T.C.
İSTANBUL MEDENİYET ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
MÜHENDİSLİK YÖNETİMİ ANABİLİM DALI

VERİ MADENCİLİĞİ MARKET SEPETİ ANALİZİ: APRIORI VE FP-GROWTH ALGORİTMALARI KARŞILAŞTIRMASI

YÜKSEK LİSANS TEZİ

MERVE KARADEMİR

OCAK 2021

T.C.
İSTANBUL MEDENİYET ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
MÜHENDİSLİK YÖNETİMİ ANABİLİM DALI

VERİ MADENCİLİĞİ MARKET SEPETİ ANALİZİ: APRIORI VE FP-GROWTH ALGORİTMALARI KARŞILAŞTIRMASI

YÜKSEK LİSANS TEZİ

MERVE KARADEMİR

DANIŞMANLAR
DR. ÖĞR. ÜYESİ FATİH ÖZTÜRK

OCAK 2021

BİLDİRİM

Hazırladığım tezin tamamen kendi çalışmam olduğunu, akademik ve etik kuralları gözeterek çalıştığımı ve her alıntıya kaynak gösterdiğimi taahhüt ederim.

İmza

Merve Karademir

Danışmanlığını yaptığım işbu tezin tamamen öğrencinin çalışması olduğunu, akademik ve etik kuralları gözeterek çalıştığını taahhüt ederim.

Dr. Öğr. Üyesi Fatih Öztürk

İMZA SAYFASI

Merve KARADEMİR tarafından hazırlanan 'Veri Madenciliği Market Sepeti Analizi: Apriori ve Fp-Growth Algoritmaları Karşılaştırması' başlıklı bu yüksek lisans tezi, Mühendislik Yönetimi Anabilim Dalında hazırlanmış ve jürimiz tarafından kabul edilmiştir.

JÜRİ ÜYELERİ	**İMZA**
Tez Danışmanı:	
Dr. Öğr. Üyesi Fatih ÖZTÜRK Kurumu: İstanbul Medeniyet Üniversitesi	.....
Üyeler:	
Doç. Dr. Hasan KÖTEN Kurumu: İstanbul Medeniyet Üniversitesi	.....
Doç. Dr. Berk AYVAZ Kurumu: İstanbul Ticaret Üniversitesi	.....

Tez Savunma Tarihi: …./ ……/ 202…

ÖNSÖZ

Tez çalışmamın planlanmasında kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösteren ve çalışmayı tamamlamak için de desteğini esirgemeyen değerli danışmanım Sayın Dr. Öğr. Üyesi Fatih Öztürk hocama, yine çalışmamızda bizden desteğini esirgemeyen Sayın Doç. Dr. Berk Ayvaz hocama, her zaman yanımda olan, benden desteğini esirgemeyen kıymetli eşim Tugay ve bana enerji veren canım oğlum Uras'a teşekkür ederim.

TÜRKÇE ÖZET

VERİ MADENCİLİĞİ MARKET SEPETİ ANALİZİ: APRIORI VE FP-GROWTH ALGORİTMALARI KARŞILAŞTIRMASI

Yüksek Lisans Tezi, Mühendislik Yönetimi Anabilim Dalı, Tezli Yüksek Lisans Programı
Danışman: Dr. Öğr. Üyesi Fatih Öztürk
Ocak 2021

Verinin ve veri biliminin öneminin her geçen gün katlanarak arttığı hepimiz tarafından yadsınamaz bir gerçek haline gelmiştir. Şirketler bu konuda yatırımlar yapmakta ve en doğru ve güvenilir veriyi elde etmek ve bu yönde adımlar atmayı hedeflemektedirler.

Veri yığınlarından değerli ve şirket çıkarları doğrultusunda işe yarar veriyi elde etme süreçlerinin kalıbını çizmiş olan veri madenciliği teknikleri ile ilgili veriler elde edilmekte ve birçok alanda bu altın değerindeki veriler kullanılmaktadır. Gerçek alışveriş verilerinin kullanıldığı bu çalışmada veri madenciliği algoritmalarından Market Sepeti Analizi sürecinde en sık kullanılan Apriori ve FP-Growth algoritmaları ile çalışıldı. Weka ve Python ile yapılan çalışma sonucunda Fp-Growth Algoritmasının Weka üzerinde Apriori Algoritmasına oranla çok daha hızlı çalıştığı fakat Python üzerinde yapılan çalışmada Apriori Algoritmasının, literatürdeki çoğu çalışmaya zıt olarak, FP-Growth algoritmasına oranla daha hızlı çalıştığı bilgisi ortaya konmuştur. Gerçek market verileri ile yapılan çalışmanın sonucunda market sahibine hangi ürünlerinin birlikte alındığı sonucunu sunularak satış politikasında karar verme sürecine destek sağlanmıştır.

Anahtar kelimeler: Veri Madenciliği, Market Sepeti Analizi, Birliktelik Analizi, WEKA, PyCharm, PYTHON, APRIORI, FP-Growth

İNGİLİZCE ÖZET

DATA MINING MARKET BASKET ANALYSIS: APRIORI AND FP-GROWTH ALGORITHMS COMPARISON

Master Thesis, Engineering Management Department, Engineering Management Master Program

Advisor: Asst. Prof. Dr. Fatih Ozturk

January 2021

It is the fact that the importance of data and data science is increasing day by day across the world. Companies make investments in this regard and aim to obtain the most accurate and reliable data and to take steps in this direction. It has been obtained data from the data mining techniques that draw the pattern of the processes of obtaining valuable and useful data in line with the interests of companies from the huge sets of data and these precious data has used in many areas. In this real market data study, Apriori and FP-Growth algorithms, which are the most frequently used data mining algorithms in the Market Basket Analysis process, were used. At the result of the study with using real market data were showed that Fp-Growth algorithtm produced results faster than the Apriori algorithm on WEKA but in contrast to the most of literature studies, the Apriori algorithm produced results much faster than the FP-Growth algorithm on Python. The results of which products were purchased together is presented to the market owner in order to support the decision-making process in the sales policy.

Keywords: Data mining, Market Basket Anaysis, Association rules, WEKA, PyCharm, Python, Apriori, FP-Growth

İçindekiler

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

1. GİRİŞ

Teknoloji hızla artmakta ve en küçük işletmelerden en büyük işletmelere kadar teknoloji kullanımı artık zorunluluktan çok gereklilik haline gelmektedir. Sadece işletmecilerin değil, işletmeci ile müşteri arasındaki köprüyü oluşturan iletişim aracı da yine teknoloji olmaktadır. Teknoloji sayesinde müşteriye ait işlem anında ki tüm konuşma detayları, alışveriş bilgileri, iade bilgileri, kişisel bilgiler vb. tutulmakta ve çoğu durumda bu verilerin tutulması zorunlu hale gelmektedir. Büyük müşteri kitlesi olan bir işletmeyi düşündüğümüzde bu gibi verilerin kaydedilmesi ve yıllarca saklanması ciddi bir maliyet kalemi olmaktadır. Dolayısı ile bu büyük veri yığını ve maliyetinin şirket için pozitif geri dönüşümünü sağlamak ciddi önem arz etmektedir.

Piyasanın değişim hızının fazla olması, bu değişim hızına ayak uydurmayı gereklilik haline getirmektedir. Piyasanın anlık değişimine ayak uydurmak, müşterinin yönelimlerini önceden tahmin etmek veya değişime rakip işletmelerden daha hızlı adapte olmak için şirketler sürekli arayış halindedirler. Veri madenciliği teknikleri bu şekilde verilerin yığınlar halini aldığı ve anlamlı verilerin türetilmesi gerektiği durumlarda kullanılan yöntemler içermektedir. Verinin madenciliğinin yapılarak şirketler için müşteri verilerinden ve birçok parametreye (yaş, cinsiyet, mevsim vb.) göre değerli veriler üretilir ve şirket için zararı önleyen ve kar elde etmeyi sağlayan sonuçlar ortaya konur.

Müşterilerin sıklıkla birlikte aldıkları ürünlerin tespit edilmesi işletmelerin yapacağı kampanyalar, satış esnasında ki dizaynları ve mevsimsel satış politikaları için önemli bir yer tutmaktadır. Birliktelik analizi, veri madenciliği tekniklerinden biridir ve birliktelik analizlerinden 'Market Sepeti Analizi' sayesinde büyük veri yığınlarından bu şekilde birbiri ile ilişkisi olan veriler ortaya çıkartılmaktadır. Market Sepeti Analizi olarak adlandırılmakta olmasına rağmen tıp, finans, bankacılık gibi birçok alanda kullanılmaktadır.

Birliktelik analizi ile ilgili çalışmalar 90'lı yıllardan bu yana yapılmaktadır. Birliktelik analizi ile ilgili en sık kullanılan algoritmalardan ikisi Apriori ve Fp-Growth 'dir. Algoritmaların birbirine göre artı ve eski yönlerinin bulunduğu durumlar bulunmakta ve bu nedenle koşullara göre uygun algoritma seçilmektedir.

Bu Tezin amacı, ismini X Süper Market olarak vereceğimiz İstanbul'daki bir marketin gerçek alışveriş verileri ile birliktelik analizini yaparak, sıklıkla bir arada satılan ürünlerini ortaya koymaktır. Bu analizi yaparken veri madenciliği birliktelik analizi algoritmalarından Apriori ve Fp-Growth algoritmaları kullanılmıştır. Her iki algoritmanın ürettiği sonuçlar ortaya konmuş ve üretim süreleri kıyaslanmıştır. Aynı zamanda diğer çalışmalardan farklı olarak bu çalışmada algoritmalar tek bir uygulama aracı ile değil birden fazla uygulama aracı kullanılarak analiz edilmiştir. Öncelikli olarak veri madenciliğinde sıklıkla kullanılan WEKA uygulaması kullanılarak her iki algoritma ile analiz yapılıp birliktelik kural setleri elde edilmiştir. Ardından PyCharm uygulaması ile Python dilinde yazılmış hazır kütüphaneler kullanılarak her iki algoritma çalıştırılmış ve birliktelik kural setleri elde edilmiştir. Her iki uygulama ve algoritmaların sonucunda çıkan analizler değerlendirilmiş ve aynı zamanda süreleri de kıyaslanmıştır.

Sonuç olarak FP-Growth algoritması WEKA üzerinde daha hızlı çıktı üretirken, Python üzerinde Apriori algoritması daha hızlı sonuç vermiş ve X Süper Marketi için %5 destek değerinin üzerinde gelen tüm ürün ilişkileri listelenmiş, maksimum %87 birliktelik oranı ile "Meyve ve Yeşillik alanların Sebze aldığı" sonucu ortaya çıkmıştır.

2. LİTERATÜR TARAMASI

90'lı yıllardan bu yana veri madenciliği ile ilgili birçok konu, kitap ve makale yazılıp yayınlanmıştır. Son yıllarda ise veri madenciliğinin önemi sağlık, finans, pazarlama, trafik, eğitim, üretim hatları, insan kaynakları vs. birçok alanda artmaktadır. Bunların yanı sıra araştırmacılar veri madenciliği algoritmalarına ve performansına da büyük önem vermektedirler.

Agrawal, Imielinski ve Swami (1993), büyük verilerin verildiği ve aralarındaki ilişkinin kurulmasının istendiği veri madenciliği işlemleri için bir algoritma geliştirdiler. Algoritma alışılmışın dışında tahminleme budama işlemlerini içeriyordu. O yıllarda veri tabanı madenciliğinin ticari ilgi ile de birleştirildiğinde veri tabanı için önemli yeni bir uygulama alanı olduğuna belirtmiştir.

Mythili & Shanavas (2013), "Performance Evaluation of Apriori and FP-Growth Algorithms" başlıklı çalışmasında Apriori ve FP-Growth Algoritmalarının performansları farklı destek değerleri ile denemiş ve destek değeri hangi seviyede olursa olsun FP-Growth algoritmasının Apriori algoritmasına göre çok daha fazla performanslı olduğunu ortaya koymuştur.

Ying, Sindakis, Aggarwal, Chen ve Su (2020), Singapur perakende sektöründe Big Data (büyük veri) yönetimi ile ilgili Singapur perakende sektöründeki 500 katılımcı ile bir çalışma yapmışlardır. Çalışmada gösteriyor ki büyük veriyi yönetmede önlerine çıkan engelleri aşmak için bölgedeki katılımcıların %26 'sı modern metod ve yazılımlar kullanıyor ve katılımcıların %40 'ı da büyük veriyi yönetmede kullanılan tekniklerin satışları arttırdığını, %34 'ü ise maliyeti azalttığını belirtiyor.

Leeflang, Verhoef, Dahlström ve Freundt (2014), dijital çağda pazarlamanın zorlukları ve çözümleri ile ilgili, dünya çapında 777 pazarlama yöneticisinden oluşan bir örneklemde, anket çalışması yapmışlardır. Çalışma sonucunda 'yetenek boşlukları' doldurmanın, 'kurumsal tasarım' ayarlamasının ve "eyleme dönüştürülebilir metrikler" uygulamanın sektördeki şirketlerin en büyük iyileştirme fırsatları olduğunu vurgulamıştır. Tablolar ve figürler ile göstermektedir ki dijital pazarlama dünyasında ki en önemli zorluk müşterilerin düşüncelerini kavramadır. Big data ile müşterinin satin alma yolcuğunu uçtan uca takip ederek müşteri eğilimlerini gözlemlemek, kampanyaları sunmak ve bütçelerini optimize etmek için gereklilik haline gelmekte olduğunu belirtmiştir.

Christian (2005) çalışmasında; FP-Growth Algoritması ile Apriori, Eclat ve Relim algoritmalarını karşılaştırılmıştır. C programlama dili üzerinde yapılan bu çalışmada FP-Growth algoritmasının çalışmadaki diğer algoritmalara göre daha iyi bir performas gösterdiği sonucuna varılmıştır.

Kumar ve Rukmani (2010) "Implementation of Web Usage Mining Using" adlı çalışmalarında, web madenciliği üzerinde Apriori ve FP-Growth algoritmalarının bellek ve zaman kullanımlarını karşılaştırmışlardır. Apriori algoritmasının temel dezavantajının, özellikle büyük ve uzun desenlerde aday kümesi oluşturmasının maliyetli oluşu, FP-Growth algoritmasının temel dez avantajının ise iyi bir aday oluşturma yönteminden yoksun oluşu şeklinde sonuç çıkarmışlardır.

Huang Yuan (2009), Carma algoritmasını Apriori algoritması ile karşılaştırmış ve verimliliğini değerlendirmiştir. Carma algoritması ile Apriori algoritmasının aynı destek eşik değerinde aynı sonuçları verdiği gözlenmiştir. Aynı zamanda Carma algoritması sonucunda üretilen veri gruplarının Apriori algoritması sonucunda üretilen veri gruplarının alt kümesi olduğu sonucuna varmışlardır.

Sherdiwala ve Khanna (2015), birliktelik kuralları algoritmalarında verimliliği artırmak için veri tabanında ki verilerinizin üzerinden geçme sayınızın azaltılması, veritabanınızı örneklemeniz, eksta kısıtlar konulması ve parallelleştirmenin gerektiğini belirtmişlerdir.

Lin (2009), yaptığı çalışmada, tedarikçi seçimi problemi için FP-Growth algoritmasını kullanmıştır. Tedarikçi seçimi tedarik zinciri yönetimi için önemli problemlerden biridir. Çalışmanın amacı esas ve yedek tedarikçiler seçimindeki karmaşıklığı indirgemektir. FP-Growth algoritmasının genellikle çalışma süresi ve verimliği ile ilgili çalışmaların yanı sıra operasyon yönetimi ve tedarik zinciri alanlarında sıklıkla kullanılmaya başlandığını ve tedarikçi elimine edilmesinde ve karmaşıklık azaltması etkili olduğu belirlenmiştir.

Mostafaei, Ganjavi ve Ghodsi (2018), "New Approaches to Analyze Gasoline Rationing" adlı çalışmasında 2005 ile 2011 yılları arasındaki karayolu taşımacılığı sektöründe meydana gelen değişiklikleri araştırmışlardır. Apriori ve Carma algoritmaları kullanılarak benzin tüketiminin durumu incelenmiştir. Benzin tüketiminin inceleme sonucunda doğalgaz, toplu taşıma ve metro kullanımı ile ilişkili olduğu belirlenmiştir. Apriori ve Carma algoritmalarından

elde edilen sonuçları karşılaştırılmış ve Carma algoritmasından elde edilen sonuçların Apriori algoritmasından elde edilen sonuçlar ile aynı olduğu gözlemlenmiştir

Mayilvaganan ve Kalpanadevi (2018), yaptıkları çalışmada Apriori, FP-Tree ve Fuzzy FP-Tree algoritmalarını karşılaştırmıştır. Çalışma sonucunda Fuzzy FP-Tree algoritmasının zaman açısından Apriori ve FP-Tree algoritmalarından daha verimli olduğu ortaya çıkmıştır.

Said, Dominic ve Abdullah (2009), yaptıkları çalışmada FP-Growth algoritması ve bu algoritmanın çeşitli varyasyonlarının yeterlilik ve etkililiklerini zaman ve bellek tüketimi açısından karşılaştırmışlar. Sonuç olarak ise her durum için tek bir algoritmanın olmadığı ve başarılı bir algoritmanın ana sınırlayıcısının verinin istatistik zenginliğinin olduğu sonucunu elde etmişlerdir.

Hunyadi (2011), yapmış olduğu çalışmasında Apriori ve Fp-Growth algoritmalarını kıyaslamış ve Apriori algoritmasının, sık görülmeyenleri eleme şeklinde, işleyişinin Fp-Growth ve Creative Association Rules ‘e göre farklı olmasına rağmen ürettiği sonucun diğer iki algoritma ile aynı olduğunu ve çalışmasında görselini ilettiği regresyon doğrusu boyunca açık bir şekilde görüldüğünü ifade etmiştir.

Fp-Growth Algoritmasının çok daha hızlı ve etkili bir şekilde çalışması için birçok araştırma yapılmış ve hem algoritma , hem veri tabanı hem de çeşitli donanım öğeleri eklenerek bu araştırmalar desteklenmiş ve sonuç alınmıştır (Pramudiono & Kitsuregawa, 2003) (Zhou, ve diğerleri, IEEE Youth Conference on Information) (Racz, 2004) (Zeng, Yin, Liu, & Zhang, 2015) (Özdoğan, Abul, & Yazıcı, 2009).

Al-Maolegi ve Arkok (2014), veri madenciliği algoritmalarından en populer algoritmalardan biri olan Apriori algoritmasının sınırlayıcı özelliği olan tüm veri tabanının tamamını tararken harcadığı zaman kaybı ile ilgili bir çalışma yaparak yeni bir algoritma türetniş ve kendi türettikleri algoritmanın orjinal Apriori algoritmasına oranla %67.38 daha az zaman harcadığını tespit etmişlerdir.

Yuan (2017), geleneksel Apriori algoritmaları ile ilgili çeşitli çalışmalar yaptıktan sonra bu algoritmaların iki büyük darboğazı olduğunu ortaya çıkarmıştır. Bunlardan biri veri tabanını sık sık taraması diğeri ise çok sayıda aday küme üretmesi. Apriori'nin doğasında var olan kusurlara dayanarak çalışma kapsamında türettiği algoritmasıyla ilgili;

1) defalarca veri tabanını taramaktan kaçınmak için yeni veri tabanı haritalama yöntemi kullanmak; 2) birleştirme verimliliğini artırmak için sık öğe kümelerini ve aday öğe kümelerini

daha fazla budamak; 3)yüksek verimlilik elde etme adına destek değeri sayma işleminde örtüşme(overlap) stratejisi kullanmak şeklinde çözümler üretmiş ve aynı koşullar altında, Yuan'ın Apriori algoritması, diğer geliştirilmiş algoritmalara kıyasla işletim verimliliğini arttırmıştır.

Singh, Ram ve Sodhi (2013), Apriori Algoritması adına yaptıkları çalışmada, veri madenciliğinde sık kullanılan bu algoritmanın veri tabanını çok fazla taraması nedeniyle verimsiz olduğu ve büyük veri tabanlarında ise bu taramadan dolayı çok fazla zaman kaybettiğini vurgulayarak yeni bir algoritma geliştirmiş ve tarama esnasında gereksiz olan transactionları keserek aslında direk sık görülen kümeye yaklaşmayı sağlıyor ve tarama miktarını düşürüyor. Bu nedenle klasik Apriori algoritmasına göre çok daha hızlı çalışan bir algoritma türettiler.

Fp-Growth algoritmasında olduğu gibi Apriori algoritmasında da, hızlandırmak ve verimliliği arttırmak amacı ile birçok çalışma yapılmıştır (Orlando, Palmerini, & Perego, Enhancing the Apriori Algorithm for Frequent, 2001) (Ye & Chiang, 2006) (Li, Zeng, He, & Shi, 2012) (Agarwal, Yadav, & Anand, 2013) (Chai, Yang, & Cheng, 2007) (Yu, Wang, Wang, & Chen, 2008) (Xie, Li, Wang, & Lu, 2008) (Changsheng, Zhongyue, & Dongsong, 2009) (Chang & Liu, 2011) (Savasere, Omiecinski, & Navathe, 1995) (Jiang, ve diğerleri, 2018).

Heaton (2016) çalışmasında Fp-Growth, Eclat ve Apriori algoritmalarının performanslarını, mevcut çalışmalarda olduğu gibi aynı data seti üzerinden değil benzer boyutlarda fakat farklı data setleri üzerinde incelemiştir. Çalışma farklı veri setlerinin performansı nasıl etkileyeceğini araştırmıştır. Araştırma sonucunda Fp-Growth ve Eclat algoritmalarının daha performanslı, Apriori'nin ise çok daha az performanslı çalıştığı tespit edilmiştir (Heaton, 2016).

Hossain, Sattar ve Paul (2019), yaptıkları çalışmada Market Sepeti Analiz çalışmalarında en sık kullanılan algoritmalardan FP-Growth ve Aprori algoritmalarının ön tanımında verilen minimum destek (support) değerinin çok düşük verildiğinde çok fazla hesaplama gerektiren büyük aday kümelerinin türetildiğinden bahsetmiş ve farklı yüzdesel oranlarda en çok satan ürünlerden oluşan kümeyi bu iki algoritmaya vermişler. Sonuçlar gösteriyor ki tüm listeye nazaran en çok satan listenin alınması çok daha kısa sürede aynı birliktelik kuralları ile aynı sıklık kümesinin türetildiğini göstermiş aynı zamanda FP-Growth algoritmasının Apriori algoritmasından daha performanslı çalışmıştır.

Sağın ve Ayvaz (2018), yaptıkları çalışmada Market Sepeti Analizi, Apriori ve Fp-Growth algoritmalarını kullanarak Perakende sektöründe gerçek veriler ile çalışmış ve iki Algoritmanın

WEKA program üzerinde verdiği sonuçları kıyaslayarak hem iki algoritmanın hızını kıyaslamış hem de verilerin analizinden ortaya çıkarılan birlikte satılabilecek ürünleri listelemişlerdir.

Kaur ve Kang (2016) çalışmalarında, mevcut veri madenciliği algoritmalarının static veriler üzerinde çalıştığını belirterek, çalışmalarında ortaya koydukları algoritmanın dinamik verileri de ele aldığını belirtmiş ve çalışmalarının müşteri davranışlarını deneyimlemeye ve geliri arttırmaya yardımcı olduğunu belirtmişlerdir (Kaur & Kang).

Chen ve arkadaşları (2004), müşteri satışlarını inceleyerek satış , pazarlama , servis ve operasyonel stratejiler ve daha da artan bir çok alanda satıcıya fayda sağlayan market sepeti analizi ile ilgili bir çalışma yapmışlardır. Çalışmalarında , birden fazla zinciri olan mağazaları ele almış ve mevcut araştırmaların, başarısız olma ihtimali olan ürünlerin tüm mağazalarda raflarda tutulduğu varsayımı olacağını ortaya koyarak , kendilerine bu soruna yeni bir yöntem ortaya koymuşlar ve bu yöntemde veri oluştururken mağaza ve dönem sayısı , mağaza boyutu ve ürün değiştirme oranlarını dikkate almış ve chain-store (mağaza zinciri) adını verdikleri bu yaklaşım sonuç olarak mevcutlarına oranla daha efektif bir sonuç ortaya koymuştur

Aguinis, Forcum ve Joo (2012), market sepeti analizinin sadece marketler için değil ilişki kuralları türetmek için ortaya konduğunu ve artık birçok farklı alanda da kullanıldığını belirtmiş ve çalışmalarında ise insan kaynakları yönetiminde bu tekniği kullanmışlardır.

Raorane ve arkadaşları (2012), veri madenciliği araçlarının, büyük miktarda veriyi analiz etmek için en emin silah haline geldiğini ve doğru kararlar vermede büyük bir atılım olduğunu belitmiş ve bu çalışmalarında, büyük miktarda veriyi analiz ederek tüketici davranışı ve rakiplere karşı rekabet üstünlüğüne yol açan doğru kararı vermeyi hedeflemişler ve deneysel çalışmalarını market sepeti analizi ile tamamlamış ve sonuca ulaşmışlardır (A., V., & D., 2012).

Yun, Chuang ve Chen (2006) çalışmalarında, çok mağazalı ortamlarda geleneksel birliktelik kuralları madenciliğinin sağlıklı sonuçlar ortaya koymadığını düşünmüşlerdir. Bu düşüncelerini mağazalardaki birliktelik kurallarını inceleyebilecekleri yeni bir algoritma algoritma geliştirilmiştir. Geliştirilen algoritma ile üretilen kurallar zaman periyodu ve mağaza konumu hakkında da bilgi vermektedir. Boyut bakımından farklılık gösteren ve sürekli değişen ürün portföyüne sahip mağazalarda yapılan çalışmaya göre sunulan bu yöntemin, geleneksel Apriori algoritmasına göre daha iyi sonuç verdiği görülmüştür.

Erdem ve Özdağoğlu (2008), ege bölgesinde ki bir hastahanenin acil servisine gelen hastalar ile ilgili birliktelik kuralı analizi yaparak 65 sonuç elde etmişlerdir. Bu analizi yaparken Apriori algoritması kullanmış ve yaş ve cinsiyet gruplarına göre acil servise başvuru zamanları ve kalış sürelerine ilişkin anlamlı bilgiler elde edilmiştir.

Literatür taramalarında görüldüğü üzere özetle Market sepeti analizinde özellikle Apriori ve Fp-Growth algoritmaları kullanılmıştır. Eclat gibi algoritmalar ile de çalışıldığı görülmüştür. Market sepeti analizinin sadece market ürünleri için kullanılmadığı, sağlık, insan kaynakları yönetimi, tedarik zinciri, petrol, mağaza zincirleri, trafik, eğitim vs birçok alanda kullanıldığını göstermektedir. Aynı zamanda veri boyutlarının büyümesi ve büyük veriler arasındaki ilişkilerin kurulması için algoritmaların yetersiz kaldığı durumlar için ise birçok çalışmada kodların hızlandırılması için yöntemler geliştirilmiştir.

Literatür taraması özet tablosu aşağıda Tablo-1 'deki gibidir.

Yazarlar	Yıl	Açıklama	Yöntem
Agrawal, Rakesh, Imielinski, Tomasz ve Swami, Arun	1993	Mining Association Rules between Sets of Items in Large Databases	Geliştirme
Mythili, M.S. ve Shanavas, A.R. Mohamed	2013	Performance Evaluation of Apriori and FP-Growth Algorithms	Apriori ve Fp-Growth Algoritması
Christian, Borgelt	2005	An Implementation Of The Fp-Growth Algorithm	Fp-Growth Algoritması
Kumar, B Santhosh ve Rukmani, K V	2010	Implementation of Web Usage Mining Using	Web Madenciliği
Huang Yuan, Wang Xing, Shıa Ben- Chang	2009	Efficiency And Consistency Study On Carma	Carma Algoritması
Sherdiwala, Kainaz B. ve Khanna, Samrat O.	2015	Association Rule Mining: An Overview	Veri Madenciliği

Sağın, Ayşe Nur ve Ayvaz, Berk	2018	Determination of Association Rules with Market Basket Analysis: An Application in the Retail Sector.	Apriori ve Fp-Growth Algorithms
Lin, R-H.	2009	Potential use of FP-growth algorithm for identifying competitive suppliers in SCM	Fp-Growth Algoritması
Mostafaei, S, Ganjavi, H Shakouri ve Ghodsi, R.	2018	New Approaches to Analyze Gasoline Rationing	Apriori ve Carma Algoritmaları
Mayilvaganan, M ve Kalpanadevi, D	2018	COMPARISON OF APRIORI, FP-TREE GROWTH AND FUZZY FP-TREE	Apriori, Fp-Growth ve Fuzzy Fp-Growth Algoritması karşılaştırması
Said, Aiman Moyaid, Dominic, D ve Abdullah, Azween B	2009	A Comparative Study of FP-growth Variations	Fp-Growth Algoritması
HUNYADI, DANIEL	2011	Performance comparison of Apriori and FP-Growth algorithms	Apriori ve Fp-Grwth performans karşılaştırma
Pramudiono, Iko ve Kitsuregawa, Masaru	2003	Parallel FP-Growth on PC Cluster	Fp-Growth Algortiması
Zhou, Le ve diğerleri	2004	Balanced parallel FP-Growth with MapReduce	Fp-Growth Algortiması
Racz, Balazs	2004	An FP-Growth Variation without Rebuilding the FP-Tree	Fp-Growth Algortiması
Zeng, Yi ve diğerleri	2015	Research of Improved FP-Growth Algorithm in Association Rules Mining	Fp-Growth Algortiması
Özdoğan, Gülistan Özdemir, Abul, Osman ve Yazıcı, Ali	2009	Paralel Veri Madenciliği Algoritmaları	Fp-Growth– Parallel Programlama

Al-Maolegi, Mohammed ve Arkok, Bassam	2014	AN IMPROVED APRIORI ALGORITHM FOR ASSOCIATION RULES	Apriori Algoritması
Yuan, Xiuli	2017	An Improved Apriori Algorithm for Mining Association	Apriori Algoritması
Singh, Jaishree, Ram, Hari ve Sodhi, Dr. J.S.	2013	International Journal of Scientific and Research Publications	Apriori Algoritması
Orlando, Salvatore, Palmerini, P. ve Perego, Raffaele	2001	Enhancing the Apriori Algorithm for Frequent	Apriori Algoritması
Ye, Yanbin ve Chiang, Chia-Chu	2006	A Parallel Apriori Algorithm for Frequent Itemsets Mining	Apriori Algoritması
Li, N. Ve diğerleri	2012	Parallel Implementation of Apriori Algorithm Based on MapReduce	Apriori Algoritması
Agarwal, Pragya, Yadav, Madan Lal ve Anand, Nupur	2013	Study on Apriori Algorithm and its Application in	Apriori Algoritması
Chai, Sheng, Yang, Jia ve Cheng, Yang	2007	The Research of Improved Apriori	Apriori Algoritması
Yu, Wanjun ve diğerleri	2008	The research of improved apriori algorithm for mining association rules	Apriori Algoritması
Xie, Yiwu ve diğerleri	2008	The Optimization and Improvement of the Apriori Algorithm	Apriori Algoritması
Changsheng, Zhang, Zhongyue, Li ve Dongsong, Zheng	2009	An Improved Algorithm for Apriori.	Apriori Algoritması
Chang, Rui ve Liu, Zhiyi	2011	An improved apriori algorithm	Apriori Algoritması

Savasere, A., Omiecinski, E. ve Navathe, S.	1995	An Efficient Algorithm for Mining Association Rules in Large Databases.	Fp-Growth Algortiması
Jiang, Yu ve dğerleri	2018	A parallel FP-growth algorithm on World Ocean Atlas data with multi-core CPU	Fp-Growth Algortiması
Heaton, Jeff	2016	Comparing Dataset Characteristics that Favor the Apriori, Eclat or FP-Growth Frequent Itemset Mining Algorithms	Fp-Growth Apriori ve Eclat Algortiması
Hossain, Maliha, Sattar, A H M Sarowar ve Paul, Mahit Kumar	2019	Market Basket Analysis Using Apriori and FP Growth Algorithm.	Fp-Growth ve Apriori Algortiması
Kaur, Manpreet ve Kang, Shivani	2016	Market Basket Analysis: Identify the changing trends of market data using association rule mining.	Algoritma Geliştirme
Chen, Yen Liang ve diğerleri	2004	Market basket analysis in a multiple store environment	Algoritma Geliştirme
Aguinis, Herman, Forcum, Lura E. ve Joo, Harry	2012	Using Market Basket Analysis in Management Research.	Veri analizi
A., Raorane A., V., Kulkarni R. ve D., Jitkar B	2012	Association Rule – Extracting Knowledge Using Market Basket Analysis.	Market Sepeti Analizi
Yun, Ching-Huang, Chuang, Kun-Ta ve Chen, Ming-Syan	2006	An Efficient Clustering Algorithm for Market Basket Data Based on Small Large Ratios.	Algoritma Geliştirme
Erdem, Sabri ve ÖZDAĞOĞLU, Güzin	2008	ANALYZING OF EMERGENCY DATA OF A TRAINING AND RESEARCH	Apriori Algoritması

Han, Jiawei, Kamber, Micheline ve Pei, Jian.	2011	Data Mining: Concepts and Techniques 3rd Edition. s.l.: Morgan Kaufmann	Apriori Algoritması
Orlando, Salvatore, Perego, Raffaele ve Silvestri, Claudio.	2004	A new algorithm for gap constrained sequence mining	Algoritma Geliştirme

Tablo 1 *Literatür Taraması Özeti*

3. VERİ, VERİ TABANI VE VERİ AMBARI

3.1.Veri

Ölçüm, sayım, deney, gözlem ya da araştırma ile elde edilen işlenmemiş bilgi parçacığına veri denilmektedir (Wikipedia, 2020). Tanımda da belirtildiği gibi işlenmemiş ve tek başına bir anlam ifade etmeyen fakat bilgiye dönüşebilmesi için işlenip, işe yarar ve anlamlı hale getirilmesi gereken bilginin yapı taşı diye adlandırabiliriz. Aynı şekilde tanımdan bilginin ise işlenmiş verilerden çıkarılan sonuç olarak görülmesi sonu da çıkmaktadır.

Günümüzde depolamanın ucuzlayarak artması da verinin ve veri biliminin değerini arttırmaktadır. Tanımında ifade edilen işlenmemiş bir halde tutulan ve içeriğinde maden yatan bu verileri anlamlı hale getirmek öneminin her geçen gün arttığı bir konu olmaktadır.

Birçok bilim insanı ham veriyi işleyerek, ilişkilendirerek, analiz ederek daha hızlı ve en anlamlı veriyi ortaya koymak amacı ile çalışmaktadırlar. Bu çalışmaların artması ve teknolojinin ilerlemesiyle ucuzlayan saklama alanları ile birlikte anlamsız olan verilerin işlenmesi ve bilgiye dönüşmesi kolaylaşmış ve birçok kurum bundan kar etmekte ve amacına uygun verileri saklamaya yönelik adımlar atmaktadır.

3.2.Veri Tabanı

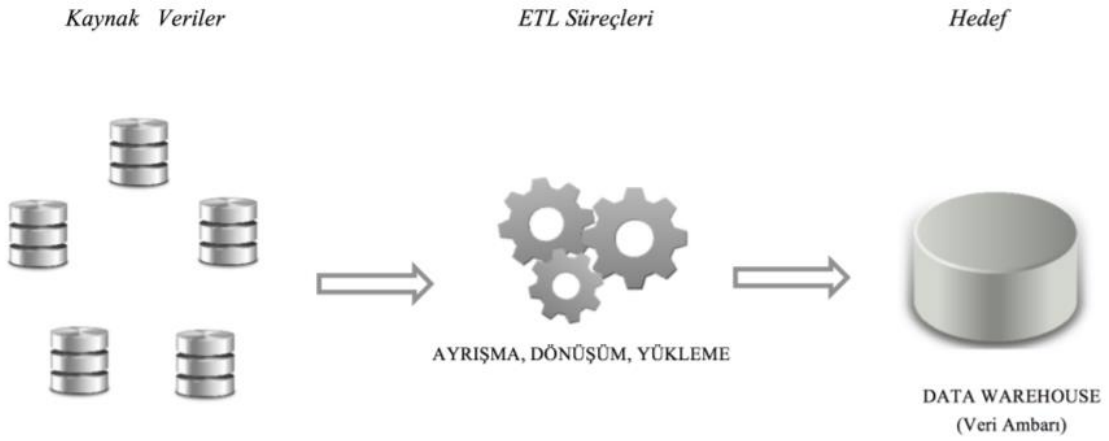
Verileri teknolojik ortamda düzenli bir formda saklayan yapılar olarak adlandırılmaktadırlar. Bir veri tabanı genellikle Veri tabanı Yönetim Sistemleri (VTYS) tarafından kontrol edilmektedirler. Veri tabanı çerisinde işe yarar verilerimizi saklamak, güncellemek, rapor elde etmek, verileri analiz etmek vs. birçok işlemi çok rahat ve hızlı bir şekilde yapmak mümkün.

Bu şekilde veriyi saklama, erişme ve analiz edebilme kolaylığı veri madenciliği süreçlerini desteklemektedir.

3.3.Veri Ambarı

Veri ambarı, ilişkili verilerin sorgulandığı ve analizlerin yapılabildiği bir depodur (Wikipedia, 2020). Veri ambarını veri tabanının belirli konulara göre özelleşmiş ve daha hızlı hareket kabiliyeti sağlayan bir parçası gibi düşünülebilir. Aynı zamanda veri ambarları veri tabanlarından veri akışı ile kendilerini güncelleyerek en güncel verilerin alınmasını sağlayabilmektedir.

Veri ambarlarına, veri tabanlarından çalışan bilgileri, maaş verileri, izin yönetimi, pazarlama, finansman verileri gibi anlık veriler yüklenir. Veri tabanlarından alınan verilerin veri ambarlarına aktarım esnasında ETL (Extract, Transform, Load) denilen ve literatürde çıkarım, dönüşüm ve yükleme süreçlerinden geçmektedir.



Şekil 1 *ETL Süreci (Eşmekaya & Şeker, 2017)*

Şekil 1'de görülen ETL sürecinin aşağıdaki şekilde 3 adımı bulunmaktadır:

Extract (Ayrışma, Çıkarım): Kaynak sistemden verilerin alınması.

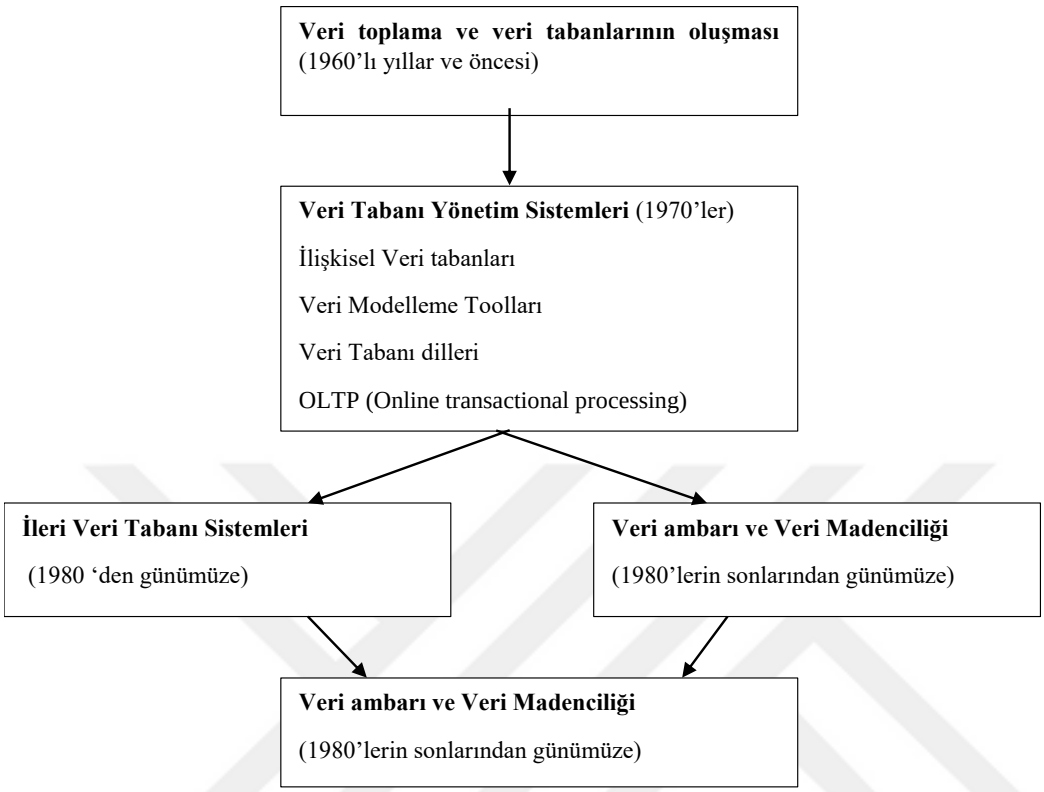
Transform (Dönüşüm): Farklı kaynak sistem veya sistemlerden alınan verilerin konuya özel istenen şekilde dönüştürme sürecidir.

Load (Yükleme): Dönüşümü sağlanmış ve istenen formattaki verilerin hedef sisteme yüklenmesidir. Burada bahsedilen hedef ilgili veri ambarıdır.

4. VERİ MADENCİLİĞİ

Günümüzde hızla artan teknoloji nedeniyle kişisel veya kişisel olmayan birçok veri doğmakta ve saklanmaktadır. Saklanan bu veri yığınları arasından insan eli ile anlamlı verilerin çıkması çok uzun ve zahmetli hatta neredeyse imkânsız bir hal alırken Veri Madenciliği kavramı ve teknikleri doğmuş ve bu büyük veri yığınlarından anlamlı veriler çok kısa sürede elde edilmeye başlanmıştır.

Veri madenciliği adının içinde yatan anlamda olduğu gibi bir maden içinden değerli şeyler elde etmek ile aynı manayı taşımakta ve şirketlerin, kurumların ve birçok farklı alanda hizmet veren her bir kuruluşun sakladığı ve aslında ilk bakışta anlam ifade etmeyen veri yığınlarının arasından çok kıymetli ve o kuruluşa yön verecek ve kazanç getirecek verilerin elde edilebildiği bir yapıdır. Veri madenciliğinin ortaya çıkmasının temelinin veri tabanlarının olduğu ve gelişimi Şekil-2 (Han & Kamber, Data Mining: Concepts and Techniques, 2000) de görülmektedir



Şekil-2 *Veri Madenciliği Tarihsel Gelişim Süreci (Han & Kamber, Data Mining: Concepts and Techniques, 2000)*

Veri tabanlarında verilerin saklanması kolaylığı veri anlamında çok zengin olmamızı sağlasa da bilgiye dönüşmediği sürece çokta anlam ifade etmeyen yığınlardan farksızdır. Yığınlar içerisinde belki de kurumları belli bir süre içerisinde inanması güç bir seviyeye taşıyabilecek bilgi bulunmakta fakat uygun bir süreçten geçirilmediği ve amaca uygun değerlendirilemediğinden potansiyeli ortaya konulamamaktadır.

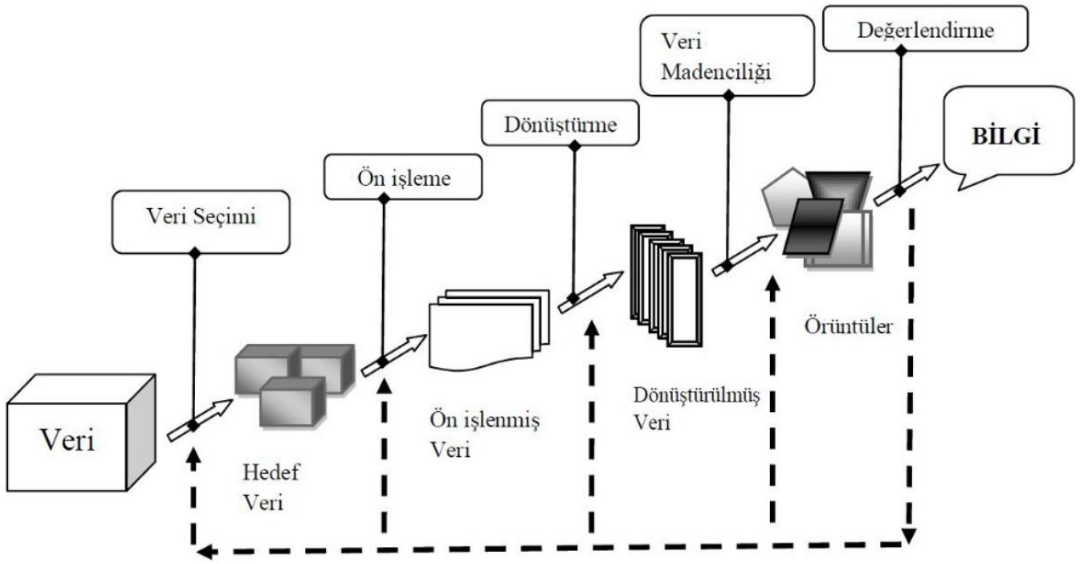
Literatür Taraması bölümünde de görüleceği üzere veri madenciliğinin birçok kullanım alanını bulunmaktadır. Eğitim, finans, sağlık, pazarlama, spor, telekomünikasyon, sigortacılık, borsa, mühendislik, sanayi gibi birçok alanda kullanılmaktadır.

Verileri saklamak, bir veri tabanına kaydetmek tabi ki işin en temel kısmı. Günümüzde büyük verileri saklanması için günümüzdeki cihazların depolama kapasiteleri her geçen gün arttırılırken eskiye nazaran temin etme ve fiyatının azalmasından dolayı elde etme oranı da artmaktadır. Fakat sadece bu verilerin saklanması yeterli değildir. Elimizdeki bu veri

madenlerini çeşitli işlemlerden geçirmek ancak onda var olan değerli madeni açığa çıkarmamızda yardımcı olacaktır.

4.1.Bilgi Keşfi Süreci ve CRISP-DM Modeli

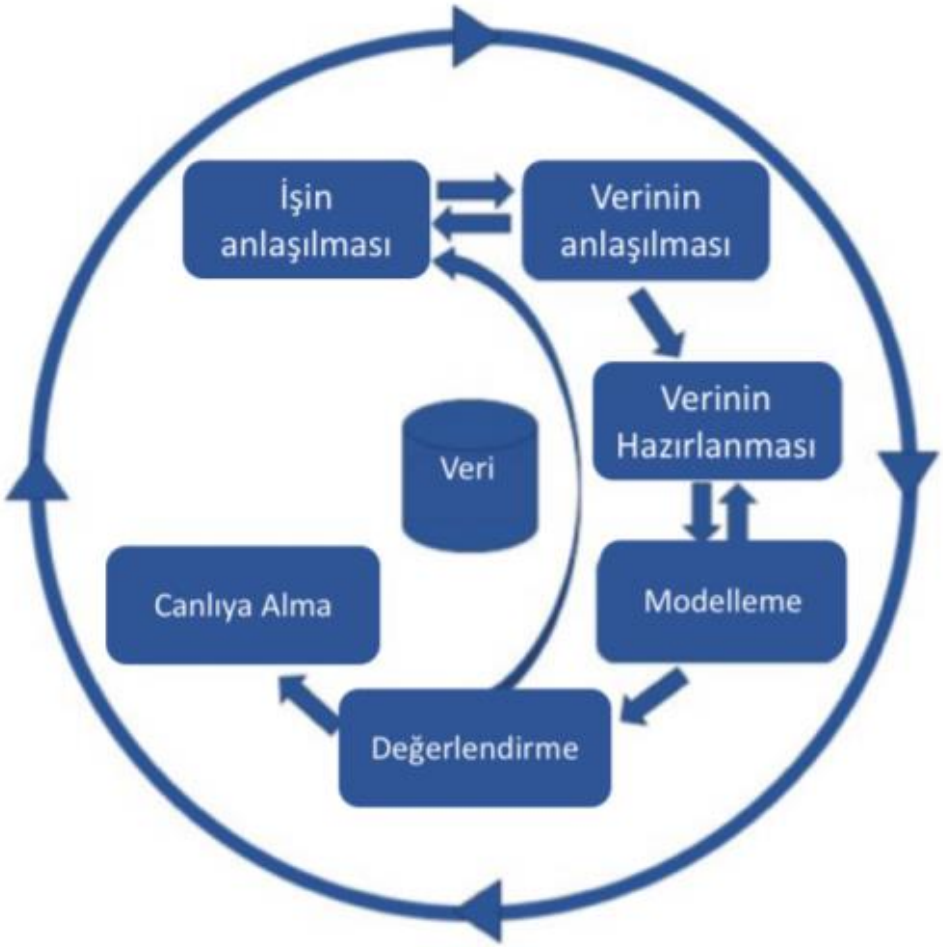
Bilgi keşfi süreci veri madenciliği sürecini de içerisinde barındıran ve en temel anlamda veri yığınlarından bilgiye erişme sürecindeki adımlar bütünü olarak tanımlanabilir. Süreç Şekil-3 (Albayrak, 2017) içerisinde görülmektedir.



Şekil-3 *Veri Madenciliği Bilgi Keşfi Süreci (Albayrak, 2017)*

Veri tabanlarından verinin elde edilmesi ile bilgi keşfi düreci başlamaktadır. Veri tabanlarından verilerin elde edilmesi sürecinin ardından bu veriler içerisinde bazı temizleme işlemleri yapılmaktadır. Bu rürece ise ön işleme süreci denmektedir. Ön işlenmiş veriler kurulan veri ambarları tarafından alınır ve ardından veri madenciliği süreçleri ilgili algoritmalar ile veri ambarından tabiri caizse madenden kazı çalışmalarına başlar. Ardından çıkan sonuçlar değerlendirilerek günümüzün neredeyse en önemli madeni olan bilgi elde edilir.

Veri madenciliği işleyişinin gerçekleşmesinde kullanılan en yaygın modellerden biri CRIPS-DM (Cross Industry Standart Process Model for Data Mining) modelidir.



Şekil-4 *CRISP-DM Adımları (Şeker, 2018)*

Şekil-4 'de belirtilen CRSP-DM süreçlerinin 6 temel adımı vardır :

İşin anlaşılması: İlk aşama var olan sorun somut bir şekilde tanımlanır, bu soruna sebep olan süreçteki tüm adımlar tespit edilir ve çözüm beklentileri tanımlanır.

Verinin anlaşılması: İş süreçleri tam olarak anlaşıldıktan sonra bu soruna uygun olan verilerin toplanması. Veri madenciliğinin temeli olan verilerin doğru seçilmesi ve hazırlanması sürecin tekrar başa dönülmesini engelleyecek dolayısı ile de zaman ve iş gücü kaybını en aza indireceğinden , ilk adımdaki problemin tanımı ile verilerin uyumlu ve eksiksiz olduğunun tespiti bu adımda yapılmalıdır.

Modelleme ve Değerlendirme: Veri madenciliği penceresinden değerlendirdiğimiz bu aşamalardan modelleme süreci veri madenciliğine başvurduğumuz kısımdır. Doğru modelin tespitini yapaibilmek için bir çok modeli denemek , değerlendirmek ve en iyi modele erişebilmek için bu sürecin tekrarlanması ile mümkün olacaktır.

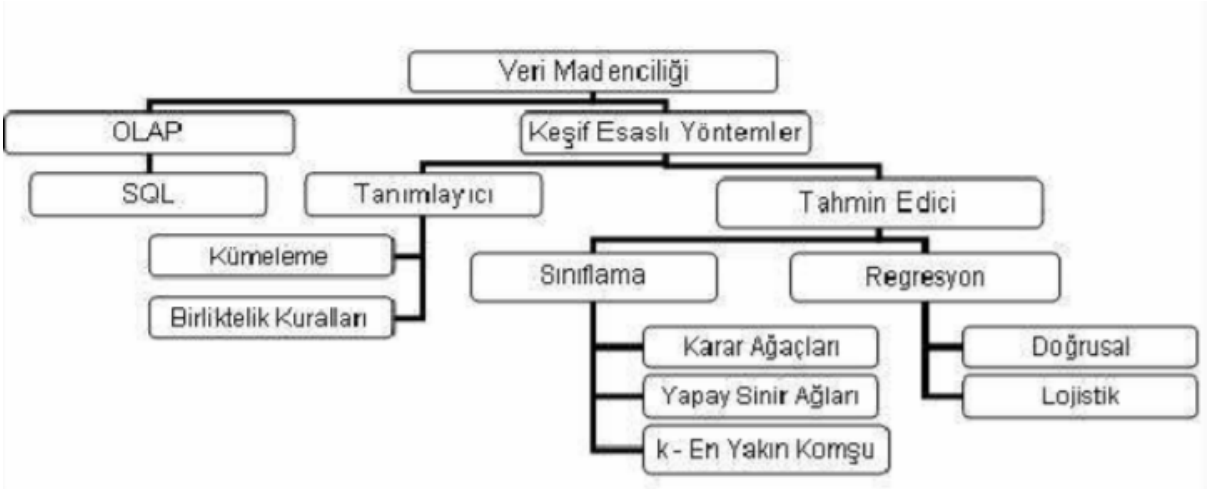
Canlıya alma: En uygun modeli tespit etmenin ardından artık bu modelin kullanılması sürecidir. Bu hayata geçirilen ve kullanılmaya başlanan model sürekli izlenip yeniden düzenleme gerekip gerekmediğine bakılmalıdır.

4.2.Veri Madenciliği Modelleri

Veri madenciliği modelleri tahmin edici (predictive) ve tanımlayıcı (descriptive) olmak üzere iki başlık altında toplanmıştır.

Tahmin edici model, daha önce deneyimlenen ve sonucu bilinen verilerden model üretme ve kurulan bu modelden yararlanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesi amaçlanmaktadır (Akpınar, 2000). Örneğin bir alışveriş sitesi önceki dönemlerde müşterilerine sattığı ürünlere ilişkin gerekli detaylara sahiptir. Bu detaylı verilerde bağımsız değişkenler sitede ürün satın alan müşterilerin özellikleri, bağımlı değişkenler ise sitede satılan ürünlerdir. Bu verilere uygun olarak kurulan model, daha sonraki alışverişlerde müşteri özelliklerine göre reklam verilebilecek veya kampanya yapılabilecek ürünlerin tahmininde kullanılmaktadır.

Tanımlayıcı model ise karar vermeye rehberlik etmede kullanılabilecek mevcut verilerdeki örüntülerin tanımlanması sağlanmaktadır (Akpınar, 2000).



Şekil-5 *Veri Madenciliği Modelleri (Şen, 2008)*

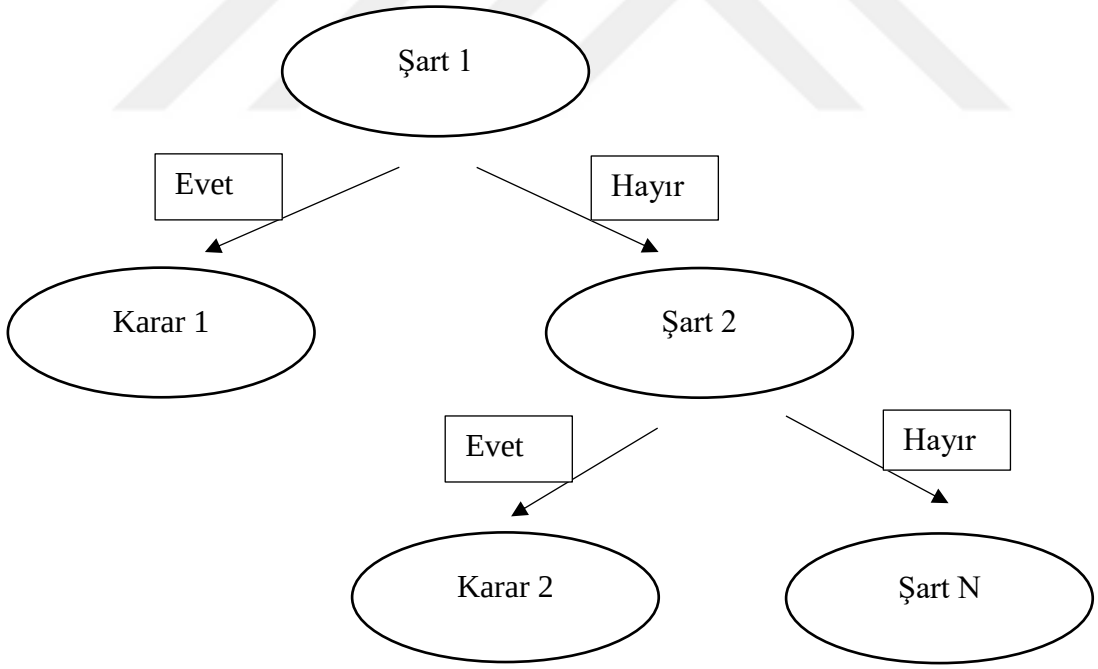
4.2.1. Sınıflandırma ve Regresyon

Tahmin edici (pedictive) veri madenciliği modellerinden sınıflandırma ve regresyon en yaygın kullanıma sahiptirler.

Sınıflandırma, bir nesnenin daha önceden incelenerek özellikleri belirlenmiş olan nesnelerin grubuna atanmasıdır. Sınıflama kategorik değerleri tahmin ederken, regresyon süreklilik gösteren değerlerin tahmin edilmesinde kullanılır (Şen, 2008).

Sınıflandırmaya kredi başvurusu değerlendirme, kredi kartı harcamasının sahtekarlık olup olmadığına karar verme, hastalık teşhisi, ses tanıma, karakter tanıma, gazete haberlerini konularına göre ayırma, kullanıcı davranışları belirleme gibi örnekler verilebilir.

Karar ağaçları; sınıflandırma problemlerinin çözümünde en sık kullanılan çözümlerden biridir. Karar ağacı yöntemi için öncelikle sorunun nedenleri anlaşılır, ardından bir ağaç yapısı kurulup ağacın yaprakları ilgili sınıfları ifade etmekte ve uygun sorular ile sonuca ulaşılmaktadır.



Şekil-6 *Karar Ağacı Yapısı*

Regresyon, aralarında ilişki olan ve bağımlı, diğerleri bağımsız Değişken(ler) olarak ele alınan ilişkinin fonksiyonel (matematiksel) eşitlik ile ifade edilmesi sürecidir (Akbulut, 2020).

Regresyon, hedef verilerin bilinen bazı fonksiyon tiplerine (Örneğin; doğrusal, lojistik vb.) uygun olduğunu varsayar ve daha sonra veri modellerine en uygun fonksiyonun hangisi olduğu belirler (Çelik, 2019).

4.2.2. Birliktelik Kuralları ve Ardışık zamanlı Örüntüler

Bir alışveriş esnasında veya birbirini izleyen alışverişlerde müşterinin hangi ürün veya hizmetleri satın almaya eğilimli olduğunun belirlenmesi, müşteriye daha fazla ürünün satılmasını sağlama yollarından biridir (Akpınar, 2000). Sadece alışveriş değil tıp, finans ve eğitim gibi farklı disiplinlerde de birliktelikleri analiz etmek ve önemli kazanımlar elde etmeye olanak sağlamaktadır.

Birliktelik kuralına aşağıdaki durumlar örnek olarak verilebilir.

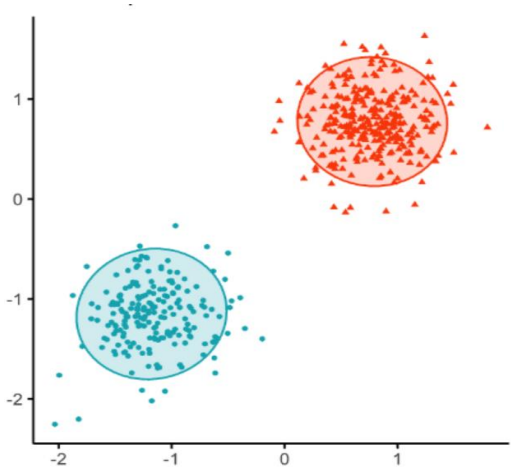
- Bebek bezi alanlar %50 ihtimalle ıslak mendil alırlar,
- Patates cipsi alanların %80 i ketçap alırlar.

Ardışık zamanlı örüntüler birbiri ile ilişkisi olan ancak birbirlerini izleyen dönemlerde gerçekleşen ilişkilerin tanımlanmasında kullanılır. Aşağıdaki örnekler bu modele örnek verilebilir.

- A ilacı verilen bir kanser hastası 7 gün içerisinde %40 ihtimalle B semptomlarını görecektir,
- X firmasının hisselerinin %15 in altına düşmesinin ardından, 5 iş günü içerisinde Y hissesinin değeri %35 oranında artar.

4.2.3. Kümeleme

Kümeleme yönteminin sınıflama yönteminden ayrılan en önemli özelliği sınıflama yönteminde olduğu gibi daha önceden bilinen ve deneyimlenen bir grubun olmamasıdır. Verilerin kümelenmesi diğer verilere olan yakınlığından-benzerliğinden kaynaklanır. Kümeleme analizini aslında büyük bir kümeyi küçük alt kümelere ayırma işlemi olarak da düşünebiliriz.



Şekil-7 *Küme Analizi*

Veri madenciliğinde kümeleme analizinin tipik gereksinimleri aşağıdaki şekilde sıralanmaktadır (Han J. P., (2011))

- Ölçeklenebilir olmalıdır öyle ki küçük veriler ile de çok büyük veriler ile de çalışabilmelidir
- Farklı veri tipleri ile de çalışabilmelidir. Günümüzde birçok veri tipi mevcuttur fakat genelde numerik kümeleme algoritmaları yazılmaktadır
- Rastgele şekle sahip kümeleri de keşfetmelidir
- Gürültü içeren veriler ile de kullanılmaya uygun olmalıdır.
- Çok boyutlu veri tabanlarında da kullanılabilmelidir
- Veri kümesinin sahip olduğu kısıtları dikkate alabilen yapıda olmalıdır.
- Rahat yorum yapılabilir sonuçlar ortaya koyabilmeli ve fonksiyonel olmalıdır.

5. BİRLİKTELİK KURALLARI TEMEL KAVRAMLARI & MARKET SEPETİ ANALİZİ

Birliktelik kuralları, mevcut veriler kullanılarak verilerin analiz edildiği ve ileriye yönelik kullanılmak üzere veriler arasında bir birliktelik bulgusu var ise olasılık yapılarıyla ortaya koyan yaklaşımdır.

90'lı yıllarda saklanan verilere artık müşterilerin ve ürünlerin verileri eklenmeye başlandı. 1993 yılında Agrawal ve arkadaşları tarafından geliştirilen algoritmanın da etkisi ve büyük ölçüde sağladığı kolaylık sayesinde büyük verilerin aralarındaki ilişki kurulmaya başlanmış ve artık veri tabanında yapılan madenciliğin ticari etkileri daha görünür olmaya başlamıştır.

Birliktelik kuralı en basit anlamda, bir X verisi ise bir Y verisi arasındaki bağı olasılık olarak ifade eden kurallar bütünüdür. X => Y şeklinde ifade edildiğinde "X olduğu durumda Y'nin olma olasılığı" şeklinde okunabilir.

5.1.Birliktelik Kuralının Matematiksel Gösterimi

Birliktelik kurallarınınn (kural gösterimi {X1, ..., Xn} => {Y1, …, Yn} temel alınarak) başlıca terimleri:

Öncül (Antecedent); kuralın solunda yer alan kümeyi ifade eder. Adından da anlaşıldığı gibi koşulun öncül yani ilk şartıdır ve öncül sağlamış ise diğer süreçler kontrol edilmeye devam edilir.

Ardıl (Consequent); kuralın sağında kalan kümeyi ifade eder. Öncül durum var olduysa ardıl (öncülün peşi sıra gelen) ifadesinden bahsedilebilir.

Minimum destek değeri (Minimum Support Value); öncül ve ardıl değerin birlikte bulunduğu kayıt sayısının tüm nesnenin kayıt sayısına oranını ifade eder. Algoritmalara bu değerin verilmesi veri setinin sadece istenen orandaki neslere erişmesini hızlandıracaktır.

P (X ve Y) = X ve Y'nin birlikte olduğu kayıt sayısı / Toplam kayıt sayısı

Minimum güven değeri (Minimum Confidence Value); öncül değer olması durumunda hangi olasılıkla ardıl değerin olduğu ifade eden olasılık değeridir. Minimum destek değerinde olduğu gibi bu değer için de algoritmaya istenen eşik belirtilerek hedef değer ve üzerindeki veri setine odaklanılır ve daha hızlı sonuç eldi edilmesini sağlar.

P(Y|X) = X ve Y'nin birlikte olduğu kayıt sayısı / X'in bulunduğu toplam kayıt sayısı

Birliktelik analizinin modeli;
VT: Veri Tabanı,
N = {n1, n2, n3, …, nm} nesneler kümesi,
D: işlemler kümesi,
VT veritabanında her işlem W, W ⊆ I olacak şekilde tanımlanan nesnelerin kümesi olsun.

W işlemler kümesinin alt kümeleri olan X ve Y nesneleri N nesneler kümesinden alınan elemanlardan oluşan birer küme olsun.
Bir birliktelik kuralı A ⇒ B formunda ifade edilir. A öncül ve B ardıl (izleyen) olarak adlandırılır.
Burada; A ⊂ I, B ⊂ I ve A ∩ B = ∅ dır.

Bü modeli bir market alışverişi ile eşleştirecek olursak, VT veri tabanında bulunan ürünler tablosu N yani nesneler kümesine eşdeğer olacaktır. D işlemler kümesinin içindeki her bir işlemi ifade eden W ise her bir alışveriş fişi olarak düşünülebilir. Bir W yani bir fiş içerisinde en fazla marketteki ürünlerin tamamı ya da daha az sayıda ürün alınmış olabilir. Bu her bir fiş olan W işlemlerinin içerisinde bulunan ürünlerden bazılarının alındığı durumda (A öncül kümesi), diğer ürün /ürün kümesinin (B ardıl kümesi) alınma olasılığının %c olarak belirtebiliriz.

Olasılık belirttiği ifade edilen birliktelik analizinde değerler 1 ile 0 arasında olacaktır. Yine ürünler üzerinden örneklendirilecek olursa M ürünü alındığında T ürününün alınma olasılığı en fazla %100 yani olasılık olarak 1 değerini gösterebilir. Algoritmaya bu değerin %50 den fazla olması gibi belirli kurallar verilerek istenen setin budanması kolaylaştırılmış olacaktır. Güven oranı ne kadar yüksek ise ürünlerin birlikteliinin o kadar güçlü olduğu söylenebilir.

5.1.1. Tek Boyutlu Birliktelik Kuralı

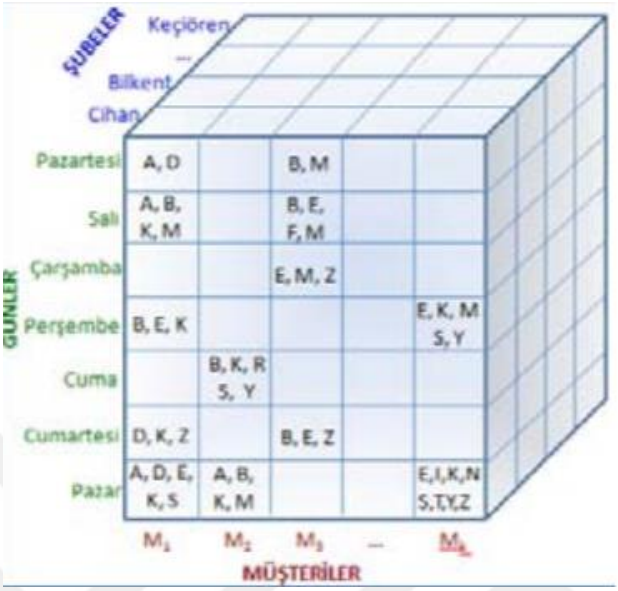
Birliktelik kurallarının boyutuna göre ayrılan parçasından biri olan tek boyutlu birliktelik kuralı iki nesne arasındaki ilişkiyi ifade etmektedir. Bir satın alma senaryosunda sadece ürünler arasındaki ilişkiden bahsediliyorsa bu tek boyutlu bir birliktelik analizidir.

Örneğin; X (Ürünü) => Y(ürünü) (%60), X ürününü alanların Y ürününü alma olasığı %60'dır şeklindeki birliktelik kuralı tek boyutlu bir birliktelik kuralıdır.

5.2.Çok Boyutlu Birliktelik Kuralı

Birliktelik kurallarının boyutuna göre ayrılan parçasından biri olan çok boyutlu birliktelik kuralı iki nesneden daha fazla nesne arasındaki ilişkiyi ifade etmektedir. Tek boyutlu birliktelik analizindeki sürece ek olarak satın alma senaryosuna ürünleri alan kişilerin cinsiyetleri, ürünlerin alındığı şube, ürünlerin alındığı mevsimler gibi çoğaltılabilen birden fazla nesne ile mukayese edilmesi süreci tek boyuttan çok boyuta çıkarmaktadır.

Çok boyutlu birliktelik kuralı analizleri OLAP küpü adı verilen çok boyutlu analiz aracılığı ile yapılmaktadır. Şekil-8 (Birant, ve diğerleri, 2010)'de OLAP küp görselinde müşterinin hangi günler ve hangi şubelerde hangi ürünleri aldığı bilgisi verilmektedir.

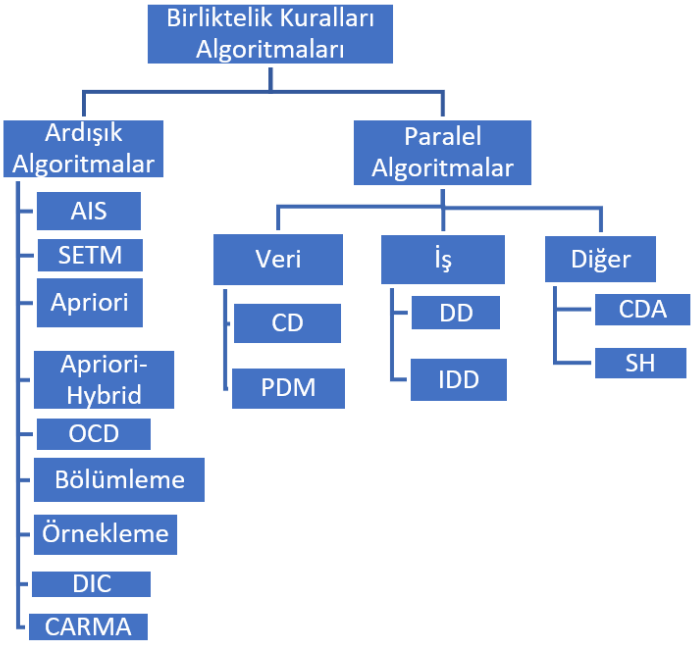


Şekil-8 *OLAP Küpleri Örneği (Birant, ve diğerleri, 2010)*

5.3.Birliktelik Algoritmaları

Birliktelik kurallarının öneminin artmasının ardından geliştirilen birçok algoritma olmuş ve her yeni geliştirilen algoritma gibi birliktelik algoritmaları da sektörün ihtiyaçları nispetinde güncellenerek gelişimini sürdürmeye devam etmiştir.

Çok sayıdaki birliktelik kuralları algoritmasını birbiriden ayırt edebilmek için yaklaşım tarzlarına göre bir sınıflandırılmış düzeni Şekil-9 (Gürgen, 2008) 'da belirtilmiştir.



***Şekil-9** Birliktelik Kuralı Algoritmaları (Gürgen, 2008)*

5.3.1. AIS Algoritması

AIS Algoritması, 1993 yılında geliştirilmiş, adını geliştiricileri olan Agrawal, Imielinski ve Swami isimlerinin baş harflerinden alan ve bir veri setinden yaygın nesne kümesi/kümeleri oluşturmak amacıyla geliştirilmiş bilinen ilk algoritmadır. İşlenecek olan veri isimlerinin ardışık olarak sıralanması şartını taşır. Dolayısı ile ardışık algoritmalar kategorisinde bulunmaktadır. Algoritma veri tabaınındaki fonksiyonların arttırılarak karar destek süreçlerini gerçekleştirmeye odaklanmaktadır.

AIS Algoritması, veritabanının tamamının üzerinden birden fazla kez geçer. İlk döngüde, ayrı ürünlerin destek değerlerini sayar ve bu sayede büyük veya sık geçen ürün kümelerine karar verir. Her döngü de sayılan destek değerleri ile karar verilen sık öğe kümesi ile aday küme listesi bir arttır. Bir işlem tarandıktan sonra, bir önceki işlem taramasında bulunan büyük ürün kümeleri ile bu işlemde bulunan ürünler arasındaki ortak ürün kümeleri belirlenir. Belirlenen ortak ürün kümeleri, yeni aday öğe kümeleri oluşturmak için işlemdeki diğer öğelerle genişletilir. Bu öğe kümesi alfabetik olarak sıralı bir şekilde tutulur. Bu sürecin etkili ve performaslı bir şekilde yürütülebilmesi için tahmin ve budama teknikleri kullanılır. Tahmin ve budama teknikleri, aday ürünlerden ister kümesininin dışında kalacak gereksiz olan ürün kümelerini çaıkararak aday ürün kümelerinin belirlenmesinde rol oynar. Ardından, her bir aday kümenin destek değerleri hesaplanır. Belirlenen minimum güven değerine eşit veya daha büyük güven değerine sahip aday kümeleri büyük ürün kümeleri olarak seçilir. Seçilen büyük

ürün kümeleri bir sonraki döngü için aday kümelerinin oluşturulması amacıyla arttırılır. Daha fazla ürün kümesi kalmayıncaya bu işlem sona erer (Dunham, Xiao, Gruenwald, & Hossain, 2000).

AIS Algoritmasında çok sayıda aday kümesinin oluşturulması tampon belleğin (buffer memory) gereğinden fazla dolmasına sebep olmaktadır. Bu sorunun çözülmesi için gerekli olan durumlarda kullanılmak üzere uygun bir tampon bellek yönetim şeması bulunması gerekmektedir. m üründen oluşan ve tüm ürünlerin işlemde kullanılması söz konusu olduğunda m cinsinden üstel bir karmaşıklığa sahip olduğunu göstermektedir. Birliktelik kuralları adına yayınlanan bu algoritmada var olan bu gibi dezavantajların üstesinden gelinebilmesi adına birçok çalışma ve kod geliştirme yapılmaıştır.

5.3.2. SETM Algoritması

SETM Algoritması, 1995 yılında Houtsmal tarafından SQL ile büyük veri kümelerini hesaplamak için kullanılmıştır. Diğer algoritmalardan farkı iki parametre ile hareket etmesidir. Nesne kümesindeki her bir eleman bu parametrelere sahiptir. Parametrelerden biri nesnenin ismi diğeri ise değeridir.

SETM algoritması, tıpkı AIS algoritmasında olduğu gibi verileri birçok kez tarar ve aynı süreçlerden geçiririr. AIS algoritmasından farkı SETM aday nesne kümelerinin TID'lerini de tutmasıdır. Aday nesne kümeleri alfabetik olarak sıralanır ve küçük nesne kümeleri budanır. Veritabanı bu şekilde birden fazla kez taranır ve eğer başka bir geniş nesne kümesi yok ise algoritma sonlandırılır.

TID bilgisinin de tutulması ve aday nesne kümesinin desteği hesaplanırken sıralanmamış olması SETM algoritmasnda nesne kümelerinin bir kez daha sıraya dizilmesi, algoritmada zaman ve yer karmaşıklığına neden olmaktadır.

5.3.3. Apriori Algoritması

Birliktelik kuralları madenciliği çalışmaları ilk kez 1993 yılında, Agrawal tarafından başlatılmıştır. Agrawal ve Srikant 1994 yılında, günümüzde en çok bilinen ve kullanılan algoritma olan, Apriori algoritmasını geliştirmişlerdir. Algoritma sık geçen nesnekümelerin bir önceki adımdan tahmin edilmesi üzerine kurulmuştur. Adı da önceki anlamındaki 'prior' kelimesinden gelmektedir (Agrawal, Imielinski, & Swami, 1993). Algoritmanın adı, sık kullanılan öğe kümesi özellikleri hakkında önceki (prior) bilgileri kullanmasına dayanmaktadır. Apriori, birliktelikleri keşfetmek için hazırlanmış yinelemeli bir yaklaşım

kullanmaktadır. İlk aşamada, kümede bulunan her bir öğenin toplam sayısını bulmak için veri tabanını tarar ve minimum desteği sağlayan öğeleri bir araya getirerek, bir öğe kümesi bulur. Elde edilen küme daha sonra alt kümeleri bulmak için kullanılır ve tüm birliktelikler bulunana kadar bu işlem devam eder. Her birliktelik alt kümesinin bulunması, veri tabanının bir tam taramasını gerektirir. Apriori algoritması bu taramalar sırasında verimliliği arttırmak ve arama alanını azaltmak için Apriori özelliği adı verilen bir hesaplama kullanır (Han, Kamber, & Pei, Data Mining: Concepts and Techniques 3rd Edition, 2011). Bu özelliğe göre, bir öğe kümesi yaygın bir öğe kümesi ise bu öğe kümesinin tüm alt kümeleri de yaygın öğe kümesi olmalıdır (Orlando, Perego, & Silvestri, A new algorithm for gap constrained sequence mining, 2004).

Apriori algoritması iki aşamalı bir süreçtir. İlk aşamada birleştirme ikinci aşamada budama işlemi gerçekleştirilir. Sık geçen k öğeli birliktelik kümeleri Lk, k aday altküme sayısı ve k öğeli aday birliktelik kümeleri Ck olarak adlandırılır. Birleştirme adımında Lk'yı bulmak için k öğeli her bir aday küme (Ck), Lk-1'in kendi arasında birleştirilmesiyle oluşturulur. Ck'nın elemanları bütün birliktelikleri içerir ve Lk bu kümenin alt kümesidir. Ck'daki aday sayısını belirlemek için bir veri tabanında taramayapılır ve Lk'lar hesaplanır. Ck sayısı çok büyük olduğu durumlarda çok yüklü hesaplamalar oluşacağından Ck sayısını azaltmak için budama aşaması gerçekleştirilir. Budama aşamasında, yaygın olmayan herhangi bir (k-1) aday kümesi bir yaygın aday kümesinin alt kümesi olamaz ve eğer bir aday k alt kümesinin herhangi bir (k-1) alt kümesi Lk-1'de yoksa aday yaygın aday kümesi olamayacağından ve Ck'dan kaldırılır. Bu sayede yaygın olmayan kurallar budanarak yaygın olan kuralların kalması sağlanır (Han, Kamber, & Pei, Data Mining: Concepts and Techniques 3rd Edition, 2011).

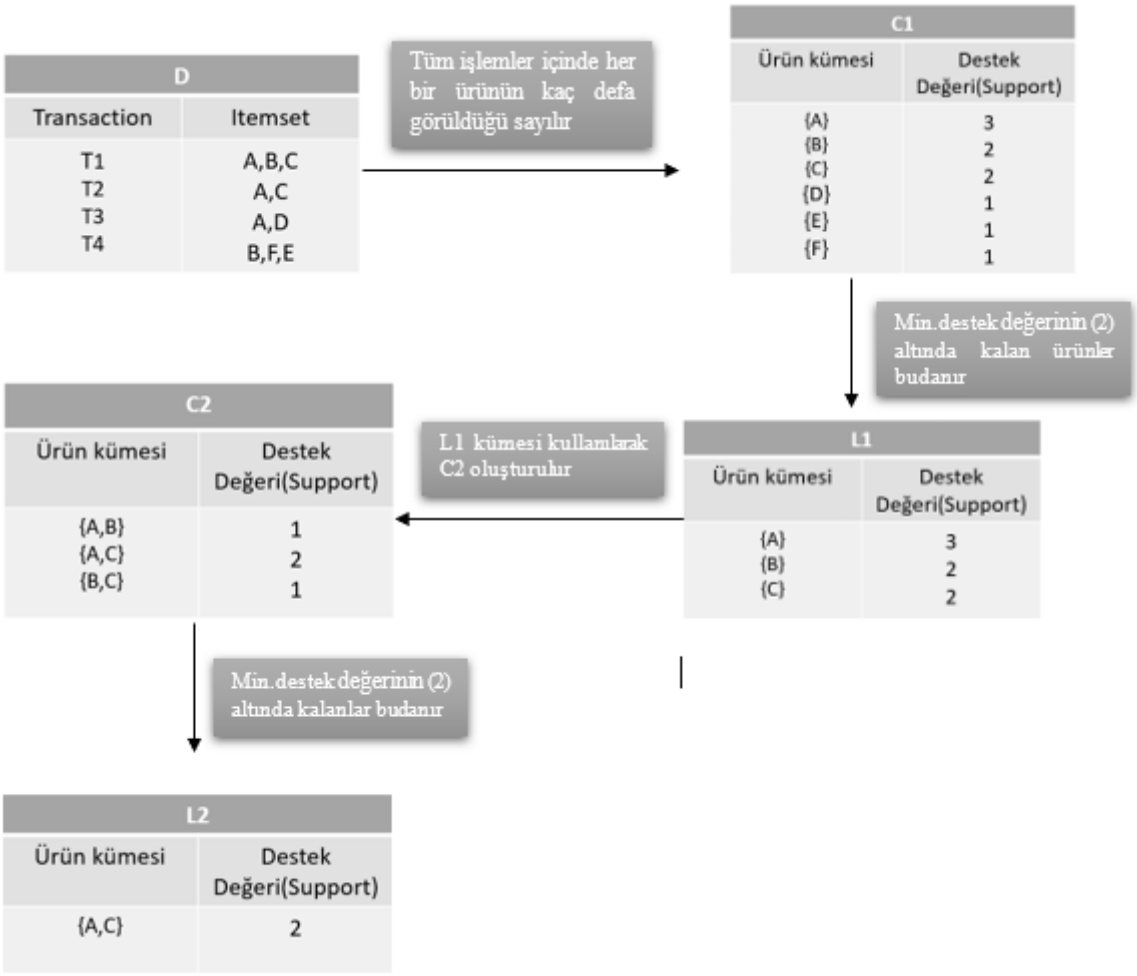
Apriori Algoritmasının pseudo (sözde) kodu Şekil – 10 (Çelik, 2019) 'daki gibidir.

```
1. L1 = bul {1 ögeli sık geçen öge kümeler};
2. için (k=2; Lk-1 ≠ ∅; k++ ) başla
3. Ck = apriori-gen (Lk-1); // yeni adaylar
4. tüm işlemler t ∈ D başla // D destek sayıları için taranır
5. Ct = alt küme (Ck, t); // Aday olan t' lerin alt kümeleri elde edilir
6. tüm adaylar c ∈ Ct olsun
7. c.sayısı ++;
8. son
9. Lk = { c ∈ Ck | c.sayısı ≥ min_s }
10. son
11. başa dön = UkLk;
Apriori_gen prosedürü (Lk-1: sık geçen (k-1)-öge kümeler)
1. tüm öge seti l1 ∈ Lk-1
2. tüm öge seti l2 ∈ Lk-1
3. Eğer (l1[1] = l2[1]) ∧ (l1[2] = l2[2]) ∧··· ∧ (l1[k-2] = l2[k-
2]) ∧ (l1[k-1] = l2[k-1])
ise {
4. c = l1∞ l2; // birleştirme adımı: aday oluşturma
5. Eğer sık olmayan alt kümeler c, Lk-1 varsa ise
6. sil c; // budama adımı: gereksiz adaylar çıkarılır
7. değilse c' yi Ck' ya ekle; }
8. başa dön Ck' ;
Sık olmayan alt kümeler prosedürü (c: aday k-öge set; Lk-1: yaygın (k-1)-
öge setleri);
// prior bilgisini kullan
1. tüm (k-1)-alt küme c' nin s' i
2. eğer s Lk-1 ise
3. başa dön TRUE;
4. başa dön FALSE;
```

Şekil-10 *Apriori Algoritmasının sözde(pseudo) kodları (Çelik, 2019)*

Apriori Algoritmasının çalışma mantığına örnek olması açısından aşağıdaki Şekil-11 (Apriori-Algorithm, 2020) şeması eklenmiştir. D veritabanında 4 adet alışverişin tutulduğu, minimum

destek değerinin 2 ve minimum güven değerinin de %50 olarak belirlendiği varsyılarak Şekil değerlendirilmiştir.



Şekil-11 *Apriori algoritması örnek çalışma (Apriori-Algorithm, 2020)*

Şekil-11'de verilen örnek kısa ve algoritmanın mantığını anlamak üzere olduğundan çıkan sonuç ikili bir değer vermiş ve sonuç olarak %50 minimum güven değerini sağladığından herhangi bir birliktelik çıkama durumu oluşmadı. Fakat sonuç olarak 2'den fazla ürünlü bir liste çıkmış olsaydı her bir ikilinin alındığında üçünkü ürünün alınma olasılığını da hesaplayarak minumum güven altında kalanların elenmesi gerekecekti.

Apriori algoritmasının avantajları:

- Diğer birliktelik analizi algoritmalarına göre anlaşılması en kolay olanıdır.
- Ortaya çıkan kurallar sezgiseldir ve son kullanıcıya iletilmesi kolaydır.
- Etiketlenmş veri kullanmanıza gerek olmadığından birçok farklı durumda kullanılabilir.

- Kapsamlı bir algoritmadır ve verilen support ve confidence değerine göre tüm kuralları çıkarır.

Apriori algoritmasının dez avantajları:

- Veri tabanını çok fazla tarama ihtiyacı duyar
- Çok yavaştır

5.3.4. Fp-Growth Algoritması

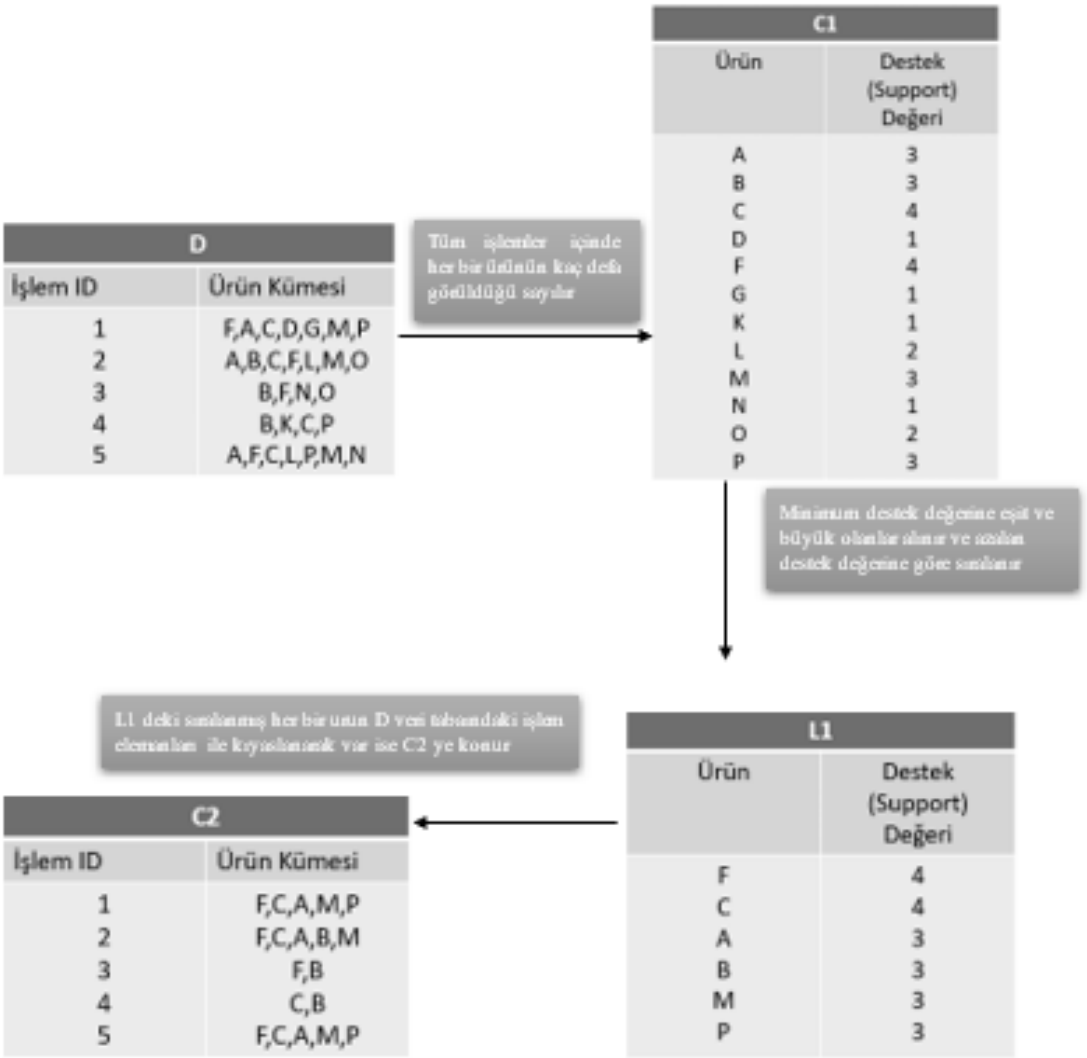
FP-Growth, sık örüntüleri bulmak için kullanılan bir birliktelik (association) algoritmasıdır. Bu algoritmanın önceki çoğu algoritmadan daha etkili bir şekilde çalışarak maliyeti azalttığı görülmüştür. Bunun en büyük nedeni, tüm veritabanını daha küçük ve daha yoğun bir veri yapısı, sık örüntü ağacı (FP-Tree), içinde tutmasıdır. Apriori tabanlı algoritmalardan farklı olarak FP-Growth içinde tüm veritabanı sadece iki kez taranır. İlki tüm ögelerin destek değerinin hesaplanması için, ikincisi ise ağaç yapısının oluşturulması içindir.

FP-Growth, yeni aday üretimine ve her seferinde veritabanının taranmasını ortadan kaldırdığı için büyük veritabanları için bir kazanımdır. Algoritmanın çalışması (Özdoğan, Abul, & Yazıcı, 2009)'da da anlatıldığı gibi şu şekildedir: Veritabanı bir kez taranarak her ögenin (item) destek değeri (support) hesaplanır. Destek değerleri, algoritma içerisinde atanan destek eşik değerine büyük ve eşit olan ögeler büyükten küçüğe sıralanarak bir liste içine konur. Aynı şekilde veritabanında bulunan her hareket (transaction) içerisindeki ögelerde destek değerine göre büyükten küçüğe sıralanır.

Sık örüntü ağacını oluşturmak için ilk olarak root adında yeni bir düğüm (node) oluşturulur. Sonrasında her bir hareket öğre sırasına uygun şekilde ağaç içerisine yerleştirilir. Bu süreç şu şekilde gerçekleşir: veri kümesinde yer alan bir öge eğer ağaçta yoksa o öge için yeni bir düğüm oluşturulur ve destek değeri 1 yapılır. Destek değerleri de ögelerle beraber tutulur. Eğer o öge daha önce oluşturulmuşsa sadece o düğümün destek değeri 1 arttırılır. Düğümler arasındaki ilişkiyi tutmak için de bir başlık tablosu (header table) tutulur. Bu tabloda her düğümün başlangıç noktası işaretlenir. Aynı zamanda ağaç içerisindeki aynı düğümler birbirine işaretçilerle bağlanır. Ağaç oluşturma işlemi tamamlandıktan sonra bu yapının üzerine sıklığı en az olan öğeden başlanarak Fp-Growth algoritması çalıştırılır. İçinde o ögenin geçtiği yollar belirlenir. Her yol içinde, o ögenin destek değeri o yolun destek değeri olarak atanır. Bu yollar

o ögenin şartlı örüntü temelini (conditional pattern base) oluşturur. Her bir şartlı örüntü temelinden şartlı örüntü ağacı (conditional pattern tree) oluşturulur. Daha sonra bu şartlı örüntü ağacı üzerinde algoritma özyineli (recursive) olarak yeniden çalışır. Tablo içindeki her bir öge için bu süreç tekrarlanır ve böylece sık ögeler kümesi belirlenir. Algoritma böl ve yönet yaklaşımına uygun olarak ana görevin kendi içinde alt görevlere ayrılmasına olanak verir.

Fp-Growt algortimasının çalışma mantığına örnek olması açısından Şekil-12 (Fp-Growth-Algorithm, 2020)şeması eklenmiştir. Minimum destek değeri 3 olarak belirlenmiştir.



Şekil-12 *Fp-Growth algoritması örnek çalışma (Fp-Growth-Algorithm, 2020)*

Sık örüntü ağacı (FP-Tree) yapısı Şekil-13 (Shah, Shah, Shetty, & Shah, 2016) 'de anlaşım kolaylığı için örneklendirilmiştir.

Yapılan işlemlerin aşağıdaki sıra ile işlediğini varsayalım:

T1: A B C

T2: A B C E F

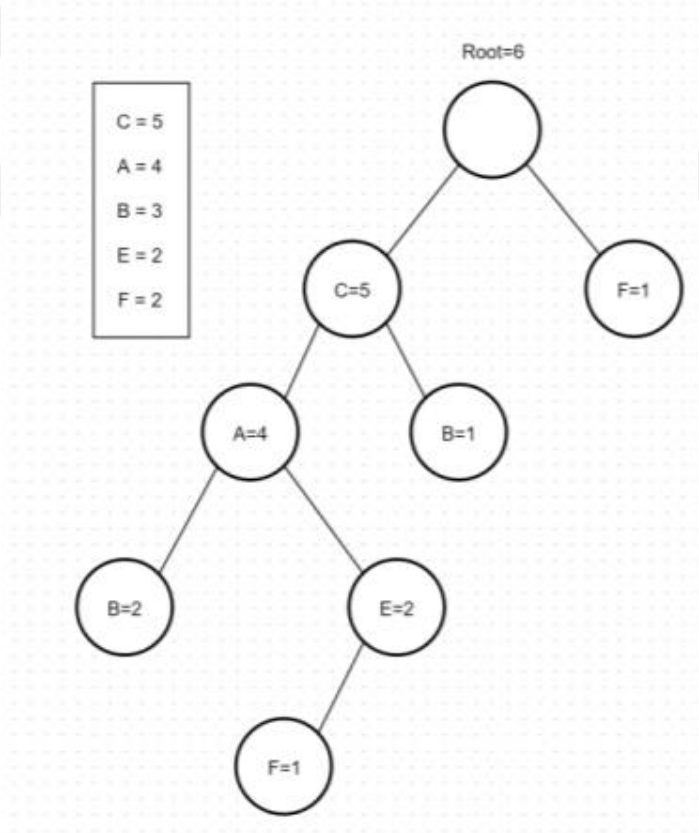
T3: D F

T4: A B C

T5: A C E G

T6: B C

Her bir işlemde ki ürün destek değerinin büyüklüğü baz alınarak şekil içerisindeki dikdörtgen kutucukta görüldüğü gibi yerleştirilmiştir ve sık örüntü ağacı (Fp-Tree) bunu baz alarak ilerlemektedir.



Şekil-13 *Sık Örüntü Ağacı (Fp-Tree) (Shah, Shah, Shetty, & Shah, 2016)*

Fp-Growth Algoritmasının avantajları:

- Apriori Algoritmasından daha hızlıdır

- Database içerisindeki datalar üzerinden sadece iki defa geçer
- Aday küme üretimi yoktur

Fp-Growth algoritmasının dezavantajları:

- Fp-Tree hafızaya sığmayabilir
- Fp-Tree build edilmesi maliyetlidir

5.4.Market Sepeti Analizi

Market Sepeti Analizi birliktelik kuralı madenciliği veya afinite (yakınlık, benzerlik) analizi olarak bilinen, ürün, öğe ve kategori grupları arasındaki ilişkileri tanımlamak amacı ile pazarlama alanında ortaya çıkan bir veri madenciliği tekniğidir. Market sepeti analizi birçok alanda birbirleri arasında herhangi bir bağ olmayan ya da bağı olduğu keşfedilmemiş bağları ortaya çıkarmaktadır.

Birliktelik kuralları analizinin en yaygın kullanıldığı veri madenciliği tekniklerinden biridir. Literatüre adını da Market Sepeti Analizi olarak benimseten bu madencilik tekniği müşterilerin yaptıkları alışverişlerden ileriye dönük satın alma eğilimleri ve alıkşanlıklarını tespit etmek için kullanılır. Müşteri bazlı bir eğilimden ziyade satın alınan ürünlerin birlikteliklerinin de ortaya çıkmasına yardımcı olan bu analiz sayesinde satış yapılan mağazaların yapacakları kampanyalarda veya mağaza yerleşiminde ürünlerin kombinasyonu ve birlikteliği açısından önemli olasılıklar ortaya koymaktadır. Dolayısı ile analizler neticesinde mağazalar müşterilerine daha etkin stratejiler ile satış yaparak hizmet verebilmektedirler.

Market sepeti analizi birliktelik kuralları analizi yapılırken 3 tip kural gözlemlenmiştir (Shah, Shah, Shetty, & Shah, 2016) :

Anlaşılır kural: Anlaşılır olmasını sağlayan, eyleme geçirilebilir ve yüksek kaliteki bilgilierin sağlanmasıdır. Aynı zamanda etkili promosyon önerileri sunan kurallardır. Örneğin; tahmin edilemeyecek bira ve bebek bezi kuralı gibi.

Sıradan kural: Geçmiş pazar bilgileri ve sıradan sağduyu ile belirlenebilen kurallardır. Örneğin; diş fırçası ile düş macununun alınması gibi.

Anlaşılması güç kural: Esrarengiz ve tamamen tesadüfi görünen, herhangi bir açıklaması ya da net bir hareket tarzı olmayan kurallar. Örneğin; a araba yedek parçası ile mutfak gereçlerinin alınması gibi.

Market sepeti analizi geçmiş verilerden yukarıda bahsedilen 3 farklı tipte veriler üretir ve bu verileri servis eder. Servis edilen bu verilerin değerlendirilmesinin ardından uygulamaya alınır ve gözlemlenir. Şirket gelirlerine olan katkısı nispetinde bu yöntem ile devam edilip edilmeyeceği kara. Belirli bir süre sonrasında analizler tekrar edilip sürecin etkinliği tekrar gözden geçirilir.

Market sepeti analizinin uygulanmasının ardından ne gibi sonuçlar elde edildiğine dair çeşitli markaların bir arada bulunduğu bir mağaza analizinden edinilen bilgiler (Shah, Shah, Shetty, & Shah, 2016) şu şekildedir:

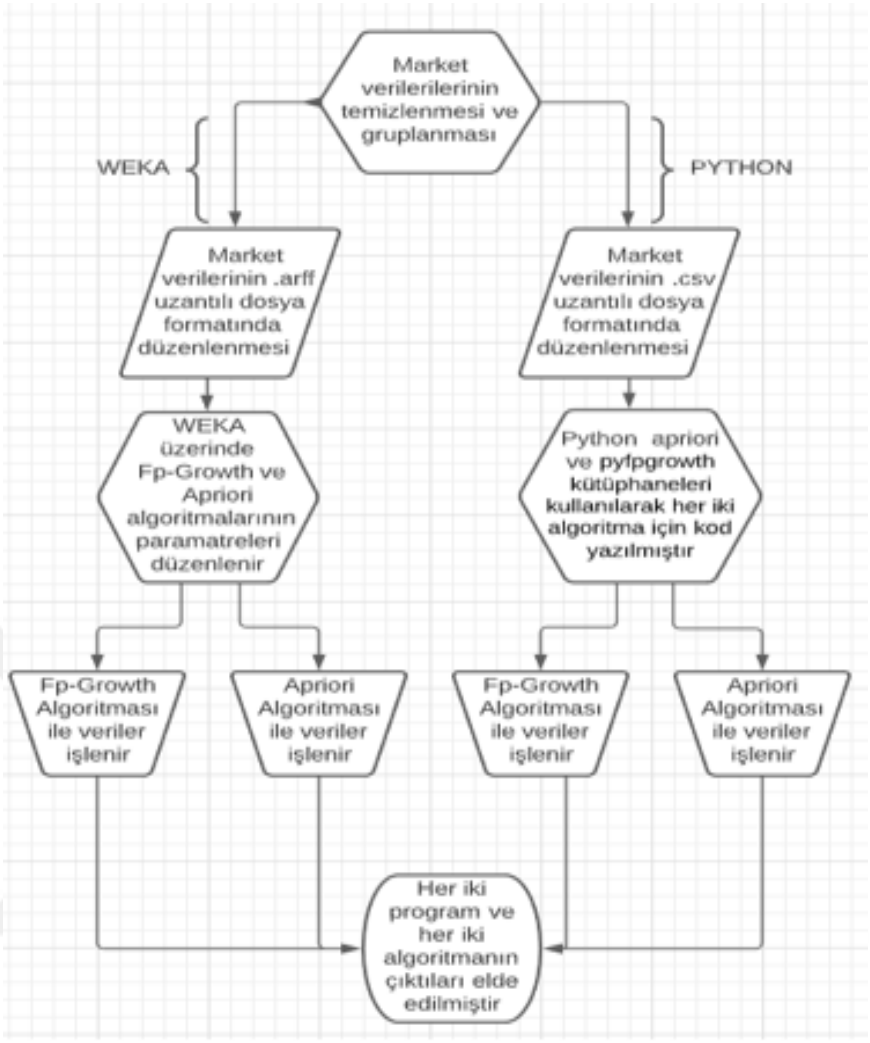
- Promosyon ürünlerinin ana ürün ile birlikte satılmasının, ana ürün gelirlerinde azalmaya neden olmanın aksine satış hacminin artışı ile birlikte gelirlerinin artasına neden olmuştur.
- Eğer market sepeti analizi doğru ürün ikililerini (ürün birliktelikleri ikiden fazla da olabilir) bulmada kullanılmış ise "3 Al 2 Öde" kampanyaları satışları çok verimli bir şekilde etkilemektedir.
- Market sepeti analizi yapılarak satış hacimlerini arttırmak için belirlenen ve birlikte satışı denenen ürün setlerinin teşfikleri genellikle iskonto şeklindedir.

6. METOD VE METODOLOJİ

6.1.Problem ve Kullanılan algoritmalar

Bu çalışmada X Süper Marketinin hangi ürünleri birarada veya market dizaynında yan yana satması gerektiğini ve mevcut market yapısı düzenini buna göre değiştirip değiştirmemesi gerektiği sorununun yanıtını en hızlı sürede verebilecek veri madenciliği algoritmasını ve kullanılacak tool'u bulmaktır.

Bu kapsamda en sık kullanılan Apriori ve FP-Growth Algoritmaları kullanıldı. Şekil-14 içerisinde gösterildiği gibi verilerin hazırlık aşamasından, her iki tool üzerinde hangi aşamalardan geçtiği ve sonuca ulaşıldığı bilgisi bir akış ile özetlenmiştir.



Şekil-14 *Çalışma akış diyagramı*

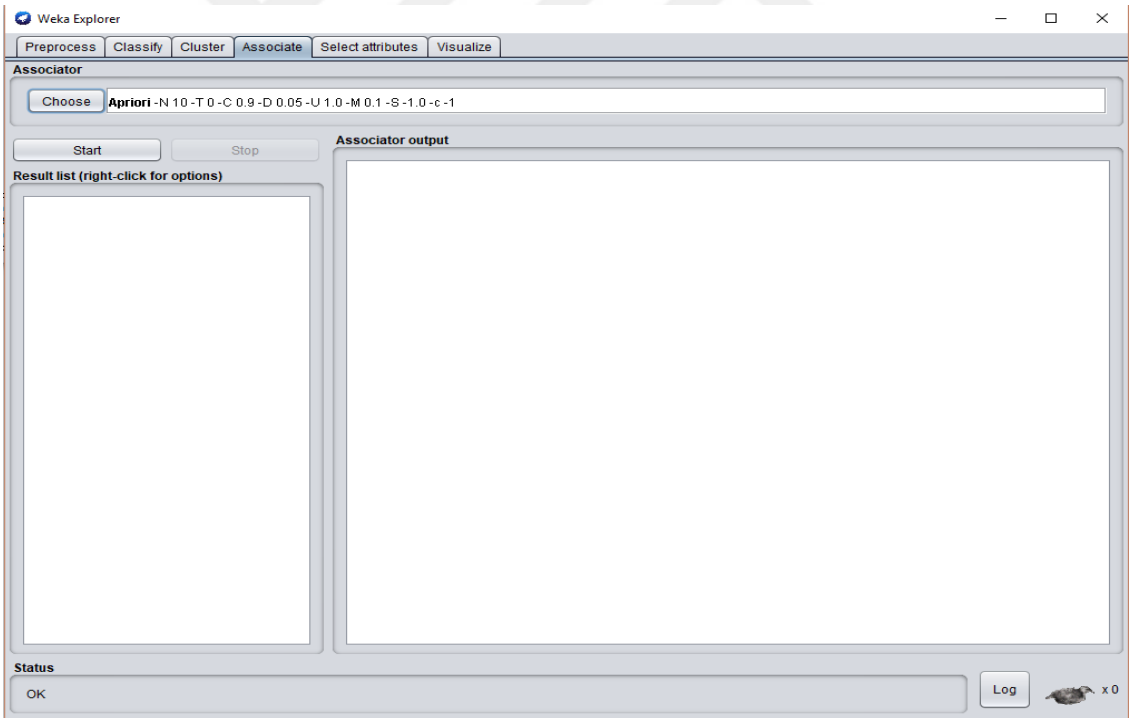
6.2. Kullanılan Uygulamalar

Çalışma kapsamında veri madenciliği alanında sıklıkla kullanılan bir tool olan WEKA uygulaması ve Python kütüphanesinde bulunan Apriori ve Fp-Growth algoritmalarını rahatlıkla uygulayabilmek adına PyCharm IDE (Integrated Development Environment) ortamı kullanılacaktır.

6.2.1. Weka Uygulaması

Weka, veri madenciliği görevleri için makina öğrenimi algoritmalarının toplandığı paketlerden biridir. Algoritmalar direk uygulama içerisinden veya kendi java kodunuzdan çağırılabilir. WEKA üzerinde makine öğrenmesi ve istatistik ile ilgili pekçok kütüphane hazır olarak gelmektedir. Örneğin veri ön işlemesi (data preprocessing), ilkelleme (regression), sınıflandırma (classification), gruplandırma (clustering), özellik seçimi veya özellik çıkarımı (feature extraction) bunlardan bazılarıdır. Ayrıca bu işlemler sonucunda çıkan neticelerinde görsel olarak gösterilmesini sağlayan görüntüleme (visualization) araçları bulunmaktadır.

Çalışma kapsamında birliktelik analizi yapılacağından Association paneli Şekil-15'de verilen ekran üzerinden çalışılmıştır.



Şekil-15 *Weka Uygulaması Birliktelik (Association) Analizi Ekranı*

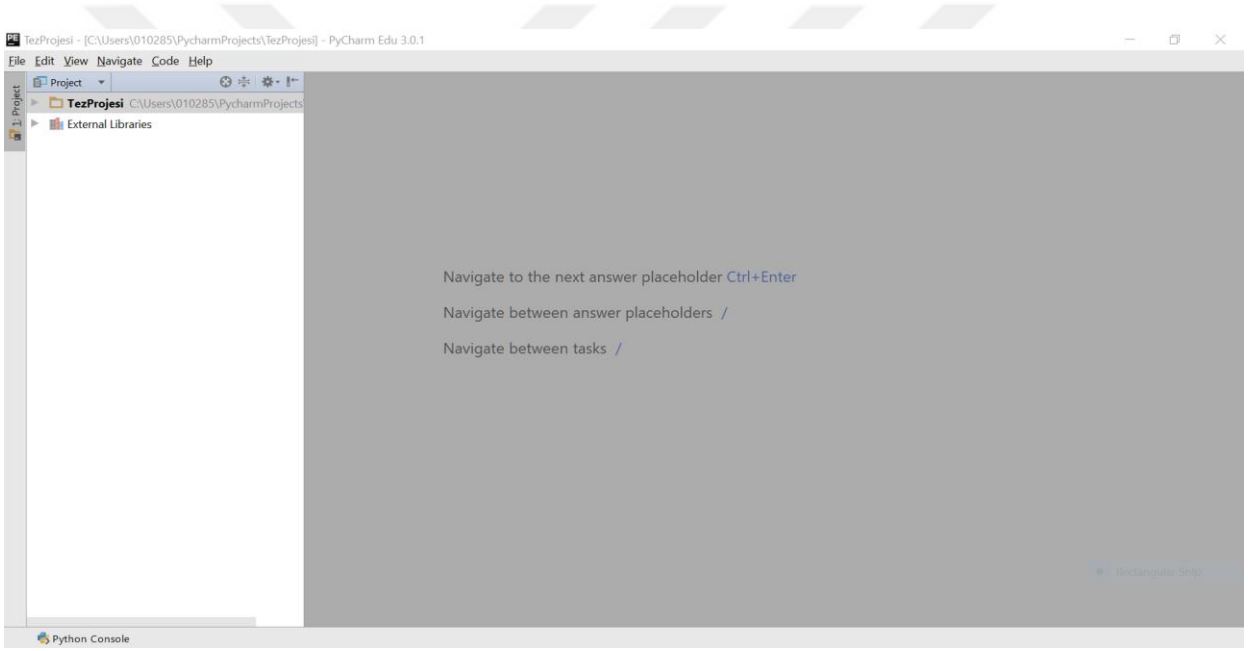
Panelde ilk görülen Apriori Algoritmasıdır. Choose butonu ile Fp-Growth Algoritması da seçilebilmektedir. Çalışmada her iki algoritmada kullanıldığından oluşturulan set üzerinden her iki algoritma da çalıştırılarak Asociator Output ekranından sonuçlar okunmuştur.

6.2.2. Pycharm Uygulaması

Günümüzde Python genel amaçlı programlama için popüler bir üst seviye programlama dili haline gelmiştir. Python'ın oldukça hızlı bir şekilde kavranmasını kolaylaştıran temiz sözdizimi ve girinti yapısına sahiptir.

Py-Charm ise Python dilini yazabileceğiniz ve tüm geliştirme araçlarını biraraya getiren güçlü ve platformlar arası bir IDE (Entegre Geliştrme Ortamı) 'dir. Ücretsiz ve açık kaynak kodludur.

IDE açılış ekran görseli



Şekil-16 *PyCharm IDE*

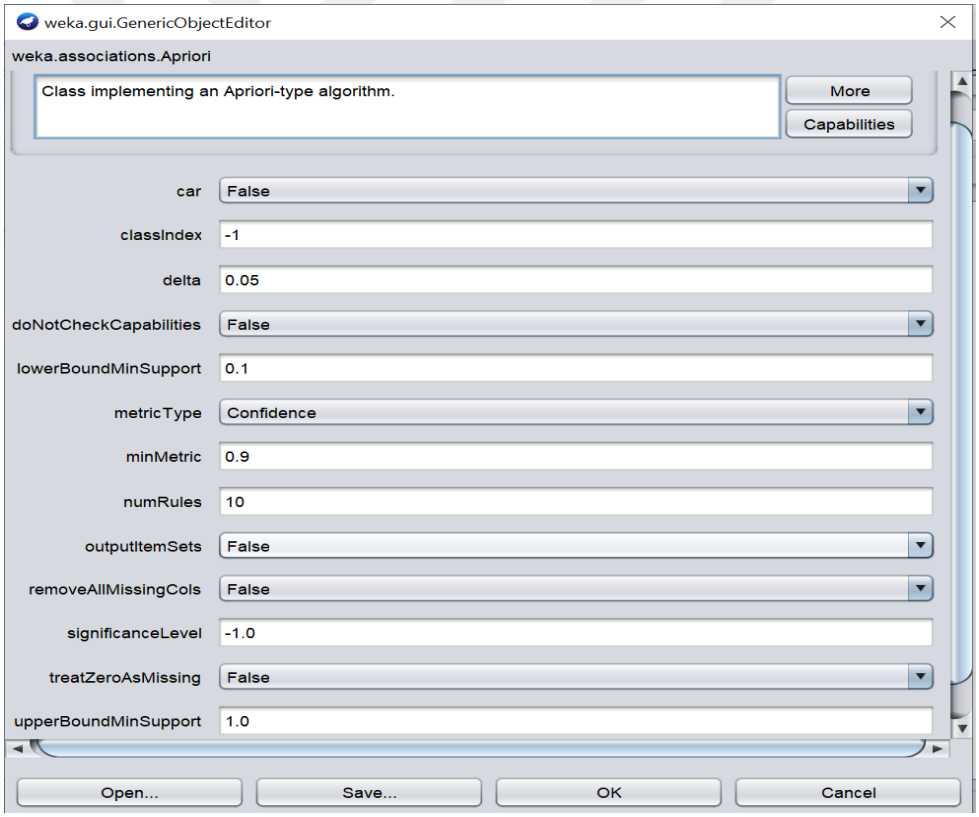
6.3.WEKA Uygulaması Algoritma Karşılaştırması

WEKA uygulamasında Apriori ve Fp-growth algoritmalarının karşılaştırmasını yapmak için düzenlediğimiz X süper market verilerini .arff uzantılı dosya haline getirdik. Market verileri toplamda 8230 adet olduğundan özellikle her iki algoritmanın arasıdndaki farkı biraz daha fazla verinin olduğu bir sette daha rahat gözlemleyebilmek için veriler çoklanarak 10 katına çıkarılmıştır. Bu çoklama verilerin herhangi bir şekilde destek ve güven aralıklarına etki etmemekle birlikte sadece performans açısından gözlemlemek içindir.

6.3.1. WEKA – Apriori Algoritması

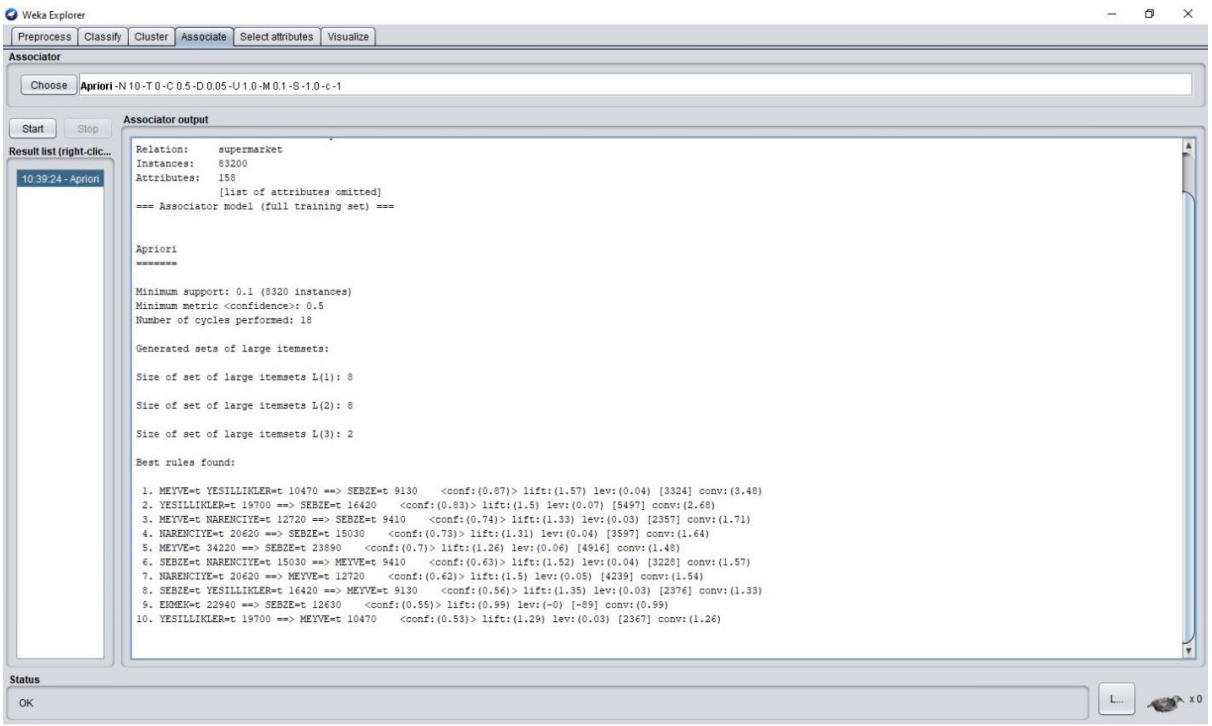
Weka uygulamasında öncelikle Apriori algoritması çalıştırılmıştır. Algoritmayı çalışmadan önce yapılan adımlar sırasıyla şu şekildedir:

1. Kullanılacak olan .arff uzantılı dosya seçilir ve hatasız yüklendiği görülür
2. Associate sekmesine geçilir
3. Choose butonuna basılarak Apriori algoritması seçilir
4. Algoritma Şekil-17 de olduğu gibi temelde seçili olan değerlerle gelir, bu değerleri kendi isterlerimize göre girip Save edebilir ya da Ok tuşuna basarak algoritmanın kısıtlarını belirlenir



Şekil-17 *Algoritma Kısıtlarını belirleme ekranı*

5. Start butonuna basılır ve Associator Output ekranında Şekil-18'de olduğu gibi sonuçlar görülür.



Şekil-18 *WEKA-Apriori Algoritması Sonuçları*

Algoritmanın kısıtlarından biri de kaç adet birliktelik sonucunun listelenmesini istediğinizdir. Biz en iyi 10 adet sonucun listelenmesini istedik ve Apriori Algoritmasının WEKA kullanılarak elde dilen sonuçları şağıdaki şekilde yorumlanabilir:

1. MEYVE=t YESILLIKLER=t 10470 ==> SEBZE=t 9130 <conf:(0.87)> lift:(1.57) lev:(0.04) [3324] conv:(3.48)

MEYVE VE YEŞİLLİK alanların %87 ‘si SEBZE almaktadır.

2. YESILLIKLER=t 19700 ==> SEBZE=t 16420 <conf:(0.83)> lift:(1.5) lev:(0.07) [5497] conv:(2.68)

YEŞİLLİK alanların %83’ü SEBZE almaktadır.

3. MEYVE=t NARENCIYE=t 12720 ==> SEBZE=t 9410 <conf:(0.74)> lift:(1.33) lev:(0.03) [2357] conv:(1.71)

MEYVE ve NARENCIYE alanların %74 ‘ü SEBZE almaktadır.

4. NARENCIYE=t 20620 ==> SEBZE=t 15030 <conf:(0.73)> lift:(1.31) lev:(0.04) [3597] conv:(1.64)

NARENCİYE alanların %73’ü SEBZE almaktadır.

5.MEYVE=t 34220 ==> SEBZE=t 23890 <conf:(0.7)> lift:(1.26) lev:(0.06) [4916] conv:(1.48)

MEYVE alanların %70’I SEBZE almaktadır.

6. SEBZE=t NARENCIYE=t 15030 ==> MEYVE=t 9410 <conf:(0.63)> lift:(1.52) lev:(0.04) [3228] conv:(1.57)

SEBZE VE NARENCİYE alanların %63’ü MEYVE almaktadır.

7.NARENCIYE=t 20620 ==> MEYVE=t 12720 <conf:(0.62)> lift:(1.5) lev:(0.05) [4239] conv:(1.54)

NARENCİYE alanların %62’si MEYVE almaktadır.

8. SEBZE=t YESILLIKLER=t 16420 ==> MEYVE=t 9130 <conf:(0.56)> lift:(1.35) lev:(0.03) [2376] conv:(1.33)

SEBZE ve YEŞİLLİK ALANLARIN %56’sı MEYVE almaktadır.

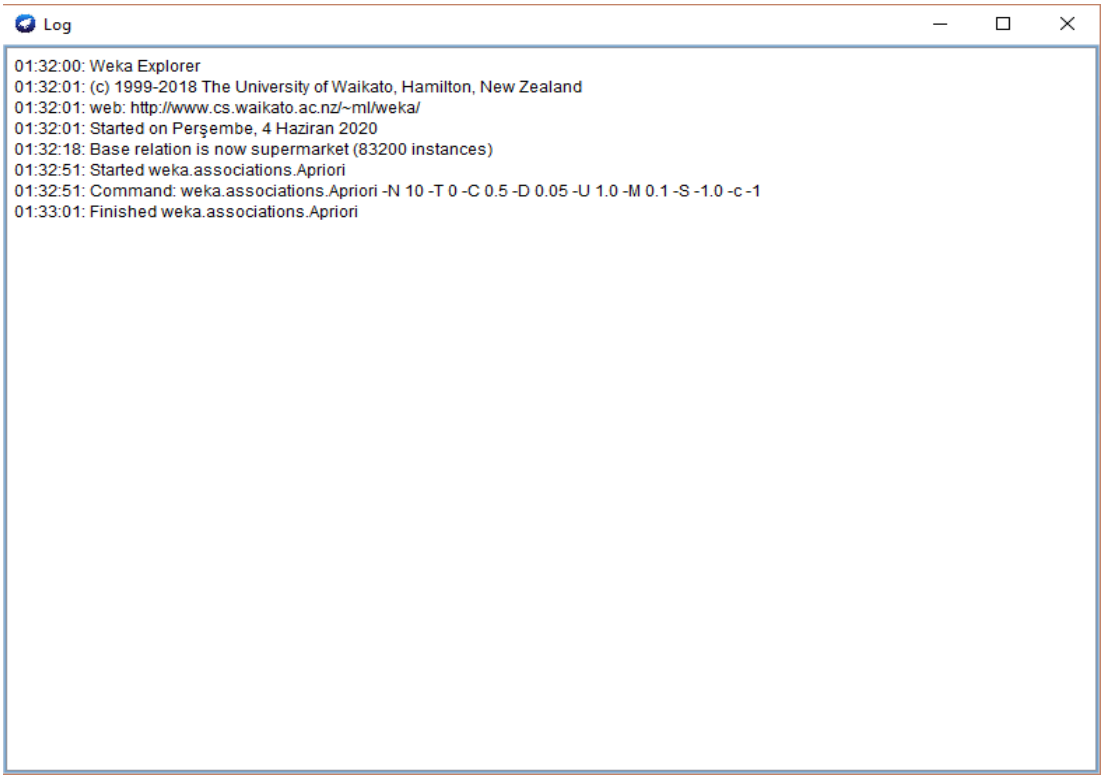
9. EKMEK=t 22940 ==> SEBZE=t 12630 <conf:(0.55)> lift:(0.99) lev:(-0) [-89] conv:(0.99)

EKMEK alanların %55’I SEBZE almaktadır.

10. YESILLIKLER=t 19700 ==> MEYVE=t 10470 <conf:(0.53)> lift:(1.29) lev:(0.03) [2367] conv:(1.26)

YEŞİLLİK alanların %53'ü MEYVE almaktadır.

Weka programının Şekil-15 görselinde sağ alt köşede Log butonu bulunmaktadır. Bu butona tıklanarak çalıştırdığımız algoritma ile ilgili logları Şekil-19'da görüldüğü gibi analiz edebiliyoruz.



Şekil-19 *WEKA- Apriori Logları*

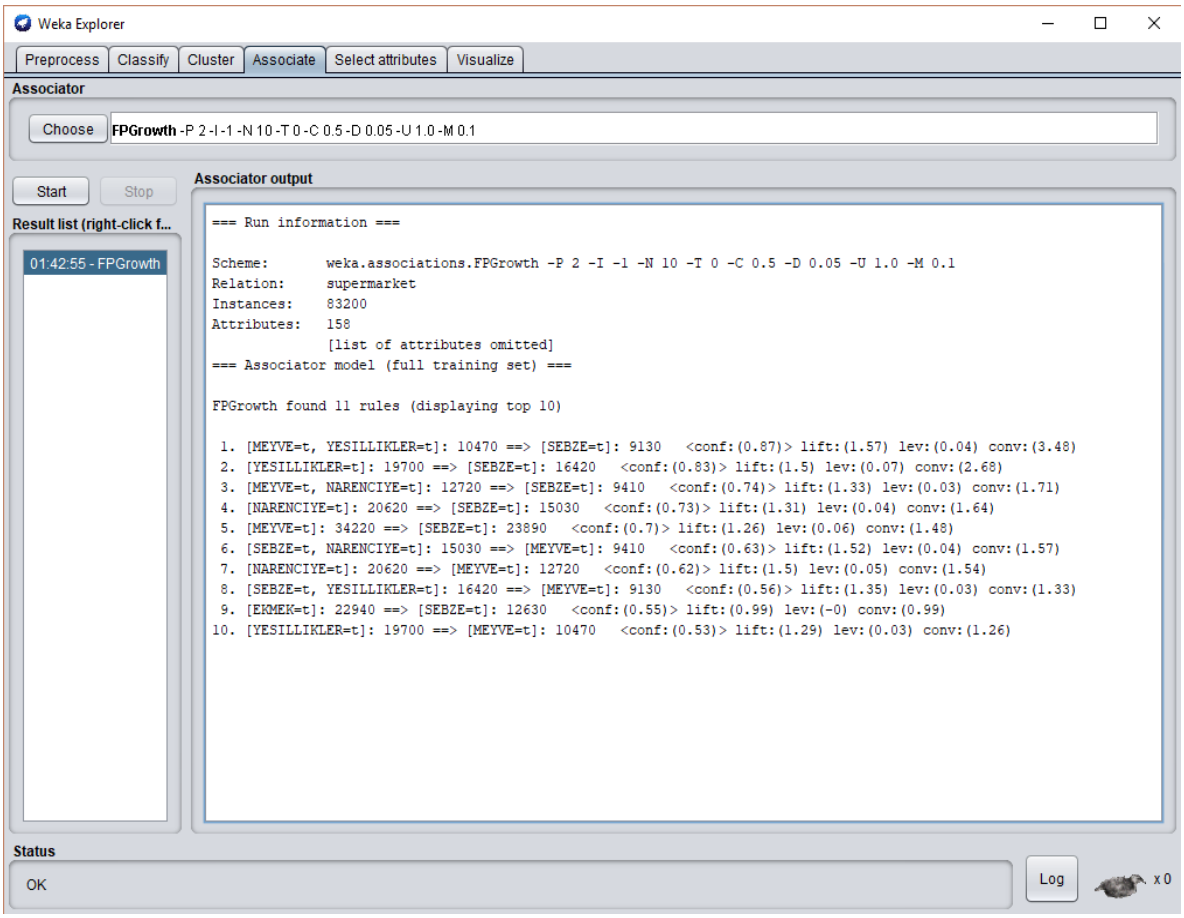
Apriori Algoritması log da görüldüğü üzere aşağıdaki saatlerde başlamış ve bitmiştir. 10 saniye içerisinde 83200 verinin işlenmesi tamalanmıştır.

Tablo-2 *WEKA- Apriori Algoritma Zaman logları*

01:32:51	***Started weka.associations.Apriori***
01:33:01	***Finished weka.associations.Apriori***

6.3.2. WEKA- Fp-Growth Algoritması

WEKA uygulaması üzerinden Apriori algoritmasında gerçekleştirilen tüm adımlar Fp-growth algoritmasında da gerçekleştirilmiştir. Tek farkı 3. Adımda Apriori algoritması değil Fp-Growth algoritması seçilmiştir. Şekil-20 ‘de Fp-Growth algoritmasının Weka sonuçları bulunmaktadır.



Şekil-20 *WEKA-FP-Growth Algoritması Sonuçları*

Sonuçlar tamamen Apriori algoritmasının uygulamasında alınan sonuçlar ile aynı çıkmıştır. Fakat burada iki algroritmanın işlemleri tamamlama zamanları arasında 10 kat fark bulunmaktadır. Şekil-9 ‘da Fp-Growth algoritmasının log bilgileri görülmektedir.

Şekil-21 *WEKA-Fp-Growth Algoritması Logları*

Fp-Growth Algoritması Log'da görüldüğü üzere aşağıdaki saatlerde başlamış ve bitmişti. 1 saniye içerisinde 83200 verinin işlenmesi tamamlanmıştır.

Tablo-3 *WEKA-Fp-Growth Algoritması Zaman logları*

01:42:55	***Started weka.associations.FPGrowth***
01:42:56	***Finished weka.associations.FPGrowth***

6.4.PyCharm Uygulaması (Python Programlama dili) ile Algoritmalarının Karşılaştırılması

PyCharm uygulaması ile çalışma kapsamında Python dili ve veri madenciliği birliktelik kurallari kapsamında Apriori (apyori, 2020) ve FP-Growth (pyfpgrowth, 2020) algoritmaları adına yazılan bazı kütüphaneleri kullanılarak algoritmalarının çıkardığı birliktelik kuralları ve performansları karşılaştırılmıştır.

6.4.1. PyCharm-Apriori Algoritması

Apriori algoritması adına Python dili kullanılmış ve apyori kütüphanesi ile çalışılmıştır. WEKA uygulamasında kullanılan verilerin birbir aynısı kullanılmış fakat .csv formatında koda verilmiştir. Şekil-22'de kodun bir kısmının görseli verilmiştir.

tezApriori.py

```
# -*- coding: utf-8 -*-

import pandas as pd
import numpy as np
from apyori import apriori
from datetime import datetime


store_data = pd.read_csv('C:\\Users\\010285\\Desktop\\Thesis_Data_Python.csv',error_bad_lines=False ,
                         encoding='utf-8',header=None)

store_data.head()

records = []

for sublist in store_data.values.tolist():
    clean_sublist = [item for item in sublist if item is not np.nan]
    for i in range(0, 10):
        records.append(clean_sublist)

print(len(records))
print (datetime.utcnow().isoformat(sep=' ', timespec='milliseconds'))
association_rules = apriori(records,  min_support=0.05, min_confidence=0.5)
association_results = list(association_rules)
print (datetime.utcnow().isoformat(sep=' ', timespec='milliseconds'))
association_results.sort()

for k in range (0 , len(association_results)):
    print(association_results[k])
```

Şekil-22 *PyCharm-Apriori Algoritması*

Kodun başına ve birliktelik kurallarının çıkarımının ardından son kısmına yerleştirilen zaman gösterimleri ile performans süresini analiz etmiş oluyoruz. Verilerin 10 kat çoklandığı kod parçasının süresi milisaniyeler seviyesinde olduğundan kodun başına ve sonuna yerleştirilen zaman gösterimleri ile devam edilmeye karar verildi.

Apriori Algoritması kullanılmış Python kodu ile elde ettiğimiz çıktılar ve zaman bilgileri aşağıdaki gibidir.

RelationRecord(items=frozenset({'SEBZE', 'DİIER EKMEK(İHE)'}), support=0.05666747046057391,

ordered_statistics=[OrderedStatistic(items_base=frozenset({'DİIER EKMEK(İHE)'}), items_add=frozenset({'SEBZE'}), confidence=0.5497076023391814, lift=0.9885678347357264)])

RelationRecord(items=frozenset({'SEBZE', 'EKMEK'}), support=0.15167591029659996, ordered_statistics=[OrderedStatistic(items_base=frozenset({'EKMEK'}), items_add=frozenset({'SEBZE'}), confidence=0.55054704595186, lift=0.9900774499403137)])

RelationRecord(items=frozenset({'NARENCİYE', 'MEYVE'}), support=0.15300217024354956, ordered_statistics=[OrderedStatistic(items_base=frozenset({'NARENCİYE'}), items_add=frozenset({'MEYVE'}), confidence=0.6160194174757282, lift=1.4943741001882684)])

RelationRecord(items=frozenset({'SEBZE', 'MEYVE'}), support=0.28779840848806365, ordered_statistics=[OrderedStatistic(items_base=frozenset({'MEYVE'}), items_add=frozenset({'SEBZE'}), confidence=0.6981573559520328, lift=1.2555327645850303), OrderedStatistic(items_base=frozenset({'SEBZE'}), items_add=frozenset({'MEYVE'}), confidence=0.5175628794449262, lift=1.25553276458503)])

RelationRecord(items=frozenset({'YEŞİLLİKLER', 'MEYVE'}), support=0.12623583313238484, ordered_statistics=[OrderedStatistic(items_base=frozenset({'YEŞİLLİKLER'}), items_add=frozenset({'MEYVE'}), confidence=0.5312024353120243, lift=1.2886203563843022)])

RelationRecord(items=frozenset({'SEBZE', 'NARENCİYE'}), support=0.18097419821557753, ordered_statistics=[OrderedStatistic(items_base=frozenset({'NARENCİYE'}), items_add=frozenset({'SEBZE'}), confidence=0.7286407766990292, lift=1.3103526890593555)])

RelationRecord(items=frozenset({'SEBZE', 'SÜT'}), support=0.05027730889799855, ordered_statistics=[OrderedStatistic(items_base=frozenset({'SÜT'}), items_add=frozenset({'SEBZE'}), confidence=0.5545212765957446, lift=0.9972245160635527)])

RelationRecord(items=frozenset({'SEBZE', 'TATLI BİSKÜVİ-ATIŞTIRMALIK'}), support=0.05980226669881842, ordered_statistics=[OrderedStatistic(items_base=frozenset({'TATLI BİSKÜVİ-ATIŞTIRMALIK'}), items_add=frozenset({'SEBZE'}), confidence=0.5310492505353319, lift=0.9550135481222989)])

RelationRecord(items=frozenset({'SEBZE', 'TAVUK'}), support=0.05980226669881842, ordered_statistics=[OrderedStatistic(items_base=frozenset({'TAVUK'}), items_add=frozenset({'SEBZE'}), confidence=0.6350832266325224, lift=1.142103270097602)])

RelationRecord(items=frozenset({'SEBZE', 'YEŞİLLİKLER'}), support=0.19809500843983602, ordered_statistics=[OrderedStatistic(items_base=frozenset({'YEŞİLLİKLER'}), items_add=frozenset({'SEBZE'}), confidence=0.8335870116692033, lift=1.49908297371734)])

RelationRecord(items=frozenset({'SEBZE', 'YUMURTA'}), support=0.054858934169279, ordered_statistics=[OrderedStatistic(items_base=frozenset({'YUMURTA'}), items_add=frozenset({'SEBZE'}), confidence=0.5893782383419689, lift=1.05990960728714)])

RelationRecord(items=frozenset({'SEBZE', 'EKMEK', 'MEYVE'}), support=0.07607909332047263, ordered_statistics=[OrderedStatistic(items_base=frozenset({'EKMEK', 'MEYVE'}), items_add=frozenset({'SEBZE'}), confidence=0.7236238532110092, lift=1.3013304940442563), OrderedStatistic(items_base=frozenset({'SEBZE',

'EKMEK'}), items_add=frozenset({'MEYVE'}), confidence=0.5015898251192369, lift=1.2167844426846888)])

RelationRecord(items=frozenset({'SEBZE', 'NARENCİYE', 'MEYVE'}), support=0.11309380274897517, ordered_statistics=[OrderedStatistic(items_base=frozenset({'NARENCİYE', 'MEYVE'}), items_add=frozenset({'SEBZE'}), confidence=0.7391646966115051, lift=1.3292784027961455), OrderedStatistic(items_base=frozenset({'SEBZE', 'NARENCİYE'}), items_add=frozenset({'MEYVE'}), confidence=0.6249167221852099, lift=1.5159576758713458)])

RelationRecord(items=frozenset({'YEŞİLLİKLER', 'NARENCİYE', 'MEYVE'}), support=0.06631299734748011, ordered_statistics=[OrderedStatistic(items_base=frozenset({'YEŞİLLİKLER', 'MEYVE'}), items_add=frozenset({'NARENCİYE'}), confidence=0.5253104106972303, lift=2.115011915690693), OrderedStatistic(items_base=frozenset({'YEŞİLLİKLER', 'NARENCİYE'}), items_add=frozenset({'MEYVE'}), confidence=0.6145251396648046, lift=1.4907491981222256)])

RelationRecord(items=frozenset({'SEBZE', 'YEŞİLLİKLER', 'MEYVE'}), support=0.11007957559681697, ordered_statistics=[OrderedStatistic(items_base=frozenset({'YEŞİLLİKLER', 'MEYVE'}), items_add=frozenset({'SEBZE'}), confidence=0.8720152817574022, lift=1.56819053488636), OrderedStatistic(items_base=frozenset({'SEBZE', 'YEŞİLLİKLER'}), items_add=frozenset({'MEYVE'}), confidence=0.5556908094948265, lift=1.3480256139076021)])

RelationRecord(items=frozenset({'SEBZE', 'NARENCİYE', 'YEŞİLLİKLER'}), support=0.09054738365083193, ordered_statistics=[OrderedStatistic(items_base=frozenset({'SEBZE', 'NARENCİYE'}), items_add=frozenset({'YEŞİLLİKLER'}), confidence=0.5003331112591606, lift=2.1054098552934946), OrderedStatistic(items_base=frozenset({'YEŞİLLİKLER', 'NARENCİYE'}), items_add=frozenset({'SEBZE'}), confidence=0.8391061452513967,

lift=1.509008319322438)])

RelationRecord(items=frozenset({'SEBZE', 'NARENCİYE', 'YEŞİLLİKLER', 'MEYVE'}), support=0.05763202314926453, ordered_statistics=[OrderedStatistic(items_base=frozenset({'YEŞİLLİKLER', 'NARENCİYE'}), items_add=frozenset({'SEBZE', 'MEYVE'}), confidence=0.5340782122905028, lift=1.8557371984656181), OrderedStatistic(items_base=frozenset({'SEBZE', 'NARENCİYE', 'MEYVE'}), items_add=frozenset({'YEŞİLLİKLER'}), confidence=0.509594882729211, lift=2.1443835400081563), OrderedStatistic(items_base=frozenset({'YEŞİLLİKLER', 'NARENCİYE', 'MEYVE'}), items_add=frozenset({'SEBZE'}), confidence=0.869090909090909, lift=1.5629314830875973), OrderedStatistic(items_base=frozenset({'SEBZE', 'YEŞİLLİKLER', 'MEYVE'}), items_add=frozenset({'NARENCİYE'}), confidence=0.5235487404162104, lift=2.107919054860218), OrderedStatistic(items_base=frozenset({'SEBZE', 'NARENCİYE', 'YEŞİLLİKLER'}), items_add=frozenset({'MEYVE'}), confidence=0.6364846870838882, lift=1.5440198872985578)])

Çıktılara bakıldığında ise WEKA ile neredeyse aynı verileri vermis ve confidence değerleri de birebir aynı çıkmıştır. Analiz sonucunda en yüksek oran aşağıdaki şekilde meyve ve yeşillik alanların sebze aldığı %87 oranıdır.

OrderedStatistic(items_base=frozenset({'YEŞİLLİKLER', 'MEYVE'}), items_add=frozenset({'SEBZE'}), confidence=0.8720152817574022, lift=1.56819053488636)

Zaman olarak ise aşağıdaki başlangıç ve bitiş tarihleri Tablo-3 içerisinde verilmiş ve 1saniyeden kısa bir süre içerisinde apriori algoritması tamamlandığı bilgisi elde edilmiştir.

Tablo-4 *PyCharm- Apriori Zaman Logları*

Started Time	***21:41:46.174***
Finished Time	***21:41:46.504***

6.4.2. PyCharm-Fp-Growth Algoritması

FP-Growth (Frequent Pattern) algoritması için de aşağıdaki kod parçası içerisinde yine WEKA Tool ‘da kullanılan veriler .csv uzantılı dosyaya dönüştürülmüş ve bu veri kullanılmıştır.

Kodun başına ve sonuna zaman gösterimleri yerleştirilerek kodun zaman performansı ölçülmüştür.

tezFpGrowth.py

```
# -*- coding: utf-8 -*-

import pandas as pd
import numpy as np
import pyfpgrowth
from datetime import datetime

print(datetime.utcnow().isoformat(sep=' ', timespec='milliseconds'))
dataset = pd.read_csv('C:\\Users\\010285\\Desktop\\YÜKSEK LİSANS\\Makale- Tez KOD\\Thesis_Data_Python_v3.csv', error_bad_lines=False,
                      encoding='utf-8', header=None)
transactions = []

for sublist in dataset.values.tolist():
    clean_sublist = [item for item in sublist if item is not np.nan]
    for i in range(0, 10):
        transactions.append(clean_sublist)


patterns = pyfpgrowth.find_frequent_patterns(transactions, 4147)
rules = pyfpgrowth.generate_association_rules(patterns, 0.5)
rules_df = pd.DataFrame(data=rules)


def print_inventory(dct):
    for item, amount in dct.items():
        print("{} ({})".format(item, amount))

print_inventory(rules)

print(datetime.utcnow().isoformat(sep=' ', timespec='milliseconds'))

```

Şekil-23 *PyCharm-Fp-Growth Algortması*

Kodun çıktıları aşağıdaki şekildedir:

('MEYVE', 'YEŞİLLİKLER') ((('SEBZE',), 0.8720152817574021))

('MEYVE', 'NARENCİYE', 'YEŞİLLİKLER') ((('SEBZE',), 0.8690909090909091))

('MEYVE', 'NARENCİYE') ((('SEBZE',), 0.7391646966115051))

('EKMEK', 'MEYVE') ((('SEBZE',), 0.7236238532110092))

('MEYVE',) ((('SEBZE',), 0.6981573559520328))

('NARENCİYE', 'SEBZE', 'YEŞİLLİKLER') ((('MEYVE',), 0.6364846870838882))

('TAVUK',) ((('SEBZE',), 0.6350832266325224))

('NARENCİYE', 'SEBZE') ((('MEYVE',), 0.6249167221852099))

('YUMURTA',) ((('SEBZE',), 0.5893782383419689))

('SEBZE', 'YEŞİLLİKLER') ((('MEYVE',), 0.5556908094948265))

('SÜT',) ((('SEBZE',), 0.5545212765957447))

('DİIER EKMEK(İHE)',) ((('SEBZE',), 0.5497076023391813))

('TATLI BİSKÜVİ-ATIŞTIRMALIK',) ((('SEBZE',), 0.5310492505353319))

('MEYVE', 'SEBZE', 'YEŞİLLİKLER') ((('NARENCİYE',), 0.5235487404162102))

('SEBZE',) ((('MEYVE',), 0.5175628794449263))

('MEYVE', 'NARENCİYE', 'SEBZE') ((('YEŞİLLİKLER',), 0.509594882729211))

('EKMEK', 'SEBZE') ((('MEYVE',), 0.5015898251192369))

Analiz sonucunda en yüksek oran aşağıdaki şekilde Meyve ve yeşillik alanların sebze aldığı %87 oranıdır.

('MEYVE', 'YEŞİLLİKLER') ((('SEBZE',), 0.8720152817574021))

Zaman olarak ise aşağıdaki başlangıç ve bitiş tarihleri çıktı olarak verilmiştir:

Tablo-5 *PyCharm-FpGrowth Algoritması Zaman Logları*

Başlama Zamanı	***21:53:06.094***
Bitiş Zamanı	***21:53:06.875***

1 saniyeden daha az bir sürede FP-Growth algoritması çıktıyı türetmiştir.

Özet bir performans karşılaştırma tablosu çıkartarak aşağıdaki şekilde sayı ve içerik olarak aynı olan very setinin Algoritma ve IDE bazlı başlangıç ve bitiş süreleri aşağıda Tablo -1 de ortaya konmuştur.

Tablo-6 *PyCharm- Fp-Growth ve Apriori Algoritması Süre Karşılaştırması*

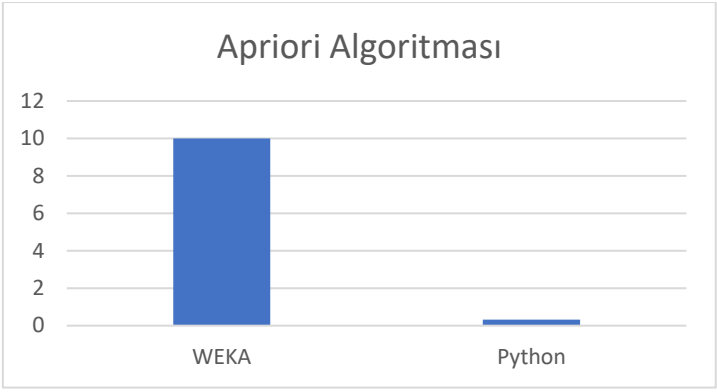
	WEKA	***PYCHARM(PYTHON)***
FP-GROWTH	***1 sn***	***781 ms***
APRIORI	***10sn***	***330 ms***

Tablo-5 gösteriyor ki Fp-Growth algoritması milisaniye seviyesinde Python da daha hızlı sonuç vermekte, Apriori algoritması ise çok büyük bir fark göstererek WEKA uygulamasına oranla Python üzerinde çok daha performanslı çalıştığını göstermiştir. Literatür taramalarında sıkça karşılaşılan FP-Growth algoritmasının Apriori algoritmasına göre daha hızlı çalıştığı bilgisini bu çalışma ile milisaniye seviyesinde de olsa Python programlama dilinde kırıldığını göstermiş oluyoruz. Her algoritma ve her tool içerisinden çıkan birliktelik analiz sonuçlarında da aşağıdaki şekilde Meyve ve Yeşillik alanların %87'sinin Sebze aldığı sonucu çıkmıştır.

7. SONUÇ

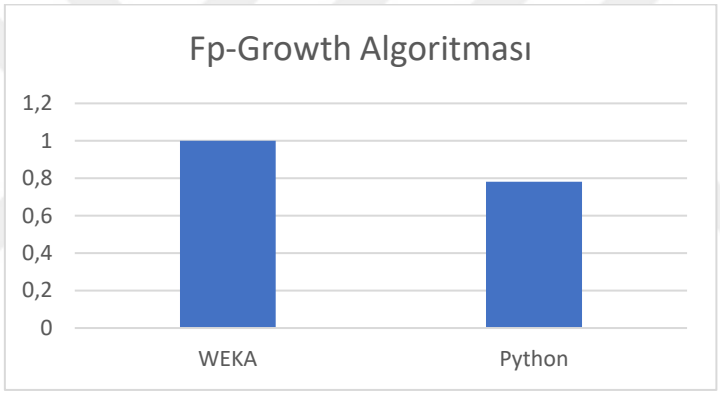
Mevcutta çoğu çalışma herhangi bir tool ve bu tool üzerinde çalışan herhangi bir algoritma seçerek analizler yapılarak sonuçlar ortaya koymuştur. Fakat bu çalışmada bu standart yöntemlerin dışına çıkılmış ve birden fazla tool ve birden fazla algoritma kullanılmış ve performansları kıyaslanmıştır.

Apriori algoritması Şekil 24 'de görüldüğü gibi WEKA üzerinde çok daha uzun sürede Python üzerinde milisaniye seviyelerinde çok büyü bir hız farkı ile çalışmaktadır.



Şekil 24 *Apriori Zaman Çizelgesi*

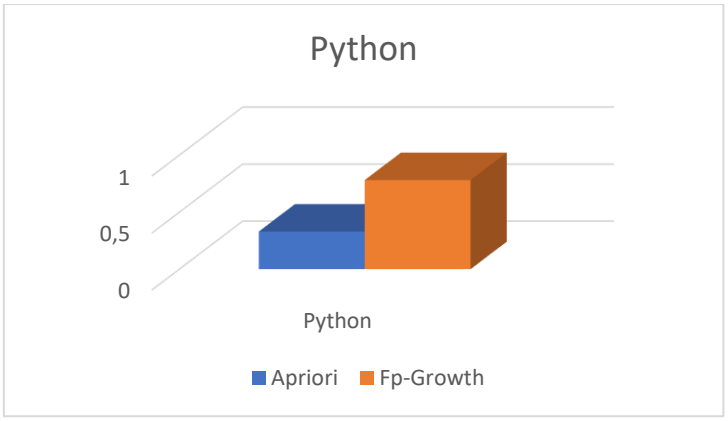
Fp-Growth algoritması Şekil 25 'de görüldüğü gibi WEKA üzerinde 1 saniye Python üzerinde ise 1 saniyeden kısa bir sürede analizini tamamlamaktadır.



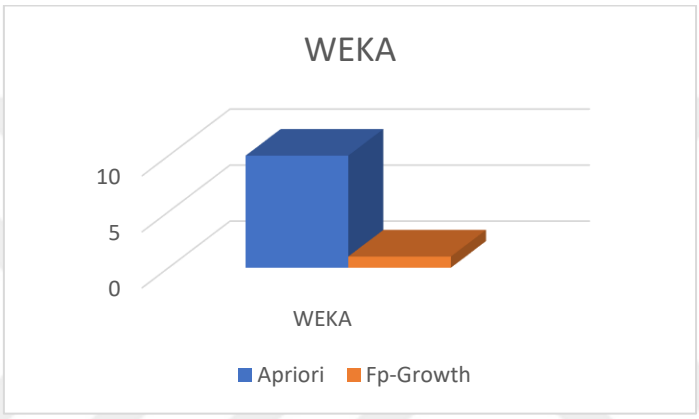
Şekil 25 *Fp-Growth Algoritması Zaman Çizelgesi*

Fp-Growth algoritmasının çok daha performanslı çalıştığı literatür taramalarında azımsanmayacak oranda dile getirilen bir kabuldür. Bu çalışma kapsamında WEKA üzerinde bu kabul doğrulanmış olsa da Python üzerinde bu kabule zıt bir durum milisaniye seviyesinde gözlemlenmiştir.

Şekil-26 ve Şekil-27 'de görüldüğü üzere Python üzerinde her iki algoritma da milisaniye seviyesinde analizlerini tamamlamıştır. WEKA üzerinde ise Apriori algoritmasının her iki uygulama ve Fp-Growth algoritmasıyla kıyaslandığında çok daha yavaş çıktı verdiği görülüştür.



Şekil 26 *Algoritma Süreleri- Python*



Şekil 27*Algoritma Süreleri-WEKA*

Dönen veriler açısından WEKA üzerinde her iki algoritmanın verdiği çıktı birebir aynı olmuştur. Python üzerinden ise her iki algoritmanın çıktı olaraka verdiği en yüksek olasıktaki ürün ilişkisi ve değerleri birebir aynı çıkmış fakat daha düşük confidence değerlerinde Fp-Growth algoritması Apriori algortimasına oranla farklı ilişkileri de listelemiştir.

WEKA ve benzeri toollar masaüstü uygulamaları olmalarından dolayı herhangi bir sisteme entegre edilememektedir. Python gibi veri madenciliği alanında birçok kütüphane sunan açık kaynak kodlu bir dilin kullanılması bu çalışmada görüldüğü gibi masaüstü uygulamalarına nazaran daha hızlı sonuçlar vermektedir. Büyük ya da küçük ölçekli işletmelerin dijital dünyanın hızına ayak uydurma girişimlerinde ileriye dönük python kütüphaneleri ile yazılacak servisler ile analiz sonuçlarının çok kısa sürede paylaşarak destek olunabileceğini düşünüyoruz.

8. Kaynaklar

(2020, Aralık 13). apyori: https://pypi.org/project/apyori/ adresinden alındı

(2020, 12 13). pyfpgrowth: https://pypi.org/project/pyfpgrowth/ adresinden alındı

A., R. A., V., K. R., & D., J. B. (2012). Association Rule – Extracting Knowledge Using Market Basket Analysis.

Agarwal, P., Yadav, M. L., & Anand, N. (2013). Study on Apriori Algorithm and its Application in. *74*.

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining Association Rules between Sets of Items in Large Databases. Proceedings of the 1993 ACM SIGMOD international conference on Management of data.

Aguinis, H., Forcum , L. E., & Joo, H. (2012). Using Market Basket Analysis in Management Research.

Akbulut, M. C. (2020, Aralık 1). acikders.ankara.edu.tr: https://acikders.ankara.edu.tr/pluginfile.php/73932/mod_resource/content/1/Unite_10.pdf adresinden alındı

Akpınar, H. (2000). VERİ TABANLARINDA BİLGİ KEŞFİ ve VERİ MADENCİLİĞİ. *İ. U. işletme Fakültesi Dergisi*, 1-22.

Albayrak, M. (2017). Bilimsel araştırmalarda veri madenciliği kullanımı. *International Journal of Social Sciences and Education Research*.

Al-Maolegi, M., & Arkok, B. (2014). AN IMPROVED APRIORI ALGORITHM FOR ASSOCIATION RULES. *3*.

Apriori-Algorithm. (2020, Aralık 12). 08 30, 2020 tarihinde lastnightstud: http://www.lastnightstudy.com/Show?id=68/Apriori-Algorithm adresinden alındı

Birant, D., Kut, A., Ventura, M., Altınok, H., Altınok, B., Altınok, E., & Ihlamur, M. (2010). İş Zekası Çözümleri için Çok Boyutlu Birliktelik Kuralları Analizi. *Akademik Bilişim'10 - XII. Akademik Bilişim Konferansı Bildirileri.*

Chai, S., Yang, J., & Cheng, Y. (2007). The Research of Improved Apriori. Service System and Service Management, 2007 International Conference.

Chang, R., & Liu, Z. (2011). An improved apriori algorithm. Electronics and Optoelectronics (ICEOE), 2011 International Conference.

Changsheng, Z., Zhongyue, L., & Dongsong, Z. (2009). An Improved Algorithm for Apriori. Education Technology and Computer Science,2009. ETCS '09. First International Workshop.

Chen, Y. L., Tang, K., Shen, R.-J., & Hu, Y.-H. (2004). Market basket analysis in a multiple store environment.

Christian, B. (2005). An Implementation Of The Fp-Growth Algorithm. Association for Computing MachineryNew YorkNYUnited States.

Çelik, B. (2019). Mekanik Arızaların Veri Madenciliği Apriori Algoritması ile Analiz Edilmesi.

Dunham, M. H., Xiao, Y., Gruenwald, L., & Hossain, Z. (2000). A SURVEY OF ASSOCIATION RULES. *Technical Report, Southern Methodist University, Department of Computer Science, Technical Report TR 00-CSE-8.*

Erdem, S., & ÖZDAĞOĞLU, G. (2008). ANALYZING OF EMERGENCY DATA OF A TRAINING AND RESEARCH.

Eşmekaya, E., & Şeker, Ş. E. (2017). ETL Süreçleri (ETL Processes). *YBS Ansiklopedi*, 18-34.

Fp-Growth-Algorithm. (2020, Aralık 1). 08 30, 2020 tarihinde http://www.lastnightstudy.com: http://www.lastnightstudy.com/Show?id=69/Fp-Growth-Algorithm adresinden alındı

Gürgen, G. (2008). Birliktelik Kuralları İle Sepet Analizi ve Uygulaması. *Yüksek Lisans Tezi, Marmara Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul.*

Han, J. P. ((2011)). *Data mining: concepts and techniques.* Elsevier.

Han, J., & Kamber, M. (2000). *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers.

Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques 3rd Edition.* Morgan Kaufmann.

Heaton, J. (2016). Comparing Dataset Characteristics that Favor the Apriori, Eclat or FP-Growth Frequent Itemset Mining Algorithms.

Hossain, M., Sattar, A., & Paul, M. K. (2019). Market Basket Analysis Using Apriori and FP Growth Algorithm.

Huang Yuan, W. X.-C. (2009). Efficiency And Consistency Study On Carma. Fifth International Joint Conference On Inc, Ims And Idc.

Huang Yuan, W. X.-C. (209). Efficiency And Consistency Study On Carma. Fifth International Joint Conference On Inc, Ims And Idc.

Hunyadi, D. (2011). Performance comparison of Apriori and FP-Growth algorithms. Proceedings of the European computing conference.

Jiang, Y., Zhao, M., Hu, C., He, L., Bai, H., & Wang , J. (2018). A parallel FP-growth algorithm on World Ocean Atlas data with multi-core CPU.

Kaur, M., & Kang, S. (tarih yok). Market Basket Analysis: Identify the changing trends of market data using association rule mining. 2016: International Conference on Computational Modeling and Security - Procedia Computer Science.

Kumar, B. S., & Rukmani, K. V. (2010). Implementation of Web Usage Mining Using. *01*(06).

Kumar, B. S., & Rukmani, K. V. (2010). Implementation of Web Usage Mining Using. *Int. J. of Advanced Networking and Applications, 01*(06), Vol.01.

Leeflang, P. S., Verhoef, P. C., Dahlström, P., & Freundt, T. (2014). Challenges and solutions for marketing in a digital era.

Li, N., Zeng, L., He, Q., & Shi, Z. (2012). Parallel Implementation of Apriori Algorithm Based on MapReduce. Kyoto: 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing.

Lin, R.-H. (2009). Potential use of FP-growth algorithm for identifying competitive suppliers in SCM. *60(8)*(pages 1135 -1141).

Mayilvaganan, M., & Kalpanadevi, D. (2018). COMPARISON OF APRIORI, FP-TREE GROWTH AND FUZZY FP-TREE. *6*(2).

Mostafaei, S., Ganjavi, H. S., & Ghodsi, R. (2018). New Approaches to Analyze Gasoline Rationing. *6*.

Mythili, M., & Shanavas, A. (2013). Performance Evaluation of Apriori and FP-Growth Algorithms. *International Journal of Computer Applications*, 0975 – 8887.

Orlando, S., Palmerini, P., & Perego, R. (2001). Enhancing the Apriori Algorithm for Frequent. *2114*.

Orlando, S., Perego, R., & Silvestri, C. (2004). *A new algorithm for gap constrained sequence mining.* ACM Symposium on Applied Computing.

Özdoğan, G. Ö., Abul, O., & Yazıcı, A. (2009). Paralel Veri Madenciliği Algoritmaları. Ankara: 1. Ulusal Yüksek Baúarım Ve Grid Konferansı (ODTÜ).

Pramudiono, I., & Kitsuregawa, M. (2003). Parallel FP-Growth on PC Cluster. *2637*(ISBN : 978-3-540-04760-5).

Racz, B. (2004). An FP-Growth Variation without Rebuilding the FP-Tree. FIMI.

Sağın, A. N., & Ayvaz, B. (2018). Determination of Association Rules with Market Basket Analysis: An Application in the Retail Sector. *7*.

Said, A. M., Dominic, D., & Abdullah, A. B. (2009). A Comparative Study of FP-growth Variations. *9*.

Savasere, A., Omiecinski, E., & Navathe, S. (1995). An Efficient Algorithm for Mining Association Rules in Large Databases. Zurich ,Switzerland: VLDB'95.

Shah, M., Shah, N., Shetty, A., & Shah, D. (2016). A Comparative Study of Pattern Recognition Algorithms on Sales Data. *International Journal of Computer Applications*, Vol 141-No.1.

Sherdiwala, K. B., & Khanna, S. O. (2015, Nisan). Ağustos 26, 2020 tarihinde http://oaji.net/: http://oaji.net/articles/2015/1250-1428333425.pdf adresinden alındı

Singh, J., Ram, H., & Sodhi, D. (2013). Improving Efficiency of Apriori Algorithm Using. *3*(1).

Şeker, Ş. E. (2018). CRISP-DM: Endüstriler Arası Standart İşleme – Veri Madenciliği için (Cross Industry Standard Processing – Data Mining) . *YBS Ansiklopedi*, Cilt 5, Sayı 2.

Şen, F. (2008). VERİ MADENCİLİĞİ İLE BİRLİKTELİK KURALLARININ BULUNMASI.

Wikipedia. (2020, November 28). tr.wikipedia.org: https://tr.wikipedia.org/wiki/Veri#cite_note-1 adresinden alındı

Wikipedia. (2020, November 30). https://tr.wikipedia.org/: https://tr.wikipedia.org/wiki/Veri_ambar%C4%B1#cite_note-1 adresinden alındı

Xie, Y., Li, Y., Wang, C., & Lu, M. (2008). The Optimization and Improvement of the Apriori Algorithm. International Workshop on Geoscience and Remote Sensing. ETT and GRS 20008.

Ye , Y., & Chiang, C.-C. (2006). A Parallel Apriori Algorithm for Frequent Itemsets Mining. Seattle: Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06).

Ying, S., Sindakis, S., Aggarwal, S., Chen, C., & Su, J. (2020). Managing big data in the retail industry of Singapore: Examining the impact on customer satisfaction and organizational performance.

Yu, W., Wang, X., Wang, E., & Chen, B. (2008). The research of improved apriori algorithm for mining association rules. Communication Technology, 2008. ICCT 2008 11th IEEE International.

Yuan , X. (2017). An Improved Apriori Algorithm for Mining Association. China: AIP Conference Proceedings.

Yun, C.-H., Chuang, K.-T., & Chen, M.-S. (2006). An Efficient Clustering Algorithm for Market Basket Data Based on Small Large Ratios.

Zeng, Y., Yin, S., Liu, J., & Zhang, M. (2015). Research of Improved FP-Growth Algorithm in Association Rules Mining. *2015*.

Zhou, L., Zhong, Z., Zhong, J., Li, J., Huang , J. Z., & Feng, S. (IEEE Youth Conference on Information). Balanced parallel FP-Growth with MapReduce. Beijing: IEEE Youth Conference on Information, Computing and Telecommunications.