

ARAŞTIRMA MAKALESİ / RESEARCH ARTICLE

DAĞITIK SİSTEMLERDE BİRLİKTELİK KURALLARI İLE SEPET ANALİZİ*

Tuğçe YÜKSEL¹¹İstanbul Aydın Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul
ttug.cee@hotmail.com ORCID No: 0000-0002-1487-6041Metin ZONTUL²²İstanbul Arel Üniversitesi, Mühendislik-Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul
metinzontul@arel.edu.tr ORCID No: 0000-0002-7557-2981**Geliş Tarihi/ Received Date:** 20/02/2018. **Kabul Tarihi/ Accepted Date:** 05/07/2019.**Öz**

Büyük miktardaki veri yığınları içerisinde anlamlı ve kullanılabilir olan bilgileri ortaya çıkarabilmek için veri madenciliği teknikleri ve algoritmaları kullanılmaktadır. Birliktelik kuralları veri tabanı sistemlerinde önceden bilinmeyen, kullanışlı örüntülere ulaşabilmek için kullanılan tekniklerden biridir. Birliktelik kurallarının yaygın olarak kullanıldığı bir alan olan sepet analizi müşterilerin satın aldıkları ürünler arasındaki ilişkileri analiz etmektedir.

Bir iş yükünü parçalara bölerek eş zamanlı olarak gerçekleştiren bilgisayar ağları dağıtık hesaplama sistemleri olarak adlandırılmaktadır. Bu çalışmada birliktelik kurallarını dağıtık sistemlere taşıyarak daha kısa süreli sonuçlar elde etmek amaçlanmıştır. FP- Growth algoritması kullanılarak kurallar oluşturulmuş, dağıtık sistem mimarisinde büyük boyutlu veriler işlenerek üzerinde anlık sepet analizi yapılmış ve daha hızlı sonuçlara ulaşılmıştır.

Anahtar Kelimeler: Dağıtık Sistemler, Sepet Analizi, FP-Growth Algoritması, Birliktelik Kuralı, Veri.

BASKET ANALYSIS WITH ASSOCIATION RULES IN DISTRIBUTED SYSTEMS**Abstract**

Data mining techniques and algorithms are used to reveal information that is meaningful and usable in large amounts of data. Association rules are one of the techniques used to access previously unknown, useful patterns in database systems. Basket analysis, a field where association rules are widely used, analyzes the relationships between the products purchased by customers.

Computer networks that perform a workload simultaneously by dividing them into parts are called distributed computing systems. In this study, it is aimed to obtain the shorter results by moving the association rules to distributed systems. Rules were created by using FP- Growth algorithm, large-scale data were processed in distributed system architecture and instant basket analysis was performed and faster results were obtained.

Keywords: Distributed Systems, Basket Analysis, FP-Growth Algorithm, Association Rule, Data.

* Bu makale büyük ölçüde "Dağıtık Sistemlerde Birliktelik Kuralları İle Sepet Analizi, İstanbul Aydın Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Yüksek Lisans Tezi,2018" den yararlanılarak hazırlanmıştır.

1. GİRİŞ

İçinde bulunduğumuz Bilgi ve Teknoloji çağında verilerin hızlı bir şekilde işlenerek, gelişmiş veri analizinin yapılabilmesi, verilerin önemini her geçen gün daha da artırmaktadır. Veri analizi ile daha anlamlı sonuçlar elde edilerek, veriler daha etkili bir şekilde kullanılmaktadır (Sönmez, Zontul ve Kaynar, 2018). Büyük miktardaki verilerin içinden geleceğe dair tahmin yapılabilmesini sağlayan ilişkiler veri madenciliği ile analiz edilmektedir (Savaş, Topaloğlu ve Yılmaz, 2012). Veri madenciliği teknikleri ve algoritmaları ile kullanışlı bilgiler ortaya çıkarılmaktadır (Gemici, 2012).

Birliktelik kuralları büyük miktardaki veri yığınları arasından daha önceden bilinmeyen örüntüleri bulmak için kullanılan tekniklerden biridir (Erdoğan, Gülcan ve Karamaşa, 2015). Birliktelik kurallarının kullanıldığı en belirgin örnek olarak sepet analizi verilebilir. Sepet analizi ile müşterilerin alışverişlerde satın aldıkları ürünler arasındaki ilişkiler analiz edilerek, satın alma alışkanlıkları belirlenmektedir (Takçı ve Hayta, 2014).

Büyük boyutlu hesaplama problemlerinin parçalara ayrılarak, birbirlerine bir bilgisayar ağı ile bağlı olan makinelerde her bir parçanın çözülmesine dağıtık hesaplama adı verilir (Kuzu, 2014). Bu çalışmada diğer çalışmalardan farklı olarak büyük veriyi işleyebilmek ve anlık sepet analizi yapabilmek için dağıtık sistemler kullanılmıştır. Veriler birden fazla işlemci üzerinde dağıtılmış ve FP-Growth algoritması kullanılarak birliktelik kuralları daha hızlı bir şekilde oluşturulmuştur.

Literatürde birliktelik kuralları ve sepet analizi üzerine çeşitli çalışmalar yapılmıştır. Ayhan Döşlü tarafından 2008 yılında yazılan yüksek lisans tezinde, veri madenciliğinde yer alan temel kavramlar, yöntem ve teknikler incelenerek, birliktelik kuralları ve bu kuralların oluşturulması için kullanılan algoritmalar araştırılmıştır. Uygulama sonucunda, FP-Growth algoritmasının Apriori algoritmasına göre daha iyi performans gösterdiği tespit edilmiştir (Döşlü, 2008).

Ufuk Ekim tarafından 2011 yılında yapılan çalışmada, öğrenci başarısına etki eden faktörlerin birlikte bulunması amacıyla Selçuk Üniversitesi otomasyonundaki anket verilerine, Karar Ağacı ve Apriori algoritmaları uygulanarak sonuçlar karşılaştırılmıştır (Ekim, 2011).

Mine Durdu 2012 yılındaki yüksek lisans tezinde, müşteri verileri ve satış bilgilerini anlamlı verilere dönüştürerek, market sepet analizi uygulamıştır. Apriori algoritmasını kullanarak market veri seti üzerinden birliktelik kurallarını oluşturan bir uygulama tasarlamıştır (Durdu, 2012).

Elif Şafak Sivri 2015 yılında hazırladığı yüksek lisans tezinde, bir e-ticaret sitesine ait giyim verilerini kullanarak birliktelik analizi yapmıştır. Müşterilerin birlikte satın aldığı ürünler FP- Growth ve Apriori algoritmaları ile tespit edilmiştir (Sivri, 2015).

Gamze Yılmaz Erduran tarafından 2017 yılında hazırlanan doktora tezinde, Türkiye’de faaliyet gösteren bankalara ait online müşteri şikayetleri üzerinde veri madenciliği uygulanmış, FP- Growth algoritması kullanılarak birliktelik kuralı oluşturulmuştur (Erduran, 2017).

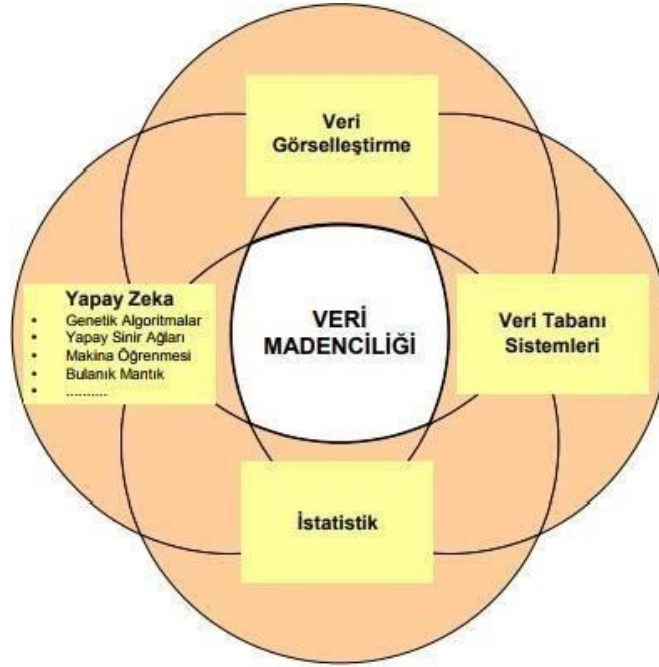
Çalışmanın ikinci bölümünde veri madenciliği hakkında bilgi verilmiş, üçüncü bölümde ise birliktelik kuralı ve sepet analizi açıklanarak, birliktelik kuralı çıkarma algoritmalarından FP- Growth algoritması

incelenmiştir. Dördüncü bölümünde dağıtık sistemler hakkında bilgi verilen çalışmanın beşinci bölümünde, işlemcilerde dağıtık mimari kullanılarak FP-Growth algoritması ile biriktelik kuralı yapan bir uygulama geliştirilmiştir. Çalışma, sonuçların aktarıldığı altıncı bölüm ile sona ermektedir.

2. VERİ MADENCİLİĞİ

Veri madenciliği geçerli tahminler yapmak için çeşitli analiz aracı kullanılarak veri içerisindeki örüntü ve ilişkileri keşfetmeye yarayan bir süreçtir (Koyuncugil ve Özgülbaş, 2009). 1990'lı yıllardan itibaren kullanılan veri madenciliği, Gregory Piatetsky-Shapiro ve William Frawley tarafından "verideki gizli, önceden bilinmeyen ve potansiyel olarak faydalı enformasyonun önemsiz olmayanlarının açığa çıkarılması" şeklinde tanımlanır (Koyuncugil, 2006).

Şekil 1'de ilgi alanları verilen veri madenciliği istatistik, veri görselleştirme, makine öğrenimi, yapay zekâ ve örüntü tanımlama gibi çeşitli alanlarda kullanılmaktadır (Hand, 1998). Çeşitli amaçlarla pek çok alanda kullanılan veri madenciliğinde veri tanımlama, sınıflandırma, kümeleme, ilişkilendirme, tahmin etme gibi uygulama türleri kullanımı yaygındır (Larose, 2005).



Şekil 1. Veri madenciliği ilgi alanları (Baykasoğlu, 2005)

Veri madenciliği veri, bilimsel hesaplamalar, donanım, bilgisayar ağları ve ticari eğilimler olmak üzere beş ana faktörden etkilenmektedir. Veri ortamlarında bulunan veri miktarı arttıkça sorunlar da artabileceğinden boş veri, belirsizlik, eksik veri, veri tabanı boyutu, artık veri, sınırlı bilgi veri madenciliğinde ortaya çıkabilecek problemlerden bazılarıdır (Savaş, Topaloğlu ve Yılmaz, 2002).

Veri madenciliği algoritmalarından verimli sonuçlar alınabilmesi için, veri madenciliği sürecine başlamadan önce, üzerinde çalışılacak verinin özellikleri detaylı analiz edilmelidir. Genel olarak veri madenciliği sürecinde uygulanan adımlar; problemin tanımlanması, verilerin hazırlanması, modelin kurulması ve değerlendirilmesi, modelin kullanılması, modelin izlenmesi şeklindedir (Savaş, Topaloğlu ve Yılmaz, 2002).

Veri madenciliğinde tahmin edici ve tanımlayıcı modeller kullanılmaktadır (Özekes, 2003). Tahmin edici modelde, sonucu belli olan veriler üzerinden bir model oluşturularak sonuçları belli olmayan veri kümeleri için tahminler yapılır. Tanımlayıcı modellerde ise mevcut verilerde bulunan ve karar vermeye yardımcı olabilecek örüntüler tanımlanır (Şimşek, 2006). Veri madenciliğinde kullanılan modeller; Sınıflama - Regresyon, Kümeleme ve Birliktelik Kuralları olarak ayrılır. Sınıflama ve regresyon tahmin edici, kümeleme ve birliktelik kuralları tanımlayıcı modellerdir (Savaş, Topaloğlu ve Yılmaz, 2002).

Veri madenciliğinde bilgilerin verimli kullanılabilmesini sağlayan tekniklerden biri birliktelik kuralıdır (Ateş ve Karabatak, 2017). Birliktelik kuralı; veri tabanında bulunan nesnelere arasındaki ilişkileri analiz ederek, eş zamanlı olarak gerçekleşebilecek olayları belirlemektedir (Miholca, Czibula ve Crivei, 2018).

3. BİRLİKTELİK KURALI VE SEPET ANALİZİ

İlk defa 1993 yılında market sepet verisi üzerinde uygulanan birliktelik kuralı, veri yığınları içerisinde önceden keşfedilmemiş örüntüleri bulmak için kullanılmaktadır (Ateş ve Karabatak, 2017). Açık ve anlaşılır sonuçlar elde edilen, büyük miktardaki veriler üzerinde çalışılabilen birliktelik kurallarında, yeni bilgi analizinin zaman alıcı olması, doğru özellik sayısı bulunurken zorluk yaşanması, seyrek görülen özelliklerin göz ardı edilmesi gibi sorunlar yaşanabilmektedir (Koyuncugil, 2006).

Literatürde pazar sepet analizi olarak da adlandırılan birliktelik kuralı, müşterilerin çapraz satın alma davranışları ve birlikte satın alınma eğilimi olan ürünler hakkında bilgi vermektir (Erdoğan, Gülcan ve Karamaşa, 2015). "Eğer A ürününü alıyorsa % x ihtimalle B ürününü de almaya da yatkındırlar." biçiminde elde edilen bir sonuç, A ürününü satan bir market için fayda sağlayabilmektedir (Ateş ve Karabatak, 2017)

İşlem / Hareket	Elemanlar
t1	Ekmek, Jöle, Yerfistiği yağı
t2	Ekmek, Yerfistiği yağı
t3	Ekmek, Süt, Yerfistiği yağı
t4	Bira, Ekmek
t5	Bira, Süt

Tablo 1. Birliktelik kurallarını göstermek için örnek veri seti

Yukarıdaki tabloda birliktelik kuralları açısından süpermarket nakit kayıt işlemlerine yönelik veriler gösterilmiştir (Ateş ve Karabatak, 2017). Özel kural formunda olan bu kurallar sol ve sağ olarak birbiriyle bağlantılı iki bölümden oluşmaktadır. Bu iki bölümde nesnelere ya da yapılan iş yer alır ve eğer-sonra ifadeleri aracılığıyla veriler arasındaki ilişkiler gösterilir (Tüzüntürk, 2010).

Sepet analizinde destek ve güven değerleri kullanılarak ürünler arasındaki bağıntı hesaplanmaktadır. Kuralın gücünü ölçmek için güvenilirlik, veri tabanında kuralın ne kadar sıklıkla görüldüğünü belirlemek için ise destek ölçütü kullanılır (Erdoğan, Gülcan ve Karamaşa, 2015). Aşağıda $A \rightarrow B$ kuralı için destek ve güvenilirlik değerlerine ait formüller yer almaktadır:

$$\text{Destek değeri : } P(A \rightarrow B) = \frac{\text{A ve B mallarını satın alan müşteri sayısı}}{\text{Toplam müşteri sayısı}} \quad (1)$$

$$\text{Güven: } P(A/B) = \frac{P(A \wedge B) \{A \text{ ve B mallarını satın alan müşterisayısı}\}}{P(A) \{A \text{ malını satın alan müşteri sayısı}\}} \quad (2)$$

Her iki değer de yüksek olması sürekli ilginç ve önemi yüksek kuralların elde edileceği anlamına gelmediğinden, bir kuralın ne derece ilginç olduğu lift değeri kullanılarak tespit edilmektedir (Ateş ve Karabatak, 2017).. Lift ölçütünün 1'den küçük ya da büyük değerler alması ilginçliğin arttığını, "1" değerini alması ise ilginçliğin olmadığını belirtmektedir (Jabbour, Mazouri ve Sais, 2018). Lift değeri denklemi aşağıda gösterildiği gibi hesaplanmaktadır.

$$\text{Lift } (A \rightarrow B) = \frac{\text{destek } (A \wedge B)}{\text{destek}(A) \cdot \text{destek}(B)} = \frac{P(B/A)}{P(B)} \quad (3)$$

Birliktelik kuralı oluşturmak için Conviction (Kanaat) ve Leverage (Kaldıraç) ölçütleri de kullanılmaktadır. Conviction değeri hesaplanırken, A elemanlarının, B elemanı olmaksızın görülme olasılıkları hesaplanır. Eğer conviction değeri 1 ise A ve B birbirinden bağımsızdır. Conviction değeri 1'den uzak ise ilişkili kural oluşturulabilir (Şeker, 2011) .

$$\text{Conviction } (A \rightarrow B) = \frac{1 - \text{destek}(B)}{1 - \text{güven}(A,B)} \quad (4)$$

Leverage değeri ise bir satış verisi üzerinde A ve B ürünlerinin birlikte satılmasının A ve B'nin ayrı ayrı satılmasından ne kadar fazla olduğunu bulmaktadır (Şeker, 2011).

$$\text{Leverage } (A \rightarrow B) = P(A \text{ ve } B) - (P(A)P(B)) \quad (5)$$

Tablo 1'deki veri setinden oluşturulan bazı birliktelik kuralları için bulunan destek ve güvenilirlik değerleri Tablo 2'de yer almaktadır.

$A \bowtie B$	Destek değeri (s)	Güvenilirlik değeri (a)
Ekmek \bowtie Yerfistiği yağı	%60	%75
Yerfistiği yağı \bowtie Ekmek	%60	%100
Bira \bowtie Ekmek	%20	%50
Yerfistiği yağı \bowtie Jöle	%20	%33,3
Jöle \bowtie Yerfistiği yağı	%20	%100
Jöle \bowtie Süt	%0	%0

Tablo 2. Birliktelik kuralları için destek ve güvenilirlik değerleri

FP-Growth (Frequent Pattern Growth) Algoritması birliktelik kuralı ortaya çıkarmak için tasarlanmış yöntemlerden biridir. Sık örüntüleri bulmak için kullanılan FP-Growth algoritması, tüm veri tabanını sadece iki kez tarayarak maliyeti azaltmaktadır (Erdoğan, 2010).

FP-Growth Algoritması sistem kaynaklarını verimli bir şekilde kullanabilme, yaygın nesne kümelerini aday kümeleri üretmeye gerek olmadan test edebilme ve büyük veri kümelerinde hızlı çalışabilme gibi özelliklere sahip olmasıyla diğer algoritmalarından ayrılır. Böl ve yönet yaklaşımını kullanarak, bir arada sık bulunan nesnelere kümesini ortaya çıkartan FP-Growth algoritmasının sözde kodu Şekil 2'de verilmiştir (Erpolat, 2012).

```

Girdi: FPTree
Çıktı: Tüm sık ögekümleri
Metod: FP-Growth(Fp-tree,null)
FP-Growth(Tree, a)
{
  if Tree tek bir yol P içeriyorsa then
    for each  $\beta$  (P yolu içerisindeki düğümlerin
      kombinasyonu) do
       $sup \leftarrow \min\{b.support \mid b \in \beta\}$ 
      Sık ögekümesi  $\alpha U\{b.item \mid b \in \beta\}$  y1
      sup destek değeri ile üret
    else
      for each item  $\alpha$  ( $\alpha$  in Tree.header)
         $sup \leftarrow a.support$ 
         $\alpha U \alpha$  için sık ögekümlerini sup destek değeri
          ile üret
         $\beta = \alpha U \alpha$  için şartlı örüntü temellerini ve sonra
           $\beta$  için şartlı örüntü ağacını  $Tree_{\beta}$  oluştur
        If  $Tree_{\beta} \neq \emptyset$  then
          FP-Growth( $Tree_{\beta}$ ,  $\beta$ )
}

```

Şekil 2. FP-Growth Algoritması sözde kodu (Han, Pei ve Yin, 2000)

4. DAĞITIK SİSTEMLER

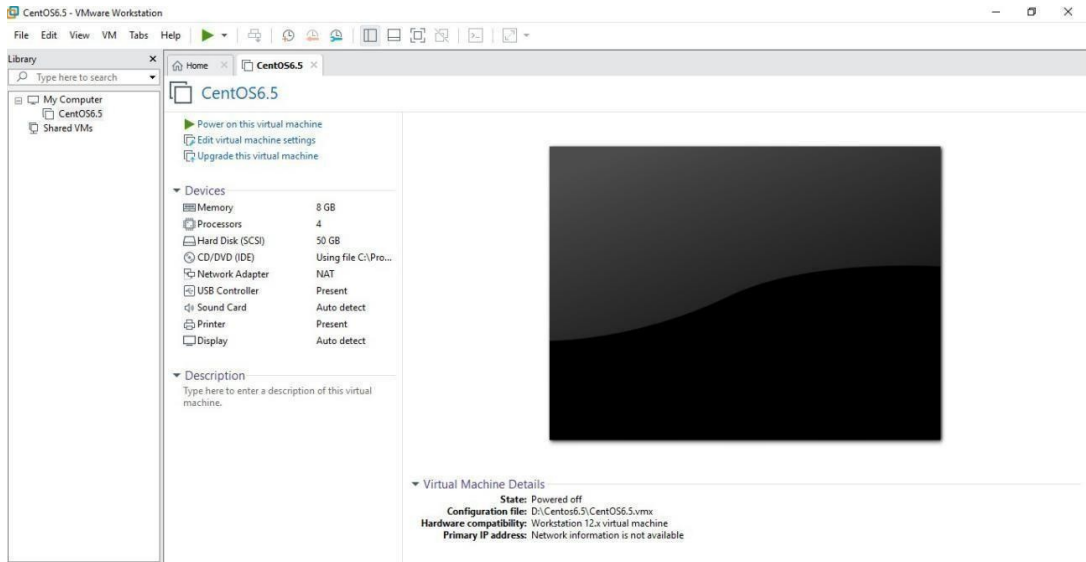
Bir problemin küçük iş parçacıklarına ayrılarak eş zamanlı koordine edilmesi paralel hesaplama olarak adlandırılır. Seri hesaplamada problem her bir zaman aralığında bir komut olmak üzere CPU (Merkezi İşlem Birimi)'ne gönderilirken, paralel hesaplamada öncelikle problem belirli sayıda parçalara ayrılarak her bir problem parçası bir zaman aralığında bir komut olarak CPU'ya gönderilir (Kuzu, 2014).

Dağıtık hesaplama sistemleri büyük bir probleme uygun görev dağılımı ve bağlantı ile ortak çözüm getiren veya bir işi parçalara bölerek eşzamanlı gerçekleştiren bilgisayar ağlarıdır. (Türkoğlu ve Arslan, 2002). Paralel ve dağıtık hesaplama teknolojileri, bilgisayar bilimlerinin üzerinde çalıştığı, ticari ve ekonomik anlamda pek çok araştırmaların yapıldığı etkinliğini sürdüren alanlardan biridir. Bu alanda Hadoop ve MapReduce teknolojileri son yıllarda ön plana çıkmaktadır (Yıldırım, Aydın, Alli ve Tatar, 2014).

5. UYGULAMA

Bu çalışmada bir giyim firmasına ait veriler üzerinde FP-Growth algoritması kullanılarak birliktelik kuralı analizi yapılmıştır. İlk olarak Python Flask ile tek işlemci üzerinde çalışan bir uygulama oluşturulmuş, daha sonra veriler birden fazla işlemciye dağıtılarak birliktelik kuralı analizi yapılmıştır. Orientdb ile veri tabanı oluşturulmuş ve veri tabanı örnek sunucuda veriler saklanmıştır (<http://localhost:2480>). VMware programı ile bilgisayarda sanal bir makine oluşturularak mevcut Windows makine üzerinde Linux işletim sistemi kurulmuştur.

Şekil 3'te başlangıç ekranı verilen sanal makine 8 GB hafıza, 50 GB sabit disk, 4 işlemciye sahiptir.



Şekil 3. VMware Centos6.5 başlangıç ekranı

FP-Growth algoritması kullanılarak birliktelik kuralı oluşturmak için geliştirilen uygulama ilk çalıştırıldığında <http://localhost:5000> veri tabanı sunucusuna bağlantı yapılmasını istemektedir. Sunucuya bağlanıldığında veriler okunmaya başlayacaktır.

```

client = pyorient.OrientDB("localhost", 2424)
client.set_session_token(True)
client.db_open("pcrmdatabase", "root", "123456")
sessionToken = client.get_session_token()

del client

client = pyorient.OrientDB("localhost", 2424)
client.set_session_token(sessionToken)
client.set_session_token(True) # set true
client.db_open("pcrmdatabase", "root", "123456")
new_sessionToken = client.get_session_token()

assert sessionToken != new_sessionToken
## product get category for analysis
class getproductcategory(Resource):
    def get(self):
        myarraylist = []
        print(time.strftime("%H:%M:%S"), 'Data Okunuyor')
        records = client.query("Select list(distinct(product_name)) f")
        print(time.strftime("%H:%M:%S"), 'Data basariyla okundu')
        for x in range(0, len(records)):

```

Şekil 4. Tek işlemcili uygulamanın kod yapısı

```

        for x in range(0, len(records)):
            try:
                myarraylist.append(records[x].oRecordData)

            except:
                print(sys.exc_info()[0])
        print(time.strftime("%H:%M:%S"), 'Fp Growth baslatiliyor')
        trans = read_from_json_basket(myarraylist)

        rules = find_association_rules(trans, 0.5, 0.7)
        a = json.dumps(rules)
        print(time.strftime("%H:%M:%S"), 'Fp Growth basariyla bitirildi.')
        time.sleep(1)
        print(rules)
        return rules

api.add_resource(getproductcategory, '/', methods = ['GET'])

if __name__ == '__main__':
    app.run(debug=True, host='localhost', port=5000)

```

Şekil 5. Tek işlemcili uygulamanın kod yapısı (devam)

Şekil 4 ve Şekil 5'te kodları verilen tek işlemcili uygulama 100 adet veri üzerinde çalıştırıldığında analiz 54 saniye sürmüş ve sonucunda çok sayıda kural üretilmiştir.


```
{
  "consequents": "BL2539 CONSPIRACY RAZOR OUTDRY",
  "confidence": 1.0,
  "lift": 34.5,
  "antecedent": "SL7904 Bugaboo Interchange Jacket",
  "sup_count_AC": 1,
  "sup_count_A": 1,
  "conviction": 0.0,
  "sup_count_C": 2,
  "leverage": 0.014072673808023524
},
```

Şekil 6. Kural 1

Şekil 6'da elde edilen kurala göre ayakkabı alınırken %100 olasılıkla (confidence:1.0) ceket de alınmıştır. İki ürün faturada toplam (sup_count_C) 2 kez birlikte alınmıştır. Elde edilen kuralın lift değeri 34.5 olarak tespit edilmiştir. İki ürünün birlikte satılmasının ayrı ayrı satılmasından ne kadar farklı olduğunu belirlemek için leverage ölçütü kullanılmıştır. Conviction değeri 1'den uzak olduğu için ilişkili bir kural oluşturulabilmiştir. Şekil 7'de uygulamaya ait tüm analiz sonuçları verilmiştir.

```
/home/user/Desktop/Pycharm/tugceproje/virtualenv/bin/python
/home/user/Desktop/Pycharm/tugceproje/webservice/ram_ile_web_uygulamasi.py
* Running on http://localhost:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 143-045-047
09:44:56 Data Okunuyor
09:45:50 Data basariyla okundu
09:45:50 Fp Growth baslatiliyor
09:45:50 Fp Growth basariyla bitirildi.
[{'consequents': 'CL9040 W THERMARATOR GLOVE, SM9481 M Wind Bloc Glove', 'confidence': 1.0, 'lift': 69.0, 'antecedent': 'WL5378 VARALUCK III MID JACKET', 'sup_count_AC': 1, 'sup_count_A': 1, 'conviction': 0.0, 'sup_count_C': 1, 'leverage': 0.014282713715605966}, {'consequents': 'CL9040 W THERMARATOR GLOVE, WL5378 VARALUCK III MID JACKET', 'confidence': 1.0, 'lift': 69.0, 'antecedent': 'SM9481 M Wind Bloc Glove', 'sup_count_AC': 1, 'sup_count_A': 1, 'conviction': 0.0, 'sup_count_C': 1, 'leverage': 0.014282713715605966}, {'consequents': 'SM9481 M Wind Bloc Glove', 'confidence': 1.0, 'lift': 69.0, 'antecedent': 'CL9040 W THERMARATOR GLOVE, WL5378 VARALUCK III MID JACKET', 'sup_count_AC': 1, 'sup_count_A': 1, 'conviction': 0.0, 'sup_count_C': 1, 'leverage': 0.014282713715605966}, {'consequents': 'WL5378 VARALUCK III MID JACKET', 'confidence': 1.0, 'lift': 69.0, 'antecedent': 'CL9040 W THERMARATOR GLOVE, SM9481 M Wind Bloc Glove', 'sup_count_AC': 1, 'sup_count_A': 1, 'conviction': 0.0, 'sup_count_C': 1, 'leverage': 0.014282713715605966}, {'consequents': 'CL9040 W THERMARATOR GLOVE', 'confidence': 1.0, 'lift': 13.799999999999999, 'antecedent': 'WL5378 VARALUCK III MID JACKET, SM9481 M Wind Bloc Glove', 'sup_count_AC': 1,
```

Şekil 7. 100 adet veri ile tek işlemci üzerinde uygulama sonucu

Şekil 8'de tek işlemci üzerinde çalışan uygulamanın dört işlemci üzerinde dağıtılarak çalıştırılmasını sağlayan kodlar verilmiştir.

```

sys.setdefaultencoding().encode('utf-8')
client = pyorient.OrientDB("localhost", 2424)
client.set_session_token(True)
client.db_open("pcrmdatabase", "root", "123456")
sessionToken = client.get_session_token()
del client
client = pyorient.OrientDB("localhost", 2424)
client.set_session_token(sessionToken)
client.set_session_token(True) # set true
client.db_open("pcrmdatabase", "root", "123456")
new_sessionToken = client.get_session_token()
assert sessionToken != new_sessionToken
myarraylist = []
print(time.strftime("%H:%M:%S"), 'Data Listesi Oluşturma baslatildi.')
records = client.query("Select list(distinct(product_name) from "
                        "tbl_product_transaction_line where transaction_group_code in "
                        "(Select transaction_group_code as siparis from "
                        "tbl_product_transaction_line limit 100) group by "
                        "transaction_group_code")

print(time.strftime("%H:%M:%S"), 'Data Liste olusturuldu.')
for x in range(0, len(records)):
    try:
        myarraylist.append(records[x].oRecordData)
    except:
        print(sys.exc_info()[0])
print(time.strftime("%H:%M:%S"), 'Process Start')
def a():
    trans = read_from_json_basket(myarraylist)
    rules = find_association_rules(trans, 0.5, 0.7)
    a = json.dumps(rules)
    time.sleep(1)
    print(rules)
    return rules
results = [a() for i in range(4)]
print(time.strftime("%H:%M:%S"), 'Process Finished')

```

Şekil 8. İşlemcilerde dağıtık mimari oluşturma kodu

Kodları dört işlemci üzerinde 100 adet veriyi dağıtarak çalıştırdığımızda oluşan sonuç Şekil 9 ve Şekil 10'da verilmiştir. Dört işlemci üzerinde 48 saniye sürede bir adet aynı kural üretilmiştir. Dört işlemci de aynı anda çalıştırılmış ve tek işlemci üzerinde çalıştırılan uygulama ile elde edilen kuralın aynısı dört işlemci tarafından daha kısa sürede üretilmiştir.

```

/home/user/Desktop/Pycharm/tugceproje/virtualenv/bin/python
/home/user/Desktop/Pycharm/tugceproje/webservice/islemci_ile_konsol_uygulamasi.py
10:16:39 Data Listesi Oluşturma baslatildi.
10:17:23 Data Liste olusturuldu.
10:17:23 Process Start
[{'sup_count_A': 1, 'sup_count_AC': 1, 'lift': 69.0, 'consequents': 'WL5378 VARALUCK
\ III MID JACKET', 'conviction': 0.0, 'sup_count_C': 1, 'leverage': 0.014282713715605966,
\ 'confidence': 1.0, 'antecedent': 'SM9481 M Wind Bloc Glove'}, {'sup_count_A': 1,
\ 'sup_count_AC': 1, 'lift': 69.0, 'consequents': 'SM9481 M Wind Bloc Glove',
\ 'conviction': 0.0, 'sup_count_C': 1, 'leverage': 0.014282713715605966, 'confidence':
\ 1.0, 'antecedent': 'WL5378 VARALUCK III MID JACKET'}, {'sup_count_A': 1,
\ 'sup_count_AC': 1, 'lift': 69.0, 'consequents': 'CU8204 Columbia Bugascarf',
\ 'conviction': 0.0, 'sup_count_C': 1, 'leverage': 0.014282713715605966, 'confidence':
\ 1.0, 'antecedent': 'CL9724 W Thermarator Glove'}, {'sup_count_A': 1, 'sup_count_AC':
\ 1, 'lift': 69.0, 'consequents': 'CL9724 W Thermarator Glove', 'conviction': 0.0,
\ 'sup_count_C': 1, 'leverage': 0.014282713715605966, 'confidence': 1.0, 'antecedent':
\ 'CU8204 Columbia Bugascarf'}, {'sup_count_A': 1, 'sup_count_AC': 1, 'lift': 34.5,
\ 'consequents': 'AL8996 SATURDAY TRAIL II STRETCH LINED PANT', 'conviction': 0.0,
\ 'sup_count_C': 2, 'leverage': 0.014072673808023524, 'confidence': 1.0, 'antecedent':
\ 'AL6500 Dotswarm II Fleece Full Zip'}, {'sup_count_A': 1, 'sup_count_AC': 1, 'lift':
\ 23.0, 'consequents': 'SM8360 BUGABOO II PANT', 'conviction': 0.0, 'sup_count_C': 3,
\ 'leverage': 0.013862633900441084, 'confidence': 1.0, 'antecedent': 'AM6197 KLAMATH
RANGE FULL ZIP'}, {'sup_count_A': 1, 'sup_count_AC': 1, 'lift': 69.0, 'consequents':
\ 'WM7118 RARE EARTH PARKA', 'conviction': 0.0, 'sup_count_C': 1, 'leverage': 0

```

Şekil 9. 100 adet verinin işlemcilere dağıtılarak uygulamanın çalıştırılması (devam)

```
'sup_count_AC': 1, 'lift': 34.5, 'consequents': 'WM1053 BUGABOO INTERCHANGE JACKET',
'conviction': 0.0, 'sup_count_C': 2, 'leverage': 0.014072673808023524, 'confidence':
1.0, 'antecedent': 'EL4735 Mia Monte II Jacket'}, {'sup_count_A': 1, 'sup_count_AC':
1, 'lift': 69.0, 'consequents': 'CU9857 Thermarator Hat', 'conviction': 0.0,
'sup_count_C': 1, 'leverage': 0.014282713715605966, 'confidence': 1.0, 'antecedent':
'AM6889 Mens Heavyweight Long Sleeve Top'}, {'sup_count_A': 1, 'sup_count_AC': 1,
'lift': 69.0, 'consequents': 'AM6889 Mens Heavyweight Long Sleeve Top', 'conviction':
0.0, 'sup_count_C': 1, 'leverage': 0.014282713715605966, 'confidence': 1.0,
'antecedent': 'CU9857 Thermarator Hat'}, {'sup_count_A': 1, 'sup_count_AC': 1, 'lift':
34.5, 'consequents': 'BL2539 CONSPIRACY RAZOR OUTDRY', 'conviction': 0.0,
'sup_count_C': 2, 'leverage': 0.014072673808023524, 'confidence': 1.0, 'antecedent':
'WM7118 RARE EARTH PARKA'}, {'sup_count_A': 1, 'sup_count_AC': 1, 'lift': 23.0,
'consequents': 'BCK04 CONVERSE BAGCIK 140 CM DESENLİ PAKET', 'conviction': 0.0,
'sup_count_C': 3, 'leverage': 0.013862633900441084, 'confidence': 1.0, 'antecedent':
'3J235C CT KIDS CHUCK TAYLOR ALL STAR CORE/BLACK'}, {'sup_count_A': 1, 'sup_count_AC':
1, 'lift': 69.0, 'consequents': 'CU8204 Columbia Bugascarf', 'conviction': 0.0,
'sup_count_C': 1, 'leverage': 0.014282713715605966, 'confidence': 1.0, 'antecedent':
'CU9857 Thermarator Hat'}, {'sup_count_A': 1, 'sup_count_AC': 1, 'lift': 69.0,
'consequents': 'CU9857 Thermarator Hat', 'conviction': 0.0, 'sup_count_C': 1,
'leverage': 0.014282713715605966, 'confidence': 1.0, 'antecedent': 'CU8204 Columbia
Bugascarf'}}
10:17:27 Process Finished
```

Process finished with exit code 0

Şekil 10. 100 adet verinin işlemcilerle dağıtılarak uygulamanın çalıştırılması (devam)

Farklı boyutlardaki veriler üzerinde uygulama çalıştırılarak deneyler yapılmış ve elde edilen sonuçlar Tablo 3'te gösterilmiştir. Web uygulaması ile işlemcilerle dağıtılarak çalıştırılan uygulama arasındaki kural üretme süresinin veri miktarı arttıkça arttığı görülmüştür.

Veri (Satır)	Süre (Tek işlemci ile)	Süre (4 işlemci ile)	Fark
100	54 saniye	48 saniye	6 saniye
500	4 dakika 3 saniye	3 dakika 15 saniye	48 saniye
1000	7 dakika 23 saniye	6 dakika	1 dakika 23 saniye
2000	14 dakika 31 saniye	12 dakika 7 saniye	2 dakika 24 saniye
3000	20 dakika 21 saniye	17 dakika 7 saniye	3 dakika 14 saniye
4000	33 dakika 9 saniye	27 dakika 21 saniye	5 dakika 48 saniye
5000	44 dakika 7 saniye	31 dakika 29 saniye	12 dakika 38 saniye

Tablo 3. Oluşan fark tablosu

6. SONUÇ

Veri madenciliği kullanılarak işletmelerin daha etkili kararlar alınmasını sağlayan eğilimler ve davranış kalıpları ortaya çıkarılmaktadır. Birliktelik kuralı veri madenciliğinde yararlı ve kullanışlı bilgi elde etmek için kullanılan tekniklerden birisidir. Çalışma kapsamında Python programlama dili kullanılarak FP-Growth algoritması ile veri eklendikçe yeni kurallar üreten bir uygulama geliştirilmiştir. Birliktelik kuralı analizi için,

veriler işlemcilerle dağıtılarak uygulama çalıştırılmış ve daha kısa sürede kurallar oluşturulmuştur. Farklı veri boyutları üzerinde deneyler yapıldığında, veri miktarı arttıkça tek işlemcili uygulama ile dört işlemci üzerinde dağıtık olarak çalışan uygulama arasındaki süre farkının da arttığı görülmüştür.

Conflict of Interests/Çıkar Çatışması

Authors declare no conflict of interests/Yazarlar çıkar çatışması olmadığını belirtmişlerdir.

7. KAYNAKLAR

Apriori ve FP-Growth Algoritmalarının Karşılaştırılması. Anadolu Üniversitesi Sosyal Bilimler Dergisi, 12(1), sayfa. 151-166.

Ateş, Y. ve Karabatak, M. 2017. Nicel Birliklilik Kuralları ile Çoklu Minimum Destek Değeri. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 29(2), sayfa. 57-65.

Baykasoğlu, A. 2005. Veri Madenciliği ve Çimento Sektöründe Bir Uygulama. Akademik Bilişim '05- VII. Akademik Bilişim Konferansı. Gaziantep Üniversitesi Endüstri Mühendisliği Bölümü, Gaziantep.

Döşlü, A. 2008. Veri Madenciliğinde Market Sepet Analizi ve Birliklilik Kurallarının Belirlenmesi, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Durdu, M. 2012. Application of Data Mining in Customer Relationship Management Market Basket Analysis in a Retailer Store, Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi, Fen Bilimleri Enstitüsü, İzmir.

Ekim, U. 2011. Veri Madenciliği Algoritmalarını Kullanarak Öğrenci Verilerinden Birliklilik Kurallarının Çıkarılması, Yüksek Lisans Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Konya.

Erdoğan, G.Ö. 2010. Öbek Bilgisayarlarda Paralel FP-Growth Gerçekleştirimi, Yüksek Lisans Tezi, TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

Erdoğan, N.K., Gülcan, B. ve Karamaşa, Ç. 2015. Birliklilik Kuralları ve Uygulamaları: Literatür Taraması (2000-2014). 13. Uluslararası Türk Dünyası Sosyal Bilimler Kongresi, Azerbaycan Devlet İktisat Üniversite, Bakü, Azerbaycan.

Erduran, G.Y. 2017. Online Müşteri Şikayetlerinin Veri Madenciliği ile İncelenmesi, Doktora Tezi, Trakya Üniversitesi, Sosyal Bilimleri Enstitüsü, Edirne.

Erpolat, S. 2012. Otomobil Yetkili Servislerinde Birliklilik Kurallarının Belirlenmesinde

Gemici, B. 2012. Veri Madenciliği ve Bir Uygulaması, Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi, Sosyal Bilimler Enstitüsü, İzmir.

Han, J., Pei, J. ve Yin, Y. 2000. Mining Frequent Patterns Without Candidate Generation, Proc. of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, Texas, United States, Sf. 1-12.

- Hand, D.J.** 1998. Data Mining: Statistics and More?, The American Statistician, Cilt 52,
- Jabbour, S., Mazouri, F. ve Sais, L.** 2018. Mining Negatives Association Rules Using Constraints. The First International Conference On Intelligent Computing in Data Sciences, 127. Sf. 481-488.
- Koyuncugil, A.S.** 2006. Bulanık Veri Madenciliği ve Sermaye Piyasalarına Uygulanması, Doktora Tezi, Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Koyuncugil, A.S. ve Özgülbaş, N.** 2009. Veri Madenciliği: Tıp ve Sağlık Hizmetlerinde Kullanımı ve Uygulamaları. Bilişim Teknolojileri Dergisi, 2(2). Sayfa. 21-30.
- Kuzu, E.** 2014. Hesaplama Ağırlıklı Algoritmaların Programlanmasında Grafik İşlemci (GPU) Kullanımının İncelenmesi, Yüksek Lisans Tezi, Bahçeşehir Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Larose, D. T.** 2005. Discovering Knowledge in Data. An Introduction to Data Mining. Hoboken, New Jersey: John Wiley & Sons, Inc.
- Özkes, S.** 2003. Veri Madenciliği Modelleri ve Uygulama Alanları. İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, 2 (3), sayfa. 65-82.
- Savaş, S., Topaloğlu, N. ve Yılmaz, M.** 2012. Veri Madenciliği ve Türkiye'deki Uygulama Örnekleri. İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, sayfa. 21, 1-23. Sf.112-118.
- Sivri, E.Ş.** 2015. Veri Madenciliği/E-Ticaret İçin Ürün Tavsiye Sistemi Geliştirilmesi, Yüksek Lisans Tezi, İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Sönmez F., Zontul, M. ve Kaynar, O., Tutar, H.** 2018. Anomaly Detection Using Data Mining Methods in IT Systems: A Decision Support Application. Sakarya University Journal of Science, 22(4), ISSN: 2147-835X. S. 1109-1123.
- Şeker S. E.** 2011. Bağlantı Adresi: <http://bilgisayarkavramlari.sadievrenseker.com/2011/09/09/birliktelik-kurallarini-n-pay-olcumleri-interest-measures-for-association-rules/> Erişim Tarihi: Eylül 2018
- Şimşek, U.T.** 2006. Veri Madenciliği ve Müşteri İlişkileri Yönetiminde (CRM) Bir Uygulama. Doktora Tezi, İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul.
- Takçı, H. ve Hayta, Ş.** 2014. Suç Veri Madenciliği Yardımıyla Hırsızlık Suçları Hakkında Kural Çıkarımı, Eleco 2014 Elektrik-Elektronik-Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu, Bursa.
- Türkoğlu, İ. ve Arslan, A.** 2002. Çok Katmanlı Yapay Sinir Ağlarında En İyi Etkinleştirme Fonksiyonu Seçimi İçin Çok Ölçekli Bir Yaklaşım. Anadolu Üniversitesi Bilim ve Teknoloji Dergisi, 3(1), sayfa. 137-142.
- Tüzüntürk, S.** 2010. Veri Madenciliği ve İstatistik. Uludağ Üniversitesi, İktisadi ve İdari Bilimler Fakültesi Dergisi, 29(1), sayfa. 65-90.
- Yıldırım, G., Aydın, G., Alli, H. ve Tatar, Y.** 2014. Hadoop ile Kaos Temelli FCW Optimizasyon Algoritmasının Analizi. Eleco 2014 Elektrik-Elektronik- Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu, Bursa.

