiyatifli olması negatif bir etmen olmakla beraber, uygulamayı cazip kılacak hediye mil, kesintili ücret iadesi veya bilet hakkı verilmesi, kullanımı artırabilecektir.

Fazla ikrama neden olan ikinci sebep ise, yolcunun kendi rızası dışında, polis zoruyla alıkoyulmasıdır. Bu durum, dış hatlar yolcularında ise pasaport ve vize sorunlarından, iç hat yolcularında ise ihbar ve şüphe kaynaklı olabilmektedir. Bu durumu çözmek, havayolu şirketlerinin tek başlarına kendi ellerinde değildir. Havalimanı meydan işletmeleri ve havaalanı polisinin, durdurulan yolcunun bilgilerini bir sisteme girebilmesi, havayolu şirketlerinin de, bu kişinin kendi yolcularından biri olup olmadığını kontrol ederek gerekli ikram yüklemesini bu durumu göz önüne alarak yapması en optimal olandır.

Bu iki öneri, problemin araştırılması, havayolu şirketlerinin iş süreçleri, meydan otoritelerinin uygulamaları göz önünde bulundurularak yapılmıştır.

Veri analiz kısmında, havayolu şirketinden alınan uçuş bilgileri incelenmiştir. Bu bilgiler içinde, kalkış ve varış bilgileri, uçak tipi, yolcu kapasitesi, uçağa biniş yapan yolcu sayısı ve yüklenen ikram sayısı bulunmaktadır. Günde 43 farklı havalimanı arasında sefer düzenleyen şirket, Ek 3'deki tabloya göre toplamda bir günde 293 farklı uçuş gerçekleştirmektedir. Ocak 2016 da toplam 9083 uçuş bulunmaktadır. Bu veriler, SPSS 20 paket programı aracılığı ile K-Means algoritmasından geçirilmiştir. Hedef küme sayısı, uzman görüşü alınarak 3 ve 4 olarak belirlenmiştir. K=4 seçildiğinde üç kümenin değerlerinin birbirine çok yakın çıkmasından dolayı analizler K=3'e göre yapılmıştır. K=3 için algoritma çalıştırılmış ve her uçuşun ait olduğu sınıf veri setine ilave bir sütun olarak eklenmiştir. Bu analiz ile toplam uçuşlardaki dağılım gözlenmiştir ve genel çerçevede operasyonel bir sorun olup olmadığı ortaya koyulmuştur.

Bir sonraki adımda 43 seferin ayrı ayrı hangi kümede kaç elemanının olduğunun belirlenmesi amaçlanmıştır. 9083 veri için bu aramanın yapılabilmesi amacıyla, Ek 4 deki program yazılmış, her sefer için ayrı ayrı küme aidiyetleri bulunmuş ve Ek 4'de

43 ayrı grafik ile elde edilen sonuçlar gösterilmiştir. Sözkonusu olan bu ikinci analizin yapılmasındaki amaç, daha çok sorun yaşanan havalimanlarında gerekli olan önlemleri alabilmektir. Sorun yaşanan havalimanlarında ikram yüklemesini, uçağa binişe (boarding), operasyonun elverdiği en yakın zamanda veya mümkünse uçağa biniş başladıktan sonra yapmak, uçuş listesinde olup, uçağa binmemiş yolcuların tespit edilme ihtimalini artırır, dolayısı ile fazla ikram yüklemenin önüne geçilmiş olmaktadır.

Gerçekleştirilen nicel araştırma ile veri madenciliğinin amacı olan analiz ve yönetime rapor çıkarma işlemleri gerçekleştirilebilmektedir. Sözkonusu rapor ile karar alcılara öneriler sunulmakta ve alınması gereken tedbirler sıralanabilmektedir.

KAYNAKÇA

Akpınar, Haldun. **Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği**, İstanbul Üniversitesi İşletme Fakültesi Dergisi, 2000

Berry Michael, Browne Murray, **Lecture Notes In Data Mining Singapore: World Scientific Publishing**, 2006

Burma, Zehra Alakoç. **Veri Tabanı Yönetim Sistemleri**, Seçkin Yayıncılık, 2009

Chapman, Clinton Julian, **Crisp-Dm 1.0 Step-By-Step Data Mining Guide**, 2000

Codd, Edgar Frank. **The Relatıonal Model for Database Management**, 2000

Çakır, Ferhat. **Analysis of Some Data Relating to Small and Medium-Sized Enterprises Using Data Mining Methods**, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2012

Çam, Selim. **Veri Madenciliğinde Kümeleme Analizi ve Sağlık Sektöründe Bir Uygulama**, Cumhuriyet Üniversitesi, Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, 2014

Çamurcu, Yılmaz. **DBSCAN, OPTICS ve K-Means Kümeleme Algoritmalarının Uygulamalı Karşılaştırılması**, Politeknik Dergisi, Cilt: 8 Sayı: 2, 2005

Çetinkaya, Melih Başaraner, **Kentsel Alanlarda Lokal Bina Örüntülerinin Elde Edilmesi: Karşılaştırmalı Bir Çalışma,** Tmmob Coğrafi Bilgi Sistemleri Kongresi Antalya, 2011

Darakçı, Halil Çağdaş. **Kümeleme Analizi Kullanılarak Benzin istasyonlarının Operasyonel Değerlendirilmesi**, 2014

Dunham, Margaret. **Data Mining Introductory and Advanced Topics,** Prentice Hall, Pearson Education Inc., New Jersey, 2003

Elmasri, Shamkant Navathe. **Fundementals of Database Systems**, Fourth Edition, 2003

Fayyad Usama, Shapiro Gregory. **From Data Mining to Knowledge Discovery in Databases**, American Associatin for Artificial Intelligence, 1996

Gemici, Burhan. **Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Ekonometri Anabilim Dalı Ekonometri Programı Yüksek Lisans Tezi,** 2012

Gökmen, Şenol. **Müşteri İlişkileri Yönetiminde Bir Araç Olarak Veri Madenciliği ve Perakende Sektöründe Bir Uygulama**, 2014

Kevin, Beyer. **When is Nearest Neighbor Meaningfull**, 7. Uluslararası Database Theory Konferansı, İsrail, 1999

Koldere Akın, Yasemin. **Veri Madenciliğinde Kümeleme Algoritmaları ve Kümeleme Analizi**, Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, 2008

Köse, Yunus. **Değerli Müşterilerde Ürün Kategorileri Arasındaki Satış İlişkilerinin Veri Madenciliği Yöntemlerinden Birliktelik Kuralları ve Kümeleme Analizi ile Belirlenmesi ve Ulusal bir Perakendecide Örnek Uygulama**, Selçuk Üniversitesi Yüksek Lisans Tezi, 2015

Kunttu Iivari, Visa Ari. **Classification Method for Defect Images Based on Association and Clustering**, Data Mining and Knowledge DiscoveryUSA, 2003

Larose, Daniel. **Discovering Knowledge In Data: An Introduction To Data Mining**. New Jersey: A. John Willey&Sons, 2005

Olson, Yong Shi. **Introduction To Busıness Data Mining**, McGraw Hill, 2007

Ordu, Bülent. **Veri Madenciliğinde Sınıflayıcı Teknikler ile Demir Çelik Sektöründe Bir Uygulama**, 2013

Özdemir, Abdülkadir. **Lise Türü Ve Lise Mezuniyet Başarısının, Kazanılan Fakülte İle İlişkisinin Veri Madenciliği Tekniği İle Analizi**, Atatürk Üniversitesi, Sosyal Bilimleri Enstitüsü Dergisi, Cilt 10, Sayı 2, 2007

Özkan, Gökhan. **Veri Ambarlama ve Veri Madenciliği ''Chameleon Algoritması** ,Gazi Üniversitesi Bilişim Enstitüsü, https://groups.google.com/forum/#!msg/veri_madenciligi/-5y0NfGR8WU/8SXACBgvy24J, 21.05.2016

Özkan, Yalçın. **Veri Madenciliği Yöntemleri**, Papatya Yayıncılık, 2013

Pei, Jian. **Data Mining Concepts and Techniques**, 2012

Quinlan, John Ross. **Induction of Decision Trees**, Machine Learning, 1986

Rakes, Srikant Ramakrishnan. **Fast Algorithms for Mining Assosication Rules,** 20. VLDB Konferansı, Şili, 1994

Savaş, Topaloğlu. **İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi**, Yıl:11 Sayı: 21 Bahar 2012

Serkan, Nurettin. **Veri Madenciliği Ve Türkiye'deki Uygulama Örnekleri**, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi Yıl:11 Sayı: 21 Bahar

Silahtaroğlu**,** Gökhan**. Veri Madenciliği**, Papatya Yayıncılık, 2008

Sivil Havacılık Genel Müdürlüğü, **2014 Faaliyet Raporu**, http://web.shgm.gov.tr/documents/sivilhavacilik/files/pdf/kurumsal/raporlar/2014faaliyetraporuv2.pdf 23.05.2016

Tuğba, Şimşek Gürsoy**, Veri Madenciliği ve Bilgi Keşfi**, 2009

Turan, Emel Seymen. **Bir Telekomünikasyon Firmasında Müşteri Segmentasyonu,** 2010

Tutar, Erol. **Veri Madenciliği Yöntemiyle Döviz Kuru Tahmini**, 2011

Türkiye'deki Havalimanları, http://www.dhmi.gov.tr/havaalanlari.aspx 24.05.2016

Uçan, Ömer. **Dijital Kütüphanelerde Veri Madenciliği Uygulamaları**: Akdeniz Üniversitesi Merkez Kütüphanesi Örneği, 2010

Wang, Yang. **A Statistical Information Grid Approach To Spatial Data Mining**, University of California,1997

Yıldızeli, Arıkan. **Bilgi Çağında Varoluş: "Fırsatlar ve Tehditler" Sempozyumu Yeditepe Üniversitesi**, 2009

EKLER

Ek 1. Türkiye'deki Havalimanları[92]

Konum	ICAO	IATA	Havalimanı	Koordinatlar
Uluslararası havalimanları				
Adana	LTAF	ADA	Şakirpaşa Havalimanı	36°58′55″K 35°16′49″ D
Ankara	LTAC	ESB	Ankara Esenboğa Havalimanı	40°07′41″K 32°59′42″ D
Antakya	LTAK	HTY	Hatay Havalimanı	36°21′46″K 36°16′56″ D
Antalya	LTAI	AYT	Antalya Havalimanı	36°54′01″K 30°47′34″ D
Milas	LTFE	BJV	Milas-Bodrum Havalimanı	37°15′02″K 27°39′51″ D
Bursa	LTBR	YEI	Bursa Yenişehir Havalimanı	40°15′18″K 29°33′45″ D
Dalaman	LTBS	DLM	Dalaman Havalimanı	36°42′53″K 28°47′34″ D
Diyarbakır	LTCC	DIY	Diyarbakır Havalimanı	37°53′38″K 40°12′03″ D
Erzurum	LTCE	ERZ	Erzurum Havalimanı	39°57′19″K 41°10′09″ D
Eskişehir	LTBY	AOE	Eskişehir Anadolu Havalimanı	39°48′36″K 30°31′10″ D
Ordu-Giresun	LTCB	OGU	Ordu-Giresun Havalimanı	40°58′05″K 38°04′35″ D
Gaziantep	LTAJ	GZT	Gaziantep Havalimanı	36°56′52″K 37°28′44″ D
İstanbul	LTBA	IST	Atatürk Havalimanı	40°58′34″K 28°48′51″ D
İstanbul	LTFJ	SAW	Sabiha Gökçen Havalimanı	40°53′54″K 29°18′33″ D
İzmir	LTBJ	ADB	Adnan Menderes Havalimanı	38°17′21″K 27°09′18″ D
Kayseri	LTAU	ASR	Erkilet Havalimanı	38°46′13″K 35°29′43″ D
Konya	LTAN	KYA	Konya Havalimanı	37°58′44″K 32°33′42″

[92] http://www.dhmi.gov.tr/havaalanlari.aspx Sitesinden Derlenmiştir. 24.05.2016
IATA: International Air Transport Association
ICAO : International Civil Aviation Organization

				D
Kütahya	LTBZ	KZR	Zafer Havalimanı	39°06′41″K 30°07′47″ D
Malatya	LTAT	MLX	Erhaç Havalimanı	38°26′07″K 38°05′27″
Nevşehir	LTAZ	NAV	Nevşehir Kapadokya Havalimanı	38°46′08″K 34°31′35″ D
Samsun	LTFH	SZF	Samsun Çarşamba Havalimanı	41°15′56″K 36°32′55″ D
Şanlıurfa	LTCS	GNY	Şanlıurfa GAP Havalimanı	37°27′00″K 38°54′00″ D
Trabzon	LTCG	TZX	Trabzon Havalimanı	40.9951°K 39.7897°D
Yerel havalimanları				
Adıyaman	LTCP	ADF	Adıyaman Havalimanı	37°43′54″K 38°28′08″ D
Ağrı	LTCO	AJI	Ağrı Havalimanı	39°39′16″K 43°01′38″ D
Amasya	LTAP	MZH	Amasya Merzifon Havalimanı	40°49′45″K 35°31′19″ D
Antalya	LTGP	GZP	Gazipaşa Havalimanı	36.2993°K 32.3014°D
Balıkesir	LTBF	BZI	Balıkesir Merkez Havalimanı	39°37′09″K 27°55′33″ D
Balıkesir	LTFD	EDO	Balıkesir Koca Seyit Havalimanı	39°33′16″K 27°00′49″ D
Batman	LTCJ	BAL	Batman Havalimanı	37°55′56″K 41°06′59″ D
Bingöl	LTCU	BGG	Bingöl Havalimanı	38°51′36″K 40°35′38″ D
Çanakkale	LTBH	CKZ	Çanakkale Havalimanı	40°08′15″K 26°25′36″ D
Denizli	LTAY	DNZ	Denizli Çardak Havalimanı	37°47′08″K 29°42′04″ D
Elâzığ	LTCA	EZS	Elazığ Havalimanı	38°36′24″K 39°17′29″ D
Erzincan	LTCD	ERC	Erzincan Havalimanı	39°42′36″K 39°31′37″ D
Gökçeada	LTFK	GKD	Gökçeada Havalimanı	40°12`04"N, 025°52`56"E
Hakkâri	LTCW	YKO	Hakkâri Yüksekova Selahaddin Eyyubi Havalimanı	37°33′6″K 44°14′01″D
Iğdır	LTCT	IGD	Iğdır Havalimanı	39°58′59″K 43°51′59″ D
Isparta	LTFC	ISE	Isparta Süleyman Demirel Havalimanı	37°51′54″K 30°22′55″ D
İstanbul	LTBW	-	İstanbul Hezarfen Havalimanı	41°06′16″K 28°33′00″ D
Kahramanmara	LTCN	KCM	Kahramanmaraş	37°32′20″K 36°57′12″

ş			Havalimanı	D
Kars	LTCF	KSY	Kars Harakani Havalimanı	40°33′44″K 43°06′54″ D
Kastamonu	LTAL	KFS	Kastamonu Havalimanı	41.3138°K 33.7949°D
Kocaeli	LTBQ	KCO	Cengiz Topel Havalimanı	40°44′06″K 30°05′00″ D
Mardin	LTCR	MQM	Mardin Havalimanı	37°13′58″K 40°38′26″ D
Muş	LTCK	MSR	Muş Havalimanı	38°44′41″K 41°39′14″ D
Siirt	LTCL	SXZ	Siirt Havalimanı	37°58′00″K 41°50′00″ D
Sinop	LTCM	NOP	Sinop Havalimanı	42°00′57″K 35°03′59″ D
Sivas	LTAR	VAS	Sivas Nuri Demirağ Havalimanı	39°46′00″K 37°01′00″ D
Şırnak	LTCV	NKT	Şırnak Şerafettin Elçi Havalimanı	37°21′48″K 42°03′35″ D
Tekirdağ	LTBU	TEQ	Tekirdağ Çorlu Havalimanı	41°08′18″K 27°55′09″ D
Tokat	LTAW	TJK	Tokat Havalimanı	40°18′42″K 36°22′25″ D
Uşak	LTBO	USQ	Uşak Havalimanı	38.6796°K 29.4816°D
Van	LTCI	VAN	Ferit Melen Havalimanı	38°28′06″K 43°19′56″ D
Zonguldak	LTAS	ONQ	Zonguldak Havalimanı	41.5064°K 32.0886°D

Ek 2. Günlük İç Hat Sefer Bilgileri

SEFER_NO	SEFER_LİMANLARI	SEFER_SAYISI
1	İstanbulAtatürk-Nevşehir	4
2	İstanbulAtatürk-Kayseri	8
3	İstanbulAtatürk-Konya	6
4	İstanbulAtatürk-Kütahya	2
5	İstanbulAtatürk-Ankara	36
6	İstanbulAtatürk-Kahramanmaraş	4
7	İstanbulAtatürk-Adıyaman	2
8	İstanbulAtatürk-Gaziantep	9
9	İstanbulAtatürk-Şanlıurfa	4
10	İstanbulAtatürk-Hatay	6
11	İstanbulAtatürk-İzmir	24
12	İstanbulAtatürk-Antalya	15
13	İstanbulAtatürk-Adana	19
14	İstanbulAtatürk-MilasBodrum	4
15	İstanbulAtatürk-Dalaman	4
16	İstanbulAtatürk-Denizli	6
17	İstanbulAtatürk-AntalyaGazipaşa	6
18	İstanbulAtatürk-Diyarbakır	7
19	İstanbulAtatürk-Malatya	4
20	İstanbulAtatürk-Elazığ	4
21	İstanbulAtatürk-Bingöl	2
22	İstanbulAtatürk-Erzincan	2
23	İstanbulAtatürk-Sivas	4
24	İstanbulAtatürk-Şırnak	2
25	İstanbulAtatürk-Hakkari	2
26	İstanbulAtatürk-Mardin	4
27	İstanbulAtatürk-Balıkesir	6
28	İstanbulAtatürk-Muş	2
29	İstanbulAtatürk-Erzurum	4
30	İstanbulAtatürk-Kars	2
31	İstanbulAtatürk-Iğdır	2
32	İstanbulAtatürk-Ağrı	2
33	İstanbulAtatürk-Kastamonu	2
34	İstanbulAtatürk-Van	4
35	İstanbulAtatürk-Samsun	8
36	İstanbulAtatürk-Trabzon	14
37	İstanbulAtatürk-Sinop	2
38	İstanbulAtatürk-OrduGiresun	4
39	İstanbulAtatürk-Amasya	2
40	İstanbulAtatürk-Isparta	2

41	İstanbulSabiha-İzmir	16
42	İstanbulSabiha-Antalya	12
43	İstanbulSabiha-Ankara	19
	TOPLAM	**293**

Ek 3. Kümeleme Analizi Sonrası, Belirlenen Kümelerin Seferlere Göre Gruplandırılması için Kullanılan Java Programı ve Çıktısı

DataTransform.java

```java
package com.spss.analyze;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author Burak Levent
 *
 */
public class DataTransform {

    public static void main(String args[])throws IOException{

        File file = new File("C:\\Users\\E_LEVENT\\Desktop\\k3_seferKumeSayisi.txt");
        FileReader fr = new FileReader(file);
        BufferedReader br = new BufferedReader(fr);

        String sCurrentLine;
        List<String> flights = new ArrayList<String>();
        List<String> clusters = new ArrayList<String>();

        while ((sCurrentLine = br.readLine()) != null) {
            //System.out.println(sCurrentLine);
            String value[] = sCurrentLine.split(":");
            flights.add(value[0]);
            clusters.add(value[1]);
        }

        int k;
        for (k=1; k<=43; k++ ){
            int j1=0;//küme1
            int j2=0;//küme2
            int j3=0;//küme3

            int eachCluster=0;
            for(int i=0; i<flights.size() ; i++ ){
                if( flights.get(i).equals(String.valueOf(k))){
                    eachCluster++;
                    if(clusters.get(i).equals("1")) j1++;
                    if(clusters.get(i).equals("2")) j2++;
                    if(clusters.get(i).equals("3")) j3++;

                }
            }
            System.out.println(k + ". seferin toplam sayisi=" + eachCluster + " , kume sayilari >>");
            System.out.println(j1);
            System.out.println(j2);
            System.out.println(j3);

        }

        br.close();
    }
}
```

Çıktı:

```
1. seferin toplam sayisi=124 , kume sayilari >>
29
46
49
2. seferin toplam sayisi=248 , kume sayilari >>
56
91
101
3. seferin toplam sayisi=186 , kume sayilari >>
33
65
88
4. seferin toplam sayisi=62 , kume sayilari >>
16
22
24
5. seferin toplam sayisi=1116 , kume sayilari >>
243
400
473
6. seferin toplam sayisi=124 , kume sayilari >>
19
51
54
7. seferin toplam sayisi=62 , kume sayilari >>
14
23
25
8. seferin toplam sayisi=279 , kume sayilari >>
54
107
118
9. seferin toplam sayisi=124 , kume sayilari >>
27
45
52
10. seferin toplam sayisi=186 , kume sayilari >>
33
58
95
11. seferin toplam sayisi=744 , kume sayilari >>
177
245
322
12. seferin toplam sayisi=465 , kume sayilari >>
94
174
197
```

13. seferin toplam sayisi=589 , kume sayilari >>
126
215
248
14. seferin toplam sayisi=124 , kume sayilari >>
32
44
48
15. seferin toplam sayisi=124 , kume sayilari >>
23
51
50
16. seferin toplam sayisi=186 , kume sayilari >>
36
62
88
17. seferin toplam sayisi=186 , kume sayilari >>
44
78
64
18. seferin toplam sayisi=217 , kume sayilari >>
50
90
77
19. seferin toplam sayisi=124 , kume sayilari >>
28
46
50
20. seferin toplam sayisi=124 , kume sayilari >>
24
48
52
21. seferin toplam sayisi=62 , kume sayilari >>
8
28
26
22. seferin toplam sayisi=62 , kume sayilari >>
16
20
26
23. seferin toplam sayisi=124 , kume sayilari >>
22
50
52
24. seferin toplam sayisi=62 , kume sayilari >>
14
24
24
25. seferin toplam sayisi=62 , kume sayilari >>
10

24
28
26. seferin toplam sayisi=124 , kume sayilari >>
31
47
46
27. seferin toplam sayisi=186 , kume sayilari >>
29
63
94
28. seferin toplam sayisi=62 , kume sayilari >>
15
16
31
29. seferin toplam sayisi=124 , kume sayilari >>
28
49
47
30. seferin toplam sayisi=62 , kume sayilari >>
15
22
25
31. seferin toplam sayisi=62 , kume sayilari >>
17
19
26
32. seferin toplam sayisi=62 , kume sayilari >>
14
23
25
33. seferin toplam sayisi=62 , kume sayilari >>
16
15
31
34. seferin toplam sayisi=124 , kume sayilari >>
28
45
51
35. seferin toplam sayisi=248 , kume sayilari >>
60
94
94
36. seferin toplam sayisi=434 , kume sayilari >>
84
160
190
37. seferin toplam sayisi=62 , kume sayilari >>
8
29
25

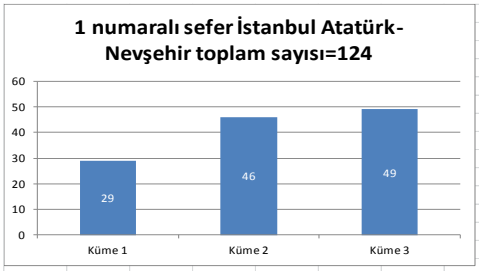
38. seferin toplam sayisi=124 , kume sayilari >>
26
48
50
39. seferin toplam sayisi=62 , kume sayilari >>
12
22
28
40. seferin toplam sayisi=62 , kume sayilari >>
8
21
33
41. seferin toplam sayisi=496 , kume sayilari >>
107
197
192
42. seferin toplam sayisi=372 , kume sayilari >>
82
144
146
43. seferin toplam sayisi=589 , kume sayilari >>
130
221
238

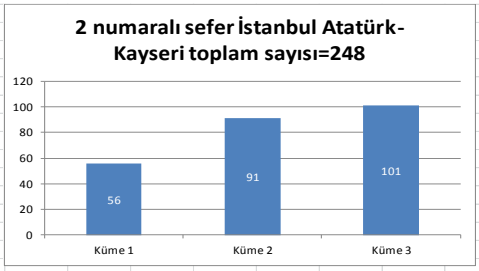
Ek 4. Aylık İç Hat Sefer Bilgileri ve Küme Dağılımları

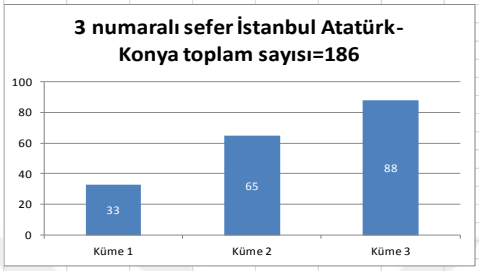
SEFER NO	SEFER_LİMANLARI	SEFER SAYISI	Küme1	Küme2	Küme3
1	İstanbulAtatürk-Nevşehir	124	29	46	49
2	İstanbulAtatürk-Kayseri	248	56	91	101
3	İstanbulAtatürk-Konya	186	33	65	88
4	İstanbulAtatürk-Kütahya	62	16	22	24
5	İstanbulAtatürk-Ankara	1116	243	400	473
6	İstanbulAtatürk-Kahramanmaraş	124	19	51	54
7	İstanbulAtatürk-Adıyaman	62	14	23	25
8	İstanbulAtatürk-Gaziantep	279	54	107	118
9	İstanbulAtatürk-Şanlıurfa	124	27	45	52
10	İstanbulAtatürk-Hatay	186	33	58	95
11	İstanbulAtatürk-İzmir	744	177	245	322
12	İstanbulAtatürk-Antalya	465	94	174	197
13	İstanbulAtatürk-Adana	589	126	215	248
14	İstanbulAtatürk-MilasBodrum	124	32	44	48
15	İstanbulAtatürk-Dalaman	124	23	51	50
16	İstanbulAtatürk-Denizli	186	36	62	88
17	İstanbulAtatürk-AntalyaGazipaşa	186	44	78	64
18	İstanbulAtatürk-Diyarbakır	217	50	90	77
19	İstanbulAtatürk-Malatya	124	28	46	50
20	İstanbulAtatürk-Elazığ	124	24	48	52
21	İstanbulAtatürk-Bingöl	62	8	28	26
22	İstanbulAtatürk-Erzincan	62	16	20	26
23	İstanbulAtatürk-Sivas	124	22	50	52
24	İstanbulAtatürk-Şırnak	62	14	24	24
25	İstanbulAtatürk-Hakkari	62	10	24	28
26	İstanbulAtatürk-Mardin	124	31	47	46
27	İstanbulAtatürk-Balıkesir	186	29	63	94
28	İstanbulAtatürk-Muş	62	15	16	31
29	İstanbulAtatürk-Erzurum	124	28	49	47
30	İstanbulAtatürk-Kars	62	15	22	25
31	İstanbulAtatürk-Iğdır	62	17	19	26

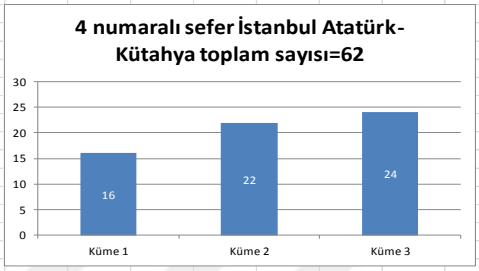
32	İstanbulAtatürk-Ağrı	62	14	23	25
33	İstanbulAtatürk-Kastamonu	62	16	15	31
34	İstanbulAtatürk-Van	124	28	45	51
35	İstanbulAtatürk-Samsun	248	60	94	94
36	İstanbulAtatürk-Trabzon	434	84	160	190
37	İstanbulAtatürk-Sinop	62	8	29	25
38	İstanbulAtatürk-OrduGiresun	124	26	48	50
39	İstanbulAtatürk-Amasya	62	12	22	28
40	İstanbulAtatürk-Isparta	62	8	21	33
41	İstanbulSabiha-İzmir	496	107	197	192
42	İstanbulSabiha-Antalya	372	82	144	146
43	İstanbulSabiha-Ankara	589	130	221	238
	TOPLAM	**9083**	**1938**	**3342**	**3803**

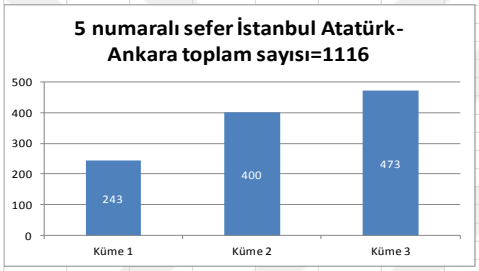
Ek 5. Tüm Seferlerin Küme Sayıları Grafikleri

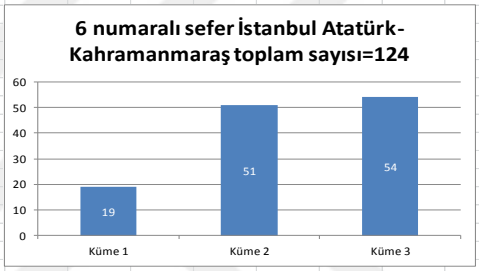


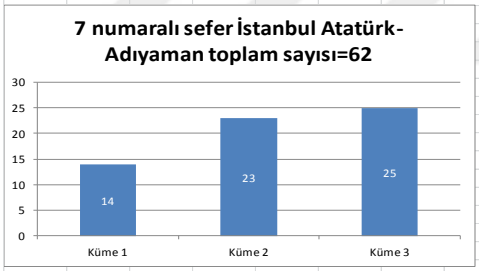


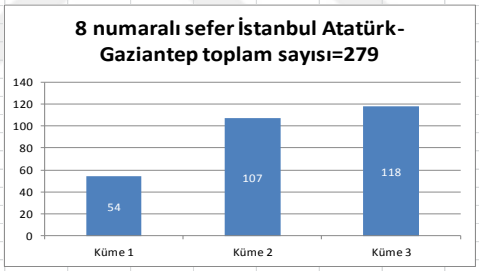


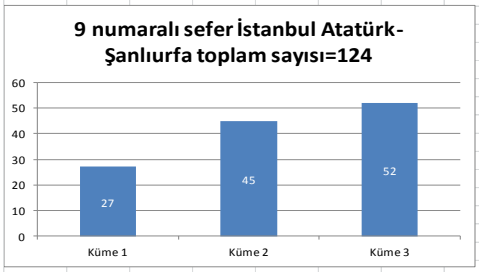


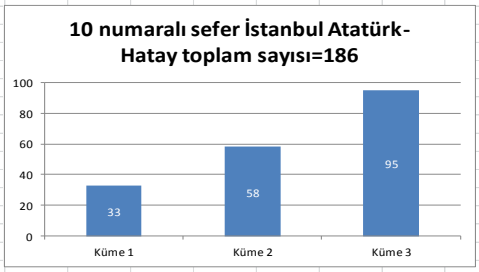


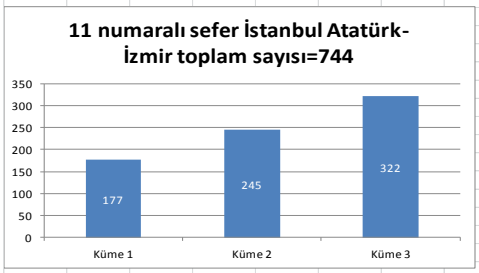


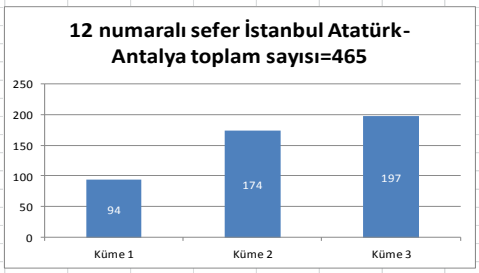


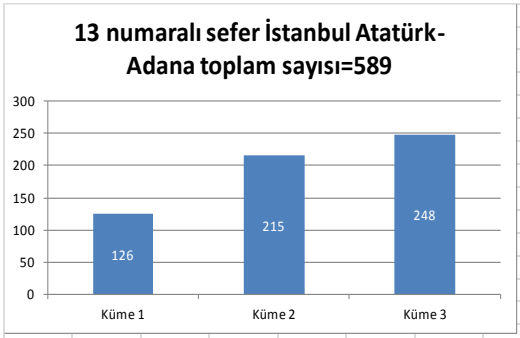


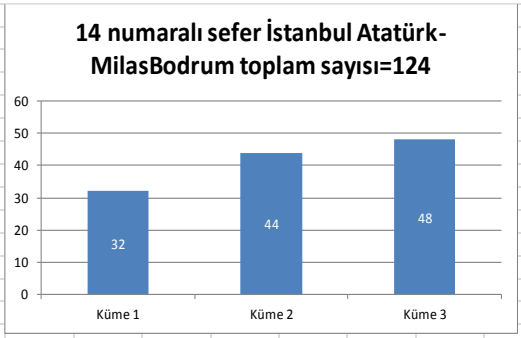


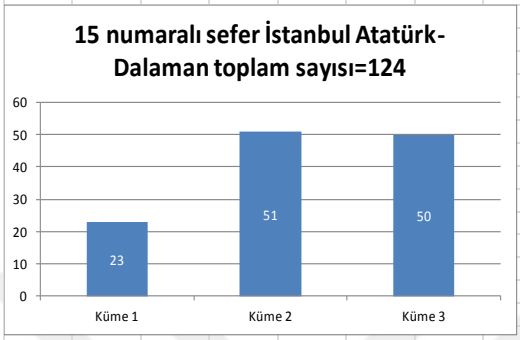


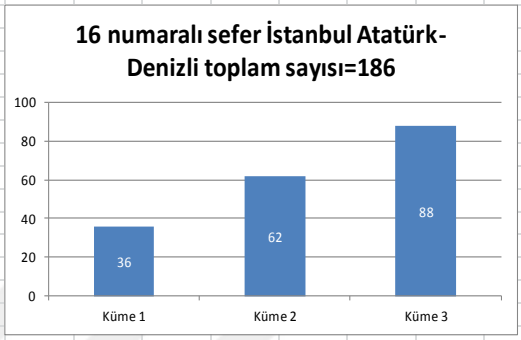


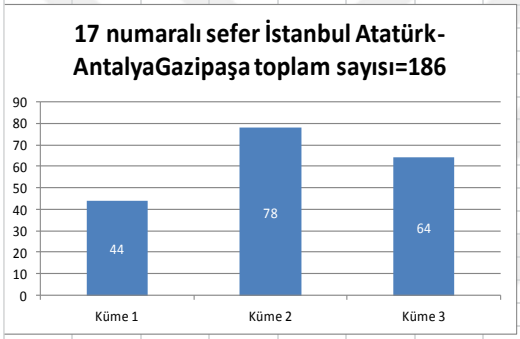


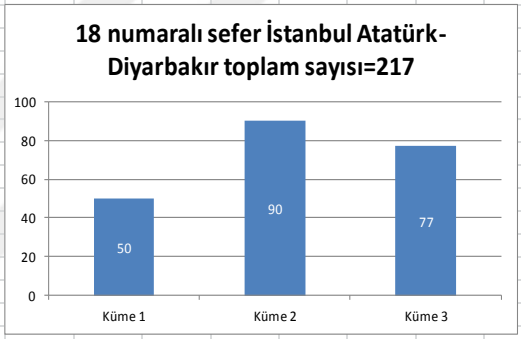

13 numaralı sefer İstanbul Atatürk-Adana toplam sayısı=589
300
250
200
150
100
50
0
126
215
248
Küme 1
Küme 2
Küme 3

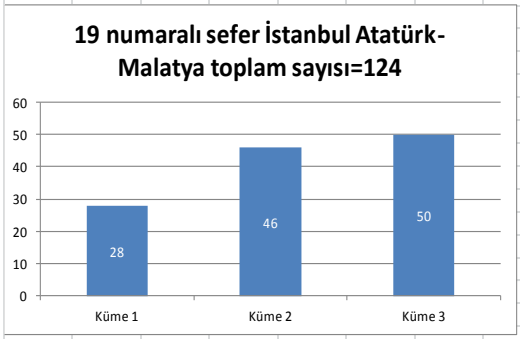

14 numaralı sefer İstanbul Atatürk-MilasBodrum toplam sayısı=124
60
50
40
30
20
10
0
32
44
48
Küme 1
Küme 2
Küme 3

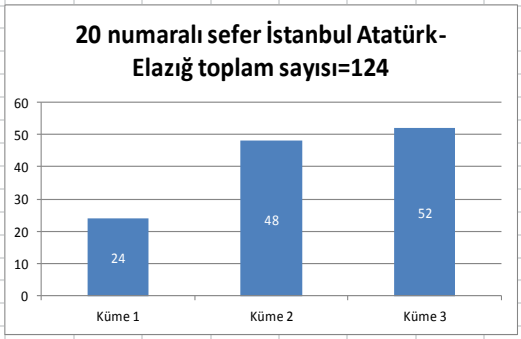

15 numaralı sefer İstanbul Atatürk-Dalaman toplam sayısı=124
60
50
40
30
20
10
0
23
51
50
Küme 1
Küme 2
Küme 3

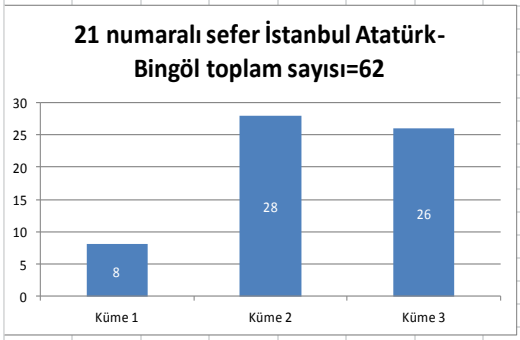

16 numaralı sefer İstanbul Atatürk-Denizli toplam sayısı=186
100
80
60
40
20
0
36
62
88
Küme 1
Küme 2
Küme 3

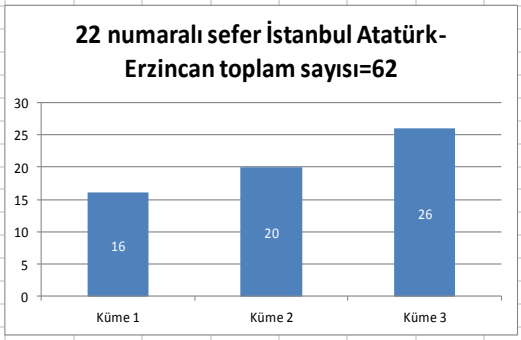

17 numaralı sefer İstanbul Atatürk-AntalyaGazipaşa toplam sayısı=186
90
80
70
60
50
40
30
20
10
0
44
78
64
Küme 1
Küme 2
Küme 3

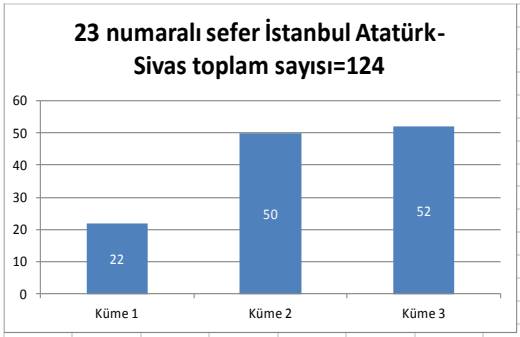

18 numaralı sefer İstanbul Atatürk-Diyarbakır toplam sayısı=217
100
80
60
40
20
0
50
90
77
Küme 1
Küme 2
Küme 3

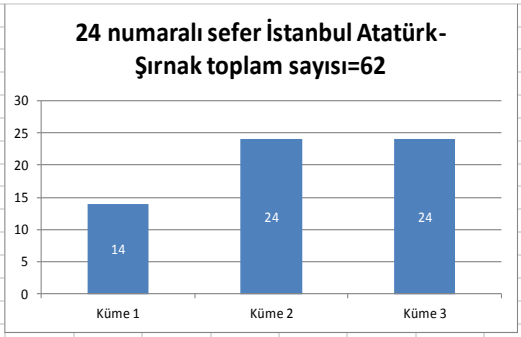

19 numaralı sefer İstanbul Atatürk-Malatya toplam sayısı=124
60
50
40
30
20
10
0
28
46
50
Küme 1
Küme 2
Küme 3

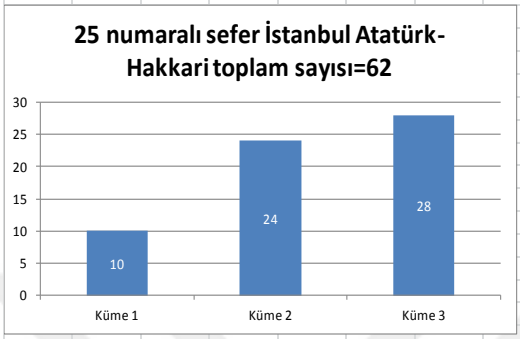

20 numaralı sefer İstanbul Atatürk-Elazığ toplam sayısı=124
60
50
40
30
20
10
0
24
48
52
Küme 1
Küme 2
Küme 3

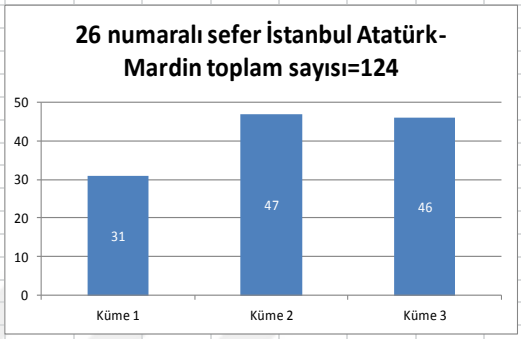

21 numaralı sefer İstanbul Atatürk-Bingöl toplam sayısı=62
30
25
20
15
10
5
0
8
28
26
Küme 1
Küme 2
Küme 3

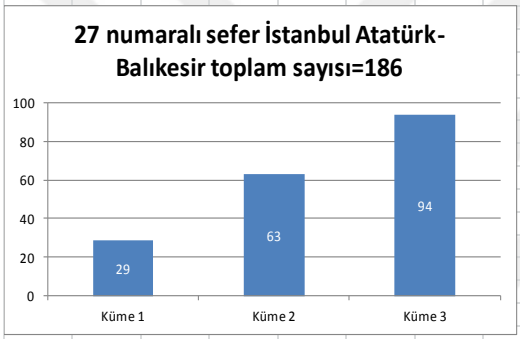

22 numaralı sefer İstanbul Atatürk-Erzincan toplam sayısı=62
30
25
20
15
10
5
0
16
20
26
Küme 1
Küme 2
Küme 3

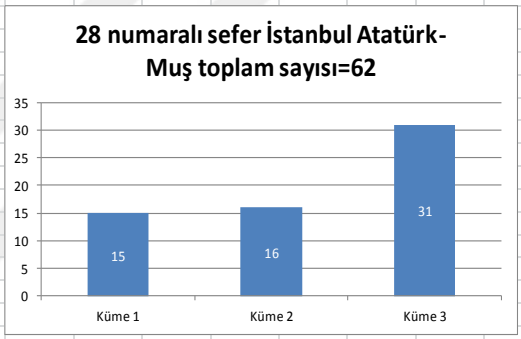

23 numaralı sefer İstanbul Atatürk-
Sivas toplam sayısı=124
60
50
40
30
20
10
0
22
50
52
Küme 1
Küme 2
Küme 3

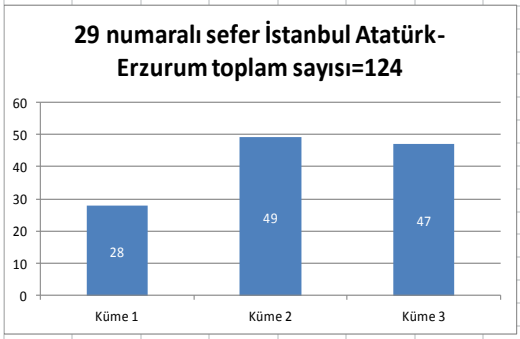

24 numaralı sefer İstanbul Atatürk-
Şırnak toplam sayısı=62
30
25
20
15
10
5
0
14
24
24
Küme 1
Küme 2
Küme 3

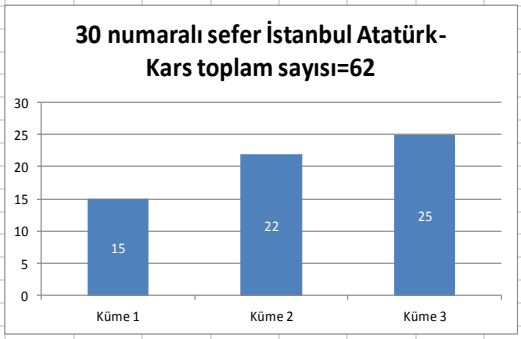

25 numaralı sefer İstanbul Atatürk-
Hakkari toplam sayısı=62
30
25
20
15
10
5
0
10
24
28
Küme 1
Küme 2
Küme 3

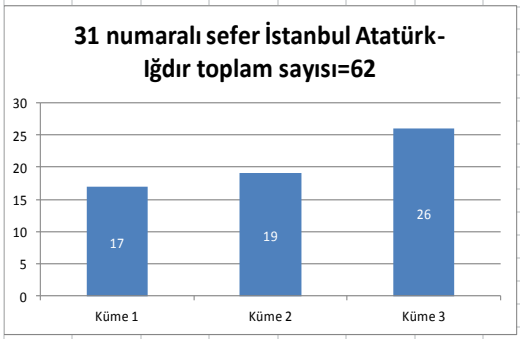

26 numaralı sefer İstanbul Atatürk-
Mardin toplam sayısı=124
50
40
30
20
10
0
31
47
46
Küme 1
Küme 2
Küme 3

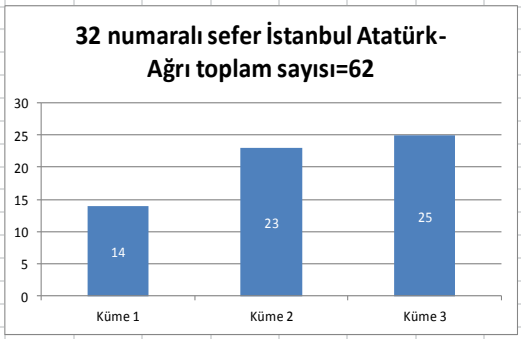

27 numaralı sefer İstanbul Atatürk-
Balıkesir toplam sayısı=186
100
80
60
40
20
0
29
63
94
Küme 1
Küme 2
Küme 3

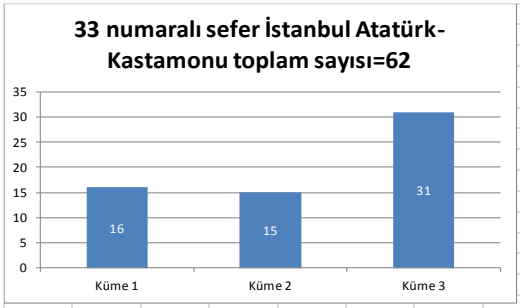

28 numaralı sefer İstanbul Atatürk-
Muş toplam sayısı=62
35
30
25
20
15
10
5
0
15
16
31
Küme 1
Küme 2
Küme 3

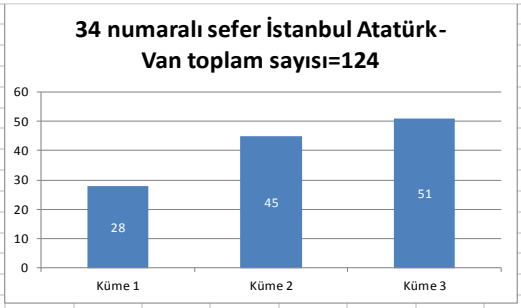

29 numaralı sefer İstanbul Atatürk-
Erzurum toplam sayısı=124
60
50
40
30
20
10
0
28
49
47
Küme 1
Küme 2
Küme 3

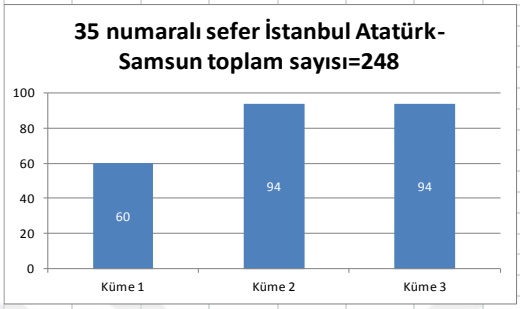

30 numaralı sefer İstanbul Atatürk-
Kars toplam sayısı=62
30
25
20
15
10
5
0
15
22
25
Küme 1
Küme 2
Küme 3

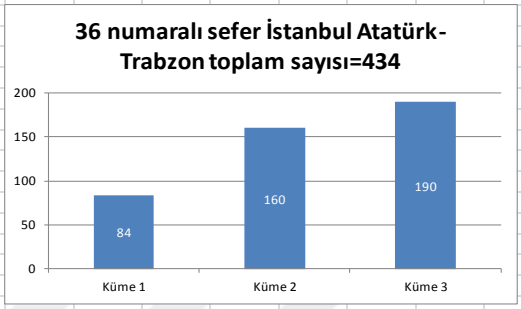

31 numaralı sefer İstanbul Atatürk-
Iğdır toplam sayısı=62
30
25
20
15
10
5
0
17
19
26
Küme 1
Küme 2
Küme 3

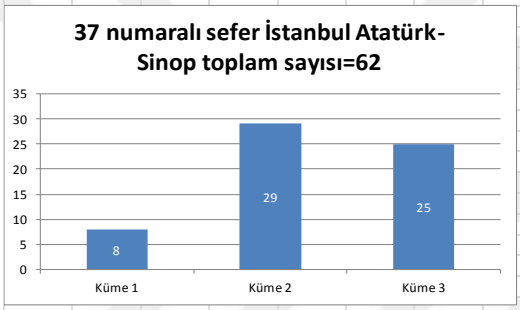

32 numaralı sefer İstanbul Atatürk-
Ağrı toplam sayısı=62
30
25
20
15
10
5
0
14
23
25
Küme 1
Küme 2
Küme 3

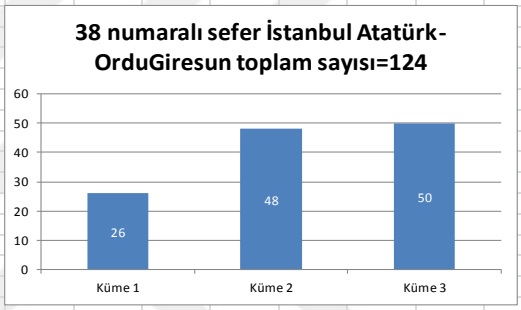

33 numaralı sefer İstanbul Atatürk-Kastamonu toplam sayısı=62
35
30
25
20
15
10
5
0
16
15
31
Küme 1
Küme 2
Küme 3

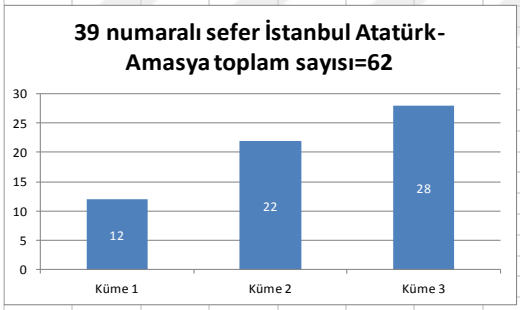

34 numaralı sefer İstanbul Atatürk-Van toplam sayısı=124
60
50
40
30
20
10
0
28
45
51
Küme 1
Küme 2
Küme 3

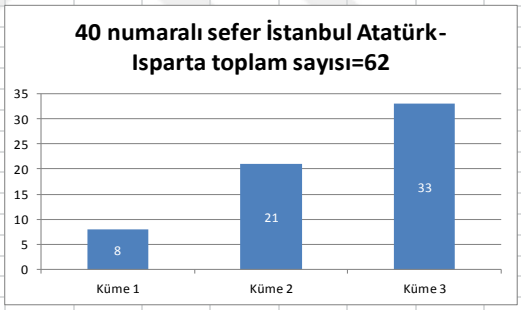

35 numaralı sefer İstanbul Atatürk-Samsun toplam sayısı=248
100
80
60
40
20
0
60
94
94
Küme 1
Küme 2
Küme 3

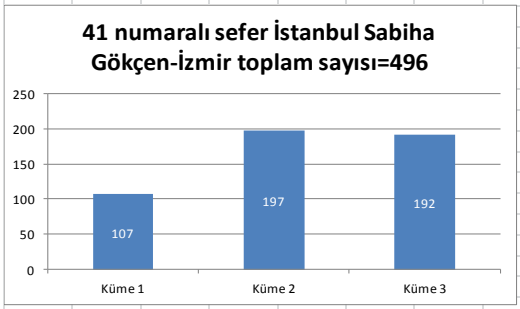

36 numaralı sefer İstanbul Atatürk-Trabzon toplam sayısı=434
200
150
100
50
0
84
160
190
Küme 1
Küme 2
Küme 3

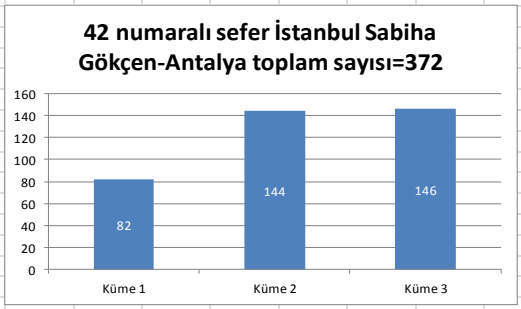

37 numaralı sefer İstanbul Atatürk-Sinop toplam sayısı=62
35
30
25
20
15
10
5
0
8
29
25
Küme 1
Küme 2
Küme 3

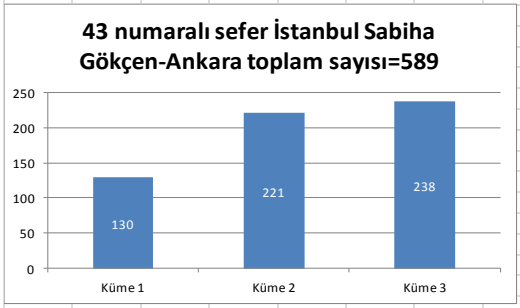

38 numaralı sefer İstanbul Atatürk-OrduGiresun toplam sayısı=124
60
50
40
30
20
10
0
26
48
50
Küme 1
Küme 2
Küme 3


39 numaralı sefer İstanbul Atatürk-Amasya toplam sayısı=62
30
25
20
15
10
5
0
12
22
28
Küme 1
Küme 2
Küme 3

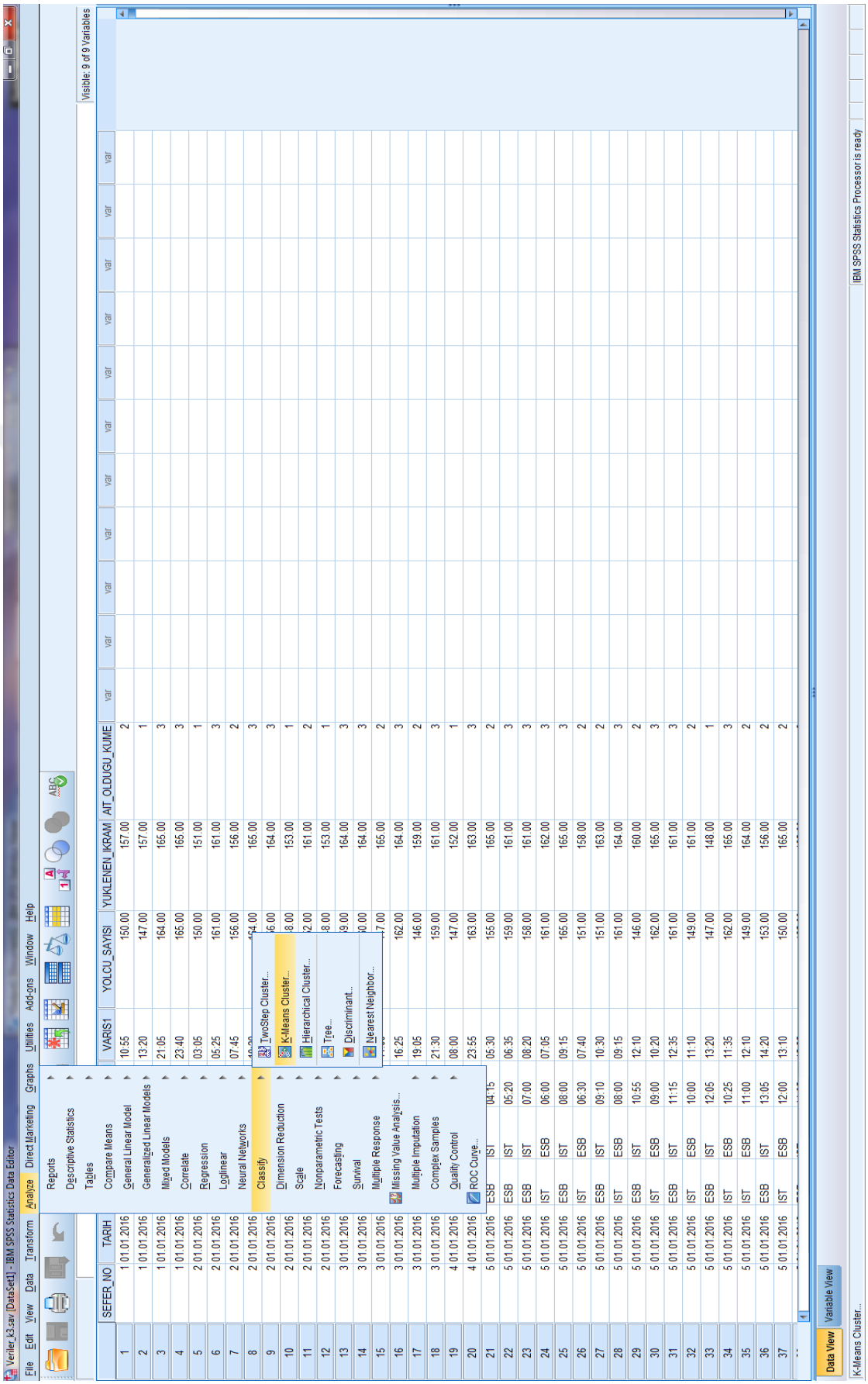

40 numaralı sefer İstanbul Atatürk-Isparta toplam sayısı=62
35
30
25
20
15
10
5
0
8
21
33
Küme 1
Küme 2
Küme 3

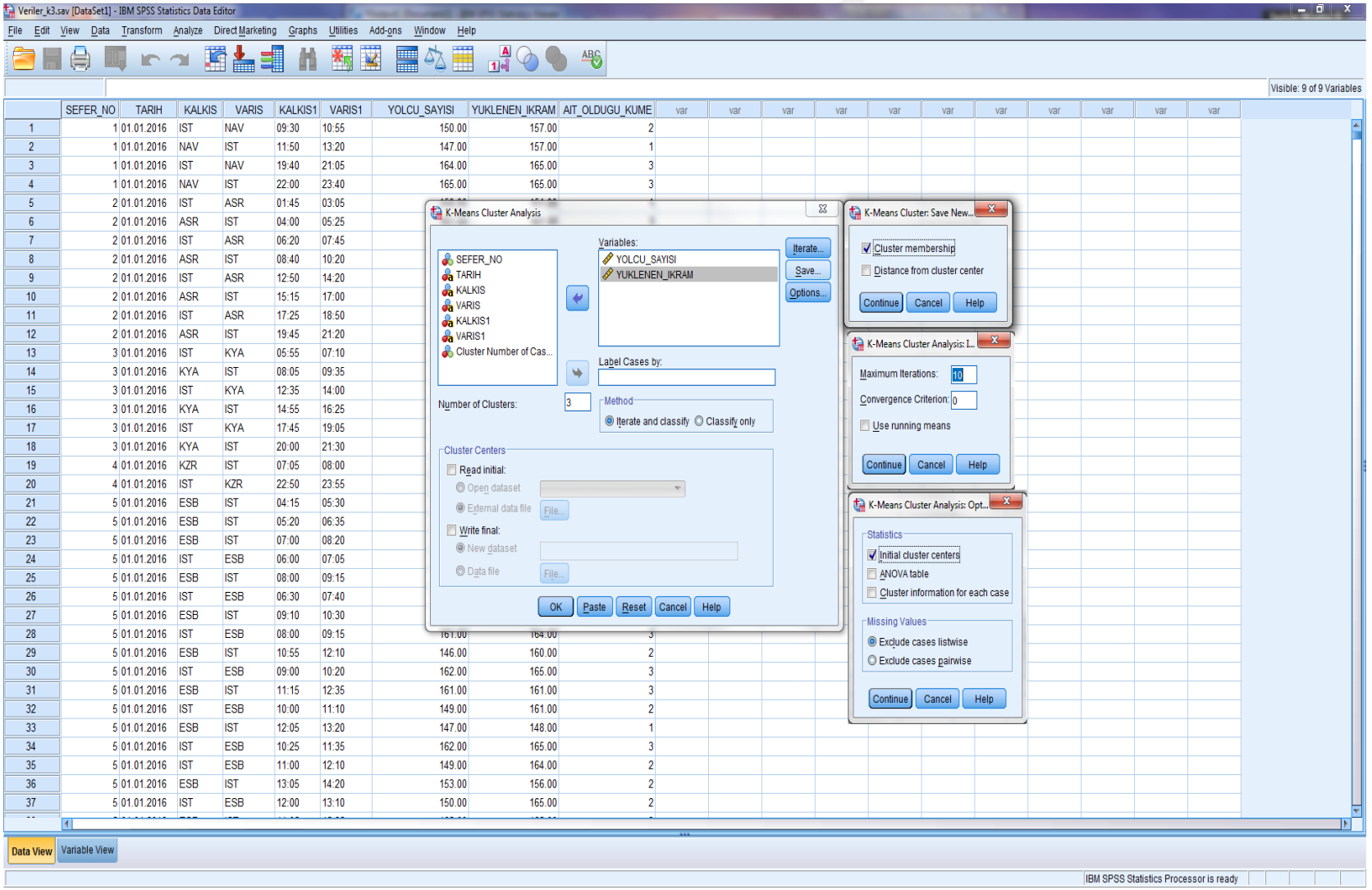

41 numaralı sefer İstanbul Sabiha Gökçen-İzmir toplam sayısı=496
250
200
150
100
50
0
107
197
192
Küme 1
Küme 2
Küme 3


42 numaralı sefer İstanbul Sabiha Gökçen-Antalya toplam sayısı=372
160
140
120
100
80
60
40
20
0
82
144
146
Küme 1
Küme 2
Küme 3


43 numaralı sefer İstanbul Sabiha Gökçen-Ankara toplam sayısı=589
250
200
150
100
50
0
130
221
238
Küme 1
Küme 2
Küme 3

Ek 6. SPSS Ekran Görüntüleri




Veriler_k3.sav [DataSet1] - IBM SPSS Statistics Data Editor
File Edit View Data Transform Analyze Direct Marketing Graphs Utilities Add-ons Window Help
Visible: 9 of 9 Variables
SEFER_NO TARIH KALKIS VARIS KALKIS1 VARIS1 YOLCU_SAYISI YUKLENEN_IKRAM AIT_OLDUGU_KUME
1 1 01.01.2016 IST NAV 09:30 10:55 150.00 157.00 2
2 1 01.01.2016 NAV IST 11:50 13:20 147.00 157.00 1
3 1 01.01.2016 IST NAV 19:40 21:05 164.00 165.00 3
4 1 01.01.2016 NAV IST 22:00 23:40 165.00 165.00 3
5 2 01.01.2016 IST ASR 01:45 03:05
6 2 01.01.2016 ASR IST 04:00 05:25
7 2 01.01.2016 IST ASR 06:20 07:45
8 2 01.01.2016 ASR IST 08:40 10:20
9 2 01.01.2016 IST ASR 12:50 14:20
10 2 01.01.2016 ASR IST 15:15 17:00
11 2 01.01.2016 IST ASR 17:25 18:50
12 2 01.01.2016 ASR IST 19:45 21:20
13 3 01.01.2016 IST KYA 05:55 07:10
14 3 01.01.2016 KYA IST 08:05 09:35
15 3 01.01.2016 IST KYA 12:35 14:00
16 3 01.01.2016 KYA IST 14:55 16:25
17 3 01.01.2016 IST KYA 17:45 19:05
18 3 01.01.2016 KYA IST 20:00 21:30
19 4 01.01.2016 KZR IST 07:05 08:00
20 4 01.01.2016 IST KZR 22:50 23:55
21 5 01.01.2016 ESB IST 04:15 05:30
22 5 01.01.2016 ESB IST 05:20 06:35
23 5 01.01.2016 ESB IST 07:00 08:20
24 5 01.01.2016 IST ESB 06:00 07:05
25 5 01.01.2016 ESB IST 08:00 09:15
26 5 01.01.2016 IST ESB 06:30 07:40
27 5 01.01.2016 ESB IST 09:10 10:30
28 5 01.01.2016 IST ESB 08:00 09:15 161.00 164.00 3
29 5 01.01.2016 ESB IST 10:55 12:10 146.00 160.00 2
30 5 01.01.2016 IST ESB 09:00 10:20 162.00 165.00 3
31 5 01.01.2016 ESB IST 11:15 12:35 161.00 161.00 3
32 5 01.01.2016 IST ESB 10:00 11:10 149.00 161.00 2
33 5 01.01.2016 ESB IST 12:05 13:20 147.00 148.00 1
34 5 01.01.2016 IST ESB 10:25 11:35 162.00 165.00 3
35 5 01.01.2016 IST ESB 11:00 12:10 149.00 164.00 2
36 5 01.01.2016 ESB IST 13:05 14:20 153.00 156.00 2
37 5 01.01.2016 IST ESB 12:00 13:10 150.00 165.00 2
K-Means Cluster Analysis
SEFER_NO
TARIH
KALKIS
VARIS
KALKIS1
VARIS1
Cluster Number of Cas...
Variables:
YOLCU_SAYISI
YUKLENEN_IKRAM
Iterate...
Save...
Options...
Label Cases by:
Number of Clusters: 3
Method
Iterate and classify
Classify only
Cluster Centers
Read initial:
Open dataset
External data file
File...
Write final:
New dataset
Data file
File...
OK
Paste
Reset
Cancel
Help
K-Means Cluster: Save New...
Cluster membership
Distance from cluster center
Continue
Cancel
Help
K-Means Cluster Analysis: I...
Maximum Iterations: 10
Convergence Criterion: 0
Use running means
Continue
Cancel
Help
K-Means Cluster Analysis: Opt...
Statistics
Initial cluster centers
ANOVA table
Cluster information for each case
Missing Values
Exclude cases listwise
Exclude cases pairwise
Continue
Cancel
Help
Data View
Variable View
IBM SPSS Statistics Processor is ready

ÖZ GEÇMİŞ

Eyyüp Burak Levent

Eğitim

Adana Ticaret Odası Anadolu Lisesi, 2006

Yıldız Teknik Üniversitesi, Lisans, Bilgisayar Mühendisliği, 2012

Yıldız Teknik Üniversitesi, Yüksek Lisans, İşletme Yönetimi, 2016

Kurs ve Seminerler

CEH (Certified Ethical Hacker) Netron Bilişim Akademisi, 2009

Oracle Java Yaz Okulu, 2011

Age of Globalization from The University of Texas on Edx, 2013