T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

VERİ MADENCİLİĞİNDE APRIORI ALGORİTMASININ SINAV VERİLERİ ÜZERİNDE UYGULANMASI

MUHAMMED EMİN EKER

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

SAMSUN
2016

TEZ ONAYI

Muhammed Emin EKER tarafından hazırlanan "Veri Madenciliğinde Apriori Algoritmasının Sınav Verileri Üzerinde Uygulanması" adlı tez çalışması 27/10/2016 tarihinde aşağıdaki jüri tarafından Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans Tezi olarak kabul edilmiştir.

Danışman Yrd. Doç. Dr. Recai OKTAŞ
Bilgisayar Mühendisliği Anabilim Dalı

Jüri Üyeleri

Başkan Yrd. Doç. Dr. Gökhan KAYHAN
Ondokuz Mayıs Üniversitesi
Bilgisayar Mühendisliği Anabilim Dalı

Üye Yrd. Doç. Dr. Recai OKTAŞ
Ondokuz Mayıs Üniversitesi
Bilgisayar Mühendisliği Anabilim Dalı

Üye Yrd. Doç. Dr. Kerem ERZURUMLU
Ordu Üniversitesi
Teknik Bilimler Meslek Yüksek Okulu

Yukarıdaki sonucu onaylarım. .../.../2016

Prof. Dr. Bahtiyar ÖZTÜRK
Enstitü Müdürü

ETİK BEYAN

Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez içindeki bütün bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davrandığımı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim.

27/10/2016

Muhammed Emin EKER

ÖZET

Yüksek Lisans Tezi

VERİ MADENCİLİĞİNDE APRIORI ALGORİTMASININ SINAV VERİLERİ ÜZERİNDE UYGULANMASI

Muhammed Emin Eker

Ondokuz Mayıs Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Recai Oktaş

Günümüz teknolojisi sayesinde büyük miktarlarda veri depolanabilmektedir. Depolanan bu sade veriler bulundukları halde anlamsızdırlar. İstenilen bilgilere erişebilmek için verilerin bir amaç doğrultusunda işlenmesi gerekir. Büyük miktarlardaki verilerin, depolandıkları veri tabanlarında bilgisayar programları aracılığıyla aranıp, bulunan sonuçlar kullanılarak gelecekle ilgili tahmin yapılması işlemlerine veri madenciliği denir. Veri madenciliği işlemleri için birçok yöntem ve algoritma geliştirilmiştir. Bu algoritmalardan biri de Apriori algoritmasıdır. Apriori algoritması, veriler içerisinde birlikte bulunan öğelerin keşfedilmesi için kullanılan ve en çok bilinen birliktelik kuralı algoritmasıdır. Birlikte bulunan öğeleri bulmak için birçok kez verilerin taranması gerekir. Bu taramalar aşamasında Apriori algoritmasının özellikleri kullanılarak aralarında birliktelik ilişkisi olan öğeler bulunur. Bu tez çalışmasında bir eğitim yazılımından elde edilen veriler içerisinden bilgi çıkarılmaya çalışılmıştır. Bilgi çıkarma işlemi sırasında veri madenciliği ile ilgili kavramlar ile özellikle birliktelik kuralları üreten Apriori algoritması detaylı bir şekilde ele alınmıştır. Apriori algoritması bilinen market sepet analizinden farklı bir veri üzerine uygulanmıştır. Uygulama sırasında hazırlanan yazılım, verilerin elde edildiği eğitim yazılımına dâhil edilmiş ve elde edilen sonuçlar bu eğitim yazılımını kullanan kurum ve kişilerin hizmetine sunulmuştur.

Ekim 2016, 62 Sayfa

Anahtar Kelimeler: Veri madenciliği, Apriori algoritması, Birliktelik kuralları, Market sepet analizi, Apriori algoritmasının farklı veri kümelerine uygulanması

ABSTRACT

Master's Thesis

APRIORI ALGORITHM IMPLEMENTATION
ON EXAM DATA IN DATA MINING

Muhammed Emin Eker

Ondokuz Mayıs University
Graduate School of Sciences
Department of Computer Engineering
Supervisor: Asst. Prof. Dr. Recai Oktaş

Due to today's technology large amounts of data can be stored. These plain data are not significant in this state. The data must be processed for a purpose to reach the desired information. Data mining is the process of analyzing from large amounts of data in databases searched through computer programs to making predictions about the future using the obtained results. Several methods and algorithms have been developed for data mining operations. One of these algorithms is Apriori algorithm. Apriori algorithm, used for the discovery of elements in the concomitant data and the most known association rule algorithm. To find items that together the data must be scanned multiple times. On this screening stage, items that are associated relationship is determined with using the features of Apriori algorithm. In this thesis trying to achieve to bring out information with the data obtained from an educational software. During bringing out the information discussed in detail about concepts related to data mining and especially Priori Algorithm produces association rules. In this study Apriori Algorithm applied on different data from known market basket analysis. During the implementation of this study, a software has been created and the obtained data was integrated into the educational software. The results were presented to organizations and individuals that uses the educational software.

October 2016, 62 pages

Keywords: Data mining, Apriori algorithm, Association rules, Market basket analysis, Apriori algorithm applied to different data sets

ÖNSÖZ VE TEŞEKKÜR

Akademik eğitim sürecimin bir üst noktası olan yüksek lisans eğitimlerim boyunca yardım ve desteğini benden esirgemeyen danışman hocam Yrd. Doç. Dr. Recai Oktaş ve Yrd. Doç. Dr. Gökhan Kayhan'a teşekkürü bir borç bilirim.

Beni bugünlere getirmek için hiçbir fedakârlıktan kaçınmayan anneme, babama ve kardeşlerime sonsuz teşekkür ederim.

Yüksek lisans eğitimim boyunca benden desteklerini esirgemeyen çalışma arkadaşlarıma da teşekkür ederim.

Çalışmalarım için bana her türlü desteği gösterdiği için PİVOT Bilgi Teknolojileri Şirketi'ne çalışmamın sonuçlanmasındaki büyük katkılarından dolayı teşekkürü kendime bir borç bilirim.

Ekim 2016, SAMSUN

Muhammed Emin EKER

İÇİNDEKİLER DİZİNİ

SİMGELER VE KISALTMALAR

SİMGELER

C_k	K adet veri içeren aday veri seti
L_k	K adet veri içeren veri seti
X => Y	X'in sağlandığı durumlarda Y'nin de sağlanma durumu
X ≼ Y	X'in, Y'in alt kümesi olma durumu
X ^ Y	X ve Y'in aynı anda bulunma durumu

KISALTMALAR

VM	Veri Madenciliği
VTBK	Veri Tabanı Bilgi Keşfi
SQL	Structured Query Language
ORM	Object to Relational Mapping
Confidence	Kabul edilen minimum güven değeri
Support	Kabul edilen minimum destek değeri

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

1. GİRİŞ

Sürekli gelişen teknoloji ile bilişimde de üst düzey gelişmeler yaşanmaktadır. Bu gelişim sonucunda bilgisayar sistemlerinin teknik kapasiteleri artarken fiyatları da oldukça ucuzlamaktadır. İşlemci ve disklerin fiziksel yapıları küçülürken teknik kapasiteleri de ters orantılı bir şekilde büyümektedir. Bu sayede bilgi sistemleri daha büyük miktardaki verileri saklayabilirken bu verileri daha kısa sürede sunabilmekte ve işleyebilmektedir. Bilgi sistemlerindeki gelişmeler ile bilgisayar ağlarındaki alt yapı kalitesi de artmıştır. Bilgisayarlar arası veri transfer hızlarında üst düzey artışlar söz konusudur. Bu gelişmeler de doğrudan ekonomik yapılanmayı tetiklemektedir. Yeni bir bilgi sistemi geliştirilip piyasaya sürüldüğünde bir önceki sürümlerinin fiyatları ucuzlamaktadır. Ucuzlayan bilgi sistemleri ise kurum ve kuruluşları daha fazla sayısal teknolojiyi kullanmaya itmiştir. Teknolojinin yayılması ise veri depolama ortamlarında sayısal sistemlerin kullanılmasına imkân sağlamaktadır. Veriler artık daha çok sayısal olarak toplanabilir, saklanabilir ve hizmete sunulabilir hale gelmiştir. Bu da ayrıntılı ve doğru bilgiye daha kolay erişimi sağlamaktadır.

Günlük hayatta kullandığımız birçok araç ve gereç de teknolojideki bu gelişmelerden doğrudan etkilenmiştir. Kullandığımız cep telefonları artık sadece bir telefon olmaktan çıkarak, telefon defteri, hesap makinesi, fotoğraf makinesi gibi özelliklerle vazgeçilmez bir ihtiyaç haline gelmiştir. Marketlerde kullanılan yazarkasalar basit bir hesap makinesinden ibaret olup sadece satılan ürünlerin bedelini hesaplayabilir. Günümüzde ise gelişmiş cihazlar yardımıyla yapılan alışverişlerin bütün detayları sayısal veri olarak saklanabilmektedir. Bu alışverişler ile elde bulunan ve satılan binlerce ürün ve müşterilerin ayrıntılı hareket bilgileri saklanıp takip edilebilmektedir. Böylece zaman aralıkları içerisinde müşteri takibi yapılarak işlem hareketlerine göre kampanyalar oluşturulabilir. Bu kampanyalar sayesinde de müşterilerin daha fazla alışveriş yapmaları ve alışverişleri sırasında farklı ürünlere yönelmeleri sağlanabilir.

Veri, bulunduğu saf haliyle bir anlam ifade etmeyebilir. Amaca yönelik olarak üzerinde bir takım işlemler yapılması halinde bilgi elde edilebilir. Veriyi bilgiye çevirmeye veri analizi denmektedir (Akpınar, 2000). Bunun yanında ihtiyaca cevap

verebilmek için veriden elde edilen bilgi olarak da tanımlanabilir. Veri sadece karakterlerden ibaret değil aynı zamanda onların anlamlarıdır. Bu şekilde tanımlanana veriye meta veri denmektedir (Akpınar, 2000).

Örneğin market alışverişleri ele alındığında, tüm market ürünleri ve girdi çıktılar müşteri tabanlı incelenebilir. Böylece müşteri bazında davranış bilgileri elde edilebilir. Bu bilgilerin daha önceden yapılan satışlar ile sonraki aylarda bu satışların nasıl gerçekleşeceği öngörüsü de yapılabilir. Satılan ürünler ile satın alan müşteriler arasında bağlantılar kurularak ürünler kategorilere ayrılır ve bu bilgiler ile yeni bir satış planı oluşturulabilir. Satılabilen ürünlerin çeşitleri her geçen gün artmaktadır. Çeşitlenen ürünlerin ise farklı kategorilerde müşterileri oluşmaktadır. Her iki sayının da artması yapılan alışverişlerden büyük miktarlarda veri oluşması sonucunu doğurmaktadır. Sürekli artan bu verilerin incelenmesi için de yazılımlara ihtiyaç duyulmaktadır. Bu noktada da veri madenciliği tekniklerine ihtiyaç duyulur.

Veri madenciliği tekniklerinden olan Apriori algoritması genelde market alışverişlerinde ürünler arası ilişkileri bulmakta kullanılır. Bunların yanında tüm tüketim ürünlerinin depolama sistemlerinin optimizasyonunda da kullanılabilir. Genellikle birlikte taşınan ürünlerin yakın raflara yerleştirilmesi, depo içindeki hareketi ve taşıma kapasitesini azaltıcı sonuçlar sağlayacaktır. Bu yöntem restoranlarda servis hızının artırılabilmesi için de çözüm oluşturabilir. Müşterilerin sipariş etme olasılığı yüksek olan ürünleri önceden tahmin ederek hazırlamak veya onlarla ilişkili ürünlerden mönüler oluşturmak gibi çözümler üretilebilir.

Bu örneklerin yanı sıra daha birçok alanda Apriori algoritması kullanılabilmektedir. Apriori algoritması ile yapılan araştırmalar, bildiriler, makaleler yüksek lisans tez çalışmalarından birkaçı aşağıda listelenmiştir. Bu derleme, bu algoritmanın Türkiye'de hangi farklı alanlarda uygulandığını tespit etmek için yapılmıştır.

Ulaş tarafından yapılan bir yüksek lisans tezi çalışmasında, sepet analizi gerçekleştirilmiştir. Büyük süpermarket zinciri olan Gima Türk A.Ş.'nin verileri üzerine Apriori algoritması uygulanmış ve ortaya çıkan sonuçlar incelenmiştir. Ayrıca mal satışları arasındaki ilişkileri bulmak amacıyla da, bileşen analizi ve k-means metotları kullanılmıştır (Ulaş, 1999).

Toprak tarafından yapılan bir çalışmada ilişkisel veri tabanları üzerinde çoklu ilişkisel yapıdaki ortak kuralları bulmayı sağlayan bir uygulama geliştirilmiştir. Uygulama altyapısı olarak ilişkisel veri tabanlarındaki desenleri tanımlayabilen, bu desenleri eklerle geliştirebilen ve bu desenlerin çeşitli ölçmeleri için gerekli sayımları veri tabanından temel yetilerle alan bir yapı kullanılmıştır. Bu altyapı, veri tabanının tanımında yer alan bilgileri kullanarak arama alanının daraltılmasını sağlamıştır. Bu çalışma, Apriori algoritmasını arama alanını daha da küçültmek için kullanarak ve altyapı tarafından desteklenmeyen özyinelemeli desenlerin bulunmasını sağlayarak altyapıya yenilikler getirmiştir. Apriori algoritması her tablo üzerinde sık karşılaşılan desenleri bulmak için kullanılmış ve bu algoritmanın gerekli destek değerini bulma yöntemi değiştirilmiştir. Veri tabanındaki özyinelemeli ilişkileri belirlemek için bir yöntem sunulmuş ve uygulama bu durumlar için tablo kısaltmalarının kullanıldığı bir çözüm sağlamıştır. Veri tabanı alanlarında saklanan sürekli değerleri bölümleyebilmek için eşit derinlik yöntemi kullanılmıştır. Uygulama bir veri madenciliği yarışması olan KDD Cup 2001'den alınan örnek genlerde yer tahmini problemi ile test edilmiş ve ortaya çıkan sonuçlar yarışmayı kazanan yaklaşımın sonuçlarıyla karşılaştırılmıştır (Toprak, 2004).

Sıramkaya tarafından hazırlanan bir uygulamada internet üzerinden ulaşılabilen basın-yayın kaynaklarında yer alan görsel ve metinsel verilerin hızlı ve etkin bir şekilde erişimi ve bu kaynaklardan anlamlı ve önemli bilgilerin çıkarılması hedeflenmiştir. Çalışmalar istihbarat açısından önem taşıyan kişi ve örgütlerle ilgili haberler üzerinde yoğunlaşmıştır. Sunucu bilgisayarda internet üzerinde yer alan haber kaynaklarından toplanmış ve işlenmiş metinsel belgelerden oluşan veri tabanı ile bu bilgileri işleyen uygulama yazılımları bulunmaktadır. Bir arayüz ile kullanıcının bu bilgileri sorgulaması sağlanmıştır. Çalışma, Birliktelik Kural Madenciliği tekniği ile uygulanmıştır. Bu teknik uygulanırken Apriori Algoritması kullanılmıştır. Yapılan veri madenciliği çalışmasında Bulanık Mantık çalışması, kişi-kişi ilişkilerini bulmakta uygulanmıştır. Bu uygulamadaki amaç kullanıcıların aram a yapmak istedikleri kişilerin isimlerini yazarken yapabilecekleri yazım hatalarını elemektir. İsimlerdeki harflerin konumlarının birbirlerine göre uzaklıklarını temel alarak bulanık mantık kurallarının uygulandığı bir algoritma kullanılmıştır (Sıramkaya, 2005).

Toprak tarafından yapılan bir yüksek lisans tezi çalışmasında, yeni bir melez çok ilişkili veri madenciliği tekniği 2005 yılında gerçekleştirilmiştir. Bu çalışmada kavram öğrenme, kavram ile kavramı gerçekleme önkoşulları arasındaki eşleştirme olarak tanımlanmış ve ilişkisel kural madenciliği alanında buluşsal yöntem olarak kullanılan Apriori kuralı örüntü uzayını küçültmek amacı ile kullanılmıştır. Önerilen sistem, kavram örneklerinden ters çözünürlük operatörü kullanılarak genel kavram tanımlarını oluşturan ve bu genel örüntüleri Apriori kuralını temel alan bir operatör yardımı ile özelleştirerek güçlü kavram tanımlamaları elde eden melez bir öğrenme sistemi olarak tanımlanmıştır. Sistemin iki farklı sürümü, üç popüler veri madenciliği problemi için test edilmiş ve sonuçlar önerilen sistemin, en gelişkin ilişkisel veri madenciliği sistemleri ile karşılaştırılabilir durumda olduğunu göstermiştir (Toprak, 2005).

Kılınç tarafından yapılan bir yüksek lisans tezi çalışmasında, yapılan bir çalışmada birliktelik kuralları için bir yöntem sunmuştur. Apriori algoritmasının ürettiği kurallar elenerek bir elektronik firmasında üretim ve mal giriş kalite verileri üzerinde uygulanmıştır. Ortaya çıkarılan kurallar test verileri ile doğrulanmış ve sonuçlar analiz edilmiştir (Kılınç, 2009).

Yıldız tarafından yapılan bir yüksek lisans tezi çalışmasında, gizliliği koruyan bir yaklaşım önermiştir. Ayrıca bu çalışmada Matrix Apriori algoritması üzerinde değişiklikler yapılmış ve sık küme gizleme çerçevesi de geliştirilmiştir (Yıldız, 2010).

Gülen ve Özdemir tarafından yapılan çalışmanın amacı eğitimsel veri madenciliği yöntemleri ile üstün yetenekli öğrencilerin ilgi alanlarını tahmin etmek ve bu öğrencilerin bir arada ilgi duydukları alanları belirlemektir. Araştırmanın çalışma grubunu Ankara'da yer alan Yasemin Karakaya Bilim ve Sanat Merkezi'nde öğrenim gören yaşları 12 ve daha büyük üstün yetenekli öğrenciler oluşturmaktadır. Bu öğrencilerden veriler Akademik Benlik Kavramı Ölçeği, araştırmacılar tarafından geliştirilmiş olan Boş Zamanları Değerlendirme Anketi ve Ebeveyn Veri Toplama Formu ile toplanmıştır. Ayrıca öğrencilerin WISC-R testi ve Temel Kabiliyetler Testi 7-11 sonuçları da araştırma kapsamında kullanılmıştır. Üstün yetenekli öğrencilerin ilgi alanlarını tahmin etmek için 10 sınıflandırma algoritması seçilmiş ve bu algoritmaların doğruluk testi sonuçları karşılaştırılarak problem tanımı için en uygun olan algoritma belirlenmiştir. Seçilen sınıflandırma algoritmasının çıktılarından

yararlanarak ilgi alanları üzerinde etkili olan nitelikler de ortaya çıkarılmıştır. Üstün yetenekli öğrencilerin sıklıkla bir arada ilgi duydukları alanlar Apriori birliktelik algoritması ile tespit edilmiştir. Çalışmada elde edilen eğitimsel veri madenciliği bulguları Bilim ve Sanat Merkezlerinde üstün yetenekli eğitiminin bireysel ihtiyaçlara göre farklılaştırılması ve ders programlarının daha etkin düzenlenmesi gibi konularda pek çok fayda sağlayacaktır (Gülen & Özdemir, 2013).

Karaibrahimoğlu ve Genç tarafından yapılan çalışmada Meram Tıp Fakültesi Onkoloji Hastanesine ait retrospektif çalışma sonucu elde edilen meme kanseri verileri üzerinde Apriori algoritması uygulanmış ve verilerdeki birliktelik örüntüleri ortaya çıkarılmaya çalışılmıştır (Karaibrahimoğlu & Genç, 2014).

Özçakır ve Çamurcu çalışmasında, bir firmanın pastane satış verileri üzerinde veri madenciliği uygulamak için birliktelik kuralları ile bir yazılım tasarlanmıştır. Veri tabanlarında bilgi keşfi sürecindeki işlemler gerçekleştirilmiştir. Veri seçme işlemi ile operasyon veri tabanından uygulama veri tabanına veriler transfer edilmiştir. Veri tabanı içindeki veriler üzerinde veri önişleme ve veri indirgeme süreçleri uygulanarak veri madenciliğine uygun veri seti elde edilmiştir. Tasarlanan yazılımda, Apriori algoritması kullanılmıştır. Uygulanan Apriori algoritması ile farklı zaman dilimi, farklı satış lokasyonu girdi değerleri doğrultusunda birlikte satın alınan ürünler ile ilgili bağıntılar olduğu gözlemlenmiştir. Genelde aynı ürün grubuna ait ürünlerin, en sık birlikte satın alınan ürünler olduğu görülmüştür. Yazılımın özel tasarımının sağladığı imkân ile yazılımının çalışması esnasında algoritmanın her aşaması izlenebilmiştir (Özçakır & Çamurcu, 2007).

Karabatak ve İnce tarafından yapılan çalışmada Veri Madenciliği teknikleri kullanılarak Fırat Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Eğitimi bölümü öğrencilerinin notları kullanılarak öğrenci başarılarının analizi yapılmıştır. Bu analizi yapmak için Veri Madenciliğinde, birliktelik kuralı çıkarım algoritmalarından biri olan Apriori algoritması kullanılmıştır. Çalışmada kullanılan öğrenci notları, F.Ü. Öğrenci İşleri biriminden SQL server veri tabanı ortamında alınarak gerekli dönüşümler yapılmış ve Apriori algoritması uygulanabilecek boolean tipe çevrilmiştir. Bu aşamadan sonra MATLAB ortamında bir program hazırlanarak veri tabanı programa giriş olarak verilmiş ve bilgisayar ortamından otomatik olarak kurallar üretilmiştir (Karabatak & İnce, 2004).

Taş ve Adak çalışmasında Sakarya Üniversitesi Bilgisayar ve Bilişim Bilimleri Fakültesi Bilgisayar Mühendisliği Bölümü'nde öğrenim görmüş/gören öğrencilerin 1999-2012 yılları arasındaki zorunlu donanım ve yazılım stajı kayıtları kullanılarak bölüm öğrencilerinin staj eğilimlerini belirlenmeye çalışılmıştır. Çalışmada veri madenciliği literatüründe kabul görmüş bir süreç olan CRISP-DM Metodolojisinden yararlanılmış; SPSS Clementine veri madenciliği programının Apriori algoritması aracı ile ilgili veriler üzerinde birliktelik kuralları analizi gerçekleştirilmiştir. Elde edilen bulgulardan, bölüm staj sorumlularına, staj yapan bölüm öğrencileri ile ilgili bilgiler verilmeye çalışılarak staj etkinliğini arttırmaya ve bölüm staj politikasını iyileştirmeye yönelik kararlar almalarında destek sağlanmıştır (Taş & Adak, 2014).

Erdem ve Özdağoğlu tarafından yapılan çalışmada belirli bir dönem boyunca Ege Bölgesi'ndeki bir araştırma ve uygulama hastanesinin acil servisine başvuruda bulunan 214 bin hasta verisi ele alınarak, Veri Madenciliğinde sıklıkla kullanılan birliktelik kuralı yöntemiyle, veri setindeki gizli ancak anlamlılık içeren ilişkiler ortaya çıkarılmaya çalışılmıştır. Çalışma sonuçları, bölgesel özellikler taşıyabileceği düşünülen acil servislere hastaların başvuru nedenleri ve hasta profilleri açısından bir fikir vermekte, ayrıca acil servis bölümlerinin yeniden yapılanma çalışmalarına da farklı bir açıdan yol göstererek katkıda bulunmaktadır (Erdem & Özdağoğlu, 2008).

Erpolat tarafından yapılan çalışmada Türkiye'de otomotiv sektöründe faaliyet gösteren bir yetkili servisin müşterilerine ait alış-veriş verileri, Apriori ve FP-Growth Algoritmaları kullanılarak analiz edilmiştir. Böylelikle müşterilerin hangi ürünleri birlikte satın aldıkları gözlemlenmiş ve bu doğrultuda karı artırmaya yönelik uygulanacak kampanya ve promosyonlara yön verilmeye çalışılmıştır (Erpolat, 2012).

2. VERİ MADENCİLİĞİ

Bilgi sistemlerinin hayatımızın her alanına girmesiyle birlikte yapılan tüm işlemler sayısal ortamlarda tutulmaya kayıt altına alınmaya ve hatta paylaşılmaya başlandı. Yaptığımız alışverişlerde birlikte aldığımız ürünler firma veri tabanlarında saklanmaya başlandı. Ticaretin var olduğu her alanda bir de veri tabanı bulunuyor ve yapılan işlemler kayıt altına alınıyor. Hatta belediye ve hastane gibi kamu kurum ve kuruluşlarından aldığımız hizmetler, otoyol ve kavşaklarda bulunan kameralar ile geçiş güzergâhlarımız dahi kayıt altına alınıyor. Bilgi sistemleri sayesinde bu verilere ihtiyaç duyulduğunda ise büyük veri tabanlarından anlık olarak erişim mümkün olabilir. Ama bu veriler tek başlarına bir anlam ifade etmeyebilir. Hayatımızın her anında kaydedilen bu veriler işlenmeyi bekleyen birer maden gibidir. Bu madenlerde birçok bilgi saklanıyor olabilir.

Veri madenciliği özellikle 1990'lı yıllarda barkod sistemlerinin gelişmesiyle birlikte gelişme göstermiştir. Kayıt altına alınan verilerin anlamlı hale getirilebilmesi için üzerinde durulan bir konudur. Bu ve sonraki yıllarda teknolojinin hızla ilerlemesinin ardından elde edilen ve saklanabilen verilerin hızlı artışıyla birlikte veri madenciliği konusu da oldukça fazla mesafe kaydetmiştir. Bu gelişim ise veri madenciliğini tanımlamayı zorlaştırmıştır. En kapsamlı şekilde ifade edilebilecek olursa şu tanım yerinde sayılabilir:

> "Veri madenciliği daha önceden bilinmeyen geçerli ve uygulanabilir bilgilerin geniş veri tabanlarından elde edilmesi ve bu bilgilerin işletme kararları verirken kullanılmasıdır" (Silahtaroğlu, 2014).

Veri madenciliğinde önemli olan daha öncesinde sahip olunmayan veya tahmin edilmeyen bilgilerin elde edilmesidir. Bu sayede işletme kararları alınırken bu bilgilere başvurulur ve bir farklılık ortaya koyulabilir. Örnek vermek gerekirse veri madenciliği dendiğinde akla gelen müşteri-alışveriş ilişkisi olmaktadır. Yapılan alışverişlerde tutulan kayıtlar, alışverişlerde alınan ürünler ve alışveriş yapan kişinin bilgileri bir araya geldiğinde ve bu verilen işlendiğinde ortaya ilginç bilgiler çıkabilmektedir.

2.1. Veri Madenciliğinin Gelişimi

Veri miktarının her geçen gün kat ve kat artıyor olması veri madenciliğinin gelişmesindeki en önemli etkendir. Sayısal verilerin depolanması ile aynı zamanda bu verilerden bilgi çıkarımı ihtiyacı doğmuştur.

Verilerin saklanabilmesi için ise yüksek kapasiteli depolama alanlarına ihtiyaç duyulur. Günümüz teknolojisi ile depolama cihazlarının kapasiteleri yükselirken fiyatları da oldukça düşmüştür. Bunların yanı sıra bu verilerin hızlı bir şekilde işlenmesi için gerekli olan işlemcilerin kapasitelerinin artması da veri işleme işlemini cazip hale getirmiştir.

Bilgi sistemlerindeki bu gelişmeler de bilgisayar ağlarını oldukça güçlendirmiş ve hızlandırmıştır. Artık herkes tarafından bilgiye erişim ve bu bilgilerin depolanması çok kısa bir süre almaktadır. Çeşitli bilgisayar yazılımları ile uzaktaki verilere hızlı bir şekilde erişilip, depolanıp veri madenciliği işlemleri için hazır hale getirilebilir.

Özellikle ticaretin web ortamına taşındığı şu günlerde rekabet daha bir önem kazanmış ve satış işlemleri için tüm teknolojik gelişmeler kullanılmaya başlanmıştır. Yapılan alışverişler ile oluşan verilerin incelenerek ilgili kişilere e-posta ve kısa mesaj ile kampanya bildirimi de bunların birer sonucu olmuştur. Bu sayede firmalar tüm müşterilerine değil kampanyalara ilgi duyabilecek müşterilerine bildirim yaparak maliyet tasarrufu yapabilmektedirler.

2.2. Veri Madenciliğinin Uygulama Alanları

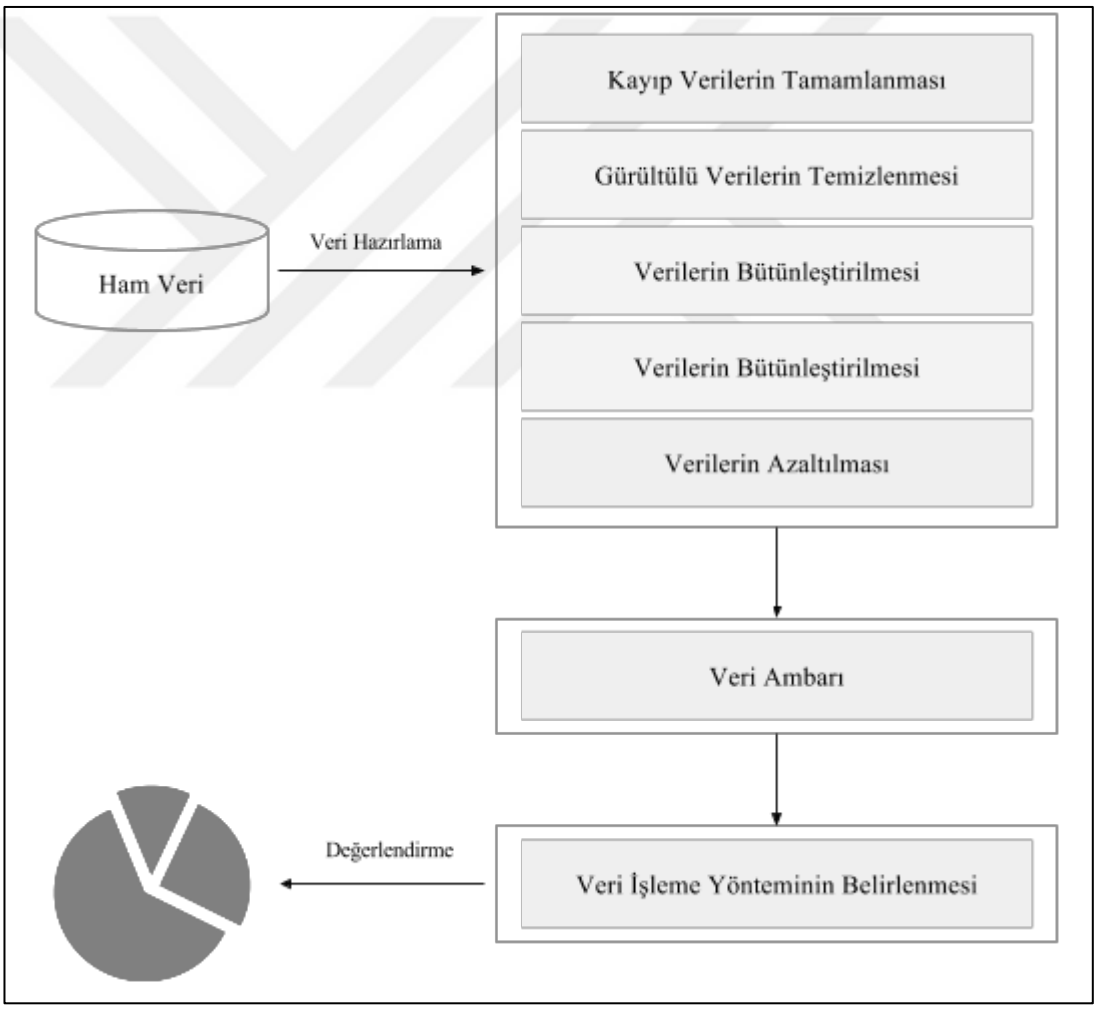
Sağlık, biyoteknoloji, eğitim, sigortacılık, bankacılık gibi daha birçok birbirinden farklı alanda veri madenciliği uygulamalarına rastlamak mümkündür. Veri madenciliği teknikleri geniş veri tabanlarının bulunduğu birçok ortamdan bilgi elde edebilmek için kullanılabilir. Veri madenciliğinin kullanım amaçlarına yönelik Çizelge 2.1'deki gibi özet bir liste yapılabilir.

Çizelge 2.1. Veri madenciliği örnek kullanım alanları ve amaçları

Kullanım Alanı	**Kullanım Amacı**
E-Ticaret	Kaybedilen müşterilerin yeniden kazanılmasına yönelik stratejilerin geliştirilmesi
E-Ticaret	Ürün memnuniyetinin ölçülmesi
Bankacılık	Müşterilerin kredi risklerinin belirlenmesi
Bankacılık	Geçmiş ödeme alışkanlıkları ile yeni ödeme planlarının oluşturulması
Bankacılık	Kredi kartı dolandırıcılıklarının ve usulsüzlüklerin tespitinde
Bankacılık	Kredi taleplerinin değerlendirilmesinde
Bankacılık	Sigorta dolandırıcılıklarının tespitinde
Pazarlama	Mevcut müşteri profillerinin çıkarılması
Pazarlama	Satış noktası veri analizleri
Pazarlama	Alışveriş sepeti analizleri
Pazarlama	Tedarik ve mağaza yerleşim optimizasyonu
Finans	Karlı müşterilerin belirlenerek onlara hitap edep yeni projeler üretilmesi
Borsa	Hisse senedi fiyat tahmini
Borsa	Genel piyasa analizleri
Borsa	Alım-satım stratejilerinin belirlenmesi
İletişim	Kalite ve iyileştirme analizlerinde
İletişim	Hatların yoğunluk tahminlerinde
İletişim	İletişim desenlerinin belirlenmesi
Sağlık	Ürün geliştirme
Sağlık	Üretilen yeni bir ilacın bölgesel etkilerinin belirlenmesi
Sağlık	Tedavi sürecinin belirlenmesinde
Sağlık	Tıbbi teşhis
Sağlık	Semptomlara göre hastalık tespiti

2.3. Veri Madenciliği için Verilerin Hazırlanması

Veri madenciliğinin günümüzde bu kadar yaygın kullanılmasının sebebi saklanabilen verilen büyüklüğü ve bu veri tabanlarına kolay erişim yapılabilmesidir. Bugün herhangi bir mağazadan yapılan alışveriş verilerine ilgili veri tabanlarından anında erişilebilmektedir. Gün, saat, satış personeli alıcı özellikleri gibi birçok veriye ulaşmak mümkündür. Bunların yanı sıra bu bilgilerin tamamen doğru olduğu da kesin olarak söylenemez. Yapılacak bir çalışma için bu verilerin doğru veri olup olmadığı da hemen anlaşılamaz. Bu sebeple ulaşılmak istenen sonuçlar için ilgili verilerin bir hazırlık sürecinden geçmesi gereklidir.



Şekil 2.1. Verilerin hazırlanma süreci

Üzerinde çalışma yapılacak veri tabanında birçok eksik bilgi bulunuyor olabilir. Bu eksikliğe literatürde kayıp veri denmektedir. Eksik verilerin yanı sıra veriler içinde farklı farklı durumlar olması söz konusudur. Elle girilen alış veriş tutarının fazla bir sıfırla girilmesi ve kayıt sırasındaki hatalı bir giriş ile alışveriş miktarının sıfır görünmesi söz konusu olabilir. Bu durumdaki verilere de gürültülü veri denir.

Veri işleme çalışmasından hemen önce veriler hazırlanmalıdır. Hazırlık aşamasında yukarıdaki bahsedilen durumların bir şekilde idare edilmesi gerekir. Bu idare durumları yapılacak çalışmaya ve uygulanacak veriye göre değişebilir. Bu işlemler genel olarak aşağıdaki başlıklar ile ifade edilebilir.

2.3.1. Kayıp verilerin tamamlanması

Kayıp veriler tamamlanması için 5 temel işlemden bahsedilebilir. Bunlar;

2.3.1.1. Kayıp bir veriye sahip kaydı veri tabanından silmek

Kayıp veriye sahip olan kayıtlar veri tabanı içerisinde çok küçük bir orana denk geliyorsa bu kayıtlar veri tabanından kaldırılabilir. Eğer birçok kayıt bu durumda ise o zaman bu yöntem tercih edilmez.

2.3.1.2. Kayıp verileri teker teker el ile tamamlamak

Üzerinde çalışılacak olan veri tabanı küçükse, eksik verilere kolayca ulaşılıp doldurmak mümkünse ve bu veriler kritik öneme sahipse verileri elle tamamlama yöntemi uygulanabilir.

2.3.1.3. Tüm kayıp verilere aynı değeri girmek

Kayıp verilere aynı değeri girmek için dikkat edilmesi gereken bazı durumlar vardır. İlgili alanın türü ve çeşitliliği göz önünde bulundurulmalıdır. Çünkü bir kişinin yaş değerine aynı değeri girmek ile cinsiyet sütununa aynı değeri girmek çalışma sonucunda birbirinden farklı sonuçlar doğurabilir. Yaş değeri ortalama 0-80 arasında değişebilecekken, cinsiyet sadece iki değer alabilmektedir. Erkek için E kadın için K harfinin bulunduğu ilgili sütuna değerin olmadığını ifade edebilecek bir Y harfi girilebilir. Çalışma sonucundaki çıkarımlarda bu Y harfinin önemli bir yer tuttuğu görülebilir. Bu gibi oluşabilecek durumlar için girilecek değer dikkatli seçilmelidir.

2.3.1.4. Tüm kayıp olan verilere ilgili sütunun ortalama değerini girmek

Kayıp olan veriler için ortalama değer girme yönteminde tüm sütunun değil sadece ilgili eksik kayıtların ait olduğu belli bir sınıfın ortalaması seçilebilir. Basit bir örnek ile anlatılacak olursa maaş verisi boş olan bir kaydın ortalama bir değer ile doldurulması gerekebilir. Bu durumda kişinin hangi meslek kolunda çalıştığı önemli bir kıstastır. Eğer tüm meslek kollarındaki ortalama değer bu alana girilecek olursa aslında yine sabit bir değer ile tüm alanların doldurulması söz konusu oluyor. Ancak meslek grubunun aldığı maaşların ortalama değerleri belirlendiğinde birbirinden farklı değerler ile kayıp veriler doldurulmuş olur.

2.3.1.5. Kayıp verilerin sahip olduğu diğer bilgiler ile tahmin edilmesi

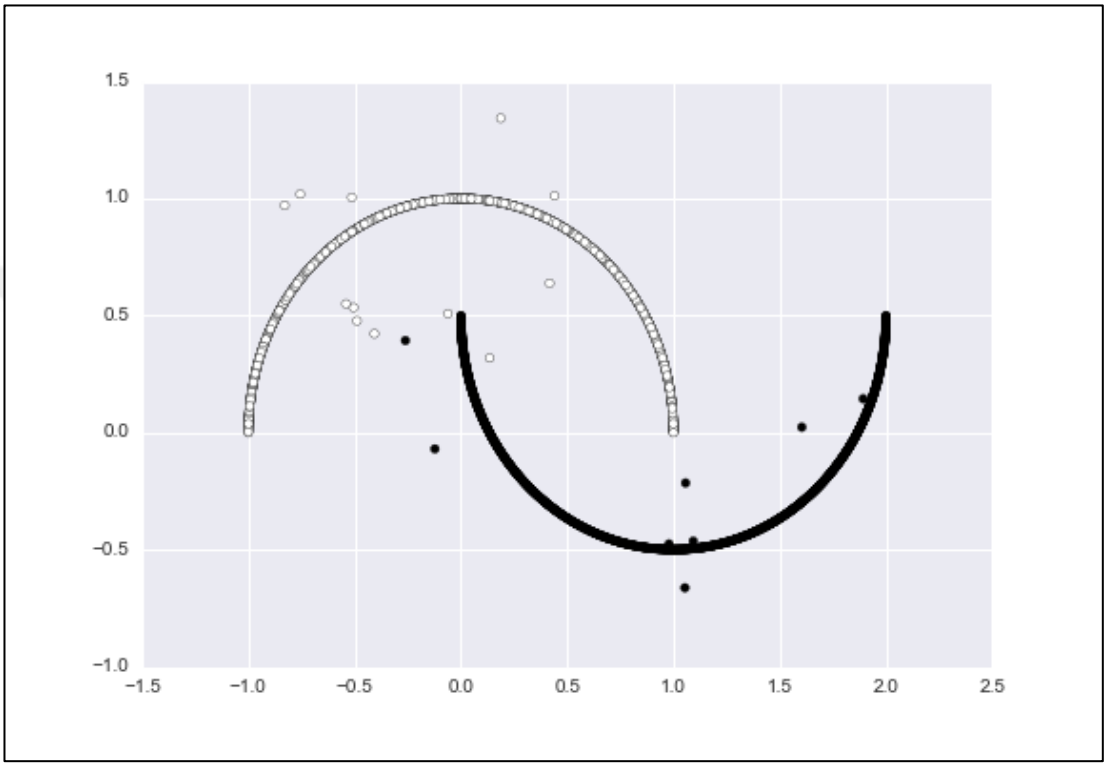
Kayıp olan veriler ait oldukları sınıfın ortalama değerlerine göre tamamlanabileceği gibi yine ait oldukları sınıfın değerlerine uygulanacak farklı bir yöntem ile de hesaplanabilir. Kullanılabilecek bu yöntemler veri türüne göre farklılık gösterir.

Örneğin maaş verisi eksik olan bir kişi için ait olduğu sınıf önemlidir. Ait olduğu sınıfın ise farklı özelliklerine ihtiyaç duyulabilir. İlgili sınıfta işe yeni başlamış bir kişinin aldığı ücret ile uzun yıllardır bu işi yapan bir kişinin aldığı ücret bir olmaz. İşte burada kişinin yaşı veya ilgili meslekteki çalışma yılı esas alınabilir. Bu hesaplama şekline doğrusal interpolasyon denmektedir. Bu ve bunun gibi farklı hesaplama yöntemleri de mevcuttur. Bu hesaplamalar sonucunda oluşacak veriler elbette ki tamamen saklı veriler olmayabilir. Ancak yaklaşık bir değere sahip olduğu için bu değerler ile kayıp değerler doldurulabilir.

2.3.2. Gürültülü Verilerin Temizlenmesi

Eksik olan kayıp verilerin doldurulmasının yanında bir de gürültülü olarak adlandırılan verilerin temizlenmesi gerekmektedir. Gürültülü veriler hatalı girilmiş veriler olabilir. Bu veriler yapılacak çalışmayı etkileyeceğinden bir şekilde kaldırılmaları veya düzeltilmeleri gerekir. Bir verinin gürültülü olup olmadığı temizleme işlemi için kullanılacak yöntemlerle bulunur. Kutulama, sınırlar yardımıyla düzleştirme, kümele gibi istatistiki yöntemler en bilinen temizleme yöntemleridir.

Örneğin kümeleme yöntemiyle verilerin temizlenmesi işlemi için veriler üzerine kümeleme algoritmaları kullanılabilir. Uygulanan yöntem sonucunda veriler kümelere ayrılırlar. Küme dışında kalan veriler ise temizlenmesi gereken verileri gösterir. Bu veriler ilgili kümelerin özelliklerine göre temizlenebilirler. Küme dışında kalan bir veri en yakınında bulunan kümenin ortalama değeri ile güncellenebilir veya yine en yakınında bulunan kümenin üst sınırı bu verinin yeni değeri olabilir.



Şekil 2.2. Kümeleme sonucunda oluşan gürültülü veri

2.3.3. Verilerin bütünleştirilmesi

Farklı noktalardan elde edilen verilerin bir araya getirilmesinden kaynaklı aynı verilerin farklı şekillerde sembolleştirilmesinden kaynaklanan durumlar ortaya çıkar. Buna şu şekilde örnek verilebilir. Bir veri tabanında kişilerin cinsiyeti 0/1, E/K, Erkek/Kadın veya ERKEK/KADIN şeklinde tutuluyor olabilir. İşte bu gibi farklı şekillerde sembolleştirilen verilerin aynı noktada bütünleşmeleri gerekir. Bu işlem için basit güncelleme yöntemleri kullanılır.

2.3.4. Verilerin dönüştürülmesi

İşlenecek veri tabanında bulunan veriler oldukça fazla çeşitliliğe sahip olabilir. Bu çeşitlilik ilgili algoritmaların yavaş çalışmasına ve çıkan sonuçların çok ayrıntılı olmasına sebep olmaktadır. Örneğin cinsiyet değişkeni için sadece iki durum vardır ve veriler bu iki durum ile sembolize edilir. Ancak söz konusu yaş değişkeni olursa ve insan ömrünü 0-80 aralığında düşünürsek 80 farklı değer yaş değişkenini sembolize etmek gerekir. Bunların yanında bir de maaş değişkeni olsa işler daha da karmaşıklaşır. Çünkü minimum maaş değeri ile maksimum maaş değeri arasında birçok farklı değer söz konusudur. Tabi bir de bu değerlerin küsuratı olunca çok daha fazla değer söz konusu oluyor. İşte bu verilerin kısmen gruplandırılmaları yani dönüştürülmeleri gerekebilir. Bu dönüştürme işlemi yaş değişkeni için genç, orta yaş ve yaşlı şeklinde yapılabilir. Maaş değişkeni için ise düşük, orta ve yüksek şeklinde gruplara ayrılabilir.

Bu dönüştürme işlemleri bazı parametrelere göre yapılmalıdır. Bu parametreler verilerin sahip olduğu diğer özelliklerden oluşur. Örneğin 2500 Türk Lirası kazanan bir mühendis orta gelirli olarak kabul edilebilirken, 2000 Türk Lirasına sahip bir temizlik işçisi yüksek gelirli olarak nitelendirilebilir. İşte maaş konusunu etki eden diğer bir değişken meslek sınıfıdır. İşte bu maaş sınıflandırmalarını yaparken sınıflama algoritmalarından faydalanılabilir.

2.3.5. Verilerin azaltılması

İşlenecek veriler büyükse ve eğer imkân dâhilinde ise verilerde indirgeme yapılabilir. Bunların yanı sıra veri işlemeden önce kontrol edilen verilerden kullanılmayacak olan sütunlar var ise bunların kaldırılması da bir indirgeme işlemidir. İndirgeme işlemi için çeşitli yöntemler mevcuttur.

2.4. Veri Madenciliği Uygulamalarında Karşılaşılan Problemler

Veri madenciliğinde işin gereği olarak kullanılan öğeler var olan verilerdir. Bu verilerin yapılacak çalışmaya uygun olmasının yanı sıra eksiksiz ve düzgün verilerden oluşması gerekir Aksi halde anlamsız sonuçlar hatalı çıkarımlar yapmamıza neden olur. Bu problem temel birkaç başlık içinde tanımlanabilir.

Sınırlı bilgi: Verilerin bulunduğu veri tabanlarının asıl amacı veri madenciliğine hizmet sunmak değildir. Bu yüzden saf ve eksik içeren veriler barındırabilir.

Gürültülü veriler: Verilerde bulunan fakat ait oldukları sınıfın özelliklerine göre aykırı olan verilere gürültülü veri denir. Bu aykırı veriler hatalı bilgi demektir ve istenmeyen sonuçlara yol açabilir.

Veri boyutu ve yapılan işlemler: Üzerinde çalışma yapılan verilerin bulunduğu veri tabanlarının üzerinde sürekli işlem yapılıyor olabilir. Bu yapılan işlemler yeni veri eklenmesi olabileceği gibi daha önceki eklenmiş verilerin güncellenmesi de olabilir. Bu durum bilgi eksikliğine yol açmaktadır. Veri madenciliği yapılacak verilerin üzerinde güncelleme veya yeni kayıt ekleme gibi durumların oluşmaması gerekir. Ancak önceki bilgilerin güncellenmesi yani kısmen eksik bilgilerinin tamamlanması da farklı bir durumu ortaya çıkarmaktadır.

2.5. Veri Madenciliği Yöntemleri

Veri madenciliği işlevi kullanıldıkları konu ve veri türlerine göre değişik yöntemlere ayrılmaktadır. Bu yöntemler sınıflandırma, kümeleme ve birliktelik kuralları çıkarma gibi temel başlıklarda incelenir. Bu yöntemler ise birçok farklı algoritma içerirler.

2.5.1. Sınıflandırma

En çok bilinen veri madenciliği tekniklerinden olan sınıflandırma tekniği kategorik sonuçları tahmin etmek için kullanılır. Bu tekniği uygulayabilmek için, önceden bilinen durum-sonuç ilişkisi ve bu durumlarda ilgili değişkenlerin aldığı değerler gereklidir. Bu değerlere eğitim verisi denir. Veri çalışması yapıldıktan sonra elde edilen sonuçlar belirli bir oranla birlikte bir sonuç vererek sunulur. Söz konusu sonuçlar bir oranla birlikte oluştuğu için diğer ihtimallerinde olabileceği bilinir.

Genç kadınlar küçük; yaşlı ve zengin erkekler büyük ve lüks araba satın alır şeklinde genel bir sonuç elde edilmiştir. Tüm yaşlı kadınlar küçük arabaya binmeyebilir. Tüm zengin erkekler de büyük ve lüks arabaya binmeyebilirler. Ancak bu konuda eldeki tüm verilerin analizinden sonra bu şekilde genel bir yargı ortaya çıkmıştır. Konu ile ilgili çalışma yapılacak veriler öğrenme ve test verisi olarak ikiye ayrılır. Öğrenme kümesi modelin oluşturulmasında, test kümesi ise modelin doğrulanmasında kullanılır.

Uygulama Alanları: Gerçekleştirilen kampanyalara yapılan dönüşler, mevcut müşterilerin hizmet almaktan vazgeçme olasılığı, hava tahmini kredi başvuru ve sonuç riskleri, hastalık ihtimalleri, vb.

2.5.1.1. Karar ağaçları

Sınıflandırma problemlerinde en çok bilinen ve kullanılan algoritmalardan birisidir. Diğer algoritmalara göre uygulaması daha kolaydır. Öncelikle bir ağaç yapısı oluşturulur. Bu ağaç yapısında her dal o sonucu, her yaprak ise o düğümdeki sınıfı temsil eder. Sonra tüm elde veriler bu ağaç yapısı üzerinde uygulanır. Böylece iki adımda sınıflandırma işlemi gerçekleştirilmiş olur. ID3, C4.5, C5 ve SPRINT en bilindik karar ağaçları algoritmalarıdır.

2.5.1.2. İstatistiğe dayalı algoritmalar

Sınıflandırma tekniklerinde öncelikli olarak ilgili sınıflar belirlenir. Bu belirleme yapılırken tamamen bir tahmin söz konusudur. Gelecek yeni verilerin de bu sınıflardan hangisine ait olduğu tamamen olasılık değerleriyle belirlenerek istatistiğe dayalı yöntemlerle sınıflandırma yapılabilir. Bayesyen sınıflandırıcı, Regresyon, ve CHAID en bilindik algoritmalarıdır.

2.5.1.3. Mesafeye dayalı algoritmalar

Sınıflandırma tekniklerinden biri de eldeki verilerin birbirine olan uzaklığını kullanarak gruplama işlemi yapmaktır. Veriler arasındaki mesafeler birbirine olan benzerlikleri ifade etmektedir. Bu yöntemler de mesafeye dayalı algoritmalar olarak tanımlanırlar. K- En yakın komşu ve en küçük mesafe sınıflandırıcısı en bilindik algoritmalarıdır.

2.5.1.4. Yapay sinir ağları

Biyolojik sinir ağlarından esinlenilerek geliştirilen bir sınıflandırma işlemidir. Yapay sinir hücreleri birbirleriyle çeşitli şekillerde bağlantılar içerirler. Bu bağlantılar gerçekleşme noktası olan yapay nöron bağlantıları arasındaki katsayılar hesaplanarak gerçekleştirilir. Oluşan nöronların bir de iç katsayıları mevcuttur. İç katsayıya aktivasyon seviyesi denir. Aktivasyon seviyeleri ve bağlantı katsayıları ile birlikte oluşturulan yapay sinir ağları sayesinde de sınıflandırma işlemi yapılabilir. İleri yayılımlı ve geri yayımlı olarak iki ana gruptan oluşan yöntemleri mevcuttur.

2.5.2. Kümeleme

Eldeki verinin birbirlerine benzeyen elemanlarından yeni gruplar oluşturma işlemine kümeleme işlemi denir. Kümelemenin temel amacı büyük veri yığınlarından amaca yönelik kullanılacak verileri ayırma işlemidir. Böylece üzerinde işlem yapılacak veri azalmış olur. Öncelikle verinin içeriğine, özelliklerine ve işlenme amacına göre oluşacak kümeler tahmin edilir.

Kümeleme algoritmalarının temeli, oluşacak kümelerin birbirleri arasındaki benzerliklerinin en az, kümeler içerisindeki benzerliklerin ise en fazla olmasına dayanır. Sonuç olarak oluşan farklı kümelere ait elemanlar arasındaki benzerlikler azdır. Kümeleme ile sınıflandırma arasındaki en önemli fark, kümelemenin önceden tanımlanmış değerlere dayanıyor olmasıdır. Sınıflandırma daha önceden tanımlanan değişkenler ve bu değişkenlerin aldıkları değerler ile yapılır. Kümelemede ise tanımlı değerler yoktur. Benzerliği tanımlayacak özellikler modeli kuran tarafından tahmin edilir.

Kümeleme işlemi gerektiği durumlarda diğer bir veri madenciliği işlemi öncesinde kullanılabilir. Örneğin hangi kampanyada müşterilerden en iyi tepki alınır şeklinde değerlendirmek yerine, öncelikli olarak müşterilerin belirli kümelere ayrılması ve bunun ardından oluşan her küme için en iyi kampanyanın ne olacağı belirlenebilir.

Uygulama Alanları: Benzer verileri tanımlamak, benzer davranışlar gösteren müşterilerini tanımlamak, ürün grupları oluşturmak, DNA analizleri, hastalık belirtileri analizi.

2.5.2.1. Hiyerarşik yöntemler

Hiyerarşik yöntemler de ağaç yapısını temel alır. Ağaç yapısındaki her dalın alt dalları yani her bir düğümlerin alt düğümleri mevcuttur. Ağacın yapraklarından gövdesine ve gövdesinden yapraklarına doğru iki farklı algoritma çeşidi mevcuttur. Bu yöntemler hemen hemen tüm veriler üzerine uygulanabilir. Bunun temel dayanağı mesafe değerine yani verilerin birbirlerine benzeme ilişkisine bağlı olmasıdır. BIRCH, CHAMELEON, CURE ve SLINK algoritmaları en bilindik algoritmalarıdır.

2.5.2.2. Bölümlemeli yöntemler

Verilen N adet nokta ve verilen K küme sayısına göre kümelere ayırma işlemlerine bölümlemeli yöntemler denir. Bu yöntemleri diğerlerinden ayıran en önemli özellik oluşturulacak küme sayısının, kümeler arası en az ve en fazla uzaklık mesafelerinin tanımlanıyor olmasıdır. Bu değerlerin tanımlanıyor olması algoritmanın akışına bırakılmaması anlamına gelir. Bunun sonucunda da aynı değerler ile farklı sonuçlar bulunması olasıdır. Farklı sonuçlar bulunuyor olması da hangi çözümün en uygun olduğunun kesin olarak söylenememesidir. CLARA, CLARANS, K-Means ve PAM algoritmaları en bilindik algoritmalarıdır.

2.5.2.3. Izgara tabanlı yöntemler

Yüksek miktarda bellek gerektiren büyük veri tabanlarındaki verilerin işlenebilmesi için sonlu sayıdaki kare şeklindeki hücreler kullanılır. Bu sebeple ızgara ismi ile anılırlar. Kullanılan kare sayısı arttıkça hesaplama zamanı da doğru orantılı olarak artar. CLIQUE, STING ve Wave Cluster algoritmaları en bilindik algoritmalarıdır.

2.5.2.4. Model tabanlı yöntemler

Eldeki verilerin matematiksel bir model ile ifade edildiği yöntemlere model tabanlı yöntemler denir. Bu yöntemlerde veriler bir mantık çerçevesinde olasılık değerleri ile veri uzayına yerleştirildikleri düşünülerek veri işlemesi yapılır.

2.5.2.5. Yoğunluk tabanlı yöntemler

Nesnelerin dağılımını bir yoğunluk fonksiyonu aracılığı ile tespit ederek kabul edilen eşik yoğunluk değerini aşan grupları küme olarak adlandıran yöntemlere yoğunluk tabanlı yöntemler nedir. Bu yöntemler gürültülü verilerden etkilenmeme ve az işlem

ile sonuca ulaşılabilir olmasıyla düzgün olmayan veriler üzerinde kullanımı tercih edilir.

2.5.3. Birliktelik Kuralları

Belirli türlerdeki verilerin bir arada bulunma ilişkilerini tanımlayan modele birliktelik kuralı denir. 1990'lı yıllardan itibaren veri toplama ve işleme uygulamalarında gelişmeler sağlanmış ve bu doğrultuda firmalar satış noktalarında otomatik ürün veya müşteri tanıma sistemleri gibi yeni teknolojiler kullanmaya başlamışlardır. Bu noktadaki teknolojik gelişmeler bir satış hareketine ait verilerin toplanmasına ve elektronik ortama aktarılabilmesine olanak tanımıştır. Günümüzde, küçük marketlerden süper marketlere kadar bütün satış noktalarında akıllı satış sistemlerinin kullanımı oldukça yaygındır. Bu satışlar sırasında, işlem tarihi, satın alınan ürünlere ait bilgiler (fiyat, miktar, indirim, ürün kodu vb.) gibi bilgiler elde edilir. Bir takım kuruluşlar, bu tip işlem hareketlerini içeren veri tabanlarını pazarlama stratejilerinin önemli parçalarından biri olarak görmekte ve bu verileri kullanabilmek için çaba harcamaktadırlar.

Market sepeti uygulaması, birliktelik kurallarının kullanıldığı en tipik örnektir. Birliktelik kurallarının çıkarımı için müşterilerin alışverişleri esnasında satın aldıkları ürünler arasındaki bağlantılar incelenerek müşterilerin satın alma alışkanlıkları tespit edilir. Elde edilen birliktelik kuralları, müşterilerin hangi ürünleri bir arada aldıkları bilgisini ortaya çıkarır ve işletme yöneticileri de bu bilgiler ile daha farklı ve etkili satış stratejileri geliştirebilirler. Market alışverişi esnasında müşteriler peynir aldıktan sonra genelde aynı alışverişte zeytin de satın alıyorlarsa, bu iki ürün arasında kuvvetli bir ilişki olduğu bilgisi elde edilir. Elde edilen bu bilgiler sayesinde birlikte satılma olasılığı güçlü olan ürünlerin yanına ek bir ürün daha satılıp satılamayacağı gibi satış temelli düzenlemeler yapılabilir. Bu elde edilen veri sayesinde, bu ürünlere ek ürün satışı yapmak için düzenlemeler yapılabilir. Birliktelik kuralı çıkarımı için en bilindik algoritmalar AIS, SETM, Apriori ve AprioriTid algoritmalarıdır.

2.5.3.1. AIS Algoritması

1993 yılında geniş nesne kümeleri üretmek için tasarlanmış bir algoritmadır. Algoritma veri tabanı nesne isimlerinin alfabetik olarak sıralanmasını temel alarak tasarlanmıştır. Nesne kümeleri hazırlanırken veri tabanı birçok kez taranır. Bu

taramalarda sırası ile oluşturulmuş nesne kümesi, önceki taramalarda geniş oldukları bulunan nesne kümeleri ile karşılaştırılır. Her seferinde ortak elemanlar diğer verilerle birleştirilerek daha geniş nesne kümesi elde edilmeye çalışılır.

2.5.3.2. SETM Algoritması

SETM algoritması diğer algoritmalardan farklı olarak iki parametreli bir yapıya sahiptir. Nesne kümesinin her bir elamanının sahip olduğu bu parametrelerden birisi nesnenin ismi iken diğeri nesnenin değeridir. Nesne kümeleri hazırlanırken veri tabanı birçok kez taranır. AIS algoritmasındaki gibi her oluşturulan nesne kümesi bir önceki ile karşılaştırılır ve ortak elemanlarla yeni nesne kümeleri elde edilir. Ancak bu algoritmada veri tabanı taranırken nesne değerine sahip olan parametre kullanılır.

2.5.3.3. Apriori Algoritması

Apriori algoritmasının temel yaklaşımı k-öğe kümesi, minimum destek şartını sağlıyorsa bu kümenin alt kümeleri de minimum destek şartını sağlar şeklindedir. Nesne kümeleri hazırlanırken veri tabanı birçok kez taranır. Sık geçen öğe kümelerinin bulunmasının amaçlandığı Apriori algoritmasında, birinci taramada bir elemanlı minimum destek şartını sağlayan sık geçen öğe kümeleri bulunur. Sonraki taramalarda önceki taramalardan elde edilen sık geçen öğe kümeleri, yeni ve daha geniş öğe kümelerini üretmek için kullanılır. Tarama sırasında daha geniş olmaya aday nesne kümelerinin destek metriği hesaplanır. Kümeler içerisinde sık bulunan ve minimum destek şartını sağlayan kümeler, aday kümelerden çıkarılır. Sık geçen öğe kümeleri bir sonraki tarama için aday nesne kümesi olurlar. Bu şekilde sık geçen öğe kümesi bulunmayana kadar devam edilir.

2.5.3.4. AprioriTid Algoritması

Algoritmalar sonuca ulaşabilmek için veri üzerinde çok fazla arama işlemi gerçekleştirirler. Apriori algoritmasını literatüre kazandıran Agrawal, veri üzerinde her zaman bu kadar fazla sorgulama işlemi gerekmemesini düşünerek AprioriTid algoritmasını da sunmuşlardır. Apriori algoritmasından farkı ilk destek seviyesi hesaplama aşamasından sonra bir daha tarama işlemi yapmamasıdır.

2.5.3.5. Diğer Algoritmalar

Literatürde birliktelik kuralları çıkarımı yapılabilen birçok algoritma mevcuttur. Ancak veri tabanı taramaları, geniş nesne kümelerinin bir önceki ile ilişkisi gibi yaklaşımlar genel olarak aynıdır. OCD, CARMA, Apriori-Hybrid algoritmaları bunlardan birkaçıdır.

2.6. Birliktelik Kuralları

Teknolojiyi kullanan her kurum ve kuruluş sahip oldukları veri tabanlarında bulunan verilerden anlam çıkarma çabası içerisindedir. Veri tabanlarında miktarı her geçen gün artan veriler arasındaki ilişkiler iş süreçlerinin geliştirilmesi ve sonuçlarından en üst düzeyde fayda sağlanabilmesi için sahipleri açısından oldukça büyük bir öneme sahiptir. Veri tabanındaki verilerin birbirleri ile olan ilişkilerinin ortaya çıkarılmasına ilişki analizi denmektedir. İlişki analizi var olan veriler içerisinde bulunan herhangi bir verinin bir diğer veri veya daha fazla veri ile birlikte olma olasılığını ve bu olasılıkların kurallarını ortaya çıkarır.

İlişki analizi hastalık tespiti, yatırım ve pazarlama stratejisine kadar birçok alanda doğrudan kullanılmaktadır. Örneğin herhangi bir satış noktasından bir ürün satın alınırken bu ürünün yanında diğer bir ürün veya ürünlerin alınması bu ürünler arasındaki bağlantıyı ortaya koyar. Bu bağlantıların elde edilmesi ve elde edilen bilgilerin bir kural olarak tanımlanabilmesi ilişki analizinin konusudur. Elde edilen kurallar ise birliktelik kuralları olarak tanımlanır. Literatürde genel olarak bu şekilde veriler arasında bir ilişki tespit etme işlemlerine pazar sepet analizi denilir. Pazar sepeti analizi müşterilerin alışverişleri sonucunda veri tabanında oluşan veriler aracılığıyla ortaya çıkarılması işlemidir. Müşterilerin alışveriş alışkanlıklarının bu şekilde ortaya çıkarılması, satış noktasının performansının doğrudan etkileyecek kararların alınmasını sağlar. Elde edilen sonuçlarla satış noktası içerisindeki ürünlerin yerleştirme düzeninde değişiklikler yapılabilir. Satılamayan bir ürün kaldırılabilir, tercih edilmesi muhtemel olan yeni bir ürün ise var olan ürünlerin yanına eklenebilir.

2.6.1. Birliktelik Kuralının Matematiksel Gösterimi

Agrawal ve Srikant yayınlarında birliktelik kurallarını matematiksel olarak aşağıdaki şekilde ifade etmiştir (Agrawal & Srikant, 1993):

$I = \{ I_1, I_2, \ldots I_m \}$ bir dizi – nesne kümesi olsun.
$T = \{ t_1, t_2, \ldots t_n \}$ ise veri tabanındaki işlemleri (alışverişi) göstersin.

Her bir t_k'nın alacağı değer 0 veya 1'dir. Eğer $T_k = 0$ ise I_k satın alınmamış, eğer $t_k = 1$ ise I_k satın alınmış demektir. Her bir işlem için veri tabanında ayrı bir kayıt vardır. Şimdi x $X \subseteq I$için X'teki her bir I_k'ya karşılık gelen t_k değeri $t_k = 1$'dir (Silahtaroğlu, 2014).

Bu birliktelik kuralı şu şekilde ifade edilmektedir: X => I_j, X, I'nın bir alt kümesidir. I_j ise I içindeki herhangi bir elemandır ve bu eleman X içinde yer almamaktadır. X => I_j kuralının T için uygun olduğunun söylenebilmesi için belli bir güven seviyesinden söz etmek gerekecektir. Dolayısıyla, T için deki tüm X'lerin ne kadarının I_k'yı sağladığı %c değeriyle ifade edilmelidir. Bu durumda, birliktelik kuralını $0 \leqslant c \leqslant 1$güven seviyesiyle birlikte şöyle ifade edebiliriz. X => Ij | c. Güven seviyesi, kuralın gücünü ifade etmektedir (Silahtaroğlu, 2014).

2.6.2. Destek ve Güven Değerleri

Birliktelik kurallarının çıkarılması sırasında bazı değerlerin temel alınması gerekir. Bunlar destek ve güven değerleridir. Destek seviyesi, T içindeki işlemlerin ne kadarını X'in sağladığıdır. Mevcut veri tabanından birçok kural çıkarılabilir. Ancak bu kuralların tamamı önem arz etmeyebilir. Hazırlanan algoritma, tanımlanacak destek ve güven seviyeleri ile elde edilmeye başlanan kurallarda bu seviyeleri aşma durumlarına göre filtrelenir. Bu şekilde belirli bir seviyenin üzerinde kalan daha anlamlı kurallar elde edilmiş olur. Bu şekilde en küçük destek seviyesini sağlayan nesne kümelerine geniş, diğerlerine küçük nesne kümesi denilir (Agrawal & Srikant, 1993).

Veriler üzerinde bağlantı analizi yapıldıktan sonra ortaya çıkan sonuç genellikle aşağıdaki gibi ifade edilir.

Yaş (kişi, "30 - 40")

→

Satın alır(Kişi, "CEP TELEFONU") [DESTEK = %4, Güven = %17]

Yukarıdaki ifade, 30 - 40 yaş aralığındaki kişilerin cep telefonu almaya yatkın olduğu, bugüne kadar alışveriş yapan kişilerin içinden %4 sinin bu durumda olduğu ve yaşı 30 - 40 arasında değişen müşterilerin %17'sinin cep telefonu aldığını ifade eder.

Yaş (kişi, "30 - 40") ^ Cinsiyet(Kişi, "erkek")

→

Satın alır(Kişi, "CEP TELEFONU")[DESTEK = %3, Güven = %42]

Yukarıdaki ifade, 30 – 40 yaş aralığında ve aynı zamanda erkek olan müşterilerden cep telefonu alanların tüm müşterilere oranının %3 olduğunu ve %60'ının cep telefonu aldığını ifade eder. İlk örnekteki nesne kümesi ikinci örnekteki nesne kümesine göre daha küçüktür. Kural çıkarımı sırasında ilgili destek ve güven seviyesi aşan nesne kümesi ikinci örnekte daha fazladır. Nesne kümesi fazla olan bir başka örnek de şu şekildedir.

Yaş (kişi, "30 - 40") ^ Cinsiyet(Kişi, "erkek") ^ satın alır(Kişi, "CEP TELEFONU")

→

Satın alır(Kişi, "LED TV") [DESTEK = %2, Güven = %65]

Yukarıdaki ifade, 30 – 40 yaş aralığında ve aynı zamanda erkek olan müşterilerin LED TV satın alanların %65'inin aynı zamanda LED TV satın aldığını ifade eder.

2.6.3. Apriori Algoritması

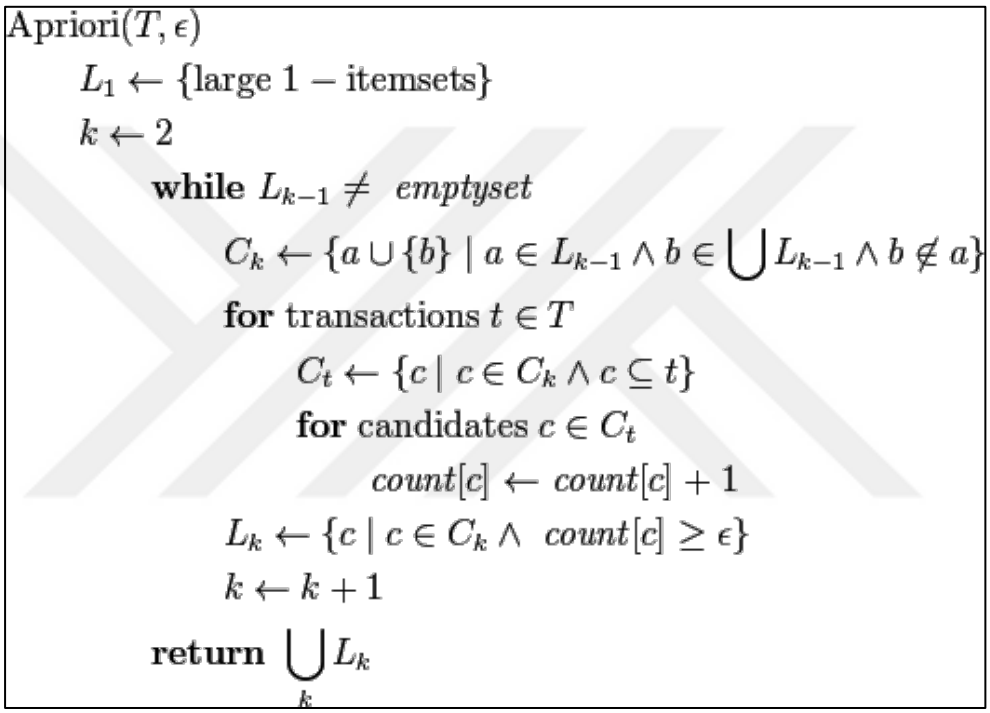
Birliktelik analizlerinin yapılıp ilişki kurallarının ortaya çıkarılması konusunda en çok bilinen ve kullanılan algoritma Apriori algoritmasıdır. İlişki analizini ortaya çıkarmak için tasarlanan algoritmalar veri tabanını birçok kez tararlar. İlişki analizlerinin amacı geniş nesne kümelerinin ortaya çıkarılmasıdır.

Nesne kümeleri ortaya çıkarılmaya başlandığında öncelikle her bir nesnenin destek seviyesi hesaplanır ve tanımlanan destek seviyesi ise karşılaştırılır. Destek seviyesi hesaplanan her bir nesne kümesine aday nesne kümesi, tanımlı destek seviyesini aşan nesneler kümeleri ise geniş nesne kümesi olarak adlandırılır. Hangi aday nesne kümesinin geniş nesne kümesi olduğuna burada karar verilir. Algoritmanın amacı daima en geniş nesne kümelerini bulmaktır. Bu sebeple destek seviyesini aşamayan nesne kümeleri bir sonraki adımda önemsenmezler. Her bir sonraki adımda bir önceki adımda belirlenen geniş nesne kümeleri tüm veriler içerisinde taranır. Bu adımlar en geniş nesne kümesinden daha genişi bulunamayana kadar devam eder.

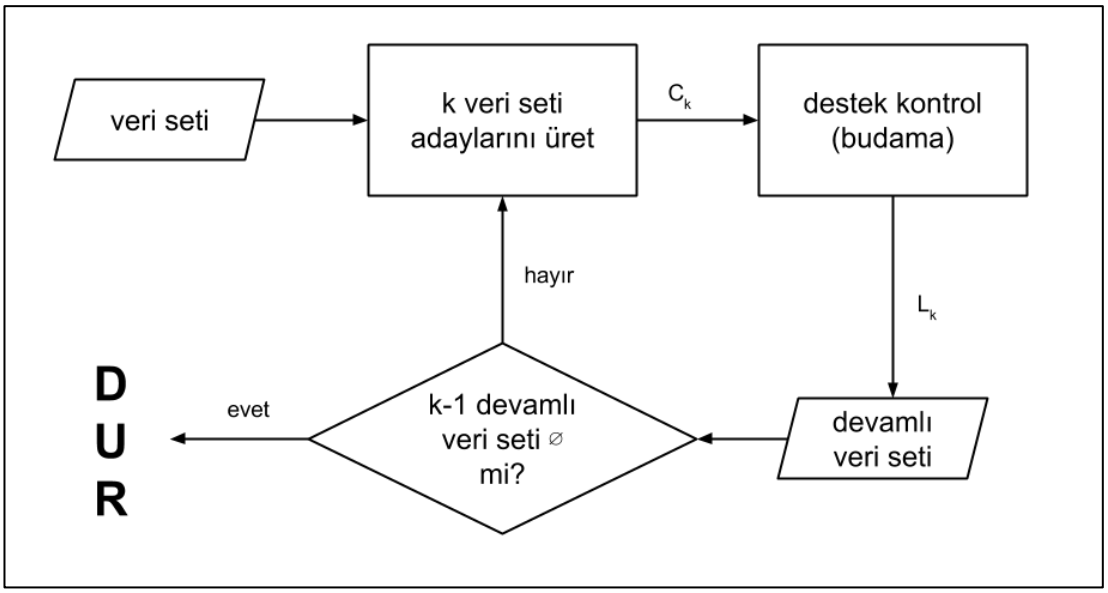
Bağlantı kurallarının ortaya çıkarılabilmesi için daha önceden yayınlanmış AIS ve SETM algoritmalarından farklı olarak Apriori algoritmasında her bir geçişte aday nesne kümelerinin oluşturulma ve sayılma şekli farklıdır. Bağlantı kurallarının ortaya çıkarılabilmesi için daha önceden yayınlanmış AIS ve SETM algoritmalarından nesnelerin sayılması ve aday nesne kümelerinin oluşturulması yönünden farklılıklar vardır. Bu algoritmalarda daha veriler sayılırken aday nesne kümeleri oluşturulur. Bu durumda henüz aday nesne kümesi olan bir nesne kümesi dahi geniş nesne kümesi sayılır. Bu da gereksiz işlemlere yol açarak algoritmalarda gereksiz zaman kayıpları oluşturur. Apriori algoritmasında ise aday nesne kümeleri veri tabanındaki işlemler önemsenmeden sadece bir önceki tarama işleminde elde edilen geniş nesne kümeleri kullanılarak oluşturulur. Bu durum her bir taramada elde edilen geniş bir nesne kümesinin herhangi bir alt kümesinin de geniş nesne kümesi olacağı kabulüne dayanır.

1994 yılında Agrawal ve Srikant tarafından geliştirilen Apriori algoritması 20. Very Large Database Endowment konferansında sunulmuştur. Bu sunumda sözde kodu Şekil 2.3'de, akış diyagramı Şekil 2.4'de verilen Apriori algoritmasının çalışma ayrıntıları ve kaba kodu şu şekilde sunulmuştur (Agrawal & Srikant, 1993):

1. İlk tarama sırasında geniş nesne kümelerinin tespiti için tüm nesneler sayılır.
2. Bir sonraki k. tarama iki aşamadan oluşur;
    a. Apriori-gen fonksiyonu kullanılarak, (k-1). taramada elde edilen, L_{k-1} nesne kümeleriyle C_k aday nesne kümeleri oluşturulur,
    b. Veri tabanı taranarak C_k'daki adayların desteği hesaplanır.
3. Hızlı bir sayım işlemi için, verilen bir I işlemindeki C_k'yı oluşturan adayların çok iyi bilinmesi gerekir.



Şekil 2.3. Apriori algoritmasının sözde kodu (Agrawal, Imielinski & Swami, 1993).



Şekil 2.4. Apriori algoritmasının akış diyagramı

Örnek

Çizelge 2.2'de bulunan alışveriş listesi örneğindeki verilere, minimum destek %25 ve güven %55 olacak şekilde Apriori algoritması uygulanması.

Çizelge 2.2. Alışveriş sepeti örneği

ID	SEPET
1	Peynir, Zeytin, Bal, Ekmek
2	Peynir, Zeytin, Ekmek
3	Yumurta, Ekmek
4	Yumurta, Çikolata
5	Peynir, Zeytin
6	Peynir, Zeytin, Yumurta

Veri tabanında bulunan her farklı tip veri tespit edilerek bu verilerin veri tabanında bulunma miktarları ve toplam kayıt sayısına oranları (destek değeri) hesaplanır.

Çizelge 2.3. Birinci tarama sonucu

Ürün	**Miktar**	**Destek**
Peynir	4	%67
Zeytin	4	%67
Bal	1	%17
Ekmek	3	%50
Yumurta	3	%50
Çikolata	1	%17

Çizelge 2.3'de verilen birinci taramada elde edilen sonuçlardan destek değerini aşan veri grupları, birbirleri ile farklı gruplar oluşturarak tekrar veri tabanı üzerinde tarama yapılır. Yine aynı şekilde destek değerleri hesaplanır.

Çizelge 2.4. İkinci tarama sonucu

Ürün	**Miktar**	**Destek**
Peynir, Zeytin	4	%67
Peynir, Ekmek	2	%33
Peynir, Yumurta	1	%17
Zeytin, Ekmek	2	%33
Zeytin, Yumurta	1	%17
Ekmek, Yumurta	1	%17

Çizelge 2.4'de ikinci tarama sonucu verilmiştir. İkinci taramadan da elde edilen veriler tekrar birbirleri ile gruplanarak aynı işlemler tekrarlanır ve destek değerleri hesaplanır.

Çizelge 2.5. Üçüncü tarama sonucu

Ürün	**Miktar**	**Destek**
Peynir, Zeytin, Ekmek	2	%33

Çizelge 2.5'de elde edilen maksimum veri setleri gösterilmiştir. Algoritmadaki bu işlemler, maksimum elde edilecek veri setleri bulununcaya kadar devam edilir. Bu örnekte üçüncü taramada maksimum veri seti elde edildiğinden yapılacak işlemler sonlandırılır.

En geniş kümeler ve yüzdelikleri:

Peynir alanlar (4 kayıt), Zeytin ve Ekmek alır (2 Kayıt) [Destek %33, Güven = %50]
Zeytin alanlar (4 kayıt), Peynir ve Ekmek alır (2 Kayıt) [Destek %33, Güven = %50]
Ekmek alanlar (3 kayıt), Peynir ve Zeytin alır (2 Kayıt) [Destek %33, Güven = %67]
Peynir ve Zeytin alanlar (4 kayıt), Ekmek alır (2 Kayıt) [Destek %33, Güven = %50]
Peynir ve Ekmek alanlar (2 kayıt), Zeytin alır (2 Kayıt) [Destek %33, Güven =%100]

Yukarıdaki çıkarımlarda 3 ve 5 numaralı nesne kümeleri, başlangıçta tanımlanan %25 destek ve %55 güven seviyeleri aşarak bu aşamada en geniş nesne kümeleri olmuştur.

Ekmek alanlar (3 kayıt), Peynir ve Zeytin alır (2 Kayıt) [Destek %33, Güven = %67]
Peynir ve Ekmek alanlar (2 kayıt), Zeytin alır (2 Kayıt) [Destek %33, Güven =%100]

Mevcut verilere göre Peynir – Zeytin – Ekmek en geniş nesne kümesidir. Güven seviyesi ise değerlendirme şeklinde göre değişmektedir. Örneğin her peynir ve ekmek alan, mutlaka zeytin de satın almıştır; yorumu yapılabilirken her peynir ve zeytin alan, mutlaka ekmek almıştır şeklinde bir yorum yapılamamaktadır.

3. MATERYAL VE YÖNTEM

Bir eğitim yazılımından alınan sınav verileri üzerine Apriori algoritması uygulanarak ilgili verilerden kurallar çıkarmak amaçlanmıştır. Bu veriler, ilgili sistem üzerinden hazırlanan sınavların, bu sınavlardaki soruların ve öğrencilerin özelliklerinden ve öğrencilerin sorulara verdikleri cevaplardan oluşmaktadır.

3.1. Kullanılan Teknolojiler

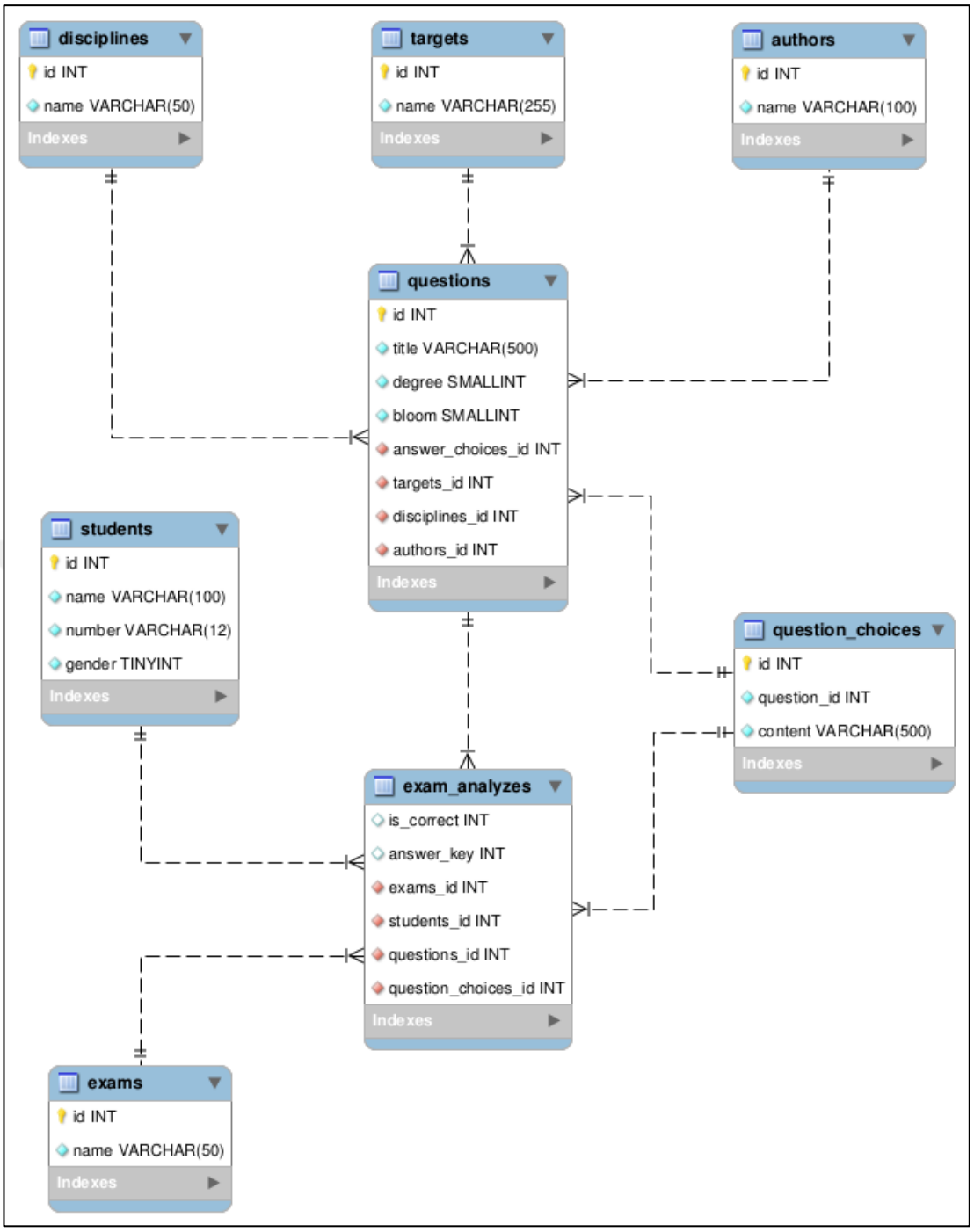
Üzerinde çalışılan eğitim yazılımı web tabanlı olarak çalışmaktadır. Ruby on Rails web çatısı üzerine Ruby programlama dili ile geliştirilen bu yazılım Linux işletim sistemi üzerinde çalışmaktadır. Hazırladığımız uygulama ise doğrudan sistemle birlikte çalışabilmesi için Ruby programlama dili ve teknikleri kullanılarak hazırlanmıştır.

3.2. Verilerin Hazırlanması

Sistem üzerinde doğrudan çalışılmasının risklerinin olması sebebiyle sistemin bir kopyası oluşturulmuştur. Oluşturulan bu kopyanı veri tabanı olduğu gibi değil sadece ilgili alanları kullanılmıştır. Oluşturulan veri tabanı Şekil 3.1'de gösterildiği gibi 8 ayrı bölümden oluşmaktadır.

İlgili tablolardan elde edilen veriler üzerinde veri madenciliği işlemlerinin gerçekleştirilebilmesi için bu veriler işlenebilir bir şekilde farklı bir alanda bir araya getirilmeleri gerekmektedir. Bunun için "data_mining" adında yeni bir tablo oluşturularak özellikleri Çizelge 3.1'de gösterildiği şekilde düzenlenmiştir.

Hazırlanan tablonun içeriği veri tabanındaki veriler ile doldurulur. Ancak bu verilerin yeni tabloya geçişi sırasında veri madenciliği için veri hazırlama süreçlerinin gerçekleştirilmesi gerekir. Bu süreçler, hazırlanan yeni tablo üzerine veriler aktarılırken veya sonrasında güncelleme işlemleri ile tamamlanmıştır.



Şekil 3.1. Veri tabanı grafiksel gösterimi

Çizelge 3.1. Veri madenciliği için hazırlanan veri tabanı tablosunun özellikleri

Alan Adı	**Alan Özelliği**	**Anlamı**
exam_id	INTEGER	Sınav ID'si
user_id	INTEGER	Öğrenci ID'si
question_id	INTEGER	Soru ID'si
answer_id	INTEGER	Soru cevabı ID'si
user_answer_id	INTEGER	Öğrenci soru cevabı ID'si
author_id	INTEGER	Soru yazarı ID'si
discipline_id	INTEGER	Sorunun ait olduğu anabilim dalı ID'si
system_id	INTEGER	Sorunun ait olduğu ID'si
block_id	INTEGER	Sınavın ait olduğu kurul ID'si
ncc_id	INTEGER	Sorunun ait olduğu öğrenim hedefi ID'si
efp_id	INTEGER	Sorunun ait olduğu hastalık ID'si
info_degree	INTEGER	Hastalığın öğrenme düzeyi
user_term	INTEGER	Öğrencinin dönemi
bloom	INTEGER	Sorunun bloom düzeyi
degree	INTEGER	Sorunun zorluk derecesi
is_correct	INTEGER	Sorunun cevaplanma durumu
answer_key	INTEGER	Sorunun cevap indeksi
booklet_type	INTEGER	Öğrencinin sınava katıldığı kitapçık türü
gender	INTEGER	Öğrencinin cinsiyeti
pre_exam	INTEGER	İlgili kurulun ön sınavına katılma bilgisi
grade	STRING	Öğrencinin sınavdan aldığı not aralığı
qlength	STRING	Sorunun uzunluğu

Sistem üzerinde en fazla 4 kitapçık olacak şekilde hazırlanan her sınav belirlenen tarihlerde gerçekleştirildikten sonra öğrencilerin doldurdukları optik çıktılar optik okuyucu programı ile okunur ve her öğrencinin sınav sıralarına ait işaretlemeleri bir çıktı dosyası içinde elde edilir. Bu dosya sisteme yüklenir ve sınav analizi yapılmak istenir. Sistem sınav analizi gerçekleştirerek farklı türlerde istatistiki veriler üretir.

3.2.1. Kayıp verilerin tamamlanması

Var olan verilerin tamamı sistem süreçleri sonunda oluşan tanımlı verilerdir. Oluşturulan yeni tabloda boş bir alan bulunmamaktadır. Örneğin tüm soruların bir zorluk derecesi olduğu gibi tüm öğrenciler bir cinsiyete sahiptir. Tüm bu veriler sistemde eksiksiz mevcuttur. Öğrencilerin sınavda karşılaştıkları soruları boş bırakma durumları söz konusudur. Bu durumda ise soruların boş bırakılma durumları belirli sabit bir değer ile tanımlı hale getirilmiştir. Bu sebeple kayıp verilerin tamamlanması işlemi için herhangi bir farklı yöntem kullanma ihtiyacı yoktur.

3.2.2. Gürültülü verilerin temizlenmesi

Oluşturulan yeni tabloya aktarılan bütün verilerin bir standardı mevcuttur. Farklı bir tablodaki bir veriyi simgeleyen her bir alanın belirli bir alt ve üst sınırı mevcuttur. Bu değerler Çizelge 3.2'de gösterilmiştir. Bu sebeple gürültülü verilerin temizlenmesi işlemi için herhangi bir farklı yöntem kullanma ihtiyacı yoktur.

3.2.3. Verilerin bütünleştirilmesi

Yeni tabloya aktarılan bütün veriler sistem kaynaklarından elde edildiklerinden aynı değerleri farklı şekillerde sembolleştirilmeleri söz konusu değildir. Örneğin tüm sistemde cinsiyet durumu kadın için 0 erkek için 1 ile sembolleştirilmiştir. Bu sebeple verilerin bütünleştirilmesi işlemi için herhangi bir farklı yöntem kullanma ihtiyacı yoktur.

Çizelge 3.2. Veri madenciliği için hazırlanan tablonun alabileceği değerler

Alan Adı	**Alabileceği değerler**
exam_id	0'dan büyük sayısal değer
user_id	0'dan büyük sayısal değer
question_id	0'dan büyük sayısal değer
answer_id	0'dan büyük sayısal değer
user_answer_id	0'dan büyük sayısal değer
author_id	0'dan büyük sayısal değer
discipline_id	0'dan büyük sayısal değer
system_id	0'dan büyük sayısal değer
block_id	0'dan büyük sayısal değer
ncc_id	0'dan büyük sayısal değer
efp_id	0'dan büyük sayısal değer
info_degree	0'dan 14'e kadar olan sayısal değer
user_term	1'den 6'ya kadar olan sayısal değer
bloom	1'den 6'ya kadar olan sayısal değer
degree	1'den 5'e kadar olan sayısal değer
is_correct	0: Doğru, 1: Yanlış, 2: Boş olan sayısal değer
answer_key	0: A, 1: B, 2: C, 3: D, 4: E, 5: Boş olan sayısal değer
booklet_type	0: A, 1: B, 2: C, 3: D olan sayısal değer
gender	0: Kadın, 1: Erkek olan sayısal değer
pre_exam	0: Evet, 1: Hayır olan sayısal değer
grade	Çok kötü, Kötü, Orta, İyi ve Çok iyi şeklinde olan değer
qlength	Kısa, Normal ve Uzun şeklinde olan değer

3.2.4. Verilerin dönüştürülmesi

Üzerinde çalışma yapılacak veriler için o verilerin sistem üzerinde temsil edildiği değerler kullanılır. Mevcut verilerin bu sembolleri üzerinde genellikle herhangi bir değişikliğe ihtiyaç duymuyoruz. Ancak sınav sorularının uzunluğu ve öğrencilerin aldığı notların dönüştürülmesine ihtiyaç duyulmuştur. Sınav sorularının uzunluğunun ucu açık olması, öğrenci notlarının 0 ile 100 değerleri arasında değişiyor olması veri çeşitliliğini artırmaktadır. Bu nedenle bu değerlerin belirli standartlara göre gruplanması ihtiyacı doğmuştur.

Sınav sorusu uzunluğunun dönüştürülmesinde Çizelge 3.3'de belirlenen yöntem uygulanmıştır.

Çizelge 3.3. Sınav sorusu uzunluğunun dönüştürülmesi

Aralık	**Dönüştürülen Değer**
0 - 100	Kısa
101 - 250	Normal
250 - ∞	Uzun

Öğrenci notlarının dönüştürülmesinde için Çizelge 3.4'de belirlenen yöntem uygulanmıştır.

Çizelge 3.4. Öğrenci notlarının dönüştürülmesi

Aralık	**Dönüştürülen Değer**
0 – 20	Çok kötü
21 – 40	Kötü
41 – 60	Orta
61 – 80	İyi
81 – 100	Çok iyi

3.2.5. Verilerin azaltılması

Sistem üzerindeki sınav verileri her geçen gün yapılan sınavlarla birlikte artıyor. Ancak bir sınava ait veri çeşitliliği veya kapasitesi daima sabittir. Sistem bu verileri minimum kapasite harcayarak tüm istatistiki analizleri verebilecek şekilde tasarlanmıştır. Sistemin analiz işlemleri için kullandığı tablo ile üzerinde veri madenciliği işlemleri yapılacak tablo farklıdır. Sadece veri madenciliği işlemi için yeni bir tablo hazırlanmıştır. Bu sırada ilgili sınav ve özellikleri alınırken gereksiz alanların kullanımından kaçınılmıştır.

Verilerin sadece ihtiyaç duyulan kısımlarından oluşan farklı bir alan oluşturmak da aslında bir indirgeme işlemidir. Veriler doğrudan ilgili tablodan anlık olarak okunarak işlenir. Bu da gereksiz verileri kullanmadan, istenilen sorguları yapılarak sadece istenilen sonuçlara erişimi sağlar. Bu sebeple verilerin azaltılması işlemi için herhangi bir farklı yöntem kullanma ihtiyacı yoktur.

3.3. Apriori Algoritması ile Uygulama

Uygulama bölümünün ilk adımında var olan sistemden veri madenciliği işlemlerini uygulayabileceğimiz bir yeni alan oluşturduk. Bu yeni alanı oluştururken veri madenciliği adımlarını takip ettik. Bu aşamada ise veri üzerine Apriori algoritması uygulanarak birliktelik kurallarının ortaya çıkarılması amaçlanmıştır.

Apriori algoritması çeşitli veri madenciliği tekniklerinin barındırıldığı yazılımlar üzerinden kullanılabileceği gibi kullanılacak programlama dili ile betik olarak hazırlanmış hazır yazılımlar kullanılabilir. Tüm bu süreçlerde verinin ayrı bir formatta üretilip kullanılması gerekir.

Bu tür kullanımların yerine Apriori algoritması sistemin dili olan Ruby programlama dili ile hazırlanarak doğrudan ilgili veri tabanı tablosu üzerinde çalıştırılmıştır. Böylece herhangi bir farklı yazılıma ihtiyaç duyulmamıştır.

3.3.1. Apriori algoritmasının yazılımının hazırlanması

Hazırlanan yazılım, eğitim yazılımının da dili olan Ruby programlama dili ile hazırlanmıştır. Bu sırada kullanılan web çatısına doğrudan dâhil olabilmesi için veri tabanı ilişkilerinin nesnel olarak ifade edilebildiği ve yönetilebildiği bir ORM olan

(Object to Relational Mapping) Active Record nesne kümesinden faydalanılmıştır. İlgili kod bloğu EK 1'de verilmiştir.

3.3.2. Yazılımdan elde edilen sonuçların yorumlanması

Algoritma, veriler üzerinde uygulanırken tanımlanması gereken destek ve güven seviyeleri farklı değerler ile tanımlanarak birden fazla sonuç elde edilmiştir. Sonuçlar veri tabanının ilgili tablosunun sütun isimlerine göre ve değerlerin sembol karşılıklarına göre oluşturulmuştur. Hazırlanan veriler %30 destek ve %50 güven değerleri ile analiz edildiğinde Çizelge 3.5'deki gibi bir sonuç elde edilmektedir.

Çizelge 3.5. Uygulama çıktısının yorumlanmamış sonuçları

Öğe Kümesi		**Destek**	**Güven**
is_correct = 0	qlength = Kısa	0.38	0.60
is_correct = 0	grade = İyi	0.41	0.64
qlength = Kısa	is_correct = 0	0.38	0.62
qlength = Kısa	grade = İyi	0.38	0.61
gender = 0	is_correct = 0	0.35	0.65
gender = 1	grade = İyi	0.30	0.65
pre_exam = 0	is_correct = 0	0.30	0.62
pre_exam = 0	qlength = Kısa	0.30	0.62
pre_exam = 1	is_correct = 0	0.34	0.66
pre_exam = 1	grade = İyi	0.34	0.67
grade = İyi	is_correct = 0	0.41	0.67
grade = İyi	qlength = Kısa	0.38	0.62
grade = İyi	gender = 0	0.31	0.51
grade = İyi	pre_exam = 1	0.34	0.56

Çizelge 3.5'deki veriler doğrudan anlaşılabilir sonuçlar değildir. Hazırlanan yazılıma eklenen Apriori algoritmasından elde edilen ilişki kuralları anlaşılabilir bir hale getirilebilmektedir. Çıkarılan kurallar Çizelge 3.6 içerisinde belirtilmiştir.

Çizelge 3.6. Uygulama çıktısının yorumlanmış sonuçları

Öğe Kümesi		**Destek**	**Güven**
Cevap = Doğru	Soru Uzunluğu = Kısa	0.38	0.60
Cevap = Doğru	Not = İyi	0.41	0.64
Soru Uzunluğu = Kısa	Cevap = Doğru	0.38	0.62
Soru Uzunluğu = Kısa	Not = İyi	0.38	0.61
Cinsiyet = Kadın	Cevap = Doğru	0.35	0.65
Cinsiyet = Erkek	Not = İyi	0.30	0.65
Ön Sınav = Hayır	Cevap = Doğru	0.30	0.62
Ön Sınav = Hayır	Soru Uzunluğu = Kısa	0.30	0.62
Ön Sınav = Evet	Cevap = Doğru	0.34	0.66
Ön Sınav = Evet	Not = İyi	0.34	0.67
Not = İyi	Cevap = Doğru	0.41	0.67
Not = İyi	Soru Uzunluğu = Kısa	0.38	0.62
Not = İyi	Cinsiyet = Kadın	0.31	0.51
Not = İyi	Ön Sınav = Evet	0.34	0.56

Çizelge 3.5'deki sonuçlar ve Çizelge 3.6'daki çıkarımlar yalnızca bir örnektir. Bu sonuçların azlığı veya çokluğu var olan verilerin özelliklerine bağlıdır.

3.3.3. Sınavların analiz edilmesi

Analiz için kullanılacak sınav sonuçlarının alındığı eğitim yazılımı, tıp fakültelerinin eğitim ve sınavlarının yönetimini sağlayan bir programdır. Tıp fakültelerinin eğitim sistemi dört yıllık eğitim veren fakültelerin eğitim sisteminden farklıdır. Dört yıllık fakültelerin eğitimlerinde bir eğitimci bir dersten sorumlu olur ve bu dersin sınavını dönem sonunda gerçekleştirir. Tıp fakültelerinde ise yine bir dersten bir eğitimci sorumludur. Ancak bir dersin sınavı yapılmaz birçok anabilim dalından belirli bir süre içerisinde verilen derslerin sonunda bir sınavı yapılır. Dolayısıyla sınav soruları farklı anabilim dallarından farklı eğitimciler tarafından hazırlanır. Bu farklı anabilim dallarından verilen derslerin bulunduğu yapıya kurul adı verilirken kurul sonunda yapılan sınav ise kurul sonu sınavı olarak adlandırılmaktadır.

Yılsonunda ise yıl içinde gerçekleştirilen tüm kurullarda verilen eğitim, yılsonu sınavı adı altında yapılan bir sınav ile tekrar değerlendirilir. Öğrenciler kurullardan ve yılsonu sınavından aldıkları notlar ile değerlendirilirler. Her kurulun kendine has not hesaplama yöntemi olabilir.

Kurul sonu sınavlarının özelikleri gereği hazırlanan data_mining tablosunun belirli sütunları kullanılacaktır. Bu sütunlar Çizelge 3.7'de verilmiştir.

Analiz işlemini birinci, ikinci ve üçüncü sınıfların birer kurul sınavlarında ve yılsonu sınavlarında gerçekleştirerek elde edilen sonuçları inceleyeceğiz. Sınav isimleri şunlardır:

- 1. sınıf 1. kurul sonu sınavı
- 1. sınıf yılsonu sınavı
- 2. sınıf 1. kurul sonu sınavı
- 2. sınıf yılsonu sınavı
- 3. sınıf 1. kurul sonu sınavı
- 3. sınıf yılsonu sınavı

Çizelge 3.7. Kurul sonu sınavlarda kullanılacak alanlar

Alan Adı	**Anlamı**
author_id	Soru yazarı ID'si
discipline_id	Sorunun ait olduğu anabilim dalı ID'si
ncc_id	Sorunun ait olduğu öğrenim hedefi ID'si
degree	Sorunun zorluk derecesi
is_correct	Sorunun cevaplanma durumu
answer_key	Sorunun cevap indeksi
booklet_type	Öğrencinin sınava katıldığı kitapçık türü
gender	Öğrencinin cinsiyeti
pre_exam	İlgili kurulun ön sınavına katılma bilgisi
grade	Öğrencinin sınavdan aldığı not aralığı
qlength	Sorunun uzunluğu

Sınav verileri işlendikten sonra bulunan ilişki kurallarının tamamı anlamlı olmayabilir. Bu birliktelik ilişkileri bir insanın yorumlaması yardımıyla sonuçlara dönüştürülebilir. Sınav verilerinden elde edilen ilişki kurallarından bazıları yorumlanarak çıkarımlar adı altında listelenmiştir.

Not: Ön sınav olarak adlandırılan sınav, kurul sonu sınavlarından önce kurul sonu sınavını hazırlayacak hocalar tarafından hazırlanan doğru yanlış şeklinde cevap verilebilen tanımlamalardan oluşan bir sınavdır. Fakülteler bu sınavları isteğe bağlı olarak hazırlayabileceği gibi öğrenciler de bu sınavlara katılmak zorunda değillerdir. Bu sayede bu sınava katılan ve katılmayan öğrencilerin durumları da ilişki kuralları içerisinde görülebilmektedir.

1. sınıf 1. kurul sonu sınavı özellikleri ve bu sınav sonuçlarından elde edilen ilişki kuralları Çizelge 3.8 ve Çizelge 3.9'da gösterilmiştir.

Çizelge 3.8. 1. sınıf 1. kurul sonu sınavı özellikleri

Özellik	**Değer**
Soru sayısı	100
Katılan öğrenci sayısı	176
Sorusu bulunan eğitimci sayısı	8
Sorusu bulunan anabilim dalı sayısı	5
Sorusu bulunan öğrenim hedefi sayısı	29
Kitapçık sayısı	2
Toplam veri sayısı	17600

Çizelge 3.9'da, %30 destek ve %60 güven seviyesi ile elde edilen ilişki kuralları gösterilmiştir.

Çizelge 3.9. 1. sınıf 1. kurul sonu sınavı ilişki kuralları

Öğe Kümesi		**Destek**	**Güven**
Soru Yazarı = Eğitimci-1	Anabilim Dalı = Tıbbi Genetik	0.34	1.00
Soru Yazarı = Eğitimci-1	Zorluk Derecesi = 3	0.31	0.91
Anabilim Dalı = Tıbbi Genetik	Soru Yazarı = Eğitimci-1	0.34	1.00
Anabilim Dalı = Tıbbi Genetik	Zorluk Derecesi = 3	0.31	0.91
Anabilim Dalı = Anatomi	Zorluk Derecesi = 1	0.32	1.00
Zorluk Derecesi = 1	Anabilim Dalı = Anatomi	0.32	0.62
Zorluk Derecesi = 1	Cevap = Doğru	0.33	0.64
Zorluk Derecesi = 3	Soru Yazarı = Eğitimci-1	0.31	0.70
Zorluk Derecesi = 3	Anabilim Dalı = Tıbbi Genetik	0.31	0.70
Kitapçık Türü = A	Cevap = Doğru	0.32	0.64
Kitapçık Türü = A	Cinsiyet = Kadın	0.31	0.61
Kitapçık Türü = B	Cevap = Doğru	0.32	0.64

Soru Uzunluğu = Kısa	Zorluk Derecesi = 1	0.30	0.65
Soru Uzunluğu = Kısa	Cevap = Doğru	0.31	0.67
Cinsiyet = Kadın	Cevap = Doğru	0.37	0.67
Ön Sınav = Hayır	Cevap = Doğru	0.34	0.60
Ön Sınav = Evet	Cevap = Doğru	0.31	0.69
Not = Kötü	Ön Sınav = Hayır	0.31	0.63
Not = Orta	Cevap = Doğru	0.35	0.74
Soru Yazarı = Eğitimci-1	Anabilim Dalı = Tıbbi Genetik Zorluk Derecesi = 3	0.31	0.91
Anabilim Dalı = Tıbbi Genetik Zorluk Derecesi = 3	Soru Yazarı = Eğitimci-1	0.31	1.00
Soru Yazarı = Eğitimci-1	Zorluk Derecesi = 3 Anabilim Dalı = Tıbbi Genetik	0.31	0.91
Zorluk Derecesi = 3 Anabilim Dalı = Tıbbi Genetik	Soru Yazarı = Eğitimci-1	0.31	1.00
Anabilim Dalı = Tıbbi Genetik	Soru Yazarı = Eğitimci-1 Zorluk Derecesi = 3	0.31	0.91
Soru Yazarı = Eğitimci-1 Zorluk Derecesi = 3	Anabilim Dalı = Tıbbi Genetik	0.31	1.00
Anabilim Dalı = Tıbbi Genetik	Zorluk Derecesi = 3 Soru Yazarı = Eğitimci-1	0.31	0.91
Zorluk Derecesi = 3 Soru Yazarı = Eğitimci-1	Anabilim Dalı = Tıbbi Genetik	0.31	1.00
Zorluk Derecesi = 3	Soru Yazarı = Eğitimci-1 Anabilim Dalı = Tıbbi Genetik	0.31	0.70
Soru Yazarı = Eğitimci-1 Anabilim Dalı = Tıbbi Genetik	Zorluk Derecesi = 3	0.31	0.91
Zorluk Derecesi = 3	Anabilim Dalı = Tıbbi Genetik Soru Yazarı = Eğitimci-1	0.31	0.70
Anabilim Dalı = Tıbbi Genetik Soru Yazarı = Eğitimci-1	Zorluk Derecesi = 3	0.31	0.91

Çizelge 3.9'dan elde edilen çıkarımlar:

1. Eğitimci-1 tarafından yazılan ve Tıbbi Genetik anabilim dalına ait soruların tüm sorulara oranı %34'tür ve bu soruların tamamı Tıbbi Genetik anabilim dalına aittir. Dolayısıyla tüm Tıbbi Genetik soruları Eğitimci-1 tarafından hazırlanmıştır.
2. Zorluk derecesi 1 ve Anatomi anabilim dalına ait sorular tüm soruların %32'sidir ve Anatomi anabilim dalına ait sorular zorluk derecesi 1 olan soruların %62'ini oluşturmaktadır.
3. Zorluk derecesi 1 olan sorulara ait doğru olarak yapılan işaretlemelerin tüm işaretlemelere oranı %33'tür ve zorluk derecesi 1 olan soruların %64'ü doğru olarak işaretlenmiştir.
4. Soru uzunluğu Kısa ve doğru olan işaretlemelerin tüm işaretlemelere oranı %31'dir ve soru uzunluğu Kısa olan soruların %67'i doğru olarak işaretlenmiştir.
5. Sınav sonucunda Kötü not olan ve ön sınava girmeyen öğrencilerin tüm öğrencilere oranı %31'dir ve ön sınava girmeyen öğrencilerin %63'ü Kötü not almıştır.

1. sınıf yılsonu sınavı özellikleri ve bu sınav sonuçlarından elde edilen ilişki kuralları Çizelge 3.10 ve Çizelge 3.11'de gösterilmiştir.

Çizelge 3.10. 1. sınıf yılsonu sınavı özellikleri

Özellik	**Değer**
Soru sayısı	134
Katılan öğrenci sayısı	169
Sorusu bulunan eğitimci sayısı	30
Sorusu bulunan anabilim dalı sayısı	13
Sorusu bulunan öğrenim hedefi sayısı	110
Kitapçık sayısı	2
Toplam veri sayısı	22646

Çizelge 3.11'de, %30 destek ve %60 güven seviyesi ile elde edilen ilişki kuralları gösterilmiştir.

Çizelge 3.11. 1. sınıf yılsonu sınavı ilişki kuralları

Öğe Kümesi		**Destek**	**Güven**
Cevap = Doğru	Zorluk Derecesi = 1	0.42	0.72
Cevap = Yanlış	Zorluk Derecesi = 1	0.31	0.78
Kitapçık Türü = A	Zorluk Derecesi = 1	0.39	0.75
Kitapçık Türü = B	Zorluk Derecesi = 1	0.36	0.75
Soru Uzunluğu = Kısa	Zorluk Derecesi = 1	0.36	0.75
Cinsiyet = Kadın	Zorluk Derecesi = 1	0.42	0.75
Cinsiyet = Kadın	Not = Kötü	0.34	0.61
Cinsiyet = Erkek	Zorluk Derecesi = 1	0.33	0.75
Not = Kötü	Zorluk Derecesi = 1	0.42	0.75

2. sınıf 1. kurul sonu sınavı özellikleri ve bu sınav sonuçlarından elde edilen ilişki kuralları Çizelge 3.12 ve Çizelge 3.13'de gösterilmiştir.

Çizelge 3.12. 2. sınıf 1. kurul sonu sınavı özellikleri

Özellik	**Değer**
Soru sayısı	125
Katılan öğrenci sayısı	198
Sorusu bulunan eğitimci sayısı	8
Sorusu bulunan anabilim dalı sayısı	5
Sorusu bulunan öğrenim hedefi sayısı	55
Kitapçık sayısı	3
Toplam veri sayısı	24750

Çizelge 3.13'de, %30 destek ve %60 güven seviyesi ile elde edilen ilişki kuralları gösterilmiştir.

Çizelge 3.13. 2. sınıf 1. kurul sonu sınavı ilişki kuralları

Öğe Kümesi		**Destek**	**Güven**
Anabilim Dalı = Anatomi	Zorluk Derecesi = 1	0.41	1.00
Anabilim Dalı = Fizyoloji	Soru Uzunluğu = Kısa	0.34	0.88
Zorluk Derecesi = 1	Anabilim Dalı = Anatomi	0.41	0.65
Zorluk Derecesi = 1	Soru Uzunluğu = Kısa	0.38	0.61
Zorluk Derecesi = 1	Ön Sınav = Hayır	0.45	0.71
Cevap = Doğru	Zorluk Derecesi = 1	0.34	0.62
Cevap = Doğru	Soru Uzunluğu = Kısa	0.42	0.77
Cevap = Doğru	Ön Sınav = Hayır	0.37	0.69
Cevap = Yanlış	Soru Uzunluğu = Kısa	0.33	0.73
Cevap = Yanlış	Ön Sınav = Hayır	0.34	0.74
Soru Uzunluğu = Kısa	Ön Sınav = Hayır	0.54	0.71
Cinsiyet = Kadın	Soru Uzunluğu = Kısa	0.35	0.75

Cinsiyet = Erkek	Zorluk Derecesi = 1	0.34	0.63
Cinsiyet = Erkek	Soru Uzunluğu = Kısa	0.40	0.75
Cinsiyet = Erkek	Ön Sınav = Hayır	0.42	0.78
Ön Sınav = Hayır	Zorluk Derecesi = 1	0.45	0.63
Ön Sınav = Hayır	Soru Uzunluğu = Kısa	0.54	0.75
Not = Kötü	Zorluk Derecesi = 1	0.32	0.63
Not = Kötü	Soru Uzunluğu = Kısa	0.38	0.75
Not = Kötü	Cinsiyet = Erkek	0.30	0.60
Not = Kötü	Ön Sınav = Hayır	0.37	0.73

Çizelge 3.13'den elde edilen çıkarımlar:

1. Anatomi anabilim dalından gelen ve zorluk derecesi 1 olan soruların tüm sorulara oranı %41'dir ve Anatomi sorularının %100'ünün zorluk derecesi 1'dir. Anatomi anabilim dalından gelen soruların tamamının zorluk derecesinin 1'dir. Bununla birlikte zorluk derecesi 1 olan soruların %65'i Anatomi anabilim dalından gelmiştir.
2. Tüm öğrenciler tarafından Doğru olarak yapılan işaretlemeler ile zorluk derecesi 1 olan soruların işaretlemelerinin tüm işaretlemelere oranı %34'tür ve zorluk derecesi 1 olan işaretlemelerin doğru işaretlemelere oranı %62'dir.
3. Doğru olarak yapılan işaretlemeler ile soru uzunluğu kısa olan soruların birlikte bulunma oranı %42'dir ve Doğru olarak yapılan işaretlemelerin %77'si soru uzunluğu Kısa olan sorulardan oluşmaktadır.
4. Yanlış olarak yapılmış işaretlemeler ile ön sınava girmemiş öğrenciler tüm işaretlemelerin %34'ünü oluşturuyor ve ön sınava girmemiş öğrencilerin tüm yanlış işaretlemeler içerisindeki oranı %74'tür.
5. Sınav notu itibariyle Kötü not alan öğrenciler ile ön sınava katılmayan öğrencilerin tüm öğrencilere oranı %38'dir ve notu Kötü olan öğrencilerin %73'ü ön sınava katılmamıştır.

2. sınıf yılsonu sınavı özellikleri ve bu sınav sonuçlarından elde edilen ilişki kuralları Çizelge 3.14 ve Çizelge 3.15'de gösterilmiştir.

Çizelge 3.14. 2. sınıf yılsonu sınavı özellikleri

Özellik	**Değer**
Soru sayısı	125
Katılan öğrenci sayısı	185
Sorusu bulunan eğitimci sayısı	25
Sorusu bulunan anabilim dalı sayısı	6
Sorusu bulunan öğrenim hedefi sayısı	118
Kitapçık sayısı	3
Toplam veri sayısı	23125

Çizelge 3.15'de, %30 destek ve %60 güven seviyesi ile elde edilen ilişki kuralları gösterilmiştir.

Çizelge 3.15. 2. sınıf yılsonu sınavı ilişki kuralları

Öğe Kümesi		**Destek**	**Güven**
Zorluk Derecesi = 1	Soru Uzunluğu = Kısa	0.55	0.64
Cevap = Doğru	Zorluk Derecesi = 1	0.48	0.86
Cevap = Doğru	Soru Uzunluğu = Kısa	0.38	0.68
Cevap = Yanlış	Zorluk Derecesi = 1	0.36	0.86
Kitapçık Türü = A	Zorluk Derecesi = 1	0.31	0.86
Kitapçık Türü = B	Zorluk Derecesi = 1	0.31	0.86
Soru Uzunluğu = Kısa	Zorluk Derecesi = 1	0.55	0.81
Cinsiyet = Kadın	Zorluk Derecesi = 1	0.41	0.86
Cinsiyet = Kadın	Soru Uzunluğu = Kısa	0.33	0.68
Cinsiyet = Erkek	Zorluk Derecesi = 1	0.44	0.86
Cinsiyet = Erkek	Soru Uzunluğu = Kısa	0.35	0.68
Not = Kötü	Zorluk Derecesi = 1	0.42	0.86

Not = Kötü	Soru Uzunluğu = Kısa	0.33	0.68
Not = Orta	Zorluk Derecesi = 1	0.31	0.86

Çizelge 3.15'den elde edilen çıkarımlar:

1. Zorluk derecesi 1 ve uzunluğu Kısa olan sorular tüm soruların %55'ini oluştururken Kısa uzunluktaki zorular zorluk derecesi 1 olan soruların %64'ünü oluşturmaktadır.
2. Zorluk derecesi 1 olan sorular üzerinde Doğru olarak yapılan işaretlemelerin tüm işaretlemelere oranı %48'dir ve Doğru olarak işaretlenen soruların %86'sının zorluk derecesi 1'dir.
3. Uzunluğu Kısa olan sorular üzerinde Doğru olarak yapılan işaretlemelerin tüm işaretlemelere oranı %38'dir ve Doğru olarak işaretlenen soruların %68'i Kısa uzunluktadır.
4. Zorluk derecesi 1 olan sorular üzerinde Yanlış olarak yapılan işaretlemelerin tüm işaretlemelere oranı %36'dır ve Yanlış olarak işaretlenen soruların %86'sının zorluk derecesi 1'dir.

3. sınıf 1. kurul sonu sınavı özellikleri ve bu sınav sonuçlarından elde edilen ilişki kuralları Çizelge 3.16 ve Çizelge 3.17'de gösterilmiştir.

Çizelge 3.16. 3. sınıf 1. kurul sonu sınavı özellikleri

Özellik	**Değer**
Soru sayısı	130
Katılan öğrenci sayısı	150
Sorusu bulunan eğitimci sayısı	33
Sorusu bulunan anabilim dalı sayısı	18
Sorusu bulunan öğrenim hedefi sayısı	92
Kitapçık sayısı	3
Toplam veri sayısı	19500

Çizelge 3.17'de, %30 destek ve %60 güven seviyesi ile elde edilen ilişki kuralları gösterilmiştir.

Çizelge 3.17. 3. sınıf 1. kurul sonu sınavı ilişki kuralları

Öğe Kümesi		**Destek**	**Güven**
Zorluk Derecesi = 1	Cevap = Doğru	0.50	0.71
Zorluk Derecesi = 1	Ön Sınav = Hayır	0.43	0.61
Cevap = Doğru	Zorluk Derecesi = 1	0.50	0.70
Cevap = Doğru	Soru Uzunluğu = Kısa	0.45	0.63
Cevap = Doğru	Ön Sınav = Hayır	0.43	0.60
Cevap = Doğru	Not = İyi	0.45	0.62
Kitapçık Türü = A	Zorluk Derecesi = 1	0.30	0.71
Kitapçık Türü = A	Cevap = Doğru	0.30	0.71
Kitapçık Türü = B	Zorluk Derecesi = 1	0.31	0.71
Kitapçık Türü = B	Cevap = Doğru	0.31	0.71
Kitapçık Türü = B	Ön Sınav = Hayır	0.33	0.76
Soru Uzunluğu = Kısa	Zorluk Derecesi = 1	0.42	0.68
Soru Uzunluğu = Kısa	Cevap = Doğru	0.45	0.72
Soru Uzunluğu = Kısa	Ön Sınav = Hayır	0.38	0.61
Cinsiyet = Kadın	Zorluk Derecesi = 1	0.36	0.71
Cinsiyet = Kadın	Cevap = Doğru	0.38	0.75
Cinsiyet = Kadın	Soru Uzunluğu = Kısa	0.32	0.62
Cinsiyet = Kadın	Not = İyi	0.35	0.70
Cinsiyet = Erkek	Zorluk Derecesi = 1	0.35	0.71
Cinsiyet = Erkek	Cevap = Doğru	0.34	0.69
Cinsiyet = Erkek	Soru Uzunluğu = Kısa	0.31	0.62
Cinsiyet = Erkek	Ön Sınav = Hayır	0.37	0.74
Ön Sınav = Hayır	Zorluk Derecesi = 1	0.43	0.71
Ön Sınav = Hayır	Cevap = Doğru	0.43	0.71
Ön Sınav = Hayır	Soru Uzunluğu = Kısa	0.38	0.62

Not = İyi	Zorluk Derecesi = 1	0.40	0.71
Not = İyi	Cevap = Doğru	0.45	0.79
Not = İyi	Soru Uzunluğu = Kısa	0.35	0.62
Not = İyi	Cinsiyet = Kadın	0.35	0.62

Çizelge 3.17'den elde edilen çıkarımlar:

1. Zorluk derecesi 1 olan sorular üzerinde Doğru olarak yapılan işaretlemelerin tüm işaretlemelere oranı %50'dir ve zorluk derecesi 1 olan soruların %71'i Doğru olarak işaretlenmiştir. Bununla birlikte zorluk derecesi 1 olan sorular üzerinde yapılan Doğru işaretlemelerin tüm işaretlemelere oranı %50'dir ama Doğru olarak işaretlenen soruların %70'inin zorluk derecesi 1'dir.
2. Cinsiyeti Kadın olan ve işaretlemelerde Doğru yanıtı veren öğrencilerin tüm işaretlemelere oranı %38'dir ve Kadın öğrencilerin %71'i işaretlemelerini Doğru olarak yapmıştır.
3. Ön sınava katılmayan ve Doğru olarak yapılan işaretlemelerin tüm işaretlemelere oranı %43'tür ve ön sınava katılmayan öğrencilerin %71'i işaretlemelerini Doğru olarak yapmıştır.

3. sınıf yılsonu sınavı özellikleri ve bu sınav sonuçlarından elde edilen ilişki kuralları Çizelge 3.18 ve Çizelge 3.19'da gösterilmiştir.

Çizelge 3.18. 3. sınıf yılsonu sınavı özellikleri

Özellik	**Değer**
Soru sayısı	150
Katılan öğrenci sayısı	151
Sorusu bulunan eğitimci sayısı	49
Sorusu bulunan anabilim dalı sayısı	27
Sorusu bulunan öğrenim hedefi sayısı	140
Kitapçık sayısı	3
Toplam veri sayısı	22650

Çizelge 3.19'da, %30 destek ve %60 güven seviyesi ile elde edilen ilişki kuralları gösterilmiştir.

Çizelge 3.19. 3. sınıf yılsonu sınavı ilişki kuralları

Öğe Kümesi		**Destek**	**Güven**
Zorluk Derecesi = 1	Not = Orta	0.55	0.67
Cevap = Doğru	Zorluk Derecesi = 1	0.49	0.78
Cevap = Doğru	Not = Orta	0.42	0.67
Cevap = Yanlış	Zorluk Derecesi = 1	0.31	0.88
Soru Uzunluğu = Kısa	Zorluk Derecesi = 1	0.47	0.84
Soru Uzunluğu = Kısa	Not = Orta	0.38	0.67
Soru Uzunluğu = Normal	Zorluk Derecesi = 1	0.30	0.78
Cinsiyet = Kadın	Zorluk Derecesi = 1	0.42	0.82
Cinsiyet = Kadın	Cevap = Doğru	0.33	0.64
Cinsiyet = Erkek	Zorluk Derecesi = 1	0.40	0.82
Cinsiyet = Erkek	Not = Orta	0.37	0.76
Not = Orta	Zorluk Derecesi = 1	0.55	0.82
Not = Orta	Cevap = Doğru	0.42	0.63
Cinsiyet = Erkek	Zorluk Derecesi = 1 Not = Orta	0.30	0.62
Zorluk Derecesi = 1 Not = Orta	Cinsiyet = Erkek	0.30	0.55
Cinsiyet = Erkek	Not = Orta Zorluk Derecesi = 1	0.30	0.62
Not = Orta Zorluk Derecesi = 1	Cinsiyet = Erkek	0.30	0.55

Çizelge 3.19'dan elde edilen çıkarımlar:

1. Zorluk derecesi 1 olan sorular üzerinde Doğru olarak yapılan işaretlemelerin tüm işaretlemelere oranı %49'dur ve Doğru olarak işaretlenen soruların %78'inin zorluk derecesi 1'dir.
2. Öğrencilerin yaptığı doğru işaretlemelere ile aldıkları Orta notun diğer tüm durumlara oranı %42'dir ve Doğru olarak yapılan işaretlemeler sonucunda öğrencilerin %67'i Orta not almıştır.
3. Zorluk derecesi 1 olan sorular üzerinde Yanlış olarak yapılan işaretlemelerin tüm işaretlemelere oranı %31'dir ve Yanlış olarak işaretlenen soruların %88'inin zorluk derecesi 1'dir.
4. Cinsiyeti Kadın olan ve öğrencilerin yaptığı Doğru işaretlemelerinin diğer tüm işaretlemelere oranı %33'tür ve Kadın öğrencilerin %64'ü işaretlemelerinin %64'ünü doğru olarak gerçekleştirmiştir.

4. SONUÇLAR VE ÖNERİLER

Birçok kurum ve kuruluş iş süreçlerindeki yol haritalarını sahip oldukları ve yönettikleri veriler üzerinde uyguladıkları veri madenciliği teknikleri ile belirlemektedir. Sağlık alanında veri madenciliği ile karar verme süreçlerinde kullanılan birçok uzman sistem mevcuttur. Üretilen tıbbi cihazlarda dahi veri madenciliği kullanılmaya başlanmıştır. Artık günlük yaşantımızın birçok alanında kullanılır hale gelmiştir.

Tıp fakültelerinde kullanılan bir eğitim yazılımı içerisinde bir sınav yönetim sisteminde vardır. Bu sınav yönetim sisteminde tıp fakülteleri, öğrencilerine yapacakları sınavları eğitim yazılım üzerinden hazırlamaktadırlar. Hazırlanan sınavlar yapıldıktan sonra öğrencilerin işaretleme yaptıkları optik formlar sisteme yüklenir ve veriler istatistiki olarak analiz edilir.

Yapılan bu çalışmada sınav sonuçlarına Apriori algoritması uygulanmış ve çıkarılan ilişki kuralları yorumlanarak farklı sonuçlar elde edilmeye çalışılmıştır. İstatistiki sonuçlar veren yazılım içerisine özel bir yazılım eklenmiştir. Bu yazılım veriler üzerine Apriori algoritmasını uygulamaktadır. Bu sayede yapılan her sınav sonrasında oluşan sınav verileri içerisinden anında ilişki kuralları çıkarılabilmektedir. Yapılan çıkarımlar kurum yöneticilerine ve sınava giren öğrencilere raporlanmaktadır.

Çalışma içerisinde 6 farklı sınavdan ilişki kuralları çıkarılmıştır. Bu sınavlar minimum 100 maksimum 150 sorudan oluşurken sınavlara minimum 150 maksimum 198 öğrenci katılmıştır. Bu değerlerle birlikte incelenen toplam veriler bir sınıf için minimum 17600 adet, maksimum 24750 adettir. Toplam incelenen veri sayısı 130271'dir.

Genel çıkarımlar:

1. Sınavlarda sorulan soruların zorluk derecelerinin genellikle 1 olması bu değer ile ilgili çok fazla ilişki çıkmasına sebep olmuştur.
2. Sınavlarda sorusu bulunan eğitimcilerin fazla olması eğitimci başına düşen soru oranının düşmesine ve bu da eğitimcilerin %30'luk destek seviyesinin altında kalmasına sebep olmuştur.
3. Sınavlarda bulunan soruların öğrenim hedeflerinin fazla olması öğrenim hedefi başına düşen soru oranının düşmesine ve bu da öğrenim hedeflerinin %30'luk destek seviyesinin altında kalmasına sebep olmuştur.
4. Tüm sınavlarda kısa uzunluktaki soruların diğer sorulara oranının yüksek olduğu ve bu soruların doğru cevaplanma oranının diğer sorulara göre çoğu zaman yüksek olduğu görülmektedir.
5. 1. sınıfta yapılan sınavlarda ön sınava katılmanın, sınav sonuçlarına doğrudan bir etkisi gözlemlenmemiştir. 2. ve 3. sınıflarda yapılan sınavlarda, ön sınava katılan öğrencilerin katılmayan öğrencilerden daha yüksek notlar aldıkları görülmektedir.
6. Zorluk derecesi 1 olan soruların cevaplanma oranının sınav ortalamaları ile doğrudan örtüştüğü görülmektedir.
7. İncelenen sınavların %67'sinde kadın öğrencilerin erkek öğrencilere göre daha fazla soruyu doğru cevapladığı görülmektedir.
8. Kitapçık türünün sınavlar üzerinde herhangi bir etkisinin olmadığı gözlemlenmiştir.
9. Sorulara ait cevap seçeneklerinin işaretlenme durumlarının sınavlar üzerinde herhangi bir etkisinin olmadığı gözlemlenmiştir.

Genel olarak yapılan çıkarımlar tüm sınavlar için örtüşmektedir. Buna karşı sınava özgü sonuçların da çıktığı gözlemlenmiştir. Sınav sürecinin en önemli adımı olan soru hazırlama sürecinde soru özelliklerinin daha özenilerek hazırlanması çıkarımların çok daha sağlıklı olmasını sağlayabilir. Özellikle soruların zorluk dereceleri daha dikkatli seçilirse, öğrencilerin bu soruları cevaplama oranları çok daha güvenilir sonuçlar verebilir.

Çizelge 4.1. Apriori algoritmasının uygulandığı bilgisayarın özellikleri

Özellik	
İşlemci	Intel Core i5-4570T CPU @ 2.90GHz x 4
Bellek	12 GB DDR3
Disk	240 GB SSD
İşletim Sistemi	Ubuntu 14.04(Trusty) 64 bit
Linux Çekirdeği	3.13.0-100-generic
Ruby on Rails	5.0.0
MySQL	5.5

Çizelge 4.2. Apriori algoritmasının veri setleri üzerindeki çalışma süreleri

Sınav	**Veri Sayısı**	**Çalışma Süresi**
1. sınıf 1. kurul sonu sınavı	17600	0.312s
1. sınıf yılsonu sınavı	22646	0.326s
2. sınıf 1. kurul sonu sınavı	24750	0.348s
2. sınıf yılsonu sınavı	23125	0.322s
3. sınıf 1. kurul sonu sınavı	19500	0.306s
3. sınıf yılsonu sınavı	22650	0.316s

Algoritmanın üzerinde çalıştırıldığı bilgisayarın özellikleri Çizelge 4.1'de verilmiştir. 6 farklı sınav üzerinde %30 destek ve %60 güven değerleriyle uygulanan algoritmanın çalışma süreleri de Çizelge 4.2'de verilmiştir. Çizelge 4.2'ye bakıldığında Apriori algoritmasının bu veriler içerisinden çok küçük sürelerde birliktelik kurallarını çıkarılabildiği görülmektedir.

Sınav verileri içerisinden birliktelik kurallarının çıkarılması için Apriori algoritması seçilmiştir. Bu algoritmanın uygulama alanları oldukça geniştir. Apriori algoritmasının seçilmesindeki en önemli sebep ise uygulama maliyetinin yüksek olmasına rağmen elde edilen sonuçların güvenirliliğidir.

Apriori algoritmasının kullanımı sonucunda elde edilen çıkarımlardan yola çıkılarak, bu algoritmanın bu veriler üzerinde kullanımının olumlu ve olumsuz sonuçları birkaç noktada değerlendirilebilir. Bunlardan en önemlisi bellek kullanımı sorunudur. Apriori algoritması bellek gereksinimi yüksek olan bir algoritmadır. Bu noktada hazırlanan eklentide tüm veri bir anda üretilip eklentiye verilmemiş, sadece gerektiğinde veri tabanından alınmıştır. Böylece on binlerce satırlık veri için kullanılacak bellek alanı yerine düşük bir bellek alanı ile algoritmanın çalışması sağlanmıştır.

Bu algoritmanın temel amacı belirlenen destek değeri üzerinde kalan ve yine belirlenen bir güven oranını yakalamış veri gruplarını bulmaktır. Veriler üzerinde ilgili veri tabanının herhangi bir sütunun, yani veriye ait herhangi bir özelliğin veriler içerisindeki istatistiksel durumunu ortaya koyacak bir sonuçlar basit veri tabanı sorguları ile elde edilebilir. Elde edilen bu sonuçlar verinin her bir özelliği için incelenir ve veriye ait özellik ile birlikte incelenir. Hangi verinin oransal olarak yüksek ve düşük olduğu bu sorgulardan elde edilen sonuçlarda gözlemlenebilir. Apriori algoritması ile verinin tüm özellikleri ile bir arada bulunma durumları elde edilir. Burada ise eşik seviyeleri kullanılarak sadece belirli bir oran üzerinde bir arada bulunan veriler elde edilebilir. Ancak istatistiksel çıktılarda elde edilen düşük oranlara sahip özellikler algoritma çıktısından elde edilemez. Bu durum Apriori algoritmasının veriler üzerindeki bir eksikliği olarak görülebilirken, algoritmanın asıl amacının belirli oranların üzerindeki veri birlikteliklerini bulmak olduğundan, burada asıl söylenmesi gereken oluşan bu durum itibariyle düşük oranlara sahip veriler de elde edilmek isteniyorsa Apriori algoritmasının kullanılmaması gerektiğidir.

KAYNAKLAR

Agrawal R, Imielinski T & Swami A (1993). Mining association rules between sets of items in large databases. *ACM SIGMOD Conference on Management of Data, Washington.* 207-229. doi: 10.1145/170035.170072

Agrawal R & Srikant R (1993). Fast algorithms for mining association rules. *Conference on Very Large databases*, Santiago, Chile. 487-499

Akpınar H, (2000). Veri tabanlarında bilgi keşfi ve veri madenciliği. İstanbul Üniversitesi İşletme Fakültesi Dergisi, Cilt 29, Sayı 1, 1-22

Argüden Y & Erşahin B (2010). Veri Madenciliği. ISBN: 978-975-93641-9-9

Silahtaroğlu, G (2014). Veri Madenciliği. ISBN: 978-975-6797-81-5

Savaş S, Topaloğlu N & Yılmaz M (2012). Veri Madenciliği ve Türkiye'deki Uygulama Örnekleri. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi* Yıl:11 Sayı: 21 Bahar s. 1-23

Mishra R & Choubey A (2012). Comparative Analysis of Apriori Algorithm and Frequent Pattern Algorithm for Frequent Pattern Mining in Web Log Data. *International Journal of Computer Science and Information Technologies*, Vol. 3 (4), 4662 – 4665, ISSN: 0975-9646

Bansal D & Bhambhu L (2013). Execution of APRIORI Algorithm of Data Mining Directed Towards Tumultuous Crimes Concerning Women. *International Journal of Advanced Research in Computer Science and Software Engineering*, ISSN: 2277 128

Tanna R & Ghodasara Y (2014). Using Apriori with WEKA for Frequent Pattern Mining. *International Journal of Engineering Trends and Technology* (IJETT) – Volume 12 Number 3

Çöllüoğlu Ö & Özdemir S (2013). Analysis of Gifted Students' Interest Areas Using Data Mining. *Journal of Gifted Education Research*, 1(3), 213-226

Ulaş M A (1999). Market Basket Analysis For Data Mining. Master Thesis, Computer Engineering, Boğaziçi University

Toprak S (2004). Data Mining For Rule Discovery in Relational Databases. Master Thesis, Middle East Technical University, Computer Engineering

Sıramkaya E (2005). Veri Madenciliğinde Bulanık Mantık Uygulaması. Yüksek Lisans Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü

Toprak S D (2005). A New Hybrid Multi-Relational Data Mining Technique. Master Thesis, Middle East Technical University, Computer Engineering

Kılınç Y (2009). Mining Association Rules For Quality Related Data In An Electronics Company. Master Thesis, Middle East Technical University, Industrial Engineering

Yıldız B (2010). Impacts Of Frequent Itemset Hiding Algorithms On Privacy Preserving Data Mining. Master Thesis, İzmir İnstitute of Technology, Computer Engineering

Gülen Ö & Özdemir S (2013). Veri Madenciliği Teknikleri ile Üstün Yetenekleri Öğrencilerin İlgi Alanlarının Analizi. *Journel of Gifted Education research*, 1(3), 213-226

Karaibrahimoğlu A & Genç A (2014). Meme Kanseri Verisinde Apriori Algoritması ile Kural Çıkarma. *Selçuk Tıp Dergisi*

Özçakır F C & Çamurcu A Y (2007). Birliktelik Kuralı Yönetimi için Bir Veri Madenciliği Yazılımı Tasarımı ve Uygulaması. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi* Yıl: 6 Sayı:12 Güz 2 s. 21-37

Karabatak M. & İnce M C (2004). Apriori Algoritması ile Öğrenci Başarı Analizi. *Elektrik Elektronik ve Bilgisayar Mühendisleri Sempozyumu*, Bursa

Taş M & Adak M F (2014). Öğrencilerin Staj Verileri Üzerine Uygulanan Apriori Algoritması ile Birliktelik Kurallarının Çıkarılması ve Staj Eğiliminin Belirlenmesi. *Akademik Platform*

Erdem S & Özdağoğlu G (2008). Ege Bölgesindeki bir Araştırma ve Uygulama Hastanesinin Acil Hasta Verilerinin Veri Madenciliği İle Analiz Edilmesi. *Anadolu Üniversitesi Bilim ve Teknoloji Dergisi*, Cilt/Vol.:9-Sayı/No: 2 : 261-270

Erpolat S (2012). Otomobil Yetkili Servislerinde Birliktelik Kurallarının Belirlenmesinde Apriori ve FP-Growth Algoritmalarının Karşılaştırılması. *Anadolu Üniversitesi Sosyal Bilimler Dergisi,* Cilt 12, Sayı 2, 137-146

EKLER

EK 1. RUBY PROGRAMLAMA DİLİ İLE APRIORI ALGORİTMASI

```
class Apriori
  Candicate = Struct.new(:left, :right, :support, :confidence,
:count)

  attr_reader :min_support, :min_confidence, :results

  def initialize(exam, min_support, min_confidence, columns = {})
    @exam = exam
    @min_support = min_support
    @min_confidence = min_confidence
    @columns = columns
    @results = []
    @disciplines = Discipline.cache_all.index_by(&:id)
    @systems = Discipline.cache_all.index_by(&:id)
    @info_degrees = I18n.t('efp.degrees')
    @is_corrects = %w(Doğru Yanlış Boş)
    @answer_keys = ('A'..'E').to_a << 'Boş'
    @booklet_types = ('A'..'D').to_a
    @genders = I18n.t('user.genders')
    @pre_exam = %w(Hayır Evet)
  end

  def analyze
    # Veri tabanındaki toplam veri sayısı
    record_length = @exam.exam_analyzes.count.to_f
    # Veri tabanında bulunan tüm sütunların veri grubuna göre
sayıları
    column_counts = {}
    # Veri seti kümesi olmaya aday kümeler
    candicates = []
    # Minimum veri seti kümesi eleman sayısı
    step = 2

    @columns.each do |column, mean|
```

```
      # Sütun adına göre veri sayısını bul
      counts = @exam.exam_analyzes.group(column).count

      # min_support değerini aşan kayıtları seç
      counts = counts.select do |_, count|
        (count / record_length) > @min_support
      end

      if counts.presence
        counts.each do |value, count|
          candicates << Candicate.new(
            { column => value },
            {},
            count / record_length,
            0,
            count
          )
        end

        column_counts[column] = counts
      end
    end

    begin
      tmp_results = []

      column_counts.each do |column, values|
        values.each do |value, value_count|
          candicates.each do |candicate|
            left_params = { column => value }
            right_params = candicate.left.merge(candicate.right)
            query = right_params.merge(left_params)

            # Sol ve sağ tarafda aynı anahtar kelimenin bulunma
durumunu geç
            next unless query.length == step

            # Bu şartın veri tabanında bulunma sayısı
            query_count = @exam.exam_analyzes.where(query).count
```

```
            # Bu şartın destek değerini hesapla
            support = (query_count / record_length)

            # Bu şartın güven değerini hesapla
            confidence = (query_count / value_count.to_f)

            # left-right durumu kontrol et
            if support >= @min_support and confidence >=
@min_confidence
              tmp_results << Candicate.new(
                left_params,
                right_params,
                support,
                confidence,
                query_count
              )
            end

            # Sadece başlangıç durumu için adımı geç
            next if step == 2

            # Bu şartın güven değerini hesapla
            new_confidence = (query_count / candicate.count.to_f)

            # left-right durumu ile right-left durumunun güven
değerleri aynı ise
            # veri tekrarı oluşmuş demektir ve bu istenmeyen bir
durumdur.
            next if confidence == new_confidence

            # right-left durumu kontrol et
            if support >= @min_support and confidence >=
@min_confidence
              tmp_results << Candicate.new(
                right_params,
                left_params,
                support,
                new_confidence,
                query_count
              )
            end
```

```
          end
        end
      end

      candicates = tmp_results.dup
      @results += tmp_results
      step += 1
    end while tmp_results.presence
  end

  def print
    @results.each do |result|
      puts "(#{print_format(result.left)}) =>
(#{print_format(result.right)}) | Destek =
#{result.support.round(2)}, Güven = #{result.confidence.round(2)}"
    end
  end

  private

  # Elde edilen nesne kümelerinin sol ve sağ blokların
  # formatlı çıktısını üretir
  def print_format(side)
    side.map do |column, value|
      "#{@columns[column]} = #{column2value(column, value)}"
    end.join(' & ')
  end

  # Veri tabanındaki sütunun adından ve değerinden
  # o alanın verisine ulaşır
  def column2value(column, value)
    case column
    when :user_id
      User.find(value).name
    when :question_id
      Question.find(value).title
    when :answer_id
      QuestionChoice.find(value).content
    when :user_answer_id
      value == 0 ? 'Boş' : QuestionChoice.find(value).content
    when :author_id
```

```
      User.find(value).name
    when :discipline_id
      @disciplines[value].name
    when :system_id
      @systems[value].name
    when :block_id
      Block.find(value).name
    when :ncc_id
      Ncc.find(value).name
    when :efp_id
      Efp.find(value).name
    when :info_degree
      @info_degrees[value]
    when :is_correct
      @is_corrects[value]
    when :answer_key
      @answer_keys[value]
    when :booklet_type
      @booklet_types[value]
    when :gender
      @genders[value]
    when :pre_exam
      @pre_exam[value]
    else
      value
    end
  end
end
```

ÖZGEÇMİŞ

Adı ve Soyadı	: Muhammed Emin Eker
Doğum Yeri	: İskilip
Doğum Tarihi	: 1990
Yabancı Dili	: İngilizce

Eğitim Durumu

Lise	: Çorum Fen Lisesi (2009)
Lisans	: Ondokuz Mayıs Üniversitesi (2013)
Yüksek Lisans	: Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı (Şubat 2014, Kasım 2016)

Çalıştığı Kurumlar ve Yıl

USTAD Bilgi Teknolojileri / SAMSUN (2012-2013)

PİVOT Bilgi Teknolojileri / SAMSUN (2014-2016)

Yayınlar

Eker M E, Oktaş R & Kayhan G (2016). Apriori Algoritması ve Türkiye'deki Örnek Uygulamaları. *Akademik Bilişim Konferansı*, Aydın

Eker M E, Oktaş R & Kayhan G (2016). Apriori Algoritmasının Farklı Veri Kümelerine Uygulanması. *Akademik Bilişim Konferansı*, Aydın