



BİLECİK ŞEYH EDEBALI
ÜNİVERSİTESİ

**BİLECİK
ŞEYH EDEBALI ÜNİVERSİTESİ**

**Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Ana Bilim Dalı**

**BİRLİKTELİK KURAL ÇIKARIM ALGORİTMALARI
KULLANILARAK MARKET SEPET ANALİZİ**

**Emrah TOKYÜREK
Yüksek Lisans**

**Tez Danışmanı
Doç. Dr. Uğur YÜZGEÇ**

**BİLECİK, 2019
Ref. No: 10261472**



BİLECİK ŞEYH EDEBALI
ÜNİVERSİTESİ

**BİLECİK
ŞEYH EDEBALI ÜNİVERSİTESİ**

**Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Ana Bilim Dalı**

**BİRLİKTELİK KURAL ÇIKARIM ALGORİTMALARI
KULLANILARAK MARKET SEPET ANALİZİ**

**Emrah TOKYÜREK
Yüksek Lisans**

**Tez Danışmanı
Doç. Dr. Uğur YÜZGEÇ**

BİLECİK, 2019



BİLECİK ŞEYH EDEBALI
ÜNİVERSİTESİ

**BİLECİK
SEYH EDEBALI UNIVERSITY**

**Graduate School of Sciences
Department of Computer Engineering**

**MARKET BASKET ANALYSIS
BY USING ASSOCIATION RULE MINING ALGORITHM**

**Emrah TOKYÜREK
Master's Thesis**

**Theis Advisor
Assoc. Prof. Uğur YÜZGEÇ**

BİLECİK, 2019



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS

JÜRİ ONAY FORMU

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun 16/05/2019 tarih ve 27/4... sayılı kararıyla oluşturulan jüri tarafından 21.06/2019 tarihinde tez savunma sınavı yapılan Emrah TOKYÜREK'in "BİRLİKTELİK KURAL ÇIKARIM ALGORİTMALARI KULLANILARAK MARKET SEPET ANALİZİ" başlıklı tez çalışması Bilgisayar Mühendisliği Ana Bilim Dalında YÜKSEK LİSANS tezi olarak oy birliği/oy çokluğu ile kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI):

Doç. Dr. Uğur YÜZGEÇ

ÜYE:

Prof. Dr. Cihan KARAKUZU

ÜYE:

Dr. Öğr. Üyesi Oktay AYTAZ

ONAY

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun .../.../..... tarih ve/..... sayılı kararı.

İMZA/ MÜHÜR

TEŐEKKÖR

Çalıőmalarım boyunca beni yönlendiren, yardım, katkı, rehberlięi sayesinde yol alabildięim danışman hocam Doç. Dr. Uęur YÖZGEÇ'e, varlıklarından gurur duyduğum ve güç aldığım çocuklarım Mehmethan TOKYÜREK'e ve Elif TOKYÜREK'e, desteęini hiç esirgemedен hep yanımda olan eşim Ayfer TOKYÜREK'e teşekkür ederim.

BEYANNAME

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kılavuzu'na uygun olarak hazırladığım bu tez çalışmasında, tez içindeki tüm verileri akademik kurallar çerçevesinde elde ettiğimi, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun olarak sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu Üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

/07/2019

Emrah TOKYÜREK

BİRLİKTELİK KURAL ÇIKARIM ALGORİTMALARI KULLANILARAK MARKET SEPET ANALİZİ

ÖZET

Mağazalarda ürün çeşitliliği ile oluşan veri yığınları, mağaza depolarında mal girişi ve çıkışı esnasında yoğunlukların yaşanmasına neden olmaktadır. Bu yoğunluğun iş gücünü, zamanı, enerji tüketimini ve satışları olumsuz etkilediği düşünülmektedir.

Veri madenciliği çalışmaları, sektörlerde taleplerin belirlenmesi, taleplere en uygun çözümü bulma ve geliştirme konusunda çözümler üretmektedir.

Birliktelik kural analizi algoritmaları veri madenciliği alanında veri kümeleri veya veriler arasındaki ilişkileri ortaya çıkarmak için sıklıkla kullanılan yöntemlerin başında gelmektedir. Literatür taraması sonucu elde edilen bilgiler ışığında birliktelik kural analizi algoritmaları içerisinde en yaygın kullanılan algoritmaların Apriori ve Fp-Growth algoritmaları olduğu tespit edilmiştir.

Bu tez çalışmasında, bir mağaza deposuna ait 4625 hareketlilik ve 106 üründen oluşan bir veri tabanı üzerinden birliktelik kural çıkarım algoritmaları kullanılarak depodan birlikte çıkma eğilimi olan ürünler ortaya çıkarılmıştır. Apriori ve FP-Growth algoritmaları uygulama üzerinde karşılaştırmalı olarak incelenmiş ve mağaza deposunun giriş-çıkış için en uygun hale getirilmesi amaçlanmıştır.

Bu tez çalışması toplam, beş bölümden oluşmaktadır: giriş, veri madenciliği hakkında genel bilgiler, birliktelik kural analizi hakkında detaylı bilgiler, uygulama bölümü ve sonuç.

Anahtar Kelimeler: Veri Madenciliği; Birliktelik Kural Analizi; Apriori Algoritması; FP-Growth Algoritması.

MARKET BASKET ANALYSIS BY USING ASSOCIATION RULE MINING ALGORITHM

ABSTRACT

The mass of products produced by the product variety in the stores causes intensities during the goods entry and exit in the store depots. It is thought that this intensity affects labor, time, energy consumption and sales negatively.

Data mining works produce solutions to identify demands in sectors, to find and develop the most suitable solution to the demands.

Association rule analysis algorithms are one of the most frequently used methods to reveal the relationships between data sets or data in the field of data mining. In the light of the data obtained from the literature review, it was found that the most commonly used algorithms in association rule analysis algorithms are Apriori and Fp-Growth algorithms.

In this thesis, 4625 mobility and 106 products belonging to a store warehouse, products with a tendency to coexist from the warehouse by using the rule extraction algorithms. Apriori and FP-Growth algorithms are examined comparatively on the application and it is intended to optimize the store warehouse for entry-exit.

This thesis consists of a total of five chapters: introduction, general information about data mining, detailed information about the association rule analysis, the application section and the result.

Key Words: Data Mining; Association Rule Analysis; Apriori Algorithm; FP-Growth Algorithm.

İÇİNDEKİLER

	Sayfa No
TEŞEKKÜR	
BEYANNAME	
ÖZET	I
ABSTRACT	II
İÇİNDEKİLER	III
ÇİZELGELER DİZİNİ	VI
ŞEKİLLER DİZİNİ	VII
SİMGELER VE KISALTMALAR DİZİNİ	VIII
1. GİRİŞ	1
2. VERİ MADENCİLİĞİ	4
2.1. Veri Madenciliğinin Tanımı.....	4
2.2. Veri Madenciliği Uygulama Alanları.....	4
2.3. Veri Madenciliğinde Karşılaşılan Zorluklar	5
2.3.1. Veri tabanının boyutu.....	5
2.3.2. Gürültülü ve kayıp veri	6
2.3.3. Boş veri	6
2.3.4. Eksik veri	6
2.3.5. Dinamik veri.....	6
2.3.6. Farklı tipte veri analizi	7
2.4. Veri Madenciliği Süreci: CRISP-DM	7
2.4.1. Veri seçme ve oluşturma süreci	8
2.4.2. Önışleme ve temizleme süreci	8
2.4.3. Veri indirgeme süreci	9
2.4.4. Veri madenciliği programı ve algoritmasını uygulama	9
2.4.5. Sunum, doğrulama ve bilgiyi kullanma	9
2.5. Veri Madenciliği Modelleri.....	10
2.5.1. Tahmin edici modeller	10
2.5.2. Tanımlayıcı modeller	10
2.6. Veri Madenciliği Teknikleri.....	10
2.6.1. Sınıflama ve regresyon.....	10

2.6.2. Kümeleme	11
2.6.3. Ardışık örüntü	13
2.6.4. Birliktelik kuralları madenciliği	13
3. BİRLİKTELİK KURALLARI	14
3.1. Birliktelik Kuralları Tanımı	14
3.2. Birliktelik Kuralları Matematiksel Gösterimi ve Temel Kavramlar	14
3.2.1. Birliktelik kuralları matematiksel modeli	14
3.2.2. Destek ölçütü.....	15
3.2.3. Güven ölçütü	15
3.2.4. Kaldıraç ölçütü	16
3.3. Birliktelik Kurallarının Kullanım Alanları.....	17
3.4. Birliktelik Kuralların Aşamaları	17
3.5. Birliktelik Kural Algoritmaları	18
3.5.1. AIS algoritması	18
3.5.2. SETM Algoritması	19
3.5.3. CD algoritması	20
3.5.4. DD algoritması	20
3.5.5. Apriori algoritması	21
3.5.6. FP-Growth algoritması.....	27
4. MARKET SEPET ANALİZİ UYGULAMASI	35
4.1. Problemin Tespiti	35
4.2 Uygulamada Kullanılacak Teknolojiler	35
4.2.1.WEKA programı	35
4.3. Veri Madenciliği Süreçleri.....	37
4.3.1. Veri setinin oluşturulması	37
4.3.2. Veri setinin temizlenmesi.....	37
4.3.3. Veri seti üzerinde algoritmaların uygulanması	37
4.3.3.1. Veri setinin Apriori algoritması sonuçları	38
4.3.3.2. Veri setinin FP-Growth algoritması sonuçları	41
4.3.4. Farklı güven değerleri ile sonuç analizi	43
4.3.5. Farklı destek değerleri ile sonuç analizi.....	44
4.3.6. Apriori ve FP-Growth algoritmalarının karşılaştırılması.....	45

5. SONUÇLAR	47
KAYNAKLAR	48
ÖZ GEÇMİŞ.....	

ÇİZELGELER DİZİNİ

	Sayfa No
Çizelge 2.1. Veri madenciliği kullanım alanları ve amaçları.	5
Çizelge 3.1. Örnek ürün satış tablosu.	16
Çizelge 3.2. Örnek veri seti.	25
Çizelge 3.3. C1 aday kümesi.	25
Çizelge 3.4. L1 aday kümesi.	26
Çizelge 3.5. C2 aday kümesi.	26
Çizelge 3.6. L2 aday kümesi.	26
Çizelge 3.7. C3 aday kümesi.	27
Çizelge 3.8. Örnek veri tabanı.	30
Çizelge 3.9 Örnek uygulama ürün destek değerleri.	30
Çizelge 3.10. Örnek uygulama sıralı liste.	31
Çizelge 3.11. Sıralanmış liste.	31
Çizelge 3.12. Örnek uygulama kural listesi.	32
Çizelge 4.1. Apriori algoritması sonucu bulunan kurallar.	38
Çizelge 4.2. Uygulama veri tabanı kesiti.	38
Çizelge 4.3. Fp-Growth algoritması sonucu bulunan kurallar.	41
Çizelge 4.4. Apriori algoritması ile farklı güven değerleri sonuçları.	43
Çizelge 4.5. FP-Growth algoritması ile farklı güven değerleri için süre analizi.	43
Çizelge 4.6. Farklı destek değerleri ile sonuç analizi.	44

ŞEKİLLER DİZİNİ

	Sayfa No
Şekil 2.1. Veri tabanı bilgi keşif süreci adımları.	7
Şekil 2.2. Kümeleme algoritması öncesi veri seti.	12
Şekil 2.3. Kümeleme algoritması sonrası veri seti.	12
Şekil 3.1. Birliktelik kural çıkarım algoritmaları.	18
Şekil 3.2. AIS algoritması özet kodu (Zerman, M., 2018).	19
Şekil 3.3. Apriori algoritması birleştirme özelliği.	21
Şekil 3.4. Apriori algoritması budama özelliği.	22
Şekil 3.5. Apriori algoritması özet kodu.	23
Şekil 3.6. Apriori algoritması akış diyagramı.	24
Şekil 3.7. FP-Growth algoritması özet kodu.	28
Şekil 3.8. FP-Growth algoritması akış diyagramı.	29
Şekil 3.9. Fp-Growth budama özelliği.	32
Şekil 3.10. FP-Growth algoritması süre analizi.	34
Şekil 4.1. WEKA ara yüzü.	36
Şekil 4.2. WEKA süpermarket datası.	36
Şekil 4.3. Apriori algoritması uygulama kural sonuçları.	39
Şekil 4.4. Apriori algoritması süre analizi.	40
Şekil 4.5. FP-Growth algoritması uygulama kural sonuçları.	41
Şekil 4.6. FP-Growth algoritması süre analizi.	42
Şekil 4.7. Değişken güven değerleri için kural oluşturma süreleri.	44
Şekil 4.8. Uygulama ürün destek sayıları.	45

SİMGELER ve KISALTMALAR DİZİNİ**Simgeler**

c: Güven Değeri

C_k : k Adet Veri İçeren Aday Nesne küme

I: Veri Seti

L_k : k Adet Veri İçeren Sık Geçen Nesne küme

N: Gözlem Sayısı

s: Destek Değeri

T: Veri Tabanındaki İşlemler

$X \Rightarrow Y$: X'in Sağlandığı Durumlarda Y'nin de Sağlanma Durumu

Kısaltmalar

ARFF: Attribute Relationship File Format

CRISP-DM: Cross Industry Standart Process for Data Mining

CSV: Comma-Separated Values

FP-Growth: Frequent Pattern Growth

LDA: Linear Discriminant Analysis

MSA: Market Sepet Analizi

PCA: Principal Component Analysis

TID: İşlem Numarası

VM: Veri Madenciliği

VTBK: Veri Tabanı Bilgi Keşfi

WEKA: Waikato Environment for Knowledge Analysis

1. GİRİŞ

Günümüzde teknolojinin de gelişmesi ile birlikte marketlerde, internette hemen hemen her yerde karşılaştığımız tüm ürünlerde çeşitlilik artmış ve yüz binleri bulmuştur. Örneğin; elektronik bir mağazadan her hangi bir ürünü satın almayı düşündüğümüzde, bir markanın bile bize sunduğu çok fazla ürün seçeneği bulunmaktadır. Ürün çeşitliliğinin artması ürünleri tüketici ile buluşturma konusunda hizmet sunan firmaların işini daha da zor hale getirmektedir. Farklı ürünlerin müşterilere sunulması kadar, çeşitliliği olan tüm ürünleri mağaza depolarında bulundurabilmekte bir o kadar önem arz etmektedir. Tüketiciler, ürünlerin hem istenilen kalitede olmasını hem de diğer mağazalara göre daha ucuz olmasını istemektedirler. Bu haklı talebi karşılamak mağaza sahiplerine düşmektedir. Talepleri yerine getirmek için talep gören ürünleri depoda hali hazırda bulundurmak gerekir. Ürün çeşitliliği hesaba katıldığı zaman bunun oldukça güç olduğunu tahmin etmek gerekir. Devasa bir depoya sahip olmakta oldukça güçtür. Böyle büyük bir depoya sahip olmaktan daha önemlisi, birlikte aynı zamanda tüketilen ürünlerin bilgisine sahip olabilmektir. Marketler ve/veya firmalar satmış oldukları ürünlerin listelerini veri tabanlarında tutarlar. Bu verileri kullanarak, veriler arasındaki ilişkileri, müşteriler ile ürünler arasındaki birliktelikleri bulmak mağaza depolarının düzenlenmesi konusunda önemli bilgiler vermektedir. Bu bilgiler depoların daha işlevsel kullanılmasını, daha çok ürün saklanabilmesine imkân sağlar. Bu bilgiler ışığında düzenlenen depolar; iş gücünde kolaylık, ürünlerin mağaza raflarında hali hazırda bulunmasında kolaylık sağlar. Bu sonuçlar mağazanın müşteri memnuniyetini artıracak gibi satışlarda da artış sağlayacaktır.

Veri Madenciliği (VM), büyük veri tabanları içerisinde ihtiyaç duyulan, istenen faydalı bilginin cımbızlanması işidir. Cımbızlama işlemi; matematik disiplini, modelleme teknikleri, veri tabanı teknolojileri ve bilgisayar programları kullanılarak yapılır. Literatürde VM kullanılarak verilerin incelendiği oldukça fazla çalışma vardır. Han, J. & Kamber, M. & Pei, J. (2012) çalışmasında VM teknikleri ve kavramları üzerine detaylı incelemelerde bulunmuştur. VM işlemi, büyük veri tabanlarından desenleri, düzensizliklerden birliktelikleri çıkarmakta kullanılmasında işe yaradığı gibi e-alışveriş sitelerinde daha önceki müşterilerin alışverişlerinden bilgiler çıkararak müşterilere ürün önerisinde bulunur. Bazı e-mağazalarda “bu ürünü alanlar şu ürünü de aldı” gibi yönlendirme mesajları sitede gezinenlere sunulmaktadır.

Veri madenciliği alanında farklı yöntemler kullanılmaktadır. Kullanılan yöntemlerden biri de birliktelik kural analizidir. Ürün blokları arasında veya ürünler arasında anlamlı ilişkiler tespit edilerek, kurallar çıkarılmasına yardımcı olmaktadır. Birliktelik kuralları belirlemede kullanılmak üzere birçok algoritma bulunmaktadır. En çok kullanılan ve bilinen algoritmalar Apriori ve FP-Growth algoritmalarıdır Kantardzic, M. (2011).

Literatürde bu iki algoritma kullanılarak çeşitli çalışmalar yürütülmüştür.

Patil, B. M. & Joshi, R.C. & Tonsniwal, D. (2010) çalışmasında diyabet hastalığının erken teşhisi üzerine incelemeler yapmıştır. Diyabet hastalarının genel değerlerini baz alarak, yeni hasta bilgilerini birliktelik kural algoritmaları kullanarak çıkarımlarda bulunmuş ve doktorlara ön bilgi sunumu sağlamıştır.

Surendiran, R. & Rajan, K.P. & Sathish Kumar, M. (2010) FP-Growth algoritması kullanarak müşterilerin cinsiyet, yaş, gelir vb. bilgiler ışığında hangi mobil servisi tercih ettikleri konusunda kurallar oluşturmuşlardır.

Erpolat, S. (2012) çalışmasında hem Apriori hem de FP-Growth algoritmaları ile otomotiv sektöründe faaliyet gösteren yetkili servisin müşterilerine ait alış-veriş bilgilerini kullanarak müşterilerin eğilimlerini bulmaya çalışmıştır.

Rong, J. & Vu, H. & Law, R. & Li, G. (2012) internette forum ve online iletişim sitelerinde ağızdan ağıza iletişim olarak adlandırılan eWOM iletişim tekniğinin Hong Kong'da bir turizm şirketinin verileri üzerinde etkisini birliktelik kural çıkarım algoritmaları kullanarak incelemiştir. Çıkan kurallardan faydalanarak turist sayısında artış sağlanmış ve daha etkin pazarlama stratejileri belirlenmiştir.

Prasanna, S. & Ezhilmaran D., (2013) borsadaki yatırımcıların, hisse senetlerinin geçmiş zamanlardaki fiyatlarından elde edilen bilgiler ışığında ne zaman, hangi hisse senedi alınabileceği konusunda çalışma yapmışlardır.

Verma, A. & Khan, S.D, & Maiti, J. & Krishna, O.B. (2014) bu çalışmada Hindistan'da bir çelik tesisinde iş kazası, yaralanma, mal hasarı gibi olaylara etki eden durumları tespit etmek için birliktelik kural çıkarım algoritmalarını kullanılmış ve standart çalışma prosedürlerin yetersiz olması, çalışanların eğitimsiz, dikkatsiz ve yorgun olmaları bu kazaları artırdığı tespitine ulaşmıştır.

Söylemez, İ. & Doğan, A. & Özcan, U. (2016) Ankara ilinde bir yıl içerisinde gerçekleşen trafik kazalarını veri tabanı olarak kullanmış ve Apriori algoritması ile il

genelinde kaza oranı tespit edilerek, farklı yollarda ne tür önlemlerin alınması gerektiği vurgulanmıştır.

Özçalıcı, M. (2017) yaptığı çalışmada ikinci el piyasasında satışa çıkartılan araçlara ait durum değerlendirmesi yapmaktadır ve hangi özelliklerin piyasadaki araçların çoğunda birlikte bulunduğu kuralını oluşturmaktadır.

İncelenen tüm çalışmalarda veriler arasındaki birliktelikler tespit edilerek kurallar oluşturmak istenmiştir.

Bu çalışmanın ikinci bölümünde; Veri madenciliğinin tanımı, uygulama alanları, karşılaşılan zorluklar, veri madenciliği süreci ve teknikleri hakkında bilgiler verilmiştir.

Üçüncü bölümünde; Birliktelik kuralının tanımı, aşamaları, Apriori ve FP-Growth algoritmaları hakkında detaylı bilgiler verilmiş ve basit bir örnek üzerinden anlatılmıştır.

Dördüncü bölümünde; uygulamanın tespiti, veri setinin oluşumu ve indirgenmesi, uygulamanın birliktelik algoritmalarından Apriori ve FP-Growth algoritmalarını ile çalıştırılması ve sonuçlar karşılaştırmalı olarak anlatılmıştır. Uygulama farklı destek ve güven değerleri ile de çalıştırılarak destek ve güven değerlerinin önemine değinilmiştir. Kural oluştururken önem vermemiz gereken diğer bir etkende kural oluşum süresidir. Kural oluşturma süreleri Apriori ve FP-Growth algoritmaları için farklı destek ve güven değerleri için tek tek hesaplanmıştır. Farklı destek ve güven değerlerinde çıkan kural sayıları da incelenmiş ve algoritmalar tüm detaylarıyla karşılaştırılmıştır.

2. VERİ MADENCİLİĞİ

2.1. Veri Madenciliğinin Tanımı

Veri madenciliği; büyük veri yığınları içerisinde keşfedilmeyi bekleyen, önceden keşfedilmeyen, ön his ve tahminlerle ortaya çıkaramadığımız, ayrıca keşfedilmesi durumunda ihtiyacı olan birime büyük fayda sağlayacak değerli bilginin ortaya çıkarılması işlemidir (Silahtaroglu, 2016) .

Veri madenciliği veri tabanı içerisindeki veriler arasında anlamlı ilişkiler çıkarmaktır. Veri Madenciliği farklı ifadelerle de açıklanabilmekte olup bu tanımlar aşağıda belirtilmiştir.

Veri tabanları, veri ambarları veya diğer veri havuzlarında depolanmış veri yığınlarından farklı bilgilerin keşfedilme sürecidir (Han, Kamber, Pei, 2012).

Kullanışlı bilgiyi büyük veri tabanlarından ayırtmak, tanımlamak ve elde etmek için istatistiksel, matematiksel, yapay zeka ve makine öğrenme tekniklerini kullanan bir süreçtir (Ngai, E.W.T., Xiu, L., Chau, D.C.K., 2009).

Veri madenciliği gizli, bilinmeyen, potansiyel ve değerli, ilginç ya da eğitici bilgiye ulaşmada kullanılan araç ve yaklaşımlar olarak ifade edilir (Prasanna, S. & Ezhilmaran D., 2013).

Veri madenciliği önemli, işe yarar, aranan ve kullanışlı veriyi büyük veri tabanlarından ayırma, ortaya çıkarma işlemidir. Bu yüzden veri madenciliği, veri tabanından anlamlı örüntüler veya kurallar elde etmek için geniş bir araştırma alanı olarak görülebilir (Srinivasa, K.G., Venugopal, K.R., Patnaik, L.M., 2007).

2.2. Veri Madenciliği Uygulama Alanları

Veri Madenciliği geniş veri tabanlarının olduğu pazarlama, banka ve sigortacılık, borsa, haberleşme, sağlık ve ilaç, endüstri, bilim ve mühendislik gibi alanlarda kullanılmaktadır. Veri madenciliğinin alanlara göre kullanım amaçları Çizelge 2.1’de verilmektedir.

Çizelge 2.1. Veri madenciliği kullanım alanları ve amaçları.

Kullanılan Alanlar	Kullanım Amaçları
Pazarlama	Market Sepet Analizi
Pazarlama	Satış Tahmini
Pazarlama	Çapraz Satış
Pazarlama	Müşteri Analizi
Banka ve Sigortacılık	Kredi kartı Bilgi Hırsızlığının Tespitinde
Banka ve Sigortacılık	Kredi Talep Değerlendirmesi
Banka ve Sigortacılık	Sigorta Dolandırıcılığının Tespitinde
Banka ve Sigortacılık	Riskli Müşterilerin Belirlenmesinde
Borsa	Piyasa Analizi
Borsa	Alım-Satım Stratejileri Belirlenmesinde
Telekomünikasyon	Mobil Müşteri Analizi
Sağlık ve İlaç	İlaç Etkilerinin Bölgesel Analizi
Sağlık ve İlaç	Tedavi Sürecinin Belirlenmesi
Endüstri	Kalite Kontrol Tespitinde
Bilim ve Mühendislik	Hücre Analizinde
Bilim ve Mühendislik	Uzay Analizinde

2.3. Veri Madenciliğinde Karşılaşılan Zorluklar

Veri madenciliğinde temel öge veri tabanlarıdır. Veri tabanlarının hatasız, eksiksiz, net, kararlı veriler olmaması durumunda bazı sorunlar meydana gelmektedir (Aydoğan F., 2003).

Küçük veri tabanında kısa sürede doğru sonuçlar veren algoritmalar bazen büyük veri tabanlarında bu kadar kararlı çalışmamaktadır. Veri madenciliği sistemlerinin karşılaşılabileceği zorluklar aşağıda madde madde incelenmiştir.

2.3.1. Veri tabanının boyutu

Veri tabanlarında veri boyutunun büyüklüğü, VM sistemlerinde karşılaşılan en sık sorunlardan biridir. Veri tabanlarında veri sayısının fazlalığı VM sistemleri tarafından çıkan kurallara olan güveni artırmaktadır. Fakat VM sistemlerinde geliştirilen uygulamalar küçük veri tabanında kusursuz çalışabilirken, büyük veri tabanları için

uygulandığında hata verebilmektedir. Bu yüzden veri madenciliği yöntemleri ya örnekleme yöntemi yatay/dikey olarak indirgemeli, ya da sezgisel bir yaklaşımla arama uzayını taramalıdır. Dikeyde indirgeme özelliklerin bulunduğu kolonların azaltılması, yatayda indirgeme ise veri alanlarının örneklenmesi çalışmasıdır (Şen, F., 2008).

2.3.2. Gürültülü ve kayıp veri

Veri tabanlarında değerler bazen yanlış olabilir. Bu yanlışlık oranı veri tabanı büyüdükçe artmaktadır. Bu hataların çoğu veri girişini yapan kişiler tarafından yapılmaktadır. Veri girişinde yapılan hatalara ve veri toplanması esnasında oluşan hatalara gürültü denir. Gürültülü veri tabanlarında VM sistemleri bu hatalı verileri bulup sistemin dışına atmalıdır. Atılmaması veya eksik atılması durumunda kurallara olan güven azalır ve algoritmaların kural oluşturma süreci uzar.

2.3.3. Boş veri

Hiç bir değere eşit olmayan veriye boş veri denilir. Boş veriye ilişkisel veri tabanlarında daha çok karşılaşılır. Boş veri hataları için düzeltme çalışmaları yapılmakla birlikte, boş veri hataları karşılanması durumunda bu yokluklar ihmal edilmelidir. Boş değerlere niteliği doğrultusunda, en yakın değerler atanmalıdır.

2.3.4. Eksik veri

Eksik veri, atık nitelikler veya veri setine uygun olmayan veriler olabilir. Veri madenciliğinde her bir veri tabanı için, her veri nesnesi bir özellik gerektirdiğinden eksik veriler sorun oluşturmaktadır (Han, J. & Kamber, M. Pei, J., 2012).

Eksik veriler durumunda yapılması istenenler;

- Eksik veriye sahip olan öğeler veri setinden çıkarılabilir.
- Eksik verinin, kuralı etkilememesi amacıyla veri setinin ortalaması eksik veri değeri yerine kullanılabilir.
- Doğru olan verileri baz alarak en uygun veriler kullanılabilir.

2.3.5. Dinamik veri

Çevrimiçi veri tabanlarının dinamik olması, VM sürecinde önemli sorunlar doğurmaktadır. VM sistemleri var olan veri tabanında uyguladığında uygulamanın performansı oldukça düşer. Çevrimdışı veri tabanlarında ise, değişen veri örüntü kümesine yansımaz.

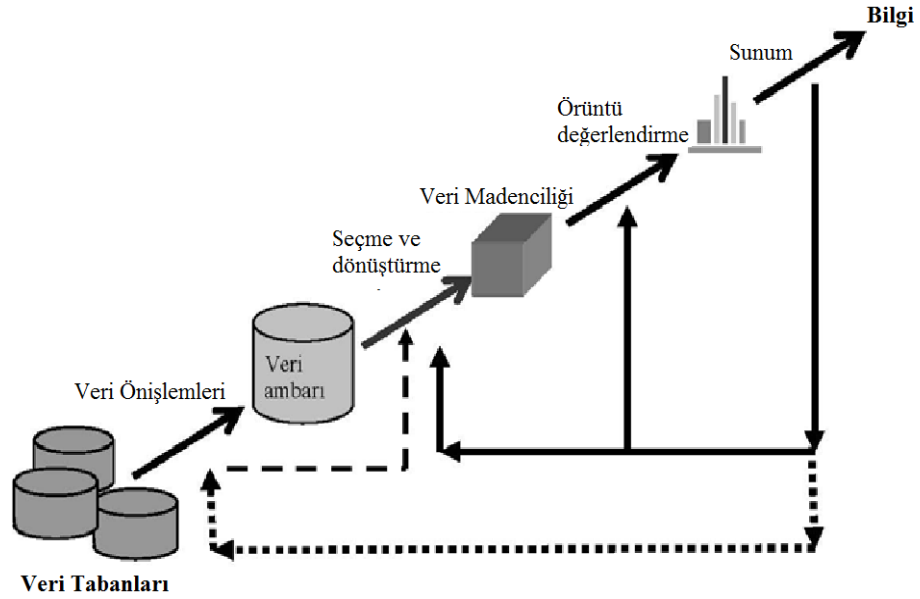
2.3.6. Farklı tipte veri analizi

Gerçek hayatta her türlü sorunla karşılaştığı gibi VM sistemleri veri tabanlarında her türlü farklı tipteki verilerle karşılaşmaktadır. Kullanılan verilerin tutulduğu ortam ilişkisel veri tabanlarında kullanılan tablolar olabilirler. Veri çeşitliliğinin fazla olması, algoritmaların kural oluşturma sürelerini arttırmaktadır.

2.4. Veri Madenciliği Süreci: CRISP-DM

Veri tabanlarından anlamlı bilgilere ulaşmak bir süreçtir. Bilgi keşif süreci adımları Şekil 2.1’de gösterilmiştir.

- Veri Seçme ve oluşturma
- Ön işleme ve temizleme
- Verileri bütünleştirme ve indirgeme
- Veri madenciliği programı ve algoritmasını kullanma
- Sunum, doğrulama ve ilgiyi kullanma



Şekil 2.1. Veri tabanı bilgi keşif süreci adımları (Han, J. & Kamber, M. & Pei, J., 2012).

Bazı durumlarda veriden veriye değişen veri madenciliği süreçleri vardır. Bu farklılıklar veri madenciliği sürecini de olumsuz etkilemektedir. Bu farklılıkları gidermek için veri madenciliği sürecinin standartlaştırılması yönünde çalışmalar

yapılmıştır. Bu çalışmalardan en çok bilinen ve kullanılanı Daimler Chrysler ve SPSS tarafından 1996 yılında oluşturulan süreçtir. Şekil 2.1’de gösterilen bu sürece Sektörler Arası Standart Veri Madenciliği Süreci (CRISP-DM: Cross-Industry Process for Data Mining) adı verilmiştir (<http://crisp-dm.eu/>, erişim: 21.03.2019).

2.4.1. Veri seçme ve oluşturma süreci

Bu süreç veriye karar vermek ile başlar. Hatalı ve süreci yanlış yöne yönlendirecek veriler temizlenir. Eksik veri var ise tahmin yolu ile bulmaya çalışılır, bulunamaz ise silinir. Bu aşama en uzun süren aşama olmakla birlikte fazla iş gücü de gerektirmektedir. Gereksiz veriler silinmediği takdirde, büyük veri kullanarak kural çıkarmak istediğimizde hem sonuca ulaşma süreci gereksiz yere uzayacak hem de önemli kural olarak sayılacak sonuçların tespit edilememesine sebep olacaktır (Akpınar, H., 2000).

2.4.2. Önleme ve temizleme süreci

Verileri hazır hale getirme işlemine önleme denir. Veriler özel sebeplerden dolayı işlemeyen önce, hatalı, eksik veya çok fazla büyük boyutlarda olabilirler. Bu durumda verilerin kullanılabilir hale getirilmesi gerekir.

İşlenmemiş verilerin özelliği, hatalı girilmiş veya zamanla bozulabilen veya değişebilen veriler olmasıdır. Bu özelliklere sahip verilere gürültülü veri denir. Veri seti oluşturmadan önce bu gürültülü verilerin düzeltilmesi gerekir.

Veri Seti içerisindeki hatalı, eksik verilerin sonucu olumsuz etkilememesi için temizlenmesi gerekir. Hatalı ve eksik verilerin sayısı toplam veri setine oranla düşük bir seviyede ise temizleme işlemi kullanılabilir iken hatalı ve eksik verilerin toplam veri setine oranı fazla olması halinde gürültülü verilerin silinmesi sonucu olumsuz etkileyerek kurallara olan güveni azaltır. Bu durumda aşağıdaki yöntemleri uygulamak gerekir Özdemir, (A., & Yalçın, F. A., & Çam, H., 2009).

a) Eksik verilerin tamamlanması; Eksik veriler az miktarda ve bu eksik verilere ulaşarak tamamlamak mümkün durumda ise bu durum gerçekleştirilebilir. Bu yöntem zaman olarak uzun sürecektir.

b) Eksik verilere aynı bilgiyi girmek; Örneğin veri tabanımızda memleketi girilmemiş alanlara boş anlamına gelen bir karakterin girilmesi işlemidir. Bu yöntem doğru veri olmadığı için doğru sonuçlarda çıkarmayabilir.

c) Verilerin ortalama deęerinin eksik verilere yansıtılması; Veri setinde yaşı bilgisi eksik olan bireylere veri havuzundaki kişilerin yaşları ortalamasının girilmesi işlemidir.

d) Kayıp verilerin tahmin edilmesi; Regresyon katsayıları ve denklemleri kullanılarak kayıp verilerin tahmin edilmesidir.

2.4.3. Veri indirgeme süreci

Veri setinden uygun olmayan nitelikteki veriler ve tekrarlayan veriler çıkarılır. Çıkartmadaki amaç daha verimli, hızlı sonuçlar verebilmektir (Özdemir, A., & Yalçın, F. A., & Çam, H., 2009). Bu alanda PCA ve LDA gibi algoritmalar vardır.

PCA (Principle Component Analysis): Temel bileşen analizi, veri kümelerini analiz etmek için kullanılan istatistiksel bir araçtır. Temel bileşen analizinin (PCA) ana fikri, veri kümesinde var olan varyasyonun mümkün olduğunca korunmasını sağlarken, çok sayıda birbiriyle ilişkili deęişkenden oluşan bir veri kümesinin boyutsallığını azaltmaktır (Özkan, D. Y., 2013).

LDA (Linear Discriminant Analysis): Standart LDA, özellik uzayının boyutuna (N) kıyasla sadece sınırlı sayıda gözlem (N) varsa ciddi şekilde azaltılabilir. Bunun olmasını önlemek için, lineer diskriminant analizinin bir temel bileşen analizi ile öncelenmesi önerilir. PCA'da, orijinal veri kümelerinin şekli ve konumu farklı bir alana dönüştürüldüğünde deęişirken, LDA konumu deęiştirmez, sadece daha fazla sınıf ayrımı sağlamaya çalışır ve verilen sınıflar arasında bir karar bölgesi çizer (Adriaans, P., & Zantinge, D., 1997).

2.4.4. Veri madencilięi programı ve algoritmasını uygulama

Bu adımda kullanılacak veri madencilięi yöntemi seçilir. Kullanılacak yöntemi belirleme noktasında daha önceki aşamalar ve bilgi keşfi süreci etkilidir. Bu aşamada amaç kararlı ve yararlı kurallar çıkarana kadar algoritmanın uygulanmasıdır (Aytaç, M. B., & Bilge, H. Ş., 2013).

2.4.5. Sunum, doęrulama ve bilgiyi kullanma

Son aşama da çalışmanın günlük hayatta kullanılmasına odaklanılır. Bu adımda elde edilen kurallar deęerlendirilir. Mümkünse grafięe yansıtılır. Deęerlendirme aşamasında çıkan sonuçlar ışığında yapılması gerekenler planlanır.

2.5. Veri Madenciliği Modelleri

VM'de tahmin edici (predictive) ve tanımlayıcı (descriptive) olmak üzere iki model kullanılmaktadır (Catherina, B. & Rinta, R. E., 2001).

2.5.1. Tahmin edici modeller

Tahmin edici modellerde, sonuçları daha önceden bilinen verilerden hareket edilerek bir model geliştirilmesi ve kurulan bu modelden yararlanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesi amaçlanmaktadır (Akpınar, H., 2000). Örnekle açıklamak gerekirse, bir futbol kulübündeki futbolcunun ligin devre arasına girerken oynadığı oyun ve attığı gol gibi istatistikler veri tabanında toplanarak, ligin sonunda bu oyuncunun kaç gol atacağı tahmininde bulunmaktır.

2.5.2. Tanımlayıcı modeller

Daha önceden herhangi bir veriye sahip olmadan, veri setinin içinden kural çıkarmayı sağlar. Büyük veri tabanlarında bilgileri incelemek, veriler arasında ilişkiler bulabilmek için doğru sorular sorarak hipotezler geliştirmek çok zordur (Zerman, M., 2018). Bu zorluk VM programları sayesinde aşılabılır. Tanımlayıcı modellere örnek olarak kümeleme, birliktelik kuralları tekniklerini örnek olarak verilebilir.

2.6. Veri Madenciliği Teknikleri

Veri Madenciliği konusu ve verilerinin çeşidine göre yöntemlere ayrılır. Sınıflandırma, kümeleme ve birliktelik kuralları çıkarma yöntemleri başlıca yöntemlerdir (Karagöz, N. E., 2007). Bu yöntemlerde kendi içlerinde birçok algoritma içerirler.

2.6.1. Sınıflama ve regresyon

En çok bilinen, kullanılan yöntem olan sınıflama ve regresyon, veri tabanlarında bilinmeyen örüntülerin tespit edilmesinde kullanılır (Özçınar, H., 2006).

Sınıflama; ortak özelliklere sahip değerleri bulmaya çalışırken, regresyon; şuan ki değerlerden çıkarım yaparak yeni değerler bulmaya çalışır.

Örnek verilecek olursa; gram altın fiyatlarının son 10 yıl içerisindeki hareketleri detaylıca incelenerek ve finansal göstergelerden faydalanarak durumu hakkında fikir üretebilmek için veri madenciliği tekniklerinden birisi olan sınıflama ve regresyon tekniği uygulanabilir.

Sınıflama ve regresyon modellerinde kullanılan başlıca yöntemler,

- Karar Ağaçları,
- Yapay Sinir Ağları,
- Genetik Algoritmalar,
- K-En Yakın Komşuluk
- Bellek Tabanlı Yöntemler,
- Naive-Bayes,

2.6.2. Kümeleme

Kümeleme, birbirine benzer özellikte olan verileri kümelere ayırma işlemidir. Veri tabanları büyüdükçe boyutları büyümekte kapladıkları alanda artmaktadır.

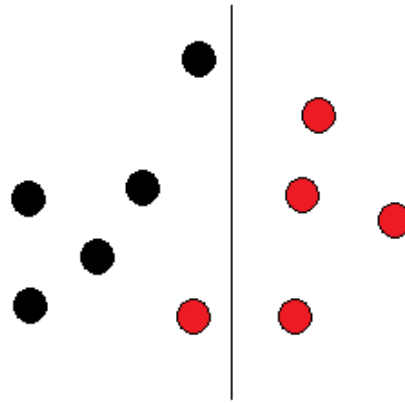
Boyutlar büyüdükçe en iyi VM yöntemleri bile sorun yaşayabilir anlamlı kurallar çıkaramayabilir. Bu sorunu çözümenin en temel yolu büyüyen veriyi küçültmektir. Büyük, karmaşık veri tabanını çözülebilecek büyüklükte daha küçük alt programlara bölmek sorunu çözecektir. Bölmekte karşılaşılan en önemli sorun ise veriler belli bir düzende ve grup halinde olmadığı için bölmek gerekli ama zor bir hal alacaktır. Bu sorunu çözmek için de kümeleme yöntemleri geliştirilmiştir (Özdemir, A., & Yalçın, F. A., & Çam, H., 2009).

Sınıflama işleminde mevcut verilere bakılarak yeni verilerin sınıfları tahmin edilirken, kümeleme yönteminde gruplama yoktur. Oluşmuş kümeler içerisinde öğelerin benzerliği olmalı, kümeler arasında ise benzerlik olmamalıdır.

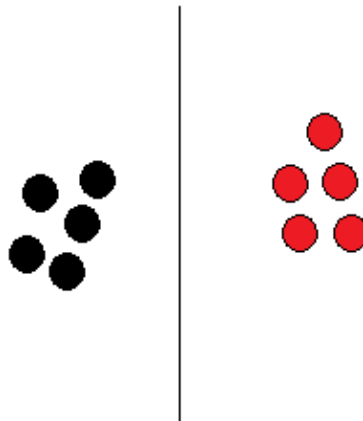
Başlıca kümeleme algoritmaları şunlardır;

- Bölme yöntemleri
- Hiyerarşik yöntemler
- Yoğunluk tabanlı yöntemler
- Izgara Tabanlı Yöntemler
- Model Tabanlı Yöntemler

Kümeleme algoritması çalıştırmadan önce iki farklı ögenin konumu Şekil 2.2'de gösterilmiştir.



Şekil 2.2. Kümeleme algoritması öncesi veri seti.



Şekil 2.3. Kümeleme algoritması sonrası veri seti.

Kümeleme algoritması kümeler içerisinde benzer özellikteki veriler arasındaki uzaklığı azaltırken, farklı özellikteki verilerle arasındaki uzaklığı artırır (Şekil 2.3).

2.6.3. Ardışık örüntü

Ardışık örüntü, birbiri ile ilişkisi olan ve birbirini izleyen zamanlarda oluşan ilişkilerin tanımlanmasında kullanılır (Bramer, M., 2007).

Ardışık örüntünün birliktelik kurallarından ayıran en önemli özelliği zaman kavramının uygulama içerisinde yer almasıdır.

Örnekler aşağıda verilmiştir.

- Şort alan müşterilerin %20'si iki hafta içerisinde şapka almaktadır.
- Araba alan müşterilerin %40'ı bir hafta içerisinde araba paspası almaktadır.

2.6.4. Birliktelik kuralları madenciliği

Birliktelik kuralları, bir veri tabanı içerisindeki veriler arasındaki bağlantıları bulmayı amaçlayan veri madenciliği modelidir.

Birliktelik kurallarının en çok kullanıldığı alan süpermarket sektörüdür. Market sepet analizi (MSA) olarak da isimlendirilir. MSA ürünler arasında birlikte satılma eğilimlerini bularak satışları artırmak amaçlanmıştır. Birliktelik kural analizi çıkış hikayesi temel alındığında iki ürün arasında hiç bir ilişki yokmuş gibi görünse de hikâyeye göre Walmart ismindeki süpermarket mağazalarının analizi şu yöndedir; “Yeni çocuk sahibi olan ebeveynler eğlenmek için vakit bulamayıp cuma günlerini partiye gitmek yerine eve bira alarak evde geçirmek zorunda kalıyorlar. Bu yüzden bebek bezi alan aileler çoğunlukla yanında bira da alırlar.” gibi bir çıkarımda bulunmaktadır. Bu doğrultuda bebek bezi ve birayı yakın yerlere koyarak satışlarını arttırmayı hedeflemektedirler.

Birliktelik kuralının gerçek hayatta kullanılan bir örneği de internet üzerinden satış yapan elektronik mağazalardır. Müşteriye satın aldığı üründen yola çıkarak daha önce bu ürün ile satın alınma olasılığı en yüksek olan diğer ürünleri önermektedir. Günümüzde birçok web sitesi bu yöntemi aktif olarak kullanmaktadır.

Birliktelik kuralları üçüncü bölümde ayrıntılı olarak incelenmiştir.

3. BİRLİKTELİK KURALLARI

3.1. Birliktelik Kuralları Tanımı

Birliktelik kuralı analizi, veri madenciliği yöntemleri içerisinde en yaygın kullanılan tekniklerin başında gelmektedir.

Birliktelik Kuralları Analizi problemi ilk olarak Agrawal ve Swami tarafından 1993 yılında ele alınmış olup ve veri madenciliğinde kullanılan ilk tekniklerdendir (Ateş, Y. & Karabatak, M., 2017). Literatürde “market sepet analizi” olarak anılmaktadır. Market sepet analizi, bir markette müşterilerin satın aldığı ürünler arasındaki ilişkileri, gizli kalmış örüntüleri, birliktelikleri bulmaya dayanmaktadır. Birlikte satın alınan ürünler belirlenerek işletmecinin müşteri davranışlarını belirlemesi, zamana göre satılan ürünlerin analizi, kar miktarını arttırmak için hangi ürünlerin birlikteliğinde indirim yapması gerektiği, ürünleri raflara nasıl yerleştirmesi gerektiği konusunda satış stratejileri oluşturmasına yardımcı olmaktadır (Agrawal, R. & Srikant, R., 1994).

Market alışverişi sırasında ekmek ve su alan müşterilerin %40'ının meyve suyu da aldığı kuralı çıkarılıyor ise, bu ürünlerin birliktelik ilişkisi doğrultusunda müşteriye ürünleri bulmada kolaylık sağlanabilir ve satışını artırmak için ürünlerin raflardaki konumu ayarlanabilir, ek ürün satışı yapmak için düzenlemeler yapılabilir.

Birliktelik kuralları büyük veriler arasından birliktelik örüntülerini bularak pazarlama, karar verme ve iş yönetimine oldukça büyük fayda sağlamaktadır. Birliktelik kuralları kullanışlı ve anlaşılması kolay olduğundan finans, haberleşme, pazarlama, perakendecilik ve elektronik ticaret gibi alanlarda yayılmıştır (Yurtsever, U., 2002).

3.2. Birliktelik Kuralları Matematiksel Gösterimi ve Temel Kavramlar

3.2.1. Birliktelik kuralları matematiksel modeli

Birliktelik kuralı öncül (antecedent) ve izleyen (consequent) olarak adlandırılan veri kümelerinden oluşur (Kılıç, Y., 2009).. İzleyen çoğunlukla bir ürün olarak oluşur.

Kural öncülden izleyene yönelik bir ok şeklinde gösterilir. Birliktelik kuralı öncül veri kümesi ile izleyen veri kümesi arasındaki ilişkiyi gösterir.

$$\text{Kurallar: } \{X_1, X_2, \dots, X_m\} \Rightarrow \{Y_1, Y_2, \dots, Y_n\}$$

Bu kural “ X_1, X_2, \dots, X_m nesnelere bulunduğu işlemlerde sıklıkla Y_1, Y_2, \dots, Y_n 'de aynı işlemlerde yer alır” anlamına gelmektedir

Birliktelik kuralının matematiksel modeli aşağıdaki şekilde gösterilebilir:

$I = \{I_1, I_2, \dots, I_m\}$, bir dizi (veri seti)

$T = \{t_1, t_2, \dots, t_n\}$ ise veri tabanındaki işlemler olsun.

Her bir t_k 'nın sahip olacağı değer 0 veya 1'dir.

Eğer $t_k = 0$ ise I_k satın alınmayan,

Eğer $t_k = 1$ ise I_k satın alınan demektir.

$X \in I$ için X 'deki her bir I_k 'ya uyan t_k değeri, $t_k = 1$ dir.

D , T hareketinin tek bir tanımlayıcısının olduğu bir veri tabanını belirtmektedir ve bir veriler kümesini kapsamaktadır, öyle ki $T \in I$ olmaktadır. X bir veri seti olmak üzere T hareketi X veri setini ancak ve ancak $X \in T$ koşulunu eğer sağlarsa içerir. İşlem numaralarıyla ulaşılan veriler arasındaki bağımlılık ilişkisinin yüzde yüz doğru olması düşünülemez. Aynı şekilde çıkarılan kuralın, mevcut işlem setinin büyük bir kısmı tarafından desteklenmesi beklenmektedir (Kantardzic, M., 2011).

Bu sebeple, $X \Rightarrow Y$ kuralı kullanıcı vasıtasıyla seviyesi minimum destek değerine (s : destek) ve güvenilirlik (c : güven) eşik değerine sahiptir. $X \Rightarrow Y$ eşleştirme kuralına destek, güvenilirlik ölçütü iliştilir ve biçimsel olarak $\Phi(D) = \langle X \Rightarrow Y, c, s \rangle$ ile gösterilir (Ateş, Y. & Karabatak, M., 2017).

3.2.2. Destek ölçütü

Birlikte kuralında ilk destek değeri tanımlanır (3.1). Destek değeri, $X \Rightarrow Y$ kuralında hem X hem Y 'nin birlikte sağlanması durumudur (Özçalıcı, M., 2017).

$$\text{destek}(X \Rightarrow Y) = P(X \cup Y)$$

$$\text{destek}(X \Rightarrow Y) = \frac{\text{X ve Y'nin bulunduğu satır sayısı}}{\text{toplam satır sayısı}} \quad (3.1)$$

3.2.3. Güven ölçütü

Güven ölçütü girildikten sonra artık öğeler arasındaki birliktelik hesaplanabilir (3.2). Güven ölçütü, $X \Rightarrow Y$ kuralında X 'in sağlanması durumunda Y 'nin sağlanması olasılığıdır (3.3) (Özçalıcı, M., 2017).

$$\text{güven}(X \Rightarrow Y) = \frac{\text{destek}(X \Rightarrow Y)}{\text{destek}(X)} \quad (3.2)$$

$$\text{güven } (X \Rightarrow Y) = \frac{\text{X ve Y'nin bulunduğu satır sayısı}}{\text{X'in bulunduğu satır sayısı}} \quad (3.3)$$

3.2.4. Kaldıraç ölçütü

Kaldıraç değeri, X ile Y'nin bağımsız olup olmadığını gösteren oransal değerdir (3.4). Eğer iki değer birbirinden bağımsız ise bunlarla ilişkili kural çıkarılamaz. Eğer kaldıraç değeri 1'den büyükse değişkenler birbirine bağımlıdır ve bunlarla ilişkili kural oluşturulabilir (Karaibrahimoğlu, K., 2014).

$$\text{kaldıraç}(X \Rightarrow Y) = \frac{\text{destek}(X \Rightarrow Y)}{\text{destek}(X) * \text{destek}(Y)} \quad (3.4)$$

Birliktelik kuralı madenciliğine başlamadan önce destek ve güven değerleri belirlenir. Destek değeri, girilen destek değerinden küçük olan öğeler nesne kümeden budanır. Bu değerler 0-1 aralığında girilir ve %0 - 100 şeklinde ifade edilir.

Çizelge 3.1. Örnek ürün satış tablosu.

TID	Ürünler
1	Ekmek, Su, Meyve Suyu
2	Kahve, Su, Ekmek
3	Su, Çay
4	Ekmek, Çay
5	Çay, Meyve Suyu

Çizelge 3.1'deki veriler ışığında {ekmek} ile {su} arasındaki ilişki şu şekilde açıklanabilir.

$$\text{destek}(\text{ekmek} \Rightarrow \text{su}) = \frac{\text{ekmek ve su'nun bulunduğu satır sayısı}}{\text{toplam satır sayısı}} = \frac{2}{5} = 0.4$$

$$\text{güven}(\text{ekmek} \Rightarrow \text{su}) = \frac{\text{ekmek ve su'nun bulunduğu satır sayısı}}{\text{ekmek'in bulunduğu satır sayısı}} = \frac{2}{3} = 0,66$$

$$\text{kaldıraç}(\text{ekmek} \Rightarrow \text{su}) = \frac{\text{destek}(\text{ekmek} \Rightarrow \text{su})}{\text{destek}(\text{ekmek}) * \text{destek}(\text{su})} = \frac{0.4}{0.6 * 0.6} = 1.11$$

Çizelge 3.1'deki örnek üzerinden çıkan bir kuralı açıklayacak olursak; ekmek \Rightarrow su (Destek=%40, Güven=%66, Kaldıraç=1.11)

- Destek: Müşterilerin %40'ı ekmek ve suyu birlikte almıştır.
- Güven: Ekmek alan müşterilerin %66'sı su da almıştır.
- Kaldıraç: Kaldıraç 1.11 çıkmıştır. Kaldıraç 1'den büyük bir değer çıktığı için ekmek ile su arasındaki ilişkiye kural denilebilir.

3.3. Birliktelik Kurallarının Kullanım Alanları

Birliktelik kuralları hemen hemen her alanda kullanılmaktadır. Bir ürün satın alırken ürünler arasındaki ilişkileri çıkararak farklı ürün ya da ürünlerin satın alınmasını sağlar, bu ürünler arasındaki ilişkiyi ortaya koyar. Ürünler arasındaki ilişkiyi temel alarak belli kurallar ortaya koyar. Market sepet analizi müşterilerin sepetlerini inceleyerek aldığı ürünlerin arasında bir ilişki kurar. Ürünlerin aynı zamanda satılma ihtimalinin tespiti mağaza sahipleri tarafından önem arz etmektedir. Ürünler arasındaki bu ilişki sayesinde ürünlerin raf yerleşimini, mağazanın tasarımını, en çok satılan ürünlerin tespitini yaparak bu ürünleri depoda tedariklerini ve depolanmasını sağlar.

Birliktelik kuralları, mühendislik, tıp, ticaret, finans, market satış analizlerinde kullanılır (Timor, M. & Şimşek, U., 2008).

3.4. Birliktelik Kuralların Aşamaları

Birliktelik kural çıkarımı iki aşamada gerçekleşir (Han, J. & Kamber, M. Pei, J., 2012).

Sık kullanılan veri kümelerinin tespiti:

Verilerin sık geçen kümede yer alabilmesi için, belirlenen destek değerinden daha büyük bir destek değerine sahip olması gerekir. Girilen destek değerinden küçük destek değerine sahip veriler sık geçen kümesine dâhil edilmez, hesaplama dışına itilir.

Sık kullanılan veri kümelerinden kural çıkarma:

Sık kullanılan veri kümesinde kalan tüm öğeler minimum destek ve güven değerinden büyük destek ve güven değerine sahip öğelerdir. Destek değeri ve güven ölçütleri sıfıra yaklaştıkça sık kullanılan veri kümelerindeki öğe sayısı artacak ve kural sayısı bu doğrultuda fazla olacaktır. Destek ve güven değeri bire yaklaştıkça sık

kullanılan öge kümesindeki veri sayısı azalacak kural sayısı da bu doğrultuda azalacaktır.

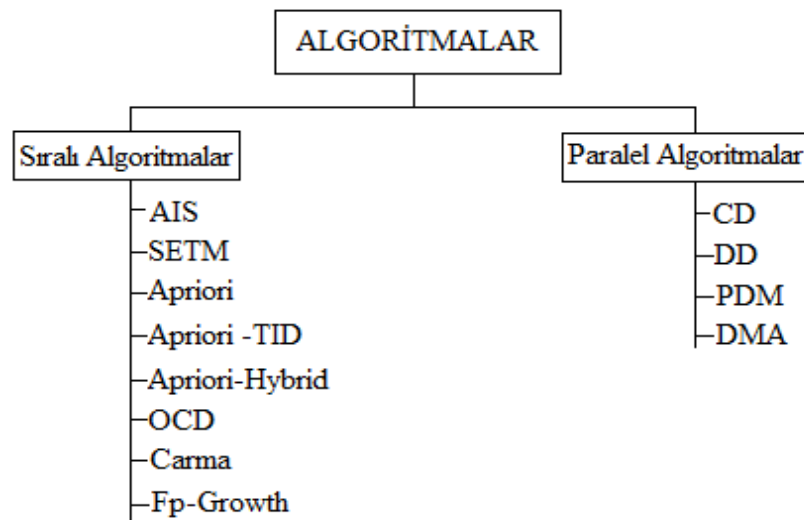
3.5. Birliktelik Kural Algoritmaları

Birliktelik kuralları oluşturmada kullanılan algoritmalar veri kümesini minimize etme ve veri kümesini en hızlı zamanda tarayarak verilerin destek değerini hesaplayıp sık kullanılan kümesini oluşturma temeline dayanır (Gürsoy, U. T. Ş., 2012).

Birliktelik kuralı algoritmalar amacı aynı olsa da farklılaştıkları noktalar şu şekilde listelenebilir.

- Aday oluşturma süreci
- Aday kümenin destek değerlerinin hesaplanması
- Veri tabanını toplam tarama sayısı
- Kullanılan veri yapıları

Veri tabanlarından gerekli kuralları çıkarmak için kullanılan birliktelik kuralı algoritmaları sıralı ve paralel olmak üzere Şekil 3.1'de gösterilmiştir (Söylemez, İ. & Doğan, A. & Özcan, U., 2016).



Şekil 3.1. Birliktelik kural çıkarım algoritmaları.

3.5.1. AIS algoritması

AIS algoritması, veri tabanındaki tüm yaygın ürün kümelerini oluşturmak için 1993 yılında Agrawal, Imielinski ve Swami tarafından geliştirilmiş ve yayınlanmış

olan ilk algoritmadır. Algoritma karar destek sorgulamaları yapmak için veri tabanlarının fonksiyonlarını artırmaya odaklanmıştır. Veri tabanında bulunan ürün isimlerinin A'dan Z'ye sıralanması kısıtını taşır (Delibaş, E., 2010).

Veri tabanları, AIS veri kümesini elde edebilmek için birçok kez taranır. İlk taramada veri tabanındaki ürünleri tek tek sayarak kullanılma sayılarını bulur ve sık kullanılan ürünleri belirler (Şekil 3.2). Sık kullanılan ürünleri aday ürün kümesine atar. İkinci aşamada, sık kullanılan ürün kümelerindeki ürünlerin ilişkilerini tespit eder. Tespit edilen bu ortak ürünler var olan yeni ürünlerle birleştirilerek aday kümeler güncelleştirilir.

AIS algoritması aday kümeleri belirlerken yok etme tekniği kullanır (Kumbhare, T. A., & Chobe, S. V., 2014). Aday kümeler belirlenirken gereksiz ürün kümeleri belirlenerek silinir. Aday kümeler belirlenirken destek değerleri hesaplanır. Belirlenen destek değerden düşük değere sahip aday kümeler silinir.

```

L1 = {nesneküme};
for (k=2; Lk-1 = 0; k++)
{
    Ck = Boş küme;
    for (t=0; t<öğe sayısı; t++)
    {
        Lt = altküme (Lk-1)
        Ct = ilişkili ürünler;
        for (c=0; c < Ct; c++)
        {
            if (c  $\sum$  Ck)
                c++;
            else
                Ck ya Ct'yi ekle;
                c++;
        }
        if (ürünün destek değeri > girilen destek değeri)
            Lk=Ck;
    }
}
Return Uk Lk;

```

Şekil 3.2. AIS algoritması özet kodu (Zerman, M., 2018).

3.5.2. SETM Algoritması

SETM algoritması, Houtsmal tarafından 1995 yılında, sık geçen nesne kümelerinin hesaplanması için önerilmiştir. AIS algoritmasından farklı olarak Şekil 3.3'de özet kodu

verilen SETM algoritması, sık geçen nesne küme adayları oluşturmak için SQL kodlarını kullanmaktadır (Silahtaroglu, G., 2016).

SETM algoritması da AIS algoritmasında olduğu gibi bir çok kez tarama yapar (Kumbhare, T. A., & Chobe, S. V., 2014). Bu iki algoritmada gereksiz aday oluşturduğu için çok tercih edilen algoritmalar değildir.

3.5.3. CD algoritması

CD (Count Distribution / Sayım Dağılımı) algoritmasından D veri tabanı $\{D_1, D_2, \dots, D_p\}$ olarak parçalara ayrılmış ve n sayıda işlemciye paylaştırılmıştır.

Algoritmanın çalışması üç aşamada gerçekleşir.

1. aşamada, yerel veri tabanı bölümlerdeki aday veri kümelerinin destek sayıları bulunur.

2. aşamada, bölümlere ayrılan işlemci, veri kümelerinin destek değerini bulmak için, bölümler birbirlerine aday veri kümelerinin yerel destek sayılarını hesaplar.

Son aşamada ise büyük veri kümeleri tanımlanır ve her işlemcide birbirinden bağımsız olarak veri kümesine aday veri kümeleri oluşturulur. CD Algoritması bu aşamalarını aday veri kümelerini sonlandırana kadar sürdürür (Zerman, M., 2018).

3.5.4. DD algoritması

DD Algoritmasında (Data Distribution / Veri Dağılımı), veri kümeleri işlemcinin bölümlerine dairesel döngü şeklinde bölünmüş ve paylaştırılmıştır. Çalışma CD algoritması gibi üç aşamada gerçekleşir.

Birinci aşamada, işlemci bölümleri, kendilerine paylaşımı yapılmış veri kümelerinin yerel sayımlarını elde etmek için yerel veri tabanının bölümlerini tarar.

İkinci aşamada, işlemci bölümleri kendine ait veri tabanı bölümlerini bir sonraki işlemci bölümüne verir ve bir önceki işlemci bölümünden veri tabanı bölümlerini alır.

Son aşamada tüm veri tabanında genel destek değerini bulmak için önceden kabul edilen veri tabanı bölümlerini tarar. En sonunda, işlemci bölümleri veri kümesi bölümündeki büyük veri kümelerini hesaplar, bunları tüm büyük veri kümelerini elde etmek için diğer işlemcilerle paylaşır ve sonrasında aday veri kümeleri oluşturur, bunları bölümlere ayırır ve bu aday veri kümelerini diğer işlemcilere dağıtır (Zerman, M., 2018). DD algoritması bu aşamalarını aday veri kümelerini sonlandırana kadar sürdürür.

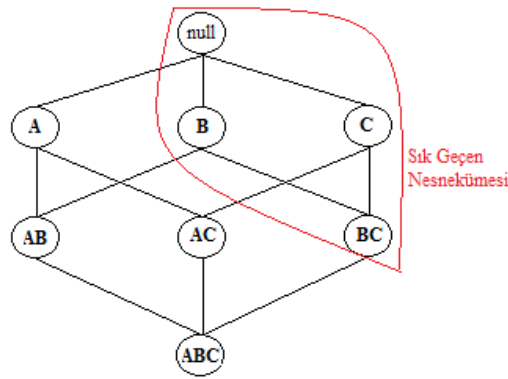
3.5.5. Apriori algoritması

Birliktelik kural analizinin temelini oluşturan, algoritmalar içerisinde en fazla bilinen ve kullanılan algoritmadır. Birliktelik kural analizi çalışmalarına 1993 yılında, Agrawal tarafından başlatılmış olup Agrawal ve Srikant 1994 yılında Apriori algoritmasının temelini atmışlardır.

Apriori algoritması sık kullanılan öğelerin bir önceki adımdan tahmin edilmesi üzerine tasarlanmıştır. Adı da önceki anlamındaki ‘prior’ kelimesinden gelmektedir (Döşlü, A., 2008).

Apriori algoritmasının temel özelliği; eğer bir nesne küme sık geçen ise bu kümenin bütün alt kümeleri de sık geçen nesne küme olmalıdır. Aynı mantıkla nesneküme sık kullanılmayan ise bu kümenin bütün alt kümeleri de sık kullanılmayan nesne küme olarak değerlendirilir.

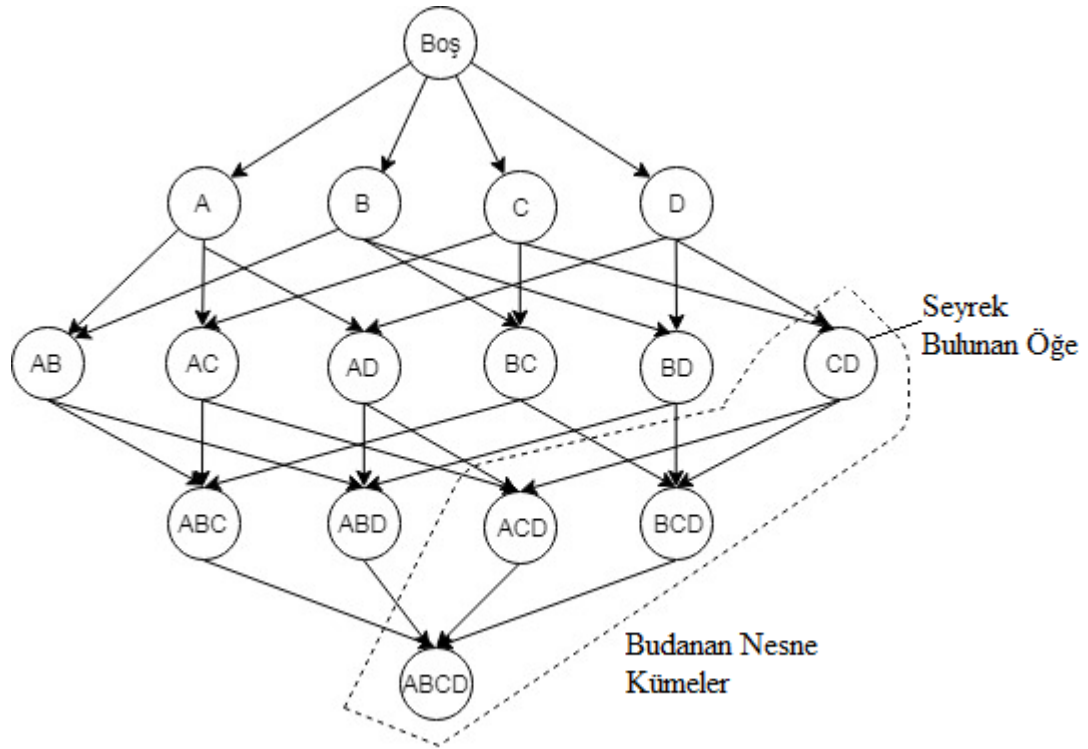
Şekil 3.3’de gösterildiği gibi kırmızı yuvarlak içerisinde alınmış alanın sık kullanılan nesne kümelerinin olduğu düşünülürse $\{BC\}$ kümesinin bir sık geçen nesneküme olduğu varsayılır. $\{BC\}$ kümelerini içeren her işlemin bu kümenin alt kümeleri olan $\{B\}$, $\{C\}$ kümelerini de içerir sonucuna varılır. $\{BC\}$ sık geçen nesne küme ise bütün alt kümeleri de sık geçen nesne küme olmalıdır.



Şekil 3.3. Apriori algoritması birleştirme özelliği.

Sık olmayan bir nesne kümenin bütün alt kümeleri de sık olmayan nesne kümeleridir. Şekil 3.4’de gösterilen $\{CD\}$ kümesinin sık olmayan küme olarak bulunduğu tespit edilmiş, $\{CD\}$ kümesini kapsayan $\{ACD\}$, $\{BCD\}$, $\{ABCD\}$ üst

kümeleri de sık olmayan küme kabul edileceği için bu kümeler budanabilir (Eker, M. E., 2016).



Şekil 3.4. Apriori algoritması budama özelliği.

Apriori algoritması başlangıçta veri tabanını tarar ve öğelerinin veri tabanında kaç kez geçtiklerine dair bilgiyi içeren destek değerlerini bulur. Başlangıçta girilen minimum destek değerinden daha küçük destek değerine sahip öğeler nesne kümelerine dâhil edilmez, bu öğeler ayıklanarak veri tabanı sadeleştirilir (Şekil 3.5).

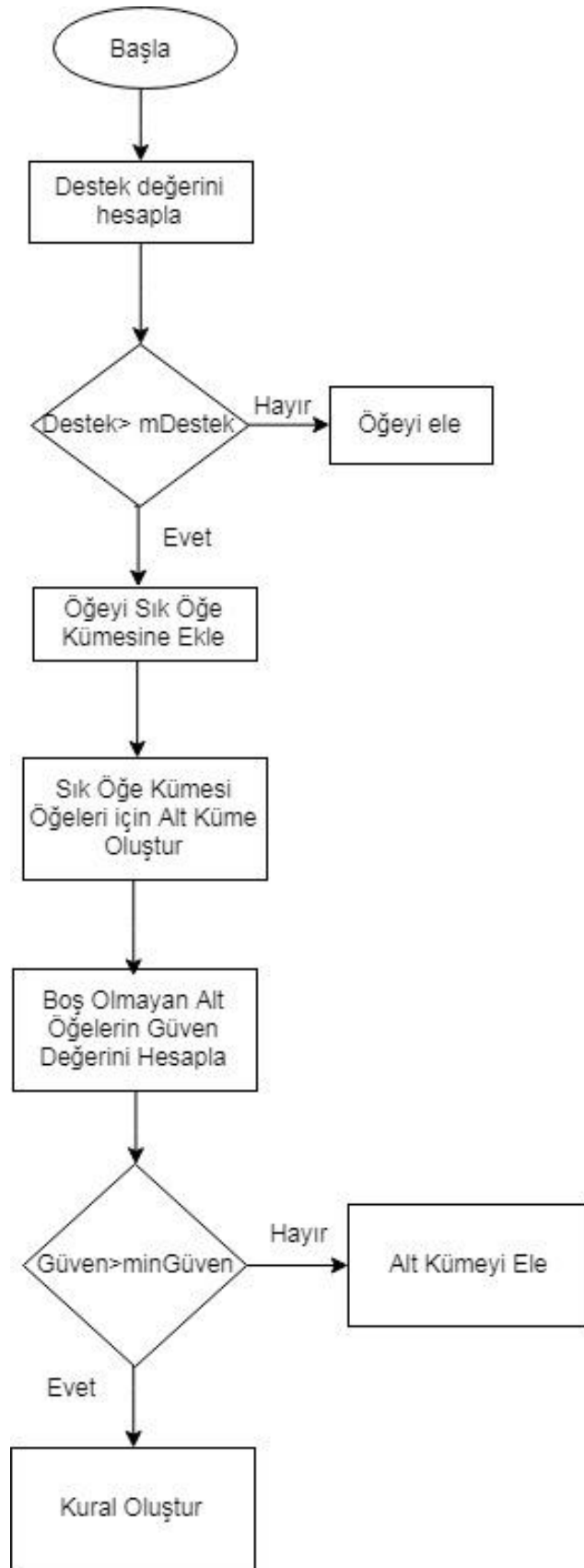
Nesne kümelerin tekli destek değerlerinden yararlanarak ikili nesne kümeler oluşturur, ikili nesne kümelerin destek değerini hesaplayarak düşük destek değere sahip öğeleri ayıklayarak sadeleştirir (Agrawal, R. & Srikant, R., 1994). Aynı işlemi ikili nesne kümelerden yararlanarak üçlü nesne kümeler oluşturarak devam eder. Bu tarama ve sadeleştirme işlemi, girilen minimum destek değerden küçük nesne kümesi kalmayana kadar devam eder (Şekil 3.6).

```

Apriori(T, MST) // T veri tabanı, MST minimum destek değeri
{
    Ck: k büyüklükte aday nesne küme;
    Lk: k büyüklükte sık kullanılan nesne küme;
    L1: sık kullanılan ürünler;
    for (k=1; Lk != Boş küme; k++)
    {
        Ck+1= Lk tarafından oluşturulan adaylar;
        for each(veri tabanında işlem olduğu sürece)
            Ck+1 inci ürünün destek değerini bir artır;
        Lk+1 = Ck+1 deki destek değeri büyük olanları;
    }
} end
Return Uk Lk;

```

Şekil 3.5. Apriori algoritması özet kodu.



Şekil 3.6. Apriori algoritması akış diyagramı.

Apriori algoritması ile sık kullanılan öğelerin tespiti, birleştirme ve budama işlemleri Çizelge 3.2’de verilen örnek veri tabanına göre özetlenmiş ve anlatılmıştır.

Çizelge 3.2. Örnek veri seti.

TID	Ürünler
1	Ekmek, Meyve Suyu, Soda
2	Su, Meyve Suyu, Çikolata
3	Ekmek, Su, Meyve Suyu, Çikolata
4	Su, Çikolata

İlk olarak kullanıcı tarafından destek değeri girilir. Bu örneklem için destek eşik değeri 2 olarak kabul edilmiştir.

Çizelge 3.2’de bir D veri tabanına ait işlemler verilmiştir. Bu veri tabanında 4 işlem bulunmaktadır, $|D|=4$ olarak belirlenir.

İlk olarak her bir nesne küme C1 aday kümesini oluşturmak için taranır ve elemanlarının destek değerleri hesaplanır (Çizelge 3.3).

Çizelge 3.3. C1 aday kümesi.

Öge Listesi	Destek Değeri
Ekmek	2
Su	3
Meyve Suyu	3
Soda	1
Çikolata	3

Çizelge 3.3’deki değerlerine bakıldığında 4 ürünün destek değeri minimum destek değerinin üzerinde olduğu için sık kullanılan nesne kümeler olarak kabul edilir. Soda sık geçen nesne kümesinden çıkarılır. Bir öğeli nesne kümeler ile sık geçen L1 kümesi oluşturulur (Çizelge 3.4).

Çizelge 3.4. L1 aday kümesi.

Öge Listesi	Destek Değeri
Ekmek	2
Su	3
Meyve Suyu	3
Çikolata	3

İki öğeli sık geçen nesnekümler (C2) oluşturulur (Çizelge 3.5). C2 aday kümesi Çizelge 3.4'deki öğelerin ikili kombinasyonu ile oluşur.

Çizelge 3.5. C2 aday kümesi.

Öge Listesi	Destek Değeri
Ekmek, Su	1
Ekmek, Meyve Suyu	2
Ekmek, Çikolata	1
Su, Meyve Suyu	2
Su, Çikolata	3
Meyve Suyu, Çikolata	2

D veri tabanı bir kez daha taranır ve C2 kümesindeki öğelerin destek değerleri hesaplanır. C2 kümesindeki minimum destek değerine sahip olan 2 öğeli nesne kümeler Çizelge 3.6'da gösterilen L2 aday kümesini oluşturur.

Çizelge 3.6. L2 aday kümesi.

Öge Listesi	Destek Değeri
Ekmek, Meyve Suyu	2
Su, Meyve Suyu	2
Su, Çikolata	3
Meyve Suyu, Çikolata	2

Üç ögeli sık geçen nesne kümeler Çizelge 3.7’de verildiği gibi kalan öğelerin üçlü kombinasyonu ile oluşturulur.

Çizelge 3.7. C3 aday kümesi.

Öge Listesi	Destek Değeri
Su, Meyve Suyu, Çikolata	2

Destek değeri 2’nin altında ürün olmadığı için L3 ve C3 aynı çıkacaktır. 3. adıma geçerken Apriori algoritmasının budama özelliği kullanılmamıştır.

Son olarak, oluşturulan 3 elemanlı {Su, Meyve Suyu, Çikolata} sık geçen nesne kümesinden %40 minimum destek kriterini sağlayan kural oluşturulmuştur.

3.5.6. FP-Growth algoritması

FP-Growth (Frequent-Pattern Growth) algoritması, sık rastlanan öğeleri tespit etmek için kullanılan, büyük veri kümelerinde verimli, hızlı kurallar oluşturabilen birliktelik kural algoritmasıdır. Birliktelik kural algoritmaları içerisindeki veri kümelerinde ilişkisi olan öğeleri tespit etme konusunda hızlı sonuçlar vererek, maliyeti azaltmıştır. Daha hızlı sonuçlar verebilmesinde en önemli etken, veri tabanını sık desen ağacı içinde tutuyor olmasıdır.

En çok kullanılan iki algoritmayı karşılaştıracak olursak; Apriori algoritmasına göre FP-Growth çok daha hızlı sonuçlar vermektedir. Çünkü Apriori algoritması veri tabanını defalarca kez tararken, FP-Growth algoritması veri tabanını büyüklüğü ne olursa olsun iki kez tarar (Özdoğan, G. Ö., 2010).

İlk taramada veri tabanındaki tüm öğelerinin destek değeri hesaplanır. İkinci taramada ise ağaç veri yapısını oluşturur. Yeni nesne kümeler oluşturmadığı ve veri tabanını defalarca kez taramayıp sadece iki kez taradığından dolayı da çok hızlı sonuçlar vermesi yönüyle büyük veri tabanlarında kural ilişkisini tespit etmek hususunda büyük bir kazanımdır. Apriori algoritmasının en büyük dezavantajı aday oluşturmasıdır ve bu adaylar için veri tabanının yüzlerce kez taranması gerekmektedir. Veri büyüdükçe aday artmakta ve kural oluşturma süreci uzamaktadır. FP-Growth algoritması zaman olarak kayıp yaşanan aday oluşturma sürecine girmez (<http://deryagunduz.com/?tag=fp-growth-algoritmasi/>).

FP-Growth algoritması böl ve yönet stratejisi kullanarak, problemi alt problemlere bölerek sonucu bulur (Erpolat, S., 2012). Böl ve yönet mantığı arama motorlarında da kullanılır.

FP-Growth algoritmasının çalışma yapısı şu şekildedir (Dunham, M. H., & Xiao, Y., & Gruenwald, L., Hossain, Z., 2008).

- Şekil 3.7’de gösterildiği gibi tüm veri tabanı taranır ve tüm öğelerin destek değerleri hesaplanır. Kullanıcı tarafından girilen destek değerine eşit ve destek değerinden büyük olan öğeler tespit edilir.

- Sık kullanılan öğelerin destek değerleri büyükten küçüğe olacak şekilde sıralanır.

- Ağaç yapısı oluşturulur.

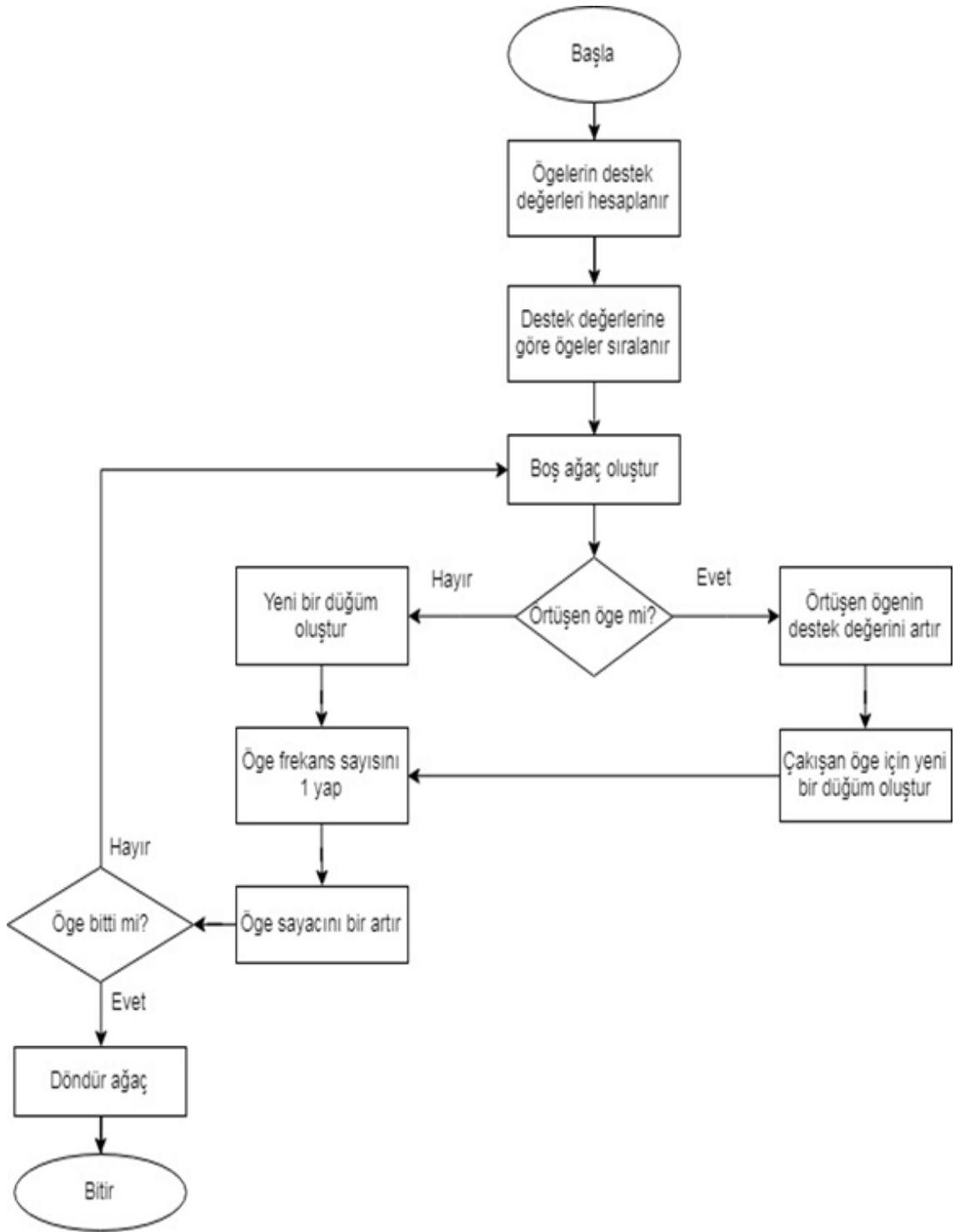
- Veri tabanı ikinci ve son kez taranır, her öğe için; sadece sık kullanılan öğeler ağaca eklenir. Tüm öğeler işlenene kadar bu işlem tekrarlanır (Şekil 3.8).

```

FPGrowth(VT,MST)
Agac Tanımla ve temizle : F[ ];
for each VT içindeki öğeler do
    for each öge_desteksayısı do
        öge_desteksayısı++;
    end
end
for each VT içindeki öğeler do
    Destek değerlerine göre sırala;
    Düğüm oluştur;
    Dügüme ekle;
End
Return ağaç

```

Şekil 3.7. FP-Growth algoritması özet kodu.



Şekil 3.8. FP-Growth algoritması akış diyagramı.

FP-Growth algoritması, Çizelge 3.8'deki örnek veri tabanı üzerinden anlatılmıştır.

Çizelge 3.8. Örnek veri tabanı.

TID	Ürünler
1	[1 2 5]
2	[2 4]
3	[2 3]
4	[1 2 4]
5	[1 3]
6	[2 3]
7	[1 3]
8	[1 2 3 5]
9	[1 2 3]

Ürünler işlem numaraları ile sıralanmıştır. Her bir numara farklı bir ürünü ifade etmektedir. Bu uygulama için destek değeri %22.22, güven değeri %60 olarak belirlenmiştir.

İlk olarak tüm öğelerinin destek değerleri hesaplanır (Çizelge 3.9).

Çizelge 3.9. Örnek uygulama ürün destek değerleri.

Ürünler	Destek Değeri
1	6
2	7
3	6
4	2
5	2

Ürünlerin destek değeri hesaplanmış ve girilen destek değerinden daha küçük bir destek değere sahip öğe tespit edilememiştir. Veri kümesinden hiçbir ürün çıkarılmamıştır (Çizelge 3.9).

Sık kullanılan öğeler büyükten küçüğe sıralanır (Çizelge 3.10).

Çizelge 3.10. Örnek uygulama sıralı liste.

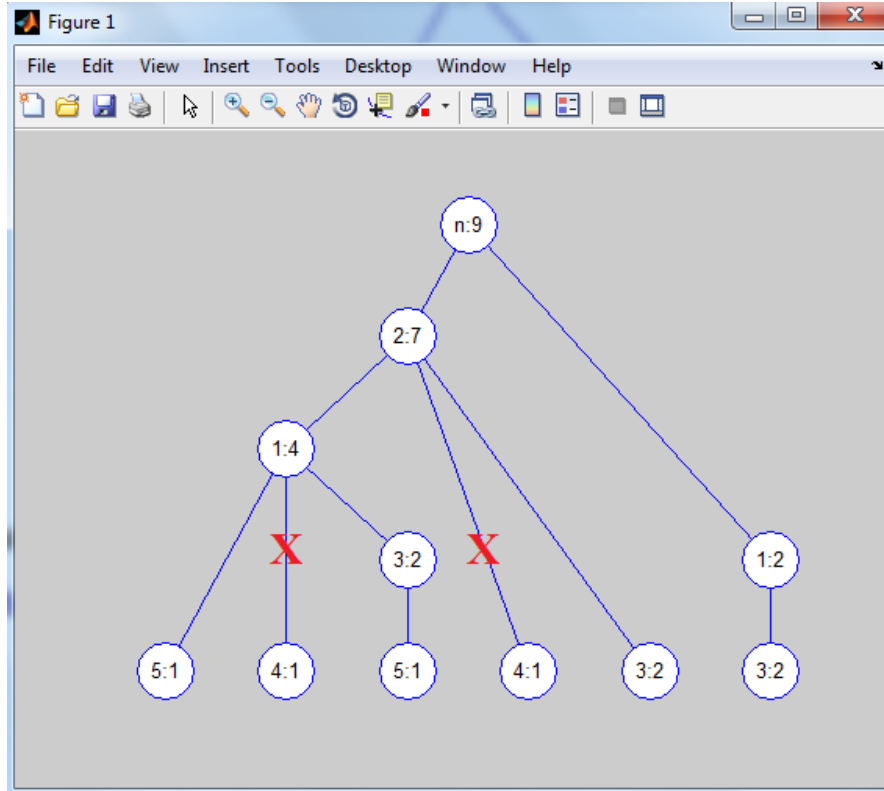
Ürünler	Destek Değeri
2	7
1	6
3	6
4	2
5	2

Çizelge 3.11. Sıralanmış liste.

TID	Ürünler
1	[2 1 5]
2	[2 4]
3	[2 3]
4	[2 1 4]
5	[1 3]
6	[2 3]
7	[1 3]
8	[2 1 3 5]
9	[2 1 3]

Birliktelik kuralları belirlerken ağacın her bir düğümünde ürünlerin destek değerleri tespit edilerek girilen destek değerinden büyük olup olmadığı kontrol edilir. Düğümdeki öğelerde destek değeri düşük olanlar kural oluştururken önemsenmez. Bu düğümler budanır yani öğeler nesne kümesinden çıkarılır. Matlab programı ile çizim fonksiyonu kullanılarak ağaç yapısının son hali oluşturulmuştur (Şekil 3.9).

Boş kümeden başlayarak işlem sırası ile ağaç yapısı adım adım oluşturulmuş ve **Ek-1**'de gösterilmiştir.



Şekil 3.9. Fp-Growth budama özelliği.

Şekil 3.9’ da görüldüğü üzere minimum destek değeri $2/9$ girildiği için destek değeri 1 olan öğeler ve alt öğeleri budanmıştır. Örnek uygulama Matlab programında çalıştırılmış ve Çizelge 3.12’ deki kural listesi oluşmuştur.

Çizelge 3.12. Örnek uygulama kural listesi.

Kural Tablosu	Ürün 1	Ürün 2	Destek Değeri	Güven Değeri	Kaldıraç Değeri
Kural 1	5	2	0,22	1	1,28
Kural 2	5	1	0,22	1	1,50
Kural 3	5	[2 1]	0,22	1	2,25
Kural 4	[2 5]	1	0,22	1	1,50
Kural 5	[1 5]	2	0,22	1	1,28
Kural 6	4	2	0,22	1	1,28
Kural 7	1	3	0,44	0,666	1
Kural 8	3	1	0,44	0,666	1

Kural 1' e göre müşterilerinin %22'si 5 ve 2 numaralı ürünleri birlikte almıştır, 5. ürünü alanların hepsi 2. ürünü de almıştır.

Kural 2' e göre müşterilerinin %22'si 5 ve 1 numaralı ürünleri birlikte almıştır, 5. ürünü alanların hepsi 1. ürünü de almıştır.

Kural 3' e göre müşterilerinin %22'si 5, 2, 1 numaralı ürünleri birlikte almıştır, 5. ürünü alanların hepsi 2. ve 1. ürünü de almıştır.

Kural 4' e göre müşterilerinin %22'si 2, 5, 1 numaralı ürünleri birlikte almıştır, 2. ve 5. ürünü alanların hepsi 1. ürünü de almıştır.

Kural 5' e göre müşterilerinin %22'si 1, 5 2 numaralı ürünleri birlikte almıştır, 1. ve 5. ürünü alanların hepsi 2. ürünü de almıştır.

Kural 6' e göre müşterilerinin %22'si 4 ve 2 numaralı ürünleri birlikte almıştır, 4. ürünü alanların hepsi 2. ürünü de almıştır.

Kural 7' e göre müşterilerinin %44'ü 1 ve 3 numaralı ürünleri birlikte almıştır, 1. ürünü alanların %66'sı 3. ürünü de almıştır.

Kural 8' e göre müşterilerinin %44'ü 3 ve 1 numaralı ürünleri birlikte almıştır, 3. ürünü alanların %66'sı 1. ürünü de almıştır.

FP-Growth algoritması küçük veri tabanı için kural çıkarma süresi 947 salise olarak tespit edilmiştir. Şekil 3.10'de FP-Growth algoritmasının kural oluşturma süresinin fonksiyonlara dağılımı gösterilmiştir.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
main	1	0.947 s	0.229 s	
FPGrowth	1	0.314 s	0.157 s	
PlotTree	1	0.297 s	0.062 s	
hold	11	0.110 s	0.107 s	
DisplayRules	1	0.093 s	0.021 s	
GetPowerSet	10	0.055 s	0.015 s	
dec2binvec	54	0.051 s	0.010 s	
treelayout	1	0.049 s	0.023 s	
num2str	54	0.045 s	0.023 s	
mat2str	16	0.044 s	0.040 s	
union	9	0.034 s	0.004 s	
ismember	54	0.031 s	0.006 s	
union>unionR2012a	9	0.031 s	0.010 s	
newplot	21	0.030 s	0.012 s	
str2num	54	0.027 s	0.017 s	
ismember>ismemberR2012a	54	0.025 s	0.013 s	
unique	9	0.021 s	0.011 s	
IsSame	933	0.019 s	0.019 s	
GetNonTrivialSubsets	8	0.018 s	0.007 s	
treelayout>fixparent	1	0.018 s	0.018 s	
axis	1	0.017 s	0.017 s	
close	1	0.015 s	0.010 s	
num2str>handleNumericPrecision	16	0.013 s	0.003 s	
ishold	21	0.012 s	0.012 s	
ismember>ismemberBuiltinTypes	54	0.012 s	0.012 s	
newplot>ObserveAxesNextPlot	21	0.011 s	0.011 s	
str2num>protected_conversion	54	0.010 s	0.010 s	
num2str>convertUsingRecycledSprintf	16	0.010 s	0.010 s	
unique>uniqueR2012a	9	0.010 s	0.010 s	
int2str	37	0.008 s	0.008 s	
spparms	1	0.008 s	0.008 s	
dec2bin	54	0.008 s	0.008 s	
newplot>ObserveFigureNextPlot	21	0.006 s	0.006 s	
mat2str>isenumeration	16	0.004 s	0.004 s	
axescheck	11	0.003 s	0.003 s	
blanks	54	0.002 s	0.002 s	
fliplr	54	0.002 s	0.002 s	
close>safegetchildren	1	0.002 s	0.002 s	
close>getEmptyHandleList	1	0.001 s	0.001 s	
close>request_close	1	0.001 s	0.001 s	
axis>allAxes	1	0.001 s	0.001 s	
close>checkfigs	1	0.001 s	0.001 s	

Şekil 3.10. FP-Growth algoritması süre analizi.

4. MARKET SEPET ANALİZİ UYGULAMASI

4.1. Problemin Tespiti

Bu tez çalışmasında marketlerdeki ürünlerin raflara gelmeden önceki aşaması olan depolarda ürünlerin uygun yerlere konulması, ürünlerin yeterli miktarlarda depoda saklanmasıyla önemine değinilmiştir. Markette çok tüketilen ürünleri depoda saklayabilmek, depodan ürün istendiği zaman kolayca çıkarabilmek, iş gücü, zaman, ürün temini konusunda oldukça önemlidir. Ürünler arasındaki birliktelik ilişkisini, ürünlerin birlikte satılma zamanlarını tespit etmiş olmak ilişkisi olan ürünleri mevsime göre depoda bulundurulma miktarlarını, depoda uygun yere koyulmaları konusunda kolaylık sağlar. Bu birlikteliğin tespiti ayrıca iş gücünde kolaylık, hızlı çözüm ve ekonomiklik sağlar.

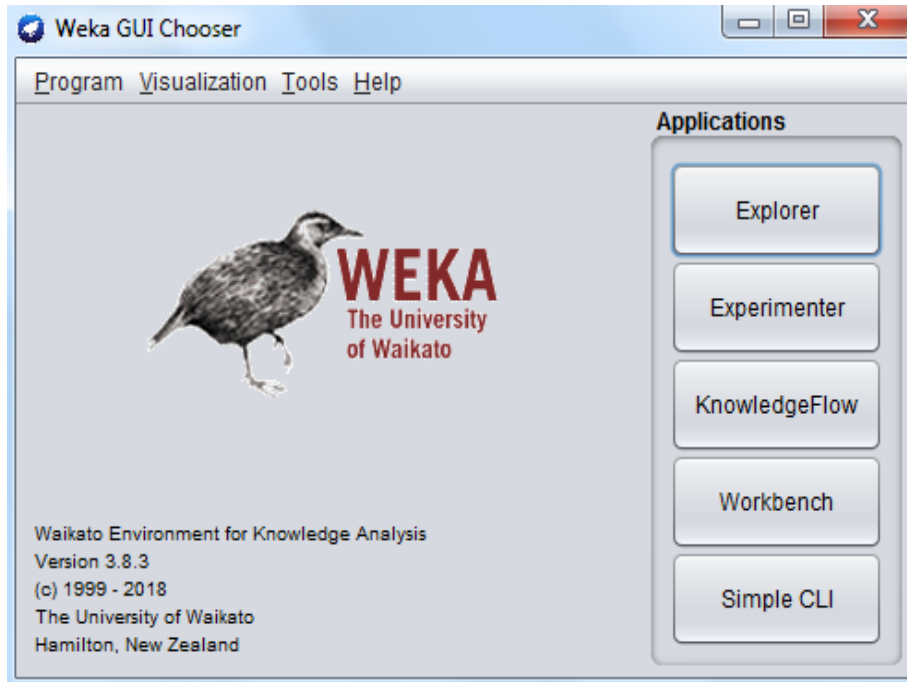
Bu tez çalışması kapsamında bir marketin deposundan birlikte çıkan ürünlerin tespit edilerek, bu ilişkiler ışığında depo içerisindeki ürünlerin en uygun yerleşme düzeyine getirilmesi üzerinde çalışılmıştır. Örneklem olarak WEKA programının market verisi kullanılmıştır.

4.2 Uygulamada Kullanılacak Teknolojiler

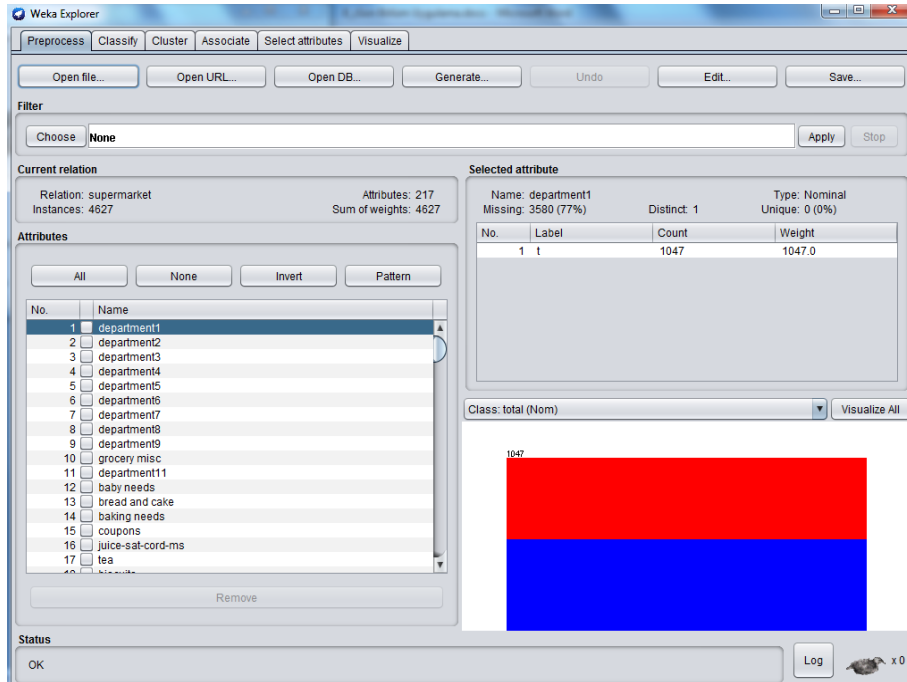
Çalışmada kullanılan bilgisayarın özellikleri; işlemci modeli Intel Core i5-3230M, işlemci hızı 2.60 GHz, sistem belleği 6 GB, sabit disk boyutu 750 GB olup işletim sistemi Window 7 Ultimate'dir. Uygulamada Microsoft Excel, Matlab ve WEKA programları kullanılmıştır.

4.2.1.WEKA programı

WEKA veri ön işleme araçlarının ve makine öğrenme algoritmalarının bir araya getirdiği açık kaynak kodlu bir yazılımdır. Yeni Zelanda'da Waikato Üniversitesinde Java yazılım dili ile oluşturulmuş ve geliştirilmiştir (Şekil 4.1). Dağıtık ve büyük veri tabanlarında kullanılabilir. Şekil 4.2'deki menülerden de anlaşılacağı üzere WEKA ile veri hazırlama, sınıflama, birliktelik analiz, kümeleme algoritmaları kullanılarak kurallar oluşturulabilir. Arff, CSV, C4.5 uzantılı dosyaları çalıştırabilmektedir (David, S. & Reutemann, P., 2006).



Şekil 4.1. WEKA ara yüzü.



Şekil 4.2. WEKA süpermarket datası.

4.3. Veri Madenciliği Süreçleri

Uygulamada kullanılacak olan veri tabanı bilgi keşfi aşamaları olan veri seçme, veri ön işleme, veri indirgeme yöntemleri ile oluşturulmuştur.

4.3.1. Veri setinin oluşturulması

Veri tabanı olarak WEKA programındaki market verisi kullanılmıştır. WEKA programındaki market verisi 4625 örneklem, 217 nitelikten oluşmaktadır. Arff uzantılı veri seti CSV dosya uzantısına ardından “mat” uzantılı dosyaya dönüştürülmüştür.

4.3.2. Veri setinin temizlenmesi

Veri temizleme sürecinde veriler incelenmiş, inceleme sonucunda veriler indirgenmiştir.

Veri setindeki tekrarlanan veriler, öğeleri listeden çıkarılmıştır. Çalışmada market rafları değil depo içerisinde düzeninin sağlanması amaçlandığı için market verisi üzerindeki öğelerde güncelleme yapılmıştır.

Birliktelik analizi yapılamayacağı için satıcısı olmayan ve tek satılan ürünlerde veri setinden çıkarılmıştır.

Yukarıdaki işlemler sonucunda 111 öğe veri setinden çıkarılarak veri seti daha güvenilir hale getirilmiştir. Veri seti 4625 örneklem ve 106 farklı öğeden oluşmuştur (Ek-2).

4.3.3. Veri seti üzerinde algoritmaların uygulanması

İndirgenen veriler Matlab programı kullanılarak birliktelik kural analizi algoritmalarından Apriori ve FP-Growth algoritması üzerinde çalıştırılmıştır. Bu algoritmalar arasındaki benzerlikler ve farklılıklar tespit edilmiştir. Tezde veri tabanı birliktelik kural analizi algoritmalarından Apriori ve FP- Growth algoritmaları ile çalıştırılmış ve belli kurallar tespit edilmiştir.

Algoritmaların çalışmaları doğrultusunda veri seti üzerinde incelemeler yapılmış ve uygun destek ve güven değerleri tespit edilmiştir. Destek ve güven değerleri [0-1] aralığındadır. Destek değeri sıfıra yaklaştıkça çok fazla kural tespit edildiği ve çalışma süresinin çok uzadığı tespit edilmiş, tersi durumda bire yaklaştıkça da kural sayısının azaldığı ve daha hızlı kurallar oluşturduğu görülmüştür. Güven değeri değiştirildiğinde kural sayısının etkisinin az olduğunu fakat çalışma süresinin sıfıra yaklaştıkça arttığı,

bire yaklařtıka azaldığı tespit edilmiştir. Bu tespitler sonucunda destek değeri 0.5, güven değeri 0.75 olarak belirlenmiştir.

4.3.3.1. Veri setinin Apriori algoritması sonuçları

Çizelge 4.1’de gösterilen dört kuralın destek değerlerinin 0.5’den, kaldıracağı değerlerinin ise 1’den büyük olduğu tespit edilmiştir.

Çizelge 4.1. Apriori algoritması sonucu bulunan kurallar.

Ürün No 1	Ürün No 2	Destek Değeri	Güven Değeri	Kaldıraç
50	3	0.505	0.795	1.105
68	3	0.502	0.784	1.090
3	50	0.505	0.701	1.105
3	68	0.502	0.698	1.090

Çizelge 4.2. Uygulama veri tabanı kesiti.

Ürün No	Ürün
3	Su(500ml)
50	Asitli İçecekler
68	Meyve Suyu

Kural çizelgesi Çizelge 4.2’de gösterilen su, asitli içecekler ve meyve suyu ürünleri arasındaki ilişkilerden kurulmaktadır.

Asitli içecekler ile su ve meyve suyu ile su arasında karşılıklı ilişki tespit edilmektedir.

Bu kuralların kesin kural olabilmeleri için girilen güven değerinden de büyük güven değerine sahip olmaları gerekmektedir. Çizelge 4.1’de görülmektedir ki depodan asitli içecek çıkarılırken %79’unda da su çıkarılmış, su çıkartılırken %70’inde asitli içecek depodan çıkarılmaktadır. Yine aynı durum meyve suyu çıkarılırken %78’inde su çıkartılmış, su çıkartılırken %69’unda meyve suyu depodan çıkartıldığı Apriori algoritması sonucunda görülmüştür. Bu dört ilişkiden iki tanesi %75 güven endeksinden düşük güven endeksine sahip olduğu için son kural olarak belirlenememiştir. Şekil 4.3’de Apriori algoritması ile son tespit edilen kurallar verilmiştir.

Apriori Kurallar:

Kural #1: 50 --> 3
Support = 0.50508
Confidence = 0.79537
Lift = 1.105

Kural #2: 68 --> 3
Support = 0.50249
Confidence = 0.78487
Lift = 1.0904

Şekil 4.3. Apriori algoritması uygulama kural sonuçları.

Algoritma kural oluşturma süresi yaklaşık 74 saniye sürmüş olup Şekil 4.4'de fonksiyonların detaylı çalışma süreleri gösterilmiştir.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
mainApriori	1	74.184 s	7.369 s	
Apriori	1	66.796 s	4.545 s	
Apriori>IsContainedIn	684500	61.459 s	6.486 s	
ismember	684587	54.979 s	11.763 s	
ismember>ismemberR2012a	684587	43.217 s	23.371 s	
ismember>ismemberBuiltinTypes	684587	19.845 s	19.845 s	
union	4636	0.758 s	0.125 s	
union>unionR2012a	4636	0.633 s	0.142 s	
unique	4636	0.491 s	0.285 s	
unique>uniqueR2012a	4636	0.206 s	0.206 s	
IsSame	1297	0.022 s	0.022 s	
DisplayRules	1	0.016 s	0.001 s	
mat2str	4	0.008 s	0.008 s	
num2str	8	0.007 s	0.003 s	
GetNonTrivialSubsets	2	0.006 s	0.001 s	
dec2binvec	4	0.005 s	-0.000 s	
close	1	0.003 s	0.001 s	
dec2bin	4	0.002 s	0.002 s	
str2num	4	0.002 s	0.002 s	
int2str	2	0.002 s	0.002 s	
num2str>handleNumericPrecision	6	0.002 s	0.000 s	
num2str>convertUsingRecycledSprintf	6	0.002 s	0.002 s	
close>safegetchildren	1	0.001 s	0.001 s	
close>request_close	1	0.001 s	0.001 s	
blanks	4	0.001 s	0.001 s	
close>getEmptyHandleList	1	0 s	0.000 s	
close>checkfigs	1	0 s	0.000 s	
fliplr	4	0 s	0.000 s	
str2num>protected_conversion	4	0 s	0.000 s	
mat2str>isenumeration	4	0 s	0.000 s	

Şekil 4.4. Apriori algoritması süre analizi.

4.3.3.2. Veri setinin FP-Growth algoritması sonuçları

Çizelge 4.3’de tespit edilen iki kuralın destek değerleri 0.5’den, kaldıraç değerlerinin 1’den büyük olduğu tespit edilmiştir.

Çizelge 4.3. Fp-Growth algoritması sonucu bulunan kurallar.

Ürün No	Ürün No	Destek Değeri	Güven Değeri	Kaldıraç
3	68	0.502	0.698	1.389
68	3	0.502	1	1.389

Bulunan sonuç kuralı su ve meyve suyu arasındaki ilişkiden kurulmaktadır. Su-meyve suyu ile meyve suyu-su ürünleri arasında karşılıklı ilişki tespit edilmektedir.

Bu kuralların kesin kural olabilmeleri için girilen güven değerinden daha büyük güven değerine sahip olmaları gerekmektedir. Çizelge 4.3’de görülmektedir ki depodan çıkan ürünlerin %50’sinde su ve meyve suyu depodan birlikte çıkarılmakta olup %69’unda depodan su çıkartılırken meyve suyunun da çıkartıldığı görülmüştür. Depodan meyve suyu ile suyun birlikte çıkarılma oranı %50 iken depodan meyve suyunun çıkartılırken tamamında su beraberinde depodan çıkarılmıştır. Şekil 4.5’de FP-Growth algoritmasının tespit ettiği kurallar gösterilmiştir.

```
Fp-Growth Kuralları:
Kural #1: 68 --> 3
    Support = 0.50249
    Confidence = 1
    Lift = 1.3893
```

Şekil 4.5. FP-Growth algoritması uygulama kural sonuçları.

FP-Growth algoritmasının kural oluşturma süresi 16 saniye sürmüş olup Şekil 4.6’da çalışan tüm fonksiyonların çalışma süreleri detaylı bir şekilde gösterilmiştir.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
mainfp	1	16.255 s	7.738 s	
FPGrowth	1	8.471 s	2.687 s	
ismember	46250	3.612 s	0.747 s	
ismember>ismemberR2012a	46250	2.865 s	1.638 s	
GetPowerSet	506	1.486 s	0.207 s	
dec2binvec	9787	1.279 s	0.249 s	
ismember>ismemberBuiltinTypes	46250	1.227 s	1.227 s	
union	4625	0.686 s	0.108 s	
union>unionR2012a	4625	0.578 s	0.126 s	
unique	4625	0.452 s	0.220 s	
dec2bin	9787	0.423 s	0.423 s	
str2num	9787	0.388 s	0.281 s	
unique>uniqueR2012a	4625	0.232 s	0.232 s	
fliplr	9787	0.125 s	0.125 s	
str2num>protected_conversion	9787	0.107 s	0.107 s	
blanks	9787	0.094 s	0.094 s	
DisplayRules	1	0.031 s	0.015 s	
mat2str	2	0.016 s	0.016 s	
close	1	0.015 s	0.000 s	
close>safefetchchildren	1	0.015 s	0.015 s	
close>getEmptyHandleList	1	0 s	0.000 s	
close>checkfigs	1	0 s	0.000 s	
close>request_close	1	0 s	0.000 s	
GetNonTrivialSubsets	1	0 s	0.000 s	
IsSame	6	0 s	0.000 s	
num2str	4	0 s	0.000 s	
int2str	2	0 s	0.000 s	
num2str>handleNumericPrecision	2	0 s	0.000 s	
num2str>convertUsingRecycledSprintf	2	0 s	0.000 s	
mat2str>isenumeration	2	0 s	0.000 s	

Şekil 4.6.FP-Growth algoritması süre analizi.

4.3.4. Farklı güven değerleri ile sonuç analizi

Destek değeri sabit 0.50, güven değeri değişken olduğu zaman Apriori algoritmasının kural oluşturma süresi Çizelge 4.4’de verilmiştir.

Çizelge 4.4. Apriori algoritması ile farklı güven değerleri sonuçları.

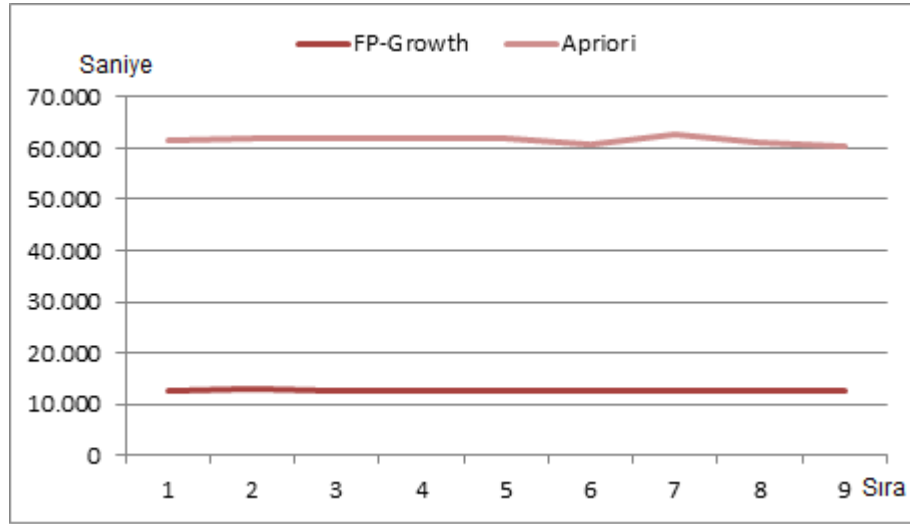
Sıra	Destek Değeri	Güven Değeri	Süre (sn)
1	0.50	0.40	12.507
2	0.50	0.45	12.925
3	0.50	0.50	12.420
4	0.50	0.55	12.496
5	0.50	0.60	12.640
6	0.50	0.65	12.651
7	0.50	0.70	12.656
8	0.50	0.75	12.460
9	0.50	0.80	12.390

Destek değeri sabit 0.50, güven değeri değişken olduğu zaman FP-Growth algoritmasının kural oluşturma süresi Çizelge 4.5’de verilmiştir.

Çizelge 4.5. FP-Growth algoritması ile farklı güven değerleri için süre analizi.

Sıra	Destek Değeri	Güven Değeri	Süre (sn)
1	0.50	0.40	48.995
2	0.50	0.45	49.108
3	0.50	0.50	49.389
4	0.50	0.55	50.453
5	0.50	0.60	49.194
6	0.50	0.65	48.084
7	0.50	0.70	50.235
8	0.50	0.75	48.844
9	0.50	0.80	48.065

Güven değerinin değişkenliği süreyi çok fazla değiştirmez. Güven değerinin artması final kural sayısını azaltacağı için süreyi saniyeler kadar azaltır. Çünkü güven değeri kural oluşumunu değil kurallar içerisindeki son kuralın bulunması konusunda yardımcı olur (Şekil 4.7).



Şekil 4.7. Değişken güven değerleri için kural oluşturma süreleri.

4.3.5. Farklı destek değerleri ile sonuç analizi

Güven değeri sabit 0.75, güven değeri değişken olduğu zaman Apriori ve FP-Growth algoritmalarının kural oluşturma süresi ve kural sayıları Çizelge 4.6'da verilmiştir.

Çizelge 4.6. Farklı destek değerleri ile sonuç analizi.

Ölçütler		FP-Growth		Apriori	
Destek	Güven	Çalışma Süresi (sn)	Kural Sayısı	Çalışma Süresi (sn)	Kural Sayısı
0.60	0.75	9.55	0	39.01 saniye	0
0.55	0.75	9.78	0	41.63 saniye	0
0.50	0.75	13.13	1	48.96 saniye	2
0,45	0.75	66.58	1	71.11 saniye	6
0,40	0.75	1518.85	1	106.52 saniye	7
0,35	0.75	10554.25	9	192.93 saniye	16

Çizelge 4.6’da destek değeri 0.35, güven değeri 0.75 girildiğinde FP-Growth algoritması kural listesi (**Ek-3**) ve Apriori algoritması için kural listesi (**Ek-4**) maksimum sayıda kural elde edilmiştir.

4.3.6. Apriori ve FP-Growth algoritmalarının karşılaştırılması

4625 hareketlilik 216 üründen oluşan veri, veri madenciliği aşamalarında biri olan veri indirgeme işlemi sonrasında 106 ürün çeşitliliğine indirgenmiştir.

Hareketliliklerde sepetteki ürün çeşitliliğinin az ve hep aynı ürünün satışı yapılmış olması sebebiyle 500 üzeri hareketlilik içerisinde yer alan ürün sayısı Şekil 4.8’de gösterildiği üzere 7 üründen oluşmaktadır.

Name	Count
'n'	4625
3	3329
68	2324
71	1791
50	1311
4	939
22	690
8	539

Şekil 4.8.Uygulama ürün destek sayıları.

Yukarıda bilgiler ışığında destek ve güven ölçütleri belirlenmiştir. Sağlıklı sonuçlar alabilmek için destek ve güven değerleri farklı girilerek farklı kural oluşumları izlenmiştir. Destek değeri 0.35, güven değeri 0.75 girildiğinde iki algoritmada maksimum sayıda kural elde edilmiştir. Çıkan kurallarda benzer kurallar olduğu gibi farklı kurallar da oluşmuştur. Apriori ve Fp-Growth algoritmaları kural sıralamasında üstlerde bulunan meyve suyu-su, asitli içecekler-su, meyve suyu-kek ilişkileri dikkat çekmektedir.

Destek ve güven değerleri oluşan kural sayısını ve kural oluşum süresini doğrudan belirlemektedir. Çizelge 4.6’da görüldüğü üzere destek değeri sıfıra yaklaştıkça her iki algoritmanın da kural oluşturma süresi artmaktadır. Bu artış miktarlarında güven değeri sabit tutularak destek değerleri 0.60 ve 0.35 olarak analiz yapıldığında;

Destek değeri 0.60 iken FP-Growth algoritması Apriori algoritmasına göre 4 kat hızlı sonuç vermekte iken destek değeri 0.35 olduğunda bu durum terse dönmüş olup Apriori algoritması FP-Growth algoritmasına göre 50 kat daha hızlı kural oluşturmaktadır. (Bkz. Çizelge 4.6) Destek değeri sıfıra yaklaştıkça sık kullanılan öğe sayısı artmakta ve ağaç büyümektedir. Bu durum kural oluşturma süresine olumsuz yansımaktadır. Destek ve güven değerleri sıfıra yaklaştıkça iki algoritmada da oluşan kural sayısı artar. Öğe sayısının artması sebebiyle de kural oluşturma süreleri de artmaktadır.

Ürünler arasındaki ilişkiyi, kural oluşum sayısını artırmak için destek değeri sıfıra yaklaştırmak gerekir. Her iki algoritmadan alınan sonuçlara bakıldığında su, meyve suyu, asitli içecek ürünlerinden sonra en yakın ürün ilişkisi %38 ile depodan çıktığı görülmüştür (**Ek-3**) (**Ek-4**). Farklı kurallar tespit etmek için destek değerini %38 yani 0.35 girmek gerektiğidir. Destek değeri %35 girildiğinde kural oluşum süresi FP-growth algoritması için yaklaşık 3 saat sürmüştür. Veri tabanı büyüdükçe algoritmalar arasındaki süre farkının katlanarak arttığı gözlenmiştir.

Kural oluşum sayısı olarak baktığımızda Apriori algoritması FP-Growth algoritmasına göre daha fazla kural oluşturmuştur.

Yapılan incelemeler sonucunda kural oluşum süresinde FP-Growth algoritmasının daha etkili ve dikkate alınması gereken kurallar oluşturduğu sonucuna varılmıştır. Veri tabanının büyük, destek değerinin sıfıra yaklaştığı durumlarda ise FP-Growth algoritmasının Apriori algoritmasına göre daha kararlı sonuçlar tespit ettiği görülmüştür. Destek değeri 0.35 girildiğinde daha fazla öğenin destek değerini sağlamış ve Apriori algoritmasında belirlenen adaya nesne küme sayısını da artırmaktadır. Ağaç oluşturan FP-Growth için aday nesne kümesi Apriori algoritmasında olduğu gibi bir artış göstermemekte olup daha geçerli sonuçlar üretmesini sağlamıştır.

5. SONUÇLAR

Marketlerde ürünlerin çeşitliliğinin artması, raflarda ürünlerin sunulmasını zorlaştırdığı kadar depoda bu ürünlerin saklanabilmesini de zorlaştırmaktadır. Raflarda müşterinin isteklerini temin etmek planlı bir depo düzeninden geçmektedir. Bu sorunların çözümü için depodaki ürünlerin depo içerisinde en uygun yere konulması amaçlanmıştır. Bu amacı gerçekleştirmek için bazı bilgilere ihtiyaç duyulmaktadır. Bu bilgiler şu şekilde sıralanabilir. Ürünlerin hangi zaman aralığında hangi ürünlerle birlikte satıldığını bilmek depo düzeni sağlamamız hususunda önemli bir bilgi olarak karşımıza çıkmaktadır. Ürünler arasındaki ilişkileri tespit ederek depoda bu ürünleri ortak tüketilme ihtimaline göre depo içerisinde uygun yere yerleştirmeleri planlanabilir. Yine ürünlerin raflarda satılma sıklıklarının arttığı dönemler tespit edilerek depoda tutulma dönemleri ve miktarları planlanabilir. Bu sayede yeterli iş gücü ve kullanışlı planlanmış bir depo ile kazanç artırılabilir. Bu amaçla bir marketin deposundaki ürünlerin birlikte mağaza raflarına çıkış ilişkileri tespit edilerek verim artırılması amaçlanmıştır.

Bu tez çalışması kapsamında veri tabanı olarak açık kaynak kodlu WEKA programı market verisi kullanılmıştır. Arff uzantılı dosya öncesinde csv dosyasına sonrasında Microsoft Excel kullanılarak Matlab *.mat uzantılı dosya içerisinde tablo oluşturularak dönüştürülmüştür.

Uygulama birliktelik kural analizi algoritmalarından Apriori ve FP-Growth algoritmaları kullanılarak çalıştırılmış ve farklı destek ve güven değerleri için analiz edilmiştir. Apriori ve FP-Growth algoritmaları yakın sonuçlar verse de kural sayılarında ve kural oluşturma sürelerinde farklılıklar gözlenmiştir.

Kural oluşturma sayısı olarak bu iki algoritmayı karşılaştıracak olursak; FP-Growth algoritması daha net ve az sayıda kural bulmaktadır. Apriori algoritmasında veri tabanı büyüdükçe ve destek değeri sıfıra yaklaştıkça FP-Growth algoritmasına göre çok daha fazla kural oluşturmaktadır. Kural sayısının fazla olması da oluşan kurallara olan güveni azaltmaktadır.

Kural oluşturma süresi olarak Apriori ve FP-Growth algoritmalarını inceleyecek olursak; tez çalışmasında 0.35 minimum destek değeri girildiğinde Apriori algoritması 3 dakika içerisinde kural oluştururken, FP-Growth algoritmasının kural oluşturma süresi 3 saate yakın sürmüştür. (Bkz. Çizelge 4.6)

KAYNAKLAR

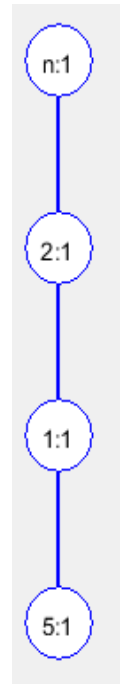
- Silahtaroglu, G. (2016). *Veri Madenciliği: Kavram ve Algoritmaları*. Papatya Bilim, İstanbul, 9-11
- Han, J. & Kamber, M. & Pei, J. (2012). *Data mining: Concepts and Techniques*. Morgan Kaufmann Publishers 3th ed., Waltham, 5-8.
- Ngai, E.W.T., Xiu, L., Chau, D.C.K. (2009), Application Of Data Mining Techniques In Customer Relationship Management: A Literature Review And Classification. *Expert Systems with Applications*, 36(2), 2592-2602.
- Prasanna, S. & Ezhilmaran D., (2013). An analysis on Stock Market Prediction using Data Mining Techniques. *International Journal of Computer Science & Engineering Technology (IJCSET)*, 4, 49-51.
- Srinivasa, K.G., Venugopal, K.R., Patnaik, L.M., (2007). A Self- Adaptive Migration Model Genetic Algorithm For Data Mining Applications, *Information Sciences*, 177, 4295-4313.
- Aydoğan F. (2003). *E-Ticarette Veri Madenciliği Yaklaşımlarıyla Müşteriye Hizmet Sunan Akıllı Modüllerin Tasarımı Ve Gerçekleştirimi*. Yüksek Lisans Tezi, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Şen, F. (2008). *Veri Madenciliği ile Birliktelik Kuralların Bulunması*. Yüksek Lisans Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Sakarya.
- <http://crisp-dm.eu/>, (Erişim Tarihi: 21.03.2019).
- Akpınar, H. (2000). Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği. *İ.Ü. İşletme Fakültesi Dergisi*, 29(1), 1-22.
- Özdemir, A., & Yalçın, F. A., & Çam, H. (2009). Veri Tabanında Bilgi Keşfi Süreci: Gümüşhane Devlet Hastanesi Uygulaması. *Sosyal Ekonomik Araştırmalar Dergisi*, 1(20), 347-366.
- Özkan, D. Y. (2013). *Veri Madenciliği Yöntemleri*, Papatya Yayıncılık, İstanbul, 28-29
- Adriaans, P., & Zantinge, D. (1997). *Data Mining*, Addison Wesley Longman Publishing, Bostan, 61-154
- Aytaç, M. B., & Bilge, H. Ş. (2013). Tele Pazarlama Verilerinin Birliktelik Kurallarıyla ve CRISP-DM Yöntemleriyle Analiz Edilmesi. *Aksaray Üniversitesi İİBF Dergisi*, 2, 25-40
- Catherina, B. & Rinta, R. E. (2001). Overview of Data Mining for Customer Behavior Modelling. *VTT Information Technology*, 18, 8-32
- Zerman, M. (2018). *Birliktelik Kuralı Algoritmaları ile Büyük Veriler Üzerinden Analitik Analizler: Havaalanı Örneği*. Yüksek Lisans Tezi, Haliç Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

KAYNAKLAR (Devam Ediyor)

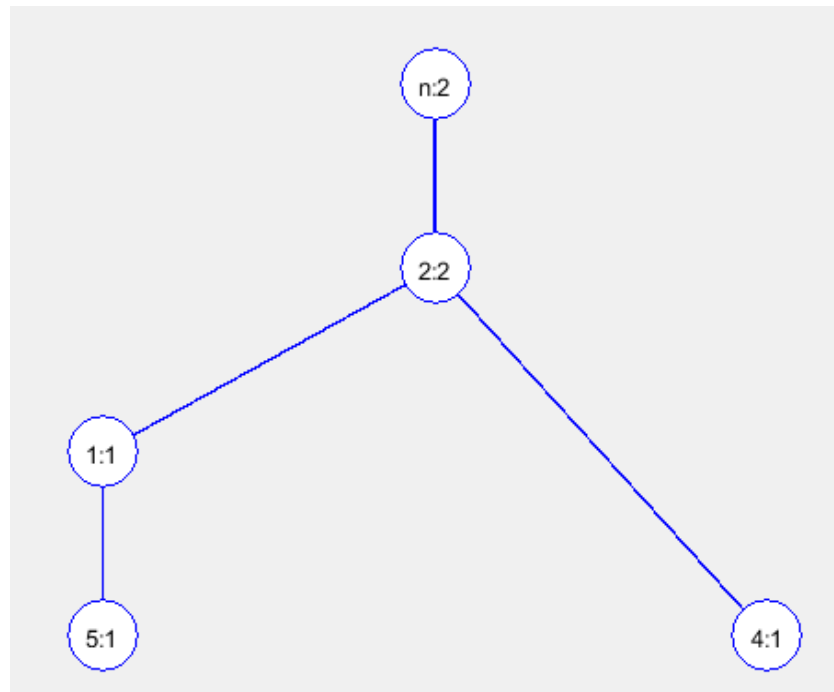
- Karagöz, N. E. (2007). *Market Veri Tabanında Veri Madenciliği Uygulaması*. Yüksek Lisans Tezi, İstanbul Ticaret Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul.
- Özçınar, H. (2006). *KPSS Sonuçlarının Veri Madenciliği Yöntemleriyle Tahmin Edilmesi*, Yüksek Lisans Tezi, Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü, Denizli.
- Bramer, M. (2007). *Principles of Data Mining*. Springer, London, 11-20
- Ateş, Y. & Karabatak, M. (2017). Nicel Birliktelik Kuralları İçin Çoklu Minimum Destek Değeri. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 29 (2), 57-65.
- Agrawal, R. & Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases, 20th *VLDB Conference*, 12-15 September, Chile, 487-499.
- Yurtsever, U.(2002). *Veri Madenciliği Ve Uygulaması*, Yüksek Lisans Tezi, Sakarya Üniversitesi, Sosyal Bilimler Enstitüsü, Sakarya.
- Kılıç, Y. (2009). *Mining Association Rules For Quality Related Data In An Electronics Company*. Master Thesis, Middle East Technical University, Industrial Engineering, Ankara.
- Kantardzic, M. (2011). *Data mining: concepts, models, methods, and algorithms*. IEEE Press, Canada, 87-138.
- Özçalıcı, M. (2017). Veri Madenciliğinde Birliktelik Kuralları ve İkinci El Otomobil Piyasası Üzerine Bir Uygulama, *Ordu Üniversitesi Sosyal Bilimler Araştırmaları Dergisi*, 7(1), 45-58.
- Karaibrahimoğlu, K. (2014). *Veri Madenciliğinden Birliktelik Kuralı ile Onkoloji Verilerinin Analiz Edilmesi: Meram Tıp Fakültesi Onkoloji Örneği*, Doktora Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Konya.
- Timor, M. & Şimşek, U. (2008). Veri Madenciliğinde Sepet Analizi İle Tüketici Davranışı Modellemesi, *Yönetim Dergisi*, 19,3-10.
- Gürsoy, U. T. Ş. (2012). *Uygulamalı Veri Madenciliği: Sektörel Analizler*, Pegem Akademi, Ankara, 125.
- Söylemez, İ. & Doğan, A. & Özcan, U. (2016). Trafik Kazalarında Birliktelik Kuralı Analizi: Ankara İli Örneği, *Ege Akademik Bakış*, 16, 11-20.
- Delibaş, E. (2010). *Birliktelik Analizi ile Reçete İlaç Satışları Üzerinde Bir Uygulama*, Yüksek Lisans Tezi, Cumhuriyet Üniversitesi, Sosyal Bilimler Enstitüsü, Sivas
- Surendiran, R. & Rajan, K.P. & Sathish Kumar, M. (2010). Study on the Customer targeting using Association Rule Mining, *International Journal on Computer Science and Engineering*, 2(1), 2483-2484

KAYNAKLAR (Devam Ediyor)

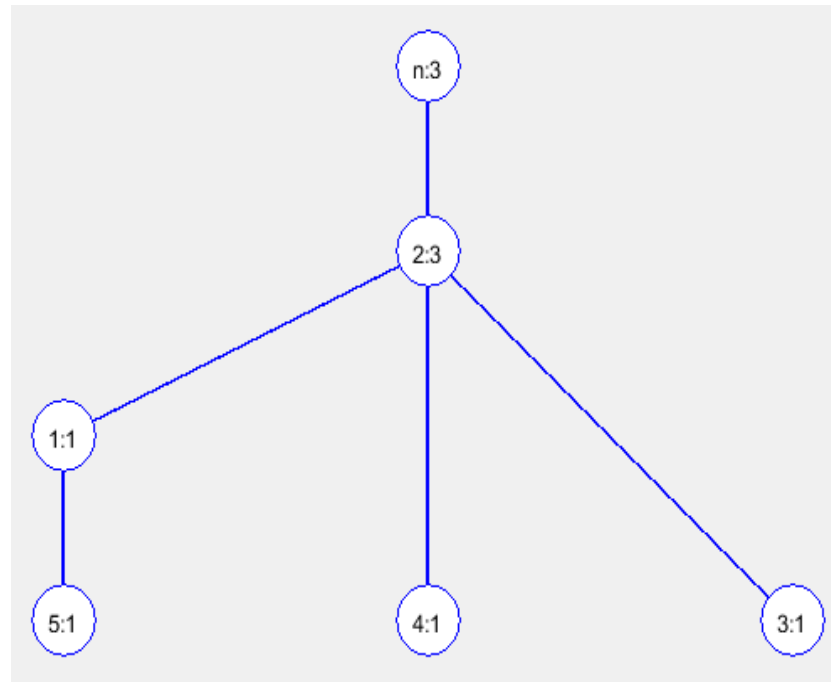
- Kumbhare, T. A., & Chobe, S. V. (2014). An overview of association rule mining algorithms, *International Journal of Computer Science and Information Technologies*, 2927, 927-930.
- Döşlü, A. (2008). *Veri Madenciliğinde Market Sepet Analizi ve Birliktelik Kurallarının Belirlenmesi*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Eker, M. E. (2016). *Veri Madenciliğinde Apriori Algoritmasının Sınava Verileri Üzerinde Uygulanması*, Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi, Fen Bilimleri Enstitüsü, Samsun.
- Dunham, M. H., & Xiao, Y., & Gruenwald, L., Hossain, Z. (2008). A survey of association rules, *2015 Fifth International Conference on Advanced Computing & Communication Technologies*, 21-22 Feb. 2015, India, 5.
- Özdoğan, G. Ö. (2010). *Öbek Bilgisayarlarda Paralel FP- Growth Gerçekleştirme*, Yüksek Lisans Tezi, TOBB Ekonomik ve Teknoloji Enstitüsü, Fen Bilimleri Enstitüsü, Ankara.
- <http://deryagunduz.com/?tag=fp-growth-algoritmasi/>, (Erişim Tarihi: 23.03.2019).
- Erpolat, S. (2012). Otomobil Yetkili Servislerinde Birliktelik Kurallarının Belirlenmesinde Apriori ve FP-Growth Algoritmalarının Karşılaştırılması, *Anadolu Üniversitesi Sosyal Bilimler Dergisi*, 12(2), 137-146.
- Dunham, M. H., & Xiao, Y., & Gruenwald, L., Hossain, Z. (2008). A survey of association rules, *2015 Fifth International Conference on Advanced Computing & Communication Technologies*, 21-22 Feb. 2015, India, 5.
- David, S. & Reutemann, P. (2006). *WEKA Experimenter Tutorial for Version 3-4*. Waikato Üniversitesi, Yeni Zelanda, 39.
- Rong, J. & Vu, H. & Law, R. & Li, G. (2012). A behavioral analysis of web sharers and browsers in Hong Kong using targeted Association Rule Mining, *Tourism Management*, 33(4), August 2012, 731-740.
- Verma, A. & Khan, S.D, & Maiti, J. & Krishna, O.B. (2014), Identifying patterns of safety related incidents in a steel plant using Association Rule Mining of incident investigation reports, *Safety Science*, 70, December 2014, 89-98
- Patil, B. M. & Joshi, R.C. & Tonsniwal, D. (2010), Association Rule for Classification of Type-2 Diabetic Patients, *2010 Second International Conference on Machine Learning and Computing*, 9-11 Feb. 2010, India,

EK-1: Fp-Growth Algoritması Ağaç Yapısı

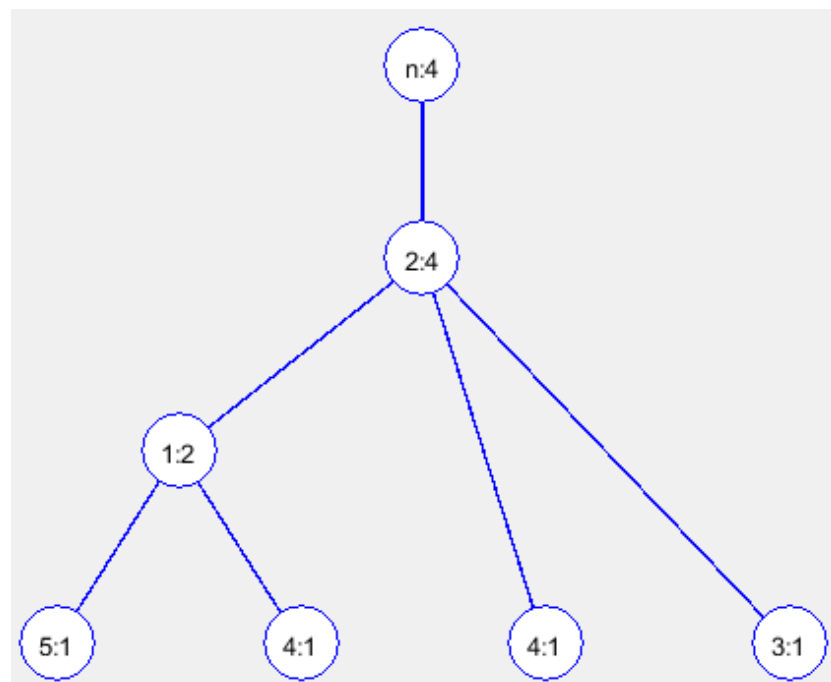
(A)



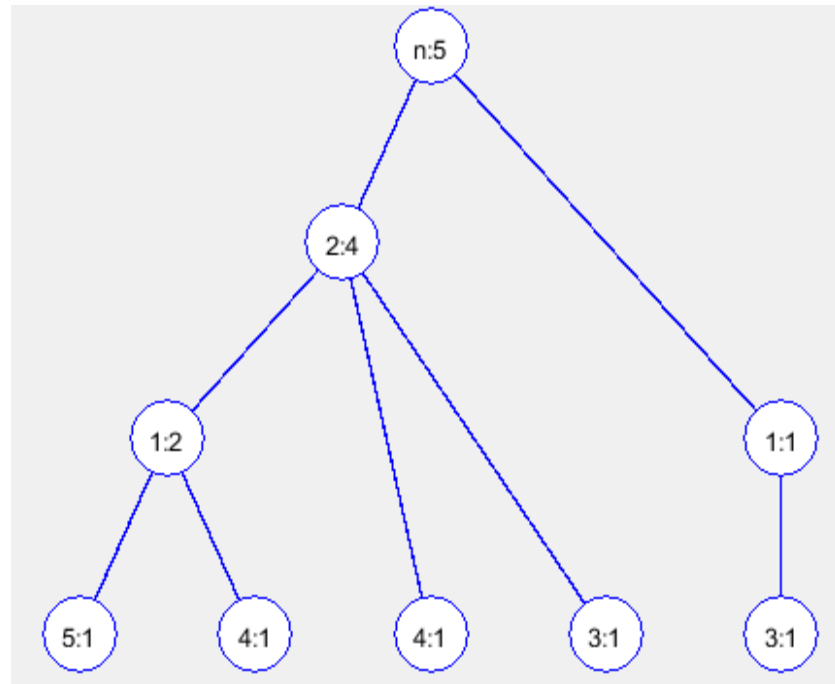
(B)



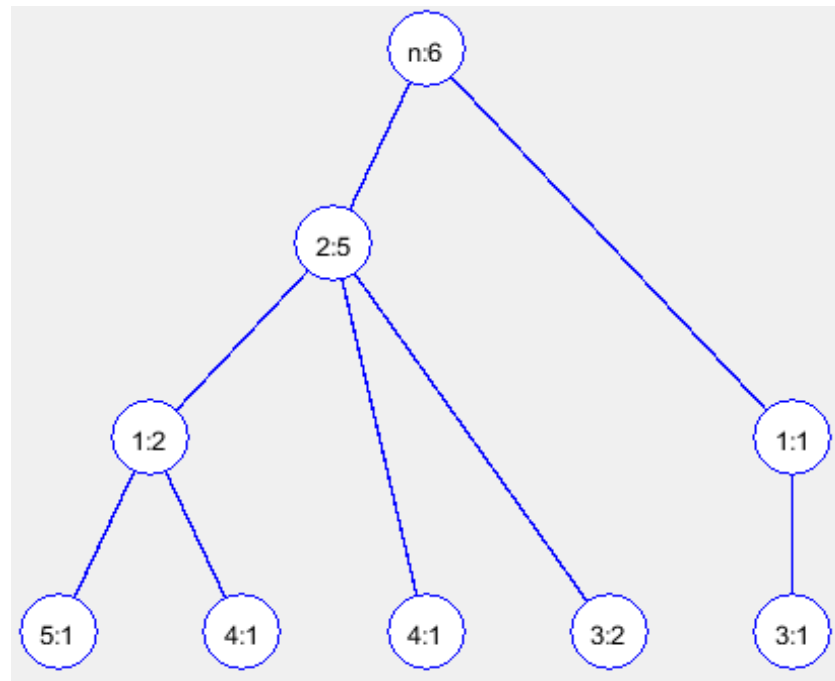
(C)



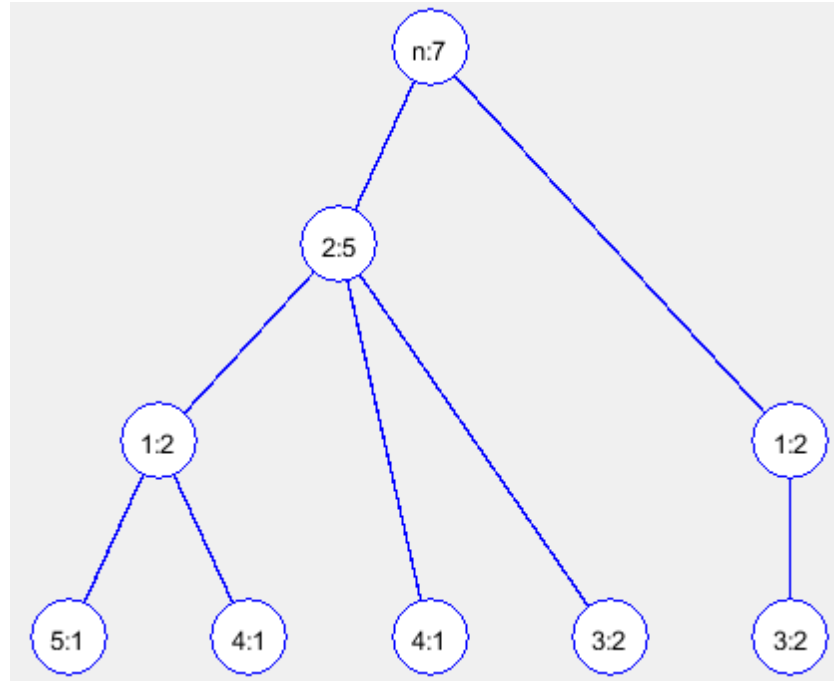
(D)



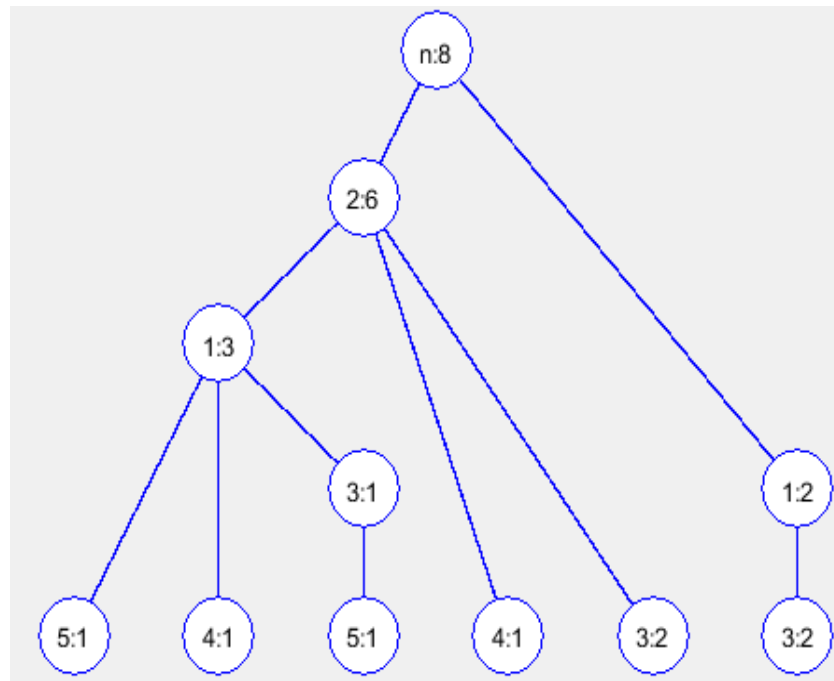
(E)



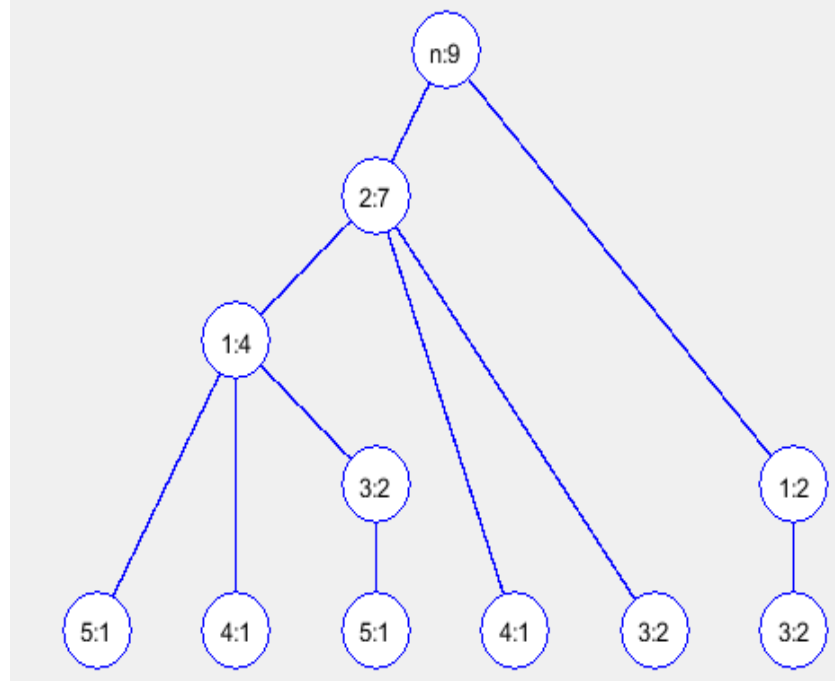
(F)



(G)



(H)



(I)

Ek-2: Uygulama Veri Seti

Sıra	Ürünler
1	Bulaşık Eldiveni
2	Un
3	Su(500ml)
4	Su(1500ml)
5	Su(5lt)
6	Su(19lt)
7	Çay
8	Bisküvi
9	Konserveler
10	Havlu Kâğıt
11	Peçete
12	Sos
13	Pekmez
14	Parti Malzemeleri
15	Yüzey Temizleyici
16	Kahve
17	Zeytin
18	Şekerleme Ürünleri
19	Puding
20	Bulaşık Bezi
21	Dezenfekte Malzemeleri
22	Dondurulmuş Ürünler
23	Jilet
24	Bahçe Malzemeleri
25	Baharatlar
26	Reçel
27	Böcek İlaçları
28	Hayvan Mamaları
29	Makine Deterjanları

Ek-2: Uygulama Veri Seti (Devam Ediyor)

Sıra	Ürünler
30	Makarna
31	Bakliyat
32	Çöp Poşeti
33	Deterjan
34	Buzdolabı Poşeti
35	Bal
36	Ped
37	Makyaj Mal.
38	Jöle
39	Deodorant
40	Tıraş Malzemeleri
41	Sağlık Malzemeleri
42	Saç Bakım Mal.
43	Sakız
44	Krem
45	Margarin
46	Fırın Poşeti
47	Kırtasiye Malzemeleri
48	Çamaşır Suyu
49	Bulaşık Deterjanı
50	Asitli İçecekler
51	Kalıp Sabun
52	Soğan
53	Tuvalet Kâğıdı
54	Soda
55	Ampul
56	Süt Ürünleri
57	Meyve
58	Islak Mendil

Ek-2: Uygulama Veri Seti (Devam Ediyor)

Sıra	Ürünler
59	Makarna
60	Şehriye
61	Bulgur
62	Pirinç
63	Kahve Kreması
64	Parfümeri
65	Şeker
66	Sıvı Sabun
67	Sıvı Yağ
68	Meyve Suyu
69	Kurutulmuş Ürünler
70	Bebek Bezi
71	Kek
72	Turşu
73	Duş Jeli
74	Süpürge
75	Elektrik Malzemeleri
76	Tuhafiye
77	Mutfak Malzemeleri
78	Tuz
79	Çorap
80	Plastik Eşyalar
81	Cips
82	Bitkiler
83	Patates
84	Sebzeler
85	Süt(1000ml)
86	Süt(200ml)
87	Hazır Çorba

Ek-2: Uygulama Veri Seti (Devam Ediyor)

Sıra	Ürünler
88	Kraker
89	Bebek Yağı
90	Bebek Pudrası
91	Bebek Maması
92	Araba Kokusu
93	Araba Döşemesi
94	Oda Spreyi
95	Diş Macunu
96	Diş Fırçası
97	Tatlandırıcılar
98	Fasulye
99	Mercimek
100	Nohut
101	Bulgur
102	Şampuan
103	Pirinç
104	Domates Salçası
105	Biber Salçası
106	İrmik

Ek-3: Uygulama FP-Growth Algoritması Kural Listesi

Fp-Growth Kurallar:

Kural #1: 68 --> 3
Support = 0.50249
Confidence = 1
Lift = 1.3893

Kural #2: 71 --> 3
Support = 0.38724
Confidence = 1
Lift = 1.3893

Kural #3: 68 --> 71
Support = 0.38724
Confidence = 0.77065
Lift = 1.9901

Kural #4: 71 --> 68
Support = 0.38724
Confidence = 1
Lift = 1.9901

Kural #5: 68 --> [3 71]
Support = 0.38724
Confidence = 0.77065
Lift = 1.9901

Kural #6: [3 68] --> 71
Support = 0.38724
Confidence = 0.77065
Lift = 1.9901

Kural #7: 71 --> [3 68]
Support = 0.38724
Confidence = 1
Lift = 1.9901

Kural #8: [3 71] --> 68
Support = 0.38724
Confidence = 1
Lift = 1.9901

Kural #9: [68 71] --> 3
Support = 0.38724
Confidence = 1
Lift = 1.3893

Elapsed time is 10554.259204 seconds.

Ek-4: Uygulama Apriori Algoritması Kural Listesi

Apriori Kurallar:

Kural #1: [50 68] --> 3
Support = 0.36389
Confidence = 0.82622
Lift = 1.1479

Kural #2: [50 71] --> 3
Support = 0.35849
Confidence = 0.81877
Lift = 1.1375

Kural #3: [68 71] --> 3
Support = 0.38724
Confidence = 0.81151
Lift = 1.1274

Kural #4: 53 --> 3
Support = 0.39589
Confidence = 0.80061
Lift = 1.1123

Kural #5: 8 --> 3
Support = 0.45016
Confidence = 0.79954
Lift = 1.1108

Kural #6: 50 --> 3
Support = 0.50508
Confidence = 0.79537
Lift = 1.105

Kural #7: 31 --> 3
Support = 0.38378
Confidence = 0.79029
Lift = 1.098

Kural #8: 68 --> 3
Support = 0.50249
Confidence = 0.78487
Lift = 1.0904

Kural #9: 4 --> 3
Support = 0.47373
Confidence = 0.78418
Lift = 1.0895

Kural #10: 22 --> 3
Support = 0.46011
Confidence = 0.78379
Lift = 1.0889

Ek-4: Uygulama Apriori Algoritması Kural Listesi (Devam Ediyor)

```
Kural #11: [3 71] --> 68
    Support = 0.38724
    Confidence = 0.77937
    Lift = 1.2174

Kural #12: 17 --> 3
    Support = 0.36973
    Confidence = 0.77692
    Lift = 1.0794

Kural #13: 71 --> 3
    Support = 0.49686
    Confidence = 0.77609
    Lift = 1.0782

Kural #14: 30 --> 3
    Support = 0.3907
    Confidence = 0.77587
    Lift = 1.0779

Kural #15: [3 68] --> 71
    Support = 0.38724
    Confidence = 0.77065
    Lift = 1.2037

Kural #16: 6 --> 3
    Support = 0.40389
    Confidence = 0.75873
    Lift = 1.0541
```

Elapsed time is 192.937654 seconds.

ÖZ GEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : Emrah TOKYÜREK
Doğum yeri ve Tarihi : Nevşehir, 12/09/1987



Eğitim Durumu

Lisans Öğrenimi :Zonguldak Bülent Ecevit Üniversitesi, Karabük Teknik Eğitim Fakültesi, Bilgisayar Öğretmenliği (2009)
Bildiği Yabancı Diller :İngilizce
Bilimsel Faaliyetler : --

İş Deneyimi

Stajlar :Bilgisayar Hastanesi – Nevşehir (2008)
Çalıştığı Kurumlar :Ahmet Toprak Ç.P.L. (2009-2010)
Bala Atatürk M.T.A.L.(2010-2016)
Gölbaşı M.T.A.L.(2016-)

İletişim

Adres :Seğmenler Mah. Cemal Gürsel Cad. 945. Sokak No:2/1
Gölbaşı/Ankara
E-Posta Adresi :tokyurek.emrah@gmail.com

Akademik Çalışmalar

1. Yüzgeç U., Tokyürek E.,” WEKA ile Birliktelik Kural Çıkarım Algoritmaları Kullanılarak Market Sepet Analizi”, International Science and Academic Congress'18 (INSAC-18), 2.Cilt pp.280-286, Konya, Aralık 8-9, 2018.